CaMKII activation supports reward-based neural network optimization through Hamiltonian sampling

Zhaofei Yu[1,*], David Kappel[2,*], Robert Legenstein[2,*], Sen Song[3], Feng Chen[1] and Wolfgang Maass[2]

1) Department of Automation
Tsinghua University
100084 Beijing, China
chenfeng@mails.tsinghua.edu.cn

2) Institute for Theoretical Computer Science
Graz University of Technology
8010 Graz, Austria
maass@igi.tugraz.at

3) Department of Biomedical Engineering
Tsinghua University
100084 Beijing, China
sen.song@gmail.com

*) These authors contributed equally to this work

May 16, 2018

Abstract

Synaptic plasticity is implemented and controlled through over thousand different types of molecules in the postsynaptic density and presynaptic boutons that assume a staggering array of different states through phosporylation and other mechanisms. One of the most prominent molecule in the postsynaptic density is CaMKII, that is described in molecular biology as a "memory molecule" that can integrate through auto-phosporylation Ca-influx signals on a relatively large time scale of dozens of seconds. The functional impact of this memory mechanism is largely unknown. We show that the experimental data on the specific role of CaMKII activation in dopamine-gated spine consolidation suggest a general functional role in speeding up reward-guided search for network configurations that maximize reward expectation. Our theoretical analysis shows that stochastic search could in principle even attain optimal network configurations by emulating one of the most well-known nonlinear optimization methods, simulated annealing. But this optimization is usually impeded by slowness of stochastic search at a given temperature. We propose that CaMKII contributes a momentum term that substantially speeds up this search. In particular, it allows the network to overcome saddle points of the fitness function. The resulting improved stochastic policy search can be understood on a more abstract level as Hamiltonian sampling, which is known to be one of the most efficient stochastic search methods.

1 Introduction

Calcium-calmodulin dependent protein kinase II (CaMKII) is the most frequently occurring complex molecule in the postsynaptic density and a key molecule for the implementation of synaptic plasticity [1] (see Fig. 1A). It is described in molecular biology as a "memory molecule" that creates through its somewhat persistent autophosphorylated (active) state a short term memory or low pass filter on the time scale of dozens of seconds for calcium influx (see e.g. Ch. 15 in [2], Fig. 1c in [3], and Fig. 3F in [4]). Calcium influx is a typical feature of the induction of longterm plasticity via NMDA receptors. More specifically, incoming calcium transforms CaMKII via calmodulin into its active state, which is maintained for a while via autophosphorylation among its 12 subunits. Furthermore CaMKII triggers in its activated state changes of synaptic efficacy through the phosphorylation of AMPA receptors, the anchoring of additional AMPA receptors in the postsynaptic density, and dopamine-gated stabilization of spines (see e.g. Fig. 3, S5, S11 in [4]).

Although numerous experimental data show that CaMKII in its activated state is essential both for LTP and LTD [5, 6], its contribution to network plasticity is still unclear. We address in this article the question how the activation dynamics of CaMKII could contribute to reward-based network optimization for specific computational tasks. Since the molecular processes that involve CaMKII and give rise to LTP and LTD contain a strong stochastic component, it is natural to view this optimization not as a deterministic but a stochastic search for good network parameters. This view is also consistent with numerous experimental data that show that synaptic connections are even in the adult cortex subject to a continuous coming and going of dendritic spines that appears to be inherently stochastic and independent of pre- or postsynaptic firing in the absence of a functional synaptic connection [7, 8]. A theoretical framework for stochastic network plasticity has been introduced in [9, 10] and termed synaptic sampling. There, it was shown that a neural network $\mathcal{N}$ with parameters $\boldsymbol{\theta}$ subject to stochastic plasticity rules samples from a stationary distribution $p_T^*(\boldsymbol{\theta})$ of network configurations through a sampling process known as Langevin sampling in the machine learning literature. This means that the network will visit – in the long run – network configurations $\boldsymbol{\theta}$ most often that have a large probability $p_T^*(\boldsymbol{\theta})$. The index T in the distribution $p_T^*(\boldsymbol{\theta})$ denotes the "temperature" of the search, which depends on the amount of noise in the plasticity process. The exact shape of $p_T^*(\boldsymbol{\theta})$ is determined by the plasticity rule and in the context of reward-based learning it can be chosen to prefer network configurations that lead frequently to large rewards. In other words, in this framework, synaptic plasticity can be shown to implement an ongoing stochastic policy search [10].

However, as synaptic sampling carries out Langevin sampling, convergence to the stationary distribution is rather slow for any fixed temperature, which is in general undesirable as it implies slow learning. In particular, the search for high fitness regions by gradient-based optimization techniques such as Langevin sampling is hindered by local optima and – even more severely as recently suggested by Dauphin et al. [11, 12] – by the presence of saddle points in $p_T^*(\boldsymbol{\theta})$. This slowness is a generic impediment for the implementation of a global optimization strategy such as simulated annealing as further detailed below.

In this article, we show that CaMKII activation dynamics can ease these issues. Compared to the synaptic sampling framework, the activation dynamics of CaMKII gives rise to an additional dynamic variable that basically low-pass filters parameter updates. This low-pass filtering implements a momentum term, a method that is well-known to improve gradient-based optimization in many circumstances, for example in the vicinity of saddle points. More abstractly, in our

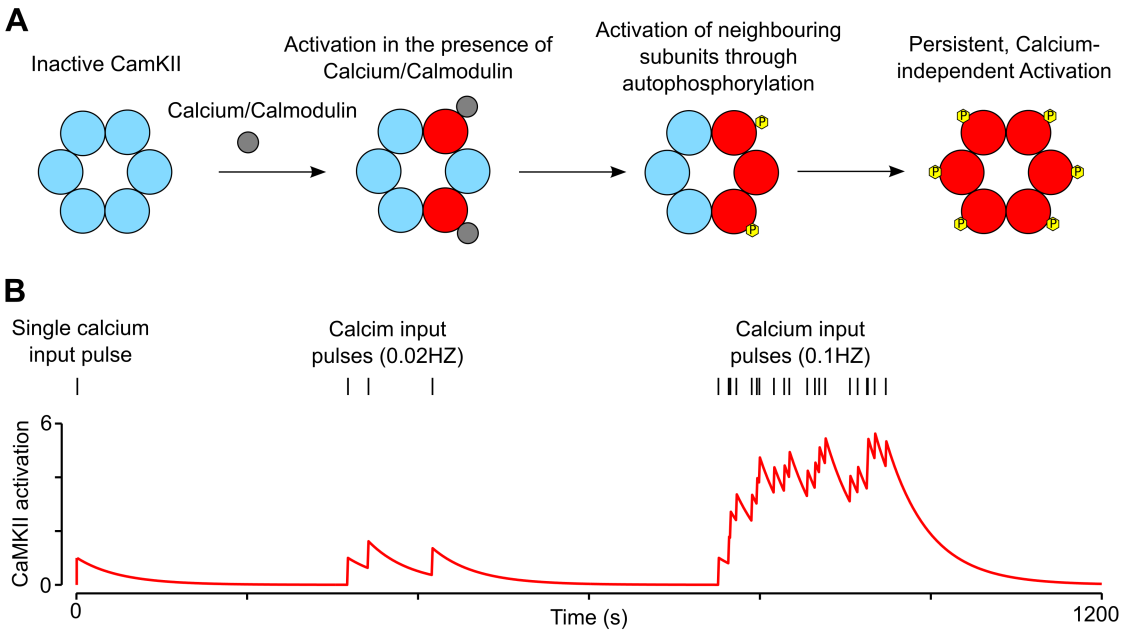

Fig. 1. **CaMKII dynamics. A:** CaMKII has a crystal structure of 12 units, which fold into two rings of six domains. Incoming calcium/calmodulin (grey) can transform CaMKII into its active state (red), and then activate the neighboring subunits through autophosphorylation. This leads to persistent and calcium-independent activation of the CaMKII on the time scale of dozens of seconds (figure adapted from [14]). **B:** The local concentration of CaMKII in its activated state changes in response to different input pulse. CaMKII activation jumps upward by 1 with a single calcium input pulse and otherwise decays exponentially. For Poisson calcium input pulses, CaMKII activation is irregular for low frequency input (0.02HZ) and almost steady for high frequency input (0.1HZ). Note that the time constant of CaMKII is set to 50 s here.

stochastic framework, we show that the resulting dynamics gives rise to a parameter sampling algorithm known as Hamiltonian sampling, that however still samples from the same stationary distribution $p_T^*(\boldsymbol{\theta})$. A well-known advantage of Hamiltonian sampling over Langevin sampling is faster convergence to the stationary distribution [13].

With such faster convergence properties, our model for CaMKII driven plasticity allows us to create a link from reward-based learning to optimization theory, which establishes conditions under which a neural circuit could attain not only functionally attractive locally optimal network configurations, but in principle even a global optimum. Simulated annealing [15, 16] is arguably one of the most powerful algorithmic approach to nonlinear optimization. Evolutionary algorithms also work well in some cases, but require a large control overhead and many competing networks in parallel for which no biological evidence exists so far. We show that reward-based network plasticity can in principle reach even globally optimal network configurations $\boldsymbol{\theta}$ if the amount of stochasticity is sufficiently slowly decreased during learning ("cooling" or "annealing"), similar to simulated annealing in continuous time. This theoretical result provides a new gold standard for reward-based network learning.

2 Results

Consider a network $\mathcal{N}$ that receives at certain times t reward signals $r(t)$, e.g., in the form of dopamine. The dynamics of each synaptic connection i in the network $\mathcal{N}$ is modeled by a parameter $\theta_i(t)$, which determines the synaptic efficacy. Therefore, we assume that the behavior of the network (i.e., its response to network input; also referred to as the network policy) is determined by the parameter vector $\boldsymbol{\theta}$ (the vector of all synaptic parameters). In biological neuronal networks, neurons are either excitatory or inhibitory, a fact that is commonly refered to as Dale's principle. This implies that their outgoing synaptic connections are exclusively excitatory (modelled as positive synaptic weights) or inhibitory (negative synaptic weights), and that these synaptic weights cannot change the their sign through plasticity processes. We will first introduce a version of the model that allows such a sign-switch of synaptic weights for demonstration purposes. We will later introduce a slightly modified version of the model where only excitatory synapses are plastic with weights constrained to be non-negative.

Previous work [10] has analyzed under which conditions such a network can perform an ongoing stochastic policy search. That is, under which conditions local stochastic synaptic plasticity processes on $\boldsymbol{\theta}$ can achieve that the network $\mathcal{N}$ seeks network configurations that provide a large expected discounted reward. Mathematically, the expected discounted reward $\mathcal{V}(\boldsymbol{\theta})$ for a given parameter vector $\boldsymbol{\theta}$ is given by

$$\mathcal{V}(\boldsymbol{\theta}) \;=\; \left\langle \int_0^\infty e^{-\frac{\tau}{\tau_e}}\, r(\tau)\, d\tau \right\rangle_{p(\mathbf{r}|\boldsymbol{\theta})} . \tag{1}$$

The integral integrates all future rewards $r(\tau)$, while discounting more remote rewards exponentially with a discount rate τ_e. The expectation is an average over multiple learning episodes where in each episode one realization of the reward trajectory $\mathbf{r}$ is encountered for the given parameters $\boldsymbol{\theta}$ according to some distribution $p(\mathbf{r}|\boldsymbol{\theta})$.

In addition, a biological network $\mathcal{N}$ needs to satisfy structural constraints, such as sparse connectivity, that can be formulated through a prior $p_{\mathcal{S}}(\boldsymbol{\theta})$ over network configurations $\boldsymbol{\theta}$ [10]. Hence, network learning can be regarded as a search for policies (i.e., network configurations $\boldsymbol{\theta}$) that both satisfy structural constraints and provide a large expected discounted reward. This can be stated more formally as a sampling from the posterior distribution $p^*(\boldsymbol{\theta})$ of parameters

$$p^*(\boldsymbol{\theta}) \propto p_{\mathcal{S}}(\boldsymbol{\theta})\; \mathcal{V}(\boldsymbol{\theta}). \tag{2}$$

It was shown in [10] that if the stochastic dynamics of each parameter θ_i can be characterized through a stochastic differential equation (SDE) of the form

$$d\theta_i \;=\; \beta \left(\frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta}) \right) dt \;+\; \sqrt{2T\beta}\, d\mathcal{W}_i \,, \tag{3}$$

then the network reaches the unique stationary distribution given by the posterior $p_T^*(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}} p^*(\boldsymbol{\theta})^{\frac{1}{T}}$ and then samples from this distribution over network configurations. The parameter $\beta > 0$ denotes a learning rate that controls the speed of the parameter dynamics. The last term $d\mathcal{W}_i$ of Eq. (4) describes infinitesimal stochastic increments and decrements of a Wiener process $\mathcal{W}_i$ – a standard model for Brownian motion in one dimension (see [17]). The amplitude of this

noise term is scaled by the temperature parameter $T > 0$, which can be used to increase or decrease random exploration of the parameter space.

