To the Graduate Council:
I am submitting herewith a thesis written by Julia Anne Steed entitled "Application of Neuromorphic Computing to Real World Scenarios." I have examined the final paper copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Dr. Catherine Schuman, Major Professor

We have read this thesis
and recommend its acceptance:

James S. Plank

Garrett S. Rose

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

To the Graduate Council:
I am submitting herewith a thesis written by Julia Anne Steed entitled "Application of Neuromorphic Computing to Real World Scenarios." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Dr. Catherine Schuman, Major Professor

We have read this thesis
and recommend its acceptance:

James S. Plank

Garrett S. Rose

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Application of Neuromorphic Computing to Real World Scenarios

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Julia Anne Steed

May 2025

© by Julia Anne Steed, 2025
All Rights Reserved.

I dedicate this work to my parents, who have supported me unceasingly throughout this journey and many others. I wouldn't be here without your love and the opportunities you've given me.

To my brother, one of my best friends, who has always been a source of comic relief and joy.

To my friends and church family, who have encouraged me along the way and with whom I've created memories I'll cherish forever.

To the Lord, who has given me all that I have, including these wonderful people, and who is behind every success.

Acknowledgements

First, I would like to thank my research advisor, Dr. Catherine Schuman, for her support and encouragement throughout my academic career, all the way back to when I attended Oak Ridge High School. She has given me countless opportunities to advance my knowledge and career and has inspired me to work hard. She has been there for me during personal and academic challenges both, and has generously shared valuable wisdom from her own experiences that has helped me to succeed. She is one of the kindest and most humble people I know.

I would like to Dr. James S. Plank for providing me with helpful advice, tools, and resources to use throughout my research. His knowledge and involvement in my work has helped me to succeed. I would like to thank Dr. Garrett S. Rose for his support and feedback throughout this process. I would also like to thank the TENNLab group as a whole for their enthusiasm and dedication to our field. Past and present members have developed valuable resources that have been instrumental to my success. I want to thank Charlie Rizzo and Bryson Gullett in particular for always being willing to answer questions, explain concepts, and help me to work through issues. This warm community has been a blessing and I have truly enjoyed getting to know each and every one of you.

I would like to thank my parents, Chad and Jessica Steed, for their love, support, and encouragement throughout my life. My dad has taught me what faithful persistence through trials looks like and has shown me how I can use my work to glorify God. My mom's strength and determination has taught me that I can do

anything I set my mind to, and she has shown me how to serve others in selfless love. I would like to thank my brother, Blake, who has encouraged me to give every endeavor my all through his involvement with the ROTC. Without their collective support, I would not be where I am today. I would like to thank my friends, in particular my community at City Church. I am so blessed to be a part of a church that feels like family and continually shows up for one another.

Abstract

Spiking neural networks (SNNs) aim to mimic the human brain in the way that they process data. Neurons connected via synapses fire, or spike, sending information to other neurons and producing an output. SNNs have been applied to a variety of problem spaces for tasks such as classification and control. They are able to process real-time data efficiently and are particularly useful for application to temporal data. This work investigates the use of hafnia-based ferroelectric devices as synapses for a pole-balancing control task in varying levels of difficulty. We found that the ferroelectric networks are able to achieve comparable performance to basic neuromorphic implementations, indicating that these devices are well-suited to neuromorphic computation and performance limitations can be overcome with additional optimization. The best performing read voltage was task-dependent but performance could be improved with additional training. Additionally, the inclusion of noise in the ferroelectric device data, which more accurately reflects the behavior of these devices in the real world, did not significantly impair performance and in some cases even produced better results. We also found that small networks with relatively low activity still achieved good performance. We used SNNs to process experimental minute ventilation data for mice of varying sex, buprenorphine dosage, obesity, and leptin statuses. We attempted to classify buprenorphine dosage, congenic line, and sex using the minute ventilation readings. We found that neuromorphic approaches were more accurate in general than standard machine learning approaches. Our

results show that including the temporal component of the data significantly improved performance due to the temporal nature of SNNs.

Table of Contents

1 Introduction **1**

2 Literature Review **4**

2.1 Evolutionary Approaches 4

2.2 Neuromorphic Co-Design 5

3 Previous Work **7**

3.1 TENNLab Framework 7

3.2 Training Algorithm: EONS 7

3.3 Neuromorphic Model: RISP 8

4 Evaluation of Ferroelectric Devices for Neuromorphic Applications **9**

4.1 Overview 9

4.2 Ferroelectric Device Modeling 10

4.3 Neuromorphic Simulator 12

4.4 Applications 14

4.5 Experimental Setup 14

4.6 Results 16

4.6.1 Performance Comparison with Baseline Implementations . . . 16

4.6.2 Impact of Read Voltage 18

4.6.3 Impact of Noise 18

4.6.4 Network Size and Structure 19

4.6.5 Network Activity 23
4.7 Discussion and Conclusion 24

5 Neuromorphic Computing Applied to Evaluate the Effects of Buprenorphine on Breathing in Four Congenic Lines of Mice 26
5.1 Overview 26
5.2 Related Work 28
5.3 Methods 29
5.3.1 Dataset Overview 29
5.3.2 Neuromorphic Simulator 31
5.3.3 Ferroelectric Devices 32
5.3.4 Training Algorithm: EONS 33
5.3.5 Experimental Setup 34
5.4 Discussion and Conclusion 53

6 Conclusions 55

Bibliography 58

Vita 63

List of Tables

4.1 Digital RISP Implementations 14
4.2 Pole-balance Difficulty Settings 15
4.3 Network Training Configurations 16

5.1 Digital RISP Implementations 32
5.2 Highest accuracy rate for neuromorphic and scikit-learn implementations for each task for 2 and 4 minute time windows 51

List of Figures

4.1 Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations. 11
4.2 Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations for a noise level of 0.01. The remaining data for a noise level of 0.01 is shown in Figure 4.3 to avoid clutter. 12
4.3 Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations for a noise level of 0.01. The remaining data for a noise level of 0.01 is shown in Figure 4.2 to avoid clutter. 13
4.4 Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations for a noise level of 0.05. The remaining data for a noise level of 0.05 is shown in Figure 4.5 to avoid clutter. 13
4.5 Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations for a noise level of 0.05. The remaining data for a noise level of 0.05 is shown in Figure 4.4 to avoid clutter. 13

4.6 Results for easy (top), medium (middle), and hard (bottom) versions of the pole-balancing task with different read voltages and noise levels, as well as how they compare with various versions of RISP. Note that the fitness access for the hard version only goes to 2000, which corresponds to 40 seconds balanced. 17
4.7 Number of synapses vs. read voltage for easy version of pole-balancing task. 20
4.8 Network structure of a pole-balancing network. The input neurons are across the top and the outputs are at the bottom. Weights are displayed as indices of the fixed weights vector. 21
4.9 Number of synapses vs. read voltage for medium version of pole-balancing task. 22
4.10 Activity vs. read voltage for easy version of pole-balancing task. . . . 23
4.11 Activity vs. read voltage for medium version of pole-balancing task. . 24

5.1 Raw minute ventilation data for 1 mouse for baseline and 10 mg/kg dosages 30
5.2 Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations. 33
5.3 Non-temporal vs temporal encoding 36
5.4 Multi-class classification fitness for various encoders and decoders . . 37
5.5 Binary classification fitness for various encoders and decoders 38
5.6 64 ferroelectric weights 40
5.7 32 ferroelectric weights 41
5.8 8 ferroelectric weights 42
5.9 32 inhibitory ferroelectric weights 43
5.10 8 inhibitory ferroelectric weights 44
5.11 Fitness for binary classification for ferroelectric weight sets in 2 minute time windows 46

5.12 Fitness for binary classification for ferroelectric weight sets in 4 minute time windows 46
5.13 Fitness for multi-class classification for ferroelectric weight sets in 4 minute time windows 47
5.14 Fitness for multi-class classification for ferroelectric weight sets in 2 minute time windows 47
5.15 Number of synapses for multi-class classification for ferroelectric weight sets in 4 minute time windows 48
5.16 Activity for multi-class classification for ferroelectric weight sets in 4 minute time windows 48
5.17 Group classification fitness for various encoders and decoders 50
5.18 Sex classification fitness for various encoders and decoders 52

Nomenclature

ARL	Army Research Laboratory
EONS	Evolutionary Optimization for Neuromorphic Systems
RISP	Reduced Instruction Spiking Processor
SNN	Spiking Neural Network

Chapter 1

Introduction

Spiking neural networks (SNNs) take inspiration from the human brain in the way that they process data. In contrast to traditional artificial neural networks (ANNs), SNNs are well-suited to process temporal data. Time is inherently incorporated into their computation through neuron fires, or spikes, which correspond to discrete events occurring at specific times, simulating the asynchronous nature of biological neurons. This quality of SNNs makes them particularly applicable to real-time data processing and pattern recognition, allowing them to capture relationships in time-series data naturally Tavanaei et al. (2019). Additionally, neuromorphic processors are able to provide greater energy efficiency than traditional low power computing devices Blouw et al. (2019), which is particularly useful in real world use cases with limited size and power resources.

Neuromorphic systems have become popular for hardware implementation in recent years, due to factors such as the end of Moore's law, power demands resulting from Dennard scaling, and the von Neumann bottleneck. Neuromorphic systems are able to address these challenges by incorporating recent advances in artificial intelligence and machine learning. A diverse variety of different types of emerging hardware are being used for these systems, including two-terminal synaptic memory devices based on Mn doped HfO_2, phase change memory, resistive memory,

electrochemical devices, 2D devices, ReRAM, diffusive memristors, phase change memory, spintronics based MRAM, and synaptic transistors Mandal et al. (2014); Wan et al. (2019); Upadhyay et al. (2019). Additionally, hafnia-based ferroelectric devices are promising for use in neuromorphic systems due to their compatibility with CMOS architectures, speed, endurance, retention, and scalability Zhu et al. (2020); Schroeder et al. (2022); Najmaei et al. (2022). In this work, we evaluated the use of these devices as synapses in the context of a benchmark neuromorphic application with various difficulty setting levels, as well as a time-series dataset.

In addition to hardware architecture improvements, the neuroscience field has been another motivation for building neuromorphic systems in recent years. This is because simulating plausible neural behavior on a traditional supercomputer is infeasible due to scale, speed, and power limitations Schuman et al. (2017). In this work, we leveraged neuromorphic computing to enhance our understanding of the impact of buprenorphine on respiration as a function of sex, obesity, and leptin status. We classified buprenorphine dosage level, sex, and mouse group using experimental minute ventilation recordings. We also applied our study of ferroelectric devices as synapses to this classification task.

Neuromorphic computing devices and materials researchers continue to build novel neuromorphic implementations, but do not always know how to best use these implementations. Conversely, applications researchers continue to present interesting problems to solve that can benefit from neuromorphic computing, but may not know how to formulate an appropriate problem in a format that suits neuromorphic computation. Neuromorphic co-design bridge the gap between these two fields by enabling connection between applications and hardware and supporting both fields independently as well. The two studies presented in this work are unique co-design efforts that enable us to understand how hardware and applications interact with one another.

The following sections of this work are structured as follows. Chapter 2 provides an overview of related research. It discusses evolutionary algorithms in relation to

the algorithm used in this work to train SNNs. It also highlights other research in neuromorphic co-design. Chapter 3 details the TENNLab framework, which we use for both studies conducted included in this work. The framework includes all the components used in this study, including neuromorphic processors, training algorithms, encoding and decoding techniques, and applications. Chapter 4 examines the impact of hafnia-based ferroelectric devices as synapses for neuromorphic systems to evaluate their potential for real-world neuromorphic applications. Chapter 5 investigates the applicability of neuromorphic computing for evaluating the impact of buprenorphine on respiration as it relates to sex, obesity, and leptin status, using experimental minute ventilation data collected from mice. Chapter 6 summarizes the research conducted in this work and provides an overview of key insight gained, also discussing potential future areas of research related to this work.

Chapter 2

Literature Review

In this section, we discuss relevant studies in evolutionary approaches and neuromorphic co-design, and highlight how this study is related to other works yet unique. Evolutionary and genetic algorithms are useful for training neural networks because they do not rely on gradient information, which is often difficult or impossible to calculate. They are also less sensitive to starting conditions and can be applied to any neural network architecture. Evolutionary algorithms have been used to train many varieties of neural networks, but we focus on SNNs, which have additional challenges due to hardware constraints that our chosen algorithm is able to address. Neuromorphic co-design seeks to bridge the gap between neuromorphic hardware and applications in order to understand their interaction.

2.1 Evolutionary Approaches

The work presented in Jin et al. (2007) utilize a Pareto-based multi-objective genetic algorithm to optimize classification performance and connectivity of spiking neural networks (SNNs) with the latency coding. Whether or not two neurons are connected, as well as the weight and delay if they are connected, is evolved during optimization. The genetic algorithm is use was adopted using gray coding, where each individual consists of two chromosomes that encode a weight and connection matrix. Another

study López-Vázquez et al. (2019) presents a grammatical evolution based model to design SNNs for supervised classification tasks. It designs SNNs by exploring the search space of three-layered feed forward topologies with configured synaptic weights and delays so that no explicit training is done. Partial connections between input and hidden layers contribute to avoid redundancy and reduce the dimensionality of inputs. Their model produced competitive results on well-known benchmark datasets from the UCI repository. The authors of Vázquez and Garro (2011) use particle swarm optimization (PSO) to adjust the synaptic weights of a spiking neuron as applied to a pattern recognition task. After the synaptic weights are adjusted using PSO, input patterns belonging to the same class generate similar firing rates, but input patterns belonging to other classes generate firing rates that are different enough to be used to differentiate classes. The PSO algorithm is also compared to a differential evolution algorithm when applied to non-linear and real object recognition problems. In this study, we use an evolutionary approach named EONS to generate SNNs. EONS is unique in that it can be used on a diverse variety of neuromorphic systems and will attempt to optimize within each system's unique constraints. Therefore, EONS is able to evaluate inclusion and exclusion of different hardware features, making it a good choice for evaluating ferroelectric devices.

2.2 Neuromorphic Co-Design

The authors of Vineyard et al. (2023) apply game theoretic analysis to computer architecture co-design, in particular the interplay between algorithm and architecture advances in neuromorphic computing, examined as a Stag Hunt game. Two hunters can team up to bring in a higher-value stag, dependent upon the coordination of the two players, or independently secure a less rewarding hare. SNNs are represented by stags, while artificial neural networks (ANNs) are represented by hares. Another study Léonard et al. (2021) presents a theoretical and computational study using free-space propagation and diffraction of light to generate neuromorphic classifiers.

It is proven that system architecture, material structure, and input light field are connected and as such must be co-designed to maximize accuracy. The work presented in Feng et al. (2023) applies the popular deep neural network model, EEGNet, to hardware. An activation-quantization-based (AQB) method converts EEGNet to a spiking neural network and is applied to the standard motion imagery dataset BCIC-IV-2a with around 64% accuracy. The circuit of the AQB EEGNet is implemented on an FPGA, and more energy efficient than using a low power MCU. This work shows that the use of neuromorphic hardware vastly increases energy savings. In this study, we use the TENNLab software framework to understand the interaction between ferroelectric devices and neuromorphic computing, minute ventilation data and neuromorphic computing, and ferroelectric devices and minute ventilation data. We show that ferroelectric devices are well-suited to neuromrophic computation and neuromorphic computing is useful for drawing conclusions from minute ventilation data.

