Fuse and Mix: MACAM-Enabled Analog Activation for Energy-Efficient Neural Acceleration

Hanqing Zhu*, Keren Zhu, Jiaqi Gu, Harrison Jin, Ray T. Chen, Jean Anne Incorvia, David Z. Pan◇
ECE Department, The University of Texas at Austin, Austin, TX, USA
*hqzhu@utexas.edu; ◇dpan@ece.utexas.edu

ABSTRACT

Analog computing has been recognized as a promising low-power alternative to digital counterparts for neural network acceleration. However, conventional analog computing is mainly in a mixed-signal manner. Tedious analog/digital (A/D) conversion cost significantly limits the overall system's energy efficiency. In this work, we devise an efficient analog activation unit with magnetic tunnel junction (MTJ)-based analog content-addressable memory (MACAM), simultaneously realizing nonlinear activation and A/D conversion in a *fused* fashion. To compensate for the nascent and therefore currently limited representation capability of MACAM, we propose to *mix* our analog activation unit with digital activation dataflow. A fully differential framework, `SuperMixer`, is developed to search for an optimized activation workload assignment, adaptive to various activation energy constraints. The effectiveness of our proposed methods is evaluated on a silicon photonic accelerator. Compared to standard activation implementation, our mixed activation system with the searched assignment can achieve competitive accuracy with >60% energy saving on A/D conversion and activation.

1 INTRODUCTION

Deep neural networks (DNNs) have received an explosion of interest due to state-of-the-art inference accuracy in a myriad of artificial intelligence tasks. In parallel, the rapidly escalating model size and data volume raise a surging need for more efficient computing solutions. However, as Moore's law winds down, it becomes increasingly challenging for conventional digital counterparts to meet the computational demands of DNN workloads. A slew of new processor architectures employing analog techniques is keenly sought to reduce power dissipation and improve computational speed. Crossbar-based processing-in-memory (PIM) architectures [2, 10, 18, 19, 26] and integrated optical neural networks (ONNs) [6, 8, 9, 20, 21, 28] are two prominent examples in this direction.

However, analog computing is mainly in a mixed-signal manner: digital inputs are transformed into analog signals for computation, and then computing results are converted back to the digital domain for the downstream operations, e.g., activation. Tedious digital/analog (D/A) and analog/digital (A/D) conversion overhead exists and hinders the overall system energy efficiency [7, 22, 27]. Figure 1 shows two case studies of energy breakdown in two representative analog convolutional NN (CNN) accelerators, where the A/D conversion achieved by costly analog-to-digital converters (ADCs) counts for a significant portion of overall power consumption. The A/D conversion overhead concern is further escalated in ONNs due to its high-speed ADC requirements.

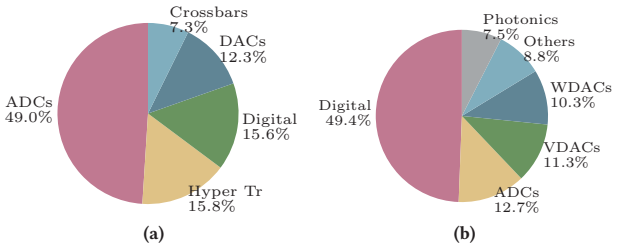


Figure 1: Power breakdown of (a) a crossbar-based CNN accelerator `ISAAC` [19] and (b) a silicon photonic accelerator `Mars` [17]. WDACs means DACs for weights, and VDACs means DACs for vector inputs. In `Mars`, it chooses a more costly DAC configuration than the ADC configuration, and most of the power consumption in *digital* part comes from data movement.

Based on the above analysis, we observe a solid demand for seeking another efficient and low-latency alternative to costly ADCs to bring analog signals back to the digital domain. A recently proposed memristor-based analog content-addressable memory (ACAM) [13] shows up to be a promising candidate with picosecond-level latency and femtojoule-level energy consumption. BRAHMS [22] has explored a RRAM-based accelerator to use ACAM to implement nonlinear activation, pooling, and A/D conversion successively, thus eliminating the usage of ADCs. However, the separate implementation of the three operations requires routing the analog signals back to ACAM three times, raising a signal noise concern. Moreover, a severe accuracy drop has been seen in [22] when the precision of A/D conversion implemented by ACAM is insufficient. These obstacles need to be overcome before viable ACAM can be utilized in analog computing to reduce the A/D overhead.

In this work, we devise an efficient analog activation unit based on MTJ-based ACAM (MACAM), simultaneously implementing nonlinear activation and A/D conversion in a *fused* manner. To compensate for the limited representation capability of MACAM, we propose a *mixed* activation system that integrates the proposed low-energy analog activation and the conventional high-precision digital activation datapaths. With a given activation energy constraint, a `SuperMixer` training flow is proposed to automatically learn how to assign activation workloads on the mixed activation system, aiming at balancing the expressiveness and energy cost.

Our main contributions are as follows,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICCAD '22, October 30-November 3, 2022, San Diego, CA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9217-4/22/10…$15.00
https://doi.org/10.1145/3508352.3549449

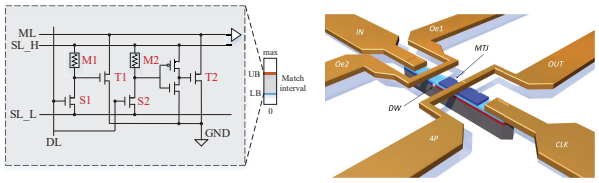
- We propose a novel analog and *mixed* activation system for energy-efficient neural network acceleration.
- We devise a ***fused* analog activation unit** based on MACAM that can *simultaneously* achieve nonlinear activation and A/D conversion with significant energy reduction.
- We propose a ***mixed* activation system** that integrates both analog and digital activation dataflows to balance expressiveness and energy efficiency.
- We develop a **`SuperMixer` framework** to automatically learn how to assign activation workloads on our mixed activation system adaptive to various energy constraints.
- A **learnable nonlinearity threshold** is proposed with an *enhanced training recipe* to boost accuracy under limited MACAM resolution, allowing practical application of very nascent technologies to use their benefits and mitigate their present challenges.
- We evaluate our methods on a photonic accelerator. Regarding the energy cost of A/D conversion and activation, experiment results show that using fully analog activation units gives ~65% energy saving, and the searched assignment on the mixed activation system can achieve >60% energy reduction with comparable accuracy.

2 PRELIMINARIES

2.1 Analog Content-addressable Memory

Memristor-based analog content-addressable memory is recently proposed in [13]. Fig. 2(a) shows the single ACAM cell design that supports search in a continuous analog interval. The lower bound (LB) and upper bound (UB) of the match interval are represented by tuning the resistance of two memristor devices, M1 and M2. The ACAM cell can take an analog voltage as the input, which is applied to the data line (DL). DL is connected to the gate of two switching transistors, S1 and S2. When the input is smaller (larger) than the LB (UB), both S1 and S2 are OFF (ON), making T1 (T2) ON. Then the match line (ML) will be pulled down, leading to a *mismatch*. If the input is within the interval, the ML will not be pulled down, resulting in a *match*. The prior study [13, 22] has proved the ACAM can achieve the search functionality with picosecond-level latency and femtojoule-level energy consumption. We can cascade multiple ACAM cells to form an ACAM array, where each cell represents one specific interval such that all intervals can consist of a large search interval. The maximum search range is decided by the minimum and maximum resistance of the memristor device. It is obvious that the number of implementable intervals is decided by the number of available resistances of the memristor device.