To integrate the role of CaMKII in the plasticity processes, we model the previously sketched transient role of CaMKII as a low pass filter in the induction of synaptic plasticity (see Fig. 1B). For each potential synapse i, we introduce another dynamic variable $\Gamma_i(t)$ that determines the change of the $\theta_i(t)$ at time t. It was found that both, LTP and LTD require the activated form of CaMKII, and that the switch between LTP and LTD is determined by other mechanisms [5, 6]. We therefore interpret the absolute value of $\Gamma_i(t)$ as the local concentration of CaMKII in its activated state. The interaction of these two variables is modeled by the stochastic differential equation (SDE) of the form

$$\begin{aligned} d\theta_i &= a\ \Gamma_i\ dt \\ d\Gamma_i &= \left(a\ \tfrac{\partial}{\partial \theta_i}\ \log p^*(\boldsymbol{\theta})\ - b\Gamma_i\right) dt\ +\ \sqrt{2Tb}\, d\mathcal{W}_{\Gamma_i}\ \ , \end{aligned} \tag{4}$$

where the change of parameter θ_i directly depends on the value of the hidden CaMKII-related variable Γ_i ($a > 0$ is a learning rate). The dynamics of Γ_i in turn is determined by three terms. The first term is the gradient of the parameter posterior. In reward-based learning, this gradient can be estimated by a rule that depends only on pre- and post-synaptic spike times and a global reward signal implemented for example as a dopaminergic signal. The friction term $-b\Gamma_i$ implements the decay of CaMKII activation with a time constant b. Detailed experimental studies suggest that this time constant depends on a variety of factors, e.g. the inactivation time constant of CaMKII activity and the mobility of CaMKII [18, 19, 20] (we used 10 s in Fig. 2 and 50 s in the remainder of the paper). The last term models noise on CaMKII activation, such as stochastic opening of N-methyl-d-aspartate (NMDA) receptor channels [21].

With these extended parameter dynamics, the network samples from the posterior $p^*_T(\boldsymbol{\theta}, \boldsymbol{\Gamma}) = \frac{1}{\mathcal{Z}} p^*(\boldsymbol{\theta})^{\frac{1}{T}} p^*(\boldsymbol{\Gamma})^{\frac{1}{T}}$ over network configurations (see Theorem 1 in Methods for details). By marginalization over the CaMKII parameters $\boldsymbol{\Gamma}$ it then follows that the stationary distribution over the synaptic parameters again is given by $p^*_T(\boldsymbol{\theta}) = \int p^*_T(\boldsymbol{\theta}, \boldsymbol{\Gamma})\ d\boldsymbol{\Gamma} = \frac{1}{\mathcal{Z}} p^*(\boldsymbol{\theta})^{\frac{1}{T}}$.

In other words, the CaMKII-enriched dynamics gives rise to the same reward-optimizing distribution over network configurations as the direct dynamics considered in [10]. Importantly however, it turns out that the dynamics (4) actually posesses advantageous properties when compared to the direct dynamics (3). For the noise-less case ($T = 0$), the dynamics (3) corresponds to a gradient ascent on $p^*(\boldsymbol{\theta})$. In comparison, the dynamics (4) introduces a momentum term which is well-known to improve gradient descent in many circumstances, for example in the presence of small local optima or in the vicinity of saddle points. In the case with noise, the dynamics (3) corresponds to Langevin sampling from $p^*(\boldsymbol{\theta})$, and the dynamics (4) to Hamiltonian sampling with friction. It is knwon that Hamiltonian sampling typically shows much faster convergence to the stationary distribution than the rather slow Langevin sampling [22]. In fact, a similar low-pass filtering of gradient updates was already implemented in [10] to improve learning performance, but without a clean mathematical background and biological motivation.

To arrive at concrete plasticity rules, one has to determine $\frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta}) = \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + \frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ in Eq. (4), for the concrete neuron model and prior $p_{\mathcal{S}}(\boldsymbol{\theta})$ at hand. As one example to be used in subsequent simulations, we consider a stochastic spiking neuron model (see Spiking neuron model) and independent zero-mean Gaussian priors with variance σ^2 for each parameter θ_i. We obtain $\frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) = -\frac{1}{\sigma^2}\theta_i$ for the derivative of the prior. Using this and Eq. (1) we find

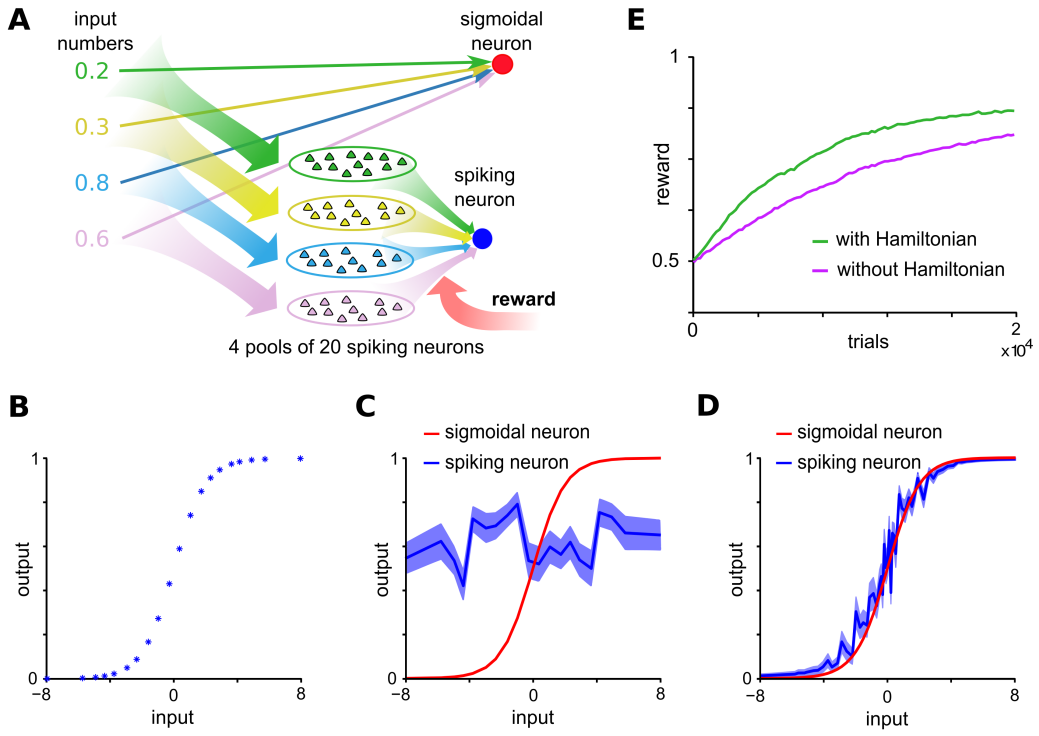

Fig. 2. **A spiking neuron learns to emulate a sigmoidal neuron. A:** Illustration of the network architecture. The target firing activity of the spiking neuron (blue) is defined by the output of a sigmoidal neuron (red) with four inputs and pre-defined weights. The spiking neuron receives inputs from 4 pools of 20 spiking neurons each, with firing rates proportional to the sigmoidal neurons' inputs. **B:** The distribution of the 20 input pattern used during learning on the input-output plane of the sigmoidal neuron (x-axis: weighted sum of the four inputs). **C**, **D:** Output of sigmoidal neuron (red) and firing probability of spiking neuron (blue) as a function of the weighted sum of inputs before (C) and after (D) learning through Hamiltonian dynamics. The spiking neuron approximates the smooth behavior of the sigmoidal neuron after learning. **E:** Comparison of the average rewards for synaptic sampling with (green) and without (magenta) Hamiltonian dynamics throughout learning (average over 50 trials).

that the derivative $\frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta})$ in Eq. (4) at time t can be approximated by (see Methods)

$$\frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta}) \;=\; \frac{\partial}{\partial \theta_i} \log p_{\mathcal{S}}(\boldsymbol{\theta}) + \frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta}) \;\approx\; r(t)\, e_i(t) - \tfrac{1}{\sigma^2} \theta_i(t) \tag{5}$$

$$\frac{de_i(t)}{dt} = -\tfrac{1}{\tau_e}\; e_i(t) \;+\; y_{\text{PRE}_i}(t)\,(z_{\text{POST}_i}(t) - f_{\text{POST}_i}(t))\;, \tag{6}$$

where $y_{\text{PRE}_i}(t)$ is the PSP activation under synapse i, $f_{\text{POST}_i}(t)$ denotes the firing probability of the postsynaptic neuron, and $z_{\text{POST}_i}(t)$ is a binary variable that is one if the postsynaptic neuron spiked at time t and zero else. Here the synaptic plasticity rule acts on Γ_i (see Eq. (4)) which is related to CaMKII activation instead of acting directly on the synaptic parameter θ_i. This learning rule is a simple version of reward-modulated spike-timing dependent plasticity (STDP). Similar rules were derived previously in the context of reward-based learning [23, 24]. The current work extends these rules to include a prior over network configurations, stochastic parameter updates, and CaMKII-induced Hamiltonian dynamics.

2.1 A spiking neuron learns to emulate a sigmoidal neuron

Learning in recurrent networks of spiking neurons is notoriously hard [25], in particular with reward-based learning. For example, interesting functionality has been acquired through reward-based synaptic plasticity in [26], but only in recurrent networks of smooth non-spiking neurons. Recently, it has been proposed that functionality of a non-spiking network can be ported to a spiking network if it has previously learned to exhibit smooth dynamics [25]. We wondered whether such smoothing of network responses can be obtained through reward-based learning. We considered a very simple basic setup where the task is to reproduce with a single spiking neuron the behavior of an artificial sigmoidal neuron model (see Fig. 2A).

The target firing rate of the spiking neuron was given by the output of a sigmoidal neuron with four inputs and pre-defined weights. Fig. 2B shows the desired input-output behavior. The spiking neuron received inputs from 4 pools of 20 spiking neurons each, with firing rates proportional to the sigmoidal neurons' inputs (and a maximum of 60 Hz). Input patterns were presented to the spiking neuron continuously while its weights were adapted through reward-based plasticity (at each presentation, one out of 20 patterns was chosen randomly and presented, see Fig. 2B). Each presentation of an input pattern lasted for 300 ms. The presentation was followed by a 10 ms phase where a reward was delivered which was given by 1 minus the absolute difference between spiking neuron and sigmoidal neuron output (see Methods for details). After reward delivery, a 400 ms delay period was introduced where input neurons were silent, followed by another pattern presentation.

Before learning, the firing rate of the spiking neuron was rather random over the whole range of inputs (Fig. 2C). After 20000 pattern presentations, the neurons' firing rate approximated the smooth behavior of the sigmoidal neuron well (Fig. 2D). Fig. 2E shows the average reward throughout learning for Hamiltonian sampling in comparison with non-Hamiltionian dynamics (synaptic sampling). One can see that Hamiltonian dynamics speeds up learning significantly.

2.2 Reward-guided network plasticity

Next we investigated whether the benefit in learning performance of Hamiltonian sampling scales up to biologically more realistic network architectures, that are larger in size and less structured.

