Scaling high-dimensional synaptic plasticity on neuromorphic hardware: parameter mapping and vectorized implementation

Amani Atoui*‡, Jakob Kaiser*, Philipp Spilger*, Eric Müller§, and Johannes Schemmel*
*Institute for Computer Engineering, Heidelberg University, Germany
§Kirchhoff Institute for Physics, Heidelberg University, Germany
‡amani.atoui@ziti.uni-heidelberg.de

***Abstract*—We present analog accelerated neuromorphic hardware, specifically BrainScaleS-2, as a reliable and efficient tool for studying high-dimensional biologically-plausible synaptic plasticity. We focus on a calcium-based synaptic plasticity rule that integrates different neuron and synapse variables on multiple timescales. Compared to our previous work, we automate the calibration of the analog circuits which emulate the calcium dynamics and scale the plasticity rule to multiple synapses using the SIMD extension of the embedded digital processor. To validate our implementation, we used standard stimulation protocols. The analog nature of the hardware, reduced-precision arithmetic, and the larger integration time steps of the plasticity rule update compared to the simulation cause numerical differences between the emulation and simulation results. Nevertheless, our results show that the implementation can faithfully emulate the model dynamics for multiple synapses in parallel. The final judgment on the implementation of the plasticity rule depends on its applications, specifically modeling cognitive functions in computational neuroscience and machine learning. Our approach aims at exploiting the energy efficiency and acceleration factor of BrainScaleS-2 as well as its analog nature for the study and application of synaptic plasticity.**

***Index Terms*—neuromorphic, computational neuroscience, synaptic plasticity, reduced precision**

I. Introduction

Research on synaptic plasticity attempted to derive realistic plasticity rules that could potentially act as learning rules [1], [2], [3]. Rules that immediately link spikes to synaptic weights have gained widespread attention among the neuroscientific community due to their simplicity. However, experimental evidence suggests that learning rules are high-dimensional and cannot be reduced to spike timings alone [2]. As a result, mechanistic models that use biochemical messengers, primarily calcium, were proposed to better match experimental data [2], [4], [5].

Different challenges arise for the study of such plasticity rules. First, the long timescales of these rules result in extended simulation runtimes to capture the required dynamics [6], [7]. Second, the simulation of high-dimensional plasticity rules can be computationally demanding. Additionally, an efficient implementation using reduced-precision arithmetic is required for these high-dimensional plasticity rules in case they operate on edge devices for machine learning tasks [8].

Our goal is to employ neuromorphic hardware for providing accelerated and reliable implementations of spiking neural networks that utilize high-dimensional biologically-plausible plasticity rules. We rely on a calcium-based plasticity rule from [4] with synaptic weights evolving over multiple timescales. Compared to our previous work [9], we scale the implementation of this plasticity rule on BrainScaleS-2 (BSS-2) to neuron populations. Specifically, the contributions of this work are: (1) automating the mapping process of the calcium signal to the analog core of BSS-2 which links the neuron dynamics and spikes to the synaptic weight dynamics, and (2) using the single instruction, multiple data (SIMD) vector extension of the processor for updating the plasticity variables for 128 synapses in parallel. The mapping of the plasticity rule to a reduced precision SIMD can be applied to other architectures. Our code is available on github: https://github.com/electronicvisions/model-hw-memory-consolidation.

A. BrainScaleS-2

BrainScaleS-2 (BSS-2) is a mixed-signal neuromorphic platform that emulates neural dynamics at a tunable acceleration factor, typically 1000, compared to real time [10]. Its analog core encloses 512 neuron circuits that implement the adaptive exponential leaky integrate-and-fire (AdEx) model [11], [12]. Spikes are communicated as digital events on and off the chip.

Each neuron circuit can integrate stimuli from up to 256 plastic synapses. Synaptic plasticity is made possible by two SIMD processors that can access observables from the analog core and perform the required computations and parameter modifications [10], [13]. These processors, referred to as plasticity processing units (PPUs), implement the Power instruction set architecture (ISA) 2.06 [14] for 32 bit and a custom vector extension including 8 bit and 16 bit unsigned integer and signed saturating fractional arithmetics on 128 B vectors [15].

Each PPU interfaces with a 512-channel columnar analog-to-digital converter (CADC), which can perform simultaneous measurements of analog quantities across neurons or synapses. BSS-2 also features a fast membrane analog-to-digital converter (MADC) that is typically used for high-resolution, single-neuron investigations such as calibrating neuron or synapse parameters.

