# MaxTracker: Continuously Tracking the Maximum Computation Progress for Energy Harvesting ReRAM-based CNN Accelerators

KENI QIU, Capital Normal University, China
NICHOLAS JAO, Pennsylvania State University, USA
KUNYU ZHOU, Capital Normal University, China
YONGPAN LIU, Tsinghua University, China
JACK SAMPSON, MAHMUT TAYLAN KANDEMIR, and
VIJAYKRISHNAN NARAYANAN, Pennsylvania State University, USA

There is an ongoing trend to increasingly offload inference tasks, such as CNNs, to edge devices in many IoT scenarios. As energy harvesting is an attractive IoT power source, recent ReRAM-based CNN accelerators have been designed for operation on harvested energy. When addressing the instability problems of harvested energy, prior optimization techniques often assume that the load is fixed, overlooking the close interactions among input power, computational load, and circuit efficiency, or adapt the dynamic load to match the just-in-time incoming power under a simple harvesting architecture with no intermediate energy storage.

Targeting a more efficient harvesting architecture equipped with both energy storage and energy delivery modules, this paper is the first effort to target whole system, end-to-end efficiency for an energy harvesting ReRAM-based accelerator. First, we model the relationships among ReRAM load power, DC-DC converter efficiency, and power failure overhead. Then, a maximum computation progress tracking scheme (*MaxTracker*) is proposed to achieve a joint optimization of the whole system by tuning the load power of the ReRAM-based accelerator. Specifically, *MaxTracker* accommodates both continuous and intermittent computing schemes and provides dynamic ReRAM load according to harvesting scenarios.

We evaluate *MaxTracker* over four input power scenarios, and the experimental results show average speedups of 38.4%/40.3% (up to 51.3%/84.4%), over a full activation scheme (with energy storage) and order-of-magnitude speedups over the recently proposed (energy storage-less) *ResiRCA* technique. Furthermore, we also explore *MaxTracker* in combination with the *Capybara* reconfigurable capacitor approach to offer more flexible tuners and thus further boost the system performance.



CCS Concepts: • **Computer systems organization → Embedded systems**; *Reconfigurable computing*; Firmware;

This work was supported in part by NSFC grant #61872251, Beijing Advanced Innovation Center for Imaging Technology, and NSF grants #1629915, #1763681, #1822923 and #2008365. This article appears as part of the ESWEEK-TECS special issue and was presented in the International Conference on International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2021.

Authors' addressess: K. Qiu, Capital Normal University, #56, West Third Ring North Road, Beijing, 100048, China; email: qiukn@cnu.edu.cn; N. Jao, J. Sampson, M. T. Kandemir, and V. Narayanan, Pennsylvania State University, University Park, State College, PA, 16802, USA; emails: naj5075@psu.edu, sampson@cse.psu.edu, mtk2@cse.psu.edu, vijay@cse.psu.edu; K. Zhou.Capital Normal University, #56, West Third Ring North Road, Beijing, 100048, China; email: philyu20@163.com; Y. Liu, Tsinghua University, Haidian District, Beijing, 100084, China; email: yp-liu@vip.163.com.



1539-9087/2021/09-ART78 $15.00
https://doi.org/10.1145/3477009

Additional Key Words and Phrases: Energy harvesting, ReRAM crossbar, CNN, DC-DC efficiency, computing schemes, maximum computation progress

**ACM Reference format:**
Keni Qiu, Nicholas Jao, Kunyu Zhou, Yongpan Liu, Jack Sampson, Mahmut Taylan Kandemir, and Vijaykrishnan Narayanan. 2021. MaxTracker: Continuously Tracking the Maximum Computation Progress for Energy Harvesting ReRAM-based CNN Accelerators. *ACM Trans. Embedd. Comput. Syst.* 20, 5s, Article 78 (September 2021), 23 pages.
https://doi.org/10.1145/3477009

## 1 INTRODUCTION

It is increasingly common to see inference tasks, such as convolutional neural networks (CNNs), processed locally on the edge devices that collect the data [7, 44], rather than being offloaded to the cloud. ReRAM crossbar-based accelerators with intrinsically high-efficiency Processing-In-Memory (PIM) capabilities [35] have been proposed to perform the convolutional computations of the CNNs. In many Internet of Things (IoT) scenarios [7, 14, 15], using ambient energy harvesting (e.g., solar, thermal, RF [11, 28, 29, 36, 41]) as a power source is appealing if the underlying hardware platform provides support for intermittent computing [3, 4, 6, 8, 24, 26, 27, 30]. Recent works have examined the intersection of these two trends [16, 37]. For example, the HAWAII runtime system [16], which uses the concept of inference footprinting, was proposed to preserve the Deep Neural Network (DNN) accelerator progress across power cycles without requiring access to the peripheral internal state. Anyhow, the exploration of the design space of energy-harvesting inference accelerators is still in its infancy nowadays.

In this context, the most commonly deployed model is the "Harvest-Store-Use" paradigm [5, 38] wherein a storage capacitor is conservatively filled with sufficient harvested energy to power the sensor node for a design-time specified period of continuous operation prior to the compute phase. A key challenge in such architectures is the efficient delivery of stored energy to the accelerator. Although prior works have extensively studied the efficiency issue [9, 40], they often assume that the electronic load driven by the power delivery components at the sensor node is fixed, *which overlooks potentially important interactions between the load and power delivery/conversion circuit efficiency*. For example, recent work [37] is implemented based on a fixed load of four $32 \times 32$ ReRAMs but with a varying, unstable solar power source. This has two potential drawbacks: 1) the power requirement of activating the ReRAM load is high, and 2) the harvesting circuits may operate at low efficiency most of time due to the fixed load power. On the other hand, although recent work [32] proposes to offer multiple load granularities by decomposing compute operations onto subsets of ReRAM arrays, it adopts a simple "Harvest-Direct Use" architecture, and neither attempts to perform a global end-to-end efficiency optimization, nor considers load granularity optimization in the presence of a non-trivial energy store, as would be present in a "Harvest-Store-Use" or hybrid "Store-Use/Direct Use architecture".

In this work, targeting a "Harvest-Store-Use" architecture, we leverage two key observations to perform end-to-end efficiency optimization: Load at sensor/compute node greatly impacts DC-DC converter efficiency $\eta_{DC-DC}$, and, even with a modest energy store, the granularity of work scheduling can be used to exert substantial control over both sensor/compute load and storage voltage. To incorporate these observations, this paper proposes a novel strategy, *MaxTracker*, that tracks the maximum computation progress through dynamically tuning the load at ReRAM compute core. To this end, *MaxTracker* performs tracking at two layers. First, *MaxTracker* evaluates two schemes for computing with harvested energy (intermittent computing and continuous computing) and selects the compute scheme suitable for specific power harvesting scenario. Second, specific to

each computing scheme, *MaxTracker* always captures the operation status, leading to the highest throughput through tuning the ReRAM activation size. This two-layer strategy achieves a good balance of the trade-offs among the key factors of harvested power, ReRAM computations, DC-DC converter efficiency, and power failure overhead of the whole system. Experimental results across four power sources and four CNN networks show that *MaxTracker*'s joint optimization achieves average speedups of 38.4%/40.3% (up to 51.3%/84.4%) compared to a fixed load baseline without the dynamic tracking technique, and order-of-magnitude speedups over the recent energy storage-less *ResiRCA* technique [32]. Furthermore, we also have explored combining *MaxTracker* with the *Capybara* reconfigurable capacitor approach [5] to offer more flexible tuners to further boost the system performance.

In summary, this work makes the following contributions:

- This is the first work to tune a ReRAM-based convolution accelerator in an energy harvesting scenario to account for the end-to-end energy delivery efficiency. We perform full system throughput and energy efficiency modeling of all key factors of the computation progress, including power failure and power leakage overhead.
- We propose the *MaxTracker* framework, consisting of a design-time tool and an online scheduling mechanism that respectively model and calculate the efficiency-preferred activation configurations corresponding to specific incoming (harvested) power levels and load conditions and manage the dynamic transitions among them. The *MaxTracker* approach is shown to significantly enhance the maximum computation progress over four energy harvesting scenarios by accommodating two computing schemes and performing dynamic ReRAM activation size tuning.
- We build a CNN computing simulator to mimic the operations of CNN inference with various power traces. The circuit parameters of the "Harvest-Store-Use" architecture and the *MaxTracker* strategy are modeled in our simulator.

The rest of this paper is organized as follows. Section 2 introduces the energy harvesting architecture,energy delivery of the key DC-DC converter components and a review of ReRAM load tuning of the most related state-of-art work. Section 3 describes the mechanisms employed by *MaxTracker* to apply two computing schemes across three power harvesting scenarios. Section 4 gives the evaluation results. Section 5 reviews the recent related studies. Finally, we conclude the paper in Section 6.

## 2 BACKGROUND INFORMATION

**Energy Harvesting Architectures:** Energy harvesting architectures can be broadly divided into two categories: "*Harvest-Direct Use*" and "*Harvest-Store-Use*" [38]. In the first architecture, energy is harvested just-in-time for use without conservatively guaranteeing the presence of sufficient energy to complete a task prior to initialization, whereas, in the latter architecture, energy is harvested, stored in a capacitor for future use, and expended only when sufficient energy exists to complete a scheduled task. The advantages of the "*Harvest-Store-Use*" architecture span three aspects. First, it allows the continued operation from stored energy during periods where incoming power alone is insufficient to sustain computation. Second, when there is incoming power, but that power is insufficient to support any useful computation, the energy can be stored until enough is accumulated to complete a meaningful unit of work (with the caveat that the average power income must still exceed the average energy storage leakage during such an accumulation period). Third, by only attempting to operate when energy reserves are within well-defined bounds, the "*Harvest-Store-Use*" architecture can provide a more predictable voltage range during active periods than a "*Harvest-Direct Use*" architecture, which offers a better circuit design target for optimization.

The *Harvest-Store-Use* architecture has several interacting components. Figure 1 shows the architectural block diagram consisting of an energy harvesting (EH) module, power booster with

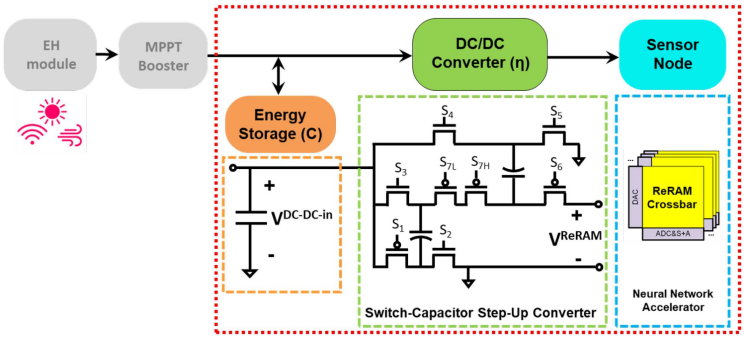


Fig. 1. The "*Harvest-Direct Use*" architecture under study, showcasing circuit topology of the configurable DC-DC converter from [2].

maximum power point tracking (MPPT) booster, energy storage, DC-DC converter, and sensor node. The EH module converts the ambient energy into electrical energy and the MPPT booster maximizes the power extraction of the EH module under varying conditions. For the energy storage component, the voltage across the storage capacitor is proportional to the charge stored. Lastly, the DC-DC converter supplies the proper operating voltage to the sensor node. In the scope of this work, as indicated in the red dash box in Figure 1, we study the interactions between the energy storage, DC-DC converter and the sensor node components to establish a hardware-aware computation scheduling to optimize upon the trade-offs and operating conditions of the different energy harvesting components during run-time.

