Neural Architecture Search for Highly Bespoke Robust Printed Neuromorphic Circuits

Priyanjana Pal*
priyanjana.pal@kit.edu
Karlsruhe Institute of Technology

Haibin Zhao*
haibin.zhao@kit.edu
Karlsruhe Institute of Technology

Tara Gheshlaghi
tara.gheshlaghi@kit.edu
Karlsruhe Institute of Technology

Michael Hefenbrock
michael.hefenbrock@revoai.de
RevoAI GmbH

Michael Beigl
michael.beigl@kit.edu
Karlsruhe Institute of Technology

Mehdi B. Tahoori
mehdi.tahoori@kit.edu
Karlsruhe Institute of Technology

ABSTRACT

The market demand for next-generation flexible electronics is experiencing a significant upsurge, particularly in cost-sensitive consumer applications like smart packaging and smart bandages. These products are beyond the reach of traditional silicon-based electronics due to their high production cost and rigid form factor. Printed electronics (PE), with its adaptable and ultra-low-cost solutions, essentially meet the unique needs of these emerging application areas. This work presents a novel approach using an evolutionary algorithm (EA) to design highly bespoke printed analog neuromorphic circuits (pNCs) offering robustness against variability inherent in the printing process. By leveraging this algorithm and designing robust activation circuits, not only the resistances (weights) in the crossbar and parameters in the activation circuits, but also the types of nonlinear circuits (i.e., functional forms of activation functions) as well as the circuit topologies (neural architecture) can be learned to enhance the circuit robustness against printing variations. Experiments on 13 benchmark datasets demonstrate that, compared to the baseline, the proposed methodology can further outperform the normalized classification error rate by $\approx$ 55.38% and $\approx$ 25.11% under high-precision ($\pm$5%) and low-precision ($\pm$10%) printing scenarios, respectively. Moreover, the algorithm suggests the ReLU as the most robust activation function (AF) circuit family with only $\approx$ 21% susceptible to low precision ($\pm$10%) printing variation.

CCS CONCEPTS

• **Hardware** → **Emerging Technologies**; **Robustness**; **Flexible and printable circuits**; • **Computing methodologies** → **Machine Learning approaches**; **Genetic algorithms**; *Bio-inspired approaches*;

KEYWORDS

printed electronics, neuromorphic computing, evolutionary algorithm, printed neuromorphic circuit, artificial neural network

*Authors contributed equally to this work.

ICCAD '24, October 27–31, 2024, New York, NY, USA

ACM ISBN 979-8-4007-1077-3/24/10.
https://doi.org/10.1145/3676536.3676812

ACM Reference Format:
Priyanjana Pal, Haibin Zhao, Tara Gheshlaghi, Michael Hefenbrock, Michael Beigl, and Mehdi B. Tahoori. 2024. Neural Architecture Search for Highly Bespoke Robust Printed Neuromorphic Circuits. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '24), October 27–31, 2024, New York, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3676536.3676812

1 INTRODUCTION

Despite the continual advancement in terms of power efficiency and transistor density, silicon-based electronics can hardly be introduced into many consumer edge applications, which includes smart packaging [1], smart bandages [2, 3], wearables or other disposable electronics [4–7] for consumer products. In such edge scenarios, devices are required to be flexible, customized, bio-compatible, and on-demand fabricated, with manufacturing costs expected to be under a few cents. However, these properties pose substantial challenge to silicon-based VLSI due to limitations such as bulky substrates and highly complicated lithography-based processing.

In this regard, printed electronics (PE) stands out as an adaptable and cost-effective alternative for such applications. The technology's standout characteristic lies in its ability to provide bespoke application-specific customization requirements, regardless of high or low volumes, owing to its low-cost nature of the additive printing process over conventional lithography-based fabrication.

The main benefit of additive printing process is the significant cost reduction achieved through maskless manufacturing. However, there are also intrinsic drawbacks in this printing methods: Due to the limited operational precision of the printing equipments and the dispersion of ink droplets [8], the geometric (thus electronic) features of the printed components inevitably deviate from their designed values, leading to greater variability in the printing process compared to the high-resolution lithography-based methods. Meanwhile, bespoke architectures used in the realization of printed neuromorphic circuits, along with the use of analog computing [9] or digital approximation [10], can drastically affect the vulnerability of these printed circuits to variation. While the variation tolerance of generic (model-agnostic) neural network (NN) hardware accelerators and other machine learning models has been extensively investigated [9, 11, 12], this aspect remains highly unexplored for custom hardwired (bespoke) activation functions (AFs) in printed analog neuromorphic circuits (pNCs) commonly used in PE.

Thus, many efforts, from manufacturing technology to algorithmic circuit design, have been devoted to mitigating this issue. In this

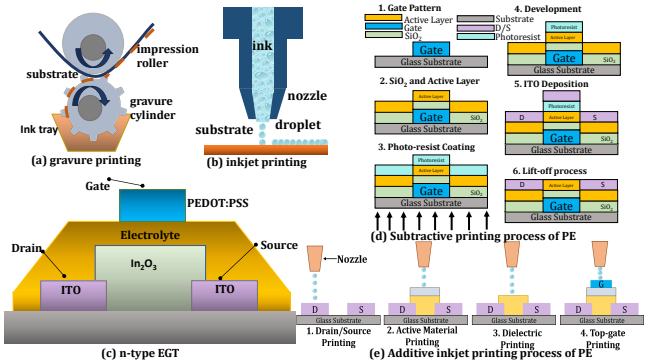

Figure 1: Schematic of (a) gravure printing (b) inkjet printing; (c) front view of a printed N-type EGT; (d) Subtractive and (e) Additive printing process of PE

context, our work have adopted a variation-aware evolutionary algorithm (EA), which utilizes various learnable AF circuits to achieve a variation-robust pNC. In short, the contributions are:

(1) This work proposes the design of **printed bespoke learnable AF** circuits at both circuit and algorithmic level.
(2) This research, for the first time, introduces an innovative EA approach to optimize pNCs, capable of not only tuning the AF circuits' component values but also selecting the most appropriate type of AF for each neuron per dataset to mitigate printing error while performing Neural Architecture Search (NAS) and training crossbar conductances.

In experiment based on SPICE simulation, the proposed method demonstrated an improvement in the normalized classification error rate by 55.38% and 25.11% under ±5% and ±10% variation respectively compared to the baseline. Additionally, it achieved ≈ 21% higher robustness when ReLU family is selected as the AF.

The rest of this paper is structured as follows: Sec. 2 introduces PE, pNC, and other preliminaries. Sec. 3 describes the design of learnable AF circuits and formulation of variation-aware EA. In Sec. 4, the proposed approach is evaluated with extensive simulations. Finally, Sec. 5 summarizes this work.

2 PRELIMINARIES

2.1 Printed Electronics

PE represents a rapidly evolving electronic fabrication technology poised to revolutionize domains like wearables, smart sensors, and the Internet of Things (IoT) [4–6]. Different from traditional silicon-based VLSI, PE focuses instead on lowering the production cost through maskless additive manufacturing at low processing temperatures. Due to abundant functional materials, such electronics boast unique advantages including mechanical flexibility, porosity, non-toxicity, and biodegradability outperforming conventional lithography-based silicon electronics.

