Robustness from structure: Inference with hierarchical spiking networks on analog neuromorphic hardware

Mihai A. Petrovici[†‡] Anna Schroeder[†]
Oliver Breitwieser[†] Andreas Grübl[†] Johannes Schemmel[†] Karlheinz Meier[†]
{mpedro, annasch, obreitwi, agruebl, schemmel, meierk}@kip.uni-heidelberg.de
[†] Heidelberg University, Kirchhoff-Institute for Physics, Im Neuenheimer Feld 227, D-69120 Heidelberg
[‡] University of Bern, Department of Physiology, Bühlplatz 5, CH-3012 Bern

***Abstract*—How spiking networks are able to perform probabilistic inference is an intriguing question, not only for understanding information processing in the brain, but also for transferring these computational principles to neuromorphic silicon circuits. A number of computationally powerful spiking network models have been proposed, but most of them have only been tested, under ideal conditions, in software simulations. Any implementation in an analog, physical system, be it *in vivo* or *in silico*, will generally lead to distorted dynamics due to the physical properties of the underlying substrate. In this paper, we discuss several such distortive effects that are difficult or impossible to remove by classical calibration routines or parameter training. We then argue that hierarchical networks of leaky integrate-and-fire neurons can offer the required robustness for physical implementation and demonstrate this with both software simulations and emulation on an accelerated analog neuromorphic device.**

I. Introduction

Over the past decades, research in neural networks has undergone an interesting branching process. On the one hand, the machine learning community has gradually increased its interest in what were originally brain-inspired neural networks. These efforts have been crowned by impressive recent success [1], [2], which has, however been obtained at the price of having strayed away from biologically plausible dynamics. On the other hand, modern computational neuroscience is pushing for ever more complex and biologically realistic simulations, in the hope to uncover the biological details of information processing in the brain [3]. Today, these two communities are investigating network models that have little in common with each other.

In the meantime, the neuromorphic community has to master an increasingly difficult balancing act. At its core, the neuromorphic approach aims to mimic various features of the neocortex in silico. For example, an essentially ubiquitous feature of neuromorphic devices is that they are built to emulate spiking neurons [4]–[9]. However, one core argument for building these devices is the hope to use them to unlock the brain's computational power by moving beyond the von Neumann computing paradigms. Consequently, a driving question for the neuromorphic community might be formulated as follows: is it possible to find relevant applications for spiking neural networks that can then profit from the typical advantages of a physical implementation such as inherent parallelism, high speed and low power consumption? The findings discussed in this article suggest a promising path towards finding an answer.

This issue is even more pronounced in the case of analog hardware, since it imposes additional constraints that stem from the physics of systems themselves. As opposed to digital systems, be they von Neumann or neuromorphic, which have the benefit of essentially perfect precision and control, analog systems have to deal with inherent imperfections. These imperfections concern, on the one hand, the *equations of motion* of the network components, which must obey the physics of the substrate and can therefore only provide an approximation of the target dynamics. On the other hand, the degree of precision to which the *parameters* of these equations can be tuned certainly depends on the hardware design, but is always fundamentally limited by fixed-pattern variations and temporal noise. Imperfections in the network dynamics and parameters necessarily distort the behavior of the emulated networks, which usually impairs their performance to some degree [10].

The question of parameter control (i.e., calibration, post-calibration tuning and training) is an essential one. It constitutes a perennial challenge for analog neuromorphic system design and operation, and has therefore been often addressed in literature [10]–[13]. A thorough discussion of parameter calibration and in-the-loop training of analog circuits can be found in [14], which represents a complement of the present study. In the present manuscript, we are mainly concerned with the distortions to the network dynamics that are imposed by the physics of the emulation device and that cannot be directly addressed by, e.g., calibration.

We begin by identifying a Bayesian spiking network model with valuable computational properties (Sec. II). It is able to learn a probabilistic model of input data and can subsequently be used as both a generative and a discriminative model — a feature that is difficult to achieve even with abstract neural networks [15]. Here, we focus on its discriminative properties. This model is, in general, susceptible to hardware-

978-1-5090-6182-2/17/$31.00 ©2017 European Union

induced distortions, as we discuss in detail in Sec. III. As a particular example, we characterize these effects on the Spikey chip [5] — the neuromorphic system that we use as an emulation back-end. Despite our model's ostensible lack of robustness, we argue that, when endowed with a particular, hierarchical connectivity structure, it becomes robust to the studied hardware-induced distortions. We substantiate this conjecture with both software simulations (Sec. IV) and hardware emulations (Sec. V). In particular, without any training to compensate for parameter noise on the hardware, we show that the network only loses a relatively small fraction of its initial performance when running on Spikey. This represents, to our knowledge, the first scalable implementation of a hierarchical probabilistic network in accelerated analog neuromorphic hardware.

II. LIF-based Boltzmann machines

Continued technological advances in large-scale processing (parallel CPU and GPU architectures) have enabled the recent resurgence of artificial neural networks. Already envisioned for decades as theoretical models of brain-like architectures [16], [17], neural networks now routinely outperform their rival models at pattern recognition tasks [1]. Here, we focus on one particular neural network model — a spiking variant of the Boltzmann machine (BM) [18], which has been shown to be compatible with biologically plausible and hardware-implementable spiking neurons [19]. We now briefly describe the structure and dynamics of these stochastic networks of leaky integrate-and-fire (LIF) neurons and discuss potential problems that can arise from their implementation in analog hardware.