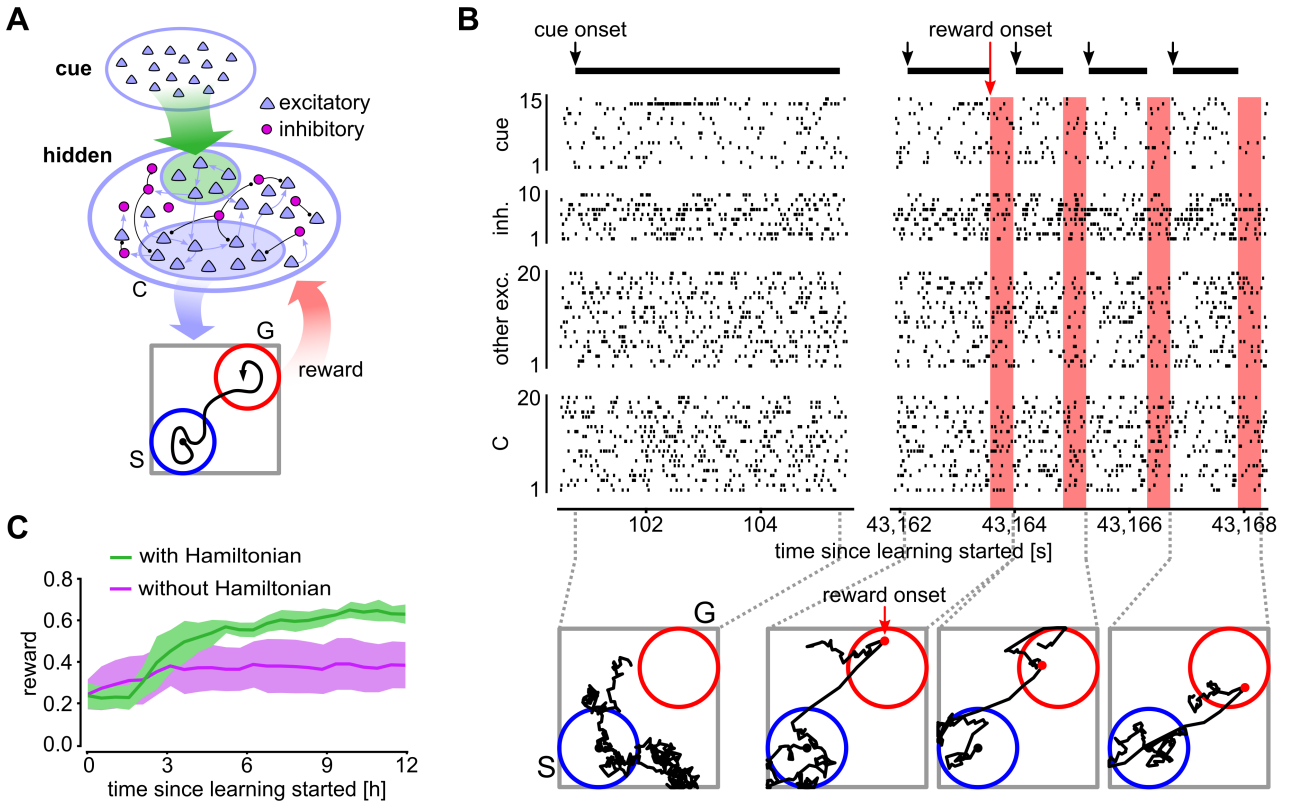

Fig. 3. **Hamiltonian synapse dynamics improves learning a blind reaching task. A:** Illustration of the network architecture and learning task. A recurrent network of inhibitory and excitatory neurons with input from a population of afferent neurons. The arrows indicate symbolically the connectivity between the excitatory neurons (blue), and inhibitory neurons (purple) (random subsets are shown). A pool C of neurons is used to control the position of the cursor in 2D space. The afferent input neurons provide a cue that indicates the phase during which the movement should be performed. Reward is delivered to the network if the cursor reaches the goal location G starting from the start location S. **B:** Activity from random subsets of the network neurons (top) and example cursor trajectories (bottom) at learning onset time and after 12 hours of learning. The black horizontal bars indicate the presentation of the cue pattern. The red vertical bars show the reward windows at the end of successful trials. Network responses and cursor movements become more stereotyped and goal-directed throughout learning. **C:** Comparison of learning curves with (green) and without (magenta) Hamiltonian synaptic dynamics. Reward is quantified here by the mean fraction of successful trials at each time point. If Hamiltonian dynamics is included the network learns the task significantly faster and better. Average results over 5 independent trials are shown, shaded area indicates STD.

To do so, we applied the Hamiltonian synaptic sampling framework outlined above to learn a blind reaching task in a simple model of motor cortex. Reward-guided changes of network activity and task-induced spine dynamics are well documented in motor cortex [27]. We used a network of 100 recurrently connected excitatory neurons and 20 inhibitory neurons to control a cursor in 2D space (see Fig. 3A,B). Connectivity parameters of this cortical network motif were taken from [28] (see Methods). In addition to recurrent connections a random subset of 30 excitatory neurons received input from 200 afferent neurons. From the remaining 70 excitatory neurons we randomly selected a neural pool C of 50 neurons to control the cursor position. For controlling the cursor we adopted the population vector model [29]. Briefly, each neuron in C was assigned a randomly selected preferred direction in 2D cursor space. At each time point the cursor was moved in the direction of the population vector (accumulated preferred directions weighted by neural activities) of the 50 neurons in C.

Each trial started with the cursor centered at the start area S (blue circle in Fig. 3A). The cursor had to be held at S for 50 ms to initiate the movement phase of the trial. The movement phase was indicated through the presentation of a cue pattern (a rate pattern for all 200 afferent input neurons, see Methods). Reward was given to the network if the cursor was moved to the target area G in Fig. 3A and held there for 50 ms. At success, the presentation of the cue pattern was stopped and a 400 ms reward window was initiated during which $r(t)$ was set to 1 (indicated by red vertical bars in Fig. 3B). If the network failed to reach the target within 5 seconds, or failed to hold the cursor at S and G, the trial was aborted and a 400 ms time window without reward was presented.

Note that this is a nontrivial reinforcement learning task, since the neurons did not "know" whether they belonged to the population C. Also, the network did not receive feedback about the cursor position, only binary information about the trial phase through the cue was provided. This is also true for the preferred directions assigned to the neurons in C, which could not be observed by the neurons. Furthermore, the neurons in C did not receive input from the cue directly, such that the routing of cue information to C had to be learned on top of the reaching task. All this information had to be discovered through random exploration from a global and sparse binary reward signal.

We used the synaptic sampling framework with and without the Hamiltonian momentum term to learn this task. Synaptic plasticity was here only active for excitatory synapses (both recurrent and feedforward), whereas inhibitory synapses were fixed. In order to guarantee that synapses didn't change their role, i.e., become inhibitory during learning, we used here a model for synaptic plasticity that does not allow synaptic weights to become negative. This was done by applying a mapping between synaptic parameters $\theta_i(t)$ and the synaptic efficacies $w_i(t)$. We used here the exponential mapping

$$w_i(t) = \exp(\theta_i(t) - \theta_0) \ , \tag{7}$$

with offset parameter $\theta_0 = 3$, such that $w_i(t)$ is positive for any value of $\theta_i(t)$. We show in Methods that by inserting equation (7) into the general Hamiltonian learning framework (4), we arrive at a slightly modified version of the eligibility trace $e_i(t)$, given by

$$\frac{de_i(t)}{dt} = -\frac{1}{\tau_e} \ e_i(t) \ + \ w_i(t)\, y_{\mathbf{pre}_i}(t)\, (z_{\mathbf{post}_i}(t) - f_{\mathbf{post}_i}(t)) \ . \tag{8}$$

This dynamics differs from equation (6) by the additional term $w_i(t)$, such that weight changes

are scaled by the current value of the synaptic efficacy. This feature of our model mimics the multiplicative dynamics observed in cortical synaptic spines [30], see [10] for a detailed analysis.

In Fig. 3 we show that the Hamiltonian momentum term in the rule (4) significantly enhances learning this task. Network responses before and after learning with the Hamiltonian momentum term are shown in Fig. 3B. Initially the rewarded goal is only reached occasionally (around 20% success rate, one example unsuccessful trial is shown). After learning for 12 hours the network is able to reach the target in most of the trials (success rate was 62% on average, see Fig. 3C). In Fig. 3C we compare the learning progress with and without Hamiltonian sampling. We found that this task is hard to learn without the Hamiltonian momentum term (success rate typically below 40% after 12 hours of learning).

2.3 Hamiltonian dynamics improves network behavior at saddle points

Traditionally, it was believed that gradient-based non-convex optimization in high-dimensional spaces is hampered by the presence of local optima in the fitness landscape. Recently, Dauphin et al. [11, 12] argued that in high-dimensional spaces there are typically only few local optima and that these local optima are nearly as good as the global optimum. Importantly, it was further noted by these authors that saddle points are much more numerous in high-dimensional fitness landscapes. Hence, stochastic procedures over high dimensional spaces, like synaptic sampling, tend to be inefficient and time consuming due to the presence of saddle points, but not so much due to local optima. One generally accepted method to speed up convergence of learning or sampling in the presence of saddle points is to use Hamiltonian dynamics (or a momentum term). We therefore hypothesized that CaMKII-induced Hamiltonian parameter dynamics should provide a benefit in this respect.

To test this hypothesis, we considered a three-layer neural network with 784 input neurons, 30 hidden neurons and 10 output neurons. The task was to learn to classify images of handwritten digits from the MNIST dataset (see Fig. 4A and Methods). Due to the large computational demands for this task, we did not consider a spiking network here but rather a network consisting of stochastic perceptrons (i.e., neurons with binary outputs which were set stochastically based on the weighted sum of inputs, similar to units in a Boltzmann machine). At each pattern presentation, one digit was chosen randomly from the MNIST dataset and presented as input. A binary reward was delivered depending on the activity of output neurons. If the output neuron corresponding to the target for the current example had larger firing probability than other output neurons, a reward of 1 was delivered, otherwise the reward was set to 0. Note that no eligibility trace was used as the network obtained feedback immediately (presentation of the pattern, computation of network output, and reward delivery were all performed in the same time step).

We first ran the network with non-Hamiltonian synaptic sampling (Fig. 4B, magenta curve). The behavior of the network during learning showed typical signs of saddle points. In particular, the test accuracy tended to get stuck at some plateau value with only slight increases during longer periods. Then, at some point performance increased significantly (the network escaped from the saddle point) until another plateau was reached (see step-like behaivor of the magenta curve in Fig. 4B). Similar behavior was observed with Hamiltonian dynamics, however, in this case, the network tended to escape from saddle points much faster (Fig. 4B, green curve). To test whether Hamiltonian dynamics can escape saddle points faster than non-Hamiltonian synaptic sampling, we considered a parameter setting obtained by synaptic sampling close to a putative saddle point

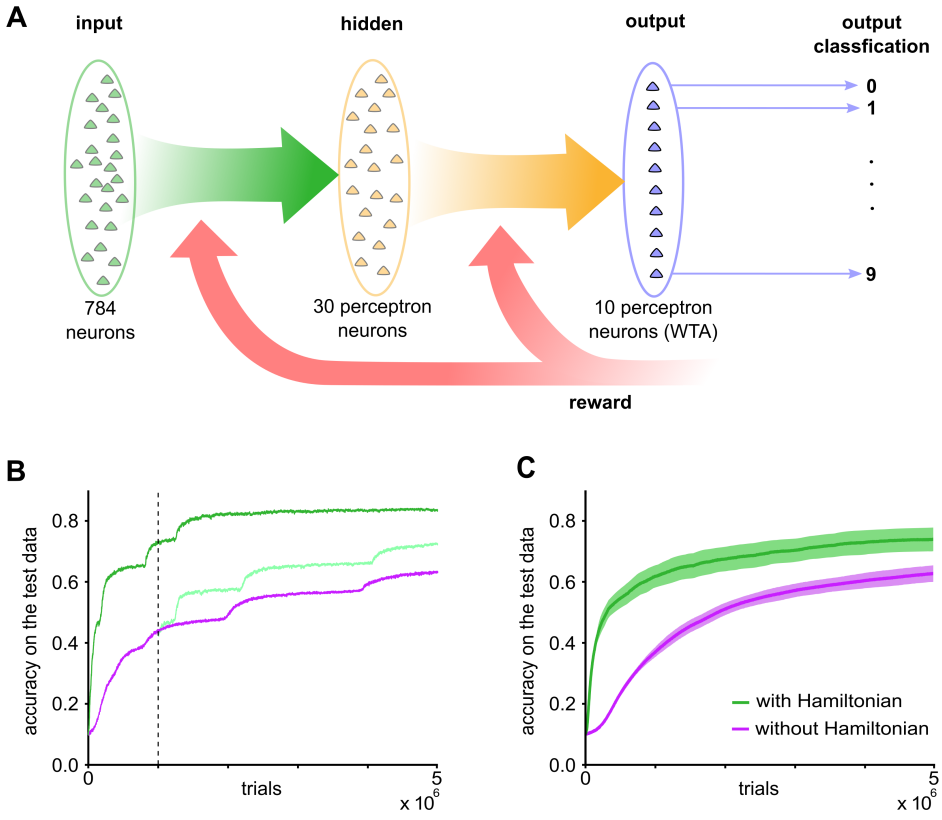

Fig. 4. **Hamiltonian dynamics improves network behavior at saddle points. A:** Network architecture. **B:** Hamiltonian synaptic sampling can ease the saddle point problem. Network performance on the MNIST task during learning with Hamiltonian dynamics (dark green) and non-Hamiltonian synaptic sampling (magenta). With the same initial weights, Hamiltonian dynamics can escape saddle point more quickly. Dashed vertical line indicates time when non-Hamiltonian dynamics (magenta) was switched to Hamiltonian dynamics (light green) in the same network. **C:** Comparison of the average accuracy on the test data for Hamiltonian synaptic sampling (green) and non-Hamiltonian synaptic sampling (magenta; shading shows STDP over 30 independent learning trials).

and continued the simulation with Hamiltonian dynamics (light green curve in Fig. 4B). We observed that the network escaped from the current saddle point much faster with the Hamiltonian dynamics. Considering the average performance for 30 independent learning trials, we found that Hamiltonian sampling accelerates learning significantly and obtains better result within reasonable learning times (Fig. 4C).

