American Journal of Multidisciplinary Research & Development (AJMRD)
Volume 08, Issue 03 (March - 2026), PP 34-44
ISSN: 2360-821X
www.ajmrd.com

Research Paper **Open Access**

Hardware-Agnostic Co-Design Framework for Near-Sensor/Edge AI Pipelines

Kartik Jain[1]
[1](*Electrical and Computer Engineering, University of California, Irvine, USA)*

ABSTRACT: In this research we propose a hardware-agnostic co-design framework for near-sensor and edge AI-pipelines in which the signal acquisition and downstream inference are jointly optimized with respect to three key metrics namely, power per channel (mW/channel), end-to-end latency, and task-accuracy. Instead of being hardware- or fabrication-process specific the framework uses closed-form models to map abstract workload descriptors of operations on MAC, memory-movements, ADC/sample workload, and event rates to energy and latency-proxies. The framework performs systematic trade-off analysis, creates synthetic and reproducible datasets of three representative modalities, i.e., frame-based vision, event-based vision, and always-on audio and inertial-sensing. The framework sweeps co-design parameters such as in-sensor precomputation, quantization-friendly architectural scaling, and event thresholds. Such findings are useful in cases where (i) early compute can cut power proxy in range of 59% with frame-based vision and no accuracy loss; (ii) event-based pipelines which are always-on and maintain <1mW/channel and 10ms windows with no loss of accuracy; and (iii) always-on audio and multi-axis IMU-gesture recognition to achieve 0.048 mW/channel besides 1-8 mW/channel at practical latencies. This work focuses on methodology in the co-design of sensors and accelerators without making hardware-commitment and is equipped with a script be run directly to generate figures and tables.

Keywords - Hardware-Agnostic Co-Design; Near-Sensor Computing; Edge AI; Energy-Efficient Machine Learning; Event-Based Vision; Always-On Sensing

I. INTRODUCTION

In the edge computing systems, it has been the requirement that sensing and inference processes are to be executed under extreme energy and latency constraints in such a way it protects user privacy whilst maintaining the responsiveness of systems [1]. One approach that may help resolve difficulties is a quantitative proximal shifting of computation to the sensor element by doing initial feature extraction in the sensor periphery in the pixel array. This scheme decreases analog-to-digital conversion as well as memory traffic. In the literature review of previous research, it has shown that pixel-level processing (P2M) is possible in CMOS-based imagers, indicating that 1-mm computation at the beginning of the computational chain can significantly improve system efficiency [2].

On-sensor AI in commercial imaging sensors implements the benefits of placing the sensor in the corresponding regions of interest (ROI) for saving bandwidth, minimizing latency, and improving privacy [3][4]. In the case of always-on audio with the uses of hardware-aware training (HAT), it helps reduce power usage down to the microwatt level without compromising state-of-the-art accuracy [5]. The methods used in research aims to be a hardware-independent approach and a completely synthetic-pipeline of experimental-work that can be replicated, altered, and calibrated by other researchers on actual silicon or board-level system designers. This means that the first consideration that we have is our emphasis on normalized cost models and self-generated data for in opposed to reliance on foundry for their process design kit, or silicon chip [6].

In research community, reaction to complex addressing of limitations has taken a different shift to the paradigm of bringing the computation to the sensor. This change of architecture attempts to preclude for their redundant information at the earliest stage possible for reducing the processing load in other areas. The concrete examples are processing-in-pixel (P2M) models that convolute, normalize, or activate the pixel array or its surroundings for decreasing the bulk of the raw data that needs to be digitalized and sent for improving accuracy, and reducing latency [7] [8]. Phosphor-based artificial intelligence systems use digital signal processors and memory located below the sensor arrays in pre-processing inference that frequently delivers only small metadata or regions of interest instead of full-frame resolution images. Event cameras do not use the traditional

frame paradigm in which layers of dense images are seldom emitted, and asynchronous events are only emitted with a change of pixel intensity, resulting in intrinsically sparse data streams that are best because to low-latency inferences. Their system is based on hardware-aware training methods in always-on audio systems jointly optimize to be with neural-network architecture with hardware budgets with ultra-low-power audio systems, and with the capability to spot keywords in the microwatt power range [9].

1.1 Research Motivation

The motivation for research as described in three-point elements are listed. The former element supports exploration of architecture at early phases to help researchers and system designers to investigate trade-offs associated with early feature extraction, window size, and model scale before hardware is chosen. The second element is to increases reproducibility using synthetic datasets and transparent modelling assumptions for measurements independent of proprietary hardware. The third step of element promotes hardware-independent planning: after a target platform is selected with the normalized coefficients recalibrated with measured constants to maintain the analytical structure and integrity.

1.2 Research Objective

The main intention of the paper is to come up with a hardware-independent code resulting in the ideal design of sensing and inference pipelines and optimization of three types of equally critical performance-metrics, namely, power-efficiency (milliwatts-per channel), end-to-end latency, and classification-accuracy.

1. Create a co-design framework that is hardware independent to optimize power, latency, and accuracy.
2. The use of normalized cost models to compute for holding memory and analog to digital conversion.
3. The comparison of the trade-offs depending on the specific use of hardware, and usage of synthetic data to evaluate reproducibility.
4. Evaluation of trade-off between early compute and backend processing in research for the frame-based contrast with event-based vision pipelines.
5. Expand the architecture to include audio and inertial-measurement unit (IMU) sensing, stride management, and model scaling.