In the neural sampling framework [20], a population of n neurons represents a binary random vector $\boldsymbol{z} \in \{0,1\}^n$. The refractory state of a neuron following a spike at time t_s is chosen to represent the 1-state of the associated random variable (see also Fig. 1A):

$$z_k^{(t)} = \begin{cases} 1 & \text{if } t_s < t < t_s + \tau_{\text{ref}} \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

In the abstract model of neural sampling, the probability of each neuron to be in the 1-state is given by a logistic activation function

$$p(z_k = 1 | \boldsymbol{z}_{\setminus k}) = \sigma(u_k^{\text{abstr}}) = \frac{1}{1 + e^{-u_k^{\text{abstr}}}} . \tag{2}$$

Such an abstract neuron's membrane potential u_k^{abstr} has a resting-state value of b_k and is linearly influenced by the state $\boldsymbol{z}_{\setminus k}$ of all other neurons in the network via (symmetric) synaptic weights $W_{kj} = W_{jk}$:

$$u_k^{\text{abstr}}(\boldsymbol{z}_{\setminus k}) = \sum_{j=1}^{n} W_{kj} z_j + b_k . \tag{3}$$

It can then be shown that, under these assumptions, a network of such stochastic neurons will sample from a Boltzmann distribution

$$p(\boldsymbol{z}) = \frac{1}{Z} \exp[-E(\boldsymbol{z})] = \frac{1}{Z} \exp\left[\frac{1}{2}\boldsymbol{z}^T \boldsymbol{W} \boldsymbol{z} + \boldsymbol{z}^T \boldsymbol{b}\right] , \tag{4}$$

with the partition function Z as a normalization factor.

In order to achieve similar dynamics with LIF neurons, an equivalent firing regime needs to be established. In the LIF sampling framework [19], each neuron receives two kinds of spiking input: information-encoding input from other neurons in the network and diffuse background input that represents the source of stochasticity, modeled by Poisson sources. These input spike trains generate two types of current onto the membrane, which we denote by I^{syn} and I^{noise}, respectively:

$$C_{\text{m}} \frac{d}{dt} u_k = g_{\text{l}}(E_{\text{l}} - u_k) + I_k^{\text{syn}} + I_k^{\text{noise}} + I_k^{\text{ext}} . \tag{5}$$

Here, C_{m} is the membrane capacitance, g_{l} and E_{l} are the leak conductance and potential, and I_k^{ext} is an external current that determines the bias b_k. While in general noisy LIF neurons do not have a logistic activation function, as required in Eqn. 2, it has been shown that in a high-conductance state the LIF activation function can be well approximated by a logistic function that is scaled with parameters α and $\bar{u}_k^0$ [19], [21]:

$$p(z_k = 1 | \boldsymbol{z}_{\setminus k}) = \sigma\left(\frac{\bar{u}_k - \bar{u}_k^0}{\alpha}\right) , \tag{6}$$

where $\bar{u}_k = \langle u_k \rangle_t$ represents the noise-free membrane potential of the kth neuron. This equivalence to the abstract model enables an LIF neuron to sample correctly from its conditional distribution $p(z_k | z_{\setminus k})$. The translation of the Boltzmann parameters $(\boldsymbol{W}, \boldsymbol{b})$ in Eqn. 4 to the conductance-based LIF domain (synaptic weights $\boldsymbol{w}$, bias potentials $\bar{\boldsymbol{u}}^0$) can then be achieved using the following rules:

$$b_k = (\bar{u}_k^b - \bar{u}_k^0)/\alpha \tag{7}$$

$$W_{kj} = \frac{1}{\alpha C_{\text{m}}} \frac{w_{kj}\left(E_{kj}^{\text{rev}} - \mu\right)}{\frac{1}{\tau^{\text{syn}}} - \frac{1}{\tau_{\text{eff}}}} \times \left[\frac{1-e}{e} - \frac{\tau_{\text{eff}}}{\tau^{\text{syn}}}\left(e^{-\frac{\tau^{\text{syn}}}{\tau_{\text{eff}}}} - 1\right)\right] , \tag{8}$$

where μ is the mean of the free membrane potential, $\tau_{\text{eff}} = C_{\text{m}}/(g_{\text{l}} + g^{\text{syn}})$ is the effective membrane time constant in the high-conductance state, g^{syn} the total synaptic conductance, τ^{syn} the synaptic time constant and E^{rev} the synaptic reversal potential. This allows a direct mapping of abstract BMs to networks of LIF neurons that sample accurately from their target distribution (Fig. 1B-D).

It is important to note that such networks are not merely more complicated replicas of classical machine learning approaches. In addition to being able to emulate the computational power of traditional Boltzmann machines, these spiking networks can also harness certain biological mechanisms to extend their functionality. It has, for example, been shown, that when endowed with short-term synaptic plasticity, LIF-based BMs can become good generative models of their learned datasets, while at the same time maintaining a high classification performance when presented with individual data samples [15].

Such LIF networks are now amenable to training with any of the established algorithms for BMs. While backpropagation [22] is more difficult to implement in a biologically

