Normative theories of synaptic plasticity for representation and perceptual discrimination

by

Colin Bredenberg

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Center for Neural Science

New York University

September, 2022

Professor Cristina Savin

Professor Eero P. Simoncelli

To my parents, whose love made this possible.

Acknowledgements

I would like to thank everyone who contributed to this work, and everyone who has supported me in producing it. Thank you Eero, and thank you Cristina for giving me the best working environment I could have ever hoped for, with the freedom to work on projects that I enjoy and the support and training that I needed to push them through to completion. Thank you also to my collaborators, Katie Martin, Jordan Lei, Ben Lyo, Rob Froemke, and Roozbeh Kiani, without whom this work could not have been completed. I would also like to thank the members of my committee—Roozbeh and Rob, again—as well as Tony Movshon, who stepped up to be my committee chair, and Brent Doiron.

I would like to thank the people who have shaped the way that I think about computational neuroscience, especially the members of my labs and first year class. In particular, I would like to thank Owen Marschall, Caroline Haimerl, Pierre-Étienne Fiquet, Roman Huszar, Pedro Herrero-Vidal, Lyndon Duong, and Nikhil Parthasarathy, whose conversations and friendship have been a real gift. I would also like to thank the people who have given me continual mental and emotional support, especially my mom, brother, and friends from back home, who all kept me sane during the pandemic. And of course, I would like to thank Jenn, who has gifted me with an incredible life and love all these years. My time in this program has been wonderful, thank you to everyone who contributed to this experience.

Contents

List of Figures

List of Tables

List of Appendices

1 Introduction

Beyond the flow of experience constantly shifting before us, we feel our identities themselves changing with time. We remember, we associate, we learn. The dynamics of the mind are fundamental to human experience, but as yet, we only have the slightest idea how changes in our minds correspond to changes in our brains. Investigating the adaptive brain promises deep philosophical insight, as well as pragmatic affordances both for therapeutic intervention and for building machines with learning capabilities as impressive as the brain's.

To understand how dynamical biological processes link to perceptual or behavioral changes throughout time, we have to start with what lasting changes occur within the brain. Of the many features of neural architecture that modify over time, from the biophysical properties of individual neurons to the development or pruning of synapses between neurons, changes in the strength of synapses themselves have long been among the most prominent candidates for the neural substrate of longitudinal perceptual and behavioral change, because many synaptic connections are easily modified, and modifications persist for extended periods of time [Bliss and Collingridge 1993]. Further, synaptic modification has been associated with many of the brain's critical adaptive functions, including: memory [Martin et al. 2000], experience-based sensory development [Levelt and Hübener 2012], operant conditioning [Ohl and Scheich 2005; Fritz et al. 2003], and compensation for stroke [Murphy and Corbett 2009] or neurodegeneration [Zigmond et al. 1990].

The goal of this thesis is to make progress on our collective understanding of learning and

adaptation in the brain. Its primary focus will be on what are called *normative synaptic plasticity theories*, which establish mathematical and simulation-based links between experimentally observed synaptic plasticity phenomena and the adaptive functions critical for behavior and development that they support. In Chapter 2, we will introduce and define this class of theory from the ground up. We will also critically review previous literature dedicated to developing and testing normative plasticity theories, and produce a set of guidelines that future modeling efforts should attempt to adhere to, in order to facilitate the testing of these theories; in many ways, these principles both influenced and were inspired by the work in later chapters. We will also provide detailed tutorials on two canonical normative plasticity theories—REINFORCE (Appendix A.3; [Williams 1992]) and the Wake-Sleep algorithm (Appendix A.4; [Hinton et al. 1995])—after which we will show in detail how these algorithms measure up to the standards that we have set.

In models of neural systems, the REINFORCE algorithm adapts synapses through what is called a reward-modulated Hebbian plasticity rule, where reward information is multiplicatively combined with pre- and postsynaptic firing rate information to produce synaptic weight updates. In Chapter 3, we show how a normative plasticity rule closely related to REINFORCE can produce sensory representations that compensate for noise and are efficient, in that they selectively represent task-relevant information without wasting metabolic resources. In Chapter 4, we observe that our algorithm has many similarities to perceptual learning in the mouse auditory cortex: we adapt it to demonstrate how reward and context information delivered by acetylcholine signals from the nucleus basalis could underlie both context-specific adaptation in auditory cortex and reward-based perceptual learning. We then compare our model to longitudinal two-photon calcium recordings of mice learning to perform a two-alternative forced choice task and demonstrate that our model is able to capture many features of the data, including learning speeds, behavioral responses, neural representations, and context-specific responses.

Chapters 3 and 4 focus on reward-based learning, but at many stages in development animals do not have access to explicit reward signals to shape their representations. This suggests that

there may be other *unsupervised* mechanisms at play to assist in sensory development, but the neural substrates of these mechanisms are poorly understood. In Chapter 5 we develop a theory called 'impression learning', which proposes a mechanism for learning sensory representations by adapting synapses to minimize a prediction error between predictive signals arriving at apical dendrites of pyramidal neurons and incoming sensory information at basal dendrites. This theory generalizes the Wake-Sleep algorithm, and improves on previous prediction-error based theories of learning by demonstrating how learning can occur continuously with sensory perception, rather than requiring an offline learning phase.

In Chapter 6, we close off the thesis with a theoretical examination of the difficulties associated with studying complex, adaptive systems experimentally. In particular, we study causal interventions, in which a subset of neurons in a circuit are either lesioned or transiently inactivated. We show how inactivated neurons can easily be involved in a task even when their inactivation does not show experimentally observable behavioral effects, especially in circuits with many nodes capable of performing a task, or in circuits that are actively learning through synaptic plasticity.

Our results across the chapters of this thesis collectively demonstrate the importance of normative theories of plasticity, both for conceptualizing learning in the brain and informing experiments that investigate adaptive neural circuits.

2 An overview of normative synaptic plasticity modeling

When asking what constitutes sufficient understanding of synaptic plasticity, we must ask what we want that understanding to grant us. Most important for any pragmatic application is a precise link between plasticity and adaptive behaviors of interest—one which is currently largely lacking. In what follows, we will distinguish a 'normative' modeling approach from other alternatives, demonstrate why it shows promise for establishing this link, and outline a set of desiderata which normative plasticity models should attempt to adhere to in order to strengthen the link between plasticity and the adaptive phenomena it underlies. Then, to provide concrete examples of these principles in action, in Appendices A.3 and A.4 we provide worked tutorials on two canonical normative plasticity models, REINFORCE [Williams 1992] and the Wake-Sleep algorithm [Dayan et al. 1995; Hinton et al. 1995] respectively, and illustrate their successes and failures to match our desiderata.

2.1 Phenomenological, mechanistic, and normative plasticity models

When discussing models of synaptic plasticity, it will be useful to make the distinction between three partially overlapping types of model: phenomenological, mechanistic, and normative (Fig.

2.1a) [Levenstein et al. 2020]. The focus of this chapter will be on normative plasticity models, but to understand their importance, we have to view them in relation to their counterparts.

In the simplest terms, a phenomenological model's focus is on describing experimental data: the primary goal is to concisely summarize relationships between observed variables. As an example, many early studies of spike-timing-dependent plasticity (STDP) described the relationship between plasticity and the relative timing of pre- and post-synaptic spikes with exponential curves fit to data [Zhang et al. 1998; Dan and Poo 2004; Sjöström et al. 2010]. Such models can reduce the complexity of data, providing interpretability and predictive power. They are incomplete descriptions of the biophysical processes that form the causal link between spike times and plasticity, but extract important features of the data.

A mechanistic model builds on the phenomenological project by attempting to describe experimental data in terms of causal interactions between biophysical quantities. For instance, since the initial characterization of STDP, a plethora of studies have emerged characterizing in detail the interactions between backpropagating action potentials [Magee and Johnston 1997], dendritic morphological properties [Froemke et al. 2005; Letzkus et al. 2006; Sjöström and Häusser 2006], local membrane voltage, NMDA ion channel properties, and calcium-sensitive molecules near the synapse; mechanistic models [Graupner and Brunel 2010] characterize how these variables all collectively contribute to the strengthening or weakening of the synapse. As a consequence of their depth and breadth, mechanistic models can often provide predictions that are radically outside of the scope of the original experiment, and provide useful targets for experimental manipulation.

The distinction between phenomenological models and mechanistic models is not always so clear, especially in areas where our scientific understanding is progressing rapidly. In nascent mechanistic models, there often exist 'black boxes' that specify interactions between known biophysical quantities, without any clear understanding of how these interactions come about [Craver 2007]. Furthermore, the status of 'biophysical' does not make a quantity or its interactions

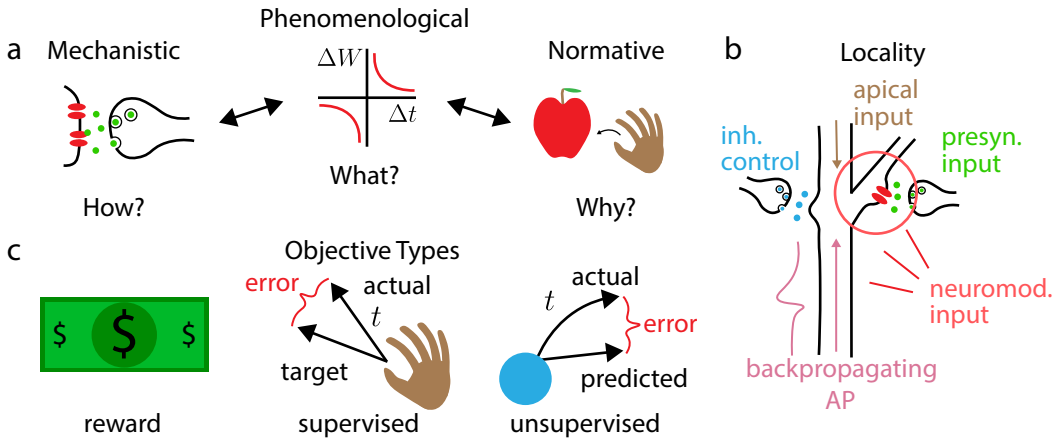

Figure 2.1: Defining normative modeling. a. Spectrum of synaptic plasticity models. Mechanistic models show how detailed biophysical interactions produce observed plasticity, phenomenological models concisely describe what changes in experimental variables (e.g. post-pre relative spike timing Δt) affect plasticity (ΔW), and normative models explain why the observed plasticity implements capabilities that are useful to the organism. **b.** Schematic illustrating the range of local variables that may be available for synaptic plasticity. These include, but are not limited to: backpropagating action potentials from the soma, apical dendritic input, pre- and postsynaptic activity, neuromodulatory signals, and potentially inhibitory input from local microcircuitry. **c.** Classes of objective function used in normative plasticity theories. Reward-based objectives involve only feedback about how well the organism or network performed, whereas supervised objectives provide explicit targets for network output. By contrast, unsupervised objectives do not require any form of explicit feedback to train the network.

any more real than the variables and relationships articulated by a phenomenological model: biophysical quantities are simply more entrenched and better understood in relation to a breadth of experimental studies of neural microbiology. In this way, we can see that there exists a spectrum between phenomenological and mechanistic models, and that oftentimes, mechanistic models grow from phenomenological ones. However, there is more to the spectrum: while phenomenological and mechanistic models articulate how synaptic plasticity works, they do not explain *why* it exists in the brain, i.e. what its importance is for neural circuits, behavior, or perception. An appeal to normative modeling is required to provide this explanation precisely.

Normative models aim to answer this 'why' question by connecting plasticity to observed network-level or behavioral-level phenomena, including memory formation [Hopfield 1982] and consolidation [Benna and Fusi 2016; Clopath et al. 2008; Fusi et al. 2005], reinforcement

learning [Frémaux and Gerstner 2016], and representation learning [Hinton et al. 1995; Rao and Ballard 1999]. This class of plasticity model, in our view, employs a fundamentally different set of methodologies from phenomenological or mechanistic models, in order to provide the missing link between plasticity and function. Guided by the intuition that plasticity processes have been optimized on an evolutionary timescale to near-optimally perform adaptive functions, normative plasticity theories are typically 'top-down', in that they begin with a set of prescriptions about how synapses 'should' modify in order to optimally perform a given learning-based function. Subsequently, with varying degrees of success, these theories attempt to show that real biology matches or approximates this optimal solution. This process is ongoing, and though experimental support for such forms of plasticity are growing, much work remains to be done. In this chapter, we will critically review existing normative plasticity approaches and discuss how they could be built upon.

2.2 Desiderata for normative models

One of the biggest challenges for a normative model of synaptic plasticity is its connection to biology: constructing an artificial neural network with simulated synapses (synaptic weight parameters) that adapt to improve performance on any of a variety of functions from sensory processing [LeCun et al. 1989a; Krizhevsky et al. 2012], to motor learning [Heess et al. 2017], to abstract game learning [Silver et al. 2017; Vinyals et al. 2019] is relatively straightforward compared to experimentally testing whether the mechanisms used by a given artificial network correspond to the mechanisms used by the brain. Compared to the simulations and mathematical analysis used to explore machine learning algorithms, neuroscience experiments are time-consuming and expensive: it is not possible to test every imaginable theoretical learning mechanism with an experiment, and many such mechanisms are so abstract that it is not even clear what to test. Further, compared to network simulations which provide total access to neural activations, stimuli,

and synaptic parameters over the whole course of learning, any one neuroscience experiment can only reveal a very small amount about what is going on in a circuit.

In what follows, we will articulate a set of desiderata that can serve as both intermediate objectives for the development of normative models of synaptic plasticity, and as intermediate criteria which can be used to invalidate (or at least distrust) such models in the absence of explicit experimental rejection. The following desiderata are not necessary (and are certainly not sufficient) conditions for a normative model to be validated: many of the desiderata could under some conditions be absent from a convincing normative model. We will only argue that each principle is desirable, for some combination of the following reasons: first, it may help ensure that the plasticity model actually qualifies as normative; second, it requires a model to accommodate known facts about biology; third, it helps ensure that models can be compared properly to existing experimental literature and generate genuinely testable experimental predictions. Most of these desiderata are relatively intuitive and simple. However, it has proven incredibly difficult for existing models, across any given normative goal from sensory processing, to memory, to reinforcement learning, to satisfy all desiderata in tandem.

2.2.1 Locality

Biological synapses can only change strengths using biochemical signals available at the synapse itself. 'Locality' refers to the idea that a postulated synaptic plasticity mechanism should only refer to variables that could be conceivably available at a given synapse. Though locality may seem like an obvious requirement for any theory of biological function, for synaptic plasticity it presents a great mystery: how does a system as a whole, whose success or failure is determined by the joint action of many neurons distributed across the entire brain, communicate information to individual synapses about how to improve? Resolving this mystery is highly nontrivial, as illustrated by the demonstrable *nonlocality* of most successful machine learning algorithms used to train artificial neural networks to perform tasks, including backpropagation [Werbos 1974;

Rumelhart et al. 1985] (See Appendix A.1), backpropagation through time [Werbos 1990], and real-time recurrent learning [Williams and Zipser 1989].

Despite its importance as a guiding principle for normative theories of synaptic plasticity, locality is a slippery concept, primarily because of the neuroscience community's insufficient understanding of the precise battery of biochemical signals available to a synapse, and how those signals could be used to approximate quantities normally used in theory. As a simple example, many normative theories require information about the pre- and postsynaptic firing rates of a neuron, similar to Hebb's Postulate [Hebb 1949]. However, neurons predominately communicate to one another through discrete action potentials, and additional cellular machinery would be required to form an estimate of a firing rates pre- and postsynaptically based on backpropagating action potentials from the soma and on postsynaptic potentials. Whether a plasticity rule derived from normative principles involves rate or spike-based information is often a function of the neuron model used in the theory, and it is often difficult to formulate predictions about how a realistic, non-idealized neuron should exactly modify its synapses based on over-simplified models. Therefore, often normative theories declare success when some standard of plausibility is reached, where derived plasticity rules roughly match the experimental literature [Payeur et al. 2021] or only require reasonably simple functions of postsynaptic and pre-synaptic activity that a synapse could hypothetically approximate [Oja 1982; Scellier and Bengio 2017; Williams 1992].

In normative models of synaptic plasticity, the requirement of locality is in perpetual tension with the general requirement for some form of 'credit assignment' [Lillicrap et al. 2020; Richards et al. 2019], i.e. a mechanism capable of signaling to a neuron that it is 'responsible' for a network-wide error, and should modify its synapses to reduce errors. Depending on a network's objective, a system's credit assignment mechanism *could* take a wide variety of forms, some small number of which may only require information about the pre- and post-synaptic activity of a cell [Oja 1982; Pehlevan et al. 2015, 2017; Obeid et al. 2019; Brendel et al. 2020], but many of which appear to require the existence of some form of error [Scellier and Bengio 2017; Lillicrap et al. 2016; Akrout

et al. 2019] or reward-based [Williams 1992; Fiete et al. 2007; Legenstein et al. 2010] signal.

The extent to which a credit assignment signal postulated by a normative theory meets the standards of 'locality' depends heavily on the nature of the signal. For instance, there is growing support for the idea that neuromodulatory systems, including dopamine [Otani et al. 2003; Calabresi et al. 2007; Reynolds and Wickens 2002], norepinephrine [Martins and Froemke 2015], oxytocin [Marlin et al. 2015], and acetylcholine [Froemke et al. 2013; Guo et al. 2019; Hangya et al. 2015; Rasmusson 2000; Shinoe et al. 2005] can distribute information about reward [Guo et al. 2019], expectation of reward [Schultz et al. 1997], and salience [Hangya et al. 2015] diffusely throughout the brain to induce or modify synaptic plasticity in their targeted circuits. Therefore, in many cases it is reasonable for normative theories to postulate that synapses have access to global reward or reward-like signals, without violating the requirement that plasticity be affected only by locally-available information [Frémaux and Gerstner 2016]. In other normative theories, credit assignment can occur through more advanced processes which have less experimental support. If, as for the algorithms used to train deep image classifiers [Yamins et al. 2014; Yamins and DiCarlo 2016], the objective is formulated in terms of success or failure to match supervised labels (e.g. the system incorrectly classifies an image as 'cow' when viewing a goat), then the onus rests on a normative theory to account for where this signal comes from and how it is calculated. Though supervised error signals do seem to modify synaptic plasticity in some neural systems (for example the cerebellum [Gao et al. 2012; Bouvier et al. 2018]), in most cases supervised learning presupposes the existence of ground truth information readily available in some neural region; providing a normative theory of synaptic plasticity without providing an account of how this 'supervisor' comes to be amounts to passing a large amount of the work of credit assignment onto an ethereal, unknown and unstudied system. The inclusion of details in a normative theory that violate locality as understood by current experimental data can serve as an excellent way to test that theory experimentally; however, if those model components are not sufficiently specific, by for example not specifying which brain area a particular supervisory signal should come

from and how it could be calculated, then the theory becomes effectively under-constrained and unverifiable.

Locality as a desideratum serves as a heuristic stand-in for the requirement that a normative model must be eventually held to the standard of experimental evidence. The quality of theoretical research is only partly determined by its correspondence to what is already known: to allow theoretical research to progress neuroscience as a field, it is vital for theories to be able to generate predictions and to motivate subsequent research. In some cases, this may necessitate a theory postulating credit assignment mechanisms that have not yet been observed in experiments. However, for such an exercise to be constructive, the theory should clearly articulate how it deviates from the current state of the experimental field, and how these deviations can be tested (Section 2.2.7). Furthermore, the process of mathematical abstraction necessitates approximation [Cartwright and McMullin 1984]: requiring a normative theory to adhere to 'locality' without necessarily requiring a perfect correspondence to experimental data allows normative theories to strive to capture the essence of synaptic learning processes without becoming mired in technical details.

2.2.2 Improving performance

One way of viewing the normative project is that it attempts to organize the diversity of synaptic dynamics existing within a neural system into the simplest explanatory framework possible for what functions the system's plasticity subserves. Usually, this framework is mathematical for pragmatic reasons: mathematics provides the precision and power necessary to establish clear relationships between plasticity and function. In particular, viewing neural plasticity as an approximate optimization process has been fruitful [Lillicrap et al. 2020; Richards et al. 2019], wherein synaptic modifications progressively reduce a scalar loss function. This process can be divided into two steps: articulating an appropriate objective, and subsequently demonstrating that a synaptic plasticity mechanism improves performance on that objective.

It can be extremely difficult to reduce the full range of functions a given circuit must perform to a scalar objective function, but as we will show subsequently, the conceptual benefits can be immense. On one side, picking too simple an objective function runs the risk of ignoring many functions a system is required to perform. For instance, early normative theories of learning in sensory systems show how synaptic plasticity could minimize the objective function underlying principal component analysis (PCA) [Oja 1982], but merely representing the principal components of an incoming sensory stream is an inadequate characterization of sensory processing for several reasons: PCA neglects the temporality of naturalistic inputs and cannot capture important phenomena exhibited by cortical neurons, including complex gain control capabilities [Simoncelli and Heeger 1998] and texture [Ziemba et al. 2016] and object class [Rust and DiCarlo 2010] selective responses. A given synaptic plasticity mechanism may only be able to minimize a restricted subset of objectives, and for a normative theory, the set of possible objectives that can be minimized must encompass a wide range of functions that the brain is known to subserve. Beyond principal component analysis, a more modern and reasonable class of objectives for unsupervised representation learning in sensory systems would be for training general hierarchical generative models (e.g. the evidence lower bound (ELBO) which underlies a wide variety of unsupervised algorithms and architectures, including PCA, factor analysis, Kalman filtering [Roweis and Ghahramani 1999], the Helmholtz machine [Dayan et al. 1995], predictive coding [Rao and Ballard 1999], and variational autoencoders [Rezende et al. 2014; Kingma and Welling 2014]). On the other side, selecting too complicated an objective function can undermine the normative process entirely. For example, if we were to postulate that the 'goal' of a neural system is to do exactly what experimentalists observe it doing at both the spiking and synaptic level, i.e. everything in a neural system happens precisely as was 'intended', then the normative project becomes vacuous: the model provides no conceptual simplification beyond what was observed experimentally, and the community does not learn anything new.

Normative theories of synaptic plasticity developed to date usually involve some combination

of supervised, unsupervised , or reinforcement learning objectives (Fig. 2.1c). The choice of objective function for a neural system is laden with philosophical assumptions about the system's purpose, and can exert a huge influence on the resultant form of the synaptic plasticity. For instance, supervised learning usually involves the existence of either an internal or external teacher. If the teacher is external, such a learning mechanism could only be leveraged under the very specific and comparatively rare conditions in which the organism is being overtly taught. If the teacher is internal, as already mentioned, to be satisfactory the normative theory must provide an account for how the internal teacher gains access to its knowledge. Ground truth information is hard to come by for neural systems, and so it may be better to adapt existing normative theories that rely on supervisory information [Ackley et al. 1985; Xie and Seung 2003; Scellier and Bengio 2017] to unsupervised or reinforcement learning objectives. Generative modeling is a form of unsupervised learning that postulates that a sensory system is actively building a probabilistic model of its sensory inputs, which can be used to simulate possible future outcomes and perform Bayesian reasoning [Fiser et al. 2010a]. This vision of sensory coding is popular both for its ability to accomodate normative plasticity theories [Rao and Ballard 1999; Dayan et al. 1995; Kappel et al. 2014; Bredenberg et al. 2021] and for its philosophical vision of sensory processing as a form of advanced model building, beyond simple sensory transformations. However, model construction is only indirectly useful for many tasks involving rewards and planning, and so such plasticity would have to occur concomitantly with reward-based [Frémaux and Gerstner 2016] or motor [Gao et al. 2012; Feulner and Clopath 2021] learning. Furthermore, alternative perspectives on sensory processing exist, including those based on maximizing the information about a sensory stimulus contained in a neural population [Attneave 1954; Atick and Redlich 1990] subject to metabolic efficiency constraints [Tishby et al. 2000; Simoncelli and Olshausen 2001], and those based on 'contrastive methods' [Oord et al. 2018; Illing et al. 2021], where some form of self-supervising internal teacher encourages the neural representation of some stimuli to grow closer together, while encouraging others to grow more discriminable.

Evaluating which objective function (or functions) best explains the properties of a neural system is very hard: while some forms of objective function may have discriminable effects on plasticity (e.g. supervised vs. unsupervised learning [Nayebi et al. 2020]), others are even provably impossible to distinguish. As a simple example, suppose that we have an N^r dimensional single-layer neural network receiving N^s dimensional stimuli through an $N^r \times N^s$ dimensional weight matrix $\mathbf{W}$. We have the response given by:

$$\mathbf{r} = f(\mathbf{W}\mathbf{s}), \tag{2.1}$$

where $f(\cdot)$ is a pointwise tanh nonlinearity. Now suppose that some setting of synaptic weights $\mathbf{W}^*$ observed in an experiment minimizes an objective function $\mathcal{L}$, i.e. $\mathcal{L}(\mathbf{W}^*) < \mathcal{L}(\mathbf{W})\ \forall \mathbf{W}$. We might be tempted to argue that because $\mathbf{W}^*$ minimizes $\mathcal{L}$, $\mathcal{L}$ must be the objective that the system is minimizing. However, there are an infinite variety of alternative objectives that share this same minimum (Appendix A.2). This motivates the idea that for a given dataset, it is possible that one objective ($\tilde{\mathcal{L}}$) can *masquerade* as another ($\mathcal{L}$). In some cases, complex objective functions can masquerade as simple objectives, which may only be epiphenomenal. For instance, it has been hypothesized that synaptic modifications may preserve the balance between inhibitory and excitatory inputs to a cell [Vogels et al. 2011]; recent theories have proposed that this E/I balance may only be a consequence of a more advanced theory of sensory predictive coding [Brendel et al. 2020]. In other cases, seemingly distinct frameworks, such as generative modeling, information maximization, or denoising may simply produce similar synaptic plasticity modifications because the frameworks often overlap heavily [Vincent et al. 2010], and may not be distinguishable on simple datasets without targeted experimental attempts to disambiguate between the two perspectives.

Furthermore, not every function performed by biological systems has been adequately incorporated into a simple optimization framework. For example, though the Hebbian plasticity proposed

by the Hopfield network model endows model circuits with associative memory, the utility of learning is characterized by the dynamical attractor structure it embeds in the neural circuit, rather than by its direct minimization of an objective function [Hopfield 1982]. In addition, the notion that some parts of the brain may have synaptic plasticity mechanisms for representation learning while other parts have plasticity for reinforcement learning suggests that the brain may be better viewed as a collection of interacting systems with only partially overlapping goals. This multiagent [Zhang et al. 2021] formulation of learning has intuitive appeal, because it can decompose broad objectives like survival into a series of intermediate objectives carried out by individual systems. Such a formulation could help explain how locality emerges, i.e. why synapses do not need information about distant neural circuits in order to improve performance. However, with this additional appeal comes additional conceptual and mathematical complexity, because improving performance on one objective could very easily harm the performance of other systems. Therefore, insofar as a collection of neural circuits and plasticity mechanisms *can* be viewed as acting in concert to improve a unified objective, simple optimization is the preferable perspective.

Having addressed many difficulties associated with choosing a good objective function, we now move to difficulties involved in demonstrating that a particular synaptic plasticity rule decreases a chosen objective[1]. How could such a property be proven? For a particular plasticity rule to reduce an objective, we need to show that the following principle holds:

$$\mathcal{L}(\mathbf{W} + \Delta\mathbf{W}) < \mathcal{L}(\mathbf{W}) \tag{2.2}$$

$$\Rightarrow \mathcal{L}(\mathbf{W} + \Delta\mathbf{W}) - \mathcal{L}(\mathbf{W}) < 0, \tag{2.3}$$

for some update $\Delta\mathbf{W}$ determined by the plasticity rule. If we accept the additional supposition that $\Delta\mathbf{W}$ is very small, we can employ the first order Taylor approximation (treating $\mathbf{W}$ as a flattened

[1]Some objectives (like reward functions) are best thought of as being maximized rather than minimized. Without loss of generality, in such cases we can minimize the negative reward function.

vector of length $N^r \times N^s$): $\mathcal{L}(\mathbf{W} + \Delta\mathbf{W}) \approx \mathcal{L}(\mathbf{W}) + \frac{d\mathcal{L}}{d\mathbf{W}}(\mathbf{W})^T \Delta\mathbf{W}$. Substituting this approximation into our reduction criterion, we have after cancellation:

$$\frac{d\mathcal{L}}{d\mathbf{W}}(\mathbf{W})^T \Delta\mathbf{W} < 0. \tag{2.4}$$

This shows that for small weight updates (slow learning rates), the inner product between a synaptic learning rule $\Delta\mathbf{W}$ and the gradient of the selected loss function $\mathcal{L}(\mathbf{W})$ with respect to the weight change must be negative. The simplest way to ensure that this is true is for $\Delta\mathbf{W}$ to equal a small scalar λ times the negative gradient of the loss ($-\lambda \frac{d\mathcal{L}}{d\mathbf{W}}(\mathbf{W})^T \frac{d\mathcal{L}}{d\mathbf{W}}(\mathbf{W}) = -\lambda \| \frac{d\mathcal{L}}{d\mathbf{W}}(\mathbf{W}) \|_2^2 < 0$). If this were true, plasticity would be guaranteed to improve performance on the objective $\mathcal{L}$. Unfortunately, for even the simplest neural networks and objective functions, naive methods of calculating this gradient will prove to be nonlocal (see Appendix A.1 for a simple example). Thus, the critical challenge for normative theories of synaptic plasticity is finding ways that neural networks can find synaptic modifications $\Delta\mathbf{W}$ that demonstrably have a negative inner product with the gradient of a desired objective $\mathcal{L}$, while still allowing the neural network to satisfy biologically realistic locality constraints. However, it is important to note that if an update $\Delta\mathbf{W}$ reduces any one objective function, there are again an infinite number of similar alternative objective functions that this update also reduces (Appendix A.2); therefore it is perhaps best to think of normative plasticity models in terms of the family of objectives or functions that they minimize—committing to any one particular objective within that family reflects the predilections of the theorist, not the system.

Different normative studies demonstrate that Eq. 2.4 holds by different methods. Some studies show empirically across many simulations that this inner product is negative [Lillicrap et al. 2016; Marschall et al. 2020]. However, this demonstration does not answer the following questions: how would we know that the network would still perform well if a different task were chosen, or if the network's architecture were different, or if various elements of the simulated plasticity

mechanism were changed? A simulation may speak for itself, but it has relatively limited power to extrapolate beyond its immediate results, especially when the neuron models used in large-scale network simulations are often very reductive [Gerstner and Kistler 2002] and when small changes in simulated network parameters can effect large qualitative differences in network behavior [Xiao et al. 2021]. One could counter by providing a battery of *in silico* simulations under a variety of different parameter settings and circumstances, but not only would such an attempt rapidly suffer the curse of dimensionality, it would be so extensive a collection of data that it would be only slightly more useful than a collection of *in vivo* or *in vitro* experiments, while also suffering from a much more dubious connection to natural biological function. As such, simulation-based justifications suffer from a lack of conciseness and an inability to easily address counterfactuals.

Other studies take the more radical approach of developing synaptic plasticity mechanisms through repeated evolutionary optimization simulations [Jordan et al. 2021], which show how approximately optimal plasticity could emerge from essentially survival-of-the-fittest principles. While this amounts to an explanation of how plasticity could come to be, which is an interesting question in itself—it is *not* a good explanation of why a given plasticity rule reduces an objective. This historical approach is dissatisfactory for explaining the connection between neural adaptation and behavior because it is subject to infinite regress: if an appeal to development or evolution were an appropriate explanation for an organism's current capabilities, we would rapidly find ourselves appealing to the origins of life or even the universe to explain the current value of synaptic plasticity for the brain. We would not, for instance, consider a detailed description of why a bird's wing improves incrementally over its ancestors' to be a satisfactory explanation of how the wing enables flight[2]. Furthermore, as with simulation-based approaches, historical justifications can neither provide precise prescriptions to experimentalists for how they can perturb network function, nor abstract away unnecessary details so that plasticity mechanisms can safely and

[2]In much the same way, normative theories of synaptic plasticity explain how neural systems adapt through time to perform better, but they do not adequately explain what features or response properties of the perfected system are desirable.

reliably be incorporated into artificial learning systems.

The most desirable, most precise (and most difficult to construct) explanation is mathematical. Mathematical theories seek to characterize the relationship between a local synaptic plasticity rule and the gradient of the chosen objective function. Some plasticity rules amount to stochastic approximations to the true gradient [Williams 1992; Scellier and Bengio 2017] and some are systematically biased but maintain a negative inner product under reasonable assumptions [Bredenberg et al. 2021; Dayan et al. 1995; Amari and Nakahara 1999; Meulemans et al. 2020]. As we will discuss below, the range of conditions under which a satisfactory proof of loss reduction can be found is a mark of the power and importance of a normative theory, and the degree of detail with which a clean mathematical relationship can be established between a plasticity rule and the gradient of a loss is highly variable across theories. Mathematical analysis allows one to know quite clearly when a particular plasticity rule will decrease a loss function, and identifies how plasticity mechanisms should change with changes in the network architecture or environment. However, analysis is often only possible under restrictive circumstances, and it is often necessary to supplement mathematical results with empirical simulations in order to demonstrate that the results extend to more general, more realistic circumstances.

2.2.3 Architectural flexibility

The learning algorithm implemented by a plasticity model often requires specific architectural motifs to exist in a neural circuit in order to deliver reward, error, or prediction signals. These might include diffuse neuromodulatory projections (Fig. A.1b) or neuron-specific top-down synapses onto apical dendrites (Fig. A.2c). Such architectural features (or alternative, isomorphic motifs) are *required* for the learning algorithm in question, and are known to exist in a wide range of cortical areas. However, not all architectural motifs that exist in normative plasticity models are so ubiquitous in the brain. There is huge diversity in cell types and dynamical properties of neurons across animals and cortical (and subcortical) areas. If a normative model of synaptic

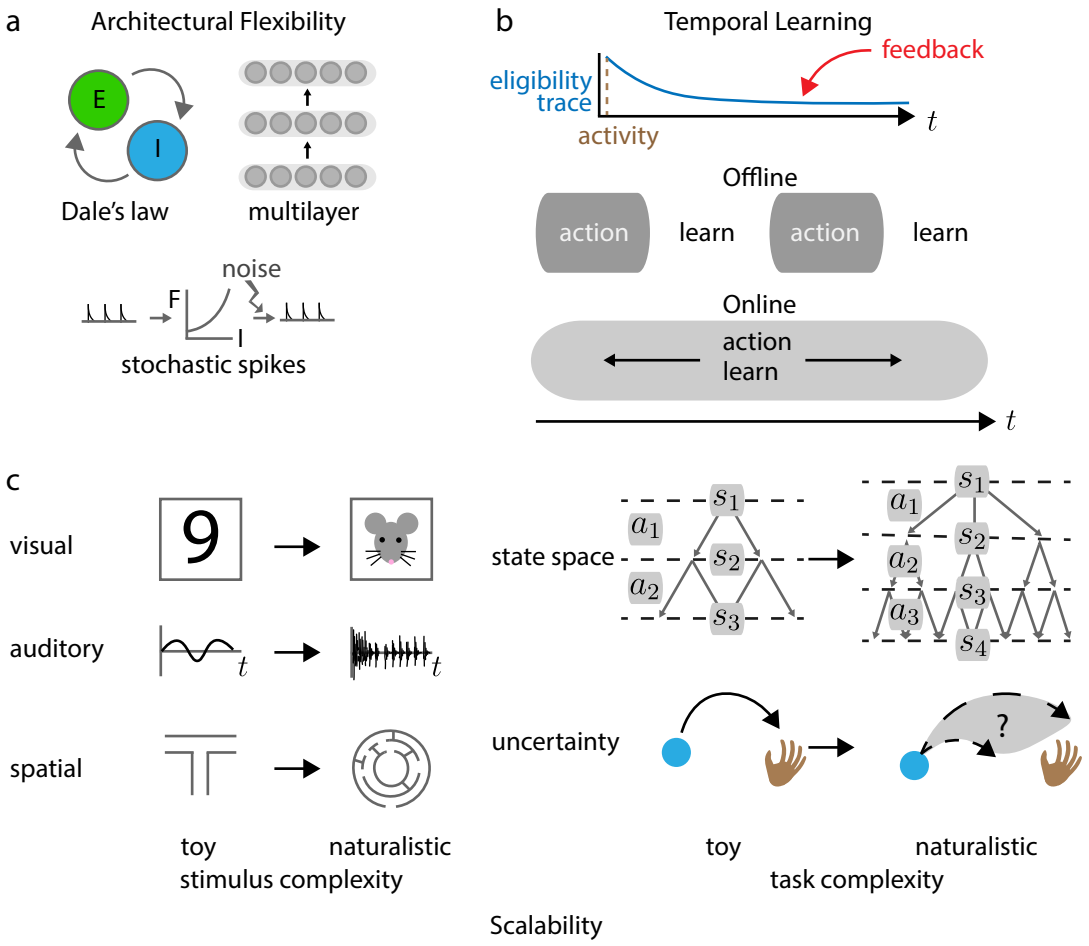

Figure 2.2: Architecture and scalability considerations for normative plasticity models. a. Features of realistic biological networks that normative plasticity theories should be able to account for: separation of excitatory and inhibitory neuron populations; stochastic and spiking input-output functions for individual neurons; and multilayer, recurrent connectivity. **b.** For actions in the past to be associated with delayed supervisory or reinforcement signals, plasticity algorithms require a mechanism of temporal association. One candidate is the 'eligibility trace,' which stores information about coactivity throughout time locally to a synapse, and subsequently modifies synaptic connections when paired with feedback information. Learning can occur offline, where some or all synaptic modification occurs in the absence of action or perception by the organism. Alternatively, it can occur online, where the organism acts and learns simultaneously. **c.** Stimuli (left) and task structure (right) can become complex in many ways. Different sensory features (e.g. visual, auditory, or spatial information) can all be made more naturalistic by training networks on stimuli organisms are exposed to and learn from in natural environments. Further, tasks can be made more naturalistic by increasing the number of action options (a) and sequential state (s) transitions required for a network to achieve its goals and by adding uncertainty into the task.

plasticity is overly dependent on specific features of a neuron model or architecture being used, then the postulated form of learning is considerably less likely to be tolerant to variations in

biophysical properties and microcircuitry both within and across brain areas. From this we distill the following principle: in the absence of explicit empirical evidence supporting specific architectural choices, normative theories of synaptic plasticity should be general, rather than restrictive, in the architectural motifs that they support. In what follows, we will highlight several particularly important motifs that normative models must accommodate.

Contrary to the highly reduced deterministic rate-based models typically used in machine learning, neurons communicate through roughly discrete action potentials. Further, they exhibit numerous forms of variability due in part to synaptic failures and constant receipt of task-irrelevant signals (Fig. 2.2a) [Faisal et al. 2008]. Normative theories which employ rate-based activations [Bredenberg et al. 2020; Scellier and Bengio 2017] or which assume that the input-output function of neurons is approximately linear [Oja 1982], may not extend to the more realistic discrete, stochastic, and highly nonlinear setting. Further, by ignoring spike timing, such theories inherently produce plasticity rules that ignore the precise relationship between pre- and post-synaptic spike times, and will consequently be unable to capture STDP results. This both limits the expressive power of such models, and prevents their experimental validation. Fortunately, several methods which were originally formulated using rate-based models have subsequently been extended to spiking network models to great effect. Reward-based Hebbian plasticity based on the REINFORCE algorithm (Appendix A.3) [Williams 1992] has been generalized to stochastic spiking networks [Frémaux et al. 2013], while backpropagation approximations [Murray 2019] and predictive coding methods [Rao and Ballard 1999] have subsequently been extended to deterministic spiking networks [Bellec et al. 2020; Brendel et al. 2020]. Therefore, a lack of a generalization to spiking networks is not necessarily a death knell for a normative theory, but many existing theories lack either an explicit generalization to spiking or a clear relationship to STDP, and the mathematical formalism that defines these methods may require significant modification to accommodate the change.

Real biological networks have a diversity of cell types with different neurotransmitters and connectivity motifs. At the bare minimum, a normative model must be able to accommodate

Dale's Law (Fig. 2.2a), which stipulates that the neurotransmitters released by a neuron are either excitatory or inhibitory, but not both (for the most part [O'Donohue et al. 1985]). Though this might seem like a simple principle, the mathematical results of *many* canonical models of synaptic modification rely on symmetric connectivity between neurons, including Hopfield networks [Hopfield 1982], Boltzmann machines [Ackley et al. 1985], contrastive Hebbian learning [Xie and Seung 2003], and predictive coding [Rao and Ballard 1999], as well as several more recent methods including weight mirror [Akrout et al. 2019] and equilibrium propagation [Scellier and Bengio 2017]; this symmetry is partially related to the symmetric connectivity required by the backpropagation algorithm (Appendix A.1). Symmetric connectivity means that the connection from neuron A to neuron B must be the same as the reciprocal connection from neuron B to neuron A. Symmetric connectivity inherently violates Dale's Law, because it means that entirely excitatory and entirely inhibitory neurons can never be connected to one another: the positive sign for one synapse and the negative sign for the reciprocal connection violates symmetry. Some models, such as Hopfield networks [Sompolinsky and Kanter 1986] and equilibrium propagation [Ernoult et al. 2020] demonstrate that moderate deviations from symmetry can exist and still preserve function. Other methods argue that the individual rate-based units in the normative theory correspond to small populations of coupled excitatory and inhibitory spiking neurons similar to the Wilson-Cowan population dynamics model [Payeur et al. 2021]. While this may well be possible, it implies that a normatively derived weight modification rule applies to functional connectivity between populations, not individual synapses, and additional work is required to connect this form of mass modification to empirically observed plasticity at individual synapses, as in [Payeur et al. 2021]. Further, such theories suffer the additional difficulty that there is no clear way of delineating which neurons in an experimentally observed population belong to which theoretical subpopulation, making experimental validation much more difficult. Lastly, the assumption that all neurons in a subpopulation act as a unified group threatens to violate sensory coding principles such as efficiency [Simoncelli and Olshausen 2001] and sparsity [Olshausen

et al. 1996] of neural responses, and unrealistically limits the dynamical repertoire of recurrent networks [Hennequin et al. 2012]. Thus we argue that, if possible, normative theories should avoid assumptions of symmetry.

Many early plasticity models, including Oja's rule [Oja 1982] and perceptron learning [Rosenblatt 1958], as well as more modern recurrent network models, focused on learning temporal tasks [Murray 2019] are designed exclusively for single-layer networks, and do not generalize to multi-layer architectures. Though greedy layer-wise optimization may be sufficient for some forms of unsupervised learning [Illing et al. 2021], a method that cannot account for how credit assignment signals are passed between cortical areas will not in general be able to support many complex supervised or reinforcement learning tasks humans are known to learn [Lillicrap et al. 2020]: we will refer to this form of multi-layer signal propagation as 'spatial' credit assignment, and will refer to relaying information across time as 'temporal' credit assignment (Fig. 2.2b; Section 2.2.4). As we will discuss in the next section, models that do not support temporal credit assignment will not be able to account for learning in inherently sequential tasks.

2.2.4 Handling temporal inputs

Because so many learned biologically-relevant tasks involving temporal decision-making [Gold and Shadlen 2007] or working memory [Compte et al. 2000; Wong and Wang 2006; Ganguli et al. 2008] inherently leverage information from the past to inform future behavior, and because neural signatures associated with these tasks exhibit rich recurrent dynamics [Brody et al. 2003; Shadlen and Newsome 2001a; Mante et al. 2013; Sohn et al. 2019], a fully sufficient normative theory of synaptic plasticity *must* work in recurrent neural architectures, and must provide an account of temporal credit assignment in the brain.

Temporal credit assignment is an important point of failure of modern deep learning methods, in part due to the inherent instabilities involved in performing gradient descent on recurrent neural architectures [Bengio et al. 1994]. That models unconstrained in their correspondence to

biology have difficulties handling temporal signals should be some indication of the difficulties posed by temporal credit assignment for normative theories of synaptic plasticity. However, recent improvements in neural architectures, including gated recurrent units [Chung et al. 2014] and long short-term memory units [Hochreiter and Schmidhuber 1997], as well as sequential reinforcement learning methods [Mnih et al. 2015], have combined to produce several high-profile advances on inherently temporal, naturalistic tasks like game-playing [Silver et al. 2017] and natural language processing [Devlin et al. 2018; Radford et al. 2018]. This may indicate that the time is ripe to begin incorporating new temporal processing developments in deep learning into normative plasticity models.

As it currently stands, the majority of normative synaptic plasticity models focus only on spatial credit assignment, which presents distinct challenges when compared to temporal credit assignment [Marschall et al. 2020]. In fact, many theories that provide a potential solution to spatial credit assignment do so by requiring networks to relax to a 'steady-state' on a timescale much faster than inputs [Hopfield 1982; Scellier and Bengio 2017; Bredenberg et al. 2020; Xie and Seung 2003; Ackley et al. 1985], which effectively prevents networks from having the rich, slow internal dynamics required for many temporal motor [Hennequin et al. 2012] and working memory [Wong and Wang 2006] tasks. While these methods might require significant modifications on a mathematical level to accommodate temporal credit assignment, other methods appear to be agnostic to the temporal properties of their inputs, and may combine well with existing plasticity rules that perform approximate temporal credit assignment within microcircuits [Dayan and Hinton 1996; Miconi 2017; Murray 2019; Bellec et al. 2020].

While most normative theories focus on spatial credit assignment, some new algorithms do provide potential solutions to temporal credit assignment, through either explicit approximation of real time recurrent learning [Marschall et al. 2020; Bellec et al. 2020; Murray 2019], by leveraging principles from control theory [Gilra and Gerstner 2017; Alemi et al. 2018], or by leveraging principals of stochastic circuits that are fundamentally different from traditional explicit gradient-

based calculation methods [Bredenberg et al. 2021; Miconi 2017]. We suggest that these models capture something fundamentally lacking from theories focused exclusively on spatial credit assignment, and future iterations of synaptic plasticity theory will likely want to draw inspiration from them.

2.2.5 Online learning

Similar to being able to handle temporal inputs, in absence of convincing experimental evidence, we argue that learning algorithms should perform online, meaning that learning can occur continuously with action and perception in an environment (Fig. 2.2b). In light of humans' ability to improve on tasks while performing them, the actively changing properties of neurons during acclimatization to new environments [Bittner et al. 2015], and the absence of evident non-perceptual phases during these periods (but see hippocampal replay [Pavlides and Winson 1989]), normative theories must acknowledge that requiring the existence of a distinct learning phase constitutes a very probably lethal testable prediction.

In spite of this, many algorithms [Ackley et al. 1985; Xie and Seung 2003; Dayan et al. 1995; Scellier and Bengio 2017] require distinct training phases, during at least one phase of which activity of neurons is driven for *learning*, rather than perceptual purposes. Some existing two-phase normative algorithms, such as the Wake-Sleep algorithm (Appendix A.4) [Hinton et al. 1995; Dayan et al. 1995], can be adapted such that the second phase becomes indistinguishable from perception [Bredenberg et al. 2021; Ernoult et al. 2020], or allow for simultaneous multiplexing of top-down learning signals and bottom-up inputs [Payeur et al. 2021]; others do not require neural activation functions to be modified by the learning process [Bellec et al. 2020; Illing et al. 2021], which allows for online learning. It is well known that significant synaptic modification and consolidation occurs during sleep [Eschenko et al. 2008; Girardeau et al. 2009], a period of time marked by either dreaming or a lack of perception: while some theories do propose that a form of learning occurs during this time period [Deperrois et al. 2021], exact experimental

confirmation of these algorithms is still pending. Further, while sleep may account for alternate phases of learning algorithms that are intended to occur on a slow timescale, such as unsupervised sensory representation learning or memory consolidation, it is not possible that these forms of learning can account for improvements in behavioral performance that are well-documented to occur within a single period of wakefulness.

2.2.6 Scaling in dimension and complexity

A point often underappreciated in computational neuroscience (and possibly overappreciated in machine learning) is that algorithms must be able to scale to human-level performance in order to be biologically plausible. As obvious as this sounds, it is a point that can be difficult to verify: how can we guarantee that adding more neurons and more complexity will not make a particular collection of plasticity rules more effective? As a case study, consider REINFORCE ([Williams 1992]; Appendix A.3), an algorithm which, for the most part, satisfies our other desiderata for normative plasticity for the limited selection of tasks in naturalistic environments which are explicitly rewarded. However, though REINFORCE demonstrably performs better than its progenitor weight perturbation [Jabri and Flower 1992], as the dimensionality of its stimuli, the number of neurons in the network, and the delay time between neural activity and reward increases, the performance of the algorithm decays rapidly, both analytically and in simulations [Werfel et al. 2003]. This is primarily caused by high variance of gradient estimates provided by the REINFORCE algorithm, and is only partially ameliorated by existing methods that reduce its variance [Bredenberg et al. 2021; Ranganath et al. 2014; Mnih and Gregor 2014; Miconi 2017]. Thus, adding additional complexity to the network architecture actually *impairs* learning.

Complexity is multifaceted, and involves features of both stimulus and task (Fig. 2.2c). Even stimuli with very high dimensional structure can fail to capture critical features of naturalistic stimuli, as evidenced by the wide gap in difficulty involved in constructing convincing models that synthesize images with low-level naturalistic features (orientation, contrast, texture [Portilla

and Simoncelli 2000]) compared to models that capture high-level image features (object identity [Rezende et al. 2014; Goodfellow et al. 2014], and semantic content [Ramesh et al. 2021]), which are only just beginning to emerge. Algorithms that scale well with the dimensionality of a stimulus can fail to capture high-level stimulus features: for example, PCA-based image models are unable to capture natural image statistics, and do not result in realistic neural receptive field properties [Olshausen et al. 1996]. For these reasons, it is critical that normative plasticity algorithms be able to scale not just to high-dimensional 'toy' datasets, but also to complex naturalistic data across sensory modalities [Bartunov et al. 2018].

Similarly, naturalistic task structures are often much more complex than those used for training general machine learning algorithms, let alone models of normative plasticity (Fig. 2.2c). In natural environments, rewards are often provided after long sequences of complex actions, supervised feedback is sparse, if present at all, and an organism's self preservation often requires navigating both uncertainty and complex multi-agent interactions. Modern reinforcement learning algorithms are only just beginning to make progress with some of these difficulties [Kaelbling et al. 1998; Zhang et al. 2021], but as yet there are no normative plasticity models that describe how any of the human capabilities used to solve these problems could be learned through cellular adaptation (for example, model-based planning [Doll et al. 2012]); similarly, none of these capabilities have been shown to be an emergent consequence of a more basic plasticity process.

We do not mean to imply that all normative plasticity algorithms should be demonstrated to meet human-level performance, or even that they should match state-of-the-art machine learning methods. Machine learning methods profit in many ways from their biological implausibility: they use stochastic backpropagation, which is demonstrably biologically implausible (Appendix A.1) but which benefits from very low variance gradient estimates [Werfel et al. 2003; Bredenberg et al. 2021]; they share weights across topographically distant space in convolutional neural networks [Fukushima and Miyake 1982]; they use rate-based units, which generally perform better than spiking units [Neftci et al. 2019]; and they are usually deterministic, which obviates the need for

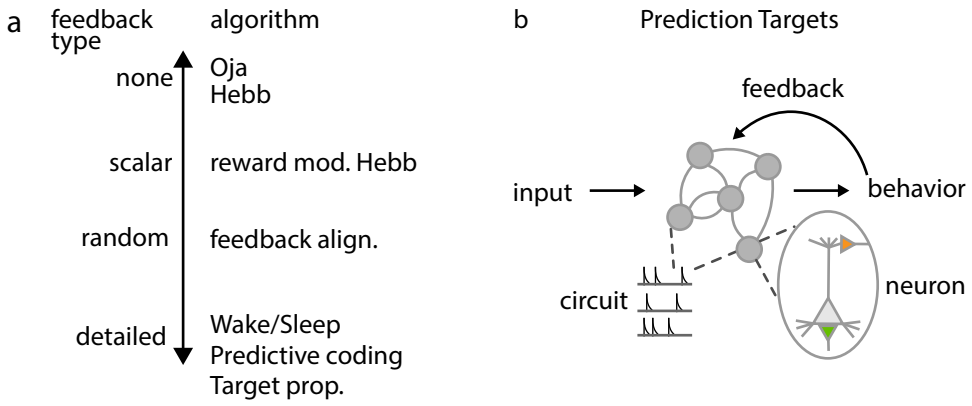

Figure 2.3: Testing normative theories a. Different normative plasticity theories postulate different levels of detail for the feedback signals received by individual neurons. **b.** Normative plasticity theories can be assessed through four different experimental lenses centered on individual neurons, circuits of collectively recorded neurons, the training signals delivered to a circuit, and the organism's overall behavior over the course of learning.

redundancy (increased neuron numbers) and increased computational demand. Beyond machine learning methods, the human brain itself has orders of magnitude more neural units and synapses than have ever been simulated on a computer, all of which are capable of processing totally in parallel. Therefore, direct comparison to the human—or any—brain is also not fair. We propose the far softer condition that as the complexity of input stimuli and tasks increase, within the range supported by current computational power, plasticity rules derived from normative theory should continue to perform well both in simulation and, preferably, analytically. Further, the performance of normative plasticity algorithms can fruitfully be compared to existing machine learning methods as long as the comparison is performed for realistic network architectures with identical conditions, as in [Bredenberg et al. 2021; Payeur et al. 2021; Marschall et al. 2020; Bartunov et al. 2018].

2.2.7 Generating testable predictions

Despite the abundance of existing normative theories, very few have been confirmed experimentally, and of those that have received partial confirmation, they are restricted to very specific

experimental preparations, for example: fear conditioning in *Aplysia* [Rayport and Schacher 1986], and reward-based learning in songbird motor systems [Fiete et al. 2007] and in mouse auditory cortex [Froemke et al. 2013; Guo et al. 2019]. This relative paucity of validation will not be overcome without a very clear articulation of which features of a normative theory constitute testable predictions, and in what way those predictions disambiguate one theory from its alternatives.

Many existing features of normative theories would be fatal to those theories if proven not to hold in biology. Some examples include: weight symmetry, reward modulation of plasticity, differential roles (and plasticity rules) for apical and basal synapses, and the existence of eligibility traces for temporal credit assignment. However, these individual features, if *proven* to hold, would eliminate alternative theories to highly variable degrees. Most, if not all models could accommodate weight symmetry, several distinct models predict reward modulation of plasticity either through precise credit assignment or global neurotransmitter delivery [Murray 2019; Williams 1992; Bellec et al. 2020; Roth et al. 2018], and several distinct supervised and unsupervised models predict different types of signaling and plasticity at apical and basal synapses on pyramidal neurons [Urbanczik and Senn 2014; Payeur et al. 2021; Bredenberg et al. 2021; Körding and König 2001; Schiess et al. 2016; Sacramento et al. 2017; Guerguiev et al. 2017; Richards and Lillicrap 2019], while nearly all models capable of temporal credit assignment assume some form of synaptic eligibility trace [Bellec et al. 2020; Marschall et al. 2020; Murray 2019; Miconi 2017; Roth et al. 2018]. It is intuitively clear that for any given normative theory of synaptic plasticity, there exist an infinite number of infinitesimal perturbations to that theory that would be impossible to disambiguate experimentally. Further, there are many features of normative theories that would be fatal if proven not to hold, but are completely unclear how to test experimentally.

The most useful predictions, we hold, are those that are fatal to the theory if proven false, are clearly testable, and disambiguate the theory from the greatest number of alternative theories. It may be that a collection of predictions is required to completely isolate one individual normative theory from closely related models, which suggests that articulating where particular models lie

within a taxonomy of predictions is the most useful way to narrow down the field of possible models. Testable predictions can be defined in terms of several different experimental lenses, of which we isolate four: experiments examining individual neurons or synapses, populations of neurons, the feedback mechanisms that shape learning in neural circuits, or learning at a behavioral level (Fig. 2.3b). Accurately distinguishing one mechanism from another will likely require a synthesis of experiments spanning all four lenses.

Individual neurons Experiments that focus on individual neurons, including paired-pulse stimulation [Markram et al. 1997], mechanistic characterizations of plasticity [Graupner and Brunel 2010], pharmacological explorations of neuromodulators that induce or modify plasticity [Bear and Singer 1986; Reynolds and Wickens 2002; Froemke et al. 2007; Gu and Singer 1995], and characterization of local dendritic or microcircuit properties mediating plasticity [Froemke et al. 2005; Letzkus et al. 2006; Sjöström and Häusser 2006] form the bulk of the classical literature underlying phenomenological and mechanistic modeling. These studies characterize what information is locally available at synapses and what can be done with that information, as well as which properties of cells can be altered in an experience-dependent fashion.

Existing normative theories differ in the nature of their predictions for plasticity at individual neurons. Reward-modulated Hebbian theories *require* feedback information be delivered by a neuromodulator like dopamine, serotonin, or acetylcholine [Frémaux and Gerstner 2016] and that this feedback modulates plasticity at the local synapse by changing the magnitude or sign of plasticity depending on the strength of feedback. In contrast, some unsupervised normative theories require no feedback modulation of plasticity [Pehlevan et al. 2015, 2017], and others argue that detailed feedback information arrives at the apical dendritic arbors of pyramidal neurons to modulate plasticity, which is also partially supported in the hippocampus [Bittner et al. 2015, 2017] and cortex [Larkum et al. 1999; Letzkus et al. 2006; Froemke et al. 2005; Sjöström and Häusser 2006].

Independent of the exact feedback mechanism, models differ in how temporal associations

are formed. Algorithms related to REINFORCE assume that local records of coactivity, called 'eligibility traces' integrate over time fluctuations in coactivity of the post- and pre-synaptic neuron local to a synapse. These postulated eligibility traces are stochastic, summing Gaussian fluctuations in activity [Miconi 2017] that consequently produce temporal profiles similar to Brownian motion. In contrast, methods based on approximations to real-time recurrent learning propose eligibility traces that are deterministic records of coactivity whose time constants are directly connected to the dynamics of the neuron itself [Bellec et al. 2020], while other hybrid approaches predict eligibility traces which are deterministic but are related more to predicted task timescale than the dynamics of the cell [Roth et al. 2018]. Though there do exist known cellular processes that naturally track coactivity, like NMDA receptors [Bi and Poo 1998], and that store traces of this coactivity longitudinally, like CaMKII [Graupner and Brunel 2010], much work remains to be done to analyze how the properties of these known biophysical quantities relate to the predictions of various normative theories, and whether there are other biological alternatives.

