Neural spiking for causal inference

Benjamin James Lansdell[1,+] and Konrad Paul Kording[1]

[1] *Department of Bioengineering, University of Pennsylvania, PA, USA*
[+] lansdell@seas.upenn.edu

Keywords: causal inference, reinforcement learning, reward-modulated learning, plasticity, noise correlations

Abstract

When a neuron is driven beyond its threshold it spikes, and the fact that it does not communicate its continuous membrane potential is usually seen as a computational liability. Here we show that this spiking mechanism allows neurons to produce an unbiased estimate of their causal influence, and a way of approximating gradient descent learning. Importantly, neither activity of upstream neurons, which act as confounders, nor downstream non-linearities bias the results. By introducing a local discontinuity with respect to their input drive, we show how spiking enables neurons to solve causal estimation and learning problems.

Introduction

Most nervous systems communicate and process information with spiking neural networks. Yet machine learning mostly uses artificial neural networks with continuous activities. Computationally, despite a lot of recent progress [58, 71, 8, 9, 52], it remains challenging to create spiking neural networks that perform comparably to artificial networks. Instead, spiking is generally seen as a disadvantage – it is difficult to propagate gradients through a discontinuity. This disparity between spiking and artificial networks raises the question, what are the computational benefits of spiking?

There are, of course, pragmatic reasons for spiking: spiking may be more energy efficient [1, 70], spiking allows for reliable transmission over long distances [59], and spike timing codes may allow for more transmission bandwidth [72]. Yet, despite these ideas, we may still wonder if there are computational benefits of spikes that balance the apparent disparity in the abilities of spiking and artificial networks.

A key computational problem in both biological and artificial settings is the credit assignment problem. When performance is sub-optimal, the brain needs to decide which activities or weights should be different. Credit assignment is fundamentally a causal estimation problem – which neurons are responsible for the bad performance, and not just correlated with bad performance? Solving such problems is difficult because of confounding: if a neuron of interest was active during bad performance it could be that it was responsible, or it could be that another neuron whose activity is correlated with the neuron of interest was responsible. In general, confounding happens if a variable affects both another variable of interest and the performance. Even when a fixed stimulus is presented repeatedly, neurons exhibit complicated correlation structures [5, 15, 37, 7, 78] which confounds a neuron's estimate of its causal effect. This prompts us to ask how neurons can solve causal estimation problems.

The gold-standard approach to causal inference is randomized perturbation. If a neuron occasionally adds an extra spike (or removes one), it could readily estimate its causal effect by correlating the extra spikes with performance. Such perturbations come at a cost, since the noise can degrade performance. This class of learning methods has been extensively explored [11, 20, 21, 44, 49, 77, 64]. However the learning speed of a network of neurons using random perturbations scales poorly with the number of neurons N (as $\mathcal{O}(1/N)$ [64, 20, 60]). Further, in general it is not clear how a neuron may know its own noise level (though see the birdsong learning literature for one plausible case [20], or other proposals about how it may be approximated in some cases [44, 30]). Thus we may wonder if neurons estimate their causal effect without random perturbations.

How could neurons estimate their causal effect? Over a short time window, a neuron either does or does not spike. Comparing the average reward when the neuron spikes versus does not spike gives a confounded estimate of the neuron's effect. Because neurons are correlated, spiking is associated with a different network state than not-spiking. This difference in network state that may account for an observed difference in reward, not specifically the neuron's activity. Simple correlations will give wrong causal estimates.

However, the story is different when comparing the average reward in times when the neuron *barely* spikes versus when it *almost* spikes. The difference in the state of the network in the barely spikes versus almost spikes case is negligible, the only difference is the fact that in one case the neuron spiked and in the other case the neuron did not. Any difference in observed reward can therefore *only* be attributed to the neuron's activity. The precise mathematical reason is that the drive satisfies the back-door criterion relative to the spike-reward relation [56]. In this way the spiking discontinuity may allow neurons to estimate their causal effect.

Here we propose the spiking discontinuity is used by a neuron to efficiently estimate its causal effect. Once a neuron can estimate its causal effect, it can use this knowledge to calculate gradients and adjust its synaptic strengths. We show that this idea suggests a learning rule that allows a network of neurons to learn to maximize reward. We demonstrate the rule in simple models. The discontinuity-based method provides a novel and plausible account of how neurons learn their causal effect.

Results

A neuron's causal effect

In causal inference, the causal effect can be understood as the expected difference in an outcome R when a treatment H is randomly, independently, assigned – a primary quantity of interest in a randomized control trial. This definition can be made precise in the context of a causal graphical model [56] (Figure 1A, see Methods; or with the potential outcomes framework [3]). Notationally, the causal effect can be defined as:

$$\beta = \mathbb{E}(R|\mathrm{do}(H = 1)) - \mathbb{E}(R|\mathrm{do}(H = 0)),$$

where do represents the do-operator, notation for an intervention. A causal effect can be measured through randomization, but sometimes can also be estimated without randomization by additionally observing the right variables to remove confounding. One such criterion is known as the back-door criterion [56]. In essence, outcomes are compared between treatment and non-treatment groups, when these additional variables satisfying the backdoor criterion are held fixed.

In a neural network setting, the causal effect of a neuron on reward can be seen, loosely, as the change in reward as a result of a change in the neuron's activity made through randomization. Alternatively, it could be estimated by invoking the backdoor criterion and estimating the change in reward for a change in the

neuron's activity, when other neurons not downstream of that neuron are held fixed. In this spiking (binary) case the causal effect can be seen as a type of finite difference approximation of the partial derivative. That is, let H_i be the spiking indicator variable of a neuron i over a time window of length T, let S_i be a filtered version of the spiking activity, which contributes to determining the reward signal R. The neuron would like to estimate the effect of its spiking on R:

$$\beta_i := \mathbb{E}(R|\mathrm{do}(H_i = 1)) - \mathbb{E}(R|\mathrm{do}(H_i = 0)) \approx \mathbb{E}\left(\frac{\partial R}{\partial S_i}\right).$$

It can be shown that this quantity, under certain assumptions, does indeed approximate the reward gradient $\frac{\partial R}{\partial S_i}$ (see Methods). This establishes a link between causal inference and gradient-based learning, and suggests methods from causal inference may provide efficient algorithms to estimate reward gradients.

Estimating a neuron's causal effect using the spiking discontinuity

A discontinuity can be used to estimate a causal effect. We will refer it here as the Spiking Discontinuity Estimator (SDE). This is equivalent to the regression discontinuity design (RDD) approach which is popular in economics [34, 2]. For reasons outlined above, estimating the size of the discontinuity in an outcome at the threshold gives a measure of causal effect. As SDE derives from RDD it is valid when [34, 36]: first, whether a neuron spikes in an interval is not affected by the neuron's spikes in that interval – there is no feedback over the interval; second, the spiking threshold is determined by intrinsic properties of the neuron, and not tuned for a specific drive; and third, noise in the network (and the fact that there are many presynaptic neurons) smooths out any discontinuities that may be a result of spiking in presynaptic inputs and thus, considering reward as a function of presynaptic drive, the neuron's spiking is the only discontinuity observed. Under these assumptions neurons can produce unbiased estimates of causal effects using SDE.

For a neuron to apply SDE-based learning, it must track how close it is to spiking, whether it spiked or not, and observe the reward signal. More specifically, neurons spike when their maximal drive Z_i exceeds a threshold and then can receive feedback or a reward signal R through neuromodulator signals (Figure 1A). Then the comparison in reward between time periods when a neuron almost reaches its firing threshold to moments when it just reaches its threshold allows an SDE estimate of its own causal effect (Figure 1B,C,D,E).

To implement SDE, a neuron can estimate a piece-wise linear model of the reward function at time periods when its inputs place it close to threshold:

$$R = \gamma_i + \beta_i H_i + [\alpha_{ri} H_i + \alpha_{li}(1 - H_i)](Z_i - \mu). \tag{1}$$

Here H_i is neuron i's spiking indicator function, γ_i, α_{li} and α_{ri} are the slopes that correct biases that would otherwise occur from having a finite bandwidth, Z_i is the maximum neural drive to the neuron over a short time period, and β_i represents the causal effect of neuron i's spiking over a fixed time window of period T. The neural drive used here is the leaky, integrated input to the neuron, that obeys the same dynamics as the membrane potential except without a reset mechanism. By tracking the maximum drive attained over a short time period, marginally super-threshold inputs can be distinguished from well-above-threshold inputs, as required to apply SDE. Proposed physiological implementations of this model are described in the discussion.

To demonstrate that a neuron can use SDE to estimate causal effects here we analyze a simple two neuron network obeying leaky integrate-and-fire (LIF) dynamics. The neurons receive an input signal x with added noise, correlated with coefficient c. Each neuron weighs the noisy input by w_i. The correlation in input noise induces a correlation in the output spike trains of the two neurons [66], thereby introducing

