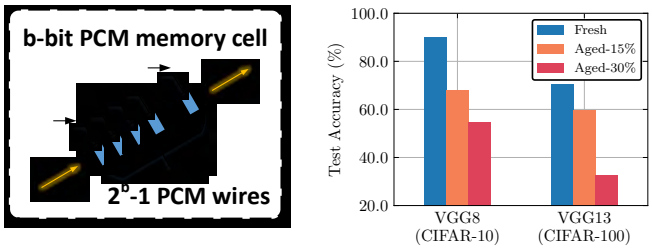
ELight: Towards Efficient and Aging-Resilient Photonic In-Memory Neurocomputing

Hanqing Zhu, *Graduate Student Member, IEEE*, Jiaqi Gu, *Graduate Student Member, IEEE*, Chenghao Feng, *Graduate Student Member, IEEE*, Mingjie Liu, *Graduate Student Member, IEEE*, Zixuan Jiang, Ray T. Chen, *Fellow, IEEE*, and David Z. Pan, *Fellow, IEEE*

***Abstract*— Optical phase change material (PCM) has emerged promising to enable photonic in-memory neurocomputing in optical neural network (ONN) designs. However, massive photonic tensor core (PTC) reuse is required to implement large matrix multiplication due to the limited single-core scale. The resultant large number of PCM writes during inference incurs serious dynamic energy costs and overwhelms the fragile PCM with limited write endurance, causing the severe aging issue. Moreover, the aged PCM would distort the stored value and significantly degrade the reliability of PTC. In this work, we propose a holistic solution, `ELight`, to tackle both the aging issue and the post-aging reliability issue, where a proactive aging-aware optimization framework minimizes the overall PCM write cost and a post-aging tolerance scheme overcomes the effect of aged PCM. Specifically, in the aging-aware optimization part, we propose write-aware training to encourage the similarity among weight blocks and combine it with a post-training optimization technique to reduce programming efforts by eliminating redundant writes. Next, an efficient group-wise row-based weight-PTC remapping scheme is introduced to tolerate the reprogrammability degradation due to the aged PCM. Experiments show that `ELight` can achieve over $20\times$ reductions in the total number of write operations and dynamic energy cost with comparable accuracy. Moreover, `ELight` can guarantee significant accuracy recovery under the aged PCM within photonic memories. With our `ELight`, photonic in-memory neurocomputing will step forward towards practical applications in machine learning with order-of-magnitude longer lifetime, lower programming energy cost, and significant resilience against PCM aging effects.**

I. Introduction

As Moore's Law winds down, it becomes challenging for conventional electrical computers to win in an arms race with the massively growing computation demands of machine learning applications. In recent years, optical neural networks (ONNs) [1]–[12] has been demonstrated as a paradigm shift in efficient neurocomputing due to its traits of sub-nanosecond latency, ultra-high energy efficiency, and ultra-high bandwidth. Recent work [13]–[15] demonstrates that phase change material (PCM) can be used to build photonic tensor core (PTC) for optical in-memory matrix multiplication. PCM can undergo nonvolatile modulation of optical properties as a weight encoding mechanism. In-memory multiplication is implemented with near-zero static power by shining light through waveguides integrated with configured PCMs, where light-matter interactions will change the amount of transmitted light passively. Moreover, the broadband transmission of PCM enables massive computing parallelism with wavelength-division multiplexing (WDM) techniques, With the above superiority, PCM-based ONN designs open a new pathway towards efficient in-memory neurocomputing via optics.



(a) (b)

Fig. 1: (a) Multi-bit photonic memory cell based on PCM wires. (b) Accuracy drop of models under a fixed ratio of aged cells.

However, before PCM-based photonic in-memory neurocomputing can make a truly viable efficient inference acceleration, there are still practical barriers that lie ahead. Firstly, the supported bit-width on the current PCM cell designs remains to be limited. In [13], a reasonable implementation of b-bit PCM cell is proposed for PTC with low programming complexity and demonstrated $4\sim5$-bit data imprint, where 2^b-1 identical PCM wires are patterned on the same waveguide as shown in Fig. 1a. Each PCM wire is electrothermally switched to represent binary phase states by complete amorphous-crystalline phase transition, thus presenting total 2^b nonlinear transmission levels. To suit the unique transmissivity distribution, a specialized quantization strategy is in high demand to save the ONN accuracy on the above low-precision PCM-based PTC.

Besides, the potential frequent weight reprogramming during inference also raises critical issues in PCM-based photonic in-memory computing. Massive reuse of PTCs is required due to the limited scale of PTC compared to the weight matrix, e.g., a 64×64 PTC is already quite large while the largest convolutional layer in ResNet-18 [16] can have a $512\times512\times3\times3$ weight matrix. However, the endurance of PCM is a limiting factor, ranging from 10^6 to 10^8 [17]. The resultant massive weight updates in PTCs potentially threaten PCM wires at a high risk of over-utilization, i.e., the *aging* issue. Once PCM

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 by NSF under Award #1718570. The preliminary version has been accepted by the ACM/IEEE Asian and South Pacific Design Automation Conference (ASP-DAC) in 2022.

H. Zhu, J. Gu, C. Feng, M. Liu, Z. Jiang, R. T. Chen and D. Z. Pan are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, TX, USA (e-mail: hqzhu@utexas.edu).

wires are aged, the physical value will deviate from the desired value as the loss of reprogrammability, causing *post-aging* reliability issue and severe accuracy drop as shown in Fig. 1b. Moreover, the massive re-writes incur a non-trivial dynamic energy cost, which dilutes the energy efficiency benefits from PCM. The above issues are related to two key metrics: (1) The total number of write operations to the PCM wires (*# total writes*); (2) The maximum number of write operations over a single PCM wire (*# max writes*). Dedicated optimization mechanisms are, therefore, needed to reduce the two metrics as a proactive solution to mitigate *aging* issue. To further prolong the executing lifetime of PTCs, a *post-aging* tolerance method is demanded as well to recover the accuracy drop when ONN is mapped onto the unreliable aged PTCs.

In this work, we propose a holistic solution, `ELight`, to enable efficient, aging-resilient, and life-prolonging photonic in-memory neurocomputing, including a proactive *aging-aware* optimization framework and a *post-aging* tolerance scheme. Based on an augmented redundant write elimination strategy, we devote ourselves to trimming down *# total writes* and *# max writes* by eliminating redundant writes on PCM wires during weight updates, and thus mitigate *aging* issue. Considering the block pattern of weight reloading in PTCs, a write-aware training method is first proposed to orchestrate the higher weight similarity among weight blocks to increase the eliminable redundant writes. Then the post-training optimization is applied to reduce write operations further. Besides the above techniques, to capacitate resilience against already-aged PCM wires, we extend `ELight` [18] with an efficient *post-aging* tolerance scheme, the group-wise row-based weight-PTC remapping, to preserve the accuracy of ONN with marginal overhead. The main contributions of this paper are listed as follows.

- **Distribution-Aware Quantization** scheme is introduced to fit the modeled unique transmissivity distribution with reduced weight encoding errors on PCM cells.
- **Write-Aware Training** is designed to boost block-wise weight similarity with negligible accuracy effect so as to increase eliminable redundant write operations during weight updates.
- **Post-Training Optimization** is proposed to further cut down redundant writes via one-shot fine-grained reordering, without changing the model output.
- **Group-wise Row-based Remapping** is proposed to tolerate aged PCM wires within PTCs via re-configuring weight-PTC row mapping in a group-wise manner, which saves ONN accuracy with marginal hardware overhead.
- To the best of our knowledge, this is the *first work* that handles the *aging* issue of optical PCM in photonic neural engines, achieving over $20\times$ reduction in the total number of reprogramming operations and dynamic energy cost during inference. Our extended *post-aging* tolerance scheme further enhances the reliability of the novel in-memory neurocomputing paradigm under aged PCM wires with significant accuracy recovering. The augmented version of `ELight` successfully provides a holistic solution to tackle both *aging* issue and *post-aging* issue in photonic in-memory neurocomputing. With ELight, PCM-based photonic in-memory neurocomputing will benefit from an order-of-magnitude longer lifetime.

The rest of this paper is organized as follows. Section II introduces the basics of PCM and photonic tensor core. Section III details the efforts to model the transmissivity distribution of photonic memory cells, followed by a distribution-aware quantization to reduce weight encoding errors. Section IV discusses details of our proposed *aging-aware* optimization framework to obtain *aging-aware* models. Section V describes our extended *post-aging* tolerance scheme for aged PCM wires. Section VI evaluates our proposed methods. VIII concludes this paper.

II. Preliminaries

This section introduces the basics of phase change material, the architecture of photonic tensor core, and barriers of current PCM-based ONNs towards practical deployment.

A. Basics of Phase Change Material (PCM)

As a promising memristive device, phase change material (PCM) has emerged as an attractive device for photonic in-memory computing. The optical characteristics of PCM change drastically when undergoing an amorphous-crystalline phase transition, where high-light-absorption crystalline (c) state and low-light-absorption amorphous (a) state represent the logical '0' and logical '1', respectively. The programmable non-volatile states underpin PCM's potential to perform ultra-fast in-memory multiplication [13], [14]. By shining light through the waveguide integrated with PCM cells on top and configuring the transmission factor t via switching PCM states, the transmitted optical power can be modulated as $P_{out} = t \cdot P_{in}$ such that the scalar multiplication can be implemented. Yet, high-precision data imprint is not supported in current PCM cell designs with a limited number of realized distinct transmission levels. In [13], an implementation of b-bit PCM cells is proposed for photonic in-memory neurocomputing, where 2^b-1 binary PCM wires are patterned on one waveguide to promise 2^b transmission levels. 4$\sim$5-bit storage ability is validated in [13]. Considering the programming complexity and binary photonic PCM devices' practicality, we will focus on this PCM cell design in this paper.

However, the lifetime of PCM in photonic devices remains to be improved, where a maximum of $\sim 10^8$ total reprogramming times [17], [19] is measured. Frequent value updating will over-utilize PCM, lose its reprogrammability, and reduce its reliability, thus shortening the lifetime of PCM. Moreover, photonic PCM failure mechanisms have not been characterized yet [17], impeding optimization techniques from the perspective of failure rules. Therefore, to boost the lifetime of PCM, a viable direction is to reduce write operations to the PCM.

