

## RESEARCH ARTICLE

## Neural spiking for causal inference and learning

Benjamin James Lansdell <sup>1†\*</sup>, Konrad Paul Kording<sup>1,2</sup>

**1** Department of Bioengineering, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America, **2** Department of Neuroscience, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America

†Current address: Department of Developmental Neurobiology, St Jude Children's Research Hospital, Memphis, Tennessee, United States of America

\* [ben.lansdell@stjude.org](mailto:ben.lansdell@stjude.org)

 OPEN ACCESS

**Citation:** Lansdell BJ, Kording KP (2023) Neural spiking for causal inference and learning. *PLoS Comput Biol* 19(4): e1011005. <https://doi.org/10.1371/journal.pcbi.1011005>

**Editor:** Ulrik R. Beierholm, Durham University, UNITED KINGDOM

**Received:** May 29, 2022

**Accepted:** March 7, 2023

**Published:** April 4, 2023

**Copyright:** © 2023 Lansdell, Kording. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** All python code used to run simulations and generate figures is available at <https://github.com/benlansdell/rdd>.

**Funding:** KPK received NIH grant. "Quantifying causality for neuroscience" 1R01EB028162-01 <https://braininitiative.nih.gov/funded-awards/quantifying-causality-neuroscience> The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors state no competing interests.

## Abstract

When a neuron is driven beyond its threshold, it spikes. 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-based learning. Importantly, neither activity of upstream neurons, which act as confounders, nor downstream non-linearities bias the results. We show how spiking enables neurons to solve causal estimation problems and that local plasticity can approximate gradient descent using spike discontinuity learning.

## Author summary

Despite significant research, models of spiking neural networks still lag behind artificial neural networks in terms of performance in machine learning and modeling cognitive tasks. Given this, we may wonder, why do neurons spike? A key problem that must be solved in any neural network is the credit assignment problem. That is, how does a neuron know its effect on downstream computation and rewards, and thus how it should change its synaptic weights to improve? Artificial neural networks solve this problem with the back-propagation algorithm. We are still seeking to understand how biological neural networks effectively solve this problem. In this work we show that the discontinuous, all-or-none spiking response of a neuron can in fact be used to estimate a neuron's causal effect on downstream processes. Inspired by methods from econometrics, we show that the thresholded response of a neuron can be used to get at that neuron's unique contribution to a reward signal, separating it from other neurons whose activity it may be correlated with. This proposal provides insights into a novel function of spiking that we explore in simple networks and learning tasks.

## Introduction

Most nervous systems communicate and process information utilizing spiking. Yet machine learning mostly uses artificial neural networks with continuous activities. Computationally, despite a lot of recent progress [1–5], it remains challenging to create spiking neural networks that perform comparably to continuous artificial networks. Instead, spiking is generally seen as a disadvantage—it is difficult to propagate gradients through a discontinuity, and thus to train spiking networks. This disparity between biological neurons that spike and artificial neurons that are continuous raises the question, what are the computational benefits of spiking?

There are, of course, pragmatic reasons for spiking: spiking may be more energy efficient [6, 7], spiking allows for reliable transmission over long distances [8], and spike timing codes may allow for more transmission bandwidth [9]. Yet, despite these ideas, we may still wonder if there are computational benefits of spikes that balance the apparent disparity in the learning abilities of spiking and artificial networks.

A key computational problem in both biological and artificial settings is the credit assignment problem [10]. 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 [11–15] 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 [16–22]. However, despite important special cases [17, 19, 23], in general it is not clear how a neuron may know its own noise level. Thus we may wonder if neurons estimate their causal effect without random perturbations.

How else 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, a given neuron spiking is associated with a different network state than that neuron not spiking. And it is 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 key insight in this paper is that 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. 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 learning rules that allows a network of neurons to learn to maximize reward, particularly in the presence of

confounded inputs. 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

### Causal inference in neural networks

Though not previously recognized as such, the credit assignment problem is a causal inference problem: how can a neuron know its causal effect on an output and subsequent reward? This section shows how this idea can be made more precise.

Causal effects are formally defined in the context of a certain type of probabilistic graphical model—the causal Bayesian network—while a spiking neural network is a dynamical, stochastic process. Thus before we can understand how a neuron can use its spiking discontinuity to do causal inference we must first understand how a causal effect can be defined for a neural network. We present two results:

- First, we lay out how a neural network can be described as a type of causal Bayesian network (CBN).
- Second, assuming such a CBN, we relate the causal effect of a neuron on a reward function to a finite difference approximation of the gradient of reward with respect to neural activity.

Together these results show how causal inference relates to gradient-based learning, particularly in spiking neural networks.

**Result I. From dynamic neural networks to probabilistic graphical models.** We develop this idea in the context of a supervised learning setting. To formalize causal effects in this setting, we thus first have to think about how supervised learning might be performed by a spiking, dynamically integrating network of neurons (see, for example, the solution by Guerguiev et al 2016 [24]). That is, for simplicity, to avoid problems to do with temporal credit assignment, we consider a neural network that receives immediate feedback/reward on the quality of the computation, potentially provided by an internal critic (similar to the setup of [24]). Spiking neural networks generally operate dynamically where activities unfold over time, yet supervised learning in an artificial neural network typically has no explicit dynamics—the state of a neuron is only a function of its current inputs, not its previous inputs.

To accommodate these differences, we consider the following learning problem. Consider a population of  $N$  neurons. Let  $v_i(t)$  denote the membrane potential of neuron  $i$  at time  $t$ , having leaky integrate-and-fire dynamics:

$$\dot{v}_i = -g_L v_i + \sum w_{ij}(x_j + \eta_j), \quad v(t^+) = v_r, \text{ if } v(t) = \theta.$$

for leak term  $g_L$ , reset voltage  $v_r$  and threshold  $\theta$ . The network is assumed to have a feedforward structure. The neurons receive inputs from an input layer  $\mathbf{x}(t)$ , along with a noise process  $\eta_j(t)$ , weighted by synaptic weights  $w_{ij}$ . The network is presented with this input stimulus for a fixed period of  $T$  seconds. Let the variable  $h_i(t)$  denote the neuron's spiking indicator function:  $h_i(t) = \sum \delta(t - t_s)$  if neuron  $i$  spikes at times  $t_s$ . Post-synaptic current,  $s_i(t)$ , is generated according to the dynamics  $\tau_s \dot{s} = -s_i + h_i(t)$ . On the basis of the output of the neurons, a reward signal  $r$  is generated, assumed to be a function of the filtered currents  $\mathbf{s}(t)$ :  $r(t) = r(\mathbf{s}(t))$ . Here we assume that  $T$  is sufficiently long for the network to have received an input, produced an output, and for feedback to have been distributed to the system (e.g. in the form of dopamine signaling a reward prediction error [25]). Given this network, then, the learning problem is for

each neuron to adjust its weights to maximize reward, using an estimate of its causal effect on reward.

As described in the introduction, to apply the spiking discontinuity method to estimate causal effects, we have to track how close a neuron is to spiking. Over the time period  $T$ , to distinguish between just-above-threshold inputs from well-above-threshold inputs, we also consider the input drive to the neuron:  $u_i(t)$ , which is the integrated input to the neuron, except without the reset mechanism.

$$\dot{u}_i = -g_L u_i + \sum w_{ij}(x_j + \eta_j).$$

By tracking integrated inputs with a reset mechanism, then the value  $Z_i = \max_{0 \leq t \leq T} u_i(t)$  tells us if neuron  $i$  received inputs that placed it well above threshold, or just above threshold.

To compute what the network’s output is for the given input over  $T$  seconds, we define a set of random variables,  $\mathbf{X}, \mathbf{Z}, \mathbf{H}, \mathbf{S}, R$  that aggregate the underlying dynamical (and spiking) variables,  $\mathbf{x}(t), \mathbf{z}(t), \mathbf{h}(t), \mathbf{s}(t)$  and  $r(t)$ , respectively. Specifically, we define aggregating functionals  $f_{[\cdot]}$  to be summaries of the underlying dynamical variable activities. When the network dynamics are irregular [26], these aggregate variables will be approximately independent across subsequent windows of sufficient duration  $T$ . The dynamics given by the noisy LIF network generate an ergodic Markov process with a stationary distribution. That is, from a dynamical network, we have a set of random variables  $\Phi$  that summarize the state of the network and can be considered I.I.D. draws from some distribution, which depends on the network’s weights and other parameters,  $\Theta$  (e.g. noise magnitude and correlation):  $\Phi := (\mathbf{X}, \mathbf{Z}, \mathbf{H}, \mathbf{S}, R) \sim \rho(\cdot; \Theta)$ .

The functionals are required to only depend on one underlying dynamical variable. The spiking discontinuity approach requires that  $H$  is an indicator functional, simply indicating the occurrence of a spike or not within window  $T$ ; it could instead be defined directly in terms of  $Z$ . The random variable  $Z$  is required to have the form defined above, a maximum of the integrated drive. The choice of functionals is required to be such that, if there is a dependence between two underlying dynamical variables (e.g.  $h$  and  $x$ ), then there is also some statistical dependence between these variables in the aggregated variables. I.e. a trivial functional  $f_R(r) = 0$  would destroy any dependence between  $X$  and  $R$ . Given these considerations, for the subsequent analysis, the following choices are used:

$$\begin{aligned} \mathbf{X} &= f_X(\mathbf{x}) = \frac{1}{T} \int_0^T \mathbf{x}(t) dt, \\ \mathbf{Z} &= f_Z(\mathbf{u}) = \max_{0 \leq t \leq T} u_i(t), \\ H_i &= f_H(h_i) = \begin{cases} 1, & \int_0^T h_i(t) dt > 0, \\ 0, & \text{otherwise} \end{cases} \\ \mathbf{S} &= f_S(\mathbf{s}) = \mathbf{s}(T) \\ R &= f_R(r) = r(\mathbf{s}(T)) \end{aligned}$$

These choices were made since they showed better empirical performance than, e.g. taking the mean over  $T$  for  $\mathbf{S}$  and  $R$ . Along with the parameters of the underlying dynamical neural network, these choices determine the form of the distribution  $\rho$ . The learning problem can now be framed as: how can parameters  $\Theta$  be adjusted such that  $\mathbb{E}_\rho(R)$  is maximized? Below we show how the causal effect of a neuron on reward can be defined and used to maximize this reward.

**A causal Bayesian network of a neural network, and a neuron’s causal effect on reward.** What is the causal effect of a neuron on a reward signal? Given the distribution  $\rho$  over the random variables  $(\mathbf{X}, \mathbf{Z}, \mathbf{H}, \mathbf{S}, R)$ , we can use the theory of causal Bayesian networks to formalize the causal effect of a neuron’s activity on reward [27].