II. RELATED WORK

The related-work section aims at locating the proposed framework within the larger scope of near-sensor and edge-AI studies. The past decade has witnessed significant advances in making intelligent sensing systems less energy consuming and less latent through reorganizing the traditional division of responsibility between sensing and computing [10]. It has been shown that carrying out the computations on the sensor array, as opposed to outsourcing the processing of raw data to a remote digital computer, can significantly reduce bandwidth requirements, spatial memory transfers, and overall power consumption. Likewise, event-based sensing models have demonstrated that the transmission of signal changes, but do not completely utilize environmental sparsity to enable milliwatt-level operation. Hardware-constrained training in the audio world again established by hardware-aware training methods where neural-network architecture is co-optimized with hardware constraints can reach the state of ultra-low power inference without affecting the accuracy of the tasks [11][12].

2.1 In/Near-Sensor Vision (Processing-in-Pixel – P2M)

In near-sensor vision in the Processing-in-Pixel (P2M) paradigm is a foremost change in architectural structure with respect to more traditional-based image-acquiring pipelines. Traditional systems have each pixel storing only the amount of light, which is converted into analog signals are completely digitized and fed to downstream processors to extract features and make inferences [13]. The subsequent separation incurs the transfer of large amounts of pixel information over ADCs and memory structures at the cost of high energy and latency. P2M architectures radically alter this framework and have neural-network functions performed at the initial stages in convolution, batch normalization, and activation functions like ReLU, incorporated directly into or near the pixel array. The implementation of partial feature extraction before full digitization of vectors significantly lowers computation of raw data to leave the sensor inhibiting the use of ADCs, reducing the memory bandwidth needed, and reducing the total energy-delay product (EDP). In the system which has careful algorithm co-design of the circuit in embedded functionality, maintains task-level accuracy notwithstanding such limitations which are limited to precision or analog variability [9] [14]. In P2M providing smart redistribution of computational load can provide significant efficiency improvements without affecting the inference performance, positioning it as a guiding paradigm of near-sensor AI optimization.

Majority of the existing contributions are closely related to specific hardware implementations, like a particular CMOS image sensor, a neuromorphic chip, a microcontroller unit (MCU), or a custom accelerator fabricated using a specific process node. The energy and latency benefits claimed by these works determined by both the architectural innovation and also their technology-specific parameters such as supply voltage, memory-hierarchy architecture, fabrication efficiency, and circuit-level optimization are still problematic to generalize

conclusions made on different platforms and create a fair comparison of alternative sensing modalities made on a single framework [15].

2.2 On-Sensor AI Architectures

This takes the near-sensor concept to a higher level by adding larger on-digit processing units. On-sensor usage of AI such as dedicated DSP blocks and on-chip SRAM as under or on the sensor array. On-sensor AI platforms prepare to train the entire or a portion of a neural-network inference prior to the off-chip transfer of data, unlike P2M which embed lightweight operations in the pixel domain [16]. An example is an image capture sensor with embedded inference like the Sony IMX500 architecture which is able to produce compact metadata, object labels, bounding boxes, or regions of interest of sending full-resolution frames.

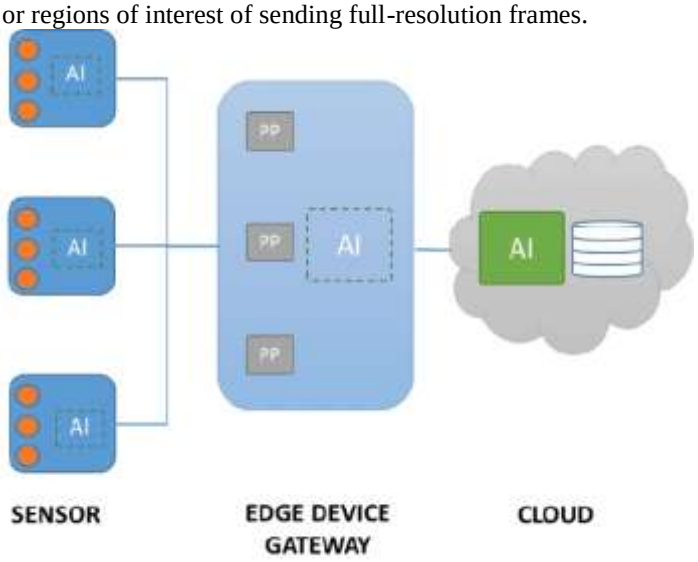

Figure 1: Sensor AI Architecture

The diagram shows a typical architectural pattern of an IoT-system where sensors with local pre-processing send data to an edge device and, potentially, to the cloud. It shows the collaboration of intelligent sensors, edge gateways, and cloud AI [17]. This architecture has several system-level benefits which significantly reduces the bandwidth requirements since only information that is related to the task is exported; it decreases end-to-end latency since there is no-need to produce large datasets and lastly increases privacy since the raw visual data is localized in the sensor module, and therefore sensitive data is not leaked. These systems often get associated with specific silicon implementations and hardware configurations difficult to generalize beyond a platform with cloud-based which is removed by the extant broadside through generalization of data in cloud memory.