Other algorithms have different predictions at a microcircuit, rather than at an individual neuron level. Impression learning, for instance, suggests that a population of inhibitory interneurons could gate the influence of apical and basal dendritic inputs to the activity of pyramidal neurons [Bredenberg et al. 2021], and some forms of predictive coding propose that top-down error signals are partially computed by local inhibitory interneurons. Therefore, to completely distinguish different theories, it may be necessary to analyze the connectivity and plasticity between small groups of different cell types.

In sum, experiments at the level of individual neurons or local microcircuits potentially have a great deal of power to identify whether a particular neural circuit is implementing any of a collection of hypothesized normative models of plasticity. It is an advantage that these methods can identify the adaptive capabilities of individual neurons and synapses, but these methods are also limited in their ability to simultaneously observe the adaptation of many neurons in a circuit. Normativity is inherently concerned with the value of plasticity for perception and behavior, and

as we will see in subsequent sections, experiments that target larger populations of neurons will be necessary to distinguish certain features of these theories.

Neural circuits How circuits encode environmental information and affect motor actions by an animal cannot be determined by looking at single neurons, and by extension, analyzing how these properties change over time requires methods that record large groups of neurons, such as 2-photon calcium imaging, multielectrode recordings, fMRI, EEG, and MEG, as well as methods that manipulate large populations, like optogenetic [Rajasethupathy et al. 2016a] stimulation. The benefits of these recording techniques for testing normative plasticity models, though less practiced compared to individual neuron studies, are manyfold. One of the challenges for characterizing a circuit with a normative plasticity model is selecting an appropriate objective function. Determining which objective fits best can partly be determined by philosophical considerations (Section 2.2.2), but empirical validation is a far more rigorous test. For instance, one can establish that explicit reward modifies a neural representation to improve coding of task-relevant variables [Froemke et al. 2013]. Another line of approaches trains neural networks on a battery of objectives, and determines which objective produces the closest correspondence between model neurons and neurons recorded brain in a variety of areas in the ventral [Yamins et al. 2014; Yamins and DiCarlo 2016] and dorsal [Mineault et al. 2021] visual streams, as well as recently in auditory cortex [Kell et al. 2018] and medial entorhinal cortex [Nayebi et al. 2021]. Oftentimes, changes in artificial neural network activity throughout time are sufficient to determine the objective optimized by the network as well as its learning algorithm [Nayebi et al. 2020], an approach which could also potentially be applied to recorded neural activity over learning.

Beyond narrowing down the objective function, recording from populations can establish critical limitations to learning that may not exist in artificial circuits. For instance, in a brain computer interface training paradigm, motor neurons adapt more slowly to decoder perturbations that lie outside of the principal axes of network activity [Golub et al. 2018]; recent results have shown that artificial networks show similar behavior only if the learning algorithm used by the

circuit is noisy or imprecise compared to perfect gradient descent [Feulner and Clopath 2021]. Further, circuit recordings could in principle test predictions about how neural circuits should function in situations that do not specifically involve learning. For instance, the Wake-Sleep algorithm [Dayan et al. 1995] (Appendix A.4) proposes that neural circuits should spend extended periods of time (e.g. during dreaming) occupying similar activity patterns to those evoked by natural stimulus sequences, whereas impression learning proposes that similar hallucinatory states could be induced by experimentally increasing the influence of apical dendrites on pyramidal neuron activity [Bredenberg et al. 2021]. An alternative learning algorithm based on generative adversarial networks proposes that during sleep networks rehearse corrupted versions of recent waking experiences [Deperrois et al. 2021]. There is plenty of room for experiments to more clearly map predictions and components of these models onto well documented neural phenomena, such as sleep or potentially replay [Girardeau et al. 2009; Eschenko et al. 2008]. Because circuit recording and manipulation methods often sacrifice temporal resolution [Hong and Lieber 2019], and have difficulty inferring biophysical properties of individual synapses and cells, these methods are best used in concert with single neuron studies to jointly tease apart the multi-level predictions of various normative models.

Feedback mechanisms One of the best ways to distinguish normative plasticity algorithms is on the basis of the nature of their feedback mechanisms (Fig. 2.3a). Though some unsupervised algorithms, like Oja's rule propose that no feedback is necessary to perform meaningful learning, no current normative theories propose any form of supervised or reinforcement learning that does not require *some* form of top-down feedback. However, across these models, the level of precision of feedback varies considerably. The simplest feedback is scalar, conveying reward [Williams 1992], state fluctuation [Payeur et al. 2021], or context (e.g. saccade [Illing et al. 2021] or attention [Roelfsema and Ooyen 2005; Pozzi et al. 2020]) information. Beyond this, the space of proposed mechanisms expands considerably: backpropagation approximations like feedback alignment [Lillicrap et al. 2016] and random-feedback online learning (RFLO) [Murray 2019]

propose random feedback between layers of neurons can provide a sufficient learning signal, whereas algorithms based on control theory propose that low-rank or partially random projections carrying supervised error signals are sufficient [Gilra and Gerstner 2017; Alemi et al. 2018]. Other algorithms propose even more detailed feedback, with individual neurons receiving precise, carefully adapted projections carrying learning-related information. These algorithms propose that top-down projections to apical dendrites [Urbanczik and Senn 2014] or local interneurons neurons [Bastos et al. 2012] perform spatial credit assignment, but the nature of this signal can differ considerably across different algorithms. It could be a supervised target, carrying information about what the neuron state 'should' be to achieve a goal [Guerguiev et al. 2017; Payeur et al. 2021], or it could be a prediction of the future state of the neuron [Bredenberg et al. 2021].

Each of these different possibilities is theoretically testable, if the focus is shifted to the postulated feedback mechanism instead of the circuit undergoing learning. However, so far the different mechanisms have received only partial support. For example, acetylcholine projections to auditory cortex that modulate perceptual learning [Froemke et al. 2013] display a diversity of responses related to both reward and attention [Hangya et al. 2015], which adapt over the course of learning in concert with auditory cortex [Guo et al. 2019]. This suggests that while traditional models of reward-modulated Hebbian plasticity may be correct to a first approximation, a more detailed study of the adaptive capabilities of neuromodulatory centers may be necessary to update the theories.

While a growing number of studies indicate that projections to apical synapses of pyramidal neurons *do* play a role in inducing plasticity, and that these projections themselves are also plastic (i.e. nonrandom) [Bittner et al. 2015, 2017], very little is known about the *nature* of the signal—a critical component for distinguishing several different theories. In the visual system, presentation of unfamiliar images without any form or reward or supervision can modify both apical and basal dendrites throughout time [Gillon et al. 2021], and in the hippocampus, apical input to CA1 pyramidal neurons while animals acclimatize to new spatial environments is sufficient to

induce synaptic plasticity [Bittner et al. 2015, 2017]. These two examples support a form of *unsupervised* learning, but evidence for supervised or reinforcement learning signals propagated through apical dendritic synapses is currently lacking. Beyond the cerebellar system, where climbing fiber pathways may carry explicit motor error signals used for plasticity [Gao et al. 2012; Bouvier et al. 2018], evidence for detailed supervised feedback is limited. In sum, beyond single neurons, or even populations recorded by traditional techniques, targeted focus on the learning feedback signals received by a population show promise to rule out algorithms on the basis of their feedback and objective function.

Behavior In much the same way that psychophysical studies of human or animal responses define constraints on what the brain's perceptual systems are capable of, behavioral studies of learning can do quite a lot to describe the range of phenomena that a model of learning must be able to capture, from operant conditioning [Niv 2009], to model-based learning [Doll et al. 2012], rapid language learning [Heibeck and Markman 1987], unsupervised sensory development [Wiesel and Hubel 1963], or consolidation effects [Stickgold 2005]. Behavioral studies can also outline key limitations in learning, which are perhaps reflective of the brain's learning algorithms, including the brain's failure to perform certain types of adaptation after critical periods of plasticity [Wiesel and Hubel 1963], and the brain's unexpected inability to learn multi-context motor movements without explicit motor differences across contexts [Sheahan et al. 2016].

These existing experimental results stand as (often unmet) targets for normative theories of plasticity, but in addition, normative theories themselves suggest further studies that may test their predictions. In particular, manipulation of learning mechanisms may have predictable effects on animals' behavior, as seen when acetylcholine receptor blockage in mouse auditory cortex prevented reward-based learning in animals [Guo et al. 2019], and nucleus basalis stimulation during tone perception longitudinally improved animals' discrimination of that tone [Froemke et al. 2013]. Other algorithms have as-yet untested predictions for behavior: for instance, experimentally increasing the influence of top-down projections should bias behavior towards

Algorithm	Local	Dec. Loss	Time	Flex. Arch.	Online	Scalable
Backpropagation	✗	U/S/R	✓	✓	✓	✓
REINFORCE	✓	U/S/R	✓	✓	✓	✗
Oja	✓	U	✗	✗	✓	✓
Predictive Coding	✓	U	✓	✗	✓	✓
Wake-Sleep	✓	U	✓	✓	✗	✓
Approx. Backprop.	✓	U/S*	✓	✓	✓	✓
Equilibrium Prop.	✓	U/S	✗	✗	✓	✓
Target Prop.	✓	U/S	✓	✓	✗	✓

Table 2.1: Satisfying the desiderata. A ✓ indicates that an algorithm has been demonstrated to satisfy a particular desideratum in at least one study, whereas an ✗ indicates that it has not been demonstrated. Asterisks (*) indicate that results have only been shown by simulation, and lack mathematical support. U, S, and R indicate whether a given algorithm supports unsupervised, supervised, or reinforcement learning, respectively.

commonly-occurring sensory stimuli according to both predictive coding [Rao and Ballard 1999; Friston 2010] and impression learning [Bredenberg et al. 2021]. For other detailed feedback algorithms (Fig. 2.3a), manipulating top-down projections may disrupt learning, but would have a much more unstructured deleterious effect on perceptual behavior.

As shown, each experimental lens has its own advantages and disadvantages. Single-neuron studies are excellent for identifying the locally available variables that affect plasticity, circuit-level studies can help narrow down the objectives that shape neural responses and identify traces of offline learning, studies of feedback mechanisms can distinguish between different algorithms that postulate different degrees of precision in their feedback and in complexity of the teaching signal, and studies of behavior can place boundaries on what can be learned, as well as serve as a readout for manipulations of the mechanisms underlying learning. Each focus alone is insufficient to distinguish between all existing normative models, but in concert show promise for identifying the neural substrates of adaptation.

2.3 Conclusions

Normative plasticity models are compelling because of their potential to connect our brains' capacity for adaptation to their constituent synaptic modifications. Generating good theories is a critical part of the scientific process, but finding ways to close the loop by testing key predictions of new normative models has proved extraordinarily difficult: in this chapter we have shown within the anatomy of a normative model the sources of this difficulty.

The core of a normative plasticity model is its plasticity rule, which dictates how a model synapse uses locally available information to modify its strength. To be a normative model—to explain why the plasticity mechanism is important for the organism—there must be a concrete demonstration that this plasticity rule supports adaptation critical for system-wide goals like processing sensory signals or obtaining rewards. These two needs of a normative plasticity model are the fundamental source of tension: it is very difficult to demonstrate that a proposed plasticity rule is both local *and* optimizes a system-wide objective (Appendix A.1). Insufficient or partial resolution of this fundamental tension produces normative models that are not well enough developed at a mathematical level to warrant testing in the first place: lacking convincing arguments that a theory is local (Section 2.2.1), reduces an objective function critical for an organism's survival (Section 2.2.2), could generalize to the full complexity of neural architecture found throughout the brain (Section 2.2.3), and can handle complex temporal stimuli and tasks online (Sections 2.2.4 and 2.2.5), there is very little reason to suppose the brain could ever make constructive use of the proposed plasticity.

Even satisfying the aforementioned desiderata, much work remains to delineate which tests would most clearly distinguish a normative model from its alternatives in a biological system. In this chapter, we have organized existing theories according to how well they satisfy our desiderata (Table 2.1) and by how they can be tested (Section 2.2.7), with the view that this organization will provide avenues for both experimental and theoretical neuroscientists to bring normative plasticity

models closer to biology. Even if existing algorithms prove not to be implemented exactly in the brain, they undoubtedly provide key insights into how local synaptic modifications can produce valuable improvements in both behavior and perception for an organism. It seems sensible to use these algorithms as a springboard to produce more biologically realistic and powerful theories.

For instance, REINFORCE suffers from scalability issues (Table 2.1), but provides one of the only known mechanisms for improving performance using only scalar signals based on raw reward. Given that the scalability of REINFORCE is primarily limited by its inability to provide structured feedback to individual neurons or small groups [Werfel et al. 2003], we propose that hybridized versions in which reward centers actively model and decompose an environment's reward contingencies into subgoals or targets for small populations of neurons will produce closer matches to biology and huge improvements in performance. Such a normative model could potentially be constructed from a model-based reinforcement learning [Doll et al. 2012], reward-based backpropagation approximation [Roelfsema and Ooyen 2005; Roelfsema et al. 2010], or active inference perspective [Sajid et al. 2021; Isomura et al. 2022]. Algorithms like Wake-Sleep [Hinton et al. 1995; Dayan et al. 1995] and variants of target propagation [Bengio 2014] are very closely related, and complementary to REINFORCE in that they scale well, but are unable to learn from reinforcement signals alone. Both algorithms are offline, and so an alternative avenue of improvement is to show how they can be adapted to online learning in the brain (see Chapter 5; [Bredenberg et al. 2021]). These algorithms involve a top-down model of neural activity, in which task or statistical information is used to project credit assignment signals to individual neurons that assert what the activity of a neuron *should* be. These algorithms, or related control-theoretic approaches [Meulemans et al. 2020; Friedrich et al. 2021] may combine well with reinforcement learning to provide more efficient, model-based forms of learning. The Wake-Sleep algorithm and REINFORCE have complementary benefits and flaws, but are mathematically very closely related. To provide a case study of how our desiderata come to be satisfied (or not) in practice, we have included tutorials for these two algorithms in Appendices A.3 and A.4. These tutorials are

by no means a complete introduction to the field, but will hopefully serve as a solid foothold for analyzing more modern normative plasticity models.

Beyond improving normative theories with respect to our desiderata, there are several incredible opportunities for actually testing their implementation in biology (Section 2.2.7). Most current theoretical studies of reward-modulated Hebbian plasticity focus on dopamine-modulated motor learning in monkeys and songbirds [Fiete et al. 2007; Legenstein et al. 2010], but there are *many* neuromodulatory systems that have been linked to learning in experiments, including serotonin-modulated fear conditioning in the amygdala [Lesch and Waider 2012], as well as acetylcholine-modulated reward learning and oxytocin-modulated social learning in mouse auditory cortex [Guo et al. 2019; Froemke et al. 2013]. Further, several experimental preparations examine the relationship between pyramidal neurons' apical and basal dendritic activity and plasticity, in both the hippocampus [Bittner et al. 2015, 2017] and visual cortex [Gillon et al. 2021; Froemke et al. 2005; Letzkus et al. 2006; Sjöström and Häusser 2006]. These could test at the level of individual neurons, circuits, behavior, and the feedback mechanisms that support plasticity, which of the many alternative normative theories underlie animals' learning.

As the diversity of aforementioned experimental preparations suggests, there are increasingly strong arguments for several fundamentally different normative plasticity theories existing in different areas of the brain, and subserving different functions. It is quite likely that many plasticity mechanisms work in concert to produce learning as it manifests in our perception and behavior. It is our belief that well-articulated normative theories can serve as the building blocks of a conceptual framework that tames this diversity and allows us to understand the brain's tremendous capacity for adaptation. In Chapters 3 and 4, we will explore how reward-based synaptic plasticity interacts with task constraints and sensory statistics to shape neural representations. Then, in Chapter 5, we will develop a theory of synaptic plasticity for *unsupervised* representation learning, which could explain how sensory systems build models of their environment in the absence of any form of task feedback. Finally, in Chapter 6, we will show how continual synaptic modification in neural

circuits can confound the interpretation of causal manipulations in perceptual discrimination tasks. These results collectively demonstrate the importance of normative synaptic plasticity theories for neuroscience.

3 | Learning efficient task-dependent representations with synaptic plasticity

A variety of forces shape neural representations in the brain. On one side, sensory circuits need to faithfully represent their inputs, in support of the broad range of tasks an animal may need to perform. On the other side, the neural 'wetware' is intrinsically noisy [Faisal et al. 2008], and computing resources are highly limited in terms of the number of neurons and metabolic energy. It remains a mystery how local synaptic learning rules can overcome these limitations to yield robust representations at the circuit level.

Past work has focused on individual aspects of this problem. Studies of efficient coding have successfully explained features of early sensory representations [Olshausen et al. 1996; Ganguli and Simoncelli 2016] in terms of the interaction between stimulus statistics and resource limitations, and several models have proposed how such representations could emerge through local unsupervised learning [Oja 1982; Rozell et al. 2008; Brendel et al. 2017]. However, the bulk of this theoretical work has ignored task constraints. This oversight might seem justified, considering that we generally think of sensory cortices as performing unsupervised learning, however a growing body of experimental evidence suggests that behavioral goals shape neural receptive fields in adult sensory cortices (A1 [Weinberger 1993; David et al. 2012; Polley et al.

2006], S1 [Recanzone et al. 1992b,a], V1 [Li et al. 2004; Schoups et al. 2001]), usually in the presence of neuromodulation [Bakin and Weinberger 1996; Kilgard and Merzenich 1998; Froemke et al. 2007]. This kind of plasticity has been modelled using tri-factor learning rules, which provide a mechanism for learning *task-specific* representations using only local information [Seung 2003; Frémaux and Gerstner 2016; Kuśmierz et al. 2017; Gerstner et al. 2018]. However, the interaction between the task, input statistics, and biological constraints remains largely unexplored (but see [Savin and Triesch 2014]).

Here we use a stochastic recurrent neural network model to derive a class of tri-factor Hebbian plasticity rules capable of solving a variety of tasks, subject to metabolic constraints. The framework leverages internal noise for estimating gradients of a task-specific objective; thus, noise provides an essential ingredient for learning, and is not simply an impediment for encoding. We systematically examine the interactions between input statistics, task constraints, and resource limitations, and show that the emerging representations select task-relevant features, while preferentially encoding commonly occurring stimuli. The network also learns to reshape its intrinsic noise in a way that reflects prior statistics and task constraints.

3.1 Task-dependent synaptic plasticity

Stochastic circuit model. We consider a simple sensory coding paradigm, in which a stimulus orientation θ is drawn from a von Mises distribution and encoded in the responses of an input population with fixed tuning, $s_j(\theta)$ (Fig. 3.1a) given by:

$$s_j(\theta) = \lfloor \cos(\theta_j)\cos(\theta) + \sin(\theta_j)\sin(\theta) \rfloor, \tag{3.1}$$

where $\lfloor \cdot \rfloor$ indicates halfwave rectification, and θ_j is the maximally-activating stimulus for input unit j. This creates a unimodal stimulus response profile, with a peak value of 1 at orientation

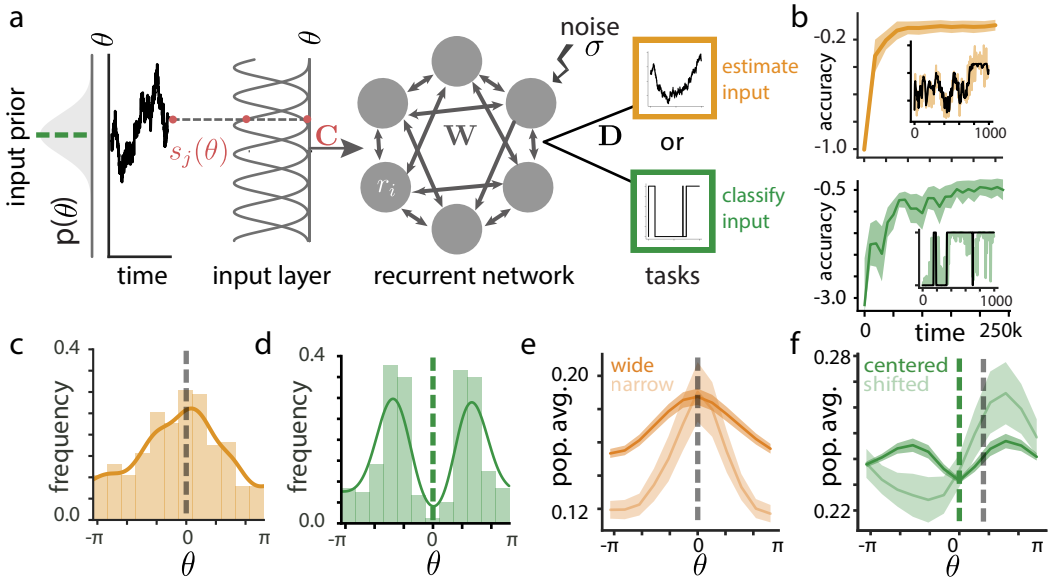

Figure 3.1: Recurrent neural network architecture and task learning. a. Model schematic. Stimuli drawn from a prior distribution (gray) are encoded in the responses of a static input layer, which feeds into the recurrent network; a linear decoder in turn produces the network output for either an estimation or a classification task. **b.** Network performance as a function of training time for the estimation (top) and classification (bottom) tasks. Shaded intervals indicate ± 1 s.e.m. across 45k test time units. Inset shows an example trained network output, with the correct output in black. **c.** Histogram of preferred stimuli and corresponding kernel density estimates (line). **d.** Same as **c**, for the classification task; decision boundary at $\theta = 0$ (dashed line). **e.** Population-averaged firing rates for two input priors. Shaded intervals are ± 1 s.e.m. averaging across all neurons in the network (light gold) or all neurons across 20 simulations (dark gold). **f.** Effects of shifting the stimulus prior relative to the classification boundary; dashed lines show the common decision boundary (green) and shifted prior (gray). Shaded intervals as in **e**.

$\theta = \theta_j$. Further, θ_j are selected to evenly tile the range $[-\pi, \pi]$.

This activity provides the input to the recurrent network via synapses with weights specified by matrix **C**. The recurrent network is nonlinear and stochastic, with neuron activities **r** and recurrent synaptic strengths **W**. A linear decoder with synaptic weights parameterized by matrix **D** transforms the network activity into a task-specific output.

The stochastic dynamics governing the activity of recurrent neurons take the form:

$$dr_i = \left[-f^{-1}(r_i) + \sum_{j=1}^{N_r} w_{ij} r_j + \sum_{k=1}^{N_s} c_{ik} s_k + b_i \right] dt + \sigma dB_i, \tag{3.2}$$

where $f(\cdot)$ is the nonlinear response function (for simplicity, a sigmoid for all neurons), N_s and N_r are the number of neurons in the input and recurrent populations, respectively, and b_i is a bias reflecting neural excitability. The parameter σ controls the standard deviation of the Brownian noise, B_i, added independently to each neuron. This intrinsic noise is one of the main constraints on the network, and could be interpreted as caused by either stochastic synaptic transmission failures or subthreshold voltage fluctuations [Faisal et al. 2008]. The $f^{-1}(r_i)$ term may seem unusual, but it allows for analytic tractability of the nonlinear dynamics. Furthermore, this formulation allows us to add Brownian noise to the current, such that fluctuations outside of those allowed by the F-I nonlinearity are sharply attenuated. In the small-noise limit, $\sigma \to 0$, the network has the same steady-state dynamics as a traditional nonlinear recurrent neural network. At equilibrium, $\bar{r}_i = f\left(\sum_j w_{ij}\bar{r}_j + \sum_k c_k s_k + b_i\right)$, with nonlinearity f determining the steady-state F-I curve for each neuron; intrinsic noise induces fluctuations about this fixed point.

When the recurrent connectivity matrix **W** is symmetric, the network dynamics are a stochastic, continuous analog of the Hopfield network [Hopfield 1982, 1987], and of the Boltzmann machine [Ackley et al. 1985]. Its corresponding energy function is:

$$E(\mathbf{r}, \mathbf{s}; \mathbf{W}) = -\frac{1}{2} \sum_{i,j} w_{ij} r_i r_j + \sum_i \int_0^{r_i} f^{-1}(x_i) dx_i - \sum_{ij} c_{ij} r_i s_j - \sum_i b_i r_i. \tag{3.3}$$

The network dynamics implement stochastic gradient descent on this energy, which corresponds to Langevin sampling [Roberts et al. 1996] from the stimulus-dependent steady-state probability distribution:

$$p(\mathbf{r}|\mathbf{s};\mathbf{W}) = \frac{\exp\left[-\frac{E(\mathbf{r},\mathbf{s};\mathbf{W})}{\sigma^2}\right]}{\int_{\mathbf{r}} \exp\left[-\frac{E(\mathbf{r},\mathbf{s};\mathbf{W})}{\sigma^2}\right]}. \tag{3.4}$$

The steady-state distribution is in the exponential family, and offers a variety of useful mathematical properties. Most importantly, the probabilistic description of the network, $p(\mathbf{r}|\mathbf{s};\mathbf{W})$, can be used to calculate the gradient of an objective function with respect to the network weights via a procedure similar to REINFORCE (Appendix A.3; [Williams 1992; Fiete et al. 2007]). In practice, we use approximate solutions to Eq. (3.2) (or equivalently, Langevin sampling from Eq.3.4) using Euler-Maruyama integration.

Task-dependent objectives. We consider a task-specific objective function of the general form:

$$O(\mathbf{W},\mathbf{D}) = \iint \alpha(\mathbf{Dr},\mathbf{s})p(\mathbf{r}|\mathbf{s};\mathbf{W})p(\mathbf{s})\ \mathbf{drds} - \lambda\left\|\mathbf{W}\right\|_2^2, \tag{3.5}$$

where $\alpha(\cdot,\cdot)$ is a task-specific loss function, computed as a function of the linear readout $\mathbf{Dr}$, in a downstream circuit. The second term ensures that synaptic weights do not grow indefinitely [Oja 1982]; it is a mathematically convenient way of introducing metabolic constraints, although regularizing the neural activity itself is also possible. For brevity, we have only included the constraint on $\mathbf{W}$ in Eq. (3.5). In practice, we also include similar constraints on $\mathbf{C}$, $\mathbf{D}$, and $\mathbf{b}$, with corresponding Lagrange multipliers λ_C, λ_D and λ_b.

The specific choice of the loss, α, determines the nature of the task. Here, we chose two example objective functions – input encoding and binary classification. For reproducing the input, we use a mean squared error (MSE) objective:

$$\alpha_{\text{MSE}}(\mathbf{Dr},\mathbf{s}) = -\left\|\mathbf{s}-\mathbf{Dr}\right\|_2^2, \tag{3.6}$$

with a negative sign reflecting the fact that we are maximizing, rather than minimizing, the objective.

For classification, we use a cross-entropy objective:

$$\alpha_{\mathrm{LL}}(\mathbf{Dr}, \mathbf{s}) = \phi(\mathbf{s}) \log(\psi(\mathbf{Dr})) + (1 - \phi(\mathbf{s})) \log(1 - \psi(\mathbf{Dr})), \tag{3.7}$$

where $\psi(\cdot)$ is a sigmoid nonlinearity and $\phi(\mathbf{s})$ gives the mapping from stimulus $\mathbf{s}$ to the corresponding binary class.

Local task-dependent learning. We derive synaptic plasticity rules by maximizing $O(\mathbf{r}, \mathbf{s}; \mathbf{W})$ using gradient ascent, averaging across the stimulus distribution $p(\mathbf{s})$, and taking advantage of the closed-form expression for the steady-state stimulus-dependent response distribution, $p(\mathbf{r}|\mathbf{s}; \mathbf{W})$.

Taking the derivative of the objective function (Eq. 3.5) with respect to w_{ij} yields:

$$\frac{\partial O}{\partial w_{ij}} = \iint p\left(\mathbf{s}\right) \alpha\left(\mathbf{Dr}, \mathbf{s}\right) \frac{\partial p\left(\mathbf{r}|\mathbf{s}\right)}{\partial w_{ij}} \mathbf{drds} - 2\lambda w_{ij}. \tag{3.8}$$

As in [Williams 1992], differentiating Eq. 3.4, we note that:

$$\frac{\partial}{\partial w_{ij}} p\left(\mathbf{r}|\mathbf{s}; \mathbf{W}\right) = \frac{-1}{\sigma^2} \left[\frac{\partial}{\partial w_{ij}} E(\mathbf{r}, \mathbf{s}; \mathbf{W}) - \left\langle \frac{\partial}{\partial w_{ij}} E\left(\mathbf{r}, \mathbf{s}; \mathbf{W}\right) \right\rangle_{p(\mathbf{r}|\mathbf{s})} \right] p\left(\mathbf{r}|\mathbf{s}; \mathbf{W}\right), \tag{3.9}$$

where the brackets denote the conditional expectation with respect to $p(\mathbf{r}|\mathbf{s}; \mathbf{W})$. Rearranging, and substituting Eq. (3.9) into Eq. (3.8) yields:

$$\begin{aligned} \frac{\partial O}{\partial w_{ij}} =& \frac{-1}{\sigma^2} \iint \alpha\left(\mathbf{Dr}, \mathbf{s}\right) \left(\frac{\partial E\left(\mathbf{r}, \mathbf{s}; \mathbf{W}\right)}{\partial w_{ij}} - \left\langle \frac{\partial E(\mathbf{r}, \mathbf{s}; \mathbf{W})}{\partial w_{ij}} \right\rangle_{p(\mathbf{r}|\mathbf{s})} \right) p(\mathbf{r}, \mathbf{s}; \mathbf{W}) \ \mathbf{drds} - 2\lambda w_{ij} \\ =& \frac{1}{\sigma^2} \iint \alpha(\mathbf{Dr}, \mathbf{s}) \left(r_i r_j - \left\langle r_i r_j \right\rangle_{p(\mathbf{r}|\mathbf{s})} \right) p(\mathbf{r}, \mathbf{s}; \mathbf{W}) \mathbf{drds} - 2\lambda w_{ij}. \end{aligned}$$

To update weights via gradient ascent, the learning rule takes the form:

$$\Delta w_{ij} \propto \frac{\partial O}{\partial w_{ij}} = \mathbb{E}\left[\alpha(\mathbf{Dr}, \mathbf{s})\left(r_i r_j - \left\langle r_i r_j \right\rangle_{p(\mathbf{r}|\mathbf{s})}\right)\right] - 2\lambda_W w_{ij},$$

where we have assigned $\lambda_W = \sigma^2 \lambda$. We further approximate the expectation by sampling $\mathbf{r}$ as part of the network dynamics and update weights according to a single sample (if Δw_{ij} is sufficiently small, this is equivalent to updating w_{ij} with an average over several samples):

$$\Delta w_{ij} \propto \alpha(\mathbf{Dr}, \mathbf{s})\left(r_i r_j - \left\langle r_i r_j \right\rangle_{p(\mathbf{r}|\mathbf{s})}\right) - 2\lambda_W w_{ij}. \tag{3.10}$$

This expression for the weight update is similar to a standard reward-modulated Hebbian plasticity rule. It is driven by correlations between pre- and post-synaptic activity, with reward, α, having a multiplicative effect on weight changes. The subtractive term ensures that updates only occur in the presence of deviations from the average correlation level. It is symmetric in the indices $\{i, j\}$, and thus preserves the symmetry of the weight matrix $\mathbf{W}$. Finally, the weight regularization adds a contribution similar in form to Oja's rule [Oja 1982]. In practice, we approximate the conditional expectation by a running average computed using a low-pass filter. We can derive similar updates for the input weights, and biases:

$$\Delta c_{ik} \propto \alpha(\mathbf{Dr}, \mathbf{s})\left(r_i s_k - \langle r_i s_k \rangle_{p(\mathbf{r}|\mathbf{s})}\right) - 2\lambda_C c_{ik}$$
$$\Delta b_i \propto \alpha(\mathbf{Dr}, \mathbf{s})\left(r_i - \langle r_i \rangle_{p(\mathbf{r}|\mathbf{s})}\right) - 2\lambda_b b_i,$$

Hence, our framework allows us to optimize parameters of the network using local operations.

It is worth comparing our plasticity rule to REINFORCE [Williams 1992], which would have, for a discrete-time RNN: $\Delta w_{ij} \propto \alpha(\mathbf{Dr}, \mathbf{s}) \sum_{t=0}^{T} f'(h_i(t))(r_i(t) - \bar{r}_i(t))r_j(t) - 2\lambda_W w_{ij}$, where h_i is the pre-activation of neuron i, and $\bar{r}_i(t)$ is the expected average activation for that neuron at time t. One notable difference is that learning is gated by deviations from the mean co-activation of

the pre- and post-synaptic neurons, and not by post-synaptic activity alone. Another difference is the presence of a sum over time (an eligibility trace), which is not required for our plasticity rule. Lastly, this rule includes an $f'(h_i(t))$ term, whereas ours only involves the firing rates of the neurons. At least some of these differences could be examined experimentally.

Learning the decoder. Because the readout weights enter $\alpha(\mathbf{Dr}, \mathbf{s})$, the optimization of D requires a slightly different treatment. Since $p(\mathbf{r}|\mathbf{s}; \mathbf{W})$ does not depend on $\mathbf{D}$, taking the derivative of Eq. (3.5) yields:

$$\frac{\partial \mathcal{O}}{\partial D_{ij}} = \int p(\mathbf{s}, \mathbf{r}; \mathbf{W}) \frac{\partial}{\partial D_{ij}} \alpha(\mathbf{Dr}, \mathbf{s}) \, \mathbf{drds} - 2\lambda_D D_{ij}.$$

Using the same stochastic update scheme as in Eq. (3.10) yields:

$$\Delta D_{ij} \propto \frac{\partial}{\partial D_{ij}} \alpha(\mathbf{Dr}, \mathbf{s}) - 2\lambda_D D_{ij}.$$

This partial derivative will be different for each choice of α. For α_{MSE}, we get:

$$\Delta D_{ij}^{(\text{MSE})} \propto 2 \left(\sum_k D_{ik} r_k - s_i \right) r_j - 2\lambda_D D_{ij}, \tag{3.11}$$

and for α_{LL} (noting that $\mathbf{D}$ here has only one row):

$$\Delta D_j^{(\text{LL})} \propto \left(\phi(\mathbf{s}) - \psi(\mathbf{Dr}) \right) r_j - 2\lambda_D D_j. \tag{3.12}$$

3.2 Numerical results

To simulate the realistic scenario of slowly changing input stimuli constantly driving the circuit, we sample inputs from a von Mises prior distribution using Langevin dynamics with a significantly slower time constant than that of the recurrent network dynamics, given by $\tau_s = 375$. We set the noise level to an intermediate regime, so that its effects on circuit computation are non-negligible,

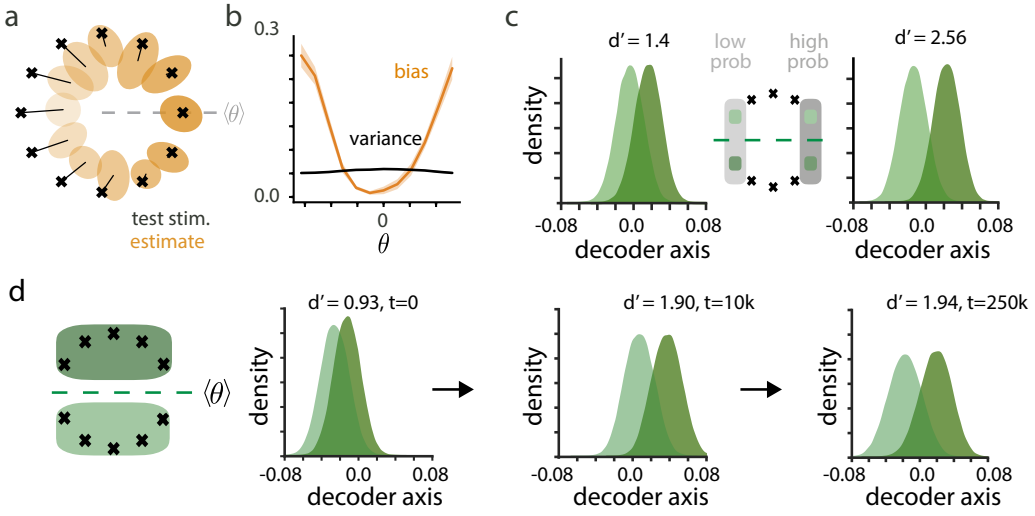

Figure 3.2: Task-specific stimulus encoding. a. Outputs for the estimation task: crosses mark the locations of test stimuli; black lines indicate the bias, or the distance between the target and the mean output; ellipses show the 95% probability region for the associated network response distributions. Lighter colors are further from the most probable stimulus ($\langle\theta\rangle = 0$). **b.** Squared bias and variance for the estimator task as a function of stimulus angle. Shaded intervals are ± 1 s.e.m. across 20 simulations. **c.** Decoder response distribution for stimuli near the boundary in the low-probability region of the space ($\theta = \pi$, left), and in the high-probability region ($\theta = 0$, right), as indicated on the center schematic. **d.** Left: network output schematic for test stimuli (black crosses). The green patches indicate the two target classes, and the green dashed line indicates the classification boundary. $\langle\theta\rangle$ indicates the highest probability stimulus. Right: discriminability of stimulus classes in the network output, measured by the sensitivity index (d'), shown before learning, after 10k time units, and after 250k time units.

but not pathological ($\sigma = 0.2$), and calibrate the hyperparameters that control the metabolic constraints (strength of regularization) to a level where they start to interfere with network performance. As learning progresses, our derived local plasticity rules quickly converge to a good solution, for both estimation and categorization (Fig. 3.1 b).

The emerging representations are noticeably different for the two tasks (compare Fig. 3.1c and d): for estimation, the distribution of preferred orientations is concentrated in the highly probable stimulus region, whereas the preferred distribution is bimodal in the case of classification. The average population activity for any stimulus provides additional quantification of the way in which learning allocates network resources (here, neural activity) to different stimuli (light colors in Fig. 3.1e and f). For the representation task, this metric confirms that neural resources are preferentially allocated to commonly occurring stimuli (Fig. 3.4a, Fig. 3.1c; the most likely stimulus is 0). Hence learning has converged to an efficient representation of stimulus statistics. Moreover, the average population activity encodes the prior probability of input stimuli (Fig. 3.1e).

The picture looks very different in the case of categorization: improbable stimuli ($\theta = \pm\pi$) still have a small contribution to the emerging neural representation, but so do the most probable stimuli (which in our example are centered on the categorization boundary). This is reflected in the distribution of preferred stimuli: the neurons distribute their tuning functions on either side of the most probable stimulus so that the most sensitive part of their tuning function lies on the decision boundary (Fig. 3.4b), and their peak responses are tuned for class identity (Fig. 3.1d). Overall, the emerging representations reflect both input statistics and task constraints.

So far all results used a particular choice of prior, but the same intuitions hold under prior variations. In the estimation task, training a new network with a tightened prior leads to a corresponding tightening of the population tuning (Fig. 3.1e, Fig. 3.4c). Firing rates decrease for peripheral stimuli; under synaptic weight regularization, the network will reduce firing rates for less probable stimuli, as they have little impact on the average error. This firing rate reduction is coupled with a concomitant increase in error for less probable stimuli (Fig. 3.4d). For the

classification task, we trained a new network with a shifted input distribution (Fig. 3.4e) as a way to segregate the effects of the decision boundaries from those of the prior distribution Fig. 3.1f). This break in symmetry leads to asymmetric errors, with larger errors for the less probable class (Fig. 3.4f), though the effect is more subtle than that observed when tightening the input distribution. The corresponding network representation also becomes asymmetric, with higher firing rates concentrating on the side of the more probable stimulus. Thus, even under transformations of our original prior, trained neurons show higher firing rates for both more probable, and more task-relevant features of the input, such that performance is consistently better for frequent stimuli.

How do these task-specific changes in representation manifest themselves in the decoded outputs? First, in the case of estimation, we can probe the distribution of outputs of the linear readout (Fig. 3.2a, gold ellipses) for a set of test stimuli (Fig. 3.2a, black crosses).[1] We found that responses are systematically biased for less probable stimuli, whereas the bias is negligible for frequently occurring stimuli. The effect on variance is much weaker (Fig. 3.2b). Second, for classification, we need to measure decoder output variability as a function of the prior stimulus probability. We use the fact that the classification boundary intersects the circle on which the input stimuli lie for the most probable and the least probable stimuli under the prior (Fig. 3.2c, inset). We compare the degree of output overlap for two test stimuli equally spaced on the two sides of the decision boundary at these two extremes. We find substantially higher discriminability for the high probability stimuli, relative to the low probability ones. As for estimation, this difference is due primarily to a better separability of the two means rather than a difference in variance. In summary, the network exhibits better performance for probable stimuli across tasks. Given limited resources, input statistics dictate not only the precision of the representation, but also task performance.

[1] Plasticity was disabled for test stimuli here, to isolate effects of internal noise without the confound of changes in network parameters.

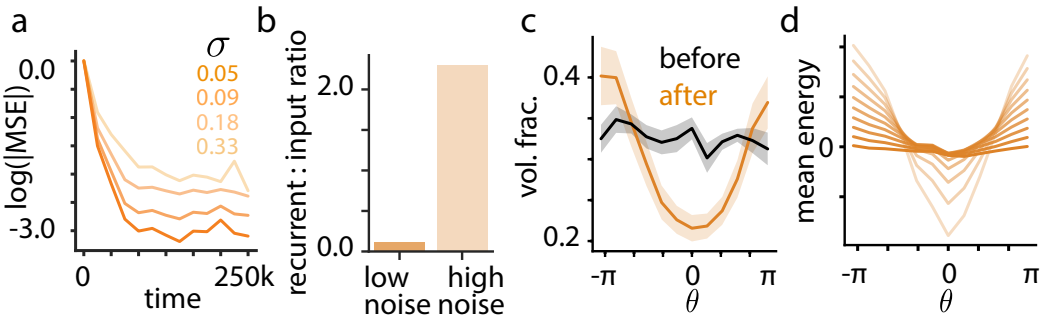

Figure 3.3: The effects of internal noise. a. Estimation performance during learning for different magnitudes of noise σ (for reference, the average external drive to the neuron is roughly twice the size of the largest noise level). **b.** Ratio of recurrent to input current for neurons in a low vs. high noise regime. **c.** Volume fraction of noise within/outside the readout manifold, as a function of stimulus angle, before and after learning ($\sigma = 0.33$). Shaded intervals are ± 1 s.e.m. across 20 simulations. **d.** Mean energy of $p(\mathbf{r}|\mathbf{s})$ as a function of input angle and different magnitudes of noise σ.

Next, we investigate the dynamics of learning and associated network representations, focusing on the classification task. We measure output discriminability at different stages of learning, with sensitivity index d' values estimated averaging over the stimulus distribution. At the outset, the output distributions for the two stimulus categories are indistinguishable (Fig. 3.2d). As learning progresses, the means of the two distributions segregate, while their variance remains approximately the same; this result prompts a more detailed examination of the degree to which the network is able to compensate for its own intrinsic noise.

Intrinsic noise in the recurrent dynamics is a key component of our solution, because the learning rule changes synaptic strengths based on whether fluctuations in the synapses' pre- and post-synaptic activities are associated with an increase or decrease in performance (Eq. 3.10). However, adding noise to the network can also make estimation more difficult: we found that increasing intrinsic noise leads to both slower learning and worse asymptotic performance (Fig. 3.3a). Given that output variance changes little across stimuli and over learning, is noise strictly deleterious for the network, and how can the network learn to counteract its effects? Interestingly, the more intrinsic noise we add in the network, the more the network relies on recurrent connectivity for its representation. Increases in noise cause an increased engagement of recurrent connectivity

after learning (Fig. 3.3b). The fact that recurrent connectivity is not strictly needed in absence of noise is intuitive, given that our chosen tasks involve a simple, instantaneous map of inputs to outputs (although note that in our framework some noise is needed to enable synaptic plasticity to occur). Though it is clear that recurrent connections alleviate some of the negative consequences of noise, their exact mechanism of action requires more investigation.

It seems like the overall level of noise in the network does not change dramatically during learning, at least not as reflected in output fluctuations. Yet, performance is systematically better for probable stimuli, indicating the possibility of noise compensation. To investigate whether (and if so, how) recurrent connections shape internal noise on the estimation task, we asked what fraction of the internal noise lies in the decoding manifold given by $\mathbf{D}$ and if it depends on the stimulus. Since the entropy of the network response distribution conditioned on the stimulus and its marginals are not analytically tractable, we resorted to numerical approximations: we used the network dynamics (with frozen weights) to sample from this conditional distribution, projected these responses a) in the readout manifold and b) in the 2D manifold of maximum variance, as defined by the first two PCA axes of the neural responses. Our final metric, which we refer to as the noise volume fraction, is computed as the ratio between the estimated noise in each of the two manifolds (using the determinant of the empirical covariance of the projected responses as a proxy for noise magnitude). The noise volume fraction is defined as follows:

$$VF(\mathbf{s}) = \frac{\det \mathbf{C}_D}{\det \mathbf{C}_{\text{PCA}}}, \tag{3.13}$$

where $\mathbf{C}_D$ is the covariance matrix of $\mathbf{r}$ projected onto the two output dimensions, and $\mathbf{C}_{\text{PCA}}$ is the covariance matrix of $\mathbf{r}$ projected onto the first two principal components of the neural activity for a fixed stimulus $\mathbf{s}$. This metric is 1 when the primary axes of internal noise lie in the decoding manifold; it is 0 when the two spaces are orthogonal, such that internal noise does not affect network outputs. After learning, the volume fraction is much smaller for probable

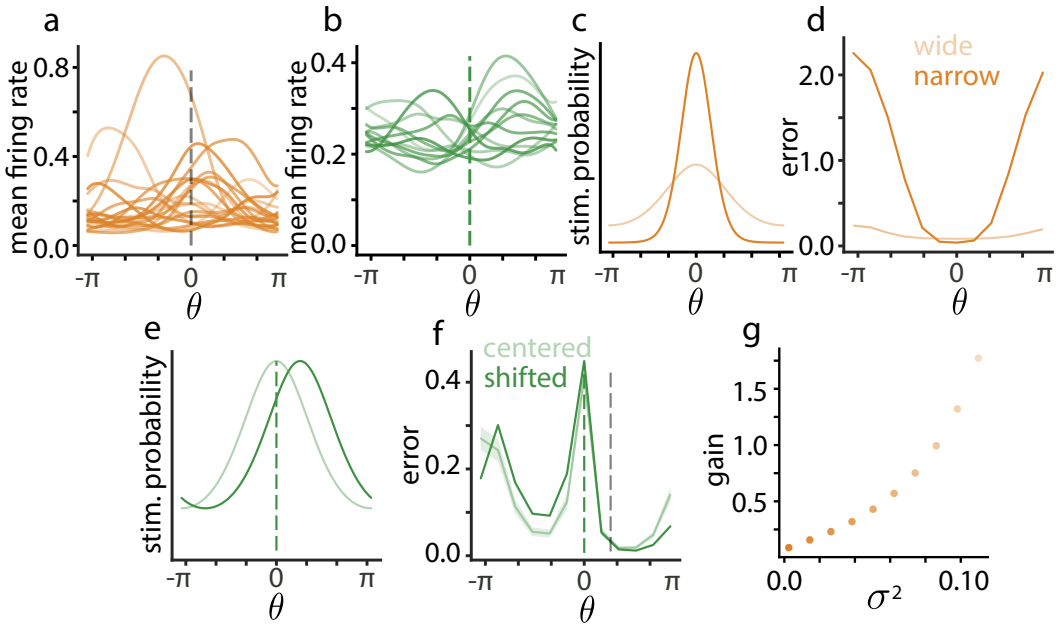

Figure 3.4: Manipulations of the input distribution. a. Sample tuning curves for the representation network. The gray dashed line indicates the prior mode. **b.** Sample tuning curves for the classification network. The green dashed line indicates the prior mode and the classification boundary. **c.** Two different input priors: the original distribution in light gold, and a sharper prior in dark gold. **d.** Error as a function of test input angle for a network trained on the original distribution (light gold), and the tightened distribution (dark gold). **e.** Schematic showing shifting the input distribution for the classification task. The original distribution is shown in light green, and the shifted distribution is shown in dark green; the green dashed line indicates the classification boundary. **f.** Error as a function of test input angle for a network trained on the original distribution (light green), and the shifted distribution (dark green). The gray dashed line indicates the shifted input distribution mode. Error bars for the original distribution indicate ± 1 s.e.m. across 20 simulations. **g.** The gain (L2 norm) of the mean energy as a function of σ^2.

stimuli (Fig. 3.3c), indicating that the network has learned to effectively 'hide' more of its noise for frequent inputs. This is not to say that output variance is lower for more probable stimuli: as we have already seen, if anything, output variance increases slightly for more probable stimuli. In general, the variability of the network output increases with the firing rates of its neurons, such that high network activity *necessarily* produces increased variability (one cannot amplify the signal without amplifying the noise to some degree). However, when normalizing for this increase, the network projects a smaller fraction of its total noise onto the decoder for more probable stimuli than it does for less probable stimuli.

An alternative way to think about the effects of intrinsic noise on the network activity is in terms of the energy function (Eq. 3.3), and corresponding steady-state stimulus response distribution (Eq. 3.4). Here, the noise variance acts as a temperature: the energy landscape flattens with increasing noise. Formally, one way to compensate for a increases in σ is to rescale the the network's stimulus-conditioned energy in proportion to the increase in σ^2, thus preserving the stimulus response distribution $p(\mathbf{r}|\mathbf{s})$. The network does employ this kind of compensation (Fig. 3.3d). As the intrinsic noise increases, the network boosts the gain of its energy function, with the mean energy increasing monotonically as a function of σ^2 (Fig. 3.4g). Hence, the network learns to compensate for its intrinsic noise so as maintain a good signal to noise ratio.

3.3 Discussion

Despite recent successes in the supervised training of recurrent networks [Pearlmutter 1989; Sussillo and Abbott 2009; Martens and Sutskever 2011; Marschall et al. 2019], it remains a mystery how biological circuits are capable of robust, flexible local learning in the face of constraints such as internal noise and limited metabolic resources. Here we have derived a class of synaptic learning rules that optimize a task-specific objective by combining local information about the firing rates of each pair of connected neurons with a single performance signal, implementable by global neuromodulation. Online learning naturally follows, since the network dynamics sample from a well-defined stimulus response distribution.[2] We further show that the derived learning rules lead to emerging neural representations that allocate neural resources and reshape internal noise to reflect both input statistics and task constraints.

The use of stochasticity as a means of estimating gradients has an extensive history [Williams 1992] and has been used to account for biological phenomena, in particular the role of variability of neurons in the songbird HVC nucleus in song learning [Fiete et al. 2007]. Our model is conceptually

[2]Note that although we are using sampling dynamics as well, our approach is different from sampling theory [Fiser et al. 2010b] in that here the sampling dynamics do not represent a Bayesian posterior.

similar, and can be thought of as a mathematically tractable instantiation of the REINFORCE framework [Williams 1992], where stochastic network responses are given by a continuous variant of the Boltzmann machine [Ackley et al. 1985]. Our model notably lacks an eligibility trace, which in previous models was required to integrate coactivity through time at synapses; it also stores an averaged measure of coactivity, $\langle r_i r_j \rangle_{p(\mathbf{r}|\mathbf{s})}$, rather than only averaging over post-synaptic activity– both of these differences are potentially experimentally testable. Further, existing literature has focused exclusively on the role of intrinsic noise on learning dynamics, ignoring its interactions with the circuit function and the emerging neural representations that support it. Here we show that there is a conflict between the positive role of stochasticity on learning and its deleterious effects on encoding. Over the course of learning, the network converges to a solution that trades off between the two, by appropriately reshaping the noise so as to increase the signal-to-noise ratio of the output.

Prior statistics play a key role in the emerging representations, with more neurons tuned to commonly occurring stimuli, and overall population activity weaker for infrequent stimuli. The inhomogeneous distribution of tuning functions aligns with optimal encoding of priors, as derived for abstract encoding models [Ganguli and Simoncelli 2014]. But unlike previous models, our network also encodes the prior as a population tuning function, a discrepancy that most likely reflects differences in the exact form of the regularizer enforcing metabolic constraints. Different choices of regularizer, in particular a sparseness-encouraging constraint on neural activity, would likely lead to representations more similar to traditional efficient coding models [Olshausen et al. 1996]. Nonetheless, our approach provides explicit local learning dynamics for these abstract models and, importantly, is successful in regimes where analytic methods are intractable, e.g. for multivariate stimuli, and for tasks that go beyond simple stimulus reproduction.

One limitation of our formulation is that reward does not depend on the history of network responses. This differentiates our approach from traditional models of reward-modulated learning, which focus on solving temporal credit assignment, especially in spiking circuits [Frémaux and

Gerstner 2016; Fiete et al. 2007; Miconi 2017; Hoerzer et al. 2014]. Despite this limitation, our mathematical derivation extends previous work by making explicit the four-way interactions between intrinsic noise, metabolic constraints, input statistics, and task structure in the circuit. Preliminary results suggest that it is possible to extend the current framework to incorporate temporal dependencies in both stimuli and reward structure, better aligning it with traditional goals of reinforcement learning.

It has been argued that learning algorithms based on stochastic gradient estimates cannot match the learning capabilities of the brain [Payeur et al. 2020], as they perform poorly in high dimensions [Werfel et al. 2004]. This has led to renewed focus on alternatives that rely on the neural system having access to a closed-form expression for the gradient (at least approximately), in particular biologically-plausible approximations to backpropagation [Lillicrap et al. 2016; Payeur et al. 2020; Marschall et al. 2019]. Still, these models can't be the whole story. While gradient information might be available for unsupervised [Pehlevan et al. 2015] or intrinsic learning objectives [Goroshin et al. 2015], this is certainly not true for external rewards, when the loss function is specified by the environment itself. Animals rarely, if ever, have access to explicit reward functions. Moreover, neither unsupervised learning nor backpropagation can account for the critical role of neuromodulation in synaptic plasticity and its contribution to perceptual learning [Bakin and Weinberger 1996; Kilgard and Merzenich 1998; Froemke et al. 2007]. The complementary nature of the two classes of learning rules suggests that they might *both* play an important role in biological learning. Bringing the two closer together is a promising direction for future research, both theoretical and experimental.

4 Explaining neural and behavioral variability in mice with a model of context-specific auditory perceptual learning

As demonstrated in the previous chapter, given limited metabolic resources, optimal sensory systems should develop representations that emphasize task-relevant and frequently occurring stimuli. However, in the auditory system, frequency information that matters for one task—e.g. localizing mouse pup calls [Sewell 1970; Marlin et al. 2015]—may not be very important for another task, such as distinguishing self-generated from environmentally-generated sounds [Rummell et al. 2016; Singla et al. 2017; Schneider et al. 2018]. In mouse auditory cortex, numerous studies have demonstrated that training animals on perceptual tasks can reorganize the tuning properties of neurons to emphasize task-relevant frequencies [Polley et al. 2006; David et al. 2012; Fritz et al. 2003; Recanzone et al. 1993]. Furthermore, stimulation of neuromodulatory centers of the brain to deliver neurmodulators to auditory cortex can rescue the plastic capabilities of the system beyond its critical period of plasticity. These studies have been performed for a variety of neuromodulators, including oxytocin [Marlin et al. 2015], acetylcholine [Kilgard and Merzenich 1998; Bakin and Weinberger 1996], and norepinephrine [Glennon et al. 2019], and collectively suggest that mouse

auditory cortex is adaptable in later life, but possibly only under task-specific conditions. But how can mouse auditory cortex act like a flexible, task-specific representational system without sacrificing its ability to represent the full breadth of acoustic experience? When auditory cortex adapts to task requirements, does it overwrite its previous representation? Furthermore, does the animal's initial representation affect its speed of perceptual learning, or explain variability across learned representations?

We address these questions in this study through a combination of modeling, behavioral experiments, and longitudinal neural recordings of mice performing auditory perceptual learning. We observe strong reorganization of neural responses over the course of learning, and also observe a striking change in neural responses in behavioral contexts compared to passive contexts. This suggests a mechanism whereby a mouse can transiently reorganize its sensory cortex during a particular task without sacrificing the system's general-purpose representational capabilities. Inspired by previous studies which demonstrate that acetylcholine delivery from the nucleus basalis can induce both synaptic plasticity [Reed et al. 2011; Bakin and Weinberger 1996; Froemke et al. 2007, 2013; Kilgard and Merzenich 1998] and transient context-specific reorganization of the circuit [Kuchibhotla et al. 2017], we construct a neural network model in which acetylcholine mediates both reward-based synaptic plasticity and attentional reorganization of the circuit. Our model is able to capture the variability we observe in both mouse behavior and neural responses over the course of learning and across contexts. Furthermore, it is able to provide predictions as to how properties of an animal's representation early in learning (in particular input discriminability and animal's choice bias) can shape an animal's final, learned representation.

