Heartbeat Classification with Spiking Neural Networks on the Loihi Neuromorphic Processor

Kyle Buettner, Alan D. George
Department of Electrical and Computer Engineering, University of Pittsburgh
NSF Center for Space, High-Performance, and Resilient Computing (SHREC)
Pittsburgh, PA, USA
{kyle.buettner, alan.george}@pitt.edu

***Abstract*—One potential method to efficiently deploy deep neural networks is through neuromorphic computing, a processing paradigm that emulates the energy-efficient spiking neural networks (SNNs) of the human brain. To evaluate the current capabilities of neuromorphic computing, this research investigates Intel's state-of-the-art Loihi processor through a heartbeat classification case study. In particular, artificial-to-spiking neural network conversion with *SNN-Toolbox* is leveraged to create a spiking 1D-convolutional neural network for Loihi. Latency and accuracy optimization strategies in this framework are explored, and the SNN performance on Loihi is compared to the performance of architecturally identical artificial neural networks (ANNs) on Intel Core i7 CPU, Intel Neural Compute Stick 2, and Google Coral Edge TPU devices. The SNN reaches an accuracy and macro-averaged F1 score of 97.8% and 87.9%, respectively, compared to 98.4% and 90.8% for the CPU-based ANN. Additionally, with the best dynamic power across devices, Loihi provides a 32 times lower energy-delay product versus the CPU baseline. Compared to the edge devices, Loihi is found to result in a larger energy-delay product due to a higher latency bottlenecked by x86 core-to-host I/O and x86 core-based management. Overall, this research highlights the benefits and limitations of a practical neuromorphic computing methodology.**

***Index Terms*—Heartbeat classification, neural network hardware, neuromorphic computing, performance analysis, spiking neural networks**

I. Introduction

A major limitation in deploying deep neural networks (DNNs) is energy consumption, as DNNs are challenging to efficiently implement on milliwatt and microwatt devices [1]. With data analytics moving to the edge to achieve benefits in bandwidth consumption, latency, and privacy, it is crucial to consider architectural methods to reduce the energy costs of neural networks. Solutions will lead to more real-time insights and enable more capabilities in the Internet of Things (IoT).

Neuromorphic computing is an attractive paradigm for energy-efficient processor design as it uses the human brain's high parallelism and low power as inspiration. The emergence of Intel's neuromorphic research chip Loihi [2], a platform for spiking neural networks (SNNs), provides the opportunity to explore applications that can benefit from neuromorphic computing. This research identifies heartbeat classification through electrocardiogram signals as a fitting case study, as it is hypothesized that an SNN can perform low-latency, low-energy identification of arrhythmias on Loihi. An energy-efficient design could be deployed in a wearable device to provide real-time insights and help in diagnosis and prevention of heart disease, a top health challenge in the USA [3].

To perform heartbeat classification on Loihi, this research investigates SNN design through artificial-to-spiking neural network conversion in the *SNN-Toolbox* framework [4]. A variety of strategies are outlined to elucidate practical design of an accurate and performant spiking 1D-convolutional neural network (1D-CNN). In comparing latency and energy performance of the SNN on Loihi versus architecturally identical artificial neural networks (ANNs) on Intel Core i7 CPU, Intel Neural Compute Stick 2, and Google Coral Edge TPU devices, this research also evaluates how such a neuromorphic approach compares to other state-of-the-art neural network devices.

This work was supported by SHREC industry and agency members and by the IUCRC Program of the National Science Foundation under Grant No. CNS-1738783.

II. Background

The background for this research is presented as two topics. First, electrocardiogram analysis is discussed to provide motivation for heartbeat classification. Second, the foundations of neuromorphic computing are described, along with information regarding the architecture of Loihi.