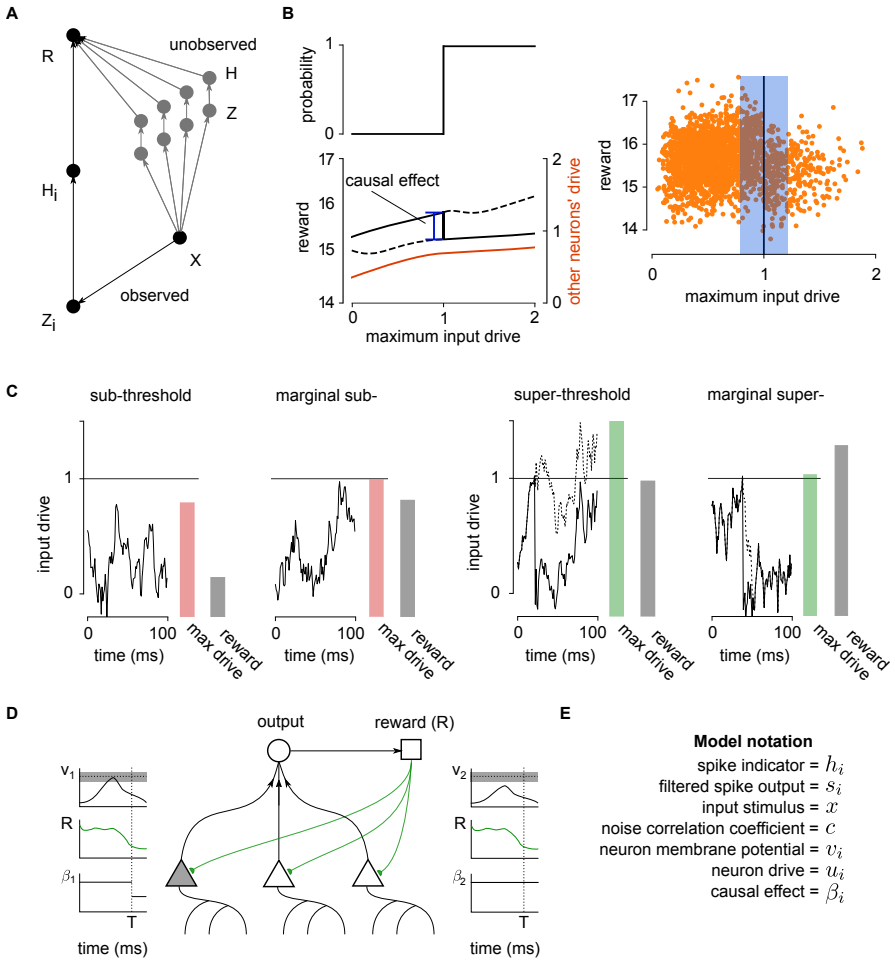

Figure 1: **Using the spiking discontinuity for neural learning.** (A) Graphical model describing neural network. Neuron H_i receives input X, which contributes to drive Z_i. If drive is above the spiking threshold, then H_i is active. The activity contributes to reward R. Though not shown, this relationship may be mediated through downstream layers of a neural network, and complicated interactions with the environment. From neuron H_i's perspective, the activity of the other neurons H which also contribute to R is unobserved. (B) The reward may be tightly correlated with other neurons' activity, which act as confounders. However any discontinuity in reward at the neuron's spiking threshold can only be attributed to that neuron. The discontinuity at the threshold is thus a meaningful estimate of the causal effect (left). The effect of a spike on a reward function can be determined by considering data when the neuron is driven to be just above or just below threshold (right). (C) This is judged by looking at the neural drive to the neuron over a short time period. Marginal sub- and super-threshold cases can be distinguished by considering the maximum drive throughout this period. (D) Schematic showing how SDE operates in network of neurons. Each neuron contributes to output, and observes a resulting reward signal. Learning takes place at end of windows of length T. Only neurons whose input drive brought it close to, or just above, threshold (gray bar in voltage traces; compare neuron 1 to 2) update their estimate of β. (E) Model notation.

confounding. The neural output determines a non-convex reward signal R. Most aspects of causal inference can be investigated in a simple model such as this [57], thus demonstrating that a neuron can estimate a causal effect with SDE in this simple case is an important first step to understanding how it can do so in a larger network.

Applying the SDE estimator shows that a neuron can estimate its causal effect (Figure 2A,B). To show how it removes confounding, we implement a simplified SDE estimator that considers only average difference in reward above and below threshold within a window of size p. When p is large this corresponds to the biased observed dependence estimator, while small p values approximate the SDE estimator and result in an unbiased estimate (Figure 2A). The window size p determines the variance of the estimator, as expected from theory [33]. Instead a locally linear SDE model, (1), can be used. This model is more robust to confounding (Figure 2B), allowing larger p values to be used. Thus the linear correction that is the basis of many RDD implementations [34] allows neurons to readily estimate their causal effect.

To investigate the robustness of the SDE estimator, we systematically vary the weights, w_i, of the network. SDE works better when activity is fluctuation-driven and at a lower firing rate (Figure 2C). Thus SDE is most applicable in irregular but synchronous activity regimes [12]. Over this range of network weights SDE is less biased than the observed dependence (Figure 2D).

An discontinuity-based learning rule

A canonical causal inference problem is learning – in order to update synaptic weights to improve performance a neuron needs to know its causal effect on that performance. The spiking discontinuity allows neurons to estimate their causal effect. We can show that this provides a rule for neurons to update their weights to maximize a reward. For gradient-based learning, to do this the neuron must relate the gradient $\frac{\partial \mathbb{E}(R)}{\partial w_i}$ to something involving the causal effect. One complication from a theoretical standpoint is that the distribution over which the above expectation is taken depends on the parameters w, meaning the derivative cannot be immediately transferred inside the expectation. Additional assumptions are needed to justify something like this. It can be shown (see Methods) that under the following assumptions:

1. The neural network parameters only affect the expected reward through their spiking activity, meaning that $\mathbb{E}(R|H)$ is independent of parameters $\mathbf{w}$.
2. The gradient term $\frac{\partial \mathbb{E}(H_i|H_{j\neq i})}{\partial w_i}$ is independent of $H_{j\neq i}$.
3. Neurons $H_{j\neq i}$ satisfy the backdoor criterion with respect to $H_i \rightarrow R$.

then:

$$\frac{\partial}{\partial w_i}\mathbb{E}(R) \approx \frac{\partial \mathbb{E}(H_i)}{\partial w_i}\beta_i. \tag{2}$$

These assumptions are not necessarily true for the dynamical networks considered here, and their validity must be tested empirically. They were shown to be reasonable over certain activity regimes in the simulations used here (Supplementary Material). Thus the estimate of causal effect can be used to update synaptic weights using a gradient-descent based rule.

To demonstrate how a neuron can learn β through SDE, we derive an online learning rule from the locally linear SDE model. The rule takes the form:

$$\Delta \mathbf{u}_i = \begin{cases} -\eta[\mathbf{u}_i^T\mathbf{a}_i - R]\mathbf{a}_i, & \theta \leq Z_i < \theta + p \text{ (just spikes)}; \\ -\eta[\mathbf{u}_i^T\mathbf{a}_i + R]\mathbf{a}_i, & \theta - p < Z_i < \theta \text{ (almost spikes)}, \end{cases}$$

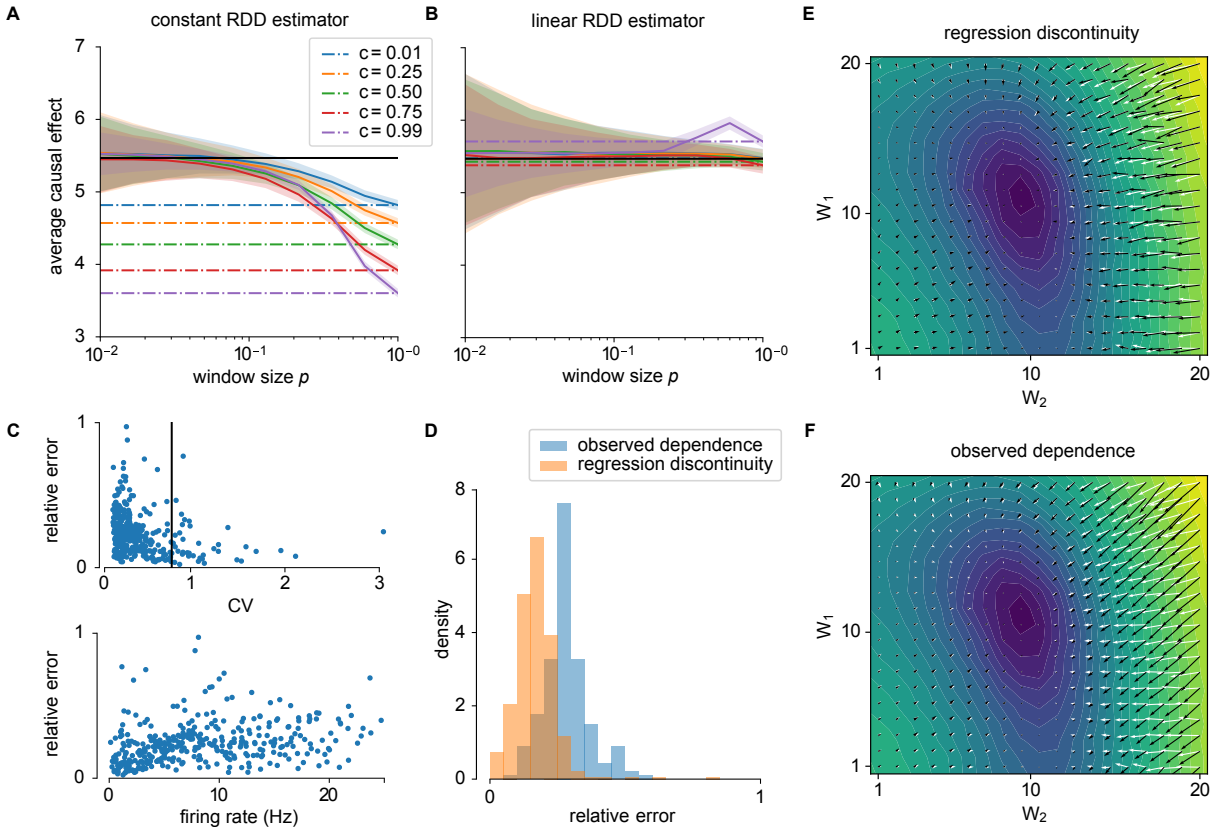

Figure 2: **Estimating reward gradient with SDE in two-neuron network.** (A) Estimates of causal effect (black line) using a constant SDE model (difference in mean reward when neuron is within a window p of threshold) reveals confounding for high p values and highly correlated activity. $p = 1$ represents the observed dependence, revealing the extent of confounding (dashed lines). Curves show mean plus/minus standard deviation over 50 simulations. (B) The linear SDE model is unbiased over larger window sizes and more highly correlated activity (high c). (C) Relative error in estimates of causal effect over a range of weights ($1 \leq w_i \leq 20$) show lower error with higher coefficient of variability (CV; top panel), and lower error with lower firing rate (bottom panel). (D) Over this range of weights, SDE estimates are less biased than just the naive observed dependence. (E,F) Approximation to the reward gradient overlaid on the expected reward landscape. The white vector field corresponds to the true gradient field, the black field correspond to the SDE (E) and OD (F) estimates. The observed dependence is biased by correlations between neuron 1 and 2 – changes in reward caused by neuron 1 are also attributed to neuron 2.

