

On the Reliability of Neuromorphic, Event-Based Systems for Space

by

Seth Roffe

Master of Science, University of Pittsburgh, 2020

Bachelor of Philosophy, University of Pittsburgh, 2017

Submitted to the Graduate Faculty of
the Swanson School of Engineering in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Pittsburgh

2022

UNIVERSITY OF PITTSBURGH
SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Seth Roffe

It was defended on

October 19, 2022

and approved by

Ryad Benosman, PhD, Professor, Department of Electrical and Computer Engineering

Rajkumar Kubendran, PhD, Assistant Professor, Department of Electrical and Computer
Engineering

Zhi-Hong Mao, PhD, Professor, Department of Electrical and Computer Engineering

Vladimir Savinov, PhD, Professor, Department of Physics and Astronomy, Dietrich School
of Arts and Sciences

Dissertation Director: Alan George, PhD, Mickle Chair Professor, Department of Electrical
and Computer Engineering

Copyright © by Seth Roffe
2022

On the Reliability of Neuromorphic, Event-Based Systems for Space

Seth Roffe, PhD

University of Pittsburgh, 2022

Neuromorphic, event-driven systems can be separated into two main sections: neuromorphic vision and neuromorphic processing. Both are remarkably efficient methods that aim to offer a new archetype of computing. The shared concept between the two is to process or sense in the temporal domain. Event-based vision sensors replicate biological retinas to make use of their high power efficiency, sparse output representation, and large dynamic range. Similarly, neuromorphic processors are modelled after the human brain to simulate how neurons fire and learn. This computational model improves power efficiency, enables native machine-learning capabilities, and overcomes the von Neumann memory bottleneck.

This research designs, creates, and evaluates a full system for reliable sensor processing within a neuromorphic classification system from end to end. This evaluation involves ensuring that the failure modes and reliability of a neuromorphic system are known at every step from sensor data, to processing data, to output data. The matrix-multiplication kernel was chosen as a common algorithm needed for ML/CV applications and evaluated for its reliability and efficiency under different dependable-computing techniques. Given the results from this evaluation, a neuromorphic vision sensor was chosen for further study due to its promise in low-power ML/CV capabilities and low data rate. This research provides the first radiation test data to observe and model the effects induced by radiation. The Event-Based Radiation-Induced-Noise Simulation Environment (Event-RINSE) is proposed as a fault injector to simulate the modeled neutron effects on event data without the need for radiation testing. Finally, a neuromorphic classification method, the Hierarchy of Event-Based Time-Surfaces (HOTS) is studied for use in a radiative environment such as space to build off of the previous two experiments. Specifically, how the Time Surface features and other common neuromorphic computations such as time delays respond to radiation noise, and how upsets affect classification accuracy, are evaluated. Given these results, methods to create a more reliable neuromorphic architecture for use in hazardous environments are

proposed. Each section provides a piece of a complete neuromorphic classification system. This research provides a starting point to realizing a reliable, fully neuromorphic sensing and processing system for future spacecraft.

Table of Contents

1.0 Introduction	1
2.0 Dependable-Computing Techniques for ML/CV	4
2.1 Background	5
2.1.1 Fault Injection	6
2.1.2 Neutron Beam Testing	7
2.1.3 Reliability Techniques	7
2.1.4 Algorithm-Based Fault Tolerance	8
2.1.5 Machine Learning and Computer Vision in Spacecraft	10
2.1.6 Computational Complexity	11
2.2 Platform Selection	12
2.3 Experimentation	12
2.4 Methodology	12
2.4.1 Fault Injection	12
2.4.2 Irradiation	13
2.4.3 Employed ABFT Configuration	15
2.4.4 Statistical Analysis	16
2.5 Experimental Results	17
2.5.1 High-Level Fault Injection	18
2.5.2 Low-Level Fault Injection	21
2.5.3 Irradiation	24
2.5.4 Statistical Significance	30
2.6 Conclusions and Future Work	31
3.0 Radiation Effects on Neuromorphic, Event-Based Vision Sensors	35
3.1 Background	37
3.1.1 Neuromorphic Event-Driven Visual Sensors	37
3.1.2 Conventional Space Situational Awareness	39

3.1.3	Event-based Sensors for Space Situational Awareness	40
3.1.4	Neutron-Beam Testing	41
3.2	Methodology	42
3.2.1	Event-Camera	42
3.2.2	Irradiation	42
3.2.3	Data Collection and Analysis	45
3.3	Results	47
3.3.1	Induced-Event Rate	47
3.3.2	Angle of Incidence Comparison	48
3.3.3	Effects of Room Brightness	51
3.3.4	Signal-to-Noise	51
3.3.5	Noise Patterns	59
3.4	Circuit-Level Interaction Interpretation	61
3.5	Event-RINSE Simulator	64
3.5.1	Simulated Noise Generation	65
3.5.2	Pattern Validation	67
3.5.3	Simulation Environment Usage	67
3.6	Conclusion	70
4.0	Reliability of Temporal Classification on Neuromorphic Processing . .	73
4.1	Background	74
4.1.1	Neuromorphic Event-Based Sensors	74
4.1.2	HOTS algorithm	75
4.1.3	Fault Injection	76
4.2	Related Work	77
4.2.1	Traditional Feature Selection	77
4.2.2	Conventional SDA	78
4.2.3	Neuromorphic Sensors for SDA	78
4.2.4	Neuromorphic Sensors Under Radiation	79
4.3	Scope and Methodology	80
4.3.1	Scope	80

4.3.2 Methodology	80
4.3.2.1 Radiation Experiment	80
4.3.2.2 Fault Injection Experiment	81
4.3.3 Platform Selection	82
4.4 Results	82
4.4.1 LANSCE Experiment	83
4.4.2 Fault Injection	84
4.5 Conclusion	85
5.0 Conclusions and Future Work	94
Bibliography	97

List of Tables

1	Fault injection reliability and runtime summary	23
2	Beam test results summary	30
3	Primary comparisons from beam test	31
4	Summary of Event-RINSE runtime options	69

List of Figures

1	ABFT Experimental setup	14
2	Neural network output difference vs. number of fault injections	18
3	Neural network execution time vs. number of injections	19
4	Fault-injection results with no mitigation	21
5	Fault injection results with ABFT	22
6	Neutron beam dosimetry	24
7	Radiation test results with no mitigation	25
8	Radiation test results with ABFT	26
9	Number of ABFT corrections and detections	27
10	Radiation test absolute runtime for matrix multiplication	28
11	Radiation test relative runtime for matrix multiplication	29
12	Event-based sensor operating principles	37
13	Neuromorphic sensor radiation test	43
14	Average induced events per second	44
15	Probability density of induced events	46
16	Angles of incidence event-rate comparison	48
17	Light and Dark room event-rate comparison	49
18	Signal-noise ratio	50
19	Pendulum frequency with and without noise	52
20	Pearson correlation for signal-to-noise ratio	53
21	Optical flow of pendulum	54
22	Clustered noise patterns	56
23	Radiation noise pattern details	57
24	Noise rate macro- and micro-pulses	58
25	Possible neutron decays	60
26	Real and simulated noise line segment projections	62

27	Event-RINSE noise model	64
28	HOTS architecture with fault injection targets	89
29	Irradiated PYNQ boards running HOTS	90
30	Corrupted vs Golden Time Surfaces	91
31	Accuracy of HOTS vs Noise Rate	92
32	Relative accuracy for different fault injection targets	93

1.0 Introduction

As technology improves, spacecraft designers are constantly searching for new sensors and computers to use in their systems. However, onboard processing still continues to be outpaced by the computational demands from modern missions. Missions often introduce stringent limitations in size, weight, power, and cost (SWaP-C), while still requiring complex algorithms to function. Similarly, the ability to downlink the massive datasets needed to run compute-intensive applications on the ground becomes infeasible due to limitations in ground communications, meaning onboard processing becomes vital to mission success.

The recent advancements in machine-learning and computer-vision (ML/CV) algorithms present novel opportunities for improving sensor processing, data analysis, and autonomy. To achieve the computational capability required to reasonably run these applications, the power-efficiency and computation capabilities of commercial-off-the-shelf (COTS) devices become appealing, as opposed to their antiquated and power-hungry radiation-hardened counterparts typically used in space missions. However, the space environment is harsh, introducing new challenges such as ionizing radiation, which can cause faults in the form of single-event-effects (SEEs) in commercial electronics. Moreover, COTS devices are even more vulnerable to radiation effects than traditional radiation-hardened devices [81]. Therefore, it is vital for designers to evaluate the reliability of novel technologies before deployment.

Neuromorphic, event-based systems are perfectly suited to overcome the described challenges in space computing. Event-based systems can be divided into two key paradigms: neuromorphic vision and neuromorphic processing. Both paradigms aim to mimic a biological equivalent. Neuromorphic-vision systems mimic the biological retina to take advantage of their low temporal resolution and motion-tracking abilities. Similarly, neuromorphic processors are modelled after the human brain's neuron-synapse system to match its energy-efficient computing. The shared concept between the two is to collect or process information asynchronously in the time domain as opposed to the spatial domain in conventional vision sensors and image-processing applications. This concept provides a fast, efficient method to realize actionable data at high resolutions while maintaining a low data rate.

Neuromorphic vision sensors detect changes in light intensity at microsecond levels of precision. Timestamped information about the event pixel location and whether the light intensity was increasing (ON) or decreasing (OFF) is sent as an event. Any static, redundant background information is therefore not passed through the sensor, reducing the size of the data being collected and providing a sparse representation of the field of view. Neuromorphic processors, on the other hand, aim to overcome the von Neumann bottleneck where computers are limited by the transfer of data between memory and the processor. Neuromorphic processors use time spikes as a method of encoding information in the temporal response of hardware, providing improved energy efficiency and higher bandwidth due to their event-driven nature. More details behind both neuromorphic vision and neuromorphic processing are covered in Chapters 3 and 4, respectively.

Before neuromorphic sensors and processors are flown, however, their resilience to radiation needs to be tested. Similarly, before any missions use neuromorphic applications, their failure modes and sensitivity to single-event upsets (SEUs), an SEE in the form of a single bit-flip in memory, must be measured and evaluated. With these measurements and noise models, spacecraft designers can test their applications beforehand to ensure that the effects will not impact the science data being gathered, and improve the applications if necessary. The purpose of this research is to perform these initial measurements on neuromorphic, event-based vision sensors and architectures and create the models needed to perform such analyses. Similarly, reliable methods for applications to combat any SEUs are also evaluated for their efficacy in detecting or correcting errors, and reliable architectures are proposed. This research aims to design and evaluate a fully neuromorphic system, ensuring reliable data collection and computation from end to end, providing the necessary information for future missions to create reliable architectures.

This manuscript consists of three chapters. Chapter 2 covers an evaluation of different dependable-computing techniques to provide error correction and detection on a matrix-multiplication kernel, which are the primary type of computation in ML/CV inference applications. Algorithm-based fault tolerance (ABFT) and duplex redundancy within two processor cores were tested as the most common dependability techniques for these kinds of computations, and any performance, reliability, and overhead tradeoffs are discussed. Chap-

ter 3 covers the first neutron irradiation of a neuromorphic-vision sensor to observe the effects of radiation on the data and to model the induced noise profile. The neuromorphic sensor was chosen for its promise in low-power ML/CV capabilities and low data rate. The Event-Based Radiation-Induced Noise Simulation Environment is introduced to simulate the observed and modeled effects in recorded neuromorphic data. Finally, Chapter 4 discusses the reliability of neuromorphic classification algorithms to observe how the use of temporal features in ML apps behave when introduced to radiation noise. The Hierarchy of Time Surfaces (HOTS) architecture is used as a case study to measure the behavior of temporal features features and other common computations used in neuromorphic networks such as exponential decays and distance computations. Methods to create more reliable neuromorphic architectures for use in hazardous environments are then designed and proposed for future missions. Overall, this manuscript provides methods and information to improve the reliability at each step in a neuromorphic sensor-processing application, from sensor data collection, through data processing to the final result.

2.0 Dependable-Computing Techniques for ML/CV

Using COTS electronics in hazardous environments such as space presents a number of difficulties leading to the development of new technologies and strategies. For example, the NSF Center for Space, High-Performance and Resilient Computing (SHREC), located at the University of Pittsburgh, created the CHREC Space Processor (CSP), employing a Xilinx Zynq-7020 SoC with a dual-core ARM Cortex-A9 processor and Artix-7 field-programmable gate array (FPGA), for radiation-tolerant, high-performance, and reconfigurable space computing. The concept behind the CSP is that of hybrid computing, where a high-performance COTS processor is used alongside radiation-tolerant and radiation-hardened components. This concept provides a model that maintains the energy efficiency and computational capabilities of commercial parts, while still remaining capable of withstanding the radiation-heavy space environment. The CSP has been flight proven on three separate missions on the International Space Station, further demonstrating its resilience. Namely, the CSP has successfully been flown on the Space Test Program - Houston 5 (STP-H5) CSP mission, STP-H6 Spacecraft Supercomputing for Image and Video Processing (SSIVP) mission, STP-H7 Configurable and Autonomous Sensor Processing Research (CASPR) mission. [74, 77, 93].

Traditional radiation-hardened electronic systems are often employed to tolerate a high total ionizing dose (TID) and single-event effects (SEEs) from the harsh space environment. However, these processors tend to be generations behind their COTS counterparts in terms of performance, energy-efficiency, and cost. Using commercial devices combined with radiation-hardened components and dependability-enhancing software allows for high-performance required for neuromorphic applications while still remaining a high degree of radiation tolerance. Better performance while still remaining close to the reliability of a radiation-hardened system enables the use of more complex applications in harsh environments such as space.

Using COTS solutions, the capability of on-board processing for these neuromorphic ML/CV applications, such as HOTS, improves. However, COTS electronics are very susceptible to radiation. As a result, the reliability of computationally complex kernels, such as

matrix multiplication for machine learning and computer vision, becomes a concern. Namely, single-event upsets (SEUs) occur when a high-energy particle strikes a microelectronic component and either changes the state of the circuit [81]. These faults can then manifest as transient-data errors, corrupting the output of the application.

Algorithm-based fault tolerance (ABFT) was introduced by Huang and Abraham as a method of information redundancy in matrix operations [32]. Their algorithm focuses on encoding a vector of data using a checksum in order to detect and correct data errors caused by SEUs. Since the dimensionality of the data is reduced on encoding, the memory overhead is relatively insignificant.

In this chapter, the efficacy of ABFT for error mitigation in neuromorphic ML/CV applications was tested by means of fault injection and neutron irradiation with a focus specifically on output-data errors as opposed to control errors or input-data errors. The results demonstrate the use of ABFT to ensure that compute time is not being wasted on corrupted data. This capabilities of this technique are then assessed by examining the reliability and performance impact on processing time.

2.1 Background

This section provides a cursory overview of reliability testing in computing, ABFT, machine learning and computer vision in spacecraft, and computational complexity. This dissertation combines strategies from variously studied fields in order to demonstrate methods to create a more reliable system for applications.

2.1.1 Fault Injection

Fault injection is a methodology that is commonly used to validate the dependability of fault-tolerant systems [98]. Typically fault-injection experiments consist of controlled tests where the behavior of a system in the presence of faults is observed by writing faults directly into the system during operations. Therefore, the reliability of a system can be measured and tested before deployment. Similarly, any vulnerabilities in the system can be found and adjusted through fault injection.

The two main categories of fault injection used in this experiment consist of hardware and software fault injection. Hardware-based fault injection happens at a physical level, where faults are induced by disturbing the hardware from the environment. This disturbance could consist of causing a dip in the voltage of the system's power supply, radiation-beam testing to cause electromagnetic interference in the computing, or even manually modifying the value of specific pins on the device-under-test [98].

Software-based fault injection can happen at a variety of levels of abstraction. Namely, the registers can be affected to induce faults on a low-level. The objective of this kind of testing would be to observe the response of the system for data faults as well as control faults. Similarly, this type of low-level injection gives a more realistic view of how the system would behave in a harsh environment. Conversely, faults can be injected at a high-level in software. This injection would consist of directly changing elements of data in software to observe the response of the tested reliability technique. Injecting directly in the data increases the number of samples by guaranteeing where the injection takes place, but it unfortunately does not give a good indication of how the system would behave in a harsh environment. This high-level fault injection would primarily be used to test the response of a specific reliability technique.

2.1.2 Neutron Beam Testing

Models describing galactic cosmic radiation show that the majority of radiation experienced in orbit would consist of protons and heavy ions [10]. However, neutron beam testing proves to be useful for testing error mitigation strategies with a lower risk of permanently damaging the device under test. Specifically, when testing the response of a mitigation technique to SEUs, the source of said SEU becomes irrelevant.

Beam testing is popular in the field of computing to classify SEUs of new systems and mitigation strategies. Knowledge of radiation data is critical to the design process for space missions by giving an overview of the form of upsets a system might see. For example, Anderson et al. performed a neutron beam experiment on the Xilinx UltraScale+ MPSoC to observe the number system crashes and silent data errors invoked from radiation [7]. With each new device destined for spaceflight, beam testing is vital in determining its viability in space.

Similarly, neutron beam testing is commonly used to test the robustness of systems to SEUs. NASA Langley Research Center and Honeywell used neutron radiation testing to test the robustness of their flight control computer architecture to test flight capabilities in a hazardous environment. In their experiment, they were able to measure the ability of their architecture to recover from neutron-induced faults [36].

The experiment detailed in this chapter follows a structure similar to the latter experiment. Namely, the ability for ABFT to recover faults in a radiation environment was measured to test its viability for applications in space missions.

2.1.3 Reliability Techniques

All reliability techniques center around one main concept: redundancy. In order to detect or correct errors in a fallen system, there must be some form of redundant information that is able to detect when something changes. Redundancy can take many forms and typically fall within one of five categories: hardware redundancy, information redundancy, network redundancy, software redundancy, and time redundancy [39]. The techniques presented in this chapter fall under hardware and information redundancy.

Hardware redundancy involves adding additional hardware to catch any faulty modules. One popular form of hardware redundancy is modular redundancy which involves having multiple modules run the same application and compare the outputs to ensure they match. One common forms of this kind of redundancy is Triple Modular Redundancy which involves using three redundant modules and taking a majority voter to get the output. This type of redundancy enables single error correction since a faulty module will be out-voted by the other two [39]. However, it tends to be expensive since three redundant hardware modules are required. Another form of modular redundancy is duplex redundancy which involves using two redundant modules and restarting the execution if the outputs differ. Duplex redundancy enables single error detection only since there is no method in determining which module has the correct output.

Conversely, information redundancy involves having additional bits of data that holds identifying information about the application. The number of errors that can be corrected or detected depends directly on how many extra pieces of information are stored [39]. This extra information causes the memory overhead to increase as the number of detectable errors increases. Information redundancy tends to be cheaper than hardware redundancy since no additional modules are needed at the cost of being less reliable. ABFT falls under this form of redundancy as it holds encoded data from a matrix in checksum vectors separate from the matrices they represent.

2.1.4 Algorithm-Based Fault Tolerance

To make matrix operations more reliable, ABFT uses extra bits to encode data in order to detect and correct errors. A popular encoding sequence for this method is performed by adding additional checksum vectors, called weighted checksums, to the matrix operands. These vectors containcontaining redundant information about the matrix, such as the sum of the rows or the sum of the columns. These weighted checksums can be expanded to enable detection and correction of more errors within the matrix as long as it contains different types of redundant data. For example, two possible row checksum vectors could be the sum of the columns and the sum of 2^k times the columns where k is the index of the element's

row. If there are n checksum vectors, then either n errors can be detected, but not necessarily corrected, or $\lfloor \frac{n}{2} \rfloor$ errors can be corrected [32]. If an error were to exist in one of the elements of the operand, the information contained in the matrix and the checksums would be different, leading to an error detection. With multiple checksums, the error can be pinpointed in the matrix using the indices of the mismatched checksum vector elements. Then, through subtracting the checksum value from the sum of the row or column while excluding the error value, the corrupted data can be recovered in the original matrix without the need to restart the entire application. There has been much research into different applications and techniques of ABFT. There are many matrix operations that are of interest to the studies of fault-tolerant computing, such as lower-upper (LU) decomposition and gaussian elimination that use similar checksum vectors to detect errors in computation [51]. As mentioned, ABFT is specific to the application under study, so each operation or algorithm needs to be verified that the output preserves the encoding scheme prior to use.

NASA's Jet Propulsion Laboratory (JPL) has investigated the use of ABFT for onboard data analysis in the presence of SEUs. They performed fault injection as a way to inject different amounts of faults into their application and view ABFT's response. With this methodology, JPL is able to simulate a radiative environment by forcing SEUs into their application. As a test case for their ABFT techniques, they used prediction on a support vector machine binary classifier. Their proposed approach used checksums before and after critical matrix-multiplication operations. Testing with 100 different SVMs whose matrix computations were exposed with varying rates of SEU faults, they found that detection improved at higher SEU rates [29].

ABFT is also a popular approach in the high-performance computing (HPC) community for its ability to detect and correct errors without stopping the computation [21]. With increasingly large runtimes from increasingly more complex applications, the mean time to failure (MTTF) sharply declines. Therefore, effects from multiple fault tolerant techniques have been studied. For example, Du et al. employ ABFT methods with checkpointing methods for dense matrix factorizations to significantly increase fault tolerance while keeping the overhead low [24].

Aside from the research into the application side of ABFT, different encoding methods has also been studied to ensure the most efficient use of data redundancy. Anfinson and Luk detail a linear algebra model of the checksum method [8]. This model provides a method of selecting "proper" weights in a weighted checksum. In other words, using different weights defined in the checksum vectors gives methods to enable error correction as well as error detection.

This experiment focuses on evaluating the efficacy of ABFT for matrix multiplication specifically with machine learning and computer vision applications in spacecraft. Past research with ABFT has shown that it is an attractive solution to applications that use matrix operations. In order to use the methodology in the field intelligently, its performance needs to be measured. Due to the algorithm-specific nature of ABFT, the methodology needs to be applied to the application at hand. Therefore, it is much more efficient to test ABFT on kernels upon which many useful applications rely as opposed to testing specific applications in order to view its effects on a broad range of cases.

2.1.5 Machine Learning and Computer Vision in Spacecraft

To enable spacecraft autonomy, researchers have been examining the use of machine-learning algorithms that employ neural networks for sensing experiments. Benediktsson et al. show that neural networks are significantly more accurate at pattern recognition and classification in remote sensing data than traditional statistical methods [13]. However, they also cite that the neural network algorithm used is more computationally intensive than statistical methods, resulting in longer execution times. These long execution times leads to a greater higher susceptibility to radiation effects.

Similarly, computer-vision applications are essential in autonomous mission operations. Ho and McClamroch detail a formulation of automatic spacecraft docking using computer vision [31]. Any application built around image or video data will make heavy use of matrix operations. Therefore, the methodologies detailed are very complex, leading to a long run-time on embedded platforms. Therefore, data reliability is essential to avoid catastrophic failures.

In the realm of space applications, neural-network training would occur on the ground using previously acquired data as the training set. Once trained, the weights would be transmitted to the onboard processor, and only inference would occur onboard for classification purposes. The desire to classify objects in real-time necessitates the need for fast neural-network inference, which relies on both hardware and software that can perform matrix multiplications quickly and resiliently due to the radiative conditions in space.

2.1.6 Computational Complexity

To increase the reliability of matrix-multiplication-based applications in software, multiple methods can be considered. Either the data must have redundancy, or the application must be made faster to avoid vulnerability. Similarly, the probability of a radiation-induced error increases with the size of the data (n) and the amount of time that the data is under computation (t). Execution time of matrix multiplication is dependent on n and expressed with *Big O* notation, describing the worst-case execution time for a given algorithm, $O(f(n))$ for some function f as n becomes sufficiently large.

Stothers' thesis covers the details of the computational complexity behind matrix multiplication. The serial matrix multiplication algorithm is known to have a complexity of $O(n^3)$. Using the recursion methodology shown in Stothers' paper, he was able to reduce the complexity to approximately $O(n^{2.807})$ [83]. However, this improvement is insufficient to significantly reduce the risk involved in matrix multiplication in a radiative environment. This research shows that the complexity of matrix multiplication is bounded from below asymptotically. Therefore, the only viable way to reduce execution time of matrix multiplication is to implement parallelization techniques. However, adding parallel processing also increases the application's vulnerability to radiation due to propagation from potential race conditions over multiple cores. Thus, for large datasets and repeated computations, data redundancies such as ABFT present an attractive solution to the reliability problem.

2.2 Platform Selection

This experiment was tested on the processing system (PS) side of the Digilent PYNQ-Z1 with a Xilinx Zynq 7020 system-on-chip (SoC) to emulate the CSP. The CSP features the same commercial ZC7020 SoC along with radiation-hardened watchdog and power circuitry to ensure the reliability of critical components [93]. While COTS electronics offer more computational power to perform intensive algorithms, they are very susceptible to the radiation environment. Therefore, if any of these algorithms are used on a space processor such as the CSP, fault mitigation techniques are necessary to ensure the accuracy of the application.

2.3 Experimentation

This chapter describes the methodology and results of the experiment. The methodology of this experiment consists of fault-injection and irradiation to test the efficacy of ABFT.

2.4 Methodology

This section explains the methodology of each step of the experiment performed. The experiment can be separated into fault injection and irradiation. Fault injection was performed separately from irradiation as an original test for the capabilities of ABFT and expected results.

2.4.1 Fault Injection

Fault injection was performed at a high-level and a low-level with respect to the application. The high-level injector injected different amounts of errors into the matrix operands of a simple neural network by choosing a random element and changing its value. This high-level injection was used to compare ABFT to another popular reliability method, duplex

redundancy. Duplex redundancy involves running the same application on two cores of the CPU and restarting if they have differing results. The number of faults injected varied to observe the response of the underlying method.

As a way to generalize neuromorphic classification, a handful of possible networks were chosen to act as examples to what would be used for HOTS classification. A neural-network classifier run on the MNIST dataset was used to measure the effects of the SEUs on accuracy and runtime for ABFT, duplex, and without any reliability techniques. 5,500 MNIST images were used for a training set over 15 epochs and 5,000 were used for a validation set. Inference was performed without any injections to obtain a golden, expected output that was used to compare to all injections. The results of these fault injections gave a direct comparison between the responses of ABFT and other reliability techniques to data errors.

Low-level fault injection involved performing bitflips on random registers within the processor. This method gives a response that is more realistic to radiation effects as it is hardware-dependent. Register fault injection was performed prior to radiation testing to simulate possible outcomes of the radiation experiment. These injections were performed on a 500×500 matrix multiplication kernel using the Dynamic robust Single Event Upset simulator (DrSEUs) [20] on the Digilent PYNQ-Z1 to ensure a consistent platform.

Low-level injections were performed at random times during executions on random registers on the ARM Cortex-A9 processor. General-purpose registers were targeted specifically to cause more data errors since other types of faults were not included in the scope of this experiment. Since ABFT is data-specific, it would be ineffective at detecting any kernel-related faults. The results of this fault injection show an idealistic view of the expected results of the radiation experiment.

2.4.2 Irradiation

Three Digilent PYNQ-Z1 boards were used as the device-under-test (DUT) for this experiment. Irradiation was done at the Los Alamos Neutron Science Center (LANSCE) Weapons Neutron Research (WNR) facility using the 4FP30R/ICE-II instrument for neutron irradiation. Despite the space environment being dominantly composed of proton and heavy-ion

radiation, neutron irradiation provides valuable insight on the response of the mitigation strategy to induced errors without causing significant permanent damage to the device. The three boards were irradiated in a line and labeled as one, two, and three. Boards one and three use ABFT-fault mitigation in matrix multiplication, while board two does not use any mitigation techniques. Neutron dosimetry was tracked using a dosimeter to observe the relative neutron flux passing in 10-second intervals. The boards were placed with the Zynq SoC chips aligned to the 1-inch collimated beam, and irradiated simultaneously to ensure a similar neutron fluence on each one, as shown in Figure 1.