4.1 Results

In order to study perceptual learning in auditory cortex, we presented head-fixed mice with a sequence of trials in which one of several possible frequencies was played (Fig. 4.1a), one of

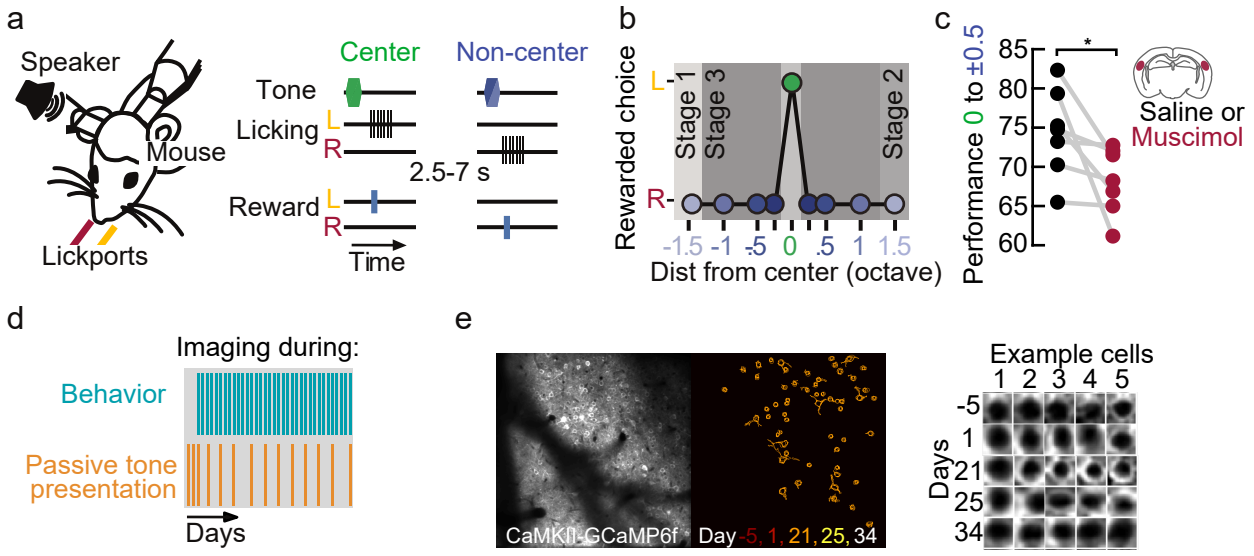

Figure 4.1: Experimental setup. a. Experiment and task setup. Head-fixed mice are presented with tones, and are trained to lick left for center tones, and right for non-center tones. **b.** Range of auditory stimuli introduced over three stages of training. Each stage progressively introduces more non-center tones. **c.** Change in percentage correct at the center tone and four closest non-center tones with muscimol injection in the recorded region (red) compared to saline controls (black). **d.** Passive tone presentation sessions were interleaved with rewarded behavior sessions over the course of training. **e.** Left: example recording window. Center: labeled regions of interest. Right: labeled cells imaged over days. Experiments were performed by Kathleen Martin.

which (the center tone) indicated that the animal should lick left, while all others indicated a right lick. Only licks in the correct direction produced a water reward. To acclimatize the mice to the complexity of the task, we introduced additional tones in three stages (Fig. 4.1b), beginning with a single, distant non-center tone and progressively adding intermediate tones which increased the difficulty of the task. We injected either saline or muscimol into primary auditory cortex while the animals performed the task, and verified that performance did drop significantly for the muscimol injection (Fig. 4.1c), indicating that neuron responses in primary auditory cortex play an important role in the animals' decision making. To understand how the response properties change over time to facilitate decision making, we longitudinally performed two-photon calcium imaging over the entire course of learning, both during behavior and while animals passively listened to tones (Fig. 4.1d-e). Using this data, we then characterized the behavioral and neural

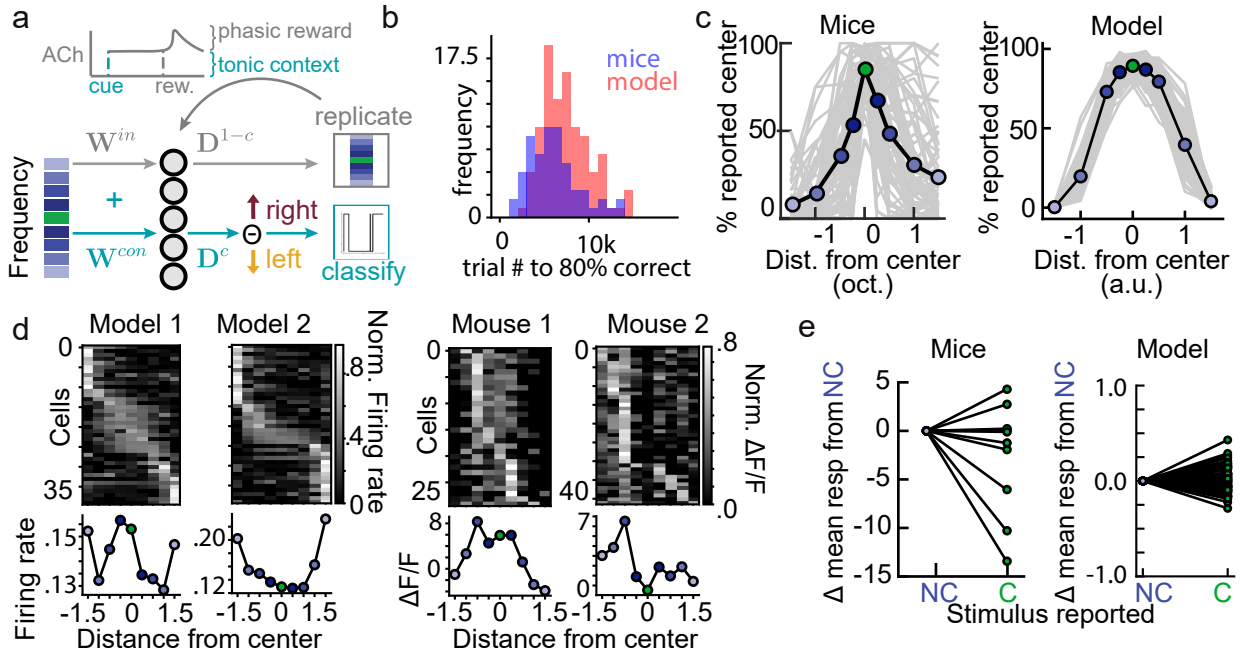

Figure 4.2: Modeling neural responses with reward-based learning. a. Model schematic. The model network receives context-dependent and context-independent information through $\mathbf{W}^{con}$ and $\mathbf{W}^{in}$, respectively. The context-dependent pathway is gated by tonic acetylcholine levels, and synapses are modified by phasic reward signals. During behavior the network is trained to classify its inputs as center versus non-center, whereas in the passive context the network is trained to replicate its inputs. **b.** Comparison between the time to reach 80% correct in Stage 1 for different mice (left), and for different model initializations (right). **c.** Example psychometric curves (grey) and the mean (black) across different mice (left) and model initializations (right). **d.** Two example sets of model tuning curves (left) compared to two example mouse tuning curve ensembles (right). Each row corresponds to a normalized tuning curve for one cell. Mean firing rates across neurons for the models and mean $\Delta F/F$ for mice are provided on the bottom. **e.** Difference in mean mouse (left) and model (right) neural responses for center tones compared to non-center tones.

diversity of responses across animals, and constructed a model that captured and explained these features.

Our model (Fig. 4.2a) is designed to capture qualitative features of learning dynamics, behavior, and context-specific responses in learning animals. The model receives feedforward frequency information through two sources: one that is a consistently present thalamic input, and one that is selectively activated during cued behavioral contexts (Section 4.2.3.2). In keeping with previous experimental results [Guo et al. 2019; Kuchibhotla et al. 2017; Takesian et al. 2018], we postulate

that this contextual gating is mediated by a tonic acetylcholine signal from the nucleus basalis that turns on when the animal is cued for the task and activates interneurons, which in turn have both inhibitory and disinhibitory effects on excitatory neurons in auditory cortex. In addition, inspired by the role nucleus basalis acetylcholine plays in inducing plasticity in the auditory cortex [Reed et al. 2011; Bakin and Weinberger 1996; Froemke et al. 2007, 2013; Kilgard and Merzenich 1998] and the fact that nucleus basalis acetylcholine neurons show phasic responses to reward [Guo et al. 2019; Hangya et al. 2015; Laszlovszky et al. 2020], we introduced an additive, phasic reward response to our model acetylcholine feedback, which induced synaptic modifications through a form of reward-modulated Hebbian plasticity (Section 4.2.3.3).

An ensemble of trained model networks show roughly the same learning profiles as mice, taking similar numbers of trials to reach proficiency in Stage 1 (Fig. 4.2b), and exhibiting similar variability. This is an important proof of principle, because the reward-modulated Hebbian plasticity based on the REINFORCE algorithm is well known to scale poorly and require slow learning rates [Werfel et al. 2003]: it is conceivable that our chosen simple form of learning could have been fundamentally unable to learn as fast as animals. After training, model networks show similar psychometric response profiles to mice on average, but with somewhat reduced variability across different simulations (Fig. 4.2c). Furthermore, mice exhibited better performance for intermediate foil tones, which was likely limited in our model by the separability of input tones (determined by σ^2_{freq}; see Section 4.2 for details). Both mice and models also exhibited remarkable diversity at the level of neural ensembles: the distribution of neurons tuned to the center frequency versus flanking frequencies varied considerably, and the mean tuning curves across neurons showed highly variable profiles, with some tuning curves showing peaked firing rate responses at center frequencies, while others showed greater firing rates for non-center frequencies (Fig. 4.2d-e). These results collectively demonstrate that our model is able to qualitatively capture diversity in both behavioral and neural responses in mice.

Having demonstrated the efficacy of our model for capturing behavioral response properties

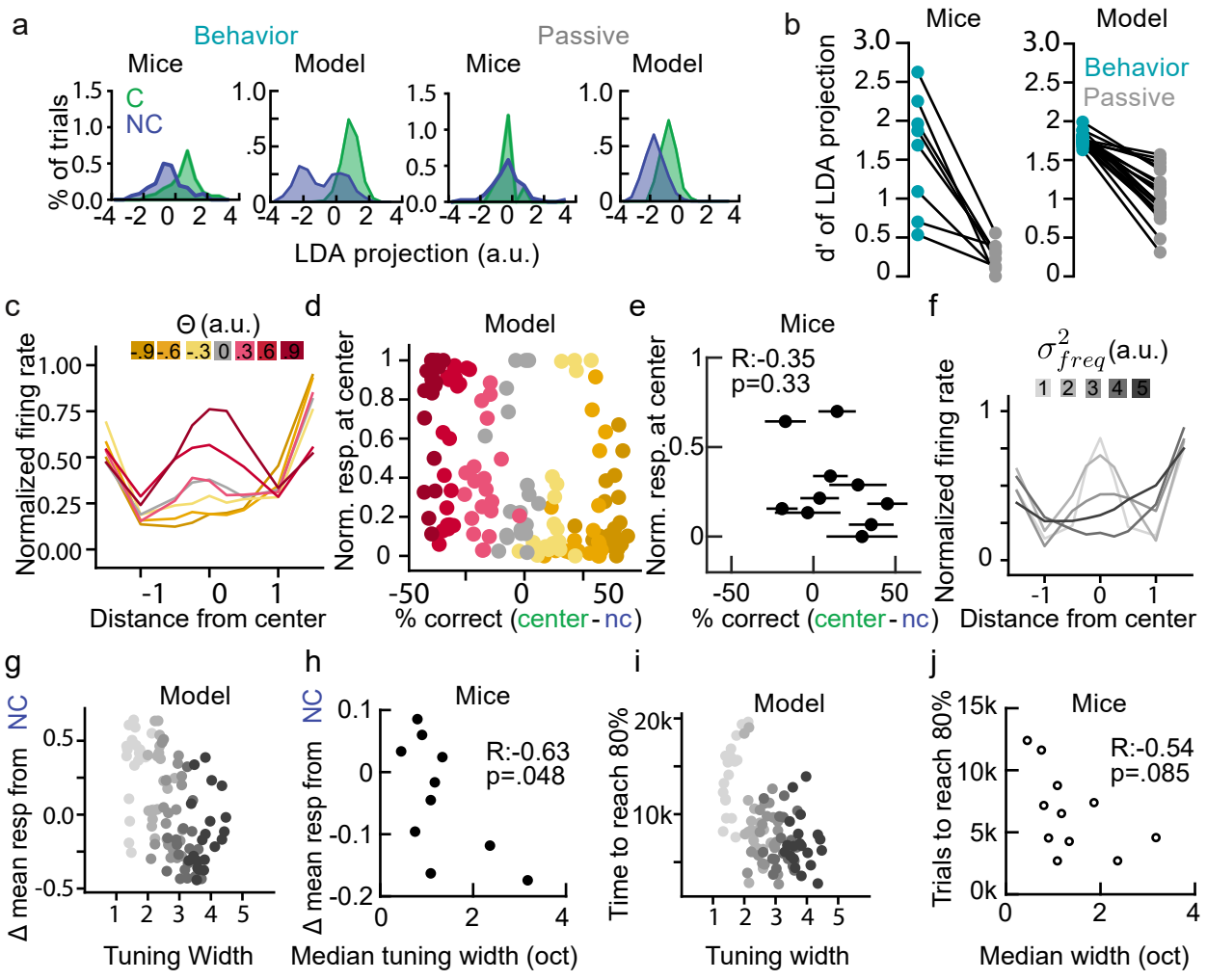

Figure 4.3: Effects of context and initial conditions. a. Center (green) and non-center (blue) mouse and model responses during behavior (left) and passive (right) sessions, projected onto the optimal linear discriminant axis calculated during behavior. **b.** Discriminability (d') of the distribution of center and non-center neural responses in behavior (blue) sessions and passive (grey) sessions, projected onto the linear discriminant axis calculated during behavior for mice (left) and for different model simulations (right). **c.** Model mean tuning curves for models initialized at different threshold Θ values. **d.** Relationship between the difference between % correct for center and non-center tones and network responses to the center frequency for models initialized at different threshold Θ values. **e.** Same as **d**, but for mouse data. **f.** Same as **c**, but as a function of the tuning width of cells in the input layer σ^2_{freq}. **g.** Difference in mean response for center tones compared to non-center tones for models as a function of their initial tuning width. **h.** Same as **g**, but for mouse data. **i.** Time to reach 80% correct during Stage 1 for models as a function of their initial tuning width. **j.** Same as **i**, but for mouse data.

and learning in animals, we next interrogated it to see if it can help eliminate hypotheses about how learning affects context-dependent responses. One hallmark of our data is that neuron responses

reorganize considerably across contexts: center and non-center responses projected onto a linear discriminant axis calculated during the behavioral context exhibit much lower discriminability during the passive context (Fig. 4.3a-b), a phenomenon also shown across different trained mice. To capture this phenomenon, we considered three possible alternative models: 1) only the separate context-dependent pathway is learned over training; 2) both the context-dependent input stream and passive feedforward inputs are modified by rewarded feedback; or 3) neurons receive no context-dependent feedback, only learned feedforward input. Only the first proposed model captured the qualitative decrease in discriminability in the passive context relative to the behavioral context observed along the linear discriminant axis (Fig. 4.3a-b), and so we used this model for all other simulations. In contrast, when learning occurs at all synapses (context invariant learning), or when there is no separate context-activated input stream (context invariant responses), discriminability is not reduced across contexts (Fig. 4.4a-f). Each of these proposed models has different implications for how a primary sensory region should adapt to reward. In particular, the first proposed model, which restricts learning to the context-dependent pathway, allows for a system that flexibly adapts to preferentially represent task-relevant stimuli, but does not sacrifice its representation of task-irrelevant stimuli in passive listening contexts. It is important to note that center versus non-center discriminability is not lost in the passive context, only reorganized: if the linear discriminant axis is calculated during the passive context, rather than the active context, center and non-center tones are still discriminable in mice and across all models (Fig. 4.4e).

We next asked whether we could use our model to identify key features of the initial network state that explain variability in the animals' learned representation. In particular, we investigated which features of the networks' initial state could contribute to their eventually learning a representation with peaked mean responses at center versus non-center frequencies. We found that the model's initial decision threshold Θ strongly modulated mean tuning in trained networks, with negative thresholds producing peak responses at non-center frequencies and positive thresholds

producing peak responses at center frequencies (Fig. 4.3c-d). However, in mice we found only a weak, non-significant correlation ($p = 0.33$) between animals' initial bias and the neural response at the center frequency (Fig. 4.3e). This discrepancy is possibly due to active shaping of stimulus statistics mice receive during learning, tailored specifically to reduce biases in their responses—if this were true, we would expect animals trained without shaping to exhibit a much stronger trend. Interestingly, we found that the tuning width of input layer neurons (parameterized by σ^2_{freq} in Section 4.2.3.1) had a similar effect on the mean response profiles of neurons, with small tuning widths increasing the normalized neuron responses to center tones (Fig. 4.3f-g), a trend that also held in mice (Fig. 4.3h, $p = 0.048$). Wider input tuning widths also affected the tuning widths of neurons in the recorded population. We found that in both model and mice, wider tuning curves early in training tended to produce *faster* learning, though the correlation in mice was non-significant (Fig. 4.3i-j, $p = 0.085$). Our model therefore suggests that both animals' choice threshold and the discriminability of input tuning curves both play a role in determining whether neurons tend to show strong responses at target frequencies or at flanking frequencies: a model's resultant representation is a combination of both its initial state and the learning process.

4.2 Methods

4.2.1 Animals

All procedures were approved under an NYU Langone Institutional Animal Care and Use Committee protocol. Male and female mice aged 6-20 weeks old were used in all experiments. Genotypes used were wild-type C57BL/6J (The Jackson Laboratory, Stock No: 000664), Gad-Cre (The Jackson Laboratory, Stock No: 028867), and Ai9 (The Jackson Laboratory, Stock No: 007909). All mice had a C57BL/6J background. Mice were housed in a temperature and humidity controlled room maintained on a 12 hour light/dark cycle. Animals used in behavior were given 1 mL water/day.

If their weight dropped below 80% of original, they were given *ad libitum* water until weight returned to ≥80% original value.

4.2.2 2AFC behavioral training

Behavioral events (lick detection, auditory stimulus delivery, water reward delivery) were monitored and controlled by custom MATLAB programs interfacing with an RZ6 auditory digital signal processor (Tucker-Davis Technologies) via RPvdsEx software (Tucker-Davis Technologies). Licks were detected using capacitance sensors (SparkFun, Part number: AT42QT1011) and water was delivered using solenoids (The Lee Company, Part number: LHDA0581215H). Animals were restrained using custom headposts (Ponoko).

Behavioral training on the auditory 2AFC task began after 7+ days of water restriction. Training started with habituation to head-fixation with water delivered to the mouse while it sat in a plexiglass tube. This was followed by lick port sampling sessions, in which the animal could receive water by alternating licking between the two ports with a minimum of 3 seconds between possible rewards. Mice typically learned to alternate ports while licking for 2-4 μL water droplets in 2-4 sessions. Once animals reliably licked to receive water from lick ports, stage 1 training was begun (i.e., animals were trained to lick left for the center frequency and lick right for one non-center frequency). The center frequency was chosen to be either 11.3, 13.4, or 16 kHz (each animal had a single consistent center frequency pseudo-randomly selected from those three values). Non-center frequencies were set per animal to be ±0.25, ±0.5, ±1.0, and ±1.5 octaves from the selected center frequency. In stage 1, the only non-center frequency was either +1.5 octaves or -1.5 octaves from center (and whether higher or lower frequency was also pseudo-randomly assigned per animal).

In stage 1, while an animal's performance remained <80% correct, they were rewarded with water regardless of behavior choice on 15% of trials to help promote consistent licking during training. Once performance reached ≥80% correct for three consecutive days in stage 1, animals

moved to stage 2 in which the other non-center frequency (either ± 1.5 octaves away) was added. After three days in stage 2, animals moved to stage 3 regardless of performance (in which all other non-center stimuli ±0.25, ±0.5, and ±1.0, octaves from the center frequency were also presented and rewarded for right-side licking).

On each trial, a 250 ms tone was presented and animals had to classify the tone as the center frequency (green) or any other frequency (shades of blue). Stimuli were presented at 70 dB SPL in a pseudorandom order, such that the likelihood of center:non-center was 1:1 (with frequency uniformly chosen from the non-center distribution on non-center trials). After a 250 ms delay, animals had to lick left to report the stimulus as 'center' and had to lick right to report the stimulus as 'non-center'. If the animal did not respond during the 2.5 seconds of the response epoch, the trial was classified as a 'no response' trial (which were excluded from analysis except where otherwise noted). If the lick response was correct, a small water reward (2-4 μL) was delivered to the corresponding lick port. Inter-trial intervals were 3±0.5 seconds (mean±s.d.) on trials with a correct response and 6±0.5 seconds (mean±s.d.) on trials with an incorrect response or without a response. Animals were not punished for licking outside of the response epoch. Animals generally performed between 350-500 trials/day.

4.2.2.1 Two-photon calcium imaging

Cranial window implantation over left auditory cortex was performed, as previously described. For cell body imaging, 1.0-1.5 μL of diluted CaMKII.GCaMP6f (AAV9, diluted 1:3 with dPBS, Addgene number: 100834) or Syn.GCaMP7f (AAV1, diluted 1:4 with dPBS, Addgene number: 104488) was injected into auditory cortex (1.5 mm from lambda, along lateral suture).

Two-photon fluorescence of GCaMP6f/s and tdTomato was excited at 900 nm using a mode locked Ti:Sapphire laser (MaiTai, Spectra-Physics, Mountain View, CA) and detected in the green channel and red channel, respectively. Imaging was performed on a multiphoton imaging system (Moveable Objective Microscope, Sutter Instruments) equipped with a water immersion objective

(20X, NA=0.95, Olympus) and the emission path was shielded from external light contamination. Images were collected using ScanImage (Janelia). To image auditory cortex, the objective was tilted to an angle of 50–60°. We imaged 300 μm^2 areas in auditory cortex (scan rate either 4 Hz or 30 Hz, laser power ≤40 mW). For animals with TdTomato labeling in interneurons, we imaged both the green and red channel to visualize both the functional and structural markers, respectively.

The speaker was 10 cm away from the ear contralateral to the window. A consistent region of excitatory neurons in layer 2/3 of A1 (based on vasculature and relative orientation of neurons) was imaged over all days of pairing. For baseline imaging, pure tones (70 dB SPL, 4–64 kHz, 250 ms, 10 ms cosine on/off ramps, quarter-octave spacing, 10 trials for each frequency) were delivered in pseudo-random sequence every 5 s.

Excitatory neuron imaging data were motion-corrected and regions of interest (ROIs) were automatically detected using suite2p [Pachitariu et al. 2017]. ROIs were manually verified and additional ROIs were manually drawn on an average image of all motion-corrected images. Calcium fluorescence was extracted from all ROIs. Semi-automated data analysis was performed using custom Matlab (MathWorks) software. For each ROI, we corrected for potential neuropil contamination as previously described. The ΔF/F (%) was calculated as the average change in fluorescence during the stimulus epoch relative to the 750 ms immediately prior to stimulus onset: $\Delta F/F(\%) = ((F_t - F_0)/F_0) \times 100$. ROIs were included in additional analysis if they had a significant response (both $p < 0.05$ Student's two-tailed, paired t-test comparing activity during any stimulus and pre-stimulus epochs and had a mean ΔF/F equal to 5% or above for all trials with a particular frequency).

4.2.3 Model

4.2.3.1 Task structure

Neural networks received $N_s = 33$ frequency-tuned inputs $\mathbf{s}$, given by:

$$\mathbf{s}_i^{(k)} = B \exp \frac{-\left(f^{(k)} - \mu_i\right)^2}{2\sigma_{freq}^2}, \tag{4.1}$$

where μ_i determines the peak response of tuning curve i, σ_{freq}^2 determines the tuning width, $f^{(k)}$ is the input frequency for trial k, and B determines the maximum response magnitude. The μ_i were selected to evenly tile frequency space. Trials were organized into multiple phases: to model unsupervised learning during development, we initially trained the network on an ensemble of unit-spaced integer frequencies ranging from 8 to 24 (in arbitrary units) selected with uniform probability, and trained the network to accurately decode the input frequency. Networks received a reward proportionate to how accurately they were able to reconstruct the received stimulus:

$$R_{1-c}^{(k)} = -(\mathbf{s}^{(k)} - z_{1-c}^{(k)})^T(\mathbf{s}^{(k)} - z_{1-c}^{(k)}). \tag{4.2}$$

Here, the binary variable c indicates whether the network is in a passive, unsupervised context ($c = 0$) or in an active trained context ($c = 1$). $z_{1-c}^{(k)}$ gives the network's output in the passive context, while $z_c^{(k)}$ gives the output in the behavioral context (see Section 4.2.3.2).

Next, we trained the network on a target frequency detection task in a curriculum learning paradigm [Kepple et al. 2021]. In the first phase of learning, the networks were trained to discriminate a target tone ($f_{targ} = 16$), from a single foil tone (24). After training networks on this task for 20 thousand trials, we introduced a second phase, where the network was required to discriminate the target from an additional foil tone (8). This phase lasted an additional 25 thousand trials. The final phase (20 thousand trials) increased the complexity of the task by introducing

several foil tones closer to the target frequency (12, 14, 15, 17, 18, 20). Targets and foil frequencies were presented with equal probability across all phases of discrimination learning.

For learning, the networks received an extrinsic reward only if the sign of the network output matched whether or not the target frequency was present:

$$R_c^{(k)} = \begin{cases} 1 & \text{if } z_c^{(k)} < 0 \text{ and } \mathbf{s}^{(k)} \neq \mathbf{s}_{targ} \text{ or } z_c^{(k)} > 0 \text{ and } \mathbf{s}^{(k)} = \mathbf{s}_{targ} \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

In biological networks, neurons cannot increase their firing rates freely: extra spiking comes with additional metabolic costs [Simoncelli and Olshausen 2001]. To model these resource constraints, we also penalized for their firing rates. The total reward for the system then became:

$$R_{total}^{(k)} = (c)R_c^{(k)} + (1-c)R_{1-c}^{(k)} - \lambda_{l_1} \sum_{i=0}^{N} |\mathbf{r}_i^{(k)}|, \tag{4.4}$$

where $\lambda_{l_1} = 0.4$ is a small positive constant. To increase the efficiency of our algorithm [Williams 1992], we separately kept track of a reward baseline, which was subtracted out from the reward signal $R^{(k)}$ as it was delivered to the network each trial. This baseline is an approximation to the average expected reward, calculated by a moving average of past rewards with time constant α:

$$R_{avg}^{(k)} = (1-\alpha)R_{avg}^{(k-1)} + \alpha R_{total}^{(k)}. \tag{4.5}$$

4.2.3.2 Network Architecture

The model neural network receives stimulus-dependent input from both context-independent and context-dependent sources. Previous studies [Kuchibhotla et al. 2017] have shown that inhibitory microcircuits activated by acetylcholine signaling from the nucleus basalis mediate contextual changes in auditory signal processing. Because these microcircuits were shown to both inhibit and disinhibit excitatory neurons, we model their input as a separate, sign-unconstrained stimulus-

dependent current to the neurons in our network. We propose that tonic acetylcholine signals correspond to an indicator of context ($A_{tonic} = c$), and gate this input. Our neuron activities are given by:

$$\mathbf{h}^{(k)} = \mathbf{W}^{in}\mathbf{s}^{(k)} + A_{tonic}\mathbf{W}^{con}\mathbf{s}^{(k)} \tag{4.6}$$

$$\mathbf{r}^{(k)} = f(\mathbf{h}^{(k)}) + \sigma\boldsymbol{\eta}_r^{(k)}, \tag{4.7}$$

where $\mathbf{h}^{(k)}$ gives the neurons' input current on trial k, $\mathbf{W}^{in}$ is the $N_r \times N_s$ context-independent feedforward weight matrix, $\mathbf{W}^{con}$ is the $N_r \times N_s$ context-dependent weight matrix, $f(\mathbf{h}) = \beta \ln(1 + \exp(\gamma\mathbf{h}))$ is a pointwise differentiable rectified linear unit, $\boldsymbol{\eta}_r$ is independent additive Gaussian noise, σ controls the standard deviation of the neurons' variability, and $\mathbf{r}^{(k)}$ gives the resulting neural firing rates for trial k. We model neural decision variables z_c and z_{1-c} with separate decoders across contexts. During pretraining, when networks are trained to reproduce their inputs, z_{1-c} is given by:

$$z_{1-c}^{(k)} = \mathbf{D}^{1-c}\mathbf{r}^{(k)} + \sigma_{rep}\boldsymbol{\eta}_z^{(k)}, \tag{4.8}$$

where $z_{1-c}^{(k)}$ is an N_s-dimensional vector, and $\mathbf{D}^{1-c}$ is an $N_s \times N_r$ decoder matrix. For tone discrimination, $z_c^{(k)}$ is a scalar decision variable, given by:

$$z_c^{(k)} = \mathbf{D}^{c}\mathbf{r}^{(k)} + \sigma_{disc}\boldsymbol{\eta}_z^{(k)}, \tag{4.9}$$

where $\mathbf{D}^c$ is a $1 \times N_r$ decoder matrix.

4.2.3.3 Learning

We trained all free parameters in our neural networks using a reward-modulated Hebbian synaptic plasticity rule based on the REINFORCE algorithm (Appendix A.3; [Williams 1992]). Under this

plasticity rule, three factors (reward, pre- and post-synaptic activity) are combined multiplicatively to perform a stochastic approximation to gradient descent on the expected reward for a given trial. We propose that in addition to relaying context information through tonic activity, acetylcholine also relays reward information through phasic firing. For a given trial, at reward time, this gives:

$$A_{total}^{(k)} = \lambda_A A_{tonic} + R_{total}^{(k)}, \tag{4.10}$$

where λ_A is a positive constant. For the REINFORCE learning algorithm, the reward signal $A_{total}^{(k)}$ used for parameter updates can vary up to an additive constant without affecting parameter updates on average, though large constants can increase the variance of parameter updates [Williams 1992]: this is why the tonic acetylcholine signal $\lambda_A(c)$ does not prevent learning. With a trial-by-trial measure of deviations from expected reward $A_{total}^{(k)}$, plasticity for our input weights and decoder parameters were given as follows:

$$\Delta \mathbf{W}_{ij}^{in} = \lambda(1-c) A_{total}^{(k)} \frac{\mathbf{r}_i^{(k)} - f_i\left(\mathbf{h}^{(k)}\right)}{\sigma^2} \mathbf{s}_j^{(k)} \tag{4.11}$$

$$\Delta \mathbf{W}_{ij}^{con} = \lambda(c) A_{total}^{(k)} \frac{\mathbf{r}_i^{(k)} - f_i\left(\mathbf{h}^{(k)}\right)}{\sigma^2} \mathbf{s}_j^{(k)} \tag{4.12}$$

$$\Delta \mathbf{D}_{1j}^{c} = \lambda(c) A_{total}^{(k)} \frac{z_c^{(k)} - \mathbf{D}^c \mathbf{r}^{(k)}}{\sigma_{disc}^2} \mathbf{s}_j^{(k)} \tag{4.13}$$

$$\Delta \mathbf{D}_{ij}^{1-c} = \lambda(1-c) A_{total}^{(k)} \frac{z_{1-c}^{(k)} - \mathbf{D}^{1-c} \mathbf{r}^{(k)}}{\sigma_{rep}^2} \mathbf{s}_j^{(k)}. \tag{4.14}$$

Notice here that the primary feedforward weights $\mathbf{W}^{in}$ are only updated in the passive context (when $1-c=1$). This captures the intuition that the primary inputs to auditory cortex are difficult to modify after an early developmental critical period of plasticity [Zhang et al. 2001; Zhou et al. 2011; Insanally et al. 2009; de Villers-Sidani et al. 2008], and suggests that perceptual learning and its effects in our model system are purely task-dependent: in the absence of the context signal c, learning is non-existent. Furthermore, the decoders were only learned during the tasks for which

they were responsible. We explored two additional learning conditions (Fig. 4.4): in the context invariant learning condition we updated $\mathbf{W}^{in}$ in both contexts, while in the context invariant response condition, we set $\mathbf{W}^{con} = 0$ and made no parameter modifications to those weights.

4.3 Discussion

In both our model and the experimental data, neural response properties after training are partially shaped by the training procedure itself, and partially by initial behavioral and neural response properties. We found that reward signals considerably reshape neurons' tuning properties, especially in the behavioral context across both our model and neural recordings. Furthermore, we found evidence for remarkable reorganization of neural responses across contexts: the principal axis along which center and non-center tones are discriminable in behavioral contexts loses the majority of its discriminability in passive contexts. We propose that acetylcholine signals from the nucleus basalis could support both context-dependent reorganization in auditory cortex *and* reward-based learning, given previous evidence for its involvement in both functions [Kuchibhotla et al. 2017; Froemke et al. 2007, 2013; Guo et al. 2019; Laszlovszky et al. 2020]. Interestingly, in our model, context information is relayed through tonic acetylcholine levels, which rise at the beginning of a trial, and reward information is relayed through phasic responses. This separation is possible because neural activity cannot be correlated with experimenter-induced changes in context, and so on average will not affect the correlation-based reward-modulated Hebbian plasticity rule we use in our model [Williams 1992].

While this joint model of attentional and reward modulation agrees with both our experimental data and previous studies, it is only partially constrained by what we currently know about learning in the auditory cortex. The simplicity of our model is a considerable advantage, but there exist many more complicated alternative learning schemes which would require further experimentation to distinguish from our approach. Of these, two axes are particularly important, each addressing the

exact nature of the nucleus basalis' role in learning. First, we have treated the reward signal as a scalar, delivered universally to all synapses. However, this method is well known to be inefficient for more complex tasks [Werfel et al. 2003]. Given that acetylcholine projections are known to exhibit regionally specific connectivity and support a variety functions from attention to learning [Záborszky et al. 2018], it may be that their responses become more heterogeneous and targeted for more complex tasks, acting as a form of spatial credit assignment [Roelfsema and Ooyen 2005; Roelfsema et al. 2010]. Second, the nucleus basalis shows some signs of being adaptive over the course of training on perceptual learning tasks [Guo et al. 2019; Laszlovszky et al. 2020]. What we have modeled here as a static system may be more advanced, for instance by actively tagging and modifying neurons whose responses are informative for the task [Haimerl et al. 2019, 2021], or by developing representations of expected future reward through temporal difference learning, as with dopamine neurons in the ventral tegmental area [Schultz et al. 1997]. Such properties could allow the auditory cortex to respond differently to multiple different contexts, and to modify synapses based on rewards that are significantly delayed in time. However, little is known about the properties of nucleus basalis neurons, and most auditory perceptual learning paradigms are not complex enough to differentiate these more advanced learning schemes from the one we have employed here.

Despite these complexities, our data and model place constraints on how learning is occurring in the system. We found that only models in which rewards modify exclusively context-specific parameters are able to capture the strong loss of center versus non-center discriminability in the passive context along the linear discriminant axis calculated during behavior. Models in which all synapses were modified by reward signals and models in which there were no context-specific inputs failed to capture this result. These results suggest that the auditory cortex is able to rapidly rearrange itself through a context-dependent mechanism during behavior, and that it is primarily this mechanism that adapts to reward. Previous experimental results [Kuchibhotla et al. 2017; Takesian et al. 2018] would suggest that inhibitory and disinhibitory microcircuits gated

by acetylcholine signaling underly these context-specific changes in response properties, but it remains for future work to verify our model's prediction that it will be synapses within these circuits that are modified by reward, and not feedforward excitatory inputs. The idea that only context-gated synapses in auditory cortex are modified through plasticity in adult mice could explain why environmental statistics have a much-reduced influence on tuning properties of neurons beyond a brief postnatal critical period of plasticity [Zhang et al. 2001; Zhou et al. 2011; Insanally et al. 2009; de Villers-Sidani et al. 2008], while task training can still modify neural responses into adulthood [Polley et al. 2006; David et al. 2012; Fritz et al. 2003; Recanzone et al. 1993]. The functional benefit of this arrangement is that auditory cortex can preserve both its general-purpose and task-specific representational capabilities. Further, we have focused on acetylcholine because of its role in both context-specific response changes *and* reward-based learning. However, several different neuromodulators are known to have similar plasticity effects [Glennon et al. 2019; Marlin et al. 2015], suggesting that perceptual learning may be better described by a model where neural tuning is modified by an ensemble of different reward signals in different contexts.

One of the most evocative features of our model is that an animal's initial neural and behavioral response properties can influence its learned representation. In particular, we found that a model's choice bias affects whether more neurons in the system exhibit strong responses to center, versus non-center frequencies. Intuitively this makes sense: if an animal has a bias to indicate non-center frequencies, center frequencies can counteract this bias by increasing their firing rates to exceed the choice threshold and vice versa. Unfortunately, we did not have enough data to conclusively confirm this prediction in the animals we tested, and the active debiasing in our training procedure for the animals may be an additional confound: it would be very interesting to examine systematically in further animal recordings how manipulations of stimulus statistics and animals' initial choice bias interact to affect learned neural responses. We also found that the width of input tuning curves can affect both the final representation and learning speed, with wider

tuning curves producing faster learning speeds and weaker center responses relative to other frequencies. Both of these factors (choice bias and initial tuning width) could provide promising targets for experimental manipulations seeking to predispose auditory cortex responses towards one representation over another.

Our results explain how the auditory cortex resolves a fundamental trade-off between general stimulus representation and task-specific specialization through a combination of reward-based learning and context-specific response modulation. We hope that future work will shed more light on the complexity of both the learning signals projected to cortex and the context-specific response capabilities of the system.

4.4 Attribution

Kathleen Martin and Robert Froemke designed the experiments, and Kathleen Martin performed the experiments and analyzed the resulting 2-photon recordings of mouse auditory cortex. Colin Bredenberg, Eero Simoncelli, and Cristina Savin designed the initial reward-based learning model, and Jordan Lei and Colin Bredenberg performed model experiments exploring the effects of context, choice bias, and input tuning width on models' learned representations. All authors co-wrote the manuscript.

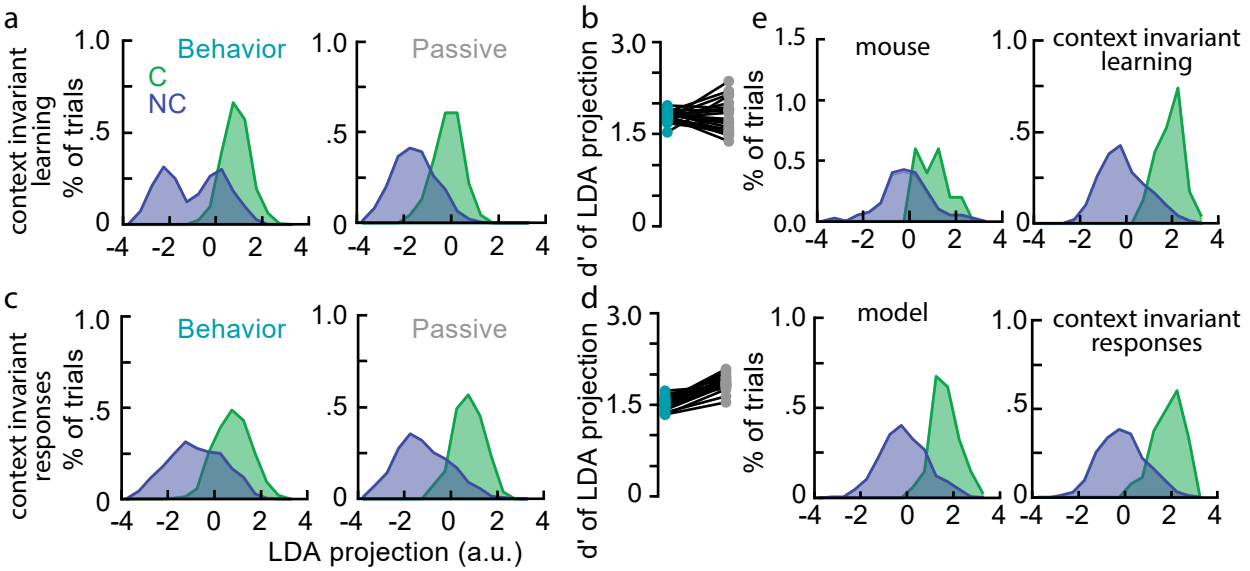

Figure 4.4: Exploring alternative learning schemes. a. Center (green) and non-center (blue) context invariant learning model responses during behavior (left) and passive (right) sessions, projected onto the optimal linear discriminant axis calculated during behavior. **b.** Discriminability (d') of the distribution of center and non-center context invariant learning model responses in behavior (blue) sessions and passive (grey sessions, projected onto the linear discriminant axis. **c.** Same as **a**, but for the context invariant responses condition. **d.** Same as **b**, but for the context invariant responses condition. **e.** Discriminability (d') of the distribution of center and non-center neural responses projected onto the linear discriminant axis calculated during the passive phase for an example mouse (top left), context invariant learning (top right), our model (bottom left), and context invariant responses (bottom right).

5 | Impression learning: Online representation learning with synaptic plasticity

Sensory systems are faced with a task analogous to the scientific process itself: given a steady stream of raw data, they must extract meaningful information about its underlying structure. So far, in Chapters 3 and 4, we have focused on forms of synaptic plasticity with explicit access to reward signals, but in contrast, because the true underlying structure of sensory data is rarely accessible, "representation learning" must be largely unsupervised. Framing perception in the language of Bayesian inference has proven fruitful in perceptual and cognitive science [Knill and Richards 1996; Weiss et al. 2002; Mamassian et al. 2002; Kersten et al. 2004], but has been difficult to connect to biology, because we still lack a satisfactory account of how the machinery of Bayesian inference and learning is implemented in neural circuits [Fiser et al. 2010a; Lange et al. 2020].

Past work includes several examples of circuits that simultaneously learn a top-down generative model of incoming stimuli and perform approximate inference with respect to these models. These differ in the nature of the approximation, from maximum a posteriori estimation [Rao and Ballard 1999], to efficient population codes that embed prior structure [Ganguli and Simoncelli 2014] to either parametric [Kingma and Welling 2013; Rezende et al. 2014] or sampling-based [Dayan

et al. 1995] variational inference. Learning generally takes the form of optimizing a probabilistic objective, either by backpropagation [Kingma and Welling 2013; Rezende et al. 2014] or through local parameter updates, which match biological learning more closely [Dayan et al. 1995; Rao and Ballard 1999; Habenschuss et al. 2012; Bill et al. 2015]. While these models are mostly restricted to static stimuli, several instances also operate over time [Dayan and Hinton 1996; Kutschireiter et al. 2017; Kappel et al. 2014].

Developing biologically plausible learning rules that are applicable to temporally structured data is hampered by the fact that optimizing a probabilistic objective function in such contexts requires access to non-local information across space and time. Previous research on local approximations to credit assignment in backpropagation address spatial credit assignment by ascribing differential functions to the apical and basal dendrites of pyramidal neurons in cortex, where apical dendrites are hypothesized to receive top-down learning signals, and basal dendrites receive bottom-up sensory signals [Körding and König 2001; Urbanczik and Senn 2014; Schiess et al. 2016; Sacramento et al. 2017; Guerguiev et al. 2017; Richards and Lillicrap 2019; Golkar et al. 2020b,a; Payeur et al. 2021]. Locally implementing temporal credit assignment is a bigger challenge [Murray 2019; Marschall et al. 2020].

Our work, which we have dubbed 'impression learning' (IL), combines the tradition of probabilistic learning [Dayan et al. 1995; Dayan and Hinton 1996] with these recent developments in local optimization, in order to learn dynamic stimuli concurrently with perception. We propose a network architecture in which top-down stimulus predictions arriving at the apical dendrites of neurons influence both network dynamics and synaptic plasticity, allowing the network to concurrently learn a probabilistic model of the stimuli and an approximate inference computation. We provide a mathematical derivation of synaptic plasticity rules that approximate gradient descent on a novel unsupervised loss function, along with detailed analyses of the biases induced by this approximation (Section 2.2.2). We explore the empirical and mathematical relationships between IL and three other methods: backpropagation (BP) [LeCun et al. 1989b], the Wake-Sleep

(WS) algorithm [Hinton et al. 1995], and a specific form of neural variational inference (NVI*) [Ranganath et al. 2014; Mnih and Gregor 2014] (which is closely related to REINFORCE). We show that learning can be implemented online (Section 2.2.5), is capable of capturing temporal dependencies in continuous input streams (Section 2.2.4), and demonstrate that IL scales to naturalistic stimuli (Section 2.2.6) and multilayer network architectures (Section 2.2.3), enabling it to learn statistics of high-dimensional, naturalistic inputs better than the reward-based alternative, NVI*.

5.1 Probabilistic inference and local learning in a recurrent circuit

We construct a network of neurons that aims to learn a generative model of the temporal sequence of stimuli that it receives, $p_m(\mathbf{r}, \mathbf{s}) = \prod_{t=0}^{T} p_m(\mathbf{r}_t, \mathbf{s}_t|\mathbf{r}_{t-1})$, in which $\mathbf{s}$ represents stimuli in an input layer.[1] The latent variables $\mathbf{r}$ are not defined by a physical model of the stimulus environment, but are learned in an unsupervised manner to provide the best generative explanation of stimuli received. We assume that stimuli are generated by a *true* probability distribution $p(\mathbf{s}|\mathbf{z})$, where $\mathbf{s}$ corresponds to the first layer of neural activations in an early sensory layer, and vector $\mathbf{z} \sim p(\mathbf{z})$ corresponds to the environmental factors which jointly caused that activity. Because learning is unsupervised, we do not enforce explicit correspondence between the internal and true latent features, $\mathbf{r}$ and $\mathbf{z}$, only a correspondence between model predictions and ground truth stimuli. We also assume that the network performs online inference with respect to its model, inferring the corresponding latent cause $\mathbf{r}$ using Bayes' rule: $p_m(\mathbf{r}|\mathbf{s}) = p_m(\mathbf{r}, \mathbf{s})/p_m(\mathbf{s})$. Because the network won't, in general, be able to explicitly calculate Bayes' rule, we will assume that the network learns an *approximate* inference distribution $q(\mathbf{r}|\mathbf{s})$, which it attempts to bring 'close' to $p_m(\mathbf{r}|\mathbf{s})$. This joint process of learning and inference, known as Bayesian latent feature extraction, provides a general framework for conceptualizing early sensory processing in the brain [Fiser et al. 2010a].

[1] We use the shorthand notation '$\mathbf{s}$' to refer to the $N \times T$ matrix of stimuli across time.

In subsequent sections, we will write a loss function for this general latent feature extraction objective, and show how local modifications at apical and basal synapses can perform approximate gradient descent on this loss.

Loss function The loss function that we propose will produce a learning algorithm where neurons alternate between sampling from the model, p_m, and performing approximate inference according to q in response to real stimuli received from $p(\mathbf{s}|\mathbf{z})$. This alternation will allow the network to learn online in a way that minimally perturbs the continuity of perception. First, consider two families of hybrid probability distributions, which we denote in shorthand $\tilde{q}_\theta$ and $\tilde{p}_\theta$:

$$\begin{aligned}
\tilde{q}_\theta &= \prod_{t=0}^{T} \tilde{q}_t(\mathbf{r}_t, \mathbf{s}_t|\mathbf{z}_t, \lambda_t; \theta) = \prod_{t=0}^{T} \left(q(\mathbf{r}_t|\mathbf{s}_t; \theta_q)p(\mathbf{s}_t|\mathbf{z}_t)\right)^{\lambda_t} p_m(\mathbf{r}_t, \mathbf{s}_t|\mathbf{r}_{t-1}, \lambda_t; \theta_p)^{1-\lambda_t} \\
\tilde{p}_\theta &= \prod_{t=0}^{T} \tilde{p}_t(\mathbf{r}_t, \mathbf{s}_t|\mathbf{z}_t, \lambda_t; \theta) = \prod_{t=0}^{T} \left(q(\mathbf{r}_t|\mathbf{s}_t; \theta_q)p(\mathbf{s}_t|\mathbf{z}_t)\right)^{1-\lambda_t} p_m(\mathbf{r}_t, \mathbf{s}_t|\mathbf{r}_{t-1}, \lambda_t; \theta_p)^{\lambda_t},
\end{aligned} \tag{5.1}$$

where a collection of binary random variables λ_t determines whether, at a given time step, sampling occurs due to $q(\mathbf{r}_t|\mathbf{s}_t; \theta_q)p(\mathbf{s}_t|\mathbf{z}_t)$ or $p_m(\mathbf{r}_t, \mathbf{s}_t|\mathbf{r}_{t-1}, \lambda_t; \theta_p)$, and the full parameter space is denoted $\theta = [\theta_p, \theta_q]$. We define an objective of the form:

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{\lambda,\mathbf{z}}\left[KL[\tilde{q}_\theta||\tilde{p}_\theta]\right] \\
&= \mathbb{E}_{\lambda,\mathbf{z}}\left[\int \left[\log \tilde{q}_\theta - \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r}d\mathbf{s}\right].
\end{aligned} \tag{5.2}$$

This loss provides a generalization of the widely-used evidence lower bound (ELBO), which corresponds to the case $\lambda_t = 1 \ \forall t$. Importantly, we can show that $\mathcal{L} = 0$ if and only if $q(\mathbf{r}_t|\mathbf{s}_t; \theta_q)p(\mathbf{s}_t|\mathbf{z}_t) = p_m(\mathbf{r}_t, \mathbf{s}_t|\mathbf{r}_{t-1}, \lambda_t; \theta_p) \ \ \forall t$. If this equality were achieved, it would also imply $p_m(\mathbf{r}, \mathbf{s}) = q(\mathbf{r}|\mathbf{s})p(\mathbf{s}|\mathbf{z})$. However, this absolute minimum will not be achievable unless $\mathbf{z}_t$ is deterministic, because $p_m(\mathbf{r}_t, \mathbf{s}_t|\mathbf{r}_{t-1}, \lambda_t; \theta_p)$ has no dependency on the latent variables in the environment. Thus, our goal is inherently unachievable, and different choices of $p(\lambda_t)$ and network architectures may lead to different local minima. However, each choice will incentivize learning a

close correspondence between these distributions, and an approximation to gradient descent with respect to *any* choice will lead to local synaptic plasticity rules, making this objective particularly interesting for the computational neuroscience community.

Update derivation We begin by taking the gradient of our new loss w.r.t. $\theta = [\theta_q, \theta_p]$:

$$\begin{aligned}
-\nabla_\theta \mathcal{L} &= -\nabla_\theta \mathbb{E}_{\lambda,\mathbf{z}} \left[\int \left[\log \tilde{q}_\theta - \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} \right] \\
&= -\mathbb{E}_{\lambda,\mathbf{z}} \left[\int \left[\nabla_\theta (\log \tilde{q}_\theta - \log \tilde{p}_\theta)\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} + \int \left[\log \tilde{q}_\theta - \log \tilde{p}_\theta\right] \nabla_\theta \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} \right],
\end{aligned}$$

where the second equality follows from the product rule. Both integrals are analytically intractable, but if we can write both as expectations, they can be approximated by averaging over samples of $\mathbf{r}$ and $\mathbf{s}$. To accomplish this, we note that $\nabla_\theta \tilde{q}_\theta = \nabla_\theta e^{\log \tilde{q}_\theta} = \left[\nabla_\theta \log \tilde{q}_\theta\right] \tilde{q}_\theta$, which allows us to rewrite our expression as an expectation over $\mathbf{r}$ and $\mathbf{s}$:

$$-\nabla_\theta \mathcal{L} = -\mathbb{E}_{\lambda,\mathbf{z}} \left[\int \left[\nabla_\theta \log \tilde{q}_\theta - \nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} + \int \left[\log \tilde{q}_\theta - \log \tilde{p}_\theta\right] (\nabla_\theta \log \tilde{q}_\theta) \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} \right].$$

We also observe that $\int \left[\nabla_\theta \log \tilde{q}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} = \nabla_\theta \int \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} = \nabla_\theta 1 = 0$, allowing the elimination of two terms:

$$\begin{aligned}
-\nabla_\theta \mathcal{L} &= \mathbb{E}_{\lambda,\mathbf{z}} \left[\int \left[\nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} + \int \left[\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right] (\nabla_\theta \log \tilde{q}_\theta) \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} \right] \\
&\approx \mathbb{E}_{\lambda,\mathbf{z}} \left[\int \left[\nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} + \int \left[\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1\right] (\nabla_\theta \log \tilde{q}_\theta) \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} \right] \\
&= \mathbb{E}_{\lambda,\mathbf{z}} \left[\int \left[\nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} + \int \left[\frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right] (\nabla_\theta \log \tilde{q}_\theta) \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} \right] \\
&= \mathbb{E}_{\lambda,\mathbf{z}} \left[\int \left[\nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} + \int \left[\nabla_\theta \log \tilde{q}_\theta\right] \tilde{p}_\theta \, d\mathbf{r} d\mathbf{s} \right].
\end{aligned} \tag{5.3}$$

The approximation in the second line comes from a Taylor expansion of $\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}$ about 0, i.e. when $\frac{\tilde{p}_\theta}{\tilde{q}_\theta} = 1$ (which introduces a bias to the parameter updates that we examine analytically

in Appendix B.1). This expansion is the core of our derivation, and not all algorithms take this approach: for this reason, in Appendix B.2 and B.3 we show how the properties of our algorithm compare to alternatives (NVI*, BP, or WS).

At this point, we have not yet defined $p(\lambda)$. We'll assume that $\lambda_0 \in \{0, 1\}$, that $p(\lambda_0 = 0) = p(\lambda_0 = 1) = 0.5$, and that the λ values alternate deterministically with a 'phase duration' K, i.e. $\lambda_{k+1} = 1 - \lambda_k$ if $\mod(k, K) = 0$, and $\lambda_{k+1} = \lambda_k$ otherwise. Under these conditions, the two integrals in Eq. (5.3) are *equivalent*, and computing our parameter updates only requires sampling from $\tilde{q}$. If we define $\lambda' = 1 - \lambda$, then we have $p(\lambda') = p(\lambda)$ and $\tilde{q}(\mathbf{r}, \mathbf{s}|\mathbf{z}, \lambda; \theta) = \tilde{p}(\mathbf{r}, \mathbf{s}|\mathbf{z}, \lambda'; \theta)$, which we make use of as follows:

$$\begin{aligned}
-\nabla_\theta \mathcal{L} \approx & \mathbb{E}_{\mathbf{z}} \left[\sum_{\lambda} \left[\int \left[\nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r}d\mathbf{s} + \int \left[\nabla_\theta \log \tilde{q}_\theta\right] \tilde{p}_\theta \, d\mathbf{r}d\mathbf{s} \right] p(\lambda) \right] \\
= & \mathbb{E}_{\mathbf{z}} \left[\sum_{\lambda} \int \left[\nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r}d\mathbf{s} p(\lambda) + \sum_{\lambda'} \int \left[\nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r}d\mathbf{s} \, p(\lambda') \right] \\
= & 2\mathbb{E}_{\mathbf{z}} \left[\sum_{\lambda} \int \left[\nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r}d\mathbf{s} \, p(\lambda) \right].
\end{aligned} \tag{5.4}$$

Using the definitions for $\tilde{q}_\theta$ and $\tilde{p}_\theta$ and the properties of the logarithm gives us the following parameter update rule:

$$\Delta\theta \propto 2\mathbb{E}_{\lambda_0, \mathbf{z}} \left[\int \left[\sum_t (1 - \lambda_t) \nabla_\theta \log q_t + (\lambda_t) \nabla_\theta \log p_{mt} \right] \tilde{q}_\theta \, d\mathbf{r}d\mathbf{s} \right]. \tag{5.5}$$

As we will show below, this parameter update equation produces updates that require only information locally available to synapses, a necessary condition for any biologically-plausible algorithm.

Basic model To make the above general learning procedure concrete, we need to specify how to sample from $\tilde{q}_\theta$, which in turn requires an architecture for performing approximate inference at each time step, $q(\mathbf{r}_t|\mathbf{s}_t; \theta_q)$, and a joint model of stimuli and neural activations,

$p_m(\mathbf{r}_t, \mathbf{s}_t|\mathbf{r}_{t-1}, \lambda_t; \theta_p)$. We map these two model components onto neural circuitry, with their own local variables corresponding to $\mathbf{s}$ and $\mathbf{r}$, and segregated synaptic parameters: the 'basal' compartment is dedicated to feedforward inference (q, index 'inf') and the 'apical' compartment is dedicated to generative sampling from the model (p_m, index 'gen'); this segregation allows their influence on neural dynamics to be selectively gated by λ_t (Fig. 5.1a).

First, the internal generative model of the circuit is implicitly defined by a set of currents to the apical dendritic compartment corresponding to generated samples for the next latent variable, $\mathbf{r}_t^{\text{gen}}$:

$$\mathbf{r}_t^{\text{gen}} = ((1 - k_t)\mathbf{D}_r + k_t\mathbf{I})\,\mathbf{r}_{t-1} + \sigma_r^{\text{gen}}\boldsymbol{\eta}_t \tag{5.6}$$

$$\mathbf{s}_t^{\text{gen}} = f(\mathbf{D}_s\mathbf{r}_t) + \sigma_s^{\text{gen}}\boldsymbol{\xi}_t, \tag{5.7}$$

where $\mathbf{D}_r$ is a diagonal transition matrix (constraining generated latent-variables to be independent AR(1) processes), $\mathbf{D}_s$ is a linear decoder, $\mathbf{I}$ is the identity function, $\boldsymbol{\eta}_t$, $\boldsymbol{\xi}_t \sim \mathcal{N}(0, \mathbf{I})$ are independent white noise samples, and σ_r^{gen} and σ_s^{gen} denote respectively the generative standard deviation for neurons at the stimulus and latent levels. We define $k_t = (1 - \delta(\lambda_t - \lambda_{t-1}))\lambda_t$, with $\delta(\cdot)$ the Dirac delta function; k_t is 1 only if $\lambda_t = 1$ and $\lambda_{t-1} = 0$. We chose a piecewise model (gated by k_t) for $\mathbf{r}_t^{\text{gen}}$ because we observed that the statistics of stimuli $\mathbf{s}_t$ given previous activities $\mathbf{r}_{t-1}$ are different if a transition has just occurred ($\lambda_t = 1$ and $\lambda_{t-1} = 0$), which will bias the training of the generative transition parameters $\mathbf{D}_r$. We chose $\mathbf{I}$ for this case, but one could alternatively have a different parametric model for after transitions have occurred.

As we will show, adding this condition to our model will never affect the *dynamics* of our network, but will cause learning for $\mathbf{D}_r$ to occur only on time steps when a transition has not just occurred. Nothing in our derivation requires the transition matrix $\mathbf{D}_r$ to be diagonal, but we constrained it in this way to allow for learning independent latent features. As is, $\mathbf{D}_r$ defines the leakiness of the apical dendritic compartment of the neuron; off-diagonal components of the

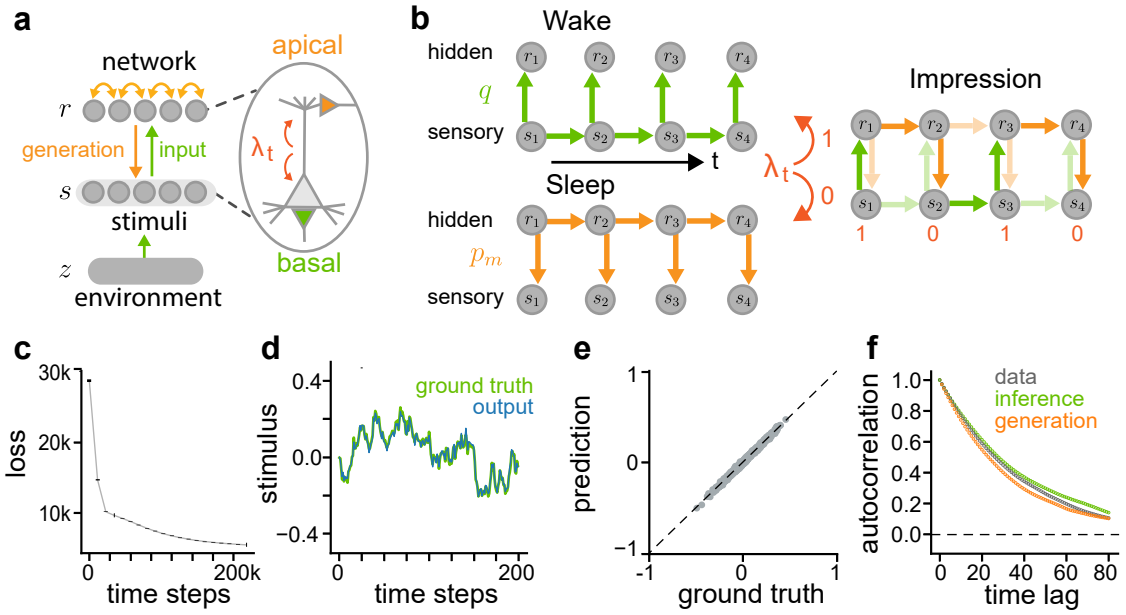

Figure 5.1: Network architecture and learning. a. Model schematic. A neural network receives stimulus inputs at its basal dendrites, and returns lateral and top-down prediction signals via apical synapses. A gate, λ_t, determines whether apical or basal influences dominate network activity. **b.** Learning schedule: the Wake-Sleep (WS) algorithm (left) trains its synapses by alternating between prolonged periods where $\lambda_t = 1$ (Wake) or $\lambda_t = 0$ (Sleep). In contrast, our IL algorithm alternates rapidly between $\lambda_t = 1$ and $\lambda_t = 0$ with phase duration $K = 2$. **c.** Network loss on the artificial stimulus task. Error bars indicate ±1 s.e.m. averaged across 20 network realizations. **d.** Comparison between a ground truth stimulus (green) and the network's prediction (blue) for a particular stimulus dimension. **e.** Same comparison across stimulus dimensions. **f.** The autocorrelation function of $\mathbf{r}$ when the network is performing approximate inference (green; $\lambda_t = 1$), or in generative mode (orange; $\lambda_t = 0$) compared to the autocorrelation of the data (grey).

transition matrix would correspond to recurrent synapses. These dynamics define a probability distribution: $p_m(\mathbf{r}, \mathbf{s}) = \prod_{t=0}^{T} p_m(\mathbf{r}_t, \mathbf{s}_t | \mathbf{r}_{t-1}, \lambda_t; \theta_p)$.

