

Flexible and Printed Electronics



PAPER

Highly-dependable printed neuromorphic circuits based on additive manufacturing

OPEN ACCESS

RECEIVED
5 March 2023

REVISED
2 May 2023

ACCEPTED FOR PUBLICATION
25 May 2023

PUBLISHED
7 June 2023

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Haibin Zhao^{1,*} , Michael Hefenbrock², Michael Beigl¹ and Mehdi B Tahoori¹

¹ Karlsruhe Institute of Technology, Karlsruhe, Germany

² RevoAI GmbH, Karlsruhe, Germany

* Author to whom any correspondence should be addressed.

E-mail: haibin.zhao@kit.edu

Keywords: printed electronics, printed neuromorphic circuits, printed neural networks, variation-aware training, aging-aware training, sensing uncertainty

Abstract

The rapid development of emerging domains, such as the Internet of Things and wearable technologies, necessitates the development of flexible, stretchable, and non-toxic devices that can be manufactured at an ultra-low cost. Printed electronics has emerged as a viable solution by offering not only the aforementioned features but also a high degree of customization, which enables the personalization of products and facilitates the low-cost product development process even in small batches. In the context of printed electronics, printed neuromorphic circuits offer highly customized and bespoke realization of artificial neural networks to achieve desired functionality with very small number of hardware components. However, since analog components are utilized, the performance of printed neuromorphic circuits can be influenced by various factors. In this work, we focus on three main factors that perturb the circuit output from the designed values, namely, variations due to printing errors, aging effects of printed resistors, and input variations originating from sensing uncertainty. In the described approach, these variations are taken into account during the design (training) to ensure the dependability of the printed neuromorphic circuits. With this approach, the expected accuracy and the robustness of printed neural networks can be increased by 27% and 74%, respectively. Moreover, the ablation study suggests that, aging effect and printing variation may have similar effects on the functionality of printed neural networks. In contrast, the impact of sensing uncertainty on printed neural networks is almost orthogonal to aging and printing variations.

1. Introduction

The rapidly advancing domains, such as Internet of Things (IoT) [1], wearable devices [2], and smart packaging [3], have resulted in a heightened requirement for flexible, bio-compatible, and low-cost electronics. In this context, printed electronics (PE) becomes a promising solution, as PE allows the manufacturing of flexible and stretchable circuitry, with possibly non-toxic materials through a spectrum of material [4, 5] and substrate [6, 7] options. Additionally, due to the additive manufacturing process, printed devices can be fabricated in a low cost.

The exceptional properties of PE have led to a growing interest in exploring the potential of printed sensing devices, ranging from environmental

sensing [8] to on-body sensing [9]. Printed sensing devices are highly regarded for their simple additive fabrication process, cost-effectiveness, and versatility. Therefore, they are poised to play a key role in the advancement of numerous fields and industries.

Apart from printed sensors, printed neuromorphic circuits [10], as a printed computing scheme, represent a cutting-edge intersection of PE and machine learning. They are capable of emulating operations of artificial neural networks (including weighted-sum operations and nonlinear activations) through the interconnection of multiple simple primitive subcircuits, such as resistor crossbars and nonlinear transformation circuits. The inherent commonality between the operations performed in printed neuromorphic circuits

and those in artificial neural networks allows printed neuromorphic circuits to express sophisticated (nonlinear) transformations. Moreover, this commonality enables a highly efficient training-based design process of printed neuromorphic circuits. Consequently, printed neuromorphic circuits serve as a promising printed computing paradigm.

However, since analog components are utilized in printed neuromorphic circuits, their correct functionality is subject to the influence of multiple factors. Firstly, input variations caused by uncertainty in the sensing process [11] may cause faulty processing. Secondly, variations in the printing process due to non-uniformly printed material (device geometry) as well as variations in ink compositions and substrates can perturb the fabricated component values from the design ones [12]. Finally, the printed components can deteriorate due to environmental influences (particularly thermal stress) over time [13]. This aging effect can also severely impact the performance of printed neuromorphic circuits.

To ensure the reliability and accuracy of the outputs of printed neuromorphic circuits, the aforementioned factors should be considered in the design of printed neuromorphic circuits. In this work, we propose an approach to enhance the robustness of printed neuromorphic circuits against influences such as stochastic sensing uncertainty (SU), stochastic printing variation (PV), and stochastic aging behaviors (AG) of printed resistors. To integrate these factors into the design of the printed neuromorphic circuit, we first model these factors with stochastic variables. Afterwards, a modified training objective that accounts for these factors is employed to improve expected circuit performance. In addition, an ablation study is conducted to analyze the significance of the aforementioned three influences.

The organization of the rest of the paper is structured as follows: section 2 provides a comprehensive overview of PE, printed neuromorphic circuits, and relevant literature. In section 3, we mathematically model the uncertain inputs, printing variations, and stochastic aging behaviors from a probabilistic perspective, and introduce the modified objective to improve the robustness of printed neural networks. The effectiveness of the proposed method is evaluated through extensive experiments in section 4 and an ablation study is performed to analyze the influence of different factors separately. Finally, section 5 concludes this work and discusses the future work.

2. Preliminary

As a preliminary, we briefly review the necessary background knowledge regarding printed electronics, printed neuromorphic circuits, printed neural networks, and some related work.

2.1. Printed electronics

Printed electronics (PE) is an emerging technology that allows fabricating electronic components and devices with various printing technology, such as gravure printing and jet printing. By adopting advanced materials and substrates, flexible, lightweight, and even bio-compatible electronic devices can be fabricated at an ultra-low cost. Although PE faces challenges in terms of performance and stability when compared to silicon-based technologies, it offers several advantages including flexibility, customizability, and cost-effectiveness. These advantages enhance PE as a promising candidate for many emerging applications, such as wearable devices and personalized products. Actually, PE and silicon-based technologies do not compete with each other, but rather work in conjunction to alleviate the limitations of each technology.

According to the flow of materials, printing technologies are broadly categorized into two main classes: purely additive manufacturing and mixed manufacturing involving both additive and subtractive processes [14]. As shown in figure 1, purely additive manufacturing only relies on the deposition of materials in successive layers to produce components, such as transistors. This approach allows significantly lower fabrication costs compared to the mixed process, but, in contrast, it is prone to lower manufacturing efficiency, larger feature sizes, and increased printing variations (errors). Conversely, mixed manufacturing incorporates both deposition (additive) and etching (subtractive) steps, which is similar to conventional silicon-based processing. Since the subtractive process generally requires specialized infrastructures, the cost is usually higher than the purely additive printing. In this regard, the inkjet-printed electrolyte-gated transistor technology is considered an effective enabler for IoT infrastructures, wearable devices, and low-energy devices, as it allows not only low operating voltage (sub-1V) but also low manufacturing cost due to its purely additive process.

Flexible PE devices are generally printed utilizing contactless printing techniques, such as inkjet-printing, on flexible substrates such as Kapton [6] or polyethylene terephthalate [7]. With highly optimized functional inks, electronic components like organic [15] or oxide-based [16] transistors can be fabricated. While the preparation of organic materials entails a relatively easy processing procedure, they exhibit less stability when subjected to environmental alterations. Conversely, oxide-based inks exhibit remarkable conductivity and environmental stability, however, their printing process may be hindered by the presence of surfactants and impurities [17].

Despite the favorable features of PE, there are several limitations that pose challenges for its widespread adoption, which are the large feature sizes

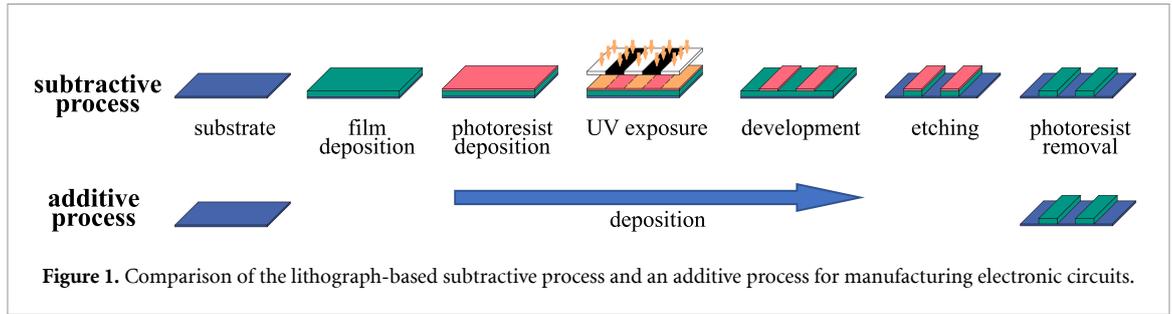


Figure 1. Comparison of the lithograph-based subtractive process and an additive process for manufacturing electronic circuits.

and high parasitic capacitances. Consequently, the printed devices generally suffer from low functional density and elevated device latency. To mitigate these challenges, the trend in printed circuits is toward low complexity and containing limited number of transistors, and thereby, reducing area utilization and increasing manufacturability. Following this trend, several fundamental components of printed computing systems have been successfully realized, including but not limited to Boolean logic [18], digital/analog storage elements [19, 20], and amplifiers [21]. Additionally, the production process of printed circuits frequently displays high variation [14] and printed components exhibit aging due to environmental impacts such as humidity and thermal stress [13]. To address these problems, the variation and aging are considered during the design phase [22, 23].