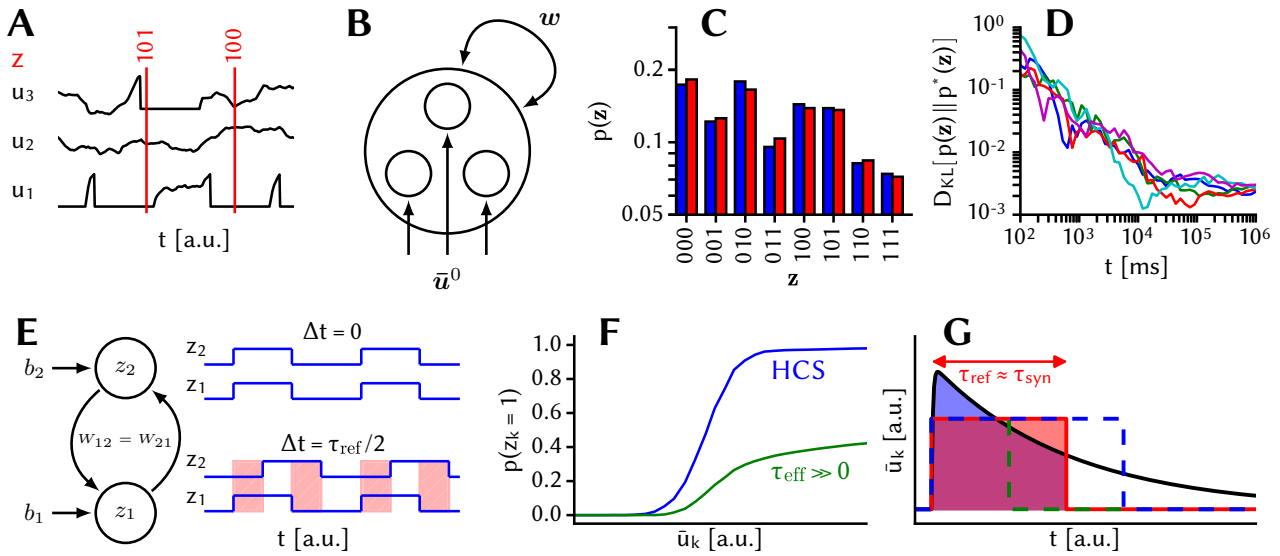

Fig. 1. Sampling with LIF neurons and distortions induced by implementation in a physical substrate. **(A)** Exemplary membrane potentials from a sampling network of LIF neurons. Each neuron has an associated random variable $z_k \in \{0, 1\}$ which is equal to 1 when the neuron is refractory. **(B)** Schematic of a recurrent network of LIF neurons in the HCS with a symmetric synaptic weight matrix $\boldsymbol{w}$ and bias potential vector $\bar{\boldsymbol{u}}^0$, which approximates a Boltzmann machine (C, D) with parameters given by Eqns 7 and 8. **(C)** Exemplary state distribution of a 3-neuron network: sampled distribution $p(\boldsymbol{z})$ (blue) vs. target distribution $p^*(\boldsymbol{z})$ (red) after 10^3 ms. **(D)** Evolution of the Kullback-Leibler divergence $D_{\mathrm{KL}}\left(p \parallel p^*\right)$ over time. Multiple runs with different random seeds are marked with different colors. **(E)** Synaptic transmission delays change temporal correlations between the states of different neurons. In this example, we consider two neurons connected with large excitatory weights $w_{12} = -2b_1 = -2b_2$. Without delays ($\Delta t = 0$, top), the network samples correctly from its target distribution $p(0, 0) = p(1, 1) \approx 0.5$, $p(0, 1) = p(1, 0) \approx 0$. With relatively large delays ($\Delta t = \tau_{\mathrm{ref}}/2$, bottom), the sampled distribution becomes completely different, with $p(0, 1) \gg 0$ and $p(1, 0) \gg 0$. The wrongly sampled mixed states, marked in red, are a direct consequence of the synaptic transmission delays. **(F)** An imperfect high-conductance state ($\tau_{\mathrm{eff}} \gg 0$) leads to a deviation of the neuronal activation function from its ideal logistic shape. This modifies the sampled distribution by reducing the probability of neurons to spike, especially for positive biases. **(G)** Refractory times and synaptic time constants are coupled to ensure that the average interaction between neurons during refractoriness (blue-shaded PSP in the LIF model) has the correct amplitude (red-shaded rectangular PSP in the abstract model), given in Eqn. 8. Spike-to-spike variability of refractory times disrupts this equivalence (green and blue dashed lines), effectively modifying the interaction strength $\boldsymbol{W}$.

plausible network, other methods exist that are more compatible with Hebbian learning. The wake-sleep algorithm, in particular, requires each synapse to only have access to the activity of its pre- and its postsynaptic neuron [23]:

$$\Delta w_{ij} = \eta(\langle z_i z_j \rangle_{\mathrm{data}} - \langle z_i z_j \rangle_{\mathrm{model}}) \quad , \tag{9}$$

$$\Delta b_i = \eta(\langle z_i \rangle_{\mathrm{data}} - \langle z_i \rangle_{\mathrm{model}}) \quad . \tag{10}$$

This learning rule tries to adapt the activity $\boldsymbol{z}_{\mathrm{model}}$ of the network in the "dreaming" phase, during which it evolves freely, to its activity $\boldsymbol{z}_{\mathrm{data}}$ in the "awake" state, where it is constrained by data, i.e., where some of the units are clamped to particular values. Despite its simplicity, this learning algorithm can be used to achieve high classification rates on various machine learning datasets [24].

III. Critical Distortions in Physical Implementations

The distribution that an LIF network samples from is uniquely determined by the neuro-synaptic dynamics and parameters. Any deviation from the model specification will alter the sampled distribution and, in general, restrict the network's ability to perform correct inference in the learned sample space. Mapping this model to an imperfect physical substrate is therefore not straightforward. In this article, we study three types of distortions of network dynamics that are caused by mapping to an analog silicon substrate.

First, we consider spike transmission delays. Since we are using point neurons, we can describe all delays as being synaptic delays. Many analog neuromorphic devices, including the one we use later on, are mixed-signal systems, meaning that spikes are transmitted digitally. Consequently, digitization, transport of the digital data, and the conversion back to the analog domain in the synapses contribute to synaptic delays. While these delays may be short in terms of wall-clock time, they become particularly critical in accelerated systems. In such systems, the neuronal and synaptic dynamics that define the characteristic time scale on which the network evolves can be orders of magnitude smaller than in biology, potentially entering the range of synaptic transmission delays [10]. Regardless of the exact nature of a network performing neural sampling, in order for each neuron to be able to calculate its correct conditional distribution $p(z_k|\boldsymbol{z}_{\setminus k})$, the information gathered by a neuron from its incoming PSPs must coincide with the true state of the corresponding presynaptic neurons, as required by Eqn. 3. This temporal coincidence is disrupted by delays, which thereby distort the sampled distribution, as exemplified in Fig. 1E.