**Definition 1.** In a causal Bayesian network, the probability distribution  $\rho$  is factored according to a graph,  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  [27]. The edges  $\mathcal{E}$  in the graph represent causal relationships between the nodes  $\mathcal{N}$ ; the graph is both directed and acyclic (a DAG). The standard definition of a causal Bayesian model imposes two constraints on the distribution  $\rho$ , relating to:

1. Conditional Independence: nodes are conditionally independent of their non-descendants given their parents,

$$\rho(\mathbf{Y} = \mathbf{y}) = \prod_{n \in \mathcal{N}} \Pr(Y_n = y_n | \text{Pa}_n),$$

where  $\text{Pa}_n$  represents the parents of node  $n$ .

2. The Effect of Interventions: when a variable is *intervened* upon, forced to take a given value, the edge between that variable and its parents is severed, changing the data-generating distribution. That is, if we intervene on a node  $j$ , then the interventional distribution  $\rho_{Y_j=y_j}$  is

$$\rho_{Y_j=y_j}(\mathbf{Y} = \mathbf{y}) = \delta(Y_j = y_j) \prod_{n \neq j} \Pr(Y_n = y_n | \text{Pa}_n).$$

Where node  $j$  has been forced to take a value,  $y_j$ , and the dependence on its parents has been severed.

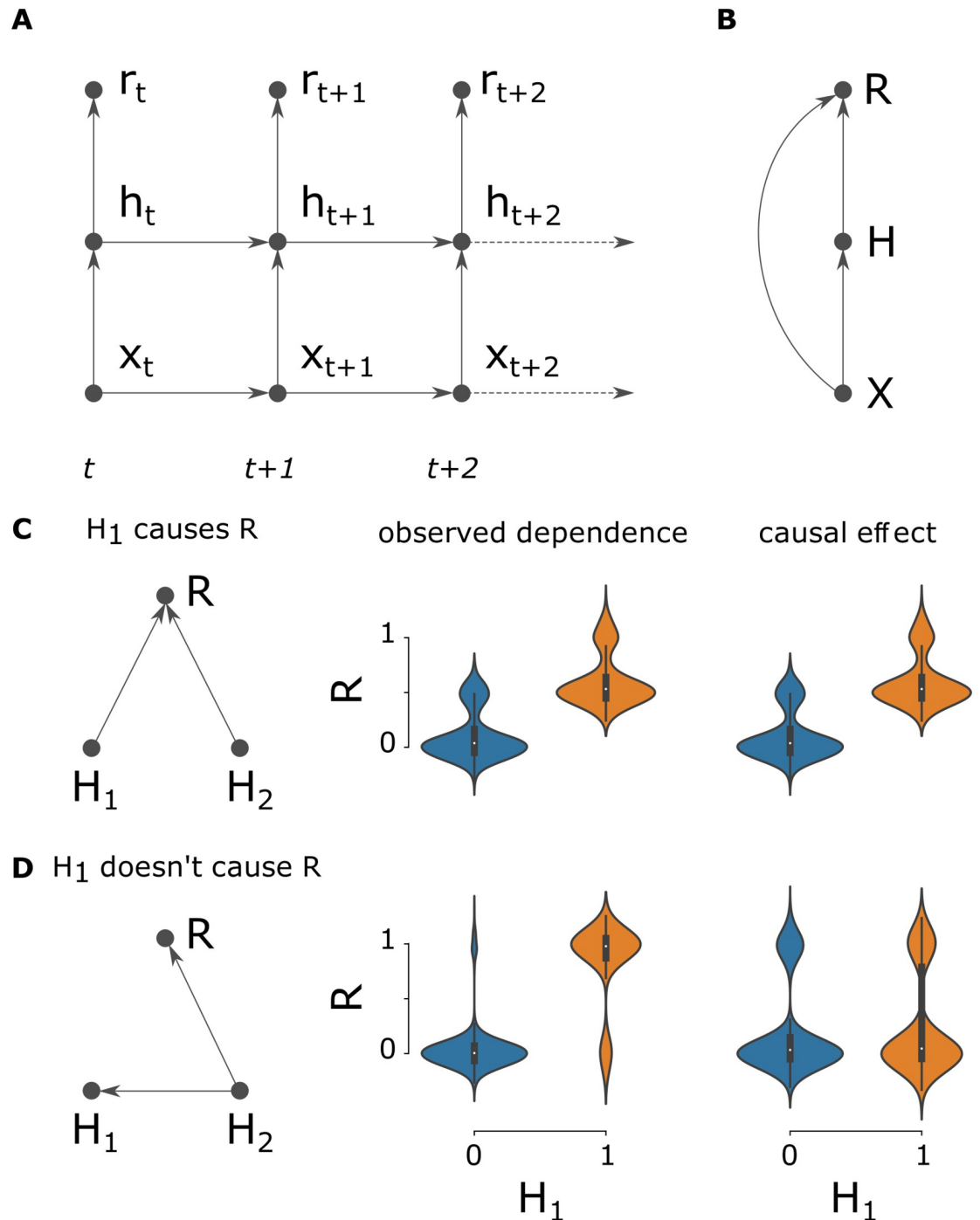
To use this theory, first, we describe a graph  $\mathcal{G}_\Phi$  such that  $\rho$  is compatible with the conditional independence requirement of the above definition. Consider the ordering of the variables  $\Phi$  that matches the feedforward structure of the underlying dynamic feedforward network (Fig 1A). From this ordering we construct the graph  $\mathcal{G}_\Phi$  over the variables  $\Phi$  (Fig 1B). This graph respects the order of variables implied in Fig 1A, but it is over-complete, in the sense that it also contains a direct link between  $X$  and  $R$ . This direct link between  $X$  and  $R$ , though absent in the underlying dynamical model, cannot be ruled out in a distribution over the aggregate variables, so must be included. The graph is directed, acyclic and fully-connected. Being fully connected in this way guarantees that we can factor the distribution  $\rho$  with the graph  $\mathcal{G}_\Phi$  and it will obey the conditional independence criterion described above.

The second criterion is that the graph can be used to describe what happens when interventions are made. Fully spelling this out is beyond the scope of this study, so here we assume the interventional distributions on nodes of  $\mathcal{G}_\Phi$  factor the distribution  $\rho$  as expected in the definition above. This is reasonable since, for instance, intervening on the underlying variable  $h_i(t)$  (to enforce a spike at a given time), *would* sever the relation between  $Z_i$  and  $H_i$  as dictated by the graph topology. Taken together, this means the graph  $\mathcal{G}_\Phi$  describes a causal Bayesian network over the distribution  $\rho$ .

Given this causal network, we can then define 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’  $H_i$  is exogenously assigned. The causal effect of neuron  $i$  on reward  $R$  is defined as:

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

where  $\text{do}$  represents the do-operator, notation for an intervention [27]. Because of the fact that the aggregated variables maintain the same ordering as the underlying dynamic variables,



**Fig 1. Graphical models formalizing a neuron's causal effect on reward.** (A) The dynamic spiking network model. The state at time bin  $t$  depends on both the previous state and a hierarchical dependence between inputs  $x_t$ , neuron activities  $h_t$ , and the reward signal  $r_t$ . Omitted for clarity are the extra variables that determine the network state ( $v(t)$  and  $s(t)$ ). (B) To formulate the supervised learning problem, these variables are aggregated in time to produce summary variables of the state of the network during the simulated window. We consider these variables being drawn IID from a distribution  $(X, Z, H, S, R) \sim \rho$ . Intervening on the underlying dynamic variables changes the distribution  $\rho$  accordingly. E.g. severing the connection from  $x_t$  to  $h_t$  for all  $t$  renders  $H$  independent of  $X$ . Thus the graphical model over  $(X, Z, H, R)$  has the same hierarchy (ordering) as the underlying dynamical model. However, the aggregate variables do not fully summarize the state of the network throughout the simulation. Therefore, unlike the structure in the underlying dynamics,  $H$  may not fully separate  $X$  from  $R$ —we must allow for the possibility of a direct connection from  $X$  to  $R$ . (C) and (D) are simple examples illustrating the difference between observed dependence and causal effect. We have omitted the dependence on  $X$  for simplicity. Violin plots show reward when  $H_1$  is active or inactive, without (left subplot) and with (right) intervening on  $H_1$ . (C) If  $H_1$  and  $H_2$  are independent, the observed dependence matches the causal effect. (D) If  $H_2$  causes  $H_1$  then  $H_2$  is an unobserved confounder, and the observed dependence and causal effects differ.

<https://doi.org/10.1371/journal.pcbi.1011005.g001>

there is a well-defined sense in which  $R$  is indeed an effect of  $H_i$ , not the other way around, and therefore that the causal effect  $\beta_i$  is a sensible, and not necessarily zero, quantity (Fig 1C and 1D). That is to say, it makes sense to associate with each neuron, for each stimulus, what its causal effect is on the output and thus reward.

**Result II: Causal effects and finite difference approximation of gradients.** The causal effect  $\beta_i$  is an important quantity for learning: if we know how a neuron contributes to the reward, the neuron can change its behavior to increase it. More specifically, in a spiking neural network, the causal effect can be seen as a type of finite difference approximation of the partial derivative (reward with a spike vs reward without a spike). That is, since we assume that  $R$  is function of some filtered neural output of the network, then it can be shown that the causal effect,  $\beta_i$ , under certain assumptions, does indeed approximate the reward gradient  $\frac{\partial R}{\partial S_i}$ . That is, there is a sense in which:

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

More specifically:

$$\beta_i \approx \Delta_s \mathbb{E}(D_i R|\text{do}(H_i = 0)),$$

where  $D_i R$  is a random variable that represents the finite difference operator of  $R$  with respect to neuron  $i$ 's firing, and  $\Delta_s$  is a constant that depends on the spike kernel  $\kappa$  and acts here like a kind of finite step size. Refer to the methods section for the derivation. This result establishes a connection between causal inference and gradient-based learning. It suggests that methods from causal inference may provide efficient algorithms to estimate reward gradients, and thus can be used to optimize reward. In this way the causal effect is a relevant quantity for learning.

### Estimating a neuron's causal effect using the spiking discontinuity

Having understood how the causal effect of a neuron can be defined, and how it is relevant to learning, we now consider how to estimate it. The key observation of this paper is to note that a discontinuity can be used to estimate causal effects, without randomization, but while retaining the benefits of randomization. We will refer to this approach as the Spiking Discontinuity Estimator (SDE). This estimator is equivalent to the regression discontinuity design (RDD) approach to causal inference which is popular in economics [28, 29].

As outlined in the introduction, the idea is that inputs that place a neuron close to its spiking threshold can be used in an unbiased causal effect estimator. For inputs that place the neuron just below or just above its spiking threshold, the difference in the state of the rest of the network becomes 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. Statistically, within a small interval around the threshold spiking becomes *as good as random* [28–30]. In this way the spiking discontinuity may allow neurons to estimate their causal effect.