2.2. Printed neuromorphic circuits

Neuromorphic computing is a computational paradigm that mimics the signal processing mechanism of human brains by adopting a sequence of weighted-sum operations followed by nonlinear activation functions. Despite the simplicity of the primitive operations, neuromorphic computing has shown strong (nonlinear) expressiveness [24] and has achieved notable successes in multiple domains [25]. Moreover, the simple and differentiable computing operations enable a highly efficient optimization process of the parameters through backpropagation and facilitates the implementation of neuromorphic computing at the hardware level. Printed neuromorphic circuit refers to printed circuit that is capable of executing equivalent operations in neuromorphic computing. Specifically, the weighted-sum operations are implemented by resistor crossbars and nonlinear activations are realized by circuits with nonlinear characteristics. In addition, to address the limitations imposed by the hardware, other subcircuits, such as negative weight circuits, may also be incorporated into the design. Figures 2(a)–(c) show some exemplary schematics of the mentioned primitive subcircuits in a printed neuromorphic circuit and figure 2(d) represents an interconnection of those subcircuits to a three-layer network with 3, 2, and 4 printed neurons in each layer.

2.2.1. Resistor crossbar

Resistor crossbar has been identified as the prevalent choice for implementing weighted-sum operation and has been widespread utilized in various applications, including in-memory computing [26] and ReRAM-based neural network acceleration [27], etc. Figure 2(a) demonstrates a typical resistor crossbar in a printed neuromorphic circuit with three input voltages V_1, V_2, V_3 and one output voltage V_z . In the crossbar, there are also two internal constant voltages, namely $V_b^C = 1V$ and GND that refers to $0V$. Each external input voltage and internal constant voltage is connected to the corresponding resistor, namely $R_1^C, R_2^C, R_3^C, R_b^C$, and R_d^C . Here, the superscript $(\cdot)^C$ indicates the variables in the crossbar subcircuit. Following Kirchhoff's law, we obtain

$$\sum_{i=1}^3 \frac{V_i - V_z}{R_i^C} + \frac{V_b^C - V_z}{R_b^C} - \frac{V_z}{R_d^C} = 0.$$

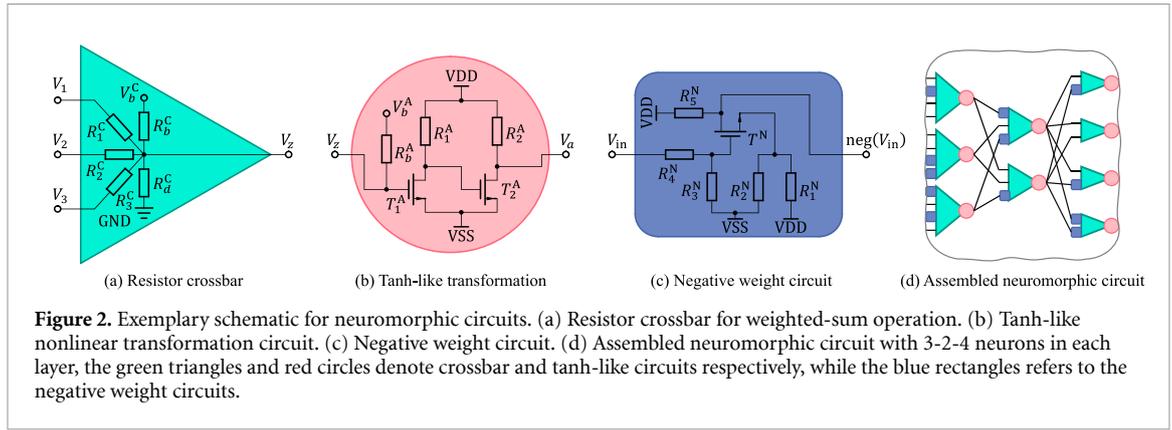
By further replacing the resistance R with the corresponding conductance $g = 1/R$, the equation can be formulated

$$V_z = \sum_{i=1}^3 \frac{g_i^C}{g_{\text{sum}}^C} V_i + \frac{g_b^C}{g_{\text{sum}}^C}, \quad (1)$$

where g_{sum}^C refers to the sum of all conductances in the crossbar. Thus, the output of the crossbar can be described by the weighted-sum of the input voltages with weights and bias expressed by conductance values. In this way, the desired weighted-sum operation can be implemented by printing specific conductance values in the crossbar. Section 2.3 gives more details about the design of resistors in crossbar.

2.2.2. Tanh-like transformation circuit

In neuromorphic computing, a nonlinear activation function is generally adopted after the weighted-sum to introduce nonlinearity to the computing system. In printed neuromorphic circuits, various nonlinear circuits with characteristic curves resembling classic activation functions have been proposed, such as ReLU function [28] and sigmoid function [29]. Figure 2(b) exemplifies an inverter-based printed tanh-like (ptanh) transformation circuit. The advantage of this circuit is the presence of a super-linear interval between input and output voltages.



This feature effectively mitigates signal loss at the layer output and can provide possibilities for cross-layer amplification [10]. Its characteristic curve can be described by an accordingly parameterized tanh function, i.e.

$$V_a = \text{ptanh}(V_z) = \eta_1^A + \eta_2^A \cdot \tanh((V_z - \eta_3^A) \cdot \eta_4^A),$$

where $\boldsymbol{\eta}^A = [\eta_1^A, \eta_2^A, \eta_3^A, \eta_4^A]$ is the set of auxiliary parameters that translate and scale the original tanh function [10]. The superscript $(\cdot)^A$ indicates that the parameters belong to the activation function. The auxiliary parameters $\boldsymbol{\eta}^A$ are determined by the circuit components $\boldsymbol{q}^A = [R_1^A, R_2^A, R_b^A, W_1^A, L_1^A, W_2^A, L_2^A]$, in which W_i^A and L_i^A are the geometric features that decide the characteristic of the printed transistor T_i^A . Thus, to be precise, we denote the function $\text{ptanh}(\cdot)$ by $\text{ptanh}_{\boldsymbol{q}^A}(\cdot)$.

2.2.3. Negative weight circuit

By observing equation (1), it is evident that the weights in the crossbar are expressed by the conductances, which are restricted to positive values only. To overcome this drawback and express negative relationships, the negative weight circuit is proposed in [10]. Compared to existing techniques [30], the proposed negative weights are only printed circuit when required. Consequently, the resulting printed circuits exhibit lower area, power consumption, and a decreased transistor counts.

As shown in figure 2(d), negative weight circuits are prepended to the input of the crossbar to convert the input voltages to negative ones whenever negative weights are required. In other words, negative weights are realized by inverting the respective inputs. The characteristic curve of the negative weight circuit is nonlinear and, similar to the ptanh circuit, can be modeled through an accordingly parameterized tanh function as

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

with $\boldsymbol{\eta}^N = [\eta_1^N, \eta_2^N, \eta_3^N, \eta_4^N]$ as the auxiliary parameters determined by its circuit component $\boldsymbol{q}^N = [R_1^N, R_2^N, R_3^N, R_4^N, R_b^N, W^N, L^N]$. Here, the superscript

$(\cdot)^N$ denotes the values corresponding to the negative weight circuit. Analogous to the ptanh circuit, we denote the function by $\text{neg}_{\boldsymbol{q}^N}(\cdot)$.

By combining these primitive subcircuits, printed neurons can be constructed. Then, through the interconnection of multiple printed neurons, more sophisticated printed neuromorphic circuits for more complicated computing tasks can be realized.

2.3. Printed neural network

In order to leverage the computing capabilities of printed neuromorphic circuits, a design and optimization process is necessary. Considering the target application domains of PE and the high demand for low-cost, these circuits do not adopt reconfigurable components for implementing on-device training during operation. Instead, their designs are conducted off-device at the software level. The fabrication process is initiated after the circuit design. For this purpose, printed neural network (pNN) is proposed in [10]. pNNs are machine learning-based models that simulate the behavior of printed neuromorphic circuits. The learnable parameters in a pNN are the design space (component values) of the corresponding printed neuromorphic circuit. In this regard, training of a pNN can be seen as designing the respective printed neuromorphic circuit.

