Flexible Phase Dynamics for Bio-Plausible Contrastive Learning

Ezekiel Williams [1 2] **Colin Bredenberg** [1 2] **Guillaume Lajoie** [1 2]

Abstract

Many learning algorithms used as normative models in neuroscience or as candidate approaches for learning on neuromorphic chips learn by contrasting one set of network states with another. These Contrastive Learning (CL) algorithms are traditionally implemented with rigid, *temporally non-local*, and periodic learning dynamics that could limit the range of physical systems capable of harnessing CL. In this study, we build on recent work exploring how CL might be implemented by biological or neurmorphic systems and show that this form of learning can be made temporally local, and can still function even if many of the dynamical requirements of standard training procedures are relaxed. Thanks to a set of general theorems corroborated by numerical experiments across several CL models, our results provide theoretical foundations for the study and development of CL methods for biological and neuromorphic neural networks.

1. Introduction

Contrastive learning (CL) represents a powerful family of algorithms for learning representations, spanning a broad range of machine learning methodologies including generative and discriminative modelling, as well as supervised and unsupervised learning. Broadly, CL can be divided into *equilibrium-based* methods, that use a (stochastic) dynamical process to calculate internal network states, and non-equilibrium methods, that calculate internal states directly in a single pass through the network. CL has been proposed as a normative model for biological learning (Baldi & Pineda, 1991; Illing et al., 2021; Hinton et al., 1984; Scellier & Bengio, 2017; Cao et al., 2020), but requires highly structured training dynamics that restrict the kinds of biological systems that could implement this form of learning. Here, we investigate alternative training dynamics for CL, to better understand the breadth of systems capable of implementing it.

CL learns representations by leveraging the statistical differences between *positive samples* (samples from the training set) and *negative samples* (artificially or internally generated data). We restrict our study to the subclass of CL algorithms with a two-term gradient, where one term depends on positive but not negative samples, and the other term on negative but not positive. This *two-term* CL class comprises a wide range of algorithms, including most algorithms of interest to the neuroscience community. For example, the Boltzmann machine (Ackley et al., 1985), equilibrium propagation (Scellier & Bengio, 2017), and energy based models optimizing the negative log-likelihood (LeCun et al., 2006) all fall within this subclass. Conversely, SimCLR (Chen et al., 2020) is an example of a non-two-term CL method.

Two-term CL algorithms have a periodic, biphasic time course of learning. First, during the *positive phase*, the model calculates internal states given a positive sample. This is followed by a *negative phase* where neural responses to a negative sample are computed. For example, in a Boltzmann machine (Ackley et al., 1985) the positive phase calculates network states conditioned on data while the negative phase calculates internally generated states free from data conditioning; in noise contrastive estimation, the positive phase calculates network states in response to real data while the negative phase calculates responses to noise (Gutmann & Hyvärinen, 2010). Finally, network responses from each phase are used to calculate the gradient and a single step of gradient descent is taken before repeating the whole process for the next learning iteration (Ackley et al., 1985; Scellier & Bengio, 2017; Hinton, 2022) (Fig.1.A). As such, CL suffers from constraints on training dynamics that restrict the set of physical neural networks capable of running the algorithm. In this work, we address four of these constraints and show they can be considerably relaxed:

1. Temporal non-locality: the use of information from past network states that is too old to be available to the network without auxiliary memory mechanisms

[1]Department of Mathematics and Statistics, Universite de Montreal, Quebec, Canada [2]Mila, Quebec AI Institute, Quebec, Canada. Correspondence to: Ezekiel Williams <ezekiel.williams@mila.quebec>, Guillaume Lajoie <g.lajoie@umontreal.ca>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

2. Strict phasic periodicity: oscillation between two distinct phases

3. Rapidly tuneable learning rates: precise modulation of network plasticity during training

4. Deterministic phase length: keeping the same phase length for all phases during training

Contributions: In this paper we address temporal non-locality and periodicity for all two-term CL methods, and investigate how variability of phase length and limited learning rate modulation affect equilibrium-based CL. We propose an importance sampling-inspired approach to estimating the gradient that makes two-term CL methods temporally local and show it is unbiased (see Appendix §A.1). It does so by stochastically selecting either a single positive or single negative sample to learn from at each gradient step, thus also eliminating the strict periodicity seen in past algorithms. The proposed approach provides a principled way to tune the amount of time spent taking gradient steps with respect to positive versus negative samples. Interestingly, we find that it is not always optimal to spend an equal amount of training time learning from positive compared to negative samples. Next, we demonstrate quite generally that equilibrium-based methods, and thus equilibrium-based CL, can still learn effectively–with asymptotically unbiased gradient estimates–even with no learning rate modulation (i.e. avoiding the wait until the end of phases to update parameters) and with a significant amount of noise in the length of each phase. Taken together, our results formally relax the previous requirements on the learning dynamics of many CL algorithms, thereby demonstrating that a broader set of systems might be capable of implementing this form of learning.

2. Background

Bio-Plausible Learning: How the brain learns is a fundamental scientific question driving research not only in neuroscience but also in AI. In particular, AI algorithms are being used more and more as *normative models* of learning in biological neural networks, a line of study that has already yielded exciting results (Richards et al., 2019; Sorscher et al., 2022; Hennequin et al., 2018; Payeur et al., 2021). However, many AI algorithms exhibit properties that go beyond gross abstractions of neural processing to be computationally incompatible with neural hardware. This means that an important component of AI-inspired modelling in neuroscience is teasing out which aspects of a candidate learning model cannot be implemented by the brain and whether the model might be modified so as to make it more plausibly implementable by biological circuits (*bio-plausible*). For example, backpropagation computes using biologically implausible mechanisms such as *spatial non-locality* (Diederich & Opper, 1987) and *weight-transport* (Lillicrap et al., 2016).

The study of bio-plausible learning is not solely applicable to neuroscience but also has relevance to AI in itself, as algorithms that function like the brain often have computational benefits (Lake et al., 2017). An especially important example of this synergy between bio-plausible learning and AI development is neuromorphic computing: neuromorphic hardware represents a potential solution to AI energy efficiency problems (Strubell et al., 2020), but often puts similar constraints on algorithms that biological systems do, for example utilizing local storage of memory (Schuman et al., 2022; Frenkel et al., 2021).

Contrastive Learning and Bio-Plausibility: Often the positive and negative phases of CL are thought of, from a psychological modelling perspective, as being implemented during wakeful processing of stimuli and periods of sleep, respectively (Hinton, 2022; Crick & Mitchison, 1983), just as in the Wake-Sleep algorithm (another classic learning model) (Dayan et al., 1995). Other studies have related these phases to oscillations in cortical circuits (Baldi & Pineda, 1991) or have suggested that the difference between phases could be delineated by saccades in the visual system (Illing et al., 2021). Given a lack of hard evidence in favour of these hypotheses, we will refer to these phases solely as positive and negative in what follows.

A subset of CL algorithms do not require backpropagation for neural network-based learning (Ackley et al., 1985; Scellier & Bengio, 2017; Hinton, 2022); it is this feature in particular that has brought attention to CL methods as bio-plausible learning models. However, the learning dynamics for CL methods exhibit properties that have not been observed in the brain.

To begin with, the majority of CL methods are *temporally non-local*. This refers to when an algorithm uses information that is not available within some short distance of the present timestep to update the parameters of the network at the current timestep (Illing et al., 2021). For CL methods, temporal non-locality appears because network responses to positive *and* negative samples must be calculated before performing a parameter update (e.g. Fig.1.A). Notably, this also introduces a precise positive-negative phase period during the learning time course. The problem with temporal non-locality is that it would require a neural network to save its neuron states in a memory buffer for a non-negligible amount of time before recalling them. Memory storage could be solved in theory by synaptic weight consolidation (introduce a candidate modification in phase 1, and consolidate the change in phase 2), but there's no known relationship between this phenomenon and biphasic network dynamics in the brain.

Equilibrium-Based Contrastive Learning: Beyond the problems of temporal non-locality and periodicity, two additional complications exist for *equilibrium-based CL* methods used to train (stochastic) dynamical system neural networks. For each data point observed during training, equilibrium-based CL must run internal dynamics on the space of the given model's hidden unit activations until the dynamics have converged to the vicinity of a fixed point. This means that, for each positive and negative sample pair, learning rates must be set to zero and dynamics run two times before learning rates are turned-on for a single parameter update (Ackley et al., 1985; Scellier & Bengio, 2017; Hinton, 2022) (Fig.1.A). Such a specific pattern of learning rate scheduling and periodic stimulus/sample presentation is as yet unsupported empirically in biological systems: this begs the question whether aspects of these training dynamics can be relaxed. Lastly, because of the convergence requirements on the dynamics of equilibrium-based learning, when training AI methods one usually fixes a particular length of time to run dynamics for each sample. Comparatively, regardless of the proposed implementation of biphasic learning (sleep/wake, neural oscillations, saccades, active/quiescent) animals usually spend a variable amount of time in each state. This motivates the question of how much variability equilibrium-based CL algorithms might tolerate in the length of phases.

Example CL system—The Boltzmann Machine: To ground this discussion of biphasic learning, we will provide a brief example of a classic equilibrium-based CL method: the Boltzmann machine (Ackley et al., 1985). The Boltzmann machine is a generative Recurrent Neural Network (RNN) model that learns a probability distribution over a d dimensional binary support from a dataset $\{x^{(i)}\}_{i=1}^{N}$, $x^{(i)} \in \{0,1\}^d$. The RNN units $z \in \{0,1\}^n$ are partitioned into visible units, v_j $j \in \{0, \dots d\}$, whose learned activity represents samples from the d dimensional support, and hidden units h_i, $i \in \{0, \dots, n_h\}$, that encode higher order interactions between the visible units. Then $z \in \{0,1\}^{d+n_h}$ with $z = [v, h]$. The units are connected via a synaptic weight matrix $\theta \in \mathbb{R}^{n \times n}$ that learns to encode the correlation structure between units via the following contrastive Hebbian learning rule:

$$\Delta\theta_{kl} = \eta z_{k,+} z_{l,+} - \eta z_{k,-} z_{l,-}, \tag{1}$$

where the '$+$' subscript denotes an activation generated from $K \in \mathbb{N}$ steps of dynamics with the visible units clamped to a data point (a positive phase) and the '$-$' subscript denotes an activation generated from K steps of recurrent dynamics with the visible units unclamped (a negative phase). The weight update finally occurs after both phases have been run and requires the temporally non-local saving of activations from the first phase. Learning rates are set to zero while the phase dynamics are run and only turned on for the weight update. These periodic learning dynamics are referred to as K-step Contrastive Divergence (CDK) (Hinton, 2002), and are visualized in Fig.1.A. While CDK is not the only algorithm for training a Boltzmann machine, it is typical of the training dynamics used by other methods (Fischer & Igel, 2014).

Related Work: Here, we non-exhaustively review approaches to learning that are equilibrium-based but not contrastive, then contrastive non-equilibrium methods, and finally the intersection of these.

The majority of bio-plausbile, non-contrastive equilibrium-based methods find their roots in Predictive Coding (PC) (Srinivasan et al., 1982; Rao & Ballard, 1999), which was retrospectively shown to be a variational method (Bishop & Nasrabadi, 2006) for learning a hierarchical Gaussian model (Friston, 2005). Given the prescribed generative model, these works are more narrow in scope than ours, which explores equilibrium learning quite broadly in the context of stochastic gradient descent and fixed point latent dynamics.

There are two non-equilibrium-based CL classes of particular relevance to this study, both inspired by noise contrastive estimation (Gutmann & Hyvärinen, 2010). The first class originates from Contrastive Predictive Coding (CPC) (Oord et al., 2018), which contrasts positive samples of the future conditioned on the present with negative samples taken at random from the analyzed time series. A bio-plausible version of CPC (Illing et al., 2021) uses the same approach taken in this study to arrive at temporally local and aperiodic learning, but focuses on the specific case of CPC and does not make the link with unbiased gradients and importance sampling. The second class encompasses the Forward-Forward algorithm, and related works, that do binary classification of positive from negative samples at each layer to achieve learning without backpropagation (Hinton, 2022; Ororbia & Mali, 2023). To our knowledge, temporally local and aperiodic training of these algorithms has not been studied.

Equilibrium-based CL was first researched in the late 80s and early 90s, with the Boltzmann machine (Ackley et al., 1985) and its deterministic variants (Baldi & Pineda, 1991; Movellan, 1991). Baldi & Pineda (Baldi & Pineda, 1991) explored temporally local learning in this context and, as in our work, achieve temporal locality by breaking up the two-term gradient into single positive and negative-phase terms. Unlike our work, they consider solely periodic updating—associating the learning process with cortical oscillations—and explain that learning follows gradients only in the small-learning rate limit. Temporally local learning was investigated more recently with two modifications

to Equilibrium Propagation (EP) (Scellier & Bengio, 2017; Ernoult et al., 2020; Laborieux & Zenke, 2022). In the first (Ernoult et al., 2020), the proposed solution still requires two network passes and, like Baldi & Pineda, relies on an infinitesimally small learning rate. This is necessary because Ernoult et al.'s solution to temporal non-locality uses continual weight updates during the positive phase. We use a similar argument, but to achieve concurrent learning and latent dynamics, rather than locality. The second study (Laborieux & Zenke, 2022) uses oscillatory clamping for training, similar to the oscillatory teacher signal used historically (Baldi & Pineda, 1991). It does so by leveraging activity in the complex plane, which provides an elegant solution for phase management but requires complex neuron activiations. This complicates the mapping to standard neuroscience and AI models which usually represent neural activity in real spaces. The second study shares another similarity with ours, in that it uses a separation of timescales argument to achieve concurrent learning and latent-space dynamics. However, the separation of timescales is left as an assumption, in contrast to our derivation of error bounds that explicitly account for the two timescales.