**DC-DC Energy Delivery:** In an energy harvesting architecture, the most important concern is how to transform the harvested energy into as much forward progress as possible in the sensor node. While the MPPT booster draws the maximum power from the harvester to the energy storage [42], the DC-DC converter's primary function is to regulate a constant voltage output even when input voltage from energy storage and the output current to the sensor node is changing. In practice, non-ideal converters consume stored energy to maintain its voltage regulation function in addition to the energy delivered to the load. Hence, we identify that the DC-DC converter efficiency $\eta_{DC-DC}$ is the key metric in how much available energy at the energy storage is spent on meaningful computation in the sensor node.

From an energy harvesting system-on-chip standpoint, the switched converter is the most suitable voltage regulator for our application due to its low-cost integration on-chip and capability to achieve high-efficiency at low output voltage [17, 34]. There are two major sources of power losses in a switch-capacitor (SC) DC-DC converter: conduction loss and switching loss [18]. Conduction loss arises from the power dissipated across parasitic resistances and dominates when the power supplied to the load is small. Switching loss comes from the dynamic power consumed by the converter at high frequencies and the main source of loss when the power delivered to the sensor node is large.

In summary, the efficiency of DC-DC converters changes with the computational intensity of sensor node (output power) and the available energy in the storage capacitor (input voltage). In the context of deploying a ReRAM accelerator at the sensor node, it is critical to take into account the hardware-level relationships to maximize the forward progress when the available energy is limited.

**Review: Resilient Computation on ReRAM Accelerators:** ReRAM crossbar-based accelerator enables a new paradigm to offload neural network (NN) tasks on IoT nodes under a bandwidth-constrained circumstance [37]. ReRAM crossbars can use the resistance values stored in the cells

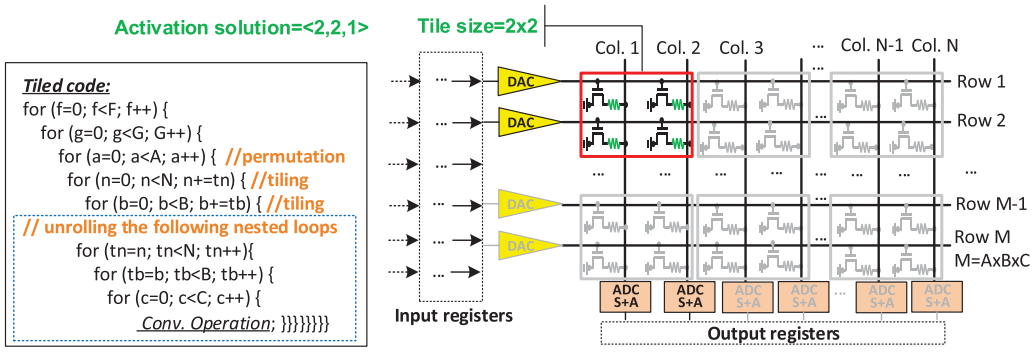


Fig. 2. Example of ReRAM tiling [32].

and the NN weights to execute multiply-accumulate (MAC) operations with high parallelism and low power.

Matching execution to available energy helps avoid power failures. For sensor nodes using a ReRAM-based accelerator, recent work [32] proposes partitioning a ReRAM crossbar into smaller tiles suitable for executing a loop-tiled decomposition of CNN operations as sets of partial ReRAM activations. As an example, Figure 2 shows that a large $m \times n$ ReRAM crossbar can be partitioned into $2 \times 2$-size tiles, and then can be activated one tile at a time. Eventually, all the MAC operations on the entire ReRAM can be completed after merging the partial results. ReRAM duplication by a factor of $d$ can also be integrated with loop tiling to achieve extensive resiliency of computing. With ReRAM duplication, activation tiles from multiple ReRAMs can be implemented in parallel. This activation solution can be represented as $\langle m, n, d \rangle$.

The tiled computing scheme grants resiliency to the ReRAM accelerator so that the ReRAM load can be finely tuned to fit the incoming power, thereby avoiding some power emergencies altogether in a *Harvest-Direct Use* architecture. The low-power ReRAM architecture, ReRAM tiling and duplication along with the convolution layer scheduling algorithm are all combined into the *ResiRCA* technique [32]. Since the CNN convolutions can be decomposed in a variety of ways in terms of both the size of partial activation and the degree of duplication, the ResiRCA technique provides a load tuning knob that *can cause ReRAM load to vary across a wide range.*

However, the *ResiRCA* technique targets a *"Harvest-Direct Use"* architecture, unaware of energy store and energy delivery components as well as their complex interaction. In this work, we target the commonly deployed *"Harvest-Store-Use"* architecture for energy harvesting systems. Although we are also motivated to exploit the ReRAM load tuning knob to achieve high performance, the load tuning strategy is very different from *ResiRCA* because our tuning strategy should be able to build the most efficient interaction among the key components of energy storage, DC-DC converter and the sensor node, and thus achieving high performance at the sensor node.

## 3 MAXIMUM COMPUTATION PROGRESS TRACKING

### 3.1 Architecture and Software Overview

As discussed above, *the load plays a significant role in overall system efficiency*. By exploiting the decomposability of parallel convolution operations on ReRAMs to tune the load power, the best operation state of the whole system can be ensured by a dedicated *"load tuner"*.

In this work, we introduce *MaxTracker*, a tracking scheme that captures the income-load-progress dynamics. Based on exploring different computing schemes (Section 3.2) and the throughput and energy efficiency models of the whole system (Section 3.3), *MaxTracker* adopts two

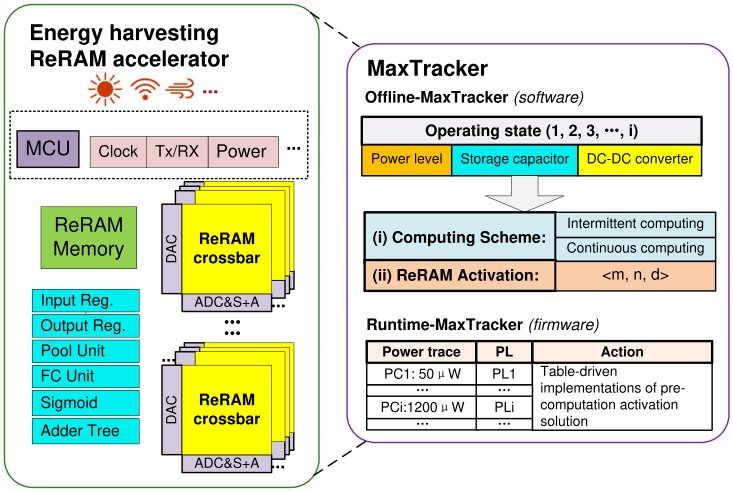


Fig. 3. Overview of the ReRAM sensor architecture and *MaxTracker* framework.

different execution schemes for three energy harvesting cases, to capture the best operating strategy (Section 3.4), so as to achieve the highest throughput, that is, the maximum computing progress.

MaxTracker primarily consists of two main components: 1) *Offline-MaxTracker* determines the computing scheme as well as ReRAM activation solution for each convolution layer under different operating environments taking into account of the interaction among energy store, DC-DC converter and ReRAM load, and 2) *Runtime-MaxTracker* uses the *Offline* ReRAM activation solutions to guide CNN execution as the operating state varies at run-time. Figure 3 illustrates the ReRAM sensor node architecture and the software and hardware overview of our proposed *MaxTracker*.

### 3.2 Computing Progress Modeling

As a bridge connecting the energy store and energy consumer, DC-DC converter efficiency is sensitive to both the input voltage and the load. Therefore, even with a fixed load, the input voltage $V_{DC-DC-in}$ keeps changing as the capacitor charge $Q^{Cap}$ is continuously consumed. This, in turn, impacts the energy conversion efficiency $\eta_{DC-DC}$. *MaxTracker* can capture these changes and then dynamically tune the load to achieve the optimal operating status. This calibration process is directed by the evaluation metric of throughput that also corresponds to the computing progress of the ReRAM accelerator. Thus, the primary work is to model the relationship between the key metrics of throughput and energy efficiency as they interact with the key factors of the circuits and ReRAM load with different computing schemes.

To provide clarity in our discussion of the ReRAM computing process, we first define the following terms, which are also labeled in Figure 4.

**Working Cycle (WC)**: A working cycle denotes a purely charging period, followed by a consecutive working period.

The purely charging period (with no operations) and the consecutive working period are denoted as $T_{idle}$ and $T_{op}$, respectively. Charging can happen all the time while discharging only happens during the $T_{op}$ period. At the end of each working cycle, it is necessary to back up the status data into the nonvolatile ReRAM memory. Correspondingly, at the beginning of each working cycle, the stored status data should be written back to the volatile registers or memory. Each power cycle usually holds more than one WC.

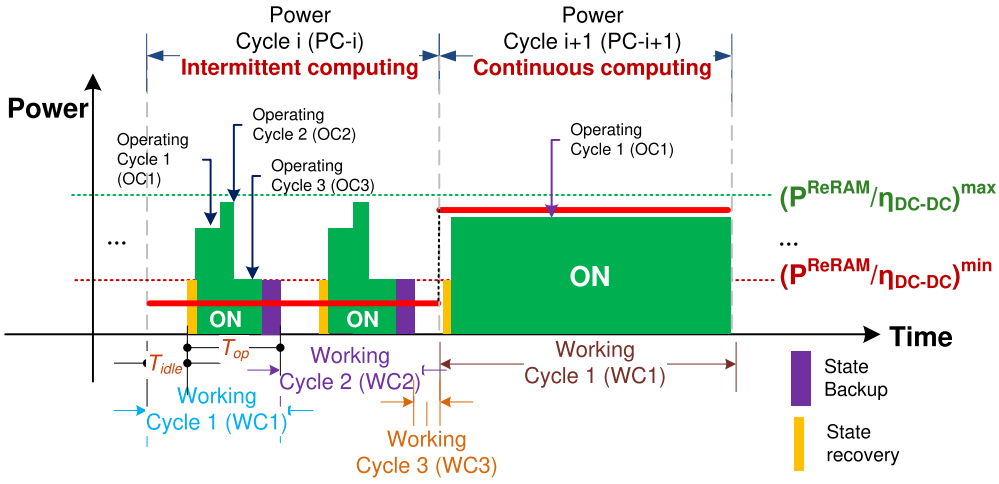


Fig. 4. Two computing schemes and key terms.

