Impression learning: Online representation learning with synaptic plasticity

Colin Bredenberg
Center for Neural Science
New York University
cjb617@nyu.edu

Benjamin S. H. Lyo
Center for Neural Science
New York University
blyo@nyu.edu

Eero P. Simoncelli
Center for Neural Science,
New York University
Flatiron Institute, Simons Foundation
eero.simoncelli@nyu.edu

Cristina Savin
Center for Neural Science,
Center for Data Science
New York University
csavin@nyu.edu

Abstract

Understanding how the brain constructs statistical models of the sensory world remains a longstanding challenge for computational neuroscience. Here, we derive an unsupervised local synaptic plasticity rule that trains neural circuits to infer latent structure from sensory stimuli via a novel loss function for approximate online Bayesian inference. The learning algorithm is driven by a local error signal computed between two factors that jointly contribute to neural activity: stimulus drive and internal predictions — the network's 'impression' of the stimulus. Physiologically, we associate these two components with the basal and apical dendrites of pyramidal neurons, respectively. We show that learning can be implemented online, is capable of capturing temporal dependencies in continuous input streams, and generalizes to hierarchical architectures. Furthermore, we demonstrate both analytically and empirically that the algorithm is more data-efficient than a three-factor plasticity alternative, enabling it to learn statistics of high-dimensional, naturalistic inputs. Overall, the model provides a bridge from mechanistic accounts of synaptic plasticity to algorithmic descriptions of unsupervised probabilistic learning and inference.

1 Introduction

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. Because the true underlying structure of the data is rarely accessible, this "representation learning" must be largely unsupervised. Framing perception in the language of Bayesian inference has proven fruitful in perceptual and cognitive science [1–4], 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 [5, 6].

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 [7], to efficient population codes that embed prior structure [8] to either parametric [9, 10] or sampling-based [11] variational inference. Learning generally takes the form of optimizing a probabilistic objective, either by backpropagation [9, 10] or through local parameter updates, which match biological learning more

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

closely [11, 7, 12, 13]. While these models are mostly restricted to static stimuli, several instances also operate over time [14–16].

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 BP 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 [17–25]. Locally implementing temporal credit assignment is a bigger challenge [26, 27].

Our work, which we have dubbed 'impression learning' (IL), combines the tradition of probabilistic learning [11, 14] 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. We explore the empirical and mathematical relationships between IL and three other methods: backpropagation (BP) [28], the Wake-Sleep (WS) algorithm [29], and a specific form of neural variational inference (NVI*) [30, 31]. We further demonstrate that IL scales to naturalistic stimuli and multilayer network architectures [1].

2 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.[2] 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 [5]. 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 $\tilde{p}_\theta$:

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

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

[1] Code provided at: `https://github.com/colinbredenberg/Impression-Learning-Camera-Ready`.

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

$\theta = [\theta_p, \theta_q]$. We define an objective of the form:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{\lambda,\mathbf{z}}\left[KL[\tilde{q}_\theta || \tilde{p}_\theta]\right] \\ &= \mathbb{E}_{\lambda,\mathbf{z}}\left[\int [\log \tilde{q}_\theta - \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r} d\mathbf{s}\right]. \end{aligned} \tag{2}$$

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. $\theta = [\theta_q, \theta_p]$:

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

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 $\nabla_\theta \tilde{q}_\theta = \nabla_\theta e^{\log \tilde{q}_\theta} = [\nabla_\theta \log \tilde{q}_\theta]\, \tilde{q}_\theta$, which allows us to rewrite our expression as an expectation over $\mathbf{r}$ and $\mathbf{s}$:

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

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

$$\begin{aligned}-\nabla_\theta \mathcal{L} &= \mathbb{E}_{\lambda,\mathbf{z}}\left[\int [\nabla_\theta \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r} d\mathbf{s} + \int \left[\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right] (\nabla_\theta \log \tilde{q}_\theta) \tilde{q}_\theta \; d\mathbf{r} d\mathbf{s}\right] \\ &\approx \mathbb{E}_{\lambda,\mathbf{z}}\left[\int [\nabla_\theta \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r} d\mathbf{s} + \int \left[\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1\right] (\nabla_\theta \log \tilde{q}_\theta) \tilde{q}_\theta \; d\mathbf{r} d\mathbf{s}\right] \\ &= \mathbb{E}_{\lambda,\mathbf{z}}\left[\int [\nabla_\theta \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r} d\mathbf{s} + \int \left[\frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right] (\nabla_\theta \log \tilde{q}_\theta) \tilde{q}_\theta \; d\mathbf{r} d\mathbf{s}\right] \\ &= \mathbb{E}_{\lambda,\mathbf{z}}\left[\int [\nabla_\theta \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r} d\mathbf{s} + \int [\nabla_\theta \log \tilde{q}_\theta]\, \tilde{p}_\theta \; d\mathbf{r} d\mathbf{s}\right]. \end{aligned} \tag{3}$$

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 A). This expansion is the core of our derivation, and not all algorithms take this approach: for this reason, in Appendix B and C 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. (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

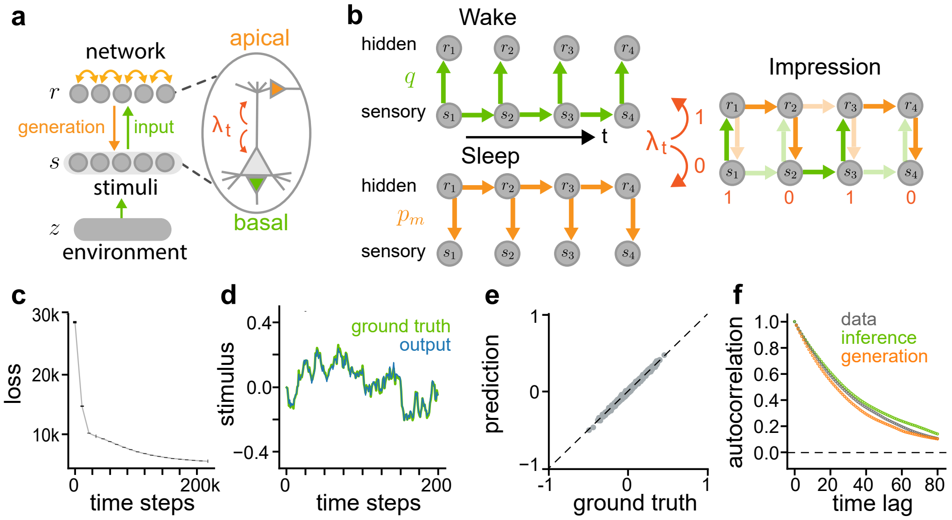

Figure 1: **Network architecture and learning. a.** Model schematic. A neural network receives stimulus inputs at its basal dendrites, and returns lateral and top-down prediction signals via apical synapses. A gate, λ_t, determines whether apical or basal influences dominate network activity. **b.** 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).

we make use of as follows:

$$\begin{aligned}
-\nabla_\theta \mathcal{L} \approx & \mathbb{E}_{\mathbf{z}}\left[\sum_{\lambda}\left[\int [\nabla_\theta \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s} + \int [\nabla_\theta \log \tilde{q}_\theta]\, \tilde{p}_\theta \; d\mathbf{r}d\mathbf{s}\right] \; p(\lambda)\right] \\
= & \mathbb{E}_{\mathbf{z}}\left[\sum_{\lambda}\int [\nabla_\theta \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}p(\lambda) + \sum_{\lambda'}\int [\nabla_\theta \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s} \; p(\lambda')\right] \\
= & 2\mathbb{E}_{\mathbf{z}}\left[\sum_{\lambda}\int [\nabla_\theta \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s} \; p(\lambda)\right]. \qquad (4)
\end{aligned}$$

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

$$\Delta\theta \propto 2\mathbb{E}_{\lambda_0,\mathbf{z}}\left[\int \left[\sum_t (1-\lambda_t)\nabla_\theta \log q_t + (\lambda_t)\nabla_\theta \log p_{mt}\right] \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}\right]. \qquad (5)$$

