Spiking Neural Network Nonlinear Demapping on Neuromorphic Hardware for IM/DD Optical Communication

Elias Arnold, Georg Böcherer, *Member, IEEE, Member, Optica,* Florian Strasser, Eric Müller, Philipp Spilger, Sebastian Billaudelle, Johannes Weis, Johannes Schemmel, *Member, IEEE,* Stefano Calabrò, Maxim Kuschnerov

Abstract—Neuromorphic computing implementing spiking neural networks (SNN) is a promising technology for reducing the footprint of optical transceivers, as required by the fast-paced growth of data center traffic. In this work, an SNN nonlinear demapper is designed and evaluated on a simulated intensity-modulation direct-detection link with chromatic dispersion. The SNN demapper is implemented in software and on the analog neuromorphic hardware system BrainScaleS-2 (BSS-2). For comparison, linear equalization (LE), Volterra nonlinear equalization (VNLE), and nonlinear demapping by an artificial neural network (ANN) implemented in software are considered. At a pre-forward error correction bit error rate of $2 \cdot 10^{-3}$, the software SNN outperforms LE by 1.5 dB, VNLE by 0.3 dB and the ANN by 0.5 dB. The hardware penalty of the SNN on BSS-2 is only 0.2 dB, i.e., also on hardware, the SNN performs better than all software implementations of the reference approaches. Hence, this work demonstrates that SNN demappers implemented on electrical analog hardware can realize powerful and accurate signal processing fulfilling the strict requirements of optical communications.

Index Terms—Spiking Neural Network, Optical Communication, Equalization, Data Centers, Intensity-Modulation Direct-Detection

I. INTRODUCTION

THE fast-paced growth of data center traffic is the driver behind the increase in bit rate and, at the same time, the footprint reduction of the optical transceivers. This trend results in an urgent need to decrease the power consumption per bit. Whereas evolutionary steps can mitigate the problem, the exponential traffic growth asks for a paradigm shift. To resolve this dilemma, recent research envisions moving parts of digital signal processing (DSP) to analog frontends with lower power consumption.

One approach is photonic neuromorphic computing [1], which has been proposed, e.g., for chromatic dispersion (CD) compensation and nonlinear equalization in short-reach optical transmission [2], [3], [4]. However, although photonics can operate faster than electronic hardware, the latter scales better in terms of footprint and power consumption.

E. Arnold, E. Müller, P. Spilger, S. Billaudelle, J. Weis, and J. Schemmel are with the Electronics Vision(s) Group at the Kirchhoff-Institute for Physics, Heidelberg University, Germany, e-mail: elias.arnold@kip.uni-heidelberg.de

G. Böcherer, F. Strasser, S. Calabrò, and M. Kuschnerov are with Huawei Technologies Düsseldorf GmbH, Munich Research Center, Germany, e-mail: georg.bocherer@huawei.com

The return to analog electrical adaptive equalizers is also gaining traction, e.g., in [5], the transmitter DSP feeds two electrical non-return-to-zero (NRZ) signals to an analog pulse-amplitude-modulation 4-level (PAM-4) encoder, whose output is filtered by a continuous-time linear equalizer (CTLE) and a 3-tap feed forward equalizer (FFE).

At the same time, the research community is striving to implement more powerful nonlinear algorithms, e.g. based on artificial intelligence (AI) techniques, on analog electronics. An important subfield is in-memory-computing (IMC) [6], which aims for efficient calculation of vector-matrix multiplications. Research on IMC is mainly driven by the urgent need for AI accelerators for artificial neural networks (ANNs). Eventually, IMC may enable the use of ANNs for signal processing in the data path of communication systems, see, e.g., [7].

Analog electronic neuromorphic computing offers an alternative path towards AI-based signal processing. Spiking neural networks (SNNs) [8] in analog hardware [9], adopt the brain's unique power efficiency by imitating the basic functioning of the human brain. They combine the sparse representation of information by event-based spiking signals with power efficient IMC. In [10], we have shown that SNN FFEs simulated in software can compensate nonlinear impairments in intensity-modulation direct-detection (IM/DD) links. In [11], SNN decision feedback equalization (DFE) is considered for compensating severe linear inter-symbol interference (ISI).

Recently, in-the-loop (ITL) training of SNNs on analog hardware [12] has shown promising results by achieving state-of-the-art performance in inference tasks [13]. In [14], we presented preliminary results on the design and evaluation of an SNN demapper on the analog neuromorphic BrainScaleS-2 (BSS-2) system [9]. Specifically, we considered the detection of a PAM-4 signal in a simulated IM/DD link, which is impaired by CD and additive white Gaussian noise (AWGN), as displayed in Fig. 1. Our results in [14] show that SNNs emulated on the neuromorphic BSS-2 hardware outperform linear equalization in software, while the gap between software and hardware SNN is slightly below 1 dB.

In this work, we detail and extend our previous work on SNN-based neuromorphic demapping [14]. For the same IM/DD link model as in [14] (see Fig. 1), we reduce the SNN software-hardware penalty to below 0.2 dB. We achieve this by optimizing the hardware operation point, tuning the training procedure, and adjusting the input-spike encoding. We
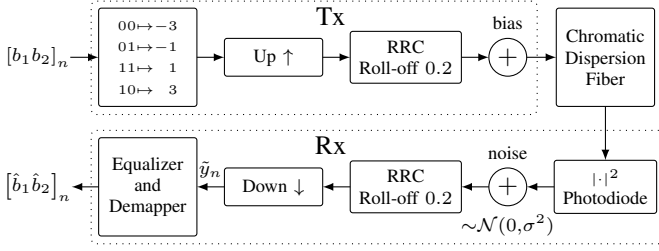
Figure 1. The simulated IM/DD link schematics. A bit sequence is mapped at the transmitter (Tx) to a PAM-4 signal and is impaired by CD in the fiber. At the receiver (Rx), after square-law detection, AWGN is added. An equalizer/demapper recovers the transmitted bits.

compare the proposed solution with software implementations of a linear equalizer, a 5-th order Volterra nonlinear equalizer (VNLE), and a nonlinear ANN demapper. Despite the nonzero hardware penalty, our hardware SNN demapper performs better than the considered simulated reference algorithms. At the assumed forward error correction (FEC) bit error rate (BER) threshold of $2 \cdot 10^{-3}$, the gain over a linear equalizer is approximately 1.5 dB.

