TOTAL IONIZING DOSE EFFECT ON DEEP NEURAL NETWORKS IMPLEMENTED WITH MULTI-LEVEL RRAM ARRAYS

A Dissertation
Presented to
The Academic Faculty

by

Jack R. Sacane

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2021

TOTAL IONIZING DOSE EFFECT ON DEEP NEURAL NETWORKS IMPLEMENTED WITH MULTI-LEVEL RRAM ARRAYS

Approved by:

Dr. Shimeng Yu, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. John D. Cressler
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Asif Islam Khan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 30, 2021

To my family for their love and support.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Shimeng Yu, for his instruction and guidance since my first semester as an MS student. I would also like to thank my mother and father for their continued love and support throughout my life and many years at Georgia Tech. None of my successes would have been possible without them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

TID	Total Ionizing Dose
RRAM	Resistive Random-Access Memory
HfO_2	Hafnium Oxide
HRS	High-Resistance State
LRS	Low-Resistance State
MLC	Multi-Level Cell
NVM	Non-Volatile Memory
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
GAN	Generative Adversarial Network
SNN	Spiking Neural Network
MAC	Multiply-and-Accumulate

SUMMARY

This research work presents a methodology for simulating the effects of total ionizing dose (TID) radiation upon RRAM-based neural network accelerators. The experimental data on irradiating a 256×256 RRAM array test chip with ^{60}Co gamma rays up to a maximum TID of 1 Mrad (Si) were characterized with statistical methods in order to model the drift in RRAM cell conductance as a function of TID level. Multiple deep neural network (DNN) models were developed in the PyTorch framework in order to evaluate the effects of TID on DNNs implemented in hardware with similar RRAM memory technology and levels of radiation exposure. Using the statistical parameters discovered from the experimental TID data, weight changes were injected into the DNNs in order to simulate TID radiation effects and evaluate the resultant change of inference accuracy. Multiple simulations were conducted adhering to this methodology and the results pertaining to TID-induced inference accuracy degradation are discussed further in this work.

CHAPTER 1. INTRODUCTION

Emerging memory technologies are of particular interest to the aerospace and nuclear industries where choosing low-cost, high-density, and radiation-hardened non-volatile memory (NVM) is one of the key challenges in designing systems for harsh environments. Edge autonomy is also becoming a more desirable feature for those industries since many of their designs include small embedded systems which are not connected to the Internet and must meet real-time software deadlines for safe functionality. One example from the aerospace industry is the Mars Perseverance rover, which has complex scheduling tasks, a limited power budget, and is too far from Earth to receive immediate feedback from mission control. To improve its science throughput, Perseverance uses onboard planning software to operate autonomously between ground control commands. One example from the nuclear industry is the hundreds of connected sensors and actuators throughout every nuclear plant which monitor for abnormal conditions, report real-time power production data, and control time-sensitive mechanisms such as uranium centrifuges. Improved radiation-hardened processing and AI are vital innovations for the future of the aerospace and nuclear activities.

Many of the aerospace and nuclear plant computer systems in operation today employ Flash memory in their designs due to its low cost and high density. Flash achieves its high density by storing charge in floating-gate MOS transistors, but as a result, suffers from relatively low program/erase endurance when compared to other memory technologies such as dynamic random-access memory (DRAM) or hard disk drives (HDD) [1]. Flash is also particularly susceptible to failures as a result of ionizing radiation which

is present both in deep space and in nuclear reactors. Today's Flash technology can only sustain a total ionizing dose (TID) up to 75 krad (Si) and suffers functional failures during writes due to radiation-induced charge pump degradation [2]. Therefore, it is necessary to assess the radiation hardness of emerging NVM technologies such as resistive random-access memory (RRAM), as well as the radiation tolerance of RRAM-based neural network architectures that may be deployed on edge devices like spacecrafts or nuclear reactors.

In this research, the cumulative effects of TID radiation on a HfO_2-based RRAM memory array were characterized from hardware tests and then simulated in software to determine the radiation tolerance of various neural network architectures implemented with HfO_2 RRAM memory. The rest of this chapter provides background information needed to understand the work presented in this thesis.

1.1 Metal-Oxide RRAM Device

Metal-oxide RRAM is an emerging NVM technology that has seen intense research and development in both academia and industry over the past decade. It is classified into the emerging "non-volatile RAM" (NVRAM) category of memory along with other emerging technologies such as phase change memory (PCM) and magneto-resistive RAM (MRAM). Similar to PCM and MRAM, the RRAM memory cell functions as a "memristor"; it is a two-terminal device whose resistance can be electrically programmed to a high resistance state (HRS) or a low resistance state (LRS) in order to encode bits (1 or 0). When operated in multi-level cell (MLC) mode, multiple resistance levels between the HRS and LRS can be programmed to encode multiple bits per cell (e.g., with 4 programmable resistance levels, a single RRAM cell could encode $\log_2(4) = 2$ bits). Metal-

oxide RRAM is constructed with an insulating metal-oxide dielectric, such as HfO_2, sandwiched between two conducting electrodes (Fig. 1). The resistance of the channel is controlled by applying a high voltage across the electrodes in order to form/destroy a conductive filament in the insulating layer.

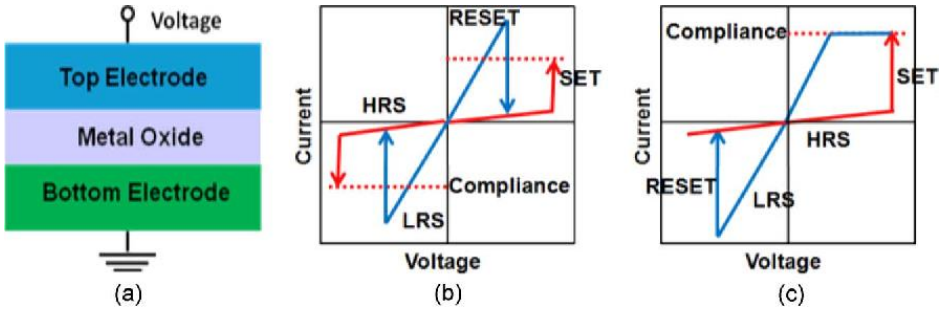


Fig. 1. Schematic of a metal-oxide RRAM cell with unipolar and bipolar I-V curves [3].

The early RRAM devices in the mid-2000s had large device areas (>> μm^2), large programming currents (~ mA), long programming times (> μs), low endurance (< 10^3 cycles), and required a large forming voltage (~10 V). In the mid-2010s, many of these deficiencies had been overcome. Device sizes down to 10 nm or below have been demonstrated, programming currents are now in the order of a few μA or tens of μA, programming speed is on the order of a few ns or tens of ns, programming endurance cycles are typically larger than 10^6, retention time can be as long as 3000 hours at 150 °C (which is extrapolated to be more than 10 years at 85 °C), and the forming process with large voltage in the first cycle can be much reduced to below 3V or even eliminated by shrinking the oxide thickness. Most of these good characteristics were reported in HfO_2 material systems, which are compatible with standard silicon CMOS processes. HfO_2 RRAM has since been demonstrated at the chip-level by various research institutions and companies

as a viable embedded NVM technology down to a 22 nm feature size (Windbond [4], TSMC [5], Intel [6], etc.)