A. Electrocardiogram Analysis

Electrocardiograms (ECGs or EKGs) are tests that capture electrical activity fluctuations in the heart [5]. Electrical activity changes when heart muscles polarize in a rhythmic manner, creating heartbeats. Heartbeats are represented in ECG signals by morphological features such as the QRS complex, P wave, and T wave. To record ECG signals, noninvasive electrodes (or leads) are attached to the skin, often in 5-lead or 12-lead configurations. In a medical setting, signals can be analyzed by clinicians to characterize heart function and diagnose conditions related to cardiac health, such as arrhythmias. It is also possible to perform portable, continuous recording or monitoring of ECG signals for several hours or days using a wearable device, such as a Holter monitor [6]. With portable devices, signals can be saved for later analysis, or anomalous heartbeats can be detected automatically in real time, providing immediate insights that can be used to trigger additional recording or alerts. These insights can ultimately result in improved opportunities for care.

B. Neuromorphic Computing

Neuromorphic computing is a type of processing that closely models the human brain. Carver Mead helped formalize neuromorphic processors in the 1980s with a book on analog VLSI-based neural circuits [7]. The field has developed to encapsulate digital and analog modeling of SNNs. A neuron in an SNN has a membrane potential that is augmented over time through the presentation of binary spikes on synapses. Once a voltage threshold is reached, that neuron fires a spike and resets its state. Different neuron models can represent spiking behavior, with one example being the leaky integrate-and-fire (LIF) model, which treats neuron membranes as leaky integrator RC circuits [8]. Moreover, information can be captured in the rates or timing of spikes. These codings are called "rate coding" and "temporal coding", respectively.

SNN energy is primarily consumed through spiking, which is asynchronous and sparse. This property has inspired designers to strive to create energy-efficient neuromorphic hardware. Such a processor provides a mode of operation in contrast to the energy-consuming synchronous clocking of CPUs and GPUs. Neuromorphic hardware also ideally avoids the processor-memory bottleneck of von Neumann architectures, where throughput is limited by processor-memory bandwidth. This benefit is a result of spiking neurons maintaining local state, thus co-locating computation and communication.

A recent neuromorphic research chip is Loihi, a digital integrated circuit fabricated on a 14nm process [2]. Loihi represents a key step forward in neuromorphic design, especially with its programmable on-chip learning engine, a unique capability versus past designs such as IBM TrueNorth [9]. Each Loihi chip contains a mesh of 128 neuron cores with 1,024 neurons per core and up to 130 million synapses for spike communication. Additionally, each Loihi chip contains three embedded x86 Lakemont cores, where spiking neural interfacing processes (SNIPs) are run to manage neuron cores. Neuron dynamics are based on current-based synapse (CUBA) LIF modeling with discretely approximated variables for membrane potential and synaptic response current.

III. Related Research

This research details three areas of related work. First, a survey of ECG-based heartbeat classification is provided, with emphasis on SNN-based approaches. Second, methods and frameworks to create SNNs, including *SNN-Toolbox*, are described. Lastly, this section concludes with past performance comparisons of SNNs on Loihi versus ANNs on other devices.

A. Electrocardiogram-Based Heartbeat Classification

Multiple stages can exist in an ECG analysis pipeline: signal preprocessing, heartbeat segmentation, feature extraction, and classification [10]. One of the most common datasets used in analysis is the MIT-BIH Arrhythmia Database, which contains 30 minute ECG recordings for 47 patients with various arrhythmia types [11]. The Association for the Advancement of Medical Instrumentation (AAMI) provides standards for model evaluation on such datasets [12].

Neuromorphic computing has inspired the exploration of SNNs for ECG analysis. For example, Corradi et al. implement a spiking recurrent network of LIF neurons on the custom VLSI-based Dynamic Neuromorphic Asynchronous Processor (DYNAP) system, attaining 95% classification accuracy over 18 classes in the MIT-BIH dataset [13]. In another approach on DYNAP, Bauer et al. use reservoir computing to perform ECG-based anomaly detection, with the approach estimated to be capable of sub-mW power [14]. Lastly, Amirshahi and Hashemi use a reward-modulated spike-timing dependent plasticity (STDP) learning rule to train a spiking ECG classifier [15]. They provide benchmarks of other neural network heartbeat classifiers on an ARM Cortex-A53 CPU and estimate SNN energy to be two to nine orders of magnitude smaller.