2.3.1. Inference

In pNNs, the learnable parameter for weighted-sum, referred to as the surrogate conductance θ_i , encodes both printing conductance through its absolute value, i.e. $g_i^C = |\theta_i|$, and the existence of a negative weight circuit by its sign, i.e. $\text{sign}(\theta_i)$. With this encoding, the weights in a printed neuron can be expressed by

$$w_i = \frac{g_i^C}{g_{\text{sum}}^C} = \frac{g_i^C}{(\boldsymbol{g}^C)^\top \cdot \mathbf{1}} = \frac{|\theta_i|}{|\boldsymbol{\theta}|^\top \cdot \mathbf{1}},$$

where $\boldsymbol{g}^C = [g_1^C, g_2^C, \dots, g_b^C, g_d^C]^\top$, $\mathbf{1} = [1, 1, \dots, 1]^\top$, $|\cdot|$ refers to element-wise absolute operation, and $\text{diag}(\cdot)$ yields a diagonal matrix from the given vector. Consequently, the weighted-sum can be expressed by

$$\sum_i V_i \cdot \frac{|\theta_i|}{|\boldsymbol{\theta}|^T \cdot \mathbf{1}} \cdot \mathbb{1}_{\{\theta_i \geq 0\}} + \text{neg}_{q^N}(V_i) \cdot \frac{|\theta_i|}{|\boldsymbol{\theta}|^T \cdot \mathbf{1}} \cdot \mathbb{1}_{\{\theta_i < 0\}}, \quad (2)$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function that is 1 if the respective condition is true, else 0. Note that, the bias term and GND term can also be included in equation (2) by augmenting the input voltage with a 1 (for V_b^C) and a 0 (for GND).

In case of batch training (with batch size B), the input voltage is often denoted by $\mathbf{V}_{\text{in}} \in \mathbb{R}^{B \times M}$, the output of the printed layer (with N neurons) is denoted by $\mathbf{V}_{\text{out}} \in \mathbb{R}^{B \times N}$, and the conductance matrix of this layer is denoted by $\boldsymbol{\Theta} \in \mathbb{R}^{M \times N}$, with each column indicating one printed neuron. Here, M implies the number of input voltages, including two augmented voltages for V_b and GND. In this context, the weight matrix can be formulated as

$$\mathbf{W} = |\boldsymbol{\Theta}| \cdot \text{diag}(|\boldsymbol{\Theta}|^T \cdot \mathbf{1}) \in \mathbb{R}^{M \times N},$$

where $\mathbf{1} \in \mathbb{R}^M$ is a vector of all ones, and thus, the weighted-sum can be described by

$$\mathbf{V}_z = \mathbf{V}_{\text{in}} \cdot (\mathbf{W} \odot \mathbb{1}_{\{\boldsymbol{\Theta} \geq 0\}}) + \text{neg}_{q^N}(\mathbf{V}_{\text{in}}) \cdot (\mathbf{W} \odot \mathbb{1}_{\{\boldsymbol{\Theta} < 0\}}) \in \mathbb{R}^{B \times N},$$

where \odot denotes the element-wise product and the indicator function $\mathbb{1}_{\{\cdot\}}$ is applied element-wise on $\boldsymbol{\Theta}$. Thus, the output of the printed layer becomes

$$\mathbf{V}_{\text{out}} = \text{ptanh}_{q^A} \left(\mathbf{V}_{\text{in}} \cdot (\mathbf{W} \odot \mathbb{1}_{\{\boldsymbol{\Theta} \geq 0\}}) + \text{neg}_{q^N}(\mathbf{V}_{\text{in}}) \cdot (\mathbf{W} \odot \mathbb{1}_{\{\boldsymbol{\Theta} < 0\}}) \right).$$

In addition to the learnable parameters $\boldsymbol{\Theta}$ for weights, the parameters for nonlinear circuits, i.e. q^A and q^N , can also be learned through the *surrogate nonlinear circuit models*, which are differentiable transformation models that can map q^A and q^N to η^A and η^N respectively. With surrogate nonlinear circuit models, the parameters q^A and q^N are firstly transformed to the corresponding auxiliary parameters η^A and η^N , and then, involved into the pNN as activation functions and negative weight functions. Finally, they can be optimized through gradient-based algorithms alongside the learnable conductance $\boldsymbol{\Theta}$ [31].

2.3.2. Constraints

The training of machine learning models is generally an unconstrained optimization process, which may lead the learned parameters to exceed the constraints imposed by the printing technology. To ensure the printability of the designed printed neuromorphic circuits, it is imperative to introduce additional processing measures that guarantee the limitations of the printing technology.

The primary constraint is the limited printable conductance range, specifically, $g \in \{0\} \cup [g_{\text{min}}, g_{\text{max}}]$

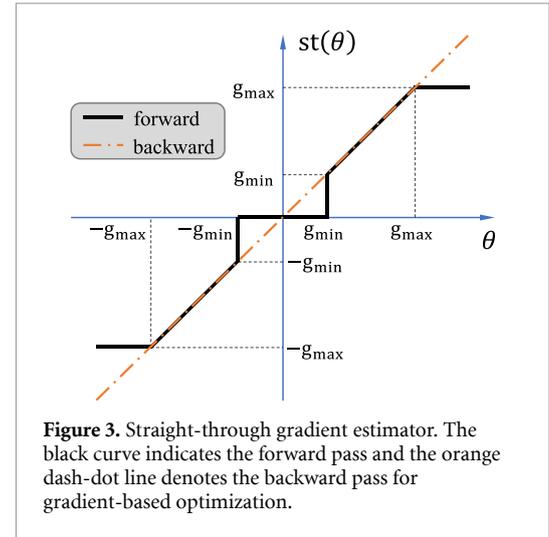


Figure 3. Straight-through gradient estimator. The black curve indicates the forward pass and the orange dash-dot line denotes the backward pass for gradient-based optimization.

(0 refers to not printing the respective resistor). This range is contingent on the technology and materials utilized for printing. In this case, each element in the learnable conductance $\boldsymbol{\Theta}$ should be limited to $[-g_{\text{min}}, -g_{\text{max}}] \cup \{0\} \cup [g_{\text{min}}, g_{\text{max}}]$. To this end, the learnable conductance $\boldsymbol{\Theta}$ are projected element-wise to the printable range before performing the weighted-sum operation, i.e.

$$\Theta_{i,j} \leftarrow \begin{cases} 0, & \text{if } |\Theta_{i,j}| < g_{\text{min}}, \\ \Theta_{i,j}, & \text{if } |\Theta_{i,j}| \in [g_{\text{min}}, g_{\text{max}}], \\ \text{sign}(\Theta_{i,j}) \cdot g_{\text{max}}, & \text{if } |\Theta_{i,j}| > g_{\text{max}}, \end{cases}$$

as shown in figure 3 by the black curve. To be able to calculate useful gradients for backpropagation despite the projection mapping, the straight-through gradient estimator [32] is employed, see figure 3. In other words, the projection is ignored for the calculation of derivatives.

The printable range also constrains the component values in the nonlinear circuits, expressed as $[q_{\text{min}}^A, q_{\text{max}}^A]$ and $[q_{\text{min}}^N, q_{\text{max}}^N]$. Since these feasible (printable) design space areas are continuous, simple functions such as sigmoid or tanh can be used to map the learnable parameters into printable ranges.

Apart from the constraints on printable ranges, printing technologies also impose constraints in the form of printing error (variation) caused by the limited printing resolution. This constraint hinders manufacturing the exact designed values of the circuit components. To address this problem, a stochastic variable can be introduced onto the learnable parameters to express the uncertainty in the manufacturing process [10]. Additionally, the PE is susceptible to not only printing variation, but also environmental effects, which changes the printed component values gradually over time (aging). To consider also the aging problem in the design of printed neuromorphic circuits, the printed component values are modeled as a time-varying variables according to the stochastic aging model [22], rather

than fixed constants. Due to the employment of these approaches, the output of the pNN and the corresponding loss are no longer constant values but rather a probabilistic distribution w.r.t. printing variations and aging time. In order to still adopt the classical gradient-based optimization strategies to train the pNN, the expected loss w.r.t. printing variation and time is calculated and used to guide the training of the pNN. More details are described in section 3.

Previous studies have exclusively focused on the separate effects of print variation and aging on printed neuromorphic circuits. Nonetheless, a comprehensive study of the joint impact of these factors on printed neuromorphic circuits has yet to be conducted. This study seeks to address this research gap by including both factors jointly. Moreover, the sensing uncertainty (uncertainty over the inputs) is also considered in this work. Additionally, to analyze the combinatorial effect of these factors on the performance of pNNs, extensive experiments followed by an ablation study is conducted.