Second, we define our inference model, a factorized conditional probability distribution $q(\mathbf{r}|\mathbf{s}) = \prod_{t=0}^{T} q(\mathbf{r}_t|\mathbf{s}_t; \theta_q)$, which applies a feedforward nonlinear transformation to incoming stimuli:

$$\mathbf{r}_t^{\text{inf}} = f(\mathbf{W}\mathbf{s}_t) + \sigma_r^{\text{inf}} \boldsymbol{\eta}_t, \tag{5.8}$$

where $\mathbf{W}$ denotes the feedforward weights, σ_r^{inf} is the inference standard deviation for neurons at the latent level, and the nonlinearity $f(\cdot)$ is the tanh function. During inference mode, the stimulus layer receives latent-associated inputs from the environment, further corrupted by the

same noise as the internal representation:

$$\mathbf{s}_t^{\text{inf}} = \bar{\mathbf{s}}(\mathbf{z}_t) + \sigma_s^{\text{inf}} \boldsymbol{\xi}_t, \tag{5.9}$$

where σ_s^{inf} denotes the standard deviation for neurons at the stimulus level, and $\bar{\mathbf{s}}(\mathbf{z}_t)$ is input from external stimuli. During simulations, samples are determined by a combination of p_m and q, given by $\tilde{q}_\theta$:

$$\mathbf{r}_t = \lambda_t \mathbf{r}_t^{\text{inf}} + (1 - \lambda_t)\mathbf{r}_t^{\text{gen}} \tag{5.10}$$

$$\mathbf{s}_t = \lambda_t \mathbf{s}_t^{\text{inf}} + (1 - \lambda_t)\mathbf{s}_t^{\text{gen}}. \tag{5.11}$$

We interpret these dynamics biologically as network of recurrently connected pyramidal neurons with two sources of input, one to the apical dendrites ($\mathbf{r}_t^{gen}$ or $\mathbf{s}_t^{\text{gen}}$) and one to the basal dendrites ($\mathbf{r}_t^{\text{inf}}$ or $\mathbf{s}_t^{\text{inf}}$). The gating variable λ_t determines which input source controls the circuit dynamics.

Plasticity rule interpretation Inserting our particular choice of q_t and p_{mt} into our approximate gradient descent derivation, the parameter updates can be interpreted as local synaptic plasticity rules at the basal (for q_t) or apical (for p_{mt}) compartments of our neuron model:

$$\log q(\mathbf{r}_t|\mathbf{s}_t;\theta_q) = -\frac{1}{2\left(\sigma_r^{\text{inf}}\right)^2} \|\mathbf{r}_t - f(\mathbf{W}\mathbf{s}_t)\|_2^2 + c_q \tag{5.12}$$

$$\begin{aligned}\log p_m(\mathbf{r}_t, \mathbf{s}_t|\mathbf{r}_{t-1}, \lambda_t;\theta_p) = &-\frac{1}{2(\sigma_r^{\text{gen}})^2} \|\mathbf{r}_t - ((1-k_t)\mathbf{D}_r + k_t\mathbf{I})\,\mathbf{r}_{t-1}\|_2^2 \\ &-\frac{1}{2(\sigma_s^{\text{gen}})^2} \|\mathbf{s}_t - f(\mathbf{D}_s\mathbf{r}_t)\|_2^2 + c_p,\end{aligned} \tag{5.13}$$

where $c_q = -N_r \log(\sqrt{2\pi(\sigma_r^{\text{inf}})^2})$ and $c_p = -N_r \log(\sqrt{2\pi(\sigma_r^{\text{gen}})^2} - N_s \log(\sqrt{2\pi(\sigma_s^{\text{gen}})^2}$ are constants that do not depend on network parameters. We can use these equations to evaluate our weight updates, by using the general formula in Eq. 5.5 and calculating derivatives. For online parameter updates, we assume that weights change stochastically at each time step, based on samples from

λ_0, $\mathbf{z}$, $\mathbf{r}$, and $\mathbf{s}$ (instead of explicitly calculating the expectation in Eq. 5.5):

$$\Delta \mathbf{W}^{(ij)} \propto \frac{1-\lambda_t}{\left(\sigma_r^{\text{inf}}\right)^2}(\mathbf{r}_t^{(i)} - f(\mathbf{W}\mathbf{s}_t)^{(i)}) f'(\mathbf{W}\mathbf{s}_t)^{(i)} \mathbf{s}_t^{(j)} \tag{5.14}$$

$$\Delta \mathbf{D}_r^{(ii)} \propto \frac{\lambda_t(1-k_t)}{\left(\sigma_r^{\text{gen}}\right)^2}(\mathbf{r}_t^{(i)} - (\mathbf{D}_r \mathbf{r}_{t-1})^{(i)}) \mathbf{r}_{t-1}^{(i)} \tag{5.15}$$

$$\Delta \mathbf{D}_s^{(ij)} \propto \frac{\lambda_t}{\left(\sigma_s^{\text{gen}}\right)^2}(\mathbf{s}_t^{(i)} - f(\mathbf{D}_s \mathbf{r}_t)^{(i)}) f'(\mathbf{D}_s \mathbf{r}_t)^{(i)} \mathbf{r}_t^{(j)}. \tag{5.16}$$

Each of these updates has the form of a local synaptic plasticity rule, under the following assumptions: $\mathbf{W}^{(ij)}$ is a basal synapse from neuron j to neuron i, $\mathbf{r}_t^{(i)}$ and $\mathbf{r}_t^{(j)}$ correspond to the pre- and post-synaptic firing rates, respectively, and $f(\mathbf{W}\mathbf{s}_t)^{(i)}$ corresponds to the local basal current injected into neuron i. Thus, assuming that a basal synapse has access to both the neuron's firing rate and its local basal synaptic current at a particular moment in time, $\Delta \mathbf{W}^{(ij)}$ is local; the same principle holds for the apical updates. If $\lambda_t = 0$, then network activity is driven by the generative inputs, and so the parameter updates for basal synapses depend on apically-driven activity, as has been observed experimentally [Bittner et al. 2015]; similarly, apical synaptic plasticity should depend on basally-driven activity. The updates for the generative transition matrix, $\mathbf{D}_r$—determining the leakiness of the apical dendritic compartments—are gated by $1-k_t$, indicating that parameter updates are delayed upon entering 'inference' mode: this could reasonably be implemented biologically by a slow cascade of biochemical processes that delay changes in neural parameters, as has been proposed by previous plasticity models [Fusi et al. 2005; Clopath et al. 2008].

5.2 Numerical Results

Validation on artificial stimuli To analyze IL performance in an environment where we have access to and control over the statistics of the latent dynamics $\mathbf{z}_t$, we constructed artificial stimuli

as follows:

$$\mathbf{z}_t = \mathbf{\Lambda}\mathbf{z}_{t-1} + \sigma^{\text{true}}\boldsymbol{\eta}_t \tag{5.17}$$

$$\bar{\mathbf{s}}(\mathbf{z}_t) = \mathbf{A}\mathbf{z}_t, \tag{5.18}$$

where $\mathbf{\Lambda}$ is a $N_z \times N_z$ diagonal matrix with $\Lambda^{ii} < 1 \; \forall i$, $\mathbf{A}$ is a $N_s \times N_z$ random matrix with $A^{ij} \sim \mathcal{N}(0, \frac{1}{N_z})$, and $\boldsymbol{\eta}_t \sim \mathcal{N}(0, 1)$.[2] For simplicity, we fix the dimension of the latent space and the generative noise in the network to the ground truth values, $N_r = N_z$ neurons, and $\sigma^{\text{true}} = \sigma_r^{\text{gen}}$, so that in principle our model $\int p_m(\mathbf{r}, \mathbf{s})d\mathbf{r}$ can match the ground truth data distribution $\int p(\mathbf{s}, \mathbf{z})d\mathbf{z}$ exactly. This also means that we can verify that the network has learned an optimal model by comparing its second-order statistics to those of the ground truth distribution.

We trained the network using IL, verifying that the online synaptic updates minimize the loss $\mathcal{L}$ (Fig. 5.1c). We further validate that the network has learned to accurately perform inference, so that $q(\mathbf{r}|\mathbf{s}) \approx p_m(\mathbf{r}|\mathbf{s})$, and that the network has learned a good model of the data, so that $\int p_m(\mathbf{r}, \mathbf{s})d\mathbf{r} \approx p(\mathbf{s})$, as per our original goals. We show that when the network is performing approximate inference, i.e. $\lambda_t = 1, \; \forall t$, stimulus reconstructions based on the network's latent state are closely matched to the actual stimuli, i.e. $\mathbf{s}_t \approx f(\mathbf{D}_s\mathbf{r}_t)$, meaning that the network is functioning as a good autoencoder across time (Fig. 5.1d), and across all stimulus dimensions (Fig. 5.1e). To verify the network's generative performance, we also show that the temporal autocorrelations for the network rates $\mathbf{r}_t$ in generative mode ($\lambda_t = 0 \; \forall t$) closely overlap with the ground truth autocorrelation structure of $\mathbf{z}$, suggesting that the learned latent features correspond (modulo a rotation) to the true latent features. Note that this latent variable match occurs because we have enforced a correspondence between the true data-generating distribution and our model, and would not necessarily happen if a different model architecture were used.

Algorithm comparisons Having verified that IL is capable of training the network on

[2]The parameter values and initialization details for all simulations are included in the supplementary code, which was run on an internal cluster; $N_s = 100$ and $N_z = 20$.

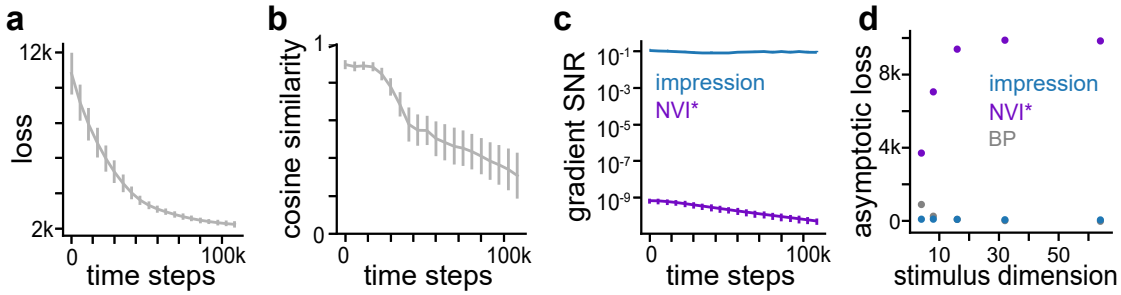

Figure 5.2: Comparing learning algorithms and effects of dimensionality. a. Loss throughout time. **b.** Cosine similarity between gradient updates given by IL and NVI*, averaged over 10^6 samples. **c.** The signal-to-noise ratio for IL (blue), compared to NVI* (purple) across learning. **d.** Asymptotic negative ELBO loss for IL (blue), NVI* (purple), and BP (gray) as a function of the stimulus dimensionality. Error bars indicate ±1 s.e.m. averaged across 20 network realizations.

simulated data, we next compared it to alternative algorithms in the literature, including neural variational inference (NVI*), BP, and WS (see Appendix B.2 for detailed mathematical comparisons and derivations). In particular, NVI* provides an alternative candidate model of how the brain could plausibly learn neural representations through variational inference [Mnih and Gregor 2014]. Because NVI* performs poorly for high-dimensional stimuli and large numbers of time steps (Appendix B.3; [Werfel et al. 2003]), we simplified the task by reducing the dimensionality of the latent space, $N_z = 2$, and stimulus space, $N_s = 4$. For twenty evenly-spaced time points over the course of the learning trajectory, we compared the inference parameter updates given by IL, $\Delta\theta_q^{\text{IL}}$, to the inference parameter updates given by NVI*, $\Delta\theta_q^{\text{NVI}}$, for a 4 time-step stimulus sequence (Fig. 5.2a). To get good estimates of the mean and variance of these sample parameter updates, we averaged over 10^6 different realizations of the network noise, and compared the samples using two measures. First, we considered the cosine similarity (normalized inner product) between the two empirical mean updates, $\overline{\Delta\theta_q^{\text{IL}}} = \frac{1}{N}\sum_{k=0}^{N}\Delta\theta_q^{\text{IL}}$ and $\overline{\Delta\theta_q^{\text{NVI}}} = \frac{1}{N}\sum_{k=0}^{N}\Delta\theta_q^{\text{NVI}}$ (Fig. 5.2b), where $\cos(\theta) \in [-1, 1]$, and $\cos(\theta) < 0$ in this case would indicate that the parameter updates are anticorrelated. Because the NVI* update is unbiased, ie. $\mathbb{E}[\Delta\theta_q^{\text{NVI}}] = -\frac{d}{d\theta_q}\mathcal{L}$, as long as we have averaged over a sufficient number of samples N, a positive cosine similarity across learning between the IL update and the NVI* update (Fig. 5.2b) indicates that our update is aligned

in expectation to the true gradient of the loss, and hence will improve performance. This is a way of empirically verifying that the bias we introduce in our derivation does not impair the learning process.

Having verified that the IL update and the true gradient are aligned on average, we next examine whether the updates given by NVI* differ in terms of their signal-to-noise ratio (SNR) from the IL updates, where we define the SNR as:

$$\mathrm{SNR}(\Delta\theta_q) = \frac{1}{N_\theta}\sum_{i=0}^{N_\theta}\frac{\left(\overline{\Delta\theta}_q^{(i)}\right)^2}{S^2(\Delta\theta_q^{(i)})}, \tag{5.19}$$

where $S^2(\cdot)$ denotes the sample variance. This measure is an average across individual parameter updates $\Delta\theta^{(i)}$, and it increases with $\left\|\overline{\Delta\theta}_q\right\|_2^2$ and decreases as the estimator variance grows. As Fig. 5.2c shows, the SNR is many orders of magnitude lower for NVI* than for IL over learning, likely due to the high estimator variance of the NVI*, which we demonstrate analytically for a simple example in the Appendix B.3. The estimator variance has direct implications for the speed of learning and asymptotic performance, so that even though NVI* and IL can have parameter updates that are aligned in expectation, due to its low variance IL will greatly outperform NVI* during training.

We verified the generality of these benefits in the same task, as we varied N_s, N_z and N_r concurrently, so that $N_s = 2N_z = 2N_r$. We optimized learning rates for NVI*, BP, and IL separately on the lowest dimensional condition by grid search across orders of magnitude (10^{-2}, 10^{-3}, etc.), and found that NVI* performed worse over the entire range, while IL and BP showed similar performance (using the negative ELBO loss as a standard). Moreover, while NVI* showed worse performance as the stimulus dimension increased, this was not the case for IL or BP (Fig. 5.2d).

Phase duration effects The previous numerical results verify that IL is able to effectively learn a generative model of artificial data, and to perform inference with respect to that model. However, for IL to be a valid candidate for online learning in the brain, the learning process should

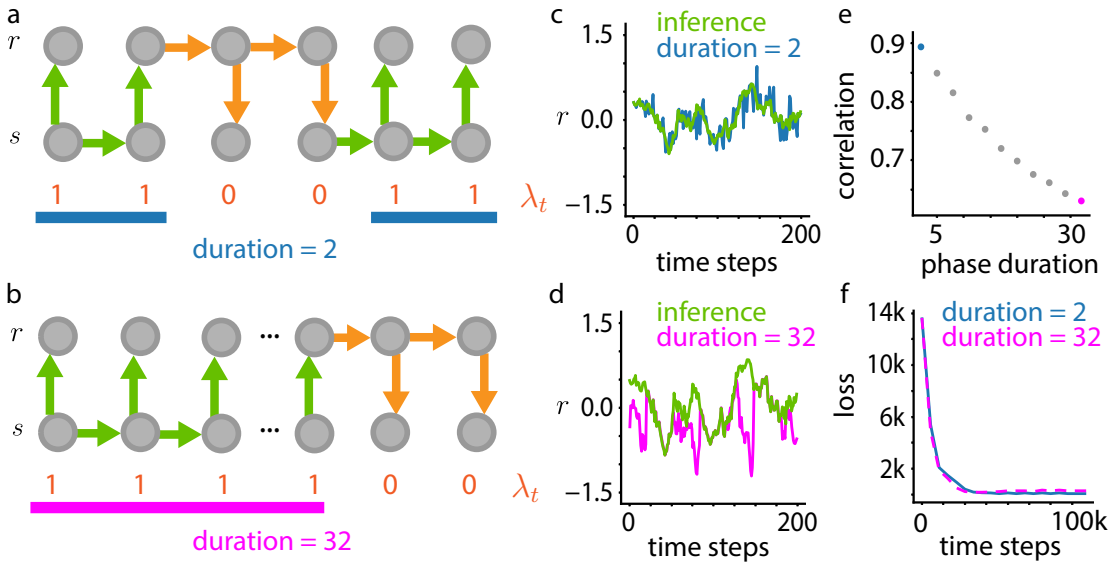

Figure 5.3: The effects of phase duration on dynamics and learning. a. Schematic of IL with a phase duration of 2. **b.** Same as **a**, but for a phase duration of 32. **c.** Comparison of an example neuron's activity through time when the network is in inference mode (green, $\lambda_t = 1$) and when the network is alternating phase with duration 2 (blue); the random seed and stimuli are identical in both cases. **d.** Same as **c**, but for a phase duration of 32 (pink). **e.** The correlation across time between neurons in inference mode vs. while alternating phase, for identical random seeds. **f.** The negative ELBO loss for a network trained with a phase duration of 2 (blue, solid line) or 32 (pink, dashed line).

not significantly interfere with perception. To test this, we explored how the 'phase duration' K affects the correlation between network activity in a simulation where $\lambda_t = 1,\ \forall t$, and a simulation where λ_t alternates phases every K time steps (for a fixed random seed and stimulus sequence). If the learning process did not interfere with perception at all, this correlation would be 1, and if it completely disrupted perception it would be 0, or even negative. In Fig. 5.3c and d, we show two example traces with $K = 2$ and $K = 32$, respectively, comparing the network in inference mode to the network during learning. While neural trajectories for the shorter phase durations are closely correlated, they deviate considerably for longer phase durations (Fig. 5.3c-e). Despite this, the loss profile (negative ELBO) is identical. Since WS can be viewed as a special case of IL for very long phase durations (Appendix B.2.3; see Fig. 5.5a for an even longer phase duration), this implies that the two methods have similar performance. However, IL operating in a mode of fast

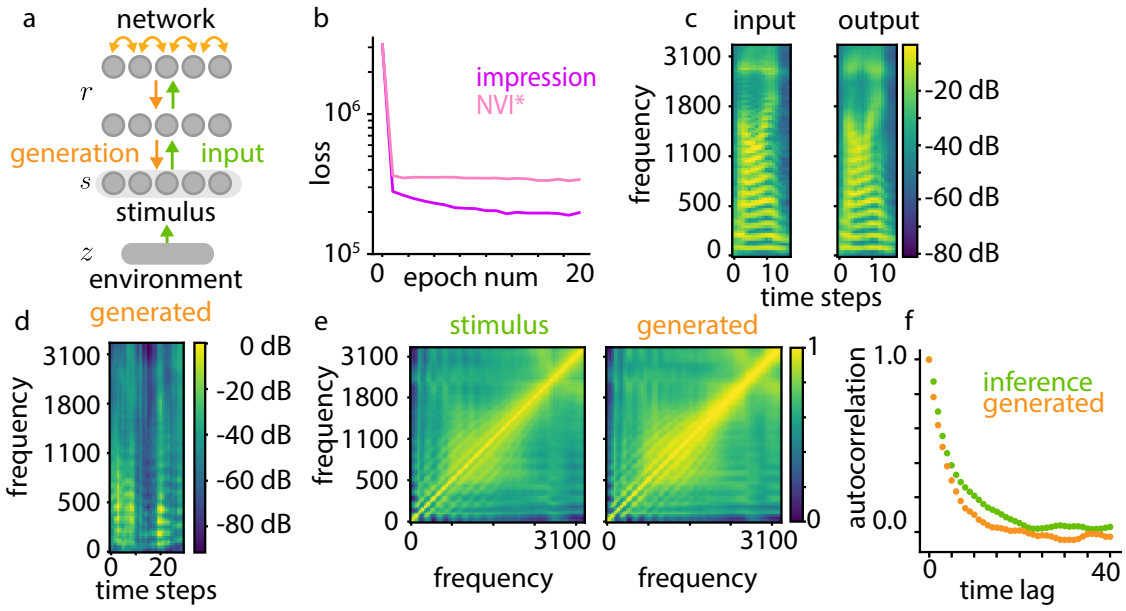

Figure 5.4: Learning auditory sequences in a multilayer network. a. Hierarchical network architecture. **b.** Test loss across epochs for IL (blue) and NVI* (purple). **c.** Comparison between an example data input and the corresponding network output in inference mode ($\lambda_t = 1$). **d.** Sample network output in generative mode ($\lambda_t = 0$). **e.** Across-frequency amplitude correlations for the data (left) and for network-generated samples (right). **f.** Auto-correlation function of a neuron in inference and generative modes.

fluctuations between inference and generation may be more biologically relevant, as this reduces the interference with perception without impairing learning. Moreover, we found that lengthening the duration of the inference phase alone while keeping very short bursts of generative activity further reduced perceptual disturbance, while only slightly increasing the time required to learn (Fig. 5.5b-d).

Spoken digits task Having verified the performance of IL on artificial stimuli, we next tested its performance on higher-dimensional and more complex naturalistic stimuli. We used the training and test sets of the Free Spoken Digits Dataset [Jackson et al. 2018], which provides audio time series of humans speaking digits 0-9.[3] We transformed these time series into log-mel spectrograms as a coarse approximation of the initial stages of the human auditory system, shifted the inputs by a constant so as to make them all positive, and divided the result by the across-

[3]The FSDD is available at https://github.com/Jakobovski/free-spoken-digit-dataset.

channel standard deviation. The results of Fig. 5.4 are shown in the original log-mel spectrogram input space.

To assess the hierarchical processing capabilities of IL, we added an additional feedforward layer to the network architecture (Fig. 5.4a); we provide the details of how this modification affects simulation and parameter updates in Appendix B.4. To compare IL to NVI*, we again optimized learning rates via grid search across orders of magnitude, and found that IL greatly outperformed NVI* when each was evaluated at its respective optimal learning rate (Fig. 5.4b). Furthermore, we observed that our trained network meets the same criteria for success as for our artificial stimuli, namely its stimulus reconstructions closely match the true stimulus while in inference mode ($\lambda_t = 1\ \forall t$; Fig. 5.4c), and sample stimuli produced while the network is in generative mode ($\lambda_t = 0\ \forall t$) qualitatively correspond to ground-truth stimuli (Fig. 5.4d), and quantitatively match the structure of both spatial (Fig. 5.4e) and temporal (Fig. 5.4f) autocorrelation of the input. These results collectively demonstrate that IL is capable of training neural representations of complex real-world stimuli. They also show that IL can function when there is a mismatch between its architecture and the structure of environmental latent variables, which are in this case unknown. In general, learning may fail if the chosen network architecture is too restrictive.

5.3 Discussion

Impression learning (IL) provides a potential mechanism for the brain to learn generative models of its sensory inputs through local synaptic plasticity, while concurrently performing approximate inference with respect to these models. IL is a direct generalization of the Wake-Sleep algorithm [Hinton et al. 1995], which replaces lengthy offline 'Sleep' phases with brief substitutions of network-generated samples in place of incoming data, in a way that minimally perturbs natural neural trajectories. Transitions between 'inference mode' and 'generative mode' are controlled by a global signal λ_t, which decides whether generative signals to the apical synapses or inference

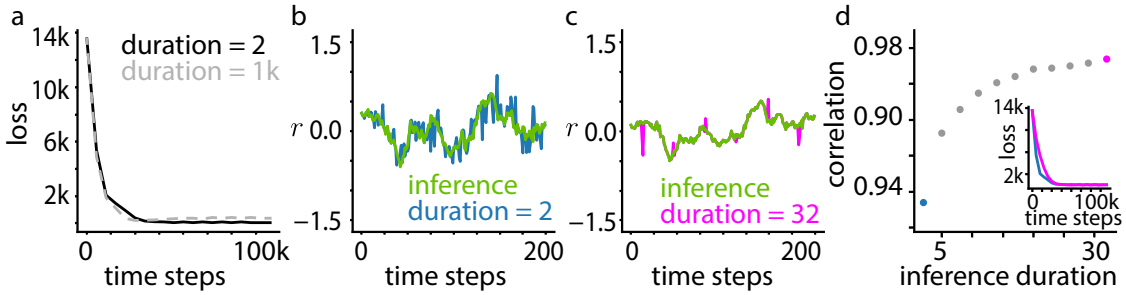

Figure 5.5: Additional variations on the phase duration. a. Comparison of ELBO loss for IL (black) to WS with a 1000 time step phase duration (gray) over training. **b.** Comparison of an example neuron's activity through time when the network is in inference mode (green, $\lambda_t = 1$) and when the network is alternating phase, spending 2 time steps in the inference phase, and two time steps the generative phase (blue); the random seed and stimuli are identical in both cases. **c.** Same as **b**, but the alternating network spends 32 time steps in the inference phase. **d.** The correlation across time between neurons in inference mode vs. while alternating phase, for identical random seeds. The inference duration is incremented, while the generative duration is kept constant at 2 time steps. Inset shows the loss for an inference duration of 2 (blue) compared to the loss for an inference duration of 32 (pink).

signals to the basal synapses dominate network activity.

Computationally, IL outperforms NVI* [Ranganath et al. 2014; Mnih and Gregor 2014], a particular instance of three-factor plasticity [Frémaux and Gerstner 2016], because its internal model provides explicit 'credit assignment' for each individual neuron, rather than implicitly calculating it via correlations between neural activity and a global reward signal. This leads to lower-variance gradient estimates and faster learning. Alternative learning algorithms such as backpropagation (through time) [Werbos 1990] are not intrinsically probabilistic, but can be used for optimizing probabilistic objectives. Like IL, BP provides explicit credit assignment, but the parameter updates it provides are nonlocal across both network layers and time. It is worth noting that IL was developed in a purely unsupervised learning setting, whereas both BP and NVI* extend to supervised and reinforcement learning [Mnih et al. 2015; Williams 1992]. In the context of supervised learning, several biologically-plausible approximations to BP leverage the apical-basal dendritic structure of pyramidal neurons to learn [Payeur et al. 2021; Guerguiev et al. 2017], based primarily on target-propagation [Bengio 2014] or its variants [Lee et al. 2015]. It would be valuable to explore the combination of such extensions with the continuous online learning capabilities of

IL.

Local computations are considered a necessary condition for learning algorithms to be biologically-plausible (Section 2.2.1). In our framework, locality is enforced through the structure of the internal graphical model (p_m) and the approximate inference distribution (q): any choice of neural network architecture with independent noise will guarantee local plasticity. Our framework is relatively agnostic to the details: neurons could be either rate-based with Gaussian intrinsic noise (as in the examples presented here), or generate spikes with Poisson variability, which would result in synaptic updates analogous to empirically observed spike-timing-dependent plasticity, as found in generalizations of WS [Dayan and Hinton 1996]. It would also be possible to make distinctions between excitatory and inhibitory neurons, by requiring all outgoing synapses from individual neurons to be either positive or negative, or to include more complex dendritic arborizations, as have been explored in recent experimental [Rashid et al. 2020] and modeling [Sezener et al. 2021] efforts. Our current model enforces hard, global phase distinctions ($\lambda_t \in \{0, 1\}$ for all neurons), which could potentially correspond to alternations between activity driven by apical dendritic calcium events and basal spiking tied to theta oscillations in the hippocampus [Bittner et al. 2015]. However, cortical data indicate that input to apical and basal dendrites contribute concurrently and constructively to spiking activity [Larkum et al. 1999]. We are currently working to extend our derivation to these circumstances, by allowing λ_t to be non-binary and heterogenous across neurons.

Traditional predictive coding [Rao and Ballard 1999] requires steady-state assumptions for learning, meaning that neural dynamics must occur on a timescale much faster than that of stimuli. In contrast, IL requires a mechanism by which the relative influence of the apical and basal dendrites of pyramidal neurons can be rapidly switched, along with learning mechanisms that operate at that timescale. If such a mechanism could be experimentally identified and controlled, our model makes the specific prediction that increasing the dominance of apical dendritic input on neural activity ($\lambda_t \approx 1$) would cause the network to sample from its generative model, i.e. the

manipulation will induce structured hallucinations that mimic realistic stimuli (and associated neural activity), without being tied to the sensory world. One candidate gating mechanism is rapid inhibition targeting apical dendrites specifically [Larkum 2013; Saudargiene et al. 2015; Guerreiro et al. 2020; Leão et al. 2012]; but much work remains to explicitly relate this mechanism to learning and plasticity.

IL predicts that synapses will use an error signal based on the difference between local dendritic compartmental currents (either apical or basal) and the neuron's total firing rate to perform learning (Section 2.2.7). There is some evidence that spiking activity driven by apical inputs to pyramidal neurons can induce plasticity at basal synapses [Bittner et al. 2015, 2017], and several studies have found systematic changes in synaptic plasticity between apical and basal synapses, in particular the sign changes induced by local dendritic inputs that IL predicts [Froemke et al. 2005; Sjöström and Häusser 2006; Letzkus et al. 2006; Froemke et al. 2010]. Hence, IL has the potential to explain the diversity of plasticity phenomena observed experimentally and inform future experiments.

5.4 Attribution

Colin Bredenberg, Eero Simoncelli, and Cristina Savin designed the impression learning model. Colin Bredenberg performed experiments assessing impression learning's performance on controlled and naturalistic stimuli. Benjamin Lyo performed experiments comparing impression learning to backpropagation. All authors co-wrote the manuscript.

6 | Recurrent neural circuits overcome partial inactivation by compensation and re-learning

In previous chapters we have developed normative theories of synaptic plasticity and compared them to neural data, highlighting the brain's incredible dynamical complexity and capacity for adaptation. In this chapter, we explore how these complexities present challenges for interpreting experiments that artificially manipulate neural circuits.

Artificial manipulations are vital tools in modern neuroscience for investigating the neural computations that underlie behavior. These manipulation techniques include lesions [Vaidya et al. 2019; Newsome and Pare 1988], pharmacological inactivation [Katz et al. 2016; Hanks et al. 2015], microstimulation [Salzman et al. 1990], optogenetic [Fetsch et al. 2018; Brown et al. 2018; Tremblay et al. 2020; Rajasethupathy et al. 2016b], and chemogenetic [Tervo et al. 2014; Eldridge et al. 2016] inactivation or excitation. It is commonly assumed that the results of these experiments are easily interpretable, i.e., that changes of behavior following artificial inactivation or excitation of a circuit demonstrate the importance of that circuit in producing that behavior. However, the converse of this statement—that the absence of changes in behavior following circuit manipulation indicate that this circuit does not play a role in producing the behavior—is much more difficult to assert.

Beyond sensory bottlenecks, the distributed nature of computations for higher brain functions

makes attribution of a single function to a single circuit much more challenging, because multiple areas may jointly contribute to a function, and may be able to mutually compensate for inactivity in other regions [Wolff and Ölveczky 2018]. Furthermore, the capacity of neural circuits to compensate for inactivation is well documented [Vaidya et al. 2019]. This implies that other neurons in the circuit or areas of the brain which are not normally causal in producing a particular behavior can adapt to play an important role. Transient manipulation may produce effects that are more difficult to adapt to, but there is evidence for compensation for even transient optogenetic inactivation [Fetsch et al. 2018].

Since the effect of perturbations may not always follow simple intuition, modelling can provide a useful way to reason about possible experimental outcomes and their interpretation. For this we turn to artificial recurrent neural networks (RNNs), which have been successfully used as a bridge between neural activity and behavior in several tasks [Rigotti et al. 2010; Mante et al. 2013; Yang et al. 2019]. RNNs are powerful model systems that share many complexities of brain circuits, while permitting direct access to the inner-workings of the system. Critically, the choice of architecture and training objectives allows the exploration of different circuit scenarios, where the contribution of network elements to the output is known. Moreover, simulations provide the ability of perturbing activity with any spatial and temporal resolution. The combination of these features results in immense control and knowledge about complex networks, at a level unattainable in real brain circuits. We build on this foundation to investigate the ability of causal interventions to reveal the role neurons and populations play in the distributed computations performed in a complex network.

As a specific example of the complexities involved in causal manipulation of neural circuits, we focus on the integration of sensory evidence for the random dots motion (RDM) task, a decision making paradigm that engages multiple frontoparietal cortices [Kiani et al. 2014b; Roitman and Shadlen 2002; Kim and Shadlen 1999; Mochol et al. 2021] and subcortical areas [Ding and Gold 2013; Horwitz and Newsome 1999; Ratcliff et al. 2011], and which has been a target for many

causal studies, with contradictory outcomes [Katz et al. 2016; Zhou and Freedman 2019a; Hanks et al. 2006; Licata et al. 2017; Erlich et al. 2015]. These studies suggest a distributed process for neural implementation of decision-making, whose complexity challenges standard experimental techniques for identifying causal relationships between individual brain regions and behavior. On the modeling side, several successful RNN-based models have been used to replicate neural response dynamics and behavior in perceptual decision making [Mante et al. 2013; Wong and Wang 2006; Rigotti et al. 2010] and their dynamical systems properties are well understood. In particular, recurrent networks can perform near-optimal evidence integration by constructing a low-dimensional attractor [Goldman et al. 2003; Wong and Wang 2006; Cain et al. 2013], whose structure provides a direct route to investigating the computational integrity of the circuit.

Here, we use similar trained RNN circuits to systematically study causal manipulations of the RDM task. We show that inactivation of subsets of neurons in the network affects behavior by damaging the low-dimensional attractor, with larger activity perturbations having a greater impact on both accuracy and reaction time, more so for the latter. In a more complex network, where integration is done collaboratively by multiple circuits, inactivation may or may not affect the computational structure of the solution and the behavior. In particular, in networks with parallel, redundant computation, inactivation of a subset of the circuit can have little to no effect on the network output. Lastly, we demonstrate that recurrent neural networks that retain plasticity — and continue learning — after the inactivation reconfigure themselves to regain the accuracy they had prior to inactivation, under some conditions much more quickly than the original training time. The speed of recovery depends on the extent and temporal profile of the inactivation, and network performance is closely related to the integrity of the network's attractors. These observations caution against simplistic interpretations of causal experiments and suggest concrete ways to avoid interpretational pitfalls and improve experimental design.

6.1 Results

6.1.1 Hierarchical recurrent networks approximate linear integration for simple sensory decisions

We begin with the simplest hierarchical recurrent network architecture (Fig. 6.1a) consisting of a sensory-like population (P1) and an integrating population (P2). Neurons in each population have dense recurrent connections between them, while the sensory population projects sparsely to the integrating population. The P2 population roughly corresponds to the collection of the recurrently connected cortical and sub-cortical neurons involved in the decision-making process; however, it does not reflect the precise anatomy of brain networks. The stimulus in each trial randomly fluctuates around a mean stimulus strength level, akin to the dynamic random dots stimuli in direction discrimination tasks [Newsome et al. 1989; Roitman and Shadlen 2002] where motion energy fluctuates around a mean dictated by the coherence of random dots. This fluctuating stimulus input is received by the sensory population P1, and relayed to the integrating population P2. The network is trained such that a linear read-out of the activity of population P2 at each moment matches the integral of the stimulus input up to that time. All connections in the network are plastic during training and modified by backpropagation through time (BPTT) (see Methods). After learning, the sensory population shows coherence tuning (see example response profiles in Fig. 6.7a-c), while the integration population develops response profiles — ramping activity — similar to those reported in posterior parietal and prefrontal cortex (Fig. 6.7d-f). The connections are fixed after the initial training, reflecting the common assumption that synapses and network dynamics do not significantly change after inactivation. We separately also explore the effects of continuous plasticity on causal experiments (Section 6.1.5).

The network is trained within a couple of thousands of trials (Fig. 6.1b), similar to training schedules for nonhuman primates [Gold et al. 2010]. After training, the P2 output closely approxi-

mates the integral of the stimulus input over time. Two hallmarks of temporal integration are linear scaling with time (Fig. 6.1c) and stimulus strength (Fig. 6.1c-d) [Gold and Shadlen 2007]: if a network is receiving a constant-mean stimulus input, the integrated output will be a linear function over time, with slope equal to the mean of the stimulus. The model output represents linear integration for a wide range of inputs and times but saturates for very large values of the integral. Because of the limited dynamic range of the neural responses, the curtailed range of the integral improves the precision of the representation of the integrated evidence for weaker stimuli, where the network precision matters the most for the accuracy of choices. Another hallmark of temporal integration of noisy inputs is linear growth of the variance of the integral over time. The motion energy of the random dots stimulus at each time is an independent sample from a normal distribution, so their sum over time—integration—should have a variance that scales linearly with time [Roitman and Shadlen 2002; Churchland et al. 2011]. Our network output captures this hallmark of optimal integration (Fig. 6.1e).

Since the network's integration process stops when the network output reaches a fixed decision bound, the model provides trial-by-trial estimates of the network's decision time and choice. The time to bound is the decision time, and the sign of the network output at the time of bound crossing determines the choice (right or leftward motion). Our model decision times are in units of the network time steps. The resulting psychometric and chronometric functions of the model show profiles qualitatively similar to experimental results, in particular, faster, more accurate responses for stronger motion stimuli (Fig. 6.1f-g) [Roitman and Shadlen 2002; Kiani et al. 2014a; Palmer et al. 2005].

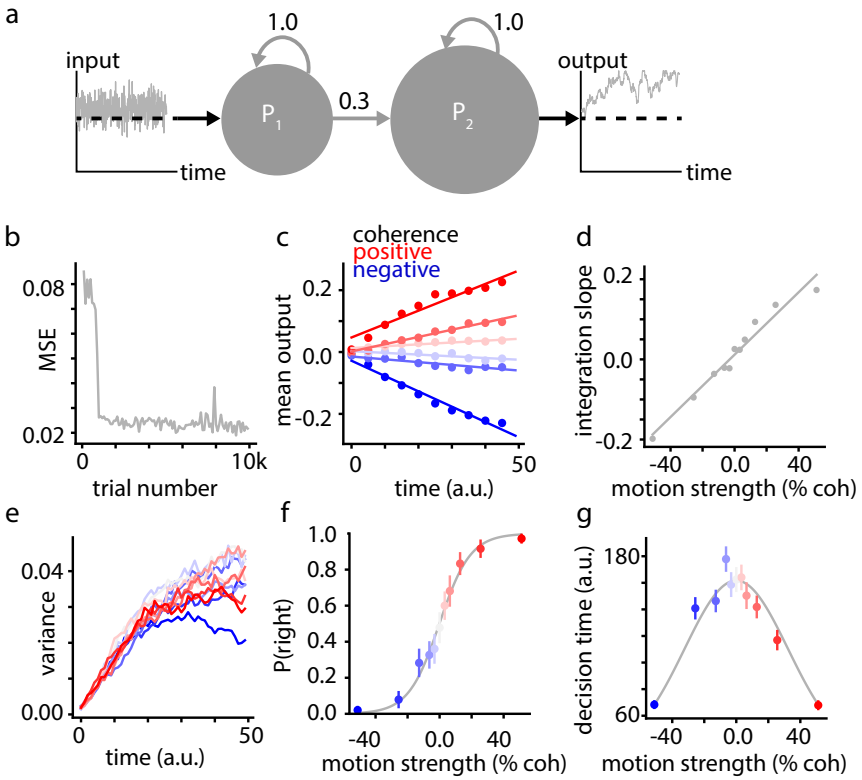

Figure 6.1: A two-stage hierarchical RNN performing linear integration of noisy inputs for a sensory decision-making task. **a.** Network schematic. **b.** Network learning throughout time in units of mean-squared error. **c)** Mean activity of the output unit after training for different stimulus strengths (motion coherence). Model outputs (solid points) increase linearly over time up to a saturating level implemented in the training procedure (see Methods). Lines are fits to the data points over the time range [0 50], measured in arbitrary network time units. **d.** The slope of changes in model output as a function of stimulus strength. **e.** Variance of model output as a function of time for different stimulus strengths. The linear increase is expected for integration of noisy inputs over time. The late saturation and decline of the variance, especially for stronger stimuli, is caused by the bound on the integration process. **f.** Psychometric function of a trained model. Data points show the probability of choosing right for different stimulus strengths ranging from strong leftward motion (negative coh) to strong rightward (positive coh). The gray curve is a logistic fit. Error bars show ±1 s.e.m. **g.** Chronometric function of a trained model. Data points show the mean time the model output takes to reach the decision bounds (decision time). The gray curve is a Gaussian function fit to the data points.

6.1.2 Behavioral effects of inactivation grow with the size of the inactivated population

We explored the effects of inactivation on this circuit by selectively silencing a proportion of neurons in the integrating population (Fig. 6.2a), and analyzing the inactivation effects on the

output of the model. For a particular trained network, we measured the change in the psychometric and chronometric functions after perturbation as a means to characterize the effects of inactivation. We found that decision times are strongly sensitive to inactivation. Weak inactivations (5-10% of the population) moderately increase the decision time of the network, and medium and strong perturbations (20% and 30% of the population, respectively) cause a much larger increase (Fig. 6.2d,g).

The effect of perturbation on choice was more variable and complex. We quantified these effects by extracting measures of the sensitivity and bias for the psychometric functions, and calculating the change in these measures with weak, medium, and strong inactivation. The sensitivity of the psychometric function decreased as more of the neurons were affected, with a corresponding decrease in the average sensitivity with inactivation size (Fig. 6.2b-c,e). The magnitude of the bias, however, minimally changed across inactivation levels (Fig. 6.2f), suggesting that the primary loss of function caused by increasing the perturbation magnitude is a loss of sensitivity. Overall, in our basic network architecture, even weak perturbations decreased sensitivity and substantially increased reaction time, with the magnitude of these effects increasing with the magnitude of inactivation.

6.1.3 Inactivation effects arise from perturbing the structure of the underlying population dynamics

The optimal solution for random dots motion discrimination involves integration along a 1-dimensional axis [Wald and Wolfowitz 1950; Shadlen et al. 2006; Drugowitsch et al. 2012; Khalvati et al. 2021], so the dynamics of the trained network are likely to lie on a low dimensional manifold [Ganguli and Sompolinsky 2012]. Indeed, simple dimensionality reduction using PCA shows that the circuit dynamics are approximately one dimensional, with the first principal axis explaining about 70% of the neural response variance (Fig. 6.3a). The low dimensional structure of the

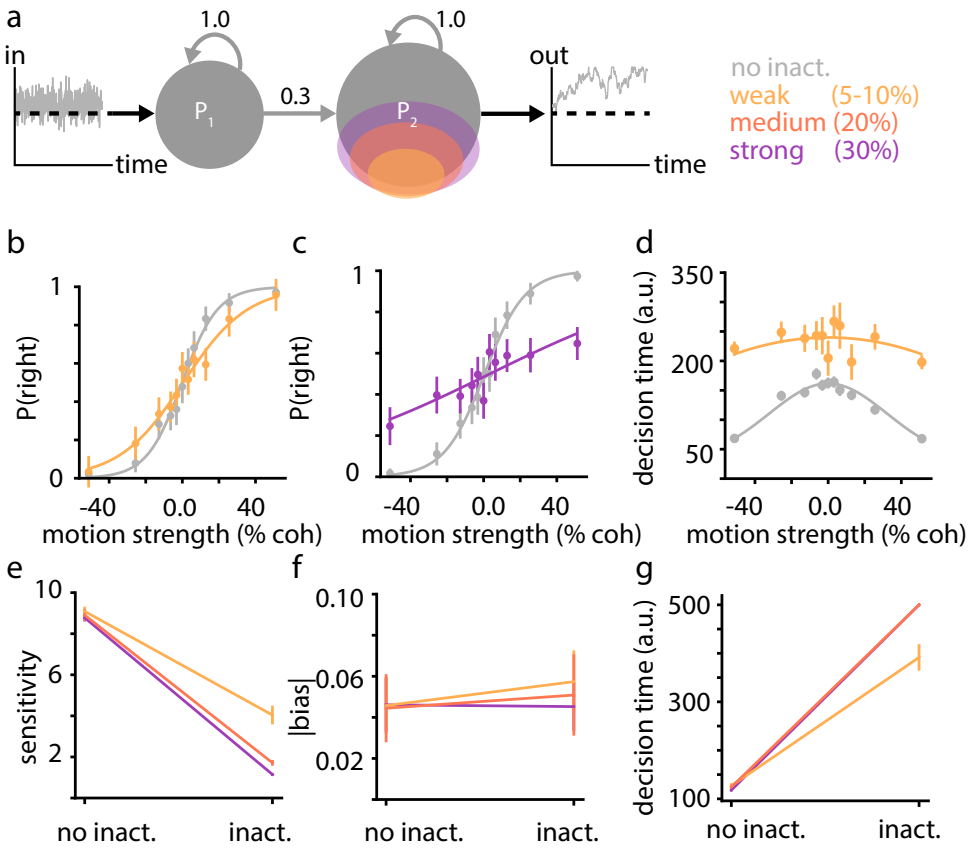

Figure 6.2: Inactivation of the integrating circuit reduces sensitivity and increases decision times, with larger effects when a larger portion of neurons are silenced. a. Inactivation schematic. Colored ovals indicate the affected proportion of the P2 network. **b-c.** Psychometric functions of example networks after training (grey) and after a weak (5-10%, **b**) or medium-size (20%, **c**) inactivation. **d.** Chronometric function of the same network in panel b following weak inactivation. **e.** Changes of sensitivity (slope of psychometric function) across 10 trained networks for various inactivation sizes. **f.** Effects of inactivation on bias (shift of the mid-point of the psychometric function from 0% coherence) across trained networks. **g.** Changes of mean decision times across trained networks. Error bars show s.e.m. Maximal trial duration set to 500 steps.

neural activity allows us to project the full network dynamics on the first principal component axis. In this space we can mathematically analyze the dynamical features of the trained network that enable it to perform evidence integration.

In the absence of a stimulus, the network activity can be approximated as one-dimensional

population dynamics of the general functional form:

$$\Delta r_t \approx (\alpha - r_{t-1}^2) r_{t-1}, \tag{6.1}$$

where Δr_t is the change in population activity across time and the value of parameter α and the constant of proportionality depend on the trained network weights (See Methods). Different settings of α change the dynamical properties of the system and its ability to solve the evidence integration task. This property is illustrated in the phase plane in Figure 6.3b, which describes the relationship between Δr_t and r_{t-1}. When α is positive, the dynamics exhibit three fixed points (corresponding to values of r_t for which Δr_t is zero; Fig. 6.3b). Two of these are attracting, separated by an unstable fixed point at $r_t = 0$: when starting from a positive value of activity r, the network will eventually converge to the positive fixed point, and similarly a negative starting condition will converge to the negative fixed point. Sensory drive to the network will push the dynamics towards one or the other, eventually converging to a final binary decision. This is similar to the phase plane of previous circuit models of evidence integration based on bistable attractor dynamics [Wong and Wang 2006].

Ideal evidence integration sums all incoming inputs, $r_t = \sum_{t=0}^{T} s_t$, and follows slightly different dynamics. In the phase plane, ideal integration means that r_t changes at every time step as

$$\Delta r_t = s_t. \tag{6.2}$$

Hence, the ideal solution for evidence integration involves a line attractor, where no change in activity occurs in the absence of input ($\Delta r_t = 0$ whenever $s_t = 0$; red line in Fig. 6.3b-d). The trained RNN approximates this solution when α is close to zero (Fig. 6.3c). As α becomes increasingly negative, the network will start to behave qualitatively differently (formally, this corresponds to a pitchfork bifurcation [Strogatz 2018], which is why we refer to α as the bifurcation criterion). In this negative regime, the network has only one stable fixed point at the origin (Fig. 6.3d). Changes

in network activity caused by new stimuli rapidly relax to this fixed point, causing the network to lose its ability to integrate.

The value of the bifurcation criterion α, which we derive directly from the RNN activity, captures the key dynamic properties of the model networks and predicts a given network's ability to perform evidence integration. Indeed, trained networks generally have a small positive α, corresponding to a shallow bi-stable attractor and close-to-ideal evidence integration (Fig. 6.3e).

Different forms of causal interventions, such as inactivation, will result in altered population dynamics, with a corresponding change in the bifurcation criterion α. In particular, our inactivation experiments push the network past the bifurcation point: as the magnitude of the inactivation increases, α values become increasingly negative (Fig. 6.3e,f). The remaining fixed point leads to forgetting past inputs and correspondingly poor performance (Fig. 6.3f).

Overall, these results establish that our network approximates integration within a bounded region of state space via a shallow bi-stable attractor, and that the loss of function caused by perturbations is due to the loss of this attractor structure. This dynamical systems analysis paints a more refined picture of causal interventions in the random dots motion discrimination task: inactivations that disrupt the computational structure embedded in the network (i.e., the bi-stable attractor) will produce behavioral impairments, while those that leave the attractor unaffected will not. While in our simple network architecture (Fig. 6.2a) all interventions disrupt the attractor structure, this may not necessarily be the case for more complex distributed networks, as we will see below.

6.1.4 In distributed architectures, inactivation effects can be variable

Exploring the effects of inactivation in a unitary circuit performing integration reveals a qualitatively similar picture across network and effect sizes. In a mammalian brain, however, sensory decisions are enabled by a distributed network consisting of multiple interacting circuits [Shadlen and Kiani 2013; Waskom et al. 2019]. Past electrophysiological studies have found neurons that

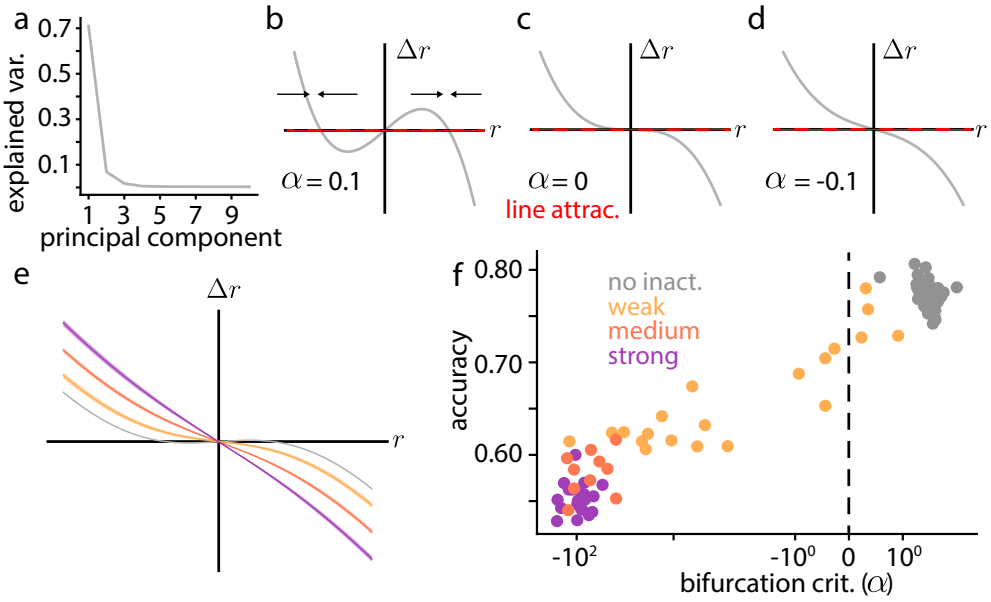

Figure 6.3: Integrating network implements a shallow bistable attractor, whose disruption determines the magnitude of the behavioral effects of inactivation. a. Fraction of explained variance as a function of the number of latent dimensions of the network responses to test stimuli. **b-d.** Schematic of a pitchfork bifurcation. The red dashed line shows the phase portrait for a line attractor implementing optimal evidence integration. For $\alpha > 0$, the network has two stable attractors (**b**, arrows indicate sign of Δr), for $\alpha = 0$, a saddle point (**c**), and for $\alpha < 0$, a single stable attractor (**d**). **e.** Phase plot for the reduced network before (gray) and after (colors) perturbation. Shaded regions indicate s.e.m. across network realizations. **f.** Fraction of correct responses as a function of the bifurcation criterion estimated for each network.

represent integration of sensory evidence in the parietal cortex [Shadlen and Newsome 2001b; Churchland et al. 2008], lateral frontal cortex [Kim and Shadlen 1999; Mante et al. 2013; Mochol et al. 2021], motor and premotor cortices [Peixoto et al. 2021; Chandrasekaran et al. 2017; Hanks et al. 2015; Thura and Cisek 2014], the basal ganglia [Ding and Gold 2013; Yartsev et al. 2018], superior colliculus [Horwitz and Newsome 1999; Basso et al. 2021], and cerebellum [Deverett et al. 2018]. The distributed nature of the computation, paired with potential circuit redundancies, make inactivation studies difficult to interpret. This is especially true when inactivation of a subcircuit in the network fails to produce measurable changes of behavior. Other nodes of the network could change their activity in responses to the inactivation, compensating for its effects [Li et al. 2016]. Furthermore, there are a variety of more complex scenarios compatible with negative

results [Yoshihara and Yoshihara 2018; Jonas and Kording 2017; Murray and Baxter 2006; Dunn 2003].

Although a detailed exploration of the distributed network that underlies decisions in the brain is beyond the scope of this paper, we take a first step in assessing the effects of architecture on inactivation experiments. In particular, we replace the unitary network structure analyzed above with a parallel architecture, where sensory inputs drive the responses of two non-interacting populations that collectively shape the network output (Fig. 6.4a). We train this parallel network to perform the same sensory integration task, followed by inactivating all of the neurons in one of the two parallel nodes and assessing the behavioral outcomes of the manipulation across a range of network instances. We find that even in this minimal version of a distributed computation the effects of inactivation can be quite variable in terms of performance.

Some networks exhibit minimal changes in the psychometric function due to inactivation (Fig. 6.4b, e), paired with a marked increase in reaction times (Fig. 6.4h). This phenomenology tracks back to the dynamical system properties of the underlying network. When examining the one-dimensional approximate phase portrait for each node in the network, we found that *both* exhibit the shallow bistable attractor dynamics indicative of approximately optimal sensory integration (Fig. 6.4c-d). The overall network output, which determines the final choice and decision time, is constructed by linearly combining the activities of both integrating populations. This architecture subsumes more specific architectures in which populations with distinct choice preference integrate evidence for their respective choices. The inactivation completely disrupts the attractor structure in the targeted sub-circuit P2 (Fig. 6.4d), but leaves the attractor in P1 intact (since they do not directly interact; Fig. 6.4c). Therefore, integration can still be performed using the intact sub-circuit. Nonetheless, the activity component from P2 is missing; as a result, the output could be weaker and it may take longer for the integrated signal to reach the same decision threshold, leading to slower responses. However, if the only measure of behavior is the choice, one may not notice any change of behavior, as evident in Fig 6.4.

A systematic investigation across networks with the same distributed architecture, but different trained connections, reveals that this inactivation-resistant solution is not universal: in some networks the sensitivity is largely unaffected, while others display a marked loss in sensitivity after inactivation (Fig. 6.4f). This variability traces back to the attractor structure of the individual solution found via learning: networks exhibit robustness to inactivation only if the unlesioned node has an attractor (Fig. 6.4e). A parallel network architecture solves the task in two ways: either both networks develop attractor dynamics, or only one does. Inactivating a network that has only one attractor disrupts performance (sensitivity, bias), indicating that the sub-circuit is in fact involved in the network computation, while inactivating the sub-circuit that does not have attractor structure leaves the output essentially unaffected. However, if both nodes have attractors, 'negative results' at the level of behavior need to be interpreted with caution, because intact nodes can enable consistent behavior even without the participation of inactivated nodes. Though we have only shown these effects for a simple network with two parallel nodes, robustness to inactivation is likely to become even more prominent in systems with many more parallel nodes. These results demonstrate that absence of change in choice behavior following inactivation is insufficient to conclude that a certain network node lacks a functional role in the task.

Overall, this analysis reveals a more complicated picture of inactivation: disabling an individual node in a network will produce a loss of function only if no other node in the network is capable of compensating for its loss. Moreover, in dynamical systems such as the RNNs we study here, redundancy and compensation exist even in very simple networks, performing very simple tasks.