1.2 Deep Neural Networks

In the field of machine learning, there are two distinct tasks which comprise most state-of-the-art inference models: feature extraction and classification. Neural networks are a type of machine learning model which have rose to prominence in the past decade due to their ability to learn feature extraction and classification simultaneously from raw input data. Advancements in the field of neural networks such as backpropagation, batch normalization, regularization, and optimization algorithms have enabled the creation of deep neural networks (DNN) which consist of many stacked layers of connected neurons and potentially millions of trainable parameters. DNNs such as AlphaGo, GoogLeNet, and DCGAN have revolutionized the field of artificial intelligence, displaying extremely high accuracies and performances which were even demonstrated to surpass human ability. Various DNN architectures exist for various purposes, with three being frequently used in today's industry: convolutional neural networks (CNN), recurrent neural networks (RNN), and generative adversarial networks (GAN).

1.2.1 Convolutional Neural Network (CNN)

Convolutional neural networks are a class of DNNs which are commonly used for image classification, image segmentation, video recognition, and signal processing. They are usually comprised of multiple interleaved convolutional layers and pooling layers. The convolutional layers convolve input data with a small kernel, producing a feature map that gets passed to the next layer. This process is similar to how individual neurons in the visual

cortex of the brain respond to stimuli only in a restricted region of the visual field (i.e., the receptive field). The pooling layers then reduce the dimension of the input data by combining the outputs of clusters of neurons into a single neuron in the next layer. This allows for the dominant features in the feature map which are positionally and rotationally invariant to be extracted and passed to the classification layer. By cascading convolutional layers and pooling layers, CNNs achieve high performance in a variety of image-related inference tasks with computational efficiency and built-in noise suppression.

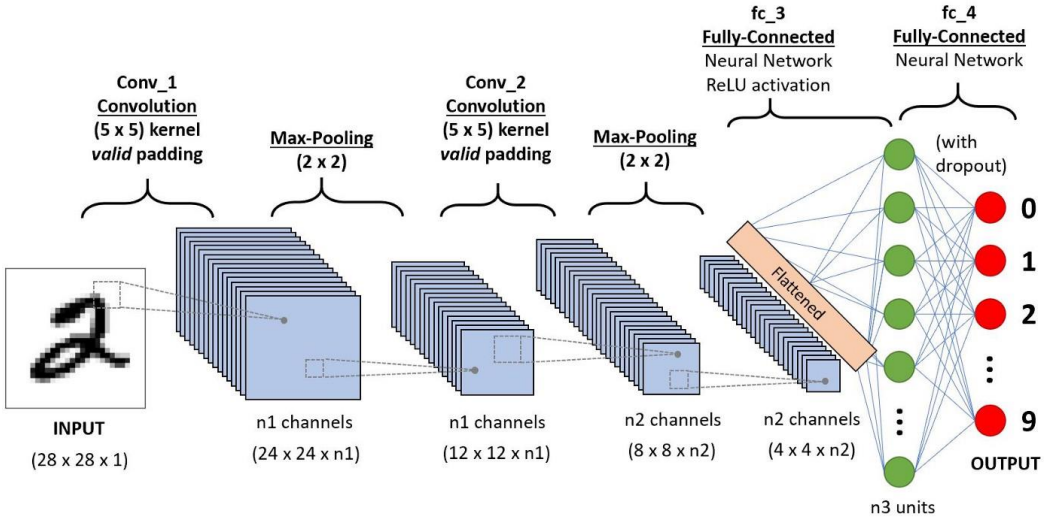


Fig. 2. CNN architecture for classifying handwritten digits from the MNIST dataset [7].

1.2.2 *Recurrent Neural Network (RNN)*

Recurrent neural networks differ from other types of DNNs in that they include feedback connections to compute current outputs from current inputs + previous outputs. This recurrent structure allows RNNs to hold “memory” over time which aids in learning temporally dependent features from time-series input data. RNNs are commonly used in natural language processing (NLP), speech recognition, and text autocompletion systems.

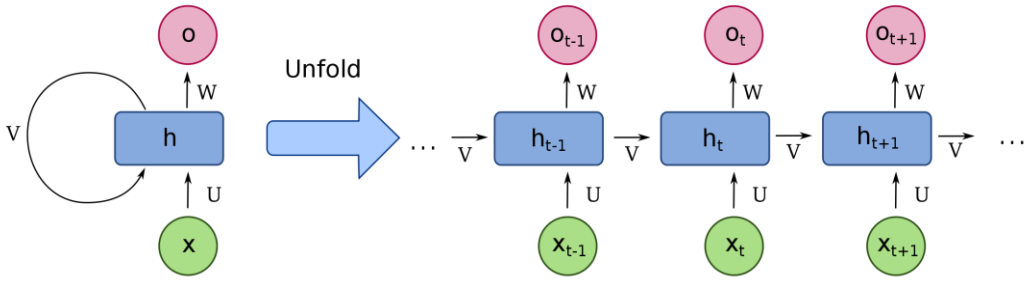


Fig. 3. Basic RNN unfolded in time. Feedback connections hold time series features in "memory" [8].

1.2.3 *Generative Adversarial Network (GAN)*

Generative adversarial networks are unique in that they train two models at the same time, namely the "generator" model and the "discriminator" model. The generator model, given a latent input vector, attempts to generate output data which mimics data from the original dataset. The discriminator model, given input data, attempts to determine the certainty that the input came from the original dataset. During training, these two models engage in a zero-sum game where the generator learns to produce fake data more representative of the dataset while the discriminator learns to be more accurate in discriminating fake data from real data. The result of this training process is a generator model that is highly capable of generating new data that appears to be from the original dataset. GANs are most often used in image reconstruction, video generation, and automatic dataset augmentation.

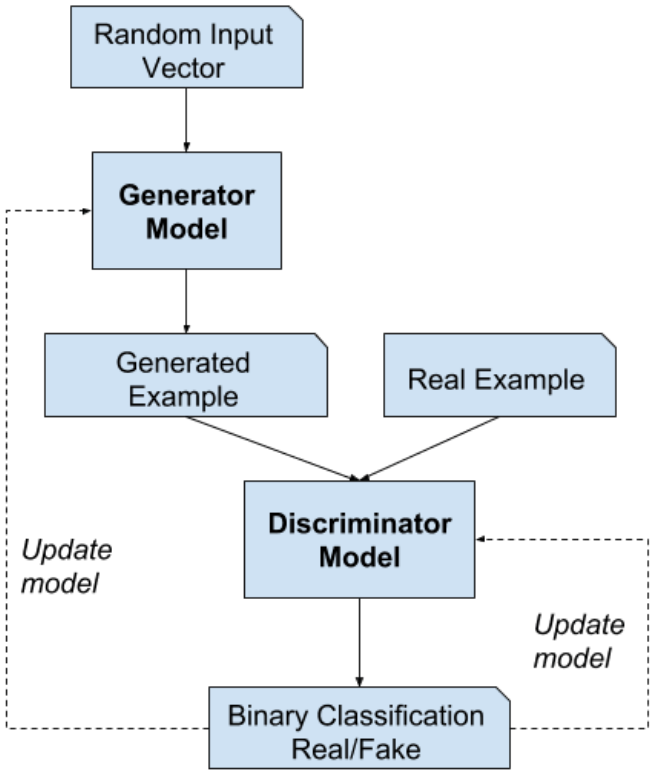


Fig. 4. GAN training process involves pitting a generator and discriminator against each other [9].

