Bayesian inference in neural circuits and synapses

Aitchison, L.D.

Bachelor's, Physics, University of Cambridge, UK (2010)
Masters, Systems Biology, University of Cambridge, UK (2011)

Gatsby Computational Neuroscience Unit
University College London
Sainsbury Wellcome Centre
25 Howland Street
London
W1T 4JG

THESIS

Submitted for the degree of
Doctor of Philosophy, University of London

2017

I, Laurence Aitchison, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Bayesian inference describes how to reason optimally under uncertainty. As the brain faces considerable uncertainty, it may be possible to understand aspects of neural computation using Bayesian inference. In this thesis, I address several questions within this broad theme. First, I show that confidence reports may, in some circumstances be Bayes optimal, by taking a "doubly Bayesian" strategy: computing the Bayesian model evidence for several different models of participant's behaviour, one of which is itself Bayesian. Second, I address a related question concerning features of the probability distributions realised by neural activity. In particular, it has been show that neural activity obeys Zipf's law, as do many other statistical distributions. We show the emergence of Zipf's law is in fact unsurprising, as it emerges from the existence of an underlying latent variable: firing rate. Third, I show that synaptic plasticity can be formulated as a Bayesian inference problem, and I give neural evidence in support of this proposition, based on the hypothesis that neurons sample from the resulting posterior distributions. Fourth, I consider how oscillatory excitatory-inhibitory circuits might perform inference by relating these circuits to a highly effective method for probabilistic inference: Hamiltonian Monte Carlo.

Acknowledgments

First and foremost, I would like to thank my supervisor, Peter Latham. He has taught me many valuable lessons about how to do research, how to write, and (particularly) how to present. Furthermore, he has admirably performed that utterly essential part of being a supervisor: driving excellence.

Second, I would like to thank all those people who make (and have made) the Gatsby Computational Neuroscience Unit such a fantastic place — and this goes to all the members of the unit, past and present, with whom I have the privilege of working. Of course, creating such an environment requires careful tending into being. As such, I would like to give particular thanks to Peter Dayan, for putting in the care and attention needed to create and maintain the Gatsby's superb intellectual environment. Finally, I would like to thank Reign and Barry, for minimizing the administrative work passed on to students.

Third, I would like to thank my external collaborators, from Mate Lengyel (who supervised the last chapter of this thesis) and Guillaume Hennequin in Cambridge, to Bahador Bahrami and Dan Bang in psychology and to Adam Packer, Lloyd Russell, and Christoph Schmidt-Hieber, and Michael Hausser. While much of the resulting work has not made it into this thesis, it has made my time at Gatsby far more varied and interesting.

Finally, I would like to thank my partner, Rose Hughes, for support and encouragement.

Contents

Front matter

Abstract . . . 3
Acknowledgments . . . 4
Contents . . . 5
List of figures . . . 7
Introduction . . . 9
Literature Review . . . 12

1 Doubly Bayesian analysis of confidence **22**
1.1 Abstract . . . 22
1.2 Introduction . . . 22
1.3 Methods . . . 27
1.4 Results . . . 36
1.5 Discussion . . . 43
1.6 Supplementary Figures . . . 47

2 Zipf's law in neural data **51**
2.1 Abstract . . . 51
2.2 Introduction . . . 51
2.3 Results . . . 53
2.4 Discussion . . . 72
2.5 Methods . . . 75

3 Probabilistic Synapses **92**
3.1 Abstract . . . 92
3.2 Introduction . . . 92
3.3 Results . . . 94
3.4 Discussion . . . 105
3.5 Methods . . . 109
3.6 Supplementary Information . . . 120

4 The Hamiltonian brain **134**
4.1 Abstract . . . 134
4.2 Introduction . . . 135

4.3 Results 137
4.4 Discussion 151
4.5 Methods 152
4.6 Supplementary Figure 161
Rear matter
Summary and future work 162
References 164

List of figures

1.1 Any monotonic transformation of one-dimensional sensory data can give the same mapping from sensory data to confidence. 25
1.2 Schematic of experimental design and task. 28
1.3 Schematic diagram of our method for mapping thresholds to confidence probabilities. 34
1.4 The probability of the three models given the data. 36
1.5 Single-subject analysis. 37
1.6 Simulated (Bayesian model) and actual confidence distributions for one subject (one response), and each target interval and contrast. . 39
1.7 Simulated (Bayesian model) and actual psychometric curves for two subjects. 40
1.8 Different models lead to different distributions over confidence. . . 41
1.9 The mapping from stimulus-space to confidence induced by different models. 42
1.10 The empirical and fitted distributions over signed confidence given the signed contrast for the one-response dataset. 49
1.11 As Figure 1.10, but for the two-responses dataset. 50

2.1 PEEV measures the average width of $P(\mathcal{E}|z)$ relative to $P(\mathcal{E})$. . . 58
2.2 Zipf's law for word frequencies, split by part of speech. 60
2.3 Li's model of random words displays Zipf's law because it mixes words of different lengths. 63
2.4 Re-analysis of amino acid sequences in the D region of 14 Zebrafish. 63
2.5 The relationship between $P(\mathcal{E})$ and Zipf plots. 67
2.6 Neural data recorded from 30 mouse retinal ganglion cells stimulated by full-field illumination. 69

3.1 Comparison of the delta rule and the optimal learning rule. 96
3.2 Updating the distribution over weights using Bayes theorem. . . . 97
3.3 Bayesian learning rules track the true weight and estimate uncertainty. 99
3.4 Bayesian learning rules have a lower mean squared error (MSE) than classical learning rules. 100

3.5 Simulations confirming that the normalized learning rate is inversely related to the square root of the firing rate. 102
3.6 Simulations confirming that that the normalized learning rate is proportional to the normalized variability. 104
3.7 Normalized variability (the ratio of the PSP variance to the mean) as a diagnostic of our theory. 105
3.8 A graph describing the dependencies in our simulations. 116
3.9 A schematic diagram of a stick-person jumping over a puddle. The probability of landing in the puddle, $P(\text{wet})$, depends not only on the mean estimate, but also on the uncertainty. 117
3.10 The mean squared error relative to the Bayesian learning rules for classical reinforcement learning rules with different settings of α_{reward} and with different settings for the learning rate, α. 121

4.1 An example of Hamiltonian dynamics. 138
4.2 Graphical model and basis functions for te Gaussian scale mixture model. 139
4.3 The architecture of the Hamiltonian network. 142
4.4 The Hamiltonian sampler is more efficient than a Langevin sampler. 144
4.5 Excitation and inhibition are balanced in the Hamiltonian network. 146
4.6 Oscillation frequency depends on stimulus contrast. 147
4.7 Large, contrast-dependent firing rate transients in the model. . . . 149
4.8 Our main results are robust to a range of ρ or equivalently τ_L. . . 161

Introduction

Every second, the brain receives a huge volume of complex, structured sensory information (Jacobson, 1951). The brain must disentangle this information — working out what opportunities and threats are in the surrounding environment and choosing actions accordingly. For instance, is the complex pattern of light on the retina a bush or a tiger, and thus, should you run away?

Marr and Poggio (1976) proposed three levels at which we can understand neural systems. At the computational level, we ask about the system's overall goal. For instance, one possible computation is probabilistic inference. At the algorithmic level, we ask what sequence of steps the brain might use to achieve its goal. For instance, one algorithm for performing inference is Metropolis-Hastings Monte Carlo (MHMC) sampling (Robert and Casella, 2011). Finally, at the implementation level, we ask how the algorithm is implemented in real biological hardware.

We focus on one computation: probabilistic inference, and one class of inference algorithms: sampling algorithms.

In general, probabilistic inference takes a model that describes how sensory data might be generated from latent causes, and then inverts the model, to find the latent causes that might have given rise to the observed sensory data (Bayes and Price, 1763). Probabilistic inference is interesting from a neuroscientific perspective because it is theoretically well-motivated, and practically effective. Theoretically, axiomatic formulations of Bayes theorem have shown that any calculus of uncertainty obeying certain, intuitively obvious, axioms, gives rise to Bayesian probability theory (Knuth and Skilling, 2012). Practically, while there are many approaches to solving simpler machine-learning tasks (for instance, there are a huge number of methods for solving supervised learning problems, one example being random forests (Breiman, 2001)), once the models reach a high-enough level of complexity (e.g. unsupervised heirarchial or non-parametric models), the only viable approach is Bayesian machine-learning (Feller and Gelman, 2014).

Performing probabilistic inference requires the use of a probabilistic inference algorithm. There are many such algorithms, each with advantages and disadvan-

tages (Frey and Jojic, 2005), and it remains unclear which one is used by the brain (Pouget et al., 2013). We focus on one particular family of algorithms, sampling algorithms, which state that neural activity at one particular instant represents one plausible explanation of the sensory data, and as neural activity changes over time, the brain is effectively exploring multiple plausible explanations for the data. Formally, a snapshot of neural activity represents a sample from the posterior distribution over latent variables given observed data. The idea that the brain samples, known as the sampling hypothesis (Hoyer and Hyvarinen, 2003; Fiser et al., 2010), has biological and compuational advantages over other competing ideas. Biologically, sampling multiple plausible explanations of the sensory data provides a natural explanation for variability in neural responses (Hoyer and Hyvarinen, 2003). Furthermore, two observed features of neural variability match predictions given by the sampling hypothesis: variability is supressed by stimulus onset (Churchland et al., 2010; Orbán et al., 2013), and spontaneous activity changes over an animal's lifetime in order to match evoked activity (Berkes et al., 2011b). Computationally, sampling algorithms can be applied to almost every probabilistic model, and give the right answer if given enough computation time (Robert and Casella, 2011). In contrast, approximate inference techniques have, by necessity, approximation biases, which cannot be eliminated by the addition of more computation time (Ghahramani et al., 2000). Furthermore, many approximation techniques can be applied only to relatively narrow classes of model.

Here, I present four projects related to these overarching themes. First, we consider whether confidence reports appear to result from underlying Bayesian-optimal neural computations, or from another heuristic process. Using a "doubly-Bayesian" procedure, in which we perform Bayesian model selection over a set of models of neural processing, one of which is itself Bayesian, we show that in some circumstances decisions appear Bayes optimal, whereas in other circumstances they do not. Second, it has been shown that neural activity displays an interesting statistical property known as Zipf's law: if we take each possible pattern of neural activity, and rank them from most to least frequent, then the frequency is inversely proportional to the rank. This pattern is observed in many other domains, including e.g. antibodies and word frequencies. We show that in neural activity, and in some of these domains, Zipf's law emerges because of an underlying latent variable (firing rate in the case of neural activity). Third, I consider synaptic plasticity as a Bayesian inference problem, and show that this approach makes a variety of predictions about how learning rates scale with presynaptic firing rates. Furthermore, if we assume that synapses sample EPSPs from their posterior distributions over firing rates, we make a further prediction about how EPSP variability should change with presynaptic firing rates; I confirm this prediction with a novel reanalysis of data from Ko et al. (2011).

Finally, I consider how sampling-based probabilistic inference might be implemented in oscillatory excitatory-inhibitory neural circuits by using Hamiltonian Monte Carlo.

The first (Aitchison et al., 2015), second (Aitchison et al., 2016) and fourth (Aitchison and Lengyel, 2016) chapters have been published. The third chapter has been reviewed by Nature, and we are preparing an appeal.

Literature Review

Here we survey computational and algorithmic models of neural circuits and synapses. In order to better understand why I focus on one computation (probabilistic inference) and one algorithm (sampling), here we consider these approaches in the context of other possible approaches. In particular, before moving on to the computation that we focus on, probabilistic inference, we consider two other computational approaches: information maximization and deep-belief networks. Likewise, before moving on to the family of algorithms that we focus on, sampling, we consider two other families of algorithms for performing inference, maximum a-posteriori algorithms, and approximate inference algorithms.

Computation: Mutual Information Maximization

Shortly after the birth of information theory (Shannon, 1948), there were proposals that sensory systems maximize the mutual information between sensory stimulus, s, and neural activity, r (Attneave, 1954; Barlow, 1961). The mutual information is defined to be the difference of two entropies (Shannon, 1948),

$$\mathrm{I}\left(r;s\right) = \mathrm{H}\left[r\right] - \mathrm{H}\left[r|s\right], \tag{1}$$

where the entropy of the neural response is,

$$\mathrm{H}\left[r\right] = -\sum_{r} P\left(r\right) \log P\left(r\right), \tag{2}$$

and the entropy of the neural response conditioned on the sensory data is,

$$\mathrm{H}\left[r|s\right] = -\sum_{s} P\left(s\right) \sum_{r} P\left(r|s\right) \log P\left(r|s\right). \tag{3}$$

Entropies are always non-negative (Cover and Thomas, 1991), with a large entropy indicating a very random distribution (e.g. a uniform distribution), and a small entropy indicating a less random, or more deterministic distribution (e.g. a very high probability for only one option). Thus, we can see there are two factors that contribute to a large mutual information. First, the transformation from

sensory data to neural activity should be as deterministic as possible, implying that $\mathrm{H}[r|s] \approx 0$. Second, the range of possible responses, r, should be as broad as possible, giving a large $\mathrm{H}[r]$. In combination, we can see that if the response is to have a large mutual information, it must deterministically map different sensory data items to as wide a range of neural responses as possible.

It was initially argued that a maximally informative representation of the sensory data might somehow disentangle the latent causes describing the state of the world (Attneave, 1954; Barlow, 1961). However, looking again at the definition of the mutual information, we see that mutual information maximization will not necesserily disentangle latent causes (Dayan and Abbott, 2001). The reasoning is simple: mutual information maximization finds a deterministic transform that uses the full range of possible neural responses as effectively as possible. This does not require that the resulting deterministic function will somehow disentangle the latent causes. That said, in some special cases, mutual information maximization does coincide with probabilistic inference, for instance, there is a link between infomax based independent components analysis (ICA) (Bell and Sejnowski, 1995), and maximum-likelihood based ICA (Pearlmutter and Parra, 1996; MacKay, 1996).

However, it does appear that mutual information maximization is important whenever there are informational bottlenecks — when there is a large amount of information that needs to travel down a constrained channel. One of the first examples of this principle was given by Laughlin (1981). He showed that the neural response in Large monopolar cells (LMCs) in the insect compound eye has a uniform distribution, maximizing the first term in Equation (1), and thus maximizing the mutual information between the sensory data and the response. Mutual information maximization is also used to explain properties of retina ganglion cells (RGCs), because these cells form an important bottleneck — all the visual information received by the retina must be communicated by 10^6 retinal ganglion cells (RGCs) (Bruesch and Arey, 1942). Mutual information maximization can therefore be used to understand many properties of RGCs, including receptive fields with an inhibitory surround that becomes facilitatory at low contrast (Atick and Redlich, 1990), the proportion and opponency of colour photoreceptors (Atick et al., 1992; Garrigan et al., 2010), properties of tiling in the retina (Borghuis et al., 2008) and the proportion of on vs off cells (Ratliff et al., 2010; Karklin and Simoncelli, 2011). Furthermore, information theoretical approaches have been applied to single synapses. Toyoizumi et al. (2004) derived an STDP rule, based on the notion that the neuron maximizes the mutual information between the postsynaptic spike train and the presynaptic spike trains. Goldman (2004) showed that vesicle release failures can, under some circumstances, lead to an increase in the information transferred across the synapse per vesicle. However, the relevance of these ideas is unclear — while one can view

the neuron as an information bottleneck, its primary function is not to transmit as much information as possible, but to compute a useful function of its input, and it remains unclear whether these notions necessarily overlap.

Computation: Deep neural networks

Recently, deep neural networks have been found to offer state of the art performance on a wide range of tasks, from handwritten digit recognition (e.g. Ciresan et al. (2010)), object recognition (e.g. Krizhevsky et al. (2012)), and speech recognition (e.g. Graves et al. (2013)), to more specialised tasks, like translation (e.g. Sutskever et al. (2014)), and finding the boundary between neurons in electron microscopy images (e.g. Turaga et al. (2010)). Such networks might therefore be expected to offer us insights into the computations performed by the brain, and indeed, the representation in each layer of a neural network has been shown to mirror, to some extent, the representation in each stage of neural visual processing (Yamins et al., 2014).

However, there are three arguments that suggest that deep neural networks, at least in their current form, will not provide fundamental insight into the computations performed by the brain. First, in order to train a deep neural network, it is necessary to have a huge amount of labelled training data (Deng, 2014; Schmidhuber, 2015). In contrast, the brain does not receive such a vast volume of labelled data. Instead, the brain must extract the statistical structure in unlabelled data — a process known as unsupervised learning (Hinton, 2007), perhaps aided by a small number of labelled examples (Hinton et al., 2006). Second, the procedure by which all such networks are trained involves backpropagating information about errors in the network's output through the whole network. While this is straightforward computationally (Rumelhart et al., 1986), it is very difficult in neural hardware, which allows signals to travel along the axon in only one direction (though there may be ways to get around this issue (Lillicrap et al., 2014; Bengio et al., 2015)). Third, these networks are trained in a supervised fashion — meaning that they are designed only to extract a relatively small amount of information, usually just object identity. While they may, incidentally, extract more information, it is by no means certain that such networks will extract, for instance, part-whole relationships (Hinton et al., 2011; Szegedy et al., 2013). Despite these difficulties, deep-neural-networks do constitute a useful proof-of-existence — they demonstrate that many difficult tasks can, in fact, be performed by networks that are similar, at least at a very shallow level, to those in the brain.

Probabilistic Inference

Probabilistic inference based approaches begin by specifying a probabilistic generative model. Probabilistic generative models explicitly describe the relationship between latent causes and observations (Pearl, 1988; Koller and Friedman, 2009), by specifying how to generate a possible observation from a particular set of latent causes. Probabilistic inference then goes backwards, inferring the latent causes that might have given rise to that particular observation. Importantly, in order to work out plausible latent causes, it is important to have not only a likelihood, describing how link between observations and latents, but also a prior, describing the probability of each latent cause (e.g. if I hear a fire alarm, it is probably because there is a drill or test, not because the building is on fire) (Adams et al., 2004; Mulder et al., 2012).

Probabilistic generative models have multiple advantages over approaches described previously. First, unlike information maximization, it should be able to disentangle the latent causes underlying the stimulus, because that is its explicit goal (Stoianov and Zorzi, 2012). Second, unlike deep belief networks, the goal is to model every latent variable that is necessary to explain the image — so all relevant information (including, for instance part-whole relationships) should be present in the inferred latent variables, unlike in a deep belief network, where the only goal is to match the labels in the training set.

Moreover, there are a significant number of studies suggesting that people's behaviour is Bayes optimal, reviewed in (Pouget et al., 2013). For instance, integration of information from different sensory modalities is believed to be Bayes optimal (Alais et al., 2010). In particular, humans optimally combine visual and haptic or proprioceptive information in order to determine the size of objects (Ernst and Banks, 2002), or to determine hand location (van Beers et al., 1999b). Similarly, animals can optimally combine visual and auditory information (Raposo et al., 2012), and humans and animals can optimally integrate information across time (Brunton et al., 2013). A separate branch of work has provided evidence that humans use Bayes-optimal models in motor control (reviewed in (Körding and Wolpert, 2006; Wolpert, 2007; Orbán and Wolpert, 2011; Wolpert and Landy, 2012)). For instance, Wolpert et al. (1995) showed that people optimally integrate information from proprioception and motor commands in order to estimate hand position after movements in the dark.

However, probabilistic generative models raise almost as many questions as they answer. In particular, there are many families of algorithms that perform inference — that extract plausible settings for the latent variables from an observation — each with advantages and disadvantages for use in neural computation. The most important distinction between probabilistic inference algorithms is whether

they find only a single best explanation for the sensory data, or find the full range of plausible explanations, represented as a full probability distribution. Thus, we begin by considering these two families of algorithm.

Formally, the best explanation for the sensory data is given by maximum a-posteriori (MAP) inference,

$$\text{latents}^* = \text{argmax}_{\text{latents}} \left[P(\text{observations}|\text{latents})\, P(\text{latents}) \right]. \tag{4}$$

MAP approaches are extremely simple and thus are relatively easy to map on to neural hardware (e.g. Olshausen and Field (1996)). Two features contribute to their simplicity. First, the representation is simple. In particular, the system only needs to represent a single setting of the latent variables, rather than a full distribution over plausible settings. Second, the inference algorithm is simple. In particular, the system simply needs to perform optimization (e.g. gradient ascent (Cauchy, 1847)) to find the setting of the latents that maximizes the objective, $P(\text{latents}, \text{observations})$, with the observations held fixed.

MAP inference has been used in two complementary approaches to neuroscientific research.

The first approach is to write down a probabilistic model of natural stimuli, perform MAP inference and learning, then compare the response of latent variables in the model to real neurons. This is exemplified by the classic study by Olshausen and Field (1996), in which they demonstrate that MAP inference and learning gives receptive fields very similar to those observed in real neurons. Using more realistic priors (or equivalently, loss functions) leads to more realistic receptive fields (Rehn and Sommer, 2007). A similar approach can also successfully explain properties of auditory receptive fields (Lewicki, 2002), and to understand some visual extra-classical receptive field effects (Rao and Ballard, 1999).

The second approach is to write down a biologically plausible neural network that implements some form of MAP inference. These include Rozell et al. (2008), who focused on a temporal version of the sparse coding problem, Deneve and colleagues, who developed a family of neural networks that use an optimal spike-based representation of continuous quantities (Boerlin and Denève, 2011; Bourdoukan et al., 2012; Barrett et al., 2013; Boerlin et al., 2013), and Hu et al. (2012) who developed methods that are fundamentally spike-based (as opposed to other methods that typically use spikes to approximately represent continuous quantities).

However, the MAP approach has a critical problem: experimentally, it is found that humans make effective use of information about uncertainty, information that is thrown away by MAP (Ernst and Banks, 2002; van Beers et al., 1999b; Raposo et al., 2012; Brunton et al., 2013; Wolpert et al., 1995). Nevertheless,

these approaches have huge value, not only as a testbed for methods that are later extended to be fully probabilistic, but also because we cannot, at present, exclude the possibility that at least some sensory areas use MAP-like inference.

This failure leads us to think about approaches that try to characterise the full range of plausible explanations for the observed data. In particular, this range is represented as a probability distribution, computed using Bayes theorem,

$$P(\text{latents}|\text{observations}) \propto P(\text{latents})\, P(\text{observations}|\text{latents}) \tag{5}$$

This approach has the critical advantage that all information about uncertainty is retained, allowing for more effective task performance. However, this approach yet again raises more questions, because there are again multiple approaches that neural systems could take for performing inference and representing the resulting inferred distributions. Here we focus on two such approaches, parametric approximate distributions, and sampling.

Neural activity could encode the parameters of a distribution that approximates the inferred posterior. These approximate distributions are usually part of the exponential family,

$$Q(\text{latents}) \propto \exp\left(\mathbf{T}(\text{latents}) \cdot \boldsymbol{\theta}\right), \tag{6}$$

where $\boldsymbol{\theta}$ is the natural parameter vector, $\mathbf{T}(\text{latents})$ is the sufficient statistic vector. Importantly, an exponential family distribution can be parameterised by its natural parameters, $\boldsymbol{\theta}$, or its mean parameters, $\boldsymbol{\mu} = \mathrm{E}\left[\mathbf{T}(\text{latents})\right]$. There is a one-to-one mapping between these parameters, so the representations are equivalent — though the inference strategies afforded by each representation are very different.

First, we consider approaches based on natural parameters, which are often known in neuroscience as probabilistic population codes (PPCs) (Ma et al., 2006; Beck et al., 2007, 2008). PPCs were regarded as interesting because they facilitate easy combination of information from multiple sources. For instance, if we have visual and haptic information, represented as,

$$P(\text{visual observations}|\text{latents}) \propto \exp\left(\mathbf{T}(\text{latents}) \cdot \boldsymbol{\theta}_{\text{visual}}\right) \tag{7}$$

$$P(\text{haptic observations}|\text{latents}) \propto \exp\left(\mathbf{T}(\text{latents}) \cdot \boldsymbol{\theta}_{\text{haptic}}\right) \tag{8}$$

and prior information represented in a similar form,

$$P(\text{latents}) \propto \exp\left(\mathbf{T}(\text{latents}) \cdot \boldsymbol{\theta}_{\text{prior}}\right) \tag{9}$$

then the posterior distribution can be represented by the sum of the natural

parameters,

$$P(\text{latents}|\text{observations}) \propto \exp\left(\mathbf{T}(x) \cdot (\boldsymbol{\theta}_{\text{prior}} + \boldsymbol{\theta}_{\text{visual}} + \boldsymbol{\theta}_{\text{haptic}})\right). \tag{10}$$

Thus, if the natural parameters are represented as firing rates, then it would be very easy to combine these distributions — the brain would simply need to add together the firing rates. However, PPC-like variational inference schemes are difficult develop for more complex and interesting models of realistic sensory stimuli (though see Beck et al. (2012)).

Second, mean parameter based approaches state, simply, that a posterior distribution is represented by the expectation, under that posterior, of a variety of functions (often these functions are the sufficient statistics vector, $\mathbf{T}(x)$). For instance, Zemel et al. (1998) proposed that neural firing rates, r_i, represent the posterior distribution over a single variable, s, by taking the expectation under the posterior of a set of basis functions,

$$r_i = \int P(s|\text{observations})\, \phi_i(s) ds. \tag{11}$$

This proposal might seem to preclude representation of uncertainty about multiple stimuli, (if, for instance, you take s to be a single scalar, describing the orientation of a stimulus). In fact, this is not true, as s can be an arbitrary object (e.g. a vector describing the orientation of multiple stimuli). It is then possible to find the posterior distribution over this function, and to report expectations under that posterior (Sahani and Dayan, 2003). Oddly, these schemes are not, at present an active subject of research in neuroscience. However, recent results in Machine learning showing that it is possible to infer expectations under a posterior distribution using only regression-like techniques (Fukumizu et al., 2013) might rejuvenate the area.

An alternative approach is to use a large number of samples from the posterior distribution $P(\text{latents}|\text{observations})$ to represent that distribution. Neuroscientifically, this implies that neural activity represents a particular setting for the latent variables, and variability over time in neural activity samples multiple plausible settings for the latent variables.

Sampling has a long history in computational neuroscience, with one particularly important early advance being the Boltzmann Machine (BM) (Ackley et al., 1985). The BM specifies a distribution over the binary vector $\mathbf{x}$, where $x_i = 1$ indicates that cell i is active, whereas $x_i = 0$ indicates that the cell is inactive. Formally, the Boltzmann Machine probability distribution is given by,

$$P(\mathbf{x}) \propto e^{\mathbf{x}^T \mathbf{W} \mathbf{x}}. \tag{12}$$

Inference and learning in the BM is relatively straightforward. To perform inference, the typical method is Gibbs sampling — repeatedly sampling one element of the vector, x_i conditioned on all the other elements, $x_{/i}$ (some of which may be observed, and hence fixed). Importantly, the exact expression for this update is remarkably similar to the update equation for stochastic binary neurons,

$$P\left(x_i = 1 | \mathbf{x}_{/i}\right) = \sigma\left(\sum_j W_{ij} x_j\right), \tag{13}$$

where σ is the sigmoid function,

$$\sigma\left(E\right) = \frac{1}{1 + e^{-E}}. \tag{14}$$

To learn a Boltzmann Machine, the typical method is the Hebbian-like Boltzmann Machine learning rule,

$$\frac{\partial \log P(\mathbf{x})}{\partial \mathbf{W}} = \left\langle \mathbf{x}\mathbf{x}^T \right\rangle_{\text{Data}} - \left\langle \mathbf{x}\mathbf{x}^T \right\rangle_{\text{Model}}, \tag{15}$$

where the first expectation is computed based on the data (Gibbs sampling any unobserved units) and the second expectation is computed based only on the model (Gibbs sampling all units).

The Boltzmann Machine is very interesting because it is able to perform sampling-based inference and learning despite its extreme simplicity. However, there is one way in which the Boltzmann Machine looks very different from a true neural circuit — the Gibbs updates are serial, only one cell is updated at a time, whereas in a typical neural circuit, you update every cell in parallel at each time step. There are two methods by which this problem can be alleviated.

The first method is to divide the units into observed units, $\mathbf{v}$, and latent, or hidden units, $\mathbf{h}$, then to ensure that there are no recurrent connections from observed to observed units, or from hidden to hidden units, so the only connections run from hidden to observed units. This restricted model is known as the restricted Boltzmann Machine (RBM) and takes the form,

$$P(\mathbf{v}, \mathbf{h}) \propto e^{\mathbf{b}_v^T \mathbf{v} + \mathbf{b}_h^T \mathbf{h} + \mathbf{h}^T \mathbf{W} \mathbf{v}}. \tag{16}$$

Importantly, this restriction makes it possible to sample all the hidden units conditioned on the visible units (or all the visible units conditioned on the hiddens),

in parallel in one step,

$$P(\mathbf{h}|\mathbf{v}) = \prod_i P(h_i|\mathbf{v}) = \prod_i \sigma\left(b_{h,i} + \sum_j W_{ij} v_j\right), \tag{17}$$

$$P(\mathbf{v}|\mathbf{h}) = \prod_j P(v_j|\mathbf{h}) = \prod_i \sigma\left(b_{v,j} + \sum_i h_i W_{ij}\right). \tag{18}$$

An alternative approach is to notice that it is, in fact, possible to sample a Boltzmann Machine in continuous time, with neuron-like units. In particular, (Buesing et al., 2011) showed that units that turn on with some probability, then turn off after a fixed length of time (representing the membrane time constants of the downstream cells), can sample from a fully-connected BM.

The Boltzmann Machine and its variants achieved considerable success in the machine learning community, but at present it has fallen out of favour, being replaced by deep neural networks based on backprop.

The next suggestion for a sampling scheme was the Helmholtz Machine (Dayan et al., 1995; Dayan and Hinton, 1996; Dayan, 1998). The Helmholtz Machine introduced the concept that there might be two models: a generative model, representing the data-generating process, and a separate recognition model, with different parameters, which is used to infer the latent states associated with incoming data. Formally, there are again latent, or hidden units, $\mathbf{h}$ and observed, or visible, units $\mathbf{v}$. The generative model is given by $P_\theta(\mathbf{v}|\mathbf{h})\,P_\theta(\mathbf{v})$, and the recognition model is given by a separate distribution, $Q_\phi(\mathbf{h}|\mathbf{v})$. The advantage of this approach is that inference is very fast — instead of having to invert the generative model, which can be very slow, as it almost always involves an iterative procedure, the data is simply pushed through the (usually feedforward) recognition model. To train the recognition and generative models, and to ensure that they are consistent, the Helmholtz Machine uses a remarkably simple scheme. To train the recognition model, we generate data from the generative model, $P_\theta(\mathbf{v}, \mathbf{h})$. We then have both $\mathbf{v}$ and $\mathbf{h}$, so training the parameters of the recognition model, ϕ, by, for instance gradient ascent, is straightforward. To train the generative model, we take data, $\mathbf{v}$, then generate $\mathbf{h}$ using the recognition model. As before, we now know both $\mathbf{v}$ and $\mathbf{h}$, so training the parameters of the generative model, θ is straightforward. This learning scheme forces the generative and recognition models to be consistent with each other, and with the observed data. Sadly, models learned by the Helmholtz Machine turn out to be relatively poor, perhaps because the training procedure lacks a single objective function. However, the fundamental innovative idea in the Helmholtz Machine — the recognition model — is now being used to great success in variational autoencoders (Mnih and Gregor, 2014).

In recent years, researchers have sought direct experimental tests of the sampling hypothesis. Thus far, this approach has yielded two interesting results. First, the sampling hypothesis predicts that variability in neural activity is suppressed upon stimulus onset (Orbán and Lengyel, 2011), as, indeed, is observed (Churchland et al., 2010) (though other models also predict this, see, for instance Deco and Hugues (2012)) The sampling hypothesis makes this prediction because, with no stimulus the animal is uncertain about the "true" state of the external world, so many different states are plausible, leading to high variability. In contrast, with a stimulus, the animal is relatively certain about the "true" state of the external world, so only a small range of states are plausible, leading to low variability. Second, the sampling hypothesis makes a strong prediction about the relationship between spontaneous activity (i.e. neural activity with no stimulus), and average evoked activity (i.e. neural activity with a stimulus, averaged over all possible stimuli). In particular, we take spontaneous activity to represent the animal's prior, and evoked activity to represent the animal's posterior inferences. If the animal has learned a good model of the visual world, then the prior should represent the baseline probability of each setting of the latent variables. Importantly, this prior is learned by observing data, inferring the latent variables, then working out how often each setting of the latent variables occurs. Thus, spontaneous activity (representing the prior) should match average evoked activity (representing average posterior inferences over many different stimuli). Moreover, this match should improve as the animal develops, and hence learns a better model. This was, indeed, observed (Berkes et al., 2011b).

This literature review has described a collection of related approaches to understanding neural computation. While each of these methods has arguments for and against their usefulness, there is, at present, too little data to enable us to conclusively rule in or out any particular approach. As such, perhaps the only way to make progress is to pick one approach, perhaps one that has more arguments in its favour, and make theoretical advances enabling that approach to be tested experimentally — an approach that I have taken here for the sampling hypothesis.

Chapter 1

Doubly Bayesian analysis of confidence

1.1 Abstract

Humans stand out from other animals in that they are able to explicitly report on the reliability of their internal operations. This ability, which is known as metacognition, is typically studied by asking people to report their confidence in the correctness of some decision. However, the computations underlying confidence reports remain unclear. In this paper, we present a fully Bayesian method for directly comparing models of confidence. Using a visual two-interval forced-choice task, we tested whether confidence reports reflect heuristic computations (e.g. the magnitude of sensory data) or Bayes optimal ones (i.e. how likely a decision is to be correct given the sensory data). In a standard design in which subjects are first asked to make a decision, and only then give their confidence, subjects were mostly Bayes optimal. In contrast, in a less-commonly used design in which subjects indicated their confidence and decision simultaneously, they were roughly equally likely to use the Bayes optimal strategy or to use a heuristic but suboptimal strategy. Our results suggest that, while people's confidence reports can reflect Bayes optimal computations, even a small unusual twist or additional element of complexity can prevent optimality.

1.2 Introduction

Humans and other animals use estimates about the reliability of their sensory data to guide behaviour (e.g. (Kepecs et al., 2008; Kiani and Shadlen, 2009; Komura et al., 2013)). For instance, a monkey will wait until their sensory data is deemed sufficiently reliable before taking a risky decision (Komura et al., 2013).

Humans can go further than other animals: they can explicitly communicate estimates of the reliability of their sensory data, by saying, for instance, "I'm sure" — an ability that is important for effective cooperation (Bahrami et al., 2010; Fusaroli et al., 2012; Shea et al., 2014). This ability to report on the reliability of our internal operations is known as "metacognition", and is typically studied by asking people to report their confidence in the correctness of some decision (Fleming et al., 2012). However, the computations underlying confidence reports remain a matter of debate (see Box 1 in Shea et al. (2014), for a brief overview). For instance, in an orientation-discrimination task, reports might — as a heuristic — reflect the perceived tilt of a bar. Alternatively, reports might reflect more sophisticated computations, like Bayesian inference about the probability that a decision is correct. An accurate understanding of confidence reports is important given its role in high-risk domains, such as financial investment (e.g. (Broihanne et al., 2014)), medical diagnosis (e.g. (Berner and Graber, 2008)), jury verdicts (e.g. (Tenney et al., 2007)), and politics (e.g. (Johnson, 2004)).

Here, we ask: how do people compute their confidence in a decision? We are particularly interested in whether confidence reports reflect heuristic or Bayes optimal computations. The latter would be consistent with a wide array of work showing that other aspects of perception and decision making are Bayes optimal (Ma and Jazayeri, 2014). However, as far as we know, whether confidence reports reflect Bayes optimal computations has not been directly tested. We use a standard psychophysical task in which subjects receive sensory data, make a decision based on this data, and report how confident they are that their decision is correct. Our goal is to determine how subjects transform sensory data into a confidence report. In essence, we are asking: if we use $\mathbf{x}$ to denote the sensory data ($\mathbf{x}$ can be multi-dimensional) and c to denote a confidence report, what is the mapping from $\mathbf{x}$ to c? Alternatively, what is the function $c(\mathbf{x})$?

To answer this question, we follow an approach inspired by signal detection theory (Green and Swets, 1966). We hypothesize that subjects compute a continuous decision variable, $z^{\mathrm{D}}(\mathbf{x})$, and compare this variable to a single threshold to generate a decision, d. Likewise, we hypothesize that subjects compute a continuous confidence variable, $z^{\mathrm{C}}(\mathbf{x}; d)$, an internal representation of the evidence in favour of the chosen decision, d, and compare this variable to a set of thresholds to generate a level of confidence, c (the evidence in favour of one decision is different from the evidence in favour of the other decision, so the confidence variable must not only depend on the sensory evidence, $\mathbf{x}$, but also the decision, d). Within this framework, a heuristic computation is a reasonable, but ultimately somewhat arbitrary, function of the sensory data. For instance, if the task is to choose the larger of two signals, x_1 or x_2, a heuristic confidence variable might be the difference between the two signals: $z^{\mathrm{C}}_{\Delta}(\mathbf{x}; d = 2) = x_2 - x_1$ (the subscript Δ denotes difference). The Bayes optimal confidence variable, on the other hand, is

the probability that a correct decision has been made: $z^{\mathrm{C}}_{\mathrm{B}}(\mathbf{x}; d) = P(\text{correct}|\mathbf{x}, d)$ (the subscript B denotes Bayesian).

The question of whether confidence reports reflect Bayes optimal (or simply Bayesian) computations has important implications for inter-personal communication. In particular, probabilities, as generated by Bayes optimal computations, can easily be compared across different tasks (e.g. perception versus general knowledge), making them easier to map onto reports. In contrast, heuristic computations typically lead to task-dependent internal representations, with ranges and distributions that depend strongly on the task, making it difficult to map them onto reports consistently, or compare them between different people.

To our knowledge, it is impossible to determine directly the confidence variable, $z^{\mathrm{C}}(\mathbf{x}; d)$; instead, we can consider several models, and ask which is most consistent with experimental data. Choosing among different models for the confidence variable, $z^{\mathrm{C}}(\mathbf{x}; d)$, is straightforward in principle, but there are some subtleties. The most important subtlety is that if the task is "too simple", it is impossible to distinguish one model from another. Here, "too simple" means that the sensory data, $\mathbf{x}$, consists of a single signal, which we write x to indicate that it is scalar. To see why, let's say we wanted to distinguish between some heuristic confidence variable, say $z^{\mathrm{C}}_{\mathrm{H}}(x; d) = x$, and the Bayes optimal confidence variable, $z^{\mathrm{C}}_{\mathrm{B}}(x; d) = P(\text{correct}|x, d)$. Suppose we found empirically that a subject reported low confidence when the heuristic variable, $z^{\mathrm{C}}_{\mathrm{H}}(x; d)$, was less than 0.3 and high confidence when the heuristic variable was greater than 0.3. Clearly there is a deterministic mapping from the heuristic variable to the confidence reports, but is it in any way unique? The answer is no. For example, if the Bayesian variable is greater than 0.4 whenever the heuristic variable is greater than 0.3, then it is also true that our subject reported low confidence when the Bayesian variable was less than 0.4 and high confidence when the Bayesian variable was greater than 0.4. Thus, there is absolutely no way of knowing whether our subjects' confidence reports reflect the heuristic or the Bayesian confidence variable. In general, there is no way to distinguish between any two functions of x that are monotonically related — one can simply map the thresholds through the relevant function, as shown in Figure 1.1.

The situation is very different when $\mathbf{x}$ is a vector (i.e. two or more sensory signals). As in the one-dimensional case, consider two models: a heuristic model, $z^{\mathrm{C}}_{\mathrm{H}}(\mathbf{x}; d)$, and a Bayes optimal model, $z^{\mathrm{C}}_{\mathrm{B}}(\mathbf{x}; d)$. In general, if $\mathbf{x}$ is a vector, it is not possible to get the same mapping from $\mathbf{x}$ to c using $z^{\mathrm{C}}_{\mathrm{H}}(\mathbf{x}; d)$ and $z^{\mathrm{C}}_{\mathrm{B}}(\mathbf{x}; d)$. In particular, when $z^{\mathrm{C}}_{\mathrm{H}}(\mathbf{x}; d)$ and $z^{\mathrm{C}}_{\mathrm{B}}(\mathbf{x}; d)$ provide a different ordering of the $\mathbf{x}$'s — whenever we have $z^{\mathrm{C}}_{\mathrm{H}}(\mathbf{x}_1; d) > z^{\mathrm{C}}_{\mathrm{H}}(\mathbf{x}_2; d)$ and simultaneously $z^{\mathrm{C}}_{\mathrm{B}}(\mathbf{x}_1; d) < z^{\mathrm{C}}_{\mathrm{B}}(\mathbf{x}_2; d)$ — then it is not possible to find pairs of thresholds that lead to the same region in $\mathbf{x}$-space. Thus, although we cannot say much about the confidence variable for

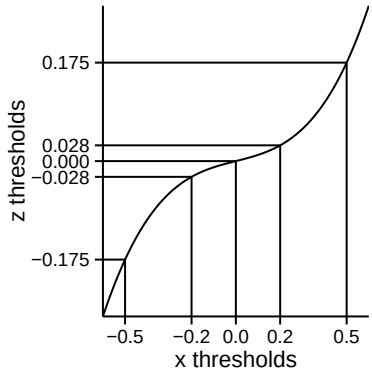

Figure 1.1: For one-dimensional sensory data, x, any monotonic transformation, $z(x)$, can give the same mapping from x to c. In particular, the best we, as experimenters, can do is to determine the mapping from x to c, which, for discrete mappings, corresponds to a set of thresholds (the vertical lines intersecting the horizontal axis). We can, however, get the same mapping from x to c by first transforming x to z (the curved black line), then thresholding z. The relevant thresholds are simply given by passing the x-thresholds through $z(x)$ (giving the horizontal lines intersecting the vertical axis). Therefore, there is no way to determine the "right" $z(x)$ — any $z(x)$ will fit the data (as long as $z(x)$ is a strictly monotonic function of x).

one-dimensional signals, we can draw strong conclusions for multi-dimensional signals.

This difference between one-dimensional and multi-dimensional sensory data is one of the key differences between our work and most prior work. Previous models based on signal detection theory have typically assumed that the sensory data is one-dimensional (e.g. (Galvin et al., 2003; Kunimoto et al., 2001; Maniscalco and Lau, 2012)), leaving them susceptible to the problem described above. There is also a variety of "dynamic" signal detection theory models in which sensory data is assumed to accumulate over time (see Pleskac & Busemeyer (2010) Pleskac and Busemeyer (2010), for an overview). Such models are able to explain the interplay between accuracy, confidence, and reaction time — something that we leave for future work. However, in these models, the sensory data is also summarised by a single scalar value, making it impossible to determine whether subjects' confidence reports reflect heuristic or Bayes optimal computations.

Here we considered multi-dimensional stimuli in a way that allows us to directly test whether subjects' confidence reports reflect heuristic or Bayes optimal computations. In our study, subjects were asked to report their confidence in a visual two-interval forced-choice task. This allowed us to model the sensory data as having two dimensions, with one dimension coming from the first interval and the other from the second interval. We considered three models for how subjects generated their confidence — all three models were different "static" versions of

the popular race model in which confidence reports are assumed to reflect the balance of evidence between two competing accumulators (originally proposed by Vickers (1979), and more recently used in studies such as Kepecs et al. (2008), and de Martino et al. (2013). The first model, the Difference model, assumed – in line with previous work – that subjects' confidence reports reflected the difference in magnitude between the sensory data from each interval. The second model, the Max model, assumed that subjects' confidence reports reflected only the magnitude of the sensory data from the interval selected on a given trial — thus implementing a "winner-take-all" dynamic (Wang, 2008). The third model, the Bayes optimal model, assumed that subjects' confidence reports reflected the probability that their decision was correct given the sensory data from each interval. Furthermore, we tested two different methods for eliciting confidence — both being used in research on metacognition (Fleming et al., 2012). In the standard two-response design, subjects first reported their decision, and only then, and on a separate scale, reported their confidence. In the less-commonly used one-response design, subjects reported their confidence and decision simultaneously on a single scale. We were interested to see whether the more complex one-response design – in which subjects, in effect, have to perform two tasks at the same time – affected the computations underlying confidence reports as expected under theories of cognitive load (e.g. (Sweller, 1988; Lavie, 2005)) and dual-task interference (e.g. (Kahneman, 1973; Pashler, 1994)).

We used Bayesian model selection to assess how well the models fit our data; thus our analysis was "doubly Bayesian" in that we used Bayesian model selection to test whether our subjects' behaviour was best explained by a Bayes optimal model (Huszár et al., 2010). We found that the commonly used Difference model was the least probable model irrespective of task design. Subjects' confidence reports in the two-response design were far more likely to reflect the Bayes optimal model rather than either heuristic model. In contrast, in the one-response design, the confidence reports of roughly half of the subjects were in line with the Bayes optimal model, and the confidence reports of the other half were in line with the Max model, indicating that, perhaps, the increased cognitive load in the one-response paradigm caused subjects to behave suboptimally. In sum, our results indicate that while it is possible to generate confidence reports using Bayes optimal computations, it is not automatic — and can be promoted by certain types of task.

1.3 Methods

Participants

Participants were undergraduate and graduate students at the University of Oxford. 26 participants aged 18-30 took part in the study. All participants had normal or corrected-to-normal vision. The local ethics committee approved the study, and all participants provided written informed consent.

Experimental details

1.3.0.0.1 Display parameters and response mode. Participants viewed an LED screen (ViewSonic VG2236wm-LED, resolution = 800 × 600) at a distance of 57 cm. The background luminance of the screen was 62.5 cd/m^2. The screen was connected to a personal laptop (Toshiba Satellite Pro C660-29W) via a VGA splitter (Startech 2 Port VGA Video Splitter) and controlled by the Cogent toolbox (http://www.vislab.ucl.ac.uk/cogent.php/) for MATLAB (Mathworks Inc). Participants responded using a standard keyboard.

1.3.0.0.2 Design and procedure. Participants performed a two-interval forced-choice contrast discrimination task. On each trial, a central black fixation cross (width: 0.75 degrees of visual angle) appeared for a variable period, drawn uniformly from the range 500-1000 milliseconds. Two viewing intervals were then presented, separated by a blank display lasting 1000 milliseconds. Each interval lasted $\sim$ 83 milliseconds. In each interval, there were six vertically oriented Gabor patches (SD of the Gaussian envelope: 0.45 degrees of visual angle; spatial frequency: 1.5 cycles/degree of visual angle; baseline contrast: 0.10) organised around an imaginary circle (radius: 8 degrees of visual angle) at equal distances from each other.

In either the first or the second interval, one of the six Gabor patches (the visual target) had a slightly higher level of contrast than the others. The interval and location of the visual target were randomized across trials. The visual target was produced by adding one of 4 possible values (0.015, 0.035, 0.07, 0.15) to the baseline contrast (0.10) of the respective Gabor patch.

After the second interval there was a blank display, which lasted 500 milliseconds, and a response display. The response display prompted participants to indicate which interval they thought contained the visual target and how confident they felt about their decision. Participants were split into two groups. Each group performed a slightly different version of the task. The difference lay only in how

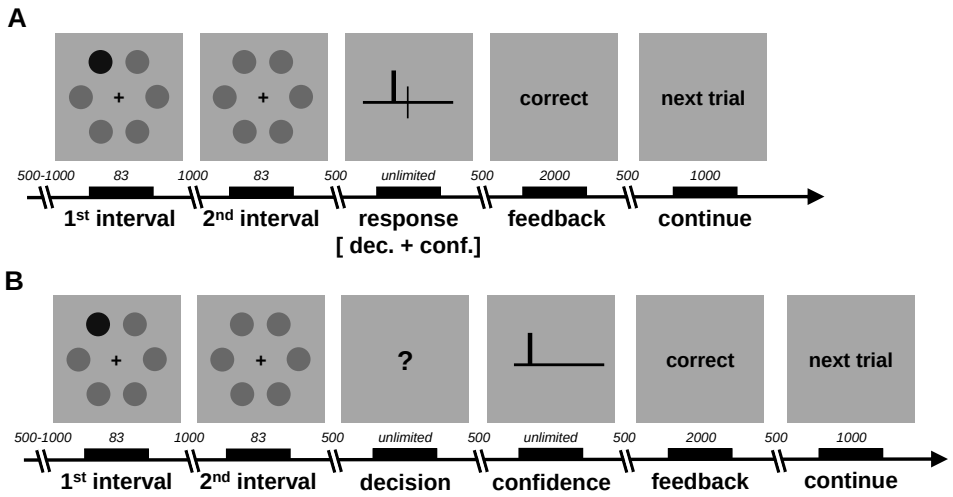

Figure 1.2: Schematic of experimental design and task. **A** One-response design. Participants indicated their decision and their confidence simultaneously. **B** Two-response design. Participants indicated their decision and their confidence sequentially. The displays have been edited for ease of illustration (e.g. Gabor patches are shown as dots, with the visual target being the darker dot). All timings are shown in milliseconds. See text for details.

decisions and confidence were indicated; the stimuli seen by the two groups were identical.

For the first group, which had 15 participants, the response display consisted of a central black horizontal line with a fixed midpoint (Figure 1.2A). The region to the left of the midpoint represented the first interval; the region to the right represented the second interval. A vertical white marker was displayed on top of the midpoint. Participants were asked to indicate which interval they thought contained the visual target by moving the vertical marker to the left (first interval) or to the right (second interval) of the midpoint. The marker could be moved along the line by up to six steps on either side, with each step indicating higher confidence (1: "uncertain"; 6: "certain"). Participants pressed "N" or "M" to move the marker left or right, respectively, and locked the marker by pressing "B".

For the second group, which had 11 participants, initially the response display consisted of a central black question mark (Figure 1.2B). Participants indicated which interval they thought contained the visual target, pressing "N" for the first interval and "M" for the second interval. After having indicated their decision, the response display switched to a central black horizontal line. A vertical white marker was displayed at the left extreme of the horizontal line. Participants indicated how confident they felt about their decision by moving the vertical marker along the line by up to six steps, with each step towards the right indicating

higher confidence (1: "uncertain"; 6: "certain"). Participants pressed "N" or "M" to move the marker left or right, respectively, and locked the marker by pressing "B".

After having made their response(s), participants were presented with central text with either "correct" if their decision about the target interval was correct or "wrong" if it it was incorrect. The feedback display lasted 2000 milliseconds. Participants were then presented with central white text saying "next trial" before continuing to the next trial. Participants completed 16 practice trials followed by 480 experimental trials. The two groups were analysed separately. We refer to the two groups as "one-response" and "two-response", respectively.

Confidence models

To model responses, we assumed the following: On each trial, subjects receive a pair of sensory signals, $\mathbf{x}$. Subjects transform those sensory signals into a continuous decision variable, $z^{\mathrm{D}}(\mathbf{x})$, and then compare this variable to a single threshold to make a decision, d. Finally, subjects transform the sensory signals and the decision into a continuous confidence variable, $z^{\mathrm{C}}(\mathbf{x}; d)$, and then compare this variable to a set of thresholds to obtain a confidence report, c. This section starts by describing our assumptions about the sensory signals, $\mathbf{x}$, then moves on to the models for how subjects might compute their decision and confidence variables. Finally, we describe the Bayesian inference technique used to fit the parameters and find the most probable model.

1.3.0.0.3 Sensory signals. We assumed that subjects on each trial receive two sensory signals, $\mathbf{x} = (x_1, x_2)$, drawn from two different Gaussian distributions, with x_1 giving information about interval 1 and x_2 giving information about interval 2. If the target is in interval 1, then

$$P(x_1|s, i = 1, \sigma) = \mathcal{N}\left(x_1; s, \sigma^2/2\right) \tag{1.1a}$$

$$P(x_2|s, i = 1, \sigma) = \mathcal{N}\left(x_2; 0, \sigma^2/2\right), \tag{1.1b}$$

whereas if the visual target is in interval 2, then

$$P(x_1|s, i = 2, \sigma) = \mathcal{N}\left(x_1; 0, \sigma^2/2\right) \tag{1.2a}$$

$$P(x_2|s, i = 2, \sigma) = \mathcal{N}\left(x_2; s, \sigma^2/2\right). \tag{1.2b}$$

Here s specifies the contrast added to the visual target, $s \in \{0.015, 0.035, 0.07, 0.15\}$ as described in Design and Procedure, $i \in \{1, 2\}$ denotes the target interval, and σ characterizes the level of noise in the subject's perceptual system. The variance of each sensory signal is $\sigma^2/2$, which means

that the variance of $x_2 - x_1$ is σ^2 as commonly assumed by psychophysical models.

1.3.0.0.4 Decision and confidence variables. We considered three models for how subjects compute their decision variable, $z^{\mathrm{D}}(\mathbf{x})$, and their confidence variable, $z^{\mathrm{C}}(\mathbf{x}; d)$. We refer to these models as the Difference model (Δ), the Max model (M), and the Bayesian model (B). The Difference model proposes that the decision and the confidence variable reflect the difference between the two sensory signals,

$$z^{\mathrm{D}}_{\Delta}(\mathbf{x}) = x_2 - x_1 \tag{1.3}$$

$$z^{\mathrm{C}}_{\Delta}(\mathbf{x}; d) = \begin{cases} x_1 - x_2 & \text{for } d = 1 \\ x_2 - x_1 & \text{for } d = 2\,. \end{cases} \tag{1.4}$$

In the next section we discuss how the decision, d (which is 1 for interval 1 and 2 for interval 2) is made.

The Max model proposes that the decision variable reflects the difference between the two sensory signals and the confidence variable reflects only the sensory signal received from the selected interval,

$$z^{\mathrm{D}}_{\mathrm{M}}(\mathbf{x}) = x_2 - x_1, \tag{1.5}$$

$$z^{\mathrm{C}}_{\mathrm{M}}(\mathbf{x}; d) = x_d. \tag{1.6}$$

Finally, the Bayesian model proposes that the decision variable reflects the probability that interval 2 contained the visual target, and that the confidence variable reflects the probability that the decision about the target interval is correct,

$$z^{\mathrm{D}}_{\mathrm{B}}(\mathbf{x}) = P(i = 2 | x_1, x_2, \sigma) \tag{1.7}$$

$$z^{\mathrm{C}}_{\mathrm{B}}(\mathbf{x}; d) = P(i = d | x_1, x_2, \sigma)\,, \tag{1.8}$$

where

$$P(i = d | x_1, x_2, \sigma) = \frac{\sum_s P(x_1 | s, i = d, \sigma)\, P(x_2 | s, i = d, \sigma)}{\sum_{s,i'} P(x_1 | s, i = i', \sigma)\, P(x_2 | s, i = i', \sigma)}. \tag{1.9}$$

To derive this expression, we used Bayes' theorem and assumed that the two conditions have equal prior probability ($P(i = 1) = P(i = 2) = 1/2$). The three models make different predictions about how the sensory signals contribute to the confidence variable, $z^{\mathrm{C}}(\mathbf{x}; d)$, and therefore give rise to different confidence reports.

1.3.0.0.5 Choosing decisions and confidence reports. To make a decision, the subject compares the decision variable to a single threshold, and chooses interval 2 if the variable is larger than the threshold, and interval 1 otherwise,

$$d(\mathbf{x}) = \begin{cases} 2 & \text{if } z^{\mathrm{D}}(\mathbf{x}) > \theta^{\mathrm{D}} \\ 1 & \text{otherwise.} \end{cases} \tag{1.10}$$

Likewise, to choose a confidence level, the subject compares their confidence variable to a set of thresholds, and the confidence level is then determined by the pair of thresholds that the confidence variable lies between. More specifically, the mapping from a confidence variable, $z^{\mathrm{C}}(\mathbf{x}; d)$, to a confidence report, c, is determined implicitly by,

$$\theta^{\mathrm{C}}_{d,c-1} < z^{\mathrm{C}}(\mathbf{x}; d) \leq \theta^{\mathrm{C}}_{d,c}, \tag{1.11}$$

Valid confidence values, c, run from 1 to 6; to ensure that the whole range of $z^{\mathrm{C}}(\mathbf{x}; d)$ is covered, we set $\theta_{d,0} = -\infty$ and $\theta_{d,6} = +\infty$.

Finally, we assumed that with some small probability b, subjects lapsed — they made a random decision and chose a random confidence level. Inclusion of this so-called lapse rate accounts for trials in which subjects made an otherwise low-probability response; e.g. they chose the first interval when there was strong evidence for the second. Such trials are probably due to some error (e.g. motor error or confusion of the two intervals), and if we did not include a lapse rate to explain these trials, they could have a strong effect on model selection.