Regarding the choice of the memristor device, since the ACAM acts as a role of memory with potential frequent access, it inherently requires the chosen memristor to be long-endurance. The magnetic tunnel junction (MTJ) turns out to be a more suitable choice with excellent endurance (10^{15} cycles) compared with PCM (10^7 cycles) and RRAM (10^5 cycles) [18]. Among MTJs, Figure 2(b) shows a three-terminal domain wall-MTJ (DW-MTJ) implementation, which isolates the read (OUT) and write (IN) paths for even better endurance while also providing analog resistance levels with excellent stability of the resistance levels over numerous cycles [12].



(a) (b)

Figure 2: (a) Schematic of the ACAM cell [13]. (b) The adopted three-terminal domain wall-MTJ [12].

Hence, in this paper, we choose the DW-MTJ as the memristor device of ACAM. But, this is a very new MTJ type, and currently the number of resistance levels implemented on DW-MTJ is rather limited, where 5 stable resistance levels are demonstrated in [12] as the state-of-the-art (SOTA) implementation.

2.2 Related Works and Motivation

Several prior works have explored reducing the A/D conversion overhead in PIM. In [16], building fully analog circuits is proposed such that the signals are transmitted between layers without A/D conversion. This fully analog manner has been proved to be accuracy unfriendly. `PRIME` [2] chooses to use sense amplifiers (SAs) to do A/D conversion instead of ADCs. But a SA can only convert one bit at a time, leading to a long latency to get the whole output. `CASCADE` [5] proposes to accumulate the partial sums in the analog domain by connecting the outputs of multiple crossbars via an additional buffer RRAM array. However, the last-mile A/D conversion is kept to convert the sum back to the digital domain for downstream tasks, e.g., activation.

Typically, in analog computing, only matrix multiplications (MMs) are conducted in the analog domain, while nonlinear activation function and pooling are implemented in the digital domain. Thus, A/D conversion is necessary. `BRAHMS` [22] proposes to put all the MMs, activation, and pooling in the analog domain, where ACAMs implement the latter two in two stages. In this way, only those analog values which are kept after activation and pooling need to be converted to the digital domain. The A/D conversion is still done by ACAMs. In this case, costly A/D conversion can be reduced by using the efficient ACAM. However, it is prone to signal noise since the analog signals need to be routed through ACAM three times to do nonlinear projection, pooling, and AD conversion. Moreover, severe accuracy drops are observed when the precision of A/D conversion implemented by ACAM is insufficient.

Therefore, in this paper, we devise a MACAM-based analog activation unit. It is capable of implementing nonlinear activation and A/D conversion at one time in a fused manner. It can replace the traditional digital activation path, i.e., costly ADCs and digital activation units. As a result, the overhead of A/D conversions can be reduced. Since choosing MTJ as the memristor device will raise the precision issue as well, a slew of efforts are dedicated to improving the expressiveness of this device type.

3 MACAM-ENABLED FUSED ANALOG ACTIVATION UNIT

Nonlinear activation functions in analog computing are normally implemented by digital logics or lookup tables in the digital domain.

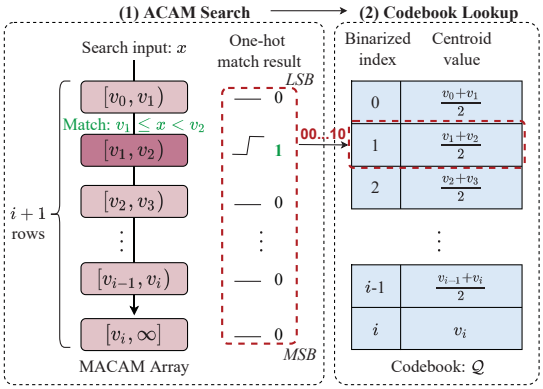


Figure 3: Illustration of implementing the positive part of ReLU-α and A/D conversion in a single ACAM array.

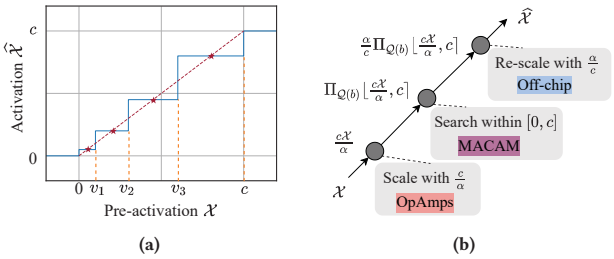


Figure 4: (a) Transformation behaviors of ReLU-α implemented with MACAM. (b) Implementation of arbitrary ReLU-α by MACAM.

Since the computation is conducted in the analog domain, analog-to-digital conversion overhead exists to bring analog signals to the digital domain for activation. We denote this conventional activation implementation as electrical *digital activation*, which includes the needed ADCs and the following digital logics or lookup tables.

Recently, BRAHMS [22] has explored the idea of using ReRAM-based ACAM to implement nonlinear activation functions and A/D conversions, benefiting from the high-speed possessing and low power consumption of ACAM. However, the nonlinear projection and A/D conversion are separate. The analog signals being nonlinearly projected need to be routed back to the ACAM to do the A/D conversion, which may suffer from signal noise. Moreover, severe accuracy drops are observed when the precision of ACAM-based A/D conversion is not sufficient.

In this paper, we propose to implement the nonlinear activation function using MACAM, where **nonlinear activation** and **A/D conversion** are *simultaneously* achieved by the same MACAM array in a *fused* fashion. The fundamental idea here is to utilize the analog search functionality of MACAM to digitize the analog input signals while introducing *in-situ* nonlinearity. We call this fused nonlinear activation unit an *analog activation* unit. In this way, activation energy cost can be largely eliminated, as most ADCs can be replaced by energy-efficient MACAM.

Considering the precision issue, instead of supporting general nonlinear functions as in [22], we choose to implement the ReLU-α function. ReLU-α is widely used in quantized models that have limited bit-widths to represent weight and activation. The ReLU-α function with a clipping threshold α works as follows,

$$\widehat{\mathcal{X}} = \begin{cases} 0, \mathcal{X} \in (-\infty, 0), \\ \mathcal{X}, \mathcal{X} \in [0, \alpha), \\ \alpha, \mathcal{X} \in [\alpha, +\infty), \end{cases} \tag{1}$$

where $\mathcal{X}$ is the pre-activation feature map and $\widehat{\mathcal{X}}$ is the final output feature map. The clipping threshold α can bound the output of the ReLU function, therefore, a small bit-width is capable of representing the bounded value range of feature maps.

