Energy-Delay Product as an Early-Decision Metric for Real-Time SNNs

Eeman Fatima*, Ali Muhtaroglu †‡
* ACIT Biomedical Engineering MS Program † Dept. of Mechanical, Electrical and Chemical Engineering
‡ Advanced Health Intelligence and Brain-Inspired Technologies (ADEPT)
Faculty of Technology, Art and Design (TKD)
Oslo Metropolitan University, PO Box 4, St. Olavs plass, 0130 Oslo, Norway

***Abstract*—The escalating energy demands of deep learning have prioritized the development of energy-efficient neuromorphic hardware for the edge. However, early-stage architectural evaluation is often hindered by fragmented metrics that fail to balance metabolic costs with real-time latency constraints. This paper proposes a comprehensive evaluation framework for Spiking Neural Networks (SNNs) based on the Energy–Delay Product (EDP). Utilizing a custom event-driven C++ emulator, we model energy via spike-activity proxy and delay via total computation steps, enabling "what-if" analysis of critical design features. Validated through a context-dependent reinforcement learning (RL) task, results demonstrate that migrating from 32-bit to 5-bit precision reduces EDP by over an order of magnitude, while structural synaptic pruning achieves significant efficiency gain without degrading classification accuracy for scenarios when unknown real-time input variations may necessitate an initially redundant network structure. Quantitative trends reveal a minimum L=100 input sequence length in a given problem for amortizing learning overhead. These findings align with established neuromorphic literature, confirming the proposed emulator as a viable tool for informed architectural decision-making prior to costly physical implementation.**

***Index Terms*—Spiking Neural Networks, energy efficiency, energy–delay product, reinforcement learning, neuromorphic computing, edge, design metrics.**

I. Introduction

The surge in deep learning computational demand has caused a critical energy crisis in modern computing [1], [2]. Traditional von Neumann architectures are constrained by a "memory wall," where data movement between separated processing and memory units incurs high energy penalties [3], [4]. Neuromorphic computing, inspired by the biological brain, offers a low-power alternative for autonomous edge applications [5]–[7]. Central to this approach are Spiking Neural Networks (SNNs) — the third generation of neural models [8]. By emulating event-driven biological signaling via discrete spikes, SNNs leverage temporal sparsity to significantly reduce power consumption [9]–[11]. This bio-inspired direction is supported by evidence that the majority of brain energy is dedicated to neural signaling, specifically action potentials and synaptic transmission [12]–[14].

Evaluating neuromorphic energy efficiency is challenging due to fragmented metrics [15], [16]. Current studies often rely on isolated parameters, such as spike counts or firing rates, which overlook execution time and workload variability [17], [18]. While power metrics ignore throughout, energy-per-spike or energy-per-synapse [19]–[21] focus on local events, neglecting peripheral communication and control overhead. Similarly, energy per Synaptic Operation (SOP) [22], [23] fails to account for learning rule complexity or the precision required for accuracy [16], [24]. Consequently, selecting optimal hardware remains difficult without comprehensive, early-stage evaluation frameworks [16].

Neuromorphic processors are increasingly deployed for real-time classification in healthcare monitoring and robotic rehabilitation [9], [25]–[27]. SNN-based reinforcement learning (RL) systems can achieve context-dependent learning with significantly lower energy consumption than rate-based deep learning [9], [11], yet latency and runtime are equally critical [9], [28], [29]. Evaluating energy in isolation is insufficient, and may mislead design for the strict real-time constraints of autonomous navigation or closed-loop prosthetic control [27], [30], [31]. This necessitates a metric that captures the trade-off between energy and speed.

The Energy-Delay Product (EDP), a standard microprocessor optimization metric that penalizes low power at high latency [1], [32], suits neuromorphic designs, where energy and responsiveness hinge on spiking dynamics [8], [33]. Utilizing an early EDP paradigm enables evaluation across architectural features, such as neuron models, synaptic densities, or learning rules, prior to costly physical implementation [34]–[37].

This paper proposes an EDP-based model for real-time edge SNNs, derived from measurable simulation variables: spike activations and computational steps. Energy is modeled as accumulated activity, where each spike activation represents a sequence of power-dissipating events, and delay as task computational steps. We utilize average EDP per input pattern to quantify architectural merit during early design. Using a spiking RL system, we analyze how context change rate, sequence length, numerical resolution, and synaptic pruning impact energy–delay trade-offs. The key contributions are:

- A custom C++ SNN emulator for early-cycle architectural evaluation that reports spike activations, computational steps, and average EDP;
- Demonstration of accuracy and EDP analysis against operating conditions, such as the rate of context change,
- Sample trade-off analysis to guide early design, including computational precision and synaptic pruning feature.