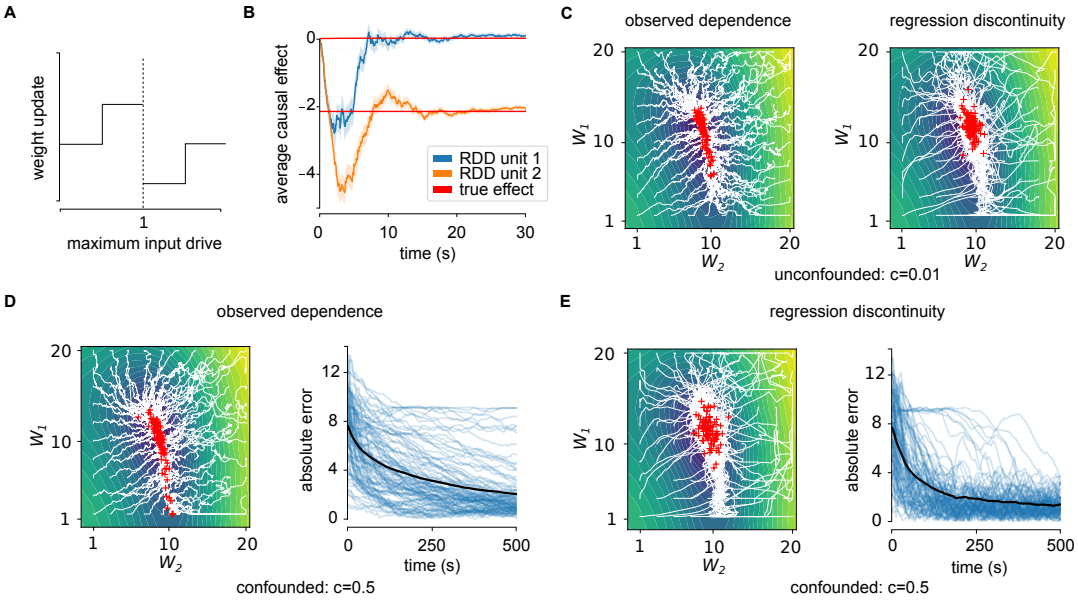

Figure 3: **Applying the discontinuity learning rule.** (A) Sign of SDE learning rule updates are based on whether neuron is driven marginally below or above threshold. (B) Applying rule to estimate β for two sample neurons shows convergence within 10s (red curves). Error bars represent standard error of the mean over 50 simulations. (C) Convergence of observed dependence (left) and SDE (right) learning rule to unconfounded network ($c = 0.01$). Observed dependence converges more directly to bottom of valley, while SDE trajectories have higher variance. (D,E) Convergence of observed dependence (D) and SDE (E) learning rule to confounded network ($c = 0.5$). Right panels: error as a function of time for individual traces (blue curves) and mean (black curve). With confounding learning based on observed dependence converges slowly or not at all, whereas SDE succeeds.

where $\mathbf{u}_i$ are the parameters of the linear model required to estimate β_i, η is a learning rate, and $\mathbf{a}_i$ are drive-dependent terms (see Methods). This rule can then be applied along with Equation (2) to update synaptic weights. The causal effect can be used to estimate $\frac{\partial R}{\partial w_i}$ (Figure 2E,F), and thus the SDE estimator may be used in a learning rule to update weights so as to maximize expected reward.

When applied to the toy network, the online learning rule (Figure 3A) estimates β over the course of seconds (Figure 3B). When the estimated β is then used to update weights to maximize expected reward in an unconfounded network (uncorrelated – $c = 0.01$), SDE-based learning exhibits higher variance than learning using the observed dependence. SDE-based learning exhibits trajectories that are initially meander while the estimate of β settles down (Figure 3C). When a confounded network (correlated – $c = 0.5$) is used SDE exhibits similar performance, while learning based on the observed dependence sometimes fails to converge due to the bias in gradient estimate. In this case SDE-based learning also converges faster than learning based on observed dependence (Figure 3D,E). Thus the SDE-based learning rule allows a network to be trained on the basis of confounded inputs.

Application to BCI learning

To demonstrate the behavior of SDE learning in a more realistic setting, we consider learning with intracortical brain-computer interfaces (BCIs) [19, 61, 44, 26, 51]. BCIs provide an excellent opportunity to test theories of learning and to understand how neurons solve causal inference problems. This is because in a BCI the exact mapping from neural activity to behavior and reward is known by construction [26] – the true causal effect of each neuron is known. Recording from neurons that directly determine BCI output as well as those that do not allows us to observe if and how neural populations distinguish a causal relationship to BCI output from a correlational one. Here we compare SDE-based learning and observed dependence-based learning to known behavioral results in a setting that involves correlations among neurons. We focus on single-unit BCIs [51], which map the activity of individual units to cursor motion. This is a meaningful test for SDE-based learning because the small numbers of units involved in the output function mean that a single spike has a measurable effect on the output.

We focus in particular on a type of BCI called a dual-control BCI, which requires a user control a BCI and simultaneously engage in motor control [6, 54, 50, 39] (Figure 4A). Dual-control BCIs engage primary motor units in a unique way: the units directly responsible for the cursor movement (henceforth control units) change their tuning and effective connectivity properties to control an output, while other observed units retain their association to motor control (Figure 4B) [50, 39]. In other words, control units modify their activity to perform a dual-control BCI task while other observed neurons, whose activity is correlated with the control neurons, do not. The issue of confounding is thus particularly relevant in dual-control BCIs.

We run simulations inspired by learning in a dual-control BCI task. Ten LIF neurons are correlated with one another, representing a population of neurons similarly tuned to wrist motion. A cost function is defined that requires the control unit to reach a target firing rate, related to the BCI output, and the remainder of the units to reach a separate firing rate, related to their role motor control. We observe that, using the SDE-based learning rule, the neurons learn to minimize this cost function – the control unit changes its weights specifically. This performance is independent of the degree of correlation between the neurons (Figure 4C). An observed-dependence based learning rule also achieves this, but the final performance depends on the degree of correlation (Figure 4D). Yet dual-control BCI studies have shown that performance is independent of a control unit's tuning to wrist motion or effective connectivity with other units [50, 39]. Examination of the synaptic weights as learning proceeds shows the control unit in SDE-based learning quickly separates itself from the other units (Figure 4E). The control unit's weight in observed-dependence based learning, on the other hand, initially increases with the other units (Figure 4F). It cannot distinguish its effect on the cost function from the other units' effect, even with relatively low, physiologically relevant correlations (e.g. $c = 0.3$, [15]). SDE-based learning may be relevant in the context of BCI learning.

Discussion

Here we have cast neural learning explicitly as a causal inference problem, and have shown that neurons can estimate their causal effect using their spiking mechanism. In this way we found that spiking can be an advantage, allowing neurons to quantify their causal effects in an unbiased way. We outlined sufficient assumptions for this causal inference to be correct, and have shown how this causal inference estimate can be turned into a learning rule that can optimize the synaptic weights of a neuron.

The rule is inspired by the regression discontinuity design commonly used in econometrics [3]. Previous work in econometrics has considered how the threshold may be changed [17] and how that can be used to optimize utility [47]. We have extended this work by deriving a rule that allows for the neuron to change its weights, instead of just its threshold, to increase utility. This provides the ability to change not just the

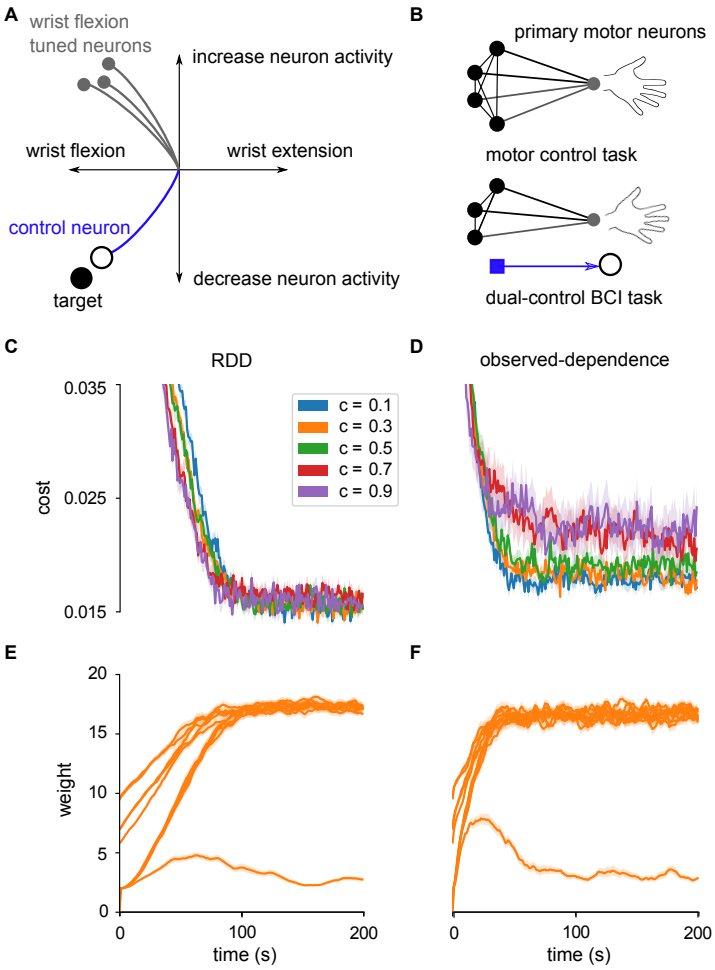

Figure 4: **Application to BCI learning.** (A) Dual-control BCI setup [50]. Vertical axis of a cursor is controlled by one control neuron, horizontal axis is controlled by wrist flexion/extension. Wrist flexion-tuned neurons (gray) would move cursor up and right when wrist is flexed. To reach a target, task may require the control neuron (blue) dissociate its relation to wrist motion – decrease its activity and flex wrist simultaneously – to move cursor (white cursor) to target (black circle). (B) A population of primary motor neurons correlated with (tuned to) wrist motion. One neuron is chosen to control the BCI output (blue square), requiring it to fire independently of other neurons, which maintain their relation to each other and wrist motion [39]. (C) SDE-based learning of dual-control BCI task for different levels of correlation. Curves show mean over 10 simulations, shaded regions indicate standard error. (D) Observed dependence-based learning of dual-control BCI task. (E) Synaptic weights for each unit with $c = 0.3$ in SDE-based learning. Control unit is single, smaller weight. (F) Synaptic weights for each unit in observed-dependence based learning.

threshold but the construction of the variable being thresholded to optimize reward.

It is important to note that other neural learning rules also perform causal inference. Thus the SDE-based learning rule can be placed in the context of other neural learning mechanisms. First, as many authors have noted, any reinforcement learning algorithm relies on estimating the effect of an agent's/neuron's activity on a reward signal. Learning by operant conditioning relies on learning a causal relationship (compared to classical conditioning, which only relies on learning a correlation) [76, 56, 28, 25]. Causal inference is, at least implicitly, the basis of reinforcement learning.

There is a large literature on how reinforcement learning algorithms can be implemented in the brain. It it well known there are many neuromodulators which may represent reward or expected reward, including dopaminergic neurons from the substantia nigra to the ventral striatum representing a reward prediction error [75, 62]. Many of these methods use something like the REINFORCE algorithm [74], a policy gradient method in which locally added noise is correlated with reward and this correlation is used to update weights. This gives an unbiased estimate of the causal effect because the noise is assumed to be independent, private to each neuron. These ideas have extensively been used to model learning in brains [11, 21, 20, 44, 49, 77, 64].

Learning in birdsong is a particularly well developed example of this form of learning [20]. In birdsong learning in zebra finches, neurons from area LMAN synapse onto neurons in area RA. These synapses are referred to as 'empiric' synapses, and are treated by the neurons as an 'experimenter', producing random perturbations which can be used to estimate causal effects. This is a compelling account of learning in birdsong, however it relies on the specific structural form of the learning circuit. It is unknown more generally how a neuron may estimate what is perturbative noise without these structural specifics, and thus if it can provide an account of learning more generally.

There are two factors that cast doubt on the use of reinforcement learning-type algorithms broadly in neural circuits. First, even for a fixed stimulus, noise is correlated across neurons [5, 15, 37, 7, 78]. Thus if the noise a neuron uses for learning is correlated with other neurons then it can not know which neuron's changes in output is responsible for changes in reward. In such a case, the synchronizing presynaptic activity acts as a confounder. Thus, as discussed, such algorithms require biophysical mechanisms to distinguish independent perturbative noise from correlated input signals in presynaptic activity, and in general it is unclear how a neuron could can do this. And, second, learning with perturbations scales poorly with network size [64, 20, 60]. Thus neurons may use alternatives to these reinforcement-learning algorithms.