1.3 Neural Network Quantization

In the field of deep learning, quantization refers to the technique of shrinking the memory footprint of a neural network by reducing the numerical precision of its learnable parameters. For example, the weights and activations of a model can often be quantized from 32-bit floating point values to integer values of lower bit width (e.g., 8-bit, 4-bit, or 2-bit) in order to save memory and reduce computation. Weight quantization is often a necessary step in order to deploy DNNs on resource-constrained devices which may not have enough memory or floating-point hardware to perform inference efficiently. Quantization often comes at a cost of reduced inference accuracy. However, many quantization methods have been demonstrated in the literature which are highly successful

in minimizing performance degradation [10, 11, 12]. CNNs have generally enjoyed the most success with quantization. Applying quantization methods to other types of DNN architectures like RNN and GAN have been less successful due to their significantly higher sensitivity to weight initialization, weight precision, and hyperparameters during training [13]. However, some promising methods for RNN and GAN quantization have been proposed [13, 14, 15].

1.4 Neural Network Hardware Acceleration

Performing DNN computations efficiently is necessary for deploying DNNs on edge devices such as airplanes, spacecraft, and nuclear reactors. However, training and evaluating a large neural network is known to be both a memory-intensive and compute-intensive task. Large neural networks can have tens of millions of parameters which must be frequently transferred between on-chip memory, off-chip memory, CPU, and GPU if available. At the same time, one or more compute units must perform thousands of multiply-and-accumulate (MAC) operations with the network's weights in order to perform matrix-vector multiplications that produce the network outputs as well as gradients for weight updates. Although various techniques have been introduced to optimize this enormous computational cost, data movement still takes up to 90% of the total energy consumption even in highly optimized ASIC designs [16]. Researchers have been studying ways to tackle this "memory wall" problem by creating efficient hardware architectures for neural network training and inference.

Most computer systems today are implemented with a von Neumann architecture, which is to say that the computer's central processing unit and main memory unit are

logically separated. In such an architecture, CPU accesses to memory result in the desired bits being read out from the memory array row-by-row and then transferred to the CPU for processing. Although von Neumann computers still dominate in industry due to their simple design and long heritage, they also suffer from performance limitations when used in memory intensive applications such as DNN training and inference since all data must be transferred out of memory and then back into memory many times over. Processing-in-memory (PIM) architectures have recently emerged as a potential solution for DNN acceleration in hardware. Instead of one or a few CPUs reading bits row-by-row from a main memory unit, many small compute units are interleaved with many small memory units in a fabric for increased parallelism. MAC operations can then be performed along memory array bit lines with analog current and voltage for increased computational efficiency. Specifically, Ohm's Law ($I = V/R = VG$) applied to each cell of the array can be used to implement analog floating-point multiplication and Kirchhoff's Current Law ($\sum I_{out} = \sum I_{in}$) applied to the bit lines of the array can be used to implement floating-point addition. Crossbar memory arrays with selector-based cell access have also shown latency and energy improvements over traditional transistor-based cell access.

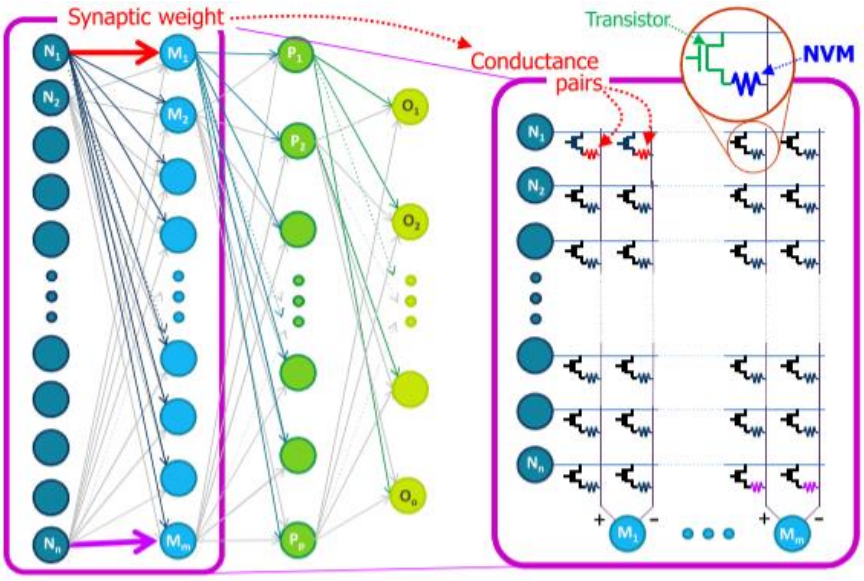


Fig. 5. Brain-inspired, non-von Neumann computing in which a dense crossbar array of memristors hold the synaptic weights of a neural network [17].

Due to its high density, high endurance, and precise programming with analog properties, RRAM is considered a promising candidate for storing synaptic weights in hardware implementations of neural networks. An RRAM-based memory array could be configured as an in-memory computing fabric like the PIM architecture described above. Then by mapping quantized weights into the RRAM array conductances and mapping network inputs into analog signals on the horizontal word lines, MACs can be efficiently performed in memory by reading out the analog currents on the vertical bit lines and converting them to digital values with ADCs. The variable resistance property of RRAM also makes it an attractive technology for use in spiking neural networks (SNN). SNNs differ from artificial neural networks in that they more closely mimic the biological structure of the brain and the electrical interactions between neurons over time. A neuron in an SNN only fires when its membrane potential exceeds some threshold. The signal from one neuron travels to other nearby neurons which, in turn, increase or decrease their

potentials in response and either fire or do not fire based on those potentials. Therefore, the resistance of an RRAM cell can function as the synapse between neurons, affecting their membrane potentials and spiking patterns. Computing paradigms which mimic the structure of the brain are also known as neuromorphic computing architectures.

CHAPTER 2. EXPERIMENTAL DATA ANALYSIS

This chapter describes the process by which experimental evidence for the effects of TID radiation on a multi-level RRAM array was collected. Then the statistical analysis used to extract conductance drift parameters from the experimental data is described. Finally, the conductance drift parameters are presented and discussed.