The remainder of this paper is organized as follows: Section II discusses early design decisions requiring EDP-focused metrics. Section III details the modeling methods, emulation framework, and experimental design. Section IV presents sample analytical results from the custom C++ emulator using a "traveling mouse" navigation problem. Finally, Section V provides conclusions.

II. EDP Utility at Early Design Phase

Adopting the EDP early in the custom neuromorphic processor design cycle enables a suite of "what-if" studies that resolve critical tensions in hardware feature selection.

A. Neural Dynamics and Thresholding Strategies: Neuron dynamics govern efficiency by shaping firing patterns and latency [38]. Features like dynamic thresholds or adaptive leak rates reduce the time to first spike [28], [39]. EDP evaluation determines if the reduced classification time justifies the circuit overhead — a critical balance for real-time tasks where latency is as detrimental as high power [27], [31].

B. Structural Sparsity and Pruning Evaluation: Neuromorphic systems exploit structural sparsity to reduce event traffic [15], [40]. Pruning weak synapses lowers inference energy, but overpruning can impede adaptability and slow down context switching [18], [41]. EDP pinpoints a connectivity sweet spot that preserves efficiency and rapid decision-making for navigation and control [31].

C. Context-Dependent Plasticity and Stability: Plasticity via STDP [21], [23], [42] or reward-modulated learning [43], [44] enables adaptation, yet RL with context-dependent actions for identical stimuli faces the stability-plasticity dilemma [45], [46]. EDP-based what-if analyses assess whether reduced spiking architectures [9] preserve adaptation while curbing energy-intensive noise and unnecessary synaptic updates [18].

III. Method

Metabolic constraints promote efficient spike-based coding via sparse, event-driven, and low-precision representations [12], [47], [48], while local and reward-modulated plasticity support online RL under real-time limits [9], [42], [43]. Because efficiency also depends on sparsity, precision, and non-stationarity [31], [48], this study uses EDP to quantify how activity, runtime, input length, precision, pruning, and context switching jointly set energy–latency efficiency [32].

A. Fast SNN emulator architecture

We use an event-driven C++ SNN emulator to assess learning dynamics and energy efficiency. By logging spike generation, synaptic transmission, and plasticity updates, it estimates spike activity and execution step count for energy and delay proxies, respectively [14], [17]. Neurons follow a Leaky Integrate-and-Fire (LIF) model to balance biological plausibility with efficient weight updates and real-time simulation [38]. Fig. 1 depicts the architecture.

B. Environment and Task: Traveling Mouse Problem

We demonstrate EDP-based analysis with the Traveling Mouse problem, a context-dependent RL task adapted from [9], [11]. The agent navigates four locations across two contexts (A and B), as shown in Fig. 2(a). Triplets of location, context, and item (X or Y) map to unique sensory neurons (Fig. 2(b)), and the agent chooses move or dig. Because rewards depend on specific triplets, the network must learn context–item associations rather than simple stimulus–response mappings [45].

C. Network Architecture and Dynamics

The SNN comprises sensory encoding, a hippocampus-inspired associative memory, and a motor layer. Fixed inhibitory connections enforce winner-take-all (WTA) dynamics in the hippocampus and motor stages, reducing computational load [39]. To stabilize low-precision operation, randomized inhibitory prioritization limits simultaneous firing and lowers exploration latency [9].

D. Learning Procedure

We use RL with alternating behavioral and replay phases. Rewarded action sequences induce synaptic potentiation, whereas unsuccessful sequences undergo depression during replay. Replay sequentially reactivates stored activity to enable