Chapter 3

Previous Work

In this chapter, we discuss previous work that is foundational to the work conducted in the remainder of this thesis. We introduce the TENNLab framework, training algorithm, and neuromorphic model.

3.1 TENNLab Framework

The TENNLab neuromorphic computing software framework Plank et al. (2018) serves as an interface between various neuromorphic processors, algorithms, and applications such as classification and control tasks. The model, algorithm, and applications used in this work are supported by the framework. The general structure of the framework is shown in **ADD FIGURE**.

3.2 Training Algorithm: EONS

The TENNLab framework supports various learning algorithms to train SNNs to solve specific problems. We focus on an evolutionary approach. Evolutionary approaches have been shown to be well-suited to the generation of spiking neural networks. The TENNLab framework supports an evolutionary algorithm called Evolutionary Optimization for Neuromorphic Systems (EONS) Schuman et al. (2020a). EONS

evolves networks by randomly initializing a population of networks, then evaluating each network using a fitness function. The fitness function is typically task-specific. Accuracy is often used for classification tasks, while the reward associated with completing the task is used for control. EONS optimizes a flexible network structure made up of edges and input, output, and hidden nodes. Networks, nodes, and edges can have any number of parameters, such as node leak or threshold, and edge weight or delay. EONS uses tournament selection, crossover, and mutation to produce a child population. The child population becomes the parent in the next epoch, and the process is repeated for a specified number of epochs.

3.3 Neuromorphic Model: RISP

The TENNLab framework supports a variety of hardware implementations, including custom hardware and simulators. In this work, we use the Reduced Instruction Spiking Processor (RISP) spiking neural network model Plank et al. (2022) to simulate networks. RISP supports various spike encoders and spike decoders, as well as different neuron and synapse settings such as delay, threshold, and potential. It utilizes integrate-and-fire neurons with discrete integration cycles and synaptic delays. Synapse weights and neuron thresholds may be configured to be discrete or analog. In processing, spikes are integrated over discrete timesteps. The neuron's action potential at the end of the timestep determines whether the neuron fires. Neurons can either retain their activation potentials or leak away them at each integration cycle. If the neuron has leak enabled and does not fire at the end the integration cycle, its action potential is reset to zero. If leak is not enabled it retains its action potential across timesteps. Synapses only have unit delays, and there are no restrictions on the number of synapses going into or coming out of a neuron. We elected to use RISP because it has simple, parameterizable neuron and synapse models, which is well-suited to our applications. Due to its simplicity, it is trivial to augment RISP for use in specific applications.

Chapter 4

Evaluation of Ferroelectric Devices for Neuromorphic Applications

4.1 Overview

The description of the problem and many of the results in this section come directly from Steed et al. (2024).

Many applications in edge computing require rapid advancement in computational throughput that is unlikely to be met by conventional computing architectures. To overcome this and harness the recent advances in artificial intelligence and machine learning (AI/ML) algorithms, computer architectures need a significant change due to limitations imposed by the slowing of Moore's law scaling and von Neumann bottleneck. Neuromorphic computing seeks to address these challenges through the development of brain-inspired computing platforms.

One of the key features of neuromorphic computers is their energy efficient processing through asynchronous processing and colocation of processing and memory. However, there is also a wide variety of different types of emerging hardware systems that are being considered to implement neuromorphic computing architectures that also contribute to additional energy savings. These hardware implementations vary

from mixed analog-digital systems that are implemented with conventional CMOS devices to architectures that exploit new properties exhibited in emerging memory devices. In this chapter, we focus on exploring the efficacy of one particular class of emerging memory devices—ferroelectric field effect transistors (FeFETs)—in a variety of classification and control tasks.

Hafnia-based ferroelectric devices are a promising platform for future neuromorphic systems due to their CMOS compatibility, speed, endurance, retention, and scalability Zhu et al. (2020); Schroeder et al. (2022); Najmaei et al. (2022). However, it is not clear how these devices will perform in real-world neuromorphic applications. In this chapter, we evaluate hafnia-based ferroelectric devices as synapses in the context of a benchmark neuromorphic application with varying levels of difficulty. We evaluate the impact of the performance of different operating (read) voltages of the ferroelectric device, and we compare the results against a basic neuromorphic implementation. We also analyze the impact of noise on application performance.

The contributions of chapter are:

- A simulation of ferroelectric devices as synapses in a neuromorphic implementation for a benchmark application.
- A comparison of neuromorphic implementations with ferroelectric devices to basic, fully digital, fully CMOS neuromorphic implementations.
- An analysis of the impact of read voltage and device noise on performance, network size, and activity.

4.2 Ferroelectric Device Modeling

In this chapter, we used 3-terminal ferroelectric field effect transistors as a non-volatile memory for storing synaptic weights between connected neurons. For the values of the weights, we generated synthetic device data for conductance versus pulse characteristics shown in Fig. 4.1. The synthetic device data is based on a

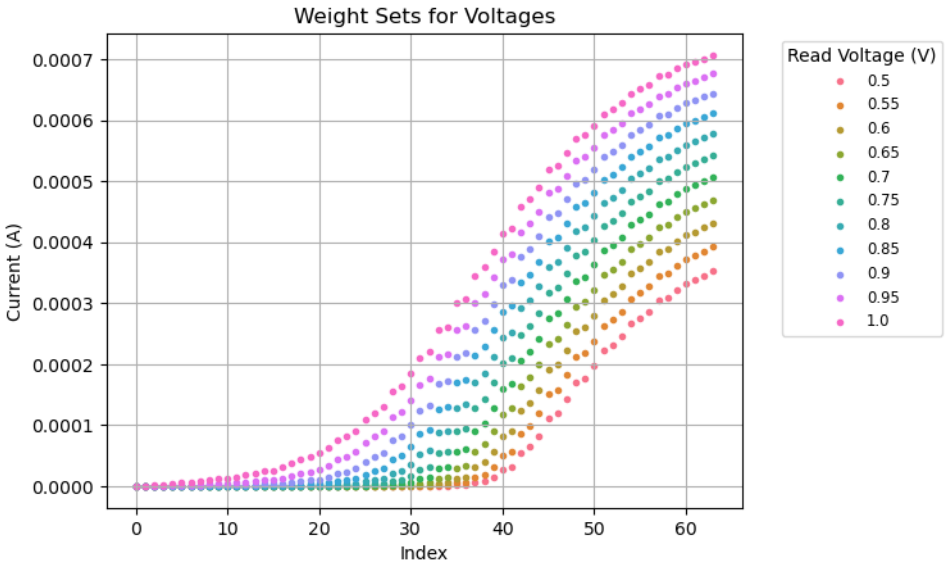


Figure 4.1: Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations.

compact device model for ferroelectric field effect transistors. The model itself is similar to other works Gaidhane et al. (2022); Jerry et al. (2017), and is comprised of two components: (*i*) a metal-ferroelectric-metal (MFM) capacitor connected to the gate of an underlying (*ii*) metal oxide field effect transistor (MOSFET). The model parameters for evaluating the performance of the FeFET are chosen to be similar to common architectures in the literature Jerry et al. (2017). The MFM capacitor uses a multi-domain Preisach model to simulate the ferroelectric polarization switching dynamics Gaidhane et al. (2022). More information on the ferroelectric model can be found elsewhere Sarney et al. (2023). The MOSFET uses the standard BSIM4 model. The subcircuit was simulated using the Xyce simulator Hutchinson et al. (2002). The synthetic data was generated with a drain-source voltage of 100 mV; the data is presented as a function of the gate-source read voltage.

In the present study, we are also interested in understanding the affect of device variability and noise on application-specific performance. Here, we used our ferroelectric device compact model to generate a synthetic representation of device-level noise by introducing normal distributions to the model's parameters with varying

standard deviations. After sampling the model 1000 times, we calculated the resulting average and standard deviation of the transfer characteristics and stored those in a lookup table. The standard deviations used for each noise level are shown Fig. 4.2, Fig. 4.3, Fig. 4.4, and Fig. 4.5. During the network-level simulation, each time a synapse was read, we sampled from a Gaussian distribution with the average and standard deviation values from our lookup table for the given read voltage, noise level, and pulse count.

4.3 Neuromorphic Simulator

We leveraged the device data described above by augmenting the Reduced Instruction Spiking Processor (RISP) spiking neural network model Plank et al. (2022). RISP is a simulator with very simple, parameterizable neuron and synapse models. Here, we use integrate-and-fire neurons with programmable thresholds, alongside synaptic devices with programmable weight and delay values. The original RISP simulator only allows for either continuous or discrete valued weights between specified minimum and maximum values. We augmented the RISP simulator to allow for the user to specify a finite list of possible synaptic weight values rather than a range of values. We then used the device data from the previous section as the possible weight values for

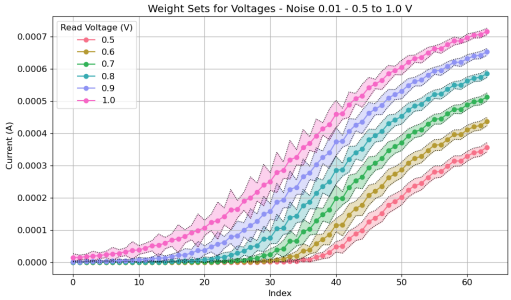


Figure 4.2: Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations for a noise level of 0.01. The remaining data for a noise level of 0.01 is shown in Figure 4.3 to avoid clutter.

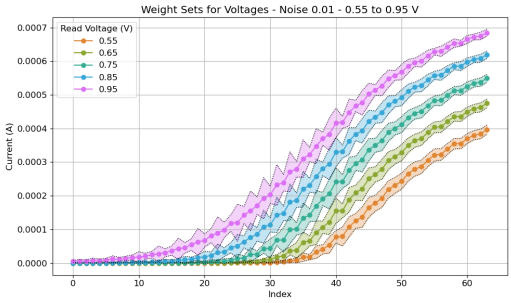


Figure 4.3: Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations for a noise level of 0.01. The remaining data for a noise level of 0.01 is shown in Figure 4.2 to avoid clutter.

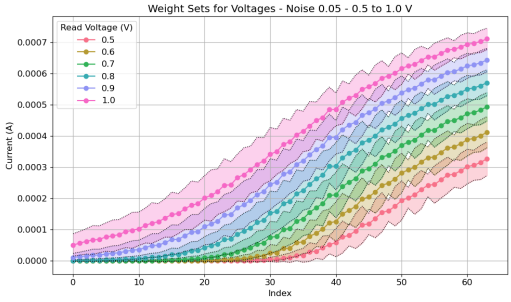


Figure 4.4: Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations for a noise level of 0.05. The remaining data for a noise level of 0.05 is shown in Figure 4.5 to avoid clutter.

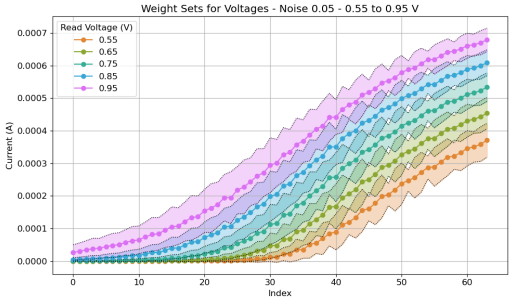


Figure 4.5: Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations for a noise level of 0.05. The remaining data for a noise level of 0.05 is shown in Figure 4.4 to avoid clutter.

the network. We compare the simulations with the ferroelectric device data to fully CMOS implementations of the RISP simulator, shown in Table 4.1.

4.4 Applications

We focused on evaluating the impact of read voltage and device noise on application performance, network size, and activity. We used the pole-balancing or cart-pole task, as we have implemented it in our framework Plank et al. (2019), though the easy version of our implementation is consistent with the CartPole-v1 environment in Gymnasium Towers et al. (2024). In pole-balancing, there is a cart and pole system, where the cart is on a track between two walls. The goal is to apply force to either side of the cart to keep the cart between the two walls while also keeping the pole from falling. The fitness score is how long the pole is kept "balanced", with a maximum score of 15000, which corresponds to keeping the pole balanced for five minutes. We have implemented three versions of this task (easy, medium, and hard), which differ in their observations, actions, and fitness scores, as shown in Table 4.2.

4.5 Experimental Setup

To evaluate the performance of ferroelectric devices as synapses, we evolved networks using a variety of levels of noise, potentiation and depression pulses, weight values,

Table 4.1: Digital RISP Implementations

RISP Version	1	1+	7	15+	127	F	F+
Minimum Weight	-1	1	-7	1	-127	-1	0
Maximum Weight	1	1	7	15	127	1	1
Minimum Threshold	0	1	0	1	0	0	0
Maximum Threshold	1	1	7	15	127	1	1
Minimum Potential	-1	0	0	0	-127	-1	0
Maximum Delay	15	15	15	15	127	15	15
Discrete	T	T	T	T	T	F	F

Table 4.2: Pole-balance Difficulty Settings

Difficulty	Observations	Actions	Fitness
Easy	4 (cart position cart velocity pole position pole velocity)	2 (hit from left hit from right)	length of time balanced
Medium	4 (cart position cart velocity pole position pole velocity)	3 (hit from left hit from right do nothing)	length of time balanced; favors do nothing
Hard	2 (cart position pole position)	2 (hit from left hit from right)	length of time balanced

and pole-balance difficulty levels. We evolved networks using 64 drain current values corresponding to 11 read voltages ranging from 0.5 V to 1.0 V, in 0.05 V increments. We used the 11 unique sets as synapse weight sets, shown in Fig. 4.1. For each weight set, we used we used EONS to train 500 networks for each pole-balance difficulty level. Additionally, we ran tests using the same 11 read voltages, but added noise to the data. We did this for two different noise levels, 0.01 and 0.05. For the noise data, we ran 500 tests for each weight set and focused on the medium and hard pole-balance settings. We trained a large amount of networks in order to explore the variation produced as a result of the random initialization that EONS employs. We compared our results to a basic neuromorphic implementation using various versions of the RISP processor, displayed in Table 4.1. We ran 500 tests for each version of the simulator. We decided to use RISP-F+ as the "default" RISP implementation to compare the ferroelectric synapse results to. We trained a total of 49,000 networks. The tests we ran are broken down in Table 4.3.

For all of our tests, we recorded key data to analyze network performance from both a hardware and software perspective. We tracked network characteristics such

Table 4.3: Network Training Configurations

Runs	Noise	Unique Weights	Difficulty Levels	# Read Voltages/ RISP Settings
500	None	64	3	11 read voltages
500	0.01	64	2	11 read voltages
450	0.05	64	2	11 read voltages
500	None	N/A	3	7 RISP settings

as the number of neurons and synapses. We also tracked the count of neuron and synapse fires to assess the network's activity.

