Accepted for publication at the 62nd Design Automation Conference (DAC '25)

Power-Constrained Printed Neuromorphic Hardware Training

Tara Gheshlaghi[1♣], 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]{tara.gheshlaghi, haibin.zhao, priyanjana.pal, michael.beigl, mehdi.tahoori}@kit.edu, [2]michael.hefenbrock@revoai.de

***Abstract*—With the rising demand for ultra-low-cost and flexible electronics in applications like smart packaging and wearable health monitoring, printed electronics provide an affordable, adaptable, and customizable alternative to conventional silicon. However, these systems often rely on printed batteries or energy harvesters with limited power capacity, making strict power budgets critical. Printed neuromorphic circuits (pNCs) are promising for their analog signal processing, reduced circuit complexity, and energy efficiency in low-power environments. Nonetheless, maintaining robust performance under strict power constraints remains challenging, necessitating advanced optimization techniques. In this work, we propose an augmented Lagrangian approach to enforce task-specific power constraints in pNCs, validated across 13 benchmark datasets. Our method preserves accuracy within strict power budgets while achieving Pareto-optimal power-accuracy trade-offs in a single training run. In contrast, the penalty-based method, which serves as the baseline, requires up to 150 runs per dataset to generate the Pareto front. For low-power scenarios ($\approx 20\%$ of the original power), our method demonstrates a $52\times$ improvement in accuracy-to-power ratio over the baseline. At higher power budgets ($\approx 80\%$ of the original power), it achieves a $59\times$ improvement, maintaining competitive performance. Experimental results demonstrate that our approach achieves 81.82% accuracy with p-tanh activation function (AF) at high power budgets and excels with p-Clipped_ReLU AF under low power constraints. This highlights the computational efficiency and effectiveness of our approach for power-constrained circuit design.**

I. Introduction

Although silicon-based electronics have witnessed continual advancements in power efficiency and transistor density, their adoption in numerous consumer edge applications such as smart packaging [1], smart bandages [2, 3], wearable devices, and other disposable electronics for consumer use [4]–[7] (as shown in Fig. 1[1]) remains a challenge. Even the most affordable traditional silicon-based electronics, such as Application-Specific Integrated Circuits (ASICs) [8] and microcontrollers [9], struggle to achieve the desired cost efficiency and adaptability for these resource-constrained applications. In this context, printed electronics (PE) emerge as a highly flexible [10, 11], biodegradabele [12] and cost-effective solution, through its additive manufacturing processes, thus enabling highly bespoke circuits at ultra-low costs. Furthermore, many of these applications must operate within a prescribed power budget to ensure a sufficient battery life, making energy efficiency [13] as crucial as cost and flexibility. So, by carefully selecting materials, substrates, and various optimization strategies [13], printed devices can also exhibit high flexibility [10, 11] and biocompatibility [12], making them ideal for constrained environments.

Fig. 1. Target application domains of printed electronics: (a) smart fruit package, (b) smart milk carton, (c) smart food packaging, (d) smart bandages.

Incorporating printed computing circuits is essential to equip printed devices with the basic capacity to process sensor data, such as performing classification tasks. Among various computing paradigms, analog computing is often chosen for PE due to its ultra-low cost and high flexibility in these applications. This approach allows for a reduction in device components, thus avoiding the expensive analog-digital converters (ADCs), and simplifies the execution of multiply-accumulate (MAC) operations. Analog neuromorphic circuits [14] stand out as an effective choice due to their expressiveness [15], using simplified circuit primitives, and benefit from machine learning (ML)-driven off-device design and optimization, making them highly suitable for resource-limited PE applications [13]. Moreover, the agile manufacturing process of PE allows highly tailored circuit designs that can meet the specific functional and physical requirements of target applications, reducing material waste, minimizing design iterations, and enabling rapid prototyping, making it ideal for ultra-low-cost, flexible solutions in constrained environments. A key advantage of this approach is the ability to treat critical components of printed analog neuromorphic circuits (pNCs), such as AFs, as learnable parameters, allowing the adaptation of both the shape and hardware realization of AFs, directly linking ML-optimized parameters to physical circuit characteristics.

Low-power electronics design remains a critical focus area due to its potential to extend device runtime and minimize its uses on high-capacity power sources. This is particularly important for portable and edge devices. While traditional electronics has extensively explored low-power solutions [13, 16, 17] these methods generally operate under soft power constraints, where reducing power consumption is desirable but not mandatory. In contrast, many target applications, especially power harvesters, wearable health monitoring devices, operate within strict power

[♣] Authors contributed equally to this work.