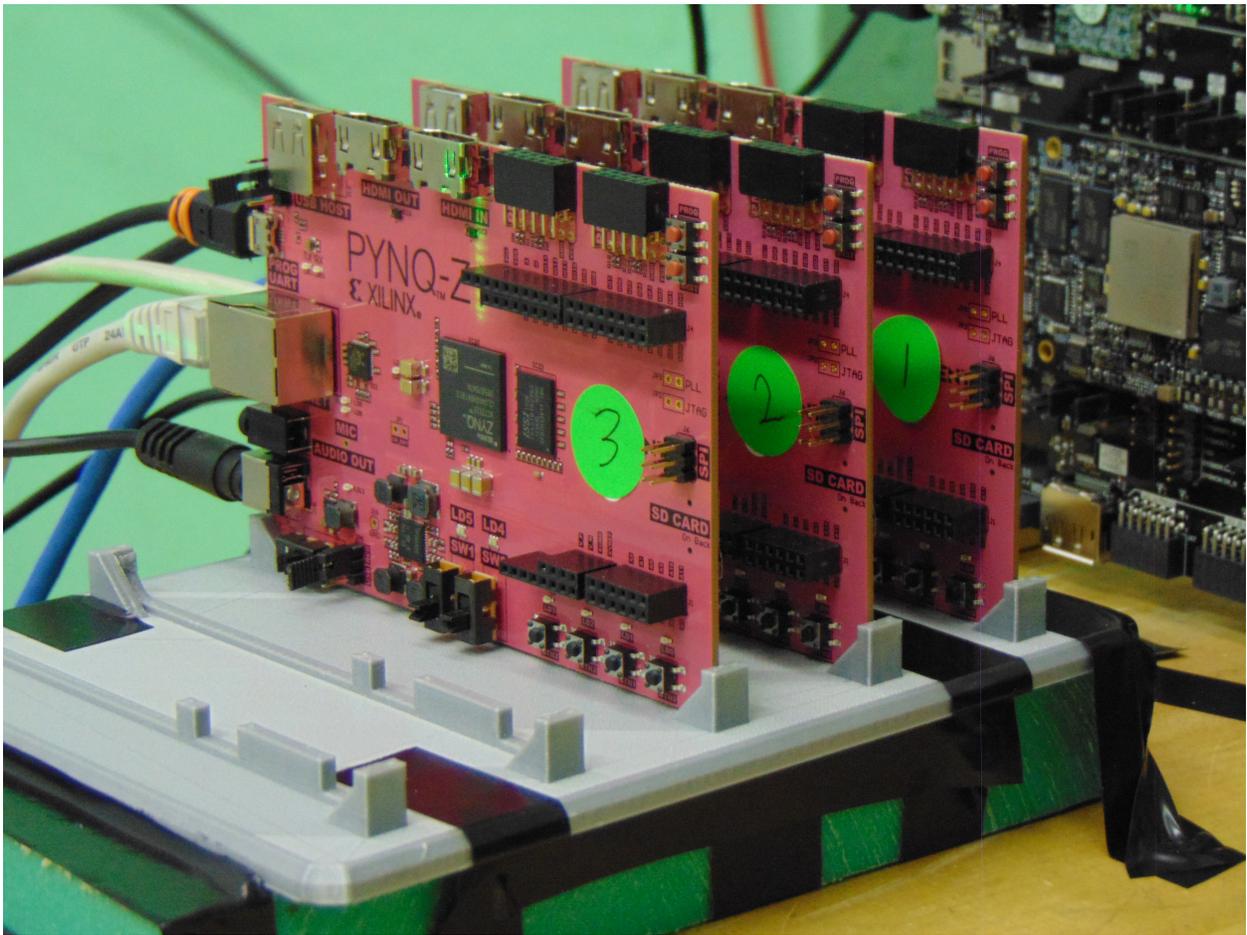


Figure 1: The experimental setup consisting of three PYNQ-Z1 boards irradiated in a line.

Each PYNQ board employed a matrix-multiplication kernel using the serial matrix-multiplication algorithm. Two 500×500 matrices of 16-bit integers were multiplied together to represent a large multiplication that would take place in a machine-learning or computer-vision algorithm, such as a neural network. These added checksum vectors give $2 \times 500 = 1000$ bytes memory overhead for each input matrix and $2 \times 2 \times 500 = 2000$ bytes of memory overhead for the output, totaling at 4000 bytes. This added memory is less than one percent of the memory needed to hold the matrices, so ABFT can be considered to have a negligible memory overhead. Boards one and three used ABFT mitigation techniques, while board two did not have any form of fault mitigation. This ordering was chosen to prevent any significant bias from specific boards being closer to the beam.

2.4.3 Employed ABFT Configuration

ABFT was chosen to enable single-error correction (SEC) for each of the three separate experiments, as recommended by Al-Yamani et. al since it shows an efficient correction-to-overhead ratio for matrix multiplication [5]. One checksum was generated for each input matrix: a row checksum representing the sum of the columns of the first and a column checksum representing the sum of the rows of the second. Multiplication of these two matrices and checksums results in an output matrix containing two checksums enabling single-error correction and double-error detection (SEC/DED). Similarly, this methodology allows for single-error detection (SED) between reading the data and the calculation in the initial matrices that are being multiplied. In order to keep the checksums separate from the data, each checksum was kept in a separate array data structure.

Before multiplying, the input matrices are compared with their checksums, and are read in again if there were any discrepancies. The focus of this experiment was on errors encountered during processing. External input/output errors were not considered due to the fact that external errors are sensor-specific and cannot be addressed with ABFT. After any multiplication, the checksum vectors are also multiplied by the matrix they are not associated with to calculate the two checksums for the output. The output matrix is then checked with these checksums and corrected if needed by cross referencing the location index

of the error by which elements of the checksums disagree with the output. The element is then restored by subtracting the sum of the rest of the row or column from the checksum element. More than one error cannot be corrected due to SEC being the assumed scope of the redundancy technique. The generalized matrix configurations for $A \times B = C$ with ABFT supporting SEC/DED in the output is used with A, B , and C defined as in Equation 2-1.

$$\begin{aligned}
 A &= \begin{bmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{bmatrix} \\
 &\quad \left[\begin{array}{ccc} \sum_{i=1}^m A_{i,1} & \cdots & \sum_{i=1}^m A_{i,k} \end{array} \right] \\
 B &= \begin{bmatrix} B_{1,1} & \cdots & B_{1,k} & \sum_{i=1}^k B_{1,i} \\ \vdots & \ddots & \vdots & \vdots \\ B_{n,1} & \cdots & B_{n,k} & \sum_{i=1}^k B_{n,i} \end{bmatrix} \\
 \Rightarrow C &= \begin{bmatrix} C_{1,1} & \cdots & C_{1,k} & \sum_{i=1}^k C_{1,i} \\ \vdots & \ddots & \vdots & \vdots \\ C_{m,1} & \cdots & C_{m,k} & \sum_{i=1}^k C_{m,i} \end{bmatrix} \\
 &\quad \left[\begin{array}{ccc} \sum_{i=1}^m C_{i,1} & \cdots & \sum_{i=1}^m C_{i,k} \end{array} \right]
 \end{aligned} \tag{2-1}$$

2.4.4 Statistical Analysis

To evaluate statistical significance, various tests were performed on the final results. In order to ensure a good distribution of observed errors, a sample size of about 16,000 events was taken. The cross-section, calculated as the number of data errors divided by the neutron fluence, was used to calculate the reliability of ABFT as an error mitigation strategy. To evaluate the efficacy of adding ABFT to the operation, the mean-work-to-failure (MWTF) was calculated. Finally, to ensure that the results were statistically significant, a z-test was calculated between the distribution using ABFT and the distribution without error mitigation.

2.5 Experimental Results

The results of this experiment have been split into two separate events: fault injection and irradiation. Fault injection was performed to give an idealistic view of the results prior to irradiation. In this way, the two scenarios can be examined separately and then compared after analysis.

2.5.1 High-Level Fault Injection

High-level fault injection was performed to compare ABFT to duplex redundancy in regard to the accuracy and runtime of a neural network. The number of mispredictions was directly compared to the number of data faults that were injected.

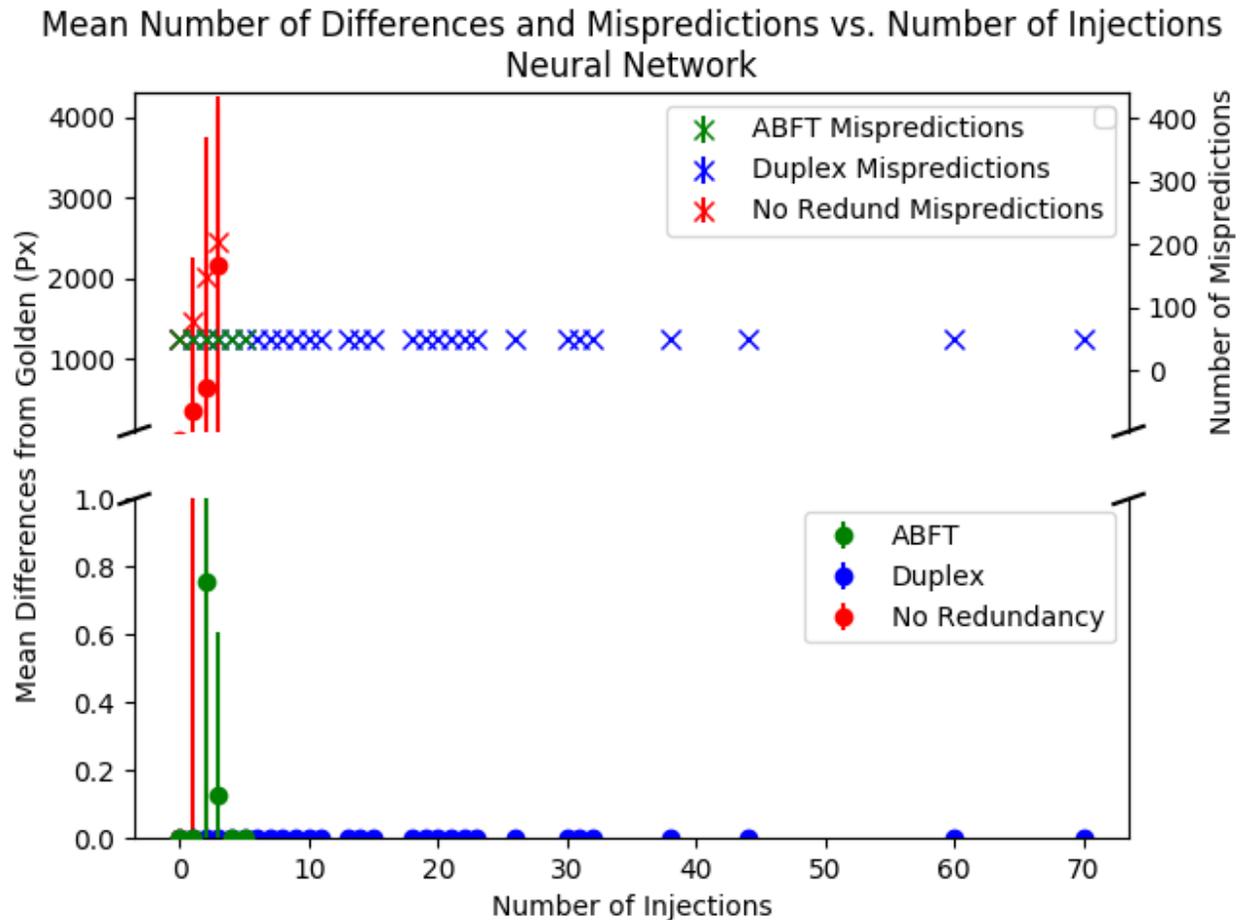


Figure 2: Neural network number of different elements in the output matrix differences and number of mispredictions vs. number of fault injections.

Figure 2 shows the mean number of mispredictions and the mean number of differences compared with the golden matrix containing classification probabilities with respect to the number of injections. In the figure, x symbols represent the average misprediction with a scale on the right side of the plot, while dots represent discrepancies with the golden output

matrix classification probabilities with a scale on the left side of the plot. Under normal operations with no injected faults, the neural network exhibited 51 mispredictions, or a 94.9% test accuracy. When faults were injected into the network, the classification accuracy degraded rapidly. For one, two, and three injections, the classification accuracy falls to 92.0%, 85.0%, and 79.8%, respectively. Adding ABFT and duplex redundancy brings the classification accuracy back to 94.9% as in golden operations. Duplex shows more injections overall since faults may be injected each time it restarts execution. Duplex did not show any difference with the golden output matrix. Conversely, ABFT showed slight discrepancies with the golden output matrix with more than one injection, as expected by the chosen design of ABFT. However, these discrepancies did not manifest as increased mispredictions.

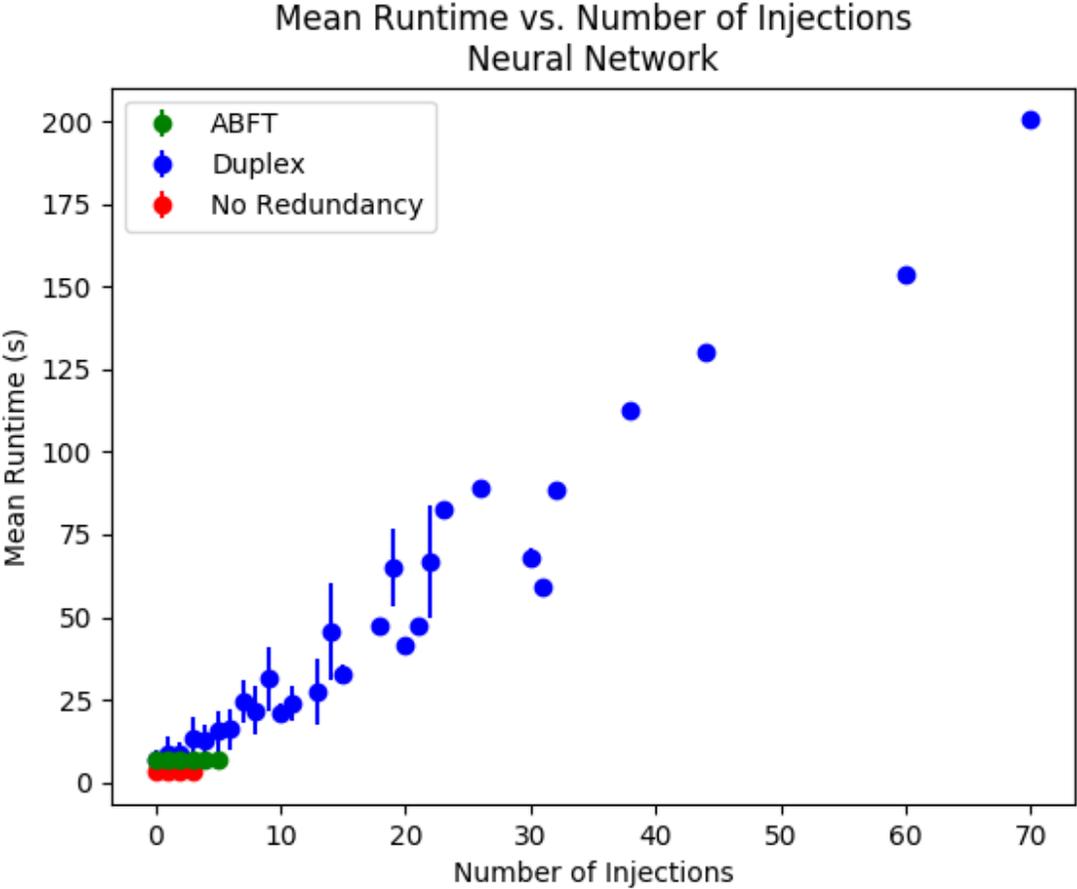


Figure 3: Neural network mean execution time vs. the number of injections.

Figure 3 shows the mean execution time for the neural network using all three methods. The runtime of the application with no redundancy technique stayed consistent around 3.04 seconds with a standard deviation on the order of 10^{-3} for any number of injections. Similarly, ABFT showed little variation in the runtime with an average around 6.62 seconds and a standard deviation on the order of 10^{-3} for any number of injections. The consistency in runtime is due to the ability to recover from errors without restarting executions. Meanwhile, duplex runtime showed a linear relationship with the number of injections performed. Therefore, while duplex showed the best reliability, it performed the worst in execution time since it needed to restart with every error.

2.5.2 Low-Level Fault Injection

Before irradiation, the reliability was examined with fault injection to compare the expected effects of data-error mitigation with ABFT. Errors seen in fault injection were categorized into “Injector Error” where the failure was internal to the fault injector, “Execution Error” where the fault manifested as a segmentation fault or system crash, and “Data Error” where the output did not match the expected golden output. This experiment focuses specifically on data errors.

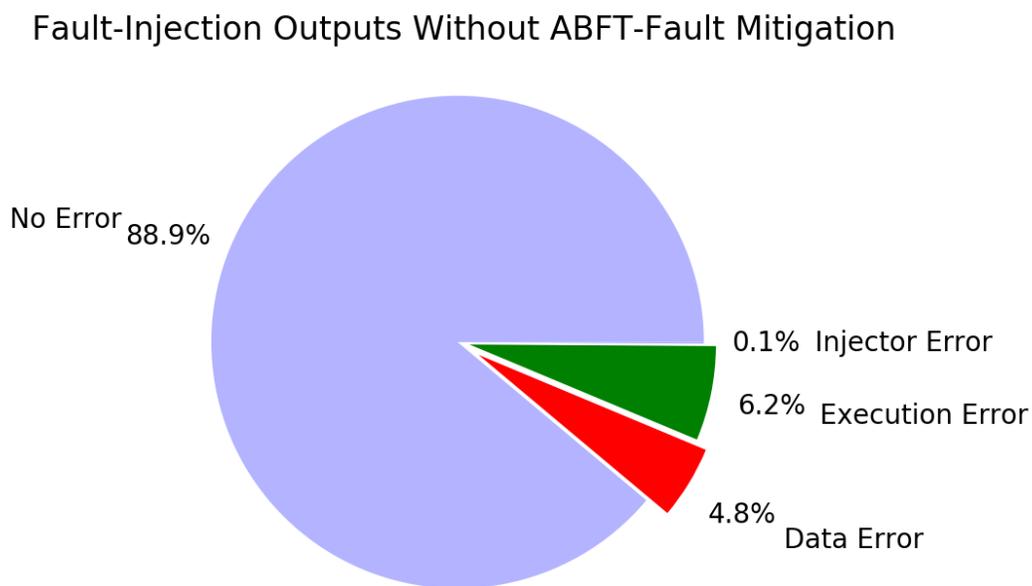


Figure 4: Fault-injection outputs from DrSEUs without any form of error mitigation.

Both forms were run and injected 1000 times. Without any mitigation, 4.8% of executions had outputs that differed from the expected matrix value giving a data error, as seen in Figure 4. The number of data errors are increased compared to what would be seen in a radiation experiment due to injecting specifically into the GPR. However, 6.2% of runs presented execution errors of the form of hangs or segmentation faults.

Fault-Injection Outputs With ABFT-Fault Mitigation

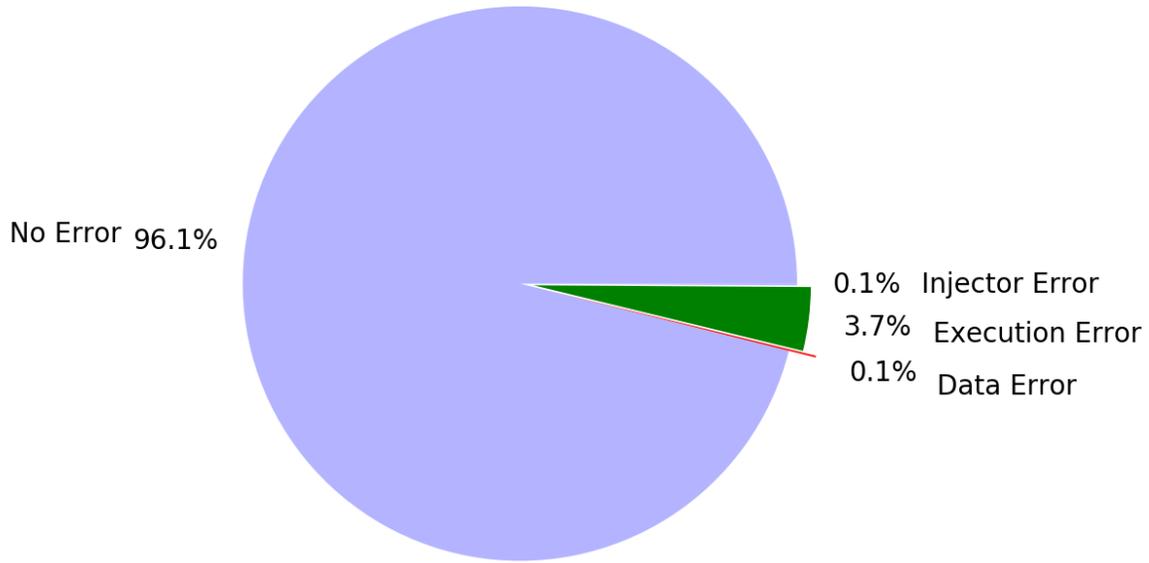


Figure 5: Fault-injection outputs from DrSEUs with ABFT.

Comparatively, only 0.1% of executions show data-output errors for the same number of injections, shown in Figure 5. Therefore, a $48\times$ improvement in reliability was observed using ABFT for data-error mitigation in fault injections. However, since these injections were targeted towards the general-purpose registers to force data errors instead of control errors, the improvement is inflated from what would be expected in irradiation. Any reduction in execution errors is an effect of having a lower sample size since ABFT has no bearing on control errors. As the number of runs for each injection experiment grows, the number of execution errors would approach roughly the same value.

Table 1: Summary of Fault-Injection Reliability and Runtime Results for 1000 Executions.

Design	Data Errors [%]	Runtime [s]
No Mitigation	4.8	15.19 ± 1.92
With ABFT	0.1	15.58 ± 1.39

A summary of the fault-injection results described above can be seen in Table 1. The two designs show no significant difference in runtime at about 15 seconds showing the minimal impact ABFT has on time overhead.

2.5.3 Irradiation

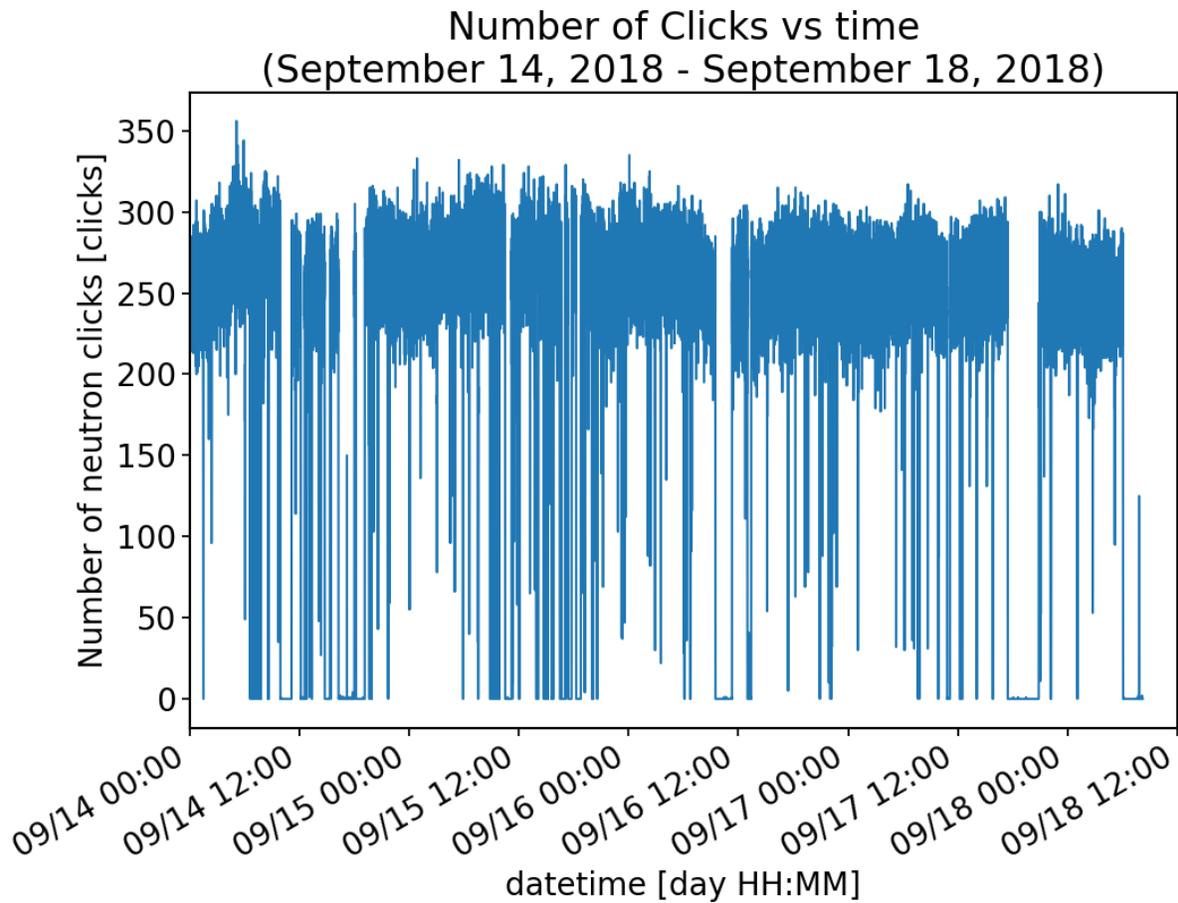


Figure 6: Neutron beam dosimetry over time over the days of irradiation: September 14, 2018 - September 17, 2018.

The measured dosimetry over the four days of experimentation provided by the dosimeter can be seen in Figure 6. Neutron beam dosimetry over the days of irradiation, September 14 to September 17, 2018, were represented by the number of clicks processed by the dosimeter. The number of clicks is multiplied by a constant for each day to represent an estimate of the number of neutrons of energies greater than 1 MeV and 10 MeV passing through the dosimeter.

This dosimetry gives a measurement of the number of neutrons passing through the dosimeter within a time interval of 10 seconds giving an estimate of the cross-section of neutrons passing through the device. The dosimetry was only considered when the devices under test were powered to calculate the cross-section

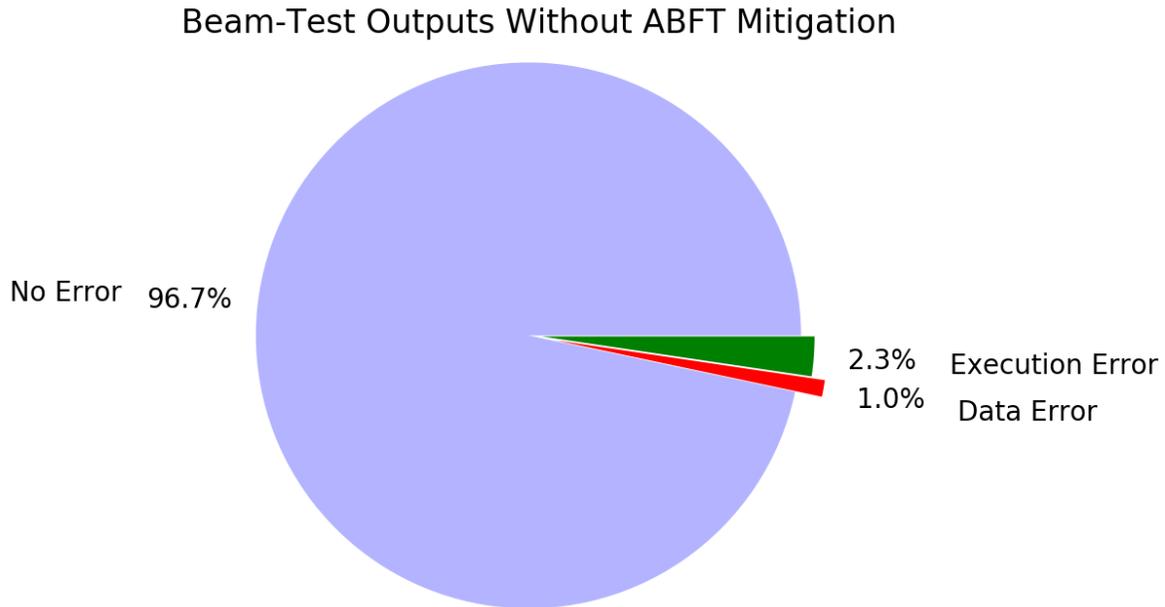


Figure 7: Results from four days of irradiation running a 500×500 matrix multiplication kernel without error mitigation.

