

Bespoke Approximation of Multiplication-Accumulation and Activation Targeting Printed Multilayer Perceptrons

Florentia Afentaki^{*†}, Gurol Saglam[‡], Argyris Kokkinis[§], Kostas Siozios[§], Georgios Zervakis^{*}, Mehdi B. Tahoori[‡]

^{*}University of Patras, Greece, [§]Aristotle University of Thessaloniki, Greece, [‡]Karlsruhe Institute of Technology, Germany

^{*}{afentaki, zervakis}@ceid.upatras.gr, [§]{arkokkin, ksiop}@auth.gr, [‡]{guerol.saglam, mehdi.tahoori}@kit.edu

Abstract—Printed Electronics (PE) feature distinct and remarkable characteristics that make them a prominent technology for achieving true ubiquitous computing. This is particularly relevant in application domains that require conformal and ultra-low cost solutions, which have experienced limited penetration of computing until now. Unlike silicon-based technologies, PE offer unparalleled features such as non-recurring engineering costs, ultra-low manufacturing cost, and on-demand fabrication of conformal, flexible, non-toxic, and stretchable hardware. However, PE face certain limitations due to their large feature sizes, that impede the realization of complex circuits, such as machine learning classifiers. In this work, we address these limitations by leveraging the principles of Approximate Computing and Bespoke (fully-customized) design. We propose an automated framework for designing ultra-low power Multilayer Perceptron (MLP) classifiers which employs, for the first time, a holistic approach to approximate all functions of the MLP's neurons: multiplication, accumulation, and activation. Through comprehensive evaluation across various MLPs of varying size, our framework demonstrates the ability to enable battery-powered operation of even the most intricate MLP architecture examined, significantly surpassing the current state of the art.

Index Terms—Approximate computing, Electrolyte-gated FET, Multilayer Perceptron, Printed Electronics

I. INTRODUCTION

Printed electronics (PE) offer a promising solution for introducing computing and intelligence into various domains, including low-end healthcare products like smart bandages, disposables, packaged foods and beverages, smart packaging, in-situ monitoring applications and the vast market of fast-moving consumer goods (FMCG) [1]–[3]. These domains impose stringent demands for ultra-low cost (even sub-cent) and conformality, requirements that cannot be met by lithography-based silicon technologies. On the other hand, PE technology features negligible non-recurring engineering (NRE) costs, low equipment costs, and ultra-low fabrication cost [2]. Considering also the inherently supported features of conformality, flexibility, stretchability, non-toxicity, and porosity; PE technology is increasingly recognized as a key enabler for the Internet of Things as part of the “Fourth Industrial Revolution”, whose core technology advances are functionality and low-cost [4].

By PE we refer to a set of fabrication techniques that are based on printing processes that can realize ultra-low cost, large scale and flexible hardware [2]. PE does not challenge silicon-based electronics in integration density, area, or speed,

mainly due to their large feature sizes arising from low-cost and low-resolution printing. Typically, operating frequency of printed circuits ranges from a few Hz to only a few kHz [5] while their feature size tends to be several microns [6]. On the other hand, due to its form-factor, conformity, low cost, and on-demand, even at low-volume fabrication, PE can target application domains untouchable by silicon VLSI [7]. However, their large feature sizes and inherent high transistor gate capacitances result in increased power and area compared to nanometer technologies. Despite the appealing features of PE, such limitations make the realization of complex circuits, as machine learning (ML) classifiers that form the core task in most printed applications [7], very challenging.

As an attempt to mitigate the aforementioned limitations, the authors in [8] exploit the potential for high customization, originating by the low-fabrication and NRE costs of printed circuits and designed bespoke ML classifiers. The term bespoke refers to fully-customized circuit implementations, tailored to specific ML model and dataset. The bespoke designs of [8] achieved remarkable area and power savings that proved however insufficient towards the realization of complex ML circuits, such as Multilayer Perceptrons (MLPs). Thus, [8] focused only on simple ML models (e.g., Decision Trees). Targeting more complex printed ML circuits, the authors in [7], [9]–[11] employed Approximate Computing. Approximate computing for ML circuit design is gaining significant attention since by trading some loss in accuracy, it can achieve high gains in area and power [12], [13]. Though, their proposed approximations are limited in scope and do not exploit the full spectrum of Approximate Computing, resulting in conservative gains. Similarly, [14] used Stochastic Computing that resulted however in large accuracy degradation.

In this work, we propose an automated design framework that through means of bespoke design and approximation enables printed-battery powered MLP classifiers. Unlike the state of the art, we implement a holistic approximation that targets the core components of a MLP circuit: multiplication, accumulation, and activation function. Specifically, we use power-of-2 weight quantization to eliminate multiplications and quantized Relu to reduce the size of the outputs of the hidden layer. Moreover, we propose an accumulation approximation that through a genetic optimization reduces the number of summand bits in each accumulator. Finally, we

also approximate the activation function of the output layer by selectively comparing subsets of its inputs, decreasing, thus, the size of the comparators. Compared to the state-of-the-art exact baseline, our evaluation shows that, across six MLPs of varying complexity, our framework delivers more than 2.6x area and 8x power reduction for less than 5% accuracy loss.

Our novel contributions within this work are as follows:

- 1) This is the first comprehensive approximation framework¹ for printed MLPs circuits that apply a holistic approximation across all the MLP components: multiplication, accumulation, and activation function.
- 2) We propose an activation-aware accumulation approximation, customized for bespoke MLP circuits, that is applied through a multi-objective genetic optimization. Our proposed area model and accuracy evaluation of approximate printed MLP circuits enable fast and high-level exploration of the corresponding approximation space.
- 3) Our framework enables printed-battery powered operation of complex printed MLP circuits for up to 5% accuracy loss. Specifically, our framework surpasses the current state of the art by increasing the number of parameters that can be integrated into a printed MLP circuit by 20x.