For ReLU-α, since the input values can be either positive or negative, we need two MACAM arrays to handle positive and negative inputs, respectively. The negative part of ReLU-α is easy to implement by generating a constant digital result '0'. Fig. 3 illustrates the implementation for the positive part of ReLU-α in a single MACAM array. In the i-th row of the MACAM array, by tuning the resistance of the MTJ, one MACAM cell represents an acceptance search interval $[v_{i-1}, v_i)$ for a match. All the intervals of MACAM cells in the top i rows consist of a searchable range of $[v_0, v_i)$. i is decided by the number of implementable resistances of the MTJ. To achieve the search interval $[v_i, \infty]$, we encode ∞ similar to [22] by using a resistor with considerably larger resistance than the resistance upper bound of the MTJ. It is sufficient to cover the possible maximum voltage output from the analog engine as it cannot be arbitrarily large. The match result of all match lines consists of a one-hot vector. After obtaining the one-hot match result, we then transfer it into a binarized index using a digital priority encoder such that it can be stored at a minimized memory storage cost. The digital priority encoder can also handle corner cases when input is on the bounds of interval. Each binarized index represents a group of inputs that falls into one specific interval. Similar to quantization, we set the binarized index to correspond to one shared value, in which we use the mid-point of the interval's lower bound and upper bound in our implementation. For interval $[v_i, \infty]$, the corresponding value is set to v_i. This mapping relationship can be defined as a codebook $\mathcal{Q}$ as shown in Fig. 3. Within this process, we actually digitize the analog input signals. For example, in Fig. 3, the input x falls into $[v_1, v_2)$, the one-hot match result corresponds to the binarized index '1', which is treated as $\frac{v_1+v_2}{2}$. Since the minimum voltage v_0 cannot be 0, the input needs to be biased to support the search of inputs starting from 0. In the following, for simplification of illustration, we assume $v_0 = 0$ and $v_i = c$.

Now we explain how the projection of the MACAM array can realize *in-situ* nonlinearity of ReLU-α. Figure 4(a) show the transformation behavior of the MACAM array. The range $[0, c]$ is divided into four intervals, represented by the MTJ with 5 resistance levels using the measured data from [12]. The inputs within each interval are treated as the same value after projection based on the $\mathcal{Q}$. If we draw the red dashed line through the interval's midpoint, which is labeled by the star symbol, it implements $y = x$. Actually, we can treat this as a nonlinear quantized version of a ReLU-α function.

The MACAM array can only support a fixed searchable range of $[0, c]$. In order to implement any arbitrary ReLU-α, we need to first scale the $\mathcal{X}$ with $\frac{c}{\alpha}$ to fit the search range, feed the scaled input into ACAM, and then scale it back to $[0, \alpha]$, as shown in Fig 4(b). The first scaling operations before the MACAM array can be done using OpAmps. We can use shared OpAmps at the output

of the computation units without extra overhead [23]. The off-chip computers can do the second scaling operation.

All in all, the behavior of MACAM-based activation can be modeled as follows,

$$\widehat{\mathcal{X}} = \frac{\alpha}{c}\Pi_{\mathcal{Q}}\lfloor\frac{c\mathcal{X}}{\alpha}, c\rceil, \tag{2}$$

where $\Pi_{\mathcal{Q}}$ indicates the discretized projection function of the ACAM within $[0, c]$ following the codebook $\mathcal{Q}$. $\lfloor\cdot, c\rceil$ denotes clipping to the range of $[0, c]$.

4 MIXED ACTIVATION SYSTEM

4.1 Proposed Mixed Activation System

As the DW-MTJ can currently only demonstrate a small number of resistance levels, the number of implementable intervals in MACAM is restricted, further limiting the representation capability of the MACAM-based analog activation unit. Hence, we are motivated to provide a *mixed* activation system, which integrates both the *fused* analog activation and the traditional digital activation datapaths. The traditional digital activation datapath first uses high-precision ADCs to do A/D conversion, then performs nonlinearity in dedicated digital activation units. The output feature map can choose either of the two paths for nonlinear activation and A/D conversion. With the *mixed* activation system, we can jointly utilize the low-energy analog activation datapath and the high-precision digital datapath to balance expressiveness and energy efficiency.

Figure 5 shows the system architecture overview with the tensor computing system and the proposed *mixed* activation system. We represent the fundamental tensor computing unit as vector-dot-product (VDP) unit, with each unit supporting the dot-product between two length-N vectors. Multiple VDPs implement a large-size vector product due to the limited single VDP size, generating multiple partial sums (PS). Note that nonlinear activation is not applied to any intermediate PS but to the *final* computation results, thus applying our MACAM-based analog activation requires in-place partial sum accumulation in the analog domain, which is doable by partial sum summation units [5, 22, 24]. Then, the final results are assigned to either the analog activation or the digital activation datapath to do nonlinear activation and A/D conversion.

We elaborate a weight-stationary dataflow to meet the requirement of the overall system following [22]. Take a convolutional layer as an example, where the 2-D weights $W \in \mathbb{R}^{C_o \times (C_i k^2)}$ and 2-D output feature map $\mathcal{X} \in \mathbb{R}^{C_o \times (H' W')}$. The size of vector-dot-product in its computaion is $C_i k^2$, which is distributed onto $\lceil \frac{C_i k^2}{N} \rceil$ VDPs. This is doable since the largest vector product in modern ResNet and VGG models is only $(512 \times 3 \times 3)$. At each cycle, we obtain the convolution result of the whole dot-product, ensured by in-place partial sum summation units. Fig. 6 illustrates our dataflow on a simple filter with 2 input channels. The weights can be stationary in the VDP units to continually complete the convolutions of adjacent sliding windows. Instead of frequently fetching inputs from costly memory, inputs can be reused in the convolutions of adjacent sliding windows. After obtaining the whole result, it can be sent to the activation datapath to do A/D conversion and activation.

4.2 Fully Differentiable `SuperMixer` Training

Our *mixed* activation system provides a more costly but higher precision digital activation datapath to compensate for the expressiveness due to analog activation datapath. This intuitively raises the question, "how do we *assign* the activation tasks of output feature maps onto the two different activation datapaths to balance expressiveness and energy efficiency?"

Problem Formulation. Considering the energy gap between the two activation datapaths, our target is to assign the activation task of each value of output feature map $X \in \mathcal{X}$ to either of the two activation paths with high expressiveness under a given activation energy constraint. We define the assignment as $\mathcal{A}$. In this way, the problem can be formulated as follows,

$$\begin{aligned} &\min \mathcal{L}\big(W^{*\mathcal{A}};\ \mathcal{D}^{val}\big) \\ &\text{s.t. } W^* = \underset{W}{\operatorname{argmin}}\ \mathcal{L}(W^{\mathcal{A}};\ \mathcal{D}^{trn}), \\ &\quad E_{act,min} \le E_{act}(\mathcal{A}) \le E_{act,max}, \\ &\quad \widehat{X}^l = \sum_{i=1}^{2} a^l_{X,i} f_i(X^l),\ X^l \in \mathcal{X}^l, \\ &\quad \sum_{i=1}^{2} a^l_{X,i} = 1,\ a^l_{X,i} = \{0, 1\}. \end{aligned} \tag{3}$$

The binary selection variable $a_{X,i}$ decides that each output feature value $X \in \mathcal{X}^l$ in the l-th layer is passed through either $f_1(\cdot)$ (MACAM-based analog activation datapath) or $f_2(\cdot)$ (digital activation path). The selection variables consist of the assignment $\mathcal{A}$, which is our primary search target with a activation energy cost $E_{act}(\mathcal{A})$ satisfying its constraints $[E_{act,min}, E_{act,max}]$.