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. 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, $\mathbf{r}_t^{\text{gen}}$:

$$\mathbf{r}_t^{\text{gen}} = \left((1-k_t)\mathbf{D}_r + k_t\mathbf{I}\right)\mathbf{r}_{t-1} + \sigma_r^{\text{gen}}\boldsymbol{\eta}_t \tag{6}$$

$$\mathbf{s}_t^{\text{gen}} = f(\mathbf{D}_s\mathbf{r}_t) + \sigma_s^{\text{gen}}\boldsymbol{\xi}_t, \tag{7}$$

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 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:

$$\mathbf{r}_t^{\text{inf}} = f(\mathbf{W}\mathbf{s}_t) + \sigma_r^{\text{inf}}\boldsymbol{\eta}_t, \tag{8}$$

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:

$$\mathbf{s}_t^{\text{inf}} = \bar{\mathbf{s}}(\mathbf{z}_t) + \sigma_s^{\text{inf}}\boldsymbol{\xi}_t, \tag{9}$$

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 $\tilde{q}_\theta$:

$$\mathbf{r}_t = \lambda_t\mathbf{r}_t^{\text{inf}} + (1-\lambda_t)\mathbf{r}_t^{\text{gen}} \tag{10}$$

$$\mathbf{s}_t = \lambda_t\mathbf{s}_t^{\text{inf}} + (1-\lambda_t)\mathbf{s}_t^{\text{gen}}. \tag{11}$$

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:

$$\log q(\mathbf{r}_t|\mathbf{s}_t; \theta_q) = -\frac{1}{2\left(\sigma_r^{\text{inf}}\right)^2}\left\|\mathbf{r}_t - f(\mathbf{W}\mathbf{s}_t)\right\|_2^2 + c_q \tag{12}$$

$$\begin{aligned}\log p_m(\mathbf{r}_t, \mathbf{s}_t|\mathbf{r}_{t-1}, \lambda_t; \theta_p) = &-\frac{1}{2(\sigma_r^{\text{gen}})^2}\left\|\mathbf{r}_t - \left((1-k_t)\mathbf{D}_r + k_t\mathbf{I}\right)\mathbf{r}_{t-1}\right\|_2^2 \\ &-\frac{1}{2(\sigma_s^{\text{gen}})^2}\left\|\mathbf{s}_t - f(\mathbf{D}_s\mathbf{r}_t)\right\|_2^2 + c_p,\end{aligned} \tag{13}$$

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 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):

$$\Delta \mathbf{W}^{(ij)} \propto \frac{1-\lambda_t}{(\sigma_r^{\text{inf}})^2}(\mathbf{r}_t^{(i)} - f(\mathbf{W}\mathbf{s}_t)^{(i)})f'(\mathbf{W}\mathbf{s}_t)^{(i)}\mathbf{s}_t^{(j)} \tag{14}$$

$$\Delta \mathbf{D}_r^{(ii)} \propto \frac{\lambda_t(1-k_t)}{(\sigma_r^{\text{gen}})^2}(\mathbf{r}_t^{(i)} - (\mathbf{D}_r\mathbf{r}_{t-1})^{(i)})\mathbf{r}_{t-1}^{(i)} \tag{15}$$

$$\Delta \mathbf{D}_s^{(ij)} \propto \frac{\lambda_t}{(\sigma_s^{\text{gen}})^2}(\mathbf{s}_t^{(i)} - f(\mathbf{D}_s\mathbf{r}_t)^{(i)})f'(\mathbf{D}_s\mathbf{r}_t)^{(i)}\mathbf{r}_t^{(j)}. \tag{16}$$

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 [32]; 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 [33, 34].

3 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:

$$\mathbf{z}_t = \mathbf{\Lambda}\mathbf{z}_{t-1} + \sigma^{\text{true}}\boldsymbol{\eta}_t \tag{17}$$

$$\bar{\mathbf{s}}(\mathbf{z}_t) = \mathbf{A}\mathbf{z}_t, \tag{18}$$

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 $\boldsymbol{\eta}_t \sim \mathcal{N}(0, 1)$.[3] 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. 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. 1d), and across all stimulus dimensions (Fig. 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 simulated data, we next compared it to alternative algorithms in the literature, including neural variational

[3] 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 $N_z = 20$.

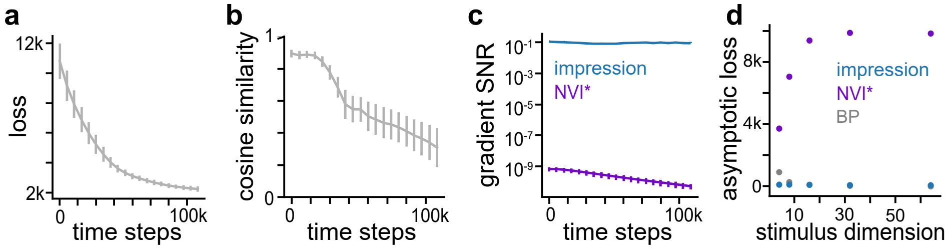

Figure 2: **Comparing learning algorithms and effects of dimensionality. a.** Loss throughout time. **b.** Cosine similarity between gradient updates given by IL and NVI*, averaged over 10^6 samples. **c.** The signal-to-noise ratio for IL (blue), compared to NVI* (purple) across learning. **d.** Asymptotic 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.

inference (NVI*), BP, and WS (see Appendix B 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 [31]. Because NVI* performs poorly for high-dimensional stimuli and large numbers of time steps (Appendix C; [35]), 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, $\Delta\theta_q^{\mathrm{IL}}$, to the inference parameter updates given by NVI*, $\Delta\theta_q^{\mathrm{NVI}}$, for a 4 time-step stimulus sequence (Fig. 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, $\overline{\Delta\theta}_q^{\mathrm{IL}} = \frac{1}{N}\sum_{k=0}^{N} \Delta\theta_q^{\mathrm{IL}}$ and $\overline{\Delta\theta}_q^{\mathrm{NVI}} = \frac{1}{N}\sum_{k=0}^{N} \Delta\theta_q^{\mathrm{NVI}}$ (Fig. 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. $\mathbb{E}[\Delta\theta_q^{\mathrm{NVI}}] = -\frac{d}{d\theta_q}\mathcal{L}$, 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. 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:

$$\mathrm{SNR}(\Delta\theta_q) = \frac{1}{N_\theta}\sum_{i=0}^{N_\theta} \frac{\left(\overline{\Delta\theta}_q^{(i)}\right)^2}{S^2(\Delta\theta_q^{(i)})}, \tag{19}$$

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. 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 C. 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. 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

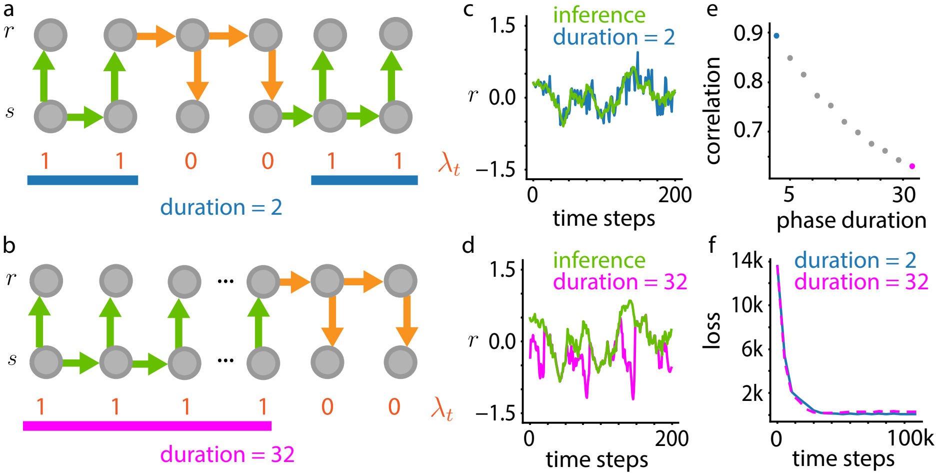

Figure 3: **The effects of phase duration on dynamics and learning. a.** Schematic of IL with a phase duration of 2 **b.** Same as a, but for a switch period of 32. **c.** Comparison of an example neuron's activity through time when the network is in inference mode (green, $\lambda_t = 1$) and when the 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).