B. Spiking Neural Network Creation

Deep SNNs have been challenging to accurately train as they are unable to directly use backpropagation [16]. This inability stems from the non-differentiable nature of spike functions. Recent work has aimed to overcome this challenge, such as SLAYER [17], which uses temporal credit assignment to backpropagate error for SNN training, and Nengo [18], which provides functionality for converting ANNs to SNNs. The focus of this work is on the ANN-to-SNN conversion framework, *SNN-Toolbox* [4]. This framework operates by converting each artificial neuron to a spiking integrate-and-fire neuron and computing SNN parameters that best correlate ANN activation values to average SNN spike rates. A key strategy employed to achieve high accuracy is data-based weight normalization, which avoids too low or high firing rates by scaling weights and biases with the maximum activation values calculated over a subset of data. Moreover, *SNN-Toolbox* allows users to transfer models from deep learning frameworks like TensorFlow (TF) and supports TF layer types such as softmax, batch normalization, and pooling. Models converted in this framework can be mapped to various neuromorphic backends including Loihi. For more information on the fundamentals of *SNN-Toolbox*, please refer to [4].

C. Spiking Neural Network Performance on Loihi

Prior Loihi benchmarking has focused on dynamic energy cost per inference, the product of dynamic power and latency. For instance, Blouw et al. create a two-layer perceptron for keyword spotting with Nengo on Loihi's Wolf Mountain board [19]. Loihi was estimated to provide 5.3$\times$ and 20.5$\times$ better dynamic energy cost per unbatched inference versus the Intel Neural Compute Stick 1 and NVIDIA Jetson TX1, respectively. Other work has evaluated Loihi with energy-delay product (EDP), the product of energy and latency, to account for low-energy, but slow models. For example, Ceolini et al. use SLAYER to create a SNN with three convolutional layers, two pooling layers, and two fully-connected layers for gesture recognition [20]. The SNN on Loihi achieved better than non-spiking accuracy with fused dynamic vision sensor and electromyography data. Loihi also provided an EDP estimated to be 26$\times$ better than that of a NVIDIA Jetson Nano GPU.

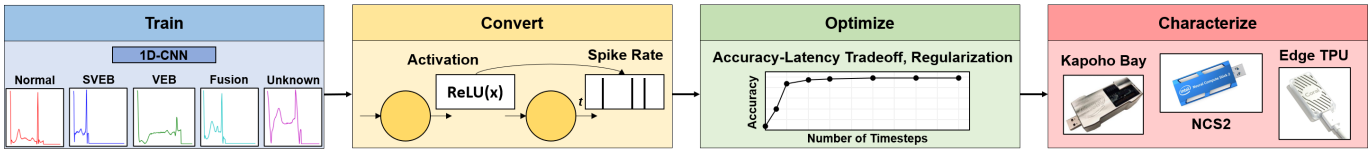

Fig. 1. Heartbeat Classification Methodology. First, an ANN is trained to classify normal heartbeats versus common arrhythmia types. Next, *SNN-Toolbox* creates an SNN with parameters that correlate ANN activation values to spike rates. Then, strategies are explored to improve SNN accuracy and latency. Lastly, the SNN on Loihi is compared to ANNs on devices such as the Edge TPU and NCS2 in terms of latency and energy.

IV. Methodology

The development of neuromorphic platforms like Loihi provides opportunities for competitive benchmarking and exploration of energy-efficient SNN designs. This research explores ANN-to-SNN conversion through *SNN-Toolbox* to perform heartbeat classification on Loihi. While research exists outlining design considerations of *SNN-Toolbox* for MNIST, CIFAR-10, and DvsGesture datasets [21], the performance of *SNN-Toolbox* models on Loihi versus models on edge neural network hardware has yet to be comprehensively explored. The methodology of this work (shown in Fig. 1) aims to bridge *SNN-Toolbox* design considerations to practical deployment and to compare approaches across neural network hardware.