Model comparison

We wish to compute the probability of the various models given our data. The required probability is, via Bayes' theorem,

$$P(m|\text{data}) \propto P(m)\, P(\text{data}|m) \tag{1.12}$$

where m is either Δ (Difference model), M (Max model) or B (Bayesian model). The data from subject l consists of two experimenter-defined variables: the target intervals, $\mathbf{i}_l$, and the target contrasts, $\mathbf{s}_l$, and two subject-defined variables: the subject's decisions, $\mathbf{d}_l$, and the subject's confidence reports, $\mathbf{c}_l$. Here, the bold symbols denote a vector, listing the value of that variable on every trial, for instance the interval on the k^{th} trial is i_{lk}. We fit different parameters to every subject, so the full likelihood, $P(\text{data}|m)$, is given by a product of single-subject

likelihoods,

$$P(\text{data}|m) = \prod_l P(\mathbf{d}_l, \mathbf{c}_l, \mathbf{i}_l, \mathbf{s}_l|m)\,. \tag{1.13}$$

Because $\mathbf{i}_l$ and $\mathbf{s}_l$ are independent of the model, m, we may write

$$P(\text{data}|m) \propto \prod_l P(\mathbf{d}_l, \mathbf{c}_l|\mathbf{i}_l, \mathbf{s}_l, m)\,. \tag{1.14}$$

To compute the single-subject likelihood we cannot simply choose one setting for the parameters, because the data does not pin down the exact value of the parameters. Instead we integrate over possible parameter settings,

$$P(\mathbf{d}_l, \mathbf{c}_l|\mathbf{i}_l, \mathbf{s}_l, m) = \int P(\mathbf{d}_l, \mathbf{c}_l|\mathbf{i}_l, \mathbf{s}_l, m, \boldsymbol{\theta}_l, \sigma_l, b_l)\, P(\boldsymbol{\theta}_l)\, P(\sigma_l)\, P(b_l)\, d\boldsymbol{\theta}_l d\sigma_l db_l, \tag{1.15}$$

where $\boldsymbol{\theta}_l$ collects that subject's decision and confidence thresholds. This integral is large if the best fitting parameters explain the data well (i.e. if $P(\mathbf{d}_l, \mathbf{c}_l|\mathbf{i}_l, \mathbf{s}_l, m, \boldsymbol{\theta}_l, \sigma_l, b_l)$ is large for the best fitting parameters), as one might expect. However, this integral also takes into account a second important factor, the robustness of the model. In particular, a good model is not overly sensitive to the exact settings of the parameters — so you can perturb the parameters away from the best values, and still fit the data reasonably well. This integral optimally combines these two contributions: how well the best fitting model explains the data, and the model's robustness. For a single subject (dropping the subject index, l, for simplicity, but still fitting different parameters for each subject), the probability of $\mathbf{d}$ and $\mathbf{c}$ given that subject's parameters is the product of terms from each trial,

$$P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m, \boldsymbol{\theta}, \sigma, b) = \prod_k P(d_k, c_k|i_k, s_k, m, \boldsymbol{\theta}, \sigma, b)\ , \tag{1.16}$$

We therefore need to compute the probability of a subject making a decision, d_k, and choosing a confidence level, c_k, given the subject's parameters, the target interval, i_k, and target contrast, s_k. We do this numerically, by sampling: given a set of parameters, $\boldsymbol{\theta}$, σ and b we generate an $\mathbf{x}$ from either Eq. (1.1) or (1.2) (depending on whether i_k is 1 or 2). We compute $z^{\mathrm{D}}(\mathbf{x})$ from either Eq. (1.3), (1.5) or (1.7) (depending on the model), and threshold $z^{\mathrm{D}}(\mathbf{x})$ to get a decision, d. Next, we combine $\mathbf{x}$ and d to compute $z^{\mathrm{C}}(\mathbf{x}; d)$ from either Eq. (1.4), (1.6) or (1.8) (again, depending on the model), and threshold $z^{\mathrm{C}}(\mathbf{x}; d)$ to get a confidence report, c. We do this many times (10^5 in our simulations); $P(d_k, c_k|i_k, s_k, m, \boldsymbol{\theta}, \sigma, b)$ is proportion of times the above procedure yields $d = d_k$ and $c = c_k$.

To perform the integral in Eq. (1.15), we must specify prior distributions over the

parameters σ, b and $\boldsymbol{\theta}$. While it is straightforward to write down sensible priors over two of these parameters, σ and b, it is much more difficult to write down a sensible prior for the thresholds, $\boldsymbol{\theta}$. This difficulty arises because the thresholds depend on $z^{\mathrm{D}}(\mathbf{x})$ and $z^{\mathrm{C}}(\mathbf{x}; d)$, which change drastically from model to model. To get around this difficulty, we reparametrise the thresholds, as described in the next section.

1.3.0.0.6 Representation of thresholds. We reparametrise the decision and confidence thresholds in essentially the same way, but it is helpful to start with the decision threshold, as it is simpler. We exploit the fact that for a given model, there is a one to one relationship between the threshold, θ^D, and the probability that the subject chooses interval 1,

$$p_{d=1} \equiv P(d = 1|m, \boldsymbol{\theta}, \sigma, b) = \int_{-\infty}^{\theta^D} P(z^{\mathrm{D}}|m, \sigma, b)\, dz^{\mathrm{D}}\,. \tag{1.17}$$

Therefore, if we specify the threshold, we specify $p_{d=1}$. Importantly, the converse is also true: if we specify $p_{d=1}$, we specify the threshold. Thus, we can use $p_{d=1}$ to parametrise the threshold. To compute the threshold from $p_{d=1}$, we represent $P(z^{\mathrm{D}}|m, \sigma, b)$ using samples of z^{D}, which we can compute as described at the end of the previous section. To find the threshold, we sweep across possible values for the threshold, until the right proportion of samples are below the threshold ($p_{d=1}$), and the right proportion of samples are above the threshold ($p_{d=2}$).

The situation is exactly the same for confidence reports: if we specify the thresholds, we specify the distribution over confidence reports, $p_{c|d}$,

$$p_{c|d} \equiv P(c|d, m, \boldsymbol{\theta}, \sigma, b) = \int_{\theta_{d,c-1}}^{\theta_{d,c}} P(z^{\mathrm{C}}|d, m, \sigma, b)\, dz^{\mathrm{C}}\,. \tag{1.18}$$

Combining decision and confidence thresholds, we obtain the joint distribution over decisions and confidence reports, $\mathbf{p}$, whose elements are

$$p_{d,c} \equiv P(d, c|m, \boldsymbol{\theta}, \sigma, b)\,, \tag{1.19}$$

Thus, specifying the confidence and decision threshold specifies the joint distribution over decisions and confidence reports, $\mathbf{p}$. Importantly, the reverse is also true: specifying $\mathbf{p}$ specifies the confidence and decision thresholds. In order to find the confidence thresholds, we take the same strategy as for decisions — we represent $P(z^{\mathrm{C}}|d, m, \sigma, b)$ using samples of z^{C}, then sweep across all possible values for the thresholds, until we get $c = 1$ the right fraction of the time (i.e. $p_{c=1|d}$), and $c = 2$ the right fraction of the time (i.e. $p_{c=2|d}$) etc. (see Figure 1.3 or 2 for a schematic diagram of this method). Note that, to condition on a particular decision, we simply throw away those values of z^{C} associated with the wrong

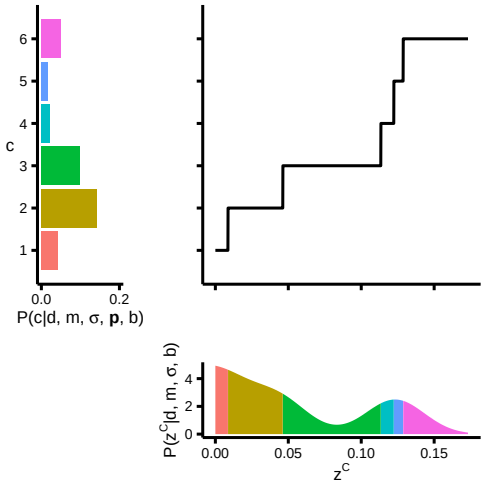

Figure 1.3: Schematic diagram of our method for mapping thresholds to confidence probabilities. The lower panel displays the (fixed) distribution over z^{C}, $P(z^{\mathrm{C}}|d, m, \sigma, b)$ (which does not depend on the thresholds). The left panel displays the distribution over confidence reports, determined by $\mathbf{p}$. The large central panel displays the fitted function mapping from z^{C} to c, which consists of a set of jumps, with each jump corresponding to a threshold. The thresholds are chosen so that the total probability density in $P(z^{\mathrm{C}}|d, m, \sigma, b)$ between jumps is exactly equal to the probability of the corresponding confidence level (see colours).

decision.

1.3.0.0.7 Performing the integral in Eq. (1.15). Changing the representation from thresholds, $\boldsymbol{\theta}$, to probabilities, $\mathbf{p}$, gives a new single-subject likelihood,

$$P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m) = \int P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m, \mathbf{p}, \sigma, b)\, P(\mathbf{p})\, P(\sigma)\, P(b)\, d\mathbf{p} d\sigma db. \tag{1.20}$$

To perform the integral, we need to specify prior distributions over the parameters, σ, b, and $\mathbf{p}$. For σ, we use

$$P(\sigma) = \mathrm{Gamma}\,(2, 0.05) \propto \sigma e^{-\sigma/0.05} \tag{1.21}$$

as this broadly covered the range of plausible values of σ. We chose a very broad range of values for b — evenly distributed in log space between 10^{-3} and 10^{-1},

$$P(\log_{10} b) = \mathrm{Uniform}\,(-3, -1)\,. \tag{1.22}$$

Finally, we chose an uninformative, uniform prior distribution over $\mathbf{p}$,

$$P(\mathbf{p}) = \mathrm{Dirichlet}\,(\mathbf{p}; \mathbf{1})\,, \tag{1.23}$$

where $\mathbf{1}$ is a matrix whose elements are all 1.

The most straightforward way to compute the single-subject likelihood in Eq. (1.20) is to find the average (expected) value of $P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m, \mathbf{p}, \sigma, b)$ when we sample values of $\mathbf{p}$, σ and b from the prior,

$$P(\text{data}|m) = \mathrm{E}_{P(\mathbf{p})P(\sigma)P(b)} \left[P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m, \mathbf{p}, \sigma, b) \right]. \tag{1.24}$$

However, the likelihood, $P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m, \mathbf{p}, \sigma, b)$, is very sharply peaked; being very high in a very small region around the subject's true parameters, and very low elsewhere. The estimated value of the integral is therefore dominated by the few samples that are close to the true parameters, and as there are only a few such samples, the sample-based estimate of $P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m, \mathbf{p}, \sigma, b)$ has high variance.

Instead, we use a technique called importance sampling. The aim is to find an equivalent expectation, in which the quantity to be averaged does not vary much, allowing the distribution to be estimated using a smaller number of samples — in fact, if the term inside the expectation is constant, then the expectation can be estimated using only one sample. Importance sampling uses

$$P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m) = \mathrm{E}_{Q(\mathbf{p})P(\sigma)P(b)} \left[\frac{P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m, \mathbf{p}, \sigma, b)\, P(\mathbf{p})}{Q(\mathbf{p})} \right]. \tag{1.25}$$

The integral form for this expectation is,

$$P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m) = \int \frac{P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, \mathbf{p}, \sigma, b)\, P(\mathbf{p})}{Q(\mathbf{p})} Q(\mathbf{p})\, P(\sigma)\, P(b)\, d\mathbf{p} d\sigma db, \tag{1.26}$$

which is trivially equal to Eq. (1.20). To ensure that the term inside the expectation in Eq. (1.25) does not vary much, we need to choose the denominator, $Q(\mathbf{p})$, so it is approximately proportional to the numerator, $P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m, \mathbf{p}, \sigma, b)\, P(\mathbf{p})$. To do so, we exploit the fact that the numerator is proportional to a posterior distribution over $\mathbf{p}$ (considering only dependence on $\mathbf{p}$),

$$P(\mathbf{d}, \mathbf{c}|\mathbf{i}, \mathbf{s}, m, \mathbf{p}, \sigma, b)\, P(\mathbf{p}) \propto P(\mathbf{p}|\mathbf{d}, \mathbf{c}, \mathbf{i}, \mathbf{s}, m, \sigma, b). \tag{1.27}$$

Remembering that $p_{d,c}$ is just the probability of a particular decision and confidence value, aggregating across all trial types, it is straightforward to construct a good approximation to the posterior over $\mathbf{p}$. In particular, we ignore the influence of $\mathbf{i}$, $\mathbf{s}$, m, σ and b, so the only remaining information is the decisions and the confidence reports, $\mathbf{d}$ and $\mathbf{c}$, irrespective of trial-type. These variables can be summarised by $\mathbf{n}$, where $n_{d,c}$, is the number of times that a subject chose decision d and confidence level c. The resulting distribution over $\mathbf{p}$ can be written,

$$Q(\mathbf{p}) = P(\mathbf{p}|\mathbf{d}, \mathbf{c}) = \text{Dirichlet}(\mathbf{p}; \mathbf{1} + \mathbf{n}), \tag{1.28}$$

which turns out to be a good proposal distribution for our importance sampler.

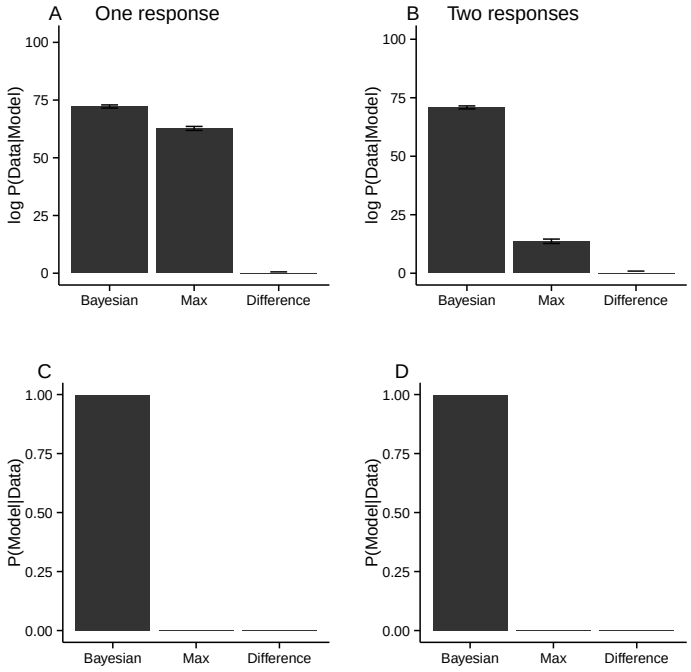

Figure 1.4: The probability of the three models given the data. **AB** The log-likelihood differences between the models, using the Difference model as a baseline. Note the small error bars, representing two standard-errors, given by running the algorithm 10 times, and each time using 1000 samples to estimate the model evidence (Eq. (1.25)). **CD** The posterior probability of the models, assuming a uniform prior. Left column, one response. Right column, two responses.

1.4 Results

Model selection

To compare models, we look at the posterior probability of each of our models given the data, $P(m|\text{data})$. As, a-priori, we have no reason to prefer one model over another, we use a uniform prior, $P(m) = 1/3$, so, assuming that every subject uses the same model, then the posterior is proportional to $P(\text{data}|m)$, which we showed how to compute in the Model Comparison Section. The Bayesian model is better by a factor of around 10^4 for the one-response data and around 10^{25} for the two-response data (Figure 1.4).

For the above model comparison, we assumed that all subjects used the same model to generate their confidence reports. It is quite possible, however, that different subjects use different models to generate their confidence reports. In particular, we might expect that there is some probability with which a random subject uses each model, $P(m_l)$ (where l is the subject index, so m_l is the model chosen by subject l). Under this assumption, we can analyse how well the models fit the data by inferring the probability with which subjects choose

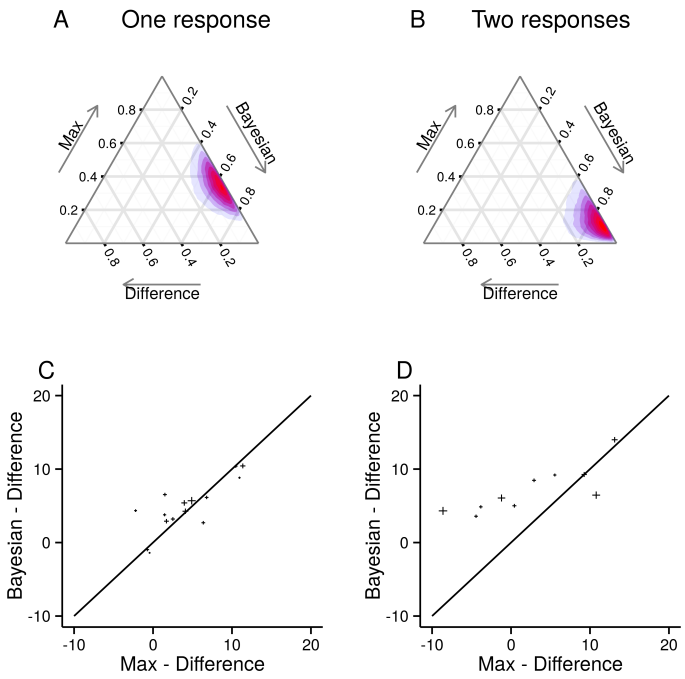

Figure 1.5: Single-subject analysis. **AB** Subjects are assumed to use each model with some probability. The coloured regions represent plausible settings for these probabilities. For the one-response dataset, we see that subjects are roughly equally likely to use the Max and Bayesian models. For the two-responses dataset, we see that subjects are far more likely to use the Bayesian model. To read these plots, follow the grid lines in the same direction as axis ticks and labels, so for instance, lines of equal probability for the Max model run horizontally, and lines of equal probability for the Bayesian model run up and to the right. **CD** The difference in log-likelihood between the Bayesian model and the Difference model (on the y-axis) against the difference in log-likelihood between the Max model and Difference model (on the x-axis). The size of the crosses represents the uncertainty (two standard errors) along each axis (based on the 10 runs of the model selection procedure, mentioned in Figure 1.4).

to use each model, $P(m_l)$, using a variational Bayesian method presented by Stephan et al. (2009). In agreement with the previous analysis, we find that for the two-response dataset, the probability of any subject using the Bayesian model is high: subjects are significantly more likely to use the Bayesian model than either the Max or Difference models ($p < 0.006$; exceedence probability (Stephan et al., 2009); Figure 1.5B). For the one-response dataset, on the other hand, subjects use the Bayesian model only slightly more than the Max model (Figure 1.5A). The log-likelihood differences for individual subjects are plotted in Figure 1.5CD, with uncertainty given by the size of the crosses. Again, for the two-response dataset, but not for the one-response dataset, the difference between each subject's log-likelihood for the Bayesian and Max models is larger than 0 (two-response: $t(10) = 3.47, p < .006$; one-response: $t(14) = 0.954, p \approx .35$; two-sided one-sample t-test).

Model fits

While the model evidence is the right way to compare models, it is important to check that the inferred models and parameter settings (for inferred parameters for each subject see Tables 1.1 and 1.2) are plausible. We therefore plotted the raw data — the number of times a participant reported a particular decision and confidence level for a particular target interval and target contrast — along with the predictions from the Bayesian model. In particular, in Figure 1.6, we plot fitted and empirical distributions over confidence reports given a target interval and contrast from an example participant (for all subjects and all models see Figs 1.10 and 1.11). To make this comparison, we defined "signed confidence", whose absolute value gives the confidence level, and whose sign gives the decision,

$$\text{Signed confidence} = \begin{cases} -c & \text{for } d = 1 \\ c & \text{for } d = 2. \end{cases} \tag{1.29}$$

These plots show that our model is, at least, plausible, and highlights the fact that our model selection procedure is able to find extremely subtle differences between models. Plotting psychometric curves (Figure 1.7) gave similar results. Again, to plot psychometric curves, we defined "signed contrast", whose absolute value gives the contrast, and whose sign gives the target interval,

$$\text{Signed contrast} = \begin{cases} -s & \text{for } i = 1 \\ s & \text{for } i = 2. \end{cases} \tag{1.30}$$

Differences between models

For model selection to actually work, there need to be differences between the predictions made by the three models. Here, we show that the models do indeed make different predictions under representative settings for the parameters.

To understand which predictions are most relevant, we have to think about exactly what form our data takes. In our experiment, we present subjects with a target in one of the two intervals, i, with one of four contrast levels, s, then observe their decision, d and confidence report, c. Overall, we therefore obtain an empirical estimate of each subject's distribution over decision and confidence reports (or equivalently signed confidence, see previous section), given a target interval and contrast. This suggests that we should examine the predictions that each model makes about each subject's distribution over decisions and confidence reports, given the target interval, i, and contrast, s. While these distributions are superficially very similar (Figure 1.8), closer examination reveals two interesting, albeit small, differences. Importantly, these plots display theoretical, and

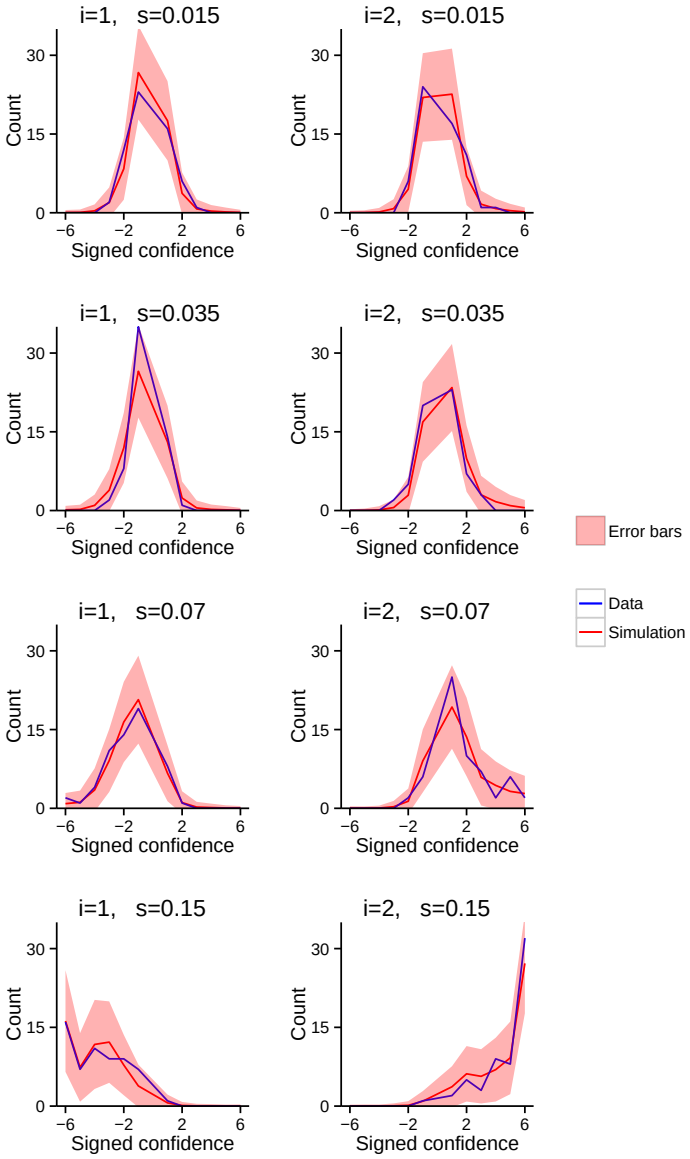

Figure 1.6: Simulated (Bayesian model) and actual confidence distributions for one subject (one response), and each target interval and contrast. The plots on the left are for targets in interval 1 (i.e. $i = 1$), whereas the plots on the right are for targets in interval 2 (i.e. $i = 2$). We use signed confidence on the horizontal axis (the sign indicates the decision, and the absolute value indicates the confidence level). The blue line is the empirically measured confidence distribution. The red line is Bayesian model's fitted confidence distribution. The red area is the region around the fitted mean confidence distribution that we expect the data to lie within. We computed the error bars by sampling settings for the model parameters, then sampling datasets conditioned on those parameters. The error bars represent two standard deviations of those samples. This plot demonstrates that the Bayesian model is, at least, plausible.

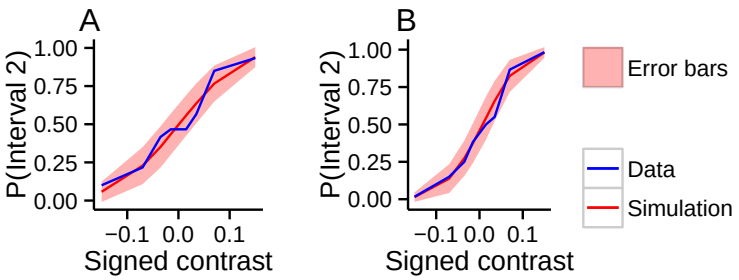

Figure 1.7: Simulated (Bayesian model) and actual psychometric curves for two subjects. The horizontal axis displays signed contrast (the sign gives the target interval, and the absolute value gives the contrast level). Colour code is the same as in Figure 1.6: the blue line is the empirically measured psychometric curve; the red line is the Bayesian model's fitted mean psychometric curve; and the red area represents Bayesian error bars. **A** One subject from the one-response design. **B** One subject from the two-response design. As with Figure 1.6, this plot demonstrates the plausibility of the Bayesian model.

hence noise-free results, so even small differences are meaningful, and are not fluctuations due to noise.

First, the Max model differs from the other two models at intermediate contrast levels, especially $s = 0.07$, where the Max model displays bimodality in the confidence distribution. In particular, and unexpectedly, an error with confidence level 1 is less likely than an error with confidence levels 2 to 4. In contrast, the other models display smooth, unimodal behaviour across the different confidence levels. This pattern arises because the Max model uses only one of the two sensory signals. For example, when $s = 0.07$ and $i = 2$ (so the target is fairly easy to see, and is in interval 2), then x_2 is usually large. Therefore, for x_1 to be larger than x_2, prompting an error, x_1 must also be large. Under the Max model, x_1 being large implies high confidence, and, in this case, a high confidence error.

Second, the three models exist on a continuum, with the Max model using the narrowest range of confidence levels, the Bayesian model using an intermediate range, and the Difference model using the broadest range. These trends are particularly evident at the lowest and highest contrast levels. At the lowest contrast level, $s = 0.015$, the distribution for the Max model is more peaked, whereas the distribution for the Difference model is lower and broader, and the Bayesian model lies somewhere between them. At the highest contrast level, $s = 0.15$, the Max model decays most rapidly, followed by the Bayesian model, and then the Difference model.

To understand this apparent continuum, we need to look at how the models map sensory data, defined by x_1 and x_2, onto confidence reports. We therefore plotted black contours dividing the regions of sensory-space (i.e. (x_1, x_2)-space)

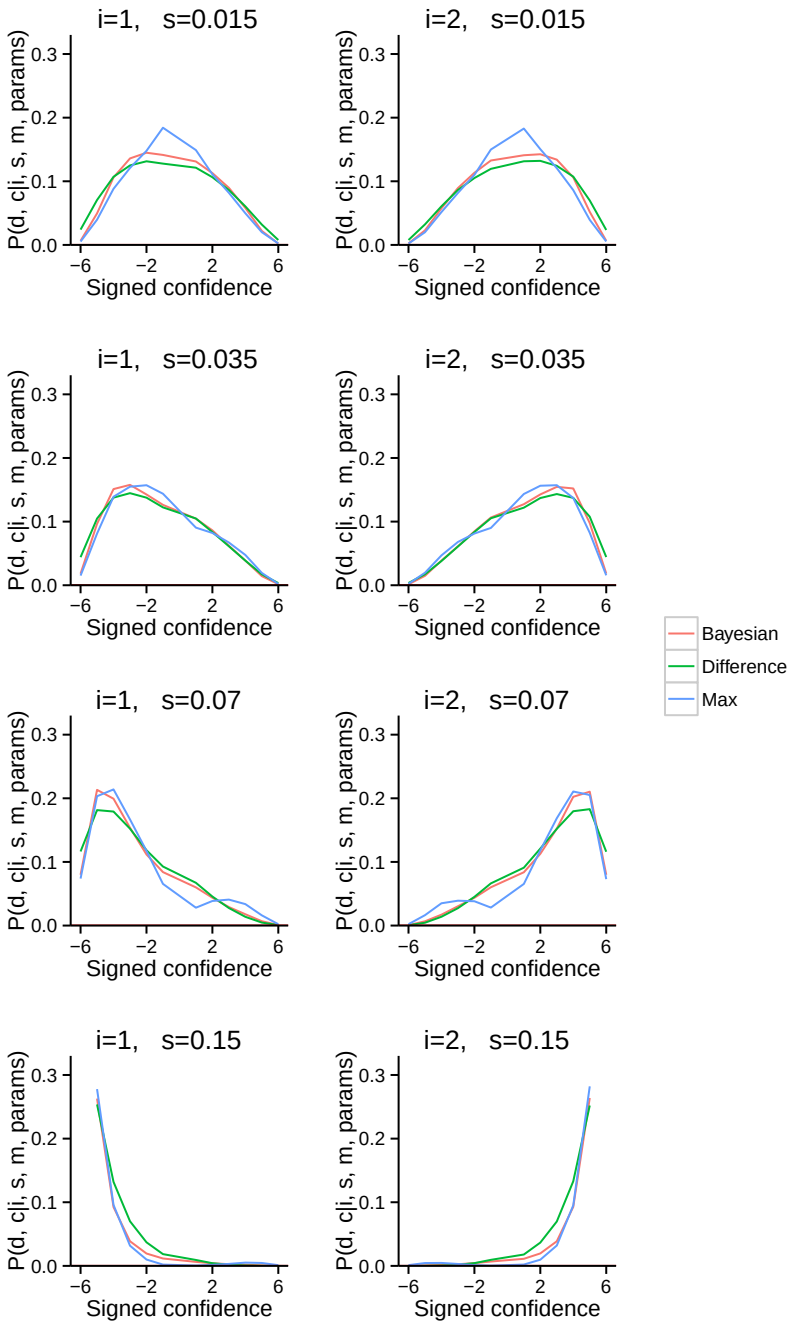

Figure 1.8: Different models lead to different distributions over confidence. Same as Figure 1.6, but displaying theoretical distributions induced by the three different models. The parameters were not fit to data; instead, they were set to fixed (but reasonable) values: $\sigma = 0.07$, $b = 0$ and $p_{d,c} = 1/12$.

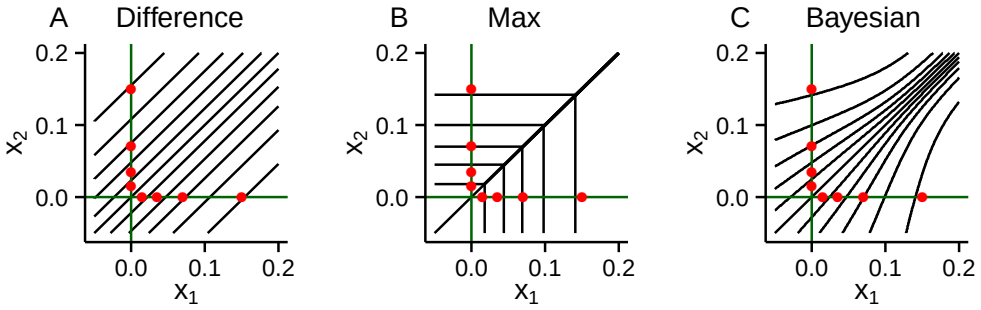

Figure 1.9: The mapping from stimulus-space to confidence induced by different models. The axes represent the two stimulus dimensions (cf. interval 1 and 2). The red dots represent the mean values of x_1 and x_2 for each stimulus. The black lines separate regions in stimulus space that map to a given confidence level. **A** Difference model. **B** Max model. **C** Bayesian model. The model parameters are the same as in Figure 1.8.

that map to different confidence levels (Figure 1.9). These plots highlight striking differences between the models. In particular, the Difference model has diagonal contours, whereas the Max model has contours that run horizontally, vertically or along the central diagonal at $x_1 = x_2$. In further contrast, the Bayesian model has curved contours with a shape somewhere between the Difference and Max models. In particular, for large values of x_1 and x_2, the contours are almost diagonal, as in the Difference model whereas for small values of x_1 and x_2, the contours are more horizontally or vertically aligned, as in the Max model.

To see how differences in the mapping from sensory-space to confidence reports translate into differences in the probability distribution over confidence reports, we consider the red dots, representing different target intervals and contrasts. For instance, a high-contrast target in interval 2 ($s = 0.15$), is represented by the uppermost red dot in each subplot. Importantly, red dots representing stimuli lie along the horizontal and vertical axes (green). The angle at which the contours cross these axes therefore becomes critically important. In particular, for the Difference model the contours pass diagonally through the axes, and therefore close to many red dots (representing stimuli), giving a relatively broad range of confidence levels for each stimulus type. In contrast, for the Max model, the contours pass perpendicularly through the axes, minimizing the number of red dots (representing stimuli) that each contour passes close to, giving a narrower range of confidence levels for each stimulus type. The contours of the Bayesian model pass through the axes at an angle between the extremes of the Difference and Max models — as expected, giving rise to a range of confidence levels between the extremes of the Difference and the Max model.

In principle, these differences might allow us to choose between models based only on visual inspection of $P(d, c|i, s, m, \text{params})$. However, in practice, the distribution over decision and confidence reports, averaging over trial type, $p_{d,c}$ is

not constant, as we assumed above, but is far more complicated. This additional complexity makes it impossible to find the correct model by simple visual inspection. More powerful methods, like Bayesian model selection, are needed to pick out these differences.

1.5 Discussion

We tested whether subjects' confidence reports in a visual two-interval forced-choice task reflect heuristic or Bayes optimal computations. We assumed that subjects receive a two-dimensional sensory signal, $\mathbf{x}$, and, based on that signal, make a decision (about which interval a target is in), and report their confidence in that decision. We also assumed that this process is mediated by intermediate variables: subjects transform those sensory signals into a continuous decision variable, $z^{\mathrm{D}}(\mathbf{x})$, compare this variable to a single threshold to make a decision, d, transform the sensory signals and the decision into a continuous confidence variable, $z^{\mathrm{C}}(\mathbf{x}; d)$, and compare this variable to a set of thresholds to obtain a confidence level, c. We compared three possible ways of computing the confidence variable, $z^{\mathrm{C}}(\mathbf{x}; d)$: the Difference model, which computes the difference between the sensory signals; the Max model, which uses only the sensory signal from the selected interval; and the Bayesian model, which computes the probability that a correct decision has been made. We used Bayesian model selection to directly compare these models. For the more standard, and perhaps more natural, design in which subjects first make a decision, and only then give a confidence rating (i.e. the two-response design), the Bayesian model emerged as the clear winner. However, for the less standard design, in which subjects make a decision and give a confidence rating simultaneously (i.e. the one-response design), the results were more ambiguous — our data indicated that around half of the subjects favoured the Bayesian model while the other half favoured the Max model.

One possible reason for the difference is that, in the one-response design, the computations underlying confidence reports were simplified so as not to interfere with the computation of the decision, as expected under theories of cognitive load (e.g. (Sweller, 1988; Lavie, 2005)) and dual-task interference (e.g. (Kahneman, 1973; Pashler, 1994)). Alternatively, despite the instructions being the same, the two types of task design might simply promote qualitatively different computations, with the one-response design promoting a "first-order" judgement about the stimulus intensity, whereas the two-response design promotes a "second-order" judgement about the correctness of a decision which — perhaps critically — has already been made. Surprisingly, the commonly used Difference model was by far the least probable model in both task designs.

A caveat in any Bayesian model selection is that we cannot test all possible

heuristic computations. However, given the results in Figure 1.8 and 1.9, it seems our three models range across the continuum of sensible models — though it is certainly possible that, perhaps, the best model (at least for the one-response data) sits somewhere between the Bayesian and the Max models. More generally, our results indicate that very subtle changes in a task can lead to large changes in the computations performed, and in particular whether subjects use Bayes optimal computations.

Relation to other studies

Barthelmé and Mamassian (2009) went part-way towards realizing the potential of using multidimensional stimuli. Subjects were asked to indicate which of two Gabor patches they would prefer to make an orientation judgement about. Interestingly, and in contrast to our results, they found that subjects were more likely to use a heuristic strategy (similar to the Max model) than a Bayes optimal strategy. However, there were three aspects of their study that make it potentially less relevant to the question of whether confidence reports reflects Bayes optimal computations. First, our model selection procedure is fully Bayesian, and therefore takes account of uncertainty in model predictions, whereas their procedure was not. In particular, under some circumstances a model will make strong predictions (e.g. "the subject must make this decision"), whereas under other circumstances, the model might make weaker predictions (e.g. "the subject is most likely to make this decision, but I'm not sure — they could also do other things"). Bayesian model selection takes into account the strength or weakness of a prediction. Second, in real life (and in our study), people tend to report confidence using verbal (e.g. "not sure" to "very" sure) or numerical (e.g. 1 to 10) scales. In contrast, in Barthelmé and Mamassian (2009), subjects simply made a forced choice between two stimuli. Third, in their study, the Difference model made exactly the same predictions as the Bayes optimal model, making it impossible to distinguish these computations.

There are, of course, other approaches for addressing the question of whether the confidence variable is Bayes optimal. Barthelmé and Mamassian (2010) showed that subject's confidence variable can take into account two factors (contrast and crowding) that might lead to uncertainty — as opposed to using only one factor. Similarly, de Gardelle and Mamassian (2014) showed that subjects were able to accurately compare the confidence variable across different classes of stimuli (in this case orientation discrimination versus spatial frequency discrimination). These studies provide some, albeit indirect, evidence that confidence reports might indeed reflect probability correct, in agreement with our work.

Variability in confidence

Confidence reports have been observed to vary with a range of factors that we did not consider here. For example, people have been shown to be overconfident about the accuracy of their knowledge-based judgements, but underconfident about the accuracy of their perceptual judgements (see Harvey (1997) for a review). People's general level of confidence may also vary with social context. When groups of people resolve disagreement, the opinions expressed with higher confidence tend to carry more weight (e.g. (Sniezek and Henry, 1989)), so group members tend to increase their confidence to maximize their influence on the group decision (Bang et al., 2014; Mahmoodi et al., 2013). They may also adjust their confidence reports to indicate submission or dominance, or cut their losses if they should turn out to be wrong (e.g. (Fleming and Dolan, 2010)). Lastly, people's confidence reports may vary with more general social factors such as profession, gender and culture: finance professionals are more confident than the average population (e.g. (Broihanne et al., 2014)); men are more confident than women (e.g. (Barber and Odean, 2001)); and people from Western cultures are more confident than people from East Asian cultures (e.g. (Mann et al., 1998)).

Our method allows us to think about the variability in confidence reports as having two dimensions. The first (perhaps more superficial) dimension relates to the average confidence level, or confidence distribution. We might imagine that this dimension is primarily modulated by social context, as described above. The second (perhaps deeper) dimension relates to the computations underlying confidence reports. In our data, there do indeed appear to be individual differences in how people generate their confidence reports, and very subtle changes to the task appear to affect this process. We might therefore expect shifts in how people generate their confidence reports for tasks of different complexity. For example, it is not straightforward to solve general-knowledge questions, such as "What is the capital of Brazil?", using Bayesian inference. While one could in principle compute the probability that one's answer is correct, the computational load may be so high that people resort to heuristic computations (e.g. using the population size of the reported city). Future research should seek to identify how confidence reports change between task domains and social contexts — in particular, whether such changes are mostly due to changes in the computation used to generate the confidence variable (cf., $z^{\mathrm{C}}(\mathbf{x}; d)$), or due to changes in the mapping of this variable onto some confidence scale (e.g. using thresholds to map to a discrete scale).

Two types of optimality

Many studies have asked whether confidence reports, and hence metacognitive ability, are optimal (see Fleming and Lau (2014), for a review of measures of metacognitive ability). However, our work suggests that there are (at least) two kinds of optimality. First, the transformation of incoming data into an internal confidence variable (i.e. $z^{\mathrm{C}}(\mathbf{x}; d)$) could be optimal — that is, computed using Bayesian inference. Second, the mapping of the confidence variable onto some external scale of confidence could be optimal (i.e. $c\left(z^{\mathrm{C}}(\mathbf{x}; d)\right)$), but this depends entirely on the details of the task at hand. For instance, without some incentive structure, there is no reason why subjects should opt for any particular mapping, as long as their mapping is deterministic (i.e. reported confidence increases strictly with their confidence variable). Importantly, it does not seem that subjects use an optimal mapping, as evidenced by the large amount of research on "poor calibration" — that is, the extent to which the reported probability of being correct matches the objective probability of being correct for a given decision problem (e.g. (Harvey, 1997; Moore and Healy, 2008)). Even when there is an incentive structure, subjects only improve their calibration and never reach perfection (e.g. (Fleming and Dolan, 2010; Zylberberg et al., 2014)). Future research should seek to identify why poor calibration arises, and how it can be corrected.

Conclusions

We asked how people generate their confidence reports. Do they take a heuristic approach, and compute some reasonable, but ultimately arbitrary, function of the sensory input, or do they take a more principled approach, and compute the probability that they are correct using Bayesian inference? When subjects first made a decision and then reported their confidence in that decision, we found that their confidence reports overwhelmingly reflected the Bayesian strategy. However, when subjects simultaneously made a decision and reported confidence, we found the confidence reports of around half of the subjects were better explained Bayesian strategy, while the confidence reports of the other half of the subjects were better explained by a heuristic strategy.

1.6 Supplementary Figures

Participant	σ	b	Model
1	0.087	0.016	Difference
2	0.057	0.063	Difference
3	0.067	0.021	Difference
4	0.072	0.033	Difference
5	0.074	0.007	Difference
6	0.087	0.001	Difference
7	0.075	0.003	Difference
8	0.091	0.011	Difference
9	0.067	0.026	Difference
10	0.159	0.003	Difference
11	0.068	0.003	Difference
12	0.061	0.015	Difference
13	0.091	0.004	Difference
14	0.050	0.004	Difference
15	0.128	0.004	Difference
1	0.099	0.004	Max
2	0.072	0.021	Max
3	0.076	0.006	Max
4	0.092	0.007	Max
5	0.086	0.005	Max
6	0.096	0.002	Max
7	0.089	0.001	Max
8	0.105	0.004	Max
9	0.082	0.015	Max
10	0.150	0.001	Max
11	0.080	0.002	Max
12	0.077	0.001	Max
13	0.112	0.001	Max
14	0.060	0.002	Max
15	0.144	0.002	Max
1	0.094	0.001	Bayesian
2	0.066	0.041	Bayesian
3	0.077	0.017	Bayesian
4	0.083	0.059	Bayesian
5	0.082	0.011	Bayesian
6	0.095	0.002	Bayesian
7	0.085	0.008	Bayesian
8	0.107	0.019	Bayesian
9	0.082	0.001	Bayesian
10	0.136	0.002	Bayesian
11	0.075	0.002	Bayesian
12	0.073	0.002	Bayesian
13	0.100	0.002	Bayesian
14	0.058	0.002	Bayesian
15	0.143	0.005	Bayesian

Table 1.1: The best fitting parameters for the one-responses dataset. The first variable, σ, represents the subject's noise level, and the second variable, b, represents their lapse rate. These parameters are sensible: σ is of the order of values used to generate a target Gabor patch, which ranges up to 0.15, and b is typically lower than 1%.

Participant	σ	b	Model
1	0.059	0.002	Difference
2	0.080	0.116	Difference
3	0.053	0.052	Difference
4	0.060	0.008	Difference
5	0.072	0.007	Difference
6	0.075	0.025	Difference
7	0.079	0.004	Difference
8	0.141	0.002	Difference
9	0.092	0.003	Difference
10	0.078	0.016	Difference
11	0.062	0.052	Difference
1	0.073	0.001	Max
2	0.087	0.112	Max
3	0.056	0.068	Max
4	0.074	0.003	Max
5	0.081	0.005	Max
6	0.094	0.022	Max
7	0.090	0.001	Max
8	0.148	0.004	Max
9	0.106	0.018	Max
10	0.096	0.063	Max
11	0.072	0.074	Max
1	0.068	0.002	Bayesian
2	0.097	0.001	Bayesian
3	0.060	0.039	Bayesian
4	0.069	0.002	Bayesian
5	0.079	0.003	Bayesian
6	0.082	0.015	Bayesian
7	0.087	0.006	Bayesian
8	0.134	0.002	Bayesian
9	0.102	0.002	Bayesian
10	0.085	0.018	Bayesian
11	0.079	0.001	Bayesian

Table 1.2: As Table 1.1, but for the two-responses dataset.

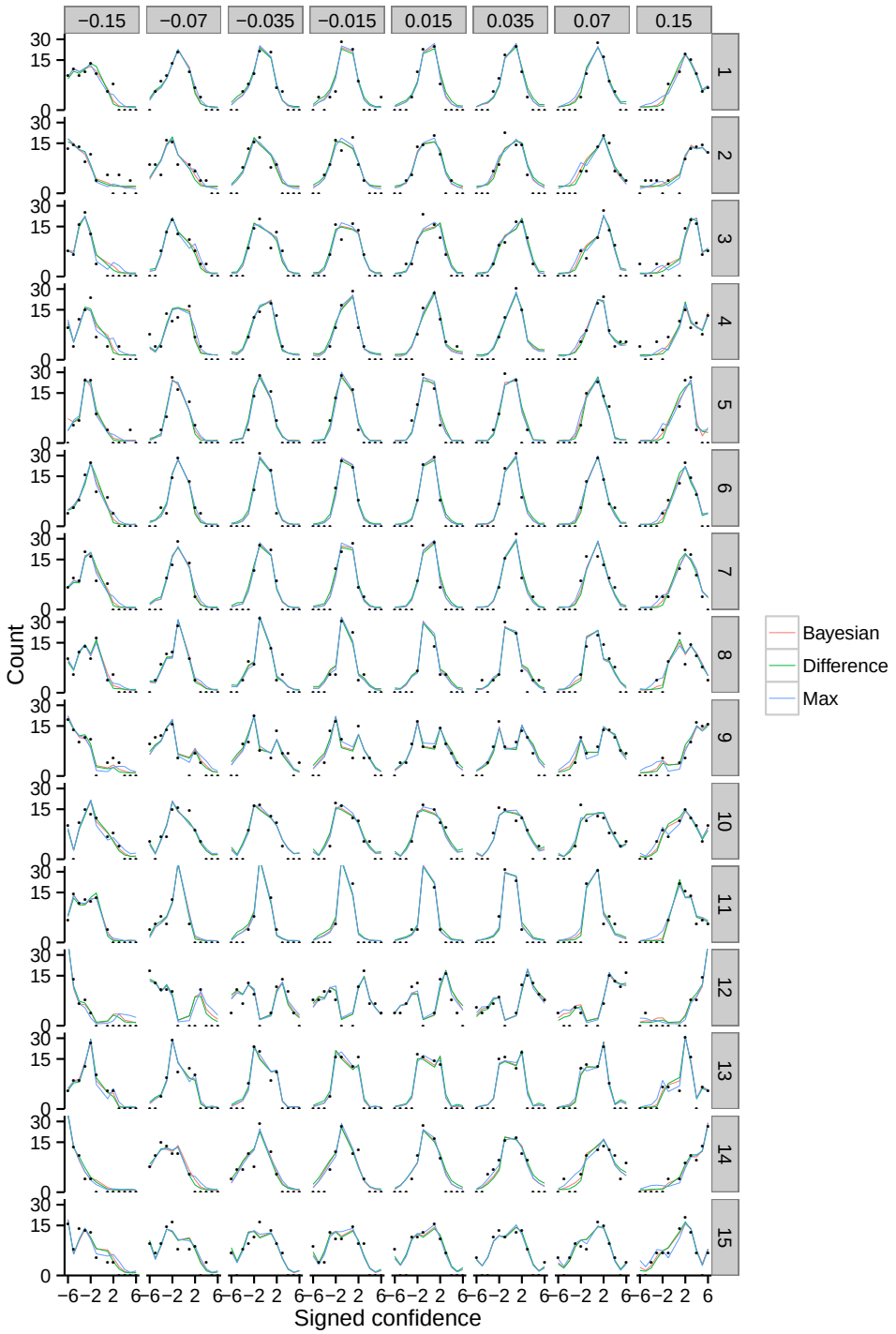

Figure 1.10: The empirical and fitted distributions over signed confidence given the signed contrast for the one-response dataset. The lines show the fitted models, and the points show the data. Each row gives the complete responses for one subject. Each column gives the responses for a particular signed contrast value. The axis has been square-root transformed, in order to emphasize differences in low probabilities.

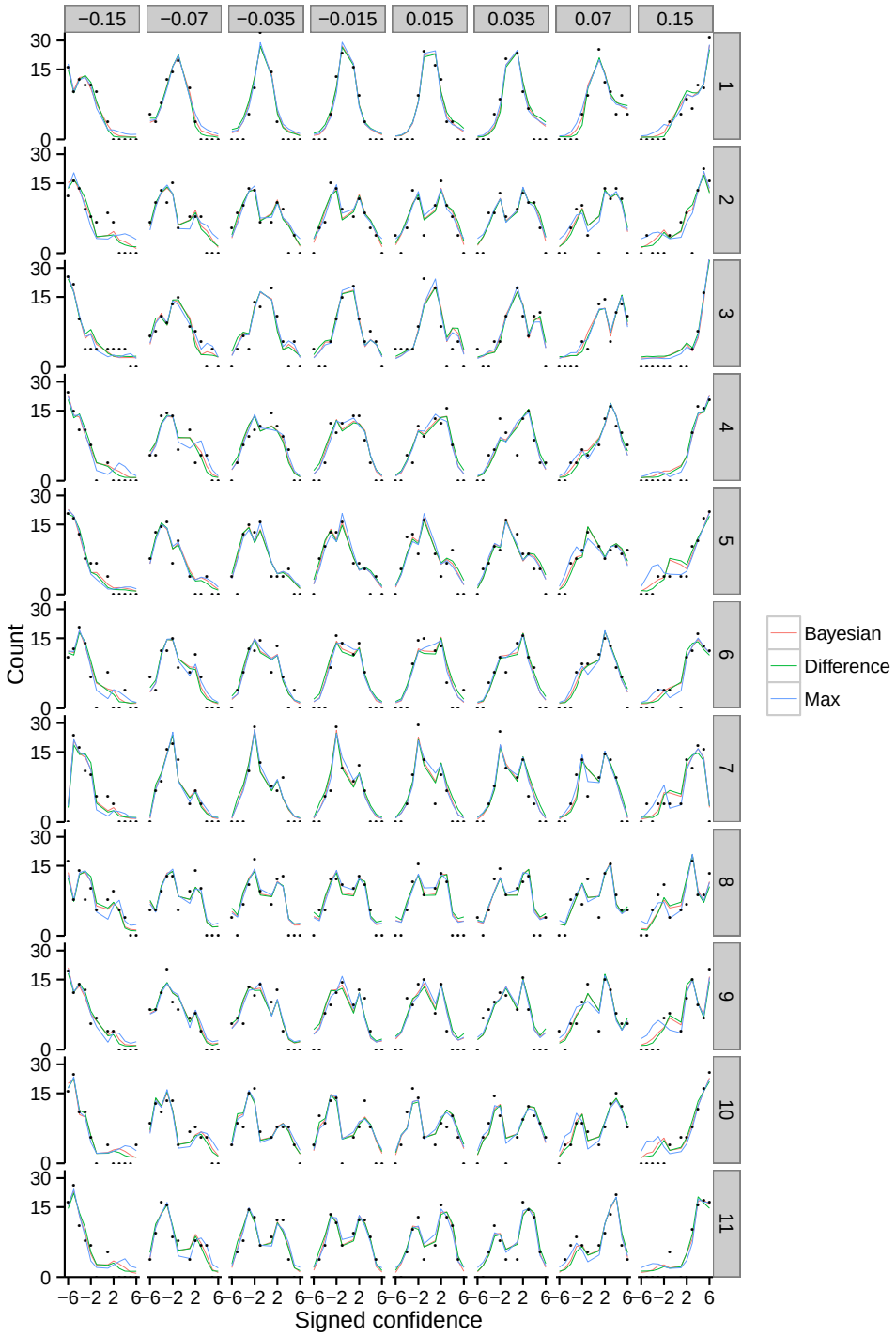

Figure 1.11: As Figure 1.10, but for the two-responses dataset.

Chapter 2

Zipf's law in neural data

2.1 Abstract

Zipf's law, which states that the probability of an observation is inversely proportional to its rank, has been observed in many domains. While there are models that explain Zipf's law in each of them, those explanations are typically domain specific. Recently, methods from statistical physics were used to show that a fairly broad class of models does provide a general explanation of Zipf's law. This explanation rests on the observation that real world data is often generated from underlying causes, known as latent variables. Those latent variables mix together multiple models that do not obey Zipf's law, giving a model that does. Here we extend that work both theoretically and empirically. Theoretically, we provide a far simpler and more intuitive explanation of Zipf's law, which at the same time considerably extends the class of models to which this explanation can apply. Furthermore, we also give methods for verifying whether this explanation applies to a particular dataset. Empirically, these advances allowed us extend this explanation to important classes of data, including word frequencies (the first domain in which Zipf's law was discovered), data with variable sequence length, and multi-neuron spiking activity.

2.2 Introduction

Both natural and artificial systems often exhibit a surprising degree of statistical regularity. One such regularity is Zipf's law. Originally formulated for word frequency (Zipf, 1932), Zipf's law has since been observed in a broad range of domains, including city size (Gabaix, 1999), firm size (Axtell, 2001), mutual fund size (Gabaix et al., 2003), amino acid sequences (Mora et al., 2010), and neural activity (Mora and Bialek, 2011; Tyrcha et al., 2013).

Zipf's law is a relation between rank order and frequency of occurrence: it states that when observations (e.g., words) are ranked by their frequency, the frequency of a particular observation is inversely proportional to its rank,

$$\text{Frequency} \propto \frac{1}{\text{Rank}}\,. \tag{2.1}$$

Partly because it is so unexpected, a great deal of effort has gone into explaining Zipf's law. So far, almost all explanations are either domain specific or require fine-tuning. For language, there are a variety of domain-specific models, beginning with the suggestion that Zipf's law could be explained by imposing a balance between the effort of the listener and speaker (Zipf, 1949; Cancho i and Solé, 2003; Corominas-Murtra et al., 2011). Other explanations include minimizing the number of letters (or phonemes) necessary to communicate a message (Mandelbrot, 1953), or by considering the generation of random words (Li, 1992). There are also domain-specific models for the distribution of city and firm sizes. These models propose a process in which cities or firms grow by random amounts (Gabaix, 1999; Axtell, 2001; Ioannides and Overman, 2003), with a fixed total population or wealth and a fixed minimum size. Other explanations of Zipf's law require fine tuning. For instance, there are many mechanisms that can generate power laws (Newman, 2005), and these can be fine tuned to give an exponent of -1. Possibly the most important fine-tuned proposal is the notion that some systems sit at a highly unusual thermodynamic state — a critical point (Mora and Bialek, 2011; Saremi and Sejnowski, 2013, 2014; Tkačik et al., 2014, 2015).

Only very recently has there been an explanation, by Schwab et al. (2014), that does not require fine tuning. This explanation exploits the fact that most real-world datasets have hidden structure that can be described using an unobserved variable. For such models — commonly called latent variable models — the unobserved (or latent) variable, z, is drawn from a distribution, $P(z)$, and the observation, x, is drawn from a conditional distribution, $P(x|z)$. The distribution over x is therefore given by

$$P(x) = \int dz\, P(x|z)\, P(z)\,. \tag{2.2}$$

For example, for neural data the latent variable could be the underlying firing rate or the time since stimulus onset.

While Schwab *et al.*'s result was a major advance, it came with some restrictions: the observations, x, had to be a high dimensional vector, and the conditional distribution, $P(x|z)$, had to lie in the exponential family with a small number of natural parameters. In addition, the result relied on nontrivial concepts from statistical physics, making it difficult to gain intuition into why latent variable models generally lead to Zipf's law, and, just as importantly, why they sometimes

do not. Here we use the same starting point as Schwab *et al.* (Eq. (2.2)), but take a very different theoretical approach — one that considerably extends our theoretical and empirical understanding of the relationship between latent variable models and Zipf's law. This approach not only gives additional insight into the underlying mechanism by which Zipf's law emerges, but also gives insight into where and how that mechanism breaks down. Moreover, our theoretical approach relaxes the restrictions inherent in Schwab *et al.*'s model (high dimensional observations and an exponential family distribution with a small number of natural parameters). Consequently, we are able to apply our theory to three important types of data, all of which are inaccessible under Schwab *et al.*'s model: word frequencies, models where the latent variable is the sequence length, and complex datasets with high-dimensional observations.