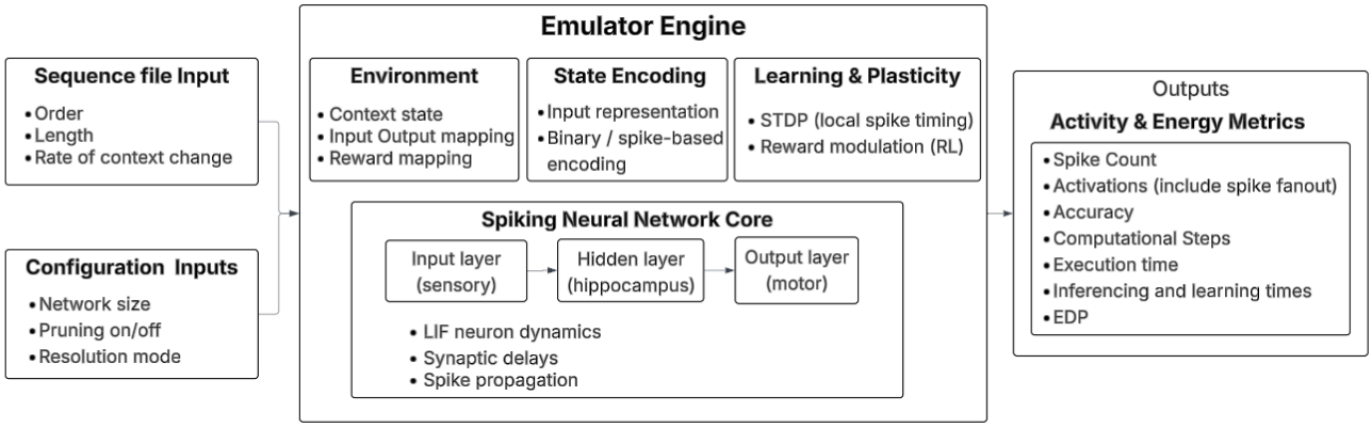

Figure 1: Block diagram of SNN emulator architecture.

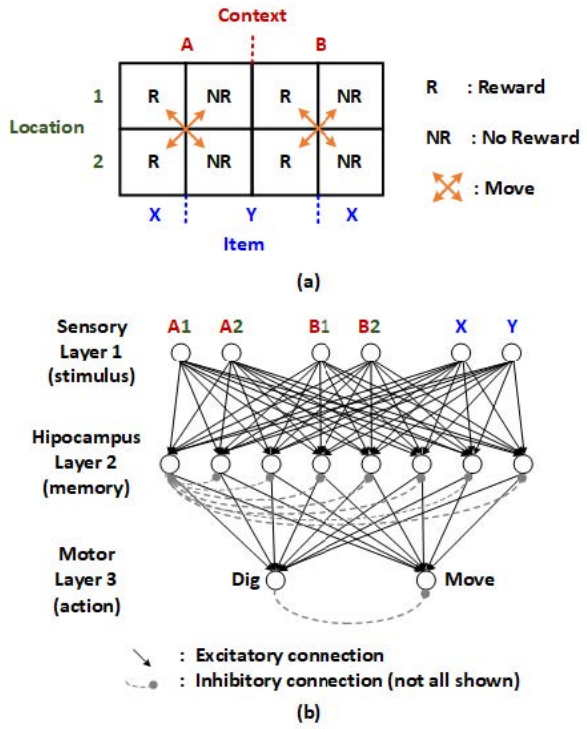

Figure 2: Illustration of (a) the context dependent task, and (b) SNN to realise RL for the given task [9].

temporal credit assignment [43], [44]. This yields a hybrid scheme in which RL modulates unsupervised baseline dynamics [22], [42].

E. Energy Model Based on Spike Activity

Energy consumption is modelled from synapse level activation, reflecting the dominance of action potentials and synaptic transmission in biological cost [12], [49]. Each receiver activation counts as one unit. Let $N_{\text{activations}}$ be total receiver activations and L the input sequence length. Total energy E satisfies:

$$E \propto N_{activations}. \tag{1}$$

Energy per input is normalized as:

$$E/Input \propto N_{activations}/L. \tag{2}$$

The proportionality scales with synapse count, fanout, and numerical precision, following FPGA studies [9]. Normalization enables comparison across input size and high level design characteristics.

F. Execution Time and Energy–Delay Product

Computational steps (N_{steps}) count the temporal cost of spike propagation and learning computations across architectural configurations. Receiver activations per episode serve as the primary activity-based energy proxy [17]. To couple activity with latency, we define EDP per input as:

$$EDP \propto \frac{N_{activations}}{L} \times \frac{N_{steps}}{L}. \tag{3}$$

R-STDP weight-update operations are excluded from $N_{\text{activations}}$ because learning engines run in parallel with feedforward inference in dedicated neuromorphic hardware [42], [43]. Normalization by input length ensures fair comparison across sequence lengths and resolutions. Consistent with low-power practice, this metric favors architectures that minimize activations and steps per input [1], [32].