Optimal device performance in PE is typically attained with vacuum-deposited, highly purified molecular substrates [13]. However, solution-based fabrication techniques, like spin-coating and inkjet printing, have gained enough attention for their potential to optimize manufacturing efficiency and reduce costs [14]. Printing technologies (as shown in Fig. 1) are generally divided into two main categories: (i) replication printing, exemplified by gravure printing (Fig. 1 (a)), which is optimized for high-throughput manufacturing, and (ii) jet printing, with inkjet printing (Fig. 1 (b)) serving as a key paradigm, for the bespoke fabrication of electronic circuits in smaller quantities. PE utilizes either additive or subtractive manufacturing methods. As illustrated in Fig. 1 (d)), subtractive processes involve alternating deposition and etching, and are typically costlier due to specialized equipment requirements. Additive processes, in contrast (Fig. 1 (e)), sequentially deposit materials to form electronic components, as seen in the inkjet printing process. Despite the lower resolution and increased variability of additive methods, they offer substantial cost advantages.

Current research in inorganic PE is focused on the development of N-type Electrolyte-Gated Transistors (n-EGTs) (as shown in Fig. 1 (c)), where the band structure of metal oxides enables significantly higher electron mobility, allowing these n-EGTs to function efficiently at sub-1V supply voltages [15].

However, additive manufacturing also introduces challenges to PE, such as low device counts, large device dimensions, and high variability. Hence, PE does not target to compete with silicon-based devices in terms of performance within VLSI applications, but rather to complement them in resource-limited and cost-sensitive edge computing areas, like disposable electronics or wearable devices. In these scenarios, computational tasks are typically simple and have a relatively high tolerance to compute imprecision and thus require only small-scale circuits. Nonetheless, the high printing variation caused by ink dispersion [8] and printer inaccuracy pose a substantial influence on printed computing systems.

2.2 Printed Analog Neuromorphic Circuits

With the progression of artificial intelligence, neuromorphic computing has emerged as an effective approach for solving complex and nonlinear tasks. Small NNs, as enabled by pNCs, are sufficiently powerful enough for real-time and sensor data processing, without requiring vast computational power and resources associated with larger NN architectures commonly used in deep learning. This compatibility is particularly beneficial for various PE target applications, such as wearable sensors, flexible displays, and smart bandages [4–6], having specific resource constraints and functional requirements. Their capacity for on-demand fabrication facilitates the embedding of pNCs directly onto flexible substrates—something unattainable with conventional silicon-based hardware, thus making them an ideal candidate for applications requiring low device counts and bespoke customized configurations.

Notably, this computational capacity generally requires only basic operations such as weighted-sum and nonlinear activation. Therefore, this streamlined but efficient computing paradigm becomes attractive for circuit design, as desired functionalities can be achieved by the interconnection of simple circuit primitives. In the context of pNCs, by printing appropriate resistor crossbars (for weighted-sum) and nonlinear transformation circuits (for AFs), pNCs can achieve bespoke design for the computational requirements of the target applications of PE.

2.2.1 Hardware Primitives. Fig. 2 exemplifies the circuit schematics of a neuron in printed analog neuromorphic circuits. Fig. 2(a) shows a neuromorphic circuit resembling an artificial neural network (ANN) with a 6-4-3 topology. Fig. 2(b) show the schematic of a

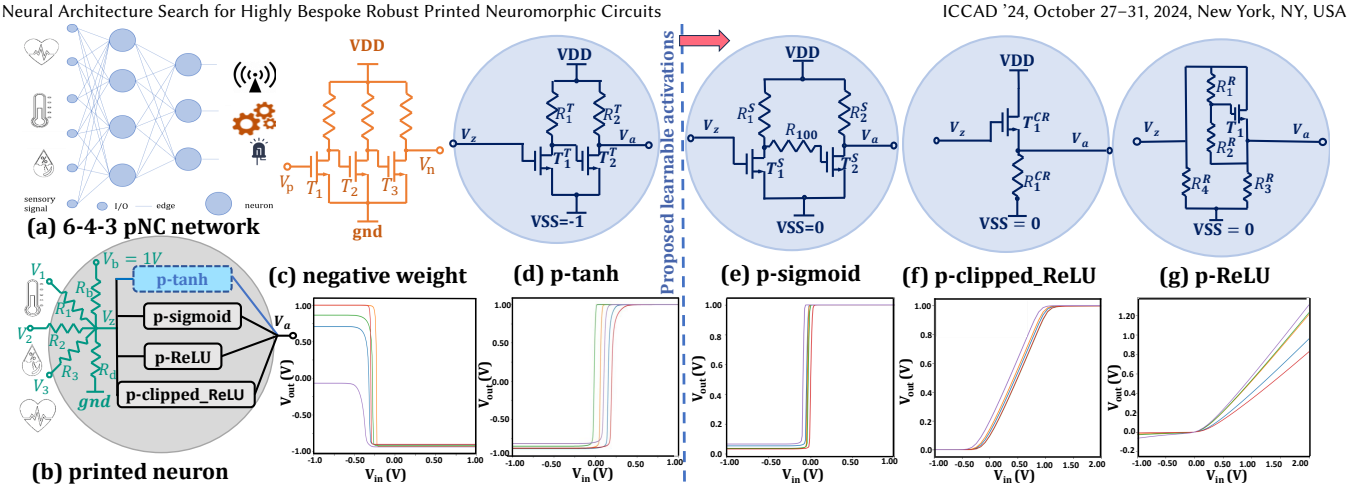

Figure 2: Overview of (a) 6-4-3 printed neuromorphic network. (b) printed neuron ;(c) negative weight circuit, activation circuit design and transfer characteristics curve of (d) p-tanh (e) p-sigmoid (f) p-clipped_ReLU and (g) p-ReLU.

printed neuron. Some negative weight circuits (Fig. 2 (c)) are also incorporated in case required.

Resistor Crossbars. The left part of Fig. 2(b) shows the circuit schematics of a resistor crossbar in printed neuron. Following Ohm's Law and Kirchhoff's Laws, the resulting output voltage of the crossbar V_z can be calculated as

$$V_z = \frac{g_1}{G}V_1 + \frac{g_2}{G}V_2 + \frac{g_3}{G}V_3 + \frac{g_b}{G}V_b, \tag{1}$$

where g_i signifies the conductances of the resistors R_i and G represents the aggregate conductance $\sum_j g_j + g_b + g_d$. Evidently, the output voltage V_z is the weighted-sum of the input voltages V_i, with the weights embodied by the ratio of conductance values between g_i, g_b and G. So, analogous to training ANNs, by designing and printing suitable conductances, desired weights can be achieved.

Printed negative weight circuit. As the conductances in the crossbar resistor array are limited to representing only positive weights, certain resistors, as shown in Fig. 2(b), are augmented with inverter-based circuits to introduce negative weight capabilities, as shown by the detailed circuit schematics in Fig. 2(c). This facilitates the emulation of multiplication operation with negative weights through the inversion of input voltages. The transfer characteristic of these negative weight circuits is characterized by a modified negative tanh function.

$$\mathrm{neg}(V_z) = -\left(\eta_1^{\mathrm{N}} + \eta_2^{\mathrm{N}} \cdot \tanh\left(\left(V_z - \eta_3^{\mathrm{N}}\right) \cdot \eta_4^{\mathrm{N}}\right)\right), \tag{2}$$

where $\boldsymbol{\eta}^{\mathrm{N}} = [\eta_1^{\mathrm{N}}, \eta_2^{\mathrm{N}}, \eta_3^{\mathrm{N}}, \eta_4^{\mathrm{N}}]$ are auxiliary parameters that modify the original tanh function, which is ultimately determined by the physical quantities $\boldsymbol{q}^{\mathrm{N}} = [R_1^{\mathrm{N}}, R_2^{\mathrm{N}}, R_3^{\mathrm{N}}, W_1^{\mathrm{N}}, L_1^{\mathrm{N}}, W_2^{\mathrm{N}}, L_2^{\mathrm{N}}, W_3^{\mathrm{N}}, L_3^{\mathrm{N}}]$ in the circuit. Here, W_i^{N} and L_i^{N} are geometric features (width and length) of the transistor T_i^{N}. Here, the superscript $(\cdot)^{\mathrm{N}}$ denotes the variables in the Negative weight circuits. Moreover, by optimizing $\boldsymbol{q}^{\mathrm{N}}$, the shape of negative weight function can be tuned to better fit specific target tasks. A surrogate circuit model for training $\boldsymbol{q}^{\mathrm{N}}$ was introduced in [9].