6.1.5 Short periods of relearning can compensate for inactivation

A key hidden assumption in our simulated experiments in the previous sections is that no additional task-specific learning can happen prior to testing the effects of the manipulation on behavior. This assumption is unlikely to be completely true, as plasticity and reinforcement mechanisms may continue to operate. In fact, cortical inactivation studies show many behaviors are only

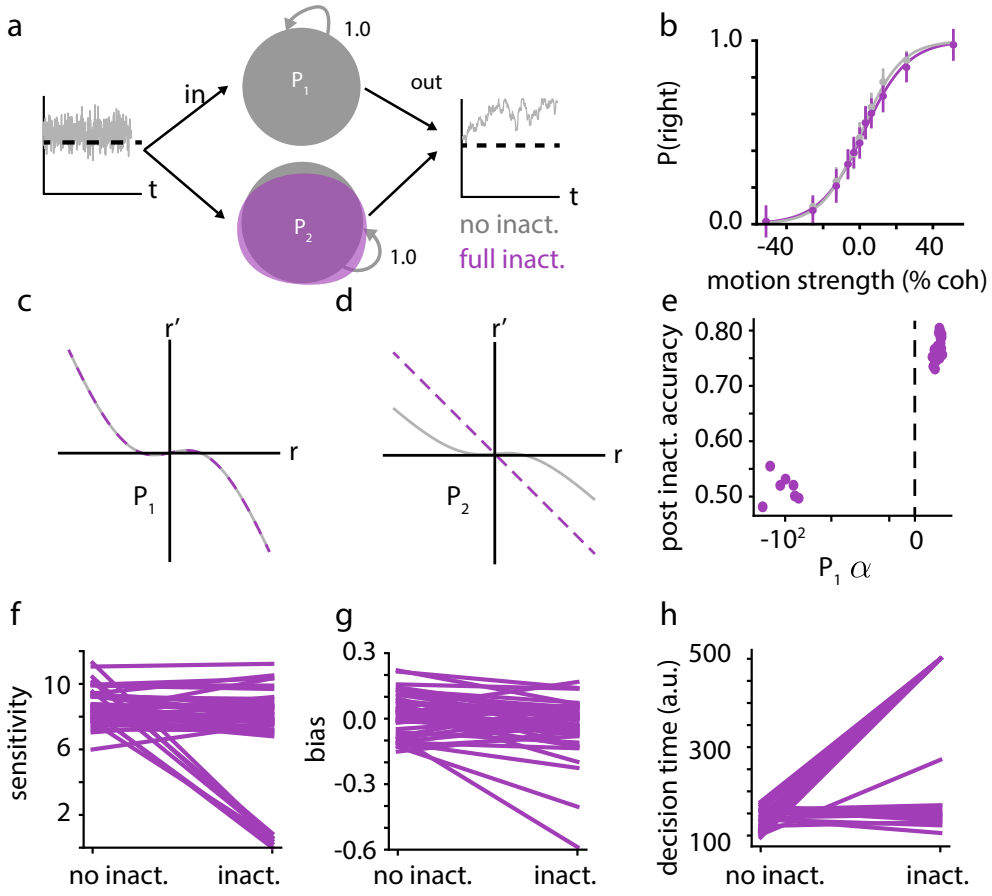

Figure 6.4: Distributing integration across multiple network nodes makes it resilient to disruptions in any single node. a. Schematic for a network with two parallel nodes (circuits) for integration computation. **b.** Psychometric curve for the parallel node network before and after a strong inactivation. The lines are logistic fits to the choices. Errors bars are s.e.m. **c.** Approximate phase portrait for the intact node, indicating a shallow bistable attractor. **d.** Phase portrait for the inactivated node. **e.** Proportion correct after inactivation as a function of the bifurcation criterion for the node that was not inactivated. **f.** Sensitivity for the psychometric function before and after a strong inactivation. Each line shows an instance of the parallel node network with unique starting points and training history. Inactivation affected sensitivity in only one of the ten instances. **g.** Bias of the psychometric function before and after inactivation. **h.** Mean reaction time before and after inactivation.

temporarily affected following the inactivation [Newsome and Pare 1988; Murray and Baxter 2006; Schiller et al. 1979; Rudolph and Pasternak 1999], a clear illustration of the brain's remarkable capacity for learning through re-organization of its circuits. Similar recoveries may be expected in less severe experimental manipulations in which neurons are transiently inactivated, but the extent of additional learning required for adaptation to occur is less clear. To investigate the

capacity of networks to adapt to inactivation and regain their performance through further task specific learning, we modified our model to allow network connections to continue to change at test time and investigated two biologically relevant variants of inactivation: long lasting and intermittent.

The first type of inactivation is implemented as a long-lasting disabling of the involved neurons, as used for all the previous sections. It is intended as an analogue of experimental manipulations using muscimol or other pharmacological agents, or designer receptors exclusively activated by designer drugs (DREADDs) [Wiegert et al. 2017], which affect target circuits for many minutes to days. In these cases, since the inactivation lasts for the majority of an experimental session (or multiple experimental sessions), circuits could eventually learn to compensate for the perturbation with sufficient additional task experience. What is remarkable in the context of the model is how little additional training is required.

Depending on the extent of manipulation, a few hundreds of trials were sufficient to compensate for the inactivation, much fewer than the number of trials required for the initial training of the network. Fig. 6.5a shows an example run where inactivation of 30% of the integrating population transiently caused the network to initially perform as poorly as it did before learning. To describe the trajectory of re-learning across networks, we measured the percentage of correct responses as a function of the number of retraining trials, for the same size of inactivation (30%). We found that the circuit robustly reached pre-inactivation performance with fewer than 500 retraining trials (Fig. 6.5b). This return to pre-inactivation performance was also mimicked in the underlying bi-stable attractor, with the bifurcation criterion α returning to positive values on the same time scale (Fig. 6.5c), indicating that the network has reconstructed its shallow bi-stable attractor.

To directly visualize the impact of inactivation and relearning along the first axis of network variance, we compared the projection of the network activity onto its first principal component at the end of training (Fig. 6.5d), just after inactivation (Fig. 6.5e), and after relearning (Fig 6.5f; PCA performed separately on the neural activity at different time points). These show that the

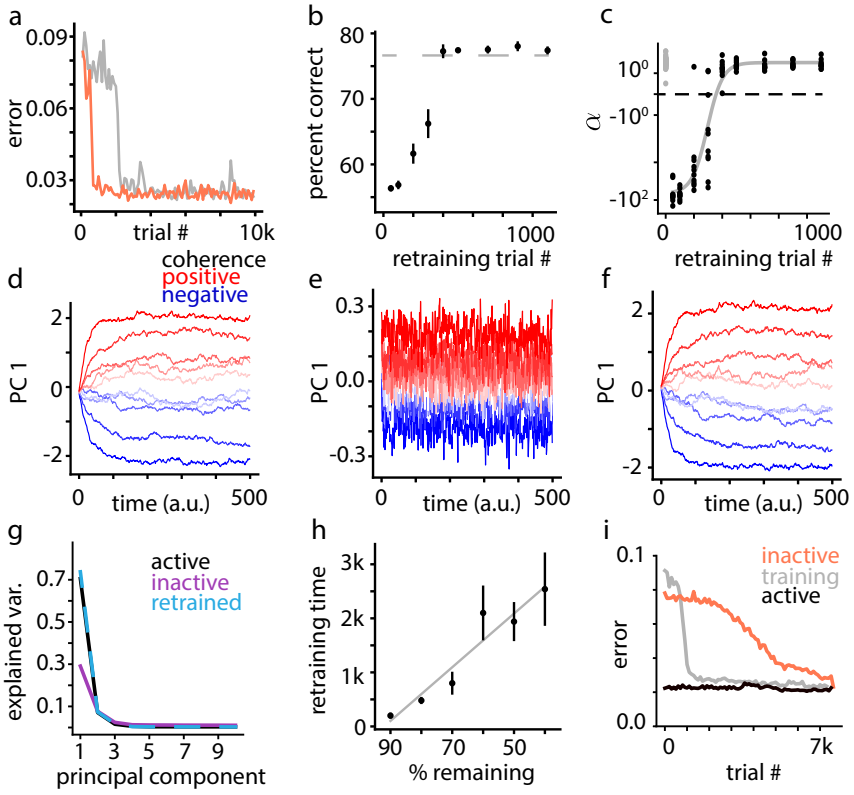

Figure 6.5: Perturbed networks can learn to compensate for inactivation but the speed of recovery depends on the timescale of inactivation. a. Post-inactivation training can be much faster than the initial training. The lines show changes in the MSE error of the hierarchical network of Fig. 6.1a during the initial training (gray) and during training after inactivation of 30% of P2 neurons. **b.** Percent correct as a function of retraining trials. The gray dashed line indicates the average network performance after initial training. **c.** The stability criterion as a function of the number of retraining trials. Data points show 10 different networks before inactivation (gray) and after inactivation of 30% of P2 neurons (black). **d.** Activity projected onto the first principal component for the network prior to inactivation. **e.** Activity projected onto the first principal component for the network after inactivation of 30% of P2 neurons. **f.** Activity projected on the first principal component for the network after post-inactivation retraining. **g.** Fraction explained variance as a function of the number of latent dimensions of the network responses after learning (black), after inactivation (purple), and after retraining (blue dashes). **h.** Mean retraining time as a function of the percentage of unperturbed neurons in the network. **i.** Retraining slows down considerably if inactivation of neurons occurs on a fast timescale that allows mixing perturbed and unperturbed trials during retraining. In this simulation, inactivation was limited to a random half of training trials. Integration error goes down slowly for perturbed trials (orange); here, trial # indicates only perturbed trials, not interleaved unperturbed trials. For the unperturbed trials (black), integration error remains close to the levels following the initial training (gray).

integration properties of the network largely collapse (although not completely) immediately after inactivation, and are fully and quickly restored by relearning. The relearning speed — time to

reach virtually the same accuracy as the pre-inactivation network (within 0.5%) — is strongly correlated with the extent of inactivation: the larger the inactivated population, the longer it takes for function to be recovered by retraining (Fig. 6.5h). In our simulated network, inactivations as large as up to 30% of the neurons in the population still exhibited significantly faster relearning compared to the initial training time. This suggests there may be cases where compensation happens on the scale of one or a few sessions, similar to what an experimenter may use to assess the effects of the manipulation on behavior, potentially confounding these results.

The advent of optogenetics allows controlling the activity of neurons with millisecond resolution, leading to new experiments which interleave perturbed trials with unperturbed ones. These techniques are commonly considered the gold standard for causal manipulations as they offer millisecond temporal precision and enable targeting specific cell types. This improved precision and specificity is quite beneficial but does not remove the re-learning challenge mentioned above. The intact vs. the inactivated network can be thought of as two distinct dynamic states of the circuit. Repeated inactivation of largely the same group of neurons in a circuit, as in most optogenetic experiments, can provide opportunity for compensation even when inactivation is infrequent. Biological circuits could learn to use the silence of inactivated neurons as a contextual cue to switch behavior, or could redirect the computation in both states to the neurons that are not being directly manipulated.

To model an intermittent inactivation scenario similar to optogenetic manipulation experiments, we inactivated the network on a random subset (50%) of training trials, instead of tonically inactivating all neurons throughout retraining. In alignment to general intuitions that adaptation is less likely during transient inactivation, we found that it takes the network more inactivation trials to re-learn when inactivation is transient and infrequent (Fig. 6.5i). When neurons are only inactivated on 50% of re-training trials, it takes our network longer than its initial training time to compensate. This implies that transient inactivation techniques are likely more effective against inactivation-induced adaptation in biological networks, although compensation is still possible.

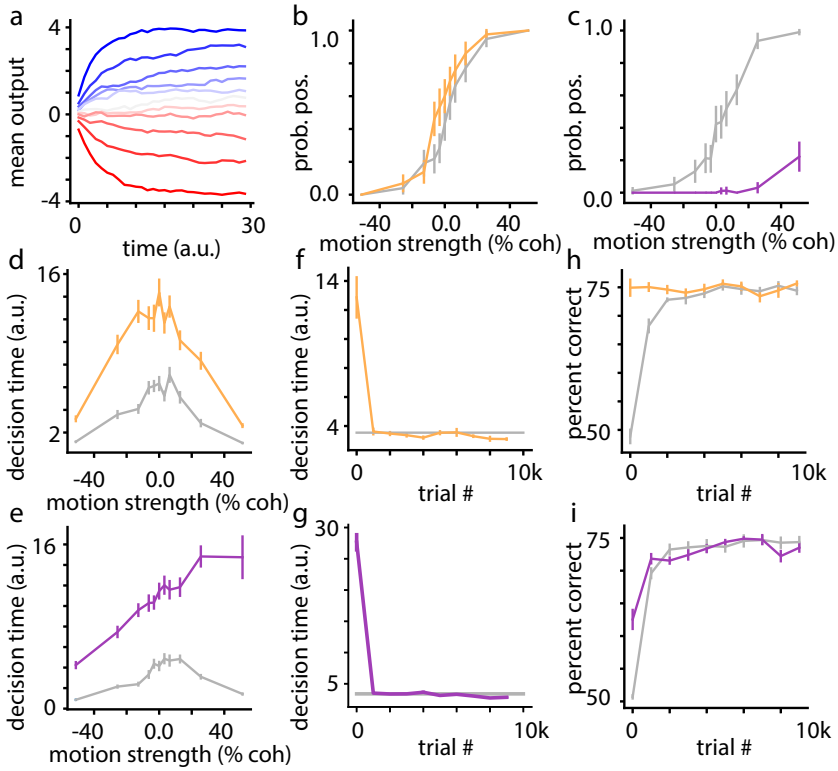

Figure 6.6: Inactivation and relearning analysis for a network trained with biologically-plausible learning. a. Mean output throughout time, stratified by coherence. **b.** Psychometric function after training (grey), and after a 40% inactivation (yellow). **c.** Same as **b**, but for a 75% inactivation (purple). **d.** Chronometric function after training (grey) and after a 40% inactivation (yellow). **e.** Same as **d**, but for a 75% inactivation (purple). **f.** Mean decision time for a 40% inactivation throughout retraining (yellow), compared to the asymptotic mean decision time prior to inactivation (grey). **g.** Same as **f**, but for a 75% inactivation (purple). **h.** Percent correct for a 40% inactivation throughout retraining (yellow), compared to the percent correct through the original training. **i.** Same as **h**, but for a 75% inactivation (purple). Bars indicate ±1 s.e.m. across 1000 test trials for **b**, **c**, **d**, and **e**, and ±1 s.e.m. across 10 simulated networks for **f**, **g**, **h**, and **i**.

A possible criticism when interpreting these re-learning results is that the optimization procedure used for learning is not biologically realistic, and that the dynamics of re-learning might look very different when the network connections adapt via local synaptic plasticity rules. To assess the generality of our results, we trained the network using Random Feedback Local Online (RFLO) learning, a biologically-plausible alternative to backpropagation through time [Murray 2019]. We also replaced mean-squared error with a loss based on binary decision outcomes, as a more realistic feedback signal to the network (see Methods). In Fig. 6.6b-g we repeat the training

and inactivation experiments for different inactivation sizes in this new model. We find that the qualitative features of the network solution and the post-inactivation loss of function match those shown in Fig. 6.2. In particular, the network is learning bounded integration (Fig. 6.6a), with a moderate loss-of-function in the psychometric and chronometric functions for a 40% inactivation, and near-total loss-of-function for a 75% inactivation. As the network continues to learn after inactivation, it is able to restore its mean decision time and performance much more rapidly than the original training time for both the 40% (Fig. 6.6f,h) and 75% (Fig. 6.6g,i) inactivations, suggesting that local synaptic plasticity can also support fast recovery of function after partial circuit inactivation.

6.2 Methods

Cognitive functions depend on interactions of neurons in large, recurrent networks. To explore the utility and limitations of inactivation and lesion studies for discovering the flow of information and causal interactions in these networks, we simulated recurrent neural network (RNN) models with different degrees of complexity and selectively inactivated sub-populations of neurons within the simulated networks. The models were trained to perform simple perceptual decisions, commonly used for investigating cortical and subcortical neural responses and their causal contributions to behavior [Katz et al. 2016; Fetsch et al. 2018; Zhou and Freedman 2019b; Hanks et al. 2015]. Our simulations and theoretical exploration focus on the direction discrimination task with random dots [Newsome and Pare 1988; Roitman and Shadlen 2002] as a canonical example of perceptual decision-making tasks.

6.2.1 Implementation of RNNs

We simulated an RNN performing a random dots task. To ensure convergence to an optimal set of weight parameters, we trained the RNNs in PyTorch [Paszke et al. 2019] using backpropagation

through time (BPTT) and Adam [Kingma and Ba 2014] with a learning rate of 2×10^{-6}. Each network was trained over 25,000 trials and tested on a separate group of 1500 trials for investigating network computations, task performance, and susceptibility to various activity perturbations. The time steps can be mapped to physical units of time (e.g., 10 milliseconds), but we avoid doing so as our conclusions are invariant to the exact definition of time steps. A univariate input sampled from a Gaussian distribution, $s \sim \mathrm{N}(kC, 1)$, was applied at each time step. C is the motion strength (coherence), and k is a sensitivity parameter that translates motion strength to sensory evidence. The variance of the input evidence to the network was set to 1. In our simulations, the sensitivity was $k = 0.4$, and C was randomly drawn on each trial from a discrete set: [-0.512, -0.256, -0.128, -0.064, -0.032, 0, 0.032, 0.064, 0.128, 0.256, 0.512]. Positive and negative motion strengths indicate rightward and leftward directions, respectively. The network was trained to discriminate the two motion directions based on input evidence, as explained below.

Independent normal noise (η) was injected into each neuron at each time step with variance 0.01. These variables combined give the following update equation for the RNN:

$$\mathbf{r}(t) = f(\mathbf{W}\mathbf{r}(t-1) + \mathbf{W}^{in}\mathbf{s}(t) + \boldsymbol{\eta}(t)), \tag{6.3}$$

where $\mathbf{W}$ is the recurrent weight matrix, $\mathbf{W}^{in}$ is the input weight matrix, and $f(\cdot)$ is the tanh nonlinearity. The network output is given by:

$$o(t) = \mathbf{D}\mathbf{r}(t), \tag{6.4}$$

where $\mathbf{D}$ is a $1 \times N$ linear decoder, and N is the number of neurons in the network.

We trained the network to integrate these inputs through time, setting our loss to the mean-squared error (MSE) between the network output and an integrated decision variable (DV) given

by:

$$DV(t) = \min\left(\max\left(\sum_{k=0}^{t} as_k, -B\right), B\right), \tag{6.5}$$

with proportionality constant $a = 0.025$ to keep the integrated variable within the dynamic range of the RNN, and $B = 0.5$ giving the bounds on integration. This results in the loss $\mathcal{L}$ is given by:

$$\mathcal{L} = \sum_{t=0}^{T} (DV(t) - o(t))^2 \tag{6.6}$$

The nonlinearity in Eq. 6.5 limits the dynamic range of the DV and ensures accurate representation of low-magnitude DVs in the small pool of neurons in the RNN model. Since the majority of motion strengths in our simulations are weak, accurate representation of low-magnitude DVs are crucial for task performance.

During training, the length of each trial was selected randomly from an exponential distribution, $T \sim 100 + \text{exprand}(200)$, with a maximum duration of 500 time steps. During testing, we used the maximum stimulus duration to ensure all trials terminated by reaching the decision bounds, enabling us to determine both choice and decision time on each trial.

6.2.2 Simple circuits

To begin, we trained a two-population network, where the first population (P_1) receives the evidence input, $s(t)$, and a linear decoder (Eq. 6.4) reads out the integrated input from the second population ($P2$). Connections between the two populations are feedforward, with each feedforward connection having a probability of 0.3 of being nonzero, but connections within each population are all-to-all, as shown in Fig. 6.1a For the sake of simplicity, there were no feedback connections from P_2 to P_1 (connection probabilities are shown in Fig. 6.1a). The first population had 30 neurons, and the second had 60. These two populations fulfill the roles of a low-level sensory population that relays input information, and a higher-order population that integrates the information for

a decision. This network is much simpler than the circuit that underlies sensory decisions in a mammalian brain, where motion selective sensory neurons (e.g., MT neurons in the primate brain) pass information about the sensory stimulus [Newsome and Pare 1988; Salzman et al. 1990; Britten et al. 1992] to a large network of association cortex, motor cortex, and subcortical areas that form the decision [Horwitz and Newsome 1999; Ding and Gold 2013; Roitman and Shadlen 2002; Mante et al. 2013; Kiani et al. 2014a; Kim and Shadlen 1999]. However, our simple circuit lends itself to mathematical analysis, can be trained without adding structural complications, and can be used for systematic exploration of inactivation effects.

To emulate lesion or inactivation experiments, we selectively inactivated a fixed group of neurons in the network. In lesion or long-term inactivation experiments (e.g., muscimol injections or DREADS) the connections remained affected throughout all trials in a testing block. In inactivation experiments with fast timescales (e.g., optogenetic perturbations), the connections were affected for a random subset of trials intermixed with other trials in which all connections and neurons functioned normally.

We systematically varied the proportion of affected neurons in the population in distinct simulations. A weak perturbation affected 5-10% of neurons, a medium-strength perturbation 20% of neurons, and a strong perturbation 30% of neurons (Fig.6.1a).

6.2.3 Complex circuits

In the distributed network that subserves perceptual decision-making, multiple circuits could operate in parallel, performing similar operations. To investigate the impact of this possibility on our inactivation results, we organized the network into two unconnected populations, each with 30 neurons. The connection probabilities are given in Fig. 6.4a. We totally inactivated one of the two sub-populations and analyzed the effect on network responses. Note that our simulation is not meant to capture the full complexity of the equivalent brain circuits but rather to offer a minimalist design that captures a key complexity commonly observed in brain networks: parallel

processing of sensory information in a variety of frontal and parietal cortical regions (e.g., lateral intraparietal, frontal eye fields, and lateral and medial prefrontal areas of the monkey brain).

6.2.4 Analysis of neural responses

After training, we applied several analyses to characterize the nature of the network computations and effects of perturbations. Because there is not an explicit reaction time in our training framework, we set symmetric decision boundaries on the network output $o(t)$ as a proxy for reaction time. We quantified the time until $o(t)$ reached one of the boundaries on each trial. The crossed boundary dictated the choice on the trial and the time to bound determined the decision time. Formally, the reaction time is given by:

$$RT = \underset{t}{\operatorname{argmin}}\, o(t) \quad \text{s.t.} \quad |o(t)| > 0.4 \tag{6.7}$$

and the choice is given by:

$$choice = \operatorname{sign}(o(RT)). \tag{6.8}$$

For each trained RNN, we constructed a psychometric function by measuring the proportion of 'left' and 'right' motion choices, and we fit the psychometric function using the following logistic regression:

$$p(right) = \frac{1}{1 + \exp(b_0 + b_1 C)}, \tag{6.9}$$

where $p(right)$ is the proportion of 'right' choices, and b_i are regression coefficients. b_0 reflects the choice bias, and b_1 the sensitivity of choices to changes in motion strength.

We constructed chronometric functions (Fig. 6.2c) by stratifying the mean decision times as a function of motion strength. For simplicity, we fit the chronometric functions with nonlinear regression using the following bell-shaped function:

$$RT = b_0 + b_1 \exp(-(C/b_2)^2), \tag{6.10}$$

where RT is the network's decision time.

To explore the dynamics of neural responses, we performed PCA on the network activity over time and trials. We found that the majority of population response fluctuations lie within a single dimension (See Fig. 6.1h). We analyzed neural trajectories associated with each choice by averaging neural firing rates within the output population for choices to the left, for each motion strength. A perfect integrator would have a mean response linearly increasing through time, and the slope of that linear increase would vary linearly with changes in motion strength [Shadlen et al. 2006]. To verify this, we fit the mean output (o_t) through time by linear regression (Fig. 6.1c), and plotted the slope of this fit as a function of coherence (Fig. 6.1d). Further, a perfect integrator would show a linear increase of variance over time. We measured the variability of the network responses at each time point and for each motion strength by quantifying the empirical variance of the network output across test trials with the same motion strength.

6.2.5 One-dimensional approximate dynamics and pitchfork bifurcation

Given the low dimensional structure of the trained RNN dynamics, we can provide a one-dimensional approximation of our RNN by projecting the network dynamics along its first principal component. Let $\mathbf{V}$ be eigenvectors corresponding to distinct eigenvalues of the covariance matrix of the network activity, obtained by combining trials across time and stimuli. These normalized vectors define an orthonormal basis, with the first and n-th axis corresponding to the direction of maximum and minimum variance, respectively. The activity of the network in this rotated coordinate system becomes $\mathbf{r}_{rot}(t) = \mathbf{V}^\top \mathbf{r}(\mathbf{t})$. Using Eq. 6.3, this leads to dynamics:

$$\mathbf{V}^\top \mathbf{r}(t) = \mathbf{V}^\top f\left(\mathbf{W}\mathbf{V}\mathbf{V}^\top \mathbf{r}(t-1) + \mathbf{W}^{in}\mathbf{s}(t) + \boldsymbol{\eta}(t)\right), \tag{6.11}$$

where we have used the fact that the basis is orthonormal, i.e. $\mathbf{V}\mathbf{V}^\top = \mathbf{I}$. Substituting our definition for $\mathbf{r}_{rot}$, we have:

$$\mathbf{r}_{rot}(t) = \mathbf{V}^\top f\left(\mathbf{W}\mathbf{V}\mathbf{r}_{rot}(t-1) + \mathbf{W}^{in}\mathbf{s}(t) + \boldsymbol{\eta}\right). \tag{6.12}$$

Focusing on the first dimension, along the axis of maximum variance, yields:

$$r^1_{rot}(t) = \mathbf{v}_1^\top f\left(\sum_{k=1}^{n} r^k_{rot}(t-1)\mathbf{W}\mathbf{v}_k + ((\mathbf{W}^{in})^T\mathbf{s}(t))_1 + \eta_1\right). \tag{6.13}$$

where $\mathbf{v}_k$ denotes the k-th eigenvector, and $r^k_{rot}(t)$ is the k-th entry of $\mathbf{r}_{rot}(t)$. Assuming that the system is largely one-dimensional, the expression for the dynamics can be further simplified as:

$$r^1_{rot}(t) \approx \mathbf{v}_1^\top f\left(r^1_{rot}(t-1)\mathbf{W}\mathbf{v}_1 + ((\mathbf{W}^{in})^T\mathbf{s}(t))_1 + \eta_1\right). \tag{6.14}$$

This approximation effectively discards the contribution of the remaining dimensions, under the assumption that their effect on the network dynamics is minimal, i.e. $\mathrm{E}\left[\sum_{k=2}^{N} r^k_{rot}(t-1)\mathbf{W}\mathbf{v}_k\right] \approx 0$, which holds empirically for our trained networks. In Fig. 6.8, we demonstrate for a trained network that the approximate dynamics closely match the true neural dynamics along its highest variance dimension r^1_{rot}.

Having derived a one-dimensional dynamical system approximation to the RNN activity, we can use phase-plane methods to determine the nature of the learned dynamics. We are interested in the geometry of the solution our network finds, which leads us to assess its fixed point dynamics in the absence of input and noise ($\mathbf{s}(t) = 0, \eta_1(t) = 0$). Finding these fixed points involves finding the solutions of equation:

$$\Delta r_{rot} = \mathbf{v}_1^\top f\left(r_{rot}(t-1)\mathbf{W}\mathbf{v}_1\right) - \mathbf{v}_1^\top r_{rot}(t-1)\mathbf{v}_1 = 0, \tag{6.15}$$

where $\Delta r_{rot} = r_{rot}(t) - r_{rot}(t-1)$, and we have used the fact that the eigenvectors are normalized,

i.e. $\mathbf{v}_1^\top \mathbf{v}_1 = 1$. Further, using a Taylor approximation of tanh about 0, $f(x) \approx x - x^3$, and rearranging the terms simplifies the equation to:

$$\begin{aligned} \Delta r_{rot} &\approx \gamma r_{rot}(t-1) - \beta r_{rot}(t-1)^3 \\ &\propto \gamma/\beta r_{rot}(t-1) - r_{rot}(t-1)^3, \end{aligned} \tag{6.16}$$

where $\gamma = \mathbf{v}_1^\top(\mathbf{W} - \mathbf{I})\mathbf{v}_1)$, $\beta = \mathbf{v}_1^\top (\mathbf{W}\mathbf{v}_1)^{\circ 3}$ is empirically positive, and $(\cdot)^{\circ 3}$ denotes an element-wise cube. The resulting equation is cubic, meaning its fixed point equation ($\Delta r(t) = 0$) has up to 3 solutions. This generally results in a topology with two stable fixed points separated by one unstable fixed point. These points coalesce into a single stable fixed point, $r_{rot}(t) = 0$, when the coefficient of $r_{rot}(t-1)$ changes from positive to negative, with the system undergoing a *supercritical pitchfork bifurcation* [Strogatz 2018].

For the network to work properly, it needs to be in the regime with two stable attractors, with an abrupt degradation once reaching the critical point for the phase transition. For our approximate dynamics, this transition occurs once:

$$\gamma/\beta < 0. \tag{6.17}$$

For this reason, for all of our experiments, we refer to the value $\alpha = \gamma/(\epsilon + \beta)$ as the bifurcation criterion, where we have included $\epsilon = 5 \times 10^{-3}$ in the denominator to prevent ill conditioning caused by dividing by β values close to zero. Though the exact point of transition from one fixed point to two may vary due to our approximations, our results rest on identifying regimes in which $\alpha \approx 0$ or where $\alpha \ll 0$, which are well identified by our approximation.

6.2.6 Re-learning with feedback after perturbation

To investigate whether perturbations have a lasting impact on the performance of a network with plastic neurons, we permitted the simplified hierarchical network in Fig. 6.1a to be trained following inactivation. Two training regimes were used to emulate different experimental techniques with slow and fast timescales for inactivation. In both regimes, we silenced a fraction of neurons in the P2 population and allowed the connection weights of the remaining neurons to change through relearning. The first retraining regime was designed to emulate lesion and pharmacological inactivation studies, which affect the circuit for an extended time, ranging from a whole experimental session to permanent. In this regime, the affected neurons remained inactive throughout the retraining period. The second regime was designed to emulate optogenetic or other techniques with faster timescales, which allow interleaving perturbations with unperturbed trials. In this regime, we silenced the affected neurons in a random half of retraining trials and allowed them to function in the other half; synapses were modified in all trials.

To assess the efficacy of retraining in restoring the network performance, we used the state of synapses at various times during retraining to simulate 1500 test trials and calculate the percentage of correct responses. Connection weights were kept constant in the test trials. Additionally, we calculated the projection of the network activation onto its first principal component following the initial training, after the inactivation and prior to retraining, and at various times during retraining. Finally, we calculated the stability criterion (Eq. 6.17).

6.2.7 Biologically plausible learning

It is highly unlikely that a neural system could receive detailed feedback about the difference between a decision variable and an integrated target trajectory. There is no supervised signal for this target trajectory, and if a neural system was able to construct the target, why not use it to solve the task instead? Instead, an animal is much more likely to use reward feedback that it

receives about its classification. To verify that our results hold in this situation, we adapted our training to use a cross-entropy loss function:

$$\mathcal{L}(t) = -c \log(\hat{c}_t) - (1 - c) \log(1 - \hat{c}(t)), \tag{6.18}$$

where $c = \text{sign}(C)$, and $\hat{c} = \sigma(o(t))$, where $\sigma(\cdot)$ is a sigmoid nonlinearity.

For the simulations here, we evaluated the loss at every time step, though we achieved qualitatively similar results with only end-of-trial evaluation.

BPTT is well-established as a biologically implausible learning algorithm [Werbos 1990]. Many studies have constructed approximations or alternative formulations of BPTT that are biologically plausible, but these algorithms often have different stability properties or biases [Marschall et al. 2020], or are not guaranteed convergence to the same solution (or convergence at all). To verify that our results still hold for a biologically plausible learning algorithm, we selected RFLO [Murray 2019], where recurrent weight updates are given by:

$$\begin{aligned}
\Delta w_{ij}(t) =& -\lambda \frac{d\mathcal{L}(t)}{dr_i(t)} e^w_{ij}(t) \\
e^w_{ij}(t) =& f'(\mathbf{W}\mathbf{r}(t-1) + \mathbf{W}^{in}\mathbf{s}(t) + \eta) \left(w_{ii} e^w_{ij}(t-1) + r_j(t-1) \right),
\end{aligned}$$

where $\lambda = 0.001$ is the learning rate, and where the second equation is an 'eligibility trace', which is updated continuously at each synapse, and requires only information available at the pre- and post-synapse.

This has the form of a three-factor plasticity rule [Frémaux and Gerstner 2016], where a reward signal ($\frac{d\mathcal{L}(t)}{dr_i(t)}$) is fed back and combined with pre-synaptic and post-synaptic Hebbian coactivation to produce the weight update. In our case, we allowed the feedback weights to be given by direct differentiation of the objective function, but for added biological realism, these weights could be

learned [Akrout et al. 2019] or random [Murray 2019; Lillicrap et al. 2016] and still achieve good performance.

The updates for the input weights are analogous:

$$\Delta w_{ij}^{in}(t) = -\lambda \frac{d\mathcal{L}(t)}{dr_i(t)} e_{ij}^{in}(t)$$
$$e_{ij}^{in}(t) = f'(\mathbf{W}\mathbf{r}(t-1) + \mathbf{W}^{in}\mathbf{s}(t) + \eta)\left(w_{ii} e_{ij}^{in}(t-1) + s_j(t)\right),$$

and the updates for the decoder are simply given by:

$$\Delta D_{1j}(t) = -\lambda \frac{d\mathcal{L}(t)}{dD_{1j}}.$$

For the sake of computational efficiency, for these simulations we decreased the number of time steps to 30 steps per trial with a fixed duration (10,000 trials), and increased the signal-to-noise ratio of individual stimuli by taking $s \sim \mathcal{N}(kC, 0.1)$ for $k = 0.4$. Further, we gave our network only one population of neurons ($N = 60$) with all-to-all connectivity. We also trained our networks with a larger amount of intrinsic noise ($\sigma = 0.6$) to verify that our results hold for noisier neurons.

Because these simulations had modified parameters and a different objective function, we had to reset our decision threshold to achieve qualitatively similar psychometric functions. We set the threshold for decisions in these simulations to 1: we arrived at this value by requiring near-perfect choice accuracy for strong coherence stimuli, and a chronometric function whose mean response times peaks at 0 coherence. These features clearly need not be achievable for *any* coherence if the task has not been well-learned, but we found in practice that a threshold value of 1 gives psychometric and chronometric functions similar to experimental data.

Because our networks had a different number of neurons and a different objective function, we also recalibrated the magnitudes of our inactivations. Our 'weak' inactivation in these simulations

targeted 40% of neurons, and our 'strong' inactivation targeted 75% of neurons. The method of performing our inactivation was identical to the previous section.

6.3 Discussion

A main quest of neuroscience is to identify the neural circuits and computations that underlie cognition and behavior. A common approach to achieve this goal is to first use correlational studies (e.g., electrophysiological recordings) to identify the circuits whose activity co-fluctuates with task variables (e.g., stimulus or choice), and then perturb those circuits one by one as subjects perform the task. Loss of task accuracy following lesions or transient inactivation of a circuit is commonly interpreted as evidence that the circuit is "necessary" for the underlying computations. The converse, however, need not be true. Of course, if the inactivated circuit is not involved, the behavior remains unaffected. But negative results can also arise because of other reasons, which challenge the impulse of embracing the null hypothesis.

The conclusion that negative results in perturbation experiments are not readily interpretable is not new. In fact, it is common knowledge in statistics that inability to reject a null hypothesis (e.g., circuit X is not involved in function Y) is not evidence that the null hypothesis is correct, especially if the true causes of the results remain unexplored and not included in hypothesis testing. In practice, however, there is a growing abundance of publications that interpret negative results of perturbation experiments as lack of involvement of a circuit in a mental function. In many cases, experimenters perceive their negative results as important because they seem to contradict established theories (e.g., role of posterior parietal cortex in perceptual decisions [Katz et al. 2016]). Our results reveal key challenges often ignored in the interpretation of negative results: they can emerge from robustness to perturbation due to the architecture of the affected circuit or the bigger network that the circuit belongs too; circuits may continue to learn, at a much faster scale than we tend to expect.

Our results emphasize the need for further exploration following negative results and point at exciting directions for followup experiments. There is already evidence that circuits adapt to both the transient inactivations [Jeurissen et al. 2021; Fetsch et al. 2018] and permanent lesions—for example the brain's impressive robustness to extensive and gradual dopamine neuron loss in Parkinson's disease [Zigmond et al. 1990]. However, we do not know which brain circuits adapt to perturbations, or under what experimental conditions. Answering this key question would be aided if experiments document behavior from the very first administration of the perturbation protocols and devise methods to quantify different learning opportunities inside and outside of the experimental context. It is also essential to know about the larger network engaged in a task and the parallel pathways that could mediate behavior. Association cortex, for example, often includes recurrent and diverse connections, which give rise to many possible parallel pathways. Such complex networks often prevent straightforward conclusions from a simple experimental approach that perturbs a single region in the network. More elaborate experimental designs and carefully developed computational models aid overcoming this complexity. We recommend quantification of behavior not be limited to choice (a discrete measure) and include more sensitive, analog measures, such as reaction time (Results 6.1.4), which was affected in our distributed network inactivation simulations, even when accuracy was seemingly unaffected. We also see strong advantage in augmenting single region perturbations with simultaneous perturbation of a collection of network nodes, chosen based on network models. Another valuable approach is to simultaneously record unperturbed network nodes to quantify the effects of perturbation on brain-wide response dynamics and identify adaptive mechanisms that could rescue behavior [Li et al. 2016]. From our perspective, negative behavioral results in a perturbation experiment are not the end point of the experiment. Rather, they are just a step toward a deeper understanding of the neural mechanisms that shape the behavior.

Our models in this paper focus on a well-studied perceptual decision-making task: direction discrimination with random dots. Understanding the dynamical mechanisms of computation in

our circuits proved necessary for understanding their response to inactivation. Trained networks implement an approximation of the drift-diffusion model, and exhibit low-dimensional integration dynamics in response to sensory stimuli. We characterized the learned solutions by their phase portrait properties, and found that networks approximated sensory integration using a shallow bistable attractor [Wong and Wang 2006; Strogatz 2018], whose disruption was closely correlated with loss-of-function in inactivated networks; further, relearning reconstructed a bistable attractor within the network. Our analysis shows that what matters in a circuit, irrespective of the status of individual neurons, is the integrity of its computational structure. This makes statistical approaches that aim to extract this structure directly from measured population responses particularly valuable [Zhao and Park 2016; Nassar et al. 2018; Duncker et al. 2019].

Though different tasks and network structures will likely lead to variations in the nature of learned solutions and responses to inactivation, in many respects the random dots motion discrimination task and our neural network architectures serve as a microcosm of a more general phenomenon, which encompasses both artificial and biological systems. Results in the recurrent neural network literature have shown that significant variations in the response properties of individual network units (vanilla RNNs, GRUs, or LSTMs) tend to produce similar canonical solutions to simple decision-making tasks, embedded in a low-dimensional subspace of the network's dynamics [Maheswaranathan et al. 2019]. The exact mechanism of loss-of-function in response to inactivation may differ, but our expectation is that independent of architecture, any trained system–including those used by biological systems–will learn to approximate the optimal canonical computations (here, integration) required for the task. We examined only the simplest possible implementation of parallelism in our network architecture, and showed that even this was sufficient to greatly increase the system's robustness to neural inactivation. Real neural circuits likely show this phenomenon on a much larger scale by virtue of involving far more neurons, with more natural redundancy.

The particular learning algorithm that drives the organization of the circuit is not important

for the observed effects: a biologically-plausible learning algorithm [Murray 2019], with more realistic feedback, learned a similar computational structure, and showed similar inactivation and relearning effects to brute-force optimization via backpropagation through time. It is likely that *any* algorithm that closely aligns with gradient descent on a task-specific loss will produce similar effects.

Though we found that the simplest network architectures were in general not robust to inactivations, many basic architectural modifications could produce resistance to perturbations. In particular, redundancy of function (two parallel, independent attractors) embedded within the network produced inactivation resistance; this suggests that multi-area recordings are important for assessing whether the effects of inactivation have been compensated for by another neural population. Further, in our distributed circuit, inactivation still affected reaction time, suggesting that reaction time or other analog aspects of behavior may be more sensitive than choice to inactivation effects.

Furthermore, we found that even if neural circuits play a direct causal role in a computation, loss-of-function in response to inactivation of a subset of neurons can be transient. Longer-term inactivation of neurons in our circuit, in the presence of active learning, allowed networks to rapidly compensate. This compensation occurred on a time scale much faster than the original training time, likely because inactivation does not completely destroy the network's previously learned computations. Recovery of behavior has been observed before in experiments [Murray and Baxter 2006; Newsome and Pare 1988; Fetsch et al. 2018; Rudolph and Pasternak 1999; Jeurissen et al. 2021]. Overall, drawing conclusions about the causal role a circuit plays in a given computation can be difficult without first analyzing the transient responses of animals immediately after inactivation — a commonly omitted or poorly documented aspect in many studies.

Fast time-scale inactivation techniques (e.g., optogenetics) [De et al. 2020; Luo et al. 2018; Wiegert et al. 2017; Afraz et al. 2015] have greatly increased in popularity as they allow precise control of the affected neurons with sub-second resolution. As we show here, brief periods of

inactivity interspersed with normal activity also make it harder for a learning system to identify and adapt to the perturbation. However, compensation can occur even for fast optogenetic perturbations (Fig. 6.5), as has been observed experimentally [Fetsch et al. 2018]. But such compensations tend to take longer compared to techniques in which the inactivation is more sustained (Fig. 6.5). This longer adaptation may be a result of destructive interference during re-learning, where the synaptic changes needed to improve performance during perturbation are misaligned and cancel those in the absence of perturbation, thus slowing down learning overall. For our example direction-discrimination task, once compensation does occur, it could take the form of two separate attractors, one dedicated to performing during the perturbed condition, and the other for the unperturbed condition. Alternatively, the network may converge to a single attractor, modified from its original solution such that does not include the inactivated subset of neurons.

The phenomena observed in this paper that can produce negative results in an inactivation study—redundancy, rapid relearning—are problems more general than just evidence integration [Vaidya et al. 2019; Wolff and Ölveczky 2018]. Here we have provided several suggestions for identifying the effects of causal manipulations in neural circuits, and have provided several cautionary tales based on the choice of a particular architecture and task. Circuits before and after manipulation are only tenuously related, and drawing conclusions about the function of natural circuits from the effects of inactivation can be quite difficult. Implementing proper controls for these effects and applying careful interpretations of observed experimental results in terms of the system's computational structure will benefit inactivation studies across a breadth of subfields.

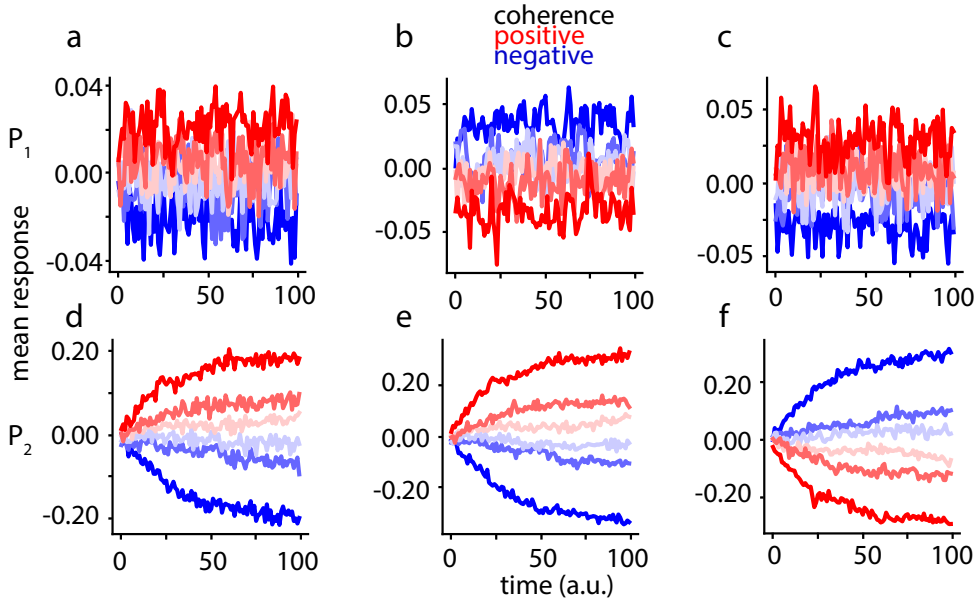

Figure 6.7: Example neuron response profiles from the simple integration network. a-c) Example stimulus-conditioned mean neuron responses from P1. **d-f)** Example stimulus-conditioned mean neuron responses from P2. Colors indicate the coherence strength and sign (blue indicates leftward motion, red indicates rightward motion).

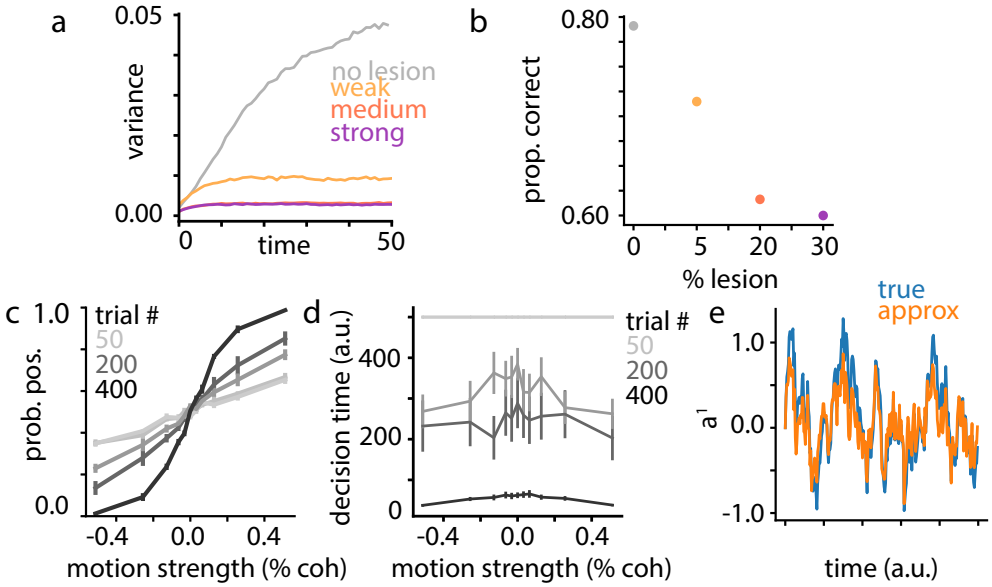

Figure 6.8: Additional analysis on the effects of inactivation and relearning. a) Variance of the network output through time as a function of the perturbation magnitude **b)** Proportion of incorrect choices as a function of the number of inactivated neurons in the output population. **c)** Psychometric function as a function of the number of retraining trials given to the network for a strong inactivation **d)** Same as **c**, but for the chronometric function. Bars indicate ±1 s.e.m. across 10 simulated networks **e)** Difference between the true (Eq. 6.14) and approximate (Eq. 6.16) dynamics for a trained network over 1 trial projected onto the 1st principal component.

7 | Conclusions and future directions

Our work in this thesis has made headway along three distinct axes towards the ultimate goal of characterizing the relationship between synaptic plasticity in the brain and learning at a perceptual or behavioral level. The first axis deals with developing good, testable hypotheses for how the brain learns under different conditions. As outlined in Chapter 1, there are many markers of a good normative synaptic plasticity model, in particular the locality of its parameter updates, its demonstrable ability to reduce an objective function, its scalability, testability, architectural flexibility, and ability to learn from continuous temporal inputs in an online fashion. Our work on impression learning (Chapter 5) was directly informed by these criteria: it generalizes previous canonical models of unsupervised sensory plasticity models like the Wake-Sleep algorithm [Hinton et al. 1995] to handle continuous temporal inputs online, and maps the algorithm onto a particular form of synaptic plasticity at apical and basal dendrites of pyramidal neurons so that the model makes concrete, testable predictions about how real neurons should adapt to their inputs.

The second axis of improvement deals with concretely testing predictions of normative models in neural circuits. We first constructed a model of efficient reward-based learning in sensory circuits (Chapter 3), and subsequently modified it to match context-dependent adaption observed in 2-photon calcium recordings of animals performing perceptual learning (Chapter 4). Not only were we able to find a striking correspondence to the neural data, but within our model we were able to compare several different possibilities for how the circuit adapts during behavior, and found the only model able to explain context-dependent responses had distinct, context-

sensitive synapses that activate during behavior and are modified by reward signals into adulthood. Furthermore, our model is now able to function as an *in silico* tool for experimental design, generating predictions about how different stimuli during development and behavior will affect learned neural representations.

The final axis deals with implications of normative plasticity for the neuroscience community at large. In Chapter 6, we explored how experimental inactivations of neurons that are undergoing continual task-driven plasticity can fail to identify neurons important for the task: the inactivated neurons *were* important before inactivation, but if the inactivation is persistent, the system can rapidly compensate for their loss through synaptic plasticity. This simple example illustrates the incredible complexity involved in studying adaptive neural systems: when a neural circuit adapts in real time to experimental manipulations, it is very difficult to draw conclusions about that circuit's natural function without first characterizing in detail the adaptive process itself and how it interacts with the manipulation in question.

Each of these axes has plenty of room for further development. Impression learning has improved considerably on previous models of unsupervised sensory learning, but two critical questions remain: how far can the model be pushed at a theoretical level, and how can its specific predictions be tested? There are many possible ways that apical and basal dendrites could conceivably interact in impression learning. As it currently stands, we have only explored periods of neural activity dominated by either apical dendritic activity or basal dendritic activity, but not both: understanding whether it is possible under impression learning for apical and basal dendritic activity to collectively contribute to neural spiking is an important theoretical precondition for validating the algorithm experimentally. Furthermore, impression learning was designed for unsupervised sensory learning, but can a similar learning scheme be used for supervised or reinforcement learning? Impression learning provides huge performance improvements for unsupervised learning relative to algorithms that project scalar reward signals to synapses like REINFORCE (or Neural Variational Inference; Appendix B.2.1), but extending the mathematics

underlying impression learning to reinforcement learning will require a nontrivial generalization.

Beyond theoretical developments, there is much to be done analyzing plasticity in apical and basal dendrites at an experimental level. Preliminary experiments in the hippocampus [Bittner et al. 2015, 2017] and cortex [Letzkus et al. 2006; Froemke et al. 2005; Sjöström and Häusser 2006] roughly align to the types of plasticity predicted by impression learning, but work remains to validate the theory in detail. In particular, the following questions are outstanding: do dendritic calcium events in apical dendrites align to spiking activity at the soma over the course of adaptation? Are the signals arriving at apical dendrites predictions of future activity, supervisory targets, or do they just similarly to incoming signals at basal dendrites? And lastly, are neurons able to sample from their generative model using apical dendritic activity, i.e. if a manipulation were to cause apical dendritic signals to dominate neural activity in the absence of any stimulus, would neural activity be similar to responses to realistic stimuli? It is a merit of impression learning that answers to these questions can validate the theory and distinguish it from other alternatives, but work remains to complete this validation.

In the auditory system, we have made quite a lot of progress by characterizing plasticity induced by acetylcholine signals from the nucleus basalis as a form of context-dependent reward-modulated Hebbian plasticity. However, this form of rewarded plasticity seems to interact with unsupervised developmental plasticity based on the statistics of the animals' sensory environment. It would be fascinating to explore how these different types of adaptation at different points in an animal's life can collectively shape its representation.

Our model provides a very good fit to experimental data, but for more complex tasks with longer temporal delays, more complicated class categories, or multiple different behavioral contexts, we might expect the learning process itself to be more complex. Studying the properties of neurons in the nucleus basalis and the neurons that they project to in auditory cortex under these circumstances could uncover more powerful forms of spatial and temporal credit assignment than the scalar reward signal that we have envisioned in our model, and may lead to a vision of

learning in which the projected acetylcholine signals are *themselves* learning to provide more precise and informative teaching signals to neurons. Such a system with *detailed credit assignment* may end up looking closer to a reward-based variant of impression learning, suggesting that these two forms of plasticity could be more similar than they appear. We have already demonstrated in Chapter 5 that detailed credit assignment can give remarkable performance increases for simple tasks, but beyond this, animals are required to learn multiple tasks in many different contexts and environments throughout their lives. More complicated credit assignment algorithms for multi-task learning can help ensure that learning for one task does not interfere with learning for another task and can help ameliorate the catastrophic forgetting caused by overwriting previously learned tasks [French 1999]. However, how auditory learning adapts to multi-task conditions is not currently known. A complex system can behave simply under simple conditions: we quite likely have much to learn about auditory learning under conditions that approach ecological realism.

Our modeling of causal manipulations has so far focused on proper interpretation of experiments on adaptive systems in which adaptation itself is not the focus. However, these interventions also have a lot of promise for eliminating or validating different forms of plasticity present in a circuit. The *way* in which a circuit responds to inactivation could determine whether compensation is purely homeostatic, some form of unsupervised learning, task-based learning, or some combination of the three, based on the way that other neurons in the circuit adapt in response. For instance, do they not modify unless the subject is explicitly rewarded? Do the networks have to be engaged by a task at all? Do neurons reorganize in a way that respects basic principals of sensory coding, or do they seem to simply be balancing their firing rates homeostatically? There is incredible potential for using not only inactivation tools, but also manipulations of sensory statistics, or even artificial neural readouts like brain-computer interfaces to probe the learning capacities and limitations of neural circuits.

These questions are for future work to resolve. As it stands, this thesis is a celebration of the

brain as a dynamic construct: the moment that we believe that we have grasped it, it adapts and slips away from us. By studying how and *why* the brain adapts, we can begin to understand the principles underlying what appears at face value to be protean chaos. We have presented a brief moment in this investigation, but this study is *itself* a dynamic process. It is fitting that even in analyzing the mechanisms of neural adaptation we have raised as many questions as we have addressed: it demonstrates that this work comprises just one step in our collective dance towards understanding our own minds.

A | Appendix: An overview of normative synaptic plasticity modeling

A.1 Why can't the brain do gradient descent?

We have provided one surefire way to decrease an objective function by modifying the parameters of a neural network—'simply' take small steps in the direction of the gradient of the loss (Section 2.2.2). To appreciate the challenges faced by theories of normative plasticity, it's important to understand why a biological system *could not* do this: in this section we will provide a simplified argument as to why gradient descent within multilayer neural networks produces *nonlocal* parameter updates, thus failing our most critical desideratum for a normative plasticity theory (Section 2.2.1). More detailed arguments for multilayer neural networks can be found here [Lillicrap et al. 2020], and descriptions of why gradient descent becomes even more implausible for recurrent neural networks trained with either backpropagation through time [Werbos 1990] or real-time recurrent learning [Williams and Zipser 1989] can be found here [Marschall et al. 2020].

The 'weight transport problem' is the most basic reason that gradient descent is implausible for neural networks. Suppose that we have a stimulus-dependent network response, $\mathbf{r}(\mathbf{W}^{in}, \mathbf{s}) = f(\mathbf{W}^{in}\mathbf{s})$, where $\mathbf{r}$ is an $N \times 1$ vector, and $\mathbf{W}^{in}$ is an $N \times N^s$ weight matrix mapping stimuli $\mathbf{s}$ into responses after a pointwise nonlinearity $f(\cdot)$. This network response is decoded into a network output, $o(\mathbf{W}^{in}, \mathbf{s}) = \mathbf{W}^{out}\mathbf{r}(\mathbf{W}^{in}, \mathbf{s})$, where $\mathbf{W}^{out}$ is a $1 \times N$ vector mapping network responses into

a scalar output. Now suppose for simplicity that our loss for a single stimulus example is given by:

$$\mathcal{L} = \frac{1}{2}\left(\hat{o} - o(\mathbf{W}^{in}, \mathbf{s})\right)^2. \tag{A.1}$$

This objective is trying to bring the stimulus-dependent network response $o(\mathbf{W}^{in}, \mathbf{s})$ close to the target output $\hat{o}$, and is zero if and only if $o = \hat{o}$. A reasonable hypothesis would be that the gradient of this objective function with respect to a synaptic weight, $\mathbf{W}^{in}_{ij}$, will produce a parameter update that is local: we will see that this is not true. Taking the gradient, we have:

$$\frac{d}{d\mathbf{W}^{in}_{ij}}\mathcal{L} = \frac{1}{2}\frac{d}{d\mathbf{W}^{in}_{ij}}\left(\hat{o} - o(\mathbf{W}^{in}, \mathbf{s})\right)^2 \tag{A.2}$$

$$= (\hat{o} - o)\,\frac{d}{d\mathbf{W}^{in}_{ij}} o(\mathbf{W}^{in}, \mathbf{s}) \tag{A.3}$$

$$= (\hat{o} - o)\,\mathbf{W}^{out}_i \frac{d}{d\mathbf{W}^{in}_{ij}} f_i(\mathbf{W}^{in}\mathbf{s}) \tag{A.4}$$

$$= (\hat{o} - o)\,\mathbf{W}^{out}_i f_i'(\mathbf{W}^{in}\mathbf{s})\mathbf{s}_j. \tag{A.5}$$

Breaking down this final update, we can see three terms: an error, $(\hat{o} - o)$, the neuron's *output weight* $\mathbf{W}^{out}_i$, and an approximately Hebbian term $f_i'(\mathbf{W}^{in}\mathbf{s})\mathbf{s}_j$, which requires only a combination of pre- and post-synaptic activity. One might be tempted to organize the plasticity rule into a error feedback signal received by the neuron, scaled by a neuron-specific synaptic weight $\mathbf{W}^{out}_i$, and then combined with Hebbian coactivity to produce a synaptic update (Fig. A.1a). This would have the form of a three-factor plasticity rule [Frémaux and Gerstner 2016], combining weighted feedback with pre- and post-synaptic activity. However, the weight transport problem is as follows: $\mathbf{W}^{out}_i$ provides the strength of a synapse in the *feedforward* pathway—how could it possibly come to be that a feedback learning pathway would have access to the *same* synaptic weight? The answer is that there is no evidence for such a system of weight sharing across feedforward and

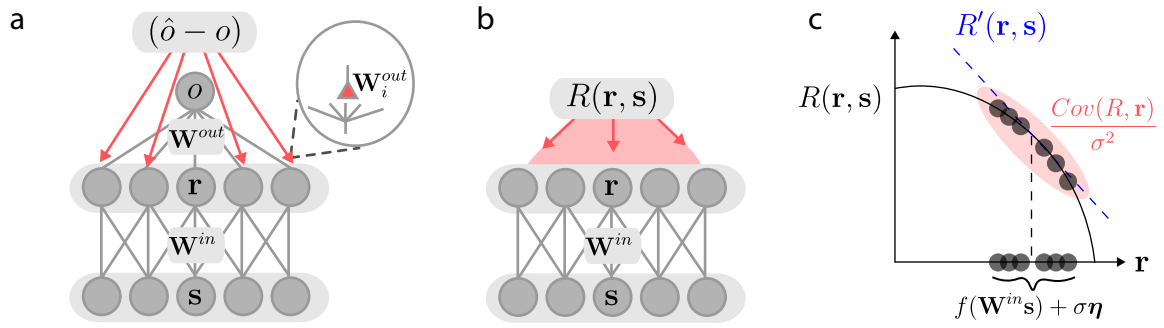

Figure A.1: Weight transport and REINFORCE. a. Traditional gradient descent propagates a credit assignment signal $(\hat{o} - o)\mathbf{W}_i^{out}$ to each neuron $\mathbf{r}_i$. How this pathway could have access to $\mathbf{W}_i^{out}$ is unclear: this is the 'weight transport' problem. **b.** REINFORCE resolves the weight transport problem by projecting a scalar reward signal $R(\mathbf{r}, \mathbf{s})$ to all synapses. **c.** By correlating this reward with fluctuations in neural activity, neurons can approximate the true gradient.

feedback pathways in the brain, though there are many hypotheses about how such a system could, in theory, be approximated by a normative plasticity algorithm. This problem becomes more pronounced in multilayer networks, where the error signal must be propagated through many interconnected connectivity layers.