B. Architecture of Photonic Tensor Core

Recent work [13], [14] demonstrate the implementation of photonic tensor cores for optical in-memory matrix multiplication, i.e, $Y = WX + b$. Both utilize the photonic PCM

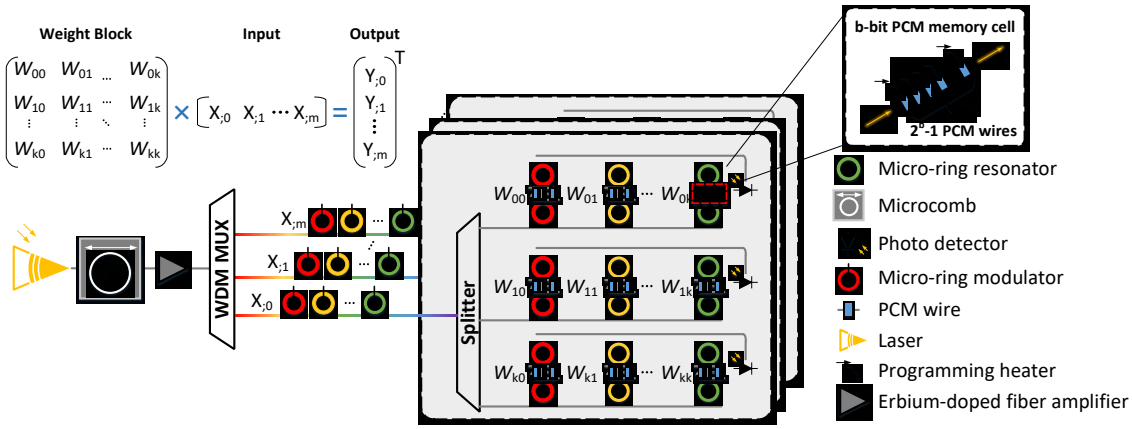


Fig. 2: The architecture of photonic tensor cores based on phase-change material.

arrays' storage and light interactions capability, with W being encoded in the PCM states.

Consider the matrix-matrix multiplication (MM) between the weight block $W \in \mathbb{R}^{k\times k}$ and the input matrix $X \in \mathbb{R}^{k\times m}$. Fig. 2 illustrates one architecture of PCM-based PTC [13]. Each PCM array carries out the matrix-vector multiplication (MVM) and each PTC duplicates PCM array m times to achieve MM. An input laser and an on-chip frequency comb work together to generate light with multiple wavelengths $(\lambda_0, \lambda_1, \cdots)$. A WDM multiplexer is then used to evenly distribute the light into m rows. Each row consists of a series of narrowband micro-ring modulators to encode one column of the input matrix X in the power of the optical signals. Then, a PCM array with k rows and k columns stores the weight block and processes the MVM between encoded W and one column of X. Concretely, in the i-th rail, the inputs are filtered by the on-resonance micro-rings based on wavelength and weighted via light-matter interaction with the PCM. Photo-detectors are put at the end of the drop port to accumulate the intensity of the WDM optical signals, i.e., the weighted inputs, and generate the desired result $Y_{im} = \sum_j^k W_{ij} X_{jm}$.

In Fig. 2, the b-bit PCM memory cell is used to store weights in a non-volatile way. 2^b-1 binary PCM wires are placed on the top of the same waveguide to allow 2^b transmission levels, thus enabling b-bit weight storage. The PCM wires within the photonic memory cell are selectively switched between crystalline and amorphous phase states to imprint the desired value. The a-c and c-a transition of PCM wires can be achieved via electrothermal heating in parallel by simultaneously sending electrical pulses to individual thermal heaters. For example, to encode '1100' in a 4-bit photonic memory cell, twelve out of fifteen PCM wires are randomly chosen to be set to the high-light-transmission amorphous state while the others are programmed to crystalline state.

C. Barriers in Practical Deployment of PCM-based ONNs

Photonic in-memory neurocomputing has gained much attention as a promising next-generation platform for machine learning (ML) workloads. However, several practical challenges still lie ahead towards real applications. First, the limited programming resolution and unique discretized transmission levels of PCM-based PTCs call for a specialized quantization scheme. Besides, the limited scale of PTC cannot promise the one-shot realization of large matrix multiplication. Considering the area cost and light loss [14], the scale of PTC cannot be too large, where a 64×64 PTC is already quite large. Hence, to implement large matrix multiplication during inference, massive reuse of PTCs is needed, causing frequent weight value updates within PTCs. Consequently, PCM wires with limited endurance are threatened to be over-utilized, thus shortening the executing lifetime of the computing engine as a key limiting factor. Moreover, massive reprogramming over PCM incurs significant dynamic power during inference, raising concerns about PCM's energy superiority. Previous works [20]–[23] on emerging neuromorphic computing systems such as ReRAM mainly focus on minimizing frequent weight updates when on-chip training is performed. In this work, aware of the curse of the limited PTC scale, we keep our emphasis on the potential frequent weight reprogramming during the inference process, which plagues the lifetime of photonic in-memory neurocomputing.

To tackle the above issues, two key metrics are desired to be optimized. The total number of write operations to PCM (*# total writes*) reflects the averaged degree of utilization of PCM wires and the dynamic energy cost. The maximum number of wire write operations over a single photonic cell (*# max writes*) can represent the status of the most over-utilized memory cell. Redundant Write Elimination (RWE) strategy [24]–[27] is widely used as a basic optimization scheme during PCM-based Phase Change Memory programming to reduce the number of write operations. Since data are represented by the binary-state PCM wires within photonic memory cells, in this paper, we augment the RWE strategy in a more fine-grained granularity by identifying identical writes over PCM wires and eliminating them. If we assume that the PTC is totally unusable after a major portion of PCM wires wore out, in this case, suppose the write endurance of PCM wire is 1×10^7 and the averaged degree of utilization of PCM wires is 10^3 per inference pass for a given model. delete "In that case" If the PTC is employed to perform 10^2 inference

tasks per day as the edge device, the lifetime of PTC can be prolonged from 100 days to 2000 days if we can reduce the required write operations by 20×.

Other nonideal defects such as the device-to-device variation and temporal drift are also critical challenges for PCM-based ONNs. However, related models and mechanisms have not been well-characterized in photonic PCM devices [17]. We are interested in solving these challenges in the future, while in this paper, we keep our primary emphasis on the endurance issue of PCM in PTCs.

III. Proposed Distribution-Aware Quantization Scheme

In this section, we give out a dedicated *distribution-aware* quantization scheme based on the analysis of transmissivity distribution of the adopted b-bit PCM memory cell design, to reduce weight encoding errors.

A. Transmission Model of Multi-Level PCM Memory Cell

As discussed above, the light transmissivity of a b-bit photonic memory cell is determined by the phase states of 2^b-1 PCM wires. Assuming that transmission level i refers to the transmissivity when i wires in the photonic memory cell are programmed to c state while the others are written to a state, its extinction ratio (ER) is computed as the ratio of the transmitted optical power in level i and level 0, i.e., $10\log_{10}(\frac{P_i}{P_0})$. As demonstrated in [13], the ER uniformly increases as a function of i with a step Δe. Given that the transmitted optical power is the product of the transmission factor and input light power, ER can be further expressed as

$$10\log_{10}(\frac{t_i \cdot P_{in}}{t_0 \cdot P_{in}}) = 10\log_{10}(\frac{t_0 \cdot P_{in}}{t_0 \cdot P_{in}}) + i\Delta e. \tag{1}$$

where t_i is the transmission factor in the i-th transmission level. We can further derive t_i as

$$t_i = t_0 \cdot 10^{\frac{1}{10}\Delta e \times i} = t_0 \times c^i,\ c = 10^{\frac{1}{10}\Delta e}. \tag{2}$$

Here, c represents the percentage of light power transmitted through one PCM wire, which is less than 1. The 0-th transmission level corresponds to all wires being in a (logic '1') state, where t_0 is approximately 1 [13].

Hence, the distribution of 2^b transmission levels of the b-bit PCM memory cell is finally formulated as an *exponential model*,

$$t_i = c^i,\ i = 0, 1, \ldots, 2^b - 1. \tag{3}$$

B. Augmented Base-c Quantization

To accomplish computing within PCM-based photonic memories, we first need to effectively map full-precision weights w to the discrete transmission levels of memory cells with low encoding error. However, traditional uniform quantization with a uniformly divided quantization interval fails to fit the unique exponential transmissivity distribution, which would cause severe encoding error. Therefore, a dedicated quantizer $q(w, b)$ is demanded to minimize the error between the actual value of w and its quantized value $\hat{w}$ as follows,

$$\min\ \|\hat{w} - w\|_2^2, \quad \text{s.t.}\ \ \hat{w} = q(w, b) \in Q_b, \tag{4}$$

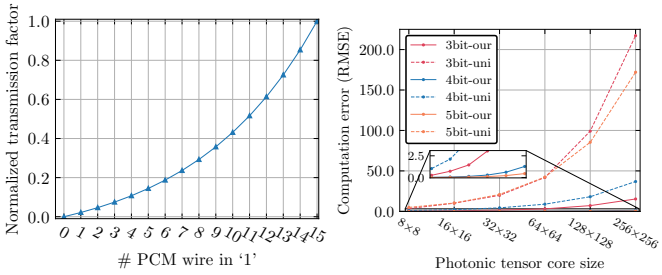


Fig. 3: (a) Normalized transmissivity distribution of a 4-bit photonic memory cell. (b) Computing Root-Mean-Square Error (RMSE) of matrix multiplication under uniform and our distribution-aware quantizer.

where b is the bit-width. Q_b denotes a set of quantization levels, i.e., the transmission levels of a b-bit photonic memory cell.

However, as only positive light transmission can be demonstrated in PCM, to support full-range weight, we adopt the positive PTCs and negative PTCs to store the positive and negative values of weight matrix W, respectively. Then, the differential photo-detection module will generate balanced output. Based on the simple but effective *weight extension strategy*, each scalar weight w in W is expressed as

$$w = w_{pos} - w_{neg}. \tag{5}$$

Here, w_{pos} and w_{neg} are physical values in positive and negative photonic memory cells, where one is chosen in terms of the sign of w to store the value of w while the other is programmed to the lowest light transmission level $\delta = c^{2^b-1}$. In such manner, the quantization codebook Q_b is augmented with the differential weight encoding mechanism in (5) as follows,

$$Q_b = \{c^{2^b-1} - \delta, \pm(c^{2^b-2} - \delta), \ldots, \pm(c^0 - \delta)\},\ \delta = c^{2^b-1}. \tag{6}$$

The number of implementable quantization levels in Q_b is almost doubled, which promises a higher model expressivity under a low-bit quantization scheme.