Printed activation circuits. Following the crossbar, the signals are passed through a printed activation circuit to resemble the AFs in ANNs. The circuit diagram, in Fig. 2(c), is taken from [9]. Analogous to the negative weight circuit, the characteristic curve of the printed tanh (p-tanh) activation circuit can be represented by a parameterized tanh function, specifically,

$$V_a = \mathrm{ptanh}(V) = \eta_1^{\mathrm{T}} + \eta_2^{\mathrm{T}} \cdot \tanh\left(\left(V - \eta_3^{\mathrm{T}}\right) \cdot \eta_4^{\mathrm{T}}\right) \tag{3}$$

with the auxiliary parameters $\boldsymbol{\eta}^{\mathrm{T}} = [\eta_1^{\mathrm{T}}, \eta_2^{\mathrm{T}}, \eta_3^{\mathrm{T}}, \eta_4^{\mathrm{T}}]$ determined by $\boldsymbol{q}^{\mathrm{T}} = [R_1^{\mathrm{T}}, R_2^{\mathrm{T}}, W_1^{\mathrm{T}}, L_1^{\mathrm{T}}, W_2^{\mathrm{T}}, L_2^{\mathrm{T}}]$. Similarly, $\boldsymbol{q}^{\mathrm{T}}$ can also be trained to fit specific target tasks. Also, the superscript $(\cdot)^{\mathrm{T}}$ denotes the variables in the tanh circuits.

In addition to this circuit, we designed three additional activation circuits mimicking other AFs, i.e., a printed sigmoid (p-sigmoid) circuit, a printed clipped ReLU (p-clipped ReLU) circuit, and a printed ReLU (p-ReLU) circuit. With these novel designs, we explore their abilities in improving the circuit robustness with respect to printing variations. Detailed descriptions are introduced in Sec. 3.

2.3 Algorithmic Level Design and Optimization

The additive manufacturing process enables the capability of PE to engineer components with predetermined values through the specification of their geometric attributes. Thus, by printing customized conductance values in the resistor crossbar array, which correspond to the requisite synaptic weights in ANNs, pNCs are allowed to have bespoke design for target computational functionalities. Exploiting this capability of pNCs necessitates precise modeling of the circuit architecture and hardware-aware training process.

Modeling of Printed Neuromorphic Circuit. Based on printed neurons described in Sec. 2.2, the mathematical model of a printed neuron, taking p-tanh as the example for activation circuit, is given by

$$\mathrm{ptanh}_{\boldsymbol{q}^{\mathrm{T}}}\left(\sum_i w_i \left(V_i \cdot \mathbb{1}_{\{\theta_i \geq 0\}} + \mathrm{neg}_{\boldsymbol{q}^{\mathrm{N}}}(V_i) \cdot \mathbb{1}_{\{\theta_i < 0\}}\right)\right), \tag{4}$$

where θ_i is the learnable parameter encoding conductance by $g_i = |\theta_i|$ and the presence of a *negation circuit* via its sign. Moreover, the indicator function $\mathbb{1}_{\{\cdot\}}$ returns 1 if its condition is true, else 0. Additionally, the weights w_i are given by

$$w_i = \frac{|\theta_i|}{\sum_j |\theta_j|}.$$

Note that, $\boldsymbol{q}^{\mathrm{T}}$ and $\boldsymbol{q}^{\mathrm{N}}$ are also factors that can influence the output of the printed neuron. To facilitate their training, a NN-based surrogate model is employed, transforming the activation

circuit parameters to their corresponding $\boldsymbol{\eta}$. For training, these surrogate models are used to approximate the complex mappings of the learnable circuits [9].

Gradient-Based Optimization. Gradient-based learning with back-propagation [16] forms the backbone of training modern ANNs. However, straightforward gradient-based optimization is unable to handle problems that encompass differentiable operations. Additionally, some functions do not provide useful update information through their gradient, an example being piece-wise constant functions. Another example that relates to this work is that the type of AF in Eq. (4) cannot be selected via gradient-based methods, as it amounts to a discrete decision. While the topic of *NAS* has received increasing attention, existing methods [17, 18] may be beneficial for optimizing circuit topologies but have not been adapted for pNCs yet.

Evolutionary-Based Optimization. EAs draw inspiration from natural selection and biological evolution. These algorithms utilize operations such as crossover and mutation to evolve candidate solutions over time, thereby optimizing them in an incremental fashion. One of the key benefits of EAs is their high degree of versatility and adaptability; for instance, they do not require the problem at hand to be differentiable. This characteristic makes EAs particularly suitable for a wide range of problems that are otherwise challenging for traditional optimization methods.

Although it is generally known that EAs may not match the efficiency of gradient-based optimization methods, particularly in the context of large-scale problems, this limitation is less significant when it comes to the design and optimization of small-scale NNs. Such NNs are often found in many computing applications where the flexibility and the computational resources are limited. Fortunately, these applications are exactly the target domains of the PE. Moreover, as EAs can be parallelized efficiently, their computational time can be substantially reduced. In sum, despite their perceived inefficiency in large-scale scenarios, the practical use of EAs in optimizing small-scale PE for target applications devices underscores their relevance.

2.4 Related Work

Despite the unique advancements of the printing technology and PE, the nature of additive manufacturing process introduces multiple sources of variation in printed components, such as ink dispersion on substrates, irregularities in droplet jetting, satellite droplets, and missing droplets [8, 19]. These variations are sometimes modeled by a uniform distribution of the electrical characteristics of printed devices with in the printed resolution, specifically the parameters of the electrolyte-gated transistors (n-EGTs) and is addressed by a Gaussian Mixture Model at the device level [20, 21].

Efforts from multiple aspects has been paid to mitigate the impact of printing variations, including quality control engineering [22] and material science [23]. From the algorithmic level, advanced design strategies that incorporate variation-aware training [9, 24] are employed. These strategies involve designing circuits with robustness [25–27] in mind, ensuring that they can tolerate a certain degree of variation without any performance degradation.

In this work, we focus on device variability in PE and analog computing by using an EA. In short, our research aims to bridge this gap of variation-aware training by developing not only bespoke pNCs, but also more resilient bespoke AF circuits that maintain high robustness even with the inherent additive manufacturing printing variation.

3 DESIGN AND METHODOLOGY

In the following, we will provide a detailed description of learnable AFs and their transfer characteristic curves based on our circuit design in PE. Fig. 2 (d, e, f) depicts the schematic of different learnable activation circuits, i.e., printed sigmoid (p-sigmoid), printed clipped ReLU (p-clipped_ReLU) and printed ReLU (p-ReLU) design.

