ADAPT-pNC: Mitigating Device Variability and Sensor Noise in Printed Neuromorphic Circuits with SO Adaptive Learnable Filters

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

***Abstract*—The rise of the Internet of Things demands flexible, biocompatible, and cost-effective devices. Printed electronics provide a solution through low-cost and on-demand additive manufacturing on flexible substrates, making them ideal for IoT applications. However, variations in additive manufacturing processes pose challenges for reliable circuit fabrication. Adapting neuromorphic computing to printed electronics could address these issues. Printed neuromorphic circuits offer robust computational capabilities for near-sensor processing in IoT. One limitation of existing printed neuromorphic circuits is their inability to process temporal sensory inputs. To address this, integrating temporal components in printed neuromorphic circuit architectures enables the effective processing of time-series sensory data.**

Printed neuromorphic circuits face challenges from manufacturing variations such as ink dispersion, sensor noise, and temporal fluctuations, especially when processing temporal data and using time-dependent components like capacitors. To mitigate these challenges, we propose robustness-aware temporal processing neuromorphic circuits with low-pass second-order learnable filters (SO-LF). This approach integrates variation awareness by considering the variation potential of component values during training and using data augmentation to enhance adaptability against physical and sensor data variations. Simulations on 15 benchmark time-series datasets show our circuit effectively handles noisy temporal information under 10% process variations, achieving an average accuracy and power improvement of $\approx 24.7\%$ and $\approx 91\%$ respectively compared to models lacking variation with $\approx 1.9\times$ more devices.

I. Introduction

The rapid expansion of the Internet of Things (IoT) [1] has created a growing demand for next-generation smart electronics, including smart packaging [2], smart labels [3], smart band-aids [4], or other disposable electronics [5]–[8] (as shown in Fig. 1) for consumer products. These applications require technologies that are lightweight, flexible, and cost-efficient. Despite continuous improvements in power consumption and integration density, silicon-based electronics, which depend on lithography processes, face significant challenges in meeting these requirements. The main obstacles include bulky substrates and the necessity for complex manufacturing equipment, which limit the adaptability and cost-effectiveness needed for IoT edge devices [3, 4].

In this context, printed electronics (PE) emerge as a highly adaptable and cost-effective solution. The key advantage of PE is its ability to offer bespoke, application-specific customization, regardless of production volume. This flexibility

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

is primarily due to the low-cost nature of additive printing processes, which allows deposition of functional materials, facilitating desired properties, such as softness [9], non-toxicity [10], and bio-degradability [11] and thus stand in contrast to the expensive and complex silicon-based lithography fabrication methods. PE enables the production of customized electronic devices efficiently and economically, making it an ideal choice for the diverse and evolving demands of IoT [1] applications[1]. While PE offer significant advantages in terms of cost and flexibility, they come with their inherent challenges of larger feature sizes, lower integration density, and reduced performance compared to silicon-based devices. To mitigate these issues, printed analog neuromorphic circuits (pNCs) have emerged as a promising computing solution. These circuits operate directly in the analog domain, eliminating the need for the costly analog-to-digital converters (ADCs) and thus reducing device counts and costs. Neuromorphic computing has shown robust computational capabilities through simple operational circuit primitives [12]. The fundamental concept involves performing computational operations, such as weighted-sums and nonlinear activation functions typical of artificial neural networks (ANNs), using specific analog circuit designs like resistor crossbars and nonlinear transfer circuits. This approach leverages the powerful computational potential of neuromorphic computing, offering a viable solution to the limitations inherent in PE.

Existing research on pNCs has largely aimed to bridge the gap between realistic printed circuit design and ANN models. From a circuit design perspective, studies [13] and [14] have enabled various activation functions. At the algorithmic level, research by [12] has extended the learnable parameters from weights to nonlinear functions, [15] has incorporated manufacturability and printing variations into the circuit optimization process, while [8, 16] has assumed to operate with circuit's time-dependent input signals to store temporal-data informa-

♣Authors contributed equally to this work.

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

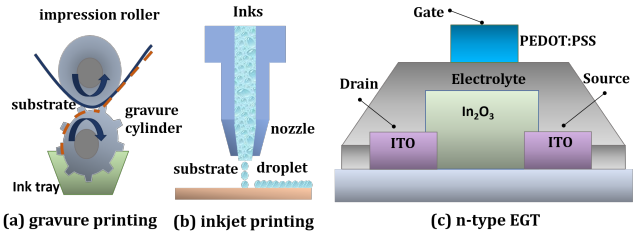

Fig. 2. Schematic of typical printing technologies: (a) gravure printing (b) inkjet printing (c) Schematic of a printed n-type electrolyte gated transistor (n-EGT).

tion. Additionally, the introduction of novel functional materials and devices, such as memristors [17] holds the potential to further expand the capabilities of pNCs.

Although significant efforts have been made to implement various pNCs, very few studies [8] have yet been reported on addressing the variation in sensory temporal input data [18] as well as in the temporal processing units. To address this limitation, this work introduces a robust and variation-aware printed temporal processing block (pTPB) that, when combined with pNCs, forms adaptive temporal printed neuromorphic circuits (ADAPT-pNCs). In addition, we address variation in sensory input signals using data augmentation techniques to enhance the model's robustness. In short, the contributions of this work are as follows:

- We propose a variation-aware second-order pTPB, for the first time, enabling the circuits to process time-series sensory inputs recognizing from all previous time steps.
- We employ data augmentation techniques including frequency domain augmentation, random cropping, jittering, time warping and magnitude scaling, to mitigate noisy inputs and enhance the reliability in handling real-world data.

