# Spiking allows neurons to estimate their causal effect

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

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

**Neural learning aims at the maximization of reward and the typical gradient descent learning is an approximate causal estimator. However real neurons spike based on pre-synaptic drive, creating discontinuities. The regression discontinuity method, popularized by economics, uses such discontinuities to estimate causal effects. Here we show how the spiking discontinuity can thus reveal the influence of a neuron's activity on reward, producing a deep link between simple learning rules and quasi-experimental causal inference.**

Learning is typically conceptualized as changing a neuron's properties to cause better performance or improve the reward $R$. This is a problem of causality: to learn, a neuron needs to estimate its causal influence on reward, $\beta_i$. The typical solution linearizes the problem and leads to popular gradient descent-based (GD) approaches of the form $\beta_i^{GD} = \frac{\partial R}{\partial h_i}$. However gradient descent is just one possible approximation to the estimation of causal influences and one that does not work when gradients are undefined, e.g. in the case of spiking neurons. Focusing on the underlying causality problem promises new ways of understanding learning.

Gradient descent is problematic as a model for biological learning for two reasons. First, real neurons spike, as opposed to units in artificial neural networks (ANNs), and their rates are usually quite slow so that the discreteness of their output matters (e.g. [30]). Further, given physiological constraints on information transmission, it remains unclear how mechanistically neurons might implement gradient descent. Given these challenges we may ask if the brain uses different approaches for causal inference.

The most obvious approach to causality is intervention: if some spiking is random then the correlation of performance with those random perturbations reveal causality. Perturbation-based methods have been demonstrated in various settings [3, 7, 8, 20, 24]. Other approaches rely on intrinsically generated noise to infer causality [35, 29], however these methods fail when the noise utilized for causal inference is correlated among neurons. Yet noise correlations are a common phenomenon in neuroscience (e.g. Zylberberg et al 2016 [36]), which limits these methods' applicability. One can think of injecting noise as equivalent to randomized controlled trials in medicine [22] and A/B tests in computer science [16]. However requiring the injection of extra noise decreases performance, prompting us to ask if it is necessary.

Econometricians have deeply thought about causality [1]. One of the most popular techniques is regression discontinuity design (RDD) [15]. In RDD a binary treatment of interest, $G$, is based on thresholding an input variable, called a forcing or running variable. We are interested in the treatment's effect on an output variable $I$. An example from education might be an exam cutoff for admittance to a selective high school (Fig. 1a) [21]. How can we estimate the causal effect of the high school on future academic performance?

A naive estimate is just to compare the students who attend the selective school to those who attend a less selective school, which we will term the observed dependence (OD):