**Operating Cycle (OC)**: The consecutive working period consists of a series of operation cycles, each denoting one particular operation style with a specific tile activation on ReRAMs.

Figure 4 presents an example of two typical computing schemes in power cycles (PC) $i$ and $i+1$. *PC-i* consists of three WCs. For the first two WCs, each has three OCs corresponding to different ReRAM activation styles. The third WC only has an idle period for charging. However, *PC-i+1* only consists of one WC with one OC, which implies a *continuous execution scheme*. Note that each operation period ends when the discharging voltage drops to a specified threshold $V_{th}$ (e.g. 0.24v) in order to avoid the cold boot overhead of the capacitor [5].

In general, there are two types of ReRAM computing schemes: *intermittent computing* and *continuous computing*. The intermittent computing scheme occurs when the scheduled load power will drain storage faster than incoming power can refill it. The continuous computing scheme occurs when scheduled load power is less than the delivered power i.e. the power supply is sufficient to support a continuous computing at the load side. In Figure 4, the computation in *PC-i* is intermittent, whereas that in *PC-i+1* is continuous. Since the ReRAM load is decomposable and can be activated with different tile sizes, the execution style in each power cycle is controllable accordingly.

The intermittent computing style can work under different intermittent patterns in a power cycle. An *intermittent pattern* means a series of activation solutions with different ReRAM tiling strategies, that consists of the purely capacitor charging time $T_{idle}$ and a series of OCs in a working cycle. The intermittent pattern can be controlled by the charged voltage $V_{top}$ and discharged voltage $V_{bottom}$. Since this work focuses on the effectiveness of the two computing schemes, and as a result, we do not explore different patterns of intermittent computing. Instead, we just choose one intermittent degree, where we always charge the storage capacitor to $V_{top}$ and then run the ReRAM accelerator until the discharging voltage drops to $V_{bottom}$. Note that the discharging voltage of the storage capacitor is the input voltage $V^{DC-DC-in}$ to the DC-DC converter. As $V^{DC-DC-in}$ drops during ReRAM computing, we iteratively choose the ReRAM tile size $\langle m, n, d\rangle^{interm}$ with the highest throughput in terms of one WC under different values of $V^{DC-DC-in}$. When $V^{DC-DC-in}$ decreases to $V_{bottom}$, the accelerator enters a power failure status, indicated as $T_{idle}$ period.

With the continuous computing style for a power cycle as shown in Figure 4, we just choose one ReRAM tiling strategy that matches the harvested power so that computation can be continuous

Table 1. Parameters of the Energy Delivery and Performance Models

| Component | Parameter | Description |
|---|---|---|
| Capacitor side | $Q^{Cap}$, $E^{Cap}$ | Charge and energy of the storage capacitor |
| | $Size^{Cap}$ | storage capacitor size |
| | $Q^{harv}$ | Harvested charge at the capacitor input side |
| ReRAM load side | m, n, d | ReRAM activation tile for one convolution layer:<br>m: row size, n: column size; d: duplication factor |
| | $Q^{ReRAM}$, $E^{ReRAM}$, $P^{ReRAM}$ | Charge, consumed energy and power by ReRAM |
| | $Q^{loss}$ | Leakage charge of the storage capacitor |
| | $P^{ReRAM}$ | ReRAM load power |
| | $V^{ReRAM}$ | ReRAM input voltage |
| DC-DC side | $V^{DC-DC-in}$ | DC-DC converter input voltage |
| | $\eta^{DC-DC}$ | DC-DC converter efficiency |
| OC-level | $E^{ovh}$ | Energy consumed on status backup and recovery upon a power failure |
| | $t^{ovh}$ | Time consumed on status backup and recovery upon a power failure |
| | $t^{OC}$ | Time length of operating cycle OCi |
| | $Q^{OC0}$ | Capacitor charge at the beginning of the current operating cycle |
| | $T_{PC}$ | Time length of power cycle |

throughout the whole power cycle. This also implies that if the input power at the load side is less than the minimum requirement to activate the ReRAMs, it is regarded as power failure.

Although intermittent computing and continuous computing can be determined by tuning the load power, both have their own favorable working zones, considering the trade-offs among computation progress, idle overhead and data backup & recovery overhead. The selection of the computing scheme will be further described in Section 3.4.

### 3.3 Energy Delivery and Performance Modeling

The most efficient computing scheme under different incoming power levels can be determined based on the energy delivery and performance models of the whole system. For each OC, we can model the relationship among the key components of ReRAM load, DC-DC converter, and capacitor storage. The relevant parameters are described in Table 1. Note that the two functions $P^{ReRAM} = \text{Func}(\langle m,\ n,\ d\rangle)$ and $\eta_{DC-DC} = \text{Func}(V^{DC-DC-in}, P^{ReRAM})$ are formulated by HSpice, taking $Q^{loss}$ into account. The details of how power is extracted are further elaborated in Section 4.1.

By analyzing the relationship between the key components of storage capacitor, DC-DC converter and ReRAM load, we can derive the following equations which are the basis of system-level modeling.

- At the capacitor side:

$$\begin{cases} Q^{Cap} = Q^{Cap}_{OC0} + dQ^{harv} \times t_{OC} - (dQ^{ReRAM} + dQ^{loss}) \times t_{OC} \\ E^{Cap} = Q^{Cap} \times V^{DC-DC-in} \\ \Delta V^{DC-DC-in} = \Delta Q^{Cap} / Size^{Cap} \end{cases} \tag{1}$$

Equation 1 finds the stored charge at the storage node which is received from the harvester and delivered to the load during every operating cycle. Based on the voltage of the storage capacitor, we can find the energy storage at any given point in time.

• At the ReRAM load side:

$$\begin{cases} E^{ReRAM} = P^{ReRAM} \times (t_{OC} - t_{ovh}) + E^{ovh} \\ P^{ReRAM} = Func(\langle m,\ n,\ d\rangle) \\ dQ^{ReRAM} = P^{ReRAM}/V^{ReRAM} \end{cases} \tag{2}$$

Equation 2 calculates the total energy consumed during each operating cycle, which includes the status backup overhead on top of the static power dissipation of the analog compute.

• At the DC-DC converter side:

$$\begin{cases} E^{Cap} = E^{ReRAM}/\eta_{DC-DC} \\ \eta_{DC-DC} = Func(V^{DC-DC-in}, P^{ReRAM}) \end{cases} \tag{3}$$

Lastly, Equation 3 calculates the total energy delivered by the storage node to the ReRAM accelerator which includes the energy wasted at the DC-DC converter. The efficiency of the DC-DC converted is the ratio of the power consumed at the input to the power delivered at the output. Varying from design to design, the DC-DC converter's peak efficiency depends on the input voltage range and the output load power.

On top of the above modeling, scrutinizing the energy delivery relationship from the storage capacitor side to the ReRAM load side through the DC-DC converter yields the following insights:

• $\eta_{DC-DC}$ is dynamically changing with $V^{DC-DC-in}$ and ReRAM load.

• ReRAM load can be exploited as an energy delivery tuner of the whole system. These models provide quantitative foundations for the reconfiguration of ReRAM load tuner to impact the whole system.

By putting the above models into a working cycle, we can measure the performance and energy efficiency. Since the objective of this work is to make as much forward progress as possible in a limited energy environment, both throughput and energy efficiency are key evaluation metrics. Equation 4 models the throughput measured by MAC operations per time unit for a working cycle ($WCj$) including $i$ OCs.

$$Thr^{WCj} = \left(\sum_{OC1}^{OCi} MACs\right)/T_{WCj} \quad = \left(\sum_{OC1}^{OCi} (\langle m,\ n,\ d\rangle)^{tile-OCi} \cdot (t_{OCi}/(\#cyc \cdot freq))\right)/T_{WCj} \tag{4}$$

On the other hand, for a power cycle ($PCk$) including $j$ WCs, the throughput can be modeled as in Equation 5. The measurement unit for throughput is MACs/second.

$$Thr^{PCk} = \left(\sum_{WC1}^{WCj}\sum_{OC1}^{OCi} (\langle m,\ n,\ d\rangle)^{tile-OCi} \cdot (t_{OCi}/(\#cyc \cdot freq))\right)/T_{PCk} \tag{5}$$

Similarly, the energy efficiency for $PCk$ is modeled by Equation 6. The measurement unit for throughput is Joules/inference.

$$Eff^{PCk} = (P^{harv} \cdot T_{PCk})/\left(\sum_{WC1}^{WCj}\sum_{OC1}^{OCi} (\langle m,\ n,\ d\rangle)^{tile-OCi} \cdot (t_{OCi}/(\#cyc \cdot freq))/\#MAC\,per\,inf\right) \tag{6}$$

### 3.4 MaxTracker Algorithm

It is known that the DC-DC converter efficiency keeps changing during the charging and discharging processes. In this work, the proposed *MaxTracker* is capable of selecting the best ReRAM activation tile under different operating states.

*3.4.1 Harvesting Case Study.* Taking into account the relationship of the power levels at the capacitor output side and ReRAM input side, there exist three cases. For each case, *MaxTracker* can provide the best computing scheme and ReRAM tiling solution to achieve high performance.

**Case 1:** $P^{harv} > (P^{ReRAM}/\eta_{DC-DC})^{max}$

Here, the harvested power is sufficient to meet the requirement of continuous computing with the largest ReRAM activation size even under the smallest DC-DC converter efficiency. In this case, we do not care about energy delivery efficiency any more. To achieve the largest throughput, the best solution is to always choose the ReRAM activation with the full size for each convolution layer and adopt the continuous computing scheme.

**Case 2:** $(P^{ReRAM}/\eta_{DC-DC})^{min} \leq P^{harv} \leq (P^{ReRAM}/\eta_{DC-DC})^{max}$

Alternatively, the harvested power could be between the maximum and minimum required power at the ReRAM side. In this case, if we choose large activation sizes that will deplete capacitor charge faster than it can be refilled, it implies an intermittent computing style that incurs extra overhead of state backup/recovery. In contrast, if we utilize small activation sizes with which the capacitor charge is increasing, it implies that the incoming power is sufficient to support a continuous computing style. Since the intermittent computing and the continuous computing have their own favorable activation solutions for ReRAMs, we select the one by comparing the statically estimated throughput according to Equation 5. An offline throughput lookup table provides the throughput (computation progress) that is built considering the key variables of incoming power level, capacitor size, and activation factor $\langle m,\ n,\ d \rangle$ as shown in Figure 3.

