Astromorphic Self-Repair of Neuromorphic Hardware Systems

Zhuangyu Han, A N M Nafiul Islam, Abhronil Sengupta

School of Electrical Engineering and Computer Science
Penn State University
University Park, PA 16802
zfh5141@psu.edu, nafiul@psu.edu, sengupta@psu.edu

Abstract

While neuromorphic computing architectures based on Spiking Neural Networks (SNNs) are increasingly gaining interest as a pathway toward bio-plausible machine learning, attention is still focused on computational units like the neuron and synapse. Shifting from this neuro-synaptic perspective, this paper attempts to explore the self-repair role of glial cells, in particular, astrocytes. The work investigates stronger correlations with astrocyte computational neuroscience models to develop macro-models with a higher degree of bio-fidelity that accurately captures the dynamic behavior of the self-repair process. Hardware-software co-design analysis reveals that bio-morphic astrocytic regulation has the potential to self-repair hardware realistic faults in neuromorphic hardware systems with significantly better accuracy and repair convergence for unsupervised learning tasks on the MNIST and F-MNIST datasets. Our implementation source code and trained models are available at https://github.com/NeuroCompLab-psu/Astromorphic_Self_Repair.

Introduction

As a pathway to enable next-generation neuromorphic intelligence, Spiking Neural Networks (SNNs) are emerging as a disruptive computing paradigm where the information transmitted and processed in the artificial neurons of the computing system is embodied in a temporal series of binary spikes which mimic the action potentials propagating between neurons in the mammalian brain. SNNs have demonstrated significant advantages in comparison to traditional non-spiking computational architectures in several aspects, such as low latency and low energy consumption (Davies et al. 2018; Merolla et al. 2014; Sengupta and Roy 2017; Diehl and Cook 2015; Lu and Sengupta 2020; Neftci, Mostafa, and Zenke 2019).

Recent research in neuromorphic computing has started focusing on other cellular components in the brain that might contribute to cognition in addition to neuronal spiking behavior and synaptic plasticity. Specifically, this work explores the contribution of one such component - glial cells, in particular, the astrocytes (Oberheim et al. 2006). It has been observed that neurotransmitters released from a neuron can trigger diffusing Ca^{2+} waves inside the cytoplasm

of neighboring astrocytes, which later regulates the strength of the synapses surrounded by these astrocytes (Cornell-Bell et al. 1990; Perea and Araque 2007). Previous works have considered the regulation mechanism of the astrocytes as a self-repair capability of artificial spiking neural networks (Wade et al. 2012; Liu et al. 2017, 2018; Rastogi et al. 2021) with synaptic faults. This is especially timely given the significant advancements in Artificial Intelligence (AI) memristive hardware over the past few years enabled by devices where the core physics serve as a natural hardware substrate for designing compact, energy-efficient neuromorphic hardware accelerators (Sung, Hwang, and Yoo 2018; Park et al. 2022). However, such devices often suffer from non-idealities and faults (Liu et al. 2014; Radetzki et al. 2013) and therefore mitigation of such issues is critical to ensure minimal accuracy degradation in AI hardware platforms. While astrocytic regulation of synaptic transmission probability offers motivation toward autonomous self-repair of faulty neuromorphic hardware, current work significantly falls short of this goal due to a lack of hardware insights. The neuroscience inspired algorithm frameworks are often loosely bio-inspired and depend on *global network level parameters* like synaptic weight percentiles (Rastogi et al. 2021) which are not practical to calculate in real-time under energy and resource constraints in edge AI systems.

This work forges stronger connections with neuroscience computational models of astrocytes to develop self-repair algorithmic frameworks driven by a software-hardware co-design perspective. The key distinguishing factors of our work against prior proposals are:

(i) Neuroscience Inspired Self-Repair Learning Algorithm Formulation: Based on the bio-inspiration premise that self-repair learning algorithm formulations need to have stronger neuroscience correlation in order to be hardware efficient and compatible, we design a self-repair rule by theoretically analyzing the dynamic temporal behavior of the self-repair process enabled by astrocytes. We develop a macro-model of the self-repair process by benchmarking it to astrocyte computational neuroscience models. As shown in the paper, the bio-inspiration route enables us to construct self-repair-based synaptic learning rules that capture the dynamic temporal repair process while simultaneously depending on *local network level parameters* that make it hardware compatible. Performance evaluation of the pro-

posed learning rule is demonstrated for an unsupervised learning framework on the MNIST and F-MNIST datasets.

(ii) Algorithm-Hardware Co-Design: Going beyond software simulated faults where synapse transmission probabilities (PR) are assumed to be zero, we will analyze the impact of non-idealities occurring in the neuromorphic hardware, which are significantly more complex than stuck-at-zero faults. In particular, we focus on synaptic weight drift effects observed in memristive AI hardware (Ambrogio et al. 2019), in addition to stuck-at faults. This can therefore lead to the development of highly adaptive neuromorphic systems that can self-repair faults (device level non-idealities) in an autonomous fashion.

Related Work and Main Contributions

Existing research on AI hardware design with device level faults ranges from fault masking and redundancy repair schemes (Chen, Li, and Tseng 2012) to retraining (Xia et al. 2017). Both approaches require complex fault detection schemes. In contrast, our repair scheme does not require any fault localization scheme. Further, our self-repair algorithm is entirely local, thereby making it hardware-compatible. Prior retraining based approaches depend on global learning rules. Implementation of local learning rules is an active research area (Zenke and Neftci 2021).

In the domain of astromorphic unsupervised self-repair, while prior work like (Rastogi et al. 2021; Liu et al. 2017, 2018) discusses astrocyte computational modelling, they only draw qualitative observations from the model. For instance, the learning algorithm developed in (Rastogi et al. 2021) is primarily based on the qualitative observation that after the introduction of the faults, the synaptic transmission probability (PR) of the synapses with the higher initial PR value is enhanced greatly compared to the one with the lower initial PR. On the other hand, our work develops the self-repair algorithm based on quantitative macro-modelling of the astrocyte computational model. This enables the critical modelling of the temporal dynamics of the self-repair process which results in faster repair.

Another significant contribution of the work is hardware-software co-design where we evaluate the performance of our developed algorithm in presence of hardware non-idealities like conductance drift which is much more complex than prior simulated stuck-at-faults.