[1] Images are generated by the DALL-E AI tool.

and energy envelopes that demand hard constraints arising from different physical limitations like weight, cost, reliability over extended periods etc. This distinction is pivotal, as soft constraint solutions often fail to meet the rigid power requirements of such applications. In real-world scenarios, strict power constraints are crucial. For example, wearable health devices, such as ECG or glucose monitors, must operate continuously on compact batteries with limited capacity, requiring precise power management to ensure reliable functionality over their intended lifespan. Disposable medical sensors designed for short-term vital monitoring must similarly adhere to minimal power budgets. Supply chain IoT smart labels, tasked with tracking perishable goods' conditions like temperature over extended transit times, rely on small, non-rechargeable batteries. In outdoor applications such as GPS trackers powered by flexible solar panels, strict power limits must be maintained to guarantee performance in intermittent sunlight conditions.

In this work, we introduce an approach for designing pNCs under strict power constraints at the algorithmic level while ensuring high accuracy. Leveraging the augmented Lagrangian approach together with a surrogate power model of the circuit, we integrate strict power budgets directly into the optimization process, ensuring power consumption stays within pre-defined limits.

In short, the contributions of this work are:

- We propose an algorithmic framework based on the augmented Lagrangian method that enables training under strict power constraints while improving classification accuracy in a single training run, in contrast to penalty-based methods, which require multiple runs to approximate Pareto-optimal results.
- We developed data-driven surrogate power models for four different AFs, demonstrating that the choice of activation function and its learnable parameters have a significant impact on power consumption in pNCs.

Experiments show high accuracy within strict power budgets: p-tanh reaches 81.82% accuracy at 80% power, p-Clipped_ReLU maintains 74.65% at 20%, and p-ReLU reduces device count by up to 37% at low power budgets, highlighting its efficiency in power-constrained environments. Unlike the penalty-based approach, which requires numerous training runs to approximate the Pareto front, the augmented Lagrangian method achieves competitive results within a single training run, significantly reducing computational time and effort. This efficiency makes our approach more practical for circuit design applications.

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

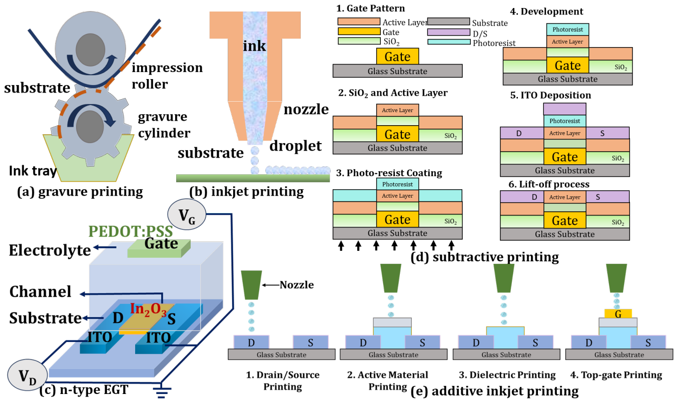

Fig. 2. Schematic of typical printing technologies: (a) gravure printing and (b) inkjet printing (c) a printed nEGT (d) Subtractive and (e) Additive printing process of PE.

II. Preliminaries

A. Printed Electronics

Printed solution-processed electronics (PE) is an additive manufacturing approach that requires less printing process steps than traditional silicon electronics, resulting in lightweight, ultra-low cost and flexible circuits. By selecting appropriate printing technologies, PE devices can be tailored to different production volumes and precision requirements. Furthermore, a wide range of printing materials enables important features for next-generation electronics, such as flexibility [10] and biocompatibility [12]. These makes PE a competitor in IoT applications like wearables [18], RFID [19], disposable electronics [20], energy harvesters, and implantable sensors [21].

Although vacuum-deposited methods achieve the best performance, simpler solution processes, such as spin-coating and inkjet printing are gaining popularity for their cost-effectiveness. PE technologies are categorized as either replication printing (e.g., gravure) for high-volume needs or jet printing (e.g., inkjet) for low-volume, bespoke fabricated circuits.

Most printed FETs are organic in nature and uses P-type materials that require high voltages ($\geq 25\text{V}$) [22]. This restricts their use in low-power applications. In contrast, inorganic N-type Electrolyte-Gated Transistors (nEGTs) offer higher electron mobility, enabling sub-1V operation [23], making nEGTs ideal for low-power devices reliant on small batteries or energy harvesters.

B. Printed Analog Neuromorphic Circuits

In the analog computing paradigm, neuromorphic computing has emerged as a preferred solution and is largely driven by advancements in artificial intelligence (AI), where artificial neural networks (ANNs) have outperformed in solving complex tasks [24].