**Case 3:** $P^{harv} < (P^{ReRAM}/\eta_{DC-DC})^{min}$

In this scenario, the harvested power is weak and cannot even support the minimum required power for ReRAM activation. In this case, it is not possible for the harvested power to support the continuous computing even with the smallest tile size activation under the highest $\eta_{DC-DC}$. It implies an intermittent computing with appropriate tile size activation. Since energy delivery efficiency is the most critical factor in this case, we first start with the activation strategy with the highest $\eta_{DC-DC}$. Then our algorithm, iteratively tracks and selects the best tile activation strategy for the operating cycles of each working cycle.

Our proposed *offline-MaxTracker* strategy, by recognizing different operating states, determines the best combination of computing scheme and the ReRAM activation strategy. This idea is also abstracted at a software-level of *offline-MaxTracker* in Figure 3.

*3.4.2 MaxTracker Algorithm.* For the different cases discussed above, we can determine the best activation tile for different operating states offline. Then, at runtime, we implement the solution to perform the computations based on the current operating state. The runtime *MaxTracker* algorithm is given in Algorithm 3. The two procedures *Thr-INTERM* and *Thr-CONTI* estimated the throughput for the concerned working cycle with the intermittent computing scheme and continuous computing scheme, respectively, given the operating state and the circuit-level lookup table. This lookup table is derived from circuit-level simulation where the relationship between the load power, capacitor discharging voltage and DC-DC conversion efficiency is quantitatively revealed. The main program *MaxTracker* first judges the use case and calls the two functions to determine and implement the computing scheme and ReRAM activation solutions. The links of the two offline procedures (Algorithm 1 and Algorithm 2) and the online tracking strategy (Algorithm 3) are depicted in Figure 5.

## 4 EXPERIMENTS

In this section, we first introduce the circuit setup and simulation methodology and then present the collected results, observations, and insights. Note that all the experiment sources will be made available for others to replicate after blind review.

**ALGORITHM 1:** Thr_INTERM (offline)

**Require:** the parameters listed in Table 1;
the lookup table derived from circuit-level simulation;
$T_{WC}$: the total time length of the concerned working cycle;
$T_{op}$: the time length of the current operating cycles regarding the concerned working cycle;
$WC$: the concerned working cycle;
**Ensure:** the estimated throughput along with ReRAM activation solution with the intermittent computing scheme of a working cycle;
1: **procedure** THR_INTERM $T_{WC}$=0; $T_{op}$=0;
2: Calculate the instantaneous throughput for each possible operating state tuple $\langle V^{DC-DC-in}, Q^{Cap}$, Power level, $\eta_{DC-DC} \rangle$ with each $\langle m, n, d \rangle$ solution;
3: Select $\langle m, n, d \rangle^{OC1}$ activation solution with which the maximum throughput can be obtained for OC1 of the concerned WC from the lookup table;
4: **Step 1**: Charge under the current power level until $V^{top}$ is reached with charging time of $T_{idle}$; $T_{WC}$=$T_{WC}$+$T_{idle}$;
5: **Step a**: Calculate the operating time $T_{oc}$ with $\langle m, n, d \rangle^{OC1}$ until $\Delta V^{DC-DC-in} \geq \Delta V_{\epsilon}$ (e.g. $V_{\epsilon}$=0.05V);
6: **Step b**: Update operating state tuple $\langle V^{DC-DC-in}, Q^{Cap}$, Power level, $\eta_{DC-DC} \rangle$;
7: **Step 2**: Search activation solution $\langle m, n, d \rangle^{max-OC2}$ with which the maximum throughput can be obtained under current operating state from the lookup table;
8: **Step c**: $T_{op}$=$T_{op}$+$T_{oc}$; $T_{WC}$=$T_{WC}$+$T_{op}$;
9: **Step 3**: Repeat Step a-c until $Q^{Cap}$ cannot support the operating with the smallest activation tile size $\langle m, n, d \rangle^{min}$ any more;
10: Record the OC activation solution sequences $\langle m, n, d \rangle^{interm}$: $\langle \langle m, n, d \rangle^{OC1}, \langle m, n, d \rangle^{OC2}$ ...$\rangle$
11: Calculate the throughput $Thr^{interm}$ according to Equation 4;
12: Return $Thr^{interm}$ along with $\langle m, n, d \rangle^{interm}$;
13: **end procedure**

**ALGORITHM 2:** Thr_CONTI (offline)

**Require:** the parameters listed in Table 1;
the lookup table derived from circuit-level simulation;
$T_{WC}$= $T_{PC}$; //In the continuous computing scheme, thw working cycle time is equal to the power cycle time
**Ensure:** the estimated throughput along with ReRAM activation solution with the continuous computing scheme;
1: **procedure** THR_CONTI
2: Calculate the instantaneous throughput for each possible operating state tuple $\langle V^{DC-DC-in}, Q^{Cap}$, Power level, $\eta_{DC-DC} \rangle$ with each $\langle m, n, d \rangle$ solution;
3: Select the largest activation solution $\langle m, n, d \rangle^{conti}$ with which $P^{ReRAM}/\eta_{DC-DC}$ across all operating states is less than the power level during the current power cycle;
4: Calculate the throughput $Thr^{conti}$ at this power level according to Equation 4;
5: Return $Thr^{conti}$ along with $\langle m, n, d \rangle^{conti}$;
6: **end procedure**

### 4.1 Circuit Setup

To evaluate the benefits of our scheduling technique on non-ideal real world systems with unstable incoming power, we choose an on-chip, low voltage rating storage capacitor array powered by piezoelectric, thermal, WiFi or RF harvesters. Unlike off-chip super-capacitors, on-chip energy storage has lower charge capacity but is faster to charge. In our experimental set-up, the default storage capacitor sizes under study are 1 nF for piezoelectric and WiFi harvesters, 4.7 nF for thermal harvesters and 22 nF for TV-RF harvesters. The optimal storage capacitor size is chosen specific to each power source to such that the storage node does not frequently overflow or enter power failure, which can affect the overall throughput during run-time. We further analyze the impact of the storage capacitor size on the system-level performance in Section 4.5.

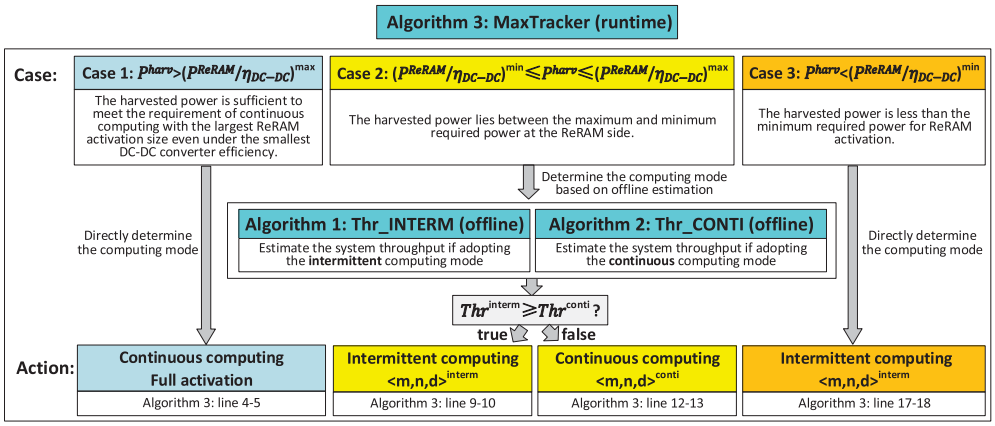


Fig. 5. The algorithm structure of MaxTracker.

**ALGORITHM 3:** MaxTracker (runtime)

**Require:** the parameters listed in Table 1; $T_{PC-time}$; $T_{WC-count}$; $T_{OC-count}$;
**Ensure:** computing scheme and activation tile $\langle m,\ n,\ d\rangle$ for each convolution layer for the first OC of a WC given an operating state;

**for** each WC **do**
  **switch** $P^{harv}$ **do**
    **case** $P^{harv} > (P^{ReRAM}/\eta_{DC-DC})^{max}$
      Computing scheme = *continuous*;
      ReRAM activation solution = full-activation tile $\langle m,\ n,\ d\rangle$ for each layer;
      $break$;
    **case** $(P^{ReRAM}/\eta_{DC-DC})^{min} \leq P^{harv} \leq (P^{ReRAM}/\eta_{DC-DC})^{max}$
      **if** $Thr^{interm} \geq Thr^{conti}$ **then**
        Computing scheme = *intermittent*;
        ReRAM activation solution = $\langle m,\ n,\ d\rangle^{interm}$;
      **else**
        Computing scheme = *continuous*;
        ReRAM activation solution = $\langle m,\ n,\ d\rangle^{conti}$;
      **end if**
      $break$;
    **case** $P^{harv} < (P^{ReRAM}/\eta_{DC-DC})^{min}$
      Computing scheme = *intermittent*;
      ReRAM activation solution = $\langle m,\ n,\ d\rangle^{interm}$;
      $break$;
**end for**

When the supply voltage is too low to operate the sensor node, the deployed system enters power failure and will need to wait until the energy storage is sufficiently charged to resume computation. To reduce the frequency of such failures, we consider step-up DC-DC converters to allow low voltages at the storage capacitor to supply the ReRAM accelerator.

We choose an appropriate state-of-the-art step up DC-DC converter [2] and extract the efficiency, $\eta_{DC-DC}$, as a function of input voltage and output load current. The chosen converter design features configurable conversion ratios of 2:3, 1:2 and 2:5 modes. Previous studies have shown that this multi-ratio DC-DC converter can be leveraged for a system to operate efficiently at various storage capacitor voltages [1, 31, 33]. Lastly, we resize the reactive components, transistor switches and gate-drive circuits of the design to calibrate the DC-DC converter to support the

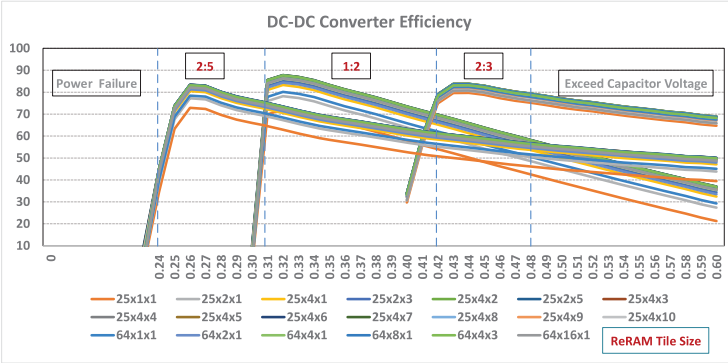


Fig. 6. A reconfigurable DC-DC Converter Efficiency.