The overall reliability in the output data was compared between the matrix multiplications employing SEU mitigation with ABFT and without under the wide-spectrum neutron radiation. Segmentation faults, system crashes, memory dumps etc. were classified as execution errors, and faults resulting in an out different from the expected was classified as a data error. Without any form of error mitigation, the matrix multiplication kernel show 1% of data errors over four days of irradiation. The ratio of unmitigated results can be seen in Figure 7.

Beam-Test Outputs With ABFT Mitigation

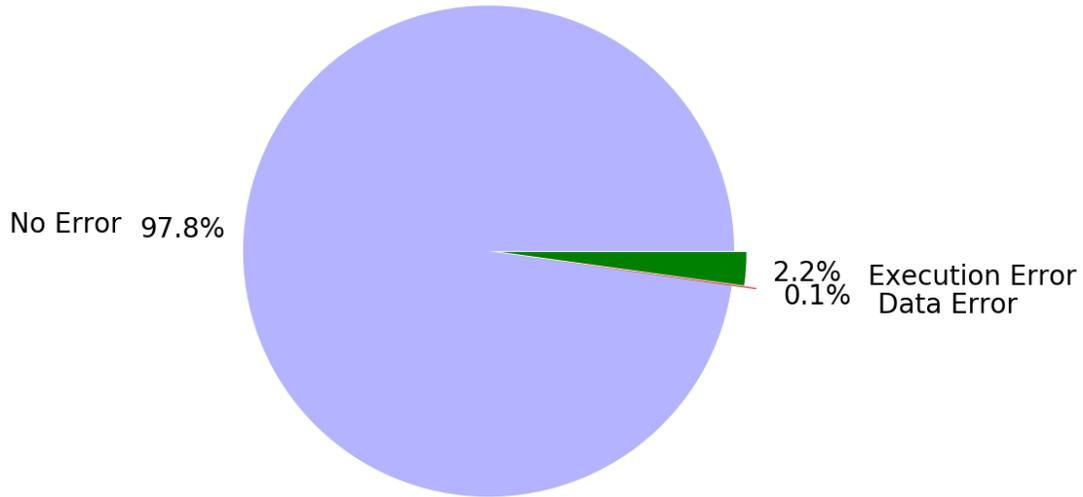


Figure 8: Results from four days of irradiation running a 500×500 matrix multiplication kernel while using ABFT for data-error mitigation.

Meanwhile, using ABFT for error mitigation shows only 0.1% data errors. Dividing this percentage by the number of data errors seen without mitigation gives a $10\times$ improvement in reliability. It can be seen that in both cases the number of execution errors remains the same at around 2.2% as expected. The beam test results with ABFT mitigation can be seen in Figure 8.

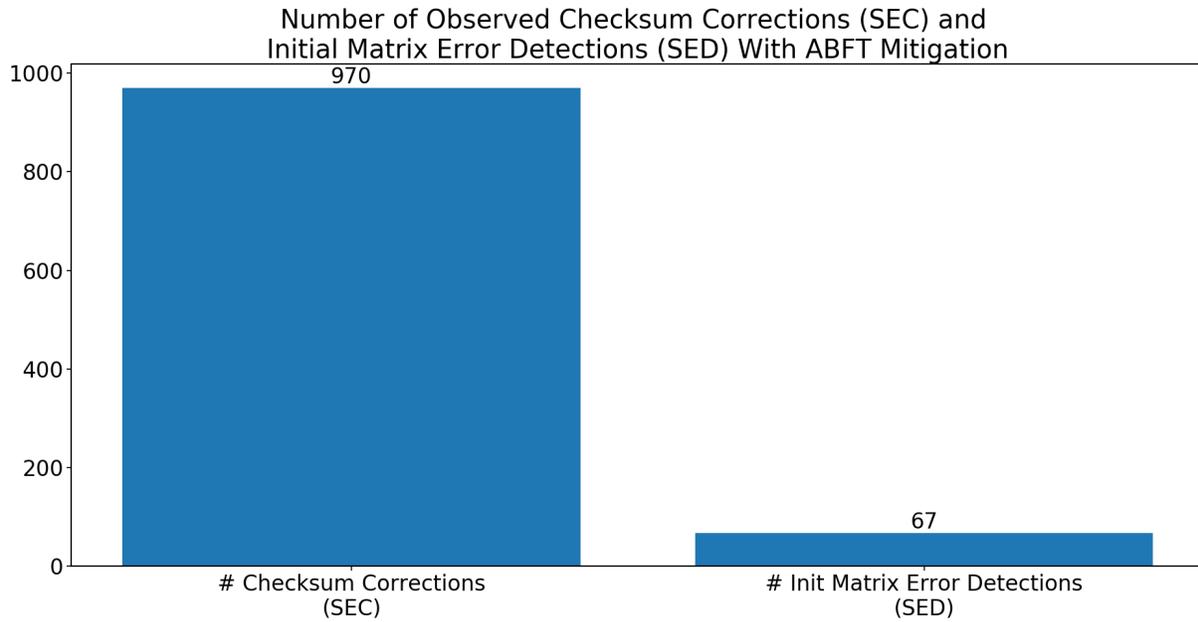


Figure 9: Number of errors corrected in the output matrix (SEC) and errors detected in the input matrices (SED).

To ensure that the reliability technique was, in fact, correcting errors, the number of errors caught and corrected in the output (SEC) were measured. Additionally, the number of errors detected in the input matrices (SED), restarting the calculation, was also measured. These values can be seen in Figure 9, showing a significant number of errors observed and corrected.

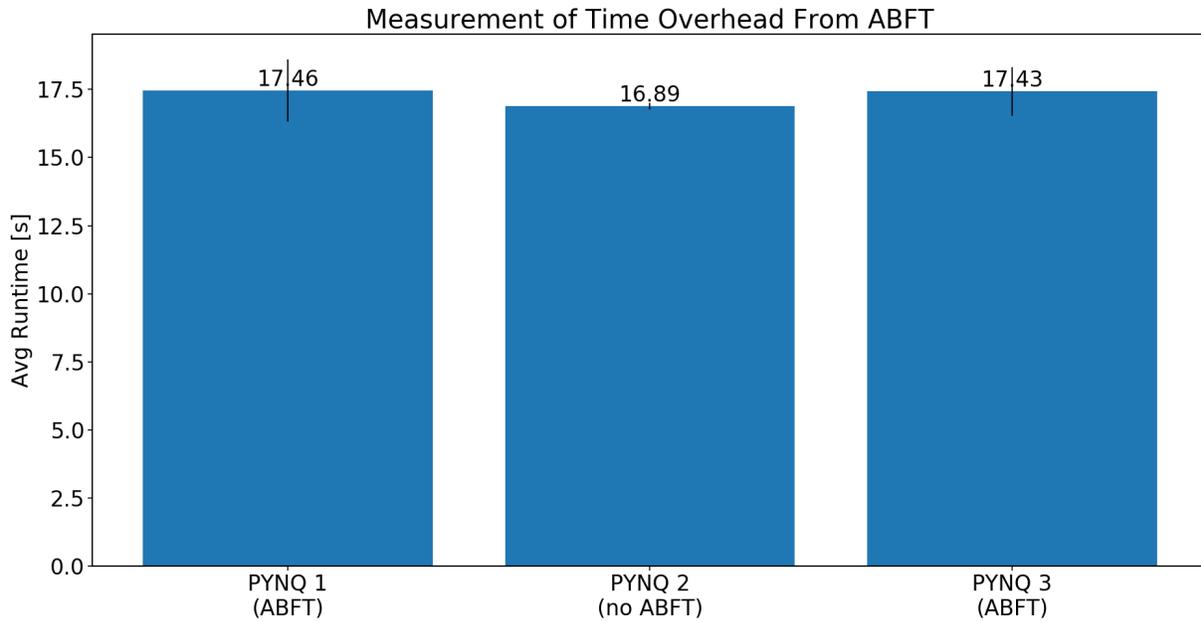


Figure 10: Average runtime of 500×500 matrix multiplication on the processor side of the Xilinx Zynq 7020 for both tested configurations.

In order to show overhead, the runtime of each execution was measured. Adding ABFT to the kernel did not result in a significant time increase. The average execution time increased from 16.89 seconds without mitigation to 17.43 seconds as seen in Figure 10, leading to a 3% increase in execution time.

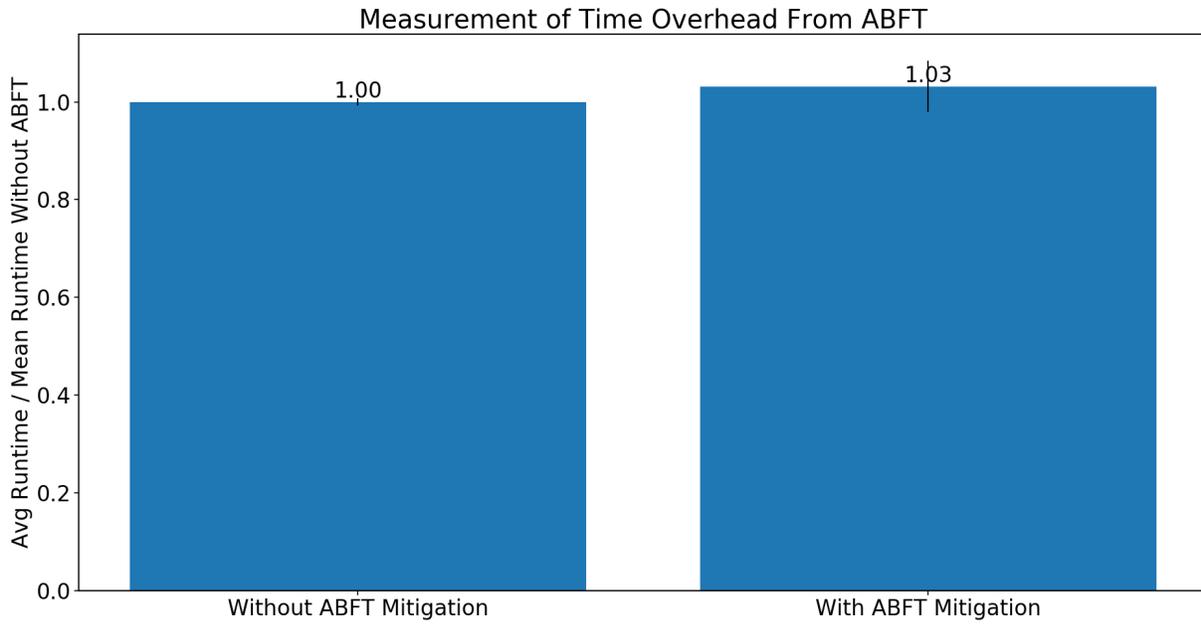


Figure 11: Execution times relative to the average runtime without any fault mitigation.

This increase in execution time is mostly due to the input matrices need to be reread if an error is detected in the inputs, and the calculation time for the checksum vectors. The relative increase in time can be seen in Figure 11.

Table 2: Summary of Beam-Test Reliability and Runtime Results Over Four Days of Irradiation.

Design	Data Errors [%]	Runtime [s]
No Mitigation	1.0	16.89 ± 0.11
With ABFT	0.1	17.43 ± 0.89

A summary of the beam test results can be seen in Table 2. These results agree with the results of fault injection and show a $10\times$ reduction in data errors with minimal time-overhead.

2.5.4 Statistical Significance

The cross-section was calculated by dividing the number of observed data errors by the calculated neutron fluence seen in Figure 6. To account for when the boards were powered off, only the effective fluence was calculated. Quantitative comparisons show a $5.33\times$ reduction in cross-section when ABFT is applied for fault-mitigation. In other words, there are about 5 times fewer errors per neutron that passes through the device-under-test. Therefore, adding ABFT as a fault-mitigation strategy will increase reliability in the data of a system by more than 5 times with a confidence level greater than 99% ($p < 10^{-5}$).

Similarly, the mean work to failure (MWTF) with fault mitigation was about 15.05 times that without any mitigation. This improvement shows that about 15 times more work can be done before a failure occurs than a system without any mitigation. Calculating a z-test between the two populations shows a confidence level of 99% with a p-value $< 10^{-5}$.

Table 3: Summary of Primary Comparisons Between Matrix Multiplication With ABFT for Fault Mitigation and Without Error Mitigation

Design	Eff. Fluence [n^0]	# Events	Exec. Errors	Data Errors	Cross-Section [cm^2]	Mean Work to Failure	95% Conf. Interval
No Mitigation	1.25×10^{11}	16649	389	163	1.308×10^{-9}	1668	$0.01 \pm 1.17 \times 10^{-5}$
With ABFT	4.49×10^{10}	16209	353	11	2.452×10^{-10}	25107	$0.001 \pm 3.82 \times 10^{-6}$
Improve					$5.33 \times$	$15.05 \times$	

The 95% confidence interval is calculated to be $0.01 \pm 1.17 \times 10^{-5}$ without mitigation and $0.001 \pm 3.82 \times 10^{-6}$ with ABFT. Since the true means do not overlap within the confidence interval, a statistically significant improvement in reliability is observed with high confidence. A summary of the comparison results can be seen in Table 3.

There was no significant difference in the number of execution errors between the two designs since execution errors were not addressed under the scope of this experiment. ABFT does not protect against segmentation faults, nor is it capable of detecting kernel errors. Therefore, the methodology presented in this chapter has no bearing on problems with the execution of the application under test or control of the system.

2.6 Conclusions and Future Work

The purpose of this experiment was to evaluate the use of algorithm-based fault tolerance to mitigate data errors in the matrix-multiplication kernels of machine-learning and computer-vision applications. The results show that ABFT is a viable solution to increase reliability on applications using a matrix-multiplication kernel.

The complexity of matrix multiplication has been studied exhaustively and show no

improvement in runtime. Therefore, the only way of improving reliability of matrix multiplication applications is by adding data redundancy. Using ABFT to correct errors in the output matrix and detect errors in the input matrices show a reduction of data errors by $10\times$ under radiation. There were still a small amount of data errors in the mitigated design most likely due to upsets consisting of two or more bit-flips or damage to the board.

High-level fault injection was performed to compare ABFT's performance to duplex redundancy for matrix multiplication in a neural network. Both ABFT and duplex showed no increase in mispredictions as compared to the control with no redundancy, although ABFT did begin to exhibit matrix errors in the presence of multiple fault injections. The runtime for the ABFT enabled network was almost twice that of the base algorithm, but did not increase with an increasing number of faults. This increase in time overhead is inversely proportional to the size of the matrix multiplication kernel, as shown by the radiation experiment. On the other hand, duplex showed a linear increase in runtime with an increasing number of injections, culminating in a runtime that can be orders of magnitude larger than that of the control and the ABFT multiplications. As a result, ABFT demonstrates the best performance in terms of speed and accuracy in a radiation environment for a neural network application.

Fault-injection results show a more idealized effect of fault mitigation on the number of data-errors due the general-purpose registers being the target of injection. The fault-injection results agree with the beam-test results demonstrating consistency in the ability of ABFT to improve a system's reliability. Significant discrepancies in the fault-injection and radiation results are due to the randomized nature of SEUs in a radiation-beam experiment and the ideal nature of fault injection into general-purpose registers. This lack of controlled variables in radiation testing causes control errors that were not considered in the scope of this experiment, reducing the overall percentage of observed data errors.

Irradiation results show a statistically significant improvement in reliability in terms of the cross-section and the MWTF with a negligible increase in runtime. Therefore, ABFT has been shown to improve the resilience of data in a radiative environment. Similarly, significantly more work can be done before an error is occurred, effectively increasing the number of applications that can be run before encountering a failure. This increase can thus

help improve the autonomy of missions.

In both fault injection and irradiation, the time overhead of ABFT is shown to be minimal, averaging at about a 3% increase in execution time. This overhead is mostly due to calculation of the checksum vectors and when a problem was found in the input matrices with single-error detection. With the increase in reliability, ABFT is seen to have a significant improvement in dependability with minimal overhead.

Execution errors were shown to be more prevalent than data errors even without mitigation. However, due to the methodology ABFT is built on, execution errors were not addressed in the scope of this experiment. Data errors were specifically targeted for detection and correction in this experiment.

In general, data errors were not shown to be commonly occurring, accounting for less than 1% of all executions without mitigation. However, due to the number of decisions spacecraft can make using machine-learning or computer-vision applications, this small percentage can become non-negligible with time, especially if a single error can cause catastrophic failure. Therefore, employing ABFT as data redundancy has been shown to be a viable solution to significantly increase the amount of time and work done before failure while introducing minimal overhead.

Moving forward with error mitigation in complex applications requires two main steps. First, a new method of reducing errors in the execution of the application must be developed and addressed. ABFT is data specific, and does nothing to prevent control failures. With kernel protection, it is more likely that the application will complete. Then, combining that protection with ABFT, the system can guarantee with reasonable certainty that the application-under-test will produce not only a result, but an accurate one free from corruption.

The second step in future uses of ABFT is to test the reliability of commonly-used applications in spacecraft and with data from novel sensors. Many computer-vision and machine-learning applications make use of the matrix multiplication kernel as their main computation. An example of one such application is the support-vector machine (SVM) used for image classification. Testing how ABFT affects the correctness of an SVM, or other classification algorithms would show ABFT's efficacy in ensuring accurate predictions and

classifications in autonomous missions. If accurate predictions and classifications can be made, then proper and intelligent decisions can be assured in future spacecraft. Similarly, the benefit of using ABFT can be assessed when running applications parallelized with multiple cores. With the ability to run parallel processing reliably in radiation environment, the complexity of applications that missions are capable of increases even more.

3.0 Radiation Effects on Neuromorphic, Event-Based Vision Sensors

Neuromorphic, event-based cameras are remarkably efficient, robust, and capable of operating over a large range of light intensities. These sensors mimic the design of biological retinas to make full use of their high power efficiency, sparse output, large dynamic range, real-time computation, and low data rate. Neuromorphic sensors are built by copying aspects of their biological counterparts, and are therefore massively parallel and highly non-redundant [65]. Each pixel of the sensor works independently, sensing changes in light and providing output in the form of discrete events, signifying increasing or decreasing light intensity.

Event-based cameras are a perfectly suited solution to space missions where the resource budget is limited due to the reduced average bandwidth between the sensor and host needed to collect data. These sensors have the potential to improve numerous space applications, including those involved in space domain awareness, target tracking, Earth observation, and astronomical data collection [58]. Due to the harsh conditions entailed, however, the performance of such sensors in space is yet to be explored. The scope of this work is to test the resilience of neuromorphic sensors to neutrons impacting the sensor in a highly radiative environment. The goal is to determine the failure modes of the neuromorphic camera as seen under the same spectrum as that produced by cosmic rays and to measure the possible impact of neutrons on the temporal precision of output events, noise levels, and computation.

Although studies have been carried out into the behavior of various optoelectronic devices under neutron radiation [6, 25, 33, 40, 89, 90], no work to date has addressed the radiation-tolerance aspects of event-based visual sensors to analyze if this technology is capable of retaining its efficacy under radiative conditions. To observe and evaluate SEEs, we irradiated a neuromorphic event-based sensor at Los Alamos National Lab's (LANL) ICE-II neutron facility.

The measured neutron energy distribution at LANL-ICE-II is significantly more intense than the flux of cosmic-ray-induced neutrons, and this allows for testing at greatly accelerated rates. An ICE-II radiation test is about 5×10^7 more intense than neutrons from cosmic

radiation [85]. Neutrons are known to interact with the materials in the semiconductor and produce daughter particles, which may deposit or remove charge in sensitive volumes of the chip. If the deposited charge is significant enough, it can change the state of a bit in the system. In a digital system, this change of state is known as a bit-flip. Sensors include analog circuitry, and therefore produce more complex behavior than simple bit-flips. Beam testing is popular in sensor processing to classify SEEs in new computing systems and test the robustness of systems to SEUs. Different systems may respond in different ways to the radiation that brings about SEEs, producing faults and errors of varying degrees. The effect of SEEs can range from negligible, where an unused area of memory is affected, to single-event latch-ups that could damage the system permanently.

Knowing how a system may respond to radiation is vital to the success of a space mission in that it provides an overview of the kind of upsets that may arise. This information allows designers to plan for any problems that may be encountered in flight. SEUs are transient in that they do not permanently damage the device, but they may cause some silent data or control errors which, if uncaught, may lead to a loss of performance or accuracy. To reduce risk, it is therefore vital to know how a new system will respond to radiation before deployment.

In this chapter, we measured the effect of radiation and categorized the SEEs observed in the sensor. We also tested how radiation affects pure event-based computation in the context of optical-flow estimation, which is known to be sensitive to noise and temporal imprecision, under both radiation and non-radiation conditions. Finally, we also used this preliminary data to develop a simulator that makes it possible to inject events with radiation-noise effects into any data stream. We call this simulator the “Event-based Radiation-Induced Noise Simulation Environment (Event-RINSE).” Event-RINSE allows realistic neutron-beaming effects to be added to any event-based data sequence. These simulated radiation effects enable designers to test developed algorithms prior to mission deployment.

3.1 Background

This section gives an overview of the neuromorphic event-driven visual sensor, its data acquisition principles, and its data types. The use of event-driven sensors for space applications is also discussed.

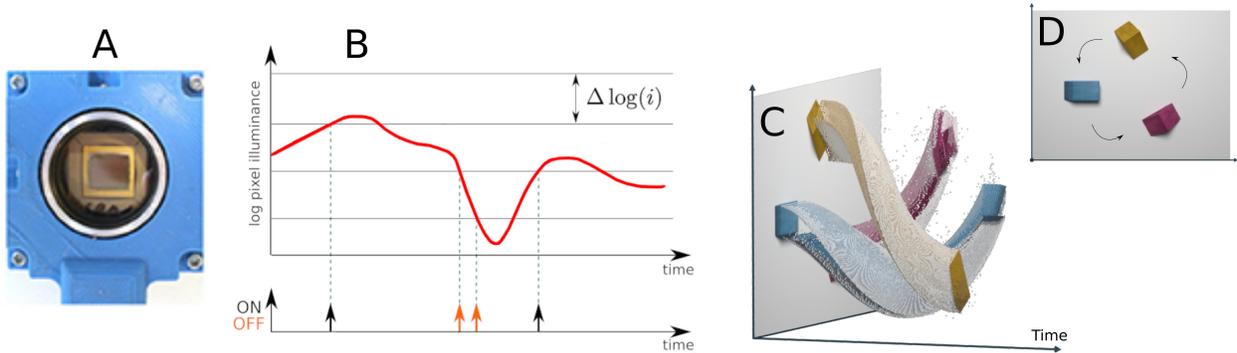


Figure 12: Event-based sensor operating principles: (A) The event-based sensor used in this experiment. (B) When a given pixel’s luminosity change reaches a given threshold, it produces a visual event with an x and y address, a timestamp, and a polarity which is either ON or OFF depending on the change in relative luminosity. (C,D) The stream of events generated by three rotating shapes, shown here in a color version of the sensor’s absolute light measurement output that comes with every event.

3.1.1 Neuromorphic Event-Driven Visual Sensors

Biomimetic, event-based cameras [47] are a novel type of vision sensors that, like their biological counterparts, are made of independent cells/pixels which are driven by events taking place in their field of view, generating an asynchronous stream of spikes/events. This method of data collection is in contrast to conventional vision sensors which are driven by artificially created timing and control signals (frame clock) to create full images that have no relation to either the content or the temporal dynamics of the visual scene. Over the past few years, several types of these event-based cameras have been designed. These

include temporal contrast vision sensors sensitive to change in relative luminance, gradient-based sensors sensitive to static edges, devices sensitive to edge-orientation, and optical-flow sensors.

Most of these vision sensors output visual information about the scene in the form of discrete events using Address-Event Representation (AER) [16, 45, 53]. The data encodes the visual information by sending out tuples $[x; y; t; p]$ — of space (the pixel where change occurred), time (when the change occurred), and polarity (whether luminance increased or decreased) — as ON or OFF events, respectively. The event-based camera used in this work is a time-domain encoding event-based sensor with VGA resolution. The sensor contains a 640×480 array of fully autonomous pixels, each relying on an illuminance-change detector circuit. In this study, we will only consider the luminance change circuit that is common to all existing event-based sensors [22].

The operating principle of an event-based pixel is shown in Figure 12. The change detector of each pixel individually detects a change in brightness in the field-of-view. Since event-based cameras are not clocked like conventional cameras, the timing of events can be conveyed with a very accurate temporal resolution on the order of microseconds and below¹.

These sensors capture information predominantly in the time domain as opposed to conventional frame-based cameras, which currently provide greater amount of spatial information. Since the pixels only detect temporal changes, redundant information like static background is not captured or communicated, resulting in a sparse representation of the scene. Consequently, event-based cameras can have a high temporal-resolution with a very low data-rate [14] compared to conventional cameras, thus conforming to low-resource requirements. Since the pixels are independent of one another and do not need a clock, an error in a few of them will not lead to a catastrophic failure of the device and the sensor will be able to remain operational.

¹The highest reported neuromorphic sensor event output rate to date is 1.3×10^9 events per second [84].

3.1.2 Conventional Space Situational Awareness

Space situational awareness (SSA) has been an important topic in military applications for many years [58] [37] [27] [62]. SSA is the ability to detect and keep track of surrounding objects and debris to avoid collisions. For SSA, vision systems with high temporal-resolution and low latency are required to accurately detect objects. Event-based cameras are therefore the perfect candidate to replace limited conventional sensing methods in satellite awareness.

Ender et al. [34] details the use of radar in SSA for collision detection, orbit estimation, and propagation. The benefit of radar is that it has a very large coverage, meaning it can consistently observe a wide area in an arc of almost 5000 km. However, since radio uses long wavelengths, this methodology would only work for larger objects [34]. Smaller objects would be impossible to detect via radio waves.

One difficulty in object detection to avoid collisions in space is the modeling of non-linear orbits in real-time. Several methods have been proposed to predict non-linear orbits for SSA. One is to use Gaussian mixture modeling to exploit properties of linear systems to extrapolate information about a non-linear system, and then to use Gaussian splitting to reduce the errors induced by that extrapolation [35]. The mixture model enables complex, non-linear orbits to be mapped more accurately, providing a better judgment of potential collisions. The issue arises when this kind of surveillance for object avoidance needs to be done autonomously. The calculations presented are too complex to be performed efficiently by a satellite's embedded platform. Also, since the analysis carried out by such platforms is based on statistical manipulation, it needs to be verified by human intervention in order to avoid any statistical anomalies that may cause potential collisions.