II. PREAMBLE

A. Background on Printed Electronics

Moore’s law has been driving the lithography-based silicon VLSI technologies for higher integration density. However, such technologies are governed by a lower cost bound due to the expensive manufacturing, e.g., wafer processing, lithography, and material processing. This in turn increases the cost for testing, assembly, and packaging. PE technology, based on low-cost additive manufacturing technologies, has emerged as an alternative approach that is gaining popularity, especially targeting disposables and domains with ultra-low cost margins, particularly those with conformality requirements.

Printing technologies commonly utilize mask-less, portable, and additive manufacturing methods. Such methods can greatly reduce manufacturing costs and decrease production timelines [15]. PE rely on printing processes, such as jet printing, screen or gravure printing [16]. The simple additive manufacturing and the low equipment costs enable remarkably low-cost (even sub-cent) electronic circuits. Though, due to the large feature sizes that result in elevated device latencies and low integration density (orders of magnitude lower compared to silicon VLSI), PE cannot match the area and performance achieved by silicon systems. Nevertheless, in the target domains, the performance and precision requirements are typically very low, e.g., sampling rate of only a few Hz and few bits precision [13]. Such requirements could effectively be fulfilled by printing technologies under acceptable area and energy constraints. In this work we consider the Electrolyte-Gated FET (EGFET) technology that has good supply voltage and mobility characteristics, being thus good fit for battery-powered applications [2].

¹Our framework is available at <https://github.com/floAfentaki/Approximation-Techniques-Targeting-Printed-MLPs>

TABLE I
PRINTED MLP CIRCUITS STATE OF THE ART

Works	Bespoke	Approx. Multiplication	Approx. Addition	Approx. Activation
[8]	✓	✗	✗	✗
[9], [10]	✓	✓	✗	✗
[7]	✓	✓	✓	✗
[14]	✓	✗	✗	✓
Ours	✓	✓	✓	✓

B. Related Work

In recent years, the design of complex systems based on flexible technologies has gained vast research interest. Briefly, in 2020 Ozer et al., fabricated a processing system for odour detection on flexible electronics [17]. Weller et al., fabricated a neuromorphic circuit based on the flexible EGFET technology that operates at 1V [18]. Similarly, in 2021, ARM fabricated the first 32-bit processor on a flexible plastic technology [19]. In 2023 PragmatIC fabricated ML classifiers with a low area and power footprint on polyamide substrate using the 0.8 μ m FlexIC TFT technology [20]. However, these works do not leverage the hardware-efficiency of approximate computing while [17], [19], [20] do not consider a printed technology.

Design methodologies that aim to shrink the size of neural networks and deploy them on FPGAs at the deep edge are introduced in [21]–[23]. Although FPGAs support bespoke designs, they feature orders of magnitude higher computing capabilities compared to PE. Approximating arithmetic blocks for Deep Neural Networks have also been suggested as candidate solutions for the generation of low-power DNNs [12]. Nevertheless, those methodologies target conventional, non-bespoke implementations and therefore are not suitable for printed applications.

Targeting specifically printed ML classifiers, the authors in [7]–[10], [24] consider bespoke implementations. [8] does not leverage approximate computing while [8] and [24] mainly consider simple classifiers. [9] introduced approximate printed ML circuits but approximated only the multiplications and then applied a generic gate-level pruning approximation. In [10] the authors extend [9] by applying also voltage over-scaling. The authors in [11] evaluate the impact of neural compression on printed MLPs but they presented only preliminary results. Finally, [7] approximates both the multiplication and accumulation. However, [7] applied only coarse-grain truncation on the accumulators, limiting thus the potential gains. Table I summarizes relevant works on printed MLP circuits. Besides approximate computing techniques, stochastic computing has been suggested as a candidate approach to mitigate the excessive area and power overhead [14]. Although the stochastic schemes can yield significant area and power gains, they may also result in a high degradation in the classifier’s accuracy [14] and potentially increased classification latency.

Our work differentiates from the state of the art as it combines the bespoke design paradigm along with a holistic approximation approach that considers approximate multipli-

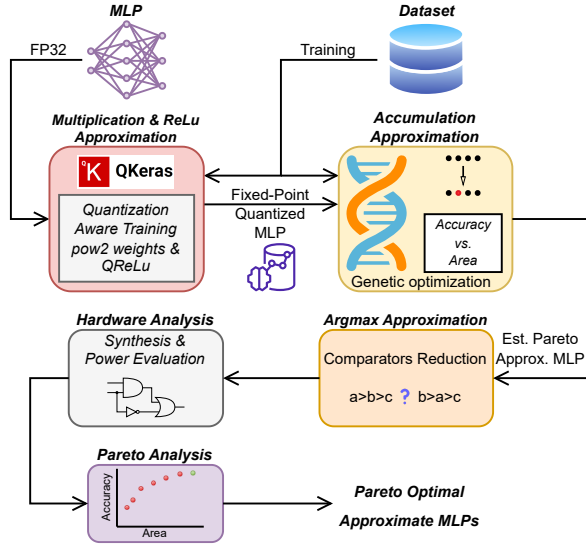


Fig. 1. Overview of our proposed framework.

cation, accumulation, and activation.

III. PROPOSED FRAMEWORK

This section presents our approximation framework (Fig. 1) which aims to minimize the area-overhead of a printed MLP circuit while maintaining high accuracy. Our framework takes as inputs a trained MLP model and the corresponding train and test datasets. Without loss of generality, if the MLP is not pre-trained, it can be trained as described in [7], [8] and Section III-A. Operating in a fully automated manner, our framework produces a set of area-accuracy Pareto-optimal printed MLP circuits by employing bespoke design and a holistic approximation approach that encompasses the approximation of all components within a MLP neuron, i.e., the multiplication, accumulation, and activation circuits. Sections III-B to III-D describe the approximations applied by our framework while Section III-E describes its overall flow.

A. Preliminaries

The baseline MLPs considered in our work use the same topology as in [7], [8] in order to enable fair comparisons in Section IV. The datasets are obtained from the UCI ML repository [25]. We train the MLPs using scikit-learn and the randomized parameter optimization with 5-fold cross validation. Similar to [8], [14], the inputs are normalized to $[0, 1]$ and we randomly split the datasets to 70%/30% train/test. The Relu activation function is used in the hidden layer.