Table 2. ReRAM Accelerator Components

| Single Tile Breakdown | | | | |
|---|---|---|---|---|
| **Component** | **Quantity** | **Active Size** | **ON Power ($\mu$W)** | **OFF Power ($\mu$W)** |
| ReRAM Crossbar | 256 × 256 | 25 × 1 | 14.4 | 0.032 |
| | | 150 × 16 | 253.4 | |
| 1-bit DAC | 256 | 25 | 46.5 | 0.811 |
| | | 150 | 272.4 | |
| 4-bit ADC | 16 | 1 | 1.06 | 0.043 |
| | | 16 | 21.8 | |
| Shift & Add | 16 | 1 | 0.302 | 0.046 |
| | | 16 | 4.10 | |
| **Tile Total** | **64 Kb** | **25 × 1** | **62.2** | 0.932 |
| | | **150 × 16** | **551.8** | |

| ReRAM PIM Core Breakdown | | | | |
|---|---|---|---|---|
| **Component** | **Quantity** | **Active Size** | **ON Power (mW)** | **OFF Power (mW)** |
| ReRAM Tile | 4 | 25 × 1 × 1 | 0.062 | 0.0037 |
| | | 150 × 16 × 4 | 2.21 | |
| Global Control | 1 | 1 | 0.013 | 0.005 |
| **Core Total** | **256 Kb** | **25 × 1 × 1** | **0.075** | 0.0087 |
| | | **150 × 16 × 4** | **2.22** | |

| System Parameters | | |
|---|---|---|
| **Module** | **Parameter** | **Specification** |
| ReRAM PIM Core | Supply Voltage | 0.6 V |
| | Clock Frequency | 200 MHz |
| Energy Storage | Storage Capacitor Size | 0.47 - 47 nF |
| DC-DC Converter | Conversion Ratio(s) | 2:3, 1:2, 2:5 |
| | Peak Efficiency | 88% |
| ReRAM Memory (256 KB) | Max Bandwidth | 1.7 GB/s |
| | Read Latency/Energy | 1.6 ns/2.22 pJ |
| | Write Latency/Energy | 100 ns/88.2 pJ |

entire load current range of the ReRAM accelerator. Figure 6 shows the comparison of the three single ratio DC-DC converters and the multi-ratio DC-DC converter. In our design, the step up ratios for the DC-DC converter input voltage zones of 0.24V-0.31V, 0.31V-0.42V and 0.42V-0.48V are 2:5, 1:2 and 2:3, respectively.

Table 2 summarizes the hardware component details and circuit-level parameters used in our evaluations. For the ReRAM-based sensor node, we conduct HSPICE circuit simulation to characterize the average power, leakage power and charge consumed by different ReRAM tile sizes. The ReRAM peripheral circuit components are implemented using Predicted Technology Model

(PTM) [43] in 22nm CMOS and the ReRAM devices are calibrated to [19]. Table 2 also shows a detailed breakdown of the power consumed by the smallest and the largest activation sizes supported within a single 256 KB ReRAM Core.

At every row of the crossbar, the 1-bit digital-analog converter (DAC) component consists of a simple two-level voltage driver circuit which serially passes the input activations one bit at a time. Pitch-matched across 16 columns, the 4-bit analog-digital converter (ADC) consists of a multi-reference current sense amplifier [39] connected to 4 latches and a simple control logic which successively finds the output 1 bit per clock cycle. Lastly, the shift and add unit consists of a low power accumulator register to calculate the final output of the convolution. Because the latches and registers consume a significant amount of leakage power, we optimize the volatile data buffers to shut off during lower activation sizes.

The ReRAM memory unit contains both the back-up data and the activation solution look-up table as further elaborated in Section 4.2. The latency and energy overheads of the ReRAM memory are evaluated using NVsim [10] configured with low power routing strategy. During OCs, the non-volatile memory is primarily used as a table look-up to select the activation solution $\langle m,\ n,\ d\rangle$, represented by a 3-tuple of 8-bit unsigned integers.

### 4.2 System Simulation

ReRAM activation solutions (*Thr_INTERM* and *Thr_CONTI*) are precomputed. They determine the ReRAM activation solution $\langle m,\ n,\ d\rangle$ for each operating environment on top of their own computing scheme. At runtime, we simulate CNN computations using *Runtime-MaxTracker* algorithm in a cycle-accurate fashion, where the tables generated by *Offline-MaxTracker* are looked up to dynamically adapt to status changes of the whole circuit systems. Specifically, the lookup table stores the information of ReRAM tiling solutions, capacitor discharging voltage, harvested power levels and DC-DC conversion efficiency. In our evaluation, the table size ranges from $231 \times 58$ to $871 \times 58$. The energy overhead of the ReRAM look-up table is 2.22 pJ per 24-bit read as reported in Table 2. The activation solution $\langle m,\ n,\ d\rangle$ stored in each entry is represented by 3 bytes $\langle 8\text{-}bit, 8\text{-}bit, 8\text{-}bit$ to represent the different configuration sizes. Including routing and data buffers, the look-up energy accounts for 20% of the total energy consumed by the maximal tile activation where $\langle m,\ n,\ d\rangle = \langle 150,\ 16,\ 4\rangle$.

The time required to prepare the offline lookup table depends on the power level number and discharging voltage granularity. For the lookup table with 10 power levels and discharging voltage ranging from 0.24V to 0.48V with a granularity of 0.1V, the preparation time is 99.7s with our PC (2.3GHz CPU and 24GB main memory). The time to look up the offline table is decided by multiple factors such as power levels, discharging voltage granularity and the searching approach. Take LeNet for example, it was observed that the average table lookup time is $8\mu$s on our PC under different power sources. For the power cycle with continuous computing mode, only once table lookup is needed. For the power cycle with intermittent computing mode, the lookup number varies from several to thousands per power cycle, depending on power inputs. In terms of the overall execution, the ratio of the table lookup time to the entire execution time can be negligible (mostly less than 1%). Note that the lookup time overhead has been counted in the performance and energy efficiency evaluations.

The entire system runs on a 200MHz clock. The basic MCU is an Ultra-Low-Power (ULP) component with fixed power consumption of $10\mu$W. To record the runtime status, twenty registers are needed. During backup and recovery upon a power failure, the ReRAM memory is also used to backup system status and store intermediate MAC results between convolution layers. ~90% of the computations performed on the ReRAM accelerators are MAC operations which are simulated at an execution cycle level [45]. Other functional units (e.g., Pooling, FC, Sigmoid) of the CNNs are

Table 3. IoT-Practical CNN Workloads

| CNN | Layer | Kernel | ReRAM Size | Cycles | Acti. power | Input |
|---|---|---|---|---|---|---|
| PV | Input | | | | | 1@50 × 50 |
| | Conv1 | 8@6 × 6 × 1 | 36 × 8 | 8 | 242.9$\mu$W | 8@45 × 45 |
| | Conv2 | 12@3 × 3 × 8 | 72 × 12 | 12 | 356.9$\mu$W | 12@20 × 20 |
| | Conv3 | 16@3 × 3 × 12 | 108 × 16 | 12 | 425.8$\mu$W | 16@8 × 8 |
| | Conv4 | 10@3 × 3 × 16 | 144 × 10 | 16 | 460.6$\mu$W | 10@6 × 6 |
| | Conv5 | 6@3 × 3 × 10 | 90 × 6 | 12 | 253.6 $\mu$W | 6@4 × 4 |
| FR | Input | | | | | 1@32 × 32 |
| | Conv1 | 4@5 × 5 × 1 | 25 × 4 | 4 | 122.8$\mu$W | 4@28 × 28 |
| | Conv2 | 16@4 × 4 × 4 | 64 × 16 | 12 | 425.8 $\mu$W | 16@10 × 10 |
| LeNet | Input | | | | | 1@32 × 32 |
| | Conv1 | 6@5 × 5 × 1 | 25 × 6 | 4 | 154.1$\mu$W | 6@28 × 28 |
| | Conv2 | 16@5 × 5 × 6 | 150 × 16 | 16 | 565.3$\mu$W | 16@10 × 10 |
| HG | Input | | | | | 1@28 × 28 |
| | Conv1 | 6@5 × 5 × 1 | 25 × 6 | 4 | 154.1$\mu$W | 6@24 × 24 |
| | Conv2 | 12@4 × 4 × 6 | 96 × 12 | 12 | 356.9$\mu$W | 12@8 × 8 |

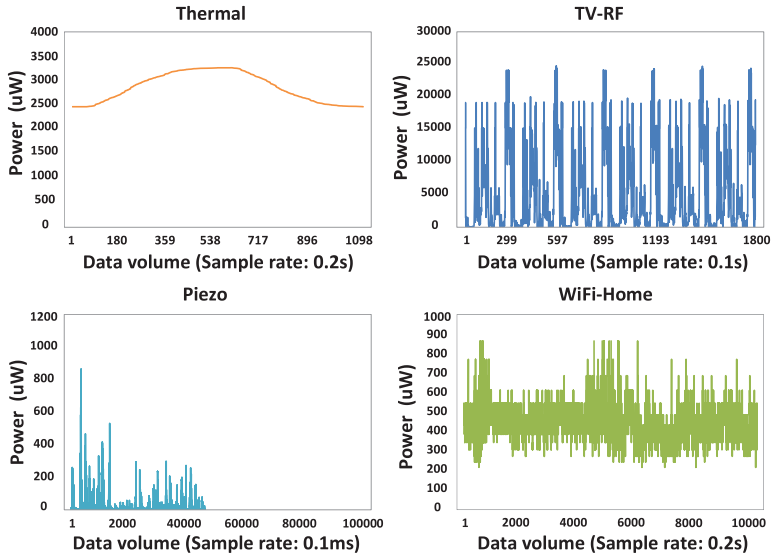


Fig. 7. blue Four harvesting power sources.

assigned a fixed latency and power. Four lightweight CNNs listed in Table 3 are evaluated on five power traces depicted in Figure 7.