4.6 Results

4.6.1 Performance Comparison with Baseline Implementations

Fig. 4.6 shows the results for the easy, medium, and hard versions of the pole-balancing task with different read voltages and noise levels, as well as how they compare with various versions of RISP. First, it is worth noting that in both the easy and medium versions of the task, the results for the ferroelectric networks are comparable to the closest RISP implementation, RISP-F+. It is clear, however, that the inclusion of negative weights can significantly improve performance, particularly on the medium version of the pole-balancing task, as RISP-F has a higher testing fitness score than RISP-F+ for this task. This has direct implications for practical implementations of ferroelectric-based neuromorphic hardware. For example, this indicates that future neurmorphic systems based on spiking networks should employ ferroelectric devices as inhibitory synapses as well. Second, on average, ferroelectric-based networks performed significantly worse than the CMOS baseline RISP networks; the ferroelectric networks demonstrated comparable performance to the RISP networks that have only a single weight value available to it (RISP-1+). It is possible that restricting the weights from 64 values to fewer (16 or 32), while lowering the synapse

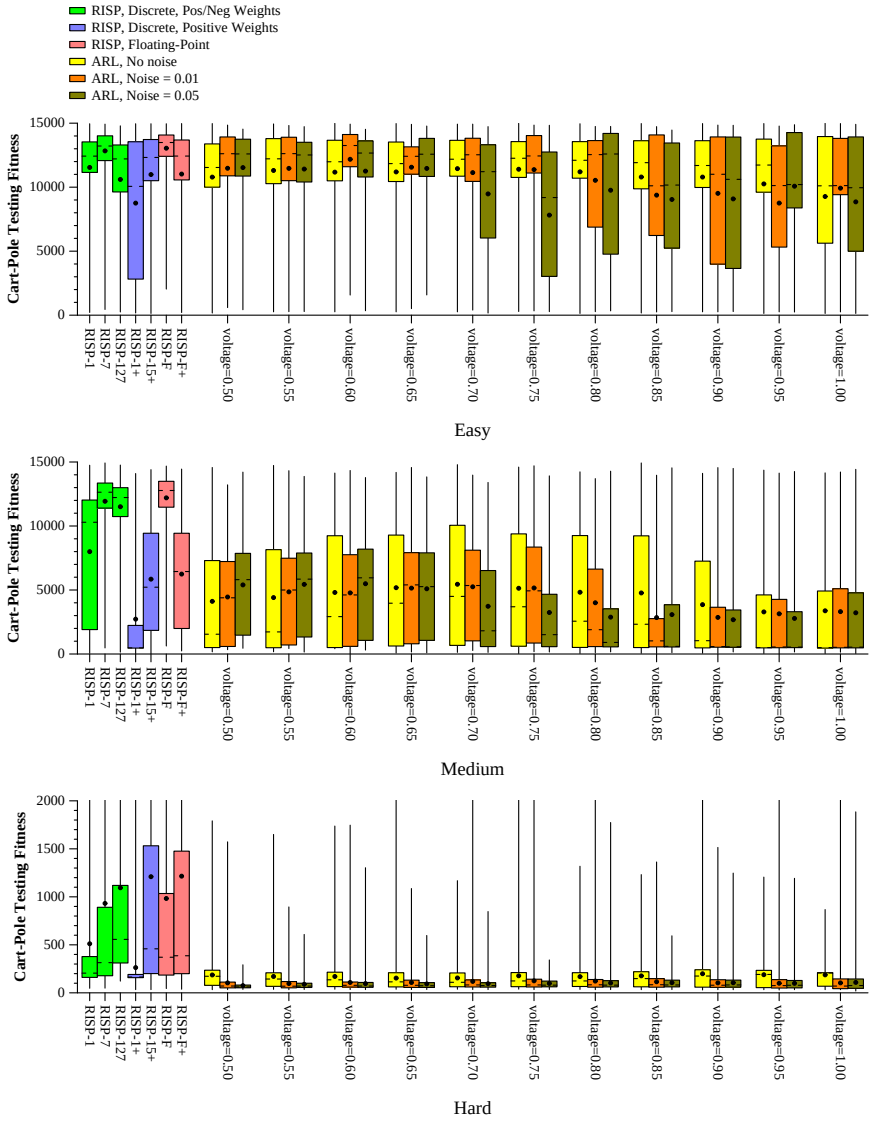


Figure 4.6: Results for easy (top), medium (middle), and hard (bottom) versions of the pole-balancing task with different read voltages and noise levels, as well as how they compare with various versions of RISP. Note that the fitness access for the hard version only goes to 2000, which corresponds to 40 seconds balanced.

precision, may improve the ability of EONS to discover better performing networks by reducing the dimensionality of the parameter search space.

4.6.2 Impact of Read Voltage

We can also see in Fig. 4.6 that for the easy version of the task, low read voltage (less than 0.7) gave the best performance overall and were comparable across read voltages from 0.5 to 0.7, while for the medium version of the task, the best performing networks on average were around a read voltage of 0.7, with other read voltages performing worse. From a device perspective, 0.7 V has the highest dynamic range, which could contribute to its performance. For the hard version of the task, the outliers were best as the read voltage increased from 0.7 to 1.0. This indicates that the best performing read voltage (in terms of performance on the task) is task dependent. If performance on the task is paramount and other factors such as network size and activity do not matter, it may be worthwhile to optimize the read-voltage for each individual task. However, it is also worth noting that in most of the cases, outliers for each read-voltage performed very well. As such, it may be possible to find a well-performing network for any of the read voltages given, if a particular read voltage is needed.

4.6.3 Impact of Noise

Fig. 4.6 also shows results for different levels of noise for the ferroelectric devices. We can see that for the easy and medium versions of the task, ferroelectric networks with noise performed on par with or better than networks without noise, indicating that these devices are still viable for these tasks even with significant noise in their operation. For the hard version of the task, there are some cases (read voltages of 0.7, 0.8, 0.95, and 1.0) where the presence of noise improves the performance of the best performing networks overall. We speculate the inclusion of noise during training may actually help with generalization performance in some cases, as the networks are forced to learn to work within the noise rather than with "perfect" hardware.

4.6.4 Network Size and Structure

Fig. 4.7 shows the network size in terms of synapses for the best performing networks on the easy version of the pole-balancing task (achieving a fitness value of 14,000, or balancing for at least 280 seconds) for the ferroelectric networks in the presence of noise and for the RISP-F+ networks. Note that we only include the RISP-F+ results here as it is the most similar to the ferroelectric devices (positive and continuous weights).

There are several points of interest to note from Fig. 4.7. First, the networks themselves are extremely small, with the vast majority of the networks having fewer than 18 synapses. Second, networks trained with higher levels of noise tended to be the same size or slightly smaller than networks trained without noise. Though this may be counter intuitive, we speculate that EONS trained networks with fewer synapses in the presence of noise on the synapses in order to mitigate issues associated with noise; that is, fewer synapses means less opportunity for noise. Third, networks with higher read voltage values (greater than 0.8) tended to be slightly smaller overall, with median values around 9 synapses, while lower read voltages (0.5-0.65) tended to be slightly larger, with median values around 10-12 synapses. We also include the network sizes for RISP-F+, which are comparable in size to the ferroelectric networks trained without noise. Fig. 4.8 displays the structure of a sample network trained with the ferroelectric weights. Note that there are many connections between the input neurons and there are no hidden neurons used at all.

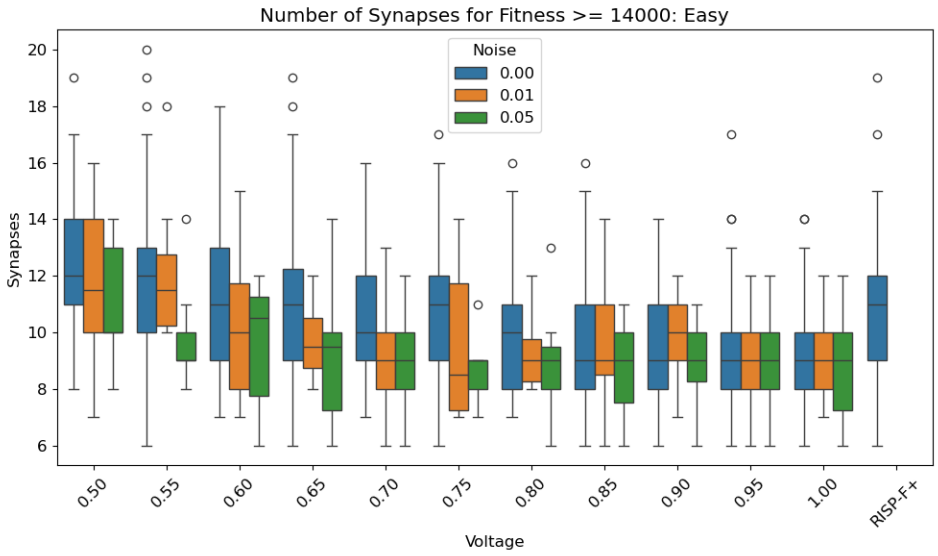


Figure 4.7: Number of synapses vs. read voltage for easy version of pole-balancing task.

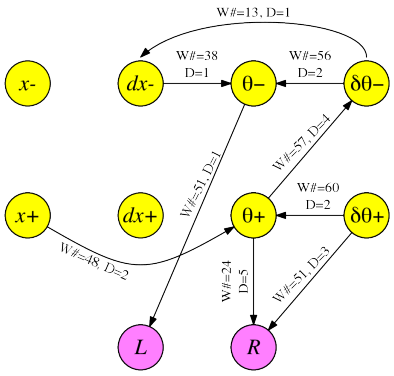


Figure 4.8: Network structure of a pole-balancing network. The input neurons are across the top and the outputs are at the bottom. Weights are displayed as indices of the fixed weights vector.

Fig. 4.9 shows the network size in terms of synapses for all of the networks from the medium version of the pole-balancing task (achieving a fitness value of 12,000, or balancing for at least 240 seconds). In this case, we see that the networks are slightly larger than those in the easy version of the task. We also see the similar trends for the medium version, i.e., that networks with more noise tend to be slightly smaller and that networks with higher read voltage tend to be slightly smaller. Once again, we see that the RISP-F+ networks are comparable in size to lower read voltage ferroelectric networks without noise. We omit the network size results for the hard version of the pole-balancing problem, as the performance of those networks is not very good overall.

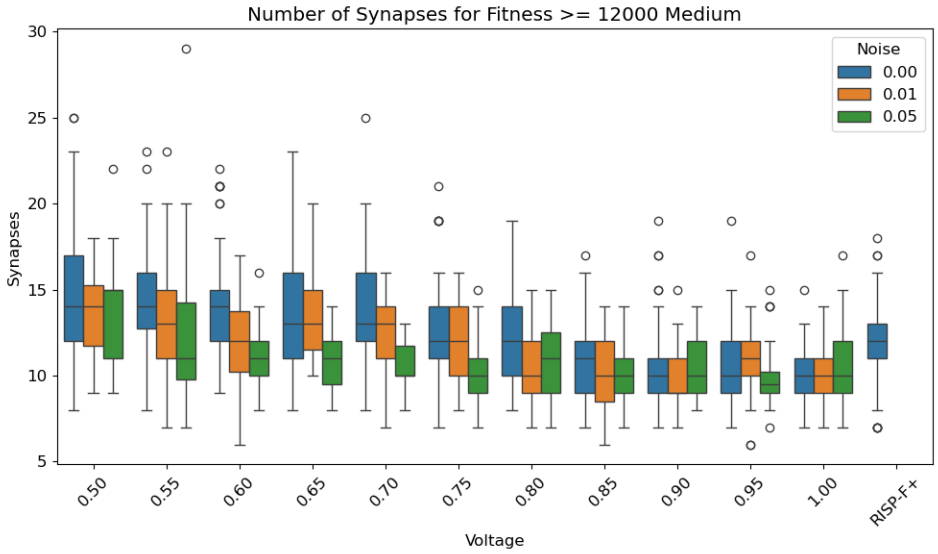


Figure 4.9: Number of synapses vs. read voltage for medium version of pole-balancing task.

4.6.5 Network Activity

Fig. 4.10 shows the results for activity in terms of synapse events for the easy version of the pole-balancing task. We only include networks that were able to balanced for at least 280 seconds. Here, we see that the inclusion of noise can cause an increase in activity in many cases. We can also see that higher read voltages corresponded to higher activity values. This is perhaps unintuitive based on Fig. 4.7, where we saw that the network sizes were smaller for higher read voltages. This tells us that though these networks are smaller, that does not necessarily correspond to less activity, and thus less energy.

Fig. 4.11 shows the results for activity in terms of synapse events for the medium version of the pole-balancing task. Here, we only include networks that were able to balance for at least 240 seconds. Here we can see that neither read voltage nor noise had a meaningful impact on activity; however, it is also worth noting that nearly all of the ferroelectric networks required more activity than the corresponding RISP-F+ network. This indicates that fewer weight values being available may require more activity for this harder version of the task.

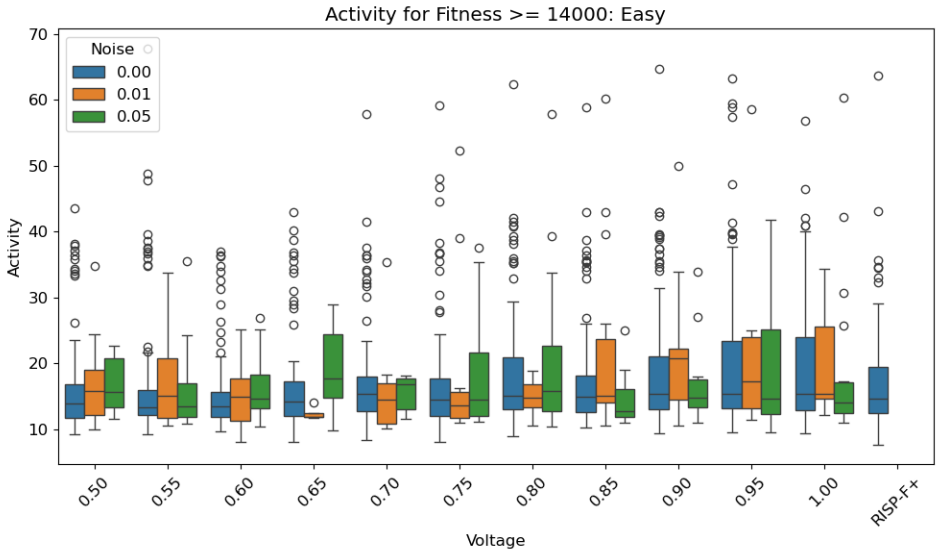


Figure 4.10: Activity vs. read voltage for easy version of pole-balancing task.

