Bayesian Models for Self-organization and Rewiring in Recurrent Networks of Spiking Neurons

by

David Kappel

DISSERTATION

submitted for the degree of

Doctor Technicae

Institute for Theoretical Computer Science

Graz University of Technology

Thesis Advisor

Prof. Dr. Wolfgang Maass

Graz, February 2018

This document is set in Palatino, compiled with pdfLaTeX2e and `Biber`.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, _____ _____

Date Signature

Eidesstattliche Erklärung[1]

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am _____ _____

Datum Unterschrift

[1] Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Abstract

Experimental data have revealed that neural activity and synaptic dynamics are driven to a surprisingly large extend by spontaneous processes that are not correlated with the history of neural activity or the network inputs. If exposed to the same stimulus, neurons respond with quite different activity patterns. Also, neural networks in the brain undergo permanent rewiring that continues in the adult brain and is driven to a large extend by synapse-autonomous processes. These variabilities in synapse and neuron dynamics are seemingly the results of intrinsic stochastic processes, such as random opening and closing of ion channels and spontaneous decay of proteins in synapses and neurons. These results suggest that the brain operates in a regime of high levels of variability, which is not compatible with the classic theory of computation that works with deterministic elements. It has been proposed previously that a Bayesian theory of brain function provides a possible solution to this problem. Bayesian statistics is an elegant mathematical framework to include internal and external noise sources in a single model and to derive rules for neuron and synapse dynamics that best function in the presence of these noise sources. In this thesis the Bayesian framework is applied to models of recurrent spiking neural networks and synaptic rewiring. This approach leads to rules for neuron and synapse dynamics that enable powerful self-organization capabilities in spiking neural networks. First, this modeling framework is applied to a cortical network motif, a network with lateral inhibition and recurrent excitatory connections. The emerging network model and learning rules install capabilities of a hidden Markov model, a well-known statistical model for sequential data, in the network. Computer simulations demonstrate the ability of these networks to automatically detect and acquire sequential patterns in their inputs and enables them to spontaneously reverberate the sequential structure underlying the learned input patterns. The same theoretical framework is then applied to the synaptic dynamics and rewiring of spiking networks. The learning rules that emerge in this case optimize the network through rewiring and synaptic plasticity by performing a guided random walk that automatically adapts the network to solve complex learning tasks. This model is applied to a task for unsupervised and reward-based learning and is able to reproduce a number of experimental finding on spontaneous rewiring and compensation for lesions and perturbations in the brain. These results provide an important step towards understanding the role of noise in the brain and the processes that underlie its self-organization capabilities.

Zusammenfassung

Experimentelle Daten zeigen, dass neuronale Aktivität und synaptische Dynamik zu einem überraschend großen Anteil von autonomen Prozessen, welche nicht mit der Netzwerkaktivität korreliert sind, getrieben werden. Werden Neuronen mehrmals demselben Eingangs-Stimulus ausgesetzt, antworten sie mit sehr unterschiedlichen Aktivitätsmustern. Außerdem konnte experimentell nachgewiesen werden, dass das Gehirn sich permanent umorganisiert, indem synaptische Verbindungen auf- und abgebaut werden. Diese permanente Umorganisation bleibt auch im erwachsenen Gehirn aufrecht. Dieser hohe Anteil an Variabilität in der Dynamik von Synapsen und Neuronen ist scheinbar das Resultat von intrinsischer Stochastizität, wie zum Beispiel dem zufälligen Öffnen und Schließen von Ionenkanälen, oder dem spontanen Zerfall von Proteinen innerhalb der Neuronen und Synapsen. Diese Ergebnisse deuten darauf hin, dass das Gehirn in einem Regime sehr starker Variabilität operiert. Dies ist nicht kompatibel mit der klassischen Berechnungstheorie, die auf deterministischen Elementen aufgebaut ist. Als möglicher Lösungsansatz für dieses Problems wurden Bayes'sche Modelle vorgeschlagen, welche es erlauben Zufallsquellen innerhalb des Netzwerkes und in seinen Eingängen in einem Modell abzubilden. In dieser Dissertation wird die Bayes'sche Rahmentheorie auf rekurrente neuronale Netzwerke angewandt. Dieser Ansatz führt zu einem Regelwerk für die Netzwerkdynamik, welches Neuronen und Synapsen zur Selbstorganisation befähigt. Zuerst wird dieses Modell auf ein rekurrentes Netzwerk angewandt, welches ein statistisches Modell für sequenzielle Daten approximiert. In Computersimulationen wird gezeigt, dass dieses Netzwerkmodell selbständig in der Lage ist sequenzielle Muster in seinen Eingängen zu erkennen und die darunterliegende zeitliche Struktur spontan wiederzugeben. Dieselbe Rahmentheorie wird danach auf die synaptische Dynamik und Umorganisation von Verbindungen angewandt. Daraus resultieren Lernregeln, welche die Verbindungen des Netzwerkes automatisch umorganisieren, indem sie eine zielgerichtete Dynamik ausführen, die von einem Zufallsprozess überlagert ist. Dieses Modell für synaptische Plastizität erlaubt es den Netzwerken komplexe Lernaufgaben zu lösen. Danach wird dieses Modell auf Lernen ohne Supervision und Lernen mit Belohnung angewandt. Die daraus resultierenden Netzwerkmodelle sind in der Lage experimentelle Ergebnisse zu Selbstorganisation und Kompensation von Lesionen und Perturbationen zu reproduzieren. Diese Ergebnisse sind ein wichtiger Schritt um die Rolle von Zufallsprozessen im Gehirn und seine Fähigkeit zur Selbstorganisation zu verstehen.

Acknowledgements / Danksagung

I would like to thank my supervisor Wolfgang Maass for his support and guidance, but also for giving me the freedom to work on my own ideas. I also thank Christian Mayr for taking the time to be the second referee of this thesis. I would like to thank Robert Legenstein and Bernhard Nessler for their valuable input, their feedback and their patients during our joint projects. I also want to express my gratitude to the administrative staff of the university, first and foremost Daniela Potzinger, Charlotte Rumpf, Oliver Friedl and Nicoletta Kähling for their help and great support. Thank you, Zeno Jonke, Johannes Bill, Stefan Habenschuss, Elmar Rückert, Guillaume Bellec, Arjun Rao, Michael Hoff and Anand Subramoney for the great discussions and for being such supportive and likeable colleagues. Furthermore, I would like to thank Noam Ziv, Jason MacLean, Christos Papadimitriou and Matthew Larkum for the very inspiring discussions and their valuable feedback to my work.

Mein besonderer Dank gilt auch meinen Eltern Eva und Konrad, meiner Schwester Lisa, meinem Bruder Paul und meinen Großeltern Helga, Johanna und Erwin, für ihre Zuversicht, ihre Unterstützung und dafür, dass sie meine Stärken erkannt und gefördert haben.

Contents

1 Introduction **1**
- 1.1 Bayesian models of the brain . . . 2
- 1.2 Spiking neuron models . . . 4
- 1.3 Organization of the thesis . . . 5
- 1.4 Publications not included in this thesis . . . 6
- 1.5 Related work and future developments . . . 6

2 STDP in winner-take-all circuits approximates hidden Markov model learning **9**
- 2.1 Introduction . . . 10
- 2.2 Approximate hidden Markov model learning in spiking neural networks . . . 13
- 2.3 STDP instantiates a stochastic approximation to EM parameter learning 18
- 2.4 A refined EM approximation using rejection sampling . . . 32
- 2.5 Discussion . . . 38

3 Network plasticity as Bayesian inference **43**
- 3.1 Introduction . . . 44
- 3.2 Learning a posterior distribution through stochastic synaptic plasticity 47
- 3.3 Synaptic sampling improves the generalization capability of a neural network . . . 51
- 3.4 Spine motility as synaptic sampling . . . 54
- 3.5 Fast adaptation of synaptic connections and weights to a changing input statistics . . . 57
- 3.6 Inherent network compensation capability through synaptic sampling 61
- 3.7 Discussion . . . 64

4 Reward-based self-configuration of neural circuits **69**
- 4.1 Introduction . . . 70
- 4.2 Synaptic sampling for reward-based synaptic plasticity and rewiring 71
- 4.3 Reward-based rewiring and synaptic plasticity as Bayesian policy sampling . . . 77
- 4.4 Reward-based learning of task-dependent routing of information . . 79
- 4.5 Bayesian perspective on policy sampling . . . 82
- 4.6 A model for task-dependent self-configuration of a recurrent network of spiking neurons . . . 83
- 4.7 Compensation for network perturbations . . . 87
- 4.8 Discussion . . . 91

Appendices 94

A List of publications 97

B Appendix to Chapter 2: STDP in winner-take-all circuits approximates HMM learning 99
B.1 Spiking network model . . . 99
B.2 Details to: Forward sampling in WTA circuits . . . 99
B.3 Details to: STDP installs a stochastic approximation to EM parameter learning . . . 102
B.4 Details to: A refined EM approximation using rejection sampling . . . 104
B.5 Simulations and data analysis . . . 106
B.6 Details to: Learning to predict spike sequences through STDP . . . 107
B.7 Details to: Mixed selectivity emerges in multiple interconnected WTA circuits . . . 108
B.8 Details to: Trajectories in network assemblies emerge for stationary input patterns . . . 108
B.9 Details to: Learning the temporal structure of an artificial grammar model . . . 109
B.10 Details to: Comparison of the convergence speed and performance of the approximate algorithms . . . 109

C Appendix to Chapter 3: Network plasticity as Bayesian inference 111
C.1 Details to: Learning a posterior distribution through stochastic synaptic plasticity . . . 111
C.2 Details to: Figure 3.1 . . . 117
C.3 Details to: Improving the generalization capability of a neural network through synaptic sampling . . . 118
C.4 Details to: Spine motility as synaptic sampling . . . 121
C.5 Details to: to Figure 3.3 . . . 124
C.6 Details to: Fast adaptation to changing input statistics . . . 124
C.7 Details to: Inherent compensation capabilities of networks with synaptic sampling . . . 128

D Appendix to Chapter 4: Reward-based self-configuration of neural circuits 135
D.1 Bayesian framework for reward-modulated learning . . . 135
D.2 Analysis of Bayesian policy sampling . . . 136
D.3 Network model . . . 139
D.4 Synaptic dynamics for the reward-based synaptic sampling model . . . 140
D.5 Online learning . . . 144
D.6 Simulation details . . . 145

Bibliography 153

List of Figures

2.1 Illustration of the network model 13
2.2 Emergence of working memory encoded in neural assemblies through weak HMM learning in a WTA circuit through STDP 20
2.3 Spontaneous replay of pattern sequences 23
2.4 Mixed selectivity in networks of multiple interconnected WTA circuits 26
2.5 Neural trajectories emerge for stationary input patterns 29
2.6 Fast learning of an artificial grammar 30
2.7 Rejection sampling enhances the classification performance of the network 34
2.8 Comparison of the convergence speed and learning performance of different sampling methods 37

3.1 Maximum likelihood (ML) learning vs. synaptic sampling 48
3.2 Priors for synaptic weights improve generalization capability 52
3.3 Integration of spine motility into the synaptic sampling model . . . 56
3.4 Adaptation of synaptic connections to changing input statistics through synaptic sampling 58
3.5 Inherent compensation for network perturbations 60

4.1 Illustration of the theoretical framework 72
4.2 Reward-based routing of input patterns 76
4.3 The temperature parameter controls the exploration speed of policy sampling 81
4.4 Reward-based self-configuration and compensation capability of a recurrent neural network 84
4.5 Contribution of spontaneous and neural activity-dependent processes to synaptic dynamics 89

C.1 Comparison of the learning performance for different priors. 118
C.2 Emergent assembly sequences and functional connectivity in a simplified model of multi-modal sensory integration 130
C.3 Comparison between synaptic sampling and approximate HMM learning 131
C.4 Comparison of the reconstruction performance of networks with different temperatures T for learning 132

D.1 Drifts of neural codes while performance remained constant 148

List of Tables

D.1 Parameters of the synapse model 145

Chapter 1

Introduction

Contents

1.1 Bayesian models of the brain 2
1.2 Spiking neuron models 4
1.3 Organization of the thesis 5
1.4 Publications not included in this thesis 6
1.5 Related work and future developments 6

The human brain is a vast network of around 86 billion neurons connected by trillions of synapses (Herculano-Houzel, 2012; Azevedo et al., 2009). Our understanding of this incredibly complex organ is still very limited, but the knowledge has been growing fast in recent years fueled by innovative experimental methods such as two-photon microscopy or patch-clamp recording, that allow us to take a close look at living neural tissue and even larger sections of a complete living brain (Tao et al., 2015; Packer et al., 2013). At the same time new methods have emerged in statistics and machine learning that provide the computational backbone to automatically analyse large-scale experimental data (T. J. Sejnowski et al., 2014).

One striking result that has emerged from these new methods is the surprisingly large variability in biological neural systems. Experimental data show that biological neurons are rather unreliable, in the sense that they respond with quite different activity patterns to repeated presentations of the same input stimulus (Fiser et al., 2004). This variability is driven, to a large extend, by intrinsic neural properties, for example, by the stochastic opening and closing of ion channels and the unreliability of synaptic release sites (Faisal et al., 2008; Yarom and Hounsgaard, 2011; Clarke, 2012; McDonnell and Ward, 2011; Borst, 2010; Yarom and Hounsgaard, 2011). Another source of variability in the brain is the permanently ongoing rewiring of synaptic connections between neurons (A. J. Holtmaat et al., 2005; Loewenstein et al., 2011; Rumpel and Triesch, 2016). Synapses in the mammalian brain come and go on time scales of hours or days. The dynamics that underlies this process was found to be surprisingly stochastic, i.e. purely stochastic contributions to network rewiring explain more than 50 % of the total synaptic dynamics (Dvorkin and N. E. Ziv, 2016).

Whether this variability plays a functional role in the brain or not is subject to an ongoing debate (see Yarom and Hounsgaard, 2011 and Maass, 2014 for a review), but its abundant presence in vivo suggests that the mammalian brain

functions in this noisy regime. This poses an intellectual challenge to computational neuroscience: How can networks of neurons serve a computational function in the presence of such strong noise sources or maybe even benefit from them?

Several authors have suggested that Bayesian models are a suitable theoretical framework to tackle this problem (Knill and Pouget, 2004; Doya, 2007). Bayesian models support integrating uncertainties from sensory inputs and intrinsic variabilities of the neural network, and derive network dynamics and learning rules that function in the presence of these uncertainties. This thesis develops a Bayesian approach to learning and self-organization in recurrent spiking neural networks. We provide a rigorous theoretical analysis of the resulting models for synapse and neuron dynamics deduced from this framework and demonstrate the stability of their network dynamics and learning behavior in noisy environments using in silico experiments.

1.1 Bayesian models of the brain

Bayesian theory uses probabilities to model uncertainty (Neal, 2012). In contrast to conventional statistical models (the *frequentist approach*), where probabilities are only used to model uncertainty in the observations, a Bayesian approach also maintains an estimate about the uncertainty of the model. For example, consider a sequence of observations or measurements $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots\}$. In the frequentist approach one would estimate the parameters $\boldsymbol{\theta}$ of a conditional probability distribution $p(\boldsymbol{X} \mid \boldsymbol{\theta})$ that keeps track of how likely outcomes of the measurements are (e.g. mean and variance of a normal distribution). In addition, the Bayesian framework maintains a belief about the uncertainty of the parameters, which is captured in a *prior distribution* $p(\boldsymbol{\theta})$.

Bayes' theorem enables us to draw conclusions in the realm of uncertainty. For the example outlined above we can recover the distribution over values for the parameters $\boldsymbol{\theta}$ for a given set of observations $\boldsymbol{X}$ using the simple relation

$$p(\boldsymbol{\theta} \mid \boldsymbol{X}) = \frac{p(\boldsymbol{X} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\sum_{\boldsymbol{\theta}'} p(\boldsymbol{X} \mid \boldsymbol{\theta}')\, p(\boldsymbol{\theta}')} = \frac{p(\boldsymbol{X} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\boldsymbol{X})}\,. \tag{1.1}$$

Instead of a fixed value for the parameters $\boldsymbol{\theta}$ we recover a probability distribution $p(\boldsymbol{\theta} \mid \boldsymbol{X})$ that informs us about how likely each outcome of the parameters is for the given data $\boldsymbol{X}$ and prior distribution $p(\boldsymbol{\theta})$. A Bayesian theory of the brain assumes that all conclusions are drawn based on Bayesian inference, not just on the level of cognition, but more importantly also unconscious processes follow this strategy (Knill and Pouget, 2004; Doya, 2007). Probably the first to describe the ability of the human brain to draw this kind of unconscious conclusions was *Hermann von Helmholtz* in his seminal book "Handbuch der physiologischen Optik" where we noted

> *"Indessen mag es erlaubt sein, die psychischen Acte der gewöhnlichen Wahrnehmung als* unbewusste Schlüsse *zu bezeichnen, da dieser Name sie hinreichend von den gewöhnlich so genannten bewussten Schlüssen unterscheidet, und wenn auch die Aehnlichkeit der psychischen Thätigkeit in beiden bezweifelt worden ist, und vielleicht auch bezweifelt werden wird, doch die Aehnlichkeit der Resultate solcher unbewussten und der bewussten Schlüsse keinem Zweifel unterliegt."*

translation (Helmholtz and Southall, 2005):

> *"Still it may be permissible to speak of the psychic acts of ordinary perception as* unconscious conclusion*, thereby making a distinction of some sort between them and the common so-called conscious conclusion. And while it is true that there has been, and probably always will be, a measure of doubt as to the similarity of the psychic activity in the two cases, there can be no doubt as to the similarity between the results of such unconscious conclusions and those of conscious conclusions."* Helmholtz, 1867

To date, numerous experimental studies support Helmholtz' intuition about sensory perception (Liu et al., 1995; Eagle and Blake, 1995; Knill, 1998; Ee et al., 2003), and similar results have been found for sensorimotor integration (Wolpert et al., 1995; Harris and Wolpert, 1998; Beers et al., 2001; Beers et al., 2002). Furthermore, several models have been proposed to ground the Bayesian brain hypothesis on computational neuron models (briefly discussed in Sec. 1.5). These models are concerned with the question of how random variables and their uncertainties are represented in neurons or populations of neurons (Knill and Pouget, 2004; Doya, 2007). A successful Bayesian neuron model has to be able to compute the elementary operations of probability calculus, i.e. marginals (the denominator of Eq. (1.1)) and conditionals (the numerator of Eq. (1.1)).

In this thesis I explore an approach to Bayesian inference in neural networks that has entered the literature under the name *neural sampling*. The neural sampling hypothesis makes the natural assumption that uncertainties are encoded in the variability of neural activity. This approach has been explored in a number of studies on Bayesian inference (Buesing et al., 2011; Pecevski et al., 2011; Savin and Deneve, 2014; Hennequin et al., 2014) and learning (Nessler et al., 2013; Pecevski and Maass, 2016). In this theoretical framework each neuron or population of neurons represents a binary variable and their activity encodes one particular outcome of that variable. In contrast to approaches based on convolutional codes (reviewed in Sec. 1.5) where all possible solutions and their likelihoods are encoded simultaneously in a deterministic neuron model, the network randomly switches between different solutions but spends most time in the most likely states. The neurons therefore use the time domain to encode uncertainties, such that the time spend in a certain state corresponds to the likelihood of that particular state. This has the advantage that the model inherently uses the noisy of neural responses to explore different solutions.

Furthermore, neural sampling provides simple means to compute marginals by just observing a subset of the state space (encoded by a subpopulation of neurons). The state space of the remaining network is naturally marginalized over by just allowing the network to sample from its intrinsic probability distribution. Also, conditional distributions can be computed easily by fixing the activity of a subset of neurons and observing the activity of the remaining network (Buesing et al., 2011). A potential problem of neural sampling is that the convergence to a state with high probability may take quite long. However, we show in this thesis using computer simulations that sampling in spiking neural networks is fast enough to solve quite complex tasks on biologically realistic time scales. We further extend the sampling model to learning problems, such that also synaptic efficacies and network connectivities realize a sampling process from a posterior distribution using a similar theoretical framework.

1.2 Spiking neuron models

The main emphasis of this thesis is on analyzing the dynamics of spiking neuron and synapse models during learning in the presence of intrinsic noise.

The most common form of communication between neurons in the mammalian brain is through brief stereotypical current pulses called action potentials (or spikes). When a neuron receives a sufficiently strong stimulus that depolarizes the membrane potential near the axon hillock, an action potential is triggered that propagates across the axon to adjacent neurons. Action potentials are "all-or-none" events, meaning that their waveform is independent of the stimulus amplitude that arrives at the soma of the neuron (Barnett and Larkman, 2007) (although experimental data exists which show that the waveform can be modulated after the generation of a spike, see e.g., Sasaki et al., 2011). It is therefore commonly believed that the information conveyed from one neuron to the other is encoded by the timing of action potentials, the frequency or the probability of triggering an action potential (Abeles, 1991; Maass, 1997).

Spiking neuron models capture the generation of action potentials. They were pioneered by Alan Lloyd Hodgkin and Andrew Fielding Huxley (Hodgkin et al., 1952). These early models captured the main electrical features of spike initiation and propagation in a set of cable equations. They were later simplified and refined to also include basic properties of synaptic transmission leading to simple models such as the leaky integrate and fire neuron model, which captures neuron dynamics and spike generation in an abstract form (Gerstner and Kistler, 2002). Although, they come at some degree of biological detail, spiking neuron models are designed to capture particular features of biological neurons while others are simplified or completely suppressed. In this thesis I focus on *point neuron models*, that capture the membrane potential only at a single point (the soma) while assuming linear interactions with synaptic inputs.

Noise can be captured in a spiking neuron model using an activation function $f(u(t))$ of the membrane potential $u(t)$, that denotes the firing probability of a neuron at time t. Spikes are then generated according to a Poisson process with rate $f(u(t))$. This approach provides a simple model for the spike generation with a fixed firing threshold when the membrane potential is superimposed by noise (Gerstner and Kistler, 2002). We use this type of neuron models in all studies presented in this thesis. In the next section I give an overview over the studies provided in the main section of this thesis.

1.3 Organization of the thesis

All results presented in this thesis are based on publications to which I have contributed as first- or co-first author during my PhD studies. A detailed statement about the author contributions is given at the beginning of each chapter. I present in each chapter only the main results, while methods and materials are kept in separate appendices provided at the end of this thesis.

In Chapter 2 we address the question of how recurrent networks of spiking neurons can learn and maintain stable network function in the presence of noise. We reanalysis a network architecture and learning rules that were proposed in Nessler et al., 2013. The network consists of excitatory neurons with lateral inhibition, a network motif that is commonly observed in the mammalian cortex. We show that if one takes also the experimentally observed excitatory lateral connections into account, then network dynamics and learning in this model realize inference and learning of a hidden Markov model, a well-known model for sequential data. We show that these networks are able to learn sequential input patterns and to spontaneously reverberate these patterns.

In Chapter 3 we investigate the question of how neural networks can learn with unreliably synaptic connectivities. We present a theoretical framework that describes the dynamics of synapses as stochastic processes. This model suggests that synapses in the brain do not solve learning problems by converging to a fixed-point solution, as suggested by many other computational models of synaptic plasticity, but it maintains a stochastic equilibrium of connections and their synaptic strengths. We call this model *synaptic sampling*.

Finally, in Chapter 4 we present a model that combines the synaptic sampling model for stochastic rewiring in Chapter 3 with a framework for reward-based learning. We perform a number of experiments that mimic common experimental paradigms from the neuroscience literature. We show that our model is able to qualitatively reproduce the statistics of the experimentally found synapse motility. Furthermore, our model is able to cope with the experimentally observed high levels of spontaneous synapse motility (Dvorkin and N. E. Ziv, 2016), and even benefits from the enhanced exploration driven by the strong variability.

1.4 Publications not included in this thesis

In (Pecevski et al., 2014) we introduce NEVESIM, a simulation tool that is optimized for event-based simulation of neural circuits. It was used to implement the neural sampling in continuous time. I contributed to developing and testing of the software. The paper that was published in Frontiers in Neuroinformatics. (Kappel et al., 2015b) is a conference paper that preceded (Kappel et al., 2015a). It provides the basic idea behind synaptic sampling and first experimental results. In (Rueckert et al., 2016) we explored a model for mental planning in spiking neural networks. I contributed to conducting the experiments and the theoretical analysis of this study which was published in Scientific Reports. In (Yu et al., 2016) we extend the synaptic sampling model to include a momentum term and link the resulting model to a Hamiltonian sampling process. I contributed to conducting the experiments and developing the theory of this study which is currently submitted for publication. In (Bellec et al., 2017) we apply the synaptic sampling model to deep learning models and extend the theory to the case where a fixed number of synapses is maintained. This case is interesting for neuromorphic hardware applications to efficiently use the computational resources. I contributed to developing the theory of this study, which is submitted for publication.

1.5 Related work and future developments

In this thesis we focus on the neural sampling approach to the Bayesian Brain hypothesis. However, a number of alternative approaches have been proposed. Here, I will discuss two of them: Convolutional codes and predictive coding.

Convolutional codes provide a simple mechanism to encode random variables in neural networks. In this approach (usually deterministic) neurons are assigned to a tuning function that represents their preferred outcome of a random variable (Zemel et al., 1998; Zemel and Dayan, 1997; Barber et al., 2003; Pouget et al., 2003; Eliasmith and Anderson, 2004; R. P. Rao, 2004). The outcomes of random variables are encoded in the population activity of multiple neurons, where the activity of each neuron represents the likelihood that its assigned preferred outcome corresponds to the value of the random variable. Likelihood values and uncertainties can be directly read out from the activity amplitudes (or firing rates). Probability calculus in these networks usually requires a suitable wiring pattern between populations representing the random variables. E.g. for computing conditional probabilities the neurons have to compute an element-wise multiplication of population activities.

The *predictive coding* hypothesis states that neural responses are the outcomes of a permanently ongoing alignment of bottom-up sensory inputs and top-down predictions based on more abstract representations in higher brain areas (Clark, 2013). To achieve this alignment across levels of a hierarchy, higher brain areas maintain networks, the outcomes of which are matched against the upward information

stream from lower areas which provide the mismatch between prediction and actual experiences in the form of prediction-error signals (Clark, 2013). Neural activity is, according to this hypothesis, a manifestation of a hierarchical inference process. There exists a close relationship to Bayesian inference and several authors have used this framework or the closely related free energy minimization principle to develop predictive coding models (R. Rao and Ballard, 1999; Friston, 2009; Friston, 2008; Lee and Mumford, 2003; Huang and R. P. Rao, 2011).

The predictive coding hypothesis is compatible with neural sampling and combining these two frameworks would be an interesting topic for future research. In this approach, predictive coding would be on the computational level describing providing a more abstract view on brain function, while neural sampling would provide the algorithmic level that provides details to how the predictive coding model is implemented (Marr and Poggio, 1976). This approach can be further extended using the developments that have emerged in the context of the predictive coding framework, such as active sensing (Friston et al., 2011; Fitzgerald et al., 2014) where the loop to sensing organs is closed using the predictive coding theory, which suggests that attention is focused on areas in the sensory domain which best minimize prediction errors.

Chapter 2

STDP in winner-take-all circuits approximates hidden Markov model learning

Contents

2.1 Introduction 10
2.2 Approximate hidden Markov model learning in spiking neural networks 13
2.3 STDP instantiates a stochastic approximation to EM parameter learning 18
2.4 A refined EM approximation using rejection sampling 32
2.5 Discussion 38

Abstract. In order to cross a street without being run over, we need to be able to extract very fast hidden causes of dynamically changing multi-modal sensory stimuli, and to predict their future evolution. We show here that a generic cortical microcircuit motif, pyramidal cells with lateral excitation and inhibition, provides the basis for this difficult but all-important information processing capability. This capability emerges in the presence of noise automatically through effects of STDP on connections between pyramidal cells in Winner-Take-All circuits with lateral excitation. In fact, one can show that these motifs endow cortical microcircuits with functional properties of a hidden Markov model, a generic model for solving such tasks through probabilistic inference. Whereas in engineering applications this model is adapted to specific tasks through offline learning, we show here that a major portion of the functionality of hidden Markov models arises already from online applications of STDP, without any supervision or rewards. We demonstrate the emergent computing capabilities of the model through several computer simulations. The full power of hidden Markov model learning can be attained through reward-gated STDP. This is due to the fact that these mechanisms enable a rejection sampling approximation to theoretically optimal learning. We investigate the possible performance gain that can be achieved with this more accurate learning method for an artificial grammar task.

Acknowledgments and author contributions. This chapter is based on the manuscript

David Kappel, Bernhard Nessler, Wolfgang Maass (2014). "STDP Installs in Winner-Take-All Circuits an Online Approximation to Hidden Markov Model Learning." *PLoS Computational Biology*.

To this study, I contributed as first author. The study was conceived by DK, BM and WM, with the theory being developed by DK and BM. The experiments were designed by DK, BM and WM, and were conducted by DK. The manuscript was written by DK, BM and WM. The authors thank Stefan Habenschuss and Johannes Bill for helpful comments on the manuscript.

2.1 Introduction

An ubiquitous motif of cortical microcircuits is ensembles of pyramidal cells (in layers 2/3 and in layer 5) with lateral inhibition (Berger et al., 2009; Okun and Lampl, 2008; Avermann et al., 2012). This network motif is called a *winner-take-all* (WTA) circuit, since inhibition induces competition between pyramidal neurons (Douglas and Martin, 2004). We investigate in this article which computational capabilities emerge in WTA circuits if one also takes into account the existence of lateral excitatory synaptic connections within such ensembles of pyramidal cells (Fig. 2.1A). This augmented architecture will be our default notion of a WTA circuit throughout this paper.

We show that this network motif endows cortical microcircuits with the capability to encode and process information in a highly dynamic environment. This dynamic environment of generic cortical mircocircuits results from quickly varying activity of neurons at the sensory periphery, caused for example by visual, auditory, and somatosensory stimuli impinging on a moving organism that actively probes the environment for salient information. Quickly changing sensory inputs are also caused by movements and communication acts of other organisms that need to be interpreted and predicted. Finally, a generic cortical microcircuit also receives massive inputs from other cortical areas. Experimental data with simultaneous recordings of many neurons suggest that these internal cortical codes are also highly dynamic, and often take the form of characteristic assembly sequences or trajectories of local network states (Han et al., 2008; Luczak et al., 2009; Luczak et al., 2007; Ji and Wilson, 2007; Fujisawa et al., 2008; C. D. Harvey et al., 2012). We show in this article that WTA circuits have emergent coding and computing capabilities that are especially suited for this highly dynamic context of cortical microcircuits.

We show that spike-timing-dependent plasticity (STDP) (Caporale and Dan, 2008; Markram et al., 2011), applied on both the lateral excitatory synapses and synapses from afferent neurons, implements in these networks the capability to represent the underlying statistical structure of such spatiotemporal input patterns. This implies the challenge to solve two different learning tasks in parallel. First it is necessary to learn to recognize the salient high-dimensional patterns from the afferent neurons,

which was already investigated in (Nessler et al., 2013). The second task consists in learning the temporal structure underlying the input spike sequences. We show that augmented WTA circuits are able to detect the sequential arrangements of the learned salient patterns. Synaptic plasticity for lateral excitatory connections provides the ability to discriminate even identical input patterns according to the temporal context in which they appear. The same STDP rule, that leads to the emergence of sparse codes for individual input patterns in the absence of lateral excitatory connections (Nessler et al., 2013) now leads to the emergence of context specific neural codes and even predictions for temporal sequences of such patterns. The resulting neural codes are sparse with respect to the number of neurons that are tuned for a specific salient pattern and the temporal context in which it appears.

The basic principles of learning sequences of forced spike activations in general recurrent networks were studied in previous work (Rezende et al., 2011; Brea et al., 2011) and resulted in the finding that an otherwise local learning rule (like STDP) has to be enhanced by a global third factor which acts as an *importance weight*, in order to provide a – theoretically provable – approximation to temporal sequence learning. The possible role of such importance weights for probabilistic computations in spiking neural networks with lateral inhibition was already investigated earlier in (Shi and Griffiths, 2009).

In this article we establish a rigorous theoretical framework which reveals that each spike train generated by WTA circuits can be viewed as a sample from the state space of a *hidden Markov model* (HMM). The HMM has emerged in machine learning and engineering applications as a standard probabilistic model for detecting hidden regularities in sequential input patterns, and for learning to predict their continuation from initial segments (Rabiner, 1989; Murty and Devi, 2011; Bishop, 2006). The HMM is a generative model which relies on the assumption that the statistics of input patterns $\boldsymbol{X} = (\boldsymbol{x}_1 \dots \boldsymbol{x}_M)$ over M time steps is governed by a sequence of hidden states $\boldsymbol{S} = (s_1 \dots s_M)$, such that the m^{th} hidden state s_m "explains" or generates the input pattern $\boldsymbol{x}_m$. We show that the instantaneous state s_m of the HMM is realized by the joint activity of all neurons of a WTA circuit, i.e. the spikes themselves and their resulting postsynaptic potentials. The stochastic dynamics of the WTA circuit implements a *forward sampler* that approximates exact HMM inference by propagating a single sample from the hidden state s_m forward in time (Bishop, 2006; Koller and Friedman, 2009).

We show analytically that a suitable STDP rule in the WTA circuit – notably the same rule on both the recurrent and the feedforward synaptic connections – realizes theoretically optimal parameter acquisition in terms of an online *expectation-maximization* (EM) algorithm (Celeux and Diebolt, 1985; Neal and G. E. Hinton, 1998), for a certain pair $\boldsymbol{S}, \boldsymbol{X}$ *if* the stochastic network dynamics describes the state sequence $\boldsymbol{S}$ upon the input sequence $\boldsymbol{X}$. We further show that when the STDP rule is applied within the approximative forward sampling network dynamics of the WTA circuit, it instantiates a weak but well defined approximation of theoretically optimal HMM learning through EM. This is remarkable insofar as no additional mechanisms are needed for this approximation – it is automatically implemented

through the stochastic dynamics of the WTA circuit, in combination with STDP. In this paper we focus on the analysis of this approximation scheme, its limits and its behavioral relevance.

We test this model in computer simulations that duplicate a number of experimental paradigms for evaluating emergent neural codes and behavioral performance in recognizing and predicting temporal sequences. We analyze evoked and spontaneous dynamics that emerges in our model network after learning an object sequence memory task as in the experiments of (Berdyyeva and Olson, 2009; Warden and Miller, 2010). We show that the pyramidal cells of a WTA circuit learn through STDP to encode the hidden states that underlie the input statistics in such tasks, which enables these cells to recognize and distinguish multiple pattern sequences and to autonomously predict their continuation from initial segments. Furthermore, we find neural assemblies emerging in neighboring interconnected WTA circuits that encode different abstract features underlying the task. The resulting neural codes resemble the highly heterogeneous codes found in the cortex (Rigotti et al., 2013). Furthermore, neurons often learn to fire preferentially after specific predecessors, building up stereotypical neural trajectories within neural assemblies, that are also commonly observed in cortical activity (Han et al., 2008; Luczak et al., 2007; Luczak et al., 2009; W. Xu et al., 2007).

Our generative probabilistic perspective of synaptic plasticity in WTA circuits naturally leads to the question whether the proposed learning approximation is able to solve complex problems beyond simple sequence learning. Therefore we reanalyze data on artificial grammar learning experiments from cognitive science (Conway and Christiansen, 2005), where subjects were exposed to sequences of symbols generated by some hidden artificial grammar, and then had to judge whether subsequently presented unseen test sequences had been generated by the same grammar. We show that STDP learning in our WTA circuits is able to infer the underlying grammar model from a small number of training sequences.

The simple approximation by forward sampling, however, clearly limits the learning performance. We show that the full power of HMM-learning can be attained in a WTA circuit based on the *rejection sampling* principle (Bishop, 2006; Koller and Friedman, 2009). A binary factor is added to the STDP learning rule, that gates the expression of synaptic plasticity through a subsequent global modulatory signal. The improvement in accuracy of this more powerful learning method comes at the cost that every input sequence has to be repeated a number of times, until one generated state sequence is accepted. We show that a significant performance increase can be achieved already with a small number of repetitions. We demonstrate this for a simple and a more complex grammar learning task.

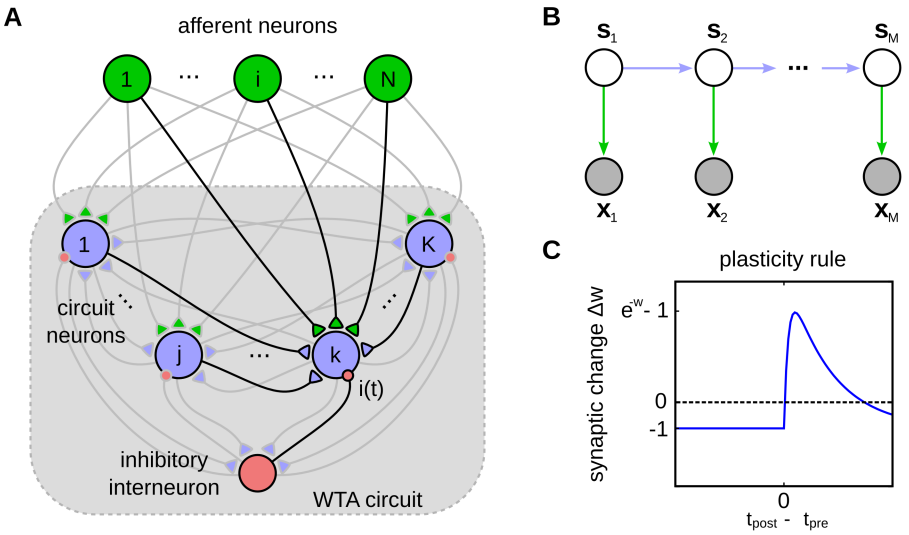

Fig. 2.1: Illustration of the network model. A: The structure of the network. It consists of K excitatory neurons (blue) that receive feedforward inputs (green synapses) and lateral excitatory all-to-all connections (blue synapses). Interneurons (red) install soft winner-take-all behavior by injecting a global inhibition to all neurons of the circuit in response to the network's spiking activity. **B:** The Bayesian network representing the HMM over M time steps. The prediction model (blue arrows) is implemented by the lateral synapses. It determines the evolution of the hidden states s_m over time. The observation model (green arrows) is implemented by feedforward connections. The inference task for the HMM is to determine a sequence of hidden states $\boldsymbol{S} = (s_1 \dots s_M)$ (white), given the afferent activity $\boldsymbol{X} = (\boldsymbol{x}_1 \dots \boldsymbol{x}_M)$ (gray). **C:** The STDP window that is used to update the excitatory synapses. The synaptic weight change is plotted against the time difference between pre- and postsynaptic spike events.

2.2 Approximate hidden Markov model learning in spiking neural networks

We first define the spiking neural network model for the winner-take-all (WTA) circuit considered throughout this paper. The architecture of the network is illustrated in Fig. 2.1A. It consists of stochastic spiking neurons, which receive excitatory input from an afferent population (green synapses) and from lateral excitatory connections (blue synapses) between neighboring pyramidal neurons. To clarify the distinction between these connections, we denote the synaptic efficacies of feedforward and lateral synapses by different weight matrices $\boldsymbol{W} \in \mathbb{R}^{K \times N}$ and $\boldsymbol{V} \in \mathbb{R}^{K \times K}$, respectively, where N denotes the number of afferent neurons and K the size of the circuit (i.e., the number of pyramidal cells in the circuit). In addition, all neurons within the WTA circuit project to interneurons and in turn all receive the same common inhibition $i(t)$. Thus the membrane potential of neuron k at time t is given by

$$u_k(t) = \hat{u}_k(t) - i(t) \quad \text{with} \quad \hat{u}_k(t) = \sum_{i=1}^{N} w_{ki} \cdot x_i(t) + \sum_{j=1}^{K} v_{kj} \cdot y_j(t) + b_k, \tag{2.1}$$

$$\text{with} \quad x_i(t) = \sum_{t'} \epsilon\left(t - t'\right) \quad \text{and} \quad y_j(t) = \sum_{t'} \epsilon\left(t - t'\right), \tag{2.2}$$

where $w_{ki} \cdot x_i(t)$ and $v_{kj} \cdot y_j(t)$ denote the time courses of the excitatory postsynaptic potentials (EPSP) under the feedforward and lateral synapses, where w_{ki} and v_{kj} are the elements of $\boldsymbol{W}$ and $\boldsymbol{V}$ respectively, and b_k is a parameter that controls the excitability of the neuron. The two sums in (2.1) describe the time courses of the membrane potential in response to synaptic inputs from feedforward and lateral synapses. In equation (2.2) we used the assumption of additive EPSPs, where $\epsilon(t)$ denotes a kernel function that determines the time course of an EPSP (Gerstner and Kistler, 2002). The sums run over all spike times of the presynaptic neuron. For the theoretical analysis we used a single exponential decay for the sake of simplicity, throughout the simulations we used double exponential kernels, if not stated otherwise. Our theoretical model can be further extended to other EPSP shapes (see Appendix B for details).

As proposed in (Jolivet et al., 2006), we employ an exponential dependence between the membrane potential and the firing probability. Therefore the instantaneous rate of neuron k is given by $\nu_k(t) = \hat{\nu} \cdot e^{u_k(t)}$, where $\hat{\nu}$ is a constant that scales the firing rate. The inhibitory feedback loop $i(t)$ in equation (2.1), that depresses the membrane potentials whenever the network activity rises, has a normalizing effect on the circuit-wide output rate. Although, each neuron k generates spikes according to an individual Poisson process, this inhibition couples the neural activities and thereby installs the required competition between all cells in the circuit. We model the effect of this inhibition in an abstract way, where we assume, that all WTA neurons receive the same inhibitory signal $i(t)$ such that the overall spiking rate of the WTA circuit stays approximately constant. Ideal WTA behavior is attained if the network rate is normalized to the same value at any point in time, i.e. $\sum_{l=1}^{K} \nu_l(t) = \hat{\nu}$. Using this, we find the circuit dynamics to be determined by

$$\nu_k(t) = \hat{\nu} \cdot e^{u_k(t)} = \hat{\nu} \cdot e^{\hat{u}_k(t) - i(t)} = \hat{\nu} \cdot \frac{e^{\hat{u}_k(t)}}{\sum_{l=1}^{K} \mathrm{e}^{\hat{u}_l(t)}}, \quad \text{with} \quad i(t) = \log \sum_{l=1}^{K} \mathrm{e}^{\hat{u}_l(t)}. \tag{2.3}$$

This ideal WTA circuit realizes a soft-max or soft WTA function, granting the highest firing rate to the neuron with the highest membrane potential, but still allowing all other neurons to fire with non-zero probability.

Recapitulation of hidden Markov model theory

In this section we briefly summarize the relevant concepts for deriving our theoretical results. An exhaustive discussion on hidden Markov model theory can be found in (Rabiner, 1989; Murty and Devi, 2011; Bishop, 2006). Throughout the paper, to keep the notation uncluttered we use the common short-hand notation $p(z)$ to denote $p(Z = z)$, i.e. the probability that the random variable Z takes on the value z. If it is not clear from the context, we will use the notation $p(z \equiv k)$ to remind the reader of the underlying random variable, that is only implicitly defined.

The HMM is a generative model for input pattern sequences over M time steps $\boldsymbol{X} = (\boldsymbol{x}_1 \dots \boldsymbol{x}_M)$ (the input patterns are traditionally called observations in the context of HMMs). It relies on the assumption that a sequence of hidden states $\boldsymbol{S} = (s_1 \dots s_M)$ and a set of parameters $\boldsymbol{\theta}$ exist, which govern the statistics of $\boldsymbol{X}$. This assumption allows to write the joint distribution of $\boldsymbol{X}$ and $\boldsymbol{S}$ as

$$p\left(\boldsymbol{S}, \boldsymbol{X} \mid \boldsymbol{\theta}\right) = \prod_{m=1}^{M} p\left(\boldsymbol{x}_m \mid s_m, \boldsymbol{\theta}\right) p\left(s_m \mid s_{m-1}, \boldsymbol{\theta}\right), \tag{2.4}$$

where we suppress an explicit representation of the initial state s_0, for the sake of brevity. The joint distribution (2.4) factorizes in each time step into the *observation model* $p\left(\boldsymbol{x}_m \mid s_m, \boldsymbol{\theta}\right)$ and the state transition or *prediction model* $p\left(s_m \mid s_{m-1}, \boldsymbol{\theta}\right)$ (Bishop, 2006). This independence property is illustrated by the Bayesian network for a HMM in Fig. 2.1B.

The HMM is a generative model and therefore we can recover the distribution over input patterns by marginalizing out the hidden state sequences $p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right) = \int p\left(\boldsymbol{S}', \boldsymbol{X} \mid \boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{S}'$. Learning in this model means to adapt the model parameters θ such that this marginal distribution $p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right)$ comes as close as possible to the empirical distribution $p^*\left(\boldsymbol{X}\right)$ of the observable input sequences. A generic method for learning in generative models with hidden variables is the *expectation-maximization* (EM) algorithm (Dempster et al., 1977), and its application to HMMs is known as the Baum-Welch algorithm (Baum and Petrie, 1966). This algorithm consists of iterating two steps, the *E-step* and the *M-step*, where the model parameters $\boldsymbol{\theta}$ are adjusted at each M-step (for the updated posterior generated at the preceding E-step). A remarkable feature of the algorithm is that the fitting of the model to the data is guaranteed to improve at each M-step of this iterative process. Whereas the classical EM algorithm is restricted to offline learning (where all training data are available right at the beginning), there exist also stochastic online versions of EM learning.

In its stochastic online variant (Celeux and Diebolt, 1985; Neal and G. E. Hinton, 1998) the E-step consists of generating one sample $\boldsymbol{S}$ from the *posterior distribution* $p\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right)$, given one currently observed input sequence $\boldsymbol{X}$. Given these sampled values for $\boldsymbol{S}$, the subsequent M-step adapts the model parameters θ such that the probability $p\left(\boldsymbol{S}, \boldsymbol{X} \mid \boldsymbol{\theta}\right)$ increases. The adaptation is confined to acquiring the conditional probabilities that govern the observation and the prediction model.

It would be also desirable to realize the inference and sampling of one such posterior sample sequence $\boldsymbol{S}$ in a fully online processing, i.e. generating each state s_m in parallel to the arrival of the corresponding input pattern $\boldsymbol{x}_m$. Yet this seems to be impossible as the probabilistic model according to (2.4) implies a statistical dependence between any s_m and the whole future observation sequence $\boldsymbol{x}_{m+1} \dots \boldsymbol{x}_M$. However, it is well known that the inference of $p\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right)$ can be approximated by a so-called *forward sampling* process (Bishop, 2006; Koller and Friedman, 2009), where every single time step s_m of the sequence $\boldsymbol{S}$ is sampled online, based solely on the knowledge of the observations $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m$ received

so far, rather than the observation of the complete sequence $\boldsymbol{X}$. Hence sampling the sequence $\boldsymbol{S}$ is approximated by propagating a single sample from the HMM state space forward in time.

Forward sampling in WTA circuits

In this section we show that the dynamics of the network realizes a forward sampler for the HMM. We make use of the fact that equations (2.1), (2.2) and (2.3) realize a Markov process, in the sense that future network dynamics is independent from the past, given the current network state (for a suitable notion of network state). This property holds true for most reasonable choices of EPSP kernels. For the sake of brevity we focus in the theoretical analysis on the simple case of a single exponential decay with time constant τ.

We seek a description of the continuous-time network dynamics in response to afferent spike trains over a time span of length T that can be mapped to the state space of a corresponding HMM with discrete time steps. Although the network works in continuous time, its dynamics can be fully described taking only those points in time into account, where one of the neurons in the recurrent circuit produces a spike. This allows to directly link spike trains generated by the network to a sequence of samples from the state space of a corresponding HMM.

Let the M spike times produced during this time window be given by $\hat{t}_1 \dots \hat{t}_M$. The neuron dynamics are determined by the membrane time courses (2.2). For convenience let us introduce the notation $\boldsymbol{y}_m := (y_{m1} \dots y_{mK})$, with $y_{mj} := y_j(\hat{t}_m)$ and by analogy $\boldsymbol{x}_m := (x_{m1} \dots x_{mN})$, with $x_{mi} := x_i(\hat{t}_m)$.

Due to the exponentially decaying EPSPs the synaptic activation $\boldsymbol{y}_m$ at time $\hat{t}_m$ is fully defined by the synaptic activation $\boldsymbol{y}_{m-1}$ at the time of the previous spike $\hat{t}_{m-1}$, and the identity of the neuron that spiked in that previous time step, which we denote by a discrete variable $z_{m-1} \in \{1 \dots K\}$. We thus conclude that the sequence of tuples $\{z_m, \boldsymbol{y}_m, \Delta_m\}$ (with $\Delta_m := \hat{t}_{m+1} - \hat{t}_m$) fulfills the Markov condition, i.e. the conditional independence $p\left(s_m \mid \boldsymbol{x}_1 \dots \boldsymbol{x}_m, s_{m-1}\right) = p\left(s_m \mid \boldsymbol{x}_m, s_{m-1}\right)$ and thus fully represents the continuous dynamics of the network (see Appendix B). We call $s_m := \{z_m, \boldsymbol{y}_m, \Delta_m\}$ the *network state*. The corresponding HMM forward sampler follows a simple update scheme that samples a new state s_m given the current observation $\boldsymbol{x}_m$ and the previous state s_{m-1}. This dynamic is equivalent to the WTA network model.

This state representation allows us to update the network dynamics online, jumping from one spike time $\hat{t}$ to the next. Using this property, we find that the dynamics of the network realizes a probability distribution over state sequences $\boldsymbol{S} = (s_1 \dots s_M)$,

given an afferent sequence $\boldsymbol{X} = (\boldsymbol{x}_1 \dots \boldsymbol{x}_M)$, which can be written as

$$\begin{aligned} q(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}) &= \prod_{m=1}^{M} p(s_m \mid \boldsymbol{x}_m, s_{m-1}, \boldsymbol{\theta}) \\ &= \prod_{m=1}^{M} p(z_m \mid \boldsymbol{x}_m, \boldsymbol{y}_m, \boldsymbol{\theta})\, p(\boldsymbol{y}_m \mid z_{m-1}, \boldsymbol{y}_{m-1}, \Delta_{m-1})\, p(\Delta_{m-1}), \end{aligned} \tag{2.5}$$

where $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{V}, b_1 \dots b_K\}$ is the set of network parameters. The factorization and independence properties in (2.5) are induced by the state representation and the circuit dynamics. We assume here that the lateral inhibition within the WTA circuit ensures that the output rate of the whole circuit is normalized, i.e. $\sum_l \nu_l(t) = \hat{\nu}$ at all times t. This allows to introduce the distribution over the inter-spike-time intervals Δ_m independent from $\boldsymbol{X}$ (see Appendix B for details). Note, that Δ_m determines the interval between spikes of *all* circuit neurons, realized by a *homogeneous* Poisson process with a constant rate $\hat{\nu}$. The second term in the second line of (2.5) determines the course of the membrane potential, i.e. it assures that $\boldsymbol{y}_m$ follows the membrane dynamics. Since the EPSP kernels are deterministic functions this distribution has a single mass point, where (2.2) is satisfied. The first factor in the second line of (2.5) is given by the probability of each individual neuron to spike. This probability depends on the membrane potential (2.1), which in turn is determined by $\boldsymbol{x}_m$, $\boldsymbol{y}_m$ and the network parameters $\boldsymbol{\theta}$. Given that the circuit spikes at time $\hat{t}_m$, the firing probability of neuron k can be expressed as a conditional distribution $p(z_m \equiv k \mid \boldsymbol{x}_m, \boldsymbol{y}_m, \boldsymbol{\theta}) = \mathrm{e}^{u_k(\hat{t}_m)}$. The lateral inhibition in (2.1) ensures that this probability distribution is correctly normalized. Therefore, the winner neuron $k \in \{1 \dots K\}$ is drawn from a multinomial distribution at each spike time.

For the given architecture the functional parts of the network can be related directly to hidden Markov model dynamics. In Appendix B we show in detail that by rewriting $p(z_m \mid \boldsymbol{x}_m, \boldsymbol{y}_m, \boldsymbol{\theta})$ the membrane potential (2.1) can be decomposed into three functional parts

$$\begin{aligned} p(z_m \equiv k \mid \boldsymbol{x}_m, \boldsymbol{y}_m, \boldsymbol{\theta}) \;&=\; \mathrm{e}^{u_k(\hat{t}_m)} \\ &= \frac{\overbrace{\exp\left(\sum_{i=1}^{N} w_{ki} \cdot x_{mi}\right)}^{\text{observation}} \cdot \overbrace{\exp\left(\sum_{j=1}^{K} v_{kj} \cdot y_{mj} + b_k\right)}^{\text{prediction}}}{\underbrace{\exp\left(i\left(\hat{t}_m\right)\right)}_{\text{normalization}}}. \end{aligned} \tag{2.6}$$

The lateral excitatory connections predict a prior belief about the current network activity and the feedforward synapses match this prediction against the afferent input. The inhibition $i(\hat{t}_m)$ implements the normalization that is required to make (2.6) a valid multinomial distribution. The functional parts of the membrane potential can be directly linked to the prediction and observation models of a HMM,

where the network state is equivalent to the hidden state of this HMM. The WTA circuit realizes a forward-sampler for this HMM, which approximates sampling from the posterior distribution $p(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta})$ in an online fashion (Koller and Friedman, 2009). Its sampling is carried out step by step, i.e. it generates through each spike a new sample from the network state space, taking only the previous time step sample into account. Furthermore this forward sampling requires no additional computational organization, but is achieved by the inherent dynamics of the stochastically firing WTA circuit.

2.3 STDP instantiates a stochastic approximation to EM parameter learning

Formulating the network dynamics in terms of a probabilistic model is beneficial for two reasons: First, it gives rise to a better understanding of the network dynamics by relating it to samples from the HMM state space. Second, the underlying model allows us to derive parameter estimation algorithms and to compare them with biological mechanisms for synaptic plasticity. For the HMM, this approach results in an instantiation of the EM algorithm (Dempster et al., 1977; Bishop, 2006) in a network of spiking neurons (stochastic WTA circuit). In Appendix B we derive this algorithm for the WTA circuit and show that the M-step evaluates to weight updates that need to be applied whenever neuron k emits a spike at time $\hat{t}$, according to

$$\Delta w_{ki}(\hat{t}) = \xi \cdot \left(e^{-w_{ki}} x_i(\hat{t}) - 1\right) \quad \text{and} \quad \Delta v_{kj}(\hat{t}) = \xi \cdot \left(e^{-v_{kj}} y_j(\hat{t}) - 1\right), \tag{2.7}$$

where ξ is a positive constant that controls the learning rate. Note that the update rules for the feedforward and the recurrent connections are identical, and thus all excitatory synapses in the network are handled uniformly. These plasticity rules (2.7) are equivalent to the updates that previously emerged as theoretically optimal synaptic weight changes, for learning to recognize repeating high-dimensional patterns in spike trains from afferent neurons, in related studies (Nessler et al., 2010; Habenschuss et al., 2013; Nessler et al., 2013). The update rules consist of two parts: A Hebbian long-term potentiating (LTP) part that depends on presynaptic activity and a constant depression term. The dependence on the EPSP time courses (2.2) makes the first part implicitly dependent on the history of presynaptic spikes. The STDP window is shown in Fig. 2.1C for α-shaped EPSPs. Potentiation is triggered when the postsynaptic neuron fires after the presynaptic neuron. This term is commonly found in synaptic plasticity measured in biological neurons, and for common EPSP windows it closely resembles the shape of the pre-before-post part of standard forms of STDP (Caporale and Dan, 2008; Markram et al., 2011). The dependence on the current value of the synaptic weight has a local stabilizing effect on the synapse. The depressing part of the update rule is triggered whenever the postsynaptic neuron fires independent of presynaptic activity. It contrasts LTP and assures that the synaptic weights stay globally in a bounded regime. It is shown

in Fig. 4 of (Nessler et al., 2013) that the simple rule (2.7) reproduces the standard form of STDP curves when it is applied with an intermediate pairing rate.

While these M-step updates emerge as exact solutions for the underlying HMM, the WTA circuit implements an approximation of the *E-step*, using forward sampling from the distribution in equation (2.5). In the following experiments we will first focus on this simple approximation, and analyze what computational function emerges in the network using the STDP updates (2.7) without any third signal related to reward or a "teacher". In the last part of the Results section we will introduce a possible implementation of a refined approximation, and assess the advantages and disadvantages of this method.

Learning to predict spike sequences through STDP

In this section we show through computer simulations that our WTA circuits learn to encode the hidden state that underlies the input statistics via the STDP rule (2.7). We demonstrate this for a simple sequence memory task and analyze in detail how the hidden state underlying this task is represented in the network. The experimental paradigm reproduces the structure of object sequence memory tasks, where monkeys had to memorize a sequence of movements and reproduce it after a delay period (Shima and Tanji, 2000; Isoda and Tanji, 2003; Berdyyeva and Olson, 2009; Warden and Miller, 2010). The task consisted of three phases: An initial cue phase, a delay phase and a recall phase. Each phase is characterized by a different input sequence, where the cue sequence defines the identity of the recall sequence. We used four cue/recall pairs in this experiment.

The structure of this task is illustrated in Fig. 2.2A. The graph represents a finite state grammar that can be used to generate symbol sequences by following a path from *Start* to *Exit*. In this first illustrative example the only stochastic decision is made at the beginning, randomly choosing one of the four cue phases with equal probabilities while the rest of the sequence is deterministic. On each arc that is passed, the symbol next to the arc is generated, e.g. *AB-delay-ab* is one possible symbolic sequence. Note that all symbols can appear in different temporal contexts, e.g. *A* appears in sequence *AB-delay-ab* and in *BA-delay-ba*. The *delay* symbol is completely unspecific since it appears in all four possible sequences. Therefore this task does not fulfill the Markov condition with respect to the input symbols, e.g. knowing that the current symbol is *delay* does not identify the next one as it might be any of *a*,*b*,*c*,*d*. Only additional knowledge about the temporal context of the symbol allows to uniquely identify the continuation of the sequence.

This additional knowledge can be represented in a hidden state that encodes the required information, which renders this task a simple example of a HMM. The hidden states of this HMM have to encode the input patterns and the temporal context in which they appear in order to maintain the Markov property throughout the sequences, e.g. a distinct state $s_{B,AB}$ encodes pattern *B* when it appears in sequence *AB-delay-ab*. The temporal structure of the hidden state can be related to

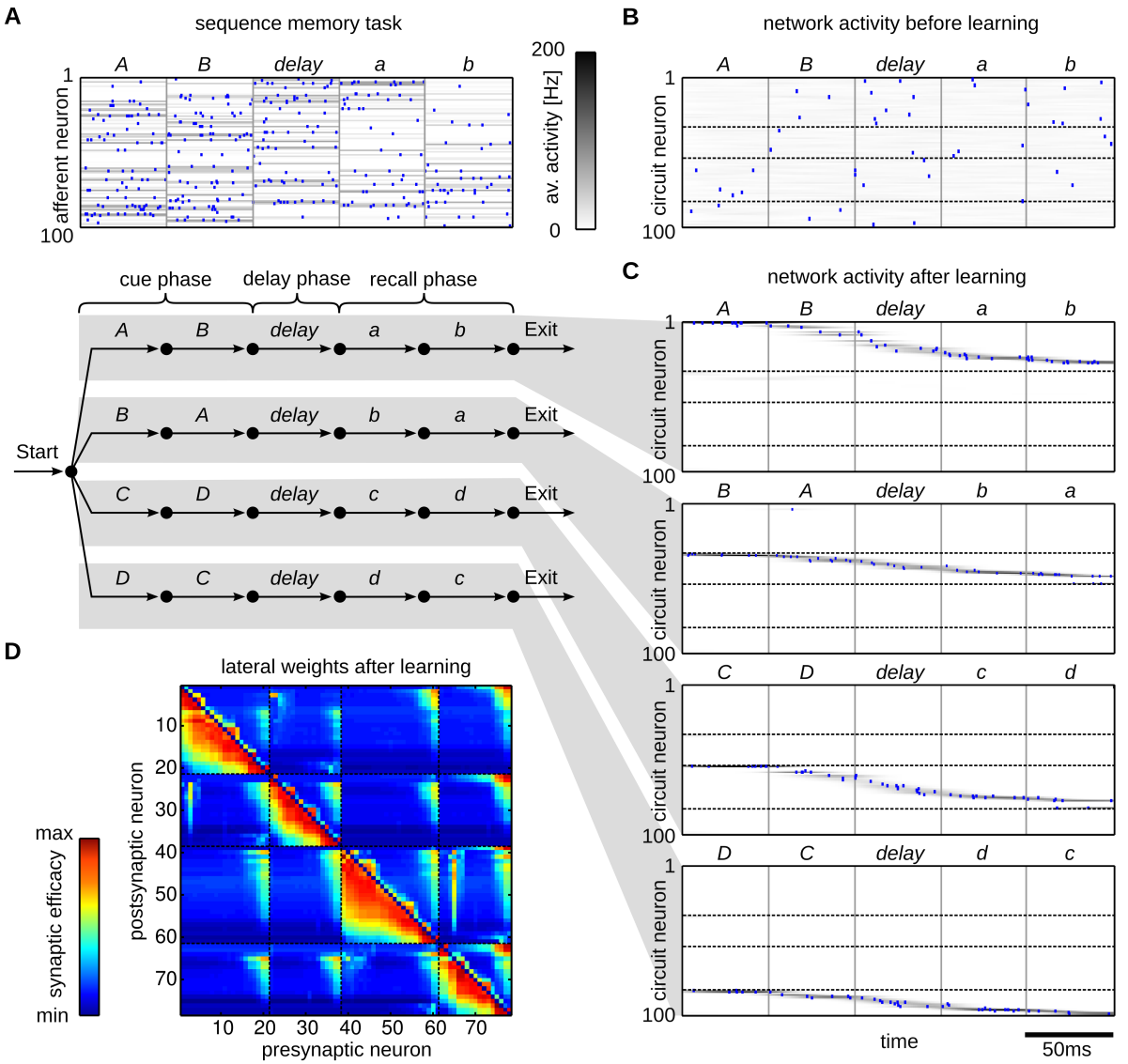

Fig. 2.2: Emergence of working memory encoded in neural assemblies through weak HMM learning in a WTA circuit through STDP. A: Illustration of the input encoding for sequence *AB-delay-ab*. The upper plot shows one example input spike train (blue dots) plotted on top of the mean firing rate (100 out of 200 afferent neurons shown). The lower panel shows the finite state grammar graph that represents the simple working memory task. The graph can be used to generate symbol sequences by following any path from *Start* to *Exit*. In the first state (*Start*) a random decision is made, which of the four paths to take. This decision determines all arcs that are passed throughout the sequence. On each arc that is passed the symbol next to the arc is emitted (and provided as input to the WTA circuit in the form of some 200-dimensional rate pattern). **B,C:** Evoked activity of the WTA circuit for one example input sequence before learning (B) and for each of the four sequences after learning (C). The network activity is averaged and smoothed over 100 trial runs (gray traces), the blue dots show the spiking activity for one trial run. The input sequences are labeled by their pattern symbols on top of each plot. The neurons are sorted by the time of their highest average activity over all four sequences, after learning. For each sequence a different assembly of neurons becomes active in the WTA circuit. Dotted black lines indicate the boundaries between assemblies. Since the 4 assemblies that emerged have virtually no overlap, the WTA circuit has recovered the structure of the hidden states that underlie the task. **D:** The lateral weights v_{kj} that emerged through STDP. The neurons are sorted →

→ using the same sorting algorithm as in (B,C). The black dotted lines correspond to assembly boundaries, neurons that fired on average less than one spike per sequence are not shown. Each neuron has learned to fire after a distinct set of predecessors, which reflects the sequential order of assembly firing. The stochastic switches between sequences are represented by enhanced weights between neurons active at the sequence onsets.

the finite state grammar in Fig. 2.2A. The arcs of the grammar directly correspond to the hidden states, i.e. given knowledge about the currently visited arc allows us to complete the sequence. The symbols next to the arcs define the observation model, i.e. the most likely symbol throughout each state. In this simple symbolic HMM the observation model is in fact deterministic, since exactly one symbol is allowed in each state.