The rest of this paper is organized as follows: Sec. II introduces the concept of PE, pNCs, recurrent neural networks (RNNs) and printed temporal processing neuromorphic circuits (pTPNCs). Sec. III motivates this research, describes the modeling of variation-aware second order pTPBs and ADAPT-pNCs, data augmentation techniques as well as propose the corresponding training objective. Subsequently, Sec. IV and Sec. IV-C validates the proposed method and discusses the experiment results. Finally, Sec. V summarizes this work.

II. Preliminaries

A. Printed Electronics

PE is a rapidly evolving sustainable technology revolutionizing wearables, smart sensors, and IoT [5]–[8] by reducing production costs through maskless additive manufacturing at low processing temperature. With functional materials, PE offers flexibility, porosity, non-toxicity, and biodegradability, surpassing traditional silicon electronics.

Optimal PE performance is often achieved with vacuum-deposited molecular substrates, but solution-based techniques like spin coating and inkjet printing are gaining attention for their efficiency and ultra-low costs. Printing technologies include replication printing (e.g., gravure printing) for high-throughput manufacturing and jet printing (e.g., Fig. 2 inkjet printing) for small-scale, bespoke fabrication. Despite these advantages, additive manufacturing in PE faces challenges due

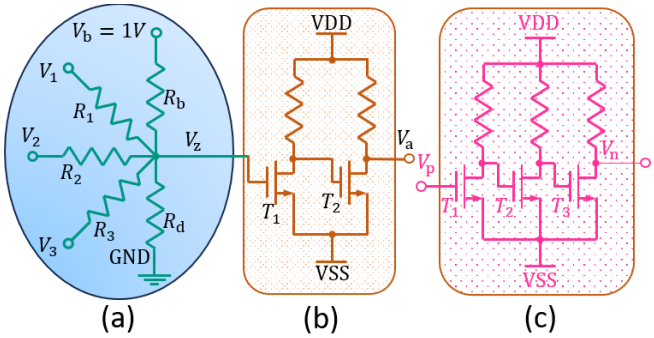

Fig. 3. Circuit primitives of the printed neuromorphic circuit (pNC). (a) Example of a 3-input, 1-output printed resistor crossbar. (b) printed tanh-like (ptanh) nonlinear activation circuit. (c) Inverter-based printed negative weight circuit.

to their low device counts, large device dimensions, and high variability. Therefore, PE is not intended to compete with silicon-based devices in performance for VLSI applications but to complement them in resource-limited and cost-sensitive edge computing areas, like disposable electronics or wearable devices [5]–[8]. In these scenarios, computational tasks are simple and tolerate compute imprecision, requiring only small-scale circuits. However, high printing variation caused by ink dispersion [19], wetting [20] and printer inaccuracy significantly impacts printed computing systems.

B. Printed Analog Neuromorphic Circuits

With the progression of artificial intelligence, neuromorphic computing has emerged as an effective approach for solving complex tasks as they can directly operate sensory data in the analog domain and thus require significantly reduced hardware footprints compared to digital counterparts. pNCs enable small neural networks (NNs) that are powerful enough for real-time sensor data processing without the computational resources needed for larger NNs. This makes them ideal for low device counts and customized configurations. By printing basic circuit primtives like weighted-sum, analog inverters for negative weights and nonlinear activations, pNCs can meet the computational needs of various PE applications.

a) Resistor crossbar: Fig. 3(a) shows a resistor crossbar in a pNC, simulating weighted-sum operations used in ANNs, and also common in in-memory computing [21]. The crossbar's behavior is described by

$$\sum_i \frac{V_i^{\mathrm{C}} - V_{\mathrm{out}}^{\mathrm{C}}}{R_i^{\mathrm{C}}} + \frac{V_{\mathrm{b}}^{\mathrm{C}} - V_{\mathrm{out}}^{\mathrm{C}}}{R_{\mathrm{b}}^{\mathrm{C}}} - \frac{V_{\mathrm{out}}^{\mathrm{C}}}{R_{\mathrm{d}}^{\mathrm{C}}} = I_{\mathrm{out}}^{\mathrm{C}}.$$

Here, $(\cdot)^{\mathrm{C}}$ denotes crossbar variables. The high resistivity of the non-linear activation (printed tanh-like) circuit allows neglecting $I_{\mathrm{out}}^{\mathrm{C}}$ [22]. Rewriting with conductance $g = 1/R$ and $V_{\mathrm{b}} = 1\mathrm{V}$, we get:

$$V_{\mathrm{out}}^{\mathrm{C}} = \sum_i \frac{g_i^{\mathrm{C}}}{G} V_i^{\mathrm{C}} + \frac{g_{\mathrm{b}}^{\mathrm{C}}}{G}, \tag{1}$$

where $G = \sum_i g_i^{\mathrm{C}} + g_{\mathrm{b}} + g_{\mathrm{d}}$. The output voltage $V_{\mathrm{out}}^{\mathrm{C}}$ is the weighted-sum of input voltages V_i^{C}, with weights and bias as conductance ratios. By designing these conductances, desired weights and biases are set. Unlike ANNs, weights and biases in Eq. (1) are limited to values below 1 and are positive. For negative weights or biases, inverter circuits (Fig. 3(c)) transform input voltages to negative values.