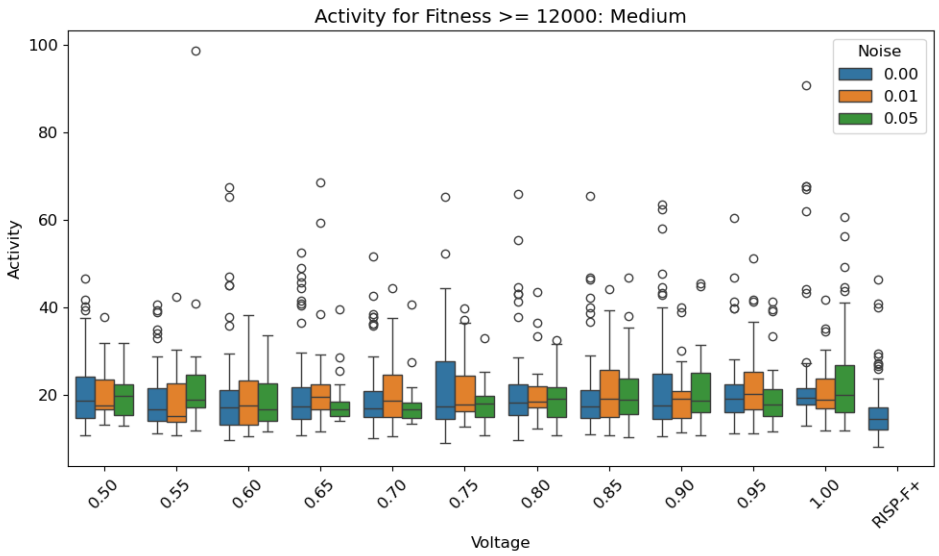


Figure 4.11: Activity vs. read voltage for medium version of pole-balancing task.

4.7 Discussion and Conclusion

In this chapter, we evaluate the performance of ferroelectric devices on multiple versions of the pole-balancing task. We compare how the ferroelectric networks compare against basic neuromorphic implementations, as well as how different read voltages and noise levels in the devices impact performance.

Several key points arose as part of this study. First, the ferroelectric networks are able to achieve comparable performance to basic neuromorphic implementations on both the easy and medium versions of the pole-balancing task, with the best performing networks on the hard task also achieving competitive performance. This indicates that these devices are capable of neuromorphic computation and that any limitations they may have can be overcome with additional optimization and training.

In terms of the impact of read voltage on the performance of the networks, we found that the best performing read voltage was dependent on the task itself, though once again, all read voltages could be optimized to produce high performing networks. This indicates that, no matter the read voltage selected, it is still possible to produce well-performing networks.

Additionally, we were also able to show that though noisy devices can have an impact on performance, often that impact can either improve performance or can be mitigated through additional optimization runs (to achieve the performance seen by the outliers). As such, though we expect there to be noise in the operation of these devices, we can still train to produce well-performing networks, even under noisy operating conditions.

In terms of network size, we were able to show that very small networks with relatively low activity could still achieve good performance. This is important for evaluation on real hardware and the development of hardware platforms that can be deployed to scenarios that have size, weight, and power restrictions.

In future work, we plan to investigate additional tasks, including temporal data classification tasks, event-based camera processing, and autonomous robots. We also plan to investigate the interplay between the neuron model and the ferroelectric synapses, as well as investigate more limited weight sets from the ferroelectric devices (lower precision memory). Additionally, we plan to validate our results on experimental data, in addition to continuing to refine our simulated devices to accurately reflect real device behavior.

Chapter 5

Neuromorphic Computing Applied to Evaluate the Effects of Buprenorphine on Breathing in Four Congenic Lines of Mice

5.1 Overview

Machine learning and other computational approaches have been used for decades to provide insights and form better understanding of experimental data across a variety of domains, including medical applications Deo (2015). Neuromorphic computing approaches are attractive for use in implementing machine learning applied to medical applications for a variety of reasons, including the ability to deploy to edge systems, where data could be monitored in real time, as well as the temporal processing capabilities offered by spiking neural networks. In this work, we investigate the ability of neuromorphic computing approaches to understand the impact of buprenorphine on respiration. Neuromorphic architectures are attractive for this application because

they enable lower power consumption, longer battery life, and less invasive sensors than traditional CMOS architectures Aboumerhi et al. (2023).

Buprenorphine is a synthetic opioid that is used to treat pain and opioid use disorders. Opiate-induced respiratory depression is associated with increased risk for obese and female humans. A functional measure of breathing in mice is minute ventilation, which refers to mL/min/g body weight of air that moves in and out of the lungs per minute. Obesity is associated with dysfunction of the satiety hormone, leptin and/or leptin receptors. A previous study found that in C57BL/6J (B6) mice, minute ventilation was significantly decreased by buprenorphine as a function of leptin dysfunction, but not sex Angel et al. (2018). Buprenorphine administered to intact, behaving mice also caused a decrease in minute ventilation variability. Another study found that buprenorphine caused dose-dependent decreases in respiratory frequency while simultaneously increasing minute ventilation Glovak et al. (2022). The effects of buprenorphine on breathing varied with leptin status and decreased minute ventilation variability in all four lines of mice. However, in contrast to the previous study, the effects of buprenorphine also varied significantly with sex.

In this study, we use neuromorphic computing to enhance our understanding of the impact of buprenorphine on respiration as a function of sex, obesity, and leptin status. Modern edge computing technology requires rapid advancements that conventional computing architectures struggle to meet. To leverage artificial intelligence and machine learning (AI/ML) algorithms in embedded systems, computer architectures must be changed due to limitations resulting from Moore's law scaling and the von Neumann bottleneck. Neuromorphic, brain-inspired computing can address these challenges. In this study, we trained spiking neural networks (SNNs) to classify dosage, mouse group, and sex based on mouse minute ventilation data.

Neuromorphic computers are able to process data efficiently through asynchronous processing and colocation of processing and memory. Additionally, implementations of neuromorphic computing architectures employ different types of hardware systems that contribute to energy savings and decrease latency, improving energy efficiency

and speed of computation. These systems include mixed analog-digital systems implemented with conventional CMOS devices as well as architectures that leverage emerging memory devices. In this work, we explored the efficacy of ferroelectric field effect transistors (FeFETs), a class of emerging non-volatile memory, for our classification task. Hafnia-based ferroelectric devices show potential for neuromorphic systems. Ferroelectric field effect transistors are appealing due to their CMOS compatiblility, speed, and scalabibility, retention, and endurance Zhu et al. (2020); Schroeder et al. (2022); Najmaei et al. (2022). However, these devices are yet largely unevaluated on many real-world neuromorphic applications. In this study, we evaluated hafnia-based ferroelectric devices as synapses to classify mouse minute ventilation data by dosage. We evaluated the impact of the performance of three different read voltages of the ferroelectric device and compared the results to a basic, digital neuromorphic implementation.

This key contributions of this work are:

- An exploration of mouse minute ventilation data for four congenic lines of mice, separated by buprenorphine dosage and sex.
- An application of neuromorphic computing to experimental minute ventilation data.
- An evaluation of hafnia-based ferroelectric devices for neuromorphic synapses on a real-world application.

5.2 Related Work

There have been several efforts in applying neuromorphic computing or spiking neural networks to analyze medical data. One study was conducted to detect sleep apnea based on electrocardiogram (ECG) data Tyagi and Agrawal (2024). The authors employed leaky integrate and fire SNNs using temporal encoding to transform the ECG signal into spikes, achieving around 95% accuracy. Another study compared

the performance of a deep two-dimensional convolutional neural network (CNN) with a spiking neural network to detect sleep apnea based on ECG data Salam and Rafi (2023). The SNN outperformed the CNN, obtaining 89% accuracy. Another work used a feed-forward artificial neural network to detect sleep apnea using measure of ECG and oxygen saturation level signal data T et al. (2022). The models trained using oxygen saturation level outperformed the ECG models, achieving 91% and 80% accuracy, respectively.

5.3 Methods

To evaluate the effect of buprenorphine injections on mice, we leveraged the TENNLab Neuromorphic Computing software framework Plank et al. (2018). The TENNLab framework provides a common interface for network and neuromorphic processor representations, as well as spike encoding and decoding approaches. The TENNLab software framework includes support for a variety of spiking neural network and neuromorphic hardware simulators, algorithms, and applications.

5.3.1 Dataset Overview

The dataset contains minute ventilation readings for mice of varying sex, buprenorphine dosage, obesity, and leptin statuses. The four different groups of mice studied were:

- **C57BL/6J (B6)**: Common inbred strain of laboratory mouse with a normal weight
- **B6.BKS(D)-Leprdb/J (db/db)**: Mice with mutated leptin receptor, causing morbid obesity
- **B6.Cg-Lepob/J (ob/ob)**: Mice with gene mutation that disrupts production of the leptin

- **DIO (diet induced obesity)**: B6 mice raised on 60% fat diet

Ten male and ten female mice were included in each group except for the DIO group, which contains twelve male mice. Female mice were excluded from the DIO group because female and male mice gain weight at different rates Casimiro et al. (2021). Minute ventilation data was recorded after systemic administration of saline for sixty minutes. Recordings were taken in 10 second bins. Recordings of breathing were obtained at baseline (no injection), a saline control. A baseline control dosage of saline was administered to each mouse, as well as 0.1, 0.3, 1, 3, and 10 mg/kg dosages of buprenorphine. Fig. 5.1 displays a sample of the data for 1 mouse.

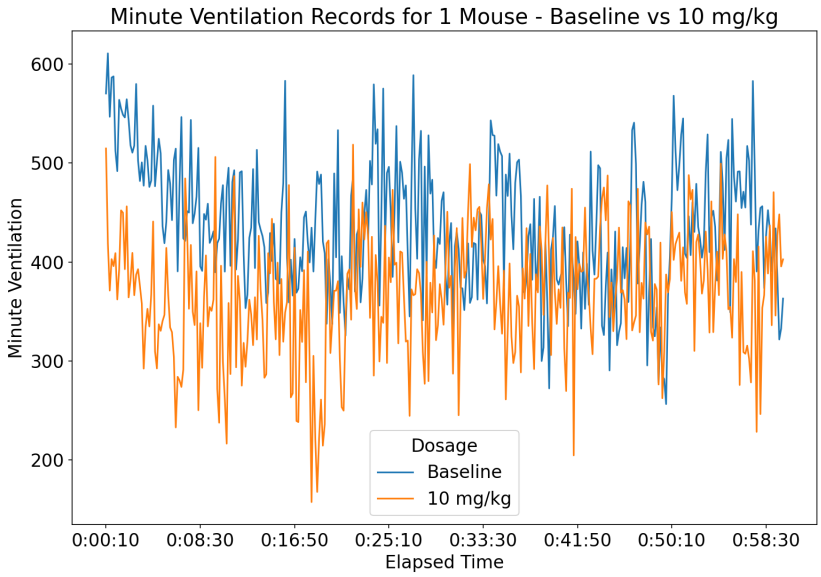


Figure 5.1: Raw minute ventilation data for 1 mouse for baseline and 10 mg/kg dosages

5.3.2 Neuromorphic Simulator

We used the Reduced Instruction Spiking Processor (RISP) spiking neural network model Plank et al. (2022) to simulate the networks. We chose to use RISP because it has very simple, parameterized neuron and synapse models, and is computationally efficient to simulate. We investigated the effects of various RISP settings such as threshold, potential, and delay, shown in Table 5.1. We also evaluated the use of different spike encoders, described below:

- **Val**: Features produce one spike at time 0 whose value is scaled between 0 and 1.
- **Spikes-1**: Features produce one spike at time 0 whose value is 1. The minimum interval between spikes is 1.
- **Spikes-2**: Spikes-1, but the minimum interval between spikes is 2.
- **BinSpike**: Features are converted to a spike count, and those spikes are produced at fixed intervals all having values of 1. 8 bins are used.
- **BinVal**: Features produce one spike at time 0 whose value is scaled between 0 and 1. 8 bins are used.

For encoders using bins, multiple neurons, or "bins" are used to encode values, where each value is converted to a (bin, percentage) pair. Spikes are applied to the neuron corresponding to the bin and the percentage determines the spikes, values, and times within the bin. Lastly, we evaluated two different spike decoders, described as follows:

- **Count**: Spikes on a single neuron are counted and turned into a number, which is the output.
- **TTLS (Time-to-last-spike)**: The time of the last spike is used as the output, where later spikes correspond to higher values.

Table 5.1: Digital RISP Implementations

RISP Version	1	1+	7	15+	127	F	F+
Minimum Weight	-1	1	-7	1	-127	-1	0
Maximum Weight	1	1	7	15	127	1	1
Minimum Threshold	0	1	0	1	0	0	0
Maximum Threshold	1	1	7	15	127	1	1
Minimum Potential	-1	0	0	0	-127	-1	0
Maximum Delay	15	15	15	15	127	15	15
Discrete	T	T	T	T	T	F	F

We chose these particular spike encoders and decoders to evaluate the TENNLab framework's capabilities and determine which encoding and decoding methods were best suited to our data.

5.3.3 Ferroelectric Devices

In addition to evaluating the data with non-specific synapse weights, we set the weights to certain drain current values associated with simulated ferroelectric devices operating at 3 different read voltages. We used 3-terminal ferroelectric field effect transistors to store synaptic weights between connected neurons as non-volatile memory. We generated synthetic device data for conductance versus pulse characteristics to obtain the values of the weights shown in Fig. 5.2. The index refers to the number of pulses at each conductance value. The device model for ferroelectric field effect transistors is similar to other works Gaidhane et al. (2022); Jerry et al. (2017). The model includes two components: (*i*) a metal-ferroelectric-metal (MFM) capacitor connected to the gate of a (*ii*) metal oxide field effect transistor (MOSFET). The model parameters used to evaluate the performance of the FeFET are similar to common architectures Jerry et al. (2017). The MFM capacitor uses a multi-domain Preisach model for simulation of the ferroelectric polarization switching dynamics Gaidhane et al. (2022); Sarney et al. (2023). The MOSFET uses the BSIM4 model, and the subcircuit was simulated using the Xyce simulator Hutchinson et al.

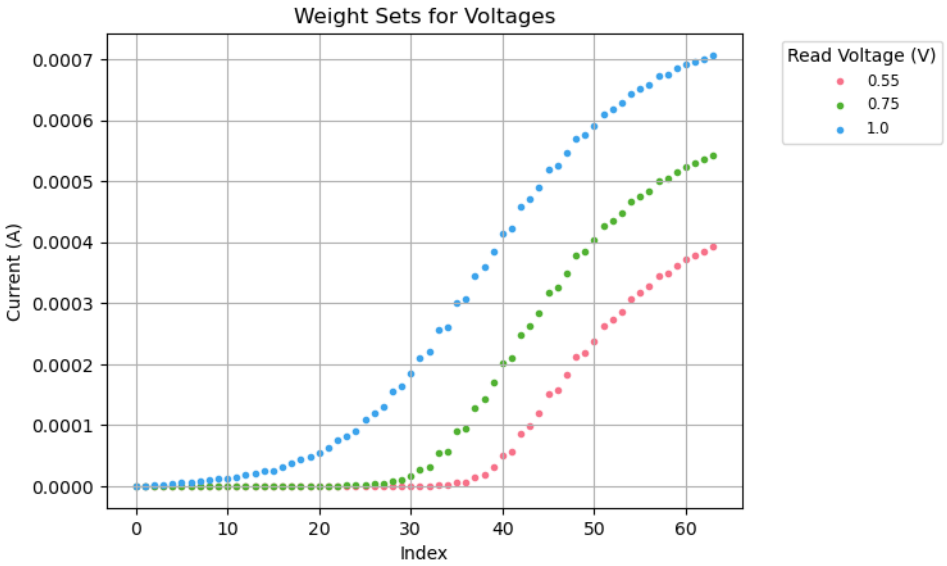


Figure 5.2: Conductance versus pulse data from FeFET compact model used as weight sets within the neural network simulations.