2.3 Event-Based Sensing

Event-based sensing follows a radically different sensing principle of acquiring visual data by means of avoiding the classic frame-based paradigm. Dynamic Vision Sensors (DVS) do not attempt to record entire images at fixed frame rates as they asynchronously produce events when individual pixel intensities change relative to a specified threshold. The events represent spatial location, polarity, and time stamp, and yield a sparse and temporally accurate description of the dynamics of the scene [18]. This asynchronous functionality allows microsecond-level latency since information is relayed on-the-fly.

Table 1: Key Characteristics of Event-Based Sensing [18]

Aspect	Description	Benefit
Data Transmission	Data is sent only when a significant change or event occurs	Reduces unnecessary data flow
Power Consumption	Sensors remain mostly idle until an event is detected	Improves energy efficiency
Latency	Immediate response to detected events	Enables real-time processing
Data Volume	Only relevant information is captured	Reduces storage and bandwidth requirements
Application Areas	IoT, edge computing, smart monitoring, autonomous systems	Supports efficient intelligent systems

The above table 2 in which Event-based systems is combined with neural models designed to run sparse or event-driven computations are like spiking neural networks or lightweight convolutional architectures, that can reach end-to-end power consumption in the milliwatt range with comparable task performance. This makes them especially well adapted to motion-dominated conditions where the temporal variations can transmit more information textures that are not moving [19]. There are some other near-sensor methods, which have the reported energy efficiencies usually dependent on their neuromorphic hardware or bespoke accelerators in the point of evaluation and limited to hardware-constrained situations. A generalized modelling method suggested in this paper is therefore being used.

2.4 Always-On Audio with Hardware-Aware Training

The keyword spotting (KWS) applications of Always-on audio-systems are a field where energy efficiency is of paramount importance. Devices such as smart speakers and wearable assistants have to constantly monitor the audio stream and use minimal amounts of power to conserve battery life. Hardware-aware training (HAT) has become an impressive tool to meet this challenge, co-optimizing the neural-network architecture and training procedure against hardware-related implementation constraints. Instead of using a post hoc mapping of a predefined model, HAT has incorporated factors such as quantization precision, memory footprint, compute budget, and architecture limitations as a part of the training process. This technique produces small, power-efficient devices in the microwatt (uW) range on microcontrollers or custom accelerators. Techniques such as low-bit quantization, structured pruning, and temporal-window optimization are used to cut the inference cost [20].

This approach considers factors such as computational complexity, memory footprint, and energy efficiency during training. The applied techniques such as network pruning, quantization, and architecture scaling are commonly used to adapt neural networks for efficient execution on embedded processors or specialized AI accelerators [19]. The alignment of training process with the capabilities of the target hardware, always-on audio models are able to strike a balance between performance and efficiency. The system is based on real-time keyword spotting and acoustic event detection on edge devices with minimal latency and extremely low power consumption [21].

III. HARDWARE-AGNOSTIC MODELING

The proposed hardware-agnostic modeling framework designed to evaluate sensing and AI processing systems independently of specific hardware platforms. The design framework shown below in figure 2, in which three core aspects - performance objectives, normalized energy cost modeling, and synthetic data generation are focused. The first step is to design a framework that defines optimization goals including minimizing power consumption per channel, reducing inference latency, and maximizing task accuracy. The usage of normalized cost model is then introduced using energy-coefficients for compute-operations, memory movement, and ADC sampling for analytical estimation of system power. In the end, synthetic multimodal datasets including vision, event-streams, audio-keywords, and IMU gestures are generated to simulate realistic sensing scenarios. This structured process ensures usage of systematic evaluation of sensor-level processing in the backend inference, and overall power–performance trade-offs.