Finally, while there have been prior efforts at mimicking astrocyte functionality in hardware (Rahiminejad et al. 2021), they primarily focus on implementing the detailed computational model in hardware without considering aspects of astrocyte functionality relevant to self-repair of neuromorphic hardware.

Methods

Astrocyte Mediated Synaptic Dynamics

Astrocytes modulate the transmission characteristics, fundamentally the spike transmission probability, of the synapses they ensheathe (Wade et al. 2012). Endocannabinoids-mediated synaptic potentiation (e-SP) (Navarrete and Araque, 2010) and Depolarization-induced suppression of excitation (DSE) (E. Alger 2002) are two dynamic modulating factors that influence the PR evolution of synapses. Through a signal pathway inside the astrocyte cytoplasm, e-SP increases the PR after each post-synaptic neuron firing event. In contrast, DSE provides direct negative feedback to the firing rate of post-synaptic neurons. Prior work on computational models of astrocytes has outlined the quantitative change of 2-arachidonoyl glycerol (2-AG) as the first process initiating PR modulation (Wade et al. 2012). The post-synaptic neuron releases 2-AG every time it fires, and the 2-AG decays exponentially when there is no release from the dendrite of the post-synaptic neuron. The dynamics of 2-AG can be described as:

$$\frac{d(\text{AG})}{dt} = \frac{-\text{AG}}{\tau_{\text{AG}}} + r_{\text{AG}}\delta(t - t_{sp}) \qquad (1)$$

where, AG is the quantity of 2-AG released by the post-synaptic neuron, τ_{AG} is the decay time constant of 2-AG, r_{AG} is the production rate of 2-AG and t_{sp} is the time when the post-synaptic neuron spikes. Note that 2-AG is a neuron-specific variable, which implies that the spikes of all synapses of a neuron will contribute to the evolution of 2-AG of this neuron.

The binding process of 2-AG to CB1R receptors on the pre-synaptic neuron triggers DSE. The relationship between 2-AG and DSE can be described by a simple linear equation:

$$\text{DSE} = -\text{AG} \times K_{\text{AG}} \qquad (2)$$

where, AG is the quantity of 2-AG released by the post-synaptic neuron and K_{AG} is a linear scaling factor. DSE is also a neuron-specific variable.

Concurrently, 2-AG also binds to the CB1Rs on the plasma membrane of the astrocyte, which subsequently triggers the generation of IP$_3$ inside the cytoplasm of astrocytes. IP$_3$ later binds on the IP$_3$ receptor in the Endoplasmic Reticulum (ER), which releases Ca^{2+} into the cytoplasm. The Ca^{2+} dynamics, described by the Li-Rinzel model (Li and Rinzel 1994), involves three Ca^{2+} currents which are J_{chan} (the current through Ca^{2+} channel controlled by mutual gating of Ca^{2+} and IP$_3$), J_{leak} (the leakage current from ER into the cytoplasm) and J_{pump} (the pumping current where the Ca^{2+} is absorbed into ER through Sacro-Endoplasmic-Reticulum Ca^{2+}-ATPase (SERCA) pumps). Interested readers are referred to Refs. Wade et al. (2012) and De Pittà et al. (2009) for details on the computational model. The intracellular Ca^{2+} dynamics can be described as:

$$\frac{d\text{Ca}^{2+}}{dt} = J_{\text{chan}} + J_{\text{leak}} - J_{\text{pump}} \qquad (3)$$

The exocytosis of glutamate is regulated by Ca^{2+}, which can be described as:

$$\frac{d(\text{Glu})}{dt} = \frac{-\text{Glu}}{\tau_{\text{Glu}}} + r_{\text{Glu}}\delta(t - t_{\text{Ca}^{2+}}) \qquad (4)$$

where, Glu is the total quantity of glutamate released by the astrocyte, τ_{Glu} is the glutamate decay time constant, r_{Glu} is the fixed amount of glutamate release each time the Ca^{2+}

concentration crosses the threshold and $t_{\text{Ca}^{2+}}$ is the time instant when the Ca^{2+} concentration crosses the threshold.

Glutamate binding is the final stage of potentiation where the synapse has its PR increased when the group I metabotropic Glutamate Receptors (mGluRs) on the membrane of the pre-synaptic neuron receives the glutamate, whose dynamics is described by:

$$\tau_{\text{eSP}} \frac{d(\text{eSP})}{dt} = -\text{eSP} + m_{\text{eSP}} \text{Glu}(t) \qquad (5)$$

where, eSP is the intensity of e-SP, τ_{eSP} is the decay time constant of eSP, m_{eSP} is a scaling factor and $\text{Glu}(t)$ is the released glutamate quantity at time t. Please note here that eSP is an astrocyte-specific variable which influences all the synapses of all the neurons ensheathed by the astrocyte.

Potentiation provides positive feedback to synaptic firing, while DSE depresses synaptic firing. The PR dynamics as a function of time can be described as a combined effect of DSE and eSP as follows:

$$\text{PR}(t) = \text{PR}(0) + \text{PR}(0) \times \left(\frac{\text{DSE}(t) + \text{eSP}(t)}{100} \right) \qquad (6)$$

where, $\text{PR}(0)$ is the PR of the synapse before any firing.

As described by the dynamic astrocyte model, DSE is established by the direct binding of 2-AG to the pre-synaptic neuron, which produces a restricted effect on a single neuron. However, the e-SP feedback through the cytoplasm of the astrocyte is globally effective for all synapses ensheathed by the corresponding astrocyte. During the self-repair process enabled by the astrocyte, when the PR of a synapse drops due to faults, the magnitude of DSE of that specific neuron (considering that DSE is always negative) will immediately decline caused by the decreasing 2-AG. However, the global e-SP maintains its intensity due to the continuous firing of the neighboring neurons. Consequently, the PR of healthy synapses for the affected neuron with faults will increase based on Equation 6 and recover the spiking activity of the neuron to the initial baseline value. The next section discusses the formulation of a benchmarked astrocyte macro-model that captures the dynamical repair process for hardware realistic faults based on this computational neuroscience model.