$$\beta^{OD} \coloneqq \mathbb{E}(I|G=1) - \mathbb{E}(I|G=0).$$

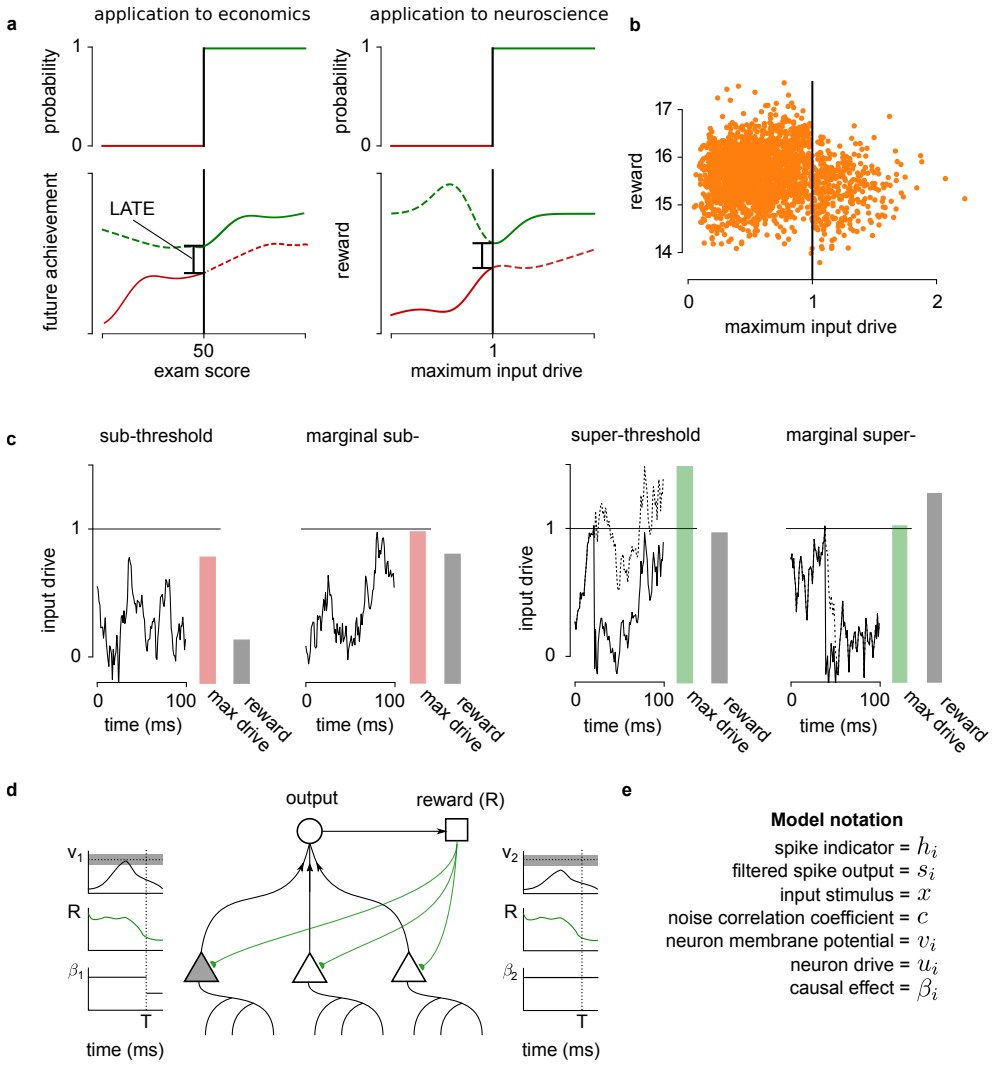


Figure 1: **Applications of regression discontinuity design. a**, (left) In education, the effect of mandatory classes given to students who fail an exam can be used to infer the effect of the classes by focusing on students at the threshold. The discontinuity at the threshold is then a meaningful estimate of the local average treatment effect (LATE), or causal effect. (right) In neuroscience, the effect of a spike on a reward function can be determined by considering cases when the neuron is driven to be just above or just below threshold. **b**, The maximum drive versus the reward shows a discontinuity at the spiking threshold, which represents the causal effect. **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 RDD 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.

However there will be differences between the two groups, e.g. stronger students will tend have been admitted to the more selective school in the first place. Effects based on student skills and the high school attended will be superimposed, confounding the estimate.

A more meaningful estimate comes from focusing on marginal cases. If we compare the students that are right below the threshold and those that are right above the threshold then they will effectively have the same exam performance. And, since exam performance is noisy, the statistical difference between marginally sub- and super- threshold students will be negligible. Therefore the difference in outcome between these two populations of students will be attributable *only* to the high school attended, providing a measure of causal effect (Fig. 1a). If $\chi$ is the threshold exam score, then RDD computes

$$\beta^{RD} := \lim_{x \to \chi^+} \mathbb{E}(I|G = x) - \lim_{x \to \chi^-} \mathbb{E}(I|G = x).$$

This estimates the causal effect of treatments without requiring the injection of noise. RDD uses local regression near the threshold to obtain statistical power while avoiding confounding.