3.1 Learnable Activation Function (AF) Circuits

Printed Sigmoid Circuit. Similar to the learnable p-tanh AF circuit design in Fig. 2 (c), the p-sigmoid AF can also be obtained from this configuration by replacing the supply voltage VSS = 0 and adding a small 100 Ω resistance as shown in Fig. 2 (d) and can be modeled by a suitable mathematical equation with its own distinct parameters.

Table 1: FEASIBLE DESIGN SPACE OF P-SIGMOID CIRCUIT

Range	R_1^{S} (kΩ)	R_2^{S} (kΩ)	W_1^{S} (µm)	L_1^{S} (µm)	W_2^{S} (µm)	L_2^{S} (µm)
minimal	350	40	80	80	500	40
maximal	750	80	600	200	800	80

The equation for a printed sigmoid function is given by

$$V_{\mathrm{a}} = \eta_1^{\mathrm{S}} + \eta_2^{\mathrm{S}} \cdot \mathrm{sigmoid}\left(\left(V_{\mathrm{z}} - \eta_3^{\mathrm{S}}\right) \cdot \eta_4^{\mathrm{S}}\right), \tag{5}$$

where sigmoid$(\cdot)$ function is defined by

$$\mathrm{sigmoid}(x) = \frac{1}{1+\mathrm{e}^{-x}}.$$

where $\boldsymbol{\eta}^{\mathrm{S}} = [\eta_1^{\mathrm{S}}, \eta_2^{\mathrm{S}}, \eta_3^{\mathrm{S}}, \eta_4^{\mathrm{S}}]$ are auxiliary parameters determined by the physical quantities $\boldsymbol{q}^{\mathrm{S}} = [R_1^{\mathrm{S}}, R_2^{\mathrm{S}}, W_1^{\mathrm{S}}, L_1^{\mathrm{S}}, W_2^{\mathrm{S}}, L_2^{\mathrm{S}}]$ in the circuit. Adjusting these parameters allows to adapt the shape of the AF within the printed circuit.

Printed Clipped ReLU Circuit. A clipped ReLU activation circuit uses only one transistor and one resistor in series as shown in Fig. 2 (e) and extends the basic ReLU by imposing an upper bound to the activation. The output voltage, V_{a}, is constrained between 0 and a predefined maximum value V_{max}.

Table 2: FEASIBLE DESIGN SPACE OF P-CLIPPED RELU CIRCUIT

Range	R_1^{CR} (MΩ)	W_1^{CR} (µm)	L_1^{CR} (µm)
minimal	1	40	80
maximal	10	100	200

The mathematical representation of the clipped ReLU is:

$$V_{\mathrm{a}} = \begin{cases} \eta_1^{\mathrm{CR}}, & V_{\mathrm{z}} < \eta_3^{\mathrm{CR}} \\ \eta_2^{\mathrm{CR}}, & V_{\mathrm{z}} > \eta_4^{\mathrm{CR}} \\ \dfrac{\eta_2^{\mathrm{CR}} - \eta_1^{\mathrm{CR}}}{\eta_4^{\mathrm{CR}} - \eta_3^{\mathrm{CR}}} V_{\mathrm{z}} + \dfrac{\eta_1^{\mathrm{CR}}\eta_4^{\mathrm{CR}} - \eta_2^{\mathrm{CR}}\eta_3^{\mathrm{CR}}}{\eta_4^{\mathrm{CR}} - \eta_3^{\mathrm{CR}}}, & \text{otherwise,} \end{cases} \tag{6}$$

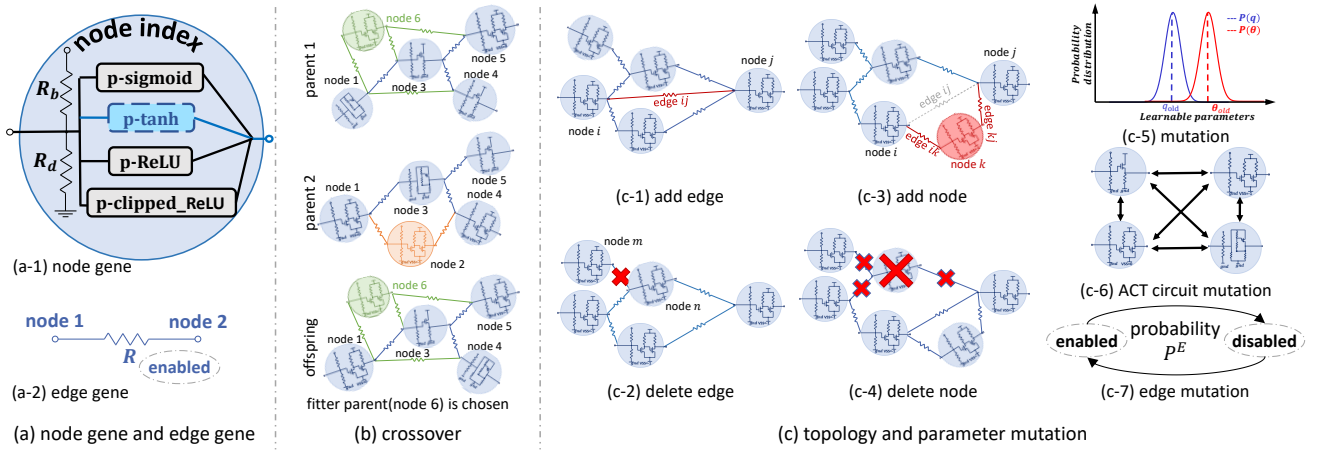

Figure 3: Overview of the EA-based training of printed neuromorphic circuits (pNCs). (a) Genes that encode nodes and edges. (b) Crossover from the parent genomes to offsprings. (c) Mutation of the topology and learnable parameters.

where $\boldsymbol{\eta}^{\mathrm{CR}} = [\eta_1^{\mathrm{CR}}, \eta_2^{\mathrm{CR}}, \eta_3^{\mathrm{CR}}, \eta_4^{\mathrm{CR}}]$ are auxiliary parameters determined by the physical quantities $\boldsymbol{q}^{\mathrm{CR}} = [R_1^{\mathrm{CR}}, W_1^{\mathrm{CR}}, L_1^{\mathrm{CR}}]$.

The printed clipped ReLU circuit thus models a modified ReLU function commonly used in ANNs to prevent overactivation by limiting the maximum output value. The parameters can be tuned to fit specific requirements, providing flexibility in shaping the activation characteristics of the ANN. The ability to clip the activation at a certain level also helps with issues like exploding gradients during the training process of ANNs.