For the design of the corresponding printed MLP circuit, either approximate or accurate, we employ the efficiency of bespoke design paradigm [8]. For each MLP a fully parallel architecture (i.e., one inference per cycle) is implemented in which the weight values are hardwired in the circuit. In addition, we follow design optimizations of [7], [11]. Since we implement a bespoke design and the input activations of all neurons are positive (e.g., Relu), we split the weights of

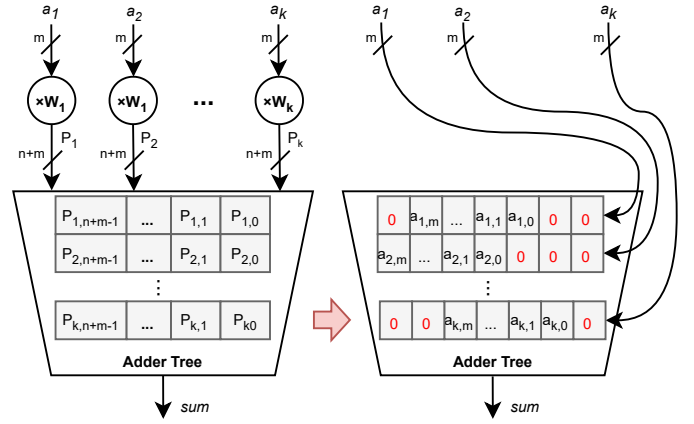


Fig. 2. Showcase of the impact of power-of-2 weights on a bespoke MAC circuit. On the left bespoke multipliers and a generic adder tree are used. With power-of-2 weights, only a simpler and narrower adder tree is required.

each neuron to positive and negative ones. For the negative ones the absolute value is used. The respective products are accumulated separately, i.e., two distinct accumulators are used. Finally, the two obtained sums are subtracted. As a result, we almost completely avoid signed arithmetic and the associated hardware overhead of sign-bit extension, etc.

Finally, we truncate the inputs of the MLP down to 4 bits. An input size of 4 bits is small enough and doesn't result in any accuracy degradation across all the examined datasets.

B. Multiplication Approximation

The multipliers within a neuron consume the largest part of its area [12]. Though, in bespoke circuit design (as in our work), the coefficients of a machine learning model, such as the weights of MLPs, are hardwired in the circuit description. Consequently, the area overhead of a neuron's multipliers is strongly influenced by the values of its weights. In an effort to leverage this, the state-of-the-art [7], [9], [10] explored custom approaches that replace the MLP weights with more hardware-friendly values (i.e., weights that instantiate smaller bespoke multipliers). However, even with these modifications, the resulting circuits still require multiplications and the associated hardware overheads remain prohibitive.

This observation has motivated us to consider that the elimination of the multiplication operation is mandatory towards realizing complex printed MLP classifiers. To achieve this, we replace the weights with power-of-2 values. Since the weights are hardwired in the circuit, power-of-2 weights transform every multiplication to simply wiring. Thus, the area of multipliers is nullified. An illustrative example is depicted in Fig. 2. As shown, in Fig. 2 not only the multipliers are removed from the neuron but also a semi-bespoke adder tree is required for the accumulation since several summand bits in the tree (see Fig. 2 left) are replaced by constant zero values (see Fig. 2 right). Power-of-2 weight representation has been widely explored to improve ML inference performance but in many cases it is not preferred since it may incur an unacceptable accuracy loss. However, it is important to

acknowledge that in printed applications, the feasibility of a design is the primary concern and takes precedence over achieving the utmost classification accuracy [7], [8].

We utilize power-of-2 quantization to transform the MLP’s weights into powers of two. Our framework uses Google QKeras [26], a tool specifically designed for such purposes, and utilizes its power-of-two (po2) quantizer. The weight size is set to 8 bits, which is a commonly used size in neural network quantization [12]. Biases are also quantized along with the weights in a similar manner. We perform Quantization aware re-training (QAT) using Qkeras to effectively recover any tentative accuracy degradation due to the po2 quantization. Note that the MLP models considered for printed applications are fairly small in size [8] (compared to contemporary Deep Neural Networks) and as a result QAT requires only few retraining epochs, even for the most complex printed MLPs.

C. Activation Approximation

To minimize hardware cost, printed MLPs are trained with a single hidden layer. Similarly, we use the Relu activation function at the hidden layer and Argmax at the output layer.

1) *QRelu*: While Relu implementation only requires a few AND gates, it should be noted that Relu is unbounded and produces large bit-width outputs as the accumulation of the weights-input activations products is performed in full precision. Consequently, this results in a significant area overhead at the neurons of the subsequent layer, as they operate over inputs with a large bit-width. To mitigate this overhead, our framework employs quantized Relu (QRelu) where the quantization size is set again to 8 bits. For hardware efficiency, linear QRelu with truncation is used. The circuit complexity of QRelu remains insignificant since it requires only a few AND gates for nullification and a few OR gates for clipping. To retain high accuracy, we incorporate the activation quantization (QRelu) of the hidden layer in QAT.

2) *Approximate Argmax*: Argmax is the activation function of the output layer and is implemented as a tree of comparators determining the neuron with the highest value. Typically, the comparators compare the outputs of the neurons in the order they appear, i.e., 1st neuron with 2nd neuron, 3rd neuron with 4th neuron, etc. However, we observe that there are correlations between the neurons’ outputs. For example, we observe that, in most cases, when neuron e has the maximum output, neuron e and neuron z feature so close values that only a few LSBs might be sufficient for an accurate-enough comparison. Similarly, when neuron f has the maximum output, the difference of neuron f and neuron g is so high that only a few MSBs suffice for a good-enough comparison.

