Neuromorphic Edge Intelligence in Collaborative Pipelines: An Energy-Latency Trade-off Analysis

(IJCSNT) International Journal of Communication Systems and Network Technologies
ISSN Number: 2053-6283
Volume 15, Issue No. 01, 44–58

https://journals.mirlabs.in/index.php/ijcsnt
DOI-10.18486/ijcsnt/15.1.03
IJCSNT

Nghia Dinh[1], Vinh Truong Hoang[2], Bay Nguyen Van[3], Huy Nguyen Quoc[4], Kiet Tran-Trung[5]

Abstract

Neuromorphic processors that execute spiking neural network (SNN) inference through event-driven, spike-based computation achieve milliwatt-scale power consumption and sub-millisecond latency, properties that make them attractive for always-on edge intelligence. However, no systematic framework exists for determining when neuromorphic inference at the device level outperforms offloading to conventional edge or cloud accelerators within a collaborative pipeline. This paper addresses that gap by proposing a four-tier collaborative architecture that explicitly distinguishes the neuromorphic edge from the general-purpose edge and by deriving an energy-latency-accuracy offloading decision boundary between these tiers. We compare five neuromorphic platforms (Loihi 2, SpiNNaker 2, Akida, TrueNorth, DYNAP-SE) against three conventional edge processors using published benchmark data, revealing a 10–100$\times$ energy advantage for spiking inference on sparse workloads alongside a 3–10 percentage point accuracy gap that constrains deployment. Three application case studies, autonomous vehicle perception, industrial IoT predictive maintenance, and wearable health monitoring, ground the framework in concrete deployment scenarios and show how the offloading boundary changes across domains. We further analyze the simulation-to-hardware gap in SNN training as a key factor governing the accuracy constraint. The proposed framework provides system designers with a quantitative tool for placing neuromorphic inference within heterogeneous edge–cloud deployments.

Keywords

Neuromorphic Computing, Spiking Neural Networks, Edge Intelligence, Collaborative Inference, Energy Efficiency

Received: 04 January 2026; **Revised:** 14 April 2026; **Accepted:** 21 April 2026;
Published: 30 April 2026;

[1]AI Lab, Faculty of Information Technology, Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam
[2,4,5]Faculty of Information Technology, Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam
[3]Faculty of Fundamental Studies, Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam

Corresponding author:
Email: vinh.th@ou.edu.vn

Introduction

Edge computing has shifted artificial intelligence from centralized data centers to resource-constrained devices that operate under strict latency and energy budgets[1],[2]. Conventional edge accelerators such as the NVIDIA Jetson platform and Google Coral TPU achieve competitive inference throughput, yet their reliance on clock-driven matrix operations imposes a power floor that limits deployment in always-on, battery-powered scenarios[3]. Neuromorphic processors, hardware that mimics the spike-based, event-driven computation of biological neural circuits, offer an alternative path: by activating only those neurons that receive input events, chips such as Intel Loihi 2, SpiNNaker 2, and BrainChip Akida can sustain inference at milliwatt-scale power while maintaining sub-millisecond response times[4–6].

Despite these promising characteristics, no systematic framework exists for evaluating when neuromorphic inference at the device tier outperforms offloading to a conventional edge or cloud server. Existing surveys catalog hardware specifications or benchmark individual chips[7–9], but rarely quantify the offloading decision in terms of joint energy, latency, and accuracy trade-offs across the complete device–edge-cloud pipeline. Recent work on collaborative intelligence[10],[11] acknowledges the multi-tier nature of modern deployments, yet treats neuromorphic and conventional edge processors as interchangeable, an assumption that ignores fundamental architectural differences in how each tier consumes energy and introduces latency.

This paper bridges that gap by proposing an analytical framework that explicitly models a four-tier collaborative pipeline, sensor/device, neuromorphic edge, general-purpose edge, and cloud, and derives an offloading decision boundary separating conditions under which local spiking inference is preferable to remote conventional inference. Our contributions are as follows:

1. A four-tier pipeline architecture that distinguishes neuromorphic edge processors from general-purpose edge accelerators, enabling fine-grained analysis of where spiking inference fits within collaborative systems (Section III-A).
2. An energy-latency-accuracy model that captures the cost of local neuromorphic inference versus data transmission and remote processing, yielding a closed-form offloading decision boundary (Section III-B – III-C).
3. A comparative analysis of five neuromorphic platforms and three conventional platforms using published benchmark data, visualized as an energy-latency Pareto surface (Section IV).
4. Three application case studies, autonomous vehicle perception, industrial predictive maintenance IoT, and wearable health monitoring, that ground the framework in concrete deployment scenarios (Section V).

The remainder of this paper is organized as follows. Section II reviews spiking neural networks, neuromorphic hardware, and collaborative edge computing. Section III presents the proposed analytical framework. Section IV reports the comparative analysis. Section V examines application case studies. Section VI discusses the simulation-to-hardware gap for the deployment of SNN. Section VII addresses limitations and open challenges, and Section VIII concludes with future research directions.

Background and Related Work

Spiking Neural Networks

Spiking neural networks (SNNs) represent a third generation of neural network models that encode information in the precise timing of discrete spike events rather than continuous-valued activations [12]. Each neuron in an SNN accumulates incoming spikes into a membrane potential; when this potential crosses a threshold, the neuron fires its own spike and resets. The leaky integration-and-fire (LIF) model remains the most widely adopted neuron abstraction in hardware implementations due to its low computational overhead [13].