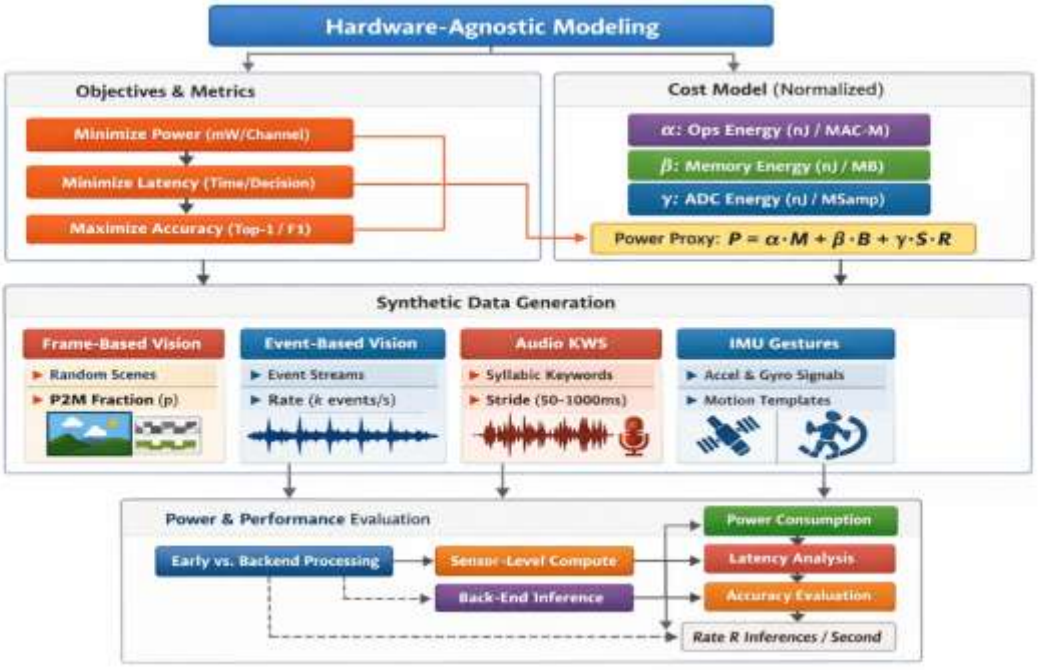

Figure 2: Proposed Framework of hardware-agnostic modeling

3.1 Objectives and Metrics

The structure is given a sensing problem with C physical channels such as microphones, EMG/US probes, or grouped event streams system aimed to minimize three key metrics comprising of energy per channel, latency, and accuracy of task.

Energy per channel is defined as the total time-average power divided by the number of sensing channels [6]. The formulation is given as:

$$\frac{Ptot}{C} \quad (1)$$

Where Ptot represents the time-averaged power proxy and C-represents the number of sensing channels.

Latency represents the total time required to generate a decision from sensor input. It includes sensing, usage of ADC conversion, preprocessing, inference, and input/output communication [7][8]. The latency formulation is:

$$L = L_{sense} + L_{ADC} + L_{pre} + L_{infer} + \frac{L_I}{0} \quad (2)$$

Accuracy-score represents the task-level performance of the system. This is evaluated using metrics such as Top-1 accuracy-score, F1-score, or regression-error depending on the application.

3.2 Cost Model (normalized)

We define normalized coefficients (units in nano-Joules, nJ) that are not tied to any process or chip, yet preserve relative trade-offs:

1. k_{MAC}: Energy per-million MACs (e.g., 1.5 nJ/MAC-M) [9].
2. k_{MEM}: Energy per-MB of memory movement (e.g., 50,000 nJ/MB) including interconnect effects.
3. k_{ADC}: Energy per-MSample of ADC (e.g., 120,000 nJ/MSamp) [9][10].

In system for their configuration with $MACs_M$,B_{mem} MB moved, and S_{ADC} MSamples:

$$E_{\inf(nJ)} = k_{MAC}.MACs_M + k_{MEM}.B_{mem} + k_{ADC}.S_{ADC} \quad (3)$$

Power proxy for rate R inferences:

$$P_{mW} = E_{nj}(nJ) * 10^{-6} * R \quad (3)$$

The pushing of early compute into or near the sensor reduces S_{ADC} (fewer pixels digitized), shrinks B_{mem} (smaller features leave the sensor), and can also reduce $MACs_M$ in the back-end (fewer layers remain) [16]. This mirrors outcomes reported in P2M and on-sensor AI studies.

3.3 Synthetic Data Generators

We introduce task-agnostic synthesized datasets across a variety of sensorial modalities to test a hardware-agnostic co-design approach with no dataset-specific bias [17]. The rigor of synthetic dataset control offers reproducible experiments and accurate evaluation of energy, latency, and accuracy trade-offs in terms of signal complexity, noise properties, and temporal dynamics.

1. **Frame-Based Vision:** In order to simulate the visual complexity of the real world, geometric scenes with heterogeneous texture are randomly generated. Deterministic composition rules produce scene labels that guarantee uniformity and reproducibility across trials. The P2M fraction, which is considered the percentage of initial neural-network layers that are run at the sensor or in its vicinity directly, is parameterized by p 0,1. With changing p, we analyse the effects of shifting computation up to the backend on the workload, power usage, latency, and classification accuracy [5]. This model allows for experimentation with the early-execution algorithms controlled and simulates by the traditional CNN pipelines, which do not require the use of real image datasets.
2. **Event-based Vision**: Artificial event streams are created to model neuromorphic or event-camera sensing. The variability in sensor output and environmental conditions is described with the stochastic event streams that are defined by an event rate (k events/s) and a noise level. The temporal sensitivity of event-based sensors is high, as the processing signals indicate the processing of signals in 10 ms frames. The use of this modality allows the evaluation of sparse, asynchronous methods of sensing and shows how localized processing and minimized data transfer can reduce latency, energy expenditure, and system-wide performance [12].
3. **Audio Keyword Spotting (KWS):** Artificial syllabic tokens are added to the controlled background noise to simulate the audio environment of the real world. Analysis windows (strides) used to determine inference frequency and system responsiveness in the range of 50 ms up to 1000 ms [9]. In the reflections of trade-offs b/w latency, energy usage, and accuracy, the width of the neural network is scaled with strides using thinner based models. The setup is controlled to conduct systematic testing of always-on audio-recognition systems and explain how model configuration, and co-design interact.
4. **IMU Gestures:** IMU Gestures of six-axis acceleration of synthetic data in the form 3D space data in gyroscopes are synthesized with controlled perturbations to variability in the amplitude, timing, and trajectory. The simulated real-world inter-user variations are also included in setup. The synthetic-dataset are used by models in systematic testing of motion-based recognition like DTW-Lite, Tiny LSTM, and 1D CNNs, with different stride in maintaining reproducibility and ensuring full control over both temporal and spatial signal characteristics [4].