With the augmented quantization codebook, for w within $[-1, 1]$, an augmented base-c quantizer is hence proposed to optimize (4) as

$$w_q = q(w, b) = \frac{\texttt{sign}(w)}{s} \cdot (c^{\texttt{Clip}(\texttt{R}(\log_c(s|w|+\delta)),\ 0,\ 2^b-1)} - \delta), \tag{7}$$

where $\texttt{R}(\cdot)$ is a round function. The quantized value is transformed into $[-1, 1]$ with the scaling factor $s = c^0 - c^{2^b-1}$. Our augmented base-c quantizer is implemented within a quantization-aware training procedure [28] to train the PCM-based ONNs. In the forward pass, the weight w in W are quantized as

$$w_q = q(\frac{\texttt{Tanh}(w)}{\max(\texttt{Tanh}(W))}, b),\ b > 1. \tag{8}$$

During the backward propagation, the whole b-bit quantization process $q(w, b)$ is coarsened as an entirety, where its gradient g_q is estimated by Straight Through Estimation (STE) [29] as

$$g_q = \frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial W_q}\frac{\partial W_q}{\partial W} = \frac{\partial \mathcal{L}}{\partial W_q}. \tag{9}$$

The layer input is also discretized by a uniform quantizer in [30] considering limited on-chip storage.

Fig. 3b depicts the computing root-mean-square errors of randomly generated 1500 matrix multiplications on different PTC scales and bit-widths under our proposed distribution-aware quantization method and traditional uniform quantization method. Our quantization method achieves significantly minor computing errors with a better fit for the unique non-linear transmissivity distribution of photonic memory cells.

IV. Proposed Aging-Aware Co-Optimization Framework

In this section, we introduce the proposed *aging-aware* optimization framework of `ELight` to minimize both *# total writes* and *# max writes*. We first illustrate the adopted *augmented redundant write elimination* (ARWE) strategy for PCM wire-level write elimination and formulate the optimization targets. Then we describe the proposed *write-aware training* method to encourage the similarity among weight blocks. At last, a *fine-grained reordering* method is proposed to work as a one-shot post-training optimization to further trim down redundant writes.

A. Problem Formulation

Matrix multiplication operations in convolutional layers and fully-connected layers are accelerated on PCM-based PTCs. Especially, to convert convolutions in convolutional layers into general matrix multiplication (GEMM), an im2col algorithm [31] is used. Given the limited single photonic tensor core size, blocking matrix multiplications is adopted for practical considerations. The weight matrix $W \in \mathbb{R}^{M\times N}$ is partitioned into $P\times Q$ sub-matrices, where each $k\times k$ block can be deployed onto one PTC. Since we have $M, N \gg k$, the number of sub-matrices is quite large. Moreover, when assigning the set B of sub-matrices to a cluster C of on-chip PTCs, as the number of photonic tensor cores is limited, one PTC is assigned with multiple sub-matrices, causing massive PTC reuse during inference. Here, without loss of generality, a simple assignment strategy to assign sub-matrices to PTCs is adopted for the following discussion. To complete the computation of one layer with $P \times Q$ weight sub-matrices, a cluster of PTCs is dedicated for the MMs, where a row of weight blocks is assigned to the same PTC. In this way, P PTCs carry out blocking MM in parallel with shared input. For instance, suppose that the scale of PTC is 64×64, the weight matrix of the 5^{th} convolutional layer of VGG8 is partitioned into a set B with 8×72 blocks, where each PTC is assigned with 72 blocks. We assume the data stored in PTC is updated only after all the block MMs on the current stored block are completed such that reprogramming cost are reduced. Note that our proposed methods in this article can work with other assignment schemes.

To trim down write operations during weight updates, we propose an *augmented* redundant write elimination (ARWE) strategy in a more fine-grained way than the vanilla RWE strategy [24]. As data are represented by the binary-state PCM wires within photonic memories, we identify the identical wire-level writes first and then exclude these identical parts during reprogramming. Concretely, instead of directly cleaning the old value and writing the new data, we preserve the current states of PCM wires at the largest extent and only perturb the smallest number of wires to demonstrate desired value. For a clear illustration of the ARWE strategy, Fig. 4 shows one example of writing a sequence of weight blocks into 3-bit PTCs. One positive photonic tensor core and one negative photonic tensor core are used together to demonstrate full-range weights. For example, to write transmission level +7 into the memory cell with a stored +5, two c-state PCM wires in the positive PTC are re-written to a states (logical '0'→logical '1') such that the smallest number of writes is achieved. Besides, to minimize the influence of temporal drift, the ARWE strategy is only adopted within the execution cycle of one layer. When implementing computations for the next layer, we first initialize all PTCs by setting all PCM wires to c state and then write the next layer's weights into photonic memories. To program weights based on the ARWE strategy, we need to compare the new value and the original value to identify identical parts. Instead of detecting stored value in the RWE strategy [24]–[27], since we exactly know the programming profile of each memory cell, i.e., the binary state of each PCM wire within the memory cell, we choose to store the state of PCM wires at the current time to help determine the programming solution for the next time. The required on-chip memory overhead is acceptable given the limited number of on-chip PTCs, the limited scale of single PTC, and the limited number of PCM wires within PCM memory cells.

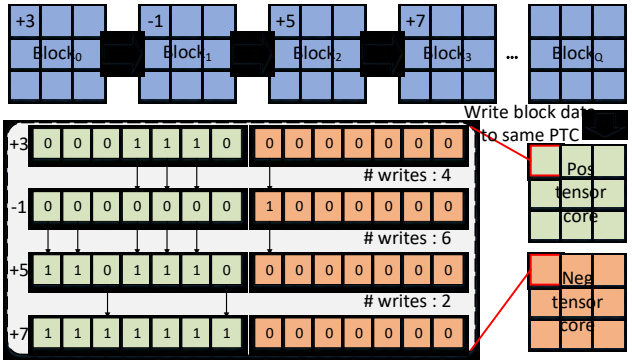


Fig. 4: The illustration of writing multiple weight blocks in the same PTC with redundant wire writes elimination. Each PCM wire within 3-bit memory is in a(1) or c(0) state to demonstrate quantization levels in -7∼+7. Wire-level redundant writes are excluded to reduce *# writes*.

Thus, considering write cost in both positive and negative PTCs, the number of writes (WT) between two b-bit numbers $w^{'}$ and w is computed as follows,

$$WT(w^{'}, w) = |l^{+}(w^{'}) - l^{+}(w)| + |l^{-}(w^{'}) - l^{-}(w)|, \tag{10}$$

where l^{+} and l^{-} denote the transmission levels in positive and negative PTCs, respectively. The absolute value of l^{+} and l^{-} also represent the number of a-state wires out of $2^{b}-1$ wires in the corresponding photonic memory cell, which can be

TABLE I: Layer-wise statistics of writes of 5-bit VGG8 model.

	Layer 2	Layer 3	Layer 4	Layer 5
# total writes	1.14×10^6	4.87×10^6	1.66×10^7	3.26×10^7
# max writes	294	534	914	1425

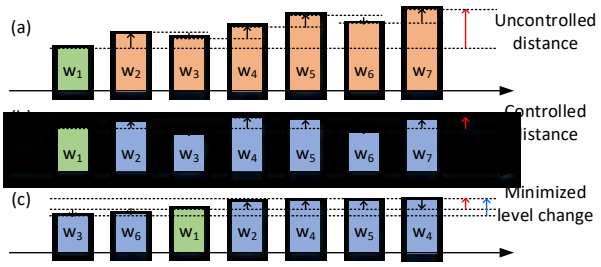


Fig. 5: An example of constraining a weight sequence starting from w_1. (a) Constrain the distance between neighbors. (b) Constrain weights around reference value (w_1). (c) Sort weights in (b) in an ascending order.

computed based on (7) as

$$l^+(w) = \begin{cases} (2^b-1) - \texttt{Clip}(\texttt{R}(\log_t(s|w|+\delta)), 0, 2^b-1), & w \geq 0 \\ 0, & w < 0 \end{cases} \tag{11}$$

$$l^-(w) = \begin{cases} 0, & w \geq 0 \\ \texttt{Clip}(\texttt{R}(\log_t(s|w|+\delta)), 0, 2^b-1) - (2^b-1), & w < 0 \end{cases}. \tag{12}$$

We can further merge the transmission levels $l^+(w)$ and $l^-(w)$ to define the combined transmission level $l(w)$ for w as $l^+(w) - l^-(w)$, ranging from $-(2^b-1)$ to (2^b-1).

Accordingly, when writing new block $A^{'}$ to photonic memories storing block A, the total number of write operations is counted as

$$WT(A^{'},A) = \sum_i^k \sum_j^k (|l^+(a_{ij}^{'}) - l^+(a_{ij})| + |l^-(a_{ij}^{'}) - l^-(a_{ij})|). \tag{13}$$

where the block size is $k \times k$.

Now consider the number of write operations to implement one layer (LWT). Assuming the weight matrix W in layer j is partitioned into a set B of multiple $k \times k$ weight sub-matrices, each PTC t in a cluster C is assigned with n_t blocks, i.e., $\{B_1^t, B_2^t, \ldots, B_{n_t}^t\}$. In our adopted assignment scheme, n_t is equal to the number of blocks in one row of the partitioned weight matrix. Then, the total number of write operations for layer j can be derived as

$$LWT^j = \sum_t^C \sum_i^{n_t} WT(B_i^t, B_{i-1}^t), \tag{14}$$

where PTCs are initialized with all PCM wires being a state before weight blocks are mapped onto.

Hence, when deploying a model, the total number of write operations is computed as the sum of layer-wise write operations, i.e., $\#total\ writes = \sum_j^L LWT^j$. To precisely reflect the status of the most over-utilized memory cell, we define a layer-wise metric, *# max writes*, which counts the maximum number of write operations to one single memory cell among PTCs. However, although our ARWE strategy is applied, a significant number of write operations are still observed. In Table I, the statistics of *# total writes* and *# max writes* for the convolutional layers of 5-bit VGG8 is shown, where massive write operations challenge the PCM endurance. Therefore, in the following discussion, several techniques are introduced to minimize both *# total writes* and *# max writes*.