To demonstrate the idea that a neuron can use its spiking non-linearity to estimate causal effects, here we analyze a simple two neuron network obeying leaky integrate-and-fire (LIF) dynamics. The neurons receive a shared scalar input signal  $x(t)$ , with added separate noise inputs  $\eta_i(t)$ , that are 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 [31], thereby introducing confounding. At the end of a trial period  $T$ , the neural output determines a reward signal  $R$ . Most aspects of causal inference can be investigated in a simple, few-variable model such as this [32], thus demonstrating that a neuron can estimate a causal

effect in this simple case is an important first step to understanding how it can do so in a larger network.

To give intuition into how this confounding problem manifests in a neural learning setting, consider how a neuron can estimate its causal effect. A basic model of a neuron's effect on reward is that it can be estimated from the following piece-wise constant model of the reward function:

$$R = \gamma_i + \beta_i H_i, \quad (2)$$

That is, there is some baseline expected reward,  $\gamma_i$ , and a neuron-specific contribution  $\beta_i$ , where  $H_i$  represents the spiking indicator function for neuron  $i$  over the trial of period  $T$ . Then denote by  $\beta_i$  the causal effect of neuron  $i$  on the resulting reward  $R$ .

Estimating  $\beta_i$  naively as

$$\beta_i^{OD} = \mathbb{E}(R|H_i = 1) - \mathbb{E}(R|H_i = 0), \quad (3)$$

is biased if  $H_i$  is correlated with other neuron's activity (Fig 2A). Call this naive estimator the *observed dependence*.

Instead, we can estimate  $\beta_i$  only for inputs that placed the neuron close to its threshold. That is, let  $Z_i$  be the maximum integrated neural drive to the neuron over the trial period. If  $\theta$  is the neuron's spiking threshold, then a maximum drive above  $\theta$  results in a spike, and below  $\theta$  results in no spike. 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 the trial period, we can track when inputs placed the neuron close to its threshold, and marginally super-threshold inputs can be distinguished from well-above-threshold inputs, as required for SDE (Fig 2C). Let  $p$  be a window size within which we are going to call the integrated inputs  $Z_i$  'close' to threshold, then the SDE estimator of  $\beta_i$  is:

$$\beta_i^{SDE} = \mathbb{E}(R|\theta \leq Z_i < \theta + p) - \mathbb{E}(R|\theta - p < Z_i < \theta). \quad (4)$$

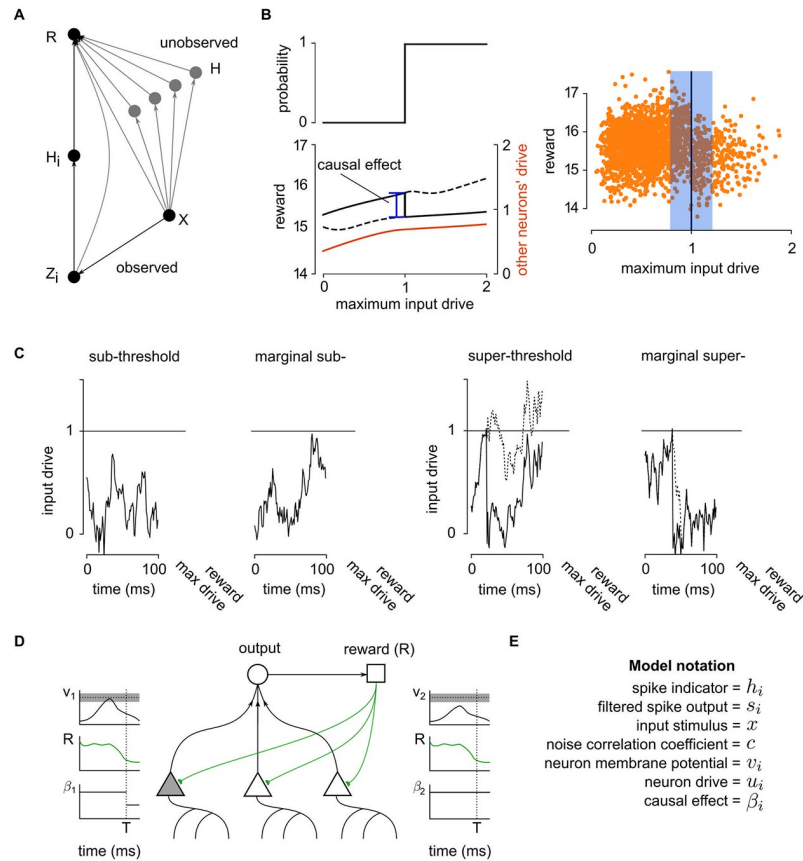
A one-order-higher model of the reward adds a linear correction, resulting in the piece-wise linear model of the reward function:

$$R = \gamma_i + \beta_i H_i + [\alpha_{ri} H_i + \alpha_{li} (1 - H_i)](Z_i - \theta), \quad (5)$$

where  $\gamma_i$ ,  $\alpha_{li}$  and  $\alpha_{ri}$  are the linear regression parameters. This higher-order model can allow for larger window sizes  $p$ , and thus a lower variance estimator. Call the causal effect estimate using the piecewise constant model  $\beta_i^{SDE,0}$  and the causal effect estimate using the piecewise-linear model  $\beta_i^{SDE,1}$ . Both such models are explored in the simulations below.

This approach relies on some assumptions. First, a neuron assumes its effect on the expected reward can be written as a function of  $Z_i$  which has a discontinuity at  $Z_i = \theta$ , such that, in the neighborhood of  $Z_i = \theta$ , the function can be approximated by either its 0-degree (piecewise constant version) or 1-degree Taylor expansion (piecewise linear). This approach also assumes that the input variable  $Z_i$  is itself a continuous variable. This is the case in simulations explored here.

To summarize the idea: for a neuron to apply spiking discontinuity estimation, it simply must track if it was close to spiking, whether it spiked or not, and observe the reward signal. 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 for an unbiased estimate of its own causal effect (Fig 2D and 2E). Below we gain intuition about how the estimator works, and how it differs from the naive estimate.

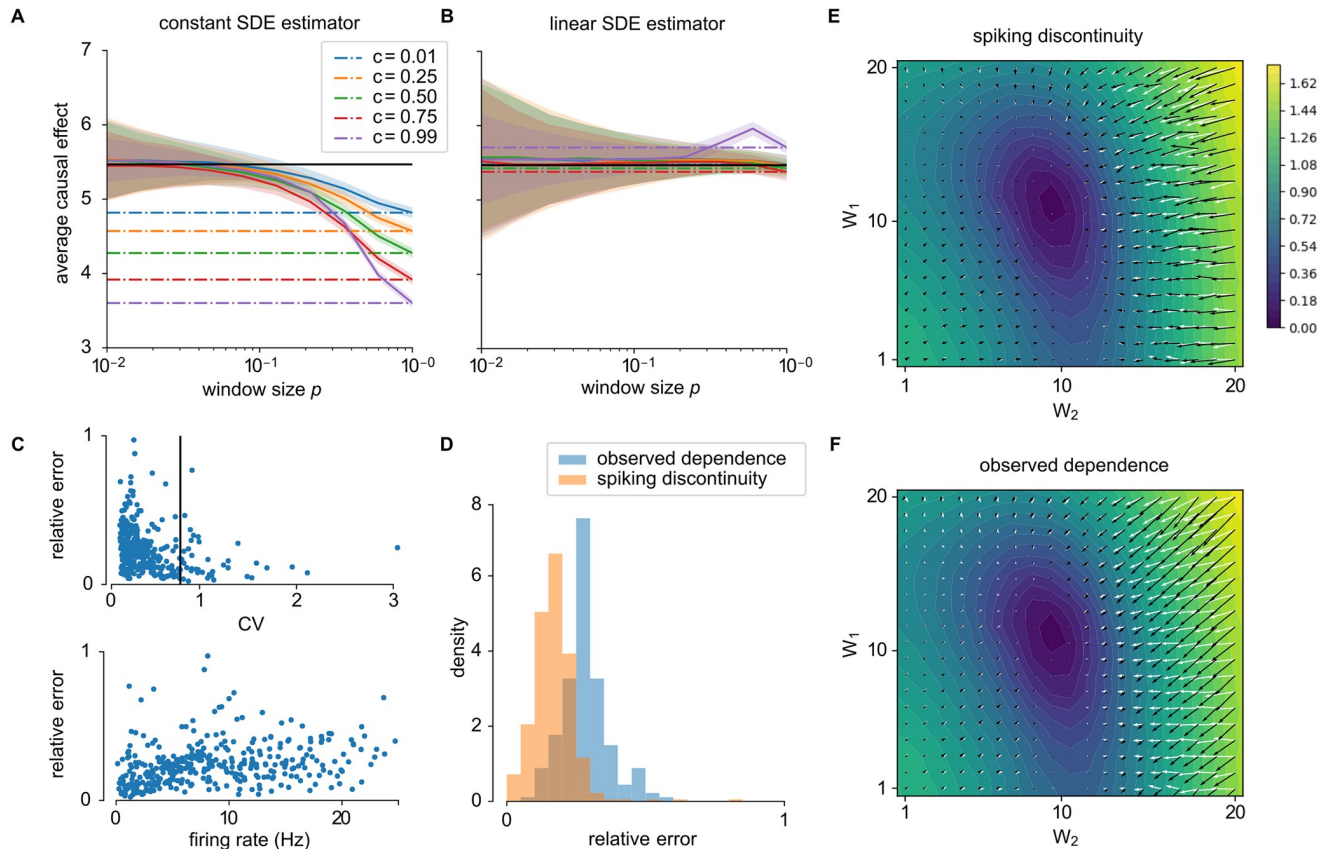


**Fig 2. Using the spiking discontinuity to estimate causal effects.** (A) Graphical model describing neural network. Neuron  $H_i$  receives input  $X$ , which contributes to drive  $Z_i$ . If drive is above the spiking threshold, then  $H_i$  is active. The activity contributes to reward  $R$ . Though 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. For clarity, other neuron's  $Z$  variables have been omitted from this graph. Unobserved dependencies of  $X \rightarrow R$  and  $Z_i \rightarrow R$  are shown here, even though not part of the underlying dynamical model, such dependencies in the graphical model may still exist, as discussed in the text. (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 spiking discontinuity 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  $\beta$ . (E) Model notation.