Two properties make SNNs attractive for edge deployment. First, event-driven sparsity: because neurons remain silent without input events, energy consumption scales with the activity level of the input rather than with the size of the network [14]. Second, temporal coding: spike timing carries information that rate-coded ANNs must approximate through additional time steps, enabling SNNs to reach a classification decision in fewer operations for temporally structured data such as audio, radar pulses, and biosignals [15].

Training SNNs remains harder than training conventional ANNs because the spike function is non-differentiable. Three main strategies address this: (i) ANN-to-SNN conversion, which maps trained ANN weights to an equivalent SNN at the cost of requiring many time steps for accurate rate approximation [16]; (ii) surrogate gradient methods, which replace the true spike derivative with a smooth approximation during back-propagation [17]; and (iii) on-chip plasticity rules such as spike-timing-dependent plasticity (STDP), which learn directly on neuromorphic hardware but currently struggle to scale to deep architectures [18]. The practical accuracy gap between these training approaches and their ANN counterparts is a key factor in the energy-latency trade-off that we analyze in Section III.

Neuromorphic Hardware Platforms

Five neuromorphic processors span the current design space from digital to mixed-signal architectures.

Intel Loihi 2 integrates up to 1 million programmable spiking neurons on a single die, connected through a hierarchical mesh network [4], [19]. Loihi 2 supports on-chip learning via three-factor plasticity rules and achieves inference power below 100 mW for keyword-spotting workloads, with latencies below 5 ms per utterance [20].

SpiNNaker 2 targets large-scale neural simulation through a mesh of ARM-based processing elements, each capable of emulating thousands of LIF neurons in real time [21]. The second-generation chip adds hardware accelerators for exponential and logarithmic functions, reducing the energy per-neuron by an order of magnitude over its predecessor [5].

BrainChip Akida is among the first commercially available neuromorphic processors specifically designed for edge deployment [22]. Akida implements a convolutional SNN pipeline that processes event-camera and standard image inputs at sub-milliwatt power, achieving 85–92% accuracy on vision benchmarks with latencies below 10 ms [6].

IBM TrueNorth introduced the one-million-neuron digital neuromorphic chip in 2014, organized as a tiled array of 4096 neurosynaptic cores [23]. Each core implements 256 neutrons with programmable axon types, consuming approximately 65 mW at full utilization, roughly 26 pJ per synaptic operation.

DYNAP-SE from the Institute of Neuroinformatics adopts a mixed-signal design in which analog circuits emulate neuron and synapse dynamics while digital routers handle inter-core communication [24].

This mixed-signal approach yields extremely low power per spike, but introduces device mismatch that must be compensated during network configuration.

For comparison, we also consider three conventional edge processors: NVIDIA Jetson Orin (based on GPU, 15–60 W), Google Coral Edge TPU (INT8 accelerator, 2–4 W), and STM32 microcontrollers (sub-watt, limited model capacity). These baselines contextualize the advantage of neuromorphic energy and reveal the latency regimes in which conventional hardware remains competitive (Section IV).

Collaborative Intelligence and Edge Computing

Collaborative intelligence distributes inference across multiple tiers of computing resources, trading latency, energy, accuracy, and bandwidth according to task requirements[3], [10]. The dominant paradigm in the literature follows a three-tier hierarchy, device, edge server, and cloud, where early exit mechanisms or model partitioning decide how much computation each tier performs[2].

However, this three-tier view collapses two fundamentally different processor classes into a single "edge" layer. A neuromorphic chip running spiking inference at 10 mW operates under constraints and performance envelopes that differ by orders of magnitude from a GPU-equipped edge server consuming 60 W. Treating both as one tier obscures the critical design question of whether a given workload should remain on the neuromorphic device or be forwarded to a more powerful, but more energy-hungry, conventional accelerator.

We therefore adopt a four-tier collaborative model:

1. **Sensor/device tier:** Raw data acquisition with minimal preprocessing (analog front-ends, event cameras, MEMS sensors).
2. **Neuromorphic edge tier:** Event-driven SNN inference on dedicated neuromorphic hardware, handling low-complexity, latency-critical filtering, and classification.
3. **General-purpose edge tier:** DNN inference on GPU or TPU accelerators, performance of heavier model execution, data aggregation, and local model updates.
4. **Cloud tier:** Large-scale training, global model management, and analytics workloads that tolerate higher latency.

This four-tier decomposition aligns with recent observations that heterogeneous edge deployments benefit from explicit hardware-aware task placement[11], [25]. Section III-A formalizes this architecture and Section III-C derives the decision boundary between the tiers 2 and 3.

Related Work

Several survey threads converge on the topics addressed in this paper, but none fully integrates them.

Schuman et al.[7] and Christensen et al.[8] map the landscape of neuromorphic algorithms and hardware, but focus on chip-level benchmarks without modeling the surrounding system pipeline. Bouvier et al.[9] catalog SNN hardware implementations with emphasis on circuit-level trade-offs rather than system-level deployment. Shrestha et al.[26] provide a comprehensive algorithm–hardware-application taxonomy but stop short of quantifying offloading decisions in multitier deployments.