3. Methodology

Since high variations are more common in additive PE compared to conventional subtractive manufactured electronics, and moreover, since analog circuits are more sensitive to component variations compared to digital circuits, it is essential to account for factors affecting circuit output during the design of printed analog neuromorphic circuits. In this work, three principal effects are considered: printing variations due to limited printing resolution; environmental effects that cause aging of circuit components; and measurement uncertainties, arising from multiple noise sources. By considering and addressing these key factors from software level, the reliability and robustness of printed neuromorphic circuits can be substantially improved. Note that, this approach combat the aforementioned factors purely at the algorithmic level, which is orthogonal to the research in improving printing technologies or materials. Therefore, it can improve the circuit dependability even if the influences could not be controlled or reduced from the fabrication standpoint.

3.1. Printing variation

In the manufacturing of PE, the desired component values, like conductances, can generally not be printed exactly. This variation primarily arises from the constrained print resolution, which stems from the physical properties of the functional inks and limitations of the printing technology. The printing resolution is principally determined by, e.g. the volume of the smallest printable volume of the droplets [33]. Consequently, by assuming that, the printing variation is determined by the geometric variation of the printing shape which varies within one printing pixel, the printing variation is often modeled as a uniformly

distributed stochastic variable within the minimum resolution, as shown on the left in figure 4, i.e.

$$\Theta_{i,j}^{PV} \sim \mathcal{U} \left[(1 - \epsilon)\Theta_{i,j}^{\text{ideal}}, (1 + \epsilon)\Theta_{i,j}^{\text{ideal}} \right].$$

Here, the value for ϵ is selected based on the specific printing technology to accommodate printing variations. To facilitate the training process for the learnable parameter Θ^{ideal} , we utilized a random variable ϵ to independently parameterize and extract Θ^{ideal} by

$$\Theta^{PV} = \epsilon \odot \Theta^{\text{ideal}},$$

where Θ^{PV} models the manufactured conductance with printing variation and ϵ is a stochastic variable denoting the printing variation with each element in ϵ following a uniform distribution $\mathcal{U}[1 - \epsilon, 1 + \epsilon]$.

The impact of printing variation on weights within the resistor crossbar is rather intuitive: as the printed conductances deviate from ideal values, their corresponding weights will also deviate. However, this impact on nonlinear circuits is rather intricate, therefore, we visualize the impact of the variation in nonlinear circuits, as shown in figure 5. The middle and right figures exemplifies some varied characteristic curves with 10% printing variation (incorporating with aging effect).

3.2. Aging effect

Due to environmental influences and particularly thermal stress in the field, the thin-film printed devices exhibit run-time degradation through usage (aging) [34–36]. As a result, the conductances of printed resistors will change over time [13]. These effects lead to deviations from the intended design values in the neuromorphic circuit over time and may ultimately result in misclassifications.

In general, the aging process of printed resistors consists of two phases: a fast degradation phase followed by a more gradual decline [13, 22]. A stochastic aging model that describes the multiplicative change in initial conductance has been proposed in [22], i.e.

$$\Theta^{\text{aged}}(t) = \Theta_0 \odot A_{\omega}(t),$$

where $\Theta^{\text{aged}}(t)$ refers to the changing conductance values w.r.t. time, Θ_0 denotes the initial conductances directly after the manufacturing (i.e. Θ^{PV}), while $A_{\omega}(t)$ summarizes the stochastic aging degradation of the conductances with each element in $A_{\omega}(t)$ being

$$A_{\omega}(t) = \omega_1 \cdot e^{-\omega_2 \cdot t} - \omega_1 + 1, \quad \omega \sim p_{\omega}(\omega).$$

Here, $\omega = [\omega_1, \omega_2]$ is a stochastic variable with the probability density $p_{\omega}(\omega)$. The density $p_{\omega}(\omega)$ can be obtained by fitting a log-normal distribution to the measured ω from several aging experiments [22]. Figure 4 (middle) indicates some examples of stochastic aging decay $A_{\omega}(t)$ by sampling ω from $p_{\omega}(\omega)$. The left part of figure 5 visualizes multiple

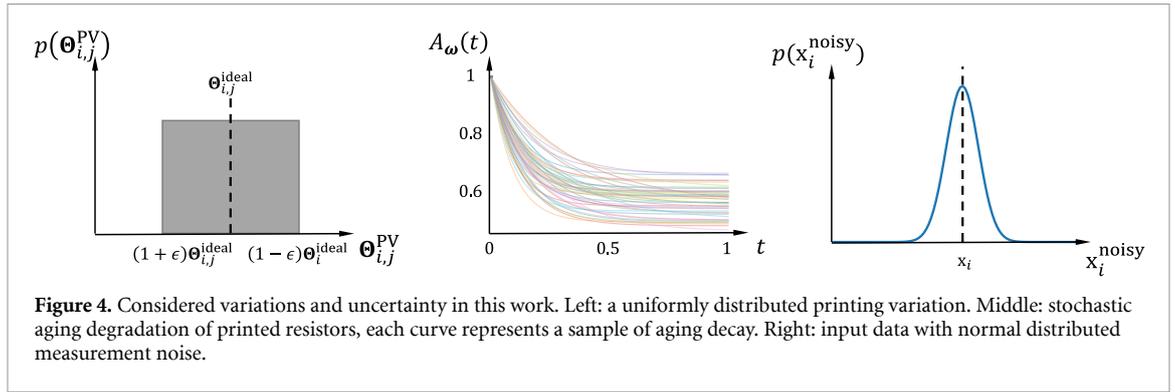


Figure 4. Considered variations and uncertainty in this work. Left: a uniformly distributed printing variation. Middle: stochastic aging degradation of printed resistors, each curve represents a sample of aging decay. Right: input data with normal distributed measurement noise.

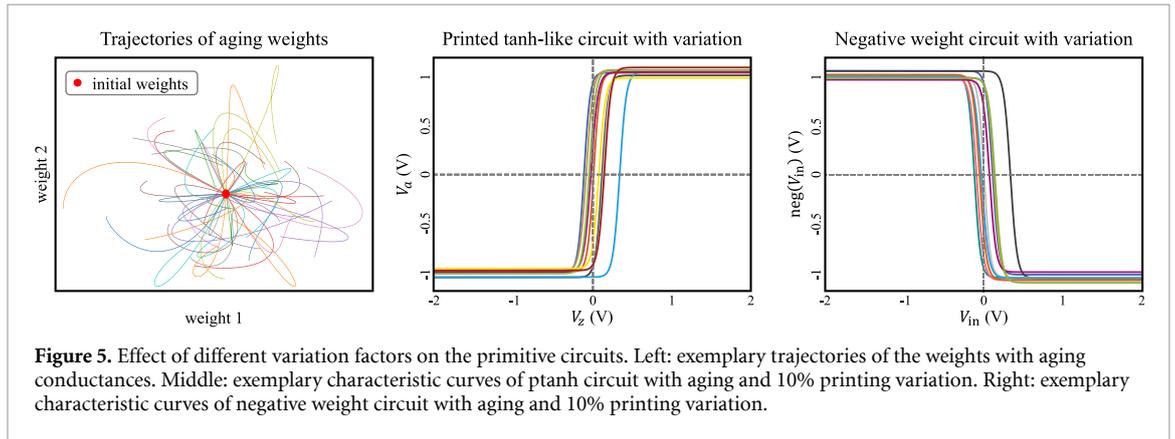


Figure 5. Effect of different variation factors on the primitive circuits. Left: exemplary trajectories of the weights with aging conductances. Middle: exemplary characteristic curves of ptanh circuit with aging and 10% printing variation. Right: exemplary characteristic curves of negative weight circuit with aging and 10% printing variation.

weight trajectories due to the stochastic aging of conductances in a resistor crossbar. Notably, as the crossbar functions as a voltage divider and each conductance ages following a stochastic curve, the resulted weights may experience aging along distinct trajectories.

3.3. Sensing uncertainty

Measurement uncertainty is a quantitative assessment that provides an estimate of the potential range of the true value of a physical quantity with a specific level of confidence [37]. The uncertainty arises due to various processes during the measurement, including the intrinsic error of the measuring instrument, the coupling between the measuring instrument and the system being measured, changes in measurement conditions, and the imperfections in the calibration procedure. Therefore, it is imperative to respect the measurement uncertainty during the design of a robust and reliable printed neuromorphic circuit. Moreover, since the pNN works directly with sensors in analog domain instead of digital, it is more sensitive to sensing uncertainties.