IV. EXPERIMENT SETUP

This chapter describes the main aim of the experimental design to explore the trade-offs between power consumption, latency plus accuracy of sensing modalities. The investigation uses hardware platforms, ordinary energy models and artificial workloads to ensure reproducibility and fair comparison between dissimilar configurations. The energy coefficient uses a variety of sensing modalities which are frame-based vision, event-based vision, audio keyword spotting and inertial-measurement-unit (IMU) gesture recognition and a range of architectural parameters in the proportion of early compute and stride length. The framework is controlled to be varying in each of these parameters towards computation placement, sampling strategies and model configurations for determining complete system efficiency in near-sensor and edge AI systems.

4.1 Coefficients

The normalized energy coefficients used in the experimental framework for evaluating the hardware-agnostic performance are given below.

- **MAC operations:** k_{MAC}=1.5 nJ/(MAC-M)
- **Memory movement:** k_{MEM}=50,000 nJ/MB
- **ADC sampling:** k_{ADC}=120,000 nJ/MSamp

The listed above coefficients provide a uniform hardware-independent basis to quantify energy and latency trade-offs across all modalities [9][15].

4.2 Sensing Modalities and Configurations

The systematic comparison of diverse sensing strategies and model architectures on power consumption, latency and accuracy in the hardware-agnostic evaluation of models is described in table 2 below.

Table 2: Modalities Sensors configurations

Modality	Sampling / Window	Model / Processing	MACs / Memory / ADC	Stride / Early Compute	Notes
Frame Vision	30 fps	Baseline CNN	12 M MACs/frame, 6 MB, 50 MSamp	(p = 0,0.25,0.5,0.75)	Early compute reduces back-end MACs, memory, ADC
Event Vision	10 ms windows	Event-based pipeline	–	Event rate: 50–200 k events/s	Base sensor overhead 0.8 mW/channel
Audio KWS	16 kHz mono	Model width scaled with stride	–	Strides: 1000, 250, 50 ms	Smaller stride → narrower model
IMU Gestures	6 channels @100 Hz	DTW Lite, Tiny LSTM, 1D CNN	–	Strides: 500, 250, 200 ms	Low power µW/channel, stride affects latency & accuracy

4.3 Key Observations

There are few key observations which are listed below:

1. **Early Compute Reduces Power:** The early fame vision as in Table 2 shows increasing the P2M fraction from 0% to 75% reduces the power proxy from 189.0 mW/channel to 77.0 mW/channel. This is 59% decrease with negligible accuracy-loss of ≤1 percentage point. This trend aligns with prior P2M studies of normalized modelling [19].
2. **Efficiency of Event Pipelines:** The event power range is 50–200 k events, the event-based vision pipeline maintains ~0.80 mW/channel at 10 ms windows.
3. **Ultra-Low Consumption for Audio and IMU:** Always-on KWS ratio is 0.042–0.482 mW/channel as stride decreases from 1 s to 50 ms, with minor accuracy loss at the smallest stride. The ultra-low and IMU gesture models operate in the µW/channel range under typical strides and model configurations.

V. RESULTS EVALUATIONS

In below mention of Table 3 is each modality, key architectural parameters such as, analysis stride, model structure, event rate and early compute fraction (P2M), which were systematically varied to determine their effects on the workload properties, energy proxies and early inference. The ensuing measurements provide quantitative information on the effects of sensing modality, the location of the computer, and model scaling on the trade-offs of power efficiency, latency, and accuracy in the proposed hardware-agnostic model.

Table 3: Configuration of tasks, power, latency,accuracy-score

Task	Configuration	Power (mW/channel)	Latency (ms)	Accuracy
AUDIO-KWS	stride = 1000ms	0.042	1000	0.965
AUDIO-KWS	stride = 250ms	0.13	250	0.965
AUDIO-KWS	stride = 50ms	0.482	50	0.955
IMU_Gesture	DTW-lite, stride = 50ms	0.001	500	0.94
IMU_Gesture	Tiny-LSTM, stride = 250ms	0.004	250	0.96
IMU_Gesture	CNN1D, stride = 200ms	0.008	200	0.965
Vision-Event	50k events/s	0.801	10	0.84
Vision-Event	100k events/s	0.801	10	0.89
Vision-Event	200k events/s	0.802	10	0.9
Vision-Frame	P2M = 75%	76.95	33.3	0.9
Vision-Frame	P2M = 50%	114.3	33.3	0.91
Vision-Frame	P2M = 25%	151.65	33.3	0.91
Vision-Frame	P2M = 0%	189.001	33.3	0.91

5.1 Power (mW/channel)

The power dissipation curve in Fig 3 shows that early computing can significantly decrease the power

usage in frame-based vision pipelines. At a P2M of 0 the power density values to around 189mW/channel; reducing to 150mW/channel at P2M of 25; and 112mW/channel at P2M of 75; achieving a cumulative power reduction of about 59. Event-based vision, by contrast, has a steady low power density of the order of 0.80mW/channel over a frequency range of 50k to 200k events. In the case of audio KWS, power decreases with stride length with values of 0.042 0.482 mW/channel when the stride is 1000 m and 50 m, respectively.