(2002). The synthetic data was generated using a drain-source voltage of 100-mV and is presented as a function of the gate-source read voltage.

We leveraged this ferroelectric device data by augmenting the RISP spiking neural network model. Here, we used integrate-and-fire neurons with programmable thresholds alongside synaptic devices with programmable weight and delay values. The original RISP simulator only allows for either continuous or discrete valued weights between specified minimum and maximum values. We augmented the RISP simulator to allow for the user to specify a finite list of possible synaptic weight values rather than a range of values. We then used the ferroelectric device data as the possible weight values for the network.

5.3.4 Training Algorithm: EONS

The training approach we used in this work is Evolutionary Optimization for Neuromorphic Systems (EONS) Schuman et al. (2016, 2020b). EONS uses an evolutionary algorithm to optimize the topology and parameters of spiking neural networks for various applications. EONS optimizes within the context of a particular

hardware configuration and application. Thus, EONS is customizing the network design (network topology and parameters) for the particular hardware and application of interest. As such, EONS begins with an application and hardware implementation in mind, as well as a fixed encoding and decoding approach. Then EONS creates a population of randomly initialized networks that have the same input and output neurons (as dictated by the application and input/output encoding schemes) but different connectivity and hidden neuron initialization. Then, each network in the population is evaluated using a fitness score. This score is application-dependent but is structured so that higher scores correspond to better performance on the application. The fitness scores for the networks in the population drive the selection and reproduction process. EONS uses tournament selection to select parents from a population of networks. These parents undergo reproduction operations, such as crossover and mutation, to produce children networks that inherit characteristics from their parents. The children networks replace the population and the evaluation, selection, and reproduction process is repeated over the course of several generations.

5.3.5 Experimental Setup

Dosage Classification

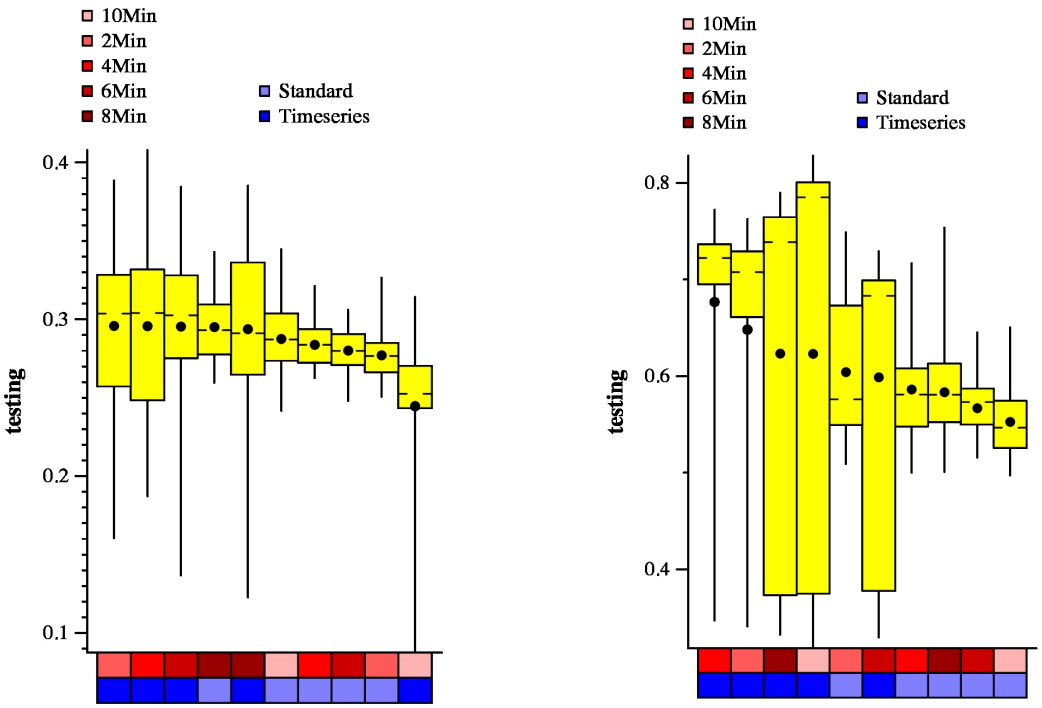
In our initial experiments, we focused on classifying dosage levels from minute ventilation using five different time windows: 2, 4, 6, 8, and 10 minutes. We evaluated this task using both non-temporal encoding and temporal encoding. By testing time windows of various lengths, we sought to determine the optimal level of granularity in the data for distinguishing between dosage levels. In real world applications, shorter time windows might be favorable because they allow for quicker detection, but longer windows might be favorable for cases involving periodic monitoring. We formulated two different classification tasks:

- Binary: Baseline and 10 mg/kg buprenorphine

- Multi-class: Baseline, 0.1, 3, and 10 mg/kg buprenorphine

As expected, the networks were able to perform better on binary classification than multi-class classification. The binary classification task was able to achieve around 68% accuracy, while the multi-class task only achieved around 30% accuracy, shown in Fig. 5.3. For both tasks, it is clear that temporal encoding yields better performing networks than non-temporal encoding. Additionally, the 2 and 4 minute time windows yielded the best performance for both tasks. We compared our results to standard scikit-learn methods Pedregosa et al. (2011) as a baseline, and found that our networks performed better on average. For the binary classification task, the scikit-learn MLPClassifier achieved roughly 52% and 56% testing accuracy for the 2 and 4 minute time windows, respectively, when fed the same raw data as our neuromorphic implementation. All scikit-learn MLPClassifiers trained in this study used the ReLU activation function, the Adam solver, a train-test-split of 67/33, a learning rate of 0.001, one hidden layer with 100 neurons, and ran for a maximum of 1000 iterations. For the multi-class task, the scikit-learn MLPClassifier achieved 26% and 28% testing accuracy for the 2 and 4 minute time windows, respectively.

We repeated these two dosage classification tasks, narrowed down to the two top performing time windows of 2 and 4 minutes, and varied the spike encoder and decoder settings. The results are shown in Fig. 5.4 and Fig. 5.5. For both tasks, the 4 minute time window yielded the best performance, achieving around 79% accuracy on the binary task and 41% accuracy on the multi-class task. The spike and val encoders using eight bins performed the best for both tasks, however the decoder results varied across tasks. Multi-class classification benefited from using the count decoder significantly over TTLS. The TTLS decoder yielded the best performance for binary classification, but the count decoder achieved similar performance.



(a) Multi-class classification

(b) Binary classification

Figure 5.3: Non-temporal vs temporal encoding

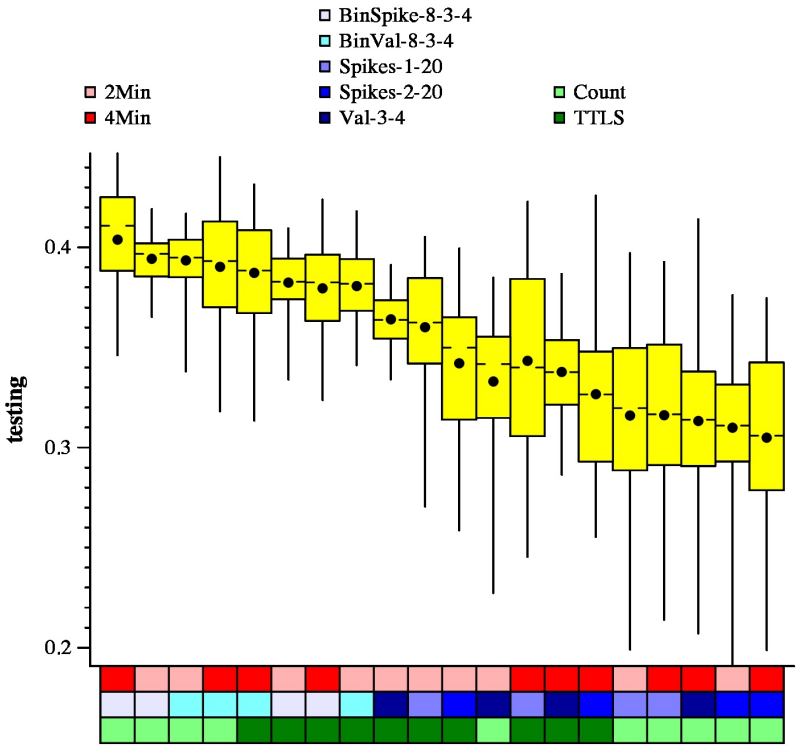


Figure 5.4: Multi-class classification fitness for various encoders and decoders

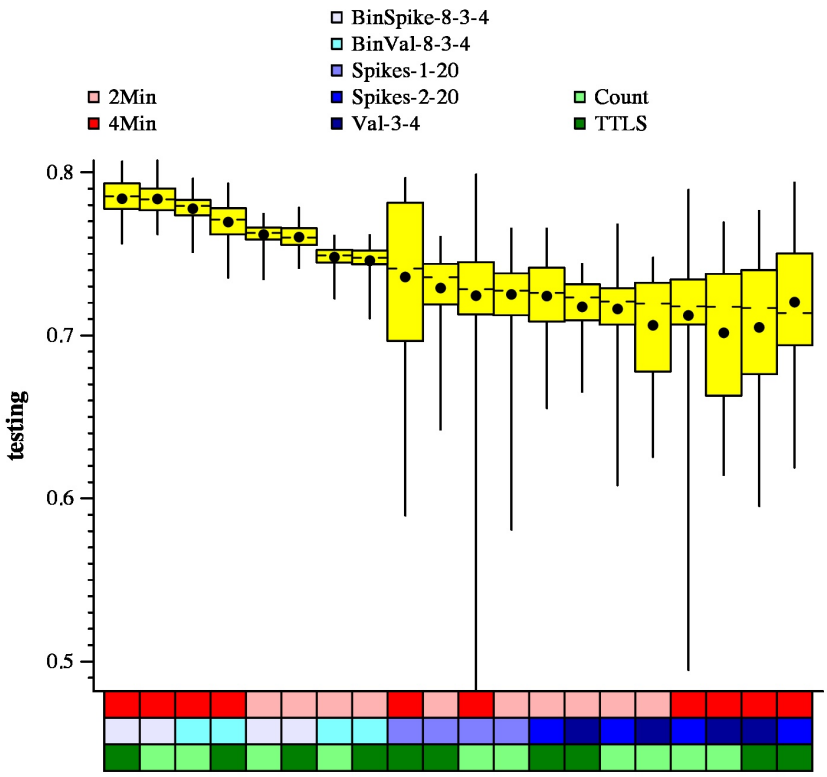
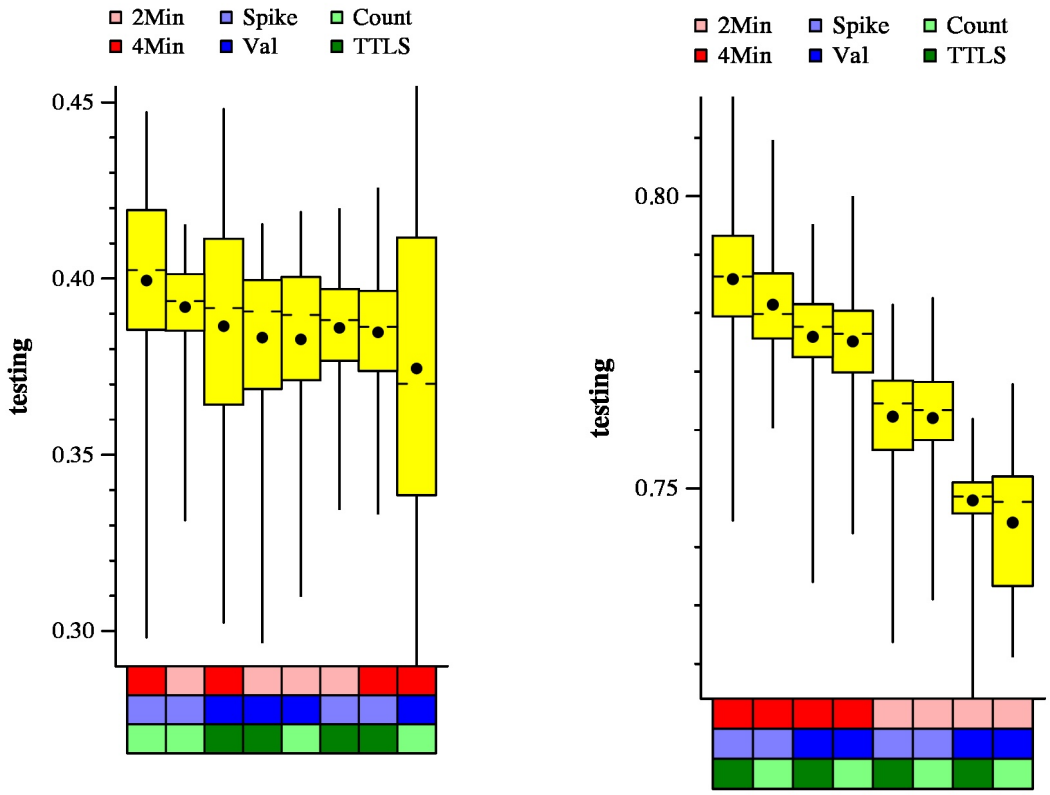


Figure 5.5: Binary classification fitness for various encoders and decoders

Next, we incorporated the ferroelectric device weights into our study. We used drain current values corresponding to 0.55, 0.75, and 1.0 V as weights. These unique read voltages relate to various device characteristics, such as dynamic range, bit variability, memory level resolution, and other performance metrics. Using different read voltage values allows for a comprehensive understanding of how device-level characteristics impact overall network behavior, facilitating the evaluation and optimization of device processing and operations in relation to network requirements. We trained networks using five variations of these weight values per voltage:

- **64 weights**: Both potentiation and depression pulses
- **32 weights**: Potentiation pulses only
- **8 weights**: Limited set of potentiation pulses, equally spaced
- **32 inhibitory weights**: Potentiation pulses only, both positive and negative counterparts
- **8 inhibitory weights**: Limited set of potentiation pulses, both positive and negative counterparts