3. Theoretical Results

We first describe our results on generalizing CL to temporally local learning dynamics, followed by results on learning rate and phase length in equilibrium-based CL systems. We begin by considering the set of CL methods that employ Stochastic Gradient Descent (SGD) to optimize a two-term objective function such that a single SGD step uses the following gradient estimate:

$$g(\theta) = g_+(\theta, z_+) - g_-(\theta, z_-). \tag{2}$$

Here, $z_+ \in \mathcal{Z}_+$, where $\mathcal{Z}_+$ is the set of positive-phase network states, and $z_- \in \mathcal{Z}_-$, where $\mathcal{Z}_-$ is the set of negative-phase states, θ is the parameters, and g_+ and g_- are the functions that map parameters and network states to positive and negative phase gradient terms. Temporal locality and periodicity are inherent in this gradient estimate, as it requires two different sets of network states to compute: separate passes through the neural network must be used to estimate each term of the gradient. We remark that, in a similar vein, batch learning is temporally non-local, as it requires a running sum of network weights to be held in memory during each learning phase. As such, in what follows, we study SGD (batch size of 1). Many classic CL algorithms seen in the machine learning and computational neuroscience literature have this form of gradient, including contrastive Hebbian learning (Movellan, 1991), equilibrium propagation (Scellier & Bengio, 2017), maximum likelihood learning of energy based models (LeCun et al., 2006), and forms of noise contrastive estimation (Gutmann & Hyvärinen, 2010). The first contribution of this work is to construct a temporally local estimate of the gradient in Equation 2.

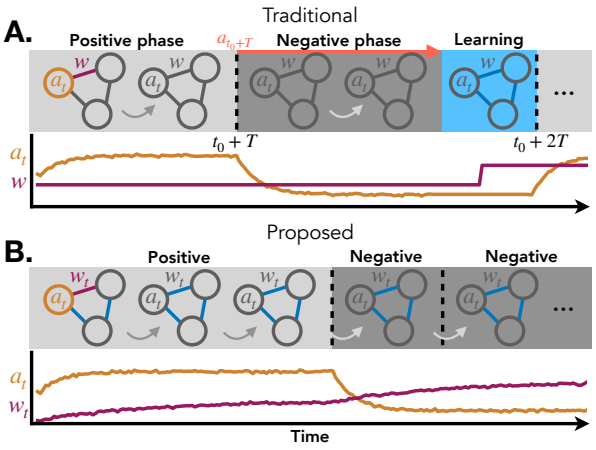

Figure 1. Comparison of learning dynamics for two algorithms. Equilibrium-based CL dynamics are shown, but the pattern of phases and temporal (non-)locality generalizes to non-equilibrium CL methods. **A.** Traditional CL using the gradient given by Equation 2. Network states, a, at the end of a positive phase (light grey) must be transported through the negative phase (dark grey) to the learning step (red arrow), introducing temporal non-locality. Learning (blue) only occurs after both phases have been completed, and phase lengths, T, are fixed. The plotted trajectories are temporal dynamics corresponding to the activation (orange) and network weight (purple) highlighted in the first frame of the diagram. Grey arrows denote dynamics in activation space. **B.** Equilibrium-based CL using the gradient given by Equation 3, and constant learning rate with stochastic phase lengths. It is temporally local because no network states must be saved in memory to enable learning, and the phase type is chosen randomly at each phase, eliminating periodicity (e.g. two negative phases in a row). There is no highlighted learning step because learning is always on, denoted by the blue weights and apparent in the constantly evolving purple weight dynamics in the bottom plot. Phase lengths are stochastic (e.g. the positive phase is 3x as long as the first negative phase). Data in the two plots is fabricated for the purpose of illustration.

3.1. Gradient Sampling to Avoid Non-Locality

We can begin to resolve the temporal non-locality of Equation 2 by considering the situation where each term g_+ and g_- are used separately to do parameter updates. This means that at the end of a positive phase g_+ is used for an update, and g_- is used similarly for the negative phase, avoiding the need for a "memory buffer" to combine these terms. This might not be a good idea in the case of periodic phases, but it turns out that introducing stochasticity into the phase distribution proves very useful. We therefore propose a

probabilistic phase selection process, where the next phase type (positive or negative) to be performed during learning is chosen by a Bernouilli random variable B. In this setting, the sum in Equation 2 can be replaced by the following

$$\hat{g}(\theta) = \frac{B}{b} g_+(\theta, z_+) - \frac{1-B}{1-b} g_-(\theta, z_-), \tag{3}$$

where $B \sim \text{Bernoulli}(b)$ so that b is the probability of selecting a positive phase for the current SGD step. A realization of such a single, stochastically sampled term becomes a form of importance sampling (Robert et al., 1999) that estimates the true gradient from Equation 2. We refer to this gradient estimator as ISD (Importance Sampling-Discrete, given the discrete nature of the Bernoulli variable) which is subject to the following proposition (see Appendix §A.1 for proof).

Proposition 3.1. *For $b \in (0, 1)$, the ISD gradient estimate of Equation 3 is an unbiased estimate of the gradient given in Equation 2.*

This implies that the mean of $\hat{g}(\theta)$ is still the same as that of the original gradient. Furthermore, Equation (3) has the advantage of allowing flexibility in the frequency of each CL phase without sacrificing the unbiased quality. Indeed, in the case where the gradient is that of a scalar-valued loss or energy function, the guaranties of convergence to a fixed point of the loss that SGD relies on still apply (Shamir & Zhang, 2013; Davis & Drusvyatskiy, 2019; Ghadimi et al., 2016).

Crucially, we note that this gradient estimator remains unbiased for any $b \in (0, 1)$. However, not surprisingly, this new gradient will have higher variance in many cases, resulting in noisier estimates. The following lemmas provide some details about this variance (see Appendix §A.1 for proofs).

Lemma 3.2. *Assume $g_+(\theta, z_+)$ and $g_-(\theta, z_-)$ are statistically independent and the inner product of their means is non-negative. If $b \in (0, 1)$, the ISD gradient estimate of Equation 3 has strictly greater variance (quantified by the trace of the covariance of the gradient vector) than the estimate given by Equation 2.*

Furthermore, the variance, quantified by the trace of the covariance of the gradients, depends non-trivially on the parameter b, making the parameter a target for optimization. To this end:

Lemma 3.3. *The variance of the gradient estimator given by Equation 3 is a convex function of b.*

Notably, this minimum can be determined analytically, given estimates of the first and second moments of $g_+(\theta, z_+)$, $g_-(\theta, z_-)$. We explore these properties empirically in numerical experiments outlined in §4, and find that (i) networks can still learn effectively with this higher-variance estimator, and (ii) that analytically derived estimator variance effects outlined in the Lemmas 3.2 and 3.3 appear to predict observed variance.

Finally, we highlight that the variance, quantified by the trace of the covariance matrix, of the ISD estimator scales linearly w.r.t. the covariances and the square magnitudes of the means of the gradients of the two gradient terms.

3.2. Bias of Always-On-Learning and Variable Phase Length in Equilibrium-Based CL

The second contribution of this study is to investigate less restrictive learning dynamics for equilibrium-based CL. We drop the subscripts $+$ or $-$ on the network states, z, when the equations discussed apply to either phase. Assume that learning occurs via gradient estimate $\hat{g}$. Applying Equation 3 to a standard equilibrium-based CL paradigm would require the following learning dynamics:

$$\begin{aligned} b_{t+1} &= \mathbb{1}_{t \neq kT}\, b_t + \mathbb{1}_{t=kT} B \\ x_{t+1} &= \mathbb{1}_{t \neq kT}\, x_t + \mathbb{1}_{t=kT} X(b_t) \\ z_{t+1} &\sim p(z_{t+1} | \theta_t, z_t, x_{t+1}, b_{t+1}) \\ \theta_{t+1} &= \theta_t - \eta(t) \hat{g}(\theta_t, z_t, b_{t+1}), \end{aligned} \tag{4}$$

where p is a Markov transition kernel, B is sampled according to its role in Equation 3, and X is sampled randomly from the set of positive samples if $B = 1$ or negative samples if $B = 0$. To indicate that parameter updates only take place at the end of a phase, we set $\eta(t) = \eta$ if $t = kT$ and set $\eta(t) = 0$ otherwise. Here, T represents phase length and T and k are both in $\mathbb{N}$. While these dynamics are temporally local in the sense that it is no longer necessary to hold distinct positive and negative gradient terms in memory, they still require precise manipulation of η and use a fixed phase length. That is to say, we still require the explicit identification of the end of a phase to toggle a parameter update. We now relax these requirements by investigating two novel situations: one where the learning rate is fixed to $\eta(t) = \frac{\eta}{\tau} \forall t$, and another where the learning rate is fixed to the same value *and* T is chosen randomly (τ is the phase length or the mean length respectively). We label these two situations *Always-On-Learning* (AoL) and *AoL Random T* respectively. The learning dynamics of AoL Random T results in Equation 4 being updated as follows:

$$\begin{aligned} b_{t+1} &= \mathbb{1}_{\omega_t = 0} b_t + \mathbb{1}_{\omega_t = 1} B \\ x_{t+1} &= \mathbb{1}_{\omega_t = 0} x_t + \mathbb{1}_{\omega_t = 1} X(b_t) \\ z_{t+1} &\sim p(z_{t+1} | \theta_t, z_t, x_{t+1}, b_{t+1}) \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\tau} \hat{g}(\theta_t, z_t, b_{t+1}), \end{aligned} \tag{5}$$

where $\omega_t \sim \text{Bernoulli}(\mu)\ \forall t$ and $\mu << 1$. We remark that this form of continuous updating is only possible with the

ISD gradient estimator (Equation 3); the standard estimator (Equation 2) requires both phases to have been run before the next gradient estimate can be calculated.

As with the gradient estimates presented earlier, we would like to show that the learning dynamics with constant learning rate and variable phase length will, at least within some error bound, follow unbiased gradient steps. The following theorem provides a bound on the bias introduced by training with constant learning rate and stochastic phase lengths. We prove and comment on the assumptions of the theorem in §B of the Appendix.

Theorem 3.4. *Consider a single phase of equilibrium-based learning where (i) an update is performed at every time-step of the network dynamics with learning rate $\frac{\eta}{\tau}$, and (ii) the phase is either fixed to value τ or ends stochastically at each step with small probability $\frac{1}{\tau}$. Assume that equilibrium dynamics evolve for fixed θ according to the Markov transition kernel $p(z_t|z_{t-1}, \theta)$, and that these equilibrium dynamics weakly converge to a stationary distribution at a rate at least as fast as $\mathcal{O}(\frac{1}{t^2})$. Then with mild regularity conditions on $\hat{g}(\theta)$, and integrability conditions on the transition kernel, the bias of the learning updates are of the following order*

$$\mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}(\eta^2 \tau), \tag{6}$$

where $s = \tau^m$ for $0 < m < 1$.

Remark: For asymptotic unbiasedness, and for learning to occur, we need the error to go to zero with a rate faster than the learning rate, $\mathcal{O}(\eta)$. If we set $\tau = \frac{1}{\sqrt{\eta}} = s^2$, then the error rate is $\mathcal{O}(\eta^{\frac{5}{4}})$, satisfying this constraint.

If one assumes that the equilibrium dynamics follow a deterministic dynamical system, $z_{t+1} = F(z_t, \theta_t)$, instead of stochastic Markov dynamics, one can prove the following theorem, which relies on slightly milder assumptions and yields an improved convergence rate (see Appendix §C for proof):

Theorem 3.5. *Assume the same equilibrium learning paradigm of Theorem 3.4, but with dynamics that evolve for fixed θ according to the homogeneous discrete time dynamical system given by the map $F(z, \theta)$. Assume that $\hat{g}(\theta)$ is differentiable and bounded with bounded partial derivatives for all network state values, and that F is differentiable w.r.t. z and θ, also with bounded partial derivatives. Finally, assume that the equilibrium dynamics converge to a fixed point at a rate at least as fast as $\mathcal{O}(\frac{1}{t^2})$. Then the bias of the learning updates are of the following order:*

$$\mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}(\eta^2), \tag{7}$$

where $s = \tau^m$ for $0 < m < 1$.

Importantly, both Theorems 3.4 and 3.5 apply not only to equilibrium-based CL methods, but to *any method* satisfying the theorem assumptions that uses equilibrium-based gradient updates.

Taken together, these results define (mild) conditions for which CL can still converge if parameters are updated at each time step of network dynamics, both with a set phase length and with phases having variable length and switching times. In the next section, we illustrate how these relaxed CL conditions maintain successful learning at a limited cost compared to standard CL.