The remainder of this work is organized as follows. In Section II, we outline the IM/DD link and explain the implementation of the reference demappers. Section III details the SNN demapper and the input encoding scheme. Subsequently, we provide an overview of the BSS-2 platform in Section IV. The training procedure is explained in Section V. In Section VI, we show our results and in Section VII, we present our conclusions.

II. IM/DD Model and Reference Demappers

In this section, we detail our IM/DD link model and specify the reference algorithms, i.e., linear equalization (LE) and VNLE followed by hard decision (HD) demapping, and ANN nonlinear demapping. All reference demappers are simulated in double-precision floating-point arithmetic, except for the ANN, which uses single-precision floating-point arithmetic. The considered ANN and VNLE architectures are rather complex, i.e., the ANN has two nonlinear hidden layers and the VNLE uses the full filter length also for the higher order terms. The purpose of considering complex ANN and VNLE processing is to benchmark what performance we can achieve by nonlinear processing without considering resource usage, and then to compare the SNN performance to such benchmark.

A. Simulated IM/DD Link

We simulate the transmission of PAM-4 symbols in the O-band at a baudrate of 112 GBd. Assuming an FEC overhead of 12 % with a BER threshold of $2 \cdot 10^{-3}$, we target a corresponding net bit rate of 200 Gbit s^{-1}.

We display the simulated link in Fig. 1 and the corresponding parameters in Table I. At the transmitter, a bit sequence $[b_1 b_2]^N$ is mapped to a length N PAM-4 signal $\boldsymbol{y} = y^N$ according to a Gray-labelled alphabet $\mathcal{A} = \{-3, -1, 1, 3\}$. This signal is upsampled, root-raised-cosine (RRC) filtered, and offset by a bias. The resulting sequence is impaired by CD,

Table I
PARAMETERS OF THE SIMULATED IM/DD LINK

Parameter	Value
Net bit rate	200 Gbit s^{-1}
FEC threshold	12 %
Baudrate	112 GBd
Wavelength λ	1270 nm
Dispersion D_{CD}	-5 ps nm^{-1} km^{-1}
Fiber length l	4 km
Alphabet $\mathcal{A}$	$\{-3, -1, 1, 3\}$
Sequence length N	10000
Bias b	2.25
RRC roll-off α	0.2
Upsampling n_{up}	3
Downsampling n_{down}	3

Remarks on the simulated IM/DD link parameters:

1) Wavelength and dispersion are in the range specified in [16, Table 9.6].
2) For the considered baudrate and fiber length, the dispersion in terms of delay spread between the frequency components at $\pm$ Nyquist frequency is 1.35 symbols.
3) The bias results in a carrier-to-signal-power-ratio (CSPR) of 9.6 dB.
4) The combination of CD and PD results in a band limitation, despite the fact that CD alone acts as an allpass filter. Consider

$|\text{signal} + \text{carrier}|^2 = |\text{signal}|^2 + 2\text{Re}(\text{signal} \cdot \text{carrier}) + |\text{carrier}|^2.$

For the considered parameters, CD and PD cause for the linear term $2\text{Re}(\text{signal} \cdot \text{carrier})$ an attenuation of 6.2 dB at the Nyquist frequency, compared to frequency 0.

modelled linearly following, e.g., [15, Sec. 3.2], to simulate the effect of the fibre on the propagating optical signal. We assume that the power dissipated into the fiber is low and we ignore fiber non-linearities in our simulated link. At the receiver, the signal goes through a photodiode (PD), which is modeled as a square-law device, and AWGN is added. The resulting signal is RRC filtered and downsampled, resulting in the received sequence $\tilde{\boldsymbol{y}} = \tilde{y}^N$. Finally, bit decisions $[\hat{b}_1 \hat{b}_2]^N$ are output by the respective device. We index the bit sequence and signal elements with n, $0 \le n < N$. Note that a constellation with non-equidistant signal points to precompensate the squaring of the PD may be beneficial, however, this is beyond the scope of this work.

B. Linear Minimum Mean Squared Error (LMMSE) Equalization

Our first reference detector consists of LE followed by HD demapping. To simplify the notation in the following, we specify the samples considered for equalizing the n-th sample via double-indexing,

$$\tilde{\boldsymbol{y}}_n = \left[\tilde{y}_{n,0}, \tilde{y}_{n,1}, \ldots, \tilde{y}_{n,n_{\text{tap}}-1}\right], \tag{1}$$

$$:= \left[\tilde{y}_{n-\lfloor n_{\text{tap}}/2 \rfloor}, \ldots, \tilde{y}_n, \ldots, \tilde{y}_{n+\lfloor n_{\text{tap}}/2 \rfloor}\right]. \tag{2}$$

Specifically, the LE calculates

$$\hat{y}_n = c + \sum_{j=0}^{n_{\text{tap}}-1} \tilde{y}_{n,j} h_j, \tag{3}$$

where the bias c accounts for residual direct current (DC) and $\boldsymbol{h}$ are the filter coefficients. The number n_{tap} of taps is the filter width and is assumed to be odd. We use data-aided training to calculate $\boldsymbol{h}$ and c so as to minimize the mean squared error

(MSE), $1/N \sum_n (\hat{y}_n - y_n)^2$. To this end, we form the feature matrix

$$\boldsymbol{A} = \begin{bmatrix} \cdots & 1 & 1 & 1 & \cdots \\ \cdots & \tilde{\boldsymbol{y}}_{n-1}^\top & \tilde{\boldsymbol{y}}_n^\top & \tilde{\boldsymbol{y}}_{n+1}^\top & \cdots \end{bmatrix}^\top \qquad (4)$$

and we then solve

$$[c^*, \boldsymbol{h}^*] = \underset{c, \boldsymbol{h}}{\operatorname{argmin}} \| \boldsymbol{A}[c, \boldsymbol{h}]^\top - \boldsymbol{y}^\top \|_2^2. \qquad (5)$$

The demapper calculates an HD $[\hat{b}_1 \hat{b}_2]_n$ from the equalized sample $\hat{y}_n$ via three decision boundaries, which are chosen such that the BER is minimized. Note that at the transmitter, the signal points in the PAM-4 constellation are equidistant, while the received signal points are not equidistant anymore, because of the nonlinear transfer function of the PD. The LE cannot compensate nonlinear distortions, so the received signal points remain non-equidistant after LE. This is compensated in part by the demapper, as the decision boundaries are optimized with respect to the received and equalized signal points $\hat{y}_n$, not the transmitted signal points y_n. In the following, we refer to the combination of a LE and a memoryless demapper by linear minimum mean square error (LMMSE) equalization.

