Analog Printed Spiking Neuromorphic Circuit

Priyanjana Pal[1]♣, Haibin Zhao[1]♣, Maha Shatta[1]♣, Michael Hefenbrock[2],
Sina Bakhtavari Mamaghani[1], Sani Nassif[3], Michael Beigl[1], Mehdi B. Tahoori[1]
[1]*Karlsruhe Institute of Technology,* [2]*RevoAI GmbH,* [3]*Radyalis LLC*
[1]{priyanjana.pal, haibin.zhao, maha.shatta, sina.mamaghani, michael.beigl, mehdi.tahoori}@kit.edu
[2]michael.hefenbrock@revoai.de, [3]srn@radyalis.com

***Abstract*—Biologically-inspired Spiking Neural Networks have emerged as a promising avenue for energy-efficient, high-performance neuromorphic computing. With the demand for highly-customized and cost-effective solutions in emerging application domains like soft robotics, wearables, or IoT-devices, Printed Electronics has emerged as an alternative to traditional silicon technologies leveraging soft materials and flexible substrates. In this paper, we propose an energy-efficient analog printed spiking neuromorphic circuit and a corresponding learning algorithm. Simulations on 13 benchmark datasets show an average of 3.86× power improvement with similar classification accuracy compared to previous works.**

I. Introduction

Next-generation electronics encompassing soft robotics, wearables, near-sensor processing, and the Internet of Things (IoTs) are shifting towards lightweight, flexible, and low-cost solutions. However, conventional silicon-based technologies fall short of meeting these requirements due to their bulky substrates and high production costs [1]. In this context, printed electronics (PE) has emerged as a promising solution, providing design flexibility, cost-efficiency, and fast prototyping, especially for low-volume applications [2].

Advancements in machine learning and neuromorphic computing have gained significant attention in the area of cognitive computational tasks. This paradigm shift in computing is driven by the limitation of conventional von Neumann architectures, which underpin most digital computers, in terms of energy efficiency, adaptability, and cognitive capabilities. In this regard, Spiking Neural Networks (SNNs), inspired by the information processing in biological brains, represent a transformative evolution. In contrast to traditional Artificial Neural Networks (ANNs) that rely on continuous activations [3], SNNs communicate through discrete, event-driven spikes, mimicking the behavior of neurons in the human brain [4, 5]. This unique behavior not only allows for more biologically plausible modeling but also presents opportunities for achieving unprecedented energy efficiency.

The decision to opt between digital and analog SNN implementation is crucial in neuromorphic computing, as it is influenced by the specific requirements of the tasks at hand. Digital SNNs provide exceptional robustness in noisy conditions and scalability for more complex networks [6, 7]. On the contrary, analog SNNs are more competitive in area, energy efficiency, and real-time processing, making them a promising choice for PE applications with strict power constraints.

Leveraging the benefits offered by PE and neuromorphic computing, prior studies successfully realized the implementation of analog printed spiking neurons on organic substrates [8, 9]. Additionally, researchers presented adaptations to the architectures and training algorithms for analog Printed Artificial Neural Networks (P-ANNs) [10]. Also, they introduced training algorithms that account for variation, aging effects, and power considerations [11, 12]. Although significant efforts have been made to implement various printed computing paradigms, very few studies have been reported on programmable analog Spiking Neural Network using PE.

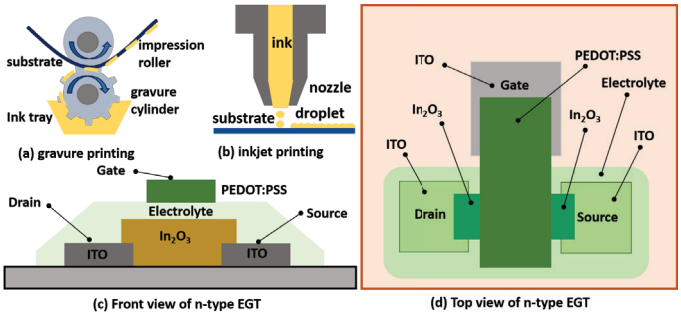

Fig. 1. Schematic of (a) gravure printing (b) inkjet printing; (c) front view and (d) top view of a printed N-type EGT.

In this work, we propose an analog printed spiking neuron circuit design and its associated learning algorithm for neural network computation. To the best of our knowledge, this is the first demonstration of a complete spiking neuromorphic computing system implemented using inorganic printed electronics. The contributions of this work are:

- We designed a novel programmable energy-efficient spiking neuron using only printed N-type electrolyte-gated transistor (EGT) technology, suitable for low-voltage and energy-harvested edge applications.
- We proposed a differential Transformer-based machine learning model to enable the training of this circuitry for specific target tasks.
- We validated the simulation-based performance of the proposed neuron on a trained SNN and evaluated its associated learning algorithm on a set of benchmark datasets.