IL to be a valid candidate for online learning in the brain, the learning process should 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. 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. 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.3; see Fig. S1a for an even longer phase duration), this implies that the two methods have similar performance. However, IL operating in a mode of fast 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. S1b-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 [36], which provides audio time series of humans speaking digits 0-9.[4] 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-channel standard deviation. The results of Fig. 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. 4a); we provide the details of how this modification affects simulation and parameter updates in Appendix D. 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. 4b). Furthermore, we observed that our

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

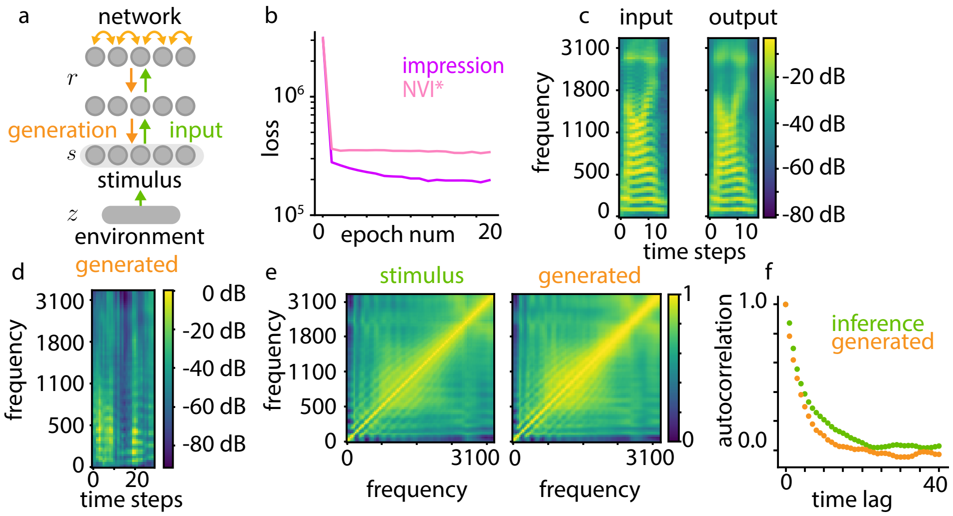

Figure 4: **Learning auditory sequences in a multilayer network. a.** Hierarchical network architecture. **b.** Test loss across epochs for IL (blue) and NVI* (purple). **c.** Comparison between an example data input and the corresponding network output in inference mode ($\lambda_t = 1$). **d.** Sample 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.

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. 4c), and sample stimuli produced while the network is in generative mode ($\lambda_t = 0\ \forall t$) qualitatively correspond to ground-truth stimuli (Fig. 4d), and quantitatively match the structure of both spatial (Fig. 4e) and temporal (Fig. 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.

4 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 [29], 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 signals to the basal synapses dominate network activity.

Computationally, IL outperforms NVI* [30, 31], a particular instance of three-factor plasticity [37], 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) [38] 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 [39, 40]. In the context of supervised learning, several biologically-plausible approximations to BP leverage the apical-basal dendritic structure of pyramidal neurons to learn

[25, 21], based primarily on target-propagation [41] or its variants [42]. 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. 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 [14]. 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 [43] and modeling [44] 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 [32]. However, cortical data indicate that input to apical and basal dendrites contribute concurrently and constructively to spiking activity [45]. 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 [7] 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 [46–49]; 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. There is some evidence that spiking activity driven by apical inputs to pyramidal neurons can induce plasticity at basal synapses [32, 50], 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 [51–54]. Hence, IL has the potential to explain the diversity of plasticity phenomena observed experimentally and inform future experiments.

Acknowledgments and Disclosure of Funding

We thank Camille Rullán Buxó, Caroline Haimerl, Owen Marschall, Pedro Herrero-Vidal, Siavash Golkar, David Lipshutz, Yanis Bahroun, Tiberiu Tesileanu, Eilif Muller, Blake Richards, Guillaume Lajoie, Maximilian Puelma Touzel, and Alexandre Payeur for helpful discussions and feedback on earlier versions of this manuscript. We gratefully acknowledge the Howard Hughes Medical Institute and the Simons Foundation for their support of this work. CS is supported by National Institute of Mental Health Award 1R01MH125571-01, by the National Science Foundation under NSF Award No.1922658 and a Google faculty award.

References

[1] David C Knill and Whitman Richards. *Perception as Bayesian inference*. Cambridge University Press, 1996.

[2] Yair Weiss, Eero P Simoncelli, and Edward H Adelson. Motion illusions as optimal percepts. *Nature neuroscience*, 5(6):598–604, 2002.

[3] Pascal Mamassian, Michael Landy, and Laurence T Maloney. Bayesian modelling of visual perception. *Probabilistic models of the brain*, pages 13–36, 2002.

[4] Daniel Kersten, Pascal Mamassian, and Alan Yuille. Object perception as Bayesian inference. *Annu. Rev. Psychol.*, 55:271–304, 2004.

[5] József Fiser, Pietro Berkes, Gergő Orbán, and Máté Lengyel. Statistically optimal perception and learning: from behavior to neural representations. *Trends in cognitive sciences*, 14(3):119–130, 2010.

[6] Richard D Lange, Sabyasachi Shivkumar, Ankani Chattoraj, and Ralf M Haefner. Bayesian encoding and decoding as distinct perspectives on neural coding. *bioRxiv*, 2020.

[7] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

[8] Deep Ganguli and Eero P Simoncelli. Efficient sensory encoding and Bayesian inference with heterogeneous neural populations. *Neural computation*, 26(10):2103–2134, 2014.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[11] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The Helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

[12] Stefan Habenschuss, Johannes Bill, and Bernhard Nessler. Homeostatic plasticity in Bayesian spiking networks as Expectation Maximization with posterior constraints. *Advances in neural information processing systems*, 25:773–781, 2012.

[13] Johannes Bill, Lars Buesing, Stefan Habenschuss, Bernhard Nessler, Wolfgang Maass, and Robert Legenstein. Distributed Bayesian computation and self-organized learning in sheets of spiking neurons with local lateral inhibition. *PloS one*, 10(8):e0134356, 2015.

[14] Peter Dayan and Geoffrey E Hinton. Varieties of Helmholtz machine. *Neural Networks*, 9(8):1385–1403, 1996.

[15] Anna Kutschireiter, Simone Carlo Surace, Henning Sprekeler, and Jean-Pascal Pfister. Nonlinear Bayesian filtering and learning: a neuronal dynamics for perception. *Scientific reports*, 7(1):1–13, 2017.

[16] David Kappel, Bernhard Nessler, and Wolfgang Maass. STDP installs in winner-take-all circuits an online approximation to hidden markov model learning. *PLoS Comput Biol*, 10(3):e1003511, 2014.

[17] Konrad P Körding and Peter König. Supervised and unsupervised learning with two sites of synaptic integration. *Journal of computational neuroscience*, 11(3):207–215, 2001.

[18] Robert Urbanczik and Walter Senn. Learning by the dendritic prediction of somatic spiking. *Neuron*, 81(3):521–528, 2014.

[19] Mathieu Schiess, Robert Urbanczik, and Walter Senn. Somato-dendritic synaptic plasticity and error-backpropagation in active dendrites. *PLoS computational biology*, 12(2):e1004638, 2016.

[20] Joao Sacramento, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. Dendritic error backpropagation in deep cortical microcircuits. *arXiv preprint arXiv:1801.00062*, 2017.

[21] Jordan Guerguiev, Timothy P Lillicrap, and Blake A Richards. Towards deep learning with segregated dendrites. *ELife*, 6:e22901, 2017.

[22] Blake A Richards and Timothy P Lillicrap. Dendritic solutions to the credit assignment problem. *Current opinion in neurobiology*, 54:28–36, 2019.

[23] Siavash Golkar, David Lipshutz, Yanis Bahroun, Anirvan M Sengupta, and Dmitri B Chklovskii. A biologically plausible neural network for local supervision in cortical microcircuits. *arXiv preprint arXiv:2011.15031*, 2020.

[24] Siavash Golkar, David Lipshutz, Yanis Bahroun, Anirvan Sengupta, and Dmitri Chklovskii. A simple normative network approximates local non-hebbian learning in the cortex. *Advances in Neural Information Processing Systems*, 33, 2020.