B. Write-Aware Training via Block Matching

When mapping a group of weight blocks, weight blocks are desired to be similar to introduce more eliminable redundant writes such that the number of write operations can be reduced, which can be carefully handled during model training. Figure 5 shows one simple example of minimizing write operations when a sequence of weights $w_1, ..., w_7$ is programmed to one memory cell. One straightforward method is to pull the neighboring weights closer, as shown in Fig. 5(a). Neighboring weights are constrained to be similar by penalizing large weight distances. Therefore, more redundant writes are boosted. However, the distance between the largest and smallest values is not promised to be minimized as no such constraint is put to constrain the value range of the weight sequence. This might lead more PCM wires within photonic memories to be written as photonic memory cells need to represent a wider value range. Hence, a wise method is to constrain weights around a reference value such that the weights are drawn to be similar and the value range is constrained, shown in Fig. 5(b).

Motivated by this, we propose a write-aware training procedure to boost the block-level weight similarity, shown as Phase 1 in Fig. 6. For weight blocks assigned to the same PTC, we first average them as the reference block and then penalize their transmission level distance from the reference block. Instead of using Eq. (13) to directly optimize the transmission level difference between blocks in an L1 way, we recalculate the level difference (LD) between block W and W^{ref} by using an L2 regularization term as,

$$LD(W^{ref}, W) = \sum_i^k \sum_j^k \|\tilde{l}^+(w_{ij}^{ref}) - \tilde{l}^+(w_{ij})\|^2 + \|\tilde{l}^-(w_{ij}^{ref}) - \tilde{l}^-(w_{ij})\|^2. \tag{15}$$

Here, transmission level l^+ and l^- are normalized by dividing $\alpha_b = 2^b - 1$ to obtain transmission level $\tilde{l}^+$ and $\tilde{l}^-$, which are in $[-1, 1]$. Normalizing level data can help healthy gradient propagation. The choice of L2 regularization term is to ensure the model expressivity, where the L2 regularization term heavily penalizes large value distance, but the slight deviation is allowed to make weights diverse enough. In this way, not only *# max writes* can be optimized through rejecting large value deviation, but the model expressivity can be mostly maintained. Using an L1 regularization term in Eq. (13) would put a hard push to diminish the subtle difference, harm the model expressivity, and make the model sensitive to the choice of the regularization strength.

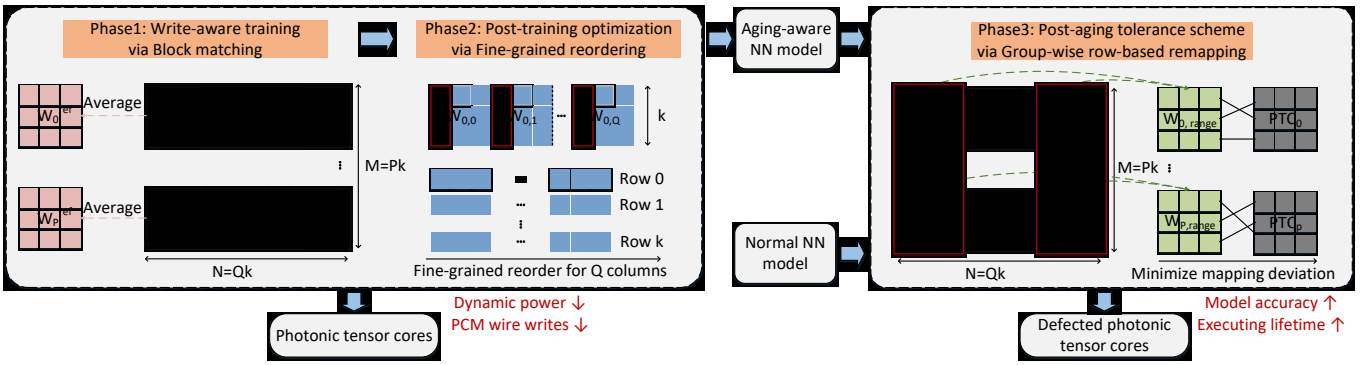


Fig. 6: Proposed three-phase ELight to enable efficient and aging-resilient photonic in-memory neurocomputing.

With the modified LD, the block matching loss is defined as

$$\mathcal{L}_{BM} = \sum_{l}^{L}\sum_{t}^{G}\sum_{i}^{n_g} \frac{1}{\beta^B} LD(B_i^t, B_{avr}^t). \tag{16}$$

The weight blocks is assigned to the same group if they are assigned to the same PTC, where the similarity among them is explicitly orchestrated. β^B is the block size to normalize the level distance.

By controlling λ, we trade off between the accuracy and block similarity with the modified loss function,

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{BM}, \tag{17}$$

where $\mathcal{L}_{CE}$ is the original cross-entropy loss.

However, there are three issues to optimize $\mathcal{L}_{BM}$. First, the Round$(\cdot)$ operations in Eq. (11) and Eq. (12) are not differentiable, making the $\mathcal{L}_{BM}$ not differentiable as well. Second, the gradient approximation through the *logarithmic* operations needs to be carefully handled. Third, when computing LD, level differences in both positive and negative PTCs are countered, while the physical levels of weights are only implemented on either positive or negative PTCs based on the sign. Thus, only those physically implemented levels need to be involved in gradient evaluation. In other words, only the gradient from either $\|\tilde{l}^+(w) - \tilde{l}^+(\delta)\|^2$ or $\|\tilde{l}^-(w) - \tilde{l}^-(\delta)\|^2$ need to be propagated back to compute the gradient w.r.t w depending on its sign. Considering the above issues, with straight-through-estimator (STE) to approximate the gradients for Round$(\cdot)$, the gradient of $\mathcal{L}_{BM}$ is propagated back as,

$$\frac{\partial \mathcal{L}_{BM}}{\partial W} = \frac{1}{\beta^B}\Big(\frac{\partial \mathcal{L}_{BM}}{\partial \tilde{l}^+(W)}\frac{\mathrm{d}\tilde{l}^+(W)}{\mathrm{d}W}\odot\mathcal{M}^+ + \frac{\partial \mathcal{L}_{BM}}{\partial \tilde{l}^-(W)}\frac{\mathrm{d}\tilde{l}^-(W)}{\mathrm{d}W}\odot\mathcal{M}^-\Big), \tag{18}$$

where $\mathcal{M}^+$ and $\mathcal{M}^-$ are non-negative and negative masks of W to extract the needed gradients from $\frac{\mathrm{d}\tilde{l}^+(W)}{\mathrm{d}W}$ and $\frac{\mathrm{d}\tilde{l}^-(W)}{\mathrm{d}W}$, computed by,

$$\frac{\mathrm{d}\tilde{l}^+(W)}{\mathrm{d}W}\Big|_{W\geq 0} = \frac{-\mathrm{d}(\log_t(s|W|+\delta))}{\alpha_b \mathrm{d}W} = \frac{-s}{\alpha_b \ln(t)(s|W|+\delta)}, \tag{19}$$

$$\frac{\mathrm{d}\tilde{l}^-(W)}{\mathrm{d}W}\Big|_{W<0} = \frac{\mathrm{d}(\log_t(s|W|+\delta))}{\alpha_b \mathrm{d}W} = \frac{-s}{\alpha_b \ln(t)(s|W|+\delta)}. \tag{20}$$

C. Post-Training Optimization via Fine-grained Reordering

The proposed *write-aware training* successfully boosts the similarity among weight blocks and thus helps the redundant write elimination strategy. However, it doesn't consider the mapping order of weight blocks, while the write operations strongly rely on the mapping order. We still take Fig. 5(b) as an example. The sum of neighboring distances is not explicitly optimized as all weights are only limited around one reference value. There is still room for further optimization by considering the order. Given the optimized weight sequence, by sorting the weight sequence in either ascending or descending order, the sum of neighboring distances can be further reduced, shown in Fig. 5(c).

Inspired by this heuristic, we propose a *fine-grained reordering* method to sort weights located at the same position of weight blocks so as to minimize reprogramming cost, as they share the same photonic memory cells in PTCs. The Phase 2 in Fig. 6 illustrates our idea. We perform the sorting within the group of weight blocks assigned to one PTC. Weights in the same position are first shaped into 1-D sequences. Then weight sequences are separately sorted in either ascending order or descending order. Finally, The weights are scattered back to weight blocks in the new order. The above process does not affect the final results as it is equivalent to swapping columns element-wise. With the aid of the sorting heuristic, weights are written into PTCs in an optimized order, i.e., ascending or descending order, augmenting the write operation reductions with the redundant write elimination strategy. Moreover, *# max writes* over a single photonic memory cell is upper-bounded by the level range of the mapped weights, i.e., $2^{b+1}-1$.

To this end, we propose a joint optimization flow with write-aware training and one-shot post-training optimization. *# total writes* and *# max writes* can be significantly reduced to mitigate the aging issue and the tedious programming cost.

V. Extended Post-Aging Tolerance Scheme

To handle the reliability issue against aged PCM wires within photonic memories, we further equip our framework ELight with a *post-aging* tolerance scheme, in which an efficient row-based weight-PTC remapping method is proposed to find the unified optimal solution to mapping a group of weight blocks to defected PTCs with aged PCM wires inside.

A. Post-aging Reliability Issue

As discussed before, the write endurance of PCM in photonic devices is limited with a measured maximum of

$\sim 10^8$ write budget [19]. When utilizing the novel PCM-based PTCs as an inference acceleration engine, frequent weight updates within photonic memories would wear out PCM wires. Once PCM is aged, as PCM's failure mechanisms and modes in photonic devices are not clear yet [17], the phase state of aged PCM is complex and hard to analyze. In this sense, aged PCM wires within photonic memories would cause uncontrolled drifts from desired stored values, distort in-memory multiplication results, and thus cause the *post-aging* reliability issue of defected PTCs.