As measurement uncertainty is the cumulative outcome of various stochastic variables arising from the processes mentioned above, it is often modeled by a Gaussian distribution in the signal processing community, in accordance with the central limit theorem [38] and the principle of maximum entropy [39]. In this work, we model the noisy input signals by multiplying the original input

\mathbf{x} by a Gaussian distributed random variable ν . Consequently, higher inputs exhibit more variation. Formally, this is expressed as

$$\mathbf{x}^{\text{noisy}} = \nu \odot \mathbf{x}, \quad \nu \sim \mathcal{N}(1, \sigma),$$

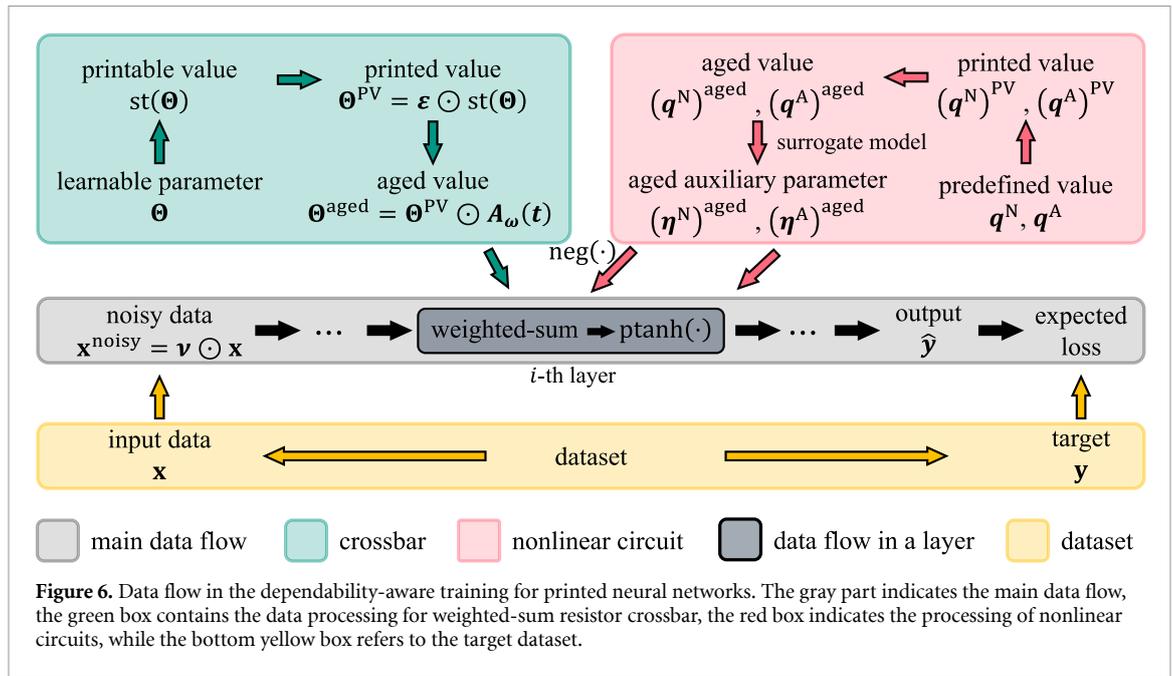
where σ refers to the uncertainty of the measurement.

3.4. Dependability-aware training of pNNs

To include the aforementioned influence into the design of dependable printed neuromorphic circuits, we propose a framework that is capable of incorporating all relevant factors into the training of pNNs, as illustrated in figure 6.

For each resistor crossbar, the learnable conductance is processed by a straight-through estimator to maintain its printability. Once converted to a printable value, the stochastic variable ϵ is element-wise multiplied to simulate printing variations for each conductance. Subsequently, the aging decay $A_\omega(t)$ is multiplied to reflect the aging behaviors of conductances already affected by printing variations. Finally, the resulting conductances are transformed to the corresponding weights in the weighted-sum operation.

Regarding the nonlinear circuits, we do not consider their parameters learnable but rather fixed to certain design values. For details on how these parameters can be learned, see [31]. Nevertheless, we still account for their printing variations and aging behaviors during training. The nonlinear circuits comprise two types of components, namely resistors,



characterized by their conductances, and transistors, characterized by their geometric features (width W and length L). We consider both printed neural networks and aging effects for the printed resistors, but for the transistors, we only consider printed neural networks on W and L , as their aging behavior is still not sufficiently studied. After the aged parameters for the nonlinear circuits, i.e. $(q^A)^{\text{aged}}$ and $(q^N)^{\text{aged}}$, have been calculated, they can be mapped to the auxiliary parameters $(\eta^A)^{\text{aged}}$ and $(\eta^N)^{\text{aged}}$ via differentiable surrogate nonlinear circuit models. These auxiliary parameters can then be utilized to construct negative weight functions and tanh-like transformation functions, which will be integrated into pNNs for weighted-sum and activation functions, respectively.

As machine learning is generally a data-driven optimization process, training machine learning models is tailored to certain target datasets. A dataset typically comprises two components, i.e. the measurement \mathbf{x} , which serves as the input to the machine learning model, and the corresponding ground truth \mathbf{y} , which is considered the target output (e.g. the class in a classification task) from the model. The training process involves finding a set of parameters (usually, weights) that allows the model output to approximate the desired output with a given input signal. It should be noted that, despite the inherent measurement error present in the input signal \mathbf{x} in the dataset, applying extra noise to the input data is still highly desirable to enhance the tolerance to input noise [40].

With consideration of these stochastic processes, the output $\hat{\mathbf{y}}$ of pNNs, and consequently, the loss $L(\hat{\mathbf{y}}, \mathbf{y})$ is no longer a deterministic value, but rather a stochastic distribution with respect to ν, ε, ω . Therefore, the loss of the pNN can be denoted by $L(\mathbf{x}, \mathbf{y}, \Theta, \nu, \varepsilon, \omega, t)$. In general, gradient-based

numerical optimizers for neural networks require the loss to be expressed as a deterministic scalar value instead of a function or distribution. To solve this problem, we adopt the expectation to assess the value of the loss and obtain the **dependability-aware training** objective function, i.e.

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \mathbf{y}, \Theta) &= \mathbb{E}_{\nu} \left\{ \mathbb{E}_{\varepsilon} \left\{ \mathbb{E}_{\omega} \left\{ \int_0^1 L(\mathbf{x}, \mathbf{y}, \Theta, \nu, \varepsilon, \omega, t) dt \right\} \right\} \right\} \\ &= \iiint_{\nu, \varepsilon, \omega, t} L(\mathbf{x}, \mathbf{y}, \Theta, \nu, \varepsilon, \omega, t) p_t(t) p_{\omega}(\omega) p_{\varepsilon}(\varepsilon) p_{\nu}(\nu) dt d\omega d\varepsilon d\nu. \end{aligned} \quad (3)$$

For simplify, the lifetime was normalized to $t \in [0, 1]$. Additionally, the integral over the lifetime was expressed as a mathematically equivalent expected value with respect to a uniform distribution $p_t(t) = \mathcal{U}[0, 1]$ to allow for a consistent treatment with the random variables ν, ε, ω in the following. Consequently, ν, ε, ω , and t can be summarized by one stochastic variable $\gamma := [\nu, \varepsilon, \omega, t]$ with density $p_{\gamma}(\gamma) = p_{\nu}(\nu) \cdot p_{\varepsilon}(\varepsilon) \cdot p_{\omega}(\omega) \cdot p_t(t)$. Thus, equation (3) can be written as

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \Theta) = \int_{\gamma} L(\mathbf{x}, \mathbf{y}, \Theta, \gamma) p_{\gamma}(\gamma) d\gamma. \quad (4)$$

Unfortunately, due to the complexity of $L(\cdot)$, usually no analytical solution for equation (4) can be found. We thus employ Monte Carlo integration [41] to obtain an approximation of equation (4). According to the law of large numbers [42], the expected value

of a function can be estimated by the average of the results obtained from multiple samples, i.e.

$$\begin{aligned}\mathbb{E}_{\gamma}\{L(\mathbf{x}, \mathbf{y}, \Theta, \gamma)\} &= \int_{\gamma} L(\mathbf{x}, \mathbf{y}, \Theta, \gamma) p_{\gamma}(\gamma) d\gamma \\ &\approx \frac{1}{N_{\gamma}} \sum_{\gamma'} L(\mathbf{x}, \mathbf{y}, \Theta, \gamma'),\end{aligned}\quad (5)$$

where $\gamma' = [\nu', \varepsilon', \omega', t']$ describes a set of N_{γ} samples drawn from the distribution $p_{\gamma}(\gamma)$. With this approach, each time the loss need to be computed, equation (5) can be employed to obtain an estimate by drawing N_{γ} samples from $p_{\gamma}(\gamma)$. Similar to the objective, also gradients for equation (4) can be obtained via Monte Carlo gradient estimation via

$$\nabla_{\Theta} \mathcal{L}(\mathbf{x}, \mathbf{y}, \Theta) \approx \frac{1}{N_{\gamma}} \sum_{\gamma'} \nabla_{\Theta} L(\mathbf{x}, \mathbf{y}, \Theta, \gamma').$$

These estimated gradients can then be used by common gradient-based optimization algorithms such as SGD [43] or Adam [44].