Second, while most neuromorphic systems have controllable neuron and synapse parameters, these can only be configured within a certain range and resolution. As an example, consider the membrane time constant τ_{m} of a neuron. This

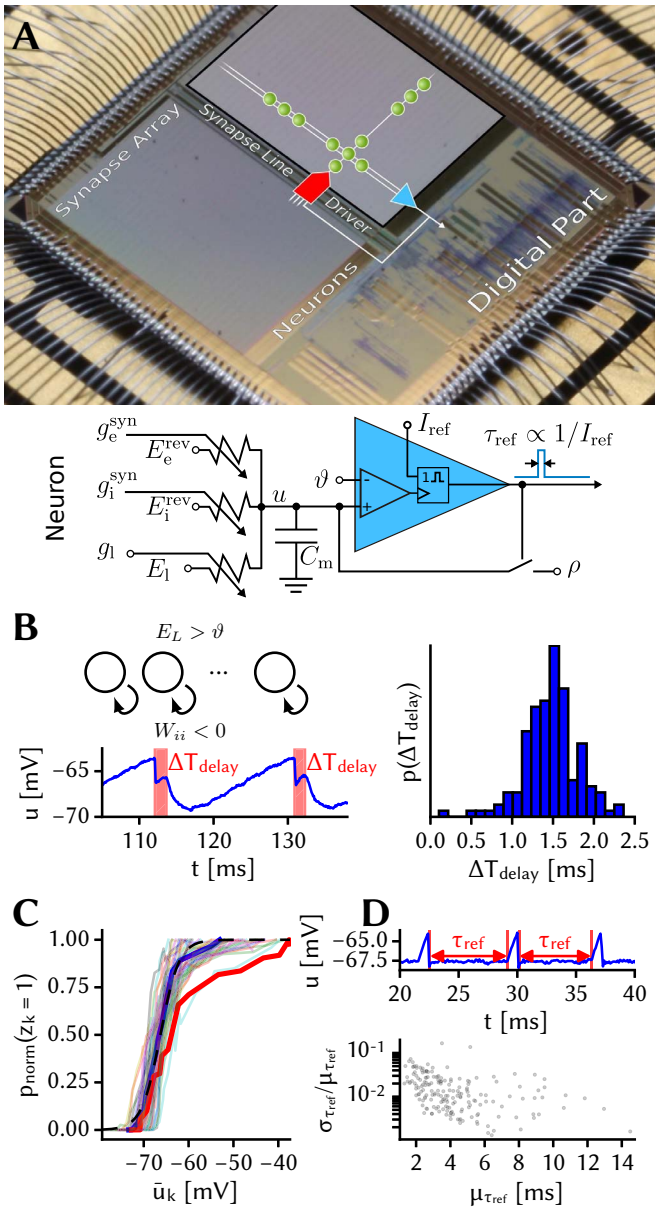

Fig. 2. Characterization measurements for the employed neuromorphic system. **(A)** *Top:* Neuromorphic Spikey chip with overlaid sketch of neural network components (taken from [5]). *Bottom:* Simplified schematic of a single neuron. **(B)** *Top left:* Synaptic delays were measured by recurrently connecting each neuron to itself through an inhibitory synapse. *Bottom left:* The relatively sharp onset of the inhibitory PSP allows a precise measurement despite the temporal noise on the membrane potential. *Right:* Synaptic delay distribution of 192 neurons for a single synapse driver. **(C)** Activation functions of 44 Spikey neurons (thin solid lines) compared to the nearly ideal logistic activation function achieved in the high-conductance state (dashed line). Two exemplary activation functions are drawn with thicker lines: the blue and red activation functions belong to neurons with short and long τ_{eff}, respectively. **(D)** *Top:* Refractory times were measured by choosing a suprathreshold leak potential for all neurons and subtracting the reset-to-threshold first passage time from the interspike interval. *Bottom:* Relative spike-to-spike variability of the refractory time vs. mean refractory time for 192 Spikey neurons.

time constant can be considered to define the reaction speed of a neuron to external stimuli. In neuromorphic systems, τ_{m} is usually configured with an adjustable leak conductance, which can not become arbitrarily large. Such a limit in the reaction speed of neurons can impair the functionality of the entire network. For LIF neurons, a large τ_{m} slows the saturation of the activation function (Fig. 1F) and thereby distorts the logistic shape (Eqn. 6) required for sampling from Boltzmann distributions.

Third, temporal noise can also affect computation. Depending on the particular in-silico implementation, any analog system will be subject to some degree of temporal noise on all of its electronic signals, including those that directly influence neuro-synaptic dynamics. In our particular case, the largest temporal noise component affects refractory times τ_{ref}. Since the relevant neuron and synaptic circuits are physically disconnected on the chip, the spike-to-spike variation of τ_{ref} is independent from the synaptic time constant, which can be considered as fixed. Consequently, the state $z_k(t)$ of a neuron will not coincide anymore with the information it transmits via PSPs to its postsynaptic partners (Fig. 1G), leading to a distortion of the sampled probability distribution in a conceptually similar manner as synaptic delays do.

In this study, we use the Spikey single-chip system as a physical emulation substrate [5]. This mixed-signal device combines analog components for modeling membrane and synapse dynamics with digital circuitry for the spike-based communication. Fig. 2 shows a photo of the device, along with a sketch of the neuron circuit which illustrates the origin of the three distortive effects discussed above.

The overlay in Fig. 2A shows how a spike emitted by a neuron (blue triangle) travels through communication buses (white line) to a synapse driver (red pentagon), which generates a voltage ramp that is fed into the synapse array (green circles). Inside the synapse, the voltage ramp is converted to an approximately exponential signal that is added to the total synaptic conductance of the neuron circuit. This sequence of processing stages causes the effective synaptic delays seen in Fig. 2B.

The neuron schematic in Fig. 2A explains the cause of non-logistic activation functions and noisy refractory times.

All reversal potentials are connected to the membrane by conductances that saturate at a certain amplitude. The maximum total conductance defines a minimum achievable effective membrane time constant, which limits the gain of the LIF activation function, as seen in Fig. 2C.