Printed ReLU Circuit. The printed ReLU activation circuit as shown in Fig. 2 (f), uses four resistors and one transistor and implements a piecewise linear function that maintains the linearity for positive inputs while nullifying negative inputs. As the transfer characteristic curve has a slope in the negative half and has a smooth transition at $V_z = 0$, neither the ideal ReLU function, nor its variation, e.g., LeakyReLU [28] function nor thesoftplus [29] function, is sufficient to precisely describe the printed ReLU circuit. Thus, we combine a softplus function to provide the smoothness at $V_z = 0$ and a constant linear function to provide the slope at negative half. Consequently, the function that describes printed

Table 3: FEASIBLE DESIGN SPACE OF P-RELU CIRCUIT

Range	R_1^{R} (kΩ)	R_2^{R} (kΩ)	R_3^{R} (kΩ)	R_4^{R} (kΩ)	W_1^{R} (μm)	L_1^{R} (μm)
minimal	10	500	1	30	200	80
maximal	100	2000	20	100	800	120

ReLU circuit is : designed as

$$V_{\mathrm{a}} = \eta_1^{\mathrm{R}} \cdot (x - \eta_3^{\mathrm{R}}) + \eta_2^{\mathrm{R}} \cdot \mathrm{softplus}(V_{\mathrm{z}} - \eta_3^{\mathrm{R}}, \eta_5^{\mathrm{R}}) + \eta_4^{\mathrm{R}}, \qquad (7)$$

where the softplus$(\cdot,\cdot)$ function is expressed by

$$\mathrm{softplus}(x,k) = \frac{1}{k} \cdot \log(1 + \mathrm{e}^{k \cdot x})$$

where $\boldsymbol{\eta}^{\mathrm{R}} = [\eta_1^{\mathrm{R}}, \eta_2^{\mathrm{R}}, \eta_3^{\mathrm{R}}, \eta_4^{\mathrm{R}}, \eta_5^{\mathrm{R}}]$ are auxiliary parameters determined by the physical quantities $\boldsymbol{q}^{\mathrm{R}} = [R_1^{\mathrm{R}}, R_2^{\mathrm{R}}, R_3^{\mathrm{R}}, R_4^{\mathrm{R}}, W_1^{\mathrm{R}}, L_1^{\mathrm{R}}]$.

This function has now been mostly used due to its simplicity and efficiency in promoting sparse activations in NNs. The component values can be tuned to modify the activation threshold, allowing for the emulation of various ReLU-like behaviors. This tuning capability makes printed ReLU versatile for different applications within ANNs, providing a more dynamic response compared to fixed AFs.

3.2 Evolutionary Architecture Search

We propose an innovative EA inspired by the classical NeuroEvolution of Augmenting Topologies (NEAT) [30] to design highly bespoke pNC. The proposed algorithm not only tunes the component values within the AF circuits but also selects the most appropriate type of AF per layer per dataset to mitigate the effects of printing variation and outperform the error rate. In this problem statement, the parameters have certain constraints; conductances in the crossbar resistor arrays are limited to representing only positive weights. EAs are more convenient for considering the constraints on learnable parameters and are also suitable for non-differentiable problems, unlike gradient-based methods. As designing the circuit topology (i.e., the neural architecture) and selecting the optimal activation functional forms represent a discrete decision space, we can leverage the ability of EAs for such discrete problems.

We strategically encode the printed pNCs so that the circuit topology are jointly optimized during evolution, leading to neural architecture search. Additionally, the algorithm trains the crossbar conductances (i.e., weights) and optimizes the *type* of AF circuits for each printed neuron per dataset, along with the learnable parameters $\boldsymbol{q}$ in the AF circuits. With this enhanced search space, the resulting pNCs are expected to be more robust against printing variation, leading to a reduction in the post-mapping accuracy degradation compared to those trained by gradient methods, where only crossbar conductances and parameters in AF circuits are learned.

The key components of the proposed algorithm are shown in Fig. 3. Each genome represents a pNCs and comprises two types of genes encoding the circuit specifications: node genes (Fig. 3(a-1)) and edge genes (Fig. 3(a-2)). Initially, a population of genomes is created and categorized into multiple species based on node and edge similarities. The number of offspring for each species is determined based on its mean fitness. Top-performing genomes are preserved unchanged for the next generation, while others undergo crossover and mutation. When the termination criteria are met, the most optimized solution is obtained according to the defined fitness function, as will be discussed in Sec. 3.3. In the following,

we explain how the genome encoding, as well as the processes of crossover and mutation for the proposed algorithm, are adjusted.

Encoding. The **node gene** represents a neuron and consists of learnable parameters, including resistors R_b and R_d and learnable parameters of the candidate AF circuit and the negative weight circuit, i.e., $\boldsymbol{q}$. Additionally, it also contains a learnable and discrete variable that determines which candidate activation circuit is selected to connect to the crossbar output. Each **node gene** is uniquely identified by a global index. On the other hand, the **edge gene** denotes connections between neurons and comprises a learnable resistance R in the crossbar for weights, along with a learnable boolean parameter indicating the connectivity state (enabled/disabled) for circuit topology. Each **edge gene** is distinguished by the indices of the connected nodes, and all edges are directional, meaning $(i, j) \neq (j, i)$. Here, we denote the set of genes from all the genomes in the population as 🧬.

Crossover. In the proposed algorithm, unique genes (not shared between parents) of the fitter parent are directly inherited by the offspring, as illustrated by *node 6* in the offspring from *parent 1* in Fig. 3(b). For common genes, crossover involves the random inheritance of genes from the shared topological structure (genes with identical global index) of both parent genomes, as demonstrated in Fig. 3(b) for *node 3*, which is common between the two parents. During crossover, the features of each gene are exchanged randomly between the parents, with a higher probability of selection for the fitter parent. Subsequently, the mutation process is applied to the offspring.

Mutation. Mutation, as depicted in Fig. 3(c), is a two-stage process involving **genome-level** mutation for neural architecture and gene-level mutation for the parameters related to node and edge genes. At the **genome-level**, mutations can involve either the addition (Fig. 3(c-1)) or deletion (Fig. 3(c-2)) of edges between existing nodes, and similarly, nodes can be added (Fig. 3(c-3)) or deleted (Fig. 3(c-4)) at existing edges. When deleting a edge, a random edge is selected. Before deleting a node, edges associated with that node are removed to prevent disruption. Importantly, to avoid the extinction of new genomes, new structures should not influence genome fitness. Therefore, to maintain unchanged circuit output (and thus performance), the conductance of new edges should be initialized to zero when adding edges. Additionally, when adding a node to an edge, as shown in Fig. 3(c-3), the existing edge is disabled (not deleted) and the new node is introduced with two connections to replace said edge. To preserve the output, the conductance on edge (k, j) should be initialized by that of the edge (i, j), whereas the output of the node k should be the same as that of node i. At the **gene-level**, mutation involves perturbing crossbar conductance $\boldsymbol{\theta}$ and nonlinear circuit parameters $\boldsymbol{q}$ by adding scaled samples from a normal distribution $p(\boldsymbol{\theta})$ and $p(\boldsymbol{q})$ respectively (Fig. 3(c-5)). The type of selected activation circuit mutates randomly among [p-sigmoid, p-tanh, p-ReLU, p-clipped_ReLU] (Fig. 3(c-6)), while the state parameter of the edge is determined by a Bernoulli variable (Fig. 3(c-7)).

3.3 Variation-aware Training with NAS

Variation-aware training is critical in optimizing the reliability and performance of pNCs, which often suffer from intrinsic printing variations. We therefore integrate the proposed NAS to dynamically adjust the design of pNCs, ensuring that they not only meet desired classification accuracy, but also demonstrate resilience to printing variations. This methodology not only enhances the adaptability of pNCs but also their usefulness in real scenarios, where variations are a critical concern.