4. Experimental Findings

While the unbiased (and asymptotically unbiased, in the case of equilibrium CL) property of the proposed algorithms provides theoretical evidence for their utility, we wished to test whether the algorithms would work in practice. We empirically evaluated the proposed algorithms by using them to modify two different CL methods: maximum likelihood learning in energy based models (LeCun et al., 2006) and the recently proposed, noise contrastive estimation-inspired (Gutmann & Hyvärinen, 2010) Forward-Forward (FF) algorithm (Hinton, 2022). For the former paradigm we trained a Restricted Boltzmann Machine (RBM) (Fischer & Igel, 2014), which we selected given its history as a classic model in the computational neuroscience community. For the former, we trained a feed-forward neural network with two hidden layers. We note that the RBM is an equilibrium-based generative model, while the FF algorithm is non-equilbrium-based and is used to train a discriminative model. Testing was performed on the binarized MNIST (bMNIST) and the Bars And Stripes (BAS) (Fischer & Igel, 2014) datasets for the RBM, and MNIST for the FF-trained network. We chose these datasets because the log-likelihood is tractable for BAS, and because both datasets have been used to benchmark related models. Because of the issues of temporal non-locality inherent in batch learning we use batch sizes of 1 throughout. For simplicity, we tested only vanilla SGD (Robbins & Monro, 1951), without averaging, momentum, or adaptation, for the RBM; we used ADAM (Kingma & Ba, 2014) to train the forward-forward network. We stress that we are not comparing performance of FF and RBM; rather, we are using them as two different tests of the bio-plausible algorithms proposed in this paper. Implementation details can be found in appendix E.

4.1. Estimators Learn Comparably to Classic Methods

We first compared ISD with the gradient estimator given by Equation 3 to the traditional two-term CL estimator (Equation 2) for training a feed-forward neural network (Fig.2.A,C) and a RBM (Fig.2.B,D). In all cases, Fig.2 shows the loss v.s. training time, measured either over gra-

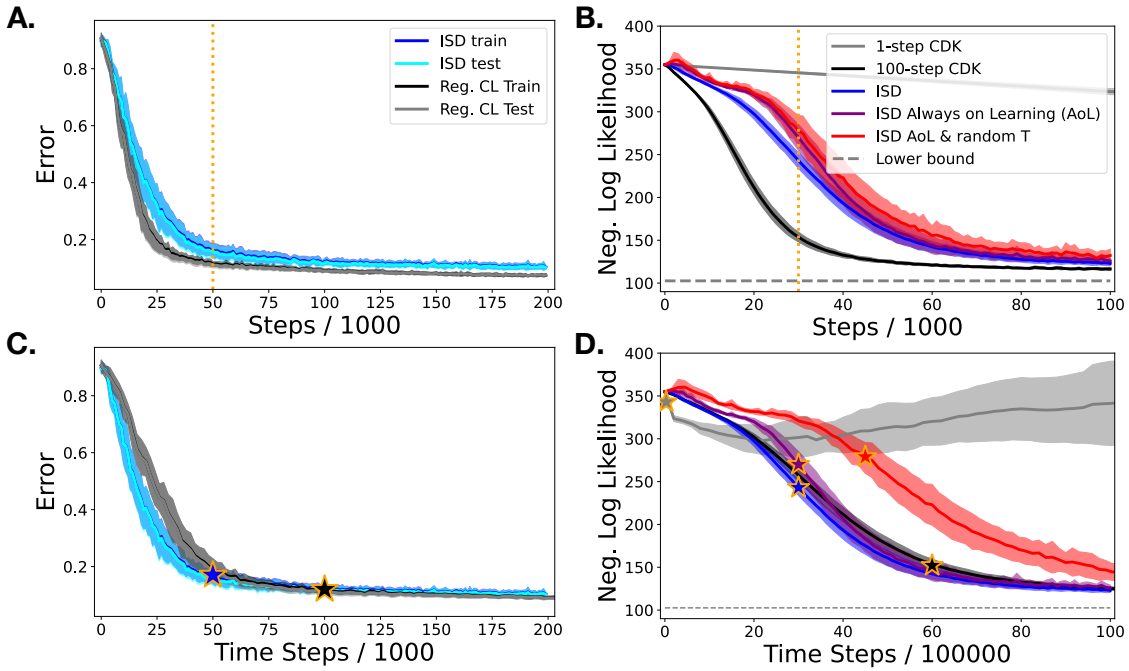

Figure 2. Comparison of ISD, AoL ISD, and AoL random T ISD variants (ours) with standard temporally non-local methods. **A.** CL on MNIST: comparison of test and train data learning curves for the temporally non-local Forward-Forward algorithm versus the ISD version. Y-axis is mean classification error; x-axis is gradient steps divided by 1000. **B.** Equilibrium-based CL on BAS: comparison of 1-step and 100-step CDK algorithm to ISD, AoL ISD, and AoL random T ISD for training an RBM on the BAS dataset. Y-axis is negative log-likelihood; x-axis is gradient steps (phase counts) divided by 1000 for CDK and regular ISD (ISD AoL and ISD AoL Random T). **C. & D.** Same plots as A. and B. but with the x-axis rescaled to show the number of 'time' steps used in network dynamics for each algorithm, rather than number of gradient steps or phases (recall that certain algorithms require more or fewer time steps for a single phase). Dotted lines in A. and B. show a step count that corresponds to the number of steps marked by stars in C. and D for each method. In all cases, ISD methods perform only slightly worse than the temporally non-local, learning rate modulated and fixed phase CDK methods, but with greater variance and some reduction in learning speed over phase counts. This speed distinction shrinks when measuring learning over network dynamics time steps rather than gradient steps (phase counts). All plotted quantities are means ± 1 standard deviation. For ISD AoL random T the x-axis of D is mean time steps.

dient steps—or phase counts for the two AoL algorithm versions—(top row) or number of time steps in network dynamics (bottom row). We highlight the importance of contrasting these distinct temporal units, as standard CL updates parameters only at the end of positive/negative phase pairs, each containing several network dynamics iterates (time-steps). In contrast, ISD performs updates more often (either at the end of each phase or at each time step), and the ISD AoL random T version also has a variable phase length. As such, comparing learning curves based on the number of network iterates—the main compute requirement for biological systems—reveals how close ISD methods are to standard CL.

As expected, ISD exhibited higher variance learning curves—a result of using a higher variance gradient estimate—but was still able to learn quite robustly and to a degree of accuracy only slightly worse than standard two-term gradient estimates. In the case of equilibrium learning, the extra bias introduced by the constant learning paradigm led, again, to a further slight reduction in training accuracy and higher noise, which was further accentuated by introducing random phase lengths. While we did not perform an extensive hyper-parameter optimization, we noted that the ISD estimators tended to require a slightly lower learning rate, and converged slightly slower than the two-term estimators.

4.2. Robustness to Changes in Positive Phase Proportion

Next, we wished to explore the effect of parameter b, the proportion of time spent in positive phases during training, for algorithms making use of the gradient in Equation 3. We investigated this property for the RBM model (Fig.3.A). Interestingly, we found that learning is robust to changes in b, and that the optimal value is not always for equal time spent in positive versus negative phases. In fact, we saw better results for a higher b on the bMNIST dataset (see §D

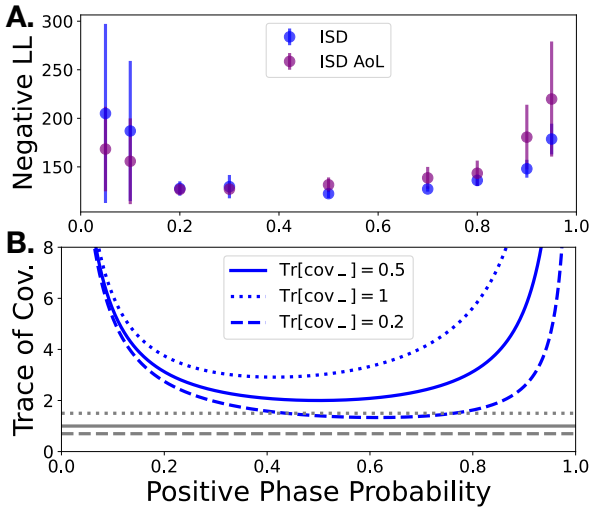

Figure 3. Algorithm performance as function of positive phase probability, b. Performance is robust to changes in this parameter, and the optimal value is not always at $b = 0.5$. We expect this relationship to be due to the variance of the estimator. **A.** Performance on BARS dataset for RBM trained using ISD and ISD with constant learning rate. Error bars are ± 1 standard deviation. **B.** Gradient estimator covariance matrix trace as function of positive-phase probability, calculated analytically. Different scenarios depict distinct values of the negative phase's gradient estimator covariance trace, holding the positive phase covariance trace constant at 0.5; e.g. solid line is for $\mathrm{TR}\big[\mathrm{cov}\big(g_+(\theta, z_+)\big)\big] = \mathrm{Tr}\big[\mathrm{cov}\big(g_-(\theta, z_-)\big)\big] = 0.5$. These quantities are a measure of variance of the estimators. Blue lines are the traces of the covariance matrix of the gradient estimate in Equation 3, grey are for that of Equation 2. To simplify the plot we have assumed that the means of $g_+(\theta, z_+)$ and $g_-(\theta, z_-)$ are both the zero vector.

of the Appendix), and better results for lower values of b on the BAS dataset. Robustness to parameter changes was substantial, with values as high as 0.8 and as low as 0.2 not performing drastically worse than the best-performing b.

The mean of the ISD estimator is the same regardless of the ratio of positive to negative phases, so ISD will remain an unbiased estimator as b changes. Conversely, the *variance* of the ISD estimator depends on b, a phenomenon that we conjecture to underlie the observed robustness of learning to changes in this parameter. High variance detrimentally affects learning by requiring lower learning rates for an accurate gradient estimate—essentially averaging over more learning steps—and, in absence of this, can result in instability. We investigate ISD estimator variance in the next section.

4.3. Variance of the Gradient Estimate

To further interrogate the effect of b on learning, we explored how b affects the variance of the ISD gradient estimator (see Fig.3.B). The gradient estimate is generally a multivariate random variable and thus one uses a covariance matrix to quantify variance. To generalize the variance to higher dimensions we consider the expected Euclidean distance of the random variable from the mean, which is the trace of the covariance matrix and is equal to the variance in one dimension. Given the means and covariances of $g_+(\theta, z_+)$ and $g_-(\theta, z_-)$, one can derive the trace of the covariance matrix of $\hat{g}(\theta)$ as a function of b (see §A.1). It is this function that is plotted for different assumed values of the means and traces of the covariances of $g_+(\theta, z_+)$ and $g_-(\theta, z_-)$ in Fig.3.B. We observe that higher (lower) values of b lead to less variance when $g_+(\theta, z_+)$ has lower (higher) variance than $g_-(\theta, z_-)$. We hypothesize that the effect of b on estimator variance may underlie b's effect on learning. Specifically, the variance of the ISD estimator is, for appropriately chosen gradient step-sizes, a convex function of b with rather low curvature around the local minimum (Fig.3.B). Because the variance of the ISD estimator, as a function of b, does not increase quickly near the minimum, there is a range of values with sufficiently low variance to allow for robust learning.

Lastly, we know from the theory section that the trace of the covariance of $\hat{g}(\theta)$ is a convex function of the positive phase parameter $b \in (0, 1)$, so the value of b leading to minimal estimator variance is unique. If the effect of b on learning is primarily via its effect on the variance of the ISD estimator, the convexity of this function could be used to simplify hyper-parameter initialization. Specifically, one can estimate means and covariances of the positive and negative phase gradient terms and then use our theoretical results (§A.1 Equation 11) to solve for the value of b that produces the minimal estimator variance analytically.

5. Discussion

This work has implications for experimental and theoretical neuroscience, and neuromorphic computing. For neuroscience, it indicates that a lack of periodicity or sharp learning rate modulation in a neural circuit is not a basis for discounting CL as a potential algorithm implemented by that circuit. Our results add to a body of literature (Illing et al., 2021; Laborieux & Zenke, 2022; Pogodin & Latham, 2020) motivating an experimental search for global state variables, e.g. b, that should be broadcast to all neurons in a network–perhaps via neuromodulatory signals–to enable learning. In particular, we demonstrate that biological systems implementing CL need not necessarily rely on an oscillatory global signal, as has been previously proposed (Baldi & Pineda, 1991; Laborieux & Zenke, 2022), but could instead

implement phase switching through a more stochastic form of gating. This could correspond behaviourally to saccades (Illing et al., 2021) or periods of rest associated with hippocampal replay events (Davidson et al., 2009), for example. Furthermore, our research exhibits two opportunities for synergy with other recent neuroscience literature. First, it has been hypothesized that neural activation dynamics encode probabilities by sampling (Echeveste et al., 2020). Our work provides a potential path to learning bio-plausible probabilistic sampling networks: probabilistic models can be framed as energy based models with a negative log-likelihood energy function (Ackley et al., 1985; LeCun et al., 2006), and the gradient of such a model is of the two-term CL style discussed herein. Second, several recent studies of equilibrium learning provide powerful new learning models, but lack theoretical guarantees for concurrent, online, latent space and learning dynamics (Laborieux & Zenke, 2022; Meulemans et al., 2022). Our study provides guarantees that may be of use, with Theorems 3.4 and 3.5.