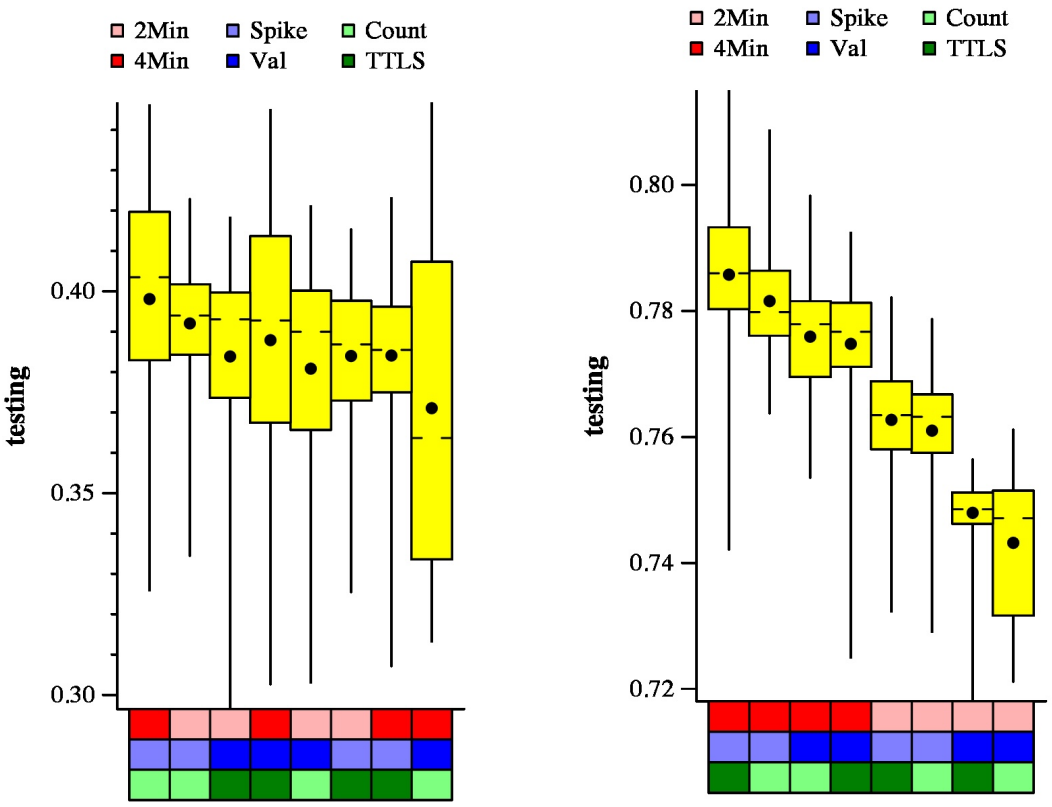
Based on our findings from the previous encoder and decoder tests, we only tested the two best performing encoders along with both decoders and only used 2 and 4 minute windows. The results are shown in Fig. 5.6-Fig. 5.10, where "Spike" and "Val" refer to BinSpike and BinVal, as described in section 5.3.2. Again, we attempted binary and multi-class classification. All five variations of the weight sets were able to obtain roughly the same accuracies for the two tasks. However, the inhibitory weights performed slightly worse than the other weight sets for binary classification.



(a) Multi-class classification

(b) Binary classification

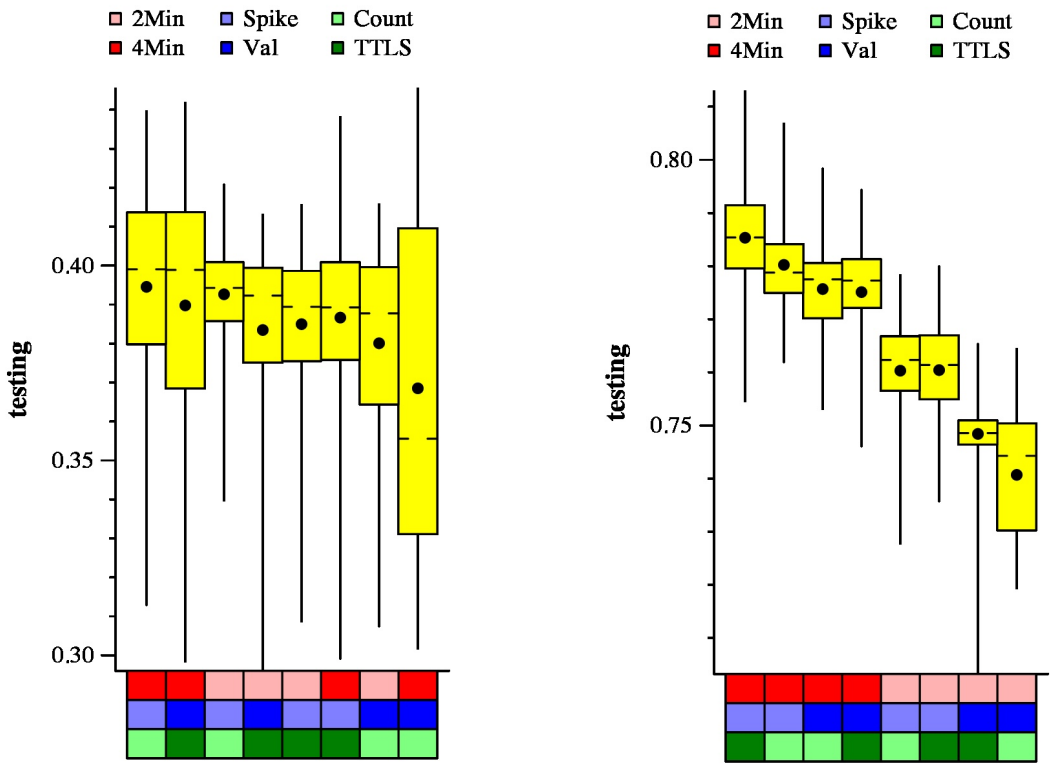
Figure 5.6: 64 ferroelectric weights



(a) Multi-class classification

(b) Binary classification

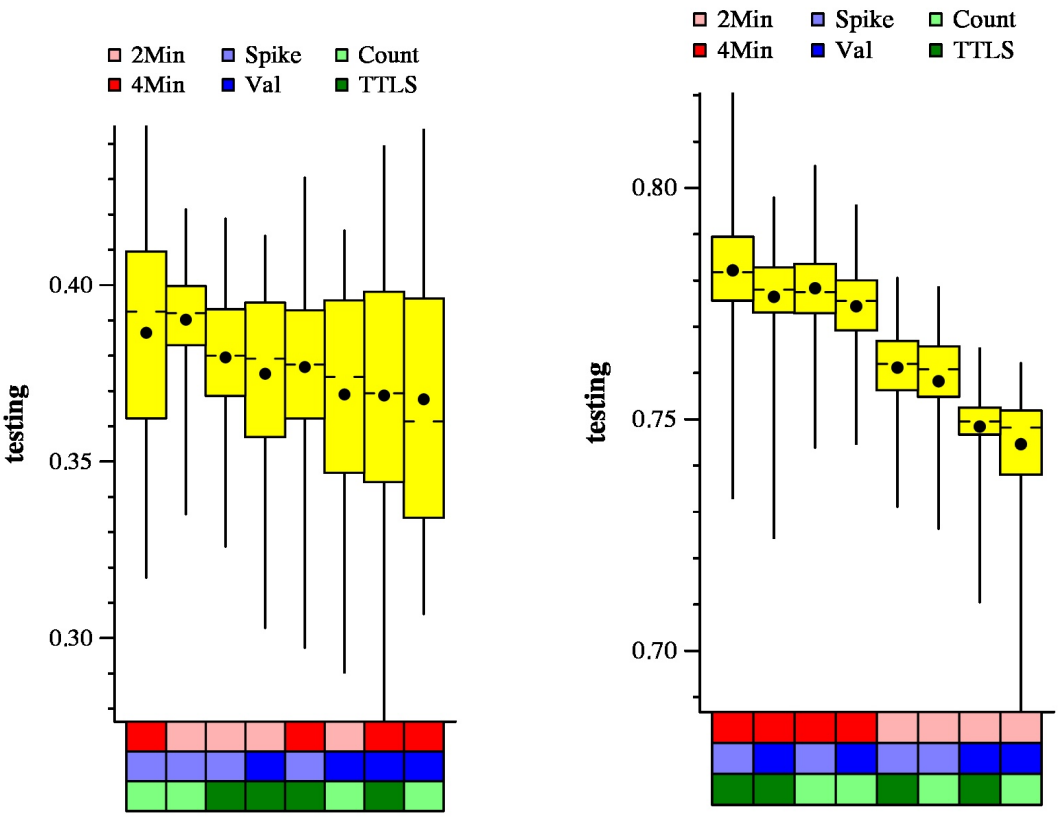
Figure 5.7: 32 ferroelectric weights



(a) Multi-class classification

(b) Binary classification

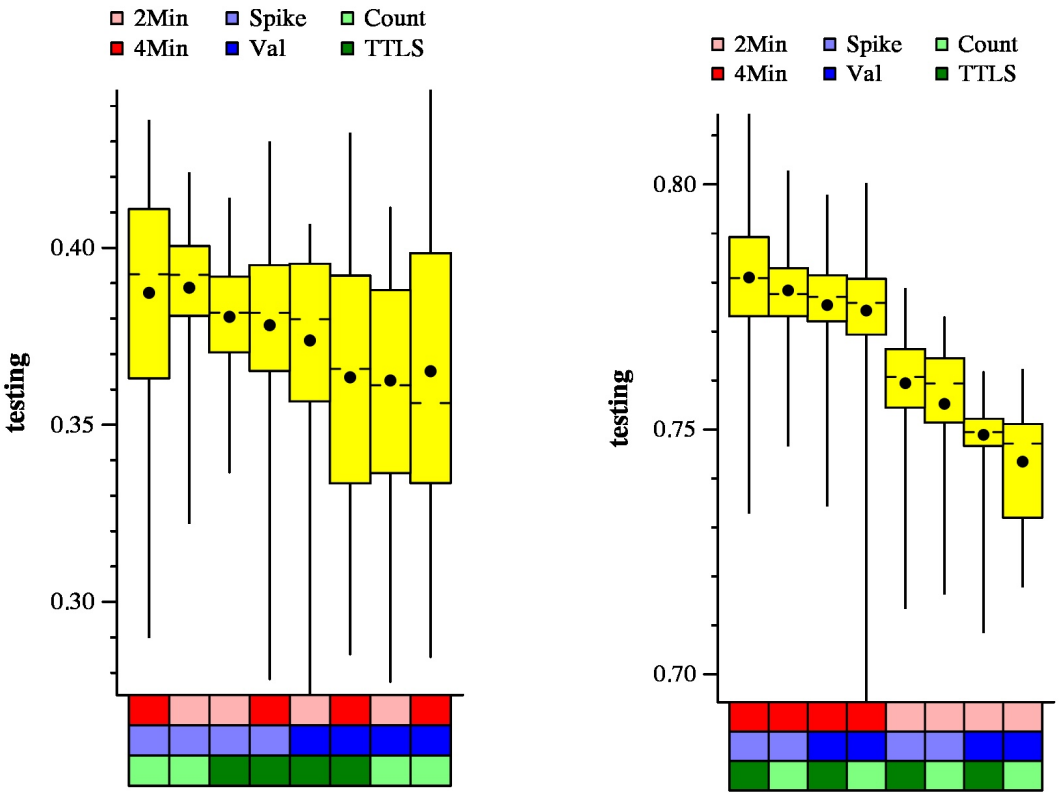
Figure 5.8: 8 ferroelectric weights



(a) Multi-class classification

(b) Binary classification

Figure 5.9: 32 inhibitory ferroelectric weights



(a) Multi-class classification

(b) Binary classification

Figure 5.10: 8 inhibitory ferroelectric weights

Across all weight sets, the 4 minute time window achieved significantly better performance than the 2 minute window for the binary task. For the multi-class task, the 4 minute window achieved the best performance for all weight sets, but the 2 minute window produced competitive results. The spike encoder yielded notably better results than the val encoder for all weight sets for the binary task. However, the val encoder achieved competitive performance for the inhibitory weights in particular, shown in Fig. 5.9b and 5.10b.

For multi-class classification, the encoder results are similar. The spike encoder produced the best networks for all weight sets. However, there is a larger disparity between the two encoders' performance for the two inhibitory weight sets, seen in Fig. 5.9a and 5.10a. The TTLS decoder yielded the best performance for the binary task, while the count decoder performed the best for the multi-class task.

In addition to comparing the performance of different spike encoders, spike decoders, and time windows across weight sets, we compared the performance of the 3 different read voltages used in this study. Fig. 5.11 and Fig. 5.12 show that for the binary classification task, read voltage did not have a significant impact on performance. Similarly, for the multi-class task with a 4 minute time window, Fig. 5.13 read voltage does not appear to significantly impact performance. However, Fig. 5.14 indicates that for the 2 minute time window, 1.0 V produces slightly more accurate networks than the 0.55 or 0.75 V. In particular, the inhibitory weight sets' ranges of accuracy scores are smaller and higher for 1.0 V.

In addition to investigating the five different weight sets impact on fitness, we were interested in comparing network size and activity for hardware applications. Inhibitory weights consistently produced larger networks, having roughly 5-10 more synapses than networks with the other weight sets, shown in Fig. 5.15. However, the inhibitory weight sets tended to produce networks with lower activity, shown in Fig. 5.16. The average activity decreased by roughly 0.1 neuron spikes.

