Power-Aware Training for Energy-Efficient Printed Neuromorphic Circuits

Haibin Zhao[1♣], Priyanjana Pal[1♣], Michael Hefenbrock[2]
Michael Beigl[1], and Mehdi B. Tahoori[1]
[1]*Karlsruhe Institute of Technology,* [2]*RevoAI GmbH*
[1]{haibin.zhao, priyanjana.pal, michael.beigl, mehdi.tahoori}@kit.edu
[2]michael.hefenbrock@revoai.de

***Abstract*—There is an increasing demand for next-generation flexible electronics in emerging low-cost applications such as smart packaging and smart bandages, where conventional silicon electronics cannot enter due to cost and form factor. In these domains, ultra-low-cost, high flexibility, and customizability are required. In this regard, printed electronics emerge as a complementary solution offering the aforementioned properties. To respect the constraints in those application scenarios and equip printed devices with the fundamental capability to process information, analog printed neuromorphic circuits offer multiple advantages, including strong expressiveness, streamlined circuit primitives, and a highly efficient machine learning-based design process. In this work, we focus on designing low-power printed neuromorphic circuits at the algorithmic level. By developing accurate power models for the circuit primitives, the power consumption can be considered into the design process. Subsequently, Pareto analysis is employed to examine the relationship between accuracy and power consumption. Experimental results reveal that, with the proposed approach, $2\times$ reduction of the power consumption can be realized while maintaining 95% of classification accuracy. This approach has significant implications for the future development of energy-efficient printed neuromorphic circuits and their potential applications in IoT and AI intersections.**

I. Introduction

As the proliferation of the Internet of Things (IoT) and artificial intelligence (AI) continues, numerous emerging applications, including smart packaging [1] and smart bandages [2], necessitate the implementation of ultra-low-cost, super-soft, and highly customizable electronics for the measuring and processing signals at the edge [3]. However, even the cheapest traditional silicon-based electronics, such as Application-Specific Integrated Circuits (ASICs)[1] or Micro-controller Units[2], are unable to achieve the desired cost efficiency and flexibility. In this regard, printed electronics (PE) is considered as a viable complementary solution. Since PE can be manufactured additively, highly bespoke circuits can be fabricated at (sub-cent) ultra-low costs, regardless of the volume of demand. Furthermore, by selecting appropriate materials and substrates, printed devices can also exhibit highly flexibility [4, 5] and biocompatibility [6].

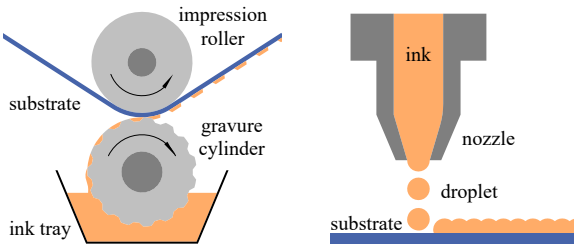
To provide printed devices with the foundational capacity to handle basic sensor processing tasks, such as classification, the incorporation of printed computing circuits become necessary. Considering the characteristics of PE and the necessity for low-cost in the application scenario, analog computing is typically adopted. Because it can significantly reduce the device counts by eliminating various components, such as analog-digital converters (ADCs), and simplifying arithmetic operations. Among the various analog computing paradigms, analog neuromorphic circuits [7] are deemed a favorable option owing to their strong (nonlinear) expressiveness [8], streamlined circuit primitives, and machine learning (ML) based off-device design and optimization. Printed neuromorphic circuits refer to printed circuits that perform equivalent operations to those in articifial neural networks (ANNs), primarily including weighted-sums and nonlinear activations. The weighted-sum operations are realized through *resistor crossbars*, while nonlinear activations are generally facilitated by *inverter-based circuitry*. In addition, as crossbars can only emulate positive weights, *negative weight circuits* for converting the inputs to negative ones are also indispensable.

The design of low-power electronics an ongoing area of interest, as it increases battery life and diminishes reliance on high-capacity power suppliers, which is critical for mobile, portable, and edge devices. Although many studies have been conducted for the low-power design of traditional electronics [9, 10], research for printed analog neuromorphic circuits has predominantly concentrated on printing technologies and materials [11], device and circuit configuration [7], dependability design [12], etc. In contrast, there has been a relative scarcity of studies aimed at designing energy-efficient printed neuromorphic circuits.

In this work, we present an approach for reducing the power consumption of given printed neuromorphic circuits from algorithmic level. By leveraging circuit principles and SPICE simulations, we establish accurate power consumption models for resistor crossbars and nonlinear circuits, which allows for estimating the power of the circuits during the designing and training phase. By modifying the objective of the circuit design, the power consumption of the circuit can be optimized in consideration of different trade-offs with respect to the circuits' performance (e.g., accuracy in classification task). Ultimately, a Pareto front is employed to enable the selection of the optimal accuracy-power trade-off based on design requirements.

In short, the contributions of this work are summarized as follows:

♣Authors contributed equally to this work.
[1]https://www.sigenics.com
[2]https://pic-microcontroller.com/world-top-10-cheapest-microcontrollers-mcus



(a) gravure printing (b) inkjet printing

Fig. 1. Schematic of typical printing technologies: (a) gravure printing and (b) inkjet printing.

- We modify the structure of printed neuromorphic circuits to enable lower power consumption.
- We develop accurate and differentiable power consumption models for printed crossbars and printed nonlinear circuits.
- We use the aforementioned models to estimate the power consumption and modify the design objective to jointly optimize for low energy consumption and circuit performance (e.g., accuracy).
- We train a great number of models to examine the trade-off between accuracy and power consumption. This allows us to identify Pareto-optimal solutions between accuracy and power consumption.

The experiments demonstrate that $2\times$ reduction in power could be achieved, while maintaining 95% of the baseline classification accuracy.

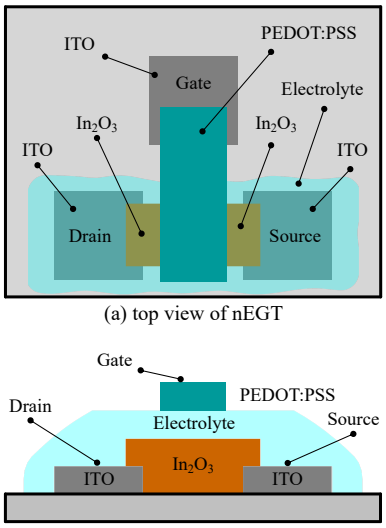
The rest of this paper is structured as follows: Sec. II introduces PE, printed neuromorphic circuits, and related works. Sec. III describes the development of power models and their integration into the design objective of the printed neuromorphic circuits. In Sec. IV, the proposed approach is evaluated and discussed. Finally, Sec. V concludes this work and discusses possible future works.

II. Preliminaries

A. Printed Electronics

Printed solution-processed electronics (PE) is an additive manufacturing process that encompasses a set of emerging technologies. Compared to conventional lithograph-based silicon electronics, PE requires less infrastructure and procedures for fabrication, and thus, exhibits lower manufacturing cost. By selecting appropriate printing technologies, printed electronic devices can be adapted to various production quantities and various requirements on device precision. In addition, diverse selections of printing materials can provide printed devices with important features in next-generation electronics, such as flexibility [4] and bio-compatibility [6]. These unique advantages make PE a strong competitor to traditional silicon-based technologies in numerous emerging IoT applications, such as wearables devices [13], Radio-Frequency identification (RFID) [14], disposable electronics [15], and implantable sensors [16].