4. Experiments

To evaluate the effectiveness of the dependability-aware training of pNNs, we implemented the proposed training approach with PyTorch [45] and conduct experiments on the 13 benchmark datasets³ which were also used in the comparable works, such as [22, 31]. The benchmark dataset exhibits a complexity and scenario that align with the target application domains of PE. Specifically, datasets with a modest number of inputs and outputs (generally fewer than ten) are more appropriate, due to the large feature size and low integration density of PE.

Due to the large number of circuits required to obtain statistical results w.r.t. to the stochastic variations, evaluating the proposed dependability-aware training by printing hardware circuits poses a significant challenge and high cost. Therefore, we verify the proposed algorithm at simulation level based on the well-established printed process design kit [23]. The functionality of the real printed neuromorphic hardware has been experimentally validated in [10, 46].

4.1. Experiment setup

For the experiment, we use a consistent topology (*#inputs-3-#outputs*) for all pNNs on each dataset. For the nonlinear circuits, predefined values $\eta^A = \eta^N = [0, 1, 0, 12.3]$ for tanh-like activation circuits and negative weight circuits are employed. Regarding the training, we employ full-batch training with the

Adam [44] optimizer (in default parameterization) to update parameters in pNNs. In each training epoch, we sampled $N_{\gamma} = 5000$ stochastic variables γ' for Monte Carlo integration. To prevent overfitting, we calculated the loss on validation set for early-stopping [47] after each parameter update. We start with an initial learning rate of 0.1 and halve it after a patience (updates without improvement) of 100-epochs on the validation set. Additionally, the training process is stopped, when the learning rate decreases below 10^{-4} . We choose $\epsilon = 0.1$ to reflect the printing variation, as typical printing resolutions range from 20 μm to 100 μm [48], whereas the component feature sizes in printed neuromorphic circuits are on the order of 1 μm [10]. Therefore, $\pm 10\%$ can be seen as a reasonable estimate. Moreover, we take $\sigma = 0.1$ for ν to simulate sensing uncertainty of the inputs \mathbf{x} . As loss function, we adopted the same function provided in [10], which is a hardware-aware loss function designed for pNNs.

4.2. Ablation study

In this work, multiple factors that could potentially impact the results are considered. To investigate the effects of these factors independently and jointly, we conduct an ablation study. Specifically, we conducted experiments on all possible combinations of the three factors (eight combinations in total) to assess their combinatorial effects. To facilitate the identification of individual experiments, they are numbered from **Exp. 1** to **Exp. 8**. Furthermore, the terms ‘aging behavior,’ ‘printing variation,’ and ‘sensing uncertainty’ are abbreviated to ‘AG,’ ‘PV,’ and ‘SU,’ respectively (see table 1 for details). The abbreviation with an additional over line indicates that, the experiment is unaware of the corresponding factor, e.g. AG refers to AG-unaware training. Hence, the specific pNNs are only trained with consideration of the specific factors, while for testing, all effects, i.e. AG, PV, and SU are included.

4.3. Evaluation

After training, we choose pNNs based on the best validation loss, as it would be the one selected for fabrication. We evaluate the resulted pNNs on the test set base on $N_{\gamma} = 5000$ samples. As an evaluation metric, we adopt the measuring-aware accuracy (MAA) [10], which is hardware-related accuracy that considers the threshold for reliably measuring output voltages for classification.

In this work, **dependability** is conceptualized to reflect two aspects of the performance of the pNNs, i.e. **accuracy** and **robustness**. Here, the accuracy and robustness are indicated by the **mean MAA** and the **standard deviation (std) of MAA** w.r.t. the stochastic variable γ . Thus, the metrics are calculated on each dataset and reported in table 1. As a summary, the averaged values of all the dataset for each experiment are also reported.

³ Datasets: ¹Acute Inflammation, ²Balance Scale, ³Breast Cancer Wisconsin, ⁴Cardiotocography, ⁵Energy Efficiency (γ_1), ⁶Energy Efficiency (γ_2), ⁷Iris, ⁸Mammographic Mass, ⁹Pendigits, ¹⁰Seeds, ¹¹Tic-Tac-Toe Endgame, ¹²Vertebral Column (2 cl.), ¹³Vertebral Column (3 cl.).

Table 1. The mean and standard deviation of measuring-aware accuracy (MAA) w.r.t. stochastic variable γ on each dataset for each experiment.

Notation	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8
Awareness	$\overline{AG}, \overline{PV}, \overline{SU}$							
Dataset	1	2	3	4	5	6	7	8
	0.92 ± 0.10	0.92 ± 0.08	0.98 ± 0.06	0.98 ± 0.06	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.01	1.00 ± 0.00
	0.63 ± 0.27	0.63 ± 0.28	0.75 ± 0.17	0.76 ± 0.18	0.81 ± 0.03	0.80 ± 0.03	0.81 ± 0.03	0.81 ± 0.02
	0.88 ± 0.12	0.89 ± 0.13	0.97 ± 0.01	0.96 ± 0.01	0.97 ± 0.00	0.97 ± 0.00	0.96 ± 0.01	0.96 ± 0.00
	0.75 ± 0.12	0.70 ± 0.19	0.78 ± 0.16	0.81 ± 0.09	0.84 ± 0.03	0.84 ± 0.03	0.85 ± 0.05	0.85 ± 0.03
	0.69 ± 0.22	0.64 ± 0.25	0.83 ± 0.13	0.84 ± 0.11	0.91 ± 0.04	0.90 ± 0.06	0.90 ± 0.05	0.92 ± 0.03
	0.74 ± 0.13	0.75 ± 0.09	0.82 ± 0.09	0.83 ± 0.06	0.84 ± 0.05	0.85 ± 0.04	0.85 ± 0.04	0.86 ± 0.03
	0.84 ± 0.08	0.85 ± 0.11	0.87 ± 0.11	0.87 ± 0.11	0.92 ± 0.05	0.94 ± 0.03	0.93 ± 0.06	0.92 ± 0.05
	0.54 ± 0.18	0.56 ± 0.14	0.66 ± 0.14	0.68 ± 0.13	0.72 ± 0.10	0.74 ± 0.09	0.75 ± 0.06	0.75 ± 0.06
	0.10 ± 0.10	0.10 ± 0.10	0.37 ± 0.10	0.51 ± 0.07	0.44 ± 0.06	0.57 ± 0.05	0.48 ± 0.05	0.54 ± 0.05
	0.76 ± 0.13	0.74 ± 0.11	0.78 ± 0.12	0.82 ± 0.07	0.82 ± 0.04	0.86 ± 0.03	0.85 ± 0.05	0.86 ± 0.03
	0.59 ± 0.19	0.66 ± 0.15	0.80 ± 0.08	0.75 ± 0.10	0.76 ± 0.06	0.79 ± 0.09	0.73 ± 0.09	0.81 ± 0.07
	0.57 ± 0.13	0.65 ± 0.09	0.64 ± 0.10	0.76 ± 0.07	0.61 ± 0.10	0.73 ± 0.07	0.61 ± 0.07	0.77 ± 0.06
	0.50 ± 0.15	0.60 ± 0.12	0.59 ± 0.13	0.71 ± 0.08	0.63 ± 0.11	0.76 ± 0.06	0.63 ± 0.09	0.75 ± 0.07
Average	0.66 ± 0.15	0.67 ± 0.14	0.76 ± 0.11	0.79 ± 0.09	0.79 ± 0.05	0.83 ± 0.04	0.80 ± 0.05	0.83 ± 0.04

Table 2. Independent effects of aging (AG), printing variation (PV), and sensing uncertainty (SU) in the dependability-aware training of pNNs.

Awareness	Averaged experiments	Averaged MAA	Improvement	
			Accuracy (mean MAA)	Robustness (std MAA)
AG-aware	Exp. 5, 6, 7, 8	0.811 ± 0.046	13.03%	61.58%
AG-unaware	Exp. 1, 2, 3, 4	0.718 ± 0.120		
PV-aware	Exp. 3, 4, 7, 8	0.794 ± 0.071	7.99%	26.46%
PV-unaware	Exp. 1, 2, 5, 6	0.735 ± 0.096		
SU-aware	Exp. 2, 4, 6, 8	0.779 ± 0.077	3.89%	13.26%
SU-unaware	Exp. 1, 3, 5, 7	0.750 ± 0.089		

Through the comparison of Exp. 8 and Exp. 1, we conclude that, with consideration of all three factors in the training process, a substantial 27% improvement in accuracy and a 74% improvement in robustness.