Given this potential for decreasing the size of the required comparators, we approximate the Argmax activation by identifying the appropriate order of comparisons as well as the minimum subset of bits that need to be compared each time. First, for each neuron i and each neuron j we compare them in an approximate way while the rest comparisons are performed accurately. For the approximate comparison we employ a greedy approach to extract the minimum subset of bits that

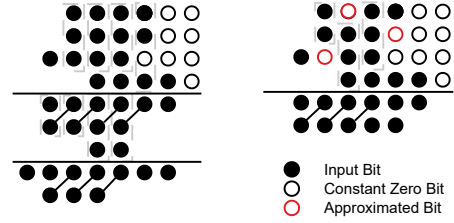


Fig. 3. Example of our implemented accumulation approximation.

need to be compared so that the classification accuracy (on the train dataset) remains almost the same (i.e., does not drop more than 0.5%). Our greedy approach is straightforward: it starts from the MSB and decides if the corresponding bit should be kept or discarded based on the accuracy obtained without that bit. After this procedure is over $\forall i, j$, we fill a 2-D matrix that contains the minimum set of bits that will be kept for each comparison. Finally, we use the Hungarian algorithm [27] to select the combination (i.e., which (i, j) will be compared each time) that gives the lowest cost (i.e., smallest number of bits to be compared in total). Each i, j can be selected only once. The Hungarian algorithm is commonly used in assignment problems such as is our case. Overall the size of the matrix is fairly small (up to 16×16 for the examined MLPs) so the algorithm advances very fast. The above procedure is repeated for all subsequent (few) comparison stages.

D. Accumulation Approximation

After QAT, the weights are in power-of-2, the input activations of each layer exhibit reduced size and semi-bespoke adder trees are required for the accumulation. Next, to further improve hardware efficiency, our framework approximates the accumulation operation by selectively removing certain summand bits from the adder trees. Removing a summand bit is equivalent to replacing it by a constant zero in the hardware description of the MLP. Hence, unlike custom arithmetic approximations that mainly alter the circuit’s logic [28], we fully leverage the IPs and optimization capabilities of the EDA synthesis tool, which among others includes constant propagation, to optimize the obtained circuit even further.

A descriptive example of our accumulation approximation is illustrated in Fig. 3. In this example, the summation of four 4-bit operands is showcased. The black dots represent input bits, the values of which are not known beforehand, while the white dots indicate zero values resulting from the multiplication by the constant power-of-2 weights. As shown in Fig. 3, the exact addition requires 6 full-adders, 2 half-adders, and necessitates three accumulation stages. In contrast, by selectively removing only three bits (out of 16), the approximate adder tree reduces the hardware requirements to 2 full-adders, 1 half-adder, and eliminates one accumulation stage (i.e., delay gain as well).

The state-of-the-art arithmetic circuit approximation approaches typically focus on approximating a few least significant bits (LSBs) until a certain accuracy threshold is reached [28]. However, this intuitive approach may not always

be applicable in our specific case due to the QRelu activation. As described in Section III-C1, the hidden layer uses QRelu that truncates certain LSBs of the accumulation result and also applies clipping to a maximum value. Hence, in that case the middle bits might be more significant than the higher order bits. Moreover, due to QRelu (i.e., non-linear function) the impact of removing each bit on the final classification accuracy becomes intricate and challenging to model. Additionally, the gains of removing a specific bit also depend on its location (column in Fig. 2). Although removing different bits from the same column may lead to similar hardware gains, this does not necessarily hold true for the classification accuracy due to the different distributions of each input that may affect the probability of a removed bit being zero or one.

As a result, to minimize the hardware overheads of a printed MLP circuit, our framework needs also to identify, for each adder tree in the MLP circuit, which bits shall be removed. However, our framework must maximize the area gains while also preserving high classification accuracy. To address this optimization problem we employ a Genetic Algorithm due to its inherent parallelism and its ability to effectively explore the solution space. Though, other heuristics or optimization techniques may also be employed.

Overall, our accumulation approximation differs from traditional arithmetic approximation [28] in several ways. Firstly, our method is activation-aware as it accounts the configuration of QRelu to identify the more/less important columns for approximation. Additionally, our approach considers the accuracy of the entire MLP, capturing dependencies or synergies of different approximations. Furthermore, it is input-aware, as the distribution of each input plays a crucial role in our approach. For instance, unlike the state-of-the-art arithmetic approximation [28], our approach does not consider different bits in the same column as equivalent for approximation.

1) *Genetic Optimization*: In our framework, each candidate approximate solution for the accumulation approximation is represented by a set of integers (which we refer to as a “chromosome” from further on) in order to facilitate easy manipulation during the optimization process. A tentative approximate solution includes all summand bits of each adder tree² in the MLP which can be either removed (value 0) or not-approximated (value 1):

$$\text{Candidates} = \{(b, v) : v \in \{0, 1\}, \forall b \in \text{AddTree}, \forall \text{AddTree} \in \text{MLP}\}. \quad (1)$$

For example, possible b values are the $a_{i,j}$ in Fig. 3, and can be represented by the tuple (i, j) for the specific tree.

To traverse the design space we use the multi-objective Non-dominated Sorting Genetic Algorithm II (NSGA-II) [29]. NSGA-II receives the approximation candidates to generate approximate solutions and evaluate them. Targeting to incentivize the exploration of solutions with high accuracy at the initial stages of evolution, we create an initial population

of semi-random chromosomes that are biased towards non-approximated summand bits. Our optimization targets two objectives: classification accuracy and area overhead. Thus, the obtained approximate solutions will exhibit the most dominant combination of accuracy-area trade-off. Additionally, we set an upper bound of 15% at the accuracy loss to discourage the exploration of solutions with unacceptably low accuracy. Finally, we apply random mutation to the generated chromosomes.

Evaluating the accuracy and area of each candidate solution would typically involve generating its corresponding HDL description and use EDA tools to perform synthesis to get the area and simulation to get the accuracy of the approximate MLP. However, given the large number of approximate solutions that need to be evaluated in each iteration and the potential licensing constraints of EDA tools, this approach can adversely affect the parallelism and performance of our optimization process or even render it infeasible. To address this challenge, we employ two high-level methods for evaluating accuracy and estimating area of each approximate solution.