Abbot and Wallace [71] tackle the SSA problem of decision support for tracking large amounts of orbiting space debris. They claim that the limited number of sensors leads to inconsistent surveillance of the objects under observation, and therefore propose a cooperative monitoring algorithm for geosynchronous earth orbit satellites to address collision prevention and provide automated alerts. However, this methodology relies on Bayesian modeling, which can be computationally intensive for embedded platforms and requires publicly available data to create the models. With satellites of unknown orbits, unexpected collisions

could therefore become an issue.

These techniques also require fast positional capture of the observed objects which is difficult with the video cameras currently available for space exploration. Event-based cameras could fill this space by providing low-latency and low-resources sensing for SSA.

3.1.3 Event-based Sensors for Space Situational Awareness

The high dynamic range of event-based sensors with both low-light and bright-light sources allows visual information to be inferred even in the darkness of space or when a bright sun is in the sensor's field-of-view (FoV). It also means that the area around the sun can be observed, even when the sun is coming up over the horizon of a satellite's orbit.

The use of event-based cameras in space-related applications is not well developed. Most of the work has been carried out in the context of terrestrial telescope observation of low brightness objects in Low-Earth Orbit (LEO) and Geostationary-Earth Orbit (GEO) [67] [1].

Event-based cameras can offer a promising solution to collision avoidance in space provided their high temporal precision and sparsity of data are properly taken into account when designing algorithms. The current trend of generating frames of events, and gray levels to recycle decades of conventional computer vision and machine learning techniques has led to their being used as simple high dynamic range conventional cameras. In this work we focus only on the temporal properties of these sensors, considering cases of per-event computation that preserve the temporal properties of event-based cameras that have been shown to be the key to developing new applications [4].

There has been extensive research into event-based cameras for real-time tracking and low-power computer systems within the last decade. Many algorithms have been developed that allow for objects to be tracked within the visual space of an event-driven sensor. Reverter et al. developed one such method that makes it possible to track many different shapes, as long as the pattern of the shapes is known a-priori [87]. Similarly, Lagorce et al. provide a multi-kernel Gaussian mixture model tracker for the detection and tracking of different shaped objects [43]. Other methods use spatial matching to allow object tracking even in occluded conditions [57] [18] and provide haptic stability by tracking gripper positions in

microrobotics applications [56]. The low computational requirements of event-based sensors even allow tracking systems to be implemented on embedded platforms [49] and on FPGAs [48]. Newer improved spatio-temporal feature detection could improve these methods further [42]. Novel methods can even detect and track objects in conditions where both the camera and the objects are moving independently [56, 69] [55].

3.1.4 Neutron-Beam Testing

Srouf and McGarrity [81] detail the effects of space radiation on microelectronic circuits, discussing damage, ionization, and SEEs on optoelectronic devices. Modern models describe the most of the radiation experienced in the space environment as consisting of protons and heavy ions [10]. However, this experiment primarily uses wide-spectrum neutrons to test the sensor of interest. In general, neutron beam testing is useful for classifying single-event effects in electronics. Since interest is focused on the response of the device, the source of the upsets become irrelevant. Neutron testing is also useful to test the robustness of systems to SEUs. As an example, NASA Langley Research Center and Honeywell performed neutron beam tests to study the robustness of their flight control computer architecture [36]. Their primary goal was to show that they were able to recover from neutron-induced SEUs. The recovery demonstrated system's capabilities in a hazardous environment, even though the radiation source was not neutrons.

When radiation impacts a device, energy is deposited into the target material, causing various faults in the hardware. These faults can have different effects such as memory corruption or glitches in analog and digital hardware [11]. In an imaging sensor, these errors would manifest as corrupted pixels or improper output. One type of effect, single-event effects (SEEs), occurs when a high-energy particle strikes a microelectronic component and changes a single state of the internals in the device [81]. These effects can then manifest as transient-data errors, corrupting the data output.

3.2 Methodology

This section gives an overview of how the radiation experiment was performed, explaining the Los Alamos Neutron Science Center’s neutron beam and detailing how data was collected during irradiation.

3.2.1 Event-Camera

The sensor used for the experiments in this chapter was an event-based sensor based on [22] with VGA resolution (640×480 pixels) fabricated in $180nm$ CMOS-CIS technology. The chip has a total die size of $9.6 \times 7.2mm^2$, with a pixel size of $15 \times 15\mu m^2$, and a fill factor (ratio of photo-diode area over total pixel area) of 25%. The maximum event-rate for this camera is specified as $66Meps$ (mega-events per second). During recordings, output events were time-stamped with micro-second resolution by the camera interface and communicated via USB to a host computer for storage. In our recordings we observed a maximum of about 30 events captured with the same micro-second timestamp, meaning that the maximum sensor throughput was not reached.

3.2.2 Irradiation

The event-camera under test was irradiated at ICE-II, Los Alamos Neutron Science Center’s wide-spectrum neutron-beam facility. The Los Alamos Neutron Science Center (LANSCE) provides the scientific community with intense sources of neutrons, which can be used to perform experiments supporting civilian and national security research. The ICE facility was built to perform accelerated neutron testing of semiconductor devices. Flight Path 30L and 30R, known as ICE House and ICE-II, allow users to efficiently set up and conduct measurements [85]. The sensor was irradiated for two days, from November 23, 2019 to November 24, 2019 under wide-spectrum neutrons of energies ranging from $0.1MeV$ to $> 600MeV$. The general setup is shown in Figure 13.

An event-based camera was placed at a fixed distance in the beam to act as a control on the effective neutron flux. The sensor was placed at different angles of incidence from the

(A)



(B)

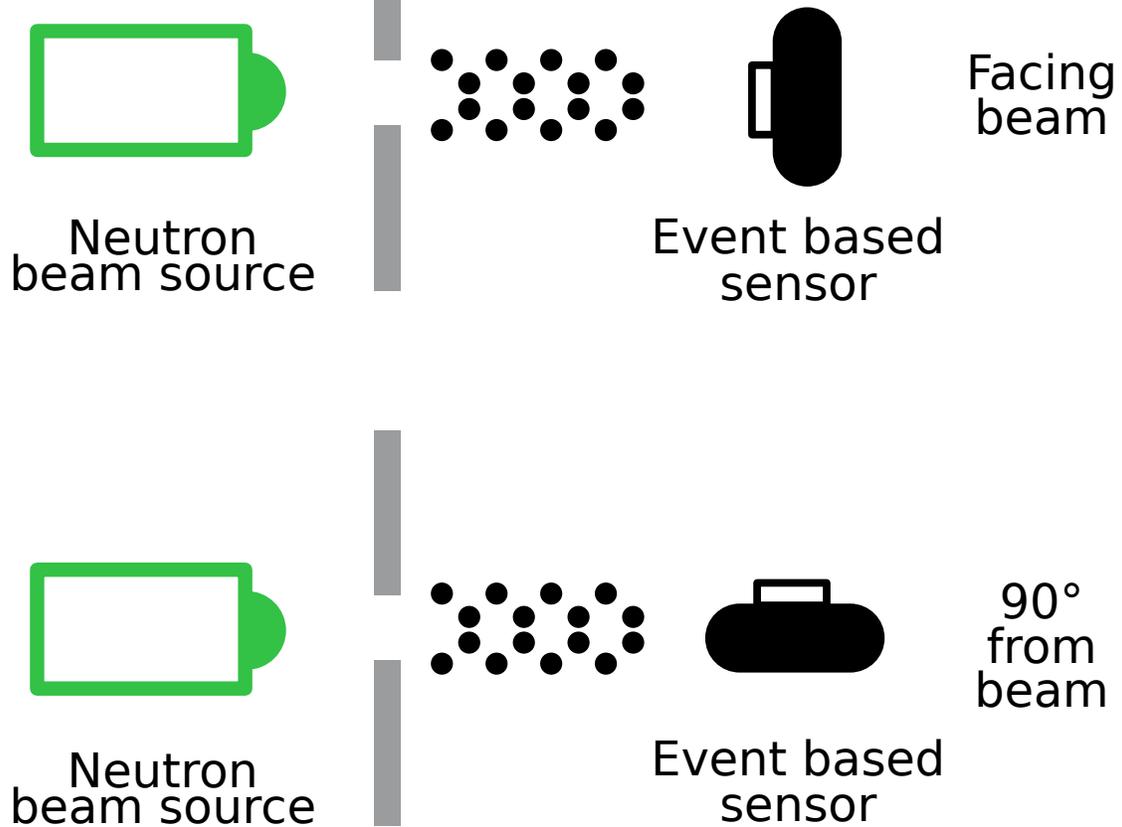


Figure 13: (A) The event-driven sensor under test sitting on a stand that is non-reactive to neutron radiation. To ensure that the neutrons passed through the sensor, the green laser was used to aim the beam. (B) Schematics showing the sensor placed at a fixed distance from the beam source in two conditions - facing the beam directly and at a 90 ° angle of incidence.

beam as shown in Figure 13(B) to detect any potential differences in the effects observed. The placement of the sensor relative to the neutron beam is achieved by manually lining up a laser guide that is centered on the 1-inch collimator. Minor misalignment leads to variable density of radiation across the sensor, but was found to not affect the final outcome of the overall measurement at individual pixels. Data is collected at an angle of 90° from the beam and directly facing the beam source.

In this experiment, the event-camera was irradiated with the lens cap on to avoid any light or environmental noise on the sensor. Thus, the noise recorded from the sensor in this experiment primarily come from the effects of the radiation rather than those induced by the light sources in the environment.

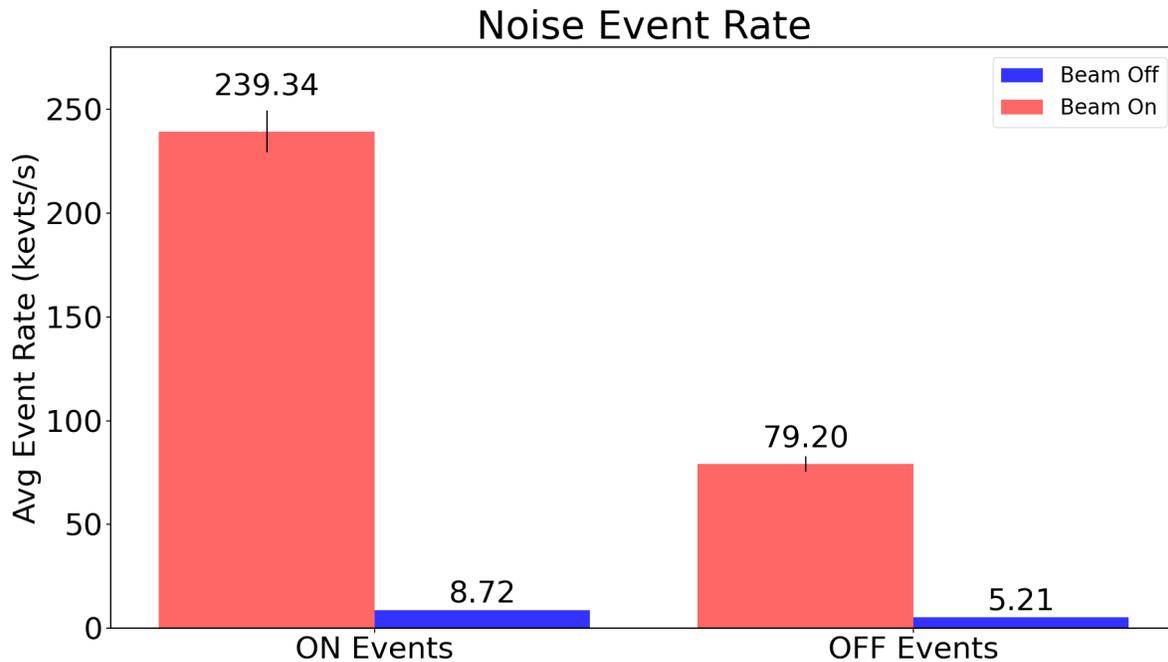


Figure 14: Average number of noise events per second induced due to radiation compared to noise without irradiation over 2 days of irradiation across all tested angles-of-incidence. The recordings were taken with the lens cap on the camera so the induced events were due either to the inherent thermal noise or to noise induced through the neutrons. Radiation induced more ON events than OFF events (3:1 ratio).

3.2.3 Data Collection and Analysis

The sensor was connected to a computer running software which interfaced with the sensor to record events. Events were later processed offline. Data was taken with the beam on and off in order to observe the increase in noise caused by irradiation. Radiation-induced noise can be seen in the form of clustered noise-like patterns and line streaks of moving particles in the focal plane, as will be detailed in the following sections. The recorded data was parsed to get an event rate to measure the number of events generated by the sensor per second. The counted events were then separated into ON and OFF events. The average events per second were calculated for each experiment with standard deviation as error.

Data was collected with the sensor facing the beam source and at 90° , to observe how the angle of incidence affected the incoming radiation noise. The number of events was measured for both ON and OFF events in each orientation and compared. A Mann-Whitney U test was used to determine statistical significance in the differences between the two orientation distributions [54].

This experiment measured patterns influenced by the effective neutron flux and the number of ON events and OFF events. The patterns were analyzed using an understanding of the sensor's internal circuitry to determine the physical effect of radiation on the sensor. This methodology presents a categorization of SEEs in the form of radiation-induced noise.

To ensure the radiation-induced noise would not overwhelm signal integrity, a pendulum was placed in the visual field to measure the signal-to-noise ratio. Since the signal could be observed with and without radiation-induced noise, the signal-to-noise ratio could be calculated by simply dividing the number of signal events by the noise events produced by radiation. This ratio could then be used to determine the robustness of the sensor to radiation in terms of loss of signal integrity. To validate the signal-to-noise ratio, a correlation test was performed between the radiated data and the non-radiated data.

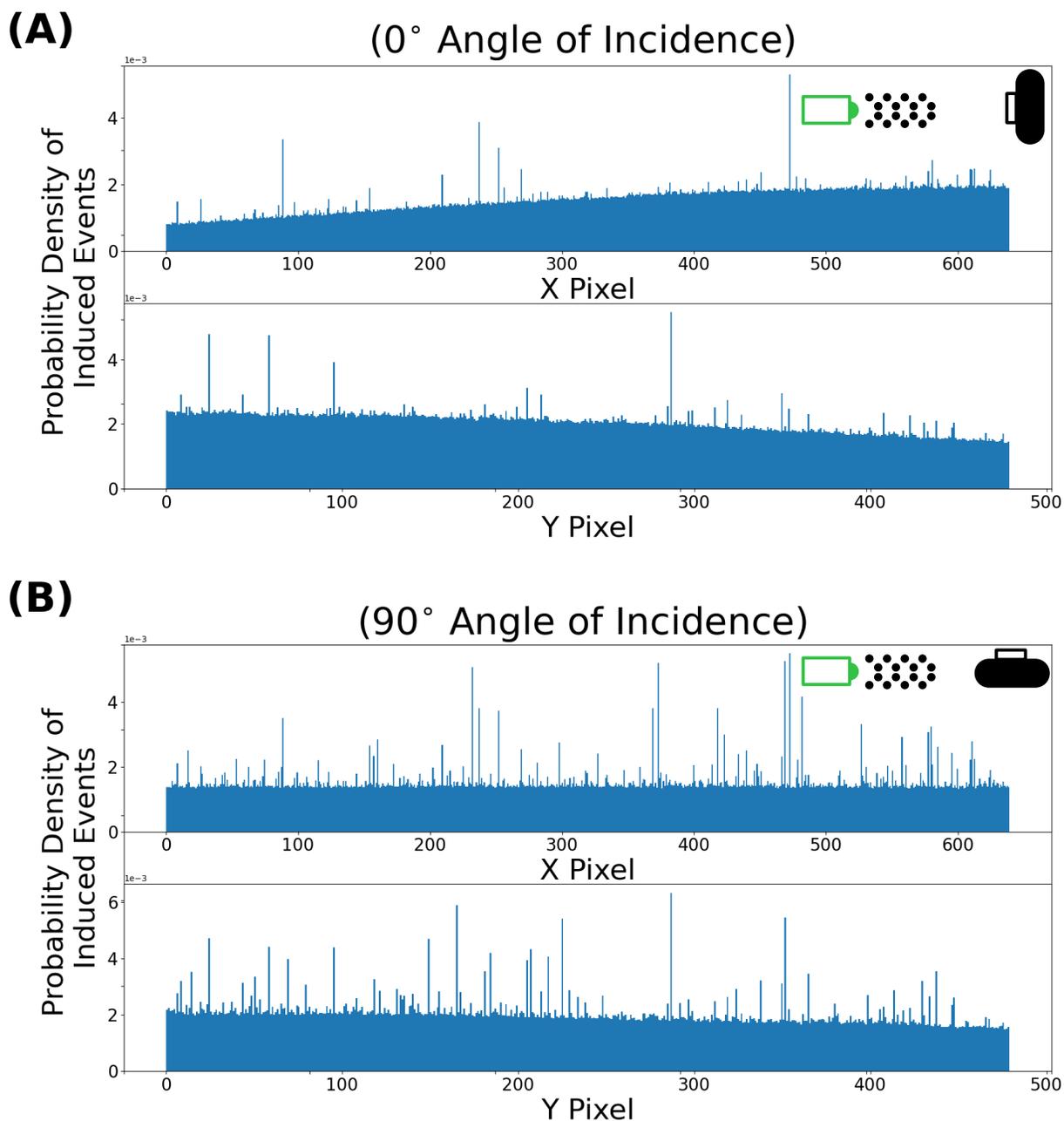


Figure 15: Probability density of events by location on the sensor for (A) 0° angle of incidence and (B) 90° angle of incidence. At 0° , more events were produced at high x and low y values than for the opposite corner. This inconsistency is the result of an inability to precisely align the sensor relative to the center of the beam path. However, it does not affect local response at individual pixels that showed similar noise measurement.

3.3 Results

This section gives an overview of the results of the radiation experiment, discussing noise rates, patterns, and analyses.

3.3.1 Induced-Event Rate

Data was collected with the lens cap on the sensor to minimize environmental influence from external lighting. First, the mean number of radiation-induced ON and OFF events per second was measured. The average number of events can be seen in Figure 14. A significant bias towards ON events was observed although the sensor was biased to generate a similar number of ON and OFF events under normal conditions. When there is an imbalance between the number of ON and OFF events in a DVS, it may mean that there are sudden very fast transients. This behavior is because a DVS pixel is designed with a built-in refractory time that causes the pixel to be inactive (not sensing any further light change) immediately after producing an event [47]. This behavior prevents very active pixels from taking over most of the camera's I/O bandwidth. This refractory time is typically in the range of 1-2ms. Therefore, if a DVS pixel is exposed to very fast ON-OFF (or OFF-ON) transients, most likely only the first ON (or OFF) event will be produced, while the majority of the rest will be filtered out. In Figure 14, the imbalance is towards ON events, meaning there could be very fast ms-range ON-OFF transients. This effect will become more apparent later in Section 3.4.

The induced-event probability density was plotted against the pixel coordinates of the sensor to observe any location preferences for upsets. To measure this location preference, the pixel location of each induced event was divided by the total number of events measured for both angles of incidence. These measurements can be seen in Figures 15(A) and 15(B).

In both cases, the induced events are quite uniform across the sensor, with the 0° angle of incidence tending to bias towards the location of the neutron beam's 1 inch diameter. We can see that about twice as many events were produced for high x and low y values than for the opposite corner. However, this inconsistency is due to human error in placing the sensor

in the beam path. In other words, there is no particular area of the sensor that is more vulnerable to neutron radiation effects than other areas. This result is further demonstrated in the 90° angle of incidence result. Every pixel across the sensor showed a similar response.

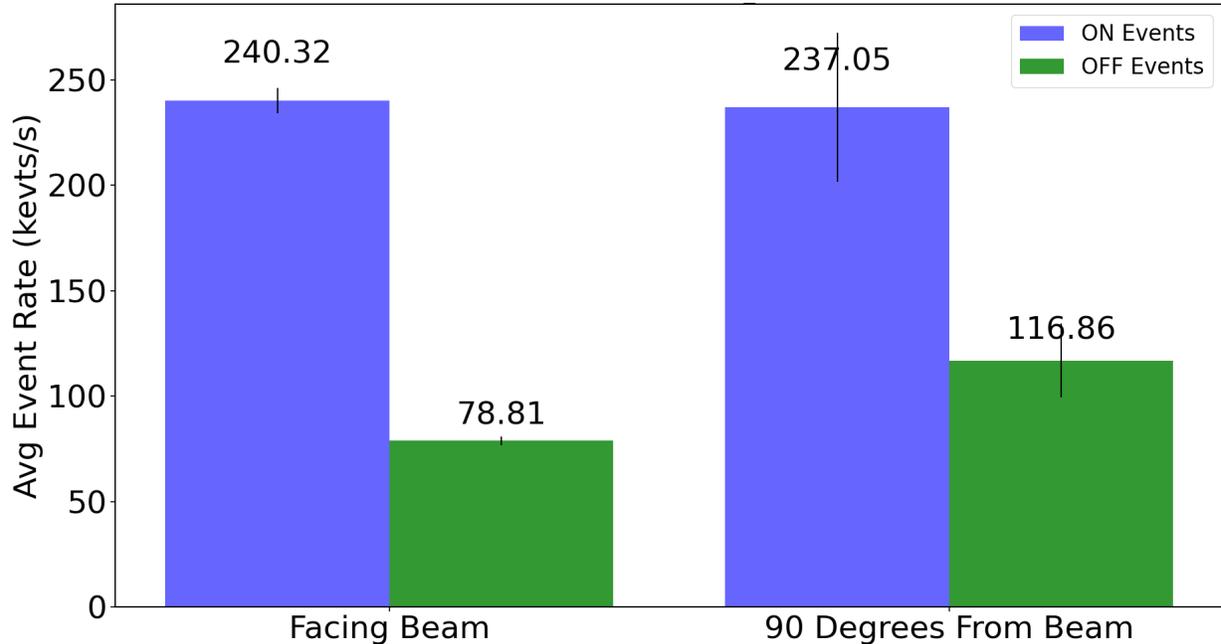


Figure 16: Events observed at different angles of incidence. Data was collected at 90° from the beam and facing directly towards the beam. No significant difference was found between the number of noise events generated for the two conditions even though the sensor would be expected to interact with more neutrons when facing the beam. This outcome is due to more long event streaks being induced in the 90° case. Noise streaks amount to a significantly higher event-rate per neutron than induced clusters.

3.3.2 Angle of Incidence Comparison

Data was collected at two orientations: facing the beam with an angle of incidence of 0° and at an angle of incidence of 90° from the beam source. These two distributions were then analyzed separately to observe any significant differences.

Figure 16 shows that there was a slight difference between the number of OFF events

per second induced between the two orientations. A Mann-Whitney U test was performed on the two distributions to test for statistical significance [54], but no statistically significant difference was found. We found that for 0° the radiation induced noise primarily generated cluster noise representing a random burst of events in a small area, whereas 90° angle of incidence generates more streaks of noise [refer Section 3.3.5 for details]. This result means that while the 0° of incidence has higher probability of inducing noise events, these only generate events in small areas. Whereas the 90° of angle of incidence may have lower probability of inducing noise, but the generated noise affects multiple pixels leading to higher noise event rate. Thus, the noise event rate over the long duration of recordings come out to be statistically similar.

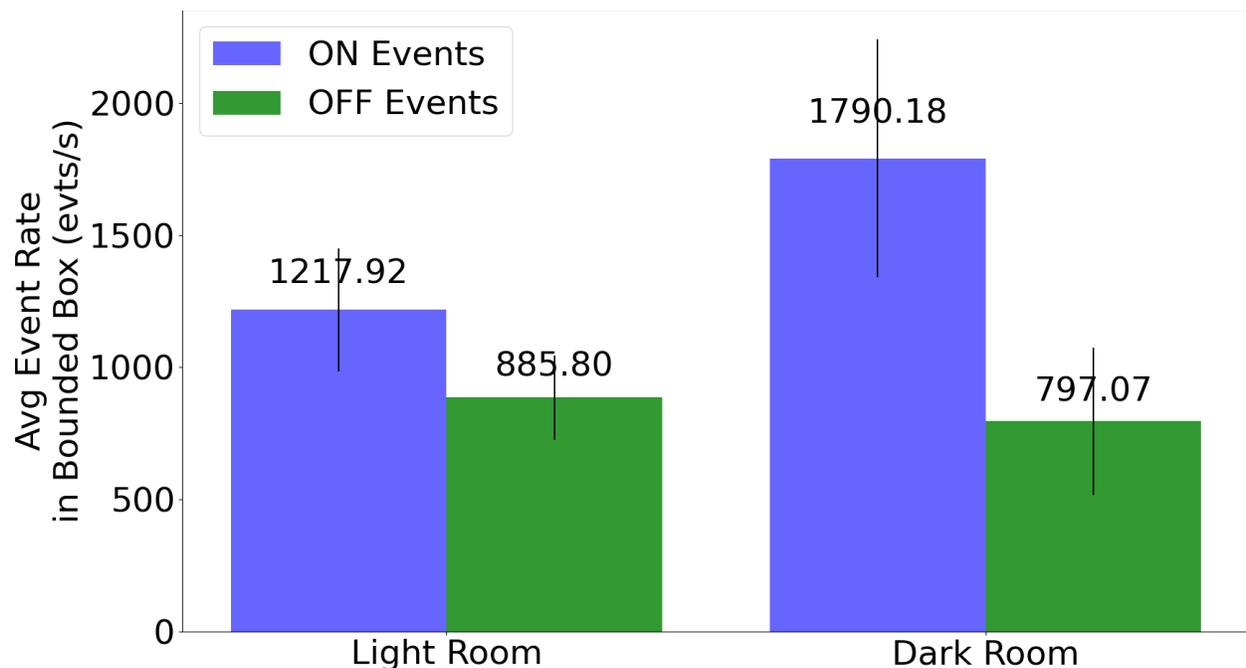


Figure 17: Number of events induced in a 50×50 pixel bounded box for a light room vs a dark room. Given the contrast sensitive nature of the sensor, and as expected, we observed that more ON noise events were generated in the case of dark room since the neutron interactions allowed for the event generation threshold to be crossed more often. The OFF noise events did not increase significantly.

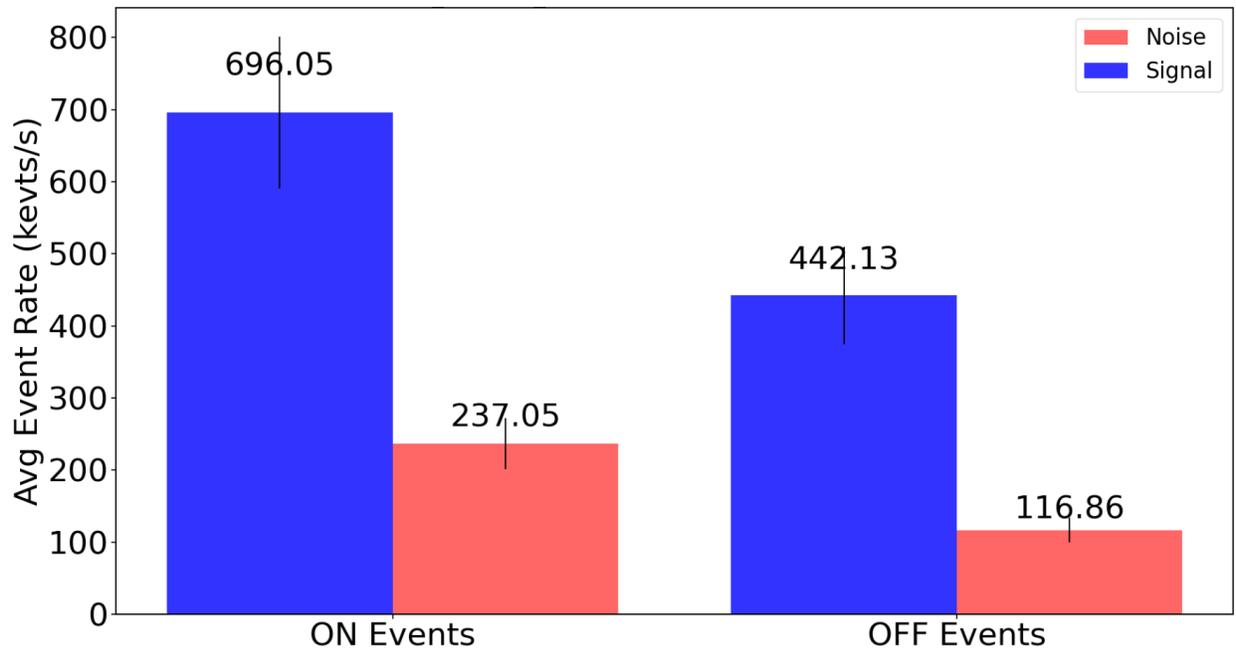


Figure 18: Number of signal events observed vs radiation-induced noise events. Signal events were calculated as the rate of events while recording a cyclic pendulum whereas the noise rate was computed from isolated radiation induced events. The signal-to-noise ratio for the sensor even under strong neutron radiation was found to be 3.355.