G. Experimental Protocol

Experiments span variations in input length L, temporal resolution, learning paradigm, and pruning. For each configuration, we record $N_{\text{activations}}$, N_{steps}, classification accuracy, and EDP. We introduce mid-experiment context changes to assess adaptability and relearning efficiency [9], [45] and modulate the rate of unlabelled inputs to test stability. This protocol systematically evaluates energy–performance trade-offs under non-stationary conditions, including irregular sequence lengths and varying context-switch rates, beyond static benchmarks.

IV. Results

Fig. 3 illustrates energy (activations) and runtime (computational steps) proxies along with EDP, averaged per input sequence; in hardware, these can map to units such as *nJ*, *ns*, and *nJ.ns*. The fully connected RL SNN has 12 (input) – 24 (hidden) – 2 (output) neurons (overprovisioned for the task), 32-bit resolution, pruning disabled, and a 20% context-change rate. Context changes briefly increase energy and latency due to replay-driven learning, yet switching between related contexts lowers the average EDP per input. Classification accuracy improves from 60% to 80% (Fig. 4), which is the maximum attainable under the chosen context-change statistics.

A. Rate of Context Change

Frequent context changes result in EDP increase due to additional energy spent on learning for a typical input sequence. This phenomenon is depicted in Fig. 5 for a pattern size of 10000. At all tested rates the classification accuracy follows Accuracy $\sim (1-r) \times 100\%$, where r is the context-change learning rate.

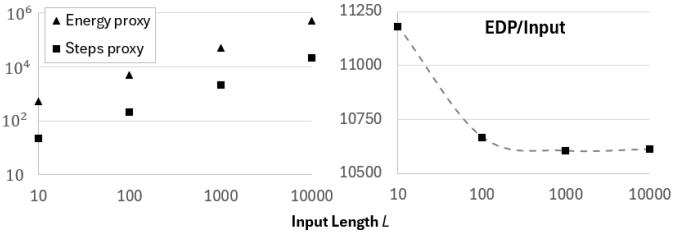

Figure 3: Trends of spike count adjusted by design properties (energy) and adjusted computational step (latency) on the left (log scale), and average EDP on the right, against no. of input sequences, L (log scale), for 12-24-2 RL SNN, 32-bit precision, no pruning.

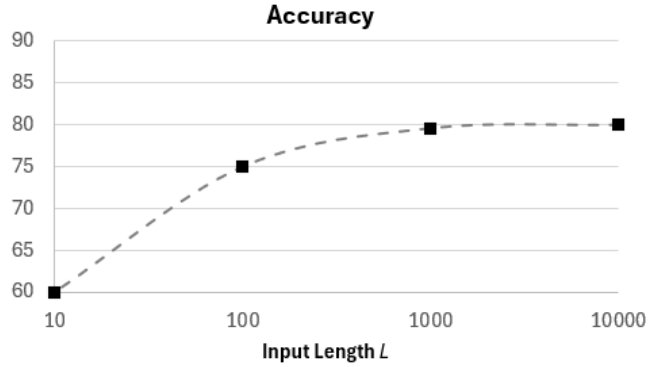

Figure 4: Accuracy trend for logarithmically changing pattern size with configured 20% context switching rate in each case.

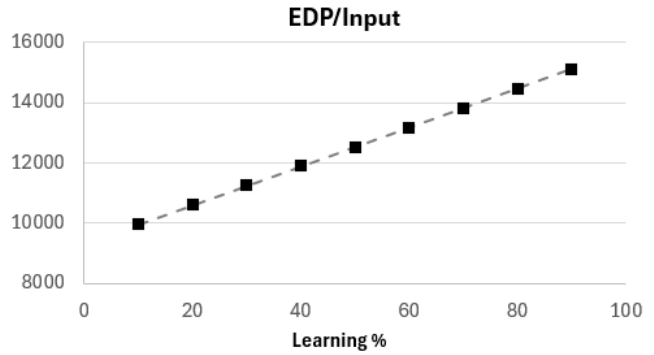

Figure 5: Overall effect of % of context change on aveage EDP for a pattern size of 10000, 32-bit precision, no pruning.