2) *High-level Accuracy Evaluation*: Obtaining the accuracy of a MLP with our accumulation approximation can be easily implemented at a high-level. To accomplish this, we have developed a custom MLP classifier class that utilizes the pairs (b, v) from the chromosomes to mask the summands (if a bit is removed the corresponding mask bit is zero). A bitwise AND between each mask³ and summand is performed and then addition is just computed on the masked summands. Weights and inputs are by definition in fixed point representation (quantized inputs and weights) enabling our masking approach.

3) *High-level Area Estimation*: Evaluating the area, on the other hand, without synthesis is more complex. Therefore, we employ a surrogate model to estimate the area overheads of an approximate candidate solution. After QAT the multipliers are removed from the circuit and thus, the adders mainly contribute to the overall MLP’s area. Hence, estimating the area of the adder trees can provide a good enough estimation of the overall MLP area. To achieve this, we assume carry-save operation and for each adder tree in the MLP we count the full-adders (FAs) required for reduction stage. In other words, for each column in the tree we need to calculate how many FAs are required to reduce the number of summand bits in that column down to two. Note that a full adder is a 3-to-2 compressor but one of its outputs is of higher order. Hence, if L is the number of non-zero bits in a column $\lceil \frac{L-2}{2} \rceil$ FAs are required. However, we need also to account for the carries coming from the column to the right (each FA gives a carry). Therefore, for a column k , the number of FAs required are:

$$\text{FA}_k = \lceil \frac{L_k + \text{FA}_{k-1} - 2}{2} \rceil, \quad \text{FA}_{-1} = 0. \quad (2)$$

Note that L_k can be easily obtained from the (b, v) values of the corresponding chromosome. Hence, the total number of

²Each neuron uses two adder trees, one for the “positive” and one for the “negative” products accumulation (see Section III-A).

³The masks are also shifted w.r.t. to the weight values to align the summand and the mask.

TABLE II
SPEARMAN’S RANK CORRELATIONS OF OUR AREA ESTIMATOR

Dataset	Spearman’s Rank Correlation
Arrhythmia	0.96
Breast Cancer	0.96
Cardio	0.99
Pendigits	0.99
RedWine	0.96
WhiteWine	0.98
Average	0.97

FAs is estimated by:

$$\begin{aligned} \text{FA}_{AddTree} &= \sum_{\forall k} \text{FA}_k \\ \text{and } \text{FA}_{MLP} &= \sum_{\forall AddTree} \text{FA}_{AddTree}. \end{aligned} \quad (3)$$

Our area model assumes only full-adders and no half-adders. Moreover, it does not consider a specific reduction strategy (e.g., Wallace, Dadda etc.) that might affect the total number of FAs. However, for our optimization we do not need an area model that precisely captures the area of an approximate MLP. We just need a surrogate model that captures accurately enough, the relative order of different accumulation approximated MLPs in order to guide our genetic algorithm towards more area efficient solutions. In Table II, we evaluate the Spearman’s rank correlation of our area estimator. For each MLP considered (see Section IV), we run QAT and then we randomly create 1000 chromosomes and generate the respective approximate MLP circuits (HDL description). We synthesize the obtained circuits with the EDA tool and measure their area. Finally, we use our area model to estimate the area of the respective MLP-chromosome combination and calculate the corresponding Spearman Correlation across all designs. In total, 6000 designs are synthesized for the evaluation of Table II. As shown, our area estimator features almost perfect correlation and thus it is able to efficiently drive our optimization search. Specifically, it achieves more than 0.96 Spearman correlation while its average value is 0.97.

Overall, our high-level accuracy evaluation and area estimation enable fast exploration of the associated design space. At the worst case (i.e., Arrhythmia MLP), our genetic optimization requires only 3h. The experiments are conducted on an AMD EPYC 7552 with 256GB RAM. The population size is set to 1000 and our genetic run for 30 iterations.

E. Holistic Approximation Flow

In this section, we describe the flow of our framework towards implementing a holistic approximation across all the core components of a MLP. Overall, as shown in Fig. 1, our framework operates as follows. First, applies QAT on the given MLP to approximate the multiplications (power-of-2 weights) and the activations of the hidden layer (QRelu). Our accumulation approximation is then applied through the described genetic optimization. The output of this phase is a set of

estimated area-accuracy Pareto-optimal approximated printed MLPs. Next, for each circuit, our framework approximates the activation function of the output layer (Argmax approximation). This step is performed last since it leverages and depends on the distribution of the outputs of the output neurons. The obtained approximation configurations are translated in HDL description and then a hardware analysis is performed on the obtained circuits. Finally, a Pareto analysis is performed to extract the designs with the best accuracy-area trade-off. In our framework all optimizations are performed on the train dataset while the test dataset is used only for the final assessment of the obtained Pareto-optimal approximate designs.

IV. RESULTS AND EVALUATION

In this section, we present a comprehensive evaluation of our framework. First, we analyze the area-efficiency of our implemented approximations. Then, we compare our framework against the current state-of-the-art printed MLPs [7], [8], [10], [14]. Finally, we evaluate the effectiveness of our framework on enabling printed-battery powered MLP classifiers. We consider the Cardiotocography, Pendigits, Red Wine, White Wine, Arrhythmia, and Breast Cancer datasets as in [8], [14]. Synopsys Design Compiler S-2021.06 and VCS T-2022.06 are used for circuit synthesis and simulation respectively, while PrimeTime T-2022.03 is used for circuit simulations. All circuits are mapped to the open-source printed EGFET library [2]. The accuracy numbers reported hereafter regard the test dataset while all designs have been synthesized at a relaxed clock period to improve even further area efficiency. Specifically, to align with the state of the art, we consider 200ms for all the datasets except for Pendigits and Arrhythmia that require 250 and 320ms, respectively. Note that such low clock frequencies are in compliance with typical printed electronics performance [5]. Hereafter, as baseline circuits we refer to the exact bespoke MLP circuits that use 8-bit fixed point weights and 4-bit inputs and are designed as in [8].