1) Hardware Primitives: Fig. 3 (a) illustrates the circuit schematic of a pNC with 4-3-3 topology. Fig. 3(b) shows a 3-input, 3-output printed neuron, composed of a crossbar array and printed activation circuits (indicated in red). Additional circuits (highlighted in blue) are incorporated to handle negative weights, $\text{neg}(\cdot)$, by connecting resistors to either V_{in}^i or $\text{neg}(V_{\text{in}}^i)$, based on the sign of the corresponding weights. Figures 3(c)-(f) illustrate the schematics of various printed

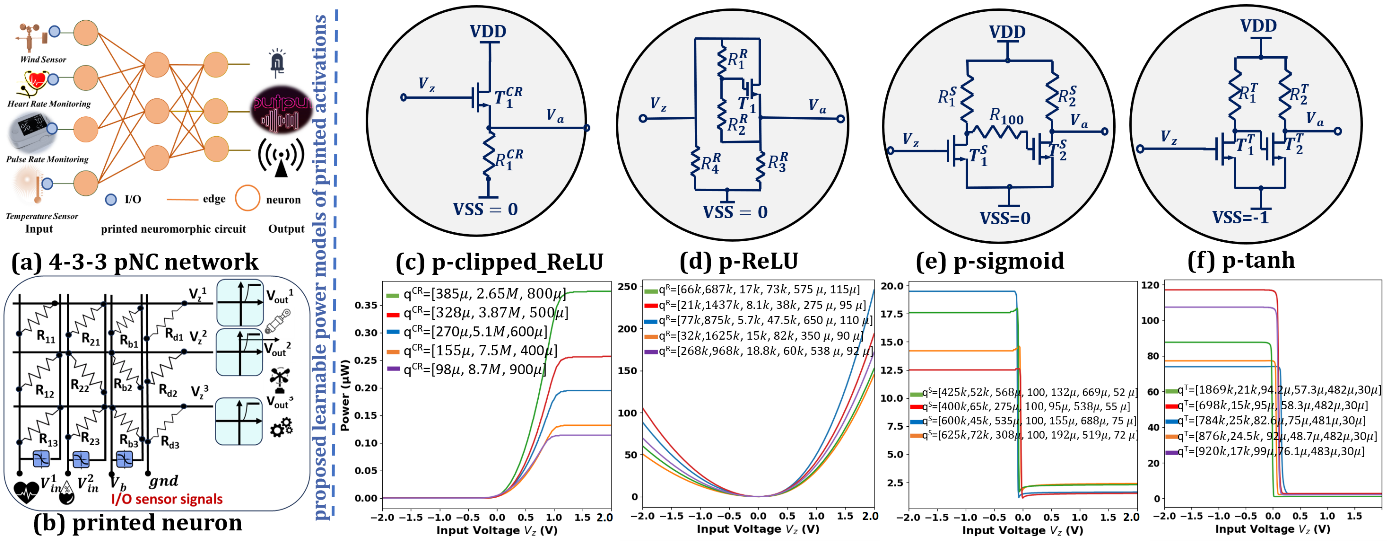

Fig. 3. Schematic of printed neuromorphic circuits. (a) Example of a 4-3-3 pNC network, (b) 3-input-3-output printed neuron based on crossbar array; surrogate power models of: (c) p-clipped_ReLU, (d) p-ReLU, (e) p-sigmoid and (f) p-tanh activation circuit.

activation circuits. In the following sections, we provide a comprehensive overview of these circuit primitives.

a) Crossbar array: The array in Fig. 3 (b) illustrates a basic resistor crossbar architecture in pNCs, emulating weighted-sum operations of ANNs [14]. Using Kirchhoff's law [25] and expressing resistance as conductance ($g = 1/R$), the output voltage V_z^1 is calculated as

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

where G_1 is the summed conductance of the first row in the crossbar, i.e., $\sum_i g_{i1}^\mathrm{C} + g_\mathrm{b1} + g_\mathrm{d1}$. Here, V_z^1 represents a weighted sum of input voltages V_in^j, with conductance values defining weights and bias, enabling control over weights and biases by adjusting these conductance values.

The resistive nature of the crossbar array enables a direct analytical power model using the electronic power formula. For each resistor, the power is calculated as

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

where ΔV is the potential difference across the resistor. Therefore, the power consumption of the crossbar, excluding negative weight circuits, is expressed as

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

Here, $\boldsymbol{\Theta} \in \mathbb{R}^{(M+2)\times N}$ is a matrix of learnable parameters for the crossbar array, where M and N denote the number of inputs and outputs, respectively. Each element in $\boldsymbol{\Theta}$ represents a surrogate conductance θ, where the absolute value $|\theta|$ encodes the physical conductance g of the corresponding resistor. Meanwhile, the sign of θ specifies whether a negative weight circuit must be pre-connected to emulate a negative weight. The operation $(\cdot)^2$ denotes an element-wise square, and $\mathbb{1}_{\{\cdot\}}$ is the element-wise indicator function that outputs 1 for true conditions and 0 otherwise. Each element in the matrix $\boldsymbol{P}^\mathrm{C}$ represents the power of the corresponding resistor. Thus, the total power of the crossbar is given by

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

where $\mathbf{1}_{M+2}$ and $\mathbf{1}_N$ are vectors of ones.