The rest of this paper is structured as follows: Sec. II, provides the background of this work. Sec. III proposes the printed analog spiking neuron design and develops a learning algorithm to solve specific classification tasks. In Sec. IV, we evaluate the effectiveness of the proposed models on benchmark datasets, compare our results with the existing printed analog P-ANNs, and discuss its application. Finally, Sec. V concludes this paper.

II. Preliminaries

A. Printed Electronics (PE)

PE is an additive manufacturing approach where printed materials are deposited layer-upon-layer to realize active devices,

♣Authors contributed equally to this work.

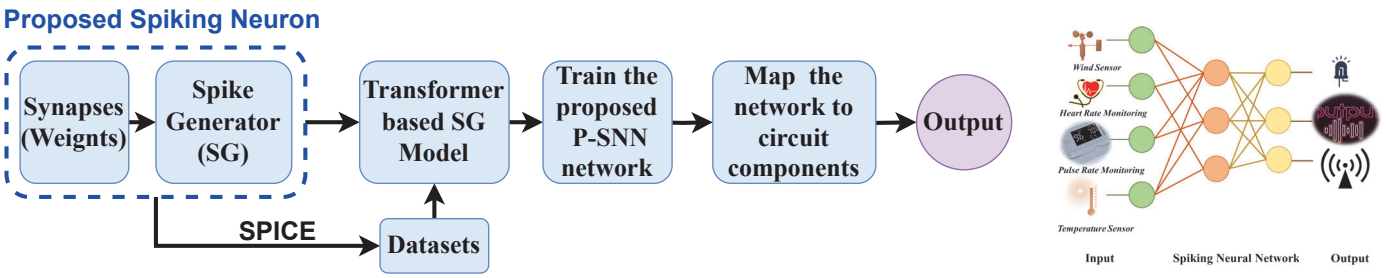

Fig. 2. Proposed flow for an on-demand printed spiking neuron design given a specification of a desired functionality realized through P-SNN training.

passive components, interconnects, and crossovers. Unlike conventional silicon electronics, PE requires fewer manufacturing steps and inexpensive fabrication processes. Various materials can be used for targeted applications of printed electronics, enabling flexibility and biocompatibility in next-generation electronics. PE can be categorized into high-volume replication (e.g., gravure printing) and customized jet printing (e.g., inkjet printing), as shown in Fig. 1(a) and Fig. 1(b) respectively [13]. Organic inkjet-printed FETs use lithographically structured semiconductors for source-drain channels which require higher supply voltages (≥ 5V) [14], while inorganic FETs can operate at sub-1V voltage range and are thus more promising for low-power applications. Fig. 1(c) (front view) and Fig. 1(d) (top view) show a typical N-type EGT with a top-gated geometry. The indium oxide (In_2O_3) semiconductor is used as a channel material while the gate dielectric is replaced with a solid polymer electrolyte.

B. Printed Artificial Neural Networks (P-ANNs)

P-ANNs emulate the functions of Artificial Neural Networks via operations such as weighted-sums and nonlinear activations. P-ANNs have the capability to directly operate on analog sensory input with significantly reduced hardware footprints compared to digital counterparts. Leveraging the analog processing, P-ANNs use printed resistor crossbars for weighted-sum operation, analog inverters for negative weights, and nonlinear tanh or ReLU activation functions [10]. However, P-ANNs lack the capacity to process temporal data and are not energy-efficient.

C. Spiking Neural Networks (SNNs)

Spiking Neural Networks (SNNs) derive inspiration from the brain's efficient information processing, utilizing discrete pulses for communication. They emulate the operational principles of a biological neuron, combining three components; namely dendrite, soma, and axon. For a biological neuron, the dendrite receives the signal, the soma integrates incoming signals and determines spike initiation based on membrane potential. Afterward, the axon facilitates the transmission of the soma's output. Unlike conventional ANNs, SNNs mirror the brain's event-driven nature, offering energy savings and real-time processing enhancements [15]. SNNs employ spiking neurons, transmitting data through spike timing or frequency.

D. Low-Power and Energy-Efficient Neuromorphic Design

Although neuromorphic computing has already been proven to be significantly power and energy-efficient compared to conventional approaches [16], ongoing research aims to further reduce the power consumption of these neuromorphic circuits. For instance, [17] developed a novel device to decrease the power required for the activation functions. The work of [18] utilized hardware-software co-design to optimize circuit structure for data flow in the computing process. Regarding computational paradigms, numerous silicon circuits have adopted brain-inspired SNN to minimize power in analog, digital, or mixed-signal by integrating synaptic inputs, generating action potentials, and transmitting them along neuron axons to connect with post-synaptic terminals.