Given the inefficiency of these reinforcement learning algorithms, a number of authors have looked to learning in artificial neural networks for inspiration. In artificial neural networks, the credit assignment problem is efficiently solved using the backpropagation algorithm, which allows efficiently calculating gradients. Backpropagation requires differentiable systems, which spiking neurons are not. Indeed, cortical networks often have low firing rates in which the stochastic and discontinuous nature of spiking output cannot be neglected [65]. It also requires full knowledge of the system, which is often not the case if parts of the system relate to the outside world. No known structures exist in the brain that could exactly implement backpropagation. Yet, backpropagation is significantly more powerful than perturbation-based methods – it is the only known algorithm able to solve large-scale problems at a human-level [42]. The success of backpropagation suggests that efficient methods for computing gradients are needed for solving large-scale learning problems.

A number of recent approaches have looked for neural mechanisms which allow for more specific feedback signals to support learning, inspired by the efficiency of the backpropagation algorithm [27, 38, 45]. By providing neuron-specific error feedback, these methods provide information about each neuron's causal effect. In particular, this work has proposed neural learning in cortex uses neuron-specific feedback by separating the computation into two compartments. In one component the neuron integrates feedforward inputs, and in the other component the neuron integrates feedback signals in order to estimate something like a reward gradient. This separation is plausibly implementable in apical and basal dendritic compartments of

cortical pyramidal neurons. This approach is shared with SDE-based learning, in which the casual effect is estimated separately and then used to drive learning. We may thus expect that an SDE-based learning rule could be readily combined with these compartmental models of learning in cortex to provide a biologically plausible account of efficient learning.

Finally, it must be noted how the learning rule derived here relates to the dominant spike-based learning paradigm – spike timing dependent plasticity (STDP [10]). STDP performs unsupervised learning, so is not directly related to the type of optimization considered here. Reward-modulated STDP (R-STDP) can be shown to approximate the reinforcement learning policy gradient type algorithms described above [46, 24, 35]. Thus R-STDP can be cast as performing a type of causal inference on a reward signal, and shares the same features and caveats as outlined above. Thus we see that learning rules that aim at maximizing some reward either implicitly or explicitly involve a neuron estimating its causal effect on that reward signal. Explicitly recognizing this can lead to new methods and understanding.

Caveats and advantages

There are multiple caveats for the use of an SDE-based learning rule. The first is that the rule is only applicable in cases where the effect of a single spike is relevant. Depending on the way a network is constructed, the importance of each neuron may decrease as the size of the network is increased. As the influence of a neuron vanishes, it becomes hard to estimate this influence. While this general scaling behavior is shared with other algorithms (e.g. backpropagation), it is more crucial for SDE where there will be some noise in the evaluation of the outcome.

A second caveat is that the SDE rule does not solve the temporal credit assignment problem. It requires us to know which output is associated with which kind of activity. There are multiple approaches that can help solve this kind of problem, including actor-critic methods and eligibility traces from reinforcement learning theory [68].

A third caveat is that, as implemented here, the rule learns the effect of a neuron's activity on a reward signal for a fixed input. Thus the rule is applicable in cases where a fixed output is required. This includes learning stereotyped motor actions, or learning to mimic a parent's birdsong [20, 21]. Applying the learning rule to networks with varying inputs, as in many supervised learning tasks, would require extensions. One possible extension that may address these caveats is, rather than directly estimate the effect of a neuron's output on a reward function, to use the method to learn weights on feedback signals so as to approximate the causal effect – that is, to use the SDE rule to "learn how to learn"[40]. This approach has been shown to work in artificial neural networks, suggesting SDE-based learning may provide a biologically plausible and scale-able approach to learning [16, 48, 27].

This paper introduces the SDE method to neuronal learning and artificial neural networks. It illustrates the difference in behavior of SDE and observed-dependence learning in the presence of confounding. While it is of a speculative nature, at least in cases where reward signals are observed, it does provide a biologically plausible account of neural learning. It addresses a number of issues with other learning mechanisms.

SDE-based learning does not require independent noise. It is sufficient that something, in fact anything that is presynaptic, produce variability. As such, SDE approaches do not require the noise source to be directly measured. This allows to rule to be applied in a wider range of neural circuits or artificial systems.

SDE-based learning removes confounding due to noise correlations. Noise correlations are known to be significant in many sensory processing areas [15]. While noise correlations' role in sensory encoding has been well studied [5, 15, 37, 7, 78], their role in learning has been less studied. This work suggests that understanding learning as a causal inference problem can provide insight into the role of noise correlations in learning.

Finally, in a lot of theoretical work, spiking is seen as a disadvantage, and models thus aim to remove spiking discontinuities through smoothing responses [32, 31, 43]. The SDE rule, on the other hand, exploits the spiking discontinuity. Moreover, the rule can operate in environments with non-differentiable or discontinuous reward functions. In many real-world cases, gradient descent would be useless: even if the brain could implement it, the outside world does not provide gradients (but see [73]). Our approach may thus be useful even in scenarios, such as reinforcement learning, where spiking is not necessary. Spiking may, in this sense, allow a natural way of understanding a neuron's causal influence in a complex world.

Compatibility with known physiology

If neurons perform something like SDE-based learning we should expect that they exhibit certain physiological properties. We thus want to discuss the concrete demands of SDE-based learning and how they relate to past experiments.

The SDE learning rule is applied only when a neuron's membrane potential is close to threshold, regardless of spiking. This means inputs that place a neuron close to threshold, but do not elicit a spike, still result in plasticity. This type of sub-threshold dependent plasticity is known to occur [22, 23, 67]. This also means that plasticity will not occur for inputs that place a neuron too far below threshold. In past models of voltage-dependent plasticity and experimental findings, changes do not occur when postsynaptic voltages are too low (Figure 5A) [14, 4]. And the SDE learning rule predicts that plasticity does not occur when postsynaptic voltages are too high. However, in many voltage-dependent plasticity models, potentiation does occur for inputs well-above the spiking threshold. But, as SDE-based learning occurs in low-firing rate regimes, inputs that place a neuron too far above threshold are rare. So this discrepancy may not be relevant. Thus threshold-adjacent plasticity as required for SDE-based learning appears to be compatible with neuronal physiology.

The SDE-based learning predicts that spiking switches the sign of plasticity. Some experiments and phenomenological models of voltage-dependent synaptic plasticity do capture this behavior [4, 14, 13]. In these models, plasticity changes from depression to potentiation near the spiking threshold (Figure 5A). Thus this property of SDE-based learning also appears to agree with some models and experimental findings.

The SDE-based learning is dependent on neuromodulation. Neuromodulated-STDP is well studied in models [24, 63, 55]. However, SDE-based learning requires plasticity switch sign under different levels of reward. This may be communicated by neuromodulation. There is some evidence that the relative balance between adrenergic and M1 muscarinic agonists alters both the sign and magnitude of STDP in layer II/III visual cortical neurons [63] (Figure 5B). To the best of our knowledge, how such behavior interacts with postsynaptic voltage dependence as required by SDE-learning is unknown. Thus, taken together, these factors show SDE-based learning may well be compatible with known neuronal physiology, and leads to predictions that can demonstrate SDE-based learning.

Experimental predictions

A number of experiments can be used to test if neurons use SDE to perform causal inference. This has two aspects, neither of which have been explicitly tested before: (1) do neurons learn in the presence of strong confounding, which is do causal inference and (2) do they use SDE to do so? To test both of these we propose scenarios that let an experimenter control the degree of confounding in a population of neurons' inputs and control a reward signal.

To test if neurons learn in the presence of strong confounding, we propose an experiment where a lot of neurons are stimulated in a time-varying way but only one of the neurons has a causal influence, which is

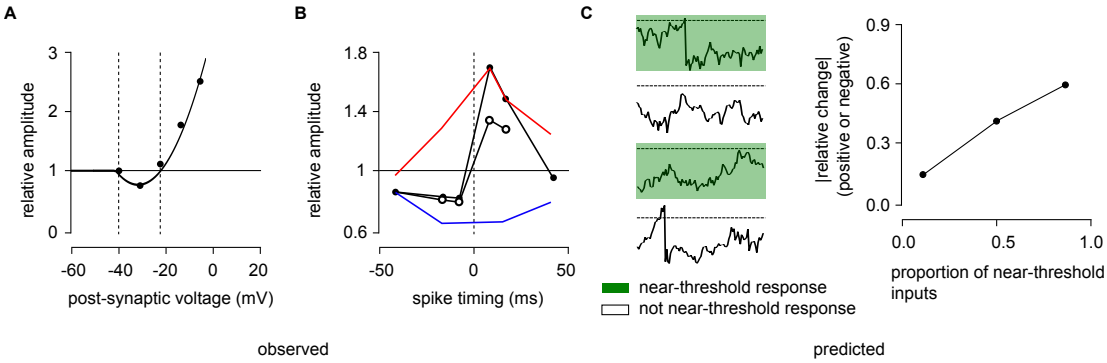

Figure 5: **Feasibility and predictions of SDE-based learning.** (A) Voltage-clamp experiments in adult mice CA1 hippocampal neurons [53]. Amplitude of potentiation or depression depends on post-synaptic voltage. (B) Neuromodulators affect magnitude of STDP. High proportion of β-adrenergic agonist to M1 muscarinic agonist (filled circles), compared with lower proportion (unfilled circles) in visual cortex layer II/III neurons. Only β-adrenergic agonist produces potentiation (red curve), and only M1 muscarinic agonist produces depression (blue) [63]. (C) Prediction of SDE-based learning. Over a fixed time window a reward is administered when neuron spikes. Stimuli are identified which place the neuron's input drive close to spiking threshold. SDE-based learning predicts an increase synaptic changes for a set of stimuli containing a high proportion of near threshold inputs, but that keeps overall firing rate constant.

defined in a brain computer interface (BCI) experiment (like [18]). Due to the joint stimulation, all neurons will be correlated at first and therefore all neurons will correlate with the reward. But as only one neuron has a causal influence, only that neuron should increase its firing (as in Fig. 4. Testing how each neuron adjusts its firing properties in the presence of different strengths of correlation would establish if the neuron causally related to the reward signal can be distinguished from the others.

A similar experiment can also test if neurons specifically use SDE to achieve deconfounding. SDE-based learning happens when a neuron is close to threshold. We could use manipulations to affect when a neuron is close to threshold, e.g. through patch clamping or optical stimulation. By identifying these inputs, and then varying the proportion of trials in which a neuron is close to threshold, while keeping the total firing rate the same, the SDE hypothesis can be tested (Figure 5C). SDE-based learning predicts that a set of trials that contain many near-threshold events will result in faster learning than in a set of trials that place a neuron either well above or well below threshold.

