A Greener Edge: A Framework on Carbon-aware Edge ML System Design

Xuesi Chen
Cornell Tech
New York, New York, USA
xc562@cornell.edu

Ilan Mandel
Cornell Tech
New York, New York, USA
im334@cornell.edu

Eren Yildiz
Georgia Institute of Technology
Atlanta, Georgia, USA
eyildiz8@gatech.edu

Josiah Hester
Georgia Institute of Technology
Atlanta, Georgia, USA
josiah@gatech.edu

Udit Gupta
Cornell Tech
New York, New York, USA
ugupta@cornell.edu

Abstract

Edge devices are often deployed at scale, yet their environmental impact, shaped by complex interactions between hardware choices, workload demands, power systems, and deployment context, has been overlooked by the mobile computing community. We present *MicroGreen*, a design-time framework that enables carbon-aware design for edge ML systems. By combining component-level carbon models with workload profiling and environment-aware energy analysis, *MicroGreen* identifies carbon-optimal configurations across diverse conditions. Our results show that the most energy-efficient processor is not always the most sustainable, and that ambient energy availability, inference rate, and deployment lifetime can shift the carbon-optimal design by over an order of magnitude. Through a real vision-based visitor detection and counting deployment in New York City parks, we demonstrate that heterogeneous, location-aware designs reduce total emissions by 47.72% compared to a homogeneous baseline.

CCS Concepts

• **Hardware → Impact on the environment**; • **Computer systems organization → Embedded hardware**.

Keywords

Sustainable Computing, Edge Computing, Internet of Things, Design Space Exploration, Heterogeneous Deployment

ACM Reference Format:
Xuesi Chen, Ilan Mandel, Eren Yildiz, Josiah Hester, and Udit Gupta. 2026. A Greener Edge: A Framework on Carbon-aware Edge ML System Design . In *The 24th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '26), June 21–25, 2026, Cambridge, United Kingdom.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3745756.3809201

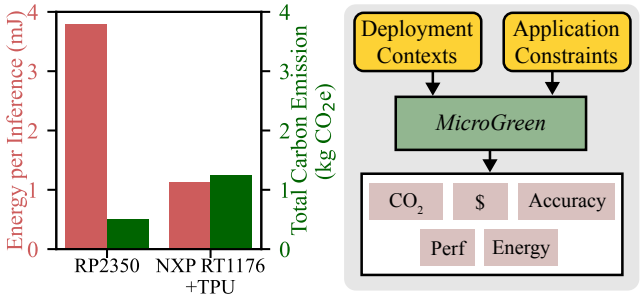

Figure 1: Left: Energy efficiency alone does not determine carbon emissions under solar harvesting, as shown by running keyword spotting at 1 IPS under 900 W/m² solar condition on two systems built using different MCUs. The total carbon emissions here includes not just the MCU used but also energy harvesting devices and other necessary peripherals components. Right: *MicroGreen* incorporates ambient energy, usage frequency and deployment factors to estimate carbon across scenarios.

1 Introduction

Increasingly prevalent edge devices already play a crucial role across diverse applications, from air quality monitoring [24] and environmental sensing [4] to personal health tracking via wearables and smartphones [13], spanning the broader Internet of Things (IoT) ecosystem. Despite their myriad applications, edge devices incur high environmental costs. Embedded and mobile systems contribute nearly one third of the ICT footprint, which itself accounts for roughly 4% of global carbon emissions, on par with the aviation industry [32, 39]. While sustainable computing research has largely focused on data centers and AI models [2, 39, 58, 62], the high environmental impact of edge devices remains significantly understudied.

To ensure the sustainability of the IoT ecosystem, we must consider and optimize the lifecycle carbon footprint of IoT systems. Life cycle carbon emissions include two key components — (1) *embodied carbon*, which refers to emissions resulting from the manufacturing of their components, such as PCBs, processors, sensors, and batteries, and (2) *operational carbon*, which encompasses emissions associated with powering, recharging the workload over time [38, 39].

Unlike energy-intensive datacenters [38, 39], edge devices use low-power microcontrollers[5, 35], domain-specific accelerators [12,

MobiSys '26, Cambridge, United Kingdom

ACM ISBN 979-8-4007-2027-7/26/06
https://doi.org/10.1145/3745756.3809201

97], and efficient inference pipelines[22, 28, 81] to minimize runtime energy [6, 26, 98] The carbon emissions edge systems produce during operation do not dominate the total lifecycle emissions. Embodied carbon accounts for nearly 75% of the overall environmental impact of edge devices [39]. Additionally, embodied and operational carbon cannot be evaluated independently for embedded devices; compute, sensing, storage and power must be co-provisioned based on application demands and environmental conditions. Small differences in deployment context can shift optimal configurations and dramatically change total carbon impact (up to 47.72% in our case study). Energy-efficient MCUs can reduce operational energy but increase embodied carbon. As shown in Figure 1, the NXP RT1176+TPU is more energy efficient than the RP2350, yet it has higher total carbon emissions due to greater embodied carbon. Component-level life cycle assessment (LCA) [33, 78, 104] and runtime energy optimization [5, 6, 80] cannot capture these interconnected tradeoffs.

Despite the importance of design-time trade-offs, tools remain fragmented. LCA tools [43, 63, 72, 74, 104] provide coarse assessments and do not account for workload-dependent behavior or environmental variability. Power-management [44] and energy-harvesting tools [5, 6, 35, 71, 98] focus on runtime energy and ignore manufacturing emissions. No existing approach jointly models compute, sensing, storage, and power choices with application requirements and spatio-temporal environmental conditions. Designers lack a systematic method to explore how device architecture, workload, and deployment site collectively shape lifecycle emissions. The combinatorial and contextual nature of these interactions, demonstrated in Figure 2, makes manual design exploration error-prone and impractical.

We propose *MicroGreen*, a design space exploration framework for sustainable edge devices. *MicroGreen* integrates embodied-carbon modeling, workload characterization, and environment-dependent operational analysis to guide carbon-aware hardware and system design decisions. It incorporates parametric embodied-carbon models for MCUs, sensors, regulators, storage, batteries, capacitors, and solar panels, leveraging LCA databases [38, 55, 104]. Using these models, *MicroGreen* takes as input application requirements (e.g., workload, duty cycle, deployment lifetime) and environmental traces (e.g., solar availability) and identifies carbon-optimal configurations. We empirically characterize a repository of MCUs under diverse workloads, integrate peripheral and power-source options, and generate Pareto-optimal designs to illustrate trade-offs between carbon footprint, performance, and cost. We have open-sourced the *MicroGreen* framework, as well as the carbon modeling and profiling tools on github[1], supporting robust and sustainable edge ML system design.

The core contributions of this paper are listed below:

- We propose *MicroGreen*, a design space exploration framework with a component-level carbon model that enables carbon-aware design decisions for IoT devices. *MicroGreen*'s rapid design space exploration allows developers to tailor devices to individual environmental conditions and use cases with sustainability in mind.

[1]https://github.com/S4AI-CornellTech/MicroGreen

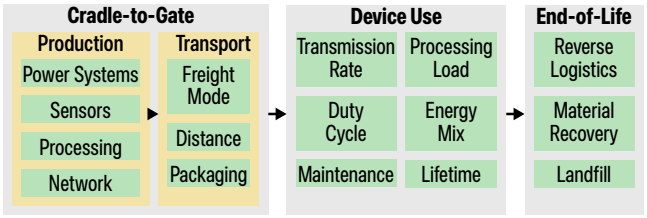

Figure 2: Key factors influencing lifecycle carbon emissions of IoT devices.

- Using *MicroGreen* we show ambient energy availability, device lifetime, inference frequency (e.g., duty cycle), and power-system choice all impact the carbon-optimal design.
- We demonstrate our framework by designing heterogeneous visitor counters for multiple entrance locations in Central Park, each with different light availability and visitor flow. This approach reduces carbon emissions by up to 47.72% compared to a standard homogeneous baseline.

2 Background & Related Work

Edge deployments involve many distributed devices operating continuously under strict energy constraints. Previous research [48, 53, 54] has focused on optimizing energy efficiency. Real-world deployments operate under variable conditions: fluctuating power, environmental unpredictability, and diverse operational constraints. Field studies show that shading, weather, link instability, and maintenance significantly impact performance and lifetime [5, 23, 35, 92, 100]. Deployment-dependent dynamics affect both system reliability and environmental impact. Although energy efficiency is essential for operating under these constraints, it does not guarantee sustainability; the two are fundamentally distinct.

2.1 Sustainability of IoT Systems

Manufacturing edge devices requires resource-intensive processes. Semiconductor production depletes aquifers through ultrapure water consumption and damages ecosystems through critical material extraction [89]. This work focuses on carbon sustainability, quantified using Global Warming Potential, measured in CO_2-equivalent units or CO_2eq.

Carbon emissions (CO_2eq) can be categorized into two types: operational and embodied. Although edge device consume only small amounts of power (e.g., milliwatts for embedded devices), the embodied carbon footprint incurs high CO_2eq. Crucially, while improving energy efficiency reduces operational energy use, it does not address the significantly larger embodied emissions linked to the hardware itself.

We evaluate all systems cradle-to-gate, focusing on the sustainability obligations which designers and manufacturers have the most control. Unlike datacenters, which have well-studied secondary markets and end-of-life procedures [93], the full lifecycle of edge devices is difficult to evaluate comprehensively. While recent work has explored reuse [64, 82, 95] and cradle-to-grave environmental costs [72], the distributed and heterogeneous deployment of edge devices limits the generalizability of device use and full lifecyle assessments [14].

2.2 Carbon Modeling and Lifecycle Assessment

Prior research on carbon modeling has examined emissions across multiple layers of the computing stack. Works, such as ACT [38], Chasing Carbon [39] and Bashir et al. [8], show that embodied emissions from manufacturing and infrastructure dominate the lifecycle footprint of modern compute platforms. System-level assessments [49, 77], , including SCARIF [49] and Google's TPU Lifecycle Analysis [77] and many others [57, 60, 102], demonstrate how larger dies, expanded memory hierarchies, and heterogeneous integration increase lifecycle emissions. For edge devices, embodied carbon from production typically dominates total lifecycle emissions [36, 39, 104].

Device carbon costs are evaluated through lifecycle assessment (LCA) following ISO 14040, ISO 14044, and ITUL.1410 [31, 45, 46]. For each component, a LCA quantifies energy consumption, material usage, and production yield during manufacturing. Recent work has applied LCA analysis to device-specific emerging technologies such as FeFET eNVMs [94] and photonic chips [30]. However, emissions factors must be mapped from LCA databases with limited data availability [9], and LCAs typically occur late in the design process when changes are costly.

Bottom-up assessments [73, 75] reveal substantial variability in IoT footprints due to component selection, battery maintenance, and database uncertainties. Component-specific studies of sensors [69], processors [11, 38, 49], memory [83, 91], and capacitors [78] provide finer-grained emissions insights. Product-level disclosures such as Fairphone [25] offer practical baselines, while DeltaLCA [104] enables comparative trade-off analysis without absolute values that can be operationalized at design time.

However, existing approaches lack a unified view that jointly considers hardware selection, workload behavior, deployment conditions, and energy availability. IoT systems are often deployed as networked fleets where deployment context, such as irradiance levels and operational patterns, significantly impacts carbon footprint. Our work enables customizing device configurations for specific locations and applications rather than using standardized designs across heterogeneous deployments.

2.3 Carbon-Aware System Design and Optimization

Mobile and networked edge devices are widely integrated across diverse environments: offices use motion sensors for energy management [18], municipalities deploy vehicle counting cameras [47], and wearables perform real-time health monitoring using on-device ML [16]. IoT systems span agriculture [87], manufacturing [96], healthcare [52], and environmental monitoring [65], with hardware ranging from lightweight MCUs to embedded GPUs, ASICs, and FPGAs [40, 51, 67].

Commercial-off-the-shelf (COTS) platforms enable rapid design iteration while leveraging established hardware and software ecosystems [66]. We categorize edge system built using COTS microcontrollers into four key subsystems:

- **Power**: Energy harvesters (solar panels, thermoelectric generators) and storage (supercapacitors, batteries).
- **Sensors**: Accelerometers, microphones, and cameras with varying complexity, resolution, and carbon footprints.
- **Processing**: Microprocessors, sometimes supplemented by AI accelerators for ML workloads, plus volatile (RAM) and non-volatile (flash, EEPROM) memory.
- **Networking**: Wireless protocols (WiFi, Bluetooth, BLE, LoRa, LTE/5G) requiring specialized radio ICs and antennas.