[25] Alexandre Payeur, Jordan Guerguiev, Friedemann Zenke, Blake A Richards, and Richard Naud. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature neuroscience*, pages 1–10, 2021.

[26] James M Murray. Local online learning in recurrent networks with random feedback. *ELife*, 8:e43299, 2019.

[27] Owen Marschall, Kyunghyun Cho, and Cristina Savin. A unified framework of online learning algorithms for training recurrent neural networks. *Journal of machine learning research*, 2020.

[28] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[29] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

[30] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

[31] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799. PMLR, 2014.

[32] Katie C Bittner, Christine Grienberger, Sachin P Vaidya, Aaron D Milstein, John J Macklin, Junghyup Suh, Susumu Tonegawa, and Jeffrey C Magee. Conjunctive input processing drives feature selectivity in hippocampal ca1 neurons. *Nature neuroscience*, 18(8):1133–1142, 2015.

[33] Stefano Fusi, Patrick J Drew, and Larry F Abbott. Cascade models of synaptically stored memories. *Neuron*, 45(4):599–611, 2005.

[34] Claudia Clopath, Lorric Ziegler, Eleni Vasilaki, Lars Büsing, and Wulfram Gerstner. Tag-trigger-consolidation: a model of early and late long-term-potentiation and depression. *PLoS computational biology*, 4(12):e1000248, 2008.

[35] Justin Werfel, Xiaohui Xie, and H Sebastian Seung. Learning curves for stochastic gradient descent in linear feedforward networks. In *NIPS*, pages 1197–1204. Citeseer, 2003.

[36] Zohar Jackson, César Souza, Jason Flaks, Yuxin Pan, Hereman Nicolas, and Adhish Thite. Jakobovski/free-spoken-digit-dataset: v1. 0.8, 2018.

[37] Nicolas Frémaux and Wulfram Gerstner. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in neural circuits*, 9:85, 2016.

[38] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[39] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[40] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[41] Yoshua Bengio. How auto-encoders could provide credit assignment in deep networks via target propagation. *arXiv preprint arXiv:1407.7906*, 2014.

[42] Dong-Hyun Lee, Saizheng Zhang, Asja Fischer, and Yoshua Bengio. Difference target propagation. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 498–515. Springer, 2015.

[43] Shannon K Rashid, Victor Pedrosa, Martial A Dufour, Jason J Moore, Spyridon Chavlis, Rodrigo G Delatorre, Panayiota Poirazi, Claudia Clopath, and Jayeeta Basu. The dendritic spatial code: branch-specific place tuning and its experience-dependent decoupling. *bioRxiv*, 2020.

[44] Eren Sezener, Agnieszka Grabska-Barwinska, Dimitar Kostadinov, Maxime Beau, Sanjukta Krishnagopal, David Budden, Marcus Hutter, Joel Veness, Matthew Botvinick, Claudia Clopath, et al. A rapid and efficient learning rule for biological neural circuits. *bioRxiv*, 2021.

[45] Matthew E Larkum, J Julius Zhu, and Bert Sakmann. A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398(6725):338–341, 1999.

[46] Matthew Larkum. A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in neurosciences*, 36(3):141–151, 2013.

[47] Ausra Saudargiene, Stuart Cobb, and Bruce P Graham. A computational study on plasticity during theta cycles at Schaffer collateral synapses on CA1 pyramidal cells in the hippocampus. *Hippocampus*, 25(2):208–218, 2015.

[48] Ines Guerreiro, Zhenling Gu, Jerrel Yakel, and Boris Gutkin. Disinhibitory and neuromodulatory regulation of hippocampal synaptic plasticity. *bioRxiv*, 2020.

[49] Richardson N Leão, Sanja Mikulovic, Katarina E Leão, Hermany Munguba, Henrik Gezelius, Anders Enjin, Kalicharan Patra, Anders Eriksson, Leslie M Loew, Adriano BL Tort, et al. Olm interneurons differentially modulate ca3 and entorhinal inputs to hippocampal CA1 neurons. *Nature neuroscience*, 15(11):1524–1530, 2012.

[50] Katie C Bittner, Aaron D Milstein, Christine Grienberger, Sandro Romani, and Jeffrey C Magee. Behavioral time scale synaptic plasticity underlies ca1 place fields. *Science*, 357(6355):1033–1036, 2017.

[51] Robert C Froemke, Mu-ming Poo, and Yang Dan. Spike-timing-dependent synaptic plasticity depends on dendritic location. *Nature*, 434(7030):221–225, 2005.

[52] Per Jesper Sjöström and Michael Häusser. A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons. *Neuron*, 51(2):227–238, 2006.

[53] Johannes J Letzkus, Björn M Kampa, and Greg J Stuart. Learning rules for spike timing-dependent plasticity depend on dendritic synapse location. *Journal of Neuroscience*, 26(41):10420–10429, 2006.

[54] Robert C Froemke, Johannes J Letzkus, Bjorn Kampa, Giao B Hang, and Greg Stuart. Dendritic synapse location and neocortical spike-timing-dependent plasticity. *Frontiers in synaptic neuroscience*, 2:29, 2010.

Impression learning: Online representation learning with synaptic plasticity
–Appendices–

Colin Bredenberg
Center for Neural Science
New York University
cjb617@nyu.edu

Benjamin S. H. Lyo
Center for Neural Science
New York University
blyo@nyu.edu

Eero P. Simoncelli
Center for Neural Science,
New York University
Flatiron Institute, Simons Foundation
eero.simoncelli@nyu.edu

Cristina Savin
Center for Neural Science,
Center for Data Science
New York University
csavin@nyu.edu

A 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 $\frac{\tilde{p}_\theta}{\tilde{q}_\theta} = 1$:

$$\begin{aligned}\int \left[\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right] (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \; d\mathbf{r}d\mathbf{s} = &\int \left[\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1\right] (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \; d\mathbf{r}d\mathbf{s} \\ &- \frac{1}{2}\int \left[\frac{(\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1)}{1+\epsilon(\mathbf{r},\mathbf{s})}\right]^2 (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \; d\mathbf{r}d\mathbf{s},\end{aligned} \tag{S1}$$

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:

$$\begin{aligned}|\text{bias}| &= \frac{1}{2}\left|\int \left[\frac{(\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1)}{1+\epsilon(\mathbf{r},\mathbf{s})}\right]^2 (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}\right| \\ &\leq \frac{1}{2}\sqrt{\int \left[\frac{(\frac{\tilde{p}_\theta}{\tilde{q}_\theta} - 1)}{1+\epsilon(\mathbf{r},\mathbf{s})}\right]^4 \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}}\sqrt{\int (\nabla_\theta \log \tilde{q}_\theta)^2 \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}},\end{aligned} \tag{S2}$$

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:

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

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

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:

$$
\begin{aligned}
|\text{bias}| &\leq \frac{B}{2}\sqrt{\sum_t \frac{\lambda_t^2}{(\sigma_r^{\text{inf}})^4}\int (\mathbf{r}_t^{(i)} - f(\mathbf{W}\mathbf{s}_t)^{(i)})^2(f'(\mathbf{W}\mathbf{s}_t)^{(i)}\mathbf{s}_t^{(j)})^2\tilde{q}_\theta(\mathbf{r},\mathbf{s}) \; d\mathbf{r}d\mathbf{s}} \\
&= \frac{B}{2}\sqrt{\sum_t \frac{\lambda_t^2}{(\sigma_r^{\text{inf}})^2}\int (f'(\mathbf{W}\mathbf{s}_t)^{(i)}\mathbf{s}_t^{(j)})^2\tilde{q}_\theta(\mathbf{s}) \; d\mathbf{s}} \\
&\leq \frac{B}{2}\sqrt{\frac{S}{(\sigma_r^{\text{inf}})^2}\sum_t \lambda_t^2} \\
&= \frac{B}{2}\sqrt{\frac{ST}{2(\sigma_r^{\text{inf}})^2}}, \qquad \text{(S3)}
\end{aligned}
$$

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. (S1) is $\mathcal{O}(|\frac{\tilde{p}}{\tilde{q}} - 1|)$, 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 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.1 Neural Variational Inference