C. Volterra Nonlinear Equalizer (VNLE)

As nonlinear reference, we consider a 5-th order VNLE, see, e.g., [17, Chap. 14], [18]. The VNLE is very similar to the LE, with the difference that the VNLE feature matrix contains additional columns for the monomials of order higher than 1. The 0-th and 1-st order feature vectors considered by the LE are

$$\boldsymbol{f}_{n,0} = [1], \qquad (6)$$

$$\boldsymbol{f}_{n,1} = \left[\tilde{y}_{n,j} \Big| 0 \leq j < n_{\text{tap}} \right] = \tilde{\boldsymbol{y}}_n. \qquad (7)$$

Accordingly, the feature vectors of order two and higher are defined by

$$\boldsymbol{f}_{n,2} = \left[\tilde{y}_{n,j} \cdot \tilde{y}_{n,k} \Big| 0 \leq j \leq k < n_{\text{tap}} \right], \qquad (8)$$

$$\boldsymbol{f}_{n,3} = \left[\tilde{y}_{n,j} \cdot \tilde{y}_{n,k} \cdot \tilde{y}_{n,\ell} \Big| 0 \leq j \leq k \leq \ell < n_{\text{tap}} \right], \qquad (9)$$

$$\vdots$$

The n-th row of the feature matrix then consists of the concatenation of the feature vectors $\boldsymbol{f}_{n,m}$, $m = 0, \ldots, 5$. The number of features of order m is given by

$$\operatorname{length}(\boldsymbol{f}_{n,m}) = \binom{m + n_{\text{tap}} - 1}{m} \qquad (10)$$

and accordingly, for $n_{\text{tap}} = 7$, the total number of coefficients of the VNLE is

$$\sum_{m=0}^{5} \operatorname{length}(\boldsymbol{f}_{n,m}) = 792. \qquad (11)$$

The optimization of the coefficients and the specification of the demapper is data-aided and follows exactly the LE procedure described above, where the key step is solving (5).

D. Nonlinear ANN Demapper

We consider an ANN with $n_{\text{tap}} = 7$ input units, a first hidden layer with 40 neurons, followed by a second hidden layer with 20 neurons, both activated by the tanh function, and a linear output layer with 4 neurons. The output values are interpreted as log-probabilities providing a soft decision (SD) on the PAM-4 symbols. A symbol-wise HD is obtained by choosing the symbol of highest probability and the bitwise HD is obtained from the symbol decisions via the Gray label.

III. SPIKING NEURAL NETWORKS FOR EQUALIZATION

This section outlines the SNN demappers. We detail their emulation on BSS-2 in Sec. IV.

SNNs consist of neurons, evolving in time t, and communicating via binary spike events. The leaky integrate-and-fire (LIF) spiking neuron model [8, Sec. 1.3] captures some of the core dynamics observed in biological neurons while at the same time maintaining a tractable complexity. LIF neurons integrate synaptic input current $I(t)$ onto their internal membrane voltage state $v_{\text{m}}(t)$ according to the dynamics described by the ordinary differential equation (ODE)

$$\tau_{\text{m}} \frac{\mathrm{d}v_{\text{m}}(t)}{\mathrm{d}t} = \left[v_{\text{l}} - v_{\text{m}}(t) \right] + R_{\text{l}} \cdot I(t). \qquad (12)$$

Here, τ_{m} is the membrane time constant, R_{l} is the leakage resistance, and v_{l} is the leakage potential. When the membrane potential reaches a threshold potential ϑ at time t^{s}, the neuron emits a spike $z(t) = \delta(t - t^{\text{s}})$, with δ being the Dirac delta distribution, and v_{m} is reset to a potential $v_{\text{m}}(t^{\text{s}}) = v_{\text{r}}$. The synaptic current I is induced by *presynaptic* neurons $\{n_i\}$, projecting spike events $z_i(t) = \delta(t - t_i^{\text{s}})$ at times $\{t_i^{\text{s}}\}$ onto the *postsynaptic* neuron through synapses with weights w_i, thereby causing an exponentially decaying current described by the ODE

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = -\frac{I(t)}{\tau_{\text{s}}} + \sum_i w_i z_i(t). \qquad (13)$$

τ_{s} denotes the synaptic time constant. The LIF dynamics are exemplified in Fig. 2A. Neurons with a disabled spiking mechanism and membrane dynamics according to (12), are referred to as leaky integrator (LI) neurons [8, Sec. 1.3].

In the following, we consider an SNN with the structure outlined in [14] and depicted in Fig. 2B. It consists of one hidden layer constituted by N^{h} LIF neurons $\{n_j^{\text{h}}\}$, projecting its spike events onto one output layer with $N^{\text{o}} = 4$ non-spiking LI readout neurons $\{n_k^{\text{o}}\}$. The hidden layer receives spike events from the input layer, encoding a set of input samples $\tilde{\boldsymbol{y}}_n$. The readout layer's outputs are translated to symbol-level log-probabilities. Spike-input encoding and output decoding are explained in the following.