b) Printed tanh-like circuit: As shown in Fig. 3(b), a printed tanh-like (ptanh) circuit follows the resistor crossbar to mimic nonlinear activation functions in ANNs. The transfer characteristic is:

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

where $(\cdot)^{\text{A}}$ denotes activation circuit variables. Parameters η_i adjust the tanh function and are determined by component values $\boldsymbol{q}^{\text{A}} = [R_1^{\text{A}}, R_2^{\text{A}}, T_1^{\text{A}}, T_2^{\text{A}}]$.

C. Recurrent Neural Networks

RNNs are designed to process inputs of variable lengths by maintaining an internal hidden state $\boldsymbol{h}$, which is updated at each time step. This capability allows RNNs to capture temporal dependencies and patterns in sequential data. RNNs have demonstrated significant success in applications such as handwriting recognition and are theoretically Turing-complete, as they can execute any sequence of operations. The general RNN state equations are:

$$\boldsymbol{h}_k = f_1(f_2(\boldsymbol{h}_{k-1}) + f_3(\boldsymbol{x}_k)); \boldsymbol{y}_k \quad = f_4(\boldsymbol{h}_k) \qquad (2)$$

where k is the time step, $\boldsymbol{h}_k$ is the hidden state, $\boldsymbol{x}_k$ is the input, and $\boldsymbol{y}_k$ is the output. The functions $f_1(\cdot), \cdots, f_4(\cdot)$ are classic ANN operations, such as learnable affine mappings and activation functions. Recent advancements have also explored the integration of attention mechanisms, traditionally used in Transformer models, within RNNs frameworks which enhances the efficiency and versatility of RNNs, enabling better handling of temporal data by leveraging both sequential processing and attention-based parallelism.

D. Printed Temporal Processing Neuromorphic Circuit

pNCs are designed to complement silicon-based electronics by serving as cost-effective edge computing units in IoT systems. Recent advancements have enabled pNCs to process time-series data, and is crucial for applications like stress detection, where temporal changes are more informative than absolute signal values [8]. By combining printed capacitors with existing pNCs primitives, researchers have developed pTPBs and stacked multiple pTPBs for the realization of more complex functions. To the end, a mathematical model for the pTPBs and pTPNCs have been developed, along with an objective to optimize the components such as capacitance and crossbar resistances that represent weights and biases, to fulfill the specific computational requirements of various applications.

E. Related Work

Despite significant advancements in PE, additive manufacturing introduces variability in printed components due to many factors like ink dispersion, droplet irregularities, and missing droplets [20, 23]. These variations are often modeled using a uniform distribution for electrical characteristics and addressed by a Gaussian Mixture Model at the device level [24]. Multiple efforts have been made to mitigate these variation which include quality control engineering [25], material science innovations [20], and variation-aware training strategies [12].

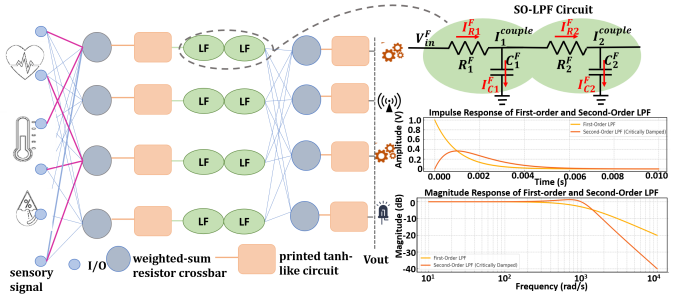

Fig. 4. Schematic of a 6-input 4-output printed temporal processing block (pTPB) including sensory signals from various inputs, processed through a weighted-sum resistor crossbar, second-order low-pass filters (LPF), and a printed tanh-like activation circuit. Signal flows are shown in both time and frequency domain of both printed filters.

However, variation in temporal processing units and in temporal input-data have not yet been adequately addressed. To address this limitation, our research introduces a variation-aware second-order pTPB that, when combined with pNCs, forms robust pTPNCs. Additionally, we address variation in sensory input signals using data augmentation techniques to enhance model robustness.

III. Robustness-Aware Adaptive Temporal Processing Block

pNCs are designed to enhance silicon-based electronics, particularly in cost-effective edge computing applications like wearable devices or smart packaging. In some scenarios, such as stress detection [26], the absolute values of sensory signals may not provide significant insights due to individual variability. Instead, the temporal dynamics of these signals are more informative. Previous approaches have utilized first-order low-pass filters (LPFs) in pNCs for basic input signal processing tasks. While the first-order LPFs can reduce high-frequency noise and capture signal trends, they are often inadequate for the dynamic nature of sensor signals, which require more complex analysis. This shortfall limits their utility in accurately processing data for applications such as stress detection, where signals can vary widely between time and amplitude.

To overcome these limitations, we propose a learnable second-order filter (SO-LF) within pNCs. Second-order filters offer a superior dynamic response compared to first-order filters, making them more adapted to handling the intricacies of temporal sensory data due to their sharper cutoff and better signal component separation, and are essential for capturing and analyzing both the temporal changes in input signals and also noisy input-data.

