A Tale of Two Domains: Exploring Efficient Architecture Design for Truly Autonomous Things

Xiaofeng Hou †
Shanghai Jiao Tong University
Shanghai, China
hou-xf@cs.sjtu.edu.cn

Tongqiao Xu †
Shanghai Jiao Tong University
Shanghai, China
tqxu23@sjtu.edu.cn

Chao Li
Shanghai Jiao Tong University
Shanghai, China
lichao@cs.sjtu.edu.cn

Cheng Xu
Shanghai Jiao Tong University
Shanghai, China
jerryxu@sjtu.edu.cn

Jiacheng Liu
CUHK
Hong Kong, China
jcliu@cse.cuhk.edu.hk

Yang Hu
Tsinghua University
Beijing, China
hu_yang@tsinghua.edu.cn

Jieru Zhao
Shanghai Jiao Tong University
Shanghai, China
zhao-jieru@sjtu.edu.cn

Jingwen Leng
Shanghai Jiao Tong University
Shanghai, China
leng-jw@sjtu.edu.cn

Kwang-Ting Cheng
HKUST
Hong Kong, China
timcheng@ust.hk

Minyi Guo
Shanghai Jiao Tong University
Shanghai, China
guo-my@cs.sjtu.edu.cn

***Abstract*—Autonomous Things (AuT) refers to a collection of self-sufficient tiny devices capable of performing intelligent computations. Looking ahead, AuT promises to enable ubiquitous deployment of intelligence on many emerging consumer electronics and mission-critical infrastructures. Nevertheless, there is an important research gap to date: architecting efficient AuT systems requires both *energy autonomy* (EA) and *inference autonomy* (IA). In other words, practical AuT application scenarios necessitate tailored architectures with significantly expanded inference performance and more efficient use of energy.**

We present CHRYSALIS, a novel automated EA/IA co-design methodology for autonomous things. It aims to guide the transition from a traditional EA-only and IA-only design approach to a truly AuT-oriented architecture design. To fully understand the interrelationship between the EA domain and the IA domain, CHRYSALIS first introduces an architectural modeling framework encompassing every key AuT module involving energy harvesting, intermittent execution, and accelerator control. Based on the holistic system model, we design an intelligent architecture generation tool that can help find the ideal design for targeted AuT scenarios adhering to different SWaP (Size, Weight and Power) constraints. To validate our work, we use CHRYSALIS for fast construction and exploration of efficient AuT design and pre-RTL design in representative AuT scenarios. Extensive evaluation shows that CHRYSALIS outperforms state-of-the-art designs and our proposed technique shows 56.4% better performance on average. We believe that the methodology and tools developed in this paper will foster the development of more performant and practical architectures in the upcoming AuT era.

***Index Terms*—Edge Artificial Intelligence, Autonomous Embedded Systems, Intermittent Computing, Accelerator, Deep Learning**

I. Introduction

Autonomous things (AuT), or Internet of Autonomous Things (IoAT), is emerging as a catalyst for a transformative edge computing paradigm [16], [22], [61], [63]. Some types of AuT, including wearable devices and specialized sensors, could empower users with intelligent computing services in a highly sustainable and efficient manner in the wild [23], [72]. In certain circumstances, such as data analytics during a worldwide pandemic or continuous volcano hazards monitoring, AuT would perform tasks much more economically and safely than human labor [6], [51]. Today, many countries on our planet are actively accelerating their development and deployment of various autonomous devices for smart control of traffic, street lights, transportation of goods and waste disposal [16]. Looking ahead, spaceflight and deep space exploration require more powerful space-based IoAT for computation autonomy.

A true AuT system must achieve two unique goals: *energy autonomy* (EA) and *inference autonomy* (IA). The former means battery-free and self-powered, allowing the system to get rid of bulky energy storage devices; and the latter refers to the ability to perform in-situ AI computation, thus eliminating the dependency on additional remote computing engine. Ideally, AuT systems can diligently accumulate energy capacity from ambient energy resources to support scenario-aware edge AI applications. Due to power variability, the actual AuT systems have to frequently resort to intermittent AI computation, the performance and efficiency of which heavily rely on both energy capacity and inference capability.

Currently, early-stage autonomous systems still suffer from poor performance and low efficiency. They rely on traditional energy-harvesting based IoT devices (EH-IoT) which are governed by microcontroller units (MCUs) such as MSP430x [9], [30], [31], [50]. These devices have severely constrained computational capabilities rendering them inadequate for running computationally intensive AI inference tasks [24]. Oftentimes,

†These two authors contribute equally to the work.

EH-IoT devices choose to offload large amounts of data to external servers. It results in increased power consumption and operational expenditure due to region-wide communication.

Meanwhile, although extensive work has been done on edge AI accelerators and micro datacenters in recent years [8], [15], [38], their superior computational capabilities are limited to stable power environments [8], [38]. In scenarios with depleted energy capacity, they frequently encounter power exceptions, leading to reduced efficiency or unavailability [49]. Equipping existing high-performance edge AI accelerators with a large energy subsystem, such as oversized solar panels and capacitors, would enhance the inference speed of DNNs. Doing so unavoidably introduces longer charging latency and increased energy leakage [50], [59], ultimately reducing the efficiency and practicality of AuTs. On the other hand, simply reducing the size of accelerators to match the capacity of ambient energy is an alternative approach. However, it sacrifices inference accuracy, computing performance, and data reuse opportunities [9], [31]. Additionally, it may introduce more overhead in saving the inference checkpoints.

In this paper, we argue that the co-design of energy autonomy and inference autonomy is non-trivial, and there is an important research gap at the architecture level. It is imperative to develop a comprehensive methodology that accurately describes the interrelationship between the energy subsystem and the inference subsystem as well as their implications on the performance and efficiency of intermittent inference.

Specifically, the architecture design for an AuT system encompasses numerous hardware and software modules of both the energy subsystem and inference subsystem, with each module offering a large number of configurations. Iterating over the configurations of these components yields an explosion of possible combinations, ranging from tens of thousands to billions. Furthermore, many AuT systems are part of mission-critical infrastructures in land, sea, air, and space. Each of the AuT faces rigorous and specific Space, Weight, and Power (SWaP) constraints that span the entire computing stack [48]. These constraints further complicate the design process. Previous EH-IoT research has dedicated effort to designing the parameters of various DNN inference components to suit specific tasks [2], [9], [33], [50], [59]. However, conventional design space exploration methods [37], [42], [53], [70] fall short of meeting the general requirements of AuT design. They neither consider intermittent inference scenarios nor account for the exponential growth of architecture candidates resulting from the co-design of energy autonomy.

We introduce CHRYSALIS, a set of new techniques designed to generate the ideal AuT architectures that meet specific SWaP constraints. We first propose a comprehensive modeling approach that captures the interrelationship between the energy subsystems and the inference subsystems. Our AuT-oriented approach evaluates different modules for both energy autonomy and inference autonomy, assessing their impact on energy consumption and computing performance during intermittent inference. Leveraging this model, we further develop an automated AuT architecture generation framework that

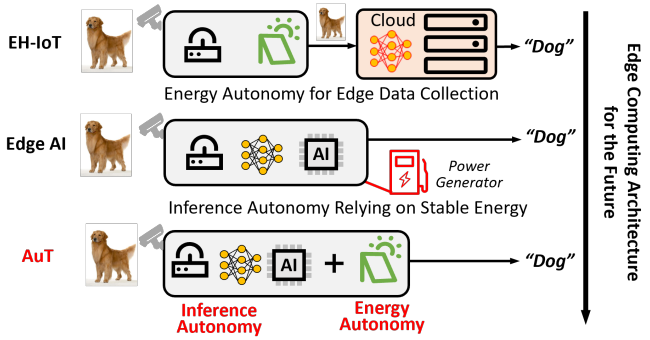


Figure 1: An ideal AuT harnesses the synergy of EH-IoT and Edge AI to enable full autonomy in both computing capability and power supply.

incorporates various hardware and software components, along with configurable parameters within the energy subsystem and inference subsystems. Our framework employs a bi-level search strategy to efficiently explore alternative architectures and identify the most suitable solution for scenario customization. The significance of CHRYSALIS lies in its ability to speedup the transition from traditional EA-oriented and IA-oriented design to a new class of AuT-oriented holistic design. Moreover, it serves as a valuable tool for exploring and guiding the design of more powerful AuT devices suitable for versatile edge applications. Overall, we make the following contributions:

1) We analyze the characteristics of AuT and identify the limitations of existing architectures. We propose a novel methodology for the comprehensive modeling of AuT systems. It can effectively capture the dependencies among different AuT subsystems and assess the impact of various combinations on AuT intermittent inference.
2) We develop an automated AuT architecture generation tool that enables the evaluation of AuT system performance under different architecture alternatives. This tool efficiently generates ideal architectural solutions tailored to specific domain-specific tasks in an automated way.
3) We showcase the utility of CHRYSALIS and demonstrate its capability to enhance the performance of existing AuT devices while designing customized architecture for more AuT applications. Our experimental results reveal that the architectures obtained through CHRYSALIS exhibit an average performance improvement of 56.4%.

The remainder of this paper is organized as follows: Section II presents the background and motivation. Next, Section III describes the design of CHRYSALIS. Then, Section IV and Section V show the experimental methodology and result. After that, Section VI summarizes the related work. Finally, Section VII concludes the paper.

II. Background and Motivation

The autonomous things (AuT) features distinctive operation mode, as shown in Figure 1. EH-IoT puts an emphasis on utilizing ambient energy for achieving energy autonomy in some simple data collection tasks. Edge AI ensures in-situ inference autonomy by integrating high-performance DNN accelerators. Differently, AuT is expect to exhibit a synergism

Table I: Investigation into the existing AuT platforms.

AuT Design Methodology	Subsystem Design		Properties of Adaptability	
	Energy	*Inference*	*Scalability*	*Sustainability*
WISPCam [50], Botoks [13]	✓	×	×	×
SONIC [24], RAD [30]	×	✓	×	×
HAWAII [35], Stateful [71]	×	✓	×	×
Protean [2]	✓	×	×	✓
CHRYSALIS (**Ours**)	✓	✓	✓	✓

in which the system can yield a design effectiveness far better than that can be offered by either EH-IoT or Edge AI.

A. The Two Domains of an AuT System

An AuT system consists of two key domains: an energy subsystem and an inference subsystem. They work together to achieve energy autonomy (EA) and inference autonomy (IA). The inference subsystem allows edge devices to process data and perform intelligent inferences locally, without relying on the costly backend infrastructure. Meanwhile, energy autonomy allows AuT to be battery-free and self-powered, eliminating the need for inconvenient and bulky batteries.