In the neural implementation of this task, the symbolic sequences are presented to the WTA circuit encoded by afferent spike trains. Every symbol *A,B,C,D,a,b,c,d,delay* is represented by a rate pattern with fixed length of 50ms, during which each afferent neuron emits spikes with a symbol-specific, fixed Poisson rate (see Appendix B). One example input spike train encoding the symbolic sequence *AB-delay-ab* is shown in the top panel of Fig. 2.2A. The input spike times are not kept fixed but newly drawn for each pattern presentation. This input encoding adds extra variability to the task, which is not directly reflected by the simple symbolic finite state grammar. Still, the statistics underlying the input sequences $\boldsymbol{X}$ follow the dynamics of a HMM of the form (2.4), and therefore our WTA circuit and the spike trains that encode sequences generated by the artificial grammar share a common underlying model.

The observation model $p\left(\boldsymbol{x}_m \mid s_m, \boldsymbol{\theta}\right)$ of that HMM covers the uncertainty induced by the noisy rate patterns by assigning a certain likelihood to each observed input activation $\boldsymbol{x}_m$. The hidden state representation has to encode the context-dependent symbol identity and the temporal structure of the sequences, i.e. the duration of each individual symbol. In our continuous-time formulation the hidden state is updated at the time points $\hat{t}_1 \ldots \hat{t}_M$. Therefore, throughout the presentation of a rate pattern of 50ms length, several state updates are encountered during which the hidden state has to be maintained. In principle this can be done by allowing each hidden state to persist over multiple update steps by assigning non-zero probabilities to $p\left(s_m = k \mid s_{m-1} = k, \theta\right)$. However, this approach is well known to result in a poor representation of time as it induces an exponential distribution over the state durations, which is inappropriate in most physical systems and obviously also for the case of deterministic pattern lengths, considered here (Rabiner, 1989; Bishop, 2006). The accuracy of the model can be increased at the cost of a larger state space by introducing intermediate states, e.g. by representing pattern B in sequence *AB-delay-ab* by an assembly of states $s_{B,AB,1}, s_{B,AB,2}, \ldots$ that form an ordered state sequence throughout the pattern presentation. Each of these assemblies encodes a specific input pattern, the temporal context and its sequential structure throughout the pattern, and with sufficiently large assemblies the temporal resolution of the model achieves reasonable accuracy. We found that this coding strategy emerges

unsupervised in our WTA circuits through the STDP rule (2.7).

To show this, we trained a WTA circuit with $N = 200$ afferent cells and $K = 100$ circuit neurons by randomly presenting input spike sequences until convergence. In this experiment, the patterns were presented as a continuous stream of input spikes, without intermediate pauses or resetting the network activity at the beginning of the sequences. Training started from random initial weights, and therefore the observation and prediction model had to be learned from the presented spike sequences. Prior to learning the neural activity was unspecific to the patterns and their temporal context (see Fig. 2.2B). Fig. 2.2C shows the evoked activities for all four sequences after training. The output of the network is represented by the perievent time histogram (PETH) averaged over 100 trial runs and a single spike train that is plotted on top. To simplify the interpretation of the network output we sorted the neurons according to their preferred firing times (see Appendix B). Each sequence is encoded by a different assembly of neurons. This reflects the structure of the hidden state that underlies the task. Since the input is presented as continuous spike train, the network has also learned intermediate states that represent a gradual blending between patterns. About 25 neurons were used to encode the information required to represent the hidden state of each sequence.

This coding scheme installs different representations of the patterns depending on the temporal context they appeared in, e.g. the pattern *delay* within the sequence *AB-delay-ab* was represented by another assembly of neurons than the one in the sequence *BA-delay-ba*. Small assemblies of about five neurons became tuned for each pattern and temporal context. This sparse representation emerged through learning and is not merely a consequence of the inherent sparseness of the WTA dynamics. Prior to learning all WTA neurons are broadly tuned and show firing patterns that are unordered and nonspecific (see Fig. 2.2B). After learning their afferent synapses are tuned for specific input patterns, whereas the temporal contexts in which they appear are encoded in the excitatory lateral synapses. The latter can be seen by inspecting the synaptic weights v_{kj} shown in Fig. 2.2D. They reflect the sparse code and also the sequential order in which the neurons are activated. They also learned to encode the stochastic transitions at the beginning of the cue phase, where randomly one of the four sequences is selected. These stochastic switches are reflected in increased strength of synapses that connect neurons activated at the end and the beginning of the sequences.

The behavior of the circuit is further examined in Fig. 2.3. The average network activity over 100 trial runs of the neurons that became most active during sequence *AB-delay-ab* are shown in Fig. 2.3A. In addition the spike trains for 20 trials are shown for three example neurons. The same sorting was applied as in Fig. 2.2. Using the hidden state encoded by the network it should be possible to predict the recall patterns after seeing the cue, if it correctly learned the input statistics. We demonstrate this by presenting incomplete inputs to the network. After presentation of the delay pattern the input was turned off and the network was allowed to run freely. The delay pattern was played three times longer than in the training phase (150ms). During this time the network was required to store its current state (the

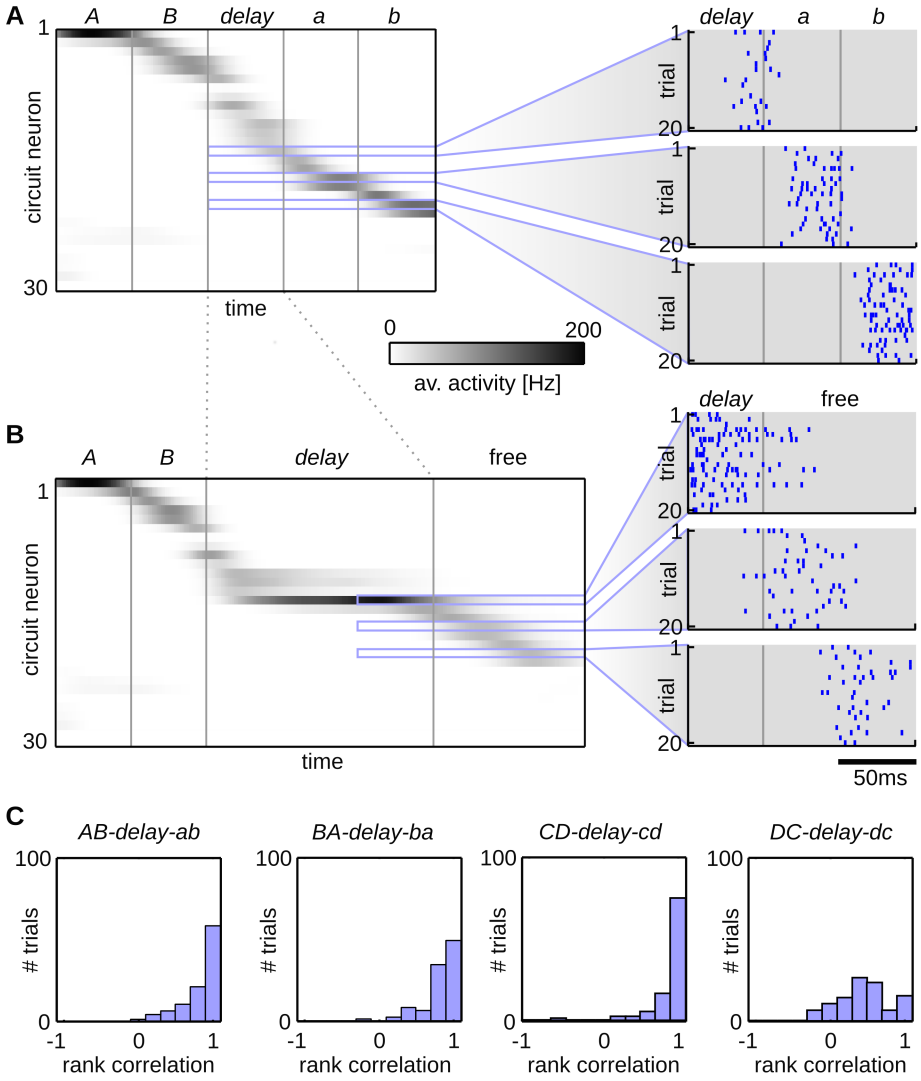

Fig. 2.3: Spontaneous replay of pattern sequences. A,B: The output behavior of a trained network for sequence *AB-delay-ab*. The network input is indicated by pattern symbols on top of the plot and pattern borders (gray vertical lines). **A:** The average firing behavior of the network during evoked activity. The 30 circuit neurons that showed highest activity for this sequence are shown. The remaining neurons were almost perfectly silent. The network activity is averaged over 100 trial runs and neurons are sorted by the time of maximum average activity. Detailed spiking activities for three example neurons that became active after the delay pattern are shown. Each plot shows 20 example spike trains. **B:** Spontaneous completion of sequence *AB-delay-free*. After presenting the cue sequence *AB* and the delay pattern for 150ms the afferent input was turned off, letting the network run driven solely by lateral connections. During this spontaneous activity, the neurons are activated in the same sequential order as in the evoked trials. Detailed spiking activity is shown for the same three example neurons as in (A). **C:** Histograms of the rank order correlation between the evoked and spontaneous network activity for all four sequences, computed over 100 trial runs. The sequential order of neural firing is reliably reproduced during the spontaneous activity and thus the structure of the hidden state is correctly completed.

identity of the cue sequence). After this delay time the input was turned off – no spikes were generated by the afferent neurons during this phase, the network was purely driven by the lateral connections. Since the delay time was much longer than the EPSP windows the network had to keep track of the sequence identity in its activity pattern throughout this time to solve the task. Fig. 2.3B shows the output behavior of the network for sequence *AB-delay-free* (where *free* denotes a 100ms time window with no external input). After the initial sequence *AB* was presented, a small assembly of neurons became active that represents the delay pattern that was associated with that specific sequence. After the delay pattern was turned off, the network completed the hidden state sequence using its memorized activity, which can be seen by comparing the evoked and spontaneous spike trains in Fig. 2.3A and B, respectively.

In order to quantify the ability of the network to reproduce the structure of the hidden state, we evaluated the similarity between the spontaneous and evoked network activity using the rank order correlation coefficient, which is a similarity measure normalized between -1 and 1, where 1 means that the order is perfectly preserved. This measure has been previously proposed to detect stereotypical temporal order in neural firing patterns (Luczak et al., 2009). Fig. 2.3C shows the histograms over the correlation coefficients for all four sequences. The histograms were created by calculating the rank order correlation between the spontaneous sequences and the PETH of the evoked sequences. It can be seen that the temporal order of the evoked sequence was reliably reproduced during the free run. To that end, for each of the input sequences, a stable representation has been trained into the network, that is encoded in the lateral synapses. This structure emerged completely unsupervised using the local STDP rule, solely from the intrinsic dynamics of the network.

Mixed selectivity emerges in multiple interconnected WTA circuits

The first experiment demonstrated that through STDP, single neurons of a WTA circuit get tuned for distinct input patterns and the temporal context in which they appear. The neural code that emerged is reminiscent of some features found in cortical activity of monkeys solving similar tasks, namely the emergence of context cells that respond specifically to certain symbols when they appear in a specific temporal context (Barone and Joseph, 1989; Shima and Tanji, 2000; Shima et al., 2006). However, the overall competition of a single WTA circuit hinders the building of codes for more abstract features, which are also found in the cortex in the very same experiments where neurons in the same cortical area encode different functional aspects of stimuli and actions. They seem to integrate information on different levels of abstraction which results in a diverse and rich neural code, where close-by neurons are often tuned to different task-related features (Rigotti et al., 2013).

We show that our model reproduces this mixed selectivity of cortical neurons

if multiple interconnected WTAs are trained on a common input. The strong competition is restricted to neurons within every single WTA, whereas there is no competition between neurons of different circuits and lateral connections allow full information exchange between the circuits. Therefore, the model is extended by splitting the network into smaller WTA groups, each of which receives input from a distinct inhibitory feedback loop that implements competition between members of that group. In addition all neurons receive lateral excitatory input from the whole network. Every WTA group still follows the dynamics of a forward sampler for a HMM. Each of these WTA circuits adapts its synaptic weights through STDP to best represent the observed input spike sequences. In addition, the lateral connections between WTA groups introduce a coupling between the network states of individual groups. The dynamics of the whole network of WTA circuits can be understood as a forward sampler for a coupled HMM (Brand, 1997), where every WTA group encodes one multinomial variable of a compound state such that from one time step to the next all single state variables have influence on each other (Brand, 1997; Koller and Friedman, 2009).

In the first experiment we have seen that the WTA circuit learned to use about 25% of the available neurons to encode each of the four sequences. We have also seen that the network used small assemblies of neurons to represent each of the patterns in favor of a finer temporal resolution. This implies that WTA circuits of different size can learn to decode the input sequence on different levels of detail, where small circuits only learn the most salient features of the input sequences. To show this we trained a network with 10 WTA groups of random size between 10 and 50 units, giving a total network size of $K = 318$, on the simple object sequence memory task (Fig. 2.2A). The neural code that emerges in this network after training is shown in Fig. 2.4. The output rates of the circuit neurons were measured during the presentation of pattern *a* appearing in the sequence *AB-delay-ab*, *BA-delay-ba*, shown in Fig. 2.4A,B respectively. Three classes of neurons can be distinguished: 10 neurons were tuned to pattern *a* in the context *AB-delay-ab* only (shown in red), 12 neurons were tuned to pattern *a* exclusievly in the context *BA-delay-ba* (shown in blue) and 5 additional neurons encode pattern *a* independent of its context (green), i.e. they get activated by the pattern *a* in both sequences *AB-delay-ab* and *BA-delay-ba*. The remaining neurons were not significantly tuned for pattern *a* (average firing rate during pattern *a* was less than 10Hz, not shown in the plot).

To pinpoint the computational function that emerged in the network we compared the spontaneous activity of individual neurons from different WTA circuits. Spike trains for one context-specific and one non-specific neuron are compared in Fig. 2.4C and D, respectively. Both panels show spike raster plots over 20 trial runs and averaged neuron activities (PETH) for sequences *AB-delay-free* and *BA-delay-free*. The neuron in Fig. 2.4C belongs to a small WTA group with a total size of 15 neurons and shows context unspecific behavior, whereas the neuron in Fig. 2.4D which belongs to a larger WTA group (42 neurons) is context specific (see Fig. 2.4A,B). This behavior is also reproduced during the free run, when the neurons are only driven by their lateral synapses. The neuron in Fig. 2.4D remains silent during

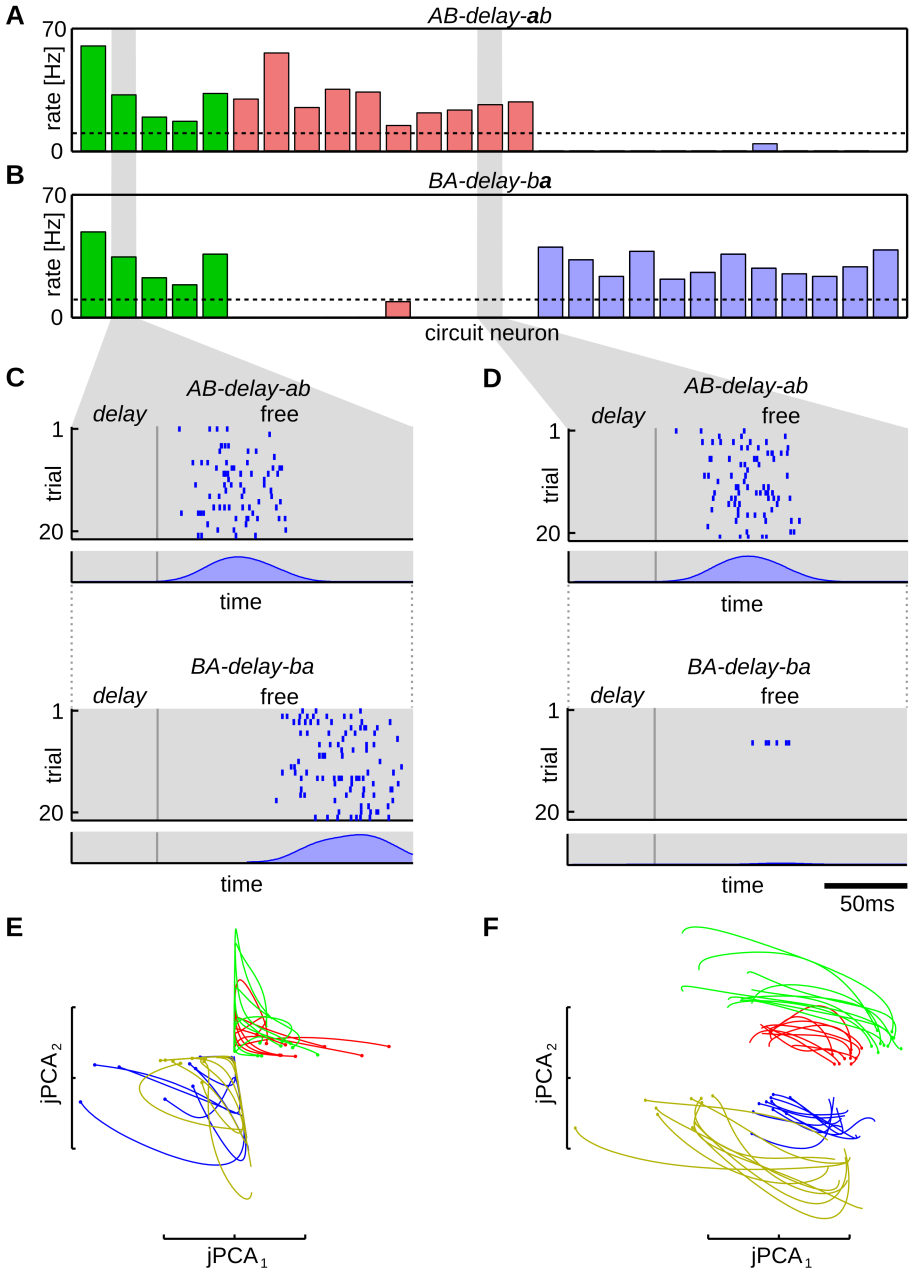

Fig. 2.4: Mixed selectivity in networks of multiple interconnected WTA circuits. A,B: Mean firing rate of the circuit neurons for evoked activity during pattern *a* in sequence *AB-delay-ab* (A) and *BA-delay-ba* (B). A threshold of 10Hz (dashed line) was used to distinguish between neurons that were active or inactive during the pattern. Firing rates of neurons that were not context selective are shown in green, that of neurons selective for starting sequences *AB* and *BA* are shown in red and blue, respectively. Neurons that did not fall in one of these groups are not shown. **C,D:** Spike trains of one context selective (D) and one non-selective (C) neuron are presented for spontaneous completion of sequence *AB-delay-ab* (upper) and *BA-delay-ba* (lower) (cue phase is not shown). Spike raster plots over 20 trial runs and corresponding averaged neural activity (PETH) are shown. The two neurons encode the input on different levels of abstraction. The neuron in panel (D) shows context cell →

→ behavior, since it encodes pattern *a* only if it occurs in the context of sequence *ab*. During *ba* it remains (almost) perfectly silent. The neuron in (C) is not context selective, but nevertheless fires reliably during the time slot of pattern *a* during the free run by integrating information from other (context selective) neurons. It belongs to a WTA circuit with 15 neurons, for which the network state projection is shown in panel (E). **E,F:** Linear projection of the network activity during the delay phase to the first two components of the jPCA, for a single WTA circuit with 15 neurons (E) and for the whole network (F). 10 trajectories are plotted for each sequence (*AB-delay-ab* red, *BA-delay-ba* green, *CD-delay-cd* blue, *DC-delay-dc* yellow). The dots at the beginning of each line, indicate the onsets of the delay state, i.e. the beginning of the trajectories. The plots have arbitrary scale. The projection of the WTA circuit in (E) does not allow a linear separation between all four sequences, whereas the activity of the whole network (F) clusters into four sequence-specific regions. The network neurons use this state representation to modulate their behavior during spontaneous activity.

BA-delay-free and thus shows the properties of context cells observed in the cortex, whereas the neuron in Fig. 2.4C is active during both sequences. Still, during spontaneous replay that neuron correctly reproduces the temporal structure of the input sequences. In sequences starting with *AB* the neural activity peaks at 50ms after the onset of the free run – the time pattern *a* was presented in the evoked phase. If the sequence starts with *BA* this behavior is modulated and the activity is delayed by roughly 50ms, to the time point *a* would appear in the recall phase. The required information to control this modulation was not available within the small WTA group the neuron belongs to, but provided by neighboring context-specific neurons from other groups.

To see this we trained a linear classifier on the evoked activity during the delay phase of *AB-delay-ab* and *BA-delay-ba* (see Appendix B for details). If the neurons reliably encode the sequence identity a separating plane should divide the K-dimensional space of network activities between the sequences. Training the classifier only on the 15-dimensional state space of the group the neuron in Fig. 2.4C belongs to, did not reveal such a plane (the classification performance was 72.5%). Therefore, this small WTA circuit did not encode the required memory item to distinguish between the two sequences after the delay phase. However, the whole network of all WTA groups reliably encoded this information and the classifier trained on the K-dimensional state space could distinguish between the delay phases of *AB-delay-ab* and *BA-delay-ba* with 100% accuracy.

To illustrate the different emergent representations, we compared linear projections of the state of the small WTA group with 15 neurons and the state of the whole network in Fig. 2.4E,F, respectively. The plots show the network activity during the delay phase for all four sequences. Each line corresponds to a trajectory of the evoked network activity, where the line colors indicate the sequence identity. The state trajectories were projected onto the first two dimensions of the dynamic principal component analysis (jPCA), that was recently introduced as an alternative to normal PCA that is applicable to data with rotational dynamics (Churchland et al., 2012). Empirically, we found this analysis method superior to normal PCA in finding linear projections that separate the network states for different input sequences. One explanation for this lies in the dynamical properties of WTA

circuits. Due to the global normalization which induces a constant network rate, the dynamics of the network are roughly energy-preserving. Since this implies that the corresponding linear dynamical system is largely non-expanding/contracting, a method that identifies purely rotational dynamics such as the jPCA was found to be beneficial here.

Fig. 2.4E shows the first two jPCA components of the neural activities during the delay phase for the WTA circuit with 15 neurons, which the neuron in Fig. 2.4C belongs to. This circuit was not able to distinguish between all four input sequences, since it activated the same neurons to encode them. This is also reflected in the jPCA projections shown in Fig. 2.4E, which show a large overlap for sequences *AB-delay-ab* and *BA-delay-ba*. On the other hand, the network state comprising all K neurons reliably encoded the sequence identities (see Fig. 2.4F). The delay state for each sequence spans an area in the 2-D projection and therefore the network found a state space that allows a linear separation between the sequences. Such a representation is important since the neuron model employs a linear combination of the network state in the membrane dynamics (2.1) and therefore provides the information required by the neurons in Fig. 2.4C,D to modulate their spontaneous behavior.

Trajectories in network assemblies emerge for stationary input patterns

Information about transient stimuli is often kept available over long time spans in trajectories of neural activity in the mammalian cortex (Han et al., 2008; Luczak et al., 2007; W. Xu et al., 2007; Jin et al., 2009) and in songbirds (Fiete et al., 2010; Kozhevnikov and Fee, 2007; Hahnloser et al., 2002). In the previous experiment we saw that our model is in principle capable to develop such trajectories in neural assemblies (see Fig.2.3B), which emerged to encode salient input patterns and the temporal structure throughout them. However, in that experiment the input sequences comprised a rich temporal structure, since each pattern was only shown for a 50ms time bin which might have facilitated the development of these activity patterns. In this section we study whether a similar behavior also emerges when the input signal is stationary over long time spans.

In analogy to the previous experiment we generated two input sequences *A-delay* and *B-delay*. The patterns A, B were played for 100ms and the pattern *delay* for 500ms. As in all other experiments, the patterns were rate patterns, i.e. each input neuron fired with a constant Poisson rate during the pattern and spike times were not kept fixed throughout trials. One example input spike train is shown in Fig. 2.5A.

Although the input was stationary for 500ms during the *delay* pattern, we could still observe the emergence of neural trajectories in the network after training. Again, we used a network composed of multiple interconnected WTA circuits to learn these patterns. We employed a network of 20 WTA groups of random size in the range from 10 to 100 neurons. The total network had a size of $K = 704$

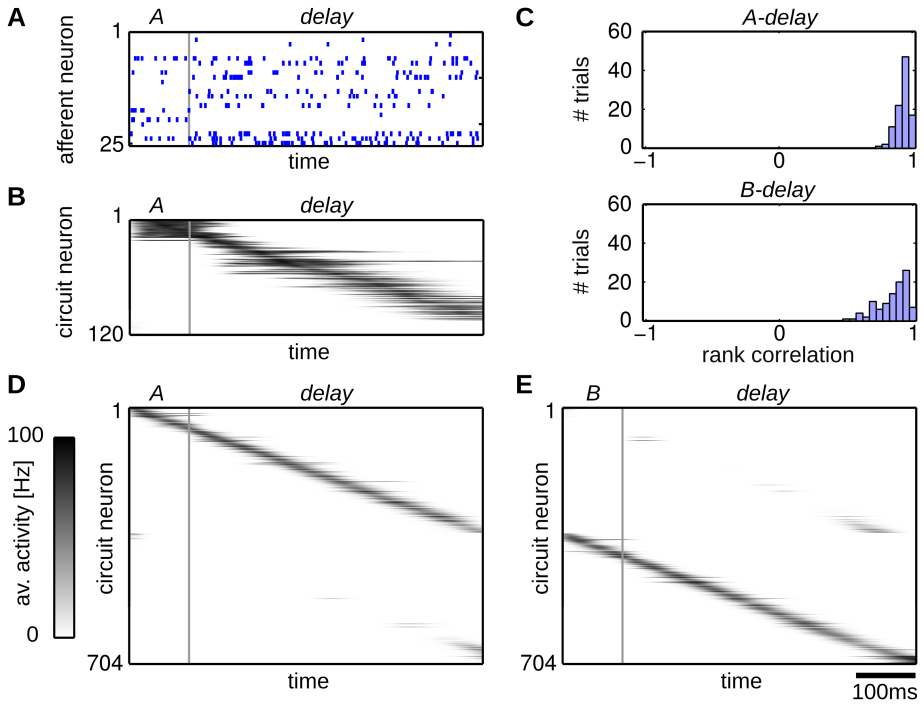

Fig. 2.5: Neural trajectories emerge for stationary input patterns. A: A network was trained with an extended delay phase of 500ms. Input spike trains of a single run for sequence *A-delay* (25 out of 100 afferent neurons). Throughout the delay phase the afferent neurons fire with fixed stationary Poisson rates. **B:** The output behavior for sequence *A-delay* averaged over 100 trial runs. The circuit neurons are sorted according to their mean firing time within the sequences (120 out of 704 neurons are shown). **C:** Histograms of the rank order correlation between the evoked and spontaneous network activity. The sequential order of neural firing is preserved during spontaneous activity. **D,E:** Homeostatic plasticity enhances the formation of this sequential structure. The output behavior of the network trained with STDP and the homeostatic plasticity mechanism is shown. Approximately 50% of the neurons encode each of the two sequence. The neurons learn to fire at a specific point in time within the delay patterns, building up stable trajectories.

circuit neurons and we used $N = 100$ afferent cells. Fig. 2.5B shows the sorted average output activity after training. For each of the two sequences a distinct assembly of neurons emerged and the neurons composing these assemblies fired in a distinct sequential order. Fig. 2.5C shows the rank order correlations between the evoked and spontaneous activities. The trajectories of neural firing were reliably reproduced during spontaneous activity, but only about 100 neurons were used for each of the two assemblies, leaving the remaining 500 neurons (almost) perfectly silent.

The emergence of these trajectories can be further enhanced using a homeostatic intrinsic plasticity mechanism which enforces that on average all network neurons participate equally in the representation of the hidden state. This can be achieved by a mechanism that regulates the excitability b_k of each neuron, such that the

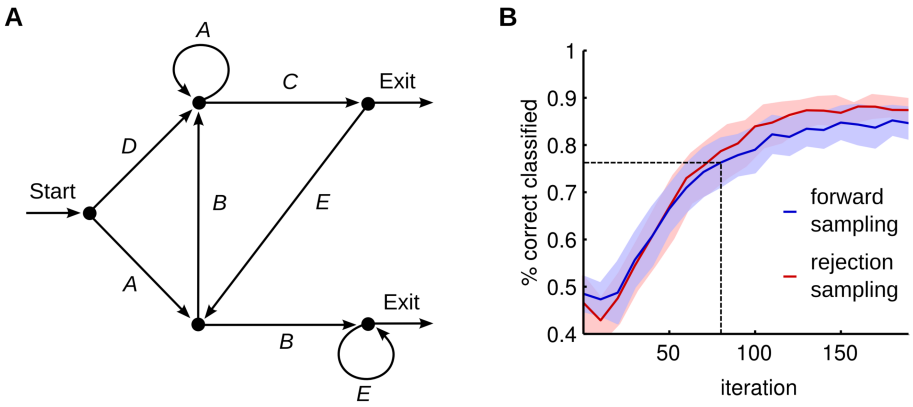

Fig. 2.6: Fast learning of an artificial grammar. A: The artificial grammar from (Conway and Christiansen, 2005; Gomez and Gerken, 1999) represented as a finite state grammar graph. Grammatical sequences are generated by following a path from *Start* to *Exit*. If a node has more than one outgoing arc one is chosen at random with equal probability to continue the path. **B:** Convergence of the network performance on that task. The blue curve shows the evolution of the mean classification performance against the number of training samples, when forward sampling was used. The blue shaded area indicates the standard deviation over 20 trial runs. After 80 training samples the network exceeds human performance reported in (Conway and Christiansen, 2005). Using rejection sampling with 10 samples on average (red curve) does not significantly outperform forward sampling on this task.

overall output rate ν_k of neuron k (measured over a long time window) converges to a given target rate $\nu_k = \frac{1}{K}\hat{\nu}$. (see (Habenschuss et al., 2012) and Appendix B). Augmenting the dynamics of the network with this intrinsic plasticity rule prevents neurons from becoming inactive if their synaptic weights decrease and by that assures that each neuron joins one of the assemblies. This can be seen in Fig. 2.5C,D which shows the output activity after training with STDP augmented with the homeostatic mechanism. The neurons formed a fixed ordered sequence and thus showed a clear preference for a certain point in time within the pattern. Even though the delay pattern had no salient temporal structure (the rates of all afferent neurons were constant throughout the pattern) these trajectories were formed by imprinting the sequential order of the neural activity into the lateral excitatory connections. As in the first experiment each neuron has learned to fire after a distinct group of preferred predecessors, resulting in neural trajectories through the network. Therefore, the time that has elapsed since the *delay* pattern started could be inferred from the neural population activity. In addition the identity of the initial pattern was also memorized, since about half of the population became active for each of the two sequences.

Learning the temporal structure of an artificial grammar model

The finite state grammar used in the previous experiments (Fig. 2.2A) did not utilize the full expressive power of HMMs since it only allowed stochastic switches at the beginning of each sequence. In this section we consider the problem of learning more general finite state grammars in WTA circuits, a problem that has also been extensively studied in cognitive science in artificial grammar learning (AGL) experiments (Reber, 1967). Fig. 2.6A shows the artificial grammar that was used in (Conway and Christiansen, 2005) to train subjects using different stimulus modalities (visual, auditory and tactile). There it was shown that humans can acquire the basic statistics of such grammars extremely fast. On this particular task humans showed a performance of 62% to 75% percent (depending on the stimulus modality that was used) after only a few dozens of stimulus presentations (Conway and Christiansen, 2005).

We show that our network model can extract the basic structure of this grammar. This internal representation can be subsequently used to classify unseen sequences as grammatical or not. Through STDP the network adapts the parameters $\boldsymbol{\theta}$ such that they reflect the statistics underlying the training sequences, and the emergent HMM can then be used to evaluate the sequence likelihood $p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right)$. The ability of the network to distinguish between grammatical and ungrammatical sequences was assessed by applying a threshold on the sequence log-likelihood, an approximation of which was computed over a single sample $\boldsymbol{S}$ from (2.5) (see Appendix B). The threshold was assigned to the mean of the log-likelihood values computed for all test sequences. Likelihoods that laid above that threshold were reported as grammatical.

In this experiment we used a sparse input coding, where only a small subset of afferent neurons is activated for each of the symbols. This representation could be realized by another WTA circuit used as input for the network to decode more complex input patterns. We trained a single WTA circuit with $K = 10$ neurons on this sparse input. Using this model, we were able to achieve high learning speeds. In each training iteration one of the 12 training data sets from (Conway and Christiansen, 2005) (using only the first sequence of each match/mismatch pair) was chosen at random and presented to the network. For testing we used the 20 test sequences from (Conway and Christiansen, 2005) to evaluate the learning performance. Training was interrupted after every 10^{th} sequence presentation to assess the classification performance. The resulting learning curve is shown in Fig. 2.6B. The classification rate of 75% that was reported in the behavioral experiment was exceeded after only 80 iterations. By training the network beyond this point performances up to 85% were reached. Note that none of the training sequences appeared in the test set. Therefore the network has not just learned a fixed set of sequences, but extracted relevant statistical features that allowed it to generalize to new data.

2.4 A refined EM approximation using rejection sampling

So far in all experiments the simple forward sampling approximation was used for learning the model parameters. Although this learning paradigm has shown to be surprisingly powerful, it is limited and will not be sufficient if the network is required to learn more complex tasks or acquire probabilistic models with a high level of detail. In this section we derive the refined approximation toward evaluating the HMM E-Step in a recurrent WTA circuit based on rejection sampling.

Exactly solving the E-step requires to evaluate the posterior probability of $\boldsymbol{S}$, given by

$$p\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right) = \frac{p\left(\boldsymbol{S}, \boldsymbol{X} \mid \boldsymbol{\theta}\right)}{p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right)} \quad \text{with} \quad p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right) = \int p\left(\boldsymbol{S}', \boldsymbol{X} \mid \boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{S}', \tag{2.8}$$

where $p\left(\boldsymbol{S}, \boldsymbol{X} \mid \boldsymbol{\theta}\right)$ is the HMM joint distribution, given by equation (2.4). A stochastic EM update is realized by drawing a state sequence from the posterior for which the M-step parameter updates are performed. However, directly sampling from (2.8) is not possible for a spiking neural network, since it requires the integration of information over the whole state sequence and thus, looking into the future. This can be seen by noting that the integral in (2.8) runs over the state space of the whole sequence. To that end, the network is not able to sample from this distribution directly. Nevertheless, it is possible to indirectly evaluate (2.8) using samples generated from (2.5), which can be expressed by

$$p\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right) = q\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right) \cdot \frac{r(\boldsymbol{S})}{\left\langle r(\boldsymbol{S}')\right\rangle_{q(\boldsymbol{S}' \mid \boldsymbol{X}, \boldsymbol{\theta})}}, \tag{2.9}$$

where $\left\langle \cdot \right\rangle_{q(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta})}$ denotes the expected value over $q\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right)$, which in this context is called a *proposal distribution* since it is used to propose samples, which are then used to indirectly evaluate the target distribution $p\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right)$. The scalar $r\left(\boldsymbol{S}\right)$ is the *importance weight* between the target and the proposal distribution, which is used to scale the influence of the sample $\boldsymbol{S}$ (Bishop, 2006; Koller and Friedman, 2009; Neal, 1993).

The expectation in the denominator of (2.9) is again not easy to evaluate, since it requires us to integrate over multiple sequences. The most pragmatic solution to this problem is to approximate this term using a single sample from the proposal distribution $\left\langle r(\boldsymbol{S}')\right\rangle_{q(\boldsymbol{S}' \mid \boldsymbol{X}, \boldsymbol{\theta})} \approx r(\boldsymbol{S})$. Under this approximation the importance weight in (2.9) cancels out and we arrive at the trivial approximation $p\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right) \approx q\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right)$, i.e. each sample from the proposal distribution is accepted as a valid sample from the posterior. This is the forward sampling approximation that was used so far throughout all experiments.

In order to improve this approximation we use the stochasticity of the network, which assures that different state sequences $\boldsymbol{S}$ are proposed if the same input sequence $\boldsymbol{X}$ is presented several times. Rejection sampling utilizes this stochasticity

and preferentially selects sequences with high likelihood throughout the whole input. The required information to do this selection is a global quantity that must be tracked over the whole sequence. The probability to accept a state sequence $\boldsymbol{S}$ is directly proportional to the importance weight $r(\boldsymbol{S})$, which computes to

$$r\left(\boldsymbol{S}\right) = \prod_{m=1}^{M} p\left(\boldsymbol{x}_m \mid s_{m-1}, \boldsymbol{\theta}\right). \tag{2.10}$$

Note that (2.10) can be easily computed forward in time, since in each time step, it only needs to be updated using the instantaneous input likelihood $p\left(\boldsymbol{x}_m \mid s_{m-1}, \boldsymbol{\theta}\right)$. Further note that this is a measure for surprise or prediction error – the probability of observing the current input given the previous state. The information to decide whether to accept $\boldsymbol{S}$ is the accumulated prediction error over the whole sequence. This approach also naturally extends to the case of multiple interconnected WTAs. There, the contributions to the importance weight of every single circuit have to be multiplied in every time step and therefore, a possible rejection is in that case effective for the whole network of all WTAs at once.

Since the importance weights need to be accumulated over the whole sequence of spike events of length M, the weight update rules (2.7) can not be applied instantaneously. In the neural implementation we achieved this using a synaptic eligibility trace as proposed in (Izhikevich, 2007). Instead of updating the weights directly they are tagged and consolidation of the tags is delayed until the whole sequence is read. The probability to accept these tags is proportional to the importance weights, i.e.

$$p\left(\text{accept sequence } \boldsymbol{S}\right) = c \cdot r\left(\boldsymbol{S}\right), \tag{2.11}$$

where c is a constant that scales the acceptance rate. If a sequence $\boldsymbol{S}$ is accepted, the synaptic tags are consolidated. If the circuit decides not to accept, the synaptic weight changes for the whole sequence have to be discarded. This result is analytically similar to (Brea et al., 2011), where the importance weights (2.10) were introduced by weighting the eligibility traces with a deterministic scalar factor (importance sampling). Here, in the rejection sampling framework a stochastic variant of this method is used. The advantage of the rejection sampling method is that it is not necessary to explicitly compute the normalization in (2.8). The normalization can be approximated by replaying in every training iteration the input sequence multiple times until it gets accepted once, instead of using a constant number of replays as with importance sampling. In practice however it is necessary to adapt the parameter c throughout learning in order to get a reasonable number of replays. We used a simple linear tracking mechanism for c throughout the experiments (see Appendix B). A performance comparison of these different sampling approximations is provided at the end of the Results section.

We assume that the circuit interacts with a mechanism that allows the replay of the afferent stimulus multiple times. By enforcing that each input is accepted once, we guarantee that the network learns the statistics of all input sequences with equal accuracy. This view allows us to make an interesting theoretical prediction: when

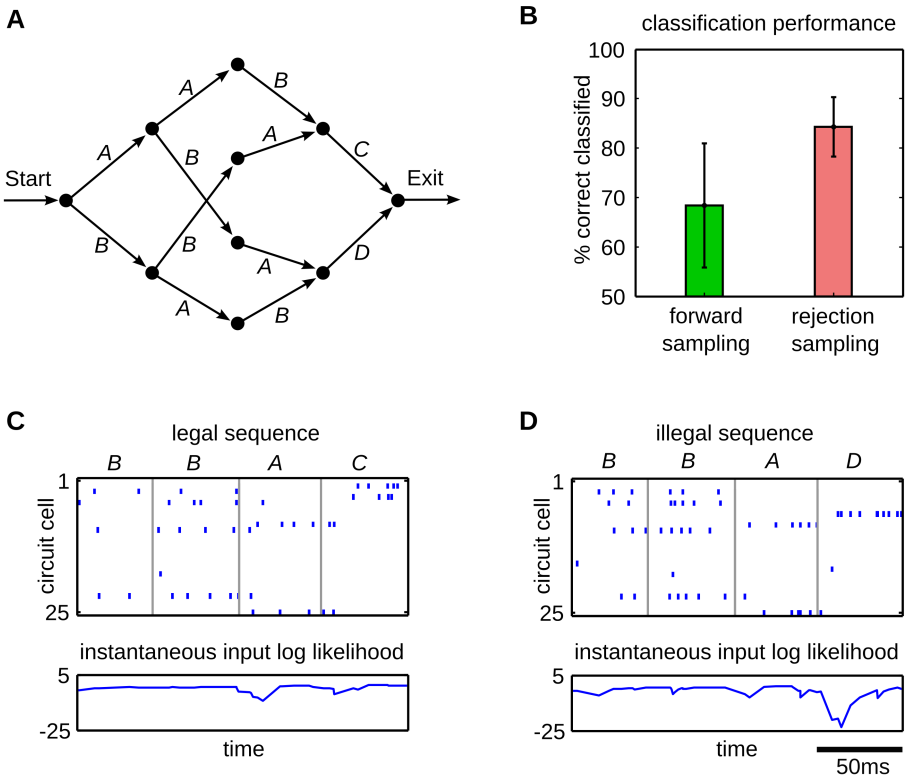

Fig. 2.7: Rejection sampling enhances the classification performance of the network. A: The grammar graph used for this task. A three letter sequence composed of *A*s and *B*s identifies the last symbol, *C* or *D*. Therefore, the most salient information is provided at the end of the sequence. **B:** The classification rate on this task is plotted for forward (green) and rejection sampling (red). The error bars indicate the standard deviation over 10 trial runs. Rejection sampling significantly increases the classification performance on this task. **C,D:** Comparison of the time courses of the instantaneous input log likelihood for a legal input sequence *BBAC* (C) and an illegal sequence *BBAD* (D). Input patterns are indicated by the pattern symbols on top of the plots. The upper plot shows the output spike trains of the network, the lower plot shows the traces of the instantaneous input likelihood plotted in the log domain, which indicates the ability of the network to predict the continuation of the afferent spike train. The trace in (D) shows a strong negative peak at the illegal transition at 150ms. The prediction model that emerged through STDP augmented with rejection sampling, enables the network to detect illegal sequences.

an input is not well represented by the network it is more likely to be rejected and therefore, the number of rejected and resampled sequences represents a notion of novelty. Literally speaking, the network pays more attention to novel inputs, by resampling them multiple times (see Appendix B for details).

Rejection sampling enhances the learning capabilities of STDP

In the following experiments we investigate the possible performance gain that can be achieved if the network has access to this rejection sampling mechanism. We have previously seen that the grammar from Fig. 1 in (Conway and Christiansen, 2005) can be learned almost perfectly using pure forward sampling. However, this data set had a very simple structure. To distinguish between grammatical and ungrammatical sequences only required the analysis of the local statistics of the input. E.g. it is easy to see that the sequence *DEAC* is not grammatical since it contains the bigram *DE*, which never appears in the training data. Each of the ungrammatical sequences contains at least one illegal bigram and thus can be classified based on a simple model of symbol transitions. This simple structure was already recovered with the online learning scheme and therefore using rejection sampling on that task did not result in a significant performance increase (see Fig. 2.6).

To demonstrate the advantage of rejection sampling, we created a grammar that required integration of information over a longer time span, shown in Fig. 2.7A. Although this grammar only allows to create four sequences *AABC*, *BBAC*, *ABAD* and *BABD*, the underlying structure is more complex than in the previous tasks. The identity of the last symbol can only be inferred if the identity and context of the first symbol is integrated and memorized over the whole sequence. To that end, the rejection sampling algorithm that allows the network to propagate information over the whole sequence, should bring a definite benefit over forward sampling for this task.

The quantity that is needed to update the importance weights (2.10) and also to estimate the sequence likelihood for classifying grammatical against ungrammatical inputs, is given by the instantaneous input likelihood $p\left(\boldsymbol{x}_m \mid s_{m-1}, \boldsymbol{\theta}\right)$ (see Appendix B). As pointed out earlier, this quantity is a measure for surprise, i.e. the probability of observing the current input pattern given the network state. The ability of the network to exploit this prediction error to classify sequences is illustrated in Fig. 2.7. The input-output behavior of a network after training with rejection sampling is shown for the grammatical sequence *BBAC* and the ungrammatical sequence *BBAD*, in Fig. 2.7C,D respectively. The bottom plots show traces of the instantaneous input log-likelihood. Throughout the grammatical sequence in Fig. 2.7C the trace stays near baseline, which indicates that the network is capable of predicting the sequence. Within the patterns, the trace only shows small deviations due to input noise. Switches between the input patterns e.g. at the border from pattern *A* to *C* cause modest levels of surprise, due to the sudden change of the network state. However, the illegal transition to pattern *D* in Fig. 2.7D causes a strong negative peak. At this point the network is not capable of predicting the final pattern. Thus the input is assigned to a low overall sequence likelihood and will therefore be classified as ungrammatical.

In the rejection sampling algorithm this quantity is also used throughout training, to learn preferably from sequences that are best capable of predicting the input

sequences. To quantify the advantage of this method over online learning we compared the performance on the AGL task. As in the previous experiment, the ability of the network to distinguish between grammatical and ungrammatical sequences was evaluated by applying a threshold on the sequence likelihood. The threshold was assigned to the mean of the log-likelihood values computed for all tested sequences. The network parameters were tuned such that the number of rejected samples in each iteration, averaged over the whole training session was equal to the desired number of samples (see Appendix B). The classification errors are compared in Fig. 2.7B for learning with forward and rejection sampling. The parameter c that scales the number of rejected samples was tracked to give an average number of 10 rejected samples per iteration. Despite this relatively small number of times the sequences is resampled, it can be seen that the performance on this task significantly increased with rejection sampling. Online learning achieved a classification rate of $68.38 \pm 12.52\%$. With rejection sampling the network achieved $84.30 \pm 5.98\%$ classification rate. Hence we confirmed, that having access to the rejection sampling mechanism allows the network to learn the input statistics with higher levels of accuracy. Furthermore, for the example given here, this was achieved with a relatively small average number of resampled state sequences.

Comparison of the convergence speed and performance of the approximate algorithms

In order to give a quantitative notion of how the sampling approximations affect the learning performance, we applied the methods to solve a generic HMM learning task. To allow a direct comparison with standard machine learning algorithms for HMMs, we used a time-discrete version of our model in this section. Therefore, we set the inter-spike-intervals Δ_m to a fixed constant value and used rectangular EPSP kernels of the same length. With this modification our model is equivalent to a discrete input, discrete state HMM, commonly considered in the machine learning literature (Bishop, 2006). We created random HMMs and used them to generate a training and a test data set. Using this data we compared the training performance of different approximation algorithms.

The accuracy of the rejection sampling algorithm crucially depends on how the parameter c in equation (2.11) is selected. If it is set to a very large constant value, every sample gets accepted and we arrive at the simple forward sampling approximation. We compared this forward sampling algorithm with the simple tracking algorithm that was used in the previous experiment and with the optimal mechanism, which computes c over a batch of sampled sequences (see Appendix B). In addition we compared these methods with the importance sampling algorithm considered in (Brea et al., 2011), where the scalar values of the importance weights were directly used to weight synaptic tags. All sampling methods were compared for an average number of 10 and 100 resampled sequences. Furthermore we applied standard EM learning for HMMs (the Baum-Welch algorithm) as reference method (Baum and Petrie, 1966; Bishop, 2006).

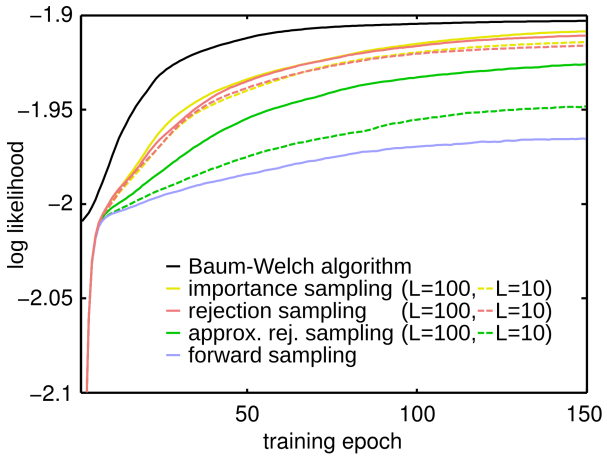

Fig. 2.8: Comparison of the convergence speed and learning performance of different sampling methods. Comparison of the sampling approximations to standard HMM learning. The performance is assessed by the log likelihood averaged over 50 trial runs. The plots show average convergence properties of: forward sampling (solid blue), importance sampling over 10 (dashed yellow) and 100 trials on average (solid yellow), rejection sampling over 10 (dashed red) and 100 trials (solid red), rejection sampling with the simple linear tracking of c over 10 (dashed green) and 100 trials on average (solid green), and the Baum-Welch algorithm (solid black). With increased number of samples the performance of the algorithm converges towards the solution of the standard EM algorithm. There was no significant performance difference between rejection and importance sampling. The simple tracking mechanism for the rejection sampler is outperformed by the exact algorithm, but still a significant performance gain with increased number of samples can be observed.

The results of the eight different training algorithms are compared in Fig. 2.8. The figure shows the log-likelihood on the test data averaged over the 50 learning trials. As can be seen, pure forward sampling shows poor performance on this task compared with Baum-Welch learning, but with increasing number of samples the approximation approaches the performance of the exact EM updates. Interestingly we found that importance sampling and rejection sampling show almost the same performance. We believe that the reason for this lies in the high variance of the importance weights. The weights of consecutive samples can differ several orders of magnitude. After normalization, effectively only the sample with the highest importance weight has non-zero influence on the weight updates. Therefore the two algorithms are numerically almost identical for the task considered here. Using the tracking mechanism for c resulted in decreased performance compared to the exact algorithm. Still, a significant performance gain can be observed with increased average number of samples.

2.5 Discussion

We have shown that STDP in WTA circuits with lateral excitatory connections implements the capability to represent the statistical structure underlying time-varying input patterns. The different types of excitatory synapses in the network serve different computational functions. Lateral connections recurrently feed back past network spikes which are used to predict a prior belief about the current network activity. The feedforward synapses match this prediction against the belief inferred from afferent inputs. The sparse code that emerges in this circuit allows to represent the activity of the whole network as samples from the state space of a HMM, implementing a forward sampler, which provides the circuit with a simple online approximation to exact HMM inference.

We have focused in this article on an idealized version of the STDP rule that implements the maximization step of the EM algorithm for the HMM. Similar rules also emerged in earlier studies as stochastic approximations to EM implemented in networks of spiking neurons (Nessler et al., 2010; Habenschuss et al., 2013; Nessler et al., 2013; Keck et al., 2012), for learning instantaneous afferent spike patterns. The only structural difference in the network architecture for temporal models is the presence of lateral excitatory connections. We have shown that if a WTA circuit is passively exposed to spatiotemporal rate patterns STDP implements a crude online approximation of EM. The emerging neural codes represent the hidden states that underlie the spatiotemporal input patterns. Different neurons are activated for the same input pattern if it appears in different temporal contexts. Furthermore, we have shown that if multiple WTA circuits are recurrently interconnected the network activity becomes more diverse and encodes various abstract features.

Throughout our analysis we realized the WTA dynamics using a feedback loop, where the required inhibition was given in its theoretically optimal form, according to equation (2.3). This optimal inhibition predicted by our model is strongly correlated with the activation of excitatory neurons within the WTA circuit. Such strong balance and correlation between excitation and inhibition has been observed in the cortex in vivo (Okun and Lampl, 2008; Haider et al., 2006). A consequence of this inhibitory feedback in our network model is that the total output rate is constant. Yet individual neurons in the network may exhibit complex behavior, and our experimental results have shown that they exploit a wide dynamical range. Furthermore it has been shown in (Nessler et al., 2013) that the assumption of a constant overall output rates can be lifted for the case of WTA circuits without lateral synapses. The only requirement identified there was that the circuit-wide output rate and the input had to be stochastically independent, which was theoretically shown and experimentally verified in an experiment where the network rate was modulated with a global oscillation throughout learning. The constant output rate considered in the present study is the simplest case which is compatible with our model. Identifying a more general class of rate functions which can be incorporated into our theoretical framework will be the subject of future work.

The activity patterns that emerge in our WTA circuits share important features with experimentally observed cortical dynamics. One feature is that neurons become tuned for mixtures of different task-relevant aspects as commonly observed in cortical neurons (Rigotti et al., 2013; Barone and Joseph, 1989; Shima and Tanji, 2000). The neural assemblies that encode these temporal features imprint their stereotypical sequential activation pattern within the lateral synapses. Another common feature is the emergence of stereotypical firing sequences during evoked and spontaneous activity, which is also found in cortical activity (Han et al., 2008; Luczak et al., 2007; W. Xu et al., 2007). This analysis also provides a theoretical foundation for the results that were reported recently in (Klampfl and Maass, 2013). There, a similar network of stochastic WTA circuits was used to learn spike patterns superimposed with Poisson noise, and similar stimulus-specific assemblies emerged. But no theoretical framework was provided there.

The network of multiple interconnected WTA circuits has a very interesting theoretical interpretation as it implements in that case a forward sampler for a coupled HMM (Brand, 1997), where multiple HMMs run in parallel to jointly encode the hidden state. In our experiment this coupling between neighboring WTA circuits allowed them to reproduce typical sequences of hidden states in the absence of input, even if some circuits did not have enough expressive power to store this information. An interesting future extension of this model would be to present different coupled stimuli (e.g. speech and audio from a common source) to different WTA groups in this circuit. Individual WTA circuits would then learn the temporal structure of these stimuli and the lateral excitatory synapses between WTA circuits would detect relevant correlations between them.

We have also shown that STDP installs in WTA circuits capabilities that go beyond just learning afferent sequences. From few presentations the network extracted relevant statistical properties underlying the afferent patterns. We demonstrated this on an artificial grammar learning task. The network extracted parts of the structure of this grammar, which allowed it to subsequently classify unseen sequences as stemming from the same grammar or not. Interestingly, the learning speed and classification performance achieved with the forward sampling approximation, in the early learning phase, is comparable to the performance reported for humans on the same task (Conway and Christiansen, 2005). This is also interesting because the network considered here is similar to the single recurrent network (SRN) previously suggested as a model for artificial grammar learning (Elman, 1990; Jordan, 1997). The context layer, that is used in the SRN to store the hidden layer activity from previous time steps, is implicitly implemented in the lateral synapses of our WTA circuit. The SRN was successfully used to model human capabilities in artificial grammar learning tasks (Cleeremans and McClelland, 1991) (but see (Boucher and Dienes, 2003; Pothos, 2007), for alternative theories and models of artificial grammar learning).

We have also exhibited a strategy to increase the computational power of WTA circuits by using more advanced learning methods. The rejection sampling algorithm that was proposed here is one possible solution to this problem. It enables

the network to learn the temporal statistics with a much higher degree of accuracy, but at the same time it considerably increases the complexity of learning. Each input sequence must be replayed multiple times and thus, the convergence speed is decreased since many sampled paths will be rejected (in the experiment we resampled each path 10 times on average, therefore the learning time increased 10-fold). This makes learning possible on a long time scale only. However, the two mechanisms – pure forward sampling and rejection sampling – should not be seen as mutual exclusive strategies. Possibly both mechanisms could be found in biological systems. STDP might subserve to learn a quick preliminary representation of novel input statistics, while more complex models could emerge on a long time scale by selectively modulating the learning rate with global information. We demonstrated that in some cases a significant increase in learning performance can be achieved with only a small average number of resampled sequences. The experimental results suggest that for learning temporal sequences and simple grammars the pure implementation of STDP in WTA circuits is sufficient, whereas third-factor STDP rules become relevant for learning complex temporal structures.

Related work

The close relation between HMMs and recurrent neural networks was previously discovered and employed for deriving models for Bayesian computation in the cortex. These studies targeted the implementation of Bayesian filtering (R. P. Rao, 2004; Bobrowski et al., 2009), capturing the forward message of the belief propagation algorithm in a rate-based neural code, or using a two-state HMM to capture the dynamics of single neurons (Deneve, 2008a; Boerlin and Deneve, 2011). In the present study we directly analyzed spikes produced by WTA circuits in terms of samples from the state space of a HMM. For the HMM this results in an arguably weaker form of inference than belief propagation, but led in a straightforward manner to an analysis of learning in the network.

The emergence of predictive population codes in recurrent networks through synaptic plasticity and their importance for sequence learning was previously suggested and experimentally verified (Abbott and Blum, 1996; R. Rao and T. Sejnowski, 2001). In (Deneve, 2008b) it was shown that spiking neurons can learn the parameters of a 2-state HMM using synaptic plasticity, thereby implementing an online EM algorithm (Stiller and Radons, 1999; Mongillo and Deneve, 2008). In (Rezende et al., 2011) learning of temporal models was implemented through a variational approximation, and revealed STDP-like learning rules. In (Brea et al., 2011) it was shown that a network of neurons can learn to encode and reproduce a sequence of fixed spike times. The learning rules were derived using an importance sampling algorithm that yielded synaptic updates similar to the third-factor STDP rule presented here.

The crucial difference between (Brea et al., 2011) and our approach is the usage of WTA circuits as building blocks for the recurrent network instead of individual

neurons. Due to the possibility to use multiple WTAs our model has the freedom to factorize the multinomial HMM state space into smaller coupled variables, whereas (Brea et al., 2011) always fully factorizes the state space down to single binary variables. However, under the assumption of linear neurons the state-transition probabilities in all these models are always represented by only K^2 recurrent synapses. Thus the expressive power of all these models (with the same number of neurons) should be more or less identical. The optimal factorization of the state space may strongly depend on the task. Our experiments suggest that the restriction on the number of possible activity patterns due to the usage of WTAs seems minor compared to the crucial advantage of their intrinsic stabilizing effects of the network's activity. To the best of our knowledge this stabilization is the reason why the pure forward sampling learning approach performed so well in our experiments.

Contribution to a principled understanding of computation and plasticity in cortical microcircuits

The theoretical framework that we have introduced in this article provides a new and more principled understanding for the role of STDP in a generic cortical microcircuit motif (ensembles of pyramidal cells with lateral excitation and inhibition): Even in the absence of global signals related to reward, STDP installs in these microcircuit motifs an approximation to a HMM through forward sampling. The underlying theoretical analysis provides a new understanding of the role of spikes in such WTA circuits as samples from a (potentially very large) set of hidden states that enable generic cortical microcircuits to detect generic neural codes for afferent spike patterns that can reflect their temporal context and support predictions of future stimuli.

A remarkable feature of our model is that it postulates that noise in neural responses plays a very important role for the emergence of such "intelligent" temporal processing: We have shown that it provides in WTA circuits the basis for enabling probabilistic inference and learning through sampling, i.e. through an "embodiment" of probability distributions through neural activity. Thus stochasticity of neural responses provides an interesting alternative to models for probabilistic inference in biological neural systems through belief propagation (see (Lochmann and Deneve, 2011) for a review), i.e. through an emulation of an inherently deterministic calculation.