We achieve this by integrating two learnable printed resistors and capacitors connected back-to-back (as shown in Fig. 4), thus enabling the circuits to retain and process temporal sensory information and emulating the characteristics of second-order RC low-pass filters. To design the bespoke signal processing behaviors for target tasks, we have developed the mathematical model for the proposed variation-aware second-order pTPB and the corresponding ADAPT-pNC. This includes an optimization objective that allows the components within pTPBs (such as capacitance) to be fine-tuned along with the resistances in the crossbars, which act as weights and biases. This approach enables us to understand and optimize the performance of pNCs

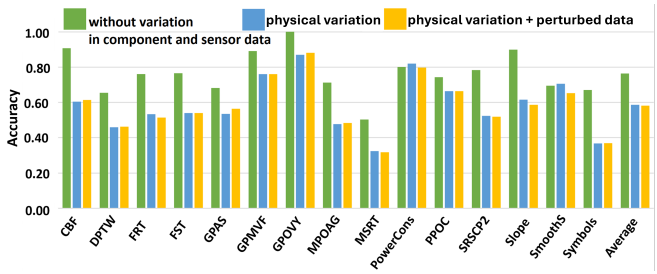

Fig. 5. Baseline tested under physical variations and perturbed sensor inputs.

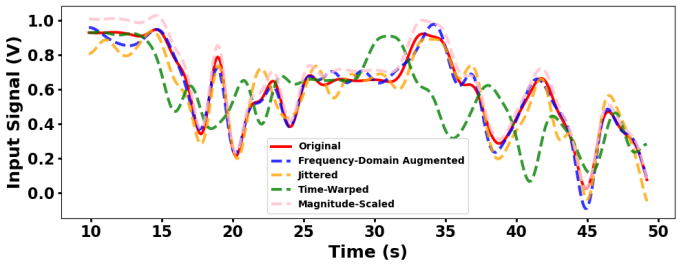

Fig. 6. Time-series augmentation techniques applied on PowerCons dataset: original, jittering, time-warping, magnitude scaling, and frequency-domain augmentation.

and adapt to variations in input sensors and physical units. Additionally, in the proposed robustness-aware framework, we incorporate data augmentation techniques during both the training and testing phases.

1) Modeling of second-order filter: To model the second-order learnable filter (SO-LF), we consider it as two successive first-order learnable filters. Initially, we analyze the model of a first-order learnable filter without considering its coupling to the subsequent circuit. The relationship between the input and output voltage is derived and represented in Eq. (3). The discrete-time update equations for $V_{\text{out},K}^{F_1}$ and $V_{\text{out},K}^{F_2}$ are shown in Eq. (4) and Eq. (5), respectively.

$$V_{\text{out},K}^{F} = \frac{R^F C^F}{R^F C^F + \Delta t} V_{\text{out},K}^{F} + \frac{\Delta t}{R^F C^F + \Delta t} V_{\text{in},K}^{F} \tag{3}$$

$$V_{\text{out},K}^{F_1} = \frac{R_1^F C_1^F}{R_1^F C_1^F + \Delta t} V_{\text{out},K}^{F_1} + \frac{\Delta t}{R_1^F C_1^F + \Delta t} V_{\text{in},K}^{F_1} \tag{4}$$

$$V_{\text{out},K}^{F_2} = \frac{R_2^F C_2^F}{R_2^F C_2^F + \Delta t} V_{\text{out},K}^{F_2} + \frac{\Delta t}{R_2^F C_2^F + \Delta t} V_{\text{out},K}^{F_1} \tag{5}$$

In these equations, Δt refers to the step size of the temporal discretization. The variables $V_{\text{in},k}^{F_1}$, $V_{\text{out},k}^{F_1}$, and $V_{\text{out},k}^{F_2}$ represent the input, intermediate, and output voltages of the filter at time step k, respectively. By appropriately selecting the resistor values R_1^F, R_2^F and the capacitor values C_1^F, C_2^F, a desired filtering behavior can be achieved. Despite previous work, in our approach, the resistors and capacitors are trained separately. In this work, these values are learned along with the crossbar resistors to optimize performance for specific tasks.

2) Modeling of second-order coupled filter: To connect the learnable filters with the resistor crossbars, as shown in [8], it is imperative to consider their coupling effects. This coupling primarily arises because the current flowing through the resistor R_1^F does not entirely feed into the capacitor C_1^F; similarly, the current through R_2^F does not fully enter C_2^F. Instead, a portion of the current is shunted towards R_2^F and the crossbar, respectively, as indicated by the red arrows (I_1^{couple}) and (I_2^{couple}) in Fig. 4. Eq. (6) and Eq. (7) depict this decoupling, while Eq. (8) and Eq. (9) represent the decoupling factors.