Although the best performance of printed devices is generally achieved using vacuum-deposited highly purified molecules, solution-processed methods like spin-coating and inkjet printing have gained significant interest due to their simple fabrication processes and therefore low manufacturing cost. Printing



(a) top view of nEGT

(b) front view of nEGT

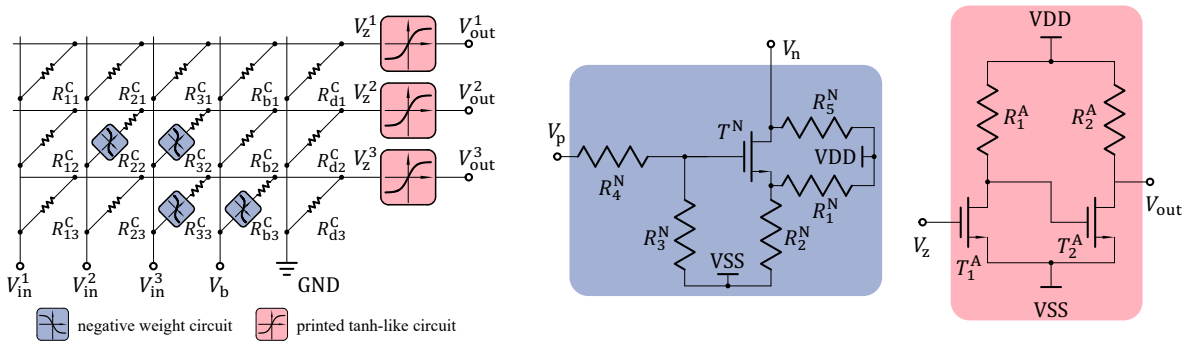
Fig. 2. Schematic of a printed nEGT: (a) top view and (b) front view.

technologies can be broadly divided into two categories based on production scale: a) replication printing, such as gravure printing (Fig. 1, left) and b) jet printing, such as inkjet printing (Fig. 1, right). The former is designed for high-volume printing, while the latter is suitable specifically for manufacturing highly customized circuits with low quantity demand. Hence, by rationally selecting and combining different printing technologies, PE can effectively cater to the requirements of manufacturing devices with different quantities and types.

Most state-of-the-art inkjet printed field-effect transistors (FETs) are implemented using organic materials, which use lithographically structured organic semiconductors as channels between source and drain electrodes. Generally, organic FET (OFET) structures are based on P-type materials, which have very low field-effect mobility [17] and operate at a high supply voltage range (≥ 25V). Consequently, OFET technology exhibits limited attractiveness for the intended application domains of PE, as the electronics in these scenarios are generally powered by energy harvesters or low-capacity power suppliers. Therefore, the low-power devices are more favored. In this respect, inorganic oxide semiconductors are more feasible candidates [18]. Current research in inorganic PE is focused on inkjet printing using N-type Electrolyte-Gated Transistor (nEGT) channels (Fig. 2), as no reliable P-type EGT has been reported yet [17, 19]. This may be due to the band-structure of electrons and holes, as the metallic oxide's band structure favors high field-effect charge carrier mobility for electrons compared to holes. Owing to their high gate-capacitance, nEGTs can be operated at a supply voltage of sub-1V, making them well-suited for applications powered by low-capacity batteries or reliant on energy harvesting systems.

B. Printed Analog Neuromorphic Circuits

To tackle challenges in targeted application domains, such as determining fruit ripeness [1], monitoring wound healing [2], or detecting human stress levels [20], printed circuits should be equipped with the foundational computing capabilities. Since the target tasks of PE generally exhibit low complexity and that PE is characterized by the large feature sizes, printed computing circuits should be designed with a few device counts. In this



(a) Exemplary printed neuron (b) Negative weight circuit (c) Printed tanh-like circuit

Fig. 3. Schematic of printed neuromorphic circuits. (a) Example of a 3-input and 3-output printed neuron based on crossbar array. (b) Schematic of inverter-based negative weight circuit. (c) Schematic of inverter-based printed tanh-like circuit.

regard, the circuits are favored to process signals directly in the analog domain, as the analog approach enables the exclusion of ADCs and significantly reduces the need for hundreds of transistors in digital circuits [7].

Within the realm of analog computing approaches, neuromorphic computing has emerged as one of the most favored solutions. This is primarily due to the development of AI, where ANNs have demonstrated excellent capabilities in solving highly complex problems [21]. More notably, these powerful capabilities rely solely on a series of simple primitive operations, namely weighted-sum operations and nonlinear activation functions. Therefore, this streamlined but efficient computing paradigm has resulted in a growing interest in printed neuromorphic circuits.

1) Hardware Primitives: Fig. 3 exemplifies the circuit schematics of a neuron in printed neuromorphic circuits. Fig. 3(a) represents a printed neuron with 3-input and 3-output based on crossbar array and printed tanh-like circuits (red blocks). Some negative weight circuits (blue blocks) are also incorporated in case required. Fig. 3(b) and (c) show the specific schematics of the negative weight circuit and the printed tanh-like circuit. In the following, we will provide a detailed introduction of these circuit primitives.

a) Crossbar array: The array-like structure in Fig. 3(a) shows the most fundamental architecture in printed neuromorphic circuits, which is the resistor crossbar array, resembling the weighted-sum operations in ANNs. This structure has been widely adopted in various applications, including in-memory computing [22] and ReRAM-based accelerators for ANNs [23]. According to Kirchhoff's law [24],

$$\sum_j \frac{V_{\text{in}}^j - V_{\text{z}}^1}{R_{j1}^{\text{C}}} + \frac{V_{\text{b}} - V_{\text{z}}^1}{R_{\text{b1}}^{\text{C}}} - \frac{V_{\text{z}}^1}{R_{\text{d1}}^{\text{C}}} = 0.$$

By expressing the resistance R as the corresponding conductance $g = 1/R$ and fixing $V_{\text{b}} = 1\text{V}$, this equation can be formulated to

$$V_{\text{z}}^1 = \sum_j \frac{g_{j1}^{\text{C}}}{G_1} V_{\text{in}}^j + \frac{g_{\text{b1}}^{\text{C}}}{G_1}, \tag{1}$$

where G_1 refers to the summed conductance of the first row in the crossbar array, i.e., $\sum_i g_{i1}^{\text{C}} + g_{\text{b1}} + g_{\text{d1}}$. In this case, the output voltage V_{z}^1 can be seen as the weighted-sum of the input voltages V_{in}^j, with conductances representing the weights and bias. Therefore, by designing proper conductance values, the desired weights and biases can be implemented.

b) Negative weight circuit: Since the conductances of the crossbar resistors can only represent positive weights, some resistors in Fig. 3(a) are prepended by inverter-based negative weight circuits, see Fig. 3(b) for detailed circuit schematic. This is done to emulate a multiplication with negative weights via inverted input voltages. The transfer characteristic of the negative weight circuit can be described by a modified negative $\tanh$ function, namely

$$V_{\text{n}} = \text{neg}(V_{\text{p}}) = -\left(\eta_1^{\text{N}} + \eta_2^{\text{N}} \cdot \tanh\left(\left(V_{\text{p}} - \eta_3^{\text{N}}\right) \cdot \eta_4^{\text{N}}\right)\right),$$

where $\boldsymbol{\eta}^{\text{N}} = [\eta_1^{\text{N}}, \eta_2^{\text{N}}, \eta_3^{\text{N}}, \eta_4^{\text{N}}]$ are auxiliary parameters that modify the original $\tanh$ function, which is ultimately determined by the physical quantities $\boldsymbol{q}^{\text{N}} = [R_1^{\text{N}}, R_2^{\text{N}}, R_3^{\text{N}}, R_4^{\text{N}}, R_5^{\text{N}}, W^{\text{N}}, L^{\text{N}}]$ in the circuit. Here, W^{N} and L^{N} are the geometric features of the transistor T^{N}.

c) Printed tanh-like circuit: Following the crossbar array, the signals are passed through the printed tanh-like circuits to resemble the activation functions in ANNs. The circuit diagram is illustrated in Fig. 3(c). Analogous to the negative weight circuit, the characteristic curve can be represented by a modified $\tanh$ function, i.e.,

$$V_{\text{out}} = \text{ptanh}(V_{\text{z}}) = \eta_1^{\text{A}} + \eta_2^{\text{A}} \cdot \tanh\left(\left(V_{\text{z}} - \eta_3^{\text{A}}\right) \cdot \eta_4^{\text{A}}\right)$$

with the auxiliary parameters $\boldsymbol{\eta}^{\text{A}} = [\eta_1^{\text{A}}, \eta_2^{\text{A}}, \eta_3^{\text{A}}, \eta_4^{\text{A}}]$ determined by $\boldsymbol{q}^{\text{A}} = [R_1^{\text{A}}, R_2^{\text{A}}, W_1^{\text{A}}, L_1^{\text{A}}, W_2^{\text{A}}, L_2^{\text{A}}]$. By optimizing $\boldsymbol{q}^{\text{A}}$, the shape of tanh-like function can be tuned to better fit specific target tasks.