The duration of the refractory time is determined by a monoflop controlled by a current $I_{\text{ref}} \propto 1/\tau_{\text{ref}}$. In order to offset, at least to some extent, the effect of delays (see Fig. 1E) we require long refractory times, i.e., small currents, which cause some of the transistors in the monoflop to leave saturation and operate in a sub-threshold regime. This transition is accompanied by an increase in the relative amplitude of temporal noise, which increases the variability of τ_{ref}. This represents the primary cause of the large spike-to-spike fluctuations of the refractory time seen in Fig. 2D.

Having identified the origins for critical distortions in physical implementations of LIF sampling, we next turn to the central question of this study: is it possible to recover the computational capabilities of LIF sampling networks

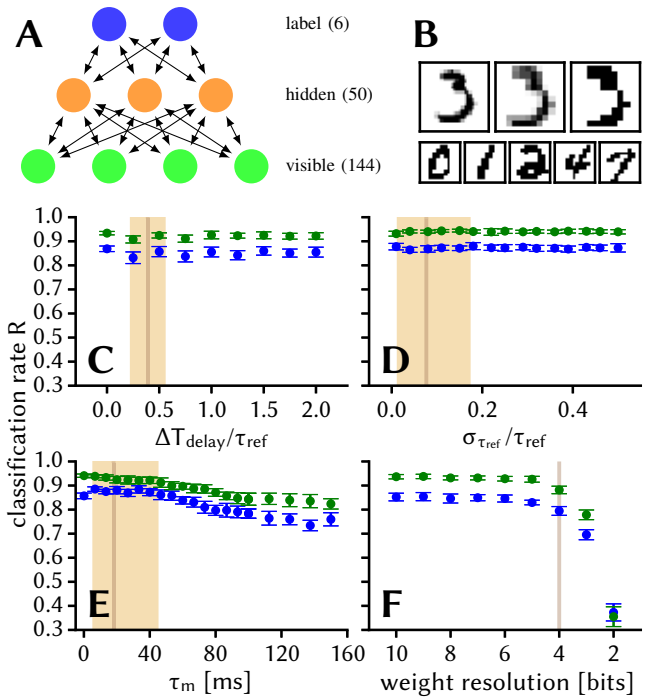

Fig. 3. Hierarchical LIF networks and their robustness to hardware-induced distortions. **(A)** Structure of the studied LIF network. **(B)** *Top row:* Exemplary training sample obtained from the MNIST dataset after resolution reduction and binarization. *Bottom row:* Exemplary training samples from all other classes. **(C)–(F)** Influence of simulated hardware-induced distortions on the classification performance of the network from A. Error bars represent the standard deviation over multiple runs with different random seeds. Green: performance on training data. Blue: performance on test data. Brown: mean value and standard deviation of the respective parameter measured on Spikey (see Fig. 2).

by finding a network architecture that is robust to these substrate-induced effects?

IV. Robust hierarchical LIF networks

The general framework of LIF sampling does not impose any restrictions on the network topology apart from the requirement of a zero-diagonal symmetric synaptic weight matrix $W_{ii} = 0$, $\boldsymbol{W} = \boldsymbol{W}^T$. However, imposing further restrictions on connectivity is of practical use.

When building a network that is able to learn and generalize from data, a rather natural hierarchization consists in subdividing the network into a layer representing the "visible" data, one or more hidden layers that recognize common features of data samples, and a final classification layer that assigns each sample a particular category or label. Indeed, this has been the guiding principle behind hierarchical neural networks, from multilayer perceptrons to deep convolutional nets [25]. In the case of BMs, the further removal of lateral connections within a layer has proven particularly beneficial for learning [26]. The resulting networks are so-called restricted Boltzmann machines (RBMs) and can be emulated by LIF networks with appropriate parameters as described in Sec. II.

The core insight of our present work is that hierarchical LIF networks that emulate RBMs exhibit notable robustness to the hardware-induced distortions discussed above. In this section, we argue why this is the case and demonstrate this robustness with software simulations of such a network with 3 layers (Fig. 3A).

Synaptic delays and noisy refractory times have similar effects on the sampled distribution. However, the nature of the information flow in LIF-based RBMs is expected to counter them both simultaneously. When presented with unambiguous input data, the mean firing rates of the visible neurons v_k are fixed; in our application, for example, they encode the grayscale values $g_k \in \{0, 1, \ldots, 255\}$ of pixels in the input image: $p(v_k = 1) = g_k/255$. In this regime, spike transmission delays have no effect, as the visible layer essentially operates in a rate-based mode for which time shifts do not matter. In this operating mode, the refractory noise is also averaged out.

Transmission delays and noisy refractory times remain critical for the interaction between hidden and label neurons. However, this interaction is comparatively weak in our 3-layer architecture (see Fig. 3A). In real-world scenarios, the label space typically has a much smaller dimensionality than the input space. Each hidden neuron therefore receives input from many visible neurons but only from few label neurons. Therefore, even though the visible-to-hidden synaptic weights are approximately as large as those between the hidden and label layer, the summed input from the visible layer is completely dominant by virtue of sheer numbers. Therefore, as the hidden layer is mostly driven by the input layer, the distorted interaction between the hidden and label neurons is likely to become insignificant.

The finite membrane time constant, on the other hand, can affect neurons in all layers and can not be neglected. However, this effect can be countered, at least to some extent, by the nature of the sampled distribution in well-trained networks. Wake-sleep training has the effect of carving of deep troughs in the network's energy landscape $E(\boldsymbol{z})$. These energy minima (probability maxima) correspond to particular patterns in each of the network layers, which are local attractors in the state space. Thus, if the deviations in the sampled distribution are small, the attractor landscape will not change significantly. Consequently, when the visible layer is clamped to input data, the above layers are still likely to fall into the corresponding attractor state, thus conserving the classification performance.