Neurons that are not subject to external noise injection have to solve exactly the same causal inference problem (Fig. 1a). They spike when their maximal drive $Z_i$ exceeds a threshold, in analogy to the score in the schooling example. Through neuromodulator signals a neuron may receive feedback on a reward signal $R$ [27, 5], analogue to the future achievement. The comparison in reward between time periods when a neuron almost reaches its firing threshold to moments when it just reaches its threshold analogously allows an RDD estimate of its own causal effect (Fig. 1b, c). Rather than using randomized perturbations from an additional noise source, a neuron can take advantage of the interaction of its threshold with its drive.

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

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

Here $H_i$ is neuron $i$'s spiking indicator function, $\gamma_i, \alpha_{li}$ and $\alpha_{ri}$ 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 $\beta_i$ represents the causal effect of neuron $i$'s spiking. The neural drive we will use 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 RDD.

How could a neuron use RDD to estimate causal effects? 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 [31], thereby introducing confounding. The neural output determines a non-convex reward signal $R$. This setting allows us to test if neurons can conceivably implement RDD.

The difficulty in estimating a causal effect is that other neurons' activity confounds activation with reward. In the economics analogy we may think of large teams of interacting workers in a company producing some goods: what is everyone's contribution? A simplified RDD estimator that considers only average difference in reward above and below threshold within a window $p$, rather than a linear model, reveals this confounding (Fig. 2a). The locally linear RDD model, on the other hand, is more robust to this confounding (Fig. 2b). Thus the linear correction that is the basis of many RDD implementations [15] allows neurons to readily estimate their causal effect.

To investigate the robustness of the RDD estimator, we systematically vary the weights, $w_i$, of the network. RDD works better when activity is fluctuation-driven and at a lower firing rate (Fig. 2c). RDD