<https://doi.org/10.1371/journal.pcbi.1011005.g002>

### A simple empirical demonstration of SDE

Simulating this simple two-neuron network shows how a neuron can estimate its causal effect using the SDE (Fig 3A and 3B). To show how it removes confounding, we implement both the piece-wise constant and piece-wise linear models for a range of window sizes  $p$ . When  $p$  is large, the piece-wise constant model corresponds to the biased observed-dependence estimator,  $\beta_i^{OD}$ , while small  $p$  values approximate the SDE estimator and result in an unbiased estimate (Fig 3A). The window size  $p$  determines the variance of the estimator, as expected from theory [33]. The piece-wise linear model, Eq (5), is more robust to confounding (Fig 3B), allowing larger  $p$  values to be used. Thus the linear correction that is the basis of many RDD implementations



**Fig 3. Estimating reward gradient with spiking discontinuity in two-neuron network.** (A) Estimates of causal effect (black line) using a constant spiking discontinuity 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 model is unbiased over larger window sizes and more highly correlated activity (high  $c$ ). (C) Relative error in estimates of causal effect  $\beta_i^{SDE,0}$  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, spiking discontinuity estimates  $\beta_i^{SDE,0}$  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 spiking discontinuity estimate  $\beta_i^{SDE,0}$  (E) and observed dependence (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.

<https://doi.org/10.1371/journal.pcbi.1011005.g003>

[28] allows neurons to more readily estimate their causal effect. This linear dependence on the maximal voltage of the neuron may be approximated by plasticity that depends on calcium concentration; such implementation issues are considered in the discussion.

The same simple model can be used to estimate the dependence of the quality of spike discontinuity estimates on network parameters. To investigate the robustness of this estimator, we systematically vary the weights,  $w_i$ , of the network. This allows for an exploration SDE’s performance in a range of network states. SDE works better when activity is fluctuation-driven and at a lower firing rate (Fig 3C). Thus spiking discontinuity is most applicable in irregular but synchronous activity regimes [26]. Over this range of network weights, spiking discontinuity is less biased than the observed dependence (Fig 3D). Overall, corrected estimates based on spiking considerably improve on the naive implementation.

### Learning with a discontinuity-based causal effect estimator

We just showed that the spiking discontinuity allows neurons to estimate their causal effect. We can implement this as a simple learning rule that illustrates how knowing the causal effect

impacts learning. We derive an online learning rule that estimates  $\beta$ , and the linear model parameters, if needed. That is, let  $\mathbf{u}_i$  be the vector of parameters required to estimate  $\beta_i$  for neuron  $i$ . For the piece-wise constant reward model that is  $\mathbf{u}_i = [\gamma_i, \beta_i]$  and for the piece-wise linear model it is  $\mathbf{u}_i = [\gamma_i, \beta_i, \alpha_{ri}, \alpha_{li}]$ . Then the learning rule takes the form:

$$\Delta \mathbf{u}_i = \begin{cases} -\eta[\mathbf{u}_i^T \mathbf{a}_i - R] \mathbf{a}_i, & \theta - p \leq Z_i < \theta + p \text{ (just/almost spikes);} \\ 0, & \text{(otherwise),} \end{cases} \quad (6)$$

where  $\eta$  is a learning rate, and  $\mathbf{a}_i$  are drive-dependent terms (see [Methods](#) for details and the derivation). Using this learning rule to update  $\mathbf{u}_i$ , along with the relation

$$\frac{\partial}{\partial w_i} \mathbb{E}(R) \approx \frac{\partial \mathbb{E}(H_i)}{\partial w_i} \beta_i, \quad (7)$$

allows us to update the weights according to a stochastic gradient-like update rule:

$$\Delta w_i \propto \frac{\partial}{\partial w_i} \mathbb{E}(R),$$

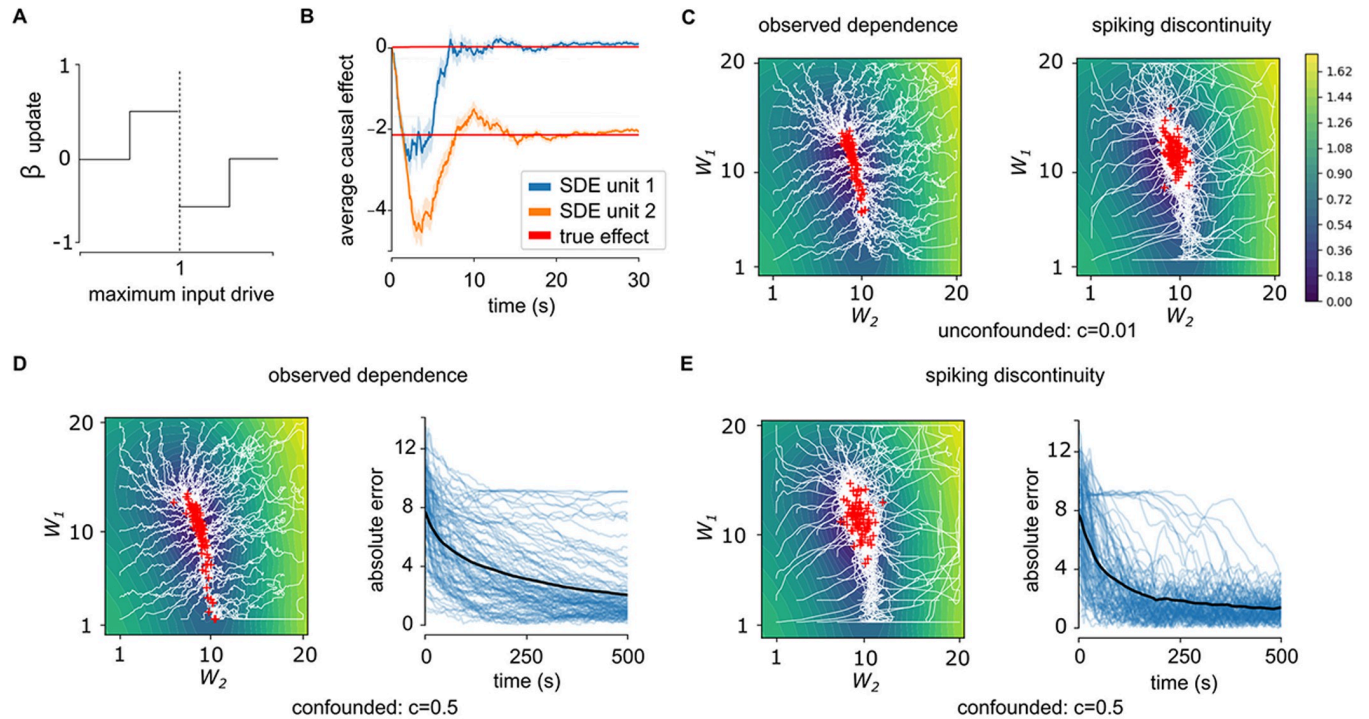
in order to maximize reward. This allows us to use the causal effect to estimate  $\frac{\partial R}{\partial w_i}$  ([Fig 3E and 3F](#)), and thus gives a local learning rule that approximates gradient-descent.

When applied to the toy LIF network, the online learning rule ([Fig 4A](#)) estimates  $\beta$  over the course of seconds ([Fig 4B](#)). The estimated  $\beta$  is then used to update weights to maximize expected reward in an unconfounded network (uncorrelated noise— $c = 0.01$ ). In these simulations updates to  $\beta$  are made when the neuron is close to threshold, while updates to  $w_i$  are made for all time periods of length  $T$ . Learning exhibits trajectories that initially meander while the estimate of  $\beta$  settles down ([Fig 4C](#)). When a confounded network (correlated noise— $c = 0.5$ ) is used the spike discontinuity learning exhibits similar performance, while learning based on the observed dependence sometimes fails to converge due to the bias in gradient estimate. In this case convergence is faster than learning based on observed dependence ([Fig 4D and 4E](#)). Thus the spike discontinuity learning rule allows a network to be trained even in the presence of confounded inputs.

## The impact of network depth and width on causal effect estimation

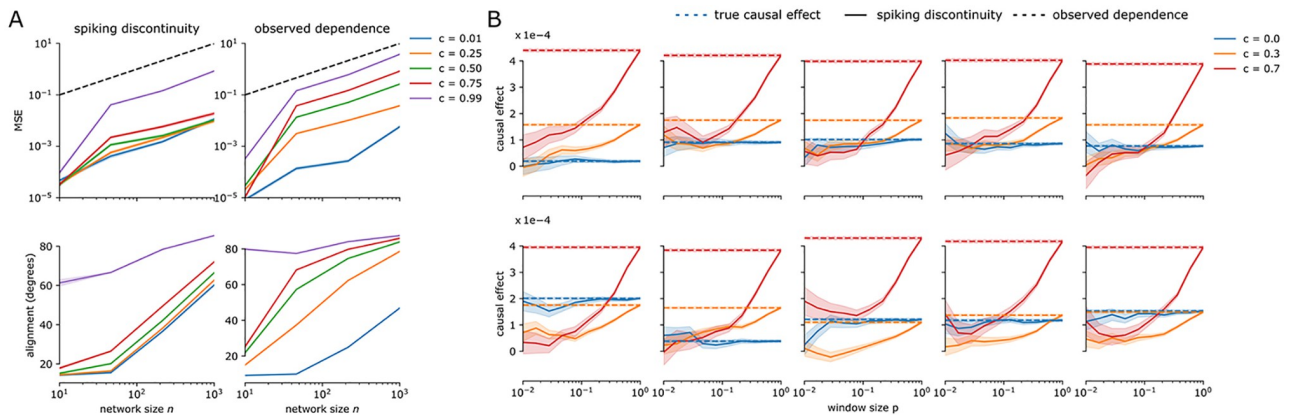
Having validated spiking discontinuity-based causal inference in a small network, we investigate how well we can estimate causal effects in wider and deeper networks. First we investigate the effects of network width on performance. We simulate a single hidden layer neural network of varying width ([Fig 5A](#); refer to the Methods for implementation details). The mean squared error in estimating causal effects shows an approximately linear dependence on the number of neurons in the layer, for both the observed-dependence estimator and the spiking discontinuity. For low correlation coefficients, representing low confounding, the observed dependence estimator has a lower error. However, once confounding is introduced, the error increases dramatically, varying over three orders of magnitude as a function of correlation coefficient. In contrast, the spiking discontinuity error is more or less constant as a function of correlation coefficient, except for the most extreme case of  $c = 0.99$ . This shows that over a range of network sizes and confounding levels, a spiking discontinuity estimator is robust to confounding.

Furthermore, when considering a networks' estimates as a whole, we can compare the vector of estimated causal effects  $\hat{\beta}^{SDE,0}$  to the true causal effects  $\beta$  ([Fig 5A](#), bottom panels). The angle between these two vectors gives an idea of how a learning algorithm will perform when



**Fig 4. Maximizing reward with a spike-discontinuity learning rule.** (A) Parameters for causal effect model,  $\mathbf{u}$ , are updated based on whether neuron is driven marginally below or above threshold. (B) Applying rule to estimate  $\beta_i^{SDE,0}$  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 spike discontinuity (right) learning rule to unconfounded network ( $c = 0.01$ ). Observed dependence converges more directly to bottom of valley, while spiking discontinuity learning trajectories meander more, as the initial estimate of causal effect takes more inputs to update. (D,E) Convergence of observed dependence (D) and spiking discontinuity (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 spike discontinuity learning succeeds.