B. Synapse model

We are interested in the synapse model from [4] that integrates calcium dynamics with the synaptic tagging-and-capture (STC) hypothesis [16], [17], [18], [19]. In this model, the neuron dynamics follow the leaky integrate-and-fire (LIF) model. In addition, each synapse tracks a local calcium amount $c_{ji}(t)$. The total synaptic weight $w_{ji}(t)$ is divided into two components, the early-phase weight $h_{ji}(t)$ and the late-phase weight $z_{ji}(t)$. The update of the late-phase weight $z_{ji}(t)$ depends on a protein amount $p_i(t)$ of the postsynaptic neuron. All mentioned variables are governed by differential equations described in details in [4]. We will only mention here the equations that are useful for the proposed methods.

Given a synapse connecting a presynaptic neuron j and a postsynaptic neuron i, the calcium dynamics are governed by

$$\begin{aligned}\frac{\mathrm{d}c_{ji}(t)}{\mathrm{d}t} = &-\frac{c_{ji}(t)}{\tau_\mathrm{c}} + \mathrm{C}_\mathrm{pre}\sum_n \delta(t-t_j^n) \\ &+ \mathrm{C}_\mathrm{post}\sum_\mathrm{m}\delta(\mathrm{t}-\mathrm{t}_\mathrm{i}^\mathrm{m}),\end{aligned} \tag{1}$$

with the calcium time constant τ_c, the presynaptic spike times t_j^n, the postsynaptic spike times t_i^m, the contribution of presynaptic spikes C_pre, and the contribution of postsynaptic spikes C_post.

The calcium dynamics directly influence the early-phase weight $h_{ji}(t)$. The dynamics of early-phase weight are given by

$$\begin{aligned}\tau_h\frac{\mathrm{d}h_{ji}(t)}{\mathrm{d}t} = &\ 0.1\ (h_0 - h_{ji}(t)) \\ &+ \gamma_\mathrm{p}(h_\mathrm{max}-h_{ji}(t))\cdot\Theta[c_{ji}(t)-\theta_\mathrm{p}] \\ &- \gamma_\mathrm{d}(h_{ji}(t)-h_\mathrm{min})\cdot\Theta[c_{ji}(t)-\theta_\mathrm{d}],\end{aligned} \tag{2}$$

with $\Theta[\cdot]$ being the Heaviside function and τ_h being a time constant. The first term of eq. (2) describes the convergence of the early-phase weight to its initial value h_0, the second term describes early-phase long-term potentiation (LTP) with a rate γ_p for calcium concentration above the potentiation threshold θ_p, and the third term describes early-phase long-term depression (LTD) with a rate γ_d for calcium concentration above the depression threshold θ_d.

C. Previous Work

In our previous study [9], we emulated the plasticity rule for a single synapse and achieved a comparable accuracy to the simulation in [4]. Our approach utilized analog and digital signals that account for the different timescales summarized in Table I. Specifically, the neuron and calcium dynamics were emulated as analog signals on the analog core of BSS-2. Conversely, the synaptic weights and protein amount, which evolve on slower timescales, were computed as digital signals using the PPU. These plasticity variables were represented using $8\,\mathrm{bit}$ integers, and stochastic rounding (SR) was used to compensate for the resulting numerical errors. All operations in [9] were performed in their scalar form with SR operations on $32\,\mathrm{bit}$ integers. More details on the single synapse implementation are documented in our previous work [9] which is a foundation for the current work.

Model variable	Symbol	Timescale
Membrane potential	V	$\tau_\mathrm{mem} = 10\,\mathrm{ms}$ $\tau_\mathrm{syn} = 5\,\mathrm{ms}$
Calcium concentration	c	$\tau_\mathrm{c} = 48.8\,\mathrm{ms}$
Early-phase weight	h	$\tau_{h,\text{potentiation}} = 0.35\,\mathrm{s}$ $\tau_{h,\text{depression}} = 2.19\,\mathrm{s}$ $\tau_{h,\text{steady-state}} = 6884\,\mathrm{s}$
Protein amount	p	$\tau_\mathrm{p} = 3600\,\mathrm{s}$
Late-phase weight	z	$\tau_\mathrm{z} = 3600\,\mathrm{s}$

Table I. Biological timescales of the model variables from [4].