Four approaches are evaluated: a traditional pure circuit optimizer (*Fixed-Full*), a typical domain-specific pure circuit optimizer (*Fixed-Tiling*), a coarse-grained dynamic tiling optimizer *Coarse Dynamic Tiling* and the proposed *MaxTracker*. In the *Fixed-Full* version, the ReRAM load is fixed as the largest activation ReRAM size for each layer. In the *Fixed-Tiling* version, on the other hand, the ReRAM load is fixed to the activation ReRAM tiling size which is favorable to the most typical power level of each source. In the *Coarse Dynamic Tiling* optimizer, the ReRAM load varies depending on the energy delivery efficiency at runtime. There are only three granularities for the ReRAM tuner. In Load Level 1, the ReRAMs are fully activated for each convolutional layer which is the same as the *Fixed-Full* version. In Load Level 2, the activation size is a half of that in Load Level 1. Similarly, the activation size in Load Level 3 is a half of that in Load Level 2. In this way, the accelerator runs under only four power levels with the three load levels. Finally, our proposed *MaxTracker* takes into account the interaction among ReRAM activation size tuning, capacitor

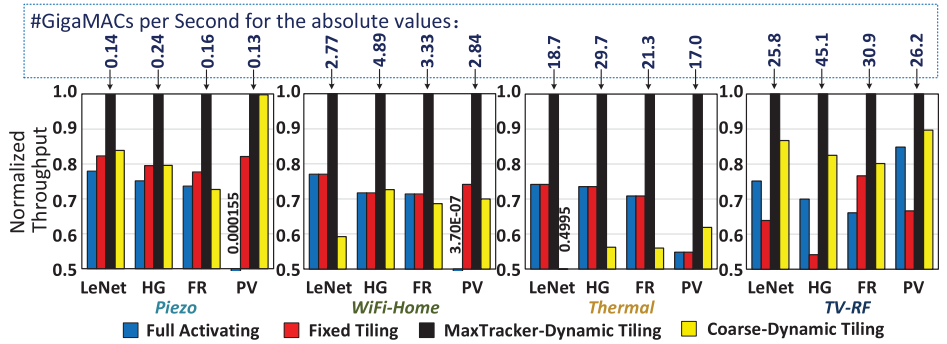


Fig. 8. Throughput of four CNNs across four power sources.

voltage changing, and DC-DC converter efficiency, and select the most appropriate find-grained ReRAM tile to fit the runtime system condition.

We evaluate performance, energy efficiency, sensitivity to capacitor size. We compare as well as combine our proposed method with recent state-of-the-art techniques. Note that the cases where the power inputs are too weak to support meaningful ReRAM activations are removed from our statistics, to avoid bias to our results. However, those extremely large or small results are retained and marked in the figures.

### 4.3 Performance

For each CNN on each power trace, we report throughput and normalized performance as shown in Figure 8. The following observations are made from the results:

• Overall, *MaxTracker* always achieves the best throughput over pure circuit optimizers with fixed full ReRAM activation, *Full Activation* and fixed tile-size activation on ReRAMs, *Fixed Tiling* across all the CNNs and all the power sources. An average speedup of 38.4% (up to 51.3% ) over the *Full Activation* scheme, 40.3% (up to 84.8%) over the *Fixed Tiling* scheme, and 41.8% (up to 100.2%) over the *Coarse Dynamic Tiling* are observed. These results are compelling evidence of the benefits of the proposed adaptation strategy for end-to-end efficiency optimization. The underlying reason can lie in the aspect that *MaxTracker* is capable to dynamically decide the best ReRAM activation tiles to accommodate the changing system status.

• Comparing between the two fixed baseline strategies, we observe *Full activation* slightly outperforms *Fixed tiling* approach for *LeNet*, *HG* and *PV* networks with the *TV-RF* power source. The use of a fixed ReRAM activation size suitable for the most common power level degrades the throughput, when the incoming power varies frequently. In the case of the TV-RF power profile, a *Full activation* approach utilizes the ReRAM accelerator more effectively when the incoming power is higher than the typical power level.

• Comparing *Coarse Dynamic Tiling* to the two fixed baseline strategies, we observe that the former does not always outperform the latter two. For the cases of *Thermal* and *WiFi-Home* power sources, because only one power level is dominant, the fixed strategies can already provide the favorable load for it. The simple *Coarse Dynamic Tiling*, however, cannot select an appropriate load from the three options for this dominant power level. This is why *Coarse Dynamic Tiling* sometimes delivers worse performance than the fixed strategies.

• The *Full activation* approach for *PV* suffers extremely low throughput for *Piezo* and *WiFi-Home* harvesters because the signal from the power sources are too weak to support five-layer full

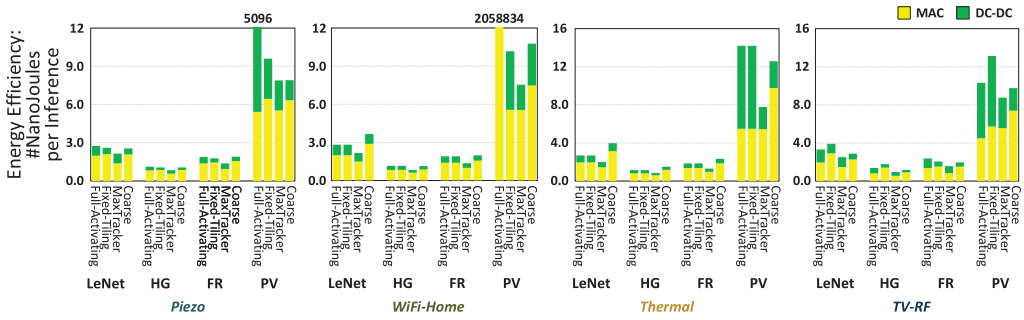


Fig. 9. Energy efficiency of four CNNs networks across four power sources.

activation of *PV*. Consequently, this result in our performance evaluation is an anomaly and is not factored into the average speed-up reported of *MaxTracker* over *Full Activation*.

• The absolute throughput numbers are directly correlated with the signal strength of power sources. Piezo is the weakest, leading to the smallest throughput, while the strongest TV-RF leads to the largest throughput.

### 4.4 Energy Efficiency

We evaluate energy efficiency by measuring nanojoules per inference, as shown in Figure 9. The smaller value means higher energy efficiency. In other words, this metric quantifies the relative energy saved per inference by *MaxTracker* as compared to the other three approaches. At ReRAM load side, besides the MAC computations, the energy overheads such as data movements and other functional units are all included. Figure 9 also shows the energy breakdown between ReRAM MAC and DC-DC converter.

• Overall, the *MaxTracker* approach achieves an average of 39.8%, 43.6% and 35.6% energy savings over *Full Activation*, *Fixed Tiling*, and *Coarse Dynamic Tiling* respectively. Our evaluation also shows that the highest energy efficiency gains of *MaxTracker* reaches 71.0% against all the baselines.

• Consistent with the performance observations, the *Fixed Tiling* strategy competes with the *Full Activation* strategy for several scenarios. This is determined by how frequently the most common power level occurs in the power profiles. Again, *Coarse Dynamic Tiling* does not always compete with the fixed tiling strategies. This is because the *Coarse Dynamic Tiling* only provides three tiling solutions which may mismatch the dominant power levels.

• Except for the extremely weak energy harvesting scenarios, the maximal energy saving percentages of *MaxTracker vs. Full Activation* and *MaxTracker vs. Fixed Tiling* are similar. This is because the maximum results occur at the cases with *PV&Thermal* where the *Full Activation* scheme chooses the same activation solution as that of the *Fixed Tiling* scheme.

• The energy breakdown between ReRAM MAC and DC-DC converter shows that the MAC part consumes the larger portion (68.7% on average) of the energy for all strategies and all scenarios. This is partially due to the multi-ratio DC-DC converter design which can guarantee a rather high energy delivery. And it is further observed that the portion of MAC for the *MaxTracker* is much larger than the others. This further validates that the *MaxTracker* scheme can better deliver energy to the ReRAM load than the fixed tiling and coarse dynamic tiling schemes. For the *PV* CNN, it is found that the energy percentage of DC-DC converter is larger than that of other CNNs. This is possibly caused by the low energy efficiency of *PV* execution, which leading to a high energy portion consumed by DC-DC converter.

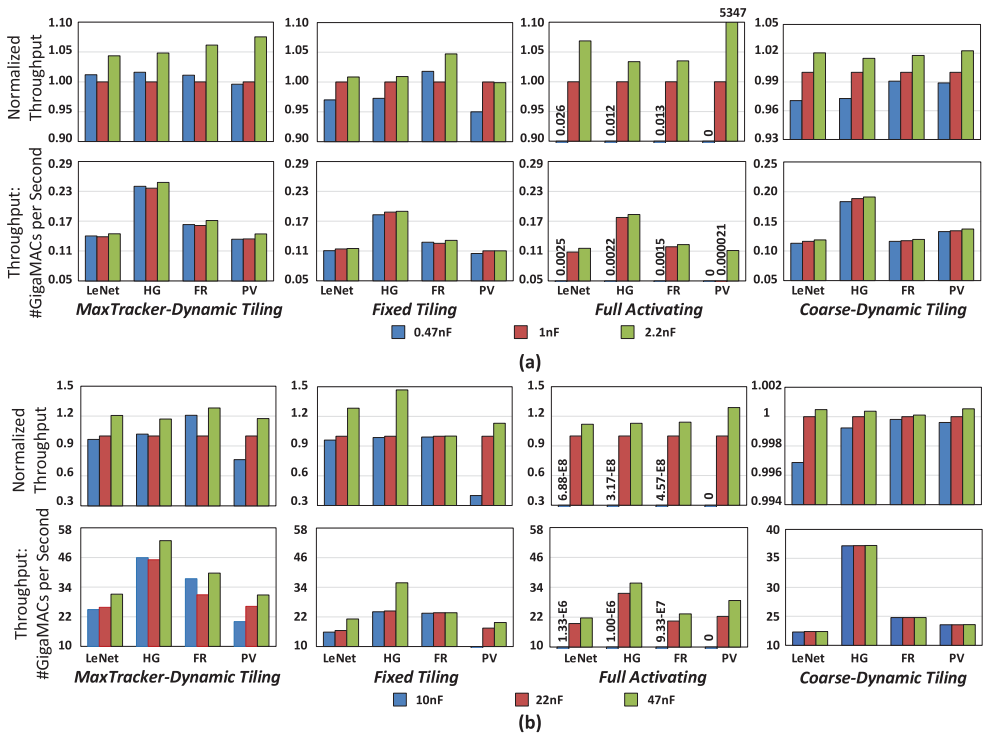


Fig. 10. Sensitivity study of storage capacitor size on performance. (a) Throughput with piezo, and (b) Throughput with TV-RF.

### 4.5 Sensitivity Study of Storage Capacitor Size

Figure 10 presents the throughput and normalized throughput so as to compare across different CNN networks and capacitor sizes with the two power sources, *Piezo* and *TV-RF*, as case studies. Among the four power traces in Figure 7, the *Piezo* and *TV-RF* harvests the weakest and strongest signal respectively. Therefore, the analysis in Figure 10 highlights the cases of lowest and highest incoming power of how the storage size affects the overall throughput. Regarding 2.2nF vs. 1nF with *MaxTracker*, average speedups of 5.7% (up to 7.52%) with *Piezo*. Regarding 47nF vs. 22nF with *MaxTracker*, average speedups of 20.9% (up to 28.2%) with *TV-RF* can be observed.