A. Model Design, Tuning, and Optimization

This research assumes efficient and accurate heartbeat segmentation from ECG signals and focuses on classification. A preprocessed version of the MIT-BIH dataset is selected for classification [22]. This dataset has 109,446 segmented heartbeats, each 187 samples in length, captured from lead II of an ECG with sampling frequency 125 Hz. The dataset has five classes of heart rhythms: N (normal), SVEB (supraventricular ectopic beat), VEB (ventricular ectopic beat), F (fusion beat), and Q (unknown). These classes have imbalanced proportions: N (82.77%), SVEB (2.54%), VEB (6.61%), F (0.73%), and Q (7.35%). The authors of [22] provide training and test sets in an equivalent 80/20 split across classes. The training set is further partitioned into a 90/10 split for a validation set.

A 1D-CNN architecture is selected for study of *SNN-Toolbox*. The model, shown in Table I, is designed to achieve high accuracy, be large enough for insightful benchmarks, but not have too many layers to significantly impact *SNN-Toolbox* conversion, as rate approximation error can accumulate with increasing network depth [4]. This model is trained for 50 epochs with TensorFlow 2.2.0, using categorical crossentropy loss, a batch size of 32, and the Adam optimizer with learning rate 0.001. Evaluation is performed with AAMI recommended metrics of precision (positive predictivity), recall (sensitivity), false positive rate (FPR), and accuracy, along with a concise macro-averaged F1 score metric for model comparison.

After training, the model is converted to an architecturally identical SNN with *SNN-Toolbox*. The ANN and SNN have the same layers and number of neurons, but the SNN operates with a spiking neuron model, which requires key hyperparameters to be tuned. For instance, an inference is performed by presenting each sample as bias current to the SNN for n timesteps. The parameter n thus determines the number of computational operations per sample, which in turn affects test accuracy and latency. The optimal value of n is captured through exploration of this accuracy-latency tradeoff. Another parameter is the neuron reset mechanism, which determines how neurons reset voltage after reaching a threshold. A soft reset mechanism is selected, which means that at the time a neuron's voltage moves past the threshold, the membrane potential is reset to the current membrane potential minus the voltage threshold. Lastly, a set of data samples is needed for conversion. A subset corresponding to 10% of the training set with equivalent class distribution is used for this purpose.

TABLE I
Baseline 1D-CNN Architecture

Layer Type	Activation Function	Output Shape	Filter-Kernel-Stride Configuration	Parameter Count
Input	-	(187, 1)	-	-
Conv1D	ReLU	(92, 8)	(8, 5, 2)	48
Conv1D	ReLU	(44, 16)	(16, 5, 2)	656
Conv1D	ReLU	(20, 32)	(32, 5, 2)	2,592
Flatten	-	(640)	-	-
FC	ReLU	(32)	-	20,512
FC	Softmax	(5)	-	165

Conversion is not guaranteed to generate high accuracy, as poor conversion may result from variance or outliers in the weight, bias, or activity distributions of the ANN [4]. This research considers further strategies to improve SNN accuracy to overcome these challenges. For one, bias values are constrained to zero during training to prevent large values from impacting conversion, as performed in [23]. Second, regularization in the form of dropout is added between the final convolutional layer and first fully-connected layer to penalize large weights and biases.

The SNN is mapped to Loihi's neuron cores through the Intel NxSDK software, which provides a NxTF interface to implement spiking TF layers. In running inference on Loihi, each sample is sent from a host CPU to Loihi's x86 cores to Loihi's neuron cores. Each sample is run for n timesteps. In each timestep, there is spiking time for neurons to send spikes and update state. There is also management time for SNIPs to run on the x86 cores to interact with neuron cores. The SNIPs needed for *SNN-Toolbox* include: input injection from the x86 cores to the neuron cores, classification readout from the neuron cores to the x86 cores, and reset of neuron core state between samples. After each sample, results are read out from the x86 cores to the host CPU.