The rejection sampling algorithm that was proposed here as a method for emulating the full power of HMM learning requires in addition a mechanism that allows to replay input patterns multiple times. Such replay of complex spatiotemporal patterns is well documented in the hippocampus and was proposed as a mechanism for memory consolidation in the cortex (Buhry et al., 2011). This view is also supported by findings that showed that coordinated reactivation of temporal patterns can be observed in the cortex (Hoffman and McNaughton, 2002; Ji and Wilson, 2007;

Fujisawa et al., 2008; Peyrache et al., 2009). In our framework, samples generated by the WTA circuit must be replayed several times until the network produces a spike train that provides a sequence of hidden states that gives satisfactory explanations and predictions for all segments of the sequence. The number of times a sequence is replayed is proportional to the prediction error accumulated over the sequence, which is a measure for the sample quality. Thus, sequences that are novel and to that end not well represented in the network should be replayed more often and thus, they get more attention in the learning process. This view is supported by experimental data that revealed that transient novel experiences are replayed more prominently than familiar stimuli (Ribeiro et al., 2004; Cheng and L. M. Frank, 2008; S. Xu et al., 2012).

Altogether our results show that hidden Markov models provide a promising theoretical framework for understanding the emergence of all-important capabilities of the brain to understand and predict hidden states of complex time-varying sensory stimuli.

Chapter 3

Network plasticity as Bayesian inference

Contents

3.1 Introduction 44
3.2 Learning a posterior distribution through stochastic synaptic plasticity 47
3.3 Synaptic sampling improves the generalization capability of a neural network 51
3.4 Spine motility as synaptic sampling 54
3.5 Fast adaptation of synaptic connections and weights to a changing input statistics 57
3.6 Inherent network compensation capability through synaptic sampling 61
3.7 Discussion 64

Abstract. General results from statistical learning theory suggest to understand not only brain computations, but also brain plasticity as probabilistic inference. But a model for that has been missing. We propose that inherently stochastic features of synaptic plasticity and spine motility enable cortical networks of neurons to carry out probabilistic inference by sampling from a posterior distribution of network configurations. This model provides a viable alternative to existing models that propose convergence of parameters to maximum likelihood values. It explains how priors on weight distributions and connection probabilities can be merged optimally with learned experience, how cortical networks can generalize learned information so well to novel experiences, and how they can compensate continuously for unforeseen disturbances of the network. The resulting new theory of network plasticity explains from a functional perspective a number of experimental data on stochastic aspects of synaptic plasticity that previously appeared to be quite puzzling.

Acknowledgments and author contributions. This chapter is based on the manuscript

David Kappel, Stefan Habenschuss, Robert Legenstein, Wolfgang Maass (2015). "Network Plasticity as Bayesian Inference." *PLoS Computational Biology*.

To this study, I contributed as joint first author together with SH. The study was conceived by DK, SH, RL and WM, with the theory being developed by SH, RL and DK. The experiments were designed by DK, RL and WM, and were conducted by DK. The manuscript was written by DK, RL, SH and WM. The authors thank Seth Grant, Christopher Harvey, Jason MacLean and Simon Rumpel for helpful comments on the manuscript.

3.1 Introduction

We reexamine in this article the conceptual and mathematical framework for understanding the organization of plasticity in networks of neurons in the brain. We will focus on synaptic plasticity and network rewiring (spine motility) in this article, but our framework is also applicable to other network plasticity processes. One commonly assumes, that plasticity moves network parameters $\boldsymbol{\theta}$ (such as synaptic connections between neurons and synaptic weights) to values $\boldsymbol{\theta}^*$ that are optimal for the current computational function of the network. In learning theory, this view is made precise for example as maximum likelihood learning, where model parameters $\boldsymbol{\theta}$ are moved to values $\boldsymbol{\theta}^*$ that maximize the fit of the resulting internal model to the inputs $\mathbf{x}$ that impinge on the network from its environment (by maximizing the likelihood of these inputs $\mathbf{x}$). The convergence to $\boldsymbol{\theta}^*$ is often assumed to be facilitated by some external regulation of learning rates, that reduces the learning rate when the network approaches an optimal solution.

This view of network plasticity has been challenged on several grounds. From the theoretical perspective it is problematic because in the absence of an intelligent external controller it is likely to lead to overfitting of the internal model to the inputs $\mathbf{x}$ it has received, thereby reducing its capability to generalize learned knowledge to new inputs. Furthermore, networks of neurons in the brain are apparently exposed to a multitude of internal and external changes and perturbations, to which they have to respond quickly in order to maintain stable functionality.

Other experimental data point to surprising ongoing fluctuations in dendritic spines and spine volumes, to some extent even in the adult brain (A. Holtmaat and Svoboda, 2009) and in the absence of synaptic activity (Yasumatsu et al., 2008). Also a significant portion of axonal side branches and axonal boutons were found to appear and disapper within a week in adult visual cortex, even in the absence of imposed learning and lesions (Stettler et al., 2006). Furthermore surprising random drifts of tuning curves of neurons in motor cortex were observed (Rokni et al., 2007). Apart from such continuously ongoing changes in synaptic connections and tuning curves of neurons, massive changes in synaptic connectivity were found to accompany functional reorganization of primary visual cortex after lesions, see e.g. (Yamahachi et al., 2009).

We therefore propose to view network plasticity as a process that continuously moves high-dimensional network parameters $\boldsymbol{\theta}$ within some low-dimensional manifold that represents a compromise between overriding structural rules and different ways of fitting the internal model to external inputs $\mathbf{x}$. We propose that ongoing stochastic fluctuations (not unlike Brownian motion) continuously drive network parameters $\boldsymbol{\theta}$ within such low-dimensional manifold. The primary conceptual innovation is the departure from the traditional view of learning as moving parameters to values $\boldsymbol{\theta}^*$ that represent optimal (or locally optimal) fits to network inputs $\mathbf{x}$. We show that our alternative view can be turned into a precise learning model within the framework of probability theory. This new model satisfies theoretical requirements for handling priors such as structural constraints and rules in a principled manner, that have previously already been formulated and explored in the context of artificial neural networks (MacKay, 1992; Bishop, 2006), as well as more recent challenges that arise from probabilistic brain models (Pouget et al., 2013). The low-dimensional manifold of parameters $\boldsymbol{\theta}$ that becomes the new learning goal in our model can be characterized mathematically as the high probability regions of the posterior distribution $p^*(\boldsymbol{\theta}|\mathbf{x})$ of network parameters $\boldsymbol{\theta}$. This posterior arises as product of a general prior $p_{\mathcal{S}}(\boldsymbol{\theta})$ for network parameters (that enforces structural rules) with a term that describes the quality of the current internal model (e.g. in a predictive coding or generative modeling framework: the likelihood $p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta})$ of inputs $\mathbf{x}$ for the current parameter values $\boldsymbol{\theta}$ of the network $\mathcal{N}$). More precisely, we propose that brain plasticity mechanisms are designed to enable brain networks to sample from this posterior distribution $p^*(\boldsymbol{\theta}|\mathbf{x})$ through inherent stochastic features of their molecular implementation. In this way synaptic and other plasticity processes are able to carry out probabilistic (or Bayesian) inference through sampling from a posterior distribution that takes into account both structural rules and fitting to external inputs. Hence this model provides a solution to the challenge of (Pouget et al., 2013) to understand how posterior distributions of weights can be represented and learned by networks of neurons in the brain.

This new model proposes to reexamine rules for synaptic plasticity. Rather than viewing trial-to-trial variability and ongoing fluctuations of synaptic parameters as the result of a suboptimal implementation of an inherently deterministic plasticity process, it proposes to model experimental data on synaptic plasticity by rules that consist of three terms: the standard (typically deterministic) activity-dependent (e.g., Hebbian or STDP) term that fits the model to external inputs, a second term that enforces structural rules (priors), and a third term that provides the stochastic driving force. This stochastic force enables network parameters to sample from the posterior, i.e., to fluctuate between different possible solutions of the learning task. The stochastic third term can be modeled by a standard formalism (stochastic Wiener process) that had been developed to model Brownian motion. The first two terms can be modeled as drift terms in a stochastic process. A key insight is that one can easily relate details of the resulting more complex rules for the dynamics of network parameters $\boldsymbol{\theta}$, which now become stochastic differential equations, to specific features of the resulting posterior distribution

$p^*(\boldsymbol{\theta}|\mathbf{x})$ of parameter vectors $\boldsymbol{\theta}$ from which the network samples. Thereby, this theory provides a new framework for relating experimentally observed details of local plasticity mechanisms (including their typically stochastic implementation on the molecular scale) to functional consequences of network learning. For example, one gets a theoretically founded framework for relating experimental data on spine motility to experimentally observed network properties, such as sparse connectivity, specific distributions of synaptic weights, and the capability to compensate against perturbations (Marder, 2011).

We demonstrate the resulting new style of modeling network plasticity in three examples. These examples demonstrate how previously mentioned functional demands on network plasticity, such as incorporation of structural rules, automatic avoidance of overfitting, and inherent and immediate compensation for network perturbances, can be accomplished through stochastic local plasticity processes. We focus here on common models for unsupervised learning in networks of neurons: generative models. We first develop the general learning theory for this class of models, and then describe applications to common non-spiking and spiking generative network models. Both structural plasticity (see (May, 2011; Caroni et al., 2012) for reviews) and synaptic plasticity (STDP) are integrated into the resulting theory of network plasticity.

We present a new theoretical framework for analyzing and understanding local plasticity mechanisms of networks of neurons in the brain as stochastic processes, that generate specific distributions $p(\boldsymbol{\theta})$ of network parameters $\boldsymbol{\theta}$ over which these parameters fluctuate. This framework can be used to analyze and model many types of learning processes. We illustrate it here for the case of unsupervised learning, i.e., learning without a teacher or rewards. Obviously many learning processes in biological organisms are of this nature, especially learning processes in early sensory areas, and in other brain areas that have to provide and maintain on their own an adequate level of functionality, even in the face of internal or external perturbations.

A common framework for modeling unsupervised learning in networks of neurons are generative models, which date back to the 19th century, when Helmholtz proposed that perception could be understood as unconscious inference (Hatfield, 2002). Since then the hypothesis of the "generative brain" has been receiving considerable attention, fueling interest in various aspects of the relation between Bayesian inference and the brain (R. P. N. Rao et al., 2002; Doya et al., 2007; Pouget et al., 2013). The basic assumption of the "Bayesian brain" theory is that the activity $\boldsymbol{z}$ of neuronal networks in the brain can be viewed as an internal model for hidden variables in the outside world that give rise to sensory experiences $\boldsymbol{x}$ (such as the response $\boldsymbol{x}$ of auditory sensory neurons to spoken words that are guessed by an internal model $\boldsymbol{z}$). The internal model $\boldsymbol{z}$ is usually assumed to be represented by the activity of neurons in the network, e.g., in terms of the firing rates of neurons, or in terms of spatio-temporal spike patterns. A network $\mathcal{N}$ of stochastically firing neuron is modeled in this framework by a probability distribution $p_{\mathcal{N}}(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ that describes the probabilistic relationships between N input patterns $\mathbf{x} = \boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$

and corresponding network responses $\mathbf{z} = \boldsymbol{z}^1, \ldots, \boldsymbol{z}^N$, where $\boldsymbol{\theta}$ denotes the vector of network parameters that shape this distribution, e.g., via synaptic efficacies and network connectivity. The marginal probability $p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta}) = \sum_{\mathbf{z}} p_{\mathcal{N}}(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ of the actually occurring inputs $\mathbf{x} = \boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$ under the resulting internal model of the neural network $\mathcal{N}$ with parameters $\boldsymbol{\theta}$ can then be viewed as a measure for the agreement between this internal model (which carries out "predictive coding" (Winkler et al., 2012)) and its environment (which generates the inputs $\mathbf{x}$).

The goal of network learning is usually described in this probabilistic generative framework as finding parameter values $\boldsymbol{\theta}^*$ that maximize this agreement, or equivalently the likelihood of the inputs $\mathbf{x}$ (maximum likelihood learning):

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta}). \tag{3.1}$$

Locally optimal parameter solutions are usually determined by gradient ascent on the data likelihood $p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta})$.

3.2 Learning a posterior distribution through stochastic synaptic plasticity

In contrast, we assume here that not only a neural network $\mathcal{N}$, but also a prior $p_{\mathcal{S}}(\boldsymbol{\theta})$ for its parameters are given. This prior $p_{\mathcal{S}}$ can encode both structural constraints (such as sparse connectivity) and structural rules (e.g., a heavy-tailed distribution of synaptic weights). Then the goal of network learning becomes:

> learn the posterior distribution $p^*(\boldsymbol{\theta}|\mathbf{x})$ defined (up to normalization) by
>
> $p_{\mathcal{S}}(\boldsymbol{\theta}) \cdot p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta})$. (3.2)

The patterns $\mathbf{x} = \boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$ are assumed here to be regularly reoccurring network inputs.

A key insight (see Fig. 3.1 for an illustration) is that stochastic local plasticity rules for the parameters θ_i enable a network to achieve the learning goal (3.2): The distribution of network parameters $\boldsymbol{\theta}$ will converge after a while to the posterior distribution $p^*(\boldsymbol{\theta}) = p^*(\boldsymbol{\theta}|\mathbf{x})$ – and produce samples from it – if each network parameter θ_i obeys the dynamics

$$d\theta_i \;=\; b\left(\frac{\partial}{\partial\theta_i}\log p_{\mathcal{S}}(\boldsymbol{\theta}) + \frac{\partial}{\partial\theta_i}\log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta})\right)dt \;+\; \sqrt{2b}\,d\mathcal{W}_i\,, \tag{3.3}$$

where the learning rate $b > 0$ controls the speed of the parameter dynamics. Eq. (3.3) is a stochastic differential equation (see (Gardiner, 2004)), which differs from commonly considered differential equations through the stochastic term

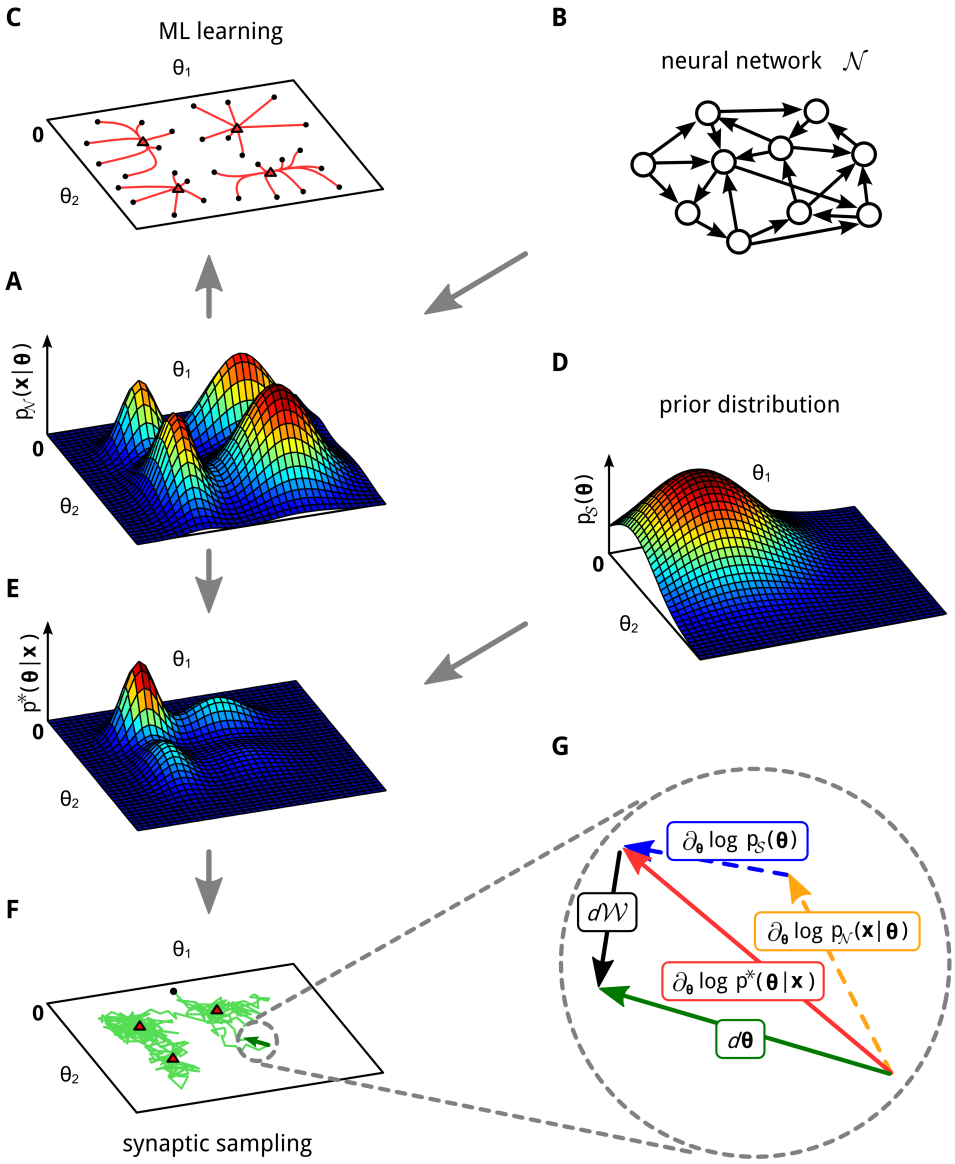

Fig. 3.1: Maximum likelihood (ML) learning vs. synaptic sampling. A,B,C: Illustration of ML learning for two parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)$ of a neural network $\mathcal{N}$. **A:** 3D plot of an example likelihood function. For a fixed set of inputs $\mathbf{x}$ it assigns a probability density (amplitude on z-axis) to each parameter setting $\boldsymbol{\theta}$. **B:** This likelihood function is defined by some underlying neural network $\mathcal{N}$. **C:** Multiple trajectories along the gradient of the likelihood function in (A). The parameters are initialized at random initial values (black dots) and then follow the gradient to a local maximum (red triangles). **D:** Example for a prior that prefers small values for $\boldsymbol{\theta}$. **E:** The posterior that results as product of the prior (D) and the likelihood (A). **F:** A single trajectory of synaptic sampling from the posterior (E), starting at the black dot. The parameter vector $\boldsymbol{\theta}$ fluctuates between different solutions, the visited values cluster near local optima (red triangles). **G:** Cartoon illustrating the dynamic forces (plasticity rule (3.3)) that enable →

→ the network to sample from the posterior distribution $p^*(\boldsymbol{\theta}|\mathbf{x})$ in (E). Magnification of one synaptic sampling step $d\boldsymbol{\theta}$ of the trajectory in (F) (green). The three forces acting on $\boldsymbol{\theta}$: the deterministic drift term (red) is directed to the next local maximum (red triangle), it consists of the first two terms in Eq. (3.3); the stochastic diffusion term $d\mathcal{W}$ (black) has a random direction. See Sec. C.2 for figure details.

$d\mathcal{W}_i$ that describes infinitesimal stochastic increments and decrements of a Wiener process $\mathcal{W}_i$. A Wiener process is a standard model for Brownian motion in one dimension (more precisely: the limit of a random walk with infinitesimal step size and normally distributed increments $\mathcal{W}_i^t - \mathcal{W}_i^s \sim \text{NORMAL}(0, t-s)$ between times t and s). Thus in an approximation of (3.3) for discrete time steps Δt the term $d\mathcal{W}_i$ can be replaced by Gaussian noise with variance Δt (see Eq. (3.7)). Note that Eq. (3.3) does not have a single solution $\theta_i(t)$, but a continuum of stochastic sample paths (see Fig. 3.1F for an example) that each describe one possible time course of the parameter θ_i.

Rigorous mathematical results based on Fokker-Planck equations (see Appendix C for details) allow us to infer from the stochastic local dynamics of the parameters θ_i given by a stochastic differential equation of the form (3.3) the probability that the parameter vector $\boldsymbol{\theta}$ can be found after a while in a particular region of the high-dimensional space in which it moves. The key result is that for the case of the stochastic dynamics according to Eq. (3.3) this probability is equal to the posterior $p^*(\boldsymbol{\theta}|\mathbf{x})$ given by Eq. (3.2). Hence the stochastic dynamics (3.3) of network parameters θ_i enables a network to achieve the learning goal (3.2): to learn the posterior distribution $p^*(\boldsymbol{\theta}|\mathbf{x})$. This posterior distribution is not represented in the network through any explicit neural code, but through its stochastic dynamics, as the unique stationary distribution of a Markov process from which it samples continuously. In particular, if most of the mass of this posterior distribution is concentrated on some low-dimensional manifold, the network parameters $\boldsymbol{\theta}$ will move most of the time within this low-dimensional manifold. Since this realization of the posterior distribution $p^*(\boldsymbol{\theta}|\mathbf{x})$ is achieved by sampling from it, we refer to this model defined by Eq. (3.3) (in the case where the parameters θ_i represent synaptic parameters) as *synaptic sampling*.

The stochastic term $d\mathcal{W}_i$ in Eq. (3.3) provides a simple integrative model for a multitude of biological and biochemical stochastic processes that effect the efficacy of a synaptic connection. The mammalian postsynaptic density comprises over 1000 different types of proteins (Coba et al., 2009). Many of those proteins that effect the amplitude of postsynaptic potentials and synaptic plasticity, for example NMDA receptors, occur in small numbers, and are subject to Brownian motion within the membrane (Ribrault et al., 2011). In addition, the turnover of important scaffolding proteins in the postsynaptic density such as PSD-95, which clusters glutamate receptors and is thought to have a substantial impact on synaptic efficacy, is relatively fast, on the time-scale of hours to days, depending on developmental state and environmental condition (Gray et al., 2006). Also the volume of spines at dendrites, which is assumed to be directly related to synaptic efficacy (Engert and

Bonhoeffer, 1999; Ho et al., 2011) is reported to fluctuate continuously, even in the absence of synaptic activity (Yasumatsu et al., 2008). Furthermore the stochastically varying internal states of multiple interacting biochemical signaling pathways in the postsynaptic neuron are likely to effect synaptic transmission and plasticity (Bhalla and Iyengar, 1999).

The contribution of the stochastic term dW_i in (3.3) can be scaled by a temperature parameter $\sqrt{T}$, where T can be any positive number. The resulting stationary distribution of $\boldsymbol{\theta}$ is proportional to $p^*(\boldsymbol{\theta})^{\frac{1}{T}}$, so that the dynamics of the stochastic process can be described by the energy landscape $\frac{\log p^*(\boldsymbol{\theta})}{T}$. For high values of T this energy landscape is flattened, i.e., the main modes of $p^*(\boldsymbol{\theta})$ become less pronounced. For $T \to 0$ the dynamics of $\boldsymbol{\theta}$ approaches a deterministic process and converges to the next local maximum of $p^*(\boldsymbol{\theta})$. Thus the learning process approximates for low values of T maximum a posteriori (MAP) inference (Bishop, 2006). We propose that this temperature parameter T is regulated in biological networks of neurons dependent on the developmental state, environment, and behavior of an organism. One can also accommodate a modulation of the dynamics of each individual parameter θ_i by a learning rate $b(\theta_i)$ that depends on its current value (see Appendix C).

Online synaptic sampling

For online learning one assumes that the likelihood $p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta}) = p_{\mathcal{N}}(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N|\boldsymbol{\theta})$ of the network inputs can be factorized:

$$p_{\mathcal{N}}(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N|\boldsymbol{\theta}) = \prod_{n=1}^{N} p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}), \tag{3.4}$$

i.e., each network input $\boldsymbol{x}^n$ can be explained as being drawn individually from $p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta})$, independently from other inputs.

The weight update rule (3.3) depends on all inputs $\mathbf{x} = \boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$, hence synapses have to keep track of the whole set of all network inputs for the exact dynamics (batch learning). In an online scenario, we assume that only the current network input $\boldsymbol{x}^n$ is available for synaptic sampling. One then arrives at the following online-approximation to (3.3)

$$d\theta_i \;=\; b\left(\frac{\partial}{\partial\theta_i}\log p_{\mathcal{S}}(\boldsymbol{\theta}) + N\,\frac{\partial}{\partial\theta_i}\,\log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta})\right)\,dt \;+\; \sqrt{2b}\,dW_i\,. \tag{3.5}$$

Note the additional factor N in the rule. It compensates for the N-fold summation of the first and last term in (3.5) when one moves through all N inputs $\boldsymbol{x}^n$. Although convergence to the correct posterior distribution cannot be guaranteed theoretically for this online rule, we show in Appendix C that the rule is a reasonable approximation to the batch-rule (3.3). Furthermore, all subsequent simulations are based on this online rule, which demonstrates the viability of this approximation.

Relationship to maximum likelihood learning

Typically, synaptic plasticity in generative network models is modeled as maximum likelihood learning. Time is often discretized into small discrete time steps Δt. For gradient-based approaches the parameter change $\Delta\theta_i^{ML}$ is then given by the gradient of the log likelihood multiplied with some learning rate η:

$$\Delta\theta_i^{ML} = \eta \frac{\partial}{\partial\theta_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}) \; . \tag{3.6}$$

To compare this maximum likelihood update with synaptic sampling, we consider a version of the parameter dynamics (3.5) for discrete time (see Appendix C for a derivation):

$$\Delta\theta_i = \eta \left(\frac{\partial}{\partial\theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + N \frac{\partial}{\partial\theta_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}) \right) + \sqrt{2\eta}\, \nu_i^t \;\; , \tag{3.7}$$

where the learning rate η is given by $\eta = b\,\Delta t$ and ν_i^t denotes Gaussian noise with zero mean and variance 1, drawn independently for each parameter θ_i and each update time t. We see that the maximum likelihood update (3.6) becomes one term in this online version of synaptic sampling. Equation (3.7) is a special case of the online Langevin sampler that was introduced in (Welling and Teh, 2011).

The first term $\frac{\partial}{\partial\theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta})$ in (3.7) arises from the prior $p_{\mathcal{S}}(\boldsymbol{\theta})$, and has apparently not been considered in previous rules for synaptic plasticity. An additional novel component is the Gaussian noise term ν_i^t (see also Fig. 3.1G). It arises because the accumulated impact of the Wiener process $\mathcal{W}_i$ over a time interval of length Δt is distributed according to a normal distribution with variance Δt. In contrast to traditional maximum likelihood optimization based on additive noise for escaping local optima, this noise term is not scaled down when learning approaches a local optimum. This ongoing noise is essential for enabling the network to sample from the posterior distribution $p^*(\boldsymbol{\theta})$ via continuously ongoing synaptic plasticity (see Fig. 3.1F).

3.3 Synaptic sampling improves the generalization capability of a neural network

The previously described theory for learning a posterior distribution over parameters $\boldsymbol{\theta}$ can be applied to all neural network models $\mathcal{N}$ where the derivative $\frac{\partial}{\partial\theta_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta})$ in (3.5) can be efficiently estimated. Since this term also has to be estimated for maximum likelihood learning (3.6), synaptic sampling can basically be applied to all neuron and network models that are amenable to maximum likelihood learning. We illustrate salient new features that result from synaptic sampling (i.e., plasticity rules (3.5) or (3.7)) for some of these models. We begin with the Boltzmann machine (Ackley et al., 1985), one of the oldest generative

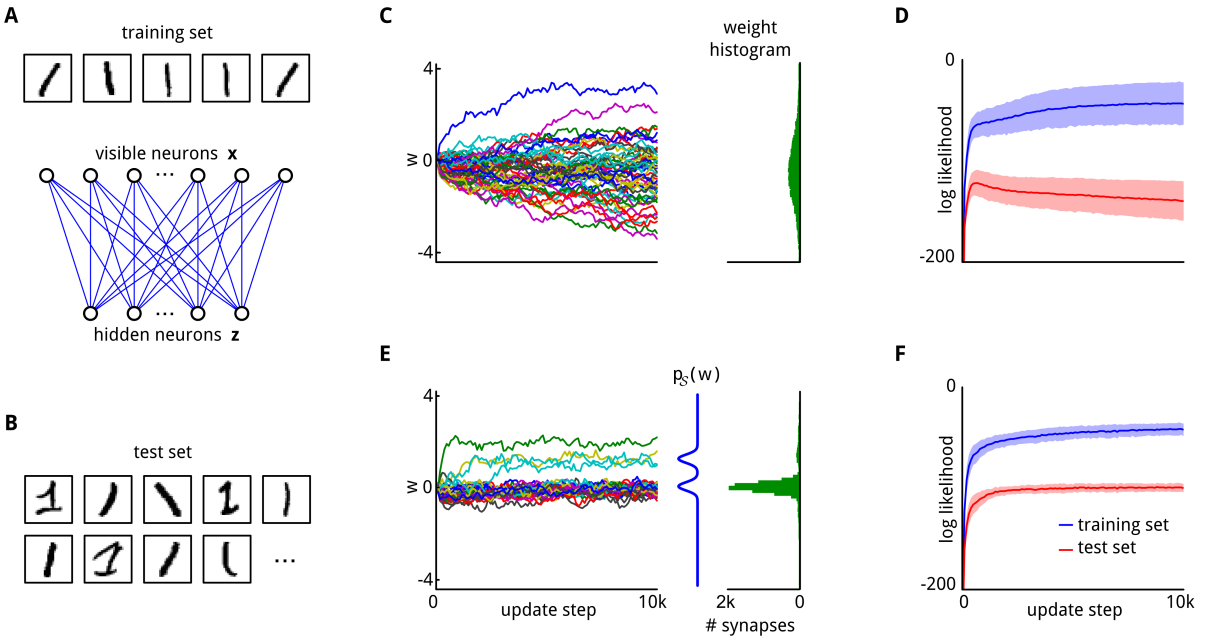

Fig. 3.2: Priors for synaptic weights improve generalization capability. A: The training set, consisting of five samples of a handwritten *1*. Below a cartoon illustrating the network architecture of the restricted Boltzmann machine (RBM), composed of a layer of 784 visible neurons $\mathbf{x}$ and a layer of 9 hidden neurons $\mathbf{z}$. **B:** Examples from the test set. It contains many different styles of writing that are not part of the training set. **C:** Evolution of 50 randomly selected synaptic weights throughout learning (on the training set). The weight histogram (right) shows the distribution of synaptic weights at the end of learning. 80 histogram bins were equally spaced between -4 and 4. **D:** Performance of the network in terms of log likelihood on the training set (blue) and on the test set (red) throughout learning. Mean values over 100 trial runs are shown, shaded area indicates std. The performance on the test set initially increases but degrades for prolonged learning. **E:** Evolution of 50 weights for the same network but with a bimodal prior. The prior $p_S(w)$ is indicated by the blue curve. Most synaptic weights settle in the mode around 0, but a few larger weights also emerge and stabilize in the larger mode. Weight histogram (green) as in (C). **F:** The log likelihood on the test set maintains a constant high value throughout the whole learning session, compare to (D).

neural network models. It is currently still extensively investigated in the context of deep learning (G. Hinton et al., 2006; Salakhutdinov and G. Hinton, 2012). We demonstrate in Fig. 3.2D,F the improved generalization capability of this model for the learning approach (3.2) (learning of the posterior), compared with maximum likelihood learning (approach (3.1)), which had been theoretically predicted by (MacKay, 1992) and (Bishop, 2006). But this model for learning the posterior (approach (3.2)) in Boltzmann machines is now based on local plasticity rules. Note that the Boltzmann machine with synaptic sampling samples simultaneously on two different time scales: In addition to sampling for given parameters $\boldsymbol{\theta}$ from likely network states in the usual manner, it now samples simultaneously on a slower time scale according to (3.7) from the posterior of network parameters $\boldsymbol{\theta}$.

A Boltzmann machine employs extremely simple non-spiking neuron models with binary outputs. Neuron y_i outputs 1 with probability $\sigma(\sum_j w_{ij} y_j + b_i)$, else 0, where σ is the logistic sigmoid $\sigma(u) = \frac{1}{1+e^{-u}}$, with synaptic weights w_{ij} and bias

parameters b_i. Synaptic connections in a Boltzmann machine are bidirectional, with symmetric weights ($w_{ij} = w_{ji}$). The parameters $\boldsymbol{\theta}$ for the Boltzmann machine consist of all weights w_{ij} and biases b_i in the network. For the special case of a restricted Boltzmann machine (RBM), maximum likelihood learning of these parameters can be done efficiently (G. E. Hinton, 2002), and therefore RBM's are typically used for deep learning. An RBM has a layered structure with one layer of visible neurons $\mathbf{x}$ and a second layer of hidden neurons $\mathbf{z}$. Synaptic connections are formed only between neurons on different layers (Fig. 3.2A). The maximum likelihood gradients $\Delta w_{ij}^{ML} = \frac{\partial}{\partial w_{ij}} \log p_{\mathcal{N}}(\boldsymbol{x}|\boldsymbol{\theta})$ and $\Delta b_i^{ML} = \frac{\partial}{\partial b_i} \log p_{\mathcal{N}}(\boldsymbol{x}|\boldsymbol{\theta})$ can be efficiently approximated for this model, for example

$$\frac{\partial}{\partial w_{ij}} \log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}) \approx z_i^n x_j^n - \hat{z}_i^n \hat{x}_j^n \ , \tag{3.8}$$

where x_j^n is the output of input neuron j while input $\boldsymbol{x}^n$ is presented, and $\hat{x}_j^n$ its output during a subsequent phase of spontaneous activity ("reconstruction phase"); analogously for the hidden neuron z_j (see Appendix C).

We integrated this maximum likelihood estimate (3.8) into the synaptic sampling rule (3.7) in order to test whether a suitable prior $p_{\mathcal{S}}(\mathbf{w})$ for the weights improves the generalization capability of the network. The network received as input just five samples $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^5$ of a handwritten Arabic number *1* from the MNIST dataset (the training set, shown in Fig. 3.2A) that were repeatedly presented. Each pixel of the digit images was represented by one neuron in the visible layer (which consisted of 784 neurons). We selected a second set of 100 samples of the handwritten digit *1* from the MNIST dataset as test set (Fig. 3.2B). These samples include completely different styles of writing that were not present in the training set. After allowing the network to learn the five input samples from Fig. 3.2A for various numbers of update steps (horizontal axis of Fig. 3.2D,F), we evaluated the learned internal model of this network $\mathcal{N}$ for the digit *1* by measuring the average log-likelihood $\log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta})$ for the test data. The result is indicated in Fig. 3.2D,F for the training samples by the blue curves, and for the new test examples, that were never shown while synaptic plasticity was active, by the red curves.

First, a uniform prior over the synaptic weights was used (Fig. 3.2C), which corresponds to the common maximum likelihood learning paradigm (3.8). The performance on the test set (shown on vertical axis) initially increases but degrades for prolonged exposure to the training set (length of that prior exposure shown on horizontal axis). This effect is known as overfitting (Bishop, 2006; MacKay, 1992). It can be reduced by choosing a suitable prior $p_{\mathcal{S}}(\boldsymbol{\theta})$ in the synaptic sampling rule (3.7). The choice for the prior distribution is best if it matches the statistics of the training samples (MacKay, 1992), which has in this case two modes (resulting from black and white pixels). The presence of this prior in the learning rule maintains good generalization capability for test samples even after prolonged exposure to the training set (red curve in Fig. 3.2F).

The improved generalization capability of the network is a result of the prior distribution. It is well known that the prior in Bayesian inference allows to effectively

prevent overfitting by making solutions that use fewer or smaller parameters more likely. Similar results would therefore emerge in any other implementation of Bayesian learning in neural networks. A thorough discussion on this topic which is known as *Bayesian regularization* can be found in (MacKay, 1992; Bishop, 2006).

As a consequence, the choice of the prior distribution can have a significant impact on the learning result. In Appendix C we compared a set of different priors and demonstrate this effect more systematically. There it can also be seen that if the choice of the prior is bad, the learning performance can even get worse than in the case without a prior.

3.4 Spine motility as synaptic sampling

In the following sections we apply our synaptic sampling framework to networks of spiking neurons and biological models for network plasticity. The number and volume of spines for a synaptic connection is thought to be directly related to its synaptic weight (Loewenstein et al., 2011). Experimental studies have provided a wealth of information about the stochastic dynamics of dendritic spines (see e.g. (Trachtenberg et al., 2002; Zuo et al., 2005; A. J. Holtmaat et al., 2005; A. Holtmaat and Svoboda, 2009; Loewenstein et al., 2011; Loewenstein et al., 2015)). They demonstrate that the volume of a substantial fraction of dendritic spines varies continuously over time, and that all the time new spines and synaptic connections are formed and existing ones are eliminated. We show that these experimental data on spine motility can be understood as special cases of synaptic sampling. The synaptic sampling framework is however very general, and many different models for spine motility can be derived from it as special cases. We demonstrate this here for one simple model, induced by the following assumptions:

1. We restrict ourselves to plasticity of excitatory synapses, although the framework is general enough to apply to inhibitory synapses as well.
2. In accordance with experimental studies (Loewenstein et al., 2011), we require that spine sizes have a multiplicative dynamics, i.e., that the amount of change within some given time window is proportional to the current size of the spine.
3. We assume here for simplicity that a synaptic connection between two neurons is realized by a single spine and that there is a single parameter θ_i for each potential synaptic connection i.

The last requirement can be met by encoding the state of the synapse in an abstract form, that represents synaptic connectivity and synaptic plasticity in a single parameter θ_i. We define that negative values of θ_i represent a current disconnection and positive values represent a functional synaptic connection. The distance of the current value of θ_i from zero indicates how likely it is that the synapse will soon reconnect (for negative values) or withdraw (for positive values), see Fig. 3.3A. In

addition the synaptic parameter θ_i encodes for positive values the synaptic efficacy w_i, i.e., the resulting EPSP amplitudes, by a simple mapping $w_i = f(\theta_i)$.

A large class of mapping functions f is supported by our theory (see Sec. C.4 for details). The second assumption which requires multiplicative synaptic dynamics supports an exponential function f in our model, in accordance with previous models of spine motility (Loewenstein et al., 2011). Thus, we assume in the following that the efficacy w_i of synapse i is given by

$$w_i = \exp(\theta_i - \theta_0) \ , \tag{3.9}$$

see Fig. 3.3C. Note that for a large enough offset θ_0, negative parameter values θ_i (which model a non-functional synaptic connection) are automatically mapped onto a tiny region close to zero in the w-space, so that retracted spines have essentially zero synaptic efficacy. The general rule for online synaptic sampling (3.5) for the exponential mapping (3.9) can be written as (see Sec. C.4)

$$d\theta_i = b\left(\frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) \ + \ N w_i \, \frac{\partial}{\partial w_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n | \boldsymbol{w})\right) dt \ + \ \sqrt{2b}\, d\mathcal{W}_i \ . \tag{3.10}$$

In equation (3.10) the multiplicative synaptic dynamics becomes explicit. The gradient $\frac{\partial}{\partial w_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n | \boldsymbol{w})$, i.e., the activity-dependent contribution to synaptic plasticity, is weighted by w_i. Hence, for negative values of θ_i (non-functional synaptic connection), the activities of the pre- and post-synaptic neurons have negligible impact on the dynamics of the synapse. Assuming a large enough θ_0, retracted synapses therefore evolve solely according to the prior $p_{\mathcal{S}}(\boldsymbol{\theta})$ and the random fluctuations $d\mathcal{W}_i$. For large values of θ_i the opposite is the case. The influence of the prior $\frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta})$ and the Wiener process $d\mathcal{W}_i$ become negligible, and the dynamics is dominated by the activity-dependent likelihood term. Large synapses can therefore become quite stable if the presynaptic activity is strong and reliable (see Fig. 3.3B). Through the use of parameters $\boldsymbol{\theta}$ which determine both synaptic connectivity and synaptic efficacies, the synaptic sampling framework provides a unified model for structural and synaptic plasticity. The prior distribution can have significant impact on the spine motility, encouraging for example sparser or denser synaptic connectivity. If the activity-dependent second term in Eq. (3.10), that tries to maximize the likelihood, is small (e.g., because θ_i is small or parameters are near a mode of the likelihood) then Eq. (3.10) implements an Ornstein Uhlenbeck process. This prediction of our model is consistent with a previous analysis which showed that an Ornstein Uhlenbeck process is a viable model for synaptic spine motility (Loewenstein et al., 2011).

The weight dynamics that emerges through the stochastic process (3.10) is illustrated in the right panel of Fig. 3.3D. A Gaussian parameter prior $p_{\mathcal{S}}(\theta_i)$ results in a log-normal prior $p_{\mathcal{S}}(w_i)$ in a corresponding stochastic differential equation for synaptic efficacies w_i (see Sec. C.4 for details).

The last term (noise term) in our synaptic sampling rule (3.10) predicts that eliminated connections spontaneously regrow at irregular intervals. In this way they can

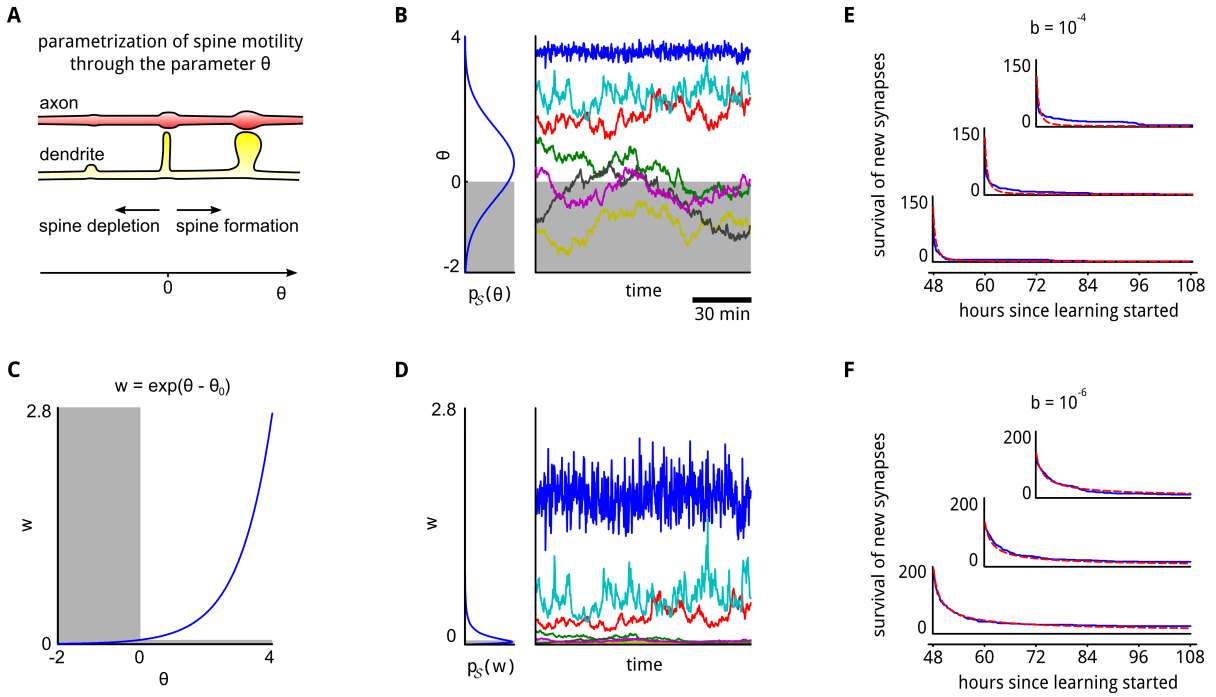

Fig. 3.3: Integration of spine motility into the synaptic sampling model. A: Illustration of the parametrization of spine motility. Values $\theta > 0$ indicate a functional synaptic connection. **B:** A Gaussian prior $p_S(\theta)$, and a few stochastic sample trajectories of θ according to the synaptic sampling rule (3.10). Negative values of θ (gray area) are interpreted as non-functional connections. Some stable synaptic connections emerge (traces in the upper half), whereas other synaptic connections come and go (traces in lower half). All traces, as well as survival statistics shown in (E,F), are taken from the network simulation described in detail in the next section and Appendix C. **C:** The exponential function maps synapse parameters θ to synaptic efficacies w. Negative values of θ, corresponding to (retracted) spines are mapped to a tiny region close to zero in the w-space. **D:** The Gaussian prior in the θ-space translates to a log-normal distribution in the w-space. The traces from (B) are shown in the right panel transformed into the w-space. Only persistent synaptic connections contribute substantial synaptic efficacies. **E,F:** The emergent survival statistics of newly formed synaptic connections, (i.e., formed during the preceding 12 hours) evaluated at three different start times throughout learning (blue traces, axes are aligned with start times of the analyses). The survival statistics exhibit in our synaptic sampling model a power-law behavior (red curves, see Sec. C.5). The time-scale (and exponent of the power-law) depends on the learning rate b in equation (3.10), and can assume any value in our quite general model (shown is $b = 10^{-4}$ in (E) and $b = 10^{-6}$ in (F)).

test whether they can contribute to explaining the input. If they cannot contribute, they disappear again. The resulting power-law behavior of the survival of newly formed synaptic connections (Fig. 3.3E,F) matches corresponding new experimental data (Loewenstein et al., 2015) and is qualitatively similar to earlier experimental results which revealed a quick decay of transient dendritic spines (Yang et al., 2009; Zuo et al., 2005; A. J. Holtmaat et al., 2005). Functional consequences of this structural plasticity are explored in the following sections.

3.5 Fast adaptation of synaptic connections and weights to a changing input statistics

We will explore in this and the next section implications of the synaptic sampling rule (3.10) for network plasticity in simple generative spike-based neural network models.

The main types of spike-based generative neural network models that have been proposed are (Brea et al., 2013; Boerlin et al., 2013; Nessler et al., 2013; Habenschuss et al., 2013). We focus here on the type of models introduced by (Nessler et al., 2013; Habenschuss et al., 2013; Kappel et al., 2014), since these models allow an easy estimation of the likelihood gradient (the second term in (3.10)) and can relate this likelihood term to STDP. Since these spike-based neural network models have non-symmetric synaptic connections (that model chemical synapses between pyramidal cells in the cortex), they do not allow to regenerate inputs $\boldsymbol{x}$ from internal responses $\boldsymbol{z}$ by running the network backwards (like in a Boltzmann machine). Rather they are *implicit* generative models, where synaptic weights from inputs to hidden neurons are interpreted as implicit models for presynaptic activity, given that the postsynaptic neuron fires.

We focus in this section on a simple model for an ubiquitous cortical microcircuit motif: an ensemble of pyramidal cells with lateral inhibition, often referred to as Winner-Take-All (WTA) circuit. It has been proposed that this microcircuit motif provides for computational analysis an important bridge between single neurons and larger brain systems (Carandini, 2012). We employ a simple form of divisive normalization (as proposed by (Carandini, 2012); see Appendix C) to model lateral inhibition, thereby arriving at a theoretically tractable version of this microcircuit motif that allows us to compute the maximum likelihood term (second term in (3.10)) in the synaptic sampling rule. We assumed Gaussian prior distributions $p_S(\theta_i)$, with mean μ and variance σ^2 over the synaptic parameters θ_i (as in Fig. 3.3B). Then the synaptic sampling rule (3.10) yields for this model

$$d\theta_i = b\left(\frac{1}{\sigma^2}\,(\mu - \theta_i) \;+\; N w_i\, S(t)\,(x_i(t) - \alpha\, e^{w_i})\right)dt \;+\; \sqrt{2b}\, d\mathcal{W}_i\,, \tag{3.11}$$

where $S(t)$ denotes the spike train of the postsynaptic neuron and $x_i(t)$ denotes the weight-normalized value of the sum of EPSPs from presynaptic neuron i at

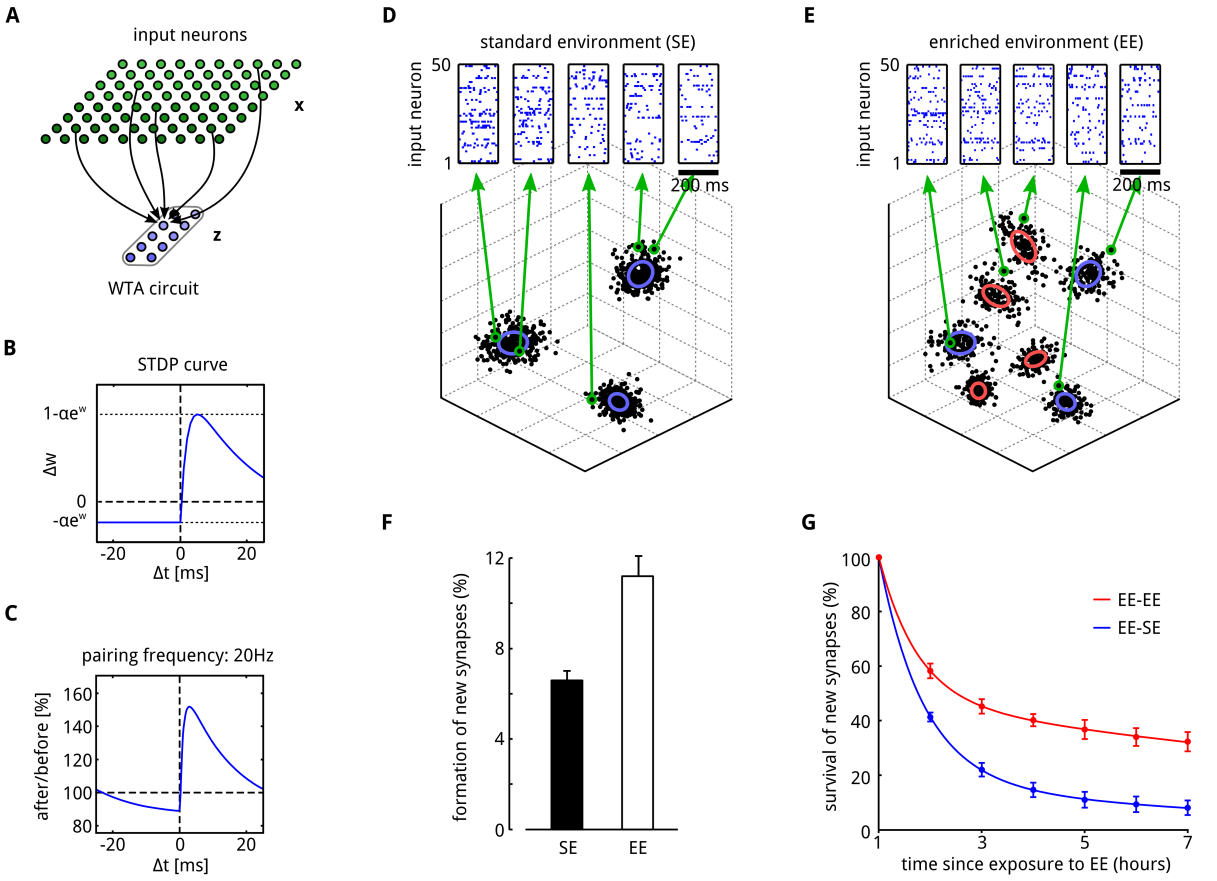

Fig. 3.4: Adaptation of synaptic connections to changing input statistics through synaptic sampling. A: Illustration of the network architecture. A WTA circuit consisting of ten neurons $\mathbf{z}$ receives afferent stimuli from input neurons $\mathbf{x}$ (few connections shown for a single neuron in $\mathbf{z}$). **B:** The STDP learning curve that arises from the likelihood term in equation (3.11). **C:** Measured STDP curve that results from a related STDP rule for a moderate pairing frequency of 20 Hz, as in (Sjöström et al., 2001). (Figure adapted from (Nessler et al., 2013)). **D,E:** Each sensory experience was modeled by 200 ms long spiking activity of 1000 input neurons, that covered some 3D data space with Gaussian tuning curves (the results do not depend on the finite dimension of the data space, we chose 3 dimension for easier visualization). Insets show the firing activity of randomly chosen 50 of the 1000 input neurons for the sample data points marked by green circles. Objects in the environment were represented by Gaussian clusters (ellipses) in this finite dimensional data space. **F:** During learning phase 1 (3 hours) only samples from SE were presented to the network, in phase 2 (which lasted 1 hour) samples from EE. Shortly after the transition from SE to EE the number of newly formed synaptic connections significantly increases (compare to Fig. 1h in (Yang et al., 2009)). **G:** Comparison of the survival of synapses for a network with persistent exposure to EE (EE-EE condition) and a network that was returned to SE (EE-SE condition). Newly formed synaptic connections are transient and quickly decay after formation. A significantly larger fraction of synapses persists if the network continuously receives EE inputs (compare to Fig. 2c in (Yang et al., 2009)). The dots show the means of measurements taken every 30 minutes, the lines represent two-term exponential fits ($r^2 = 1$). The results in (F, G) show means over 5 trial runs. Error bars indicate STD.

time t (i.e., the summed EPSPs that would arise for weight $w_i = 1$; see Appendix C for details). α is a parameter that scales the impact of synaptic plasticity depending on the current synaptic efficacy. The resulting activity-dependent component $S(t)(x_i(t) - \alpha\, e^{w_i})$ of the likelihood term is a simplified version of the standard STDP learning rule (Fig. 3.4B, C), like in (Nessler et al., 2013; Klampfl and Maass, 2013). Synaptic plasticity (STDP) for connections from input neurons to pyramidal cells in the WTA circuit can be understood from the generative aspect as fitting a mixture of Poisson (or other exponential family) distributions to high-dimensional spike inputs (Nessler et al., 2013; Habenschuss et al., 2013). The factor $w_i = \exp(\theta_i - \theta_0)$ had been discussed in (Nessler et al., 2013), because it is compatible with the underlying generative model, but provides in addition a better fit to the experimental data of (Sjöström et al., 2001). We examine in this section emergent properties of network plasticity in this simple spike-based neural network under the synaptic sampling rule (3.11).

It is well documented that cortical dendritic spines are transient and that spine turnover is enhanced by novel experience and training (Yang et al., 2009; Hofer et al., 2009; Kuhlman et al., 2014). For example, enhanced spine formation as a consequence of sensory enrichment was found in mouse somatosensory cortex (Yang et al., 2009). In this study the animals were exposed to a new sensory environment by adding additional objects to their home cage. This sensory enrichment resulted in a rapid increase in the formation of new spines. If the exposure to the enriched environment was only brief, the newly formed spines quickly decayed.

We wondered whether these experimentally observed effects also emerge in our synaptic sampling model. As in (Yang et al., 2009) we exposed the network to different sensory environments to study these effects. Sensory experiences typically involve several processing steps and interactions between multiple brain systems, and precise knowledge about their cortical representation is still missing. Therefore we used here a simple symbolic representation of the sensory environment. We represented each sensory experience by a point in some finite dimensional space which is covered by the tuning curves of a large number of input neurons. Their spike output was then communicated to the WTA circuit in the form of 200 ms-long spike patterns of the 1000 input neurons (see Fig. 3.4D,E and Appendix C for details). Independently drawn sensory experiences were presented sequentially and synaptic sampling according to (3.11) was applied continuously to all synapses from the 1000 input neurons to the ten neurons in the WTA circuit.

Each environment was represented as a mixture of Gaussians (clusters) of points in the finite-dimensional sensory space. Each cluster could represent for example different sensory experiences with some object in the environment. Consequently we modelled an enriched environment (EE) simply by adding a few new clusters to the standard environment (SE). In phase 1 the network was exposed to an environment with 3 clusters (standard environment (SE), see Fig. 3.4D). After 3 hours the network input was enriched by adding 4 additional clusters (enriched environment (EE), see Fig. 3.4E). We found that exposure to EE significantly

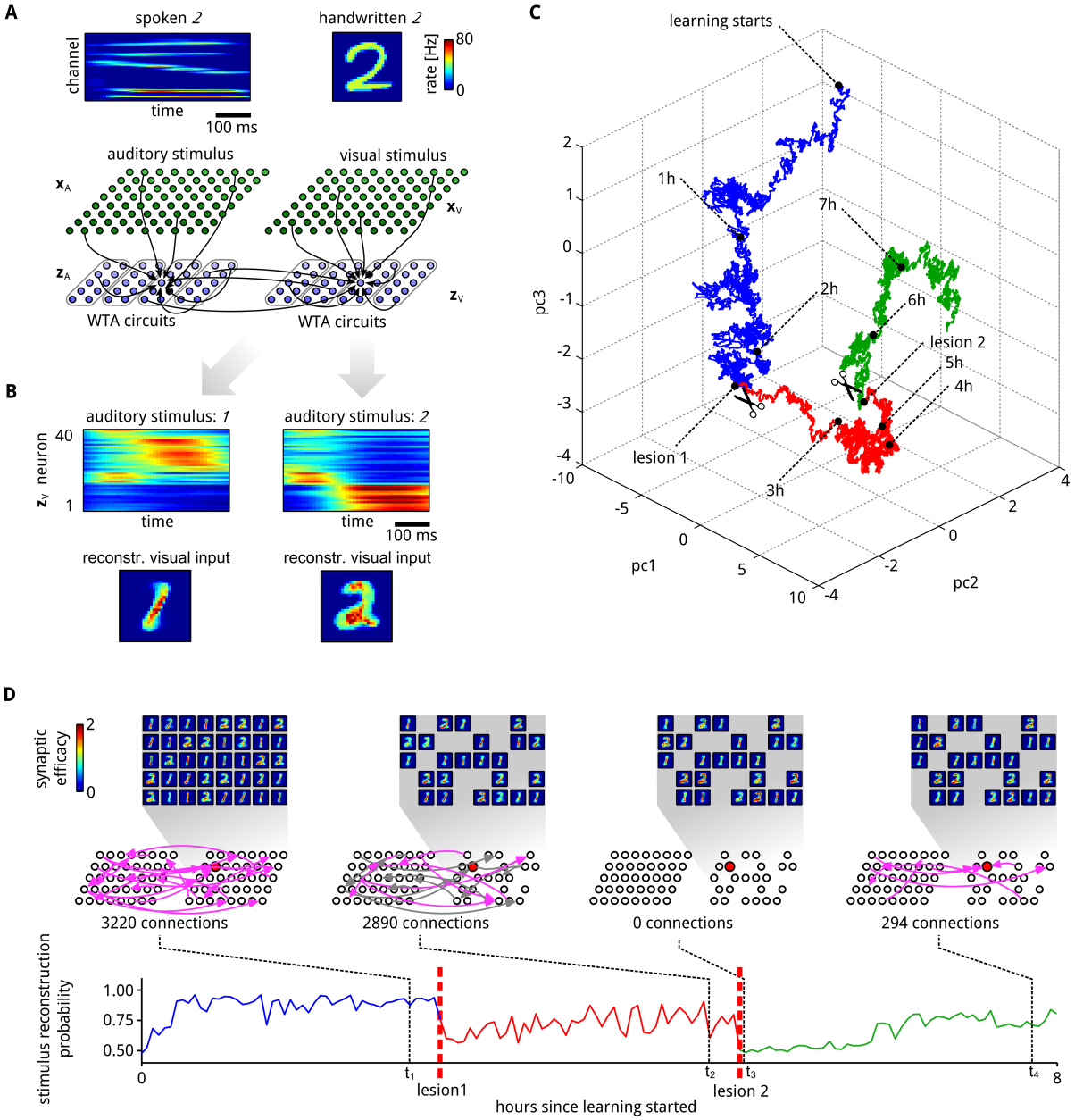

Fig. 3.5: Inherent compensation for network perturbations. A: A spike-based generative neural network (illustrated at the bottom) received simultaneously spoken and handwritten representations of the same digit (and for tests only spoken digits, see (B)). Stimulus examples for spoken and written digit *2* are shown at the top. These inputs are presented to the network through corresponding firing rates of "auditory" ($\mathbf{x}_A$) and "visual" ($\mathbf{x}_V$) input neurons. Two populations $\mathbf{z}_A$ and $\mathbf{z}_V$ of 40 neurons, each consisting of four WTA circuits like in Fig. 3.4, receive exclusively auditory or visual inputs. In addition, arbitrary lateral excitatory connections between these "hidden" neurons are allowed. **B:** Assemblies of hidden neurons emerge that encode the presented digit (*1* or *2*). Top panel shows PETH of all neurons from $\mathbf{z}_V$ for stimulus *1* (left) and *2* (right) after learning, when only an auditory stimulus is presented. Neurons are sorted by the time of their highest average firing. Although only auditory stimuli are presented, it is possible to reconstruct an internally generated "guessed" visual stimulus that represents the same digit (bottom). **C:** First three PCA components of the temporal evolution of a subset θ' of network parameters θ. Two $\rightarrow$

→ major lesions were applied to the network. In the first lesion (transition to red) all neurons that significantly encode stimulus *2* were removed from the population $\mathbf{z}_V$. In the second lesion (transition to green) all currently existing synaptic connections between neuron in $\mathbf{z}_A$ and $\mathbf{z}_V$ were removed, and not allowed to regrow. After each lesion the network parameters $\boldsymbol{\theta}'$ migrate to a new manifold. **D:** The generative reconstruction performance of the "visual" neurons $\mathbf{z}_V$ for the test case when only an auditory stimulus is presented was tracked throughout the whole learning session, including lesions *1* and *2* (bottom panel). After each lesion the performance strongly degrades, but reliably recovers. Insets show at the top the synaptic weights of neurons in $\mathbf{z}_V$ at 4 time points $t_1, \ldots, t_4$, projected back into the input space like in Fig. 3.4E. Network diagrams in the middle show ongoing network rewiring for synaptic connections between the hidden neurons $\mathbf{z}_A$ and $\mathbf{z}_V$. Each arrow indicates a functional connection between two neurons. To keep the figure uncluttered only subsets of synapses are shown (1% randomly drawn from the total set of possible lateral connections). Connections at time t_2 that were already functional at time t_1 are plotted in gray. The neuron whose parameter vector $\boldsymbol{\theta}'$ is tracked in (C) is highlighted in red. The text under the network diagrams shows the total number of functional connections between hidden neurons at the time point.

increased the rate of new synapse formation as in the experimental result of (Yang et al., 2009) (Fig. 3.4F).