III. Proposed Printed Spiking Neural Networks (P-SNNs)

The overall design of the proposed Printed Spiking Neural Network was initiated by designing the spiking neuron circuit, modeling its Transformer-based surrogate machine learning model, training the resulting spiking network, and finally mapping the P-SNN back to its corresponding circuit components as shown in Fig. 2.

A. Implementation of Printed Spiking Neuron

The design of the proposed Printed Spiking Neuron circuit comprises of three stages: synapses, charge network, and reset and discharge network. The synapses contain the input to the neuron, which resembles the functionality of the biological neuron. The charge network and the reset and discharge network illustrate the spike-generator circuit's functionality.

a) Synapses: Synapses are the part of the neuron that transmit signals from the presynaptic neuron's axon to the postsynaptic neuron's dendrite. For the sake of circuit realization, presynaptic neurons are represented as voltage inputs, and each presynaptic neuron is assigned a weight through a resistor crossbar as shown in Fig. 3. Using Nodal Analysis, the weighted sum is expressed by

$$\frac{V_g^1}{R_w^0} + \frac{V_g^1 - V_{in}^1}{R_w^1} \cdots + \frac{V_g^1 - V_{in}^N}{R_w^N} = 0, \tag{1}$$

where V_{g}^1 is the gate voltage of M_1, V_{in}^1 is the input voltage to the neuron, R_{w}^0 works as a voltage divider, R_{w}^1 and R_{w}^N

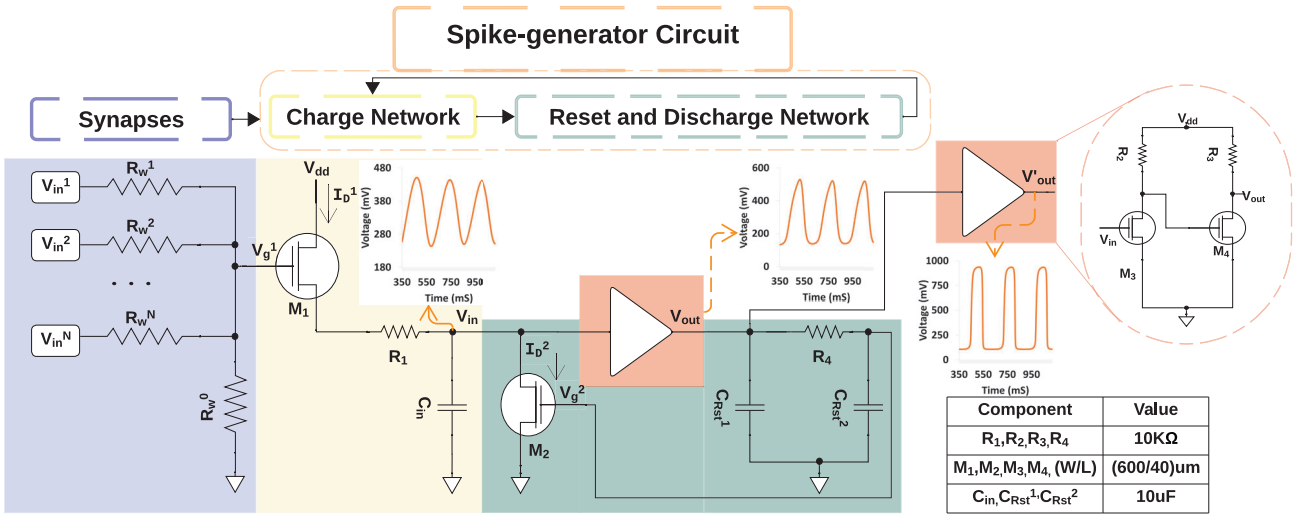

Fig. 3. Circuit level implementation of Printed Spiking Neuron which includes three stages: Synapses, Charge Network, and Reset and Discharge Network.

are used to resemble the weights of the connected neurons and N is the number of the presynaptic neurons connected as inputs. Additionally, another circuit is required for the negative weights, as proposed in [10]. To determine V_{g}^1, please refer to Sec. III-B for detailed calculation procedures.