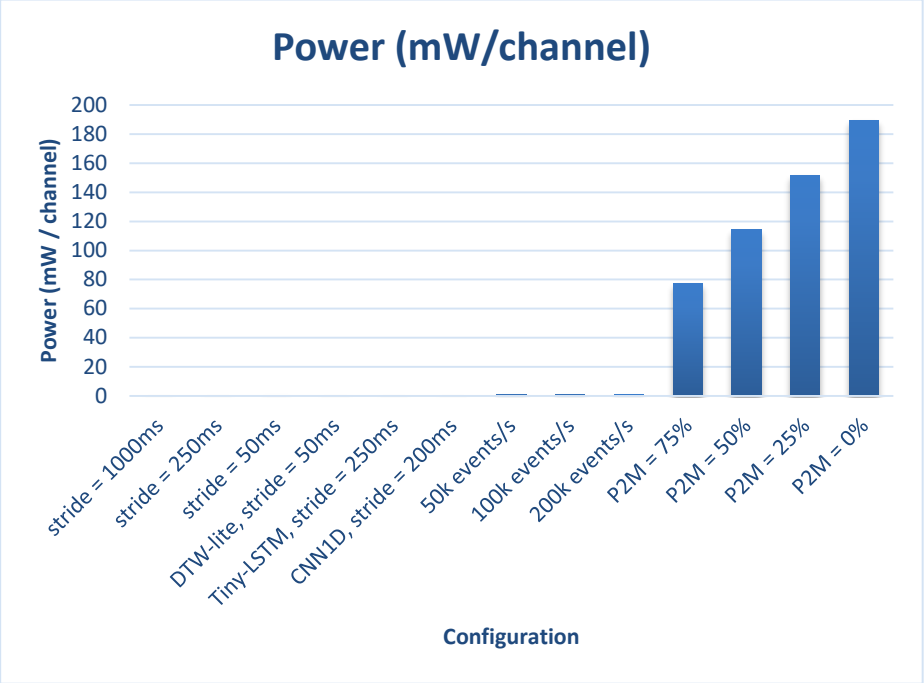

Figure 3: Power-(mW-channel) confgurations

5.2 Latency (ms)

This is because the latency profile in Fig 4 shows that the response time is very much affected by the stride length. In the case of audio sensing, the latency is estimated to be 1000 ms at a 1000 ms stride, 250 ms at a 250 ms stride and approximately 50 ms at a 50 ms stride. The Tiny-LSTM model at 250 ms stride has a latency of approximately 280 ms, as compared to DTW-Lite that has an approximate latency of 500 ms because of overheads in its sequence-processing. The event-based vision has a low latency of operation, approximately 10 ms windows, thus allowing the fast processing of images in real time.

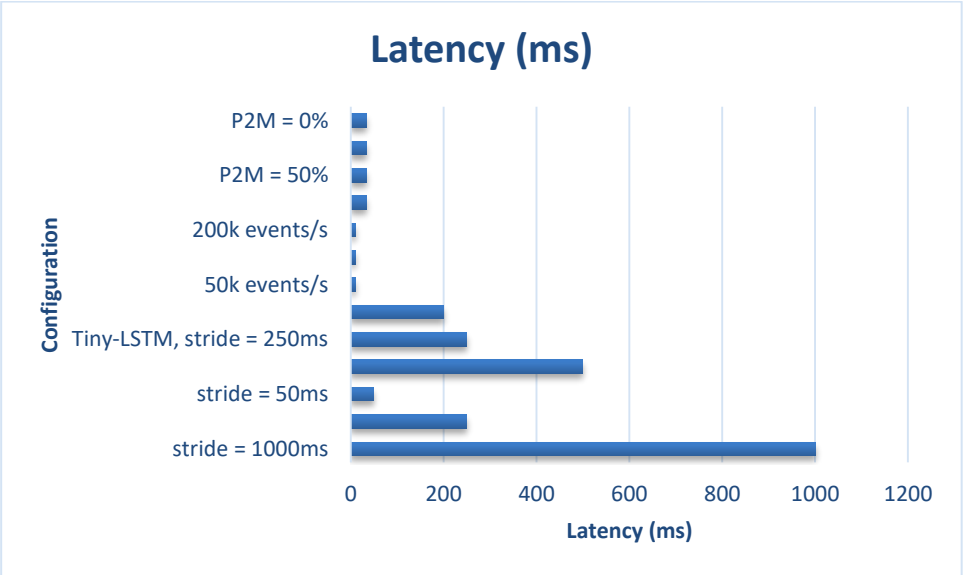

Figure 4: configurations of Latency

5.3 Accuracy

Accuracy test demonstrates that the performance is high between configurations. In Fig 5 it is shown for audio KWS the accuracy is about 0.96 with a 1000-ms stride, 0.96 with a 250-ms stride, and reduces slightly to 0.95 with a 50-ms stride. DTW-Lite, tiny-LSTM, and 1D CNN give about 0.94, 0.96, and 0.97 accuracy in IMU gesture recognition, respectively. In event-based vision, accuracy increases with event rate, with 0.84 at 50000 events per second, and 0.90 at 200000 events per second, whereas frame vision P2M systems are nearly identical with a accuracy of about 0.91-0.92, so a significant reduction in power does not significantly reduce accuracy.