<https://doi.org/10.1371/journal.pcbi.1011005.g004>



**Fig 5. Effect of network architecture on spiking discontinuity.** (A) Mean square error (MSE) as a function of network size and noise correlation coefficient,  $c$ . MSE is computed as squared difference from the true causal effect, where the true causal effect is estimated using the observed dependence estimator with  $c = 0$  (unconfounded). Alignment between the true causal effects  $\beta$  and the estimated effects  $\hat{\beta}^{SDE,0}$  is the angle between these two vectors. (B) A two hidden layer neural network, with hidden layers of width 10. Plots show the causal effect of each of the first hidden layer neurons on the reward signal. Dashed lines show the observed-dependence estimator, solid lines show the spiking discontinuity estimator, for correlated and uncorrelated (unconfounded) inputs, over a range of window sizes  $p$ . The observed dependence estimator is significantly biased with confounded inputs.

<https://doi.org/10.1371/journal.pcbi.1011005.g005>

using these estimates of the causal effect. If considered as a gradient then any angle well below ninety represents a descent direction in the reward landscape, and thus shifting parameters in this direction will lead to improvements. In this case there is a more striking difference between the spiking discontinuity and observed dependence estimators. Only for extremely high correlation values or networks with a layer of one thousand neurons does it fail to produce estimates that are not very well aligned with the true causal effects. In contrast, the observed dependence estimator is only well-aligned with the true gradient for small networks, and with a small correlation coefficient. This suggests the SDE provides a more scale-able and robust estimator of causal effect for the purposes of learning in the presence of confounders.

We also want to know whether spiking discontinuity can estimate causal effects in deep neural networks. It effectively estimates the causal effect in a spiking neural network with two hidden layers (Fig 5B). We compare a network simulated with correlated inputs, and one with uncorrelated inputs. The estimates of causal effect in the uncorrelated case, obtained using the observed dependence estimator, provide an unbiased estimator the true causal effect (blue dashed line). The causal effect in the correlated inputs case is indeed close to this unbiased value. In contrast, using the observed-dependence estimator on the confounded inputs significantly deviates from the true causal effect. These results show spiking discontinuity can estimate causal effects in both wide and deep neural networks.

## Discussion

Here we focused on the relation between gradient-based learning and causal inference. Our approach is inspired by the regression discontinuity design commonly used in econometrics [34]. We 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 effect in an unbiased way.

It is important to note that other neural learning rules also perform causal inference. Thus the spiking discontinuity 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) [27, 35–37]. 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 is 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 [25, 38]. Many of these methods use something like the REINFORCE algorithm [39], 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 [16–22].

Learning in birdsong is a particularly well developed example of this form of learning [17]. 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 broadly how a neuron may estimate what is perturbative noise without these structural specifics, and thus if it can provide an account of learning in general.

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 [11–15]. 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 can do this. Though well characterized in sensory coding, noise correlation 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.

Learning with perturbations scales poorly with network size [17, 22, 40]. Thus neurons may use alternatives to 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 [41]. 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 efficient 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.

An important disclaimer is that the performance of local update rules like SDE-based learning are likely to share similar scaling to that observed by REINFORCE-based methods, e.g. significantly slower than backpropagation. SDE-based learning, on its own, is not a learning rule that is significantly more efficient than REINFORCE, instead it is a rule that is more robust to the structure of noise that REINFORCE-based methods utilize. Thus it is an approach that can be used in more neural circuits than just those with special circuitry for independent noise perturbations. Such an approach can be built into neural architectures alongside backpropagation-like learning mechanisms, to solve the credit assignment problem. Indeed, this exact approach is taken by [43]. Thus SDE-based learning has relevance to both spiking neural networks in the presence of noise correlations, and as part of a biologically plausible solution to the credit assignment problem.

Further, the insights made here are relevant to models that attempt to mimic backpropagation through time. A lot of recurrent neural networks, when applied to spiking neural networks, have to deal with propagating gradients through the discontinuous spiking function [5, 44–48]. A common strategy is to replace the true derivative of the spiking response function (either zero or undefined), with a pseudo-derivative. A number of replacements have been explored: [44] uses a boxcar function, [47] uses the negative slope of the sigmoid function, [45] explores the so-called straight-through estimator—pretending the spike response function is the identity function, with a gradient of 1. Thus a number of choices are possible. As a supplementary analysis (S1 Text and S3 Fig), we demonstrated that the width of the non-zero component of this pseudo-derivative can be adjusted to account for correlated inputs. We found that in highly correlated cases, learning is more efficient when the window is smaller. Thus the exact same considerations raised by framing learning as a causal inference problem provides insight into other biologically-plausible, spiking learning models.

Finally, it must be noted how the learning rule derived here relates to the dominant spike-based learning paradigm—spike timing dependent plasticity (STDP [49]). 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 [50, 51]. 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.

### Compatibility with known physiology

There are a number of ways that the simulations presented here are simplifications of true learning circuits. For instance, the LIF neural network implemented in Figs 2–4 has a fixed threshold. Yet cortical neurons do not have a fixed threshold, but rather one that can adapt to recent inputs [52]. We note that our exploration of learning in this more complicated case—the delayed XOR model (S1 Text)—consists of populations of LIF and adaptive LIF neurons. The adaptive LIF neurons do have a threshold that adapts, based on recent spiking activity. The results in this model exhibit the same behavior as that observed in previous sections—for sufficiently highly correlated activity, performance is better for a narrow spiking discontinuity parameter  $p$  (cf. Fig 2A). This suggests populations of adaptive spiking threshold neurons show the same behavior as non-adaptive ones. This makes sense since what is important for the spike discontinuity method is that the learning rule use a signal that is unique to an individual neuron. The stochastic, all-or-none spiking response provides this, regardless of the exact value of the threshold. The neuron just needs to know if it was close to its threshold or not. Of course, given our simulations are based on a simplified model, it makes sense to ask what neuro-physiological features may allow spiking discontinuity learning in more realistic learning circuits.

If neurons perform something like spiking discontinuity learning we should expect that they exhibit certain physiological properties. In this section we discuss the concrete demands of such learning and how they relate to past experiments. First, the spiking regime is required to be irregular [26] in order to produce spike trains with randomness in their response for repeated inputs. Irregular spiking regimes are common in cortical networks (e.g. [53]). Further, for spiking discontinuity learning, plasticity should be confined to cases where 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 [54–56]. This also means that plasticity will not occur for inputs that place a neuron too far below threshold. In fact, in past models and experiments testing voltage-dependent plasticity, changes do not occur when postsynaptic voltages are too low [57, 58]. Spiking discontinuity 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. To test if this is an important difference between what is statistically correct and what is more readily implementable in neurophysiology, we experimented with a modification of the learning rule, which does *not* distinguish between barely above threshold inputs and well above threshold inputs. For the piecewise-linear model, this modification did not adversely affect the neurons' ability to estimate causal effects (S2 Fig). Thus threshold-adjacent plasticity as required for spike discontinuity learning appears to be compatible with neuronal physiology.

The learning rules presented here also need knowledge of outcomes and would thus likely be dependent on neuromodulation. Neuromodulated-STDP is well studied in models [51, 59, 60]. However, this learning requires reward-dependent plasticity that differs depending on if the neuron spiked or not. This may be communicated by neuromodulation. For instance,

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 [59]. To the best of our knowledge, how such behavior interacts with postsynaptic voltage dependence as required by spike discontinuity is unknown.

Overall, the structure of the learning rule is that a global reward signal (potentially transmitted through neuromodulators) drives learning of a variable,  $\beta$ , inside single neurons. This internal variable is combined with a term  $\frac{d}{dw} \mathbb{E}(H)$  to update synaptic weights. The update rule for the weights depends only on pre- and post-synaptic terms, with the post- term getting updated over time, independently of the weight updates. Such an interpretation is interestingly in line with recently proposed ideas on inter-neuron learning, e.g., Gershman 2023 [61], who proposes an interaction of intra-cellular variables and synaptic learning rules can provide a substrate for memory. Thus, taken together, these factors show that SDE-based learning may well be compatible with known neuronal physiology.

## Conclusion

Here we presented the first exploration of the idea that neurons can perform causal inference using their spiking mechanism. There are a number of avenues for future work to develop the idea further. In particular, we primarily presented empirical results demonstrating the idea in numerous settings. Further experiments with the impact of learning window size in other learning rules where a pseudo-derivative type approach to gradient-based learning is applied can be performed, to establish the broader relevance of the insights made here. And the theoretical results that we presented were made under the strong assumption that the neural network activity, when appropriately aggregated, can be described by a causal Bayesian network. More rigorous results are needed. The problem of coarse-graining, or aggregating, low-level variables to obtain a higher-level model amenable to causal modeling is a topic of active research [62, 63]. Further fleshing out an explicit theory that relates neural network activity to a formal causal model is an important future direction.

In this paper SDE based learning was explored in the context of maximizing a reward function or minimizing a loss function. But the same mechanism can be exploited to learn other signals—for instance, surprise (e.g. [64]) or novelty (e.g. [65]). SDE-based learning is a mechanism that a spiking network can use in many learning scenarios.

The most important aspect of spike discontinuity 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 [37, 66]. 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 simulations and noise

We consider the activity of a simulated network of  $n$  neurons whose activity is described by their spike times,  $\{t_s^i\}_{s=1}^{M_i}$ :

$$h_i(t) = \sum_{s=1}^{M_i} \delta(t - t_s^i).$$

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

$$\dot{v}_i = -g_L v_i + w_i \eta_i, \tag{8}$$

where integrate and fire means simply:

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

A refractory period of 3ms is added to the neurons. Different choices of refractory period were not shown to affect SDE performance (S1 Fig). Noisy input  $\eta_i$  is comprised of a common DC current,  $x$ , and noise term,  $\xi(t)$ , plus an individual noise term,  $\xi_i(t)$ :

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

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  $c$ .  $\sigma = 10$  was used for the noise magnitude unless otherwise stated. Simulations are performed with a step size of  $\Delta t = 1\text{ms}$ . Here the reset potential was set to  $v_r = 0$ . The firing rate of a noisy integrate and fire neuron is

$$\mu_i = \left[ \frac{1}{g_L} \int_0^\infty \frac{1}{u} (\exp(-u^2 + 2y_i^{th}u) - \exp(-u^2 + 2y_i^r u)) du \right]^{-1}, \tag{9}$$

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 [21]. This is used in the learning rule derived below.

We define the input drive to the neuron,  $u_i$ , 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  $T = 50\text{ms}$  was used. Reward is administered at the end of this period:  $R = R(\mathbf{s}_T)$ .