We also evaluated the impact of read voltage on network size. Fig. 5.15 shows that 1.0 V produced smaller networks than the other read voltages, particularly in

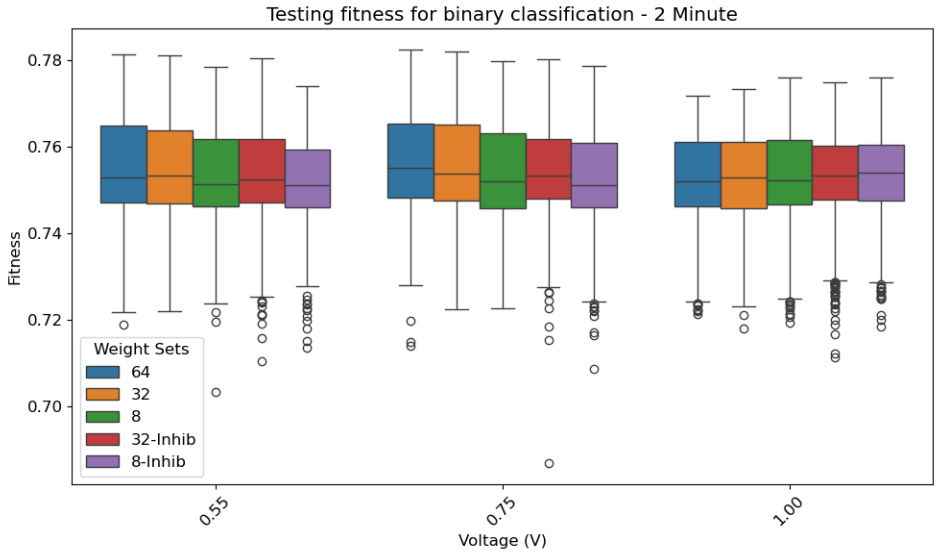


Figure 5.11: Fitness for binary classification for ferroelectric weight sets in 2 minute time windows

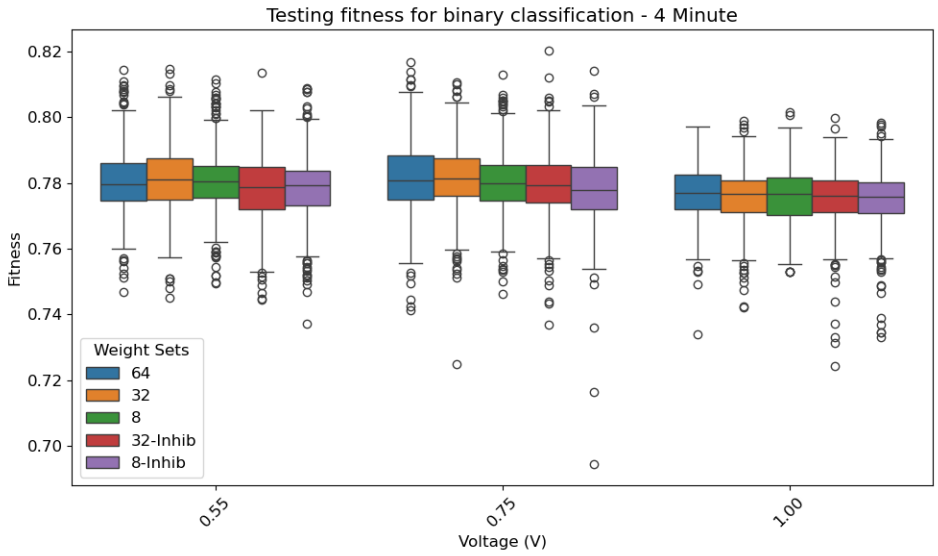


Figure 5.12: Fitness for binary classification for ferroelectric weight sets in 4 minute time windows

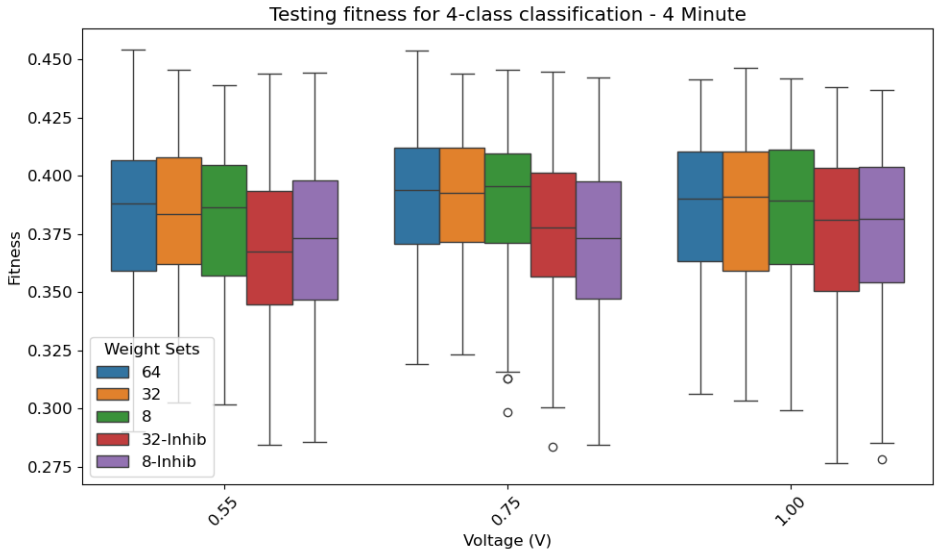


Figure 5.13: Fitness for multi-class classification for ferroelectric weight sets in 4 minute time windows

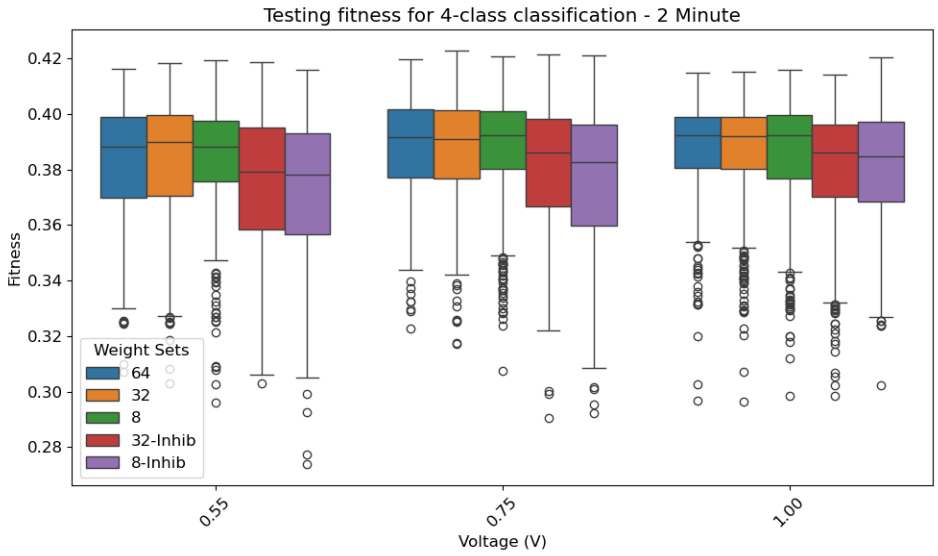


Figure 5.14: Fitness for multi-class classification for ferroelectric weight sets in 2 minute time windows

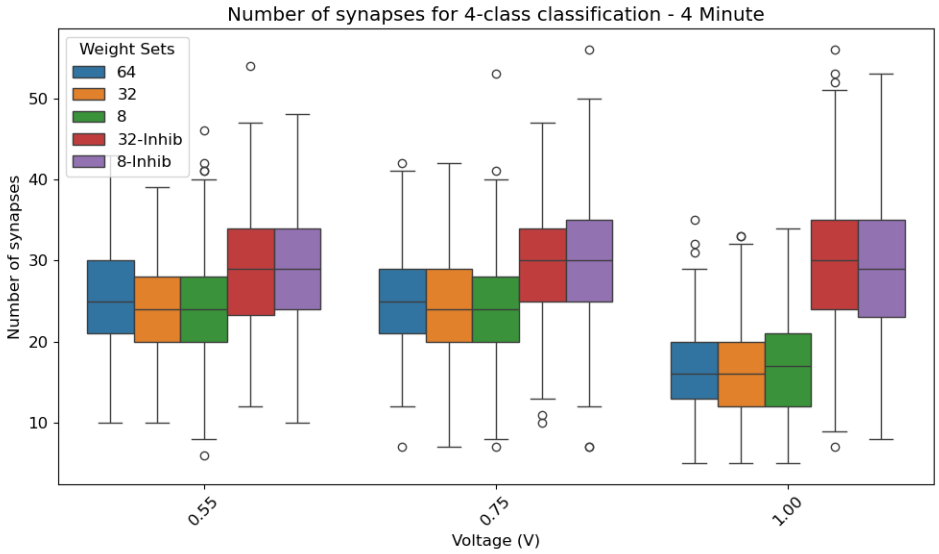


Figure 5.15: Number of synapses for multi-class classification for ferroelectric weight sets in 4 minute time windows

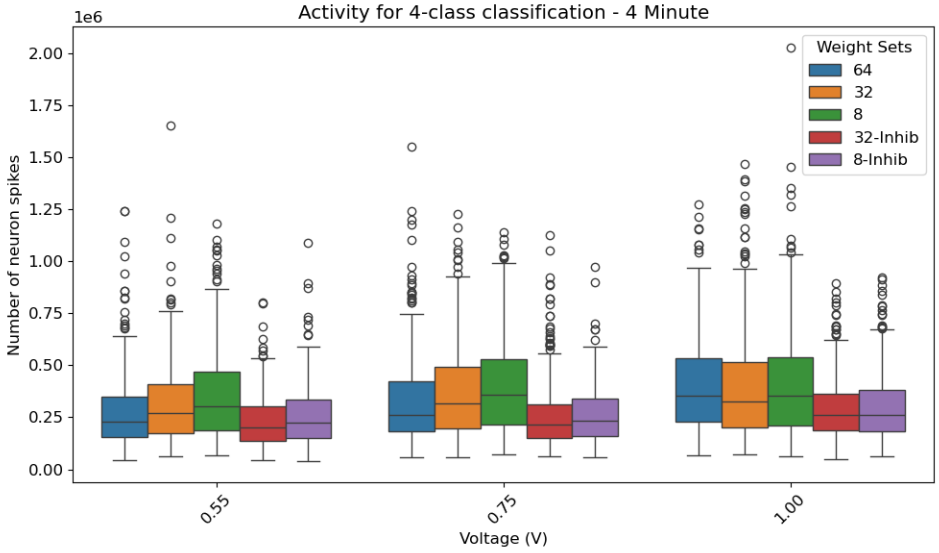


Figure 5.16: Activity for multi-class classification for ferroelectric weight sets in 4 minute time windows

the case of non-inhibitory weight sets. Fig. 5.16 shows that 1.0 V produced slightly more active networks than the other read voltages.

Group Classification

In addition to classifying dosage level, we attempted to classify mouse obesity group. Due to the absence of female mice in the DIO group, we only classified male mice. In contrast to the dosage classification task, longer time windows achieved the best performance, particularly the 10 and 8 minute windows, at around 48% accuracy, shown in Fig. 5.17. The spike encoder yielded significantly better performance than the val encoder, and the count decoder outperformed the TTLS decoder. We compared our results to standard scikit-learn methods as a baseline, again finding that our networks performed better on average. The scikit-learn MLPClassifier achieved roughly 26% testing accuracy for both the 2 and 4 minute time windows.

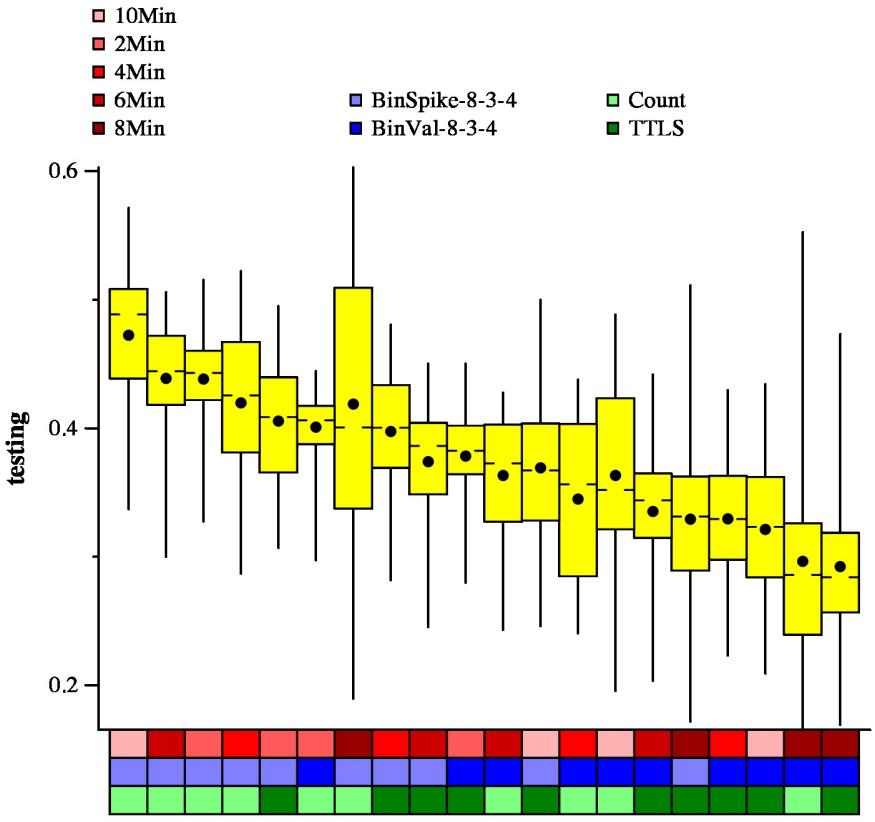


Figure 5.17: Group classification fitness for various encoders and decoders

Sex Classification

The final classification task we attempted was sex classification. Due to the absence of female mice in the DIO group, we only classified mice in the B6, db, and ob groups. Similar to the group classification, the 10 minute time window performed the best, achieving around 54% accuracy, shown in Fig. 5.18. However, the 2 and 4 minute windows consistently achieved the second best performance, indicating that the 10 minute window may be an outlier. In contrast to the dosage and group classification tasks, the val encoder produced better networks than the spike encoder, as measured by accuracy. The count decoder achieved better results than the TTLS decoder. We compared our results to standard scikit-learn methods as a baseline and found that our networks performed better on average. The scikit-learn MLPClassifier achieved roughly 49% and 51% testing accuracy for the 2 and 4 minute time windows, respectively. Table 5.2 summarizes the best performing networks for the scikit-learn and neuromorphic implementations for each time window and each task.

Table 5.2: Highest accuracy rate for neuromorphic and scikit-learn implementations for each task for 2 and 4 minute time windows

	SNN - 2	**SNN - 4**	**ANN - 2**	**ANN - 4**
Binary dosage	66	68	52	56
Multi dosage	30	30	26	28
Group	45	43	26	26
Sex	54	54	49	51

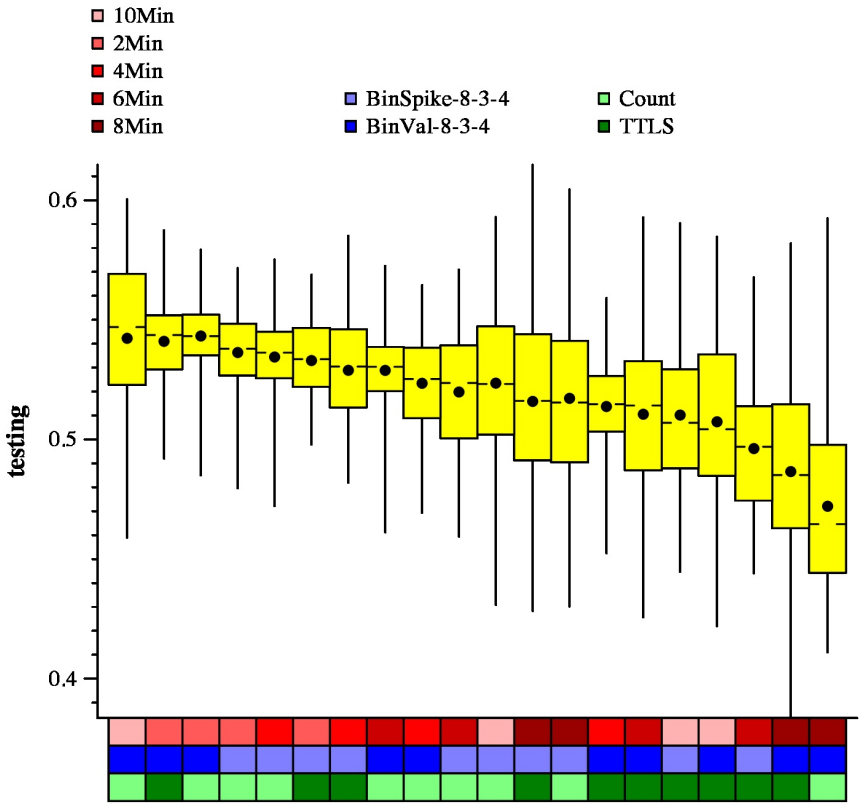


Figure 5.18: Sex classification fitness for various encoders and decoders

5.4 Discussion and Conclusion

In the present study we leveraged neuromorphic computing to enrich our understanding of the impact of buprenorphine on respiration as it relates to sex, obesity, and leptin status. We compared our neuromorphic model to standard machine learning methods, finding that our model performed better on average than standard implementations for the mouse dosage, group, and sex classification tasks. We found that incorporating the temporal component of our dataset yielded the best performance. Our results show that 4 minutes appears to be an appropriate amount of time to evaluate the data to determine how much buprenorphine was administered a mouse. Additionally, we see that it is possible to predict a mouse's sex by learning from its minute ventilation data. Lastly, we show that we are able to understand which group a mouse belongs to from its minute ventilation data.