2.4 From reward-based learning to global network optimization

Virtually all previous approaches for reward-based learning in spiking neural networks are based on the policy gradient method, that is, the parameters of the network are gradually adjusted in the direction that increases the expected reward locally. Hence, for sufficiently long learning, the parameter setting of the network converges to a local optimum and stays at this local optimum thereafter. The proposed mathematical framework of Hamiltonian sampling allows us to create a link from reinforcement learning to nonlinear optimization theory and the simulated annealing algorithm. This link implies that (spiking or artificial) neural networks can in principle attain through learning not only functionally attractive locally optimal network configurations, but in principle even a global optimum. This theoretical result hence reveals a fundamental advantage of Hamiltonian synaptic dynamics over previous approaches for reward-based network optimization.

The link to nonlinear optimization becomes apparent when one takes a closer look at the temperature parameter T in our plasticity dynamics (4) that scales the amount of noise in the parameter updates. Since for a given T, the network samples from $p_T^*(\boldsymbol{\theta}) = \frac{1}{Z} p^*(\boldsymbol{\theta})^{\frac{1}{T}}$, a decreased temperature $T < 1$ concentrates parameter samples at values that lead to large rewards (for an uninformative prior) and therefore increases the expected reward of the network. In the limit $T \to 0$, the stationary distribution $p_T^*(\boldsymbol{\theta})$ converges to the uniform distribution over optimal parameter settings with other parameter settings assuming zero probability

$$\lim_{T \to 0} p_T^*(\boldsymbol{\theta}) = \begin{cases} \frac{1}{|\mathcal{S}_{\text{opt}}|} & , \quad \text{for } \boldsymbol{\theta} \in \mathcal{S}_{\text{opt}} \\ 0 & , \quad \text{for } \boldsymbol{\theta} \notin \mathcal{S}_{\text{opt}} \end{cases} , \tag{9}$$

where we have defined $\mathcal{S}_{\text{opt}}$ as the set of optimal network parameters and $|\mathcal{S}_{\text{opt}}| \equiv \int_{\boldsymbol{\theta} \in \mathcal{S}_{\text{opt}}} d\boldsymbol{\theta}$ denotes the measure of this set, see Methods. Further, the expected reward also assumes its global optimum in this limit. One attempt to attain such an optimum is to start with a large temperature and reduce it slowly towards 0. Such an annealing procedure is used in simulated annealing, a non-linear optimization technique [16]. This cooling technique however needs convergence to the associated stationary distribution for each temperature T within a reasonable time. While some data suggest that the genetic program for developmental learning has some features that are reminiscent of a cooling schedule [31], a Hamiltonian sampling dynamics is likely to improve the convergence speed for each temperature.

We studied the benefit of a cooling schedule by considering a spiking neural network that learns the exclusive-or (XOR) function through reward-based learning (Fig. 5A,B). The XOR function maps two binary variables to one binary output in the following manner: $(0,0) \to 0, (1,1) \to 0, (0,1) \to 1, (1,0) \to 1$. It is a classical task for artificial neural networks. The spiking neural network that we used for this task is shown in Fig. 5A. It consisted of 2 input neurons, 10 hidden neurons and 1 output neuron. Each input neuron encoded one binary input variable. It produced a Poisson spike train at its output with a rate of 80 Hz for the input 1 and 3 Hz for the input

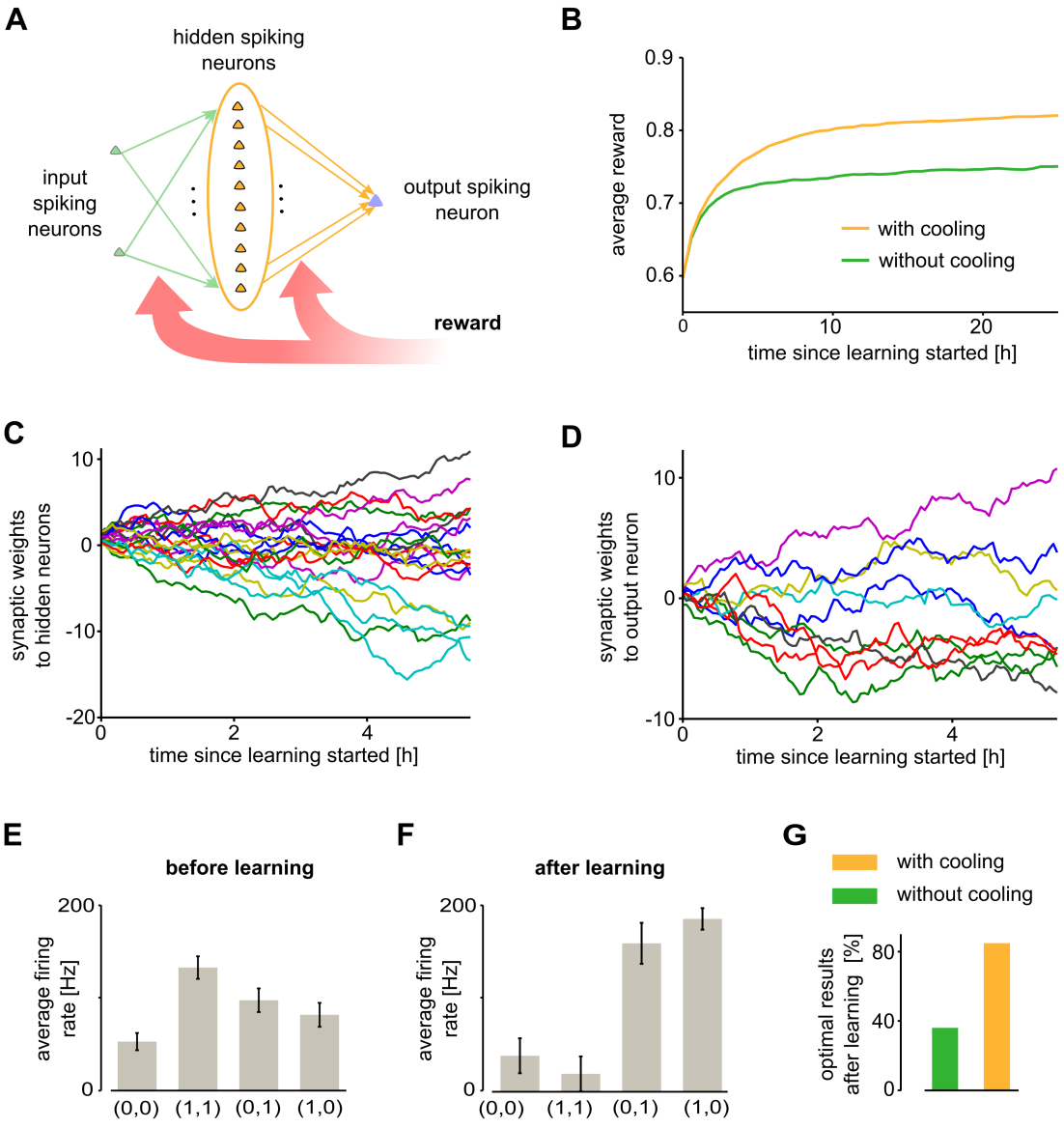

Fig. 5. **Cooling improves reward-based learning in spiking neural networks. A:** Illustration of the network architecture. The network consist of 2 input neurons, 10 hidden neurons and 1 output neuron. The task was to learn the XOR function. **B:** Comparison of the average reward obtained during learning for Hamiltonian dynamics with (orange) and without (green) cooling of the temperature T. **C,D:** Evolution of synaptic weights to neurons in the hidden layer (C) and to the output neurons (D) during the first 6 hours of learning. **E,F:** Average firing rate of the output neuron for the four input patterns before (E) and after (F) learning. **G:** Fraction of learning trials at which the network finds the optimal solution with (orange) and without (green) cooling.

0. The hidden neurons and output neuron were stochastic spiking neurons with a refractory time of 5 ms. Each layer was fully connected to the next layer and initial synaptic weights were set randomly (see Methods for details).

During learning, a pattern was chosen randomly and presented to the network for 400 ms. During this time, the output of the network was compared to the target output and a binary reward was delivered accordingly. More specifically, every 5 ms the reward was recomputed and delivered to the network – being 1 if the output neuron spiked (was silent) in the past 5 ms for a target of 1 (0 respectively) and 0 otherwise. The pattern presentation was followed by a delay period of 100 ms where no input or reward was delivered to the network. Then, another randomly chosen pattern was presented and so on.

The evolution of the synaptic weights during learning is shown in Fig. 5C,D. The weights of both layers change significantly throughout learning and contribute to learning the task. Synapses also remain plastic throughout the whole learning time and explore different solutions. Network responses before and after learning are shown in Fig. 5E and Fig. 5F. Before learning, the average firing rate of the output neuron for all the input patterns were 52.7, 132, 97.2, and 81.5 Hz respectively (see Fig. 5E). After learning for 6 hours, the output neuron has maximized the firing rate for the input patterns $(0, 1)$ and $(1, 0)$ and significantly reduced it for patterns $(0, 0)$ and $(1, 1)$ (see Fig. 5F).

This task was considered before by Seung and Xie [32, 33]. They also considered a stochastic spiking neuron model, however with zero refractory time. Further, in their model, positive or negative reward was delivered to the network every millisecond. It was noted in [33] that learning does not work reliably if positive and negative rewards are not balanced. In fact using our highly unbalanced reward schedule with rewards being either 0 or 1 , the network often does not achieve optimal performance if a constant temperature is chosen for learning (Fig. 5G). In this case, optimal results were obtained in only 40 % of the learning trials (where each trial was started with a different random initialization of the parameters). When we introduced a "cooling" schedule in which the temperature was decreased during learning, this ratio increased to 90 %. The superiority of the annealed optimization is also visible in average reward attained during learning, see Fig. 5B. This shows that parameter optimization with annealed noise can significantly improve performance of spiking neural networks. A similar observation was reported for deep artificial neural networks [34]. Our theoretical framework of Hamiltonian sampling provides an explanation for this phenomenon as an optimization through annealed sampling similar to simulated annealing and thus opens the door to apply the toolkit of stochastic optimization to gradient-based neural network learning in a principled manner.

3 Discussion

We have presented a new theoretical framework for reward-based neural network optimization that integrates a hidden synaptic parameter in the plasticity process. We suggest that this synaptic parameter could be implemented in the synapse through CaMKII, that is abundantly present in the postsynaptic density and acts as low pass filter in the induction of synaptic plasticity. We have shown that the CaMKII-enriched dynamics supports a special type of ongoing stochastic policy search – Hamiltonian sampling with friction – and convergences to the stationary distribution much faster than Langevin sampling (synaptic sampling).

David Marr famously proposed to treat brain computation at three distinct, complementary levels of analysis [35], which is today known as Marr's Tri-Level Hypothesis. It is of interest to realize that biological data on the activation dynamics of the kinase CaMKII, corresponds to the implementational/physical level of Marr's Tri-Level Hypothesis. Our proposed model for network plasticity suggests that CaMKII enables the brain to perform Hamiltonian sampling on the algorithm level (algorithmic/representational level of Marr's Tri-Level Hypothesis). To be specific, biological networks of neurons are able to approximate Hamiltonian sampling of network configurations, rather than slower Langevin sampling or gradient descent.

We have demonstrated several advantages of Hamiltonian sampling over previously considered approaches to reward-based learning in spiking neural networks. We have shown in Fig. 2 that this Hamiltonian synaptic sampling framework can be use to learn smooth responses of spiking neurons through reward-based learning, and that it further can scale up to learn recurrent networks of spiking neurons (shown in Fig. 3). In Fig. 4 we have shown that synaptic sampling is prone to be slow near saddle points of the objective function, and that Hamiltonian synaptic sampling can significantly speed up learning in these cases. Finally, we have demonstrated that reward-based network plasticity is in principle able to acquire through down-regulation of the stochastic component in parameter updates the full power of simulated annealing for optimizing the network. This allows neural networks to attain through learning not only locally optimal network configurations, but in principle even a global optimum. This theoretical result provides a new gold standard for reward-based network learning.