We tested these predictions in a series of software simulations. We trained a 3-layer LIF-based RBM (Fig. 3A) on a reduced version of the MNIST dataset [27]. To ensure compatibility with the Spikey chip, the network size was restricted to 144 visible, 50 hidden and 6 label neurons. The 12×12 pixel images belonging to 6 digit classes ("0", "1", "2", "3", "4", "7") were produced by first reducing the MNIST digit resolution, followed by binarization of the pixel values (Fig. 3B). For each class, both the training and the test set consisted of 20 randomly chosen images. The training consisted only of layer-by-layer pre-training with a wake-sleep-style algorithm [28]. We have deliberately refrained from fine-tuning the weights with, e.g., backpropagation, in

order to maintain compatibility with Hebbian plasticity.

Fig. 3C-E show the simulated effects of the three hardware-induced distortion mechanisms discussed above. As expected, neither synaptic transmission delays (Fig. 3C) nor variability of refractory times (Fig. 3D) affected the performance of the network significantly. Over a surprisingly large range of membrane time constants, the classification rate remained almost unaffected. Only after the activation function became significantly distorted by large τ_{m} did the attractor landscape change significantly enough to cause a decay in the classification rate (Fig. 3E). Overall, within the parameter ranges of the Spikey chip, our network remained only weakly affected by the studied mechanisms.

However, since in a later step the network was mapped to the hardware without any further training, we needed to also consider the effect of discretized synaptic weights. By default, synaptic weights on the Spikey chip are only controllable up to 4-bit precision [5]. It is important to note that this does not pose a fundamental problem to networks of this type; the effects of weight discretization can be countered by appropriate in-the-loop training, as discussed in, e.g., [14], [29]. Here, we only take this effect into account as a preparation of the hardware experiments in Sec. V. Fig. 3F shows the effect of weight discretization on the network's classification performance. For the Spikey chip, the performance decay lies at approximately $5.6\,\%$. Note that this effect is significantly larger than the effects caused by each of the other distortion mechanisms.

A combined simulation of all distortive effects was used to provide a reference for the later emulation on Spikey. All effects were simulated with amplitudes corresponding to values measured on Spikey (blue bars in Fig. 3C-F, see also Fig. 2). In the ideal, undistorted case, the LIF network had a classification performance of

$$\begin{aligned} R^{\mathrm{train}} &= 93.4 \pm 0.9\,\% \\ R^{\mathrm{test}} &= 86.6 \pm 1.7\,\% \end{aligned} \quad , \tag{11}$$

which was reduced to

$$\begin{aligned} R^{\mathrm{train}} &= 90.7 \pm 1.7\,\% \\ R^{\mathrm{test}} &= 78.1 \pm 1.5\,\% \end{aligned} \tag{12}$$

when all distortive effects were simultaneously present. A comparison to Fig. 3F shows that most of this performance decay was due to the 4-bit weight discretization.

V. Neuromorphic Implementation

The mapping of the network to Spikey required a series of modifications, which we discuss in the following.

In the previous section, we argued that visible neurons essentially operate in a rate-based mode during clamping. This allows the activity ν_i of each visible neuron to be modeled as an effective bias

$$\tilde{b}_{ki} = p(v_i = 1) \cdot w_{ki} \quad , \tag{13}$$

to the kth hidden neuron, where v_i represents the state of the ith visible neuron and w_{ki} the synaptic weight between the ith visible and the kth hidden neuron. The complete

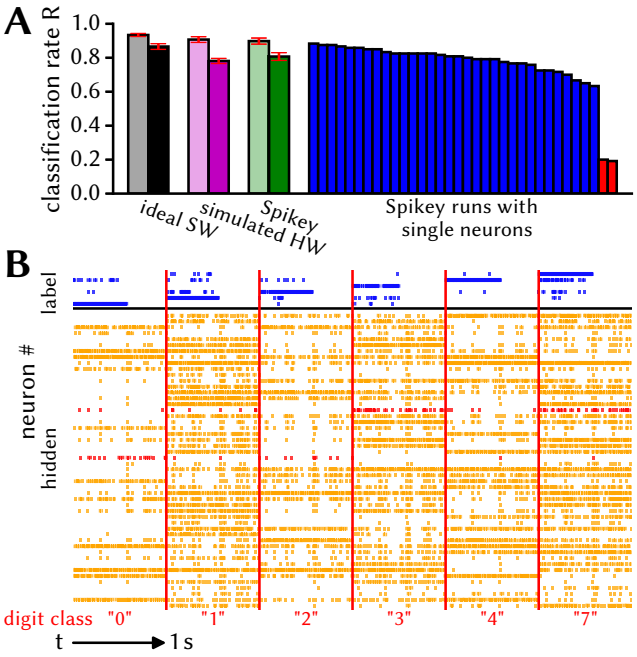

Fig. 4. Study of a direct-to-hardware mapping of the hierarchical LIF network from Fig. 3A. **(A)** Classification performance. Black: Software simulation of a distortion-free LIF network (cf. leftmost data points in Fig. 3C-E). Purple: Software simulation of the LIF network with all distortion mechanisms being present simultaneously, with amplitudes and variances as measured on Spikey (cf. areas in Fig. 3C-E highlighted in brown). Green: Hardware emulation of the hidden layer with software evaluation of the label layer. Blue/red: Emulation of the hidden layer with repeated use of single neurons, followed by software evaluation of the label layer. The two "bad" neurons marked in red were not well configurable and therefore performed at chance level. **(B)** Exemplary spike trains of a subset of neurons in the LIF network with the hidden layer running on hardware. Spike trains belonging to the two "bad" neurons from A are marked in red.

visible layer can then be omitted altogether and replaced by an effective bias $\tilde{b}_k$ for each hidden neuron:

$$\tilde{b}_k = \sum_i \tilde{b}_{ki} + b_k \quad , \tag{14}$$

where b_k is the original bias of the kth hidden neuron. In our particular case, since we use 12×12–pixel binarized images and have set all biases to zero during training in order to simplify the transition to hardware, this reduces to

$$\tilde{b}_k = \sum_{i=1}^{144} w_{ki} v_i \quad . \tag{15}$$

On the chip, biases are implemented as high-frequency regular spike trains connected to the hidden neurons with weights w_k^b. For an arbitrary synaptic kernel scaled with w_k^b, the average effect of a regular spike train on the membrane potential of an LIF neuron is proportional to w_k^b. Therefore, $\tilde{b}_k$ can be controlled, within the imposed 4-bit precision, by appropriately configuring w_k^b. Note that these spike trains also need to be routed across the chip (Fig. 2), so this does not circumvent synaptic delays.