Design choices made for one subsystem constrain choices made in the others. More powerful MCUs require more power generation and storage, higher frequency sensing requires processing and networking systems capable of transforming and sending that data.

Prior work has explored carbon- and energy-aware optimization for these systems. Carbon-aware schedulers such as GreenScale [79] and CASPER [80] adjust execution based on spatial and temporal variation in carbon intensity across mobile, edge, and cloud resources. Complementary work by Desai et al. [43] illustrates how sensing, communication, and cloud analytics jointly influence system carbon impact. Batteryless and energy-harvested computing systems such as REHASH [6] and Amulet [44] provide adaptive runtime support for intermittent devices or long-lived sensing devices. CO2CoDe [71] demonstrates how hardware choices, such as capacitor type, can directly influence embodied carbon.

These approaches primarily optimize at execution time and treat deployment conditions, lifecycle modeling, and energy-aware execution as separate concerns. In contrast, our work provides a unified design-time framework that jointly considers hardware selection, embodied costs, workload requirements, deployment environments, and energy availability. By co-optimizing across these dimensions, we reveal carbon trade-offs that existing LCA tools and runtime optimizers cannot capture, enabling heterogeneous deployments where devices are customized for specific locations and conditions rather than standardized across a fleet.

3 Embodied Carbon Modeling

As mentioned in background, modeling embodied carbon emissions at the component level is key to supporting carbon-aware design decisions for edge devices, particularly given the diversity of components used across platforms. Thus, to understand the environmental footprint of edge devices and enable carbon-aware system design, we decompose each subsystem into its fundamental electronic components, for example integrated circuits, resistors, batteries, and auxiliary parts such as connectors. This section explains how *MicroGreen* quantifies the embodied carbon of edge devices by modeling emissions across basic electronic components.

To illustrate the modeling results, Table 1 reports the embodied emissions of minimal implementation boards for eight example COTS processors. For each device, we extract the full Bill of Materials (BoM) from schematics or publicly available documentation, categorize every component on the board, and compute embodied carbon using process-specific literature or emissions factors from industrial life-cycle databases.

3.1 Integrated Circuits

Given their complexity and manufacturing cost, integrated circuits (as shown in Figure 3) typically account for a significant portion of edge device's embodied carbon. For the purposes of this work, integrated circuits span system-on-chips and MCU's, memories devices, and CMOS image sensors.

Table 1: List of microcontrollers used in the experiment with processor information, carbon emission and cost.

Board	CPU	ML Accelerator	RAM	Flash	Framework	kg CO_2 eq	Cost ($)
Coral Dev Board Micro	NXP i.MX RT1176 MCU Cortex-M7 @ 800 MHz Cortex-M4 @ 400 MHz	Google Edge TPU	64 MB	128 MB	FreeRTOS	1.14 ± 0.1151	79.99
Adafruit Feather nRF52840 Express	Nordic nRF52840 Cortex-M4F @ 64 MHz	N/A	256 KB	1 MB	Arduino	0.18 ± 0.0307	24.95
Raspberry Pi Pico	RP2040 Dual-core Cortex-M0+@ 133 MHz	N/A	264 KB	2 MB	Pico SDK	0.32 ± 0.0066	4.00
Raspberry Pi Pico 2	RP2350 Dual-core Cortex-M33@ 150 MHz	N/A	520 KB	4 MB	Pico SDK	0.39 ± 0.0129	7.00
ESP32	Dual-core Xtensa LX6 @ up to 240 MHz	N/A	520 KB	4 MB	ESP-IDF	0.21 ± 0.0206	8.00
ESP32-C6	Single-core RISC-V @ up to 160 MHz	N/A	512 KB	4 MB	ESP-IDF	0.20 ± 0.0165	14.95
ESP32-S3	Dual-core Xtensa LX7 @ up to 240 MHz	N/A	512 KB	8 MB	ESP-IDF	0.26 ± 0.0323	17.5
STM32F411 BlackPill	ARM Cortex-M4F @ 100 MHz	N/A	128 KB	512 KB	STM32Cube	0.45 ± 0.0233	16.5

System-on-Chip (SoC) and Microcontroller Units. We estimate the embodied carbon of each SoC following the methodology in ACT [38], an widely used, open-source tool in the computer architecture and systems community to quantify the embodied emissions of SoC's [2, 42, 58, 88]. ACT models emissions as the product of two main parameters: die area and carbon intensity per unit area. Die areas when not directly accessible, are inferred from package-to-die ratios from DeltaLCA [104] and empirical relations in prior studies [56]. Carbon intensity depends on the semiconductor technology node and fabrication region, with values derived from ACT [38], iMec's device characterization [10, 34] and the Life-Cycle Assessment of Semiconductors [11]. While carbon accounting for integrated circuit can faces a lot of uncertainties due to manufacturing time and location, we include the potential uncertainty range of SoC accounting based on CarbonClarity [15] in Table 1.

Memory Devices. For DRAM, SRAM, and NAND flash, we estimate embodied carbon based on memory density and fabrication technology using published characterizations of semiconductor memory manufacturing [38, 83, 91].

CMOS Image Sensors. The embodied carbon of CMOS image sensors is influenced by pixel size, resolution, and technology node. We estimate sensor area using pixel-array dimension with an additional 0.3 mm margin for interconnects [69]. When datasheets omit the technology node, we approximate it using a pixel-pitch–to–node ratio of 20:1 [85].

3.2 Peripheral Components

Capacitors, Resistors, and Inductors We estimate the embodied carbon of passive components using a combination of package-based and weight-based methods. When available, package-resolved data from DeltaLCA [104] is used directly. When capacitor package information goes beyond the coverage of DeltaLCA, we apply emission factors from prior studies specific to capacitor chemistries [71, 78, 99]. For resistors and inductors not covered by package-specific datasets, we use generic emission factors from EcoInvent [33].

Printed Circuit Boards (PCBs) Printed circuit boards serve as the structural and electrical foundation of embedded systems. We estimate PCB embodied carbon using an emission factor of 0.006125 g CO_2e per mm^2 per 1 mm-thick layer, based on established PCB LCAs [29, 61, 104]. The total emissions scale with board area and layer count, which are obtained from schematics and vendor documentation.

Connectors, Switches, and Other Components Several common board components lack detailed peer-reviewed LCA documents and reports, including connectors, diodes, transistors, switches, thermistors, ferrite beads, timing devices, and microphones. For those component categories, we use emissions factors from industrial LCA databases EcoInvent v3.11 [33].

3.3 Power Systems

For the carbon emissions of batteries, we use an embodied footprint of 0.107 kg CO_2e per AA alkaline cell [41], and 21.8 kg CO_2e per kilogram of rechargeable Li-ion battery mass, based on EcoInvent data [33]. For solar panels, we adopt the EcoInvent value of 122 kg CO_2e per square meter for crystalline silicon photovoltaic panels.

3.4 MCU LCA Results and Discussion

Table 1 documents the absolute embodied carbon emission for 8 example COTS devices. The absolute emissions vary from 0.18 kg CO_2e to 1.14 kg CO_2e for the Adafruit Feather nRF52840 and Coral Dev Board micro, respectively. Despite this nearly 6.3× variation in embodied carbon the cost difference between the two devices is 3.2×, highlighting the distinct range difference between cost and embodied carbon.

Figure 3 shows a breakdown by component category for each of the 8 COTS devices. Generally, integrated circuits (e.g., SoCs, memory, and coprocessors) are the dominant contributors for over 42.1% of emissions for all the devices and up to 77.6% of emissions for the

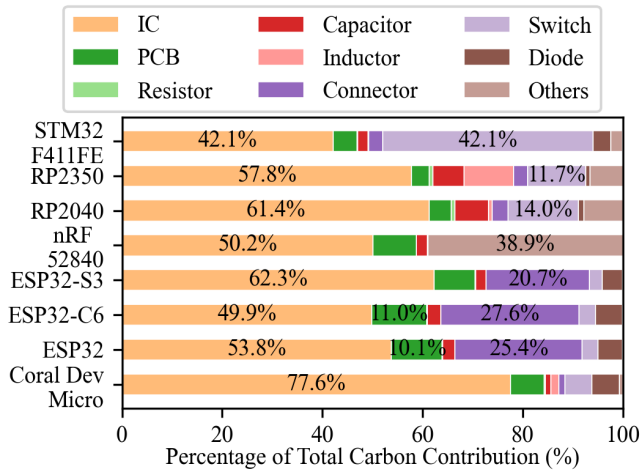

Figure 3: The carbon breakdown of various components in the minimal implementation of the processors we examined. Integrated circuit is the number one carbon emission contributor.

higher compute intensity Coral Dev Board Micro (RT1176 w/TPU). The high embodied carbon intensity of IC's reflects the the high energy intensity and material use of semiconductor fabrication [38]. For compute-lean microcontroller boards such as the STM32F411FE and nRF52840, auxiliary components such as switches and timing devices (Others) can contribute a notable share, up to around 40%, of the total footprint because of their relative size and mass on the board. The following section proposes a framework that uses the embodied carbon models to co-optimize carbon emissions of devices based on not only their fixed design time costs but also operational characteristics.

4 MicroGreen Design Framework

Designing sustainable edge systems involves navigating a large design space that spans application requirements, environmental conditions, and power system constraints. As shown in Figure 4, *MicroGreen* enables designers to consider tradeoffs between performance, carbon, and cost when designing edge devices. *MicroGreen* implements three main phases. First, given example workloads and available hardware components (e.g., MCU's, power sub-system, sensors) we conduct offline profiling to produce performance, energy, and carbon databases. Second, based on target workflows, power modes, deployment time and locations, *MicroGreen* conducts carbon-aware design-space exploration and produces a list of Pareto-optimal devices. Finally, designers can design purpose-built heterogeneous platforms to minimize environmental impacts and costs of edge device deployments. This section details each of these three key phases below.

4.1 Offline Profiling and Database Construction

MicroGreen begins with detailed offline profiling to construct per-component databases used during design-time exploration. For MCUs and COTS processors, we measure performance, power, and energy for representative ML workloads (e.g., keyword spotting [7, 84]) across diverse embedded hardware, forming the core characterization database in *MicroGreen*. We also build databases for power subsystems (solar panels, capacitors, batteries), wireless modules (BLE, WiFi), and sensors. Power components are characterized by material type, size (e.g., panel area, battery capacity, capacitor charge), and recharge or recovery time. Wireless modules capture communication range, connection latency, energy per transmission, and throughput, while sensors include pixel dimensions and physical footprint. These properties, along with embodied carbon modeling, enable mix-and-match evaluation of heterogeneous device designs and support carbon-aware design space exploration within *MicroGreen*.

To populate characterization databases, we collect fine-grained power measurements using a Nordic Power Profiler Kit II [68], isolating MCU and accelerator power by supplying a regulated 3.3V directly to the MCU rails. For each MCU, we record startup behavior, a loop of 10 inferences, and entry into the deepest available sleep state. Networking measurements similarly capture BLE and WiFi power for connection establishment and data transmission, using 244-byte BLE packets at 33ms intervals over a 0.5m link and 1460-byte WiFi packets sent to a router 4m away. Throughput and success rates are logged to compute energy-per-bit across platforms. These profiling results form the component databases that underpin *MicroGreen*'s carbon-aware design space exploration.

4.2 Carbon-aware Design Space Exploration

In step two, *MicroGreen* uses the inputs shown in yellow in Figure 4—the application workflow, deployment duration, deployment location, and power mode—to generate Pareto-optimal device configurations. The inputs configurations are used to filter viable device components (e.g., MCU's, sensors, power systems, wireless modules) based on the offline component databases. Each of the four inputs influences the design space exploration is distinct ways:

Workflow. The workflow specifies the ML models and operations to be executed and their invocation frequency. Figure 4 shows an example keyword spotting model running once every second and wirelessly transmitting results when keyword is spotted. Workflow specifications determine the space of feasible sensors, processors, and wireless modules. For instance, a vision-based workflow requires selecting a camera with adequate resolution, a wireless module with appropriate range, and processors capable of sustaining the desired frame rate.

Deployment Location. Deployment location dictates both the energy available for harvesting and the frequency of events the system must process. Solar irradiance varies by time of day, season, and geography, while event arrival rates influence processor usage and wireless transmission frequency. Based on the specified location, the framework simulates both solar energy traces and event arrival patterns to approximate real deployment conditions and estimate harvesting capability and expected energy demand.