B. Performance Evaluation

The spiking 1D-CNN on Loihi is compared to architecturally identical models on CPU and low-power, AI-targeted edge hardware. An Intel Core i7-6700 CPU @ 3.40GHz with 16 GB RAM is chosen as a CPU baseline and the host for the accelerator devices. The Kapoho Bay platform, with two Loihi chips, is selected as an edge, USB form factor of Loihi. The Intel Neural Compute Stick 2 (NCS2) and Google Coral Edge TPU (Edge TPU) are chosen as devices that represent the state-of-the-art for neural network inferencing at the edge [24]. All devices interface from a TensorFlow 2.2.0 script in Python 3.5, but require different frameworks for mapping: Intel NxSDK 0.9.9 and *SNN-Toolbox* 0.5.0 for Loihi, OpenVINO 2021.1.110 for the NCS2, and TFLite from TensorFlow 2.2.0 for the Edge TPU. Model quantization also differs with INT8 for Loihi and the Edge TPU and a minimum quantization of FP16 for the NCS2. The quantized accuracies for the NCS2 and Edge TPU are equivalent to original accuracy. The metrics for device evaluation include latency, power (idle, running, and dynamic), dynamic energy cost per inference, and EDP. Dynamic energy cost per inference is calculated by multiplying dynamic power by latency and EDP by multiplying dynamic energy cost per inference by latency. Devices are compared primarily through EDP to represent tradeoffs that can be made in CMOS circuitry to improve energy or delay. These metrics are calculated for unbatched inferencing to match this online use case.

Latency is measured as the average inference time computed over a loop of 1,000 test heartbeats, averaged over 10 trials. The power benchmarking methodology is inspired by [19]. Across all devices, idle power is measured from boot. For all devices except Loihi, average running power is calculated over a two-minute inference loop. Dynamic power is then calculated by subtracting idle power from running power. The CPU uses the software command line utility s-tui, collecting samples every 200 ms, to get package power readings, while the NCS2 and Edge TPU use inline voltage-current USB and USB-C meters, respectively, with readings recorded every 10 seconds. The USB meter has two-decimal precision, and the USB-C meter has three-decimal precision. Due to the nature of measurement, USB I/O is included in power estimates.

NxSDK probes are used to estimate idle, dynamic, and total power of a Loihi chip on the Kapoho Bay platform. These probes also provide power breakdown in terms of neuron core and x86 core usage. The frequency of power measurements on Kapoho Bay makes it not possible to accurately capture neuron core readings at low timestep counts, so samples are extended to run for 8,192 timesteps to collect an adequate number of readings. This collection is performed for five samples, which is an adequate amount of data to gather readings without overwhelming embedded memory. To calculate dynamic energy, x86 and neuron core power are treated as always active during an inference. This choice may be pessimistic, as neuron cores can be idle during part of the inference. However, given the indirect power estimation methodology, assuming both components as always on provides the safest power estimates.

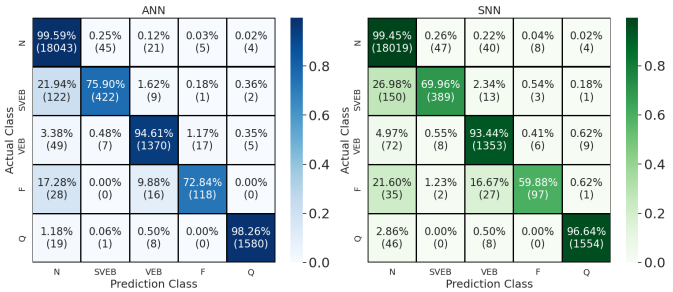

Fig. 2. Confusion Matrix Heatmaps Normalized By Class Size for ANN (blue) and SNN (green).

V. Results

The results of this research are presented as two main categories. First, the optimized SNN is evaluated versus the baseline ANN architecture, with insights into the strategies used to achieve maximum SNN accuracy. Second, the SNN on Loihi is compared to the ANN on CPU, Intel NCS2, and TPU hardware in terms of runtime and energy.