a) Input Spike-Encoding: To demap a sample $\tilde{y}_n$, we consider the chunk $\tilde{\boldsymbol{y}}_n$ defined in (1) and assign to each sample $\tilde{y}_{n,\ell}$ a set of input neurons $\{n_{i,\ell}^{\text{i}}\}_{i=0}^{<\bar{N}_\ell^{\text{i}}}$, encoding the sample value in their spike times $\{t_{i,\ell}^{\text{s}}\}$. Here, ℓ indexes the samples within $\tilde{\boldsymbol{y}}_n$ and $\bar{N}_\ell^{\text{i}} \in \mathbb{N}$ is the number of neurons associated to sample $\tilde{y}_{n,\ell}$, such that $N^{\text{i}} = \sum_{\ell=0}^{<n_{\text{tap}}} \bar{N}_\ell^{\text{i}}$ is the size of the input layer. Further, we assign each input neuron $n_{i,\ell}^{\text{i}}$ a
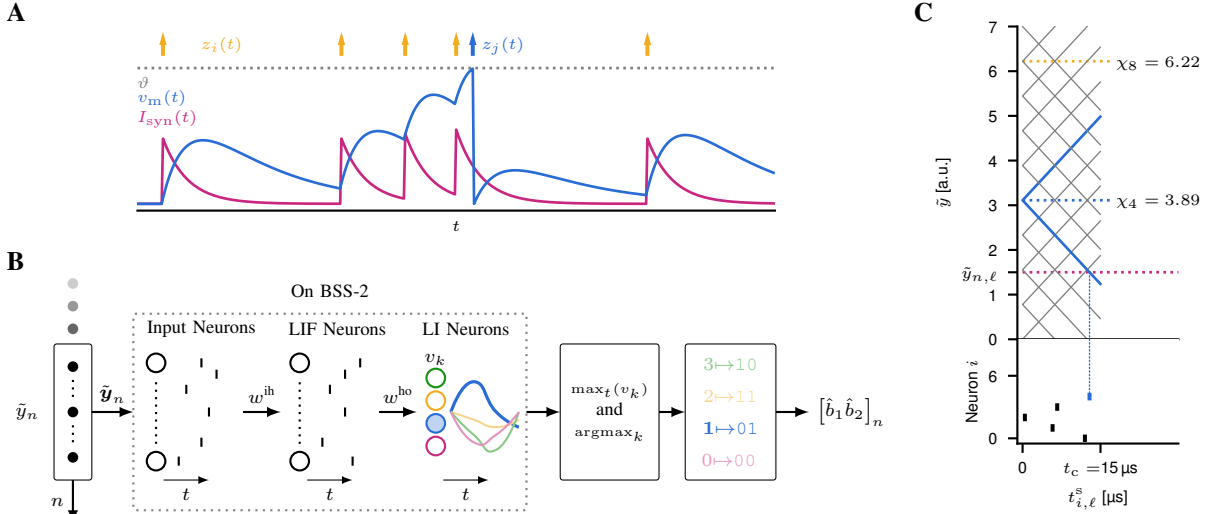
Figure 2. **(A)** LIF dynamic: a LIF neuron receives input spikes (yellow arrows), causing a synaptic current (purple) onto the neuron's membrane, thereby deflecting its membrane potential (blue). As the potential crosses the threshold (gray), the neuron sends out a spike (blue arrow) and is reset. **(B)** The considered SNN setup for joint equalization and demapping. A chunk $\tilde{\boldsymbol{y}}_n$ of samples is translated to spike times of input neurons, projected onto one hidden LIF layer. A readout layer of LI neurons adjust their membrane voltage by integrating these hidden spike events. Symbol-level log-probabilities are calculated by taking the maximum membrane value over time of the readout neurons, allowing to infer bit-decisions. **(C)** Schematic drawing of the input spike-encoding: a sample $\tilde{y}_{n,\ell}$ (purple dotted) is translated into spike times according to the distance to reference points assigned to 10 neurons (gray lines). Spikes occurring after a cutoff time t_c are omitted.

reference point $\chi_{i,\ell}$, which we choose together with $\bar{N}_\ell^\mathrm{i}$ to be independent of ℓ, $\chi_{i,\ell} = \chi_i$ and $\bar{N}_\ell^\mathrm{i} = \bar{N}^\mathrm{i}$. Finally, we compute the spike time $t_{i,\ell}^\mathrm{s}$ by scaling the distance of $\tilde{y}_{n,\ell}$ to χ_i,

$$t_{i,\ell}^\mathrm{s} = \alpha \big| \tilde{y}_{n,\ell} - \chi_i \big| + o, \qquad (14)$$

where α is a scaling factor and o is an offset. This spike-encoding preserves all information and encodes the value $\tilde{y}_{n,\ell}$ redundantly in $\bar{N}^\mathrm{i}$ spike times in order to increase the network's activity and enrich information in time. The values χ_i, $\bar{N}^\mathrm{i}$ and α are subject to tuning and are chosen to augment the network's activity by the right amount to achieve optimal performance. Here, the χ_is are equidistantly spaced in the domain of $\tilde{y}_{n,\ell}$ and α is selected to obtain spike times comparable to the membrane time constants. Note, while larger $\bar{N}^\mathrm{i}$ increases the network's complexity, it potentially stabilizes the network's performance on a noisy analog substrate like BSS-2, see Section IV. We further introduce a cutoff time t_c after which input neurons are not allowed to emit spike events and we do not expect the SNN to gain information afterwards. The spike encoding is illustrated in Fig. 2C. A sample $\tilde{y}_{n,\ell}$ (purple, dotted) is translated into spike times according to its distance to the reference points, e.g., the distance to χ_4 (blue, solid) results in a spike from input neuron $n_{4,\ell}^\mathrm{i}$ depicted in blue. The input neuron $n_{8,\ell}^\mathrm{i}$, corresponding to χ_8 (yellow, dotted), remains silent.

b) Output Membrane-Decoding: Each of the 4 neurons in the readout layer is assigned to one element in the PAM-4 alphabet $\mathcal{A}$. We take the maximum membrane voltages produced over time, i.e., $\hat{s}_k = \max_t v_k(t)$, which are interpreted as $\log$-probabilities providing an SD on the PAM-4 symbols. Then, the symbol-wise HD is obtained by choosing the symbol of highest probability and the bitwise HD is obtained from the symbol decisions via the Gray label. Hence, the network learns to place its hidden layer spike events in time, such that the membrane trace of the correct output neuron is deflected upwards while the traces of the others are suppressed.

A. Training

Time-discretized SNNs are mathematically recurrent neural networks (RNNs) [19] and can be trained with the gradient-based backpropagation through time (BPTT). For this, the derivative of the spiking output of the LIF neurons with respect to their membrane potential has to be known. This derivative is ill-defined due to the threshold activation function. Often surrogate gradients, smoothing out the neurons' activation functions, are used to bypass this issue and allow backpropagating the gradient. Here, we rely on the SuperSpike [19] surrogate gradient. The model parameters are optimized by the Adam optimizer [20].