Currently, inference autonomy is primarily achieved through MCUs for runtime management [36], [49], [59] and low-power accelerators for domain-specific DNN inference tasks [24], [30], [35]. This forms a non-trivial edge computing stack, including both inference control logic and acceleration logic, to meet the performance requirements of the application. For example, unmanned aerial vehicle (UAV) systems deploy specialized AI chips or system-on-chips (SOCs) to run DNN models for object detection [40]. Previous works have focused on co-designing DNN models and accelerators to support DNN computation and acceleration at the edge [2], [24], [30].

Typically, energy autonomy is achieved by utilizing renewable energy through energy harvesting (EH) technologies [43], [44]. The energy subsystem comprises EH devices such as solar panels or coils to receive energy from the ambient environment. The collected energy is buffered in energy storage components such as capacitors [1], [46], [48], [56]. When sufficient energy is available, the computing subsystem is activated to perform tasks. An energy manager chip controls the entire process, and a runtime control unit (e.g., MCU) executes the runtime layer. Note that the power supply in EH-based systems can be intermittent and low, depending on environmental conditions. Since DNN execution is lengthy and energy-consuming, power interruptions can occur frequently. Some research focuses on developing efficient analog power circuits [57], [67] or co-designing applications to reduce the overhead caused by energy-related interruptions [12], [29], [34].

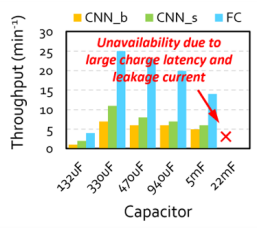
B. Limitations of Existing Design Approach

Although multiple works have made efforts to build power- and task-aware AI systems, they still face challenges in terms of performance and efficiency due to a lack of co-design of the energy and inference subsystems. As summarized in Table I, our investigation into existing research has identified two major limitations that contribute to this research gap.

First, an AuT scalability issue often arises due to the inefficient architecture design for the AuT inference domain [24], [35], [36], [49], [59]. Most AuT systems are implemented

Inference HW	MSP430	Eyeriss V1
Test Model	MINIST-CNN	AlexNet
Input	28*28	4*227*227
Time (ms/input)	1447	115.3
MOPs	**1.608**	**2663**
Power (mW)	**7.5**	**278**
Energy (uJ)	**10.85**	**32.05**

(a)



(b)

Figure 2: (a) Comparison between the current intermittent inference system (HAWAII [35]) and the popular AI accelerator (Eyeriss V1 [8]), in the condition of non-intermittent cases. (b) The cases for HAWAII, with different capacitor sizes and three applications.

using classic intermittent computing platforms [24], [34], [35], [49], which typically consist of MCUs (such as the MSP430x series of cores [66]) with a low-energy accelerator (LEA) for DNN execution, limited SRAM for shared memory, and FRAM as non-volatile memory (NVM). This architecture could become limiting factors for DNN inference due to their small memory size (only 8KB SRAM and 256KB FRAM) and poor computation performance. In addition, previous research has mainly focused on optimizing individual components of AuT systems [35], [49], [71], such as computational units and AI accelerators [8], [53], [70]. However, many of these studies emphasize increasing millions of operations per second (MOPs) without considering the energy harvesting (EH) issue. The resulted designs often demand excessive power, making them unsuitable for low-power AuT applications. For instance, as depicted in Figure 2(a), Eyeriss V1 exhibits significantly higher energy consumption compared to an MCU [8], [66], rendering it impractical for energy harvesting scenarios.

Second, an AuT sustainability issue often arises as a result of ad hoc solutions for the AuT energy domain [50], [59]. Existing EH-based power subsystems are often designed based on empirical knowledge without systematic exploration. This can lead to system inefficiency or even unavailability. For example, the significance of capacitor size design within existing EH-based AI inference systems, exemplified by the HAWAII framework [35], is illustrated in Figure 2(b). Larger capacitor sizes enable longer energy cycles but at the expense of decreased throughput and leakage current. Similarly, larger energy harvesters provide more energy budget but are constrained by specific application scenarios. While some dynamic strategies have been proposed to adjust capacitor size or cutoff voltage using dedicated circuits [12], very limited attention has been given to the quantitative analysis of critical hardware components, which should be tailored to the specific application and deployment scenarios.

To overcome the above limitations, prioritizing full-system EA/IA co-design is imperative. We need to identify the ideal combination of hardware accelerators, mapping strategies, and energy harvesters.

C. Challenges of AuT Design

Given that the existing approaches have limited the performance of AuT, we take the first step to explore truly scalable

and sustainable AuT design. In this section, we outline two key obstacles that need to be addressed for effective AuT design.

The first challenge lies in accurately describing a holistic AuT system that encompasses both the energy subsystem and the inference subsystem, while also considering their implications on the overall system performance. Traditional design methodologies often focus on individual subsystems in isolation, neglecting the intricate interplay between EA and IA. To overcome this challenge, a comprehensive methodology is required that captures the interactions and dependencies between these subsystems. It should provide a unified framework to analyze and optimize the AuT system as a whole, considering factors such as energy availability, intermittent operation, and the impact of energy constraints on inference performance. Developing such a methodology is crucial for achieving efficient and effective AuT designs.

The second challenge involves determining the ideal AuT architecture from the vast number of potential candidates, considering the specific SWaP constraints imposed by diverse AuT application scenarios. Different AuT systems operate in various scenarios, such as land, sea, air, and space, each with its unique requirements and constraints. These constraints extend beyond the computing stacks and encompass factors like size, weight, energy availability, and environmental conditions. Designing an AuT architecture that meets these constraints while maximizing performance and efficiency is a complex task. Conventional design space exploration methods often fall short in addressing the exponential growth of architecture candidates resulting from the co-design of energy autonomy. Thus, novel techniques are needed to efficiently navigate the design space, considering both the energy and inference subsystems, to identify ideal configurations that align with the specific SWaP constraints of versatile AuT application scenarios.

To tackle the above challenges, it is quintessential to develop a comprehensive and automated tool that can effectively explore and optimize the design of AuT. This requires interdisciplinary research efforts that integrate knowledge from architecture design, energy systems, machine learning, and application domains. By tackling these obstacles, we can unlock the potential of high-performance AuT systems that effectively balance energy autonomy and inference autonomy to meet the demands of diverse real-world applications.

III. The CHRYSALIS Design

A. Overview

This paper introduces CHRYSALIS, a comprehensive framework that seeks a synergism of the energy autonomy domain and the inference autonomy domain in AuT design. The proposed methodology takes a holistic perspective, considering multiple aspects of different subsystems in an automated way. The usage model of CHRYSALIS is as follows: Given a domain-specific DNN model along with its corresponding dataset, the high-level specifications of the AuT (including environment and technology constraints) as well as specific objective demands, the tool can automatically generate the ideal AuT solution that encompasses the configurations of energy harvester hardware (EH HW), inference hardware (Infer HW), and the dataflow of the workload. The generated solution is tailored specifically to the provided inputs, resulting in a customized and efficient AuT architecture design. To provide a clear summary of the usage model, we have compiled the inputs and outputs of CHRYSALIS in Table II. This table outlines the essential parameters and notations used in the usage model, facilitating a better understanding of the design process facilitated by CHRYSALIS.

Figure 3 provides an overview of CHRYSALIS. By default, CHRYSALIS needs the following inputs: *(1) DNN models* which include the specific DNN tasks that need to be performed by the AuT, along with the corresponding datasets required for inference. *(2) Platform constraints* which specify the hardware architecture to be used for the AuT design and encompasses the available ranges of the design space, such as the size of capacitors, the size of the energy harvester (e.g. the solar panel), and other relevant factors that impact the system's operation. *(3) Objectives* which define the optimization targets and constraints for the AuT. It could include parameters such as latency, system size, and combinations of multiple indicators that need to be optimized while considering the constraints imposed by the system's requirements and limitations. *(4) Dataflow strategies* which specify the tiles' size and dataflow taxonomy including weight stationary (WS), output stationary (OS) and input stationary (IS) of the used AI accelerators.

Given these inputs, it consists of several components that work together to generate the ideal AuT architecture, considering different Size, Weight, and Power (SWaP) requirements. These components include:

1) *The AuT HW and SW Describer*: This component describes the multifaceted components of both the energy subsystem and inference subsystem in AuT systems. It encompasses the hardware and software aspects, capturing the intricacies of the system's architecture.
2) *The CHRYSALIS Evaluator*: This component assesses the performance and energy consumption of the AuT under different architectural designs. It enables the comparison and evaluation of various configurations, providing insights into their effectiveness and efficiency.
3) *The CHRYSALIS Explorer*: This component is responsible for generating the ideal AuT architecture. It leverages the information from the AuT HW and SW Describer and the evaluation results from the CHRYSALIS Evaluator to explore the design space and identify the configurations that meet the specified SWaP requirements.

These components collectively form a powerful framework that automates the development process of high-performance and energy-efficient AuT systems. By considering the holistic perspective of the system and leveraging automated techniques, CHRYSALIS enables the generation of ideal solutions, reducing the time and effort required for manual optimization. Importantly, when combined with state-of-the-art ubiquitous Internet of Things (IoT) system optimization techniques such as XiUOS [5], CHRYSALIS will provide researchers and