For word frequencies – the domain in which Zipf's law was originally discovered – we show that taking the latent variable to be the part of speech (e.g. noun/verb) can explain Zipf's law. As part of this explanation, we show that if we take only one part of speech (e.g. only nouns) then Zipf's law does not emerge – a phenomenon that is not, to our knowledge, taken into account by any other explanation of Zipf's law for words. For models in which the latent variable is sequence length (i.e. observations in which the dimension of the vector, x, is variable), we show that Zipf's law emerges under very mild conditions. Finally, for models that are high dimensional and sufficiently realistic and complex that the conditional distribution, $P(x|z)$, falls outside Schwab *et al.*'s model class, we show that Zipf's law still emerges very naturally, again under mild conditions. In addition, we introduce a quantity that allows us to assess how much a given latent variable contributes to the observation of Zipf's law in a particular dataset. This is important because it allows us to determine, quantitatively, whether a particular latent variable really does contribute significantly to Zipf's law.

2.3 Results

Under Zipf's law (Eq. (2.1)) frequency falls off relatively slowly with rank. This means, loosely, that rare observations are more common than one would typically expect. Consequently, under Zipf's law, one should observe a fairly broad range of frequencies. This is the case, for instance, for words — just look at the previous sentence: there are some very common words (e.g. "a", "of"), and other words that are many orders of magnitude rarer (e.g. "frequencies", "consequently"). This is a remarkable property: you might initially expect to see rare words only rarely. However, while a particular rare word (e.g. "frequencies") is far less likely to occur than a particular common word (e.g. "a"), there are far more rare words than common words, and these factors balance almost exactly, so that a

random word drawn from a body of text is roughly equally likely to be rare, like "frequencies" as it is to be common, like "a".

Our explanation of Zipf's law consists of two parts. The first part is the above observation — that Zipf's law implies a broad range of frequencies. This notion was quantified by Mora and Bialek, who showed that a perfectly flat distribution over a range of frequencies is mathematically equivalent to Zipf's law over that range (Mora and Bialek, 2011) — a result that applies in any and all domains. However, it is important to understand the realistic case: how a finite range of frequencies with an uneven distribution might lead to something similar to, but not exactly, Zipf's law. We therefore extend Mora and Bialek's result, and derive a general relationship that quantifies deviations from Zipf's law for arbitrary distributions over frequency — from very broad to very narrow, and even to multi-modal distributions. That relationship tells us that Zipf's law emerges when the distribution over frequency is sufficiently broad, even if it is not very flat. We complete the explanation of Zipf's law by showing that latent variables can, but do not have to, induce a broad range of frequencies. Finally, we demonstrate theoretically and empirically that, in a variety of important domains, it is indeed latent variables that give rise to a broad range of frequencies, and hence Zipf's law. In particular, we explain Zipf's law in three domains by showing that, in each of them, the existence of a latent variable leads to a broad range of frequencies. Furthermore, we demonstrate that data with both a varying number of dimensions, and fixed but high dimension, leads to Zipf's law under very mild conditions.

A broad range of frequencies implies Zipf's law

By "a broad range of frequencies", we mean the frequency varies by many orders of magnitude, as is the case, for instance, for words: "a" is indeed many orders of magnitude more common than "frequencies". It is therefore convenient to work with the energy, defined by

$$\mathcal{E}(x) \equiv -\log P(x) = -\log \text{Frequency}\,(x) + \text{const.} \tag{2.3}$$

where, as above, x is an observation, and we have switched from frequency to probability. To translate Zipf's law from observations to energy, we take the log of both sides of Eq. (2.1) and use Eq. (2.3) for the energy; this gives us

$$\text{Zipf's law holds exactly} \iff \log r(\mathcal{E}) = \mathcal{E} + \text{const.}, \tag{2.4}$$

where $r(\mathcal{E})$ is the rank of an observation whose energy is $\mathcal{E}$.

Given, as discussed above, that Zipf's law implies a broad range of frequencies, we

expect Zipf's law to hold whenever the low and high energies (which translate into high and low frequencies) have about the same probability. Indeed, previous work (Mora and Bialek, 2011) showed that when the distribution over energy, $P(\mathcal{E})$, is perfectly constant over a broad range, Zipf's law holds exactly in that range. However, in practice the distribution over energy is never perfectly constant; the real world is simply not that neat. Consequently, to understand Zipf's law in real-world data, it is necessary to understand how deviations from a perfectly flat distribution over energy affect Zipf plots. For that we need to find the exact relationship between the distribution over energy and the rank.

To find this exact relationship, we note, using an approach similar to (Mora and Bialek, 2011), that if we were to plot rank versus energy, we would see a stepwise increase at the energy of each observation, x. Consequently, the gradient of the rank is 0 almost everywhere, and a delta-function at the location of each step,

$$\frac{dr(\mathcal{E})}{d\mathcal{E}} = \sum_x \delta\big(\mathcal{E} - \mathcal{E}(x)\big). \tag{2.5}$$

The right hand side is closely related to the probability distribution over energy. That distribution can be thought of as a sum of delta-functions, each one located at the energy associated with a particular x and weighted by its probability,

$$P(\mathcal{E}) = \sum_x P(x)\,\delta\big(\mathcal{E} - \mathcal{E}(x)\big) = e^{-\mathcal{E}} \sum_x \delta\big(\mathcal{E} - \mathcal{E}(x)\big), \tag{2.6}$$

with the second equality following from Eq. (2.3). This expression says that the probability distribution over energy is proportional to $e^{-\mathcal{E}} \times$ the density of states, a standard result from statistical physics (Pathria and Beale, 2011). Comparing Eqs. (2.5) and Eq. (2.6), we see that

$$\frac{dr(\mathcal{E})}{d\mathcal{E}} = e^{\mathcal{E}} P(\mathcal{E})\,. \tag{2.7}$$

Integrating both sides from $-\infty$ to $\mathcal{E}$ and taking the logarithm gives

$$\log r(\mathcal{E}) = \mathcal{E} + \log P_S(\mathcal{E}) \tag{2.8}$$

where $P_S(\mathcal{E})$ is $P(\mathcal{E})$ smoothed with an exponential kernel,

$$P_S(\mathcal{E}) \equiv \int_{-\infty}^{\mathcal{E}} d\mathcal{E}' P\big(\mathcal{E}'\big)\, e^{\mathcal{E}'-\mathcal{E}}\,. \tag{2.9}$$

Comparing Eq. (2.8) to Eq. (2.4), we see that for Zipf's law to hold exactly over some range (i.e. $\log r(\mathcal{E}) = \mathcal{E} + \text{const.}$, or $r(\mathbf{x}) \propto 1/P(\mathbf{x})$), we need $P_S(\mathcal{E}) = \text{const.}$ over that range. This is not new; it was shown previously by Mora and Bialek using essentially the same arguments we used here (Mora and Bialek, 2011). What

is new is the exact relationship between $P(\mathcal{E})$ and $r(\mathcal{E})$ given in Eq. (2.8), which is valid whether or not Zipf's law holds exactly. This is important because the distribution over energy is never perfectly flat, so we need to reason about how deviations from $P_S(\mathcal{E}) = \text{const.}$ affect Zipf plots — something that our analysis allows us to do. In particular, Eq. (2.8) tells us that departures from Zipf's law are due solely to variations in $\log P_S(\mathcal{E})$. Consequently, Zipf's law emerges if variations in $\log P_S(\mathcal{E})$ are small compared to the range of observed energies. This requires the distribution over energy to be broad, but not necessarily very flat (see Eq. (2.22) and surrounding text for an explicit example). Much of the focus of this paper is on showing that latent variable models typically produce sufficient broadening in the distribution over energy for Zipf's law to emerge.

Narrow distributions over energy are typical

The analysis in the previous section can be used to tell us why a broad (i.e. Zipfian) distribution over energy is special, and a narrow distribution over energy is generic. Integrating Eq. (2.6) over a small range (from $\mathcal{E}$ to $\mathcal{E} + \Delta\mathcal{E}$) we see that

$$P(\mathcal{E} \text{ to } \mathcal{E} + \Delta\mathcal{E}) \approx e^{-\mathcal{E}} \mathcal{N}(\mathcal{E} \text{ to } \mathcal{E} + \Delta\mathcal{E}) \tag{2.10}$$

where $\mathcal{N}(\mathcal{E} \text{ to } \mathcal{E}+\Delta\mathcal{E})$ is the number of states with energy between $\mathcal{E}$ and $\mathcal{E}+\Delta\mathcal{E}$. As we just saw, for a broad, Zipfian distribution over energy, we require $P(\mathcal{E})$ to be nearly constant. Thus, Eq. (2.10) tells us that for Zipf's law to emerge, we must have $\mathcal{N}(\mathcal{E} \text{ to } \mathcal{E} + \Delta\mathcal{E}) \propto e^{\mathcal{E}}$ (an observation that has been made previously, but couched in terms of entropy rather than density of states (Mora and Bialek, 2011; Schwab et al., 2014; Tkačik et al., 2014, 2015)). However, there is no reason for the number of states to take this particular form, so we do not, in general, see Zipf's law. Moreover, because of the exponential term in Eq. (2.10), whenever the range of energies is large, even small imbalances between the number of states and the energy lead to highly peaked probabilities. Thus, narrow distributions over energy are generic — a standard result from statistical physics (Pathria and Beale, 2011).

The fact that broad distributions are not generic tells us that Zipf's law is not generic. However, the above analysis suggests a natural way to induce Zipf's law: stack together many narrow distributions, each with a peak at a different energy. In the following sections we expand on this idea.

Latent variables lead to a broad range of frequencies

We now demonstrate that latent variables can broaden the distribution over energy sufficiently to give Zipf's law. We begin with generic arguments showing that latent variables typically broaden the distribution over energy. We then show empirically that, in three domains of interest, this broadening leads to Zipf's law. We also show that Zipf's law emerges generically in data with varying dimensions and in latent variable models describing data with fixed, but high, dimension.

General principles

To obtain Zipf's law, we need a dataset displaying a broad range of frequencies (or energies). It is straightforward to see how latent variables might help: if the energy depends strongly on the latent variable, then mixing across many different settings of the latent variable leads to a broad range of energies. We can formalise this intuition by noting that for a latent variable model, the distribution over x is found by integrating $P(x|z)$ over the latent variable, z (Eq. (2.2)). Likewise, the distribution over energy is found by integrating $P(\mathcal{E}|z)$ over the latent variable,

$$P(\mathcal{E}) = \int dz P(\mathcal{E}|z)\, P(z)\,. \tag{2.11}$$

Therefore, mixing multiple narrow (and hence non-Zipfian) distributions, $P(\mathcal{E}|z)$, with sufficiently different means (e.g., coloured lines in Fig. 2.1A) gives rise to a broad (and hence Zipfian) distribution, $P(\mathcal{E})$ (solid black line Fig. 2.1A). This tells us something very important: "special" Zipfian distributions, with a broad range of energies, can be constructed merely by combining many "generic" non-Zipfian distributions, each with a narrow range of energies. Critically, to achieve large broadening, the mean energy, and thus the typical frequency, of an observation must depend on the latent variable; i.e. the mean of the conditional distribution, $P(\mathcal{E}|z)$, must depend on z. Taking words as an example, one setting of the latent variable should lead mainly to common (and thus low energy) words, like "a", whereas another setting of the latent variable should lead mainly to rare (and thus high energy) words, like "frequencies".

Our mechanism (mixing together many narrow distributions over energy to give a broad distribution) is one of many possible ways that Zipf's law could emerge in real datasets. It is thus important to be able to tell whether Zipf's law in a particular dataset emerges because of our mechanism, or another one. Critically, if our mechanism is operative, even though the full dataset displays Zipf's law (and hence has a broad distribution over energy), the subset of the data associated with any particular setting of the latent variable will be non-Zipfian (and hence have a narrow distribution over energy). In this case, a broad distribution over

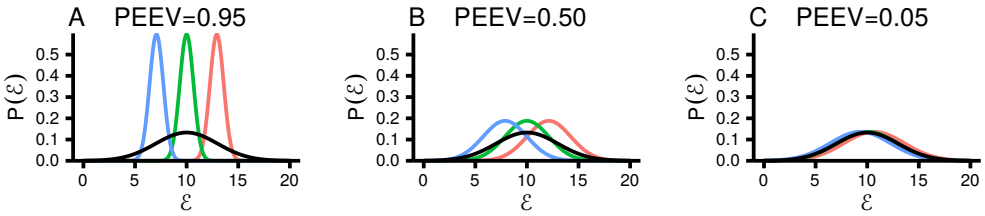

Figure 2.1: PEEV measures the average width of $P(\mathcal{E}|z)$ relative to $P(\mathcal{E})$. PEEV is close to 0 if the widths are the same, and close to 1 if $P(\mathcal{E}|z)$ is, on average, much narrower that $P(\mathcal{E})$. In all panels, the black line is $P(\mathcal{E})$, and the coloured lines are $P(\mathcal{E}|z)$ for three different settings of the latent variable, z. **A**. For high PEEV, the conditional distributions, $P(\mathcal{E}|z)$, are narrow, and have very different means. **B**. For intermediate PEEV, the conditional distributions are broader, and their means are more similar. **C**. For low PEEV, the conditional distributions are very broad, and their means are very similar.

energy, and hence Zipf's law, emerges because of the mixing of multiple narrow, non-Zipfian distributions (each with a different setting of the latent variable). To complete the explanation of Zipf's law, we only need to explain why, in that particular dataset, it is reasonable for there to be a latent variable that controls the location of the peak in the energy distribution.

Of course there is, in reality, a continuum — there are two contributions to the width of $P(\mathcal{E})$. One, corresponding to our mechanism, comes from changes in the mean of $P(\mathcal{E}|z)$ as the latent variable changes; the other comes from the width of $P(\mathcal{E}|z)$. To quantify the contribution of each mechanism towards an observation of Zipf's law, we use the standard formula for the proportion of explained variance (or R^2) to define the proportion of explained energy variance (PEEV; see Methods 2.5 for further details). PEEV gives the proportion of the total energy variance that can be explained by changes in the mean of $P(\mathcal{E}|z)$ as the latent variable, z, changes. PEEV ranges from 0, indicating that z explains none of the energy variance, so the latent variable does not contribute to the observation of Zipf's law, to 1, indicating that z explains all of the energy variance, so our mechanism is entirely responsible for the observation of Zipf's law. As an example, we plot energy distributions with a range of values for PEEV (Fig. 2.1). The black line is $P(\mathcal{E})$, and the coloured lines are $P(\mathcal{E}|z)$ for different settings of z. For high values of PEEV, the distributions $P(\mathcal{E}|z)$ are narrow, but have very different means (Fig. 2.1A). In contrast, for low values of PEEV, the distributions $P(\mathcal{E}|z)$ are broad, yet have very similar means, so the width of $P(\mathcal{E})$ comes mainly from the width of $P(\mathcal{E}|z)$ (Fig. 2.1C).

Categorical data (word frequencies)

It has been known for many decades that word frequencies obey Zipf's law (Zipf, 1932), and many explanations for this finding have been suggested (Zipf, 1949; Mandelbrot, 1953; Li, 1992; Cancho i and Solé, 2003; Corominas-Murtra et al., 2011). However, none of these explanations accounts for the observation that, while word frequencies overall display Zipf's law (solid black line, Fig. 2.2B), word frequencies for individual parts of speech (e.g. nouns vs conjunctions) do not (coloured lines, Fig. 2.2B; except perhaps for verbs, which we discuss below). We can see directly from these plots that the mechanism discussed in the previous section gives rise to Zipf's law: different parts of speech have narrow distributions over energy (coloured lines, Fig. 2.2A), and they have different means. Mixing across different parts of speech therefore gives a broad range of energies (solid black line, Fig. 2.2A), and hence Zipf's law. In practice, the fact that different parts of speech have different mean energies implies that some parts of speech (e.g. nouns, like "ream") consist of many different words, each of which is relatively rare, whereas other parts of speech (e.g. conjunctions, like "and") consist of only a few words, each of which is relatively common. We can therefore conclude that Zipf's law for words emerges because there is a latent variable, the part-of-speech, and the latent variable controls the mean energy. We can confirm quantitatively that Zipf's law arises primarily through our mechanism by noting that PEEV is relatively high, 0.58 (for details on how we compute PEEV, see Methods 2.5).

We have demonstrated that Zipf's law for words emerges because of the combination of different parts of speech with different characteristic frequencies. However, to truly explain Zipf's law for words, we have to explain why different parts of speech have such different characteristic frequencies. While this is really a task for linguists, we can speculate. One potential explanation is that different parts of speech have different functions within the sentence. For instance, words with a purely grammatical function (e.g. conjunctions, like "and") are common, because they can be used in a sentence describing anything. In contrast, words denoting something in the world (e.g. nouns, like "ream") are more rare, because they can be used only in the relatively few sentences about that object. Mixing together these two classes of words gives a broad range of frequencies, or energies, and hence, Zipf's law. Finally, using similar arguments, we can see why verbs have a broader range of frequencies than other parts of speech — some verbs (like "is") can be used in almost any context (and one might argue that they have a grammatical function) whereas other verbs (like "gather") refer to a specific type of action, and hence can only be used in a few contexts. In fact, verbs, like words in general, fall into classes (Levin, 1993).

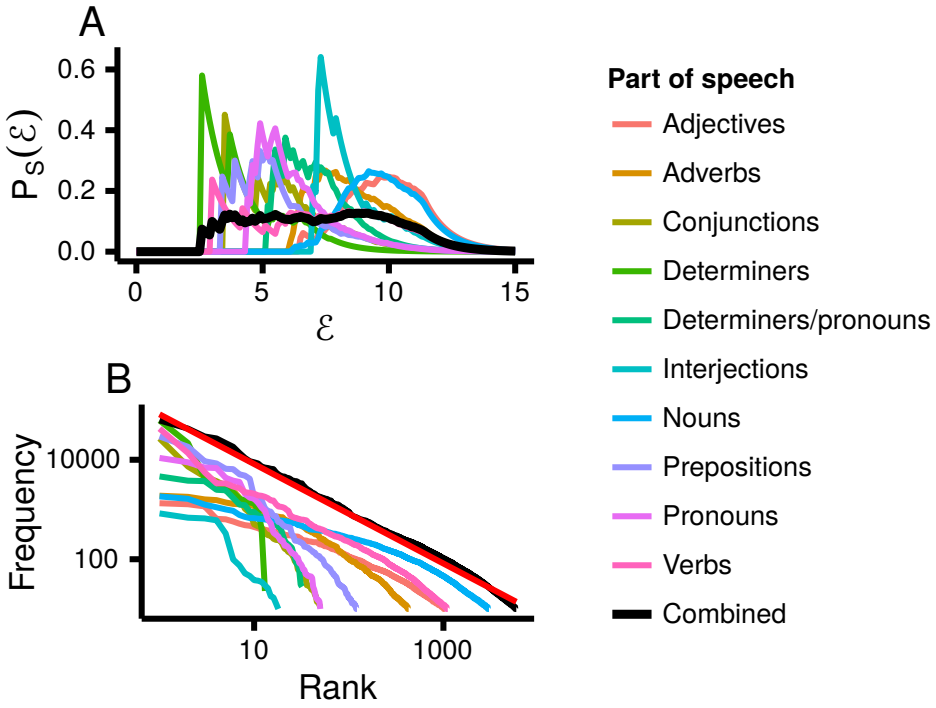

Figure 2.2: Zipf's law for word frequencies, split by part of speech (data from (Leech et al., 2001)). The coloured lines are for individual parts of speech, the black line is for all the words. **A**. The distribution over energy is broad for words in general, but the distribution over energy for individual parts of speech is narrow. **B**. Therefore, words in general obey Zipf's law, but individual parts of speech do not (except for verbs, which too can be divided into classes (Levin, 1993)). The red line has a slope of -1, and closely matches the combined data.

Data with variable dimension

Two models in which the data consists of sequences with variable length have been shown to give rise to Zipf's law (Li, 1992; Mora et al., 2010). These models fit easily into our framework, as there is a natural latent variable, the sequence length. We show that if the distribution over sequence length is sufficiently broad, Zipf's law emerges.

First, Li (1992) noted that randomly generated words with different lengths obey Zipf's law. Here "randomly generated" means the following: a word is generated by randomly selecting a symbol that can be either one of M letters or a space, all with equal probability; the symbols are concatenated; and the word is terminated when a space is encountered. We can turn this into a latent variable model by first drawing the sequence length, z, from a distribution, then choosing z letters randomly. Thus, the sequence length, z, is "latent", as it is chosen first, before the data are generated — it does not matter that in this particular case, the latent variable can be inferred perfectly from an observation.

Second, Mora et al. (2010) found that amino acid sequences in the D region of Zebrafish IgM obey Zipf's law. The latent variable is again z, the length of the amino acid sequence. The authors found that, conditioned on length, the data was well fit by an Ising-like model with translation-invariant coupling,

$$P(\mathbf{x}|z) \propto \exp\left(\sum_{i=1}^{z} h\left(x_i\right) + \sum_{i,j=1}^{z} J_{|i-j|}\left(x_i, x_j\right)\right) \tag{2.12}$$

where $\mathbf{x}$ denotes a vector, $\mathbf{x} = (x_1, x_2, ..., x_z)$, and x_i represents a single amino acid (of which there were 21).

The basic principle underlying Zipf's law in models with variable sequence length is that there are few short sequences, so each short sequence has a high probability and hence a low energy. In contrast, there are many long sequences, so each long sequence has a low probability and hence a high energy. Mixing together short and long sequences therefore gives a broad distribution over energy and hence Zipf's law.

Models in which sequence length is the latent variable are particularly easy to analyze because there is a simple relationship between the total and conditional distributions,

$$P(\mathbf{x}) = P(z|\mathbf{x})\, P(\mathbf{x}) = P(\mathbf{x}|z)\, P(z)\,. \tag{2.13}$$

The first equality holds because z, the length of the word, is a deterministic function of $\mathbf{x}$, so $P(z|\mathbf{x}) = 1$ (as long as z is the length of the vector $\mathbf{x}$, which is what we assume here); the second follows from Bayes theorem. To illustrate the

general approach, we use this to analyze Li's model (as it is relatively simple). For that model, each element of $\mathbf{x}$ is drawn from a uniform, independent distribution with M elements, so the probability of observing any particular configuration with a sequence length of z is M^{-z}. Consequently

$$P(\mathbf{x}) = M^{-z} P(z)\,. \tag{2.14}$$

Taking the log of both sides of this expression and negating gives us the energy of a particular configuration,

$$\mathcal{E}\left(\mathbf{x}\right) = z \log M - \log P(z) \approx z \log M. \tag{2.15}$$

The approximation holds because $\log P(z)$ varies little with z (in this case its variance cannot be greater than $(M+1)/M$, and in the worst case its variance is $\mathcal{O}((\log \mathrm{Var}\left[z\right])^2)$; see Methods 2.5). Therefore, the variance of the energy is approximately proportional to the variance of the sequence length, z,

$$\mathrm{Var}\left[\mathcal{E}\left(\mathbf{x}\right)\right] \approx \left(\log M\right)^2 \mathrm{Var}\left[z\right]. \tag{2.16}$$

If there is a broad range of sequence lengths (meaning the standard deviation of z is large), then the energy has a broad range, and Zipf's law emerges. More quantitatively, our analysis for high-dimensional data below suggests that in the limit of large average sequence length, Zipf's law emerges when the standard deviation of z is on the order of the average sequence length. For Li's model (Li, 1992), the standard deviation and mean of z both scale with M, so we expect Zipf's law to emerge when M is large. To check this, we simulated random words with $M = 4$. Even for this relatively modest value, $P(\mathcal{E})$ (black line, Fig. 2.3A) is relatively flat over a broad range, but the distributions for individual word lengths (coloured lines, Fig. 2.3A) are extremely narrow. Therefore, data for a single word length does not give Zipf's law (coloured lines, Fig. 2.3B), but combining across different word lengths does give Zipf's law (black line, Fig. 2.3B; though with steps, because all words with the same sequence length have the same energy).

Of course, this derivation becomes more complex for models, like the antibody data, in which elements of the sequence are not independently and identically distributed. However, even in such models the basic intuition holds: there are few short sequences, so each short sequence has high probability and low energy, whereas the opposite is true for longer sequences. In fact, the energy is still approximately proportional to sequence length, as it was in Eq. (2.15), because the number of possible configurations is exponential in the sequence length, and the energy is approximately the logarithm of that number (see Methods 2.5, for a more principled explanation). Consequently, in general a broad range of sequence lengths gives a broad distribution over energy, and hence Zipf's law.

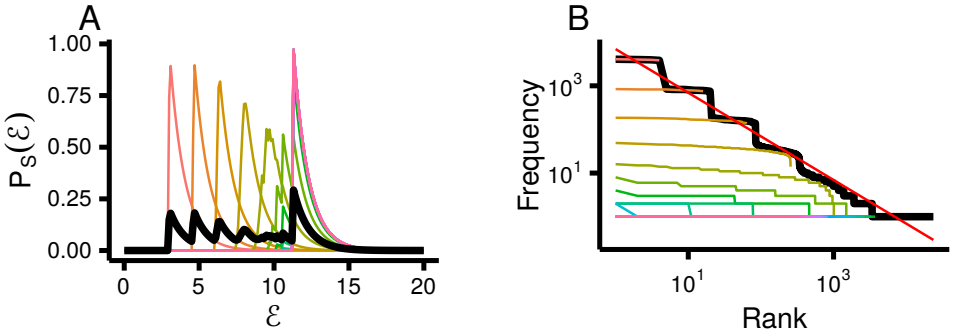

Figure 2.3: Li's model of random words displays Zipf's law because it mixes words of different lengths. **A**. The distribution over energy. **B**. Zipf plot. In both plots the black lines use all the data and each coloured line corresponds to a different word length. The red line has a slope of -1, and so corresponds to Zipf's law.

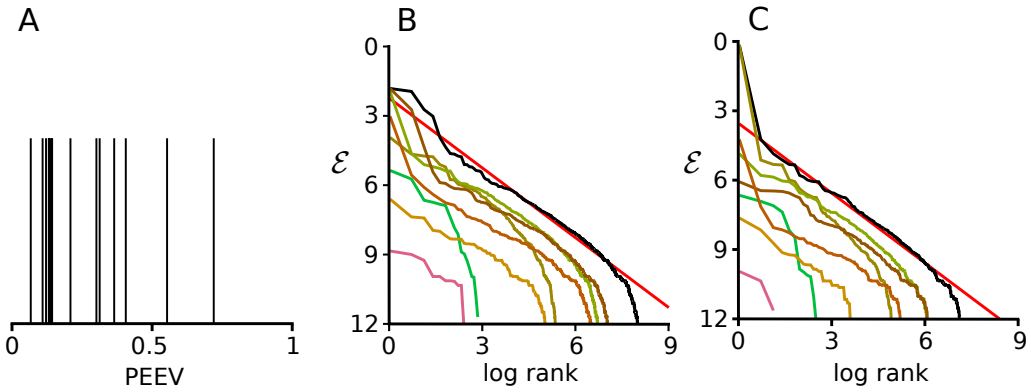

Figure 2.4: Re-analysis of amino acid sequences in the D region of 14 Zebrafish. **A**. Proportion of the variance explained by sequence length (PEEV) for the 14 datasets. Most are low, and all but two are less than 0.5. **B** and **C**. Zipf plots for the dataset with the lowest (B) and highest (C) PEEV. In both plots the black line uses all the data and the coloured lines correspond to sequence lengths ranging from 1 to 7. The red line has a slope of -1, and so corresponds to Zipf's law. Data from Ref. (Mora et al., 2010), kindly supplied by Thierry Mora. (Note that $\mathcal{E}$ increases downward on the y-axis, in keeping with standard conventions.)

However, as discussed above, just because a latent variable could give rise to Zipf's law does not mean it is entirely responsible for Zipf's law in a particular dataset. To quantify the role of sequence length in Mora *et al.*'s antibody data, we computed PEEV (the proportion of the variance of the energy explained by sequence length) for the 14 datasets used in their analysis. As can be seen in Fig. 2.4A, PEEV is generally small: less than 0.5 in 12 out of the 14 datasets. And indeed, for the dataset with the smallest PEEV (0.07), Zipf's law is obeyed at each sequence length (Fig. 2.4B). This in fact turns out to hold for all the datasets, even the one with the highest PEEV (0.72; Fig. 2.4C).

The fact that Zipf's law is observed at each sequence length complicates the interpretation of this data. Our mechanism — adding together many distributions,

each at different mean energy — plays only a small role in producing Zipf's law over the whole dataset. And indeed, an additional mechanism has been found: a recent study showed that antibody data is well modelled by random growth and decay processes (Desponds et al., 2016), which leads to Zipf's law at each sequence length.

High-dimensional data

A very important class of models are those where the data is high-dimensional. We show two things for this class. First, the distribution over energy is broadened by latent variables — more specifically, for latent variable models, the variance typically scales as n^2. Second, the n^2 scaling is sufficiently large that deviations from Zipf's law become negligible in the large n limit.

The reasoning is the same as it was above: we can obtain a broad distribution over energy by mixing together multiple, narrowly peaked (and thus non-Zipfian) distributions. Intuitively, if the peaks of those distributions cover a broad enough range of energies, Zipf's law should emerge. To quantify this intuition, we use the law of total variance (Weiss, 2006),

$$\mathrm{Var}_{\mathbf{x}}\left[\mathcal{E}(\mathbf{x})\right] = \mathrm{Var}_{z}\left[\mathrm{E}_{\mathbf{x}|z}\left[\mathcal{E}(\mathbf{x})\right]\right] + \mathrm{E}_{z}\left[\mathrm{Var}_{\mathbf{x}|z}\left[\mathcal{E}(\mathbf{x})\right]\right] \tag{2.17}$$

where again $\mathbf{x}$ is a vector, this time with n, rather than z, elements. This expression tells us that the variance of the energy (the left hand side) must be greater than the variance of the mean energy (the first term on the right hand side). (As an aside, this decomposition is the essence of PEEV; see Methods 2.5).

As discussed above, the reason latent variable models often lead to Zipf's law is that the latent variable typically has a strong effect on the mean energy (see in particular Fig. 2.1). We thus focus on the first term in Eq. (2.17), the variance of the mean energy. We show next that it is typically $\mathcal{O}(n^2)$, and that this is sufficiently broad to induce Zipf's law.

The mean energy is given by

$$\mathrm{E}_{\mathbf{x}|z}\left[\mathcal{E}(\mathbf{x})\right] = -\sum_{\mathbf{x}} P(\mathbf{x}|z)\log P(\mathbf{x})\,. \tag{2.18}$$

This is somewhat unfamiliar, but can be converted into a very standard quantity by noting that in the large n limit we may replace $P(\mathbf{x})$ with $P(\mathbf{x}|z)$, which converts the mean energy to the entropy of $P(\mathbf{x}|z)$. To see why, we write

$$\mathrm{E}_{\mathbf{x}|z}\left[\mathcal{E}(\mathbf{x})\right] = -\sum_{\mathbf{x}} P(\mathbf{x}|z)\log P(\mathbf{x}|z) + \sum_{\mathbf{x}} P(\mathbf{x}|z)\log\frac{P(\mathbf{x}|z)}{P(\mathbf{x})}. \tag{2.19}$$

For low dimensional latent variable models (more specifically, for models in which z is k dimensional with $k \ll n$), the second term on the right hand side is $\mathcal{O}(k/2 \log n)$. Loosely, that's because it's positive and its expectation over z is the mutual information between $\mathbf{x}$ and z, which is typically $\mathcal{O}(k/2 \log n)$ (Bialek et al., 2001). Here, and in almost all of our analysis, we consider low dimensional latent variables; in this regime, the second term on the right hand side is small compared to the energy, which is $\mathcal{O}(n)$ (recall, from the previous section, that the energy is proportional to the sequence length, which here is n). Thus, in the large n and small k limit — the limit of interest — the second term can be ignored, and the mean energy is approximately equal to the entropy of $P(\mathbf{x}|z)$,

$$\mathrm{E}_{\mathbf{x}|z}\left[\mathcal{E}(\mathbf{x})\right] \approx -\sum_{\mathbf{x}} P(\mathbf{x}|z) \log P(\mathbf{x}|z) \equiv H_{\mathbf{x}|z}(z). \tag{2.20}$$

Approximating the energy by the entropy is convenient because the latter is intuitive, and often easy to estimate. This approximation breaks down (as does the $\mathcal{O}(k/2 \log n)$ scaling (Bialek et al., 2001)) for high dimensional latent variables, those for which k is on the same order as n. However, the approximation is not critical to any of our arguments, so we can use our framework to show that high dimensional latent variables can also lead to Zipf's law; see Methods 2.5.

At least in the simple case in which each element of $\mathbf{x}$ is independent and identically distributed conditioned on z, it is straightforward to show that the variance of the entropy is $\mathcal{O}(n^2)$. That is because the entropy is n times the entropy of one element ($H_{\mathbf{x}|z}(z) = nH_{x_i|z}(z)$), so the variance of the total entropy is n^2 times the variance of the entropy of one element,

$$\mathrm{Var}_z\left[H_{\mathbf{x}|z}(z)\right] = n^2 \mathrm{Var}_z\left[H_{x_i|z}(z)\right], \tag{2.21}$$

which is $\mathcal{O}(n^2)$, and hence the variance of the energy is also $\mathcal{O}(n^2)$. Importantly, to obtain this scaling, all we need is that $\mathrm{Var}_z\left[H_{x_i|z}(z)\right] \sim \mathcal{O}(1)$.

In the slightly more complex case in which each element of $\mathbf{x}$ is independent, but not identically distributed conditioned on z, the total entropy is still the sum of the element-wise entropies: $H_{\mathbf{x}|z}(z) = \sum_i H_{x_i|z}(z)$. Now, though, each of the $H_{x_i|z}(z)$ can be different. In this case, for the variance to scale as n^2, the element-wise entropies must covary, with $\mathcal{O}(1)$ and, on average, positive, covariance. Intuitively, the latent variable must control the entropy, such that for some settings of the latent variable the entropy of most of the elements is high, and for other settings the entropy of most of the elements is low.

For the completely general case, in which the elements of x_i are not independent, essentially the same reasoning holds: for Zipf's law to emerge the entropies of each element (suitably defined; see Methods 2.5) must covary, with $\mathcal{O}(1)$ and, on average, positive, covariance. This result — that the variance of the energy scales

as n^2 when the elementwise entropies covary — has been confirmed empirically for multi-neuron spiking data (Tkačik et al., 2014, 2015) (though they did not assess Zipf's law).

We have shown that the variance of the energy is typically $\mathcal{O}(n^2)$. But is that broad enough to produce Zipf's law? The answer is yes, for the following reason. For Zipf's law to emerge, we need the distribution over energy to be broad over the whole range of ranks. For high-dimensional data, the number of possible observations, and hence the range of possible ranks, increases with n. In particular, the number of possible observations scales exponentially with n (e.g. if each element of the observation is binary, the number of possible observations is 2^n), so the logarithm of the number of possible observations, and hence the range of possible log-ranks, scales with n. Therefore, to obtain Zipf's law, the distribution over energy must be roughly constant over a region that scales with n. But that is exactly what latent variable models give us: the variance scales as n^2, so the width of the distribution is proportional to n, matching the range of log-ranks. Thus, the fact that the variance scales as n^2 means that Zipf's law is, very generically, likely to emerge for latent variable models in which the data is high dimensional.

We can, in fact, show that when the variance of the energy is $\mathcal{O}(n^2)$, Zipf's law is obeyed ever more closely as n increases. Rewriting Eq. (2.8), but normalizing by n, we have

$$\frac{1}{n}\log r(\mathcal{E}) = \frac{\mathcal{E}}{n} + \frac{1}{n}\log P_S(\mathcal{E})\,. \tag{2.22}$$

The normalized log-rank and normalized energy now vary across an $\mathcal{O}(1)$ range, so if $\log P_S(\mathcal{E}) \sim \mathcal{O}(1)$, the last term will be small, and Zipf's law will emerge. If the variance of the energy is $\mathcal{O}(n^2)$, then $\log P_S(\mathcal{E})$ typically has this scaling. For example, consider a Gaussian distribution, for which $\log P_S(\mathcal{E}) \sim -(\mathcal{E}-\mathcal{E}_0)^2/(2n^2)$. Because, as we have seen, the energy is proportional to n, the numerator and denominator both scale with n^2, giving $\log P_S(\mathcal{E})$ the required $\mathcal{O}(1)$ scaling. This argument is not specific to Gaussian distributions: if the variance of the energy is $\mathcal{O}(n^2)$, we expect $\log P_S(\mathcal{E})$ to display only $\mathcal{O}(1)$ changes as the energy changes by an $\mathcal{O}(n)$ amount.

This result turns out to be very robust. For instance, as we show in Methods 2.5, even delta-function spikes in the distribution over energy (Fig. 2.5A) do not disrupt the emergence of Zipf's law as n increases (Fig. 2.5B). (The distribution over energy is, of course, always a sum of delta-functions, as can be seen in Eq. (2.6). However, the delta-functions in Eq. (2.6) are typically very close together, and each one is weighted by a very small number, $e^{-\mathcal{E}}$. Here we are considering a delta-function with a large weight, as shown by the large spike in

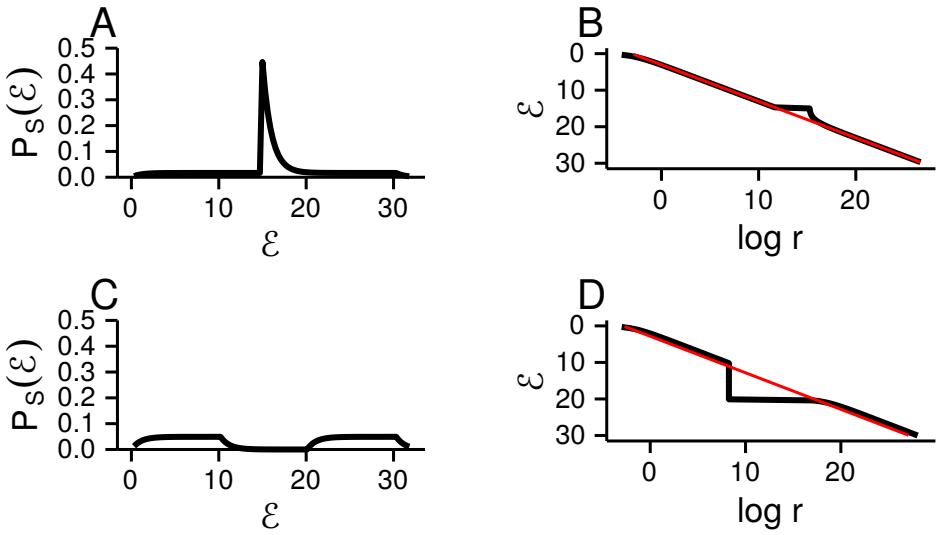

Figure 2.5: The relationship between $P(\mathcal{E})$ (left panel) and Zipf plots ($\mathcal{E}$ versus log-rank, right panel). As in Fig. 2.4, $\mathcal{E}$ increases downward on the y-axis in panels B and D. **A** and **B**. We bypassed an explicit latent variable model, and set $P(\mathcal{E}) = \text{Uniform}(\mathcal{E}; 0, 30)/2 + \delta(\mathcal{E} - 15)/2$. The deviation from Zipf's law, shown as a blip around $\mathcal{E} = 15$, is small. This is general: as we show in Methods 2.5, departures from Zipf's law scale as $1/n$ even for large delta-function perturbations. **C** and **D**. We again bypassed an explicit latent variable model, and set $P(\mathcal{E}) = \text{Uniform}(\mathcal{E}; 0, 10)/2 + \text{Uniform}(\mathcal{E}; 20, 30)/2$. The resulting hole between $\mathcal{E} = 10$ and 20 causes a large deviation from Zipf's law.

Fig. 2.5A.) However, "holes" in the probability distribution of the energy (i.e. regions of 0 probability, as in Fig. 2.5C) do disrupt the Zipf plot. That is because in regions where $P(\mathcal{E})$ is low, the energy decreases rapidly without the rank changing; this makes $\log P_S(\mathcal{E})$ very large and negative, disrupting Zipf's law (Fig. 2.5D). Between holes, however, we expect Zipf's law to be obeyed, as illustrated in Fig. 2.5D.

Importantly, we can now see why a model in which there is no latent variable, so the variance of the energy is $\mathcal{O}(n)$, does not give Zipf's law. (To see why the $\mathcal{O}(n)$ scaling of the variance is generic, see Pathria and Beale (2011)). In this case, the range of energies is $\mathcal{O}(\sqrt{n})$. This is much smaller than the $\mathcal{O}(n)$ range of the log ranks, and so Zipf's law will not emerge.

We have shown that high dimensional latent variable models lead to Zipf's law under two relatively mild conditions. First, the average entropy of each individual element of the data, $\mathbf{x}$, must covary as z changes, and the average covariance must be $\mathcal{O}(1)$ (again, see Methods 2.5, for the definition of elementwise entropy for non-independent models). Second, $P(\mathcal{E})$ cannot have holes; that is, it cannot have large regions where the probability approaches zero between regions of non-zero probability. These conditions are typically satisfied for real world data.

Neural data

Neural data has been shown, in some cases, to obey Zipf's law (Mora and Bialek, 2011; Tyrcha et al., 2013). Here the data, which consists of spike trains from n neurons, is converted to binary vectors, $\mathbf{x}(t) = (x_1(t), x_2(t), ...)$, with $x_i(t) = 1$ if neuron i spiked in timestep t and $x_i(t) = 0$ if there was no spike. The time index is then ignored, and the vectors are treated as independent draws from a probability distribution.

To model data of this type, we follow Tyrcha et al. (2013) and assume that each cell has its own probability of firing, which we denote $p_i(z)$. Here z, the latent variable, is the time since stimulus onset. This results in a model in which the distribution over each element conditioned on the latent variable is given by

$$P(x_i|z) = p_i(z)^{x_i}\big(1 - p_i(z)\big)^{1-x_i}. \tag{2.23}$$

The entropy of an individual element of $\mathbf{x}$ is, therefore,

$$H_{x_i|z}(z) = -p_i(z)\log p_i(z) - \big(1 - p_i(z)\big)\log\big(1 - p_i(z)\big). \tag{2.24}$$

The entropy is high when $p_i(z)$ is close to $1/2$, and low when $p_i(z)$ is close to 0 or 1. Because time bins are typically sufficiently small that the probability of a spike is less than $1/2$, probability and entropy are positively correlated. Thus, if the latent variable (time since stimulus onset) strongly and coherently modulates most cells' firing probabilities — with high probabilities soon after stimulus onset (giving high entropy), and low probabilities long after stimulus onset (giving low entropy) — then the changes in entropy across different cells will reinforce, giving an $\mathcal{O}(n)$ change in entropy, and thus $\mathcal{O}(n^2)$ variance.

In our data, we do indeed see that firing rates are strongly and coherently modulated by the stimulus — firing rates are high just after stimulus onset, but they fall off as time goes by (Fig. 2.6A). Thus, when we combine data across all times, we see a broad distribution over energy (black line in Fig. 2.6B), and hence Zipf's law (black line in Fig. 2.6C). However, in any one time bin the firing rates do not vary much from one presentation of the stimulus to another, and so the energy distribution is relatively narrow (coloured lines in Fig. 2.6B). Consequently, Zipf's law is not obeyed (or at least is obeyed less strongly; coloured lines in Fig. 2.6C).

In our model of the neural data, Eq. (2.23), and in the neural data itself (Methods 2.5), we assumed that the x_i were independent conditioned on the latent variable. However, the independence assumption was not critical; it was made primarily to simplify the analysis. What is critical is that there is a latent variable that controls the population averaged firing rate, such that variations in the population averaged firing rate are $\mathcal{O}(1)$ — much larger than expected for neurons that are

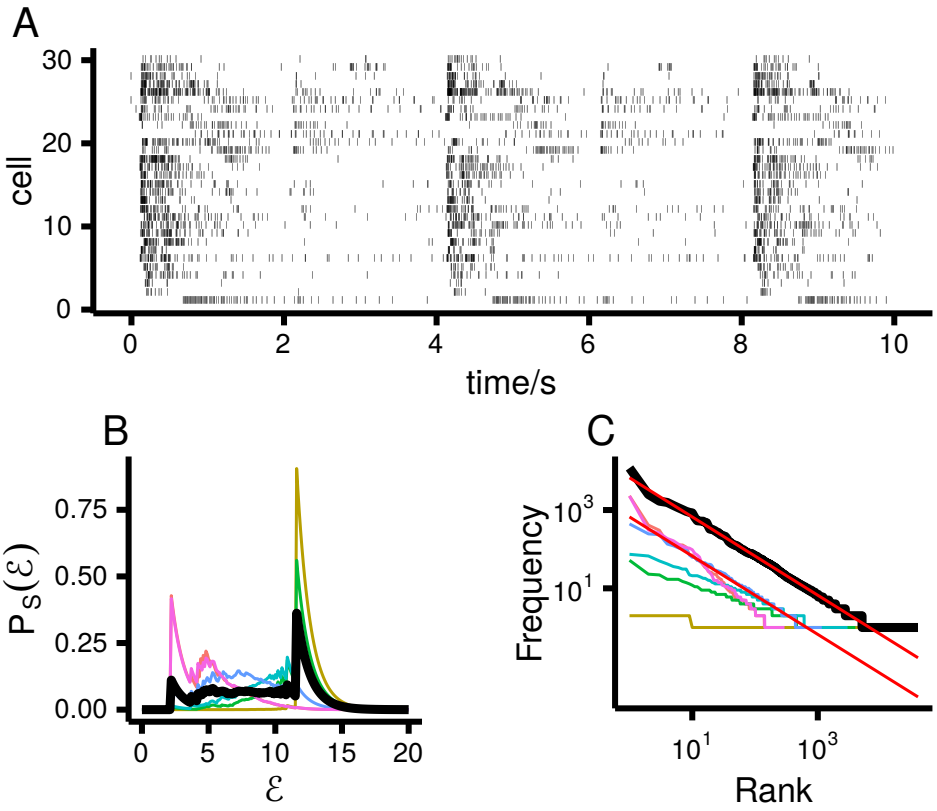

Figure 2.6: Neural data recorded from 30 mouse retinal ganglion cells stimulated by full-field illumination; see Methods 2.5, for details. **A**. Spike trains from all 30 neurons. Note that the firing rates are strongly correlated across time. **B**. $P_S(\mathcal{E}|z)$ (coloured lines) when time relative to stimulus onset is the latent variable (see text and Methods 2.5). The thick black line is $P_S(\mathcal{E})$. **C**. Zipf plots for the data conditioned on time (coloured lines) and for all the data (black line). The red lines have slope -1.

either independent or very weakly correlated. When that happens, the variance of the energy scales as n^2 (as has been observed (Tkačik et al., 2014, 2015)), and Zipf's law emerges (see Methods 2.5).

Exponential family latent variable models

Recently, Schwab et al. (2014) showed that a relatively broad class of models for high-dimensional data, a generalization of a so-called superstatistical latent variable model Beck and Cohen (2003),

$$P(\mathbf{x}|\mathbf{g}) \propto \exp\left[-n\sum_{\mu=1}^{m} g_\mu O_\mu(\mathbf{x})\right], \tag{2.25}$$

can give rise to Zipf's law. Importantly, in Schwab *et al.*'s model, when they refer to "latent variables," they are not referring to our fully general latent variables (which we call z) but to g_μ, the natural parameters of an exponential family distribution. To make this explicit, and to also make contact with our model, we rewrite Eq. (2.25) as

$$P(\mathbf{x}|z) \propto \exp\left[-n\sum_{\mu=1}^{m} g_\mu(z) O_\mu(\mathbf{x})\right] \tag{2.26}$$

where the dimensionality of z can be lower than m. (See Methods 2.5 for the link between Eqs. (2.25) and (2.26).)

If m were allowed to be arbitrarily large, Eq. (2.26) could describe any distribution $P(\mathbf{x}|z)$. However, under Schwab *et al.*'s model m can't be arbitrarily large; it must be much less than n (as we show explicitly in Methods 2.5). This puts several restrictions on Schwab *et al.*'s model class. In particular, it does not include many flexible models that have been fit to data. A simple example is our model of neural data (Eq. (2.23)). Writing this distribution in exponential family form gives

$$P(\mathbf{x}|z) \propto \exp\left[-n\sum_{\mu=1}^{n} \log\left(p_\mu(z)^{-1} - 1\right)(x_\mu/n)\right]. \tag{2.27}$$

Even though there is only one "real" latent variable, z (the time since stimulus onset), there are n natural parameters, $g_\mu = \log\left(p_\mu(z)^{-1} - 1\right)$. Consequently, this distribution falls outside of Schwab *et al.*'s model class. This is but one example; more generally, any distribution with n natural parameters $g_\mu(z)$ falls outside of Schwab *et al.*'s model class whenever the $g_\mu(z)$ have a nontrivial dependence on μ and z (as they did in Eq. (2.27)). This includes models in which sequence length is the latent variable, as these models require a large number

of natural parameters (something that is not immediately obvious; see Methods 2.5).

The restriction to a small number of natural parameters also rules out high dimensional latent variable models — models in which the number of latent variable is on the order of n. That is because such models would require at least $\mathcal{O}(n)$ natural parameters, much more than are allowed by Schwab *et al.*'s analysis. Although we have so far restricted our analysis to low dimensional latent variable models, our framework can easily handle high dimensional ones. In fact, the restriction to low dimensional latent variables was needed only to approximate the mean energy by the entropy. That approximation, however, was not necessary; we can instead reason directly: as long as changes in the latent variable (now a high dimensional vector) lead to $\mathcal{O}(n)$ changes in the mean energy — more specifically, as long as the variance of the mean energy with respect to the latent variable is $\mathcal{O}(n^2)$ — Zipf's law will emerge. Alternatively, whenever we can reduce a model with a high dimensional latent variable to a model with a low dimensional latent variable, we can use the framework we developed for low dimensional latent variables (see Methods 2.5). The same reduction cannot be carried out on Schwab *et al.*'s model, as in general that will take it out of the exponential family with a small number of natural parameters (see Methods 2.5).

Besides the restrictions associated with a small number of natural parameters, there are two further restrictions; both prevent Schwab *et al.*'s model from applying to word frequencies. First, the observations must be high-dimensional vectors. However, words have no real notion of dimension. In contrast, our theory is applicable even in cases for which there is no notion of dimension (here we are referring to the theory in earlier sections; the later sections on data with variable and high-dimension are only applicable in those cases). Second, the latent variable must be continuous, or sufficiently dense that it can be treated as continuous. However, the latent variable for words is categorical, with a fixed, small number of categories (the part-of-speech).

Finally, our analysis makes it is relatively easy to identify scenarios in which Zipf's law does not emerge, something that can be hard to do under Schwab *et al.*'s framework. Consider, for example, the following model of data consisting of n-dimensional binary vectors,

$$P(\mathbf{x}|z) \propto \exp\left[-h\sum_i x_i + A\cos z\sum_i x_i\cos\theta_i + A\sin z\sum_i x_i\sin\theta_i\right] \tag{2.28}$$

where $\theta_i \equiv 2\pi i/n$, h and A are constant, and z ranges from 0 to 2π. Although this is in Schwab *et al.*'s model class, it does not display Zipf's law. To see why,

note that it can be written

$$P(\mathbf{x}|z) \propto \prod_i \exp\left[-hx_i + A\cos\left(z-\theta_i\right)x_i\right]. \tag{2.29}$$

This is a model of place fields on a ring: the activity of neuron i is largest when its preferred orientation, θ_i, is equal to z, and smallest when its preferred orientation is $z+\pi$. Because of the high symmetry of the model, the entropy is almost independent of z. In particular, changes in z produce $\mathcal{O}(1)$ variations in the entropy (see Methods 2.5); much smaller than the $\mathcal{O}(n)$ variations needed to produce Zipf's law.

This example suggests that any model in which changes in the latent variable cause uniform translation of place fields, without changing their height or shape, should not display Zipf's law. And indeed, non-Zipfian behaviour was found in a numerical study of Gaussian place fields in one dimension (Tkačik et al., 2015). Note, though, that if the amplitude of the place fields (A in our model) or the overall firing rate (h in our model) depends on a latent variable, then the population would exhibit Zipf's law. These conclusions emerge easily from our framework, but are harder to extract from that of Schwab et al. (2014).

In conclusion, while Schwab *et al.*'s approach is extremely valuable, it does have some constraints. We were able to relax those constraints, and thus show that latent variables induce Zipf's law in a wide array of practically relevant cases (word frequencies, data with variable sequence length, and simultaneously recorded neural data). Notably, all of these lie outside the class that Schwab *et al.*'s approach can handle. In addition, our analysis allowed us to easily identify scenarios in which the latent variable model lies in Schwab *et al.*'s model class, but Zipf's law does not emerge.

2.4 Discussion

We have shown that it is possible to understand, and explain, Zipf's law in a variety of domains. Our explanation consists of two parts. First, we derived an exact relationship between the shape of a distribution over log frequencies (energies) and Zipf's law. In particular, we showed that the broader the distribution, the closer the data comes to obeying Zipf's law. This was an extension of previous work showing that if a dataset has a broad, and perfectly flat, distribution over log frequencies (e.g. if a random draw gives very common elements, like "a" and rare elements, like "frequencies" the same proportion of the time), then Zipf's law must emerge (Mora and Bialek, 2011). Importantly, our extension allowed us to reason about how deviations from a perfectly flat distribution over energy manifest in Zipf plots. Second, we showed that if there is a latent variable that controls

the typical frequency of observations, then mixing together different settings of the latent variable gives a broad range of frequencies, and hence Zipf's law. This is true even if the distributions over frequency conditioned on the latent variable are very narrow. Thus, Zipf's law can emerge when we mix together multiple non-Zipfian distributions. This is important because non-Zipfian distributions are the typical case, and are thus easy to understand.

When Zipf's law is observed, it is an empirical question whether or not it is due to our mechanism. Motivated by this observation, we derive a measure (percentage of explained variance, or PEEV) that allows us to separate out, and account for, the contribution of different latent variables to the observation of Zipf's law. We found that our mechanism was indeed operative in three domains: word frequencies, data with variable sequence length, and neural data. We were also able to show that while variable sequence length can give rise to Zipf's law on it's own, it was not the primary cause of Zipf's law in an antibody sequence dataset.

For words, the latent variable is the part of speech. As we described, parts of speech with a grammatical function (e.g. conjunctions, like "a") have a few, common words, whereas parts of speech that denote something in the world (e.g. nouns, like "frequencies") have many, rare words. Varying the latent variable therefore induces a broad range of characteristic energies (or frequencies), giving rise to Zipf's law.

For data with variable sequence length, we take the latent variable to be the sequence length itself. There are many possible long sequences, so each long sequence is rare (high-energy). In contrast, there are few possible short sequences, so each short sequence is common (low-energy). Mixing across short and long sequences, and everything in between, gives a broad range of energies, and hence Zipf's law. We examined the role of sequence length in two datasets: randomly generated words and antibody sequences, both of which display Zipf's law (Li, 1992; Mora et al., 2010). For the former, randomly generated words, sequence length was wholly responsible for Zipf's law. For the latter, antibody sequences, it formed only a small contribution. We were able to make these assessments quantitative, by computing the percentage of explained variance, or PEEV. And indeed, a recent model by Desponds *et al.* indicates that for antibodies, Zipf's law at each sequence length is most likely due to random growth and decay processes (Desponds et al., 2016).

For high-dimensional data, small changes in the energy (or entropy) of each element of the observation can reinforce to give a large change overall, and hence Zipf's law. As an example, we considered multi-neuron spiking data, for which the latent variable is the time since stimulus onset. Just after stimulus onset, the firing rate of almost every cell (and hence the energy associated with those cells), is elevated. In contrast, long after stimulus onset, the firing rate of almost every

cell (and hence the energy associated with those cells) is lower. As all the cells' energies change in the same direction (high just after stimulus onset, and low long after stimulus onset), the changes reinforce, and so produce $\mathcal{O}(n)$ changes in the total energy. Consequently, whenever the population firing rate varies with time, Zipf's law will almost always appear. This is true regardless of what is causing the variation: it could be a stimulus, or it could be low dimensional internal network dynamics. Thus, our framework is consistent with the recent observation that in salamander retina the variance of the energy scales as n^2 (the scaling needed for Zip's law to emerge), with higher variance when the stimulus induces larger covariation in the firing rates (Tkačik et al., 2014, 2015). This does not, of course, imply that the retina implements an uninteresting transformation from stimulus to neural response. However, our findings do have implications for the interpretation of observations of Zipf's law.

Our work shows that there are two types of datasets in which we expect Zipf's law to emerge generically. First, for the reason mentioned above, any dataset in which the sequence length varies (and is thus a latent variable) will display Zipf's law if the distribution over sequence length is sufficiently broad. Second, any high-dimensional dataset will display Zipf's law if the entropy of each element of the observation changes with the latent variable, and if those changes are correlated.

Previous authors have pointed out that latent variables models have interesting properties when the data is high-dimensional. As we discussed, Schwab et al. (2014) were the first to show that a relatively broad class of latent variable models describing high-dimensional data give rise to Zipf's law. Their result, however, carries some restrictions: it applies only to exponential family distributions with continuous latent variables and a small number of natural parameters. We took a far more general approach that relaxes all of these restrictions: it does not require high-dimensional data, continuous latent variables, or an exponential family distribution with a small number of latent variables. Importantly, none of the datasets that we considered lie within the class considered by Schwab et al. (2014). However, the fact that Schwab *et al.*'s analysis applies to a restricted class of models should not detract from its importance: they were the first that we know of to show that Zipf's law could arise without fine tuning.