b) Spike-generator Circuit: As shown in Fig. 3, both R_1 and C_{in} form a network to provide V_{in} (the voltage across C_{in}) to the amplifier with a delay directly proportional to both $R_1 C_{\mathrm{in}}$ and the frequency of the spikes. Additionally, the input voltage at M_1 gate, V_{g}^1 controls the current I_{D}^1, affecting the charging delay of the capacitor, thus influencing the spikes' frequency. The signal across the capacitor V_{in} is then strengthened through an amplifier circuit that consists of two analog inverters: the pull-down network uses an N-type transistor for discharge, and the pull-up network employs a single resistor for charging. While a single analog inverter can achieve the required amplification, it adds a 180-degree phase shift, necessitating the second inverter for phase correction. The amplifier output is connected to two RC networks: the first one consists of the pull-up resistor R_3 with C_{Rst}^1, and the second is R_4 and C_{Rst}^2. These two RC networks are essential for the oscillation functionality by adding a small phase shift due to capacitance charging. The voltage across the final capacitor, C_{Rst}^2, controls transistor M_2 which manages the discharging of C_{in}. As V_{g}^2 exceeds the threshold voltage of M_2, the current I_{D}^2 flows through M_2, initiating C_{in} discharge. However, as the input voltage from the synapses continues to supply the charge network, C_{in} gets recharged again, maintaining the circuit's spike oscillations.

The amplification of the signal's amplitude at V_{in} is shown in Fig. 3. However, the output signal, V_{out}, still remains considerably below supply voltage Vdd. So, a second amplifier is added to boost the neuron's output voltage to drive the postsynaptic neuron, aiming for a maximum amplitude at $V_{\mathrm{out}}^{'}$ (close to 1V Vdd).

B. Modeling and training of P-SNN

By connecting multiple printed spiking neurons, the computational paradigm of the SNN can be emulated, thereby offering the potential to achieve the desired functionalities. However, to fully leverage this potential, the component values of the circuits (e.g., the crossbar conductances representing the weights) should be designed and optimized for specific target tasks. For this, it is necessary to establish a P-SNN optimization model.

1) Modeling of P-SNN: An analytical model for the behavior of the resistor crossbar for weighted-sum is given in [11] as

$$V_g^1 = \sum_{n=1}^{N} V_{\mathrm{in}}^n \left(w_n \cdot \mathbb{1}_{\{g_n \geq 0\}}\right) + \mathrm{inv}(V_{\mathrm{in}}^n) \left(w_n \cdot \mathbb{1}_{\{g_n < 0\}}\right), \quad (2)$$

where g_n denotes the crossbar conductance by its absolute value and encodes with its sign, if the respective input is inverted (to express negative weight). Furthermore, $\mathrm{inv}(\cdot)$ refers to the negative tanh function that describes the transfer characteristic of the negation circuit. Finally, $\mathbb{1}_{\{\cdot\}}$ is an indicator function that returns 1 if the respective condition is true, else 0. Additionally, w_n refers to the weight given by

$$w_n = \frac{|g_n|}{\sum_m |g_m|}. \quad (3)$$

To enable gradient-based training via backpropagation [19], a fully differentiable model to describe the transfer characteristic of the printed spike-generator circuit is needed. However, given that the common hardware-agnostic SNN training frameworks are based on (Leaky-) Integration-Fire mechanism [20, 21], they are incompatible with proposed circuits, due to their device and material constraints, and so can not describe their transfer behavior. So, considering the circuit complexity, we utilize a Transformer-based neural network as the *surrogate spike-generator (SG) model* to learn the circuit behavior for mapping the input voltages into the output voltages. A Transformer [22]

is a neural network model initially proposed for natural language processing. It is, therefore, aptly suited for processing sequential data. The essential part of the Transformer is the attention mechanism, which enables the model to account for positional and value correlations. The effectiveness of Transformer has been shown by numerous state-of-the-art models like BERT [23] and GPT [24].

To prepare the data required for training the SG model, we conducted 5,000 SPICE simulations for a single spike-generator circuit in Fig. 3 based on the Printed Process Design Kit (P-PDK) [25]. The duration of the input voltage (V_g^1) is 3 s and the temporal step size is 1ms. To ensure that the surrogate model can comprehensively and accurately mimic the behavior of the original spike-generator circuit in any operating scenario, we designed the following patterns of input voltages (V_g^1), namely,

1) constant voltages ranging from 0V to 2V, serving to represent the case of stable inputs;
2) the output voltages obtained from 1), i.e., V'_{out}, representing the case of a cascade of multiple neurons; and
3) diverse harmonic signals with varying frequencies ($0-5$Hz), amplitudes ($0-1$A), phases ($0-2\pi$), and their combinations, expressing the circuit behavior in other complex situations.