Alternatively, rather than changing input activity to bring a neuron close to threshold, or not, SDE-based learning can be tested by varying when reward is administered. Again, identifying trials when a neuron is close to threshold and introducing a neurotransmitter signalling reward (e.g. dopamine) at the end of these trials should, according to SDE learning, result in an increased learning rate, compared to introducing the same amount of neurotransmitter whenever the neuron spiked. These experiments are important future work that would demonstrate SDE-based learning.

Conclusion

The most important aspect of SDE-based learning is the explicit focus on causality. A causal model is one that can describe the effects of an agent's actions on an environment. Learning through the reinforcement of an agent's actions relies, even if implicitly, on a causal understanding of the environment [25, 41]. Here,

by explicitly casting learning as a problem of causal inference we have developed a novel learning rule for spiking neural networks. We present the first model to propose a neuron does causal inference. We believe that focusing on causality is essential when thinking about the brain or, in fact, any system that interacts with the real world.

Methods

Neuron, noise and reward model

We consider the activity of a network of n neurons whose activity is described by their spike times

$$h_i(t) = \sum \delta(t - t_s^i).$$

Here $n = 2$. Synaptic dynamics $\mathbf{s} \in \mathbb{R}^n$ are given by

$$\tau_s \dot{s}_i = -s_i + h_i(t), \tag{3}$$

for synaptic time scale τ_s. An instantaneous reward is given by $R(\mathbf{s}) \in \mathbb{R}$. In order to have a more smooth reward signal, R is a function of $\mathbf{s}$ rather than $\mathbf{h}$. The reward function used here has the form of a Rosenbrock function:

$$R(s_1, s_2) = (a - s_1)^2 + b(s_2 - s_1^2)^2.$$

The neurons obey leaky integrate-and-fire (LIF) dynamics

$$\dot{v}_i = -g_L v_i + w_i \eta_i, \tag{4}$$

where integrate and fire means simply:

$$v_i(t^+) = v_r, \quad \text{when } v_i(t) = \theta.$$

Noisy input η_i is comprised of a common DC current, x, and noise term, $\xi(t)$, plus an individual noise term, $\xi_i(t)$:

$$\eta_i(t) = x + \sigma_i \left[\sqrt{1-c}\xi_i(t) + \sqrt{c}\xi(t)\right].$$

The noise processes are independent white noise: $\mathbb{E}(\xi_i(t)\xi_j(t')) = \sigma^2\delta_{ij}\delta(t-t')$. This parameterization is chosen so that the inputs $\eta_{1,2}$ have correlation coefficient c. Simulations are performed with a step size of $\Delta t = 1$ms. Here the reset potential was set to $v_r = 0$. Borrowing notation from Xie and Seung 2004 [77], the firing rate of a noisy integrate and fire neuron is

$$\mu_i = \left[\frac{1}{g_L}\int_0^\infty \frac{1}{u}\left(\exp\left(-u^2 + 2y_i^{th}u\right) - \exp\left(-u^2 + 2y_i^r u\right)\right) du\right]^{-1},$$

where $y_i^{th} = (\theta - w_i x)/\sigma_i$ and $y_i^r = -w_i x/\sigma_i$, $\sigma_i = \sigma w_i$ is the input noise standard deviation.

We define the input drive to the neuron as the leaky integrated input without a reset mechanism. That is, over each simulated window of length T:

$$\dot{u}_i = -g_L u_i + w_i \eta_i, \quad u_i(0) = v_i(0).$$

The SDE method operates when a neuron receives inputs that place it close to its spiking threshold – either nearly spiking or barely spiking – over a given time window. In order to identify these time periods, the method uses the maximum input drive to the neuron:

$$Z_i = \max_{0 \leq t \leq T} u_i(t).$$

The input drive is used here instead of membrane potential directly because it can distinguish between marginally super-threshold inputs and easily super-threshold inputs, whereas this information is lost in the voltage dynamics once a reset occurs. Here a time period of $T = 50$ms was used. Reward is administered at the end of this period: $R = R(\mathbf{s}_T)$.

Policy gradient methods in neural networks

The dynamics given by (4) generate an ergodic Markov process with a stationary distribution denoted ρ. We consider the problem of finding network parameters that maximize the expected reward with respect to ρ. In reinforcement learning, performing optimization directly on the expected reward leads to policy gradient methods [69]. These typically rely on either finite difference approximations or a likelihood-ratio decomposition [74]. Both approaches ultimately can be seen as performing stochastic gradient descent, updating parameters by approximating the expected reward gradient:

$$\nabla_{\mathbf{w}} \mathbb{E}_\rho R, \tag{5}$$

for neural network parameters $\mathbf{w}$.

Here capital letters are used to denote the random variables drawn from the stationary distribution, corresponding to their dynamic lower-case equivalent above. Density ρ represents a joint distribution over variables (Z, H, S, R), the maximum input drive, spiking indicator function, filtered spiking output, and reward variable, respectively. The spiking indicator function is defined as $H_i = \mathbb{I}(Z_i \geq \theta)$, for threshold θ.

We wish to evaluate (5). In general there is no reason to expect that taking a derivative of an expectation with respect to some parameters will have the same form as the corresponding derivative of a deterministic function. However in some cases this is true, for example when the parameters are separable from the distribution over which the expectation is taken (sometimes relying on what is called the reparameterization trick [60, 29]). Here we show that, even when the reparameterization trick is unavailable, if the system contains a Bernoulli (spiking indicator) variable then the expression for the reward gradient also matches a form we might expect from taking the gradient of a deterministic function.

The expected reward can be expressed as

$$\mathbb{E}(R) = \mathbb{E}(R|H_i = 1)P(H_i = 1) + \mathbb{E}(R|H_i = 0)P(H_i = 0), \tag{6}$$

for a neuron i. The key assumption we make is the following:

Assumption 1. The neural network parameters only affect the expected reward through their spiking activity, meaning that $\mathbb{E}(R|H)$ is independent of parameters $\mathbf{w}$.

We would like to take derivatives of $\mathbb{E}(R)$ with respect to $\mathbf{w}$ using (6). However even if, under Assumption 1, the joint conditional expectation $\mathbb{E}(R|H)$ is independent of $\mathbf{w}$, it is not necessarily the case that the marginal conditional expectation, $\mathbb{E}(R|H_i)$, is independent of $\mathbf{w}$. This is because it involves marginalization over unobserved neurons $H_{j \neq i}$, which may have some relation to H_i that is dependent on $\mathbf{w}$. That is,

$$\mathbb{E}(R|H_i = 1) = \sum_{j \neq i} \mathbb{E}(R|H_i = 1, H_j)P(H_j|H_i = 1; \mathbf{w}),$$

where the conditional probabilities may depend on $\mathbf{w}$. This dependence complicates taking derivatives of the decomposition (6) with respect to $\mathbf{w}$.

Unconfounded network

We can gain intuition about how to proceed by first making a simplifying assumption. If we assume that $H_i \perp\!\!\!\perp H_{j\neq i}$, then the reward gradient is simple to compute from (6). This is because now the parameter w_i only affects the probability of neuron i spiking:

$$\begin{aligned}\frac{\partial}{\partial w_i}\mathbb{E}(R) &= \frac{\partial}{\partial w_i}\left(\mathbb{E}(R|H_i=1)P(H_i=1;w_i)+\mathbb{E}(R|H_i=0)P(H_i=0;w_i)\right)\\ \text{(Assumption 1)}\quad &= \frac{\partial P(H_i=1;w_i)}{\partial w_i}\left(\mathbb{E}(R|H_i=1)-\mathbb{E}(R|H_i=0)\right)\\ &= \frac{\partial \mathbb{E}(H_i;w_i)}{\partial w_i}\left(\mathbb{E}(R|H_i=1)-\mathbb{E}(R|H_i=0)\right).\end{aligned}$$

This resembles a type of finite difference estimate of the gradient we might use if the system were deterministic and H were differentiable:

$$\frac{\partial R}{\partial w} \approx \frac{\partial H}{\partial w}\frac{R(H=1)-R(H=0)}{\Delta H}.$$

Based on the independence assumption we call this the unconfounded case. In fact the same decomposition is utilized in a REINFORCE-based method derived by Seung 2003 [64].

Confounded network

Generally, of course, it is not the case that $H_i \perp\!\!\!\perp H_{j\neq i}$, and then we must decompose the expected reward into:

$$\begin{aligned}\mathbb{E}(R) = \sum_{h_{j\neq i}} &P(H_{j\neq i}=h_{j\neq i})(\mathbb{E}(R|H_i=1,H_{j\neq i}=h_{j\neq i})P(H_i=1|H_{j\neq i}=h_{j\neq i})\\ &+\mathbb{E}(R|H_i=0,H_{j\neq i}=h_{j\neq i})P(H_i=0|H_{j\neq i}=h_{j\neq i})).\end{aligned}$$

This means the expected reward gradient is given by:

$$\begin{aligned}\frac{\partial}{\partial w_i}\mathbb{E}(R) &= \sum_{h_{j\neq i}} P(H_{j\neq i}=h_{j\neq i})\frac{\partial P(H_i=1|H_{j\neq i}=h_{j\neq i})}{\partial w_i}(\mathbb{E}(R|H_i=1,H_{j\neq i}=h_{j\neq i})-\mathbb{E}(R|H_i=0,H_{j\neq i}=h_{j\neq i}))\\ &= \mathbb{E}\left(\frac{\partial \mathbb{E}(H_i|H_{j\neq i})}{\partial w_i}\left(\mathbb{E}(R|H_i=1,H_{j\neq i})-\mathbb{E}(R|H_i=0,H_{j\neq i})\right)\right),\end{aligned}$$

again making use of Assumption 1. We additionally make the following approximation:

Assumption 2. The gradient term $\frac{\partial \mathbb{E}(H_i|H_{j\neq i})}{\partial w_i}$ is independent of $H_{j\neq i}$.

This means we can move the gradient term out of the expectation to give:

$$\frac{\partial}{\partial w_i}\mathbb{E}(R) \approx \frac{\partial \mathbb{E}(H_i)}{\partial w_i}\mathbb{E}\left(\mathbb{E}(R|H_i=1,H_{j\neq i})-\mathbb{E}(R|H_i=0,H_{j\neq i})\right). \tag{7}$$

We assume that how the neuron's activity responds to changes in synaptic weights, $\frac{\partial \mathbb{E}(H_i)}{\partial w_i}$, is known by the neuron. Thus it remains to estimate $\mathbb{E}(R|H_i=1,H_{j\neq i})-\mathbb{E}(R|H_i=0,H_{j\neq i})$. It would seem this relies on a neuron observing other neurons' activity. Below we show how it can be estimated, however, using methods from causal inference.

The unbiased gradient estimator as a causal effect

We can identify the unbiased estimator (Equation (7)) as a causal effect estimator. To understand precisely what this means, here we describe a causal model.

A causal model is a Bayesian network along with a mechanism to determine how the network will respond to intervention. This means a causal model is a directed acyclic graph (DAG) $\mathcal{G}$ over a set of random variables $\mathcal{X} = \{X_i\}_{i=1}^N$ and a probability distribution P that factorizes over $\mathcal{G}$ [56].