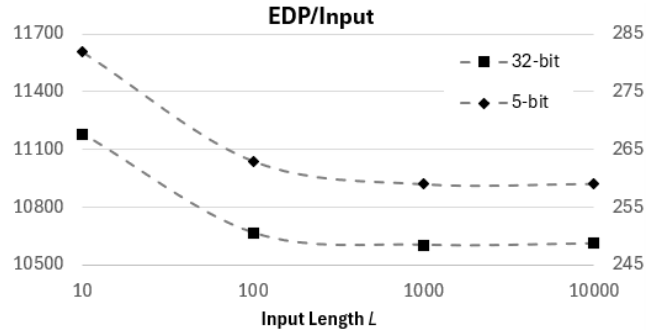

Figure 6: Effect of computation precision on EDP per input across different *L* (12-24-2 RL SNN with no pruning) – 32-bit (left axis) and 5-bit (right axis) comparison.

B. Computation Precision

Computation precision systematically impacts energy efficiency. Across pattern sizes, 32-bit precision consistently incurs higher activations/input than 5-bit, even when spiking activity is similar, because delay and power scale with hardware width. Fig. 6 plots 32-bit EDP per input (left axis) and 5-bit (right), showing an ($\sim 41\times$) reduction with 5-bit. The 5-bit design employs uniform fixed-point quantization: FP32 weights are linearly mapped to 5-bit unsigned integers in $[0, 31]$, with a scale factor of 1.0 (versus 6.4 at 32-bit), reflecting reduced switching-capacitance energy [1]. Accuracy remains unchanged at the tested sequence lengths.

C. Synaptic Pruning

When pruning is enabled, redundant synapses are removed once inputs yield no new combinations, reducing the SNN from 12 – 24 – 2 to 6 – 8 – 2 neurons (with corresponding synapses), which suffices for all Travelling Mouse contexts (Fig. 2). Hardware emulation uses clock gating of unused paths. Pruning leaves the runtime proxy essentially unchanged since the layer activation schedule is identical (bottom panel, Fig. 7). In contrast, pruning substantially reduces receiver activations in proportion to the $\sim 63\%$ of disabled neurons and synapses (middle panel), driving an EDP drop from $10,600$ (pruning OFF) to $3,900$ (pruning ON) at $L{=}10,000$ (top panel). Accuracy is unaffected across tested configurations. R-STDP weight-update operations are excluded from both activation and step counts because dedicated neuromorphic hardware performs learning in parallel with feedforward inference and off the critical path [42], [43]. Including learning overhead would increase both proxies proportionally without changing relative rankings, but it renders energy estimates optimistic and should be quantified in future hardware studies.

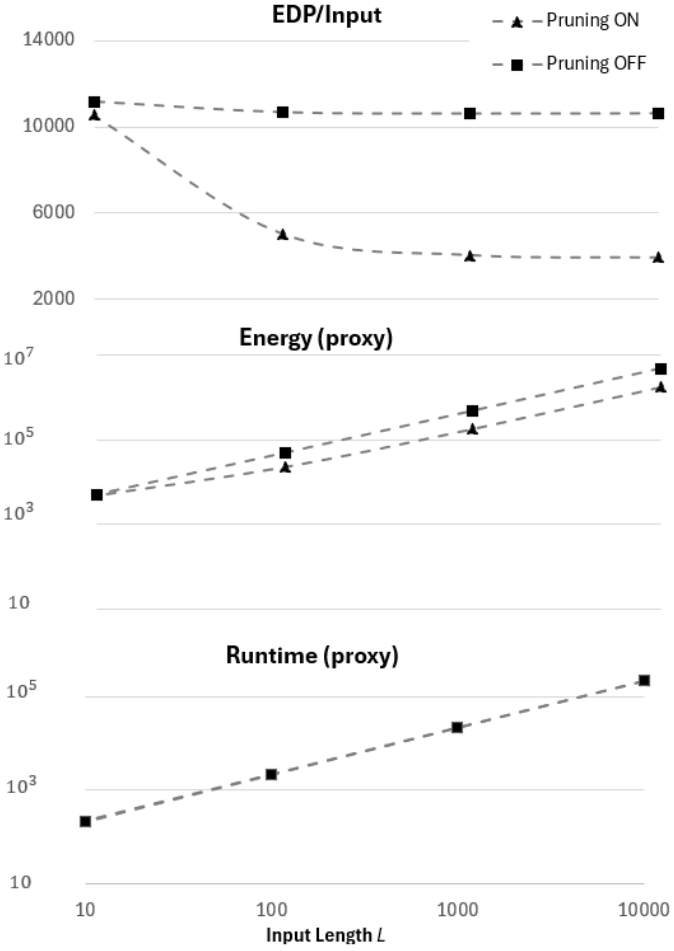

Figure 7: Effect of architectural pruning feature (clock disabling) on ET proxy (bottom), Energy proxy (middle) and EDP per input sequence (top) for a logarithmic range *L*.