In our simulations, the dynamics are standard integrate and fire dynamics,  $\mathbf{s} \in \mathbb{R}^n$  and are given by

$$\tau_s \dot{s}_i = -s_i + h_i(t), \tag{10}$$

for synaptic time scale  $\tau_s = 0.02\text{s}$ . We thus use a standard model for the dynamics of all the neurons.

### Reward model and causal effect estimation

The simulations for Figs 3 and 4 are about standard supervised learning and there 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 following form:

$$R(s_1, s_2) = (as_1 + bs_2 + bs_2^2 + x)^2$$

We choose  $a = -30$ ,  $b = 20$ ,  $x = -4$ .

The true causal effect was estimated from simulations with zero noise correlation and with a large window size—in order to produce the most accurate estimate possible. Determining the causal effect analytically is in general intractable.

### Learning causal effects through the spiking discontinuity

To remove confounding, spiking discontinuity learning considers only the marginal super- and sub-threshold periods of time to estimate  $\beta$ . 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 estimates [28]:

$$\beta_i^{SDE} := \lim_{x \rightarrow \theta^+} \mathbb{E}(R|Z_i = x) - \lim_{x \rightarrow \theta^-} \mathbb{E}(R|Z_i = x),$$

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

To estimate  $\beta_i^{SDE}$ , a neuron can estimate a piece-wise linear model of the reward function:

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

locally, when  $Z_i$  is within a small window  $p$  of threshold. Here  $\gamma_i$ ,  $\alpha_{li}$  and  $\alpha_{ri}$  are nuisance parameters, and  $\beta_i$  is the causal effect of interest. This means we can estimate  $\beta_i^{SDE}$  from

$$\beta_i \approx \mathcal{R}(1, \theta) - \mathcal{R}(0, \theta).$$

A neuron can learn an estimate of  $\beta_i^{SDE}$  through a least squares minimization on the model parameters  $\beta_i$ ,  $\alpha_{li}$ ,  $\alpha_{ri}$ . That is, for time period  $n$  (of length  $T$ ), if we let  $\mathbf{u}_i = [\gamma_i, \beta_i, \alpha_{ri}, \alpha_{li}]^T$  and  $\mathbf{a}_{i,n} = [1, H_{i,n}, H_{i,n}(Z_{i,n} - \theta), (1 - H_{i,n})(Z_{i,n} - \theta)]^T$ , then if the neuron solves the following minimization:

$$\hat{\mathbf{u}}_i = \operatorname{argmin}_{\mathbf{u}} \sum_{n: (\theta-p < z_{i,n} < \theta+p)}^N [\mathbf{u}_i^T \mathbf{a}_{i,n} - R_n]^2,$$

then that allows it to estimate its causal effect.

A neuron can perform stochastic gradient descent on this minimization problem, giving the learning rule:

$$\Delta \mathbf{u}_i = -\eta [\mathbf{u}_i^T \mathbf{a}_{i,n} - R_n] \mathbf{a}_{i,n}$$

for learning rate  $\eta$  and for all time periods at which  $z_{i,n}$  is within  $p$  of threshold  $\theta$ .

When estimating the simpler, piece-wise constant model for either side of the threshold, the learning rule simplifies to:

$$\begin{aligned} \Delta \beta_i &= -\eta [\gamma_i + \beta_i - R_n], & \theta \leq z_{i,n} < \theta + p \text{ (neuron } i \text{ just spikes)} \\ \Delta \gamma_i &= -\eta [\gamma_i - R_n]. \end{aligned}$$

again for all time periods at which  $z_{i,n}$  is within  $p$  of threshold  $\theta$ . This causal inference strategy, established by econometrics, is ultimately what allows neurons to produce unbiased estimates of causal effects.

### The causal effect as a finite-difference operator

As a discrete event, we are interested not necessarily in the local gradient but in the finite difference between spiking and non-spiking. Importantly, this finite-difference approximation is exactly what our estimator gets at. We present a derivation here.

In particular, 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:

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

Here  $\mathbf{Q}_{j \neq i} \subset \mathcal{X}$  is a set of nodes that satisfy the back-door criterion [27] with respect to  $H_i \rightarrow R$ . By satisfying the backdoor criterion we can relate the interventional distribution to the observational distribution. When  $R$  is a deterministic, differentiable function of  $\mathbf{S}$  and  $\Delta_s \rightarrow 0$  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  $\Delta_s$  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  $\mathbf{Q}_{j \neq i}$ :

$$\begin{aligned} \beta_i &= \mathbb{E}(R|\text{do}(H_i = 1)) - \mathbb{E}(R|\text{do}(H_i = 0)) \\ &= \mathbb{E}(\mathbb{E}(R|\mathbf{Q}_{j \neq i}, H_i = 1) - \mathbb{E}(R|\mathbf{Q}_{j \neq i}, H_i = 0)) \\ &= \mathbb{E}(\mathbb{E}(\mathbb{E}(R|S_i, \mathbf{Q}_{j \neq i})|\mathbf{Q}_{j \neq i}, H_i = 1) - \mathbb{E}(\mathbb{E}(R|S_i, \mathbf{Q}_{j \neq i})|\mathbf{Q}_{j \neq i}, H_i = 0)). \end{aligned} \tag{11}$$

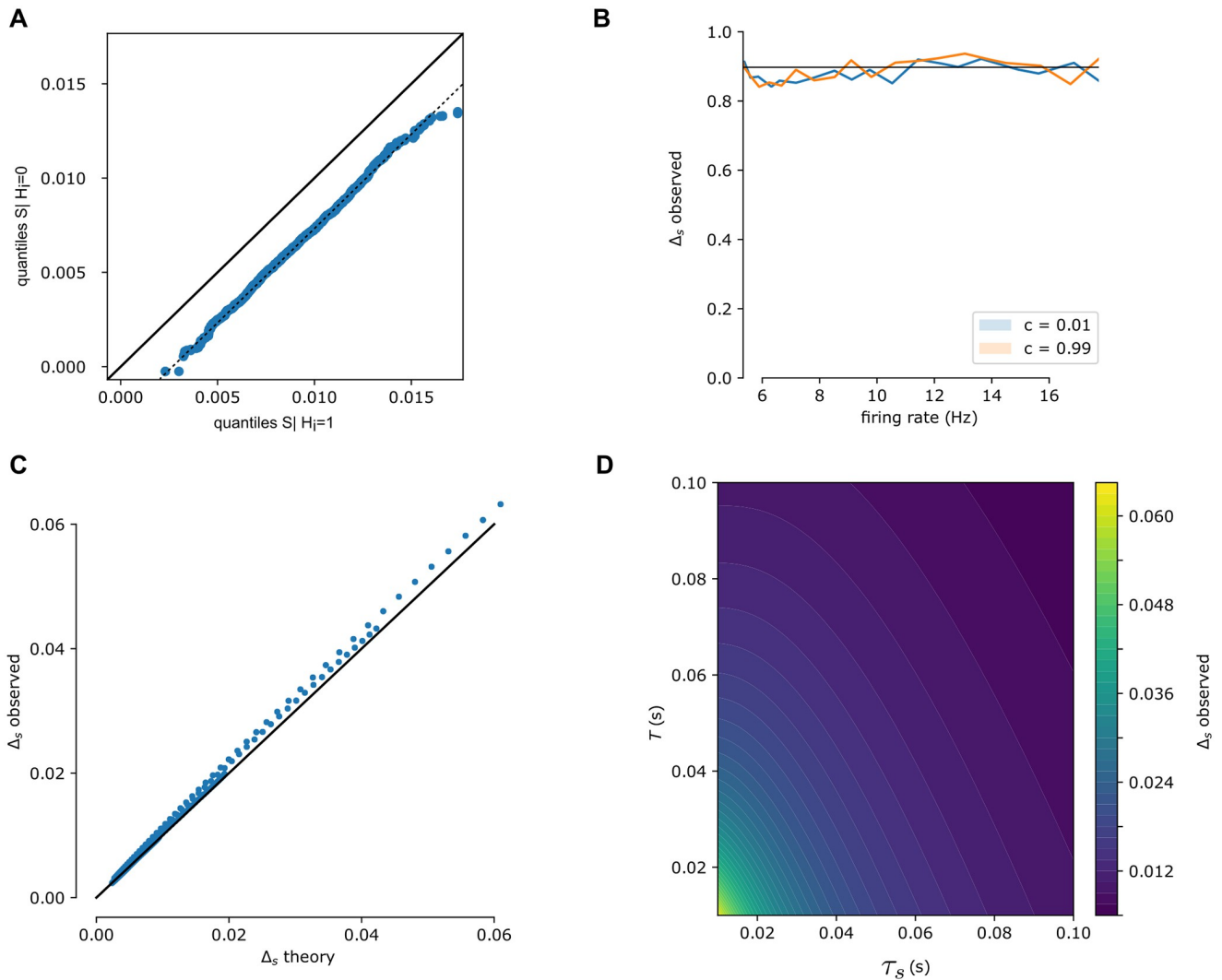
Now if we assume that on average  $H_i$  spiking induces a change of  $\Delta_s$  in  $S_i$  within the same time period, compared with not spiking, then:

$$\rho(s_i|\mathbf{Q}_{j \neq i}, H_i = 1) \approx \rho(s_i - \Delta_s|\mathbf{Q}_{j \neq i}, H_i = 0). \tag{12}$$

Note that  $S_i$  given no spike (the distribution on the RHS) is not necessarily zero, even if no spike occurs in that time period, because  $s_i(0)$ , the filtered spiking dynamics at  $t = 0$ , is not necessarily equal to zero.  $s_i(0)$  instead has its own, non-zero distribution, representing activity from previous time periods. This approximation is reasonable because the linearity of the synaptic dynamics means that the difference in  $S_i$  between spiking and non-spiking windows is simply  $\exp(-(T - t_{si})/\tau_s)/\tau_s$ , for spike time  $t_{si}$ . We approximate this term with its mean:

$$\begin{aligned} \Delta_s &= \mathbb{E}\left(\frac{1}{\tau_s} e^{-t_{si}/\tau_s} | \mathbf{Q}_{j \neq i}, H_i = 1\right) \\ &\approx \frac{1}{T} (1 - e^{-T/\tau_s}), \end{aligned} \tag{13}$$

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



**Fig 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 (12). (B) This offset,  $\Delta_s$ , 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  $\Delta_s$  (Eq (13)) is compared to simulated  $\Delta_s$ . Proximity to the diagonal line (black curve) shows these match. (D)  $\Delta_s$  as a function of window size  $T$  and synaptic time constant  $\tau_s$ . Larger time windows and longer time constants lower the change in  $S_i$  due to a single spike.