II. Methods

To build up on our previous work, we propose an automated method for mapping the calcium dynamics to the analog core of BSS-2 which include the time constant and the plasticity thresholds. Additionally, we implement a vectorized version of the plasticity rule that accounts for the constraints of the vector unit. In the following, a tilde symbol over a model variable (e.g., $\tilde{c}$) denotes its hardware realization on BSS-2.

A. Mapping calcium dynamics

Following the approach in [9], we configure the BSS-2 neurons as LIF neurons by disabling specific AdEx features while re-purposing the adaptation circuitry to track the calcium signal. The adaptation component of each neuron is used to compute the calcium component arising from the corresponding neuron's spikes. This is possible because the adaptation state can be decoupled from the neuron's membrane potential due to the modular design of the BSS-2 neuron circuits [12].

On BSS-2, the neuron tracks the adaptation state in the form of a voltage which is then converted to a current [12]. The adaptation state operates around an adaptation baseline voltage. Setting the subthreshold adaptation $a = 0$ and matching the adaptation equation of the AdEx model with eq. (1) yields

$$\frac{\mathrm{d}\tilde{c}_i(t)}{\mathrm{d}t} = -\frac{\tilde{c}_i(t)-\tilde{c}_{\text{baseline},i}}{\tilde{\tau_c}} + \tilde{b}\sum_f \delta(t-t_{\mathrm{f},i}), \tag{3}$$

where $\tilde{c}_i$ is the analog adaptation state emulating the calcium component of a neuron i, $\tilde{\tau_c}$ is the adaptation time constant $\tilde{c}_{\text{baseline},i}$ is the adaptation baseline voltage of neuron i, and $\tilde{b}$ is the spike-triggered adaptation.

Mapping $\tilde{\tau_c}$, $\tilde{b}$, and $\tilde{c}_{\text{baseline},i}$ of a neuron i to the desired calcium signal requires calibration on BSS-2. These parameters are interdependent, so calibrating each parameter requires tracking the other parameters. The choice of $\tilde{\tau_c}$ is directly related to the model where $\tilde{\tau_c} := \frac{\tau_c}{1000}$ due to the chosen acceleration factor of 1000. On the other hand, $\tilde{b}$ and $\tilde{c}_{\text{baseline},i}$ should be chosen to ensure linear behavior of the adaptation circuit, especially around the operating points $\tilde{\theta}_d$ and $\tilde{\theta}_p$.

Initially, $\tilde{c}_{\text{baseline},i}$ is calibrated to the same level for all neurons. To calibrate $\tilde{b}$ and $\tilde{\tau_c}$ for each neuron i, the calibration protocol is as follows: using eq. (1), we calculate the number of

spikes and interspike-interval required to elevate the adaptation potential to values around θ_d and θ_p at arbitrary adaptation parameters. The adaptation potential is then sampled using the MADC and is used to quantify the resulting spike-triggered adaptation potential $\tilde{b}$ and the adaptation time constant $\tilde{\tau}_c$. These parameters are compared with the target parameters and adjusted accordingly using a simple binary-search algorithm.

After using the MADC for calibration, we use the CADC to measure $\tilde{b}$ which corresponds to calcium influx resulting from a single spike. Therefore, the potentiation and depression thresholds are scaled by $\tilde{b}$ and can be computed respectively as

$$\tilde{\theta}_p = \theta_p \cdot \tilde{b} \tag{4}$$

$$\tilde{\theta}_d = \theta_d \cdot \tilde{b} \tag{5}$$

B. Scaling the plasticity rule

The SIMD extension on BSS-2 can perform addition, subtraction, and multiplication operations on $128 \times 8\,\mathrm{bit}$ integers or $64 \times 16\,\mathrm{bit}$ integers in parallel. In the previous work [9], SR relied on $32\,\mathrm{bit}$ pseudo-random numbers generated by a xor-shift pseudorandom number generator (PRNG) [20] and scalar operations which include division and operations on $32\,\mathrm{bit}$ integers. To scale the plasticity rule for 128 synapses using the vector unit, the following methods are adopted:

1) Division Operations: Right bit-shifts are used for implementing division operations. This implies that only division by integer powers of 2 can be performed. The resulting numerical error is corrected in the numerator at the beginning of each experiment run using SR.