In the simulation, the SNN is integrated with a step size $\Delta t = 0.5\,\mu\mathrm{s}$ for $T = 30\,\mu\mathrm{s}$, suitable for BSS-2 (see Section IV). Our simulated SNN demappers are implemented using the `PyTorch`-based Norse [21] framework. To estimate the hardware gradient for the SNN demappers emulated on BSS-2 in continuous time with the BPTT algorithm, we discretize the hardware observables assuming the same step size, see Section IV. We allocate $\bar{N}^\mathrm{i} = 10$ input neurons per sample of which only a subset is active, depending on the sample value, see Fig. 5. We use the cross entropy on the max-over-time voltage values as the objective function. The parameters of the SNN and input encoding are listed in Table II. In case of emulation on BSS-2, these parameters are used as calibration targets, resp. for ITL training (see Section IV).

IV. BRAINSCALES-2 NEUROMORPHIC SYSTEM

We now discuss the emulation of the SNN demappers on the BrainScaleS-2 (BSS-2) system [9].

Table II
PARAMETERS OF THE INPUT ENCODING AND SNN

Input Encoding		SNN	
α	$8\,\mu s$	N^{h}	40
o in SW	$0\,\mu s$	N^{o}	4
o on BSS-2	$1\,\mu s$	τ_{m}	$6\,\mu s$
$\bar{N}^{\mathrm{i}}$	10	τ_{s}	$6\,\mu s$
χ_i	$0.\bar{7}\cdot i$ with $i\in\mathbb{N}_0^{<\bar{N}^{\mathrm{i}}}$	v_{l}	0 a.u.
t_{c}	$15\,\mu s$	v_{r}	0 a.u.
Δt	$0.5\,\mu s$	ϑ	1 a.u.
T	$30\,\mu s$	R_{l}	1 a.u.
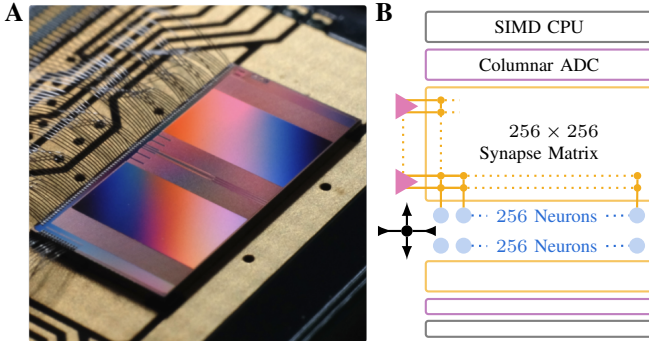
Figure 3. **(A)** The BrainScaleS-2 neuromorphic ASIC bonded onto its carrier board: the ASIC is about $4\,\mathrm{mm}\times 8\,\mathrm{mm}$ in size. It is organized in two hemispheres, each containing 256 spiking neurons in analog circuits. **(B)** The schematic of BSS-2: one parallel columnar ADC and one general-purpose single instruction, multiple data (SIMD) processor are available per hemisphere.

BSS-2, depicted in Fig. 3, is an accelerated neuromorphic mixed-signal hardware platform developed at Heidelberg University. Its analog neural network core features 512 adaptive exponential integrate-and-fire (AdEx) [22] neuron compartments on two chip hemispheres, each implemented as an analog circuit and emulated in continuous time. Each neuron compartment receives stimuli from a column of 256 synapses with 6-bit weights. Increased fan-in can be achieved by building larger 'logical' neurons from multiple compartments. Synapses can be configured to be inhibitory (negative sign) or excitatory (positive sign) in a row-wise fashion. To realize a specific network topology, on-chip spike routing connects neurons to target synapses. In addition, off-chip spikes are injected as an input stimulus for the network.

Neuron circuits can be configured to emulate LIF neurons as well as LI neurons by disabling the spiking mechanism. Each hemisphere on BSS-2 has one columnar ADC (CADC) to measure and digitize in parallel the membrane potentials of the neurons of each column in the synapse matrix with an effective sampling period of about $2\,\mu s$ including time stamping and writing to memory. CADC measurements and spike events can be recorded on an FPGA-managed dynamic random access memory (DRAM) and read out by the host computer.

Recording hardware observables allows for hardware ITL training. In the case of our SNN demapper on BSS-2, the forward pass is performed on BSS-2 and the hardware gradient is estimated on the host computer by utilizing the network's recorded hardware observables to calculate the weight updates [12], [13].

To obtain an equivalent experiment configuration on BSS-2, our software stack translates the high-level SNN experiment description to a data flow graph representation, places and routes neurons and synapses on the hardware substrate, and compiles stimulus inputs, recording settings and other runtime dynamics into an experiment program [23].

The analog circuits on BSS-2 are subject to device variations (fixed-pattern noise) that can be compensated for by calibration. Therefore, one part of the system configuration consists of a calibration data set that is loaded to obtain a chip operating point, which most closely resembles the desired target dynamics with minimal variation, e.g., with respect to model parameters such as neuron membrane time constants or synaptic efficacy.

To represent one signed software weight w_{sw} on BSS-2, two hardware synapses, with the respectively excitatory and inhibitory weights

$$w_{\mathrm{hw}}^{\mathrm{inh}} = \max\left(0, -w_{\mathrm{sw}}\right) \quad \text{and} \quad w_{\mathrm{hw}}^{\mathrm{exc}} = \max\left(0, w_{\mathrm{sw}}\right), \quad (15)$$

are allocated and constitute one signed hardware weight $w_{\mathrm{hw}} \in [-63, 63]$. We scale each weight w_{sw} linearly into a hardware-compatible range and round it to the nearest value representable on BSS-2. The batched input spikes are injected into BSS-2 and the SNN is emulated for $T = 30\,\mu s$ per batch entry, i.e., for demapping a single sample. During emulation, spike events are recorded and the CADC samples membrane voltages of the hidden neurons and the readout neurons. After the emulation, the host computer reads back and post-processes the recorded data. The post-processing step includes a linear interpolation to convert event-based CADC recordings to a `torch::Tensor` expressed on a fixed time grid. To facilitate hardware-ITL training on BSS-2, we utilized `hxtorch.snn` [23], a `PyTorch`-based [24] library that automates and abstracts away hardware-specific procedures and provides data conversions from and to `PyTorch`.