Consequently, Eq. (10) and Eq. (11) are reformulated to account for these factors. Additionally, μ is influenced by the frequency of the input signal, which is typically unknown during the design stage. To incorporate circuit coupling into the training process, we determine the general range of μ through SPICE simulations using the printed Process Design Kit (pPDK) [27]. Specifically, by performing SPICE simulations on the target datasets with general printable resistances and capacitances, we empirically determined that $\mu \in [1, 1.3]$ for the given applications.

$$I_{R1}^F = I_{C1}^F + I_1^{\text{couple}} \tag{6}$$

$$I_{R2}^F = I_{C2}^F + I_2^{\text{couple}} \tag{7}$$

$$I_{R1}^F = \mu I_{C1}^F \tag{8}$$

$$I_{R2}^F = \mu I_{C2}^F \tag{9}$$

$$V_{\text{out},K}^{F_1} = \frac{R_1^F C_1^F}{\mu_1 R_1^F C_1^F + \Delta t} V_{\text{out},K}^{F_1} + \frac{\Delta t}{\mu_1 R_1^F C_1^F + \Delta t} V_{\text{in},K}^{F_1} \tag{10}$$

$$V_{\text{out},K}^{F_2} = \frac{R_2^F C_2^F}{\mu_2 R_2^F C_2^F + \Delta t} V_{\text{out},K}^{F_2} + \frac{\Delta t}{\mu_2 R_2^F C_2^F + \Delta t} V_{\text{out},K}^{F_2} \tag{11}$$

The SO-LF enhances the robustness of the model under variations. During training, we consider variations on both the components and the input data to further improve robustness.

A. Variation-Aware Training of proposed Second-Order Filter

In this framework, the parameters θ (corresponding to the printed resistances), R (resistors), and C (capacitors) of the SO-LFs are trainable and subject to process variations. These trainable parameters are modeled as random variables during training: $\theta \sim p(\theta)$, $C \sim p(C)$, and $R \sim p(R)$, with distributions representing the process variations. Furthermore, the decoupling factor (μ) and the initial voltage (V_0^F) are modeled as random variables but are not trainable.

To account for these variations, we use a reparameterization strategy: $\theta = \theta_0 \odot \epsilon_\theta$, $C = C_0 \odot \epsilon_C$, and $R = R_0 \odot \epsilon_R$, where ϵ_θ, ϵ_C, and ϵ_R are random variables that follow distributions $p(\epsilon)$. This allows us to model process variations while still optimizing the trainable parameters θ, R, and C.

The training objective is to minimize the expected loss:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{p(\theta),p(C),p(R),p(\mu),p(V_0^F)}\left[\mathcal{L}(\theta, C, R, \mathbf{x}_K, \cdots, x_0, \mathbf{y}, \mu, V_0^F)\right] \\ &= \int \mathcal{L}(\theta, C, R, \mathbf{x}_K, \cdots, x_0, \mathbf{y}, \mu, V_0^F)\, d\theta\, dC\, dR\, d\mu\, dV_0^F\end{aligned} \tag{12}$$

which is approximated via Monte Carlo sampling:

$$\mathcal{L} \approx \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\theta_i, C_i, R_i, \mathbf{x}_K, \cdots, x_0, \mathbf{y}, \mu_i, V_{0,i}^F) \tag{13}$$

This leads to the final training objective in Eq. (14), which ensures optimal accuracy and robustness against variations.

$$\min_{\theta,C,R} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\theta_i, C_i, R_i, \mathbf{x}_K, \cdots, x_0, \mathbf{y}, \mu_i, V_{0,i}^F) \tag{14}$$

TABLE I
RESULT ON 15 BENCHMARK TIME-SERIES DATASETS: (A) HARDWARE-AGNOSTIC ELMAN RECURRENT NEURAL NETWORK (RNN), (B) BASELINE PRINTED TEMPORAL PROCESSING NEUROMORPHIC CIRCUIT (PTPNC), (C) ROBUSTNESS-AWARE ADAPT PNC UNDER PRECISION PRINTING ($\pm 10\%$ VARIATION) AND PERTURBED INPUT DATA.

Dataset	Elman RNN (Reference acc.)	pTPNC (Baseline)	Robustness-Aware ADAPT-pNC
CBF	0.683 ± 0.036	0.615 ± 0.142	0.877 ± 0.006
DPTW	0.507 ± 0.006	0.462 ± 0.003	0.700 ± 0.008
FRT	0.597 ± 0.120	0.514 ± 0.003	0.677 ± 0.021
FST	0.509 ± 0.066	0.540 ± 0.000	0.591 ± 0.086
GPAS	0.452 ± 0.003	0.564 ± 0.012	0.568 ± 0.004
GPMVF	0.637 ± 0.054	0.760 ± 0.010	0.900 ± 0.010
GPOVY	0.540 ± 0.007	0.881 ± 0.030	1.000 ± 0.000
MPOAG	0.560 ± 0.042	0.483 ± 0.006	0.654 ± 0.021
MSRT	0.261 ± 0.008	0.317 ± 0.022	0.531 ± 0.004
PowerCons	0.651 ± 0.010	0.797 ± 0.865	0.880 ± 0.018
PPOC	0.711 ± 0.001	0.664 ± 0.005	0.660 ± 0.000
SRSCP2	0.489 ± 0.011	0.519 ± 0.010	0.525 ± 0.001
Slope	0.559 ± 0.002	0.587 ± 0.041	0.765 ± 0.016
SmoothS	0.447 ± 0.011	0.653 ± 0.015	0.864 ± 0.011
Symbols	0.141 ± 0.002	0.369 ± 0.050	0.697 ± 0.003
Average	**0.501 ± 0.025**	**0.582 ± 0.031**	**0.726 ± 0.014**