2) Two-Stage Stochastic Rounding: BSS-2 does not have a $32\,\mathrm{bit}$ vector unit, and generating $128 \times 32\,\mathrm{bit}$ random integers sequentially is time-consuming. Therefore, a single $32\,\mathrm{bit}$ integer is used to apply SR for each vector of plasticity variables. In addition, the required update, which is interpreted as the rounding probability, depends on the plasticity variables [9], so we divide the rounding event into two independent events. Specifically, assume that an integer vector $\mathbf{X}$ is to be updated as

$$\mathbf{X}^{'} = \mathbf{X} + \Delta\mathbf{X} \quad \text{with} \quad \Delta\mathbf{X} < 1, \tag{6}$$

where the update $\Delta\mathbf{X}$ is small due to long timescales that result in small updates over time. The $32\,\mathrm{bit}$ stochastic integers generated by BSS-2 can only represent updates greater than 2^{-32}. In SR, $\Delta\mathbf{X}$ is interpreted as the probability of updating $\mathbf{X}$ which is rewritten as

$$\begin{aligned} \mathbf{X}^{'} &= \mathbf{X} + \Delta\mathbf{X} = \mathbf{X} + \mathrm{K} \cdot \delta\mathbf{X} \\ \text{with} \quad & 2^{-16} < \delta\mathbf{X} < 1 \quad \text{and} \quad \mathrm{K} < 1, \end{aligned} \tag{7}$$

where $\delta\mathbf{X}$ has a lower bound of 2^{-16} so that it can be represented by the $16\,\mathrm{bit}$ vector unit. On BSS-2, two $32\,\mathrm{bit}$ random numbers are drawn and two comparisons are performed:

1) A comparison for the $32\,\mathrm{bit}$ random number P_{32} against the required probability K which is independent of $\mathbf{X}$

$$P_{32} < \Theta_{\mathbf{K}} \quad \text{with} \quad \Theta_{\mathbf{K}} = \mathrm{MAX\_UINT}_{32} \cdot \mathrm{K}.$$

2) A comparison of the vector against the upper 16 bits of the second random number P_{16} for higher quality of random integers

$$P_{16} < \Theta_{\mathbf{X}} \quad \text{with} \quad \Theta_{\mathbf{X}} = \mathrm{MAX\_UINT}_{16} \cdot \delta\mathbf{X}.$$

This yields a final update result for $\mathbf{X}$ as

$$\mathbf{X}' = \begin{cases} \mathbf{X} + 1 & P_{32} < \Theta_{\mathbf{K}} \quad \text{and} \quad P_{16} < \Theta_{\mathbf{X}} \\ \mathbf{X} & \text{otherwise.} \end{cases} \tag{8}$$

This decoupling is possible for all the probabilities computed in [9].

C. Stimulation protocols

To assess our implementation of the plasticity rule on BSS-2, we use standard stimulation protocols that induce different forms of plasticity for single synapses [4]. The first protocol is the strong tetanic stimulation (STET) protocol which results in late LTP characterized by an increase in the late-phase weight z for a stable change in synaptic efficacy. The second protocol is the strong low-frequency stimulation (SLFS) protocol which results in late LTD characterized by a decrease in the late-phase weight z.

III. Results

A. Mapping calcium parameters

We perform calibration of the adaptation parameters for 128 neurons. Figure 1-A presents the results of the calibration scheme showing an example calcium trace of a neuron stimulated with a different number of spikes. The desired parameters are $\tilde{b} = 60$ in MADC units and $\tilde{\tau}_c = 48.8\,\mu\mathrm{s}$. The plasticity rule is expected to operate linearly around the thresholds $\tilde{\theta}_p$ and $\tilde{\theta}_d$, so we assess the stability of the parameters $\tilde{b}$ and $\tilde{\tau}_c$ depending on the number of spike inputs in Figure 1-B. We choose the inter-spike interval and the number of spike inputs such that the adaptation potential exceeds $\tilde{\theta}_d$ and $\tilde{\theta}_p$. The distribution of $\tilde{\theta}_d$ and $\tilde{\theta}_p$ for 128 neurons is shown in Figure 1-C. The spread of $\tilde{\theta}_p$ compared to $\tilde{\theta}_d$ is attributed to imperfection in circuit linearity with the elevation of the adaptation potential.

B. Scaling the plasticity rule

We run the plasticity rule for $12\,\mathrm{s}$ on BSS-2 which amounts to $200\,\mathrm{min}$ in biological time using the vectorization methods in section II-B and the stimulation protocols in section II-C. The experiment is repeated for 100 different spike trials, and the plasticity rule is executed every $50\,\mu\mathrm{s}$. The behavior of the synaptic weights aligns with the expected biological behavior where STET induces late long-term potentiation and SLFS induces late long-term depression, see fig. 2.