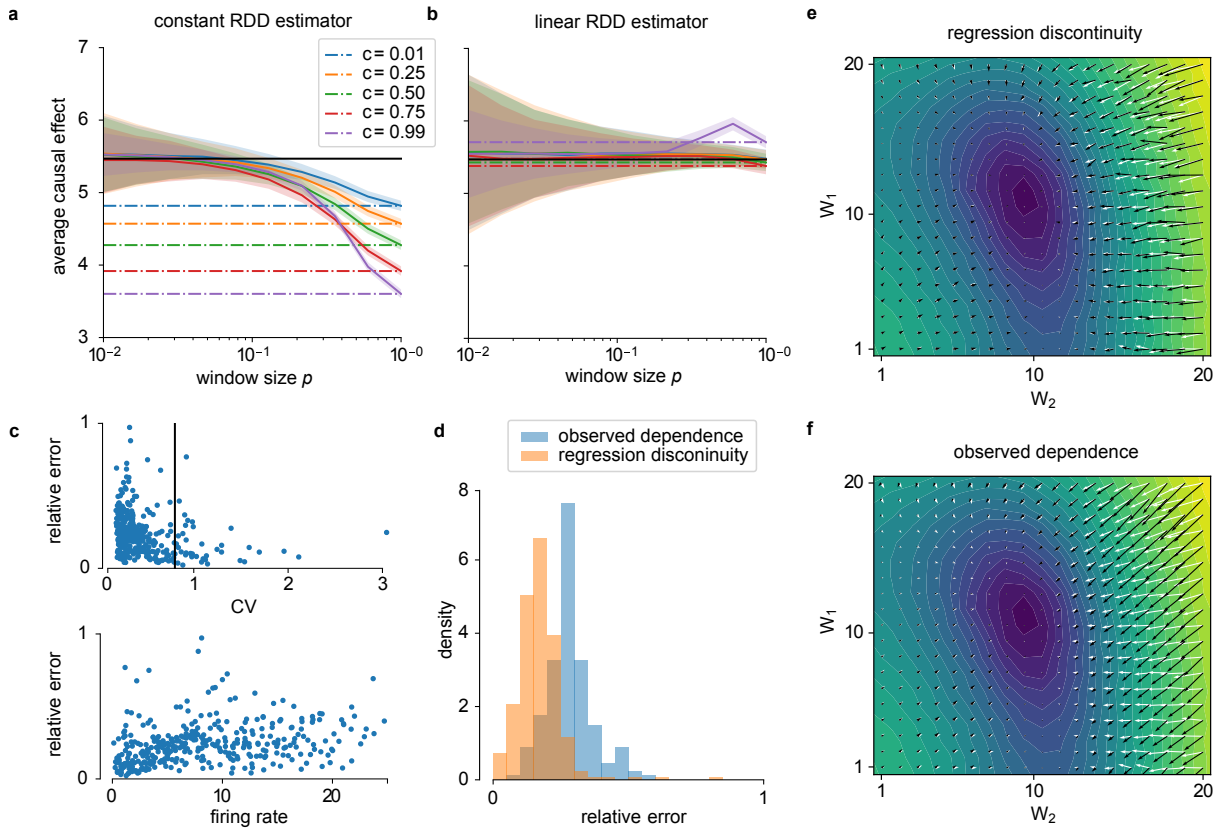


Figure 2: **Estimating reward gradient with RDD in two-neuron network.** **a**, Estimates of causal effect (black line) using a constant RDD model (difference in mean reward when neuron is within a window $p$ of threshold) reveals confounding for high $p$ values and highly correlated activity. $p = 1$ represents the observed dependence, revealing the extent of confounding (dashed lines). **b**, The linear RDD 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, RDD 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 RDD (**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.

is less biased than the observed dependence (Fig. 2d). Thus RDD is most applicable in irregular but synchronous activity regimes [4]. The causal effect can be used to estimate $\frac{\partial R}{\partial w_i}$ (Fig. 2e,f), and thus the RDD estimator may be used for learning weights that maximize the expected reward (see Methods).

To demonstrate how a neuron can learn $\beta$ through RDD, we derive an online learning rule from the linear model. The rule takes the form:

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

where $\mathbf{u}_i$ are the parameters of the linear model required to estimate $\beta_i$, $\eta$ is a learning rate, and $\mathbf{a}_i$ are drive-dependent terms (see Methods). This plasticity rule, where both a reward signal and activation can switch the sign of plasticity, is compatible with the interaction of modulatory influences of neuromodulators and neuronal firing [28, 2].

When applied to the toy network, the online learning rule (Fig. 3a) estimates $\beta$ over the course of seconds (Fig. 3b). When the estimated $\beta$ is then used to maximize expected reward in an unconfounded network (uncorrelated – $c = 0.01$), RDD-based learning exhibits higher variance than learning using the observed dependence. RDD-based learning exhibits trajectories that are initially meander while the estimate of $\beta$ settles down (Fig. 3c). When a confounded network (correlated – $c = 0.5$) is used RDD exhibits similar performance, while learning based on the observed dependence sometimes fails to converge due to the bias in gradient estimate. In this case RDD also converges faster than learning based on observed dependence (Fig. 3d,e).

This paper is a first step to introduce the RDD to neuronal learning. It serves to illustrate the difference in behavior of RDD and observed-dependence learning in the presence of confounding, but is by no means optimized for performance. Further, in many ways it can and should be extended: our model does not solve temporal credit assignment; it does not deal with large, interesting, systems; and it does not specify where presynaptic variance comes from. Nonetheless, RDD is one of the few known ways of statistically dealing with confounders, and an example of a larger class of methods called pseudo-experiments [23]. Demonstrations that segregated neuronal models [11, 17] and synthetic gradient methods [6] can solve deep learning problems at scale inspire future work.