For neuromorphic computing, this work provides theoretical and empirical motivation for CL algorithms with more relaxed memory requirements than those of traditional, temporally non-local approaches. Thus, it provides a potential path towards simplifying CL and equilibrium CL algorithms to be more easily implemented on neuromorphic hardware, analogous to past work on Equilibrium Propagation (Ernoult et al., 2020) but for a much broader algorithm class. It is our hope that this will aid in the development of more energy-efficient deep learning.

The lesser memory requirements of the proposed gradient estimator come at the cost of higher estimator variance. In our study this manifested in a need for reduced learning rates, effectively averaging over more gradient steps to combat this variance increase, and further resulted in slight reductions in algorithm performance. This is a clear example of an (at least approximately) unbiased algorithm that trades lower variance for locality. Interestingly, such a trade-off has been noted before for spatial locality (Richards et al., 2019), as opposed to the temporal locality explored in this study. Further work is warranted to determine whether such a locality-variance trade-off is a general phenomenon for unbiased gradient estimators.

It bears mentioning that the added noise in our learning rule is not a priori effected by the curse of dimensionality, as is the case for certain classic computational neuroscience algorithms (Werfel et al., 2003). The additional variance introduced by ISD is caused by the variation of a single, scalar Bernoulli variable, b; therefore, as shown in Equation 10, ISD inherits much of its variance from the behaviour of the positive and negative phase terms of the CL learning rule it is based upon.

This work is, to our knowledge, the first to provide a principled method of manipulating the amount of time spent in the positive versus negative phase for CL, via the importance sampling positive-phase probability b. Further work is warranted to determine how this might be generalized to other biphasic algorithms, like Wake-Sleep (Dayan et al., 1995), and how this parameter might be leveraged for computational benefits. In particular one might be able to trade compute for data in sparse data regimes. In the case of a very low b an algorithm might still perform well using few positive phases—and thus few data points—compared with the equal number of positive and negative phases required by a traditional CL approach. Ultimately, our results provide a compelling example where a gradient estimator with markedly more variance still enables effective learning, while providing important benefits for memory storage and temporal locality.

6. Conclusion

In this study we proposed an alternative, temporally local gradient estimate for CL, which can be applied to any CL algorithm that uses a two-term gradient to learn from positive and negative phases. We further proved two theorems applying to algorithms that require equilibrium dynamics to be run to obtain the gradient estimate used at each step of gradient descent. This theorem shows that equilibrium-based algorithms that allow learning to continue during the dynamic phases and, moreover, have stochastic phase lengths, will still exhibit asymptotically unbiased gradients in the small learning rate long mean phase-length limit. Notably, this theorem applies, with some basic assumptions, to all equilibrium-based algorithms, contrastive or otherwise. This theory work was supported by the experiments presented above. In summary, this work suggests that CL does not require the precisely regulated, periodic learning dynamics that are classically used to train this form of AI, thus extending the set of systems capable of implementing CL beyond what might have been previously thought possible.

Acknowledgements

The authors acknowledge support from Samsung Electronics Co., Ldt. and the Simons Foundation Collaboration for the Global Brain. GL acknowledges the support from Canada CIFAR AI Chair Program, as well as NSERC Discovery Grant [RGPIN-2018-04821] and the Canada Research Chair in Neural Computations and Interfacing (CIHR, tier 2). EW acknowledges support from three scholarships: a NSERC CGS-D, a FRQNT B2X, and a UNIQUE Excellence Scholarship. Authors wish to thank the other members of the Lajoie lab for their support; in particular, Maximilian Puelma Touzel and Alexandre Payeur for the helpful conversations, and Mohammad Pezeshki for inspiration in the coding of the forward-forward algorithm.

References

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

Baldi, P. and Pineda, F. Contrastive learning and neural oscillations. *Neural computation*, 3(4):526–545, 1991.

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Cao, Y., Summerfield, C., and Saxe, A. Characterizing emergent representations in a space of candidate learning rules for deep networks. *Advances in Neural Information Processing Systems*, 33:8660–8670, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Crick, F. and Mitchison, G. The function of dream sleep. *Nature*, 304(5922):111–114, 1983.

Davidson, T. J., Kloosterman, F., and Wilson, M. A. Hippocampal replay of extended experience. *Neuron*, 63(4): 497–507, 2009.

Davis, D. and Drusvyatskiy, D. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

Diederich, S. and Opper, M. Learning of correlated patterns in spin-glass networks by local learning rules. *Physical review letters*, 58(9):949, 1987.

Echeveste, R., Aitchison, L., Hennequin, G., and Lengyel, M. Cortical-like dynamics in recurrent circuits optimized for sampling-based probabilistic inference. *Nature neuroscience*, 23(9):1138–1149, 2020.

Ernoult, M., Grollier, J., Querlioz, D., Bengio, Y., and Scellier, B. Equilibrium propagation with continual weight updates. *arXiv preprint arXiv:2005.04168*, 2020.

Fischer, A. and Igel, C. Training restricted boltzmann machines: An introduction. *Pattern Recognition*, 47(1): 25–39, 2014.

Frenkel, C., Bol, D., and Indiveri, G. Bottom-up and top-down neural processing systems design: Neuromorphic intelligence as the convergence of natural and artificial intelligence. *arXiv preprint arXiv:2106.01288*, 2021.

Friston, K. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456):815–836, 2005.

Ghadimi, S., Lan, G., and Zhang, H. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1): 267–305, 2016.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Hennequin, G., Ahmadian, Y., Rubin, D. B., Lengyel, M., and Miller, K. D. The dynamical regime of sensory cortex: stable dynamics around a single stimulus-tuned attractor account for patterns of noise variability. *Neuron*, 98(4): 846–860, 2018.

Hinton, G. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Hinton, G. E., Sejnowski, T. J., and Ackley, D. H. *Boltzmann machines: Constraint satisfaction networks that learn*. Carnegie-Mellon University, Department of Computer Science Pittsburgh, PA, 1984.

Illing, B., Ventura, J., Bellec, G., and Gerstner, W. Local plasticity rules can learn deep representations using self-supervised contrastive predictions. *Advances in Neural Information Processing Systems*, 34:30365–30379, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Laborieux, A. and Zenke, F. Holomorphic equilibrium propagation computes exact gradients through finite size oscillations. *arXiv preprint arXiv:2209.00530*, 2022.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Lasota, A. and Mackey, M. C. *Chaos, fractals, and noise: stochastic aspects of dynamics*, volume 97. Springer Science & Business Media, 1998.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):1–10, 2016.

Meulemans, A., Zucchet, N., Kobayashi, S., Von Oswald, J., and Sacramento, J. The least-control principle for local learning at equilibrium. *Advances in Neural Information Processing Systems*, 35:33603–33617, 2022.

Movellan, J. R. Contrastive hebbian learning in the continuous hopfield model. In *Connectionist models*, pp. 10–17. Elsevier, 1991.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ororbia, A. and Mali, A. The predictive forward-forward algorithm. *arXiv preprint arXiv:2301.01452*, 2023.

Payeur, A., Guerguiev, J., Zenke, F., Richards, B. A., and Naud, R. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature neuroscience*, 24(7):1010–1019, 2021.

Pogodin, R. and Latham, P. Kernelized information bottleneck leads to biologically plausible 3-factor hebbian learning in deep networks. *Advances in Neural Information Processing Systems*, 33:7296–7307, 2020.

Rao, R. P. and Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R. P., de Berker, A., Ganguli, S., et al. A deep learning framework for neuroscience. *Nature neuroscience*, 22(11): 1761–1770, 2019.

Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Robert, C. P., Casella, G., and Casella, G. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

Rosenthal, J. S. Convergence rates for markov chains. *Siam Review*, 37(3):387–405, 1995.

Scellier, B. and Bengio, Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017.

Schuman, C. D., Kulkarni, S. R., Parsa, M., Mitchell, J. P., Date, P., and Kay, B. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022.

Shamir, O. and Zhang, T. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning*, pp. 71–79. PMLR, 2013.

Sorscher, B., Mel, G. C., Ocko, S. A., Giocomo, L. M., and Ganguli, S. A unified theory for the computational and mechanistic origins of grid cells. *Neuron*, 2022.

Srinivasan, M. V., Laughlin, S. B., and Dubs, A. Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 216(1205):427–459, 1982.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13693–13696, 2020.

Werfel, J., Xie, X., and Seung, H. Learning curves for stochastic gradient descent in linear feedforward networks. *Advances in neural information processing systems*, 16, 2003.

Yin, G. and Zhang, Q. *Discrete-time Markov chains: two-time-scale methods and applications*, volume 55. Springer Science & Business Media, 2005.

A. Importance Sampling - Discrete (ISD) Gradient Estimate

A.1. Bias and Variance of ISD

In this section we investigate the bias and variance of the ISD estimator. First, we restate the standard CL gradient estimator and the ISD estimator, along with Proposition 3.1 and Lemmas 3.2 and 3.3 for ease of reference. The standard two term estimate, Equation 2 in the main paper, is

$$\hat{g}(\theta) = g_+(\theta, z_+) - g_-(\theta, z_-), \tag{8}$$

and the ISD gradient estimate, Equation 3 in the main paper, is

$$\hat{g}(\theta) = \frac{B}{b} g_+(\theta, z_+) - \frac{1-B}{1-b} g_-(\theta, z_-). \tag{9}$$

Proposition A.1. *For $b \in (0, 1)$, the ISD gradient estimate of Equation 3 is an unbiased estimate of the gradient given in Equation 2.*

Proof. Follows trivially by taking expectations w.r.t. B and observing that, for a Bernoulli random variable, the expectation is equal to the parameter b so that the b and $1 - b$ fraction denominators disappear. □

Remark 1: We further note that this is, by definition, a single-sample importance sampling approach to evaluating the expected value $\mathbb{E}\big(f(B)\big)$, this being the two-term sum of Equation 8, if $f(1) = g_+(\theta, z_+)$ and $f(0) = g_-(\theta, z_-)$.

Remark 2: In the case where $b = 0.5$, an alternative perspective on this estimate is that it is a version of stochastic gradient descent style uniform sampling from an augmented dataset. This augmented set is composed of all positive and negative samples, each sample augmented with an extra variable indicating whether the sample is positive or negative. This extra variable is then used to determine whether the gradient is added or subtracted to the current parameter value.

Lemma A.2. *Assume $g_+(\theta, z_+)$ and $g_-(\theta, z_-)$ are statistically independent and the inner product of their means is non-negative. If $b \in (0, 1)$, the ISD gradient estimate of Equation 3 has strictly greater variance (quantified by the trace of the covariance of the gradient vector) than the estimate given by Equation 2.*

Lemma A.3. *The variance of the gradient estimator given by Equation 3 is a convex function of b on $(0, 1)$.*

Let us define $\Sigma_+ = \mathrm{cov}\big(g_+(\theta, z_+)\big)$, $\Sigma_- = \mathrm{cov}\big(g_-(\theta, z_-)\big)$, $\Sigma_{+-} = \mathrm{cov}\big(g_+(\theta, z_+), g_-(\theta, z_-)\big)$, $\mu_+ = \mathbb{E}\big(g_+(\theta, z_+)\big)$, and $\mu_- = \mathbb{E}\big(g_-(\theta, z_-)\big)$ To prove the first Lemma, we first note that the covariance matrix of the standard CL gradient estimate given by Equation 8 is $\Sigma_+ + \Sigma_- - 2\Sigma_{+-}$. We now prove the following proposition, which provides the bulk of the proof of the two lemmas:

Proposition A.4. *Assume $b \in (0, 1)$. The covariance matrix for the ISD gradient estimate vector defined by Equation 9 is*

$$\Sigma = \frac{1}{b}\Sigma_+ + \frac{1}{1-b}\Sigma_- + \frac{1-b}{b}\mu_+\mu_+^\top + \frac{b}{1-b}\mu_-\mu_-^\top + \mu_+\mu_-^\top + \mu_-\mu_+^\top. \tag{10}$$

Moreover, the trace of this matrix is a convex function of b, with minimum at

$$b_{\min} = \frac{\mathrm{Tr}[\Sigma_+ + \mu_+\mu_+^\top] - \sqrt{\mathrm{Tr}[\Sigma_+ + \mu_+\mu_+^\top]\mathrm{Tr}[\Sigma_- + \mu_-\mu_-^\top]}}{\mathrm{Tr}[\Sigma_+ + \mu_+\mu_+^\top] - \mathrm{Tr}[\Sigma_- + \mu_-\mu_-^\top]}, \tag{11}$$

Proof. We wish to calculate the covariance of the gradient estimate, as a function of b, to compare with the covariance of the original gradient estimate, and to see when the trace of this covariance is minimized w.r.t. b. Note that we use the trace of the covariance instead of the variance since we are working in a multi-dimensional space. We begin by deriving the matrix of second moments of the gradient vector, which we denote by corr. We also define $f_+(\theta, z_+) = \frac{B}{b} g_+(\theta, z_+)$ and $f_-(\theta, z_-) = -\frac{1-\mathcal{B}}{1-b} g_-(\theta, z_-)$ to simplify notation. Now, note that under this notation, the i^{th} element of the ISD gradient

vector estimate is $f_+(\theta, z_+)_i + f_-(\theta, z_-)_i$. Thus

$$\begin{aligned}
&\operatorname{corr}\big(f_+(\theta, z_+)_i + f_-(\theta, z_-)_i, f_+(\theta, z_+)_j + f_-(\theta, z_-)_j\big) \\
&= \mathbb{E}\Big[f_+(\theta, z_+)_i f_+(\theta, z_+)_j + f_+(\theta, z_+)_i f_-(\theta, z_-)_j + f_+(\theta, z_+)_j f_-(\theta, z_-)_i + f_-(\theta, z_-)_i f_-(\theta, z_-)_j\Big] \\
&= \mathbb{E}\Big[f_+(\theta, z_+)_i f_+(\theta, z_+)_j + f_-(\theta, z_-)_i f_-(\theta, z_-)_j\Big] \\
&= \frac{1}{b}[\Sigma_{+ij} + \mu_{+i}\mu_{+j}] + \frac{1}{1-b}[\Sigma_{-ij} + \mu_{-i}\mu_{-j}],
\end{aligned} \tag{12}$$

where in the second line we used linearity of expectation and the fact that $B(1-B) = 0$, and the in third line we used the form of the second moments of a Bernoulli random variable and the relationship between covariance and the second moment. Subtracting the means to arrive at the covariance and rearranging, we get Equation 10.