TABLE II
COMPARISON OF RUNTIME (AVERAGE)

Models	Elman RNN (Reference acc.)	pTPNC (Baseline)	Robustness-Aware ADAPT-pNC
Runtime (avg.)	2.345 ms	0.230 s	2.537 s

B. Variation in Input Sensor

To improve robustness, we employ several data augmentation techniques during training, enhancing dataset variability and enabling the model to generalize to unseen conditions. The augmentations used for time-series data include frequency-domain noise to simulate signal distortions (applied to datasets like PowerCons and SmoothS), random cropping to mimic partial data availability (effective for datasets such as MSRT and Symbols), jittering to introduce sensor inaccuracies, time warping to alter the temporal dynamics, and magnitude scaling to simulate changes in sensor readings. These techniques collectively simulate real-world sensor variations, improving model robustness by creating a more diverse and resilient training set. Fig. 6 shows the application of the combined data augmentation techniques on the PowerCons dataset.

IV. EVALUATION

A. Experiment Setup

1) Circuit Design Setup: The temporal processing block primitives in Fig. 4 (top (c)-(g)) were designed using the well-established n-EGT P-PDK [28]. We obtained the filter magnitude, impulse response and the cutoff frequencies from the SPICE simulations in Cadence Virtuoso[2]. The resistances in the filters are designed with lower values ($<1\text{k}\Omega$) than that of the resistors in crossbars ($100\text{k}\Omega$-$10\text{M}\Omega$), while the capacitances are designed as high as the printing technology allows (100nF-$100\mu\text{F}$) to minimize the coupling effect.

[2]https://www.cadence.com/en_US/home.html

2) Dataset Preparation: We have considered 15 datasets from the UCR Time Series Classification Archive [29]. The datasets were preprocessed by uniformly resizing the series lengths to 64, normalizing the signal values to the range of $[-1, 1]$, and reshuffling and splitting the datasets into training (60%), validation (20%), and test (20%) sets. We used a 2-layer RNN as a reference accuracy model and pTPNC as baseline.

For data augmentation, we employed the `tsaug` framework [30] to apply the augmentation techniques. The augmented data was combined with the original unaugmented data, and both were used during training, validation and testing.

3) Training Setup: For each dataset, we utilize a 2-layer robustness-aware pTPNC, which includes two second-order pTPB layers per layer, with the number of learnable filters N_F matching the number of inputs for that layer. To optimize the objective function, we employ the AdamW optimizer [31] with default settings for full-batch training to adjust the optimization parameters. The initial learning rate is set at 0.1 and is halved after every 100 epochs of no improvement in the validation loss (patience). Training is terminated once the learning rate falls below 10^{-5}. This training process is repeated 10 times, each with a different random seed ranging from 0 to 9, to reduce the risk of poor initialization and ensure a robust solution.

For experiments involving data augmentation, we used Ray Tune [32] to tune hyperparameters such as crop size, noise level, and time warping. The optimal configurations for each dataset were applied during training. All implementations and training were done in PyTorch [33].

B. Results

To evaluate the accuracy of our proposed robustness-aware ADAPT-pNCs[3], we conducted a series of experiments and compared the results against a baseline model and a reference model. The baseline is a 2-layer no-variation-aware pTPNC, while the reference model is a 2-layer Elman RNN, as implemented in PyTorch [33].

After training, we selected the top three models for each dataset based on their accuracy on the test set. The selected models were evaluated on an augmented test set with a 10% variation in physical components. For each dataset, we report the mean accuracy across random seeds and the corresponding standard deviation, as shown in Tab. I. The averaged accuracy and standard deviation across all datasets are provided in the last row. Additionally, the average runtimes for the proposed ADAPT-pNC, baseline, and Elman are reported in Tab. II. Tab. III presents the hardware component and power metrics for the proposed model. Furthermore, we conducted an ablation study to evaluate the effectiveness of different configurations under a 10% physical variation scenario: variation-aware (VA) during training, augmented training (AT), and second-order learnable filters (SO-LF).

C. Discussion

The impact of manufacturing variability on printed electronics cannot be overstated. Variations such as ink dispersion,

[3]Code is available at https://github.com/KIT-Neuromorphic-Computing/ADAPT-pNC.

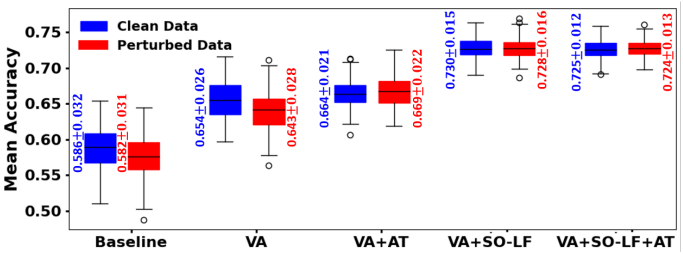

Fig. 7. Average accuracy comparison of variation aware (VA), augmented training (AT), and second-order learnable filters (SO-LF) and combined (VA +SO-LF+AT) with baseline using an ablation study.

droplet irregularities, and missing droplets [20, 23] introduce significant challenges. These variations are often modeled using uniform distributions and Gaussian Mixture Models at the device level [24]. As shown in Fig. 5, a trained no-variation-aware pTPNC, when tested under these variations, exhibits a significant drop in accuracy. This clearly illustrates the necessity for a robustness-aware framework, which motivates our exploration in this direction.