Macro-model Formulation for Hardware Realistic Synaptic Fault Repair

Neuron-Astrocyte Structure Configuration The two-neuron-one-astrocyte computational simulation model (Wade et al. 2012), shown in Figure 1, is adopted in this paper. In such a model, there are two neurons, N1 and N2, with identical biological characteristics and one astrocyte functioning as the e-SP pathway for both neurons. The neuron firing model is discussed in detail in the next subsection. The two neurons have the same number of synapses and each synapse receives a Poisson spike train with a globally constant rate parameter, i.e. 10 spikes per second. Please note that Figure 1 shows only one dendrite for each neuron, for simplicity, which implies that the DSE shown in the figure is not restricted to the single synapse, instead it is effective neuron-wide. The initial PR values of
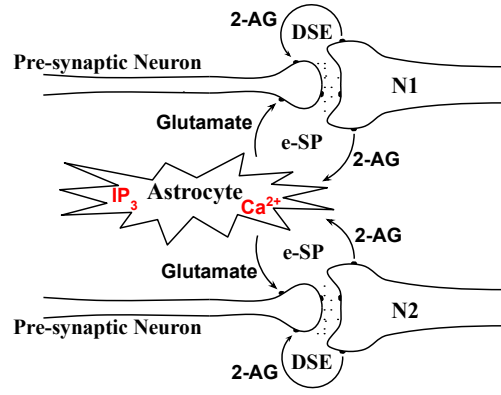

Figure 1: Two neurons interacting with a common astrocyte.

each synapse are initialized randomly and are sampled from a uniform distribution whose mean is pre-defined.

The signalling pathway of the astrocyte is computationally realized by a group of global variables, such as the quantity of IP_3, Ca^{2+} and glutamate. The interaction between the neurons and the astrocyte is represented by the change of variable 2-AG and glutamate quantity.

The total simulation duration is 400 s and a resolution of 1 ms is used for modelling the temporal dynamics. The Poisson spike trains are continuously received by all the synapses for the entire duration of the simulation. A specific amount of current (6650 pA in this paper) is injected into the neuron if a spike is successfully transmitted through the synapse.

Neuron Firing Model The biological attributes of the neurons are critical to the dynamics of the neuron-astrocyte structure. For the computational simulations performed in this paper, the Leaky-Integrate-Fire model is used for modeling the neuron dynamics. The evolution of neuron membrane potential follows the equation below:

$$\tau_v \frac{dv(t)}{dt} = -(v(t) - v_{\text{res}}) + I(t) \qquad (7)$$

where, v is the membrane potential, τ_v is the decay time constant of the membrane potential, v_{res} is the resting state potential and $I(t)$ is the injected current which is the total current the neuron receives from all the synapses. When the membrane potential reaches a threshold potential, v_{th}, the neuron fires once and immediately the membrane potential is set to a pre-determined value, v_{reset}. No refractory period, δ_{ref}, is considered in this computational model.

Memristive Hardware Modelling and Fault Simulation At the hardware level, the PR values (equivalently synaptic weights for artificial spiking neural networks used for machine intelligence) of the network are mapped to the conductance states of the memristive devices. In this paper, we specifically consider device characteristics corresponding to a representative memristive technology - Phase Change Memory (PCM) (Fong, Neumann, and Wong 2017). Organized in a crossbar fashion, as shown in Figure 2, these arrays of devices can implement the dot product operation. The input voltage V_i coming from the pre-synaptic neurons
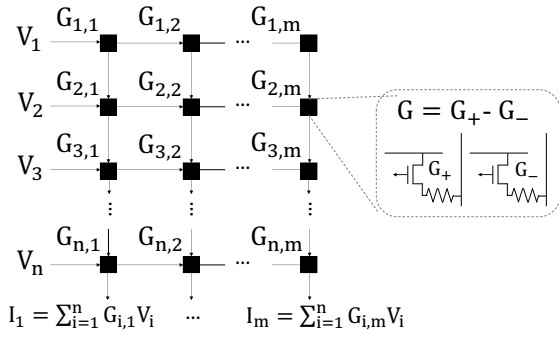
Figure 2: Schematic of the network mapped to a memristive crossbar array. Each weight is encoded by the conductance of a pair of PCM devices. Additional peripheral circuitry is omitted.

to each row of synapses are modulated by the conductance of the synaptic devices and are summed along the columns by Kirchhoff's current law. Thus, the post-synaptic current fed to the post-synaptic neuron j is given by:

$$I_j = \sum_{i=1}^{n} G_{i,j} V_i \qquad (8)$$

where, n is the total number of rows, i.e. total number of pre-synaptic neurons. Note, as conductance values are always positive, in order to represent both negative and positive weights, each weight is actually mapped to a pair of PCM devices $(G_+, G_-,$ shown in Figure 2 inset) where the positive (negative) conductance encodes the positive (negative) weight value while the other conductance is set to a high OFF resistive state. For sake of simplicity, in our following discussions, we will consider the conductance to be given by $G = G_+ - G_-$. The intrinsic device physics of PCM devices can be exploited to implement synaptic learning rules like Spike Timing Dependent Plasticity (STDP) (Sebastian et al. 2018) while the array structure enables the area and energy-efficient implementation of synaptic dot-product computation required in artificial neural networks (Fong, Neumann, and Wong 2017).

However, memristive devices suffer from several non-idealities. While previous works on astrocyte enabled self-repair in neural networks (Wade et al. 2012; Liu et al. 2017, 2018; Rastogi et al. 2021) have considered stuck-at-faults, hardware realistic faults are much more complex. In this work, we consider two specific hardware non-idealities: stuck-at-fault and weight drift. The stuck-at-fault is a type of fault where a pre-determined percentage of synapses are faulty or disabled, i.e. PR = 0, and no longer affected by any dynamics. The weight drift scenario occurs in the healthy synapses and is representative of PCM device technology. Therefore, along with accounting for a portion of the synapses to be disabled (representing the simulation of stuck-at-faults) in our modelling, the PRs of healthy synapses drift with a randomly generated drift ratio. The drift fault pattern is extracted from the temporal PCM device characteristics (Ambrogio et al. 2019). The temporal decay
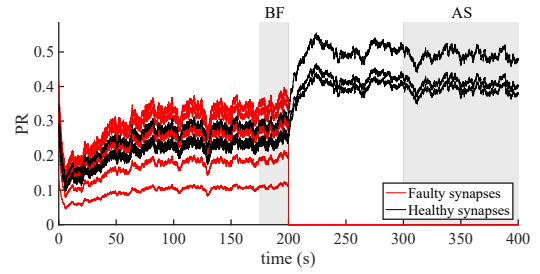