b) Printed non-linear transformation circuits: In crossbar resistor arrays, conductances represent only positive weights. To incorporate negative weights, some resistors are paired with inverter-based circuits [13], as shown in Fig. 3(b) (indicated by blue boxes), which emulate negative weights by inverting the input voltages V_i. After passing through the crossbar, signals are processed by printed AFs, shown in Fig. 3 (c-f) (top). These printed AFs simulate the standard AFs in ANNs, introducing the non-linearity necessary for modeling complex functions. It should also be noted that a significant portion of the overall power is often consumed by AFs, whose non-linear behaviors make the modeling and optimization more complex. The power consumption of these circuits depends on physical parameters like transistor size, resistor values, and input voltage, requiring surrogate models ($\mathcal{P}^N$ and $\mathcal{P}^T$) trained on the SPICE simulation data [13]. This is further complicated as AFs are learnable parameters, meaning that their shapes are adapted during optimization, influencing both accuracy and power.

C. Power-Constrained Electronic Design

While traditional neuromorphic computing has made significant strides in low-power design [26], most existing methods rely on best-effort power optimization with soft constraints, where minimizing power is desirable but not mandatory [27, 28]. However, many real-world applications require strict (hard) power constraints to ensure reliable operation within predefined energy budgets. For example, wearable health devices, disposable medical sensors, and IoT smart labels depend on limited power sources like small batteries or energy harvesters, making it critical to enforce precise power budgets.

Our work directly addresses this gap by modeling and optimizing strict power constraints during the training of pNCs. Unlike soft constraints, where achieving optimal power-performance trade-offs is the goal, we enforce hard constraints to achieve the best accuracy and efficiency without exceeding the predefined power budget. This approach integrates AFs as learnable parameters and uses SPICE simulation for accurate power estimation, ensuring a robust and practical methodology for power-constrained electronic design.

III. Methodology

To enable training under strict power constraints, we propose a training framework based on the augmented Lagrangian method. The primary objective is to maximize classification accuracy given the circuit parameters $\boldsymbol{\theta}$ and $\boldsymbol{q}$, while ensuring that power consumption below a predefined limit. Formally, this can be described by

$$\underset{\boldsymbol{\theta},\boldsymbol{q}}{\text{minimize}}\, \mathcal{L}(\mathcal{D},\boldsymbol{\theta},\boldsymbol{q}) \;\text{ s.t. }\; c(\boldsymbol{\theta},\boldsymbol{q}) = P(\boldsymbol{\theta},\boldsymbol{q}) - \overline{P} \leq 0, \quad (1)$$

where $\mathcal{L}(\mathcal{D},\boldsymbol{\theta},\boldsymbol{q})$ denotes the cross-entropy loss on the training data $\mathcal{D}$, which ensures high classification accuracy, and $c(\boldsymbol{\theta},\boldsymbol{q}) = P(\boldsymbol{\theta},\boldsymbol{q}) - \overline{P}$ represents the constraint function, ensuring that the estimated power consumption $P(\boldsymbol{\theta},\boldsymbol{q})$ remains within the strict task-specific upper limit $\overline{P}$. Evaluating the power contraint $c(\boldsymbol{\theta},\boldsymbol{q})$ requires accurate power estimation. Furthermore, since we will require derivatives through the contraint in the training procedure described later, we require a differentiable power model $P(\boldsymbol{\theta},\boldsymbol{q})$.

A. Surrogate power Modeling for Power Estimation

To simplify power analysis for resistor crossbar arrays, we adopt analytical solutions derived in [13]. However, modeling power consumption for nonlinear circuits, such as printed activation circuits (AFs), is more complex. We address this by approximating power consumption through SPICE simulations using the pPDK [29]. For each AF (p-ReLU, p-clipped ReLU, and p-sigmoid), we run 10,000 SPICE simulations, revealing distinct power behaviors as illustrated in Fig. 3 (c)-(f) (bottom). For instance, in p-clipped_ReLU, power spikes near a threshold as transistors conduct more current, then stabilizes due to the clipping effect; p-ReLU exhibits a smooth increase in power with input voltage, reflecting its unbounded nature; and p-sigmoid shows asymmetric power patterns due to higher current demands at negative voltages. Thus, the choice of AFs and their learnable parameters $q^{\mathrm{AF}} = [\mathbf{R},\mathbf{W},\mathbf{L}]$ (representing vectors of resistance, transistor widths, and transistor lengths) significantly impacts power consumption in pNCs.

To model learnable AFs, we define a feasible design space, $\mathbb{Q}^{\mathrm{AF}}$, with specified bounds for each parameter $\boldsymbol{q}^{\mathrm{AF}}$. We sample 10,000 circuit configurations using a Sobol sequence and simulate their power consumption using SPICE.