2) Design and Optimization: By interconnecting the aforementioned circuit primitives, expressive computing functionalities can potentially be achieved. To fully harness the potential of the circuits, a design and optimization process is required. Considering the target application and the significant requirement for cost-efficiency, these circuits do not employ reconfigurable components for implementing on-device training during the usage. Rather, they are designed and optimized off-device at the software level. Subsequently, the fabrication process takes place after the circuit design has been completed. Note that, this does not limit applicability, as PE, due to additive manufacturing, allows for convenient and low-cost on-demand fabrication via diverse printing technologies.

To design and optimize printed neuromorphic circuits, such as determining suitable conductances in the crossbar array and

identifying the requirement of negative weight circuits, a ML-based design framework named printed neural network (pNN) is proposed [7]. This framework serve as simulation models for printed neuromorphic circuits. By training a pNN on the target dataset (of the target task), optimal physical quantities can be obtained and subsequently employed for the fabrication process.

In pNNs, the learnable parameter for each crossbar array is denoted as $\boldsymbol{\Theta} \in \mathbb{R}^{(M+2)\times N}$, where M and N refer to the number of inputs and outputs. The absolute value of each element in $\boldsymbol{\Theta}$ represents the conductance, while the sign of each element indicates whether a negative weight circuit needs to be pre-connected for emulating a negative weight. In this way, the output voltages of the crossbar array can be written as

$$\boldsymbol{V}_{\mathrm{z}} = \boldsymbol{V}_{\mathrm{in}} \cdot \left(\boldsymbol{W} \odot \mathbb{1}_{\{\boldsymbol{\Theta}\geq \mathbf{0}\}}\right) + \mathrm{neg}(\boldsymbol{V}_{\mathrm{in}}) \cdot \left(\boldsymbol{W} \odot \mathbb{1}_{\{\boldsymbol{\Theta}< \mathbf{0}\}}\right),$$

where $\boldsymbol{V}_{\mathrm{z}} = [V_{\mathrm{z}}^1, ..., V_{\mathrm{z}}^N]$ summarizes the weighted-summed voltages, $\boldsymbol{V}_{\mathrm{in}} = [V_{\mathrm{in}}^1, ..., V_{\mathrm{in}}^M, 1\mathrm{V}, 0\mathrm{V}]$ summarizes the input voltages extended with V_{b} and GND, and $\odot$ denotes the element-wise product. $\mathbb{1}_{\{\cdot\}}$ is an indicator function that returns 1 if the respective condition is true, else 0. It is applied element-wise on $\boldsymbol{\Theta}$. Moreover, the weight matrix $\boldsymbol{W}$ is composed by

$$\boldsymbol{W} = |\boldsymbol{\Theta}| \cdot \mathrm{diag}(|\boldsymbol{\Theta}|^{\top} \cdot \mathbf{1}_{M+2})^{-1} \in \mathbb{R}^{(M+2)\times N}, \qquad (2)$$

where $\mathbf{1}_{M+2} \in \mathbb{R}^{M+2}$ is a vector of all ones, $|\cdot|$ refers to an element-wise absolute operation, and $\mathrm{diag}(\cdot)$ yields a diagonal matrix from the given vector. Subsequently, the output voltages $\boldsymbol{V}_{\mathrm{z}}$ will pass through the ptanh function for activation, i.e., the output of the printed neuron $\boldsymbol{V}_{\mathrm{out}} = [V_{\mathrm{out}}^1, ..., V_{\mathrm{out}}^N]$ can be obtained through

$$\boldsymbol{V}_{\mathrm{out}} = \mathrm{ptanh}\left(\boldsymbol{V}_{\mathrm{z}}\right).$$

By cascading the weighted-sum operations and activation functions multiple times, deeper and more complicated printed neuromorphic circuits can be established.

Apart from $\boldsymbol{\Theta}$, the physical quantities $\boldsymbol{q}^{\mathrm{N}}$ and $\boldsymbol{q}^{\mathrm{A}}$ of the nonlinear circuits can also be learned [25]. By employing differentiable surrogate nonlinear circuit models, $\boldsymbol{q}^{\mathrm{N}}$ and $\boldsymbol{q}^{\mathrm{A}}$ can be transformed into the auxiliary parameter $\boldsymbol{\eta}^{\mathrm{N}}$ and $\boldsymbol{\eta}^{\mathrm{A}}$, which can then be integrated into the inference process of pNNs. Consequently, the partial derivatives of the loss function with respect to $\boldsymbol{q}^{\mathrm{N}}$ and $\boldsymbol{q}^{\mathrm{A}}$ can be obtained through backpropagation [26], allowing them to be learned alongside $\boldsymbol{\Theta}$ through gradient-based optimization.

C. Low-Power Electronic Design

With the growing prominence of edge devices in IoT contexts, an increasing focus on low-power electronics design can be observed. Because this shift towards high power-efficiency not only improves user experience by prolonging device working time per charge, but also promotes environmental conservation.

Although neuromorphic computing has already been shown to be significantly more power-efficient compared to conventional approaches [27, 28], there is ongoing research to further reduce the power consumption of neuromorphic circuits. For example, [29] developed novel devices to decrease the power required for the activation functions, such as ReLU. The work of [30] utilized hardware-software co-design to optimize circuit structure for data flow in the computing process. Regarding computational paradigms, numerous studies have adopted brain-inspired spiking neural networks to minimize power consumption in analog [31], digital [32], or mixed-signal [33] domains. Moreover, various ML techniques can also indirectly contribute to the reduction of the circuit power consumption at the software level by reducing the complexity of the ML models, and thus the corresponding hardware. These approaches may involve network quantization [34], pruning [35], neural architecture search [36], and other techniques [37, 38].

Nevertheless, most of the existing research primarily addresses the reduction of circuit power as an implicit objective. In contrast, this work emphasizes modelling the power consumption of the circuits and explicitly incorporating these models into the circuit design objectives at the algorithmic level. Moreover, the research subject in this work is printed analog neuromorphic circuits implementing multilayer perceptrons (MLPs), where comparable methodologies remain scarce in the existing approaches.

III. Methodology

In many target applications of PE such as smart packaging, the printed devices are possibly disposable and consequently may not be accessible for recharging. Therefore, they are generally powered by their initial printed batteries [39] or printed energy harvesters [40]. In this case, the low power consumption of the circuit becomes particularly crucial. Moreover, due to resistive nature of weighted sum crossbar and lack of P-type transistors in this printed technology, the need for low-power design is even further justified.

In this work, we first modify the existing circuit structure in a more power-efficient way, and then propose power-aware training for pNN by explicitly integrating power models into the objective function. Specifically, we derive the accurate power models for the circuit primitives in the printed neuromorphic circuits. Afterwards, by integrating these models into the pNN framework, the power of the circuits can be estimated during the training process. Finally, by combining the original loss function (for classification accuracy) with the estimated power, a Pareto front of power-accuracy trade-offs can be established.