A. Evaluation of Our Framework

Table III presents the topology of each MLP and reports the hardware requirements of the baseline printed MLPs. As shown, the baseline MLPs feature unbearable area overheads (71cm² on average) that prohibit realistic application. Moreover, their power consumption is so high that none of the examined MLPs can be powered by an existing printed power source [8]. In Table III, we also present the respective values when we apply QAT only, i.e., eliminate the multipliers with power-of-2 weight quantization and use of QRelu. As shown, compared to the baseline, when applying QAT the accuracy loss is 1.25% on average and goes up to 4.4%. On the other hand, for this small accuracy loss, the area gains range from 2.5x up to 5x and the power savings are from 2.5x up to 5.5x. Still, despite the impressive gains, the area remains relatively high for most MLPs, while only Breast Cancer and Red Wine can be powered by a printed battery (e.g., Molex 30mW).

Next, we assess the effectiveness of our accumulation approximation in further reducing the area of printed MLPs.

TABLE III
EVALUATION OF BASELINE AND POWER-OF-2 QUANTIZED PRINTED MLPs

MLP	Topology ¹	Baseline			QAT Only		
		Acc ²	Area (cm ²)	Power (mW)	Acc ²	Area (cm ²)	Power (mW)
Arrhythmia	(274,5,16)	0.620	266	998	0.610	92.5	258
Breast Cancer	(10,3,2)	0.980	12.0	40.0	0.965	4.6	16.6
Cardio	(21,3,3)	0.881	33.4	124	0.884	8.8	34.1
Pendigits	(16,5,10)	0.937	67.0	213	0.893	19.5	77.3
RedWine	(11,2,6)	0.564	17.6	73.5	0.568	3.4	13.7
WhiteWine	(11,4,7)	0.537	31.2	126	0.524	8.1	31.3

¹ MLP topology. ² Accuracy.

TABLE IV
EVALUATION OF ARGMAX APPROXIMATION.

MLP	Avg. Accuracy Loss ¹	Avg. Area Reduction ¹	Avg. Comparator Size Reduction ¹
Arrhythmia	0.007	12%	11.4x
Breast Cancer	-0.008	21%	4.8x
Cardio	-0.001	16%	6.1x
Pendigits	0.000	9%	4.0x
RedWine	0.007	7%	11.0x
WhiteWine	0.002	18%	8.9x

¹ Values calculated over the respective QAT & Approximate Accumulation MLP (i.e., green points in Fig. 4).

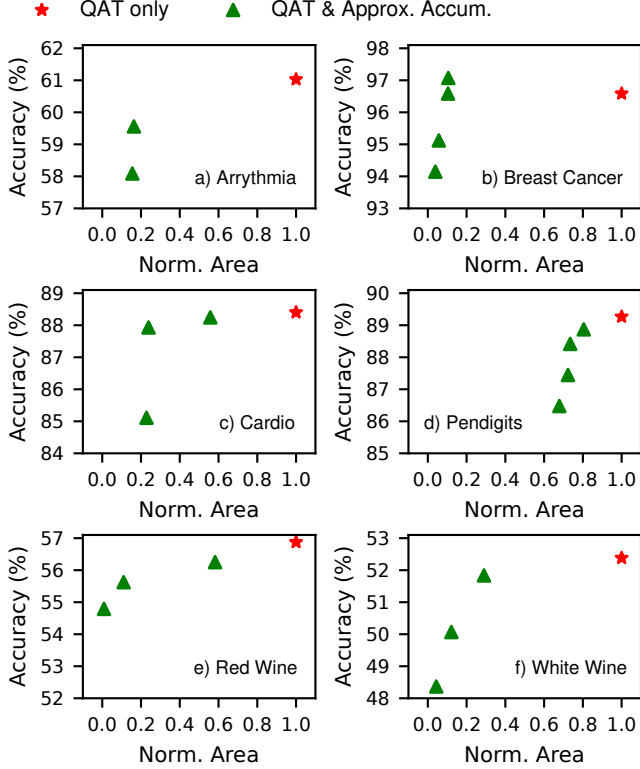


Fig. 4. Evaluation of the effectiveness of our accumulation approximation. Area is normalized w.r.t. the corresponding QAT-only approximate MLP.

To do this, we execute our framework without applying the Argmax approximation step. Fig. 4 illustrates the Pareto-front of the obtained designs (i.e., designs that apply QAT & accumulation approximation). The area value is normalized w.r.t. the area of the corresponding QAT-only design. Designs with up to 5% accuracy loss w.r.t. the corresponding QAT-only MLP are depicted in Fig. 4. As shown, compared to QAT-only, our accumulation approximation achieves 24x area reduction on average for less than 2% lower accuracy. At the worst case (Pendigits at 1% lower accuracy), our approximate accumulation reduces the area by 1.3x. Fig. 4 demonstrates that applying our accumulation approximation on top of QAT delivers a substantial improvement in area efficiency, without significantly compromising the accuracy of the printed MLPs.

Finally, we examine the additional area reduction that can be achieved when considering also our Argmax approximation.

After eliminating the multiplications and approximating the accumulations, Argmax might occupy a considerable part of the overall printed MLP circuit. Table IV presents the impact of applying Argmax approximation on the green points of Fig. 4, i.e., designs that apply QAT and accumulation approximation. On each design in Fig. 4 (each green point), we apply our Argmax approximation and compute the area reduction and accuracy loss w.r.t. the initial MLP (green point). Table IV presents the average area reduction and accuracy loss for each case. Moreover, Table IV evaluates the efficacy of our Argmax approximation in decreasing the size of the required comparators. Indicatively, if the initial MLP requires 16-bit comparators while the Argmax-approximated requires 4-bit comparators, on average, the achieved average comparator size reduction is a 4x. As shown, our Argmax approximation reduces the size of the required comparators by 7.6x on average. In terms of area, applying our Argmax on the QAT & approximate accumulation MLPs, reduces the area by an additional 14% while the additional accuracy drop is 0.1%.

Overall, the above analysis demonstrates that only when applying our holistic approximation we can minimize the area of a printed MLP. It is noteworthy that applying the state-of-the-art power-of-2 quantization alone is insufficient to enable battery-powered operation of printed MLPs; therefore, our additional approximations (such as the accumulation approximation) are essential in achieving this objective.