In this framework, all parameter corresponding to printed resistances, i.e., $\boldsymbol{\theta}$ and transistors, i.e., $\boldsymbol{q}$, are subject to processs variation arising from ink dispersion on the substrate, droplet jetting oddness and satellite drops wetting [31]. Each printing/processing step of the resistances and the n-EGTs (channel, dielectric, and top-gate) introduces variations resulting in non-Gaussian distributions for both the process and electrical parameters of this technology[32]. Consequently, these parameters are modeled as random variables to account for the inherent printing variations, i.e., $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$ and $\boldsymbol{q} \sim p(\boldsymbol{q})$ respectively. These variables adhere to their respective probability distributions to mirror potential deviations arising from the printing process. To evaluate the robustness of different architectures against these variabilities and thus guide the evolution process, the expected loss with respect to parameter variation is used as the training objective, namely

$$\begin{aligned} \underset{\text{🧬}}{\text{minimize}}\ \mathcal{L} &= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{q}}\left\{L(\boldsymbol{\theta},\boldsymbol{q},\mathcal{D})\right\}, \\ &= \int_{\boldsymbol{\theta}}\int_{\boldsymbol{q}} L(\boldsymbol{\theta},\boldsymbol{q},\mathcal{D})p(\boldsymbol{\theta})p(\boldsymbol{q})\,\mathrm{d}\boldsymbol{\theta}\,\mathrm{d}\boldsymbol{q}, \end{aligned} \tag{8}$$

where $\mathcal{D} = \{\boldsymbol{x}, \boldsymbol{y}\}$ refers to the target datasets, while $L(\cdot)$ refers to the cross-entropy loss [33], which is commonly used to improve classification accuracy. However, Eq. (8) poses a challenge that the optimization variable $\boldsymbol{\theta}$ and $\boldsymbol{q}$ will be integrated out. To facilitate the training of these parameters, we introduce a reparameterization strategy [34] to decouple the learnable parameters from the random variables expressing the variation. Consequently, $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \odot \boldsymbol{\varepsilon}_\theta$ and $\boldsymbol{q} = \boldsymbol{q}_0 \odot \boldsymbol{\varepsilon}_q$, where $\boldsymbol{\theta}_0$ and $\boldsymbol{q}_0$ denote target values to be optimized, while each element in $\boldsymbol{\varepsilon}_\theta$ and $\boldsymbol{\varepsilon}_q$ follows a distribution $p(\varepsilon)$ respectively. With this approach, the training objective can be reformulated as

$$\begin{aligned} \underset{\text{🧬}}{\text{minimize}}\ \mathcal{L} &= \mathbb{E}_{\boldsymbol{\varepsilon}_\theta,\boldsymbol{\varepsilon}_q}\left\{L(\boldsymbol{\theta}_0 \odot \boldsymbol{\varepsilon}_\theta, \boldsymbol{q}_0 \odot \boldsymbol{\varepsilon}_q, \mathcal{D})\right\}, \\ &= \int_{\boldsymbol{\varepsilon}_\theta}\int_{\boldsymbol{\varepsilon}_q} L(\boldsymbol{\theta}_0 \odot \boldsymbol{\varepsilon}_\theta, \boldsymbol{q}_0 \odot \boldsymbol{\varepsilon}_q, \mathcal{D})p(\boldsymbol{\varepsilon}_\theta)p(\boldsymbol{\varepsilon}_q)\,\mathrm{d}\boldsymbol{\varepsilon}_\theta\,\mathrm{d}\boldsymbol{\varepsilon}_q, \end{aligned} \tag{9}$$

With Eq. (9), the parameters $\boldsymbol{\theta}_0$ and $\boldsymbol{q}_0$ can be trained to guarantee the optimal classification accuracy under the expectation of given variation $p(\boldsymbol{\varepsilon}_\theta)$ and $p(\boldsymbol{\varepsilon}_q)$. However, the integration in Eq. (9) still has no closed form, which poses challenge to its optimization. For this, we employ an estimation of the integration through Monte-Carlo sampling, i.e.,

$$\mathcal{L} \approx \frac{1}{N}\sum_{n=1}^{N} L(\boldsymbol{\theta}_0 \odot \boldsymbol{\varepsilon}'_\theta, \boldsymbol{q}_0 \odot \boldsymbol{\varepsilon}'_q, \mathcal{D}), \tag{10}$$

Table 4: Simulation Result and Runtime of gradient-based approach without variation and comparison with EA with baseline in (i) high precision printing (5% variation) and (ii) low-precision printing (10% variation) on 13 Benchmark Datasets.

Dataset	Reference accuracy (without variation)	High-precision printing (±5%)		Low-precision printing (±10%)		Runtime	
		Baseline	EA	Baseline	EA	Baseline (min)	EA (min/pop)
Acute Inflammation	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.999 ± 0.012	1.000 ± 0.000	183.9	9.5
Balance Scale	0.902 ± 0.017	0.880 ± 0.004	0.896 ± 0.008	0.877 ± 0.008	0.881 ± 0.012	205.9	21.6
Breast Cancer Wisconsin	0.971 ± 0.001	0.963 ± 0.008	0.966 ± 0.006	0.931 ± 0.039	0.949 ± 0.012	180.9	11.6
Cardiotocography	0.879 ± 0.007	0.774 ± 0.004	0.857 ± 0.005	0.763 ± 0.002	0.794 ± 0.007	178.0	19
Energy Efficiency (y_1)	0.915 ± 0.019	0.889 ± 0.032	0.916 ± 0.026	0.847 ± 0.012	0.866 ± 0.011	194.6	15.1
Energy Efficiency (y_2)	0.894 ± 0.016	0.883 ± 0.023	0.891 ± 0.038	0.867 ± 0.026	0.866 ± 0.021	189.4	10.3
Iris	0.965 ± 0.005	0.912 ± 0.034	0.923 ± 0.050	0.843 ± 0.045	0.882 ± 0.039	178.8	7.3
Mammographic Mass	0.788 ± 0.003	0.782 ± 0.017	0.810 ± 0.018	0.766 ± 0.053	0.764 ± 0.055	190.9	5.0
Pendigits	0.577 ± 0.054	0.554 ± 0.038	0.559 ± 0.039	0.548 ± 0.047	0.553 ± 0.050	198.3	14.2
Seeds	0.891 ± 0.031	0.820 ± 0.034	0.851 ± 0.023	0.820 ± 0.041	0.827 ± 0.007	176.0	6.4
Tic-Tac-Toe Endgame	1.000 ± 0.001	0.713 ± 0.012	0.765 ± 0.018	0.660 ± 0.017	0.716 ± 0.019	177.1	6.4
Vertebral Column (2 cl.)	0.830 ± 0.007	0.716 ± 0.007	0.794 ± 0.004	0.661 ± 0.000	0.685 ± 0.004	180.6	4.6
Vertebral Column (3 cl.)	0.811 ± 0.010	0.634 ± 0.086	0.791 ± 0.016	0.634 ± 0.075	0.734 ± 0.059	130.9	9.6
Average	**0.879 ± 0.013**	**0.809 ± 0.023**	**0.848 ± 0.019**	**0.786 ± 0.029**	**0.809 ± 0.023**	181.9	10.81

where $\boldsymbol{\varepsilon}'_\theta \sim p(\boldsymbol{\varepsilon}_\theta)$ and $\boldsymbol{\varepsilon}'_q \sim p(\boldsymbol{\varepsilon}_q)$ are samples drawn from their respective distribution in each calculation of L. Moreover, N denotes the number of samples utilized to estimate the integration.

Finally, Eq. (10) is utilized within the training objective (i.e., fitness function $f(x)$)) of the variation-aware training. For the training objective, we also consider the expected classification accuracy ACC, i.e.,

$$f(\mathbf{x}) = ACC - \mathcal{L}, \tag{11}$$

With this objective, NAS aims not only to improve the classification accuracy of pNCs, but also improve the robustness against intrinsic stability with respect to variations. This process involves an assortment of optimization techniques, which may extend beyond conventional gradient-based methods, to seek out architectures that guarantee reliable performance despite the unpredictable nature of the printing process. The ultimate aim of using NAS in this context is to strike an optimal balance between performance and resilience, ensuring proper operation in real scenarios.