Power Mode. We provide three power modes: solar harvesting only, battery only, and hybrid. Power mode selection matters because some deployments occur in locations where human maintenance is infeasible. A design that relies exclusively on battery replacement or recharging may therefore be unsuitable. *Solar harvesting mode* assumes energy is captured through a solar panel, stored in capacitors, and used to power the compute system. Different capacitor

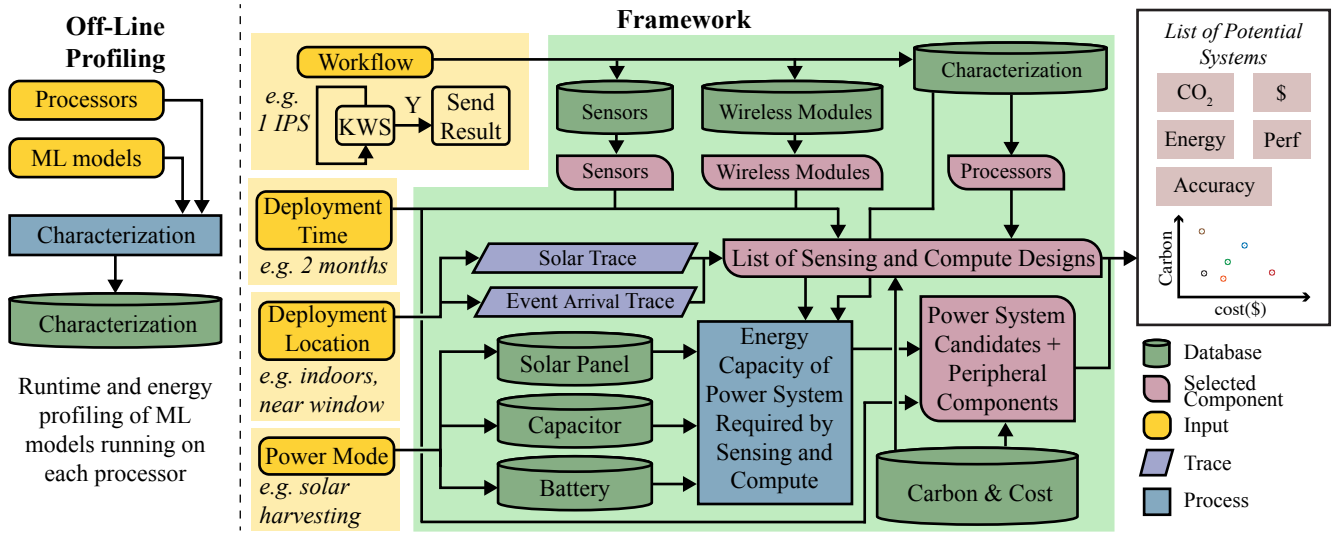

Figure 4: Overview of the *MicroGreen* framework. The system takes application requirements—including workload characteristics and deployment duration—and combines them with environmental factors such as deployment location and target power mode. It then selects compatible processors, sensors, wireless modules, and power systems, and evaluates each candidate design using a carbon modeling database. The evaluation reports estimated carbon emissions, monetary cost, and accuracy, enabling identification of Pareto-optimal configurations.

types provide different carbon footprints emission factor, and degradation patterns. Solar panel size also influences harvestable energy; we focus on crystalline silicon panels because they are the most common in IoT applications [3]. *Battery powered mode* draws all inference energy from batteries, either single-use or rechargeable. Maintenance is assumed to be available for battery replacement or charging. For rechargeable batteries, we assume the carbon intensity of electricity follows the US grid mix at 380 g CO_2e per kWh [38]. *Hybrid mode* allows users to specify a solar panel size for harvesting, with the system automatically falling back to battery power when solar energy is insufficient.

Deployment Time. Deployment duration affects sustainability in two primary ways. First, it determines total lifetime energy consumption, which affects the number of batteries or recharge cycles required. Second, it interacts with component degradation, which is particularly significant for capacitors. Most components are designed for three to five years of typical IoT lifetime, so degradation is negligible in many cases. Capacitors are an exception. Frequent charging and discharging under energy harvesting can noticeably reduce effective capacitance. Studies show that, under realistic harvesting conditions with a power failure frequency of 0.3 Hz [17, 76], supercapacitors may lose more than ten percent of their capacitance within a few days [17]. *MicroGreen* assess carbon footprint over device lifetimes by capturing both recoverable and non recoverable degradation mechanisms. Although supercapacitors can recover their nominal capacitance after a rest period [17, 76], other capacitor types such as ceramic MLCCs cannot, and must be replaced once degraded. For this study, we assume supercapacitors degrade by ten percent every one hundred fifty thousand power cycles, consistent with the five-day cycle observed in prior studies, and to recover after a two hour rest period. MLCCs are treated as non-recovering components with a degradation horizon of one month, consistent with prior characterizations [17].

For each candidate device, based on filtered device components, the framework reports total carbon emissions, monetary cost, accuracy, and performance or energy traces when needed. Designers can evaluate designs through Pareto optimal analysis across carbon emissions, cost, and accuracy to select the design best suited to their application.

4.3 Heterogeneous Device Deployments

Given the Pareto analysis based on the deployment-aware design space exploration, the final step in *MicroGreen* is enabling heterogeneous device deployments. When deploying a network of sensing nodes or edge devices, designers typically build devices with a single homogeneous configuration. Rather than deploying uniform devices, *MicroGreen* enables designers to construct heterogeneous edge device systems tailored to diverse constraints. Each device can be optimized for its available harvesting potential, and performance requirements, allowing practical per-device co-design for sustainable IoT deployments. Specifically, *MicroGreen* proposes deploying devices with the minimum carbon footprint that meet the application requirements (e.g., frame rate or duty cycle, accuracy). A use case of this heterogeneous system design is presented and evaluated in Section 6.

5 Evaluation

We evaluate *MicroGreen* by estimating the total carbon emissions of edge systems configured with different processors under varying light conditions, workloads, inference-per-second (IPS) targets, and power modes. Our goal is to identify design choices that minimize overall carbon footprint across diverse deployment scenarios. We begin by presenting our offline profiling results for the selected processors and ML workloads, then analyze high-level carbon trends under hybrid power to illustrate how irradiance and workload demands impact the carbon-optimal choice in processor. Finally, we

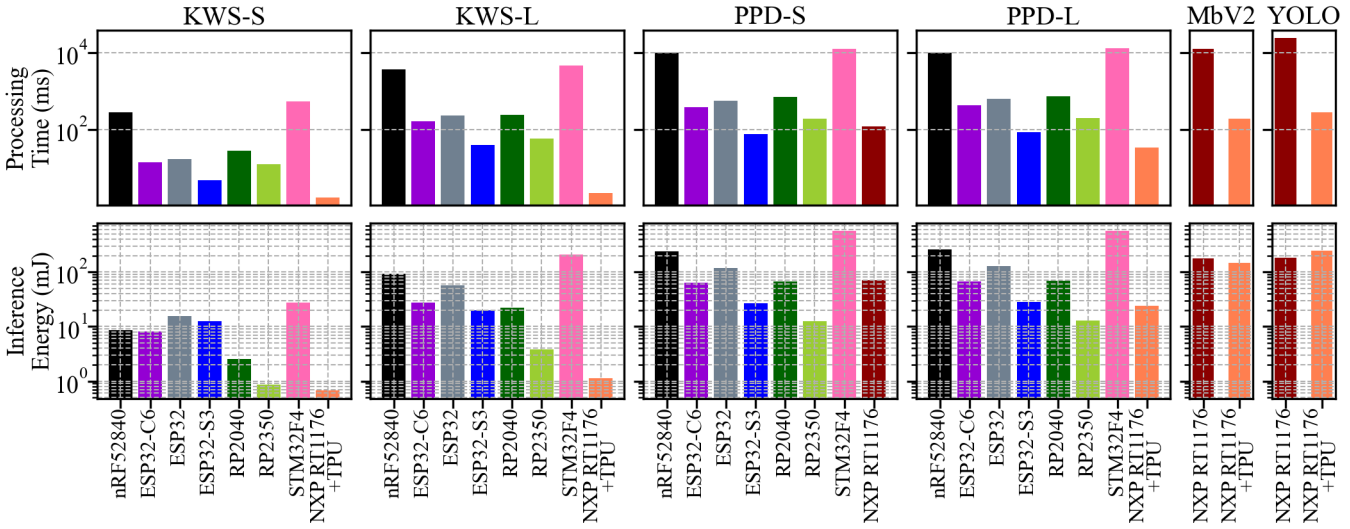

Figure 5: Runtime per inference and energy per inference for all processors running all workloads we characterized.

Table 2: Inference workloads used in our experiments, including person presence detection small (PPD-S) and large (PPD-L) and keyword spotting models—small (KWS-S) and large (KWS-L) and person detection .

Workload	Dataset	Model	Size
KWS-S [84]	Speech Commands [90]	CNN	117 KB
KWS-L [7]	Speech Commands [90]	DS-CNN	384 KB
PPD-S [84]	VWW [59]	MobileNet	294 KB
PPD-L [7]	VWW [59]	MobileNet	325 KB
PD-MbV2 [20]	COCO [19]	MobileNetV2	6.7 MB
PD-YOLO [50]	WiderPerson [101]	YOLOv8	3.3 MB

provide detailed carbon breakdowns across components, compare solar- and battery-powered deployments, and examine how deployment lifetime affects total emissions across solar, battery, and hybrid systems. The evaluation highlights the need for *MicroGreen*—a sustainability-aware IoT design space exploration framework.

5.1 Offline Profiling Results

Table 1 lists the boards used in our study. We select MCUs that span a range of cost, architectures, and performance levels and are designed for wireless or mobile computing. The ML workloads in Table 2 represent established edge benchmarks [7, 20, 50, 84], covering models from 117 KB to 6.7 MB—the upper bounds of what our devices can deploy.

For each workload, we measure inference latency and energy across all boards, as shown in Figure 5. The two largest models, PD-MbV2 and PD-YOLO, run only on the Coral Dev Board Micro due to flash constraints. PPD-S does not compile for the Coral's Edge TPU and is therefore executed on its CPU. Model accuracy values were obtained from the MLPerf Tiny [7] and Tensorflow Lite [84] benchmark's respectively. Once weights are compiled for Tensorflow Lite, inference runs single-threaded and sequentially, operate deterministically. Accuracy rates from the test set should not vary when compiled for a specific microcontroller.

Figure 5 orders processors by embodied carbon from low to high (left to right) and highlights how runtime and energy per inference vary substantially across workloads. Performance and energy efficiency are not monotonic with respect to carbon footprint: lower-carbon processors do not always deliver slower latency or higher energy, and this pattern holds across all workloads. For example, the Coral Dev Board Micro is up to 3.38×-186.6× more energy efficient than other devices for KWS-L, but it carries a significantly higher embodied carbon cost (Table 1), which underscores that sustainability depends on more than raw energy efficiency alone.

5.2 Overall Carbon Trends

We first evaluate how irradiance and workflow requirements (IPS) influence total carbon emissions in hybrid-powered deployments, where systems use both a battery and solar harvester with the solar-panel area capped at 611 cm^2 [3]. Figure 6 presents the ranking of total carbon emissions for systems built around different processors, evaluated across a range of IPS values and three irradiance levels. A lower rank indicates lower total emissions, with Rank 1 representing the most carbon-efficient design.

The results demonstrate that the optimal design varies significantly with both the workload and environmental conditions. We derive several key insights from our framework based on the processors and workloads evaluated. First, light availability significantly influences the relative carbon efficiency of different processors. Processors that trade off energy efficiency for lower embodied carbon benefit in high-irradiance scenarios, as greater solar input compensates for their higher operational energy demands, such as RP2040 for KWS-S in Figure 6.

Second, the computational capability of each processor constrains the achievable inference rate. Due to differences in processing time per inference, only more compute-intensive processors can sustain higher inference-per-second (IPS) targets. For instance, for PPD-L, only the NXP RT1176 paired with the Edge TPU is able to achieve inference rates beyond 11 IPS.