An intervention on a single variable is denoted $\text{do}(X_i = y)$. Intervening on a variable removes the edges to that variable from its parents, Pa_{X_i}, and forces the variable to take on a specific value: $P(x_i|\text{Pa}_{X_i} = \mathbf{x_i}) = \delta(x_i = y)$. Given the ability to intervene, the average treatment effect (ATE), or causal effect, between an outcome variable X_j and a binary variable X_i can be defined as:

$$ATE \coloneqq \mathbb{E}(X_j|\text{do}(X_i = 1)) - \mathbb{E}(X_j|\text{do}(X_i = 0)).$$

We make use of the following result: if $\mathbf{S}_{ij} \subset \mathcal{X}$ is a set of variables that satisfy the *back-door criteria* with respect to $X_i \to X_j$, then it satisfies the following: (i) $\mathbf{S}_{ij}$ blocks all paths from X_i to X_j that go into S_i, and (ii) no variable in $\mathbf{S}_{ij}$ is a descendant of X_i. In this case the interventional expectation can be inferred from

$$\mathbb{E}(X_j|\text{do}(X_i = y)) = \mathbb{E}\left(\mathbb{E}(X_j|\mathbf{S}_{ij}, X_i = y)\right).$$

Given this framework, here we will define the causal effect of a neuron as the average causal effect of a neuron H_i spiking or not spiking on a reward signal, R:

$$\beta_i \coloneqq \mathbb{E}(R|\text{do}(H_i = 1)) - \mathbb{E}(R|\text{do}(H_i = 0)),$$

where H_i and R are evaluated over a short time window of length T.

We make the final assumption:

Assumption 3. Neurons $H_{j\neq i}$ satisfy the backdoor criterion with respect to $H_i \to R$.

Then it is the case that the reward gradient estimator, (7), in fact corresponds to:

$$\frac{\partial}{\partial w_i}\mathbb{E}(R) \approx \frac{\partial \mathbb{E}(H_i)}{\partial w_i}\beta_i. \tag{8}$$

Thus we have the result that, in a confounded, spiking network, gradient descent learning corresponds to causal learning. The validity of the three above assumptions for a network of integrate and fire neurons is demonstrated in the Supplementary Material (Section S1). The relation between the causal effect and a finite difference estimate of the reward gradient is presented in the Supplementary Material (Section S2).

Using the spiking discontinuity

To remove confounding, SDE considers only the marginal super- and sub-threshold periods of time to estimate (8). This works because the discontinuity in the neuron's response induces a detectable difference in outcome for only a negligible difference between sampled populations (sub- and super-threshold periods). The SDE method estimates [34]:

$$\beta_i^{RD} \coloneqq \lim_{x\to\theta^+} \mathbb{E}(R|Z_i = x) - \lim_{x\to\theta^-} \mathbb{E}(R|Z_i = x),$$

for maximum input drive obtained over a short time window, Z_i, and spiking threshold, θ; thus, $Z_i < \theta$ means neuron i does not spike and $Z_i \geq \theta$ means it does.

To estimate β_i^{RD}, a neuron can estimate a piece-wise linear model of the reward function:

$$R = \gamma_i + \beta_i H_i + [\alpha_{ri} H_i + \alpha_{li}(1 - H_i)](Z_i - \theta),$$

locally, when Z_i is within a small window p of threshold. Here γ_i, α_{li} and α_{ri} are nuisance parameters, and β_i is the causal effect of interest. This means we can estimate β_i^{RD} from

$$\beta_i \approx \mathbb{E}(R - \alpha_r(Z_i - \theta)|\theta \leq Z_i < \theta + p) - \mathbb{E}(R - \alpha_l(Z_i - \theta)|\theta - p < Z_i < \theta).$$

A neuron can learn an estimate of β_i^{RD} through a least squares minimization on the model parameters $\beta_i, \alpha_l, \alpha_r$. That is, if we let $\mathbf{u}_i = [\beta_i, \alpha_r, \alpha_l]^T$ and $\mathbf{a}_t = [1, h_{i,t}(z_{i,t} - \theta), (1 - h_{i,t})(z_{i,t} - \theta)]^T$, then the neuron solves:

$$\hat{\mathbf{u}}_i = \text{argmin}_u \sum_{t:(\theta - p < z_{i,t} < \theta + p)}^{T} \left[\mathbf{u}_i^T \mathbf{a}_t - (2h_{i,t} - 1)R_t\right]^2.$$

Performing stochastic gradient descent on this minimization problem gives the learning rule:

$$\Delta \mathbf{u}_i = \begin{cases} -\eta[\mathbf{u}_i^T \mathbf{a}_i - R_t]\mathbf{a}_i, & \theta \leq z_{i,t} < \theta + p \text{ (just spikes)}; \\ -\eta[\mathbf{u}_i^T \mathbf{a}_i + R_t]\mathbf{a}_i, & \theta - p < z_{i,t} < \theta \text{ (almost spikes)}, \end{cases}$$

for all time periods at which $z_{i,t}$ is within p of threshold θ.

The causal effect as a finite-difference operator

The estimator can also be considered as a type of finite-difference approximation to the reward gradient we would compute in the deterministic case. This relation is fleshed out here. Specifically, we show that

$$\beta_i = \mathbb{E}(R|\text{do}(H_i = 1)) - \mathbb{E}(R|\text{do}(H_i = 0)) \approx \mathbb{E}\left(\frac{\partial R}{\partial S_i}\right).$$

To show this we replace $\frac{\partial}{\partial S_i}$ with a type of finite difference operator:

$$D_i R(S_i, \mathbf{S}_{j \neq i}) \coloneqq \frac{1}{\Delta_s}\left(\mathbb{E}(R|S_i + \Delta_s, \mathbf{S}_{j \neq i}) - \mathbb{E}(R|S_i, \mathbf{S}_{j \neq i})\right).$$

Here $\mathbf{S}_{j \neq i} \subset \mathcal{X}$ is a set of nodes that satisfy the back-door criterion with respect to $H_i \to R$. When R is a deterministic, differentiable function of $\mathbf{S}$ and $\Delta_s \to 0$ this recovers the reward gradient $\frac{\partial R}{\partial S_i}$ and we recover gradient descent-based learning.

To consider the effect of a single spike, note that unit i spiking will cause a jump in S_i compared to not spiking (according to synaptic dynamics). If we let Δ_s equal this jump then it can be shown that $\mathbb{E}(D_i R)$ is related to the causal effect.

First, assuming the conditional independence of R from H_i given S_i and $\mathbf{S}_{j \neq i}$:

$$\begin{aligned} \beta_i &= \mathbb{E}\left(R|\text{do}(H_i = 1)\right) - \mathbb{E}\left(R|\text{do}(H_i = 0)\right) \\ &= \mathbb{E}\left(\mathbb{E}\left(R|\mathbf{S}_{j \neq i}, H_i = 1\right) - \mathbb{E}\left(R|\mathbf{S}_{j \neq i}, H_i = 0\right)\right) \\ &= \mathbb{E}\left(\mathbb{E}\left(\mathbb{E}\left(R|S_i, \mathbf{S}_{j \neq i}\right)|\mathbf{S}_{j \neq i}, H_i = 1\right) - \mathbb{E}\left(\mathbb{E}\left(R|S_i, \mathbf{S}_{j \neq i}\right)|\mathbf{S}_{j \neq i}, H_i = 0\right)\right). \end{aligned} \tag{9}$$

Now if we assume that on average H_i spiking induces a change of Δ_s in S_i within the same time period, compared with not spiking, then:

$$\rho(s_i|\mathbf{S}_{j\neq i}, H_i = 1) \approx \rho(s_i - \Delta_s|\mathbf{S}_{j\neq i}, H_i = 0). \tag{10}$$

This is reasonable because the linearity of the synaptic dynamics means that the difference in S_i between spiking and non-spiking windows is simply $\exp(-t_{si}/\tau_s)/\tau_s$, for spike time t_{si}. We approximate this term with its mean:

$$\begin{aligned}\Delta_s &= \mathbb{E}\left(\frac{1}{\tau_s}e^{-t_{si}/\tau_s}|\mathbf{S}_{j\neq i}, H_i = 1\right)\\ &\approx \frac{1}{T}\left(1 - e^{-T/\tau_s}\right),\end{aligned} \tag{11}$$

under the assumption that spike times occur uniformly throughout the length T window. These assumptions are supported numerically (Figure 6).

Writing out the inner two expectations of (9) gives:

$$\begin{aligned}&\mathbb{E}\left(\mathbb{E}\left(R|S_i, \mathbf{S}_{j\neq i}\right)|\mathbf{S}_{j\neq i}, H_i = 1\right) - \mathbb{E}\left(\mathbb{E}\left(R|S_i, \mathbf{S}_{j\neq i}\right)|\mathbf{S}_{j\neq i}, H_i = 0\right)\\ &= \int_0^\infty \mathbb{E}(R|\mathbf{S}_{j\neq i}, S_i = s_i)\left[\rho(s_i|\mathbf{S}_{j\neq i}, H_i = 1) - \rho(s_i|\mathbf{S}_{j\neq i}, H_i = 0)\right] ds_i\\ \text{from (10)} \quad &= \int_0^\infty \mathbb{E}(R|S_i = s_i + \Delta_s, \mathbf{S}_{j\neq i})\rho(s_i|\mathbf{S}_{j\neq i}, H_i = 0) - \mathbb{E}(R|S_i = s_i, \mathbf{S}_{j\neq i})\rho(s_i|\mathbf{S}_{j\neq i}, H_i = 0)\, ds_i,\end{aligned}$$

after making the substitution $s_i \to s_i + \Delta_s$ in the first term. Writing this back in terms of expectations gives:

$$\begin{aligned}\beta &\approx \mathbb{E}\left(\mathbb{E}\left(\mathbb{E}\left(R|S_i + \Delta_s, \mathbf{S}_{j\neq i}\right) - \mathbb{E}\left(R|S_i, \mathbf{S}_{j\neq i}\right)|\mathbf{S}_{j\neq i}, H_i = 0\right)\right)\\ &= \Delta_s\mathbb{E}(\mathbb{E}\left(D_iR(S_i, \mathbf{S}_{j\neq i})|\mathbf{S}_{j\neq i}, H_i = 0\right))\\ &= \Delta_s\mathbb{E}(D_iR(S_i, \mathbf{S}_{j\neq i})|\mathrm{do}(H_i = 0)).\end{aligned}$$

Thus estimating the causal effect is similar to taking a finite difference approximation of the reward gradient.

Implementation

`python` code used to run simulations and generate figures is available at `https://github.com/benlansdell/rdd`.

Acknowledgements

The authors would like to thank Roozbeh Farhoodi, Ari Benjamin and David Rolnick for valuable discussion and feedback.

Author contributions

K.P.K and B.J.L. devised the study, B.J.L. performed the analysis, and K.P.K and B.J.L. wrote the manuscript.