Search Space Specification. Considering the on-chip routing issue and computation regularity, it is not realistic to tediously assign individual output feature map value to different activation units. Instead, in this work, we propose to assign the workloads in the *filter* level. Concretely, take a convolutional layer l as an example. Suppose it contains C_o filters and each filter has C_i input channels and kernel size k, where its 2-D input $\mathcal{X}^{l-1} \in \mathbb{R}^{C_i \times (H \times W)}$ and 2-D output feature $\mathcal{X}^l \in \mathbb{R}^{C_o \times (H' \times W')}$. The entire b-th output channel of the output feature map $\mathcal{X}^l$ is passed through either the analog activation path $f_1(\cdot)$ or the electrical activation path $f_2(\cdot)$, decided by a binary selection variable $a^l_{b,i}$. The behavior is given as follows,

$$\widehat{\mathcal{X}}^l_b = \sum_{i=1}^{2} a^l_{b,i} f_i(\mathcal{X}^l_b),\ a^l_{b,i} \in \{0, 1\},\ \sum_i a^l_{b,i} = 1. \tag{4}$$

In this way, our search space is defined as the assignment $\mathcal{A}$ that assigns each channel of output feature maps to either the analog or the electrical activation path. For layer l, the number of total combinations can be 2^{C_o}. Thus, the total search space for a model with L layers is extremely large, which is $O(\prod_l^L 2^{C^l_o})$.

Fully Differentiable `SuperMixer` Training. Considering the enormous search space of $\mathcal{A}$ and its discreteness, we propose a differentiable `SuperMixer` training flow as shown in Fig. 7.

In the `SuperMixer` training, we need to optimize weight W, clipping threshold α in ReLU-α, and assignment $\mathcal{A}$. It is highly ill-conditioned to jointly optimize all those continuous and discrete variables. Aware of this, in this work, we divide our `SuperMixer` flow into two phases. The first `SuperMixer` *Warmup* phase aims

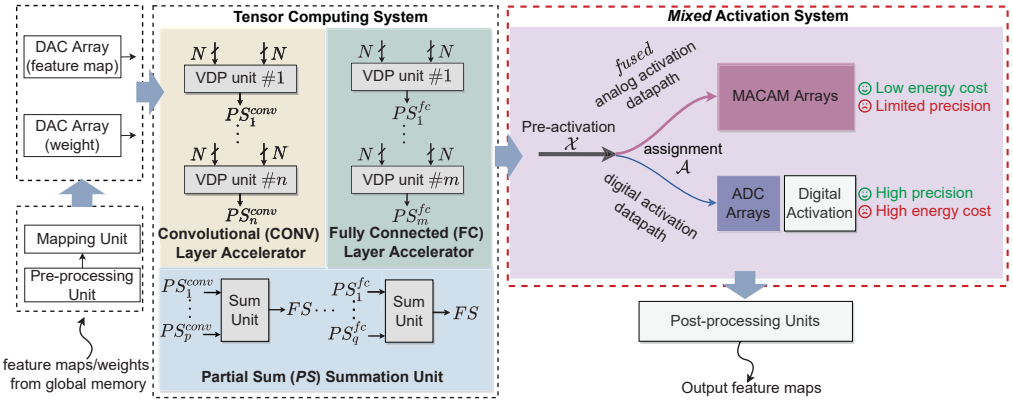


Figure 5: The system architecture overview, including tensor computing system and the proposed *mixed* activation system.

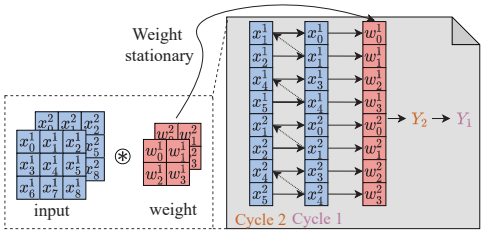


Figure 6: Illustration of our adopted weight-stationary dataflow to combine convolution and activation. We show the convolution of one filter with 2 channels.

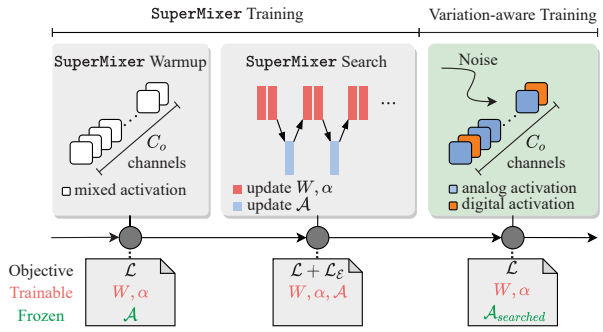


Figure 7: The proposed `SuperMixer` training flow and variation-aware training with injected noises.

at stabilizing our following search stage, where only W and α are optimized to obtain a good initial point. The second `SuperMixer` *Search* phase aims at searching for the assignment $\mathcal{A}$ to boost the expressiveness. It optimizes (1) W, α and (2) $\mathcal{A}$ alternatively to avoid prohibitive co-optimization difficulty. We periodically enter the optimization of (1) and (2) with a ratio of 2:1. Moreover, during the `SuperMixer` *Search* phase, we need to satisfy the activation energy cost constraint, thus, an energy cost penalty $\mathcal{L}_\mathcal{E}$ is added besides the original loss $\mathcal{L}$. After `SuperMixer` training, we fix the searched optimal assignment $\mathcal{A}_{searched}$ and conduct variation-aware training to improve the model's robustness regarding on-chip variations, e.g., the resistance variation of memristors in MACAM.

Now we explain how to optimize assignment $\mathcal{A}$ in a differentiable way. As shown in Eq. (4), the selection variable $a^l_{b,i}$ is a binary variable. Instead of searching $\mathcal{A}$ in such a discrete space, we relax the optimization problem by constructing a stochastic mixed activation unit. During the inference, the activation of the b-th channel of feature in layer l, $\mathcal{X}^l_b$, is using either the analog activation path $f_1(\cdot)$ or the digital activation path $f_2(\cdot)$ with the sampling probability of

$$P_{\theta^l_b}(f_b = f_i) = \frac{e^{\theta^l_{b,i}}}{\sum_i e^{\theta^l_{b,i}}}. \tag{5}$$

Equivalently, the output of the stochastic mixed activation, $\widehat{\mathcal{X}}^l_b$, can be expressed as,

$$\widehat{\mathcal{X}}^l_b = \sum_{i=1}^{2} a^l_{b,i} f_i(\mathcal{X}^l_b), \tag{6}$$

where $a_{b,i}$ is a random variable in $\{0, 1\}$ and is evaluated based on the sampling probability in (5). Therefore, through parameterizing the probability distribution of activation unit choices by the sampling coefficient θ^l_b, we can relax the problem as the optimization of probability of P_θ. However, we cannot propagate the gradient back through the discrete variable $a_{b,i}$ to $\theta_{b,i}$. To sidestep this issue, Gumbel-Softmax (GS) trick is adopted to relax $a_{b,i}$ to be a continuous varible as follows,

$$a^l_{b,i} = \text{GumbelSoftmax}(\theta^l_{b,i}|\theta^l_b) = \frac{e^{(\theta^l_{b,i}+g^l_{b,i})/\tau}}{\sum_i e^{(\theta^l_{b,i}+g^l_{b,i})/\tau}}. \tag{7}$$

$g^l_{l,i}$ follows the Gumbel distribution Gumbel$(0, 1)$ as a Gumbel noise. A temperature parameter τ is used to control the GS function. As τ is close to 0, the Gumbel-Softmax function approximates categorical samples based on (5). A larger τ introduces randomness to encourage exploitation of the assignment $\mathcal{A}$. Therefore, during the `SuperMixer` *Search* phase, we gradually decay τ such that `SuperMixer` can automatically exploit the search space and learn the optimal assignment to augment $\mathcal{A}$ the models' expressiveness.