Most of the newly formed synapses decayed within a few hours after return to the standard environment (EE-SE situation, see Fig. 3.4G). In this case only about about 8% become stable. A fraction of about 30% becomes stable when the enriched environment was maintained (EE-EE situation). These results qualitatively reproduce the findings from mouse barrel cortex (compare Figures 1h and 2c in (Yang et al., 2009)). Note that we used here relatively large update rates b to keep simulation times in a feasible range, which results in spine dynamics on the time scale of hours instead of days as in biological synapses (Yang et al., 2009).

3.6 Inherent network compensation capability through synaptic sampling

Numerous experimental data show that the same function of a neural circuit is achieved in different individuals with drastically different parameters, and also that a single organism can compensate for disturbances by moving to a new parameter vector (Tang et al., 2010; Grashow et al., 2010; Marder and Taylor, 2011; Marder, 2011; Prinz et al., 2004). These results suggest that there exists some low-dimensional submanifold of values for the high-dimensional parameter vector $\boldsymbol{\theta}$ of a biological neural network that all provide stable network function (degeneracy). We propose that the previously discussed posterior distribution of network parameters $\boldsymbol{\theta}$ provides a mathematical model for such low-dimensional submanifold. Furthermore we propose that the underlying continuous stochastic fluctuation $d\mathcal{W}$ provides a driving force that automatically moves network parameters (with high probability) to a functionally more attractive regime when the current solution performs worse because of perturbations, such as lesions of neurons or network connections. This compensation capability is not an add-on to the synaptic sampling model, but an inherent feature of its organization.

We demonstrate this inherent compensation capability in Fig. 3.5 for a generative spiking neural network with synaptic parameters $\boldsymbol{\theta}$ that regulate simultaneously structural plasticity and synaptic plasticity (dynamics of weights) as in Fig. 3.3 and 3.4. The prior $p_S(\boldsymbol{\theta})$ for these parameters is here the same as in the preceding section (see Fig. 3.4G on the left). But in contrast to the previous section we consider here a network that allows us to study the self-organization of connections *between* hidden neurons. The network consists of eight WTA-circuits, but in contrast to Fig. 3.4 we allow here arbitrary excitatory synaptic connections between neurons within the same or different ones of these WTA circuits. This network models multi-modal sensory integration and association in a simplified manner. Two populations of "auditory" and "visual" input neurons $\mathbf{x}_A$ and $\mathbf{x}_V$ project onto corresponding populations $\mathbf{z}_A$ and $\mathbf{z}_V$ of hidden neurons (each consisting of one half of the WTA circuits, see lower panel of Fig. 3.5A). Only a fraction of the potential synaptic connections became functional (see Fig. S2A in Sec. C.7) through the synaptic sampling rule (3.11) that integrates structural and synaptic plasticity. Synaptic weights and connections were not forced to be symmetric or bidirectional.

As in the previous demonstrations we do not use external rewards or teacher-inputs for guiding network plasticity. Rather, we allow the model to discover on its own regularities in its network inputs. The "auditory" hidden neurons $\mathbf{z}_A$ on the left in Fig. 3.5A received temporal spike patterns from the auditory input neurons $\mathbf{x}_A$ that were generated from spoken utterings of the digit *1* and *2* (which lasted between 320 ms and 520 ms). Simultaneously we presented to the "visual" hidden neurons $\mathbf{z}_V$ on the right for the same time period a (symbolic) visual representation of the same digit (randomly drawn from the MNIST database like in Fig. 3.2).

The emergent associations between the two populations $\mathbf{z}_A$ and $\mathbf{z}_V$ of hidden neurons were tested by presenting auditory input only and observing the activity of the "visual" hidden neurons $\mathbf{z}_V$. Fig. 3.5B shows the emergent activity of the neurons $\mathbf{z}_V$ when only the auditory stimulus was presented (visual input neurons $\mathbf{x}_V$ remained silent). The generative aspect of the network can be demonstrated by reconstructing for this case the visual stimulus from the activity of the "visual" hidden neurons $\mathbf{z}_V$. Fig. 3.5B shows reconstructed visual stimuli from a single run where only the auditory stimuli $\mathbf{x}_A$ for digits *1* (left) and *2* (right) were presented to the network. Digit images were reconstructed by multiplying the synaptic efficacies of synapses from neurons in $\mathbf{x}_V$ to neurons in $\mathbf{z}_V$ (which did not receive any input from $\mathbf{x}_V$ in this experiment) with the instantaneous firing rates of the corresponding $\mathbf{z}_V$-neurons.

Interestingly we found that synaptic sampling significantly outperforms the pure deterministic STDP updates introduced in (Kappel et al., 2014), which do not impose a prior distribution over synaptic parameters. The structural prior that favors solutions with only a small number of large synaptic weights seems to be beneficial for this task as it allows to learn few but pronounced associations between the neurons (see Sec. C.7).

In order to investigate the inherent compensation capability of synaptic sampling,

we applied two lesions to the network within a learning session of 8 hours. In the first lesion all neurons (16 out of 40) that became tuned for digit *2* in the preceding learning (see Fig. 3.5D and Sec. C.7) were removed. The lesion significantly impaired the performance of the network in stimulus reconstruction, but it was able to recover from the lesion after about one hour of continuing network plasticity according to Eq. (3.11) (Fig. 3.5D). The reconstruction performance of the network was measured here continuously through the capability of a linear readout neuron from the visual ensemble to classify the current auditory stimulus (*1* or *2*).

In the second lesion all synaptic connections between hidden neurons that were present after recovery from the first lesion were removed and not allowed to regrow (2936 synapses in total). After about two hours of continuing synaptic sampling 294 new synaptic connections between hidden neurons emerged. These made it again possible to infer the auditory stimulus from the activity of the remaining 24 hidden neurons in the population $\mathbf{z}_V$ (in the absence of any input from the population $\mathbf{x}_V$), at about 75% of the performance level before the second lesion (see bottom panel of Fig. 3.5D).

In order to illustrate the ongoing network reconfiguration we track in Fig. 3.5C the temporal evolution of a subset $\boldsymbol{\theta}'$ of network parameters (35 parameters θ_i associated with the potential synaptic connections of the neuron marked in red in the middle of Fig. 3.5D from or to other hidden neurons, excluding those that were removed at lesion *2* and not allowed to regrow). The first three PCA components of this 35-dimensional parameter vector are shown. The vector $\boldsymbol{\theta}'$ fluctuates first within one region of the parameter space while probing different solutions to the learning problem, e.g., high probability regions of the posterior distribution (blue trace). Each lesions induced a fast switch to a different region (red,green), accompanied by a recovery of the visual stimulus reconstruction performance (see Fig. 3.5D).

The random fluctuations were found to be an integral part of the fast recovery form lesions. In Sec. C.7 we analyzed the impact of the diffusion term in (3.11) on the learning speed. We found that it acts as a temperature parameter that allows to scale the speed of exploration in the parameter space (see also Appendix C for a detailed derivation).

Altogether this experiment showed that continuously ongoing synaptic sampling maintains stable network function also in a more complex network architecture. Another consequence of synaptic sampling was that the neural codes (assembly sequences) that emerged for the two digit classes within the hidden neurons $\mathbf{z}_A$ and $\mathbf{z}_V$ (see Fig. S2B in Sec. C.7) drifted over larger periods of time (also in the absence of lesions), similarly as observed for place cells in (Y. Ziv et al., 2013) and for tuning curves of motor cortex neurons in (Rokni et al., 2007).

3.7 Discussion

We have shown that stochasticity may provide an important function for network plasticity. It enables networks to sample parameters from some low-dimensional manifold in a high-dimensional parameter space that represents attractive combinations of structural constraints and rules (such as sparse connectivity and heavy-tailed distributions of synaptic weights) and a good fit to empirical evidence (e.g., sensory inputs). We have developed a normative model for this new learning perspective, where the traditional gold standard of maximum likelihood optimization is replaced by theoretically optimal sampling from a posterior distribution of parameter settings, where regions of high probability provide a theoretically optimal model for the low-dimensional manifold from which parameter settings should be sampled. The postulate that networks should learn such posterior distributions of parameters, rather than maximum likelihood values, had been proposed already for quite some while for artificial neural networks (MacKay, 1992; Bishop, 2006), since such organization of learning promises better generalization capability to new examples. The open problem how such posterior distributions could be learned by networks of neurons in the brain, in a way that is consistent with experimental data, has been highlighted in (Pouget et al., 2013) as a key challenge for computational neuroscience. We have presented here such a model, whose primary innovation is to view experimentally found trial-to-trial variability and ongoing fluctuations of parameters such as spine volumes no longer as a nuisance, but as a functionally important component of the organization of network learning, since it enables sampling from a distribution of network configurations. The mathematical framework that we have presented provides a normative model for evaluating such empirically found stochastic dynamics of network parameters, and for relating specific properties of this "noise" to functional aspects of network learning.

Reports of trial-to-trial variability and ongoing fluctuations of parameters related to synaptic weights are ubiquitous in experimental studies of synaptic plasticity and its molecular implementation, from fluctuations of proteins such as PSD-95 (Gray et al., 2006) in the postsynaptic density that are thought to be related to synaptic strength, over intrinsic fluctuations in spine volumes and synaptic connections (Yasumatsu et al., 2008; A. J. Holtmaat et al., 2005; Stettler et al., 2006; Yamahachi et al., 2009; A. Holtmaat and Svoboda, 2009; Loewenstein et al., 2011; Loewenstein et al., 2015), to surprising shifts of neural codes on a larger time scale (Rokni et al., 2007; Y. Ziv et al., 2013). These fluctuations may have numerous causes, from noise in the external environment over noise and fluctuations of internal states in sensory neurons and brain networks, to noise in the pre- and postsynaptic molecular machinery that implements changes in synaptic efficacies on various time scales (Ribrault et al., 2011). One might even hypothesize, that it would be very hard for this molecular machinery to implement synaptic weights that remain constant in the absence of learning, and deterministic rules for synaptic plasticity, because the half-life of many key proteins that are involved is relatively short, and receptors and other membrane-bound proteins are subject to Brownian motion.

In this context the finding that neural codes shift over time (Rokni et al., 2007; Y. Ziv et al., 2013) appears to be less surprising. In fact, our model predicts (see Appendix C) that also stereotypical assembly sequences that emerge in our model through learning, similarly as in the experimental data of (C. D. Harvey et al., 2012), are subject to such shifts on a larger time scale. However it should be pointed out that our model is agnostic with regard to the time scale on which these changes occur, since this time scale can be defined arbitrarily through the parameter b (learning rate) in Eq. (3.3).

The model that we have presented makes no assumptions about the actual sources of noise. It only assumes that salient network parameters are subject to stochastic processes, that are qualitatively similar to those which have been studied and modeled in the context of Brownian motion of particles as random walk on the microscale. One can scale the influence of these stochastic forces in the model by a parameter T that regulates the "temperature" of the stochastic dynamics of network parameters $\boldsymbol{\theta}$. This parameter T regulates the tradeoff between trying out different regions (or modes) of the posterior distribution of $\boldsymbol{\theta}$ (exploration), and staying for longer time periods in a high probability region of the posterior (exploitation). We conjecture that this parameter T varies in the brain between different brain regions, and possibly also between different types of synaptic connections within a cortical column. For example, spine turnover is increased for large values of T, and network parameters $\boldsymbol{\theta}$ can move faster to a new peak in the posterior distribution, thereby supporting faster learning (and faster forgetting). Since spine turnover is reported to be higher in the hippocampus than in the cortex (Attardo et al., 2015), such higher value of T could for example be more adequate for modeling network plasticity in the hippocampus. This model would then also support the hypothesis of (Attardo et al., 2015) that memories are more transient in the hippocampus. In addition T is likely to be regulated on a larger time scale by developmental processes, and on a shorter time scale by neuromodulators and attentional control. The view that synaptic plasticity is stochastic had already been explored through simulation studies in (Rokni et al., 2007; Ajemian et al., 2013). Artificial neural networks were trained in (Ajemian et al., 2013) through supervised learning with high learning rates and high amounts of noise both on neuron outputs and synaptic weight changes. The authors explored the influence of various combinations of noise levels and learning rates on the success of learning, which can be understood as varying the temperature parameters T in the synaptic sampling framework. In order to measure this parameter T experimentally in a direct manner, one would have to apply repeatedly the same plasticity induction protocol to the same synapse, with a complete reset of the internal state of the synapse between trials, and measure the resulting trial-to-trial variability of changes of its synaptic efficacy. Since such complete reset of a synaptic state appears to be impossible at present, one can only try to approximate it by the variability that can be measured between different instances of the same type of synaptic connection.

We have shown that the Fokker-Planck equation, a standard tool in physics for analyzing the temporal evolution of the spatial probability density function for

particles under Brownian motion, can be used to create bridges between details of local stochastic plasticity processes on the microscale and the probability distribution of the vector $\boldsymbol{\theta}$ of all parameters on the network level. This theoretical result provides the basis for the new theory of network plasticity that we are proposing. In particular, this link allows us to derive rules for synaptic plasticity which enable the network to learn, and represent in a stochastic manner, a desirable posterior distribution of network parameters; in other words: to approximate Bayesian inference.

We find that resulting rules for synaptic plasticity contain the familiar term for maximum likelihood learning. But another new term, apart from the Brownian-motion-like stochastic term, is the term $\frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\theta_i)$ that results from a prior distributions $p_{\mathcal{S}}(\theta_i)$, which could actually be different for each biological parameter θ_i and enforce structural requirements and preferences of networks of neurons in the brain. Some systematic dependencies of changes in synaptic weights (for the same pairing of pre- and postsynaptic activity) on their current values had already been reported in (Liao et al., 1992; Bi and Poo, 1998; Sjöström et al., 2001; Montgomery et al., 2001). These can be modeled as impact of priors. Other potential functional benefits of priors (on emergent selectivity of neurons) have recently been demonstrated in (Xiong et al., 2014) for a restricted Boltzmann machine. An interesting open question is whether the non-local learning rules of their model can be approximated through biologically more realistic local plasticity rules, e.g. through synaptic sampling. We have also demonstrated in Fig. 3.3 and Fig. 3.4 that suitable priors can model experimental data from (Loewenstein et al., 2015) and (Yang et al., 2009) on the survival statistics of dendritic spines. The transient behavior of synaptic turnover in our model fits a two-term exponential function, the long-term (stationary) behavior is well described by a power-law. Both findings are in accordance with experimental data.

The results reported in (Fiser et al., 2010) suggest that learned neural representations integrate experience with a priori beliefs about the sensory environment. The model presented here could be used to further investigate this hypothesis. Also the Fokker-Planck formalism was previously applied to describe the dynamics of dendritic spines in hippocampus (O'Donnell et al., 2011). The methods described there to integrate experimental data into computational models could be combined with the synaptic sampling framework to further improve the fit to biology.

Finally, we have demonstrated in Fig. 3.4 and 3.5 that suitable priors for network parameters θ_i that model spine volumes endow a neural network with the capability to respond to changes in the input distribution and network perturbations with a network rewiring that maintains or restores the network function, while simultaneously observing structural constraints such as sparse connectivity.

Our model underlines the importance of further experimental investigation of priors for network parameters. How are they implemented on a molecular level? What role does gene regulation have in their implementation? How does the history of a synapse affect its prior? In particular, can consolidation of a synaptic weight θ_i

be modeled in an adequate manner as a modification of its prior? This would be attractive from a functional perspective, because according to our model it both allows long-term storage of learned information and flexible network responses to subsequent perturbations.

Besides the use of parameter priors, dropout (G. E. Hinton et al., 2012) and dropconnect (Wan et al., 2013) can be used to avoid overfitting in artificial neural networks. In particular, dropconnect, which drops randomly chosen synaptic connections during training, is reminiscent of stochastic synaptic release in biological neuronal networks. In synaptic sampling, synaptic parameters are assumed to be stochastic, however, this stochastic dynamics evolves on a much slower time scale than stochastic release, which was not modeled in our simulations. An interesting open question is whether synaptic sampling combined with stochastic synaptic release would further improve generalization capabilities of spiking neural networks in a similar manner as dropconnect for artificial neural networks.

We have focused in the examples for our model on the plasticity of synaptic weights and synaptic connections. But the synaptic sampling framework can also be used for studying the plasticity of other synaptic parameters, e.g., parameters that control the short term dynamics of synapses, i.e., their individual mixture of short term facilitation and depression. The corresponding parameters U, D, F of the models from (Varela et al., 1997; Markram et al., 1998) are known to depend in a systematic manner on the type of pre- and postsynaptic neuron (Markram et al., 2004), indicative of a corresponding prior. However also a substantial variability within the same type of synaptic connections, had been found (Markram et al., 2004). Hence it would be interesting to investigate functional properties and experimentally testable consequences of stochastic plasticity rules of type (3.5) for U, D, F, and to compare the results with those of previously considered deterministic plasticity rules for U, D, F (see e.g., (Natschlaeger et al., 2001)).

Early theoretical work on activity-dependent formation and elimination of synapses has been used to model ocular dominance in the visual cortex (Elliott and N. R. Shadbolt, 1998; Elliott and N. Shadbolt, 1998). Theoretical models for structural plasticity have also shown that simple plasticity models combined with mechanisms for rewiring are able to model cortical reorganization after lesions (Butz and Ooyen, 2013; Butz et al., 2014). In (Deger et al., 2012) a model was presented that combines structural plasticity and STDP. This model was able to reproduce the existence of transient and persistent spines in the cortex. A recently introduced probabilistic model of structural plasticity was also able to reproduced the statistics of the number of synaptic connections between pairs of neurons in the cortex (Fauth et al., 2015). Furthermore a simple model of structural synaptic plasticity has been introduced that was able to explain cognitive phenomena such as graded amnesia and catastrophic forgetting (Knoblauch et al., 2014). In contrast to these previous studies, the goal of the current work was to establish a model of structural plasticity that follows from a first functional principle, that is, sampling from the posterior distribution over parameters.

We have demonstrated that this framework provides a new and principled way of modeling structural plasticity (May, 2011; Caroni et al., 2012). The challenge to find a biologically plausible way of modeling structural plasticity as Bayesian inference has been highlighted by (Pouget et al., 2013). In addition, the proposed framework does not treat rewiring and synaptic plasticity separately, but provides a unified theory for both phenomena, that can be directly related to functional aspects of the network via the resulting posterior distribution. We have shown in Fig. 3.3 and 3.4 that this rule produces a population of persistent synapses that remain stable over long periods of time, and another population of transient synaptic connections which disappear and reappear randomly, thereby supporting automatic adaptation of the network structure to changes in the distribution of external inputs (Fig. 3.4) and network perturbation (Fig. 3.5).

On a more general level we propose that a framework for network plasticity where network parameters are sampled continuously from a posterior distribution will automatically be less brittle than previously considered maximum likelihood learning frameworks. The latter require some intelligent supervisor who recognizes that the solution given by the current parameter vector is no longer useful, and induces the network to resume plasticity. In contrast, plasticity processes remain active all the time in our sampling-based framework. Hence network compensation for external or internal perturbations is automatic and inherent in the organization of network plasticity.

The need to rethink observed parameter values and plasticity processes in biological networks of neurons in a way which takes into account their astounding variability and compensation capabilities has been emphasized by Eve Marder (see e.g. (Prinz et al., 2004; Marder and Goaillard, 2006; Marder, 2011)) and others. This article has introduced a new conceptual and mathematical framework for network plasticity that promises to provide a foundation for such rethinking of network plasticity.

Chapter 4

Reward-based self-configuration of neural circuits

Contents

4.1 Introduction 70
4.2 Synaptic sampling for reward-based synaptic plasticity and rewiring 71
4.3 Reward-based rewiring and synaptic plasticity as Bayesian policy sampling 77
4.4 Reward-based learning of task-dependent routing of information 79
4.5 Bayesian perspective on policy sampling 82
4.6 A model for task-dependent self-configuration of a recurrent network of spiking neurons 83
4.7 Compensation for network perturbations 87
4.8 Discussion 91

Abstract. Synaptic connections between neurons in the brain are dynamic because of continuously ongoing spine dynamics, axonal sprouting, and other processes. In fact, it was recently shown that the spontaneous synapse-autonomous component of spine dynamics is at least as large as the component that depends on the history of pre- and postsynaptic neural activity. These data are inconsistent with common models for network plasticity, and raise the questions how neural circuits can maintain a stable computational function in spite of these continuously ongoing processes, and what functional uses these ongoing processes might have. We show that spontaneous synapse-autonomous processes, in combination with reward signals such as dopamine, can explain the capability of networks of neurons in the brain to configure themselves for specific computational tasks, and to compensate automatically for later changes in the network or task. Furthermore we show theoretically and through computer simulations that stable computational performance is compatible with continuously ongoing synapse-autonomous changes. After reaching good computational performance it causes primarily a slow drift of network architecture and dynamics in task-irrelevant dimensions, as observed for neural activity in motor cortex and other areas. On the more abstract level of reinforcement learning the resulting model gives rise to an understanding of reward-driven network plasticity as Bayesian policy sampling.

Acknowledgments and author contributions. This chapter is based on the manuscript

> David Kappel, Robert Legenstein, Stefan Habenschuss, Michael Hsieh, Wolfgang Maass (2016). "Reward-based self-configuration of neural circuits." *(under review)*.

To this study, I contributed as joint first author together with RL. The study was conceived by DK, RL, SH, and WM, with the theory being developed by DK, RL and SH. The experiments were designed by DK, RL and WM, and were conducted by DK. The manuscript was written by DK, RL and WM. The authors thank Rodney Douglas, Guillaume Bellec, Anand Subramoney and Jian Liu for helpful comments to the manuscript.

4.1 Introduction

The connectome is dynamic: Networks of neurons in the brain rewire themselves on a time scale of hours to days (A. J. Holtmaat et al., 2005; Stettler et al., 2006; Yang et al., 2009; A. Holtmaat and Svoboda, 2009; N. E. Ziv and Ahissar, 2009; Minerbi et al., 2009; Kasai et al., 2010; Loewenstein et al., 2011; Loewenstein et al., 2015; Rumpel and Triesch, 2016; Chambers and Rumpel, 2017; Ooyen and Butz-Ostendorf, 2017). This rewiring is to a large extent driven by the growth and shrinking of dendritic spines, which is known to take place even in the absence of neural activity (Yasumatsu et al., 2008). The recent study of (Dvorkin and N. E. Ziv, 2016), which includes in Fig. 8 a reanalysis of mouse brain data from (Kasthuri et al., 2015), showed that this spontaneous component is surprisingly large, at least as large as the impact of pre- and postsynaptic neural activity.

Other experimental data show that not only the connectome, but also the dynamics and function of neural circuits is subject to continuously ongoing changes. Continuously ongoing drifts of neural codes were reported in (Y. Ziv et al., 2013; Driscoll and C. Harvey, 2016). Further data show that the mapping of inputs to outputs by neural networks that plan and control motor behavior are subject to a random walk on a slow time-scale of minutes to days, that is conjectured to be related to stochastic synaptic rewiring and plasticity (Beers et al., 2013; Chaisanguanthum et al., 2014).

We address two questions that are raised by these data:

i) How can stable network performance be achieved in spite of the experimentally found continuously ongoing rewiring and activity-independent synaptic plasticity in neural circuits?
ii) What could be a functional role of these processes?

Similar as (Statman et al., 2014; Loewenstein et al., 2015) we model spontaneous synapse-autonomous spine dynamics of each potential synaptic connection i

through a stochastic process that modulates a corresponding parameter θ_i. One can then describe the network configuration, i.e., the current state of the dynamic connectome and the strengths of all currently functional synapses, at any time point by a vector $\boldsymbol{\theta}$ that encodes the current values θ_i for all potential synaptic connections i. The stochastic dynamics of this high-dimensional vector $\boldsymbol{\theta}$ defines a Markov chain whose stationary distribution (illustrated in Fig. 4.1D) provides insight into questions that address the relation between properties of local synaptic processes and the computational function of a neural network.

We propose the following answer to question i): As long as most of the mass of this stationary distribution lies in regions or low-dimensional manifolds of the parameter space that produce good performance, stable network performance can be assured in spite of continuously ongoing movement of $\boldsymbol{\theta}$. Fig. 4.1F, Fig. 4.2I, and Fig. 4.4G suggest that when a computational task has been learnt, most of the subsequent dynamics of $\boldsymbol{\theta}$ takes place in task-irrelevant dimensions, such as the axis along the ridge of the stationary distribution of Fig. 4.1D.

The same model also provides an answer to question ii): Stochastic dynamics of the parameter vector $\boldsymbol{\theta}$ enables the network not only to find in a high-dimensional space regions with good network performance, but also to compensate immediately and automatically for changes in the network or task. We analyze how the strength of the stochastic component of synaptic plasticity affects this compensation capability, and arrive at the conclusion that compensation works best if it is as large as in experimental data (Dvorkin and N. E. Ziv, 2016).

On the more abstract level of reinforcement learning, our theoretical framework for reward-driven network plasticity suggests a new algorithmic paradigm for network learning: Bayesian policy sampling. Compared with the familiar policy gradient learning (Williams, 1992; Baxter and Bartlett, 2000; J. Peters and Schaal, 2006) this paradigm is more consistent with experimental data that suggest a continuously ongoing drift of network parameters.

The resulting model for reward-gated network plasticity builds on the approach from (Kappel et al., 2015a) for unsupervised learning, that was only applicable to a specific neuron model and a specific STDP-rule. Since the new approach can be applied to arbitrary neuron models, in particular also to large data-based models of neural circuits and systems, it can be used to explore how data-based models for neural circuits and brain areas can attain and maintain a computational function.

4.2 Synaptic sampling for reward-based synaptic plasticity and rewiring

We first address the design of a suitable theoretical framework for investigating the self-organization of neural circuits for specific computational tasks in the presence

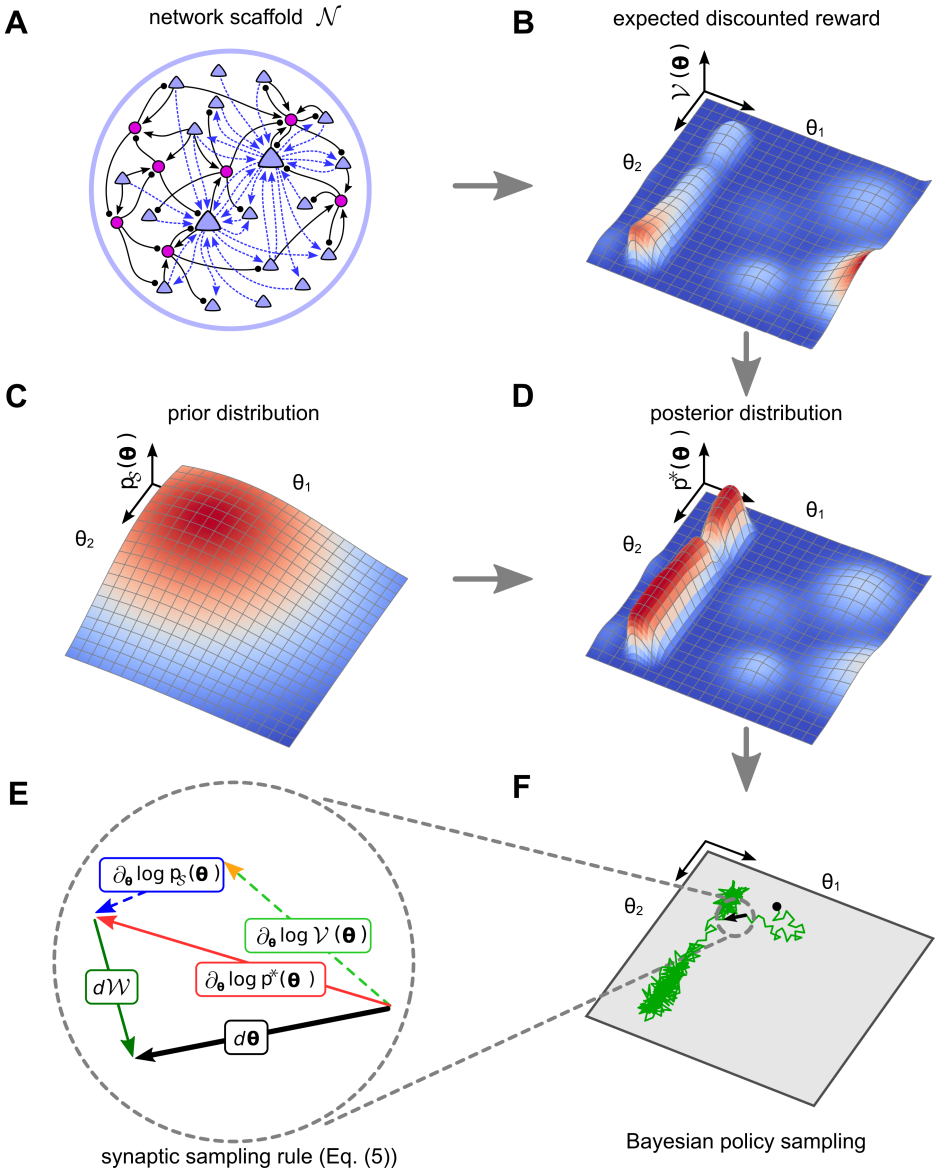

Fig. 4.1: Illustration of the theoretical framework. A: A neural network scaffold $\mathcal{N}$ of excitatory (blue triangles) and inhibitory (purple circles) neurons. Potential synaptic connections (broken blue arrows) of only two excitatory neurons are shown to keep the figure uncluttered. Synaptic connections (black connections) from and to inhibitory neurons are assumed to be fixed for simplicity. **B:** A reward landscape for two parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2\}$ with several local optima. **C:** Example prior that prefers small values for θ_1 and θ_2. **D:** The posterior distribution $p^*(\boldsymbol{\theta})$ that results as product of the prior from panel (C) and the expected discounted reward of panel (B). **E:** Illustration of the dynamic forces (plasticity rule Eq. (4.5)) that act on $\boldsymbol{\theta}$ in each sampling step $d\boldsymbol{\theta}$ (black) while sampling from the posterior distribution. The deterministic term (red), which consists of the first two terms (prior and reward expectation) in Eq. (4.5), is directed to the next local maximum of the posterior. The stochastic term $d\mathcal{W}$ (green) of Eq. (4.5) has a random direction. **F:** A single trajectory of Bayesian policy sampling from the posterior distribution of panel (D) under Eq. (4.5), starting at the black dot. The parameter vector $\boldsymbol{\theta}$ fluctuates between different solutions, and moves primarily along the task-irrelevant dimension θ_2.

of spontaneous synapse-autonomous processes and rewards. There exist well-established models for reward-modulated synaptic plasticity, see e.g. (Frémaux et al., 2010), where reward signals gate common rules for synaptic plasticity, such as STDP. But these rules are lacking two components that we need here:

- an integration of rewiring with plasticity rules that govern the modulation of the strengths of already existing synaptic connections
- a term that reflects the spontaneous synapse-autonomous component of synaptic plasticity and rewiring.

In order to illustrate our approach we consider a neural network scaffold, see Fig. 4.1A, with a large number of potential synaptic connections between excitatory neurons. Only a subset of these potential connections is assumed to be functional at any point in time. For simplicity we assume that only excitatory connections are plastic, but the model can be easily extended to also reflect plasticity of inhibitory synapses. For each potential synaptic connection i, we introduce a parameter θ_i that describes its state both for the case when this potential connection i is currently not functional (this is the case when $\theta_i \leq 0$) and when it is functional (i.e., $\theta_i > 0$). More precisely, θ_i encodes the current strength or weight w_i of this synaptic connection through the formula

$$w_i \;=\; \begin{cases} \exp(\theta_i - \theta_0) & \text{if } \theta_i > 0 \qquad (\textit{functional synaptic connection}) \\ 0 & \text{if } \theta_i \leq 0 \qquad (\textit{non-functional potential connection}) \end{cases} , \tag{4.1}$$

with a positive offset parameter θ_0 that regulates the initial strength of new functional synaptic connections (we set $\theta_0 = 3$ in our simulations). The exponential function in Eq. (4.1) turns out to be useful for relating the dynamics of θ_i to experimental data on the dynamics of synaptic weights. The volume – or image brightness in Ca-imaging – of a dendritic spine is commonly assumed to be proportional to the strength w_i of a synapse (A. J. Holtmaat et al., 2005). The logarithm of this estimate for w_i was shown in Fig. 2i of (A. Holtmaat et al., 2006) and also in (Yasumatsu et al., 2008; Loewenstein et al., 2011) to exhibit a dynamics similar to that of an Ornstein-Uhlenbeck process, i.e., a random walk in conjunction with a force that draws the random walk back to its initial state. Hence if θ_i is chosen to be proportional to the logarithm of w_i, it is justified to model the spontaneous dynamics of θ_i as an Ornstein-Uhlenbeck process. This is done in our model, as we will explain after Eq. (4.5) and demonstrate in Fig. 4.2c. The logarithmic transformation also ensures that additive increments of θ_i yield multiplicative updates of w_i, which have been observed experimentally (Loewenstein et al., 2011).

Altogether our model needs to create a dynamics for θ_i that is not only consistent with experimental data on spontaneous spine dynamics, but is for the case $\theta_i > 0$ also consistent with rules for reward-modulated synaptic plasticity as in (Frémaux et al., 2010). This suggests to look for plasticity rules of the form

$$d\theta_i \;=\; \beta \;\times\; (\text{deterministic plasticity rule}) \times dt \;+\; \sqrt{2\beta T}\, d\mathcal{W}_i \,, \tag{4.2}$$

where the deterministic plasticity rule could for example be a standard reward-based plasticity rule. We will argue below that it makes sense to include also a prior in this deterministic component of rule (4.2), both for functional reasons and in order to fit data on spontaneous spine dynamics. The stochastic term $d\mathcal{W}_i$ in Eq. (4.2) is an infinitesimal step of a random walk, more precisely for a Wiener process $\mathcal{W}_i$. A Wiener process is a standard model for Brownian motion in one dimension (Gardiner, 2004). The term $\sqrt{2\beta T}$ scales the strength of this stochastic component in terms of a "temperature" T and a learning rate β, and is chosen to be of a form that supports analogies to statistical physics. The presence of this stochastic term makes it unrealistic to expect that θ_i converges to a particular value under the dynamics defined by Eq. (4.2). In fact, in contrast to many standard differential equations, the stochastic differential equation or SDE (4.2) does not have a single trajectory of θ_i as solution, but an infinite family of trajectories that result from different random walks.

We propose to focus – instead of the common analysis of the convergence of weights to specific values as invariants – on the most prominent invariant that a stochastic process can offer: the longterm stationary distribution of synaptic connections and weights. The stationary distribution of the vector $\boldsymbol{\theta}$ of all synaptic parameters θ_i informs us about the statistics of the infinitely many different solutions of a stochastic differential equation of the form (4.2). In particular, it informs us about the fraction of time at which particular values of $\boldsymbol{\theta}$ will be visited by these solutions. We show that a large class of reward-based plasticity rules produce in the context of an equation of the form (4.2) a stationary distribution of $\boldsymbol{\theta}$ that can be clearly related to reward expectation for the neural network, and hence to its computational function.

More precisely, if one allows rewiring then the concept of a neural network becomes problematic, since the definition of a neural network typically includes its synaptic connections. Hence we refer to the set of neurons of a network, its set of potential synaptic connections, and its set of definite synaptic connections – such as in our case connections from and to inhibitory neurons (see Fig. 4.1A) – as a *network scaffold*. A network scaffold $\mathcal{N}$ together with a parameter vector $\boldsymbol{\theta}$ that specifies a particular selection of functional synaptic connections out of the set of potential connections and particular synaptic weights for these defines a concrete neural network, to which we also refer as *network configuration*.

We want to address the question which reward-based plasticity rules achieve in the context with other terms in Eq. (4.2) that the resulting stationary distribution of network configurations has most of its mass on highly rewarded network configurations. A key observation is that if the first term on the right-hand-side of (4.2) can be written for all potential synaptic connections i in the form $\frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta})$, where $p^*(\boldsymbol{\theta})$ is some arbitrary given distribution and $\frac{\partial}{\partial \theta_i}$ denotes the partial derivative with respect to parameter θ_i, then these stochastic processes

$$d\theta_i \;=\; \beta \, \frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta}) \, dt \;+\; \sqrt{2\beta T} \, d\mathcal{W}_i \; . \tag{4.3}$$

give rise to a stationary distribution that is proportional to $p^*(\boldsymbol{\theta})^{\frac{1}{T}}$. Hence, a rule for reward-based synaptic plasticity that can be written in the form $\frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta})$, where $p^*(\boldsymbol{\theta})$ has most of its mass on highly rewarded network configurations $\boldsymbol{\theta}$, achieves that the network will spend most of its time in highly rewarded network configurations. This will hold even if the network does not converge to or stay in any particular network configuration $\boldsymbol{\theta}$ (see Fig. 4.1D,F for an illustration). Furthermore the role of the temperature T in (4.3) becomes clearly visible in this result: if T is large the resulting stationary distribution flattens the distribution $p^*(\boldsymbol{\theta})$, whereas for $0 < T < 1$ the network will remain for larger fractions of the time in those regions of the parameter space where $p^*(\boldsymbol{\theta})$ achieves its largest values. In fact, if the temperature T converges to 0, the resulting stationary distribution degenerates to one that has all of its mass on the network configuration $\boldsymbol{\theta}$ for which $p^*(\boldsymbol{\theta})$ reaches its global maximum, as in simulated annealing (Kirkpatrick and Vecchi, 1983).

We will focus on target distributions $p^*(\boldsymbol{\theta})$ of the form

$$p^*(\boldsymbol{\theta}) \ \propto \ p_S(\boldsymbol{\theta}) \ \times \ \mathcal{V}(\boldsymbol{\theta}) \ , \tag{4.4}$$

where $\propto$ denotes proportionality up to a positive normalizing constant. $p_S(\boldsymbol{\theta})$ can encode structural priors of the network scaffold $\mathcal{N}$. For example, it can encode a preference for sparsely connected networks. This happens when $p_S(\boldsymbol{\theta})$ has most of its mass near $\mathbf{0}$, see Fig. 4.1C for an illustration. But it could also convey genetically encoded or previously learnt information, such as a preference for having strong synaptic connections between two specific populations of neurons. The term $\mathcal{V}(\boldsymbol{\theta})$ in Eq. (4.4) denotes the expected discounted reward associated with a given parameter vector $\boldsymbol{\theta}$ (see Fig. 4.1B). Eq. (4.3) for the stochastic dynamics of parameters takes then the form

$$d\theta_i \ = \ \beta \left(\frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}} \left(\boldsymbol{\theta} \right) \ + \ \frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta}) \right) dt \ + \ \sqrt{2\beta T} \, d\mathcal{W}_i \ . \tag{4.5}$$

When the term $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ vanishes, this equation models spontaneous spine dynamics. We will make sure that this term vanishes for all potential synaptic connections i that are currently not functional, i.e., where $\theta_i \leq 0$. If one chooses a Gaussian distribution as prior $p_{\mathcal{S}} \left(\boldsymbol{\theta} \right)$, the dynamics of (4.5) amounts in the case $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta}) = 0$ to an Ornstein-Uhlenbeck process. This process was previously already proposed as simple model for experimentally observed spontaneous spine dynamics (Loewenstein et al., 2011; Loewenstein et al., 2015; Statman et al., 2014). We use in our simulations for the prior $p_{\mathcal{S}} \left(\boldsymbol{\theta} \right)$ a Gaussian distribution that prefers small but nonzero weights. Hence our model (4.5) is consistent with previous models for spontaneous spine dynamics.

Thus altogether we arrive at a model for the interaction of stochastic spine dynamics with reward where the usually considered deterministic convergence to network configurations $\boldsymbol{\theta}$ that represent local maxima of expected reward $\mathcal{V}(\boldsymbol{\theta})$ (e.g. to the local maxima in Fig. 4.1B) is replaced by a stochastic model. If the stochastic dynamics of $\boldsymbol{\theta}$ is defined by local stochastic processes of the form (4.5), as indicated

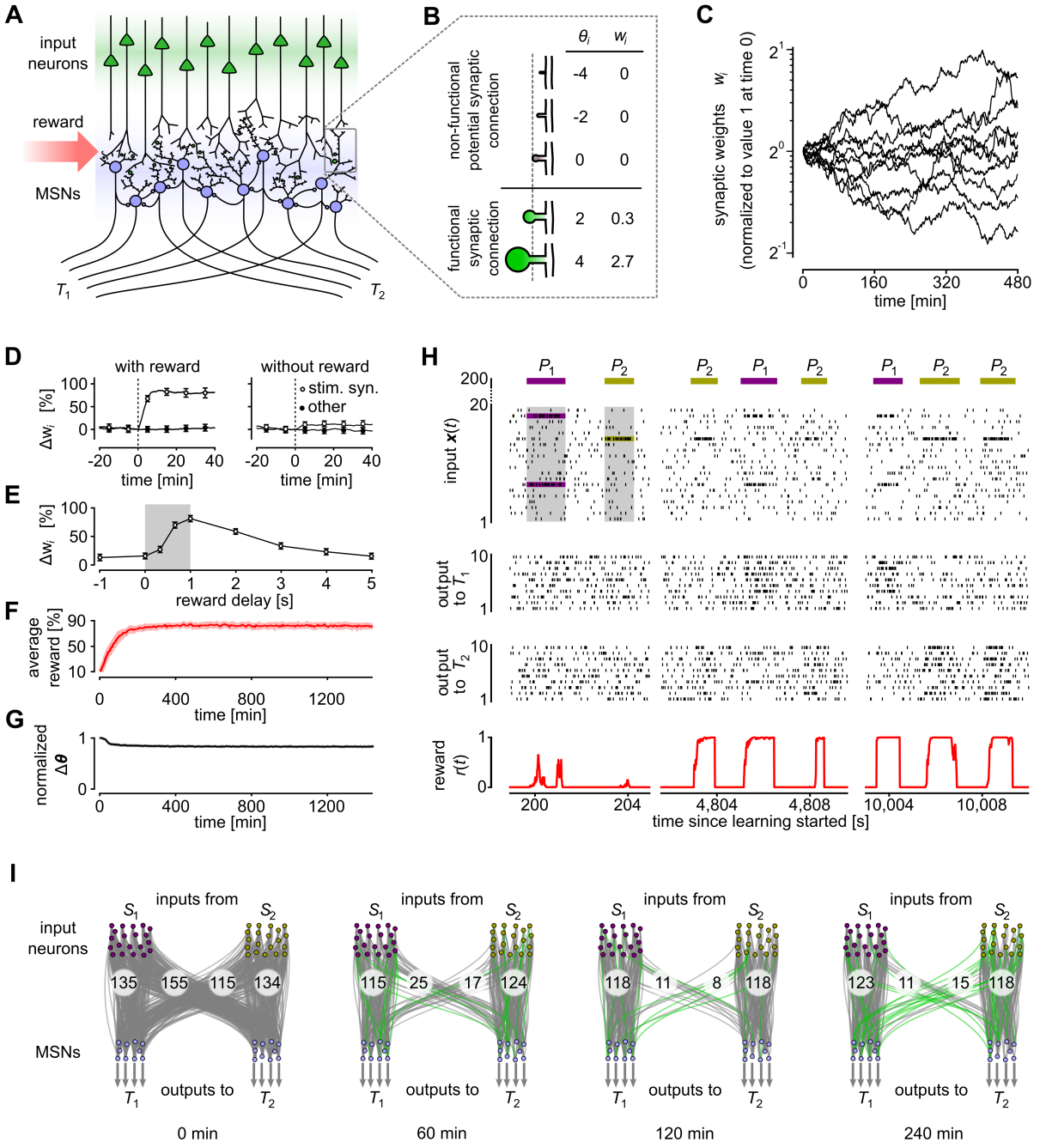

Fig. 4.2: Reward-based routing of input patterns. A: Illustration of the network scaffold. A population of 20 model MSNs (blue) receives input from 200 excitatory input neurons (green) that model cortical neurons. Potential synaptic connections between these 2 populations of neurons were subject to reward-based synaptic sampling. In addition, fixed lateral connections provided recurrent inhibitory input to the MSNs. The MSNs were divided into two groups, each projecting exclusively to one of two target areas T_1 and T_2. Reward was delivered whenever the network managed to route an input pattern P_i primarily to that group of MSNs that projected to target area T_i. **B:** Illustration of the model for spine dynamics. Five potential synaptic connections at different states are shown. Synaptic spines are represented by circular volumes with diameters proportional to $\sqrt[3]{w_i}$ for functional connections, assuming a linear correlation between spine-head volume and synaptic efficacy w_i (Matsuzaki et al., 2001). **C:** Dynamics of weights w_i in log-scale for 10 potential synaptic connections i when the activity-dependent term $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta}) dt \quad \rightarrow$

→ in Eq. (4.5) is set equal to zero. Consistent with experimental date (see e.g. Fig. 2i of (A. Holtmaat et al., 2006)) the dynamics is in this case consistent with an Ornstein-Uhlenbeck process in the logarithmic scale. Weight values are plotted relative to the initial value at time 0. **D, E:** Dynamics of a model synapse when a reward-modulated STDP pairing protocol as in (Yagishita et al., 2014) was applied. **D:** Reward delivery after repeated firing of the presynaptic neuron before the postsynaptic neuron resulted in a strong weight increase (left). This effect was reduced without reward (right), and prevented completely if no presynaptic stimulus was applied. Values in (D) and (E) represent percentage of weight changes relative the pairing onset time (dashed line, means and s.e.m. over 50 synapses). Compare with Fig. 1F,G in (Yagishita et al., 2014). **E:** Dependence of resulting changes in synaptic weights in our model as a function of the delay of reward delivery. Gray shaded rectangle indicates the time window of STDP pairing application. Reward delays denote time between paring and reward onset. Compare to Figure 1O in (Yagishita et al., 2014). **F:** The average reward achieved by the network increased quickly during learning according to Eq. (4.5) (mean over 5 independent trial runs; shaded area indicates s.e.m.). **G:** Synaptic parameters kept changing throughout the experiment in (F). The magnitude of the change of the synaptic parameter vector $\boldsymbol{\theta}$ is shown (mean $\pm$ s.e.m. as in (F); Euclidean norm, normalized to the maximum value). The parameter change peaks at the onset of learning, but remains high (larger than 80% of the maximum value) even when stable performance has been reached. **H:** Spiking activity of the network during learning. Activities of 20 randomly selected input neurons and all MSNs are shown. 3 salient input neurons (belonging to pools S_1 or S_2 in (I)) are highlighted. Most neurons have learnt to fire at a higher rate for the input pattern P_j that corresponds to the target area T_j to which they are projecting. Bottom: reward delivered to the network. **I:** Dynamics of network rewiring throughout learning. Snapshots of network configurations for the times t indicated below the plots are shown. Gray lines indicate active connections between neurons; connections that were not present at the preceding snapshot are highlighted in green. All output neurons and two subsets of input neurons that fire strongly in pattern P_1 or P_2 are shown (pools S_1 and S_2, 20 neurons each). Numbers denote total counts of functional connections between pools. The connectivity was initially dense and then rapidly restructured and became sparser. Rewiring took place all the time throughout learning.

in Fig. 4.1E, the resulting stochastic model for network plasticity will spend most of its time in network configurations $\boldsymbol{\theta}$ where the posterior $p^*(\boldsymbol{\theta})$, illustrated in Fig. 4.1D, approximately reaches its maximal value. This provides on the statistical level a guarantee of stable network function, in spite of ongoing stochastic dynamics of all the parameters θ_i.

4.3 Reward-based rewiring and synaptic plasticity as Bayesian policy sampling

We assume that the network scaffold $\mathcal{N}$ receives reward signals $r(t)$ at certain times t, corresponding for example to dopamine signals in the brain (see (Collins and M. J. Frank, 2016) for a recent discussion of related experimental data). The expected discounted reward $\mathcal{V}(\boldsymbol{\theta})$ that occurs in the second term of Eq. (4.5) is the integral over all future rewards $r(t)$, while discounting more remote rewards exponentially, see Eq. (D.1) in Appendix D. Fig. 4.1B shows a hypothetical $\mathcal{V}(\boldsymbol{\theta})$-landscape over two parameters θ_1, θ_2. The posterior $p^*(\boldsymbol{\theta})$ shown in Fig. 4.1D is then proportional to the product of $\mathcal{V}(\boldsymbol{\theta})$ (panel b) and the prior (panel c).

The computational behavior of the network configuration, i.e., the mapping of

network inputs to network outputs that is encoded by the parameter vector $\boldsymbol{\theta}$, is referred to as a policy in the context of reinforcement learning theory. When the parameter dynamics is given solely by the second term in the parenthesis of Eq. (4.5), $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$, we recover for the case $\theta_i > 0$ deterministic policy gradient learning (Williams, 1992; Baxter and Bartlett, 2000; J. Peters and Schaal, 2006). These parameters (and therefore the policy) are gradually changed through Eq. (4.5) such that the expected discounted reward $\mathcal{V}(\boldsymbol{\theta})$ is increased: The parameter dynamics follows the gradient of $\log \mathcal{V}(\boldsymbol{\theta})$, i.e., $\frac{d\theta_i}{dt} = \beta \frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$, where $\beta > 0$ is a small learning rate.

For the simulations described below we took a network scaffold $\mathcal{N}$ consisting of spiking neurons (see *Network model* in Appendix D). In this case, the derivative $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ gives rise to synaptic updates at a synapse i that are essentially given by the product of the current reward signal $r(t)$ and an eligibility trace that depends on pre- or postsynaptic firing times, see *Synaptic dynamics for the reward-based synaptic sampling model* in Appendix D. Such plasticity rules have previously been proposed by (Seung, 2003; Xie and Seung, 2004; Izhikevich, 2007; Pfister et al., 2006; Florian, 2007; Legenstein et al., 2008; Urbanczik and Senn, 2009). For non-spiking neural networks, a similar update rule was first introduced by Williams and termed the REINFORCE rule (Williams, 1992). In fact, when one discretizes time and assumes that rewards and parameter updates are only realized at the end of each episode, the REINFORCE rule is recovered.

In contrast to policy gradient, reinforcement learning in the presence of the stochastic last term in Eq. (4.5) cannot converge to any network configuration. Instead, the dynamics of Eq. (4.5) produces continuously changing network configurations, with a preference for configurations that both satisfy constraints from the prior $p_S(\boldsymbol{\theta})$ and provide a large expected reward $\mathcal{V}(\boldsymbol{\theta})$, see Fig. 4.1D,F. Hence this type of reinforcement learning samples continuously from a posterior distribution of network configurations. This is rigorously proven in Theorem 2 of Appendix D. We refer to this reinforcement learning model as *Bayesian policy sampling*, and to the family of reward-based plasticity rules that are defined by Eq. (4.5) as *reward-based synaptic sampling*.

Another key difference to previous models for reward-gated synaptic plasticity and policy gradient learning is, apart from the stochastic last term of Eq. (4.5), that the deterministic first term of Eq. (4.5) also contains a reward-independent component $\frac{\partial}{\partial \theta_i} \log p_S(\boldsymbol{\theta})$ that arises from a prior $p_S(\boldsymbol{\theta})$ for network configurations. In our simulations we consider a simple Gaussian prior $p_S(\boldsymbol{\theta})$ with mean $\mathbf{0}$ that encodes a preference for sparse connectivity (see Eq. (D.12)).

It is important that the dynamics of disconnected synapses, i.e., of synapses i with $\theta_i \leq 0$ or equivalently $w_i = 0$, does not depend on pre- or postsynaptic neural activity since non-functional synapses do not have access to such information. This is automatically achieved through our ansatz $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ for the reward-dependent component in Eq. (4.5), since a simple derivation shows that it entails that the factor w_i appears in front of the term that depends on pre- and postsynaptic activity, see

Eq. (B.18). Instead, the dynamics of θ_i depends for $\theta_i \leq 0$ only on the prior and the stochastic term dW_i. This results in a distribution over waiting times between downwards and upwards crossing of the threshold $\theta_i = 0$ that was found to be similar to the distribution of inter-event times of a Poisson point process, see (Ding and Rangarajan, 2004) for a detailed analysis. This theoretical result suggest a simple approximation of the dynamics of Eq. (4.5) for currently non-functional synaptic connections, where the process (4.5) is suspended whenever θ_i becomes negative, and continued with $\theta_i = 0$ after a waiting time that is drawn from an exponential distribution. As in (Deger et al., 2016) this can be realized by letting a non-functional synapse become functional at any discrete time step with some fixed probability (Poisson process). We have compared in Fig. 4.4C the resulting learning dynamics of the network for this simple approximation with that of the process defined by Eq. (4.5).

4.4 Reward-based learning of task-dependent routing of information

Experimental evidence about gating of spine dynamics by reward signals in the form of dopamine is available for the synaptic connections from the cortex to the entrance stage of the basal ganglia, the medium spiny neurons (MSNs) in the striatum (Yagishita et al., 2014). They report that the volumes of their dendritic spines show significant changes only when pre- and postsynaptic activity is paired with precisely timed delivery of dopamine (see (Yagishita et al., 2014), Fig. 1 E-G, O). More precisely, an STDP pairing protocol followed by dopamine uncaging induced strong LTP in synapses onto MSNs, whereas the same protocol without dopamine uncaging lead only to a minor increase of synaptic efficacies.

MSNs can be viewed as readouts from a large number of cortical areas, that become specialized for particular motor functions, e.g. movements of the hand or leg. We asked whether reward gating of spine dynamics according to the experimental data of (Yagishita et al., 2014) can explain such task dependent specialization of MSNs. More concretely, we asked whether it can achieve that two different distributed activity patterns P_1, P_2 of upstream neurons in the cortex get routed to two different ensembles of MSNs, and thereby to two different downstream targets T_1 and T_2 of these MSNs (see Fig. 4.2A,H,I). We assumed that for each upstream activity pattern P_j a particular subset S_j of upstream neurons is most active, $j = 1, 2$. Hence this routing task amounted to routing synaptic input from S_j to those MSNs that project to downstream neuron T_j.

We applied to all potential synaptic connections i from upstream neurons to MSNs a learning rule according to Eq. (4.5), more precisely, the rule for reward-gated STDP (Eq. (B.18), Eq. (D.11) and Eq. (D.13)) that results from this general framework. The parameters of the model were adapted to qualitatively reproduce the results from Figures 1F,G of (Yagishita et al., 2014) when the same STDP protocol was applied

to our model (see Fig. 4.2D,E). The parameter values are reported in Tab. D.1 in Appendix D. If not stated otherwise, we applied these parameters in all following experiments.

Our simple model consisted of 20 inhibitory model MSNs with lateral recurrent connections. These received excitatory input from 200 input neurons. The synapses from input neurons to model MSNs were subject to our plasticity rule. Multiple connections were allowed between each pair of input neuron and MSN (see Appendix D). The MSNs were randomly divided into two assemblies, each projecting exclusively to one of two downstream target areas T_1 and T_2. Cortical input $\boldsymbol{x}(t)$ was modeled as Poisson spike trains from the 200 input neurons with instantaneous rates defined by two prototype rate patterns P_1 and P_2, see Fig. 4.2H. The task was to learn to activate T_1-projecting neurons and to silence T_2-projecting neurons whenever pattern P_1 was presented as cortical input. For pattern P_2, the activation should be reversed: activate T_2-projecting neurons and silence those projecting to T_1. This desired function was defined through a reward signal $r(t)$ that was proportional to the ratio between the mean firing rate of MSNs projecting to the desired target and that of MSNs projecting to the non-desired target area (see Appendix D).

Fig. 4.2H shows the firing activity and reward signal of the network during segments of one simulation run. After about 80 minutes of simulated biological time, each group of MSNs had learned to increase its firing rate when the activity pattern P_j associated with its projection target T_j was presented. Fig. 4.2F shows the average reward throughout learning. After 3 hours of learning about 82% of the maximum reward was acquired on average, and this level was maintained during prolonged learning.

Fig. 4.2G shows that the parameter vector $\boldsymbol{\theta}$ kept moving at almost the same speed even after a high plateau of rewards had been reached. Hence these ongoing parameter changes took place in dimensions that were irrelevant for the reward-level.

Fig. 4.2I provides snapshots of the underlying "dynamic connectome" (Rumpel and Triesch, 2016) at different points of time. New synaptic connections that were not present at the preceding snapshot are colored green. One sees that the bulk of the connections maintained a solution of the task to route inputs from S_1 to target area T_1 and inputs from S_2 to target area S_2. But the identity of these connections, a task-irrelevant dimension, kept changing. In addition the network always maintained some connections to the currently undesired target area, thereby providing the basis for a swift built-up of these connections if these connections would suddenly also become rewarded. This simulation experiment showed that reward-gated spine dynamics as analyzed in (Yagishita et al., 2014) is sufficiently powerful from the functional perspective to rewire networks so that each signal is delivered to its intended target.

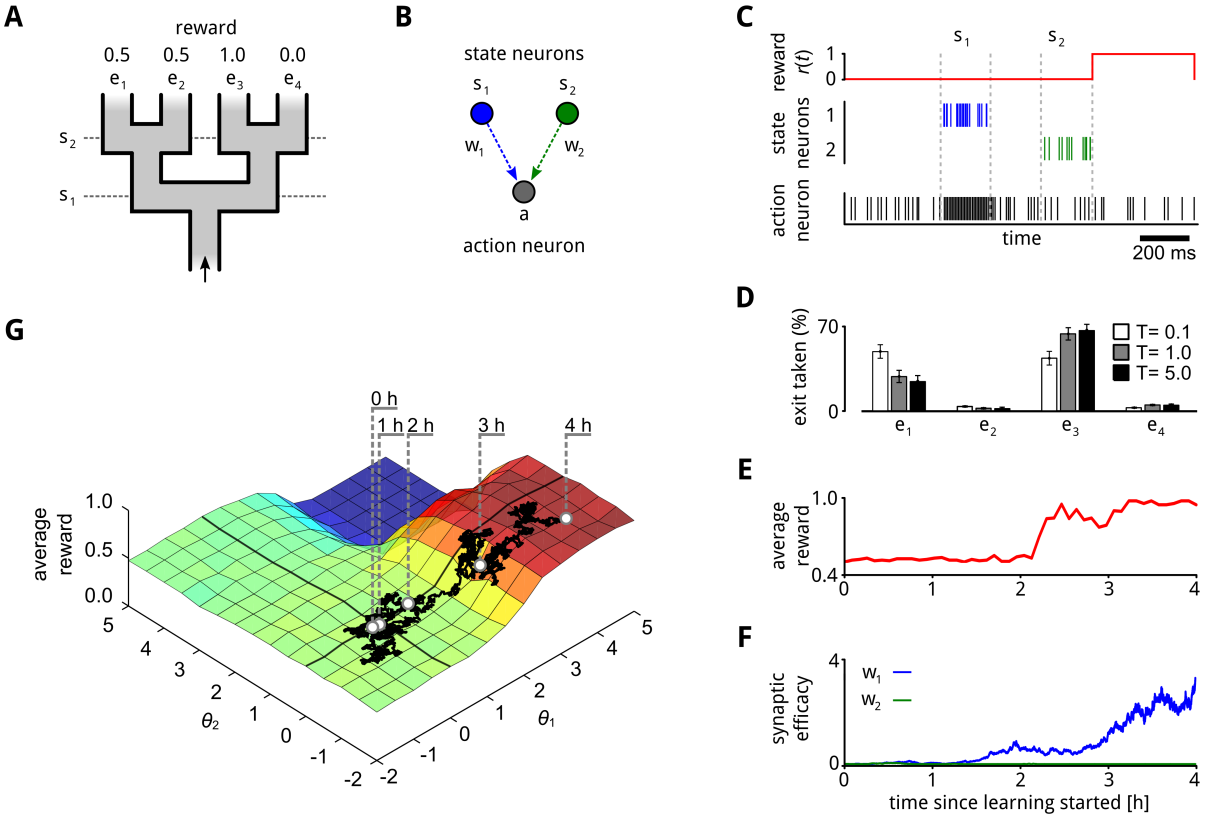

Fig. 4.3: The temperature parameter controls the exploration speed of policy sampling. A: Illustration of the double-T-maze. At the junctions (s_1 and s_2) a left-or-right decision had to be taken to navigate the maze. The arrow marks the entry to the maze. The four exits ($e_1, \ldots, e_4$) were associated with a reward between 0 and 1 (numbers on top). **B:** Illustration of the network that was used to solve this task. Two input neurons encoded the current state (s_1 or s_2) within the task. A single action neuron a encoded the action in each state (go right if the neuron fires above a threshold, else go left). Broken lines with arrows indicate potential synaptic connections. The network had two synaptic parameters θ_1 and θ_2 which encoded the connectivity and synaptic weights w_1 and w_2 between input neurons s_1 and s_2 and the output neuron a, respectively. **C:** A single successful trial run showing the spike train for input neurons and the response of the output neuron. The trace of the reward is shown on top. **D:** Histogram of final states over 100 trial runs for different temperatures. For $T = 0.1$ most trajectories ended in exit e_1. The network was not able to find the optimal policy in most cases. A temperature increase enhanced parameter exploration. For $T = 5$ most state trajectories ended in exit e_3, which corresponds to the optimal policy for this maze. Average results over 200 independent learning experiments are shown, errorbars indicate 95% confidence interval. **E:** Average reward throughout learning for one experiment run. **F:** Example temporal evolution of the synaptic efficacies w_i throughout learning (same run shown in (F)). **G:** Surface plot of average rewards associated to configurations of network parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)$. The trajectory of synaptic parameters throughout learning (corresponding to the efficacies shown in (F)) are projected onto the surface (black trace). The synaptic parameters converged to a setting with high reward.