CaMKII dynamics has previously been studied in [36]. While this work focused on detailed molecular dynamics and its implications for STDP on the level of pairing protocols, we treated CaMKII dynamics in the current study more abstractly as a low-pass filtering process and studied the implications for system level reward-based learning. It is interesting to note that the low-pass filtering effect was also predicted in the model of [36]. In addition they proposed a role of CaMKII for binary-state behavior of synapses in hippocampus. The underlying hypothesis that synaptic efficacies can attain only two possible states, a depressed state and a potentiated state, has been put into question by recent experimental data [37].

Our model makes a number of experimentally testable predictions. It was shown in previous work that synaptic spine dynamics can be modeled by a stochastic process (Ornstein-Uhlenbeck process) with two time-constants on the temporal scale of several days. Our model that includes the Hamiltonian momentum term suggests that also on short time scales (minutes to few hours), models of synaptic dynamics with two time constants should provide better fits. Moreover, the proposed role of CaMKII suggests that these time constant should correspond to rates of dephosphorylation.

Our result on network optimization in Fig. 5 suggests that biological networks are able to control the level of stochasticity, and that stochasticity decreases during long lasting learning processes (cooling). Experimental results revealed that learning a new behavioral task is accompanied by increased synaptic spine numbers and spine dynamics [27, 38]. In [10] we analyzed a simple model for synaptic turnover and found that the statistics of spine regrowth during task acquisition can be explained by a brief increase of the learning temperature T. These findings suggest that the brain employs – in addition to deterministic synaptic updates – a mechanism to regulate the speed of random exploration in the high-dimensional space of synaptic parameters over several hours to days. This article has introduced a mathematical framework that provides a step towards understanding the complex interplay of deterministic and stochastic strategies employed by the brain, to solve complex learning problems.

4 Methods

4.1 Details to learning the stationary distribution of network configurations through synaptic plasticity rules

Here we present the general mathematical framework of synaptic parameter dynamics and derive the emerging stationary distribution of network configurations that results from this dynamics. The generalized model, that includes both Hamiltonian synaptic sampling (4) and synaptic sampling without momentum (3) as special cases, is given by the following set of SDEs:

$$\begin{aligned} d\theta_i(t) &= \left(-a\frac{\partial \log p^*(\mathbf{\Gamma})}{\partial \Gamma_i}\Big|_{\mathbf{\Gamma}(t)} + c\frac{\partial \log p^*(\boldsymbol{\theta})}{\partial \theta_i}\Big|_{\boldsymbol{\theta}(t)} \right) dt + \sqrt{2Tc}\; d\mathcal{W}_{\theta_i} \\ d\Gamma_i(t) &= \left(a\frac{\partial \log p^*(\boldsymbol{\theta})}{\partial \theta_i}\Big|_{\boldsymbol{\theta}(t)} + b\frac{\partial \log p^*(\mathbf{\Gamma})}{\partial \Gamma_i}\Big|_{\mathbf{\Gamma}(t)} \right) dt + \sqrt{2Tb}\; d\mathcal{W}_{\Gamma_i}, \end{aligned} \tag{10}$$

where $p^*(\boldsymbol{\theta})$ is the posterior distribution over the network parameter given by equation (2) and $p^*(\mathbf{\Gamma})$ is the distribution over the CaMKII-related hidden synaptic parameter. The notation, $\frac{\partial \log p^*(\mathbf{\Gamma})}{\partial \Gamma_i}\Big|_{\mathbf{\Gamma}(t)}$, denotes the derivative of $\log p^*(\mathbf{\Gamma})$ evaluated at the parameter vector $\mathbf{\Gamma}(t)$. In Results we suppressed this time-dependences in order to simplify the notation. $T > 0$ is the temperature parameter, $\mathcal{W}_{\theta_i}, \mathcal{W}_{\Gamma_i}$ are independent one-dimensional Wiener processes, and a, b, c are positive constants.

This dynamics describes a general noisy first-order interaction between visible synaptic parameters θ_i, that determine the efficacy of the synapse, and hidden synaptic parameters Γ_i, the absolute value of which model the local concentration of CaMKII in its activated state. The dynamics can thus be seen as a generalization of standard gradient-based synaptic plasticity rules (e.g. for maximum likelihood learning) that includes structural constraints, CaMKII activation and stochastic plasticity. For the general dynamics, the joint distribution over the sets of all parameters $\boldsymbol{\theta}$ and $\mathbf{\Gamma}$ will converge after a while to $p_T^*(\boldsymbol{\theta}, \mathbf{\Gamma}) = \frac{1}{Z} p^*(\boldsymbol{\theta})^{\frac{1}{T}} p^*(\mathbf{\Gamma})^{\frac{1}{T}}$ and produce samples from it. This result can be formalized in the following theorem:

Theorem 4.1. *Let $p^*(\boldsymbol{\theta})$,$p^*(\mathbf{\Gamma})$ be strictly positive, continuous probability distributions over parameters $\boldsymbol{\theta}$ and $\mathbf{\Gamma}$ respectively, twice continuously differentiable with respect to $\boldsymbol{\theta}$ and $\mathbf{\Gamma}$. Let a, b, c be postitive constants. Then the set of stochastic differential equations (10) leaves the distribution $p_T^*(\boldsymbol{\theta}, \mathbf{\Gamma}) = \frac{1}{Z} p^*(\boldsymbol{\theta})^{\frac{1}{T}} p^*(\mathbf{\Gamma})^{\frac{1}{T}}$ invariant. Furthermore, this is the unique stationary distribution of the sampling dynamics.*

Proof. Eq. (10) has two drift terms $A_i(\boldsymbol{\theta})$, $A_i(\mathbf{\Gamma})$ and two diffusion terms $B_{i,s}(\boldsymbol{\theta})$, $B_{i,s}(\mathbf{\Gamma})$:

$$A_i(\boldsymbol{\theta}) = -a\frac{\partial \log p^*(\mathbf{\Gamma})}{\partial \Gamma_i} + c\frac{\partial \log p^*(\boldsymbol{\theta})}{\partial \theta_i} \tag{11}$$

$$A_i(\mathbf{\Gamma}) = a\frac{\partial \log p^*(\boldsymbol{\theta})}{\partial \theta_i} + b\frac{\partial \log p^*(\mathbf{\Gamma})}{\partial \Gamma_i} \tag{12}$$

$$B_{i,s}(\boldsymbol{\theta}) = \begin{cases} 2Tc, i = s \\ 0,\ others \end{cases} \tag{13}$$

$$B_{i,s}\left(\mathbf{\Gamma}\right)=\begin{cases} 2Tb, i=s \\ 0,\ others \end{cases} \tag{14}$$

Hence the SDEs (10) can be translate into the following Fokker-Planck equation:

$$\begin{aligned}\frac{dp_{FP}(\mathbf{\Gamma},\boldsymbol{\theta},t)}{dt}=&\sum_i -\frac{\partial}{\partial\theta_i}\left(\left(-a\frac{\partial\log p^*(\mathbf{\Gamma})}{\partial\Gamma_i}+c\frac{\partial\log p^*(\boldsymbol{\theta})}{\partial\theta_i}\right)p_{FP}\left(\mathbf{\Gamma},\boldsymbol{\theta},t\right)\right)\\&+\sum_i -\frac{\partial}{\partial\Gamma_i}\left(\left(a\frac{\partial\log p^*(\boldsymbol{\theta})}{\partial\theta_i}+b\frac{\partial\log p^*(\mathbf{\Gamma})}{\partial\Gamma_i}\right)p_{FP}\left(\mathbf{\Gamma},\boldsymbol{\theta},t\right)\right)\\&+\sum_i\frac{\partial^2}{\partial\theta_i^2}\left(Tc\ p_{FP}\left(\mathbf{\Gamma},\boldsymbol{\theta},t\right)\right)+\sum_i\frac{\partial^2}{\partial\Gamma_i^2}\left(Tb\ p_{FP}\left(\mathbf{\Gamma},\boldsymbol{\theta},t\right)\right),\end{aligned} \tag{15}$$

where $\frac{dp_{FP}(\mathbf{\Gamma},\boldsymbol{\theta},t)}{dt}$ denotes the distribution over network parameters at time t. If we plug the stationary distribution $p_T^*\left(\boldsymbol{\theta},\mathbf{\Gamma}\right)=\frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}$ to the right side of eq. (15), we have:

$$\begin{aligned}\frac{dp_{FP}(\mathbf{\Gamma},\boldsymbol{\theta},t)}{dt}=&\sum_i -\frac{\partial}{\partial\theta_i}\left(\left(-a\frac{\partial\log p^*(\mathbf{\Gamma})}{\partial\Gamma_i}+c\frac{\partial\log p^*(\boldsymbol{\theta})}{\partial\theta_i}\right)\frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}\right)\\&+\sum_i -\frac{\partial}{\partial\Gamma_i}\left(\left(a\frac{\partial\log p^*(\boldsymbol{\theta})}{\partial\theta_i}+b\frac{\partial\log p^*(\mathbf{\Gamma})}{\partial\Gamma_i}\right)\frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}\right)\\&+\sum_i\frac{\partial^2}{\partial\theta_i^2}\left(Tc\ \frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}\right)+\sum_i\frac{\partial^2}{\partial\Gamma_i^2}\left(Tb\ \frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}\right)\\=&\sum_i -\frac{\partial}{\partial\theta_i}\left(c\frac{\partial\log p^*(\boldsymbol{\theta})}{\partial\theta_i}\frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}\right)+\sum_i\frac{\partial^2}{\partial\theta_i^2}\left(Tc\ \frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}\right)\\&+\sum_i -\frac{\partial}{\partial\Gamma_i}\left(b\frac{\partial\log p^*(\mathbf{\Gamma})}{\partial\Gamma_i}\frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}\right)\ +\sum_i\frac{\partial^2}{\partial\Gamma_i^2}\left(Tb\ \frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}\right)\\=&\sum_i\frac{\partial}{\partial\theta_i}\left(-c\frac{\partial\log p^*(\boldsymbol{\theta})}{\partial\theta_i}\frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}+Tc\frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}\frac{\partial\log p^*(\boldsymbol{\theta})^{\frac{1}{T}}}{\partial\theta_i}\right)\\&+\sum_i\frac{\partial}{\partial\Gamma_i}\left(-b\frac{\partial\log p^*(\mathbf{\Gamma})}{\partial\Gamma_i}\frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}+Tb\frac{1}{\mathcal{Z}}\left(p^*\left(\boldsymbol{\theta}\right)p^*\left(\mathbf{\Gamma}\right)\right)^{\frac{1}{T}}\frac{\partial\log p^*(\mathbf{\Gamma})^{\frac{1}{T}}}{\partial\Gamma_i}\right)\\=&\ 0\end{aligned} \tag{16}$$

This proves that $\frac{1}{\mathcal{Z}}p_T^*(\boldsymbol{\theta})^{\frac{1}{T}}p^*(\mathbf{\Gamma})^{\frac{1}{T}}$ is the stationary distribution of the parameters dynamic (10). Under the assumption that b and c are strictly positive, this stationary distribution is also unique. If the matrix of diffusion coefficients is invertible, and the potential conditions are satisfied, the stationary distribution can be obtained (uniquely) by simple integration. Since the matrix of diffusion coefficients is diagonal in our model, the diffusion coefficient matrix is trivially invertible if all diagonal elements, i.e. all b and c are strictly positive. Also the potential conditions are fulfilled (by design), as can be verified by substituting eqs. (11 – 14) into Equation (5.3.22) in [17],

$$\begin{aligned}Z_{\theta_i}\left(\boldsymbol{\theta},\mathbf{\Gamma}\right)&=B_{i,i}^{-1}\left(\boldsymbol{\theta}\right)\left(2A_i(\boldsymbol{\theta})-\frac{\partial}{\partial\theta_i}B_{i,i}(\boldsymbol{\theta})\right)\\&=\frac{1}{2Tc}\left(-2a\frac{\partial\log p^*\left(\mathbf{\Gamma}\right)}{\partial\Gamma_i}+2c\frac{\partial\log p^*\left(\boldsymbol{\theta}\right)}{\partial\theta_i}\right)\end{aligned} \tag{17}$$

$$\begin{aligned}Z_{\Gamma_i}\left(\boldsymbol{\theta},\mathbf{\Gamma}\right)&=B_{i,i}^{-1}\left(\mathbf{\Gamma}\right)\left(2A_i(\mathbf{\Gamma})-\frac{\partial}{\partial\Gamma_i}B_{i,i}(\mathbf{\Gamma})\right)\\&=\frac{1}{2Tb}\left(a\frac{\partial\log p^*\left(\boldsymbol{\theta}\right)}{\partial\theta_i}+b\frac{\partial\log p^*\left(\mathbf{\Gamma}\right)}{\partial\Gamma_i}\right)\end{aligned} \tag{18}$$

This shows that $Z(\boldsymbol{\theta}, \boldsymbol{\Gamma}) = (Z_{\theta_i}(\boldsymbol{\theta}, \boldsymbol{\Gamma}), Z_{\Gamma_i}(\boldsymbol{\theta}, \boldsymbol{\Gamma}))$ is a gradient. Thus, the potential conditions are met and the stationary distribution is unique.