Activation Energy-Constrained Optimization. The assignment $\mathcal{A}$ directly impacts the activation energy cost, as it defines the mixed way of using low-precision analog activation units and high-precision electrical activation units. Fully using analog activation datapath results in an ADC-free system to convert computation

results to the digital domain, while fully using electrical activation datapath raised a serious A/D conversion cost. With the assignment $\mathcal{A}$, we can get the activation energy cost $E(\mathcal{A})$ as follows,

$$\begin{aligned} E_{act}(\mathcal{A}) &= \#act_{anlg} \cdot E_{anlg} + \#act_{digi} \cdot E_{digi} \\ &= \sum_{l}^{L} \sum_{b}^{C_o} (a^l_{b,i} \cdot E_{anlg} + a^l_{b,2} \cdot E_{digi}) \cdot H^{'} W^{'}. \end{aligned} \tag{8}$$

$\#act_{anlg}$ and $\#act_{digi}$ denote the number of output feature maps being passed through the analog (*anlg*) activation datapath and the digital (*digi*) activation datapath, respectively. The E_{anlg} and E_{digi} represent the energy cost of two datapaths. The former indicates the MACAM array's cost, and the latter contains ADC and digital activation unit costs. As the ADC cost is far larger [19] and the digital activation unit cost is dependent on the digital part's frequency, we use ADC cost to represent E_{digi} during the search phase.

To honor the energy cost constraint $[E_{act,min}, E_{act,max}]$, we add a *probabilistic activation energy cost penalty* term $\mathcal{L}_{\mathcal{E}}$,

$$\mathcal{L}_{\mathcal{E}} = \begin{cases} \beta \left(E_{act}(\mathcal{A})/(1-\gamma)E_{max}\right), & E_{act}(\mathcal{A}) > (1-\gamma)E_{max}, \\ -\beta \left(E_{act}(\mathcal{A})/(1+\gamma)E_{min}\right), & E_{act}(\mathcal{A}) < (1+\gamma)E_{min}, \\ 0, \text{otherwise.} \end{cases} \tag{9}$$

We set the margin γ to 5% to tighten the constraint. Note that $a^l_{b,i}$ in Eq. (8) is differentiable guaranteed by the GS trick.

Precision-Adaptive α Optimization. Implementing ReLU-α can ease the issue of insufficient precision of MACAM. However, it is not enough to address this since the SOTA MTJ can only provide five stable resistance levels, as mentioned before. The precision of ReLU-α implemented by ACAM is only around 2-bit. Thus, a solution is in great need to further tolerate the precision issue.

Instead of choosing a fixed α in Eq. (1), inspired by PACT [3], we propose to adopt a *learnable* α with an *enhanced training recipe* to *accommodate the low resolution of MACAM for accuracy boost.* This can be done by configuring the gain of the OpAmps [23]. In [3], during the learning process of α, the gradient to α is computed by $\frac{\partial \widehat{\mathcal{X}}}{\partial \alpha} = \texttt{Sign}(\mathcal{X})$ when $\mathcal{X} \geq \alpha$. Activation $\mathcal{X}$ that is smaller than α cannot contribute to the gradient, resulting in inaccurate gradient estimation to α. Instead of updating α in the same way of [3], we re-formulate the gradient to α based on Eq.(2) as follows,

$$\begin{aligned} \frac{\partial \widehat{\mathcal{X}}}{\partial \alpha} &= \frac{\partial c}{\partial \alpha} \cdot \frac{1}{c} \Pi_{Q(b)} \lfloor \frac{c\mathcal{X}}{\alpha}, c \rceil + \frac{\partial \Pi_{Q(b)} \lfloor \frac{c\mathcal{X}}{\alpha}, c \rceil}{\partial \lfloor \frac{c\mathcal{X}}{\alpha}, c \rceil} \frac{\partial \lfloor \frac{c\mathcal{X}}{\alpha}, c \rceil}{\partial \alpha} \cdot \frac{\alpha}{c} \\ &= \begin{cases} 0, \mathcal{X} \in (-\infty, 0), \\ 1 \cdot \frac{1}{c} \Pi_{Q(b)} \lfloor \frac{c\mathcal{X}}{\alpha}, c \rceil + 1 \cdot \frac{-c\mathcal{X}}{\alpha^2} \cdot \frac{\alpha}{c}, \mathcal{X} \in [0, \alpha), \\ 1 \cdot \frac{1}{c} \cdot \texttt{Sign}(\frac{c\mathcal{X}}{\alpha}) \cdot c + \frac{\alpha}{c} \cdot 0, \mathcal{X} \in [\alpha, +\infty), \end{cases} \\ &= \begin{cases} 0, \mathcal{X} \in (-\infty, 0), \\ \frac{1}{c} \Pi_{Q(b)} \lfloor \frac{c\mathcal{X}}{\alpha}, c \rceil - \frac{\mathcal{X}}{\alpha}, \mathcal{X} \in [0, \alpha), \\ 1, \mathcal{X} \in [\alpha, +\infty), \end{cases} \end{aligned} \tag{10}$$

where our scaling operation works as a reparameterization trick to preserve the gradient contribution from $\mathcal{X} \in [0, \alpha)$. Thus, it can correct the inaccurate gradient estimation to α, which is proved to get a significantly larger accuracy boost than [3] in our experiments.

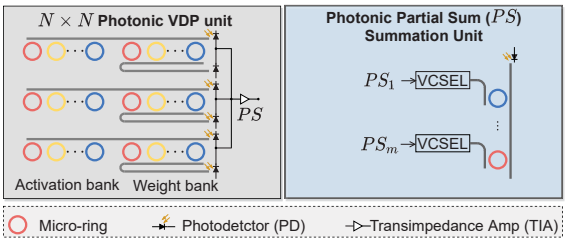


Figure 8: The adopted photonic vector-dot-product unit [24, 25] and photonic partial sum summation unit [24].

5 CASE STUDY: PHOTONIC ACCELERATOR

We demonstrate our design principle on a photonic accelerator as a case study. We focus on a SOTA incoherent photonic design based on micro-ring (MR) resonators [24, 25]. Other accelerators can also benefit from our methods. For example, we can simply replace the ACAM design in a RRAM-based accelerator [22] with ours but own better tolerance to signal noise and the precision of ACAM.

Fig. 8 shows the adopted photonic vector-dot-product (VDP) engine [24, 25] and photonic in-place partial sum (PS) summation unit [24] for our system in Fig. 5. The computation of the convolutional (CONV) layer and fully-connected (FC) layer are unfolded to matrix multiplication, where each vector dot product is implemented by the vector dot product (VDP) units based on micro-ring (MR) resonators. To support two length-N vector multiplication within each VDP unit, the N size vectors are decomposed into small-sized vector chunks. Each small-sized vector dot-product is performed using MRs in each arm of the VDP unit. In this way, a large $N \times N$ vector dot product can be achieved in one unit, e.g., 100×100 in [24]. Across VDP units, photonic partial summation units are used to accumulate the partial sums from multiple VDPs. The partial sums from multiple VDPs are converted from the analog domain to the photonic domain by VCSELs, multiplexed into one single waveguide, and summed via another photodetector.