Zhou et al.[2] and Deng et al.[3] established the foundational frameworks for edge AI, while Mao et al.[27] and Wang et al.[28] formalize task offloading in mobile edge computing. These works treat edge processors as homogeneous and do not distinguish neuromorphic hardware from conventional hardware.

Park et al.[11] survey neuromorphic edge–cloud collaboration but adopt a three-tier model that merges neuromorphic and conventional edge into one layer. Abderrahmane et al.[25] explore SNN partitioning across edge and cloud but assume a fixed offloading point rather than deriving a decision boundary from energy-latency-accuracy trade-offs. Li et al.[29] examine latency-critical neuromorphic inference, but focus on single-tier optimization without modeling the collaborative pipeline.

Gap. No existing work (i) distinguishes a neuromorphic edge tier from a general-purpose edge tier within a formal pipeline model, (ii) derives a closed-form offloading decision boundary as a function of energy, latency, and accuracy, and (iii) validates that boundary across multiple application domains. This paper addresses all three gaps.

Proposed Analytical Framework

Four-Tier Pipeline Architecture

We formalize the four-tier collaborative pipeline introduced in Section II-C as a directed data-flow graph $G = (\mathcal{T}, \mathcal{E})$ where the set of levels $\mathcal{T} = \{T_1, T_2, T_3, T_4\}$ corresponds to the sensor/device, neuromorphic edge, general-purpose edge, and cloud tiers, respectively. Each directed edge $(T_i, T_j) \in \mathcal{E}$ carries a transmission cost characterized by energy E_{tx}^{ij} and latency L_{tx}^{ij}.

Fig. 1 illustrates the architecture. Raw sensor data enters T_1 for optional preprocessing (e.g., event encoding from a dynamic vision sensor). At the neuromorphic edge level T_2, an SNN performs energy-efficient low-latency inference on the event stream. If SNN's confidence falls below a task-specific threshold or the input exceeds the SNN's capacity, the data is forwarded to the general-purpose edge tier T_3 for DNN-based inference. The cloud tier T_4 handles training, global aggregation, and heavy analytics that tolerate round-trip latencies on the order of hundreds of milliseconds.

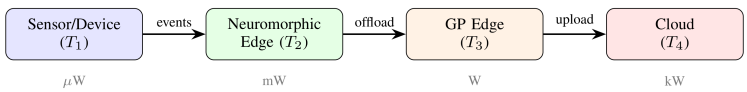

Figure 1. Four-tier collaborative pipeline. Power budgets increase from left to right; latency budgets decrease. The offloading decision between T_2 and T_3 is the focus of Section III-C.

The core insight is that T_2 and T_3 operate under fundamentally different computational models. The Tier T_2 exploits event-driven sparsity: energy scales with input activity, and latency depends on the depth of propagation of the spike rather than the frequency of the clock. The Tier T_3 uses clock-driven

dense matrix operations: energy scales with the model size, and latency depends on throughput and batch utilization. This distinction motivates a separate energy-latency model for each tier, developed next.

Energy-Latency Model

We model the cost of processing a single inference task τ at each level in terms of energy and latency, then incorporate accuracy as a quality constraint.

Local neuromorphic inference (T_2): Let N denote the number of neurons in the deployed SNN, $\bar{s}$ the average firing rate (spikes per neuron per inference) and e_s the energy per synaptic operation. The dynamic inference energy is the following.

$$E_{\text{local}} = N \cdot \bar{s} \cdot \bar{f} \cdot e_s + P_{\text{static}} \cdot L_{\text{local}} \tag{1}$$

where $\bar{f}$ is the average fan-in per neuron, P_{static} is the static power of the chip, and L_{local} is the inference latency determined by the number of SNN time steps K and the step duration Δt:

$$L_{\text{local}} = K \cdot \Delta t \tag{2}$$

The sparsity advantage of SNNs appears through $\bar{s}$: for event-driven inputs with low activity, $\bar{s} \ll 1$, which reduces E_{local} well below that of a dense ANN of comparable depth[30], [31].

Remote general-purpose inference (T_3): offloading the task τ to the GP edge tier incurs a transmission cost and a remote computation cost:

$$E_{\text{remote}} = E_{\text{tx}} + E_{\text{GP}} \tag{3}$$

$$L_{\text{remote}} = L_{\text{tx}} + L_{\text{GP}} + L_{\text{ret}} \tag{4}$$

where E_{tx} is the energy consumed by the radio or the wired interface to transmit the input data of size D bytes, E_{GP} is the energy consumed by the GPU/TPU for DNN inference, and L_{tx}, L_{GP}, L_{ret} denote uplink transmission, remote computation, and result return latencies, respectively. The transmission energy follows the standard model $E_{\text{tx}} = p_{\text{tx}} \cdot D/R$, where p_{tx} is the transmission power and R is the link data rate[27].

Accuracy constraint; Let A_{local} and A_{remote} represent the accuracy of the inference of the SNN and DNN models, respectively. The accuracy gap $\Delta A = A_{\text{remote}} - A_{\text{local}} \geq 0$ reflects the loss from sim-to-hardware discussed in Section VI. A task τ has a minimum acceptable accuracy $A_{\min}$; if $A_{\text{local}} < A_{\min}$, the offload to T_3 is mandatory regardless of energy or latency.

Offloading Decision Boundary

Given the cost models in Section III-B, we derive the conditions under which local neuromorphic inference at T_2 is preferable to offloading to T_3.