V. TRAINING AND TESTING

To measure the BER of the demappers against the noise-level in the IM/DD link, we train our models with successively increasing noise-levels σ^2. At each noise-level, we perform validation runs on independent data and store the model parameters of the best performing demapper. At the next noise-level, we restore the best model from the previous noise-level and continue training. This procedure is repeated for five different random seeds, affecting model initialization, IM/DD-data generation and sampling permutations. We select the best-performing demappers for each noise-level over the seeds according to their respective validation runs and benchmark the models on independent test data. The tests are run until a minimum of 2000 bit error events are encountered.

VI. RESULTS

In Figure 4, we compare our 7-tap SNN demapper emulated on the analog neuromorphic BSS-2 system ($\mathrm{SNN}^{\mathrm{hw}}$) to a 7-tap SNN demapper simulated in software ($\mathrm{SNN}^{\mathrm{sw}}$) in terms of BER versus noise-level σ^2 in the link. We benchmark our SNN performances against the LMMSE, with 7 taps (LE7) and
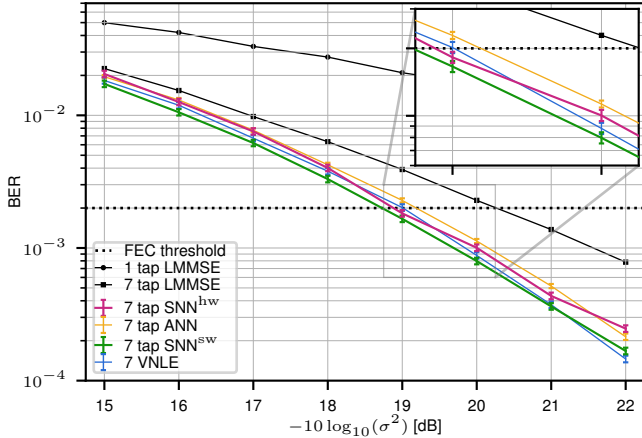
Figure 4. The BER of SNN equalizers in simulation and on the BSS-2 system over the noise-levels σ^2 in the IM/DD link compared to ANN, VNLE, and LMMSE reference equalizers. The error bars denote the 99% credibility intervals.

Table III
DEMAPPER DEFINITIONS

Name	Type	Layers	n_{tap}
LE1	LMMSE	$1 - 1$	1
LE7	LMMSE	$7 - 1$	7
ANN	ANN	$7 - 40 - 20 - 4$	7
VNLE	VNLE	-	7
SNN^{sw}	SNN in sim.	$70 - 40 - 4$	7
SNN^{hw}	SNN on BSS-2	$70 - 40 - 4$	7

without memory (LE1). As additional nonlinear references, we consider a 7-tap ANN demapper with two hidden layers (see Section II-D) and a 7-tap VNLE. All demapper configurations are specified in Table III.

Both the simulated SNN^{sw} demapper and the SNN^{hw} demapper on BSS-2 outperform the LMMSE demapper. At a pre-FEC BER of $2 \cdot 10^{-3}$, we observe a gain of about 1.5 dB of the SNN^{sw} demapper to the LE7 demapper and a gain of 0.5 dB to the nonlinear ANN demapper. Compared to the VNLE demapper, the SNN^{sw} demapper shows superior performance for noise levels higher than -21 dB, in particular at the considered pre-FEC BER threshold, it shows a 0.3 dB improvement, however, for noise levels lower than -21 dB the VNLE demapper achieves a lower BER.

The SNN^{hw} demapper on BSS-2 approaches the performance observed with the simulated SNN and only suffers from a small hardware penalty with respect to the SNN^{sw} of about 0.2 dB at a BER of $2 \cdot 10^{-3}$, outperforming all reference strategies.

In Fig. 5A, we visualize the process of joint equalization and demapping on BSS-2 on four different samples. The upper row indicates the sample set $\tilde{\boldsymbol{y}}_n$, with the sample of interest $\tilde{y}_n$ highlighted. Each sample in this set is translated to spike times of 10 input neurons, depicted in the second row. For $n_{\text{tap}} = 7$, the hidden LIF layer receives spike events from 70 input neurons, of which the majority are silent due to a cutoff time of 15 µs (see Section III). These input spike events activate the 40 LIF neurons in the hidden layer, exciting them

to emit spike events themselves as shown in the third row. These spikes events constitute a meaningful pattern, driving the membrane voltage of the correct LI output neuron to the maximum voltage value over time, from which the bits are inferred via an HD. This behavior is observed in the analog membrane traces in the lowermost row. The membrane voltage of the readout neuron corresponding to the estimated symbol is deflected upwards while the others drop below zero and hence do not intervene in the decision. Note that the dynamics visualized in each column from the second to fourth row all happen simultaneously in BSS-2's analog circuits.

The weight matrices learned on BSS-2 are shown in Fig. 5B. The input-to-hidden weight matrix w_{ij}^{ih} shows a greater weight magnitude for rows with indices $i \in [30, 39]$. This is expected as these rows receive the input spike events encoding the most significant sample to demap $\tilde{y}_n$ in the innermost tap. For the outer rows, one can observe a pattern repeating with the number of input rows per sample, $\bar{N}^{\text{i}} = 10$. The lower plot depicts the hidden-to-output weight matrix w_{jk}^{ho}.

VII. CONCLUSION

This work successfully showcases the implementation of SNN-based joint equalization and demapping emulated on the accelerated analog neuromorphic hardware system BSS-2. Our demapper on BSS-2 approaches the performance of an SNN demapper simulated in software while outperforming an LMMSE equalizer and performing better than a nonlinear ANN reference demapper, both with the same number of taps. A gain of 1.5 dB at a BER of $2 \cdot 10^{-3}$ of the simulated SNN over the LMMSE clearly demonstrates the nonlinear processing capability of the SNN demapper. A small hardware penalty of about 0.2 dB at the same BER with respect to the SNN simulated in software is observed and is attributed to hardware imperfections like noise in the physical substrate, fixed-pattern noise artifacts of the production process, and potentially a sub-optimal hardware operation point. Typically, the fixed-pattern noise effects are widely absorbed by gradient-based training. An additional cause might be the limited precision of 6-bit hardware weights and the 8-bit CADC. Despite having multiple sources of noise and loss of information owing to limited precision, the SNN demapper on BSS-2 shows an excellent performance and resilience to hardware impairments. We conjecture that the chosen size of the SNN with 40 hidden neurons ensures a robust behavior by encoding information redundantly. Accordingly, we expect to observe a larger hardware penalty as the number of hidden neurons decreases. An interesting direction for future research is to investigate how the complexity of the SNN on BSS-2 can be reduced while maintaining its performance.