Energy Modeling of the A/D Conversion and Activation Cost. Here, we model the energy cost of the A/D conversion and activation, $E_{A/D+act}$ based on `O-HAS` [14]. Our mixed activation system provides two datapaths to do nonlinear activation and A/D conversion, with an extra photonic summation unit overhead. Consider one convolutional layer with C_o filters and output feature map $\mathcal{X}^l \in \mathbb{R}^{C_o \times (H^{'} \times W^{'})}$. Each filter has C_i input channels and a kernel size of k. The energy model of $E_{A/D+act}$ can be modeled as follows,

$$\begin{aligned} E_{A/D+act} &= E_{act}(\mathcal{A}) + E_{sum} \\ &= (C_{o,anlg} E_{anlg} + C_{o,digi}(E_{digi,adc} + E_{digi,act})) H^{'} W^{'} \\ &+ (E_{VCSEL}) C_o H^{'} W^{'} \left\lceil \frac{C_i k^2}{N} \right\rceil + C_o H^{'} W^{'} E_{PD}. \end{aligned} \tag{11}$$

E_{anlg} denotes the energy of the analog activation datapath, $E_{digi,adc}$ and $E_{digi,act}$ denote the A/D conversion and activation energy of the digital activation datapth. E_{VCSEL} and E_{PD} denote the energy for VCSEL and photodetector in the photonic PS summation unit.

In conventional implementation, the A/D conversion is done right after obtaining partial sum and the digital activation datapath

Table 1: 6-bit ADC configurations considered in this paper.

	ADC-1 [4]	ADC-2 [1]
Bit-width	6	6
Sampling rate (GS/s)	1	6
Power (mW)	1.26	14
Latency (ns)	8	1.33

is fully used. The energy cost can be modeled as,

$$\begin{aligned} E_{A/D+act} &= E_{act} + E_{A/D} \\ &= C_o H^{'} W^{'} (E_{digi,act}) \\ &+ (E_{ADC} + E_{S+A} + E_{PD}) C_o H^{'} W^{'} \left\lceil \frac{C_i k^2}{N} \right\rceil. \end{aligned} \tag{12}$$

6 EXPERIMENTAL RESULTS

6.1 Experiment Setup

Models and Datasets. We evaluate our methods on two modern CNNs (VGG13 and ResNet18) and CIFAR100 [11], with 6-bit weight precision. In VGG13, we replace the last three FC layers with one to avoid over-fitting. In ResNet18, we move the residual path after activation such that no extra addition is needed before activation.
Training Settings. In `SuperMixer` training flow, we train for 90 epochs using an SGD optimizer with an initial learning rate (lr) of 0.02, a momentum of 0.9, and a cosine lr scheduler. The Gumbel-softmax temperature τ exponentially decreases from 5 to 0.5. γ is set to 0.6 for the activation energy cost penalty. The initial α of the adaptive ReLU-α is set to 8. We set 10 epochs for `SuperMixer` *Warmup* phase and 80 epochs for *search* phase. We sample an assignment $\mathcal{A}$ from the learned distribution P_θ, then enter the variation-aware training. We train all models for 200 epochs during variation-aware retraining using an SGD optimizer with an initial learning rate (lr) of 0.02, a momentum of 0.9, and a cosine lr scheduler.
MACAM and ADC Designs. We consider two MACAM designs with two MTJ devices in this paper. MACAM-1 uses the DW-MTJ with 5 resistance levels with around 2-bit precision. MACAM-2 uses the DW-MTJ with 3 resistance levels with around 1-bit precision. We choose two 6-bit ADC designs shown in Table. 1.
Noise Injection. We set the variation of MRs following a Gaussian distribution $\mathcal{N}(0, 0.05^2)$. For MACAM, we set the MTJ resistance device-to-device variation as $\mathcal{N}(0, 0.128^2)$ [15], run Monte Carlo simulation 10^4 times to capture the noisy distribution of MACAM's intervals, and equivalently add it to the input following [22].
A/D and Activation Energy Simulation. The performance and power dissipation of MACAM are evaluated with Cadence ADE and spectre simulations. We estimate the A/D conversion and activation energy by Eq. (11) based on `O-HAS` [14]. The size of VDP is set as 128. Since the final FC layer is the output layer without activation, for a fair comparison, we don't include its energy consumption.

6.2 Main Results

Evaluation of Our Precision-adaptive ReLU-α. In Table 2, we validate the expressiveness of our proposed precision-adaptive ReLU-α on VGG13. We compare it with the adaptive ReLU-α in PACT [3] and the ReLU-α with a fixed α. For the ReLU-α with fixed α, the commonly used ReLU6 is adopted, while ReLU2 is adopted in the extremely low activation bit-width. Under 6-bit weight bit-width, our implementation achieves the highest accuracy, especially on low activation bit-width cases. This attributes to our proposed precision-adaptive α optimization scheme, which learns the α to accommodate the low resolution. Compared to PACT [3], our enhanced training recipe can correct its inaccurate gradient, resulting in a better accuracy boost. Hence, our adaptive ReLU-α can address the accuracy drop issue in [22] due to the low activation bit-width. It is essential to boost the expressiveness of the analog activation unit so as to achieve competitive model accuracy.

Table 2: Compare different ReLU variants on CIFAR100. 2+ and 1+ represents the precision of ACAM-1 and ACAM-2, respectively.

Model	Weight bit	Act. bit	ReLU	Accuracy (%)
VGG13	32	32	ReLU	74.76
	6	6	ReLU6	71.31
	6	6	ReLU-α-PACT [3]	73.80
	6	6	ReLU-α-Ours	**73.91**
	6	2+	ReLU6	68.17
	6	2+	ReLU-α-PACT [3]	70.96
	6	2+	ReLU-α-Ours	**72.52**
	6	1+	ReLU2	54.98
	6	1+	ReLU-α-PACT [3]	67.11
	6	1+	ReLU-α-Ours	**70.84**

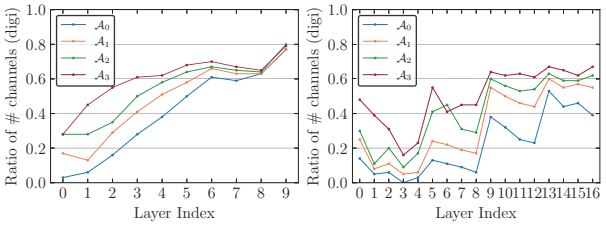
Evaluation of Our `SuperMixer`. We search the assignment $\mathcal{A}$ with the proposed `SuperMixer` flow on different MACAM designs, ADC designs, and activation energy constraints. We denote searched assignments as searched $\mathcal{A}_0$ to searched $\mathcal{A}_3$. Table 3 and 4 show the test accuracy of searched assignments on VGG13 and ResNet18. Our searched assignment series show improved expressiveness in terms of accuracy with constrained activation energy cost. Especially, given a large activation energy cost budget on MACAM-1, our `SuperMixer` can find an assignment with comparable accuracy to the case fully using high-precision digital activation. In conclusion, our `SuperMixer` flow successfully utilizes the provided *mixed* activation system to boost the expressiveness.
Energy Saving on AD Conversion and Activation. We simulate the energy cost of A/D conversion and activation cost on VGG13. Configurations of ADC-1 and ACAM-1 are adopted. The conventional implementation of using ADCs for A/D conversion of partial sums and using digital activation datapath consumes 35.7 μJ. With photonic summation units to eliminate A/D conversion needs of partial sums, fully using digital activation datapath consumes 17.24 μJ with a 51.7% reduction, and fully using analog activation datapath uses 12.44 μJ with a 65.2% reduction. Our `SuperMixer` enables the mixed use of electrical and analog activation with a trade-off between energy cost and expressiveness, where the searched-$\mathcal{A}_3$ consumes 14.2μJ with 60.2% reduction but comparable accuracy.
Searched Layer-wise Assignment $\mathcal{A}$. We further validate the success of `SuperMixer` to learn a good assignment $\mathcal{A}$ for boosting expressiveness. Fig. 9(a) and Fig. 9(b) visualize the ratio of feature map channels being assigned to digital activation path in each layer of VGG13 and ResNet18. Under different activation energy cost constraints, `SuperMixer` learns a similar tendency to assign the workloads. In former layers, the size of each channel of the feature map is larger, thus fewer channels are assigned to the electrical activation units to avoid violation of the energy constraint. In contrast,