A. Model Evaluation

Evaluation of the ANN versus the final, optimized SNN at 64 timesteps is presented in Fig. 2, which shows confusion matrices, Table II, which shows per-class metrics of recall, precision, and false positive rate, and Table III, which shows total accuracy and macro-averaged F1 score. As with past literature [25], the prediction of a heartbeat as class VEB when it is class F or class Q does not count as a false positive for class VEB. Similarly, a prediction of class SVEB from class Q is not considered a false positive for class SVEB. In general, the ANN achieves better recall, precision, and false positive rate across classes, but the SNN is close in metrics especially for classes N, VEB, and Q. The SNN recall values for classes SVEB and F trail the ANN recall values more significantly. Of note, these classes have the lowest proportions of training samples. Overall, the SNN achieves 0.6% lower accuracy and 2.9% lower macro-averaged F1 score than the ANN.

TABLE II
Per-Class Metric Comparison

Class	ANN			SNN		
	Recall **(%)**	***Precision*** **(%)**	***FPR*** **(%)**	***Recall*** **(%)**	***Precision*** **(%)**	***FPR*** **(%)**
N	99.6	98.8	5.78	99.5	98.3	8.03
SVEB	75.9	89.0	0.244	70.0	87.2	0.267
VEB	94.6	97.9	0.147	93.4	96.2	0.260
F	72.8	83.7	0.106	59.9	85.1	0.0782
Q	98.3	99.3	0.0542	96.6	99.0	0.0739

TABLE III
Overall Metric Comparison

Metric	ANN	SNN
Accuracy (%)	98.4	97.8
Macro-Averaged F1 Score (%)	90.8	87.9

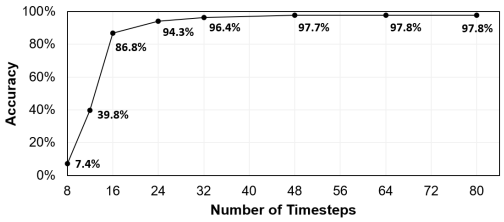

Fig. 3. Accuracy-Latency Tradeoff. Test set accuracy for the SNN is dependent on the number of timesteps that each sample is presented. Maximum accuracy is achieved when each sample runs for at least 64 timesteps.

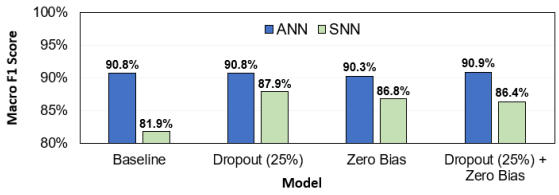

Fig. 4. Exploration of Strategies to Improve SNN Accuracy. Presented are ANN and SNN macro-averaged F1 scores for the baseline model, a model with dropout of 25% before the fully-connected layer, a model with bias values restricted to zero, and a model with both dropout and zero biases. Top performance was achieved with only dropout.

To get maximum SNN accuracy on the test set, a variety of design parameters are considered. For one, the accuracy-latency tradeoff of this SNN, shown in Fig. 3, is explored to get an optimum number of timesteps n to run each sample. It is demonstrated that the SNN achieves its top accuracy at 64 timesteps. It is also found that this timestep count produces the best macro-averaged F1 score. The SNN is optimized to run at 64 timesteps to achieve the best latency for this accuracy. Moreover, solely converting the network in Table I does not lead to the SNN with the best performance, reaching a macro-averaged F1 score of 81.9%. To improve accuracy, this work explores bias restriction, by creating a model with all neuron bias values set as zero, and regularization, by adding a dropout layer with 25% drop rate between the last convolutional layer and first fully-connected layer in the model. The results of these potential optimizations alone and together are shown in Fig. 4. Restricting bias values and adding regularization help in isolation and together. The model with just dropout achieves the best performance and is the final, optimized SNN.

B. Model Energy and Runtime Performance

SNN performance is evaluated on the Loihi Kapoho Bay platform. In terms of resource usage, the SNN occupies 9 neuron cores, and a total of 72 neuron cores are active due to Loihi's barrier synchronization constraints. Latency-wise, the model takes on average 7.222 ms for each inference. Through NxSDK profiling, it is found that on average, 0.798 ms of this time is spiking computation on the neuron cores, 3.464 ms is management time to run SNIPs on x86 cores and manage neuron cores, and 2.960 ms is to read classifications from the x86 cores to the host CPU. These results show that Loihi runtime is bottlenecked by x86 core-based management and x86 core-to-host I/O rather than by spiking computation.