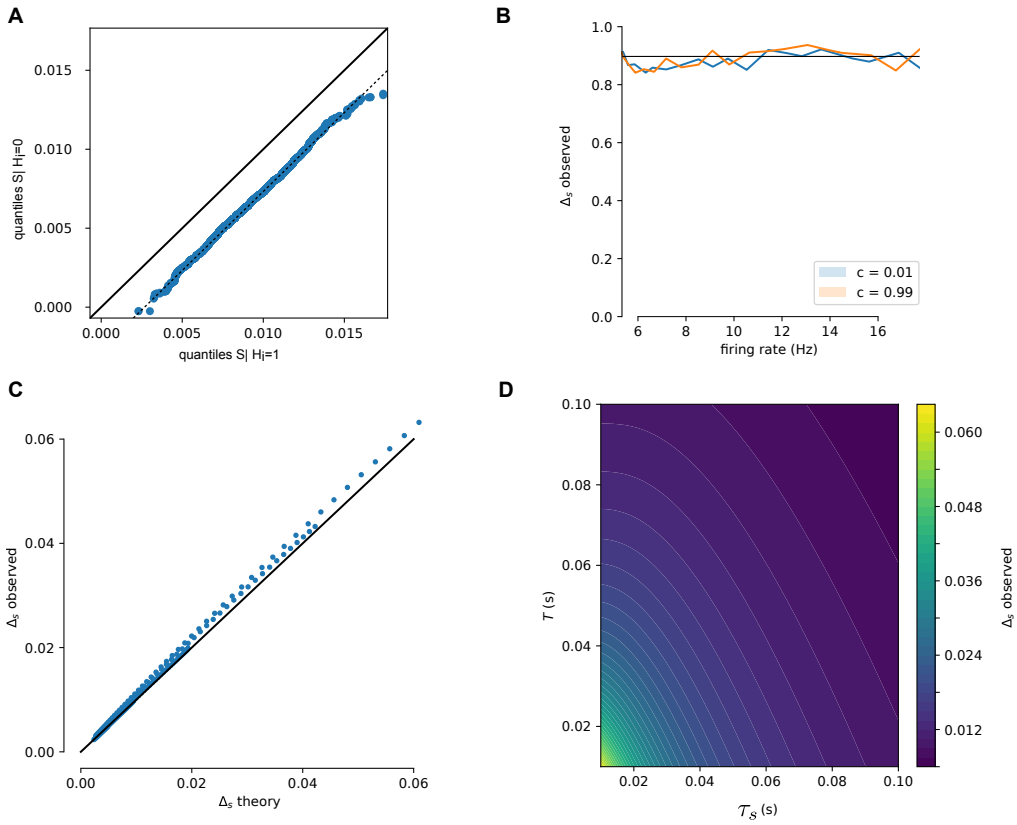

Figure 6: **Relation between S_i and H_i over window T.** (A) Simulated spike trains are used to generate $S_i|H_i = 0$ and $S_i|H_i = 1$. QQ-plot shows that S_i following a spike is distributed as a translation of S_i in windows with no spike, as assumed in (10). (B) This offset, Δ_s, is independent of firing rate and is unaffected by correlated spike trains. (C) Over a range of values ($0.01 < T < 0.1, 0.01 < \tau_s < 0.1$) the derived estimate of Δ_s (Equation (11)) is compared to simulated Δ_s. Proximity to the diagonal line (black curve) shows these match. (D) Δ_s as a function of window size T and synaptic time constant τ_s. Larger time windows and longer time constants lower the change in S_i due to a single spike.

References

[1] Mohammed Alawad, Hong-jun Yoon, and Georgia Tourassi. Energy Efficient Stochastic-Based Deep Spiking Neural Networks for Sparse Datasets. *IEEE Conference on Big Data*, pages 311–318, 2017.

[2] Joshua Angrist and Jorn-Steffen Pischke. The Credibility Revolution in Empirical Economics: How Better Research Design is Taking the Con Out of Economics. *Journal of Economic Perspectives*, 24(2), 2010.

[3] Joshua D Angrist and Jorn-Steffen Pischke. *Mostly Harmless Econometrics : An Empiricist ' s Companion*. Princeton University Press, 2009.

[4] A Artola, S Brocher, and W Singer. Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature*, 347(6288):69–72, 1990.

[5] Cody Baker, Christopher Ebsch, Ilan Lampl, and Robert Rosenbaum. The correlated state in balanced neuronal networks. *bioRxiv*, 2018.

[6] Luke Bashford, Jing Wu, Devapratim Sarma, Kelly Collins, Jeff Ojemann, and Carsten Mehring. Natural movement with concurrent brain-computer interface control induces persistent dissociation of neural activity. In *Proceedings of the 6th International Brain-Computer Interface Meeting*, pages 11–12, CA, USA, 2016.

[7] Vikranth R. Bejjanki, Rava Azeredo da Silveira, Jonathan D. Cohen, and Nicholas B. Turk-Browne. Noise correlations in the human brain and their impact on pattern classification. *PLoS computational biology*, 13(8):e1005674, 2017.

[8] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and Learning-to-learn in networks of spiking neurons. *ArXiv e-prints*, pages 1–17, 2018.

[9] Guillaume Bellec, Franz Scherr, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets. *arXiv preprint*, pages 1–34, 2019.

[10] Guo-qiang Bi and Mu-ming Poo. Synaptic Modifications in Cultured Hippocampal Neurons : Dependence on Spike Timing , Synaptic Strength , and Postsynaptic Cell Type. *The Journal of Neuroscience*, 18(24):10464–10472, 1998.

[11] Guy Bouvier, Claudia Clopath, Célian Bimbard, Jean-Pierre Nadal, Nicolas Brunel, Vincent Hakim, and Boris Barbour. Cerebellar learning using perturbations. *bioRxiv*, page 053785, 2016.

[12] N Brunel. Dynamics of sparsely connected networls of excitatory and inhibitory neurons. *Computational Neuroscience*, 8:183–208, 2000.

[13] Claudia Clopath, Lars Büsing, Eleni Vasilaki, and Wulfram Gerstner. Connectivity reflects coding: A model of voltage-based STDP with homeostasis. *Nature Neuroscience*, 13(3):344–352, 2010.

[14] Claudia Clopath and Wulfram Gerstner. Voltage and spike timing interact in STDP - a unified model. *Frontiers in Synaptic Neuroscience*, 2(JUL):1–11, 2010.

[15] Marlene R. Cohen and Adam Kohn. Measuring and interpreting neuronal correlations. *Nature Neuroscience*, 14(7):811–819, 2011.

[16] Wojciech Marian Czarnecki, Grzegorz Świrszcz, Max Jaderberg, Simon Osindero, Oriol Vinyals, and Koray Kavukcuoglu. Understanding Synthetic Gradients and Decoupled Neural Interfaces. *ArXiv e-prints*, 2017.

[17] Yingying Dong and Arthur Lewbel. Identifying the effect of changing the policy threshold in regression discontinuity models. *The review of economics and statistics*, 97(December):1081–1092, 2015.

[18] E E Fetz and M a Baker. Operantly conditioned patterns on precentral unit activity and correlated responses in adjacent cells and contralateral muscles. *Journal of neurophysiology*, 36(2):179–204, 1973.

[19] Eberhard E Fetz. Volitional control of neural activity: implications for brain-computer interfaces. *The Journal of Physiology*, 579(3):571–579, 2007.

[20] Ila R Fiete, Michale S Fee, and H Sebastian Seung. Model of Birdsong Learning Based on Gradient Estimation by Dynamic Perturbation of Neural Conductances. *Journal of neurophysiology*, 98:2038–2057, 2007.

[21] Ila R Fiete and H Sebastian Seung. Gradient learning in spiking neural networks by dynamic perturbation of conductances. *Physical Review Letters*, 97, 2006.

[22] Elodie Fino, Jean Michel Deniau, and Laurent Venance. Brief subthreshold events can act as Hebbian signals for long-term plasticity. *PLoS ONE*, 4(8), 2009.

[23] Elodie Fino and Laurent Venance. Spike-timing dependent plasticity in the striatum. *Frontiers in synaptic neuroscience*, 2(June):1–10, 2010.

[24] Nicolas Frémaux and Wulfram Gerstner. Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules. *Frontiers in Neural Circuits*, 9(January), 2016.

[25] Samuel J Gershman. Reinforcement learning and causal models. In *Oxford Handbook of Causal Reasoning*, pages 1–32. Oxford university press, 2017.

[26] Matthew D Golub, Steven M Chase, Aaron P Batista, and Byron M Yu. Brain-computer interfaces for dissecting cognitive processes underlying sensorimotor control. *Current Opinion in Neurobiology*, 37:53–58, 2016.

[27] Jordan Guergiuev, Timothy P. Lillicrap, and Blake A. Richards. Towards deep learning with segregated dendrites. *eLife*, 6:1–37, 2017.

[28] York Hagmayer and Philip Fernbach. *Causality in Decision-Making*, volume 1. 2017.

[29] Nicolas Heess, Greg Wayne, David Silver, Timothy Lillicrap, Yuval Tassa, and Tom Erez. Learning Continuous Control Policies by Stochastic Value Gradients. *Advances in Neural Information Processing Systems*, 28:1–13, 2015.

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

[31] Dongsung Huh and Terrence J Sejnowski. Gradient Descent for Spiking Neural Networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[32] Eric Hunsberger and Chris Eliasmith. Spiking Deep Networks with LIF Neurons. *Advances in Neural Information Processing Systems*, 28:1–9, 2015.

[33] Gguido Imbens and Karthik Kalyanaraman. Optimal bandwidth choice for the regression discontinuity estimator. *Review of Economic Studies*, 79(3):933–959, 2012.

[34] Guido W Imbens and Thomas Lemieux. Regression discontinuity designs: A guide to practice. *Journal of Econometrics*, 142(2):615–635, 2008.

[35] Eugene M Izhikevich, John Jay, Hopkins Drive, and San Diego. Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling. *Cerebral Cortex*, 17(September), 2007.

[36] Robin Jacob, Pei Zhu, Marie-andrée Somers, and Howard Bloom. A Practical Guide to Regression Discontinuity. *MDRC*, (July), 2012.

[37] Ingmar Kanitscheider, Ruben Coen-cagli, and Alexandre Pouget. Origin of information-limiting noise correlations. 2015.

[38] Konrad Kording and Peter Konig. Supervised and Unsupervised Learning with Two Sites of Synaptic Integration. *Journal of Computational Neuroscience*, 11:207–215, 2001.

[39] Benjamin Lansdell, Ivana Milovanovic, Cooper Mellema, Eberhard E Fetz, Adrienne L Fairhall, and Chet T Moritz. Reconfiguring motor circuits for a joint manual and BCI task. *arXiv*, 1702.07368:1–18, 2017.

[40] Benjamin James Lansdell and Konrad Paul Kording. Towards learning-to-learn. pages 1–8, 2018.

[41] Mike E. Le Pelley, Oren Griffiths, and Tom Beesley. Associative Accounts of Causal Cognition. In *Oxford Handbook of Causal Reasoning*, volume 1, pages 1–27. Oxford university press, 2017.