V. Discussion

This study examines spiking RL efficiency by relating receiver activations, computational steps, and architectural constraints. We evaluate three proxies: total receiver activations adjusted for physical design changes (e.g. precision and pruning), total sequential layer-activation steps during learning and inference adjusted accordingly, and the normalized Energy–Delay Product (EDP) per input sequence.

A. Energy as Integrated Spiking Activity

Energy tracks receiver activations over time. In Fig. 3 (left), both energy and runtime proxies scale linearly with input length on a log-log plot, indicating stable average activation rates. Thus, raw energy is driven by workload volume, not architectural non-linearities [17]. Efficiency gains should therefore target activations per operation in addition to latency of operations.

B. EDP and Amortization of Learning Overhead

Normalized EDP per input is non-monotonic: it peaks at short sequences ($L = 10$) and drops by 5% at $L = 100$ before plateauing, matching the accuracy rise from 60% to 80% for 20% context change (Fig. 3 right; Fig. 4). The initial EDP reflects warm-up costs from plasticity, and beyond $L = 100$, learning overhead is amortized [32]. While per-input normalization fairly captures steady-state efficiency, it partially masks early transients. For more complex tasks, the amortization point scales with the number of context-stimulus associations, but remains identifiable from the EDP-vs.-L curve.

C. Trade-off: Learning Rate vs. Efficiency

Higher adaptability imposes a direct efficiency penalty. Fig. 5 illustrates a linear increase in EDP per input as the context change rate (learning percentage) increases from 20% to 90% while accuracy decreases correspondingly from 80% to 10%. Since activation rates (energy rate or power dissipation) remain relatively constant, this degradation is driven almost entirely by the increased number of computational steps required for frequent synaptic weight updates. This confirms a fundamental trade-off: systems configured for high volatility environments (high learning rates) operate at a lower energy efficiency than those in stable contexts [18].

D. Impact of Numerical Resolution

Numerical precision dictates the baseline execution latency. Fig. 6 highlights a substantial and constant disparity between 32-bit and 5-bit implementations. The 32-bit configuration (top curve) maintains a significantly higher EDP baseline compared to the 5-bit configuration (bottom curve) across all sequence lengths. Since the number of computational steps is independent of variable bit-width in this model, the efficiency gap is attributable to the increased switching capacitance of high-precision arithmetic, which raises the energy dissipation per operation [1]. The $\sim 41\times$ EDP ratio is constant across L, validating the use of low-precision representations for edge neuromorphic hardware [47]. The 5-bit quantisation used here applies uniform fixed-point rounding; no differences in RL convergence speed were observed at the tested sequence lengths, as the winner-take-all mechanism provides inherent robustness to small weight perturbations.

E. Structural Sparity via Pruning

Pruning offers the most consistent efficiency improvements. Fig. 7 demonstrates that enabling pruning yields significant gains especially as the input size increases.

- Energy proxy: Receiver activations decrease by $\sim 63\%$ (Fig. 7, middle), as removing inactive synaptic paths directly eliminates the corresponding switching events.
- Runtime proxy: Computational steps are virtually unchanged (Fig. 7, bottom), since the sequential layer-activation schedule is determined by network depth rather than synapse count. Pruning therefore does not reduce inference latency in this sequential emulator.
- Net efficiency: Consequently, EDP / input (Fig. 7, top) drops from $10,600$ (Pruning OFF) to $3,900$ (Pruning ON) at L=10000, a $\sim 63\%$ reduction driven entirely by the energy term.

Crucially, pruning does not degrade accuracy, indicating that structural sparsity successfully removes redundant computations without degrading functional adaptation [15], [40].

As noted in Section IV.C, R-STDP weight-update operations are excluded from both proxies on the basis that dedicated neuromorphic hardware executes learning in parallel with feedforward inference [42], [43]. This is consistent with the emulator's design intent, but the energy cost of on-chip STDP circuits is non-trivial [21] and represents a direction for refinement in subsequent hardware-level studies.

VI. Conclusion

This work adapts the Energy–Delay Product (EDP) to address the critical energy constraints of neuromorphic computing. By integrating this metric into a hardware emulation framework for RL SNNs, robust proxies are established for execution time and spike-based energy. These metrics accurately capture critical architectural dependencies, including computational precision and synaptic pruning. The derivation of normalized EDP per sequence provides a seamless quantitative tool for early-stage design optimization. The utility of this framework is validated through analytical scenarios using the context-dependent Traveling Mouse problem, demonstrating its effectiveness in balancing bio-inspired adaptability with hardware efficiency. Future work includes benchmarking on more complex problems.