Observe that this function has vertical asymptotes at $b = 0$ and $b = 1$. We are interested in finding the minimum of the trace of the matrix w.r.t. b on the interval $\alpha \in (0, 1)$. Differentiating gives

$$\begin{aligned}
\frac{\partial \operatorname{Tr}[\Sigma]}{\partial b} &= -\frac{1}{b^2}\operatorname{Tr}[\Sigma_+] + \frac{1}{(1-b)^2}\operatorname{Tr}[\Sigma_-] - \frac{1}{b^2}\operatorname{Tr}[\mu_+\mu_+^\top] + \frac{1}{(1-b)^2}\operatorname{Tr}[\mu_-\mu_-^\top] \\
&= -\frac{1}{b^2}\operatorname{Tr}[\Sigma_+ + \mu_+\mu_+^\top] + \frac{1}{(1-b)^2}\operatorname{Tr}[\Sigma_- + \mu_-\mu_-^\top]
\end{aligned} \tag{13}$$

Setting this equal to zero and solving for b using the quadratic formula gives:

$$b = \frac{\operatorname{Tr}[\Sigma_+ + \mu_+\mu_+^\top] \pm \sqrt{\operatorname{Tr}[\Sigma_+ + \mu_+\mu_+^\top]\operatorname{Tr}[\Sigma_- + \mu_-\mu_-^\top]}}{\operatorname{Tr}[\Sigma_+ + \mu_+\mu_+^\top] - \operatorname{Tr}[\Sigma_- + \mu_-\mu_-^\top]}. \tag{14}$$

Some simple algebra shows that only the difference and not the sum in the above equation yields a critical point on the interval $(0, 1)$, so the difference is the minimum we desire. Finally, differentiating one more time gives:

$$\frac{\partial^2 \operatorname{Tr}[\Sigma]}{\partial b^2} = \frac{2}{b^3}\operatorname{Tr}[\Sigma_+ + \mu_+\mu_+^\top] + \frac{2}{(1-b)^3}\operatorname{Tr}[\Sigma_- + \mu_-\mu_-^\top]. \tag{15}$$

The trace of the sum of a covariance matrix and a vector outer product is positive, so this expression is positive on $(0, 1)$, thus we get a convex function of b on this interval. □

Lemma A.3 is proven by the proof of the above proposition. For Lemma A.2, it suffices to note that if $g_+(\theta, z_+)$ is statistically independent of $g_-(\theta, z_-)$ then their covariance is given by $\Sigma_+ + \Sigma_-$. The proof of Lemma A.2 follows by the linearity of the trace and the assumption that $\operatorname{Tr}(\mu_+\mu_-^\top) = \operatorname{Tr}(\mu_-\mu_+^\top) = \mu_+^\top\mu_- \geq 0$.

B. Bias of Constant Learning Rate for Equilibrium Learning Gradient Estimates

We investigate an equilibrium learning system where, to estimate the gradient needed for one parameter update, $\hat{g}(\theta)$, one must run internal dynamics, following a Markov chain, on some activation space $\mathcal{Z}$ until these dynamics are sufficiently close to an equilibrium.

We investigate the question of how much bias is introduced into the SGD dynamics of this equilibrium-learning system when, instead of fixing θ during each phase of dynamics before each gradient step, one allows gradient steps to be taken at *every step of the dynamics*. Specifically, we are interested in proving Theorem 3.4.

In an abuse of notation we allow p to represent any probability density (or mass function) appearing in the proof, with each distribution being discerned by the arguments—e.g. $p(z_t|z_0, \theta) = \int_{\mathcal{Z}\times\ldots\times\mathcal{Z}} \prod_{k=1}^{t} p(z_k|z_{k-1}, \theta)\mathrm{d}z_1 \ldots \mathrm{d}z_{t-1}$ is a t step transition kernel and $p(z_k|z_{k-1}, \theta)$ is a single step transition. We adopt a discrete time framework with $t \in \mathbb{N}$ given that SGD is typically implemented in discrete time, though we expect similar results for continuous time. The proposed

algorithm views each stimulus, or samples from its internal model, for T time-steps and the learning rate at each time-step is scaled by a factor of $\frac{1}{\tau}$, which is the mean of T. We use $\theta_t \in \Theta \,\forall\, t$ to denote the parameter value at time t, and z_t to denote the network state at this time. η denotes the learning rate before scaling by the inverse of τ.

Theorem B.1. *Consider a single phase of equilibrium-based learning where the phase is either fixed to value τ or ends stochastically at each step with small probability $\frac{1}{\tau}$ and, instead of waiting until the end of the phase to do an update with learning rate η, an update is performed at every step of the equilibrium dynamics with learning rate $\frac{\eta}{\tau}$. Assume that equilibrium dynamics evolve for fixed θ according to the Markov transition kernel $p(z_t|z_{t-1}, \theta)$. Further assume that $\hat{g}(\theta)$ is differentiable and bounded, with a bounded derivative, for all network state values, the Markov kernel for the equilibrium dynamics is continuously differentiable w.r.t. θ for all network state values, the k-step kernel $p(z_t|z_{t-k}, \theta)$ is bounded w.r.t. z_{t-1} for all parameter values, and the function $\Gamma(x) = \int_{\mathcal{Z}} \left|\frac{\partial p(x|y,\xi_{\max})}{\partial \theta}\right| p(y|z_0, \theta_0)\mathrm{d}y$ is integrable for the given initial conditions and all values of $\xi_{\max}$. Finally, assume that the equilibrium dynamics converge to a stationary distribution weakly with a rate as least as fast as $\mathcal{O}(\frac{1}{t^2})$. Then the bias of the learning updates are of the following order:*

$$\mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}(\eta^2 \tau), \tag{16}$$

where $s = \tau^m$ for $0 < m < 1$ and $\mathcal{O}$ is big-O notation.

Remark on dimension of parameters: The following proof relies very heavily on the multivariate corollary of the mean value theorem, which we state in section B.3 for completeness.

B.1. Proof of Theorem 3.4

Proof. In what follows we restrict t to the interval $[0, T)$—that is, one period of learning. T can either be chosen deterministically or stochastically, as we will discuss below. The full dynamics of network state and parameter updates are given by the following equations:

$$\begin{aligned} z_{t+1} &\sim p(z_{t+1}|z_t, \theta_t) \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\tau}\hat{g}(z_t, \theta_t), \end{aligned} \tag{17}$$

where $\hat{g}$ is the gradient estimate for the model, $p(z_{t+1}|z_t, \theta)$ is, for fixed θ, the transition function of a homogeneous Markov chain that we assume converges uniquely to the stationary distribution of the model. Thus, if we set $\theta_t = \theta$ for all t, $\{z_t\}_{t=0}^T$ would be a Markov chain performing MCMC sampling from our probability model with parameter θ. However, given the dependence on θ_t, $\{z_t\}_{t=0}^T$ is not performing exact MCMC sampling and is, in fact, not even a Markov chain, due to the sequence of temporal dependencies induced by θ_t. Lastly, we note that, conditioned on z_t, θ_{t+1} is a deterministic variable since the input stimulus is not changing over the period $[0, T)$ and the only source of randomness is via z.

Our goal is to prove that the sum of parameter changes over the phase $t \in [0, T)$, which we denote by the random variable $G(z_0, \theta_0)$, is asymptotically unbiased in the limit of small learning rate, η, and long mean stimulus presentation time, τ. That is, we will prove the following:

$$\mathbb{E}\big[G(z_0, \theta)\big] = \eta\hat{g}(\theta) + \mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}(\eta^2\tau). \tag{18}$$

To this end, we simply linearize about θ_0 so as to remove the temporal dependence brought about by the continued learning. This allows us to make use of the convergence properties of the homogeneous Markov chain. We now proceed in three steps: (1) expand $\hat{g}$ to remove the dependence of the gradient, via θ_t, on the slow learning timescale; (2) expand $p(z_t|z_0, \theta_0)$ to remove the rest of the expectation's dependence—via the system's internal dynamics—on the slow timescale; (3) leverage the convergence of the homogeneous Markov chain to the stationary distribution to obtain samples to approximate the desired expectations. Steps (1) and (2) jointly lead to the error term that is quadratic in η, in Equation 18, and step (3) results in the other error term. Before embarking on the three main steps of the proof, we must first engage in a preliminary analysis to account for stochastic phase lengths.

Preliminary for Stochastic Phase Length: Using the law of total expectation (tower property), and defining the random variable $T_s = \mathbb{1}_{T \geq s}$

$$\begin{aligned}\mathbb{E}\big[G(z_0,\theta)\big] &= \mathbb{E}\big[\mathbb{E}[G(z_0,\theta)|T_s]\big] = \mathbb{E}[G(z_0,\theta)|T_s=1]p(T_s=1) + \mathbb{E}[G(z_0,\theta)|T_s=0]p(T_s=0) \\ &= \mathbb{E}[G(z_0,\theta)|T_s=1] + \big(\mathbb{E}[G(z_0,\theta)|T_s=0] - \mathbb{E}[G(z_0,\theta)|T_s=1]\big)p(T_s=0).\end{aligned} \tag{19}$$

Observe that, by the assumption that the gradient updates are bounded, $G(z_0,\theta_0) \leq K_1\eta T/\tau$ for some constant K_1. Furthermore, it can be shown that $p(T_s=0) = \mathcal{O}(s/\tau)$ (as we demonstrate in §B.4). Thus

$$\mathbb{E}\big[G(z_0,\theta)\big] = \mathbb{E}[G(z_0,\theta)|T_s=1] + \mathcal{O}\Big(\frac{\eta s\mathbb{E}(T)}{\tau^2}\Big) = \mathbb{E}[G(z_0,\theta)|T_s=1] + \mathcal{O}\Big(\frac{\eta s}{\tau}\Big), \tag{20}$$

where we have used $\mathbb{E}(T) = \tau$ (see §B.5).

What Equation 20 means is that, for a large enough average phase length, we can assume that we are dealing with a value of T that is larger than some deterministic value, s, even in the case of stochastic phase lengths–albeit with the above error term. Moreover, if we apply the tower property again, we get:

$$\mathbb{E}[G(z_0,\theta)|T_s=1] = \mathbb{E}_T\big[\mathbb{E}[G(z_0,\theta)|T]|T_s=1\big]. \tag{21}$$

For the three main steps of the proof we will thus analyze the inner expectation, $\mathbb{E}[G(z_0,\theta)|T]$ (in an abuse of notation we will refrain from explicitly writing the condition that $T \geq s$), the analysis of which is equivalent to the case of a deterministic phase length. At the end of the proof we will take into account the stochasticity in T. We now proceed to the three main steps of the proof.

Step 1: This section is a straightforward application of Lemma B.3. Observe that one can write the parameter dynamics as:

$$\theta_t = \theta_0 + \frac{\eta}{\tau}\sum_{k=0}^{t-1}\hat{g}(z_k,\theta_k), \tag{22}$$

and thus,

$$G(z_0,\theta_0) = \frac{\eta}{\tau}\sum_{t=0}^{T-1}\hat{g}(z_t,\theta_t). \tag{23}$$

Let us focus on the j^{th} element of $\hat{g}$. Assuming that $\hat{g} \in \mathcal{C}^0(\theta)$ (differentiable w.r.t. θ), we can expand it to first order about θ_0:

$$\hat{g}_j(z_t,\theta_t) = \hat{g}_j\Big(z_t,\theta_0 + \frac{\eta}{\tau}\sum_{k=0}^{t-1}\hat{g}(z_k,\theta_k)\Big) = \hat{g}_j(z_t,\theta_0) + \mathcal{O}\Big(\frac{\eta}{\tau}\Big|\Big|\sum_{k=0}^{t-1}\hat{g}(z_k,\theta_k)\Big|\Big|_1\Big). \tag{24}$$

If we assume that the gradient is bounded (e.g. we clip the gradient above some sufficiently large value), and that the partial derivatives of the gradient w.r.t. θ are bounded, then the second term on the RHS is $\mathcal{O}\big(t\frac{\eta}{\tau}\big)$. Thus,

$$G(z_0,\theta_0) = \frac{\eta}{\tau}\sum_{t=0}^{T-1}\hat{g}(z_t,\theta_0) + \mathcal{O}\Big(\frac{\eta^2}{\tau^2}\Big)\sum_{t=0}^{T-1}t. \tag{25}$$

The last factor in the final sum is order T^2 by the quadratic nature of the triangular summation.