In Theorem 4.1, b,c need to be strictly positive. Note that we can relax it to b or c is strictly positive (or both) – which means there exists diffusion noise – and can prove $\frac{1}{\mathcal{Z}} p_T^*(\boldsymbol{\theta})^{\frac{1}{T}} p^*(\boldsymbol{\Gamma})^{\frac{1}{T}}$ is a unique stationary distribution of stochastic differential equations (10) in the same way.

Hamiltonian synaptic sampling (4) and synaptic sampling (3) are special cases of the more general parameter dynamics (10). Hamiltonian synaptic sampling as defined in (4) is obtained by choosing $c = 0$ and a Gaussian distribution for the hidden parameters $p^*(\boldsymbol{\Gamma}) \sim NORMAL(0, 1)$. Synaptic sampling as defined in (3) is obtained by choosing $a = b = 0$. We remark that various types of gradient descent can also be recovered from the generalized dynamics for $T = 0$, e.g. gradient descent with momentum for the noiseless Hamiltonian dynamics. Equation (10) can be seen as the continuous version of Hamiltonian sampling [22], where a Metropolis update is performed after simulating Hamiltonian dynamic. Equation (10) can also be seen as an extension of stochastic gradient Hamiltonian Monte Carlo with friction [39, 40] to the case where the temperature T is used to shape the static distribution $p_T^*(\boldsymbol{\theta})$.

4.2 Spiking neuron model

Spiking neurons were modeled by a stochastic variant of the spike response model [41]. We use $w_i(t)$ to donate the synaptic efficacy of the i-th synapse in the network at time t. Each neuron z_k of network $\mathcal{N}$ is then modeled as a point neuron with membrane potential $u_k(t)$ at time t

$$u_k(t) \;=\; \sum_{i \in \mathrm{SYN}_k} y_{\mathrm{PRE}_i}(t)\, w_i(t) \;+\; \varphi_k(t)\;, \tag{19}$$

where SYN_k is the index set of synapses that project to neuron z_k, PRE_i denote the index of the presynaptic neuron of synapse i, $\varphi_k(t)$ denotes the bias potential of neuron z_k. In the recurrent network in Fig. 3 we used a slowly changing bias potential to ensures that the output rate of each neuron stays within finite bounds (described in detail below). In all other experiments we used a constant bias potential. $y_{\mathrm{PRE}_i}(t)$ denote the trace of the (unweighted) postsynaptic potentials (PSPs) from presynaptic neuron of synapse i at time t. Throughout this paper, we used standard double-exponential PSP kernels with a brief finite rise and exponential decay, of the form $\epsilon(t) = \frac{\tau_r}{\tau_m - \tau_r} \left(e^{-\frac{t}{\tau_m}} - e^{-\frac{t}{\tau_r}} \right)$, with time constants τ_m and τ_r (any other PSP shape may be used in principle without further adaptations of the theoretical model).

We denote the output spike train of neuron z_k by $z_k(t)$, which is defined as a sum of Dirac delta pulses positioned at the spike times $t_k^{(1)}, t_k^{(2)}, \ldots$, i.e., $z_k(t) = \sum_l \delta(t - t_k^{(l)})$. Neuron fires according to the link function $f_k(t)$ which denotes the firing probability of neuron k at time t. Due to the lasting effects of PSPs, the firing probability may depend on the history of past spiking activities of all K input neurons up to time t which we denote by $\mathbf{x}(t) = \{x_i(\tau) \mid 1 \leq i \leq K, 0 \leq \tau < t\}$, which is defined as:

$$p_{\mathcal{N}}\left(z_k(t) = 1 \,|\, \mathbf{x}(t), \boldsymbol{\theta}\right) = \; f_k(t) \;=\; f(u_k(t), \rho_k(t))\;, \tag{20}$$

where $\rho_k(t)$ denotes a refractory variable that is given by the time elapsed since the last spike of neuron z_k. In this article, we set $f(u_k, \rho_k) = \sigma(u_k)\Theta(\rho_k - t_{ref})$, where $\sigma(u_k)$ is a sigmoid activation function $\sigma(u_k) = \frac{1}{1+e^{-u_k}}$ and $\Theta(\cdot)$ denotes the Heaviside step function, i.e. $\Theta(x) = 1$ for $x \geq 0$ and 0 otherwise. In our simulation, we set refectory time t_{ref} to 5 ms.

4.3 Reward-modulated synaptic plasticity rule

Here we derive the reward-based learning rules for the spiking neural network model outline above. In particular, we compute here the gradient of the expected reward:

$$\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta}) \;=\; \frac{\partial}{\partial \theta_i} \log \left\langle \int_0^\infty e^{-\frac{\tau}{\tau_e}}\, r(\tau)\, d\tau \right\rangle_{p(\mathbf{r}|\boldsymbol{\theta})} . \tag{21}$$

We only consider the recurrent network in section Reward-guided network plasticity as an example and show that the parameter dynamic (6), (5) approximate this gradient. Actually one can compute the gradient of the expected reward for the feed-forward neural network in other simulations and get similar learning rule. In order to simplify notation, we use $\mathbf{z}(t)$ to represent the history of past spiking activity of all neurons $z_k (1 \le k \le K)$ up to time t. Supposing that the reward signal $r(\tau)$ is only decided by $\mathbf{z}(t)$, we can rewrite $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ as the expectation over all possible spike trains $\mathbf{z}(t)$ up to time t:

$$\begin{aligned} \frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta}) &= \frac{1}{\mathcal{V}(\boldsymbol{\theta})} \left\langle \int_0^\infty e^{-\frac{\tau}{\tau_e}}\, r(\tau)\, \frac{\partial}{\partial \theta_i} \log p\left(r(\tau), \mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right)\, d\tau \right\rangle_{p(\mathbf{r},\mathbf{z}|\boldsymbol{\theta})} \\ &= \left\langle \int_0^\infty e^{-\frac{\tau}{\tau_e}}\, \frac{r(\tau)}{\mathcal{V}(\boldsymbol{\theta})}\, \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right)\, d\tau \right\rangle_{p(\mathbf{r},\mathbf{z}|\boldsymbol{\theta})} . \end{aligned} \tag{22}$$

Note that we use the fact $\log p\left(r(\tau), \mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right) = \log p\left(r(\tau) \,|\, \mathbf{z}(\tau)\right) + \log p\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right)$. The problem now is to estimate the gradient of the probability of observing the spike train $\mathbf{z}(\tau)$ in the time interval 0 to τ. According to Eqs. (19) and (20), the logarithm of the probability distribution $p_{\mathcal{N}}\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right)$ can be rewritten as:

$$\log p_{\mathcal{N}}\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right) \;=\; \int_0^t \Big(z_{\text{POST}_i}(s)\, \log f_{\text{POST}_i}(s) \;-\; (1 - z_{\text{POST}_i}(s))\, \log(1 - f_{\text{POST}_i}(s)) \Big) ds \tag{23}$$

where the integration runs from time 0 to t. Using this, the gradient $\frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right)$ can be estimated

$$\begin{aligned} \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}\left(\mathbf{z}(\tau) \,|\, \boldsymbol{\theta}\right) &= \int_0^\tau \frac{\partial w_i}{\partial \theta_i} \frac{\partial}{\partial w_i} \Big(z_{\text{POST}_i}(s)\, \log f_{\text{POST}_i}(s) \;-\; (1 - z_{\text{POST}_i}(s))\, \log(1 - f_{\text{POST}_i}(s)) \Big)\, ds \\ &= \int_0^\tau w_i\, y_{\text{PRE}_i}(s)\, (z_{\text{POST}_i}(s) - f_{\text{POST}_i}(s))\, ds . \end{aligned} \tag{24}$$

The dependence on w_i (the current value of the synaptic weight), is a result of applying the chain rule and using the exponential mapping function (7). If a linear mapping between θ_i and w_i is used this term vanishes as in Eq. (6). The learning rules are similar to previous ones which were found in the context of maximum likelihood and reinforcement learning in neural networks [42, 24].

Eq. (22) defines a batch learning rule with an average taken over learning episodes where in each episode network responses and rewards are drawn according to the distribution $p(\mathbf{r}, \mathbf{z}|\boldsymbol{\theta})$. In order to arrive at an online learning rule for this scenario, we consider an estimator of Eq. (22) that approximates its value at each time $t > \tau_g$ based on the recent network activity and rewards during time $[t - \tau_g, t]$ for some suitable $\tau_g > 0$. We denote the estimator at time t by $G_i(t)$ where

we want $G_i(t) \approx \frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ for all $t > \tau_g$. To arrive at such an estimator, we approximate the average over episodes in Eq. (22) by an average over time where each time point is treated as the start of an episode. The average is taken over a long sequence of network activity that starts at time t and ends at time $t+\tau_g$. Here, one systematic difference to the batch setup is that one cannot guarantee a time-invariant distribution over initial network conditions as we did there since those will depend on the current network parameter setting. However, under the assumption that the influence of initial conditions (such as initial membrane potentials and refractory states) decays quickly compared to the time scale of the environmental dynamics, it is reasonable to assume that the induced error is negligible. We thus rewrite Eq. (22) in the form

$$\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta}) \;\approx\; G_i(t) = \frac{1}{\tau_g} \int_t^{t+\tau_g} \int_\zeta^{t+\tau_g} \; e^{-\frac{\tau-\zeta}{\tau_e}} \; \frac{r(\tau)}{\mathcal{V}(\boldsymbol{\theta})} \; \int_\zeta^{\tau} \; w_i(s)\, y_{\text{PRE}_i}(s)\, (z_{\text{POST}_i}(s) - f_{\text{POST}_i}(s)))\; ds\; d\tau\; d\zeta \;,$$

where τ_g is the length of the sequence of network activity over which the empirical expectation is taken. Finally, we can combine the second and third integral into a single one, rearrange terms and substitute s and τ so that integrals run into the past rather than the future, to obtain

$$G_i(t) \;\approx\; \frac{1}{\tau_g} \int_{t-\tau_g}^{t} \; \frac{r(\tau)}{\mathcal{V}(\boldsymbol{\theta})} \; \int_0^{\tau} \; e^{-\frac{s}{\tau_e}}\, w_i(\tau - s)\, y_{\text{PRE}_i}(\tau - s)\, (z_{\text{POST}_i}(\tau - s) \;-\; f_{\text{POST}_i}(\tau - s))\; ds\; d\tau \;. \quad (25)$$

Supposing that τ_g tends to 0, we get a simple on-line learning rule to approximate $G_i(t)$:

$$G_i(t) \approx r(t)\, e_i(t) \;. \quad (26)$$

$$\frac{de_i(t)}{dt} = \Big(-\tfrac{1}{\tau_e} \;\; e_i(t) \;+\; w_i(t)\, y_{\text{PRE}_i}(t)\, (z_{\text{POST}_i}(t) - f_{\text{POST}_i}(t))\Big) \quad (27)$$

A similar learning rule has already been proposed by Seung and Xie [33]. In fact, as the learning rule only estimate Eq. (25) based on the reward at time t, it ignores outer integral in Eq. (25) and thus can't approximate $G_i(t)$ accurately. A better estimation has been given by Kappel et al. [10] to improve learning performance, but without biological plausible motivation. Actually in our Hamiltonian synaptic sampling framework, CaMKII works as a momentum term that computes the average of the gradient $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ instead of current gradient during on-line learning, which corresponds to the outer integral and thereby supporting better estimate of the gradient of expected reward.