Acknowledgement

The authors would like to thank Mark von Rossum (University of Nottingham) and Federico Corradi (Eindhoven University of Technology) for their valuable input during the early stages of this work.

REFERENCES

[1] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14, 2014.

[2] A. Andrae, "New perspectives on internet electricity use in 2030," *Engineering and Applied Science Letters*, vol. 3, pp. 19–31, 2020.

[3] J. Backus, "Can programming be liberated from the von neumann style? a functional style and its algebra of programs," *Commun. ACM*, vol. 21, no. 8, p. 613–641, 1978.

[4] N. P. J. et al., "In-datacenter performance analysis of a tensor processing unit," *SIGARCH Comput. Archit. News*, p. 1–12, June 2017.

[5] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.

[6] G. Li, L. Deng, H. Tang, G. Pan, Y. Tian, K. Roy, and W. Maass, "Brain-inspired computing: A systematic survey and future trends," *Proceedings of the IEEE*, vol. 112, no. 6, pp. 544–584, 2024.

[7] A. Madhavan, "Brain-inspired computing can help us create faster, more energy-efficient devices — if we win the race," 2023.

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

[9] H. Rasheed, P. Mirtaheri, and A. Muhtaroglu, "A reduced spiking neural network architecture for energy efficient context-dependent reinforcement learning tasks," in *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2024.

[10] A. Rubino and et al., "Ultra-low pwr. si nrn. crct. for extreme-edge neuromorphic intelligence," in *Proc. of 26th Int. Conf. on Electronics, Crts. and Sys. (ICECS)*, pp. 458–461, 2019.

[11] J. I. Okonkwo, M. S. Abdelfattah, P. Mirtaheri, and A. Muhtaroglu, "Energy-aware bio-inspired spiking reinforcement learning system architecture for real-time autonomous edge applications," *Frontiers in Neuroscience*, vol. 18, 2024.

[12] D. Attwell and S. B. Laughlin, "An energy budget for signaling in the grey matter of the brain," *J Cereb Blood Flow Metab*, vol. 21, no. 10, pp. 1133–45, 2001.

[13] P. Lennie, "The cost of cortical computation," *Current Biology*, vol. 13, no. 6, pp. 493–497, 2003.

[14] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, "A survey of neuromorphic computing and neural networks in hardware," 2017.

[15] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, and E. Eleftheriou, "Low-power neuromorphic hardware for signal processing applications: A review of architectural and system-level design approaches," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 97–110, 2019.

[16] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "How to evaluate deep neural network processors: Tops/w (alone) considered harmful," *IEEE Solid-State Circuits Magazine*, vol. 12, no. 3, pp. 28–41, 2020.

[17] W. Lo and Y. C. Wong, "Spiking neural network for energy efficient learning and recognition," *International Journal of Scientific & Technology Research*, 2021.

[18] S. Schug, F. Benzing, and A. Steger, "Presynaptic stochasticity improves energy efficiency and helps alleviate the stability-plasticity dilemma," *Elife*, vol. 10, 2021.

[19] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, and T. Prodromakis, "Integration of nanoscale memristor synapses in neuromorphic computing architectures," *Nanotechnology*, vol. 24, no. 38, p. 384010, 2013.

[20] G. Indiveri, "Dynap-sel: An ultra-low power mixed signal dynamic neuromorphic asynchronous processor with self learning abilities," *Neuro morphic Computing nd Engineering*, 2019.

[21] J. M. Cruz-Albrecht and et al., "Energy-efficient neuron, synapse and stdp integrated circuits," *IEEE Trans. on Biomedical Crts. and Sys.*, vol. 6, no. 3, pp. 246–256, 2012.

[22] D. O. Hebb, *The organization of behavior; a neuropsychological theory*. The organization of behavior; a neuropsychological theory., Oxford, England: Wiley, 1949.

[23] G. K. Chen and et al., "A 4096-nrn. 1m-synapse 3.8-pj/sop spiking nrl. netw. w/ on-chip stdp learning & sprs. wghts in 10-nm finfet cmos," *IEEE J. of Sld-St. Crts.*, vol. 54, no. 4, pp. 992–1002, 2019.

[24] Y. Li, Z. Wang, and R. Midya, "Memristive crossbars for ultra-efficient neuromorphic computing," *Nature Electronics*, vol. 4, pp. 579–586, 2021.