As not all PCM wires in the memory cells wear out simultaneously in the *post-aging* scheme, we can further utilize aged photonic memory cells for computing with an understanding of the transmissivity change within them. In such wise, the executing lifetime of PTCs can be elongated. However, the phase state of aged PCM wires is hard to analyze. One practical way is to tie PCM wires to one known phase state before being aged, without the need to tediously detect the exact values of aged PCM wires within photonic memories. Off-chip computers can record the number of write operations to each PCM wire within photonic memory cells and stop reconfiguring the phase state of PCM wires anymore when approaching their write endurance budget. The strategy leads to acceptable memory overhead considering the limited number and scale of available on-chip PTCs. Moreover, as a typical bell-shaped distribution with more weights around the mean for neural networks [32], the ability to demonstrate small values in photonic memories is crucial to carry out inference tasks. Therefore, regarding the choice of the fixed phase state for aged PCM wires, setting aged wires to the low-light-transmission c state is preferred to ensure the representability of small transmission values. By doing so, the transmissivity range of aged photonic memory cells will degrade. Assuming x PCM wires are aged within one b-bit photonic memory cell, the maximum implementable transmission factor is degraded to c^x based on the derived model (3). Since the distribution of transmission levels follows an exponential model, a small number of aged PCM wires would lead to serious transmissivity range degradation. For instance, in our adopted design wherein $c = 0.872$, the maximum transmission factor is reduced to 0.58 with only 4 of 15 wires being aged in a 4-bit photonic memory cell. Besides, in higher bit-width photonic memory, the shrink of weight representability is more severe. Half of the PCM wires in a 6-bit memory cell being aged will lead to the implementable value range $<$0.015, while the transmission factor that an aged 4-bit memory cell can provide in the worst case is $\sim$0.128.

B. Proposed Group-wise Row-based Remapping Method

As discussed above, the upper bound t_{max} and the lower bound t_{min} of transmission factors decrease in the aged photonic memory cell compared to the fresh cell. The weight mapping error occurs once the mapped weight value exceeds the supported transmission range. The deviation of demonstrated and desired values is the source of computing error, resulting in model accuracy degradation. Thanks to the intrinsic error-resilience of neural networks, the self-healing characteristic can tolerate minor mapping errors. Thus, we can save model accuracy on defected PTCs by reconfiguring the weight-PTC mapping to control mapping errors.

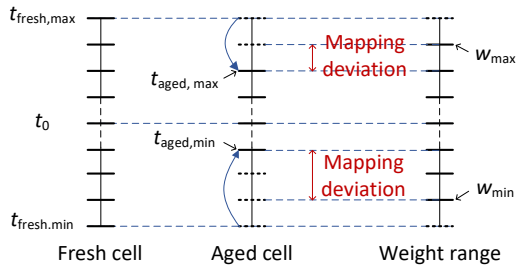


Fig. 7: The mapping deviations when mapping a group of weights onto one aged photonic memory cell. t_0 indicates the transmission level demonstrating value 0.

Consider mapping a group B of weight sub-matrices onto one defected $k \times k$ PTC A, where each weight block also has k rows and k columns. In the proposed write-aware training, our block-matching mechanism can boost weight similarity between weight blocks in B. The weight values at the same position in different blocks are compelled to be close. Those weights should also follow a similar mapping deviation when mapping to the same photonic memory cell. Hence, this unique characteristic provides a chance to jump out of dedicated optimization for each weight block mapping but find a unified mapping solution for a group of similar weight blocks to reduce mapping errors. Inspired by this, we propose a group-wise row-based weight-PTC remapping method, shown as Phase 3 in Fig. 6, where each row of a group of weight blocks with boosted weight similarity are mapped to one chosen row of defected PTCs based on the same optimized mapping relationship.

We first derive the row mapping deviation (RMD) metric to evaluate the mapping error when the m-th row of weight blocks in B are all mapped onto one specific row p of the PTC A. Concretely, the weights in the m-th row and n-th column of different weight blocks in B are correspondingly mapped onto the photonic memory cell in the p-th row and n-th column of PTC A, We can extract the value range of those weights as $[B_{mn,min}, B_{mn,max}]$ and the implementable transmission range at A_{pn} as $[A_{pn,min}, A_{pn,max}]$. By comparing the two ranges as demonstrated in Fig. 7, we can define the mapping deviation metric as the sum of representability gap to indicate whether the two ranges can match well, which is expressed as

$$\gamma^{+}_{mn}|B_{mn,max} - A_{pn,max}| + \gamma^{-}_{mn}|B_{mn,min} - A_{pn,min}|, \quad (21)$$

$$\gamma^{max}_{mn} = \begin{cases} 1, & B_{mn,max} > A_{pn,max} \\ 0, & \text{otherwise} \end{cases}$$

$$\gamma^{min}_{mn} = \begin{cases} 1, & B_{mn,min} < A_{pn,min} \\ 0, & \text{otherwise} \end{cases}$$

wherein γ^{max}_{mn} and γ^{min}_{mn} indicate whether the range of weights exceeds the supported range. Actually, this metric aggressively evaluates the maximum mapping deviation when a group of weights is mapped onto the same cell. Moreover, it should be noted that since the proposed sorting heuristic already sorts weights in ascending or descending order, we can obtain the value range of weights at each position by fetching the first block and the last block, denoted as B_{min} and B_{max}.

Hence, the RMD can be easily computed without extra sorting overhead as

$$RMD(B_m, A_p) = \sum_{i}^{k} \gamma_{mi}^{max} |B_{mi,max} - A_{qi,max}| + \gamma_{mi}^{min} |B_{mi,min} - A_{qi,min}|, \quad (22)$$

where k is the length of the q-th row of PTC A.

Therefore, the remapping scheme can be formulated as the following optimization problem,

$$\pi^* = \text{argmin} \sum_{r}^{k} RMD(B_{\pi(r)}, A_r). \quad (23)$$

where the $\pi(r)$-th row of all weight blocks in the group B is mapped to the r-th row of PTC A. We desire to find the optimized row mapping function π^* for the k rows of a group of weight blocks simultaneously to minimize the mapping deviation.

This remapping problem can be further viewed as a *Minimum weight perfect matching* problem. Specifically, there are k weight rows to be mapped on k rows of PTC, and one row of PTC is linked to exactly one row of weight blocks. Therefore, rows in weight blocks and PTC consist of two disjoint and independent vertex sets of a bipartite graph, where each vertex set has k nodes. The mapping deviation between m-th row of weight blocks and q-th row of PTC, i.e., $RMD(B_m, A_q)$ in Eq. (22), corresponds to the edge cost. Hence, our goal is equivalent to finding an optimized perfect matching π minimizing the total edge cost, i.e., summed mapping deviation. We adopt Hungarian algorithm [33] to solve the problem in polynomial time.

Given that the sizes of two bi-partition of vertex sets are still k, the *Minimum weight perfect matching* problem size is exactly the same with one weight block mapping. But our proposed remapping scheme can obtain the optimized remapping for a group of weight blocks by solving the optimization problem once. Hence, tedious efforts to configure each mapping of weight block are avoided, and efficiency is much improved.

To support our proposed remapping scheme, we need to figure out the correct addresses to fetch input data and weights from memories and write back computing results to memories, introducing extra index cost of the memory address. However, we change the weight-PTC mapping in a group-wise manner instead of dedicating different weight-PTC mapping relationships for each weight block. Thus, our method can tackle the post-aging issue at a small address remapping overhead, which is linear. For example, we divide the groups following Phase 3 in Fig. 6, where the solid black lines indicate the remapping relationship. The needed extra index cost is linear to $O(Pk)$ as the product of the number of groups and the number of PTC rows.

In this section, an efficient remedy to help defected PTCs handle the *post-aging* reliability issue is provided, which prolongs the executing lifetime of the computing engine from a different perspective compared to the proactive *aging-aware* optimization solution.

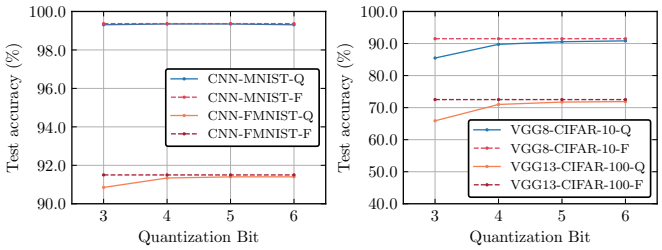


Fig. 8: Quantization evaluation on (a) CNN and (b) VGG models. F means full-precision models. Q means models with both input and weight quantization in the same bit-width.

VI. Experimental Results

A. Experimental Setup

To demonstrate the effectiveness of our proposed three-phase framework `ELight`, extensive experiments are conducted on MNIST [34], FashionMNIST [35], CIFAR-10 and CIFAR-100 [36] datasets. On the first two tasks, a simple CNN model is adopted with a configuration C32K4-C32K4-P5-F64-F10. C32K4 means the convolutional layer has 32 4×4 kernels, P5 is an average pooling layer with output size 5×5, and F64 is a fully-connected (fc) layer with 64 neurons. On CIFAR-10 and CIFAR-100, VGG8 [37], VGG13 [38] and ResNet-18 [16] are adopted. The last three FC layers in VGG8 and VGG13 are replaced with one FC layer to avoid over-fitting. All models are implemented based on a PyTorch-centric ONN library torchonn [39]. Our code will be available at https://github.com/zhuhanqing/ELight. We train the CNN model for 100 epochs. VGG and ResNet models are trained for 200 epochs. The SGD optimizer with a momentum of 0.9 is used during training. Regarding the photonic tensor core size, we assume 16×16 for the small CNN model and 64×64 for VGG and ResNet model. The supported bit-width in photonic memories is set to 3$\sim$6 bit for practical consideration. The *augmented redundant write elimination* (ARWE) strategy is used as the basic optimization technique.

B. Evaluation of the Distribution-Aware Quantization Scheme

Figure 8 shows the accuracy of simple CNN and VGG models under 3- to 6-bit quantization with our augmented base-c quantizer. Our distribution-aware quantization scheme can successfully fit non-linear transmission level distribution, enlarges the solution space with double quantization levels, and achieves high accuracy under low-bit quantization. Under 4- to 6-bit quantization, our proposed method can achieve small accuracy losses on all tasks. Under 3-bit quantization, for relatively complicated tasks, i.e., CIFAR-10 and CIFAR-100, we still get $> 85\%$ and $> 65\%$ accuracy, respectively.

C. Evaluation of Proposed Aging-Aware Optimization Framework

1) Evaluation of write-aware training via block matching: To figure out the effect of write-aware training, we visualize the normalized *# total writes* and accuracy with various degrees of λ on a 5-bit VGG8 model in Fig. 9a.