Neural variational inference is a learning algorithm for neural networks 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:

$$
\begin{aligned}
-\nabla_\theta \mathcal{L} &= -\nabla_\theta \mathbb{E}_{\lambda,\mathbf{z}}\left[\int [\log \tilde{q}_\theta - \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}\right] \\
&= -\mathbb{E}_{\lambda,\mathbf{z}}\left[\int [\nabla_\theta(\log \tilde{q}_\theta - \log \tilde{p}_\theta)]\, \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s} + \int [\log \tilde{q}_\theta - \log \tilde{p}_\theta]\, \nabla_\theta \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}\right] \\
&= -\mathbb{E}_{\lambda,\mathbf{z}}\left[\int [\nabla_\theta \log \tilde{q}_\theta - \nabla_\theta \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s} + \int [\log \tilde{q}_\theta - \log \tilde{p}_\theta]\, (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}\right] \\
&= \mathbb{E}_{\lambda,\mathbf{z}}\left[\int [\nabla_\theta \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r}d\mathbf{s} + \int \left[\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right] (\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta \; d\mathbf{r}d\mathbf{s}\right] \qquad \text{(S4)}
\end{aligned}
$$

Updates calculated by these samples will be unbiased in expectation, because there are no approximations. However, we will show in Appendix C 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 [1, 2]. The first involves subtracting a control variate from our second term:

$$-\nabla_\theta \mathcal{L} = \mathbb{E}_{\lambda,\mathbf{z}}\left[\iint [\nabla_\theta \log \tilde{p}_\theta]\,\tilde{q}_\theta\; d\mathbf{r}d\mathbf{s} + \int \left(\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta} - \mathbb{E}\left[\log \frac{\tilde{p}_\theta}{\tilde{q}_\theta}\right]\right)(\nabla_\theta \log \tilde{q}_\theta)\tilde{q}_\theta\; d\mathbf{r}d\mathbf{s}\right]. \quad \text{(S5)}$$

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. S5 by summing over samples from $\tilde{q}_\theta$, and updating our weights at every time point:

$$\begin{aligned}\Delta\theta &\propto [\nabla_\theta \log \tilde{p}_t(\mathbf{r}_t, \mathbf{s}_t; \theta)] + \left[\log \frac{\tilde{p}_t}{\tilde{q}_t} - \bar{\mathcal{L}}\right] \sum_{s=0}^{t} (\nabla_\theta \log \tilde{q}_t(\mathbf{r}_t, \mathbf{s}_t; \theta)) \\ &\propto [\nabla_\theta \log \tilde{p}_t(\mathbf{r}_t, \mathbf{s}_t; \theta)] + \left[\log \frac{\tilde{p}_t}{\tilde{q}_t} - \bar{\mathcal{L}}\right] g^\theta, \qquad \text{(S6)}\end{aligned}$$

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' [3], is computed by a running integral. These quantities are both computed online as follows:

$$\begin{aligned}\bar{\mathcal{L}}_t &= \gamma_{\mathcal{L}} \log \frac{\tilde{p}_t}{\tilde{q}_t} + (1 - \gamma_{\mathcal{L}})\bar{\mathcal{L}}_{t-1} \qquad \text{(S7)} \\ g_t^\theta &= \nabla_\theta \log \tilde{q}_t(\mathbf{r}_t, \mathbf{s}_t; \theta) + \gamma_g g_{t-1}^\theta, \qquad \text{(S8)}\end{aligned}$$

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 [4, 5], and can be thought of as an analog to temporal windowing in backpropagation through time [6]. 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 [7, 4, 8, 9]. However, the origins of the global 'loss' in our *unsupervised* case are unclear. Furthermore, as we show in Appendix C, 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 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 [10, 11], 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 $\boldsymbol{\eta}_t = \mathbf{r}_t - \bar{\mathbf{r}}^q_t(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})$, where $\bar{\mathbf{r}}^q_t(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})$ is the mean firing rate conditioned on noise, stimulus, and λ values from previous time steps (given by $\tilde{q}$). Similarly, we define $\boldsymbol{\xi}_t = \mathbf{s}_t - \bar{\mathbf{s}}^q_t(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})$. 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 $\bar{\mathbf{r}}^q_t(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1}) + \boldsymbol{\eta}_t$ and $\mathbf{s}_t$ with $\bar{\mathbf{s}}^q_t(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1}) + \boldsymbol{\xi}_t$. We will refer to the mean parameters of $\tilde{p}_\theta$ where these substitutions have been made as $\bar{\mathbf{r}}^p_t(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})$ and $\bar{\mathbf{s}}^q_t(\theta, \lambda, \boldsymbol{\eta}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})$. 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:

$$
\begin{aligned}
-\nabla_\theta \mathcal{L} =& -\nabla_\theta \int [\log \tilde{q}_\theta - \log \tilde{p}_\theta]\, \tilde{q}_\theta \; d\mathbf{r} d\mathbf{s} \\
=& -\nabla_\theta \int \left[\log \prod_t \frac{1}{Z} \exp(\frac{-\boldsymbol{\eta}_t^2}{2(\lambda_t \sigma_s^{\text{inf}} + (1-\lambda_t)\sigma_s^{\text{gen}})^2}) \right] p(\boldsymbol{\eta}, \boldsymbol{\xi}) \; d\boldsymbol{\eta} d\boldsymbol{\xi} \\
& -\nabla_\theta \int \left[\log \prod_t \frac{1}{Z} \exp(\frac{-\boldsymbol{\xi}_t^2}{2(\lambda_t \sigma_s^{\text{inf}} + (1-\lambda_t)\sigma_s^{\text{gen}})^2}) \right] p(\eta, \xi) \; d\boldsymbol{\eta} d\boldsymbol{\xi} \\
& +\nabla_\theta \int \left[\log \prod_t \frac{1}{Z} \exp(\frac{-(\bar{\mathbf{r}}^q_t + \boldsymbol{\eta}_t - \bar{\mathbf{r}}^p_t)^2}{2((1-\lambda_t)\sigma_r^{\text{inf}} + \lambda_t \sigma_r^{\text{gen}})^2}) \right] p(\boldsymbol{\eta}, \boldsymbol{\xi}) \; d\boldsymbol{\eta} d\boldsymbol{\xi} \\
& +\nabla_\theta \int \left[\log \prod_t \frac{1}{Z} \exp(\frac{-(\bar{\mathbf{s}}^q_t + \boldsymbol{\xi}_t - \bar{\mathbf{s}}^p_t)^2}{2((1-\lambda_t)\sigma_s^{\text{inf}} + \lambda_t \sigma_s^{\text{gen}})^2}) \right] p(\boldsymbol{\eta}, \boldsymbol{\xi}) \; d\boldsymbol{\eta} d\boldsymbol{\xi} \\
=& \mathbb{E}_{\boldsymbol{\eta}, \boldsymbol{\xi}} \left[\nabla_\theta \sum_t - \frac{\left(\bar{\mathbf{r}}^q_t(\theta, \boldsymbol{\eta}, \boldsymbol{\xi}) + \boldsymbol{\eta}_t - \bar{\mathbf{r}}^p_t(\theta, \boldsymbol{\eta}, \boldsymbol{\xi})\right)^2}{2((1-\lambda_t)\sigma_r^{\text{inf}} + \lambda_t \sigma_r^{\text{gen}})^2} - \frac{\left(\bar{\mathbf{s}}^q_t(\theta, \boldsymbol{\eta}, \boldsymbol{\xi}) + \boldsymbol{\xi}_t - \bar{\mathbf{s}}^p_t(\theta, \boldsymbol{\eta}, \boldsymbol{\xi})\right)^2}{2((1-\lambda_t)\sigma_s^{\text{inf}} + \lambda_t \sigma_s^{\text{gen}})^2} \right],
\end{aligned}
\tag{S9}
$$

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 $\bar{\mathbf{r}}^q_t$, $\bar{\mathbf{r}}^p_t$, $\bar{\mathbf{s}}^q_t$, and $\bar{\mathbf{s}}^p_t$, 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 [12]. 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 [13]. In subsequent sections, we will show that weight updates computed using samples from this expectation will generally have much lower variance than NVI*.