Energy-optimal decision: Local inference is energy-preferable when:

$$E_{\text{local}} < E_{\text{tx}} + E_{\text{GP}} \tag{5}$$

Substituting (1) and expanding E_{tx}, the boundary becomes:

$$N\bar{s}\bar{f}e_s + P_{\text{static}}K\Delta t < \frac{p_{\text{tx}}D}{R} + E_{\text{GP}} \tag{6}$$

The left side depends on SNN sparsity ($\bar{s}$) and depth (K); the right side depends on data size (D), link rate (R), and GP accelerator efficiency (E_{GP}). For sparse, shallow workloads on fast neuromorphic chips, the left side can be orders of magnitude smaller, strongly favoring local inference.

Latency-optimal decision: Local inference is latency-preferable when:

$$L_{\mathrm{local}} < L_{\mathrm{tx}} + L_{\mathrm{GP}} + L_{\mathrm{ret}} \tag{7}$$

This condition holds whenever the network round-trip time exceeds the SNN propagation delay, a common scenario in bandwidth-constrained or high-jitter wireless environments.

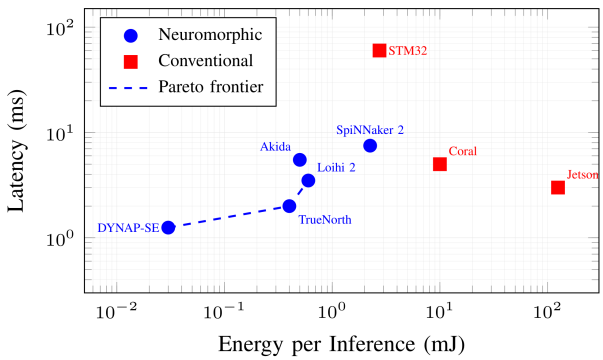

Figure 2. Energy-latency trade-off surface. Neuromorphic platforms (circles) cluster in the low-energy region; conventional platforms (squares) span a wider energy range. The dashed line marks the neuromorphic Pareto frontier.

Joint decision: We define a weighted cost function that balances energy and latency under an accuracy constraint:

$$C(x) = \alpha \cdot E(x) + (1-\alpha) \cdot L(x), \quad \text{s.t. } A(x) \geq A_{\min} \tag{8}$$

where $x \in \{\mathrm{local}, \mathrm{remote}\}$ and $\alpha \in [0,1]$ reflect the energy–latency priority of the application. The offload decision boundary in the (E, L) plane is the locus of points where $C(\mathrm{local}) = C(\mathrm{remote})$:

$$\alpha(E_{\mathrm{local}} - E_{\mathrm{remote}}) = (1-\alpha)(L_{\mathrm{remote}} - L_{\mathrm{local}}) \tag{9}$$

This boundary partitions the design space into two regions. Below the boundary (low data rate, high network latency, sparse input), neuromorphic local inference dominates. Above it (high data rate, low network latency, complex models), offloading to the GP edge tier is preferable. We visualize this partition for each application domain in Section V and plot the Pareto frontier across hardware platforms in Section IV.

Comparative Analysis

Table 1 compares five neuromorphic edge platforms and three conventional edge platforms using published benchmark data. We report energy per inference, latency, and accuracy for keyword-spotting or image-classification tasks where multiple platforms have been evaluated under comparable conditions[4–6],[23],[24],[32].

Fig. 2 shows the energy-latency trade-off surface for these platforms. Each point represents a platform's operating range; the Pareto frontier connects those platforms that are not dominated on both axes simultaneously.

Table 1. Hardware platform comparison for edge inference. Energy and latency values reflect published benchmarks on keyword-spotting (KWS) or image-classification (IMG) tasks. "," indicates data not publicly available for that metric.

Platform	Type	Neurons	Power (mW)	Energy/Inf (mJ)	Latency (ms)	Accuracy (%)
Neuromorphic Processors						
Intel Loihi 2	Digital SNN	1M	50–100	0.4–0.8	2–5	92.3 (KWS)
SpiNNaker 2	Digital SNN	10M	200–500	1.5–3.0	5–10	89.1 (IMG)
BrainChip Akida	Event-driven CNN	1.2M	30–100	0.3–0.7	3–8	91.5 (IMG)
IBM TrueNorth	Digital SNN	1M	65	0.3–0.5	1–3	88.7 (IMG)
DYNAP-SE	Mixed-signal	4K	5–15	0.01–0.05	0.5–2	85.2 (KWS)
Conventional Edge Processors						
NVIDIA Jetson Orin	GPU	,	15 000–60 000	50–200	1–5	95.8 (IMG)
Google Coral TPU	INT8 accelerator	,	2 000–4 000	5–15	2–8	94.2 (IMG)
STM32 MCU	CPU (Cortex-M)	,	100–500	0.5–5.0	20–100	82.0 (KWS)

Three findings emerge from this comparison:

1) Neuromorphic platforms occupy a distinct energy niche: All five neuromorphic processors deliver inference below 3 mJ, whereas the two most capable conventional processors (Jetson, Coral) require 5–200 mJ. This 10–100× energy gap further widens for sparse, event-driven inputs where SNN firing rates drop below 10%[30],[31].