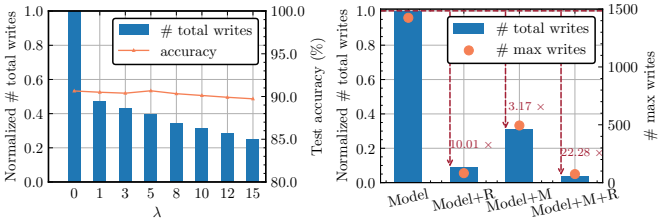


Fig. 9: Evaluation of proposed aging-aware optimization techniques on 5-bit VGG8. (a) Normalized *# total writes* and accuracy comparison with different λ for write-aware training. (b) Comparison between *# total writes* and *# max writes* of the 5^{th} convolutional layer.

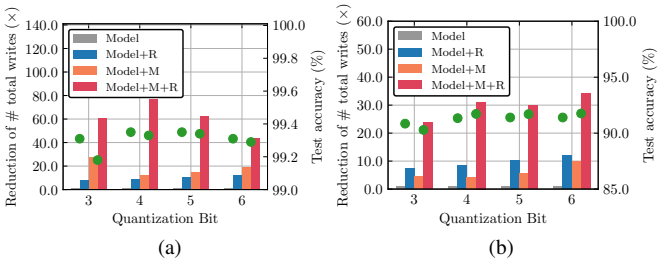


Fig. 10: The reduction of *# total writes* and accuracy of CNN model trained on (a) MNIST and (b) FashionMNIST under different bit-width.

With the increase of λ, *# total writes* decreases as stronger block similarity leads to more redundant write elimination. However, too large λ leads to accuracy decline as weights are penalized for being too identical, making it hard to capture subtle and diverse features. A sweet point exists by trading off the accuracy and *# total writes*, where choosing ($\lambda = 10$) can demonstrate $3.17\times$ reduction in *# total writes* with 0.44% accuracy drop. Interestingly, sometimes a proper λ leads to better accuracy as a regularization mechanism.

2) Evaluation of post-training optimization via fine-grained reordering: Figure 9b compares *# total writes* of a 5-bit VGG8 with/without the write-aware training (M) and fine-grained reordering (R). The *# max writes* of the 5-th convolutional layer is also shown. The results provide three insights: (1) The fine-grained reordering can solely reduce *# total writes* a lot with a $10.01\times$ reduction without the help of extra write-aware training efforts to orchestrate the weights similarity. (2) The fine-grained reordering can work more effectively based on the boosted block similarity provided by the write-aware training, achieving the best of reduction on *# total writes* by **22.28**×. The results justify our efforts on the proposed joint aging-aware optimization framework. (3) Our proposed fine-grained reordering eliminates redundant writes at most with minimized *# max writes*, which re-writes only a small number of PCM wires. Moreover, the proposed method put a upper-bound on *# max writes* as the number of transmission levels, i.e., $2^{b+1}-1$.

3) Evaluation on the synergistic optimization framework: To further testify the effectiveness of the joint optimization techniques in the optimization framework, we evaluate different models under different bit-width quantization. For the choice of λ, we choose a sweet point to guarantee the accuracy within a $\sim 1\%$ drop. Figure 10 demonstrates the accuracy and the *# total writes* reduction of the simple CNN model trained on MNIST and FashionMNIST under 3- to 6-bit quantization, in which *# total writes* is significantly reduced with negligible accuracy drop. Furthermore, Table II and Table III evaluate the effectiveness of our proposed techniques on VGG8, VGG13 and ResNet-18 trained on CIFAR-10 and CIFAR-100. By combining the proposed write-aware training and post-training optimization techniques, the largest reduction on *# total writes* and *# max writes* is achieved, where $>$**20**$\times$ on *# total writes* is observed with less than 1% accuracy degradation. Hence, our joint optimization framework can work orthogonally and successfully mitigate the aging issue by largely reducing write operations.

We also have a more detailed comparison of the reduction of *# total writes* and *# max writes* for each layer of 5-bit VGG13, as shown in Fig. 11. By combining proposed aging-aware optimization techniques, for each layer, *# total writes* can be reduced by over $7\times$ and *# max writes* can be constrained to the smallest number. A large reduction of *# total writes* is obtained on the convolutional layers with larger input and output channels. For those layers, larger weight redundancy exists with more channels such that the weight block similarity can be better boosted.

4) Evaluation of power saving: We further verify the energy efficiency brought by the above optimization methods. We trace the detailed energy cost of writing weight block data onto PTCs during the inference process. The a-c and c-a transition programming pulse profiles are shown in Table IV. Assuming the resistance of heaters is consistent during heating, the ratio of write energy cost between a-c and c-a transition is $40:9$. As our propose optimization techniques largely optimize *# total writes*, the energy cost of deploying VGG8, VGG13 and ResNet-18 is generally reduced by over **25**$\times$ under different bit-widths, as shown in Table II and Table III. The results prove the ability of our aging-aware optimization framework to effectively saves dynamic programming energy cost.

D. Evaluation of Proposed Post-aging Tolerance Scheme

We further evaluate the effectiveness of our proposed group-wise row-based remapping method to enable post-aging tolerance against aged PCM wires. In order to emulate the post-aging status of aged PTCs, we randomly set the number of aged PCM wires in aged memory cells from 1 to 2^b-1. All statistics are collected by multiple runs with different random seeds to ensure thorough and unbiased evaluations. Figure 12 illustrates the inference accuracy of VGG8, VGG13, ResNet-18 under the different ratios of aged memory cells in both positive and negative PTCs.

Through globally reordering the rows for a group of weights blocks, on models trained with/without write-aware training, our method can successfully tolerate aged wires under moderate ratios of aged memory cells in an efficient way, without dedicated reordering for each block or retraining. Under 4- and 5-bit, when the ratio of aged memory cells reaches 0.2, it still achieves $<10\%$ accuracy degradation on VGG8, VGG13, and ResNet-18 models. Though the implementable

TABLE II: Performance of ELight on VGG networks on CIFAR-10 and CIFAR-100 dataset. AC: accuracy change, R: fine-grained reordering. The *# max writes* of one largest convolutional layer (conv5 layer for VGG8 and conv8 layer for VGG13) is shown here.

Network	Dataset	Bitwidth	λ	Acc(%)/AC	*# total writes* ↓ (×)		Energy cost ↓ (×)		*# max writes*	
					-	+R	-	+R	-	+R
VGG8	CIFAR-10	3	0	86.71	1	6.52	1	9.27	128	15
			8	86.02/-0.69	22.12	**46.11**	6.63	**69.29**	14	**7**
		4	0	89.75	1	7.84	1	11.31	401	36
			10	89.94/+0.19	3.83	**24.45**	3.92	**35.48**	95	**19**
		5	0	90.56	1	10.01	1	14.35	1425	82
			10	90.12/-0.44	3.17	**22.28**	3.20	**31.17**	494	**74**
		6	0	90.83	1	12.31	1	16.89	4464	180
			5	89.88/-0.95	6.82	**26.35**	7.15	**32.48**	1560	**146**
VGG13	CIFAR-100	4	0	70.99	1	9.66	1	13.84	542	39
			10	70.44/-0.55	3.54	**29.25**	3.57	**42.02**	173	**33**
		5	0	71.73	1	12.06	1	17.29	1771	84
			3	71.95/+0.22	2.19	**21.93**	2.21	**31.41**	921	**55**
		6	0	71.88	1	14.37	1	17.62	4926	182
			3	70.97/-0.91	3.11	**22.65**	3.19	**29.85**	3577	**156**

TABLE III: Performance of ELight on ResNet-18 [16] networks on CIFAR-10 and CIFAR-100 dataset. AC: accuracy change, R: fine-grained reordering. The *# max writes* of the largest (last) convolutional layer is shown here.

Network	Dataset	Bitwidth	λ	Acc(%)/AC	*# total writes* ↓ (×)		Energy cost ↓ (×)		*# max writes*	
					-	+R	-	+R	-	+R
ResNet-18	CIFAR-10	4	0	92.75	1	8.16	1	11.59	596	43
			10	92.94/+0.19	13.30	**65.72**	13.72	**90.33**	53	**30**
		5	0	92.50	1	9.83	1	13.80	1742	86
			10	91.92/-0.58	5.44	**39.40**	5.50	**51.83**	564	**64**
		6	0	94.07	1	11.27	1	15.13	4989	186
			5	92.98/-1.09	15.51	**38.42**	16.24	**41.15**	730	**140**
ResNet-18	CIFAR-100	4	0	71.32	1	8.58	1	12.27	608	40
			10	71.01/-0.31	4.86	**37.59**	4.98	**53.11**	106	**19**
		5	0	73.11	1	10.06	1	14.15	1846	89
			3	72.18/-0.93	2.82	**25.41**	2.84	**35.05**	687	**53**
		6	0	72.37	1	11.41	1	15.35	5008	183
			2	71.54/-0.64	2.67	**21.16**	2.71	**27.27**	3318	**145**

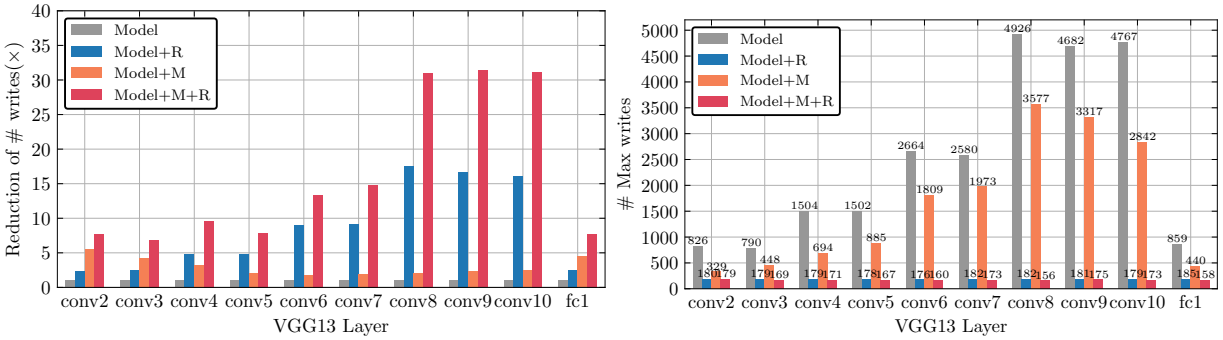


Fig. 11: The comparison of *# writes* and *# max writes* of each convolutional (conv) and fully-connected (fc) layer for 5-bit VGG13. The first convolution layer is not shown here as its implementation needs no PTC reuse.