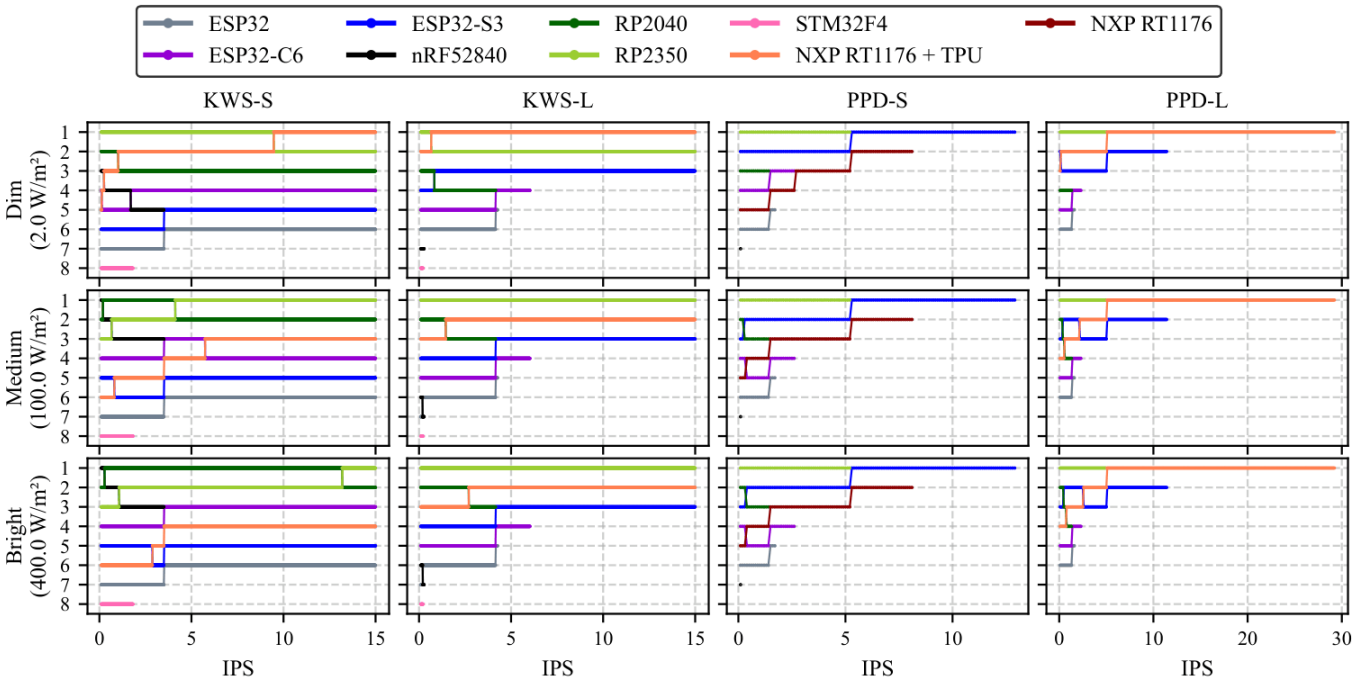

Figure 6: Total carbon emissions of edge systems designed with specific processors, ranked across varying inference-per-second (IPS) settings under three irradiance levels (Rank 1 = lowest carbon emission = best). Carbon emissions are computed under a hybrid power mode, with the solar-panel area capped at 611 cm^2 [3]. Results show that the carbon-optimal processor varies with irradiance, and achievable IPS is limited by each device's per-inference processing time, rendering some processors infeasible at higher IPS targets.

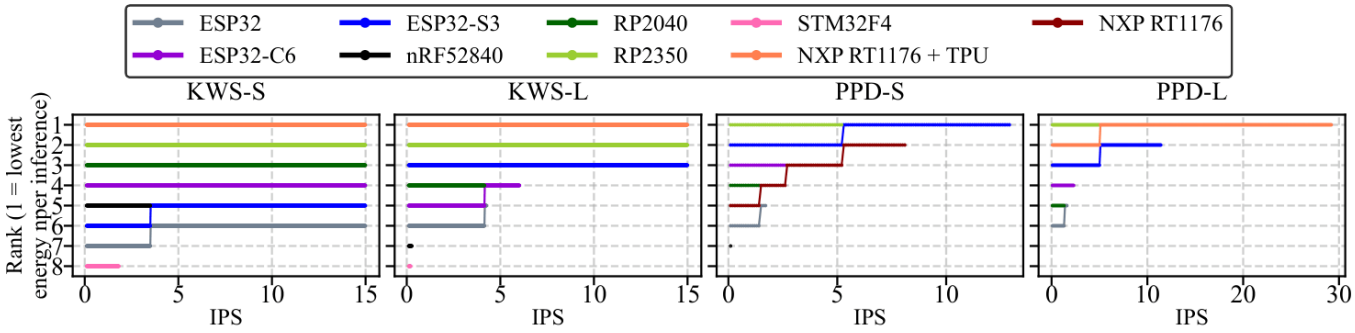

Figure 7: Energy-efficiency ranking of different microcontroller units (MCUs) across varying inference-per-second (IPS) rates under distinct workload profiles. The processor configuration that is optimal with respect to energy consumption does not necessarily coincide with the configuration that minimizes carbon emissions, as illustrated in Figure 6. IPS and ambient energy availability can differentially influence the total carbon footprint of the system in ways that extend beyond what is captured by energy-efficiency metrics alone.

Third, the presence of an Edge TPU accelerator introduces a substantial tradeoffs in performance and carbon-efficiency. In the PPD-S scenario, the model cannot be compiled for the Edge TPU, and therefore runs solely on the NXP RT1176 CPU, making it less competitive than the ESP32-S3. In contrast, PPD-L, which can be offloaded to the Edge TPU on the Coral Dev Board Micro, benefits from substantial acceleration and improved energy efficiency. However, the sustainability gains of using the accelerator only materialize under high inference rates or low–to–medium irradiance levels, where its high energy efficiency offsets its much larger embodied carbon cost (Table 1). This highlights a key trade-off: while the accelerator delivers clear performance and energy advantages, its sustainability benefits depend on deployment conditions rather than being universally dominant.

Last but not the least, we compare the carbon-emission-optimal configurations in Figure 6 against the energy-efficiency-optimal configurations in Figure 7 to demonstrate that systems built with the most energy-efficient processors do not necessarily yield the lowest carbon emissions. For example, under the KWS-S workload,

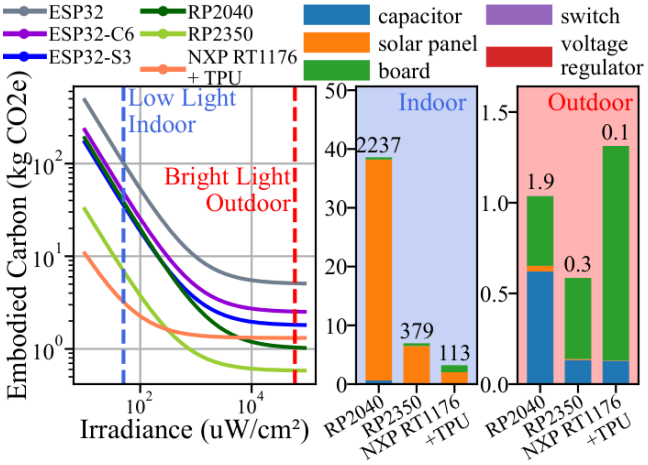

Figure 8: Carbon emissions of different boards operating under various light conditions for solar-harvesting-only deployments running KWS-L at 1 inference per second. The number on the bars represent the size the solar pannel needed in cm².

the most energy-efficient solution, NXPRT1176 with EdgeTPU, is carbon-optimal only in dim lighting conditions above 9 IPS, whereas the third most energy-efficient solution, RP2040, is the carbon-optimal choice under bright light conditions at below 13 IPS. These results highlight that carbon emissions must be evaluated more holistically, accounting for deployment environment (e.g., ambient energy availability) and application requirements (e.g., duty cycle).

In the following evaluation section, we provide a detailed breakdown of embodied carbon to further analyze these observations illustrated in Figure 6.

5.3 Detailed Carbon Insights

We present the estimated carbon emissions analysis for the solar-powered deployment in Figure 8 for KWS-L running at 1 IPS for deployment lifetime of a year. The graph on the left displays total carbon emissions across different irradiance levels for each board. The nRF52840 and STM32F4 are excluded from this analysis because they do not meet the performance requirements of 1 IPS. The graph illustrates that irradiance density significantly influences the optimal design choice. For instance, while the NXP RT1176 w/ TPU has the lowest emissions (3.20 kg CO_2e) under low-light indoor conditions (50 µW/cm²), it drops to the third-best option (1.31 kg CO_2e) under bright outdoor irradiance of 60000 µW/cm², where the RP2350 (0.59 kg CO_2e) emerges as the most carbon-efficient configuration.

.

Additionally, Figure 8 provides a component-level breakdown of the three processors with the lowest emissions in both low-light and bright-light scenarios. In dim indoor conditions, the embodied cost of the solar panel dominates total emissions, with the required panel area increasing to approximately 113–2237 cm² depending on the processors used, overshadowing contributions from the board, capacitors, and other passive components. Conversely, under bright outdoor conditions, the required panel area shrinks to just 0.1–1.9

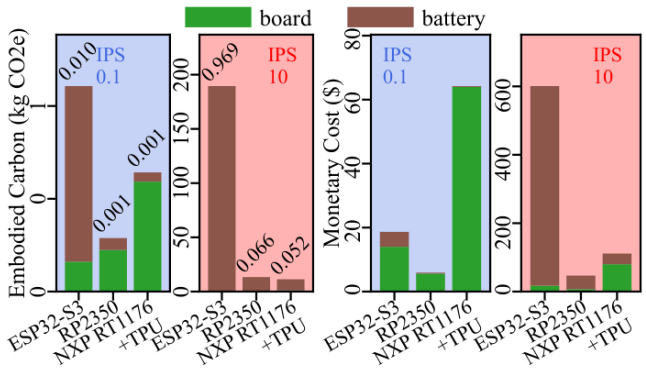

Figure 9: Total Carbon emission and monetary cost for battery powered scenario of KWS-S with varying IPS under five-year deployment. The number on the embodied carbon bar is the needed number of AA batteries per day.

cm², causing board level components, primarily integrated circuits and capacitors, to contribute up to 87% of total emissions.

Although the NXP RT1176 with TPU is the most energy-efficient processor for KWS-L (see Figure 5), it is only carbon-optimal for indoor use due to its smaller solar panel requirement. In contrast, under higher/outdoor irradiance, its total carbon footprint becomes 2.22× higher than that of the lowest carbon emission solution provided by RP2350. **Takeaway 1: Ambient energy availability strongly shapes carbon-optimal IoT design: under low-light conditions, devices require larger solar panels, making energy-efficient processors more favorable, whereas in the bright-light environments, leaner processors with smaller panels become the more sustainable choice.**

Figure 9 illustrates the carbon emissions and monetary costs of battery-powered deployments running the KWS-S model at two IPS rates: 0.1 and 10. Only three processors, ESP32-S3, RP2350, and NXP RT1176 with Edge TPU, can sustain 10 IPS, and thus they are selected for detailed breakdown.

Without solar harvesting, total emissions are dominated by the embodied carbon of the board and the number of AA batteries required over a five-year deployment, and actual maintenance for battery replacement may introduce additional carbon burdens beyond those captured here. The numbers shown above each bar represent the daily AA battery requirements (e.g., 0.01 for the ESP32-S3 at an IPS of 0.1). Processors with higher per-inference energy consumption, such as the ESP32-S3 (see Figure 5), require more frequent battery replacements, which substantially increases their carbon footprint.

At lower IPS rates, the RP2350 stands out as the most carbon-efficient option due to its low embodied cost and modest energy consumption. However, at an IPS of 10, the NXP RT1176 with TPU becomes the carbon-optimal choice because its higher energy efficiency reduces the total number of required battery replacements by 26.9% compared to the RP2350. Nonetheless, this improvement does not translate to cost, as the RP2350 remains the least expensive option at both IPS levels. As shown in Figure 9, carbon emissions and monetary cost diverge across inference rates. In the blue panel (low IPS), the RP2350 has the lowest carbon emissions due to requiring the fewest AA batteries per day, yet its cost is similar to

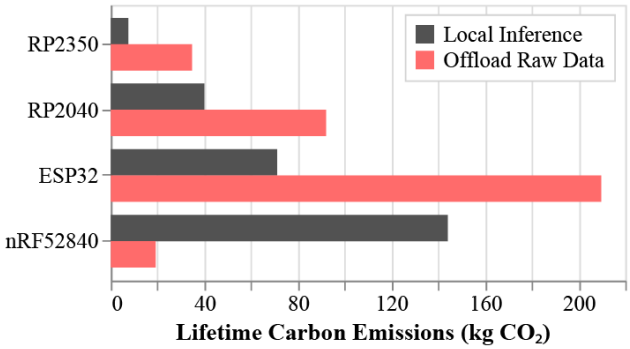

Figure 10: Lifetime carbon emissions for local inference versus data offloading over BLE for PPD-S across different MCUs.

higher-emission boards such as the ESP32-S3. In the red panel (high IPS), the NXP RT1176 with TPU becomes the carbon-optimal design because of its higher energy efficiency, despite its larger embodied footprint and higher price. These trends illustrate that market cost does not track carbon impact, and both must be jointly considered during design. **Takeaway 2: Application-level requirements such as inference rate constrain viable processors and shift the carbon-optimal choice, while cost and carbon efficiency often diverge.**

Beyond local processing efficiency, networked deployments must also consider the carbon costs of data transmission. Networked and mobile edge devices send data back to a host system. *MicroGreen* considers the combination of inference and transmission costs to determine whether it is better to perform inference locally or offload the raw sensor data for remote processing. Figure 10 compares lifetime carbon emissions for two inference strategies applied to the PPD-S workload at 1 IPS over BLE: (1) local inference with transmission of inference results and (2) offloading raw input data for remote inference. For most processors local inference is more carbon efficient as the energy required to transmit raw sensor data exceeds the cost of on-device processing. However, for less capable processors like the nRF52840, offloading and running inference remotely may be more carbon-efficient. This would depend on the carbon cost and grid intensity of remote inference. **Takeaway 3: The optimal choice between local inference and data offloading depends on processor efficiency; transmission costs do not always favor edge processing.**

Figure 11 shows how deployment lifetime affects the carbon emissions for three types of power mode: solar-only, battery-only, and hybrid systems. The total carbon emissions for the solar-only deployment remain constant throughout the entire lifetime since the system operates without requiring any external energy or component replacement.