4.4. Analysis

Upon evaluating the performance of pNNs for each experiment on the test set, we conducted a quantitative analysis to determine the independent influence of each factor on the final results, as well as the relationships between them.

4.4.1. Independent analysis

To analyze the impact of a certain factor independently of other factors, we divided the eight experiments into two groups (e.g. experiments with, and without AG-aware training) and average the performance respectively. Table 2 summarizes the analysis of each factor.

It is evident from table 2, that AG-aware training provides the most significant improvement in both accuracy and robustness, namely 13.03% and 61.58%. This is followed by PV-aware training, which achieves an improvement of 7.99% in average MAA and 26.46% in robustness. Lastly,

the SU-aware training approach delivers the lowest accuracy improvement of 3.89% and lowest robustness improvement of 13.26%.

Based on the given comparison, we conclude that the three stochastic factors exhibit different degrees of influence on the pNNs: as AG and PV lead to changes in every conductance (thus weight) of the pNNs, whereas SU only explicitly affects the first weighted-sum operation (multiplicatively), which results in the weaker impact on the performance of the pNNs. On the other hand, for the comparison of AG and PV, we hypothesize that AG has a more substantial impact on the conductance than PV (based on the input noise, variations, and AG behavior assumed in this experiment). Consequently, AG-aware training yields greater improvements compared to PV-aware training.

4.4.2. Joint analysis

Despite the independent analysis of the impact of each factor on pNNs, their actual effects are not entirely independent of each other. To assess their relationships, we evaluate the improvement provided by each factor in different settings (see figure 7 for accuracy and figure 8 for robustness).

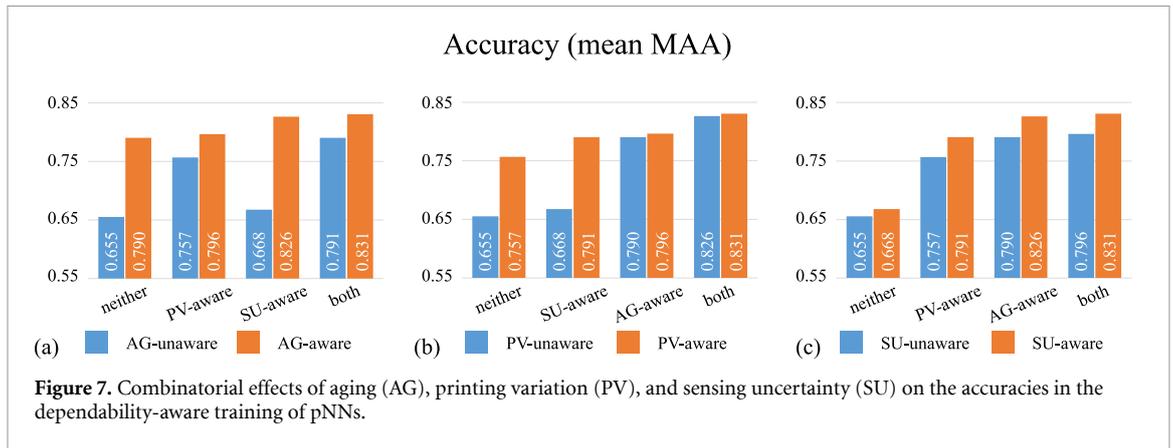


Figure 7. Combinatorial effects of aging (AG), printing variation (PV), and sensing uncertainty (SU) on the accuracies in the dependability-aware training of pNNs.

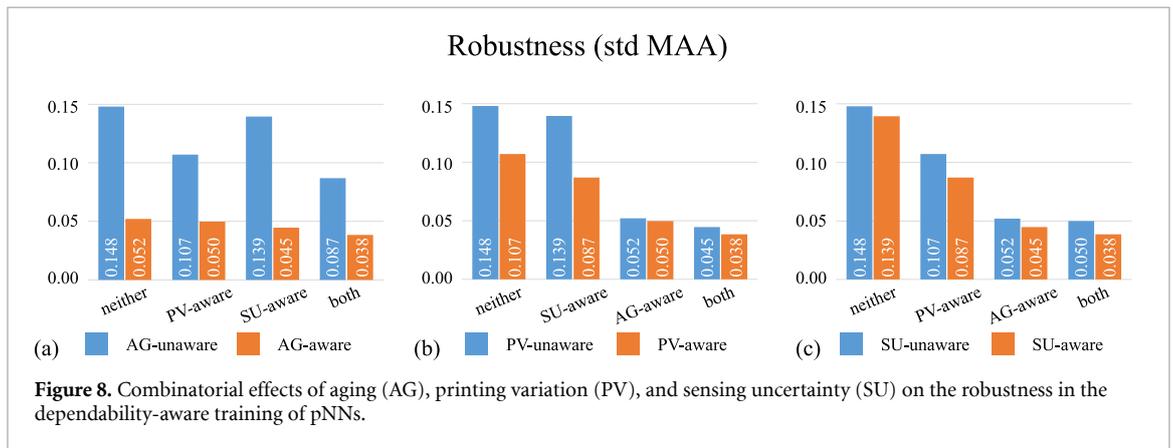


Figure 8. Combinatorial effects of aging (AG), printing variation (PV), and sensing uncertainty (SU) on the robustness in the dependability-aware training of pNNs.

Figure 7(a) demonstrates that AG-aware training can substantially improve the accuracy of pNNs from standard training (unaware of neither AG nor PV nor SU). The same is true when only SU-aware training is used. Conversely, it offers less improvement for pNNs that have already trained with consideration of PVs. We hypothesize that PV and AG effect may have similar effects in training pNNs, which renders the additional improvement of AG-aware training over PV-aware training insignificant. This hypothesis is also supported by figure 7(b).

In figure 7(b), it can be seen that PV-aware training can significantly increase the accuracy of pNNs from standard training and SU-aware training. However, PV-aware training offers only less improvement when AG-aware training is already employed. This suggests that the AG effects not only have a similar influence as PV, namely perturbing the resulting weights, but also exert a stronger impact than PV. Therefore, additional PV-aware training has little benefit when AG-aware training is already utilized. In contrast to PV- and AG-aware training, SU-aware training offers around 2%–3% improvement in all cases, as shown in figure 7(c). This indicates that the impact of SU may be orthogonal to that of PV or AG.

Regarding robustness, similar effects are observed. From figure 8, it is evident that both AG- and PV-aware training can substantially enhance the robustness of pNNs. However, as they might have

similar mechanisms of influence on pNNs, their combined effect exhibits some overlapping. Moreover, since the impact of AG is more significant than that of PV, the additional PV-aware training in conjunction with AG-aware training does not bring significant benefits. Orthogonal to them, SU-aware training consistently provides stable improvement in robustness.

4.5. Discussion

In this section, we perform experiments to confirm the effectiveness of the dependability-aware training of pNNs. Our results demonstrate that the proposed method can enhance the accuracy and robustness of pNNs by 27% and 74%, respectively. Among all effects considered in training, AG-aware training yields the most significant improvement. While PV-aware training also contributes significantly, our ablation study reveals that PV and AG effect may have similar potential mechanisms on the pNNs, suggesting that the contribution of AG-aware training may partly cover that of PV-aware training.

Notably, SU-aware training consistently delivers improvement in accuracy and robustness across all experiments. This suggests that, SU might have a distinct mode of effect on pNNs compared to the other two factors. Even though the improvement from SU-aware training is slightly lower than that from PV- and AG-aware trainings, it is possibly due to the choice of σ value. We suspect that, the effect

of SU-aware training may change with different σ values. Nevertheless, due to the possibly orthogonal effects to the other two factors, it is meaningful to consider SU to improve the dependability of pNNs.

Although the dependability-aware training is conducted at fully algorithmic level, the outcomes from the ablation study may also indicate the impact of various factors on network performance, and thus, guide the development of the hardware technologies. For instance, AG-aware training demonstrates a substantial enhancement in network performance, implying that, the AG process may exert a more pronounced effect on circuits. Consequently, in fabrication, it is suggestible to prioritize the efforts both in the materials as well as in the process to reduce the impact of AG, including the adoption of passivation techniques or superior materials that display lower AGvariation.

5. Conclusion

PE has gained increasing interest in emerging domains, such as the IoT and wearable devices, as it offers several advantages over traditional silicon-based electronics. These advantages include flexibility, bio-compatibility, lightweight, high customizability, and ultra-low cost. However, PE also suffers from some limitations, such as large feature sizes, large latencies, low integration density, high variation, and low device count. To this regard, printed analog neuromorphic circuits become attractive, as they leverage the benefits of PE while evading its drawbacks. Moreover, by interconnecting multiple simple primitive subcircuits, printed neuromorphic circuits are capable of implementing even complex functionalities.