4.4 Relating Hamiltonian synaptic sampling to synaptic sampling

Here we build the relationship between Hamiltonian synaptic sampling and synaptic sampling and show that synaptic sampling is included in Hamiltonian synaptic sampling. For simplicity and brevity, here we consider a version of the parameters dynamics for discrete time. According to Eq. (3), the parameter change $\Delta\theta_i^{syn}$ of synaptic sampling during a small discrete time step Δt can be written as:

$$\Delta\theta_i^{syn} \;=\; \beta\, \Delta t\, \frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta}) \;+\; \sqrt{2T\beta\Delta t}\; v_i^t \;, \quad (28)$$

where $\beta > 0$ denotes a learning rate that controls the speed of the parameter dynamics.v_i^t represents Gaussian noise with zero mean and variance 1. These noises are independent for each parameter θ_i and each update time t.

To compare synaptic sampling with Hamiltonian synaptic sampling, we rewrite Eq. (4) into the discrete version with the same time step Δt:

$$\begin{aligned} \Delta\theta_i^{ham} &= a\ \Delta t\ \Gamma_i(t+\Delta t), \\ \Gamma_i(t+\Delta t) &= (1-b\ \Delta t)\ \Gamma_i(t)\ +\ b\ \Delta t\left(\tfrac{a}{b}\tfrac{\partial}{\partial\theta_i}\log p^*(\boldsymbol{\theta})\ +\ \sqrt{\tfrac{2T}{b\Delta t}}\ v_i^t\right). \end{aligned} \tag{29}$$

Eq. (29) seems different from Eq. (28). Actually, we can build the relationship between Eq. (29) and (28) with the assumption that the momentum term Γ_i has transient time constant (tends to zeros). To be specific, the parameter b is very large and $b\Delta t$ tends to be 1. We thus rewrite the discrete version of Hamiltonian synaptic sampling (29) as:

$$\Delta\theta_i^{ham} = \frac{a^2}{b}\ \Delta t\frac{\partial}{\partial\theta_i}\ \log p^*(\boldsymbol{\theta})\ +\ \sqrt{\frac{2Ta^2\Delta t}{b}}\ v_i^t\ . \tag{30}$$

Note that Eq. (30) and (28) are the same if $\frac{a^2}{b} = \beta$ holds. Hence we conclude that synaptic sampling is a special case of Hamiltonian synaptic sampling that the momentum term changes on transient time constant.

4.5 Global network optimization through stochastic synaptic plasticity

Here, we show that in principle, stochastic plasticity with a cooling schedule (i.e., with a slow decrease of the noise amplitude) can produce globally optimal network configurations.

Temperature-dependent expected reward: We first calculate the expected reward that is attained by a network with parameters that have converged to the stationary distribution $p_T^*(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}}p^*(\boldsymbol{\theta}|R=1)^{\frac{1}{T}}$ at temperature T. We denote by R_T the Bernoulli random variable that indicates reward at temperature T. When the network has reached the stationary distribution $p_T^*(\boldsymbol{\theta})$, the expected reward $E[R_T]$ is given by

$$\begin{aligned} E[R_T] &= \sum_{r\in\{0,1\}} r\ p_{\mathcal{N}}(R_T=r) = p_{\mathcal{N}}(R_T=1) = \\ &= \int p_{\mathcal{N}}\left(R=1\,|\,\boldsymbol{\theta}\right)p_T^*(\boldsymbol{\theta})d\boldsymbol{\theta} = \frac{1}{\mathcal{Z}}\int p_{\mathcal{N}}\left(R=1\,|\,\boldsymbol{\theta}\right)p^*(\boldsymbol{\theta}|R=1)^{\frac{1}{T}}d\boldsymbol{\theta} \\ &= \frac{1}{\mathcal{Z}}\int p_{\mathcal{N}}\left(R=1\,|\,\boldsymbol{\theta}\right)\frac{p_{\mathcal{N}}\left(R=1\,|\,\boldsymbol{\theta}\right)^{\frac{1}{T}}p_{\mathcal{S}}\left(\boldsymbol{\theta}\right)^{\frac{1}{T}}}{p_{\mathcal{N}}(R=1)^{\frac{1}{T}}}d\boldsymbol{\theta}. \end{aligned}$$

Assuming an uninformative prior $p_{\mathcal{S}}\left(\boldsymbol{\theta}\right)$, we obtain

$$E[R_T] = \frac{1}{\mathcal{Z}_T}\int p_{\mathcal{N}}\left(R=1\,|\,\boldsymbol{\theta}\right)p_{\mathcal{N}}\left(R=1\,|\,\boldsymbol{\theta}\right)^{\frac{1}{T}}d\boldsymbol{\theta} = \frac{1}{\mathcal{Z}_T}\int p_{\mathcal{N}}\left(R=1\,|\,\boldsymbol{\theta}\right)^{1+\frac{1}{T}}d\boldsymbol{\theta},$$

where $\mathcal{Z}_T = \int p_{\mathcal{N}}\left(R=1\,|\,\boldsymbol{\theta}\right)^{\frac{1}{T}}d\boldsymbol{\theta}$ normalizes $p_{\mathcal{N}}\left(R=1\,|\,\boldsymbol{\theta}\right)^{\frac{1}{T}}$ with respect to $\boldsymbol{\theta}$. In other words, sampling from the posterior $p^*(\boldsymbol{\theta})$ with decreased temperature $T<1$ concentrates parameter samples at values that lead to large rewards and therefore increases the expected reward of the network.

Temperature annealing for global optimization: For small temperatures, the posterior is concentrated at the global optimum of the reward landscape. In practice, the sampling process

mixes extremely slowly at low temperatures due to low probabilities for non-optimal states. Hence, an annealing schedule that decreases the temperature slowly over time has to be employed in order to give synaptic sampling enough time to settle at the global optimum, similar to the simulated annealing optimization algorithm.

Here we show that in the limit $T \to 0$, the network achieves maximal possible performance (the derivation is similar to the one in [15] for simulated annealing). Let $\mathcal{S}_{\text{opt}}$ denote the set of optimal circuit parameters, i.e., $\mathcal{S}_{\text{opt}} = \{\boldsymbol{\theta} \,|\, p_{\mathcal{N}}(R=1 \,|\, \boldsymbol{\theta}) = R_{\max}\}$ for $R_{\max} \equiv \max_{\boldsymbol{\theta}}\{p_{\mathcal{N}}(R=1 \,|\, \boldsymbol{\theta})\}$. For an uninformative prior we have

$$\lim_{T\to 0} p_T^*(\boldsymbol{\theta}) = \lim_{T\to 0} \frac{p_{\mathcal{N}}(R=1 \,|\, \boldsymbol{\theta})^{\frac{1}{T}}}{\mathcal{Z}_T} = \lim_{T\to 0} \frac{p_{\mathcal{N}}(R=1 \,|\, \boldsymbol{\theta})^{\frac{1}{T}}}{\int p_{\mathcal{N}}(R=1 \,|\, \boldsymbol{\theta})^{\frac{1}{T}}\, d\boldsymbol{\theta}} \tag{31}$$

$$= \lim_{T\to 0} \frac{R_{\max}^{-\frac{1}{T}} p_{\mathcal{N}}(R=1 \,|\, \boldsymbol{\theta})^{\frac{1}{T}}}{\int R_{\max}^{-\frac{1}{T}} p_{\mathcal{N}}(R=1 \,|\, \boldsymbol{\theta})^{\frac{1}{T}}\, d\boldsymbol{\theta}} \tag{32}$$

$$= \lim_{T\to 0} \frac{\left(\frac{p_{\mathcal{N}}(R=1 \,|\, \boldsymbol{\theta})}{R_{\max}}\right)^{\frac{1}{T}}}{\int \left(\frac{p_{\mathcal{N}}(R=1 \,|\, \boldsymbol{\theta})}{R_{\max}}\right)^{\frac{1}{T}} d\boldsymbol{\theta}}. \tag{33}$$

This evaluates to

$$\lim_{T\to 0} p_T^*(\boldsymbol{\theta}) = \lim_{T\to 0} \frac{p_{\mathcal{N}}(R=1 \,|\, \boldsymbol{\theta})^{\frac{1}{T}}}{\mathcal{Z}_T} = \begin{cases} \frac{1}{\int_{\boldsymbol{\theta}\in\mathcal{S}_{\text{opt}}} d\boldsymbol{\theta}} = \frac{1}{|\mathcal{S}_{\text{opt}}|} & , \quad \text{for } \boldsymbol{\theta} \in \mathcal{S}_{\text{opt}} \\ 0 & , \quad \text{for } \boldsymbol{\theta} \notin \mathcal{S}_{\text{opt}} \end{cases}, \tag{34}$$

where we have defined $|\mathcal{S}_{\text{opt}}| \equiv \int_{\boldsymbol{\theta}\in\mathcal{S}_{\text{opt}}} d\boldsymbol{\theta}$. Hence, in the limit $T \to 0$, the distribution is a uniform distribution over optimal parameter values. For the expected reward, we thus have

$$\lim_{T\to 0} E[R_T] = \int_{\boldsymbol{\theta}\in\mathcal{S}_{\text{opt}}} R_{\max} \frac{1}{|\mathcal{S}_{\text{opt}}|} d\boldsymbol{\theta} = R_{\max}. \tag{35}$$

4.6 Simulation details and parameters

Details to: A spiking neuron learns to emulate a sigmoidal neuron (Fig. 2). The network architecture is shown in Fig. 2A. The sigmoidal neuron receives inputs from 4 input numbers with the pre-defined weights $4, 3, -3, -6$ and basis 1. The spiking neuron receives inputs from 80 Poisson spiking neurons, which are divided into 4 pools. Each pool of 20 neurons encodes the same input number. In order to generate the input patters, we first generate 2000 random vectors with dimension 4 by sampling from a uniform distribution on $[0, 1]$, and then choose 20 vectors as input. Fig. 2B shows how the 20 examples are distributed along the input (i.e., the weighted sum of the four inputs) - output plane of a sigmoidal neuron. For each node, the X-coordinate represents the weighted sum of the four input numbers and the Y-coordinate represents the output of the sigmoidal neuron. The mapping between x and y is defined as $y = \frac{1}{1+e^{-x}}$ for this sigmoidal neuron.

At each presentation, one out of the 20 input patterns is chosen randomly as input, which is converted to the Poisson spike trains with space-rate coding. The maximum firing rate of each neuron is set to 60 HZ. After presenting the example for 300 ms, a reward is delivered to the

network for 10 ms. The reward amplitude is given by computing the difference between the output of the sigmoidal neuron and scaled firing rate of spiking neuron, that is, $r = 1 - |f_1 - f_2|$, where f_1 denotes the firing probability of the sigmoidal neuron , f_2 denotes the scaled firing rate (average firing probability) of the spiking neuron during the 300 ms time window of pattern presentation. Note that f_2 equals to the rate of the total firing times of the spiking neuron during the 300 time window to the maximum firing times 60 (as the refectory time is 5 ms). The time constants for the eligibility trace and the momentum are set to 0.2 s and 10 s respectively.

Details to: Reward-guided network plasticity (Fig. 3). The network connectivity between excitatory and inhibitory neurons was as suggested in [28]. Excitatory and Inhibitory neurons were randomly connected with connection probabilities given as in Table 2 of [28]. Connections include lateral inhibition between excitatory and inhibitory neurons. The connectivity to and from inhibitory neurons was kept fixed (not subject to learning). The connection probability from excitatory to inhibitory neurons was given by 0.575. The synaptic weights were drawn from a Gaussian distribution (truncated at zero) with $\mu = 1$ and $\sigma = 0.1$. Inhibitory neurons were connected to their targets with probability 0.6 (to excitatory neurons) and 0.55 (to inhibitory neurons) and the synaptic weights were drawn from a truncated Gaussian distribution with $\mu = -2$ and $\sigma = 0.2$. The connectivity from and to inhibitory neurons is kept fixed throughout the simulation. Plastic synaptic connections were allowed between all pairs of input and excitatory hidden neurons and among excitatory hidden neurons. The number of potential excitatory synaptic connections between each pair of neurons was drawn from a Binomial distribution ($p = 0.5$, $n = 10$).

The refractory period was 5 ms for excitatory and 2 ms for inhibitory neurons. For the post-synaptic potentials $y_i(t)$ of excitatory neurons, we used time constants $\tau_m = 20$ ms and $\tau_r = 2$ ms. For inhibitory synapses we used a faster kernel of the same form with $\tau_m = 10$ ms and $\tau_r = 1$ ms. The bias potential $\vartheta_k(t)$ in Eq. (19) was initialized at -3 and then followed the dynamics

$$\tau_\vartheta \frac{d\vartheta_k(t)}{dt} \;=\; \nu_0 - z_k(t) \;, \tag{36}$$

where $\tau_\vartheta = 50$ s is the time constant of the adaptation mechanism and $\nu_0 = 5$ Hz is the desired output rate of the neuron. Eq. (36) is a simplified version of the mechanism proposed in [43] to balance activity in networks of excitatory and inhibitory neurons. We found that this regularization significantly increased the performance and learning speed of our network model.