4.5 Bayesian perspective on policy sampling

We have previously noted in Eq. (4.5) that our model can integrate structural priors $p_{\mathcal{S}}(\boldsymbol{\theta})$ into its continuously ongoing policy sampling. Network parameters $\boldsymbol{\theta}$ are sampled from the posterior distribution $p^*(\boldsymbol{\theta})$ that combines the structural constraints $p_{\mathcal{S}}(\boldsymbol{\theta})$ with the expected discounted reward $\mathcal{V}(\boldsymbol{\theta})$. In this sense the network carries out Bayesian inference over network configurations. One intriguing question, which we pursue in this section, is whether our model can learn to solve a decision making tasks.

Solway and Botvinick have proposed an interesting abstract model for goal-directed decision making through probabilistic inference along with a neural network implementation that could realize this model (Solway and Botvinick, 2012). Their neural network implements an internal model of the environment and predicts reward values associated with configurations of the environment. Specialized subnetworks in their model are constructed to carry out Bayesian inference to decide which actions to take in order to obtain future rewards. The learning problem of how the model is fit to the environment was not addressed in (Solway and Botvinick, 2012). We consider in Fig. 4.3A a decision making task that is equivalent to the one considered in (Solway and Botvinick, 2012) (lower right panel of their Fig. 1). This decision making task is nontrivial because an optimal decision at the first branching point s_1 depends on the planned decision at the second branching level s_2. An action sequence (right, left) is optimal for this maze. This task is solved in the model of (Solway and Botvinick, 2012) by employing a probabilistic generative model for reward, that is implemented in their neural network through a subnetwork that carries out probabilistic inference by approximating belief propagation (see Fig. 8 in (Solway and Botvinick, 2012)). We demonstrate in Fig. 4.3 that both suitable network configurations and parameters of a minimal neural network model for decision making in this maze – see Fig. 4.3b – emerge through reward-based learning in our model. But our learning approach is independent of the precise network architecture – synaptic parameters subject to the dynamics Eq. (4.5) are always attracted to network configurations that lead to high rewards. Random exploration is driven by the temperature parameter T that scales the amplitude of the noise in the synaptic dynamics (Eq. (4.5)). In Fig. 4.4f we have already shown that a brief temperature increase amplifies spine formation. Using this task, we further investigated parameter search strategies employed by the synapses and the role of the temperature in enhancing exploration.

An action neuron a in our network (see Fig. 4.3b) encoded the left-or-right decision that was required in each stage (arbitrarily defined to go right if the neuron was active and go left else). The action neuron received input $\boldsymbol{x}(t)$ from two neurons s_1 and s_2 that encoded the current state of the task. The states corresponded to the two junctions s_1 and s_2 that were visited when traveling the maze. We assumed that each neuron s_i fires only when the corresponding junction in the maze is reached. Therefore the network had only two synaptic parameters θ_1 and θ_2. The action neuron a determined the path through the maze, going right if the firing rate was

above a threshold of 60 Hz and left otherwise. At the end of each trial a reward was delivered by setting $r(t)$ to the reward amplitude associated with the taken exit according to Fig. 4.3a. Fig. 4.3C shows one representative trial run after learning that led to maximal reward (exit e_3).

Fig. 4.3D shows the histogram of the exits taken after learning. The bar plots show average numbers over the last 100 trials of 200 independent learning experiments (see Appendix D). The policy that was learned depended on the temperature parameter T. The optimal policy for this task is to always take exit e_3. Initially both synaptic parameters θ_1 and θ_2 were set close to zero where exit e_1 was taken most of the time. This already led to a reward of 0.5 (see Fig. 4.3G). Exploration is driven by the stochastic nature of the neural network and can be further enhanced by the temperature parameter T. For $T = 0.1$ exploration was mostly driven by the stochastic activity of the network. This randomness was sufficient to find the optimal policy in $43.5 \pm 5.6\%$ of the trials. For a large temperature of $T = 5$, exploration was enhanced and the optimal policy was found in $66.9 \pm 5.0\%$ of the trials. A further increase of the temperature led to a performance decrease.

Fig. 4.3E shows the average reward throughout learning for 4 hours (4800 trials) and Fig. 4.3F,G show the corresponding evolution of the synaptic parameters for one learning experiment. The synaptic parameters were initialized close to zero and then slowly explored the parameter space. After about 3 hours of learning the region of highest reward was found. The average reward that is associated with different parameter settings is shown in Fig. 4.3G. Red corresponds to high reward, blue to low reward (see Appendix D). The synaptic connections randomly retracted and reappeared while the parameter space was explored. The most highly rewarded network configuration is one where the synaptic connection from neuron s_1 to a is strong, and the one from s_2 is retracted. After this configuration was reached a synaptic connection from s_2 to a still randomly reappeared from time to time and then decayed again.

4.6 A model for task-dependent self-configuration of a recurrent network of spiking neurons

We next asked, whether our simple integrated model for reward-modulated rewiring and synaptic plasticity of neural circuits according to Eq. (4.5) could also explain the emergence of specific computations in recurrent networks of spiking neurons. As paradigm for a specific computational task we took a simplified version of the task that mice learned to carry out in the experimental setup of (A. J. Peters et al., 2014). There a reward was given whenever a lever was pressed within a given time window so that it crossed two given thresholds. This task is particular suitable for our context, since spine turnover and changes of network activity were continuously monitored in (A. J. Peters et al., 2014) while the animals learned this task.

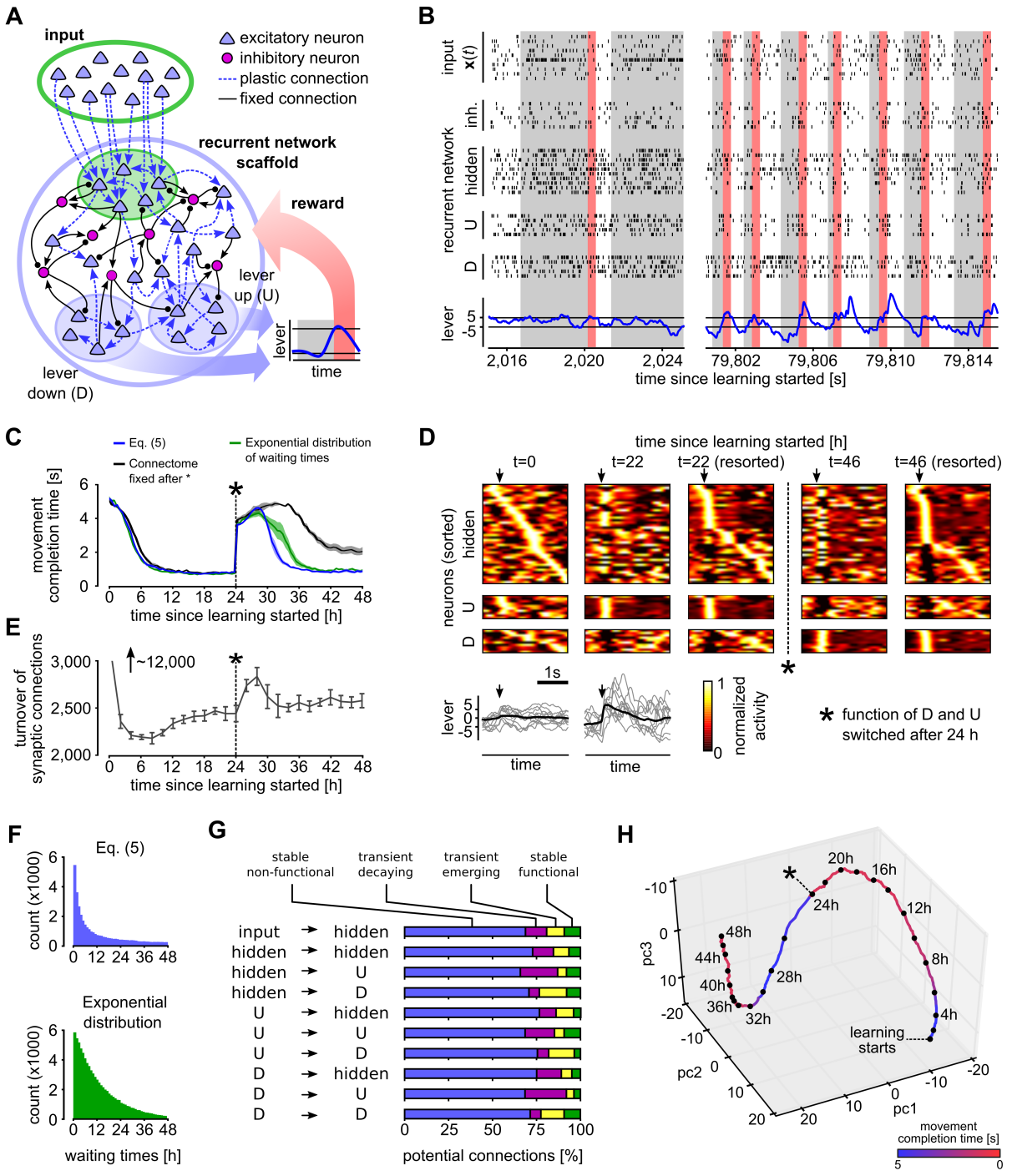

Fig. 4.4: Reward-based self-configuration and compensation capability of a recurrent neural network. A: Network scaffold and task schematic. Symbol convention as in Fig. 4.1A. A recurrent network scaffold of excitatory and inhibitory neurons (large blue circle); a subset of excitatory neurons received input from afferent excitatory neurons (indicated by green shading). *Caption of Fig. 4.4 continued:* From the remaining excitatory neurons, two pools D and U were randomly selected to control lever movement (blue shaded areas).Bottom inset: stereotypical movement that had to be generated to receive a reward. **B:** Spiking activity of the network at learning onset and after 22 hours of learning. Activities of random subsets of neurons from all populations are shown (hidden: excitatory neurons of the recurrent network, which are not in pool D or U). Bottom: lever position inferred from the neural →

→ activity in pools D and U. Rewards are indicated by red bars. Gray shaded areas indicate cue presentation. **C:** Task performance quantified by the average time from cue presentation onset to movement completion. The network was able to solve this task in less than 1 seconds on average after about 8 hours of learning. A task change was introduced at time 24 h (asterisk; function of D and U switched), which was quickly compensated by the network. Using a simplified version of the learning rule, where the re-introduction of non-functional potential connections was approximated using exponentially distributed waiting times (green), yielded similar results (see also panel e). If the connectome was kept fixed after the task change at 24 h performance was significantly worse (black). **D:** Trial-averaged network activity (top) and lever movements (bottom). Activity traces are aligned to movement onsets (arrows). Y-axis of trial-averaged activity plots are sorted by the time of highest firing rate within the movement at various times during learning: sorting of the first and second plot is based on the activity at $t = 0$ h, third and fourth by that at $t = 22$ h, fifth is resorted by the activity at $t = 46$ h. Network activity is clearly restructured through learning with particularly stereotypical assemblies for sharp upward movements. Bottom: average lever movement (black) and 10 individual movements (gray). **E:** Turnover of synaptic connections for the experiment shown in (D). Y-axis is clipped at 3000. Turnover rate during the first two hours was around 12.000 synapses ($\sim 25\%$) and then decreased rapidly. Another increase in spine turnover rate can be observed after the task change at time 24 h. **F:** Histograms of time intervals between disappearance and reappearance of synapses (waiting times) for the exact (upper plot) and approximate (lower plot) learning rule. **G:** Relative fraction of potential synaptic connections that were stably non-functional, transiently decaying, transiently emerging or stably function during the re-learning phase for the experiment shown in (D). **H:** PCA of a random subset of the parameters θ_i. The plot suggests continuing dynamics in task-irrelevant dimensions after the learning goal has been reached (indicated by red color). When the function of the neuron pools U and D was switched after 24 h, the synaptic parameters migrated to a new region. All plots show means over 5 independent runs (error bars: s.e.m.).

We adapted the learning task of (A. J. Peters et al., 2014) in the following way for our model (see Fig. 4.4A). The beginning of a trial was indicated through the presentation of a cue input pattern $x(t)$: a fixed, randomly generated rate pattern for all 200 input neurons that lasted until the task was completed, but at most 10s. When the lever position crossed the threshold +5 after first crossing a lower threshold -5 (black horizontal lines in Fig. 4.4A,B) within 10 s after cue onset a 400 ms reward window was initiated during which $r(t)$ was set to 1 (red vertical bars in Fig. 4.4b). Unsuccessful trials were aborted after 10 seconds and no reward was delivered. After each trial a brief holding phase of random length was inserted, during which input neurons were set to a background input rate of 2 Hz.

As network scaffold $\mathcal{N}$ we took a generic recurrent network of excitatory and inhibitory spiking neurons with connectivity parameters for connections between excitatory and inhibitory neurons according to data from layer 2/3 in mouse cortex (Avermann et al., 2012). The network consisted of 60 excitatory and 20 inhibitory neurons (see Fig. 4.4A). Half of the excitatory neurons could potentially receive synaptic connections from the 200 excitatory input neurons. From the remaining 30 neurons we randomly selected one pool D of 10 excitatory neurons to cause downwards movements of the lever, and another pool U of 10 neurons for upwards movements. We refer to the 40 excitatory neurons that were not members of D or U as hidden neurons. All excitatory synaptic connections from the external input (cue) and between the 60 excitatory neurons (including those in the pools D and U) in the network were subjected to reward-based synaptic sampling. Thus, the

network had to learn without any guidance, except for the reward in response to good performance, to create after the onset of the cue first higher firing in pool D, and then higher firing in pool U. This task was challenging, since the network had no information which neurons belonged to pools D and U. Moreover, the synapses did not "know" whether they connected to hidden neurons, neurons within a pool, hidden neurons and pool-neurons, or input neurons with other neurons. The plasticity of all these different synapses was gated by the same global reward signal. Since the pools D and U were not able to receive direct synaptic connections from the input neurons, the network also had to learn to communicate the presence of the cue pattern via disynaptic connections from the input neurons to these pools.

Network responses before and after learning are shown in Fig. 4.4B. Initially, the rewarded goal was only reached occasionally, while the turnover of synaptic connections (number of synaptic connections that became functional or became non-functional in a time window of 2 hours) remained very high (see Fig. 4.4E). After about 3 h, performance improved drastically (Fig. 4.4C), and simultaneously the turnover of synaptic connections slowed down (Fig. 4.4E). After learning for 8 hours, the network was able to solve the task in most of the trials, and the average trial duration (movement completion time) had decreased to less than 1 second (851 ± 46 ms, Fig. 4.4C). Improved performance was accompanied by more stereotyped network activity and lever movement patterns as in the experimental data of (A. J. Peters et al., 2014): compare our Fig. 4.4D with Fig. 1b and Fig. 2j of (A. J. Peters et al., 2014). In Fig. 4.4D we show the trial-averaged activity of the 60 excitatory neurons before and after learning for 22 hours. The neurons are sorted in the first two plots of Fig. 4.4D by the time of maximum activity after movement onset times before learning, and in the 3rd plot resorted according to times of maximum activity after 22 hours of learning (see Appendix D). These plots show that reward-based learning led to a restructuring of the network activity: an assembly of neurons emerged that controlled a sharp upwards movement. Also, less background activity was observed after 22 hours of learning, in particular for neurons with early activity peaks. Lower panels in Fig. 4.4D show the average lever movement and 10 individual movement traces at the beginning and after 22 hours of learning. Similar as in (A. J. Peters et al., 2014) the lever movements became more stereotyped during learning, featuring a sharp upwards movement at cue onset followed by a slower downwards movement in preparation for the next trial.

Next we tested whether similar results could be achieved with a simplified version of the stochastic synapse dynamics while a potential synaptic connection i is non-functional, i.e., $\theta_i \leq 0$. Eq. (4.5) defines for such non-functional synapses an Ornstein-Uhlenbeck process, which yields a heavy-tailed distribution for the waiting time until reappearance (Fig. 4.4F, top). We tested whether similar learning performance can be achieved if one approximates the distribution by an exponential distribution, for which we chose a mean of 12 h. The small distance between the blue and green curve in Fig. 4.4C shows that this is in fact the case for the overall computational task that includes a task switch at 24 h that we describe below. This holds in spite of the fact that the approximating exponential distribution is

less heavy-tailed (Fig. 4.4F, bottom). Altogether these results show that rewiring and synaptic plasticity according to Eq. (4.5) yields self-organization of a generic recurrent network of spiking neurons so that it can control an arbitrarily chosen motor control task.

4.7 Compensation for network perturbations

We wondered whether this model for the task of [Peters et al., 2014] would in addition be able to compensate for a drastic change in the task, an extra challenge that had not been considered in the experiments of (A. J. Peters et al., 2014). To test this we suddenly interchanged after 24 h the actions that were triggered by the pools D and U. D now caused upwards and U downwards lever movement.

We found that our model compensated immediately (see the faster movement in the parameter space depicted in Fig. 4.4H) for this perturbation and reached after about 8 h a similar performance level as before (Fig. 4.4C). This compensation phase was accompanied by a substantial increase in the turnover of synaptic connections similar as in experiments for learning of a new task, see e.g. (T. Xu et al., 2009) (Fig. 4.4E). The turnover rate also remained slightly elevated during the subsequent learning period. Furthermore, a new assembly of neurons emerged that now triggered a sharp onset of activity in the pool D (compare the activity neural traces at h = 22 and h = 46 in Fig. 4.4D). Drifts of neural codes also emerge in our model during phases of the experiment without perturbations, while the task performance stays constant, similar to experimental data in (Driscoll and C. Harvey, 2016) (see Fig. D.1 in Appendix D).

If rewiring was disabled after the task change at 24 h the compensation was significantly delayed and overall performance declined (see black curve in Fig. 4.4C). Here, we disallowed any turnover of potential synaptic connections such that the connectivity remained the same after 24 h. This result suggests that rewiring is necessary for adapting to the task change. In Fig. 4.4G we further analyzed the profile of synaptic turnover for the different populations of the network scaffold in Fig. 4.4A. The synaptic parameters were measured immediately before the task change at 24 h and compared to the connectivity after compensation at 48 h for the experiment shown in Fig. 4.4C (blue). Most synapses (66-75%) were non-functional before and after the task change (stable non-functional). About 20% of the synapses changed their behavior and either became functional or non-functional. Most prominently a large fraction (21.9%) of the synapses from hidden neurons to U became non-functional while only few (5.9%) new connections were introduced. The connections from hidden to D showed the opposite behavior. This modification of the network connectome reflects the requirement to reliably route information about the presence of the cue pattern encoded in the activity of hidden neurons to the pool D (and not to U) to initiate the lever movement after the task change.

A structural difference between stochastic learning models such as Bayesian policy sampling and learning models that focus on convergence of parameters to a (locally) optimal setting becomes apparent when one tracks the temporal evolution of the network parameters $\boldsymbol{\theta}$ over larger periods of time during the previously discussed learning process (Fig. 4.4H). Although performance no longer improved after 5 hours, both network connectivity and parameters kept changing in task-irrelevant dimensions. For Fig. 4.4H we randomly selected 5% of the roughly 47000 parameters θ_i and plotted the first 3 principal components of their dynamics. The task change after 24 hours caused the parameter vector $\boldsymbol{\theta}$ to migrate to a new region within about 8 hours of continuing learning. Again we observe that Bayesian policy sampling keeps exploring different equally good solutions after the learning process has reached stable performance.

Relative contributions of spontaneous and activity-dependent synaptic processes

(Dvorkin and N. E. Ziv, 2016) analyzed the correlation of sizes of postsynaptic densities and spine volumes for synapses that shared the same pre- and postsynaptic neuron, called commonly innervated (CI) synapses, and also for synapses that shared in addition the same dendrite (CI_{SD}). Activity-dependent rules for synaptic plasticity, such as Hebbian or STDP rules – on which previous models for network plasticity relied – suggest that the strength of CI and especially CI_{SD} synapses should be highly correlated. But both data from ex-vivo (Kasthuri et al., 2015) and neural circuits in culture (Dvorkin and N. E. Ziv, 2016) show that postsynaptic density sizes and spine volumes of CI_{SD} synapses are only weakly correlated, with correlation coefficients between 0.23 and 0.34. Thus even with a conservative estimate that corrects for possible influences of their experimental procedure, more than 50% of the observed synaptic strength appears to result from activity-independent stochastic processes (Fig. 8E of (Dvorkin and N. E. Ziv, 2016)). A smaller data set (based on 17 CI_{SD} pairs instead of the 72 pairs, 10 triplets, and 2 quadruplets in the ex-vivo data from (Kasthuri et al., 2015)) had previously been analyzed for correlations of synapse strengths of CI_{SD} synapses in (Bartol Jr et al., 2015). They found that the spine volumes differed in these pairs on average by a factor around 2. Their data also contained a CI_{SD} triplet, depicted at the bottom of Fig. 4B, that apparently had larger differences but was excluded from the data analysis.

We asked how such a strong contribution of activity-independent synaptic dynamics affects network learning capabilities, such as the ones that were examined in Fig. 4.4. We were able to carry out this test because many synaptic connections between neurons that were formed in our model consisted of more than one synapse (to be precise: 49% of connections consisted of multiple synapses). We classified pairs of synapses that had the same pre- and postsynaptic neuron as CI synapses (one could also call them CI_{SD} synapses, since the neuron model did not have different dendrites), and pairs with the same postsynaptic but different presynaptic neurons

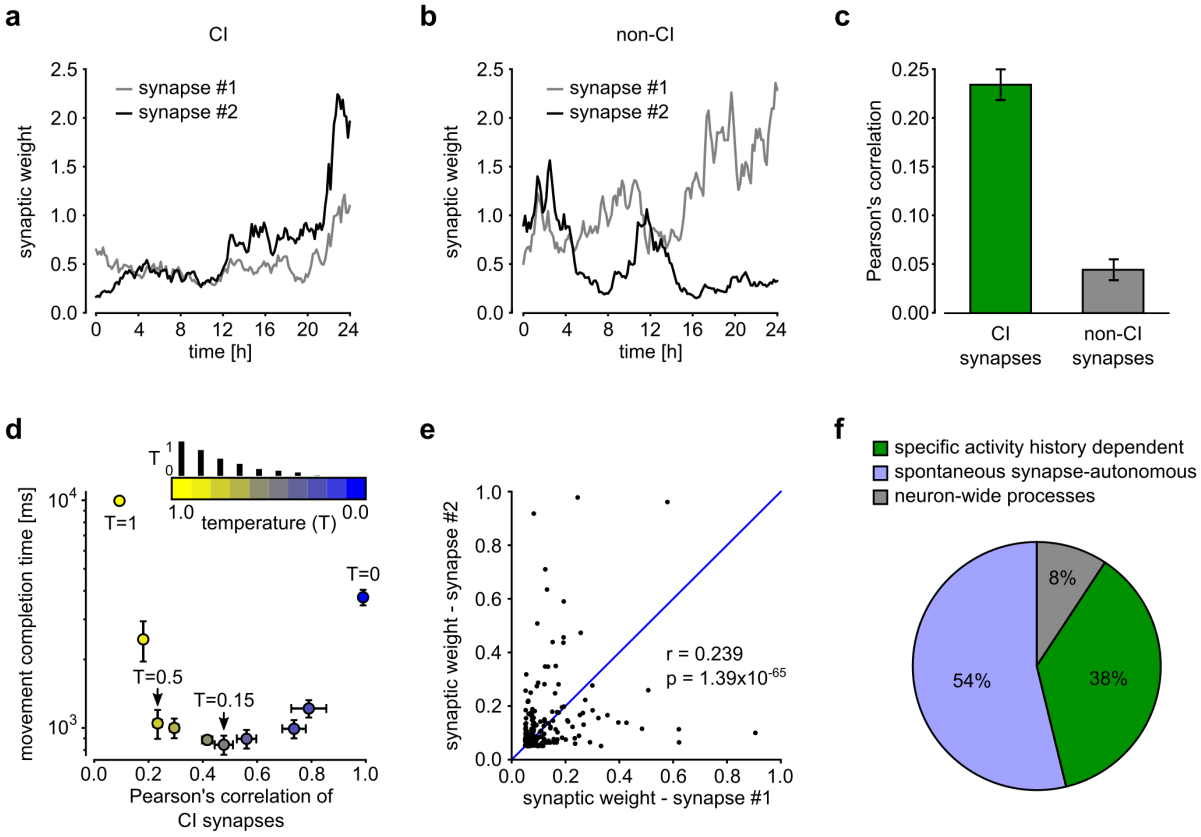

Fig. 4.5: Contribution of spontaneous and neural activity-dependent processes to synaptic dynamics A,B: Evolution of synaptic weights w_i plotted against time for a pair of CI synapses in (A), and non-CI synapses in (B), for temperature $T = 0.5$. **C:** Pearson's correlation coefficient computed between synaptic weights of CI and non-CI synapses of a network with $T = 0.5$ after 48 h of learning as in Fig. 4.4D-G. CI synapses were only weakly, but significantly stronger correlated than non-CI synapses. **D:** Impact of T on correlation of CI synapses (x-axis) and learning performance (y-axis). Each dot represents averaged data for one particular temperature value, indicated by the color. Values for T were 1.0, 0.75, 0.5, 0.35, 0.2, 0.15, 0.1, 0.01, 0.001, 0.0. These values are proportional to the small vertical bars above the color bar. The performance (measured in movement completion time) is measured after 48 hours for the learning experiment as in Fig. 4.4D-G, where the network changed completely after 24 h. Good performance was achieved for a range of temperature values between 0.01 and 0.5. Too low (< 0.01) or too high (> 0.5) values impaired learning. Means + s.e.m. over 5 independent trials are shown. **E:** Synaptic weights of 100 pairs of CI synapses that emerged from a run with $T = 0.5$. Pearson's correlation is 0.239, comparable to the experimental data in Fig. 8A-D of (Dvorkin and N. E. Ziv, 2016). **F:** Estimated contributions of activity history dependent (green), spontaneous synapse-autonomous (blue) and neuron-wide (gray) processes to the synaptic dynamics for a run with $T = 0.15$. The resulting fractions are very similar to those in the experimental data, see Fig. 8E of (Dvorkin and N. E. Ziv, 2016).

as non-CI synapses. Example traces of synaptic weights for CI and non-CI synapse pairs of our network model from Fig. 4.4 are shown in Fig. 4.5A,B. CI pairs were found to be more strongly correlated than non-CI pairs (Fig. 4.5C). However also the correlation of CI pairs was quite low, and varied with the temperature parameter T in Eq. (4.5), see Fig. 4.5D. The correlation was measured in terms of the Pearson correlation (covariance of synapse pairs normalized between -1 and 1).

Since, the correlation of CI pairs in our model depends on the temperate T, we analyzed the model of Fig. 4.4 for different temperatures (the temperature had been fixed at T=0.1 throughout the experiments for Fig. 4.4). In Fig. 4.5D the Pearson correlation coefficient for CI synapses is plotted together with the average performance achieved on the task of Fig. 4.4D-G for networks with different temperatures T. The best performing temperature region for the task ($0.01 \leq T \leq 0.5$) roughly coincided with the region of experimentally measured values of Pearson's correlation for CI-synapses. Fig. 4.5E shows the correlation of 100 CI synapse pairs that emerged from a run with $T = 0.5$. We found a value of $r = 0.239$ in this case. This value is in the order of the lowest experimentally found correlation coefficients in (Dvorkin and N. E. Ziv, 2016) (both in culture and ex-vivo, see Fig. 8A-D in (Dvorkin and N. E. Ziv, 2016)). For $T = 0.15$ we found the best task performance and the closest match to experimentally measured correlations when the results of (Dvorkin and N. E. Ziv, 2016) were corrected for measurement limitations: A correlation coefficient of $r = 0.46 \pm 0.034$ for CI synapses and 0.08 ± 0.015 for non-CI synapse pairs (mean $\pm$ s.e.m. over 5 trials, 2-tailed p-value below 0.005 in all trials).

(Dvorkin and N. E. Ziv, 2016) further analyzed the ratio of contributions from different processes to the measured synaptic dynamics. They analyzed the contribution of neural activity history dependent processes, which amount for 36% of synapse dynamics in their data, and that of neuron-wide processes that were not specific to presynaptic activity, but specific to the activity of the postsynaptic neuron (8%). Spontaneous synapse-autonomous processes were found to explain 56% of the observed dynamics (see Fig. 8E in (Dvorkin and N. E. Ziv, 2016)). The results from our model, that are plotted in Fig. 4.5F, match these experimentally found values quite well. Altogether we found that the results of (Dvorkin and N. E. Ziv, 2016) are best explained by our model for a temperature parameter between $T = 0.5$ (corresponding to their lowest measured correlation coefficient) and $T = 0.15$ (corresponding to their most conservative estimate). This range of parameters coincided with well-functioning learning behavior in our model, which included a test of compensation capability for a change of the task after 24 h (Fig. 4.5D). Hence our model suggests that a large component of stochastic synapse-autonomous processes, as it occurs in the data, supports efficient network learning and compensation for changes in the task.

4.8 Discussion

Recent experimental data ((Dvorkin and N. E. Ziv, 2016), where in Fig. 8 also mouse brain data from (Kasthuri et al., 2015) were reanalyzed) suggest that common models for learning in neural networks of the brain need to be revised, since synapses are subject to powerful processes that do not depend on pre- and postsynaptic neural activity. In addition, experimentally found network rewiring has so far not been integrated into models for reward-gated network plasticity. We have presented a theoretical framework that enables us to investigate and understand reward-based network rewiring and synaptic plasticity in the context of the experimentally found high level of activity-independent fluctuations of synaptic connectivity and synaptic strength. We have shown that the analysis of the stationary distribution of network configurations, in particular the Fokker-Planck equation from theoretical physics, allows us to understand how large numbers of local stochastic processes at different synapses can orchestrate global goal-directed network learning. This approach provides a new normative model for reward-gated network plasticity.

We have shown in Fig. 4.2 that the resulting model is consistent with experimental data on dopamine-dependent spine dynamics reported in (Yagishita et al., 2014), and that it provides an understanding how these local stochastic processes can produce function-oriented cortical-striatal connectivity. We have shown in Fig. 4.4 that this model also elucidates reward-based self-organization of generic recurrent neural networks for a given computational task. We chose as benchmark task the production of a specific motor output in response to a cue, like in the experiments of (A. J. Peters et al., 2014). Similarly as reported in (A. J. Peters et al., 2014), the network connectivity and dynamics reorganized itself in our model, just driven by stochastic processes and rewards for successful task completion, and reached a high level of computational performance. Furthermore it maintained this computational function in spite of continuously ongoing further rewiring and network plasticity. A quantitative analysis of the impact of stochasticity on this process has shown in Fig. 4.5 that the network learns best when the component of synaptic plasticity that does not depend on neural activity is fairly large, as large as reported in the experimental data of (Kasthuri et al., 2015; Dvorkin and N. E. Ziv, 2016).

Our approach is based on experimental data for the biological implementation level of network plasticity, i.e., for the lowest level of the Marr hierarchy of models (Marr and Poggio, 1976). However, we have shown that these experimental data have significant implications for understanding network plasticity on the top level ("what is the functional goal?") and the intermediate algorithmic level ("what is the underlying algorithm?") of the Marr hierarchy. They suggest for the top level that the goal of network plasticity is to sample from a posterior distribution of network configurations. This posterior should integrate functional demands with priors that represent structural constraints as well as results of preceding learning experiences and innate programs. In other words, our model suggests to view reward-gated network plasticity as Bayesian inference on a slow time scale. This Bayesian perspective also creates a link to previous work on Bayesian reinforcement

learning (Vlassis et al., 2012; Rawlik et al., 2013). The experimental data suggest for the intermediate algorithmic level of the Marr hierarchy a strong reliance on stochastic search ("synaptic sampling"). The essence of the resulting model for reward-gated network learning is illustrated in Fig. 4.1: The traditional view of deterministic gradient ascent (policy gradient) in the landscape (panel b) of reward expectation is first modified through the integration of a prior (panel c), and then through the replacement of gradient ascent by continuously ongoing stochastic sampling (Bayesian policy sampling) from the posterior distribution of panel d, which is illustrated in panels e and f.

This model makes a number of experimentally testable predictions. Continuously ongoing stochastic sampling of network configurations suggests that synaptic connectivity does not converge to a fixed point solution but rather undergoes permanent modifications (Fig. 4.4G). This prediction is compatible with reports of continuously ongoing spine dynamics and axonal sprouting even in the adult brain (A. Holtmaat and Svoboda, 2009; Yasumatsu et al., 2008; Stettler et al., 2006; Yamahachi et al., 2009; Loewenstein et al., 2011; A. J. Holtmaat et al., 2005; Loewenstein et al., 2015). These continuously ongoing parameter changes predict continuously ongoing changes in the assembly sequences that accompany and control a motor response (see Fig. 4.4D). Our model predicts, that these changes do not impair the performance of the network, but rather induce the network to explore different but equally good solutions when exposed for many hours to the same task (see Fig. 4.4G). Such continuously ongoing drifts of neural codes in functionally less relevant dimensions have already been observed experimentally in some brain areas (Y. Ziv et al., 2013; Driscoll and C. Harvey, 2016). Our model also predicts that the same computational function is realized by the same neural circuit in different individuals with drastically different parameters, a feature which has already been addressed in (Tang et al., 2010; Grashow et al., 2010; Marder, 2011; Prinz et al., 2004). In fact, this *degeneracy* of neural circuits is thought to be an important property of biological neural networks (Marder, 2011; Prinz et al., 2004; Marder and Goaillard, 2006). In addition, our model predicts that neural networks automatically compensate for disturbances by moving their continuously ongoing sampling of network configurations to a new region of the parameter space, as illustrated by the response to the disturbance marked by $*$ in Fig. 4.4G.

In conclusion the mathematical framework presented in this article provides a principled way of understanding the complex interplay of deterministic and stochastic processes that underlie the implementation of goal-directed learning in neural circuits of the brain. It also offers a solution to the problem how reliable network computations can be achieved with a “dynamic connectome” (Rumpel and Triesch, 2016). We have argued that the stationary distribution of the high-dimensional parameter vector $\boldsymbol{\theta}$ that results from large numbers of local stochastic processes at the synapses provides a time-invariant perspective of salient properties of a network. Standard reward-gated plasticity rules can achieve that this stationary distribution has most of its mass on regions in the parameter space that provide good network performance. The stochastic component of synaptic dynamics can

flatten or sharpen the resulting stationary distribution, depending on whether the scaling parameter T ("temperature") of the stochastic component is larger or smaller than 1. A functional benefit of this stochastic component is that the network keeps exploring its parameter space even after a well-performing region has been found. This enables the network to migrate quickly and automatically to a better performing region when the network or task change. We found in the case of the motor learning task of Fig. 4.4 that a temperature T around 0.15, which lies in the same range as related experimental data (see Fig. 4.5d), suffices to provide this functionally important compensation capability.

Appendix

Appendix A

List of publications

1. David Kappel*, Robert Legenstein*, Stefan Habenschuss, Michael Hsieh, Wolfgang Maass (2017). "Reward-based self-configuration of neural circuits." *(submitted for publication)*.
2. Guillaume Bellec, David Kappel, Robert Legenstein, Wolfgang Maass (2017). "Deep Rewiring : Training very sparse deep networks." *(submitted for publication)*.
3. Zhaofei Yu1*, David Kappel*, Robert Legenstein*, Sen Song, Feng Chen, Wolfgang Maass (2016). "CaMKII activation supports reward-based neural network optimization through Hamiltonian sampling." *(submitted for publication)*.
4. Elmar Rueckert, David Kappel, Daniel Tanneberg, Dejan Pecevski, Jan Peters (2016). "Recurrent Spiking Networks Solve Planning Tasks." *Scientific Reports*.
5. David Kappel*, Stefan Habenschuss*, Robert Legenstein, Wolfgang Maass (2015). "Network Plasticity as Bayesian Inference." *PLoS Computational Biology*.
6. David Kappel*, Stefan Habenschuss*, Robert Legenstein, Wolfgang Maass (2015). "Synaptic Sampling: A Bayesian Approach to Neural Network Plasticity and Rewiring." *Neural Information Processing Systems*.
7. Dejan Pecevski, David Kappel, Zeno Jonke (2014). "NEVESIM: event-driven neural simulation framework with a Python interface." *Frontiers in Neuroinformatics*.
8. David Kappel, Bernhard Nessler, Wolfgang Maass (2014). "STDP installs in winner-take-all circuit an online approximation to hidden Markov model learning." *PLoS Computational Biology*.

* equal contribution

Appendix B

Appendix to Chapter 2: STDP in winner-take-all circuits approximates hidden Markov model learning

B.1 Spiking network model

In this section we provide additional details to the derivations of the network model and its stochastic dynamics. For the sake of simplicity, throughout the theoretical analysis we use a simple EPSP kernel of the form

$$\epsilon(s) = \exp(-s/\tau) \cdot \Theta(s). \tag{B.1}$$

Thus, a kernel with a single exponential decay with time constant τ. Here, $\Theta(s)$ determines the Heaviside step function which is 1 for $s > 0$ and zero else.

The derivation provided here can be extended to more complex EPSP shapes, if two prerequisites are fulfilled. First, a suitable Markov state must be found that describes the dynamics of the EPSP kernel, i.e. a state s_m must exist for which we can write $p(s_m \mid s_{m-1}, s_{m-2}, \ldots, \boldsymbol{\theta}) = p(s_m \mid s_{m-1}, \boldsymbol{\theta})$. In fact, this property holds true for any deterministic function, although the required Markov state can be very complex. Second, the statistics of the EPSPs induced by the kernel must be readily described by an exponential family distribution. For this latter requirement the same considerations as for the afferent synapses apply, which have been addressed in (Nessler et al., 2010; Habenschuss et al., 2013; Nessler et al., 2013). The simplest case for which these conditions are fulfilled is the one considered in the last experiment where rectangular EPSPs and constant inter-spike intervals Δ_m of the same length were used. In that case the network state collapses to $s_m = z_m$, which follows a multinomial distribution as considered in (Nessler et al., 2010).

B.2 Details to: Forward sampling in WTA circuits

We show here that the WTA circuit correctly implements forward sampling in a HMM. In particular we show that a HMM with an observation model from the exponential family can be directly mapped to the network dynamics. Many of the theoretical details for the special case of stationary input patterns were analyzed in

(Nessler et al., 2010; Habenschuss et al., 2013; Nessler et al., 2013), here we focus on the derivations specific for the network with lateral excitatory connections.

First we define a HMM with observations $\boldsymbol{x}_m$ and hidden state $s_m = \{z_m, \boldsymbol{y}_m, \Delta_m\}$ to reflect the dynamics of the WTA circuit. The HMM joint distribution is given by equation (2.4). Each time step m factorizes into the *observation model* $p\left(\boldsymbol{x}_m \mid s_m, \boldsymbol{\theta}\right)$ and the *prediction model* $p\left(s_m \mid s_{m-1}, \boldsymbol{\theta}\right)$. We assume a mixture of exponential family distributions for the observation model. Many interesting distributions are members of this family, e.g. the Poisson or the Normal distribution. The network output z_m determines which mixture component is responsible for the observation $\boldsymbol{x}_m$. In its generic form the likelihood of the N-dimensional observations $\boldsymbol{x}_m$ given mixture component k can be written as

$$p\left(\boldsymbol{x}_m \mid z_m \equiv k, \boldsymbol{\theta}\right) \quad = \quad h^{(x)}\left(\boldsymbol{x}_m\right) \cdot \exp\left(\sum_{i=1}^{N} w_{ki} \cdot x_{mi} - A_k^{(x)}(\boldsymbol{W})\right) \tag{B.2}$$

where $h^{(x)}\left(\boldsymbol{x}_m\right)$ is a base measure and $A_k^{(x)}(\boldsymbol{W})$ is the log-partition function, which assure that (B.2) is correctly normalized. In this framework $\boldsymbol{x}_m$ determines the sufficient statistics of the input distribution, e.g. the current input rate for the Poisson distribution, which is estimated by filtering the input spike train with the EPSP kernel. Since the input and output spike times are independent, given optimal WTA behavior, we exploit the conditional independence $p\left(\boldsymbol{x}_m \mid s_m, \boldsymbol{\theta}\right) = p\left(\boldsymbol{x}_m \mid z_m, \boldsymbol{\theta}\right)$. We assume that inputs are homogeneous, meaning that the sums over all input channels are constant. More precisely, we assume that $\sum_{i=1}^{N} x_i(t) = A_0^{(x)}$ and $\sum_{j=1}^{K} y_j(t) = A_0^{(y)}$ holds true at all times. These assumptions were never perfectly fulfilled in the simulations, but nevertheless the algorithm was robust against deviations from these constraints throughout all experiments. The choice of the log-partition function determines the member from the exponential family. The derivations here were done for Poisson distributed inputs but they equally apply to other members. Given the homogeneity assumption for the input we find the log-partition to be $A_k^{(x)}(\boldsymbol{w}) = \sum_{i=1}^{N} \mathrm{e}^{w_{ki}}$ (Habenschuss et al., 2013).

The prediction model has to reflect the dynamics of the state s_m. At each time point $\hat{t}_m$ the spiking output projects the K-dimensional state $\boldsymbol{y}_m$ to the discrete value z_m, which is then projected in the next step to $\boldsymbol{y}_{m+1}$. Using the independence properties, that emerge from these dynamics, the prediction model factorizes to

$$p\left(s_m \mid s_{m-1}, \boldsymbol{\theta}\right) = p\left(z_m \mid \boldsymbol{y}_m, \boldsymbol{\theta}\right) p\left(\boldsymbol{y}_m \mid z_{m-1}, \boldsymbol{y}_{m-1}, \Delta_{m-1}\right) p\left(\Delta_{m-1}\right). \tag{B.3}$$

The last term determines the distribution over the inter-spike intervals Δ_m. Assuming Poisson distributed spike times $\hat{t}_m$ this is given by an exponential distribution with mean $\hat{\nu}^{-1}$, i.e.

$$p\left(\Delta_m\right) = \frac{1}{\hat{\nu}} \mathrm{e}^{\frac{\Delta_m}{\hat{\nu}}}. \tag{B.4}$$

The second part of (B.3) deterministically updates the EPSPs. Using the simple kernel function (B.1) the lateral EPSPs can be updated online

$$y_j(\hat{t}_m + \Delta) = \begin{cases} e^{-\Delta/\tau} \cdot (y_{jm} + 1), & \text{if neuron } j \text{ spiked at time } \hat{t}_m \\ e^{-\Delta/\tau} \cdot y_{jm}, & \text{else} \end{cases} . \tag{B.5}$$

Since Eq. (B.5) is deterministic, $p(\boldsymbol{y}_m \mid z_{m-1}, \boldsymbol{y}_{m-1}, \Delta_{m-1})$ in (B.3) collapses to a single mass point, where the update equation (B.5) is fulfilled. The second and third parts of (B.3) project the spiking network output to the K-dimensional space of the EPSP time courses (2.2). The first part of the prediction model projects it back to a discrete variable drawn from a multinomial distribution $p(z_m \mid \boldsymbol{y}_m, \boldsymbol{\theta})$. Using Bayes rule, we can decompose this into

$$p(z_m \mid \boldsymbol{y}_m, \boldsymbol{\theta}) \quad = \quad p(\boldsymbol{y}_m \mid z_m, \boldsymbol{\theta}) \, \frac{p(z_m \mid \boldsymbol{\theta})}{p(\boldsymbol{y}_m \mid \boldsymbol{\theta})} . \tag{B.6}$$

The likelihood term can again be expressed in terms of an exponential family distribution

$$p(\boldsymbol{y}_m \mid z_m \equiv k, \boldsymbol{\theta}) \quad = \quad h^{(y)}(\boldsymbol{y}_m) \cdot \exp\left(\sum_{j=1}^{K} v_{kj} \cdot y_{mj} - A_k^{(y)}(\boldsymbol{v}) \right) . \tag{B.7}$$

For each neuron k the prior probability to fire is determined by the excitability parameter, i.e. $p(z_m \equiv k \mid \boldsymbol{\theta}) = \mathrm{e}^{b_k}$. In (Nessler et al., 2010) a learning rule was presented for these network parameters, which equally applies to the framework presented here. For simplicity however, we can assume that all neurons have the same prior probability to fire, i.e. $p(z_m \equiv k \mid \boldsymbol{\theta}) = \frac{1}{K}$.

Under the homogeneity condition and if the synaptic weights obey $\sum_{j=1}^{K} e^{v_{kj}} = A_0^{(y)}$, the log-partition function $A_k^{(y)}(\boldsymbol{v})$ becomes constant over k. It has been shown that this condition emerges automatically from the STDP rules (2.7) (Habenschuss et al., 2013). Using this, the probability of generating a state sequence $\boldsymbol{S}$ using forward sampling can be directly linked to the network dynamics. The true posterior distribution (2.8) and the proposal distribution for forward sampling only differ in the normalization. Forward sampling is done, by explicitly normalizing the state update in (2.4) at each time point m (Koller and Friedman, 2009). This normalization is given by $\int p(\boldsymbol{x}_m \mid s'_m)\, p(s'_m \mid s_{m-1})\, \mathrm{d}s'_m = p(\boldsymbol{x}_m \mid s_{m-1})$, from which we find the proposal distribution to be given by

$$\begin{aligned} q(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}) &= \prod_{m=1}^{M} \frac{p(\boldsymbol{x}_m \mid s_m)\, p(s_m \mid s_{m-1})}{p(\boldsymbol{x}_m \mid s_{m-1})} \\ &= \prod_{m=1}^{M} p(z_m \mid \boldsymbol{x}_m, \boldsymbol{y}_m, \boldsymbol{\theta})\, p(\boldsymbol{y}_m \mid z_{m-1}, \boldsymbol{y}_{m-1}, \Delta_{m-1})\, p(\Delta_{m-1}) . \end{aligned} \tag{B.8}$$

This recovers the result of equation (2.5). The first term of the second line can be written using (B.2), (B.6) and (B.7)

$$\begin{aligned} p\left(z_m \equiv k \mid \boldsymbol{x}_m, \boldsymbol{y}_m, \boldsymbol{\theta}\right) &= \frac{p\left(\boldsymbol{x}_m \mid z_m \equiv k, \boldsymbol{\theta}\right) p\left(z_m \equiv k \mid \boldsymbol{y}_m, \boldsymbol{\theta}\right)}{\sum_{l=1}^{K} p\left(\boldsymbol{x}_m \mid z_m \equiv l, \boldsymbol{\theta}\right) p\left(z_m \equiv l \mid \boldsymbol{y}_m, \boldsymbol{\theta}\right)} \\ &= \exp\left(\sum_{i=1}^{N} w_{ki} \cdot x_{mi} + \sum_{j=1}^{K} v_{kj} \cdot y_{mj} + b_k - i(\hat{t}_m)\right) , \end{aligned}$$

with $i(\hat{t}_m)$ given by (2.3). Here we have used that the marginal in the denominator of (B.6) does not depend on k and so do $A_k^{(x)}(\boldsymbol{w})$ and $A_k^{(y)}(\boldsymbol{v})$ under the conditions described above. Therefore they cancel out through the normalization (2.3). Comparing this result with the neuron dynamics (2.6) and (2.5) it is easy to verify that the WTA circuit correctly realizes the HMM forward sampler (B.8).

B.3 Details to: STDP installs a stochastic approximation to EM parameter learning

In this section we derive the optimal updates for the model parameters in terms of the expectation-maximization (EM)-algorithm and show that the STDP rules (2.7) are stochastic approximations. The goal of the EM optimization is to minimize the error between the model likelihood $p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right)$ and the empirical distribution over input sequences, which we denote by $p^*\left(\boldsymbol{X}\right)$. A natural way to express this error is the Kullback-Leibler divergence. Thus, the update can be derived by minimizing

$$\begin{aligned} \mathrm{KL}\left(p^*\left(\boldsymbol{X}\right) \parallel p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right)\right) &= \int p^*\left(\boldsymbol{X}'\right) \log \frac{p^*\left(\boldsymbol{X}'\right)}{p\left(\boldsymbol{X}' \mid \boldsymbol{\theta}\right)} \mathrm{d}\boldsymbol{X}' \\ &= -\mathrm{H}_{p^*}\left(\boldsymbol{X}\right) - \left\langle \log p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right) \right\rangle_{p^*} , \end{aligned} \tag{B.9}$$

where $\mathrm{H}_{p^*}\left(\boldsymbol{X}\right)$ is the entropy of the true input distribution and $\left\langle \cdot \right\rangle_{p^*}$ denotes the expectation with respect to $p^*\left(\boldsymbol{X}\right)$. We are interested in a solution to $\boldsymbol{\theta}$ that minimizes (B.9). Since $\mathrm{H}_{p^*}\left(\boldsymbol{X}\right)$ is constant for a given input sequence $\boldsymbol{X}$, it can be ignored and minimizing (B.9) becomes equivalent to maximizing the expected log-likelihood $\left\langle \log p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right) \right\rangle_{p^*(\boldsymbol{X})}$. The derivative of the log-likelihood can be simplified to

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\theta}} \log p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right) &= \frac{\partial}{\partial \boldsymbol{\theta}} \log \int p\left(\boldsymbol{X}, \boldsymbol{S}' \mid \boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{S}' \\ &= \int p\left(\boldsymbol{S}' \mid \boldsymbol{X}, \boldsymbol{\theta}\right) \frac{\partial}{\partial \boldsymbol{\theta}} \log p\left(\boldsymbol{X}, \boldsymbol{S}' \mid \boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{S}' . \end{aligned} \tag{B.10}$$

The integral can again be written in terms of an expectation. The condition for the maximum likelihood becomes

$$\left\langle \frac{\partial}{\partial \boldsymbol{\theta}} \log p\left(\boldsymbol{X}, \boldsymbol{S} \mid \boldsymbol{\theta}\right) \right\rangle_{\hat{p}} = 0 , \tag{B.11}$$

where $\langle \cdot \rangle_{\hat{p}}$ denotes the expectation with respect to $p^*(\boldsymbol{X})\, p(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta})$. The derivative in this last form can be easily calculated. By inserting the model joint distribution (2.4) it yields for the model parameters v_{kj} of neuron k

$$\begin{aligned}
&\frac{\partial}{\partial v_{kj}} \log p(\boldsymbol{X}, \boldsymbol{S} \mid \boldsymbol{\theta}) = \\
&= \frac{\partial}{\partial v_{kj}} \log \prod_{m=1}^{M} p(\boldsymbol{x}_m \mid z_m, \boldsymbol{\theta})\, p(z_m \mid \boldsymbol{y}_m, \boldsymbol{\theta})\, p(\boldsymbol{y}_m \mid z_{m-1}, \boldsymbol{y}_{m-1}, \Delta_{m-1})\, p(\Delta_{m-1}) \\
&= \sum_{m=1}^{M} \frac{\partial}{\partial v_{kj}} \delta_{k,z_m} \left(\sum_{j=1}^{K} v_{kj} \cdot y_{mj} - A_k(\boldsymbol{v}) \right) \\
&= \sum_{m=1}^{M} \delta_{k,z_m} \left(y_{mj} - \mathrm{e}^{v_{kj}} \right) . \qquad \text{(B.12)}
\end{aligned}$$

A similar result can be found for w_{ki}. Here, δ_{ij} is the Kronecker delta, which is one if $i = j$ and zero otherwise. Inserting this result into (B.11), setting the derivative to zero and rearranging the terms, we identify the optimal model parameters

$$\begin{aligned}
w_{ki}^* &= \log\langle \delta_{k,z_m} \cdot x_i(\hat{t}) \rangle_{\hat{p}} - \log\langle \delta_{k,z_m} \rangle_{\hat{p}} \\
v_{kj}^* &= \log\langle \delta_{k,z_m} \cdot y_j(\hat{t}) \rangle_{\hat{p}} - \log\langle \delta_{k,z_m} \rangle_{\hat{p}} .
\end{aligned} \qquad \text{(B.13)}$$

In the E-step the expectations $\langle \cdot \rangle_{\hat{p}}$ are evaluated. Note that the expectations are taken over the whole sequence of M output spikes. In the M-step the parameters are updated to their new values. An estimate of these expectations is computed by the network generating output spike sequences. A local minimum of (B.9) can be found by iteratively evaluating the E- and M-step.

We will now show that the STDP protocol introduced here converges stochastically to the same result as the EM updates (B.13). We will derive this results for the lateral weights v_{kj} only, since adaption for other parameters is straightforward. Including the reward mechanism, the weight update consists of two stochastic processes: the forward sampling and the stochastic decision for the rejection step (2.11). The updates are made for each output spike of the network and therefore they will always fluctuate. In our analysis we are interested in the equilibrium point of these fluctuations for some given target distribution $p^*(\boldsymbol{X})$. This can be expressed for the expected weight update using (2.7),(2.9) and (2.11), for which we get

$$\begin{aligned}
&\left\langle \int q(\boldsymbol{S}' \mid \boldsymbol{X}, \boldsymbol{\theta})\ \Delta v_{kj}(\hat{t})\, \mathrm{d}\boldsymbol{S}' \right\rangle_{p^*} = 0 \\
&\leftrightarrow \quad \left\langle \int q(\boldsymbol{S}' \mid \boldsymbol{X}, \boldsymbol{\theta}) \cdot \frac{r(\boldsymbol{S}')}{\langle r(\boldsymbol{S}') \rangle_{q(\boldsymbol{S}' \mid \boldsymbol{X}, \boldsymbol{\theta})}} \cdot \delta_{k,z_m} \cdot \left(e^{-v_{kj}} y_j(\hat{t}) - 1 \right) \mathrm{d}\boldsymbol{S}' \right\rangle_{p^*} = 0 ,
\end{aligned}$$

which by inserting equation (2.10) yields

$$\begin{aligned}
&\leftrightarrow \quad \Big\langle \int p\left(\boldsymbol{S}' \mid \boldsymbol{X}, \boldsymbol{\theta}\right) \cdot \delta_{k,z_m} \cdot \left(e^{-v_{kj}} y_j(\hat{t}) - 1\right) \mathrm{d}\boldsymbol{S}' \Big\rangle_{p^*} = 0 \\
&\leftrightarrow \quad \left\langle \, \delta_{k,z_m} \cdot e^{-v_{kj}} y_j(\hat{t}) - \delta_{k,z_m} \, \right\rangle_{\hat{p}} = 0 \\
&\leftrightarrow \quad v_{kj} = \log\!\left\langle \, \delta_{k,z_m} \cdot y_j(\hat{t}) \, \right\rangle_{\hat{p}} \; - \; \log\!\left\langle \, \delta_{k,z_m} \, \right\rangle_{\hat{p}} \, .
\end{aligned}$$

The last line is equivalent to the solution of the EM-algorithm (B.13).

B.4 Details to: A refined EM approximation using rejection sampling

Here we present additional details to the rejection sampling algorithm that was used throughout the numerical experiments. The algorithm requires to evaluate two quantities that evolve on different time scales. The synaptic weight updates need to be updated on each spike, whereas the importance weights (2.10) need to be tracked over a whole input sequence.

The importance weight over sequence $\boldsymbol{S}$ is given by (2.10) which can be verified by inserting equation (B.8) and (2.4) into (2.10). Using (2.3) we find that this quantity computes to

$$r\left(\boldsymbol{S}\right) = \exp\left(\sum_{m=1}^{M} i(\hat{t}_m) - \log p\left(\boldsymbol{x}_m \mid \boldsymbol{\theta}\right) - \log p\left(\boldsymbol{y}_m \mid \boldsymbol{\theta}\right)\right) \, , \tag{B.14}$$

where the marginals $p\left(\boldsymbol{x}_m \mid \boldsymbol{\theta}\right)$ and $p\left(\boldsymbol{y}_m \mid \boldsymbol{\theta}\right)$ are given by

$$\begin{aligned}
p\left(\boldsymbol{x}_m \mid \boldsymbol{\theta}\right) \; &= \; \sum_{l=1}^{K} \exp\left(\sum_{i=1}^{N} w_{li} \cdot x_{mi}\right) \, , \\
p\left(\boldsymbol{y}_m \mid \boldsymbol{\theta}\right) \; &= \; \sum_{l=1}^{K} \exp\left(\sum_{j=1}^{K} v_{lj} \cdot y_{mj}\right) \, .
\end{aligned} \tag{B.15}$$

The term $p\left(\boldsymbol{x}_m \mid \boldsymbol{\theta}\right)$ is arbitrary since it cancels out in the rejection sampling algorithm, but we found that (B.15) achieves better performance including the dependence on $p\left(\boldsymbol{x}_m \mid \boldsymbol{\theta}\right)$, when using the the rejection sampler with the simple tracking mechanism for c. The performance of the rejection sampling algorithm essentially depends on the variance of the importance weights. The lower this variance is, the more generated sequences will be accepted. Since the importance weights are only needed to compare the quality of different proposed hidden state trajectories, all fluctuations that depend on the feedforward weights and inputs only, can be discarded. Explicitly subtracting $p\left(\boldsymbol{x}_m \mid \boldsymbol{\theta}\right)$ allows to minimize the fluctuations injected by the feedforward synapses. This modification had a large

impact on reducing the number of rejected trajectories and therefore increased learning speed.

The likelihood of an input sequence $\boldsymbol{X}$ can be approximated using a set of L paths $\boldsymbol{S}_1 \ldots \boldsymbol{S}_L$ sampled from (2.5) given by

$$p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right) = \left\langle\, r(\boldsymbol{S}) \,\right\rangle_{q(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta})} \;\approx\; \frac{1}{L} \sum_{l=1}^{L} r\left(\boldsymbol{S}_l\right) := \mathcal{L}_L(\boldsymbol{X}) \,. \tag{B.16}$$

In the simplest approximation the expectation can be taken over a single path. Thus, we find that the sequence log-likelihood can be directly approximated by (B.14), i.e.

$$\log p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right) \;\approx\; \sum_{m=1}^{M} \log p\left(\boldsymbol{x}_m \mid s_{m-1}, \boldsymbol{\theta}\right) := \log \hat{\mathcal{L}}(\boldsymbol{X}) \,. \tag{B.17}$$

In the AGL experiments this simple approximate likelihood was used to distinguish between grammatical and non-grammatical sequences.

The probability of generating a sequence $\boldsymbol{S}$ through the state space is given by the proposal distribution $q\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right)$, which was defined in equation (2.5). The bias between this and the model distribution $p\left(\boldsymbol{S} \mid \boldsymbol{X}, \boldsymbol{\theta}\right)$, is given by the importance weight $r\left(\boldsymbol{S}\right)$, which we have derived earlier (2.10). This bias can be eliminated using rejection sampling, i.e. accepting the sampled sequences based on a stochastic decision proportional to the importance weight. In the neural network we implemented this using an eligibility trace of synaptic weight changes (Izhikevich, 2007). The weight updates were accumulated over the whole input sequence

$$\Delta \hat{w}_{ki} = \sum_{m=1}^{M} \Delta w_{ki}(\hat{t}_m) \quad \text{and} \quad \Delta \hat{v}_{kj} = \sum_{m=1}^{M} \Delta v_{kj}(\hat{t}_m) \,. \tag{B.18}$$

The synaptic weights can be learned by modulating the learning rate ξ when incorporating the synaptic weight changes (B.18) at the end of a sequence. The learning rate must be modulated according to the importance weights. In the simulations we used a stochastic binary decision, whether to accept or reject the sampled sequence

$$p\left(\xi = \xi_0 \mid \boldsymbol{S}\right) = c \cdot r\left(\boldsymbol{S}\right), \quad p\left(\xi = 0 \mid \boldsymbol{S}\right) = 1 - c \cdot r\left(\boldsymbol{S}\right) \,, \tag{B.19}$$

where ξ_0 is a constant learning rate and c is a constant that scales the average number of rejected samples. The probability of accepting a path $\boldsymbol{S}$ is directly proportional to the importance weights. Using this, we immediately find that the mean number of rejected samples $\hat{L}_{\boldsymbol{X}}$ for an input sequence $\boldsymbol{X}$ is inversely proportional to the sequence likelihood, i.e.

$$\hat{L}_{\boldsymbol{X}} = \frac{1}{c \cdot \left\langle\, r(\boldsymbol{S'}) \,\right\rangle_{q(\boldsymbol{S'} \mid \boldsymbol{X}, \boldsymbol{\theta})}} = \frac{1}{c \cdot p\left(\boldsymbol{X} \mid \boldsymbol{\theta}\right)} \,. \tag{B.20}$$

The average learning rate assigned to a sampled sequence $\boldsymbol{S}$ depends on the probability of sampling $\boldsymbol{S}$ from the proposal distribution and the number of times the sequence is resampled. Using this, (B.20) and (B.19) we find the expected learning rate associated with a state sequence $\boldsymbol{S}$ to be given by

$$\langle \xi \rangle_{q(\boldsymbol{S'} \mid \boldsymbol{X},\boldsymbol{\theta})} = \frac{\xi_0 \cdot r(\boldsymbol{S})}{\langle r(\boldsymbol{S'}) \rangle_{q(\boldsymbol{S'} \mid \boldsymbol{X},\boldsymbol{\theta})}} \,. \tag{B.21}$$

The constant c in (B.19) can be used to control the average number of rejected samples. We used a simple linear tracking algorithm for c in the logarithmic domain. Whenever a path was accepted $\log c$ was decreased by $L^* \cdot 10^{-4}$, if the path was rejected $\log c$ was increased by 10^{-4}. As learning proceeds the network converges to an equilibrium acceptance rate, determined by L^*. Throughout the experiments this parameter was tuned to achieve the desired mean number of samples over the whole training session. A quantitative comparison between the learning performances achieved with the batch algorithm and this tracking mechanism, is given in Fig. 2.8.