Table II compares the simulation and emulation results numerically. The final value of the late-phase weight z is used as a statistic, since it is a measure of the stability of synaptic weight, and its computation is affected by all neuron and synapse variables. The simulation at an update time step of $0.2\,\mathrm{ms}$ was performed in [4]. On the other hand, the simulation

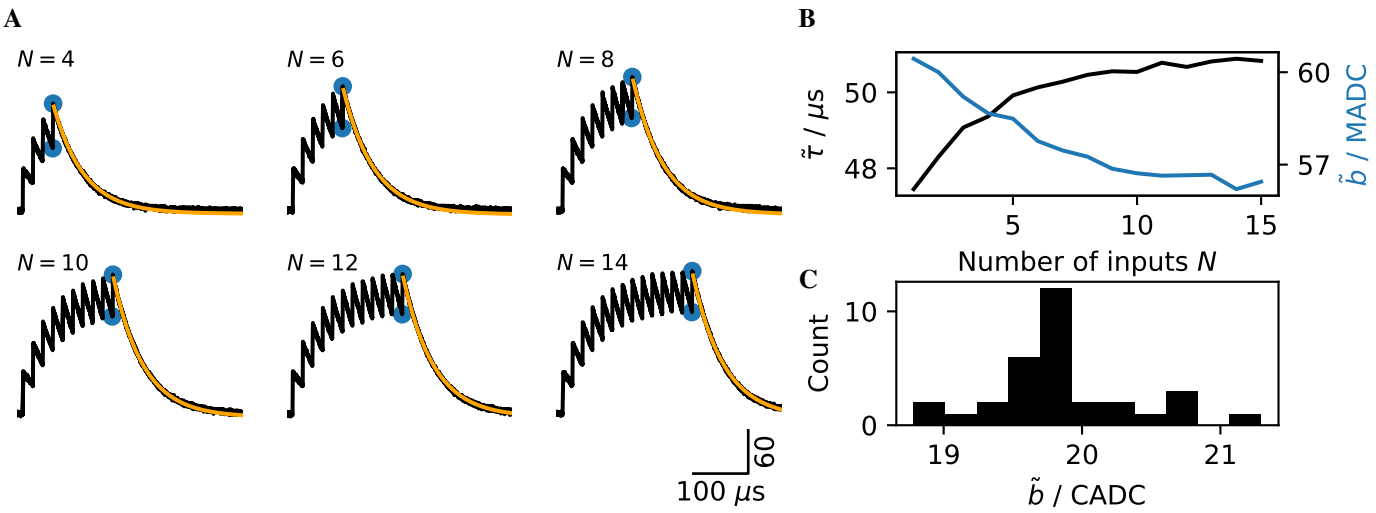

Figure 1: Calibration results of adaptation parameters for 128 neurons. (A) The calibration protocol consists of different periods. In each period, a number of spikes are injected. The spike-triggered adaptation $\tilde{b}$ is quantified as the jump after the last spike input (see blue dots). The adaptation time constant is computed using the final decay of the adaptation potential in each period (orange line). Data is measured using the MADC. (B) The variation of the hardware parameters depending on the number of inputs. (C) The histogram shows results for measuring $\tilde{b}$ in CADC units for 128 neurons on BSS-2 which will be used for mapping the potentiation and depression thresholds $\tilde{\theta}_p$ and $\tilde{\theta}_d$. The mean thresholds are used as common thresholds for all neurons.

Protocol	Simulation		Emulation (50 µs)	
	0.2 ms	50 ms	Mean	Standard deviation
STET	0.74 ±0.02	0.74 ±0.02	0.744 ±0.003	0.058 ±0.001
SLFS	−0.29 ± 0.06	−0.3 ±0.1	−0.24 ±0.02	0.110 ±0.004

Table II. Average value and standard deviation of the final late-phase weight z across 100 spike trials. The simulation is performed for two time steps of the numerical solver. The emulation results are reported for 16 synapses obtained in parallel in the form of ensemble statistics. The statistics of the emulation correspond to the grand mean and pooled standard deviation of the late-phase weight.

at a time step of 50 ms agrees with the update time step used in the emulation. The results show that the average in the final late-phase weight of the emulation is closely aligned to that of the simulation, yet exhibits a systematic error. The systematic error can be attributed to the dependence of the adaptation time constant of the current adaptation potential, leading to more accumulation of calcium at higher calcium values. Additionally, the variance in the emulation is higher than the simulation due to the analog nature of the hardware and SR.