3.3.3 Effects of Room Brightness

When deployed in space, these vision sensors may be subject to varying levels of background light intensity. To understand how neutron radiation would affect the sensor under such varying conditions, we recorded background noise events during radiation while placing the sensor in an artificially lit room with illuminance levels of around 500lux and with a lens cap covering the sensor to simulate a low-light intensity condition with a light level close to 0 lux . The intrinsic characteristics of the sensor pixels allow them to be invariant to the background lighting conditions thanks to the relative change operation mode and the log scale. Figure 17 shows the number of ON and OFF events induced by neutron radiation in the artificial lit “light room” and in the low-intensity “dark room” case. We find that the number of ON events induced in the dark room was nearly 1.5 times higher than in the light room. Conversely, no significant difference were observed in the OFF events induced in the two conditions. Details of this process are explained in Section 3.4.

3.3.4 Signal-to-Noise

In order to measure the signal-to-noise ratio, events were compared with the beam ON and OFF while the sensor observed a dynamic scene composed of a cyclic-pendulum, as shown in Figure 19(A). To calculate the ratio between the two values, the number of signal events measured with the cyclic-pendulum were compared directly with the number of isolated radiation-induced noise events. This comparison can be seen in Figure 18. Comparing these values gives a signal-to-noise ratio of 3.355.

To ensure the signal can be seen even when radiation is introduced, events in a 50×50 -pixels bounding box, shown in blue and red in Figure 19(A), were measured and plotted to compare signal data with and without radiation. Since the pendulum’s movement is cyclic, we calculated the event rates data over time using a moving window of 1 ms. The frequency of this rate data calculated using Fourier transform should ideally give us the frequency of oscillation of the pendulum. The Fourier transform of the signal with and without radiation is shown in Figure 19(B). With the addition of radiation noise, the signal’s major frequency can still be estimated with some slight noise at low frequencies.

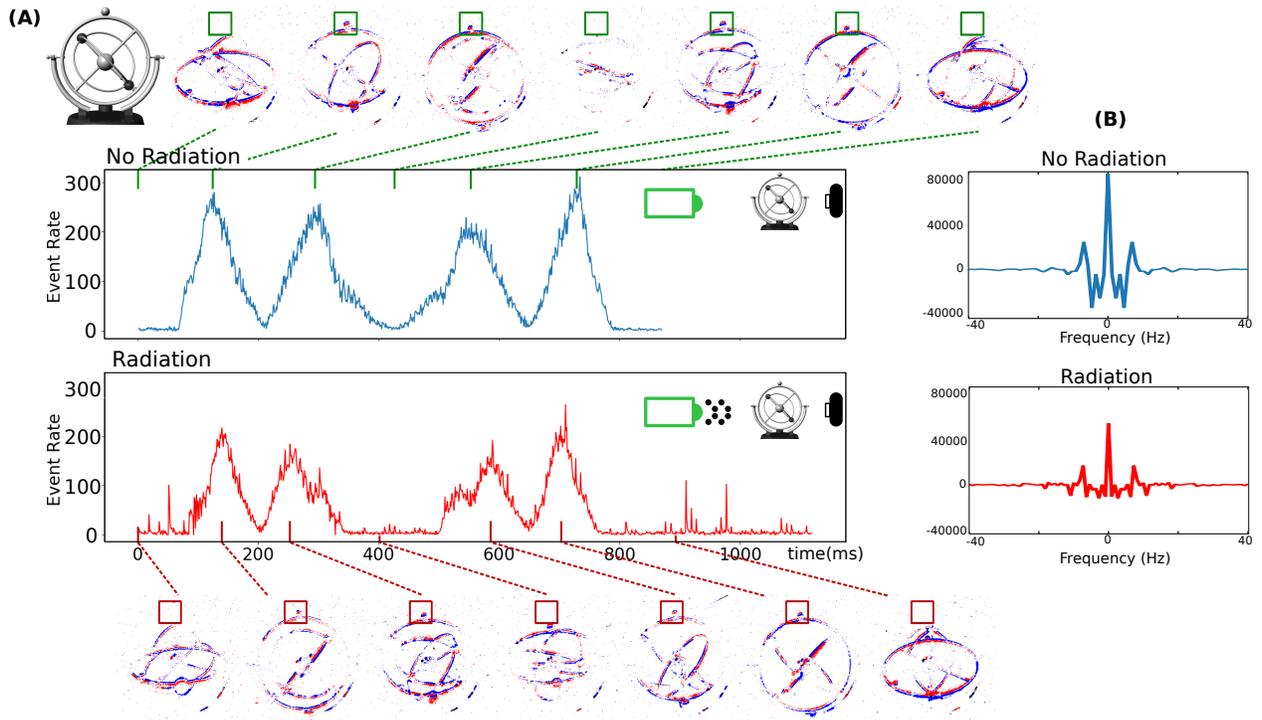


Figure 19: (A) An orbital pendulum recorded using the sensor and the event rate calculated as the number of events within a 1 ms moving window with (red) and without (blue) radiation turned on within a bounded box, as shown in the image panels. The images show the event frames obtained within the time window at different time points in the recording. Qualitatively, the sensor produced similar images for both conditions. (B) Calculated frequency of the pendulum using the event rates. The frequency of the pendulum's motion could be obtained using the FFT in each case.

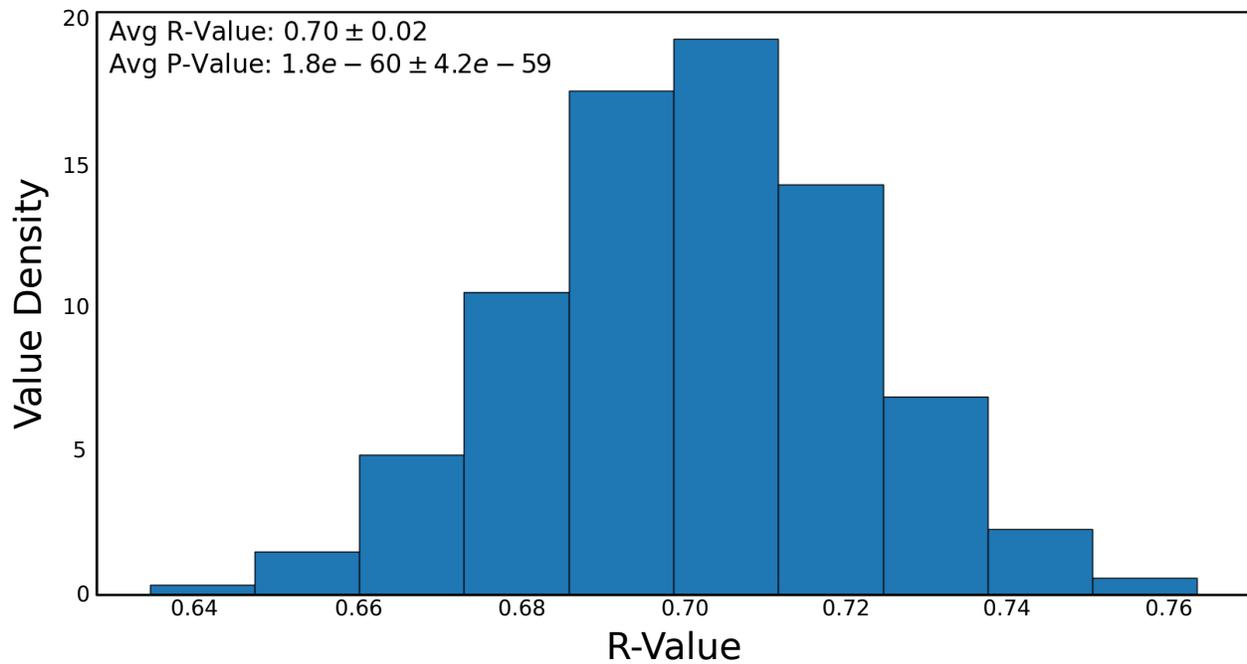


Figure 20: A Pearson correlation test was performed for the events obtained from the pendulum's movement with and without radiation. The high correlation and small standard deviation show that the signals obtained from the two conditions were quantitatively similar.

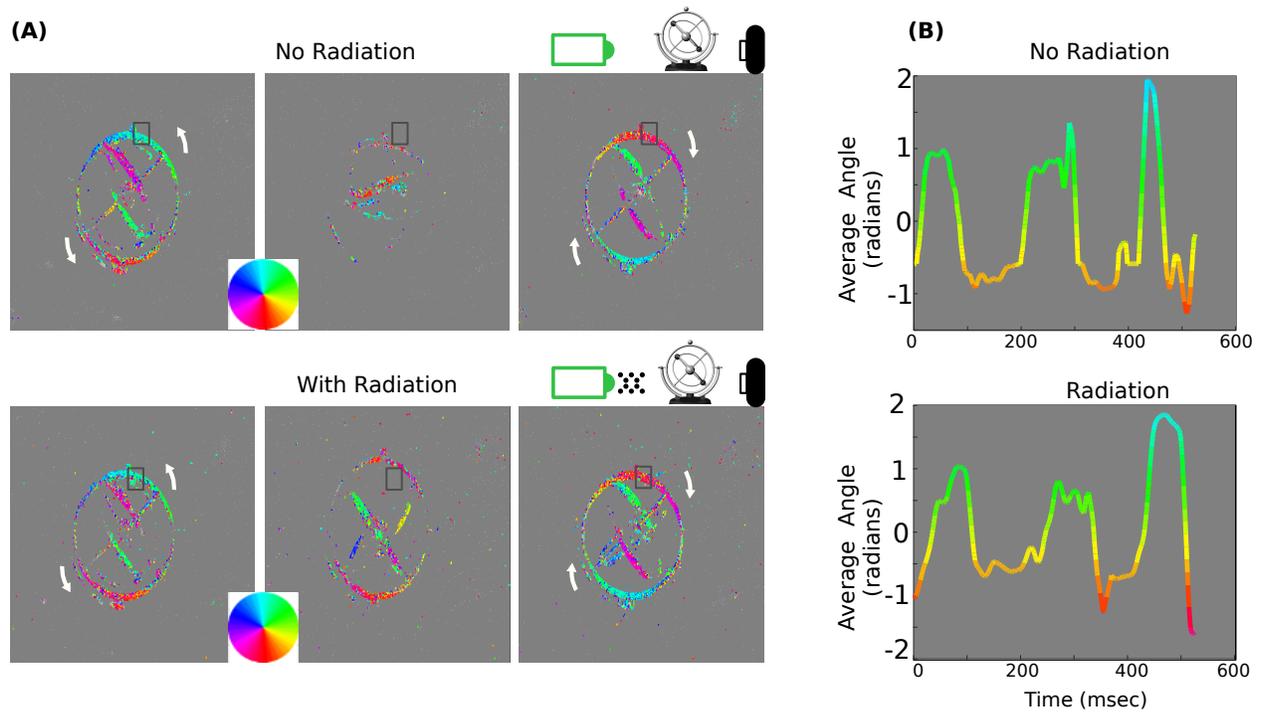


Figure 21: The movement directions of different parts of the pendulum system computed from the recorded event streams with and without radiation. (A) The colors represent the movement directions of the events as indicated by the color wheel. (B) Graphs showing the computed average movement directions for events occurring in a 5ms moving window within the black bounding box shown in the images. The Pearson correlation coefficient between two signals was 0.7189 indicating that the direction computation was not affected by the radiation.

To validate the signal-noise ratio of the radiated sensor, a Pearson-correlation test was performed between the radiation data and the non-radiation data. With a high correlation, it can be shown that the two distributions follow each other closely with minor linear transformations. Due to the varying size of samples, sub-samples were taken and analyzed to estimate the correlation R-value. The distribution of R-values can be seen in Figure 20. The measured R-value was 0.70 ± 0.02 with a negligible p-value. It can therefore be deduced with high confidence that the radiation-induced noise is not enough to significantly change the data output from the original, non-radiated data.

The ultimate goal of deploying sensors on missions is to obtain useful information from them while in space. One of the most fundamental, low-level features that can be extracted from the event stream is motion flow. The optical flow provides the speed and direction of an object's movement in the camera plane, where its precision is related to the temporal properties of events. We computed optical flow on events captured from the sensor recording the moving pendulum system using the aperture-robust event-per-event optical flow technique introduced in [3]. The average direction of movement of one arm of the pendulum inside a bounding box (shown in Figure 21(A)) is plotted in Figure 21(B). The average angle values follow the expected wave as the arm of the pendulum moves up and down vertically. The Pearson correlation between the two conditions was found to be 0.7189, showing that movement computation is not affected by radiation.

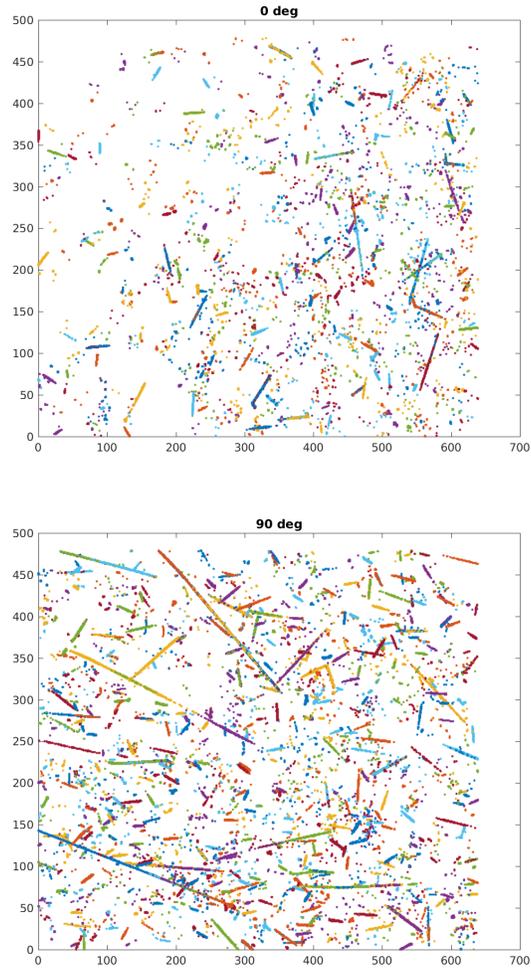


Figure 22: Clustered patterns of noise obtained by searching for clusters with minimum sizes of 10 pixels in 30-second recordings. Time slices of 5 ms were processed consecutively, searching for 10-pixel clusters. All clusters detected during 30 seconds are grouped in the plots. Events were recorded for a 0° angle of incidence and a 90° angle of incidence. Significantly more line segments can be seen at 90° . At 0° , fewer, smaller (average 30 pixels) and more clustered noise patterns were observed than at 90° , where longer (up to 300 pixels) and more frequent (up to 7 times) line segments were observed.

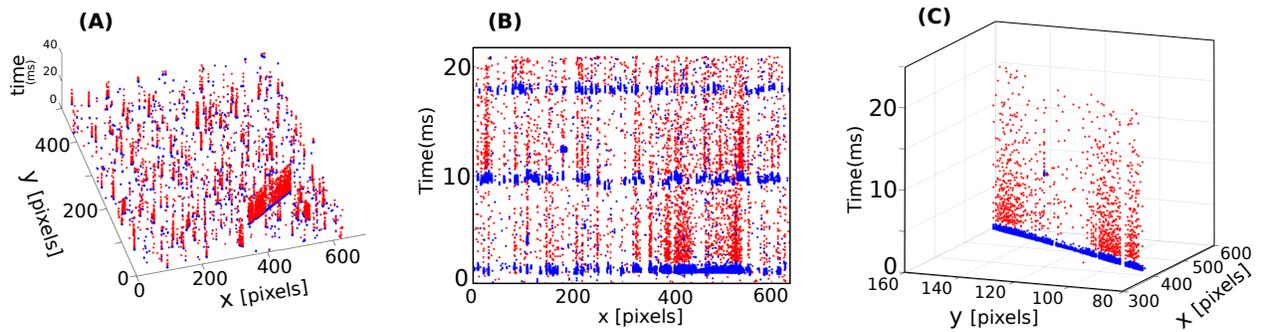


Figure 23: Details of 20ms event capture when exposed to neutron beam without visual stimulus. The blue dots represent positive events and the red dots represent negative events. Positive events are mostly concentrated in 600-800 μs time intervals separated by about 8ms intervals in which mostly negative events are recorded. (A) 3D plot (x,y,time) of events captured during the 20ms interval. Small scattered dots/clusters can be observed plus a line segment in the lower right part. (B) Time vs x-coordinate projection of the recorded events. (C) Events corresponding to the line segment in (A) which have been isolated for better visibility.

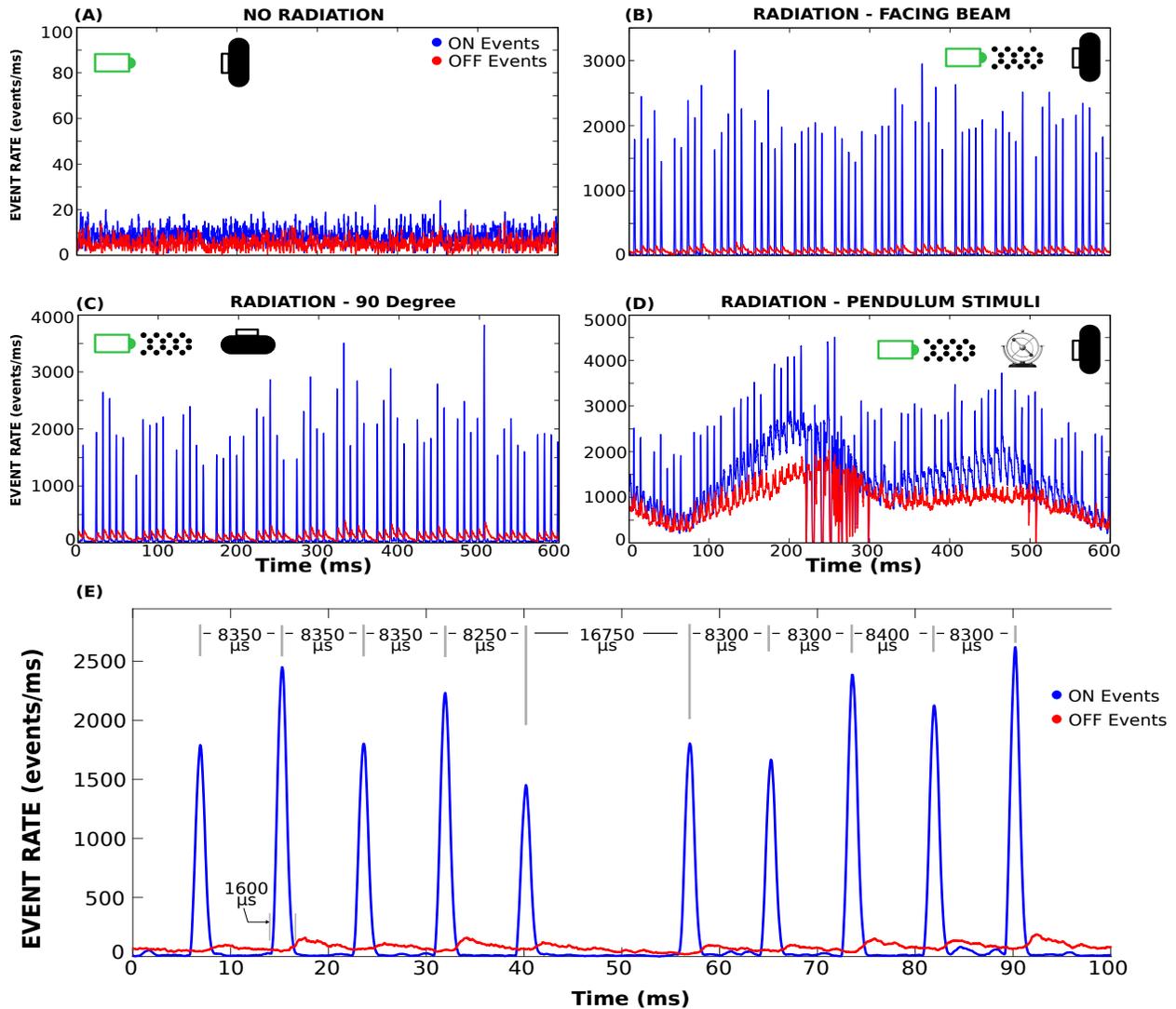


Figure 24: Noise rates for different conditions with and without neutron radiation. (A) The overall noise without radiation is very low. (B, C) Radiation noise when the sensor was placed at 0° (facing) (B) and at 90° (C) to the beam source. In each case, we recorded the bursts of noise most likely due to neutron pulses from beam generation. (D) Similar noise was found in the recording when a circular pendulum was recorded with the camera. The burst noise was superimposed on the low frequency events generated by the pendulum motion. (E) Details of the neutron's macro-pulse sequence can be observed from a zoomed-in plot of the event bursts in (B). Each neutron macro-pulse produced positive event bursts with duration of about 1.6ms , and with peaks separated on average by 8.3ms . Five macro-pulse responses appear, with a duration between the first and the fifth of 33.25ms , while the time between two 5-macro-pulse trains is 16.75ms .

3.3.5 Noise Patterns

Radiation-induced noise, as shown in Figure 22 for both orientations, can be categorized into two main groups: clusters and line segments. Line segments represent a line of events that appear across the frame due to a neutron impacting the sensor at a non-zero angle of incidence. Clusters represent a random burst of events in a small area. The angle of incidence between the sensor and the radiation source affects the number of line segments. About 5-7 times more line segments appear with a 90° angle of incidence than with a 0° angle of incidence. Conversely, about twice as many clusters appear with a 0° angle of incidence than with a 90° angle of incidence. Significantly longer lengths of line segments occurred at 90° , where streaks of up to 300-pixel lengths were observed, whereas smaller streaks, with maximum lengths of 30-50 pixels, were seen at 0° . An example of differences in noise cluster patterns can be seen in Figure 22. These figures were obtained by analyzing recordings of 10^7 events, with a duration about a 30-second, while searching for unconnected clusters not exceeding 10-pixel in size. Note that in Figure 22(A) more event density can be seen in the corner of high x and low y values than in the opposite corner. This result is similar to what was observed in Figure 15 due to human error in placing the sensor in the beam path.

Analysis of noise line segments showed a burst of ON events over a fast time frame, followed by a long relaxation-period of OFF events after a short wait time, as shown in Figure 23. This pattern is due to an influx of positive current in the sensor's photo-diodes creating a burst of ON events, followed by a relaxation period for the current to return to normal, creating OFF events. The ON events burst over about $600\text{-}800\mu\text{s}$ and the negative event tail is about 10ms long.

Viewing the event rate of the bursts, we see peaks of ON events followed by a long tail of OFF events. This effect is seen within all noise-types and is shown in Figures 24 (B) and (C). Figure 24(E) shows a zoomed view with finer details. Bursts of 5 peaks separated by a time of 16.75ms can be seen. Each peak has a duration of about 1.6ms of positive events. Consecutive bursts are separated by 8.25ms within the 5 peaks. Consequently, on average, the five peaks occur every $33.25\text{ms} + 16.75\text{ms} = 50\text{ms}$, which is equivalent to 100Hz peaks. This frequency coincides with the LANSCE neutron source description [92], where

the neutron source emits a pulse of neutrons at a rate of about 100Hz . Each such neutron peak is referred to as a neutron “macro-pulse”.

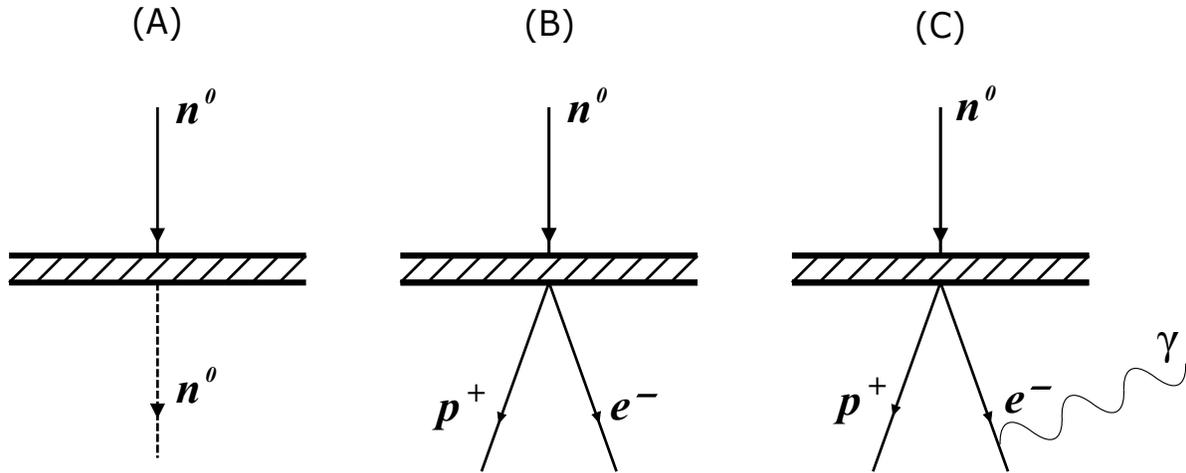


Figure 25: Three possible free neutron decays. (A) The neutron passes through the sensor casing without decaying. (B) The neutron decays into a proton and electron. (C) The neutron decays into a proton and an electron which emits gamma radiation.

3.4 Circuit-Level Interaction Interpretation

High-energy neutron beams are thought of as ionizing radiation, which can instantaneously change the charge of an electric circuit node within the camera sensor chip. Since an event-camera can capture internal changes with microsecond resolution, these sensors provide a new way of “seeing” fine interactions taking place between fast radiation particles and the electronic chip while it is operating.

For the free neutrons passing through the sensor, there are three main possibilities: the neutron can pass through as a neutron without decaying, the neutron can decay into a proton and an electron, or the neutron can decay into a proton and an electron which emits a gamma photon due to internal brehmsstrahlung [30]. A diagram of these three possibilities can be seen in Figure 25. Due to quantum uncertainties and the inability to distinguish between particles, it is impossible to distinguish the cases’ impact on the sensor in this experiment. Further research must therefore be performed to detail the exact cause of the induced noise patterns.