While our technique provides consistent performance benefits, the magnitude of improvement does not exhibit a trend with varying capacitor sizes. The three schemes exhibit somewhat different improvements on performance as capacitor size grows, leading to different efficiency factors dominating different points in the design space.

The results demonstrate the degree of sensitivity to the size of storage capacitor $Size^{Cap}$ for different schemes. For each scenario corresponding to each "Power&CNN" pair, the largest sized storage capacitor can lead to the best results on performance for most cases. Although the larger capacitor can deliver the best performance for most cases, it does not mean that the largest capacitor is always the optimal deployment. This is due to two reasons. One is that larger capacitors also imply larger leakage power. There exists an optimal point by trading off both the benefits and leakage power of a storage capacitor. The other reason is that a larger capacitor often means a bigger physical form factor, which may not be appropriate for tiny devices. In our future work, we will explore quantitative approaches to determine the optimal storage capacitor for a system, given an expected workload and input power distribution.

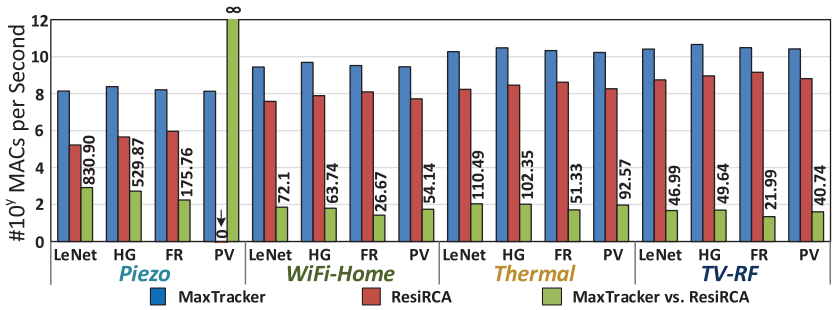


Fig. 11. Throughput comparison with *ResiRCA* [32].

### 4.6 Comparison with *ResiRCA* approach [32]

Figure 11 shows the throughput comparison with *ResiRCA* technique. To simulate the *ResiRCA* [32], we just schedule the ReRAM load to match the power level for each power cycle of a power profile, since there is no energy store and DC-DC converter with *ResiRCA*. It is assumed that the uncompleted results will be discarded if an inference cannot be finished in one power cycle.

The results show that *MaxTracker* outperforms *ResiRCA* significantly for all scenarios. By examining the simulation logs, we found that the primary reason for this is that the "Harvest-Store-Use" architecture supports energy accumulation to perform computing, thereby *MaxTracker* can achieve better performance. However, *ResiRCA* can only use the just-in-time power for computations. Hence, incoming power less than the minimum or more than the maximum requirements of ReRAM load is wasted.

A typical example is with the power *Piezo*, where *MaxTracker* outcompetes *ResiRCA* by hundred× and larger speedups. The reason is that *Piezo* is both extremely weak and strongly fluctuating, such that most power cycles cannot accommodate even the smallest ReRAM tile. This explains the abnormally large speedup with the largest five-layer network *PV*, because the tiny power input cannot support even one inference in a power cycle, without energy accumulation across cycles.

### 4.7 Comparison and Combination of *Capybara* Approach [5]

From the perspective of a co-designed hardware and software system with a reconfigurable power system, A. Colin et al. proposed *Capybara* [5], which supports declarative specification of tasks' energy demands or mixed demands with a reconfigurable storage capacity mechanism. With this capacitor reconfiguration ability, the system can flexibly meet the demands of capacity-constrained tasks and temporally-constrained tasks.

Different from the *Capybara* approach where the capacitor size is reconfigured to fit task reactivity, our proposed *MaxTracker* approach schedules the ReRAM activation to accommodate to the best operating state of the whole system. Actually these two techniques focus on different components of the system, and they can be combined to offer a further improvement based on their own.

Figure 12 presents the comparison and combination of *MaxTracker* and *Capybara* regarding the throughput with the power inputs of *Piezo* and *TV-RF*. In the experimental settings, the ReRAM load is fixed as a full activation size of *Traditional* style for *Capybara*, while for *MaxTracker*, the storage capacitor size is fixed as 2/3 full size of that in *Capybara*. Figure 12 shows that *MaxTracker* is competitive with *Capybara* with a speedup of around 13.6%.

*MaxTracker* and *Capybara* can also be combined to cooperate in a more flexible style. That is, both the ReRAM activation solution and the storage capacitor can be reconfigured at runtime to well exploit the harvested energy and boost the performance. The combination results show

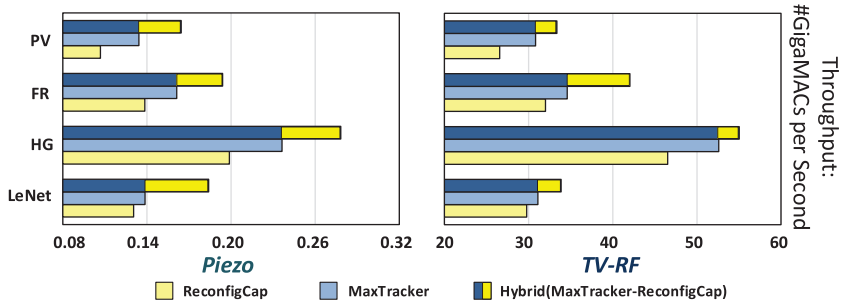


Fig. 12. Comparison and combination of *Capybara* approach.

that the hybrid *MaxTracker* and *Capybara* can provide further speedup. Their hybrid can offer an average of 17.0% (up to 32.7%) speedup on top of *MaxTracker*, and an average of 33.0% (up to 54.0%) speedup on top of *Capybara*. This significant improvement results from their two-fold benefits: (i) the reconfiguring capacitor efficiently extracts and accumulates the harvested energy; and (ii) the reconfiguring ReRAM load efficiently utilizes the stored energy.

## 5 RELATED WORK

Recent studies about how system firmware fits the fluctuating power supply can be categorized into the following three classes.

### 5.1 Hardware Reconfiguration

A. Colin et al. proposed *Capybara* [5], which supports declarative specification of tasks' energy demands or mixed demands with a reconfigurable storage capacity mechanism. With this capacitor reconfiguration ability, the system can flexibly meet the demands of capacity-constrained tasks and temporally-constrained tasks. Ma et al. propose a machine learning-based integrated architecture of NVP [22] on the basis of architecture exploration in [24]. The architecture integrates three micro-architectures of NP, NSP and OoO with different power requirements. In [22], energy-dependent data are fed into a lightweight neural network for (NN) future power level prediction, and then the most appropriate architecture which maximizes progress of an application for the next power level can be chosen.

Expanding the above idea further, two techniques are combined to improve energy efficiency [23]. One is resource scaling, which manages bottleneck resources in a reconfigurable OoO processor targeting lower energy per instruction (EPI). The other is dynamic frequency scaling, which aggressively leverages harvested energy. Both approaches are directed by a lightweight machine learning algorithm.

Aside from NN-based resource scaling techniques, a novel idea has been explored from the viewpoint of opportunistic responsiveness [20, 21]. The approach begins with the observation that the quality of older computations targeted for incidental computing can be gradually improved iteratively, if picked up over multiple incidental computing passes. The ultimate optimization of energy efficiency is achieved through a well matched retention time.

### 5.2 Dynamic Task Schedule

Besides allocating adaptive hardware resources to match the power input, there are works exploring finer-grained strategies to accommodate the changing harvesting power through adaptive task scheduling. Majid et al. proposed *Coala* [27], an adaptive task-based execution model. By means of task coalescing and splitting, Coala allows efficient execution on a sub-task scale so as to preserve

computation progress. The challenging issues of task transition and task termination are also well handled by Coala.

As the power requirement of the modern practical NN inference tasks such as DNN are several orders-of-magnitude greater than the current energy-harvesting systems, DNN inferences in intermittent computing are limited to extremely short bursts. To address this issue, Gobieski et al. proposed an intermittent DNN inference framework, *SONIC* [12, 13], to flexibly size tasks to grow or shrink to fit the energy budget. SONIC also uses Alpaca [25] tasks to avoid checkpointing and thus imposing very low overheads.

The most related technique to this work, *ResiRCA*, is able to dynamically activate different scaled ReRAM tiles [32] as described in Section 2. However, our targeted basic "Harvest-Store-Use"architecture is much more complex than the "Harvest-Use" architecture considered in *ResiRCA*. The results in Section 4.6 show that our proposed *MaxTracker* can provide better performance due to its capability to exploit energy accumulation.

### 5.3 Energy Efficiency Optimization

Prior work [42] avoids the efficiency loss and integration overheads of DC-DC converters by proposing a converter-less MPPT architecture. By matching the current-voltage characteristics of the photovoltaic cell to the sensor node, the sensor node directly draws the maximum energy efficiency from the harvester. Subsequently, [40] improves the system-level energy efficiency for converter-less architectures by tuning the clock frequency of the sensor node based on the voltage supplied by the harvester.

Our work differs from prior works in the fact that our algorithm considers the energy efficiency of on-chip DC-DC converters to maximize forward progress of the sensor node in a conventional energy harvesting with an on-chip energy storage component.

## 6 CONCLUSION

Targeting energy harvesting ReRAM-based CNN accelerators running on the edge, this paper proposes and evaluates a novel strategy, called *MaxTracker*, that tracks the maximum computation progress through dynamically tuning the ReRAM load power. *MaxTracker* can consistently maximize computation progress by tuning both computing schemes and activation tiling sizes of the ReRAM load to best match just-in-time operating states. Experimental results under various energy harvesting scenarios demonstrate the high efficacy of the *MaxTracker* strategy over a range of harvesting scenarios and CNN workloads. Furthermore, it is found that *MaxTracker* incorporated with the capacitor reconfiguring technique, *Capybara*, can further enhance the tunability and boost system performance.

## REFERENCES

[1] M. Al-Soeidat, H. Aljarajreh, H. Khawaldeh, D. D. Lu, and J. G. Zhu. 2019. A reconfigurable three-port DC-DC converter for integrated PV-Battery system. *IEEE Journal of Emerging and Selected Topics in Power Electronics* (2019), 1–1.

[2] A. Biswas, Y. Sinangil, and A. P. Chandrakasan. 2015. A 28 nm FDSOI integrated reconfigurable switched-capacitor based Step-Up DC-DC converter with 88% peak efficiency. *IEEE Journal of Solid-State Circuits (JSSC)* 50, 7 (2015), 1540–1549.

[3] Wei-Ming Chen, Yi-Ting Chen, Pi-Cheng Hsiu, and Tei-Wei Kuo. 2019. Multiversion concurrency control on intermittent systems. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD'19)*. 1–8.