A. Modified Power-Efficient Circuit Structure

In Fig. 3(a), negative weight circuits are prepended to the respective resistors whenever negative weights are necessitated. However, this approach is suboptimal regarding power conservation, as some inputs are repetitively converted to their corresponding negatives (e.g., V_{in}^3 at R_{32}^{C} and R_{33}^{C}). To eliminate this redundancy and thus reduce the power, we modified the circuit design, as shown in Fig. 4. With this modified structure, only one single negative weight circuit is required for each input. Subsequently, resistors may be connected to either V_{in}^i or $\mathrm{neg}(V_{\mathrm{in}}^i)$, depending on the sign of the corresponding weights.

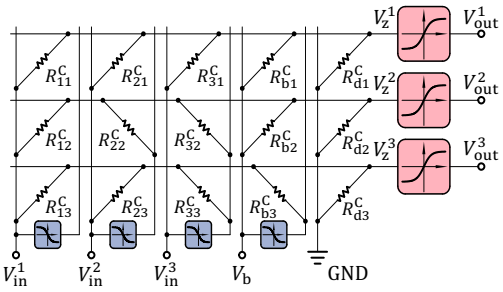


Fig. 4. Modified power-efficient design of a printed neuron.

To validate the new circuit design and assess other characteristics such as latency of both circuits (Fig. 3 and Fig. 4), we performed SPICE simulation with the printed Process Design Kit (pPDK) [41]. The new circuit structure yields the same output and similar latency as the previous design, however, the power consumption w.r.t. V_{in}^3 decreases due to the reduced number of negative weight circuits.

B. Power Consumption Model

Due to the structural simplicity of resistor crossbar arrays, we directly employ fundamental circuit formulae to derive analytical solutions for the power consumption. In contrast, due to the complexity of the nonlinear circuits, we obtain the power models by approximating data from SPICE simulation based on the pPDK [41]. Specifically, after the SPICE simulation with various circuit configurations, ANNs are employed to approximate the transformation from the physical quantity ($\boldsymbol{q}^{\text{N}}$ and $\boldsymbol{q}^{\text{A}}$) to the circuit power ($P^{\text{N}}$ and P^{A}). We refer to this ANN-based transformation as the *surrogate power consumption models* for the nonlinear circuits.

1) Power Consumption Model for the Crossbar: Due to the pure resistivity of the resistor crossbar array (excluding the negative weight circuits), the analytical power model can be directly obtained from the formula of electronic power. For each individual resistor, the power can be calculated by

$$P = \frac{\Delta V^2}{R} = \Delta V^2 \cdot g,$$

wherein ΔV refers to the potential difference between the two ends of the resistor. Therefore, the power consumption for the crossbar excluding negative weight circuits can be modeled as

$$\boldsymbol{P}^{\text{C}} = ((\tilde{\boldsymbol{V}}_{\text{in}} \odot \mathbb{1}_{\{\boldsymbol{\Theta} \geq \mathbf{0}\}} + \text{neg}(\tilde{\boldsymbol{V}}_{\text{in}}) \odot \mathbb{1}_{\{\boldsymbol{\Theta} < \mathbf{0}\}}) - \tilde{\boldsymbol{V}}_{\text{z}})^2 \odot |\boldsymbol{\Theta}|,$$

where $(\cdot)^2$ denotes an element-wise square operation, moreover,

$$\tilde{\boldsymbol{V}}_{\text{in}} = \left[\boldsymbol{V}_{\text{in}}^\top, \cdots, \boldsymbol{V}_{\text{in}}^\top\right] \in \mathbb{R}^{(M+2)\times N},$$

and

$$\tilde{\boldsymbol{V}}_{\text{z}} = \begin{bmatrix} \boldsymbol{V}_{\text{z}} \\ \vdots \\ \boldsymbol{V}_{\text{z}} \end{bmatrix} \in \mathbb{R}^{(M+2)\times N}.$$

In this way, each element in the matrix $\boldsymbol{P}^{\text{C}}$ represents the power of the corresponding resistor. By summing all elements in $\boldsymbol{P}^{\text{C}}$, the over all power consumption of the crossbar can be obtained by

$$\mathcal{P}^{\text{C}} = \mathbf{1}_{M+2}^\top \cdot \boldsymbol{P}^{\text{C}} \cdot \mathbf{1}_N, \tag{3}$$

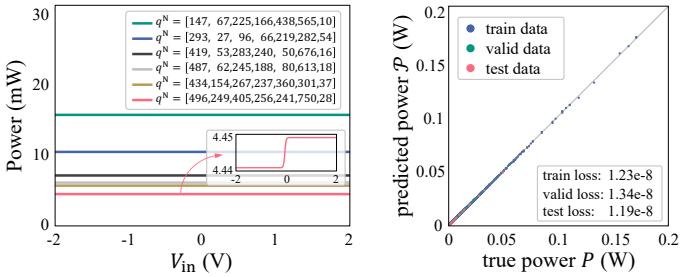


Fig. 5. Left: Power of some negative weight circuits with input voltages V_{in} ranging from -2V to 2V, the legend shows the configuration of the circuit components $\boldsymbol{q}^{\text{N}}$, the right bottom box shows the shape of the pink curve. Right: visualization of the results from the surrogate power consumption model. The x-axis and the y-axis refer to the true power P and predicted value $\mathcal{P}$. Blue, green, and red colors denotes the data from training, validation, and test sets.

where $\mathbf{1}_{M+2} \in \mathbb{R}^{M+2}$ and $\mathbf{1}_N \in \mathbb{R}^N$ are a vector with all the elements being 1.

2) Surrogate Power Consumption Models for Nonlinear Circuits: For the nonlinear circuits, i.e., negative weight circuits and printed tanh-like circuits, estimating the power consumption based on the physical quantities $\boldsymbol{q}^{\text{N}}$ and $\boldsymbol{q}^{\text{A}}$ is challenging. We therefore train ANNs to approximate the power consumption of these circuits based on SPICE simulations. Note that since the operations in ANNs are fully differentiable, the physical quantities can be optimized for reducing the circuit power through gradient-based algorithms.

Since the methodologies for both negative weight circuits and the printed tanh-like circuits are identical, we only describe our approach for the negative weight circuit as an example.

We firstly define the feasible design space $\mathbb{Q}^{\text{N}}$ to guarantee the desired negative $\tanh$ shapes of the characteristic curves. $\mathbb{Q}^{\text{N}}$ consists of MIN-MAX constraint on each physical quantity, i.e., $\boldsymbol{q}^{\text{N}} \in [\mathbf{q}_{\text{min}}^{\text{N}}, \mathbf{q}_{\text{max}}^{\text{N}}]$ and inequality constraints among individual values, i.e., $R_1^{\text{N}} > R_2^{\text{N}}$, $R_3^{\text{N}} > R_4^{\text{N}}$, and $W^{\text{N}} > L^{\text{N}}$. Subsequently, we employ a Quasi Monte-Carlo method with Sobol sequence to sample 10 000 points within the feasible space, with each point referring to a unique circuit configuration. Afterwards, we performed SPICE simulations using the pPDK [41] to gain the power consumption of each sampled circuit.

The left side in Fig. 5 exemplifies the power of multiple negative weight circuits, with input voltage V_{in} ranging from -2V to 2V. The legend denotes the corresponding circuit configuration $\boldsymbol{q}^{\text{N}}$. It is notable that, although the power varies with changing input voltage, as shown by the pink curve in the right bottom box, the variation is so small that the power consumption can be regarded as a constant w.r.t. the DC input voltage V_{in}. Moreover, due to the absence of a priori knowledge for the magnitude of input voltages, the distribution of the input voltages should be assumed as a uniform distribution ranging between -2V and 2V according to the principle of maximum entropy [42]. Consequently, the expected power consumption P^{N} is represented by the mean value w.r.t. input voltages.