For the hidden layer, we have chosen those 50 neurons on the chip which responded best to the bias stimulus described

above. Only half of the chip was used for these experiments in order to simplify on-chip routing.

The label layer was implemented in software. Spikes produced by the hidden layer were fed into six label neurons simulated with NEST [30]. With this, we essentially broke the hidden→label→hidden feedback loop, but as we argued in Sec. IV, it should not significantly affect the classification performance of the network. Furthermore, this allowed a more detailed investigation of the quality of single neurons on the chip, as discussed below. The label assigned by the network to the input image was determined by the label neuron which produced the most spikes during the clamping period.

Fig. 4 shows the classification performance of this setup, along with the spike trains from several exemplary classification runs. The performance of the hardware implementation was

$$\begin{aligned} R^{\text{train}} &= 89.8 \pm 1.8\,\% \\ R^{\text{test}} &= 80.7 \pm 2.3\,\% \quad . \end{aligned} \tag{16}$$

Within the error margins, this corresponds very well to the reference software simulations (12). The slightly better average classification can be attributed to the explicit selection of the 50 hidden neurons. Indeed, this result not only confirms the robustness of our network model, but also highlights its robustness to various other hardware-induced distortions that we did not explicitly account for, such as parameter noise and crosstalk [5]. Furthermore, this implementation is surprisingly robust even towards few neurons having strongly deviant firing characteristics, as discussed below.

In our network model, hidden neurons are not laterally interconnected. Furthermore, as the label layer was simulated in software, there was also no label-mediated lateral interaction between hidden neurons. Therefore, it was possible to emulate the entire network with one Spikey neuron at a time. In an alternative emulation setup, a sequence of $k = (1, \ldots, 50)$ emulation runs containing a single hardware neuron was performed. In the kth run, the hardware neuron was configured to represent the kth hidden neuron by receiving the corresponding input spike train. The output spike trains from these runs were aggregated and fed into the label layer, as before. This experiment was repeated for a subset of 38 out of the 50 selected hardware neurons, with the results plotted as thin bars in Fig. 4A.

The overall performance of each neuron quantifies its quality for the task at hand. The main reason for the differences between the neurons is the shape of their activation function, some of which can be seen in Fig. 2C. Some neurons perform poorly because their activation function is too shallow, thus strongly skewing the sampled distribution. At the other extreme, a very steep activation function is also detrimental, because the resolution of the synaptic weights does not permit an arbitrarily fine-grained tuning of effective weights and biases. Note, in particular, how two of the neurons perform at chance level (Fig. 4A, red bars). However, the existence of such neurons does not appear to have a strong effect on the classification performance of the network as a whole (Fig. 4A, purple vs. green bar).

VI. Discussion

One of the most important challenges for analog neuromorphic computing is the design of neural network architectures that are robust to hardware-induced distortions of network dynamics and parameters. In this paper, we have argued that hierarchical spiking sampling networks emulating restricted Boltzmann machines are inherently resistant to such distortions. We have studied three specific distortion mechanisms that are, in general, strongly disruptive to the ongoing computation in sampling LIF networks: synaptic transmission delays, variability of refractory times and saturating membrane conductances. Despite their apparent sensitivity, we have shown how a hierarchical topology shapes the information flow in a way that makes them largely resilient to these effects. The results obtained in software simulations were also confirmed in experiments on an accelerated analog neuromorphic device. Furthermore, in addition to being robust to the studied distortion mechanisms, our network model also displayed an encouraging degree of resilience to other hardware-induced effects which can not be quantified as systematically as the studied ones.

The choice of our neuron model (LIF) was made, on one hand, for analytical tractability, but, more importantly, due to the fact that this model represents a common denominator for many other spiking neuron models (Hodgkin-Huxley, Izhikevich, AdEx). With appropriate parameter choices, all of these models can achieve dynamics that are close to those required for sampling. Furthermore, the LIF model represents a de-facto standard in neuromorphic engineering [4]–[9].

Altogether, the observed properties of these networks encourage further theoretical and experimental investigation. Here, we only studied relative performance losses, so we only used a relatively small network and a very small dataset. Software simulations show that larger-scale versions of these networks enable efficient and powerful inference in more complex data spaces [15]. It will be interesting to see whether such large networks remain as robust to hardware-induced distortions as their smaller instantiations studied here. Large-scale accelerated analog devices are already in place [6] and will be able to accomodate these experiments. With our proposed architecture and rather conservative clamping schedule of 1 biological second per image (Fig. 4B), the currently achieved acceleration factor of about 10^4 will, for example, enable the classification of the full MNIST dataset within $1\,\mathrm{s}$ of wall-clock time.

Although we only used our hierarchical LIF networks for classification, they can also be used as generative models to perform, for example, pattern completion. The nature of the energy landscape in these networks suggests that their generative properties could also be robust to hardware-induced distortions. Since a clear image corresponds to a deep mode in the energy landscape, small distortions in the sampled distribution are unlikely to strongly disrupt the generative properties of the network. Probabilistic switching between different modes when ambiguous input is present can then be facilitated by short-term plasticity as shown

in [15], a mechanism that is readily available on several accelerated neuromorphic platforms [5], [6].