[25] C. Lo, "Ai in healthcare: Key highlights in 2021," *Pharmaceutical Technology*, 2021.

[26] L. A. Brenner and et al., *Handbook of Rehabilitation Psychology*, ch. 30: AI and Robotics in Rehabilitation. American Psychological Association, 3rd ed., 2019.

[27] K. Roy and et al., "Re-engineering computing for next generation autonomous intelligent systems: Devices, circuits, and algorithms," tech. rep., C-BRIC Seminar, 2016.

[28] X. Wu, Y. Zhao, Y. Song, Y. Jiang, Y. Bai, X. Li, Y. Zhou, X. Yang, and Q. Hao, "Dynamic threshold integrate and fire neuron model for low latency spiking neural networks," *Neurocomputing*, vol. 544, p. 126247, 2023.

[29] N. Qiao, H. Zheng, B. Wu, and A. van Schaik, "<3 mw gesture recognition with event-driven neuromorphic system," in *IEEE AICAS*, pp. 1–5, 2021.

[30] Hailo, "A practical guide to edge ai power efficiency," 2021.

[31] Y. Yan, A. Dixius, J. Partzsch, and C. Mayr, "Dynamic synaptic rewiring for energy-efficient robot navigation in spiking neural networks," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 16, no. 1, pp. 202–215, 2024.

[32] R. Gonzalez and M. Horowitz, "Energy dissipation in general purpose microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 9, pp. 1277–1284, 1996.

[33] T. Gollisch and M. Meister, "Rapid neural coding in the retina with relative spike latencies," *Science*, vol. 319, no. 5866, pp. 1108–1111, 2008.

[34] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, P. Joshi, A. Lines, A. Wild, H. Wang, and D. Mathaikutty, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. PP, pp. 1–1, 2018.

[35] M. Liang, J. Zhang, and H. Chen, "A 1.13μj/classification spiking neural network accelerator with a single-spike neuron model and sparse weights," *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2021.

[36] F. Cai and et al., "A fully integrated reprogrammable memristor–cmos system for efficient multiply–accumulate operations," *Nature Electronics*, vol. 2, pp. 290–299, 2019.

[37] S. R. Nandakumar and et al., "Building brain-inspired computing systems," *IEEE Nanotechnology Magazine*, vol. 12, no. 4, pp. 19–35, 2018.

[38] E. M. Izhikevich, "Which model to use for cortical spiking neurons?," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[39] J. D. Yang, B. Dong, F. Heide, Y. Ding, Y. Zhou, B. Yin, and Xin, "Biologically inspired dynamic thresholds for spiking neural networks," in *Proc. of Neural Information Processing Systems (NeurIPS)*, NIPS '22, Curran Associates Inc., 2022.

[40] 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.

[41] S. Shivkumar, V. Muralidharan, and V. S. Chakravarthy, "A biologically plausible architecture of the striatum to solve context-dependent reinforcement learning tasks," *Frontiers in Neural Circuits*, vol. 11, 2017.

[42] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength and postsynaptic cell type," *J Neurosci*, vol. 18, no. 24, pp. 10464–72, 1998.

[43] W. Schultz, "Multiple dopamine functions at different time courses," *Annu Rev Neurosci*, vol. 30, pp. 259–88, 2007.

[44] W. Schultz, "Behavioral dopamine signals," *Trends Neurosci*, vol. 30, no. 5, pp. 203–10, 2007.

[45] F. Raudies and M. E. Hasselmo, "A model of hippocampal spiking responses to items during learning of a context-dependent task," *Frontiers in Systems Neuroscience*, vol. 8, 2014.

[46] M. E. Hasselmo, "The role of acetylcholine in learning and memory," *Current Opinion in Neurobiology*, vol. 16, no. 6, pp. 710–715, 2006.

[47] Z. Padamsey and N. L. Rochefort, "Paying the brain's energy bill," *Current Opinion in Neurobiology*, vol. 78, p. 102668, 2023.

[48] P. Zahid, K. Danai, D. Nathalie, and L. R. Nathalie, "Neocortex saves energy by reducing coding precision during food scarcity," *Neuron*, vol. 110, no. 2, pp. 280–296.e10, 2022.

[49] J. J. Harris, R. Jolivet, and D. Attwell, "Synaptic energy use and supply," *Neuron*, vol. 75, no. 5, pp. 762–777, 2012.