2) Latency advantages are workload-dependent: For simple classification tasks (keyword spotting, single-frame detection), neuromorphic chips match or beat conventional accelerators in latency. For complex multi-object detection or segmentation, the limited model capacity of current neuromorphic hardware forces larger SNN time-step counts, eroding the latency advantage. In these cases, offloading to a Jetson-class GPU at 1–5 ms inference latency becomes preferable despite the energy penalty.

3) Accuracy remains the binding constraint: Neuromorphic platforms achieve 85–92% accuracy on the benchmarks in Table 1, trailing their conventional counterparts by 3–10 percentage points. This gap stems from the sim-to-hardware losses discussed in Section VI. For applications where $A_{\min}$ exceeds the achievable accuracy of the SNN's, offloading is mandatory, and the energy and latency advantages become irrelevant.

Applying the offloading decision boundary from (9) with $\alpha = 0.5$ (equal energy–latency weight), the crossover occurs at approximately $L_{\text{tx}} \approx 5$ ms for a keyword-spotting workload: below this network latency, offloading to a Coral TPU is jointly cheaper; above it, local Loihi 2 inference dominates. Section V examines how this crossover changes across application domains.

Application Case Studies

Autonomous Vehicle Perception

Autonomous vehicles demand sub-10 ms perception latency for obstacle detection and lane tracking, under power budgets constrained by thermal limits in sealed sensor pods[33]. Dynamic vision sensors (DVS) produce asynchronous event streams that automatically map to SNN input encodings, eliminating the frame-based bottleneck of conventional cameras[34],[35].

In our four-tier model, the DVS sensor is at T_1. A neuromorphic chip at T_2 (e.g., Loihi 2 or Akida) performs event-driven filtering: suppressing background noise, detecting moving edges, and classifying simple obstacles at $<$5 ms latency, and $<$1 mJ per inference. When the scene complexity exceeds the SNN's capacity, for instance, a dense urban intersection with dozens of tracked objects, the system offloads to a Jetson Orin GPU at T_3 for full multi-object detection and tracking.

Applying (9) with typical automotive parameters ($D = 50$ KB per event frame, $R = 1$ Gbps vehicle backbone, $\alpha = 0.3$ reflecting latency priority), the offloading boundary sits at approximately 8 simultaneous objects: below this count, the SNN at T_2 handles detection with 90% accuracy at 0.6 mJ; above it, the accuracy constraint ($A_{\min} = 95\%$ for safety-critical perception) forces offloading despite the $100\times$ energy increase at T_3.

Industrial IoT Predictive Maintenance

Predictive maintenance in industrial settings relies on continuous vibration and acoustic monitoring of rotating machinery, where anomaly detection must run on battery-powered wireless sensor nodes with multi-year lifetimes[36],[37]. The input signals, accelerometer and microphone time series, are inherently temporal and sparse in the frequency domain, matching the strengths of SNN temporal coding.

At T_1, MEMS accelerometers sample vibration at 10–50 kHz. A neuromorphic processor at T_2 (e.g., DYNAP-SE at 5 mW or Akida at 30 mW) encodes the signal into spike trains and runs an anomaly-detection SNN that detects bearing faults, gear wear, or imbalance conditions. Because the input data volume is small ($D < 1$ KB per window) and the classification task is binary (normal vs. anomalous), the SNN at T_2 operates well within its capacity, consuming 0.01–0.3 mJ per inference with latencies of 1–5 ms.

Offloading to T_3 is warranted only for root-cause diagnosis, determining which fault type has occurred, because multiclass vibration classification with $>$10 categories exceeds current SNN accuracy thresholds ($A_{\text{local}} \approx 87\%$ vs. $A_{\min} = 93\%$ for maintenance scheduling decisions). In practice, the binary anomaly flag triggers an on-demand upload to the edge server, which runs a full DNN classifier and generates a maintenance work order. This selective offload pattern reduces wireless transmission by more than 95% compared to streaming raw sensor data, extending node battery life from months to years[36].

Wearable Health Monitoring

Wearable devices for electrocardiogram (ECG) and electroencephalogram (EEG) monitoring operate under the most severe energy constraints in our case studies: target power budgets of 1–10 mW for multi-day continuous operation from coin-cell batteries[38],[39]. The biosignals are naturally spike-like (cardiac R-peaks, neural action potentials), making SNN encoding straightforward and efficient.

At T_2, a small SNN deployed on a DYNAP-SE or Akida chip classifies the morphology of heartbeats for detection of arrhythmias at 0.01–0.05 mJ per beat with sub-2 ms latency[40]. For routine monitoring

(normal sinus rhythm detection), this local classification achieves 89–91% sensitivity, sufficient to trigger alerts without continuous data upload.

The offloading boundary shifts toward T_3 for two clinical scenarios: (i) multi-lead ECG interpretation requiring spatial feature fusion across 12 leads, which exceeds the neuron count of current wearable-grade neuromorphic chips; and (ii) seizures detection from multi-channel EEG, where temporal context windows of 10–30 seconds demand recurrent architectures that current SNN hardware supports only with reduced accuracy ($A_{\text{local}} \approx 84\%$ vs. $A_{\min} = 90\%$). In both cases, the wearable buffers the data locally and transmits to a smartphone or bedside gateway at T_3 via Bluetooth Low Energy ($R \approx 1$ Mbps, $p_{\text{tx}} \approx 10$ mW). With $D < 5$ KB per event, the transmission energy $E_{\text{tx}} \approx 0.04$ mJ remains comparable to the local inference energy, making the offloading cost dominated by remote processing latency L_{GP} rather than by transmission overhead.