B.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:

$$-\nabla_\theta \mathcal{L} \approx 2\mathbb{E}_{\lambda_0,\mathbf{z}}\left[\int\left[\sum_t (1-\lambda_t)\nabla_\theta \log q_t + (\lambda_t)\nabla_\theta \log p_{mt}\right]\tilde{q}_\theta \, d\mathbf{r}d\mathbf{s}\right]$$
$$= \mathbb{E}_{\mathbf{z}}\left[\int\left[\sum_t \nabla_\theta \log q_t\right] p_m(\mathbf{r},\mathbf{s}) \, d\mathbf{r}d\mathbf{s} + \int\left[\sum_t \nabla_\theta \log p_{mt}\right] q(\mathbf{r}|\mathbf{s})p(\mathbf{s}|\mathbf{z}) \, d\mathbf{r}d\mathbf{s}\right]. \tag{S10}$$

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 [14]. 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.

C Estimator variance

In Appendix A, 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 [11].

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:

$$\tilde{p}(r|\lambda_t) = \prod_{t=0}^{T} p(r_t)^{\lambda_t} q(r_t)^{(1-\lambda_t)} \tag{S11}$$

$$\tilde{q}(r|\lambda_t) = \prod_{t=0}^{T} p(r_t)^{(1-\lambda_t)} q(r_t)^{\lambda_t}. \tag{S12}$$

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

$$\mathcal{L} = \mathbb{E}_{\lambda_t}\left[KL(\tilde{q}||\tilde{p})\right] = \int\left[\int (\log \tilde{q}(r|\lambda_t) - \log \tilde{p}(r|\lambda_t))\tilde{q}(r|\lambda_t) dr\right] p(\lambda_t) d\lambda_t. \tag{S13}$$

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:

$$\begin{aligned} MSE(\Delta\mu_q) &= \mathbb{E}_{\Delta\mu_q^{(n)}}\left[\left(-\frac{d}{d\mu_q}\mathcal{L} - \frac{1}{N}\sum_{n=0}^{N}\Delta\mu_q^{(n)}\right)^2\right] \\ &= \left(-\frac{d}{d\mu_q}\mathcal{L} - \mathbb{E}_{\Delta\mu_q^{(n)}}\left[\frac{1}{N}\sum_{n=0}^{N}\Delta\mu_q^{(n)}\right]\right)^2 + Var\left[\frac{1}{N}\sum_{n=0}^{N}\Delta\mu_q^{(n)}\right]. \end{aligned} \tag{S14}$$

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:

$$MSE(\Delta\mu_q) = Var\left[\frac{1}{N}\sum_{n=0}^{N}\Delta\mu_q^{(n)}\right] = \frac{Var\left[\Delta\mu_q^{(n)}\right]}{N}. \tag{S15}$$

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:

$$\frac{Var\left[\Delta\mu_q^{(n)}\right]}{N} < \epsilon \;\Rightarrow\; \frac{Var\left[\Delta\mu_q^{(n)}\right]}{\epsilon} < N. \tag{S16}$$

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.

C.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:

$$Var_{\text{IL}} = Var_{\text{BP}} = \frac{T}{\sigma^2} \tag{S17}$$

$$Var_{\text{NVI}} = \frac{T}{2\sigma^2} + \frac{\mathcal{L}}{8\sigma^2}(3T+5) \tag{S18}$$

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.

C.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:

$$\begin{aligned} -\frac{d}{d\mu_q}\mathcal{L} &= \frac{d}{d\mu_q}\int\left[\int(\log\tilde{p}(r|\lambda))\tilde{q}(r|\lambda)dr\right]p(\lambda)d\lambda \\ &= \frac{d}{d\mu_q}\sum_t\left[\int\frac{1}{2}(\log p(r_t))q(r_t)dr_t + \int\frac{1}{2}(\log q(r_t))p(r_t)dr_t\right] \\ &= -\frac{d}{d\mu_q}\sum_t\left[\int\frac{1}{4\sigma^2}((r_t-\mu_p)^2)q(r_t)dr_t + \int\frac{1}{4\sigma^2}((r_t-\mu_q)^2)p(r_t)dr_t\right]. \end{aligned} \tag{S19}$$

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:

$$\begin{aligned}
-\frac{d}{d\mu_q}\mathcal{L} &= -\frac{d}{d\mu_q}\sum_{t=0}^{T}\left[\int \frac{1}{4\sigma^2}((\mu_q + \eta_t - \mu_p)^2)p(\eta_t)d\eta_t + \int \frac{1}{4\sigma^2}((\mu_p + \eta_t - \mu_q)^2)p(\eta_t)d\eta_t\right] \\
&= -\frac{d}{d\mu_q}\sum_{t=0}^{T}\int \frac{1}{2\sigma^2}((\mu_q + \eta_t - \mu_p)^2)p(\eta_t)d\eta_t \\
&= \sum_{t=0}^{T}\int \frac{1}{\sigma^2}(\mu_p + \eta_t - \mu_q)p(\eta_t)d\eta_t. \qquad \text{(S20)}
\end{aligned}$$

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:

$$\begin{aligned}
Var(\Delta\mu_q) &= \int \left(\frac{1}{\sigma^2}(\sum_{t=0}^{T}(\mu_p + \eta_t - \mu_q - (\mu_p - \mu_q)))\right)^2 p(\eta_t)d\eta_t \\
&= \int \sum_{t=0}^{T}\frac{\eta_t^2}{\sigma^4}p(\eta_t)d\eta_t \\
&= \frac{T}{\sigma^2}. \qquad \text{(S21)}
\end{aligned}$$

C.3 Impression learning

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

$$\begin{aligned}
-\frac{d}{d\mu_q}\mathcal{L} &\approx 2\sum_{t=0}^{T}\left[\int\left[(1-\lambda_t)\frac{d}{d\mu_q}\log q(r_t) + (\lambda_t)\frac{d}{d\mu_q}\log p\right]\tilde{q}(r_t|\lambda_t)dr_t\right]p(\lambda_t)d\lambda_t \\
&= 2\sum_{t=0}^{T}\left[\int(1-\lambda_t)\frac{d}{d\mu_q}\log q(r_t)]\tilde{q}(r_t|\lambda)dr_t\right]p(\lambda_t)d\lambda_t \\
&= \sum_{t=0}^{T}\int\frac{d}{d\mu_q}\log q(r_t)p(r_t)dr_t \qquad \text{(S22)}
\end{aligned}$$

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:

$$\begin{aligned}
-\frac{d}{d\mu_q}\mathcal{L} &\approx \sum_{t=0}^{T}\int -\frac{d}{d\mu_q}\frac{1}{2\sigma^2}(r_t - \mu_q)^2 p(r_t)dr_t \\
&= \sum_{t=0}^{T}\int\frac{1}{\sigma^2}(r_t - \mu_q)p(r_t)dr_t. \qquad \text{(S23)}
\end{aligned}$$

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:

$$
\begin{aligned}
Var(\Delta\mu_q) &= \int \left(\frac{1}{\sigma^2}(\sum_{t=0}^{T} r_t - \mu_q - (\mu_p - \mu_q)) \right)^2 p(r_t) dr_t \\
&= \int \frac{1}{\sigma^4}(\sum_{t=0}^{T}(r_t - \mu_p))^2 p(r_t) dr_t \\
&= \int \frac{1}{\sigma^4}\sum_{t=0}^{T}\sum_{t'=0}^{T}(r_t - \mu_p)(r_{t'} - \mu_p) p(r_t) dr_t \\
&= \int \frac{1}{\sigma^4}\sum_{t=0}^{T}(r_t - \mu_p)^2 p(r_t) dr_t \\
&= \frac{T}{\sigma^2},
\end{aligned}
\tag{S24}
$$

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.