It is also worth noting two key differentiability assumptions inherent to this approach. For one, we assume not only that the loss function $\mathcal{L}$ is differentiable, but that some 'error calculating' part of the brain does differentiate it. This requires knowledge of what the desired network output should be $\hat{o}$, which for many real-world tasks is not possible. Second, we assume that the network activation function $f(\cdot)$ is differentiable. Since neurons typically emit binary spikes, this differentiability assumption is not necessarily valid, though several modern methods have circumvented this problem by using either stochastic neuron models [Williams 1992; Dayan and Hinton 1996] or by using clever optimization tricks [Bellec et al. 2020]. In subsequent sections, we will outline two canonical algorithms that employ clever tricks to circumvent the weight transport problem.

A.2 The unidentifiability of an objective

In this section we illustrate why the choice of objective function for a normative plasticity model is never uniquely determined by data. We will consider two situations: the system has already settled to its optimal setting of its weights, $\mathbf{W}^*$, and in the second we are able to observe the system's plasticity update $\Delta\mathbf{W}$.

A.2.1 Unidentifiability based on an optimum

Suppose that some setting of synaptic weights $\mathbf{W}^*$ minimizes an objective function $\mathcal{L}$, i.e. $\mathcal{L}(\mathbf{W}^*) < \mathcal{L}(\mathbf{W})\ \forall\mathbf{W}$. We might be tempted to argue that because $\mathbf{W}^*$ minimizes $\mathcal{L}$, $\mathcal{L}$ must be *the* objective that the system is minimizing. However, there are an infinite variety of alternative objectives that share the same minimum. To see this, take a new objective $\tilde{\mathcal{L}} = \sigma\left(\mathcal{L}(\mathbf{W})\right)$ for any differentiable, monotonically increasing function $\sigma(\cdot)$. Then we have:

$$\mathcal{L}(\mathbf{W}^*) < \mathcal{L}(\mathbf{W})\ \forall\mathbf{W} \tag{A.6}$$

$$\Rightarrow \sigma\left(\mathcal{L}(\mathbf{W}^*)\right) < \sigma\left(\mathcal{L}(\mathbf{W})\right)\ \forall\mathbf{W} \tag{A.7}$$

$$\Rightarrow \tilde{\mathcal{L}}(\mathbf{W}^*) < \tilde{\mathcal{L}}(\mathbf{W})\ \forall\mathbf{W}, \tag{A.8}$$

where the second equality follows from the order preservation property of the monotonically increasing $\sigma(\cdot)$. This means that $\mathbf{W}^*$ also minimizes $\tilde{\mathcal{L}}$, i.e. we will be unable to arbitrate between whether the system is 'attempting' to minimize $\tilde{\mathcal{L}}$ or $\mathcal{L}$ on the basis of the optimized network state given by $\mathbf{W}^*$.

A.2.2 Unidentifiability based on an update rule

Suppose instead that we were able to observe the adaptive plasticity mechanism of a system, and were able to verify that it really does decrease an objective function $\mathcal{L}$, i.e. by Eq. 2.4,

$$\frac{d\mathcal{L}}{d\mathbf{W}}(\mathbf{W})^T \Delta\mathbf{W} < 0 \; \forall \mathbf{W}. \tag{A.9}$$

We might now be tempted to argue that, by observing the plasticity rule itself, $\Delta\mathbf{W}$, we will be more able to assert that the system, by virtue of consistently decreasing $\mathcal{L}$, is 'attempting' to minimize $\mathcal{L}$. However, the *exact same* family of alternative objectives will also be minimized ($\tilde{\mathcal{L}} = \sigma\left(\mathcal{L}(\mathbf{W})\right)$ for any differentiable, monotonically increasing function $\sigma(\cdot)$. To see this, we observe:

$$\frac{d\mathcal{L}}{d\mathbf{W}}(\mathbf{W})^T \Delta\mathbf{W} < 0 \; \forall \mathbf{W} \tag{A.10}$$

$$\Rightarrow \frac{d\sigma(\mathcal{L}(\mathbf{W}))}{d\mathcal{L}(\mathbf{W})} \frac{d\mathcal{L}}{d\mathbf{W}}(\mathbf{W})^T \Delta\mathbf{W} < 0 \; \forall \mathbf{W} \tag{A.11}$$

$$\Rightarrow \frac{d\tilde{\mathcal{L}}}{d\mathbf{W}}(\mathbf{W})^T \Delta\mathbf{W} < 0 \; \forall \mathbf{W}, \tag{A.12}$$

where the first implication follows from the fact that $\sigma(\cdot)$ is differentiable and increasing (it has strictly positive derivative), and the second implication follows from the chain rule. This implies that plasticity rules ($\Delta\mathbf{W}$) and trained neural circuits ($\mathbf{W}^*$) can at most partially constrain the space of viable objective functions the system could be minimizing.

A.3 REINFORCE

In this section, we will provide a mathematical tutorial on the REINFORCE learning algorithm [Williams 1992], which is a mechanism for updating the parameters in a stochastic neural network

for reinforcement learning objective functions. It's chief advantages are twofold: first, it only requires you to be able to evaluate an objective function (i.e. the reward received on any given trial), not the gradient of the objective function with respect to the parameters (Fig. A.1b). This is very useful in situations in which the relationship between rewards and network outputs is not clear to an agent, as would be the case in many reinforcement learning scenarios. Second, under a broad range of biologically reasonable assumptions about a neural network architecture, the parameter updates produced by this algorithm are 'local,' meaning the only information required for a parameter update would reasonably be available to a synapse in the brain. This algorithm produces updates that are within the class of 'reward-modulated Hebbian plasticity rules.' The chief disadvantage of this algorithm is its comparative data-inefficiency relative to backpropagation. In practice, far more data samples (or equivalently, much lower learning rates) will be required to produce the same improvements in performance compared to backpropagation [Werfel et al. 2003].

The REINFORCE algorithm and minor variations appears in different fields with different names. It is useful to keep track of these alternative names, because they all use roughly the same derivation, with some improvements or field-specific modifications. In machine learning, the algorithm is often referred to as *node perturbation* [Richards et al. 2019; Lillicrap et al. 2020; Werfel et al. 2003], because it involves correlating fluctuations in neuron (node) activity with reward signals. In computational neuroscience, it is sometimes called *3-factor* or *reward-modulated Hebbian* plasticity [Frémaux and Gerstner 2016], though REINFORCE is only one of several algorithms referred to by these blanket terms. In reinforcement learning, REINFORCE is often treated as a member of the more general class of *policy gradient* [Sutton and Barto 2018] methods, which can be used to train any parameterized stochastic agent through reinforcement. Policy gradient methods need not commit to a neural network architecture, and are consequently not always local. Lastly, very similar methods are used for fitting variational Bayesian models, and are in these contexts referred to as either *black box variational inference* [Ranganath et al. 2014] or

neural variational inference [Mnih and Gregor 2014].

In what follows, we will provide a brief derivation of the REINFORCE learning algorithm for a 1-layer feedforward neural network. We will then discuss the many extensions of the algorithm as well as its strengths and limitations as a normative plasticity model.

A.3.1 Network model

Most neural networks used in machine learning are deterministic. However, neurons in biological systems fluctuate across trials and stimulus presentations, so modeling them as stochastic is often more appropriate. It will turn out that these fluctuations can be used to produce parameter updates in a way that a deterministic system could not.

First, we will assume that there are stimuli drawn from some stimulus distribution, $p(\mathbf{s})$, and we will define the neural network response to a given stimulus drawn from this distribution as:

$$\mathbf{r} = f(\mathbf{W}^{in}\mathbf{s}) + \sigma\boldsymbol{\eta}, \tag{A.13}$$

where the $\boldsymbol{\eta}$ is the source of random fluctuations which, for simplicity, is drawn from a standard normal distribution ($\mathcal{N}(0,1)$). In this equation, $\mathbf{s}$ is an $N_s \times 1$ vector, $\mathbf{W}^{in}$ is an $N_r \times N_s$ matrix, $f(\cdot)$ is the tanh nonlinearity, and $\boldsymbol{\eta}$ is an $N_r \times 1$ vector.

This equation defines a conditional probability distribution, $p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in}) \sim \mathcal{N}(f(\mathbf{W}^{in}\mathbf{s}),\sigma^2)$. There is an interesting point here: neuron activities are now samples from this conditional probability distribution, and so we can study how neurons behave on average by taking expectations over the probability distribution.

For simplicity and clarity we will restrict ourselves to this neural architecture for our derivation, but the basic principles apply more generally to a variety of noise sources and neural architectures (see Section A.3.5).

A.3.2 Defining the objective

We will assume that our goal is to maximize some instantaneous reward $R(\mathbf{r}, \mathbf{s})$ on average across many different samples of $R(\mathbf{r}, \mathbf{s})$ and $\mathbf{s}$. This allows us to write our objective function $O(\mathbf{W}^{in})$ as:

$$O(\mathbf{W}^{in}) = \int R(\mathbf{r}, \mathbf{s}) p(\mathbf{r}|\mathbf{s}; \mathbf{W}^{in}) p(\mathbf{s}) d\mathbf{r} d\mathbf{s}. \tag{A.14}$$

In practice, this integral might be analytically impossible to integrate, but we can always approximate it (because it is an expectation) using samples from $p(\mathbf{r}|\mathbf{s}; \mathbf{W}^{in})$ and $p(\mathbf{s})$ as an empirical average over K samples $\mathbf{r}_k$ and $\mathbf{s}_k$:

$$O(\mathbf{W}^{in}) \approx \frac{1}{K} \sum_{k=0}^{K} R(\mathbf{r}^{(k)}, \mathbf{s}^{(k)}). \tag{A.15}$$

Procedurally, this would amount to sampling $\mathbf{s}$ and $\mathbf{r}$ each K times, calculating the reward for each trial, and taking an average.

A.3.3 Taking the gradient

Now that we have our objective function, we can evaluate its derivative with respect to a particular synapse $\mathbf{W}^{in}_{ij}$ in the network:

$$\frac{dO(\mathbf{W}^{in})}{d\mathbf{W}^{in}_{ij}} = \frac{d}{d\mathbf{W}} \int R(\mathbf{r}, \mathbf{s}) p(\mathbf{r}|\mathbf{s}; \mathbf{W}^{in}) p(\mathbf{s}) d\mathbf{r} d\mathbf{s} \tag{A.16}$$

$$= \int R(\mathbf{r}, \mathbf{s}) \left[\frac{d}{d\mathbf{W}^{in}_{ij}} p(\mathbf{r}|\mathbf{s}; \mathbf{W}^{in}) \right] p(\mathbf{s}) d\mathbf{r} d\mathbf{s}. \tag{A.17}$$

We could theoretically stop here and evaluate $\frac{d}{d\mathbf{W}^{in}_{ij}} p(\mathbf{r}|\mathbf{s}; \mathbf{W}^{in})$ explicitly. However, in the same way that we can approximate $O(\mathbf{W}^{in})$ as an empirical average over samples, we would like to be able to approximate our derivative as an average. To do this requires us to keep our loss in the

form of an expectation over $p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in})p(\mathbf{s})$. We notice a convenient identity: $\frac{d}{d\mathbf{W}^{in}_{ij}}p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in}) = \frac{d}{d\mathbf{W}^{in}_{ij}}\exp(\log p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in})) = \left[\frac{d}{d\mathbf{W}^{in}_{ij}}\log p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in})\right] p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in})$, which is a simple application of the chain rule. Inserting this identity into the above equation, we get:

$$\frac{dO(\mathbf{W}^{in})}{d\mathbf{W}^{in}_{ij}} = \int R(\mathbf{r},\mathbf{s})\left[\frac{d}{d\mathbf{W}^{in}_{ij}}\log p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in})\right] p(\mathbf{s})d\mathbf{r}d\mathbf{s} \tag{A.18}$$

$$\approx \frac{1}{K}\sum_{k=0}^{K} R(\mathbf{r}^{(k)},\mathbf{s}^{(k)})\left[\frac{d}{d\mathbf{W}^{in}_{ij}}\log p(\mathbf{r}^{(k)}|\mathbf{s}^{(k)};\mathbf{W}^{in})\right]. \tag{A.19}$$

Though this is an approximation, we note that by the Law of Large Numbers, we can improve its accuracy arbitrarily by increasing our number of samples K. In practice, however, taking $K = 1$ will prove to be the most straightforward way to get an update that is local in time—although such an update will still on average match the true gradient exactly, its high variance can lead to very inefficient learning.

We have left the derivation completely general up until this point. Different choices of $p(\mathbf{r}|\mathbf{s};\mathbf{W})$ will produce different updates. Our particular choice gives:

$$\frac{d}{d\mathbf{W}^{in}_{ij}}\log p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in}) = \frac{d}{d\mathbf{W}^{in}_{ij}}\sum_{i=0}^{N_r}\frac{1}{2\sigma^2}(\mathbf{r}_i - f_i(\mathbf{W}^{in}\mathbf{s}))^2 + C \tag{A.20}$$

$$= \frac{1}{\sigma^2}\sum_{n=0}^{N_r}(\mathbf{r}_i - f_i(\mathbf{W}^{in}\mathbf{s}))\frac{df_i(\mathbf{W}\mathbf{s})}{d\mathbf{W}^{in}_{ij}}. \tag{A.21}$$

For a particular weight $\mathbf{W}^{in}_{ij}$, $\frac{df_l(\mathbf{W}^{in}\mathbf{s})}{d\mathbf{W}_{ij}} = 0$ if $i \neq l$, so we have:

$$\frac{d}{d\mathbf{W}^{in}ij}\log p(\mathbf{r}|\mathbf{s};\mathbf{W}) = \frac{1}{\sigma^2}(\mathbf{r}_i - f_i(\mathbf{W}\mathbf{s}))f_i'(\mathbf{W}\mathbf{s})\mathbf{s}_j. \tag{A.22}$$

Plugging this equation into Eq. A.16 gives the following parameter update:

$$\Delta \mathbf{W}_{ij}^{in} \propto \frac{1}{K} \sum_{k=0}^{K} R(\mathbf{r}^{(k)}, \mathbf{s}^{(k)}) \left[\frac{1}{\sigma^2} (\mathbf{r}_i^{(k)} - f_i(\mathbf{W}^{in}\mathbf{s}^{(k)})) f_i'(\mathbf{W}^{in}\mathbf{s}^{(k)}) \mathbf{s}_j^{(k)} \right] \approx \frac{dO(\mathbf{W}^{in})}{d\mathbf{W}_{ij}^{in}}. \tag{A.23}$$

If we want to update all of our parameters simultaneously using parallelized matrix operations, we can write this as an outer product:

$$\Delta \mathbf{W}^{in} \propto \frac{1}{K} \sum_{k=0}^{K} R(\mathbf{r}^{(k)}, \mathbf{s}^{(k)}) \left[\frac{1}{\sigma^2} (\mathbf{r}^{(k)} - f(\mathbf{W}^{in}\mathbf{s}^{(k)})) \odot f'(\mathbf{W}^{in}\mathbf{s}^{(k)}) \right] \mathbf{s}^{(k)T}, \tag{A.24}$$

where $\odot$ denotes a Hadamard (elementwise) vector product. Interestingly, the $\frac{1}{\sigma^2}(\mathbf{r} - f(\mathbf{W}^{in}\mathbf{s}))$ term here is exactly equal to $\boldsymbol{\eta}$.

A.3.4 Why don't we need the derivative of the loss?

One way of interpreting this parameter update is that neural units are correlating fluctuations in their neural activity with the rewards received to approximate $\frac{dR(\mathbf{r},\mathbf{s})}{d\mathbf{r}}$ (Fig. A.1c). To see this, first notice that:

$$\mathbb{E}\left[b \left[\frac{1}{\sigma^2} (\mathbf{r} - f(\mathbf{W}^{in}\mathbf{s})) \odot f'(\mathbf{W}^{in}\mathbf{s}) \right] \mathbf{s}^T \right]_{p(\mathbf{r}|\mathbf{s})} = 0, \tag{A.25}$$

for any constant b, because $\mathbb{E}\left[\mathbf{r} - f(\mathbf{W}^{in}\mathbf{s})\right]_{p(\mathbf{r}|\mathbf{s})} = 0$. If we take $b = \mathbb{E}\left[R(\mathbf{r}, \mathbf{s})\right]_{p(\mathbf{r}|\mathbf{s})}$, then we can rewrite the gradient without changing its expected value:

$$\frac{dO(\mathbf{W}^{in})}{d\mathbf{W}^{in}_{ij}} = \int \left(R(\mathbf{r},\mathbf{s}) - \mathbb{E}\left[R(\mathbf{r},\mathbf{s})\right]_{p(\mathbf{r}|\mathbf{s})}\right)\left[\frac{1}{\sigma^2}(\mathbf{r}_i - f_i(\mathbf{W}^{in}\mathbf{s}))f_i'(\mathbf{W}^{in}\mathbf{s})\mathbf{s}_j\right] p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in})p(\mathbf{s})d\mathbf{r}d\mathbf{s} \tag{A.26}$$

$$= \int \frac{1}{\sigma^2} Cov(R(\mathbf{r},\mathbf{s}),\mathbf{r}_i)\left[f_i'(\mathbf{W}^{in}\mathbf{s})\mathbf{s}_j\right] p(\mathbf{s})d\mathbf{s}, \tag{A.27}$$

where $Cov(R(\mathbf{r},\mathbf{s}),\mathbf{r}_i) = \int (R - \mathbb{E}\left[R\right]_{p(\mathbf{r}|\mathbf{s})})(\mathbf{r}_i - \mathbb{E}\left[\mathbf{r}_i\right]_{p(\mathbf{r}|\mathbf{s})})p(\mathbf{r}|\mathbf{s})d\mathbf{r}$ is the stimulus-conditioned covariance between network firing rates and reward. The sample-based parameter update is therefore using the fluctuations in neural activity to compute this covariance.

A.3.5 Assessing REINFORCE

Now that we have derived REINFORCE, we can examine its qualities as a normative plasticity theory. First, we ask: is this algorithm 'local' (Section 2.2.1)? The gradient for a particular synapse, $\frac{dO(\mathbf{W}^{in})}{d\mathbf{W}^{in}_{ij}}$ can be approximated with samples in an environment with stimuli $\mathbf{s}$, firing rates $\mathbf{r}$, and rewards $R(\mathbf{r},\mathbf{s})$ by $R(\mathbf{r},\mathbf{s})\left[\frac{1}{\sigma^2}(\mathbf{r}_i - f_i(\mathbf{W}^{in}\mathbf{s}))f_i'(\mathbf{W}^{in}\mathbf{s})\mathbf{s}_j\right]$. To decide whether this could be a plasticity rule implemented (or more realistically, approximated) by a biological system, we need to think about what pieces of information a synapse would have to have available.

First, the synapse needs $\mathbf{s}_j$, which amounts to just the presynaptic input, a common feature of any Hebbian synaptic plasticity rule. Second, the synapse needs $\frac{1}{\sigma^2}(\mathbf{r}_i - f_i(\mathbf{W}^{in}\mathbf{s}))f_i'(\mathbf{W}^{in}\mathbf{s})$. $\frac{1}{\sigma^2}$ is a constant, and so can be absorbed into the learning rate. $\mathbf{r}_i$ is the postsynaptic firing rate, which is also a common feature of any Hebbian plasticity rule. $(\mathbf{W}^{in}\mathbf{s})_i$ is the current injected into the postsynaptic neuron, and $f_i(\cdot)$ and $f_i'(\cdot)$ are both monotonic functions of this current, so it is quite conceivable that these values could be approximated by a biochemical process. Third, every synapse needs access to the scalar reward value received on a given trial, $R(\mathbf{r},\mathbf{s})$. This is the most 'nonlocal' information involved in the parameter update, however, there exist many theories

about how neuromodulatory systems in the brain can deliver information about reward diffusely to many synapses and induce plasticity (Section 2.2.1).

Now, we have already demonstrated that REINFORCE is able to perform approximate gradient descent for reinforcement learning objective functions—this in itself makes the algorithm very promising as a normative plasticity model (Section 2.2.2). Its chief advantage is that it does not require detailed knowledge of the reward function $R(\mathbf{r}, \mathbf{s})$ (i.e. how to differentiate it), which means that an animal could simply receive a reward from its environment, and relay that reward signal diffusely to its synapses. However, this also restricts the types of objectives that could plausibly be learned by a neural system. Unsupervised learning objectives like the ELBO require detailed knowledge of every neural activity of every neuron in the circuit in order to be calculable (Appendix A.4), and there is no evidence for downstream neural circuits that perform such calculations. Therefore, even though in principle REINFORCE can be used to train a neural network on *any* objective, explicit reinforcement is much more plausible than other alternatives.

We have only provided a derivation for a single-layer rate-based neural network with additive Gaussian noise, but REINFORCE extends quite readily to multilayer [Williams 1992], spiking [Frémaux et al. 2013], and recurrent networks [Miconi 2017] without any loss of locality. This indicates that the algorithm is both architecture-general (Section 2.2.3) and can handle temporal environmental structure (Section 2.2.4). Further, because a weight update can be calculated in a single trial, animals could use it to learn online (Section 2.2.5). The biggest point of failure for REINFORCE is that it scales poorly with high complexity in stimuli or task, large numbers of neurons, or prolonged delays in receipt of reward [Werfel et al. 2003; Fiete 2004; Bredenberg et al. 2021] (Section 2.2.6). The greater the number of neurons that contribute to reward and the higher the complexity of the reward function, the harder it becomes to estimate the correlation between a single neuron and reward, which is a prerequisite for the algorithm's function. Thus, though the algorithm is an unbiased estimator of the gradient, it can still be so variable an estimate as to be effectively useless in complex contexts. This suggests that if animals exploit the principles

of REINFORCE to update synapses, it is likely an approach paired with other algorithms, or hybridized in a way that allows for better scalability.

The last way to assess REINFORCE is on the basis of how it can be tested (Section 2.2.7). The simplest way to test this algorithm is by examining whether scalar reward-like signals (i.e. $R(\mathbf{r}, \mathbf{s})$) have a multiplicative effect on local plasticity in a circuit. At a single-neuron level this corresponds to identifying neuromodulators that affect plasticity. At a feedback level this corresponds to identifying neuromodulatory systems that project to the circuit in question, and observing whether their stimulation or silencing improves or blocks circuit-level plasticity or behavioral learning performance, respectively. These steps do not identify REINFORCE as the only possibility, but it narrows down the field of possibilities considerably, removing all candidate algorithms that either do not require any feedback, or that require more detailed feedback signals (Fig. 2.3a).

A.4 Wake-Sleep

Here we will provide a mathematical tutorial on the Wake-Sleep algorithm [Hinton et al. 1995; Dayan et al. 1995], which is one candidate biologically plausible learning algorithm for constructing a representation in sensory cortices. We will first provide one possible formulation of representation learning as an optimization problem [Roweis and Ghahramani 1999], and then introduce the Wake-Sleep algorithm[1], showing how the components necessary to the algorithm could be mapped onto a multicompartmental dendritic neuron model with local synaptic learning. We will then discuss how the algorithm can be extended beyond our simplified introduction, and provide a supplementary implementation of the algorithm performing a simple form of unsupervised learning.

[1] For another excellent tutorial with more of a machine learning focus, see [Kirby 2006].

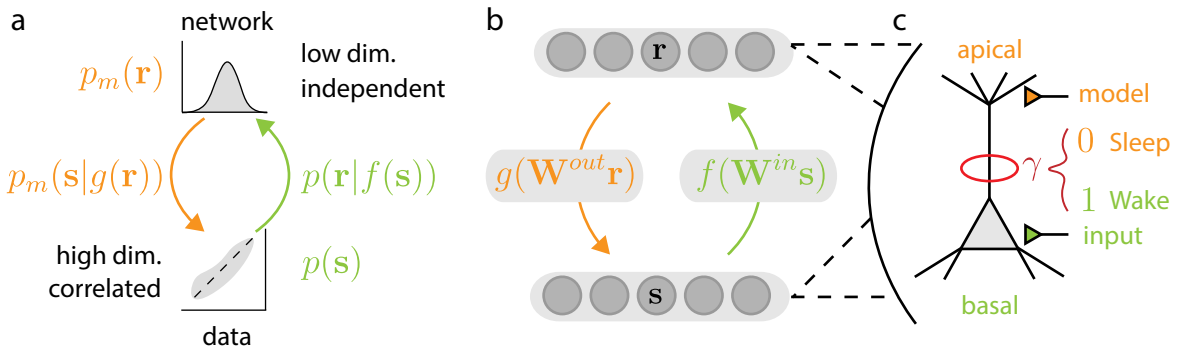

Figure A.2: The Wake-Sleep algorithm. a. The four components of a good representation: $p_m(\mathbf{s}|\mathbf{r})$ and $p(\mathbf{r}|\mathbf{s})$ map $\mathbf{r}$ to $\mathbf{s}$ and back again from $\mathbf{s}$ to $\mathbf{r}$, respectively. $p_m(\mathbf{r})$ defines 'useful' features of a neural representation by constraining its topology. $p(\mathbf{s})$ provides the environmental input distribution, which the neural representation must match. **b.** The architecture of the Wake-Sleep algorithm: the decoder, $g(\mathbf{W}^{out}\mathbf{r})$ maps $\mathbf{r}$ to $\mathbf{s}$, and the forward map, $f(\mathbf{W}^{in}\mathbf{s})$ maps $\mathbf{s}$ to $\mathbf{r}$. **c.** Physically, these maps correspond to a multicompartmental pyramidal neuron model for each layer, where the 'model' synapses are on the apical dendrites, and the 'forward map' synapses are on the basal dendrites. γ gates which synapses determine neural activity, putting the network in the Wake phase $\gamma = 1$ or the Sleep phase $\gamma = 0$.

A.4.1 Defining a good objective

Suppose that at any given moment in time, a neural network is receiving sensory stimuli $\mathbf{s}$ from its environment. Our first challenge is to articulate what it would mean to form a good neural representation $\mathbf{r}$ of these stimuli (Fig. A.2a). First of all, 'represented' stimuli should be decodable from neural firing rates, i.e. there should exist a mapping $g(\cdot) : \mathbb{R}^{N^r} \to \mathbb{R}^{N^s}$ such that $\mathbf{s} \approx g(\mathbf{r})$. Second, we will also argue that neural firing rates should be decodable from *stimuli*, i.e. there should exist a mapping $f(\cdot) : \mathbb{R}^{N^s} \to \mathbb{R}^{N^r}$ such that $\mathbf{r} \approx f(\mathbf{s})$—this means that there cannot be 'extra' features of neural activity that are not contained within the stimuli themselves. This amounts to postulating an approximately bijective relationship between stimuli and firing rates. It means that neural activities should directly correspond to stimuli that have been received.

If these two requirements were sufficient, we might want to simply have one neuron per stimulus dimension, and have it faithfully replicate its immediate input as accurately as possible, i.e. we would take $f(\mathbf{s}) = \mathbb{I}\mathbf{s}$ and $g(\mathbf{r}) = \mathbb{I}\mathbf{r}$, where $\mathbb{I}$ is an identity matrix, so that $\mathbf{r} = \mathbb{I}\mathbf{s} = \mathbf{s}$. This identity transformation is obviously not useful, which makes one wonder—what does it

mean for a transformation to be useful? Most, if not all unsupervised machine learning and neuroscientific conceptions of a 'useful' representation reduce to some formulation of either metabolic or coding efficiency. Approaches within this 'efficiency' umbrella include dimensionality reduction [Roweis and Ghahramani 1999], clustering [Illing et al. 2021; Dayan et al. 1995], gain control [Simoncelli and Heeger 1998], whitening/factorization [Rezende et al. 2014], and sparsity [Simoncelli and Olshausen 2001]. Each of these definitions of 'usefulness' can be formulated as statements about the distribution of neural activities, independent of particular received stimuli, e.g. there are fewer neurons than stimulus dimensions (dimensionality reduction), neural activations occupy roughly discrete clusters in state space (clustering), neurons tend to be uncorrelated with one another (whitening/factorization), or neurons typicaly have low, sparse firing rates (gain control/sparsity/metabolic efficiency). In our formulation, ultimately learning will be unsupervised because we have made *a priori* determinations of what constitutes an efficient representation, and seek to transform incoming data to match those determinations.

Under our definition outlined so far, there are four components of a representation: the stimuli $\mathbf{s}$ themselves, distributed according to some probability distribution $p(\mathbf{s})$ determined by the environment; a decoder, which we will formulate probabilistically as $p_m(\mathbf{s}|g(\mathbf{r};\theta_m))$, which models the probability of $\mathbf{s}$ given our mapping from neural firing rates $\mathbf{r}$; a forward mapping from $\mathbf{s}$ to $\mathbf{r}$, which we will also formulate probabilistically as $p(\mathbf{r}|f(\mathbf{s};\theta))$; and our definition of efficiency, which dictates how neural firing rates 'should' be distributed, independently of stimuli themselves $p_m(\mathbf{r})$. Notice that here we have parameterized the forward map $p(\mathbf{r}|f(\mathbf{s};\theta))$ and the decoder (inverse map) $p_m(\mathbf{s}|g(\mathbf{r};\theta_m))$: once we formulate our objective, these will be the parameters that are adjusted to minimize it. $p(\mathbf{s})$—the environmental data distribution—obviously cannot change, but we could (and in practice would often want to) parameterize $p_m(\mathbf{r})$ and also fit those parameters. We have formulated our four components using probability distributions: after describing our objective function in these terms, we will show one possible way of mapping the components onto neural architecture.

Now, we have evocatively organized our components into two groups: $p_m(\mathbf{s}|g(\mathbf{r};\theta_m))$ and $p_m(\mathbf{r})$, versus $p(\mathbf{s})$ and $p(\mathbf{r}|f(\mathbf{s};\theta))$. The first group forms a joint distribution $p_m(\mathbf{r},\mathbf{s};\theta_m)$ which has the subscript m to indicate that it is a generative *model* of the data. Ideally, if its parameters were accurately fit, we could sample $\mathbf{r} \sim p_m(\mathbf{r})$, and then sample $\mathbf{s} \sim p_m(\mathbf{s}|g(\mathbf{r};\theta_m))$ and get a stimulus that looks like realistic environmental data. The second group also forms a joint distribution $p(\mathbf{r},\mathbf{s};\theta)$, which amounts to a forward mapping: we could receive a stimulus from the environment, and then have the probability distribution for firing rates $\mathbf{r}$ that correspond to it. Organizing our models in this way will allow us to achieve biophysical realism: $g(\cdot;\theta)$ and $f(\cdot;\theta)$ will correspond to actual synaptic connections in a model neural network. In practice, ordinary perception as we traditionally conceive of it would correspond to the forward mapping $f(\cdot;\theta)$. Interestingly, at the end of our derivation, it will become clear how an additional representational feature, 'detachability' [Clark and Toribio 1994]—a mechanism to activate neurons in the absence of the sensory stimuli that correspond to them—will be an emergent property of our formulation. We will show how a neural system might be able to leverage the $g(\cdot;\theta_m)$ to accomplish 'detachment', which one might imagine mapping perceptually to imagination, planning, prediction, hallucination, or possibly dreaming in different contexts.

For our representation to be good, the forward map should match its inverse, i.e. $p(\mathbf{r},\mathbf{s};\theta) \approx p_m(\mathbf{r},\mathbf{s};\theta_m)$. We could imagine formulating many objective functions that could accomplish this goal, but most of them will not accommodate an approximate optimization algorithm that will end up corresponding to a viable normative plasticity model. We will select the Kullback-Liebler (KL) divergence between these two distributions, precisely because it will produce such a normative plasticity model. Notice, though our presentation of the derivation is top-down, it is disingenuous to characterize normative plasticity model development strictly as top-down: locality would not magically emerge from an arbitrary choice of objective function, but rather this choice of objective function is superior to its many alternatives only *because* it produces locality (we won't be able to see why locality emerges until after we have defined p and p_m explicitly and have derived

parameter updates). We take our objective function to be:

$$\begin{aligned}\mathcal{L}_{Wake} &= D_{KL}(p(\mathbf{r}, \mathbf{s}; \theta)||p_m(\mathbf{r}, \mathbf{s}; \theta_m)) \\ &= \int \ln\left(\frac{p(\mathbf{r}, \mathbf{s}; \theta)}{p_m(\mathbf{r}, \mathbf{s}; \theta_m)}\right) p(\mathbf{r}, \mathbf{s}; \theta) d\mathbf{r} d\mathbf{s}. \end{aligned} \tag{A.28}$$

We have evocatively named this loss $\mathcal{L}_{Wake}$ because we will be optimizing this objective function during the Wake phase of the algorithm. We will also later appeal to the opposite KL divergence, which we will be optimizing during the Sleep phase:

$$\begin{aligned}\mathcal{L}_{Sleep} &= D_{KL}(p_m(\mathbf{r}, \mathbf{s}; \theta_m)||p(\mathbf{r}, \mathbf{s}; \theta)) \\ &= \int \ln\left(\frac{p_m(\mathbf{r}, \mathbf{s}; \theta_m)}{p(\mathbf{r}, \mathbf{s}; \theta)}\right) p_m(\mathbf{r}, \mathbf{s}; \theta_m) d\mathbf{r} d\mathbf{s}. \end{aligned} \tag{A.29}$$

These objectives share a global minimum ($p_m = p$), if it exists, but are not the same objective function, because unlike a traditional distance metric, the KL divergence is not symmetric. However, *near* the global minimum, they become approximately equivalent [Dayan et al. 1995; Bredenberg et al. 2021], which will be an important consideration in assessing the convergence properties of the Wake-Sleep algorithm. Unlike REINFORCE, which will work for any reward function $R(\mathbf{r}, \mathbf{s})$, the Wake-Sleep algorithm will only work for objectives formulated in this way: in this case the choice of objective function is intimately related to the resultant plasticity rule.

A.4.1.1 Equivalence to the Evidence Lower Bound*

It should be noted that $\mathcal{L}_{Wake}$ has a long history in unsupervised machine learning, and does not always appear in the context of training a sensory representational system through normative plasticity. In fact, minimizing $\mathcal{L}_{Wake}$ is equivalent to minimizing the variational free energy or

maximizing the evidence lower bound (ELBO), the objective underlying the variational autoencoder [Rezende et al. 2014; Kingma and Welling 2014] and the Expectation-Maximization algorithm for latent state models [Roweis and Ghahramani 1999]. Here, to help relate to the broader literature, we will elaborate on this equivalence for the interested reader. This section is a technical aside, which the uninterested reader may safely skip. In traditional machine learning terms, as we will see, the $\mathcal{L}_{Wake}$ objective is equivalent to maximizing the ELBO, and will fit a generative model $p_m(\mathbf{r}, \mathbf{s}; \theta_m)$ to data, as well as train a forward map $p(\mathbf{r}|\mathbf{s}; \theta)$ to perform approximate Bayesian inference with respect to that model (i.e. we want $p(\mathbf{r}|\mathbf{s}; \theta) \approx p_m(\mathbf{r}|\mathbf{s}; \theta_m)$).

To fit a generative model to data, we would typically use maximum likelihood estimation: we would find the parameters of our generative model $p_m(\mathbf{r}, \mathbf{s}; \theta_m)$ that match the distribution of data points as accurately as possible by minimizing with respect to θ:

$$D_{KL}(p(\mathbf{s})||p_m(\mathbf{s}; \theta_m)) = \int \ln\left(\frac{p(\mathbf{s})}{p_m(\mathbf{s}; \theta_m)}\right) p(\mathbf{s}) d\mathbf{s}. \tag{A.30}$$

When this objective is 0, samples drawn from $p_m(\mathbf{s}; \theta_m)$ will be indistinguishable from samples drawn from $p(\mathbf{s})$, indicating that we have an accurate model of the data distribution. But we are not only interested in fitting a generative model: when our network receives a stimulus $\mathbf{s}$, we would like it to infer the probability distribution over latent representational states that could correspond to that stimulus, $p_m(\mathbf{r}|\mathbf{s}; \theta_m)$. However, we haven't defined this quantity, only $p_m(\mathbf{r})$ and $p_m(\mathbf{s}|\mathbf{r}; \theta_m)$. From a purely machine learning perspective, we might just try to compute $p_m(\mathbf{r}|\mathbf{s}; \theta_m)$ explicitly using Bayes' Theorem:

$$p_m(\mathbf{r}|\mathbf{s}; \theta_m) = \frac{p_m(\mathbf{r}) p_m(\mathbf{s}|\mathbf{r}; \theta)}{\int p_m(\mathbf{r}) p_m(\mathbf{s}|\mathbf{r}; \theta) d\mathbf{r} d\mathbf{s}}, \tag{A.31}$$

and for simple generative models this might work. However, for complex, nonlinear models, calculating the high-dimensional integral in the denominator analytically is impossible, and

approximating it through Monte Carlo methods is time consuming to the point of intractability. This is motivation enough for machine learning applications, but further, it is not clear how biological system could compute such an integral rapidly upon receiving a single stimulus. So instead, we might try a different approach. We can take our explicitly defined and parameterized forward map $p(\mathbf{r}|\mathbf{s};\theta)$ and train it to approximate $p_m(\mathbf{r}|\mathbf{s};\theta_m)$ as closely as possible by minimizing the expected KL divergence:

$$\mathbb{E}\left[D_{KL}(p(\mathbf{r}|\mathbf{s};\theta)||p_m(\mathbf{r}|\mathbf{s};\theta_m))\right]_{p(\mathbf{s})} = \int \ln\left(\frac{p(\mathbf{r}|\mathbf{s};\theta)}{p_m(\mathbf{r}|\mathbf{s};\theta_m)}\right) p(\mathbf{r}|\mathbf{s};\theta)p(\mathbf{s})d\mathbf{r}d\mathbf{s}. \tag{A.32}$$

If objective is approximately 0, then we do not need to perform Bayes' theorem to calculate the posterior $p_m(\mathbf{r}|\mathbf{s};\theta_m)$, because we have access to a perfect (or near-perfect) approximation $p(\mathbf{r}|\mathbf{s};\theta)$ that we can calculate explicitly or sample from. If $p(\mathbf{r}|\mathbf{s};\theta)$ is parameterized appropriately, this is usually much easier, and potentially could be implemented by a neural network. Now we have two objectives that we want to minimize: one to fit our generative model, and the other to perform approximate inference. It seems natural to add them and minimize them jointly. First, we notice that adding our second objective defines the following inequality:

$$D_{KL}(p(\mathbf{s})||p_m(\mathbf{s};\theta_m)) \leq D_{KL}(p(\mathbf{s})||p_m(\mathbf{s};\theta_m)) + \mathbb{E}\left[D_{KL}(p(\mathbf{r}|\mathbf{s};\theta)||p_m(\mathbf{r}|\mathbf{s};\theta_m))\right]_{p(\mathbf{s})}, \tag{A.33}$$

due to the positivity of the KL divergence. Second, we note that adding these two objectives together really just gives us $\mathcal{L}_{Wake}$:

$$D_{KL}(p(\mathbf{s})||p_m(\mathbf{s};\theta_m)) + \mathbb{E}\left[D_{KL}(p(\mathbf{r}|\mathbf{s};\theta)||p_m(\mathbf{r}|\mathbf{s};\theta_m))\right]_{p(\mathbf{s})} = D_{KL}(p(\mathbf{r},\mathbf{s};\theta)||p_m(\mathbf{r},\mathbf{s};\theta)) \quad \text{(A.34)}$$

$$= \mathcal{L}_{Wake}, \quad \text{(A.35)}$$

where the first equality follows from adding Eqs. A.30 and A.32 and using the properties of the logarithm and expectations.

This alternative construction demonstrates that minimizing that our objective function $\mathcal{L}_{Wake}$ trains our system to perform two separate model-fitting functions: training a generative model and training an approximate inference distribution. From here we can also see its equivalence to the variational free energy and the ELBO:

$$D_{KL}(p(\mathbf{s})||p_m(\mathbf{s};\theta_m)) \leq \mathcal{L}_{Wake} \quad \text{(A.36)}$$

$$= \int \ln\left(\frac{p(\mathbf{r},\mathbf{s};\theta)}{p_m(\mathbf{r},\mathbf{s};\theta_m)}\right) p(\mathbf{r},\mathbf{s};\theta) d\mathbf{r} d\mathbf{s} \quad \text{(A.37)}$$

$$= \int \ln\left(\frac{p(\mathbf{r}|\mathbf{s};\theta)}{p_m(\mathbf{r},\mathbf{s};\theta_m)}\right) p(\mathbf{r},\mathbf{s};\theta) d\mathbf{r} d\mathbf{s} + \int \left(\ln p(\mathbf{s})\right) p(\mathbf{r}|\mathbf{s};\theta) p(\mathbf{s}) d\mathbf{r} d\mathbf{s} \quad \text{(A.38)}$$

$$= \int \ln\left(\frac{p(\mathbf{r}|\mathbf{s};\theta)}{p_m(\mathbf{r},\mathbf{s};\theta_m)}\right) p(\mathbf{r},\mathbf{s};\theta) d\mathbf{r} d\mathbf{s} + \int \left(\ln p(\mathbf{s})\right) p(\mathbf{s}) d\mathbf{s}. \quad \text{(A.39)}$$

Now, by definition $D_{KL}(p(\mathbf{s})||p_m(\mathbf{s};\theta_m)) = \int \left(\ln p(\mathbf{s})\right) p(\mathbf{s}) d\mathbf{s} - \int \left(\ln p_m(\mathbf{s};\theta_m)\right) p(\mathbf{s}) d\mathbf{s}$, the first term of which also appears on the right hand side of our inequality. Furthermore, $\int \left(\ln p(\mathbf{s})\right) p(\mathbf{s}) d\mathbf{s}$ is not a function θ_m or θ, so from the perspective of optimization, it is an irrelevant additive constant. We subtract it from both sides to get:

$$-\int \left(\ln p_m(\mathbf{s};\theta_m)\right) p(\mathbf{s}) d\mathbf{s} \leq \int \ln\left(\frac{p(\mathbf{r}|\mathbf{s};\theta)}{p_m(\mathbf{r},\mathbf{s};\theta_m)}\right) p(\mathbf{r},\mathbf{s};\theta) d\mathbf{r} d\mathbf{s}. \quad \text{(A.40)}$$

This expression on the left is the negative log-likelihood, and the expression on the right is the variational free energy, which is the negative of the ELBO. This shows that $\mathcal{L}_{Wake}$ and the variational free energy differ only by an additive constant from the perspective of optimization: minimizing one is the same as minimizing the other. Similarly, $\mathcal{L}_{Sleep}$ corresponds to an upper bound on the reverse KL divergence, $D_{KL}(p_m(\mathbf{s};\theta_m)||p(\mathbf{s}))$.

A.4.2 Defining p and p_m

Now that we have defined our objective function, we can begin to make things more concrete by defining our probability models p and p_m. Let us start by selecting three features of our representation that we think will be useful, i.e. efficient. First, we want our neurons to be metabolically efficient: a biological system cannot have neurons wasting energetic resources by firing too much [Simoncelli 2003]. One way of requiring this would be to stipulate that the squared norm of our neural firing rate vector, $\|\mathbf{r}\|_2^2$ lies within some reasonable range of activation values. Second, we want to reduce the dimensionality of our representation: many naturalistic datasets are low-dimensional, and it may be wasteful to represent some high-dimensional features of stimuli that are just due to sensor noise. To accomplish this, we will stipulate that $N_r \ll N_s$, where N_r is the representation's dimensionality, and N_s is the stimulus dimension. Third, we will require that individual neural activations should be independent from one another, which will allow individual neurons to extract important features of the data without requiring full knowledge of the activity of other neurons in the representation. To achieve a representation that embodies these three desired features, we define $p_m(\mathbf{r})$ as follows:

$$p_m(\mathbf{r}) \sim \mathcal{N}(0,1), \tag{A.41}$$

i.e. we will require that the representation, averaged over stimuli, will match an N_r-dimensional multivariate normal distribution, where individual axes $\mathbf{r}_i$ are independent from one another

(uncorrelated), and where the normal distribution naturally restricts the probable range of neural activities to lie within bounds determined by the variance (arbitrarily set to 1). Though this distribution captures several intuitions for how neural representations should function, it is clearly a toy model for several reasons: it does not restrict firing rates to be positive, it does not allow for activities to be discrete spikes, it does not account for temporal dynamics, etc. We will discuss later how each of these extensions have been done before, but for now, many features of our model $p_m(\mathbf{r}, \mathbf{s})$ and our forward map $p(\mathbf{r}|\mathbf{s})$ will be unrealistic for didactic purposes.

Now we define the probabilistic decoder $p_m(\mathbf{s}|\mathbf{r})$ (Fig. A.2b), which takes neural firing rates and produces estimates of stimuli, as follows:

$$p_m(\mathbf{s}|\mathbf{r}; \mathbf{W}^{out}) \sim \mathcal{N}(g(\mathbf{W}^{out}\mathbf{r}), \sigma_s), \tag{A.42}$$

where $g(\cdot)$ is an arbitrary nonlinearity, and σ_s^2 is the variance of the decoder. In this probability distribution, we will treat the $N_s \times N_r$ matrix $\mathbf{W}^{out}$ as a free parameter which we will train to optimize our objective.

Similarly, we can define the forward map $p(\mathbf{r}|\mathbf{s})$, which takes environmental stimuli and produces firing rates, as follows:

$$p(\mathbf{r}|\mathbf{s}; \mathbf{W}^{in}) = \mathcal{N}(f(\mathbf{W}^{in}\mathbf{s}), \sigma_r), \tag{A.43}$$

where $f(\cdot)$ is an arbitrary (potentially different) nonlinearity, and σ_r will ultimately correspond to intrinsic neural variability. Here, the $N_r \times N_s$ matrix $\mathbf{W}^{in}$ is the free parameter. Thus, $\mathbf{W}^{in}$ and $\mathbf{W}^{out}$, are the free parameters in our simple construction.

We have not yet made clear how these parameters and functions could map onto an actual neural architecture: we will do this after defining the learning algorithm, so that it is clear what the necessary components of the algorithm are. Interestingly, we do not have to define $p(\mathbf{s})$ at all. This distribution is determined by the environment. In fact, a learning system should ideally

be as agnostic as possible to the specific form of $p(\mathbf{s})$, in order to be able to adapt strange and unforeseen changes in the statistics of the world. The Wake-Sleep algorithm is ideal in that it makes little-to-no assumption about $p(\mathbf{s})$, but as we will see, it may perform poorly if it is not possible to obtain a close match between p and p_m. This might occur if the environmental distribution of $\mathbf{s}$ is much higher dimensional than the number of neurons, or is in some other way more complex than the generative model.

A.4.3 Approximating the loss gradient

Having defined our objective function and probability distributions p and p_m, we can now derive the Wake-Sleep algorithm. First, we will show that we can obtain a promising update for $\mathbf{W}^{out}$ by performing gradient descent on $\mathcal{L}_{Wake}$ (the Wake phase of learning). We will next show that we can obtain a similarly promising update for $\mathbf{W}^{in}$ by performing gradient descent on $\mathcal{L}_{Sleep}$ (the Sleep phase of learning). One might easily wonder why we did not perform gradient descent on $\mathcal{L}_{Wake}$ with respect to $\mathbf{W}^{in}$, instead of $\mathcal{L}_{Sleep}$: we will next show why it would be a bad idea to do this. Lastly, we will describe two perspectives on how these resultant updates can be viewed as a unified form of approximate optimization.

A.4.3.1 Wake

We start by calculating the negative gradient of $\mathcal{L}_{Wake}$ with respect to a particular parameter $\mathbf{W}^{out}_{ij}$ from $p_m(\mathbf{s}|\mathbf{r};\mathbf{W}^{out})$:

$$-\frac{d\mathcal{L}_{Wake}}{d\mathbf{W}_{ij}^{out}} = -\frac{d}{d\mathbf{W}_{ij}^{out}} \int \ln\left(\frac{p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})}{p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})}\right) p(\mathbf{r},\mathbf{s};\mathbf{W}^{in}) d\mathbf{r}d\mathbf{s} \tag{A.44}$$

$$= -\frac{d}{d\mathbf{W}_{ij}^{out}} \int \left[\ln p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in}) - \ln p_m(\mathbf{s}|\mathbf{r};\mathbf{W}^{out}) - \ln p_m(\mathbf{r})\right] p(\mathbf{r},\mathbf{s};\mathbf{W}^{in}) d\mathbf{r}d\mathbf{s} \tag{A.45}$$

$$= -\int \frac{d}{d\mathbf{W}_{ij}^{out}} \left[\ln p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in}) - \ln p_m(\mathbf{s}|\mathbf{r};\mathbf{W}^{out}) - \ln p_m(\mathbf{r})\right] p(\mathbf{r},\mathbf{s};\mathbf{W}^{in}) d\mathbf{r}d\mathbf{s} \tag{A.46}$$

$$= \int \left[\frac{d}{d\mathbf{W}_{ij}^{out}} \ln p_m(\mathbf{s}|\mathbf{r};\mathbf{W}^{out})\right] p(\mathbf{r},\mathbf{s};\mathbf{W}^{in}) d\mathbf{r}d\mathbf{s} \tag{A.47}$$

Plugging in the probability density function for $p_m(\mathbf{s}|\mathbf{r};\mathbf{W}^{out}) \sim \mathcal{N}(g(\mathbf{W}^{out}\mathbf{r}), \sigma_s^2)$, we end up with:

$$-\frac{d\mathcal{L}_{Wake}}{d\mathbf{W}_{ij}^{out}} = \int \left[\frac{d}{d\mathbf{W}_{ij}^{out}} \frac{1}{2\sigma_s^2} \sum_{i=0}^{N_s} (\mathbf{s} - g(\mathbf{W}^{out}\mathbf{r}))^2\right] p(\mathbf{r},\mathbf{s};\mathbf{W}^{in}) d\mathbf{r}d\mathbf{s}. \tag{A.48}$$

Similar to our derivation for REINFORCE, we see that for a particular weight $\mathbf{W}_{ij}^{out}$, $\frac{dg_l(\mathbf{W}^{out}\mathbf{r})}{d\mathbf{W}_{ij}^{out}} = 0$ if $i \neq l$. Thus, we have:

$$-\frac{d\mathcal{L}_{Wake}}{d\mathbf{W}_{ij}^{out}} = \int \frac{1}{\sigma_s^2} \left[(\mathbf{s}_i - g_i(\mathbf{W}^{out}\mathbf{r}))g_i'(\mathbf{W}^{out}\mathbf{r})\mathbf{r}_j\right] p(\mathbf{r},\mathbf{s};\mathbf{W}^{in}) d\mathbf{r}d\mathbf{s}. \tag{A.49}$$

Again, similar to REINFORCE, we can approximate this update as the network actively 'perceives': we receive a sampled environmental stimulus $\mathbf{s}^{(k)}$, and then sample from the probability distribution $p(\mathbf{r}|\mathbf{s}^{(k)};\mathbf{W}^{in})$ to obtain a firing rate sample $\mathbf{r}^{(k)}$. Then across K samples, we calculate the approximate parameter update:

$$\Delta\mathbf{W}_{ij}^{out} \propto \frac{1}{\sigma_s^2 K} \sum_{k=0}^{K} \left[(\mathbf{s}_i^{(k)} - g_i(\mathbf{W}^{out}\mathbf{r}^{(k)}))g_i'(\mathbf{W}^{out}\mathbf{r}^{(k)})\mathbf{r}_j^{(k)}\right] \approx -\frac{d\mathcal{L}_{Wake}}{d\mathbf{W}_{ij}^{out}}. \tag{A.50}$$

If we want learning to be able to occur online (Section 2.2.5), then we can take $K = 1$, and sacrifice some precision of our estimate. This update has the form of a prediction error, where the the error between the true stimulus $\mathbf{s}_i^{(k)}$ and the network's decoded estimate $g_i(\mathbf{W}^{out}\mathbf{r}^{(k)})$ combine with presynaptic inputs $r_j^{(k)}$ to produce parameter updates. In Section A.4.4 we will analyze in detail how this parameter update could correspond to a local synaptic update for a particular neuron model.

A.4.3.2 Sleep

So far, other than performing stochastic gradient descent over K samples, we have introduced no approximation into our algorithm. We might be tempted to perform gradient descent on $\mathcal{L}_{Wake}$ with respect to $\mathbf{W}^{in}$ too: though we will defer the discussion of this point for later, it turns out to be a bad idea (see Section A.4.4.1). Instead, we will perform an *almost identical* procedure, but perform gradient descent on $\mathcal{L}_{Sleep}$ instead. As discussed in Section 3.5, one way of interpreting this change in loss is that we now have two different sets of parameters (i.e. synapses) in our system, $\mathbf{W}^{in}$ and $\mathbf{W}^{out}$ which are optimizing two different, albeit closely related objectives, $\mathcal{L}_{Sleep}$ and $\mathcal{L}_{Wake}$, respectively. An alternative perspective that we will discuss is that $\mathbf{W}^{in}$ is also optimizing $\mathcal{L}_{Wake}$, but is only performing an approximate gradient descent. We will discuss in Section A.4.4.2 how this added complexity affects the convergence and quality of the algorithm. Starting with $\mathcal{L}_{Sleep}$, we have:

$$-\frac{d\mathcal{L}_{Sleep}}{d\mathbf{W}_{ij}^{in}} = -\frac{d}{d\mathbf{W}_{ij}^{in}} \int \ln\left(\frac{p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})}{p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})}\right) p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out}) d\mathbf{r}d\mathbf{s} \tag{A.51}$$

$$= \int \left[\frac{d}{d\mathbf{W}_{ij}^{in}} \frac{1}{2\sigma_r^2} \sum_{i=0}^{N_r} (\mathbf{r} - f(\mathbf{W}^{in}\mathbf{s}))^2\right] p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out}) d\mathbf{r}d\mathbf{s}, \tag{A.52}$$

where we have followed exactly the same steps as in Eqs. A.44-A.48.

As before, we notice that for a particular weight $\mathbf{W}^{in}_{ij}$, $\frac{df_l(\mathbf{W}^{in}\mathbf{s})}{d\mathbf{W}^{in}_{ij}} = 0$ if $i \neq l$. Thus, we have:

$$-\frac{d\mathcal{L}_{Sleep}}{d\mathbf{W}^{in}_{ij}} = \int \frac{1}{\sigma_r^2}\left[(\mathbf{r}_i - f_i(\mathbf{W}^{in}\mathbf{s}))f_i'(\mathbf{W}^{in}\mathbf{s})\mathbf{s}_j\right] p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})d\mathbf{r}d\mathbf{s}. \tag{A.53}$$

Now we can approximate this update with samples from $p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})$. Notice that we are no longer actively perceiving via the forward mapping $p(\mathbf{r}|\mathbf{s})$ in response to sampled environmental stimuli. Instead, activity is first internally generated via $\mathbf{r}^{(k)} \sim p_m(\mathbf{r})$, before propagating to the stimulus layer to produce artificial stimuli via $\mathbf{s}^{(k)} \sim p_m(\mathbf{s}|\mathbf{r}^{(k)};\mathbf{W}^{out})$. This is termed the Sleep phase of the algorithm evocatively: an animal could not perform this type of learning while actively moving through an environment, and if it did perceive, such percepts would appear hallucinatory or dream-like, being reflective of the animal's model rather than reality. Given our K samples, we calculate the approximate parameter update:

$$\Delta\mathbf{W}^{in}_{ij} \propto \frac{1}{\sigma_r^2 K}\sum_{k=0}^{K}\left[(\mathbf{r}_i^{(k)} - f_i(\mathbf{W}^{in}\mathbf{s}^{(k)}))f_i'(\mathbf{W}^{in}\mathbf{s}^{(k)})\mathbf{s}_j^{(k)}\right] \approx -\frac{d\mathcal{L}_{Sleep}}{d\mathbf{W}^{in}_{ij}}. \tag{A.54}$$

Now, this update should look almost equivalent to the Wake update for $\mathbf{W}^{out}$ (Eq. A.50). As with the Wake update, if we want learning to occur online we can take $K = 1$. It turns out that the variability induced by this sampled approximation is *much* less than the variability induced by the REINFORCE algorithm, and is the chief reason for its superior performance and scalability (Appendix B.3). However, it is very important to note that we are sampling from p_m instead of p. Because our two parameter updates, Eq. A.50 and Eq. A.54 require sampling from two different probability distributions and individual neurons $\mathbf{r}$ could only be sampling from one probability distribution at a time, the updates are necessarily computed during different *phases*. The Wake-Sleep algorithm consists of alternating between sampling from p to compute updates

for $\mathbf{W}^{out}$ (the Wake phase; Eq. A.50) and sampling from p_m to compute updates for $\mathbf{W}^{in}$ (the Sleep phase; Eq. A.54). As we discuss in Section A.4.4, we should be appropriately cautious about what these alternative phases could possibly mean for a biological organism.