With the implementation at hand, the equalization and demapping of a single sample take about $T = 30$ µs. Therefore, the BSS-2 platform supports a maximum symbol rate of 30 kBd. However, this upper bound is due to the specific design target of BSS-2 as a general purpose experimentation platform and does not follow from an intrinsic limitation of the underlying complementary metal–oxide–semiconductor (CMOS) technology itself. Significantly faster inference, and
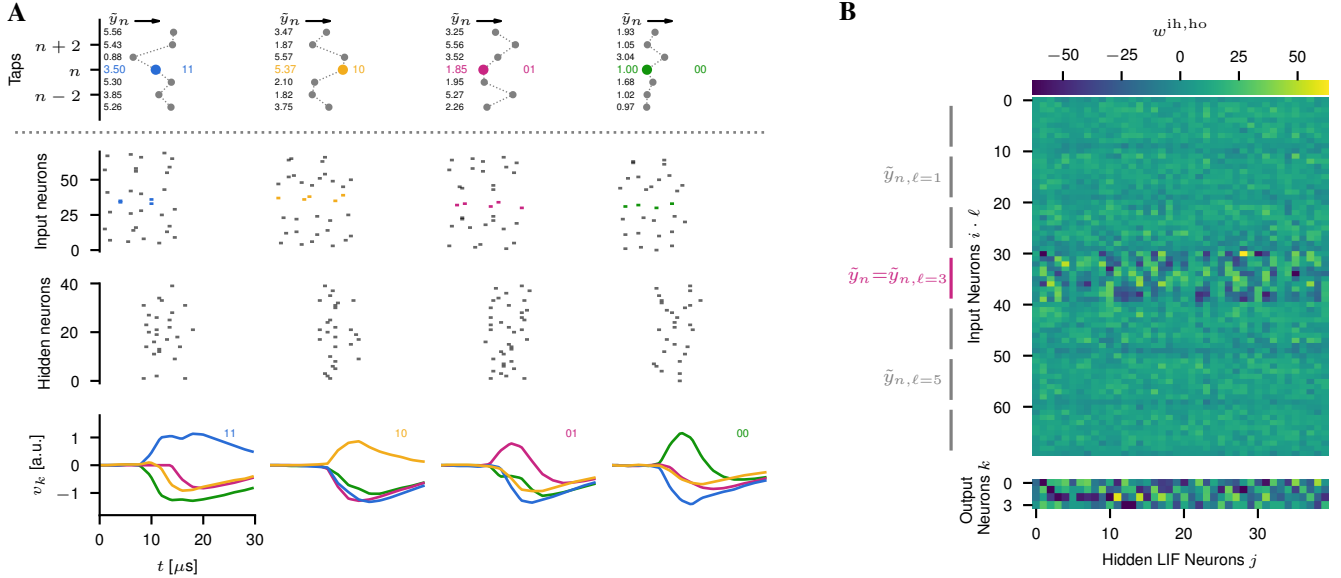
Figure 5. **(A)** Four different examples visualizing the bit inference process on BSS-2. The upper row depicts the chunk of samples used for demapping the innermost sample (colored). This chunk is translated to the spike events of the input neurons (10 per sample) as shown in the the second row. The third rows depicted the corresponding spike events of the hidden LIF neurons. In the last row, the membrane voltage traces of the four LI neurons are plotted. The output neuron corresponding to the estimated symbol produces the maximum voltage value over time. A Gray demapper provides bit decisions. **(B)** The SNN demapper's learned synapse matrices on BSS-2. The upper matrix shows the weights from the input to the hidden layer. Note, each of the $n_{\text{tap}} = 7$ samples has 10 input neurons assigned, resulting in 70 input rows. The rows corresponding to the middle sample $\tilde{y}_n$ contribute most to the decision. The lower matrix shows the weights from the hidden to the output layer.

thus throughput, might be achieved by accelerating the emulation of the LIF dynamics. [25] presented a neuromorphic ASIC exhibiting an acceleration of up to two additional orders of magnitude (OOMs) with respect to BSS-2. Given the fact that the cited implementation was fabricated in a 180 nm CMOS process, it is reasonable to assume that a modern FinFET process could potentially gain at least another 2 OOMs. This would result in a processing time in the order of nanoseconds per sample. The throughput can further be increased by parallelization. Several spiking network cores could be deployed in parallel, of which each could process multiple samples on the same physical substrate at once. To get nanoseconds per sample to 200 Gbit/s, a parallel processing factor of a few hundreds is enough, which is similar to the time-interleaving of multiple analog-to-digital converters (ADCs) used in standard optical DSP solutions [26].

The spatio-temporal sparsity of SNNs promises an intrinsically favorable energy footprint when contrasted to traditional ANN-based solutions – especially when combined with analog IMC [6]. Currently, the power consumption is dominated by I/O as well as the clock distribution and biasing of the individual subsystems – a fact largely attributed to the flexible general-purpose approach of BSS-2. Optimizing or omitting these subsystems in future, more specialized ASICs could dramatically reduce the overall energy footprint.

Future research aims to increase hardware resource efficiency by decreasing the architectural SNN complexity and investigate feature sharing in order to increase the throughput by parallelization. Importantly, the power consumption of neuromorphic signal processing shall be analyzed, compared to a digital implementation, and optimized by minimization of the firing activity of the neurons and efficient design of the input and output interfaces of the SNN.

The presented results demonstrate that electrical neuromorphic hardware can implement signal processing with the accuracy required in optical transceivers. To successfully integrate SNN equalization in optical transceivers, efficient conversion of received signals into input spikes must be researched.

REFERENCES

[1] B. J. Shastri *et al.*, "Photonics for artificial intelligence and neuromorphic computing," *Nature Photonics*, vol. 15, no. 2, 2021.

[2] S. Li, S. Dev, S. Kühl, K. Jamshidi, and S. Pachnicke, "Micro-ring resonator based photonic reservoir computing for PAM equalization," *IEEE Photonics Technology Letters*, vol. 33, no. 18, pp. 978–981, 2021.

[3] S. M. Ranzini, R. Dischler, F. Da Ros, H. Bülow, and D. Zibar, "Experimental investigation of optoelectronic receiver with reservoir computing in short reach optical fiber communications," *Journal of Lightwave Technology*, vol. 39, no. 8, pp. 2460–2467, 2021.