In the batch version of the algorithm a set of sampled paths with a fixed size L was used to compute c directly, which was chosen such that the distribution over the L paths was correctly normalized. Using (B.19) we find this to be fulfilled for

$$c = \frac{1}{\sum_{l=1}^{L} r\left(\boldsymbol{S}_l\right)} \,. \tag{B.22}$$

A sequence $\boldsymbol{S}$ was then chosen at random from the set of L sampled sequences. The importance sampler was realized by directly weighting the synaptic changes by the scalar value of the normalized importance weight, i.e. $\xi = \frac{\xi_0 \cdot r(\boldsymbol{S})}{\sum_{l=1}^{L} r(\boldsymbol{S}_l)}$.

B.5 Simulations and data analysis

All simulations were done in Matlab (Mathworks), directly implementing the derived equations without discrete time approximations. The population output rate $\hat{\nu}$ was tuned to give an average output rate of 5-20Hz per neuron. Prior to learning all weights were set to small equally distributed random values. The weight updates were incorporated using a constant learning rate $\xi = 0.005$.

Other than in the theoretical analysis where synaptic delays were neglected for the sake of simplicity, we used synaptic delays of 5ms for the lateral excitatory synapses in the numerical experiments. We also used a more realistic double exponential EPSP kernel of the form (Gerstner and Kistler, 2002)

$$\epsilon\left(s\right) = \left(\exp\left(-s/\tau_s\right) - \exp\left(-s/\tau_r\right)\right) \cdot \Theta(s) \,, \tag{B.23}$$

where $\tau_s = 2ms$ and $\tau_r = 20ms$ are the time constants of the falling and rising edges of the EPSP kernel, respectively. The above theoretical analysis applies equally to

this kernel, but would be slightly more complex since each of the two exponential decay terms comprises a piece of memory which has to be reflected in the network state s_m.

The diagonal of the weight matrix v_{kk} was set to zero and these weights were excluded from learning. Instead, a refractory mechanism was used with a kernel given by (Gerstner and Kistler, 2002)

$$\eta(s) = \eta_0 \cdot \exp\left(-s/\tau_{refr}\right) \ , \tag{B.24}$$

where $\eta_0 = 10$ is the maximum amplitude of the refractory kernel, $\tau_{refr} = 5ms$ is the refractory time constant and s is the time elapsed since the last output spike. Equation (B.24) was subtracted from the membrane potential (2.1).

B.6 Details to: Learning to predict spike sequences through STDP

The input patterns were generated by drawing for each afferent neuron and each pattern a value from the Beta distribution with parameters $\alpha = 0.2$, $\beta = 0.8$ and multiplying this value with the maximum rate of $75Hz$. Using these rate patterns, input spikes were then generated by creating independent spike events from a Poisson process.

To facilitate the interpretability of the network output, we applied a smoothing and sorting algorithm. The spike statistics were estimated using the perievent time histogram (PETH) on the network output (Luczak et al., 2009). The network output rates were computed for time bins of 1ms and then filtered with a Gaussian filter function ($\sigma = 10ms$) to give the smoothed single-trial estimated rates $\bar{\nu}_k(t)$. These spike histograms were averaged over 100 trial runs to give the time estimated rates $\nu_k^{\mathrm{PETH}}(t)$ for each neuron k. For neuron sorting we evaluated the point in time with the highest activity

$$t_k^* = \arg\max_t \nu_k^{\mathrm{PETH}}(t) \ . \tag{B.25}$$

This was used as criterion to determine the rank index of the output neurons for sorting. The PETHs for all sequences of a learning problem were concatenated before evaluating the maximum firing time (B.25) to ensure a visual separation between neurons that fired preferentially during one specific sequence. This neuron order was also used to sort the rows and columns of the synaptic weight matrix shown in Fig. 2.2 (neurons that fired on average less than one spike per sequence were excluded from this plot).

To quantify the similarity between spontaneous and evoked network activity we used Spearman's rank correlation (Luczak et al., 2009). The correlations were computed by evaluating the rank correlation between the PETH computed on a single spontaneous sequence and the evoked activity averaged over 100 trial runs. For the evaluation only the neurons that produced at least one spike during the

spontaneous run were used. The firing rates in Fig. 2.4A,B were estimated over 100 input sequences. Only the time window during which pattern a was present on the input was analyzed. Neurons that fired with average rates less than 10Hz during these time windows were excluded from the analysis. Neurons with rates above 10Hz for patterns appearing in one sequence, but not the other, were classified as context specific. Those that fired rates above 10Hz during both sequences were classified as context unspecific.

B.7 Details to: Mixed selectivity emerges in multiple interconnected WTA circuits

Here a linear classifier was used to identify separating planes in the network activity. We trained a soft-margin support vector machine with linear kernels (Cortes and Vapnik, 1995; Schölkopf et al., 1999; Bishop, 2006) to classify the network activity during the delay phase of sequence *AB-delay-ab* against that of *BA-delay-ba*. The resulting linear models were used to classify 50 test samples from each of the two sequences. Sequences that were at any point in time on the wrong side of the separating plane were reported as wrongly classified. The mean classification rates over these test samples were reported.

To illustrate the network state during the holding phase we used the dynamic PCA (jPCA) method in experiment 2. This method was recently introduced as an extension to normal PCA, with better applicability to dynamical data (Churchland et al., 2012). We applied this method on the smoothed network activities $\bar{v}_k(t)$ of all network neurons. The jPCA identifies the plane that is aligned with the fastest rotation in the data set. Briefly, the jPCA first uses a preprocessing step in which normal PCA is performed on the data to reduce the dimensionality. We used the first 6 PCA components as suggested in (Churchland et al., 2012). Subsequently a projection from the neural state to its slope is found. A skew-symmetric matrix is constructed that projects the PCA components into its first order derivatives. The solution to this constraint optimization problem is a matrix defining the best-fitting rotational linear dynamical system which can describe the data set (see the supplementary derivation of (Churchland et al., 2012) for details). The orthogonal basis of the jPCA is then given by the real plane associated with the eigenvectors with largest imaginary eigenvalues of this projection matrix. This plane is aligned with the fastest rotation in the data set.

B.8 Details to: Trajectories in network assemblies emerge for stationary input patterns

In this experiment we employed homeostatic mechanism to control the excitabilities b_k. A detailed derivation of this intrinsic plasticity was presented in (Habenschuss

et al., 2012). Following this approach we slowly regulated b_k over time to maximizes the entropy of the network output by demanding that the overall output rate of each neuron, measured over a long time window $T^{(H)}$, converges to the target rate $\frac{1}{K}\hat{\nu}$, i.e. $\int_0^{T^{(H)}} \nu_k(t)dt = \frac{\hat{\nu}}{K}$ for each neuron k. A stochastic approximation to that can be achieved by updating the excitabilities b_k in (2.1)

$$\Delta b_k = \begin{cases} \mu \cdot \left(\frac{1}{K} - 1\right) & \text{if neuron } k \text{ spikes} \\ \mu \cdot \frac{1}{K} & \text{else} \end{cases}, \tag{B.26}$$

where μ is an update rate we have chosen to be $\mu = 0.1$ in this experiment. This mechanism assures that all network neurons participate on average equally in the representation of the hidden state. If μ is chosen small enough this method assures that all network neurons participate equally in the representation of the hidden state (Habenschuss et al., 2012).

B.9 Details to: Learning the temporal structure of an artificial grammar model

Here we used two data sets from (Conway and Christiansen, 2005) - the 12 sequences reported there in appendix A for training and the 20 sequences from table 1 for testing. In each training iteration we randomly drew one example input sequence from the train set. For testing we created 100 legal and illegal sequences randomly drawn from the test set. The sequences were encoded using sparse input patterns, encoded by 10 input neurons, two of which fired with a rate of $100Hz$ for 50ms for each of the five input pattern, while the others remained silent. All spike patterns were not kept fixed but generated newly at each occurrence of the pattern and also during replay for rejection sampling. In the AGL experiments, the initial network state was reset to zero $\boldsymbol{x}_0 = \mathbf{0}$, $\boldsymbol{y}_0 = \mathbf{0}$ before a new input sequence was presented. To classify grammatical against non-grammatical sequences the one-sample approximation of the log-likelihood (B.17) was computed for all test sequences. A threshold was computed by taking the mean of these log-likelihoods. Sequences $\boldsymbol{S}$ for which $\log \hat{\mathcal{L}}(\boldsymbol{S})$ lied above this threshold were classified as grammatical, all others as non-grammatical.

B.10 Details to: Comparison of the convergence speed and performance of the approximate algorithms

In this experiment, random teacher HMMs were generated by drawing initial state, observation and transition probability tables from a Beta distribution with $\alpha = 0.2$ and $\beta = 0.8$ and then normalizing the tables to proper conditional probabilities. The models had $K = 5$ states and $N = 10$ discrete observations. These models

were then used to generate observation sequences. We drew a training set of 200 and a validation set of 2000 sequences of length $M = 25$. The complete training data set was repeatedly present to the network. We refer to the presentation of the whole batch of training sequences as an epoch. The weight updates for the WTA circuit were applied at the end of each training sequence. In each epoch all sequences were presented in random order. For the Baum-Welch algorithm (which is not an online algorithm) the updates were computed over all sequences in each epoch (batch learning). We generated 50 trials using 50 different teacher HMMs. The performance of rejection sampling was assessed for the two algorithms to evaluate the normalizing constant c – the exact version (B.22) and the linear tracking algorithm.

Appendix C

Appendix to Chapter 3: Network plasticity as Bayesian inference

C.1 Details to: Learning a posterior distribution through stochastic synaptic plasticity

Here we prove that $p^*(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\,\mathbf{x})$ is the unique stationary distribution of the parameter dynamics (3.3) that operate on the network parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)$. Convergence to this stationary distribution then follows for strictly positive $p^*(\boldsymbol{\theta})$. In fact, we prove here a more general result for parameter dynamics given by

$$\begin{aligned} d\theta_i \;=\; & \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta}) + T\, b'(\theta_i) \right) dt \\ & + \sqrt{2Tb(\theta_i)}\, d\mathcal{W}_i \end{aligned} \tag{C.1}$$

for $i = 1, \ldots, M$ and $b'(\theta_i) := \frac{\partial}{\partial \theta_i} b(\theta_i)$. This dynamics includes a temperature parameter T and a sampling-speed factor $b(\theta_i)$ that can in general depend on the current value of the parameter θ_i. The temperature parameter T can be used to scale the diffusion term (i.e., the noise). The sampling-speed factor controls the speed of sampling, i.e., how fast the parameter space is explored. It can be made dependent on the individual parameter value without changing the stationary distribution. For example, the sampling speed of a synaptic weight can be slowed down if it reaches very high or very low values. Note that the dynamics (3.3) is a special case of the dynamics (C.1) with unit temperature $T = 1$ and constant sampling speed $b(\theta_i) \equiv b$. We show that the stochastic dynamics (C.1) leaves the distribution

$$p^*(\boldsymbol{\theta}) \equiv \frac{1}{\mathcal{Z}} q^*(\boldsymbol{\theta}) \tag{C.2}$$

invariant, where $\mathcal{Z}$ is a normalizing constant $\mathcal{Z} = \int q^*(\boldsymbol{\theta})\, d\boldsymbol{\theta}$ and

$$q^*(\boldsymbol{\theta}) \;=\; p(\boldsymbol{\theta}\,|\,\mathbf{x})^{\frac{1}{T}}\,. \tag{C.3}$$

Note that the stationary distribution $p^*(\boldsymbol{\theta})$ is shaped by the temperature parameter T, in the sense that $p^*(\boldsymbol{\theta})$ is a flattened version of the posterior for high temperature.

The provided proof applies for standard Wiener processes $\mathcal{W}_i$, where process increments over time $t-s$ are normally distributed with zero mean and variance $t-s$:

$$\mathcal{W}_i^t - \mathcal{W}_i^s \;\sim\; \textsc{Normal}(0, t-s)\,, \tag{C.4}$$

where $\mathcal{W}_i^t$ denotes the value of an instantiation of the process at time t.

Using this we can formulate the following theorem:

Theorem 1. *Let $p(\mathbf{x}, \boldsymbol{\theta})$ be a strictly positive, continuous probability distribution over continuous or discrete states $\mathbf{x} = \boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$ and continuous parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)$, twice continuously differentiable with respect to $\boldsymbol{\theta}$. Let $b(\theta)$ be a strictly positive, twice continuously differentiable function. Then the set of stochastic differential equations* (C.1) *leaves the distribution $p^*(\boldsymbol{\theta})$ invariant. Furthermore, $p^*(\boldsymbol{\theta})$ is the unique stationary distribution of the sampling dynamics.*

Proof of Theorem 1

First, note that the first two terms in the drift term of Eq. (C.1) can be written as

$$\begin{aligned}
&b(\theta_i)\frac{\partial}{\partial\theta_i}\log p_{\mathcal{S}}(\boldsymbol{\theta}) + b(\theta_i)\frac{\partial}{\partial\theta_i}\log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta}) \\
&\quad= b(\theta_i)\frac{\partial}{\partial\theta_i}\log(p_{\mathcal{S}}(\boldsymbol{\theta})p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta})) \\
&\quad= b(\theta_i)\frac{\partial}{\partial\theta_i}\log p(\mathbf{x},\boldsymbol{\theta}) \\
&\quad= b(\theta_i)\frac{\partial}{\partial\theta_i}\log(p(\theta_i|\mathbf{x},\boldsymbol{\theta}_{\setminus i})p(\mathbf{x},\boldsymbol{\theta}_{\setminus i})) \\
&\quad= b(\theta_i)\left(\frac{\partial}{\partial\theta_i}\log(p(\theta_i|\mathbf{x},\boldsymbol{\theta}_{\setminus i}) + \frac{\partial}{\partial\theta_i}\log p(\mathbf{x},\boldsymbol{\theta}_{\setminus i}))\right) \\
&\quad= b(\theta_i)\frac{\partial}{\partial\theta_i}\log(p(\theta_i|\mathbf{x},\boldsymbol{\theta}_{\setminus i})\,,
\end{aligned}$$

where $\boldsymbol{\theta}_{\setminus i}$ denotes the vector of parameters excluding parameter θ_i. Hence, the dynamics (C.1) can be written as

$$d\theta_i = \left(b(\theta_i)\frac{\partial}{\partial\theta_i}\log p(\theta_i|\mathbf{x},\boldsymbol{\theta}_{\setminus i}) + T\,b'(\theta_i)\right)dt + \sqrt{2Tb(\theta_i)}\,d\mathcal{W}_i \tag{C.5}$$

(for $i = 1, \ldots, M$). Eq. (C.5) has drift $A_k(\boldsymbol{\theta})$ and diffusion $B_{ik}(\boldsymbol{\theta})$:

$$\begin{aligned}
A_k(\boldsymbol{\theta}) &= b(\theta_i)\frac{\partial}{\partial\theta_i}\log p(\theta_i|\mathbf{x},\boldsymbol{\theta}_{\setminus i}) + T\,b'(\theta_i)\,, \\
B_{ii}(\boldsymbol{\theta}) &= 2T\,b(\theta_i)\,, \\
B_{ik}(\boldsymbol{\theta}) &= 0 \quad \text{for } i \neq k\,.
\end{aligned} \tag{C.6}$$

Hence, the Itô stochastic differential equations (C.5) translate into the following Fokker-Planck equation,

$$\frac{d}{dt}p_{\mathrm{FP}}(\boldsymbol{\theta},t) = \sum_i -\frac{\partial}{\partial\theta_i}\left(\left(b(\theta_i)\,\frac{\partial}{\partial\theta_i}\log p(\theta_i|\mathbf{x},\boldsymbol{\theta}_{\setminus i}) + T\,b'(\theta_i)\right)p_{\mathrm{FP}}(\boldsymbol{\theta},t)\right) + \frac{\partial^2}{\partial\theta_i^2}\left(T\,b(\theta_i)p_{\mathrm{FP}}(\boldsymbol{\theta},t)\right), \tag{C.7}$$

where $p_{\mathrm{FP}}(\boldsymbol{\theta},t)$ denotes the distribution over network parameters at time t. Plugging in the presumed stationary distribution $p^*(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}}q^*(\boldsymbol{\theta})$ on the right hand side of Eq. (C.7), one obtains

$$\begin{aligned}
\frac{d}{dt}p_{\mathrm{FP}}(\boldsymbol{\theta},t) &= \sum_i -\frac{\partial}{\partial\theta_i}\left(\left(b(\theta_i)\,\frac{\partial}{\partial\theta_i}\log p(\theta_i|\mathbf{x},\boldsymbol{\theta}_{\setminus i}) + T\,b'(\theta_i)\right)\frac{q^*(\boldsymbol{\theta})}{\mathcal{Z}}\right) \\
&\quad + \frac{\partial^2}{\partial\theta_i^2}\left(T\,b(\theta_i)\frac{q^*(\boldsymbol{\theta})}{\mathcal{Z}}\right) \\
&= \frac{1}{\mathcal{Z}}\left[\sum_i -\frac{\partial}{\partial\theta_i}\left(\left(b(\theta_i)\,\frac{\partial}{\partial\theta_i}\log p(\theta_i|\mathbf{x},\boldsymbol{\theta}_{\setminus i}) + T\,b'(\theta_i)\right)q^*(\boldsymbol{\theta})\right)\right. \\
&\quad \left. + \frac{\partial}{\partial\theta_i}\left(T\,b'(\theta_i)q^*(\boldsymbol{\theta}) + T\,b(\theta_i)\frac{\partial}{\partial\theta_i}q^*(\boldsymbol{\theta})\right)\right] \\
&= \frac{1}{\mathcal{Z}}\left[\sum_i -\frac{\partial}{\partial\theta_i}\left(b(\theta_i)\,\frac{\partial}{\partial\theta_i}\log p(\theta_i|\mathbf{x},\boldsymbol{\theta}_{\setminus i})\,q^*(\boldsymbol{\theta})\right)\right. \\
&\quad \left. + \frac{\partial}{\partial\theta_i}\left(T\,b(\theta_i)\frac{\partial}{\partial\theta_i}q^*(\boldsymbol{\theta})\right)\right] \\
&= \frac{1}{\mathcal{Z}}\left[\sum_i -\frac{\partial}{\partial\theta_i}\left(b(\theta_i)\,\frac{\partial}{\partial\theta_i}\log p(\theta_i|\mathbf{x},\boldsymbol{\theta}_{\setminus i})\,q^*(\boldsymbol{\theta})\right)\right. \\
&\quad \left. + \frac{\partial}{\partial\theta_i}\left(T\,b(\theta_i)\,q^*(\boldsymbol{\theta})\,\frac{\partial}{\partial\theta_i}\log q^*(\boldsymbol{\theta})\right)\right],
\end{aligned}$$

which by inserting $q^*(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \mid \mathbf{x})^{\frac{1}{T}}$ becomes

$$\begin{aligned}
\frac{d}{dt} p_{\mathrm{FP}}(\boldsymbol{\theta}, t) &= \frac{1}{\mathcal{Z}} \Bigg[\sum_i -\frac{\partial}{\partial \theta_i} \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p(\theta_i | \mathbf{x}, \boldsymbol{\theta}_{\setminus i}) \, q^*(\boldsymbol{\theta}) \right) \\
&\qquad + \frac{\partial}{\partial \theta_i} \left(T \, b(\theta_i) \, q^*(\boldsymbol{\theta}) \, \frac{1}{T} \frac{\partial}{\partial \theta_i} \log p(\boldsymbol{\theta} | \mathbf{x}) \right) \Bigg] \\
&= \frac{1}{\mathcal{Z}} \Bigg[\sum_i -\frac{\partial}{\partial \theta_i} \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p(\theta_i | \mathbf{x}, \boldsymbol{\theta}_{\setminus i}) \, q^*(\boldsymbol{\theta}) \right) \\
&\qquad + \frac{\partial}{\partial \theta_i} \left(b(\theta_i) \, q^*(\boldsymbol{\theta}) \, \frac{\partial}{\partial \theta_i} [\log p(\boldsymbol{\theta}_{\setminus i} | \mathbf{x}) + \log p(\theta_i | \mathbf{x}, \boldsymbol{\theta}_{\setminus i})] \right) \Bigg] \\
&= \frac{1}{\mathcal{Z}} \Bigg[\sum_i -\frac{\partial}{\partial \theta_i} \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p(\theta_i | \mathbf{x}, \boldsymbol{\theta}_{\setminus i}) \, q^*(\boldsymbol{\theta}) \right) \\
&\qquad + \frac{\partial}{\partial \theta_i} \left(b(\theta_i) \, q^*(\boldsymbol{\theta}) \, \frac{\partial}{\partial \theta_i} \log p(\theta_i | \mathbf{x}, \boldsymbol{\theta}_{\setminus i}) \right) \Bigg] \\
&= \sum_i 0 = 0 \ .
\end{aligned}$$

This proves that $p^*(\boldsymbol{\theta})$ is a stationary distribution of the parameter sampling dynamics (C.5). Under the assumption that $b(\theta_i)$ is strictly positive, this stationary distribution is also unique. If the matrix of diffusion coefficients is invertible, and the potential conditions are satisfied, the stationary distribution can be obtained (uniquely) by simple integration. Since the matrix of diffusion coefficients is diagonal in our model, the diffusion coefficient matrix is trivially invertible if all diagonal elements, i.e. all $b(\theta_i)$, are positive. Also the potential conditions are fulfilled (by design), as can be verified by substituting (C.6) into Equation (5.3.22) in (Gardiner, 2004),

$$\begin{aligned}
Z_i(\boldsymbol{\theta}) &= B_{ii}^{-1}(\boldsymbol{\theta}) \left(2A_i(\boldsymbol{\theta}) - \frac{\partial}{\partial \theta_i} B_{ii}(\boldsymbol{\theta}) \right) \\
&= \frac{1}{2Tb(\theta_i)} \left(2b(\theta_i) \frac{\partial}{\partial \theta_i} \log p(\theta_i | \mathbf{x}, \boldsymbol{\theta}_{\setminus i}) + 2T \, b'(\theta_i) - 2T \, b'(\theta_i) \right) \\
&= \frac{1}{T} \frac{\partial}{\partial \theta_i} \log p(\theta_i | \mathbf{x}, \boldsymbol{\theta}_{\setminus i}) \ ,
\end{aligned}$$

and by using that the normalization constant $\mathcal{Z}$ is independent of θ_i we can write

$$\begin{aligned}
Z_i(\boldsymbol{\theta}) = \frac{1}{T} \frac{\partial}{\partial \theta_i} \log p(\theta_i | \mathbf{x}, \boldsymbol{\theta}_{\setminus i}) &= \frac{1}{T} \frac{\partial}{\partial \theta_i} \left(\log p(\theta_i | \mathbf{x}, \boldsymbol{\theta}_{\setminus i}) + \log p(\boldsymbol{\theta}_{\setminus i} | \mathbf{x}) - \log \mathcal{Z}^T \right) \\
&= \frac{1}{T} \frac{\partial}{\partial \theta_i} \log \frac{p(\boldsymbol{\theta} | \mathbf{x})}{\mathcal{Z}^T} \\
&= \frac{\partial}{\partial \theta_i} \log \frac{p(\boldsymbol{\theta} | \mathbf{x})^{1/T}}{\mathcal{Z}} = \frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta}) \ .
\end{aligned}$$

This shows that $\boldsymbol{Z}(\boldsymbol{\theta}) = (Z_1(\boldsymbol{\theta}), \ldots, Z_M(\boldsymbol{\theta}))$ is a gradient. Thus, the potential conditions are met and the stationary distribution is unique, q.e.d.

For strictly positive $b(\theta)$, the diffusion matrix B (Eq. (C.6)) is positive definite. Convergence to the stationary distribution follows then directly for strictly positive $p^*(\boldsymbol{\theta})$ (see Section 3.7.2 in (Gardiner, 2004)).

Online approximation of the batch learning rule

We show here that the rule (3.5) is a reasonable approximation to the batch-rule (3.3). According to the dynamics (C.1), synaptic plasticity rules that implement synaptic sampling have to compute the log likelihood derivative $\frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta})$. We assume that every τ_x time units a different input $\boldsymbol{x}^n$ is presented to the network. For simplicity, assume that $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$ are visited in a fixed regular order. Under the assumption that input patterns are drawn independently, the likelihood of the generative model factorizes

$$p_{\mathcal{N}}(\mathbf{x}, |\boldsymbol{\theta}) = \prod_{n=1}^{N} p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}) \; . \tag{C.8}$$

The derivative of the log likelihood is then given by

$$\frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}) \; . \tag{C.9}$$

Using Eq. (C.9) in the dynamics (C.1), one obtains

$$\begin{aligned} d\theta_i \; = \; & \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + b(\theta_i) \sum_{n=1}^{N} \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}) \; + \; T \, b'(\theta_i) \right) \, dt \\ & + \; \sqrt{2Tb(\theta_i)} \, d\mathcal{W}_i \; . \end{aligned} \tag{C.10}$$

Hence, the parameter dynamics depends at any time on all network inputs and network responses.

This "batch" dynamics does not map readily onto a network implementation because the weight update requires at any time knowledge of all inputs $\boldsymbol{x}^n$. We provide here an online approximation for small sampling speeds. To obtain an online learning rule, we consider the parameter dynamics

$$\begin{aligned} d\theta_i \; = \; & \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + N b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}) + T \, b'(\theta_i) \right) \, dt \\ & + \; \sqrt{2Tb(\theta_i)} \, d\mathcal{W}_i \; . \end{aligned} \tag{C.11}$$

As in the batch learning setting, we assume that each input $\boldsymbol{x}^n$ is presented for a time interval of τ_x. Integrating the parameter changes (C.11) over one full presentation

of the data $\mathbf{x}$, i.e., starting from $t = 0$ with some initial parameter values $\boldsymbol{\theta}(0)$ up to time $t = N\tau_x$, we obtain for slow sampling speeds ($N\tau_x b(\theta_i) \ll 1$)

$$\theta_i(N\tau_x) - \theta_i(0) \approx N\tau_x \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + b(\theta_i) \sum_{n=1}^{N} \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}) + T\, b'(\theta_i) \right) + \sqrt{2Tb(\theta_i)}\, (\mathcal{W}_i^{N\tau_x} - \mathcal{W}_i^0) \,.$$

This is also what one obtains when integrating Eq. (C.10) for $N\tau_x$ time units (for slow $b(\theta_i)$). Hence, for slow enough $b(\theta_i)$, Eq. (C.11) is a good approximation of optimal weight sampling. The update rule (3.5) follows from (C.11) for $T = 1$ and $b(\theta_i) \equiv b$.

Discrete time approximation

Here we provide the derivation for the approximate discrete time learning rule (3.7). For a discrete time parameter update at time t with discrete time step Δt during which $\boldsymbol{x}^n$ is presented, a corresponding rule can be obtained by short integration of the continuous time rule (C.11) over the time interval from t to $t + \Delta t$:

$$\begin{aligned} \Delta\theta_i &= \Delta t \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + Nb(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}) + T\, b'(\theta_i) \right) \\ &\quad + \sqrt{2Tb(\theta_i)}\, (\mathcal{W}_i^{t+\Delta t} - \mathcal{W}_i^t) \\ &= \Delta t \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + Nb(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{\theta}) + T\, b'(\theta_i) \right) \\ &\quad + \sqrt{2T\Delta t\, b(\theta_i)}\, \nu_i^t \,, \end{aligned} \tag{C.12}$$

where ν_i^t denotes Gaussian noise $\nu_i^t \sim \text{Normal}(0, 1)$. The update rule (3.7) is obtained by choosing a constant $b(\theta) \equiv b$, $T = 1$, and defining $\eta = \Delta t\, b$.

Synaptic sampling with hidden states

When there is a direct relationship between network parameters $\boldsymbol{\theta}$ and the distribution over input patterns $\boldsymbol{x}^n$, the parameter dynamics can directly be derived from the derivative of the data log likelihood and the derivative of the parameter prior. Typically however, generative models for brain computation assume that the network response $\boldsymbol{z}^n$ to input pattern $\boldsymbol{x}^n$ represents in some manner the value of hidden variables that explain the current input pattern. In the presence of hidden variables, maximum likelihood learning cannot be applied directly, since the state of the hidden variables is not known from the observed data. The expectation maximization algorithm (Bishop, 2006) can be used to overcome this problem. We adopt this approach here. In the online setting, when pattern $\boldsymbol{x}^n$ is applied to the

network, it responds with network state $\boldsymbol{z}^n$ according to $p_{\mathcal{N}}(\boldsymbol{z}|\boldsymbol{x}^n, \boldsymbol{\theta})$, where the current network parameters are used in this inference process. The parameters are updated in parallel according to the dynamics

$$\begin{aligned} d\theta_i \;=\; & \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + N b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\boldsymbol{x}^n, \boldsymbol{z}^n | \boldsymbol{\theta}) \;+\; T\, b'(\theta_i) \right) dt \\ & +\; \sqrt{2T b(\theta_i)}\, d\mathcal{W}_i \,. \end{aligned} \tag{C.13}$$

Note that in comparison with the dynamics (C.11), the likelihood term now also contains the current network response $\boldsymbol{z}^n$. It can be shown that this dynamics leaves the stationary distribution

$$p^*(\boldsymbol{\theta}) \equiv \frac{1}{\mathcal{Z}} p(\boldsymbol{\theta} \,|\, \mathbf{x}, \mathbf{z})^{\frac{1}{T}} \,, \tag{C.14}$$

invariant, where $\mathcal{Z}$ is again a normalizing constant (the dynamics (C.13) is again the online-approximation). Hence, in this setup, the network samples concurrently from circuit states (given $\boldsymbol{\theta}$) and network parameters (given the network state $\boldsymbol{z}^n$), which can be seen as a sampling-based version of online expectation maximization.

C.2 Details to: Figure 3.1

For the example likelihood function in Fig. 3.1A we used a mixture of Gaussian distributions, of the form

$$p_{\mathcal{N}}(\boldsymbol{x} \mid \boldsymbol{\theta}) \;=\; p_{\mathcal{N}}(\boldsymbol{x} \mid \theta_1)\; p_{\mathcal{N}}(\boldsymbol{x} \mid \theta_2) \,, \qquad \text{with} \tag{C.15}$$

$$p_{\mathcal{N}}(\boldsymbol{x} \mid \theta) \;=\; c\, \textsc{Normal}\left(\theta \;\middle|\; \mu_1, \sigma_1^2\right) \;+\; (1-c)\, \textsc{Normal}\left(\theta \;\middle|\; \mu_2, \sigma_2^2\right) \,, \tag{C.16}$$

$$\text{and} \qquad \textsc{Normal}\left(\theta \;\middle|\; \mu, \sigma^2\right) \;\propto\; \exp\left(-\frac{1}{2\sigma^2} (\theta - \mu)^2 \right) \,, \tag{C.17}$$

where $\mu_1 = 0.3$, $\mu_2 = 0.9$, $\sigma_1 = 0.1$, $\sigma_2 = 0.2$ and $c = 0.3$. In Fig. 3.1D we used a prior $p_{\mathcal{S}}(\boldsymbol{\theta}) = p_{\mathcal{S}}(\theta_1) p_{\mathcal{S}}(\theta_2)$, with $p_{\mathcal{S}}(\theta_i)$ given by a normal distribution ($\mu = 0.3$, $\sigma = 0.35$). A learning rate of $\eta = 0.005$ was used to sampled trajectories which had a length of 50 and 300 time steps in Fig. 3.1C and F, respectively. In Fig. 3.1F the time-discrete version of the synaptic sampling algorithm (3.7) was used, with $N = T = 1$. In Fig. 3.1C the same dynamics were used, but the diffusion term and the contribution of the prior $\frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta})$ were set to zero.

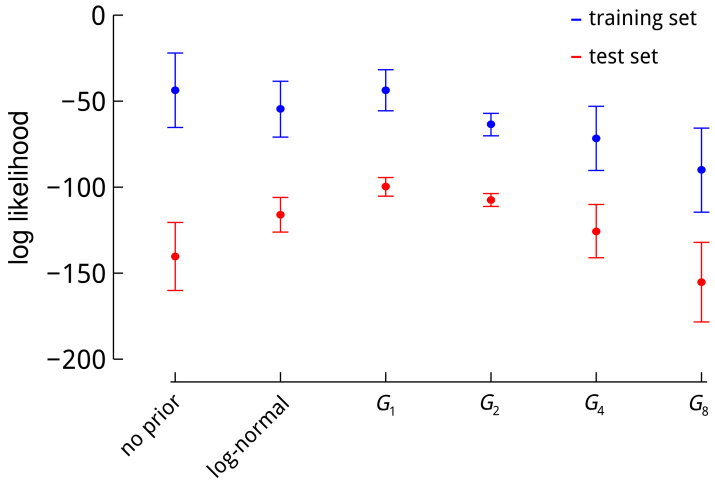

Fig. C.1: Comparison of the learning performance for different priors. Learning performances are shown for the training set (blue) and the test set (red). Dots represent average values of the log likelihood after 10000 training steps. The average values were computed based on 100 individual trial runs for each prior distribution. The error bars indicate STD.

C.3 Details to: Improving the generalization capability of a neural network through synaptic sampling

Restricted Boltzmann machine (RBM)

For learning the distribution over different writings of digit *1* with different priors in Fig. 3.2, a restricted Boltzmann machine (RBM) with 748 visible and 9 hidden neurons was used.

A RBM consists of two layers of neurons, the visible layer $\mathbf{x}$, and the hidden layer $\mathbf{z}$. Synaptic connections are formed only between neurons on different layers (Fig. 3.2A). Weights of synaptic connections are assumed to be symmetric, i.e., the weight value w_{ij} denotes both the weight of the connection from visible neuron x_j to hidden neuron z_i and the weight of the connection from hidden neuron z_i to visible neuron x_j. Neurons in these layers are stochastic non-spiking neurons with binary output. For given outputs $\boldsymbol{x}$ of visible neurons, each neuron z_i in the hidden layer computes its output in a stochastic manner

$$z_i = \begin{cases} 1 & , \quad \text{with probability } \sigma(\sum_j w_{ij} x_j + b_i^{\text{hid}}) \\ 0 & , \quad \text{with probability } 1 - \sigma(\sum_j w_{ij} x_j + b_i^{\text{hid}}) \end{cases} , \tag{C.18}$$

where b_i^{hid} is the bias of hidden neuron z_i. Analogously, for given outputs of hidden neurons $\boldsymbol{z}$, each neuron z_i in the visible layer computes its stochastic output according to

$$x_i = \begin{cases} 1 & , \quad \text{with probability } \sigma(\sum_j w_{ji} z_j + b_i^{\text{vis}}) \\ 0 & , \quad \text{with probability } 1 - \sigma(\sum_j w_{ji} z_j + b_i^{\text{vis}}) \end{cases} , \tag{C.19}$$

where b_i^{vis} is the bias of visible neuron x_i. Typically, for given outputs of hidden neurons $\boldsymbol{z}$, the output of the whole visible layer is sampled. In total, the parameter vector $\boldsymbol{\theta}$ for the RBM consists of all weight w_{ij} and all biases b_i^{hid}, b_i^{vis}.

Evaluation of model log likelihood

The (non-normalized) log likelihood $\hat{\mathcal{L}}(\mathbf{x} \,|\, \boldsymbol{\theta})$ measure in Fig. 3.2D,F was computed according to the assumed underlying model, given by the Boltzmann distribution. For a dataset $\mathbf{x} = \boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$, we get

$$\hat{\mathcal{L}}(\mathbf{x} \,|\, \boldsymbol{\theta}) \;=\; \log \left(\sum_{n=1}^{N} \sum_{\boldsymbol{z}} \exp \left(\sum_i \sum_j w_{ij}\, x_j^n\, z_i \;+\; b_i^{\text{hid}} z_i \;+\; b_j^{\text{vis}} x_j^n \right) \right) , \tag{C.20}$$

where the sum $\sum_{\boldsymbol{z}}$ runs over all possible states of the hidden neurons. This quantity is equivalent to the exact log likelihood up to a normalizing constant $\mathcal{Z}$, i.e. $\hat{\mathcal{L}}(\mathbf{x} \,|\, \boldsymbol{\theta}) = \log p(\mathbf{x} \,|\, \boldsymbol{\theta}) + \mathcal{Z}$. The current sets of network weights and biases were recorded after every 100 update steps and the log likelihood was evaluated for these parameter values. The plots in Fig. 3.2C-F show linear interpolations between these values. The likelihood plots in Fig. 3.2D,F show mean and std over 100 individually trained RBMs, all trained and evaluated on the same training and test set.

Comparison of the learning performance under different prior distributions

Here we compare the learning performance and generalization capabilities of the Boltzmann machine under different prior distributions. In addition to the uninformative (i.e., uniform) prior on weights we used a set of factorized priors for individual weights $p_{\mathcal{S}}(\mathbf{w}) = \prod_{i,j} p_{\mathcal{S}}(w_{ij})$. Here we used a scaled version of the mixture of two Gaussians, given by

$$\begin{aligned} G_k \;:=\; p_{\mathcal{S}}(w_{ij}) \;&=\; 0.5\, \textsc{Normal}\left(w_{ij} \;\middle|\; k \cdot \mu_1, k \cdot \sigma_1\right) \\ &+\; 0.5\, \textsc{Normal}\left(w_{ij} \;\middle|\; k \cdot \mu_2, k \cdot \sigma_2\right) \end{aligned} , \tag{C.21}$$

with means $\mu_1 = 1.0$, $\mu_2 = 0.0$, and standard deviations $\sigma_1 = \sigma_2 = 0.15$. Therefore, G_k denotes a scaled version of the prior used in Fig. 3.2E, with G_1 being identical to equation (C.26). In addition we used a log-normal prior with location $\mu = 0$ and scale $\sigma = 1.2$ of the form

$$p_{\mathcal{S}}(w_{ij}) \;=\; \frac{1}{w_{ij}\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2} (\log w_{ij} - \mu + \theta_0)^2 \right) . \tag{C.22}$$

Using these prior distributions we repeated the experiment in Sec. 3.3. A comparison of the performance of the Boltzmann machine for different prior distributions and for the learning scenario without prior is provided in Fig. C.1. The choice

of the prior can have a significant impact on the learning performance and with respect to overfitting. Nevertheless, we found that most prior distributions had a positive impact on the performance on the test set, thereby decreasing overfitting effects. However, the extreme case of the prior G_8 shows that a bad choice for the prior can result in performance that is worse than without a prior.

Network inputs

Handwritten digit images were taken from the MNIST dataset (LeCun et al., 1998). In MNIST, each instance of a handwritten digit is represented by a 784-dimensional vector $\boldsymbol{x}^n$. Each entry is given by the gray-scale value of a pixel in the 28×28 pixel image of the handwritten digit. The pixel values were scaled to the interval $[0, 1]$. In the RBM, each pixel was represented by a single visible neuron. When an input was presented to the network, the output of a visible neuron was set to 1 with probability as given by the scaled gray-scale value of the corresponding pixel.

Learning procedure

In each parameter update step the contrastive divergence algorithm of (G. E. Hinton, 2002) was used to estimate the likelihood gradients. Therefore, each update step consisted of a "wake" phase, a "reconstruction" phase, and the update of the parameters. The "wake" samples were generated by setting the outputs of the visible neurons to the values of a randomly chosen digit $\boldsymbol{x}^n$ from the training set and drawing the outputs z_i^n of all hidden layer neurons for the given visible output. The "reconstruction" activities $\hat{x}_j^n$ and $\hat{z}_i^n$ were generated by starting from this state of the hidden neurons and then drawing outputs of all visible neurons. After that, the hidden neurons were again updated and so on. In this way we performed five cycles of alternating visible and hidden neuron updates. The outputs of the network neurons after the fifth cycle were taken as the resulting "reconstruction" samples $\hat{x}_j^n$ and $\hat{z}_i^n$ and used for the parameter updates (C.23)–(C.25) given below. This update of parameters concluded one update step.

Log likelihood derivatives for the biases b_i^{hid} of hidden neurons are approximated in the contrastive divergence algorithm (G. E. Hinton, 2002) as $\frac{\partial}{\partial b_i^{\text{hid}}} \log p_{\mathcal{N}}(\boldsymbol{x}^n, \boldsymbol{z}^n|\theta) \approx z_i^n - \hat{z}_i^n$ (the derivatives for visible biases b_j^{vis} are analogous). Using Eq. (3.7), the synaptic sampling update rules for the biases are thus given by

$$\Delta b_i^{\text{hid}} = \eta\, N\, (z_i^n - \hat{z}_i^n) \,+\, \sqrt{2\eta}\, \nu_i^t \,, \tag{C.23}$$

$$\Delta b_j^{\text{vis}} = \eta\, N \left(x_j^n - \hat{x}_j^n\right) \,+\, \sqrt{2\eta}\, \nu_j^t \,. \tag{C.24}$$

Note that the parameter prior does not show up in these equations since no priors were used for the biases in our experiments. Contrastive divergence approximates

the log likelihood derivatives for the weights w_{ij} as $\frac{\partial}{\partial w_{ij}} \log p_{\mathcal{N}}(\boldsymbol{x}^n, \boldsymbol{z}^n|\theta) \approx z_i^n x_j^n - \hat{z}_i^n \hat{x}_j^n$. This leads to the synaptic sampling rule

$$\Delta w_{ij} = \eta \left(\frac{\partial}{\partial w_{ij}} \log p_{\mathcal{S}}(\mathbf{w}) + N \left(z_i^n x_j^n - \hat{z}_i^n \hat{x}_j^n \right) \right) + \sqrt{2\eta}\, \nu_{ij}^t \,. \tag{C.25}$$

In the simulations, we used this rule with $\eta = 10^{-4}$ and $N = 100$. Learning started from random initial parameters drawn from a Gaussian distribution with standard deviation 0.25 and means at 0 and -1 for weights w_{ij} and biases (b_i^{hid}, b_j^{vis}), respectively.

To compare learning with and without parameter priors, we performed simulations with an uninformative (i.e., uniform) prior on weights (Fig. 3.2C,D), which was implemented by setting $\frac{\partial}{\partial w_{ij}} \log p_{\mathcal{S}}(\mathbf{w})$ to zero. In simulations with a parameter prior (Fig. 3.2E,F), we used a local prior for each weight in order to obtain local plasticity rules. In other words, the prior $p_{\mathcal{S}}(\mathbf{w})$ was assumed to factorize into priors for individual weights $p_{\mathcal{S}}(\mathbf{w}) = \prod_{i,j} p_{\mathcal{S}}(w_{ij})$. For each individual weight prior, we used a bimodal distribution implemented by a mixture of two Gaussians

$$p_{\mathcal{S}}(w_{ij}) \;=\; 0.5\,\textsc{Normal}\left(w_{ij} \mid \mu_1, \sigma_1\right) \;+\; 0.5\,\textsc{Normal}\left(w_{ij} \mid \mu_2, \sigma_2\right) \,, \tag{C.26}$$

with means $\mu_1 = 1.0$, $\mu_2 = 0.0$, and standard deviations $\sigma_1 = \sigma_2 = 0.15$.

C.4 Details to: Spine motility as synaptic sampling

Here we derive the synaptic sampling model for spine motility given in Eq. (3.10), of the main text. The theory applies for mapping functions $f : \mathbb{R} \to \mathbb{R}$ which are continuous, strictly monotonic and twice differentiable, such that they uniquely map values from θ_i to w_i, i.e. $w_i = f(\theta)$. One example is the exponential mapping (3.9) provided in the main text. Further we define $f(\boldsymbol{\theta}) = (f(\theta_1), \ldots, f(\theta_M))$. Let $\boldsymbol{w} = f(\boldsymbol{\theta})$ and thus $p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{w}) = p_{\mathcal{N}}(\mathbf{x}|f(\boldsymbol{\theta}))$. Then the synaptic sampling dynamics, with prior $p_{\mathcal{S}}(\boldsymbol{\theta}) = \prod_i p_{\mathcal{S}}(\theta_i)$ and likelihood $p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{w})$, given by eq. (C.1) can be rewritten in the from

$$\begin{aligned} d\theta_i \;=\; & \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) \;+\; b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{w}) \;+\; T\, b'(\theta_i) \right) dt \\ & + \sqrt{2T\, b(\theta_i)}\, d\mathcal{W}_i \end{aligned} \tag{C.27}$$

$$\begin{aligned} = & \left(b(\theta_i) \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) \;+\; b(\theta_i)\, f'(\theta) \frac{\partial}{\partial w_i} \log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{w}) \;+\; T\, b'(\theta_i) \right) dt \\ & + \sqrt{2T\, b(\theta_i)}\, d\mathcal{W}_i \,. \end{aligned} \tag{C.28}$$

Thus, for the parameter dynamics an additional term $f'(\theta_i) = \frac{\partial}{\partial \theta_i} f(\theta_i)$ has to be taken into account that scales the effect of spike-triggered weight changes.

For the particular choice of the exponential mapping (3.9) this term evaluates to $f'(\theta_i) = \exp(\theta_i - \theta_0)$. Inserting this and using the simplifying choices of $b(\theta_i) = b$ and $T = 1$, we get

$$d\theta_i = b\left(\frac{\partial}{\partial\theta_i}\log p_{\mathcal{S}}(\boldsymbol{\theta}) \;+\; N\exp(\theta_i-\theta_0)\,\frac{\partial}{\partial w_i}\log p_{\mathcal{N}}(\boldsymbol{x}^n|\boldsymbol{w})\right)dt \;+\; \sqrt{2b}\,d\mathcal{W}_i\;,$$

which is the result (3.10).

Resulting log-normal priors over synaptic weights

Throughout all simulations of the spiking WTA circuits we used independent Gaussian priors, $p_{\mathcal{S}}(\boldsymbol{\theta}) = \prod_i \textsc{Normal}(\theta_i \mid \mu, \sigma^2)$ for the synaptic parameters θ_i. We show here that this choice for the prior $p_{\mathcal{S}}(\boldsymbol{\theta})$ together with $b(\theta_i) = b$ and the exponential parameter mapping $w_i = \exp(\theta_i - \theta_0)$ induces a log-normal prior distribution over the synaptic efficacies w_i, parametrized by μ, θ_0 and σ, given by

$$\hat{p}_{\mathcal{S}}(w_i) = \frac{1}{w_i\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2\sigma^2}(\log w_i - \mu + \theta_0)^2\right)\;. \tag{C.29}$$

First, we derive this result for general mapping functions $f(\cdot)$, which can be formalized in the following way: For every $f(\boldsymbol{\theta})$, $b(\theta_i)$ and $p_{\mathcal{S}}(\boldsymbol{\theta})$, as defined above, the stochastic dynamics of $\boldsymbol{w} = f(\boldsymbol{\theta})$ can be described explicitly in the sampling space of $\boldsymbol{w}$, and the resulting stochastic differential equations have again the form (C.1), with a new set of functions $\hat{b}(w_i)$ and $\hat{p}_{\mathcal{S}}(\boldsymbol{w}) = \prod_i \hat{p}_{\mathcal{S}}(w_i)$, given by (see proof below)

$$\begin{aligned} dw_i \;=\; &\left(\hat{b}(w_i)\,\frac{\partial}{\partial w_i}\log\hat{p}_{\mathcal{S}}(\boldsymbol{w}) \;+\; \hat{b}(w_i)\,\frac{\partial}{\partial w_i}\log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{w}) \;+\; T\,\hat{b}'(w_i)\right)dt \qquad \text{(C.30)}\\ &+\; \sqrt{2T\,\hat{b}(w_i)}\,d\mathcal{W}_i\;, \end{aligned}$$

$$\text{with}\qquad \hat{p}_{\mathcal{S}}(w_i) \;=\; \frac{p_{\mathcal{S}}\left(f^{-1}(w_i)\right)}{f'\left(f^{-1}(w_i)\right)} = \frac{p_{\mathcal{S}}(\theta_i)}{f'(\theta_i)} \tag{C.31}$$

$$\text{and}\qquad \hat{b}(w_i) \;=\; f'^2\left(f^{-1}(w_i)\right)\,b\left(f^{-1}(w_i)\right) = f'^2(\theta_i)\,b(\theta_i)\;,$$

where $f^{-1}(w_i) = \theta_i$ is the inverse function of $f(\cdot)$. Note that since (C.31) is of the form (C.1), the proof provided in Theorem 1 for the stationary distribution of $\boldsymbol{\theta}$ applies also to $\boldsymbol{w}$. The unique stationary distribution over the synaptic weights is therefore given by $p^*(\boldsymbol{w}) \equiv \frac{1}{Z}q^*(\boldsymbol{w})$, with $q^*(\boldsymbol{w}) = \left(\hat{p}_{\mathcal{S}}(\boldsymbol{w})\,p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{w})\right)^{\frac{1}{T}}$.

For the choices $p_{\mathcal{S}}(\theta_i) = \textsc{Normal}(\theta_i \mid \mu, \sigma^2)$, $b(\theta_i) = b$ and $w_i = \exp(\theta_i - \theta_0)$ (thus: $\theta_i = f^{-1}(w_i) = \log w_i + \theta_0$), plugged into the general result (C.31), we get

$$\hat{p}_{\mathcal{S}}(w_i) \;=\; \frac{p_{\mathcal{S}}\left(f^{-1}(w_i)\right)}{f'\left(f^{-1}(w_i)\right)} \;=\; \frac{1}{w_i \sigma \sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(\log w_i - \mu + \theta_0)^2\right) \quad \text{(C.32)}$$

$$\begin{aligned} \text{and} \qquad \hat{b}(w_i) \;&=\; f'^2\left(f^{-1}(w_i)\right)\, b\left(f^{-1}(w_i)\right) \\ &=\; c\, \exp(\log w_i + \theta_0 - \theta_0)^2 \;=\; c\, w_i^2 \,. \end{aligned} \quad \text{(C.33)}$$

Eq. (C.29) is the log-normal distribution and thus recovers the result (C.29). Note that (C.33) suggests that the resulting diffusion of the synaptic weights grows quadratically with the strength of the synaptic efficacies.

Proof

We prove the result (C.31) by deriving the stochastic process that governs $\boldsymbol{w} = f(\boldsymbol{\theta})$. From (C.28), we identify the drift $A_i(\boldsymbol{\theta})$ and diffusion $B_{ik}(\boldsymbol{\theta})$ according to

$$A_i(\boldsymbol{\theta}) = b(\theta_i)\, \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) \;+\; b(\theta_i)\, f'(\theta_i)\, \frac{\partial}{\partial w_i} \log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{w}) \;+\; T\, b'(\theta_i) \quad \text{(C.34)}$$

$$B_{ii}(\boldsymbol{\theta}) = 2T\, b(\theta_i) \qquad \text{and} \qquad B_{ik}(\boldsymbol{\theta}) = 0, \quad \text{for} \quad i \neq k \,. \quad \text{(C.35)}$$

Applying the rule for change of variables for stochastic differential equations to this expression yields (see (Gardiner, 2004), p. 95f)

$$\begin{aligned} dw_i \;&=\; df\,(\theta_i) = \left(A(\theta_i) f'(\theta_i) \;+\; \frac{1}{2}\, B(\theta_i)\, f''(\theta_i)\right) dt + \sqrt{B(\theta_i)}\, f'(\theta_i)\, d\mathcal{W}_i \\ &=\; \Big(f'(\theta_i)\, b(\theta_i)\, \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) \;+\; f'^2(\theta_i)\, b(\theta_i)\, \frac{\partial}{\partial w_i} \log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{w}) \qquad \text{(C.36)} \\ &\qquad + f'(\theta_i)\, b'(\theta_i) \;+\; f''(\theta_i)\, b(\theta_i)\Big) dt + f'(\theta_i)\, \sqrt{2T\, b(\theta_i)}\, d\mathcal{W}_i. \end{aligned}$$

By rearranging and expanding the terms we get

$$\begin{aligned} dw_i = \Bigg(& f'^2(\theta_i)\, b(\theta_i) \left(\frac{\frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}\,(\boldsymbol{\theta})}{f'\,(\theta_i)} - \frac{f''\,(\theta_i)}{f'^2\,(\theta_i)}\right) \;+\; f'^2(\theta_i)\, b(\theta_i)\, \frac{\partial}{\partial w_i} \log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{w}) \\ &+\; f'\,(\theta_i)\, b'\,(\theta_i) \;+\; 2\, f''\,(\theta_i)\, b\,(\theta_i)\Bigg) dt + \sqrt{2T\, f'^2(\theta_i)\, b(\theta_i)}\, d\mathcal{W}_i \,. \end{aligned}$$

Finally, by using the expressions for $\hat{b}(w)$ and $\hat{p}(w)$, taking the derivatives and comparing the terms we recover the result (C.31)

$$\begin{aligned} dw_i \;=\; & \left(\hat{b}(w_i)\, \frac{\partial}{\partial w_i} \log \hat{p}_{\mathcal{S}}(\boldsymbol{w}) \;+\; \hat{b}(w_i)\, \frac{\partial}{\partial w_i} \log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{w}) \;+\; \hat{b}'(w_i)\right) dt \\ & + \sqrt{2T\, \hat{b}(w_i)}\, d\mathcal{W}_i \end{aligned}$$

which completes the proof.

C.5 Details to: to Figure 3.3

The survival functions and parameter traces in Fig. 3.3B,D-F were based on the dynamics of synaptic parameters for prolonged runs of phase 1 of the experiment reported in Fig. 3.4. The network architecture was as described in Sec. 3.5. The network inputs were given by different realizations of digit *1* as described in Sec. 3.6. Learning was done for 108 hours of simulated biological time. In Fig. 3.3F we used $100\times$ slower learning dynamics ($b = 10^{-6}$). Each plot shows the survival of synapses that were newly formed during the preceding 12 hours, i.e. only active synapses that were retraced 12 hours ago are analyzed. The first 48 hours of learning were not evaluated. For the power-law fits, we adapted the method reported in (Loewenstein et al., 2015). Power law functions were given by $y = (t+1)^{-\gamma}$, where t is the survival time in hours, y is the fraction of remaining synapses and γ is the decay parameter. To fit γ to the data we measured $t_{1/4}$, i.e. the time it takes until 3/4 of the synapses have decayed. The mean over the three trials shown in Fig. 3.3E,F was then used to evaluated γ. For the fast dynamics in Fig. 3.3E this yielded $t_{1/4} = 1.23 \pm 0.58$ hours and $\gamma = 2.00$. For the slow dynamics in Fig. 3.3F we measured $t_{1/4} = 11.59 \pm 2.54$ hours, resulting in $\gamma = 0.64$.

C.6 Details to: Fast adaptation to changing input statistics

Spike-based Winner-Take-All network model

Network neurons were modeled as stochastic spike response neurons with a firing rate that depends exponentially on the membrane voltage (Jolivet et al., 2006; Mensi et al., 2011). The membrane potential $u_k(t)$ of neuron k is given by

$$u_k(t) = \sum_i w_{ki}\, x_i(t) \;+\; \beta_k(t)\,, \tag{C.37}$$

where $x_i(t)$ denotes the (unweighted) input from input neuron i, w_{ki} denotes the efficacy of the synapse from input neuron i, and $\beta_k(t)$ denotes a homeostatic adaptation current (see below). The input $x_i(t)$ models the influence of additive excitatory postsynaptic potentials (EPSPs) on the membrane potential of the neuron. Let $t_i^{(1)}, t_i^{(2)}, \ldots$ denote the spike times of input neuron i. Then, $x_i(t)$ is given by

$$x_i(t) = \sum_f \epsilon(t - t_i^{(f)})\,, \tag{C.38}$$

where ϵ is the response kernel for synaptic input, i.e., the shape of the EPSP, that had a double-exponential form in our simulations:

$$\epsilon(s) = \Theta(s)\left(e^{-\frac{s}{\tau_f}} - e^{-\frac{s}{\tau_r}}\right)\,, \tag{C.39}$$

with the rise-time constant τ_r = 2 ms, the fall-time constant τ_f = 20 ms. $\Theta(\cdot)$ denotes the Heaviside step function. The instantaneous firing rate $\rho_k(t)$ of network neuron k depends exponentially on the membrane potential and is subject to divisive lateral inhibition $I_{\text{lat}}(t)$ (described below):

$$\rho_k(t) = \frac{\rho_{\text{net}}}{I_{\text{lat}}(t)} \exp(u_k(t)) \,, \tag{C.40}$$

where ρ_{net} = 100 Hz scales the firing rate of neurons. Such exponential relationship between the membrane potential and the firing rate has been proposed as a good approximation to the firing properties of cortical pyramidal neurons (Jolivet et al., 2006). Spike trains were then drawn from independent Poisson processes with instantaneous rate $\rho_k(t)$ for each neuron. We denote the resulting spike train of the k^{th} neuron by $S_k(t)$.

Homeostatic adaptation current

Each output spike caused a slow depressing current, giving rise to the adaptation current $\beta_k(t)$. This implements a slow homeostatic mechanism that regulates the output rate of individual neurons (see (Habenschuss et al., 2012) for details). It was implemented as

$$\beta_k(t) \;=\; \gamma \sum_f \kappa(t - t_k^{(f)}) \,, \tag{C.41}$$

where $t_k^{(f)}$ denotes the f-th spike of neuron k and κ is an adaptation kernel that was modeled as a double exponential (Eq. (C.39)) with time constants τ_r = 12 s and τ_f = 30 s. The scaling parameter γ was set to γ = -8.

Lateral inhibition

Divisive inhibition (Carandini, 2012) between the K neurons in the WTA network was implemented in an idealized form (Nessler et al., 2013)

$$I_{\text{lat}}(t) = \sum_{l=1}^{K} \exp(u_l(t)) \,. \tag{C.42}$$

This form of lateral inhibition, that assumes an idealized access to neuronal membrane potentials, has been shown to implement a well-defined generative network model (Nessler et al., 2013), see below.

Synaptic sampling in spike-based Winner-Take-All networks as stochastic STDP

It has been shown in (Habenschuss et al., 2013) that the WTA-network defined above implicitly defines a generative model that is a mixture of Poissonian distributions. In this generative model, inputs $\boldsymbol{x}^n$ are assumed to be generated in dependence on the value of a hidden multinomial random variable h^n that can take on K possible values $1, \ldots, K$. Each neuron k in the WTA circuit corresponds to one value k of this hidden variable. In the generative model, for a given value of $h^n = k$, the value of an input x_i^n is then distributed according to a Poisson distribution with a mean that is determined by the synaptic weight w_{ki} from input neuron i to network neuron k:

$$p_{\mathcal{N}}(x_i^n | h^n = k, \boldsymbol{w}) = \textsc{Poisson}(x_i^n | \alpha e^{w_{ki}}) \;, \tag{C.43}$$

with a scaling parameter $\alpha > 0$. In other words, the synaptic weight w_{ki} encodes (in log-space) the firing rate of input neuron i, given that the hidden cause is k. For a given hidden cause, inputs are assumed to be independent, hence one obtains the probability of an input vector for a given hidden cause as

$$p_{\mathcal{N}}(\boldsymbol{x}^n | h^n = k, \boldsymbol{w}) = \prod_i \textsc{Poisson}(x_i^n | \alpha e^{w_{ki}}) \;. \tag{C.44}$$

The network implements inference in this generative model, i.e., for a given input $\boldsymbol{x}^n$, the firing rate of network neuron z_k is proportional to the posterior probability $p(h^n = k | \boldsymbol{x}^n, \boldsymbol{w})$ of the corresponding hidden cause. An online maximum likelihood learning rule for this generative model was derived in (Habenschuss et al., 2013). It changes synaptic weights according to

$$\frac{\partial}{\partial w_{ki}} \log p_{\mathcal{N}}(\boldsymbol{x}^n \,|\, \boldsymbol{w}) \approx S_k(t) \; \left(x_i(t) - \alpha \, e^{w_{ki}} \right) \;, \tag{C.45}$$

where $S_k(t)$ denotes the spike train of the postsynaptic neuron and $x_i(t)$ denotes the weight-normalized value of the sum of EPSPs from presynaptic neuron i at time t (i.e., the summed EPSPs that would arise for weight $w_{ki} = 1$). To define the synaptic sampling learning rule completely, we also need to define the parameter prior. In our experiments, we used a simple Gaussian prior on each parameter $p_{\mathcal{S}}(\boldsymbol{\theta}) = \prod_{k,i} \textsc{Normal}(\theta_{ki} | \mu, \sigma^2)$ with $\mu = 0.5$ and $\sigma = 1$. The derivative of the log-prior is given by

$$\frac{\partial}{\partial \theta_{ki}} \log p_{\mathcal{S}}(\boldsymbol{\theta}) \;=\; \frac{1}{\sigma^2} \left(\mu - \theta_{ki} \right) \;. \tag{C.46}$$

Inserting Eqs. (C.45) and (C.46) into the general form (3.10), we find that the synaptic sampling rule is given by

$$d\theta_{ki} \;=\; b \left(\frac{1}{\sigma^2} \left(\mu - \theta_{ki} \right) \;+\; N w_{ki} \, S_k(t) \; \left(x_i(t) - \alpha \, e^{w_{ki}} \right) \right) dt \;+ \sqrt{2b} \, d\mathcal{W}_{ki} \;, \tag{C.47}$$

which corresponds to rule (3.11) with double indices ki replaced by single parameter indexing i to simplify notation.