For the synaptic dynamics we used a Gaussian prior $p_{\mathcal{S}}(\boldsymbol{\theta})$ with $\mu = 0$ and $\sigma = 2$ and synaptic parameters were initially drawn form a Gaussian distribution with $\mu = -0.5$ and $\sigma = 0.5$. Synaptic parameter changes were clipped at ± 40 and synaptic parameters were not allowed to exceed $[-2, 5]$ for the sake of numerical stability. The weights of synapses for which the synaptic parameters $\theta_i(t)$ became smaller than zero, where clamped to $w_i(t) = 0$ as in our previous model [10]. The temperature parameter T was kept here constant at $T = 0.1$. The time constants for the eligibility trace and the momentum were set here to 1 s and 50 s, respectively.

To generate the cue pattern we adapted the method from [9]. The afferent inputs were given here by representations of a simple symbolic sensory environment. Inputs were randomly generated realizations of inhomogeneous Poisson spike trains. To generate these spike patterns, each of the 200 input neurons was assigned to a Gaussian tuning curve with $\sigma = 0.2$. Tuning curve centers were independently and equally scattered over the unit cube. The cue was represented by a

randomly selected point in this 3-dimensional space which is covered by the tuning curves of the input neurons. The stimulus positions was overlaid with small-amplitude jitter ($\sigma = 0.05$). For each presentation of the cue the firing rate of an individual input neuron was given by the support of sensory experience under the input neuron's tuning curve. The maximum firing rate of each input neuron was 60 Hz. In addition an offset of 2 Hz background noise was added. If no cue was present, all input neurons were set to homogeneous 2 Hz Poisson firing.

The preferred direction for the population vector decoding of each neuron was drawn independently for the X- and Y-axis in cursor space from a uniform distribution in the interval ± 0.025. Cursor movement was implemented using a simple version of the population vector method [29]. Each spike of a neuron in C caused the cursor to jump into the direction of the neuron's preferred direction (summed direction vectors were applied if multiple neurons fired within one millisecond). At the end of each trial the cursor position was reset to the start location at $(0.25, 0.25)$.

Details to: Hamiltonian dynamic improves network behavior at saddle points (Fig. 4). A three-layer perceptron network consists of 784 input neurons, 30 hidden neurons and 10 output neurons are used to learn the MNIST data set. As shown in Fig. 4A, the activation function of the hidden layer and out layer are sigmoid function and Winner-Take-All (WTA) respectively. In each trial, one digital is chosen randomly from the MNIST data set as input. As the network gets immediate reward, no eligibility trace is used here. To be specific, the gradients of the expected reward $\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta})$ for synapse i are $r\, y_i\, (z_i - f_i(u))$ if connected to neurons in the hidden layer and $r\, y_i\, (z_i - g_i(u))$ otherwise, where r is the binary reward, y_i and z_i denotes the outputs of presynaptic and postsynaptic neurons of synapse i. u is the weighted sum of inputs to postsynaptic neuron of synapse i. In our simulation, we set the same learning rate 0.02 for both Hamiltonian sampling and synaptic sampling. The other parameters a, c and β is chosen to be 2, 2 and 0.2. In order to test whether Hamiltonian dynamic can help to overcome saddle point problem, we first train the network with synaptic sampling and then continue to train it with Hamiltonian sampling (see Fig. 4B) with the current parameter setting. Note that the initial value of the momentum term is 0. The result shows the network escaped from the current saddle point much faster with Hamiltonian dynamics.

Details to: From reward-based learning to global network optimization (Fig. 5). A three-layer perceptron network consists of 784 input neurons, 30 hidden neurons and 10 output neurons are used to learn the MNIST data set. As shown in Fig. 4A, the activation function of the hidden layer and out layer are sigmoid function and soft Winner-Take-All (WTA) respectively. In each trial, one digit is chosen randomly from the MNIST data set as input. As the network gets immediate reward, no eligibility trace is used here. To be specific, the estimator of the gradients of the expected reward was directly given according to Eq. 6, by

$$\frac{\partial}{\partial \theta_i} \log \mathcal{V}(\boldsymbol{\theta}) \;\approx\; r(t)\, y_{\text{PRE}_i}(t)\, (z_{\text{POST}_i}(t) - f_{\text{POST}_i}(t)) \tag{37}$$

In our simulation, we used the same learning rate 0.02 for both Hamiltonian sampling and synaptic sampling. The other parameters a, c and β were chosen to be 2, 2 and 0.2. In Fig 5B we first trained the network with synaptic sampling for 10^6 trials. We then continued training with Hamiltonian sampling using the parameter setting at that time point (initially the momentum term was 0).

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/20072013) under EU grants #604102 and #720270 (Human Brain Project).

References

[1] Sheng M and Kim E. The postsynaptic organization of synapses. *Cold Spring Harbor perspectives in biology*, 3(12):a005678, 2011.

[2] Alberts B, Johnson A, Lewis J, Morgan D, Raff M, Roberts K, and Walter P. *Molecular Biology of the Cell, 6th edition*. Garland Science, 2014.

[3] Lisman J, Yasuda R, and Raghavachari S. Mechanisms of CaMKII action in long-term potentiation. *Nature Reviews Neuroscience*, 13(3):169–182, 2012.

[4] Yagishita S, Hayashi-Takagi A, Ellis-Davies GC, Urakubo H, Ishii S, and Kasai H. A critical time window for dopamine actions on the structural plasticity of dendritic spines. *Science*, 345(6204):1616–1620, 2014.

[5] Coultrap SJ, Freund RK, O'Leary H, Sanderson JL, Roche KW, DellAcqua ML, and Bayer KU. Autonomous CaMKIImediates both LTP and LTD using a mechanism for differential substrate site selection. *Cell Reports*, 6(3):431–437, 2014.

[6] Connor SA and Wang YT. A place at the table: LTD as a mediator of memory genesis. *The Neuroscientist*, pages 1–13, 2015.

[7] Yasumatsu N, Matsuzaki M, Miyazaki T, Noguchi J, and Kasai H. Principles of long-term dynamics of dendritic spines. *The Journal of Neuroscience*, 28(50):13592–13608, 2008.

[8] Holtmaat A and Svoboda K. Experience-dependent structural synaptic plasticity in the mammalian brain. *Nature Reviews Neuroscience*, 10(9):647–658, 2009.

[9] Kappel D, Habenschuss S, Legenstein R, and Maass W. Network plasticity as Bayesian inference. *PLoS Comput Biol*, 11(11):e1004485, 2015.

[10] Kappel D, Legenstein R, Habenschuss S, Hsieh M, and Maass W. Reward-based self-configuration of neural circuits. *in preparation*.

[11] Dauphin YN, Pascanu Y, Gulcehre C, Cho K, Ganguli S, and Bengio Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems*, pages 2933–2941, 2014.

[12] Dauphin YN, De Vries H, and Bengio Y. Equilibrated adaptive learning rates for non-convex optimization. In *Advances in Neural Information Processing Systems*, pages 1504–1512, 2015.

[13] Sutskever I, Martens J, Dahl GE, and Hinton GE. On the importance of initialization and momentum in deep learning. *ICML (3)*, 28:1139–1147, 2013.

[14] Wikimedia Commons. Activation and autoregulation of Calmodulin, SVG redrawn from raster image: CaMKII diagram by team L&M, 2011. File: `Psy161ST_redrawn.svg`, URL: `https://commons.wikimedia.org/wiki/File:Psy161ST_redrawn.svg`, downloaded on: September 27, 2016.

[15] Aarts E and Korst J. *Simulated annealing and Boltzmann machines*. New York, NY; John Wiley and Sons Inc., 1988.

[16] Dekkers A and Aarts E. Global optimization and simulated annealing. *Mathematical programming*, 50(1-3):367–393, 1991.

[17] Gardiner CW. *Handbook of stochastic methods, 3rd edition*. Springer, 2014.

[18] Lee SJR, Escobedo-Lozoya Y, Szatmari EM, and Yasuda R. Activation of CaMKII in single dendritic spines during long-term potentiation. *Nature*, 458(7236):299–304, 2009.

[19] Li L, Stefan MI, and Le Novère N. Calcium input frequency, duration and amplitude differentially modulate the relative activation of calcineurin and CaMKII. *PloS one*, 7(9):e43810, 2012.

[20] Bhattacharyya M, Stratton MM, Going CC, McSpadden ED, Huang Y, Susa AC, Elleman A, Cao YM, Pappireddi N, Burkhardt P, et al. Molecular mechanism of activation-triggered subunit exchange in Ca2+/calmodulin-dependent protein kinase II. *eLife*, 5:e13405, 2016.

[21] Zeng S and Holmes WR. The effect of noise on CaMKII activation in a dendritic spine during LTP induction. *Journal of neurophysiology*, 103(4):1798–1808, 2010.

[22] Neal RM. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162, 2011.

[23] Urbanczik R and Senn W. Reinforcement learning in populations of spiking neurons. *Nature Neuroscience*, 12(3):250–252, 2009.

[24] Brea J, Senn W, and Pfister JP. Matching recall and storage in sequence learning with spiking neural networks. *The Journal of Neuroscience*, 33(23):9565–9575, 2013.

[25] Abbott LF, DePasquale B, and Memmesheimer RM. Building functional networks of spiking model neurons. *Nature Neuroscience*, 19(3):350–355, 2016.

[26] Hoerzer GM, Legenstein R, and Maass W. Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning. *Cerebral Cortex*, 24(3):677–690, 2014.

[27] Peters AJ, Chen SX, and Komiyama T. Emergence of reproducible spatiotemporal activity during motor learning. *Nature*, 510(7504):263–267, 2014.

[28] Avermann M, Tomm C, Mateo C, Gerstner W, and Petersen CC. Microcircuits of excitatory and inhibitory neurons in layer 2/3 of mouse barrel cortex. *Journal of Neurophysiology*, 107(11):3116–3134, 2012.

[29] Apostolos P Georgopoulos, Andrew B Schwartz, Ronald E Kettner, et al. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

[30] Loewenstein Y, Kuras A, and Rumpel S. Multiplicative dynamics underlie the emergence of the log-normal distribution of spine sizes in the neocortex in vivo. *J. Neurosci.*, 31(26):9481–9488, 2011.

[31] Gopnik A, Griffiths TL, and Lucas CG. When younger learners can be better (or at least more open-minded) than older ones. *Current Directions in Psychological Science*, 24(2):87–92, 2015.

[32] Seung HS. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, 40(6):1063–1073, 2003.

[33] Xie X and Seung HS. Learning in neural networks by reinforcement of irregular spiking. *Physical Review E*, 69(4):041909, 2004.

[34] Neelakantan A, Vilnis L, Le QV, Sutskever I, Kaiser L, Kurach K, and Martens J. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

[35] Marr D and Poggio T. From understanding computation to understanding neural circuitry. 1976.

[36] Graupner M and Brunel N. STDP in a bistable synapse model based on CaMKII and associated signaling pathways. *PLoS Computational Biology*, 3(11):2299–2323, 2007.

[37] Thomas M Bartol Jr, Cailey Bromer, Justin Kinney, Michael A Chirillo, Jennifer N Bourne, Kristen M Harris, and Terrence J Sejnowski. Nanoconnectomic upper bound on the variability of synaptic plasticity. *Elife*, 4:e10778, 2015.

[38] Xu T, Yu X, Perlik AJ, Tobin WF, Zweig JA, Tennant K, Jones T, and Zuo Y. Rapid formation and selective stabilization of synapses for enduring motor memories. *Nature*, 462(7275):915–919, 2009.

[39] Chen T, Fox EB, and Guestrin C. Stochastic gradient hamiltonian monte carlo. In *ICML*, pages 1683–1691, 2014.

[40] Ma YA, Chen T, and Fox E. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, pages 2899–2907, 2015.

[41] Gerstner W, Kistler WM, Naud R, and Paninski L. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[42] Pfister JP, Toyoizumi T, Barber D, and Gerstner W. Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural computation*, 18(6):1318–1348, 2006.

[43] Michiel WH Remme and Wytse J Wadman. Homeostatic scaling of excitability in recurrent neural networks. *PLoS Comput Biol*, 8(5):e1002494, 2012.