A.4.4 Assessing Wake Sleep

Having derived our Wake-Sleep parameter updates, we are finally in a position to assess the degree to which it satisfies our desiderata. We have provided a very simplified derivation of the Wake-Sleep algorithm, for a single-layer rate-based network. However, the algorithm generalizes well to recurrent, spiking, and multilayer architectures [Dayan and Hinton 1996] (Section 2.2.3), and these modifications do make the algorithm more realistic as a normative plasticity model. However, it will still be very useful to show how the various components of the algorithm as we have derived it could potentially map onto realistic biological structures (Fig. A.2c). First of all, we observe that both $\mathbf{s}$ and $\mathbf{r}$ need to be able to sample from either p_m or p—for this to be possible, $\mathbf{s}$ must be *internal* to the brain, since sampling from p_m affects both $\mathbf{r}$ and $\mathbf{s}$ simultaneously and would have to occur while an animal is not consciously acting in its environment. Therefore, it is best to think of $\mathbf{s}$ as a stimulus layer of neurons, and of $\mathbf{r}$ as a downstream layer of neurons receiving feedforward inputs. Next, we suppose that there is a global gating signal γ that determines the phase of the network— if $\gamma = 1$, the network is in the Wake phase, and if $\gamma = 0$, the network is in the Sleep phase. Now we observe that the following equations will produce valid samples:

$$\mathbf{r} = \gamma f(\mathbf{W}^{in}\mathbf{s}) + (\gamma\sigma_r + (1-\gamma))\boldsymbol{\eta}_r \tag{A.55}$$

$$\mathbf{s} = \gamma\mathbf{s}_p + (1-\gamma)(g(\mathbf{W}^{out}\mathbf{r}) + \sigma_s\boldsymbol{\eta}_s), \tag{A.56}$$

where $\mathbf{s}_p \sim p(\mathbf{s})$ is an incoming sensory input, and $\boldsymbol{\eta}_s, \boldsymbol{\eta}_r \sim \mathcal{N}(0,1)$ are sources of intrinsic noise for neurons in the stimulus, and downstream layers, respectively. Because p_m and p both

assume exactly the same dimensionality of $\mathbf{s}$ and $\mathbf{r}$, the only reasonable mapping of these two different sampling phases is onto one neuron with two different *modes* of activity. In Figure A.2c, we show that one possible biological mapping is to propose that feedforward inputs (active when $\gamma = 1$) to the basal dendrites of pyramidal neurons allow neurons to sample from p, and top-down inputs (active when $\gamma = 0$) to the apical dendrites of pyramidal neurons allows neurons to sample from p_m: interestingly, a corollary of this mapping is that a network could achieve 'detachability' by manipulating γ to generate sample network states in the absence of stimuli.

It is important to note that several normative plasticity models have proposed that top-down signals to the apical dendrites could serve as some form of training signal. We will adopt a similar attitude, and now assess the locality of the Wake-Sleep parameter updates with respect to this model formulation. If we take the sample size for our updates to be $K = 1$, based on Eqs. A.50 and A.54, for a single pair of samples $\mathbf{r}$, $\mathbf{s}$, we have:

$$\Delta \mathbf{W}^{in}_{ij} \propto \frac{1-\gamma}{\sigma_r^2} \left[(\mathbf{r}_i - f_i(\mathbf{W}^{in}\mathbf{s})) f_i'(\mathbf{W}^{in}\mathbf{s}) \mathbf{s}_j \right] \tag{A.57}$$

$$\Delta \mathbf{W}^{out}_{ij} \propto \frac{\gamma}{\sigma_s^2} \left[(\mathbf{s}_i - g_i(\mathbf{W}^{out}\mathbf{r})) g_i'(\mathbf{W}^{out}\mathbf{r}) \mathbf{r}_j \right]. \tag{A.58}$$

As with REINFORCE, both σ_r and σ_s are proportionality constants and can be disregarded. For $\Delta \mathbf{W}^{in}_{ij}$, a basal synapse on $\mathbf{r}_i$, several variables are required. First, the same signal that gates the influence of apical versus basal inputs, γ, must also *deactivate* plasticity at basal synapses. γ could be implemented in a neural circuit by either global inhibitory gating or by a neuromodulatory signal [Bredenberg et al. 2021]—whichever candidate signal would also have to gate plasticity. The synapse needs the postsynaptic firing rate $\mathbf{r}_i$, which is readily available, and a subtracted measure of current local to the basal compartment, $f_i(\mathbf{W}^{in}\mathbf{s})$—there is some indication that local dendritic voltage levels can affect synaptic plasticity, but the sign and exact form of this effect is variable across studies [Letzkus et al. 2006; Froemke et al. 2005; Sjöström and Häusser 2006]. As

with REINFORCE, the synapse would require $f_i'(\mathbf{W}^{in}\mathbf{s})$, which is simply a monotonic function of $(\mathbf{W}^{in}\mathbf{s})_i$, and could be easily approximated; lastly, it would need the presynaptic firing rate $\mathbf{s}_j$. The information requirements for $\mathbf{W}^{out}_{ij}$ are almost exactly the same.

In terms of requiring only functions of pre- and postsynaptic activity, with the addition of some limited global context signal γ, these plasticity rules are plausibly local (Section 2.2.1). However, several features of this setup are unconfirmed, the most obviously testable being the Wake-Sleep sampling dynamics postulated by Eqs. A.57 and A.58: it seems unlikely that a neural network would entirely and synchronously switch into a 'generative' or hallucinatory regime for an extended period of time when $\gamma = 0$, and such a regime could not possibly occur in an awake, behaving animal, meaning that $\mathbf{W}^{in}$ could not be learned online (Section 2.2.5). However, we have proposed softer form of Wake-Sleep (Chapter 5), which does allow for online learning, and does not interfere with active perception, suggesting that the principles established by Wake-Sleep may extend to more realistic formulations of γ. The strongest test (Section 2.2.7) of this family of algorithms is that artificially magnifying the influence of apical dendrites in a neural circuit should induce generative sampling, i.e. hallucination; other models of apical dendritic learning [Sacramento et al. 2017; Guerguiev et al. 2017; Payeur et al. 2021; Urbanczik and Senn 2014] do not propose this as a mechanism. Notice that this prediction requires our specific mapping of the Wake-Sleep algorithm onto neural circuitry: other interpretations are conceivable, and would have different predictions.

As we have discussed in Section A.4.1, the Wake-Sleep algorithm is capable of optimizing a broad range of *unsupervised* learning objectives, considerably more general than for instance Oja's rule [Oja 1982] (though the specific toy example we provide is just a nonlinear form of probabilistic PCA). Unlike REINFORCE, the Wake-Sleep algorithm is unable to optimize reinforcement learning objectives, however, within the range of objectives that Wake-Sleep *can* optimize, it is typically much more scalable than REINFORCE (Section 2.2.6)[2]: in this way, it is an ideal complement,

[2]Though it still performs worse than backpropagation [Kingma and Welling 2014; Rezende et al. 2014].

and having both algorithms or some hybridized form present in a neural circuit could be very powerful. The Wake-Sleep algorithm involves more approximation than REINFORCE. One could very easily wonder: since we have presented two sets of parameters in the Wake-Sleep algorithm minimizing two different objective functions, why should we expect the algorithm to converge or reliably improve performance on either objective?

To this point, we have identified two strange features of the Wake-Sleep algorithm that go hand-in-hand. First, it is strange that we should require a period of hallucinatory activity to train our parameters. Second, it is hard to interpret the convergence of an algorithm that is alternatively minimizing two slightly different objective functions: why all the work and extra conceptual baggage? Why not just do approximate gradient descent as we did with the REINFORCE algorithm and be done with it? In Section A.4.4.1 we will motivate why more standard gradient descent methods are not appropriate for this type of unsupervised learning, and in Section A.4.4.2 we will address the convergence properties of the Wake-Sleep algorithm from two different perspectives, explaining why the algorithm has such good empirical performance despite its approximations.

A.4.4.1 Why gradient descent with $\mathbf{W}^{in}$ won't work

Sometimes, to genuinely understand an algorithm, it's important to understand the weaknesses of alternative approaches. For didactic reasons, we will explore what happens if we simply take the gradient of $\mathcal{L}_{Wake}$ with respect to $\mathbf{W}^{in}$. We have:

$$-\frac{d\mathcal{L}_{Wake}}{d\mathbf{W}_{ij}^{in}} = -\frac{d}{d\mathbf{W}_{ij}^{in}}\int \ln\left(\frac{p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})}{p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})}\right) p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s} \tag{A.59}$$

$$= -\int \left[\frac{d}{d\mathbf{W}_{ij}^{in}} \ln\left(\frac{p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})}{p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})}\right)\right] p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s}$$

$$-\int \ln\left(\frac{p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})}{p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})}\right) \frac{d}{d\mathbf{W}_{ij}^{in}} p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s} \tag{A.60}$$

$$= -\int \left[\frac{d}{d\mathbf{W}_{ij}^{in}} \ln\left(p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})\right)\right] p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s}$$

$$-\int \ln\left(\frac{p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})}{p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})}\right) \frac{d}{d\mathbf{W}_{ij}^{in}} p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s}, \tag{A.61}$$

where the second equality follows from the product rule, and the third equality follows from the fact that $\ln p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})$ does not depend on $\mathbf{W}^{in}$. Interestingly, the first term in this equation is zero. To see this, we note the following sequence of identities:

$$\int \left[\frac{d}{d\mathbf{W}_{ij}^{in}} \ln\left(p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})\right)\right] p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s} = \int \left[\frac{d}{d\mathbf{W}_{ij}^{in}} e^{\ln\left(p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})\right)}\right] d\mathbf{r}d\mathbf{s}$$

$$= \int \frac{d}{d\mathbf{W}_{ij}^{in}} p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s}$$

$$= \frac{d}{d\mathbf{W}_{ij}^{in}} \int p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s} = \frac{d}{d\mathbf{W}_{ij}^{in}} 1 = 0. \tag{A.62}$$

The first term is zero, which leaves only the second term of Eq. A.61. It gives us:

$$-\frac{d\mathcal{L}_{Wake}}{d\mathbf{W}_{ij}^{in}} = -\int \ln\left(\frac{p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})}{p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})}\right) \frac{d}{d\mathbf{W}_{ij}^{in}} p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s} \tag{A.63}$$

$$= \int \ln\left(\frac{p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})}{p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})}\right) \left(\frac{d}{d\mathbf{W}_{ij}^{in}} \ln p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})\right) p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s}, \tag{A.64}$$

where for the second equality we have once again used the identity in Eq. A.62. Fascinatingly enough, this is exactly equivalent to the REINFORCE update (Eq. A.18), if we take $R(\mathbf{r}, \mathbf{s}) = \ln(p/p_m)$. Though the REINFORCE update might be practical for environmental rewards that an animal might receive, this particular choice of $R(\mathbf{r}, \mathbf{s})$ requires detailed knowledge of the inner workings of a neural representation. Not only is it not possible for an environmental signal to carry this information, there is no evidence that any neuromodulatory center in the brain is able to compute such a complicated signal based on neural network activity. Thus, even though this update appears to have the form of a reward-modulated Hebbian plasticity rule, there is very little reason to believe that it is local (Section 2.2.1). Furthermore, this form of update is well-known to have severe scalability (Section 2.2.6) issues, and demonstrably performs worse than Wake-Sleep on high-dimensional datasets (Chapter 5; [Werfel et al. 2003]). The Wake-Sleep algorithm is very much a response to these failings, using a local error signal specific to each neuron, rather than correlating each neuron's activity with a global reward signal. However, the Wake-Sleep algorithm employs more approximations than REINFORCE. In Section A.4.4.2, we will analyze the convergence properties of Wake-Sleep.

A.4.4.2 The convergence of Wake-Sleep

Currently, we have two updates that are approximating gradient descent on two different objectives: $\Delta \mathbf{W}_{ij}^{out} \approx -\lambda \frac{d\mathcal{L}_{Wake}}{d\mathbf{W}_{ij}^{out}}$, and $\Delta \mathbf{W}_{ij}^{in} \approx -\lambda \frac{d\mathcal{L}_{Sleep}}{d\mathbf{W}_{ij}^{in}}$, where λ is a small positive learning rate. In Section 3.5, we stressed the importance of viewing plasticity updates as decreasing a *unified* objective, but here we have two. How do we know that $\Delta \mathbf{W}_{ij}^{in}$ won't *increase* $\mathcal{L}_{Wake}$ and vice versa? Clearly, $\mathcal{L}_{Sleep}$ and $\mathcal{L}_{Wake}$ are closely related: one way of resolving this difficulty is by demonstrating that $\Delta \mathbf{W}_{ij}^{in} \approx -\lambda \frac{d\mathcal{L}_{Wake}}{d\mathbf{W}_{ij}^{in}}$. In this case, during the Wake phase, the system would optimize $\mathcal{L}_{Wake}$ with respect to $\mathbf{W}^{out}$, and during the Sleep phase, it would approximately optimize the same objective with respect to $\mathbf{W}^{in}$—this would amount to an approximation of coordinate descent. In fact, under certain conditions, it turns out that this is exactly what the Wake-Sleep algorithm is doing.

To see this, we begin with the REINFORCE-like update (Eq. A.64) for gradient descent on $\mathcal{L}_{Wake}$:

$$-\frac{d\mathcal{L}_{Wake}}{d\mathbf{W}^{in}_{ij}} = \int \ln\left(\frac{p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})}{p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})}\right)\left(\frac{d}{d\mathbf{W}^{in}_{ij}}\ln p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})\right)p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s}. \tag{A.65}$$

Interestingly, we notice that if $p_m \approx p$, then by first-order Taylor expansion, $\ln(p_m/p) \approx p_m/p - 1$. Plugging this approximation in (see Appendix B.1 for a more detailed justification of this approximation), we get:

$$-\frac{d\mathcal{L}_{Wake}}{d\mathbf{W}^{in}_{ij}} \approx \int \left(\frac{p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})}{p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})} - 1\right)\left(\frac{d}{d\mathbf{W}^{in}_{ij}}\ln p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})\right)p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})d\mathbf{r}d\mathbf{s} \tag{A.66}$$

$$= \int \left(\frac{d}{d\mathbf{W}^{in}_{ij}}\ln p(\mathbf{r},\mathbf{s};\mathbf{W}^{in})\right)p_m(\mathbf{r},\mathbf{s};\mathbf{W}^{out})d\mathbf{r}d\mathbf{s} \tag{A.67}$$

$$= -\frac{d\mathcal{L}_{Sleep}}{d\mathbf{W}^{in}_{ij}}, \tag{A.68}$$

where for the first equality we have once again used the identity Eq. A.62. Essentially, if a global optimum such that $p_m = p$ exists, it is shared by both $\mathcal{L}_{Wake}$ and $\mathcal{L}_{Sleep}$. Thus, we can expect the gradients of these two objective functions to behave very similarly if p_m is close to p. Because the Wake phase (updating $\mathbf{W}^{out}$) occurs without approximation, the algorithm has the opportunity to enter this regime before the approximating Sleep phase ever occurs.

An alternative analysis of the Wake-Sleep algorithm [Dayan et al. 1995] observes that for fixed $\mathbf{W}^{out}$, $\mathcal{L}_{Sleep}$ and $\mathcal{L}_{Wake}$ share a global minimum with respect to $\mathbf{W}^{in}$ when $p_m(\mathbf{r}|\mathbf{s};\mathbf{W}^{out}) = p(\mathbf{r}|\mathbf{s};\mathbf{W}^{in})$, as long as there exists a $\mathbf{W}^{in}_{opt}$ such that this equality holds. If $\mathcal{L}_{Sleep}$ is convex and this global minimum is attainable, fully optimizing $\mathcal{L}_{Sleep}$ with respect to $\mathbf{W}^{in}$ during the Sleep phase is therefore guaranteed to also optimize $\mathcal{L}_{Wake}$. Therefore, as long as these to conditions of convexity

and attainability of the global minimum are satisfied (they are not in general, but do hold for simple examples like Factor Analysis [Amari and Nakahara 1999]), both phases decrease $\mathcal{L}_{Wake}$. Rather than an approximation of coordinate descent, this can be viewed as an approximation of the Expectation-Maximization (EM) algorithm [Dempster et al. 1977].

We see that there are two different ways of interpreting Wake-Sleep: first, it is an approximation of coordinate descent that becomes a better approximation the closer to the optimum it becomes. Second, under restricted conditions, Wake-Sleep can be viewed as an approximation of the EM algorithm. Both of these perspectives are conditional on assumptions about the probability models being trained, requiring a generative model $p_m(\mathbf{r}, \mathbf{s})$ and a forward map $p(\mathbf{r}|\mathbf{s})$ capable of mutually reaching good performance for an environmental stimulus distribution $p(\mathbf{s})$. Though Wake-Sleep empirically performs quite well under a variety of stimulus conditions and network models [Dayan and Hinton 1996], these are important caveats: the comparative weakness of the demonstrations of Wake-Sleep's convergence relative to gradient descent or EM is a common point of criticism of the algorithm [Rezende et al. 2014; Kingma and Welling 2014; Mnih and Gregor 2014].

B | Appendix: Impression learning

B.1 Bias calculation

Our derivation of the update for IL (Eq. 3) is based on an expansion of $\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}$ about $\frac{\tilde{p}_\theta}{\tilde{q}_\theta} = 1$:

$$\int \left[\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right] (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \, d\mathbf{r}d\mathbf{s} = \int \left[\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1\right] (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \, d\mathbf{r}d\mathbf{s} \\ - \frac{1}{2}\int \left[\frac{(\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1)}{1 + \epsilon(\mathbf{r}, \mathbf{s})}\right]^2 (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \, d\mathbf{r}d\mathbf{s}, \tag{B.1}$$

for some $\epsilon(\mathbf{r}, \mathbf{s})$ st. $|\epsilon(\mathbf{r}, \mathbf{s})| < |\frac{\tilde{p}}{\tilde{q}} - 1|$. Note that this is not a truncated Taylor series approximation: we are instead using Taylor's theorem, and the second term provides an exact expression for the bias. We can use the Caucy-Schwartz inequality for expectations to bound this as follows:

$$|\text{bias}| = \frac{1}{2}\left|\int \left[\frac{(\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1)}{1 + \epsilon(\mathbf{r}, \mathbf{s})}\right]^2 (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \, d\mathbf{r}d\mathbf{s}\right| \\ \leq \frac{1}{2}\sqrt{\int \left[\frac{(\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1)}{1 + \epsilon(\mathbf{r}, \mathbf{s})}\right]^4 \tilde{q}_\theta \, d\mathbf{r}d\mathbf{s}}\sqrt{\int (\nabla_\theta \log \tilde{q}_\theta)^2 \tilde{q}_\theta \, d\mathbf{r}d\mathbf{s}}, \tag{B.2}$$

We examine the consequences of this bias formula for our specific model. Consider the

component of the gradient with respect to the feedforward weight $\mathbf{W}^{(ij)}$:

$$\frac{d}{dW^{(ij)}} \log \tilde{q}_\theta = \sum_t \frac{\lambda_t}{(\sigma_r^{\text{inf}})^2} (\mathbf{r}_t^{(i)} - f(\mathbf{W}\mathbf{s}_t)^{(i)}) f'(\mathbf{W}\mathbf{s}_t)^{(i)} \mathbf{s}_t^{(j)}.$$

Note that $f(\cdot) < 1$ and $f'(\cdot) < 1$ for the tanh function, and assume that $(s_t^{(j)})^2 < S \ \ \forall t$ for some constant S. Defining $B = \sqrt{\int \left[\frac{(\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1)}{1 + \epsilon(\mathbf{r}, \mathbf{s})} \right]^4 \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s}}$, and substituting the gradient component gives:

$$\begin{aligned}
|\text{bias}| &\leq \frac{B}{2} \sqrt{\int \left(\sum_t \frac{\lambda_t}{(\sigma_r^{\text{inf}})^2} (\mathbf{r}_t^{(i)} - f(\mathbf{W}\mathbf{s}_t)^{(i)}) f'(\mathbf{W}\mathbf{s}_t)^{(i)} \mathbf{s}_t^{(j)} \right)^2 \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s}} \\
&= \frac{B}{2} \sqrt{\int \sum_t \sum_{t'} \frac{\lambda_t \lambda_{t'}}{(\sigma_r^{\text{inf}})^4} (\mathbf{r}_t^{(i)} - f(\mathbf{W}\mathbf{s}_t)^{(i)}) (\mathbf{r}_{t'}^{(i)} - f(\mathbf{W}\mathbf{s}_{t'})^{(i)}) f'(\mathbf{W}\mathbf{s}_t)^{(i)} f'(\mathbf{W}\mathbf{s}_{t'})^{(i)} \mathbf{s}_t^{(j)} \mathbf{s}_{t'}^{(j)} \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s}} \\
&= \frac{B}{2} \sqrt{\int \sum_t \frac{\lambda_t^2}{(\sigma_r^{\text{inf}})^4} (\mathbf{r}_t^{(i)} - f(\mathbf{W}\mathbf{s}_t)^{(i)})^2 (f'(\mathbf{W}\mathbf{s}_t)^{(i)} \mathbf{s}_t^{(j)})^2 \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s}},
\end{aligned}$$

where this second equality follows from the fact that $\mathbf{r}_t^{(i)} - f(\mathbf{W}\mathbf{s}_t)^{(i)} \sim \mathcal{N}(0, \sigma_r^{\text{inf}})$ without any temporal correlation, so that $\mathbb{E}\left[(\mathbf{r}_t^{(i)} - f(\mathbf{W}\mathbf{s}_t)^{(i)}) (\mathbf{r}_{t'}^{(i)} - f(\mathbf{W}\mathbf{s}_{t'})^{(i)}) \right]_{\mathbf{r}|\mathbf{s}} = 0$ for $t \neq t'$. Continuing our derivation, we have:

$$\begin{aligned}
|\text{bias}| &\leq \frac{B}{2} \sqrt{\sum_t \frac{\lambda_t^2}{(\sigma_r^{\text{inf}})^4} \int (\mathbf{r}_t^{(i)} - f(\mathbf{W}\mathbf{s}_t)^{(i)})^2 (f'(\mathbf{W}\mathbf{s}_t)^{(i)} \mathbf{s}_t^{(j)})^2 \tilde{q}_\theta(\mathbf{r}, \mathbf{s}) \, d\mathbf{r} d\mathbf{s}} \\
&= \frac{B}{2} \sqrt{\sum_t \frac{\lambda_t^2}{(\sigma_r^{\text{inf}})^2} \int (f'(\mathbf{W}\mathbf{s}_t)^{(i)} \mathbf{s}_t^{(j)})^2 \tilde{q}_\theta(\mathbf{s}) \, d\mathbf{s}} \\
&\leq \frac{B}{2} \sqrt{\frac{S}{(\sigma_r^{\text{inf}})^2} \sum_t \lambda_t^2} \\
&= \frac{B}{2} \sqrt{\frac{ST}{2(\sigma_r^{\text{inf}})^2}},
\end{aligned} \tag{B.3}$$

where T is the total time. Thus, for our particular choice of neural model, the bias is proportional to B, which vanishes as performance improves. Note that the update term in Eq. (B.1) is $O(|\frac{\tilde{p}}{\tilde{q}} - 1|)$,

so its magnitude is expected to be much larger than the bias in the vicinity of a global optimum. The $\sqrt{T/(\sigma_r^{\text{inf}})^2}$ proportionality constant also should not be a cause for concern: the gradient itself scales with $T/(\sigma_r^{\text{inf}})^2$, and thus small values of $(\sigma_r^{\text{inf}})^2$ will not make the relative error explode.

B.2 Comparison to other algorithms

In this section, we explore the relationships between impression learning (IL) and other stochastic learning algorithms. Specifically, we consider a variant of neural variational inference (NVI*), backpropagation (BP), and Wake-Sleep (WS).

B.2.1 Neural Variational Inference

Neural variational inference is a learning algorithm for neural networks very closely related to REINFORCE (Appendix A.3) that optimizes the evidence lower bound (ELBO) objective function. Here, we modify the algorithm by incorporating our novel loss (Eq. 2), producing a variant that we call NVI*. We first take the derivative of our loss, without approximations. These steps are identical to the initial steps in our derivation of IL, up to the Taylor expansion:

$$\begin{aligned}
-\nabla_\theta \mathcal{L} =& -\nabla_\theta \mathbb{E}_{\lambda,\mathbf{z}}\left[\int \left[\log \tilde{q}_\theta - \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s}\right] \\
=& -\mathbb{E}_{\lambda,\mathbf{z}}\left[\int \left[\nabla_\theta(\log \tilde{q}_\theta - \log \tilde{p}_\theta)\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} + \int \left[\log \tilde{q}_\theta - \log \tilde{p}_\theta\right] \nabla_\theta \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s}\right] \\
=& -\mathbb{E}_{\lambda,\mathbf{z}}\left[\int \left[\nabla_\theta \log \tilde{q}_\theta - \nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} + \int \left[\log \tilde{q}_\theta - \log \tilde{p}_\theta\right] (\nabla_\theta \log \tilde{q}_\theta) \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s}\right] \\
=& \mathbb{E}_{\lambda,\mathbf{z}}\left[\int \left[\nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} + \int \left[\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right] (\nabla_\theta \log \tilde{q}_\theta) \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s}\right]
\end{aligned} \tag{B.4}$$

Updates calculated by these samples will be unbiased in expectation, because there are no approximations. However, we will show in Appendix B.3 that these updates may have high variance.

To provide a fair comparison to IL, we have added two additional features that have been shown to reduce the variance of sample estimates [Ranganath et al. 2014; Mnih and Gregor 2014]. The first involves subtracting a control variate from our second term:

$$-\nabla_\theta \mathcal{L} = \mathbb{E}_{\lambda,\mathbf{z}}\left[\int \left[\nabla_\theta \log \tilde{p}_\theta\right] \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s} + \int \left(\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta} - \mathbb{E}\left[\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right]\right)(\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}\right]. \tag{B.5}$$

The subtracted term, $\mathbb{E}\left[\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right] \int (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}$, is zero because it is a constant times the expectation of the score function. As such, it keeps the weight updates unbiased, but can still significantly reduce the variance.

The original NVI method employs a dynamic baseline estimated with a neural network as a function of inputs $\mathbf{s}$. It is likely that this more flexible control variate can further reduce the variance of parameter estimates beyond the baseline that we explore here. However, this baseline was trained with backpropagation, and as such, would not provide a biologically-plausible comparison. We can approximate Eq. B.5 by summing over samples from $\tilde{q}_\theta$, and updating our weights at every time point:

$$\begin{aligned}\Delta\theta &\propto \left[\nabla_\theta \log \tilde{p}_t(\mathbf{r}_t, \mathbf{s}_t; \theta)\right] + \left[\log \frac{\tilde{p}_t}{\tilde{q}_t} - \bar{\mathcal{L}}\right] \sum_{s=0}^{t} (\nabla_\theta \log \tilde{q}_t(\mathbf{r}_t, \mathbf{s}_t; \theta)) \\ &\propto \left[\nabla_\theta \log \tilde{p}_t(\mathbf{r}_t, \mathbf{s}_t; \theta)\right] + \left[\log \frac{\tilde{p}_t}{\tilde{q}_t} - \bar{\mathcal{L}}\right] g^\theta,\end{aligned} \tag{B.6}$$

where $\bar{\mathcal{L}}$ is approximated online according to a running average of the loss at each time step, and g^θ, called an 'eligibility trace' [Gerstner et al. 2018], is computed by a running integral. These quantities are both computed online as follows:

$$\bar{\mathcal{L}}_t = \gamma_{\mathcal{L}} \log \frac{\tilde{p}_t}{\tilde{q}_t} + (1 - \gamma_{\mathcal{L}})\bar{\mathcal{L}}_{t-1} \tag{B.7}$$

$$g^\theta_t = \nabla_\theta \log \tilde{q}_t(\mathbf{r}_t, \mathbf{s}_t; \theta) + \gamma_g g^\theta_{t-1}, \tag{B.8}$$

where $\gamma_{\mathcal{L}} \ll 1$, so that $\bar{\mathcal{L}}_t$ is a weighted average of past losses. If we want an unbiased estimate of the gradient, then we would take $\gamma_g = 1$, so that $g_t^\theta = \sum_{s=0}^{t}(\nabla_\theta \log \tilde{q}_t(\mathbf{r}_t, \mathbf{s}_t; \theta))$. However, the variance of this eligibility trace grows without bound as $T \to \infty$, which makes online learning using this algorithm nearly impossible without approximation. For this reason, we take γ_e as a constant less than, but close to 1 when we compare NVI* to IL performance, which introduces a small bias, with the benefit of allowing for online learning. This is a technique commonly employed in the three-factor plasticity literature [Frémaux and Gerstner 2016; Miconi 2017], and can be thought of as an analog to temporal windowing in backpropagation through time [Werbos 1990]. For our numerical gradient comparisons (Fig. 2), however, we used a short number of time steps, but took $\gamma_g = 1$ to remove all bias.

This method of differentiation is particularly important to compare to IL, because it can be thought of as a three-factor synaptic plasticity rule, where for a neural network, the parameter update becomes a global 'loss' signal $\log \frac{\tilde{p}_t}{\tilde{q}_t} - \bar{\mathcal{L}}$ combined with synaptically local terms g^θ and $\nabla_\theta \log \tilde{p}_t(\mathbf{r}_t, \mathbf{s}_t; \theta)$, the second of which comprises the entirety of the IL update. Typically for reinforcement learning, the global 'reward' signal is justified by referencing neuromodulatory signals that project broadly to synapses throughout the cortex and carry information about reward [Fiete et al. 2007; Frémaux and Gerstner 2016; Hoerzer et al. 2014; Bredenberg et al. 2020]. However, the origins of the global 'loss' in our *unsupervised* case are unclear. Furthermore, as we show in Appendix B.3, the term $\left[\log \frac{\tilde{p}_t}{\tilde{q}_t} - \bar{\mathcal{L}}\right] g^\theta$ is high variance, and requires orders of magnitude more samples (or lower learning rates) in order to get a useful gradient estimate. A technical way of viewing our contribution in this paper is that we have shown that the $\left[\log \frac{\tilde{p}_t}{\tilde{q}_t} - \bar{\mathcal{L}}\right] g^\theta$ term is largely redundant and unnecessary for effective learning on our unsupervised objective, and that discarding it produces substantial performance increases while allowing the parameter update to remain a completely local synaptic plasticity rule for neural networks.

B.2.2 Backpropagation

Backpropagation (BP) cannot be performed for stochastic variables $\mathbf{r}_t$, because under an expectation, these are integration variables with no explicit dependency on any parameters. For this reason, when computing a derivative of our loss using NVI*, we differentiate the *probability distribution*, which depends on network parameters. However, as we will show below, this straightforward method can result in high variance parameter estimates. The classical alternative to NVI* is to perform the 'reparameterization trick,' in which a change of variables allows the use of stochastic gradient descent with BP. This trick is largely responsible for the success of the variational autoencoder [Kingma and Welling 2013; Rezende et al. 2014], though it is well known that BP does not produce synaptically local parameter updates. Here, we use BP as an upper bound for comparison, with the understanding that local learning algorithms are unlikely to be able to completely match its performance. Below, we review its calculation, starting with changing our variable of integration.

It is worth noting that this 'reparameterization' will work only for additive Gaussian noise. As such, applying BP to our network will only be possible for a restricted set of noise models, and can fail in particular for Poisson-spiking network models, where IL, NVI*, and WS will not. For each time point, we define $\boldsymbol{\eta}_t = \mathbf{r}_t - \bar{\mathbf{r}}_t^q(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})$, where $\bar{\mathbf{r}}_t^q(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})$ is the mean firing rate conditioned on noise, stimulus, and λ values from previous time steps (given by $\tilde{q}$). Similarly, we define $\boldsymbol{\xi}_t = \mathbf{s}_t - \bar{\mathbf{s}}_t^q(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})$. This defines $\boldsymbol{\eta}_t$ and $\boldsymbol{\xi}_t$ as the noise added on top of every firing rate and stimulus at time t. Instead of integrating over the rates and stimuli, we integrate over these fluctuations, replacing each instance of $\mathbf{r}_t$ with $\bar{\mathbf{r}}_t^q(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1}) + \boldsymbol{\eta}_t$ and $\mathbf{s}_t$ with $\bar{\mathbf{s}}_t^q(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1}) + \boldsymbol{\xi}_t$. We will refer to the mean parameters of $\tilde{p}_\theta$ where these substitutions have been made as $\bar{\mathbf{r}}_t^p(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})$ and $\bar{\mathbf{s}}_t^q(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})$. Our new random variables have the probability distributions: $p(\boldsymbol{\eta}_t) = \mathcal{N}(0, \lambda_t \sigma_r^{inf} + (1-\lambda_t)\sigma_r^{\text{gen}})$ and $p(\boldsymbol{\xi}_t) = \mathcal{N}(0, \lambda_t \sigma_s^{\text{inf}} + (1-\lambda_t)\sigma_s^{\text{gen}})$.

Performing our change of variables gives:

$$
\begin{aligned}
-\nabla_\theta \mathcal{L} =& -\nabla_\theta \int \left[\log \tilde{q}_\theta - \log \tilde{p}_\theta\right] \tilde{q}_\theta \, d\mathbf{r} d\mathbf{s} \\
=& -\nabla_\theta \int \left[\log \prod_t \frac{1}{Z} \exp\left(\frac{-\boldsymbol{\eta}_t^2}{2(\lambda_t \sigma_s^{\text{inf}} + (1-\lambda_t)\sigma_s^{\text{gen}})^2}\right)\right] p(\boldsymbol{\eta}, \boldsymbol{\xi}) \, d\boldsymbol{\eta} d\boldsymbol{\xi} \\
& -\nabla_\theta \int \left[\log \prod_t \frac{1}{Z} \exp\left(\frac{-\boldsymbol{\xi}_t^2}{2(\lambda_t \sigma_s^{\text{inf}} + (1-\lambda_t)\sigma_s^{\text{gen}})^2}\right)\right] p(\eta, \xi) \, d\boldsymbol{\eta} d\boldsymbol{\xi} \\
& +\nabla_\theta \int \left[\log \prod_t \frac{1}{Z} \exp\left(\frac{-(\bar{\mathbf{r}}_t^q + \boldsymbol{\eta}_t - \bar{\mathbf{r}}_t^p)^2}{2((1-\lambda_t)\sigma_r^{\text{inf}} + \lambda_t \sigma_r^{\text{gen}})^2}\right)\right] p(\boldsymbol{\eta}, \boldsymbol{\xi}) \, d\boldsymbol{\eta} d\boldsymbol{\xi} \\
& +\nabla_\theta \int \left[\log \prod_t \frac{1}{Z} \exp\left(\frac{-(\bar{\mathbf{s}}_t^q + \boldsymbol{\xi}_t - \bar{\mathbf{s}}_t^p)^2}{2((1-\lambda_t)\sigma_s^{\text{inf}} + \lambda_t \sigma_s^{\text{gen}})^2}\right)\right] p(\boldsymbol{\eta}, \boldsymbol{\xi}) \, d\boldsymbol{\eta} d\boldsymbol{\xi} \\
=& \mathbb{E}_{\boldsymbol{\eta}, \boldsymbol{\xi}} \left[\nabla_\theta \sum_t -\frac{\left(\bar{\mathbf{r}}_t^q(\theta, \boldsymbol{\eta}, \boldsymbol{\xi}) + \boldsymbol{\eta}_t - \bar{\mathbf{r}}_t^p(\theta, \boldsymbol{\eta}, \boldsymbol{\xi})\right)^2}{2((1-\lambda_t)\sigma_r^{\text{inf}} + \lambda_t \sigma_r^{\text{gen}})^2} - \frac{\left(\bar{\mathbf{s}}_t^q(\theta, \boldsymbol{\eta}, \boldsymbol{\xi}) + \boldsymbol{\xi}_t - \bar{\mathbf{s}}_t^p(\theta, \boldsymbol{\eta}, \boldsymbol{\xi})\right)^2}{2((1-\lambda_t)\sigma_s^{\text{inf}} + \lambda_t \sigma_s^{\text{gen}})^2}\right], \quad \text{(B.9)}
\end{aligned}
$$

where the last equality follows from the fact that η_t and ξ_t have no dependence on the network parameters. Now, the parameter dependence is contained in $\bar{\mathbf{r}}_t^q$, $\bar{\mathbf{r}}_t^p$, $\bar{\mathbf{s}}_t^q$, and $\bar{\mathbf{s}}_t^p$, all of which depend on the mean firing rates at *each previous time step*: using BP to compute the gradients of these mean parameters leads to nonlocal updates, which is the key reason BP is a biologically-implausible algorithm [Lillicrap et al. 2020]. For our simulations, we set $\lambda_t = 1 \; \forall t$, so that our parameter updates were equivalent to minimizing the negative ELBO, and gradients were computed using Pytorch [Paszke et al. 2019]. In subsequent sections, we will show that weight updates computed using samples from this expectation will generally have much lower variance than NVI*.

B.2.3 Wake-Sleep

As already mentioned, WS can be viewed as a special case of IL. To show this, we can take $\lambda_t = \lambda_0 \; \forall t$, with $p(\lambda_0 = 0) = p(\lambda_0 = 1) = 0.5$ (for IL, λ_t alternates with phase duration $K = 2$). For

this choice of λ, we follow our IL derivation (Eq. 5), and get:

$$-\nabla_\theta \mathcal{L} \approx 2\mathbb{E}_{\lambda_0,\mathbf{z}}\left[\int \left[\sum_t (1-\lambda_t)\nabla_\theta \log q_t + (\lambda_t)\nabla_\theta \log p_{mt}\right] \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}\right]$$
$$= \mathbb{E}_{\mathbf{z}}\left[\int \left[\sum_t \nabla_\theta \log q_t\right] p_m(\mathbf{r},\mathbf{s}) \; d\mathbf{r}d\mathbf{s} + \int \left[\sum_t \nabla_\theta \log p_{mt}\right] q(\mathbf{r}|\mathbf{s})p(\mathbf{s}|\mathbf{z}) \; d\mathbf{r}d\mathbf{s}\right]. \quad \text{(B.10)}$$

Since WS is a special case of IL, the bias properties of its individual samples are identical. However, typically WS weight updates are computed coordinate-wise, updating parameters for p_m and q separately, whose updates are computed after averaging over many samples. This can lead to behavior that approximates the EM algorithm under restrictive conditions, a fact that is used in the proofs of convergence of the WS algorithm for simple models [Amari and Nakahara 1999]. Because our algorithm does not perform coordinate descent, it is best viewed as an approximation to gradient descent with a well-behaved bias, rather than an approximation of the EM algorithm.

The WS parameter updates can also be interpreted as synaptic plasticity at apical and basal dendrites of pyramidal neurons, as with IL. The key difference is that WS requires lengthy phases where $\lambda_t = 1 \; \forall t$ (Wake) and where $\lambda_t = 0 \; \forall t$ (Sleep). The requirement that the network remain in a generative state while training the inference parameters θ_q would require a biological organism to explicitly hallucinate while training its parameters. Though such generative states may be possible in some restricted form, and WS could perfectly coexist with IL in a biological organism, we believe the more general perspective afforded by IL is much more likely to correspond to biology than the phase distinctions required by WS. The benefits to perceptual continuity given by IL over WS come from its ability to leverage temporal predictability in both network states and stimuli by only staying in a generative state for a brief period of time. However, for static images and neural architectures, IL and WS are much more similar, effectively amounting to different schedules for updating generative and inference parameters in alternating sequence.

B.3 Estimator variance

In Appendix B.1, we explored the bias introduced by the approximations used in the derivation of IL. Here, we consider the variance of sample weight updates, and compare to the variability of samples obtained from more standard methods, in particular BP and NVI*, whose sampling-based estimates have can have very different variances [Rezende et al. 2014].

To keep the analysis tractable, we will study a simple example: maximizing our modified KL divergence between two time series composed of temporally-uncorrelated univariate normal distributions with identical variance and different means: $p(r_t) \sim \mathcal{N}(\mu_p, \sigma^2)$, $q(r_t) \sim \mathcal{N}(\mu_q, \sigma^2)$. We define λ_t such that $p(\lambda_t = 0) = p(\lambda_t = 1) = 0.5 \; \forall t$. This produces the two hybrid distributions:

$$\tilde{p}(r|\lambda_t) = \prod_{t=0}^{T} p(r_t)^{\lambda_t} q(r_t)^{(1-\lambda_t)} \tag{B.11}$$

$$\tilde{q}(r|\lambda_t) = \prod_{t=0}^{T} p(r_t)^{(1-\lambda_t)} q(r_t)^{\lambda_t}. \tag{B.12}$$

Using these hybrid distributions, we can write our objective function as:

$$\mathcal{L} = \mathbb{E}_{\lambda_t}\left[KL(\tilde{q}||\tilde{p})\right] = \int \left[\int (\log \tilde{q}(r|\lambda_t) - \log \tilde{p}(r|\lambda_t))\tilde{q}(r|\lambda_t)dr\right] p(\lambda_t)d\lambda_t. \tag{B.13}$$

We will show that our three methods: NVI*, BP, and IL (which here will coincide exactly with WS), all produce unbiased stochastic gradient estimates, with very different variance properties.

It is worth explicitly outlining why variance is such an important quantity for stochastic gradient estimates. Suppose we obtain N independent samples of a weight update $\Delta\mu_q$, and want to compute the MSE of our estimated weight update to the *true* gradient, in expectation over

samples:

$$MSE(\Delta\mu_q) = \mathbb{E}_{\Delta\mu_q^{(n)}}\left[\left(-\frac{d}{d\mu_q}\mathcal{L} - \frac{1}{N}\sum_{n=0}^{N}\Delta\mu_q^{(n)}\right)^2\right]$$
$$= \left(-\frac{d}{d\mu_q}\mathcal{L} - \mathbb{E}_{\Delta\mu_q^{(n)}}\left[\frac{1}{N}\sum_{n=0}^{N}\Delta\mu_q^{(n)}\right]\right)^2 + Var\left[\frac{1}{N}\sum_{n=0}^{N}\Delta\mu_q^{(n)}\right]. \tag{B.14}$$

Here, the equality follows from bias-variance decomposition of the mean-squared error. In our toy example (but not in general) the biases for IL, BP, and NVI* will all be 0. This gives:

$$MSE(\Delta\mu_q) = Var\left[\frac{1}{N}\sum_{n=0}^{N}\Delta\mu_q^{(n)}\right] = \frac{Var\left[\Delta\mu_q^{(n)}\right]}{N}. \tag{B.15}$$

Suppose we want the mean-squared error to be less than some value $\epsilon \ll 1$. How many samples (N) do we need to take to bring ourselves below this error on average? We have:

$$\frac{Var\left[\Delta\mu_q^{(n)}\right]}{N} < \epsilon \quad \Rightarrow \quad \frac{Var\left[\Delta\mu_q^{(n)}\right]}{\epsilon} < N. \tag{B.16}$$

This means that increases in the variance of a weight estimate require proportionate increases in the number of samples required to reduce the error of the estimate. In practice, this requires high variance methods to process more data and to have lower learning rates, in some cases by several orders of magnitude. Even if a stochastic weight update is 'local' in a biologically-plausible sense, it may still require so much data for learning to occur as to be completely impractical.

B.3.1 Comparing Variances

Analytic variance calculations are only possible for the simplest of examples, but the intuitions they provide are nevertheless valuable. In the sections that follow, we will show that samples from all three methods have exactly the same expectation (the 'signal'), but only IL and BP agree on

their variance, while NVI* typically has much higher variance. For univariate normal distributions with identical variance, the loss $\mathcal{L} = \mathbb{E}_\lambda\left[KL(\tilde{q}||\tilde{p})\right] = KL[q||p] = T(\mu_p - \mu_q)^2/2\sigma^2$. Writing the variances in terms of the loss, we have:

$$Var_{\text{IL}} = Var_{\text{BP}} = \frac{T}{\sigma^2} \tag{B.17}$$

$$Var_{\text{NVI}} = \frac{T}{2\sigma^2} + \frac{\mathcal{L}}{8\sigma^2}\left(3T + 5\right) \tag{B.18}$$

This shows that for the most part, IL and BP hugely outperform NVI*. However, it is possible for NVI* to outperform these methods in the limit as $\mathcal{L} \to 0$ (a regime only achieved *after* successful optimization). Here, as with our numerical results, we have incorporated two methods that partially ameliorate the high variance of the NVI* estimate, which for reasonably low-dimensional tasks, can still allow it to perform comparably to BP; however, NVI* is unlikely to scale well to high dimensions, even with these additions. The purpose for our analysis is to show that these high variance difficulties do not apply to IL, whose scaling properties are much closer to BP.

B.3.2 Backpropagation

Expectation We will focus only on $\frac{d}{d\mu_q}$ for simplicity. Because the entropy of $\tilde{q}$ is constant with respect to the mean μ_q, we don't have to worry about the second term in the objective function. Instead, we focus on:

$$\begin{aligned}
-\frac{d}{d\mu_q}\mathcal{L} &= \frac{d}{d\mu_q}\int\left[\int(\log\tilde{p}(r|\lambda))\tilde{q}(r|\lambda)dr\right]p(\lambda)d\lambda \\
&= \frac{d}{d\mu_q}\sum_t\left[\int\frac{1}{2}(\log p(r_t))q(r_t)dr_t + \int\frac{1}{2}(\log q(r_t))p(r_t)dr_t\right] \\
&= -\frac{d}{d\mu_q}\sum_t\left[\int\frac{1}{4\sigma^2}((r_t-\mu_p)^2)q(r_t)dr_t + \int\frac{1}{4\sigma^2}((r_t-\mu_q)^2)p(r_t)dr_t\right].
\end{aligned} \tag{B.19}$$

At this point, we employ the 'reparameterization trick,' which reduces the variance of the weight update relative to NVI*. For the first integral we use the change of variables $r_t = \mu_q + \eta_t$, and for the second integral we use the change of variables $r_t = \mu_p + \eta_t$, where $\eta_t \sim \mathcal{N}(0, \sigma^2)$. This gives:

$$\begin{aligned}
-\frac{d}{d\mu_q}\mathcal{L} &= -\frac{d}{d\mu_q}\sum_{t=0}^{T}\left[\int \frac{1}{4\sigma^2}((\mu_q + \eta_t - \mu_p)^2)p(\eta_t)d\eta_t + \int \frac{1}{4\sigma^2}((\mu_p + \eta_t - \mu_q)^2)p(\eta_t)d\eta_t\right] \\
&= -\frac{d}{d\mu_q}\sum_{t=0}^{T}\int \frac{1}{2\sigma^2}((\mu_q + \eta_t - \mu_p)^2)p(\eta_t)d\eta_t \\
&= \sum_{t=0}^{T}\int \frac{1}{\sigma^2}(\mu_p + \eta_t - \mu_q)p(\eta_t)d\eta_t. \qquad \text{(B.20)}
\end{aligned}$$

Computing this expectation analytically, we have: $-\frac{d}{d\mu_q}\mathcal{L} = \frac{T}{\sigma^2}(\mu_p - \mu_q)$, which is unbiased, because we have not employed any approximations. If we were to compute this expectation using samples from $p(\eta_t)$, each individual parameter update would be given by $\Delta\mu_q \propto \sum_{t=0}^{T}\frac{1}{\sigma^2}(\mu_p + \eta_t - \mu_q)$ for a given sample from η. Given our expected weight update, we now ask for the variance.

Variance The variance of a sample, $\sum_{t=0}^{T}\frac{1}{\sigma^2}(\mu_p + \eta_t - \mu_q)$, is given by:

$$\begin{aligned}
Var(\Delta\mu_q) &= \int \left(\frac{1}{\sigma^2}(\sum_{t=0}^{T}(\mu_p + \eta_t - \mu_q - (\mu_p - \mu_q)))\right)^2 p(\eta_t)d\eta_t \\
&= \int \sum_{t=0}^{T}\frac{\eta_t^2}{\sigma^4}p(\eta_t)d\eta_t \\
&= \frac{T}{\sigma^2}. \qquad \text{(B.21)}
\end{aligned}$$

B.3.3 Impression learning

Expectation We can use our previous derivation of the IL weight update to write:

$$
\begin{aligned}
-\frac{d}{d\mu_q}\mathcal{L} &\approx 2\sum_{t=0}^{T}\left[\int\left[(1-\lambda_t)\frac{d}{d\mu_q}\log q(r_t) + (\lambda_t)\frac{d}{d\mu_q}\log p\right]\tilde{q}(r_t|\lambda_t)dr_t\right]p(\lambda_t)d\lambda_t \\
&= 2\sum_{t=0}^{T}\left[\int(1-\lambda_t)\frac{d}{d\mu_q}\log q(r_t)]\tilde{q}(r_t|\lambda)dr_t\right]p(\lambda_t)d\lambda_t \\
&= \sum_{t=0}^{T}\int\frac{d}{d\mu_q}\log q(r_t)p(r_t)dr_t
\end{aligned}
\tag{B.22}
$$

where this last equality follows from the fact that $\tilde{q}(r_t|\lambda) = p(r_t)$ whenever $1-\lambda_t = 1$. Continuing our derivation by substituting in $\log q(r_t)$ and discarding constants, we have:

$$
\begin{aligned}
-\frac{d}{d\mu_q}\mathcal{L} &\approx \sum_{t=0}^{T}\int -\frac{d}{d\mu_q}\frac{1}{2\sigma^2}(r_t-\mu_q)^2 p(r_t)dr_t \\
&= \sum_{t=0}^{T}\int\frac{1}{\sigma^2}(r_t-\mu_q)p(r_t)dr_t.
\end{aligned}
\tag{B.23}
$$

Computing this expectation analytically gives: $-\frac{d}{d\mu_q}\mathcal{L} \approx \frac{T}{\sigma^2}(\mu_p - \mu_q)$. Interestingly, in this case, the expected weight update coincides directly with the update given by BP, meaning that for this contrived example, IL is unbiased. This is clearly not the case in general, but works because our simplified network has no temporal interdependencies between variables and lacks hierarchical structure. In fact, the IL update also directly corresponds to the WS update in this case for the same reason. As with BP, we can ask about the variance of an individual sample of an update given by IL, assuming $\Delta\mu_q \propto \sum_{t=0}^{T}\frac{1}{\sigma^2}(r_t - \mu_q)$.

Variance The variance of a sample, $\sum_{t=0}^{T} \frac{1}{\sigma^2}(r_t - \mu_q)$, is given by:

$$\begin{aligned} Var(\Delta\mu_q) &= \int \left(\frac{1}{\sigma^2} (\sum_{t=0}^{T} r_t - \mu_q - (\mu_p - \mu_q)) \right)^2 p(r_t) dr_t \\ &= \int \frac{1}{\sigma^4} (\sum_{t=0}^{T} (r_t - \mu_p))^2 p(r_t) dr_t \\ &= \int \frac{1}{\sigma^4} \sum_{t=0}^{T} \sum_{t'=0}^{T} (r_t - \mu_p)(r_{t'} - \mu_p) p(r_t) dr_t \\ &= \int \frac{1}{\sigma^4} \sum_{t=0}^{T} (r_t - \mu_p)^2 p(r_t) dr_t \\ &= \frac{T}{\sigma^2}, \end{aligned} \tag{B.24}$$

where here we have exploited the fact that $\mathbb{E}[(r_t - \mu_p)(r_{t'} - \mu_p)] = 0 \;\; \forall t \neq t'$. This shows that for this simple example, there is a perfect correspondence between both the expectation and the variance of IL compared to BP.

B.3.4 Neural Variational Inference

Expectation The difference between NVI* and BP is that we do not use a change of variables. Given our previous derivation of the NVI* update (Eq. B.4), we have:

$$\begin{aligned} -\frac{d}{d\mu_q}\mathcal{L} &= \int \left[\int \frac{d}{d\mu_q} \log \tilde{p}(r|\lambda_t) \tilde{q}(r|\lambda) + (\log \tilde{p} - \log \tilde{q}) \, (\frac{d}{d\mu_q} \log \tilde{q}(r|\lambda)) \tilde{q}(r|\lambda) dr \right] p(\lambda_t) d\lambda_t \\ &= \int \left[\int \left(\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2} (r_t - \mu_q) + (\log \tilde{p} - \log \tilde{q}) \sum_{t=0}^{T} \frac{\lambda_t}{\sigma^2} (r_t - \mu_q) \right) \tilde{q}(r|\lambda) dr \right] p(\lambda_t) d\lambda_t, \end{aligned}$$

where the second equality follows from substituting in $\frac{d}{d\mu_q} \log \tilde{p}(r|\lambda_t)$ and $\frac{d}{d\mu_q} \log \tilde{q}(r|\lambda)$. Noting that $\log \tilde{p} - \log \tilde{q} = \log p - \log q$ when $\lambda_t = 1$, we continue:

$$
\begin{aligned}
-\frac{d}{d\mu_q}\mathcal{L} &= \int \left[\int \left(\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) + (\log p - \log q) \sum_{t=0}^{T} \frac{\lambda_t}{\sigma^2}(r_t - \mu_q) \right) \tilde{q}(r|\lambda) dr \right] p(\lambda_t) d\lambda_t \\
&= \mathbb{E}_{r,\lambda} \left[\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) - \left(\sum_{t=0}^{T} (r_t - \mu_p)^2 - (r_t - \mu_q)^2 \right) \sum_{t=0}^{T} \frac{\lambda_t}{2\sigma^4}(r_t - \mu_q) \right] \\
&= \mathbb{E}_{r,\lambda} \left[\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) - \left(\sum_{t=0}^{T} 2r_t(\mu_q - \mu_p) + \mu_p^2 - \mu_q^2 \right) \sum_{t=0}^{T} \frac{\lambda_t}{2\sigma^4}(r_t - \mu_q) \right]. \qquad \text{(B.25)}
\end{aligned}
$$

At this point, we'll allow ourselves to exploit the structure of our problem in two ways commonly employed in NVI*. First, we observe that the loss at a particular time step, $2r_t(\mu_q - \mu_p) + \mu_p^2 - \mu_q^2$ is independent of $r_{t'} - \mu_q$ for $t' > t$, i.e. fluctuations in variables at future time steps do not influence the current loss. Incorporating this fact modifies our update to give:

$$
-\frac{d}{d\mu_q}\mathcal{L} = \mathbb{E}_{r,\lambda} \left[\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) - \sum_{t=0}^{T} \sum_{t' \leq t} \frac{\lambda_t}{2\sigma^4} \left(2r_t(\mu_q - \mu_p) + \mu_p^2 - \mu_q^2 \right) (r'_t - \mu_q) \right]. \qquad \text{(B.26)}
$$

Next, we notice that $\mathbb{E}\left[\sum_{t' \leq t} \frac{\lambda_t}{2\sigma^4}(r'_t - \mu_q) \right] = 0$, so we can subtract from our update $a \times \sum_{t' \leq t} \frac{\lambda_t}{2\sigma^4}(r'_t - \mu_q)$ for some constant a, without modifying the expectation of our loss. Choosing a constant a that will reduce the variance of the parameter update is a common technique used in NVI*, called using a 'control variate' [Ranganath et al. 2014; Mnih and Gregor 2014]. We notice that the average loss contributes nothing to the expectation, so we take $a = 2\mu_q(\mu_q - \mu_p) + \mu_p^2 - \mu_q^2$, giving the improved-variance update:

$$
-\frac{d}{d\mu_q}\mathcal{L} = \mathbb{E}_{r,\lambda} \left[\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) - \sum_{t=0}^{T} \sum_{t' \leq t} \frac{\lambda_t}{\sigma^4}(r_t - \mu_q)(\mu_q - \mu_p)(r'_t - \mu_q) \right]. \qquad \text{(B.27)}
$$

Individual samples from this method of differentiation are more complicated (and higher variance) than IL or BP. An individual sample would give: $\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) - \sum_{t=0}^{T} \sum_{t' \leq t} \frac{\lambda_t}{\sigma^4}(r_t - \mu_q)(\mu_q - \mu_p)(r'_t - \mu_q)$. We'll first compute the expectation of this expression (to verify that it is equivalent

to BP and IL), and then we'll compute its variance. Continuing our calculation, we get:

$$
\begin{aligned}
-\frac{d}{d\mu_q}\mathcal{L} &= \mathbb{E}_{r,\lambda}\left[\sum_{t=0}^{T}\frac{1-\lambda_t}{\sigma^2}(r_t-\mu_q) - \sum_{t=0}^{T}\sum_{t'\le t}\frac{\lambda_t}{\sigma^4}(r_t-\mu_q)(\mu_q-\mu_p)(r'_t-\mu_q)\right] \\
&= \int\sum_{t=0}^{T}\frac{(1-\lambda_t)}{\sigma^2}(r_t-\mu_q)p(r)dr + \int\frac{1}{2\sigma^4}\sum_{t=0}^{T}\sum_{t'\le t}(r_t-\mu_q)(\mu_p-\mu_q)(r'_t-\mu_q)q(r)dr \\
&= \frac{T}{2\sigma^2}(\mu_p-\mu_q) + \int\frac{(\mu_p-\mu_q)}{2\sigma^4}\sum_{t=0}^{T}\sum_{t'\le t}\left(r_t-\mu_q\right)\left(r'_t-\mu_q\right)q(r)dr \\
&= \frac{T}{2\sigma^2}(\mu_p-\mu_q) + \int\frac{(\mu_p-\mu_q)}{2\sigma^4}\sum_{t=0}^{T}\sum_{t'\le t}\left(\eta_t\right)\left(\eta_{t'}\right)p(\eta)d\eta \\
&= \frac{T}{2\sigma^2}(\mu_p-\mu_q) + \int\frac{(\mu_p-\mu_q)}{2\sigma^4}\sum_{t=0}^{T}\eta_t^2 p(\eta)d\eta \\
&= \frac{T}{\sigma^2}(\mu_p-\mu_q),
\end{aligned}
\tag{B.28}
$$

where the fourth equality comes from reparameterizing with the transformation $\eta_t = r_t - \mu_q$ and the fifth equality stems from the fact that $\mathbb{E}\left[\eta_t\right] = 0$ and $\mathbb{E}\left[\eta_t\eta_{t'}\right] = 0$. This verifies that whether we sample over r using the black-box differentiation method, or over η using the reparameterization trick, or use IL, we will arrive at the same weight update in *expectation*. The variance of sample estimates thus distinguishes IL from NVI* (on this example at least).