Simulation details for spiking network simulations

Computer simulations of spiking neural networks (Figs. 3.3, 3.4, and 3.5) were based on adapted event-based simulation software from (Kappel et al., 2014). In all spiking neural network simulations, synaptic weights were updated according to the rule (3.11) with parameters $N = 100$, $\alpha = e^{-2}$, and $b = 10^{-4}$, except for panel 3.3F where $b = 10^{-6}$ was used as a control. In the simulations, we directly implemented the time-continuous evolution of the network parameters in an event-based update scheme. Before learning, initial synaptic parameters were independently drawn from the prior distribution $p_{\mathcal{S}}(\boldsymbol{\theta})$.

For the mapping (3.9) from synaptic parameters θ_{ki} to synaptic efficacies w_{ki}, we used as offset $\theta_0 = 3$. This results in synaptic weights that shrink to small values (< 0.05) when synaptic parameters are below zero. In the simulation, we clipped the synaptic weights to zero for negative synaptic parameters θ to account for retracted synapses. More precisely, the actual weights $\hat{w}_{ki}$ used for the computation of the membrane potential (C.37) were given by $\hat{w}_{ki} = \max\{0, w_{ki} - \exp(-\theta_0)\}$. To avoid numerical problems, we clipped the synaptic parameters at -5 and the maximum amplitude of instantaneous parameter changes to $5b$.

Network inputs

The spatiotemporal spike patterns in Fig. 3.4 are realizations of Poisson spike trains, each representing a certain point in the 3-dimensional sensory environment (a unit cube). Each of the 1000 input neurons was assigned to a Gaussian tuning curve with $\sigma = 0.3$. Tuning curve centers were independently and equally scattered over the unit cube. For each sensory experience the firing rate of an individual input neuron was given by the support of sensory experience under the neuron's tuning curve (normalized between 0 Hz and 80 Hz). In addition an offset of 2 Hz background noise was added. The patterns had a duration of 200 ms. During that time the firing rates of input neurons were kept fixed and independent Poisson spike trains were drawn.

The two environments (SE and EE) in Fig. 3.4 were realized by Gaussian mixture models. The means of the Gaussians were randomly placed around the center of the unit cube (each component was independently drawn from $\textsc{Normal}(0.5, 0.2)$). The covariance matrices of the Gaussian cluster centers were randomly given by $0.04\mathbb{I} + 0.01\xi$, where $\mathbb{I}$ is the 3-dimensional identity matrix and ξ is a matrix of randomly drawn values from $\textsc{Normal}(0, 1)$. Sensory experiences were generated by randomly selecting one Gaussian cluster (with equal probability) and then drawing a sample position from the corresponding multivariate Gaussian.

Learning schedule and data analysis

The network was first exposed to samples from the standard environment (SE, Fig. 3.4D) for 3 hours (54000 input sample presentations). In the second learning phase input samples from the enriched environment (EE, Fig. 3.4E) were given for 1 hour (18000 samples). In the third phase samples from either SE (EE-SE condition) or EE (EE-EE condition) were presented for additional 5 hours (90000 samples, the two cases are compared in Fig. 3.4G).

Formation rates of synaptic connections shown in Fig. 3.4F represent the number of spines that were formed during a time window of Δt = 30 minutes, i.e. the number of synaptic connections that were not present ($\theta_i \leq 0$) at time $t - \Delta t$ but at time t. The SE condition in Fig. 3.4F was evaluated at the end of learning phase 1, the EE condition was evaluated at the beginning of EE exposure.

For the survival plots in Fig. 3.4G the newly formed synaptic connections at the end of the EE condition were taken into account (see above). Networks from the EE-EE or EE-SE condition were compared. The presence of synaptic connections ($\theta_i > 0$) was evaluated in intervals of 30 minutes. The plot in Fig. 3.4E,F show mean values and standard deviations over 5 individual trial runs.

C.7 Details to: Inherent compensation capabilities of networks with synaptic sampling

Here we provide details to the network model and spiking inputs for the recurrent WTA circuits (Fig. 3.5), describe the method that was used to evaluate the reconstruction performance, and provide further details to the emergent assembly sequences among the hidden neurons.

Network model

In Fig. 3.5 two recurrently connected ensembles, each consisting of four WTA circuits, were used. The parameters of neuron and synapse dynamics were as described in the previous section. All synapses, lateral and feedforward, were subject to the same learning rule (3.11). Lateral connections within and between the WTA Circuit neurons were unconstrained (allowing potentially all-to-all connectivity). Connections from input neurons were constraint as shown in Fig. 3.5. The lateral synapses were treated in the same way as synapses from input neurons but had a synaptic delay of 5 ms.

Network inputs

Handwritten digit images for Fig. 3.5 were taken from the MNIST dataset (LeCun et al., 1998). Each pixel was represented by a single afferent neuron. Gray scale values where scaled to 0 - 50 Hz Poisson input rate and 1 Hz input noise was added on top. These Poisson rates were kept fixed for each example input digit for the duration of the input presentation.

The spoken digit presentations in Fig. 3.5 were given by reconstructed cochleagrams of speech samples of isolated spoken digits from the TI 46 dataset (also used in (Klampfl and Maass, 2013; Hopfield and Brody, 2001)). Each of the 77 channels of the cochleagrams was represented by 10 afferent neurons, giving a total of 770. Cochleagrams were normalized between 0 Hz and 80 Hz and used to draw individual Poisson spike trains for each afferent neuron. In addition 1 Hz Poisson noise was added on top. We used 10 different utterances of digits *1* and *2* of a single speaker. We selected 7 utterances for training and 3 for testing. For training, one randomly selected utterance from the training set was presented together with a randomly chosen instance of the corresponding handwritten digit. The spike patterns for the written digits (see above) had the same duration as the spoken digits. Each digit presentation was padded with 25 ms, 1 Hz Poisson noise before and after the digit pattern.

For test trials in which only the auditory stimulus was presented, the activity of the visual input neurons was set to 1 Hz throughout the whole pattern presentation. The learning rate b was set to zero during these trials. The PETH plots were computed over 100 trial responses of the network to the same stimulus class (e.g. presentation of digit *1*). Spike patterns for input stimuli were randomly drawn in each trial for the given rates. Spike trains were then filtered with a Gaussian filter with σ = 50 ms and summed in a time-discrete matrix with 10 ms bin length. Maximum firing times were assigned to the time bin with the highest PETH amplitude for each neuron.

Evaluating the reconstruction performance

The reconstructed visual stimuli were generated by producing an auditory stimulus via the $\mathbf{x}_A$ neurons and evaluating the corresponding activity of visual $\mathbf{z}_V$ neurons. A sample auditory stimulus from the test set was randomly chosen and spike patterns were generated as for the training session (see main text). The resulting spike trains from the $\mathbf{z}_V$ neurons were smoothed with the EPSP kernel (C.39). The current strengths of the feedforward synapses were then weighted by these smoothed responses, evaluated 300ms after stimulus onset. Fig. 3.5B shows two example reconstructed stimuli. The pixel values were rescaled to the color range of the images.

The reconstruction performance was assessed by the performance of linear classifiers trained on the response of $\mathbf{z}_V$ neurons. The classifiers were trained on 50

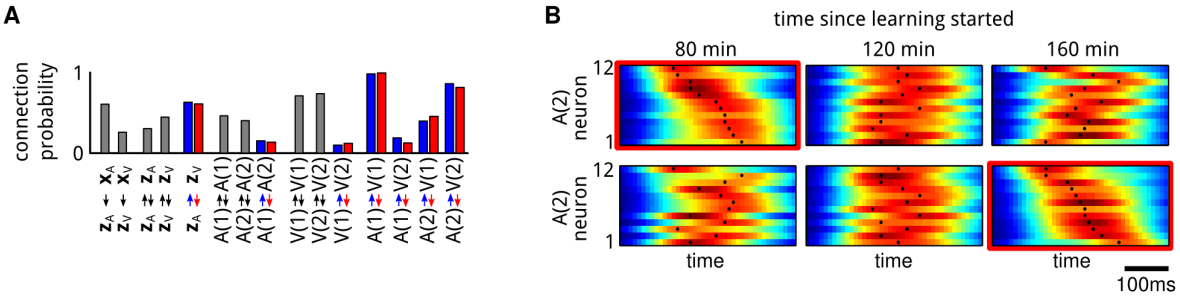

Fig. C.2: Emergent assembly sequences and functional connectivity in a simplified model of multi-modal sensory integration. A: Only a fraction of the structurally possible connectivity (all to all) emerges as functional connectivity after learning in the network shown in Fig. 3.5 of the main text. Connection probabilities (number of functional connections normalized to the number of possible connections) are shown between input and hidden neurons and between hidden neurons that are recruited for assembly sequences. The colors of the bars match the direction of the connections (colored arrows). Neurons from assembly sequences that encode the same digit class are more likely to be connected after learning. **B:** Neurons within the auditory assembly sequence $A(2)$ fire in a characteristic sequential order. PETH evaluated at different training times before the lesions are shown. Neurons are sorted by the time points of highest activity (black dots) after 80 minutes (top) and 160 minutes (bottom) of learning (plot used for sorting is highlighted by red border). The sequential firing order changes during prolonged learning.

samples of reconstructed visual stimuli of each digit class, generated by producing each time fresh Poisson trains in the $\mathbf{x}_A$ neurons with time-varying firing rates according to the spoken digit samples, using Matlab's build-in naive Bayes classifier method. Additional 50 samples of each class were then used to evaluate the reconstruction performance (number of correctly labeled samples). The values shown in Fig. 3.5E are mean values over 20 classifiers, trained and tested for independently generated Poisson spike trains for $\mathbf{x}_A$ neurons as described above.

Assembly sequence analysis

To further evaluate the emergent activity patterns and connectivity in the network of hidden neurons we focused on the emergent assembly sequences within the hidden neuron (see e.g. (C. D. Harvey et al., 2012) for experimental data on assembly sequences). Affiliation of neurons to assembly sequences was assessed through the PETH (see main text). PETHs were computed for both digits over 100 trial responses from all $\mathbf{z}_A$ and $\mathbf{z}_V$ neurons. Neurons were assigned to the assembly sequence corresponding to the digit for which the neuron showed the maximum PETH amplitude. Neurons for which the maximum was outside the time interval [50ms, 450ms] after stimulus onset, were excluded from the analysis (not assigned to an assembly sequence). We refer to the set of neurons that take part in these assembly sequences in the visual and auditory populations $\mathbf{z}_A$ and $\mathbf{z}_V$ as $V(1), A(1)$ for digit 1, and $V(2), A(2)$ for digit 2. We find that synaptic plasticity generates associations between corresponding components of the assembly sequences in the visual and auditory ensemble, (i.e., between $V(1)$ and $A(1)$, $V(2)$ and $A(2)$) in spite of the fact that synaptic connections are asymmetric in this model, as in

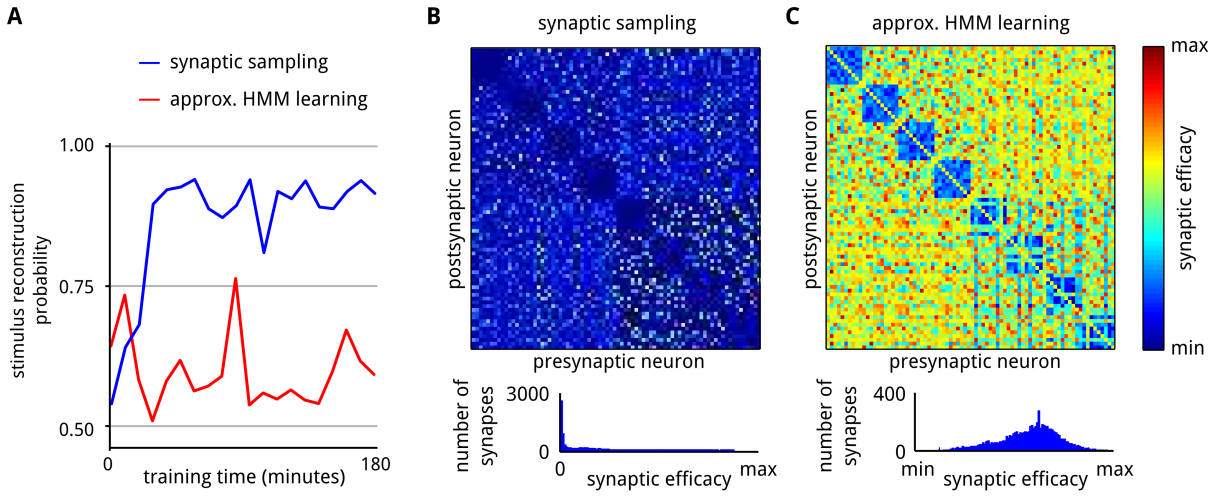

Fig. C.3: Comparison between synaptic sampling and approximate HMM learning. A: The stimulus reconstruction performance of synaptic sampling (blue) and the approximate HMM learning (red). **B,C:** Comparison of the lateral synaptic weights that result from synaptic sampling (B) and approximate HMM learning. Insets on the bottom show the histograms over the synaptic parameters.

most biological networks of neurons. Fig. C.2A shows the connection probabilities between the pairs of assembly sequences after 160 minutes of training. Assemblies that encode correlated stimuli are more likely to be connected.

In Fig. C.2B we study the drift of the preferred firing time of neurons within an assembly sequence that takes place on a larger time scale throughout learning, due to the stochastic term in the learning rule. The auditory assembly sequence $A(2)$ is analyzed. Neurons within this assembly sequence fire in a specific order, as in (C. D. Harvey et al., 2012). Our learning model predicts that this sequential order changes during larger periods of learning. The result shown in Fig. C.2B is qualitatively similar to the data reported in (Y. Ziv et al., 2013) for a different type of learned neural code (place cells). The time scale of these fluctuations can be regulated through the parameter b (learning rate) in the synaptic sampling rule (3.3). We had chosen here a faster time scale of hours (rather than days, as in (Y. Ziv et al., 2013)) in order to achieve tractable computer simulation times.

Comparison to deterministic STDP

Here we compare the synaptic sampling learning to the approximate spike-based expectation-maximization (EM) algorithm for hidden Markov models implemented through spiking neurons, which was introduced in (Kappel et al., 2014). The algorithm is a deterministic STDP-like update scheme and realizes the data-dependent drift term of the synaptic sampling rule (3.11). More precisely we used deterministic updates of the form

$$dw_i \;=\; b\,N\,S(t)\,(x_i(t) - \alpha\, e^{w_i}) \tag{C.48}$$

for the lateral and feedforward synaptic weights of the WTA networks.

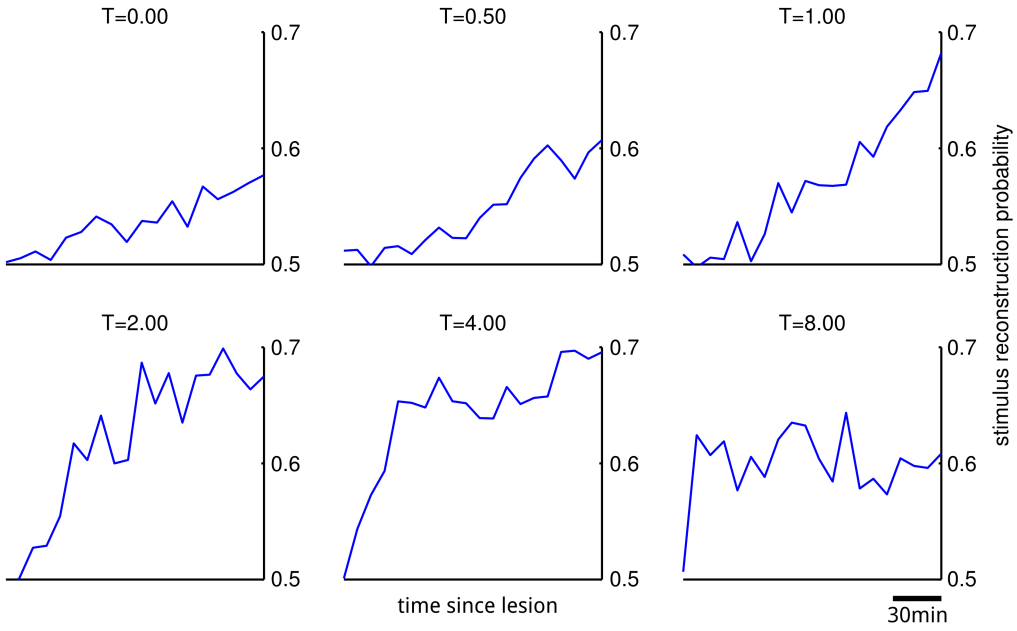

Fig. C.4: Comparison of the reconstruction performance of networks with different temperatures T for learning. The plots show mean values over 20 individual trial runs as the one shown in Fig. 3.5. Reconstruction performance was evaluated in an 8 minute interval. Plots show linear interpolations between these values.

We trained a network using the the approximate HMM learning rule (C.48) on phase 1 of the learning task in Fig. 3.5. The results are compared in Fig. C.3A. Approximate HMM learning was not able to learn this task accurately which results in a low reconstruction performance throughout the whole learning time. In Fig. C.3B,C we compare the matrices of synaptic weights that result from the two algorithms. The color ranges are rescaled to the min/max values of the synaptic weights for each plot. Note that learning rule (C.48) – unlike synaptic sampling – can produce negative synaptic weights. Due to the prior distribution over synaptic parameters the representation learned by synaptic sampling is much sparser and associations are more pronounced which allows for a more reliable recall of input stimuli.

Impact of the temperature on the reconstruction performance

In Fig. C.4 we analyze the impact of the temperature (parameter T in equation (C.1) of the main text) on the reconstruction performance in the last phase of the learning task in Fig. 3.5 (all lateral synapses are removed). The speed of the regrowth of retracted synapses is determined by the prior and the random fluctuations due to the Wiener process. Therefore we found that the temperature parameter T has a large impact on the time it takes until the network recovers from the lesion.

With a temperature of zero (deterministic updates) the network requires significantly more time until the reconstruction probability starts to increase. The prior that is close to zero only very slowly drives a sufficient number of synapses above the threshold. With increasing temperature the speed to recover from the lesion increases. For too large temperatures (e.g. $T = 8$) the performance degrades since the network diffuses quickly from solutions. The optimal value for the temperature was found to be between $T = 2$ and $T = 4$ for this particular learning problem (see Fig. C.4).

Appendix D

Appendix to Chapter 4: Reward-based self-configuration of neural circuits

D.1 Bayesian framework for reward-modulated learning

The classical goal of reinforcement learning is to maximize the expected future discounted reward $\mathcal{V}(\boldsymbol{\theta})$ given by

$$\mathcal{V}(\boldsymbol{\theta}) \;=\; \langle \int_0^\infty e^{-\frac{\tau}{\tau_e}} r(\tau)\, d\tau \rangle_{p(\boldsymbol{r}|\boldsymbol{\theta})} \,. \tag{D.1}$$

In Eq. (D.1) we integrate over all future rewards $r(\tau)$, while discounting more remote rewards exponentially with a discount rate τ_e, which for simplicity was set equal to 1 s in this paper. We find (see Eq. (D.10)) that this time constant τ_e is immediately related to the experimentally studied time window or eligibility trace for the influence of dopamine on synaptic plasticity (Yagishita et al., 2014). The expectation in Eq. (D.1) is taken with respect to the distribution $p(\boldsymbol{r}|\boldsymbol{\theta})$ over sequences $\boldsymbol{r} = \{r(\tau), \tau \geq 0\}$ of future rewards that result from the given set of synaptic parameters $\boldsymbol{\theta}$. The stochasticity of the reward sequence $\boldsymbol{r}$ arises from stochastic network inputs, stochastic network responses, and stochastic reward delivery. The resulting distribution $p(\boldsymbol{r}|\boldsymbol{\theta})$ of reward sequences $\boldsymbol{r}$ for the given parameters $\boldsymbol{\theta}$ can also include influences of network initial conditions by assuming some distribution over these initial conditions. Network initial conditions include for example initial values of neuron membrane voltages and refractory states of neurons. The role of initial conditions on network learning is further discussed below when we consider the online learning scenario in *Reward-modulated synaptic plasticity approximates gradient ascent on the expected discounted reward*.

There exists a close relationship between reinforcement learning and Bayesian inference (Vlassis et al., 2012; Rawlik et al., 2013; Botvinick and Toussaint, 2012). To make this relationship apparent, we define our model for reward-gated network plasticity by introducing a binary random variable v_b that represents the currently expected future discounted reward in a probabilistic manner. The likelihood $p_{\mathcal{N}}\,(v_b = 1 \,|\, \boldsymbol{\theta})$ is determined in this theoretical framework by the expected future discounted reward Eq. (D.1) that is achieved by a network with parameter set $\boldsymbol{\theta}$ (see e.g., (Rawlik et al., 2013)):

$$p_{\mathcal{N}}\,(v_b = 1 \,|\, \boldsymbol{\theta}) \;\equiv\; \frac{1}{\mathcal{Z}_{\mathcal{V}}} \mathcal{V}(\boldsymbol{\theta}) \,, \tag{D.2}$$

where $\mathcal{Z}_{\mathcal{V}}$ denotes a constant, that assures that Eq. (D.2) is a correctly normalized probability distribution. Thus reward-based network optimization can be formalized as maximizing the likelihood $p_{\mathcal{N}}(v_b = 1 \,|\, \boldsymbol{\theta})$ with respect to the network configuration $\boldsymbol{\theta}$. Structural constraints can be integrated into a stochastic model for network plasticity through a prior $p_S(\boldsymbol{\theta})$ over network configurations. Hence reward-gated network optimization amounts from a theoretical perspective to learning of the posterior distribution $p^*(\boldsymbol{\theta}|v_b = 1)$, which by Bayes' rule is defined (up to normalization) by $p_{\mathcal{S}}(\boldsymbol{\theta}) \cdot p_{\mathcal{N}}(v_b = 1 \,|\, \boldsymbol{\theta})$. Therefore, the learning goal can be formalized in a compact form as evaluating the posterior distribution $p^*(\boldsymbol{\theta}|v_b = 1)$ of network parameters $\boldsymbol{\theta}$ under the constraint that the abstract learning goal $v_b = 1$ is achieved.

More generally, one is often interested in a tempered version of the posterior

$$p_T^*(\boldsymbol{\theta}) \equiv \frac{1}{\mathcal{Z}} p^*(\boldsymbol{\theta}|v_b = 1)^{\frac{1}{T}} \,, \tag{D.3}$$

where $\mathcal{Z}$ is a suitable normalization constant and $T > 0$ is the temperature parameter that controls the "sharpness" of $p_T^*(\boldsymbol{\theta})$. For $T = 1$, $p_T^*(\boldsymbol{\theta})$ is given by the original posterior, $T < 1$ emphasizes parameter values with high probability in the posterior, while $T > 1$ leads to parameter distributions $p_T^*(\boldsymbol{\theta})$ which are more uniformly distributed than the posterior.

D.2 Analysis of Bayesian policy sampling

Here we prove that the stochastic parameter dynamics Eq. (4.5) samples from the tempered posterior distribution $p_T^*(\boldsymbol{\theta})$ given in Eq. (D.3). In *Results* we suppressed time-dependencies in order to simplify notation. We reiterate Eq. (4.3) with explicit time-dependencies of parameters:

$$d\theta_i(t) \;=\; \beta \; \frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta}|v_b = 1)\Big|_{\boldsymbol{\theta}(t)} dt \;+\; \sqrt{2\beta T}\, d\mathcal{W}_i \,, \tag{D.4}$$

where the notation $\frac{\partial}{\partial \theta_i} f(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}(t)}$ denotes the derivative of $f(\boldsymbol{\theta})$ with respect to θ_i evaluated at the current parameter values $\boldsymbol{\theta}(t)$. By Bayes' rule, the derivative of the log posterior is the sum of the derivatives of the prior and the likelihood:

$$\begin{aligned} \frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta}|v_b = 1) \;&=\; \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(v_b = 1 \,|\, \boldsymbol{\theta}) \\ &=\; \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + \frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta}) \,, \end{aligned}$$

which allows us to rewrite Eq. (D.4) as

$$d\theta_i(t) \;=\; \beta \left(\frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}\left(\boldsymbol{\theta}\right)\Big|_{\boldsymbol{\theta}(t)} + \frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}(t)} \right) dt \;+\; \sqrt{2\beta T}\, d\mathcal{W}_i \;, \tag{D.5}$$

which is identical to the form Eq. (4.5), where the contributions of $p_{\mathcal{S}}\left(\boldsymbol{\theta}\right)$ and $\mathcal{V}(\boldsymbol{\theta})$ are given explicitly.

The fundamental property of the synaptic sampling dynamics Eq. (D.4) is formalized in Theorem 2 and proven below. Before we state the theorem, we briefly discuss its statement in simple terms. Consider some initial parameter setting $\boldsymbol{\theta}(0)$. Over time, the parameters change according to the dynamics (D.4). Since the dynamics include a noise term, the exact value of the parameters $\boldsymbol{\theta}(t)$ at some time $t > 0$ cannot be determined. However, it is possible to describe the exact distribution of parameters for each time t. We denote this distribution by $p_{\mathrm{FP}}(\boldsymbol{\theta}, t)$, where the "FP" subscript stands for "Fokker-Planck" since the evolution of this distribution is described by the Fokker-Planck equation (D.6) given below. Note that we make the dependence of this distribution on time explicit in this notation. It can be shown that for the dynamics (D.6), $p_{\mathrm{FP}}(\boldsymbol{\theta}, t)$ converges to a well-defined and unique *stationary distribution* in the limit of large t. Of practical relevance is the so-called burn-in time after which the distribution of parameters is very close to the stationary distribution. Note that the parameters will continue to change. Nevertheless, at any time t after the burn in, we can expect the parameter vector $\boldsymbol{\theta}(t)$ to be situated at a particular value with the probability (density) given by the stationary distribution, see Fig. 4.1D,F. Any distribution that is *invariant* under the parameter dynamics is a stationary distribution. Here, invariance means: when one starts with an invariant distribution over parameters in the Fokker-Planck equation, the dynamics are such that this distribution will be kept forever (we will use this below in the proof of Theorem 2). Theorem 2 states that the parameter dynamics leaves $p_T^*(\boldsymbol{\theta})$ given in Eq. (D.3) invariant, i.e., it is a stationary distribution of the network parameters. Note that in general, the stationary distribution may not be uniquely defined. That is, it could happen that for two different initial parameter values, the network reaches two different stationary distributions. Theorem 2 further states that for the synaptic sampling dynamics, the stationary distribution is unique, i.e., the distribution $p_T^*(\boldsymbol{\theta})$ is reached from *any* initial parameter setting when the conditions of the theorem apply. We now state Theorem 2 formally. To simplify notation we drop in the following the explicit time dependence of the synaptic parameters $\boldsymbol{\theta}$.

Theorem 2. *Let $p^*(\boldsymbol{\theta} \mid v_b = 1)$ be a strictly positive, continuous probability distribution over parameters $\boldsymbol{\theta}$, twice continuously differentiable with respect to $\boldsymbol{\theta}$, and let $\beta > 0$. Then the set of stochastic differential equations Eq.* (D.4) *leaves the distribution $p_T^*(\boldsymbol{\theta})$* (D.3) *invariant. Furthermore, $p_T^*(\boldsymbol{\theta})$ is the unique stationary distribution of the sampling dynamics.*

Proof. The proof is analogous to the one provided in (Kappel et al., 2015a). The stochastic differential equation Eq. (D.4) translates into a Fokker-Planck equation (Gardiner, 2004) that describes the evolution of the distribution over parameters $\boldsymbol{\theta}$

$$\frac{d}{dt}p_{\mathrm{FP}}(\boldsymbol{\theta},t) = \sum_i -\frac{\partial}{\partial\theta_i}\left(\beta\,\frac{\partial}{\partial\theta_i}\log p^*(\boldsymbol{\theta}\,|\,v_b=1)\right)p_{\mathrm{FP}}(\boldsymbol{\theta},t) + \frac{\partial^2}{\partial\theta_i^2}\left(\beta\,T\,p_{\mathrm{FP}}(\boldsymbol{\theta},t)\right), \tag{D.6}$$

where $p_{\mathrm{FP}}(\boldsymbol{\theta},t)$ denotes the distribution over network parameters at time t. To show that $p_T^*(\boldsymbol{\theta})$ leaves the distribution invariant, we have to show that $\frac{d}{dt}p_{\mathrm{FP}}(\boldsymbol{\theta},t)=0$ (i.e., $p_{\mathrm{FP}}(\boldsymbol{\theta},t)$ does not change) if we set $p_{\mathrm{FP}}(\boldsymbol{\theta},t)$ to $p_T^*(\boldsymbol{\theta})$ on the right hand side of Eq. (D.6). Plugging in the presumed stationary distribution $p_T^*(\boldsymbol{\theta})$ for $p_{\mathrm{FP}}(\boldsymbol{\theta},t)$ on the right hand side of Eq. (D.6), one obtains

$$\begin{aligned}
\frac{d}{dt}p_{\mathrm{FP}}(\boldsymbol{\theta},t) &= \sum_i -\frac{\partial}{\partial\theta_i}\left(\beta\,\frac{\partial}{\partial\theta_i}\log p^*(\boldsymbol{\theta}\,|\,v_b=1)\,p_T^*(\boldsymbol{\theta})\right) + \frac{\partial^2}{\partial\theta_i^2}\left(\beta\,T\,p_T^*(\boldsymbol{\theta})\right)\\
&= \sum_i -\frac{\partial}{\partial\theta_i}\left(\beta\,p_T^*(\boldsymbol{\theta})\,\frac{\partial}{\partial\theta_i}\log p^*(\boldsymbol{\theta}\,|\,v_b=1)\right) + \frac{\partial}{\partial\theta_i}\left(\beta\,T\,\frac{\partial}{\partial\theta_i}p_T^*(\boldsymbol{\theta})\right)\\
&= \sum_i -\frac{\partial}{\partial\theta_i}\left(\beta\,p_T^*(\boldsymbol{\theta})\,\frac{\partial}{\partial\theta_i}\log p^*(\boldsymbol{\theta}\,|\,v_b=1)\right)\\
&\quad + \frac{\partial}{\partial\theta_i}\left(\beta\,T\,p_T^*(\boldsymbol{\theta})\,\frac{\partial}{\partial\theta_i}\log p_T^*(\boldsymbol{\theta})\right),
\end{aligned}$$

which by inserting $p_T^*(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}}p^*(\boldsymbol{\theta}\,|\,v_b=1)^{\frac{1}{T}}$, with normalizing constant $\mathcal{Z}$, becomes

$$\begin{aligned}
\frac{d}{dt}p_{\mathrm{FP}}(\boldsymbol{\theta},t) &= \frac{1}{\mathcal{Z}}\sum_i -\frac{\partial}{\partial\theta_i}\left(\beta\,p^*(\boldsymbol{\theta})\,\frac{\partial}{\partial\theta_i}\log p^*(\boldsymbol{\theta}\,|\,v_b=1)\right)\\
&\quad + \frac{\partial}{\partial\theta_i}\left(\beta\,T\,p^*(\boldsymbol{\theta})\,\frac{1}{T}\frac{\partial}{\partial\theta_i}\log p^*(\boldsymbol{\theta}\,|\,v_b=1)\right)\\
&= \sum_i 0 = 0\ .
\end{aligned}$$

This proves that $p_T^*(\boldsymbol{\theta})$ is a stationary distribution of the parameter sampling dynamics Eq. (D.4). Since β is strictly positive, this stationary distribution is also unique (see Section 3.7.2 in (Gardiner, 2004)).

The unique stationary distribution of Eq. (D.6) is given by $p_T^*(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}}p^*(\boldsymbol{\theta}|v_b=1)^{\frac{1}{T}}$, i.e. $p_T^*(\boldsymbol{\theta})$ is the only solution for which $\frac{d}{dt}p_{\mathrm{FP}}(\boldsymbol{\theta},t)$ becomes 0, which completes the proof.

□

D.3 Network model

Plasticity rules for this general framework were derived based on a specific spiking neural network model, which we describe in the following. All reported computer simulations were performed with this network model. We considered a general network scaffold $\mathcal{N}$ of K neurons with potentially asymmetric recurrent connections. Neurons are indexed in an arbitrary order by integers between 1 and K. We denote the output spike train of a neuron k by $z_k(t)$. It is defined as the sum of Dirac delta pulses positioned at the spike times $t_k^{(1)}, t_k^{(2)}, \ldots$, i.e., $z_k(t) = \sum_l \delta(t - t_k^{(l)})$. Potential synaptic connections are also indexed in an arbitrary order by integers between 1 and K_{syn}, where K_{syn} denotes the number of potential synaptic connections in the network. We denote by PRE_i and POST_i the index of the pre- and postsynaptic neuron of synapse i, respectively, which unambiguously specifies the connectivity in the network. Further, we define SYN_k to be the index set of synapses that project to neuron k. Note that this indexing scheme allows us to include multiple (potential) synaptic connections between a given pair of neurons. We included this experimentally observed feature of biological neuronal networks in all our simulations. We denote by $w_i(t)$ the synaptic efficacy of the i-th synapse in the network at time t.

Network neurons were modeled by a standard stochastic variant of the spike response model (Gerstner et al., 2014). In this model, the membrane potential of a neuron k at time t is given by

$$u_k(t) \;=\; \sum_{i \,\in\, \text{SYN}_k} y_{\text{PRE}_i}(t)\, w_i(t) \;+\; \vartheta_k(t)\;, \tag{D.7}$$

where $\vartheta_k(t)$ denotes the slowly changing bias potential of neuron k, and $y_{\text{PRE}_i}(t)$ denotes the trace of the (unweighted) postsynaptic potentials (PSPs) that neuron PRE_i leaves in its postsynaptic synapses at time t. More precisely, it is defined as $y_{\text{PRE}_i}(t) = z_{\text{PRE}_i}(t) * \epsilon(t)$ given by spike trains filtered with a PSP kernel of the form $\epsilon(t) = \Theta(t)\, \frac{\tau_r}{\tau_m - \tau_r} \left(e^{-\frac{t}{\tau_m}} - e^{-\frac{t}{\tau_r}} \right)$, with time constants $\tau_m = 20$ ms and $\tau_r = 2$ ms, if not stated otherwise. Here $*$ denotes convolution and $\Theta(\cdot)$ is the Heaviside step function, i.e. $\Theta(x) = 1$ for $x \geq 0$ and 0 otherwise.

The synaptic weights $w_i(t)$ in Eq. (D.7) were determined by the synaptic parameters $\theta_i(t)$ through the mapping Eq. (4.1) for $\theta_i(t) > 0$. Synaptic connections with $\theta_i(t) \leq 0$ were interpreted as not functional (disconnected) and $w_i(t)$ was therefore set to 0 in that case.

The bias potential $\vartheta_k(t)$ in Eq. (D.7) implements a slow adaptation mechanism of the intrinsic excitability, which ensures that the output rate of each neuron stays near the firing threshold and the neuron maintains responsiveness (Desai et al., 1999; Fan et al., 2005). We used a simple adaptation mechanism which was updated according to

$$\tau_\vartheta \frac{d\vartheta_k(t)}{dt} \;=\; \nu_0 - z_k(t)\;, \tag{D.8}$$

where $\tau_\vartheta = 50$ s is the time constant of the adaptation mechanism and $\nu_0 = 5$ Hz is the desired output rate of the neuron. In our simulations, the bias potential $\vartheta_k(t)$ was initialized at -3 and then followed the dynamics given in Eq. (D.8). This regularization is a simplified version of the mechanism proposed in (Remme and Wadman, 2012) to balance activity in networks of excitatory and inhibitory neurons. We found that this regularization significantly increased the performance and learning speed of our network model, presumably due to the substantial change in neural fan-in (due to rewiring as discussed above) that may take place during learning which is counteracted by such a mechanism.

We used a simple refractory mechanism for our neuron model. The firing rate, or intensity, of neuron k at time t is defined by the function $f_k(t) = f(u_k(t), \rho_k(t))$, where $\rho_k(t)$ denotes a refractory variable that measures the time elapsed since the last spike of neuron k. We used an exponential dependence between membrane potential and firing rate, such that the instantaneous firing rate of the neuron k at time t can be written as

$$f_k(t) = f(u_k, \rho_k) = \exp(u_k)\Theta(\rho_k - t_{\text{ref}}) \,. \tag{D.9}$$

Furthermore, we denote by $f_{\text{POST}_i}(t)$ the firing rate of the neuron postsynaptic to synapse i. If not stated otherwise we set the refractory time t_{ref} to 5 ms. In addition, a subset of neurons was clamped to some given firing rates (input neurons), such that $f_k(t)$ of these input neurons was given by an arbitrary function. We denote the spike train from these neurons by $\boldsymbol{x}(t)$, the network input.

D.4 Synaptic dynamics for the reward-based synaptic sampling model

Here, we provide additional details on how the synaptic parameter dynamics Eq. (4.5) was computed. We will first provide an intuitive interpretation of the equations and then provide a detailed derivation in the next section. The second term $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ of Eq. (4.5) denotes the gradient of the expected future discounted reward Eq. (D.1). In general, optimizing this function has to account for the case where rewards are provided after some delay period. It is well known that this *distal reward problem* can be solved using plasticity mechanisms that make use of eligibility traces in the synapses that are triggered by near coincident spike patterns, but their consolidation into the synaptic weights is delayed and modulated by the reward signal $r(t)$ (Sutton and Barto, 1998; Izhikevich, 2007). The theoretically optimal shape for these eligibility traces can be derived using the reinforcement learning theory and depends on the choice of network model. For the spiking neural network model described above, the gradient $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ can be estimated through a plasticity mechanism that uses an eligibility trace $e_i(t)$ in each synapse i which gets updated according to

$$\frac{de_i(t)}{dt} = -\frac{1}{\tau_e} e_i(t) + w_i(t)\, y_{\text{PRE}_i}(t)\, (z_{\text{POST}_i}(t) - f_{\text{POST}_i}(t)) \,, \tag{D.10}$$

where $\tau_e = 1$ s is the time constant of the eligibility trace. Recall that PRE_i denotes the index of the presynaptic neuron and POST_i the index of the postsynaptic neuron for synapse i. In Eq. (D.10) $z_{\text{POST}_i}(t)$ denotes the postsynaptic spike train, $f_{\text{POST}_i}(t)$ denotes the instantaneous firing rate (Eq. (D.9)) of the postsynaptic neuron and $w_i(t)\, y_{\text{PRE}_i}(t)$ denotes the postsynaptic potential under synapse i.

The last term of Eq. (D.10) shares salient properties with standard STDP learning rules, since plasticity is enabled by the presynaptic term $y_{\text{PRE}_i}(t)$ and gated by the postsynaptic term $(z_{\text{POST}_i}(t) - f_{\text{POST}_i}(t))$ (see (Pfister et al., 2006)). The latter term also regularizes the plasticity mechanism such that synapses stop growing if the firing probability $f_{\text{POST}_i}(t)$ of the postsynaptic neuron is already close to one.

The eligibility trace Eq. (D.10) is modulated by the reward $r(t)$ and integrated in each synapse i using a second dynamic variable

$$\frac{dg_i(t)}{dt} \;=\; -\frac{1}{\tau_g} g_i(t) \;+\; \left(\frac{r(t)}{\hat{r}(t)} + \alpha\right) e_i(t)\;. \tag{D.11}$$

The variable $g_i(t)$ combines the eligibility trace and the reward, and averages over the time scale τ_g. α is an arbitrary constant offset on the reward signal. In our simulations, this offset α was chosen slightly above 0 ($\alpha = 0.02$) such that small parameter changes were also present without any reward, as observed in (Yagishita et al., 2014). In the next section we show that $g_i(t)$ approximates the gradient of the expected future reward with respect to the synaptic parameter, i.e. $g_i(t) \approx \frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ for all $t > \tau_g$. In our simulations we found that incorporating the low-pass filtered eligibility traces (Eq. (D.11)) into the synaptic parameters works significantly better than using the eligibility traces directly for weight updates, although the latter approach was taken in a number of previous studies (see e.g. (Pfister et al., 2006; Legenstein et al., 2008; Urbanczik and Senn, 2009)).

$\hat{r}(t)$ in Eq. (D.11) is a low-pass filtered version of $r(t)$ that scales the synaptic updates. It was implemented through $\tau_g \frac{d\hat{r}(t)}{dt} = -\hat{r}(t) + r(t)$, with $\tau_g = 50$ s. This scaling of the reward signal has the following effect. If the current reward $r(t)$ exceeds the average reward $\hat{r}(t)$, the effect of the neuromodulatory signal $r(t)$ will be greater than 1. On the other hand, if the current reward is below average synaptic updates will be weighted by a term significantly lower than 1. Therefore, parameter updates are preferred for which the current reward signal exceeds the average.

Similar plasticity rules with eligibility traces in spiking neural networks have previously been proposed by several authors (Seung, 2003; Xie and Seung, 2004; Izhikevich, 2007; Pfister et al., 2006; Florian, 2007; Legenstein et al., 2008; Urbanczik and Senn, 2009). The main difference to these previous approaches is that the activity-dependent last term in Eq. (D.10) is scaled by the current synaptic weight $w_i(t)$. This weight-dependence of the update equations induces multiplicative synaptic dynamics and is a consequence of the exponential mapping Eq. (4.1) (see derivation in the next section). This is an important property for a network model that includes rewiring. Note, that for retracted synapses ($w_i(t) = 0$), both $e_i(t)$ and

$g_i(t)$ decay to zero (within few minutes in our simulations). Therefore, we find that the dynamics of retracted synapses is only driven by the first (prior) and last (random fluctuations) term of Eq. (4.5) and are independent from the network activity. Thus, retracted synapses spontaneously reappear also in the absence of reward after a random amount of time.

The first term in Eq. (4.5) is the gradient of the prior distribution. We used a prior distribution that pulls the synaptic parameters towards $\theta_i(t) = 0$ such that unused synapses tend to disappear and new synapses are permanently formed. Throughout all simulations we used independent Gaussian priors for the synaptic parameters

$$p_{\mathcal{S}}(\boldsymbol{\theta}) \;=\; \prod_i p_{\mathcal{S}}(\theta_i(t)) \;, \quad \text{with} \quad p_{\mathcal{S}}(\theta_i(t)) \;=\; \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\theta_i(t)-\mu)^2}{2\sigma^2}\right) \;,$$

where σ is the standard deviation of the prior distribution. Using this, we find that the contribution of the prior to the online parameter update equation is given by

$$\frac{\partial}{\partial\theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) \;=\; \frac{1}{\sigma^2}(\mu - \theta_i(t)) \;. \tag{D.12}$$

Finally by plugging Eq. (D.12) and (D.11) into Eq. (4.5) the synaptic parameter changes at time t are given by

$$d\theta_i(t) \;=\; \beta\left(\frac{1}{\sigma^2}(\mu - \theta_i(t)) \;+\; g_i(t)\right) dt \;+\; \sqrt{2\beta T}\, d\mathcal{W}_i \;. \tag{D.13}$$

If not stated otherwise we used $\sigma = 2$ and $\mu = 0$, and a learning rate of $\beta = 10^{-5}$. By inspecting Eq. (D.13) it becomes immediately clear that the parameter dynamics follow an Ornstein-Uhlenbeck process it the activity-dependent second therm is inactive (in the absence of reward), i.e. if $g_i(t) = 0$. In this case the dynamics are given by the deterministic drift towards the mean value μ and the stochastic diffusion fueled by the Wiener process $\mathcal{W}_i$. The temperature T and the standard deviation σ scale the contribution of these two forces.

Reward-modulated synaptic plasticity approximates gradient ascent on the expected discounted reward

We first consider a theoretical setup where the network is operated in arbitrarily long episodes such that in each episode a reward sequence $\boldsymbol{r}$ is encountered. The reward sequence $\boldsymbol{r}$ can be any discrete or real-valued function that is positive and bounded. The episodic scenario is useful to derive exact batch parameter update rules, from which we will then deduce online learning rules. Due to stochastic network inputs, stochastic network responses, and stochastic reward delivery, the reward sequence $\boldsymbol{r}$ is stochastic.

The classical goal of reinforcement learning is to maximize the function $\mathcal{V}(\boldsymbol{\theta})$ of discounted expected rewards Eq. (D.1). Policy gradient algorithms perform

gradient ascent on $\mathcal{V}(\boldsymbol{\theta})$ by changing each parameter θ_i in the direction of the gradient $\partial \log \mathcal{V}(\boldsymbol{\theta})/\partial\theta_i$. Here, we show that the parameter dynamics Eq. (D.10), (D.11) approximate this gradient, i.e., $g_i(t) \approx \partial \log \mathcal{V}(\boldsymbol{\theta})/\partial\theta_i$ for all $t > \tau_g$.

It is natural to assume that the reward signal $r(\tau)$ only depends indirectly on the parameters $\boldsymbol{\theta}$, through the history of network spikes $z_k(\tau)$ up to time τ, which we write as $\mathbf{z}(\tau) = \{z_k(s) \mid 0 \le s < \tau, 1 \le k \le K\}$, i.e., $p_{\mathcal{N}}(r(t), \mathbf{z}(t) \,|\, \boldsymbol{\theta}) = p\left(r(t) \mid \mathbf{z}(t)\right) p_{\mathcal{N}}\left(\mathbf{z}(t) \,|\, \boldsymbol{\theta}\right)$. We can first expand the expectation $\langle\, \cdot \,\rangle_{p(\boldsymbol{r}|\boldsymbol{\theta})}$ in Eq. (D.1) to be taken over the joint distribution $p(\boldsymbol{r}, \mathbf{z}|\boldsymbol{\theta})$ over reward sequences $\boldsymbol{r}$ and network trajectories $\mathbf{z}$. The derivative

$$\frac{\partial}{\partial\theta_i} \log \mathcal{V}(\boldsymbol{\theta}) \;=\; \frac{1}{\mathcal{V}(\boldsymbol{\theta})} \frac{\partial}{\partial\theta_i} \mathcal{V}(\boldsymbol{\theta}) \;=\; \frac{1}{\mathcal{V}(\boldsymbol{\theta})} \frac{\partial}{\partial\theta_i} \Big\langle \int_0^\infty e^{-\frac{\tau}{\tau_e}}\, r(\tau)\, d\tau \Big\rangle_{p(\boldsymbol{r},\mathbf{z}|\boldsymbol{\theta})} \tag{D.14}$$

can be evaluated using $\frac{\partial}{\partial x} \langle\, f(a) \,\rangle_{p(a|x)} = \langle\, f(a) \frac{\partial}{\partial x} \log p(a|x) \,\rangle_{p(a|x)}$:

$$\begin{aligned}
\frac{\partial}{\partial\theta_i} \log \mathcal{V}(\boldsymbol{\theta}) \;&=\; \frac{1}{\mathcal{V}(\boldsymbol{\theta})} \Big\langle \int_0^\infty e^{-\frac{\tau}{\tau_e}}\, r(\tau)\, \frac{\partial}{\partial\theta_i} \log p\left(r(\tau), \mathbf{z}(\tau) \mid \boldsymbol{\theta}\right)\, d\tau \Big\rangle_{p(\boldsymbol{r},\mathbf{z}|\boldsymbol{\theta})} \\
&=\; \frac{1}{\mathcal{V}(\boldsymbol{\theta})} \Big\langle \int_0^\infty e^{-\frac{\tau}{\tau_e}}\, r(\tau)\, \frac{\partial}{\partial\theta_i} \big(\log p\left(r(\tau) \mid \mathbf{z}(\tau)\right) + \\
&\qquad\qquad \log p_{\mathcal{N}}\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right) \big)\, d\tau \Big\rangle_{p(\boldsymbol{r},\mathbf{z}|\boldsymbol{\theta})} \qquad \text{(D.15)} \\
&=\; \Big\langle \int_0^\infty e^{-\frac{\tau}{\tau_e}}\, \frac{r(\tau)}{\mathcal{V}(\boldsymbol{\theta})}\, \frac{\partial}{\partial\theta_i} \log p_{\mathcal{N}}\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right)\, d\tau \Big\rangle_{p(\boldsymbol{r},\mathbf{z}|\boldsymbol{\theta})} . \qquad \text{(D.16)}
\end{aligned}$$

Here, $p_{\mathcal{N}}\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right)$ is the probability of observing the spike train $\mathbf{z}(\tau)$ in the time interval 0 to τ. For the definition of the network $\mathcal{N}$ given above, the gradient $\frac{\partial}{\partial\theta_i} \log p_{\mathcal{N}}\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right)$ of this distribution can be directly evaluated. Using Eq. (D.7) and (4.1) we get (Pfister et al., 2006)

$$\begin{aligned}
\frac{\partial}{\partial\theta_i} \log p_{\mathcal{N}}\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right) \;&=\; \frac{\partial w_i}{\partial\theta_i} \frac{\partial}{\partial w_i} \int_0^\tau z_{\text{POST}_i}(s)\, \log\left(f_{\text{POST}_i}(s)\right) - f_{\text{POST}_i}(s)\, ds \\
&\approx\; \int_0^\tau w_i\, y_{\text{PRE}_i}(s)\, \left(z_{\text{POST}_i}(s) - f_{\text{POST}_i}(s)\right) ds\,, \qquad \text{(D.17)}
\end{aligned}$$

where we have used that by construction only the rate function $f_{\text{POST}_i}(s)$ depends on the parameter θ_i.

In Eq. (D.17) we used the approximation $\frac{\partial w_i}{\partial\theta_i} \approx w_i$. This expression ignores the discontinuity of Eq. (4.1) at $\theta_i = 0$, where the function is not differentiable. In practice we found that this approximation is quite accurate if θ_0 is large enough such that $\exp(\theta_i - \theta_0)$ is close to zero (which is the case for $\theta_0 = 3$ in our simulation). In control experiments we also used a smooth function $w_i = \exp(\theta_i - \theta_0)$ (without the jump at $\theta_i = 0$) for which Eq. (D.17) is exact, and found that this yields results that are not significantly different from the ones that use the mapping Eq. (4.1).

D.5 Online learning

Eq. (D.16) defines a batch learning rule with an average taken over learning episodes where in each episode network responses and rewards are drawn according to the distribution $p(\boldsymbol{r}, \mathbf{z}|\boldsymbol{\theta})$. In a biological setting, there are typically no clear episodes but rather a continuous stream of network inputs and rewards and parameter updates are performed continuously (i.e., learning is online). The analysis of online policy gradient learning is far more complicated than the batch scenario, and typically only approximate results can be obtained that however perform well in practice, see e.g., (Seung, 2003; Xie and Seung, 2004) for discussions.

In order to arrive at an online learning rule for this scenario, we consider an estimator of Eq. (D.16) that approximates its value at each time $t > \tau_g$ based on the recent network activity and rewards during time $[t - \tau_g, t]$ for some suitable $\tau_g > 0$. We denote the estimator at time t by $G_i(t)$ where we want $G_i(t) \approx \frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ for all $t > \tau_g$. To arrive at such an estimator, we approximate the average over episodes in Eq. (D.16) by an average over time where each time point is treated as the start of an episode. The average is taken over a long sequence of network activity that starts at time t and ends at time $t + \tau_g$. Here, one systematic difference to the batch setup is that one cannot guarantee a time-invariant distribution over initial network conditions as we did there since those will depend on the current network parameter setting. However, under the assumption that the influence of initial conditions (such as initial membrane potentials and refractory states) decays quickly compared to the time scale of the environmental dynamics, it is reasonable to assume that the induced error is negligible. We thus rewrite Eq. (D.16) in the form (we use the abbreviation $PSP_i(s) = w_i(s)\, y_{\text{PRE}_i}(s)$).

$$\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta}) \;\approx\; G_i(t) = \\ \frac{1}{\tau_g} \int_t^{t+\tau_g} \int_\zeta^{t+\tau_g} e^{-\frac{\tau-\zeta}{\tau_e}}\, \frac{r(\tau)}{\mathcal{V}(\boldsymbol{\theta})} \int_\zeta^{\tau} PSP_i(s)\, (z_{\text{POST}_i}(s) - f_{\text{POST}_i}(s))\, ds\, d\tau\, d\zeta \;,$$

where τ_g is the length of the sequence of network activity over which the empirical expectation is taken. Finally, we can combine the second and third integral into a single one, rearrange terms and substitute s and τ so that integrals run into the past rather than the future, to obtain

$$G_i(t) \;\approx\; \frac{1}{\tau_g} \int_{t-\tau_g}^{t} \frac{r(\tau)}{\mathcal{V}(\boldsymbol{\theta})} \int_0^{\tau} e^{-\frac{s}{\tau_e}}\, PSP_i(\tau - s)\, (z_{\text{POST}_i}(\tau - s) - f_{\text{POST}_i}(\tau - s))\, ds\, d\tau \;, \tag{D.18}$$

We now discuss the relationship between $G_i(t)$ and Eq. (D.10), (D.11) to show that the latter equations approximate $G_i(t)$. Solving Eq. (D.10) with zero initial condition $e_i(0) = 0$ yields

$$e_i(t) = \int_0^t e^{-\frac{s}{\tau_e}}\, PSP_i(t - s)\, (z_{\text{POST}_i}(t - s) - f_{\text{POST}_i}(t - s))\, ds \;. \tag{D.19}$$

symbol	*value*	*description*
T	0.1	temperature
τ_e	1 s	time constant of eligibility trace
τ_g	50 s	time constant of gradient estimator
α	0.02	offset to reward signals
β	10^{-5}	learning rate
μ	0	mean of prior
σ	2	std of prior

Table D.1: Parameters of the synapse model Eq. (D.10)**,** (D.11) **and** (D.13)**.** Parameter values were found by fitting the experimental data of (Yagishita et al., 2014). If not stated otherwise, these values were used in all experiments.

This corresponds to the inner integral in Eq. (D.18) and we can write

$$G_i(t) \;\approx\; \frac{1}{\tau_g}\int_{t-\tau_g}^{t} \frac{r(\tau)}{\mathcal{V}(\boldsymbol{\theta})}\, e_i(\tau)\, d\tau \;=\; \left\langle \frac{r(t)}{\mathcal{V}(\boldsymbol{\theta})}\, e_i(t) \right\rangle_{\tau_g} \;\approx\; \left\langle \frac{r(t)}{\hat{r}(t)}\, e_i(t) \right\rangle_{\tau_g}, \tag{D.20}$$

where $\langle\cdot\rangle_{\tau_g}$ denotes the temporal average from $t-\tau_g$ to t and $\hat{r}(t)$ estimates the expected discounted reward through a slow temporal average.

Finally, we observe that any constant α can be added to $r(\tau)/\mathcal{V}(\boldsymbol{\theta})$ in Eq. (D.16) since

$$\Big\langle \int_0^\infty e^{-\frac{\tau}{\tau_e}}\, \alpha\, \frac{\partial}{\partial\theta_i} \log p_{\mathcal{N}}\left(\mathbf{z}(\tau)\,|\,\boldsymbol{\theta}\right)\; d\tau \Big\rangle_{p(\boldsymbol{r},\mathbf{z}|\boldsymbol{\theta})} = 0 \tag{D.21}$$

for any constant α (cf. (Williams, 1992; Urbanczik and Senn, 2009)).

Hence, we have $G_i(t) \;\approx\; \left\langle \left(\frac{r(t)}{\hat{r}(t)} + \alpha\right)\, e_i(t) \right\rangle_{\tau_g}$. Eq. (D.11) implements this in the form of a running average and hence $g_i(t) \approx G_i(t) \approx \frac{\partial}{\partial\theta_i}\log\mathcal{V}(\boldsymbol{\theta})$ for $t > \tau_g$. Note that this result assumes that the parameters $\boldsymbol{\theta}$ change slowly on the time-scale of τ_g. Simulations using the batch model outlined above and the online learning model showed qualitatively the same behavior for the parameters used in our experiments.

D.6 Simulation details

Simulations were preformed with NEST (Gewaltig and Diesmann, 2007) using an in-house implementation of the synaptic sampling model; additional tests were run in Matlab R2011b (Mathworks). The code/software described in the paper is freely available online at [redacted for double-blind review]. The differential equations of the neuron and synapse models were approximated using the Euler method, with fixed time steps $\Delta t = 1$ ms. All network variables were updated based on this time grid, except for the synaptic parameters $\theta_i(t)$ according to Eq. (D.13) which were updated only every 100 ms to reduce the computation time. Control experiments with $\Delta t = 0.1$ ms, and 1 ms update steps for all synaptic parameters

showed no significant differences. If not stated otherwise synaptic parameters were initially drawn from a Gaussian distribution with $\mu = -0.5$ and $\sigma = 0.5$ and the temperature was set to $T = 0.1$. Synaptic delays were 1 ms. Synaptic parameter changes were clipped at $\pm 4 \times 10^{-4}$ and synaptic parameters θ_i were not allowed to exceed the interval $[-2, 5]$ for the sake of numerical stability.

Details to: Task-dependent routing of synaptic connections through the interaction of stochastic spine dynamics with rewards

The number of potential excitatory synaptic connections between each pair of input and MSN neurons was initially drawn from a Binomial distribution ($p = 0.5$, $n = 10$). The connections then followed the reward-based synaptic sampling dynamics Eq. (4.5) as described above. Lateral inhibitory connections were fixed and thus not subject to learning. These connections between MSN neurons were drawn from a Bernoulli distribution with $p = 0.5$ and synaptic weights were drawn from a Gaussian distribution with $\mu = -1$ and $\sigma = 0.2$, truncated at zero. Two subsets of ten neurons were connected to either one of the targets T_1 or T_2.

To generate the input patterns we adapted the method from (Kappel et al., 2015a). The inputs were representations of a simple symbolic environment, realized by Poisson spike trains that encoded sensory experiences P_1 or P_2. The 200 input neurons were assigned to Gaussian tuning curves ($\sigma = 0.2$) with centers independently and equally scattered over the unit cube. The sensory experiences P_1 and P_2 were represented by two different, randomly selected points in this 3-dimensional space. The stimulus positions were overlaid with small-amplitude jitter ($\sigma = 0.05$). For each sensory experience the firing rate of an individual input neuron was given by the support of the sensory experience under the input neuron's tuning curve (maximum firing rate was 60 Hz). An additional offset of 2 Hz background noise was added. The lengths of the spike patterns were uniformly drawn from the interval [750 ms, 1500 ms]. The spike patterns were alternated with time windows (durations uniformly drawn from the interval [1000 ms, 2000 ms]) during which only background noise of 2 Hz was presented.

The network was rewarded if the assembly associated to the current sensory experience fired stronger than the other assembly. More precisely, we used a sliding window of 500 ms length to estimate the current output rate of the neural assemblies. Let $\hat{\nu}_1(t)$ and $\hat{\nu}_2(t)$ denote the estimated output rates of neural pools projecting to T_1 and T_2, respectively, at time t and let $I(t)$ be a function that indicates the identity of the input pattern at time t, i.e. $I(t) = 1$ if pattern P_1 was present and $I(t) = -1$ if pattern P_2 was present. If $I(t)(\hat{\nu}_1(t) - \hat{\nu}_2(t)) < 0$ the reward was set to $r(t) = 0$. Otherwise the reward signal was given by $r(t) = S\left(\frac{1}{5}(I(t)\hat{\nu}_1(t) - I(t)\hat{\nu}_2(t) - \nu_0)\right)$, where $\nu_0 = 25$ Hz is a soft firing threshold and $S(\cdot)$ denotes the logistic sigmoid function. The reward was recomputed every 10 ms. During the presentation of the background patterns no reward was delivered.

In Fig. 4.2D,E we tested our reward-gated synaptic plasticity mechanism with the reward-modulated STDP pairing protocol reported in (Yagishita et al., 2014). We applied the STDP protocol to 50 synapses and reported mean and s.e.m values of synaptic weight changes in Fig. 4.2D,E. Briefly, we presented 15 pre/post pairings; one per 10 seconds. In each pre/post pairing 10 presynaptic spikes were presented at a rate of 10 Hz. Each presynaptic spike was followed ($\Delta t = 10$ ms) by a brief postsynaptic burst of 3 spikes (100 Hz). The total duration of one pairing was thus 1 s indicated by the gray shaded rectangle in Fig. 4.2E. During the pairings the membrane potential was set to $u(t) = -2.4$ and Eq. (D.9),(D.10), (D.11) and (D.13) solved for each synapse. Reward was delivered here in the form of a rectangular-shaped wave of constant amplitude 1 and duration 300 ms to mimic puff application of dopamine. Rewards were delivered for each pre/post pairing and reward delays were relative to the onset of the STDP pairings. Parameters of the synapse model were chosen to qualitatively match the results of Fig. 1 of (Yagishita et al., 2014) (see Tab. D.1). These parameters were used in all experiments if not stated otherwise.

Synaptic parameter changes in Fig. 4.2G were measured by taking snapshots of the synaptic parameter vectors every 4 minutes. Parameter changes were measured in terms of the Euclidean norm of the difference between two successively recorded vectors. The values were then normalized by the maximum value of the whole experiment and averages over 5 trials were reported.

Details to: Bayesian perspective of policy sampling

Each trial run had a duration of 3 seconds. The current phase in the task was encoded by the activity of the input neurons (s_1 and s_2) and left-right decisions are made according to the activity of the output neuron a. For the first 200 ms of each trial state neuron s_1 was active (fires a burst with Poisson rate 100 Hz) which indicated approaching to the first junction. If the action neuron fired above a firing threshold of 60 Hz during this phase the right arm of the maze was taken and otherwise the left one. This phase was followed by a 200 ms time window where the state neurons remained silent. In the next phase neuron s_2 became active for another 200 ms. Here again the left-right decision at the second junction was based on the activity of the action neuron as in phase 1. This phase was followed by a 400 ms time window during which reward was presented. The reward amplitude was given by the value assigned to the exit that was taken (see Fig. 4.3A). The reward phase was followed by a waiting period until the end of the trial (3 seconds).