Table 3: Test Accuracy of searched VGG13 on different ADC and MACAM designs, where the model is searched on CIFAR100. All activation energy cost is normalized by the activation energy of fully using digital activation datapath.

Model	MACAM design	ADC design	Metrics	Fully digital	Fully analog	Searched-$\mathcal{A}_0$	Searched-$\mathcal{A}_1$	Searched-$\mathcal{A}_2$	Searched-$\mathcal{A}_3$
VGG13	MACAM-1	ADC-1	$[E_{act,min}, E_{act,max}]$	-	-	[0.05, 0.15]	[0.15, 0.25]	[0.25, 0.35]	[0.35, 0.45]
			Activation energy $E_{act}(\mathcal{A})$	1	0.00036	0.11	0.19	0.28	0.39
			Accuracy (%)	73.91	72.52	72.94	72.73	73.09	**73.30**
		ADC-2	$[E_{act,min}, E_{act,max}]$	-	-	[0.09, 0.14]	[0.14, 0.19]	[0.19, 0.24]	[0.24, 0.30]
			Activation energy $E_{act}(\mathcal{A})$	0.54	0.00036	0.10	0.15	0.21	0.24
			Accuracy (%)	73.91	72.52	72.58	72.86	73.24	**73.54**
	MACAM-2	ADC-1	$[E_{act,min}, E_{act,max}]$	-	-	[0.05, 0.15]	[0.15, 0.25]	[0.25, 0.35]	[0.35, 0.45]
			Activation energy $E_{act}(\mathcal{A})$	1	0.00022	0.12	0.19	0.27	0.39
			Accuracy (%)	73.91	70.84	71.50	72.07	72.13	**72.50**
		ADC-2	$[E_{act,min}, E_{act,max}]$	-	-	[0.09, 0.14]	[0.14, 0.19]	[0.19, 0.24]	[0.24, 0.30]
			Activation energy $E_{act}(\mathcal{A})$	0.54	0.00022	0.10	0.16	0.21	0.26
			Accuracy (%)	73.91	70.84	71.50	71.76	72.42	**73.02**

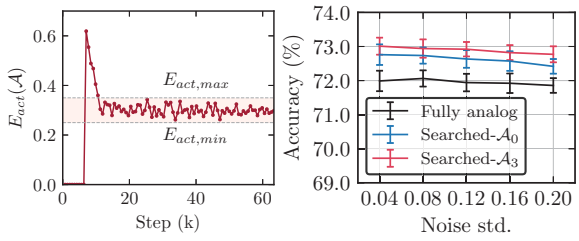
Table 4: Test Accuracy of searched ResNet18 on CIFAR100 with MACAM-1 and two different ADC designs.

Model	MACAM design	ADC design	Metrics	Fully digital	Fully analog	Searched-$\mathcal{A}_0$	Searched-$\mathcal{A}_1$	Searched-$\mathcal{A}_2$	Searched-$\mathcal{A}_3$
ResNet18	MACAM-1	ADC-1	$[E_{act,min}, E_{act,max}]$	-	-	[0.05, 0.15]	[0.15, 0.25]	[0.25, 0.35]	[0.35, 0.45]
			Activation energy $E_{act}(\mathcal{A})$	1	0.00036	0.12	0.20	0.29	0.41
			Accuracy (%)	77.63	76.41	77.01	77.18	77.35	**77.47**
		ADC-2	$[E_{act,min}, E_{act,max}]$	-	-	[0.09, 0.14]	[0.14, 0.19]	[0.19, 0.24]	[0.24, 0.30]
			Activation energy $E_{act}(\mathcal{A})$	0.54	0.00036	0.11	0.16	0.21	0.27
			Accuracy (%)	77.63	76.41	76.85	77.06	77.14	**77.40**



(a) (b)

Figure 9: The layer-wise ratio of channels assigned to the digital activation units (*digi*) on searched models. (a) VGG13 with MACAM-2 and ADC-2. (b) ResNet18 with MACAM-1 and ADC-1.



(a) (b)

Figure 10: (a) Training curve of activation energy cost $E_{act}(\mathcal{A})$ of $\mathcal{A}_2$. (b) Robustness evaluation of Fully analog, $\mathcal{A}_0$, and $\mathcal{A}_3$. Error bars show the $\pm 1 \cdot \sigma$ variance over 20 runs. All models are VGG13 on CIFAR100 with MACAM-1 and ADC-1.

more channels are assigned to the electrical activation units in the latter layers to boost the model accuracy. This demonstrates our `SuperMixer` flow can automatically learn the optimized assignment with boosted expressiveness under energy constraints.

Energy Penalty Curve. In Fig. 10(a), the activation energy $E_{act}(\mathcal{A})$ is visualized along the `SuperMixer` training process. $E_{act}(\mathcal{A})$ is well-bounded in the given constraint. It continues the exploitation of the search space and converges with the temperate τ of the Gumbel Softmax approaching 0.

Noise Robustness of Searched Models. In Fig. 10(b), we evaluate the variation-robustness between searched models and model using fully analog activation units. With the increasing variation on MACAM, our searched models show better noise robustness since of the involvement of electrical activation units.

7 CONCLUSION

In this work, we propose a novel analog and *mixed* activation system for energy-efficient neural network acceleration. We first devise a *fused* analog activation unit based on MACAM that is capable of achieving nonlinear ReLU-α and A/D conversion simultaneously, with superior energy efficiency to conventional digital activation implementation. We further integrate both the analog and digital activation dataflows to create a mixed activation system. A `SuperMixer` training flow is developed to automatically learn how to assign activation workloads to the low-energy analog activation datapath and high-precision digital activation datapath, aiming at a balance of expressiveness and energy efficiency. Our proposed methods are evaluated in a silicon photonic accelerator case study. Compared to the standard activation implementation, our mixed activation system with the searched assignment can achieve competitive accuracy with >60% energy saving on the overall A/D conversion and activation energy cost. Our MACAM-enabled analog and mixed activation system is viable to break through the curse of A/D conversion overhead in analog computing.

ACKNOWLEDGMENT

This work was supported in part by the Multidisciplinary University Research Initiative (MURI) program through the Air Force Office of Scientific Research (AFOSR) contract No. FA 9550-17-1- 0071, and the Samsung GRO. The authors would like to thank Hao Chen, Mahshid Alamdar from The University of Texas at Austin and Xiyuan Tang from Peking University for helpful discussions.