Based on this data, we train a 15-layer ANN for each AF as a surrogate model to map the corresponding physical variables $\boldsymbol{q}^{\mathrm{AF}}$ to power consumption, $\mathcal{P}^{\mathrm{AF}}(\boldsymbol{q}^{\mathrm{AF}})$. To enhance accuracy, data normalization and hyperparameter tuning are applied. This surrogate model provides efficient power estimation, enabling the total power calculation and analysis of the impact of AF configurations on power.

B. Power Estimation for a Printed Neuron

We estimate the power consumption of each printed neuron using the power of the crossbar ($\mathcal{P}^{C}$) and surrogate power models for its components: $\mathcal{P}^{\mathrm{AF}}$ and $\mathcal{P}^{\mathrm{N}}$, corresponding to the AF and negation circuits, respectively. To calculate the total power consumption, we determine N^{AF} and N^{N} [13], which represent the counts of AFs and negation circuits, respectively.

a) Counting Activation Circuits.: Initially, N^{AF} is calculated using an indicator function:

$$N^{\mathrm{AF}} = \mathbf{1}_N^{\top} \cdot \text{row max}\left\{\mathbb{1}_{\{|\boldsymbol{\theta}|>0\}}\right\}, \quad (2)$$

where $\mathbf{1}_N \in \mathbb{R}^N$ is a vector of all ones, $\mathbb{1}_{\{|\boldsymbol{\theta}|>0\}}$ assigns 1 to active circuits and 0 otherwise, and row max$(\cdot)$ returns the row-wise maximum values, as each row in $\boldsymbol{\theta}$ corresponds to a single activation circuit. If all elements in a row of surrogate conductances are zero, the corresponding activation circuit does not need to be printed or activated. This calculation ensures that any row with at least one non-zero conductance is counted only once. However, this piecewise constant function is non-differentiable, limiting its utility for gradient-based optimization.

b) Soft Count for Differentiability: For optimization, we replace $\mathbb{1}_{\{|\boldsymbol{\theta}|>0\}}$ in Eq. (2) with a sigmoid function $\sigma(|\boldsymbol{\theta}|)$ for gradient computation, defining $N^{\mathrm{AF}}_{\text{soft}}$ as

$$N^{\mathrm{AF}}_{\text{soft}} = \mathbf{1}_N^{\top} \cdot \text{row max}\left\{\sigma(|\boldsymbol{\theta}|)\right\}.$$

This relaxation allows gradient-based updates for N^{AF} by providing smooth gradients with respect to $\boldsymbol{\theta}$, as was previously done for N^{N}_{soft} in [13].

c) Final Power Estimation.: During the backward pass, the soft count facilitates gradient-based optimization. However, the final power estimation for a neuron is determined using the indicator function, as shown below:

$$\mathcal{P} = \mathcal{P}^{C} + N^{N}\cdot\mathcal{P}^{N} + N^{\mathrm{AF}}\cdot\mathcal{P}^{\mathrm{AF}}.$$

Power estimation in pNCs is influenced directly by the learnable parameters $\boldsymbol{q}^{\mathrm{N}}$, $\boldsymbol{q}^{\mathrm{AF}}$, and $\boldsymbol{\theta}$. Additionally, $\boldsymbol{\theta}$ indirectly affects the number of active elements in the circuit, which further impacts the total power consumption $P(\boldsymbol{\theta},\boldsymbol{q})$. In multi-neuron configurations, the output voltage of each neuron serves as the input for the subsequent neuron, with the final neuron's output representing the overall pNC output. The total power consumption is determined by summing the contributions from all neurons.

C. Power-constrained Training

To respect the constraints in training with Eq. (1), we require an adapation of the classic training procedure towards methods that can handle constraints. One such method that integrates particularly well backpropagation-based training schemes is the augmented Lagrangian method, see e.g. [30, 31]. It can be motivated by solving a sequence of unconstrained problems that converge to the solution of the constrained problem. Specifically, the method alternates between solving

$$\underset{\boldsymbol{\theta},\boldsymbol{q}}{\text{minimize}} \quad \max_{\lambda\geq 0}\; \mathcal{L}(\mathcal{D},\boldsymbol{\theta},\boldsymbol{q}) + \lambda\cdot c(\boldsymbol{\theta},\boldsymbol{q}) - \frac{1}{2\mu}(\lambda-\lambda')^2, \quad (3)$$

and updates to the Lagrange multiplier estimate λ' by

$$\lambda' \leftarrow \max\{0, \lambda' + \mu\cdot c(\boldsymbol{\theta},\boldsymbol{q})\}. \quad (4)$$

The parameter $\mu \in \mathbb{R}^{+}$ is a hyperparameter that controls the speed of convergence and influences the stability of the method.