We also evaluated the use of hafnia-based ferroelectric devices as synapses as applied to our mouse minute ventilation classification task. We compared our results against a basic neuromorphic implementation. The ferroelectric networks achieved comparable performance to basic neuromorphic implementations, indicating that these devices are well-suited to neuromorphic computation. We showed that small networks with low activity are able to perform well. This is significant in the context of evaluation on real hardware with limited resources. We found that the usage of different read voltages overall did not have a large impact on network performance, but 1.0 V tends to perform slightly better than the other read voltages for the multi-class task. However, changing the read voltage did have an impact on the number of synapses, with 1.0 V producing smaller networks, especially for the non-inhibitory weight sets. Read voltage also impacted activity, with 1.0 V producing networks with greater activity.

In future work, we plan to investigate more limited ferroelectric device weight sets. We also plan to experiment with different encoding schemes. We also plan to

experiment with exclusion of the first few minutes of recorded breathing time in an attempt to reduce error and potentially improve performance.

Chapter 6

Conclusions

Neuromorphic computing is growing in popularity due to its potential for energy efficiency and adaptability to hardware constraints such as size and power. However, there is a disconnect between device and applications research. Devices and material researchers often do not now know how best to use their novel neuromorphic implementations. Applications researchers often do not know how to formulate problems they want to solve in a way that is digestible to a neuromorphic computer. Neuromorphic co-design seeks to bridge this gap by linking applications and hardware to explore their interactions.

The two co-design studies presented in this work have evaluated an emerging type of hardware, ferroelectric devices, for neuromorphic computing, as well as showcased neuromorphic computing's innate qualities that make it particularly useful for certain types of tasks, such as event-driven applications. We have applied ferroelectric devices to a control application as well an an interesting classification task that highlights the relevancy of neuromorphic techniques to temporal data.

Chapter 1 introduces SNNs, emerging hardware and applications for neuromorphic implementation, and neuromorphic co-design. It highlights the qualities of SNNs that make it well-suited to temporal data and pattern recognition tasks, namely its spike-based processing mechanism. It explains the benefits of neuromorphic computing,

such as energy efficiency, and explains why neuromorphic computing is attractive for both ferroelectric devices and medical data. Chapter 2 discusses related research in evolutionary approaches and neuromorphic co-design. This background of relevant work in the field provides context for how the work in this study is related to other research, yet provides unique insight. Chapter 3 provides an overview of the tools and algorithms used in this study. The TENNLab framework is the primary resource used in this work, connecting the EONS algorithm, the RISP neuroprocessor, encoding and decoding techniques, and applications to produce train and test SNNs.

Chapter 4 evaluates ferroelectric devices for use as synapses in neuromorphic systems to solve a control task with various difficulty levels. We apply the weights with and without noise and study their impact on network fitness, size, and activity. We found that ferroelectric weights performed as well as, or better than basic neuromorphic implementations, and therefore ferroelectric devices are well-suited to neuromorphic computation. Additionally, the best performing read voltage is task dependent but can be optimized to produce well-performing networks. We found that networks can be optimized to perform well even in noisy conditions. The ferroelectric networks were generally small and had low activity, which is promising for evaluation on hardware with limitations on size, power, and weight. Chapter 5 investigates the ability of neuromorphic computing to understand the effects of buprenorphine on respiration as it relates to sex, obesity, and leptin status, using minute ventilation data collected from mice. We evaluate a basic neuromorphic implementation as well as applying a variety of limited ferroelectric weight sets to this dataset. Our neuromorphic models outperformed scikit-learn models, likely due to their incorporation of the temporal component of the data. We found 4 minutes to be an appropriate amount of time to distinguish between buprenorphine dosages, and we were able to successfully train networks to discern a mouse's sex and group from its minute ventilation.

In the future, we plan to explore the interplay between the neuron model and the ferroelectric synapses. We also plan to experiment with using more limited

ferroelectric weight sets with lower precision memory, and use newly simulated ferroelectric weight data. We also plan to evaluate the use of ferroelectric weights on other datasets, such as long range radio frequency fingerprint identification data and seismic data.

Bibliography

Aboumerhi, K., Güemes, A., Liu, H., Tenore, F., and Etienne-Cummings, R. (2023). Neuromorphic applications in medicine. *Journal of Neural Engineering*, 20(4):041004. 27

Angel, C., Glovak, Z. T., Alami, W., Mihalko, S., Price, J., Jiang, Y., Baghdoyan, H. A., and Lydic, R. (2018). Buprenorphine depresses respiratory variability in obese mice with altered leptin signaling. *Anesthesiology*, 128:984–991. 27

Blouw, P., Choo, X., Hunsberger, E., and Eliasmith, C. (2019). Benchmarking keyword spotting efficiency on neuromorphic hardware. 1

Casimiro, I., Stull, N. D., Tersey, S. A., and Mirmira, R. G. (2021). Phenotypic sexual dimorphism in response to dietary fat manipulation in c57bl/6j mice. *Journal of Diabetes and its Complications*, 35:107795. 30

Deo, R. C. (2015). Machine learning in medicine. *Circulation*, 132(20):1920–1930. 26

Feng, L., Shan, H., Fan, Z., Zhang, Y., Yang, L., and Zhu, Z. (2023). Towards neuromorphic brain-computer interfaces: Model and circuit co-design of the spiking eegnet. *Microelectronics Journal*, 134:105808. 6

Gaidhane, A. D., Dangi, R., Sahay, S., Verma, A., and Chauhan, Y. S. (2022). A computationally efficient compact model for ferroelectric switching with asymmetric non-periodic input signals. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 11, 32

Glovak, Z. T., Angel, C., O'Brien, C. B., Baghdoyan, H. A., and Lydic, R. (2022). Buprenorphine differentially alters breathing among four congenic mouse lines as a function of dose, sex, and leptin status. *Respiratory Physiology Neurobiology*, 297:103834. 27

Hutchinson, S., Keiter, E., Hoekstra, R., Watts, H., Waters, A., Russo, T., Schells, R., Wix, S., and Bogdan, C. (2002). The xyce™ parallel electronic simulator–an overview. *Parallel Computing: Advances and Current Issues*, pages 165–172. 11, 32

Jerry, M., Chen, P.-Y., Zhang, J., Sharma, P., Ni, K., Yu, S., and Datta, S. (2017). Ferroelectric fet analog synapse for acceleration of deep neural network training. In *2017 IEEE international electron devices meeting (IEDM)*, pages 6–2. IEEE. 11, 32

Jin, Y., Wen, R., and Sendhoff, B. (2007). Evolutionary multi-objective optimization of spiking neural networks. In de Sá, J. M., Alexandre, L. A., Duch, W., and Mandic, D., editors, *Artificial Neural Networks – ICANN 2007*, volume 4668 of *Lecture Notes in Computer Science*, pages 370–379. Springer, Berlin, Heidelberg. 4

Léonard, F., Backer, A. S., Fuller, E. J., Teeter, C., and Vineyard, C. M. (2021). Co-design of free-space metasurface optical neuromorphic classifiers for high performance. *ACS Photonics*, 8(7):2103–2111. 5

López-Vázquez, G., Ornelas-Rodriguez, M., Espinal, A., Soria-Alcaraz, J. A., Rojas-Domínguez, A., Puga-Soberanes, H. J., Carpio, J. M., and Rostro-Gonzalez, H. (2019). Evolutionary spiking neural networks for solving supervised classification problems. *Complexity*, 2019:1–11. First published: 28 March 2019. 5

Mandal, S., El-Amin, A., Alexander, K., et al. (2014). Novel synaptic memory device for neuromorphic computing. *Scientific Reports*, 4:5333. 2

Najmaei, S., Glasmann, A. L., Schroeder, M. A., Sarney, W. L., Chin, M. L., and Potrepka, D. M. (2022). Advancements in materials, devices, and integration schemes for a new generation of neuromorphic computers. *Materials Today*, 59:80–106. 2, 10, 28

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830. 35

Plank, J. S., Rizzo, C., Shahat, K., Bruer, G., Dixon, T., Goin, M., Zhao, G., Anantharaj, J., Schuman, C. D., Dean, M. E., Rose, G. S., Cady, N. C., and Van Nostrand, J. (2019). The TENNLab suite of LIDAR-based control applications for recurrent, spiking, neuromorphic systems. In *44th Annual GOMACTech Conference*, Albuquerque. 14

Plank, J. S., Schuman, C. D., Bruer, G., Dean, M. E., and Rose, G. S. (2018). The TENNLab exploratory neuromorphic computing framework. *IEEE Letters of the Computer Society*, 1(2):17–20. 7, 29

Plank, J. S., Zheng, C., Gullett, B., Skuda, N., Rizzo, C., Schuman, C. D., and Rose, G. S. (2022). The case for RISP: A reduced instruction spiking processor. *arXiv preprint arXiv:2206.14016*. 8, 12, 31

Salam, S. S. and Rafi, R. (2023). Deep learning approach for sleep apnea detection using single lead ecg: Comparative analysis between cnn and snn. In *2023 26th International Conference on Computer and Information Technology (ICCIT)*, pages 1–6. 29

Sarney, W. L., Glasmann, A. L., Pearson, J. S., McGinn, C. K., Litwin, P. M., Bisht, R. S., Ramanathan, S., McDonnell, S. J., Hacker, C. A., and Najmaei, S. (2023).

Monolithic integration and ferroelectric phase evolution of hafnium zirconium oxide in 2d neuromorphic synaptic devices. *Materials Today Nano*, 24:100378. 11, 32

Schroeder, U., Park, M. H., Mikolajick, T., and Hwang, C. S. (2022). The fundamentals and applications of ferroelectric hfo2. *Nature Reviews Materials*, 7(8):653–669. 2, 10, 28

Schuman, C. D., Mitchell, J. P., Patton, R. M., Potok, T. E., and Plank, J. S. (2020a). Evolutionary optimization for neuromorphic systems. In *NICE: Neuro-Inspired Computational Elements Workshop*. 7

Schuman, C. D., Mitchell, J. P., Patton, R. M., Potok, T. E., and Plank, J. S. (2020b). Evolutionary optimization for neuromorphic systems. In *Proceedings of the 2020 Annual Neuro-Inspired Computational Elements Workshop*, pages 1–9. 33

Schuman, C. D., Plank, J. S., Disney, A., and Reynolds, J. (2016). An evolutionary optimization framework for neural networks and neuromorphic architectures. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 145–154. IEEE. 33

Schuman, C. D., Potok, T. E., Patton, R. M., Birdwell, J. D., Dean, M. E., Rose, G. S., and Plank, J. S. (2017). A survey of neuromorphic computing and neural networks in hardware. 2

Steed, J., Boussard, S., Glasmann, A., Najmaei, S., Plank, J. S., and Schuman, C. D. (2024). Evaluation of ferroelectric devices for neuromorphic applications. In *IEEE International Conference on Rebooting Computing*. 9

T, P., O, H., K, A., H, I., MKZ, R., SK, I., and ASM, M. (2022). Ecg and spo2 signal-based real-time sleep apnea detection using feed-forward artificial neural network. *AMIA Joint Summits on Translational Science Proceedings*, 2022:379–385. 29

Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., and Maida, A. (2019). Deep learning in spiking neural networks. *Neural Networks*, 111:47–63. 1

Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., Cola, G. D., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Tan, H., and Younis, O. G. (2024). Gymnasium: A standard interface for reinforcement learning environments. 14

Tyagi, P. K. and Agrawal, D. (2024). Automatic detection of sleep apnea from a single-lead ecg signal based on spiking neural network model. *Computers in Biology and Medicine*, 179:108877. 28

Upadhyay, N. K., Jiang, H., Wang, Z., Asapu, S., Xia, Q., and Yang, J. J. (2019). Emerging memory devices for neuromorphic computing. *Advanced Materials Technologies*, 4:1800589. 2

Vineyard, C. M., Severa, W. M., and Aimone, J. B. (2023). Neuromorphic co-design as a game. 5

Vázquez, R. and Garro, B. (2011). Training spiking neurons by means of particle swarm optimization. In Tan, Y., Shi, Y., Chai, Y., and Wang, G., editors, *Advances in Swarm Intelligence. ICSI 2011*, volume 6728 of *Lecture Notes in Computer Science*, pages 228–237. Springer, Berlin, Heidelberg. 5

Wan, Q., Sharbati, M. T., Erickson, J. R., Du, Y., and Xiong, F. (2019). Title of the article. *Advanced Materials Technologies*, 4:1900037. 2

Zhu, J., Zhang, T., Yang, Y., and Huang, R. (2020). A comprehensive review on emerging artificial neuromorphic devices. *Applied Physics Reviews*, 7(1). 2, 10, 28

Vita

Julia Anne Steed was born in Hattiesburg, Mississippi as the first of two children to Chad and Jessica Steed. Julia showed an interest in mathematics, art, and music from an early age. In 2010 their family, now including her younger brother, Blake, moved to Oak Ridge, Tennessee. She attended Oak Ridge High School, where she played the cello in the orchestra. She completed an internship at Oak Ridge National Laboratory under Dr. Catherine Schuman in June 2019, and graduated in May 2020.

Julia chose to attend the University of Tennessee Knoxville and completed many internships throughout her undergraduate career. She worked as a software quality assurance intern for Acato Information Management, LLC in Oak Ridge during her first few summer and winter breaks. She joined TENNLab as an undergraduate research assistant in February 2022. The summer of her sophomore year, she worked as a data analytics intern at U.S. Bank in Knoxville. She traveled to Kansas City during the summer of her junior year, working as an embedded software engineering intern for Garmin International, Inc. During her senior year, she was accepted into the five-year MS/BS program for Computer Science at the University of Tennessee Knoxville. Julia graduated Summa Cum Laude with a Bachelor's degree in Computer Science in May 2024. She then returned to Kansas to complete another internship with Garmin as a High Level Software Engineering Intern. In August 2024, she began her final year of school, serving as a research assistant under Dr. Schuman, focusing on leveraging neuromorphic computing systems for real-world applications.