After obtaining the sampled value $\boldsymbol{q}_i^{\text{N}}, i = 1, ..., 10\,000$ and the corresponding power consumption P_i^{N}, we can train an ANNs as the surrogate power consumption model, which is denoted by $\mathcal{P}^{\text{N}}(\boldsymbol{q}^{\text{N}})$.

To train the ANN, we randomly split the dataset $\{\boldsymbol{q}_i^{\text{N}}, P_i^{\text{N}}\}_{i=1}^{10\,000}$ into training set (70%), validation set (20%),

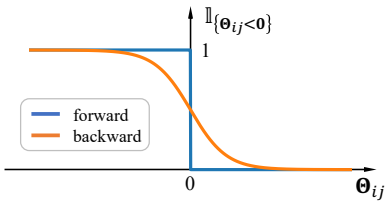


Fig. 6. Straight through gradient estimator for the soft-count of the negative weight circuits.

and test set (10%). The training set serves to guide the training process, the validation set stops the training to prevent overfitting, and the test set is used to evaluate the trained ANN.

To enhance performance of the ANN, we employ some typical techniques, such as data normalization, neural architecture search, and hyperparameter tuning on the learning rate. Finally, a 15-layer ANN is chosen as the surrogate power model.

The performance of the surrogate model is demonstrated on the right side of Fig. 5, where the horizontal axis denotes the true power consumption from SPICE simulation and the vertical axis refers to the predicted power from the surrogate model. We can qualitatively conclude that, the surrogate model generates acceptable power estimations. Moreover, the losses on training and test sets indicate that the model generalizes well.

C. Power Estimation for a Printed Neuron

Building upon the developed power consumption models, we are able to estimate the power of each printed neuron by accumulating the power of each circuit primitive, namely:

$$\mathcal{P} = \mathcal{P}^{\mathrm{C}} + N^{\mathrm{N}} \cdot \mathcal{P}^{\mathrm{N}} + N^{\mathrm{A}} \cdot \mathcal{P}^{\mathrm{A}}, \tag{4}$$

where N^{N} and N^{A} denote the number of negative weight circuits and printed tanh-like circuits. Moreover, $\mathcal{P}^{\mathrm{N}}$ and $\mathcal{P}^{\mathrm{A}}$ are the estimated power consumption from the surrogate power models.

It is notable that, according to the SPICE simulation, despite the similar inverter-based structures between negative weight circuits and printed tanh-like circuits, their power consumptions differ by orders of magnitude. Specifically, the power of the inverter circuits is at the mW level, whereas for the activation function is at the $\mu\mathrm{W}$ level. This difference can be attributed primarily to the feasible range of resistor values. Consequently, reducing the power consumption of negative weight circuits becomes even more significant. However, since we want to leverage gradient-based optimization to reduce the power consumption, we require useful gradient information of the power with respect to all our design parameters. Unfortunately, N^{N} in Eq. 4, representing the number of negative weight circuits, depends on $\boldsymbol{\Theta}$ but represents a piece-wise constant function. Specifically, N^{N} is expressed by

$$N^{\mathrm{N}} = \mathbf{1}_{M+2}^{\top} \cdot \operatorname{row\,max}\left\{\mathbb{1}_{\{\boldsymbol{\Theta}<0\}}\right\}, \tag{5}$$

where $\operatorname{row\,max}(\cdot)$ returns the row-wise maximum values. The blue curve in Fig. 6 visualizes the indicator function $\mathbb{1}_{\{\boldsymbol{\Theta}<0\}}$. It is evident that, except for $\boldsymbol{\Theta}_{ij} = 0$, all gradients are 0, meaning that, within the context of gradient-based optimization, $\boldsymbol{\Theta}$ will not be modified for the purpose of reducing N^{N}. To address this issue and enable the optimization of N^{N} through $\boldsymbol{\Theta}$, we introduce the *soft count* of negative weight circuits, denoted

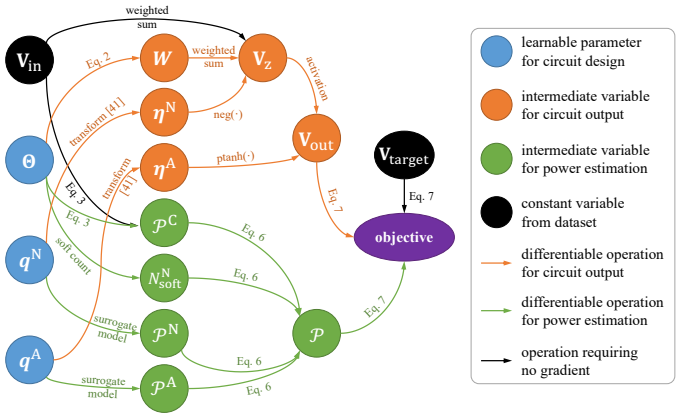


Fig. 7. Computation graph of the power-aware training of the printed neural networks with one neuron. The orange part refers to previous work, while the green part denotes the contribution of this work.

by $N^{\mathrm{N}}_{\mathrm{soft}}$. In the forward pass of the soft count, $N^{\mathrm{N}}_{\mathrm{soft}}$ is still calculated by Eq. 5, however, in the backpropagation, a relaxed function,

$$\mathbf{1}_{M+2}^{\top} \cdot \operatorname{row\,max}\left\{1 - \operatorname{sigmoid}(\boldsymbol{\Theta})\right\},$$

is employed to generate the gradient for updating $\boldsymbol{\Theta}$. Compared to Eq. 5, the indicator function is relaxed as a sigmoid function, as shown by the orange curve in Fig. 6. This kind of separate treatment for the forward and backward pass is also referred to as the straight-through gradient estimator [43].

By replacing N^{N} in Eq. 4 with soft count of the negative weight circuits, the resulting power estimation of the printed neuron can be formulated as

$$\mathcal{P} = \mathcal{P}^{\mathrm{C}} + N^{\mathrm{N}}_{\mathrm{soft}} \cdot \mathcal{P}^{\mathrm{N}} + N^{\mathrm{A}} \cdot \mathcal{P}^{\mathrm{A}}. \tag{6}$$

The computation graph for the complete power estimation is shown by the green part in Fig. 7. Note that this figure only represents the computation graph for one neuron. In case multiple neurons are adopted, the $\boldsymbol{V}_{\mathrm{out}}$ of one neuron will be passed to the next neuron as the input voltages. Consequently, the output of the last neuron will be regarded as the actual output. Moreover, the power consumption of all neurons will be summed up, serving as the final estimate for the power consumption.

D. Power-Aware Training

For training ANNs, loss functions are generally utilized to guide the optimization process and reflect the performance of the ANNs. A typical loss function for classification tasks is cross-entropy. However, to account for hardware limitations, such as minimal distinguishable voltages, a modified multi-class hinge loss [7] is employed to guide the training of pNNs. It can be expressed as

$$\begin{aligned} L(\boldsymbol{\Theta}, \boldsymbol{q}^{\mathrm{N}}, \boldsymbol{q}^{\mathrm{A}}) = &\left((m + T - \boldsymbol{V}_{\mathrm{out}}) \odot \boldsymbol{V}_{\mathrm{target}} \cdot \mathbf{1}_N\right)^{+} \\ &+ \left(\max\{(m + \boldsymbol{V}_{\mathrm{out}}) \odot (1 - \boldsymbol{V}_{\mathrm{target}})\right)^{+}, \end{aligned}$$

where T represents the measuring threshold, m denotes the sensing margin, $(\cdot)^{+} = \max\{0, \cdot\}$, and $\boldsymbol{V}_{\mathrm{target}}$ refers to the target class after one-hot encoding. This loss function encourages the voltage for the correct class to exceed $m + T$, while suppressing outputs corresponding to incorrect classes.