In contrast, battery-powered systems necessitate periodic battery replacements or recharging, leading to an accumulation of carbon emission over time. Initially, battery-powered deployments have the lowest carbon footprint (ranging from 1.6 kg to 7.5 kg) two months into the deployment, making them the most carbon-efficient choice for short-term use. However, their emissions eventually exceed those of solar-powered systems in the long run. For example, it takes

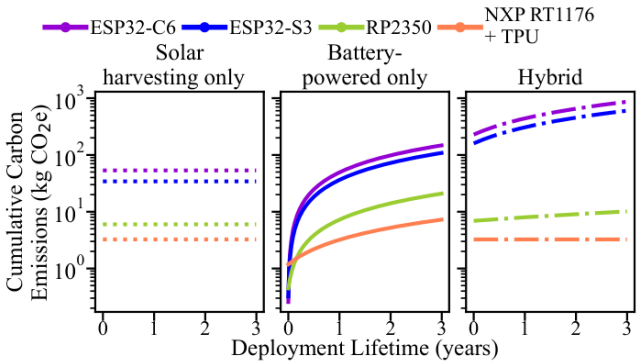

Figure 11: The total carbon emission versus deployment lifetime across various power mode with processors running KWS-L at frequency of 6 IPS under 400 μW/cm^2 irradiance. Hybrid power mode has solar panel size capped at 275 cm^2.

ESP32-C6 1 year of battery-only operation for its total emissions to surpass those of the solar-powered configuration.

Hybrid systems occupy a middle ground: they utilize harvested energy but still depend on a battery, causing their embodied carbon to reflect contributions from both subsystems. At 5 year deployment time mark, their cumulative carbon emissions are 0.29× to 1.26× of battery-powered systems and 1× to 4.27× of solar-powered systems. **Takeaway 4: Battery-powered solution offers the lowest carbon emission if the deployment time is low. Solar harvesting is more sustainable in the long run.**

6 Case Study: Central Park Visitor Counter

In this section, we demonstrate how *MicroGreen* can be applied to real edge device prototyping under diverse deployment conditions, and how it guides designers toward lower-carbon deployment choices.

Our prototype is a low-cost visitor-counting system designed for Central Park in New York City, which needs an affordable and reliable alternative to existing commercial solutions that are both expensive [27] and, according to the parks team, often significantly under counts. Accurate visitor data is crucial for guiding park resource allocation and maintenance. The prototype performs person detection and tracking directly on the edge, with visitor statistics stored locally or transmitted periodically over wireless links.

Because entrances across the park experience widely different light availability and visitor-flow patterns, a single homogeneous design is insufficient. In the case study that follows, we show how applying *MicroGreen* enables heterogeneous deployment strategies that reduce total carbon emissions across three entrance locations by 47.72%.

6.1 Prototype

Figure 12 shows the prototype developed for the visitor-counting application. The device integrates a Coral Dev Board Micro microcontroller with an onboard camera for local processing, paired with a REYAX RYLR998 LoRa wireless module for low-power communication on a hourly basis reporting the visitor counting results.

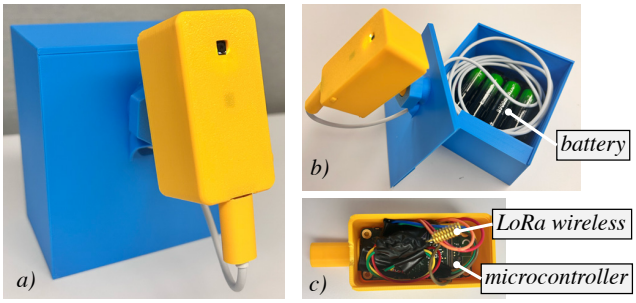

Figure 12: Prototype of the New York City park visitor counting device.

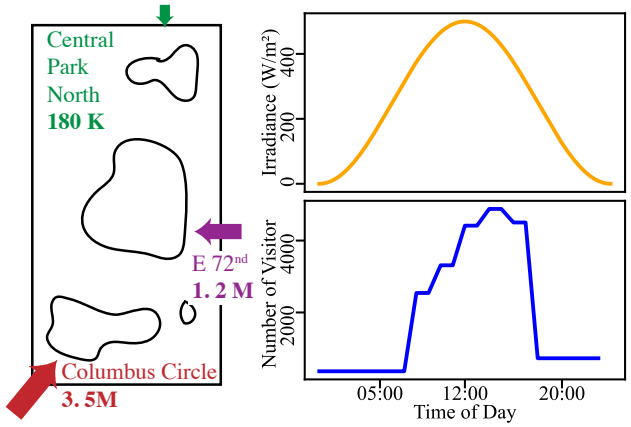

Figure 13: Left: annual visitor counts for different Central Park entrances. Top right: example solar trace for an unshaded location on a sunny day. Bottom right: example of daily visitor flow trace at Columbus Circle.

The system is powered by four 9900 mAh 18650 Li-ion rechargeable batteries and enclosed in a weather-resistant case designed for installation at park entrance gates.

The deployed workflow executes PPD-L once per second. Upon detecting a person, the system captures and stores nine consecutive frames, which YOLOv8 processes opportunistically when the microcontroller becomes free. Together with ByteTrack [103], this pipeline achieves 89.2% counting accuracy: in a 4-minute test clip collected at Central Park, the system detected 31 entering and 27 exiting visitors compared to ground truth counts of 34 and 31, respectively.

We use this battery-powered prototype as the baseline "battery-only" deployment configuration in our later analysis.

6.2 Heterogeneous Deployment

Visitor traffic and light availability vary substantially across Central Park entrances. We model three representative locations: *Central Park North*, *E 72nd Street*, and *Columbus Circle*. Annual visitor totals for these entrances, derived from the Report on Public Use of Central Park [1], are shown in the left panel of Figure 13. Daily visitor distribution is reconstructed from reported temporal trends; the resulting trace for Columbus Circle is illustrated in the bottom-right panel.

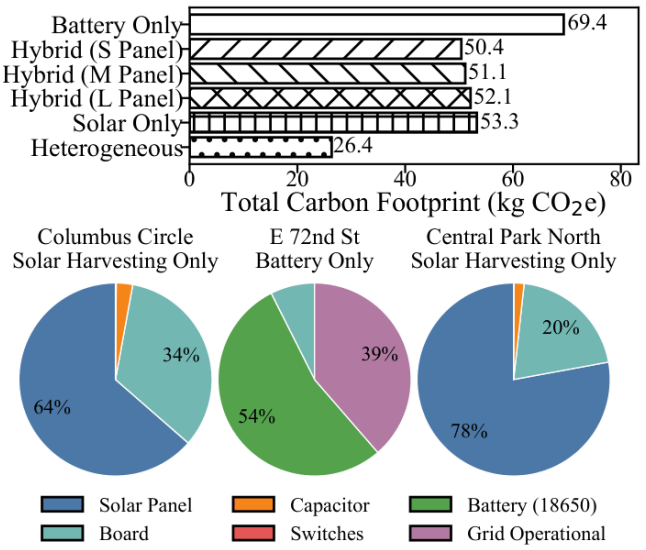

Figure 14: Carbon emissions for three entrances over one year by deployment strategy.

To simulate solar availability, we follow the EnHANTs solar trace [37] and solar irradiance data [70, 86], specifying location-dependent peak irradiance levels: 500 W/m^2 at Columbus Circle, 100 W/m^2 at Central Park North, and 50 W/m^2 at E 72nd Street. Irradiance drops to 0 W/m^2 at night. An example simulated trace is shown in the upper-right panel of Figure 13.

We compare several deployment strategies:

- **Battery-only:** All locations use the same hardware configuration, with battery capacity provisioned to satisfy the worst-case energy demand.
- **Hybrid (fixed solar panel sizes):** Devices combine a battery and a fixed-size solar panel. We evaluate panels of 74.5 cm^2, 304 cm^2, and 611 cm^2.
- **Solar-only:** Solar panel area is sized per location to fully meet energy needs. Supercapacitors [21] are provisioned onboard to provide energy buffering.
- **Heterogeneous deployment:** Our framework selects the optimal deployment mode—battery-only, one of the hybrid configurations, or solar-only—for each location based on its solar availability and visitor-flow pattern.

Figure 14 shows the carbon emissions for each strategy over a four-year deployment. The heterogeneous deployment yields the lowest total emissions, reducing carbon footprint by 47.72% relative to the next-best strategy (hybrid with a small solar panel).

As shown in Figure 3, the optimal deployment differs by location. E 72nd Street receives limited sunlight, making the battery-only configuration preferable; most emissions stem from battery provisioning and periodic grid charging. In contrast, the other two entrances receive enough sunlight such that panel and board manufacturing dominate total emissions, with only a small contribution from supercapacitors.

Overall, these results highlight that heterogeneous deployment, rather than a uniform, one-size-fits-all design, substantially reduces embodied and operational carbon emissions while meeting sensing and communication requirements.

7 Discussion and Conclusion

Going forward, optimizing the carbon footprint of mobile and IoT systems requires consideration of the entire stack and deployment context. Unlike cloud environments, mobile and edge devices face unpredictable energy availability, heterogeneous hardware, and user-driven lifecycles. These factors introduce unique sustainability challenges and present opportunities for carbon-aware design.

Applications and Workloads. Continuous-inference workloads put a strain on both energy and carbon budgets. Applications must adapt their inference frequency, batching, and offloading based on carbon constraints, revealing trade-offs between accuracy, latency, and carbon emissions to enable context-aware behavior.

Runtime Systems. Mobility and variable harvested power require schedulers that treat carbon emissions as a primary metric. There is an ongoing need for unified runtimes that can jointly optimize latency, energy, and carbon across batching, adaptive inference, and edge-cloud coordination.

Hardware Design and Provisioning. IoT deployments rely on diverse sensors, microcontrollers, and accelerators, yet emissions data remains fragmented. Carbon-aware provisioning tools integrated into design environments (e.g., KiCad, eCAD) should surface embodied and operational trade-offs to guide hardware selection. Furthermore, Low-MOQ PCB fabrication facilitates deployment-specific designs, but specialized accelerators may face short lifetimes as ML workloads evolve. There is a need for flexible accelerator designs and automated heterogeneous hardware workflows that incorporate carbon metrics for sustainable hardware scaling.

Lifecycle and End-of-Life Management. Long-term sustainability relies on reuse, refurbishment, and recycling, especially given the global variation in IoT deployments. Systems must extend device lifetimes and support modular replacements to address e-waste and enable carbon-aware lifecycle management.

Deployment Conditions and Mobility. Mobility affects energy availability, workload dynamics, and carbon amortization, making sustainability a constantly shifting target. Research avenues that explore mobility-aware provisioning, adaptive runtime techniques, and cross-infrastructure carbon optimization are crucial and remain underexplored.

Extending to Communication and Network Emissions. While our evaluation focuses on low-power communication (e.g., BLE), MicroGreen can be extended to higher-power wireless technologies such as LTE and 5G. Our results show that communication overhead is negligible for low-power protocols, but can become significant in cellular settings, introducing a trade-off between local inference and data offloading. MicroGreen can capture this by modeling communication as a system component parameterized by data volume, transmission frequency, and protocol choice, and incorporating network emissions as part of the deployment environment.