[4] K. Sozos, A. Bogris, P. Bienstman, and C. Mesaritakis, "Photonic reservoir computing based on optical filters in a loop as a high performance and low-power consumption equalizer for 100 Gbaud direct detection systems," in *2021 European Conference on Optical Communication (ECOC)*. IEEE, 2021, pp. 1–4.

[5] G. Caruso, I. N. Cano, D. Nesset, G. Talli, and R. Gaudino, "Real-time 100Gb/s downstream PAM4 PON link with 34 dB power budget," in *European Conference on Optical Communication (ECOC)*, Switzerland, Basel, Sep. 2022, paper Tu4C.3.

[6] G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, K. Virwani, M. Ishii, P. Narayanan, A. Fumarola *et al.*, "Neuromorphic computing using non-volatile memory," *Advances in Physics: X*, vol. 2, no. 1, pp. 89–124, 2017.

[7] A. Eldebiky, G. Böcherer, G. L. Zhang, B. Li, M. Schädler, S. Calabrò, and U. Schlichtmann, "Power-efficient and robust nonlinear demapper for 64QAM using in-memory computing," in *European Conference on Optical Communication (ECOC)*, Switzerland, Basel, Sep. 2022, paper We3C.3.

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

[9] C. Pehle, S. Billaudelle, B. Cramer, J. Kaiser, K. Schreiber, Y. Stradmann, J. Weis, A. Leibfried, E. Müller, and J. Schemmel, "The BrainScaleS-2 accelerated neuromorphic system with hybrid plasticity," *Front. Neurosci.*, vol. 16, 2022. doi: 10.3389/fnins.2022.795876

[10] E. Arnold, G. Böcherer, E. Müller, P. Spilger, J. Schemmel, S. Calabrò, and M. Kuschnerov, "Spiking neural network equalization for IM/DD optical communication," in *Optica Advanced Photonics Congress 2022*. Optica Publishing Group, 2022. doi: 10.1364/SPPCOM.2022.SpTu1J.2 Paper SpTu1J.2.

[11] E.-M. Bansbach, A. von Bank, and L. Schmalen, "Spiking neural network decision feedback equalization," *arXiv preprint*, 2022. [Online]. Available: https://arxiv.org/abs/2211.04756v2

[12] S. Schmitt, J. Klähn, G. Bellec, A. Grübl, M. Güttler, A. Hartel, S. Hartmann, D. Husmann, K. Husmann, S. Jeltsch, M. Kleider, C. Koke, A. Kononov, C. Mauch, E. Müller, P. Müller, J. Partzsch, M. A. Petrovici, B. Vogginger, S. Schiefer, S. Scholze, V. Thanasoulis, J. Schemmel, R. Legenstein, W. Maass, C. Mayr, and K. Meier, "Neuromorphic hardware in the loop: Training a deep spiking network on the BrainScaleS wafer-scale system," in *Proceedings of the 2017 IEEE International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2227–2234. doi: 10.1109/IJCNN.2017.7966125

[13] B. Cramer, S. Billaudelle, S. Kanya, A. Leibfried, A. Grübl, V. Karasenko, C. Pehle, K. Schreiber, Y. Stradmann, J. Weis *et al.*, "Surrogate gradients for analog neuromorphic computing," *Proceedings of the National Academy of Sciences*, vol. 119, no. 4, 2022.

[14] E. Arnold, G. Böcherer, E. Müller, P. Spilger, J. Schemmel, S. Calabrò, and M. Kuschnerov, "Spiking neural network equalization on neuromorphic hardware for IM/DD optical communication," in *European Conference on Optical Communication (ECOC) 2022*. Optica Publishing Group, Jun. 2022. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=ECEOC-2022-Th1C.5

[15] D. A. de Arruda Mello and F. A. Barbosa, *Digital Coherent Optical Systems: Architecture and Algorithms*. Springer Nature, 2021.

[16] ITU-T, "Recommendation ITU-T G.9804.3: 50-Gigabit-capable passive optical networks (50G-PON): Physical media dependent (PMD) layer specification," Sep. 2021. [Online]. Available: https://www.itu.int/rec/T-REC-G.9804.3-202109-I

[17] S. Benedetto and E. Biglieri, *Principles of digital transmission: with wireless applications*. Springer Science & Business Media, 1999.

[18] M. Schädler, G. Böcherer, and S. Pachnicke, "Soft-demapping for short reach optical communication: A comparison of deep neural networks and Volterra series," *Journal of Lightwave Technology*, vol. 39, no. 10, pp. 3095–3105, 2021.

[19] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, 2019. doi: 10.1109/MSP.2019.2931595

[20] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[21] C. Phttps://www.itu.int/rec/T-REC-G.9804.3-202109-Ie, hle and J. E. Pedersen, "Norse — A deep learning library for spiking neural networks," Jan. 2021, documentation: https://norse.ai/docs/. [Online]. Available: https://doi.org/10.5281/zenodo.4422025

[22] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *Journal of Neurophysiology*, vol. 94, no. 5, pp. 3637–3642, 2005. doi: 10.1152/jn.00686.2005

[23] E. Müller, E. Arnold, O. Breitwieser, M. Czierlinski, A. Emmel, J. Kaiser, C. Mauch, S. Schmitt, P. Spilger, R. Stock, Y. Stradmann, J. Weis, A. Baumbach, S. Billaudelle, B. Cramer, F. Ebert, J. Göltz, J. Ilmberger, V. Karasenko, M. Kleider, A. Leibfried, C. Pehle, and J. Schemmel, "A scalable approach to modeling on accelerated neuromorphic hardware," *Front. Neurosci.*, vol. 16, 2022. doi: 10.3389/fnins.2022.884128

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS 2017 Workshop on Autodiff*, 2017.

[25] J. Schemmel, A. Grübl, K. Meier, and E. Muller, "Implementing synaptic plasticity in a VLSI spiking neural network model," in *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN)*. IEEE Press, 2006. doi: 10.1109/IJCNN.2006.246651

[26] C. Laperle and M. O'Sullivan, "Advances in high-speed dacs, adcs, and dsp for optical coherent transceivers," *Journal of Lightwave Technology*, vol. 32, no. 4, pp. 629–643, 2014. doi: 10.1109/JLT.2013.2284134