TABLE I
RESULT OF THE EXPERIMENT ON 13 BENCHMARK DATASETS

Dataset	$\alpha = 0$		$\alpha = 0.25$		$\alpha = 0.5$		$\alpha = 0.75$		$\alpha = 1$	
	Accuracy	Power (mW)	Accuracy	Power (mW)	Accuracy	Power (mW)	Accuracy	Power (mW)	Accuracy	Power (mW)
Acute Inflammation	1.000 ± 0.000	52.0 ± 2.5	1.000 ± 0.000	35.1 ± 6.7	1.000 ± 0.000	31.0 ± 3.3	0.999 ± 0.002	28.8 ± 2.0	0.461 ± 0.018	8.3 ± 1.3
Balance Scale	0.901 ± 0.017	47.3 ± 4.3	0.901 ± 0.028	43.4 ± 4.3	0.895 ± 0.016	39.5 ± 2.7	0.893 ± 0.020	33.4 ± 4.0	0.435 ± 0.018	10.0 ± 1.2
Breast Cancer Wisconsin	0.971 ± 0.001	114.2 ± 9.9	0.969 ± 0.002	82.0 ± 4.0	0.966 ± 0.001	57.9 ± 4.8	0.915 ± 0.020	45.8 ± 2.3	0.740 ± 0.021	9.7 ± 3.4
Cardiotocography	0.880 ± 0.007	196.6 ± 45.0	0.863 ± 0.019	125.2 ± 11.4	0.844 ± 0.011	97.0 ± 12.6	0.824 ± 0.014	71.7 ± 6.9	0.770 ± 0.003	16.1 ± 7.7
Energy Efficiency (y_1)	0.911 ± 0.019	76.7 ± 6.3	0.914 ± 0.013	63.6 ± 4.8	0.919 ± 0.013	52.3 ± 4.4	0.918 ± 0.014	46.6 ± 4.0	0.657 ± 0.010	11.8 ± 1.9
Energy Efficiency (y_2)	0.895 ± 0.016	80.9 ± 4.7	0.897 ± 0.008	59.6 ± 5.2	0.899 ± 0.006	52.7 ± 3.0	0.892 ± 0.010	48.4 ± 2.8	0.656 ± 0.009	11.5 ± 1.3
Iris	0.964 ± 0.005	52.1 ± 3.5	0.964 ± 0.003	42.4 ± 4.4	0.962 ± 0.004	40.6 ± 2.8	0.958 ± 0.009	32.2 ± 1.6	0.539 ± 0.011	9.2 ± 1.0
Mammographic Mass	0.791 ± 0.003	63.8 ± 6.8	0.789 ± 0.003	55.2 ± 3.0	0.792 ± 0.003	42.1 ± 2.9	0.789 ± 0.006	32.2 ± 3.0	0.635 ± 0.031	10.6 ± 0.7
Pendigits	0.617 ± 0.054	160.1 ± 19.5	0.536 ± 0.041	116.8 ± 7.4	0.479 ± 0.041	91.5 ± 5.5	0.371 ± 0.034	65.9 ± 12.3	0.068 ± 0.009	9.7 ± 11.9
Seeds	0.903 ± 0.031	80.4 ± 6.6	0.900 ± 0.015	65.0 ± 9.1	0.895 ± 0.017	56.2 ± 5.7	0.900 ± 0.015	50.4 ± 2.9	0.476 ± 0.035	12.8 ± 0.8
Tic-Tac-Toe Endgame	0.999 ± 0.001	115.3 ± 6.1	0.998 ± 0.001	92.9 ± 7.5	0.926 ± 0.047	59.7 ± 6.5	0.818 ± 0.004	40.6 ± 1.2	0.594 ± 0.052	14.4 ± 4.1
Vertebral Column (2 cl.)	0.829 ± 0.007	62.7 ± 4.0	0.827 ± 0.004	55.6 ± 3.0	0.824 ± 0.009	51.6 ± 2.9	0.768 ± 0.038	35.2 ± 4.1	0.664 ± 0.017	5.5 ± 1.4
Vertebral Column (3 cl.)	0.808 ± 0.010	68.2 ± 7.1	0.817 ± 0.006	57.7 ± 8.3	0.817 ± 0.007	50.3 ± 7.8	0.819 ± 0.004	42.3 ± 3.7	0.445 ± 0.007	10.8 ± 1.3
Average	**0.882 ± 0.013**	**90.0 ± 9.7**	**0.875 ± 0.011**	**68.8 ± 6.1**	**0.863 ± 0.013**	**55.6 ± 5.0**	**0.836 ± 0.015**	**44.1 ± 3.9**	**0.549 ± 0.018**	**10.8 ± 2.9**

Consequently, to jointly optimize both classification accuracy and power consumption, the power-aware training objective of the pNN is given by

$$\mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{q}^{\mathrm{N}}, \boldsymbol{q}^{\mathrm{A}}) = (1-\alpha) \cdot L(\boldsymbol{\Theta}, \boldsymbol{q}^{\mathrm{N}}, \boldsymbol{q}^{\mathrm{A}}) + \alpha \cdot \mathcal{P}, \quad (7)$$

where $\alpha \in \mathbb{R}^+$ denotes a scaling factor to express the trade-off between loss and power consumption. If $\alpha = 0$, the training objective entirely corresponds to the accuracy of the classification tasks. In this case, the trained pNN should achieve the highest accuracy, which can be regarded as the upper bound. However, since power consumption is totally ignored, the corresponding power should also be regarded as an upper bound. Conversely, if $\alpha = 1$, power $\mathcal{P}$ dominates the training objective whereas the accuracy is disregarded. Therefore, the trained pNNs may exhibit the lowest power consumption but, at the same time, also the poorest accuracy. Since the trade-off between power and accuracy is only implicitly influenced by α, and, considering that a specific trade-off will be chosen based on different application scenarios, we decide to train pNNs with different $\alpha \in [0, 1]$ and construct a Pareto front to facilitate the selection of various trade-offs with Pareto optimality.

E. Discussion

In this section, we established the accurate power consumption models for each circuit primitive in the printed neuromorphic circuits. Subsequently, we proposed the power-aware training for pNNs for optimizing both classification accuracy and power consumption jointly. From Eq. 1 we can see that, the weights are scale-invariant with respect to the resistances. Thus, the resistances can be scaled up to save power, while the weights remain unchanged. Consequently, for estimating the lowest power for the crossbar with given weights, the resistances are first up-scaled to the highest feasible values, which depends on the printing technology and the latency of the circuit. In this work, the maximal feasible resistance has been identified to be $1\,\mathrm{M}\Omega$ through SPICE simulation. Regarding the nonlinear circuits, it is notable that the changes in $\boldsymbol{q}^{\mathrm{N}}$ and $\boldsymbol{q}^{\mathrm{A}}$ not only impact the circuit power, but also influence their transfer characteristics, and thus, the accuracy of the classification. In this work, we employ the same $\boldsymbol{q}^{\mathrm{N}}$ and same $\boldsymbol{q}^{\mathrm{A}}$ shared inside an entire pNN, rather than allowing each neuron to have independent $\boldsymbol{q}^{\mathrm{N}}$ and $\boldsymbol{q}^{\mathrm{A}}$. Although the latter strategy offers higher degrees of freedom for optimization, it empirically yields worse result than the former strategy [25, 44].

IV. EVALUATION

To evaluate the effectiveness of the power-aware training of pNNs, we implemented the proposed approach[3] with PyTorch [45] and conduct experiments on the 13 benchmark datasets, which were also used in the related works, such as [12] and [25]. Moreover, these benchmark datasets exhibit a complexity and scenario that matches the target application domains of PE. The experiment is conducted at simulation level based on the pPDK [41]. The functionality of the printed neuromorphic hardware has been experimentally validated in [7] and [46].