Figure 3: An example of temporal evolution of PR value of neuron N2 under astrocytic influence, where the total number of synapses is 10.

of the conductance of a PCM device can be described as:

$$G = G_0 t_{norm}^{-v} \qquad (9)$$

where, G_0 is the initial conductance, t_{norm} is the normalized time, which is greater than 1, and v is the log scale decay slope. The value $r_{decay} = \frac{\dot{G}}{G_0} = t_{norm}^{-v}$ is called the drift ratio. Also, instead of a constant parameter, the log scale decay slope v is sampled from a normal distribution for each individual PCM device. i.e.

$$v \sim \mathcal{N}(\mu_v, \sigma_v) \qquad (10)$$

where, μ_v and σ_v represent the mean and standard deviation of the distribution of v. In both scenarios mentioned above, the fault is instantly injected to neuron N2 at time $t_{fault} = 200$ s. Neuron N1 is not affected by any fault. For the stuck-at-fault scenario, the impacted PRs drop to 0 at t_{fault} and remain at that value. For modelling both the stuck-at-fault and drift process, at t_{fault}, the PRs of disabled synapses are set to 0 in the same way as the former. At the same time, each healthy synapse PR is multiplied by its individual r_{decay}. In the following sections, without explicit declaration, the PRs are the PR values of neuron N2, as N1 is not influenced by fault.

Self-Repair Macro-model Formulation   In this subsection, we develop a macro-model capturing the temporal dynamics of the astrocyte induced self-repair process by utilizing the detailed computational model discussed previously. For simplicity, the macro-model is developed based on the response of the computational model to stuck-at-faults while performance evaluation on machine learning tasks (discussed in next section) is performed for both stuck-at-faults and weight drift non-idealities. Based on Equation 6, it can be observed that $PR(t)$ is a product of $PR(0)$ and a global variable $1 + (DSE(t) + eSP(t))/100$. Due to the stochastic nature of PR initialization and fault injection, the amount of self-repair varies in different simulation runs. In general, the PRs of healthy synapses increase after the occurrence of faults and finally stabilize at a certain level which is regulated by the competition between direct and indirect signal feedback. From Figure 3, where a sample of the temporal evolution of PR values is plotted, it can be observed that the time cost for self-repair, which is defined as the time difference between t_{fault} and the time the PR takes to reach

its repaired stable value, is similar for each synapse. In fact, we can prove that all the PRs keep their relative magnitude unchanged at any time during the simulation window unless there is any particular PR hitting the upper bound 1. At any given time t_1 and t_2, the ratio between two arbitrary PRs, say PR_i and PR_j, is

$$
\begin{aligned}
\frac{\mathrm{PR}_i(t_1)}{\mathrm{PR}_j(t_1)} &= \frac{\mathrm{PR}_i(0)(1 + \frac{\mathrm{DSE}(t_1)+\mathrm{eSP}(t_1)}{100})}{\mathrm{PR}_j(0)(1 + \frac{\mathrm{DSE}(t_1)+\mathrm{eSP}(t_1)}{100})} \\
&= \frac{\mathrm{PR}_i(0)(1 + \frac{\mathrm{DSE}(t_2)+\mathrm{eSP}(t_2)}{100})}{\mathrm{PR}_j(0)(1 + \frac{\mathrm{DSE}(t_2)+\mathrm{eSP}(t_2)}{100})} = \frac{\mathrm{PR}_i(t_2)}{\mathrm{PR}_j(t_2)}
\end{aligned}
\tag{11}
$$

Therefore all the synapses finish their self-repair simultaneously. Also, from Equation 11, we can conclude that at any given time t_1 and t_2 and two arbitrary PRs, say PR_i and PR_j, the following relation holds:

$$
\frac{\mathrm{PR}_i(t_1)}{\mathrm{PR}_i(t_2)} = \frac{\mathrm{PR}_j(t_1)}{\mathrm{PR}_j(t_2)}
\tag{12}
$$

In this work, the PR self-repair ratio q of a synapse, say synapse i, is defined as the ratio of the PR value after self-repair and the stable PR value before fault injection:

$$
q_i = \frac{\overline{\mathrm{PR}_i(\mathrm{AS})}}{\overline{\mathrm{PR}_i(\mathrm{BF})}}
\tag{13}
$$

where, $\overline{\mathrm{PR}_i(\mathrm{AS})}$ is the mean (averaged over multiple simulation runs) of stable PR value of synapse i after self-repair and $\overline{\mathrm{PR}_i(\mathrm{BF})}$ stands for the mean of stable PR value of synapse i just before fault injection. The intervals of AS and BF are shaded in Figure 3. It is easy to infer from Equation 12 that $q_1 = q_2 = .. = q_m$ where m is the number of synapses of neuron N2. Therefore, we will denote q without a subscript for neuron N2 for the remainder of the text.

Based on the computational model simulations, we inferred that the self-repair ratio q is a function of the initial PRs and the fault magnitude. By statistics, the larger the percentage of affected synapses, larger will be the value of q. Based on data collected from 400 simulation runs with independent randomly sampled PR initialization and fault injection, it was confirmed that q is strongly correlated to the severity of the fault. Here, the severity of the fault z is defined as:

$$
\begin{aligned}
z &= \frac{\text{sum of initial PRs of non-disabled synapses}}{\text{sum of initial PRs of all the synapses}} \\
&= \frac{\sum_{\text{non-disabled synapses}} \mathrm{PR}_i(0)}{\sum_{\text{all synapses}} \mathrm{PR}_i(0)}
\end{aligned}
\tag{14}
$$

where, $\mathrm{PR}_i(0)$ is the initial PR of synapse i. In the context of terminology to be used in the next section for learning algorithm formulation in self-repair of neuromorphic hardware systems, a non-disabled synapse is a synapse whose PR is not stuck to zero and therefore includes synaptic weights with a drift factor. Quantitatively, the strong correlation found between q and z from the astrocyte computational model simulations is:

$$
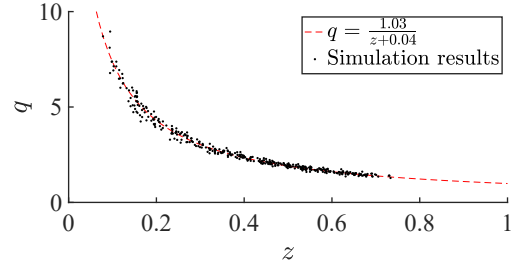q \approx \frac{1.03}{z + 0.04}
\tag{15}
$$

Figure 4: Quantitative relationship between self-repair ratio, q, and fault severity, z.

which is also displayed in Figure 4.