In this paper we introduce, *MicroGreen*, a design-time, carbon-aware provisioning framework that jointly accounts for embodied, operational, and workload-driven factors. Our evaluation shows that meaningful carbon reductions are achievable when designers consider hardware, workloads, and deployment conditions together rather than optimizing energy efficiency alone. This work provides an initial foundation for developing tools, datasets, and workflows that embed carbon awareness throughout the mobile and IoT computing stack. In a real deployment in New York City parks, heterogeneous, location-aware provisioning reduced emissions by 47.26% over a homogeneous baseline, demonstrating the practical value of carbon-aware design.

8 Acknowledgment

We thank Ariel Goldner and Shixue Bai for their assistance with the LCA modeling, and Raspberry Pi for providing the bills of materials. We also thank the New York City Parks Department and the Central Park Conservancy for supplying essential data and feedback on the visitor counter prototype, and Dani Fael and Gangming Zhang for their contributions to visitor counter device prototyping. We are grateful for in-kind support from Google and Amazon. This research was partially supported by the National Science Foundation under awards numbers CCF-2324860, CNS-2326608 and CCF-2324861. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or other supporters.

References

[1] 2011. *Report on the Public Use of Central Park.* Technical Report. Central Park Conservancy.

[2] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Kiwan Maeng, Udit Gupta, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. 2023. Carbon Explorer: A Holistic Framework for Designing Carbon Aware Datacenters. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2* (Vancouver, BC, Canada) *(ASPLOS 2023)*. Association for Computing Machinery, New York, NY, USA, 118–132. doi:10.1145/3575693.3575754

[3] Adafruit. 2024. Adafruit 5369 – Product Page. https://www.adafruit.com/product/5369. Accessed: 2025-11-19.

[4] Md Abdulla Al Mamun and Mehmet Rasit Yuce. 2019. Sensors and systems for wearable environmental monitoring toward IoT-enabled applications: A review. *IEEE Sensors Journal* 19, 18 (2019), 7771–7788.

[5] Abu Bakar, Rishabh Goel, Jasper De Winkel, Jason Huang, Saad Ahmed, Bashima Islam, Przemysław Pawełczak, Kasım Sinan Yıldırım, and Josiah Hester. 2022. 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*. 207–221.

[6] Abu Bakar, Alexander G Ross, Kasim Sinan Yildirim, and Josiah Hester. 2021. Rehash: A flexible, developer focused, heuristic adaptation platform for intermittently powered computing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–42.

[7] Colby Banbury, Vijay Janapa Reddi, Peter Torelli, Jeremy Holleman, Nat Jeffries, Csaba Kiraly, Pietro Montino, David Kanter, Sebastian Ahmed, Danilo Pau, Urmish Thakker, Antonio Torrini, Peter Warden, Jay Cordaro, Giuseppe Di Guglielmo, Javier Duarte, Stephen Gibellini, Videet Parekh, Honson Tran, Nhan Tran, Niu Wenxu, and Xu Xuesong. 2021. MLPerf Tiny Benchmark. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks* (2021). Git tag v1.1 (released Jun 27, 2023). Code: https://github.com/mlcommons/tiny/tree/v1.1.

[8] Noman Bashir, David Irwin, and Prashant Shenoy. 2023. On the Promise and Pitfalls of Optimizing Embodied Carbon. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems (HotCarbon '23)*. ACM, Boston, MA, USA. doi:10.1145/3604930.3605710

[9] Tereza Bicalho, Ildo Sauer, Alexandre Rambaud, and Yulia Altukhova. 2017. LCA data quality: A management science perspective. *Journal of Cleaner Production* 156 (2017), 888–898.

[10] L. Boakes, M. Garcia Bardon, V. Schellekens, I-Y. Liu, B. Vanhouche, G. Mirabelli, F. Sebaai, L. Van Winckel, E. Gallagher, C. Rolin, and L.-A. Ragnarsson. 2023. Cradle-to-gate Life Cycle Assessment of CMOS Logic Technologies. In *2023 International Electron Devices Meeting (IEDM)*. 1–4. doi:10.1109/IEDM45741.2023.10413725

[11] Sarah B. Boyd. 2009. *Life-cycle Assessment of Semiconductors.* Ph. D. Dissertation. UNIVERSITY OF CALIFORNIA, BERKELEY.

[12] Luca Caronti, Khakim Akhunov, Matteo Nardello, Kasım Sinan Yıldırım, and Davide Brunelli. 2023. Fine-grained hardware acceleration for efficient batteryless intermittent inference on the edge. *ACM Transactions on Embedded Computing Systems* 22, 5 (2023), 1–19.

[13] Meredith A Case, Holland A Burwick, Kevin G Volpp, and Mitesh S Patel. 2015. Accuracy of smartphone applications and wearable devices for tracking physical activity data. *Jama* 313, 6 (2015), 625–626.

[14] Xuesi Chen, Ariel Goldner, Eren Yildiz, Ilan Mandel, Tingyu Cheng, Josiah Hester, and Udit Gupta. 2025. From Component to System: Rethinking Edge Computing Design through a Carbon-Aware Lens. *SIGENERGY Energy Inform. Rev.* 5, 2 (Aug. 2025), 125–131. doi:10.1145/3757892.3757910

[15] Xuesi Chen, Leo Han, Anvita Bhagavathula, and Udit Gupta. 2025. CarbonClarity: Understanding and Addressing Uncertainty in Embodied Carbon for Sustainable Computing. In *2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. 1–9. doi:10.1109/ICCAD66269.2025.11240799

[16] Christopher C Cheung, Andrew D Krahn, and Jason G Andrade. 2018. The emerging role of wearable technologies in detection of arrhythmia. *Canadian Journal of Cardiology* 34, 8 (2018), 1083–1087.

[17] Jongouk Choi, Hyunwoo Joe, and Changhee Jung. 2022. Capos: Capacitor error resilience for energy harvesting systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 11 (2022), 4539–4550.

[18] Sung Chun and Chan-Su Lee. 2013. Applications of human motion tracking: Smart lighting control. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 387–392.

[19] COCO Consortium. [n. d.]. COCO: Common Objects in Context. https://cocodataset.org/#home Accessed: 16 November 2025.

[20] Coral. 2025. Models – Object Detection. https://www.coral.ai/models/object-detection/. Accessed: 16 November 2025.

[21] Yageo Corporation. [n. d.]. KEM S6012 FM Data Sheet. Accessed: YYYY-MM-DD.

[22] Leonardo Lucio Custode, Pietro Farina, Eren Yildiz, Renan Beran Kilic, Kasim Sinan Yildirim, and Giovanni Iacca. 2024. Fast-Inf: ultra-fast embedded intelligence on the batteryless edge. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*. 239–252.

[23] Madeleine IG Daepp, Alex Cabral, Vaishnavi Ranganathan, Vikram Iyer, Scott Counts, Paul Johns, Asta Roseway, Charlie Catlett, Gavin Jancke, Darren Gehring, et al. 2022. Eclipse: An end-to-end platform for low-cost, hyperlocal environmental sensing in cities. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 28–40.

[24] Madeleine I. G. Daepp, Alex Cabral, Vaishnavi Ranganathan, Vikram Iyer, Scott Counts, Paul Johns, Asta Roseway, Charlie Catlett, Gavin Jancke, Darren Gehring, Chuck Needham, Curtis von Veh, Tracy Tran, Lex Story, Gabriele D'Amone, and Bichlien H Nguyen. 2022. Eclipse: An End-to-End Platform for Low-Cost, Hyperlocal Environmental Sensing in Cities. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 28–40. doi:10.1109/IPSN54338.2022.00010

[25] Lara Eguren David Sánchez, Sarah-Jane Baur. 2024. *Life Cycle Assessment of the Fairphone 5*. Technical Report. Berlin: Fraunhofer IZM.

[26] Harsh Desai, Matteo Nardello, Davide Brunelli, and Brandon Lucia. 2022. Camaroptera: A Long-range Image Sensor with Local Inference for Remote Sensing Applications. *ACM Trans. Embed. Comput. Syst.* 21, 3, Article 32 (May 2022), 25 pages. doi:10.1145/3510850

[27] Eco-Counter. 2026. Pyro Box Evo: Counting Solutions. https://www.eco-counter.com/solutions/counting-solutions/pyro-box-evo. Accessed: 2026-04-13.

[28] Pietro Farina, Subrata Biswas, Eren Yıldız, Khakim Akhunov, Saad Ahmed, Bashima Islam, and Kasım Sinan Yıldırım. 2024. Memory-efficient energy-adaptive inference of pre-trained models on batteryless embedded systems. *arXiv preprint arXiv:2405.10426* (2024).

[29] Csaba Farkas, László Galá, András Csiszár, Vincent Grennerat, Pierre-Olivier Jeannin, Pascal Xavier, Dániel Rigler, Olivér Krammer, Zbyněk Plachý, Karel Dušek, Róbert Kovács, Anna Éva Fehér, and Attila Géczy. 2024. Sustainable printed circuit board substrates based on flame-retarded PLA/flax composites to reduce environmental load of electronics: Quality, reliability, degradation and application tests. *Sustainable Materials and Technologies* 40 (2024). https://doi.org/10.1016/j.susmat.2024.e00902

[30] Farbin Fayza, Cansu Demirkiran, Satyavolu Papa Rao, Darius Bunandar, Udit Gupta, and Ajay Joshi. 2025. EPiCarbon: A Carbon Modeling Tool for Electro-Photonic Accelerators. In *2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. 1–9. doi:10.1109/ICCAD66269.2025.11240809

[31] Matthias Finkbeiner, Atsushi Inaba, Reginald Tan, Kim Christiansen, and Hans-Jürgen Klüppel. 2006. The new international standards for life cycle assessment: ISO 14040 and ISO 14044. *The international journal of life cycle assessment* 11, 2 (2006), 80–85.

[32] Charlotte Freitag, Mike Berners-Lee, Kelly Widdicks, Bran Knowles, Gordon Blair, and Adrian Friday. 2021. The climate impact of ICT: A review of estimates, trends and regulations. arXiv:2102.02622 [physics.soc-ph] https://arxiv.org/abs/2102.02622

[33] Rolf Frischknecht, Niels Jungbluth, Hans-Jörg Althaus, Roland Hischier, Gabor Doka, Roberto Dones, Thomas Heck, Stefanie Hellweg, Gregor Wernet, and Thomas Nemecek. 2007. *Overview and Methodology: Ecoinvent Report No. 1.* Data v2.0. Swiss Centre for Life Cycle Inventories, Dübendorf, Switzerland.

[34] M. Garcia Bardon, P. Wuytens, L.-A. Ragnarsson, G. Mirabelli, D. Jang, G. Willems, A. Mallik, A. Spessot, J. Ryckaert, and B. Parvais. 2020. DTCO including Sustainability: Power-Performance-Area-Cost-Environmental score (PPACE) Analysis for Logic Technologies. In *2020 IEEE International Electron Devices Meeting (IEDM)*. 41.4.1–41.4.4. doi:10.1109/IEDM13553.2020.9372004

[35] Kai Geissdoerfer and Marco Zimmerling. 2024. Riotee: An Open-source Hardware and Software Platform for the Battery-free Internet of Things. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*. 198–210.

[36] Google LLC. 2023. *Fitbit Charge 6 Product Environmental Report*. Product Environmental Report. Google LLC. https://sustainability.google/reports/fitbit-charge-6-product-enviromental-report/ Model G3MP5.

[37] Maria Gorlatova, Michael Zapas, Enlin Xu, Matthias Bahlke, Ioannis (John) Kymissis, and Gil Zussman. 2022. CRAWDAD columbia/enhants. doi:10.15783/C77P47

[38] Udit Gupta, Mariam Elgamal, Gage Hills, Gu-Yeon Wei, Hsien-Hsin S Lee, David Brooks, and Carole-Jean Wu. 2022. ACT: Designing sustainable computer systems with an architectural carbon modeling tool. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*. 784–799.

[39] Udit Gupta, Young Geun Kim, Sylvia Lee, Jordan Tse, Hsien-Hsin S. Lee, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. 2021. Chasing Carbon: The Elusive Environmental Footprint of Computing. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 854–867. doi:10.1109/HPCA51647.2021.00076

[40] Amir HajiRassouliha, Andrew J Taberner, Martyn P Nash, and Poul MF Nielsen. 2018. Suitability of recent hardware accelerators (DSPs, FPGAs, and GPUs) for computer vision and image processing algorithms. *Signal Processing: Image Communication* 68 (2018), 101–119.

[41] Ramsey Hamade, Raghid Al Ayache, Makram Bou Ghanem, Sleiman El Masri, and Ali Ammouri. 2020. Life cycle analysis of AA alkaline batteries. *Procedia Manufacturing* 43 (2020), 415–422.