Loihi is further compared to CPU, NCS2, and Edge TPU devices, with benchmarks shown in Table IV. The SNN is the slowest at 7.222 ms, while the Edge TPU is the fastest at 0.204 ms. Loihi has the best dynamic power, using 58 mW. The neuron cores contribute 38 mW of this power, while the x86 cores use 20 mW. The NCS2 and Edge TPU have similar dynamic powers, of 0.61 W and 0.656 W, respectively, while the CPU dynamic power is the highest at 15.7 W. In EDP, the SNN on Loihi is found to provide a 32× improvement over the CPU approach. Also, although having a 2.6× improvement in dynamic energy cost per inference versus the NCS2, Loihi results in an EDP that is 1.6× worse than the NCS2 because of higher latency. The Edge TPU results in the best EDP of 0.0273 μJs due to best-in-class latency.

VI. Discussion

ANN-to-SNN conversion proves generally successful for the 1D-CNN tested. Tuning the number of timesteps and regularization enable the SNN to achieve close to non-spiking accuracy and F1 score on Loihi at the minimum latency needed. Achieving comparable accuracy to an ANN with such tuning strategies is important for practical neuromorphic computing. Still, SNN accuracy has room for improvement. In particular, the recall for classes SVEB and F are noticeably lower than the ANN. This problem could perhaps be remedied with a more balanced dataset and therefore a better subset of data used for conversion. Nevertheless, while ANN-to-SNN conversion works decently for the network tested, additional investigation should be conducted in how to convert networks of greater scale and diversity. Extending the capabilities of such a methodology will enable neuromorphic computing to be a competitive paradigm for low-power AI.

TABLE IV
Device Performance Comparison

Device		Power (W)			Inference Latency (ms)	Dynamic Energy Cost Per Inference (mJ)	Energy-Delay Product (μJs)
		Idle	*Running*	*Dynamic*			
Intel Core i7-6700 CPU @ 3.40GHz		11.8	27.5	15.7	2.455	38.5	94.6
Intel Loihi Kapoho Bay	x86 Cores	-	-	0.020	7.222	0.14	-
	Neuron Cores	-	-	0.038	7.222	0.27	-
	Total	0.262	0.320	0.058	7.222	0.42	3.0
Intel Neural Compute Stick 2		0.56	1.17	0.61	1.787	1.1	1.9
Google Coral Edge TPU		0.358	1.014	0.656	0.204	0.134	0.0273

In reference to benchmarking, Loihi results in a significantly lower EDP versus the CPU. Loihi is able to achieve this EDP reduction with heartbeat classification being an unbatched use case. A batched use case is likely to see better CPU performance. Moreover, the lower dynamic power of Loihi is found to not outweigh the lower latencies of the NCS2 and the TPU, leading to a higher EDP for Loihi. Further analysis shows Loihi to have a latency bottleneck due to x86 core-to-host I/O and x86 core-based management. It is fair to note that Loihi is a research chip, and the choice of x86 cores is not fundamental to its design. Improving such architectural constraints can help future neuromorphic chips improve performance. Nonetheless, as it was not possible to exclude USB I/O power from NCS2 and Edge TPU estimates, the gap in actual EDP between those devices and Loihi may be greater. In the context of energy-efficient neural network inferencing, this use case benefits from a more mature and inference-targeted device like a TPU, rather than from Loihi and this ANN-to-SNN conversion methodology.

A future route to improve upon this research is to explore more energy-efficient SNN designs. This research involves rate-coding, but temporally coded neurons could potentially provide higher energy efficiency due to sparser spike communication. Another option is to encode ECG signals as spikes, rather than as bias current, and to feed those spikes into an SNN as samples are collected. An SNN could capture temporal information across multiple heartbeats through these spikes, removing the need to perform a model state reset between samples. Lastly, as Loihi supports on-chip learning, one can explore ECG analysis in the form of a personal heartbeat learning system to enable increased health insights.