2.1 HfO_2 RRAM Test Chip

In order to quantify the TID effect, gamma-ray irradiation up to 1 Mrad (Si) was performed on a HfO_2 RRAM test chip in an experimental facility by our collaborators at Arizona State University [18]. The chip under test includes CMOS peripheral circuits (e.g., row decoder and level shifter) that control read/write access to a 256×256 one-transistor-one-resistor (1T1R) array, as shown in Fig. 6. The chip was fabricated in a 90nm technology node by courtesy of Winbond Electronics (Taiwan) and supported 2-bit MLC operation for a total of 4 programmable cell states. In order to test the effects of radiation on each cell state, 4096 cells of each state were programmed into the array as 64×64 subarrays. Table 1 shows the mapping of conductance values to cell states. State 1 is the high resistance state (HRS), State 4 is the low resistance state (LRS), and State 2 and State 3 are the intermediate states. After successfully programming all the cells, the chip was irradiated in a test chamber with ^{60}Co gamma rays. The cell conductances were then read out of the array at increasing TID levels and recorded for later data analysis. The discrete TID levels observed were 36 krad (Si), 255 krad (Si), 358 krad (Si), 652 krad (Si), and 932 krad (Si). A secondary RRAM chip was also programmed identically to the first chip but

not exposed to TID in order to be used as a control sample. Its conductances were read out at the same times as the radiation sample for comparison.

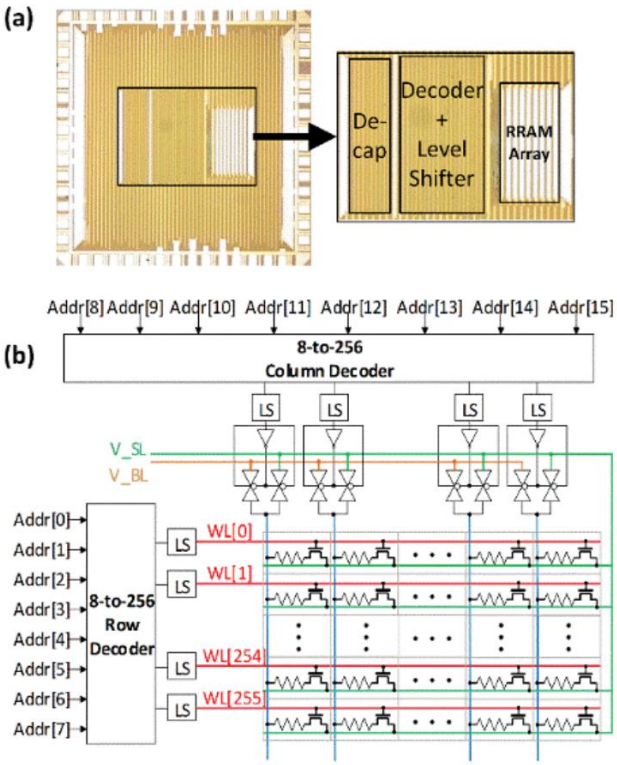


Fig. 6. (a) Die micrograph of the 64kb RRAM test chip. (b) Circuit schematic of the chip including: 256×256 1T1R RRAM array, column and row decoders, level shifter, and mux based on transmission gates [18].

TABLE 1. CONDUCTANCE MAPPING TO CELL STATE

State 1 (HRS)	<1.25 μS
State 2 (Intermediate)	52 μS ± 10%
State 3 (Intermediate)	104 μS ± 10%
State 4 (LRS)	156 μS ± 10%

2.2 Experimental Results

The heat maps in Fig. 7-9 visualize the 4 cell states of the radiation sample (into the four-quadrant plot) from the experimental data before irradiation, at TID level = 36 krad (Si) and at TID level = 932 krad (Si), respectively. During the irradiation, there is a measurable fluctuation of conductance values within each state, especially for State 2 and State 3. This observation is consistent with our current understanding of RRAM device physics, which suggests that intermediate states are more unstable due to the instability of forming weak conductive filaments that consist of oxygen vacancies.

There are also a handful of cells whose conductances shifted significantly enough to transition into other states. It is noticeable that at TID level = 36 krad (Si), some cells in State 1 flipped to State 4, while at TID level = 932 krad (Si), a portion of those cells recovered. Previous research has also observed such instability with possible flipping direction from HRS to LRS or vice versa [19]. This phenomenon was attributed to non-bridging oxygen being created by the irradiation as indicated by X-ray photoelectron spectroscopy (XPS). In the XPS spectra, the non-bridging oxygen peak significantly increased post-irradiation, indicating bond-breaking in the HfO_2 thin film and generation of new pairs of oxygen vacancies and oxygen ions. The new oxygen vacancies could form conductive filaments, triggering the HRS → LRS transition, while the creation of oxygen ions near the existing filaments could rupture the filaments, yielding the LRS → HRS transition.

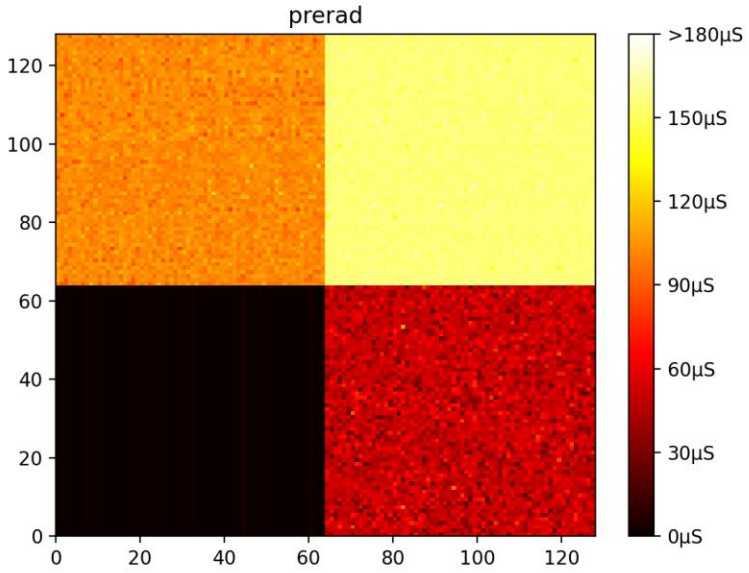


Fig. 7. Pre-irradiation heat map of cell conductances in the test chip. Bottom left = state 1 (HRS); Bottom right = state 2; Top left = state 3; Top right = state 4 (LRS).

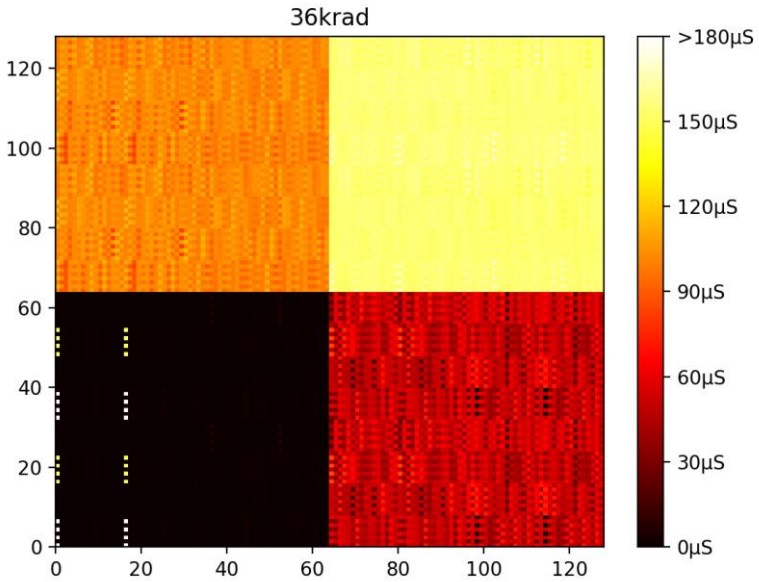


Fig. 8. Post-irradiation heat map of cell conductances in the test chip after TID of 36 krad (Si). A number of cells in state 1 (HRS) flipped to state 4 (LRS).