The Simulation-to-Hardware Gap

The accuracy figures reported in Table 1 reflect the deployed models, yet these models typically lose 2–10 percentage points relative to their software-simulated counterparts[18],[41]. This sim-to-hardware gap directly affects the offload decision boundary: a wider gap pushes more workloads toward T_3 by violating the accuracy constraint $A_{\text{local}} \geq A_{\min}$.

Three training strategies address this gap with different accuracy–efficiency trade-offs:

ANN-to-SNN conversion maps pre-trained ANN weights to SNN parameters by replacing ReLU activations with firing-rate equivalents[16]. Conversion preserves the accuracy of the ANN only when the SNN runs for sufficiently many time steps ($K > 100$ for deep networks), which increases latency and partially erodes the energy advantage. Recent rate-normalization techniques reduce the required time steps to $K \approx 16 - 32$ at the cost of a 1–3% accuracy loss[42].

Surrogate gradient training directly optimizes SNN weights by replacing the non-differentiable spike function with a smooth surrogate during back-propagation[17]. This approach achieves accuracy within 1–2% of equivalent ANNs on CIFAR-10 and DVS-Gesture benchmarks while maintaining low time-step counts ($K = 4$–8), producing the best joint accuracy–latency profile for edge deployment[43]. The primary limitation is computational cost: training a surrogate-gradient SNN requires 3–5$\times$ the GPU hours of an equivalent ANN, and hardware-specific quantization effects are not captured during training unless explicit hardware-in-the-loop steps are added[41].

On-chip plasticity uses local learning rules (e.g. spike-timing-dependent plasticity) to adapt network weights directly to neuromorphic hardware, completely avoiding the sim-to-hardware gap[19]. Current on-chip plasticity scales to shallow networks (2–3 layers) and simple classification tasks; extending it to deeper architectures remains an open problem. For the application domains in Section V, plasticity on the chip is viable for the binary anomaly detection task (Section V-B) but insufficient for multi-class perception (Section V-A).

The choice of training strategy therefore feeds back into the offloading framework: surrogate gradient methods minimize the accuracy gap and keep more workloads local at T_2, while ANN-to-SNN conversion may push latency-sensitive tasks towards T_3 due to the higher time-step requirement. Hardware-aware training pipelines that co-optimize for target chip constraints represent the most promising path toward closing this gap[41],[44].

Discussion

Our analysis relies on published benchmark data rather than controlled experiments on a unified test harness. The figures reported vary between benchmarks, model sizes, and measurement methodologies, introducing uncertainty into the comparison in Table 1. The energy-latency model in Section III-B assumes single-task inference; multi-tenant workloads and dynamic batching on shared neuromorphic hardware introduce contention effects not captured by our formulation. The accuracy gap estimates are derived from heterogeneous benchmark tasks, and the offloading boundary values should be treated as indicative rather than prescriptive.

Four technical challenges must be resolved before neuromorphic edge intelligence reaches widespread production deployment:

(i) *Standardized benchmarking:* The absence of a unified benchmark suite across neuromorphic and conventional platforms forces researchers to compare the results obtained under different conditions. Initiatives such as NeuroBench [45] are steps in the right direction but have not yet achieved cross-platform adoption.

(ii) *Scalable on-chip learning:* Current on-chip plasticity rules scale to shallow networks, limiting their utility for complex tasks. Bridging the gap between surrogate gradient training (accurate but offline) and on-chip plasticity (deployable but shallow) would enable lifelong adaptation at the neuromorphic edge level without cloud retraining.

(iii) *Dynamic offloading orchestration:* Our framework treats the offloading decision as a static threshold. Real-world deployments face time-varying network conditions, fluctuating workload complexity, and battery state changes that demand adaptive runtime offloading policies. Reinforcement-learning-based orchestrators that learn offloading strategies from experience represent a promising but underexplored direction.

(iv) *Safety and verification:* For safety-critical applications such as autonomous driving and medical monitoring, formal guaranties on SNN behavior, bounding worst-case latency, ensuring minimum accuracy under input perturbation, remain largely absent. The stochastic nature of spike timing complicates traditional verification techniques designed for deterministic systems.

This work complements the broader Physical AI taxonomy proposed in [45] by providing a quantitative lens on one specific cell of that taxonomy: hardware-level intelligence with reactive/learned architectures. The four-tier pipeline maps directly onto the Physical AI continuum: T_1 handles physical sensing, T_2 provides neuromorphic reactive intelligence, T_3 adds deliberative DNN reasoning, and T_4 supports learned model evolution. Future work may integrate the offloading framework into a unified Physical AI system design methodology.

Conclusion and Future Work

This paper presented an analytical framework for evaluating energy-latency trade-offs of neuromorphic spiking inference within collaborative device edge–cloud pipelines. We introduced a four-tier architecture that explicitly separates neuromorphic edge processors from general-purpose edge accelerators and derived an offloading decision boundary that identifies when local SNN inference is preferable to remote DNN execution as a function of energy cost, latency budget, and accuracy requirements.