B. Comparison Against the State of the Art

In this section we present a comparative study of our framework against the state-of-the-art works [7], [10], [14]. For our framework all approximations are applied, i.e., multiplication, accumulation, and activation approximation. Fig. 5 presents the area and power comparison. All values in Fig. 5 are normalized over the corresponding value of the respective exact bespoke design [8]. For our circuits and [7], [10], targeting high area efficiency and reasonable accuracy drop, we consider up to 5% accuracy loss compared to the baseline [8]. It is important to reiterate that feasibility is the primary requirement for printed ML circuits, prioritizing it over strict accuracy constraints. Though, [14] cannot achieve such high accuracy. The average accuracy loss of [14], for the respective MLPs, is 35%. In addition, note that our MLPs, [7], [10], and [14] achieve almost identical performance. Our MLPs and [7], [10]

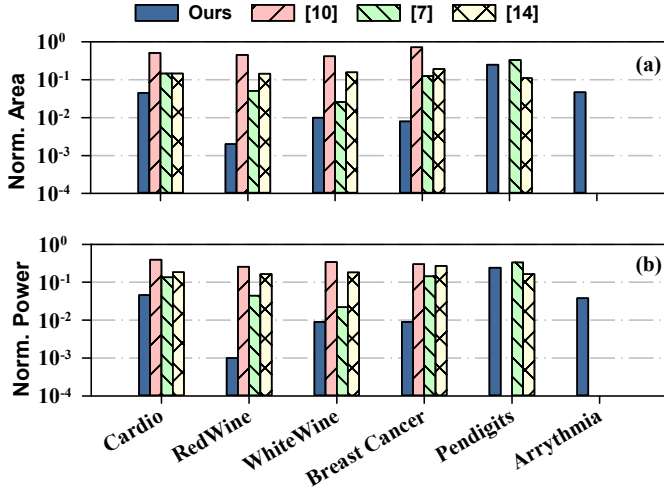


Fig. 5. (a) Area and (b) power gains of the MLPs generated by our framework compared to state-of-the-art [10], [7] and [14]. All the MLPs feature a 5% accuracy loss from our baseline [8]. Values are normalized w.r.t. [8]. Y-axis is in logarithmic scale.

produce one inference result per 200ms (250ms for Pendigits). The MLPs of [14] require 220-230ms per inference since they use a stochastic bitstream of length 1024.

As shown in Fig. 5, our framework significantly outperforms [7], [10] and [14]. Specifically, compared to [7], our MLPs achieve 10x lower area and 12.5x lower power on average. Similarly, compared to [10], our MLPs achieve 96x lower area and 86x lower power on average. Finally, our MLPs deliver 9x and 11x area and power saving, respectively, compared to [14]. As shown in Fig. 5, [7], [10], [14] do not consider the Arrhythmia MLP, most probably due to its increased complexity. As a result, for fairness, the reported average gains exclude Arrhythmia. Still, our framework achieves very high power and area reduction even for Arrhythmia. Similarly, [10] did not consider Pendigits either. It is noteworthy that our framework demonstrates superior area and power efficiency compared to [7], [10] and [14] across all but one MLPs. Only for Pendigits the stochastic MLP of [14] achieves slightly lower power and area than our approximate MLP. Though, [14] achieves only 22% accuracy while we achieve 89.6%.

C. Printed-Battery Operation

Finally, we evaluate the effectiveness of our framework in generating battery-powered printed MLP classifiers. Again, we consider the accuracy loss constraint of 5% compared to the baseline [8] and report in Table V the hardware requirements of the Pareto-optimal circuits generated by our framework that satisfy this constraint. In Sections IV-A and IV-B, for fair comparisons, we considered a voltage supply of 1V for all our circuits. However, our approximate MLPs are significantly faster than their exact baseline due to the applied approximations (e.g., multiplication elimination, shorter adder trees, etc.). As a result, we can decrease the supply voltage of our approximate circuits to achieve even higher power gains. Considering that EGFET printed circuits can operate even at 0.6V [30] and that printed batteries are customizable in

TABLE V
EVALUATING THE BATTERY OPERATION OF OUR PRINTED APPROXIMATE MLP CIRCUITS FOR 5% ACCURACY LOSS THRESHOLD.

MLP	Accuracy	Our Approximate MLPs			
		Area (cm^2)	Power (mW)	Area Reduction ¹	Power Reduction ¹
Arrhythmia	0.588	13.51	12.80	20x	78x
Breast Cancer	0.961	0.08	0.08	150x	500x
Cardio	0.851	1.35	1.57	25x	79x
Pendigits	0.896	25.15	26.60	2.6x	8x
RedWine	0.548	0.03	0.02	587x	3675x
WhiteWine	0.501	0.25	0.25	125x	506x

¹ With respect to the corresponding bespoke exact baseline [8].

terms of polarity, voltage, shape, etc., [31], we set the voltage supply of our approximate MLPs to the minimum supported value, i.e., 0.6V, and re-synthesize our designs. All of our approximate printed MLPs, except for Pendigits, meet the corresponding timing requirement at 0.6V without any issues. Due to the smaller delay gain of the approximate Pendigits (20%), re-synthesizing it targeting the 0.6V library, resulted in a larger circuit (in order to meet the timing requirement) but halved its power consumption also. As shown, in Table V all our approximate MLPs can be powered by a printed battery. Arrhythmia and Pendigits can be powered by a Molex 30mW battery, White Wine and Cardio by a Blue Spark 3mW battery, while Breast Cancer and Red Wine can be powered by only a printed energy harvester. Our MLPs achieve on average 151x lower area and 808x lower power compared to the baseline [8]. Table V highlights the effectiveness of our framework. Our framework enables battery operation of a printed MLP that features 1,450 parameters (weights). The largest MLPs that can be powered by the state of the art within a reasonable accuracy loss of 5% are the White Wine and Cardio that both feature only 72 parameters [7]. Therefore, our framework increased the size of the largest supported MLP by 20x.