After obtaining the data from SPICE simulation, we split them into three sets, a training set (70%) to guide the training of the surrogate model, a validation set (20%) for stopping to avoid overfitting, and a test set (10%) to evaluate the effectiveness of the surrogate model. We use Adam [26] with its default setup as the optimizer and the Mean Squared Error (MSE) between the model output and the SPICE simulation as the metric. After hyperparameter tuning, we choose a Transformer with three causal attention layers as the final surrogate circuit model. Causal attention thereby ensures that outputs are only determined by the signals at previous time steps. Moreover, each layer has three attention heads. The MSE is 1.1×10^{-6} on the validation set and 9.7×10^{-7} on the test set; therefore, we conclude that the model is capable of sufficiently interpolating and accurately predicting the output voltages.

Using the crossbar and the SG model, the transfer characteristic of the printed spiking neuron can be expressed as

$$\text{SG}\left(\sum_{n=1}^{N} \boldsymbol{V}_{\text{in}}^{n}\left(w_n \cdot \mathbb{1}_{\{g_n \geq 0\}}\right) + \text{inv}(\boldsymbol{V}_{\text{in}}^{n})\left(w_n \cdot \mathbb{1}_{\{g_n < 0\}}\right)\right), \tag{4}$$

where $\boldsymbol{V}_{\text{in}}^{i} \in \mathbb{R}^T$ represents the i-th input voltage sequence. Moreover, by connecting multiple spiking neurons, more complex computing tasks can be implemented.

2) Training of P-SNN: In the training of existing P-ANNs the cross-entropy loss $L(\cdot)$ is minimized w.r.t. the crossbar conductances $\boldsymbol{g}$ for maximizing the classification accuracy. However, given that the output of P-SNN is a temporal data series, temporal dynamics of the circuit output need to be considered. Therefore, to encourage the overall classification accuracy at every time step, a modified training objective can be formulated as

$$\underset{\boldsymbol{g}}{\text{minimize}} \ \frac{1}{T}\sum_{t=0}^{T} L\left(\boldsymbol{x}_t, \boldsymbol{y}, \boldsymbol{g}\right), \tag{5}$$

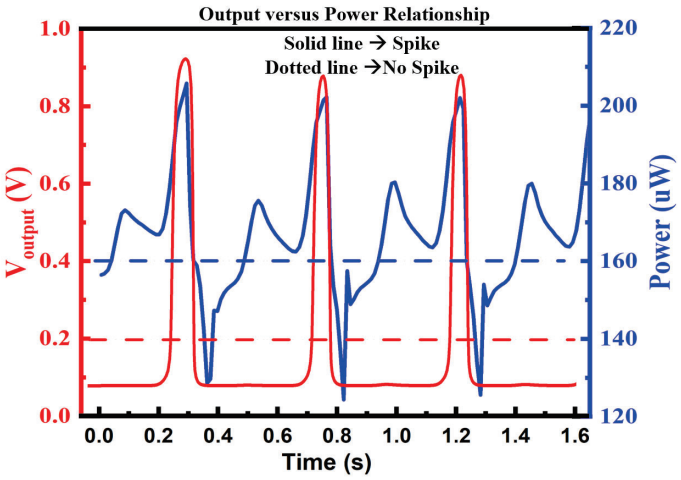

Fig. 4. Transient measurement results of the proposed printed spiking neuron.

where $\boldsymbol{x} \in \mathbb{R}^{B\times T}$ is input data series with batch size B, $\boldsymbol{y} \in \mathbb{R}^B$ denotes the corresponding classes, and $\boldsymbol{g}$ summarizes all the learnable conductances in the P-SNN. Subsequently, as all the operations in Eq. (5) are fully differentiable, the gradient serves for the parameter update can be calculated by $\frac{1}{T}\sum_{t=0}^{T} \nabla_{\boldsymbol{g}} L\left(\boldsymbol{x}_t, \boldsymbol{y}, \boldsymbol{g}\right)$. Consequently, gradient-based optimizers, such as Adam [26] and SGD [27], can be used for training.

IV. Evaluation

To evaluate the proposed P-SNN, we first designed the spike-generator circuit with synaptic inputs, implemented the proposed training framework[1] with PyTorch [28] and conducted a comparative study of P-SNN against the prior P-ANNs [10] and the benchmark SNNs [20].

A. Experiment Setup

1) Circuit Setup: The synapses and the spike-generator circuit (Fig. 3) were designed based on the well-developed n-EGT P-PDK [25]. We used Cadence Virtuoso[2] tool to simulate the output versus power relationship (Fig. 4) in SPICE.

2) Training Setup:

a) Datasets: We selected 13 benchmark datasets for the experiments, whose complexities and use cases suit the target application scenarios of PE. The datasets are split into training (60%), validation (20%), and test (20%) sets.

b) Training: We used the same topology #inputs-3-#outputs for all P-SNNs. We employed full batches for gradient calculations and used Adam [26] with default parameters. The initial learning rate was set to 0.1 and was halved once there was no improvement on the validation set over 100 epochs. Training was stopped when the learning rate dropped below 10^{-4}. To guarantee a sufficiently good solution, the aforementioned process was repeated 10 times (with random seeds ranging from 1 to 10). This aims to mitigate the potential impact of unfavorable initialization.