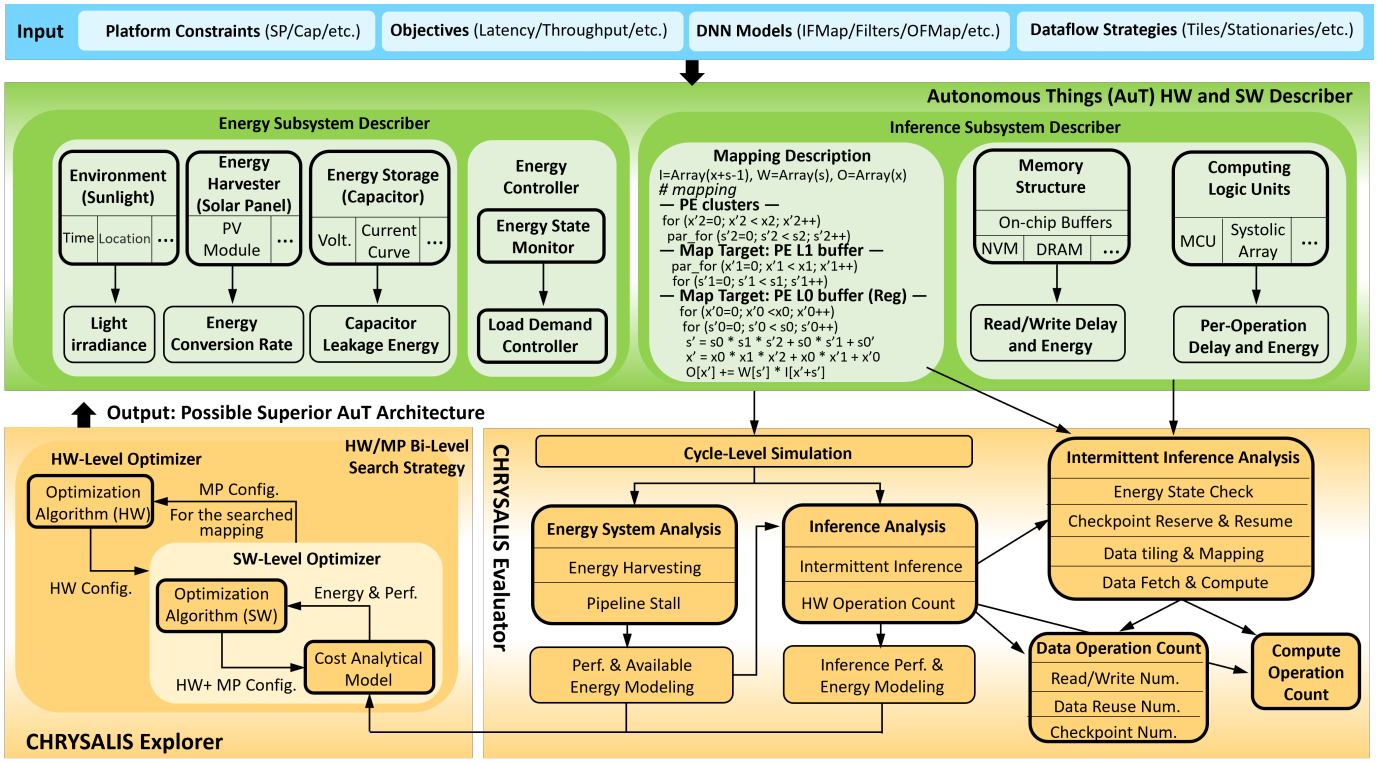


Figure 3: Overview of CHRYSALIS: Given the platform constraints, objectives and domain-specific DNN model along with its dataflow strategy, the tool can automatically generate the ideal AuT architecture.

Table II: Usage model and associated parameter notations for AuT modeling in CHRYSALIS.

Category	System	Param	Introduction
Input	Environment Constraint	k_{eh}	Environmental light coefficient
	Technology Constraint	k_{cap}	Leakage current coefficient
		U_{on} U_{off}	Threshold voltage for the system state
		e_{r} e_{w}	Energy cost of r/w each byte from NVM
		p_{mem}	Static power of each byte of memory
	Objective	π	Domain-specific objective demand function
	Workload	—	Domain-specific DNN task and its dataset
Variable	Evaluation	r_{exc}	Energy exception rate of the inference
	Dataflow	E_{df} T_{df}	Whole energy and latency of inference with 1 PE
		N_{data}	Inference data size
		N_{ckpt}	Checkpoint data size
Output	EH HW	C	Capacitor size
		A_{eh}	The size of solar panel
	Infer HW	N_{tile}	Tile number of the layer
		N_{mem}	VM memory size per PE
		N_{PE}	PE number
	Dataflow	—	Preferable dataflow of DNN task

designers with a valuable tool for developing AuT systems.

B. Modeling Comprehensive AuT Components

We first describe the holistic AuT model including the energy subsystem and inference subsystem which will be included in the evaluator and explain the complexity of the whole system. This modeling approach takes into account various variables that are essential for considering the existing relevant designs [8], [12], [49], [50]. The associated parameters and notations for AuT modeling are shown in Table II.

1) Energy Subsystem: The EH-based energy subsystem consists of an energy harvester, a small capacitor, and a management IC for control and voltage conversion [65], [67]. In CHRYSALIS, we use solar panels as examples of energy harvesters. The harvested energy is stored in the capacitor and then regulated to feed the computing subsystem. The sunlight environment model is based on existing work, and the input energy from solar panels is determined by their size and the intensity of sunlight. Specifically, assuming the solar panel size is A_{eh}, the power input P_{eh} can be estimated as,

$$P_{\text{eh}} = A_{\text{eh}} \times k_{\text{eh}} \tag{1}$$

where k_{eh} is a coefficient that reflects the complex attributes of photovoltaic modules and can be obtained using existing EH modeling tools [27].

For the capacitor, we store the collected energy at each step and calculate the overhead of the leakage current. The leakage current can be represented as,

$$I_{\text{R}} = k_{\text{cap}} C U \tag{2}$$

where C is the capacitor size, and U is the rated voltage. Within the search space, k_{cap} and U are typically constants. Larger capacitor size often results in a higher leakage current.

Therefore, the available energy during one energy cycle can be calculated by adding the stored energy and the harvested energy during the execution,

$$\begin{aligned} E_{\text{available}} &= E_{\text{store}} + E_{\text{eh}} - E_{\text{cap}} \\ &= \frac{1}{2} C (U_{\text{on}}^2 - U_{\text{off}}^2) + T(k_{\text{eh}} A_{\text{eh}} - k_{\text{cap}} C U_{\text{on}}^2) \end{aligned} \tag{3}$$

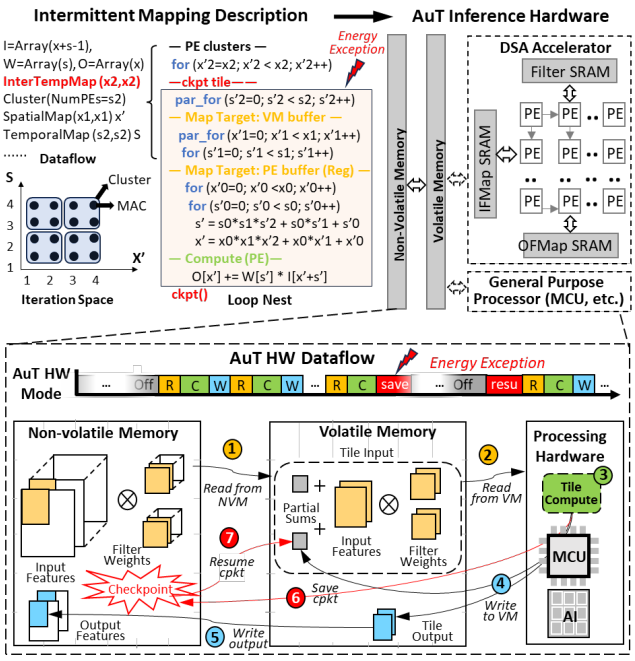


Figure 4: Intermittent inference process modeling in AuT.

where U_{on} and U_{off} are the threshold voltages of the system, and T is the execution time. The leakage energy is simplified as the voltage is unchanged.

2) Inference Subsystem: Existing literature has offered valuable insights into modeling traditional intermittent computing processes and AI inference, providing a foundation for understanding the workflow of AuT systems. However, it is important to note that the intermittent inference process of AuT introduces unique challenges that have not been adequately addressed in previous modeling approaches. In light of these considerations, we present a comprehensive redefinition of the inference system for AuT in our work. By taking into account the specific requirements and characteristics of AuT, our modeling approach aims to provide a more accurate and effective representation of the inference system in this context.

Figure 4 illustrates the inference process modeling of AuT. As shown in the figure, the intermittent mapping description pertains to the mapping process that transforms the structure of the DNN into a format suitable for execution on the AuT inference hardware. We have improved the iteration space dataflow model based on the data-centric mapping directives [42] by introducing intermittent inference adaptation (InterTempMap) and generating mapping schemes for DNN layers in intermittent scenarios. As is shown in the mapping description, the dataflow is firstly described in terms of data-centric mapping directives, based on intermittent, temporal and spatial mapping. Temporal mapping (TemporalMap) means the inference is partitioned to execute in the temporal dimension, namely they will execute one after another in the same components. Spatial mapping (SpatialMap) means to be partitioned in the spatial dimension so that they will execute in different hardware (PEs). Finally, intermittent mapping (InterTempMap) is an incremental description. It means to be partitioned into two energy cycles for power interruptions between tiles. The directives are used to describe the corresponding dataflow, as shown in the loop nest in the figure. In the Loop Nest, the checkpoint (cpkt) tile corresponds to the InterTempMap directive, enabling larger layers to be divided into multiple tiles for execution in intermittent inference scenarios.

The AuT inference hardware consists of an accelerator and a general-purpose processor, both equipped with their respective caches. Additionally, these components have access to Non-Volatile Memory (NVM) and Volatile Memory (VM). The hardware configuration imposes constraints on the dataflow, which directly affects system performance. To accurately model the hardware dataflow of intermittent inference behavior, we need to consider the following processes: read (R), compute (C), write (W), save checkpoint (save), and resume checkpoint (resu). The AuT hardware dataflow can be described as follows: Initially, all data resides in Non-Volatile Memory (NVM) and is divided into multiple tiles. During the computing process, a data tile is ❶ read from NVM into Volatile Memory (VM) and awaits computation. The processing hardware ❷ fetches a portion of the current tile's data from VM and ❸ computes a partial sum. The computed partial sum ❹ is then written back to VM. If there is remaining data within the current tile that has not been computed, the process continues to fetch and compute the remaining data until the entire tile has been fully computed. Finally, the computed result of the current tile ❺ is written back to NVM. If there is sufficient energy, the aforementioned process continues tile by tile. However, if there is insufficient energy to proceed with the computation, the current state, including all data in VM and the processing hardware, ❻ is saved as a checkpoint in NVM. Once the energy supply is replenished, the checkpoint ❼ is resumed from NVM, allowing the inference process to continue.

In the above process, the checkpoints result in additional overhead. Therefore, in quantitative modeling, the total energy cost E_{all} during the execution can be summarized as,

$$E_{\text{tile}} = E_{\text{read}} + E_{\text{infer}} + E_{\text{write}} + E_{\text{static}} \tag{4}$$

$$\begin{aligned} E_{\text{all}} = N_{\text{tile}} E_{\text{tile}} = E_{\text{df}} + T N_{\text{mem}} p_{\text{mem}} + N_{\text{data}} e_{\text{r}} \\ + N_{\text{tile}} (1 + r_{\text{exc}}) N_{\text{ckpt}} (e_{\text{r}} + e_{\text{w}}) \end{aligned} \tag{5}$$

where N_{tile} is the number of tiles divided from the layer, and E_{tile} is the energy cost of each tile. E_{df} is the cost of the calculations. N_{mem} and p_{mem} are the size of memory and the static power of each byte of memory respectively. N_{data} indicates the total required number of data for the entire inference and N_{ckpt} is the number of data contained within each checkpoint. The term r_{exc} refers to the energy exception rate for tile inference, which represents the probability of encountering an energy exception for each tile during the inference process. To simplify the model, we assume r_{exc} to be a static coefficient based on the specific scenario [59].