Step 2: By linearity of expectation, and using the results of step 1,

$$\mathbb{E}\big[G(z_0,\theta_0)|T\big] = \frac{\eta}{\tau}\sum_{t=0}^{T-1}\mathbb{E}\big[\hat{g}(z_t,\theta_0)\big] + \mathcal{O}\Big(\eta^2\frac{T^2}{\tau^2}\Big). \tag{26}$$

Focusing on the t^{th} expectation on the RHS:

$$\mathbb{E}\big[\hat{g}(z_t, \theta_0)|T\big] = \int_{\mathcal{Z}} \hat{g}(z_t, \theta_0) p(z_t|z_0, \theta_0) \mathrm{d}z_t, \tag{27}$$

Focusing specifically on the probability, we get:

$$\begin{aligned} p(z_t|z_0, \theta_0) &= \int_{\mathcal{Z}} \int_{\Theta} p(z_t, z_{t-1}, \theta_{t-1}|z_0, \theta_0) \mathrm{d}\theta_{t-1} \mathrm{d}z_{t-1} \\ &= \int_{\mathcal{Z}} \int_{\Theta} p(z_t|z_{t-1}, \theta_{t-1}, z_0, \theta_0) p(z_{t-1}, \theta_{t-1}|z_0, \theta_0) \mathrm{d}\theta_{t-1} \mathrm{d}z_{t-1} \\ &= \int_{\mathcal{Z}} \int_{\Theta} p(z_t|z_{t-1}, \theta_{t-1}) p(z_{t-1}, \theta_{t-1}|z_0, \theta_0) \mathrm{d}\theta_{t-1} \mathrm{d}z_{t-1}, \end{aligned} \tag{28}$$

where, in the final line, we used the dependence structure of the MCMC dynamics.

Let us define $\Delta\theta_t = \frac{\eta}{\tau} \sum_{k=0}^{t-1} \hat{g}(z_k, \theta_k)$ and use again our assumption that the gradient is bounded, so that in t steps $||\Delta\theta_t||_1 \leq \Delta_{\max}\theta_t$, a deterministic quantity. Assume further that the transition function is differentiable w.r.t. θ (in most cases this follows from our assumption that $\hat{g}$ is differentiable). Then, the mean value theorem (see Lemma B.3), we have:

$$= \int_{\mathcal{Z}} \int_{\Theta} \left[p(z_t|z_{t-1}, \theta_0) + \Delta\theta_{t-1} \cdot \frac{\partial p(z_t|z_{t-1}, \xi_{t-1})}{\partial \theta} \right] p(z_{t-1}, \theta_{t-1}|z_0, \theta_0) \mathrm{d}\theta_{t-1} \mathrm{d}z_{t-1} \tag{29}$$

where ξ_{t-1} is in a cube of edge length $||\Delta\theta_{t-1}||_1$ centered at θ_0, and is a random variable on account of the variability in $\Delta\theta_t$. We now proceed as follows

$$\begin{aligned} &= \int_{\mathcal{Z}} \int_{\Theta} p(z_t|z_{t-1}, \theta_0) p(z_{t-1}, \theta_{t-1}|z_0, \theta_0) \mathrm{d}\theta_{t-1} \mathrm{d}z_{t-1} \\ &+ \int_{\mathcal{Z}} \int_{\Theta} \Delta\theta_{t-1} \cdot \frac{\partial p(z_t|z_{t-1}, \xi_{t-1})}{\partial \theta} p(z_{t-1}, \theta_{t-1}|z_0, \theta_0) \mathrm{d}\theta_{t-1} \mathrm{d}z_{t-1} \\ &= \int_{\mathcal{Z}} p(z_t|z_{t-1}, \theta_0) p(z_{t-1}|z_0, \theta_0) \mathrm{d}z_{t-1} \\ &+ \int_{\mathcal{Z}} \int_{\Theta} \Delta\theta_{t-1} \cdot \frac{\partial p(z_t|z_{t-1}, \xi_{t-1})}{\partial \theta} p(z_{t-1}, \theta_{t-1}|z_0, \theta_0) \mathrm{d}\theta_{t-1} \mathrm{d}z_{t-1} \end{aligned} \tag{30}$$

The second term on the RHS of the final line is a bias term, which we will now bound. Observe:

$$\begin{aligned} &\left| \int_{\mathcal{Z}} \int_{\Theta} \Delta\theta_{t-1} \cdot \frac{\partial p(z_t|z_{t-1}, \xi_t)}{\partial \theta} p(z_{t-1}, \theta_{t-1}|z_0, \theta_0) \mathrm{d}\theta_{t-1} \mathrm{d}z_{t-1} \right| \\ &\leq \int_{\mathcal{Z}} \int_{\Theta} \Delta_{\max}\theta_{t-1} \Big|\Big| \frac{\partial p(z_t|z_{t-1}, \xi_{t-1})}{\partial \theta} \Big|\Big|_1 p(z_{t-1}, \theta_{t-1}|z_0, \theta_0) \mathrm{d}\theta_{t-1} \mathrm{d}z_{t-1}, \end{aligned} \tag{31}$$

where the inequality follows from Jensen's inequality, to bring the absolute values inside the integral, and the use of $\Delta_{\max}\theta_{t-1}$. Integrating out the parameters at time $t-1$ and applying $\Delta_{\max}\theta_{t-1} \leq \Delta_{\max}\theta_T$,

$$\begin{aligned} &= \Delta_{\max}\theta_T \int_{\mathcal{Z}} \Big|\Big| \frac{\partial p(z_t|z_{t-1}, \xi_{\max})}{\partial \theta} \Big|\Big|_1 p(z_{t-1}|z_0, \theta_0) \mathrm{d}z_{t-1} \\ &= \mathcal{O}\Big(\eta \frac{T}{\tau}\Big) \Gamma(z_t), \end{aligned} \tag{32}$$

where we have used the fact that $\mathcal{O}(\Delta_{\max}\theta_T) = \mathcal{O}\big(\eta\frac{T}{\tau}\big)$. We have further assumed that the transition function is $\mathcal{C}^1(\theta)$, so that by continuity on the compact cube of edge length $\Delta_{\max}\theta_T$ centered at θ_0 we get $\left|\left|\frac{\partial p(z_t|z_{t-1},\xi_{t-1})}{\partial\theta}\right|\right|_1 \leq \left|\left|\frac{\partial p(z_t|z_{t-1},\xi_{\max})}{\partial\theta}\right|\right|_1$ for some value of $\xi_{\max}$. Lastly, we have simplified the expression with the function $\Gamma(x) = \int_{\mathcal{Y}}\left|\left|\frac{\partial p(x|y,\xi_{\max})}{\partial\theta}\right|\right|_1 p(y|z_0,\theta_0)\mathrm{d}y$ defined in the theorem statement. We now have:

$$p(z_t|z_0,\theta_0) = \int_{\mathcal{Z}} p(z_t|z_{t-1},\theta_0)p(z_{t-1}|z_0,\theta_0)\mathrm{d}z_{t-1} + \mathcal{O}\Big(\eta\frac{T}{\tau}\Big)\Gamma(z_t). \tag{33}$$

If $t = 2$ we stop here. For $t > 2$, observe that the second factor in the integrand is simply the original transition function moved one time-step back in time, and that the steps we have gone through to arrive at this point work for an arbitrary time-step. That is, for $k \in \{2,\dots,T\}$, we have:

$$p(z_k|z_0,\theta_0) = \int_{\mathcal{Z}} p(z_k|z_{k-1},\theta_0)p(z_{k-1}|z_0,\theta_0)\mathrm{d}z_{k-1} + \mathcal{O}\Big(\eta\frac{T}{\tau}\Big)\Gamma(z_k), \tag{34}$$

This suggests taking a recursive approach to expanding $p(z_t|z_0,\theta_0)$, which is what we do now. Applying Equation 34 to expand Equation 33, and writing $p(z_t|z_{t-1},\theta_0) = p(z_t|z_{t-1})$, we get:

$$\begin{aligned} p(z_t|z_0,\theta_0) = &\int_{\mathcal{Z}\times\mathcal{Z}} p(z_t|z_{t-1})p(z_{t-1}|z_{t-2})p(z_{t-2}|z_0)\mathrm{d}z_{t-2}\times\mathrm{d}z_{t-1} \\ &+ \int_{\mathcal{Z}} p(z_t|z_{t-1})\mathcal{O}\Big(\eta\frac{T}{\tau}\Big)\Gamma(z_{t-1})\mathrm{d}z_{t-1} + \mathcal{O}\Big(\eta\frac{T}{\tau}\Big)\Gamma(z_t). \end{aligned} \tag{35}$$

Applying the recursive Equation 34 $t-3$ more times yields the following

$$\begin{aligned} &= \int_{\mathcal{Z}}\cdots\int_{\mathcal{Z}}\prod_{k=0}^{t-1} p(z_{t-k}|z_{t-k-1},\theta_0)\mathrm{d}z_1\dots\mathrm{d}z_{t-1} \\ &+ \sum_{k=1}^{t-1}\int_{\mathcal{Z}} p(z_t|z_{t-k})\mathcal{O}\Big(\eta\frac{T}{\tau}\Big)\Gamma(z_{t-k})\mathrm{d}z_{t-k} + \mathcal{O}\Big(\eta\frac{T}{\tau}\Big)\Gamma(z_t). \end{aligned} \tag{36}$$

We now insert Equation 36 into Equation 27:

$$\begin{aligned} \mathbb{E}\big[\hat{g}(z_t,\theta_0)|T\big] &= \int_{\mathcal{Z}}\hat{g}(z_t,\theta_0)\int_{\mathcal{Z}}\cdots\int_{\mathcal{Z}}\prod_{k=0}^{t-1} p(z_{t-k}|z_{t-k-1},\theta_0)\mathrm{d}z_1\dots\mathrm{d}z_t \\ &+ \int_{\mathcal{Z}}\hat{g}(z_t,\theta_0)\sum_{k=1}^{t-1}\int_{\mathcal{Z}} p(z_t|z_{t-k})\mathcal{O}\Big(\eta\frac{T}{\tau}\Big)\Gamma(z_{t-k})\mathrm{d}z_{t-k}\mathrm{d}z_t \\ &+ \int_{\mathcal{Z}}\hat{g}(z_t,\theta_0)\mathcal{O}\Big(\eta\frac{T}{\tau}\Big)\Gamma(z_t)\mathrm{d}z_t \\ &= \int_{\mathcal{Z}}\hat{g}(z_t,\theta_0)\int_{\mathcal{Z}}\cdots\int_{\mathcal{Z}}\prod_{k=0}^{t-1} p(z_{t-k}|z_{t-k-1},\theta_0)\mathrm{d}z_1\dots\mathrm{d}z_t + \mathcal{O}\Big(\eta\frac{T^2}{\tau}\Big), \end{aligned} \tag{37}$$

where for the last equality we used our previous assumption that $\hat{g}(z_t,\theta_0)$ is bounded, along with the assumption that $p(z_t|z_{t-k})$ is bounded w.r.t. z_{t-k} and that $\Gamma(z_{t-k})$ is integrable, for $k \in \{0,\dots,t-1\}$.

Now inserting this in Equation 26 we get:

$$\begin{aligned}\mathbb{E}\big[G(z_0,\theta_0)|T\big] &= \frac{\eta}{\tau}\sum_{t=0}^{T-1}\int_{\mathcal{Z}}\hat{g}(z_t,\theta_0)p_t(z_t)\mathrm{d}z_t + \mathcal{O}\Big(\eta^2\frac{T^3}{\tau^2}\Big)\\ &= \frac{\eta}{\tau}\sum_{t=0}^{T-1}\mathbb{E}_{z\sim p_t}\big(\hat{g}(z,\theta)\big) + \mathcal{O}\Big(\eta^2\frac{T^3}{\tau^2}\Big),\end{aligned} \tag{38}$$

where we have defined $p_t(z_t) = \int_{\mathcal{Z}}\cdots\int_{\mathcal{Z}}\prod_{k=0}^{t-1}p(z_{t-k}|z_{t-k-1},\theta_0)\mathrm{d}z_1\ldots\mathrm{d}z_{t-1}$.

Step 3: We now use the assumption that p_t converges weakly to our desired distribution p, with a convergence rate that is $\mathcal{O}(\frac{1}{t^2})$, to complete the proof.