[1]Code available at https://github.com/Neuromophic/Printed_Spiking_NN.
[2]https://www.cadence.com/en_US/home.html

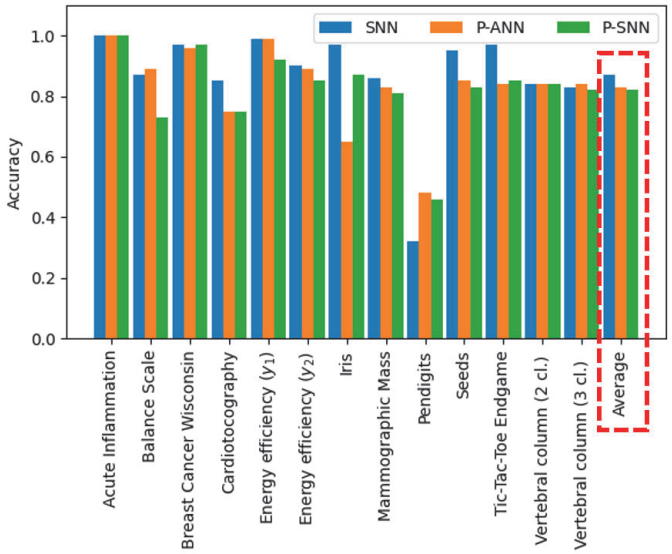

Fig. 5. Accuracy Comparison of SNN, P-ANN and P-SNN.

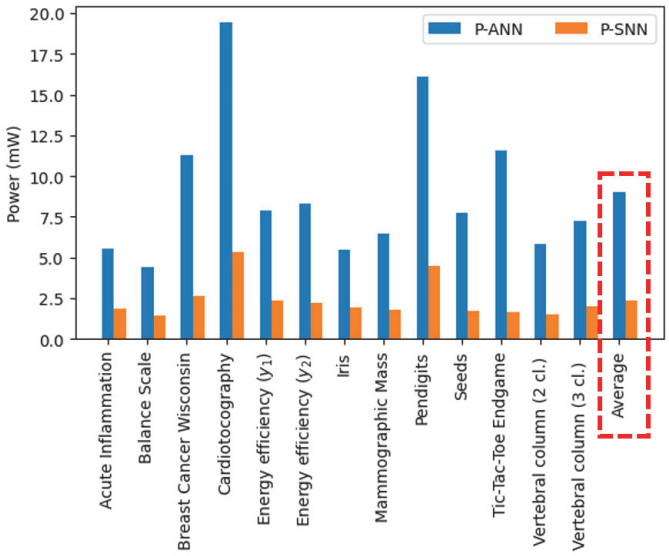

Fig. 6. Power Comparison of P-ANN vs P-SNN.

c) Baselines: We employed another two approaches as the baselines of P-SNNs to validate the major motivation of this work, that is, power saving. P-ANNs [10] with the same topologies as P-SNNs are trained on the corresponding datasets. Additionally, considering the target computing paradigm of this work, P-SNN is compared with its hardware-agnostic counterpart by conducting training on SNNs with the leaky-integration-fire mechanism [20].

3) Circuit Evaluation: The spike-generator circuit proposed in this work (Fig. 3) includes six transistors, six resistors, and four capacitors. The total area, power, and delay of the circuit have been found to be $750\mathrm{mm}^2$, $0.17\mathrm{mW}$ and $29.31\mathrm{ms}$ respectively.

The power consumption by the circuit is estimated through SPICE simulations. From Fig. 4, it is evident that, during the event of any spike, the total power consumed by the neuron is higher than that in no-spike condition. This indicates an elevated level of neural activity within the circuit during any spike events. Moreover, an increased power during spikes indicates increased synaptic activity and signal transmission in neurons, greater membrane polarization changes, and more energy consumption.