<https://doi.org/10.1371/journal.pcbi.1011005.g006>

Writing out the inner two expectations of (11) gives:

$$\begin{aligned}
 & \mathbb{E}(\mathbb{E}(R|S_i, \mathbf{Q}_{j \neq i}) | \mathbf{Q}_{j \neq i}, H_i = 1) - \mathbb{E}(\mathbb{E}(R|S_i, \mathbf{Q}_{j \neq i}) | \mathbf{Q}_{j \neq i}, H_i = 0) \\
 &= \int_0^\infty \mathbb{E}(R | \mathbf{Q}_{j \neq i}, S_i = s_i) [\rho(s_i | \mathbf{Q}_{j \neq i}, H_i = 1) - \rho(s_i | \mathbf{Q}_{j \neq i}, H_i = 0)] ds_i \\
 \text{from (12)} \quad &= \int_0^\infty \mathbb{E}(R | S_i = s_i + \Delta_s, \mathbf{Q}_{j \neq i}) \rho(s_i | \mathbf{Q}_{j \neq i}, H_i = 0) \\
 & \quad - \mathbb{E}(R | S_i = s_i, \mathbf{Q}_{j \neq i}) \rho(s_i | \mathbf{Q}_{j \neq i}, H_i = 0) ds_i,
 \end{aligned}$$

after making the substitution  $s_i \rightarrow s_i + \Delta_s$  in the first term. Writing this back in terms of

expectations gives:

$$\begin{aligned} \beta &\approx \mathbb{E}(\mathbb{E}(\mathbb{E}(R|S_i + \Delta_s, \mathbf{Q}_{j \neq i}) - \mathbb{E}(R|S_i, \mathbf{Q}_{j \neq i}) | \mathbf{Q}_{j \neq i}, H_i = 0)) \\ &= \Delta_s \mathbb{E}(\mathbb{E}(D_i R(S_i, \mathbf{Q}_{j \neq i}) | \mathbf{Q}_{j \neq i}, H_i = 0)) \\ &= \Delta_s \mathbb{E}(D_i R(S_i, \mathbf{Q}_{j \neq i}) | \text{do}(H_i = 0)). \end{aligned}$$

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

### Wider and deeper network implementations

We need to give more details for the wide network (Fig 5A) simulations. We use a single hidden layer of  $N$  neurons each receives inputs  $\eta$ . This is a mix of some fixed, deterministic signal and a noise signal,  $\xi$ :

$$\eta_i(t) = x + \sqrt{1 - c} \xi_i(t) + \sqrt{c} \xi(t).$$

As in the  $N = 2$  cases above, this is chosen so that the noise between any pair of neurons is related with a correlation coefficient  $c$ . This is then weighted by the vector  $\mathbf{w}$ , which drives the leaky integrate and fire neurons to spike. The vector  $\mathbf{w}$  was set to a vector of all ones. The output of the single hidden layer  $\mathbf{s}$  is weighed by a second vector  $\mathbf{u}$ . From this a cost/reward function of  $R = |U \cdot \mathbf{s} - Y|$  is computed. The vector  $U$  is randomly generated from a normal distribution  $\mathcal{N}(0, 5)$  such that each neuron has a different causal effect on the output. The target  $Y$  is set to  $Y = 0.1$ . From this setup, each neuron can estimate its effect on the function  $R$  using either the spiking discontinuity learning or the observed dependence estimator. Varying  $N$  allows us to study how the method scales with network size. Ten simulations were run for each value of  $N$  and  $c$  in Fig 5A.

For the deep network (Fig 5B), a two-hidden layer neural network was simulated. Each layer had  $N = 10$  neurons. The inputs to the first layer are as above:

$$\eta_i^1(t) = x + \sqrt{1 - c} \xi_i^1(t) + \sqrt{c} \xi^1(t).$$

thus neural activity in this first layer is pairwise correlated with coefficient  $c$ . The second layer receives input from the first:

$$\eta_i^2(t) = V_i \cdot \mathbf{s}(t) + \xi_i^2(t)$$

thus noise in this layer is not correlated and any correlations between neural activity in the second layer come from correlated activity in the first layer. The matrix  $V$  weighs inputs from the first to second layer. It is generated from a normal distribution:  $\mathcal{N}(100, 200^2)$ . The same reward function is used as in the wide network simulations above, except here  $U$  is a vector of ones. The target is set to  $t = 0.02$ . To test the ability of spiking discontinuity learning to estimate causal effects in deep networks, the learning rule is used to estimate causal effects of neurons in the first layer on the reward function  $R$ .

### Supporting information

**S1 Text. Delayed XOR task description, methods and results.**  
(PDF)

**S1 Fig. Error in causal effect estimation for LIF networks with different refractory periods.**  
As in Fig 3D, histograms plot error in causal effect over a range of network weights. LIF

neurons have refractory period of 1,3 or 5 ms. Error is comparable for different refractory periods.

(TIF)

**S2 Fig. One-sided discontinuity estimator.** The one-sided estimator makes updates for inputs placing the neuron  $p$  below threshold, and for *any* input that places the neuron above threshold. Statistically, the symmetric choice is the most sensible default. Regression discontinuity design, the related method in econometrics, has studied optimizing the underlying kernel, which may not be symmetric depending on the relevant distributions. To address this question, we ran extra simulations in which the window size is asymmetric. We tested, in particular, the case where  $p$  is some small value on the left of the threshold (sub-threshold inputs), and where  $p$  is large to the right of the threshold (above-threshold inputs). This deviates from the statistically correct choice, but in a way, this is a biologically plausible setting. This is because, with such a setup, the neuron does not need to distinguish between barely-above-threshold inputs and well-above-threshold inputs, which may be challenging. Instead, any spiking will result in an update to the estimate of the causal effect. We found that the asymmetric estimator performs worse when using the piecewise constant estimator of causal effect, but performs comparably to the symmetric version with using the piecewise linear estimator. Thus spiking discontinuity learning can operate using asymmetric update rules.

(TIF)

**S3 Fig. Learning delayed XOR in the presence of correlated noise.** (A) Delayed XOR task setup, shown after training. Populations of input neurons sequentially encode binary inputs  $(x_1, x_2)$ , and after a delay a population of neurons cues a response. Input value of 0 is indicated by the red population being active, and 1 is indicated by the blue population being active. Correlated Gaussian noise, with correlation coefficient  $c$ , is added to the neurons membrane potential. Softmax output above 0.5 indicates a network output of 1, and below 0.5 indicates 0. A sample raster of 20 neurons is shown here. (B) Mean validation error over 10 repeated training runs, for a range of correlation coefficients,  $c$ , and learning window sizes,  $p$ . (C) Mean number of iterations taken to reach training error below a stopping threshold of 0.07.

(TIF)

## Acknowledgments

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

## Author Contributions

**Conceptualization:** Benjamin James Lansdell, Konrad Paul Kording.

**Formal analysis:** Benjamin James Lansdell, Konrad Paul Kording.

**Funding acquisition:** Konrad Paul Kording.

**Investigation:** Konrad Paul Kording.

**Methodology:** Benjamin James Lansdell.

**Software:** Benjamin James Lansdell.

**Supervision:** Konrad Paul Kording.

**Validation:** Benjamin James Lansdell.

**Visualization:** Benjamin James Lansdell.

**Writing – original draft:** Benjamin James Lansdell, Konrad Paul Kording.

**Writing – review & editing:** Benjamin James Lansdell, Konrad Paul Kording.

## References

1. Pfeiffer M, Pfeil T. Deep Learning With Spiking Neurons : Opportunities and Challenges. *Frontiers in Neuroscience*. 2018; 12(October). <https://doi.org/10.3389/fnins.2018.00774> PMID: 30410432
2. Tavanaei A, Ghodrati M, Reza S. Deep learning in spiking neural networks. *Neural Networks*. 2019; 111:47–63. <https://doi.org/10.1016/j.neunet.2018.12.002> PMID: 30682710
3. Bellec G, Salaj D, Subramoney A, Legenstein R, Maass W. Long short-term memory and Learning-to-learn in networks of spiking neurons. *ArXiv e-prints*. 2018; p. 1–17.
4. Bellec G, Scherr F, Hajek E, Salaj D, Legenstein R, Maass W. Biologically inspired alternatives to back-propagation through time for learning in recurrent neural nets. *arXiv preprint*. 2019; p. 1–34.
5. Neftci EO, Mostafa H, Zenke F. Surrogate Gradient Learning in Spiking Neural Networks. *IEEE SPM*. 2019; p. 1–21.
6. Alawad M, Yoon HJ, Tourassi G. Energy Efficient Stochastic-Based Deep Spiking Neural Networks for Sparse Datasets. *IEEE Conference on Big Data*. 2017; p. 311–318.
7. Tang G, Shah A, Michmizos KP. Spiking Neural Network on Neuromorphic Hardware for Energy-Efficient Unidimensional SLAM. *ArXiv e-prints*. 2019;.
8. Purves D, Augustine GJ, Fitzpatrick D, Hall WC, LaMantia AS, McNamara JO, et al. *Neuroscience*. 2008. De Boeck, Sinauer, Sunderland, Mass. 2014;.
9. Theunissen F, Miller JP. Temporal Encoding in Nervous Systems : A Rigorous Definition. *Journal of computational neuroscience*. 1995; 2:149–162. <https://doi.org/10.1007/BF00961885> PMID: 8521284
10. Lansdell B, Prakash P, Kording KP. Learning to solve the credit assignment problem. *ICLR 2020*. 2020; p. 1–12.
11. Baker C, Ebsch C, Lampl I, Rosenbaum R. The correlated state in balanced neuronal networks. *bioRxiv*. 2018;.
12. Cohen MR, Kohn A. Measuring and interpreting neuronal correlations. *Nature Neuroscience*. 2011; 14(7):811–819. <https://doi.org/10.1038/nn.2842> PMID: 21709677
13. Kanitscheider I, Coen-cagli R, Pouget A. Origin of information-limiting noise correlations. *Proceedings of the National Academy of Sciences*. 2015; . <https://doi.org/10.1073/pnas.1508738112> PMID: 26621747
14. Bejjanki VR, da Silveira RA, Cohen JD, Turk-Browne NB. Noise correlations in the human brain and their impact on pattern classification. *PLoS computational biology*. 2017; 13(8):e1005674. <https://doi.org/10.1371/journal.pcbi.1005674> PMID: 28841641
15. Yim MY, Aertsen A, Kumar A. Significance of Input Correlations in Striatal Function. *PLoS Comput Biol*. 2011; 7(11). <https://doi.org/10.1371/journal.pcbi.1002254> PMID: 22125480
16. Bouvier G, Clopath C, Bimbard C, Nadal JP, Brunel N, Hakim V, et al. Cerebellar learning using perturbations. *bioRxiv*. 2016; p. 053785.
17. Fiete IR, Fee MS, Seung HS. Model of Birdsong Learning Based on Gradient Estimation by Dynamic Perturbation of Neural Conductances. *Journal of neurophysiology*. 2007; 98:2038–2057. <https://doi.org/10.1152/jn.01311.2006> PMID: 17652414
18. Fiete IR, Seung HS. Gradient learning in spiking neural networks by dynamic perturbation of conductances. *Physical Review Letters*. 2006; 97. <https://doi.org/10.1103/PhysRevLett.97.048104> PMID: 16907616
19. Legenstein R, Chase SM, Schwartz AB, Maas W, Maass W. A Reward-Modulated Hebbian Learning Rule Can Explain Experimentally Observed Network Reorganization in a Brain Control Task. *Journal of Neuroscience*. 2010; 30(25):8400–8410. <https://doi.org/10.1523/JNEUROSCI.4284-09.2010> PMID: 20573887
20. Miconi T. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *eLife*. 2017; 6:1–24. <https://doi.org/10.7554/eLife.20899> PMID: 28230528
21. Xie X, Seung HS. Learning in neural networks by reinforcement of irregular spiking. *Physical Review E*. 2004; 69. <https://doi.org/10.1103/PhysRevE.69.041909> PMID: 15169045