Within reinforcement learning, there exist two popular approaches for estimating causality, each based on utilizing different kinds of intrinsic noise. In perturbation-based methods, a separate noise process is purposefully injected into the system and a mechanism for the system to understand responses as being either 'natural' or 'perturbation-caused' is used [3, 7, 8, 20]. In REINFORCE-type schemes [34], the noise instead comes from the biophysical properties of neurons, e.g. their Poisson spiking [35, 29]. In RDD approaches, on the other hand, it is sufficient that something, in fact anything that is presynaptic, produces variability. As such, RDD approaches do not require the noise source to be directly measured.

Further, in previous work, spiking is typically seen as a disadvantage and systems aim to remove spiking discontinuities through smoothing responses [14, 13, 19]. The RDD rule, on the other hand, exploits the spiking discontinuity. Moreover, finite difference approaches like the method derived here also have the benefit that they 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 supply us with gradients (unlike its simulators [33]). Spiking may, in this sense, allow a natural way of understanding a neuron's causal influence in a complex world.

The most important aspect of RDD is the explicit focus on causality. A causal model is one that can describe the effects of an agent's actions on an environment. Thus learning through the reinforcement of an agent's actions relies, even if implicitly, on a causal understanding of the environment [9, 18]. Here, by explicitly casting learning as a problem of causal inference we have developed a novel learning rule for spiking

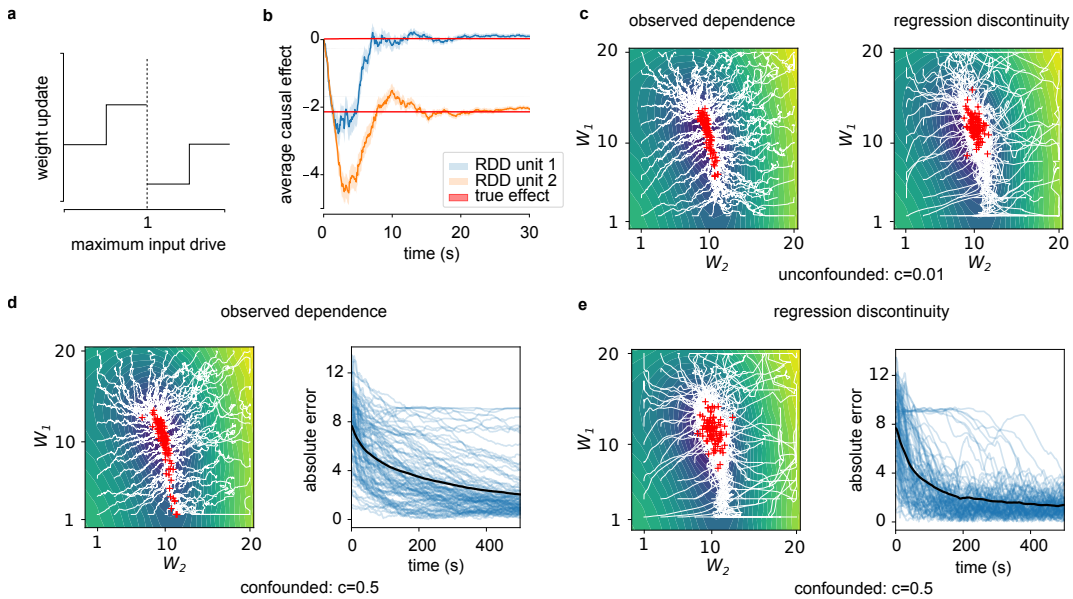


Figure 3: **Applying the RDD learning rule. a**, Sign of RDD learning rule updates are based on whether neuron is driven marginally below or above threshold. **b**, Applying rule to estimate $\beta$ for two sample neurons shows convergence within 10s (red curves). Error bars represent standard error of the mean. **c**, Convergence of observed dependence (left) and RDD (right) learning rule to unconfounded network ($c = 0.01$). Observed dependence converges more directly to bottom of valley, while RDD trajectories have higher variance. **d,e**, Convergence of observed dependence (**d**) and RDD (**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 RDD succeeds.

neural networks. Causality is what really matters in life and, as such, we believe that focusing on causality is essential when thinking about the brain.