Initial synaptic parameters were drawn from a Gaussian distribution with $\mu = 0$ and $\sigma = 0.5$ and we used a different prior with $\mu = 0.5$ and $\sigma = 1$, to enhance regrowth of synaptic connections. An additional constant offset potential of $u_0 = 7$ was added to the output neuron's membrane potential to increase its activity.

Histograms of maze exists in Fig. 4.3D were computed over the last 100 trials of the experiments. Averages over 200 independent experiments are shown. Fig. 4.3E

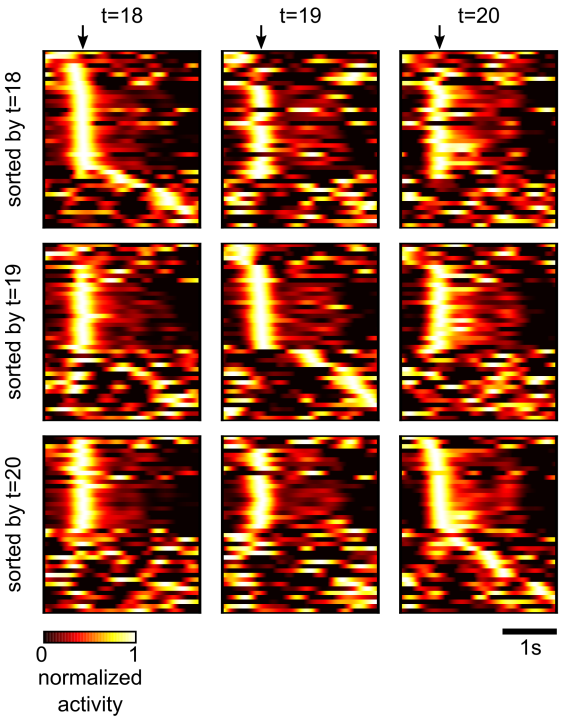

Fig. D.1: Drifts of neural codes while performance remained constant. Trial-averaged network activity as in Fig. 4.4D evaluated at three different times selected from a time window where the network performance was stable (see Fig. 4.4C). Each column shows the same trial-averaged activity plot but subject to different sorting. Rows correspond to one sorting criterion based on one evaluation time.

shows average reward values throughout one experiment of 4 hours learning time. Averages were taken over windows of 200 successive trial runs.

The surface plot in Fig. 4.3G was realized by setting the synaptic parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)$ of the network to fixed values, corresponding to the X- and Y-coordinates of the grid points and simulating the network for each parameter pair for 100 trials. The average reward over the 100 trials was assigned to the Z-coordinate. The trace of synaptic parameters of a single learning experiment (same as in Fig. 4.3F) was projected onto this surface (black trace).

Details to: A model for task-dependent self-configuration of a recurrent network of excitatory and inhibitory spiking neurons

Neuron and synapse parameters were as reported above, except for the inhibitory neurons for which we used faster dynamics with a refractory time $t_{\text{ref}} = 2$ ms and time constants $\tau_m = 10$ ms and $\tau_r = 1$ ms for the PSP kernel. The network connectivity between excitatory and inhibitory neurons was as suggested in (Avermann et al., 2012). Excitatory (pools D, U and hidden) and inhibitory neurons

were randomly connected with connection probabilities given in Table 2 in (Avermann et al., 2012). Connections include lateral inhibition between excitatory and inhibitory neurons. The connectivity to and from inhibitory neurons was kept fixed throughout the simulation (not subject to synaptic plasticity or rewiring). The connection probability from excitatory to inhibitory neurons was given by 0.575. The synaptic weights were drawn from a Gaussian distribution (truncated at zero) with $\mu = 0.5$ and $\sigma = 0.1$. Inhibitory neurons were connected to their targets with probability 0.6 (to excitatory neurons) and 0.55 (to inhibitory neurons) and the synaptic weights were drawn from a truncated normal distribution with $\mu = -1$ and $\sigma = 0.2$. The number of potential excitatory synaptic connections between each pair of excitatory neurons was drawn from a Binomial distribution ($p = 0.5$, $n = 10$). These connections were subject to the reward-based synaptic sampling and rewiring described above.

To infer the lever position from the network activity, we weighted spikes from the neuron pool D with -1 and spikes from U with $+1$, summed them and then filtered them with a long PSP kernel with $\tau_r = 50$ ms (rise) and $\tau_m = 500$ ms (decay). The cue input pattern was realized by the same method that was used to generate the patterns P_1 and P_2 outlined above. If a trial was completed successfully the reward signal $r(t)$ was set to 1 for 400 ms and was 0 otherwise. After each trial a short holding phase was inserted during which the input neurons were set to 2 Hz background noise. The lengths of these holding phases were uniformly drawn from the interval [1 s, 2 s]. In Fig. 4.4C-H the reward policy was changed after 24 hours by switching the decoding functions of the neural pools D and U and by randomly re-generating the input cue pattern.

To identify the movement onset times in Fig. 4.4D we adapted the method from (A. J. Peters et al., 2014). Lever movements were recorded at a sampling rate of 5 ms. Lever velocities were estimated by taking the difference between subsequent time steps and filtering them with a moving average filter of 5 time steps length. A Hilbert transform was applied to compute the envelope of the lever velocities. The movement onset time for each trial was then defined as the time point where the estimated lever velocity exceeded a threshold of 1.5 in the upward movement direction. If this value was never reached throughout the whole trial the time point of maximum velocity was used (most cases at learning onset).

The trial-averaged activity traces in Fig. 4.4D were generated by filtering the spiking activity of the network with a Gaussian kernel with $\sigma = 75$ ms. The activity traces were aligned with the movement onset times (indicated by black arrows in Fig. 4.4D) and averaged across 100 trials. The resulting activity traces were then normalized by the neuron's mean activity over all trials and values below the mean were clipped. The resulting activity traces were normalized to the unit interval.

Turnover statistics of synaptic connections in Fig. 4.4E were measured as follows. The synaptic parameters were recorded in intervals of 2 hours. The number of synapses that appeared (crossed the threshold of $\theta_i(t) = 0$ from below) or disappeared (crossed $\theta_i(t) = 0$ from above) between two measurements were counted

and the total number was reported as turnover rate.

For the approximation of simulating retracted potential synaptic connections in Fig. 4.4C,F we paused evaluation of the SDE (4.5) for $\theta_i \leq 0$. Instead, synaptic parameters of retracted connections where randomly set to values above zero after random waiting times drawn from an Exponential distribution with a mean of 12 hours. When a connection became functional at time t we set $\theta_i(t) = 10^{-5}$ and reset the eligibility trace $e_i(t)$ and gradient estimator $g_i(t)$ to zero and then continued the synaptic dynamics according to (4.5). Histograms in Fig. 4.4F were computed over bins of 2 hours width.

In Fig. D.1 we further analyzed the trial-averaged activity at three different time points (18 h, 19 h and 20 h) where the performance was stable (see Fig. 4.4C). Drifts of neural codes on fast time scales could also be observed during this phase of the experiment.

Details to: Compensation for network perturbations

The black curve in Fig. 4.4C shows the learning curve of a network for which rewiring was disabled after the task change at 24 h. Here, synaptic parameters were not allowed to cross the threshold at $\theta_i = 0$ and thus could not change sign after 24 h. Apart from this modification the synaptic dynamics evolved according to Eq. (D.13) as above with $T = 0.1$.

For the analysis of synaptic turnover in Fig. 4.4G we recorded the synaptic parameters at $t_1 = 24$ h and $t_2 = 48$ h. We then classified each potential synaptic connection i into one of four classes, stable non-functional: $(\theta_i(t_1) \leq 0) \wedge (\theta_i(t_2) \leq 0)$, transient decaying: $(\theta_i(t_1) > 0) \wedge (\theta_i(t_2) \leq 0)$, transient emerging: $(\theta_i(t_1) \leq 0) \wedge (\theta_i(t_2) > 0)$ and stable functional: $(\theta_i(t_1) > 0) \wedge (\theta_i(t_2) > 0)$.

In Fig. 4.4H we randomly selected 5% of the synaptic parameters θ_i and recorded their traces over a learning experiment of 48 hours (1 sample per minute). The principal component analysis (PCA) was then computed over these traces, treating the parameter vectors at each time point as one data sample. The high-dimensional trace was then projected to the first three principal components in Fig. 4.4H, and colored according to the average movement completion time that was acquired by the network at the corresponding time points.

Details to: Relative contribution of spontaneous and activity-dependent processes to synaptic plasticity

Synaptic weights in Fig. 4.5a,b were recorded in intervals of 10 minutes. We selected all pairs of synapses with common pre- and postsynaptic neurons as CI synapses and synapse pairs with the same post- but not the same presynaptic neuron as non-CI synapses. In Fig. 4.5d-f we took a snapshot of the synaptic weights after

48 hours of learning and computed the Pearson correlation of all CI and non-CI pairs for random subsets of around 5000 pairs. Data for 100 randomly chosen CI synapse pairs are plotted of Fig. 4.5E.

In Fig. 4.5F we analyzed the contribution of activity-dependent and spontaneous processes in our model. (Dvorkin and N. E. Ziv, 2016) reported that a certain degree of the stochasticity in their results could be attributed to their experimental setup. The maximum detectable correlation coefficient was limited to $0.76 - 0.78$, due to the variability of light fluorescence intensities which were used to estimate the sizes of postsynaptic densities. Since in our computer simulations we could directly read out values of the synaptic parameters we were not required to correct our results for noise sources in the experimental procedure (see p. 16ff and equations on p. 18 of (Dvorkin and N. E. Ziv, 2016)). This is also reflected in our data by the fact that we got a correlation coefficient that was close to 1.0 in the case $T = 0$ (see Fig. 4.5D). Following the procedure of (Dvorkin and N. E. Ziv, 2016) we estimated in our model the contributions of activity history dependent and spontaneous synapse-autonomous processes as in Fig. 8E of (Dvorkin and N. E. Ziv, 2016). Using the assumption of zero measurement error and thus a theoretically achievable maximum correlation coefficient of $r = 1.0$ we estimated the fraction of contributions of specific activity histories to synaptic changes (for $T = 0.15$) as $0.46 - 0.08 = 0.38$ and of spontaneous synapse-autonomous processes as $1.0 - 0.46 = 0.54$. The remaining 8% resulted from processes that were not specific to presynaptic input, but specific to the activity of the postsynaptic neuron (neuron-wide precesses).

Bibliography

Abbott, L. and K. Blum (1996). "Functional significance of long-term potentiation for sequence learning and prediction." In: *Cerebral Cortex* 6.3, pp. 406–416 (cit. on p. 40).

Abeles, M. (1991). *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press (cit. on p. 4).

Ackley, D. H., G. E. Hinton, and T. J. Sejnowski (1985). "A Learning Algorithm for Boltzmann Machines." In: *Cognitive Science* 9, pp. 147–169 (cit. on p. 51).

Ajemian, R., A. D'Ausilio, H. Moorman, and E. Bizzi (2013). "A theory for how sensorimotor skills are learned and retained in noisy and nonstationary neural circuits." In: *Proceedings of the National Academy of Sciences* 110.52, 5078–E5087 (cit. on p. 65).

Attardo, A., J. E. Fitzgerald, and M. J. Schnitzer (2015). "Impermanence of dendritic spines in live adult CA1 hippocampus." In: *Nature* 523.7562, pp. 592–596 (cit. on p. 65).

Avermann, M., C. Tomm, C. Mateo, W. Gerstner, and C. Petersen (2012). "Microcircuits of excitatory and inhibitory neurons in layer 2/3 of mouse barrel cortex." In: *Journal of Neurophysiology* 107.11, pp. 3116–3134 (cit. on pp. 10, 85, 148, 149).

Azevedo, F. A., L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, R. Lent, S. Herculano-Houzel, et al. (2009). "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain." In: *Journal of Comparative Neurology* 513.5, pp. 532–541 (cit. on p. 1).

Barber, M. J., J. W. Clark, and C. H. Anderson (2003). "Neural representation of probabilistic information." In: *Neural Computation* 15.8, pp. 1843–1864 (cit. on p. 6).

Barnett, M. W. and P. M. Larkman (2007). "The action potential." In: *Practical Neurology* 7.3, pp. 192–197 (cit. on p. 4).

Barone, P. and J. Joseph (1989). "Prefrontal cortex and spatial sequencing in macaque monkey." In: *Experimental Brain Research* 78.3, pp. 447–464 (cit. on pp. 24, 39).

Bartol Jr, T. M., C. Bromer, J. Kinney, M. A. Chirillo, J. N. Bourne, K. M. Harris, and T. J. Sejnowski (2015). "Nanoconnectomic upper bound on the variability of synaptic plasticity." In: *eLife* 4, e10778 (cit. on p. 88).

Baum, L. and T. Petrie (1966). "Statistical inference for probabilistic functions of finite state Markov chains." In: *The Annals of Mathematical Statistics* 37.6, pp. 1554–1563 (cit. on pp. 15, 36).

Baxter, J. and P. L. Bartlett (2000). "Direct gradient-based reinforcement learning." In: *The 2000 IEEE International Symposium on Circuits and Systems*. Vol. 3. IEEE, pp. 271–274 (cit. on pp. 71, 78).

Beers, R. J. van, P. Baraduc, and D. M. Wolpert (2002). "Role of uncertainty in sensorimotor control." In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 357.1424, pp. 1137–1145 (cit. on p. 3).

Beers, R. J. van, E. Brenner, and J. B. Smeets (2013). "Random walk of motor planning in task-irrelevant dimensions." In: *Journal of Neurophysiology* 109.4, pp. 969–977 (cit. on p. 70).

Beers, R. J. van, D. M. Wolpert, and P. Haggard (2001). "Sensorimotor integration compensates for visual localization errors during smooth pursuit eye movements." In: *Journal of Neurophysiology* 85.5, pp. 1914–1922 (cit. on p. 3).

Bellec, G., D. David Kappel, R. Legenstein, and W. Maass (2017). "Deep Rewiring: training very sparse deep networks." In: *arXiv preprint arXiv:1711.05136* (cit. on p. 6).

Berdyyeva, T. and C. Olson (2009). "Monkey supplementary eye field neurons signal the ordinal position of both actions and objects." In: *The Journal of Neuroscience* 29.3, p. 591 (cit. on pp. 12, 19).

Berger, T. K., R. Perin, G. Silberberg, and H. Markram (2009). "Frequency-dependent disynaptic inhibition in the pyramidal network - a ubiquitous pathway in the developing rat neocortex." In: *The Journal of Neurophysiology* 587.22, pp. 5411–5425 (cit. on p. 10).

Bhalla, U. S. and R. Iyengar (1999). "Emergent properties of networks of biological signaling pathways." In: *Science* 283.5400, pp. 381–387 (cit. on p. 50).

Bi, G. and M. Poo (1998). "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type." In: *Journal of Neuroscience* 18.24, pp. 10464–10472 (cit. on p. 66).

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer (cit. on pp. 11, 12, 14, 15, 18, 21, 32, 36, 45, 50, 52–54, 64, 108, 116).

Bobrowski, O., R. Meir, and Y. Eldar (2009). "Bayesian filtering in spiking neural networks: Noise, adaptation, and multisensory integration." In: *Neural Computation* 21.5, pp. 1277–1320 (cit. on p. 40).

Boerlin, M. and S. Deneve (2011). "Spike-based population coding and working memory." In: *PLoS Computational Biology* 7.2, e1001080 (cit. on p. 40).

Boerlin, M., C. K. Machens, and S. Deneve (2013). "Predictive coding of dynamical variables in balanced spiking networks." In: *PLoS Computational Biology* 9.11, e1003258 (cit. on p. 57).

Borst, J. G. G. (2010). "The low synaptic release probability in vivo." In: *Trends in Neurosciences* 33.6, pp. 259–266 (cit. on p. 1).

Botvinick, M. and M. Toussaint (2012). "Planning as inference." In: *Trends in Cognitive Sciences* 16.10, pp. 485–488 (cit. on p. 135).

Boucher, L. and Z. Dienes (2003). "Two ways of learning associations." In: *Cognitive Science* 27.6, pp. 807–842 (cit. on p. 39).

Brand, M. (1997). *Coupled hidden Markov models for modeling interacting processes*. Tech. rep. MIT Media Lab (cit. on pp. 25, 39).

Brea, J., W. Senn, and J.-P. Pfister (2011). "Sequence learning with hidden units in spiking neural networks." In: *Advances in Neural Information Processing Systems* 24, pp. 1422–1430 (cit. on pp. 11, 33, 36, 40, 41).

Brea, J., W. Senn, and J.-P. Pfister (2013). "Matching Recall and Storage in Sequence Learning with Spiking Neural Networks." In: *The Journal of Neuroscience* 33.23, pp. 9565–9575 (cit. on p. 57).

Buesing, L., J. Bill, B. Nessler, and W. Maass (2011). "Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons." In: *PLoS Computational Biology* 7.11, e1002211 (cit. on pp. 3, 4).

Buhry, L., A. Azizi, and S. Cheng (2011). "Reactivation, replay, and preplay: How it might all fit together." In: *Neural Plasticity* 2011, pp. 1–11 (cit. on p. 41).

Butz, M. and A. van Ooyen (2013). "A simple rule for dendritic spine and axonal bouton formation can account for cortical reorganization after focal retinal lesions." In: *PLoS Computational Biology* 9, e1003259 (cit. on p. 67).

Butz, M., I. D. Steenbuck, and A. van Ooyen (2014). "Homeostatic structural plasticity can account for topology changes following deafferentation and focal stroke." In: *Frontiers in Neuroanatomy* 8, p. 115 (cit. on p. 67).

Caporale, N. and Y. Dan (2008). "Spike timing-dependent plasticity: a Hebbian learning rule." In: *Annu Rev Neuroscience* 31, pp. 25–46 (cit. on pp. 10, 18).

Carandini, M. (2012). "From circuits to behavior: a bridge too far?" In: *Nature Neuroscience* 15.4, pp. 507–509 (cit. on pp. 57, 125).

Caroni, P., F. Donato, and D. Muller (2012). "Structural plasticity upon learning: regulation and functions." In: *Nature Reviews Neuroscience* 13.7, pp. 478–490 (cit. on pp. 46, 68).

Celeux, G. and J. Diebolt (1985). "The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem." In: *Computational Statistics Quarterly* 2.1, pp. 73–82 (cit. on pp. 11, 15).

Chaisanguanthum, K. S., H. H. Shen, and P. N. Sabes (2014). "Motor variability arises from a slow random walk in neural state." In: *Journal of Neuroscience* 34.36, pp. 12071–12080 (cit. on p. 70).

Chambers, A. R. and S. Rumpel (2017). "A stable brain from unstable components: Emerging concepts, implications for neural computation." In: *Neuroscience* 357, pp. 172–184 (cit. on p. 70).

Cheng, S. and L. M. Frank (2008). "New experiences enhance coordinated neural activity in the hippocampus." In: *Neuron* 57.2, p. 303 (cit. on p. 42).

Churchland, M., J. Cunningham, M. Kaufman, J. Foster, P. Nuyujukian, S. Ryu, and K. Shenoy (2012). "Neural population dynamics during reaching." In: *Nature* (cit. on pp. 27, 108).

Clark, A. (2013). "Whatever next? Predictive brains, situated agents, and the future of cognitive science." In: *Behavioral and Brain Sciences* 36.3, pp. 181–204 (cit. on pp. 6, 7).

Clarke, P. G. (2012). "The limits of brain determinacy." In: *Proceedings of the Royal Society of London B: Biological Sciences* 279.1734, pp. 1665–1674 (cit. on p. 1).

Cleeremans, A. and J. L. McClelland (1991). "Learning the structure of event sequences." In: *Memories, Thoughts, and Emotions: Essays in Honor of George Mandler* 120.3, pp. 235–253 (cit. on p. 39).

Coba, M. P., A. J. Pocklington, M. O. Collins, M. V. Kopanitsa, R. T. Uren, S. Swamy, M. D. R. Croning, J. S. Choudhary, and S. G. N. Grant (2009). "Neurotransmitters Drive Combinatorial Multistate Postsynaptic Density Networks." In: *Science Signaling* 2.68, pp. 1–11 (cit. on p. 49).

Collins, A. G. and M. J. Frank (2016). "Surprise! Dopamine signals mix action, value and error." In: *Nature Neuroscience* 19.1, p. 3 (cit. on p. 77).

Conway, C. and M. Christiansen (2005). "Modality-constrained statistical learning of tactile, visual, and auditory sequences." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 31.1, p. 24 (cit. on pp. 12, 30, 31, 35, 39, 109).

Cortes, C. and V. Vapnik (1995). "Support-vector networks." In: *Machine Learning* 20.3, pp. 273–297 (cit. on p. 108).

Deger, M., A. Seeholzer, and W. Gerstner (2016). "Multi-contact synapses for stable networks: a spike-timing dependent model of dendritic spine plasticity and turnover." In: *arXiv preprint arXiv:1609.05730* (cit. on p. 79).

Deger, M., M. Helias, S. Rotter, and M. Diesmann (2012). "Spike-timing dependence of structural plasticity explains cooperative synapse formation in the neocortex." In: *PLoS Computational Biology* 8.9, e1002689 (cit. on p. 67).

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1, pp. 1–38 (cit. on pp. 15, 18).

Deneve, S. (2008a). "Bayesian spiking neurons I: inference." In: *Neural Computation* 20.1, pp. 91–117 (cit. on p. 40).

Deneve, S. (2008b). "Bayesian spiking neurons II: learning." In: *Neural Computation* 20.1, pp. 118–145 (cit. on p. 40).

Desai, N. S., L. C. Rutherford, and G. G. Turrigiano (1999). "Plasticity in the intrinsic excitability of cortical pyramidal neurons." In: *Nature Neuroscience* 2.6, pp. 515–520 (cit. on p. 139).

Ding, M. and G. Rangarajan (2004). "First Passage Time Problem: A Fokker-Planck Approach." In: *New Directions in Statistical Physics*. Ed. by L. Wille. Springer, pp. 31–46 (cit. on p. 79).

Douglas, R. J. and K. A. Martin (2004). "Neuronal circuits of the neocortex." In: *Annual Review of Neuroscience* 27, pp. 419–451 (cit. on p. 10).

Doya, K., S. Ishii, A. Pouget, and R. P. N. Rao (2007). *Bayesian Brain: Probabilistic Approaches to Neural Coding*. MIT-Press (cit. on p. 46).

Doya, K. (2007). *Bayesian brain: Probabilistic approaches to neural coding*. MIT press (cit. on pp. 2, 3).

Driscoll, L. and C. Harvey (2016). "Dynamic reorganization of neuronal activity patterns in parietal cortex." In: *Proceedings of Cosyne 2016*, pp. 110–111 (cit. on pp. 70, 87, 92).

Dvorkin, R. and N. E. Ziv (2016). "Relative Contributions of Specific Activity Histories and Spontaneous Processes to Size Remodeling of Glutamatergic Synapses." In: *PLoS Biology* 14.10, e1002572 (cit. on pp. 1, 5, 70, 71, 88–91, 151).

Eagle, R. A. and A. Blake (1995). "Two-dimensional constraints on three-dimensional structure from motion tasks." In: *Vision Research* 35.20, pp. 2927–2941 (cit. on p. 3).

Ee, R. van, W. J. Adams, and P. Mamassian (2003). "Bayesian modeling of cue interaction: bistability in stereoscopic slant perception." In: *JOSA A* 20.7, pp. 1398–1406 (cit. on p. 3).

Eliasmith, C. and C. H. Anderson (2004). *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press (cit. on p. 6).

Elliott, T. and N. R. Shadbolt (1998). "Competition for neurotrophic factors: ocular dominance columns." In: *The Journal of Neuroscience* 18.15, pp. 5850–5858 (cit. on p. 67).

Elliott, T. and N. Shadbolt (1998). "Competition for neurotrophic factors: mathematical analysis." In: *Neural Computation* 10.8, pp. 1939–1981 (cit. on p. 67).

Elman, J. L. (1990). "Finding structure in time." In: *Cognitive Science* 14.2, pp. 179–211 (cit. on p. 39).

Engert, F. and T. Bonhoeffer (1999). "Dendritic spine changes associated with hippocampal long-term synaptic plasticity." In: *Nature* 399.6731, pp. 66–70 (cit. on p. 49).

Faisal, A. A., L. P. Selen, and D. M. Wolpert (2008). "Noise in the nervous system." In: *Nature Review Neuroscience* 9.4, pp. 292–303 (cit. on p. 1).

Fan, Y., D. Fricker, D. H. Brager, X. Chen, H.-C. Lu, R. A. Chitwood, and D. Johnston (2005). "Activity-dependent decrease of excitability in rat hippocampal neurons through increases in I_h." In: *Nature Neuroscience* 8.11, pp. 1542–1551 (cit. on p. 139).

Fauth, M., F. Wörgötter, and C. Tetzlaff (2015). "The Formation of Multi-synaptic Connections by the Interaction of Synaptic and Structural Plasticity and Their Functional Consequences." In: *PLoS Computational Biology* 11.1, e1004031 (cit. on p. 67).

Fiete, I. R., W. Senn, C. Z. Wang, and R. H. Hahnloser (2010). "Spike-time-dependent plasticity and heterosynaptic competition organize networks to produce long scale-free sequences of neural activity." In: *Neuron* 65.4, pp. 563–576 (cit. on p. 28).

Fiser, J., P. Berkes, G. Orbán, and M. Lengyel (2010). "Statistically optimal perception and learning: from behavior to neural representations." In: *Trends in Cognitive Sciences* 14.3, pp. 119–130 (cit. on p. 66).

Fiser, J., C. Chiu, and M. Weliky (2004). "Small modulation of ongoing cortical dynamics by sensory input during natural vision." In: *Nature* 431.7008, pp. 573–578 (cit. on p. 1).

Fitzgerald, T. H., R. J. Dolan, and K. J. Friston (2014). "Model averaging, optimal inference, and habit formation." In: *Frontiers in Human Neuroscience* 8, p. 457 (cit. on p. 7).

Florian, R. V. (2007). "Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity." In: *Neural Computation* 19.6, pp. 1468–1502 (cit. on pp. 78, 141).

Frémaux, N., H. Sprekeler, and W. Gerstner (2010). "Functional requirements for reward-modulated spike-timing-dependent plasticity." In: *Journal of Neuroscience* 30.40, pp. 13326–13337 (cit. on p. 73).

Friston, K. (2009). "The free-energy principle: a rough guide to the brain?" In: *Trends in Cognitive Sciences* 13.7, pp. 293–301 (cit. on p. 7).

Friston, K., J. Mattout, and J. Kilner (2011). "Action understanding and active inference." In: *Biological Cybernetics* 104.1, pp. 137–160 (cit. on p. 7).

Friston, K. (2008). "Variational filtering." In: *NeuroImage* 41.3, pp. 747–766 (cit. on p. 7).

Fujisawa, S., A. Amarasingham, M. Harrison, and G. Buzsáki (2008). "Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex." In: *Nature Neuroscience* 11.7, pp. 823–833 (cit. on pp. 10, 42).

Gardiner, C. (2004). *Handbook of Stochastic Methods*. 3rd ed. Springer (cit. on pp. 47, 74, 114, 115, 123, 138).

Gerstner, W. and W. M. Kistler (2002). *Spiking Neuron Models*. Cambridge: Cambridge University Press (cit. on pp. 4, 5, 14, 106, 107).

Gerstner, W., W. M. Kistler, R. Naud, and L. Paninski (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press (cit. on p. 139).

Gewaltig, M.-O. and M. Diesmann (2007). "NEST (NEural Simulation Tool)." In: *Scholarpedia* 2.4, p. 1430 (cit. on p. 145).

Gomez, R. and L. Gerken (1999). "Artificial grammar learning by 1-year-olds leads to specific and abstract knowledge." In: *Cognition* 70.2, pp. 109–135 (cit. on p. 30).

Grashow, R., T. Brookings, and E. Marder (2010). "Compensation for variable intrinsic neuronal excitability by circuit-synaptic interactions." In: *The Journal of Neuroscience* 30.27, pp. 9145–9156 (cit. on pp. 61, 92).

Gray, N. W., R. M. Weimer, I. Bureau, and K. Svoboda (2006). "Rapid redistribution of synaptic PSD-95 in the neocortex in vivo." In: *PLoS Biology* 4.11, e370 (cit. on pp. 49, 64).

Habenschuss, S., H. Puhr, and W. Maass (2013). "Emergence of optimal decoding of population codes through STDP." In: *Neural Computation* 25, pp. 1–37 (cit. on pp. 18, 38, 57, 59, 99–101, 126).

Habenschuss, S., J. Bill, and B. Nessler (2012). "Homeostatic plasticity in Bayesian spiking networks as Expectation Maximization with posterior constraints." In: *Advances in Neural Information Processing Systems*. Vol. 25, pp. 782–790 (cit. on pp. 30, 108, 109, 125).

Hahnloser, R. H., A. A. Kozhevnikov, and M. S. Fee (2002). "An ultra-sparse code underliesthe generation of neural sequences in a songbird." In: *Nature* 419.6902, pp. 65–70 (cit. on p. 28).

Haider, B., A. Duque, A. R. Hasenstaub, and D. A. McCormick (2006). "Neocortical network activity in vivo is generated through a dynamic balance of excitation and inhibition." In: *The Journal of Neuroscience* 26.17, pp. 4535–4545 (cit. on p. 38).

Han, F., N. Caporale, and Y. Dan (2008). "Reverberation of Recent Visual Experience in Spontaneous Cortical Waves." In: *Neuron* 60.2, pp. 321–327 (cit. on pp. 10, 12, 28, 39).

Harris, C. M. and D. M. Wolpert (1998). "Signal-dependent noise determines motor planning." In: *Nature* 394.6695, pp. 780–784 (cit. on p. 3).

Harvey, C. D., P. Coen, and D. W. Tank (2012). "Choice-specific sequences in parietal cortex during a virtual-navigation decision task." In: *Nature* 484.7392, pp. 62–68 (cit. on pp. 10, 65, 130, 131).

Hatfield, G. (2002). "Perception as Unconscious Inference." In: *Perception and the Physical World: Psychological and Philosophical Issues in Perception*. John Wiley & Sons, Ltd, pp. 115–143 (cit. on p. 46).

Helmholtz, H. von (1867). *Handbuch der physiologischen Optik*. Vol. 9. Voss (cit. on p. 3).

Helmholtz, H. von and J. P. C. Southall (2005). *Treatise on physiological optics*. Vol. 3. Courier Corporation (cit. on p. 3).

Hennequin, G., L. Aitchison, and M. Lengyel (2014). "Fast sampling-based inference in balanced neuronal networks." In: *Advances in Neural Information Processing Systems*, pp. 2240–2248 (cit. on p. 3).

Herculano-Houzel, S. (2012). "The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost." In: *Proceedings of the National Academy of Sciences* 109.Supplement 1, pp. 10661–10668 (cit. on p. 1).

Hinton, G. E. (2002). "Training products of experts by minimizing contrastive divergence." In: *Neural Computation* 14.8, pp. 1771–1800 (cit. on pp. 53, 120).

Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov (2012). "Improving neural networks by preventing co-adaptation of feature detectors." In: *arXiv preprint arXiv:1207.0580* (cit. on p. 67).

Hinton, G., S. Osindero, and Y.-W. Teh (2006). "A fast learning algorithm for deep belief nets." In: *Neural Computation* 18.7, pp. 1527–1554 (cit. on p. 52).

Ho, V. M., J. A. Lee, and K. C. Martin (2011). "The cell biology of synaptic plasticity." In: *Science* 334.6056, pp. 623–628 (cit. on p. 50).

Hodgkin, A. L., A. F. Huxley, and B. Katz (1952). "Measurement of current-voltage relations in the membrane of the giant axon of Loligo." In: *The Journal of Physiology* 116.4, pp. 424–448 (cit. on p. 4).

Hofer, S. B., T. D. Mrsic-Flogel, T. Bonhoeffer, and M. Hübener (2009). "Experience leaves a lasting structural trace in cortical circuits." In: *Nature* 457.7227, pp. 313–317 (cit. on p. 59).

Hoffman, K. and B. McNaughton (2002). "Coordinated reactivation of distributed memory traces in primate neocortex." In: *Science* 297.5589, pp. 2070–2073 (cit. on p. 41).

Holtmaat, A. J., J. T. Trachtenberg, L. Wilbrecht, G. M. Shepherd, X. Zhang, G. W. Knott, and K. Svoboda (2005). "Transient and persistent dendritic spines in the neocortex in vivo." In: *Neuron* 45.2, pp. 279–291 (cit. on pp. 1, 54, 57, 64, 70, 73, 92).

Holtmaat, A. and K. Svoboda (2009). "Experience-dependent structural synaptic plasticity in the mammalian brain." In: *Nature Reviews Neuroscience* 10.9, pp. 647–658 (cit. on pp. 44, 54, 64, 70, 92).

Holtmaat, A., L. Wilbrecht, G. W. Knott, E. Welker, and K. Svoboda (2006). "Experience-dependent and cell-type-specific spine growth in the neocortex." In: *Nature* 441.7096, pp. 979–983 (cit. on pp. 73, 77).

Hopfield, J. J. and C. D. Brody (2001). "What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration." In: *Proceedings of the National Academy of Sciences* 98.3, pp. 1282–1287 (cit. on p. 129).

Huang, Y. and R. P. Rao (2011). "Predictive coding." In: *Wiley Interdisciplinary Reviews: Cognitive Science* 2.5, pp. 580–593 (cit. on p. 7).

Isoda, M. and J. Tanji (2003). "Contrasting neuronal activity in the supplementary and frontal eye fields during temporal organization of multiple saccades." In: *Journal of Neurophysiology* 90.5, p. 3054 (cit. on p. 19).

Izhikevich, E. M. (2007). "Solving the distal reward problem through linkage of STDP and dopamine signaling." In: *Cerebral Cortex* 17.10, pp. 2443–2452 (cit. on pp. 33, 78, 105, 140, 141).

Ji, D. and M. Wilson (2007). "Coordinated memory replay in the visual cortex and hippocampus during sleep." In: *Nature Neuroscience* 10.1, pp. 100–107 (cit. on pp. 10, 41).

Jin, D. Z., N. Fujii, and A. M. Graybiel (2009). "Neural representation of time in cortico-basal ganglia circuits." In: *Proceedings of the National Academy of Sciences* 106.45, pp. 19156–19161 (cit. on p. 28).

Jolivet, R., A. Rauch, H.-R. Lüscher, and W. Gerstner (2006). "Predicting spike timing of neocortical pyramidal neurons by simple threshold models." In: *Journal of Computational Neuroscience* 21.1, pp. 35–49 (cit. on pp. 14, 124, 125).

Jordan, M. I. (1997). "Serial order: A parallel distributed processing approach." In: *Advances in Psychology* 121, pp. 471–495 (cit. on p. 39).

Kappel, D., S. Habenschuss, R. Legenstein, and W. Maass (2015a). "Network Plasticity as Bayesian Inference." In: *PLoS Computational Biology* 11.11, e1004485 (cit. on pp. 6, 71, 138, 146).

Kappel, D., S. Habenschuss, R. Legenstein, and W. Maass (2015b). "Synaptic sampling: A Bayesian approach to neural network plasticity and rewiring." In: *Advances in Neural Information Processing Systems*, pp. 370–378 (cit. on p. 6).

Kappel, D., B. Nessler, and W. Maass (2014). "STDP Installs in Winner-Take-All Circuits an Online Approximation to Hidden Markov Model Learning." In: *PLoS Computational Biology* 10.3, e1003511 (cit. on pp. 57, 62, 127, 131).

Kasai, H., M. Fukuda, S. Watanabe, A. Hayashi-Takagi, and J. Noguchi (2010). "Structural dynamics of dendritic spines in memory and cognition." In: *Trends in Neurosciences* 33.3, pp. 121–129 (cit. on p. 70).

Kasthuri, N., K. J. Hayworth, D. R. Berger, R. L. Schalek, J. A. Conchello, D. L. Knowles-Barley, A. Vázquez-Reina, V. Kaynig, T. R. Jones, M. Roberts, J. L. Morgan, J. C. Tapia, H. S. Seung, W. G. Roncal, J. T. Vogelstein, R. Burns, D. L. Sussman, C. E. Priebe, H. Pfister, and J. Lichtman (2015). "Saturated reconstruction of a volume of neocortex." In: *Cell* 3, pp. 648–661 (cit. on pp. 70, 88, 91).

Keck, C., C. Savin, and J. Lücke (2012). "Feedforward Inhibition and Synaptic Scaling–Two Sides of the Same Coin?" In: *PLoS Computational Biology* 8.3, e1002432 (cit. on p. 38).

Kirkpatrick, S., M. Vecchi, et al. (1983). "Optimization by simulated annealing." In: *Science* 220.4598, pp. 671–680 (cit. on p. 75).

Klampfl, S. and W. Maass (2013). "Emergence of dynamic memory traces in cortical microcircuit models through STDP." In: *Journal of Neuroscience* 33.28, pp. 11515–11529 (cit. on pp. 39, 59, 129).

Knill, D. C. (1998). "Discrimination of planar surface slant from texture: human and ideal observers compared." In: *Vision Research* 38.11, pp. 1683–1711 (cit. on p. 3).

Knill, D. C. and A. Pouget (2004). "The Bayesian brain: the role of uncertainty in neural coding and computation." In: *Trends in Neurosciences* 27.12, pp. 712–719 (cit. on pp. 2, 3).

Knoblauch, A., E. Körner, U. Körner, and F. T. Sommer (2014). "Structural Synaptic Plasticity Has High Memory Capacity and Can Explain Graded Amnesia, Catastrophic Forgetting, and the Spacing Effect." In: *PLoS One* 9.5, e96485 (cit. on p. 67).

Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning)*. MIT Press (cit. on pp. 11, 12, 15, 18, 25, 32, 101).

Kozhevnikov, A. A. and M. S. Fee (2007). "Singing-related activity of identified HVC neurons in the zebra finch." In: *Journal of Neurophysiology* 97.6, pp. 4271–4283 (cit. on p. 28).

Kuhlman, S. J., D. H. O'Connor, K. Fox, and K. Svoboda (2014). "Structural plasticity within the barrel cortex during initial phases of whisker-dependent learning." In: *The Journal of Neuroscience* 34.17, pp. 6078–6083 (cit. on p. 59).

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11, pp. 2278–2324 (cit. on pp. 120, 129).

Lee, T. S. and D. Mumford (2003). "Hierarchical Bayesian inference in the visual cortex." In: *JOSA A* 20.7, pp. 1434–1448 (cit. on p. 7).

Legenstein, R., D. Pecevski, and W. Maass (2008). "A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback." In: *PLoS Computational Biology* 4.10, e1000180 (cit. on pp. 78, 141).

Liao, D., A. Jones, and R. Malinow (1992). "Direct measurement of quantal changes underlying long-term potentiation in CA1 hippocampus." In: *Neuron* 9.6, pp. 1089–1097 (cit. on p. 66).

Liu, Z., D. C. Knill, and D. Kersten (1995). "Object classification for human and ideal observers." In: *Vision Research* 35.4, pp. 549–568 (cit. on p. 3).

Lochmann, T. and S. Deneve (2011). "Neural processing as causal inference." In: *Current Opinion in Neurobiology* (cit. on p. 41).

Loewenstein, Y., A. Kuras, and S. Rumpel (2011). "Multiplicative dynamics underlie the emergence of the log-normal distribution of spine sizes in the neocortex in vivo." In: *The Journal of Neuroscience* 31.26, pp. 9481–9488 (cit. on pp. 1, 54, 55, 64, 70, 73, 75, 92).

Loewenstein, Y., U. Yanover, and S. Rumpel (2015). "Predicting the dynamics of network connectivity in the neocortex." In: *The Journal of Neuroscience* 35.36, pp. 12535–12544 (cit. on pp. 54, 57, 64, 66, 70, 75, 92, 124).

Luczak, A., P. Barth'o, and K. D. Harris (2009). "Spontaneous events outline the realm of possible sensory responses in neocortical populations." In: *Neuron* 62.3, pp. 413–425 (cit. on pp. 10, 12, 24, 107).

Luczak, A., P. Barthó, S. L. Marguet, G. Buzáki, and K. D. Harris (2007). "Sequential structure of neocortical spontaneous activity in vivo." In: *PNAS* 104.1, pp. 347–352 (cit. on pp. 10, 12, 28, 39).

Maass, W. (1997). "Networks of spiking neurons: the third generation of neural network models." In: *Neural Networks* 10.9, pp. 1659–1671 (cit. on p. 4).

Maass, W. (2014). "Noise as a resource for computation and learning in networks of spiking neurons." In: *Proceedings of the IEEE* 102.5, pp. 860–880 (cit. on p. 1).

MacKay, D. J. (1992). "Bayesian interpolation." In: *Neural Computation* 4.3, pp. 415–447 (cit. on pp. 45, 52–54, 64).

Marder, E. and A. L. Taylor (2011). "Multiple models to capture the variability in biological neurons and networks." In: *Nature Neuroscience* 14.2, pp. 133–138 (cit. on p. 61).

Marder, E. (2011). "Variability, compensation, and modulation in neurons and circuits." In: *Proceedings of the National Academy of Sciences* 108.Supplement 3, pp. 15542–15548 (cit. on pp. 46, 61, 68, 92).

Marder, E. and J.-M. Goaillard (2006). "Variability, compensation and homeostasis in neuron and network function." In: *Nature Reviews Neuroscience* 7.7, pp. 563–574 (cit. on pp. 68, 92).

Markram, H., W. Gerstner, and P. J. Sjöström (2011). "A history of spike-timing-dependent plasticity." In: *Frontiers in Synaptic Neuroscience* 3, pp. 1–4 (cit. on pp. 10, 18).

Markram, H., M. Toledo-Rodriguez, Y. Wang, A. Gupta, G. Silberberg, and C. Wu (2004). "Interneurons of the neocortical inhibitory system." In: *Nature Review Neuroscience* 5.10, pp. 793–807 (cit. on p. 67).

Markram, H., Y. Wang, and M. Tsodyks (1998). "Differential signaling via the same axon of neocortical pyramidal neurons." In: *PNAS* 95, pp. 5323–5328 (cit. on p. 67).

Marr, D. and T. Poggio (1976). *From understanding computation to understanding neural circuitry*. Tech. rep. Cambridge, MA, USA: Massachusetts Institute of Technology (cit. on pp. 7, 91).

Matsuzaki, M., G. C. Ellis-Davies, T. Nemoto, Y. Miyashita, M. Iino, and H. Kasai (2001). "Dendritic spine geometry is critical for AMPA receptor expression in hippocampal CA1 pyramidal neurons." In: *Nature Neuroscience* 4.11, pp. 1086–1092 (cit. on p. 76).

May, A. (2011). "Experience-dependent structural plasticity in the adult human brain." In: *Trends in Cognitive Sciences* 15.10, pp. 475–482 (cit. on pp. 46, 68).

McDonnell, M. D. and L. M. Ward (2011). "The benefits of noise in neural systems: bridging theory and experiment." In: *Nature Reviews Neuroscience* 12.7, pp. 415–426 (cit. on p. 1).

Mensi, S., R. Naud, and W. Gerstner (2011). "From stochastic nonlinear integrate-and-fire to generalized linear models." In: *Advances in Neural Information Processing Systems*. Vol. 24, pp. 1377–1385 (cit. on p. 124).

Minerbi, A., R. Kahana, L. Goldfeld, M. Kaufman, S. Marom, and N. E. Ziv (2009). "Long-term relationships between synaptic tenacity, synaptic remodeling, and network activity." In: *PLoS Biology* 7.6, e1000136 (cit. on p. 70).

Mongillo, G. and S. Deneve (2008). "Online learning with hidden Markov models." In: *Neural Computation* 20.7, pp. 1706–1716 (cit. on p. 40).

Montgomery, J. M., P. Pavlidis, and D. V. Madison (2001). "Pair recordings reveal all-silent synaptic connections and the postsynaptic expression of long-term potentiation." In: *Neuron* 29.3, pp. 691–701 (cit. on p. 66).

Murty, M. N. and V. S. Devi (2011). *Hidden Markov Models*. Springer (cit. on pp. 11, 14).

Natschlaeger, T., W. Maass, and A. Zador (2001). "Efficient Temporal Processing with Biologically Realistic Dynamic Synapses." In: *Network: Computation in Neural Systems* 12, pp. 75–87 (cit. on p. 67).

Neal, R. M. (1993). *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Tech. rep. University of Toronto Department of Computer Science (cit. on p. 32).

Neal, R. M. and G. E. Hinton (1998). "A view of the EM algorithm that justifies incremental sparse, and other variants." In: *Learning in Graphical Models*. Ed. by M. I. Jordan. Kluwer Academic Press (cit. on pp. 11, 15).

Neal, R. M. (2012). *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media (cit. on p. 2).

Nessler, B., M. Pfeiffer, and W. Maass (2010). "STDP enables spiking neurons to detect hidden causes of their inputs." In: *Proceedings of NIPS, Advances in Neural Information Processing Systems* 22, pp. 1357–1365 (cit. on pp. 18, 38, 99–101).

Nessler, B., M. Pfeiffer, L. Buesing, and W. Maass (2013). "Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity." In: *PLoS Computational Biology* 9.4, e1003037 (cit. on pp. 3, 5, 11, 18, 19, 38, 57–59, 99, 100, 125).

O'Donnell, C., M. F. Nolan, and M. C. van Rossum (2011). "Dendritic spine dynamics regulate the long-term stability of synaptic plasticity." In: *The Journal of Neuroscience* 31.45, pp. 16142–16156 (cit. on p. 66).

Okun, M. and I. Lampl (2008). "Instantaneous correlation of excitation and inhibition during ongoing and sensory-evoked activities." In: *Nature Neuroscience* 11.5, pp. 535–537 (cit. on pp. 10, 38).

Ooyen, A. van and e. Butz-Ostendorf Markus (2017). *The Rewiring Brain*. Academic Press (cit. on p. 70).

Packer, A. M., B. Roska, and M. Häusser (2013). "Targeting neurons and photons for optogenetics." In: *Nature Neuroscience* 16.7, pp. 805–815 (cit. on p. 1).

Pecevski, D., L. Buesing, and W. Maass (2011). "Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons." In: *PLoS Computational Biology* 7.12, e1002294 (cit. on p. 3).

Pecevski, D., D. Kappel, and Z. Jonke (2014). "NEVESIM: Event-Driven Neural Simulation Framework with a Python Interface." In: *Frontiers in Neuroinformatics* 8, p. 70 (cit. on p. 6).

Pecevski, D. and W. Maass (2016). "Learning Probabilistic Inference through Spike-Timing-Dependent Plasticity." In: *eNeuro* 3.2, ENEURO–0048 (cit. on p. 3).

Peters, A. J., S. X. Chen, and T. Komiyama (2014). "Emergence of reproducible spatiotemporal activity during motor learning." In: *Nature* 510.7504, pp. 263–267 (cit. on pp. 83, 85–87, 91, 149).

Peters, J. and S. Schaal (2006). "Policy gradient methods for robotics." In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 2219–2225 (cit. on pp. 71, 78).

Peyrache, A., M. Khamassi, K. Benchenane, S. Wiener, and F. Battaglia (2009). "Replay of rule-learning related neural patterns in the prefrontal cortex during sleep." In: *Nature Neuroscience* 12.7, pp. 919–926 (cit. on p. 42).

Pfister, J.-P., T. Toyoizumi, D. Barber, and W. Gerstner (2006). "Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning." In: *Neural Computation* 18.6, pp. 1318–1348 (cit. on pp. 78, 141, 143).

Pothos, E. M. (2007). "Theories of artificial grammar learning." In: *Psychological Bulletin* 133.2, p. 227 (cit. on p. 39).

Pouget, A., J. M. Beck, W. J. Ma, and P. E. Latham (2013). "Probabilistic brains: knowns and unknowns." In: *Nature Neuroscience* 16.9, pp. 1170–1178 (cit. on pp. 45, 46, 64, 68).

Pouget, A., P. Dayan, and R. S. Zemel (2003). "Inference and computation with population codes." In: *Annual Review of Neuroscience* 26.1, pp. 381–410 (cit. on p. 6).

Prinz, A. A., D. Bucher, and E. Marder (2004). "Similar network activity from disparate circuit parameters." In: *Nature Neuroscience* 7.12, pp. 1345–1352 (cit. on pp. 61, 68, 92).

Rabiner, L. R. (1989). "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." In: *Proceedings of the IEEE* 77.2, pp. 257–286 (cit. on pp. 11, 14, 21).

Rao, R. P. N., B. A. Olshausen, and M. S. Lewicki (2002). *Probabilistic Models of the Brain*. MIT Press (cit. on p. 46).

Rao, R. P. (2004). "Bayesian computation in recurrent neural circuits." In: *Neural Computation* 16.1, pp. 1–38 (cit. on pp. 6, 40).

Rao, R. and D. Ballard (1999). "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects." In: *Nature Neuroscience* 2, pp. 79–87 (cit. on p. 7).

Rao, R. and T. Sejnowski (2001). "Spike-timing-dependent Hebbian plasticity as temporal difference learning." In: *Neural Computation* 13.10, pp. 2221–2237 (cit. on p. 40).

Rawlik, K., M. Toussaint, and S. Vijayakumar (2013). "On stochastic optimal control and reinforcement learning by approximate inference." In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. AAAI Press, pp. 3052–3056 (cit. on pp. 92, 135).

Reber, A. S. (1967). "Implicit learning of artificial grammars." In: *Journal of Verbal Learning and Verbal Behavior* 6.6, pp. 855–863 (cit. on p. 31).

Remme, M. W. and W. J. Wadman (2012). "Homeostatic scaling of excitability in recurrent neural networks." In: *PLoS Computational Biology* 8.5, e1002494 (cit. on p. 140).

Rezende, D. J., D. Wierstra, and W. Gerstner (2011). "Variational Learning for Recurrent Spiking Networks." In: *Advances in Neural Information Processing*

Systems 24. Ed. by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, pp. 136–144 (cit. on pp. 11, 40).

Ribeiro, S., D. Gervasoni, E. Soares, Y. Zhou, S. Lin, J. Pantoja, M. Lavine, and M. Nicolelis (2004). "Long-lasting novelty-induced neuronal reverberation during slow-wave sleep in multiple forebrain areas." In: *PLoS Biology* 2.1, e24 (cit. on p. 42).

Ribrault, C., K. Sekimoto, and A. Triller (2011). "From the stochasticity of molecular processes to the variability of synaptic transmission." In: *Nature Reviews Neuroscience* 12.7, pp. 375–387 (cit. on pp. 49, 64).

Rigotti, M., O. Barak, M. R. Warden, X.-J. Wang, N. D. Daw, E. K. Miller, and S. Fusi (2013). "The importance of mixed selectivity in complex cognitive tasks." In: *Nature* (advance online publication), pp. 1476–4687 (cit. on pp. 12, 24, 39).

Rokni, U., A. G. Richardson, E. Bizzi, and H. S. Seung (2007). "Motor learning with unstable neural representations." In: *Neuron* 54.4, pp. 653–666 (cit. on pp. 44, 63–65).

Rueckert, E., D. Kappel, D. Tanneberg, D. Pecevski, and J. Peters (2016). "Recurrent spiking networks solve planning tasks." In: *Scientific Reports* 6, p. 21142 (cit. on p. 6).

Rumpel, S. and J. Triesch (2016). "The dynamic connectome." In: *e-Neuroforum* 7.3, pp. 48–53 (cit. on pp. 1, 70, 80, 92).

Salakhutdinov, R. and G. Hinton (2012). "An Efficient Learning Procedure for Deep Boltzmann Machines." In: *Neural Computation* 24, pp. 1967–2006 (cit. on p. 52).

Sasaki, T., N. Matsuki, and Y. Ikegaya (2011). "Action-potential modulation during axonal conduction." In: *Science* 331.6017, pp. 599–601 (cit. on p. 4).

Savin, C. and S. Deneve (2014). "Spatio-temporal representations of uncertainty in spiking neural networks." In: *Advances in Neural Information Processing Systems*, pp. 2024–2032 (cit. on p. 3).

Schölkopf, B., C. J. Burges, and A. J. Smola (1999). *Advances in kernel methods: support vector learning*. The MIT press (cit. on p. 108).

Sejnowski, T. J., P. S. Churchland, and J. A. Movshon (2014). "Putting big data to good use in neuroscience." In: *Nature Neuroscience* 17.11, pp. 1440–1441 (cit. on p. 1).

Seung, H. S. (2003). "Learning in spiking neural networks by reinforcement of stochastic synaptic transmission." In: *Neuron* 40.6, pp. 1063–1073 (cit. on pp. 78, 141, 144).

Shi, L. and T. Griffiths (2009). "Neural implementation of hierarchical Bayesian inference by importance sampling." In: *Proceedings of NIPS, Advances in Neural Information Processing Systems* 22, pp. 1669–1677 (cit. on p. 11).

Shima, K., M. Isoda, H. Mushiake, and J. Tanji (2006). "Categorization of behavioural sequences in the prefrontal cortex." In: *Nature* 445.7125, pp. 315–318 (cit. on p. 24).

Shima, K. and J. Tanji (2000). "Neuronal activity in the supplementary and presupplementary motor areas for temporal organization of multiple movements." In: *Journal of Neurophysiology* 84.4, pp. 2148–2160 (cit. on pp. 19, 24, 39).

Sjöström, P. J., G. G. Turrigiano, and S. Nelson (2001). "Rate, timing, and cooperativity jointly determine cortical synaptic plasticity." In: *Neuron* 32.6, pp. 1149–1164 (cit. on pp. 58, 59, 66).

Solway, A. and M. Botvinick (2012). "Goal-Directed Decision Making as Probabilistic Inference: A Computational Framework and Potential Neural Correlates." In: *Psychological Review* 119.1, pp. 120–154 (cit. on p. 82).

Statman, A., M. Kaufman, A. Minerbi, N. E. Ziv, and N. Brenner (2014). "Synaptic Size Dynamics as an Effectively Stochastic Process." In: *PLoS Computational Biology* 10.10, e1003846 (cit. on pp. 70, 75).

Stettler, D. D., H. Yamahachi, W. Li, W. Denk, and C. D. Gilbert (2006). "Axons and synaptic boutons are highly dynamic in adult visual cortex." In: *Neuron* 49.6, pp. 877–887 (cit. on pp. 44, 64, 70, 92).

Stiller, J. and G. Radons (1999). "Online estimation of hidden Markov models." In: *IEEE Sign. Process. Lett*. Citeseer (cit. on p. 40).

Sutton, R. S. and A. G. Barto (1998). *Reinforcement learning: An introduction*. Vol. 1. 1. MIT press Cambridge (cit. on p. 140).

Tang, L. S., M. L. Goeritz, J. S. Caplan, A. L. Taylor, M. Fisek, and E. Marder (2010). "Precise temperature compensation of phase in a rhythmic motor pattern." In: *PLoS Biology* 8.8, e1000469 (cit. on pp. 61, 92).

Tao, C., G. Zhang, Y. Xiong, and Y. Zhou (2015). "Functional dissection of synaptic circuits: in vivo patch-clamp recording in neuroscience." In: *Frontiers in Neural Circuits* 9, p. 23 (cit. on p. 1).

Trachtenberg, J. T., B. E. Chen, G. W. Knott, G. Feng, J. R. Sanes, E. Welker, and K. Svoboda (2002). "Long-term in vivo imaging of experience-dependent synaptic plasticity in adult cortex." In: *Nature* 420.6917, pp. 788–794 (cit. on p. 54).

Urbanczik, R. and W. Senn (2009). "Reinforcement learning in populations of spiking neurons." In: *Nature Neuroscience* 12.3, pp. 250–252 (cit. on pp. 78, 141, 145).

Varela, J. A., K. Sen, J. Gibson, J. Fost, L. F. Abbott, and S. B. Nelson (1997). "A quantitative description of short-term plasticity at excitatory synapses in layer 2/3 of rat primary visual cortex." In: *Journal of Neuroscience* 17, pp. 220–224 (cit. on p. 67).

Vlassis, N., M. Ghavamzadeh, S. Mannor, and P. Poupart (2012). "Bayesian reinforcement learning." In: *Reinforcement Learning*. Springer, pp. 359–386 (cit. on pp. 92, 135).

Wan, L., M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus (2013). "Regularization of neural networks using dropconnect." In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 1058–1066 (cit. on p. 67).

Warden, M. R. and E. K. Miller (2010). "Task-dependent changes in short-term memory in the prefrontal cortex." In: *The Journal of Neuroscience* 30.47, pp. 15801–15810 (cit. on pp. 12, 19).

Welling, M. and Y. W. Teh (2011). "Bayesian learning via stochastic gradient Langevin dynamics." In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688 (cit. on p. 51).

Williams, R. J. (1992). "Simple statistical gradient-following algorithms for connectionist reinforcement learning." In: *Machine Learning* 8.3-4, pp. 229–256 (cit. on pp. 71, 78, 145).

Winkler, I., S. Denham, R. Mill, T. M. Böhm, and A. Bendixen (2012). "Multistability in auditory stream segregation: a predictive coding view." In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 367.1591, pp. 1001–1012 (cit. on p. 47).

Wolpert, D. M., Z. Ghahramani, and M. I. Jordan (1995). "An internal model for sensorimotor integration." In: *Science*, pp. 1880–1882 (cit. on p. 3).

Xie, X. and H. S. Seung (2004). "Learning in neural networks by reinforcement of irregular spiking." In: *Physical Review E* 69.4, p. 041909 (cit. on pp. 78, 141, 144).

Xiong, H., S. Szedmak, A. Rodr'iguez-S'anchez, and J. Piater (2014). "Towards sparsity and selectivity: Bayesian learning of restricted Boltzmann machine for early visual features." In: *In Artificial Neural Networks and Machine Learning ICANN 2014, Springer International Publishing*, pp. 419–426 (cit. on p. 66).

Xu, S., W. Jiang, M. Poo, and Y. Dan (2012). "Activity recall in a visual cortical ensemble." In: *Nature Neuroscience* (cit. on p. 42).

Xu, T., X. Yu, A. J. Perlik, W. F. Tobin, J. A. Zweig, K. Tennant, T. Jones, and Y. Zuo (2009). "Rapid formation and selective stabilization of synapses for enduring motor memories." In: *Nature* 462.7275, pp. 915–919 (cit. on p. 87).

Xu, W., X. Huang, K. Takagaki, and J. Wu (2007). "Compression and reflection of visually evoked cortical waves." In: *Neuron* 55.1, pp. 119–129 (cit. on pp. 12, 28, 39).

Yagishita, S., A. Hayashi-Takagi, G. C. Ellis-Davies, H. Urakubo, S. Ishii, and H. Kasai (2014). "A critical time window for dopamine actions on the structural plasticity of dendritic spines." In: *Science* 345.6204, pp. 1616–1620 (cit. on pp. 77, 79, 80, 91, 135, 141, 145, 147).

Yamahachi, H., S. A. Marik, J. N. J. McManus, W. Denk, and C. D. Gilbert (2009). "Rapid axonal sprouting and pruning accompany functional reorganization in primary visual cortex." In: *Neuron* 64.5, pp. 719–729 (cit. on pp. 44, 64, 92).

Yang, G., F. Pan, and W.-B. Gan (2009). "Stably maintained dendritic spines are associated with lifelong memories." In: *Nature* 462.7275, pp. 920–924 (cit. on pp. 57–59, 61, 66, 70).

Yarom, Y. and J. Hounsgaard (2011). "Voltage fluctuations in neurons: signal or noise?" In: *Physiological Reviews* 91.3, pp. 917–929 (cit. on p. 1).

Yasumatsu, N., M. Matsuzaki, T. Miyazaki, J. Noguchi, and H. Kasai (2008). "Principles of long-term dynamics of dendritic spines." In: *The Journal of Neuroscience* 28.50, pp. 13592–13608 (cit. on pp. 44, 50, 64, 70, 73, 92).

Yu, Z., D. Kappel, R. Legenstein, S. Song, F. Chen, and W. Maass (2016). "CaMKII activation supports reward-based neural network optimization through Hamiltonian sampling." In: *arXiv preprint arXiv:1606.00157* (cit. on p. 6).

Zemel, R. S. and P. Dayan (1997). "Combining probabilistic population codes." In: *IJCAI*, pp. 1114–1119 (cit. on p. 6).

Zemel, R. S., P. Dayan, and A. Pouget (1998). "Probabilistic interpretation of population codes." In: *Neural Computation* 10.2, pp. 403–430 (cit. on p. 6).

Ziv, N. E. and E. Ahissar (2009). "Neuroscience: New tricks and old spines." In: *Nature* 462.7275, pp. 859–861 (cit. on p. 70).

Ziv, Y., L. D. Burns, E. D. Cocker, E. O. Hamel, K. K. Ghosh, L. Kitch, A. E. Gama, and M. J. Schnitzer (2013). "Long-term dynamics of CA1 hippocampal place codes." In: *Nature Neuroscience* 16.3, pp. 264–266 (cit. on pp. 63–65, 70, 92, 131).

Zuo, Y., A. Lin, P. Chang, and W.-B. Gan (2005). "Development of long-term dendritic spine stability in diverse regions of cerebral cortex." In: *Neuron* 46.2, pp. 181–189 (cit. on pp. 54, 57).