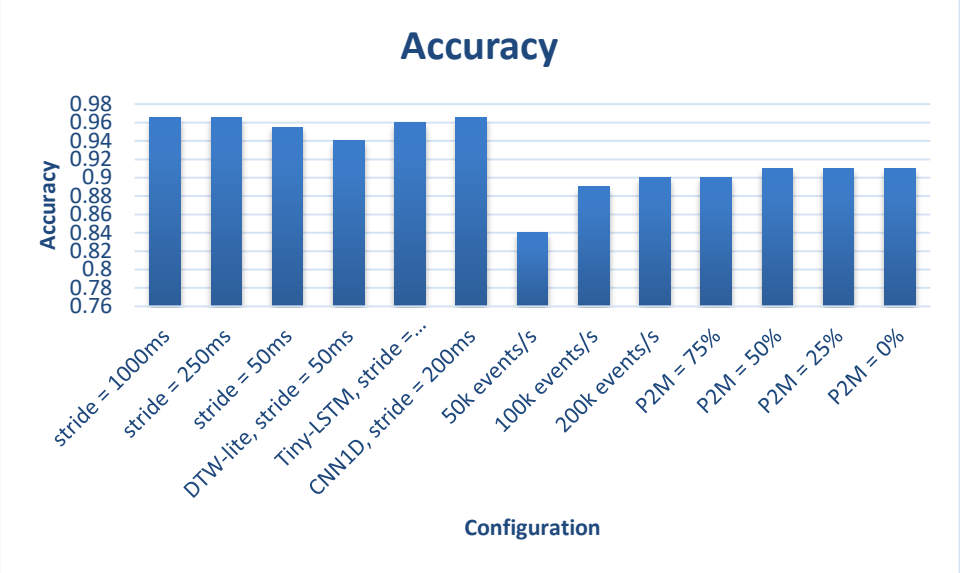

Figure 5: Accuracy-score

VI. DISCUSSION

The first step is to initiate hardware-neutral modeling, because this will decouple algorithmic decision-making and the limitations created by specific technologies. With the abstraction of hardware parameters, designers can search more easily through the design space without necessarily mixing up architectural trade-offs, like the placement of early features, temporal windowing, or event thresholds, with the actual implementation. After selection of a target platform, workload metrics and empirically determined coefficients (kMAC, kMEM, and kADC) it can help in direct, comparative evaluation of a workload without changing the evaluation methodology. The nature of the input signals defines the desirable nature of early computing as compared to event-based sensing. It is beneficial to perform convolutional, batch-normalization, and ReLU steps in/near the sensor because frame-dominant scenes with rich texture or minimal motion are characterized by such cases. This plan maintains semantics of CNN architecture and minimizes ADC activity as well as memory.

On event-based sensing coupled with sparse models or spiking neural networks (SNNs) - the latter attains milliwatt-level power consumption and milliseconds-scale latency- which are better whenever the motion is dominant. Moreover, architecture provides a direct connection with one-sensor artificial intelligence concepts. The principle of extracting compact metadata of the sensor instead of transmitting or storing whole streams of raw data in the absence of an inherently intelligent sensor is reflected in the model. The paradigm decreases the latency, and bandwidth needs and increases privacy, which are already implemented in a number of commercial sensor systems. The methodology enables the designers to analyze these trade-offs in a statistical manner, regardless of particular hardware technologies.

Lastly, the weaknesses of the study should be stated. Synthetic data and standardized coefficients give general trends and produce conclusions, which reflect the general trends and not the specific power or energy values. The framework offers useful guidance, although precise constants must be developed by calibration using actual hardware measurements, but this does not change the underlying technique or the ability to study co-design trade-offs in the early stages.

VII. CONLCUSION AND FUTURE WORK

7.1 Summary of key findings

The research paper suggests a hardware-agnostic co-design structure to the systematic analysis and optimization

of near-sensor and edge artificial intelligence pipelines among various sensing modalities such as frame-based vision, event-based vision, and continuously running audio and inertial sensing systems.

- Experimental studies indicate that early in-sensor computation can significantly decrease energy usage in frame-based vision systems, with a reduction in power per channel of about 59 percent and still achieve model-accurate performance within reasonable error bounds.
- The event-based vision pipelines have been demonstrated to maintain ultra-low power consumption of less than 1 mW per channel and low-latency processing windows in the range of 10 ms and are demonstrated to be highly appropriate in real-time and always-on edge intelligence applications.
- When applied to audio keyword spotting and IMU-based gesture recognition, the framework reveals evident trade-offs between stride length, latency, and power consumption: shorter strides improve responsiveness, although at a relatively small cost in terms of energy consumption, with high accuracy values ranging between 0.95 and 0.97.

These findings verify that the given methodology allows exploring sensing-inference co-design trade-offs systematically, allowing designers to compare the architectural options of early feature extraction, temporal windowing, and event-driven sensing without the limitations of a particular hardware implementation; these results highlight the possibility of hardware-neutral modelling to guide effective design decisions of next-generation edge AI systems.

7.2 Limitations of Study

The study gives helpful information on the energy, latency, and accuracy trade-offs of near-sensor and edge AI pipelines, but there are following limitations that are needed to be taken into consideration.

1. The first limit is the test based on the normalized cost which do not necessarily considers complexity and variability of actual sensing environments in cost models and synthetic datasets.

2. The framework fails to account for device-level limitations that might affect real power consumption and fails to consider specific hardware architecture, fabrication and system performance when implemented on real hardware platforms.