# Methods

## The causal effect

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 $\mathcal{G}$ [25]. 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, $\text{Pa}_{X_i}$, 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 local average treatment effect (LATE), or just causal effect, between an outcome variable $X_j$ and a binary variable $X_i$ can be defined as:

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

We will 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

$$\mathbb{E}(X_j|\mathrm{do}(X_i = y)) = \mathbb{E}\left(\mathbb{E}(X_j|\mathbf{S}_{ij}, X_i = y)\right). \tag{1}$$

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, $R$:

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

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

## 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

$$\tau_s \dot{s}_i = -s_i + h_i(t), \tag{2}$$

for synaptic time scale $\tau_s$. 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

$$\dot{v}_i = -g_L v_i + w_i \eta_i, \tag{3}$$

where integrate and fire means simply:

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

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

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

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 [35], the firing rate of a noisy integrate and fire neuron is

$$\mu_i = \left[\frac{1}{g_L} \int_0^\infty \frac{1}{u} \left(\exp\left(-u^2 + 2y_i^{th} u\right) - \exp\left(-u^2 + 2y_i^r u\right)\right) du\right]^{-1}, \tag{4}$$

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 $T$:

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

RDD 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:

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

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 (3) generate an ergodic Markov process with a stationary distribution denoted $\rho$. We consider the problem of finding network parameters that maximize the expected reward with respect to $\rho$. In reinforcement learning, performing optimization directly on the expected reward leads to policy gradient methods [32]. These typically rely on either finite difference approximations or a likelihood-ratio decomposition. Both approaches ultimately can be seen as performing stochastic gradient descent, updating parameters by approximating the expected reward gradient:

$$\nabla_w \mathbb{E}_\rho R, \tag{5}$$

for neural network parameters $w$. Here capital letters are used to denote the random variables drawn from the stationary distribution, corresponding to their dynamic lower-case equivalent above.

Manipulating the expectation using a likelihood-ratio trick leads to REINFORCE-based methods [34]. In neural networks, likelihood-ratio based algorithms are known to be higher variance than methods that more directly approximate back-propagation (e.g. Rezende et al 2014 [26]). This motivates considering methods that more directly estimate the gradient terms [26, 12], breaking down (5) as we would with the deterministic expression. Here we focus on cases dominated by the mean reward gradient, meaning we assume the following:

$$\nabla_w \mathbb{E}_\rho R \approx \mathbb{E}_\rho \left[ (\nabla_{\mathbf{S}} R) \frac{\partial \mu}{\partial w} \right], \tag{6}$$

where $\mu$ is the mean activity vector of the neurons.

Fig. 2e suggests that the assumption (6) is reasonable for the case presented here. Of course in general this assumption does not hold, however the method presented here can likely be extended to broader cases. For instance, if we were to assume that the stationary distribution $\rho$ can be approximated as Gaussian then we have:

$$\nabla_w \mathbb{E}_\rho R = \mathbb{E}_\rho \left[ (\nabla_{\mathbf{S}} R) \frac{\partial \mu}{\partial w} + \frac{1}{2} \mathrm{Tr} \left( (\nabla^2_{\mathbf{S}} R) \frac{\partial \Sigma}{\partial w} \right) \right], \tag{7}$$

for $\mu$ and $\Sigma$, the mean and covariance of the Gaussian random vector $\mathbf{S}$ [26]. In this case quasi-Newton methods may be able to make use of the methods derived here. Alternatively, in some episodic learning

cases, the gradient $\nabla_w$ may be computed by unrolling the network so that the parameters separate from the stochastic variables; this is sometimes known as the re-parameterization trick [10, 26, 12].