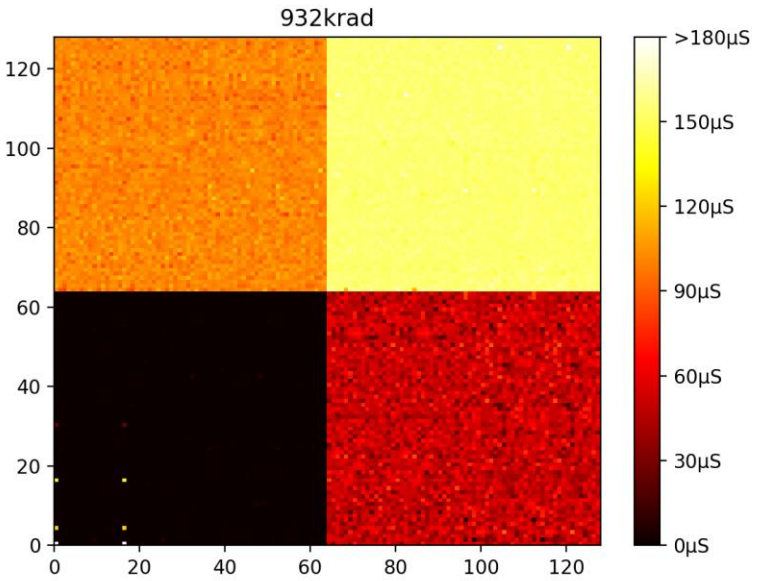


Fig. 9. Post-irradiation heat map of cell conductances in the test chip after TID of 932 krad (Si). Some cells in state 4 flipped back to HRS.

A general downward trend in the change of conductance can also be seen in the data visualizations. Fig. 10 shows the cumulative distribution function (CDF) of the conductance distributions for each state before irradiation and after 932krad (Si) of TID. The first conclusion that can be made from this data is that the distributions are strongly Gaussian as anticipated. The second conclusion is that there are slight bends in the tails of the distributions due to intrastate fluctuation, and some tail bits can be seen to have switched states entirely. Some cells in state 1 transitioned to State 3 and State 4, some cells in State 2 transitioned to State 1, and some cells in State 4 transitioned to State 3.

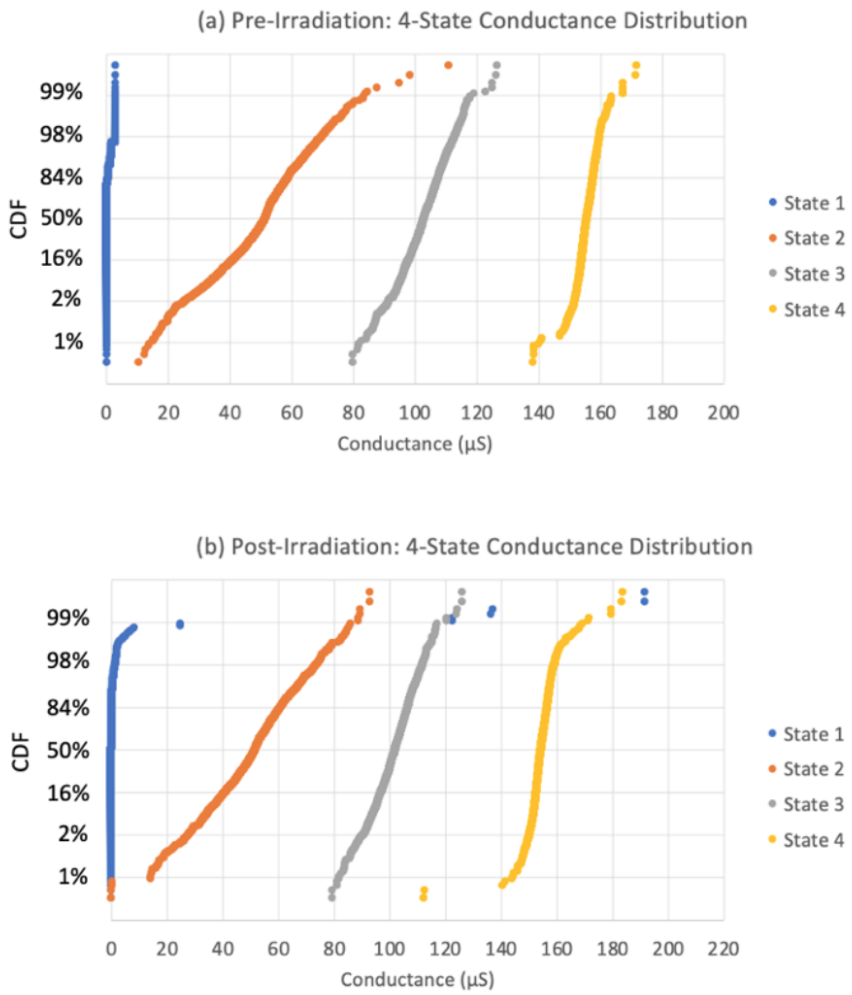
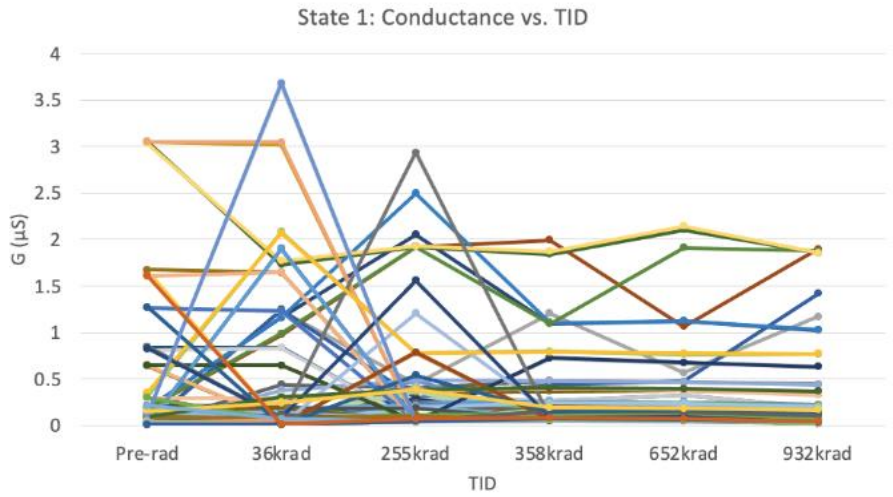
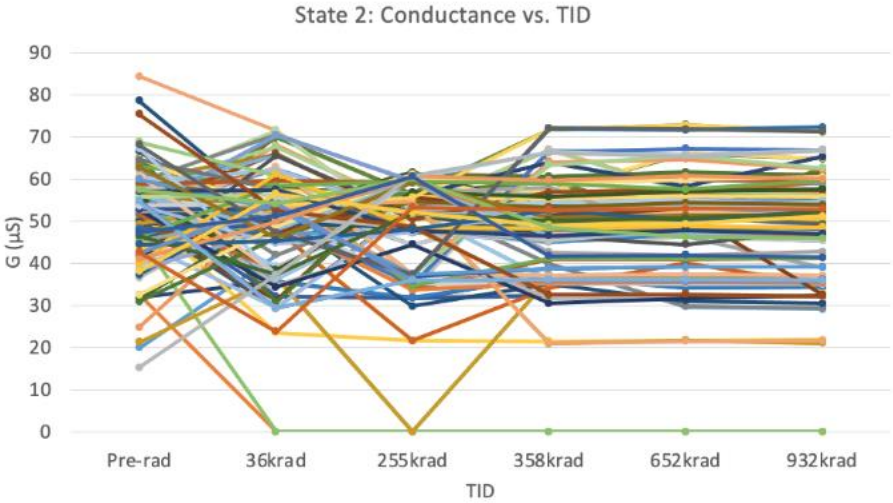
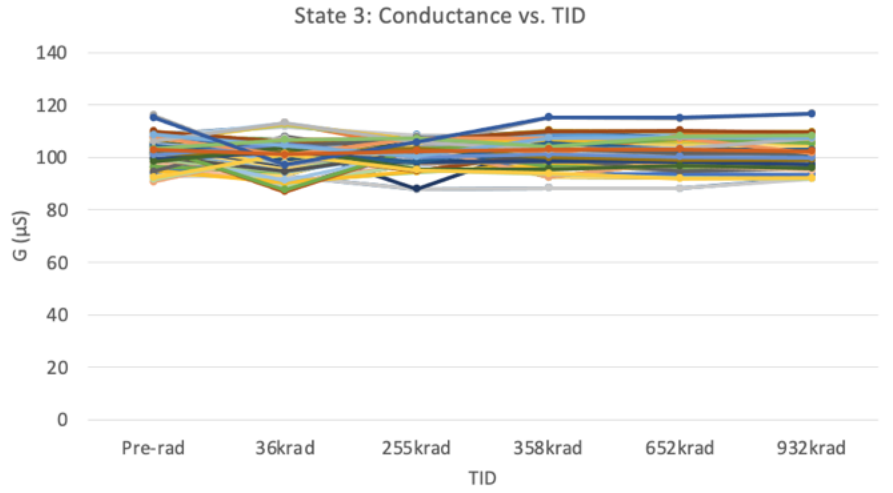