B. Results

1) Accuracy and Power: After training the baseline SNNs, P-ANNs, and the proposed P-SNNs, we selected the models with the lowest loss on the validation set, as they are the ones that would be printed. Note that, in accordance with the objective proposed in Sec. III, we computed the classification accuracy at every time step and subsequently averaged the accuracies over time to yield the overall classification accuracy of a dataset. These selected models were then evaluated on the test set. Finally, for each dataset, we summarized the mean accuracy w.r.t. random seeds and the corresponding power. The result is presented in Fig. 5 and Fig. 6. To get insights into the effectiveness of each model in various scenarios (datasets), we also averaged the accuracy and standard deviation w.r.t. target

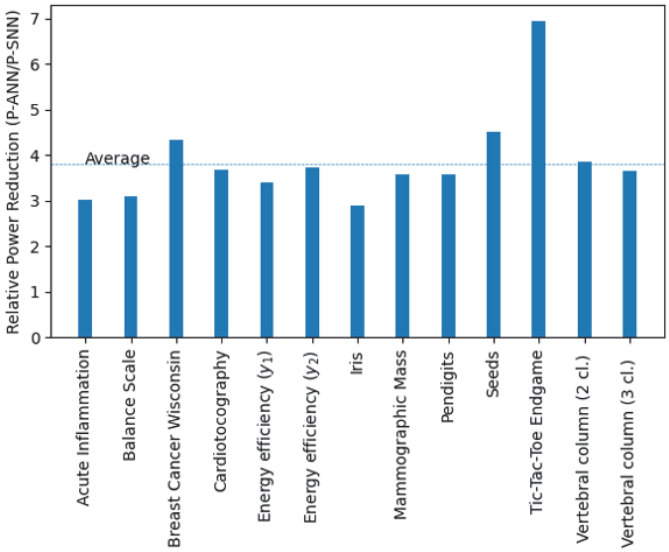

Fig. 7. Power Reduction from P-ANN to P-SNN.

tasks. The relative reduction in power for each of the datasets is reported in Fig. 7.

2) Hardware Cost: To investigate the additional hardware resources required by the spiking neuron circuit design, we collected the device counts and total power saving of both the previous P-ANNs and the proposed P-SNNs in different application scenarios (i.e., datasets). Analogously, we averaged the hardware costs across all datasets to provide a comparison regarding the hardware costs between the P-ANN and its P-SNN counterpart. The results can be seen in Tab. I.

C. Discussion

The comparative analysis among SNN, P-ANNs, and P-SNNs (illustrated in Fig. 5) reveals that P-SNNs exhibits a similar level of accuracy as SNN and P-ANNs. Across the 13 benchmark datasets, P-SNNs achieved an average accuracy only 1% lower than the established reference P-ANNs network. As for the pendigits dataset, all three neural networks

TABLE I
COMPARISON OF THE HARDWARE COSTS

Dataset	#Transistors		#Resistors		#Capacitors		#Total Device	
	P-ANN	P-SNN	P-ANN	P-SNN	P-ANN	P-SNN	P-ANN	P-SNN
Acute Inflammation	18	54	85	96	-	27	103	177
Balance Scale	14	42	79	77	-	21	93	140
Breast Cancer Wisconsin	24	72	118	129	-	36	142	237
Cardiotocography	48	144	268	264	-	72	316	480
Energy efficiency (y_1)	22	66	127	121	-	33	149	220
Energy efficiency (y_2)	22	66	131	121	-	33	153	220
Iris	14	42	83	77	-	21	97	140
Mammographic Mass	16	48	82	85	-	24	98	157
Pendigits	38	114	254	230	-	57	292	401
Seeds	20	60	107	110	-	30	127	200
Tic-Tac-Toe Endgame	24	72	116	129	-	36	140	237
Vertebral column (2 cl.)	18	54	81	96	-	27	99	177
Vertebral column (3 cl.)	18	54	100	99	-	27	118	180
Average	**23**	**69**	**126**	**126**	**-**	**35**	**149**	**228**

Device counts of Printed Artificial Neural Network (P-ANN) and proposed Printed Spiking Neural Network (P-SNN) on 13 benchmark tasks

consistently yielded low-accurate results, indicating comparable performance for the proposed P-SNNs.

In terms of power consumption, Fig. 6 shows the significant differences between P-SNN and P-ANNs, which is an improvement by approximately $3.86\times$ in power as depicted in Fig. 7. As the power represents the utilized energy rate, the proposed P-SNN demonstrates remarkable energy efficiency. This reduction is a consequence of the inherent sparsity of voltage activations within the network, aligning with the requirements of printed electronics, particularly in low-power applications where energy efficiency is a critical concern.

Conversely, the total number of devices, on average, increased by 50% when employing P-SNNs compared to P-ANNs due to a higher transistor count and the addition of capacitors, as shown in Tab. I. Although the area footprint expanded, P-SNNs achieved a significant reduction in power consumption due to their spiking nature. Also, the datasets of various sizes provide a wide range of spectrum to showcase the model's capability, tailored for PE applications.

V. CONCLUSION

This paper highlights the potential of merging printed manufacturing techniques and innovative algorithms in the field of PE and SNNs. In this work, we target the design of energy-efficient printed spiking neuromorphic circuits. By establishing a surrogate P-SNN model, the circuit can be explicitly incorporated into its design objective.