For simplicity, an inverse proportion relation is considered in our algorithm design (to be discussed in next section), i.e.

$$
q \leftarrow \frac{1}{z}
\tag{16}
$$

Such a strong correlation between q and z implies that the astrocyte self-repair mechanism will drive the PR of healthy synapses of a faulty neuron to increase until the sum of PR of the remaining healthy synapses approaches the original synaptic PR sum before faults. The temporal increment of PR of healthy synapses after fault injection follows an exponential-like trend since the self-repair becomes relatively slow as the PR of healthy synapses approach their target values. i.e.,

$$
\begin{aligned}
\mathrm{PR}_i(t) &\approx \overline{\mathrm{PR}_i(\mathrm{AS})}(1 - e^{-\frac{(t - t_{\mathrm{fault}} + t_b)}{\tau}}) \\
&= q \times \overline{\mathrm{PR}_i(\mathrm{BF})}(1 - e^{-\frac{(t - t_{\mathrm{fault}} + t_b)}{\tau}}) \\
&\forall\, t > t_{\mathrm{fault}}
\end{aligned}
\tag{17}
$$

where

$$
t_b = -\tau \log(\frac{q-1}{q})
\tag{18}
$$

is the temporal intercept and τ is the self-repair time constant. The value of the temporal intercept can be determined from the initial condition that at $t = t_{\mathrm{fault}}$, the transmission probability starts to increase from an initial value $\overline{\mathrm{PR}_i(\mathrm{BF})}$.

The next section discusses astromorphic learning algorithm formulation for unsupervised SNNs inspired by the developed macro-model.

Astromorphic Learning Algorithm Formulation

To demonstrate our proposal, we consider an SNN trained via the STDP learning algorithm where the synaptic weights are updated according to the timing of spikes of pre-synaptic and post-synaptic neurons. For detailed information about the learning algorithm and training process, interested readers are directed to (Diehl and Cook 2015). An STDP network for image recognition has a structure as elaborated in Figure 5. The dimension of the input layer, n_{input}, is dictated by the dimension of the input images for recognition while the main network consists of n_{neuron} neurons. To prevent single neurons from dominating the firing pattern during training, homeostasis is included in the neuron model:
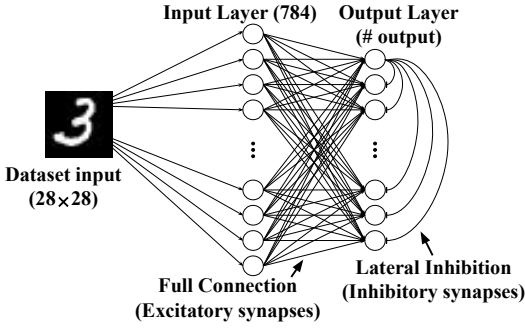
Figure 5: SNN network architecture used for unsupervised learning. Lateral inhibitory connections are only shown for one neuron in the output layer.

a neuron spike causes a constant threshold increment (θ^+) over the baseline threshold. The threshold increment decays exponentially, according to decay time constant (τ_θ), with time when the neuron does not fire. Such a mechanism prevents a neuron from firing continuously, which enables the entire network to learn uniformly. The homeostasis effect is also balanced by a lateral inhibitory effect where any neuron spike in the output layer triggers negative spikes to all the other neurons (where the weights for all inhibitory connections is a global constant, w_{inh}) and prevents them from firing, thereby promoting competitive learning. The representative counterpart of synaptic PR from the computational model is the set of synaptic weights in the SNN connecting the input and output neuron layers. There will be a positive weight update whenever the post-synaptic neuron fires after the pre-synaptic neuron and vice versa. The trace of a pre/post-synaptic neuron is set to 1 at the instant it generates a spike, and the trace decays exponentially afterward. The trace of a neuron measures the temporal closeness of the current time and its immediate previous spike. The STDP learning rule based on post/pre-synaptic neuron spike traces can be formulated as:

$$\Delta w(t) = \begin{cases} \eta_{post} \times x_{pre}(t) & \Delta t > 0 \\ -\eta_{pre} \times x_{post}(t) & \Delta t < 0 \end{cases} \quad (19)$$

where, η_{post}/η_{pre} are the post/pre-synaptic neuron learning rates and x_{pre}/x_{post} are the traces of the pre/post-synaptic neurons. Δt is the time difference between post-synaptic and pre-synaptic spikes, i.e. $\Delta t = t_{post} - t_{pre}$. Practically, the $\Delta t > 0$ case is executed when the post-synaptic neuron fires and the $\Delta t < 0$ case is executed when the pre-synaptic neuron fires.

Astrocyte feedback causes the modulation of synaptic plasticity in the presence of faults, in turn, modulating the STDP learning rule. Inspired by the inverse proportional relationship between the self-repair ratio and the fault ratio, the self-repair learning rule is expected to recover the original sum of synaptic PRs of a faulty neuron. The target repaired value of healthy synaptic weights of a faulty neuron is therefore determined by q. We note that an exponential increasing temporal pattern (exhibited by the dynamic repair process in astrocyte computational model in Equation

(17)) is commonly represented by the following equation: $\tau_x \frac{dx(t)}{dt} = x_{\text{target}} - x(t)$, where x_{target} is the value that the objective x converges to and τ_x is the convergence time constant. The slope of increment of the objective at any given time t is proportional to the difference between the target value and the current value. Therefore, the astrocyte modulated STDP learning rule can be described by the following equation where the rate of weight change is directly proportional to the difference between the target synaptic PR and the current weight:

$$\Delta w(t) = \begin{cases} \eta_{post} \times x_{pre}(t) \times \frac{qw_0 - w(t)}{\tau} & \Delta t > 0 \\ -\eta_{pre} \times x_{post}(t) & \Delta t < 0 \end{cases} \quad (20)$$

where, w_0 is the weight of the synapse before fault injection, $w(t)$ is the current weight, τ is the self-repair time constant (which can be combined with the learning rate as a single hyperparameter), q is the self-repair ratio mentioned in the previous section.