In the specific simulation process, variables such as E_{infer} are mapper and hardware-dependent. This distinction will be evaluated by utilizing different mapper and hardware architectures. In our implementation, we have implemented

two main accelerator simulations in CHRYSALIS and will be explained in the realization sections.

For the parameters, the inference time of the system can be estimated by the PE number of the architecture as,

$$T = \frac{T_{\text{df}}}{N_{\text{PE}}} \tag{6}$$

where T_{df} is the whole PE running time of the inference.

3) Holistic Model Analysis: The size of AuT and its performance are quite important for the holistic design. The size of AuT can typically be directly measured by the size of the energy harvesters, as they usually occupy the majority of the volume. The performance of the architecture can be assessed using the latency of a single inference in the model. In an AuT, the latency is mainly determined by the charging latency. Considering the energy harvester is harvesting despite the state of the architecture, the latency can be modeled as:

$$E2ELat = \frac{E_{\text{all}}}{P_{\text{eh}}} \tag{7}$$

Therefore, under specific objective conditions, improving the performance of the system can be achieved by increasing the size of the energy harvesting system according to Equation 1 or reducing the energy consumption during inference according to 5. The former represents the tradeoff relationship between size and performance, while the latter represents the optimization space introduced by architectural design.

According to Equation 5, we observe that two ways are possible to reduce the energy cost. Firstly, by designing a reasonable dataflow, we can potentially further reduce E_{infer}, data movement costs and E_{static} by reducing execution time. For a large number of existing neural network mappers, this optimization has already reached a high level of effectiveness. Secondly, by reducing N_{tile}, which represents the number of partitions in the neural network, E_{all} can be further decreased. However, N_{tile} can not be too small because every tile should be executed in one energy cycle:

$$E_{\text{tile}} \leq E_{\text{available}} \tag{8}$$

According to Equation 3 and 4, N_{tile} is constrained by:

$$N_{\text{tile}} \geq \frac{\alpha_3 + \alpha_4 \mathcal{N}_{\text{mem}}}{\alpha_1 \boldsymbol{C} + \mathrm{k}_{\text{eh}} \boldsymbol{A}_{\text{eh}} \frac{T_{\text{df}}}{\mathcal{N}_{\text{PE}}} - k_{\text{cap}} \boldsymbol{C} \frac{T_{\text{df}}}{\mathcal{N}_{\text{PE}}} - \alpha_2} \tag{9}$$

where α_{i}, k_{cap} and T_{df} are constants under the specific hardware technology and dataflow search. $\mathcal{N}_{\text{mem}}$ and $\mathcal{N}_{\text{PE}}$ are inference subsystem design parameters. $\boldsymbol{C}$ and $\boldsymbol{A}_{\text{eh}}$ are energy subsystem design parameters. k_{eh} is the coefficient related to the environment.

Based on the Equation 9, it can be observed that the size of the memory hierarchy, the energy storage capacitor, the scale of the energy harvester and the number of processing elements (PEs) (related to execution time) influence the size of N_{tile}, which further affects energy consumption as well as dataflow design. These design metrics play a crucial role in EH-IoT systems as they are directly related to system performance and energy efficiency. The variations in constants caused by external factors will also impact the design of AuT. For instance, in the case of low environmental energy, i.e., when the energy harvested (k_{eh}) is small, each layer of the network will be divided into a larger number of tiles. However, when the environmental energy is sufficiently high, the number of tiles can be reduced to conserve more energy.

In practical scenarios, this assessment becomes more complex as the optimization of dataflow is influenced by various metrics such as N_{tile}. Therefore, in evaluating the design, we employ a more rigorous step-based simulation and a bi-level search to ensure the accuracy of system exploration.

Finally, the model optimization can be summarized as,

$$\begin{aligned} Res &= \{AuT \| Objective, Workload, Constraint\} \\ &= \{AuT \| \pi(\mathbf{HW}_{\text{eh}}, \mathcal{HW}_{\text{infer}}, Df), Tech, \text{Envir})\} \end{aligned} \tag{10}$$

where π is the objective function. In the following section, we will introduce the specific CHRYSALIS implementation.

C. Generating Ideal AuT Solutions

Using the proposed modeling approach, CHRYSALIS can effectively assess the performance as well as the energy consumption of various AuT architecture candidates, and ultimately identify the ideal AuT architecture tailored to specific DNN tasks. The entire process is illustrated in Figure 3.

To begin with, CHRYSALIS utilizes the AuT HW and SW Describer to provide comprehensive descriptions of potential AuT candidates. The descriptors consist of the energy subsystem describer and the inference subsystem describer. The energy subsystem describer encompasses descriptions for the environment, energy harvester, and energy storage, along with an energy controller responsible for implementing the logic of the energy subsystem. The three descriptions are based on their respective realistic models, while the energy controller emulates the intermittent computing power logic and communicates with the inference subsystem describer. The inference subsystem describer presents the operational logic of the inference system, comprising mapping description and inference hardware description. The mapping description presents the execution strategy of the neural network, detailing the dataflow during runtime for each layer of the neural network. The inference hardware description characterizes the hardware attributes of the inference subsystem, including the memory structure and the computing logic units.

Subsequently, the CHRYSALIS Evaluator plays a crucial role in evaluating and comparing different AuT architecture candidates. It quantifies the number of data and compute operations required by the modeling approach, enabling a comprehensive analysis and comparison of various design choices in terms of performance and energy consumption. This evaluation process is performed using a step-based simulator that accurately models the intermittent inference processes in AuT systems. By simulating the intermittent inference under different architectural designs, the CHRYSALIS Evaluator provides valuable insights into the performance and energy characteristics of the system, which are then inputted into the CHRYSALIS Explorer for informed decision-making.

Ultimately, the CHRYSALIS Explorer leverages a bi-level search strategy to explore the ideal output. Initially, the HW-level optimizer generates a hardware configuration and employs

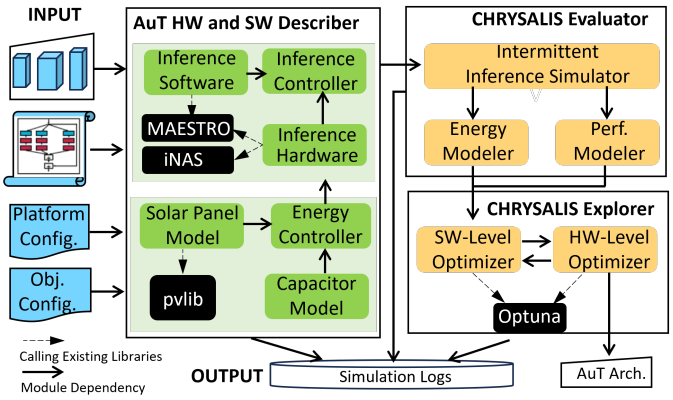


Figure 5: Implementation details of CHRYSALIS.

the current configuration as a starting point for the SW-level optimizer to conduct a mapping search. Once the SW-level optimizer iterates and generates the preferable result, the current preferable mapping configuration and objective metrics are returned to the hardware optimizer. Subsequently, considering this solution as better achievable by the current hardware, the HW-level optimizer proceeds to search for the next hardware configuration. Ultimately, the HW-level optimizer delivers the ideal hardware configuration along with its corresponding ideal mapping configuration based on the designated objective.

D. CHRYSALIS Implementation

To implement CHRYSALIS, we have developed a solar panel energy subsystem describer and two inference subsystems to accommodate both existing and future AuT setups. The energy subsystem describer incorporates a solar panel model based on `pvlib` [27] and a capacitor model [50]. We consider stable harvested energy during one inference while different light conditions in different inferences because sunlight does not undergo significant changes within a short time (<5 minutes) [27] and may change greatly in one day. The variable source during inference can be supported by component extensions for other energy harvesters. The energy subsystem can be accessed by the inference system through the energy controller interface. The first implementation of the inference subsystem describer targets the widely used MSP430FR5994 platform [66], which features a low-energy accelerator (LEA) capable of accelerating specific vector computation operations, as well as a low-power analog-to-digital converter (ADC) for energy measurement. For modeling purposes, we have adapted the existing energy and latency models [49] of the MSP430FR5994. The second implementation aims to evaluate reconfigurable accelerators, and it utilizes the data-centric representation from `MAESTRO` [42] as the dataflow description. However, intermittent inference differs from continuous inference in terms of how computation is partitioned between two energy cycles. Since power interruptions occur between the two computation blocks, all data needs to be retrieved again from non-volatile memory (NVM), which deviates from traditional spatial and temporal partitioning approaches. Consequently, we have realized the describer based on `MAESTRO` and the intermittent

Table III: Supported AuT component setups of CHRYSALIS.

Subsys.	Component	Realization	Base Model
EH	Energy Harvester	Solar Panel	`pvlib` [27]
	EH Controller	Power Management IC	BQ25570 [65]
	Capacitor	Electrolytic Capacitor	Physics Model
Infer	Infer Controller	Microcontroller Unit	MSP430 [66]
	Strategy	Tile Partition, ckpt.	`iNAS`-like [49]
	Accelerator & Mapper	Existing AuT Setup	*MSP430FR5994* [66] `iNAS` [49]
		Future AuT setup with Accelerators	`CHRYSALIS-MAESTRO` [42] `CHRYSALIS-GAMMA` [37]

Table IV: Parameters as design space for the fast construction and exploration of efficient AuT design.

Design Spaces		
Parameter Name	**Type**	**Potential Values**
Solar Panel Size	float	1cm² to 30cm²
Capacitor Size	int	1uF to 10mF
Tiling Size	list(int)	Factors of each dimensions

Applications				
Application	**Input**	**Layer**	**Parameters(k)**	**kFLOPs**
Simple Conv	(3,32,32)	1	1.2	13.8
CIFAR-10	(3,32,32)	7	77.5	9052.1
HAR	(9,128,128)	5	9.4	205.2
KWS	(250,1,1)	5	49.5	49.5

mapping description illustrated in Figure 4. Furthermore, we implement the CHRYSALIS Explorer based on the open-source library `Optuna` and utilize a genetic algorithm to generate potential architecture configurations.

To ensure the accuracy and real-time impact of energy fluctuations on the system, each component of CHRYSALIS is realized through a step-based approach. Unlike most existing accelerator and system simulators that rely on statistical methods and simply sum up the energy or time of individual components, our approach captures the dynamic nature of energy fluctuations and their real-time effects on system behavior. We achieve this by dividing the inference process into steps, each lasting several seconds (adjustable based on requirements), and tracking the changes in various metrics within each subsystem at every step. The evaluator initializes the process and iterates through the steps to drive the simulation. In each step, the evaluator invokes the energy controller, which monitors energy changes, and the inference controller, which tracks inference changes. These controllers, in turn, recursively call the corresponding simulators for each subsystem.