3. The scope of the experiment is limited to modalities such as framed-based vision, event-based vision, audio keyword spotting, and IMU of gesture recognition.

4. The key points of analysis based on methodological assessment and parameter sweeps and some real-world aspects such as communication overhead, memory-hierarchy effects, and hardware reliability, are unmodeled.

The gaps addressed will be filled in future work using real validation hardware prototypes and real datasets and different sensor applications.

7.3 Future Work

Future research directions include:

- Use of natural datasets to validate trends observed in synthetic simulations and extending experiments to physical sensors.
- The model will implement the additional sensor types likes LiDAR, radar, or bio-signals, to generalize co-design insights further.
- The use of advanced level of automated tools for dynamically selecting early compute fraction or stride parameters based on real-time conditions and constraints.
- Integrating framework with sensor platforms to evaluate actual performance in the edge devices or live operational conditions.

REFERENCES

[1]. Z. Liu, X. Chen, H. Wu, Z. Wang, X. Chen, D. Niyato, and K. Huang, Integrated sensing and edge AI: Realizing intelligent perception in 6G, *IEEE Communications Surveys & Tutorials*, 2025.

[2]. F. Ponzina, *Hardware-software co-design methodologies for edge AI optimization*, doctoral diss., EPFL, Switzerland, 2023.

[3]. [K. Wali, Hardware-software co-design for power-efficient edge-AI systems, *Journal of Artificial Intelligence Machine Learning and Data Science*, 2(4), 2024, 2754-2760.

[4]. T. Ma, *Efficient data-driven machine vision: A co-design of circuit, algorithm, and architecture for edge vision sensors*, doctoral diss., 2024.

[5]. D. Danopoulos, *Hardware-software co-design of deep learning accelerators: From custom to automated design methodologies*, doctoral diss., 2024.

[6]. B. Kim, S. Li, B. Taylor, and Y. Chen, Efficient and robust edge AI: Software, hardware, and the co-design, *ACM Transactions on Embedded Computing Systems*, 24(3), 2025, 1-34.

[7]. J. Henkel, M. Tahoori, H. Khdr, H. Nassar, V. Meyers, D. Chen, and S. Dhruvanarayan, Hardware-software co-design for highly optimized, customized, and reliable AI systems, Proc. *IEEE/ACM*

International Conference on Computer-Aided Design (ICCAD), 2025, 1-9.

[8]. M. Scherer, *Hardware-software co-design for energy-efficient neural network inference at the extreme edge*, doctoral diss., ETH Zurich, Switzerland, 2024.

[9]. M. M. A. A. Odema, *Hardware/software co-design methodologies for efficient AI systems and applications*, doctoral diss., University of California, Irvine, USA, 2024.

[10]. G. Datta, *Towards efficient edge intelligence with in-sensor and neuromorphic computing: Algorithm-hardware co-design*, doctoral diss., University of Southern California, USA, 2023.

[11]. M. Tahoori, V. Meyers, M. S. Roodsari, H. Xu, J. Becker, T. Harbaum, and M. Shelkamy Ali, Hardware-software co-design for machine learning systems made open-source, Proc. *International Conference on Hardware/Software Codesign and System Synthesis*, 2025, 23-32.

[12]. T. Goyal, P. Huang, F. Sutton, B. Maag, and P. Sommer, SMiLe: Automated end-to-end sensing and machine learning co-design, Proc. *International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2022, 12-23.

[13]. F. Porreca, F. Frustaci, and R. Gravina, A codesign framework for the development of next generation wearable computing systems, *Sensors*, 25(21), 2025, 6624.

[14]. D. Ngo, H. C. Park, and B. Kang, Edge intelligence: A review of deep neural network inference in resource-limited environments, *Electronics*, 14(12), 2025, 2495.

[15]. N. K. Jayakodi, J. R. Doppa, and P. P. Pande, A general hardware and software co-design framework for energy-efficient edge AI, Proc. *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2021, 1-7.

[16]. J. Kaur, Edge AI for real-time object detection, *XenonStack Blog*, Apr. 22, 2025. [Online]. Available: https://www.xenonstack.com/blog/edge-ai-for-real-time-object-detection

[17]. L. M. Reinhardt, Architectural co-design and approximation strategies for efficient deep neural network acceleration in edge-oriented custom hardware, *European Journal of Emerging Real-Time IoT and Edge Infrastructures*, 2(01), 2025, 1-11.

[18]. F. Ponzina, S. Machetti, M. Rios, B. W. Denkinger, A. Levisse, G. Ansaloni, and D. Atienza, A hardware/software co-design vision for deep learning at the edge, *IEEE Micro*, 42(6), 2022, 48-54.

[19]. H. Bouzidi, *Efficient deployment of deep neural networks on hardware devices for edge AI*, doctoral diss., Université Polytechnique Hauts-de-France, France, 2024.

[20]. S. Tuli, C. H. Li, R. Sharma, and N. K. Jha, CODEBench: A neural architecture and hardware accelerator co-design framework, *ACM Transactions on Embedded Computing Systems*, 22(3), 2023, 1-30.

[21]. S. Sharma, *Hardware-algorithm co-design for energy-efficient and low-latency domain-specific machine learning systems*, doctoral diss., Georgia Institute of Technology, USA, 2025.

Kartik Jain[1]
[1](Electrical and Computer Engineering, University of California, Irvine, USA)