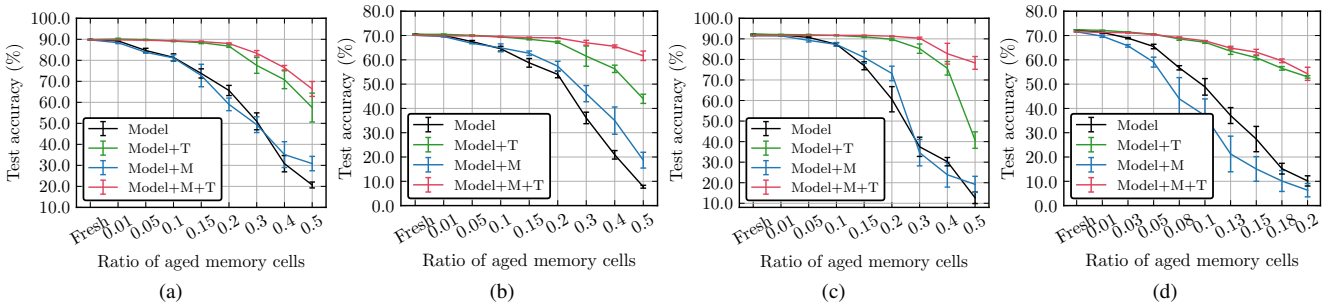


Fig. 12: The accuracy comparison under different ratios of aged memory cells. T means adopting the proposed tolerating scheme. M means adopting the write-aware training. (a) 5-bit VGG8 on CIFAR-10. (b) 4-bit VGG13 on CIFAR-100. (c) 5-bit ResNet-18 on CIFAR-10. (d) 6-bit ResNet-18 on CIFAR-100. Error bars represent the $\pm 1 \cdot \sigma$ variance for multiple runs.

TABLE IV: Pulse profiles for $a \rightarrow c$ and $c \rightarrow a$ transition [13].

	Pulse period(μs)	Pulse voltage(V)	# Pulse
$a \rightarrow c$	1	5	20
$c \rightarrow a$	0.5	15	1

transmission range degradation is much severe in high bit-width as discussed in Section V-A, under 6-bit, our method can recover the accuracy with $<10\%$ drop when the ratio of aged memory cells doesn't exceed 0.15.

Besides, assisted by our proposed write-aware training, our proposed method demonstrates the best tolerance. Especially, in Fig. 12c, when half of the PCM memory cells are aged, the proposed tolerance scheme can raise the accuracy of the 5-bit ResNet-18 trained with write-aware training from $\sim 20\%$ to $\sim 80\%$. In contrast, the accuracy on the normally trained 5-bit ResNet-18 is only recovered to $\sim 40\%$. The superiority of post-aging tolerance on models trained with write-aware training is attributed to the constrained value range of weights induced by the block-matching mechanism. It enables easier adaption of a sequence of weights to aged PCM memory cells with a downgraded transmission range.

VII. Discussion on Other NVMs

The emerging in-memory computing paradigm has attracted widespread attention in the application of neural network acceleration, while limited write endurance lies ahead as a critical challenge in non-volatile memories (NVM) technologies. Our methods can not only be applied in our unique photonic case. Still, they can be applied in other NVM techniques to reduce the number of write operations if the massive reuse of processing elements exists during inference execution, especially with the increasing size and complexity of modern NN models. Here, we briefly discuss the applicability of our methods in ReRAM-based in-memory computing.

ReRAM-based chips use multiple single-level cells (SLCs) or multi-level cells (MLCs) to demonstrate high bit-width synaptic weights. The binary coding format is used instead of the unary coding format in our photonic memory case. Considering a k-bit ReRAM cell, $N_q = \lceil \frac{n}{k} \rceil$ cells are needed to represent a n-bit weight, where $w_q = \sum_{i=1}^{N_q} c_i 2^{(i-1)k}$, where c_i represents the value stored in the i-th ReRAM cell. Since k usually is smaller than n, e.g., 2-bit and 3-bit MLC ReRAM cells are typically used, we can still apply redundant write elimination strategy by reusing identical parts. However, in binary coding format, the difference between two values cannot represent the real programming cost. For instance, with 2-bit MLCs, two cells are needed to store 4-bit weight. The difference in resistance level between 0100 and 0011 is one, but we need to program two cells. We need to modify our methods by replacing the Eq. (10) with the programming cost based on the binary coding format. Then our write-aware training method can then be applied to orchestrate the block-wise weight similarities, followed by our reordering heuristic. Ideally, only the cells for the least significant bits need to be frequently reprogrammed with boosted weight similarity, while heavy write imbalance exists. Other techniques like swapping need to be integrated to handle the unbalanced write distribution along with ReRAM cells with different significance. We are interested in solving the binary encoding case in our future work. Recently, unary coding of synaptic weights in ReRAM has been investigated to stand out with better tolerance to the resistance variations [40]. In this case, our aging-aware optimization methods can be purely transferred.

As for the post-aging tolerance scheme, we conduct weight-PTC remapping based on the collected error information. We can easily apply this technique in other NVMs if we can collect the error distribution.

VIII. Conclusion

In this work, we propose a holistic solution `ELight` to enable efficient and robust photonic in-memory neurocomputing with a prolonging lifetime. We first model the non-linear transmission distribution of PCM-based photonic memories and propose a dedicated distribution-aware quantization scheme to reduce weight encoding errors and improve ONN accuracy on the low-precision PTC. To avoid the aging issue, a write-aware training method and a post-training optimization method work jointly to trim down redundant PCM writes. As a proactive aging-aware optimization framework, our proposed method significantly reduces the number of total write operations and the number of write operations for the most over-utilized memory cell. To further tackle the post-aging reliability issue, a group-wise row-based remapping methodology is introduced to recover the accuracy drop against aged PCM wires by re-configuring weight-PTC row mapping in an efficient way. Experimental results demonstrate that the proposed solution can reduce the number of write operations and energy costs by $>20\times$ and show superior resilience against aged PCM wires. Our `ELight` can push photonic in-memory neurocomputing towards practical, long-life, and robust application in efficient inference acceleration.

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 by NSF under Award #1718570.

References

[1] Y. Shen, N. C. Harris, S. Skirlo *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, 2017.
[2] Z. Zhao, D. Liu, M. Li *et al.*, "Hardware-software co-design of slimmed optical neural networks," in *Proc. ASPDAC*, 2019.
[3] J. Gu, Z. Zhao, C. Feng *et al.*, "Towards area-efficient optical neural networks: an FFT-based architecture," in *Proc. ASPDAC*, 2020.
[4] W. Liu, W. Liu, Y. Ye, Q. Lou, Y. Xie, and L. Jiang, "Holylight: A nanophotonic accelerator for deep learning in data centers," in *Proc. DATE*, 2019.
[5] Q. Cheng, J. Kwon, M. Glick, M. Bahadori, L. P. Carloni, and K. Bergman, "Silicon Photonics Codesign for Deep Learning," *Proceedings of the IEEE*, 2020.
[6] F. Zokaee, Q. Lou, N. Youngblood *et al.*, "LightBulb: A Photonic-Nonvolatile-Memory-based Accelerator for Binarized Convolutional Neural Networks," in *Proc. DATE*, 2020.
[7] G. Wetzstein, A. Ozcan, S. Gigan, S. Fan, D. Englund, M. Soljačić, C. Denz, , D. A. B. Miller, and D. Psaltis, "Inference in artificial intelligence with deep optics and photonics," *Nature*, 2020.
[8] J. Gu, Z. Zhao, C. Feng, Z. Ying, R. T. Chen, and D. Z. Pan, "O2NN: Optical Neural Networks with Differential Detection-Enabled Optical Operands," in *Proc. DATE*, 2021.

[9] J. Gu, C. Feng, Z. Zhao, Z. Ying, M. Liu, R. T. Chen, and D. Z. Pan, "SqueezeLight: Towards Scalable Optical Neural Networks with Multi-Operand Ring Resonators," in *Proc. DATE*, 2021.
[10] B. J. Shastri, A. N. Tait, T. F. de Lima, W. H. P. Pernice, H. Bhaskaran, C. D. Wright, and P. R. Prucnal, "Photonics for artificial intelligence and neuromorphic computing," *Nature Photonics*, 2021.
[11] J. Gu, Z. Zhao, C. Feng *et al.*, "Towards Hardware-Efficient Optical Neural Networks: Beyond FFT Architecture via Joint Learnability," *IEEE TCAD*, 2020.
[12] C. Feng, J. Gu, H. Zhu, Z. Ying, Z. Zhao, D. Z. Pan, and R. T. Chen, "Silicon photonic subspace neural chip for hardware-efficient deep learning," *arXiv preprint arXiv:2111.06705*, 2021.
[13] M. Miscuglio and V. J. Sorger, "Photonic tensor cores for machine learning," *Applied Physics Review*, 2020.
[14] J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja *et al.*, "Parallel convolution processing using an integrated photonic tensor core," *Nature*, 2021.
[15] C. Wu, H. Yu, S. Lee, R. Peng, I. Takeuchi, and M. Li, "Programmable phase-change metasurfaces on waveguides for multimode photonic convolutional neural network," *Nat Comm*, 20121.
[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
[17] Y. Zhang, C. Ríos, M. Y. Shalaginov, M. Li, A. Majumdar, T. Gu, and J. Hu, "Myths and truths about optical phase change materials: A perspective," *Applied Physics Letters*, 2021.
[18] H. Zhu, J. Gu, C. Feng, M. Liu *et al.*, "ELight: Enabling Efficient Photonic In-Memory Neurocomputing with Life Enhancement," in *Proc. ASPDAC*, 2022.
[19] J.-S. Moon, H.-C. Seo, K. K. Son, E. Yalon, K. Lee, E. Flores, G. Candia, and E. Pop, "Reconfigurable infrared spectral imaging with phase change materials," in *Micro-and Nanotechnology Sensors, Systems, and Applications XI*, 2019.
[20] Y. Cai, Y. Lin, L. Xia, X. Chen, S. Han, Y. Wang, and H. Yang, "Long live time: improving lifetime for training-in-memory engines by structured gradient sparsification," in *Proc. DAC*, 2018.
[21] M. Liu, L. Xia, Y. Wang, and K. Chakrabarty, "Fault tolerance in neuromorphic computing systems," in *Proc. ASPDAC*, 2019.
[22] W. Wen, Y. Zhang, and J. Yang, "Renew: Enhancing lifetime for reram crossbar based neural network accelerators," in *Proc. ICCD*, 2019.
[23] F. Meng, Y. Xue, and C. Yang, "Power-and endurance-aware neural network training in nvm-based platforms," *IEEE TCAD*, 2018.
[24] B.-D. Yang, J.-E. Lee, J.-S. Kim, J. Cho, S.-Y. Lee, and B.-G. Yu, "A low power phase-change random access memory using a data-comparison write scheme," in *Proc. ISCAS*, 2007.
[25] S. Cho and H. Lee, "Flip-n-write: A simple deterministic technique to improve pram write performance, energy and endurance," in *Proc. MICRO*, 2009.
[26] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger, "Architecting phase change memory as a scalable dram alternative," in *Proc. ISCA*, 2009.
[27] M. Zhao, Y. Xue, C. Yang, and C. J. Xue, "Minimizing mlc pcm write energy for free through profiling-based state remapping," in *Proc. ASPDAC*, 2015.
[28] S. Zhou, Z. Ni, X. Zhou *et al.*, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2016.
[29] G. Hinton, "Neural networks for machine learning," *Coursera Video Lecture*, 2012.
[30] A. Fan, P. Stock, B. Graham, E. Grave, R. Gribonval, H. Jegou, and A. Joulin, "Training with quantization noise for extreme model compression," in *Proc. ICML*, 2021.
[31] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. Multimedia*, 2014.
[32] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.
[33] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, 1955.
[34] Y. LeCun, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/, 1998.
[35] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," *Arxiv*, 2017.
[36] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
[37] L. Deng, P. Jiao *et al.*, "Gxnor-net: Training deep neural networks with ternary weights and activations without full-precision memory under a unified discretization framework," *Neural Networks*, 2018.
[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[39] J. Gu, H. Zhu, C. Feng, Z. Jiang, R. T. Chen, and D. Z. Pan, "L2ight: Enabling On-Chip Learning for Optical Neural Networks via Efficient in-situ Subspace Optimization," in *Proc. NeurIPS*, 2021.
[40] Y. Sun, C. Ma, Z. Li, Y. Zhao, J. Jiang, W. Qian, R. Yang, Z. He, and L. Jiang, "Unary coding and variation-aware optimal mapping scheme for reliable reram-based neuromorphic computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