Figure 2 visualizes 16 synapses although the plasticity rule can execute weight updates for 128 synapses in parallel. This is due to the tradeoff between the execution period of the plasticity rule and the number of observables recorded per period explained in the following section.

C. Runtime analysis

We assess the time required by the PPU to execute the plasticity rule. The execution time includes sampling of

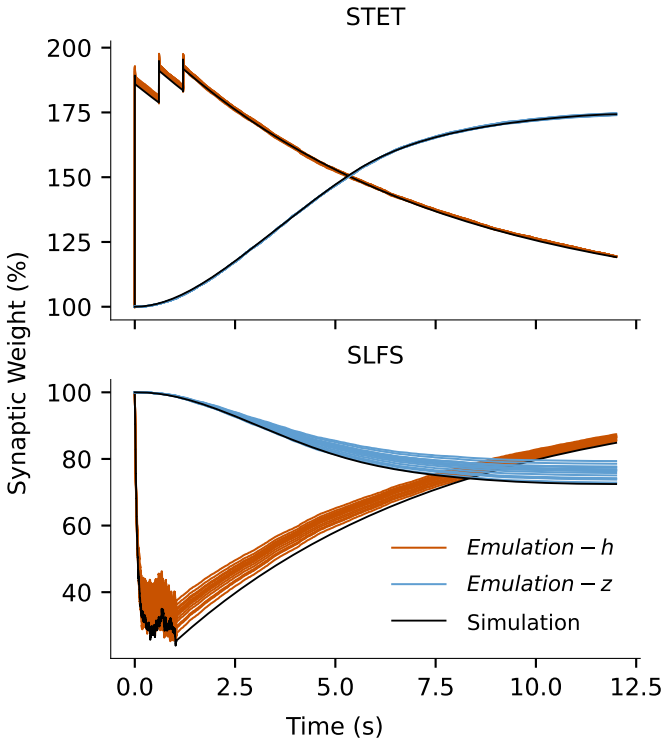

Figure 2: Emulation results for 16 synapses plotted versus simulation results obtained at a numerical timestep of 50 ms and 100 different spike trials. The simulation results are plotted at an accelerated factor of 1000 to match the timescale of the emulation results. The results show that emulation results achieve the desired behavior in terms of long-term potentiation and long-term depression for the two stimulation protocols.

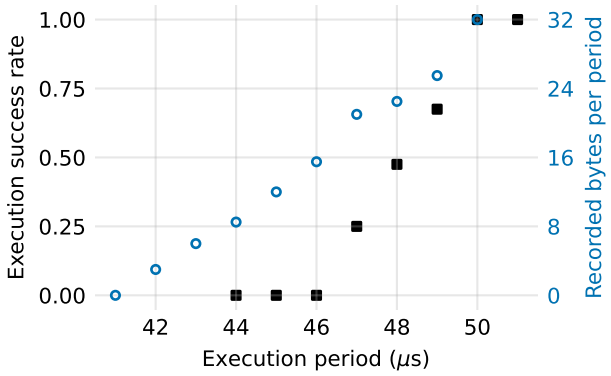

Figure 3: Average runtime of the plasticity rule on the plasticity processing unit assessed on four chips. The success rate is displayed for recording 32 B and is measured as the average success of ten runs of the experiment for a specific execution period. In a different experiment, the recorded bytes are measured as the average number of bytes that can be recorded in ten successful executions at different execution periods.

adaptation traces, numerical computations of the plasticity variables, update of synaptic weights which includes access to the synaptic weight memory, and recording of observables every period.

'Execution success' is defined as the plasticity rule adhering to the scheduled execution time. In other words, a successful execution of the plasticity rule means the PPU was able to execute all instructions within the allocated execution period. Here, we run two different experiments: the first experiment consists of running the plasticity rule while recording 32 B at different execution periods and calculating the execution success rate. The second experiment consists of running the plasticity rule at different periods while recording the maximum number of bytes that allows ten successful executions.

Figure 3 shows runtime statistics collected for four chips over ten experiment runs. The plasticity rule requires on average 50 µs to run while recording the weights of 16 synapses. The recording of the early-phase and late-phase weights for 16 synapses which amounts to 32 B requires on average 9 µs.