To ensure the scalability of CHRYSALIS, we have implemented each component of CHRYSALIS using an interface-oriented approach. This design allows for the substitution of any component within CHRYSALIS, enabling the evaluation of AuTs with different structures. By utilizing newer or more sophisticated simulators for hardware or inference models through an interface, users can explore a broader range of possibilities. Case studies serve as examples of substituting reasoning simulators, but the scope extends beyond these scenarios. For instance, by directly modifying and integrating relevant simulators, considerations such as temperature, input correlations, additional energy harvesting devices, and even user programming can be incorporated to explore specific scenarios.

Table V: Parameters as design space for AuT design with reconfigurable accelerators.

Design Spaces				
Parameter Name	**Type**	**Potential Values**		
Solar Panel Size	float	1cm² to 30cm²		
Capacitor Size	int	1uF to 10mF		
Architecture	union	TPU [11], Eyeriss [8]		
PE Number	int	1 to 168		
PE cache size	int	128bytes to 2KB		
Applications				
Application	**Input**	**Layer**	**Parameters(M)**	**GFLOPs**
BERT	(1,768)	5	56.6	1.28
Alexnet	(3,224,224)	7	58.7	1.13
VGG16	(3,224,224)	13	138.3	15.47
Resnet18	(3,224,224)	20	11.7	1.81

IV. Experimental Methodologies

Overall, we explored a series of AuT architectures using the established CHRYSALIS platform. The evaluation components deployed in CHRYSALIS are listed in Table III. Building upon the realized components in CHRYSALIS, we have the capability to perform coordinated searches for both the energy subsystem and inference subsystem, with support for two accelerators and mappers within the inference subsystem. Leveraging these two subsystems, we aim to fully show the search capabilities of CHRYSALIS and demonstrate its superiority over other search methods.

Existing AuT setup: In the first part of the experiment, we utilize the established evaluators for existing inference hardware and extensively searched the design space for EH and Mapping, showcasing the rapid prototyping capability of the CHRYSALIS system and demonstrating the rationale behind considering multiple aspects in AuT modeling. We utilize the commonly used low-energy accelerator as an inference accelerator and tile searcher for the inference mapper. The target platform [66] contains a low-energy accelerator (LEA) that can accelerate certain vector computation operations. Table IV clearly delineates our search space, which includes the size of the energy harvester and capacitor, as well as the tile size of the neural network. Specifically, the size of the energy harvester ranges from $1cm^2$ to $30cm^2$, while the capacitor's capacity extends from 1uF to 10mF. The tile size, an integral aspect of our design, is determined by a list of factors selected from each dimension. To cater to diverse application requirements, we implement four distinct neural networks: basic convolution [52], CIFAR-10 [41], HAR [58], and KWS [69]. Each of these networks is characterized by unique attributes in terms of layers, parameters, and floating-point operations per second (FLOPS), providing a robust framework to validate our system's effectiveness comprehensively.

Future AuT setup with Accelerators: In the second part, we consider comprehensive redesigns for future AuT, equipping them with AI reconfigurable accelerators for neural network inference. Here, we employ the extended accelerator evaluator and mapper developed in this work for inference system performance. The detailed configurations of the design space are presented in Table V. In the current design space, we also have the flexibility to adjust hardware-related parameters, namely the PE number and PE cache size. As detailed in Table V, our study employs two widely-used accelerator architectures, namely TPU and Eyeriss, and examines four notable network models. These include three classic Convolutional Neural Network (CNN) models and one transformer model and all of them have a suitable size for running on edge devices. AlexNet, a 7-layer CNN, exemplifies the standard architecture for image recognition tasks. VGG16, a deeper 13-layer CNN, utilizes smaller filters to achieve image recognition, contrasting with the use of larger filters in conventional designs. ResNet18, part of the ResNet family with 20 layers, introduces residual connections to the CNN architecture. Besides image recognition tasks, we also incorporate BERT, a prominent transformer model, representing Natural Language Processing (NLP) tasks. Our evaluation involves testing these four distinct networks on both accelerators to demonstrate the performance across a variety of scenarios.

In the experiment, we consider three objective functions that have been utilized in existing design targets. Firstly, we aim to minimize latency while adhering to the solar panel constraint (lat) as proposed in [24], [35], [49]. This target is suitable for scenarios where stringent hardware size requirements exist. Secondly, we aim to minimize the solar panel size while satisfying the latency constraint (sp) as discussed in [4]. This target is applicable in scenarios with specific application requirements. Lastly, we define the objective function as the product of latency and solar panel size (lat*sp) which provides a direct measure of the throughput achievable per unit area of the solar panel in the given scenario. This objective function effectively captures the overall system efficiency of AuT [67].

V. Experiment Results

A. Optimizing Existing AuTs with CHRYSALIS

In this part, we try to illustrate the benefits of the proposed CHRYSALIS for rapid AuT construction with the configuration in Table IV. Firstly, for the comprehensive search of the design space, we try 10,000 points in the hardware search and each of the layers are searched with 100 points. The whole design space reaches 10^{4+2*n}, where n is the layer number of the current network. To consider the different power environments, we use two solar environments brighter and darker environments for the design space search. When doing the search, we use the average latency under two solar environments as our results to ensure the system is able to run in both environments. We conduct each search on a workstation (Intel Core i5-12400 2.50GHz, 32GB RAM) for less than 6 hours (339 minutes, CIFAR-10). This timeframe is relatively short compared to the overall time required for AuT architecture design and can be further reduced by employing high-end computing servers.

Enhancement Analysis over Existing Designs: As is shown in Figure 6, the points are the searched hardware results. The points on the Pareto optimal curve are highlighted and the better points are shown in the figures. The better points are considered based on the objective of latency multiplied by solar panel size (lat*sp), showing the least space-time cost of the inference. According to the Pareto optimal curve, we position the tradeoff between inference latency and the energy harvester scale. The

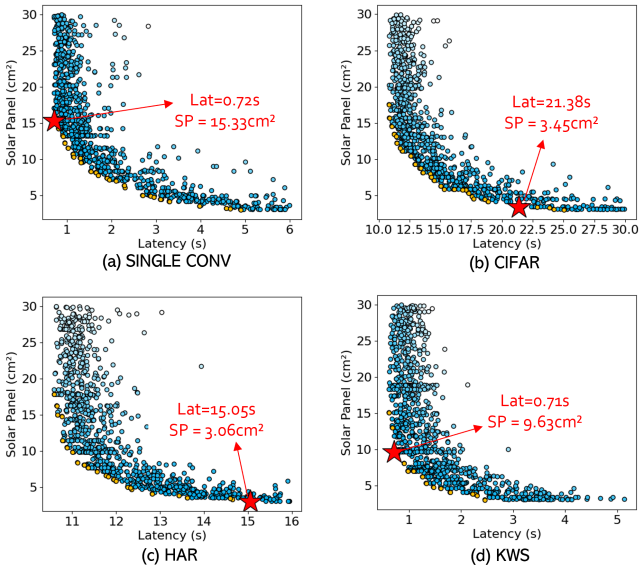


Figure 6: Searching for objective results for the existing MSP-based AuT systems: CHRYSALIS can improve the AuT architecture in the existing AuT systems.

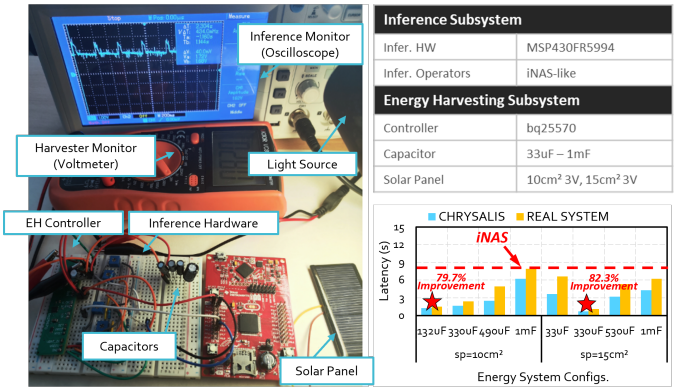


Figure 7: Validating the improved AuT system over iNAS on the real platform under different scenarios.

current search yields performance improvements compared to the original system [49]. Taking CIFAR as an example, the system performance using the original configuration was estimated to be lat*sp=150cm²*s (approximate value), whereas the final result of this search shows a 50.8% improvement over the original system.

Application to Real Platforms: Taking a single convolution layer as an example, we build the real system according to our search result, which is shown in Figure 7. We perform actual latency testing on the implemented system. We employ an oscilloscope to monitor the power supply status of the current inference subsystem, which manifested as periodic energy cycles. We confirm the capacitor voltage within a reasonable range during energy harvesting through a voltmeter. The EH Controller and Inference Hardware utilize existing chip-based solutions (BQ25570 [65], MSP430FR5994 LaunchPad [66]). We redesign PCB for the EH Subsystem to explore various capacitor configurations. As shown in the figure, we observe that: (1) The latency trends in the actual test results were similar to the simulated results, which demonstrates that our overall model is capable of effectively simulating the current system. (2) We attempt to replicate the position of the current design within the system if the design approach of `iNAS` are to be adopted without any further optimization($P_{\text{in}} = 6\text{mW}, C \geq 1\text{mF}$). Our system achieve 79.7% faster with the same solar panel size and 82.3% faster in latency with a bigger (15cm^2) solar panel. This indicates that considering hardware aspects led to significant efficiency improvements in the overall system.

Rationality Validation: To further validate the rationality of our design space and modeling approach, we conduct explorations of the search space. Firstly, we consider using capacitors of the same size (100uF) and employed solar panels of different sizes. Based on these configurations, we apply tile strategies for four different applications and obtained the energy consumption of each stage. As Figure 8 shows, we observe that a smaller solar panel size leads to excessive checkpoint energy overhead due to frequent checkpoints (Ckpt. Energy). Once the solar panel reaches a certain size, the total required energy can maintain a relatively stable state. However, the system efficiency (System Eff., $E_{\text{infer}}/E_{\text{eh}}$) starts to decrease because the additional collected energy may be wasted when the inference latency is longer than the energy harvesting latency. Finally, preferable solar panels are chosen with the better performance (lat*sp) in current occasions.