In this work, we focus on investigating the dependability of printed analog neuromorphic circuits, with a particular focus on three critical factors that influence their performance, namely, PV, AG effect, and SU. To respect these factors into the design of printed neuromorphic circuits, stochastic models of these factors are established. Subsequently, to account these factors into the training of pNNs, a modified objective function is developed. Experimental results indicate that incorporating the modeled stochastic variations in the training process can improve the expected accuracy and robustness. Additionally, the ablation study is conducted to reveal the contribution of each influencing factor independently and jointly. The ablation study suggests that AG-aware training contributes the most, and SU-aware training contributes the least to the dependability of pNNs. However, it is worth noting that, the combination of AG-aware and PV-aware training does not greatly improve the overall performance of pNNs, as their improvements may overlap. In contrast, the effect of SU-aware training appears to be independent of the other two factors.

Although the improvement from SU-aware training is relatively smaller compared to the other two factors, it is still worth to be considered for improving the dependability of pNNs, because it may represent another pattern of behavior that affects the performance of pNNs.

In future work, effort will be dedicated to studying the influence of further factors in the printing process, including printing speed, printing throughput, heat treatment, post-processing, etc. By developing mathematical models for these procedures, the influence will be explicitly incorporated into the design process of printed neuromorphic circuits, and thus enabling the improvement of the circuit dependability with respect to those factors.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgment

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

ORCID iD

Haibin Zhao  <https://orcid.org/0000-0001-7018-1159>

References

- [1] Khodadadi F *et al* 2016 Internet of things: an overview *Internet of Things* vol 80 (Elsevier) p 3
- [2] Seneviratne S, Hu Y, Nguyen T, Lan G, Khalifa S, Thilakarathna K, Hassan M and Seneviratne A 2017 *IEEE Commun. Surv. Tutor.* **19** 2573
- [3] Schaefer D, Cheung W M 2018 *Proc. CIRP* **72** 1022
- [4] Asulin M, Michael I, Shapira A and Dvir T 2021 *Adv. Sci.* **8** 2004205
- [5] Kaidarova A *et al* 2019 *NPJ Flex. Electron.* **3** 15
- [6] Labiano I I and Alomainy A 2021 *Flex. Print. Electron.* **6** 025010
- [7] Chang J *et al* 2012 Challenges of printed electronics on flexible substrates *55th Int. Midwest Symp. on Circuits and Systems* (IEEE) p 582
- [8] Reddy A, Narakathu B B, Atashbar M Z, Rebros M, Rebrosova E and Joyce M K 2011 *Proc. Eng.* **25** 120
- [9] Zhao H *et al* 2022 Printed electrodermal activity sensor with optimized filter for stress detection *Proc. 2022 ACM Int. Symp. on Wearable Computers* (ACM) p 112
- [10] Weller D D, Hefenbrock M, Beigl M, Aghassi-Hagmann J and Tahoori M B 2021 *Sci. Rep.* **11** 1
- [11] Kirkup L and Frenkel R B 2010 *Introduction to Uncertainty in Measurement (Reliability Engineering)* (New York: Springer) (https://doi.org/10.1007/978-0-387-46328-5_1)
- [12] Khan S, Lorenzelli L and Dahiya R S 2015 *IEEE Sens. J.* **15** 3164
- [13] Hamasha M M, Dhakal T, Alzoubi K, Albahri S, Qasaimeh A, Lu S and Westgate C R 2012 *IEEE J. Disp. Technol.* **8** 385

- [14] Chang J S, Facchetti A F and Reuss R 2017 *IEEE J. Emerg. Sel. Top. Circuits Syst.* **7** 7
- [15] Chung S, Kim S O, Kwon S-K, Lee C and Hong Y 2011 *IEEE Electron Device Lett.* **32** 1134
- [16] Shao F, Wan Q 2019 *J. Phys. D: Appl. Phys.* **52** 143002
- [17] Cui Z 2016 *Printed Electronics: Materials, Technologies and Applications* (New York: Wiley) (<https://doi.org/10.1002/9781118920954>)
- [18] Conti S et al 2020 *Nat. Commun.* **11** 3566
- [19] Huber B Popp P B, Kaiser M, Ruediger A and Schindler C 2017 *Appl. Phys. Lett.* **110** 143503
- [20] Weller D, Cadilha Marques G, Aghassi-Hagmann J and Tahoori M B 2018 *IEEE Electron Device Lett.* **39** 831
- [21] Kondo M et al 2018 Design of ultraflexible organic differential amplifier circuits for wearable sensor technologies *IEEE Int. Conf. on Microelectronic Test Structures (ICMTS)* p 79
- [22] Zhao H et al 2022 Aging-aware training for printed neuromorphic circuits *Proc. 41st IEEE/ACM Int. Conf. on Computer-Aided Design (ACM)* (<https://doi.org/10.1145/3508352.3549411>)
- [23] Rasheed F, Hefenbrock M, Beigl M, Tahoori M B and Aghassi-Hagmann J 2018 *IEEE Trans. Electron Devices* **66** 146
- [24] Yu Z, Abdulghani A M, Zahid A, Heidari H, Imran M A and Abbasi Q H 2020 *IEEE Access* **8** 67085
- [25] Schuman C D et al 2017 A survey of neuromorphic computing and neural networks in hardware (arXiv: [1705.06963](https://arxiv.org/abs/1705.06963))
- [26] Sebastian A, Le Gallo M, Khaddam-Aljameh R and Eleftheriou E 2020 *Nat. Nanotechnol.* **15** 529
- [27] Eshraghian J K et al 2019 Analog weights in ReRAM DNN accelerators *Int. Conf. on Artificial Intelligence Circuits and Systems (AICAS)* (IEEE) (<https://doi.org/10.1109/AICAS.2019.8771550>)
- [28] Weller D D et al 2020 Programmable neuromorphic circuit based on printed electrolyte-gated transistors *25th Asia and South Pacific Design Automation Conf. (ASP-DAC)* p 446
- [29] Nawrocki R A, Voyles R M and Shaheen S E 2014 *Trans. Electron Devices* **61** 3513
- [30] Ansari M, Fayyazi A, Banagozar A, Maleki M A, Kamal M, Afzali-Kusha A and Pedram M 2017 *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **37** 1602
- [31] Zhao H et al 2023 Highly-bespoke robust printed neuromorphic circuits *Design, Automation and Test in Europe (DATE)* (IEEE) (<https://doi.org/10.5445/IR/1000156490>)
- [32] Bengio Y et al 2013 Estimating or propagating gradients through stochastic neurons for conditional computation (arXiv:1308.3432)
- [33] Onses M S, Sutanto E, Ferreira P M, Alleyne A G and Rogers J A 2015 *Small* **11** 4237
- [34] Bielecka U, Lutsyk P, Janus K, Sworakowski J and Bartkowiak W 2011 *Org. Electron.* **12** 1768
- [35] Fraboni B, Cosseddu P, Wang Y Q, Schulze R K, Di Z F, Cavallini A, Nastasi M and Bonfiglio A 2011 *Org. Electron.* **12** 1552
- [36] Lu K, Yao R, Wang Y, Ning H, Guo D, Liu X, Tao R, Xu M, Wang L and Peng J 2019 *J. Mater. Sci.* **54** 14778
- [37] Dieck R H 2017 *Measurement Uncertainty: Methods and Applications* 5th edn (ISA)
- [38] Montgomery D C et al 2010 *Applied Statistics and Probability for Engineers* (New York: Wiley)
- [39] Jaynes E T 1957 *Phys. Rev.* **106** 620
- [40] Taylor L et al 2018 Improving deep learning with generic data augmentation *Symp. Series on Computational Intelligence (SSCI)* (IEEE) p 1542
- [41] Caflisch R E 1998 *Acta Numer.* **7** 1–49
- [42] Dekking F M et al 2005 *A Modern Introduction to Probability and Statistics: Understanding Why and How* vol 488 (Berlin: Springer) (<https://doi.org/10.1007/1-84628-168-7>)
- [43] Amari S 1993 *Neurocomputing* **5** 185–96
- [44] Kingma D P et al 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)
- [45] Paszke A et al 2019 Pytorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* **32** (Curran Associates, Inc.) p 8024
- [46] Singaraju S A, Weller D D, Gspann T S, Aghassi-Hagmann J and Tahoori M B 2022 *Sensors* **22** 4000
- [47] Yao Y, Rosasco L and Caponnetto A 2007 *Constr. Approx.* **26** 289
- [48] Khan S, Lorenzelli L and Dahiya R S 2014 *IEEE Sens. J.* **15** 3164