In digital circuits, high-energy charged particles and radiation beams tend to mainly impact memory circuits, where charge is stored on tiny parasitic capacitors, producing bit-flips and consequently altering system states and data. In our sensor, however, we observed consistent sudden positive events over many pixels followed by negative event tails, synchronously with the macro-pulse neutron emission patterns of LANSCE [92]. The fact that most responsive pixels produce a burst of positive events during each $625\mu s$ LANSCE neutron macro-pulse, rules out the possibility that the sensor is suffering bit-flip effects at temporary memory-storing nodes. If this were the case, we would expect to observe a random mix of positive and negative events within each neutron macro-pulse. However, most of the affected pixels respond by providing a synchronized burst of positive events. It can thus be inferred that it is the pixels’ photo-diodes that are responding to the neutron macro-pulses. Photo-diodes drive a photo-current proportional to incident light intensity. If a high-energy proton, neutron, or electron crosses the depletion region of a photo-diode, it will interact with the electrons flowing through it at that moment, thus producing a sudden decrease in photo-current and, consequently, negative events. However, since we observed a sud-

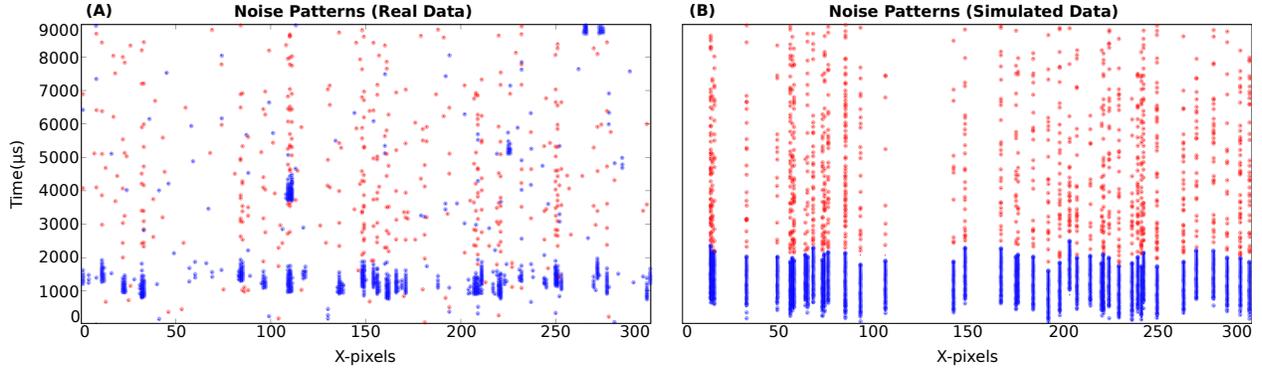


Figure 26: Examples of X-projections of noise line segments for events recorded from (A) real data and (B) simulated noise data. All cases show a burst of ON (blue) events with a long OFF (red) tail. As each example line segment was detected, the events were separated from the recording and then projected on the X-axis.

den, very significant increase in photo-current (resulting in positive events), we hypothesize that the scattered pixels are sensing sudden radiation at their locations. This reasoning would also explain the observation of segments sensed simultaneously by consecutive pixels. Figure 23 shows one such segment in a $20ms$ time slice of events, corresponding to three consecutive $625\mu s$ neutron macro-pulses separated from each other by $8.25ms$. Most of the pixel responses show small clusters of less than 10-pixels, the exception being the 190-pixel long segment. Our hypothesis is that the sensor is crossed by radiation bursts, most of them perpendicular to the chip plane, but occasionally interacting with deflected radiations at other angles and producing line segments. However, all radiation interactions occur precisely during the beam's macro-pulse times.

The electronic pixel circuitry of an event-camera chip has a limited response time in the range of $0.1ms$ to $10ms$ depending on ambient light and bias conditions [22] [79]. The LANSCE neutron source macro-pulses have a time duration of $625\mu s$, which is lower than the temporal resolution of the event sensor. The macro-pulse radiation impinging on the destination pixels produces a sudden over-stimulation of photo-current, resulting in the sudden generation of a handful of positive events per pixel during the neutron macro-pulse. Af-

ter such strong over-stimulation, the pixel circuit relaxes to its steady ambient-light-driven state with a time constant in the range of $10ms$, producing events of negative polarity over time. This behavior of sudden positive stimulation of $600-800\mu s$, where positive events are produced, followed by about $8-10ms$ of negative-event relaxation is systematically observed in the recordings. Figure 23(A) shows the $20ms$ event capture with scattered noise-like dots/clusters of fast positive events (shown in blue), followed by negative event tails (shown in red). We hypothesize that each such dot/cluster corresponds to a neutron crossing the chip. Figure 23(B) shows the events in Figure 23(A), but displayed in their corresponding time vs x-coordinate projection. We can clearly see the synchronized sequence of neutron macro-pulse-induced positive events (shown in blue), of $600-800\mu s$ duration, separated by about $8ms$ of inter-neutron macro-pulse time where mainly negative relaxation events are produced. The figure also shows a 190-pixel long segment with the same time profile. The events for this segment are isolated in Figure 23(C). In this plot there are 2,031 positive events collected over about $800\mu s$, followed by 1,090 negative events collected during over about $20ms$.

While the conditions in space are more random, modelling radiation in a pulse-like nature still holds valid. Radiation events happen in space typically due to localized cosmic events, such as supernova or solar flares and winds. Time-series analysis of the low-earth orbit radiation environment seen on the ISS is measured to be in pulses, shown in [95], where the cumulative dose increases similarly to a step function.

The suddenly induced photo-current hypothesis also explains the observations in Figure 17, where more positive events are produced under dark-room conditions than under light-room conditions. When under light room conditions, the photo-diodes are already driving some current and consequently reach their maximum saturation current earlier when suddenly impinged by high energy particles, resulting in fewer induced positive events. Under dark conditions, the photo-current can undergo a larger variation, resulting in more positive events.

3.5 Event-RINSE Simulator

This section describes the Event-RINSE simulator. The purpose of the simulator is to model the noise seen in radiation testing as close as possible to be used in testing future applications without the need for radiation testing.

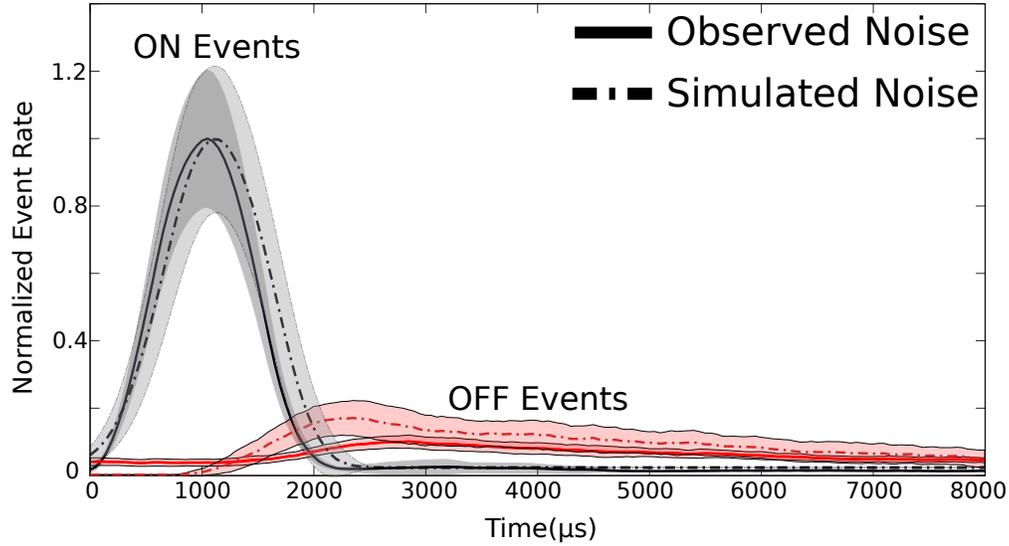


Figure 27: Average single pixel radiation-induced event rate model for observed and simulated data. From real data we observed that neutron interactions induced ON event bursts of about $1.6ms$ within the first $1ms$. These were followed by long tails of OFF events lasting up to $10ms$. The simulator was used to induce noise events into the stream of recorded non-noisy data, and the noise characteristics for single-event noise were then averaged to create the dashed curves. The simulator was able to match the real noise model within a margin of acceptable error.

3.5.1 Simulated Noise Generation

Given a stream of event-camera data as input, the simulator updates the induced noise at time-steps of $1\mu s$ each over the entire duration of the input data. The noise profile is created independent of the input stream. For every time-step, the noise signal determines the probability of occurrence of a noise event, its location in pixel, and the angle of incidence of the neutron. The probability of injection was determined using a Poisson distribution of observing $k = 1$ event with a variable event rate. Namely,

$$P(\lambda) = \lambda e^{-\lambda} \quad (3-1)$$

where λ is the frequency of an event happening per microsecond. A starting pixel is randomly chosen uniformly across the resolution of the sensor. The simulator decides whether injected noise is in the form of a cluster or a line segment based on the angle of incidence parameter. Specifically, the chance of injecting a cluster is based on the cosine of the angle of incidence with some jitter-error. Thus, the probability of the injected noise pattern is given by

$$\begin{aligned} P(Cluster) &= |\cos(\theta + \varepsilon)| \\ P(Line\ Segment) &= 1 - P(Cluster) \end{aligned} \quad (3-2)$$

where θ is the angle of incidence in radians and ε is a small amount of error. A cluster's shape is modelled by randomly chosen pixels around the neighborhood of the starting point. A line segment is modelled by a straight line with a randomly chosen angle between 1° and 360° . The setup at LANSCE only allows us to perform measurements at a single angle of incidence. However, in space, there is no preferred angle of approach of incoming radiation, nor is there a pre-defined orientation of a satellite. We found that while the angle of incidence does not affect the noise profile of individual pixels, it does affect the probability of inducing a noise-line. Thus, we used the two orientations to derive a cosine model and induce a noise-line generation based on a randomly sampled angle of incidence.

For each pixel in the shape of the generated event, the noise pattern is modelled by sampling a time window for ON events from $\mathcal{N}(2000\mu s, 200\mu s)$ which represents the length

of time for the burst of ON events. OFF events are sampled from $\mathcal{N}(8000\mu s, 1000\mu s)$. More precisely,

$$\begin{aligned}
 P(ON \text{ Noise Event}; t) &= e^{-\frac{1}{2}\left(\frac{t-t_{ON}}{\sigma_{ON}}\right)^2} \\
 t &\in [0, t_{ON}] \\
 t_{ON} &\sim \mathcal{N}(2000, 200)
 \end{aligned} \tag{3-3}$$

where the burst of ON events is simulated as a Gaussian model with the mean as the sampled ON-event time window (t_{ON}) and standard deviation $\sigma_{ON} = 340\mu s$ is used to determine the probability of generating an event over time t . The wait time between the burst of ON events and the OFF-event relaxation period is sampled from $\mathcal{N}(100, 50)$. After the wait time (t_{Wait}), the current relaxation of OFF events is modelled using an exponential with decay parameter, $\beta = 5200$ up to a total OFF time (t_{OFF}) as per Eq.(3-4).

$$\begin{aligned}
 P(OFF \text{ Noise Event}; t) &= \frac{1}{\beta} e^{-\frac{1}{\beta}t} \\
 t &\in [t_{ON} + t_{Wait}, t_{OFF}] \\
 t_{Wait} &\sim \mathcal{N}(100, 50) \\
 t_{OFF} &\sim \mathcal{N}(8000, 1000)
 \end{aligned} \tag{3-4}$$

The generated events are then added to the data file and sorted by timestamp in ascending order. Finally, the file is saved to be used in testing or evaluation. The algorithm to generate radiation-induced noise events is detailed in Algorithm 1.

3.5.2 Pattern Validation

To validate the simulation environment, noise events were generated following the pattern described in Algorithm 1 and compared with noise events from real data. The noise events were plotted against time to compare them with noise from observations. Figure 26 shows a sample of visual real noise events (Figure 26(A)) vs simulated noise events (Figure 26(B)). The model used to generate noise was compared to the average observed single-event noise. The model shown in Figure 27 fits the observed pattern with a 5% error rate for ON noise profiles and 12.3% error rate for OFF noise profile.

3.5.3 Simulation Environment Usage

The Event-RINSE simulation environment is written in Python with many supporting parameter flags that can be used to modify the simulation model. Normal Python data analysis modules are needed for the simulator, namely SciPy [61] and NumPy [86], while OpenCV [17] is used to display videos of the event data. The simulator is run using Python3 environment with runtime flags for campaign customization. Currently available flags and descriptions can be seen in Table 4. The input data file is the only input that is necessary to run the simulator. Input files are assumed to be plain text files in $\langle x \rangle \langle y \rangle \langle timestamp (\mu s) \rangle \langle polarity \rangle$ format. The Event-RINSE open-source code will be available to the scientific community after a security-clearance process.

Algorithm 1 Radiation Induced Noise Simulation Environment (Event-RINSE)

```
1: for Each time step  $t$  do
2:   Compute chance of radiation-induced noise using Eq.(3-1)
3:   if Generate Noise Event then
4:     Decide if noise is cluster or line using Eq.(3-2)
5:     Choose a random pixel  $[x_0, y_0]$ 
6:     if CLUSTER NOISE then
7:       Randomly sample a set of pixels  $[X, Y]$  in the neighborhood of  $[x_0, y_0]$ 
8:       for Each pixel  $\in [X, Y]$  in the cluster do
9:         Generate ON Events Using Eq.(3-3)
10:        Generate OFF Events Using Eq.(3-4)
11:      end for
12:    end if
13:    if LINE NOISE then
14:      Randomly sample angle of line:  $\theta \in [0, 2\pi)$ 
15:      Select a set of pixels  $[X, Y]$  forming a line L starting at  $[x_0, y_0]$  with angle  $\theta$ 
16:      for Each pixel  $[X, Y]$  of the line do
17:        Generate ON Events Using Eq.(3-3)
18:        Generate OFF Events Using Eq.(3-4)
19:      end for
20:    end if
21:  end if
22:  Append noise events to stream
23: end for
24: Sort events by ascending timestamps
```

Table 4: Summary of Event-RINSE runtime options.

Command Flag	Description	Datatype
-h/-help	Display help message and exit	N/A
-f/-input-file	The input data file path to read from	String
-o/-output-file	Custom output data file path to write to	String
-aoi/-angle of incidence	Angle of incidence between the sensor and simulated beam. Affects prevalence of lines vs. clusters	Integer
-s/-imgSize	The size of the images from the sensor data	List of 2 integers
-vi/-view-input	View the input data file as a video	N/A
-vo/-view-output	View the output data file as a video	N/A
-i/-inject	Perform injections on input file and write to output file	N/A
-d/-delta	Time-step to hold in one frame when viewing video	Float
-n/-noise	The event rate of noise with standard deviation	List of 2 integers

3.6 Conclusion

The purpose of this experiment was to irradiate an event-based camera under wide-spectrum neutrons to view and classify any SEEs that may be observed. The results show that the main SEU that affects the event-based camera is radiation-induced noise in the form of uniformly-distributed events across the sensor's field of view. We found that noise induced on single pixels resulted in both ON and OFF events with a ratio of 3:1. An average noise event rate was found to generate peaks with lags in the range of 8-10ms which corresponded directly with the macro-pulse patterns of the neutron source at LANSCE [92]. This pattern shows that the sensor acted like a naive particle detector, and was only affected by the radiation over short timescales. OFF events were also seen to follow the ON-event peaks with exponentially-decaying event-rate profile. These profiles seem to suggest that the neutrons interact with the photo-diode in individual pixels causing energy dumps leading to large photo-current, inducing the ON events in a short time period of about 1.6ms. The residual relaxation current after the radiation passes gives rise to the OFF events at much lower rates, but with a longer duration of up to 10 ms. The radiation did not cause any permanent, long-term damage to the sensor's photo-diodes or the hardware circuitry. This hypothesis was further confirmed when looking at the noise events in brighter and darker background-illumination conditions, where ON events were significantly higher in the dark environment due to sensor's higher contrast sensitivity but OFF events were not found to change significantly across the two conditions.

Focusing on induced noise, experiments were performed to observe correlations with the angle of incidence and the event rate through the sensor. Surprisingly, the null hypothesis that there is no correlation between the number of events and the angle of incidence, was supported. With a larger angle of incidence, the cross-sectional area of the sensor is smaller to the beam's point-of-view, making it less likely to be hit. When a neutron does impact the sensor, however, it travels across the field leaving a long streak of events following its trajectory. When there is a smaller angle of incidence, the sensor looks larger from the perspective of the beam. This apparent surface area implies that the sensor will be more likely to be hit, but events are shown only in the form of dots as short lines of neutrons

penetrate the sensor. These two effects thus cancel each other out, showing no difference in the induced event rate.

Comparing the number of events from a pendulum signal with radiation-induced noise shows a signal-to-noise ratio of 3.355. This ratio demonstrates the robustness of the event-based sensor to radiation in that the noise introduced does not significantly impact its ability to extract features of the desired signal. This result is further illustrated by the sensor’s ability to clearly observe the sinusoidal signal against the noisy background, and by the results of the optical flow algorithm implemented on the recorded events, which show no significant deterioration between the flow directions computed from the events when the radiation is introduced.

The results seen in this study, with random noise induced from radiation, is similar to other research performed on different optical electronic devices. Yates et al. [97] show that silicon-based video sensors have neutron sensitivity. Specifically, they show that spatially-random pixels show excitations, similar to the random “luminosity excitations” seen in the event sensor.

It has been found that transient effects of neutrons on optical electronics are directly correlated with the fluence of radiation [12]. This correlation agrees with the results shown in Figure 22 where the trajectory paths of the neutrons can be seen by excitations in the sensor. However, a direct correlation measurement between the fluence and the induced event rate would need to be performed in future research, as there was no control over the fluence in this experiment.

The results of this experiment also coincide with the measured radiation effects of traditional CMOS sensors. Sipos et al. [80] show that protons are detected in CMOS devices as bright white spots. This behavior is similar to the ON event burst seen in the event sensor. We hypothesize that a similar effect would be seen on the event sensor with proton radiation, however that would need to be verified with further research.

The Event-RINSE simulation environment created using the recorded noise data can be used to simulate the effects of radiation on pre-recorded data files. Event-RINSE was used to inject noise into the event streams recorded without radiation and was found to correspond well with the observed profile. The noise examples generated from the simulator matched

both the average single-event noise model and the average noise across the sensor. This fault injector makes it possible to test different neuromorphic-sensor algorithms, such as object tracking, under a noisy radiation environment without the need for expensive radiation testing, and thereby to assess an algorithm's viability in space or any noise suppression techniques. Future work could look at improving the parameters and probability models for more accurate noise generation.

Further development of event-cameras for space should include research into their efficacy under proton and heavy-ion radiation. These experiments will show if the sensor, as it currently stands, is capable of survival under the harsh conditions of space. Future work could also include testing the sensor's capability to perform basic object tracking under neutron irradiation. The noise shown in this experiment could pose a small problem for SSA by interfering with signal events in object tracking. However, since the noise was seen to be fairly constant under various cases, it could be modeled for background analysis. Also, the induced noise did not appear to deteriorate signal analysis enough to cause detrimental effects. With minor background suppression, the signal-to-noise ratio could therefore be improved enough to perform the necessary algorithms and analysis for SSA on future spacecraft. While the aim in this chapter is to understand the noise characteristics of the sensor in light of its application towards SSA, these neuromorphic sensors could potentially be used in many more applications beyond SSA. We plan to soon deploy one of these sensors on the International Space Station, where we will be better able to understand their noise resistance and sensing abilities in relation to space. Apart from being able to track space debris, they could also be used to provide low-latency feedback during docking procedures, extra-terrestrial landing scenarios, and unmanned automated flights in situation where the power budget is a limited constraint.

4.0 Reliability of Temporal Classification on Neuromorphic Processing

Given the results from the initial irradiation of the neuromorphic sensor, it is natural to extend this research into a neuromorphic classification application. By understanding how radiation effects impact the sensor and numerical computation, a reliable machine-learning algorithm can be designed for use on neuromorphic data. Therefore, there is one final piece of the data flow that needs to be analyzed: computations performed within common neuromorphic algorithms and architectures. Prior to performing matrix multiplications studied in Chapter 2, but after data is acquired from the sensor studied in Chapter 3, a neuromorphic architecture typically has many complex computations that also need to be protected from silent data errors.

Feature extraction is a fundamental part of object recognition in visual processing. A problem associated with this step is to understand how features should be characterized in an image. Moreover, as these complex algorithms move to hazardous environments such as space, their reliability to perform accurately needs to be considered and evaluated.

One feature extraction algorithm for neuromorphic, event-driven image classification is the Hierarchy of Event-Based Time Surfaces algorithm [44]. Neuromorphic event-driven sensors are a novel form of imaging device that offer a new paradigm of artificial vision, compared to traditional frame-based cameras. Recently, people have started to investigate using these event-driven sensors in space applications, whether it be on the ground in telescopes [23] or for use in space satellites [64, 75]. However, space environments are prone to far more radiative noise than most ground environments and typically involve additional thermal challenges [9]. Meanwhile, satellite designers need to use commercial-off-the-shelf processors, as opposed to traditional radiation-hardened processors in order to take advantage of their better performance and energy efficiency to make these complex machine-learning and computer-vision applications feasible to run onboard. Therefore, satellite designers typically perform additional tests on their hardware and software to ensure their system can survive in a radiative environment.

Before event-based sensors can be safely used in conjunction with computer-vision appli-

cations for onboard processing, the applications need to be tested under fault injection and radiation to ensure that the algorithm will perform accurately. Radiation can cause SEUs that can cause data errors, execution errors, or complete system failures, depending on what area of memory is upset [82]. Silent data errors could lead to incorrect decisions made by an autonomous system without any indication to the users of a failure [72,82]. Thus, understanding the possible failure modes of each application before deployment becomes vital to mission success.

This chapter evaluates the HOTS algorithm on classification of N-MNIST dataset [60] under a radiative environment and under simulation-based fault injections. Our experiments provide the initial evaluation of the reliability of HOTS as a case study to understand how different computations within a neuromorphic network are affected by these upsets. The results of this research provide designers the information needed to add protection to their mission, whether it be additional hardware protections or use of dependable computing techniques within software. This research also presents methods to create a more reliable neuromorphic architecture for use in hazardous environment given the observations of the fault injection and irradiation.

4.1 Background

This section provides the background needed to understand this research. This information includes event-driven sensors, the HOTS algorithm, and the use of radiation testing.

4.1.1 Neuromorphic Event-Based Sensors

Biomimetic, event-based sensors are a novel type of vision sensor that are modelled after the mammalian retina. They are made up of independent pixels that are driven by changes in light intensity in their field of view. This method provides an asynchronous stream of events that, in contrast to the traditional frame-based method of conventional cameras, also provide temporal context about the scene. The events can either have the polarity of ON, where

the light intensity of the pixel is increasing, or OFF, where the light intensity is decreasing. This local, asynchronous event method also provides the benefit of not containing a global shutter, providing neuromorphic sensors with a very high dynamic range. This large dynamic range means that bright sources will not over-saturate the entire field of view, allowing the sensor to still provide useful information. Similarly, since changes in light intensity are being measured, any static, redundant background information is not passed through the sensor, producing a very high temporal resolution with a low data rate [15].

Most neuromorphic sensors output visual information about a scene in the form of discrete, distinct events using Address-Event Representation (AER) [16, 45, 53]. AER encodes the visual information from the sensor in the form of 4-tuples containing the x- and y-pixel coordinates where the event occurred, the timestamp in microseconds, and the polarity of light intensity (x, y, t, p) .

4.1.2 HOTS algorithm

Since neuromorphic vision sensing is an entirely new paradigm of imaging, the corresponding computer vision applications need to be adapted to this new paradigm. Instead of working with frames, as traditional algorithms use, new methods needed to be developed to account for event streams. For image classification, the HOTS algorithm was developed [44]. HOTS uses the temporal context of a scene around an event to learn information about the object in the scene. Specifically, HOTS tries to answer the question of *how* features can be extracted from an event stream. To answer this question, Lagorce et al. introduced the concept of Time Surfaces [44].

Time Surfaces are the features that are extracted from an event stream. For each event in the stream, $e = [x, y, p, t]$, a square neighborhood of size R -pixels around the event pixel is chosen to create a Time Surface using the time of old events in the neighborhood. For each pixel in the neighborhood, past events are searched until an event of the same polarity as e is found. The difference between the time in neighborhood pixel and the current event is mapped using an exponential kernel to create a surface of temporal structures giving context to each event, which can then be used to extract features from a scene.

This process is repeated over larger spatial and temporal scales creating a hierarchical layered network. This network architecture thus consists of a “hierarchy” of Time Surfaces which builds and extracts features from a stream of events. Training this network architecture creates a model which can then be used as a pattern classifier on neuromorphic sensor data.

4.1.3 Fault Injection

Fault injection is the process of introducing controlled errors and noise in a system to study their impact on the resilience and performance of the system. For environments which are hard to experiment in, such as space, fault injection can be performed through modeling and simulations. For our experiments, we used two categories of fault injections, hardware and software. Hardware-based fault injection happens at a physical level, where faults are induced by disturbing the hardware from the environment, such as a radiation testing. This disturbance can manifest as faults in the low-level hardware components such state changes in transistors forming the memory structure of a processor that could lead to silent data errors [82].

Srouf and McGarrity [82] detail the effects of radiation on microelectronic circuits, such as damage, ionization, and single-event effects (SEUs) through radiation beam testing. Therefore, beam testing is popular in the field of space computing to classify SEUs of new systems and mitigation strategies. Knowledge of how an application will behave under radiation is critical to the design process for space missions, by giving an overview of the upsets or faults a system might encounter.

Software-based fault injection can be performed by injecting errors at a variety of levels of abstraction to study their effects from low-level assembly instructions to high-level meta instructions. In this study, we focus on the high-level abstraction. Namely, the injection used consists of directly changing elements of data in software to observe the response of the technique-under-test. The effect of radiation on event-driven sensors have been measured and modelled in previous studies [73]. Our high level fault injection model on HOTS can be used to complement the previous study allowing for analysis of the neuromorphic architecture starting from the input event data streams up to the final classifier layer of the HOTS

algorithm. Further details of the fault injections and their results are provided in section 4.4.2.

4.2 Related Work

This section discusses the related work on the topic of neuromorphic vision sensing. This include the use of neuromorphic sensing for space domain awareness (SDA), and previous works done in testing the reliability of neuromorphic sensors to radiation.

4.2.1 Traditional Feature Selection

Feature selection for object recognition is a fundamental problem in visual processing. Traditionally, features are defined as a function of the static information around a neighborhood in an image [46, 50]. The choice of the function determines the extracted features. Since conventional frame-based cameras primarily provide spatial information, the temporal dynamics of an image have no bearing on the received information. The drawback of this loss of dynamic information is that frames are captured at artificially timed intervals (also known as the frame-rate). These frames thus contain large amounts of redundant data if the background of an image does not change between frames.

Conventional feature extraction algorithms also typically assume that pixel illumination is the primary source of information. However, pixel luminance is not invariant to a scene [50]. Therefore, if any changes to the environment take place, the feature information will change, according to conventional methods. Similarly, the low dynamic range of conventional cameras means that accurately measuring the luminance becomes difficult [70].

In machine learning, the primary method of image recognition involves the use of convolutional neural networks (CNNs) ever since the Krihevsky et al. ImageNet Challenge win in 2012 [41]. What sets CNNs apart from other network types is that their layers make use of image convolution filters. Convolutional layers are composed by sliding different convolutional kernels across an image, projecting the information in the field onto a feature map.

This feature map then provides local perception. In other words, areas that are close to a pixel are considered more relevant than those far away [96]. Different convolutional kernels can provide different pieces of information, such as an edge detection kernel. The problem with CNNs on embedded systems are that they are computationally complex, and convolving a filter over a large image, which may have redundant background information, becomes infeasible [66].

4.2.2 Conventional SDA

Space domain awareness, previously known as space situational awareness, has been an important topic in military applications for many years [28, 38, 59, 63]. SDA is the ability to keep track of the surroundings of a satellite, such as objects or debris. Understanding the movement of the surrounding area enables mission operators to avoid collision and perform evasive maneuvers as needed. Vision systems with high-temporal resolution and low latency are required to accurately detect objects to be able to process in real time.