IV. Discussion

In this work, we parallelized the emulation of a calcium-based plasticity rule on the analog neuromorphic BrainScaleS-2 platform. With this implementation, we aim at fast and energy-efficient experiments of spiking neural networks whose synapses follow high-dimensional plasticity rules. Scaling required two steps: a calibration of the analog circuits to match the required calcium dynamics, and a vectorized implementation of the plasticity rule that exploits the SIMD vector extension of the embedded processor.

The analog circuits of BSS-2 were designed to match biologically-plausible parameters [12]. In addition, the flexibility of the neuron circuits of BSS-2 allows their usage and calibration to a wide range of parameter targets. This made it possible to map the calcium dynamics of the model in [4] to the adaptation circuit of the AdEx neuron implementation on BSS-2. The calibration results of 128 neurons revealed the configurability and the robustness of these circuits. As in any other analog circuit, BSS-2 is not perfectly linear at different operating points. Nevertheless, using a robust model and averaging across different experiment-runs cancels out these effects.

Implementing the plasticity rule with long timescales using reduced-precision arithmetic required stochastic rounding. This made the vectorized implementation challenging, therefore SR was done in two steps. The results showed numerical results comparable to those of the simulation and achieved the desirable biological behavior for potentiation and depression. The mismatch in the mean and the high variance in the synaptic weights in the emulation results can be attributed to multiple reasons. First, the plasticity rule is updated at a relatively large timestep due to the speed of the embedded processor. Second, the analog calcium signal adds a source of variability to the computations, specifically the time constant dependence on the adaptation potential. Third, using reduced-precision arithmetic instead of floating point operations as in the simulations limits the achievable precision. Despite these constraints, the implementation of the plasticity rule was consistent for different synapses.

A downside of our implementation is the use a single integer for SR which might introduce a correlation bias between synapses. When emulating several synapses in a neural network, we expect that this effect is reduced due to the different states of synapses. The current implementation exploits the full SIMD parallelism of BSS-2, i.e. operating on 128 B in parallel. For large networks, it is possible to loop over the instructions and increase the execution period of the PPU to handle a larger number of synapses.

The timestep of the numerical solver in simulations of spiking neural networks usually falls in the fraction of milliseconds [4], [6]. Due to the limited speed of the processor on BSS-2, the average time required by the plasticity rule in our implementation is 50 µs, which corresponds to 50 ms in the biological domain. Despite this limitation, the obtained results were stable due to the long timescales in the model and the dependence of the plasticity variables on thresholds. We also used an acceleration factor of 1000, so the 3 h biological experiment was performed in 12 s plus an additional time for the readout of observables. Additionally, the variables with fast timescales are emulated as analog signals, so there is no information loss arising from time discretizations. It is worth to note that the long timescales of the observables that could be computed on the PPU of BSS-2 are limited by their ability to be represented in reduced precision arithmetic.

In addition to the time requirements for sampling, computing the weight updates, and writing the weight to hardware, recording the observables slows down the execution of the plasticity rule. This imposes a constraint on how many variables can be recorded during experiments. However, recording all observables might not be necessary. For example, the measures of memory consolidation in [4] primarily rely on the firing

rate of neurons. The recording method could also be optimized in future work such that observables are recorded on a coarser time grid or in a flexible manner depending on the plasticity rule.

The success of this emulation is to be judged by future implementations of spiking neural networks whose synapses obey such plasticity rules on BSS-2. Applications range from modeling cognitive functions in neuroscience, such as in [4], to learning tasks on sequence data. As mentioned above, BSS-2 has a tunable acceleration factor depending on the chosen time parameters. This implies that the limitations of our current approach, such as recording of observables and execution time of the plasticity rule, can also be reduced by slowing down the model dynamics. A longer runtime would be required, but the speedup from the acceleration factor of BSS-2 and energy efficiency would still be advantageous for our work.

V. Acknowledgments

This work stands on the shoulders of the Electronic Visions research group; we are grateful to all past and present members for the years of hard work and persistence to build BrainScaleS-2 neuromorphic platform. *Funding:* This research has received funding from the EC Horizon 2020 Framework Programme under grant agreements 785907 (HBP SGA2), 945539 (HBP SGA3), the EC Horizon Europe Framework Programme under grant agreement 101147319 (EBRAINS 2.0), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1-390900948 (the Heidelberg STRUCTURES Excellence Cluster).