Hanqing Zhu (S'20) received the B.E. degree in Microelectronic Science and Engineering from Shanghai Jiao Tong University, Shanghai, China in 2020. He is currently pursuing his Ph.D. degree in the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, USA. His current research interests include high-performance AI computing with emerging technologies, machine learning, and its application to VLSI physical design automation.

Jiaqi Gu (S'19) received the B.E. degree in Microelectronic Science and Engineering from Fudan University, Shanghai, China in 2018. He is currently a post-graduate student studying for his Ph.D. degree in the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, USA. His current research interests include machine learning, efficient algorithm and architecture design for high-performance AI, next-generation AI computing with emerging technology, and GPU acceleration for VLSI design automation. He has received the Best Paper Award at IEEE TCAD 2021, the Best Paper Award at ASP-DAC 2020, the Best Paper Finalist at DAC 2020, the Best Poster Award at NSF Workshop on Machine Learning Hardware (2020), the ACM/SIGDA Student Research Competition First Place (2020), and the ACM Student Research Competition Grand Finals First Place (2021).

Chenghao Feng received the B.S. degree in physics from Nanjing University, Nanjing, China, in 2018. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA. His research interests include silicon photonics devices and system design for optical computing and interconnect in integrated photonics.

Mingjie Liu received his B.S degree from Peking University and M.S. degree from the University of Michigan, Ann Arbor in 2016 and 2018, respectively. He is currently pursuing his Ph.D. degree in Electrical and Computer Engineering at The University of Texas at Austin. His current research interests include applied machine learning for design automation, and physical design automation for analog and mixed-signal integrated circuits.

Zixuan Jiang received the B.E. degree in electronic information engineering from Zhejiang University, Hangzhou, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, the University of Texas at Austin, Austin, TX, USA. His current research interests involve efficient machine learning from the perspectives of software and hardware. He works on machine learning algorithms, systems, hardware synergistically. He received 2021 TCAD Best Paper Award.

Ray T. Chen (M'91–SM'98–F'04) received the B.S. degree in physics from the National Tsing Hua University, Hsinchu, Taiwan, in 1980, and the M.S. degree in physics and Ph.D. degree in electrical engineering from the University of California, in 1983 and 1988, respectively. He is the Keys and Joan Curry/Cullen Trust Endowed Chair with the University of Texas at Austin (UT Austin), Austin, TX, USA. He is the Director of the Nanophotonics and Optical Interconnects Research Lab, Microelectronics Research Center. He is also the Director of the AFOSR MURI-Center for Silicon Nanomembrane involving faculty from Stanford, UIUC, Rutgers, and UT Austin. In 1992, he joined the UT Austin to start the optical interconnect research program. From 1988 to 1992, he worked as a Research Scientist, Manager, and Director of the Department of Electro-Optic Engineering, Physical Optics Corporation, Torrance, CA, USA.

From 2000 to 2001, he served as the CTO, founder, and Chairman of the Board of Radiant Research, Inc., where he raised 18 million dollars A-Round funding to commercialize polymer-based photonic devices involving more than 20 patents, which were acquired by Finisar in 2002, a publicly traded company in the Silicon Valley (NASDAQ:FNSR). He served as the CTO, Founder, and Chairman of the Board of Radiant Research, Inc. from 2000 to 2001, where he raised 18 million dollars A-Round funding to commercialize polymer-based photonic devices involving over twenty patents, which were acquired by Finisar in 2002, a publicly traded company in the Silicon Valley (NASDAQ:FNSR). He also serves as the founder and Chairman of the Board of Omega Optics Inc. since its initiation in 2001. Omega Optics has received over five million dollars in research funding. His research work has been awarded over 145 research grants and contracts from such sponsors as Army, Navy, Air Force, DARPA, MDA, NSA, NSF, DOE, EPA, NIST, NIH, NASA, the State of Texas, and private industry. The research topics are focused on four main subjects: (1) Nano-photonic passive and active devices for bio- and EM-wave sensing and interconnect applications, (2) Thin film guided-wave optical interconnection and packaging for 2D and 3D laser beam routing and steering, (3) True time delay (TTD) wide band phased array antenna (PAA), and (4). 3D printed micro-electronics and photonics. Experiences garnered through these programs are pivotal elements for his research and further commercialization.

His group at UT Austin has reported its research findings in more than 970 publications, including over 100 invited papers and 74 patents. He has chaired or been a program-committee member for more than 130 domestic and international conferences organized by IEEE, SPIE (The International Society of Optical Engineering), OSA, and PSC. He has served as an editor, co-editor or coauthor for over twenty books. Chen has also served as a consultant for various federal agencies and private companies and delivered numerous invited talks to professional societies. Chen is a Fellow of IEEE, OSA, and SPIE. He was the recipient of the 1987 UC Regent's Dissertation Fellowship and the 1999 UT Engineering Foundation Faculty Award, for his contributions in research, teaching and services. He received the honorary citizenship award in 2003 from the Austin city council for his contribution in community service. He was also the recipient of the 2008 IEEE Teaching Award, and the 2010 IEEE HKN Loudest Professor Award. 2013 NASA Certified Technical Achievement Award for contribution on moon surveillance conformable phased array antenna. During his undergraduate years at the National Tsing Hua University he led the 1979 university debate team to the Championship of the Taiwan College-Cup Debate Contest.

David Z. Pan (S'97–M'00–SM'06–F'14) received his B.S. degree from Peking University in 1992, and his M.S. and Ph.D. degrees from University of California, Los Angeles (UCLA), in 1998 and 2000. From 2000 to 2003, he was a Research Staff Member with IBM T. J. Watson Research Center. He is currently a Full Professor and holder of the Silicon Laboratories Endowed Chair in Electrical Engineering at The University of Texas at Austin. His research interests include electronic design automation, design for manufacturing, machine learning and hardware acceleration, design/CAD for analog/mixed signal designs and emerging technologies. He has published over 430 journal articles and refereed conference papers, and is the holder of 8 U.S. patents. He has graduated 41 PhD/postdocs who are holding key academic and industry positions.

He has served as a Senior Associate Editor for ACM Transactions on Design Automation of Electronic Systems (TODAES), an Associate Editor for IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems (TCAD), IEEE Transactions on Very Large Scale Integration Systems (TVLSI), IEEE Transactions on Circuits and Systems PART I (TCAS-I), IEEE Transactions on Circuits and Systems PART II (TCASII), IEEE Design & Test, Science China Information Sciences, Journal of Computer Science and Technology, IEEE CAS Society Newsletter, etc. He has served in the Executive and Program Committees of many major conferences. He is the ISPD 2008 General Chair, DAC 2014 Tutorial Chair, ASP-DAC 2017 Program Chair, ICCAD 2018 Program Chair, and ICCAD 2019 General Chair, and DAC 2022 Panel Chair.

He has received a number of prestigious awards for his research contributions, including the SRC Technical Excellence Award in 2013, DAC Top 10 Author in Fifth Decade, DAC Prolific Author Award, ASP-DAC Frequently Cited Author Award, ASP-DAC Prolific Author Award, 20 Best Paper Awards at premier venues (TCAD 2021, ISPD 2020, ASP-DAC 2020, DAC 2019, GLSVLSI 2018, VLSI Integration 2018, HOST 2017, SPIE 2016, ISPD 2014, ICCAD 2013, ASP-DAC 2012, ISPD 2011, IBM Research 2010 Pat Goldberg Memorial Best Paper Award, ASP-DAC 2010, DATE 2009, ICICDT 2009, SRC Techcon in 1998, 2007, 2012 and 2015) and 18 additional Best Paper Award finalists, Communications of the ACM Research Highlights (2014), ACM/SIGDA Outstanding New Faculty Award (2005), NSF CAREER Award (2007), SRC Inventor Recognition Award three times, IBM Faculty Award four times, UCLA Engineering Distinguished Young Alumnus Award (2009), UT Austin RAISE Faculty Excellence Award (2014), Cadence Academic Collaboration Award (2019), and many international CAD contest awards, among others. His students have also won many awards, including the First Place of ACM Student Research Competition Grand Finals in 2018, ACM/SIGDA Student Research Competition Gold Medal (twice), ACM Outstanding PhD Dissertation in EDA (twice), EDAA Outstanding Dissertation Award (thrice), and so on. He is a Fellow of ACM, IEEE and SPIE.