A. Experiment Setup

We first split the datasets into training (60%), validation (20%), and test (20%) sets. Subsequently, we use a consistent topology (#$inputs$-3-#$outputs$) for all pNNs on each dataset. The learnable parameter $\boldsymbol{\Theta}$ is randomly initialized, while for the nonlinear circuits, $\boldsymbol{q}^{\mathrm{N}}$ and $\boldsymbol{q}^{\mathrm{A}}$ are initialized as $[463\Omega, 109\Omega, 10\mathrm{k}\Omega, 9\mathrm{k}\Omega, 24\mathrm{k}\Omega, 283\mu\mathrm{m}, 69\mu\mathrm{m}]$ and $[205\mathrm{M}\Omega, 7\mathrm{k}\Omega, 80\mu\mathrm{m}, 80\mu\mathrm{m}, 480\mu\mathrm{m}, 40\mu\mathrm{m}]$. The corresponding auxiliary parameters are $\boldsymbol{\eta}^{\mathrm{N}} = [-0.006, 1.024, 0.016, 1.006]$ and $\boldsymbol{\eta}^{\mathrm{A}} = [0.290, 0.710, -0.017, 20]$, respectively. Regarding the training, we employ full-batch training with the Adam [47] optimizer in default parameterization to update parameters in pNNs. To prevent overfitting, we calculated the loss on validation set for early-stopping [48] after each parameter update. We start with an initial learning rate of 0.1 and halve it after a patience (updates without improvement on objective function) of 100-epochs on the validation set. Additionally, the training process is stopped, when the learning rate was halved 10 times. In investigate the trade-off between accuracy and power, we uniformly select 50 values in $\alpha \in [0, 1]$.

The training is repeated 10 times (with seeds varying from 1 to 10) for different initialization for each value of α to make sure to achieve a sufficiently good solution for each value of α. Finally, the hardware-related hyperparameters in the loss

[3] https://github.com/Neuromophic/Power-Aware-Training

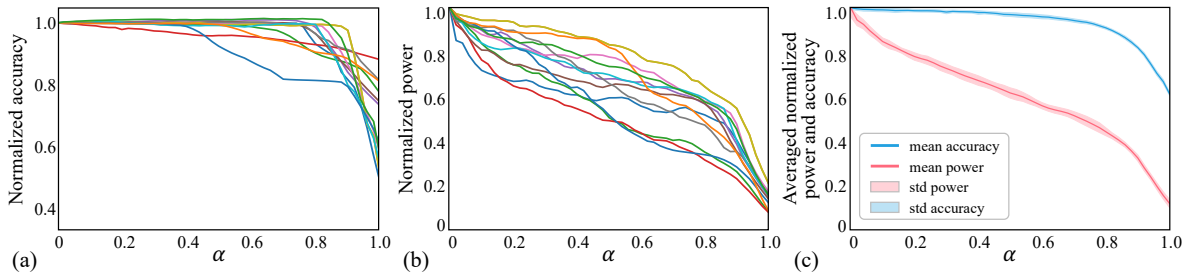


Fig. 8. Results of experiment with 50 different α values. (a) normalized accuracies of 13 tasks. Each task is indicated by a different color. (b) normalized power consumption of 13 circuits for the corresponding tasks. (c) averaged normalized accuracy and power, the curves and area denote the mean and standard deviation w.r.t. random seeds.

function, i.e., measuring threshold and margin, are chosen to be $T = 0.1$ and $m = 0.3$ to keep in line with other works [7].

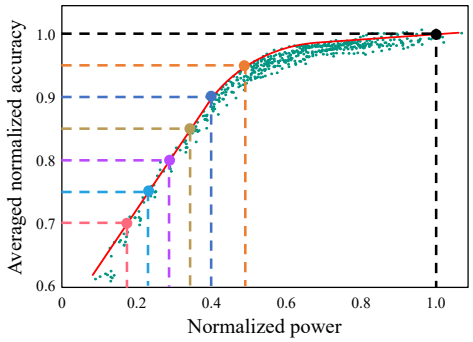


Fig. 9. Scatter plot of normalized accuracy versus power for all runs. The red curve displays the Pareto front, and the bold points denote different possible trade-offs on with Pareto optimality.

B. Result

After training, we evaluate the trained pNNs on the test sets. Tab. I reports the accuracies and powers with $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$.

To analyze the impact of α more clearly and to eliminate the disparate difficulties among different tasks, we normalize the accuracy by the baseline ($\alpha = 0$), which refers to the power-unaware training, and should theoretically achieve the best accuracy. Note that this is not always true in practice due to the complex nature of the non-convex optimization problem that neural network training resembles. The resulting curves are displayed in Fig. 8(a). Analogously, the power consumption is also normalized by the baseline power consumption. Because compared to the exact values, the relative power reduction serves as a more informative metric. The normalized powers are visualized in Fig. 8(b).

To investigate the effectiveness of the power-aware training within a comprehensive and generic scenarios, we calculate the averaged normalized accuracy across all tasks, which is intended to exemplify the expected performance of the pNNs on multiple datasets. The statistical result (w.r.t. 10 random seeds) of the averaged normalized accuracy (blue curve) and power (red curve) are summarized in Fig. 8(c).

In order to obtain the Pareto front, we plot the entirety of normalized powers versus their respective normalized accuracies for all runs (random seeds) and all values of α by the green points in Fig. 9. Subsequently, we can delineate the Pareto front by the red curve.

C. Discussion

As expected, for $\alpha = 0$ (no consideration of the power consumption) pNNs yield the highest accuracies and power consumption (Fig. 8). As α progressively increases to 1, both accuracy and power decline. However, the reduction in accuracy is less significant than that in power. This phenomenon enables the power conservation without a substantial drop in accuracy.

The Pareto front in Fig. 9 illustrates the relationship between power and accuracy. In comparison to power-unaware training (black point), if accuracy is allowed to decrease by 10%, $2.5\times$ power reduction can be achieved (blue point). Furthermore, if a 20% accuracy drop is allowed, the power consumption can be reduced to $3.6\times$. Other examples of trade-offs with Pareto optimality are reported in Tab. II. Beyond the examples listed, every point on the Pareto front may be chosen in consideration of the specific design requirements and application contexts.

It is notable that, when power decreases from 100% to 50%, the accuracy reduces in a gradual way. Conversely, a more substantial decrease in accuracy can be observed as the power budget continues to diminish from 50%. Thus, within the scope of this experiment, employing $2\times$ power reduction to achieve 95% accuracy may represent a reasonable trade-off.

TABLE II
ACCURACY-POWER TRADE-OFF

Accuracy (%)	100	95	90	85	80	75	70
Power (%)	100	50	40	34	28	23	18

V. CONCLUSION

In this work, we target the design of power-efficient printed analog neuromorphic circuits. By establishing analytical and ANN-based power consumption models, the circuit power can be explicitly incorporated into the design objective of the printed neuromorphic circuits. By introducing a variable trade-off factor in the training process of pNNs, a Pareto front can be drawn, from which any optimal trade-offs between accuracy and power can be chosen according to specific requirements or application scenarios.

Despite the preliminary progress made in this work, other methodologies for controlling the power could be explored in future research: In this study, the power is implicitly regulated by α. Nonetheless, in many design tasks, the circuits are constrained by predetermined power budgets. Consequently, future work may enable explicit constraints on power consumption.

ACKNOWLEDGMENT

This work has been partially supported by the Carl-Zeiss-Foundation as part of "stay young with robots" (JuBot) project and the European Research Council (ERC).

REFERENCES

[1] A. U. Alam *et al.*, "Fruit Quality Monitoring with Smart Packaging," *Sensors*, vol. 21, no. 4, p. 1509, 2021.

[2] Q. Sun *et al.*, "Smart Band-Aid: Multifunctional and Wearable Electronic Device for Self-Powered Motion Monitoring and Human-Machine Interaction," *Nano Energy*, vol. 92, p. 106840, 2022.

[3] F. Zhou and Y. Chai, "Near-sensor and In-sensor Computing," *Nature Electronics*, vol. 3, no. 11, pp. 664–671, 2020.