Variance Because of the NVI* sample estimate's increased complexity, the variance calculation is also much more involved:

$$
\begin{aligned}
Var(\Delta\mu_q) =& \mathbb{E}_{r,\lambda}\left[\left(\Delta\mu_q - \frac{T}{\sigma^2}(\mu_p - \mu_q)\right)^2\right] \\
=& \mathbb{E}_{r,\lambda}\left[\left(\sum_{t=0}^{T}\frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) - \sum_{t=0}^{T}\sum_{t'\le t}\frac{\lambda_t}{2\sigma^4}(r_t - \mu_q)(\mu_q - \mu_p)(r'_t - \mu_q) - \frac{T}{\sigma^2}(\mu_p - \mu_q)\right)^2\right] \\
=& \frac{1}{2}\int \frac{1}{\sigma^4}\sum_{t=0}^{T}(r_t - \mu_p)^2 p(r)dr \\
&+ \frac{1}{2}\int\left(\frac{1}{2\sigma^4}\sum_{t=0}^{T}\sum_{t'\le t}(r_t - \mu_q)(\mu_p - \mu_q)(r'_t - \mu_q) - \frac{T}{\sigma^2}(\mu_p - \mu_q)\right)^2 q(r)dr, \qquad \text{(B.29)}
\end{aligned}
$$

where in this last step we have taken an expectation over λ, observing that the first term is only nonzero if $\lambda_t = 0$, and the second term is only nonzero if $\lambda_t = 1$. Now we apply the reparameterization, taking $r_t = \eta_t + \mu_p$ in the first integral, and $r_t = \eta_t + \mu_q$ in the second integral, giving:

$$
\begin{aligned}
Var(\Delta\mu_q) =& \frac{T}{2\sigma^2} + \frac{1}{2}\int \left(\frac{1}{2\sigma^4} \sum_{t=0}^{T} \sum_{t' \leq t} \left(\eta_t(\mu_p - \mu_q)\right)(\eta_{t'}) - \frac{T}{\sigma^2}(\mu_p - \mu_q) \right)^2 p(\eta) d\eta \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{2\sigma^4} \int \left(\frac{1}{2\sigma^2} \sum_{t=0}^{T} \sum_{t' \leq t} \eta_t \eta_{t'} - T \right)^2 p(\eta) d\eta \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{2\sigma^4} \mathbb{E}_{\eta_t} \left[\left(\frac{1}{2\sigma^2} \sum_{t=0}^{T} \sum_{t' \leq t} \eta_t \eta_{t'} \right)^2 - \frac{T}{\sigma^2} \left(\sum_{t=0}^{T} \sum_{t' \leq t} \eta_t \eta_{t'} \right) + T^2 \right] \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{2\sigma^4} \mathbb{E}_{\eta_t} \left[\left(\frac{1}{2\sigma^2} \sum_{t=0}^{T} \sum_{t' \leq t} \eta_t \eta_{t'} \right)^2 - \frac{T}{\sigma^2} \left(\sum_{t=0}^{T} \eta_t^2 \right) + T^2 \right] \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{2\sigma^4} \mathbb{E}_{\eta_t} \left[\left(\frac{1}{2\sigma^2} \sum_{t=0}^{T} \sum_{t' \leq t} \eta_t \eta_{t'} \right)^2 \right] \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{8\sigma^8} \mathbb{E}_{\eta_t} \left[\sum_{t=0}^{T} \sum_{t'=0}^{T} \sum_{t'' \leq t} \sum_{t''' \leq t'} \eta_t \eta_{t'} \eta_{t''} \eta_{t'''} \right] \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{8\sigma^8} \sum_{t=0}^{T} \sum_{t'=0}^{T} \sum_{t'' \leq t} \sum_{t''' \leq t'} \mathbb{E}_{\eta_t} \left[\eta_t \eta_{t'} \eta_{t''} \eta_{t'''} \right]. \qquad \text{(B.30)}
\end{aligned}
$$

Now, we notice that there are three mutually exclusive and exhaustive conditions under which this expectation is nonzero, using the the fact that only the even moments of the normal distribution are nonzero:

$$
\mathbb{E}_{\eta_t} \left[\eta_t \eta_{t'} \eta_{t''} \eta_{t'''} \right] = \begin{cases} \sigma^4 & \text{if } t = t' \text{ and } t'' = t''' \text{ and } t \neq t'' \\ \sigma^4 & \text{if } t = t'' \text{ and } t' = t''' \text{ and } t \neq t' \\ 3\sigma^4 & \text{if } t = t' = t'' = t''' \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(B.31)}
$$

These three different conditions result in three different sums:

$$
\begin{aligned}
Var(\Delta\mu_q) &= \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{8\sigma^8}\left(\sum_{t=1}^{T}\sum_{t'<t}\sigma^4 + \sum_{t=0}^{T}\sum_{t'\neq t}\sigma^4 + \sum_{t=0}^{T}3\sigma^4\right) \\
&= \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{8\sigma^8}\left(\sigma^4\sum_{t=1}^{T}(t) + T(T-1)\sigma^4 + 3T\sigma^4\right) \\
&= \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{8\sigma^8}\left(\frac{1}{2}T(T+1)\sigma^4 + T(T-1)\sigma^4 + 3T\sigma^4\right) \\
&= \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{16\sigma^4}\left(3T^2 + 5T\right) \\
&= \frac{T}{2\sigma^2} + \frac{\mathcal{L}}{8\sigma^2}\left(3T + 5\right), \qquad \text{(B.32)}
\end{aligned}
$$

where the third equality follows from the arithmetic series identity: $\sum_{t=1}^{T}(t) = \frac{1}{2}T(T+1)$.

B.4 Multilayer Network Architecture

Here we outline the architecture for the 2-layer network used for processing the Free Spoken Digits dataset [Jackson et al. 2018] in Figure 4.

B.4.1 Model structure

Our inference architecture simply adds an additional feedforward layer of neurons to the network:

$$\mathbf{s}_t^{\text{inf}} = \mathbf{z}_t + \sigma_s^{\text{inf}}\boldsymbol{\xi}_t \qquad \text{(B.33)}$$

$$\mathbf{r}_t^{\text{inf1}} = f(\mathbf{W}_1\mathbf{s}_t + \mathbf{a}) + \sigma_1^{\text{inf}}\boldsymbol{\eta}_t^1 \qquad \text{(B.34)}$$

$$\mathbf{r}_t^{\text{inf2}} = f(\mathbf{W}_2\mathbf{r}_t^{\text{inf1}}) + \sigma_2^{\text{inf}}\boldsymbol{\eta}_t^2, \qquad \text{(B.35)}$$

where $\mathbf{W}_l$ denotes the feedforward weights from layer $l-1$ to layer l, $\mathbf{a}$ is an additive bias parameter, $\boldsymbol{\eta}_t^1, \boldsymbol{\eta}_t^2, \boldsymbol{\xi}_t \sim \mathcal{N}(0,1)$ are independent white noise samples, σ_1^{inf}, σ_2^{inf}, and σ_s^{inf} denote the inference

standard deviations for their respective layers, and the nonlinearity $f(\cdot)$ is the tanh function. The multilayer generative model includes an additional feedforward decoder step:

$$\mathbf{r}_t^{\text{gen2}} = ((1-k_t)\mathbf{D}_2 + k_t\mathbf{I}))\, r_{t-1} + \sigma_2^{\text{gen}}\boldsymbol{\eta}_t^2 \tag{B.36}$$

$$\mathbf{r}_t^{\text{gen1}} = f(\mathbf{D}_1\mathbf{r}_t^{\text{gen2}} + \mathbf{b}) + \sigma_1^{\text{gen}}\boldsymbol{\eta}_t^1 \tag{B.37}$$

$$\mathbf{s}_t^{\text{gen}} = f(\mathbf{D}_s\mathbf{r}_t^{\text{gen1}}) + \sigma_s^{\text{gen}}\boldsymbol{\xi}_t, \tag{B.38}$$

where $\mathbf{D}_2$ is a diagonal transition matrix, $\mathbf{D}_1$ and $\mathbf{D}_s$ are prediction weights to their layers from higher layers, $\mathbf{b}$ is an additive bias parameter, $\mathbf{I}$ is the identity matrix, and σ_1^{gen}, σ_2^{gen}, and σ_s^{gen} denote the generative standard deviations for their layers. We define k_t as in the 1-layer network. Also in keeping with the basic model, during simulation, samples are determined by a combination of p_m and q, given by $\tilde{q}_\theta$:

$$\mathbf{r}_t^2 = \lambda_t\mathbf{r}_t^{\text{inf2}} + (1-\lambda_t)\mathbf{r}_t^{\text{gen2}} \tag{B.39}$$

$$\mathbf{r}_t^1 = \lambda_t\mathbf{r}_t^{\text{inf1}} + (1-\lambda_t)\mathbf{r}_t^{\text{gen1}} \tag{B.40}$$

$$\mathbf{s}_t = \lambda_t\mathbf{s}_t^{\text{inf}} + (1-\lambda_t)\mathbf{s}_t^{\text{gen}}. \tag{B.41}$$

B.4.2 Parameter updates

Adding additional layers to our model does not change the fact that the parameter updates can be interpreted as local synaptic plasticity rules at the basal (for q) or apical (for p) compartments of our neuron model. Plugging our probability models into the equation for the IL parameter update (Eq. 5), calculating derivatives, and updating our parameters stochastically at every time step as with our basic model gives:

$$\Delta \mathbf{W}_1^{(ij)} \propto \frac{1-\lambda_t}{(\sigma_1^{\text{inf}})^2}((\mathbf{r}_t^1)^{(i)} - f(\mathbf{W}_1\mathbf{s}_t + \mathbf{a})^{(i)})f'(\mathbf{W}_1\mathbf{s}_t + \mathbf{a})^{(i)}\mathbf{s}_t^{(j)} \tag{B.42}$$

$$\Delta \mathbf{a}^{(i)} \propto \frac{1-\lambda_t}{(\sigma_1^{\text{inf}})^2}((\mathbf{r}_t^1)^{(i)} - f(\mathbf{W}_1\mathbf{s}_t + \mathbf{a})^{(i)})f'(\mathbf{W}_1\mathbf{s}_t + \mathbf{a})^{(i)} \tag{B.43}$$

$$\Delta \mathbf{W}_2^{(ij)} \propto \frac{1-\lambda_t}{(\sigma_2^{\text{inf}})^2}((\mathbf{r}_t^2)^{(i)} - f(\mathbf{W}_2\mathbf{r}_t^1)^{(i)})f'(\mathbf{W}_2\mathbf{r}_t^1)^{(i)}(\mathbf{r}_t^1)^{(j)} \tag{B.44}$$

$$\Delta \mathbf{D}_2^{(ii)} \propto \frac{\lambda_t(1-k_t)}{(\sigma_2^{\text{gen}})^2}((\mathbf{r}_t^2)^{(i)} - (\mathbf{D}_2\mathbf{r}_{t-1}^2)^{(i)})(\mathbf{r}_{t-1}^2)^{(i)} \tag{B.45}$$

$$\Delta \mathbf{D}_1^{(ij)} \propto \frac{\lambda_t}{(\sigma_s^{\text{gen}})^2}((\mathbf{r}_t^1)^{(i)} - f(\mathbf{D}_1\mathbf{r}_t^2 + \mathbf{b})^{(i)})f'(\mathbf{D}_1\mathbf{r}_t^2 + \mathbf{b})^{(i)}(\mathbf{r}_t^2)^{(j)} \tag{B.46}$$

$$\Delta \mathbf{b}^{(i)} \propto \frac{\lambda_t}{(\sigma_s^{\text{gen}})^2}((\mathbf{r}_t^1)^{(i)} - f(\mathbf{D}_1\mathbf{r}_t^2 + \mathbf{b})^{(i)})f'(\mathbf{D}_1\mathbf{r}_t^2 + \mathbf{b})^{(i)} \tag{B.47}$$

$$\Delta \mathbf{D}_s^{(ij)} \propto \frac{\lambda_t}{(\sigma_s^{\text{gen}})^2}(\mathbf{s}_t^i - f(\mathbf{D}_s\mathbf{r}_t^1)^{(i)})f'(\mathbf{D}_s\mathbf{r}_t^1)^{(i)}(\mathbf{r}_t^1)^{(j)}. \tag{B.48}$$

Bibliography

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169.

Afraz, A., Boyden, E. S., and DiCarlo, J. J. (2015). Optogenetic and pharmacological suppression of spatial clusters of face neurons reveal their causal role in face gender discrimination. *Proceedings of the National Academy of Sciences*, 112(21):6730–6735.

Akrout, M., Wilson, C., Humphreys, P. C., Lillicrap, T., and Tweed, D. (2019). Using weight mirrors to improve feedback alignment. *arXiv preprint arXiv:1904.05391*.

Alemi, A., Machens, C., Deneve, S., and Slotine, J.-J. (2018). Learning nonlinear dynamics in efficient, balanced spiking networks using local plasticity rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Amari, S. I. S.-i. and Nakahara, H. (1999). Convergence of the wake-sleep algorithm. In *Advances in Neural Information Processing Systems 11: Proceedings of the 1998 Conference*, volume 11, page 239. MIT Press.

Atick, J. J. and Redlich, A. N. (1990). Towards a theory of early visual processing. *Neural computation*, 2(3):308–320.

Attneave, F. (1954). Some informational aspects of visual perception. *Psychological review*, 61(3):183.

Bakin, J. S. and Weinberger, N. M. (1996). Induction of a physiological memory in the cerebral cortex by stimulation of the nucleus basalis. *Proceedings of the National Academy of Sciences*, 93(20):11219–11224.

Bartunov, S., Santoro, A., Richards, B., Marris, L., Hinton, G. E., and Lillicrap, T. (2018). Assessing the scalability of biologically-motivated deep learning algorithms and architectures. *Advances in neural information processing systems*, 31.

Basso, M. A., Bickford, M. E., and Cang, J. (2021). Unraveling circuits of visual perception and cognition through the superior colliculus. *Neuron*.

Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., and Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711.

Bear, M. F. and Singer, W. (1986). Modulation of visual cortical plasticity by acetylcholine and noradrenaline. *Nature*, 320(6058):172–176.

Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., and Maass, W. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1):1–15.

Bengio, Y. (2014). How auto-encoders could provide credit assignment in deep networks via target propagation. *arXiv preprint arXiv:1407.7906*.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Benna, M. K. and Fusi, S. (2016). Computational principles of synaptic memory consolidation. *Nature neuroscience*, 19(12):1697–1706.

Bi, G.-q. and Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18(24):10464–10472.

Bill, J., Buesing, L., Habenschuss, S., Nessler, B., Maass, W., and Legenstein, R. (2015). Distributed Bayesian computation and self-organized learning in sheets of spiking neurons with local lateral inhibition. *PloS one*, 10(8):e0134356.

Bittner, K. C., Grienberger, C., Vaidya, S. P., Milstein, A. D., Macklin, J. J., Suh, J., Tonegawa, S., and Magee, J. C. (2015). Conjunctive input processing drives feature selectivity in hippocampal ca1 neurons. *Nature neuroscience*, 18(8):1133–1142.

Bittner, K. C., Milstein, A. D., Grienberger, C., Romani, S., and Magee, J. C. (2017). Behavioral time scale synaptic plasticity underlies ca1 place fields. *Science*, 357(6355):1033–1036.

Bliss, T. V. and Collingridge, G. L. (1993). A synaptic model of memory: long-term potentiation in the hippocampus. *Nature*, 361(6407):31–39.

Bouvier, G., Aljadeff, J., Clopath, C., Bimbard, C., Ranft, J., Blot, A., Nadal, J.-P., Brunel, N., Hakim, V., and Barbour, B. (2018). Cerebellar learning using perturbations. *Elife*, 7:e31599.

Bredenberg, C., Lyo, B. S. H., Simoncelli, E. P., and Savin, C. (2021). Impression learning: Online representation learning with synaptic plasticity. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

Bredenberg, C., Simoncelli, E., and Savin, C. (2020). Learning efficient task-dependent representations with synaptic plasticity. *Advances in Neural Information Processing Systems*, 33.

Brendel, W., Bourdoukan, R., Vertechi, P., Machens, C. K., and Denéve, S. (2017). Learning to represent signals spike by spike. *arXiv preprint arXiv:1703.03777*.

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

Britten, K. H., Shadlen, M. N., Newsome, W. T., and Movshon, J. A. (1992). The analysis of visual motion: a comparison of neuronal and psychophysical performance. *Journal of Neuroscience*, 12(12):4745–4765.

Brody, C. D., Hernández, A., Zainos, A., and Romo, R. (2003). Timing and neural encoding of somatosensory parametric working memory in macaque prefrontal cortex. *Cerebral cortex*, 13(11):1196–1207.

Brown, J., Behnam, R., Coddington, L., Tervo, D. G., Martin, K., Proskurin, M., Kuleshova, E., Park, J., Phillips, J., Bergs, A. C., et al. (2018). Expanding the optogenetics toolkit by topological inversion of rhodopsins. *Cell*, 175(4):1131–1140.

Cain, N., Barreiro, A. K., Shadlen, M., and Shea-Brown, E. (2013). Neural integrators for decision making: a favorable tradeoff between robustness and sensitivity. *J Neurophysiol*, 109(10):2542–2559.

Calabresi, P., Picconi, B., Tozzi, A., and Di Filippo, M. (2007). Dopamine-mediated regulation of corticostriatal synaptic plasticity. *Trends in neurosciences*, 30(5):211–219.

Cartwright, N. and McMullin, E. (1984). How the laws of physics lie.

Chandrasekaran, C., Peixoto, D., Newsome, W. T., and Shenoy, K. V. (2017). Laminar differences in decision-related neural activity in dorsal premotor cortex. *Nature communications*, 8(1):1–16.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Churchland, A. K., Kiani, R., Chaudhuri, R., Wang, X.-J., Pouget, A., and Shadlen, M. N. (2011). Variance as a signature of neural computations during decision making. *Neuron*, 69(4):818–831.

Churchland, A. K., Kiani, R., and Shadlen, M. N. (2008). Decision-making with multiple alternatives. *Nature neuroscience*, 11(6):693–702.

Clark, A. and Toribio, J. (1994). Doing without representing? *Synthese*, 101(3):401–431.

Clopath, C., Ziegler, L., Vasilaki, E., Büsing, L., and Gerstner, W. (2008). Tag-trigger-consolidation: a model of early and late long-term-potentiation and depression. *PLoS computational biology*, 4(12):e1000248.

Compte, A., Brunel, N., Goldman-Rakic, P. S., and Wang, X.-J. (2000). Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral cortex*, 10(9):910–923.

Craver, C. F. (2007). *Explaining the brain: Mechanisms and the mosaic unity of neuroscience*. Oxford University Press.

Dan, Y. and Poo, M.-m. (2004). Spike timing-dependent plasticity of neural circuits. *Neuron*, 44(1):23–30.

David, S. V., Fritz, J. B., and Shamma, S. A. (2012). Task reward structure shapes rapid receptive field plasticity in auditory cortex. *Proceedings of the National Academy of Sciences*, 109(6):2144–2149.

Dayan, P. and Hinton, G. E. (1996). Varieties of Helmholtz machine. *Neural Networks*, 9(8):1385–1403.

Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. (1995). The Helmholtz machine. *Neural computation*, 7(5):889–904.

De, A., El-Shamayleh, Y., and Horwitz, G. D. (2020). Fast and reversible neural inactivation in macaque cortex by optogenetic stimulation of gabaergic neurons. *Elife*, 9:e52658.

de Villers-Sidani, E., Simpson, K. L., Lu, Y., Lin, R. C., and Merzenich, M. M. (2008). Manipulating critical period closure across different sectors of the primary auditory cortex. *Nature neuroscience*, 11(8):957–965.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Deperrois, N., Petrovici, M. A., Senn, W., and Jordan, J. (2021). Memory semantization through perturbed and adversarial dreaming. *arXiv preprint arXiv:2109.04261*.

Deverett, B., Koay, S. A., Oostland, M., and Wang, S. S. (2018). Cerebellar involvement in an evidence-accumulation decision-making task. *Elife*, 7:e36781.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, L. and Gold, J. I. (2013). The basal ganglia's contributions to perceptual decision making. *Neuron*, 79(4):640–649.

Doll, B. B., Simon, D. A., and Daw, N. D. (2012). The ubiquity of model-based reinforcement learning. *Current opinion in neurobiology*, 22(6):1075–1081.

Drugowitsch, J., Moreno-Bote, R., Churchland, A. K., Shadlen, M. N., and Pouget, A. (2012). The cost of accumulating evidence in perceptual decision making. *J Neurosci*, 32(11):3612–3628.

Duncker, L., Bohner, G., Boussard, J., and Sahani, M. (2019). Learning interpretable continuous-time models of latent stochastic dynamical systems. In *International Conference on Machine Learning*, pages 1726–1734. PMLR.

Dunn, J. C. (2003). The elusive dissociation. *Cortex*, 39(1):177–179.

Eldridge, M. A., Lerchner, W., Saunders, R. C., Kaneko, H., Krausz, K. W., Gonzalez, F. J., Ji, B., Higuchi, M., Minamimoto, T., and Richmond, B. J. (2016). Chemogenetic disconnection of monkey orbitofrontal and rhinal cortex reversibly disrupts reward value. *Nat Neurosci*, 19(1):37–39.

Erlich, J. C., Brunton, B. W., Duan, C. A., Hanks, T. D., and Brody, C. D. (2015). Distinct effects of prefrontal and parietal cortex inactivations on an accumulation of evidence task in the rat. *Elife*, 4:e05457.

Ernoult, M., Grollier, J., Querlioz, D., Bengio, Y., and Scellier, B. (2020). Equilibrium propagation with continual weight updates. *arXiv preprint arXiv:2005.04168*.

Eschenko, O., Ramadan, W., Mölle, M., Born, J., and Sara, S. J. (2008). Sustained increase in hippocampal sharp-wave ripple activity during slow-wave sleep after learning. *Learning & memory*, 15(4):222–228.

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

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

Feulner, B. and Clopath, C. (2021). Neural manifold under plasticity in a goal driven learning behaviour. *PLoS computational biology*, 17(2):e1008621.

Fiete, I. R. (2004). *Learning and coding in biological neural networks*. Harvard University.

Fiete, I. R., Fee, M. S., and Seung, H. S. (2007). Model of birdsong learning based on gradient estimation by dynamic perturbation of neural conductances. *Journal of neurophysiology*, 98(4):2038–2057.

Fiser, J., Berkes, P., Orbán, G., and Lengyel, M. (2010a). Statistically optimal perception and learning: from behavior to neural representations. *Trends in cognitive sciences*, 14(3):119–130.

Fiser, J., Berkes, P., Orban, G., and Lengyel, M. (2010b). Statistically optimal perception and learning: from behavior to neural representations. *Trends in Cognitive Sciences*, 14(3):119–130.

Frémaux, N. and Gerstner, W. (2016). Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in neural circuits*, 9:85.

Frémaux, N., Sprekeler, H., and Gerstner, W. (2013). Reinforcement learning using a continuous time actor-critic framework with spiking neurons. *PLoS computational biology*, 9(4):e1003024.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Friedrich, J., Golkar, S., Farashahi, S., Genkin, A., Sengupta, A., and Chklovskii, D. (2021). Neural optimal feedback control with local learning rules. *Advances in Neural Information Processing Systems*, 34.

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138.

Fritz, J., Shamma, S., Elhilali, M., and Klein, D. (2003). Rapid task-related plasticity of spectrotemporal receptive fields in primary auditory cortex. *Nature neuroscience*, 6(11):1216–1223.

Froemke, R. C., Carcea, I., Barker, A. J., Yuan, K., Seybold, B. A., Martins, A. R. O., Zaika, N., Bernstein, H., Wachs, M., Levis, P. A., et al. (2013). Long-term modification of cortical synapses improves sensory perception. *Nature neuroscience*, 16(1):79.

Froemke, R. C., Letzkus, J. J., Kampa, B., Hang, G. B., and Stuart, G. (2010). Dendritic synapse location and neocortical spike-timing-dependent plasticity. *Frontiers in synaptic neuroscience*, 2:29.

Froemke, R. C., Merzenich, M. M., and Schreiner, C. E. (2007). A synaptic memory trace for cortical receptive field plasticity. *Nature*, 450(7168):425–429.

Froemke, R. C., Poo, M.-m., and Dan, Y. (2005). Spike-timing-dependent synaptic plasticity depends on dendritic location. *Nature*, 434(7030):221–225.

Fukushima, K. and Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer.

Fusi, S., Drew, P. J., and Abbott, L. F. (2005). Cascade models of synaptically stored memories. *Neuron*, 45(4):599–611.

Ganguli, D. and Simoncelli, E. P. (2014). Efficient sensory encoding and bayesian inference with heterogeneous neural populations. *Neural computation*, 26(10):2103–2134.

Ganguli, D. and Simoncelli, E. P. (2016). Neural and perceptual signatures of efficient sensory coding. *arXiv preprint arXiv:1603.00058*.

Ganguli, S., Huh, D., and Sompolinsky, H. (2008). Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences*, 105(48):18970–18975.

Ganguli, S. and Sompolinsky, H. (2012). Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis. *Annu Rev Neurosci*, 35:485–508.

Gao, Z., Van Beugen, B. J., and De Zeeuw, C. I. (2012). Distributed synergistic plasticity and cerebellar learning. *Nature Reviews Neuroscience*, 13(9):619–635.

Gerstner, W. and Kistler, W. M. (2002). *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press.

Gerstner, W., Lehmann, M., Liakoni, V., Corneil, D., and Brea, J. (2018). Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Frontiers in neural circuits*, 12.

Gillon, C. J., Pina, J. E., Lecoq, J. A., Ahmed, R., Billeh, Y., Caldejon, S., Groblewski, P., Henley, T. M., Lee, E., Luviano, J., et al. (2021). Learning from unexpected events in the neocortical microcircuit. *bioRxiv*.

Gilra, A. and Gerstner, W. (2017). Predicting non-linear dynamics by stable local learning in a recurrent spiking neural network. *Elife*, 6:e28295.

Girardeau, G., Benchenane, K., Wiener, S. I., Buzsáki, G., and Zugaro, M. B. (2009). Selective suppression of hippocampal ripples impairs spatial memory. *Nature neuroscience*, 12(10):1222–1223.

Glennon, E., Carcea, I., Martins, A. R. O., Multani, J., Shehu, I., Svirsky, M. A., and Froemke, R. C. (2019). Locus coeruleus activation accelerates perceptual learning. *Brain research*, 1709:39–49.

Gold, J. I., Law, C.-T., Connolly, P., and Bennur, S. (2010). Relationships between the threshold and slope of psychometric and neurometric functions during perceptual learning: implications for neuronal pooling. *Journal of neurophysiology*, 103(1):140–154.

Gold, J. I. and Shadlen, M. N. (2007). The neural basis of decision making. *Annual review of neuroscience*, 30.

Goldman, M. S., Levine, J. H., Major, G., Tank, D. W., and Seung, H. (2003). Robust persistent neural activity in a model integrator with multiple hysteretic dendrites per neuron. *Cerebral cortex*, 13(11):1185–1195.

Golkar, S., Lipshutz, D., Bahroun, Y., Sengupta, A., and Chklovskii, D. (2020a). A simple normative

network approximates local non-hebbian learning in the cortex. *Advances in Neural Information Processing Systems*, 33.

Golkar, S., Lipshutz, D., Bahroun, Y., Sengupta, A. M., and Chklovskii, D. B. (2020b). A biologically plausible neural network for local supervision in cortical microcircuits. *arXiv preprint arXiv:2011.15031*.

Golub, M. D., Sadtler, P. T., Oby, E. R., Quick, K. M., Ryu, S. I., Tyler-Kabara, E. C., Batista, A. P., Chase, S. M., and Byron, M. Y. (2018). Learning by neural reassociation. *Nature neuroscience*, 21(4):607–616.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Goroshin, R., Bruna, J., Tompson, J., Eigen, D., and LeCun, Y. (2015). Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE international conference on computer vision*, pages 4086–4093.

Graupner, M. and Brunel, N. (2010). Mechanisms of induction and maintenance of spike-timing dependent plasticity in biophysical synapse models. *Frontiers in computational neuroscience*, 4:136.

Gu, Q. and Singer, W. (1995). Involvement of serotonin in developmental plasticity of kitten visual cortex. *European Journal of Neuroscience*, 7(6):1146–1153.

Guerguiev, J., Lillicrap, T. P., and Richards, B. A. (2017). Towards deep learning with segregated dendrites. *ELife*, 6:e22901.

Guerreiro, I., Gu, Z., Yakel, J., and Gutkin, B. (2020). Disinhibitory and neuromodulatory regulation of hippocampal synaptic plasticity. *bioRxiv*.

Guo, W., Robert, B., and Polley, D. B. (2019). The cholinergic basal forebrain links auditory stimuli with delayed reinforcement to support learning. *Neuron*, 103(6):1164–1177.

Habenschuss, S., Bill, J., and Nessler, B. (2012). Homeostatic plasticity in Bayesian spiking networks as Expectation Maximization with posterior constraints. *Advances in neural information processing systems*, 25:773–781.

Haimerl, C., Ruff, D. A., Cohen, M. R., Savin, C., and Simoncelli, E. P. (2021). Targeted comodulation supports flexible and accurate decoding in v1. *bioRxiv*.

Haimerl, C., Savin, C., and Simoncelli, E. (2019). Flexible information routing in neural populations through stochastic comodulation. *Advances in Neural Information Processing Systems*, 32.

Hangya, B., Ranade, S. P., Lorenc, M., and Kepecs, A. (2015). Central cholinergic neurons are rapidly recruited by reinforcement feedback. *Cell*, 162(5):1155–1168.

Hanks, T. D., Ditterich, J., and Shadlen, M. N. (2006). Microstimulation of macaque area lip affects decision-making in a motion discrimination task. *Nature neuroscience*, 9(5):682–689.

Hanks, T. D., Kopec, C. D., Brunton, B. W., Duan, C. A., Erlich, J. C., and Brody, C. D. (2015). Distinct relationships of parietal and prefrontal cortices to evidence accumulation. *Nature*, 520(7546):220–223.

Hebb, D. O. (1949). *The organisation of behaviour: a neuropsychological theory*. Science Editions New York.

Heess, N., TB, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S., et al. (2017). Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*.

Heibeck, T. H. and Markman, E. M. (1987). Word learning in children: An examination of fast mapping. *Child development*, pages 1021–1034.

Hennequin, G., Vogels, T. P., and Gerstner, W. (2012). Non-normal amplification in random balanced neuronal networks. *Physical Review E*, 86(1):011909.

Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. (1995). The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hoerzer, G. M., Legenstein, R., and Maass, W. (2014). Emergence of complex computational structures from chaotic neural networks through reward-modulated hebbian learning. *Cerebral cortex*, 24(3):677–690.

Hong, G. and Lieber, C. M. (2019). Novel electrode technologies for neural recordings. *Nature Reviews Neuroscience*, 20(6):330–345.

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

Hopfield, J. J. (1987). Neural networks and physical systems with emergent collective computational abilities. In *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*, pages 411–415. World Scientific.

Horwitz, G. D. and Newsome, W. T. (1999). Separate signals for target selection and movement specification in the superior colliculus. *Science*, 284(5417):1158–1161.

Illing, B., Ventura, J., Bellec, G., and Gerstner, W. (2021). Local plasticity rules can learn deep representations using self-supervised contrastive predictions. *Advances in Neural Information Processing Systems*, 34.

Insanally, M. N., Köver, H., Kim, H., and Bao, S. (2009). Feature-dependent sensitive periods in the development of complex sound representation. *Journal of Neuroscience*, 29(17):5456–5462.

Isomura, T., Shimazaki, H., and Friston, K. J. (2022). Canonical neural networks perform active inference. *Communications Biology*, 5(1):1–15.

Jabri, M. and Flower, B. (1992). Weight perturbation: An optimal architecture and learning technique for analog vlsi feedforward and recurrent multilayer networks. *IEEE Transactions on Neural Networks*, 3(1):154–157.

Jackson, Z., Souza, C., Flaks, J., Pan, Y., Nicolas, H., and Thite, A. (2018). Jakobovski/free-spoken-digit-dataset: v1. 0.8.

Jeurissen, D., Shushruth, S., El-Shamayleh, Y., Horwitz, G. D., and Shadlen, M. N. (2021). Deficits in decision-making induced by parietal cortex inactivation are compensated at two time scales. *bioRxiv*.

Jonas, E. and Kording, K. P. (2017). Could a neuroscientist understand a microprocessor? *PLoS computational biology*, 13(1):e1005268.

Jordan, J., Schmidt, M., Senn, W., and Petrovici, M. A. (2021). Evolving interpretable plasticity for spiking networks. *Elife*, 10.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134.

Kappel, D., Nessler, B., and Maass, W. (2014). STDP installs in winner-take-all circuits an online approximation to hidden markov model learning. *PLoS Comput Biol*, 10(3):e1003511.

Katz, L. N., Yates, J. L., Pillow, J. W., and Huk, A. C. (2016). Dissociated functional significance of decision-related activity in the primate dorsal stream. *Nature*, 535(7611):285–288.

Kell, A. J., Yamins, D. L., Shook, E. N., Norman-Haignere, S. V., and McDermott, J. H. (2018). A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644.

Kepple, D. R., Engelken, R., and Rajan, K. (2021). Curriculum learning as a tool to uncover learning principles in the brain. In *International Conference on Learning Representations*.

Kersten, D., Mamassian, P., and Yuille, A. (2004). Object perception as Bayesian inference. *Annu. Rev. Psychol.*, 55:271–304.

Khalvati, K., Kiani, R., and Rao, R. P. N. (2021). Bayesian inference with incomplete knowledge explains perceptual confidence and its deviations from accuracy. *Nat Commun*, 12(1):5704.

Kiani, R., Corthell, L., and Shadlen, M. N. (2014a). Choice certainty is informed by both evidence and decision time. *Neuron*, 84(6):1329–1342.

Kiani, R., Cueva, C. J., Reppas, J. B., and Newsome, W. T. (2014b). Dynamics of neural population responses in prefrontal cortex indicate changes of mind on single trials. *Current Biology*, 24(13):1542–1547.

Kilgard, M. P. and Merzenich, M. M. (1998). Cortical map reorganization enabled by nucleus basalis activity. *Science*, 279(5357):1714–1718.

Kim, J.-N. and Shadlen, M. N. (1999). Neural correlates of a decision in the dorsolateral prefrontal cortex of the macaque. *Nature neuroscience*, 2(2):176–185.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes.

Kirby, K. G. (2006). A tutorial on helmholtz machines. *Department of Computer Science, Northern Kentucky University*.

Knill, D. C. and Richards, W. (1996). *Perception as Bayesian inference*. Cambridge University Press.

Körding, K. P. and König, P. (2001). Supervised and unsupervised learning with two sites of synaptic integration. *Journal of computational neuroscience*, 11(3):207–215.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Kuchibhotla, K. V., Gill, J. V., Lindsay, G. W., Papadoyannis, E. S., Field, R. E., Sten, T. A. H., Miller, K. D., and Froemke, R. C. (2017). Parallel processing by cortical inhibition enables context-dependent behavior. *Nature neuroscience*, 20(1):62–71.

Kuśmierz, Ł., Isomura, T., and Toyoizumi, T. (2017). Learning with three factors: modulating hebbian plasticity with errors. *Current opinion in neurobiology*, 46:170–177.

Kutschireiter, A., Surace, S. C., Sprekeler, H., and Pfister, J.-P. (2017). Nonlinear Bayesian filtering and learning: a neuronal dynamics for perception. *Scientific reports*, 7(1):1–13.

Lange, R. D., Shivkumar, S., Chattoraj, A., and Haefner, R. M. (2020). Bayesian encoding and decoding as distinct perspectives on neural coding. *bioRxiv*.

Larkum, M. (2013). A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in neurosciences*, 36(3):141–151.

Larkum, M. E., Zhu, J. J., and Sakmann, B. (1999). A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398(6725):338–341.

Laszlovszky, T., Schlingloff, D., Hegedüs, P., Freund, T. F., Gulyás, A., Kepecs, A., and Hangya, B. (2020). Distinct synchronization, cortical coupling and behavioral function of two basal forebrain cholinergic neuron types. *Nature neuroscience*, 23(8):992–1003.

Leão, R. N., Mikulovic, S., Leão, K. E., Munguba, H., Gezelius, H., Enjin, A., Patra, K., Eriksson, A., Loew, L. M., Tort, A. B., et al. (2012). Olm interneurons differentially modulate ca3 and entorhinal inputs to hippocampal CA1 neurons. *Nature neuroscience*, 15(11):1524–1530.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1989a). Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989b). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Lee, D.-H., Zhang, S., Fischer, A., and Bengio, Y. (2015). Difference target propagation. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 498–515. Springer.

Legenstein, R., Chase, S. M., Schwartz, A. B., and Maass, W. (2010). A reward-modulated Hebbian learning rule can explain experimentally observed network reorganization in a brain control task. *Journal of Neuroscience*, 30(25):8400–8410.

Lesch, K.-P. and Waider, J. (2012). Serotonin in the modulation of neural plasticity and networks: implications for neurodevelopmental disorders. *Neuron*, 76(1):175–191.

Letzkus, J. J., Kampa, B. M., and Stuart, G. J. (2006). Learning rules for spike timing-dependent plasticity depend on dendritic synapse location. *Journal of Neuroscience*, 26(41):10420–10429.

Levelt, C. N. and Hübener, M. (2012). Critical-period plasticity in the visual cortex. *Annual review of neuroscience*, 35:309–330.

Levenstein, D., Alvarez, V. A., Amarasingham, A., Azab, H., Gerkin, R. C., Hasenstaub, A., Iyer,

R., Jolivet, R. B., Marzen, S., Monaco, J. D., et al. (2020). On the role of theory and modeling in neuroscience. *arXiv preprint arXiv:2003.13825*.

Li, N., Daie, K., Svoboda, K., and Druckmann, S. (2016). Robust neuronal dynamics in premotor cortex during motor planning. *Nature*, 532(7600):459–464.

Li, W., Piëch, V., and Gilbert, C. D. (2004). Perceptual learning and top-down influences in primary visual cortex. *Nature neuroscience*, 7(6):651–657.

Licata, A. M., Kaufman, M. T., Raposo, D., Ryan, M. B., Sheppard, J. P., and Churchland, A. K. (2017). Posterior parietal cortex guides visual decisions in rats. *Journal of Neuroscience*, 37(19):4954–4966.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):1–10.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346.

Luo, L., Callaway, E. M., and Svoboda, K. (2018). Genetic dissection of neural circuits: a decade of progress. *Neuron*, 98(2):256–281.

Magee, J. C. and Johnston, D. (1997). A synaptically controlled, associative signal for hebbian plasticity in hippocampal neurons. *Science*, 275(5297):209–213.

Maheswaranathan, N., Williams, A. H., Golub, M. D., Ganguli, S., and Sussillo, D. (2019). Universality and individuality in neural dynamics across large populations of recurrent networks. *Advances in neural information processing systems*, 2019:15629.

Mamassian, P., Landy, M., and Maloney, L. T. (2002). Bayesian modelling of visual perception. *Probabilistic models of the brain*, pages 13–36.

Mante, V., Sussillo, D., Shenoy, K. V., and Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 503(7474):78–84.

Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297):213–215.

Marlin, B. J., Mitre, M., D'amour, J. A., Chao, M. V., and Froemke, R. C. (2015). Oxytocin enables maternal behaviour by balancing cortical inhibition. *Nature*, 520(7548):499–504.

Marschall, O., Cho, K., and Savin, C. (2019). A unified framework of online learning algorithms for training recurrent neural networks. *arXiv preprint arXiv:1907.02649*.

Marschall, O., Cho, K., and Savin, C. (2020). A unified framework of online learning algorithms for training recurrent neural networks. *Journal of machine learning research*.

Martens, J. and Sutskever, I. (2011). Learning recurrent neural networks with Hessian-free optimization. In *Proc of the 28th International Conference on Machine Learning (ICML-11)*, pages 1033–1040.

Martin, S. J., Grimwood, P. D., and Morris, R. G. (2000). Synaptic plasticity and memory: an evaluation of the hypothesis. *Annual review of neuroscience*, 23(1):649–711.

Martins, A. R. O. and Froemke, R. C. (2015). Coordinated forms of noradrenergic plasticity in the locus coeruleus and primary auditory cortex. *Nature neuroscience*, 18(10):1483–1492.

Meulemans, A., Carzaniga, F. S., Suykens, J. A., Sacramento, J., and Grewe, B. F. (2020). A theoretical framework for target propagation. *arXiv preprint arXiv:2006.14331*.

Miconi, T. (2017). Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife*, 6:e20899.

Mineault, P., Bakhtiari, S., Richards, B., and Pack, C. (2021). Your head is there to move you around: Goal-driven models of the primate dorsal pathway. *Advances in Neural Information Processing Systems*, 34.

Mnih, A. and Gregor, K. (2014). Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799. PMLR.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Mochol, G., Kiani, R., and Moreno-Bote, R. (2021). Prefrontal cortex represents heuristics that shape choice bias and its integration into future behavior. *Current Biology*, 31(6):1234–1244.

Murphy, T. H. and Corbett, D. (2009). Plasticity during stroke recovery: from synapse to behaviour. *Nature reviews neuroscience*, 10(12):861–872.

Murray, E. A. and Baxter, M. G. (2006). Cognitive neuroscience and nonhuman primates: lesion studies. *Methods in mind*, 43:69.

Murray, J. M. (2019). Local online learning in recurrent networks with random feedback. *ELife*, 8:e43299.

Nassar, J., Linderman, S. W., Bugallo, M., and Park, I. M. (2018). Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. *arXiv preprint arXiv:1811.12386*.

Nayebi, A., Attinger, A., Campbell, M., Hardcastle, K., Low, I., Mallory, C., Mel, G., Sorscher, B., Williams, A., Ganguli, S., et al. (2021). Explaining heterogeneity in medial entorhinal cortex with task-driven neural networks. *Advances in Neural Information Processing Systems*, 34.

Nayebi, A., Srivastava, S., Ganguli, S., and Yamins, D. L. (2020). Identifying learning rules from neural network observables. *arXiv preprint arXiv:2010.11765*.

Neftci, E. O., Mostafa, H., and Zenke, F. (2019). Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63.

Newsome, W. T., Britten, K. H., and Movshon, J. A. (1989). Neuronal correlates of a perceptual decision. *Nature*, 341(6237):52–54.

Newsome, W. T. and Pare, E. B. (1988). A selective impairment of motion perception following lesions of the middle temporal visual area (mt). *Journal of Neuroscience*, 8(6):2201–2211.

Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3):139–154.

Obeid, D., Ramambason, H., and Pehlevan, C. (2019). Structured and deep similarity matching via structured and deep hebbian networks. *arXiv preprint arXiv:1910.04958*.

O'Donohue, T. L., Millington, W. R., Handelmann, G. E., Contreras, P. C., and Chronwall, B. M. (1985). On the 50th anniversary of dale's law: multiple neurotransmitter neurons. *Trends in Pharmacological Sciences*, 6:305–308.

Ohl, F. W. and Scheich, H. (2005). Learning-induced plasticity in animal and human auditory cortex. *Current opinion in neurobiology*, 15(4):470–477.

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273.

Olshausen, B. A. et al. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Otani, S., Daniel, H., Roisin, M.-P., and Crepel, F. (2003). Dopaminergic modulation of long-term synaptic plasticity in rat prefrontal neurons. *Cerebral cortex*, 13(11):1251–1256.

Pachitariu, M., Stringer, C., Dipoppa, M., Schröder, S., Rossi, L. F., Dalgleish, H., Carandini, M., and Harris, K. D. (2017). Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *BioRxiv*.

Palmer, J., Huk, A. C., and Shadlen, M. N. (2005). The effect of stimulus strength on the speed and accuracy of a perceptual decision. *Journal of vision*, 5(5):1–1.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Pavlides, C. and Winson, J. (1989). Influences of hippocampal place cell firing in the awake state on the activity of these cells during subsequent sleep episodes. *Journal of neuroscience*, 9(8):2907–2918.

Payeur, A., Guerguiev, J., Zenke, F., Richards, B., and Naud, R. (2020). Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *bioRxiv*.

Payeur, A., Guerguiev, J., Zenke, F., Richards, B. A., and Naud, R. (2021). Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature neuroscience*, pages 1–10.

Pearlmutter, B. A. (1989). Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1(2):263–269.

Pehlevan, C., Hu, T., and Chklovskii, D. B. (2015). A hebbian/anti-hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data. *Neural computation*, 27(7):1461–1495.

Pehlevan, C., Sengupta, A. M., and Chklovskii, D. B. (2017). Why do similarity matching objectives lead to hebbian/anti-hebbian networks? *Neural computation*, 30(1):84–124.

Peixoto, D., Verhein, J. R., Kiani, R., Kao, J. C., Nuyujukian, P., Chandrasekaran, C., Brown, J., Fong, S., Ryu, S. I., Shenoy, K. V., et al. (2021). Decoding and perturbing decision states in real time. *Nature*, 591(7851):604–609.

Polley, D. B., Steinberg, E. E., and Merzenich, M. M. (2006). Perceptual learning directs auditory cortical map reorganization through top-down influences. *Journal of neuroscience*, 26(18):4970–4982.

Portilla, J. and Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, 40(1):49–70.

Pozzi, I., Bohte, S., and Roelfsema, P. (2020). Attention-gated brain propagation: How the brain can implement reward-based error backpropagation. *Advances in Neural Information Processing Systems*, 33.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Rajasethupathy, P., Ferenczi, E., and Deisseroth, K. (2016a). Targeting neural circuits. *Cell*, 165(3):524–534.

Rajasethupathy, P., Ferenczi, E., and Deisseroth, K. (2016b). Targeting neural circuits. *Cell*, 165(3):524–534.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR.

Rao, R. P. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87.

Rashid, S. K., Pedrosa, V., Dufour, M. A., Moore, J. J., Chavlis, S., Delatorre, R. G., Poirazi, P., Clopath, C., and Basu, J. (2020). The dendritic spatial code: branch-specific place tuning and its experience-dependent decoupling. *bioRxiv*.

Rasmusson, D. (2000). The role of acetylcholine in cortical synaptic plasticity. *Behavioural brain research*, 115(2):205–218.

Ratcliff, R., Hasegawa, Y. T., Hasegawa, R. P., Childers, R., Smith, P. L., and Segraves, M. A. (2011). Inhibition in superior colliculus neurons in a brightness discrimination task? *Neural Computation*, 23(7):1790–1820.

Rayport, S. G. and Schacher, S. (1986). Synaptic plasticity in vitro: cell culture of identified aplysia neurons mediating short-term habituation and sensitization. *Journal of Neuroscience*, 6(3):759–763.

Recanzone, G. H., Merzenich, M. M., and Jenkins, W. M. (1992a). Frequency discrimination training engaging a restricted skin surface results in an emergence of a cutaneous response zone in cortical area 3a. *Journal of Neurophysiology*, 67(5):1057–1070.

Recanzone, G. H., Merzenich, M. M., Jenkins, W. M., Grajski, K. A., and Dinse, H. R. (1992b). Topographic reorganization of the hand representation in cortical area 3b owl monkeys trained in a frequency-discrimination task. *journal of Neurophysiology*, 67(5):1031–1056.

Recanzone, G. H., Schreiner, C. E., and Merzenich, M. M. (1993). Plasticity in the frequency representation of primary auditory cortex following discrimination training in adult owl monkeys. *Journal of Neuroscience*, 13(1):87–103.

Reed, A., Riley, J., Carraway, R., Carrasco, A., Perez, C., Jakkamsetti, V., and Kilgard, M. P. (2011). Cortical map plasticity improves learning but is not necessary for improved performance. *Neuron*, 70(1):121–131.

Reynolds, J. N. and Wickens, J. R. (2002). Dopamine-dependent plasticity of corticostriatal synapses. *Neural networks*, 15(4-6):507–521.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

Richards, B. A. and Lillicrap, T. P. (2019). Dendritic solutions to the credit assignment problem. *Current opinion in neurobiology*, 54:28–36.

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R. P., de Berker, A., Ganguli, S., et al. (2019). A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770.

Rigotti, M., Ben Dayan Rubin, D. D., Wang, X.-J., and Fusi, S. (2010). Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. *Frontiers in computational neuroscience*, 4:24.

Roberts, G. O., Tweedie, R. L., et al. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363.

Roelfsema, P. R. and Ooyen, A. v. (2005). Attention-gated reinforcement learning of internal representations for classification. *Neural computation*, 17(10):2176–2214.

Roelfsema, P. R., van Ooyen, A., and Watanabe, T. (2010). Perceptual learning rules based on reinforcers and attention. *Trends in cognitive sciences*, 14(2):64–71.

Roitman, J. D. and Shadlen, M. N. (2002). Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task. *Journal of neuroscience*, 22(21):9475–9489.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Roth, C., Kanitscheider, I., and Fiete, I. (2018). Kernel rnn learning (kernl). In *International Conference on Learning Representations*.

Roweis, S. and Ghahramani, Z. (1999). A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345.

Rozell, C. J., Johnson, D. H., Baraniuk, R. G., and Olshausen, B. A. (2008). Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10):2526–2563.

Rudolph, K. and Pasternak, T. (1999). Transient and permanent deficits in motion perception after lesions of cortical areas mt and mst in the macaque monkey. *Cerebral Cortex*, 9(1):90–100.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Rummell, B. P., Klee, J. L., and Sigurdsson, T. (2016). Attenuation of responses to self-generated sounds in auditory cortical neurons. *Journal of Neuroscience*, 36(47):12010–12026.

Rust, N. C. and DiCarlo, J. J. (2010). Selectivity and tolerance ("invariance") both increase as visual information propagates from cortical area v4 to it. *Journal of Neuroscience*, 30(39):12978–12995.

Sacramento, J., Costa, R. P., Bengio, Y., and Senn, W. (2017). Dendritic error backpropagation in deep cortical microcircuits. *arXiv preprint arXiv:1801.00062*.

Sajid, N., Ball, P. J., Parr, T., and Friston, K. J. (2021). Active inference: demystified and compared. *Neural Computation*, 33(3):674–712.

Salzman, C. D., Britten, K. H., and Newsome, W. T. (1990). Cortical microstimulation influences perceptual judgements of motion direction. *Nature*, 346(6280):174–177.

Saudargiene, A., Cobb, S., and Graham, B. P. (2015). A computational study on plasticity during theta cycles at Schaffer collateral synapses on CA1 pyramidal cells in the hippocampus. *Hippocampus*, 25(2):208–218.

Savin, C. and Triesch, J. (2014). Emergence of task-dependent representations in working memory circuits. *Frontiers in Computational Neuroscience*, 8:57.

Scellier, B. and Bengio, Y. (2017). Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24.

Schiess, M., Urbanczik, R., and Senn, W. (2016). Somato-dendritic synaptic plasticity and error-backpropagation in active dendrites. *PLoS computational biology*, 12(2):e1004638.

Schiller, P. H., True, S. D., and Conway, J. L. (1979). Effects of frontal eye field and superior colliculus ablations on eye movements. *Science*, 206(4418):590–592.

Schneider, D. M., Sundararajan, J., and Mooney, R. (2018). A cortical filter that learns to suppress the acoustic consequences of movement. *Nature*, 561(7723):391–395.

Schoups, A., Vogels, R., Qian, N., and Orban, G. (2001). Practising orientation identification improves orientation coding in v1 neurons. *Nature*, 412(6846):549–553.

Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599.

Seung, H. S. (2003). Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, 40(6):1063–1073.

Sewell, G. D. (1970). Ultrasonic communication in rodents. *Nature*, 227(5256):410–410.

Sezener, E., Grabska-Barwinska, A., Kostadinov, D., Beau, M., Krishnagopal, S., Budden, D., Hutter, M., Veness, J., Botvinick, M., Clopath, C., et al. (2021). A rapid and efficient learning rule for biological neural circuits. *bioRxiv*.

Shadlen, M. N., Hanks, T. D., Churchland, A. K., Kiani, R., and Yang, T. (2006). The speed and accuracy of a simple perceptual decision: a mathematical primer. *Bayesian brain: Probabilistic approaches to neural coding*, pages 209–37.

Shadlen, M. N. and Kiani, R. (2013). Decision making as a window on cognition. *Neuron*, 80(3):791–806.

Shadlen, M. N. and Newsome, W. T. (2001a). Neural basis of a perceptual decision in the parietal cortex (area lip) of the rhesus monkey. *Journal of neurophysiology*, 86(4):1916–1936.

Shadlen, M. N. and Newsome, W. T. (2001b). Neural basis of a perceptual decision in the parietal cortex (area lip) of the rhesus monkey. *Journal of neurophysiology*, 86(4):1916–1936.

Sheahan, H. R., Franklin, D. W., and Wolpert, D. M. (2016). Motor planning, not execution, separates motor memories. *Neuron*, 92(4):773–779.

Shinoe, T., Matsui, M., Taketo, M. M., and Manabe, T. (2005). Modulation of synaptic plasticity by physiological activation of m1 muscarinic acetylcholine receptors in the mouse hippocampus. *Journal of Neuroscience*, 25(48):11194–11200.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.

Simoncelli, E. P. (2003). Vision and the statistics of the visual environment. *Current opinion in neurobiology*, 13(2):144–149.

Simoncelli, E. P. and Heeger, D. J. (1998). A model of neuronal responses in visual area mt. *Vision research*, 38(5):743–761.

Simoncelli, E. P. and Olshausen, B. A. (2001). Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216.

Singla, S., Dempsey, C., Warren, R., Enikolopov, A. G., and Sawtell, N. B. (2017). A cerebellum-like circuit in the auditory system cancels responses to self-generated sounds. *Nature Neuroscience*, 20(7):943–950.

Sjöström, J., Gerstner, W., et al. (2010). Spike-timing dependent plasticity. *Spike-timing dependent plasticity*, 35(0):0–0.

Sjöström, P. J. and Häusser, M. (2006). A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons. *Neuron*, 51(2):227–238.

Sohn, H., Narain, D., Meirhaeghe, N., and Jazayeri, M. (2019). Bayesian computation through cortical latent dynamics. *Neuron*, 103(5):934–947.

Sompolinsky, H. and Kanter, I. (1986). Temporal association in asymmetric neural networks. *Physical review letters*, 57(22):2861.

Stickgold, R. (2005). Sleep-dependent memory consolidation. *Nature*, 437(7063):1272–1278.

Strogatz, S. H. (2018). *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. CRC press.

Sussillo, D. and Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Takesian, A. E., Bogart, L. J., Lichtman, J. W., and Hensch, T. K. (2018). Inhibitory circuit gating of auditory critical-period plasticity. *Nature neuroscience*, 21(2):218–227.

Tervo, D. G., Proskurin, M., Manakov, M., Kabra, M., Vollmer, A., Branson, K., and Karpova, A. Y. (2014). Behavioral variability through stochastic choice and its gating by anterior cingulate cortex. *Cell*, 159(1):21–32.

Thura, D. and Cisek, P. (2014). Deliberation and commitment in the premotor and primary motor cortex during dynamic decision making. *Neuron*, 81(6):1401–1416.

Tishby, N., Pereira, F. C., and Bialek, W. (2000). The information bottleneck method. *arXiv preprint physics/0004057*.

Tremblay, S., Acker, L., Afraz, A., Albaugh, D. L., Amita, H., Andrei, A. R., Angelucci, A., Aschner, A., Balan, P. F., Basso, M. A., et al. (2020). An open resource for non-human primate optogenetics. *Neuron*, 108(6):1075–1090.

Urbanczik, R. and Senn, W. (2014). Learning by the dendritic prediction of somatic spiking. *Neuron*, 81(3):521–528.

Vaidya, A. R., Pujara, M. S., Petrides, M., Murray, E. A., and Fellows, L. K. (2019). Lesion studies in contemporary neuroscience. *Trends in cognitive sciences*.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12).

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

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

Wald, A. and Wolfowitz, J. (1950). Bayes solutions of sequential decision problems. *The Annals of Mathematical Statistics*, pages 82–99.

Waskom, M. L., Okazawa, G., and Kiani, R. (2019). Designing and interpreting psychophysical investigations of cognition. *Neuron*, 104(1):100–112.

Weinberger, N. M. (1993). Learning-induced changes of auditory receptive fields. *Current opinion in neurobiology*, 3(4):570–577.

Weiss, Y., Simoncelli, E. P., and Adelson, E. H. (2002). Motion illusions as optimal percepts. *Nature neuroscience*, 5(6):598–604.

Werbos, P. (1974). Beyond regression:" new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Werfel, J., Xie, X., and Seung, H. S. (2003). Learning curves for stochastic gradient descent in linear feedforward networks. In *NIPS*, pages 1197–1204. Citeseer.

Werfel, J., Xie, X., and Seung, H. S. (2004). Learning curves for stochastic gradient descent in linear feedforward networks. In *Advances in neural information processing systems*, pages 1197–1204.

Wiegert, J. S., Mahn, M., Prigge, M., Printz, Y., and Yizhar, O. (2017). Silencing neurons: tools, applications, and experimental constraints. *Neuron*, 95(3):504–529.

Wiesel, T. N. and Hubel, D. H. (1963). Single-cell responses in striate cortex of kittens deprived of vision in one eye. *Journal of neurophysiology*, 26(6):1003–1017.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

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

Wong, K.-F. and Wang, X.-J. (2006). A recurrent network mechanism of time integration in perceptual decisions. *Journal of Neuroscience*, 26(4):1314–1328.

Xiao, Z.-C., Lin, K. K., and Young, L.-S. (2021). A data-informed mean-field approach to mapping of cortical parameter landscapes. *PLOS Computational Biology*, 17(12):e1009718.

Xie, X. and Seung, H. S. (2003). Equivalence of backpropagation and contrastive hebbian learning in a layered network. *Neural computation*, 15(2):441–454.

Yamins, D. L. and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624.

Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., and Wang, X.-J. (2019). Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306.

Yartsev, M. M., Hanks, T. D., Yoon, A. M., and Brody, C. D. (2018). Causal contribution and dynamical encoding in the striatum during evidence accumulation. *Elife*, 7:e34929.

Yoshihara, M. and Yoshihara, M. (2018). 'necessary and sufficient'in biology is not necessarily necessary–confusions and erroneous conclusions resulting from misapplied logic in the field of biology, especially neuroscience. *Journal of neurogenetics*, 32(2):53–64.

Záborszky, L., Gombkoto, P., Varsanyi, P., Gielow, M. R., Poe, G., Role, L. W., Ananth, M., Rajebhosale, P., Talmage, D. A., Hasselmo, M. E., et al. (2018). Specific basal forebrain–cortical cholinergic circuits coordinate cognitive operations. *Journal of Neuroscience*, 38(44):9446–9458.

Zhang, K., Yang, Z., and Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384.

Zhang, L. I., Bao, S., and Merzenich, M. M. (2001). Persistent and specific influences of early acoustic environments on primary auditory cortex. *Nature neuroscience*, 4(11):1123–1130.

Zhang, L. I., Tao, H. W., Holt, C. E., Harris, W. A., and Poo, M.-m. (1998). A critical window for cooperation and competition among developing retinotectal synapses. *Nature*, 395(6697):37–44.

Zhao, Y. and Park, I. M. (2016). Interpretable nonlinear dynamic modeling of neural trajectories. *arXiv preprint arXiv:1608.06546*.

Zhou, X., Panizzutti, R., de Villers-Sidani, É., Madeira, C., and Merzenich, M. M. (2011). Natural restoration of critical period plasticity in the juvenile and adult primary auditory cortex. *Journal of Neuroscience*, 31(15):5625–5634.

Zhou, Y. and Freedman, D. J. (2019a). Posterior parietal cortex plays a causal role in perceptual and categorical decisions. *Science*, 365(6449):180–185.

Zhou, Y. and Freedman, D. J. (2019b). Posterior parietal cortex plays a causal role in perceptual and categorical decisions. *Science*, 365(6449):180–185.

Ziemba, C. M., Freeman, J., Movshon, J. A., and Simoncelli, E. P. (2016). Selectivity and tolerance for visual texture in macaque v2. *Proceedings of the National Academy of Sciences*, 113(22):E3140–E3149.

Zigmond, M. J., Abercrombie, E. D., Berger, T. W., Grace, A. A., and Stricker, E. M. (1990). Compensations after lesions of central dopaminergic neurons: some clinical and basic implications. *Trends in neurosciences*, 13(7):290–296.