C.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. S4), 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:

$$
\begin{aligned}
-\frac{d}{d\mu_q}\mathcal{L} &= \int \left[\int \left(\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) + (\log p - \log q) \sum_{t=0}^{T} \frac{\lambda_t}{\sigma^2}(r_t - \mu_q) \right) \tilde{q}(r|\lambda) dr \right] p(\lambda_t) d\lambda_t \\
&= \mathbb{E}_{r,\lambda} \left[\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) - \left(\sum_{t=0}^{T} (r_t - \mu_p)^2 - (r_t - \mu_q)^2 \right) \sum_{t=0}^{T} \frac{\lambda_t}{2\sigma^4}(r_t - \mu_q) \right] \\
&= \mathbb{E}_{r,\lambda} \left[\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) - \left(\sum_{t=0}^{T} 2r_t(\mu_q - \mu_p) + \mu_p^2 - \mu_q^2 \right) \sum_{t=0}^{T} \frac{\lambda_t}{2\sigma^4}(r_t - \mu_q) \right].
\end{aligned}
\tag{S25}
$$

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:

$$
-\frac{d}{d\mu_q}\mathcal{L} = \mathbb{E}_{r,\lambda} \left[\sum_{t=0}^{T} \frac{(1-\lambda_t)}{\sigma^2}(r_t - \mu_q) - \sum_{t=0}^{T} \sum_{t' \leq t} \frac{\lambda_t}{2\sigma^4} \left(2r_t(\mu_q - \mu_p) + \mu_p^2 - \mu_q^2 \right) (r_t' - \mu_q) \right].
\tag{S26}
$$

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' [1, 2]. 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:

$$-\frac{d}{d\mu_q}\mathcal{L}=\mathbb{E}_{r,\lambda}\left[\sum_{t=0}^{T}\frac{(1-\lambda_t)}{\sigma^2}(r_t-\mu_q)-\sum_{t=0}^{T}\sum_{t'\leq t}\frac{\lambda_t}{\sigma^4}(r_t-\mu_q)(\mu_q-\mu_p)(r'_t-\mu_q)\right]. \quad \text{(S27)}$$

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:

$$\begin{aligned}-\frac{d}{d\mu_q}\mathcal{L}&=\mathbb{E}_{r,\lambda}\left[\sum_{t=0}^{T}\frac{1-\lambda_t}{\sigma^2}(r_t-\mu_q)-\sum_{t=0}^{T}\sum_{t'\leq t}\frac{\lambda_t}{\sigma^4}(r_t-\mu_q)(\mu_q-\mu_p)(r'_t-\mu_q)\right]\\
&=\int\sum_{t=0}^{T}\frac{(1-\lambda_t)}{\sigma^2}(r_t-\mu_q)p(r)dr+\int\frac{1}{2\sigma^4}\sum_{t=0}^{T}\sum_{t'\leq t}(r_t-\mu_q)(\mu_p-\mu_q)(r'_t-\mu_q)q(r)dr\\
&=\frac{T}{2\sigma^2}(\mu_p-\mu_q)+\int\frac{(\mu_p-\mu_q)}{2\sigma^4}\sum_{t=0}^{T}\sum_{t'\leq t}(r_t-\mu_q)\,(r'_t-\mu_q)q(r)dr\\
&=\frac{T}{2\sigma^2}(\mu_p-\mu_q)+\int\frac{(\mu_p-\mu_q)}{2\sigma^4}\sum_{t=0}^{T}\sum_{t'\leq t}(\eta_t)\,(\eta_{t'})p(\eta)d\eta\\
&=\frac{T}{2\sigma^2}(\mu_p-\mu_q)+\int\frac{(\mu_p-\mu_q)}{2\sigma^4}\sum_{t=0}^{T}\eta_t^2p(\eta)d\eta\\
&=\frac{T}{\sigma^2}(\mu_p-\mu_q), \qquad \text{(S28)}\end{aligned}$$

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 $\mathbb{E}[\eta_t]=0$ and $\mathbb{E}[\eta_t\eta_{t'}]=0$. 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:

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

$$\text{(S29)}$$

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:

$$\begin{aligned}
Var(\Delta\mu_q) =& \frac{T}{2\sigma^2} + \frac{1}{2}\int \left(\frac{1}{2\sigma^4} \sum_{t=0}^{T} \sum_{t' \leq t} \left(\eta_t(\mu_p - \mu_q)\right)\left(\eta_{t'}\right) - \frac{T}{\sigma^2}(\mu_p - \mu_q) \right)^2 p(\eta) d\eta \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{2\sigma^4} \int \left(\frac{1}{2\sigma^2} \sum_{t=0}^{T} \sum_{t' \leq t} \eta_t \eta_{t'} - T \right)^2 p(\eta) d\eta \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{2\sigma^4} \mathbb{E}_{\eta_t} \left[\left(\frac{1}{2\sigma^2} \sum_{t=0}^{T} \sum_{t' \leq t} \eta_t \eta_{t'} \right)^2 - \frac{T}{\sigma^2} \left(\sum_{t=0}^{T} \sum_{t' \leq t} \eta_t \eta_{t'} \right) + T^2 \right] \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{2\sigma^4} \mathbb{E}_{\eta_t} \left[\left(\frac{1}{2\sigma^2} \sum_{t=0}^{T} \sum_{t' \leq t} \eta_t \eta_{t'} \right)^2 - \frac{T}{\sigma^2} \left(\sum_{t=0}^{T} \eta_t^2 \right) + T^2 \right] \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{2\sigma^4} \mathbb{E}_{\eta_t} \left[\left(\frac{1}{2\sigma^2} \sum_{t=0}^{T} \sum_{t' \leq t} \eta_t \eta_{t'} \right)^2 \right] \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{8\sigma^8} \mathbb{E}_{\eta_t} \left[\sum_{t=0}^{T} \sum_{t'=0}^{T} \sum_{t'' \leq t} \sum_{t''' \leq t'} \eta_t \eta_{t'} \eta_{t''} \eta_{t'''} \right] \\
=& \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{8\sigma^8} \sum_{t=0}^{T} \sum_{t'=0}^{T} \sum_{t'' \leq t} \sum_{t''' \leq t'} \mathbb{E}_{\eta_t} \left[\eta_t \eta_{t'} \eta_{t''} \eta_{t'''} \right]. \qquad \text{(S30)}
\end{aligned}$$

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:

$$\mathbb{E}_{\eta_t} \left[\eta_t \eta_{t'} \eta_{t''} \eta_{t'''} \right] = \begin{cases} \sigma^4 & \text{if } t = t' \text{ and } t'' = t''' \text{ and } t \neq t'' \\ \sigma^4 & \text{if } t = t'' \text{ and } t' = t''' \text{ and } t \neq t' \\ 3\sigma^4 & \text{if } t = t' = t'' = t''' \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(S31)}$$

These three different conditions result in three different sums:

$$\begin{aligned}
Var(\Delta\mu_q) &= \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{8\sigma^8} \left(\sum_{t=1}^{T} \sum_{t' < t} \sigma^4 + \sum_{t=0}^{T} \sum_{t' \neq t} \sigma^4 + \sum_{t=0}^{T} 3\sigma^4 \right) \\
&= \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{8\sigma^8} \left(\sigma^4 \sum_{t=1}^{T} (t) + T(T-1)\sigma^4 + 3T\sigma^4 \right) \\
&= \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{8\sigma^8} \left(\frac{1}{2} T(T+1)\sigma^4 + T(T-1)\sigma^4 + 3T\sigma^4 \right) \\
&= \frac{T}{2\sigma^2} + \frac{(\mu_p - \mu_q)^2}{16\sigma^4} \left(3T^2 + 5T \right) \\
&= \frac{T}{2\sigma^2} + \frac{\mathcal{L}}{8\sigma^2} \left(3T + 5 \right), \qquad \text{(S32)}
\end{aligned}$$

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

D Multilayer Network Architecture

Here we outline the architecture for the 2-layer network used for processing the Free Spoken Digits dataset [15] in Figure 4.