The results in Tab. I highlight the accuracy of three models: the Elman RNN (reference model), pTPNC (baseline model), and the robustness-aware ADAPT-pNC across 15 datasets. The Elman RNN and the baseline pTPNC achieve an average accuracy of 0.501 and 0.582, respectively, while the robustness-aware ADAPT-pNC achieves the highest accuracy of 0.726, leading to $\approx 24.7\%$ improvement from the baseline, showing the effectiveness of incorporating robustness-aware techniques. Furthermore, Tab. II also shows that while the proposed model requires more computational resources ($2.537s$ on average, compared to $0.230s$ for the baseline), significant accuracy gains justify this increase. The trade-off between computational efficiency and accuracy is important for target application tasks [1, 4] where robustness to data variability is critical. Moreover, datasets CBF and GPOVY further highlight these improvements. In CBF dataset, Elman RNN shows an accuracy of 0.683, while the baseline pTPNC is reduced to 0.615. However, the robustness-aware ADAPT-pNC improves to 0.877, reflecting the model's ability to handle complex variations. Similarly, for GPOVY, Elman RNN and baseline pTPNC achieve 0.540 and 0.881, respectively, but the proposed model reaches an impressive accuracy of 1.000. This demonstrates the model's capability to generalize effectively in challenging datasets.

Fig. 7 demonstrates the improvement in mean accuracy across 15 benchmark datasets for different training configurations. In comparison with the baseline, which achieves $\approx 58\%$ accuracy in both clean and perturbed data, adding variation-aware (VA) training improves accuracy by 11.6% for clean data and 10.5% for perturbed data. Augmented training (AT) further increases accuracy by 13.3% for clean data and 15% for perturbed data, accompanied by a decrease in standard deviation. The inclusion of SO-LFs results in a significant improvement of 24.6% for clean data and 25.1% for perturbed data, while also reducing both standard deviation and variability.

Finally, the combination of VA + SO-LF + AT yields the highest accuracy, showing a 23.7% improvement for clean data and 24.4% for perturbed data, with the lowest observed variability. These improvements were calculated as the percentage increase in accuracy between the new model and the baseline.

To investigate the additional hardware resources required by the new circuit design, we collect the device counts and total power consumption of both existing pTPNC [8] and the proposed ADAPT-pNCs in different application scenarios (i.e., datasets). Analogously, we averaged the hardware costs across all datasets to report a comprehensive comparison regarding the hardware costs between the pTPNC and its pNC counterpart. The results in Tab. III shows a significant power reduction ($\approx 91\%$) at the expense of additional hardware ($\approx 1.9\times$) compared to existing works [8].

TABLE III
HARDWARE COSTS FOR BASELINE PTPNC [8] VS PROPOSED ROBUSTNESS-AWARE ADAPT-PNC

Dataset	#Transistors		#Resistors		#Capacitors		#Total Devices		Power (mW)	
	pTPNC	Proposed	pTPNC	Proposed	pTPNC	Proposed	pTPNC	Proposed	pTPNC	Proposed
CBF	24	59	84	147	6	24	114	230	0.653	0.06
DPTW	48	126	222	337	12	24	282	487	1.501	0.06
FRT	16	29	50	71	4	12	106	112	0.372	0.03
FST	16	30	50	75	4	12	70	117	0.342	0.03
GPAS	16	36	50	86	4	12	70	134	0.374	0.03
GPMVF	16	38	50	86	4	12	70	136	0.389	0.03
GPOVY	16	37	50	82	4	12	70	131	0.324	0.03
MPOAG	24	61	84	146	6	24	114	231	0.625	0.06
MSRT	44	127	170	335	6	60	210	522	1.188	0.15
PowerCons	16	34	50	79	4	12	70	125	0.363	0.03
PPOC	16	23	50	67	4	12	70	102	0.381	0.03
SRSCP2	16	32	60	89	4	12	70	133	0.472	0.03
Slope	24	27	50	72	6	12	114	111	0.388	0.03
SmoothS	16	59	84	148	4	24	70	231	0.610	0.06
Symbols	48	136	222	391	12	84	282	611	1.526	0.210
Average	**23**	**57**	**88**	**147**	**6**	**23**	**118**	**228**	**0.634**	**0.058**

In summary, our experiments show that the accuracy improvement from the Elman RNN to the pTPNC (baseline) is approximately 15%. For the robustness-aware ADAPT-pNC, this improvement increases to approximately 45%. Additionally, ADAPT-pNC requires approximately $1.9\times$ more devices compared to the pTPNC (baseline). Also, the robustness-aware ADAPT-pNC, particularly with the variation-aware (VA) + second-order learnable filter (SO-LF) + augmented training (AT) configuration, provides accuracy improvement and robustness to perturbations. Although it requires more computational time, the model's ability to handle real-world data variability makes it an ideal solution for target tasks that require high accuracy and reliability.