[4] Wei-Ming Chen, Tei-Wei Kuo, and Pi-Cheng Hsiu. 2020. Enabling failure-resilient intermittent systems without runtime checkpointing. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* 39, 12 (2020), 4399–4412.

[5] Alexei Colin, Emily Ruppel, and Brandon Lucia. 2018. A reconfigurable energy storage architecture for energy-harvesting devices. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'18)*. 767–781.

[6] Jasper de Winkel, Carlo Delle Donne, Kasim Sinan Yildirim, Przemysław Pawełczak, and Josiah Hester. 2020. Reliable timekeeping for intermittent computing. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'20)*. 53–67.

[7] Bradley Denby and Brandon Lucia. 2020. Orbital edge computing: nanosatellite constellations as a new class of computer system. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'20)*. 939–954.

[8] Harsh Desai and Brandon Lucia. 2020. A power-aware heterogeneous architecture scaling model for energy-harvesting computers. *IEEE Computer Architecture Letters* 19, 1 (2020), 68–71.

[9] H. Desai and B. Lucia. 2020. A Power-Aware heterogeneous architecture scaling model for energy-harvesting computers. *IEEE Computer Architecture Letters* 19, 1 (2020), 68–71.

[10] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi. 2012. NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* 31 (2012), 994–1007.

[11] Mingyuan Gao, Ping Wang, Yifeng Wang, and Lingkan Yao. 2018. Self-Powered ZigBee wireless sensor nodes for railway condition monitoring. *IEEE Transactions on Intelligent Transportation Systems (TITS)* 19, 3 (2018), 900–909.

[12] Graham Gobieski, Brandon Lucia, and Nathan Beckmann. 2019. Intelligence beyond the edge: inference on intermittent embedded systems. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'19)*. 199–213.

[13] Brandon Lucia Graham Gobieski, Nathan Beckmann. 2018. Intermittent deep neural network inference. In *SysML Conference*. 1–3.

[14] Vikram Iyer, Maruchi Kim, Shirley Xue, Anran Wang, and Shyamnath Gollakota. 2020. Airdropping sensor networks from drones and insects. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom'20)*. 14 pages.

[15] Vikram Iyer, Rajalakshmi Nandakumar, Anran Wang, Sawyer B. Fuller, and Shyamnath Gollakota. 2019. Living IoT: A flying wireless platform on live insects. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom'19)*. Article 5, 15 pages.

[16] Chih-Kai Kang, Hashan Roshantha Mendis, Chun-Han Lin, Ming-Syan Chen, and Pi-Cheng Hsiu. 2020. Everything leaves footprints: hardware accelerated intermittent deep inference. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* 39, 11 (2020), 3479–3491.

[17] S. Kudva, S. Chaubey, and R. Harjani. 2014. High power-density, hybrid inductive/capacitive converter with area reuse for multi-domain DVS. In *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*. 1–4.

[18] H. Le, S. R. Sanders, and E. Alon. 2011. Design techniques for fully integrated switched-capacitor DC-DC converters. *IEEE Journal of Solid-State Circuits (JSSC)* 46, 9 (2011), 2120–2131.

[19] H. Lv, X. Xu, P. Yuan, D. Dong, T. Gong, J. Liu, Z. Yu, P. Huang, K. Zhang, C. Huo, C. Chen, Y. Xie, Q. Luo, S. Long, Q. Liu, J. Kang, D. Yang, S. Yin, S. Chiu, and M. Liu. 2017. BEOL based RRAM with one extra-mask for low cost, highly reliable embedded application in 28 nm node and beyond. In *2017 IEEE International Electron Devices Meeting (IEDM'17)*. 2.4.1–2.4.4.

[20] K. Ma, J. Li, X. Li, Y. Liu, Y. Xie, M. Kandemir, J. Sampson, and V. Narayanan. 2018. IAA: Incidental approximate architectures for extremely energy-constrained energy harvesting scenarios using IoT nonvolatile processors. *IEEE Micro'18* 38, 4 (2018), 11–19.

[21] K. Ma, X. Li, J. Li, Y. Liu, Y. Xie, J. Sampson, M. T. Kandemir, and V. Narayanan. 2017. Incidental computing on IoT nonvolatile processors. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'17)*. 204–218.

[22] K. Ma, X. Li, Y. Liu, J. Sampson, Y. Xie, and V. Narayanan. 2015. Dynamic machine learning based matching of nonvolatile processor microarchitecture to harvested energy profile. In *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD'15)*. 670–675.

[23] K. Ma, X. Li, S. R. Srinivasa, Y. Liu, J. Sampson, Y. Xie, and V. Narayanan. 2017. Spendthrift: Machine learning based resource and frequency scaling for ambient energy harvesting nonvolatile processors. In *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC'17)*. 678–683.

[24] K. Ma, Y. Zheng, S. Li, K. Swaminathan, X. Li, Y. Liu, J. Sampson, Y. Xie, and V. Narayanan. 2015. Architecture exploration for ambient energy harvesting nonvolatile processors. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA'15)*. 526–537.

[25] Kiwan Maeng, Alexei Colin, and Brandon Lucia. 2017. Alpaca: Intermittent execution without checkpoints. *Proc. ACM Program. Lang. (OOPSLA'17)* 1, Article 96 (2017), 30 pages.

[26] Kiwan Maeng and Brandon Lucia. 2020. Adaptive low-overhead scheduling for periodic and reactive intermittent execution. In *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'20)*. 1005–1021.

[27] Amjad Yousef Majid, Carlo Delle Donne, Kiwan Maeng, Alexei Colin, Kasim Sinan Yildirim, Brandon Lucia, and Przemysław Pawełczak. 2020. Dynamic task-based intermittent execution for energy-harvesting devices. *ACM Trans. Sen. Netw. (TSN)* 16, 1, Article 5 (2020), 24 pages.

[28] M. Mangrulkar and S. G. Akojwar. 2016. A simple and efficient solar energy harvesting for wireless sensor node. In *2016 Second International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. 95–99.

[29] Andrew S. Holmes Mayue Shi, Eric M. Yeatman. 2019. Energy harvesting piezoelectric wind speed sensor. *Journal of Physics: Conference Series* 1407 (Nov 2019), 012–044.

[30] Hashan Roshantha Mendis and Pi-Cheng Hsiu. 2019. Accumulative display updating for intermittent systems. *ACM Trans. on Embedded Computing Systems (TECS)* 18, 5s (2019), 72:1–22.

[31] A. M. Mohey, S. A. Ibrahim, I. M. Hafez, and H. Kim. 2019. Design optimization for low-power reconfigurable switched-capacitor DC-DC voltage converter. *IEEE Transactions on Circuits and Systems I: Regular Papers (TCAS-I)* (2019), 4079–4092.

[32] K. Qiu, N. Jao, M. Zhao, C. S. Mishra, G. Gudukbay, S. Jose, J. Sampson, M. T. Kandemir, and V. Narayanan. 2020. ResiRCA: A resilient energy harvesting ReRAM crossbar-based accelerator for intelligent embedded processors. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA'20)*. 315–327.

[33] J. Sakly, A. Bennani-Ben Abdelghani, I. Slama-Belkhodja, and H. Sammoud. 2017. Reconfigurable DC/DC converter for efficiency and reliability optimization. *IEEE Journal of Emerging and Selected Topics in Power Electronics* (2017), 1216–1224.

[34] M. D. Seeman, V. W. Ng, H. Le, M. John, E. Alon, and S. R. Sanders. 2010. A comparative analysis of Switched-Capacitor and inductor-based DC-DC conversion technologies. In *IEEE 12th Workshop on Control and Modeling for Power Electronics (COMPEL'10)*. 1–7.

[35] 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. In *Proceedings of the 43rd International Symposium on Computer Architecture (ISCA'16)*. 14–26.

[36] Ryo Shigeta, Tatsuya Sasaki, Duong Minh Quan, Yoshihiro Kawahara, Rushi J. Vyas, Manos M. Tentzeris, and Tohru Asami. 2013. Ambient RF energy harvesting sensor device with capacitor-leakage-aware duty cycle control. *IEEE Sensors Journal* 13, 8 (2013), 2973–2983.

[37] F. Su, W. Chen, L. Xia, C. Lo, T. Tang, Z. Wang, K. Hsu, M. Cheng, J. Li, Y. Xie, Y. Wang, M. Chang, H. Yang, and Y. Liu. 2017. A 462GOPs/J RRAM-based nonvolatile intelligent processor for energy harvesting IoE system featuring nonvolatile logics and processing-in-memory. In *2017 Symposium on VLSI Circuits*. C260–C261.

[38] S. Sudevalayam and P. Kulkarni. 2011. Energy harvesting sensor nodes: survey and implications. *IEEE Communications Surveys Tutorials* 13 (2011), 443–461.

[39] X. Sun, S. Yin, X. Peng, R. Liu, J. Seo, and S. Yu. 2018. XNOR-RRAM: A scalable and parallel resistive synaptic architecture for binary neural networks. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE'18)*. 1423–1428.

[40] Y. Sun, Z. Yuan, Y. Liu, X. Li, Y. Wang, Q. Wei, Y. Wang, V. Narayanan, and H. Yang. 2017. Maximum energy efficiency tracking circuits for converter-less energy harvesting sensor nodes. *IEEE Transactions on Circuits and Systems II: Express Briefs (TCAS-II)* 64, 6 (2017), 670–674.

[41] Andreas Tobola, Heike Leutheuser, Markus Pollak, Peter Spies, Christian Hofmann, Christian Weigand, Bjoern M. Eskofier, and Georg Fischer. 2018. Self-powered multiparameter health sensor. *IEEE journal of biomedical and health informatics* 22, 1 (2018), 15–22.

[42] C. Wang, N. Chang, Y. Kim, S. Park, Y. Liu, H. G. Lee, R. Luo, and H. Yang. 2014. Storage-less and converter-less maximum power point tracking of photovoltaic cells for a nonvolatile microprocessor. In *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC'14)*. 379–384.

[43] Wei Zhao and Yu Cao. 2006. New generation of predictive technology model for sub-45nm design exploration. In *Proceedings of the 7th International Symposium on Quality Electronic Design (ISQED'06)*. 585–590.

[44] C. Xia, J. Zhao, H. Cui, and X. Feng. 2018. Characterizing DNN models for edge-cloud computing. In *2018 IEEE International Symposium on Workload Characterization (IISWC'18)*. 82–83.

[45] L. Xia, T. Tang, W. Huangfu, M. Cheng, X. Yin, B. Li, Y. Wang, and H. Yang. 2016. Switched by input: Power efficient structure for RRAM-based convolutional neural network. In *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC'16)*. 1–6.

Received April 2021; revised June 2021; accepted July 2021