22. Seung S. Learning in Spiking Neural Networks by Reinforcement of Stochastic Transmission. *Neuron*. 2003; 40:1063–1073.
23. Hoerzer GM, Legenstein R, Maass W. Emergence of complex computational structures from chaotic neural networks through reward-modulated hebbian learning. *Cerebral Cortex*. 2014; 24(3):677–690. <https://doi.org/10.1093/cercor/bhs348> PMID: 23146969
24. Guerguiev J, Lillicrap TP, Richards BA. Deep learning with segregated dendrites. arXiv preprint. 2016;1610(00161):1–29.
25. Schultz W. Getting formal with dopamine and reward. *Neuron*. 2002; 36(2):241–263. [https://doi.org/10.1016/S0896-6273\(02\)00967-4](https://doi.org/10.1016/S0896-6273(02)00967-4) PMID: 12383780
26. Brunel N. Dynamics of sparsely connected networks of excitatory and inhibitory neurons. *Computational Neuroscience*. 2000; 8:183–208. <https://doi.org/10.1023/A:1008925309027>
27. Pearl J. Causality: models, reasoning and inference. Cambridge Univ Press; 2000.
28. Imbens GW, Lemieux T. Regression discontinuity designs: A guide to practice. *Journal of Econometrics*. 2008; 142(2):615–635. <https://doi.org/10.1016/j.jeconom.2007.05.001>
29. Angrist J, Pischke JS. The Credibility Revolution in Empirical Economics: How Better Research Design is Taking the Con Out of Economics. *Journal of Economic Perspectives*. 2010; 24(2). <https://doi.org/10.1257/jep.24.2.3>
30. Jacob R, Zhu P, Somers Ma, Bloom H. A Practical Guide to Regression Discontinuity. MDRC. 2012; (July).
31. Shea-Brown E, Josić K, De La Rocha J, Doiron B. Correlation and synchrony transfer in integrate-and-fire neurons: Basic properties and consequences for coding. *Physical Review Letters*. 2008; 100(10):1–4. <https://doi.org/10.1103/PhysRevLett.100.108102> PMID: 18352234
32. Peters J, Janzing D, Schölkopf B. Elements of Causal Inference: Foundations and Learning Algorithms. Cambridge, MA, USA: MIT Press; 2017.
33. Imbens G, Kalyanaraman K. Optimal bandwidth choice for the regression discontinuity estimator. *Review of Economic Studies*. 2012; 79(3):933–959. <https://doi.org/10.1093/restud/rdr043>
34. Angrist JD, Pischke JS. Mostly Harmless Econometrics : An Empiricist' s Companion. Princeton University Press; 2009.
35. Woodward J. Interventionist theories of causation in psychological perspective. In: Causal learning: psychology, philosophy and computation. New York: Oxford university press; 2007.
36. Hagmayer Y, Fernbach P. Causality in Decision-Making. vol. 1; 2017. Available from: <http://oxfordhandbooks.com/view/10.1093/oxfordhb/9780199399550.001.0001/oxfordhb-9780199399550-e-27>.
37. Gershman SJ. Reinforcement learning and causal models. In: Oxford Handbook of Causal Reasoning. Oxford university press; 2017. p. 1–32.
38. Wise RaR. Dopamine, learning and motivation. *Nature reviews Neuroscience*. 2004; 5(6):483–494. <https://doi.org/10.1038/nrn1406> PMID: 15152198
39. Williams R. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*. 1992; 8:299–256. <https://doi.org/10.1007/BF00992696>
40. Rezende DJ, Mohamed S, Wierstra D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. Proceedings of the 31st International Conference on Machine Learning, PMLR. 2014;32(2):1278–1286.
41. Shafi M, Zhou Y, Quintana J, Chow C, Fuster J, Bodner M. Variability in neuronal activity in primate cortex during working memory tasks. *Neuroscience*. 2007; 146(3):1082–1108. <https://doi.org/10.1016/j.neuroscience.2006.12.072> PMID: 17418956
42. LeCun Y, Bengio Y, Hinton G. Deep Learning. *Nature*. 2015; 521. <https://doi.org/10.1038/nature14539> PMID: 26017442
43. Guerguiev J, Kording KP, Richards BA. Spike-based causal inference for weight alignment. ICLR 2020. 2020; p. 1–12.
44. Neftci EO, Augustine C, Paul S, Detorakis G. Event-driven random back-propagation: Enabling neuro-morphic deep learning machines. *Frontiers in Neuroscience*. 2017; 11(JUN):1–18. <https://doi.org/10.3389/fnins.2017.00324> PMID: 28680387
45. Bengio Y, Léonard N, Courville AC. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. CoRR. 2013;abs/1308.3432.
46. Shrestha SB, Orchard G. SLAYER: Spike Layer Error Reassignment in Time. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. NIPS'18. Red Hook, NY, USA: Curran Associates Inc.; 2018. p. 1419–1428.

47. Zenke F, Ganguli S. SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks. *Neural Computation*. 2017; 30.
48. Bellec G, Scherr F, Subramoney A, Hajek E, Salaj D, Legenstein R, et al. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*. 2020; 11(1):1–15. <https://doi.org/10.1038/s41467-020-17236-y>
49. Bi Gq, Poo Mm. Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. *The Journal of Neuroscience*. 1998; 18(24):10464–10472. <https://doi.org/10.1523/JNEUROSCI.18-24-10464.1998> PMID: 9852584
50. Loewenstein Y, Seung HS. 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*. 2006; 103(41):15224–15229. <https://doi.org/10.1073/pnas.0505220103> PMID: 17008410
51. Frémaux N, Gerstner W. Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules. *Frontiers in Neural Circuits*. 2016; 9(January). <https://doi.org/10.3389/fncir.2015.00085> PMID: 26834568
52. Azouz R, Gray C. Dynamic spike threshold reveals a mechanism for synaptic coincidence detection in cortical neurons in vivo. *Proceedings of the National Academy of Sciences of the United States of America*. 2000; 97:8110–5. <https://doi.org/10.1073/pnas.130200797> PMID: 10859358
53. Bellay T, Klaus A, Seshadri S, Plenz D. Irregular spiking of pyramidal neurons organizes as scale-invariant neuronal avalanches in the awake state. *eLife*. 2015; 4:e07224. <https://doi.org/10.7554/eLife.07224> PMID: 26151674
54. Fino E, Deniau JM, Venance L. Brief subthreshold events can act as Hebbian signals for long-term plasticity. *PLoS ONE*. 2009; 4(8). <https://doi.org/10.1371/journal.pone.0006557> PMID: 19675683
55. Fino E, Venance L. Spike-timing dependent plasticity in the striatum. *Frontiers in synaptic neuroscience*. 2010; 2(June):1–10. <https://doi.org/10.3389/fnsyn.2010.00006> PMID: 21423492
56. Sjöström PJ, Turrigiano GG, Nelson SB. Endocannabinoid-Dependent Neocortical Layer-5 LTD in the Absence of Postsynaptic Spiking. *Journal of Neurophysiology*. 2004; 92(6):3338–3343. <https://doi.org/10.1152/jn.00376.2004> PMID: 15240760
57. Clopath C, Gerstner W. Voltage and spike timing interact in STDP—a unified model. *Frontiers in Synaptic Neuroscience*. 2010; 2(JUL):1–11. <https://doi.org/10.3389/fnsyn.2010.00025> PMID: 21423511
58. Artola A, Brocher S, Singer W. Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature*. 1990; 347(6288):69–72. <https://doi.org/10.1038/347069a0> PMID: 1975639
59. Seol GH, Ziburkus J, Huang S, Song L, Kim IT, Takamiya K, et al. Neuromodulators Control the Polarity of Spike-Timing-Dependent Synaptic Plasticity. *Neuron*. 2007; 55(6):919–929. <https://doi.org/10.1016/j.neuron.2007.08.013> PMID: 17880895
60. Pawlak V, Wickens JR, Kirkwood A, Kerr JND. Timing is not everything: Neuromodulation opens the STDP gate. *Frontiers in Synaptic Neuroscience*. 2010; 2(OCT):1–14.
61. Gershman SJ. The molecular memory code and synaptic plasticity: A synthesis. *Biosystems*. 2023; 224:104825. <https://doi.org/10.1016/j.biosystems.2022.104825> PMID: 36610586
62. Rosas FE, Mediano PAM, Jensen HJ, Seth AK, Barrett AB, Carhart-Harris RL, et al. Reconciling emergences: An information-theoretic approach to identify causal emergence in multivariate data. *PLoS Computational Biology*. 2020; 16(12 December):1–22. <https://doi.org/10.1371/journal.pcbi.1008289> PMID: 33347467
63. Hoel EP, Albantakis L, Tononi G. Quantifying causal emergence shows that macro can beat micro. *Proceedings of the National Academy of Sciences of the United States of America*. 2013; 110(49):19790–19795. <https://doi.org/10.1073/pnas.1314922110> PMID: 24248356
64. Liakoni V, Modirshanechi A, Gerstner W, Brea J. Learning in Volatile Environments With the Bayes Factor Surprise. *Neural Computation*. 2021; 33(2):269–340. [https://doi.org/10.1162/neco\\_a\\_01352](https://doi.org/10.1162/neco_a_01352) PMID: 33400898
65. Xu HA, Modirshanechi A, Lehmann MP, Gerstner W, Herzog MH. Novelty is not surprise: Human exploratory and adaptive behavior in sequential decision-making. *PLOS Computational Biology*. 2021; 17(6):1–32. <https://doi.org/10.1371/journal.pcbi.1009070>
66. Le Pelley ME, Griffiths O, Beesley T. Associative Accounts of Causal Cognition. In: *Oxford Handbook of Causal Reasoning*. vol. 1. Oxford university press; 2017. p. 1–27. Available from: <http://oxfordhandbooks.com/view/10.1093/oxfordhb/9780199399550.001.0001/oxfordhb-9780199399550-e-2>.