Here, the self-repair ratio q is approximated by $\frac{1}{z}$, where z is a function of the sum of the synaptic weights of the corresponding neuron. It is therefore worth pointing out here that the resultant computational model inspired learning rule is *local* where the synaptic weight updates are dependent on the current weight (*local to synapse*) and sum of weights of the corresponding neuron (*local to neuron*). This enables our algorithm to be memristive hardware compatible (for instance, sum of weights of a particular neuron can be evaluated by applying all-one voltages along the rows of the crossbar array in Figure 2) unlike prior proposals (Rastogi et al. 2021) where the temporal dynamics of the weight updates during self-repair are governed by *global* parameters like percentile of the weight distribution of the entire network and are also lacking in bio-fidelity from modelling perspective.

The term qw_0 in Equation 20 is the self-repair target value of the corresponding weight. Note that q_{eff} is not constant with time, therefore the target value of each synaptic PR/weight can change during the re-training process.

The next section evaluates the performance of our proposed astromorphic learning rule against prior proposals in unsupervised SNNs for hardware-realistic faults. For clarity, we will refer to the basic spike-timing dependent synaptic plasticity as STDP, astrocyte mediated plasticity in prior proposals (Rastogi et al. 2021) as A-STDP (global) and our proposal as A-STDP (local).

Results

Experiment Setup

The proposed A-STDP (local) learning rule is implemented using the BindsNET (Hazan et al. 2018) framework - an open-source SNN simulation platform based on PyTorch (https://pytorch.org/). The algorithms were run on the hardware environment consisting of one Intel(R) Xeon(R) Silver 4210 CPU, one NVIDIA GeForce RTX 2080 Ti GPU with 11264 MBytes graphics memory, 187 GBytes RAM and CentOS Linux 7 operating system. The SNNs are trained using STDP on two recognition tasks, namely the MNIST (LeCun, Cortes, and Burges 2010) and Fashion MNIST (Xiao,

Dataset	p_{fault}	Acc. Norm. (%)	Acc. STDP (%)	# Steps STDP (10^4 steps)	Acc. A-STDP (global) (%)	# Steps A-STDP (global) (10^4 steps)	Acc. A-STDP (local) (%)	# Steps A-STDP (local) (10^4 steps)
MNIST (Baseline: 91.43%)	0.5	27.61 (0.96)	83.22 (0.39)	8.8 (2.5)	76.26 (0.61)	7.7 (1.8)	87.31 (0.34)	7.9 (1.6)
	0.6	27.18 (0.16)	79.92 (0.62)	9.0 (1.9)	73.62 (0.78)	5.8 (1.2)	84.89 (0.32)	6.2 (2.1)
	0.7	27.02 (0.76)	70.11 (0.90)	7.3 (1.0)	70.83 (0.50)	6.9 (1.8)	82.45 (0.51)	7.4 (2.5)
	0.8	26.33 (1.16)	51.19 (0.86)	5.0 (1.2)	66.93 (0.31)	8.0 (1.7)	77.68 (0.85)	6.3 (1.3)
	0.9	27.68 (0.25)	39.41 (1.56)	0.7 (0.6)	65.42 (0.86)	9.3 (1.4)	68.58 (0.55)	9.3 (1.2)
Fashion MNIST (Baseline: 77.60%)	0.5	32.40 (1.42)	75.75 (0.25)	19.9 (4.9)	67.30 (0.56)	17.5 (3.4)	76.14 (0.30)	1.5 (0.9)
	0.6	32.18 (1.40)	74.17 (0.47)	21.4 (1.6)	65.76 (0.12)	15.1 (6.0)	74.74 (0.23)	2.5 (1.1)
	0.7	31.12 (0.38)	72.63 (0.64)	22.9 (0.8)	64.49 (0.40)	18.0 (4.8)	72.67 (0.53)	2.1 (1.1)
	0.8	30.30 (0.65)	67.38 (0.58)	19.5 (2.7)	61.79 (1.01)	19.6 (5.0)	69.76 (0.43)	2.2 (1.6)
	0.9	29.90 (0.38)	43.48 (1.65)	0.7 (0.6)	57.75 (0.53)	23.4 (0.7)	64.54 (0.69)	4.3 (2.6)

Table 1: Self-repair accuracy and convergence speed comparison among STDP, A-STDP (global) and our proposed A-STDP (local) learning rules for 400 neuron network trained on MNIST and Fashion-MNIST datasets. Statistics is averaged over 5 runs.

Parameters	Values	Parameters	Values
Simulation time duration per image, $time$	100 ms	No. of output neurons, n_{neuron}	400
Simulation time-step size, Δt	1 ms	No. of input neurons, n_{input}	784
Membrane potential decay time const., τ_v	100 ms	Batch size	16
Refractory period, δ_{ref}	5 ms	Spike trace decay time const.	20 ms
Resting membrane potential, v_{res}	-65 mV	STDP weight normalization factor	78.4
Reset membrane potential, v_{reset}	-60 mV	Maximum input spike rate (MNIST)	128 /s
Threshold membrane potential, v_{th}	-52 mV	Maximum input spike rate (Fashion MNIST)	45 /s
Adaptive threshold increment, θ_+	0.05 mV	Inhibitory synaptic weight (MNIST), w_{inh}	-120
Adaptive threshold decay time const., τ_θ	10^7 ms	Inhibitory synaptic weight (Fashion MNIST), w_{inh}	-250
A-STDP (global) weight-percentile, α (Rastogi et al. 2021)	98	Post-synaptic learning rate (MNIST), η_{post}	10^{-2}
A-STDP (global) non-linearity, σ (Rastogi et al. 2021)	2	Post-synaptic learning rate (Fashion MNIST), η_{post}	4×10^{-3}
A-STDP (local) self-repair time const. (MNIST), τ	10^{-2}	Pre-synaptic learning rate (MNIST), η_{pre}	10^{-4}
A-STDP (local) self-repair time const. (Fashion MNIST), τ	4×10^{-3}	Pre-synaptic learning rate (Fashion MNIST), η_{pre}	4×10^{-5}
A-STDP (local) normalization period	1 batch	Weight drift normalized time, t_{norm}	10^4
Weight sum lower bound (MNIST), $\text{LB}_{\text{WeightSUM}}$	0.17	Weight drift log slope mean, μ_v	1
Weight sum lower bound (Fashion MNIST), $\text{LB}_{\text{WeightSUM}}$	0.22	Weight drift log slope standard derivation, σ_v	0.2258

Table 2: Self-repair retraining hyperparameters.