VII. Conclusion

In this study, Loihi was investigated as a neuromorphic platform for heartbeat classification. An SNN was created with *SNN-Toolbox*, reaching close to non-spiking accuracy and macro-averaged F1 score and achieving a lower energy-delay product on Loihi than that of the CPU baseline. Due to higher latency, the energy-delay product of Loihi was found to be higher than the Intel Neural Compute Stick 2 and Google Coral Edge TPU. Loihi's latency was bottlenecked by x86 core-to-host I/O and x86 core-based management instead of spiking computation. Addressing these constraints is a future direction to enhance neuromorphic architectures. The improvement of architectures, along with continued study of efficient SNN design, will increase the potential of neuromorphic computing in low-energy use cases like heartbeat classification.

References

[1] M. Verhelst and B. Moons, "Embedded deep neural network processing: Algorithmic and processor techniques bring deep learning to IoT and edge devices," *IEEE Solid-State Circuits Magazine*, vol. 9, no. 4, pp. 55–65, 2017.

[2] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[3] H. K. Weir *et al.*, "Peer reviewed: heart disease and cancer deaths—trends and projections in the United States, 1969–2020," *Preventing Chronic Disease*, vol. 13, 2016.

[4] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in Neuroscience*, vol. 11, p. 682, 2017.

[5] A. B. De Luna, *Clinical Electrocardiography, Enhanced Edition: A Textbook*. John Wiley & Sons, 2012.

[6] P. Zimetbaum and A. Goldman, "Ambulatory arrhythmia monitoring: choosing the right device," *Circulation*, vol. 122, no. 16, pp. 1629–1636, 2010.

[7] C. Mead, *Analog VLSI and Neural Systems*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[8] L. F. Abbott, "Lapicque's introduction of the integrate-and-fire model neuron (1907)," *Brain Research Bulletin*, vol. 50, no. 5-6, pp. 303–304, 1999.

[9] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[10] E. J. S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti, "ECG-based heartbeat classification for arrhythmia detection: A survey," *Computer Methods and Programs in Biomedicine*, vol. 127, pp. 144–164, 2016.

[11] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[12] AAMI, *Recommended practice for testing and reporting performance results of ventricular arrhythmia detection algorithms*. Arlington, VA, USA: Association for the Advancement of Medical Instrumentation, 1987.

[13] F. Corradi *et al.*, "ECG-based heartbeat classification in neuromorphic hardware," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[14] F. C. Bauer, D. R. Muir, and G. Indiveri, "Real-time ultra-low power ECG anomaly detection using an event-driven neuromorphic processor," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1575–1582, 2019.

[15] A. Amirshahi and M. Hashemi, "ECG classification algorithm based on STDP and R-STDP neural networks for real-time monitoring on ultra low-power personal wearable devices," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1483–1493, 2019.

[16] 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.

[17] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," *Advances in Neural Information Processing Systems*, vol. 31, pp. 1412–1421, 2018.

[18] T. Bekolay *et al.*, "Nengo: a Python tool for building large-scale functional brain models," *Frontiers in Neuroinformatics*, vol. 7, p. 48, 2014.

[19] P. Blouw, X. Choo, E. Hunsberger, and C. Eliasmith, "Benchmarking keyword spotting efficiency on neuromorphic hardware," in *Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop*, pp. 1–8, 2019.

[20] E. Ceolini *et al.*, "Hand-gesture recognition based on EMG and event-based camera sensor fusion: A benchmark in neuromorphic computing," *Frontiers in Neuroscience*, vol. 14, 2020.

[21] R. Massa, A. Marchisio, M. Martina, and M. Shafique, "An efficient spiking neural network for recognizing gestures with a DVS camera on the Loihi neuromorphic processor," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, 2020.

[22] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, "ECG heartbeat classification: A deep transferable representation," in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 443–444, IEEE, 2018.

[23] 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," in *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2015.

[24] L. Kljucaric, A. Johnson, and A. D. George, "Architectural analysis of deep learning on edge accelerators," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–7, IEEE, 2020.

[25] P. De Chazal, M. O'Dwyer, and R. B. Reilly, "Automatic classification of heartbeats using ECG morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 7, pp. 1196–1206, 2004.