Secondly, we consider using solar panels of the same size (8cm^2) and employing different capacitor configurations. As Figure 9 shows, a small capacitor size leads to excessive checkpoint energy overhead due to frequent checkpoints (Ckpt. Energy), while a large capacitor size results in an obvious capacitor leakage energy (Cap. Leakage). Preferable capacitor sizes are highlighted due to the minimized latency with the current solar panel. In the current design, we only explore a few discrete values for the capacitor size. In practice, the specific values for the capacitor can be determined based on the available options for capacitor selection. Overall, the exploration highlights the importance of capacitor search, which aligns with the conclusions drawn from our modeling.

B. AI Accelerator-based AuT design with CHRYSALIS

To enhance the inference performance of AuT, it becomes imperative to incorporate dedicated accelerator architectures. Therefore, we aim to showcase CHRYSALIS's capacity for end-to-end architecture redesign, providing pre-RTL level design references for AuT accelerator development as well as other parts of AuT. Similar to the previous experiment, we search for each condition with 10^{4+2*n} points and use two solar environments as the target environment. On our experimental workstation, each search with the workstation takes less than 30 hours (1760 minutes, Resnet18-tpu).

Adaptability to Diverse SWaP Constraints: Figure 10 illustrates the CHRYSALIS's design ability under different conditions. We use CHRYSALIS to design ideal architectures for four existing neural networks (Alexnet, Resnet18, VGG16, BERT) and two architectures (TPU, Eyeriss) with three objectives (minimize latency with solar panel constraint, minimize

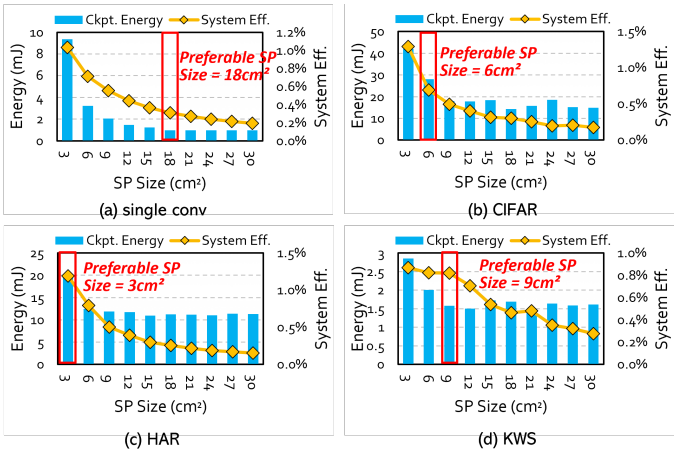


Figure 8: Optimizing solar panel for the existing AuT (capacitor=100uF[2]): Smaller solar panel size leads to excessive checkpoint energy (Ckpt. Energy), while bigger size results in decrease in system efficiency (System Eff.).

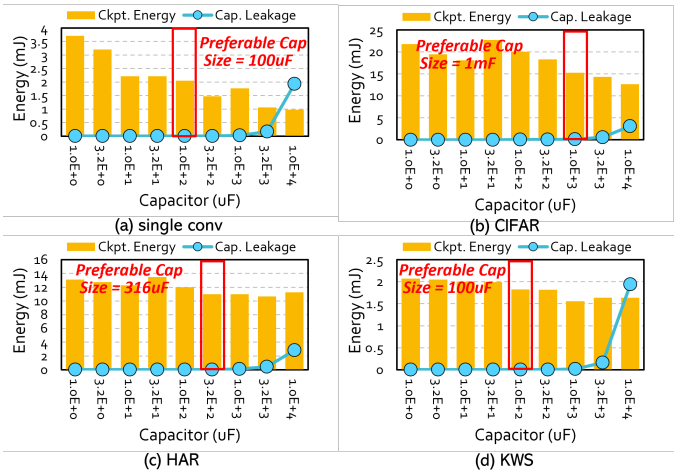


Figure 9: Optimizing capacitor size for the existing AuT (solar panel size=8cm[2]): Smaller capacitor size leads to excessive checkpoint energy overhead (Ckpt. Energy), while larger ones result in obvious capacitor leakage energy (Cap. Leakage).

solar panel with latency constraint, minimize lat*sp, latency multiplied by solar panel size). To showcase the advantages of CHRYSALIS, we compare it with six other methods as shown in Table VI, each of them lacking design considerations in certain dimensions. *wo/Cap* and *wo/SP* do not perform a search for the capacitance size and solar panel size in the energy harvesting system, but instead provide a fixed value. *wo/EA*, similarly, does not consider searching for parameters of the entire energy harvesting system. Likewise, *wo/PE* and *wo/cache* do not search for the number of PEs and cache size in the inference system, and *wo/IA* does not search for parameters of the entire inference system. From the results, we observe that CHRYSALIS consistently outperforms other methods under various conditions. Specifically, we observe that a limited design ability often leads to poorer design outcomes. For instance, results obtained from designs focuses solely on *wo/Cap* or *wo/SP* are superior to those obtained from the *wo/EA* approach (which ignores the design of both Cap and SP). Instead, CHRYSALIS can consistently outperform the other design approaches in a wide range of cases. By imposing SP constraints, the latency reduces from over 20s to below 5s (TPU, IA approach), and the average size of SP decreases by 36.2% under latency constraints (IA). These improvements make it feasible for the architecture to be implemented in real-world scenarios.

Energy Efficiency Analysis: Finally, we show the Energy Efficiency ($E_{\text{infer}}/E_{\text{eh}}$) for the better configurations across different network and architecture scenarios in Figure 11. The energy efficiency of the system obtained through CHRYSALIS approach can consistently maintain at a high level. Although some results may have slightly lower energy efficiency compared to other search results, this is because CHRYSALIS ensures energy savings during the computation process. Other search methods, particularly those that do not consider energy harvesting (EH), often yield lower energy efficiency in some scenarios. This is primarily due to the mismatch between the design of the SP and Cap components and the current

Table VI: Comparison of search spaces for different methods.

Search Methodology	EHer	Capacitor	PE Number	PE Cache
wo/Cap	✓	×	✓	✓
wo/SP [49]	×	✓	✓	✓
wo/EA [24], [35]	×	×	✓	✓
wo/PE	✓	✓	×	✓
wo/Cache	✓	✓	✓	×
wo/IA	✓	✓	×	×
CHRYSALIS (Ours)	✓	✓	✓	✓

inference subsystem. CHRYSALIS approach ensures a higher proportion of energy efficiency, indicating that energy is not wasted on leakage or system static power. This demonstrates the superiority of architecture through our search method over other approaches.

VI. Related Works

Edge AIoT Device: Energy harvesting systems and intermittent computing have been longstanding research areas, but most prior intermittent computing work targets low-computation scenarios [9], [31], [50], [56]. Early automated artificial intelligence system (AuT) research includes SONIC&TAILS, demonstrating feasible energy harvesting-based inference. RAD, ACE, and FLEX [30] explored further system optimizations. Additionally, Protean enables AuT verification [2], while iNAS [49], HAWAII [35], JAPARI [36] and Stateful [71] examine inference strategies. Nevertheless, some works tried to utilize in-memory inference for intermittent computing systems, leading to speedups and efficiency improvements [55]. However, most current AuT implementations use limited hardware like the MSP430FR5994 [66]. Bottlenecks remain for inference accelerators and memory. Unlike previous work, CHRYSALIS is a tool of EA/IA co-design that can improve hardware architectures to enable more efficient AuT designs for future edge computing devices.

Edge AI Acceleration: Developing AI accelerators for constrained hardware is an active research area [14], [20], [25], [26]. There have been several efforts to explore acceleration of

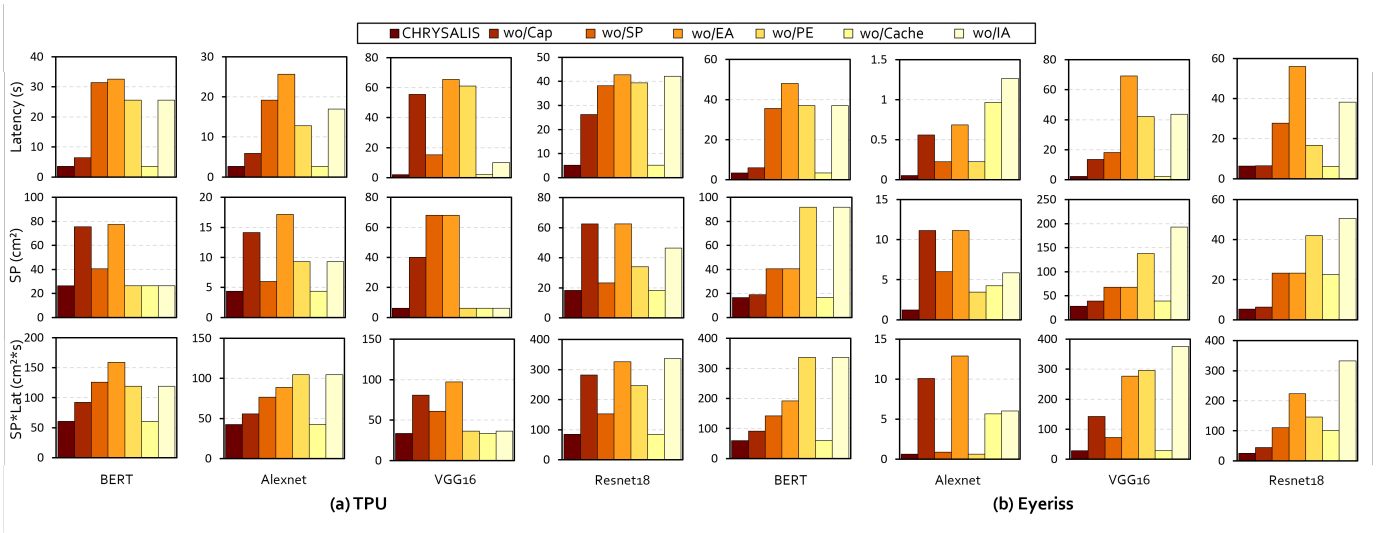


Figure 10: Design results for the four evaluated deep neural networks and the two AuT architectures with three objective functions: CHRYSALIS can consistently find the better configurations in all cases while the baselines can only work in specific circumstances. SP: Solar Panel Size. Lat: Latency.

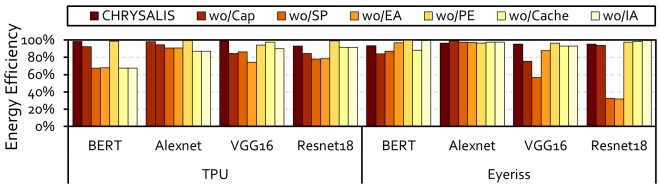


Figure 11: Comparison of energy efficiency.