References

[1] P. Mateos-Aparicio et al., "The impact of studying brain plasticity," *Front. cell. neurosci.*, vol. 13, 2019. DOI: 10.3389/fncel.2019.00066.

[2] H. Z. Shouval et al., "Spike timing dependent plasticity: A consequence of more fundamental learning rules," *Frontiers in computational neuroscience*, vol. 4, 2010.

[3] C. Tetzlaff et al., "Time scales of memory, learning, and plasticity," *Biol Cybern*, vol. 106, no. 11-12, pp. 715–726, 2012.

[4] J. Luboeinski et al., "Memory consolidation and improvement by synaptic tagging and capture in recurrent neural networks," *Commun. Biol.*, vol. 4, no. 1, pp. 1–17, Mar. 2021. DOI: 10.1038/s42003-021-01778-y.

[5] C. Clopath et al., "Tag-trigger-consolidation: A model of early and late long-term-potentiation and depression," *PLoS Computational Biology*, vol. 4, no. 12, 2008. DOI: 10.1371/journal.pcbi.1000248.

[6] F. Zenke et al., "Limits to high-speed simulations of spiking neural networks using general-purpose computers," *Front. Neuroinform.*, vol. 8, no. 76, 2014. DOI: 10.3389/fninf.2014.00076.

[7] J. Luboeinski et al., *Plastic arbor: A modern simulation framework for synaptic plasticity – from single synapses to networks of morphological neurons*, 2024.

[8] M. Croci et al., "Stochastic rounding: Implementation, error analysis and applications," *R. Soc. Open Sci.*, vol. 9, no. 3, p. 211 631, 2022. DOI: 10.1098/rsos.211631.

[9] A. Atoui et al., "Multi-timescale synaptic plasticity on analog neuromorphic hardware," in *2025 Neuro Inspired Computational Elements (NICE)*, 2025, pp. 1–9. DOI: 10.1109/NICE65350.2025.11065914.

[10] C. Pehle et al., "The BrainScaleS-2 accelerated neuromorphic system with hybrid plasticity," *Front. Neurosci.*, vol. 16, 2022. DOI: 10.3389/fnins.2022.795876.

[11] R. Brette et al., "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *J. Neurophysiol.*, vol. 94, pp. 3637–3642, 2005. DOI: 10.1152/jn.00686.2005.

[12] S. Billaudelle et al., "An accurate and flexible analog emulation of AdEx neuron dynamics in silicon," in *ICECS*, 2022, pp. 1–4. DOI: 10.1109/ICECS202256217.2022.9971058.

[13] P. Spilger et al., "Integrating programmable plasticity in experiment descriptions for analog neuromorphic hardware," in *2025 Neuro Inspired Computational Elements (NICE)*, 2025, pp. 1–8. DOI: 10.1109/NICE65350.2025.11065886.

[14] PowerISA, "PowerISA version 2.06 revision b," Power.org, Specification, Jul. 2010.

[15] S. Friedmann et al., "Demonstrating hybrid learning in a flexible neuromorphic hardware system," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 1, pp. 128–142, Feb. 2017. DOI: 10.1109/TBCAS.2016.2579164.

[16] U. Frey et al., "Synaptic tagging and long-term potentiation," *Nature*, vol. 385, no. 6616, pp. 533–536, 1997. DOI: 10.1038/385533a0.

[17] R. L. Redondo et al., "Making memories last: The synaptic tagging and capture hypothesis," *Nat Rev Neurosci*, vol. 12, no. 1, pp. 17–30, 2011. DOI: 10.1038/nrn2963.

[18] A. Prodan et al., "Synaptic tagging and capture: Functional implications and molecular mechanisms," in *Synaptic Tagging and Capture: From Synapses to Behavior*, S. Sajikumar et al., Eds., 2nd, Cham, Switzerland: Springer, 2024, ch. 1, pp. 1–41. DOI: 10.1007/978-3-031-54864-2_1.

[19] J. Luboeinski et al., "Modeling emergent dynamics arising from synaptic tagging and capture at the network level," in *Synaptic Tagging and Capture: From Synapses to Behavior*, S. Sajikumar et al., Eds., 2nd, Cham, Switzerland: Springer, 2024, ch. 23, pp. 471–503. DOI: 10.1007/978-3-031-54864-2_23.

[20] G. Marsaglia, "Xorshift RNGs," *Journal of Statistical software*, vol. 8, no. 14, pp. 1–6, 2003. DOI: 10.18637/jss.v008.i14.