4 EVALUATION

To assess the effectiveness of the proposed method, we utilized PyTorch to implement the algorithm[1] and carried out experiments on 13 benchmark datasets. These datasets are also used in other state-of-the-art studies on pNCs [12, 35], and match the complexity within the application domains of PE.

4.1 Experiment Setup

We conduct training on the pNC utilizing the EA methodology and test it on 13 benchmark datasets against the established gradient-based optimization techniques as a baseline of this work.

Circuit Setup. The AF circuits in Fig. 2 (top (e), (f), (g)) were designed based on the well-developed n-EGT P-PDK [20] and the ranges of learnable parameters are determined by performing sweep analysis. We used Cadence Virtuoso[2] tool to simulate the transfer characteristics (as shown in Fig. 2 (bottom (e), (f), (g))) in SPICE.

Initialization. Drawing insights from other works on EA and guided by a series of preliminary trials, we have strategically initialized the network topologies for all datasets as unconnected networks, which consist solely of nodes corresponding to the number of outputs, featuring only #output nodes. The population for these experiments is robustly set at 1, 000 individuals. Each node is initialized to have a random AFs circuit among the given design.

In terms of the mutation mechanisms employed, we have defined specific probabilities for the genetic alterations within the network structures: the probability of introducing either a new node or a new connection is set at a substantial rate of 0.7, while the probability for the deletion of a node or a connection is comparatively lower, at 0.3. Moreover, there exists a 0.1 chance that any given edge within the network will toggle its state from enabled to disabled, or vice versa, as part of the mutation process. Moreover, the mutation rate of changing selected AFs circuit is 0.1.

In terms of variation, we take uniform distribution, i.e., $\boldsymbol{\varepsilon}_\theta \sim \mathcal{U}[1-\epsilon, 1+\epsilon]$ and $\boldsymbol{\varepsilon}_q \sim \mathcal{U}[1-\epsilon, 1+\epsilon]$, to reflect the printing variation, because the printing variation is primarily determined by the geometric variation of the printing shape which varies within one printing pixel. More specific, we select an $\epsilon = 5\%$ to simulate a relatively high printing precision, while another $\epsilon = 10\%$ to simulates a relatively low printing precision. This is because the typical printing resolutions range from 20 µm to 100 µm [19], whereas the component feature sizes in printed neuromorphic circuits are on the order of 1 mm [35]. Moreover, for Monte-Carlo sampling, we select $N = 20$ for numerical estimation of the integration, as it can already yield sufficiently precise estimation in our experiments.

Training. In training (evolution) process, we utilize a full-batch training, with termination upon a patience threshold of 100 generations. This specific criterion hinges on observing no significant improvement in the performance metrics on the validation dataset over the aforementioned span of generations. To ensure that our findings are statistically reliable and to mitigate the variability due to stochastic elements of the training process, we repeat the training sessions ten times for each value of γ, employing different random seeds for each session, varying from 1 to 10. This repetition ensures that we achieve sufficiently optimal and robust solutions.

Baseline. To conduct experiment with baseline approach, we perform training with topologies initialized as #input-3-#output. We use the Adam [36] optimizer with default parameterization to train parameters. We start with an initial learning rate of 0.1 and halve it after a 100-epoch patience. Additionally, the training

[1] https://github.com/Neuromophic/eNAS_learnable_selectable_LNC.
[2] https://www.cadence.com/en_US/home.html

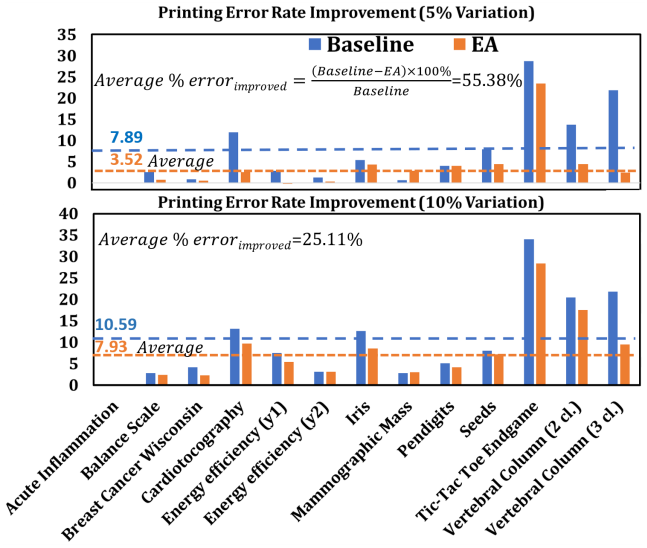

Figure 4: Results of the printing error rate enhancement in 13 benchmark datasets with (i) ±5% and (ii) ±10% variation.

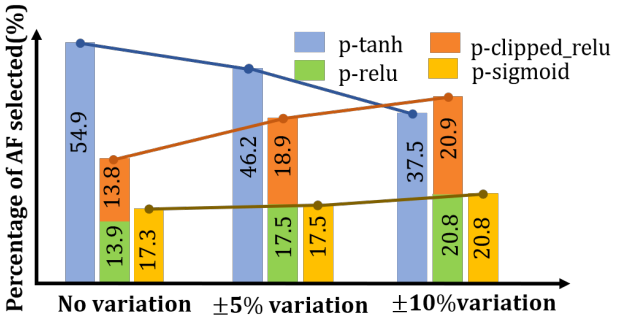

Figure 5: Percentage of AFs selected during (i) no (ii) ±5% and (iii) ±10% variation averaged over 13 benchmark datasets.

process is stopped, when the learning rate was halved 10 times. Other setups are kept the same as its EA counterpart.

To provide an upper bound classification accuracy of each dataset, we also trained the pNCs without any variation, i.e., $\epsilon = 0$. Because the accuracy in this case can be seen as the theoretically highest achievable values. We denote the accuracy in this case as the *reference accuracy*.

4.2 Result

After completing the training of all pNCs, we carefully select the most optimal pNCs w.r.t to the random seed in each experimental setup based on their performance on the validation loss. These selected circuits are the ones designed for physical realization. Subsequently, we evaluate their performance on the test sets. In testing, all pNCs are tested and trained under the identical variations.

The mean, standard deviation of their accuracy and their corresponding runtime performances are presented in Tab. 4, showing a scalar average across all datasets for a clearer comparison of different training configurations. Additionally, Fig. 4. illustrates the improvement in printing error rates. Also, Fig. 5 details the selection percentages of various learnable AF circuits, highlighting their preferences during the training process.

4.3 Discussion

Tab. 4 compares the performance of the EAs to a baseline gradient-based method across 13 benchmark datasets under conditions of high-precision (±5%) and low-precision (±10%) printing. The reference accuracy shows the performance without variation, providing a reference point for evaluating the resilience of the algorithms under variation. For both scenarios (±5% and ±10%), EA maintains comparable accuracy to the baseline across all datasets. Also, runtime analysis reveals that EA's runtime depends on the population size and the extent of parallel processing utilized. EA can significantly reduce runtime through parallel processing, contrary to the baseline method's total sequential runtime and can offer more robust solutions per iteration.

In Fig. 4, EAs demonstrate robust error rate, with a significant enhancement of 55.38% in ±5% and 25.11% in ±10% scenarios, compared to the baseline, thus highlighting the EAs capability to effectively manage printing variations, thereby reducing error rates.