ML inference with MB/GB sized models on FPGAs [7], [18], [21]. Different from these works, CHRYSALIS does not have constant energy support for executing AI application operations and hence such operations cause huge slowdowns. Recently, Qiu et al. proposed an adaptable RRAM crossbar accelerator named ResiRCA [54] for MAC operations for CNNs in energy harvesting environments. Nevertheless, it manages power at the crossbar level and neglects the importance of power source control and intra-application characteristics.

Intermittent Computing: Maximizing the energy potential of renewable sources requires power source-aware control techniques [32], [45]. Early work focused on low-power embedded systems [10], [32], [64]. Li et al. proposed load matching to leverage CPU slack time in a solar-powered wireless system [45]. Clark et al. demonstrated heuristic control for a battery-less solar RFID system [10]. Stewart et al. examined renewable intermittency and proposed power transfer systems [32]. Numerous maximum power point tracking algorithms exist [19], [64], with comparisons by Esram et al. [17]. King et al. reviewed photovoltaic panel modeling approaches [39], [60]. Unlike prior work, we propose the first renewable energy-aware power management technique coordinating provisioning control and load matching. Our approach enables joint optimization of energy utilization and application performance for EH systems.

Design Space Exploration: The increasing demand for AI has led to more complex DNN models and heterogeneous computing. To reduce manual design effort, researchers have proposed automated design space exploration categorized into three approaches: neural architecture search [3], [47], [68] to find ideal DNN models for workloads and constraints; automated DNN mapping [28] for co-design to map models onto hardware; and hardware design automation [62] to streamline chip design. In summary, these intelligent techniques aim to alleviate the cumbersome process of designing high-performing DNNs and hardware. However, they are not tailored for the complex AuT design which makes them efficient to directly apply to our scenario.

VII. Conclusion

As the significance of AuTs is rapidly growing, this paper introduces CHRYSALIS, a comprehensive co-design framework that automates the generation of ideal AuT designs tailored to various application scenarios and constraints. CHRYSALIS stands as a pioneering methodology and toolset, enabling the co-design of power budgets and computing capabilities. This advancement holds immense promise in facilitating the development and deployment of AuTs, an emerging and highly intriguing area of research and practical interest. By leveraging CHRYSALIS, researchers and practitioners can drive innovation and unlock the potential of AuTs to revolutionize a wide range of industries and domains. Further research and development in this field will undoubtedly yield transformative breakthroughs and open up new avenues for energy-efficient and autonomous systems.

Acknowledgment

We sincerely thank all the anonymous reviewers for their valuable comments and feedback. This work is supported by the National Natural Science Foundation of China (No. 62122053). Chao Li is the corresponding author.

References

[1] F. AL-Hazemi, Y. Peng, C.-H. Youn, J. Lorincz, C. Li, G. Song, and R. Boutaba, "Dynamic allocation of power delivery paths in consolidated data centers based on adaptive ups switching," in *Computer Networks (CN)*, vol. 144, 2018, pp. 254–270.

[2] A. Bakar, R. Goel, J. de Winkel, J. Huang, S. Ahmed, B. Islam, P. Pawełczak, K. S. Yıldırım, and J. Hester, "Protean: An Energy-Efficient and Heterogeneous Platform for Adaptive and Hardware-Accelerated Battery-Free Computing," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2023. [Online]. Available: https://doi.org/10.1145/3560905.3568561

[3] H. Benmeziane, K. El Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, and N. Wang, "Hardware-Aware Neural Architecture Search: Survey and Taxonomy," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. [Online]. Available: https://doi.org/10.24963/ijcai.2021/592

[4] M. Brehler, L. Camphausen, B. Heidebroek, D. Krön, H. Gründer, and S. Camphausen, "Making machine learning more energy efficient by bringing it closer to the sensor," *IEEE Micro*, vol. 43, no. 6, 2023.

[5] D. Cao, D. Xue, Z. Ma, and H. Mei, "Xiuos: an open-source ubiquitous operating system for industrial internet of things," in *Science China Information Sciences (SCIS)*, 2021, pp. 1869–1919.

[6] L. Catalan, M. Araiz, P. Aranguren, G. D. Padilla, P. A. Hernandez, N. M. Perez, C. Garcia de la Noceda, J. F. Albert, and D. Astrain, "Prospects of Autonomous Volcanic Monitoring Stations: Experimental Investigation on Thermoelectric Generation from Fumaroles," *Sensors*, vol. 20, no. 12, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/12/3547

[7] A. M. Caulfield, E. S. Chung, A. Putnam, H. Angepat, J. Fowers, M. Haselman, S. Heil, M. Humphrey, P. Kaur, J.-Y. Kim, D. Lo, T. Massengill, K. Ovtcharov, M. Papamichael, L. Woods, S. Lanka, D. Chiou, and D. Burger, "A cloud-scale acceleration architecture," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2016. [Online]. Available: https://doi.org/10.1109/MICRO.2016.7783710

[8] Y.-H. Chen, T. Krishna, J. Emer, and V. Sze, "14.5 Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, 2016. [Online]. Available: https://doi.org/10.1109/ISSCC.2016.7418007

[9] J. Choi, H. Joe, Y. Kim, and C. Jung, "Achieving Stagnation-Free Intermittent Computation with Boundary-Free Adaptive Execution," in *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2019. [Online]. Available: https://doi.org/10.1109/RTAS.2019.00035

[10] S. S. Clark, J. Gummeson, K. Fu, and D. Ganesan, "Towards Autonomously-Powered CRFIDs," in *ACM Workshop on Power Aware Computing and Systems (PACS)*, 2009.

[11] G. Cloud, "Edge TPU - AI at the Edge," https://cloud.google.com/edge-tpu, 2021.

[12] A. Colin, E. Ruppel, and B. Lucia, "A Reconfigurable Energy Storage Architecture for Energy-harvesting Devices," in *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2018. [Online]. Available: https://doi.org/10.1145/3173162.3173210

[13] J. de Winkel, C. Delle Donne, K. S. Yildirim, P. Pawełczak, and J. Hester, "Reliable Timekeeping for Intermittent Computing," in *Proceedings of the 25th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2020. [Online]. Available: https://doi.org/10.1145/3373376.3378464

[14] D. Dennis, C. Pabbaraju, H. V. Simhadri, and P. Jain, "Multiple Instance Learning for Efficient Sequential Data Classification on Resource-constrained Devices," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/d9fbed9da256e344c1fa46bb46c34c5f-Paper.pdf

[15] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, "ShiDianNao: Shifting Vision Processing Closer to the Sensor," in *International Symposium on Computer Architecture (ISCA)*, 2015. [Online]. Available: https://doi.org/10.1145/2749469.2750389

[16] EnOcean, "Energy Harvesting Wireless Power for the Internet of Things," https://www.enocean.com/wp-content/uploads/redaktion/pdf/white_paper/WhitePaper_Internet_of_Things_EnOcean_v2.0.pdf, 2019.

[17] T. Esram and P. L. Chapman, "Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques," *IEEE Transactions on Energy Conversion (TEC)*, 2007. [Online]. Available: https://doi.org/10.1109/TEC.2006.874230

[18] C. Farabet, C. Poulet, J. Y. Han, and Y. LeCun, "Cnp: An fpga-based processor for convolutional networks," *International Conference on Field Programmable Logic and Applications (FPGA)*, 2009. [Online]. Available: https://doi.org/10.1109/FPL.2009.5272559

[19] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli, "Optimization of perturb and observe maximum power point tracking method," *IEEE Transactions on Power Electronics (TPEL)*, 2005. [Online]. Available: https://doi.org/10.1109/TPEL.2005.850975

[20] Y. Feng, B. Tian, T. Xu, P. N. Whatmough, and Y. Zhu, "Mesorasi: Architecture support for point cloud analytics via delayed-aggregation," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:221140017

[21] J. Fowers, K. Ovtcharov, M. Papamichael, T. Massengill, M. Liu, D. Lo, S. Alkalay, M. Haselman, L. Adams, M. Ghandi, S. Heil, P. Patel, A. Sapek, G. Weisz, L. Woods, S. Lanka, S. K. Reinhardt, A. M. Caulfield, E. S. Chung, and D. Burger, "A Configurable Cloud-Scale DNN Processor for Real-Time AI," *International Symposium on Computer Architecture (ISCA)*, 2018. [Online]. Available: https://doi.org/10.1109/ISCA.2018.00012

[22] Gartner, "Autonomous Things: Technology Use Cases for R&D," Tech. Rep., 2023. [Online]. Available: https://www.gartner.com/en/innovation-strategy/trends/autonomous-things

[23] T. N. Gia, M. Ali, I. B. Dhaou, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "IoT-based continuous glucose monitoring system: A feasibility study," *Procedia Computer Science*, vol. 109, 2017, 8th International Conference on Ambient Systems, Networks and Technologies, ANT-2017 and the 7th International Conference on Sustainable Energy Information Technology. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050917310281

[24] G. Gobieski, B. Lucia, and N. Beckmann, "Intelligence Beyond the Edge: Inference on Intermittent Embedded Systems," in *Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2019. [Online]. Available: https://doi.org/10.1145/3297858.3304011

[25] C. Gupta, A. S. Suggala, A. Goyal, H. V. Simhadri, B. Paranjape, A. Kumar, S. Goyal, R. Udupa, M. Varma, and P. Jain, "ProtoNN: Compressed and Accurate kNN for Resource-scarce Devices," in *Proceedings of International Conference on Machine Learning (ICML)*, 2017. [Online]. Available: https://proceedings.mlr.press/v70/gupta17a.html

[26] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding," *International Conference on Learning Representations (ICLR)*, 2016. [Online]. Available: https://doi.org/https://doi.org/10.48550/arXiv.1510.00149

[27] W. F. Holmgren, C. W. Hansen, and M. A. Mikofski, "pvlib python: a python package for modeling solar energy systems," *Journal of Open Source Software*, vol. 3, no. 29, 2018. [Online]. Available: https://doi.org/10.21105/joss.00884

[28] D. Hong, K. Choi, H. Y. Lee, J. Yu, N. Park, Y. Kim, and J. Lee, "Enabling hard constraints in differentiable neural network and accelerator co-exploration," in *ACM/IEEE Design Automation Conference (DAC)*, 2022. [Online]. Available: https://doi.org/10.1145/3489517.3530507

[29] A. Hoseinghorban, M. R. Bahrami, A. Ejlali, and M. A. Abam, "CHANCE: Capacitor Charging Management Scheme in Energy Harvesting Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 40, no. 3, 2021. [Online]. Available: https://doi.org/10.1109/TCAD.2020.3003295

[30] S. Islam, J. Deng, S. Zhou, C. Pan, C. Ding, and M. Xie, "Enabling Fast Deep Learning on Tiny Energy-Harvesting IoT Devices," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022. [Online]. Available: https://doi.org/10.23919/DATE54114.2022.9774756