Weak convergence, with the assumed rate, implies that for all bounded, measurable functions f, $|\mathbb{E}_{p_t}(f) - \mathbb{E}_p(f)| \leq \frac{M^*}{t^2}$, where M^* is some positive constant. By assumption, the gradients are measurable and bounded, so we have:

$$\begin{aligned}\frac{\eta}{\tau}\sum_{t=0}^{T-1}\mathbb{E}_{z\sim p_t}\big(\hat{g}(z,\theta)\big) &= \frac{\eta}{\tau}\sum_{t=s}^{T-1}\mathbb{E}_{z\sim p_t}\big(\hat{g}(z,\theta)\big) + \frac{\eta}{\tau}\sum_{t=0}^{s-1}\mathbb{E}_{z\sim p_t}\big(\hat{g}(z,\theta)\big)\\ &= \frac{\eta}{\tau}\sum_{t=s}^{T-1}\mathbb{E}_{z\sim p}\big(\hat{g}(z,\theta)\big) + \frac{\eta}{\tau}\sum_{t=s}^{T-1}\mathcal{O}\Big(\frac{M^*}{t^2}\Big) + \frac{\eta}{\tau}\sum_{t=0}^{s-1}\mathbb{E}_{z\sim p_t}\big(\hat{g}(z,\theta)\big)\\ &= \frac{T-s}{\tau}\eta\mathbb{E}_{z\sim p}\big(\hat{g}(z,\theta)\big) + \epsilon(\mathrm{T},\mathrm{s},\eta),\end{aligned} \tag{39}$$

where $\epsilon = \frac{\eta}{\tau}\sum_{t=s}^{T-1}\mathcal{O}\Big(\frac{M^*}{t^2}\Big) + \frac{\eta}{\tau}\sum_{t=0}^{s-1}\mathbb{E}_{z\sim p_t}\big(\hat{g}(z,\theta)\big)$ is the bias term. When T is stochastic we can split up the sum in this way, using the deterministic variable s, because we originally conditioned on the fact that $T \geq s$ at the start of the proof. We now bound the bias. Observe that:

$$|\epsilon(T,s,\eta)| \leq \mathcal{O}\Big(\frac{\eta}{\tau s}\Big) + \mathcal{O}\Big(\frac{\eta s}{\tau}\Big) = \mathcal{O}\Big(\frac{\eta s}{\tau}\Big), \tag{40}$$

where, for the inequality, we have used the fact that $\sum_{t=s}^{\infty}\frac{1}{t^2} \leq \frac{1}{s}$, and that $\hat{g}$ is bounded. For the last equality one need simply note that $s > 1$. Lastly, observe that $\frac{T-s}{\tau} = \frac{T}{\tau} - \frac{s}{\tau}$, so the second term of this bias gets absorbed in the $\mathcal{O}\big(\frac{\eta s}{T}\big)$ term.

Putting this together with the previous steps, we get:

$$\mathbb{E}\big[G(z_0,\theta_0)\big|T\big] = \eta\frac{T}{\tau}\mathbb{E}_{z\sim p}\big(\hat{g}(z,\theta)\big) + \mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}\Big(\eta^2\frac{T^3}{\tau^2}\Big). \tag{41}$$

And, finally, taking the expected value w.r.t. phase length yields

$$\begin{aligned}\mathbb{E}\big[G(z_0,\theta_0)|T_s=1\big] &= \eta\frac{\mathbb{E}(T|T_s=1)}{\tau}\mathbb{E}_{z\sim p}\big(\hat{g}(z,\theta)\big) + \mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}\Big(\eta^2\frac{\mathbb{E}(T^3|T_s=1)}{\tau^2}\Big)\\ &= \eta\mathbb{E}_{z\sim p}\big(\hat{g}(z,\theta)\big) + \mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}\Big(\eta^2\tau\Big),\end{aligned} \tag{42}$$

where we have used the results on the moments of T from §B.5. We note there that the $\mathcal{O}(\eta s/\tau)$ term from the preliminary on stochastic phase length gets absorbed in the first bias term, for the stochastic phase length, yielding a convergence rate that is of the same order as with deterministic phase lengths.

If we choose $\frac{1}{\eta}$ to be very large, τ to be less large, and s to be less, but still sufficiently, large, we find that we can make the sum of the gradient steps during each phase arbitrarily close to being unbiased. For example, the scaling proposed in the theorem would be sufficient to achieve this. □

B.2. Assumptions of Proof of Theorem 3.4

Finally, we comment on the validity of the assumption that $\Gamma(x)$ is integrable, and that the Markov chain exhibits weak convergence that is order $\frac{1}{t^2}$. The other assumptions we believe to be fairly mild restrictions on the regularity of the system.

For the assumptions on $\Gamma(x)$, we simply note that these will hold in cases where the functions involved are well-defined and $\mathcal{Z}$ is finite.

While we cannot prove that convergence to the stationary distribution is $\mathcal{O}(\frac{1}{t^2})$ for every possible Markov chain we believe this holds for many Markov chains of interest. In particular, it holds for every finite state-space (finite $\mathcal{Z}$) Markov chain that is both irreducible and aperiodic. Every Markov chain satisfying these three desiderata (irreducible, aperiodic, finite state-space) also satisfies the following convergence rate result (Rosenthal, 1995):

$$||p_t - p_\infty||_{\text{tv}} \leq Ct^{l-1}\lambda^{t-l+1} \tag{43}$$

where $||\cdot||_{\text{tv}}$ is the total variation distance for probability measures, λ is the second largest eigenvalue of the transition matrix for the Markov chain, and satisfies $\lambda < 1$, C is a positive constant, and l is the size of the largest Jordan block of the transition matrix. First, observe that convergence in total variation distance implies weak convergence. Second, this rate of convergence, which we define to be $r_1(t)$, is easily shown to be faster than $\mathcal{O}(\frac{1}{t^2})$ = $r_2(t)$. To do so, it suffices to show that $\frac{r_1(t)}{r_2(t)} \to 0$. The ratio can be arranged as follows:

$$\begin{aligned}\frac{r_1(t)}{r_2(t)} &= C_0 t^{l+1}\lambda^{t-l+1} \\ &= C_0 \frac{t^{l+1}}{\lambda^{l-t-1}}.\end{aligned} \tag{44}$$

Both numerator and denominator converge to ∞ as $t \to \infty$, and both are continuously differentiable $l+1$ times, so we can apply L'Hopital's rule. Applying this rule and differentiating the top and bottom expressions $l+1$ times gives:

$$\lim_{t\to\infty} \frac{r_1(t)}{r_2(t)} = \lim_{t\to\infty} C_0 \frac{(l+1)!}{\lambda^{l-t-1}\big(-\ln(\lambda)\big)^{l+1}} = 0, \tag{45}$$

where the last equality follows because $\lambda < 1$.

Remark: We note that we have used the rather unsophisticated approach of simply Taylor expanding the parameter values during a wake/sleep phase around θ_0, the value at the start of that phase, yielding a slowly converging error term. We expect that with more sophisticated methods, for example those outlined in (Yin & Zhang, 2005), one might be able to establish a faster converging error rate.

Remark 1: The covariance requires an intractable integral to evaluate, so we once again appeal to MC sampling w.r.t v and h to evaluate it.

Remark 2: A minor complication with the above is that one requires more than one MC sample, w.r.t v and h, to approximate the covariance function and, as such, one cannot use single sample MC sampling to estimate this gradient approximation.

B.3. Multivariate Mean Value Theorem

In this section we include the multivariate mean value theorem as it features heavily in the proof of Theorem 3.4.

Theorem B.2. *Assume $f : \mathbb{R}^d \mapsto \mathbb{R}$ is a differentiable function. Then*

$$f(x+h) = f(x) + \nabla f(\xi) \cdot h \tag{46}$$

where $x, h \in \mathbb{R}^d$, and ξ is a point on the line between x and $x+h$.

We omit the proof as it can be found in any standard textbook. The following lemma follows from the above, and is used throughout the proof of Theorem 3.4.

Lemma B.3. *Assume that f is L-Lipschitz (gradients of f are bounded by L), and that $h \leq M \; \forall \, i$. Then it follows from the Mean Value Theorem that*

$$f(x + \eta h) = f(x) + \mathcal{O}(\eta) \tag{47}$$

B.4. Convergence Rate of Error Induced by Stochastic Phase Lengths

We wish to derive the rate at which this term converges to zero:

$$\big(\mathbb{E}[G(z_0, \theta)|T_s = 0] - \mathbb{E}[G(z_0, \theta)|T_s = 1]\big)p(T_s = 0). \tag{48}$$

We first consider the expression inside the brackets. Using the assumption that the gradients are bounded we get the following

$$\big(\mathbb{E}[G(z_0, \theta)|T_s = 0] - \mathbb{E}[G(z_0, \theta)|T_s = 1]\big) \leq \frac{A_1 s\eta}{\tau} + \frac{A_2 \mathbb{E}(T|T_s = 1)\eta}{\tau} = \mathcal{O}(\eta), \tag{49}$$

where A_1 and A_2 are constants. For the second term, we note that $p(T_s = 0) = 1 - p(T_s = 1) = 1 - p(\omega = 0)^s = 1 - (1 - \mu)^s$. By the definitions $s = \tau^m$ and $\mu = \frac{1}{\tau}$, we get

$$p(T_s = 0) = 1 - \Big(1 - \frac{1}{\tau}\Big)^{\tau^m}. \tag{50}$$

Because 1 is a constant, $p(T_s = 0)$ goes to zero at the same rate that $(1 - \frac{1}{\tau})^{\tau^m}$ goes to 1. We will now derive this rate.

$$\begin{aligned}
\Big(1 - \frac{1}{\tau}\Big)^s &= \exp\Big(s \log\Big[1 - \frac{1}{\tau}\Big]\Big) = \exp\Big(-s\Big[\frac{1}{\tau} + \mathcal{O}\Big(\frac{1}{\tau^2}\Big)\Big]\Big) \\
&= \exp\Big(-\frac{s}{\tau}\Big)\exp\Big(-\mathcal{O}\Big(\frac{s}{\tau^2}\Big)\Big) = \Big[1 - \frac{s}{\tau} + \mathcal{O}\Big(\frac{s^2}{\tau^2}\Big)\Big]\Big[1 - \frac{s}{\tau^2} + \mathcal{O}\Big(\frac{s^2}{\tau^4}\Big)\Big] \\
&= 1 - \frac{s}{\tau} + K_0(s, \tau).
\end{aligned} \tag{51}$$

Here, we have expanded the inner logarithm about 1, given that $1 - \frac{1}{\tau}$ goes to 1, and the exponential about 0, given that $s \log\Big[1 - \frac{1}{\tau}\Big]$ goes to zero (because $s = \tau^m$ by definition). K_0 is thus a function purely of higher order terms than $\frac{s}{\tau}$. Thus, with leading order, $p(T_s = 0)$ goes to zero with rate $\frac{s}{\tau}$.

Combining this result with the previous one gives that Equation 48 is order $\mathcal{O}\big(\frac{\eta s}{\tau}\big)$.

B.5. Moments of T

Assume that for every time step on a discrete timeline one evaluates the Bernoulli random variable ω with parameter $\mu << 1$. Assuming that, at time t_0, $\omega = 1$, one can define the random variable T that is the number of time steps until the next observance of $\omega = 1$. This is precisely the way that the stochastic phases are defined in the theorem. One can solve recursively for the moments of this random variable, using the law of total expectation and conditioning on the event that $T = 1$. Define $\tau = \frac{1}{\mu}$. Then the first three moments are found to be:

$$\begin{aligned}
\mathbb{E}(T) &= \tau \\
\mathbb{E}(T^2) &= 2\tau^2 - \tau \\
\mathbb{E}(T^3) &= 6\tau^3 - 6\tau^2 - \tau.
\end{aligned} \tag{52}$$

In the proof of Theorem 3.4, and it's deterministic analog (see §C), we are particularly interested in the moments conditioned on $T_s = 1$. Because the Bernoulli random variable that decides whether the phase ends is sampled independently at each step, we get the following relations:

$$\mathbb{E}(T|T_s = 1) = \mathbb{E}(T + s) = \tau + s \tag{53}$$
$$\mathbb{E}(T^2|T_s = 1) = \mathbb{E}[(T + s)^2] = \tau^2 + 2\tau s + s^2 \tag{54}$$
$$\mathbb{E}(T^3|T_s = 1) = \mathbb{E}[(T + s)^3] = \tau^3 + 3\tau^2 s + 3\tau s^2 + s \tag{55}$$

Importantly, these first and third moments are of leading order 1, 2 and 3 w.r.t τ. These observations are used in the proof of Theorem 3.4 and Theorem C.1.

C. Analog of Theorem 3.4 for Deterministic Equilibrium Dynamics

Theorem C.1. *Consider a single phase of equilibrium-based learning where the phase is either fixed to value τ or ends stochastically at each step with small probability $\frac{1}{\tau}$ and, instead of waiting until the end of the phase to do an update with learning rate η, an update is performed at every step of the equilibrium dynamics with learning rate $\frac{\eta}{\tau}$. Assume that equilibrium dynamics evolve for fixed θ according to the homogeneous discrete time dynamical system given by the map $F(z, \theta)$. Further assume that $\hat{g}(\theta)$ is differentiable and bounded with bounded partial derivatives for all network state values, and that F is differentiable w.r.t. z and θ, also with bounded partial derivatives. Finally, assume that the equilibrium dynamics converge to a fixed point at a rate at least as fast as $\mathcal{O}(\frac{1}{t^2})$. Then the bias of the learning updates are of the following order:*

$$\mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}(\eta^2), \tag{56}$$

where $s = \tau^m$ for $0 < m < 1$ and $\mathcal{O}$ is big-O notation.

Proof. The proof has the same structure as that of Theorem 3.4 and, accordingly, proceeds in three steps preceded by a preliminary comment on stochastic phase length.