Fig. 10. The 4-state conductance distribution measured from the RRAM test chip before (a) and after (b) TID irradiation of 932krad (Si). The y-axis is presented in Gaussian scale.

Fig. 11 shows a random selection of 128 cells from each state and how their conductances changed across all the measured TID levels. The conductance fluctuations and outliers can be clearly seen. State 1 has the most outliers and shows the most inter-state variation. State 2 is the most unstable and shows the most intra-state variation. State 3 is relatively stable, and State 4 is relatively stable with the fewest outliers.


State 1: Conductance vs. TID
G (µS)
4
3.5
3
2.5
2
1.5
1
0.5
0
Pre-rad
36krad
255krad
358krad
652krad
932krad
TID


State 2: Conductance vs. TID
G (µS)
90
80
70
60
50
40
30
20
10
0
Pre-rad
36krad
255krad
358krad
652krad
932krad
TID


State 3: Conductance vs. TID
G (µS)
140
120
100
80
60
40
20
0
Pre-rad
36krad
255krad
358krad
652krad
932krad
TID

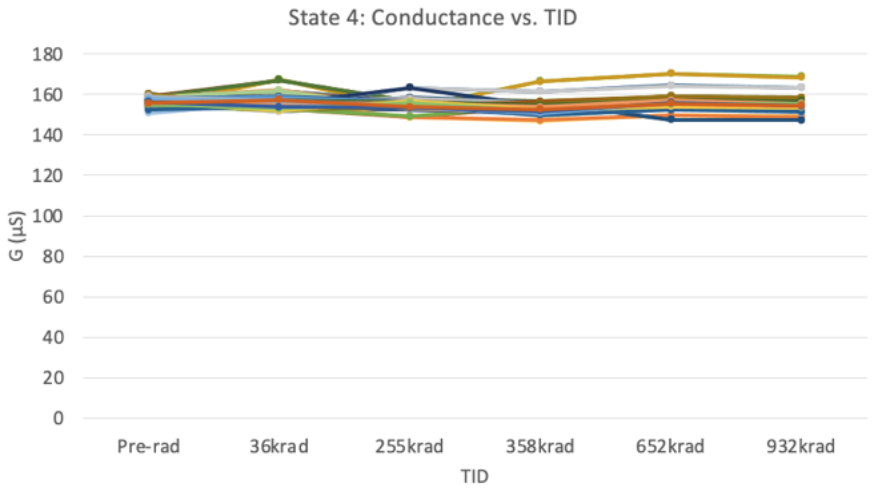


Fig. 11. A random selection of 128 cells from each state in the test chip plotted against TID level. State 1 had the most outliers. State 2 was the most unstable and showed the most intra-state variation. State 3 and State 4 had significant intra-state variation but the fewest outliers.

2.3 Statistical Analysis

The experimental data suggest that TID has an influence on both the analog and digital properties of the RRAM cells. Specifically, the conductances of the cells show slight drift and fluctuation within each state, while a small portion of cells showed large enough conductance variation to flip to other states. Therefore, the parameters of interest for TID analysis are the conductance mean drift, conductance fluctuation, and percentage of cells which flipped to other states.

The process for extracting the proportion of cells which flipped to other states was simple. By comparing the conductance values of each cell at a given TID level to their initial pre-irradiation values, the cells which transitioned into different conductance ranges could be identified and counted. The results of this process after 1 Mrad (Si) of TID are summarized in Table 2. HRS → LRS transitions were most frequent, with transitions into

the intermediate states being less frequent but non-negligible. Adopting a frequentist approach, these results can be interpreted as the independent probabilities of a cell flipping to another state after 1 Mrad (Si) of exposure.