[42] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *Nature*, 521, 2015.

[43] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training Deep Spiking Neural Networks Using Backpropagation. *Frontiers in Neuroscience*, 10:1–13, 2016.

[44] Robert Legenstein, Steven M. Chase, Andrew B. Schwartz, Wolfgang Maas, and W. Maass. A Reward-Modulated Hebbian Learning Rule Can Explain Experimentally Observed Network Reorganization in a Brain Control Task. *Journal of Neuroscience*, 30(25):8400–8410, 2010.

[45] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random feedback weights support learning in deep neural networks. *Nature Communications*, 7:13276, 2016.

[46] Y. Loewenstein and H. S. Seung. Operant matching is a generic outcome of synaptic plasticity based on the covariance between reward and neural activity. *Proceedings of the National Academy of Sciences*, 103(41):15224–15229, 2006.

[47] Ioana Elena Marinescu, Konrad Paul Kording, Sofia Triantafillou, and Konrad Paul Kording. Regression Discontinuity Threshold Optimization. (1):1–6, 2019.

[48] Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. Learning Unsupervised Learning Rules. *ArXiv e-prints*, 2018.

[49] Thomas Miconi. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *eLife*, 6:1–24, 2017.

[50] Ivana Milovanovic, Robert Robinson, Eberhard E. Fetz, and Chet T. Moritz. Simultaneous and independent control of a brain-computer interface and contralateral limb movement. *Brain-Computer Interfaces*, 2621(September):1–12, 2015.

[51] CT Moritz and EE Fetz. Volitional control of single cortical neurons in a brain–machine interface. *Journal of neural engineering*, 8, 2011.

[52] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate Gradient Learning in Spiking Neural Networks. *IEEE SPM*, pages 1–21, 2019.

[53] A. Ngezahayo, M. Schachner, and Alain Artola. Synaptic activity modulates the induction of bidirectional synaptic changes in adult mouse hippocampus. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 20(7):2451–8, 2000.

[54] Amy L. L Orsborn, Helene G. G Moorman, Simon A. A Overduin, Maryam M. M Shanechi, Dragan F. F Dimitrov, and Jose M. M Carmena. Closed-Loop Decoder Adaptation Shapes Neural Plasticity for Skillful Neuroprosthetic Control. *Neuron*, 82(6):1380–1393, 2014.

[55] Verena Pawlak, Jeffery R. Wickens, Alfredo Kirkwood, and Jason N.D. Kerr. Timing is not everything: Neuromodulation opens the STDP gate. *Frontiers in Synaptic Neuroscience*, 2(OCT):1–14, 2010.

[56] Judea Pearl. *Causality: models, reasoning and inference*. Cambridge Univ Press, 2000.

[57] J Peters, D Janzing, and B Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.

[58] Michael Pfeiffer and Thomas Pfeil. Deep Learning With Spiking Neurons : Opportunities and Challenges. *Frontiers in Neuroscience*, 12(October), 2018.

[59] Dale Purves, G J Augustine, D Fitzpatrick, W C Hall, A S LaMantia, J O McNamara, and L E White. Neuroscience, 2008. *De Boeck, Sinauer, Sunderland, Mass*, 2014.

[60] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *Proceedings of the 31st International Conference on Machine Learning, PMLR*, 32(2):1278–1286, 2014.

[61] Patrick T. Sadtler, Kristin M. Quick, Matthew D. Golub, Steven M. Chase, Stephen I. Ryu, Elizabeth C. Tyler-Kabara, Byron M. Yu, and Aaron P. Batista. Neural constraints on learning. *Nature*, 512(7515):423–426, aug 2014.

[62] Wolfram Schultz. Getting formal with dopamine and reward. *Neuron*, 36(2):241–263, 2002.

[63] Geun Hee Seol, Jokubas Ziburkus, Shiyong Huang, Lihua Song, In Tae Kim, Kogo Takamiya, Richard L Huganir, Hey-kyoung Lee, and Alfredo Kirkwood. Neuromodulators Control the Polarity of Spike-Timing-Dependent Synaptic Plasticity. *Neuron*, 55(6):919–929, 2007.

[64] Sebastian Seung. Learning in Spiking Neural Networks by Reinforcement of Stochastics Transmission. *Neuron*, 40:1063–1073, 2003.

[65] M. Shafi, Y. Zhou, J. Quintana, C. Chow, J. Fuster, and M. Bodner. Variability in neuronal activity in primate cortex during working memory tasks. *Neuroscience*, 146(3):1082–1108, 2007.

[66] Eric Shea-Brown, Krešimir Josić, Jaime De La Rocha, and Brent Doiron. Correlation and synchrony transfer in integrate-and-fire neurons: Basic properties and consequences for coding. *Physical Review Letters*, 100(10):1–4, 2008.

[67] Per Jesper Sjöström, Gina G Turrigiano, and Sacha B Nelson. Endocannabinoid-Dependent Neocortical Layer-5 LTD in the Absence of Postsynaptic Spiking. *Journal of Neurophysiology*, 92(6):3338–3343, 2004.

[68] Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2017.

[69] Richard S. Sutton, David Mcallester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. *Advances in Neural Information Processing Systems*, 12:1057–1063, 1999.

[70] Guangzhi Tang, Arpit Shah, and Konstantinos P Michmizos. Spiking Neural Network on Neuromorphic Hardware for Energy-Efficient Unidimensional SLAM. *ArXiv e-prints*, 2019.

[71] Amirhossein Tavanaei, Masoud Ghodrati, and Saeed Reza. Deep learning in spiking neural networks. *Neural Networks*, 111:47–63, 2019.

[72] F Theunissen and John P Miller. Temporal Encoding in Nervous Systems : A Rigorous Definition. *Journal of computational neuroscience*, 2:149–162, 1995.

[73] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. *IEEE International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[74] Ronald Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:299–256, 1992.

[75] Roy a RA Wise. Dopamine, learning and motivation. *Nature reviews. Neuroscience*, 5(6):483–494, jun 2004.

[76] Jim Woodward. Interventionist theories of causation in psychological perspective. In *Causal learning: psychology, philosophy and computation*. Oxford university press, New York, 2007.

[77] Xiaohui Xie and H. Sebastian Seung. Learning in neural networks by reinforcement of irregular spiking. *Physical Review E*, 69, 2004.

[78] Man Yi Yim, Ad Aertsen, and Arvind Kumar. Significance of Input Correlations in Striatal Function. *PLoS Comput Biol*, 7(11), 2011.

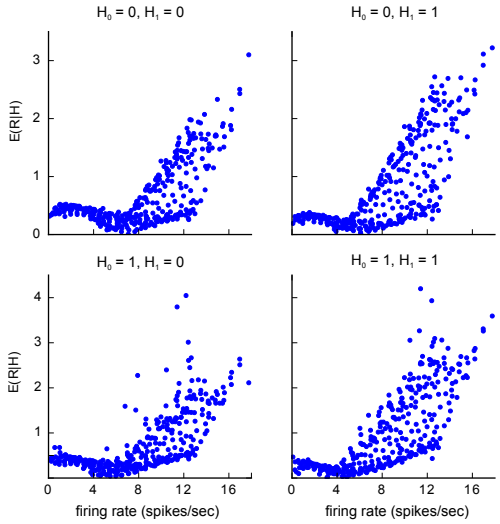

Figure 7: Supplementary Figure 1. **Validity of Assumption 1 for LIF network model.** Independence of $\mathbb{E}(R|H)$ from $\mathbf{w}$. Over a range of network weights, the expected reward is shown here for all possible values of H_0, H_1. For low firing rates, the expected reward does not change much, regardless of network weight, showing the assumption is valid. For high firing rates this is not the case.

Supplementary material

1 Validity of causal assumptions for a leaky integrate-and-fire (LIF) network

The relation derived in the Methods between causal effect and policy gradient methods relied on three assumptions:

1. $\mathbb{E}(R|H)$ is independent of w,
2. $\frac{\partial}{\partial w_i}\mathbb{E}(H_i|H_{j\neq i})$ is independent of $H_{j\neq i}$,
3. $H_{j\neq i}$ satisfies the backdoor criterion with respect to $H_i \rightarrow R$.

These assumptions are reasonable for certain parameter regimes of the LIF network used in this analysis (Supplementary Figure 1,2,3).

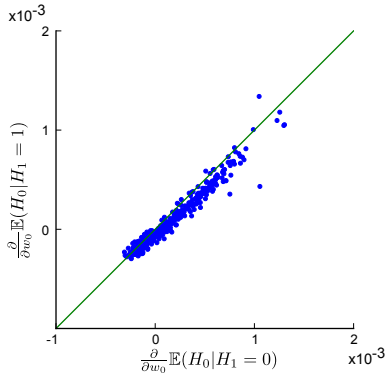

Figure 8: Supplementary Figure 2. **Validity of Assumption 2 for LIF network model.** Independence of $\frac{\partial}{\partial w_0}\mathbb{E}(H_0|H_1)$ from H_1. Over a range of network weights, the gradient terms $\frac{\partial}{\partial w_0}\mathbb{E}(H_0|H_1)$ are roughly equal, regardless of the value of H_1.

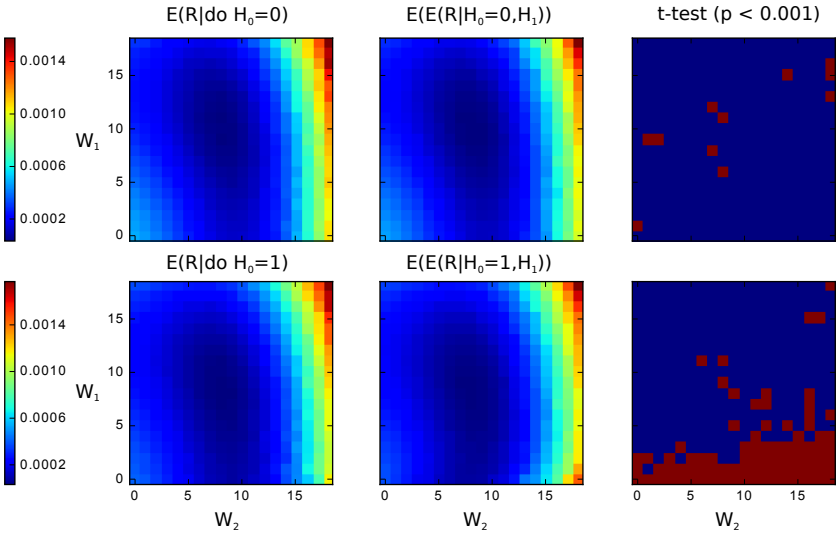

Figure 9: Supplementary Figure 3. **Validity of Assumption 3 for LIF network model.** Verifying the interventional distribution of expected reward (left) matches the backdoor corrected expected reward from observational data (middle). Samples from each distribution are compared via t-test ($\alpha = 0.001$, Bonferroni corrected), to see if they have significantly different means (right; red = significant). The majority of network weights result in activity in which the interventional distribution matches the corrected observational distribution.