[42] Leo Han, Jash Kakadia, Benjamin C. Lee, and Udit Gupta. 2025. Fair-CO2: Fair Attribution for Cloud Carbon Emissions. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture (ISCA '25)*. Association for Computing Machinery, New York, NY, USA, 646–663. doi:10.1145/3695053.3731023

[43] Nathan Beckmann Brandon Lucia Harah Desai, Sara McAllister. 2024. Towards Understanding the Carbon Impact in End-to-end Sensing Pipelines. In *HotEthics*.

[44] Josiah Hester, Travis Peters, Tianlong Yun, Ronald Peterson, Joseph Skinner, Bhargav Golla, Kevin Storer, Steven Hearndon, Kevin Freeman, Sarah Lord, et al. 2016. Amulet: An energy-efficient, multi-application wearable platform. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. 216–229.

[45] ISO. 2006. ISO 14044: 2006-life cycle assessment-requirements and guidelines. *International Standard, Geneve, Switzerland* (2006).

[46] I Iso-Norm. 2006. Environmental management—life cycle assessment—principles and framework ISO 14040: 2006. *ISO: Geneva, Switzerland* 157 (2006).

[47] Senthil Kumar Janahan, MRM Veeramanickam, S Arun, Kumar Narayanan, R Anandan, S Javed Parvez, et al. 2018. IoT based smart traffic signal monitoring system using vehicles counts. *International journal of engineering & technology* 7, 2.21 (2018), 309–312.

[48] Nitthilan Kannappan Jayakodi, Anwesha Chatterjee, Wonje Choi, Janardhan Rao Doppa, and Partha Pratim Pande. 2018. Trading-off accuracy and energy of deep inference on embedded systems: A co-design approach. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 11 (2018), 2881–2893.

[49] Shixin Ji, Zhuoping Yang, Xingzhen Chen, Stephen Cahoon, Jingtong Hu, Yiyu Shi, Alex K Jones, and Peipei Zhou. 2024. SCARIF: Towards Carbon Modeling of Cloud Servers with Accelerators. In *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 496–501.

[50] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. *Ultralytics YOLOv8*. https://github.com/ultralytics/ultralytics

[51] Kimon Karras, Evangelos Pallis, George Mastorakis, Yannis Nikoloudakis, Jordi Mongay Batalla, Constandinos X Mavromoustakis, and Evangelos Markakis. 2020. A hardware acceleration platform for AI-based inference at the edge. *Circuits, Systems, and Signal Processing* 39, 2 (2020), 1059–1070.

[52] Mostafa Haghi Kashani, Mona Madanipour, Mohammad Nikravan, Parvaneh Asghari, and Ebrahim Mahdipour. 2021. A systematic review of IoT in healthcare: Applications, techniques, and trends. *Journal of Network and Computer Applications* 192 (2021), 103164.

[53] Navroop Kaur and Sandeep K Sood. 2015. An energy-efficient architecture for the Internet of Things (IoT). *IEEE Systems Journal* 11, 2 (2015), 796–805.

[54] Bahareh Khabbazan and Sattar Mirzakuchaki. 2019. Design and implementation of a low-power, embedded CNN accelerator on a low-end FPGA. In *2019 22nd*

Euromicro Conference on Digital System Design (DSD). IEEE, 647–650.

[55] Argonne National Laboratory and U.S. Department of Energy. [n. d.]. R&D GREET Life-Cycle Assessment Model. https://www.energy.gov/eere/rd-greet-life-cycle-assessment-model.

[56] Pradeep Lall, Naveen Singh, Jeffrey Suhling, Marcia Strickland, and Jim Blanche. 2005. Thermo-mechanical reliability tradeoffs for deployment of area array packages in Harsh environments. *Components and Packaging Technologies, IEEE Transactions on* 28 (10 2005), 457 – 466. doi:10.1109/TCAPT.2005.854162

[57] Yueying Li, Omer Graif, and Udit Gupta. 2024. Towards Carbon-Efficient LLM Life Cycle. In *Proceedings of the 3rd Workshop on Sustainable Computer Systems (HotCarbon)*.

[58] Yueying Li, Zhanqiu Hu, Esha Choukse, Rodrigo Fonseca, G. Edward Suh, and Udit Gupta. 2025. EcoServe: Designing Carbon-Aware AI Inference Systems. arXiv:2502.05043 [cs.DC] https://arxiv.org/abs/2502.05043

[59] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs.CV] https://arxiv.org/abs/1405.0312

[60] Yueying Lisa Li, Leo Han, G Edward Suh, Christina Delimitrou, Fiodar Kazhamiaka, Esha Choukse, Rodrigo Fonseca, Liangcheng Yu, Jonathan Mace, and Udit Gupta. 2025. Fair, Practical, and Efficient Carbon Accounting for LLM Serving. *SIGMETRICS Perform. Eval. Rev.* 53, 2 (Aug. 2025), 99–103. doi:10.1145/3764944.3764967

[61] Jingping Liu, Cheng Yang, Haoyi Wu, Ziyin Lin, Zhexu Zhang, Ronghe Wang, Baohua Li, Feiyu Kang, Lei Shi, and Ching Ping Wong. 2014. Future paper based printed circuit boards for green electronics: fabrication and life cycle assessment. *Energy Environ. Sci.* 7 (2014), 3674–3682. Issue 11. doi:10.1039/C4EE01995D

[62] Sasha Luccioni, Yacine Jernite, and Emma Strubell. 2024. Power Hungry Processing: Watts Driving the Cost of AI Deployment?. In *The 2024 ACM Conference on Fairness Accountability and Transparency (FAccT '24)*. ACM, 85–99. doi:10.1145/3630106.3658542

[63] Pol Maistriaux, Thibault Pirson, Maxime Schramme, Jérôme Louveaux, and David Bol. 2022. Modeling the carbon footprint of battery-powered IoT sensor nodes for environmental-monitoring applications. In *Proceedings of the 12th International Conference on the Internet of Things*. 9–16.

[64] Ilan Mandel and Udit Gupta. 2025. Silicon Foraging: Harvesting Excess Compute for Sustainable Edge Computing. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference*. 2574–2584.

[65] George Mois, Silviu Folea, and Teodora Sanislav. 2017. Analysis of three IoT-based wireless sensors for environmental monitoring. *IEEE Transactions on Instrumentation and Measurement* 66, 8 (2017), 2056–2064.

[66] Huy Nguyen and Michael Vai. 2010. Rapid prototyping technology. *Lincoln laboratory journal* 18, 2 (2010), 17–27.

[67] Jon Nordby, Mark Cooke, and Adam Horvath. 2019. emlearn: Machine Learning inference engine for Microcontrollers and Embedded Devices. doi:10.5281/zenodo.2589394

[68] Nordic Semiconductor. 2021. *Power Profiler Kit II User Guide*. Nordic Semiconductor ASA, Trondheim, Norway. https://www.nordicsemi.com/Products/Development-hardware/Power-Profiler-Kit-2 Version 1.0.1, Document 4461_012.

[69] Thibaut Marty Jean-Christophe Prévotet Maxime Pelcat Olivier Weppe, Jérôme Chossat. 2024. Streamlined Models of CMOS Image Sensors Carbon Impacts. *hal-04632499v1* (2024).

[70] Pennsylvania State University. 2026. EME 812: Available Solar Radiation and How It Is Measured. https://courses.ems.psu.edu/eme812/node/644. College of Earth and Mineral Sciences. Accessed: 2026-04-13.

[71] Peter Wägemann Phillip Rafeck, Sven Posner. 2024. CO2CoDe: Towards Carbon-Aware Hardware/Software Co-Design for Intermittently-Powered Embedded Systems. In *HotCarbon*.

[72] Thibault Pirson and David Bol. 2021. Assessing the embodied carbon footprint of IoT edge devices with a bottom-up life-cycle approach. *Journal of Cleaner Production* 322 (2021), 128966. doi:10.1016/j.jclepro.2021.128966

[73] Thibault Pirson and David Bol. 2021. Assessing the embodied carbon footprint of IoT edge devices with a bottom-up life-cycle approach. *Journal of Cleaner Production* 322 (2021), 128966.

[74] Thibault Pirson, Grégoire Le Brun, Ernesto Quisbert-Trujillo, Thomas Ernst, Jean-Pierre Raskin, and David Bol. 2024. Towards Life Cycle Thinking and Judicious Ecodesign for the Internet of Things (IoT) Current Practices and Perspectives. In *Outlooking beyond Nanoelectronics and Nanosystems*. Jenny Stanford Publishing, 75–136.

[75] Ernesto Quisbert-Trujillo, Thomas Ernst, Karine Evrard Samuel, Emmanuelle Cor, and Elise Monnier. 2020. Lifecycle modeling for the eco design of the Internet of Things. *Procedia CIRP* 90 (2020), 97–101.

[76] Benjamin Ransford, Jacob Sorber, and Kevin Fu. 2011. Mementos: System support for long-running computation on RFID-scale devices. In *Proceedings of the sixteenth international conference on Architectural support for programming languages and operating systems*. 159–170.

[77] Ian Schneider, Hui Xu, Stephan Benecke, David Patterson, Keguo Huang, Parthasarathy Ranganathan, and Cooper Elsworth. 2025. Life-cycle emissions of ai hardware: A cradle-to-grave approach and generational trends. *arXiv preprint arXiv:2502.01671* (2025).

[78] Lucy Smith, Taofeeq Ibn-Mohammed, S.C. Lenny Koh, and Ian M. Reaney. 2018. Life cycle assessment and environmental profile evaluations of high volumetric efficiency capacitors. *Applied Energy* 220 (2018), 496–513. doi:10.1016/j.apenergy.2018.03.067

[79] Yonglak Son, Udit Gupta, Andrew McCrabb, Young Geun Kim, Valeria Bertacco, David Brooks, and Carole-Jean Wu. 2025. GreenScale: Carbon Optimization for Edge Computing. *IEEE Internet of Things Journal* (2025).

[80] Abel Souza, Shruti Jasoria, Basundhara Chakrabarty, Alexander Bridgwater, Axel Lundberg, Filip Skogh, Ahmed Ali-Eldin, David Irwin, and Prashant Shenoy. 2023. Casper: Carbon-aware scheduling and provisioning for distributed web services. In *Proceedings of the 14th International Green and Sustainable Computing Conference*. 67–73.

[81] Bharath Sudharsan, Simone Salerno, and Rajiv Ranjan. 2022. TinyML-CAM: 80 FPS image recognition in 1 kB RAM. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 862–864.

[82] Jennifer Switzer, Gabriel Marcano, Ryan Kastner, and Pat Pannuto. 2023. Junkyard computing: Repurposing discarded smartphones to minimize carbon. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. 400–412.

[83] Swamit Tannu and Prashant J. Nair. 2023. The Dirty Secret of SSDs: Embodied Carbon. *ACM SIGEnergy Energy Informatics Review* 3, 3 (Oct. 2023), 4–9. doi:10.1145/3630614.3630616

[84] TensorFlow contributors. 2020. TensorFlow Lite Micro Benchmarks. https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro/benchmarks. Last accessed June 2025; benchmarks for TensorFlow Lite on microcontrollers.

[85] Albert J.P. Theuwissen. 2008. CMOS image sensors: State-of-the-art. *Solid-State Electronics* 52, 9 (2008), 1401–1406. doi:10.1016/j.sse.2008.04.012 Papers Selected from the 37th European Solid-State Device Research Conference - ESSDERC'07.

[86] Tutiempo Network, S.L. [n. d.]. Solar Radiation: New York City (Central Park). https://en.tutiempo.net/solar-radiation/new-york-city-central-park.html.

[87] Antonis Tzounis, Nikolaos Katsoulas, Thomas Bartzanas, and Constantinos Kittas. 2017. Internet of Things in agriculture, recent advances and future challenges. *Biosystems engineering* 164 (2017), 31–48.

[88] Jaylen Wang, Daniel S. Berger, Fiodar Kazhamiaka, Celine Irvene, Chaojie Zhang, Esha Choukse, Kali Frost, Rodrigo Fonseca, Brijesh Warrier, Chetan Bansal, Jonathan Stern, Ricardo Bianchini, and Akshitha Sriraman. 2025. Designing Cloud Servers for Lower Carbon. In *Proceedings of the 51st Annual International Symposium on Computer Architecture* (Buenos Aires, Argentina) *(ISCA '24)*. IEEE Press, 452–470. doi:10.1109/ISCA59077.2024.00041

[89] Qi Wang, Nan Huang, Zhuo Chen, Xiaowen Chen, Hanying Cai, and Yunpeng Wu. 2023. Environmental data and facts in the semiconductor manufacturing industry: An unexpected high water and energy consumption situation. *Water Cycle* 4 (2023), 47–54.