To the end, Fig. 5 illustrates the selection percentages of different AFs used in EA under varying printing conditions. In worst-case scenario with ±10% variation, the ReLU family emerges as more favored, selected ≈ 21% of the time, up from ≈ 14% with no variation. This trend-line shift of the ReLU family suggests their ability to induce sparsity in NNs, encouraging the exploration of diverse solutions by focusing on relevant features. This leads to performance stability and better generalization, making ReLU-based NNs more suitable and robust in environments with higher uncertainty and variability. In contrast, p-tanh is the most preferred, chosen ≈ 55% of the time, due to its high sensitivity and effectiveness in stable conditions (no variation). However, p-sigmoid consistently maintains a steady selection rate across all levels of variation.

Therefore, it is worthy to conclude that the various learnable AF circuits and variation-aware training using NAS approach both contribute to a significant improvement in classification error-rate and robustness of pNCs.

5 CONCLUSION

Printed Electronics, owing to their distinctive features, are gaining significant attention for the evolution of next-generation electronics. In this realm, analog printed neuromorphic circuits are becoming increasingly popular due to their ability to provide customized computational functions at a minimal cost for target applications. However, the maskless additive manufacturing process increases variability, challenging consistent performance.

To enhance the robustness of printed neuromorphic circuits and mitigate the effects of printing variations, we propose an evolutionary algorithm to train printed neuromorphic circuits with selectable activation circuits and optimal neural architecture, including parameters like crossbar conductances and physical quantities in nonlinear circuits. This approach not only opens new optimization opportunities but also enhances robustness against printing variations. Future research may further explore parallel computing using evolutionary algorithms.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Arif U Alam et al. Fruit Quality Monitoring with Smart Packaging. *Sensors*, 21(4):1509, 2021.

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

[3] Haibin Zhao et al. Towards Temporal Information Processing–Printed Neuromorphic Circuits with Learnable Filters. In *Proceedings of the 18th ACM International Symposium on Nanoscale Architectures*, pages 1–6, 2023.

[4] Ehsan Shirzaei Sani et al. A Stretchable Wireless Wearable Bioelectronic System for Multiplexed Monitoring and Combination Treatment of Infected Chronic Wounds. *Science Advances*, 9(12):7388, 2023.

[5] William S Wong and Alberto Salleo. *Flexible electronics: materials and applications*, volume 11. Springer Science & Business Media, 2009.

[6] Wei Gao, Hiroki Ota, Daisuke Kiriya, Kuniharu Takei, and Ali Javey. Flexible electronics toward wearable sensing. *Accounts of chemical research*, 52(3):523–533, 2019.

[7] Haibin Zhao, Tobias Röddiger, and Michael 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*, pages 180–184, 2021.

[8] Hamed Abdolmaleki, Preben Kidmose, and Shweta Agarwala. Droplet-based techniques for printing of functional inks for flexible physical sensors. *Advanced Materials*, 33(20):2006792, 2021.

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

[10] Florentia Afentaki, Gurol Saglam, Argyris Kokkinis, Kostas Siozios, Georgios Zervakis, and Mehdi B. Tahoori. Bespoke approximation of multiplication-accumulation and activation targeting printed multilayer perceptrons. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, October 2023.

[11] Giorgos Armeniakos et al. Cross-layer approximation for printed machine learning circuits. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 190–195, 2022.

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

[13] Derya Baran, Daniel Corzo, and Guillermo Blazquez. Flexible electronics: Status, challenges and opportunities. *Frontiers in Electronics*, 1, 2020.

[14] Jyoti Bala Kaushal, Pratima Raut, and Sanjay Kumar. Organic electronics in biosensing: A promising frontier for medical and environmental applications. *Biosensors*, 13(11), 2023.

[15] Farhan Rasheed, Michael Hefenbrock, Michael Beigl, Mehdi B. Tahoori, and Jasmin Aghassi-Hagmann. Variability modeling for printed inorganic electrolyte-gated transistors and circuits. *IEEE Transactions on Electron Devices*, 66(1):146–152, 2019.

[16] Robert Hecht-Nielsen. Theory of the Backpropagation Neural Network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

[17] Davis Blalock et al. What is the State of Neural Network Pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

[18] Hanxiao Liu et al. DARTS: Differentiable Architecture Search. In *International Conference on Learning Representations (ICLR)*, 2018.

[19] Saleem Khan et al. Technologies for printing sensors and electronics over large flexible substrates: A review. *IEEE Sensors Journal*, 15:3164, 2014.

[20] Farhan Rasheed, Michael Hefenbrock, Rajendra Bishnoi, Michael BeigI, Jasmin Aghassi-Hagmann, and Mehdi B. Tahoori. Predictive modeling and design automation of inorganic printed electronics. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 30–35, 2019.

[21] Farhan Rasheed, Michael Hefenbrock, Michael Beigl, Mehdi B. Tahoori, and Jasmin Aghassi-Hagmann. Variability modeling for printed inorganic electrolyte-gated transistors and circuits. *IEEE Transactions on Electron Devices*, 66(1):146–152, 2019.

[22] Changwoo Lee, Hyunkyoo Kang, Hojoon Kim, Ho Anh, Duc Nguyen, and Keehyun Shin. Quality control with matching technology in roll to roll printed electronics. *Journal of Mechanical Science and Technology*, 24(1):315, 2010.

[23] Enrico Sowade, Maxim Polomoshnov, and Reinhard R. Baumann. The design challenge in printing devices and circuits: Influence of the orientation of print patterns in inkjet-printed electronics. *Organic Electronics*, 37:428–438, 2016.

[24] Min-Hye Oh. A weight variation-aware training method for hardware neuromorphic chips, 2023.

[25] Yifan Pan, Zichang He, Nanlin Guo, and Zheng Zhang. Distributionally robust circuit design optimization under variation shifts, 2023.

[26] Saurabh K Tiwary, Amith Singhee, and Vikas Chandra. Robust circuit design: Challenges and solutions. In *2009 22nd International Conference on VLSI Design*, pages 41–42, 2009.

[27] Giuseppe Nicosia, Salvatore Rinaudo, and Eva Sciacca. An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization. In Max Bramer, Frans Coenen, and Miltos Petridis, editors, *Research and Development in Intelligent Systems XXIV*, pages 7–20, London, 2008. Springer London.

[28] Jin Xu et al. Reluplex made more practical: Leaky ReLU. In *2020 IEEE Symposium on Computers and communications (ISCC)*, pages 1–7. IEEE, 2020.

[29] Qian Liu and Steve Furber. Noisy softplus: A biology inspired activation function. In *Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part IV 23*, pages 405–412. Springer, 2016.

[30] Kenneth O Stanley et al. Efficient Evolution of Neural Network Topologies. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*, volume 2, pages 1757–1762. IEEE, 2002.

[31] Enrico Sowade, Eloi Ramon, Kalyan Yoti Mitra, Carme Martínez-Domingo, Marta Pedró, Jofre Pallarès, Fausta Loffredo, Fulvia Villani, Henrique L Gomes, Lluís Terés, et al. All-inkjet-printed thin-film transistors: manufacturing process reliability by root cause analysis. *Scientific reports*, 6(1):33490, 2016.

[32] Farhan Rasheed et al. Variability Modeling for Printed Inorganic Electrolyte-gated Transistors and Circuits. *IEEE transactions on electron devices*, 66(1):146–152, 2018.

[33] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy Loss Functions: Theoretical Analysis and Applications. In *International Conference on Machine Learning*, pages 23803–23828. PMLR, 2023.

[34] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[35] Dennis D Weller et al. Realization and Training of an Inverter-based Printed Neuromorphic Computing System. *Scientific reports*, 11(1):1–13, 2021.

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