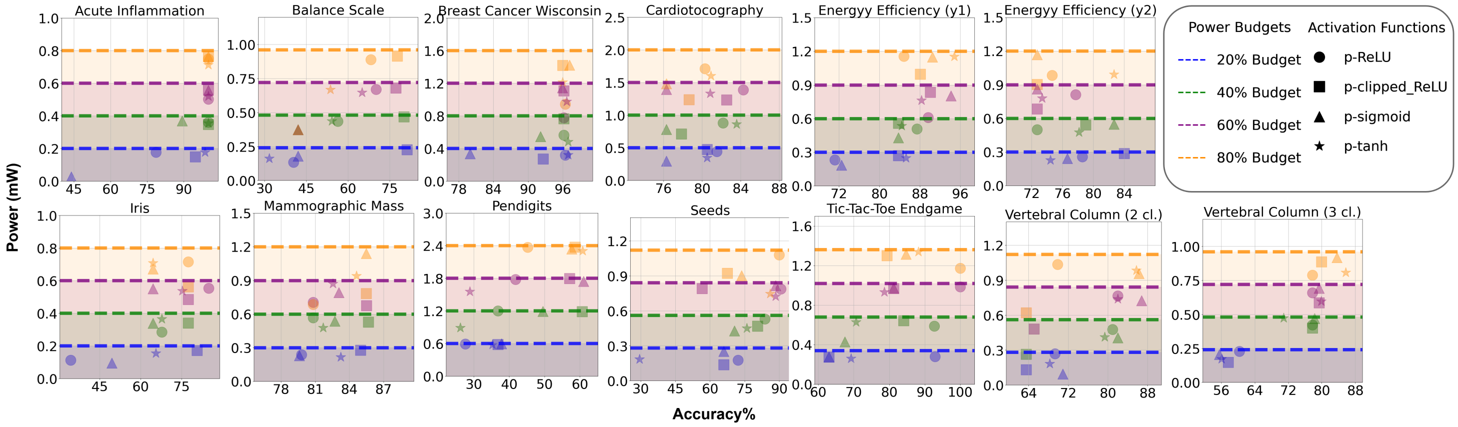

Fig. 4. Scatter plot showing the accuracy and power consumption for 13 datasets using four AFs (p-ReLU, p-clipped ReLU, p-sigmoid, p-tanh) under defined power budgets (20%, 40%, 60%, 80%). Each marker represents an AF, and dashed lines indicate the power budget thresholds. The goal is to achieve high accuracy while ensuring that power consumption stays below the specified budget constraints.

The minimization in Eq. (3) may be performed using classic backpropagation based training. The inner maximization over λ can be solved analytically for a given λ', details see [32]. After solving Eq. (3), the update of λ' in Eq. (4) adjusts the estimate for the Lagrange multiplier, encouraging the constraints to hold for the next minimization attempt. To save computation time, $\boldsymbol{\theta}$ and $\boldsymbol{q}$ should be warmstarted with the last solution obtained. The alternations of these steps can be stopped when a sufficiently good feasible solution is obtained.

IV. Evaluation

A. Experiment Setup

1) Dataset and Training Setup: To evaluate the proposed power-constrained training approach, we implemented the proposed method[2] using PyTorch [33] and tested it on 13 benchmark datasets used in prior studies [13, 34, 35]. The datasets were split into training (60%), validation (20%), and test (20%) sets to ensure consistent evaluation. Training used full-batch gradient descent with the Adam optimizer [36], starting with an initial learning rate of 0.1. Early stopping was applied to prevent overfitting by halving the learning rate after 100 epochs without improvement on the validation set and stopping training if the power constraint was violated. For the selection of μ, RayTune [37] was used. All pNCs were trained with a consistent topology ($\#inputs$-3-$\#outputs$) and randomly initialized parameters for each AF.

Following the primary training phase, a fine-tuning step was conducted to enhance accuracy while strictly adhering to power constraints. During this process, masks m^C were generated to deactivate inactive components, such as resistors with zero conductance and unused AFs. Similarly, masks m^N were introduced for negation circuits to enforce positive weights where necessary. These masks effectively pruned redundant hardware components, reducing power consumption. The model was then retrained using cross-entropy loss, optimizing accuracy without violating the power constraints. This mask-based adjustment ensured a trade-off between accuracy and power, even under stringent budget limitations.

2) Benchmark Circuit Design Setup: All experiments at the simulation level were conducted using pPDK [29], while the functionality of printed neuromorphic hardware has been validated in previous work [14, 38].

3) Algorithm Benchmarking: To benchmark the proposed method, we compared its results to a Pareto front generated by traditional penalty-based optimization with p-tanh AFs [13]. For the baseline, we uniformly sampled 50 μ values in the range $[0, 1]$ and trained each model 10 times (using 10 different seeds) to generate the Pareto front. For the augmented Lagrangian approach, power constraints were set at 20%, 40%, 60%, and 80% of the maximum power observed during unconstrained training. This setup allowed for a comprehensive evaluation of the trade-offs between power and classification accuracy.