V. CONCLUSION

With its distinctive characteristics, printed electronics technology emerges as a highly promising solution for introducing computing and intelligence to application domains that have yet to experience significant integration of computing. This includes the expansive market of fast-moving consumer goods, low-end healthcare products, and disposables, among others. Though, the large feature sizes in printed electronics hinder the realization of complex circuits. In this work, we tackle this issue and present an automated framework for generating printed MLP circuits. Our framework combines the bespoke design paradigm along with a holistic approximation across all the MLP components. Our evaluation shows that our framework advances the state of the art by enabling printed-battery operation of MLP circuits with 20x more parameters.

ACKNOWLEDGMENTS

This work is supported by the funding programme “MEDICUS” of the University of Patras and by the European Research Council (ERC).

REFERENCES

- [1] J. Isohanni, "Use of functional ink in a smart tag for fast-moving consumer goods industry," *Springer Journal of Packaging Technology and Research*, vol. 6, pp. 187–198, 2022.
- [2] N. Bleier, M. Mubarik, F. Rasheed, J. Aghassi-Hagmann, M. B. Tahoori, and R. Kumar, "Printed microprocessors," in *Annu. Int. Symp. Computer Architecture (ISCA)*, jun 2020, pp. 213–226.
- [3] P. Lacy, J. Long, and W. Spindler, "Fast-moving consumer goods (fmcg) industry profile," in *The Circular Economy Handbook*. Springer, 2020.
- [4] 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.
- [5] G. Cadilha Marques *et al.*, "Digital power and performance analysis of inkjet printed ring oscillators based on electrolyte-gated oxide electronics," *Applied Physics Letters*, vol. 111, no. 10, p. 102103, 2017.
- [6] T. Lei *et al.*, "Low-voltage high-performance flexible digital and analog circuits based on ultrahigh-purity semiconducting carbon nanotubes," *Nature communications*, vol. 10, no. 1, p. 2161, 2019.
- [7] G. Armeniakos, G. Zervakis, D. Soudris, M. B. Tahoori, and J. Henkel, "Co-design of approximate multilayer perceptron for ultra-resource constrained printed circuits," *IEEE Transactions on Computers*, pp. 1–8, 2023.
- [8] M. H. Mubarik *et al.*, "Printed machine learning classifiers," in *Annu. Int. Symp. Microarchitecture (MICRO)*, 2020, pp. 73–87.
- [9] G. Armeniakos, G. Zervakis, D. Soudris, M. B. Tahoori, and J. Henkel, "Cross-layer approximation for printed machine learning circuits," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022, pp. 190–195.
- [10] G. Armeniakos, G. Zervakis, D. Soudris, M. B. Tahoori, and J. Henkel, "Model-to-circuit cross-approximation for printed machine learning classifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2023.
- [11] A. Kokkinis *et al.*, "Hardware-aware automated neural minimization for printed multilayer perceptrons," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2023.
- [12] G. Armeniakos, G. Zervakis, D. Soudris, and J. Henkel, "Hardware approximate techniques for deep neural network accelerators: A survey," *ACM Comput. Surv.*, vol. 55, no. 4, nov 2022. [Online]. Available: <https://doi.org/10.1145/3527156>
- [13] J. Henkel *et al.*, "Approximate computing and the efficient machine learning expedition," in *International Conference On Computer Aided Design (ICCAD)*, 2022, pp. 1–9.
- [14] D. D. Weller *et al.*, "Printed stochastic computing neural networks," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2021, pp. 914–919.
- [15] 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.
- [16] Z. Cui, *Printed electronics: materials, technologies and applications*. John Wiley & Sons, 2016.
- [17] E. Özer *et al.*, "A hardwired machine learning processing engine fabricated with submicron metal-oxide thin-film transistors on a flexible substrate," *Nature Electronics*, vol. 3, pp. 1–7, 07 2020.
- [18] D. D. Weller, M. Hefenbrock, M. B. Tahoori, J. Aghassi-Hagmann, and M. Beigl, "Programmable neuromorphic circuit based on printed electrolyte-gated transistors," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020, pp. 446–451.
- [19] J. Biggs *et al.*, "A natively flexible 32-bit arm microprocessor," *Nature*, vol. 595, pp. 532–536, 2021.
- [20] K. Iordanou *et al.*, "Tiny classifier circuits: Evolving accelerators for tabular data," *arXiv:2303.00031*, 2023.
- [21] C. Sung *et al.*, "Mix and match: A novel fpga-centric deep neural network quantization framework," in *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2021.
- [22] Y. Hanchen, Z. Xiaofan, H. Zhize, C. Gengsheng, and D. Chen, "Hybridnn: A framework for high-performance hybrid dnn accelerator design and implementation," in *57th ACM/IEEE Design Automation Conference (DAC)*, 2020.
- [23] J. Meng, S. K. Venkataramanaiah, C. Zhou, P. Hansen, P. Whatmough, and J.-s. Seo, "Fixyfga: Efficient fpga accelerator for deep neural networks with high element-wise sparsity and without external memory access," in *International Conference on Field-Programmable Logic and Applications (FPL)*, 2021, pp. 9–16.
- [24] K. Balaskas, G. Zervakis, K. Siozios, M. B. Tahoori, and J. Henkel, "Approximate decision trees for machine learning classification on tiny printed circuits," in *Int. Symp. Quality Electronic Design*, 2022, pp. 1–6.
- [25] D. Dua and C. Graff, "UCI machine learning repository," 2017.
- [26] C. Coelho *et al.*, "Ultra low-latency, low-area inference accelerators using heterogeneous deep quantization with qkeras and hls4ml," *arXiv:2006.10159*, 2021.
- [27] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [28] H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, "Approximate arithmetic circuits: A survey, characterization, and recent applications," *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2108–2135, 2020.
- [29] D. Kalyanmoy, P. Amrit, A. Sameer, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Trans. Evol. Comp.*, vol. 6, no. 2, pp. 182–197, 2002.
- [30] C. Marques *et al.*, "Progress Report on "From Printed Electrolyte-Gated Metal-Oxide Devices to Circuits"," *Advanced Materials*, vol. 31, 2019.
- [31] S. Lanceros-Méndez and C. M. Costa, *Printed Batteries: Materials, Technologies and Applications*. Wiley, 2018.