In addition, in work that anticipated some forms of latent variable models, Macke and colleagues examined models with common input (Macke et al., 2011b), similar to the model in Eq. (2.23), as well as simple feedforward spiking neuron models (Nonnenmacher et al., 2016). They showed that both exhibit diverging heat capacity, for which the variance of the energy is $\mathcal{O}(n^2)$. Although they did not explicitly explore the connection to Zipf's law, in the latter study (Nonnenmacher et al., 2016) they noted that the diverging heat capacity should lead to Zipf's law.

These findings have important implications in fields as diverse as biology and linguistics. In biology, one explanation for Zipf's law is that biological systems sit at a special thermodynamic state, the critical point (Mora and Bialek, 2011; Saremi and Sejnowski, 2013, 2014; Tkačik et al., 2014, 2015). However, our findings indicate that Zipf's law emerges from phenomena much more familiar to biologists: unobserved states that influence the observed data. In fact, as mentioned above, for neural data our analysis shows that Zipf's law will emerge whenever the average firing rate in a population of neurons varies over time. Such time variation is common in neural systems, and can be due to external stimuli, low dimensional internal dynamics, or both.

For words, we showed that individual parts of speech do not obey Zipf's law; it is only by mixing together different parts of speech with different characteristic frequencies that Zipf's law emerges. This has an important consequence for other explanations of Zipf's law in language. In particular, the observation that individual parts of speech do not obey Zipf's law is inconsistent with any explanation of Zipf's law that fails to distinguish between parts of speech (Mandelbrot, 1953; Price, 1976; Li, 1992; Gabaix, 1999; Cancho i and Solé, 2003; Corominas-Murtra et al., 2011).

In all of these domains, the observation of Zipf's law is important because it may point to the existence of some latent variable structure. It is that structure, not Zipf's law itself, that is likely to provide insight into statistical regularities in the world.

2.5 Methods

Ethics statement

All procedures were performed under the regulation of the Institutional Animal Care and Use Committee of Weill Cornell Medical College (protocol #0807-769A) and in accordance with NIH guidelines.

Experimental methods

The neural data in Fig. 2.6 was acquired by electrophysiological recordings of 3 isolated mouse retinas, yielding 30 ganglion cells. The recordings were performed on a multielectrode array using the procedure described in (Bomash et al., 2013; Nirenberg and Pandarinath, 2012). Full field flashes were presented on a Sony LCD computer monitor, delivering intermittent flashes (2 s of light followed by 2 s of dark, repeated 30 times) of white light to the retina (Nirenberg and

Meister, 1997). All procedures were performed under the regulation of the Institutional Animal Care and Use Committee of Weill Cornell Medical College (protocol #0807-769A) and in accordance with NIH guidelines.

Spikes were binned at 20 ms, and x_i was set to 1 if cell i spiked in a bin and zero otherwise. To give us enough samples to plot Zipf's law, we estimated $p_i(z)$, the probability that neuron i spikes in bin z, from data using the model in Eq. (2.23), and drew 10^6 samples from that model. To construct the distributions of energy conditioned on the latent variable — the coloured lines in Figs. 2.6B and C — we treated samples that occurred within 100 ms as if they had the same latent variable (so, for example, $P_S\left(\mathcal{E}|z = 300\right)$ is shorthand for the smoothed distribution over energy for spike trains in the five bins between 300 and 400 ms). Finally, to reduce clutter, we plotted lines only for $z = 0$ ms, $z = 300$ ms etc.

PEEV, and the law of total variance

The law of total variance (Weiss, 2006) is well known in statistics; it decomposes the total variance into the sum of two terms. Here we briefly review this law in the context of latent variable models, and then discuss how it is related to PEEV.

The energy, $\mathcal{E}(x)$, can be trivially decomposed as

$$\mathcal{E}(x) = \mathrm{E}_{x|z}\left[\mathcal{E}(x)\right] + \left(\mathcal{E}(x) - \mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right) \tag{2.30}$$

where the first term, $\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]$, is the mean energy conditioned on z,

$$\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right] = \int \mathcal{E}(x) P(x|z)\, dx. \tag{2.31}$$

The two terms in Eq. (2.30), $\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]$ and $\left(\mathcal{E}(x) - \mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right)$, are uncorrelated, so the variance of $\mathcal{E}(x)$ is the sum of their variances,

$$\mathrm{Var}_x\left[\mathcal{E}(x)\right] = \mathrm{Var}_z\left[\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right] + \mathrm{Var}_{z,x}\left[\mathcal{E}(x) - \mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right], \tag{2.32}$$

where $\mathrm{Var}_x\left[...\right]$ is the variance with respect to $P(x)$ and $\mathrm{Var}_{x,z}\left[...\right]$ is the variance with respect to $P(x, z)$. As is straightforward to show, the second term can be rearranged to give the law of total variance,

$$\mathrm{Var}_x\left[\mathcal{E}(x)\right] = \mathrm{Var}_z\left[\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right] + \mathrm{E}_z\left[\mathrm{Var}_{x|z}\left[\mathcal{E}(x)\right]\right]. \tag{2.33}$$

This is the same as Eq. (2.17) of the main text, except here we use x rather than $\mathbf{x}$.

We can identify two contributions to the variance. The first, $\mathrm{Var}_z\left[\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right]$, is

the variance of the expected energy, $\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]$, induced by changes in the latent variable, z. This represents the contribution to the total energy variance from the latent variable (i.e. the contribution from changes in the peak of $P(\mathcal{E}|z)$ as z changes) and, under our mechanism, is the contribution that gives rise to Zipf's law. The second, $\mathrm{E}_z\left[\mathrm{Var}_{x|z}\left[\mathcal{E}(x)\right]\right]$, is the variance of the energy, $\mathrm{Var}_{x|z}\left[\mathcal{E}(x)\right]$, for a fixed setting of the latent variable, averaged over the latent variable, z. This represents the contribution from the width of $P(\mathcal{E}|z)$. The proportion of explained energy variance (PEEV) — that is, the portion explained by the first contribution — is the ratio of the first quantity to the total variance of the energy,

$$\mathrm{PEEV} \equiv \frac{\mathrm{Var}_z\left[\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right]}{\mathrm{Var}_x\left[\mathcal{E}(x)\right]}\,. \tag{2.34}$$

This quantity ranges from 0, indicating that z explains none of the energy variance, to 1, indicating that z explains all of the energy variance. PEEV therefore describes how much the latent variable contributes to the observation of Zipf's law, though it should be remembered that PEEV may be large even if the total energy variance is narrow, and hence Zipf's law is not obeyed.

Computing PEEV

To compute PEEV, we need to estimate, from data, the distribution over energy given the latent variable, and the distribution over the latent variable. Here we consider the case in which the latent variable is category, and each observation, x, falls into a single, known, category. In more realistic cases, $P(z|x)$ must be estimated from a model and $P(x)$ from data, from which $P(x|z)$ and $P(z)$ can be obtained using Bayes' theorem.

The starting point is the number of observations, and the category, of each possible value of x. For instance, for words, we took a list of words, their frequencies, and their parts of speech from Leech et al. (2001). We then used the frequencies to estimate the probability of each observation, and, finally, turned those into an energy via Eq. (2.3): $\mathcal{E}(x) = -\log P(x)$. The empirical distribution over energy, $P(\mathcal{E})$, and over energy given the latent variable, $P(\mathcal{E}|z)$, was therefore a set of delta functions, with each delta-function weighted by the probability of its corresponding observation,

$$P(\mathcal{E}) = \sum_x P(x)\,\delta\big(\mathcal{E} - \mathcal{E}(x)\big), \tag{2.35}$$

$$P(\mathcal{E}|z) = \sum_x P(x|z)\,\delta\big(\mathcal{E} - \mathcal{E}(x)\big). \tag{2.36}$$

The first equation is the same as Eq. (2.6); it is repeated here for convenience.

To compute the terms relevant to PEEV (Eq. (2.34)), we need moments of both

the total energy and the energy conditioned on z. These are given, respectively, by

$$\mathrm{E}_x\left[\mathcal{E}^k(x)\right] = \sum_x P(x)\,\mathcal{E}^k(x), \tag{2.37}$$

$$\mathrm{E}_{x|z}\left[\mathcal{E}^k(x)\right] = \sum_x P(x|z)\,\mathcal{E}^k(x). \tag{2.38}$$

Then, to compute the variances required for PEEV, we use

$$\mathrm{Var}_{x|z}\left[\mathcal{E}(x)\right] = \mathrm{E}_{x|z}\left[\mathcal{E}^2(x)\right] - \left(\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right)^2, \tag{2.39}$$

$$\mathrm{Var}_z\left[\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right] = \mathrm{E}_z\left[\left(\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right)^2\right] - \left(\mathrm{E}_z\left[\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right]\right)^2, \tag{2.40}$$

where

$$\mathrm{E}_z\left[\mathrm{E}_{x|z}\left[\mathcal{E}^k(x)\right]\right] = \mathrm{E}_x\left[\mathcal{E}^k(x)\right], \tag{2.41}$$

$$\mathrm{E}_z\left[\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]^k\right] = \sum_z P(z)\left(\mathrm{E}_{x|z}\left[\mathcal{E}(x)\right]\right)^k. \tag{2.42}$$

$\mathrm{Var}\left[\log P(z)\right]$ is $\mathcal{O}\left((\log \mathrm{Var}\left[z\right])^2\right)$

To compute the variance of the energy for variable length data, we stated that the variance of $\log P(z)$ is small compared to the variance of z (see in particular Eq. (2.15)). Here we first show that for Li's model Li (1992), the variance of $\log P(z)$ is $\mathcal{O}(1)$; we then show that in general the variance of $\log P(z)$ is at most $\mathcal{O}\left((\log \mathrm{Var}\left[z\right])^2\right)$.

For Li's model, the probability of observing a sequence of length z is proportional to the probability of drawing z letters followed by a blank. For an alphabet with M letters, this is given by

$$P(z) = \frac{1}{M}\left(\frac{M}{M+1}\right)^z. \tag{2.43}$$

The leading factor of $1/M$ ensures that the distribution is properly normalized (note that z ranges from 1 to ∞). Given this distribution, it is straightforward to show that

$$\mathrm{Var}_z\left[\log P(z)\right] = M(M+1)\left(\log\left[1+\frac{1}{M}\right]\right)^2. \tag{2.44}$$

Using the fact that $\log(1+\epsilon) \leq \epsilon$, we see that the right hand side is bounded by $(M+1)/M$. Thus, for Li's model, $\mathrm{Var}_z\left[\log P(z)\right]$ is indeed $\mathcal{O}(1)$.

To understand how the variance of $\log P(z)$ scales in general, we note that the

variance is bounded by the second moment,

$$\mathrm{Var}_z\left[\log P(z)\right] = \sum_z P(z)\left[\log P(z)\right]^2 - \left(\sum_z P(z)\log P(z)\right)^2 \le \sum_z P(z)\left[\log P(z)\right]^2. \tag{2.45}$$

Shortly we'll maximize the second moment with the variance of z fixed. When we do that, we find that the second moment is small compared to σ_z^2, the variance of z. However, the analysis is somewhat complicated, so first we provide the intuition.

The main idea is to note that for unimodal distributions, the number of sequence lengths with appreciable probability is proportional to the standard deviation of z. If we make the (rather crude) approximation that $P(z)$ is nonzero only for n_0 sequence lengths, where $n_0 \propto \sigma_z$, then the right hand side of Eq. (2.45) is maximum when $P(z) = 1/n_0$, and the corresponding value is $(\log n_0)^2$. Consequently, the second moment of $\log P(z)$ is at most $\mathcal{O}\big((\log \sigma_z)^2\big)$, giving us the very approximate bound

$$\mathrm{Var}_z\left[\log P(z)\right] \le \mathcal{O}\left(\frac{\log \sigma_z^2}{2}\right)^2 \tag{2.46}$$

where we used $\log \sigma_z = (1/2)\log \sigma_z^2$.

This does indeed turn out to be the correct bound. To show that rigorously, we take the usual approach: we use Lagrange multipliers to maximize the second moment of $\log P(z)$ with constraints on the total probability and the variance. This gives us

$$0 = \frac{\partial}{\partial P(z)}\Big[\sum_{z'} P\left(z'\right)\left(\log P\left(z'\right)\right)^2 - (\gamma^2+\alpha^2-1)\left(\sum_{z'} P\left(z'\right) - 1\right) - \frac{\gamma^2 Z^2}{e^2}\left(\sum_{z'} P\left(z'\right) z'^2 - \mu^2 - \sigma_z^2\right)\Big] \tag{2.47}$$

where μ is the mean value of z,

$$\mu \equiv \sum_z P(z)\, z. \tag{2.48}$$

We use $\gamma^2 + \alpha^2 - 1$ and $\gamma^2 Z^2/e^2$ as our Lagrange multiplier to simplify later expressions. As is straightforward to show (taking into account the fact that μ depends on $P(z)$), Eq. (2.47) is satisfied when $P(z)$ is given by

$$P(z) = \exp\left[-1 - \left(\gamma^2 + \alpha^2 - \frac{\gamma^2 Z^2 \mu^2}{e^2} + \frac{\gamma^2 Z^2 (z-\mu)^2}{e^2}\right)^{1/2}\right]. \tag{2.49}$$

The parameters γ, α and Z must be chosen so that $P(z)$ is normalized to 1 and has variance σ_z^2. However, because z is a positive integer, finding these parameters analytically is, as far as we know, not possible. We can, though, make two approximations that ultimately do yield analytic expressions. The first is to allow z to be continuous. This turns sums (which are needed to compute moments) into integrals, and results in an error in those sums that scales as $1/\sigma_z$. That error is negligible in the limit that σ_z is large (the limit of interest here). The second is to allow z to be negative. This will increase the maximum second moment of $\log P(z)$ at fixed σ_z^2 (because we are expanding the space of probability distributions), and so result in a slightly looser bound. But the bound will be sufficiently tight for our purposes.

The problem of choosing the parameters γ, α and Z is now much simpler, as we can do integrals rather than sums. We proceed in three steps: first, we show that none of the relevant moments depend on μ, so we set it to zero and at the same time eliminate α; second, we use the fact that $P(z)$ must be properly normalized to express Z in terms of γ; and third, we explicitly compute the second moment of $\log P(z)$ and the variance of σ^2.

To see that the second moment of $P(z)$ and the variance of z do not depend on μ, make the change of variables $z = z' + \mu$ and let $\alpha^2 = \gamma^2 Z^2 \mu^2 / e^2$. That yields a distribution $P(z')$ that is independent of μ. Thus, μ does not effect either the second moment of $\log P(z)$ or the variance of z, and so without loss of generality we can set both μ and α to zero. We thus have

$$P(z) = \exp\left[-1 - \gamma\left(1 + Z^2 z^2/e^2\right)^{1/2}\right]. \tag{2.50}$$

It is convenient to make the change of variables $z = ye/Z$, yielding

$$P(y) = \frac{e^{-\gamma(1+y^2)^{1/2}}}{Z(\gamma)} \tag{2.51}$$

where Z, which now depends on γ to ensure that $P(z)$ (and thus $P(y)$) is properly normalized, is given by

$$Z(\gamma) = \int dy\, e^{-\gamma(1+y^2)^{1/2}}. \tag{2.52}$$

In terms of $P(y)$, the two quantities of interest are

$$\mathrm{E}_z\left[(\log P(z))^2\right] = \mathrm{E}_y\left[\left(1 + \gamma(1+y^2)^{1/2}\right)^2\right] \tag{2.53}$$

$$\sigma_z^2 = \frac{e^2}{Z^2(\gamma)} \mathrm{E}_y\left[y^2\right]. \tag{2.54}$$

These expectations can be expressed as modified Bessel functions of the second

kind (as can be seen by making the change of variables $y = \sinh\theta$). However, the resulting expressions are not very useful, so instead, we consider two easy limits: large and small γ. In the large γ limit, $P(y)$ is Gaussian, yielding

$$\lim_{\gamma\to\infty} \mathrm{E}_z\left[(\log P(z))^2\right] = (\gamma + 3/2)^2 + \mathcal{O}(1) \tag{2.55}$$

$$\lim_{\gamma\to\infty} \sigma_z^2 = \frac{e^{2(\gamma+1)}}{2\pi}\big(1 + \mathcal{O}(1/\gamma)\big). \tag{2.56}$$

And in the small γ limit, $P(y)$ is Laplacian, and we have

$$\lim_{\gamma\to 0} \mathrm{E}_z\left[(\log P(z))^2\right] = 5 + \mathcal{O}(\gamma) \tag{2.57}$$

$$\lim_{\gamma\to 0} \sigma_z^2 = \frac{e^2}{2} + \mathcal{O}(\gamma). \tag{2.58}$$

As is straightforward to show, in both limits the second moment of $\log P(z)$ obeys the inequality

$$\mathrm{E}_z\left[(\log P(z))^2\right] \le \left(c_0 + \frac{\log\sigma_z^2}{2}\right)^2 \tag{2.59}$$

where

$$c_0 = \frac{\sqrt{20} - \log(e^2/2)}{2} \approx 1.58. \tag{2.60}$$

We verified numerically that the inequality in Eq. (2.59) is satisfied over the whole range of γ, from 0 to ∞. Thus, although very naive arguments were used to derive the bound given in Eq. (2.46), it is substantially correct.

Models in which the latent variable is the sequence length

For models in which the sequence length is the latent variable, for Zipf's law to hold the energy must be proportional to the sequence length, z; that is, the energy must be $\mathcal{O}(z)$. To determine whether this scaling holds, we start with Eq. (2.13) of the main text, which tells us that when the latent variable is sequence length, the total distribution is a simple function of the latent variables: $P(\mathbf{x}) = P(\mathbf{x}|z)\,P(z)$ where z is the dimension of $\mathbf{x}$ (the sequence length). Thus, the energy is given by

$$\mathcal{E}(\mathbf{x}) = \sum_{i=1}^{z} \mathcal{E}_i(\mathbf{x}) - \log P(z)\,. \tag{2.61}$$

where

$$\mathcal{E}_i(\mathbf{x}) \equiv -\log P(x_i|x_{i-1}\ldots x_1)\,. \tag{2.62}$$

Assuming the value of x_i isn't perfectly determined by the values of $x_1, ..., x_{i-1}$ (the typical case), each term in the sum over z is $\mathcal{O}(1)$, and so the first term in Eq. (2.61) is $\mathcal{O}(z)$. As we saw in the previous section, the variance of $\log P(z)$ is small compared to the variance of z. Consequently, the energy is $\mathcal{O}(z)$.

Latent variable models with high dimensional non-conditionally independent data

In the main text we argued that for a conditionally independent model — a model in which each element of $\mathbf{x}$ is independent conditioned on z — the variance of the entropy typically scales as n^2. Extending this argument to complex joint distribution is straightforward, and, in fact, follows closely the method used in the previous section.

The first step is to note that, just as in the conditionally independent case, $\log P(\mathbf{x}|z)$ can be written as a sum over each element of x_i,

$$\log P(\mathbf{x}|z) = \sum_i \log P(x_i|z, x_1, x_2, ..., x_{i-1}) \,. \tag{2.63}$$

Taking the expectation with respect to $P(\mathbf{x}|z)$ (and negating) gives the entropy, which consists of a sum of n terms,

$$H_{\mathbf{x}|z}(z) = \sum_{i=1}^{n} h_i(z) \tag{2.64}$$

where $h_i(z)$ is the entropy of $P(x_i|z, x_1, x_2, ..., x_{i-1})$, averaged over x_1 to x_{i-1}, with z fixed,

$$h_i(z) \equiv \mathrm{E}_{\mathbf{x}|z} \left[-\log P(x_i|z, x_1, x_2, ..., x_{i-1}) \right] . \tag{2.65}$$

The variance of the entropy is thus given by

$$\mathrm{Var}_z \left[H_{\mathbf{x}|z}(z) \right] = \sum_{ij} \mathrm{Cov}_z \left[h_i(z), h_j(z) \right] . \tag{2.66}$$

Just as in the main text, if the individual entropies (the h_i) have, on average, $\mathcal{O}(1)$ covariance as z changes, then the variance of the entropy is $\mathcal{O}(n^2)$. This illuminates a special case in which we do not see Zipf's law: if the $x_1, x_2, ..., x_{i-1}$ determine the value of x_i when $i > i_0$ (independent of n), then the entropy, h_i, is zero whenever $i > i_0$. If this were to happen, the variance of the entropy would scale at most as i_0^2, independent of n; far smaller than the required $\mathcal{O}(n^2)$ scaling. However, for most types of data, including neural data, each neuron has considerable independent noise (due, for instance, to synaptic failures (Branco and Staras, 2009)), so the h_i typically remain finite for all i.

For complex joint distribution, the $h_i(z)$ can be hard to reason about and/or compute. However, here we argue that it is possible to reason about the scaling of the covariance of the $h_i(z)$'s based on the scaling of the covariance of the elementwise entropies $H_{x_i|z}(z)$, which are much simpler quantities. To see this, note that the h_i can be written

$$h_i(z) = H_{x_i|z}(z) - I_i(z) \tag{2.67}$$

where, as in the main text, the first term is the elementwise entropy,

$$H_{x_i|z}(z) \equiv -\sum_{x_i} P(x_i|z) \log P(x_i|z)\,, \tag{2.68}$$

and the second term is the mutual information between x_i and x_1 to x_{i-1}, conditioned on z,

$$\begin{aligned} I_i(z) &\equiv \mathrm{E}_{\mathbf{x}|z}\left[-\log\left(\frac{P(x_i|z)}{P(x_i|z,x_1,x_2,...,x_{i-1})}\right)\right] \\ &= H_{x_i|z}(z) + \mathrm{E}_{\mathbf{x}|z}\left[\log P(x_i|z,x_1,x_2,...,x_{i-1})\right]. \end{aligned} \tag{2.69}$$

Combining Eq. (2.67) with Eq. (2.64), we see that

$$\begin{aligned} \mathrm{Var}_z\left[H_{\mathbf{x}|z}(z)\right] = &\sum_{ij} \mathrm{Cov}_z\left[H_{x_i|z}(z), H_{x_j|z}(z)\right] \\ &- \sum_{ij} 2\mathrm{Cov}_z\left[H_{x_i|z}(z), I_j(z)\right] \\ &+ \sum_{ij} \mathrm{Cov}_z\left[I_i(z), I_j(z)\right]. \end{aligned} \tag{2.70}$$

If the $H_{x_i|z}(z)$ covary, then the first term is $\mathcal{O}(n^2)$. In this situation it would require very precise cancellation for the whole expression to be $\mathcal{O}(n)$. Such cancellation could occur if, for instance, $H_{x_i|z}(z) = I_i(z) + \text{const.}$. However, unless the constant were zero, so $x_{i-1}...x_1$ determine the value of x_i (see Eq. (2.69)), it is unclear how this could occur. Thus, as claimed in the main text, except in cases in which there is highly precise cancellation, if the elementwise entropies $H_{x_i|z}(z)$ covary (with $\mathcal{O}(1)$ covariance), the variance of the total entropy will scale as n^2.

High dimensional latent variables

So far we have restricted our analysis to low dimensional latent variables. However, this is not absolutely necessary, and in fact high dimensional latent variable can induce Zipf's law the same way low dimensional ones can: if different settings of the latent variable result in $\mathcal{O}(n)$ differences in the mean energy, Zipf's law will emerge. The main difference in the analysis is that we can no longer approximate

the mean energy by the entropy, as we did in Eq. (2.20). However, it is not actually necessary to make this approximation; it is merely convenient, as it allows us to work with the entropy, an intuitive, well-understood quantity. Indeed, if we work directly with the mean energy, Eq. (2.18), we can see that covariation in the individual energies leads to Zipf's law — just as the covariation in the individual entropies led to Zipf's law in the previous section.

To show this explicitly, we break Eq. (2.18) into one term for each element of $\mathbf{x}$,

$$\mathrm{E}_{\mathbf{x}|z}\left[-\log P(\mathbf{x})\right] = \sum_{\mathbf{x}} l_i(\mathbf{x}) \tag{2.71}$$

where

$$l_i(\mathbf{x}) \equiv \mathrm{E}_{\mathbf{x}|z}\left[-\log P(x_i|x_1, x_2, ..., x_{i-1})\right]. \tag{2.72}$$

Then, writing the variance of the mean energy in terms of the l_i, we have

$$\mathrm{Var}_z\left[\mathrm{E}_{\mathbf{x}|z}\left[\log P(\mathbf{x})\right]\right] = \sum_{ij} \mathrm{Cov}_z\left[l_i, l_j\right]. \tag{2.73}$$

If the l_i have $\mathcal{O}(1)$, and positive, covariance, the variance of the energy is $\mathcal{O}(n^2)$, and Zipf's law emerges. The intuition is that each element of $\mathbf{x}$ contributes to the energy, $-\log P(\mathbf{x})$. These contributions (or their expected values) change with the latent variable, and if they all change in the same direction, then the overall change in the energy is $\mathcal{O}(n)$, so the variance is $\mathcal{O}(n^2)$.

While the above analysis provides the underlying intuition, in practical situations the l_i may be difficult to compute. We therefore provide an alternative approach. For definiteness, we'll set the dimension of the latent variable to the dimension of the data, n; to make this explicit, we'll replace z by $\mathbf{z}$ ($\equiv z_1, z_2, ..., z_n$). In addition, we'll assume, without loss of generality, that each latent variable — each z_i — has an $\mathcal{O}(1)$ range. We'll also assume that each latent variable has an $\mathcal{O}(1)$ effect on the mean energy; this ensures that the average energy has sensible scaling with n.

Because each of the latent variables has a small effect, they need to act together to produce the $\mathcal{O}(n)$ variability in the mean energy that is required for Zipf's law. Specifically, if any two latent variables, say z_i and z_j, have the same effect on the average energy (either both increasing it or both decreasing it), they need to be positively correlated; if they have the opposite effect (one increasing it and the other decreasing it), they need to be negatively correlated. When this doesn't hold — when correlations are essentially arbitrary, or non-existent — variations in $\mathbf{z}$ have an $\mathcal{O}(\sqrt{n})$ effect on the average energy. In this regime, the variance of the average energy is $\mathcal{O}(n)$, and Zipf's law does not emerge. We thus conclude, at

least tentatively (and perhaps not surprisingly) that the z_i must to be correlated for Zipf's law to emerge.

To see this more quantitatively, we make a first-order Taylor series expansion of the expected energy,

$$\mathrm{E}_{\mathbf{x}|\mathbf{z}}\left[\mathcal{E}(\mathbf{x})\right] \approx \mathrm{E}_{\mathbf{x}|\mathbf{z}=\boldsymbol{\mu}}\left[\mathcal{E}(\mathbf{x})\right] + \sum_{i=1}^{n}(z_i - \mu_i)\left.\frac{\partial \mathrm{E}_{\mathbf{x}|\mathbf{z}}\left[\mathcal{E}(\mathbf{x})\right]}{\partial z_i}\right|_{\mathbf{z}=\boldsymbol{\mu}}. \tag{2.74}$$

Because each of the z_i has an $\mathcal{O}(1)$ range and an $\mathcal{O}(1)$ effect on the mean energy, each term in the sum is $\mathcal{O}(1)$. Thus, if the higher order terms in Eq. (2.74) can be neglected, the z_i have to be correlated for the variance of the average energy to scale as $\mathcal{O}(n^2)$; if they are not correlated, the variance is $\mathcal{O}(n)$.

Of course, ignoring higher order terms in high dimensions is dangerous, as the number of terms grows rapidly with n (the number of k^{th} order terms is proportional to n^k). However, it turns out to give the right intuition: the Efron-Stein inequality (Efron and Stein, 1981; Steele, 1986; Boucheron et al., 2013), along with the assumption that each latent variable has an $\mathcal{O}(1)$ effect on the energy, ensures that if the z_i are independent, the variance of the energy is indeed $\mathcal{O}(n)$. Thus, a necessary condition for Zipf's law to emerge is that the z_i are correlated, as has been pointed out previously (Tkačik et al., 2015) (in Supporting Information).

The fact that correlations are necessary to produce Zipf's law provides a natural approach to understanding models with high dimensional latent variables. The approach relies on the observation that sufficiently correlated variables have a "long" direction — a direction along which the typical size of $|\mathbf{z}|$ is $\mathcal{O}(n)$ (rather than $\mathcal{O}(\sqrt{n})$, as it is for uncorrelated latent variables). We can, therefore, construct a low dimensional latent variable that measures distance along that direction, and then use the analysis developed above for low dimensional latent variables.

Here we illustrate this idea for binary variables, $x_i = 0$ or 1. For definiteness, and because it makes the ideas more intuitively accessible, we consider a concrete setting: neural data, with as many latent variables as neurons. As in the main text, $x_i = 1$ corresponds to one or more spikes in a small time bin and $x_i = 0$ corresponds to no spikes. Because the long direction in latent variable space depends on the distribution $P(\mathbf{z})$, it would seem difficult to make general statements. However, in this example the data comes from neural spike trains, and so we can make use of the fact that firing rates of neurons often covary. Thus, a very natural low dimensional latent variable, which we denote ν, is the

population averaged firing rate,

$$\nu = \frac{1}{n} \sum_i p_i(\mathbf{z}) \tag{2.75}$$

where $p_i(\mathbf{z})$ is the probability that $x_i = 1$ given $\mathbf{z}$,

$$p_i(\mathbf{z}) = \mathrm{E}_{\mathbf{x}|\mathbf{z}}\left[x_i\right] = \sum_{\mathbf{x}} x_i P(\mathbf{x}|\mathbf{z})\,. \tag{2.76}$$

For this model the element-wise entropies have a very simple form,

$$H_{x_i|\mathbf{z}}(\mathbf{z}) = -p_i(\mathbf{z}) \log p_i(\mathbf{z}) - \big(1 - p_i(\mathbf{z})\big) \log \big(1 - p_i(\mathbf{z})\big). \tag{2.77}$$

We'll assume that all the $p_i(\mathbf{z})$ are less than $1/2$, something that is satisfied for realistic spike trains if the time bins aren't too large. Consequently, increasing $p_i(\mathbf{z})$ increases the element-wise entropy of neuron i.

We need two conditions for Zipf's law to emerge: the variance of ν must be $\mathcal{O}(1)$, and $\mathcal{O}(1)$ changes in ν must lead to $\mathcal{O}(1)$, and positively correlated, changes in the element-wise entropies (assuming, as discussed in the previous section, there isn't very precise cancellation). So long as the firing rates go up and down together, both conditions are satisfied, and Zipf's law emerges. If, on the other hand, the firing rates are not positively correlated on average, the variance of ν is $\mathcal{O}(1/\sqrt{n})$, and the population averaged firing rate provides no information about Zipf's law. This is an important example, as the population averaged firing rate is easy to estimate from data.

In summary, high dimensional latent variables are, from a conceptual point of view, no different than low dimensional ones: both lead to Zipf's law if different settings of the latent variables lead to average energies that differ by $\mathcal{O}(n)$. However, in the high dimensional case, each latent variable has a small effect on the energy, so a necessary condition for Zipf's law to emerge is that the latent variables are correlated. This turns out to be helpful: the correlations can lead naturally to a low dimensional latent variable, for which our analysis of low dimensional latent variables applies.

Peaks in $P(\mathcal{E})$ do not disrupt Zipf's law

In the main text, we noted that while holes in the distribution over energy, $P(\mathcal{E})$, disrupt Zipf's law, peaks in this distribution do not. To see this explicitly, take an extreme case: $P(\mathcal{E})$ is composed of a delta function at $\mathcal{E} = \mathcal{E}_0$, weighted by α, combined with a smooth component, $f(\mathcal{E})$, that integrates to $1 - \alpha$. Here α may be any number between 0 and 1, and in particular it need not be exponentially small in the energy, as it is in Eq. (2.6). For this case, we can compute $P_S\left(\mathcal{E}\right)$

explicitly using Eq. (2.9),

$$\frac{1}{n}\log P_S\left(\mathcal{E}\right) = \frac{1}{n}\log\left[\alpha e^{-(\mathcal{E}-\mathcal{E}_0)}\Theta\left(\mathcal{E}-\mathcal{E}_0\right) + f_S\left(\mathcal{E}\right)\right] \tag{2.78}$$

where f_S is f smoothed by an exponential kernel, Θ is the Heaviside step function, and we have normalized by n to give us the quantity relevant for determining the size of departures from Zipf's law (see Eq. (2.22)). The term $e^{-(\mathcal{E}-\mathcal{E}_0)}\Theta\left(\mathcal{E}-\mathcal{E}_0\right)$ ranges from 0 to 1, so $\log P_S\left(\mathcal{E}\right)$ can be bounded above and below,

$$\frac{1}{n}\log\left(f_s(\mathcal{E})\right) \leq \frac{1}{n}\log P_S\left(\mathcal{E}\right) \leq \frac{1}{n}\log\left(\alpha + f_s(\mathcal{E})\right)\ . \tag{2.79}$$

Assuming the distribution $f_s(\mathcal{E})$ is such that the first term vanishes in the large n limit (so that without the delta function Zipf's law would hold), then the last term must also vanish in the large n limit. Thus, even delta-function singularities do not prevent convergence to Zipf's law, so long as they occur on top of a finite baseline.

Exponential family latent variable models: technical details

Schwab et al. (2014) showed that Zipf's law emerges for a model in which the distribution over $\mathbf{x}$ given the latent variable is in the exponential family. By itself, the fact that the distribution is in the exponential family places no restrictions on the class of models. However, their derivation required other conditions to be satisfied, and those conditions do induce restrictions. In particular, their analysis does not apply to models with a large number of natural parameters (it thus does not apply when the latent variable is high dimensional), models in which the latent variable is discrete, and models in which the latent variable is the dimension of the data. Here we show this explicitly.

The relationship between Schwab *et al.*'s model and our model

Schwab et al. (2014) formulated their model as a latent variable model conditioned on natural parameters, as written in the main text, Eq. (2.25). Hidden in Eq. (2.25) is the fact that the g_μ can be "tied": the parameters g_μ are drawn from a distribution that allows delta-functions, such as $\delta(g_1 - f(g_2))$ for some function f, or even $\delta(g_3 - g_3^*)$. To make this explicit, and to also make contact with our model, we rewrote Eq. (2.25) as a latent variable model conditioned on z (Eq. (2.26)), where z is a k-dimensional latent variable. Under this model it is easy to tie variables; for instance, letting $g_1 = z$ and $g_2 = f(z)$ (with z one-dimensional) enforces the constraint $\delta(g_1 - f(g_2))$.

Number of latent variables

Here we show that the number of natural parameters (m in Eqs. (2.25) and (2.26)) must be small compared to the dimension of the data, n. We start by sketching Schwab et al. (2014) derivation, including many steps that were left to the reader in their paper. Their starting point is the expression for the energy of an observation,

$$-\log P(\mathbf{x}) = -\log \int dz\, P(z)\, e^{-n\mathbf{g}(z)\cdot\mathbf{O}(\mathbf{x})-\log Z(z)}. \tag{2.80}$$

We have written the right hand side using the form given in Eq. (2.26), except that we explicitly include the partition function (Eq. (2.82) below), and we use dot products instead of sums. This integral is evaluated using the saddle-point method,

$$-\log P(\mathbf{x}) \approx n\mathbf{g}(z^*)\cdot\mathbf{O}(\mathbf{x}) + \log Z(z^*). \tag{2.81}$$

where z^* maximizes the integrand. For the saddle point method to work — that is, for the above approximation to hold — the number of latent variables, $\dim(z)$, must be subextensive in n (i.e., $\dim(z)/n \to 0$ as n goes to infinity; see (Shun and McCullagh, 1995) for details).

The condition $\dim(z) \ll n$ does not place any restrictions on the number of natural parameters (the dimension of $\mathbf{g}$). But the next step in their derivation, computing the partition function (which is necessary for finding the energy of an observation), does. The log of the partition function is given by the usual expression,

$$\log Z(z) = \log \sum_{\mathbf{x}} e^{-n\mathbf{g}(z)\cdot\mathbf{O}(\mathbf{x})}. \tag{2.82}$$

In the large n limit, the sum can be approximated as an integral over $\mathbf{O}$,

$$\log Z(z) = \log \int d\mathbf{O} e^{-n\mathbf{g}(z)\cdot\mathbf{O}+S(\mathbf{O})} \tag{2.83}$$

where $S(\mathbf{O})$ is the entropy at fixed $\mathbf{O}$,

$$e^{S(\mathbf{O})} = \sum_{x} \delta\left(\mathbf{O} - \mathbf{O}(\mathbf{x})\right). \tag{2.84}$$

Note that $\mathbf{O}$ is in fact a discrete variable. However, $e^{S(\mathbf{O})}$ becomes progressively denser as n increases, and as $n \to \infty$, it becomes continuous. As with Eq. (2.80), the integral can be computed using the saddle point method, yielding

$$\log Z(z) \approx -n\mathbf{g}(z)\cdot\mathbf{O}^* + S(\mathbf{O}^*). \tag{2.85}$$

For this approximation to be valid, the dimension of $\mathbf{O}$, and hence the dimension of $\mathbf{g}$ (which is m), must be subextensive in n. Thus, Schwab *et al.*'s method applies to model in which $m \ll n$ (more technically, $m/n \to 0$ as $n \to \infty$). This restricts it to a relatively small number of natural parameters.

In sum, because Schwab *et al.*'s method involves an m-dimensional saddle-point integral over $\mathbf{O}$, it requires the dimensionality of $\mathbf{O}$ (and hence $\mathbf{g}$) to be small (i.e. $m/n \to 0$ as $n \to \infty$; again, see (Shun and McCullagh, 1995) for details). There are additional steps in their derivation. However, they are not trivial, and they do not lead to additional constraints on their model, so we do not consider them further.

Although high dimensional natural parameters are ruled out by Schwab *et al.*'s method, there are many interesting cases (e.g., models of neural data), in which the elements of $\mathbf{g}$ covary. In those cases, one might think that it would be possible to reduce a high-dimensional latent variable to a low-dimensional one, as we did in previously. While such a reduction is always possible, doing so typically takes the model out of Schwab *et al.*'s class. To see this in a simple setting, we reduce a model with one low-dimensional natural parameter, g, and one high-dimensional natural parameter, $\mathbf{g}$, to a model with just the low-dimensional natural parameter. (Here g might represent the overall firing rate, and the other natural parameters, $\mathbf{g}$, might represent fluctuations around that rate.) The model is written

$$P(\mathbf{x}|g, \mathbf{g}) = e^{-gO(\mathbf{x}) - \mathbf{g}\cdot\mathbf{O}(\mathbf{x}) - \log Z(g,\mathbf{g})} \tag{2.86}$$

where $Z(g, \mathbf{g})$ is the partition function,

$$Z(g, \mathbf{g}) = \sum_{\mathbf{x}} e^{-gO(\mathbf{x}) - \mathbf{g}\cdot\mathbf{O}(\mathbf{x})}. \tag{2.87}$$

Marginalizing over $\mathbf{g}$, we have

$$P(\mathbf{x}|g) = \int d\mathbf{g}\, e^{-gO(\mathbf{x}) - \mathbf{g}\cdot\mathbf{O}(\mathbf{x}) - \log Z(g,\mathbf{g})} P(\mathbf{g}|g) \equiv e^{-gO(\mathbf{x}) - \psi(g, \mathbf{O}(\mathbf{x}))}. \tag{2.88}$$

The function $\psi(g, \mathbf{O}(\mathbf{x}))$ typically has an extremely complicated dependence on g and $\mathbf{x}$. In fact, for all but the simplest model it is not even possible to calculate it analytically, as the partition function cannot be calculated analytically. Thus, $P(\mathbf{x}|g)$ can't be written in the exponential family with a single natural parameter. It can, of course, be written in the exponential family with an exponential number of natural parameters,

$$\psi(g, \mathbf{O}(\mathbf{x})) = \sum_{\mathbf{x}'} \psi(g, \mathbf{O}(\mathbf{x}'))\delta(\mathbf{x} - \mathbf{x}') \tag{2.89}$$

where $\delta(\mathbf{x} - \mathbf{x}')$ is the Kronecker delta, but this clearly takes it out of Schwab *et al.*'s model class. This is closely related to the fact that exponential family distributions are not closed under marginalisation (Seeger, 2005).

Latent variable is the sequence length

To show that a model with sequence length as the latent variable is outside of Schwab *et al.*'s class, we begin by writing the distribution in exponential family form. The simplest way to do that is to write

$$P(\mathbf{x}|z) = \lim_{L\to\infty} e^{\log P(\mathbf{x}) - L\left(1 - \delta_{\dim(\mathbf{x}),z}\right)} \tag{2.90}$$

where δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ if $i = j$ and 0 otherwise) and, as above, $\dim(\cdot)$ denotes dimension (in this case the number of elements in $\mathbf{x}$). This distribution allows only values of $\mathbf{x}$ which have the correct length: if $\dim(\mathbf{x}) = z$, the second term in the exponent is zero, giving $P(\mathbf{x}|z) = P(\mathbf{x})$; in contrast, if $\dim(\mathbf{x}) \neq z$, the second term in the exponent is $-L$, giving a large negative contribution to the energy, and sending $P(\mathbf{x}|z \neq \dim(\mathbf{x})) \to 0$.

This distribution is not in the exponential family form, because the term, $\delta_{\dim(\mathbf{x}),z}$ is not written as the product of a natural parameter (in this case a function of z), and a sufficient statistic (in this case a function of $\mathbf{x}$). It is not possible to write it as a single product, but it can be written as the sum of multiple products,

$$\delta_{\dim(\mathbf{x}),z} = \sum_i \delta_{z,i}\delta_{i,\dim(\mathbf{x})}. \tag{2.91}$$

This is now in the required form, because each term in the sum is the product of a natural parameter ($\delta_{z,i}$, which is function of z), and a sufficient statistic, ($\delta_{i,\dim(\mathbf{x})}$, which is a function of $\mathbf{x}$). Inserting this into Eq. (2.90) gives

$$P(\mathbf{x}|z) = \lim_{L\to\infty} e^{\log P(\mathbf{x}) - L\left(1 - \sum_i \delta_{z,i}\delta_{i,\dim(\mathbf{x})}\right)}. \tag{2.92}$$

This is in the exponential family. However, there are $\mathcal{O}(n)$ terms in the sum, where n is the mean sequence length, so it is not in Schwab *et al.*'s model class.

Entropy of a place field model

Here we compute the entropy, at fixed z, of the place field model in Eq. (2.29), and show that it depends very weakly on z. Because the distribution over $\mathbf{x}$ is conditionally independent given z, the entropy has a simple form,

$$H_{\mathbf{x}|z}(z) = \sum_i H_B\big(p(z - \theta_i)\big) \tag{2.93}$$

where $p(z-\theta_i)$ is the probability that $x_i = 1$ given z,

$$p(z-\theta_i) \equiv \frac{e^{-h+A\cos(z-\theta_i)}}{1+e^{-h+A\cos(z-\theta_i)}}\,, \tag{2.94}$$

and $H_B(p)$ is the entropy (in nats) of a Bernoulli random variable,

$$H_B(p) \equiv -p\log p - (1-p)\log(1-p). \tag{2.95}$$

To understand how this scales with z, we make the change of variables

$$z = \theta_j + \delta z \tag{2.96}$$

where θ_j is chosen to minimize $|\delta z|$. The mean value theorem tells us that for any smooth function $f(z)$,

$$f(z+\delta z) = f(z) + \delta z f'(z^*) \tag{2.97}$$

where prime denotes derivative and z^* is between z and $z+\delta z$. Consequently, for some z^* close to θ_j,

$$H_{\mathbf{x}|z}(z) = \sum_i H_B\big(p(\theta_j-\theta_i)\big) + \delta z \sum_i \frac{\partial H_B\big(p(z^*-\theta_i)\big)}{\partial z^*}. \tag{2.98}$$

Because the θ_i are evenly spaced, the first term is independent of z. Except at $p=0$ or 1 (which are not allowed if h and A are finite), the sum over i of the second term is $\mathcal{O}(n)$. The spacing between adjacent θ_i is $2\pi/n$, so $|\delta z| \le \pi/n \sim \mathcal{O}(1/n)$. Consequently, the second term in Eq. (2.98) scales as $\mathcal{O}(1/n)\times\mathcal{O}(n) \sim \mathcal{O}(1)$, and so $\mathcal{O}(1)$ changes in z produce $\mathcal{O}(1)$ changes in the entropy.

Chapter 3

Probabilistic Synapses

3.1 Abstract

Learning, especially rapid learning, is critical for survival. However, learning is hard: a large number of synaptic weights must be set based on noisy, often ambiguous, sensory information. In such a high-noise regime, keeping track of probability distributions over weights — not just point estimates — is the optimal strategy. Here we hypothesize that synapses take that optimal strategy: they do not store just the mean weight; they also store their degree of uncertainty — in essence, they put error bars on the weights. They then use that uncertainty to adjust their learning rates, with higher uncertainty resulting in higher learning rates. We also make a second, independent, hypothesis: synapses communicate their uncertainty by linking it to variability, with more uncertainty leading to more variability. More concretely, the value of a synaptic weight at a given time is a sample from its probability distribution. These two hypotheses cast synaptic plasticity as a problem of Bayesian inference, and thus provide a normative view of learning. They are consistent with known learning rules, offer an explanation for the large variability in the size of post-synaptic potentials, and make several falsifiable experimental predictions.

3.2 Introduction

To survive, animals must accurately estimate the state of the external world. This estimation problem is plagued by uncertainty: not only is information often extremely limited (e.g., because it is dark) or ambiguous (e.g., a rustle in the bushes could be the wind, or a it could be a predator), but sensory receptors, and indeed all neural circuits, are noisy. Historically, models of neural computation ignored this uncertainty, and relied instead on the idea that the nervous system

represents a single point estimate (Poggio, 1990). However, this does not seem to be what animals do — not only does ignoring uncertainty lead to suboptimal decisions, it is inconsistent with a large body of experimental work (Knill and Richards, 1996; Pouget et al., 2013). Thus, the current view is that in many, if not most, cases, animals keep track of uncertainty, and use it to guide their decisions (Pouget et al., 2013).

Accurately estimating the state of the world is just one problem faced by animals. They also need to learn, and in particular they need to leverage their past experience. It is believed that learning primarily involves changing synaptic weights. But estimating the correct weight, like estimating the state of the world, is plagued by uncertainty: not only is the information available to synapses often extremely limited (in many cases just pre and post synaptic activity), but that information is extremely noisy. Historically, models of synaptic plasticity ignored this uncertainty, and relied instead on the idea that synapses make a single point estimate of their weight (Pouget et al., 2013). However, uncertainty is important for optimal learning — just as it is important for optimal inference of the state of the world.

Motivated by this observation, we propose two hypotheses. The first, Bayesian Plasticity, states that during learning, synapses do indeed take uncertainty into account. Under this hypothesis, synapses do not just try to find a point estimate of their weights, as is done in almost all learning rules in neuroscience; instead, they learn a probability distribution over their weights. This allows synapses to adjust their learning rates on the fly: when uncertainty is high, learning rates are turned up, and when uncertainty is low, learning rates are turned down. These adjustments allow synapses to learn faster, so there is likely to be considerable evolutionary pressure for such a mechanism.

Bayesian Plasticity is a hypothesis about what synapses compute. It does not, however, tell us how synapses should set their weights. For that we need a second hypothesis. Here we propose that weights are sampled from the probability distribution describing the synapses's degree of uncertainty. Under this hypothesis, which we refer to as Synaptic Sampling, trial to trial variability gives us a direct readout of uncertainty: the larger the trial to trial variability in a synaptic strength, the larger the uncertainty. Combined, these hypotheses make several strong experimental predictions. One is consistent with re-analysis of existing experimental data; the others, which are feasible in the not so distant future, could falsify the model.

3.3 Results

We begin our analysis with a derivation of learning rules under the assumption that synapses keep track of their uncertainty (Bayesian Plasticity). That gives us a set of rules for updating not just the mean weight (as all standard learning rules do), but also the uncertainty. We then add to our framework a method for choosing the PSP variability (Synaptic Sampling). Finally, we discuss the experimental implications of our two hypotheses.

We begin with a simplified model of synaptic integration. Neurons *in vivo* receive a constant barrage of spikes, and each incoming spike produces a PSP — a small change in the postsynaptic neuron's membrane potential. Very approximately, PSPs combine linearly, allowing us to write the membrane potential relative to rest as

$$V(t) = \sum_i w_i(t) x_i(t) + \eta_V(t) \tag{3.1}$$

where $x_i(t)$ is the synaptic input from neuron i, $w_i(t)$ is the corresponding PSP amplitude, and $\eta_V(t)$ is the membrane potential noise. For simplicity we work in discrete time, so $t = 0, 1, 2, ...$, and time steps are on the order of the membrane time constant, around 10 ms (Tripathy et al., 2015). The synaptic inputs, $x_i(t)$, represent the number of incoming spikes in a time step. For most of our analysis, $x_i(t)$ is either 0 (no spike) or 1 (spike), with the probability of a spike chosen to correspond to typical firing rates observed in cortex. To take into account variability in PSP amplitudes, $w_i(t)$ varies from time step to time step. See Methods, Sec. 3.5 for additional details.

We are interested in how synapses learn a set of target weights, denoted $w_{\text{tar},i}(t)$, and a target membrane potential, $V_{\text{tar}}(t)$; the two are related via

$$V_{\text{tar}}(t) = \sum_i w_{\text{tar},i}(t) x_i(t). \tag{3.2}$$

These target weights have different meanings in different contexts, but broadly, they are the weights that allow the neuron to perform its particular task as effectively as possible. For instance, in a cerebellar Purkinje cell, the target weights might allow the cell to best predict the occurence of an airpuff; in motor cortex, the target weights might allow the cell to contribute to the best possible skilled movement (e.g., a golf swing that gives a hole-in-one); and in visual cortex, the target weights might enable the cell to pick out the most interesting visual feature in its input. Note that the target weights are unlikely to be fixed, as the statistics of the external world are not fixed (e.g., the stimuli predicting an airpuff can change), nor is the organism (e.g., as you get stronger you will need

to adapt your golf swing). Thus, we expect the target weights to change over time, something we include in our analysis (see Methods, Sec. 3.5).

To learn the target weights, synapses get information from the presynaptic input, backpropagating action potentials, and, for supervised and reinforcement learning, an explicit feedback signal, denoted f. The simplest feedback signal, which corresponds to the typical supervised learning set-up (Widrow and Hoff, 1960; Albus, 1971), is $f(\delta) = \delta$, where δ is the prediction error corrupted by additive noise, η_δ,

$$\delta(t) = V_{\text{tar}}(t) - V(t) + \eta_\delta(t). \tag{3.3}$$

We refer to this as continuous feedback, because f is a continuous function of δ. However, our framework is flexible enough to cover many other supervised and reinforcement learning feedback signals, including discontinuous ones, and even unsupervised learning, for which there is no feedback signal. In particular, we consider three scenarios. The first corresponds to cerebellar learning, in which a Purkinje cell receives a complex spike if its output is too high, thus triggering long term depression (Ito et al., 1982). To mimic the all-or-nothing nature of a complex spike (Eccles et al., 1966), we use a binary feedback signal: $f(\delta) = \text{sign}(\delta - \theta)$. For this feedback signal, f is 1 if the noisy error signal, δ is above a threshold, θ, and f is -1 if it is below that threshold. The second scenario corresponds to reinforcement learning, in which the feedback, now representing the reward, reports the magnitude of the noisy error signal, but not its direction, $f(\delta) = -|\delta|$. The third corresponds to unsupervised learning, in which there is no feedback signal. Instead, synapses adjust their weights using a Hebbian-like learning rule to find the most interesting (in this case, non-Gaussian) direction in the inputs. See Methods, Sec. 3.5, for additional details.

For the continuous feedback signal, $f = \delta$, there is a well known rule for finding the optimal weights: the delta rule (Widrow and Hoff, 1960; Dayan and Abbott, 2001), which changes the mean PSP amplitude, m_i, according to

$$\Delta m_i = \alpha x_i \delta. \tag{3.4}$$

(We focus on the mean weight because the actual weight, w_i, varies considerably from one time step to the next due to stochastic vesicle release (Branco and Staras, 2009).) This is the product of a learning rate, α (red), a presynaptic term, x_i (green) and a postsynaptic term δ (blue). Importantly, the learning rate, α, is the same for all synapses, so all synapses whose presynaptic cells are active (i.e., for which $x_i = 1$) change by the same amount (the red arrow labelled "delta rule" in Fig. 3.1).

In the absence of any other information about the history of inputs, the delta rule

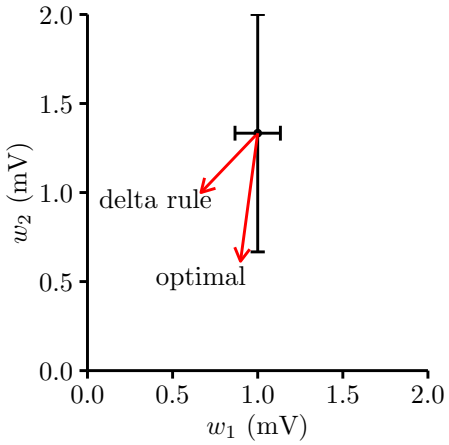

Figure 3.1: Comparison of the delta rule and the optimal learning rule. The error bars denote uncertainty in the two synapses' estimates of their synaptic weights. The first synapse (w_1) is reasonably certain; the second synapse (w_2) is less so. The red arrows denote possible changes in the weight in response to a negative feedback signal. The arrow labelled "delta rule" represents an equal decrease in the first and second weights. In contrast, the arrow labelled "optimal" takes uncertainty into account, so there is a larger change in the second, more uncertain, weight.

is perfectly reasonable. However, suppose that, based on previous information, synapse 1 is relatively certain about its weight, whereas synapse 2 is uncertain (error bars in Fig. 3.1). In that case, new information should have a larger impact on synapse 2 than synapse 1, so synapse 2 should update its weight more (red arrow labelled "optimal" in Fig. 3.1). Thus, the delta rule does not exploit information about uncertainty, even when it is available, making it suboptimal. To do better, synapses need to compute their uncertainty (essentially, provide error bars), and exploit that information when updating the weights. In essence, synapses must solve an inference problem, in which the goal is to infer the probability distribution over the target weights given available data. So instead of keeping track of point estimates and updating those when spikes arrive, as in the delta rule, synapses keep track of probability distributions over their weights, and update the whole distribution when spikes arrive. That updating process is illustrated in Fig. 3.2.

We refer to learning in which synapses keep track of probability distributions as Bayesian Plasticity, so named because the update rules are derived using Bayes' theorem. Synapses do not, of course, have the resources to keep track of arbitrary probability distributions. We therefore assume that each synapse uses an approximate form for its probability distribution, a log normal, chosen because it does not allow weights to change from excitatory to inhibitory (see Methods, Sec. 3.5). Using this approximate distribution, synapses only have to keep track of the mean and variance, denoted m_i and s_i^2, respectively. As we show in Supplementary Information, Secs. 3.6 and 3.6, in the case of supervised learning with

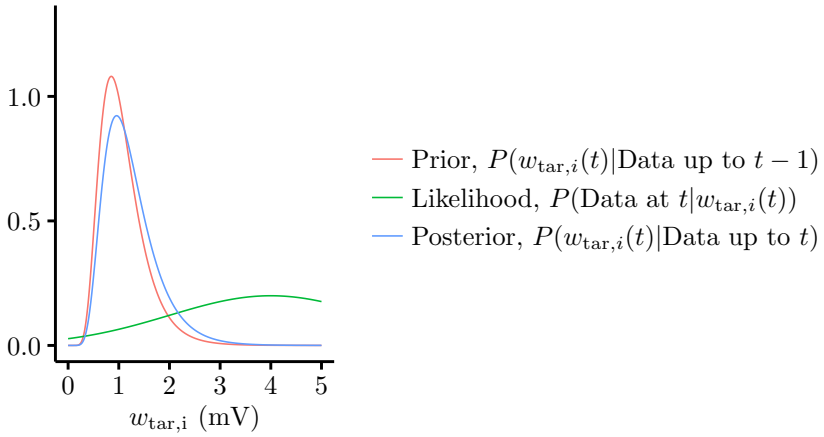

Figure 3.2: Updating the distribution over weights using Bayes theorem. At time t, synapse i's current probability distribution over the target weight, $w_{\text{tar},i}$, is given by $P(w_{\text{tar},i}(t)|\text{Data up to } t-1)$ (red curve). The neuron receives a small amount of new information via the likelihood, $P(\text{Data at } t|w_{\text{tar},i}(t))$ (green curve). This leads to a new distribution, $P(w_{\text{tar},i}(t)|\text{Data up to } t)$ (blue curve).

continuous feedback, $f = \delta$, the update rules for the mean, m_i and variance, s_i^2, are approximately,

$$\Delta m_i \approx \quad \alpha_i \; x_i \; \delta \quad -\frac{1}{\tau}\left(m_i - m_{\text{prior}}\right) \tag{3.5a}$$

$$\Delta s_i^2 \approx -\alpha_i \; x_i^2 \; s_i^2 \quad -\frac{2}{\tau}\left(s_i^2 - s_{\text{prior}}^2\right) \tag{3.5b}$$

where α_i is the learning rate, which now varies across synapses (see Eq. (3.6) below), and τ, m_{prior}, and s_{prior}^2 are fixed parameters. To move to the fully general case, including reinforcement and unsupervised learning, we simply replace the postsynaptic terms, δ in the update for the mean, and s_i^2 in the update for the variance by something slightly more complicated (see Supplementary Information, Eq. (3.59)).