Thus we derive methods to estimate $\mathbb{E}(\nabla_{\mathbf{S}} R)$, and use it with (6) and (4) for stochastic gradient descent-based learning. We deal with spiking networks, meaning $\mathbf{S}$ is discontinuous at spike times. Therefore it makes sense to consider finite difference approximations to this gradient term.

## Causal effect in neural networks

How can a neuron estimate $\mathbb{E}(\frac{\partial R}{\partial S_i})$? We show that the reward gradient term can be related to the causal effect of a neuron on the reward signal. 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. However this formulation has the advantage that it is defined when $R$ is not differentiable, it does not require $R$ is a deterministic function of $\mathbf{S}$, and does not require that $\Delta_s$ be small.

To consider the effect of a single spike, note that unit $i$ spiking will cause a jump in $S_i$ (according to (2)) compared to not spiking. 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:

$$\begin{aligned}
\beta_i &= \mathbb{E}\left(R|\mathrm{do}(H_i = 1)\right) - \mathbb{E}\left(R|\mathrm{do}(H_i = 0)\right) \\
&= \mathbb{E}\left(\mathbb{E}\left(R|\mathbf{S}_{j\neq i}, H_i = 1\right) - \mathbb{E}\left(R|\mathbf{S}_{j\neq i}, H_i = 0\right)\right) \\
&\approx \Delta_s \mathbb{E}(\mathbb{E}\left(D_i R(S_i, \mathbf{S}_{j\neq i})|\mathbf{S}_{j\neq i}, H_i = 0\right)) \\
&= \Delta_s \mathbb{E}(D_i R(S_i, \mathbf{S}_{j\neq i})|\mathrm{do}(H_i = 0)).
\end{aligned} \tag{8}$$

A derivation is presented in the supplementary material (Section A).

## Using regression discontinuity design

For comparison we define the observed dependence $\beta_i^{OD}$ as:

$$\beta_i^{OD} \coloneqq \mathbb{E}(R|H_i = 1) - \mathbb{E}(R|H_i = 0).$$

This of course provides an estimate of $\beta_i$ only when $H_i$ is independent of other neurons in the network. In general the causal effect is confounded through correlation with other units.

As described in the main text, to remove confounding, RDD considers only the marginal super- and sub-threshold periods of time. 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 RDD method estimates [15]:

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

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

To estimate $\beta_i^{RD}$, 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 $\gamma_i, \alpha_{li}$ and $\alpha_{ri}$ are nuisance parameters, and $\beta_i$ is the causal effect of interest. This means we can estimate $\beta_i^{RD}$ 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 $\beta_i^{RD}$ 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 $\theta$.

### Implementation

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

## A The relation between causal effect and the finite difference operator

Here we present a more detailed derivation of (8), which relates the causal effect to a finite difference approximation of the reward gradient. 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 $\Delta_s$ 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, (2), 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 (Suppl. Fig. 1).

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 the result:

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

# References


[1] 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.

[2] 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.

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

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

[5] Alexander A. Chubykin, Emma B. Roach, Mark F. Bear, and Marshall G Hussain Shuler. A Cholinergic Mechanism for Reward Timing within Primary Visual Cortex. *Neuron*, 77(4):723–735, 2013.

[6] 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.