Traditionally, the main method used for SDA involves radio signals. For example, Ender et al. present the use of radar in SDA for collision detection, orbit estimation, and propagation [26]. The benefit of radar is that it has a very large coverage, meaning it can observe a wide area while having constant surveillance. However, since radar is bound by its long wavelength, this method only works for larger objects, while smaller objects remain invisible.

4.2.3 Neuromorphic Sensors for SDA

Since the technology is so new, the use of event-based sensors in space-related applications is not very well developed. Most of the work has been carried out in the context of terrestrial telescope observation of low-brightness objects in Low-Earth Orbit (LEO) and Geostationary-Earth Orbit (GEO). [68] covers a series of experiments on using neuromorphic sensors through a fish-eye lens for all-sky observation. They show that the neuromorphic sensors are robust to difficult observing conditions and fast-moving objects. [2] performed SDA of space imagery from a neuromorphic telescope using various tracking algorithms. With their collected neuromorphic data, they were able to track high-speed objects moving

across the field of view.

There has been extensive research into event-based cameras for real-time tracking and low-power computer systems. Many algorithms have been developed and evaluated for objects to be tracked within the visual space of an event-driven sensor. For example, Barranco et al. were able to use neuromorphic sensors for real-time clustering and multi-target tracking, receiving an F-accuracy of 95% while reducing the computational cost by 88% compared to the conventional frame-based method [19]. Reverter et al. developed a method for object tracking that can track many different shapes, so long as the pattern of the shapes is known a-priori [88]. Similarly, Lagorce et al. provide a multi-kernel Gaussian mixture model tracker for the detection and tracking of different shaped objects [43]. Another method that uses spatial matching to allow objects to be tracked even in occluded conditions [48]. The low computational requirements of neuromorphic sensor data analysis even allows tracking systems to be implemented on embedded platforms [49] and FPGAs [48], making them perfect for space applications. Novel methods can even detect and track objects in conditions where both the camera and the objects are moving independently, as is the case for satellites [55, 56, 69].

4.2.4 Neuromorphic Sensors Under Radiation

To perform valid fault injection on HOTS, a fault model is needed. The fault model developed in [73] was used to perform software fault injection into the data. Specifically, the introduced Event-based Radiation-Induced Noise Simulation Environment (Event-RINSE) was useful in adding radiation-induced noise to pre-recorded data streams. This added noise was then used as a fault-injector, allowing HOTS to be tested in a more controlled environment. In this experiment, HOTS was tested for its sensitivity to noise.

4.3 Scope and Methodology

This section will discuss the scope and methods to be used in the experiment. The methodology will cover software fault injection as well as the radiation experiment.

4.3.1 Scope

The operating system was a lightweight version of Linux, but the complications of Linux’s monolithic kernel will be irrelevant to the reliability of HOTS or the Time Surfaces. Moreover, the operating system hosting the HOTS algorithm is arbitrary, so this experiment does not focus on execution errors. That is, any errors involving segmentation faults, kernel panics, or similar errors are not studied as these depend on the underlying operating system.

This experiment primarily focuses on silent data errors. Silent data errors are defined here as any error causing a difference in the output of each layer. Each layer was individually analyzed and compared with the final predictions of the network. Observing the outputs of each layer enables observation on how sensitive the Time Surfaces are to noise. This observation also shows how robust the output predictions are to changes in the noise.

4.3.2 Methodology

This experiment took place in two distinct phases: the radiation experiment and the high-level fault injection. These two halves provide two separate measurements of reliability. Namely, the radiation experiment shows how robust the network architecture is to data errors, and the software fault injection shows how sensitive the HOTS algorithm is to noise in the input data and corruptions in the algorithm.

4.3.2.1 Radiation Experiment

The radiation experiment were performed at Los Alamos Neutron Science Center (LAN-SCE) Weapons Neutron Research (WNR) facility. LANSCE provides a wide-spectrum neutron beam of energies ranging from ~ 1 MeV to > 10 MeV. Two Digilent PYNQ-Z2 boards,

labelled in this research as PYNQ 0 and PYNQ 1 (Fig. 29), were used as the device-under-test (DUT) and were radiated for approximately four days. The effective fluence on the DUTs was calculated by summing up the number of estimated neutrons passing through the DUT while it was powered, and dividing by a constant that drops off with the square of the distance from the dosimeter to account for any decay or absorption of the beam before reaching the target. This constant is defined in [91] as $\frac{(13.87m)^2}{(13.87m+d)^2}$, where $13.87m$ is the distance from the beam source to the dosimeter and d is the distance from the dosimeter to the DUT. In this experiment, d was $0.911m$ for PYNQ 0, and $0.937m$ for PYNQ 1.

The Neuromorphic Modified National Institute of Standards and Technology (N-MNIST) dataset, which consists of neuromorphic event streams of hand-written numerical digits (0-9), was used to test image classification with HOTS. Only one digit class was tested at a time to separate how data errors affect different classes, as opposed to using an average classification accuracy among all digits. The outputs of each layer as well as their respective Time Surfaces were logged for each execution. Three layers of Time Surfaces were used with a neighborhood size of $7 \times 7px$, $13 \times 13px$, and $25 \times 25px$ around each event pixel. This output helps determine which layers are most sensitive to radiation effects. Similarly, the results show how robust the prediction outputs are to data errors. If there is an error in Layer 1, but the prediction accuracy does not change, this shows that the layer network has intrinsic reliability associated with it. Any errors in the execution of the application, such as segmentation faults or kernel panics, were not included in this analysis. Results of the radiation experiment are discussed in Section 4.4.1.

4.3.2.2 Fault Injection Experiment

Noise was added to the N-MNIST input data using the Event-based Radiation-Induced-Noise Simulation Environment (Event-RINSE) described in [73]. This noise model was created using actual radiation of an ATIS and is used here to insert noise into the N-MNIST event data. The noise was injected into the input data using a variable noise-event rate. Changing the event rate can be used to find how much noise needs to be added to affect the prediction accuracy. Fault injecting in software provides a much more controlled environment

to test the reliability of HOTS and the Time Surface’s sensitivity to noise. The fault injection results are discussed in Section 4.4.2

The details of the fault injection can be seen in Figure 28. Each aspect of the algorithm was tested under fault injection using a simple Bernoulli test with a varying probability from a 10% to 90% chance of injection. The targets include the time surface exponential values, the input time context, and the time decay constant all highlighted in Figure 28. All targets were injected into in separate events to ensure a controlled response that simulates a single-event upset in memory.

4.3.3 Platform Selection

We used two separate platforms for our experiments. The radiation experiment was performed on the Digilent PYNQ-Z2 board, employing a Xilinx ZYNQ-7020 System-on-Chip (SoC) which has a dual-core ARM Cortex-A9 processor and an Artix-7 FPGA fabric. This SoC acts analogously to the CSP space computer developed at the NSF Center for Space, High-Performance and Resilient Computing (SHREC), which has been employed on multiple platforms aboard the international space station [76, 78, 94]. SHREC also uses the CSP alongside a neuromorphic sensor on their mission, Configurable & Autonomous Sensor Processing Research, which is the primary reason the PYNQ is used as a device-under-test [64, 75].

Since high-level fault injection was performed in software, the platform becomes irrelevant. Therefore, for software fault injection, a desktop computer, employing an AMD Ryzen 5 3600X 6-core processor, was used. This platform is arbitrary and will not affect the experimental results.

4.4 Results

This section covers the results of the radiation experiment and insights from fault injection. HOTS vulnerabilities are discussed herein.

4.4.1 LANSCE Experiment

The primary purpose of the LANSCE radiation experiment is to measure the cross-section of the HOTS algorithm on the Xilinx Zynq-7020 SoC. The cross-section is defined as the vulnerable area on the chip, where if impacted by radiation, an error event is expected. However, the error events need to be defined in such a way that provides insight into the vulnerability of the algorithm. Error events were defined as differences from the calculated and expected Time Surface for every event in each of the three layers. The cross-section, along with the 95% confidence interval, was calculated for each layer where an error event first arose, shown in Figure 29(A). Similarly, the total cross-section and 95% confidence interval for all combined layers can be seen in Figure 29(B).

The 95% confidence interval seems to increase with each layer, as expected, since the Time Surface is larger in higher layers, leading to a higher chance of vulnerability. It is important to note, however, that despite these data error events, there were no events that cause a loss in prediction accuracy. Therefore, it is seen that HOTS shows an intrinsic reliability as an application. Between the several feature points in a Time Surface, as well as the intrinsic reliability within the classifier, a small number of elements being affected in the Time Surface will most likely not cause any loss of accuracy. This result is consistent with [72], which showed intrinsic reliability within neural-network classifiers.

Furthermore, we classify the types of errors observed in the radiation experiment. There are three types of data errors that were observed:

1. Silent data errors where the Time Surface amplitude falls within $[0, 1]$, implying the error occurs before the exponential calculation
2. Silent data errors where the Time Surface amplitude does not fall within $[0, 1]$, implying the error occurs after the exponential calculation
3. Data errors where values are missing from the output Time Surface, implying the error occurs within the parameters of the architecture

Figure 30(A) demonstrates an example of error type 1, where silent data errors occurred before the exponential decay calculation in the Time Surface generation. The residuals show two points in the observed, flattened Time Surface that differ from the expected result. Both

cases are constrained within the range $[0, 1]$, which implies that the error caused an erroneous t value to be used in the calculation of the exponential decay, $e^{\frac{(t-t_0)}{\tau}}$.

Figure 30(B) shows an example of a much more extreme data error. This figure shows cases of both error type 2 and 3. At around index 50, the amplitude of the Time Surface jumps to about 5000, well out of the normal range of $[0, 1]$ constrained by the exponential calculation. Therefore, it can be assumed that the error occurs in the value of the Time Surface after the exponential decay, and not any of the data or parameters beforehand. Similarly, some values of the expected Time Surface were missing from the observed one, causing a constant offset from the expected pattern after index 100. Surprisingly, neither of these error events caused a drop in prediction accuracy, probably due to the fact that this Time Surface only represents one of many features in an event stream.

Figure 30 (C) shows a histogram of the observed time offsets from the data errors. The majority of time offsets were small, falling within ± 2 seconds. This small offset implies that the majority of data errors occurred before the exponential calculation, keeping the erroneous output bound within $[0, 1]$, most likely due to the amount of computation time spent on the Time Surface exponential calculations. The majority of the runtime is spent calculating the Time Surface exponential values, leaving the data used in the calculation vulnerable to radiation. However, this bound from the calculation means that the majority of output errors will not be significant enough to cause a change in the clustering output.

4.4.2 Fault Injection

The results from the Event-RINSE fault injection can be seen in Figure 31. The noise model for Event-RINSE consists of a burst of ON events followed by a long relaxation period of OFF events, meaning that there are several hundred induced event spikes in each noise event occurrence [73]. Induced noise rates were tested from 0 to 100 noise occurrences per second. The cumulative average among all 10 classes of N-MNIST was shown to drop to random chance at around 80 noise occurrences per second, seen in Figure 31(A). However, even small amounts of radiation noise, such as 5 effects per second causes a drop in accuracy, implying that the system is sensitive to variance in the input data. This sensitivity follows

due to the creation of many random noise features created that will interfere with the classification, especially on a memory-constrained embedded system where not all events in an input file can necessarily be used for feature extraction.

Figure 31(B) separates the classes for digit “1” and digit “5” and compares them to the cumulative average. Classes with a diverse set of features, such as digit “5” show a higher resilience to the average, typically scoring around one standard deviation above the average. Conversely, classes without many features to extract, such as digit “1”, consistently had an accuracy lower than one standard deviation below the average. This is most likely due to the random noise having a larger impact on the classification when there are fewer features to extract from a stream.

To control the response of high-level fault injections, four separate variables were individually targeted using software bit-flips with varying probabilities of injection. The results of the software fault injection effects on the accuracy of HOTS classification compared to its accuracy without injection can be seen in Figure 32. Injection into all four variables, the exponential, polarity, decay constant, and event time caused the accuracy to drop to random chance (10% accuracy) by 50% injected events. The exponential and the decay constant cause the sharpest decline in accuracy where the accuracy drops to below 25% even for 10% affected events. This is expected due to the fact that they drastically change the shape of the Time Surface feature. Surprisingly, injecting into the polarities shows the slowest drop in accuracy despite directly affecting the k-means center clustering. However, this phenomenon could be due to the fact that the polarity injections only lead to re-assignment from the original polarity to another polarity bounded within the number of clusters in each layer as used during training, leading to a lower probability of a data error occurring.

4.5 Conclusion

The results of our experiments demonstrate the reliability and vulnerabilities of feature extraction in neuromorphic data. The response of the Hierarchy of Time Surface architecture to radiation and the response of Time Surface features to targeted fault injection were

measured. Within the radiation experiment, we saw no change in accuracy due to any data errors observed within the Time Surface layers. The classifier used the output of the hierarchy of Time Surfaces, which means that only errors propagating to the output of the hierarchy could have any possible effect on the accuracy. However, as observed in [72], multi-layered perceptrons have intrinsic resilience to data errors within the nodes, providing an overall boost to the reliability of the HOTS architecture. It is also likely that the large amount of features provided by creating a Time Surface for every event in a data file compared to the low chance for a malicious error in memory to occur during computation gives a higher resilience to a layer being affected.

Within the data errors, the time offsets were then calculated, shown in Figure 30(C). This figure shows a histogram of how radiation tended to shift the temporal spikes calculated through inversion of the exponential function. The majority of the calculated offsets are positive after the natural logarithm of the exponential multiplied by the negative of the decay constant. This provides evidence that the majority of the errors encountered in the radiation test occurred before or during the calculation of the exponential. This is most likely due to the fact that the creation of the time surfaces is the most computationally complex part of the algorithm. Once the Time Surfaces, and thus the exponential values, are calculated, the rest of the inference is quick, leaving little chance for a radiation strike to corrupt the data. The most time consuming parts of an algorithm will generally be the most vulnerable to radiation since the time where an error may occur is larger.

To combat the randomness of a radiation experiment, controlled fault injection in software was also performed to find the most vulnerable parts of the HOTS architecture. Event-RINSE allows observations on how the classification accuracy decreases when there is corruption in the sensor data. Data errors within the algorithm can be detected or corrected via redundancy, but it becomes difficult to detect problems if the sensor providing the input itself is affected. Figure 31(A) shows that the accuracy of HOTS does decrease with increased radiation noise within the input event stream. However, Figure 31(B) shows that the drop in accuracy is dependent on the feature map of the data and is, therefore, data dependent. It is likely that the radiation response would be different under a different dataset. It should also be noted that due to the memory and computational constraints of embedded systems,

all events in an input file were unable to be used to create Time Surfaces. Increasing the number of events used in each sample would most likely provide more resilience to input noise by increasing the signal-to-noise ratio. known fault injection models, simply knowing the radiation noise model opens up opportunities for filtering or noise reduction before processing.

High-level fault injection shows how faults within onboard memory or cache can propagate to data errors. The primary targets for data errors within the architecture would be the exponential amplitude, decay constant τ , the event time spike t , and the polarity used in clustering. To understand the slope of accuracy decay, the accuracy with fault injection was compared relative to the accuracy without fault injection. All four variables were shown to drop the inference accuracy very quickly. The exponential value and the decay constant were shown to have the sharpest decline in accuracy with the probability of injection. This sharp decline is most likely due to those variables drastically changing the overall feature shape of the Time Surface which would lead to significant changes in clustering. The decay constant significantly impacts the shape of the feature as it affects all points in the output as opposed to one pixel in the Time Surface. Meanwhile, injections in the exponential were post-calculation of the exponential amplitude, meaning the erroneous value was not bounded by $[0, 1]$ causing severe changes in the feature shape, such as is the case in Figure 30(B). However, the radiation experiment showed that this case is unlikely. Interestingly, the polarity which defines the features used for classification had the least sharp decline in accuracy. To prevent crashes, the polarity bit-flips had to be bound to the number of features in each layer, which means there is a lower probability of the injected polarity being different from the expected one. With an actual radiation experiment, it is much more likely that a bit-flip in the polarity variable would cause a segmentation fault which were not accounted for in this experiment as they are platform dependent. The event time shows the second shallowest decline of the accuracy next to the polarity. The event time, as it is used in the exponential values, would not lead to as strong a response as the others due to the fact that the exponential calculation masks the error by keeping it bound within $[0, 1]$ and thus does not cause as severe changes in the output, similar to Figure 30(A).

In all variable cases, however, the accuracy still drops to random chance before a 50%

probability of injection, demonstrating a severe vulnerability to SEUs in these components. To combat this, traditional reliable computing techniques can be used. In the case of the exponential, a solution to prevent vulnerability at the cost of memory or runtime can be to calculate the value three or more times and take a majority vote on the output, similar to triple modular redundancy [52]. Another option for error detection is to have a quick boundary check along the time surface. Due to the nature of the exponential, any values that are larger than 1 or less than 0 would be impossible and thus would be a data error. Similarly, the polarity calculation can also be performed multiple times and put to a vote. Multiple copies of the decay constant can be stored and voted on before any computation, allowing for redundancy with minimal time and memory overhead. Unfortunately, there is no ground truth to the event time spike during computation, so any redundancy would have to occur in the input data files. However, the probability of any of these values being impacted by radiation to cause a decline in accuracy is small. Though, adding additional redundancies and checks in these variables is a possible way to improve reliability for mission-critical software with little overhead.

The overall results of these experiments can be used to infer the response to radiation of other neuromorphic applications, such as any method using an exponential kernel. With the radiation experiment, it is shown that there is intrinsic reliability within the HOTS application, where no decline in inference accuracy is seen, even with errors in the layers. Fault injections show how propagating computational errors can lead to failures in inference. Both of these experiments together can give mission designers a starting point to creating the architectures for reliable, neuromorphic applications.

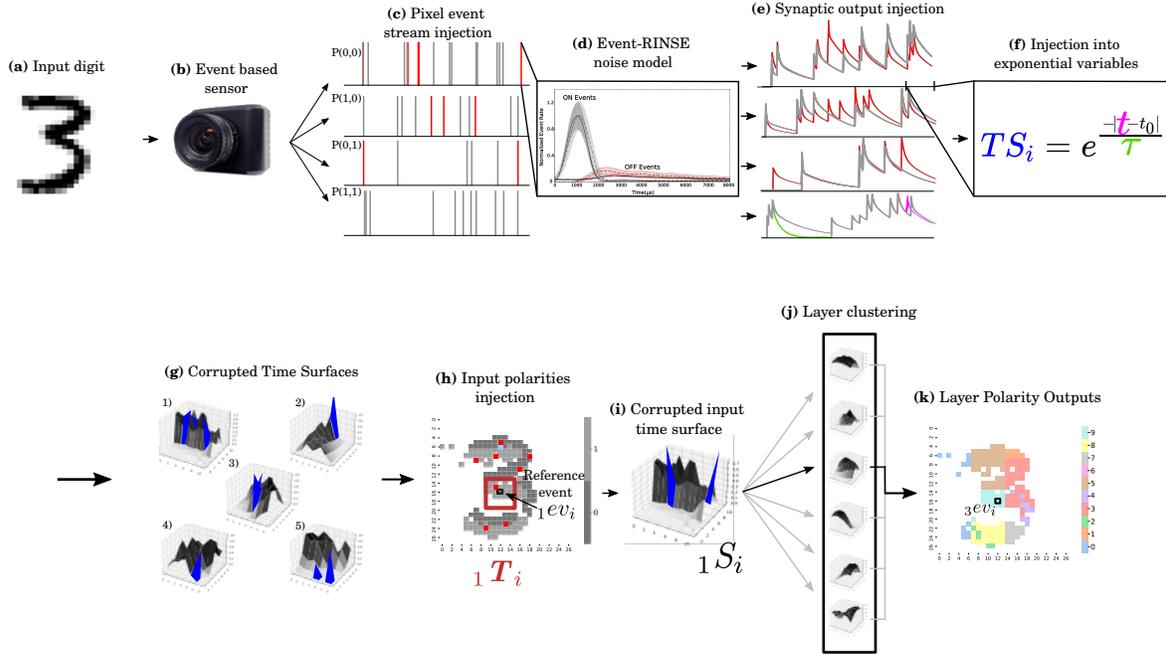


Figure 28: Fault injection into HOTS classification architecture. Noise was injected into different modules of the HOTS algorithm independently to test their resilience. An input digit, (a), is passed through the event-based sensor, (b), which produces an asynchronous event stream for each pixel. Radiation based noise injected into input events, (c), modelled using Event-RINSE fault injector, (d), described in [73], leads to erroneous spikes (in red) added to the event stream. This also leads to errors in time surface computation as shown in top three rows in (e). The bottom row of (e) further shows errors due to additional noise injected into exponential kernel computation, (f), such as the decay constant, τ (green) and time of event, t (in magenta) both of which lead to error in the decayed synaptic output. The Time Surfaces themselves are also prone to noise (blue) which lead to erroneous time surfaces values for certain pixels shown in (g). The polarities used in k-means clustering were also injected with bit-flips, shown in (h) to see how random changes in cluster assignments affect the classification. The layer outputs, i.e. the labeled events, were then passed through a multi-layered perceptron as features for classification demonstrated in (k).

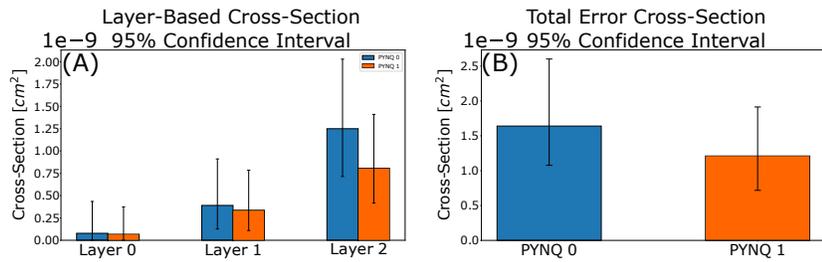
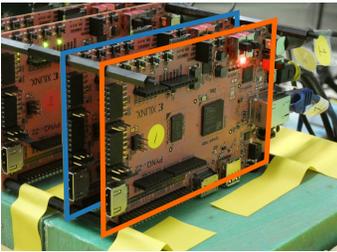


Figure 29: Two PYNQ Boards radiated at LANSCE labelled as PYNQ 0 (blue) and PYNQ 1 (orange). (A) The cross-sections when errors are separated by which layer the first error occurs. (B) The total cross section containing the total number of errors in all layers.

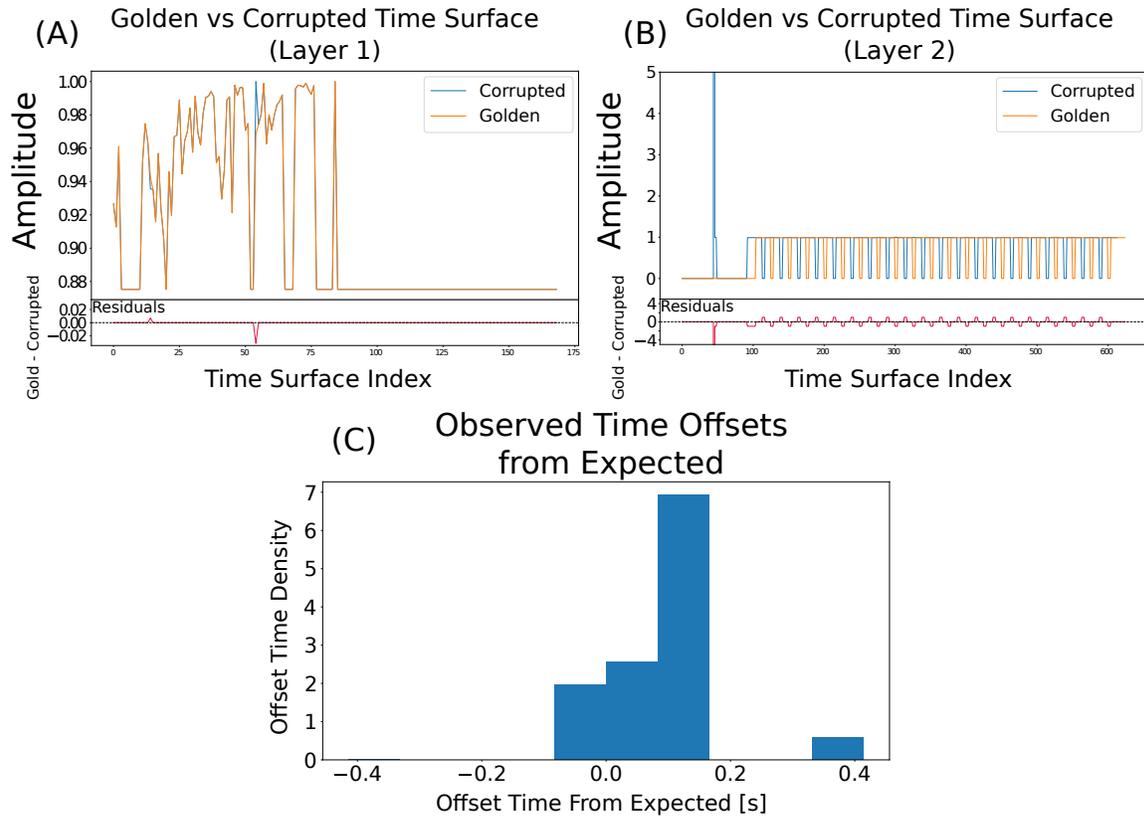


Figure 30: Examples of data errors seen in Time Surfaces during irradiation. Time Surfaces were flattened from 2 dimensions to 1 dimensions for easier visibility. (A) Corrupted and golden flattened Time Surfaces with residuals to depict an example of a data error occurring before the exponential calculation at around index 13 and 54. (B) Corrupted and golden flattened Time Surfaces with residuals to depict a data error occurring after the exponential calculation at around index 50, breaking out of the exponential bound of $[0, 1]$. A missing value occurs before index 100, causing an offset for the rest of the Time Surface. (C) Histogram of time offset errors observed in radiation test showing how the temporal spike was changed. Most results fell within small time differences, implying the errors occurred before the exponential calculation.

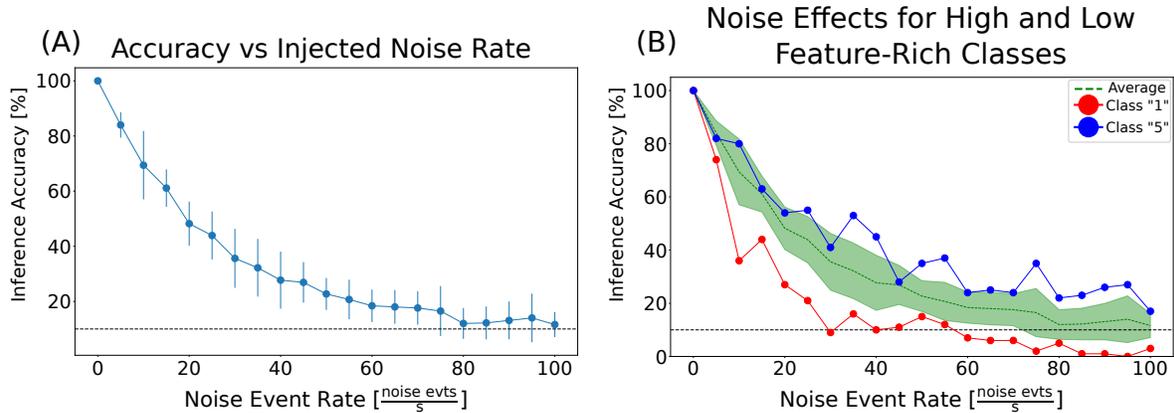


Figure 31: (A) Average accuracy of HOTS with noise introduced by the Event-RINSE fault injector at varying levels of radiation-induced-noise event rates. A single noise event is defined by the Event-RINSE model described in [73], or seen in Figure 28(d). The accuracy drops to random chance at a noise rate of around $80 \frac{\text{noise evts}}{\text{s}}$. (B) Accuracy of class 1 and class 5 for different noise rates. Class 1 (red), with very few distinct features typically falls below the one standard deviation of the average (in green) and reaching random chance by $30 \frac{\text{noise evts}}{\text{s}}$. This steeper curve implies that the noise effect has a larger effect for classes with fewer distinctive features. However, class 5 (blue), with more distinct features typically performed better at higher noise rates than the average.