The update rule for the mean weight, Eq. (3.5a), is very similar to the delta rule, in that it is composed of a learning rate (red), a presynaptic term (green) and a postsynaptic term (blue). However, there are two important differences. First, as we show in Supplementary Information, Sec. 3.6, the learning rate, α_i, is proportional to each synapse's uncertainty, as measured by s_i^2,

$$\alpha_i = \frac{s_i^2}{s_\delta^2} \tag{3.6}$$

where s_δ^2 represents the average variability in δ, and hence in the feedback signal (see Supplementary Information, Eq. (3.48), for the definition of s_δ^2). Thus, when a synapse is more uncertain about its target weight, new information causes a larger change in the mean weight — exactly what we expected, given Fig. 3.1.

In contrast, as the feedback signal gets noisier, and thus less informative, the learning rate falls. Second, there is a decay term (grey), which causes the mean to decay back to its prior value. This accounts for the fact that the underlying target weight, $w_{\text{tar},i}$, changes over time (as mentioned above), so information from the recent past is more relevant than information from the distant past.

Although the update rule for the uncertainty, s_i^2 (Eq. (3.5b)), does not have a counterpart in classical learning rules, it does have a natural interpretation. The first term in Eq. (3.5b) reduces uncertainty (note the negative sign) whenever the presynaptic cell is active ($x_i = 1$), and thus whenever the synapse updates its estimate of the weight. The second term has the opposite effect: it increases uncertainty. That term arises because random drift reduces knowledge about the target weights.

Simulations (Fig. 3.3) show that the mean weight tracks the target weight very effectively (compare the red and blue lines, which correspond to the mean of the inferred distribution and the target weight, respectively). Just as importantly, the synapse's estimate of its uncertainty tracks the difference between its estimate and the actual target (the blue line should be inside the 95% confidence intervals 95% of the time; in practice, we have: supervised continuous, 95%; supervised binary, 94%; reinforcement, 89%; unsupervised, 87%).

The critical aspect of the learning rules in Eq. (3.5) is that the learning rate — the change in mean PSP amplitude, m_i, per spike — increases as the synapse's uncertainty, s_i^2, increases. This is a general feature of our learning rules, and not specific to any one of them. Consequently, independent of the learning scenario, we expect performance to be better than for classical learning rules, which do not take uncertainty into account. To check whether this is true, we computed the mean squared error between the actual and target membrane potential, V and V_{tar}, for classical learning rules, and plotted them relative to our learning rules. The results are shown in Fig. 3.4. In this figure, the red line gives the mean squared error for the classical learning rules relative to the error for our optimal rules. Note that the Bayesian learning rules do not have an externally imposed learning rate parameter, so their mean squared error is a single value that does not vary with learning rate. Even if the learning rates for the classical learning rules are chosen optimally, performance is worse than it is for the probabilistic learning rules, and if they are chosen sub-optimally, performance can be much worse.

Fig. 3.4 indicates that there is a clear advantage to using uncertainty to adjust learning rates. But does the brain actually take this strategy? Addressing that question will require a new generation of plasticity experiments: at present, in typical plasticity experiments only changes in weights are measured; to test our hypothesis, it will be necessary to measure changes in learning rates, and at the

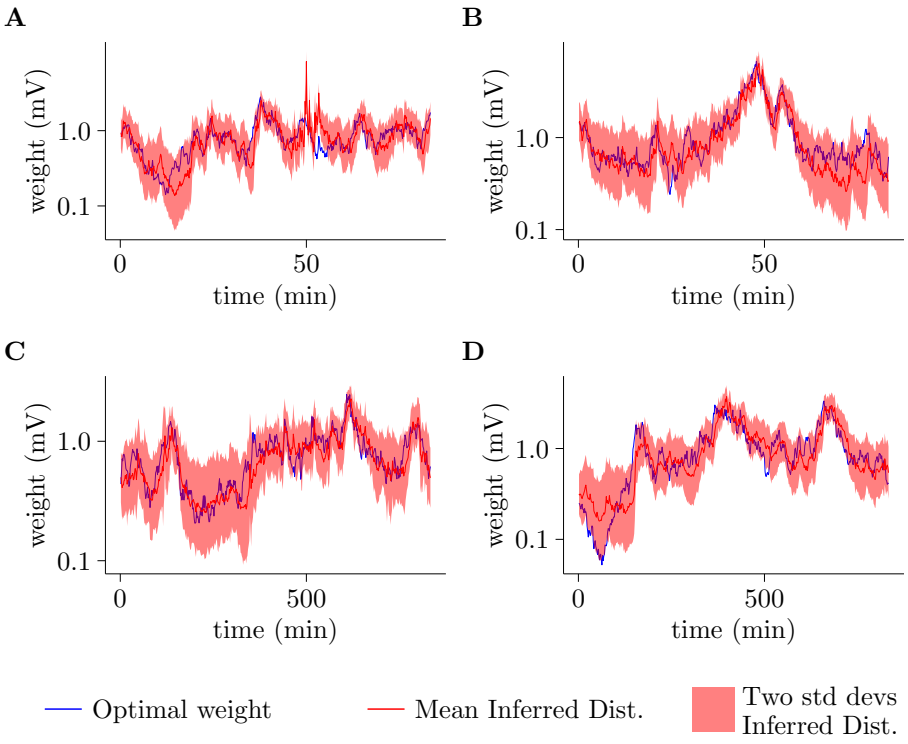

Figure 3.3: Bayesian learning rules track the true weight and estimate uncertainty. The blue line is the true weight, the red line represents the median of the inferred distribution, and the red area represents 95% confidence intervals. The total time course is 5 times the characteristic time over which the target weights change (see Methods, Sec. 3.5). **A**. Supervised learning, continuous feedback ($f = \delta$). **B**. Supervised learning, binary feedback ($f = \Theta(\delta - \theta)$). **C**. Reinforcement learning ($f = -|\delta|$). **D**. Unsupervised learning (no feedback).

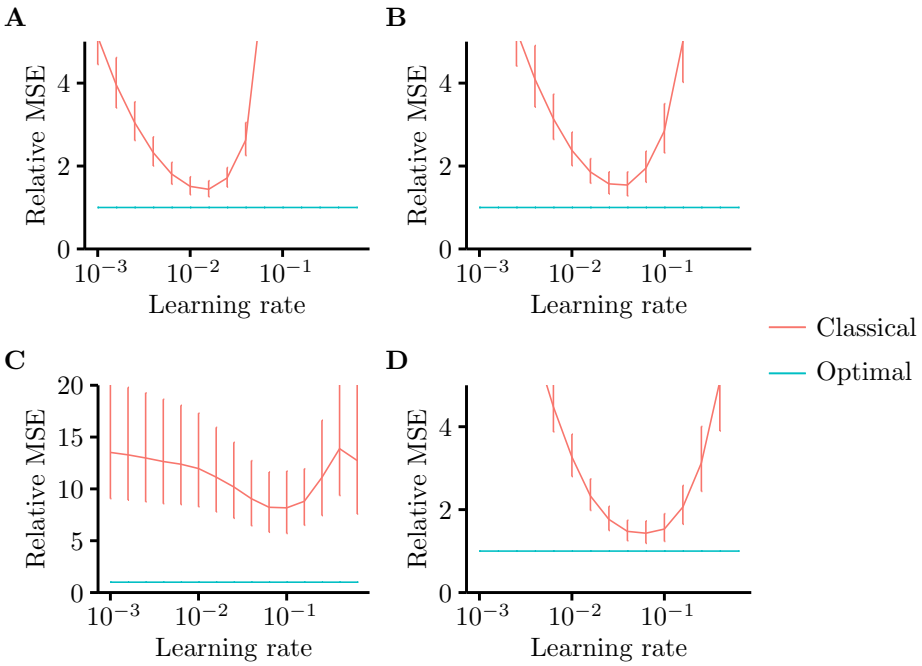

Figure 3.4: Bayesian learning rules have a lower mean squared error (MSE) than classical learning rules. The red line is the mean squared error for the classical learning rule, relative to our Bayesian learning rule (the blue line at 1). The Bayesian learning rule does not have a tuneable learning rate parameter, so the Bayesian mean squared error is the same for all learning rates. **A**. Supervised learning, continuous feedback ($f = \delta$). **B**. Supervised learning, binary feedback ($f = \Theta(\delta - \theta)$). **C**. Reinforcement learning ($f = -|\delta|$). **D**. Unsupervised learning (no feedback). See Methods, Sec. 3.5, for further details.

same time determine how those changes are related to the synapse's uncertainty. This presents two challenges. First, measuring changes in learning rates is difficult, as weights must be monitored over long periods of time and under natural conditions, preferably *in vivo*. However, with the advent of increasingly sophisticated experimental techniques, such experiments should be feasible in the not so distant future. Second, we cannot measure the synapse's uncertainty directly. It is, therefore, necessary to find a proxy. Below we discuss two possible approaches.

The first approach is indirect: use neural activity measured over long periods *in vivo* to estimate the uncertainty a synapse should have; then, armed with that estimate, test the prediction that the learning rate increases with uncertainty. To estimate the uncertainty a synapses should have, we take advantage of a general feature of essentially all learning rules: synapses get information only when the presynaptic neuron spikes. Consequently, the synapse's uncertainty should fall as the presynaptic firing rate increases. In fact, under mild assumptions, we can derive a very specific relationship: the relative change in weight under a plasticity protocol, $\Delta m_i/m_i$, should scale as $1/\sqrt{\nu_i}$ where ν_i is the firing rate of the neuron presynaptic to synapse i,

$$\frac{\Delta m_i}{m_i} \propto \frac{1}{\sqrt{\nu_i}}, \tag{3.7}$$

a relationship that holds in our simulations (Fig. 3.5; see also Supplementary Information, Sec. 3.6). In essence, firing rate is a proxy for uncertainty, with higher firing rate indicating lower uncertainty and vice versa. This prediction can be tested by observing neurons *in vivo*, estimating their firing rates, then performing long term potentiation or depression experiments to determine the relative change in synaptic strength, $\Delta m_i/m_i$.

The second approach involves the introduction of a new hypothesis, which is that PSP variability provides a proxy for uncertainty. That we might expect a relationship between variability and uncertainty is based on the following normative reasoning (see Methods, Sec. 3.5, for an extended discussion): the uncertainty associated with a particular computation should depend on the uncertainty in the weights; thus, to make optimal decisions, the brain needs to know that degree of uncertainty; one way to communicate it is via variability in PSP amplitude. This leads to the Synaptic Sampling hypothesis, which states that the variance in PSP amplitude is equal to the variance of the inferred posterior distribution over the target weight, s_i^2,

$$\text{PSP variance} = s_i^2. \tag{3.8}$$

This is analogous to setting the PSP mean to the mean of the distribution over the target weight, m_i. We call this the Synaptic Sampling hypothesis because the

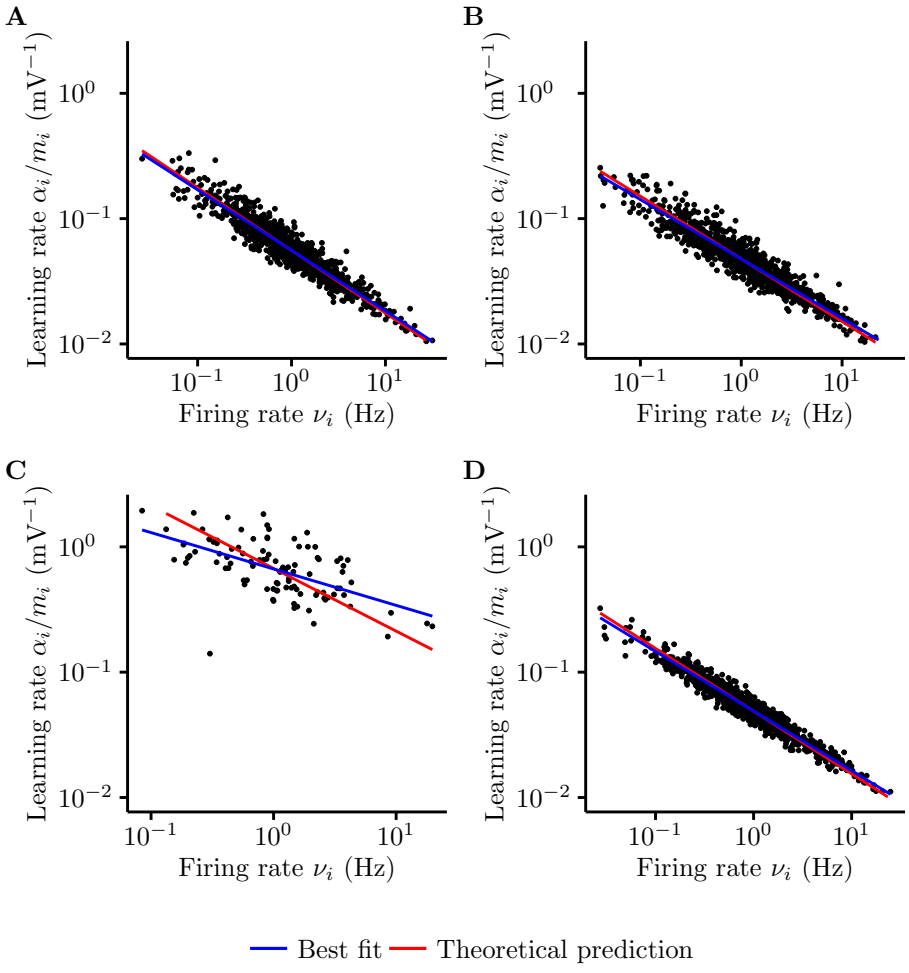

Figure 3.5: Simulations confirming that the normalized learning rate (α_i/m_i, which is proportional to $\Delta m_i/m_i$) is inversely related to the square root of the firing rate. As predicted, the best fit line on a log-log plot has a slope close to -1/2. **A**. Supervised learning, continuous feedback ($f = \delta$). **B**. Supervised learning, binary feedback ($f = \Theta(\delta - \theta)$). **C**. Reinforcement learning ($f = -|\delta|$). **D**. Unsupervised learning (no feedback).

synapses "sample" PSP amplitudes from their inferred distribution over weights.

Bayesian Plasticity combined with Synaptic Sampling tells us that synapses with higher variability (and hence higher uncertainty) should have higher learning rates. More quantitively, Bayesian Plasticity tells us that the relative change in PSP amplitude, $\Delta m_i/m_i$, is proportional to the synapase's uncertainty, (Eqs. (3.5a) and (3.6)) and Synaptic Sampling relates uncertainty to variability (Eq. (3.8)); consequently,

$$\frac{\Delta m_i}{m_i} \propto \frac{\text{PSP variance}}{\text{PSP mean}} \equiv \text{Normalized Variability} \quad , \tag{3.9}$$

where we have defined the normalized variability to be the ratio of PSP variance to its mean. We verify that this relationship holds in simulation in Fig. 3.6.

Equation (3.9) implies that when the PSP variance is high, learning is fast. Testing that experimentally is straightforward, if technically difficult: simply monitor the PSP mean and variance for long periods *in vivo*, and compare normalized variability to changes in the mean. The *in vivo* requirement is important: our analysis assumes a constant barrage of presynaptic spikes, whereas in many *in vitro* preparations the vast majority of cells are silent (see Supplementary Information, Sec. 3.6).

In addition to the experiment proposed above, there is a slightly more indirect test of Bayesian plasticity and Synaptic Sampling. Combining Eq. (3.7) and (3.9), we see that the normalized variability and firing rate obey the relationship,

$$\frac{1}{\sqrt{\nu_i}} \propto \text{Normalized Variability} \quad . \tag{3.10}$$

This is intuitively sensible: as discussed previously, higher presynaptic firing rates means the synapse is more certain, and Synaptic Sampling states that higher certainty should reduce the observed variability.

This relationship can be tested by estimating presynaptic firing rates *in vivo*, and comparing them to the normalized variability measured using paired recordings. Such data can be extracted from experiments by Ko et al. (2011). In those experiments, calcium signals in mouse visual cortex were recorded *in vivo* under a variety of stimulation conditions, which provided an estimate of firing rate; subsequently, whole cell recordings of pairs of identified neurons were made *in vitro*, and the mean and variance of the PSPs were measured. In Fig. 3.7A we plot the normalized variability versus the firing rate on a log-log scale; on this scale, our theory predicts a slope of $-1/2$ (red line). The normalized variability does indeed decrease as the firing rate increases (blue line), ($p < 0.003$), and the slope is not significantly different from $-1/2$ ($p = 0.56$). This pattern is broadly matched by simulated data (Fig. 3.7B)

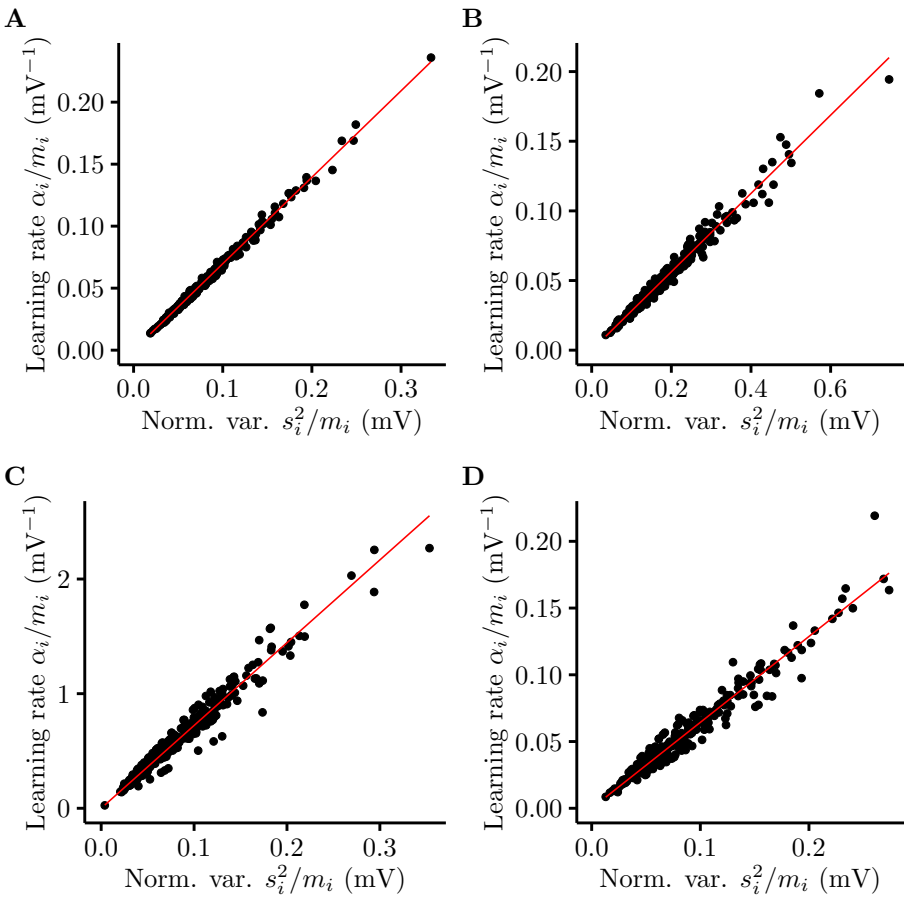

Figure 3.6: Simulations confirming that that the normalized learning rate (α_i/m_i, which is proportional to $\Delta m_i/m_i$) is proportional to the normalized variability (s_i^2/m_i). The red line is the best fitting straight-line that passes through the origin. **A**. Supervised learning, continuous feedback ($f = \delta$). **B**. Supervised learning, binary feedback ($f = \Theta(\delta - \theta)$). **C**. Reinforcement learning ($f = -|\delta|$). **D**. Unsupervised learning (no feedback).

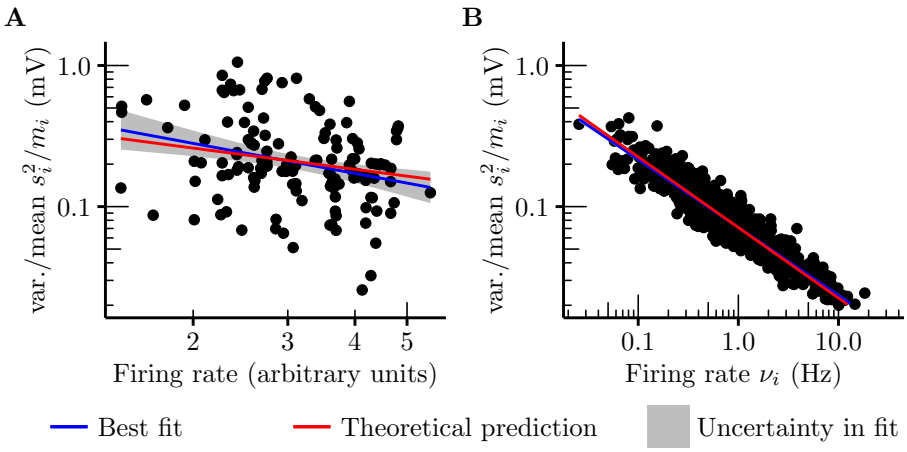

Figure 3.7: Normalized variability (the ratio of the PSP variance to the mean) as a diagnostic of our theory. **A**. Normalized variability falls as firing rate increases. The red line, which has a slope of $-1/2$, is our prediction (the intercept, for which we do not have a prediction, was chosen to give the best fit to the data). The blue line is fit by linear regression, and the grey region represents 2 standard errors. Its slope, -0.62, is statistically significantly different from 0 ($p < 0.003$) and not significantly different from $-1/2$ ($p = 0.57$). Firing rate was measured by taking the average signal from a spike deconvolution algorithm (Vogelstein et al., 2010). Units are arbitrary because the scale factor relating the average signal from the deconvolution algorithm and the firing rate is not exactly one (Packer et al., 2015). Data from layer 2/3 of mouse visual cortex (Ko et al., 2011). **B**. Simulated normalized variability versus firing rate; supervised learning with continuous feedback ($f = \delta$).

It seems unlikely that this pattern emerged spuriously, as that would require a confound that simultaneously influenced two very different types of measurement, calcium measurements of the pre-synaptic firing rate and patch-clamp measurement of the PSPs. The most obvious confound actually predicts a positive slope: if more calcium indicator is present in the presynaptic cell, then we might expect measured firing rates to be higher, and vesicle release probabilities to be lower (as the indicator buffers calcium involved in vesicle release). Lower probabilities imply higher variability, so we would expect higher measured firing rates to be associated with higher variability — the opposite of our prediction.

3.4 Discussion

In summary, based primarily on theoretical considerations of optimality we proposed that synapses do not just keep track of point estimates of their weights, as they do in classical learning rules; instead, they compute approximate probability distributions over their weights. They then use those distributions to set learning rates: the wider the distribution (that is, the more the uncertainty in the target weight) the higher the learning rate. This allows different synapses

to have different learning rates, and leads to learning rules that allow synapses to exploit all locally available information, and so learn as rapidly as possible — much more rapidly than classical learning rules, which do not keep track of uncertainty (Fig. 3.4). The critical difference between our learning rules and classical ones is that the learning rates themselves undergo plasticity; the rules for updating the mean weight are very similar to classical learning rules. Thus, our framework is consistent with the vast majority of work on synaptic plasticity (Bi and Poo, 1998; Abbott and Nelson, 2000; Turrigiano and Nelson, 2004; Pfister and Gerstner, 2006; Ponte Costa et al., 2015; Ziegler et al., 2015).

The hypotheses that synapses keep track of uncertainty, which we refer to as the Bayesian Plasticity hypothesis, makes the general prediction that learning rates, not just synaptic strengths, are a function of pre and postsynaptic activity — something that should be testable with the next generation of plasticity experiments. In particular, it makes a specific prediction about learning rates *in vivo*: learning rates should vary across synapses, being higher for synapses with lower presynaptic firing rates.

We also make a second, independent, hypothesis, Synaptic Sampling. This hypothesis states that the variability in PSP size associated with a particular synapse matches the uncertainty in the strength of that synapse. This allows synapses to communicate their uncertainty to surrounding circuitry — information that is critical if the brain is to monitor the accuracy of its own computations. The same principle has been applied to neural activity, where it is known as the neural sampling hypothesis (Hoyer and Hyvarinen, 2003; Fiser et al., 2010; Berkes et al., 2011a; Orbán et al., 2016) (except that here variability in neural activity matches uncertainty about the state of the external world). The neural sampling hypothesis meshes well with synaptic sampling: uncertainty in the weights increases uncertainty in the current estimate of the state of the world, and likewise, variability in the weights increase variability in current neural activity (see Methods, Sec. 3.5). However, while there is some experimental evidence for the neural sampling hypothesis (Berkes et al., 2011a; Haefner et al., 2016; Orbán et al., 2016), it has not been firmly established. Whether other proposals for encoding probability distribution with neural activity, such as probabilistic population codes (Pouget et al., 2013; Ma et al., 2006), can be combined with Synaptic Sampling is an open question.

By combining our two hypotheses, we were able to make additional predictions. These predictions focused on what we call the normalized variability — the ratio of the variance in PSP size to the mean. First, we predicted that plasticity should increase with normalized variability, which remains to be tested. Second, we predicted that normalized variability should decrease with presynaptic firing rate. We reanalysed data from Ko et al. (2011) to show that this is indeed the

case (Fig. 3.7).

In machine learning, the idea that it is advantageous to keep track of the distribution over weights has a long history (Buntine and Weigend, 1991; MacKay, 1992; Blundell et al., 2015). The first suggestion that such a scheme might be useful in a neuroscience context, however, was relatively recent (Pouget et al., 2013), and the first theoretical study was even more recent (Kappel et al., 2015). The latter study bore some resemblance to ours, in that weights were sampled from a distribution. However, there was an important difference: the distribution had to be fixed, and could be determined only after the animal had seen all data. Because this is unrealistic, an online algorithm was developed in which, as in our scheme, weights were updated on each time step. However, for this algorithm to agree with sampling from a fixed distribution, changes in synaptic strength per time step had to be very small (on the order of 10^{-4}). Thus, unlike in our scheme, there was almost no spike-to-spike variability in PSP size. So, although this was an important step toward a probabilistic treatment of synaptic plasticity, the algorithm was unable to deal with the realistic situation in which the distribution over synaptic weights is changing continuously as the animal receives new information, and it doesn't produce the variability in PSP size seen *in vivo*.

If the Bayesian Plasticity hypothesis is correct, synapses would have to keep track of, and store, two variables: the mean and variance of the log of the synaptic weight (or, equivalently, the mean weight and the learning rate). The complexity of synapses (Kasai et al., 2012; Südhof, 2012; Michel et al., 2015), and their ability to use interesting, non-trivial learning rules (e.g. synaptic tagging, in which activity at a synapses "tags" it for future long term changes in strength (Frey and Morris, 1997; Redondo and Morris, 2011; Rogerson et al., 2014), and metaplasticity, in which the learning rate can be modified by synaptic activity without changing the synaptic strength (Abraham and Bear, 1996; Abraham, 2008; Hulme et al., 2014)), suggests that representing uncertainty — or learning rate — is quite possible. It will be nontrivial, but important, to work out how.

Our framework has several implications, both for the interpretation of neurophysiological data and for future work. First, under the Synaptic Sampling hypothesis, PSPs are necessarily noisy. Consequently, noise in synapses (e.g., synaptic failures) is a feature, not a bug. We thus provide a normative theory for one of the major mysteries in synaptic physiology: why neurotransmitter release is probabilistic. Second, our approach allows us to derive local, biologically plausible learning rules, no matter what information is available at the synapse, and no matter what the statistics of the synaptic input. Thus, our approach provides the flexibility necessary to connect theoretical approaches based on optimality to complex biological reality.

In neuroscience, Bayes theorem is typically used to analyze high level inference

problems, such as decision-making under uncertainty. Here we have demonstrated that Bayes' theorem, being the optimal way to solve any inference problem, big or small, could be implemented in perhaps the smallest computationally relevant elements in the brain: the synapse.

3.5 Methods

Here we provide a complete description of our model (Sec. 3.5, which includes a table containing a list of all parameters), sketch the derivation of the learning rules (Sec. 3.5; the full derivation is given in Supplementary Information), discuss the advantages of our local approach to learning (Sec. 3.5), provide details of the simulations (Sec. 3.5), give a normative explanation for the Synaptic Sampling hypothesis (Sec. 3.5), and, finally, provide additional details of the statistical test used for Fig. 3.7A (Sec. 3.5).

Complete description of our model

In the main text we specified how the membrane potential depends on the weights and incoming spikes (Eq. (3.1)) and how the target membrane potential depends on the target weights (Eq. (3.2)), and we defined the prediction error (Eq. (3.3)). Here we describe how the weights, w_i, the target weights, $w_{\text{tar},i}$, and the spikes, x_i, are generated. We also provide a summary of how the feedback signal, f, depends on the prediction error, δ, and we provide details of the unsupervised learning model.

Synaptic weights

To take variability in PSP amplitudes into account, we use

$$w_i = m_i + \sqrt{k_i m_i}\,\eta_{w_i}, \tag{3.11}$$

where η_{w_i} is zero mean, unit variance noise. Under the Synaptic Sampling hypothesis, the variability is equal to the uncertainty, so $k_i = s_i^2/m_i$. However, when comparing classical and Bayesian learning rules (Figs. 3.3 and 3.4), we set $k_i = k$ for all synapses. This was necessary to make a fair comparison, as there is no way to compute uncertainty for classical learning rules. The value of k came from measured data (Song et al., 2005): we plotted s_i^2 vs m_i and fit a straight line that passed through the origin; k is the slope of that line; this resulted in $k = 0.0877$.

When plotting learning rate versus firing rate (Fig. 3.5), we also used $k_i = k$, primarily for convenience. However, in Figs. 3.6 and 3.7, which explictly involved the Synaptic Sampling hypothesis, we used $k_i = s_i^2/m_i$.

The target weights

The target weights are the weights that in some sense optimize the performance of the animal. We do not expect these weights to remain constant over time, for two reasons. First, both the general state of the world and the organism change over time, thus changing the target weights. Second, we take a local, single neuron view to learning, and define the target weights on a particular neuron to be the optimal weights given the weights on all the other neurons in the network. Consequently, as the weights on surrounding neurons change due to learning, the target weights on our neuron will also change. While these changes may be quite systematic, to a single synapse deep in the brain they are likely to appear random.

In or model we assume that the log of the target weights follow an Ornstein-Uhlenbeck process. Specifically, we define

$$\lambda_{\text{tar},i} = \log |w_{\text{tar},i}| \tag{3.12}$$

(note the absolute value sign, which allows the weights to be either positive or negative), and let $\lambda_{\text{tar},i}$, the log weight, evolve according to

$$\Delta\lambda_{\text{tar},i}(t+1) = -\frac{1}{\tau}\left(\lambda_{\text{tar},i}(t) - \mu_{\text{prior}}\right) + \sqrt{\frac{2\sigma^2_{\text{prior}}}{\tau}}\,\eta_{\text{tar},i} \tag{3.13}$$

where τ is the characteristic time scale over which the weights change. Note that τ is measured in time steps; to convert to time it needs to be multiplied by Δt, the size of the time step. Under this noise process, the mean value of $\lambda_{\text{tar},i}$, denoted μ_i, and the variance, denoted σ_i^2, evolve according to

$$\mu_i(t+1) = \left(1 - \frac{1}{\tau}\right)\mu_i(t) + \frac{\mu_{\text{prior}}}{\tau} \tag{3.14a}$$

$$\sigma_i^2(t+1) = \left(1 - \frac{1}{\tau}\right)^2 \sigma_i^2(t) + \frac{2\sigma^2_{\text{prior}}}{\tau}. \tag{3.14b}$$

We chose this particular noise process for three reasons. First, $w_{\text{tar},i}$ is equal to either $+e^{\lambda_{\text{tar},i}}$ (for excitatory weights) or $-e^{\lambda_{\text{tar},i}}$ (for inhibitory weights), and thus cannot change sign as $\lambda_{\text{tar},i}$ changes with learning. Consequently, excitatory weights cannot become inhibitory, and *vice versa*, so Dale's law is preserved. Second, spine sizes obey this stochastic process (Loewenstein et al., 2011), and while synaptic weights are not spine sizes, they are correlated (Matsuzaki et al., 2004). Third, this noise process gives a log-normal stationary distribution of weights, as is observed experimentally (Song et al., 2005).

The parameters of these dynamics, μ_{prior} and σ^2_{prior}, were set to the mean and

variance of measured log-weights using data from Song et al. (2005). We used a time step, Δt, of 10 ms, within the range of measured membrane time-constants (e.g. (Tripathy et al., 2015)), and set τ to 10^5 (corresponding to 1,000 seconds, or around 15 minutes) for both types of supervised learning, and 10^6 (corresponding to 10,000 seconds, or around 2 1/2 hours) for reinforcement and unsupervised learning. These values of τ were chosen so that uncertainty roughly matched observed variability; see Sec. 3.6.

The synaptic inputs, $x_i(t)$, with feedback

For models with a feedback signal, on each time step x_i is drawn from a Bernoulli distribution representing the number of spikes (0 or 1) from the presynaptic cell,

$$P(x_i) = (\nu_i \Delta t)^{x_i} (1 - \nu_i \Delta t)^{1 - x_i}. \tag{3.15}$$

The firing rates, ν_i, are drawn from a log-normal distribution chosen to match observed firing rates. We choose a distribution that is intermediate between the relatively narrow ranges found by some (O'Connor et al., 2010), and the extremely broad ranges found by others (Mizuseki and Buzsáki, 2013): we use a log-normal distribution, with median at 1 Hz, and with 95% of firing rates being between 0.1 Hz and 10 Hz,

$$\log \nu_i \sim \mathcal{N}\left(0, \left(\frac{\log 10}{2}\right)^2\right). \tag{3.16}$$

Feedback signals for supervised and reinforcement learning

The feedback signal is different for every type of learning (these are mentioned in the text, and are repeated here for completeness).

For supervised learning with continuous feedback, the feedback signal is simply δ,

$$f(\delta) = \delta. \tag{3.17}$$

For supervised learning with binary feedback, the feedback signal is 1 if δ is above θ, and -1 if it is below θ,

$$f(\delta) = \text{sign}(\delta - \theta). \tag{3.18}$$

Binary feedback is intended to model Purkinje cells, which receive a complex-spike feedback signal relatively rarely (around once per second; corresponding to once in 100 time steps). To match that rate, θ should be set high enough

that δ is above θ relatively rarely. While this is possible (and we have run these simulations), this makes comparison between the Bayesian and classical rules difficult: it is not sufficient simply to fix θ, as this may give rise to different values of $P(f=1)$ in Bayesian and classical learning. While it may be possible to resolve these difficulties, for the purposes of fair comparison we use $\theta = 0$. Because the distribution over δ is symmetric around 0, this implies that $P(f=1)$ remains at $1/2$ throughout the simulations for both classical and Bayesian learning.

For reinforcement learning, the feedback signal, representing the reward, is simply minus the magnitude of δ,

$$f(\delta) = -|\delta| . \tag{3.19}$$

Models without a feedback signal

For unsupervised learning, there is no feedback signal. Instead, information for setting the weights comes from structure in the synaptic inputs, $\mathbf{x}$, which is generated by a very different process from supervised and reinforcement learning (for which there was no structure in the input). Specifically, we assume that the cell's input is Gaussian in every direction except one, $\mathbf{w}_{\text{tar}}$, in which the input is Laplacian. The cell's goal is to find that one interesting direction (as was done in Intrator and Cooper (1992)).

Formally, $\mathbf{x}$ is generated by,

$$P(\mathbf{x}|\mathbf{w}_{\text{tar}}, V_{\text{tar}}) \propto \mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{\Lambda})\, \delta\left(V_{\text{tar}} - \mathbf{w}_{\text{tar}}^T \mathbf{x}\right). \tag{3.20}$$

where the target membrane potential, V_{tar}, is Laplacian distributed,

$$P(V_{\text{tar}}) = \frac{e^{-|V_{\text{tar}}|/b}}{2b}. \tag{3.21}$$

We let

$$b^2 = \frac{\mathbf{w}_{\text{tar}}^T \mathbf{\Lambda} \mathbf{w}_{\text{tar}}}{2}, \tag{3.22}$$

so that moments of $\mathbf{x}$ are the same whether we draw from the full distribution (Eq. (3.20)) or just from the Gaussian, $\mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{\Lambda})$ — as is easy to show by direct calculation. While our theory does not require it, in simulations, we use whitened input, i.e., a diagonal input covariance, to match, for instance, the whitened input from retina to V1,

$$\Lambda_{ij} = \delta_{ij} \nu_i \Delta t. \tag{3.23}$$

The diagonal elements are chosen to match the variance expected from a Poisson

process.

Note that this form allows x_i to be positive or negative. To some extent, this could be remedied by adding an offset to x_i, but considerable work will be needed to write down biologically realistic models for $P(\mathbf{x}|\mathbf{w}_{\text{tar}}, V_{\text{tar}})$ in which Bayesian inference can be performed.

Parameter settings

Parameter	Value	Basis
μ_{prior}	-0.669	Matched to data from Song et al. (2005) (Sec 3.5)
σ^2_{prior}	0.863	Matched to data from Song et al. (2005) (Sec 3.5)
n (sup., unsup.)	1000	Offers a good trade-off between biological realism (Binzegger et al., 2004) and computational tractability
n (reinforcment)	100	Uses a reduced number of synapses for reinforcement learning because of the increased difficulty of the learning problem
τ (supervised)	10^5	Supplementary Information, Sec. 3.6; corresponds to 1,000 s
τ (unsup., rein.)	10^6	Supplementary Information, Sec. 3.6; corresponds to 10,000 s
Δt	10 ms	Typical membrane time constant (Tripathy et al., 2015)
γ_V	1 mV	Small value because once the effects of stochastic vesicle release are excluded, membrane potential variability is thought to be small (Bryant and Segundo, 1976; Mainen and Sejnowski, 1995)
γ_δ	1 mV	This is difficult to determine, so we use a small nominal value for computational tractability
k	0.0877	Matched to data from Song et al. (2005) (Sec. 3.5)
θ	0	Sec. 3.5.

Inference when there is a feedback signal

Here we outline how a synapse can infer a distribution over the log of its target weight, $\lambda_{\text{tar},i}$, using all past data. We focus on supervised and reinforcement learning, for which there is a feedback signal, as it is relatively straightforward; we analyze unsupervised learning, for which there is no feedback signal, in Supplementary Information (see in particular Sec. 3.6).

As our model is in a well-understood class, hidden Markov models (HMMs), this inference process is straightforward: we use the standard, two-step procedure for inference in HMMs. In the first step the synapse incorporates new data using Bayes theorem. The data in one time step, denoted d_i, includes the presynaptic input, x_i, the feedback signal, f, the cell's membrane potential, V, and the actual PSP amplitude, w_i,

$$d_i(t) \equiv (x_i(t), f(t), V(t), w_i(t)) \,, \tag{3.24}$$

and we use $\mathcal{D}_i(t)$ to denote all past data,

$$\mathcal{D}_i(t) \equiv (d_i(t), d_i(t-1), \ldots) \,. \tag{3.25}$$

Using this notation, we have

$$P(\lambda_{\text{tar},i}|\mathcal{D}_i) = P(\lambda_{\text{tar},i}|d_i, \mathcal{D}_i(t-1)) \propto P(d_i|\lambda_{\text{tar},i})\, P(\lambda_{\text{tar},i}|\mathcal{D}_i(t-1)) \,. \tag{3.26}$$

To reduce clutter, here and in what follows all quantities without an explicitly specified time index are evaluated at time step t; so, for instance, $w_{\text{tar},i} \equiv w_{\text{tar},i}(t)$ and $\mathcal{D}_i \equiv \mathcal{D}_i(t)$.

In the second step, the synapse takes into account random changes in the target weight,

$$P(\lambda_{\text{tar},i}(t+1)|\mathcal{D}_i) = \int d\lambda_{\text{tar},i} P(\lambda_{\text{tar},i}(t+1)|\lambda_{\text{tar},i})\, P(\lambda_{\text{tar},i}|\mathcal{D}_i) \,. \tag{3.27}$$

Combining both steps takes us from the distribution at time t, $P(\lambda_{\text{tar},i}(t)|\mathcal{D}_i(t-1))$, to the distribution at the time $t+1$, $P(\lambda_{\text{tar},i}(t+1)|\mathcal{D}_i(t))$.

Equations (3.26) and (3.27) tell us how to make exact updates to the distribution over the target weight. However, the exact distribution is too complex for a synapse to work with, let alone store. To simply the problem faced by the synapse, we specify a family of approximate distributions: a Gaussian in the log-domain, with mean μ_i and variance σ_i^2,

$$P(\lambda_{\text{tar},i}|\mathcal{D}(t-1)) = \mathcal{N}\left(\lambda_{\text{tar},i}; \mu_i, \sigma_i^2\right) . \tag{3.28}$$

The corresponding mean, m_i, and variance, s_i^2, of the distribution over $w_{\text{tar},i}$ are

$$m_i \equiv \mathrm{E}\left[w_{\text{tar},i}|\mathcal{D}(t-1)\right] = e^{\mu_i+\sigma_i^2/2}, \tag{3.29a}$$

$$s_i^2 \equiv \mathrm{Var}\left[w_{\text{tar},i}|\mathcal{D}(t-1)\right] = \left(e^{\sigma_i^2}-1\right) m_i^2 \approx \sigma_i^2 m_i^2, \tag{3.29b}$$

the latter valid in the limit $\sigma_i^2 \ll 1$. This is, in fact, a good approximation: on average, $s_i^2/m_i^2 \approx 0.076$ (Supplementary Information, Eq. (3.108c)); combining this with Eq. (3.29b) gives, again on average, $\sigma_i \approx 0.073$. We thus use it throughout most of our analysis.

This approximate distribution has two advantages. First, log-normal distributions always give positive values, leading to learning rules that cannot, for instance, take an excitatory synapse and turn it inhibitory. Second, if the synapse is not given any data, then the dynamics (Equation (3.13)) imply that the distribution over $\lambda_{\text{tar},i}$ approaches a Gaussian at long times — exactly our approximating distribution.

As we will see below, the likelihood, $P(d_i|\lambda_{\text{tar},i})$ is typically not Gaussian in $\lambda_{\text{tar},i}$; consequently, even if $P(\lambda_{\text{tar},i}|\mathcal{D}_i(t-1))$ is Gaussian, $P(\lambda_{\text{tar},i}|\mathcal{D}_i)$ will not be (see Eq. (3.26)). A natural way to remedy this is Assumed Density Filtering (ADF) (Minka, 2001). Formally, this requires us to find the log-normal distribution with the smallest KL-divergence; this can be achieved by matching moments,

$$\mu_i(t+1) = \mathrm{E}\left[\lambda_{\text{tar},i}(t+1)|\mathcal{D}_i\right] \tag{3.30a}$$

$$\sigma_i^2(t+1) = \mathrm{Var}\left[\lambda_{\text{tar},i}(t+1)|\mathcal{D}_i\right]. \tag{3.30b}$$

The central difficulty is computing moments of the inferred distribution, which will require further approximations beyond the assumed density filter. This is dealt with in more depth in Supplementary Information, Secs. 3.6 and 3.6; see in particular Eq. (3.52).

To summarise our model for a single synapse, we can write down a dependency graph describing how each variable is generated (see Fig. 3.8). This is a graphical model – a compact method for describing dependencies among random variables. This graphical model has the extremely unusual feature that the results of inference at one time step influence the data at subsequent time steps.

Problems with inference at the cellular level

Our strategy of performing Bayesian inference at the level of the synapse is actually quite unusual (and is potentially the most important theoretical advance in the paper). The more typical approach is to perform some type of inference

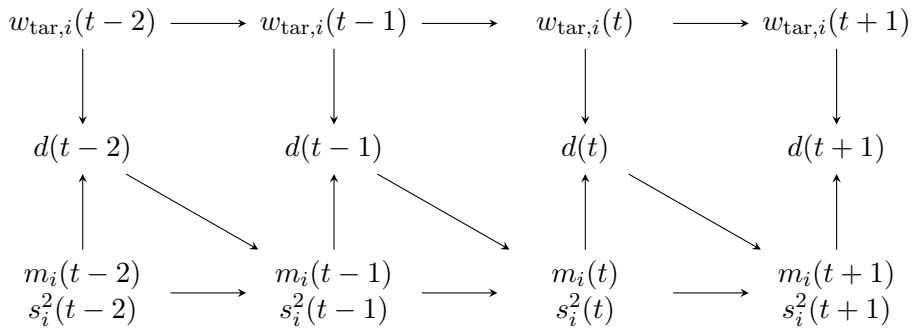

Figure 3.8: A graph describing the dependencies in our simulations. The target weight, $w_{\text{tar},i}(t)$ evolves independently of all other variables, under the exponentiated Ornstein-Uhlenbeck process described in Eq. (3.13). The data, $d_i(t)$, which includes the feedback signal, $f(t)$, the presynaptic input, $x_i(t)$, the postsynaptic activity $V(t)$ (see Eq. (3.24)), and the PSP amplitude, $w_i(t)$, depends on both the target weight, $w_{\text{tar},i}(t)$, and on past inferences, $m_i(t)$ and $s_i^2(t)$. In particular, the feedback signal, $f(t)$, depends on the target weight, and the PSP amplitude, $w_i(t)$, depends on the mean estimate of the target weight, $m_i(t)$ (see Eq. (3.11)). Finally, the mean and uncertainty at time t, $m_i(t)$ and $s_i(t)$, depend on the mean and uncertainty at the previous time step, $m_i(t-1)$ and $s_i^2(t-1)$, and also on past data, $d_i(t-1)$, through the learning rules, Eq. (3.5).

at the level of the whole cell (i.e., infer all the weights jointly). We chose our approach because it is unlikely that synapses can communicate much information to each other. The lack of communication is not a problem if we consider each synapse as performing an inference problem, conditioned on the data available to it. However, it is a problem if inference is performed at the cellular level. To illustrate this in the simplest possible context, we consider a cell with two synapses. Synapses are trying to infer their target weights based on the data, d_1 and d_2, available at synapse 1 and 2, respectively. Without communication, the best each synapse can do is to compute its target weight, based on its data, $P(w_{\text{tar},1}|d_1)$ and $P(w_{\text{tar},2}|d_2)$. However, if we try to infer both weights at the cellular level, then even making the strong approximation that the distribution over each target weight is independent,

$$P(w_{\text{tar},1}, w_{\text{tar},2}|d_1, d_2) \approx P(w_{\text{tar},1}|d_1, d_2)\, P(w_{\text{tar},2}|d_1, d_2)\,, \tag{3.31}$$

we cannot prevent each synapse from "seeing" all the data (except in the unlikely event that d_1 really gives no information about $w_{\text{tar},2}$ and *vice-versa*).

It may seem highly suboptimal for each synapse to perform inference independently, as synapses have to throw away information (for instance, $w_{\text{tar},1}$ must average over its prior uncertainty in d_2, and, likewise, $w_{\text{tar},2}$ must average over its prior uncertainty in d_1). However, from a biological point of view it is quite natural. Nonetheless, this is an unusual approach, and considerable further work is necessary to understand its theoretical properties.

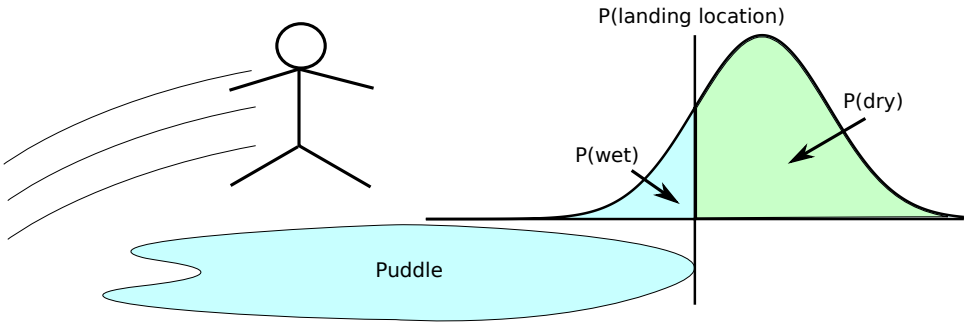

Figure 3.9: A schematic diagram of a stick-person jumping over a puddle. The probability of landing in the puddle, $P(\text{wet})$, depends not only on the mean estimate, but also on the uncertainty.

Details of simulations

We performed two sets of simulations, the first, for Bayesian Plasticity, with k_i fixed at $k = 0.0877$ (Figs. 3.3-3.5), and the second, for Synaptic Sampling, with $k_i = s_i^2/m_i$ (Figs. 3.6 and 3.7) (see Sec. 3.5).

To reduce the variability in the MSE (mean squared error) estimates, for both Bayesian and classical learning rules we ran all simulations using the same inputs, x_i, and target weights, $w_{\text{tar},i}$. We repeated the protocol 24 times, with different inputs and target weights. Using the same inputs and target weights reduced the variability in MSE measurements between learning rates below what might be expected based on the 2 s.e.m. error bars in Fig. 3.4.

To avoid error bars on the MSE that were larger than the mean (something that makes little sense, as the MSE is non-negative), we computed means and standard deviations in the log-MSE domain, which does not have a zero lower-bound, and then mapped back to the linear domain.

Synaptic Sampling

Here we provide an expanded normative argument for Synpatic Sampling. The argument starts with the observation that to select the correct action, knowing the uncertainty in task relevant quantities is critical (Ernst and Banks, 2002). For instance, to decide whether you can jump over a puddle without getting your feet wet, it is important to have not only an estimate of mean landing location, but also the uncertainty in that estimate (Fig. 3.9). Uncertainty about the landing location comes from two sources, uncertainty about the current state of the world and uncertainty about the target weights (i.e. the weights that would give the best estimate of landing location). To see how the brain might compute uncertainty in landing location, we consider a simplified scenario in which we use $\mathbf{x}_{\text{tar}}$ to denote the best possible spike-based representation of the true state of

the external world. The neuron's estimate of landing location is a function of the neuron's output, V, so the optimal estimate of landing location is given by the target output,

$$V_{\text{tar}} = \mathbf{w}_{\text{tar}} \cdot \mathbf{x}_{\text{tar}} + \text{noise}, \tag{3.32}$$

where the noise represents the small amount of uncertainty about landing location that remains when $\mathbf{w}_{\text{tar}}$ and $\mathbf{x}_{\text{tar}}$ are known precisely. Note that the assumption that the synapse combines $\mathbf{w}_{\text{tar}}$ and $\mathbf{x}_{\text{tar}}$ via a dot product is for simplicity only; the cell could use any nonlinear relationship and our arguments would hold.

Of course, the brain knows neither the target weights, $\mathbf{w}_{\text{tar}}$, nor the true state of the external world, $\mathbf{x}_{\text{tar}}$. The brain could compute a "best guess" of $\mathbf{x}_{\text{tar}}$, and the neuron could use a "best guess" of $\mathbf{w}_{\text{tar}}$, resulting in

$$V_{\text{best guess}} = \mathbf{w}_{\text{best guess}} \cdot \mathbf{x}_{\text{best guess}} + \text{noise}. \tag{3.33}$$

However, this scheme is unable to give an estimate of uncertainty — so offers little guidance as to whether or not you should jump over the puddle.

To get an estimate of uncertainty, it is necessary to account for uncertainty both in the state of the world, $\mathbf{x}_{\text{tar}}$, and in the relationship between the state of the world and jump distance, parameterised by $\mathbf{w}_{\text{tar}}$. As information about $\mathbf{x}_{\text{tar}}$ comes from sensory data, and information about $\mathbf{w}_{\text{tar}}$ comes from training data (e.g., from past jumps), we can represent our (probabilistic) knowledge about these quantities as two distributions, $P(\mathbf{x}_{\text{tar}}|\text{Sensory Data})$ and $P(\mathbf{w}_{\text{tar}}|\text{Training Data})$. To combine these distributions into a distribution over V_{tar}, we need to integrate over all possible settings of $\mathbf{x}_{\text{tar}}$ and $\mathbf{w}_{\text{tar}}$,

$$P(V_{\text{tar}}|\text{Sensory Data}, \text{Training Data}) = \tag{3.34}$$
$$\int \mathbf{dw}_{\text{tar}}\ \mathbf{dx}_{\text{tar}}\ P(V_{\text{tar}}|\mathbf{x}_{\text{tar}}, \mathbf{w}_{\text{tar}})\, P(\mathbf{x}_{\text{tar}}|\text{Sensory Data})\, P(\mathbf{w}_{\text{tar}}|\text{Training Data})\,.$$

It is difficult for neurons to compute this distribution directly (as that would involve a complicated high-dimensional integral). However, by combining neural and synaptic sampling, it is possible for neural circuits to evaluate the integral via sampling; that is, by drawing samples, V, from the distribution,

$$V \sim P(V_{\text{tar}}|\text{Sensory Data}, \text{Training Data})\,. \tag{3.35}$$

To do that, we simply need to set neural activity, $\mathbf{x}$, to a pattern that represents a plausible state of the world,

$$\mathbf{x} \sim P(\mathbf{x}_{\text{tar}}|\text{Sensory Data})\,, \tag{3.36}$$

(this is known as the neural sampling hypothesis (Hoyer and Hyvarinen, 2003; Fiser et al., 2010; Berkes et al., 2011a)), and set the synaptic weights, $\mathbf{w}$, to values that represent a plausible setting for the value of the target weights (this is our hypothesis, Synaptic Sampling),

$$\mathbf{w} \sim P(\mathbf{w}_{\text{tar}}|\text{Training Data})\,. \tag{3.37}$$

A sample of landing location is given by combining the sampled inputs and the sampled weights, which could be done by a single neuron,

$$V = \mathbf{w} \cdot \mathbf{x} + \text{noise}. \tag{3.38}$$

Thus, simply by drawing repeated samples, a single neuron can estimate uncertainty about V, and thus about landing location.

Our argument appears to assume that the brain uses the output of a single neuron to make predictions. This is not too implausible — the cerebellum does contain a large number of Purkinje cells (Dean et al., 2010) that are believed to use supervised learning to, among other things, make predictions (though perhaps not about landing location). However, it is certainly possible that such a computation is performed by a large multi-layer network. As long as that network is effectively feedforward, we can still, by the logic described above, estimate its uncertainty by combining synaptic sampling with the sampling hypothesis.

Firing rate data

To obtain the p-value for Fig. 3.7A, we performed standard linear regression: we regressed log(variance/mean) against log(firing rate) and log(mean); the former to test our prediction and the latter to eliminate the PSP amplitude as a possible confound. To estimate the firing rate, we took the mean of a FOOPSI-based firing rate estimate (Vogelstein et al., 2010) computed by the authors of (Ko et al., 2011). This estimate is proportional to the true firing rate, with a constant of proportionality that differs from one (Packer et al., 2015); because our predicted relationship was linear on a log-log plot, the constant of proportionality plays no role. Using this approach, the best fit line was statistically significantly different from zero ($p < 0.003$), and its slope, -0.62 was not significantly different from our prediction, $-1/2$ ($p = 0.57$).

However, there are multiple ways to estimate the firing rate from Calcium traces, and it is not clear a-priori which is most sensible. Thus, we also tried estimating the firing rate using the number of times the FOOPSI signal was above a threshold of 0.01 (we checked that this was a sensible threshold by plotting histograms of the FOOPSI signal). This approach also gave a significant slope ($p < 0.008$, and

the best fit-line, which had a slope of -1.05, was not significantly different from our prediction of $-1/2$ ($p = 0.16$).

3.6 Supplementary Information

Here we give detailed derivations for our learning rules and predictions. In Sec. 3.6 we derive Bayesian learning rules for supervised and reinforcement learning, for which a feedback signal is present, including the simplified learning rules used in Eq. (3.5) of the main text; in Sec. 3.6 we derive Bayesian learning rules for unsupervised learning. We then discuss how to set s_δ^2 (Sec. 3.6) and consider how to relate firing rates to uncertainty (Sec. 3.6). Finally, we provide a detailed description of how we set model parameters (Sec. 3.6).

Learning rules with feedback

We begin by considering the standard classical learning rules that we use for comparison with our Bayesian learning rules; we then move on to the derivation of the Bayesian learning rules themselves.