V. CONCLUSION

With the advent of next-generation electronics, PE have attracted considerable research interest. These innovations offer unique advantages like flexibility and biodegradability, which are not achievable with traditional silicon-based electronics. However, existing pTPNCs fail to account for manufacturing variations and sensor signal perturbation. This limitation significantly hinders the utility in domains where very sensitive temporal information needs to be processed. To address this problem, in this work, we introduced a robustness-aware framework for pTPNC to address manufacturing variability challenges. By integrating second-order learnable filters, data augmentation, and variation-aware training, our approach enhances the reliability and robustness of pTPNCs. Future work may include new architectural search methodologies, for ADAPT-pNCs to further address sensor variations and improve time-series data dependency capture.

ACKNOWLEDGMENT

This work has been supported by the the European Research Council (ERC).

REFERENCES

[1] S. Li *et al.*, "The Internet of Things: A Survey," *Information systems frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

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

[3] B. Chen, M. Zhang, H. Chen, A. S. Mujumdar, and Z. Guo, "Progress in smart labels for rapid quality detection of fruit and vegetables: A review," *Postharvest Biology and Technology*, vol. 198, p. 112261, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925521423000224

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

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

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

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

[8] H. Zhao *et al.*, "Towards temporal information processing – printed neuromorphic circuits with learnable filters," in *Proceedings of the 18th ACM International Symposium on Nanoscale Architectures*, ser. NANOARCH '23. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3611315.3633249

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

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

[11] J. Li *et al.*, "Micro and Nano Materials and Processing Techniques for Printed Biodegradable Electronics," *Materials Today Nano*, vol. 18, 2022.

[12] H. Zhao *et al.*, "Highly-Bespoke Robust Printed Neuromorphic Circuits," in *Design, Automation & Test in Europe Conference & Exhibition*. IEEE, 2023.

[13] S. D. Gardner *et al.*, "An Inkjet-Printed Artificial Neuron for Physical Reservoir Computing," *IEEE Journal on Flexible Electronics*, vol. 1, no. 3, pp. 185–193, 2022.

[14] R. A. Nawrocki *et al.*, "Neurons in Polymer: Hardware Neural Units Based on Polymer Memristive Devices and Polymer Transistors," *IEEE Transactions on Electron Devices*, vol. 61, no. 10, pp. 3513–3519, 2014.

[15] H. Zhao *et al.*, "Highly-Dependable Printed Neuromorphic Circuits Based on Additive Manufacturing," *Flexible and Printed Electronics*, vol. 8, no. 2, p. 025018, 2023.

[16] P. Pal *et al.*, "Analog printed spiking neuromorphic circuit," in *IEEE DATE*, 2024, p. 6 S.

[17] H. Hu *et al.*, "Inkjet-printed bipolar resistive switching device based on Ag/ZnO/Au structure," *Applied Physics Letters*, vol. 119, no. 11, p. 112103, 09 2021. [Online]. Available: https://doi.org/10.1063/5.0058526

[18] Y. Kim, Y. Li, H. Park, Y. Venkatesha, A. Hambitzer, and P. Panda, "Exploring temporal information dynamics in spiking neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, pp. 8308–8316, Jun. 2023. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/26002

[19] H. Abdolmaleki, P. Kidmose, and S. Agarwala, "Droplet-based techniques for printing of functional inks for flexible physical sensors," *Advanced Materials*, vol. 33, no. 20, p. 2006792, 2021.

[20] E. Sowade, M. Polomoshnov, and R. R. Baumann, "The design challenge in printing devices and circuits: Influence of the orientation of print patterns in inkjet-printed electronics," *Organic Electronics*, vol. 37, pp. 428–438, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566119916303068

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

[22] D. D. Weller *et al.*, "Programmable Neuromorphic Circuit based on Printed Electrolyte-Gated Transistors," in *2020 Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2020, pp. 446–451.

[23] H. Abdolmaleki, P. Kidmose, and S. Agarwala, "Droplet-based techniques for printing of functional inks for flexible physical sensors," *Advanced Materials*, vol. 33, no. 20, p. 2006792, 2021.

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

[25] C. Lee, H. Kang, H. Kim, H. Anh, D. Nguyen, and K. Shin, "Quality control with matching technology in roll to roll printed electronics," *Journal of Mechanical Science and Technology*, vol. 24, no. 1, p. 315, 2010.

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

[27] F. Rasheed, M. Hefenbrock, M. Beigl, M. B. Tahoori, and J. Aghassi-Hagmann, "Variability modeling for printed inorganic electrolyte-gated transistors and circuits," *IEEE transactions on electron devices*, vol. 66, no. 1, pp. 146–152, 2018.

[28] F. Rasheed, M. Hefenbrock, R. Bishnoi, M. BeigI, J. Aghassi-Hagmann, and M. B. Tahoori, "Predictive modeling and design automation of inorganic printed electronics," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019, pp. 30–35.

[29] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive," July 2015, www.cs.ucr.edu/~eamonn/time_series_data/.

[30] tsaug Contributors, "tsaug: A python package for time series data augmentation," 2021, accessed: 2024-07-11. [Online]. Available: https://github.com/arundo/tsaug

[31] I. Loshchilov, F. Hutter *et al.*, "Fixing weight decay regularization in adam," *arXiv preprint arXiv:1711.05101*, vol. 5, 2017.

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

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019. [Online]. Available: https://arxiv.org/abs/1912.01703