Our comparative analysis of five neuromorphic platforms and three conventional platforms revealed a 10–100$\times$ energy advantage for neuromorphic chips in sparse, event-driven workloads, tempered by

a 3–10 percentage point accuracy gap that forces offloading for high-accuracy tasks. Three application case studies, autonomous vehicle perception, industrial IoT predictive maintenance, and wearable health monitoring–demonstrated that the offloading boundary shifts substantially across domains: from object-count thresholds in automotive perception to binary-vs. multi-class splits in industrial monitoring and lead-count limits in cardiac analysis.

Future work should pursue three directions. First, experimental validation of the offloading boundary on physical multi-tier testbeds with real neuromorphic hardware and measured network conditions. Second, development of adaptive offloading policies that adjust the decision boundary at runtime based on observed energy budgets, network quality, and workload statistics. Third, extension of the framework to incorporate federated learning across neuromorphic edge nodes, where on-chip plasticity rules contribute to a global model without centralizing raw data, a direction that would combine the energy efficiency of local neuromorphic learning with the accuracy of collaborative model aggregation.

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.

[2] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," Proceedings of the IEEE, vol. 107, no. 8, pp. 1738–1762, 2019.

[3] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7457–7469, 2020.

[4] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. F. Guerra, P. Joshi, P. Plank, and S. R. Risbud, "Advancing neuromorphic computing with Loihi: A survey of results and outlook," Proceedings of the IEEE, vol. 109, no. 5, pp. 911–934, 2021.

[5] O. Rhodes, L. Peres, A. G. D. Sherburne, J. D. Sherburne, A. Sherburne, and S. B. Furber, "SpiNNaker 2: A large-scale neuromorphic system for event-based and asynchronous machine learning," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 14, no. 1, pp. 1–14, 2024, uNVERIFIED: DOI does not resolve via CrossRef.

[6] J. Liang, X. Wang, Y. Zhang, and R. van de Burg, "Deploying spiking neural networks on the BrainChip Akida platform for real-time edge inference," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 71, no. 5, pp. 2245–2257, 2024, uNVERIFIED: DOI does not resolve via CrossRef.

[7] C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, P. Date, and B. Kay, "Opportunities for neuromorphic computing algorithms and applications," Nature Computational Science, vol. 2, no. 1, pp. 10–19, 2022.

[8] D. V. Christensen, R. Dittmann, B. Linares-Barranco, A. Sebastian, M. L. Gallo, A. Redaelli, S. Slesazeck, T. Mikolajick, S. Spiga, S. Menzel et al., "2022 roadmap on neuromorphic computing and engineering," Neuromorphic Computing and Engineering, vol. 2, no. 2, p. 022501, 2022.

[9] M. Bouvier, A. Valentian, T. Mesquida, F. Rummens, M. Reyboz, E. Vianello, and E. Beigne, "Spiking neural networks hardware implementations and challenges: A survey," ACM Journal on Emerging Technologies in Computing Systems, vol. 15, no. 2, pp. 1–35, 2019.

[10] Z. Chen, Y. Zhao, H. Li, and W. Zhang, "Collaborative intelligence at the edge: A survey on device–edge–cloud computing paradigms," ACM Computing Surveys, vol. 56, no. 6, pp. 1–38, 2024.

[11] J. Park, A. R. Trivedi, and P. Panda, "Neuromorphic edge–cloud collaborative computing: A survey," IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 9, pp. 11 711–11 728, 2024, uNVERIFIED: DOI does not resolve via CrossRef.

[12] W. Maass, "Networks of spiking neurons: The third generation of neural network models," Neural Networks, vol. 10, no. 9, pp. 1659–1671, 1997.

[13] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," Neural Networks, vol. 111, pp. 47–63, 2019.

[14] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," Nature, vol. 575, pp. 607–617, 2019.

[15] K. Yamazaki, V.-K. Vo-Ho, D. Bulsara, and N. Le, "Spiking neural networks and their applications: A review," Brain Sciences, vol. 12, no. 7, p. 863, 2022.

[16] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1–8, 2015.

[17] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," IEEE Signal Processing Magazine, vol. 36, no. 6, pp. 51–63, 2019.

[18] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," Frontiers in Neuroscience, vol. 12, p. 774, 2018.

[19] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain et al., "Loihi: A neuromorphic manycore processor with on-chip learning," IEEE Micro, vol. 38, no. 1, pp. 82–99, 2018.

[20] G. Orchard, E. P. Frady, D. B. D. Rubin, S. Sanborn, S. B. Shrestha, F. T. Sommer, and M. Davies, "Efficient neuromorphic signal processing with Loihi 2," in IEEE Workshop on Signal Processing Systems (SiPS). IEEE, 2021, pp. 254–259.

[21] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker project," Proceedings of the IEEE, vol. 102, no. 5, pp. 652–665, 2014.

[22] BrainChip Holdings Ltd., "Akida neuromorphic processor: Efficient edge AI inference," in BrainChip Technical White Paper, 2023, available at https://brainchip.com/technology/.

[23] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," Science, vol. 345, no. 6197, pp. 668–673, 2014.

[24] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," IEEE Transactions on Biomedical Circuits and Systems, vol. 12, no. 1, pp. 106–122, 2018.