In this paper, we have deliberately refrained from further training of the hardware-emulated networks. However, it is expected that training the hardware "in the loop" will significantly improve classification. The idea behind in-the-loop training is to iteratively alternate between a forward pass on hardware, during which the emulated network activity is recorded, and a backward pass in software, where the network parameters are updated by, e.g., error backpropagation. Recent studies have demonstrated, both in software [31] and on analog neuromorphic hardware [14], that the parameter updates need not be precise, but only approximately follow the gradient of the likelihood function. Furthermore, accelerated systems currently in development [32], [33] also implement powerful on-chip learning solutions. Such architectures will not only enable accelerated classification, but, even more importantly, accelerated learning of the network parameters.

ACKNOWLEDGMENTS

The first two authors contributed equally to this work. MAP and AS designed and performed the experiments in software and on the Spikey chip. OB designed the software environment for performing the simulations. All authors contributed to writing this paper. We would like to thank Thomas Pfeil and Johannes Bill for technical support with the Spikey chip, as well as Luziwei Leng for providing the model parameters. We further thank Andreas Baumbach, Johannes Bill, Dominik Dold, Carola Fischer, Vitali Karasenko, Akos Kungl and Walter Senn for their invaluable contribution to the final manuscript. This research was supported by EU grant #604102 (Human Brain Project).

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015.

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, 2016.

[3] H. Markram, E. Muller, S. Ramaswamy, M. Reimann, M. Abdellah, C. Sanchez, A. Ailamaki *et al.*, "Reconstruction and simulation of neocortical microcircuitry," *Cell*, vol. 163, no. 2, 2015.

[4] G. Indiveri, E. Chicca, and R. Douglas, "Artificial cognitive systems: From VLSI networks of spiking neurons to neuromorphic cognition," *Cognitive Computation*, vol. 1, no. 2, 2009.

[5] T. Pfeil, A. Grübl, S. Jeltsch, E. Müller, P. Müller, M. A. Petrovici, M. Schmuker, D. Brüderle, J. Schemmel, and K. Meier, "Six networks on a universal neuromorphic computing substrate," *Frontiers in Neuroscience*, vol. 7, 2013.

[6] J. Schemmel, D. Brüderle, A. Grübl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems*, 2010.

[7] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker system architecture," *IEEE Transactions on Computers*, vol. 99, 2012.

[8] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, 2014.

[9] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza *et al.*, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, 2014.

[10] M. A. Petrovici, B. Vogginger, P. Müller, O. Breitwieser, M. Lundqvist, L. Muller, M. Ehrlich, A. Destexhe, A. Lansner, R. Schüffny, J. Schemmel, and K. Meier, "Characterization and compensation of network-level anomalies in mixed-signal neuromorphic modeling platforms," *PLOS ONE*, vol. 9, no. 10, 2014.

[11] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, 2011.

[12] S. Sheik, F. Stefanini, E. Neftci, E. Chicca, and G. Indiveri, "Systematic configuration and automatic tuning of neuromorphic systems," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, 2011.

[13] J. Bill, K. Schuch, D. Brüderle, J. Schemmel, W. Maass, and K. Meier, "Compensating inhomogeneities of neuromorphic VLSI devices via short-term synaptic plasticity," *Front. Comp. Neurosci.*, vol. 4, no. 129, 2010.

[14] S. Schmitt, J. Klähn, G. Bellec, A. Grübl, M. Güttler, A. Hartel, S. Hartmann, D. Husmann, K. Husmann, S. Jeltsch, V. Karasenko *et al.*, "Neuromorphic hardware in the loop: Training a deep spiking network on the brainscales wafer-scale system," *Proceedings of the International Joint Conference on Artificial Neural Networks*, 2016.

[15] L. Leng, M. A. Petrovici, R. Martel, I. Bytschok, O. Breitwieser, J. Bill, J. Schemmel, and K. Meier, "Spiking neural networks as superior generative and discriminative models," in *Cosyne Abstracts, Salt Lake City USA*, February 2016.

[16] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, 1943.

[17] F. Rosenblatt, "The Perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, 1958.

[18] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science*, vol. 9, 1985.

[19] M. A. Petrovici, J. Bill, I. Bytschok, J. Schemmel, and K. Meier, "Stochastic inference with spiking neurons in the high-conductance state," *Physical Review E*, vol. 94, no. 4, 2016.

[20] L. Buesing, J. Bill, B. Nessler, and W. Maass, "Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons," *PLoS Computational Biology*, vol. 7, no. 11, 2011.

[21] M. A. Petrovici, I. Bytschok, J. Bill, J. Schemmel, and K. Meier, "The high-conductance state enables neural sampling in networks of LIF neurons," *BMC Neuroscience*, vol. 16, no. Suppl 1, 2015.

[22] D. E. Rumelhart, G. E. Hinton, and W. R.J., "Learning internal representations by error propagation," *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. I, 1986.

[23] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, 2002.

[24] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann machines," in *International Conference on Artificial Intelligence and Statistics*, 2009.

[25] S. Haykin, "A comprehensive foundation," *Neural Networks*, vol. 2, no. 2004, 2004.

[26] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, 2006.

[27] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1(4), 1989.

[28] R. Salakhutdinov, "Learning deep boltzmann machines using adaptive MCMC," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.

[29] S. K. Esser, P. A. Merolla, J. V. Arthur, A. S. Cassidy, R. Appuswamy *et al.*, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proceedings of the National Academy of Sciences*, vol. 113, no. 41, 2016.

[30] M.-O. Gewaltig and M. Diesmann, "NEST (NEural Simulation Tool)," *Scholarpedia*, vol. 2, no. 4, 2007.

[31] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature Communications*, vol. 7, 2016.

[32] S. Friedmann, J. Schemmel, A. Grübl, A. Hartel, M. Hock, and K. Meier, "Demonstrating hybrid learning in a flexible neuromorphic hardware system," *IEEE Transactions on Biomedical Circuits and Systems*, no. 99, 2016.

[33] J. Schemmel, L. Kriener, P. Müller, and K. Meier, "An accelerated analog neuromorphic hardware system emulating nmda and calcium-based non-linear dendrites," *Proceedings of the International Joint Conference on Artificial Neural Networks*, 2016.