Rasul, and Vollgraf 2017) datasets. The intensity of the pixels of the input images are converted into Poisson spike trains (the firing rates for different datasets are noted in Table II). The weights between the input layer and the output layer are randomly initialized with a uniform distribution of 0-0.3 range. Two types of hardware realistic faults (stuck-at-zero and weight drift) mentioned previously are injected in the baseline trained networks. Performance assessment in terms of accuracy and repair speed is performed for three generations of self-repair learning rules - STDP, A-STDP (global) (Rastogi et al. 2021) and A-STDP (local).

First, the baseline networks (which do not possess any fault) are prepared by training a randomly initialized network with the STDP learning rule. The baseline networks, consisting of 400 neurons, are individually trained on MNIST/Fashion MNIST dataset with a final accuracy of 91.43%/77.60%. Next, stuck-at-faults are injected in the weights of the baseline network with a certain fault probability p_{fault}. i.e. each weight in the baseline network is set to zero with probability p_{fault}. Five values for $p_{\text{fault}} = 0.5$, 0.6, 0.7, 0.8 and 0.9 are tested to mimic varying fault sever-

ity. The weight drift involves three parameters: t_{norm}, μ_v and σ_v (mentioned in prior sections). Subsequently, STDP, A-STDP (local) and A-STDP (global) self-repair re-training is performed for comparative performance assessment. The best accuracy and number of data samples consumed before reaching best accuracy are recorded and analyzed. Interested readers can find the network weight maps before and after the introduction of non-idealities (stuck-at-faults and weight drift), including the self-repaired weights, in the supplementary material (Han, Islam, and Sengupta 2022). The detailed information regarding all simulation hyperparameters is included in Table 2 and typical ablation studies for important hyperparameters of the astromorphic learning model are also included in the supplementary material (Han, Islam, and Sengupta 2022).

Like previous works (Rastogi et al. 2021), the weights of all neurons are normalized during the training process:

$$\text{WeightSUM}_i \leftarrow \overline{\text{WeightSUM}} \qquad (21)$$

where, WeightSUM_i represents the sum of weights of neuron i and $\overline{\text{WeightSUM}}$ is the network-wide average of the

sum of weights for all the neurons. The normalization operation is applied after each batch of STDP learning, which guarantees that all the neurons have the same chance to be trained. Since the STDP algorithm updates the weights only when there is any spike and highly severe faults can result in very low spiking activity in the network, a lower bound of $\overline{\text{WeightSUM}}$, namely $\text{LB}_{\text{WeightSUM}}$, is configured as a network hyperparameter to improve the training efficiency.

Accuracy and Convergence Speed Analysis

Table 1 reports the best accuracy recorded in the self-repair process for hardware-realistic faults in the network (stuck-at-faults and weight drift). The results have been averaged over 5 independent runs of the network. In Table 1, Acc. Norm represents the network accuracy after fault injection and weight normalization. Acc. STDP/Acc. A-STDP (global)/Acc. A-STDP (local) is the accuracy measured after re-training using STDP/A-STDP (global)/A-STDP (local) learning rules respectively. # Steps describes the number of training samples consumed before reaching the maximum accuracy during the self-repair process. The standard deviation of each measurement is included in the parentheses. A-STDP (local) significantly outperforms A-STDP (global) in terms of self-repaired accuracy. Further, while the convergence speed of A-STDP (local) is comparable to A-STDP (global) in case of the MNIST dataset, it is significantly faster over the more complex Fashion-MNIST dataset. Training convergence graphs are also provided in the supplementary material (Han, Islam, and Sengupta 2022).

Discussion

This work strengthens the foundation for enabling astromorphic self-repair of hardware realistic faults in a neuromorphic system by forging stronger correlations with astrocyte neuroscience models. The key distinguishing aspect of the work lies in the neuroscience-hardware-software co-design aspect that delves into learning algorithm formulation that is not only neuroscience inspired but also hardware compatible. Performance assessment is provided in unsupervised learning tasks for hardware realistic faults going beyond simple stuck-at-faults considered in prior works.

Future work can focus on understanding other aspects of astrocyte functionality in the self-repair process. For instance, there is a pair of feedback pathways in the astrocyte computational model based on the firing rate of the corresponding neuron, whose details have not been fully explored in this work. Furthermore, the modulation of inhibitory synapses through astrocytes (Kang et al. 1998) for neuromorphic computing still need more research. From a machine learning perspective, unsupervised self-repair mechanisms need to be also scaled further to deeper networks and large-scale datasets.

Acknowledgments

References

Ambrogio, S.; Kumar, A.; Chen, A.; Burr, G. W.; Gallot, M.; Spoon, K.; Tsai, H.; Mackin, C.; Wesson, M.; Kariyappa, S.; Narayanan, P.; and Liu, C.-C. 2019. Reducing the Impact of Phase-Change Memory Conductance Drift on the Inference of large-scale Hardware Neural Networks. In *2019 IEEE International Electron Devices Meeting (IEDM)*, 6.1.1–6.1.4. San Francisco, CA, USA: IEEE. ISBN 978-1-72814-032-2.

Chen, T.-J.; Li, J.-F.; and Tseng, T.-W. 2012. Cost-efficient built-in redundancy analysis with optimal repair rate for RAMs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(6): 930–940.

Cornell-Bell, A. H.; Finkbeiner, S. M.; Cooper, M. S.; and Smith, S. J. 1990. Glutamate induces calcium waves in cultured astrocytes: long-range glial signaling. *Science*, 247(4941): 470–473. Publisher: American Association for the Advancement of Science.

Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S. H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; Liao, Y.; Lin, C.-K.; Lines, A.; Liu, R.; Mathaikutty, D.; McCoy, S.; Paul, A.; Tse, J.; Venkataramanan, G.; Weng, Y.-H.; Wild, A.; Yang, Y.; and Wang, H. 2018. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*, 38(1): 82–99.

Diehl, P. U.; and Cook, M. 2015. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9.

E. Alger, B. 2002. Retrograde signaling in the regulation of synaptic transmission: focus on endocannabinoids. *Progress in Neurobiology*, 68(4): 247–286.

Fong, S. W.; Neumann, C. M.; and Wong, H.-S. P. 2017. Phase-change memory—Towards a storage-class memory. *IEEE Transactions on Electron Devices*, 64(11): 4374–4385.

Han, Z.; Islam, A. N. M. N.; and Sengupta, A. 2022. Astromorphic Self-Repair of Neuromorphic Hardware Systems. ArXiv:2209.07428.

Hazan, H.; Saunders, D. J.; Khan, H.; Patel, D.; Sanghavi, D. T.; Siegelmann, H. T.; and Kozma, R. 2018. BindsNET: A Machine Learning-Oriented Spiking Neural Networks Library in Python. *Frontiers in Neuroinformatics*, 12: 89.

Kang, J.; Jiang, L.; Goldman, S. A.; and Nedergaard, M. 1998. Astrocyte-mediated potentiation of inhibitory synaptic transmission. *Nature Neuroscience*, 1(8): 683–692.

LeCun, Y.; Cortes, C.; and Burges, C. 2010. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Li, Y.-X.; and Rinzel, J. 1994. Equations for InsP3 Receptor-mediated [Ca2+]i Oscillations Derived from a Detailed Kinetic Model: A Hodgkin-Huxley Like Formalism. *Journal of Theoretical Biology*, 166(4): 461–473.

Liu, J.; Harkin, J.; Li, Y.; and Maguire, L. 2014. Online traffic-aware fault detection for networks-on-chip. *Journal of Parallel and Distributed Computing*, 74(1): 1984–1993.

Liu, J.; Harkin, J.; Maguire, L. P.; McDaid, L. J.; and Wade, J. J. 2017. SPANNER: A self-repairing spiking neural network hardware architecture. *IEEE transactions on neural networks and learning systems*, 29(4): 1287–1300.

Liu, J.; McDaid, L. J.; Harkin, J.; Karim, S.; Johnson, A. P.; Millard, A. G.; Hilder, J.; Halliday, D. M.; Tyrrell, A. M.; and Timmis, J. 2018. Exploring self-repair in a coupled spiking astrocyte neural network. *IEEE transactions on neural networks and learning systems*, 30(3): 865–875.

Lu, S.; and Sengupta, A. 2020. Exploring the Connection Between Binary and Spiking Neural Networks. *Frontiers in Neuroscience*, 14: 535.

Merolla, P. A.; Arthur, J. V.; Alvarez-Icaza, R.; Cassidy, A. S.; Sawada, J.; Akopyan, F.; Jackson, B. L.; Imam, N.; Guo, C.; Nakamura, Y.; Brezzo, B.; Vo, I.; Esser, S. K.; Appuswamy, R.; Taba, B.; Amir, A.; Flickner, M. D.; Risk, W. P.; Manohar, R.; and Modha, D. S. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197): 668–673.

Neftci, E. O.; Mostafa, H.; and Zenke, F. 2019. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks. *IEEE Signal Processing Magazine*, 36(6): 51–63.

Oberheim, N. A.; Wang, X.; Goldman, S.; and Nedergaard, M. 2006. Astrocytic complexity distinguishes the human brain. *Trends in Neurosciences*, 29(10): 547–553.

Park, T. J.; Deng, S.; Manna, S.; Islam, A. N.; Yu, H.; Yuan, Y.; Fong, D. D.; Chubykin, A. A.; Sengupta, A.; Sankaranarayanan, S. K.; et al. 2022. Complex oxides for brain-inspired computing: A review. *Advanced Materials*, 2203352.

Perea, G.; and Araque, A. 2007. Astrocytes Potentiate Transmitter Release at Single Hippocampal Synapses. *Science*, 317(5841): 1083–1086.

Radetzki, M.; Feng, C.; Zhao, X.; and Jantsch, A. 2013. Methods for fault tolerance in networks-on-chip. *ACM Computing Surveys*, 46(1): 1–38.

Rahiminejad, E.; Azad, F.; Parvizi-Fard, A.; Amiri, M.; and Linares-Barranco, B. 2021. A Neuromorphic CMOS Circuit With Self-Repairing Capability. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5): 2246–2258.

Rastogi, M.; Lu, S.; Islam, N.; and Sengupta, A. 2021. On the Self-Repair Role of Astrocytes in STDP Enabled Unsupervised SNNs. *Frontiers in Neuroscience*, 14: 603796.

Sebastian, A.; Le Gallo, M.; Burr, G. W.; Kim, S.; Bright-Sky, M.; and Eleftheriou, E. 2018. Tutorial: Brain-inspired computing using phase-change memory devices. *Journal of Applied Physics*, 124(11): 111101.

Sengupta, A.; and Roy, K. 2017. Encoding neural and synaptic functionalities in electron spin: A pathway to efficient neuromorphic computing. *Applied Physics Reviews*, 4(4): 041105.

Sung, C.; Hwang, H.; and Yoo, I. K. 2018. Perspective: A review on memristive hardware for neuromorphic computation. *Journal of Applied Physics*, 124(15): 151903.

Wade, J.; McDaid, L.; Harkin, J.; Crunelli, V.; and Kelso, S. 2012. Self-repair in a Bidirectionally Coupled Astrocyte-Neuron (AN) System based on Retrograde Signaling. *Frontiers in Computational Neuroscience*, 6.

Xia, L.; Liu, M.; Ning, X.; Chakrabarty, K.; and Wang, Y. 2017. Fault-tolerant training with on-line fault detection for RRAM-based neural computing systems. In *Proceedings of the 54th Annual Design Automation Conference 2017*, 1–6.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR*, abs/1708.07747.

Zenke, F.; and Neftci, E. O. 2021. Brain-inspired learning on neuromorphic substrates. *Proceedings of the IEEE*, 109(5): 935–950.