Classical learning rules

To make comparisons in Fig. 3.4, we need to specify classical learning rules for each type of learning. Each classical rule has a learning rate, α, which is allowed to vary.

For supervised learning with continuous feedback and for supervised learning with binary feedback, we use the delta rule, (Eq. (3.4)) Widrow and Hoff (1960). The delta rule is suitable for binary feedback because we set the threshold, θ, to 0, so the proportion of positive and negative increments is the same.

For reinforcement learning, we use a standard policy gradient method (Williams, 1992),

$$\Delta w_i = -\alpha x_i \left(f - \mathrm{E}\left[f\right]\right)\left(w_i - m_i\right). \tag{3.39}$$

We compute the expected loss, $\mathrm{E}\left[f\right]$ is over past trials, using an exponential moving average. To implement this moving average, on each timestep we updated $\mathrm{E}\left[f\right]$ via,

$$\Delta \mathrm{E}\left[f\right] = \alpha_{\text{reward}}\left(f - \mathrm{E}\left[f\right]\right). \tag{3.40}$$

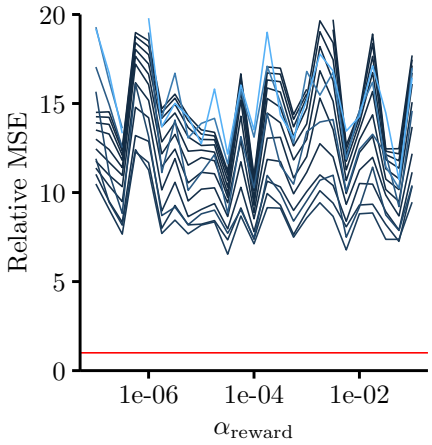

Figure 3.10: The mean squared error relative to the Bayesian learning rules (as in Fig. 3.4) for classical reinforcement learning rules with different settings of α_{reward} (on the x-axis) and with different settings for the learning rate, α (blue lines). The red line is set at a relative MSE of 1. While the relative MSE does not change much with α_{reward}, it does seem that values are more reliable between 10^{-4} to 10^{-6}, as we might expect given that the time constant in this simulation is 10^5. We thus chose $\alpha_{\text{reward}} = 10^{-5}$ for Fig. 3.4. We do not see much change in the relative MSE as we change α_{reward}, because the method asymptotically finds the correct weights even if $\mathrm{E}[f]$ is not set correctly; setting $\mathrm{E}[f]$ correctly merely minimises variance in the weight updates.

A sweep across different settings of α_{reward} (Fig. 3.10) indicated that a sensible value was $\alpha_{\text{reward}} = 10^{-5}$. However, the precise value is not so critical, as the mean squared error was relatively flat over a broad range.

Bayesian learning rules

Here we derive the update rules for the mean and variance, μ_i and σ_i^2. We begin with the difficult part: incorporating new data using Bayes theorem, Eq. (3.26). It is convenient to write the update rule as an integral over the prediction error, δ,

$$P(\lambda_{\text{tar},i}|\mathcal{D}_i) = \int d\delta P\left(\lambda_{\text{tar},i}|\delta, d_i', \mathcal{D}_i(t-1)\right) P\left(\delta|f, d_i'\right) \tag{3.41}$$

where d_i' is all the data except the feedback signal (see Methods, Eq. (3.24)),

$$d_i' = (x_i, V, w_i)\,. \tag{3.42}$$

We have not conditioned on $\mathcal{D}_i(t-1)$ in the last term in Eq. (3.41) because δ is independent of past data, and, recall, quantities without an explicit time dependence should be evaluated at time t. This approach makes it considerably easier to generalise across feedback signals, as $P(\lambda_{\text{tar},i}|\delta, d_i', \mathcal{D}_i(t-1))$ is the same across all feedback signals; only $P(\delta|f, d_i')$ differs.

We start by considering how to infer $\lambda_{\text{tar},i}$ from δ (i.e., how to compute the first term in the integral in Eq. (3.41)). As usual, we use Bayes theorem,

$$P\left(\lambda_{\text{tar},i}|\delta, d_i', \mathcal{D}_i(t-1)\right) \propto P\left(\delta|\lambda_{\text{tar},i}, d_i'\right) P\left(d_i'|\lambda_{\text{tar},i}\right) P(\lambda_{\text{tar},i}|\mathcal{D}_i(t-1)) \,. \quad (3.43)$$

This is the analog of Eq. (3.26); the only difference is that d_i in that equation has been replaced by (δ, d_i'), and we have performed a small amount of algebra. The second term, $P(d_i'|\lambda_{\text{tar},i})$, can be neglected as it is independent of $\lambda_{\text{tar},i}$ (without a feedback signal, d_i' tells us nothing about the target weight). The last term, the prior, $P(\lambda_{\text{tar},i}|\mathcal{D}_i(t-1))$, is given by the approximating Gaussian distribution from the previous time-step (Eq. (3.28)). We will obtain the constant of proportionality in Eq. (3.43) automatically, when we identify the distribution as a Gaussian.

To find an expression for the first term in Eq. (3.43), the likelihood, $P(\delta|\lambda_{\text{tar},i}, d_i')$, we note that δ is the sum of a large number of independent terms, and so, via the central limit theorem, it is Gaussian. Its mean is given by

$$\mathrm{E}\left[\delta|\lambda_{\text{tar},i}, d_i'\right] = \mathrm{E}\left[V_{\text{tar}} - V|\lambda_{\text{tar},i}, w_i\right] = \sum_j x_j \mathrm{E}\left[w_{\text{tar},j} - w_j|\lambda_{\text{tar},i}, w_i\right] \quad (3.44)$$

where the second expression follows from Eqs. (3.1) and (3.2). To evaluate the expectation, we note that for $j \neq i$, $\mathrm{E}\left[w_{\text{tar},j} - w_j|\lambda_{\text{tar},i}, w_i\right] = 0$, leaving only the ith term. Using also the fact that $w_{\text{tar},i} = \pm e^{\lambda_{\text{tar},i}}$ (positive if $w_{\text{tar},i}$ is an excitatory weight and negative if it is inhibitory), we have

$$\mathrm{E}\left[\delta|\lambda_{\text{tar},i}, d_i'\right] = x_i \left(\pm e^{\lambda_{\text{tar},i}} - w_i\right). \quad (3.45)$$

Next we compute the variance of δ. If we assume that all the inputs, $\mathbf{x}$, are known (we relax this assumption shortly), then

$$\begin{aligned} \mathrm{Var}\left[\delta|\lambda_{\text{tar},i}, V, w_i, \mathbf{x}\right] &= \gamma_\delta^2 + \mathrm{Var}\left[V_{\text{tar}} - V|\lambda_{\text{tar},i}, w_i\right]. \quad (3.46)\\ &= \gamma_\delta^2 + \gamma_V^2 + \sum_j \mathrm{Var}\left[w_{\text{tar},j} - w_j|\lambda_{\text{tar},i}, w_i\right] x_j^2. \end{aligned}$$

where again the second expression followed from Eqs. (3.1) and (3.2). Noting that the variance of $w_{\text{tar},j}$ is s_j^2 (Eq. (3.29b)), and that the noise variance in w_j is $k_j m_j$ (Eq. (3.11)), this becomes,

$$\mathrm{Var}\left[\delta|\lambda_{\text{tar},i}, d_i', \mathbf{x}\right] = s_\delta^2 - \left(s_i^2 + k_j m_i\right) x_i^2 \quad (3.47)$$

where

$$s_\delta^2 \equiv \gamma_\delta^2 + \gamma_V^2 + \sum_j \left(s_j^2 + k_j m_j\right) x_j^2. \quad (3.48)$$

Because all the dependence on the x_i is through s_δ^2, we can relax the assumption that all the x_i are known. Instead the synapse only needs to know s_δ^2 for its distribution over δ to be Gaussian,

$$P\left(\delta|\lambda_{\text{tar},i}, d_i', s_\delta^2\right) = \mathcal{N}\left(\delta; x_i\left(\pm e^{\lambda_{\text{tar},i}} - w_i\right), s_\delta^2 - x_i^2\left(s_i^2 + k_i m_i\right)\right). \tag{3.49}$$

Of course, the synapse cannot know s_δ^2, as that involves a summation over all the inputs at every time step. Instead, we use an approximate value based on the average (see Sec. 3.6).

Because of the non-linearity, $e^{\lambda_{\text{tar},i}}$, this is a complicated function of $\lambda_{\text{tar},i}$. We can linearize the problematic term using statistical linearization (Gelb, 1974). This involves finding the straight line that minimizes the expected squared error between the curve and a straight line,

$$0 = \frac{\partial}{\partial a}\mathrm{E}\left[\left(\pm e^{\lambda_{\text{tar}}} - (a(\lambda_{\text{tar}} - \mu) + b)\right)^2\right] \tag{3.50}$$

$$0 = \frac{\partial}{\partial b}\mathrm{E}\left[\left(\pm e^{\lambda_{\text{tar}}} - (a(\lambda_{\text{tar}} - \mu) + b)\right)^2\right], \tag{3.51}$$

where the expectation is taken under the prior ($P(\lambda_{\text{tar},i}|\mathcal{D}(t-1))$). The solution is $a = b = m_i$ (note that m_i is a signed quantity), which gives,

$$\pm e^{\lambda_{\text{tar},i}} \approx m_i\left(1 + \lambda_{\text{tar},i} - \mu_i\right). \tag{3.52}$$

Inserting Eq. (3.52) into Eq. (3.49), the likelihood becomes,

$$P\left(\delta|\lambda_{\text{tar},i}, d_i', s_\delta^2\right) = \exp\left(-\frac{\left(\delta - x_i\left(m_i\left(\lambda_{\text{tar},i} - \mu_i\right) - \left(w_i - m_i\right)\right)\right)^2}{2\left(s_\delta^2 - \left(s_i^2 + k_i m_i\right)x_i^2\right)}\right) \tag{3.53}$$

which is Gaussian in $\lambda_{\text{tar},i}$.

Examining Eq. (3.43) and noting, as discussed immediately after that equation, that the second term on the right hand side is independent of $\lambda_{\text{tar},i}$, we see that to compute the posterior we just need to multiply the likelihood, Eq. (3.53), by $P(\lambda_{\text{tar},i}|\mathcal{D}_i(t-1))$. The latter distribution is also Gaussian in $\lambda_{\text{tar},i}$ (Methods, Eq. (3.28)); consequently, their product is Gaussian. Straightforward, but somewhat tedious, algebra gives us their mean and variance,

$$\mathrm{E}\left[\lambda_{\text{tar},i}|\delta, s_\delta^2, d_i', \mathcal{D}_i(t-1)\right] = \mu_i + \left(\delta + x_i\left(w_i - m_i\right)\right)\frac{x_i m_i \sigma_i^2}{s_{\delta,i}^2} \tag{3.54a}$$

$$\mathrm{Var}\left[\lambda_{\text{tar},i}|\delta, s_\delta^2, d_i', \mathcal{D}_i(t-1)\right] = \sigma_i^2\left(1 - \frac{\sigma_i^2 x_i^2 m_i^2}{s_{\delta,i}^2}\right) \tag{3.54b}$$

where

$$s_{\delta,i}^2 \equiv s_\delta^2 - k_i m_i x_i^2 - \left(s_i^2 - m_i^2 \sigma_i^2\right) x_i^2 \approx s_\delta^2 - k_i m_i x_i^2. \tag{3.55}$$

The approximation is valid so long as $\sigma_i^2 \ll 1$ (see Methods, Eq. (3.29b)).

The next step is to substitute $P(\lambda_{\text{tar},i}|\delta, d_i', \mathcal{D}_i(t-1))$ (which is, to reiterate, Gaussian, with mean and variance given by Eq. (3.54)) back into Eq. (3.41) and perform the integral over δ. Once we do that, we need to take into account changes to the optimal weight across time (Methods, Eq. (3.13)), and then bring the resulting distribution back into the log normal class (Methods, Eq. (3.28)), by computing the mean and variance of $\lambda_{\text{tar},i}$. Fortunately, as is not hard to show, the above two steps commute: we can compute the mean and variance of $\lambda_{\text{tar},i}$ first, and then take into account changes in the optimal weight across time. As is also straightforward to show, the mean and variance are given by

$$\mathrm{E}\left[\lambda_{\text{tar},i}|\mathcal{D}_i\right] = \mu_i + \left(\mathrm{E}\left[\delta|d_i\right] + x_i\left(w_i - m_i\right)\right)\frac{x_i m_i \sigma_i^2}{s_{\delta,i}^2} \tag{3.56a}$$

$$\mathrm{Var}\left[\lambda_{\text{tar},i}|\mathcal{D}_i\right] = \sigma_i^2 - \frac{s_{\delta,i}^2 - \mathrm{Var}\left[\delta|d_i\right]}{s_{\delta,i}^2}\frac{\sigma_i^2 x_i^2 m_i^2}{s_{\delta,i}^2}, \tag{3.56b}$$

where the expectation and variance are with respect to $P(\delta|d_i)$, and, recall, d_i now includes the feedback signal, f (see Eq. (3.24)).

To account for the random changes in weights between time steps we use Eq. (3.14),

$$\mu_i(t+1) = \left(1 - \frac{1}{\tau}\right)\mathrm{E}\left[\lambda_{\text{tar},i}|\mathcal{D}_i\right] + \frac{\mu_{\text{prior}}}{\tau} \tag{3.57a}$$

$$\sigma_i^2(t+1) = \left(1 - \frac{1}{\tau}\right)^2 \mathrm{Var}\left[\lambda_{\text{tar},i}|\mathcal{D}_i\right] + \frac{2\sigma_{\text{prior}}^2}{\tau}. \tag{3.57b}$$

Substituting Eq. (3.56) into Eq. (3.57), and using the fact that the updates to the mean and uncertainty are small on each time step,

$$\left|\mathrm{E}\left[\lambda_{\text{tar},i}|\mathcal{D}_i\right] - \mu_i\right| \ll \mu_i \tag{3.58a}$$

$$\left|\mathrm{Var}\left[\lambda_{\text{tar},i}|\mathcal{D}_i\right] - \sigma_i^2\right| \ll \sigma_i^2, \tag{3.58b}$$

and also using the fact that $\tau \gg 1$, we have

$$\Delta\mu_i = \left(\frac{m_i\sigma_i^2}{s_{\delta,i}^2}\right) x_i\left(\mathrm{E}\left[\delta|d_i\right] + x_i\left(w_i - m_i\right)\right) - \frac{1}{\tau}\left(\mu_i - \mu_{\text{prior}}\right), \tag{3.59a}$$

$$\Delta\sigma_i^2 = -\left(\frac{\sigma_i^4 m_i^2}{s_{\delta,i}^2}\right) x_i^2 \left(\frac{s_{\delta,i}^2 - \mathrm{Var}\left[\delta|d_i\right]}{s_\delta^2}\right) - \frac{2}{\tau}\left(\sigma_i^2 - \sigma_{\text{prior}}^2\right). \tag{3.59b}$$

Finally, to compute the mean and variance of δ conditioned on the data, d_i, we need to compute $P(\delta|d_i)$. We again use Bayes theorem,

$$P(\delta|d_i) = P(\delta|f, x_i, w_i) \propto P(f|\delta)\, P(\delta|x_i, w_i) \tag{3.60}$$

where the prior is given by multiplying the right hand side of Eq. (3.53) by $P(\lambda_{\text{tar},i}|\mathcal{D}_i(t-1))$ (which is Gaussian in $\lambda_{\text{tar},i}$; Methods, Eq. (3.28)), and integrating over $\lambda_{\text{tar},i}$; this leads to

$$P(\delta|x_i, w_i) = \mathcal{N}\left(\delta; -x_i\,(w_i - m_i)\,, s^2_{\delta,i}\right). \tag{3.61}$$

The likelihood, $P(f|\delta)$, is specific to the feedback signal, and hence to the type of learning, as described below. For supervised learning with continuous feedback, the likelihood is a delta function,

$$P(f|\delta) = \delta(\delta - f), \tag{3.62}$$

so the posterior over δ (Eq. (3.60)) is a delta function located at f.

For supervised learning with binary feedback, the likelihood is a step function,

$$P(f = 1|\delta) = \Theta\,(\delta - \theta) \tag{3.63a}$$

$$P(f = -1|\delta) = 1 - \Theta\,(\delta - \theta) \tag{3.63b}$$

so the posterior over δ (Eq. (3.60)) is a truncated Gaussian, whose mean and variance can be computed in terms of the cumulative Normal function. We do not reproduce the expressions here, because they are not very illuminating.

For reinforcement learning, the likelihood is

$$P(f|\delta) = \delta\,(f + |\delta|) \tag{3.64}$$

so the posterior over δ (Eq. (3.60)) is a pair of delta-functions, with different weights, whose mean and variance are easy to compute. Again we do not reproduce those expressions because they are not very illuminating.

Simplifying the learning rules

While we used the full equations in simulation (Eq. (3.59)), for illustrative purposes we presented simplified learning rules in the main text (Eq. (3.5)), valid for continuous feedback, $f = \delta$. These simplifications involve rather severe approximations; we make them so that we can illustrate the essence of the learning rules in the simplest possible setting. We do not, though, use them in any of our simulations

Using the expressions for m_i given in Eq. (3.29a), and assuming updates are small, we have, to first order in the updates,

$$\Delta m_i = m_i \left(\Delta\mu_i + \tfrac{1}{2}\Delta\sigma_i^2\right) \tag{3.65}$$

Using the fact that σ_i^2 is small compared to m_i^2 (Methods, Eq. (3.29) and surrounding text), and assuming that the relative updates to the mean and uncertainty, $\Delta\mu_i/\mu_i$ and $\Delta\sigma_i^2/\sigma_i^2$, are about the same size, we may approximate this with the first term,

$$\Delta m_i \approx m_i \Delta\mu_i. \tag{3.66}$$

Using the approximate expression for s_i^2 given in Eq. (3.29b), and appling the same reasoning as above, we arrive at an approximate update rule for s_i^2,

$$\Delta s_i^2 \approx m_i^2 \Delta\sigma_i^2. \tag{3.67}$$

Inserting these approximate expressions for Δm_i and Δs_i^2 into Eq. (3.59), noting that for continuous feedback the mean of δ is δ and the variance is zero, again using the approximation $s_i^2 \approx \sigma_i^2 m_i^2$ (Eq. (3.29b)), and neglecting the term $w_i - m_i$ in Eq. (3.59a), we have

$$\Delta m_i \approx \left(\frac{s_i^2}{s_{\delta,i}^2}\right) x_i\delta - \frac{m_i}{\tau}\left(\mu_i - \mu_{\text{prior}}\right), \tag{3.68a}$$

$$\Delta\sigma_i^2 \approx -\left(\frac{s_i^2}{s_{\delta,i}^2}\right) x_i^2 s_i^2 - \frac{2m_i^2}{\tau}\left(\sigma_i^2 - \sigma_{\text{prior}}^2\right). \tag{3.68b}$$

To show that the decay term for the mean is approximately the form given in the main text (Eq. (3.5a)) we use Eq. (3.29a) to write

$$m_i - m_{\text{prior}} = m_i\left(1 - e^{-(\mu_i - \mu_{\text{prior}}) - \frac{1}{2}(\sigma_i^2 - \sigma_{\text{prior}}^2)}\right). \tag{3.69}$$

Taylor expanding and neglecting both σ_i^2 and σ_{prior}^2, we arrive at

$$m_i - m_{\text{prior}} \approx m_i\left(\mu_i - \mu_{\text{prior}}\right). \tag{3.70}$$

To show that the decay term for the variance is in approximately the form given in the main text (Eq. (3.5b)), we use our standard approximation for the variance,

$$s_i^2 - s_{\text{prior}}^2 \approx \sigma_i^2 m_i^2 - \sigma_{\text{prior}}^2 m_{\text{prior}}^2. \tag{3.71}$$

As $\mathrm{E}\left[m_i\right] = m_{\text{prior}}$, we replace m_{prior}^2 with m_i to give the required result.

Bayesian learning rules without feedback

We begin by deriving classical learning rules, which will give some results and intuition that will prove useful for Bayesian learning.

Classical learning rules

For unsupervised learning we use a maximum-likelihood learning rule. For maximum likelihood, there is no notion of separate target weights or membrane potential, so we let $\mathbf{w}_{\text{tar}} \to \mathbf{w}$ and $V_{\text{tar}} \to V$. We use the generative model defined in Methods, Sec. 3.5, wherein V is drawn from a Laplacian (Eq. (3.21)), and $\mathbf{x}$ depends on V through Eq. (3.20). The objective is to alter $\mathbf{w}$ so as to maximize the marginal likelihood, $P(\mathbf{x}|\mathbf{w})$, which is given by integrating out the latent variable, V,

$$P(\mathbf{x}|\mathbf{w}) = \int dV P(V)\, P(\mathbf{x}|V, \mathbf{w})\,. \tag{3.72}$$

The un-normalized version of the distribution $P(\mathbf{x}|V, \mathbf{w})$ is given in Eq. (3.20). To perform the integral over V above we need the normalizer, which depends on V,

$$Z(V) = \int d\mathbf{x}\, \frac{e^{-\mathbf{x}^T \mathbf{\Lambda}^{-1} \mathbf{x}/2}}{\text{Det}(2\pi\mathbf{\Lambda})^{1/2}} \delta\left(V - \mathbf{w}^T\mathbf{x}\right) \tag{3.73}$$

where Det denotes determinant. Using the Fourier transform representation of the delta-function, this becomes

$$\begin{aligned} Z(V) &= \int \frac{dq}{2\pi}\, e^{-iqV} \int d\mathbf{x}\, \frac{e^{-\mathbf{x}^T \mathbf{\Lambda}^{-1} \mathbf{x}/2 + iq\mathbf{w}^T\mathbf{x}}}{\text{Det}(2\pi\mathbf{\Lambda})^{1/2}} \qquad (3.74) \\ &= e^{-V^2/2\mathbf{w}^T\mathbf{\Lambda}\mathbf{w}} \int \frac{dq}{2\pi}\, e^{-(q+iV/\mathbf{w}^T\mathbf{\Lambda}\mathbf{w})^2 \mathbf{w}^T\mathbf{\Lambda}\mathbf{w}/2} \int d\mathbf{x}\, \frac{e^{-(\mathbf{x}^T - iq\mathbf{w}^T\mathbf{\Lambda})\mathbf{\Lambda}^{-1}(\mathbf{x} - iq\mathbf{\Lambda}\mathbf{w})/2}}{\text{Det}(2\pi\mathbf{\Lambda})^{1/2}}\,. \end{aligned}$$

The integrals over $\mathbf{x}$ and q are both Gaussian, and therefore straightforward, yielding

$$Z(V) = \frac{e^{-V^2/2\mathbf{w}^T\mathbf{\Lambda}\mathbf{w}}}{(2\pi\mathbf{w}^T\mathbf{\Lambda}\mathbf{w})^{1/2}}. \tag{3.75}$$

The integral in Eq. (3.72) is now straightforward. Using Eq. (3.21) for $P(V)$, we arrive at

$$P(\mathbf{x}|\mathbf{w}) = \frac{e^{-\mathbf{x}^T \mathbf{\Lambda}^{-1} \mathbf{x}/2}}{\text{Det}(2\pi\mathbf{\Lambda})^{1/2}} \frac{e^{-|\mathbf{w}^T\mathbf{x}|/b}}{2b} (2\pi\mathbf{w}^T\mathbf{\Lambda}\mathbf{w})^{1/2} e^{(\mathbf{w}^T\mathbf{x})^2/2\mathbf{w}^T\mathbf{\Lambda}\mathbf{w}}\,. \tag{3.76}$$

The gradient of the log-likelihood is, therefore, given by

$$\begin{aligned}\frac{\partial \log P(\mathbf{x})}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}}\left[-\frac{\left|\mathbf{w}^T\mathbf{x}\right|}{b} + \frac{\log \mathbf{w}^T\mathbf{\Lambda}\mathbf{w}}{2} + \frac{(\mathbf{w}^T\mathbf{x})^2}{2\mathbf{w}^T\mathbf{\Lambda}\mathbf{w}}\right] \\ &= -\frac{\operatorname{sign}\left(\mathbf{w}^T\mathbf{x}\right)\mathbf{x}}{b} + \frac{\mathbf{\Lambda}\mathbf{w}}{\mathbf{w}^T\mathbf{\Lambda}\mathbf{w}} + \frac{\mathbf{w}^T\mathbf{x}\mathbf{x}}{\mathbf{w}^T\mathbf{\Lambda}\mathbf{w}} - \frac{(\mathbf{w}^T\mathbf{x})^2\mathbf{\Lambda}\mathbf{w}}{(\mathbf{w}^T\mathbf{\Lambda}\mathbf{w})^2}.\end{aligned} \tag{3.77}$$

Using $\mathrm{E}\left[(\mathbf{w}^T\mathbf{x})^2\right] = \mathbf{w}^T\mathrm{E}\left[\mathbf{x}\mathbf{x}\right]\mathbf{w} = \mathbf{w}^T\mathbf{\Lambda}\mathbf{w}$ (see Methods, Eqs. (3.20) and following text), we see that on average the second and fourth terms cancel. Taking that into account and, in a slight abuse of notation replacing $\mathbf{w}^T\mathbf{x}$ with V, we arrive at

$$\frac{\partial \log P(\mathbf{x})}{\partial \mathbf{w}} \approx -\frac{\operatorname{sign}(V)\,\mathbf{x}}{b} + \frac{V\mathbf{x}}{\mathbf{w}^T\mathbf{\Lambda}\mathbf{w}}\,. \tag{3.78}$$

As expected, this learning rule has a classic Hebbian form: increase the weight when V is large, and decrease the weight when V is small.

Bayesian inference

For the Bayesian learning rule, we take exactly the same approach as previously (i.e. using Eq. (3.41)). Just as for the previous learning rules, all we need to do is compute the moments of the posterior distribution over δ, and insert them into the learning rules (Eq. (3.59)). In unsupervised learning, the posterior over δ simplifies considerably, as we do not have a feedback signal, and we throw away information about w_i, x_i,

$$P\left(\delta|f, d_i'\right) = P\left(\delta|d_i'\right) \approx P(\delta|V)\,. \tag{3.79}$$

For unsupervised learning, it turns out to be easier to work in terms of V_{tar} rather than δ. As V_{tar} and δ are related very simply (Eq. (3.3)) and V is known, computing the moments of δ from the moments of V_{tar}, is trivial (we have neglected γ_δ^2 for simplicity),

$$\mathrm{E}\left[\delta|V\right] = \mathrm{E}\left[V_{\mathrm{tar}}|V\right] - V, \tag{3.80a}$$

$$\mathrm{Var}\left[\delta|V\right] = \mathrm{Var}\left[V_{\mathrm{tar}}|V\right]. \tag{3.80b}$$

To compute $P(V_{\mathrm{tar}}|V)$, we use Bayes theorem,

$$P(V_{\mathrm{tar}}|V) \propto P(V_{\mathrm{tar}})\,P(V|V_{\mathrm{tar}}) \tag{3.81}$$

and introduce and integrate out other quantities that appear in the generative model,

$$P(V_{\text{tar}}|V) \propto P(V_{\text{tar}}) \int d\mathbf{x} d\mathbf{w}_{\text{tar}} P(V|\mathbf{x})\, P(\mathbf{x}|V_{\text{tar}}, \mathbf{w}_{\text{tar}})\, P(\mathbf{w}_{\text{tar}})\,. \tag{3.82}$$

To compute $P(\mathbf{x}|V_{\text{tar}}, \mathbf{w}_{\text{tar}})$ we combine the $\mathbf{x}$ dependence of $P(\mathbf{x}|V_{\text{tar}}, \mathbf{w}_{\text{tar}})$ (Eq. (3.20)) with the normalizer (Eq. (3.75)), and noting that the normalizer can be rewritten as a Gaussian, which gives,

$$P(\mathbf{x}|V_{\text{tar}}, \mathbf{w}_{\text{tar}}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{\Lambda})\, \delta\left(V_{\text{tar}} - \mathbf{w}_{\text{tar}}^T \mathbf{x}\right) \mathcal{N}\left(V_{\text{tar}}; 0, \mathbf{w}_{\text{tar}}^T \mathbf{\Lambda} \mathbf{w}_{\text{tar}}\right)^{-1}. \tag{3.83}$$

Now we make an approximation; because $\mathbf{\Lambda}$ is diagonal, $\mathbf{w}_{\text{tar}}^T \mathbf{\Lambda} \mathbf{w}_{\text{tar}}$ is the sum of a large number of non-negative terms. If those terms were independent, $\mathbf{w}_{\text{tar}}^T \mathbf{\Lambda} \mathbf{w}_{\text{tar}}$ would self-average: its standard deviation would be much smaller than its mean. Because of $P(V_{\text{tar}}|\mathbf{w}_{\text{tar}}, \mathbf{x})$, those terms are not quite independent. However, this term has minimal effect on the variance, so it still self averages. Thus, we can use,

$$\mathbf{w}_{\text{tar}}^T \mathbf{\Lambda} \mathbf{w}_{\text{tar}} \approx \Delta t \sum_i \nu_j e^{2\left(\mu_j + \sigma_j^2\right)} \equiv v \tag{3.84}$$

Substituting this into Eq. (3.81) and writing $\delta\left(V_{\text{tar}} - \mathbf{w}_{\text{tar}}^T \mathbf{x}\right)$ as $P(V_{\text{tar}}|\mathbf{w}_{\text{tar}}, \mathbf{x})$, gives,

$$P(V_{\text{tar}}|V) \propto P(V_{\text{tar}})\, \mathcal{N}(V_{\text{tar}}; 0, v)^{-1}\, Q(V, V_{\text{tar}}) \tag{3.85}$$

where

$$Q(V, V_{\text{tar}}) = \int d\mathbf{x} d\mathbf{w}_{\text{tar}} P(V|\mathbf{x})\, P(V_{\text{tar}}|\mathbf{w}_{\text{tar}}, \mathbf{x})\, P(\mathbf{w}_{\text{tar}})\, \mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{\Lambda})\,. \tag{3.86}$$

Integrating over $\mathbf{w}_{\text{tar}}$, we get,

$$Q(V, V_{\text{tar}}) = \int d\mathbf{x} P(V|\mathbf{x})\, P(V_{\text{tar}}|\mathbf{x})\, \mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{\Lambda})\,. \tag{3.87}$$

As V is known and fixed, we only care about the V_{tar} dependence, and so we can also write,

$$P(V_{\text{tar}}|V) \propto P(V_{\text{tar}})\, \mathcal{N}(V_{\text{tar}}; 0, v)^{-1}\, Q(V_{\text{tar}}|V) \tag{3.88}$$

where

$$Q(V_{\text{tar}}|V) \propto Q(V, V_{\text{tar}}) \tag{3.89}$$

Again, as V is known, instead of computing the distribution over $Q(V_{\text{tar}}|V)$ directly, it is easier to compute $Q(\delta|V) = Q(\delta)$, then convert back. The distri-

bution over δ is very simple,

$$Q\left(\delta\right) = \mathcal{N}\left(\delta; 0, s_\delta^2\right) \tag{3.90}$$

(we compute a closely related quantity in Eq. (3.49)). Given the definition of δ (Eq. (3.3)), the corresponding distribution over V_{tar} is

$$Q\left(V_{\text{tar}}|V\right) = \mathcal{N}\left(V_{\text{tar}}; V, s_\delta^2\right). \tag{3.91}$$

Thus, we can compute $P(V_{\text{tar}}|V)$ (Eq. (3.88)) by combining two Gaussian distributions, $Q\left(V_{\text{tar}}|V\right)$ and $\mathcal{N}\left(V_{\text{tar}}; 0, v\right)^{-1}$, with a Laplacian, $P(V_{\text{tar}})$. This gives rise to a mixture of two truncated Gaussian distributions, one for the rising, and one for the decaying part of the Laplacian. Thus, the mean and variance of $P(V_{\text{tar}}|V)$ can straightforwardly (if tediously) be computed – we do not reproduce these expressions here because they are not very enlightening. As described above (Eq. (3.80)), the mean and variance of $P(V_{\text{tar}}|V)$ trivially give the mean and variance of $P(\delta|V)$, which we can be inserted directly into the learning rules (Eq. (3.59)).

Setting s_δ^2

Ideally, s_δ^2 (given in Eq. (3.48)) should be updated on every timestep. In reality, of course, this requires a non-local computation that the synapse is unable to perform. Therefore, for supervised and unsupervised learning, we approximate s_δ^2 using its average value,

$$\mathrm{E}\left[s_\delta^2\right] = \gamma_\delta^2 + \gamma_V^2 + \sum_j \left(s_j^2 + k_j m_j\right) \nu_j \Delta t. \tag{3.92}$$

So long as the firing rates are stationary, this quantity changes slowly. Moreover, s_δ^2 is the same for the whole cell, so could be computed by molecular machinery in the cell (e.g. signalling cascades, tagging proteins, etc.)

For reinforcement learning, however, this approximation turns out to not be good enough. Instead we use a better approximation, and exploit the fact that δ tells us, via Bayes' theorem, something about s_δ^2,

$$P\left(s_\delta^2|\delta\right) \propto P\left(\delta|s_\delta^2\right) P\left(s_\delta^2\right). \tag{3.93}$$

The likelihood, $P\left(\delta|s_\delta^2\right)$, is given by Eq. (3.53), but with all terms in the exponent (except δ) replaced by their means,

$$P\left(\delta|s_\delta^2\right) = \mathcal{N}\left(\delta; 0, s_\delta^2\right). \tag{3.94}$$

For analytic tractability, we set the prior, $P\left(s_\delta^2\right)$, to the appropriate conjugate prior (an Inverse Gamma distribution),

$$P\left(s_\delta^2\right) = \text{InverseGamma}\left(s_\delta^2; \alpha, \beta\right) \propto s_\delta^{-2(\alpha+1)} e^{-\beta/s_\delta^2}. \tag{3.95}$$

To set α and β, we match the mean (Eq. (3.92)) and variance of s_δ^2, the latter given by

$$\text{Var}\left[s_\delta^2\right] = \sum_j \left(s_j^2 + k_j m_j\right) \nu_j \Delta t \left(1 - \nu_j \Delta t\right). \tag{3.96}$$

The mean and variance of an Inverse Gamma distribution are given by,

$$\text{E}\left[s_\delta^2\right] = \frac{\beta}{\alpha - 1} \tag{3.97a}$$

$$\text{Var}\left[s_\delta^2\right] = \frac{\beta^2}{(\alpha-1)^2 (\alpha-2)}. \tag{3.97b}$$

Solving for α and β, we have

$$\alpha = \frac{\text{E}\left[s_\delta^2\right]^2}{\text{Var}\left[s_\delta^2\right]} + 2 \tag{3.98a}$$

$$\beta = \text{E}\left[s_\delta^2\right](\alpha - 1). \tag{3.98b}$$

Substituting the prior and likelihood into Eq. (3.93) gives the posterior,

$$P\left(s_\delta^2|\delta\right) \propto s_\delta^{-2((\alpha+1/2)+1)} e^{-\left(\beta+\delta^2/2\right)/s_\delta^2}. \tag{3.99}$$

Comparing to Eq. (3.95), we see that the posterior is another Inverse Gamma distribution (as expected, given that we use a conjugate prior). Finally, we use the posterior mean, as our estimate of s_δ^2,

$$\text{E}\left[s_\delta^2|\delta\right] = \frac{2\beta + \delta^2}{2\alpha - 1} = \frac{2(\alpha-1)\text{E}\left[s_\delta^2\right] + \delta^2}{2(\alpha-1) \quad + 1}. \tag{3.100}$$

The mean value of s_δ^2 conditioned on δ, $\text{E}\left[s_\delta^2|\delta\right]$, is, therefore, a weighted sum of $\text{E}\left[s_\delta^2\right]$ and δ^2. Because α is large (both the mean and variance of s_δ^2 are proportional to n, the number of synapses, and so both are large; consequently α is also proportional to n), that quantity is weighted heavily toward $\text{E}\left[s_\delta^2\right]$. However, the small contribution from δ turns out to be important; without it, the mean squared error tends to be very large (data not shown).

The relationship between variability and firing rate

We wish to find relationships between the mean and uncertainty, m_i and s_i^2, and the firing rate, ν_i. To do so, we take the time average of Eq. (3.59b) in steady state (where $\langle \Delta\sigma_i^2 \rangle = 0$),

$$0 = \left\langle \frac{x_i \sigma_i^4 m_i^2}{s_\delta^2} \frac{s_\delta^2 - \mathrm{Var}\left[\delta | d_i\right]}{s_\delta^2} \right\rangle - \frac{2}{\tau}\left(\sigma_{\mathrm{prior}}^2 - \left\langle \sigma_i^2 \right\rangle\right) \tag{3.101}$$

Here and in what follows the angle brackets indicate an average over times that are long enough to average over fluctuations but short compared to τ, the timescale over which the target weights change. For tractability, we ignore correlations among the variables; consequently, Eq. (3.101) becomes

$$0 = \frac{\sigma_i^4 m_i^2 \nu_i \Delta t \chi_i}{s_\delta^2} + \frac{2\sigma_i^2}{\tau} - \frac{2\sigma_{\mathrm{prior}}^2}{\tau} \tag{3.102}$$

where we have replaced $\langle x_i \rangle$ with $\nu_i \Delta t$ and made the definition

$$\chi_i \equiv \frac{s_\delta^2 - \langle \mathrm{Var}\left[\delta | d_i\right] \rangle}{s_\delta^2} \,. \tag{3.103}$$

Solving for σ_i^2, we have

$$\sigma_i^2 = \frac{\left(2m_i^2 \nu_i \Delta t \chi_i \sigma_{\mathrm{prior}}^2 / s_\delta^2 \tau + 1/\tau^2\right)^{1/2} - 1/\tau}{m_i^2 \nu_i \Delta t \chi_i / s_\delta^2} \tag{3.104}$$

In the limit that $\tau \nu_i \Delta t \gg 1$, the above expression simplifies considerably,

$$\sigma_i^2 \approx \frac{s_\delta / m_i}{\sqrt{\nu_i \Delta t}} \left(\frac{2\sigma_{\mathrm{prior}}^2}{\tau \chi_i}\right)^{1/2} . \tag{3.105}$$

Using the approximation $s_i^2 \approx \sigma_i^2 m_i^2$, valid so long as $\sigma_i^2 \ll 1$ (see Methods, Eq. (3.29b) and following discussion), we arrive at

$$\frac{s_i^2}{m_i} \approx \frac{s_\delta}{\sqrt{\nu_i \Delta t}} \left(\frac{2\sigma_{\mathrm{prior}}^2}{\tau \chi_i}\right)^{1/2} . \tag{3.106}$$

Assuming the feedback signal typically removes a finite fraction of the prior variance concerning δ, χ_i will be $\mathcal{O}(1)$. Thus, because the relative learning rate, $\Delta m_i / m_i$, is proportional to s_i^2 / m_i (see main text, Eqs. (3.5b) and (3.6)), Eq. (3.106) corroborates our prediction about learning rates via Bayesian Plasticity (main text, Eq. (3.7)).

However, note that the prediction regarding plasticity will not necessarily hold in experiments in which the network does not exhibit ongoing activity. That's

because without ongoing activity, only one input (the stimulated one, say input i) is active. In that case, $s_\delta^2 \propto s_i^2$ (see Eq. (3.48), and note that the noise is small), and so the learning rate, α_i, does not change with s_i^2 (Eq. (3.6)). In contrast, if there are many other inputs active, then there are many other contributions to s_δ^2 (Eq. (3.48)), so s_δ^2 changes little with s_i^2. Because *in vitro* preparations are typically quite, this prediction must be tested *in vivo*.

Setting model parameters

To find a sensible timescale for synaptic sampling (i.e., a timescale upon which the uncertainty is similar to the variability) we solve Eq. (3.106) for τ,

$$\tau \sim \frac{2\sigma_{\text{prior}}^2}{\nu_i \Delta t \frac{s_i^2}{m_i^2} \frac{s_i^2}{s_\delta^2} \chi_i} \tag{3.107}$$

where

$$\nu_i \Delta t \frac{s_i^2}{s_\delta^2} \sim 1/2n \text{ (see Eq. (3.48))} \tag{3.108a}$$

$$n = 1000 \text{ (Methods, Sec. 3.5)} \tag{3.108b}$$

$$\frac{s_i^2}{m_i^2} \sim 0.076 \text{ (Average value from Song et al. (2005))} \tag{3.108c}$$

$$\sigma_{\text{prior}}^2 \sim 0.86 \text{ (from Song et al. (2005))}. \tag{3.108d}$$

We thus have

$$\tau \sim \frac{50,000}{\chi_i}. \tag{3.109}$$

For supervised learning, χ_i is relatively high. We thus use $\chi_i \sim 0.5$, and hence $\tau = 100,000$ timesteps, or $1,000$ s. For unsupervised and reinforcement learning, we used $\tau = 1,000,000$ timesteps or $10,000$ s to account for lower values of χ_i. For reinforcement learning, the problem is so hard (i.e. χ_i is so small) that it was also necessary to use $n = 100$.

Chapter 4

The Hamiltonian brain

4.1 Abstract

Probabilistic inference offers a principled framework for understanding both behaviour and cortical computation. However, two basic and ubiquitous properties of cortical responses seem difficult to reconcile with probabilistic inference: neural activity displays prominent oscillations in response to constant input, and large transient changes in response to stimulus onset. Indeed, cortical models of probabilistic inference have typically either concentrated on tuning curve or receptive field properties and remained agnostic as to the underlying circuit dynamics, or had simplistic dynamics that gave neither oscillations nor transients. Here we show that these dynamical behaviours may in fact be understood as hallmarks of the specific representation and algorithm that the cortex employs to perform probabilistic inference. We demonstrate that a particular family of probabilistic inference algorithms, Hamiltonian Monte Carlo (HMC), naturally maps onto the dynamics of excitatory-inhibitory neural networks. Specifically, we constructed a model of an excitatory-inhibitory circuit in primary visual cortex that performed HMC inference, and thus inherently gave rise to oscillations and transients. These oscillations were not mere epiphenomena but served an important functional role: speeding up inference by rapidly spanning a large volume of state space. Inference thus became an order of magnitude more efficient than in a non-oscillatory variant of the model. In addition, the network matched two specific properties of observed neural dynamics that would otherwise be difficult to account for in the context of probabilistic inference. First, the frequency of oscillations as well as the magnitude of transients increased with the contrast of the image stimulus. Second, excitation and inhibition were balanced, and inhibition lagged excitation. These results suggest a new functional role for the separation of cortical populations into excitatory and inhibitory neurons, and for the neural

oscillations that emerge in such excitatory-inhibitory networks: enhancing the efficiency of cortical computations.

4.2 Introduction

Uncertainty plagues neural computation. For instance, hearing the rustle of an animal at night, it may be impossible to ascertain the species, and thus whether or not it is dangerous. One approach in this scenario is to respond based on a point estimate, usually the single most probable explanation of our observations. However, this leads to a problem: if the probability of the animal being dangerous is below 50%, then the single most probable explanation is that the animal is harmless; and considering only this explanation, and thus failing to respond, could easily prove fatal. Instead, to respond appropriately, it is critical to take uncertainty into account by also considering the possibility of there being a dangerous animal, given the rustle and any other available clues.

The optimal way to perform computations and select actions under uncertainty is to represent a probability distribution that quantifies the probability with which each scenario may describe the actual state of the world, and update this probability distribution according to the laws of probability, i.e. by performing Bayesian inference. Human behaviour is consistent with Bayesian inference in many sensory (Knill, 1998; Jacobs, 1999; van Beers et al., 1999a; Ernst and Banks, 2002), motor (Wolpert et al., 1995; Körding and Wolpert, 2004) and cognitive (Gopnik et al., 2004; Chater et al., 2006; Tenenbaum et al., 2006) tasks. There is also evidence that probabilistic inference is performed already in early sensory cortical areas (Berkes et al., 2011b; Orbán et al., 2016). In particular, simple cells in the primary visual cortex (V1) respond maximally to Gabor filter-like stimuli (i.e. edges), which have been shown to provide the most parsimonious explanation of natural images in probabilistic theories of visual processing (Hyvärinen, 2010) (or mathematically equivalent regularisation-based approaches (Olshausen and Field, 1996)). Furthermore, more complex probabilistic models can account for contrast invariant tuning (Schwartz and Simoncelli, 2001) and complex cell properties (Karklin and Lewicki, 2009), as well as surround-suppression effects in neural data and behaviour (Coen-Cagli et al., 2012).

The apparent success of probabilistic inference in accounting for a diverse set of experimental observations raises the question of how neural systems might represent and compute with uncertainty (Pouget et al., 2013). Nevertheless, traditional models of neural computation ignore uncertainty, and instead rely on circuit dynamics that find the single best explanation for their inputs (Rao and Ballard, 1999; Olshausen and Field, 1996; Deneve et al., 1999). More recent approaches do allow for the representation of uncertainty, including distributional (Zemel et al.,

1998), doubly distributed (Sahani and Dayan, 2003), and probabilistic population codes (Ma et al., 2006; Beck et al., 2008, 2011), or sampling-based network dynamics (Hoyer and Hyvarinen, 2003; Buesing et al., 2011; Orbán et al., 2016). However, none of these previous models capture the rich dynamics of cortical responses. In particular, neural activities in the cortex show prominent intrinsic oscillations (Basar and Guntekin, 2008), and large transient changes in response to stimulus onset, which are observed in V1 (Müller et al., 1999, 2001; Ray and Maunsell, 2010), and other cortical areas (Armstrong and Moore, 2007; Luczak et al., 2013). In contrast, existing neural models of probabilistic inference either have no dynamics and so predict stationary responses to a fixed stimulus, or they have gradient ascent-like dynamics that display neither oscillations nor transients, and eventually also converge to a steady-state response for a fixed input. Moreover, these models typically violate Dale's law, by having neurons with both excitatory and inhibitory outputs. While there have been excitatory-inhibitory (EI) networks models that did capture some of these aspects of cortical dynamics, these have rarely been linked to any particular computation (but see Li and Dayan (1999); Rubin et al. (2015)), let alone probabilistic inference.

Here, we present an EI neural network model of V1 that performs probabilistic inference while retaining a computationally useful representation of uncertainty, and has rich, cortex-like dynamics, including oscillations and transients. In particular, our network uses a sampling-based representation of uncertainty (Hoyer and Hyvarinen, 2003; Fiser et al., 2010; Orbán et al., 2016), such that at any time it represents a single plausible interpretation of the input, and as time passes it sequentially samples many different interpretations. In other words, the network represents the probability of different scenarios implicitly, by the frequency with which it visits their representations via its dynamics. For instance, in the example above, neural activity at one moment would represent "dangerous", then "not dangerous" at some later time, and then "dangerous" again, such that a decision about how to behave can then be made based on the proportion of the time neural activity represents "dangerous" vs. "not dangerous". Thus, a fundamental consequence of a sampling-based representation for neural dynamics is that whenever there is uncertainty, neural activity will not settle down to a single fixed point but instead, it will continue to move between patterns representing the different possible states of the world. More specifically, an efficient sampling-based representation requires this continuous movement across state space to be such that the rate at which (statistically independent) samples are generated by the dynamics is as high as possible. We show that EI networks are ideally suited to achieve efficient sampling by implementing a powerful family of probabilistic inference algorithms, Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal, 2011).

HMC is based on the idea that it is possible to sample from a probability distribu-

tion by setting up a dynamical system whose dynamics is Hamiltonian (Fig. 4.1A). The state of such a system behaves as a particle moving on a (high dimensional) surface, with momentum. The surface determines the potential energy of the particle, corresponding to the negative logarithm of the probability distribution that needs to be sampled (such that high probability states correspond to low potential energy). These dynamics speed up inference because the momentum of the system prevents the random walk behaviour plaguing many other sampling-based inference schemes. In particular, the particle will accelerate as it heads towards the minimum of the potential energy landscape, but once it reaches that point, it will have a large momentum, so it will keep moving out the other side (Fig. 4.1A-D). Our key insight is that HMC dynamics are naturally implemented by the interactions of recurrently coupled excitatory and inhibitory populations in cortical circuits. Due to these interactions, our network possessed inherently oscillatory dynamics. Crucially, these oscillations were ideal for speeding up inference, as they moved rapidly across the state space and hence represented a whole range of plausible interpretations efficiently.

In the following, we first define the statistical model of natural visual scenes that served as the testbed for our simulations of V1 dynamics. We then describe the HMC-based neural network that implemented sampling under this statistical model. We demonstrate that our dynamics sample more rapidly than noisy gradient ascent (also known as Langevin dynamics), and therefore that the presence of oscillations and transients in our network speeds up inference. Next, we show by both theoretical analysis and simulation that our sampler reproduces three properties of experimentally observed cortical dynamics. First, our sampler has balanced excitation and inhibition, with inhibition lagging excitation (Okun and Lampl, 2008). Second, our sampler oscillates, and the oscillation frequency increases with stimulus contrast (Ray and Maunsell, 2010; Roberts et al., 2013). Third, there is a transient increase in firing rates upon stimulus onset, and the magnitude of this transient is also modulated by stimulus contrast (Ray and Maunsell, 2010). Thus, our work provides a principled unifying account of these dynamical motifs by relating them to a fundamental class of cortical computations: probabilistic inference.

4.3 Results

The Gaussian scale mixture model and V1 responses

In order to model the dynamics of V1 responses, we adopted a statistical model that has been widely used to capture the statistics of natural images and consequently to account for the *stationary* responses of V1 neurons in terms of prob-

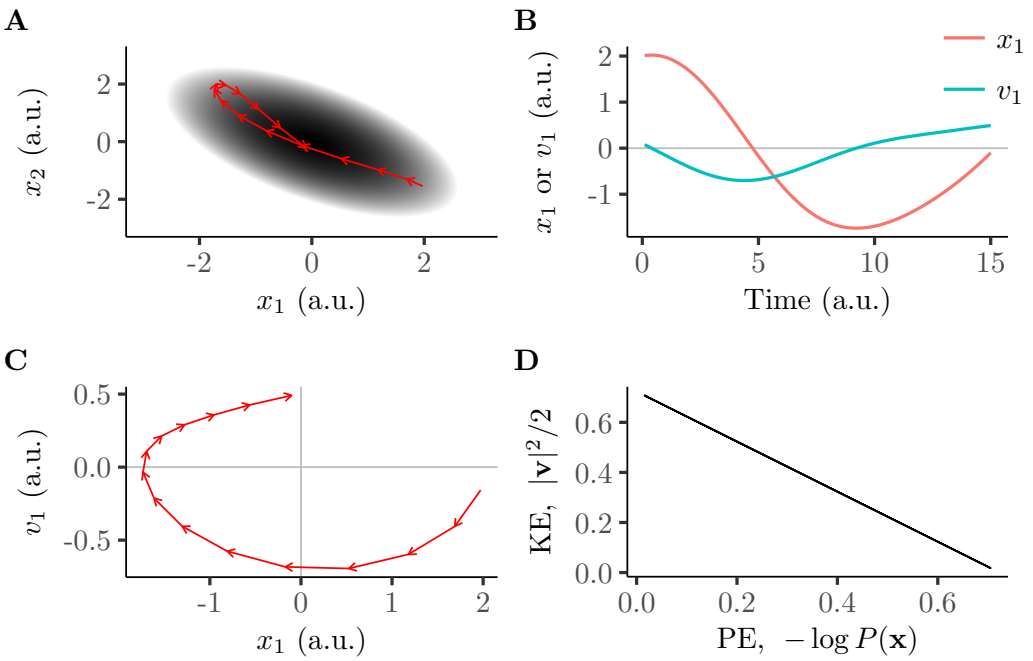

Figure 4.1: An example of Hamiltonian dynamics. **A.** Movement of a particle under Hamiltonian dynamics (i.e. with momentum) on a two-dimensional quadratic potential energy landscape (greyscale, darker means lower energy) corresponding to a multivariate Gaussian probability density. The red arrows show the trajectory, with each arrow representing an equal time interval. Note that the particle does not just go to the lowest potential energy location: it picks up momentum (kinetic energy) as it moves, leading it to oscillate around the energy well. **B.** A plot of position (red) and velocity (blue, the derivative of position) along one dimension. **C.** Plotting velocity and position directly against each other reveals explicitly that the dynamics of the system is similar to that of a harmonic oscillator. **D.** Plotting kinetic energy (KE) against potential energy (PE) reveals an exchange between kinetic energy and potential energy that contributes to the system's oscillatory behaviour.

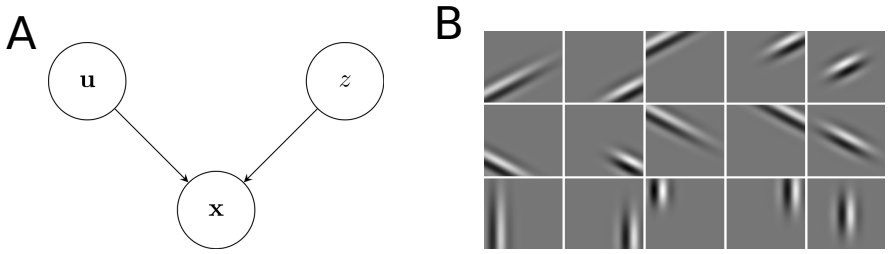

Figure 4.2: A. The graphical model representation of the Gaussian scale mixture model. The distribution over the observations (images), $\mathbf{x}$, depends on two latent variables, z and $\mathbf{u}$. The vector $\mathbf{u}$ represents the intensity of edge-like features (see panel B) in the images. The positive scalar z represents the overall contrast level in the image. **B.** The basis functions represented by $\mathbf{u}$ were 15 Gabor filters centred at five different locations, and with three different orientations.

abilistic inference. We extended this model to account for the *dynamics* of V1 responses.

The Gaussian scale mixture (GSM) model is relatively simple, yet captures some fundamental higher-order statistical properties of natural image patches by introducing latent variables, $\mathbf{u}$, coordinating the linear superposition of simple edge features and an additional latent variable, z, determining the overall contrast level of the image patch (Portilla et al., 2001) (Fig. 4.2A). Formally, the probabilistic generative model can be written as

$$P(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{C}) \tag{4.1}$$

$$P(z) = \mathcal{T}(z; 0, 1, 0) \tag{4.2}$$

$$P(\mathbf{x}|\mathbf{u}, z) = \mathcal{N}\left(\mathbf{x}; z\mathbf{A}\mathbf{u}, \sigma_{\mathrm{x}}^2\mathbf{I}\right) \tag{4.3}$$

where $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, $\mathcal{T}\left(\cdot; \mu, \sigma^2, \theta\right)$ is a truncated (univariate) normal distribution with mean μ and variance σ^2 truncated below threshold θ (so that, in our case, z is non-negative), $\mathbf{x}$ is the grey levels of pixels in an image patch, the columns of $\mathbf{A}$ include the edge-like features whose combinations are used to explain images (Fig. 4.2B), $\mathbf{C}$ describes their prior covariance (which is fitted to whitened data), and $\sigma_{\mathrm{x}}^2 = 0.1$ is the level of noise present in the images. (See Table 4.1 for all parameters in the model, and Methods for details of the procedure used to set them.)

Crucially, assuming that V1 simple cell activities represent values of $\mathbf{u}$ sampled from the posterior over $\mathbf{u}$ given an input $\mathbf{x}$ under the GSM, $P(\mathbf{u}|\mathbf{x})$, provides a natural account for a number of empirical observations. (Conversely, inference of z may provide an account of complex-cell activations (Schwartz et al., 2004; Karklin and Lewicki, 2009; Berkes et al., 2009), which we did not study in further detail here.) In particular, the posterior mean of $\mathbf{u}$, represented by the mean of

Table 4.1: Values of the parameters used in our simulations.