Comment on Stochastic Phase Length Exactly the same as the comment in the proof of Theorem 3.4.

Step 1: Exactly the same as "step 1" from the proof of Theorem 3.4.

Step 2: This step uses a proof by induction. Following step 1, we have

$$\mathbb{E}\big[G(z_0, \theta_0)|T\big] = \frac{\eta}{\tau}\sum_{t=0}^{T-1} \hat{g}(\tilde{z}_t, \theta_0) + \mathcal{O}\Big(\eta^2 \frac{T^2}{\tau^2}\Big), \tag{57}$$

where $\{\tilde{z}_t\}_{t=0}^{t=T-1}$ are the dynamics perturbed by the always learning updates to θ. We wish to demonstrate that this represents merely a negligible perturbation of the homogeneous dynamics $\{z_t\}_{t=0}^{t=T-1}$. Let us assume that $F(z, \theta)$ is differentiable w.r.t. z and θ and that all partial derivatives are bounded by L (thus, that it is L-Lipschitz). Then we have

$$\begin{aligned}
&\tilde{z}_0 = z_0 \\
&\tilde{z}_1 = F(\tilde{z}_0, \theta_0) = F(z_0, \theta_0) = z_1 \\
&\tilde{z}_2 = F(\tilde{z}_1, \theta_1) = F\Big(z_1, \theta_0 + \frac{\eta}{\tau}\hat{g}(z_0, \theta_0)\Big) = F(z_1, \theta_0) + \mathcal{O}\Big(\frac{\eta}{\tau}\Big) = z_2 + \mathcal{O}\Big(\frac{\eta}{\tau}\Big),
\end{aligned} \tag{58}$$

where we used the multivariate mean value theorem corollary Lemma B.3 and the assumption that $\hat{g}$ is bounded for the third equality in the final line. The final line of Equation 58 will be the "base case" in our proof by induction. We make the following induction hypothesis:

$$\tilde{z}_t = z_t + \mathcal{O}\Big(\frac{(t-1)\eta}{\tau}\Big). \tag{59}$$

We now complete the proof by induction—via the "induction step" (proof of the $t+1$ case)—to show that the equality given in Equation 59 holds for arbitrary t:

$$\begin{aligned}\tilde{z}_{t+1} = F(\tilde{z}_t, \theta_t) &= F\bigg(z_t + \mathcal{O}\Big(\frac{(t-1)\eta}{\tau}\Big), \theta_0 + \frac{\eta}{\tau}\sum_{k=0}^{t-1}\hat{g}(\tilde{z}_k, \theta_k)\bigg) \\ &= F(z_t, \theta_0) + \mathcal{O}\Big(\frac{t\eta}{\tau}\Big).\end{aligned} \tag{60}$$

In the second equality of the first line we used the induction hypothesis to relate $\tilde{z}_t$ to z_t; we also used the definition of the parameter dynamics. For the final line we again used the assumption that F is differentiable and Lipschitz, and the assumption that $\hat{g}$ is bounded. We also used the fact that the $\mathcal{O}\Big(\frac{(t-1)\eta}{\tau}\Big)$ term is also $\mathcal{O}\Big(\frac{t\eta}{\tau}\Big)$.

Assuming $\hat{g}$ has bounded partial derivatives w.r.t. z and applying Lemma B.3 we get:

$$\hat{g}(\tilde{z}_t, \theta_0) = \hat{g}(z_t, \theta_0) + \mathcal{O}\Big(\frac{t\eta}{\tau}\Big), \tag{61}$$

so that, applying these results to Equation 57, we get:

$$\begin{aligned}\mathbb{E}\big[G(z_0, \theta_0)|T\big] &= \frac{\eta}{\tau}\sum_{t=0}^{T-1}\hat{g}(z_t, \theta_0) + \frac{\eta}{\tau}\sum_{t=0}^{T-1}\mathcal{O}\Big(\frac{t\eta}{\tau}\Big) + \mathcal{O}\Big(\eta^2\frac{T^2}{\tau^2}\Big) \\ &= \frac{\eta}{\tau}\sum_{t=0}^{T-1}\hat{g}(z_t, \theta_0) + \mathcal{O}\Big(\eta^2\frac{T^2}{\tau^2}\Big).\end{aligned} \tag{62}$$

Step 3: This step proceeds in essentially the same way as step 3 of the proof of the Markov dynamics version. We leverage the assumption that z_t converges to a fixed point of z_∞ with a convergence rate that is $\mathcal{O}(\frac{1}{t^2})$:

$$\begin{aligned}\frac{\eta}{\tau}\sum_{t=0}^{T-1}\hat{g}(z_t, \theta) &= \frac{\eta}{\tau}\sum_{t=s}^{T-1}\hat{g}(z_t, \theta) + \frac{\eta}{\tau}\sum_{t=0}^{s-1}\hat{g}(z_t, \theta) \\ &= \frac{\eta}{\tau}\sum_{t=s}^{T-1}\hat{g}(z_\infty, \theta) + \frac{\eta}{\tau}\sum_{t=s}^{T-1}\mathcal{O}\Big(\frac{1}{t^2}\Big) + \frac{\eta}{\tau}\sum_{t=0}^{s-1}\hat{g}(z_t, \theta) \\ &= \frac{T-s}{\tau}\eta\hat{g}(z_\infty, \theta) + \epsilon(\mathrm{T}, \mathrm{s}, \eta).\end{aligned} \tag{63}$$

On line two of the above, we used our assumptions on the differentiability and boundedness of the partial derivatives of $\hat{g}$ w.r.t. z, to apply Lemma B.3 yet again. We also used the variable s defined in the theorem, and defined $\epsilon = \frac{\eta}{\tau}\sum_{t=s}^{T-1}\mathcal{O}\Big(\frac{1}{t^2}\Big) + \frac{\eta}{\tau}\sum_{t=0}^{s-1}\hat{g}(z, \theta)$ as the bias term. Importantly, we have also taken s large enough so that, for $t \geq s$, z_t is sufficiently close to z_∞ to allow us to expand $\hat{g}$ around z_∞. We now bound the bias term.

Observe that:

$$|\epsilon(T, s, \eta)| \leq \mathcal{O}\Big(\frac{\eta}{\tau s}\Big) + \mathcal{O}\Big(\frac{\eta s}{\tau}\Big) = \mathcal{O}\Big(\frac{\eta s}{\tau}\Big), \tag{64}$$

where, for the inequality, we have used the fact that $\sum_{t=s}^{\infty}\frac{1}{t^2} \leq \frac{1}{s}$, and that $\hat{g}$ is bounded. For the last equality one need simply note that $s > 1$. Lastly, observe that $\frac{T-s}{\tau} = \frac{T}{\tau} - \frac{s}{\tau}$, so the second term of this bias gets absorbed in the $\mathcal{O}\big(\frac{\eta s}{T}\big)$ term.

Putting this together with the previous steps, we get:

$$\mathbb{E}\big[G(z_0,\theta_0)\big|T\big] = \eta\frac{T}{\tau}\hat{g}(z_\infty,\theta) + \mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}\Big(\eta^2\frac{T^2}{\tau^2}\Big). \tag{65}$$

And, finally, taking the expected value w.r.t. phase length yields

$$\begin{aligned}\mathbb{E}\big[G(z_0,\theta_0)\big|T_s=1\big] &= \eta\frac{\mathbb{E}(T|T_s=1)}{\tau}\hat{g}(z_\infty,\theta) + \mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}\Big(\eta^2\frac{\mathbb{E}(T^2|T_s=1)}{\tau^2}\Big)\\ &= \eta\hat{g}(z_\infty,\theta) + \mathcal{O}\Big(\frac{\eta s}{\tau}\Big) + \mathcal{O}(\eta^2),\end{aligned} \tag{66}$$

where we have used the results on the moments of T from §B.5. If we choose η to be large, τ to be large, and s to be less large than τ, but still sufficiently large for convergence of the dynamics, we find that we can make the sum of the gradient steps during each phase arbitrarily close to being unbiased. For example, the scaling proposed in the remark after Theorem 3.4 would be sufficient to achieve this.

□

Remark on Difference Between Stochastic and Deterministic Dynamics Interestingly, we find that the bias for deterministic dynamics is lower than the bias for stochastic dynamics. Furthermore, for deterministic dynamics we don't need the learning rate to be of lower order than τ or s. The reason for these phenomena is the change from a second bias term of $\mathcal{O}(\eta^2 T)$ for stochastic dynamics to $\mathcal{O}(\eta^2)$ for deterministic dynamics. Mathematically, this change occurs because of the difference in how one removes the inhomogeneity in the dynamics for the stochastic versus the deterministic case: one has to expand a *product* of functions (inside an integral) in the stochastic case, compared to expanding within a *composition* of functions in the deterministic case.

It is an open question whether this difference in bias between deterministic and stochastic dynamics is a fundamental property or is purely a consequence of the techniques used in the proof. In particular, we wonder whether the stochastic bias could be reduced to that of the deterministic dynamics if one were to use techniques from operator theory (Lasota & Mackey, 1998; Yin & Zhang, 2005). We leave this to future work.

D. Algorithm Performance as a Function of Positive Phase Probability for MNIST

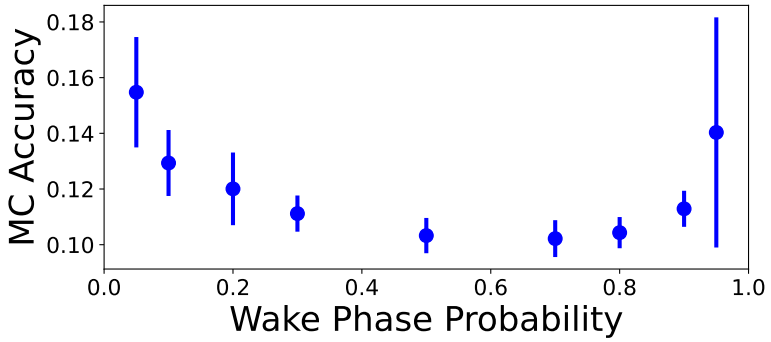

Figure 4. Same as Fig.3.B except only the AoL values are shown, and the model is an RBM trained on binarized MNIST. Observe that the minimal value of b is higher than 0.5. For this experiment, the RBM was trained on the concatenation of the images and their labels and the error plotted is classification error calculated, for a given image, by presenting the image without the label and selecting the predicted image label as the one with highest mean over a set of samples from the RBM. The RBM had 794 visible units and 128 hidden, and was trained with a learning rate of 0.005.

E. Experimental Details

General: For the experiments in Fig.2, and all distinct models, learning rates were selected by performing a line search over ten values, equally spaced on a log10 scale, to find the learning rate with lowest end-of-training training error. The learning rates in Fig.3 were simply set to 0.05 and 0.0025, for ISD AoL and ISD respectively, as the goal of this figure was not to compare algorithms and these learning rates were found to yield robust learning on a reasonable time frame. Network sizes were fixed to provide easy comparison across models. Data was divided into segregated train and test sets for the MNIST experiments, but the test set was only used with the forward-forward algorithm. The BAS dataset is small and the entire, ground-truth data was trained on; therefore there was no test/training split in this case. The code used for the experiments can be found at the following Github repository: https://github.com/zek3r/ICML2023.

RBM Parameters: For all experiments on the BAS dataset we trained a RBM with 16 hidden units and 16 visible units. The max and min values for the learning rate line search for Fig.2 were 0.04 and 0.001. Phase length was defined $T = 100$ for ISD, 100 for ISD AoL with fixed phase length, and was assigned a mean of 150 for the random phase length version. All networks were trained using vanilla SGD with no regularization. Initialization of all parameters was to white noise with standard deviation of 0.01. All training runs in Fig.2 were 10^5 steps long, except for CD1 which was 10^7 gradient steps long (and failed to converge). Here, a step is defined as a gradient step for CDK and regular ISD, and a phase for ISD AoL and ISD AoL Random T. The reason that phase was used instead of gradient step for these latter two algorithms is that these algorithms performed a gradient step at every step of MCMC sampling but, as mentioned in Section 3.2, learning rates were scaled by the mean phase length so that the sum of gradient steps taken during a phase was approximately equal to the single gradient step taken in regular ISD.

FF Parameters: With the forward-forward algorithm, we trained a network with two hidden layers of 500 units each. ADAM was used for both the standard algorithm and the ISD algorithm. The max and min values for the learning rate line search were 0.05 and 0.002. All training runs in Fig.2, for all algorithms, were 120000 gradient steps long.

F. Ethics

Energy Consumption: Given the scale of the experiments, we believe the energy consumption of this project was negligible. Moreover, the experiment was performed on an electrical grid that primarily uses hydro-electricity from long-established generating stations, meaning that the greenhouse gas emissions were likely close to zero.

Other Considerations: The main application of this work is to build basic scientific knowledge to aid computational neuroscience modelling and neuromorphic hardware development. As such, societal implications could eventually encompass clinical neuroscientific interventions or reductions in the energy consumption of deep learning methods, which we believe are largely positive. However, the commodification and weaponization of AI technology is already having negative impacts on society (e.g. use of facial recognition for discrimination, or the accentuating of wealth inequality and job losses). The authors are doing their best to stay abreast of these problems, and take steps to help combat them. For example, to combat commodification and centralization of wealth by making work publicly available.