Effects of Targeted Fault Injection on Relative Accuracy

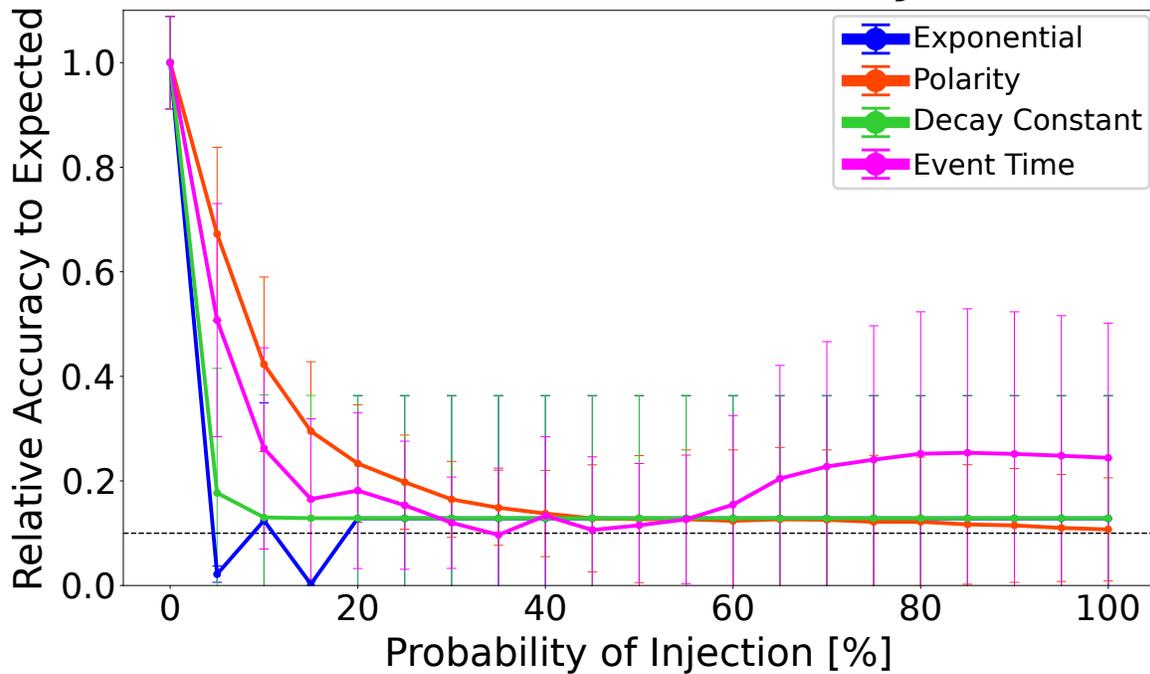


Figure 32: Average relative accuracy across classes given different probabilities of bit-flips in different variables: the value of the exponential (blue) the polarities used in k-means clustering (orange), the decay constant τ (green), and the times of input events (red). All variables show strong reactions to bit-flips, causing the accuracy to drop to random chance with the exponential value and the decay constant showing the sharpest slope.

5.0 Conclusions and Future Work

Neuromorphic systems aim to change the paradigm of computing from the spatial domain to a temporal domain. Taking advantage of the energy efficiency of asynchronous processing and sensing provides opportunity for performing complex applications, such as ML, on embedded space platforms. However, before any neuromorphic systems are flown, their failure modes must be understood to ensure mission success. Moreover, even after they are flown, dependable-computing techniques are required to ensure the data being processed is accurate to avoid any silent data errors. In this dissertation, we provided the first measurements of the reliability of neuromorphic sensors and common computations to design a neuromorphic fault model. Similarly, we presented ABFT as a valuable method to improve data reliability in any applications that use matrix multiplication, such as ML. These three pieces together created an end-to-end reliable neuromorphic classification system.

Chapter 2 presents the use of ABFT as a method of improving data reliability in ML experiments. Many ML applications, such as HOTS, uses matrix multiplication as their primary kernel. Therefore, since the reliability of matrix multiplication can be improved with ABFT, the entire application will become more dependable regardless of the underlying architecture or sensing technology used.

Chapter 3 addresses the reliability of neuromorphic vision sensors by radiating the sensor for the first time. The sensor saw radiation as induced noise events when neutrons transferred energy to the photodiodes of a circuit. A very specific model of ON events and OFF events was measured for each noise instance. Using this model, a fault injector named Event-RINSE was created that can demonstrate the same failure modes on pre-recorded data. This injector allows designers to test their applications on radiation noise before deployment without the need of expensive radiation testing.

Finally, Chapter 4 pulls the previous two together by evaluating a neuromorphic classification algorithm via radiation and fault injection. These injections demonstrated how the use of temporal-context encoding will perform in a radiative environment, along with other computations commonly needed in neuromorphic architectures. The HOTS algorithm was

evaluated under radiation and fault injection as a way to encode temporal information to perform image classification. HOTS was used as a case study to understand how different computations within a neuromorphic architecture are affected by these upsets. These experiments provide designers the information needed to add protection to their mission, whether it be additional hardware safeguards or dependable-computing techniques within software. These experiments also present methods to create more reliable neuromorphic architectures for use in hazardous environments given the observations from fault injection and irradiation campaigns.

This work is just the first step into space with neuromorphic systems. However, there are still many facets yet to be explored. For example, performing a test on a neuromorphic vision sensor with proton or heavy-ion radiation is a natural progression from the wide-spectrum neutron experiment in this work. This experiment would show how much energy can be introduced into the sensor before it fails entirely. Based on the beam measurement results found in this neutron experiment, the opportunities for using neuromorphic sensors as particle detectors can be explored, either as a trajectory mapper or giving a measurement of the surrounding radiation environment. Similarly, new neuromorphic algorithms could be created for use in various space applications.

An interesting extension of the neuromorphic algorithm research could be to perform fault injection on various architectures of processors. This extension would move from software-based fault-injection abstraction to hardware, providing a more realistic representation of the failure modes, albeit while being dependent on specific hardware. Various neuromorphic processors could also be brought to a beam for radiation testing to understand how they behave under hardware fault injection. A radiation test would evaluate how accurate and realistic the fault model assumed in software fault injection is. The ultimate test for how a neuromorphic processor would behave in space would be to simply fly one on a spacecraft and observe its performance.

Finally, ABFT could be applied to any ML application, whether neuromorphic or not, to observe how it directly improves reliability and affects the performance. An interesting extension to this research would be to tackle execution errors, such as segmentation faults or kernel panics within a system, since ABFT only handles data errors. This kind of extension

could involve observing how different operating-system kernels respond to radiation effects or identifying the primary cause of execution errors. Regardless of the direction, there are certainly many opportunities for expanding neuromorphic systems to the space environment.

Bibliography

- [1] Saeed Afshar, Andrew Nicholson, Andre van Schaik, and Gregory Cohen. Event-based object detection and tracking for space situational awareness. *IEEE Sensors Journal*, PP:1–1, 07 2020.
- [2] Saeed Afshar, Andrew Peter Nicholson, Andre Van Schaik, and Gregory Cohen. Event-based object detection and tracking for space situational awareness. *IEEE Sensors Journal*, 20:15117–15132, 12 2020.
- [3] H. Akolkar, S. H. Ieng, and R. Benosman. Real-time high speed motion prediction using fast aperture-robust event-driven visual flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(1):1–12, 2020.
- [4] Himanshu Akolkar, Cedric Meyer, Xavier Clady, Olivier Marre, Chiara Bartolozzi, Stefano Panzeri, and Ryad Benosman. What can neuromorphic event-driven precise timing add to spike-based pattern recognition? *Neural computation*, 27:561–593, 03 2015.
- [5] Ahmad A Al-Yamani, Nahmsuk Oh, and Edward J McCluskey. Performance evaluation of checksum-based abft. In *Proceedings 2001 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 461–466. IEEE, 2001.
- [6] David Alexander. Transient ionizing radiation effects in devices and circuits. *Nuclear Science, IEEE Transactions on*, 50:565 – 582, 07 2003.
- [7] Jordan D Anderson, Jennings C Leavitt, and Michael J Wirthlin. Neutron radiation beam results for the xilinx ultrascale+ mp soc. In *2018 IEEE Radiation Effects Data Workshop (REDW)*, pages 1–7. IEEE, 2018.
- [8] Cynthia J. Anfinson and Franklin T. Luk. A linear algebraic model of algorithm-based fault tolerance. *IEEE Transactions on Computers*, 37(12):1599–1604, 1988.
- [9] G. D. Badhwar and P. M. O’Neill. An improved model of galactic cosmic radiation for space exploration missions. *International Journal of Radiation Applications and Instrumentation. Part D. Nuclear Tracks and Radiation Measurements*, 20:403–410, 7 1992.

- [10] G.D. Badhwar and P.M. O'Neill. An improved model of galactic cosmic radiation for space exploration missions. *International Journal of Radiation Applications and Instrumentation. Part D. Nuclear Tracks and Radiation Measurements*, 20(3):403–410, 1992.
- [11] Marta Bagatin and Simone Gerardin. *Ionizing Radiation Effects in Electronics: From Memories to Imagers*. CRC press, 2018.
- [12] Jacques Baggio, Martial Martinez, Claudine D'hose, and Olivier Musseau. Analysis of transient effects induced by neutrons on a ccd image sensor. In *Photonics for Space and Radiation Environments II*, volume 4547, pages 105–115. International Society for Optics and Photonics, 2002.
- [13] Jon A Benediktsson, Philip H Swain, and Okan K Ersoy. Neural network approaches versus statistical methods in classification of multisource remote sensing data. 1990.
- [14] R. Benosman, Sio-Hoi Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan. Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37, 2012.
- [15] Ryad Benosman, Sio Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37, 3 2012.
- [16] K. A. Boahen. Point-to-point connectivity between neuromorphic chips using address-events. *IEEE Transactions on Circuits and Systems*, 47-5:416 – 434, 2000.
- [17] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [18] L. A. Camuñas-Mesa, T. Serrano-Gotarredona, S. Ieng, R. Benosman, and B. Linares-Barranco. Event-driven stereo visual tracking algorithm to solve object occlusion. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4223–4237, 2018.
- [19] Luis A. Camunas-Mesa, Teresa Serrano-Gotarredona, Sio Hoi Ieng, Ryad Benosman, and Bernabe Linares-Barranco. Event-driven stereo visual tracking algorithm to solve object occlusion. *IEEE Transactions on Neural Networks and Learning Systems*, 29:4223–4237, 9 2018.
- [20] Edward Carlisle IV and Alan George. Dynamic robust single-event upset simulator. *Journal of Aerospace Information Systems*, 15(5):282–296, 2018.

- [21] Zizhong Chen. Online-abft: An online algorithm based fault tolerance scheme for soft error detection in iterative methods. In *ACM SIGPLAN Notices*, volume 48, pages 167–176. ACM, 2013.
- [22] Posch Christoph, Matolin Daniel, and Wohlgenannt Rainer. A QVGA 143 db dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2011.
- [23] Gregory Cohen, Saeed Afshar, and André Van Schaik. Approaches for astrometry using event-based sensors. *Advanced Maui Optical And Space Surveillance Technologies Conference (AMOS)*, 2018.
- [24] Peng Du, Aurelien Bouteiller, George Bosilca, Thomas Herault, and Jack Dongarra. Algorithm-based fault tolerance for dense matrix factorizations. *Acm sigplan notices*, 47(8):225–234, 2012.
- [25] Mohamed El Mashade, Manal Ashry, Sh Eladl, and M.S. Rageh. Experimental measurements of some optoelectronic devices before and after gamma irradiation. 3:1–12, 01 2004.
- [26] Joachim Ender, Ludger Leushacke, Andreas Brenner, and Helmut Wilden. Radar techniques for space situational awareness. In *2011 12th International Radar Symposium (IRS)*, pages 21–26, 2011.
- [27] Giovanni Gasparini and Valérie Miranda. Space situational awareness: an overview. 01 2010.
- [28] Giovanni Gasparini and Valérie Miranda. Space situational awareness: an overview. *Studies in Space Policy*, 4:73–87, 2010.
- [29] Robert Granat, Kiri L Wagstaff, Benjamin Bornstein, Benyang Tang, and Michael Turmon. Simulating and detecting radiation-induced errors for onboard machine learning. In *2009 Third IEEE International Conference on Space Mission Challenges for Information Technology*, pages 125–131. IEEE, 2009.
- [30] Kaoru Hagiwara, K Nakamura, K Hikasa, M Tanabashi, M Aguilar-Benitez, C Amsler, RMichael Barnett, O Dahl, DE Groom, R Miquel, et al. Review of particle physics. *Physical Review. D, Particles Fields*, 66(1):010001–010001, 2002.

- [31] Chi-Chang J Ho and N Harris McClamroch. Automatic spacecraft docking using computer vision-based guidance and control techniques. *Journal of Guidance, Control, and Dynamics*, 16(2):281–288, 1993.
- [32] Kuang-Hua Huang and Jacob A Abraham. Algorithm-based fault tolerance for matrix operations. *IEEE transactions on computers*, 100(6):518–528, 1984.
- [33] Bratislav Iricanin and Dejan Gvozdic. The analytic approach in the modelling of one-dimensional electron concentration distribution in some two-valley semiconductor electron devices. *Journal of Physics A: Mathematical and General*, 31:2997, 01 1999.
- [34] Ender Joachim, Leushacke Ludger, Brenner Andreas, and Wilden Helmut. Radar techniques for space situational awareness. In *2011 12th International Radar Symposium (IRS)*, pages 21–26. IEEE, 2011.
- [35] DeMars Kyle Jordan. *Nonlinear orbit uncertainty prediction and rectification for space situational awareness*. PhD thesis, The University of Texas at Austin, 2010.
- [36] Eure Kenneth, Belcastro Celeste M, Koppen Daniel, Hess Richard, Stange Kent, Vahala Linda, and Ferguson Matthew. Closed-loop neutron particle effects testing on a recoverable flight control computer. In *The 23rd Digital Avionics Systems Conference (IEEE Cat. No. 04CH37576)*, volume 2, pages 6–. IEEE, 2004.
- [37] J.A. Kennewell and Ba-Ngu Vo. An overview of space situational awareness. pages 1029–1036, 01 2013.
- [38] John A Kennewell and Ba-Ngu Vo. An overview of space situational awareness. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1029–1036, 2013.
- [39] Israel Koren and C Mani Krishna. *Fault-tolerant systems*. Elsevier, 2010.
- [40] Z. Kovalyuk, V. Katerynychuk, OA Politanska, O. Sydor, and Volodymyr Khomyak. Effect of gamma radiation on the properties of inse photodiodes. *Technical Physics Letters*, 31:359–360, 05 2005.
- [41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

- [42] X. Lagorce, S.-H. Ieng, X. Clady, M. Pfeiffer, and R. Benosman. Spatiotemporal features for asynchronous event-based data. *Front. Neurosci.*, 9, 2015.
- [43] X. Lagorce, C. Meyer, S. H. Ieng, D. Filliat, and R. Benosman. Asynchronous event-based multikernel algorithm for high-speed visual features tracking. *IEEE Trans. Neural Netw. Learn. Syst. (TNNLS)*, 26:1710–1720, 2015.
- [44] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1346–1359, 7 2017.
- [45] J. Lazzaro and J. Wawrzynek. A multi-sender asynchronous extension to the AER protocol. In *Conference on Advanced Research in VLSI*, 1995.
- [46] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2323, 1998.
- [47] P. Lichtsteiner, C. Posch, and Tobi Delbruck. A 128×128 120 dB 15 μ s Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.
- [48] A. Linares-Barranco, F. Gómez-Rodríguez, V. Villanueva, L. Longinotti, and T. Delbruck. A USB3.0 FPGA event-based filtering and tracking framework for dynamic vision sensors. *IEEE Int. Symp. Circuits and Systems (ISCAS)*, 2015.
- [49] M. Litzenberger, C. Posch, D. Bauer, A. N. Belbachir, P. Schon, B. Kohn, and H. Garn. Embedded vision system for real-time object tracking using an asynchronous transient vision sensor. *IEEE 12th Digital Signal Proc. Workshop and 4th IEEE Signal Proc. Education Workshop*, pages 173–178, 2006.
- [50] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 2004 60:2*, 60:91–110, 11 2004.
- [51] Franklin T Luk and Haesun Park. An analysis of algorithm-based fault tolerance techniques. *Journal of Parallel and Distributed Computing*, 5(2):172–184, 1988.

- [52] Robert E Lyons and Wouter Vanderkulk. The use of triple-modular redundancy to improve computer reliability. *IBM journal of research and development*, 6(2):200–209, 1962.
- [53] M. A. Mahowald. *VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function*. PhD thesis, California Institute of Technology, 1992.
- [54] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18(1):50–60, 1947.
- [55] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos. Event-based moving object detection and tracking. *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2018.
- [56] Z. Ni, A. Bolopion, J. Agnus, R. Benosman, and S. Regnier. Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics. *IEEE Trans. Robot.*, 28:1081–1089, 2012.
- [57] Z. Ni, S. H. Ieng, C. Posch, S. Regnier, and R. Benosman. Visual tracking using neuromorphic asynchronous event-based cameras. *Neural Computation*, 27:925–953, 2015.
- [58] Daniel Oltrogge and Salvatore Alfano. The technical challenges of better space situational awareness and space traffic management. *Journal of Space Safety Engineering*, 6, 05 2019.
- [59] Daniel L. Oltrogge and Salvatore Alfano. The technical challenges of better space situational awareness and space traffic management. *Journal of Space Safety Engineering*, 6:72–79, 6 2019.
- [60] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.
- [61] Virtanen Pauli, Gommers Ralf, Oliphant Travis E., Haberland Matt, Reddy Tyler, Cournapeau David, Burovski Evgeni, Peterson Pearu, Weckesser Warren, Bright Jonathan, van der Walt Stefan J., Ribeiro Antonio H., Pedregosa Fabian, van Mulbregt Paul, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

- [62] Joseph Pelton. A path forward to better space security: Finding new solutions to space debris, space situational awareness and space traffic management. *Journal of Space Safety Engineering*, 6, 05 2019.
- [63] Joseph N. Pelton. A path forward to better space security: Finding new solutions to space debris, space situational awareness and space traffic management. *Journal of Space Safety Engineering*, 6:92–100, 6 2019.
- [64] Noah Perryman, Theodore Schwarz, Thomas Cook, Seth Roffe, Antony Gillette, Evan Gretok, Tyler Garrett, Sebastian Sabogal, Alan George, and Ricardo Lopez. Stp-h7-caspr: A transition from mission concept to launch. *digitalcommons.usu.edu*, 2021.
- [65] Christoph Posch, Teresa Serrano-Gotarredona, Bernab   Linares-Barranco, and Tobi Delbruck. Retinomorph event-based vision sensors: Bioinspired cameras with spiking output. *Proceedings of the IEEE*, 102:1470–1484, 10 2014.
- [66] Jiantao Qiu, Jie Wang, Song Yao, Kaiyuan Guo, Boxun Li, Erjin Zhou, Jincheng Yu, Tianqi Tang, Ningyi Xu, Sen Song, et al. Going deeper with embedded fpga platform for convolutional neural network. In *Proceedings of the 2016 ACM/SIGDA international symposium on field-programmable gate arrays*, pages 26–35, 2016.
- [67] Nicholas Ralph, Darren Maybour, Yeshwanth Bethi, and Gregory Cohen. Observations and design of a new neuromorphic event-based all-sky and fixed region imaging system. *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, AMOS*, PP:17–20, 09 2019.
- [68] Nicholas Ralph, Darren Maybour, Yeshwanth Bethi, Gregory Cohen, Nicholas Ralph, Darren Maybour, Yeshwanth Bethi, and Gregory Cohen. Observations and design of a new neuromorphic event-based all-sky and fixed region imaging system. *amos*, page 71, 2019.
- [69] B. Ramesh, S. Zhang, Z.-W. Lee, Z. Gao, G. Orchard, and C. Xiang. Long-term object tracking with a moving event camera. *British Machine Vision Conf. (BMVC)*, 2018.
- [70] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.

- [71] I. Abbot Richard and P. Wallace Timothy. Decision support in space situational awareness. *Lincoln Laboratory Journal*, 16(2):297, 2007.
- [72] Seth Roffe. *Evaluation of Algorithm-Based Fault Tolerance for Machine Learning and Computer Vision under Neutron Radiation*. University of Pittsburgh ETD, July 2020.
- [73] Seth Roffe, Himanshu Akolkar, Alan D. George, Bernabe Linares-Barranco, and Ryad B. Benosman. Neutron-induced, single-event effects on neuromorphic event-based vision sensor: A first step and tools to space applications. *IEEE Access*, 9:85748–85763, 2021.
- [74] Seth Roffe, Theodore Schwarz, Thomas Cook, Noah Perryman, Justin Goodwill, Evan Gretok, Aidan Phillips, Mitchell Moran, Tyler Garrett, and Alan George. Caspr: Autonomous sensor processing experiment for stp-h7. 2020.
- [75] Seth Roffe, Theodore Schwarz, Thomas Cook, Noah Perryman, Justin Goodwill, Evan Gretok, Aidan Phillips, Mitchell Moran, Tyler Garrett, and Alan George. Caspr: Autonomous sensor processing experiment for stp-h7. *digitalcommons.usu.edu*, 2020.
- [76] Dylan Rudolph, Christopher Wilson, Jacob Stewart, Patrick Gauvin, Alan George, Herman Lam, Gary Crum, Mike Wirthlin, Alex Wilson, and Aaron Stoddard. Csp: A multifaceted hybrid architecture for space computing. *Small Satellite Conference*, 8 2014.
- [77] Sebastian Sabogal, Patrick Gauvin, Brad Shea, Daniel Sabogal, Antony Gillette, Christopher Wilson, Alan George, Gary Crum, Ansel Barchowsky, and Tom Flatley. Ssivp: Spacecraft supercomputing experiment for stp-h6. 2017.
- [78] Sebastian Sabogal, Patrick Gauvin, Brad Shea, Daniel Sabogal, Antony Gillette, Christopher Wilson, Alan George, Gary Crum, Ansel Barchowsky, and Tom Flatley. Ssivp: Spacecraft supercomputing experiment for stp-h6. *Small Satellite Conference*, 8 2017.
- [79] T. Serrano-Gotarredona and B. Linares-Barranco. A 128×128 1.5% contrast sensitivity 0.9% FPN $3\mu\text{s}$ latency 4mW asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers. *IEEE Journal of Solid-State Circuits*, 48(3):827–838, 2013.
- [80] A Sipos, E Grusell, Andras Kerek, Wlodzimierz Klamra, J Molnár, L-O Norlin, D Novák, A Sanchez-Crespo, J Van der Marel, and J Végh. Visualization of neutron

- and proton induced particle production in a cmos image sensor. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 509(1-3):328–332, 2003.
- [81] Joseph R Srour and James M McGarrity. Radiation effects on microelectronics in space. *Proceedings of the IEEE*, 76(11):1443–1469, 1988.
- [82] Joseph R. Srour and James M. McGarrity. Radiation effects on microelectronics in space. *Proceedings of the IEEE*, 76:1443–1469, 1988.
- [83] Andrew James Stothers. On the complexity of matrix multiplication. 2010.
- [84] Y. Suh, S. Choi, M. Ito, J. Kim, Y. Lee, J. Seo, H. Jung, D. H. Yeo, S. Namgung, J. Bong, S. Yoo, S. H. Shin, D. Kwon, P. Kang, S. Kim, H. Na, K. Hwang, C. Shin, J. S. Kim, P. K. J. Park, J. Kim, H. Ryu, and Y. Park. A 1280x960 dynamic vision sensor with a 4.95-um pixel pitch and motion artifact minimization. In *IEEE Int. Symp. Circuits and Systems (ISCAS)*, pages 1–5, 2020.
- [85] Nowicki Suzanne, Wender Stephen, and Mocko Michael. The los alamos neutron science center spallation neutron sources. *Physics Procedia*, 90:374–380, 12 2017.
- [86] E. Oliphant Travis. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.
- [87] D. Reverter Valeiras, X. Lagorce, X. Clady, C. Bartolozzi, S.H. Ieng, and R. Benosman. An asynchronous neuromorphic event-driven visual part-based shape tracking. *IEEE Trans. Neural Netw. Learn. Syst. (TNNLS)*, 26:3045–3059, 2015.
- [88] David Reverter Valeiras, Xavier Lagorce, Xavier Clady, Chiara Bartolozzi, Sio Hoi Ieng, and Ryad Benosman. An asynchronous neuromorphic event-driven visual part-based shape tracking. *IEEE Transactions on Neural Networks and Learning Systems*, 26:3045–3059, 12 2015.
- [89] Milos Vujisic, Koviljka Stankovic, Nada Marjanovic, and Predrag Osmokrovic. Simulated effects of proton and ion beam irradiation on titanium dioxide memristors. *Nuclear Science, IEEE Transactions on*, 57:1798 – 1804, 09 2010.
- [90] S.J. Watts, John Matheson, I.H. Hopkins-Bond, A. Holmes-Siedle, A. Mohammadzadeh, and R. Pace. A new model for generation-recombination in silicon depletion regions after neutron irradiation. *Nuclear Science, IEEE Transactions on*, 43:2587 – 2594, 01 1997.

- [91] Steve Wender and Laura Dominik. Los alamos high-energy neutron testing handbook. *LA-UR-19-30813*, 2019.
- [92] Steve Wender and Laura Dominik. Los alamos high-energy neutron testing handbook. Technical report, SAE Technical Paper, 2020.
- [93] Christopher Wilson, Jacob Stewart, Patrick Gauvin, James MacKinnon, James Coole, Jonathan Urriste, Alan George, Gary Crum, Elizabeth Timmons, Jaclyn Beck, et al. Csp hybrid space computing for stp-h5/isem on iss. 2015.
- [94] Christopher Wilson, Jacob Stewart, Patrick Gauvin, James MacKinnon, James Coole, Jonathan Urriste, Alan George, Gary Crum, Elizabeth Timmons, Jaclyn Beck, Tom Flatley, Mike Wirthlin, Alex Wison, and Aaron Stoddard. Csp hybrid space computing for stp-h5/isem on iss. *Small Satellite Conference*, 8 2015.
- [95] John W Wilson, John E Nealy, Ts P Dachev, BT Tomov, FA Cucinotta, FF Badavi, G De Angelis, W Atwell, and N Leutke. Time serial analysis of the induced leo environment within the iss 6a. *Advances in Space Research*, 40(11):1562–1570, 2007.
- [96] Jiudong Yang and Jianping Li. Application of deep convolution neural network. *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2017*, 2018-February:229–232, 10 2017.
- [97] GJ Yates, JJ Bujnosek, SA Jaramillo, RB Walton, Teresa M Martinez, and JP Black. Radiation effects on video imagers. *IEEE Transactions on Nuclear Science*, 33(1):279–282, 1986.
- [98] Haissam Ziade, Rafic A Ayoubi, Raoul Velazco, et al. A survey on fault injection techniques. *Int. Arab J. Inf. Technol.*, 1(2):171–186, 2004.