B. Result

To evaluate the effectiveness of our power-constrained training approach, we compared the results with a penalty-based method [13]. Four neural networks, each using a different AF (p-ReLU, p-Clipped_ReLU, p-sigmoid, and p-anh), were trained under four power budgets: 20%, 40%, 60%, and 80% of the maximum power consumption derived from unconstrained power training. The top three models per dataset were selected based on test accuracy. As shown in Fig. 4, each point represents classification accuracy and power consumption for a dataset. Marker shapes indicate AFs (circles for p-ReLU, squares for p-Clipped_ReLU, triangles for p-sigmoid, and stars for p-tanh), and marker colors represent power budgets, aligned with horizontal dashed lines for power thresholds. The x-axis shows accuracy (percentage), and the y-axis shows power (milliwatts).

Tab. I summarizes the average performance metrics across 13 datasets, comparing power consumption, accuracy, and device count among the proposed AFs (p-ReLU, p-Clipped_ReLU, p-sigmoid, and p-tanh). On the right side of the table, the results for the penalty-based baseline method [13] are presented, showing accuracy and power under different scaling factors α (0.25, 0.5, 0.75, and 1), which indicate the trade-off between power and accuracy. Additionally, for the baseline method, we generated a Pareto front (pink curve) using 500 penalty-based runs, varying the scaling factor α over the range $[0, 1]$. Fig. 5

[2] Code is available at https://github.com/KIT-Neuromorphic-Computing/Power-Constrained-PNC.

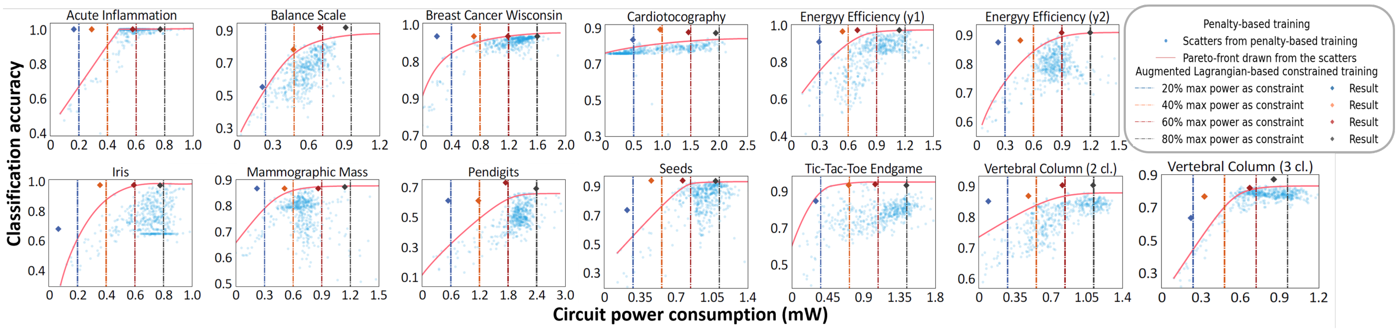

Fig. 5. Classification accuracy and power consumption of pNCs using exemplary p-tanh AF from penalty-based and Augmented Lagrangian-based training approaches. The blue scatters are the results of penalty-based training, while the pink curves are Pareto fronts drawn from the scatters. The vertical lines indicate the power constraints in the augmented Lagrangian approach, whereas the rhombus with the same color refers to the results from the augmented Lagrangian

compares the optimal solutions found by our method in a single training run against this Pareto front. Vertical dashed lines highlight power constraints, demonstrating that our method efficiently achieves power-constrained optima, often matching or surpassing the Pareto front with significantly fewer training runs.

TABLE I
AVERAGED PERFORMANCE METRICS ACROSS 13 DATASETS: COMPARISON OF METRICS (POW: POWER (mW), ACC: ACCURACY, DEV: DEVICE COUNT) ACROSS AFS AT DIFFERENT POWER BUDGETS AND PENALTY-BASED BASELINE

ACT / Metric		p-ReLU	p-clipped_ ReLU	p-sigmoid	p-tanh	Baseline	
20%	Pow	0.27	0.26	0.23	0.25	10.8	$\alpha = 1$
	Acc	67.94	**74.56**	62.60	67.29	54.9	
	#Dev	**17**	27	27	19	-	
40%	Pow	0.56	0.56	0.53	0.53	44.1	$\alpha = 0.75$
	Acc	78.11	**80.51**	73.96	74.60	83.6	
	#Dev	**28**	36	41	30	-	
60%	Pow	0.85	0.84	0.88	0.84	55.6	$\alpha = 0.5$
	Acc	**82.76**	78.54	78.89	78.45	86.3	
	#Dev	**34**	47	54	40	-	
80%	Pow	1.10	1.05	1.12	1.09	68.8	$\alpha = 0.25$
	Acc	80.42	78.84	78.05	**81.82**	87.5	
	#Dev	**37**	45	57	58	-	