TABLE 2. INTER-STATE TRANSITION PROBABILITIES AT 932KRAD

State 1 → State 4	0.146%
State 2 → State 1	0.098%
State 4 → State 3	0.049%

In order to characterize the analog MLC RRAM conductance trends as a function of TID level, the following methodology was derived: First, the outlier cells which "digitally" flipped to other states are removed in order to not bias the mean. Then, the relative conductance change of a cell, ΔG, is measured at each radiation dose by taking the difference between the current G of the cell and the initial pre-irradiation G of the cell.

$$\Delta G = G_{TID\ level} - G_{pre\ rad} \quad (1)$$

ΔG is thus calculated for each cell, and the average μ(ΔG) and standard deviation σ(ΔG) at every TID level for each state 1-4 are obtained. The results of this process after 1 Mrad (Si) of TID are summarized in Table 3. The aforementioned downward trend in conductance change can also be observed in this table. Aside from State 2 which exhibited

the most instability, the rest of the states showed an overall negative change in G when compared to the pre-irradiation baseline.

TABLE 3. MEAN AND STDEV OF ΔG FOR EACH STATE AT 932 KRAD

Cell State	μ(ΔG)	σ(ΔG)
1 (HRS)	-0.114 μS	0.434 μS
2 (Intermediate)	0.765 μS	10.99 μS
3 (Intermediate)	-1.025 μS	1.240 μS
4 (LRS)	-1.264 μS	1.209 μS

It is understood that RRAM may exhibit retention degradation even without irradiation due to thermal activation in the CMOS process. Therefore, it is important to compare μ(ΔG) and σ(ΔG) between the radiation sample and the control sample (i.e., the identical chip that did not go through the irradiation but was tested at the same time intervals). Fig. 12-13 show the μ(ΔG) and σ(ΔG) between the radiation sample and the control sample for each of the 4 states.

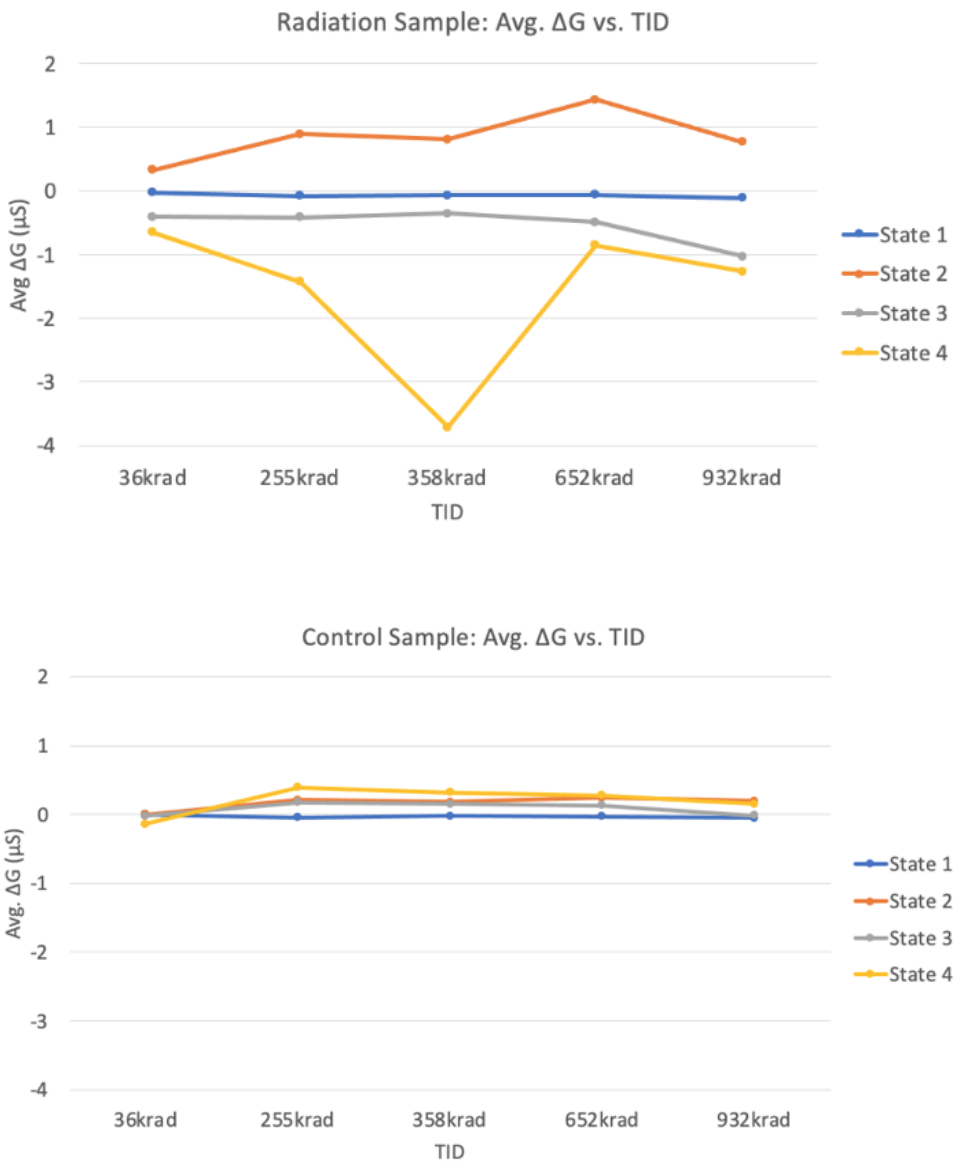


Fig. 12. Comparison between radiation sample and control sample of μ(ΔG) for each state across all the TID levels.