[90] Pete Warden. 2018. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. *arXiv preprint arXiv:1804.03209* (apr 2018). https://arxiv.org/abs/1804.03209 Speech Commands Dataset.

[91] Olivier Weppe, Thibaut Marty, Sébastien Toussaint, Nicolas Brusselmans, Jean-Christophe Prévotet, Jean-Pierre Raskin, and Maxime Pelcat. 2025. Embodied Carbon Footprint of 3D NAND Memories. In *Inter-national Conference on Computing Frontiers (CF'25)*. Cagliari (Sardinia), France. doi:10.1145/3706594.3727962

[92] Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. 2006. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the 7th symposium on Operating systems design and implementation*. 381–396.

[93] Beth Whitehead, Deborah Andrews, and Amip Shah. 2015. The life cycle assessment of a UK data centre. *The International Journal of Life Cycle Assessment* 20, 3 (2015), 332–349.

[94] Hongbang Wu, Xuesi Chen, Shubham Jadhav, Amit Lal, Lillian Pentecost, and Udit Gupta. 2026. COFFEE: A Carbon-Modeling and Optimization Framework for HZO-based FeFET eNVMs. arXiv:2602.05018 [cs.AR] https://arxiv.org/abs/2602.05018

[95] Zeyu Yan, Su Hwan Hong, Josiah Hester, Tingyu Cheng, and Huaishu Peng. 2025. DissolvPCB: Fully Recyclable 3D-Printed Electronics Using Liquid Metal Conductors and PVA Substrates. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 106, 17 pages. doi:10.1145/3746059.3747604

[96] Chen Yang, Weiming Shen, and Xianbin Wang. 2016. Applications of Internet of Things in manufacturing. In *2016 IEEE 20th international conference on computer supported cooperative work in design (CSCWD)*. IEEE, 670–675.

[97] Rongjie Yi, Ting Cao, Ao Zhou, Xiao Ma, Shangguang Wang, and Mengwei Xu. 2023. Boosting DNN Cold Inference on Devices. In *The 21st International Conference on Mobile Systems, Applications, and Services*.

[98] Eren Yildiz, Saad Ahmed, Bashima Islam, Josiah Hester, and Kasim Sinan Yildirim. 2023. Efficient and safe i/o operations for intermittent systems. In *Proceedings of the Eighteenth European Conference on Computer Systems*. 63–78.
[99] Cheng Zhang, Yu Zheng, Junfeng Jing, Yun Liu, and Haihong Huang. 2022. A comparative LCA study on aluminum electrolytic capacitors: From liquid-state electrolyte, solid-state polymer to their hybrid. *Journal of Cleaner Production* 375 (2022), 134044. doi:10.1016/j.jclepro.2022.134044
[100] Pei Zhang, Christopher M Sadler, Stephen A Lyon, and Margaret Martonosi. 2004. Hardware design experiences in ZebraNet. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 227–238.
[101] Shifeng Zhang, Yiliang Xie, Jun Wan, Hansheng Xia, Stan Z. Li, and Guodong Guo. 2019. WiderPerson: A Diverse Dataset for Dense Pedestrian Detection in the Wild. http://www.cbsr.ia.ac.cn/users/sfzhang/WiderPerson/. Accessed: 16 November 2025.
[102] Xiaoyang Zhang, Yijie Yang, and Dan Wang. 2024. Spatial-Temporal Embodied Carbon Models for the Embodied Carbon Accounting of Computer Systems. In *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems (e-Energy '24)*. ACM, 464–471. doi:10.1145/3632775.3661939
[103] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. 2022. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. (2022).
[104] Zhihan Zhang, Felix Hähnlein, Yuxuan Mei, Zachary Englhardt, Shwetak Patel, Adriana Schulz, and Vikram Iyer. 2024. DeltaLCA: Comparative Life-Cycle Assessment for Electronics Design. 8, 1, Article 29 (March 2024), 29 pages. doi:10.1145/3643561

A Artifact Appendix

A.1 Abstract

The source code for the MicroGreen framework, including its embodied carbon modeling and hardware profiling code, is available on github and Zenodo. The framework source is located under the `framework/` directory. Embodied carbon results for the evaluated MCUs and profiling results are stored under `database/`. These results are derived from the embodied carbon modeling code found in the `EmbodiedCarbonModeling/` submodule, while profiling data for hardware inference runtime, power consumption, and wireless experiments is available under `profiling/`.

The profiling setup varies across MCUs and requires different SDK support. The instructions here focus on reproducing the framework results and figures presented in the paper. Specific profiling setups and detailed steps are documented in `GUIDELINE.md` under the `profiling/` directory.

A.2 Artifact check-list (meta-information)

- **Models:** MLPerf Tiny Benchmark, TensorFlow Lite Micro Benchmark, MobileNetV2, YOLOv8 (included in the downloaded profiling code)
- **Runtime environment:** Linux and macOS preferred. Coral Dev Board Micro profiling is not supported on Windows.
- **Hardware:** Raspberry Pi Pico W, Raspberry Pi Pico 2 W, Coral Dev Board Micro, Nordic Semiconductor nRF52840-DK, STM32F411E-DISCO, HUZZAH32 - ESP32, Waveshare ESP32-S3-Zero, ESP32-C6 Super Mini, Nordic Semiconductor Power Profiler Kit II.
- **Metrics:** Inference latency, supply current to MCU
- **Experiments:** README, script
- **Approximate disk space required:** 10 GB
- **Approximate workflow preparation time:** 30 minutes
- **Approximate experiment completion time:** 1 hour for artifact evaluation using existing modeling and profiling results; 1.5–3 hours for full workload profiling per MCU, including software environment setup, compilation, flashing, and power measurements.
- **Publicly available:** Yes
- **Code license:** MIT

A.3 Description

A.3.1 How to access.

- Source code: https://github.com/S4AI-CornellTech/MicroGreen
- Zenodo: https://doi.org/10.5281/zenodo.19501837

A.3.2 Hardware dependencies.

- MCU profiling results are run on Raspberry Pi Pico W, Raspberry Pi Pico 2 W, Coral Dev Board Micro, Nordic Semiconductor nRF52840-DK, STM32F411E-DISCO, HUZZAH32 - ESP32, Waveshare ESP32-S3-Zero, ESP32-C6 Super Mini
- Power measurement is done using Nordic Semiconductor Power Profiler Kit II

A.3.3 Software Dependencies.

- `pico-sdk`, `esp-idf`, `platformio`, `coralmicro` dev toolchain for compiling program for MCUs
- nRF Connect for Desktop for power measurement

A.4 Installation

Ensure the following prerequisites are installed: Git (with submodule support), CMake=3.30.5, and Python 3.10 or later.

```
# clone the MicroGreen repository with all
    submodules

git clone --recurse-submodules git@github.com:S4AI
    -CornellTech/MicroGreen.git

cd MicroGreen
```

Next, run the setup script to install all Python dependencies:

```
# install Python dependencies
bash setup.sh
```

A.5 Evaluation and expected results

The following scripts reproduce the main figures presented in the paper. All commands should be run from the root `MicroGreen/` directory.

```
# Figure 3 - Board Carbon Breakdown
python3 scripts/
    carbon_component_composition_plotter.py

# Figure 5 - Per-Inference Runtime and Energy Plot
python3 scripts/characterization_fig.py

# Figure 6 - Carbon Rank Plot
python3 scripts/overall_eval_carbon.py --lifetime-
    years 1 \
    --solar-panel-area-cap 611

# Figure 7 - Energy Efficiency Rank Plot
python3 scripts/overall_eval_energy.py
```

```
# Figure 8 - Irradiance Analysis Plot
python3 framework/main.py --workload kws-l --solar
    -plot

# Figure 9 - Battery Analysis Plot
python3 framework/main.py --workload kws-s --
    battery-plot

# Figure 10 - Network Analysis Plot
python3 scripts/network_plot.py

# Figure 11 - Hybrid Analysis Plot
python3 framework/main.py --workload kws-l --
    lifetime-plot

# Figure 14
python3 framework/heterogeneousDeployment.py
python3 scripts/case_study_plot.py
```

Each script generates a figure saved to the `figures/` directory. The output plots should match the corresponding figures in the paper. Minor numerical differences may arise due to floating-point precision or platform-specific dependencies, but the overall trends and rankings should be consistent with the results reported.

A.6 Experiment customization

MicroGreen provides an interactive design space exploration interface built with Streamlit. The interface exposes key framework parameters via dropdown menus and sliders, allowing users to explore how workload, device lifetime, and solar panel area affect carbon outcomes without modifying any code.

To launch the interface, run the following command from the root `MicroGreen/` directory (Streamlit is installed automatically by `setup.sh`):

```
# launch the interactive design space exploration
    UI
streamlit run framework/main.py
```

The interface exposes the following parameters:

- **Power Mode**: Three checkboxes to select the device power configuration: `Use Solar Energy Harvesting`, `Use Battery Power`, or `Use Both (Hybrid Power Mode)`.
- **Workload** (`-workload`): A dropdown menu to select the target TinyML workload: `kws-s`, `kws-l`, `ppd-s`, or `ppd-l`.
- **Inferences Per Second** (`-inference-per-second`): A slider to set the target inference rate.
- **Deployment Lifetime** (`-lifetime-years`): A slider to set the assumed device operational lifetime in years.
- **Irradiance** (visible when solar or hybrid mode is enabled): A slider to set the ambient irradiance in μW/cm^2.
- **Max Solar Panel Area** (`-solar-panel-area-cap`, visible in hybrid mode only): A slider to constrain the maximum solar panel area in cm^2.

Adjusting these parameters will update the output plots in real time, allowing direct comparison against the figures reported in the paper.

B MCUs and Workloads Selection Appendix

B.1 MCUs Selection

The full list of boards used can be seen in Table 1. We focused on choosing board with MCUs that show diversity in price, architecture and performance and were designed for wireless and mobile computing in mind. Boards which lacked sufficient computational capability to execute ML inference workloads, such as MSP430-class or sub-MHz microcontrollers, were excluded from our evaluation. Similarly, we excluded high-end embedded Linux platforms such as the Jetson Nano and Raspberry Pi, as these devices operate within a separate class of edge computing with substantially higher power consumption (>5W) and embodied carbon that falls outside the scope of typical IoT deployments. We selected devices that operate at a range of clock speeds (64MHz-800MHz) and architectural diversity (ARM Cortex-M series: M0+, M4F, M7, M33, RISC-V, and Xtensa). Boards range in RAM (128KB-64MB) and Flash (512KB-128MB) which enables models of varying size and complexity to be deployed. Devices which have active developer communities and real world deployments were prioritized. ESP32 devices are ubiquitous in cheap IoT products, STM32 MCUs are dominant within industrial contexts, and the Nordic nRF52840 is extremely popular for low power and wearable applications. The Coral Dev Board Micro, featuring dedicated ML acceleration hardware, represents the high-performance boundary of our evaluation.

Hardware vendors provide software optimizations for neural network inference. For example, Espressif provides ESP-NN for optimized functions on ESP32 devices, analogous to ARM's CMSIS-NN for Cortex-M processors. Similarly, the Raspberry Pi foundation has forked TensorFlow Lite Micro to better take advantage of the dual cores their hardware provides. Where compatible with tensorflow lite, we utilized vendor-provided optimizations where available to represent realistic deployment scenarios, as developers targeting these platforms would employ the same optimizations. The STM32 has their own inference framework, STM32.AI (X-CUBE-AI v10.0) which competes with and is incompatible with tensorflow lite. For consistency, we benchmarked the STM32 using standard TensorFlow Lite Micro with CMSIS-NN, which may under-represent its performance. This approach captures the actual performance and energy consumption achievable in practice, though it means measurements reflect both hardware capability and software ecosystem maturity.

B.2 Inference Workloads Selection

The inference workloads seen in Table 2 represent established benchmarks for edge ML [7, 20, 50, 84], spanning the range from minimal (117 KB) to the limits of what our MCUs can deploy (6.7 MB) [84]. Different model architectures stress vendor hardware differently which broadens the comparisons available. Models selected cover both audio and visual modalities representative of common IoT workloads. We directly apply the person detection workload in our case study in Section 6. While model accuracy is outside the scope of this study, evaluating small and large models trained on the same task informs designers about the tradeoffs between accuracy and efficiency where larger models tend to be more accurate with a corresponding increase of energy needed per inference.