REFERENCES

[1] Alphacore. 2022. Analog, Mixed Signal & RF Solutions. https://www.alphacoreinc.com/en/analog-mixed-signal-and-rf-solutions.

[2] Ping Chi, Shuangchen Li, Cong Xu, Tao Zhang, Jishen Zhao, Yongpan Liu, Yu Wang, and Yuan Xie. 2016. PRIME: A Novel Processing-in-Memory Architecture for Neural Network Computation in ReRAM-Based Main Memory. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*.

[3] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. 2018. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085* (2018).

[4] Kyojin D Choo, John Bell, and Michael P Flynn. 2016. 27.3 Area-efficient 1GS/s 6b SAR ADC with charge-injection-cell-based DAC. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)*.

[5] Teyuh Chou, Wei Tang, Jacob Botimer, and Zhengya Zhang. 2019. CASCADE: Connecting RRAMs to Extend Analog Dataflow In An End-To-End In-Memory Processing Paradigm. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*.

[6] Chenghao Feng, Jiaqi Gu, Hanqing Zhu, Zhoufeng Ying, Zheng Zhao, David Z Pan, and Ray T Chen. 2021. Silicon photonic subspace neural chip for hardware-efficient deep learning. *arXiv preprint arXiv:2111.06705* (2021).

[7] Jiaqi Gu, Chenghao Feng, Hanqing Zhu, Ray T Chen, and David Z Pan. 2022. Light in AI: Toward Efficient Neurocomputing with Optical Neural Networks-A Tutorial. *IEEE Transactions on Circuits and Systems II: Express Briefs* (2022).

[8] Jiaqi Gu, Chenghao Feng, Hanqing Zhu, Zheng Zhao, Zhoufeng Ying, Mingjie Liu, Ray T Chen, and David Z Pan. 2022. SqueezeLight: A Multi-Operand Ring-Based Optical Neural Network with Cross-Layer Scalability. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2022).

[9] Jiaqi Gu, Hanqing Zhu, Chenghao Feng, Zixuan Jiang, Mingjie Liu, Shuhan Zhang, Ray T Chen, and David Z Pan. 2022. ADEPT: Automatic differentiable design of photonic tensor cores. In *2022 59th ACM/IEEE Design Automation Conference (DAC)*.

[10] Seungchul Jung, Hyungwoo Lee, Sungmeen Myung, Hyunsoo Kim, Seung Keun Yoon, Soon-Wan Kwon, Yongmin Ju, Minje Kim, Wooseok Yi, Shinhee Han, et al. 2022. A crossbar array of magnetoresistive memory devices for in-memory computing. *Nature* 601, 7892 (2022), 211–216.

[11] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[12] Thomas Leonard, Samuel Liu, Mahshid Alamdar, Can Cui, Otitoaleke G Akinola, Lin Xue, T Patrick Xiao, Joseph S Friedman, Matthew J Marinella, Christopher H Bennett, et al. 2021. Shape-Dependent Multi-Weight Magnetic Artificial Synapses for Neuromorphic Computing. *arXiv preprint arXiv:2111.11516* (2021).

[13] Can Li, Catherine E Graves, Xia Sheng, Darrin Miller, Martin Foltin, Giacomo Pedretti, and John Paul Strachan. 2020. Analog content-addressable memories with memristors. *Nature communications* 11, 1 (2020), 1–8.

[14] Mengquan Li, Zhongzhi Yu, Yongan Zhang, Yonggan Fu, and Yingyan Lin. 2021. O-HAS: Optical Hardware Accelerator Search for Boosting Both Acceleration Performance and Development Speed. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*.

[15] Samuel Liu, T Patrick Xiao, Can Cui, Jean Anne C Incorvia, Christopher H Bennett, and Matthew J Marinella. 2021. A domain wall-magnetic tunnel junction artificial synapse with notched geometry for accurate and efficient training of deep neural networks. *Applied Physics Letters* 118, 20 (2021), 202405.

[16] Xiaoxiao Liu, Mengjie Mao, Beiye Liu, Hai Li, Yiran Chen, Boxun Li, Yu Wang, Hao Jiang, Mark Barnell, Qing Wu, et al. 2015. RENO: A high-efficient reconfigurable neuromorphic computing accelerator design. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*.

[17] Carl Ramey et al. 2020. Silicon Photonics for Artificial Intelligence Acceleration. In *Proc. HotChips*.

[18] Kaushik Roy, Indranil Chakraborty, Mustafa Ali, Aayush Ankit, and Amogh Agrawal. 2020. In-Memory Computing in Emerging Memory Technologies for Machine Learning: An Overview. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*.

[19] Ali Shafiee, Anirban Nag, Naveen Muralimanohar, Rajeev Balasubramonian, John Paul Strachan, Miao Hu, R Stanley Williams, and Vivek Srikumar. 2016. ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. *ACM SIGARCH Computer Architecture News* 44, 3 (2016), 14–26.

[20] Bhavin J. Shastri, Alexander N. Tait, T. Ferreira de Lima, Wolfram H. P. Pernice, Harish Bhaskaran, C. D. Wright, and Paul R. Prucnal. 2021. Photonics for artificial intelligence and neuromorphic computing. *Nature Photonics* (2021).

[21] Yichen Shen, Nicholas C. Harris, Scott Skirlo, et al. 2017. Deep learning with coherent nanophotonic circuits. *Nature Photonics* (2017).

[22] Tao Song, Xiaoming Chen, Xiaoyu Zhang, and Yinhe Han. 2021. BRAHMS: Beyond Conventional RRAM-based Neural Network Accelerators Using Hybrid Analog Memory System. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*.

[23] Hanbo Sun, Zhenhua Zhu, Yi Cai, Xiaoming Chen, Yu Wang, and Huazhong Yang. 2020. An Energy-Efficient Quantized and Regularized Training Framework For Processing-In-Memory Accelerators. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*.

[24] Febin Sunny, Asif Mirza, Mahdi Nikdast, and Sudeep Pasricha. 2021. CrossLight: A Cross-Layer Optimized Silicon Photonic Neural Network Accelerator. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*.

[25] Alexander N. Tait, Thomas Ferreira de Lima, Ellen Zhou, et al. 2017. Neuromorphic photonic networks using silicon photonic weight banks. *Sci. Rep.* (2017).

[26] Zhongrui Wang, Huaqiang Wu, Geoffrey W Burr, Cheol Seong Hwang, Kang L Wang, Qiangfei Xia, and J Joshua Yang. 2020. Resistive switching materials for information processing. *Nature Reviews Materials* 5, 3 (2020), 173–195.

[27] Qilin Zheng, Zongwei Wang, Zishun Feng, Bonan Yan, Yimao Cai, Ru Huang, Yiran Chen, Chia-Lin Yang, and Hai Helen Li. 2020. Lattice: An ADC/DAC-less ReRAM-based Processing-In-Memory Architecture for Accelerating Deep Convolution Neural Networks. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*.

[28] Hanqing Zhu, Jiaqi Gu, Chenghao Feng, Mingjie Liu, Zixuan Jiang, Ray T Chen, and David Z Pan. 2022. ELight: Towards Efficient and Aging-Resilient Photonic In-Memory Neurocomputing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2022).