[31] H. Jayakumar, A. Raha, and V. Raghunathan, "QUICKRECALL: A Low Overhead HW/SW Approach for Enabling Computations across Power Cycles in Transiently Powered Computers," in *27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*, 2014. [Online]. Available: https://doi.org/10.1109/VLSID.2014.63

[32] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Fourth International Symposium on Information*

Processing in Sensor Networks (IPSN), 2005. [Online]. Available: https://doi.org/10.1109/IPSN.2005.1440974

[33] K. Johnson, Z. Englhardt, V. Arroyos, D. Yin, S. Patel, and V. Iyer, "MilliMobile: An Autonomous Battery-Free Wireless Microrobot," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2023. [Online]. Available: https://doi.org/10.1145/3570361.3613304

[34] C.-K. Kang, C.-H. Lin, P.-C. Hsiu, and M.-S. Chen, "HomeRun: HW/SW Co-Design for Program Atomicity on Self-Powered Intermittent Systems," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*, 2018. [Online]. Available: https://doi.org/10.1145/3218603.3218633

[35] C.-K. Kang, H. R. Mendis, C.-H. Lin, M.-S. Chen, and P.-C. Hsiu, "Everything Leaves Footprints: Hardware Accelerated Intermittent Deep Inference," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 39, no. 11, 2020. [Online]. Available: https://doi.org/10.1109/TCAD.2020.3012217

[36] C.-K. Kang, H. R. Mendis, C.-H. Lin, M.-S. Chen, and P.-C. Hsiu, "More Is Less: Model Augmentation for Intermittent Deep Inference," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 21, no. 5, oct 2022. [Online]. Available: https://doi.org/10.1145/3506732

[37] S.-C. Kao and T. Krishna, "GAMMA: Automating the HW Mapping of DNN Models on Accelerators via Genetic Algorithm," in *Proceedings of the 39th International Conference on Computer-Aided Design (ICCAD)*, 2020. [Online]. Available: https://doi.org/10.1145/3400302.3415639

[38] S.-C. Kao, M. Pellauer, A. Parashar, and T. Krishna, "DiGamma: domain-aware genetic algorithm for HW-mapping co-optimization for DNN accelerators," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022. [Online]. Available: https://doi.org/10.23919/DATE54114.2022.9774568

[39] D. King, J. Dudley, and W. Boyson, "PVSIM: A simulation program for photovoltaic cells, modules, and arrays," *Sandia National Labs (SNL)*, 1996. [Online]. Available: https://www.osti.gov/biblio/241578

[40] S. Krishnan, Z. Wan, K. Bhardwaj, P. Whatmough, A. Faust, S. Neuman, G.-Y. Wei, D. Brooks, and V. J. Reddi, "Automatic Domain-Specific SoC Design for Autonomous Unmanned Aerial Vehicles," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2022. [Online]. Available: https://doi.org/10.1109/MICRO56248.2022.00033

[41] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009. [Online]. Available: https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf

[42] H. Kwon, P. Chatarasi, M. Pellauer, A. Parashar, V. Sarkar, and T. Krishna, "Understanding Reuse, Performance, and Hardware Cost of DNN Dataflow: A Data-Centric Approach," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*. [Online]. Available: https://doi.org/10.1145/3352460.3358252

[43] C. Li, Y. Hu, L. Liu, J. Gu, M. Song, X. Liang, J. Yuan, and T. Li, "Towards sustainable in-situ server systems in the big data era," in *International Symposium on Computer Architecture (ISCA)*, 2015, pp. 14–26.

[44] C. Li, W. Zhang, C.-B. Cho, and T. Li, "Solarcore: Solar energy driven multi-core architecture power management," in *International Symposium on High Performance Computer Architecture (HPCA)*, 2011, pp. 205–216.

[45] D. Li and P. H. Chou, "Application/architecture power co-optimization for embedded systems powered by renewable sources," in *ACM/IEEE Design Automation Conference (DAC)*, 2005. [Online]. Available: https://doi.org/https://doi.org/10.1145/1065579.1065742

[46] L. Liu, C. Li, H. Sun, Y. Hu, J. Gu, and T. Li, "Baat: Towards dynamically managing battery aging in green datacenters," in *International Conference on Dependable Systems and Networks (DSN)*, 2015, pp. 307–318.

[47] X. Luo, D. Liu, H. Kong, S. Huai, H. Chen, and W. Liu, "You only search once: on lightweight differentiable architecture search for resource-constrained embedded platforms," in *ACM/IEEE Design Automation Conference (DAC)*, 2022. [Online]. Available: https://doi.org/10.1145/3489517.3530488

[48] M. Lv and E. Xu, "Deep Learning on Energy Harvesting IoT Devices: Survey and Future Challenges," *IEEE Access*, vol. 10, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3225092

[49] H. R. Mendis, C.-K. Kang, and P.-c. Hsiu, "Intermittent-Aware Neural Architecture Search," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 20, no. 5s, sep 2021. [Online]. Available: https://doi.org/10.1145/3476995

[50] S. Naderiparizi, A. N. Parks, Z. Kapetanovic, B. Ransford, and J. R. Smith, "WISPCam: A battery-free RFID camera," in *IEEE International Conference on RFID (RFID)*, 2015. [Online]. Available: https://doi.org/10.1109/RFID.2015.7113088

[51] M. Nasajpour, S. Pouriyeh, R. M. Parizi, M. Dorodchi, M. Valero, and H. R. Arabnia, "Internet of Things for current COVID-19 and future pandemics: An exploratory study," *Journal of healthcare informatics research*, vol. 4, 2020. [Online]. Available: https://doi.org/10.1007/s41666-020-00080-6

[52] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," 2015. [Online]. Available: https://doi.org/10.48550/arXiv.1511.08458

[53] A. Parashar, P. Raina, Y. S. Shao, Y.-H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. Emer, "Timeloop: A Systematic Approach to DNN Accelerator Evaluation," in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2019. [Online]. Available: https://doi.org/10.1109/ISPASS.2019.00042

[54] K. Qiu, N. Jao, M. Zhao, C. Mishra, G. Gudukbay Akbulut, S. Jose, J. Sampson, M. Kandemir, and V. Narayanan, "ResiRCA: A Resilient Energy Harvesting ReRAM Crossbar-Based Accelerator for Intelligent Embedded Processors," *International Symposium on High Performance Computer Architecture (HPCA)*, 2020. [Online]. Available: https://doi.org/10.1109/HPCA47549.2020.00034

[55] K. Qiu, N. Jao, M. Zhao, C. S. Mishra, G. Gudukbay, S. Jose, J. Sampson, M. T. Kandemir, and V. Narayanan, "ResiRCA: A Resilient Energy Harvesting ReRAM Crossbar-Based Accelerator for Intelligent Embedded Processors," in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2020. [Online]. Available: https://doi.org/10.1109/HPCA47549.2020.00034

[56] B. Ransford, J. Sorber, and K. Fu, "Mementos: system support for long-running computation on RFID-scale devices," in *Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2011. [Online]. Available: https://doi.org/10.1145/1950365.1950386

[57] E. C. Reuben and L. Y. Meng, "Programmable Microcontroller Based Power Management Module for Batteryless intermittent Computing Systems," in *2022 5th International Conference on Intelligent Robotics and Control Engineering (IRCE)*, 2022. [Online]. Available: https://doi.org/10.1109/IRCE55557.2022.9963031

[58] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, and X. Parra, "Human Activity Recognition Using Smartphones," UCI Machine Learning Repository, 2012. [Online]. Available: https://doi.org/10.24432/C54S4K

[59] J. San Miguel, K. Ganesan, M. Badr, C. Xia, R. Li, H. Hsiao, and N. Enright Jerger, "The EH Model: Early Design Space Exploration of Intermittent Processor Architectures," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2018. [Online]. Available: https://doi.org/10.1109/MICRO.2018.00055

[60] D. Sera, R. Teodorescu, and P. Rodriguez, "PV panel model based on datasheet values," in *IEEE International Symposium on Industrial Electronics (ISIE)*, 2007. [Online]. Available: https://doi.org/10.1109/ISIE.2007.4374981

[61] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios," *IEEE Access*, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2970118

[62] A. Sohrabizadeh, Y. Bai, Y. Sun, and J. Cong, "Automated accelerator optimization aided by graph neural networks," in *ACM/IEEE Design Automation Conference (DAC)*, 2022. [Online]. Available: https://doi.org/10.1145/3489517.3530409

[63] M. Song, K. Zhong, J. Zhang, Y. Hu, D. Liu, W. gong Zhang, J. Wang, and T. Li, "In-Situ AI: Towards Autonomous and Incremental Deep Learning for IoT Systems," *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2018. [Online]. Available: https://doi.org/10.1109/HPCA.2018.00018

[64] J. Taneja, J. Jeong, and D. Culler, "Design, Modeling, and Capacity Planning for Micro-solar Power Sensor Networks," in *Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*, 2008. [Online]. Available: https://doi.org/10.1109/IPSN.2008.67

[65] TI.com, "BQ25570 data sheet, product information and support," https://www.ti.com/product/BQ25570.

[66] TI.com, "MSP430FR5994 data sheet, product information and support," https://www.ti.com/product/MSP430FR5994.

[67] Y. Wang, Y. Liu, C. Wang, Z. Li, X. Sheng, H. G. Lee, N. Chang, and H. Yang, "Storage-Less and Converter-Less Photovoltaic Energy

Harvesting With Maximum Power Point Tracking for Internet of Things," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 35, no. 2, 2016. [Online]. Available: https://doi.org/10.1109/TCAD.2015.2446937

[68] Z. Wang, C. Wan, Y. Chen, Z. Lin, H. Jiang, and L. Qiao, "Hierarchical memory-constrained operator scheduling of neural architecture search networks," in *ACM/IEEE Design Automation Conference (DAC)*, 2022. [Online]. Available: https://doi.org/10.1145/3489517.3530472

[69] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1804.03209

[70] Y. N. Wu, J. S. Emer, and V. Sze, "Accelergy: An Architecture-Level Energy Estimation Methodology for Accelerator Designs," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019. [Online]. Available: https://doi.org/10.1109/ICCAD45719.2019.8942149

[71] C.-H. Yen, H. R. Mendis, T.-W. Kuo, and P.-C. Hsiu, "Stateful Neural Networks for Intermittent Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 41, no. 11, 2022. [Online]. Available: https://doi.org/10.1109/TCAD.2022.3197513

[72] F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy efficiency," *Energy and Buildings*, vol. 83, 2014, science behind and beyond the solar decathlon Europe 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778814003569