Parameter	Value	Role
$\mathbf{C}$	$\left(1-\sigma_{\mathbf{x}}^2\right)\left(\mathbf{A}^T\mathbf{A}\right)^{-1}$	prior covariance of $\mathbf{u}$
$\mathbf{A}$	See Fig. 4.2B and Methods	edge-detecting filters represented by model neurons
σ_{x}^2	0.1	variance of observation noise
τ	10 ms	membrane time constant
ρ^2	13 s^{-1}	rate at which stochastic vesicle release injects noise
$\mathbf{W}_{\mathrm{uu}}$, $\mathbf{W}_{\mathrm{uv}}$, etc.	See Methods	recurrent connection weights in the network

See Methods for details of the procedure used to determine the parameters. Oscillation frequency in the network was jointly determined by several of these parameters (see Eq. (4.8)), the timescale of transients was mainly determined by ρ (see S1 Figure).

model neuron activities, matches the across-trial average responses of simple cells in V1 (Schwartz and Simoncelli, 2001; Coen-cagli et al., 2009). Moreover, it can also be shown that the posterior variance of $\mathbf{u}$, represented by the variance of model neuron activities, captures important aspects of the across-trial variance of V1 responses (Orbán et al., 2016), namely the quenching of neural variability with stimulus onset (Churchland et al., 2010). This is because, in the no-stimulus condition, we have a blank image, $\mathbf{x} = \mathbf{0}$. Under the GSM, $\mathbf{x} \approx z\mathbf{A}\mathbf{u}$, so while it is possible to explain a blank image by setting every single element of $\mathbf{u}$ very close to 0 (or, more generally, tuning $\mathbf{u}$ to be in the nullspace of $\mathbf{A}$), a far more parsimonious, and probable, explanation is that z (a single scalar) is close to 0. Importantly, if z is close to 0, then $\mathbf{x}$ does not constrain $\mathbf{u}$. Plausible values for $\mathbf{u}$ therefore cover a broad range (defined by the prior over $\mathbf{u}$), so $\mathbf{u}$ and hence neural activity, can be highly variable. In contrast, if there is a stimulus, $\mathbf{x} \not\approx \mathbf{0}$, we must also have $z \not\approx 0$, in which case $\mathbf{x}$ tightly constrains the range of plausible values of $\mathbf{u}$ (as $\mathbf{x} \approx z\mathbf{A}\mathbf{u}$), leading to lower variability. Moreover, the model naturally implements a form of divisive gain control: a very large $\mathbf{x}$ can be accounted for by making z, rather than $\mathbf{u}$, large (Schwartz et al., 2009). This agreement between the probabilistic model and empirically observed patterns of neural activity is our key motivation for choosing the GSM model as our testbed and asking what plausible neural network dynamics may be appropriate for sampling from its posterior distribution.

Hamiltonian Monte Carlo in an EI network

To ensure efficient sampling from the posterior, we constructed network dynamics based on the core principles of HMC sampling. The efficiency of HMC stems

from its ability to speed up inference by preventing the random walk behaviour plaguing other sampling-based inference schemes. In particular, it introduces auxiliary variables to complement the 'principal' variables whose value needs to be inferred ($\mathbf{u}$ in the case of the GSM). Although this extension of the state space seemingly makes computations more challenging, it allows inference to be substantially more efficient when dynamical interactions between the two groups of variables are set up appropriately.

We noted that the particular interaction between principal and auxiliary variables required by HMC dynamics is naturally implemented by the recurrently connected excitatory and inhibitory populations of cortical circuits. Thus, the dynamics of our two-population neural network that sampled from the GSM posterior were (Fig. 4.3, see Methods for a full derivation):

$$\dot{\mathbf{u}} = \frac{1}{\tau}\left[\mathbf{W}_{\mathrm{uu}}\mathbf{u} - \mathbf{W}_{\mathrm{uv}}\mathbf{v} + \tfrac{1}{2}\tau\rho^2\,\mathbf{I}_{\mathrm{input}}\right] + \rho\boldsymbol{\eta}_{\mathrm{u}} \tag{4.4}$$

$$\dot{\mathbf{v}} = \frac{1}{\tau}\left[\mathbf{W}_{\mathrm{vu}}\mathbf{u} - \mathbf{W}_{\mathrm{vv}}\mathbf{v} - \mathbf{I}_{\mathrm{input}}\right] + \rho\boldsymbol{\eta}_{\mathrm{v}} \tag{4.5}$$

where $\boldsymbol{\eta}_{\mathrm{u}}$ and $\boldsymbol{\eta}_{\mathrm{v}}$ denotes standard normal white noise (or, more precisely, the differential of a Wiener processes), the $\mathbf{W}$ matrices are the recurrent synaptic weight matrices between the two populations of cells (defined in the Methods), such that all their elements are positive, and

$$\mathbf{I}_{\mathrm{input}} = \frac{z}{\sigma_{\mathrm{x}}^2}\,\mathbf{A}^T\left(\mathbf{x} - z\mathbf{A}\mathbf{u}\right) - \mathbf{C}^{-1}\mathbf{u} \tag{4.6}$$

is an input current. Under these dynamics, the principal u_i and auxiliary variables v_i corresponded to the membrane potentials of individual neurons (or the average membrane potential of small populations of cells), and for any input $\mathbf{x}$, the stationary distribution of $\mathbf{u}$ was guaranteed to be identical to the corresponding posterior distribution under the GSM.

Network dynamics consisted of three components. First, recurrent dynamics implementing HMC was specified by the first two terms in Eqs (4.4) and (4.5), $\mathbf{W}_{\mathrm{uu}}\mathbf{u} - \mathbf{W}_{\mathrm{uv}}\mathbf{v}$ and $\mathbf{W}_{\mathrm{vu}}\mathbf{u} - \mathbf{W}_{\mathrm{vv}}\mathbf{v}$. As the elements of the $\mathbf{W}$ matrices were all positive (see above), the recurrent circuit implied by these dynamics had an EI structure, with $\mathbf{u}$ corresponding to excitatory cells and $\mathbf{v}$ to inhibitory cells.

Second, there was an input current $\mathbf{I}_{\mathrm{input}}$, whose strength was scaled by the (inferred) level of contrast, z (Eq. (4.6)). Note again that while this signal might increase with z, it is a prediction error, so it has a highly non-trivial relationship with the resulting response. In fact, it can be shown that the response actually saturates as contrast increases (and results in tuning curves with contrast invariant width) (Orbán et al., 2016). This input current specified the probabilistic model by conveying a prediction error, i.e. the difference between the

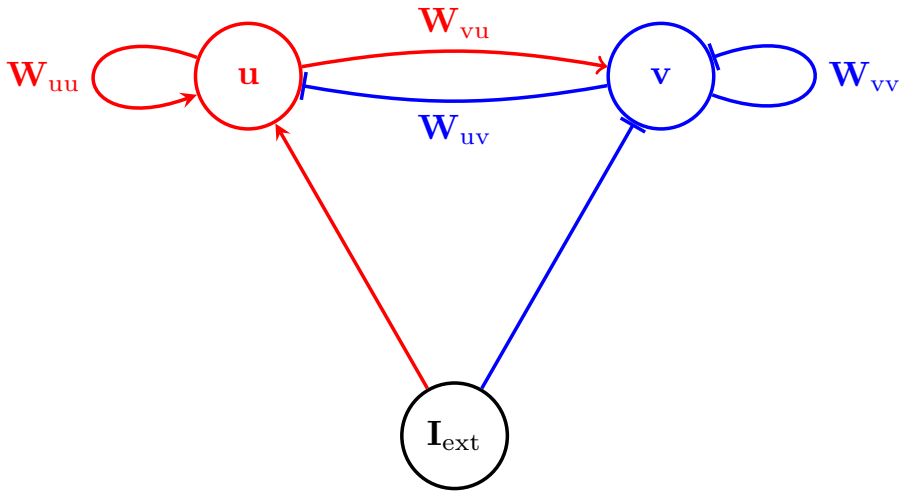

Figure 4.3: The architecture of the Hamiltonian network. The network consists of two populations of neurons, excitatory neurons with membrane potential $\mathbf{u}$, and inhibitory neurons $\mathbf{v}$, driven by external input $\mathbf{I}_{\text{input}}$. Neurons in the network are recurrently coupled by synaptic weights, $\mathbf{W}_{\text{uu}}$, $\mathbf{W}_{\text{uv}}$, $\mathbf{W}_{\text{vu}}$ and $\mathbf{W}_{\text{vv}}$. Red arrows represent excitation; blue bars represent inhibition.

input image, $\mathbf{x}$, and the image predicted by the current activities of the excitatory neurons, $z\mathbf{A}\mathbf{u}$, plus a term penalizing the violation of prior expectations about $\mathbf{u}$. While the key focus of our paper is the EI circuit implementing HMC, rather than the specific form for the input (of which the details depend on the underlying probabilistic model, here the admittedly simplified GSM model), we suggest a potential implementation of $\mathbf{I}_{\text{input}}$ by a separate population of neurons directly representing the prediction error $(\mathbf{x} - z\mathbf{A}\mathbf{u})$ as in theories of predictive coding (Rao and Ballard, 1999). Such cells (perhaps in the lateral geniculate nucleus, LGN) would have an excitatory connection from upstream areas (the retina), representing the data, and an inhibitory disynaptic connection from the excitatory cells, $\mathbf{u}$. The output from these cells needs to excite the excitatory cells and inhibit the inhibitory cells of our circuit, which can again be implemented via disynaptic inhibition. This form of input is particularly well-suited to give strong, long-lasting activation of the EI circuit, as the increase in excitation reinforces the decrease in inhibition.

Finally, the last term in Eqs (4.4) and (4.5) represented noise. Although these dynamics were clearly simplified in that they were fundamentally linear, such dynamical systems have been used to model a wide variety of neural processes (Tsodyks et al., 1997; Murphy and Miller, 2009; Hennequin et al., 2014b). Previous work has also shown that neurons combining firing-rate nonlinearities with short-term synaptic plasticity and dendritic nonlinearities can implement such effectively linear membrane potential dynamics (Pfister et al., 2010; Ujfalussy et al., 2015). Moreover, such models have been found to provide a good match to the dynamics of cortical populations at the level of field potentials (Loebel et al.,

2007), calcium signals (Turaga et al., 2013), and firing rate trajectories (Macke et al., 2011a; Hennequin et al., 2014b). We set the parameters of the network to lie in a biologically realistic regime (Table 4.1, Methods).

Oscillations contribute to efficient sampling

When given an input image, our network exhibited oscillatory dynamics due to its intrinsic excitatory-inhibitory interactions (Fig. 4.4A). Intuitively, these oscillations were useful for inference as they allowed the network to cover a broad range of plausible interpretations of its input within each oscillation cycle. In order to assess more rigorously the computational use of these oscillations, we compared our network to a non-oscillatory counterpart, called Langevin sampling (Roberts and Tweedie, 1996) (Methods). For a fair comparison of the two samplers, we set them up to sample from the same posterior, and we kept the noise level ρ the same in them.

The Langevin sampler was constructed by setting the recurrent weights in our network ($\mathbf{W}$ matrices) to zero. Although, in general, a Langevin sampler can still have recurrent connectivity, at least among the principal cells (by interpreting the dependence of $\mathbf{I}_{\text{input}}$ on $\mathbf{u}$ as recurrent connections (Hennequin et al., 2014a)), these recurrent connections are necessarily symmetric and therefore fundamentally different in nature from the EI interactions that we consider here. As a consequence, Langevin dynamics showed prominent random walk-like behaviour without oscillations (Fig. 4.4B). Comparing the autocorrelation functions for the Hamiltonian and Langevin samplers revealed that while their autocorrelation functions decayed at similar rates (controlled by the timescale of the stochastic, Langevin component), HMC had an additional, oscillating component, (Fig. 4.4C).

The oscillatory behaviour of our HMC sampler allowed it to explore a larger volume of state space in a fixed time interval than Langevin sampling (Fig. 4.4D-E). To compare the sampling performance of HMC and Langevin dynamics rigorously, we measured for both of them the error between a sample-based estimate of the posterior mean and the true mean of the posterior. The samples from the Hamiltonian sampler took very little time to give a good estimate of the mean (73 ms to get the mean square error to the level obtainable by a single statistically fair sample), whereas samples from the Langevin model took $\sim$4 times longer (273 ms, Fig. 4.4F). This difference indicated that our HMC-inspired sampler used limited noise far more efficiently than Langevin dynamics.

The efficiency of HMC is typically attributed to the suppression of the random walk behaviour of Langevin dynamics (Neal, 2011). In our network, we were able to relate this effect more specifically to the appearance of oscillations. HMC

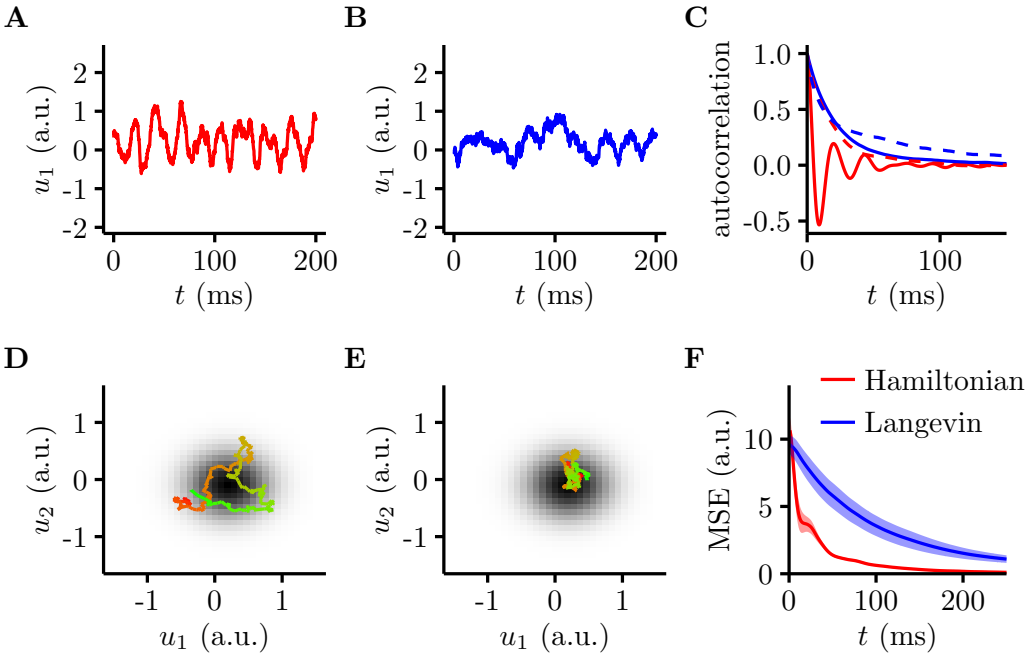

Figure 4.4: The Hamiltonian sampler is more efficient than a Langevin sampler. **A, B.** Example membrane potential traces for a randomly selected neuron in the Hamiltonian network (**A**) and the Langevin network (**B**). **C.** Solid lines: the autocorrelation of membrane potential traces in **A** and **B**, for Hamiltonian (red) and Langevin samplers (blue). Dashed lines: the autocorrelation of the joint (log) probability for Hamiltonian (red) and Langevin samplers (blue). Note that for the Hamiltonian sampler, the joint probability is over both $\mathbf{u}$ and $\mathbf{v}$. **D, E.** Joint membrane potential traces from two randomly selected neurons in the Hamiltonian network (**D**) and the Langevin network (**E**), colour indicates time (from red to green, spanning 25 ms), grey scale map shows the (logarithm of the) underlying posterior (its marginal over the two dimensions shown). **F.** Normalised mean square error (MSE) between the true mean and the mean estimate from samples taken over a time t for the Langevin (blue) and Hamiltonian dynamics (red), with 100 repetitions (mean $\pm$ 2 s.e.m.).

dynamics had both an oscillatory and a stochastic component (Fig. 4.4A, C red), whereas Langevin dynamics had only the stochastic component, so that it performed simple noisy gradient ascent, without apparent oscillations (Fig. 4.4B, C blue). In particular, oscillations in the HMC sampler had a time scale that was a factor of 15 faster than that of the stochastic component shared with Langevin dynamics. This fast time constant of the HMC sampler, τ, governed the effects of recurrent EI interactions, which were mediated by the $\mathbf{W}$ matrices that the Langevin sampler lacked (Eq. (4.32)). These architectural and dynamical differences implied a fundamentally different strategy for exploring the state space of these networks. The fast oscillations in the HMC sampler deterministically explored states in $(\mathbf{u}, \mathbf{v})$-space that lay on an equiprobability manifold, while the slow time scale implied by the input noise served to change this manifold stochastically (Fig. 4.4D). Indeed, the autocorrelogram of the energy (log posterior probability) in the HMC sampler (Fig. 4.4C, red dashed curve) was identical to the Langevin envelope of the autocorrelogram of states (Fig. 4.4C, red solid curve), indicating that energy only changed on the slow time scale governed by this stochastic component and not on the fast time scale of oscillations. (Note that while moving along equiprobability contours in the full joint $(\mathbf{u}, \mathbf{v})$ space, HMC dynamics may still cross probability contours when projected to a low dimensional marginal, as shown in Fig. 4.4D.) In contrast, Langevin dynamics could only rely on this slow stochastic component resulting in slow movement across energy levels (Fig. 4.4C, blue dashed curve) and the state space (Fig. 4.4C, blue solid curve).

Balance between excitation and inhibition

As we saw above, the advantage of HMC over Langevin dynamics could be attributed to the contribution of the recurrent connections, i.e. the $\mathbf{W}_{\mathrm{uu}}\mathbf{u} - \mathbf{W}_{\mathrm{uv}}\mathbf{v}$ and $\mathbf{W}_{\mathrm{vu}}\mathbf{u} - \mathbf{W}_{\mathrm{vv}}\mathbf{v}$ terms in the dynamics (Eq. (4.4) and (4.5)), which respectively expressed the difference between net excitation and inhibition received by excitatory and inhibitory neurons. (Note that this difference was not affected by $\mathbf{I}_{\mathrm{input}}$ as the prediction error conveyed by the input is zero on average for any input, by definition.) Importantly, for HMC to sample from the correct posterior, the dynamics of excitatory cells needed to track the prediction error conveyed by $\mathbf{I}_{\mathrm{input}}$, for which the recurrent term needed to be zero on average, which in turn suggests that excitation and inhibition needed to track each other across different stimuli (Fig. 4.5A). Indeed, the only way we could obtain Hamiltonian dynamics that complied with Dale's law was if the activity of inhibitory cells tracked that of excitatory cells, i.e. if the network was balanced. As Langevin is equivalent to having these terms set to zero, for HMC to realize its advantage over Langevin, the variance of the recurrent term needed to be sufficiently large, which implied

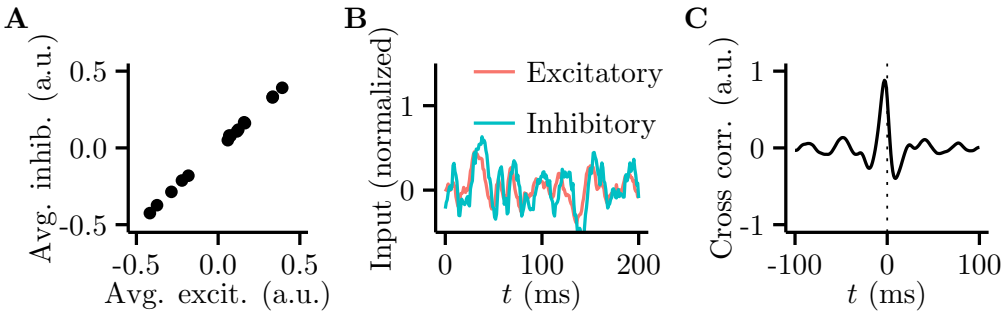

Figure 4.5: Excitation and inhibition are balanced in the Hamiltonian network. **A.** Trial-average excitatory input vs. trial-average inhibitory input across trials (dots) for a randomly selected individual cell in the network. **B.** Total inhibitory input to a single cell (blue) closely tracks but slightly lags total excitatory input (red) over the course of a trial. **C.** The cross-correlation between the average excitatory and average inhibitory membrane potentials shows a peak that is offset from 0 time.

that the magnitudes of net excitation and net inhibition each needed to be large and momentarily imbalanced (Fig. 4.5B). These features, large excitatory and inhibitory currents that are tracking each other with momentary perturbations, are thought to be fundamental properties of the dynamical regime in which the cortex operates (Okun and Lampl, 2008), and thus arise naturally from HMC dynamics in our EI network. Furthermore, as expected in a network with an EI architecture, excitation led inhibition in our network (Fig. 4.5C).

Stimulus-dependent oscillations

Oscillations are a ubiquitous property of cortical dynamics (Buzsaki, 2006), and we have shown above that efficient sampling in HMC necessarily leads to oscillatory dynamics in general (Figs. 4.4-4.5). However, when applied specifically to perform inference based on visual images (Fig. 4.2), our model also reproduced some more specific and robust properties of gamma-band oscillations in V1, namely that the precise frequency of these oscillations increases with stimulus contrast (Ray and Maunsell, 2010; Roberts et al., 2013) (Fig. 4.6).

In order to extract an LFP from our model, in line with previous approaches (e.g. (Wilson and Cowan, 1972)), we computed the sum of membrane potentials of all cells. (Using the sum of input currents instead would have yielded qualitatively similar results.) The fact that LFP oscillations in our model were in the gamma band, i.e. around 40 Hz, was simply due to our choice of a realistic single neuron time constant, $\tau = 10$ ms. However, within this band, the modulation of the oscillation frequency by the contrast of the input image was a more specific characteristic of the dynamics of our network. As contrast increased, the amount of evidence to pin down $\mathbf{u}$ increased, and so the GSM posterior from which the

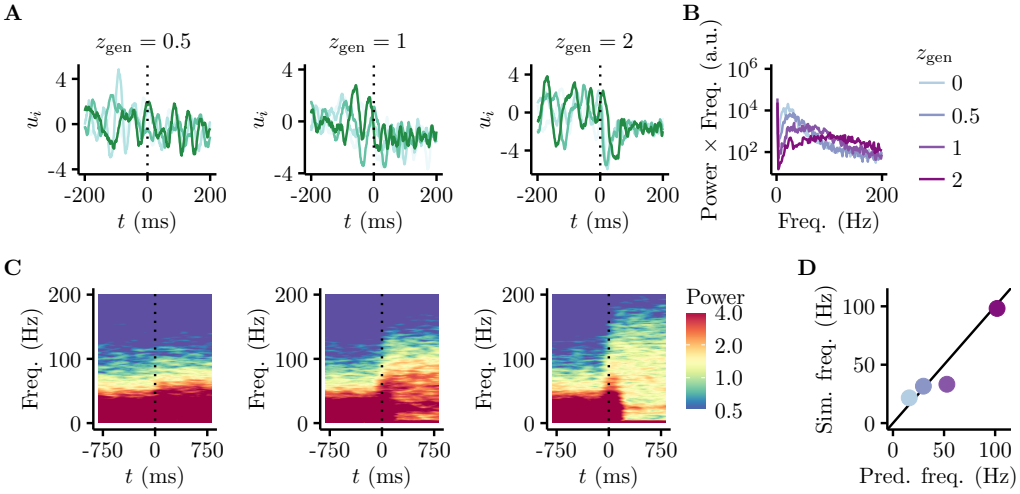

Figure 4.6: Oscillation frequency depends on stimulus contrast. **A.** The membrane potential response of one neuron to stimulus onset across 4 trials (coloured curves) shows that the variability decreases and the frequency increases as stimulus contrast increases. The true contrast of the underlying image increases left to right ($z_{\text{gen}} = 0.5$, 1, and 2). **B.** Power spectrum of the LFP (average membrane potentials) at different contrasts (coloured lines), showing that dominant oscillation frequency increases with contrast. Note that we plot power × frequency on the y-axis, in order to account for the fact that noise from a "scale-free" process has 1/f frequency dependence [59]. **C.** Time-dependent spectrum (Gaussian window, width 100 ms) of the LFP (contrast levels as in **A**). **D.** The simplified dynamics (x-axis, Eq. (4.8)) accurately predicted the dependence of oscillation frequencies on contrast (colour code as in B) in the full network (y-axis).

dynamics needed to sample became tighter (Orbán et al., 2016). At the same time, the recurrent EI interactions of the HMC dynamics which gave rise to oscillations had a fixed time scale independent of the input (Eqs. (4.4) and (4.5)). Using the same speed to traverse an equiprobability manifold of an increasingly tight posterior thus naturally led to increasing oscillation frequencies.

To further quantify this intuition, we simplified the dynamics of our network by incorporating the effects of inhibition directly into the equations describing the dynamics of the excitatory cells (see Methods):

$$\ddot{\mathbf{u}} = -\frac{1}{\tau^2}\left(\frac{z^2}{\sigma_{\mathrm{x}}^2} - \frac{1}{1-\sigma_{\mathrm{x}}^2}\right)(\mathbf{u} - \bar{\mathbf{u}}) \tag{4.7}$$

where $\bar{\mathbf{u}} = \mathrm{E}\left[\mathbf{u}|\mathbf{x}, z\right]$ is the (stimulus-dependent) mean of the posterior over $\mathbf{u}$. This form explicitly exposes that our sampler (in the limit studied here) underwent regular harmonic oscillations, whose frequency increased with stimulus contrast, z_{gen} (assuming that the inferred value of z was sufficiently close to the actual stimulus contrast, i.e. $z \simeq z_{\mathrm{gen}}$), as

$$f(z) = \frac{1}{2\pi\tau}\sqrt{\frac{z_{\mathrm{gen}}^2}{\sigma_{\mathrm{x}}^2} - \frac{1}{1-\sigma_{\mathrm{x}}^2}} \tag{4.8}$$

Indeed, as predicted by these arguments, the network exhibited contrast-dependent oscillation frequencies both in its membrane potentials (Fig. 4.6A) and LFPs (Fig. 4.6B-C; note that in B, we account for the fact that a "scale-free" noise process has $1/f$ frequency dependence (Milotti, 2002) by plotting power $\times$ frequency on the y-axis). Furthermore, the quantitative predictions made by Eq. (4.8) were in close agreement with the results of numerical simulations in the the full model, where z is not fixed, but is inferred simultaneously with $\mathbf{u}$ (Fig. 4.6D).

Stimulus-dependent transients

When we computed firing rates in the model by applying a threshold to membrane potentials (Eq. (4.60)), our simulations showed large, contrast-dependent transient increases in population firing rate at stimulus onset (Fig. 4.7A). (Were we to consider the average membrane potential, this would not display such a large transient, because some neurons undergo positive transients, and others undergo negative transients, which cancel overall.) Such transients are also a widely observed characteristic of responses in V1 (Müller et al., 2001; Ray and Maunsell, 2010) (as well as other sensory cortices (Bermudez Contreras et al., 2013; Luczak et al., 2013)). These transients were also inherent to the dynamics of our network and were not trivially predicted by simpler variants. For example, Langevin sam-

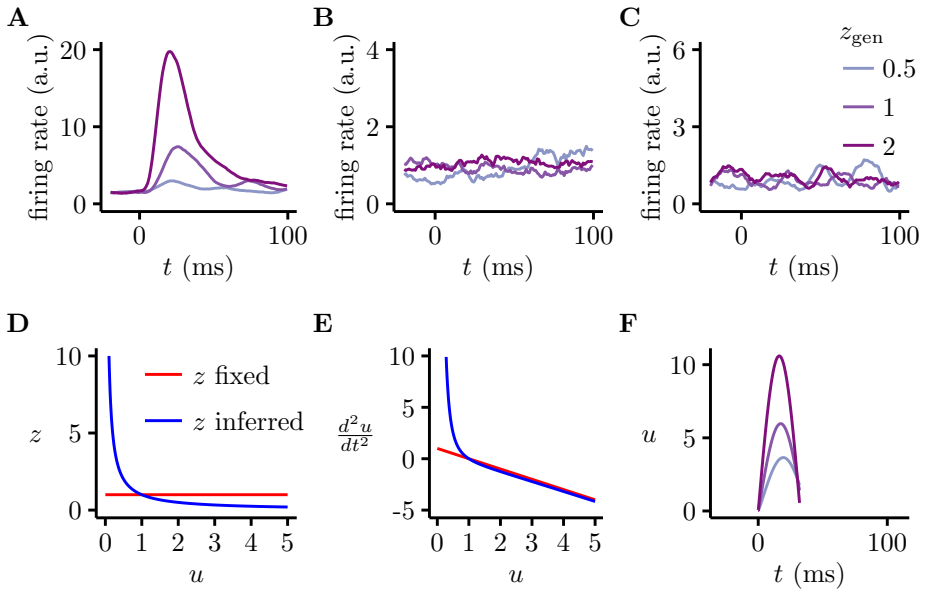

Figure 4.7: Large, contrast-dependent firing rate transients in the model. **A-C.** Transients (or lack thereof) at different contrast levels (colour) under the full dynamics (**A**), using Langevin dynamics (**B**), and under the full dynamics when the value of z is fixed, $z = z_{\text{gen}}$ (**C**). Note different scales for firing rates in the three panels to better show the full range of firing rate fluctuations in each case. **D.** Dependence of the inferred value of contrast, z, on the currently inferred magnitude of basis function intensities, u, under the simplified dynamics (blue). For reference, red shows the value of z when set to be fixed at $z = z_{\text{gen}}$. **E.** There is asymmetry in $\ddot{u}$ as a function of u, around the value of $u = \bar{u} = 1$, in the simplified model when z is inferred (blue) but not when it is fixed (red). **F.** Transients predicted by the simplified dynamics (Eq. (4.9), with parameters as in Fig. 4.6D, and initial conditions $u(0) = 0.1$ and $\dot{u}(0) = 0$) are similar to transients under the full dynamics.

pling did not give rise to any transient increase in firing rates — rates simply rose or fell towards their new steady state (Fig. 4.7B, most obvious for $z_{\text{gen}} = 0.5$). Even Hamiltonian dynamics did not necessarily yield transients. In particular, the full dynamics of our network inferred contrast, z, online together with the basis function intensities $\mathbf{u}$. Assuming instead that the brain knows $z = z_{\text{gen}}$, or uses a fixed value of z sampled from $P(z|\mathbf{x})$, the dynamics became simple noisy harmonic motion. Although harmonic motion can lead to transients when initialised properly, the transients yielded by these dynamics were much smaller in magnitude which were near-impossible to detect in simulated population firing rates (Fig. 4.7C).

In order to understand how transients emerged in the full Hamiltonian dynamics of our network, sampling $\mathbf{u}$ and z jointly, we focussed on the interaction between the dynamics of $\mathbf{u}$ and the inferred value of z. For analyzing the asymptotic behaviour in the previous section, we assumed that z was constant (and equal to z_{gen}). However, in general, z depended on the network's currently inferred value

of $\mathbf{u}$. In particular, z and $\mathbf{u}$ jointly accounted for the total contrast content of the input image $\mathbf{x}$ (Eq. (4.3)), and thus there was an inverse scaling between their magnitudes. Using the 1D variant of Eq. (4.7), $x \approx zAu$, so $z \approx x/Au$ (Fig. 4.7D). Here, we make use of a separation of time scales between the dynamics of z and $\mathbf{u}$, specifically that z will attain its stationary value (distribution) much faster than $\mathbf{u}$. This is because while the basis functions of u_i's are localised Gabor filters, z depends on the whole image patch (or, conversely, on all the u_i's), which means that the sensory evidence for z is much stronger than for $\mathbf{u}$, and consequently its distribution is much narrower, giving strong prediction error signals which rapidly drive it to equilibrium. As z effectively set the stiffness of the 'spring' underlying harmonic motions in our dynamics (Eq. (4.7)), the system had high (restoring) acceleration for low values of $|u|$ and low accelerations for high values of $|u|$, resulting in high magnitude excursions in u (Fig. 4.7E). Therefore, just after stimulus onset, u was small, so there was a large force in the positive direction (due to the large stiffness), causing a large acceleration. Eventually, u exceeded $\bar{u}$, but by that point the stiffness, and hence the restoring force had fallen, so the system's momentum allowed it to move a long distance, certainly further than if the spring constant had been fixed. This asymmetry in preferring upward to downward changes in $|u|$ was only relevant during initial transients as asymptotically the evidence in the image was sufficient to determine z with high precision and so the dynamics of u became approximately linear (as in Eq. (4.7)). Thus, the timescale of the transient was determined by the timescale at which inferences about z attained their stationary distribution, which in turn scaled with ρ (S1 Figure).

More formally, taking the 1D version of the simplified dynamics (Eq. (4.7)), and substituting $z \approx x/Au$ gives

$$\ddot{u} = -\frac{1}{\tau^2}\left(\frac{x^2}{\sigma_{\mathrm{x}}^2 A^2 u^2} + \frac{1}{1-\sigma_{\mathrm{x}}^2}\right)(u - \bar{u}) \tag{4.9}$$

Simulating this simplified dynamical system did indeed yield large transients (Fig. 4.7F) which matched full simulations (Fig. 4.7A) and recordings in macaque V1 (Ray and Maunsell, 2010) both in terms of the transient timescale (∼30 ms) and the dependence of transient magnitude on contrast level (values of z_{gen}). The fact that these large transients were retained in the model after such severe approximations indicated that they were robust to the exact method used for determining z, as long as it ensured that z was consistent with both $\mathbf{x}$ and $\mathbf{u}$.

4.4 Discussion

Previously proposed mechanisms by which the cortex could either represent and manipulate uncertainty or just find the most probable explanation for sensory data failed to explain the richness of cortical dynamics. In particular, these models either had no dynamics or only gradient ascent-like dynamics, in contrast to neural activity in the cortex that displays oscillations in response to a fixed stimulus, and large transients in response to stimulus onset. Moreover, these models typically violated Dale's law, by having neurons whose outputs were both excitatory and inhibitory. We demonstrated that it was, in fact, possible to perform probabilistic inference in an EI network that displayed oscillations and transients. Moreover, having oscillations actually improved the network, in that it was able to perform inference faster than networks that did not have oscillations. Our model displayed four further dynamical properties that did not appear, at first, to be compatible with probabilistic inference: excitation and inhibition were balanced at the level of individual cells (Okun and Lampl, 2008), inhibition lagged excitation (Okun and Lampl, 2008), oscillation frequency increased with stimulus contrast (Ray and Maunsell, 2010), and there were large transients upon stimulus onset which also scaled with contrast (Müller et al., 1999, 2001; Ray and Maunsell, 2010). In sum, we have given an approach by which successful, inference-based models of stationary activity distributions in V1 (e.g. (Orbán et al., 2016)) can be extended to match the dynamics of neural activity.

Our work suggests a new functional role for cortical oscillations, and for inhibitory neurons that are involved in their generation: speeding up inference. We have demonstrated this role in the specific context of V1, but our formalism is readily applicable to other cortical areas in which probabilistic inference is supposed to take place, and similar stimulus-controlled transients and oscillations can be observed (Wang et al., 2005; Buzsáki and Watson, 2012). Neural oscillations and probabilistic inference have been linked previously, albeit in the hippocampus rather than sensory cortices (Savin et al., 2014). The main differences between the two approaches are that in previous work, oscillations were controlled entirely externally, and implemented (approximately) an augmented sampling scheme known as tempered transitions (Neal, 1996), whereas our work builds on the theory of Hamiltonian Monte Carlo (Neal, 2011) to construct network dynamics that are intrinsically oscillating. This allowed us to study the effects of the stimulus on these oscillations that previous approaches could not address. Computationally, Hamiltonian Monte Carlo and annealing-based techniques, such as tempered transitions, have complementary advantages in allowing network dynamics to respectively explore a given posterior mode or traverse different modes efficiently. Thus, a combination of these different approaches may account for concurrent cortical oscillations at different frequencies.

While the statistical model of images underlying our network was able to capture some interesting properties of the statistics of natural images, it was nevertheless clearly simplified, in that e.g. it did not capture any notion of objects, or occlusion. Once such higher-order features are incorporated into the model, we expect a variety of interesting new dynamical properties to emerge. For example, there should be strong statistical relationships between low-level variables describing a single object, and hence strong dynamical relationships, including synchronisation, between neurons representing different parts of the same object (Womelsdorf et al., 2007; Fries, 2009). In the extreme, we might expect to see coherent oscillations between neurons representing the same object, providing a principled unifying perspective of bottom-up (e.g. contrast) and top-down influences (e.g. "binding by synchrony") on cortical oscillations (Singer, 1999).

It will also be important to understand how local learning rules, modelling synaptic plasticity, may be able to set up the weight matrices that we found were necessary for implementing efficient Hamiltonian dynamics. For example, there might be two sets of learning rules operating in parallel, one set of rules which learns that statistical structure of the input, perhaps mainly through the plasticity of excitatory-to-excitatory connections (Markram et al., 2012), and another which tunes network dynamics, perhaps primarily by inhibitory plasticity mechanisms, to speed up the inference process, without altering the sampled distribution (Kullmann et al., 2012).

Finally, while the type of linear membrane potential dynamics we used in our network could be implemented using firing rate non-linearities in combination with synaptic and dendritic nonlinearities (Pfister et al., 2010; Ujfalussy et al., 2015), it will nevertheless be important to understand whether it is possible to perform inference in networks with more realistic non-linearities.

4.5 Methods

Sampler derivation

The sampler was derived by combining an HMC step, and a Langevin step to add noise and ensure ergodicity. The most general equations describing HMC are given by

$$\dot{\mathbf{u}} = \frac{1}{\tau}\frac{\partial \log P\left(\mathbf{u}, \mathbf{v}|\mathbf{x}, z\right)}{\partial \mathbf{v}} \tag{4.10}$$

$$\dot{\mathbf{v}} = -\frac{1}{\tau}\frac{\partial \log P\left(\mathbf{u}, \mathbf{v}|\mathbf{x}, z\right)}{\partial \mathbf{u}} \tag{4.11}$$

For the HMC step, there is freedom to specify the distribution of the auxiliary variable, $P\left(\mathbf{v}|\mathbf{u}, \mathbf{x}\right)$, and freedom to set the noise distribution. Typically, the

distribution of the auxilliary variable is set to have $\mathbf{0}$ mean and be totally independent of $\mathbf{u}$, so that $P(\mathbf{v}|\mathbf{u},\mathbf{x},z) = P(\mathbf{v}) = \mathcal{N}\left(\mathbf{v};\mathbf{0},\mathbf{M}^{-1}\right)$. However, we know that inhibitory cells do, in fact, respond to input. We therefore chose to use

$$P(\mathbf{v}|\mathbf{u},\mathbf{x},z) = P(\mathbf{v}|\mathbf{u}) = \mathcal{N}\left(\mathbf{v};\mathbf{B}\mathbf{u},\mathbf{M}^{-1}\right) \tag{4.12}$$

with a free choice for $\mathbf{B}$ and $\mathbf{M}$, which we will discuss below (Setting the parameters). This allowed us to split up these probability distributions into terms that are dependent, and independent, of the data, $\mathbf{x}$:

$$\dot{\mathbf{u}} = \frac{1}{\tau}\frac{\partial \log P(\mathbf{v}|\mathbf{u})}{\partial \mathbf{v}} \tag{4.13}$$

$$\dot{\mathbf{v}} = -\frac{1}{\tau}\frac{\partial \log P(\mathbf{v}|\mathbf{u})}{\partial \mathbf{u}} - \frac{1}{\tau}\frac{\partial \log P(\mathbf{u}|\mathbf{x},z)}{\partial \mathbf{u}} \tag{4.14}$$

In order to add noise without perturbing the stationary distribution, we perform a Langevin step, that is, we simultaneously add noise and take a step along the gradient of the log-probability. Notably, this introduces a new time constant τ_L, that simply controls the rate at which noise is injected into the system. As such, τ_L is directly related to ρ,

$$\rho = \sqrt{\frac{2}{\tau_L}} \tag{4.15}$$

The dynamics therefore become

$$\dot{\mathbf{u}} = \frac{1}{\tau}\frac{\partial \log P(\mathbf{v}|\mathbf{u})}{\partial \mathbf{v}} + \frac{1}{\tau_L}\frac{\partial \log P(\mathbf{u},\mathbf{v}|\mathbf{x},z)}{\partial \mathbf{u}} + \sqrt{\frac{2}{\tau_L}}\boldsymbol{\eta}_{\mathrm{u}} \tag{4.16}$$

$$\dot{\mathbf{v}} = -\frac{1}{\tau}\frac{\partial \log P(\mathbf{v}|\mathbf{u})}{\partial \mathbf{u}} - \frac{1}{\tau}\frac{\partial \log P(\mathbf{u}|\mathbf{x},z)}{\partial \mathbf{u}} + \frac{1}{\tau_L}\frac{\partial \log P(\mathbf{u},\mathbf{v}|\mathbf{x},z)}{\partial \mathbf{v}} + \sqrt{\frac{2}{\tau_L}}\boldsymbol{\eta}_{\mathrm{v}} \tag{4.17}$$

Again, we can break up the $P(\mathbf{u},\mathbf{v}|\mathbf{x},z)$ terms into terms that are dependent, and independent, of $\mathbf{v}$:

$$\dot{\mathbf{u}} = \frac{1}{\tau}\frac{\partial \log P(\mathbf{v}|\mathbf{u})}{\partial \mathbf{v}} + \frac{1}{\tau_L}\frac{\partial \log P(\mathbf{v}|\mathbf{u})}{\partial \mathbf{u}} + \frac{1}{\tau_L}\frac{\partial \log P(\mathbf{u}|\mathbf{x},z)}{\partial \mathbf{u}} + \sqrt{\frac{2}{\tau_L}}\boldsymbol{\eta}_{\mathrm{u}} \tag{4.18}$$

$$\dot{\mathbf{v}} = -\frac{1}{\tau}\frac{\partial \log P(\mathbf{v}|\mathbf{u})}{\partial \mathbf{u}} + \frac{1}{\tau_L}\frac{\partial \log P(\mathbf{v}|\mathbf{u})}{\partial \mathbf{v}} - \frac{1}{\tau}\frac{\partial \log P(\mathbf{u}|\mathbf{x},z)}{\partial \mathbf{u}} + \sqrt{\frac{2}{\tau_L}}\boldsymbol{\eta}_{\mathrm{v}} \tag{4.19}$$

Now, we compute these gradients, and convert them into a neural-network (see S1 Code)

$$\frac{\partial \log P(\mathbf{v}|\mathbf{u})}{\partial \mathbf{u}} = -\mathbf{M}(\mathbf{B}\mathbf{u} - \mathbf{v}) \tag{4.20}$$

$$\frac{\partial \log P(\mathbf{v}|\mathbf{u})}{\partial \mathbf{v}} = \mathbf{B}^T\mathbf{M}(\mathbf{B}\mathbf{u} - \mathbf{v}) \tag{4.21}$$

where the gradient of the posterior is the external input

$$\mathbf{I}_{\text{input}} = \frac{\partial \log P\left(\mathbf{u}|\mathbf{x}, z\right)}{\partial \mathbf{u}} = \frac{1}{\sigma_{\text{x}}^2} z\mathbf{A}^T\left(\mathbf{x} - z\mathbf{A}\mathbf{u}\right) - \mathbf{C}^{-1}\mathbf{u} \tag{4.22}$$

We can thus write the dynamics of our neural network as

$$\dot{\mathbf{u}} = \frac{1}{\tau}\left(\mathbf{W}_{\text{uu}}\mathbf{u} - \mathbf{W}_{\text{uv}}\mathbf{v} + \frac{\tau}{\tau_L}\mathbf{I}_{\text{input}}\right) + \sqrt{\frac{2}{\tau_L}}\boldsymbol{\eta}_{\text{u}} \tag{4.23}$$

$$\dot{\mathbf{v}} = \frac{1}{\tau}\left(\mathbf{W}_{\text{vu}}\mathbf{u} - \mathbf{W}_{\text{vv}}\mathbf{v} - \mathbf{I}_{\text{input}}\right) + \sqrt{\frac{2}{\tau_L}}\boldsymbol{\eta}_{\text{v}} \tag{4.24}$$

where

$$\mathbf{W}_{\text{uu}} = \mathbf{B}^T\mathbf{M}\mathbf{B} - \frac{\tau}{\tau_L}\mathbf{M}\mathbf{B} \tag{4.25}$$

$$\mathbf{W}_{\text{uv}} = \mathbf{B}^T\mathbf{M} - \frac{\tau}{\tau_L}\mathbf{M} \tag{4.26}$$

$$\mathbf{W}_{\text{vu}} = \mathbf{M}\mathbf{B} + \frac{\tau}{\tau_L}\mathbf{B}^T\mathbf{M}\mathbf{B} \tag{4.27}$$

$$\mathbf{W}_{\text{vv}} = \mathbf{M} + \frac{\tau}{\tau_L}\mathbf{B}^T\mathbf{M} \tag{4.28}$$

Finally, we substitute $\tau_L = 2/\rho^2$.

Sampling z

The brain does not know z_{gen}, so it must infer z together with $\mathbf{u}$. We therefore inferred z and $\mathbf{u}$ in parallel, using an additional HMC sampler for z.

In particular, we simply extended the dynamics with an additional element for z:

$$\dot{z} = \frac{1}{\tau}\left(W_{zz}z - W_{zv}v + \frac{\tau}{\tau_L}I_{\text{input}}\right) + \sqrt{\frac{2}{\tau_L}}\eta_{\text{z}} \tag{4.29}$$

$$\dot{v} = \frac{1}{\tau}\left(W_{vz}z - W_{vv}v - I_{\text{input}}\right) + \sqrt{\frac{2}{\tau_L}}\eta_{\text{v}} \tag{4.30}$$

where W is defined as above, with $B = M = 1$, and

$$I_{\text{intput}} = \frac{\partial \log P\left(\mathbf{u}, z, \mathbf{x}\right)}{\partial z} = \frac{1}{\sigma_{\text{x}}^2}\left(\mathbf{A}\mathbf{u}\right)^T\left(\mathbf{x} - z\mathbf{A}\mathbf{u}\right) - z \tag{4.31}$$

Langevin sampler

By setting the weight matrices implementing HMC, $\mathbf{W}$, to $\mathbf{0}$, we obtain the Langevin step:

$$\dot{\mathbf{u}} = \frac{1}{\tau_L}\mathbf{I}_{\text{input}} + \sqrt{\frac{2}{\tau_L}}\boldsymbol{\eta}_{\text{u}} \tag{4.32}$$

Setting the parameters

The GSM model has three parameters, the Gabor features, $\mathbf{A}$, the covariance matrix, $\mathbf{C}$, and the observation noise, σ_{x}^2. We set $\mathbf{A}$ using known properties of the visual system: the Gabor filters-like receptive fields of V1 simple cells. In particular, we define $\mathbf{A}$ as a bank of Gabor filters at three orientations (0, $\pi/3$ and $2\pi/3$), five locations (the centre, and corners, 1/6 image-widths from the edge, where all measurements are in units of image height = image width). The Gaussian envelope of the Gabors had minor axis 0.1, and major axis uniformly distributed from 0.1 to 0.5 (where these measurements are in units of image width, and give the standard deviation along the relevant axis), and the sinusoid had wavelength 0.13 image-widths.

We can set $\mathbf{C}$ using the value for $\mathbf{A}$, and the fact that retina and LGN are known to whiten visual input (Dayan and Abbott, 2001). For a particular image, $\mathbf{x}$, and inferred contrast level, z, the posterior is

$$P\left(\mathbf{u}|\mathbf{x}, z\right) = \mathcal{N}\left(\mathbf{u}; \tfrac{z}{\sigma_{\text{x}}^2}\boldsymbol{\Sigma}(z)\,\mathbf{A}^T\mathbf{x}, \boldsymbol{\Sigma}(z)\right) \tag{4.33}$$

where

$$\boldsymbol{\Sigma}(z) = \left(\mathbf{C}^{-1} + \tfrac{z^2}{\sigma_{\text{x}}^2}\mathbf{A}^T\mathbf{A}\right)^{-1} \tag{4.34}$$

We know that the average posterior equals the prior (Dempster et al., 1977; Berkes et al., 2011b), and so the prior covariance $\mathbf{C}$ should match the average posterior covariance (averaging over data, $\mathbf{x}$, and other latent variables, z), i.e.

$$\mathbf{C} = \mathrm{E}\left[\mathbf{u}\mathbf{u}^T\right] = \mathrm{E}\left[\tfrac{z^2}{\sigma_{\text{x}}^4}\boldsymbol{\Sigma}(z)\mathbf{A}^T\mathbf{x}\mathbf{x}^T\mathbf{A}\boldsymbol{\Sigma}(z) + \boldsymbol{\Sigma}(z)\right] \tag{4.35}$$

We make the ansatz that

$$\mathbf{C} = K\left(\mathbf{A}^T\mathbf{A}\right)^{-1} \tag{4.36}$$

where K is an unknown constant. Substituting this guess into Eq. (4.34), we see that $\mathbf{\Sigma}(z)$ simplifies considerably:

$$\mathbf{\Sigma}(z) = \left(K^{-1} + \frac{z^2}{\sigma_{\mathrm{x}}^2}\right)^{-1} \left(\mathbf{A}^T\mathbf{A}\right)^{-1} \tag{4.37}$$

and as the data are whitened (assuming this is true at any contrast level, i.e. $\mathrm{E}_{\mathbf{x}|z}\left[\mathbf{x}\mathbf{x}^T\right] = c(z)\,\mathbf{I}$, with some $c(z)$), we indeed have

$$\mathrm{E}_{\mathbf{u}}\left[\mathbf{u}\mathbf{u}^T\right] \propto \left(\mathbf{A}^T\mathbf{A}\right)^{-1} \tag{4.38}$$

confirming our ansatz.

In principle, we could find K by solving Eq. (4.35) (by substituting Eq. (4.36) to its l.h.s., and Eq. (4.37) to its r.h.s.), however, in practice, we cannot because we do not know $c(z)$ in $\mathrm{E}_{\mathbf{x}|z}\left[\mathbf{x}\mathbf{x}^T\right] = c(z)\,\mathbf{I}$. Instead, we set K to ensure that the inputs, $\mathbf{A}^T\mathbf{x}$, have the right covariance (note that it is only possible to match the covariance of $\mathbf{A}^T\mathbf{x}$, and not of $\mathbf{x}$ directly, because we are using an undercomplete basis). As the data is whitened, we expect

$$\mathrm{E}\left[\mathbf{A}^T\mathbf{x}\mathbf{x}^T\mathbf{A}\right] = \mathbf{A}^T\mathbf{A} \tag{4.39}$$

while the predictive distribution of the GSM results in

$$\mathrm{E}\left[\mathbf{A}^T\mathbf{x}\mathbf{x}^T\mathbf{A}\right] = \mathbf{A}^T\left(\mathrm{E}\left[z^2\right]\mathbf{A}\mathbf{C}\mathbf{A}^T + \sigma_{\mathrm{x}}^2\mathbf{I}\right)\mathbf{A} \tag{4.40}$$

Setting these expressions equal, substituting for $\mathbf{C}$ using our ansatz (Eq. (4.36)), and using $\mathrm{E}\left[z^2\right] = 1$ gives

$$\mathbf{A}^T\mathbf{A} = \left(K + \sigma_{\mathrm{x}}^2\right)\mathbf{A}^T\mathbf{A} \tag{4.41}$$

yielding the solution

$$K = 1 - \sigma_{\mathrm{x}}^2 \tag{4.42}$$

(Note that while this derivation is valid for the complete and undercomplete case, a more complex analysis would be necessary for the overcomplete case.)

With these choices, the dynamics only depend on the probabilistic model through the product $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}$. This product controls the frequency spectrum: if $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}$ has a very broad eigenspectrum (e.g. multiple orders of magnitude), then the system will sample at different rates along different directions. This is not desirable: we want sampling to take place as fast as possible in every direction, not to be fast in some directions, and slow in others. If we were able to set $\mathbf{M}$ to $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}$, then we would indeed sample at the same rate in every direction (Neal, 2011), no matter how broad the spectrum of $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}$ (see "Deriving the

1D approximate model", below). However, to ensure that Dale's law is obeyed, we need the elements of $\mathbf{M}$ to be non-negative, so we set

$$\mathbf{B} = \mathbf{I} \tag{4.43}$$

and

$$M_{ij} = \max\left(0, \left(\mathbf{A}^T\mathbf{A}\right)^{-1}_{ij}\right) \tag{4.44}$$

For the dynamics to be correct, we need this matrix to be positive definite. While this is not guaranteed, we found that in practice the matrix turns out to satisfy this constraint. As $\mathbf{M}$ is close to, but not exactly, $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}$, the eigenspectrum of $\mathbf{A}^T\mathbf{A}$ will have some effect on our sampler. In practice, our eigenvalues range over a factor of 5 without weakening our results. Again, this is valid for the undercomplete and complete cases, and a more complex analysis would be necessary for the overcomplete case.

Next, we consider the observation noise level, σ_{x}, which describes the noise-to-signal ratio for neurons in the visual cortex. In particular, we take the input to be $\mathbf{A}^T\mathbf{x}$. This input is made up of two components, signal from the mean of $P\left(\mathbf{A}^T\mathbf{x}|\mathbf{u}, z\right)$, and noise from its covariance, (given by transforming Eq. (4.3)). The covariance of this input (Eq. (4.40)) also breaks up into signal, $\left(1-\sigma_{\mathrm{x}}^2\right)\mathbf{A}^T\mathbf{A}$, and noise, $\sigma_{\mathrm{x}}^2\mathbf{A}^T\mathbf{A}$, terms, giving the signal to noise ratio as $\sqrt{\sigma_{\mathrm{x}}^2/\left(1-\sigma_{\mathrm{x}}^2\right)} \approx \sigma_{\mathrm{x}}$. To obtain a value for σ_{x} we perform a simple estimation. We take a V1 simple cell that integrates N inputs from retinal ganglion cells (RGCs) (indirectly, via the LGN), each firing a Poisson spike train of average rate r, with a temporal integration window of Δt. In this case, the c.v. (which corresponds to σ_{x}) is

$$\sigma_{\mathrm{x}} = \frac{\text{s.d.}}{\text{mean}} = \frac{\sqrt{Nr\Delta t}}{Nr\Delta t} = \frac{1}{\sqrt{Nr\Delta t}} \tag{4.45}$$

Based on the literature, we set the values of the relevant constants as

$$r \sim 1 \text{ s}^{-1} \text{ (Zhang et al., 2009)}, \tag{4.46}$$

$$\Delta t \sim 10 \text{ to } 100 \text{ ms (Tripathy et al., 2015)}, \tag{4.47}$$

$$N \sim 100 \text{ to } 1000. \tag{4.48}$$

To obtain this range for N, we note that there are around 1000 RGCs in the stimulated region in Ray and Maunsell (2010). (This can be computed knowing the dependency of RGC density on eccentricity (Watson, 2014), and that the stimulus has s.d. 0.5 degrees, so the total area is around 1 degree^2, and is 3 to 5 degrees from the fovea, and then discounting, to account for the fact that not all of these cells will be connected (Reid et al., 1995)). Thus, we obtain the interval

$$\sigma_{\mathrm{x}} = \frac{1}{\sqrt{1}} \text{ to } \frac{1}{\sqrt{100}} \tag{4.49}$$

of which we use the geometric mean:

$$\sigma_{\mathrm{x}} = \frac{1}{\sqrt{10}} \tag{4.50}$$

To choose values for τ_L, τ and σ_{v}^2, we considered biological constraints. The external input to the inhibitory cells is governed entirely by τ, suggesting that a biologically plausible value for τ is 10 ms (Tripathy et al., 2014). The scale of the recurrent input terms are governed by the product $\frac{1}{\tau}\mathbf{M}^{-1}$, suggesting that, to ensure the recurrent input has a biologically plausible timescale of 10 ms, we should set $\mathbf{M}^{-1}$ to be O(1) (see Eq. (4.44)).

Finally, we estimated τ_L, or equivalently the amount of noise per unit time, by comparing the rate at which membrane potential variance increases in our equations, $2\sigma^2/\tau_L$, to the rate of increase given by stochastic vesicle release, the primary source of 'noise' in cortical circuits. If a neuron is connected to s presynaptic neurons, firing with average rate r, and the variance of a unitary EPSP is v, then stochastic vesicle release introduces variance at the rate srv. Setting $srv = 2\sigma^2/\tau_L$ allows us to find the Langevin timescale

$$\tau_L = \frac{2\sigma^2}{srv} \tag{4.51}$$

However, estimating τ_L is difficult, because there are huge uncertainties in σ, s, r and v. We therefore wrote our uncertainty about each parameter as a log-normal distribution, $P(\log x) = \mathcal{N}\left(\log x; \mu_x, \sigma_{\mathrm{x}}^2\right)$ where x is one of σ, s, r, or v, and computed the induced distribution on τ_L. To specify the distributions, we wrote a range, from x_{l} to x_{h}, that, we believed contained around 95% of the probability mass, taking the boundaries of the range to be two standard-deviations from the mean in the log-domain, $\log x_{\mathrm{l}} = \mu_x - 2\sigma_{\mathrm{x}}$ and $\log x_{\mathrm{h}} = \mu_x + 2\sigma_{\mathrm{x}}$.

To estimate the required ranges, we took values from the neuroscience literature. First, estimates of firing rates vary widely, from around 0.5 Hz (Mizuseki and Buzsáki, 2013) to around 10 Hz (O'Connor et al., 2010). Second, the number of synapses per cell is usually taken to be around 10000. However, it is likely that there are multiple synapses per connection (Branco and Staras, 2009), so there

could be anywhere from 1000 to 10000 input cells for a single downstream neuron. Third, the average variance per spike is relatively easy to measure, data from Song *et al.* (Song et al., 2005) put the value at 0.076 mV^2. As other measurements seem roughly consistent (Bremaud et al., 2007), we use a relatively narrow range for v, from 0.05 mV^2 to 0.1 mV^2. Finally, the scaling factor, σ, could plausibly range from 2.5 mV to 7.5 mV, giving a full (2 standard deviations, and both sides of the mean) range of membrane potential fluctuations of 10 mV to 30 mV (Stern et al., 1997).