[4] I. I. Labiano *et al.*, "Flexible Inkjet-printed Graphene Antenna on Kapton," *Flexible and Printed Electronics*, vol. 6, no. 2, p. 025010, 2021.

[5] J. Chang *et al.*, "Challenges of Printed Electronics on Flexible Substrates," in *2012 IEEE 55th international midwest symposium on circuits and systems (MWSCAS)*. IEEE, 2012, pp. 582–585.

[6] A. Kaidarova *et al.*, "Wearable Multifunctional Printed Graphene Sensors," *NPJ Flexible Electronics*, vol. 3, no. 1, pp. 1–10, 2019.

[7] D. D. Weller *et al.*, "Realization and Training of an Inverter-based Printed Neuromorphic Computing System," *Scientific reports*, vol. 11, no. 1, pp. 1–13, 2021.

[8] Z. Yu *et al.*, "An Overview of Neuromorphic Computing for Artificial Intelligence Enabled Hardware-based Hopfield Neural Network," *IEEE Access*, vol. 8, pp. 67 085–67 099, 2020.

[9] M. Alioto, "Ultra-low Power VLSI Circuit Design Demystified and Explained: A Tutorial," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 1, pp. 3–29, 2012.

[10] S. Yan and E. Sanchez-Sinencio, "Low Voltage Analog Circuit Design Techniques: A Tutorial," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 83, no. 2, pp. 179–196, 2000.

[11] F. Rasheed *et al.*, "A Smooth EKV-based DC Model for Accurate Simulation of Printed Transistors and their Process Variations," *IEEE Transactions on Electron Devices*, vol. 65, no. 2, pp. 667–673, 2018.

[12] H. Zhao *et al.*, "Aging-Aware Training for Printed Neuromorphic Circuits," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '22)*, 2022.

[13] J. C. Costa *et al.*, "Flexible Sensors — From Materials to Applications," *Technologies*, vol. 7, no. 2, 2019.

[14] S. Kim, "Inkjet-Printed Electronics on Paper for RF Identification (RFID) and Sensing," *Electronics*, vol. 9, no. 10, p. 1636, 2020.

[15] R. Martins, I. Ferreira, and E. Fortunato, "Electronics with and on Paper," *physica status solidi (RRL)–Rapid Research Letters*, vol. 5, no. 9, pp. 332–335, 2011.

[16] R. Sobot, "Implantable Technology : History, Controversies, and Social Implications," *IEEE Technology and Society Magazine*, vol. 37, no. 4, pp. 35–45, 2018.

[17] S. K. Garlapati *et al.*, "Ink-Jet Printed CMOS Electronics from Oxide Semiconductors," *Small*, vol. 11, no. 29, pp. 3591–3596, 2015.

[18] F. Rasheed and M. Tahoori, *Compact Modeling and Physical Design Automation of Inkjet-Printed Electronics Technology*. Karlsruhe Institute of Technology (KIT), 2020.

[19] G. Cadilha Marques *et al.*, "Progress Report on "From Printed Electrolyte-Gated Metal-Oxide Devices to Circuits"," *Advanced Materials*, vol. 31, no. 26, p. 1806483, 2019.

[20] H. Zhao *et al.*, "Printed Electrodermal Activity Sensor with Optimized Filter for Stress Detection," in *International Symposium on Wearable Computers (ISWC'22), Atlanta, GA and Cambridge, UK, September 11-15, 2022*.

[21] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in Artificial Neural Network Applications: A Survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[22] A. Sebastian *et al.*, "Memory Devices and Applications for In-Memory Computing," *Nature nanotechnology*, vol. 15, no. 7, pp. 529–544, 2020.

[23] J. K. Eshraghian *et al.*, "Analog Weights in ReRAM DNN Accelerators," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2019, pp. 267–271.

[24] G. Kirchhoff, "LXIV. On a Deduction of Ohm's Laws, in connexion with the Theory of Electro-statics," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 37, no. 252, pp. 463–468, 1850.

[25] H. Zhao *et al.*, "Highly-Bespoke Robust Printed Neuromorphic Circuits," in *Design, Automation and Test in Europe (DATE)*. IEEE, 2023.

[26] Y. Chauvin and D. E. Rumelhart, *Backpropagation: Theory, Architectures, and Applications*. Psychology press, 2013.

[27] B. Li *et al.*, "Build Reliable and Efficient Neuromorphic Design with Memristor Technology," in *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, 2019, pp. 224–229.

[28] A. Basu *et al.*, "Low-Power, Adaptive Neuromorphic Systems: Recent Progress and Future Directions," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 1, pp. 6–27, 2018.

[29] V. Vadde *et al.*, "Power Efficient ReLU Design for Neuromorphic Computing Using Spin Hall Effect," *arXiv preprint arXiv:2303.06463*, 2023.

[30] Y.-H. Chen *et al.*, "Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks," *ACM SIGARCH computer architecture news*, vol. 44, no. 3, pp. 367–379, 2016.

[31] Y. Li *et al.*, "One Transistor One Electrolyte-Gated Transistor Based Spiking Neural Network for Power-Efficient Neuromorphic Computing System," *Advanced Functional Materials*, vol. 31, no. 26, p. 2100042, 2021.

[32] A. Bicaku *et al.*, "A Power-Efficient Neuromorphic Digital Implementation of Neural–Glial Interactions," *Journal of Low Power Electronics and Applications*, vol. 13, no. 1, p. 10, 2023.

[33] A. Rubino *et al.*, "Ultra-Low-Power FDSOI Neural Circuits for Extreme-Edge Neuromorphic Intelligence," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 1, pp. 45–56, 2020.

[34] A. Gholami *et al.*, "A Survey of Quantization Methods for Efficient Neural Network Inference," in *Low-Power Computer Vision*. Chapman and Hall/CRC, 2021, pp. 291–326.

[35] T. Liang *et al.*, "Pruning and Quantization for Deep Neural Network Acceleration: A Survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.

[36] T. Elsken *et al.*, "Neural Architecture Search: A Survey," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019.

[37] H. Chen *et al.*, "AdderNet: Do We Really Need Multiplications in Deep Learning?" in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1468–1477.

[38] J. Ran *et al.*, "PECAN: A Product-Quantized Content Addressable Memory Network," in *Design, Automation and Test in Europe (DATE)*. IEEE, 2023.

[39] A. M. Gaikwad *et al.*, "A Flexible High Potential Printed Battery for Powering Printed Electronics," *Applied Physics Letters*, vol. 102, no. 23, p. 104, 2013.

[40] T.-H. Lin *et al.*, "Wearable Inkjet Printed Energy Harvester," in *2017 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting*. IEEE, 2017, pp. 1613–1614.

[41] F. Rasheed *et al.*, "Variability Modeling for Printed Inorganic Electrolyte-gated Transistors and Circuits," *IEEE transactions on electron devices*, vol. 66, no. 1, pp. 146–152, 2018.

[42] E. T. Jaynes, "Information Theory and Statistical Mechanics," *Physical Review*, vol. 106, p. 620, 1957.

[43] Y. Bengio *et al.*, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," *arXiv preprint arXiv:1308.3432*, 2013.

[44] M. Tavakoli *et al.*, "SPLASH: Learnable Activation Functions for Improving Accuracy and Adversarial Robustness," *Neural Networks*, vol. 140, pp. 1–12, 2021.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An Imperative Style, High-performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[46] S. A. Singaraju *et al.*, "Artificial Neurons on Flexible Substrates: A Fully Printed Approach for Neuromorphic Sensing," *Sensors*, vol. 22, no. 11, p. 4000, 2022.

[47] D. P. Kingma *et al.*, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[48] L. Prechelt, "Automatic Early Stopping Using Cross Validation: Quantifying the Criteria," *Neural networks*, vol. 11, no. 4, pp. 761–767, 1998.