C. Discussion

Our experimental results demonstrate the effectiveness of the augmented Lagrangian approach in achieving high accuracy while adhering to defined power constraints across various datasets. As shown in Fig. 4, all results lie below the defined power levels, confirming the method's capability to enforce power limits effectively. Notably, at 20% of the power budget, accuracy decreases compared to higher power budgets, highlighting the trade-off between power and performance. As shown in Tab. I, the baseline demonstrates high accuracy at $\alpha = 0.25$ but with higher power consumption, while $\alpha = 1$ reduces power at the cost of accuracy. In contrast, the proposed method achieves a balanced trade-off between power and accuracy across power budgets in a single training run. Furthermore, it delivers results comparable to or better than the Pareto front obtained from penalty-based objectives (Figure 5) without the need for extensive hyperparameter tuning.

The success of this approach is due to two main factors. First, by directly incorporating power constraints into the training objective, the augmented Lagrangian method avoids the instability often caused by high power penalties in penalty-based methods. Second, this explicit formulation focuses on achieving the best feasible accuracy for each specified power budget without unnecessarily minimizing power further.

Compared to penalty-based methods, which often create ill-conditioned optimization problems and fail to align with optimal power-accuracy trade-offs, the augmented Lagrangian method dynamically enforces constraints while maintaining accuracy. This ensures a balanced solution on the Pareto front with fewer computational resources.

Analyzing individual AFs reveals trade-offs between accuracy and device count. For example, p-tanh achieves up to 81.82% accuracy at an 80% power budget, but requires an average of 58 devices, whereas p-ReLU achieves 80.42% accuracy with only 37 devices—a 36% reduction. This makes p-ReLU particularly suitable for hardware-limited applications. These findings demonstrate that such trade-offs persist even under strict power constraints, with reduced device counts contributing to lower power consumption.

V. CONCLUSION

In this work, our objective is to introduce a methodology for power-constraint training of analog printed neuromorphic circuits. This is implemented by an augmented Lagrangian approach which leveraged the surrogate power models for activation function circuit primitives. The approach achieved en par to superior performance to penalty based methods across 13 benchmark datasets, while requiring fewer training time to find suitable pNCs. We additionally developed power models for four printed AFs, enabling analysis of power and accuracy trade-offs. While p-tanh achieves higher accuracy, p-ReLU minimizes device count, proving beneficial for resource-limited applications. Due to the flexibility of the proposed approach, future works may explore its applicability to additional circuit components and constraints.

ACKNOWLEDGMENT

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

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] H. Zhao *et al.*, "Towards Temporal Information Processing–Printed Neuromorphic Circuits with Learnable Filters," in *Proceedings of the 18th ACM International Symposium on Nanoscale Architectures*, 2023, pp. 1–6.

[4] E. Shirzaei Sani *et al.*, "A Stretchable Wireless Wearable Bioelectronic System for Multiplexed Monitoring and Combination Treatment of Infected Chronic Wounds," *Science Advances*, vol. 9, no. 12, p. 7388, 2023.

[5] W. S. Wong and A. Salleo, *Flexible electronics: materials and applications*. Springer Science & Business Media, 2009, vol. 11.

[6] W. Gao, H. Ota, D. Kiriya, K. Takei, and A. Javey, "Flexible electronics toward wearable sensing," *Accounts of chemical research*, vol. 52, no. 3, pp. 523–533, 2019.

[7] H. Zhao, T. Röddiger, and M. Beigl, "Aircase: Earable Charging Case with Air Quality Monitoring and Soundscape Sonification," in *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*, 2021, pp. 180–184.

[8] Sigenics, "Sigenics - custom integrated circuits," https://sigenics.com/, accessed: Nov. 19, 2024.

[9] I. Ayyub, "Top 10 cheapest programmable microcontrollers," 2018, accessed: Nov. 19, 2024. [Online]. Available: https://pic-microcontroller.com/world-top-10-cheapest-microcontrollers-mcus/

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

[11] 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.

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

[13] H. Zhao *et al.*, "Power-Aware Training for Energy-Efficient Printed Neuromorphic Circuits," in *42nd IEEE/ACM International Conference on Computer-Aided Design*, 2023.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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.

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

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

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

[24] 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.

[25] 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.

[26] 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.

[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] 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.

[30] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.

[31] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[32] J. K. H. Franke, M. Hefenbrock, G. Koehler, and F. Hutter, "Improving deep learning optimization through constrained parameter regularization," 2024. [Online]. Available: https://arxiv.org/abs/2311.09058

[33] 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.

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

[35] ——, "Highly-Bespoke Robust Printed Neuromorphic Circuits," in *Design, Automation and Test in Europe (DATE)*. IEEE, 2023.

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

[37] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. Gonzalez, and I. Stoica, "Tune: A research platform for distributed model selection and training," 2018, accessed: 2024-07-11. [Online]. Available: https://docs.ray.io/en/latest/tune.html

[38] 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.