These ranges give a central estimate of $\tau_L = 150$ ms, which we used in our simulations. In agreement with this back-of-the-envelope calculation, we find that our sampler's dynamics match neural dynamics when τ_L lies in a broad range, from around 60 ms to around 400 ms (see S1 Figure). While τ_L appears relatively large in comparison with typical neural timescales, which are often around 10 ms, it should be remembered that τ_L parameterises the amount of noise injected into the network at every time step, and as such, does not therefore have any necessary link to other neural time constants.

Altering the model so that u_i and v_i are always positive

One might worry that it is possible for u_i (or v_i) to go negative, meaning that they have their influence on downstream neurons will have the wrong sign. However, it is straightforward to offset $\mathbf{u}$ (and hence $\mathbf{v}$, through Eq. (4.12)), so that they rarely, if ever become negative. Moreover, if we introduce the offset as

$$P(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{b}, \mathbf{C}) \tag{4.52}$$

$$P(\mathbf{x}|\mathbf{u}, z) = \mathcal{N}(\mathbf{x}; \mathbf{A}(\mathbf{u} - \mathbf{b}), \mathbf{C}) \tag{4.53}$$

then this leaves the data distribution $P(\mathbf{x})$, and hence the dynamics intact.

Deriving the 1D approximate model

$$\dot{\mathbf{u}} = \frac{1}{\tau}\mathbf{M}(\mathbf{u} - \mathbf{v}) \tag{4.54}$$

$$\dot{\mathbf{v}} = \frac{1}{\tau}\mathbf{M}(\mathbf{u} - \mathbf{v}) - \frac{z}{\sigma_{\mathrm{x}}^2}\mathbf{A}^T(\mathbf{x} - z\mathbf{A}\mathbf{u}) - \mathbf{C}\mathbf{u} \tag{4.55}$$

Differentiating again yields

$$\ddot{\mathbf{u}} = \frac{1}{\tau}\mathbf{M}(\dot{\mathbf{u}} - \dot{\mathbf{v}}) \tag{4.56}$$

substituting for $\dot{\mathbf{u}}$ and $\dot{\mathbf{v}}$, and collecting the terms that depend on $\mathbf{u}$, we obtain

$$\ddot{\mathbf{u}} = -\frac{1}{\tau^2}\mathbf{M}\left(\frac{z^2}{\sigma_{\mathrm{x}}^2}\mathbf{A}^T\mathbf{A} - \mathbf{C}^{-1}\right)(\mathbf{u} - \bar{\mathbf{u}}) \tag{4.57}$$

where $\bar{\mathbf{u}}$ is the posterior mean of $\mathbf{u}$ with fixed z (see Eq. (4.33) (4.37) and (4.42))

$$\bar{\mathbf{u}} = \frac{z}{\sigma_{\mathrm{x}}^2}\left(\frac{z^2}{\sigma_{\mathrm{x}}^2} + \frac{1}{1-\sigma_{\mathrm{x}}^2}\right)\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{x} \tag{4.58}$$

substituting $\mathbf{M} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}$ (i.e. the ideal value for $\mathbf{M}$), and $\mathbf{C} = \left(1-\sigma_{\mathrm{x}}^2\right)\left(\mathbf{A}^T\mathbf{A}\right)^{-1}$ (Eq. (4.36)), gives

$$\ddot{\mathbf{u}} = -\frac{1}{\tau^2}\left(\frac{z^2}{\sigma_{\mathrm{x}}^2} + \frac{1}{1-\sigma_{\mathrm{x}}^2}\right)(\mathbf{u} - \bar{\mathbf{u}}) \tag{4.59}$$

Thus, for fixed z, each component of $\mathbf{u}$ evolves independently.

Simulation Protocol

We simulated stimulus onset by first running the sampler until it reached equilibrium with no stimulus, then turning on the stimulus. To represent no stimulus we sampled $\mathbf{x}$ from $P(\mathbf{x}|z=0)$, and to represent stimulus, we sampled $\mathbf{x}$ from $P(\mathbf{x}|z=z_{\mathrm{gen}})$, where $z_{\mathrm{gen}} \in \{0.5, 1, 2\}$.

Computing LFPs and firing rates

To make contact with experimental data, we also computed local field potentials (LFPs), and firing rates. There are many methods for computing LFPs, we chose the simplest, averaging the membrane potentials across neurons, as it gave similar results to the other methods, without tuneable parameters. To compute firing rates, we used a rectified linear function of the membrane potential:

$$f_i(t) = \begin{cases} u_i(t) & \text{if } u_i(t) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.60}$$

4.6 Supplementary Figure

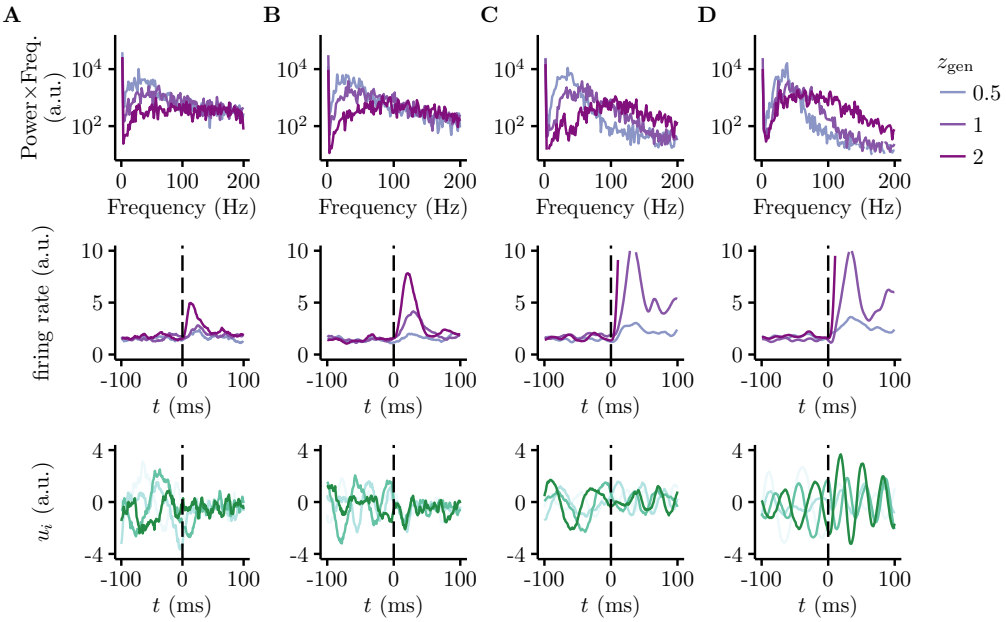

Figure 4.8: Our main results are robust to a range of ρ or equivalently τ_L. The top row is a power spectrum, the middle row displays the firing rate transient at stimulus onset, and the bottom row displays the membrane potential at stimulus onset for multiple trials and one neuron. The different lines in the first two rows correspond to different values of z_{gen}. In the bottom row, different lines correspond to different trials. **A.** For $\tau_L = 30$ ms, transients are small or non-existent, and no clear trends are present in the peak frequency. **B-C.** For $\tau_L = 60$ ms (**B**), and $\tau_L = 400$ ms (**C**) the results are similar to those in the main text. **D.** For $\tau_L = 1000$ ms, the results are quite different to those in the main text. In particular, the transient at stimulus onset lasts a long time, certainly longer than the observed value of around 50 ms.

Summary and future work

I started by taking normative theories of the brain, and attempting to either provide testable predictions, or, if that was not possible, to bring the theories closer to biologically realistic neural circuits that might prove testable in future.

First I attempted to test the hypothesis that confidence is Bayes optimal, and showed that the matter is complicated, with Bayesian inference providing the best explanation for the data in some circumstances, but not in others.

Second, I considered whether neural activity can or should be understood by referring to concepts from physics such as criticality. We found that some signatures of criticality, particularly Zipf's law, emerge simply because there are underlying latent variables, such as firing rates, and not necessarily because of any underlying analogy to critical systems.

Third, to provide testable predictions, I applied normative, Bayesian theoretical concepts to the neural synapse. I started by considering how the synapse might use Bayes theorem in order to learn more rapidly, which I called Bayesian Plasticity. I considered two predictions made by Bayesian plasticity, though there are certainly others. First, we predicted that synapses with lower presynaptic firing rates would have more uncertainty, and hence use a higher learning rate. Second, we predicted that at times when more presynaptic cells are active, learning rates should be lower, because it is difficult to know which synapse is responsible for any error. In the second chapter, I looked at how the synapse might communicate its uncertainty, derived by Bayes theorem, to downstream circuits. In particular, I supposed that the synapse might use increased variability to indicate higher uncertainty. This gave us a further one further prediction that had some experimental support, that synapses with higher presynaptic firing rates should have lower variability.

Finally, to bring normative theories of neural circuits closer to biological reality, I considered how such theories could be integrated with fundamental structural and dynamical properties of neural circuits, EI structure and oscillations. To this end, I showed that a widely used, and highly efficient algorithm, Hamiltonian Monte Carlo, can readily be mapped onto an EI network which exhibits

oscillations. Furthermore, this theory predicts, as we observe, that oscillation frequency and transient size increases with image contrast. Thus, thus allows us to make predictions not only about stationary activity (as with previous normative theories), but also dynamics.

These last two projects open out space for a large array of future work. First, the work on synapses is promising because we applied Bayes theorem, not to a whole cell (as is typical), but to a single synapse. This enables us to give local, biologically plausible learning rules whatever information the synapse might receive — perhaps the voltage and membrane potential in a full biophysical model of the cell. Thus, this approach should, in future, enable us to bring together normative computation theories with complex, messy biological reality. Second, the work on neural circuits is interesting because it raises the question of whether the brain's dynamics can give clues as to the brain's algorithm (and hence computation). In particular, it may be possible, by generalising our HMC approach, to show which dynamical sampling algorithms are compatible and incompatible with observed neural dynamics.

Bibliography

L. F. Abbott and S. B. Nelson. Synaptic plasticity: taming the beast. Nature Neuroscience, 3:1178–1183, 2000. (page 106)

W. C. Abraham. Metaplasticity: tuning synapses and networks for plasticity. Nature Reviews Neuroscience, 9(5):387, 2008. (page 107)

W. C. Abraham and M. F. Bear. Metaplasticity: the plasticity of synaptic plasticity. Trends in Neurosciences, (4):126–130, 1996. (page 107)

D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. Cognitive Science, 9(1):147–169, 1985. (page 18)

W. J. Adams, E. W. Graf, and M. O. Ernst. Experience can change the 'light-from-above' prior. Nature Neuroscience, 7(10):1057–1058, 2004. (page 15)

L. Aitchison and M. Lengyel. The hamiltonian brain: efficient probabilistic inference with excitatory-inhibitory neural circuit dynamics. PLoS computational biology, 12:e1005186, 2016. (page 11)

L. Aitchison, D. Bang, B. Bahrami, and P. E. Latham. Doubly bayesian analysis of confidence in perceptual decision-making. PLoS computational biology, 11: e1004519, 2015. (page 11)

L. Aitchison, N. Corradi, and P. E. Latham. Zipfs law arises naturally when there are underlying, unobserved variables. PLoS computational biology, 12: e1005110, 2016. (page 11)

D. Alais, F. N. Newell, and P. Mamassian. Multisensory processing in review: from physiology to behaviour. Seeing and Perceiving, 23(1):3–38, 2010. (page 15)

J. S. Albus. A theory of cerebellar function. Mathematical Biosciences, 10(1): 25–61, 1971. (page 95)

K. M. Armstrong and T. Moore. Rapid enhancement of visual cortical response discriminability by microstimulation of the frontal eye field. Proceedings of the National Academy of Sciences, 104(22):9499–9504, 2007. (page 136)

J. J. Atick and A. N. Redlich. Towards a theory of early visual processing. Neural Computation, 2(3):308–320, 1990. (page 13)

J. J. Atick, Z. Li, and A. N. Redlich. Understanding retinal color coding from first principles. Neural Computation, 4(4):559–572, 1992. (page 13)

F. Attneave. Some informational aspects of visual perception. Psychological Review, 61(3), 1954. (pages 12 and 13)

R. L. Axtell. Zipf distribution of US firm sizes. Science, 293:1818–1820, 2001. (pages 51 and 52)

B. Bahrami, K. Olsen, P. E. Latham, A. Roepstorff, G. Rees, and C. D. Frith. Optimally interacting minds. Science, 329(5995), 2010. (page 23)

D. Bang, R. Fusaroli, K. Tylén, K. Olsen, P. E. Latham, J. Y. F. Lau, A. Roepstorff, G. Rees, C. Frith, and B. Bahrami. Does interaction matter? testing whether a confidence heuristic can replace interaction in collective decision-making. Consciousness and Cognition, 26(1), 2014. (page 45)

B. Barber and T. Odean. Boys will be boys: gender, overconfidence, and common stock investment. The Quarterly Journal of Economics, 116(1), 2001. (page 45)

H. Barlow. Possible principles underlying the transformations of sensory messages. In W. Rosenblith, editor, Sensory Communication, pages 217–234. MIT Press, 1961. (pages 12 and 13)

D. G. Barrett, S. Deneve, and C. K. Machens. Firing rate predictions in optimal balanced networks. In Advances in Neural Information Processing Systems, pages 1538–1546, 2013. (page 16)

S. Barthelmé and P. Mamassian. Evaluation of objective uncertainty in the visual system. PLoS Computational Biology, 5(9), 2009. (page 44)

S. Barthelmé and P. Mamassian. Flexible mechanisms underlie the evaluation of visual confidence. Proceedings of the National Academy of Sciences, 107(48), 2010. (page 44)

E. Basar and B. Guntekin. A review of brain oscillations in cognitive disorders and the role of neurotransmitters. Brain Research, 1235:172–193, 2008. (page 136)

T. Bayes and R. Price. An essay towards solving a problem in the doctrine of chances. Philosophical Transactions, 53:370–418, 1763. (page 9)

C. Beck and E. G. D. Cohen. Superstatistics. Physica A: Statistical Mechanics and Its Applications, 322:267–275, 2003. (page 70)

J. Beck, W. J. Ma, P. E. Latham, and A. Pouget. Probabilistic population codes and the exponential family of distributions. Progress in Brain Research, 165: 509–519, 2007. (page 17)

J. Beck, A. Pouget, and K. A. Heller. Complex Inference in Neural Circuits with Probabilistic Population Codes and Topic Models. In Advances in Neural Information Processing Systems 25, pages 3059–3067. 2012. (page 18)

J. M. Beck, W. J. Ma, R. Kiani, T. Hanks, A. K. Churchland, J. Roitman, M. N. Shadlen, P. E. Latham, and A. Pouget. Probabilistic population codes for Bayesian decision making. Neuron, 60(6):1142–1152, 2008. (pages 17 and 136)

J. M. Beck, P. E. Latham, and A. Pouget. Marginalization in neural circuits with divisive normalization. The Journal of Neuroscience, 31(43):15310–15319, 2011. (page 136)

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. Neural Computation, 7(6):1129–1159, 1995. (page 13)

Y. Bengio, D.-H. Lee, J. Bornschein, and Z. Lin. Towards biologically plausible deep learning. arXiv:1502.04156, 2015. (page 14)

P. Berkes, R. E. Turner, and M. Sahani. A structured model of video reproduces primary visual cortical organisation. PLoS Computational Biology, 5:e1000495, 2009. (page 139)

P. Berkes, J. Fiser, G. Orbán, and M. Lengyel. Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. Science, 331(6013):83–87, 2011a. (pages 106 and 119)

P. Berkes, G. Orbán, M. Lengyel, and J. Fiser. Spontaneous Cortical Activity Reveals Hallmarks of an Optimal Internal Model of the Environment. Science, 331(6013):83–87, 2011b. (pages 10, 21, 135, and 155)

E. J. Bermudez Contreras, A. G. P. Schjetnan, A. Muhammad, P. Bartho, B. L. McNaughton, B. Kolb, A. J. Gruber, and A. Luczak. Formation and reverberation of sequential neural activity patterns evoked by sensory stimulation are enhanced during cortical desynchronization. Neuron, 79(3):555–566, 2013. (page 148)

E. Berner and M. Graber. Overconfidence as a cause of diagnostic error in medicine. The American Journal of Medicine, 121(5), 2008. (page 23)

G.-Q. Bi and M.-M. Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. Journal of Neuroscience, 18(24):10464–10472, 1998. (page 106)

W. Bialek, I. Nemenman, and N. Tishby. Predictability, complexity, and learning. Neural Computation, 13:2409–2463, 2001. (page 65)

T. Binzegger, R. Douglas, and K. Martin. A quantitative map of the circuit of cat primary visual cortex. The Journal of Neuroscience, 24:8441–8453, 2004. (page 113)

C. Blundell, J. Cornebise, K. Kavukcuoglu, and W. Dean. Weight uncertainty in neural networks. arXiv:1505.05424, 2015. (page 107)

M. Boerlin and S. Denève. Spike-based population coding and working memory. PLoS Computational Biology, 7(2):e1001080, 2011. (page 16)

M. Boerlin, C. K. Machens, and S. Denève. Predictive coding of dynamical variables in balanced spiking networks. 2013. (page 16)

I. Bomash, Y. Roudi, and S. Nirenberg. A virtual retina for studying population coding. PLoS One, 8:e53363, 2013. (page 75)

B. G. Borghuis, C. P. Ratliff, R. G. Smith, P. Sterling, and V. Balasubramanian. Design of a neuronal array. The Journal of Neuroscience, 28(12):3178–3189, 2008. (page 13)

S. Boucheron, O. Bousquet, G. Lugosi, and P. Massart. Concentration Inequalities: A Nonasymptotic Theory of Independence. Oxford University Press, 2013. (page 85)

R. Bourdoukan, D. Barrett, S. Deneve, and C. K. Machens. Learning optimal spike-based representations. In Advances in Neural Information Processing Systems, pages 2285–2293. 2012. (page 16)

T. Branco and K. Staras. The probability of neurotransmitter release: variability and feedback control at single synapses. Nature Reviews Neuroscience, 10: 373–383, 2009. (pages 82, 95, and 158)

L. Breiman. Random forests. Machine Learning, 45(1):5–32, 2001. (page 9)

A. Bremaud, D. C. West, and A. M. Thomson. Binomial parameters differ across neocortical layers and with different classes of connections in adult rat and cat neocortex. Proceedings of the National Academy of Sciences, 104(35):14134–14139, 2007. (page 159)

M. Broihanne, M. Merli, and P. Roger. Overconfidence, risk perception and the risk-taking behavior of finance professionals. Finance Research Letters, 11(2), 2014. (pages 23 and 45)

S. R. Bruesch and L. B. Arey. The number of myelinated and unmyelinated fibers in the optic nerve of vertebrates. Journal of Comparative Neurology, 77(3): 631–665, 1942. (page 13)

B. W. Brunton, M. M. Botvinick, and C. D. Brody. Rats and humans can optimally accumulate evidence for decision-making. Science, 340(6128):95–98, 2013. (pages 15 and 16)

H. L. Bryant and J. P. Segundo. Spike initiation by transmembrane current: a white-noise analysis. Journal of Physiology, 260(2):279–314, 1976. (page 113)

L. Buesing, J. Bill, B. Nessler, and W. Maass. Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons. PLoS Computational Biology, 7(11), 2011. (pages 20 and 136)

W. L. Buntine and A. S. Weigend. Bayesian backpropagation. Complex Systems, 5(6):603–643, 1991. (page 107)

G. Buzsaki. Rhythms of the Brain. Oxford University Press, 2006. (page 146)

G. Buzsáki and B. O. Watson. Brain rhythms and neural syntax: implications for efficient coding of cognitive content and neuropsychiatric disease. Dialogues in Clinical Neuroscience, 14:345, 2012. (page 151)

R. F. Cancho i and R. V. Solé. Least effort and the origins of scaling in human language. Proceedings of the National Academy of Sciences, 100:788–791, 2003. (pages 52, 59, and 75)

A. Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. Comp. Rend. Sci. Paris, 25(1847):536–538, 1847. (page 16)

N. Chater, J. B. Tenenbaum, and A. Yuille. Probabilistic models of cognition: Conceptual foundations. Trends in Cognitive Sciences, 10(7):287–291, 2006. (page 135)

M. M. Churchland, B. M. Yu, J. P. Cunningham, L. P. Sugrue, M. R. Cohen, G. S. Corrado, W. T. Newsome, A. M. Clark, P. Hosseini, B. B. Scott, D. C. Bradley, M. A. Smith, A. Kohn, J. A. Movshon, K. M. Armstrong, T. Moore, S. W. Chang, L. H. Snyder, S. G. Lisberger, N. J. Priebe, I. M. Finn, D. Ferster, S. I. Ryu, G. Santhanam, M. Sahani, and K. V. Shenoy. Stimulus onset quenches neural variability: a widespread cortical phenomenon. Nature Neuroscience, 13(3):369–378, 2010. (pages 10, 21, and 140)

D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. Neural Computation, 22 (12):3207–3220, 2010. (page 14)

R. Coen-cagli, P. Dayan, and O. Schwartz. Statistical models of linear and non-linear contextual interactions in early visual processing. In Advances in Neural Information Processing Systems, pages 369–377, 2009. (page 140)

R. Coen-Cagli, P. Dayan, and O. Schwartz. Cortical surround interactions and perceptual salience via natural scene statistics. PLoS Computational Biology, 8(3):e1002405, 2012. (page 135)

B. Corominas-Murtra, J. Fortuny, and R. V. Solé. Emergence of Zipf's law in the evolution of communication. Physical Review E, 83:036115, 2011. (pages 52, 59, and 75)

T. M. Cover and J. A. Thomas. Elements of information theory. John Wiley & Sons, 1991. (page 12)

P. Dayan. Recurrent sampling models for the Helmholtz machine. Neural Computation, 11(3):653–677, 1998. (page 20)

P. Dayan and L. F. Abbott. Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. MIT Press, 2001. (pages 13, 95, and 155)

P. Dayan and G. E. Hinton. Varieties of Helmholtz machine. Neural Networks, 9(8):1385–1403, 1996. (page 20)

P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The Helmholtz machine. Neural Computation, 7(5):889–904, 1995. (page 20)

V. de Gardelle and P. Mamassian. Does confidence use a common currency across two visual tasks? Psychological Science, 25(6), 2014. (page 44)

B. de Martino, S. Fleming, and R. Garrett N. Dolan. Confidence in value-based choice. Nature Neuroscience, 16(1), 2013. (page 26)

P. Dean, J. Porrill, C.-F. Ekerot, and H. Jörntell. The cerebellar microcircuit as an adaptive filter: experimental and computational evidence. Nature Reviews Neuroscience, 11(1):30–43, 2010. (page 119)

G. Deco and E. Hugues. Neural network mechanisms underlying stimulus driven variability reduction. PLoS Computational Biology, 8(3):e1002395–e1002395, 2012. (page 21)

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (methodological), pages 1–38, 1977. (page 155)

S. Deneve, P. E. Latham, and A. Pouget. Reading population codes: a neural implementation of ideal observers. Nature Neuroscience, 2(8):740–745, 1999. (page 135)

L. Deng. Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing, 7(3-4):197–387, 2014. (page 14)

J. Desponds, T. Mora, and A. M. Walczak. Fluctuating fitness shapes the clone-size distribution of immune repertoires. Proceedings of the National Academy of Sciences, 113:274–279, 2016. (pages 64 and 73)

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. Physics Letters B, 195(2):216–222, 1987. (page 136)

J. Eccles, R. Llinas, and K. Sasaki. The excitatory synaptic action of climbing fibres on the purkinje cells of the cerebellum. Journal of Physiology, 182(2): 268–296, 1966. (page 95)

B. Efron and C. Stein. The jackknife estimate of variance. Annals of Statistics, 9:586–596, 1981. (page 85)

M. O. Ernst and M. S. Banks. Humans integrate visual and haptic information in a statistically optimal fashion. Nature, 415(6870):429–433, 2002. (pages 15, 16, 117, and 135)

A. Feller and A. Gelman. Hierarchical Models for Causal Effects. Emerging Trends in the Social and Behavioral Sciences, 2014. (page 9)

J. Fiser, P. Berkes, G. Orbán, and M. Lengyel. Statistically optimal perception and learning: from behavior to neural representations. Trends in Cognitive Sciences, 14(3):119–130, 2010. (pages 10, 106, 119, and 136)

S. Fleming and R. Dolan. Effects of loss aversion on post-decision wagering: implications for measures of awareness. Consciousness and Cognition, 19(1), 2010. (pages 45 and 46)

S. Fleming and H. Lau. How to measure metacognition. Frontiers in Human Neuroscience, 8(443), 2014. (page 46)

S. Fleming, R. Dolan, and C. Frith. Metacognition: computation, biology and function. Philosophical Transactions of the Royal Society B: Biological Sciences, 367(1594), 2012. (pages 23 and 26)

B. J. Frey and N. Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. Pattern Analysis and Machine Intelligence, IEEE Transactions On, 27(9):1392–1416, 2005. (page 10)

U. Frey and R. G. Morris. Synaptic tagging and long-term potentiation. Nature, 385(6616):533–536, 1997. (page 107)

P. Fries. Neuronal Gamma-Band Synchronization as a Fundamental Process in Cortical Computation. Annual Review of Neuroscience, 32(1):209–224, 2009. (page 152)

K. Fukumizu, L. Song, and A. Gretton. Kernel Bayes' rule: Bayesian inference with positive definite kernels. The Journal of Machine Learning Research, 14 (1):3753–3783, 2013. (page 18)

R. Fusaroli, B. Bahrami, K. Olsen, A. Roepstorff, G. Rees, C. Frith, and K. Tylen. Coming to terms: quantifying the benefits of linguistic coordination. Psychological Science, 23(8), 2012. (page 23)

X. Gabaix. Zipf's law for cities: an explanation. The Quarterly Journal of Economics, 114:739–767, 1999. (pages 51, 52, and 75)

X. Gabaix, P. Gopikrishnan, V. Plerou, and H. E. Stanley. A theory of power-law distributions in financial market fluctuations. Nature, 423:267–270, 2003. (page 51)

S. Galvin, J. Podd, V. Drga, and J. Whitmore. Type 2 tasks in the theory of signal detectability: discrimination between correct and incorrect decisions. Psychonomic Bulletin and Review, 10(4), 2003. (page 25)

P. Garrigan, C. P. Ratliff, J. M. Klein, P. Sterling, D. H. Brainard, and V. Balasubramanian. Design of a trichromatic cone array. PLoS Computational Biology, 6(2):e1000677, 2010. (page 13)

A. Gelb. Applied Optimal Estimation. MIT Press, 1974. (page 123)

Z. Ghahramani, M. J. Beal, and others. Graphical models and variational methods. Advanced Mean Field Method—Theory and Practice, pages 37–50, 2000. (page 10)

M. S. Goldman. Enhancement of information transmission efficiency by synaptic failures. Neural Computation, 16(6):1137–1162, 2004. (page 13)

A. Gopnik, C. Glymour, D. M. Sobel, L. E. Schulz, T. Kushnir, and D. Danks. A theory of causal learning in children: causal maps and Bayes nets. Psychological Review, 111(1):3, 2004. (page 135)

A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference On, pages 6645–6649. IEEE, 2013. (page 14)

D. M. Green and J. A. Swets. Signal Detection Theory and Psychophysics. New York: Wiley, 1966. (page 23)

R. M. Haefner, P. Berkes, and J. Fiser. Perceptual decision-making as probabilistic inference by neural sampling. Neuron, 90(3):649–660, 2016. (page 106)

N. Harvey. Confidence in judgement. Trends in Cognitive Sciences, 1(2), 1997. (pages 45 and 46)

G. Hennequin, L. Aitchison, and M. Lengyel. Fast sampling-based inference in balanced neuronal networks. In Advances in Neural Information Processing Systems 27, pages 2240–2248, 2014a. (page 143)

G. Hennequin, T. P. Vogels, and W. Gerstner. Optimal control of transient dynamics in balanced networks supports generation of complex movements. Neuron, 82:1394–1406, 2014b. (pages 142 and 143)

G. E. Hinton. To recognize shapes, first learn to generate images. Progress in Brain Research, 165:535–547, 2007. (page 14)

G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. Neural Computation, 18(7):1527–1554, 2006. (page 14)

G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In Artificial Neural Networks and Machine Learning–ICANN 2011, pages 44–51. Springer, 2011. (page 14)

P. O. Hoyer and A. Hyvarinen. Interpreting neural response variability as Monte Carlo sampling of the posterior. Advances in Neural Information Processing Systems, pages 293–300, 2003. (pages 10, 106, 119, and 136)

T. Hu, A. Genkin, and D. B. Chklovskii. A network of spiking neurons for computing sparse representations in an energy-efficient way. Neural Computation, 24(11):2852–2872, 2012. (page 16)

S. R. Hulme, O. D. Jones, C. R. Raymond, P. Sah, and W. C. Abraham. Mechanisms of heterosynaptic metaplasticity. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 369(1633):20130148, 2014. (page 107)

F. Huszár, U. Noppeney, and M. Lengyel. Mind reading by machine learning: A doubly bayesian method for inferring mental representations. In Proceedings of the Thirty-second Annual Conference of the Cognitive Science Society, pages 2810–2815, 2010. (page 26)

A. Hyvärinen. Statistical models of natural images and cortical visual representation. Topics in Cognitive Science, 2(2):251–264, 2010. (page 135)

N. Intrator and L. N. Cooper. Objective Function Formulation of the BCM Theory of Visual Cortical Plasticity: Statistical Connections, Stability Conditions. Neural Networks, 5:3–17, 1992. (page 112)

Y. M. Ioannides and H. G. Overman. Zipf's law for cities: an empirical examination. Regional Science and Urban Economics, 33:127–137, 2003. (page 52)

M. Ito, M. Sakurai, and P. Tongroach. Climbing fibre induced depression of both mossy fibre responsiveness and glutamate sensitivity of cerebellar Purkinje cells. Journal of Physiology, 324(1):113–134, 1982. (page 95)

R. A. Jacobs. Optimal integration of texture and motion cues to depth. Vision Research, 39(21):3621–3629, 1999. (page 135)

H. Jacobson. The informational capacity of the human eye. Science, 113(2933): 292–293, 1951. (page 9)

D. Johnson. Overconfidence and War: The Havoc and Glory of Positive Illusions. Cambridge, Mass.: Harvard University Press, 2004. (page 23)

D. Kahneman. Attention and Effort. Englewood Cliffs, New Jersey: Prentice Hall, 1973. (pages 26 and 43)

D. Kappel, S. Habenschuss, R. Legenstein, and W. Maass. Network plasticity as bayesian inference. PLoS Computational Biology, 11(11):e1004485, 2015. (page 107)

Y. Karklin and M. S. Lewicki. Emergence of complex cell properties by learning to generalize in natural scenes. Nature, 457:83–86, 2009. (page 139)

Y. Karklin and M. S. Lewicki. Emergence of complex cell properties by learning to generalize in natural scenes. Nature, 457(7225):83–86, 2009. (page 135)

Y. Karklin and E. P. Simoncelli. Efficient coding of natural images with a population of noisy linear-nonlinear neurons. In Advances in Neural Information Processing Systems, pages 999–1007, 2011. (page 13)

H. Kasai, N. Takahashi, and H. Tokumaru. Distinct initial snare configurations underlying the diversity of exocytosis. Physiological Reviews, 92:1915–1964, 2012. (page 107)

A. Kepecs, N. Uchida, H. Zariwala, and Z. Mainen. Neural correlates, computation and behavioral impact of decision confidence. Nature, 455(7210), 2008. (pages 22 and 26)

R. Kiani and M. Shadlen. Representation of confidence associated with a decision by neurons in the parietal cortex. Science, 324(5928), 2009. (page 22)

D. C. Knill. Surface orientation from texture: ideal observers, generic observers and the information content of texture cues. Vision Research, 38(11):1655–1682, 1998. (page 135)

D. C. Knill and W. Richards. Perception as Bayesian Inference. Cambridge University Press, 1996. (page 93)

K. H. Knuth and J. Skilling. Foundations of inference. Axioms, 1(1):38–73, 2012. (page 9)

H. Ko, S. B. Hofer, B. Pichler, K. A. Buchanan, P. J. Sjöström, and T. D. Mrsic-Flogel. Functional specificity of local synaptic connections in neocortical networks. Nature, 473:87–91, 2011. (pages 10, 103, 105, 106, and 119)

D. Koller and N. Friedman. Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge, MA, 2009. (page 15)

Y. Komura, A. Nikkuni, N. Hirashima, T. Uetake, and A. Miyamoto. Responses of pulvinar neurons reflect a subject's confidence in visual categorization. Nature Neuroscience, 16(6), 2013. (page 22)

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, pages 1097–1105, 2012. (page 14)

D. M. Kullmann, A. W. Moreau, Y. Bakiri, and E. Nicholson. Plasticity of inhibition. Neuron, 75(6):951–962, 2012. (page 152)

C. Kunimoto, J. Miller, and H. Pashler. Confidence and accuracy of near-threshold discrimination responses. Consciousness and Cognition, 10(3), 2001. (page 25)

K. P. Körding and D. M. Wolpert. Bayesian integration in sensorimotor learning. Nature, 427(6971):244–247, 2004. (page 135)

K. P. Körding and D. M. Wolpert. Bayesian decision theory in sensorimotor control. Trends in Cognitive Sciences, 10(7):319–326, 2006. (page 15)

S. B. Laughlin. A simple coding procedure enhances a neuron's information capacity. Z. Naturforsch, 36(910-912):51, 1981. (page 13)

N. Lavie. Distracted and confused: selective attention under load. Trends in Cognitive Sciences, (2), 2005. (pages 26 and 43)

G. Leech, P. Rayson, and A. Wilson. Word frequencies in written and spoken English: based on the British National Corpus. Longman, Harlow, 2001. (pages 60 and 77)

B. Levin. English verb classes and alternations: A preliminary investigation. University of Chicago press, 1993. (pages 59 and 60)

M. S. Lewicki. Efficient coding of natural sounds. Nature Neuroscience, 5(4): 356–363, 2002. (page 16)

W. Li. Random texts exhibit Zipf's-law-like word frequency distribution. IEEE Transactions on Information Theory, 38:1842–1845, 1992. (pages 52, 59, 61, 62, 73, 75, and 78)

Z. Li and P. Dayan. Computational differences between asymmetrical and symmetrical networks. Network, 10(1):59–77, 1999. (page 136)

T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman. Random feedback weights support learning in deep neural networks. arXiv:1411.0247, 2014. (page 14)

A. Loebel, I. Nelken, and M. Tsodyks. Processing of sounds by population spikes in a model of primary auditory cortex. 2007. (page 142)

Y. Loewenstein, A. Kuras, and S. Rumpel. Multiplicative dynamics underlie the emergence of the log-Normal distribution of spine sizes in the neocortex in vivo. Journal of Neuroscience, 31(26):9481–9488, 2011. (page 110)

A. Luczak, P. Bartho, and K. D. Harris. Gating of sensory input by spontaneous cortical activity. The Journal of Neuroscience, 33(4):1684–1695, 2013. (pages 136 and 148)

W. Ma and M. Jazayeri. Neural coding of uncertainty and probability. Annual Review of Neuroscience, 37, 2014. (page 23)

W. J. Ma, J. M. Beck, P. E. Latham, and A. Pouget. Bayesian inference with probabilistic population codes. Nature Neuroscience, 9(11):1432–1438, 2006. (pages 17, 106, and 136)

D. J. MacKay. A practical Bayesian framework for backpropagation networks. Neural Computation, 4(3):448–472, 1992. (page 107)

D. J. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. Technical report, 1996. (page 13)

J. H. Macke, L. Buesing, J. P. Cunningham, M. Y. Byron, K. V. Shenoy, and M. Sahani. Empirical models of spiking in neural populations. In Advances in Neural Information Processing Systems, pages 1350–1358, 2011a. (page 143)

J. H. Macke, M. Opper, and M. Bethge. Common input explains higher-order correlations and entropy in a simple model of neural population activity. Physical Review Letters, 106(20):208102, 2011b. (page 74)

A. Mahmoodi, D. Bang, M. N. Ahmadabadi, and B. Bahrami. Learning to make collective decisions: the impact of confidence escalation. PLoS One, 8(12), 2013. (page 45)

Z. F. Mainen and T. J. Sejnowski. Reliability of spike timing in neocortical neurons. Science, 268(5216):1503–1506, 1995. (page 113)

B. Mandelbrot. An informational theory of the statistical structure of languages. In B. W. Jackson, editor, Communication Theory, pages 486–502, 1953. (pages 52, 59, and 75)

B. Maniscalco and H. Lau. A signal detection theoretic approach for estimating metacognitive sensitivity from confidence ratings. Consciousness and Cognition, 21, 2012. (page 25)

L. Mann, M. Radford, P. Burnett, S. Ford, M. Bond, K. Leung, H. Nakamura, G. Vaughan, and K.-S. Yang. Cross-cultural differences in self-reported decision-making style and confidence. International Journal of Psychology, 33 (5), 1998. (page 45)

H. Markram, W. Gerstner, and P. J. Sjöström. Spike-Timing-Dependent Plasticity: A Comprehensive Overview. Frontiers in Synaptic Neuroscience, 4, 2012. (page 152)

D. Marr and T. Poggio. From Understanding Computation to Understanding Neural Circuitry. AI Memos, 1976. (page 9)

M. Matsuzaki, N. Honkura, G. C. Ellis-Davies, and H. Kasai. Structural basis of long-term potentiation in single dendritic spines. Nature, 429(6993):761–766, 2004. (page 110)

K. Michel, J. A. Müller, A.-M. Oprisoreanu, and S. Schoch. The presynaptic active zone: A dynamic scaffold that regulates synaptic efficacy. Experimental Cell Research, 335:157–164, 2015. (page 107)

E. Milotti. 1/f noise: a pedagogical review. arXiv:physics/0204033, 2002. (page 148)

T. P. Minka. A family of algorithms for approximate Bayesian inference. PhD thesis, Massachusetts Institute of Technology, 2001. (page 115)

K. Mizuseki and G. Buzsáki. Preconfigured, Skewed Distribution of Firing Rates in the Hippocampus and Entorhinal Cortex. Cell Reports, 4(5):1010–1021, 2013. (pages 111 and 158)

A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. arXiv:1402.0030, 2014. (page 20)

D. A. Moore and P. J. Healy. The trouble with overconfidence. Psychological Review, (2), 2008. (page 46)

T. Mora and W. Bialek. Are biological systems poised at criticality? Journal of Statistical Physics, 144:268–302, 2011. (pages 51, 52, 54, 55, 56, 68, 72, and 75)

T. Mora, A. M. Walczak, W. Bialek, and C. G. Callan. Maximum entropy models for antibody diversity. Proceedings of the National Academy of Sciences, 107: 5405–5410, 2010. (pages 51, 61, 63, and 73)

M. J. Mulder, E.-J. Wagenmakers, R. Ratcliff, W. Boekel, and B. U. Forstmann. Bias in the Brain: A Diffusion Model Analysis of Prior Probability and Potential Payoff. The Journal of Neuroscience, 32(7):2335–2343, 2012. (page 15)

B. K. Murphy and K. D. Miller. Balanced amplification: a new mechanism of selective amplification of neural activity patterns. Neuron, 61:635–648, 2009. (page 142)

J. R. Müller, A. B. Metha, J. Krauskopf, and P. Lennie. Rapid adaptation in visual cortex to the structure of images. Science, 285(5432):1405–1408, 1999. (pages 136 and 151)

J. R. Müller, A. B. Metha, J. Krauskopf, and P. Lennie. Information conveyed by onset transients in responses of striate cortical neurons. The Journal of Neuroscience, 21(17):6978–6990, 2001. (pages 136, 148, and 151)

R. Neal. MCMC for Using Hamiltonian Dynamics. Handbook of Markov Chain Monte Carlo, pages 113–162, 2011. (pages 136, 143, 151, and 156)

R. M. Neal. Sampling from multimodal distributions using tempered transitions. Statistics and Computing, 6:353–366, 1996. (page 151)

M. E. Newman. Power laws, Pareto distributions and Zipf's law. Contemporary Physics, 46:323–351, 2005. (page 52)

S. Nirenberg and M. Meister. The light response of retinal ganglion cells is truncated by a displaced amacrine circuit. Neuron, 18:637–650, 1997. (page 75)

S. Nirenberg and C. Pandarinath. Retinal prosthetic strategy with the capacity to restore normal vision. Proceedings of the National Academy of Sciences, 109:15012–15017, 2012. (page 75)

M. Nonnenmacher, C. Behrens, P. Berens, M. Bethge, and J. H. Macke. Signatures of criticality arise in simple neural population models with correlations. arXiv:1603.00097, 2016. (page 74)

D. H. O'Connor, S. P. Peron, D. Huber, and K. Svoboda. Neural activity in barrel cortex underlying vibrissa-based object localization in mice. Neuron, 67 (6):1048–1061, 2010. (pages 111 and 158)

M. Okun and I. Lampl. Instantaneous correlation of excitation and inhibition during ongoing and sensory-evoked activities. Nature Neuroscience, 11(5):535–537, 2008. (pages 137, 146, and 151)

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381(6583):607–609, 1996. (pages 16 and 135)

G. Orbán, P. Berkes, J. Fiser, and M. Lengyel. Neural variability and sampling-based probabilistic representations in the visual cortex. Neuron, 92(2):530–543, 2016. (pages 106, 135, 136, 140, 141, 148, and 151)

G. Orbán and M. Lengyel. Sampling in the visual cortex: explaining (away) neural variability and spontaneous activity. COSYNE Abstract, 2011. (page 21)

G. Orbán and D. M. Wolpert. Representations of uncertainty in sensorimotor control. Current Opinion in Neurobiology, 21(4):629–635, 2011. (page 15)

G. Orbán, P.-O. Polack, P. Golshani, and M. Lengyel. Stimulus-dependence of membrane potential and spike count variability in V1 of behaving mice. COSYNE Poster, 2013. (page 10)

A. M. Packer, L. E. Russell, H. W. P. Dalgleish, and M. Häusser. Simultaneous all-optical manipulation and recording of neural circuit activity with cellular resolution in vivo. Nature Methods, 12(2):140–146, 2015. (pages 105 and 119)

H. Pashler. Dual-task interference in simple tasks: data and theory. Psychological Bulletin, (2), 1994. (pages 26 and 43)

R. Pathria and P. Beale. Statistical Mechanics. Elsevier, 3 edition, 2011. (pages 55, 56, and 67)

J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco, Calif., 1988. (page 15)

B. A. Pearlmutter and L. C. Parra. A context-sensitive generalization of ICA. Advances in Neural Information Processing Systems, 151, 1996. (page 13)

J.-P. Pfister and W. Gerstner. Triplets of spikes in a model of spike timing-dependent plasticity. Journal of Neuroscience, 26(38):9673–9682, 2006. (page 106)

J.-P. Pfister, P. Dayan, and M. Lengyel. Synapses with short-term plasticity are optimal estimators of presynaptic membrane potentials. Nature Neuroscience, 13:1271–1275, 2010. (pages 142 and 152)

T. J. Pleskac and J. R. Busemeyer. Two-stage dynamic signal detection: a theory of choice, decision time, and confidence. Psychological Review, 117(3): 864, 2010. (page 25)

T. Poggio. A theory of how the brain might work. In Cold Spring Harbor Symposia on Quantitative Biology, volume 55. Cold Spring Harbor Laboratory Press, 1990. (page 93)

R. Ponte Costa, R. C. Froemke, P. J. Sjöström, and M. C. W. van Rossum. Unified pre- and postsynaptic long-term plasticity enables reliable and flexible learning. Elife, 4, 2015. (page 106)

J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Adaptive Wiener denoising using a Gaussian scale mixture model in the wavelet domain. In International Conference on Image Processing, volume 2, pages 37–40. IEEE, 2001. (page 139)

A. Pouget, J. M. Beck, W. J. Ma, and P. E. Latham. Probabilistic brains: knowns and unknowns. Nature Neuroscience, 16(9):1170–1178, 2013. (pages 10, 15, 93, 106, 107, and 135)

D. Price. A general theory of bibliometric and other cumulative advantage processes. Journal of the American Society for Information Science, 27:292–306, 1976. (page 75)

R. P. Rao and D. H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience, 2(1):79–87, 1999. (pages 16, 135, and 142)

D. Raposo, J. P. Sheppard, P. R. Schrater, and A. K. Churchland. Multisensory Decision-Making in Rats and Humans. The Journal of Neuroscience, 32(11): 3726–3735, 2012. (pages 15 and 16)

C. P. Ratliff, B. G. Borghuis, Y.-H. Kao, P. Sterling, and V. Balasubramanian. Retina is structured to process an excess of darkness in natural scenes. Proceedings of the National Academy of Sciences, 107(40):17368–17373, 2010. (page 13)

S. Ray and J. H. Maunsell. Differences in Gamma Frequencies across Visual Cortex Restrict Their Possible Use in Computation. Neuron, 67(5):885–896, 2010. (pages 136, 137, 146, 148, 150, 151, and 158)

R. L. Redondo and R. G. M. Morris. Making memories last: the synaptic tagging and capture hypothesis. Nature Reviews Neuroscience, 12(1):17–30, 2011. (page 107)

M. Rehn and F. T. Sommer. A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. Journal of Computational Neuroscience, 22(2):135–146, 2007. (page 16)

R. C. Reid, J.-M. Alonso, et al. Specificity of monosynaptic connections from thalamus to visual cortex. Nature, 378:281–283, 1995. (page 158)

C. Robert and G. Casella. A short history of Markov Chain Monte Carlo: subjective recollections from incomplete data. Statistical Science, 26(1):102–115, 2011. (pages 9 and 10)

G. O. Roberts and R. L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. Bernoulli, 2(4):341–363, 1996. (page 143)

M. J. Roberts, E. Lowet, N. M. Brunet, M. Ter Wal, P. Tiesinga, P. Fries, and P. De Weerd. Robust Gamma Coherence between Macaque V1 and V2 by Dynamic Frequency Matching. Neuron, 78(3):523–536, 2013. (pages 137 and 146)

T. Rogerson, D. J. Cai, A. Frank, Y. Sano, J. Shobe, M. F. Lopez-Aranda, and A. J. Silva. Synaptic tagging during memory allocation. Nature Reviews Neuroscience, 15(3):157–169, 2014. (page 107)

C. J. Rozell, D. H. Johnson, R. G. Baraniuk, and B. A. Olshausen. Sparse coding via thresholding and local competition in neural circuits. Neural Computation, 20(10):2526–2563, 2008. (page 16)

D. B. Rubin, S. D. Van Hooser, and K. D. Miller. The stabilized supralinear network: a unifying circuit motif underlying multi-input integration in sensory cortex. Neuron, 85(2):402–417, 2015. (page 136)

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. Nature, 5, 1986. (page 14)

M. Sahani and P. Dayan. Doubly distributional population codes: simultaneous representation of uncertainty and multiplicity. Neural Computation, 15(10): 2255–2279, 2003. (pages 18 and 136)

S. Saremi and T. J. Sejnowski. Hierarchical model of natural images and the origin of scale invariance. Proceedings of the National Academy of Sciences, 110:3071–3076, 2013. (pages 52 and 75)

S. Saremi and T. J. Sejnowski. On criticality in high-dimensional data. Neural Computation, 26:1–11, 2014. (pages 52 and 75)

C. Savin, D. Peter, and M. Lengyel. Optimal recall from bounded metaplastic synapses: predicting functional adaptations in hippocampal area CA3. PLoS Computational Biology, 10:e1003489, 2014. (page 151)

J. Schmidhuber. Deep learning in neural networks: An overview. Neural Networks, 61:85–117, 2015. (page 14)

D. J. Schwab, I. Nemenman, and P. Mehta. Zipf's law and criticality in multivariate data without fine-tuning. Physical Review Letters, 113:068102, 2014. (pages 52, 56, 70, 72, 74, 87, and 88)

O. Schwartz and E. P. Simoncelli. Natural signal statistics and sensory gain control. Nature Neuroscience, 4(8):819–825, 2001. (pages 135 and 140)

O. Schwartz, T. J. Sejnowski, and P. Dayan. Assignment of multiplicative mixtures in natural images. In Advances in Neural Information Processing Systems 17, pages 1217–1224, 2004. (page 139)

O. Schwartz, T. J. Sejnowski, and P. Dayan. Perceptual organization in the tilt illusion. Journal of Vision, 9:19–19, 2009. (page 140)

M. Seeger. Expectation propagation for exponential families. (EPFL-REPORT-161464), 2005. (page 90)

C. Shannon. A mathematical theory of communication. The Bell System Technical Journal, 27:379–423, 623–656, 1948. (page 12)

N. Shea, A. Boldt, D. Bang, N. Yeung, C. Heyes, and C. Frith. Supra-personal cognitive control and metacognition. Trends in Cognitive Sciences, 18(4), 2014. (page 23)

Z. Shun and P. McCullagh. Laplace approximation of high dimensional integrals. Journal of the Royal Statistical Society. Series B (methodological), pages 749–760, 1995. (pages 88 and 89)

W. Singer. Neuronal synchrony: a versatile code for the definition of relations? Neuron, 24(1):49–65, 1999. (page 152)

J. A. Sniezek and R. A. Henry. Accuracy and confidence in group judgment. Organizational Behavior and Human Decision Processes, 43(1), 1989. (page 45)

S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. PLoS Biology, 3(3):e68, 2005. (pages 109, 110, 111, 113, 133, and 159)

J. Steele. An Efron-Stein inequality for nonsymmetric statistics. Annals of Statistics, 14:753–758, 1986. (page 85)

K. E. Stephan, W. D. Penny, J. Daunizeau, R. J. Moran, and K. J. Friston. Bayesian model selection for group studies. Neuroimage, (4), 2009. (page 37)

E. A. Stern, A. E. Kincaid, and C. J. Wilson. Spontaneous subthreshold membrane potential fluctuations and action potential variability of rat corticostriatal and striatal neurons in vivo. Journal of Neurophysiology, 77(4):1697–1715, 1997. (page 159)

I. Stoianov and M. Zorzi. Emergence of a 'visual number sense' in hierarchical generative models. Nature Neuroscience, 15(2):194–196, 2012. (page 15)

T. C. Südhof. The presynaptic active zone. Neuron, 75:11–25, 2012. (page 107)

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems, pages 3104–3112, 2014. (page 14)

J. Sweller. Cognitive load during problem solving: effects on learning. Cognitive Science, (2), 1988. (pages 26 and 43)

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. arXiv:1312.6199, 2013. (page 14)

J. B. Tenenbaum, T. L. Griffiths, and C. Kemp. Theory-based Bayesian models of inductive learning and reasoning. Trends in Cognitive Sciences, 10(7):309–318, 2006. (page 135)

E. Tenney, R. MacCoun, B. Spellman, and R. Hastie. Calibration trumps confidence as a basis for witness credibility. Psychological Science, 18(1), 2007. (page 23)

G. Tkačik, T. Mora, O. Marre, D. Amodei, M. J. Berry II, and W. Bialek. Thermodynamics for a network of neurons: Signatures of criticality. arXiv:1407.5946, 1407.5946, 2014. (pages 52, 56, 66, 70, 74, and 75)

G. Tkačik, T. Mora, O. Marre, D. Amodei, S. E. Palmer, M. J. Berry, and W. Bialek. Thermodynamics and signatures of criticality in a network of neurons. Proceedings of the National Academy of Sciences, 112:11508–11513, 2015. (pages 52, 56, 66, 70, 72, 74, 75, and 85)

T. Toyoizumi, J.-P. Pfister, K. Aihara, and W. Gerstner. Spike-timing dependent plasticity and mutual information maximization for a spiking neuron model. In Advances in Neural Information Processing Systems, pages 1409–1416, 2004. (page 13)

S. J. Tripathy, J. Savitskaya, S. D. Burton, N. N. Urban, and R. C. Gerkin. Neuroelectro: a window to the world's neuron electrophysiology data. Frontiers in Neuroinformatics, 8, 2014. (page 158)

S. J. Tripathy, S. D. Burton, M. Geramita, R. C. Gerkin, and N. N. Urban. Brain-wide analysis of electrophysiological diversity yields novel categorization of mammalian neuron types. Journal of Neurophysiology, 113(10):3474–3489, 2015. (pages 94, 111, 113, and 157)

M. V. Tsodyks, W. E. Skaggs, T. J. Sejnowski, and B. L. McNaughton. Paradoxical effects of external modulation of inhibitory interneurons. The Journal of Neuroscience, 17:4382–4388, 1997. (page 142)

S. Turaga, L. Buesing, A. M. Packer, H. Dalgleish, N. Pettit, M. Hausser, and J. Macke. Inferring neural population dynamics from multiple partial recordings of the same neural circuit. In Advances in Neural Information Processing Systems, pages 539–547, 2013. (page 143)

S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. S. Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. Neural Computation, 22(2):511–538, 2010. (page 14)

G. G. Turrigiano and S. B. Nelson. Homeostatic plasticity in the developing nervous system. Nature Reviews Neuroscience, 5(2):97–107, 2004. (page 106)

J. Tyrcha, Y. Roudi, M. Marsili, and J. Hertz. The effect of nonstationarity on models inferred from neural data. Journal of Statistical Mechanics: Theory and Experiment, page 03005, 2013. (pages 51 and 68)

B. B. Ujfalussy, J. K. Makara, T. Branco, and M. Lengyel. Dendritic nonlinearities are tuned for efficient spike-based computations in cortical circuits. eLife, 4:e10056, 2015. (pages 142 and 152)

R. J. van Beers, A. C. Sittig, and J. J. D. van der Gon. Integration of proprioceptive and visual position-information: An experimentally supported model. Journal of Neurophysiology, 81(3):1355–1364, 1999a. (page 135)

R. J. van Beers, A. C. Sittig, and J. J. D. van Der Gon. Integration of proprioceptive and visual position-information: An experimentally supported model. Journal of Neurophysiology, 81(3):1355–1364, 1999b. (pages 15 and 16)

D. Vickers. Decision Processes in Visual Perception. London: Academic Press, 1979. (page 26)

J. T. Vogelstein, A. M. Packer, T. A. Machado, T. Sippy, B. Babadi, R. Yuste, and L. Paninski. Fast nonnegative deconvolution for spike train inference from

population calcium imaging. Journal of Neurophysiology, 104(6):3691–3704, 2010. (pages 105 and 119)

X. Wang, T. Lu, R. K. Snider, and L. Liang. Sustained firing in auditory cortex evoked by preferred stimuli. Nature, 435:341–346, 2005. (page 151)

X.-J. Wang. Decision making in recurrent neuronal circuits. Neuron, (2), 2008. (page 26)

A. B. Watson. A formula for human retinal ganglion cell receptive field density as a function of visual field location. Journal of Vision, 14:1–17, 2014. (page 158)

N. A. Weiss. A Course in Probability. Addison-Wesley, 2006. (pages 64 and 76)

B. Widrow and M. E. Hoff. Adaptive switching circuits. 1960. (pages 95 and 120)

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3-4):229–256, 1992. (page 120)

H. R. Wilson and J. D. Cowan. Excitatory and Inhibitory interactions in localized populations of model neurons. Biophysical Journal, 12(1):1–24, 1972. (page 146)

D. M. Wolpert. Probabilistic models in human sensorimotor control. Human Movement Science, 26(4):511–524, 2007. (page 15)

D. M. Wolpert and M. S. Landy. Motor control is decision-making. Current Opinion in Neurobiology, 22(6):996–1003, 2012. (page 15)

D. M. Wolpert, Z. Ghahramani, and M. I. Jordan. An internal model for sensorimotor integration. Science, 269(5232):1880–1882, 1995. (pages 15, 16, and 135)

T. Womelsdorf, J.-M. Schoffelen, R. Oostenveld, W. Singer, R. Desimone, A. K. Engel, and P. Fries. Modulation of neuronal interactions through neuronal synchronization. Science, 316(5831):1609–1612, 2007. (page 152)

D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the National Academy of Sciences, 111 (23):8619–8624, 2014. (page 14)

R. S. Zemel, P. Dayan, and A. Pouget. Probabilistic interpretation of population codes. Neural Computation, 10(2):403–430, 1998. (pages 18 and 135)

Y.-Y. Zhang, Y. Li, H.-Q. Gong, and P.-J. Liang. Temporal and spatial properties of the retinal ganglion cells' response to natural stimuli described by treves-rolls sparsity. In 2009 3rd International Conference on Bioinformatics and Biomedical Engineering, pages 1–4, 2009. (page 157)

L. Ziegler, F. Zenke, D. B. Kastner, and W. Gerstner. Synaptic consolidation: from synapses to behavioral modeling. Journal of Neuroscience, 35(3):1319–1334, 2015. (page 106)

G. K. Zipf. Selected studies of the principle of relative frequency in language. Harvard Univ. Press, 1932. (pages 51 and 59)

G. K. Zipf. Human behavior and the principle of least effort. Addison-Wesley, 1949. (pages 52 and 59)

A. Zylberberg, P. Roelfsema, and M. Sigman. Variance misperception explains illusions of confidence in simple perceptual decisions. Consciousness & Cognition, 27(1), 2014. (page 46)