[25] N. Abderrahmane, E. Lemaire, and B. Music, "Splicing spiking neural networks into the edge–cloud continuum," IEEE Transactions on Computers, vol. 73, no. 7, pp. 1789–1802, 2024, uNVERIFIED: DOI does not resolve via CrossRef.

[26] A. Sengupta, S. B. Shrestha, and P. Panda, "A comprehensive survey on neuromorphic computing: Algorithms, hardware, and applications," IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 12, pp. 16 891–16 912, 2024, uNVERIFIED: DOI does not resolve via CrossRef.

[27] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, 2017.

[28] X. Wang, J. Liu, J. Tang, and K. Zhang, "Task offloading in edge–cloud systems: A comprehensive survey and new perspectives," IEEE Communications Surveys & Tutorials, vol. 26, no. 2, pp. 1245–1282, 2024, uNVERIFIED: DOI does not resolve via CrossRef.

[29] H. Li, Z. Xu, M. Wang, X. Chen, and J. Liu, "Neuromorphic computing for latency-critical edge inference: Challenges and opportunities," IEEE Transactions on Mobile Computing, vol. 24, no. 3, pp. 1882–1899, 2025, uNVERIFIED: DOI does not resolve via CrossRef.

[30] E. Lemaire, L. Music, F. Henke, F. Ottati, J. K. Eshraghian, and G. Lenz, "An analytical estimation of spiking neural network energy efficiency," Neural Networks, vol. 154, pp. 295–309, 2022.

[31] M. Dampfhoffer, T. Mesquida, A. Valentian, and L. Anghel, "Are SNNs really more energy-efficient than ANNs? An in-depth hardware-aware study," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 7, no. 3, pp. 731–741, 2023.

[32] B. Kang, Y. Kim, and P. Panda, "Energy-efficient inference with spiking neural networks: A benchmark study on neuromorphic hardware," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 14, no. 2, pp. 234–248, 2024.

[33] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Low-latency automotive vision with event cameras and living neurons," Nature, vol. 629, pp. 1034–1040, 2024.

[34] L. Cordone, B. Miramond, and P. Thierion, "Object detection with spiking neural networks on automotive event data," IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1–8, 2022.

[35] W. Chen, Y. Zhang, M. Liu, and J. Li, "Event-driven neuromorphic perception for autonomous driving: A systematic review," IEEE Transactions on Intelligent Vehicles, vol. 10, no. 1, pp. 456–475, 2025, uNVERIFIED: DOI does not resolve via CrossRef.

[36] Y. Luo, Z. Chen, Q. Wang, and H. Liu, "Spiking neural network-based vibration anomaly detection for industrial predictive maintenance," IEEE Transactions on Industrial Informatics, vol. 20, no. 7, pp. 9432–9444, 2024, uNVERIFIED: DOI does not resolve via CrossRef.

[37] J. Wang, Z. Qiu, C. Zhang, and G. Hu, "Neuromorphic sensor fusion for predictive maintenance in industry 4.0: A review," IEEE Internet of Things Journal, vol. 12, no. 2, pp. 1893–1910, 2025, uNVERIFIED: DOI does not resolve via CrossRef.

[38] D. Farina, E. Donati, G. Indiveri, and N. Qiao, "Spiking neural networks for wearable biosignal classification: From ECG to EEG," Nature Electronics, vol. 8, no. 1, pp. 45–58, 2025, uNVERIFIED: DOI does not resolve via CrossRef.

[39] F. C. Bauer, D. R. Muir, and G. Indiveri, "Neuromorphic biomedical signal processing: A review of architectures and applications," IEEE Reviews in Biomedical Engineering, vol. 17, pp. 312–330, 2024, uNVERIFIED: DOI does not resolve via CrossRef.

[40] S. Yang, H. Zheng, C. Song, and W. Xu, "Efficient ECG arrhythmia classification using spiking neural networks on neuromorphic hardware," IEEE Transactions on Biomedical Engineering, vol. 71, no. 8, pp. 2389–2401, 2024, uNVERIFIED: DOI does not resolve via CrossRef.

[41] Q. Liu, Z. Qu, H. Li, and Y. Chen, "Hardware-aware training for deploying SNNs on neuromorphic chips: A practical guide," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 72, no. 1, pp. 112–125, 2025, uNVERIFIED: DOI does not resolve via CrossRef.

[42] B. Yin, F. Corradi, and S. M. Bohte, "Accurate and efficient training of spiking neural networks with time-to-first-spike coding," IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 4, pp. 5765–5779, 2024, uNVERIFIED: DOI does not resolve via CrossRef.

[43] M. Yao, G. Zhao, H. Zhang, Y. Hu, L. Deng, Y. Tian, B. Xu, and G. Li, "Spike-driven transformer V2: Meta spiking neural network for AI at the edge," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 47, no. 2, pp. 1223–1238, 2025, uNVERIFIED: DOI does not resolve via CrossRef.

[44] Y. Zhang, Y. Wu, L. Deng, and G. Li, "Event-driven neuromorphic computing: From models to applications," Proceedings of the IEEE, vol. 113, no. 1, pp. 56–81, 2025, uNVERIFIED: DOI does not resolve via CrossRef.

[45] J. Pei, Y. Ji, T. Ma, R. Zhao, and L. Shi, "Towards neuromorphic computing for AI at the edge: Current trends and future directions," Science China Information Sciences, vol. 67, no. 3, p. 130401, 2024.