D.1 Model structure

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

$$\mathbf{s}_t^{\text{inf}} = \mathbf{z}_t + \sigma_s^{\text{inf}}\boldsymbol{\xi}_t \tag{S33}$$

$$\mathbf{r}_t^{\text{inf1}} = f(\mathbf{W}_1\mathbf{s}_t + \mathbf{a}) + \sigma_1^{\text{inf}}\boldsymbol{\eta}_t^1 \tag{S34}$$

$$\mathbf{r}_t^{\text{inf2}} = f(\mathbf{W}_2\mathbf{r}_t^{\text{inf1}}) + \sigma_2^{\text{inf}}\boldsymbol{\eta}_t^2, \tag{S35}$$

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:

$$\mathbf{r}_t^{\text{gen2}} = ((1-k_t)\mathbf{D}_2 + k_t\mathbf{I}))\, r_{t-1} + \sigma_2^{\text{gen}}\boldsymbol{\eta}_t^2 \tag{S36}$$

$$\mathbf{r}_t^{\text{gen1}} = f(\mathbf{D}_1\mathbf{r}_t^{\text{gen2}} + \mathbf{b}) + \sigma_1^{\text{gen}}\boldsymbol{\eta}_t^1 \tag{S37}$$

$$\mathbf{s}_t^{\text{gen}} = f(\mathbf{D}_s\mathbf{r}_t^{\text{gen1}}) + \sigma_s^{\text{gen}}\boldsymbol{\xi}_t, \tag{S38}$$

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 $\tilde{q}_\theta$:

$$\mathbf{r}_t^2 = \lambda_t\mathbf{r}_t^{\text{inf2}} + (1-\lambda_t)\mathbf{r}_t^{\text{gen2}} \tag{S39}$$

$$\mathbf{r}_t^1 = \lambda_t\mathbf{r}_t^{\text{inf1}} + (1-\lambda_t)\mathbf{r}_t^{\text{gen1}} \tag{S40}$$

$$\mathbf{s}_t = \lambda_t\mathbf{s}_t^{\text{inf}} + (1-\lambda_t)\mathbf{s}_t^{\text{gen}}. \tag{S41}$$

D.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:

$$\Delta\mathbf{W}_1^{(ij)} \propto \frac{1-\lambda_t}{(\sigma_1^{\text{inf}})^2}((\mathbf{r}_t^1)^{(i)} - f(\mathbf{W}_1\mathbf{s}_t + \mathbf{a})^{(i)})f'(\mathbf{W}_1\mathbf{s}_t + \mathbf{a})^{(i)}\mathbf{s}_t^{(j)} \tag{S42}$$

$$\Delta\mathbf{a}^{(i)} \propto \frac{1-\lambda_t}{(\sigma_1^{\text{inf}})^2}((\mathbf{r}_t^1)^{(i)} - f(\mathbf{W}_1\mathbf{s}_t + \mathbf{a})^{(i)})f'(\mathbf{W}_1\mathbf{s}_t + \mathbf{a})^{(i)} \tag{S43}$$

$$\Delta\mathbf{W}_2^{(ij)} \propto \frac{1-\lambda_t}{(\sigma_2^{\text{inf}})^2}((\mathbf{r}_t^2)^{(i)} - f(\mathbf{W}_2\mathbf{r}_t^1)^{(i)})f'(\mathbf{W}_2\mathbf{r}_t^1)^{(i)}(\mathbf{r}_t^1)^{(j)} \tag{S44}$$

$$\Delta\mathbf{D}_2^{(ii)} \propto \frac{\lambda_t(1-k_t)}{(\sigma_2^{\text{gen}})^2}((\mathbf{r}_t^2)^{(i)} - (\mathbf{D}_2\mathbf{r}_{t-1}^2)^{(i)})(\mathbf{r}_{t-1}^2)^{(i)} \tag{S45}$$

$$\Delta\mathbf{D}_1^{(ij)} \propto \frac{\lambda_t}{(\sigma_s^{\text{gen}})^2}((\mathbf{r}_t^1)^{(i)} - f(\mathbf{D}_1\mathbf{r}_t^2 + \mathbf{b})^{(i)})f'(\mathbf{D}_1\mathbf{r}_t^2 + \mathbf{b})^{(i)}(\mathbf{r}_t^2)^{(j)} \tag{S46}$$

$$\Delta\mathbf{b}^{(i)} \propto \frac{\lambda_t}{(\sigma_s^{\text{gen}})^2}((\mathbf{r}_t^1)^{(i)} - f(\mathbf{D}_1\mathbf{r}_t^2 + \mathbf{b})^{(i)})f'(\mathbf{D}_1\mathbf{r}_t^2 + \mathbf{b})^{(i)} \tag{S47}$$

$$\Delta\mathbf{D}_s^{(ij)} \propto \frac{\lambda_t}{(\sigma_s^{\text{gen}})^2}(\mathbf{s}_t^i - f(\mathbf{D}_s\mathbf{r}_t^1)^{(i)})f'(\mathbf{D}_s\mathbf{r}_t^1)^{(i)}(\mathbf{r}_t^1)^{(j)}. \tag{S48}$$

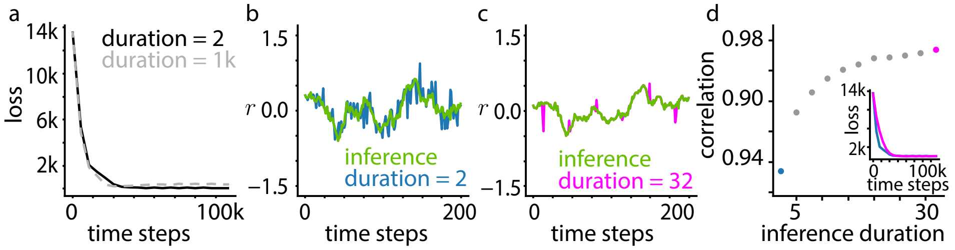

Figure S1: **Additional variations on the phase duration. a.** Comparison of ELBO loss for IL (black) to WS with a 1000 time step phase duration (gray) over training. **b.** Comparison of an example neuron's activity through time when the network is in inference mode (green, $\lambda_t = 1$) and when 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).

References

[1] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

[2] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799. PMLR, 2014.

[3] Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Frontiers in neural circuits*, 12:53, 2018.

[4] Nicolas Frémaux and Wulfram Gerstner. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in neural circuits*, 9:85, 2016.

[5] Thomas Miconi. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife*, 6:e20899, 2017.

[6] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[7] Ila R Fiete, Michale S Fee, and H Sebastian Seung. Model of birdsong learning based on gradient estimation by dynamic perturbation of neural conductances. *Journal of neurophysiology*, 98(4):2038–2057, 2007.

[8] Gregor M Hoerzer, Robert Legenstein, and Wolfgang Maass. Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning. *Cerebral cortex*, 24(3):677–690, 2014.

[9] Colin Bredenberg, Eero Simoncelli, and Cristina Savin. Learning efficient task-dependent representations with synaptic plasticity. *Advances in Neural Information Processing Systems*, 33, 2020.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[11] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[12] Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[14] Shiro Ikeda Shun-ichi Amari and Hiroyuki Nakahara. 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, 1999.

[15] Zohar Jackson, César Souza, Jason Flaks, Yuxin Pan, Hereman Nicolas, and Adhish Thite. Jakobovski/free-spoken-digit-dataset: v1. 0.8, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We provide a derivation of our online learning algorithm (Section 2), and demonstrate its relationships (Appendix B) and relative performance (Fig. 2) to existing algorithms (WS, BP, NVI*). We also demonstrate impression learning's online learning capabilities (Fig. 3) and scalability to naturalistic stimuli (Fig. 4).
   (b) Did you describe the limitations of your work? [Yes] We describe several limitations and possible extensions of our work in the Discussion (Section 4).
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] IL is a theoretical development intended to provide insight into brain function; as such, we do not foresee it having any substantial negative societal impacts. Our hope is that by demonstrating how artificial neural networks of pyramidal neurons can use their dendritic compartments and synaptic plasticity to learn models of their sensory data online, we have come closer to an understanding of how unsupervised learning can occur in early sensory systems in the brain, and will inform experiments attempting to further elucidate the brain's learning processes.).
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Our key theoretical results are described and proved in Section 2.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code used to produce the main experimental results can be found here: `https://github.com/colinbredenberg/Impression-Learning-Camera-Ready`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We used an internal cluster for all simulations.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite the Free Spoken Digits Dataset.
   (b) Did you mention the license of the assets? [Yes] The Free Spoken Digit Dataset uses a Creative Commons Attribution-ShareAlike 4.0 International license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code and the data used for training are linked in the main text.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]