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

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

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

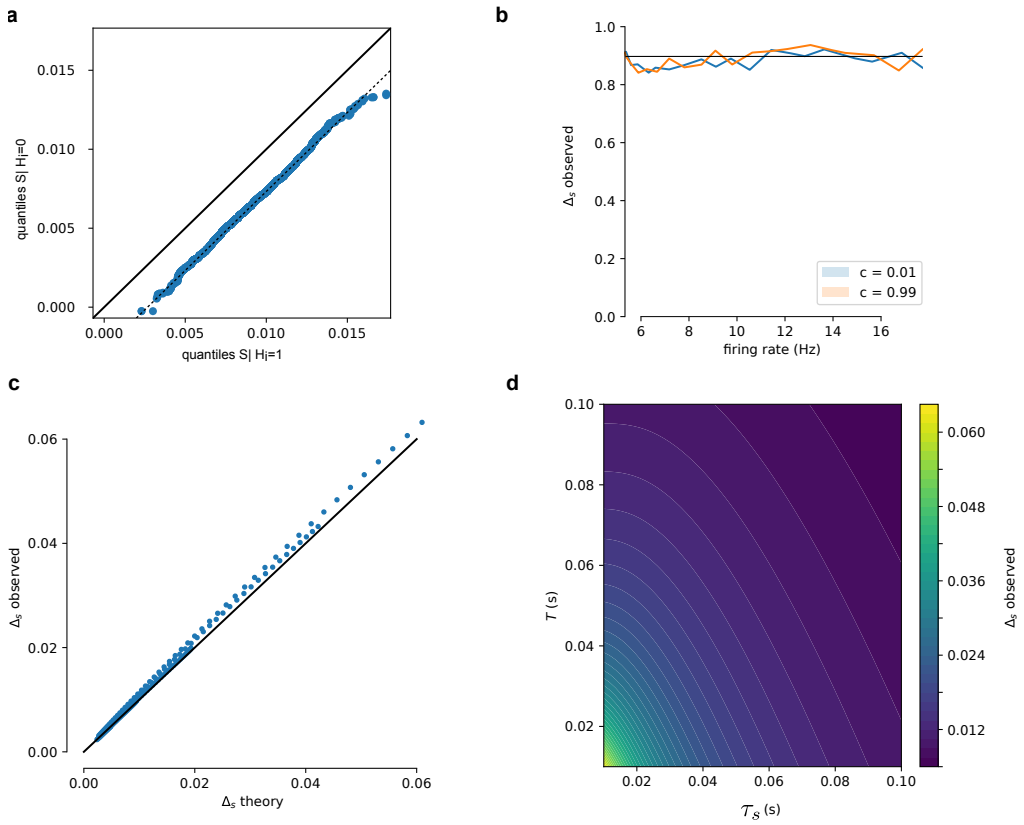


Figure 4: Supplementary Figure 1. **Relation between $S_i$ and $H_i$ over window $T$.** **a**, Simulated spike trains are used to generate $S_i|H_i = 0$ and $S_i|H_i = 1$. QQ-plot shows that $S_i$ following a spike is distributed as a translation of $S_i$ in windows with no spike, as assumed in (10). **b**, This offset, $\Delta_s$, 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 $\Delta_s$ (11) 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.

[10] Alex Graves. Practical Variational Inference for Neural Networks. *Advances in Neural Information Processing Systems*, 24:1–9, 2011.

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

[12] 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.

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

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

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

[16] Ron Kohavi, Randal M. Rm Henne, and Dan Sommerfield. Practical guide to controlled experiments on the web. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, 2007:959–967, 2007.

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

[18] 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. 2017.

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

[20] Robert Legenstein, Steven M Chase, Andrew B Schwartz, and Wolfgang Maas. 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.

[21] Am Lucas and I Mbiti. Effects of School Quality on Student Achievement: Discontinuity Evidence from Kenya. *American Economic Journal: Applied Economics*, 6(3):234–263, 2014.

[22] Marcia L. Meldrum. A brief history of the randomized controlled trial: Oranges and Lemons to the Gold Standard. *Hematology/Oncology Clinics of North America*, 14(4):745–760, 2000.

[23] Bruce Meyer. Natural and Quasi-Experiments in Economics. *Journal of Business & Economic Statistics*, 13(2):151–161, 1994.

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

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

[26] 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.

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

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

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

[36] Joel Zylberberg, Jon Cafaro, Maxwell H Turner, Eric Shea-brown, and Fred Rieke. Direction-Selective Circuits Shape Noise to Ensure a Precise Population Code. *Neuron*, 89(2):369–383, 2016.