Despite the preliminary progress made in this work, other methodologies for circuit optimization, like variation-aware, noise-aware training, and learnable spike-generator, could be explored in future research: In this study, the synaptic weights are learnable. Nonetheless, in many design tasks, the circuits are constrained by predetermined energy budgets. Consequently, future work may enable explicit constraints on energy saving.

ACKNOWLEDGMENT

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

REFERENCES

[1] Z. Cui, *Printed electronics: materials, technologies and applications*. John Wiley & Sons, 2016.

[2] L. Meng, A. P. Turner, and W. C. Mak, "Soft and flexible material-based affinity sensors," *Biotechnology advances*, vol. 39, p. 107398, 2020.

[3] J. Misra and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress," *Neurocomputing*, vol. 74, no. 1, pp. 239–255, 2010, artificial Brains. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092523121000216X

[4] E.-I. Vatajelu, G. Di Natale, and L. Anghel, "Special session: Reliability of hardware-implemented spiking neural networks (snn)," in *2019 IEEE 37th VLSI Test Symposium (VTS)*, 2019, pp. 1–8.

[5] M. Eslaminia and S. L. Beux, "A shared synapse architecture for all-optical spiking neural networks," in *2023 21st IEEE Interregional NEWCAS Conference (NEWCAS)*, 2023, pp. 1–5.

[6] D. Hu, X. Zhang, Z. Xu, S. Ferrari, and P. Mazumder, "Digital implementation of a spiking neural network (snn) capable of spike-timing-dependent plasticity (stdp) learning," in *14th IEEE International Conference on Nanotechnology*. IEEE, 2014, pp. 873–876.

[7] M. S. Asghar, S. Arslan, and H. Kim, "A low-power spiking neural network chip based on a compact lif neuron and binary exponential charge injector synapse circuits," *Sensors*, vol. 21, no. 13, p. 4462, 2021.

[8] M. J. M. Hosseini *et al.*, "Organic electronics axon-hillock neuromorphic circuit: towards biologically compatible, and physically flexible, integrate-and-fire spiking neural networks," *Journal of Physics D: Applied Physics*, vol. 54, no. 10, p. 104004, dec 2020.

[9] V. Tischler, P. Dudek, J. Wijekoon, L. A. Majewski, Y. Takeda, S. Tokito, and M. L. Turner, "An integrate-and-fire neuron circuit made from printed organic field-effect transistors," *Organic Electronics*, vol. 113, p. 106685, 2023.

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

[11] H. Zhao *et al.*, "Aging-Aware Training for Printed Neuromorphic Circuits," in *IEEE/ACM, ICCAD '22*, 2022.

[12] H. Zhao, P. Pal, M. Hefenbrock, M. Beigl, and M. Tahoori, "Power-aware training for energy-efficient printed neuromorphic circuits," in *42nd IEEE/ACM ICCAD*, 2023.

[13] M. Allen, C. Lee, B. Ahn, T. Kololuoma, K. Shin, and S. Ko, "R2r gravure and inkjet printed rf resonant tag," *Microelectronic Engineering*, vol. 88, no. 11, pp. 3293–3299, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167931711006526

[14] J. S. Chang, A. F. Facchetti, and R. Reuss, "A circuits and systems perspective of organic/printed electronics: Review, challenges, and contemporary and emerging design approaches," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 1, pp. 7–26, 2017.

[15] A. Hazan and E. Ezra Tsur, "Neuromorphic neural engineering framework-inspired online continuous learning with analog circuitry," *Applied Sciences*, vol. 12, no. 9, 2022.

[16] B. Li *et al.*, "Build Reliable and Efficient Neuromorphic Design with Memristor Technology," in *24th ASPDAC Proceedings*, 2019, pp. 224–229.

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

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

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

[20] J. K. Eshraghian *et al.*, "Training Spiking Neural Networks Using Lessons from Deep Learning," *arXiv preprint arXiv:2109.12894*, 2021.

[21] C. Tang and J. Han, "Hardware Efficient Weight-Binarized Spiking Neural Networks," in *2023,(DATE)*. IEEE, 2023, pp. 1–6.

[22] A. Vaswani *et al.*, "Attention Is All You Need," *Advances in neural information processing systems*, vol. 30, 2017.

[23] J. Devlin *et al.*, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[24] L. Ouyang *et al.*, "Training Language Models to Follow Instructions with Human Feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[25] 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, 2019.

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

[27] S.-i. Amari, "Backpropagation and Stochastic Gradient Descent Method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[28] A. Paszke *et al.*, "Pytorch: An Imperative Style, High-performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.