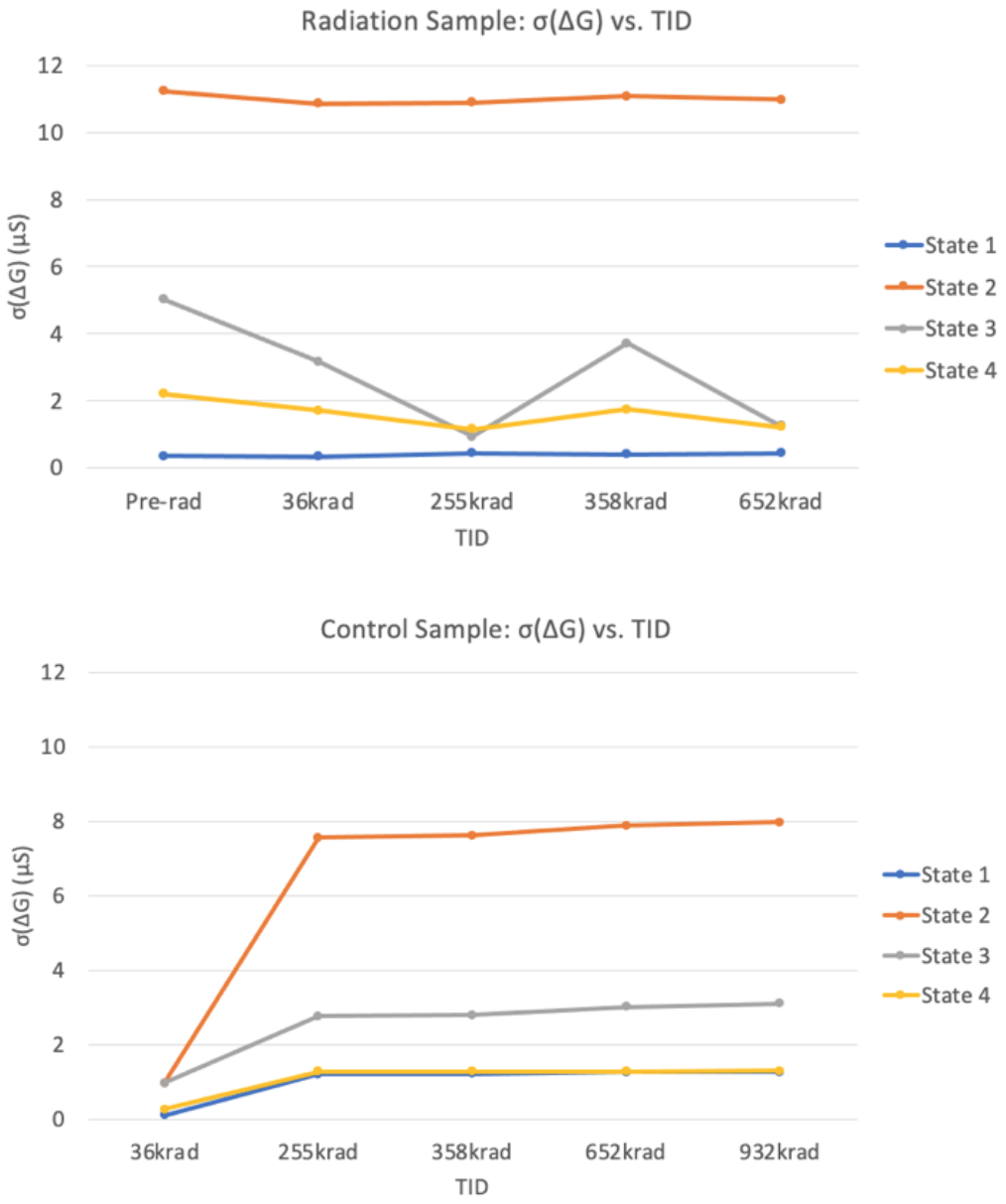


Fig. 13. Comparison between radiation sample and control sample of σ(ΔG) for each state across all the TID levels.

The data from Fig. 12-13 reveal that the radiation sample has ~3-5X larger μ(ΔG), but similar amount of σ(ΔG) when compared to the control sample. This means that the TID effect mainly causes conductance drift, while conductance fluctuation is a more intrinsic property potentially attributed to thermal vibration.

2.4 Discussion

It is clear from the experimental data that HfO_2 RRAM arrays are susceptible to conductance fluctuations under the influence of total ionizing dose radiation. Specifically, the effects of TID exhibited analog behavior in the form of intra-state conductance drift, as well as digital behavior in the form of inter-state bit flips. When simulating TID effects in an analog-based neural network accelerator, it is necessary to include the effects of both these phenomena into the weight update mechanism. The following section describes the weight update methodology, as well as the experimental setup for TID effect simulation and the results of TID simulation on multiple different DNNs.

CHAPTER 3. NEURAL NETWORK SIMULATIONS

This chapter introduces the deep neural network models that were used in the analysis of performance degradation under the influence of TID. First, the models and associated datasets are explained. Then, the TID simulation methodology is explained. Finally, the results of the simulation experiments are presented and discussed.

3.1 DNN Models

Multiple DNN models with different architectures and inference tasks were developed using the PyTorch deep learning framework in order to compare TID results across architectures and come to more general conclusions about the impact of TID on RRAM-based DNNs. Two important DNN tasks were identified: classification and generation. For the classification task, three popular CNN models were trained to classify images from the CIFAR-10 and ImageNette datasets. For the generation task, an RNN was trained to predict next characters based on previous characters in the Text8 and Penn Treebank datasets. Summaries of each architecture are given below.

3.1.1 Convolutional Neural Networks (CNN)

3.1.1.1 VGG-16

The VGG-16 model is a very popular CNN architecture that is commonplace in both industry and academia. It became famous for its simplicity and high performance in image classification at the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competition. The VGG network architecture generally consists of multiple

cascaded convolutional layers followed by max-pooling layers and then one or more fully connected layers for softmax classification. In the case of VGG-16, there are sixteen weight layers comprised of thirteen convolutional layers and three fully connected layers. Max pooling layers are placed between each convolutional filter block. The full architecture for each dataset is shown in Fig. 14.

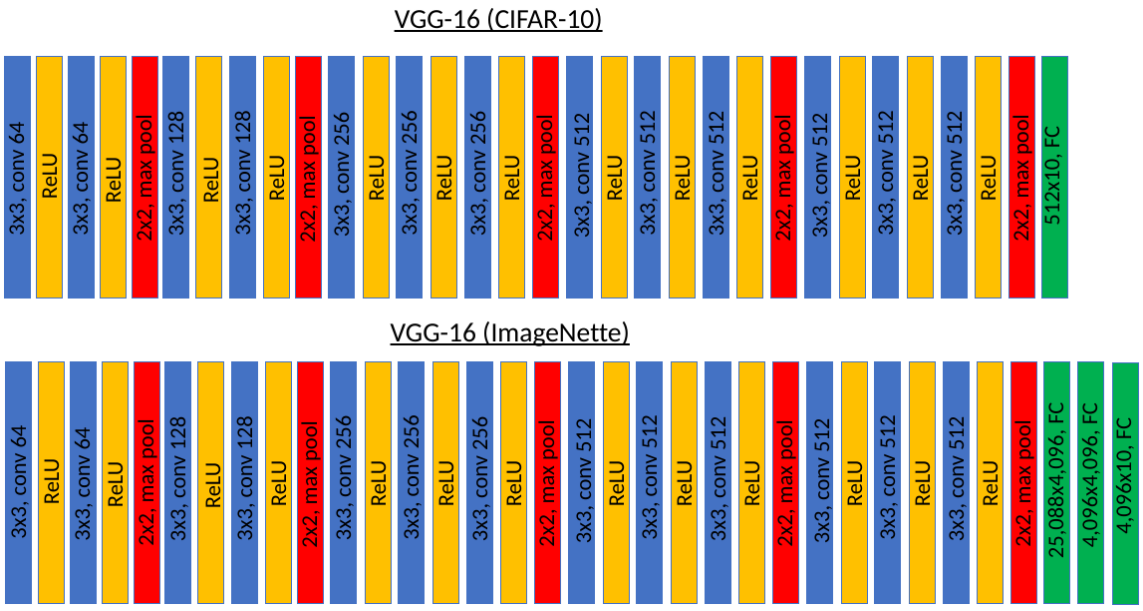


Fig. 14. The VGG-16 architecture consists of a pipeline of convolutional layers, max-pooling layers, and Rectified Linear Unit (ReLU) activations followed by one or more fully connected (FC) layers for softmax classification into the 10 different classes.

3.1.1.2 VGG-8

The VGG-8 model is a condensed version of the popular VGG-16 model. It has a similar architecture to VGG-16 but is half as deep with less layers and trainable weights. In particular, the VGG-8 model has eight weight layers comprised of three pairs of

convolutional layers with max-pooling in between and two fully connected layers. The full architecture for each dataset is shown in Fig. 15.

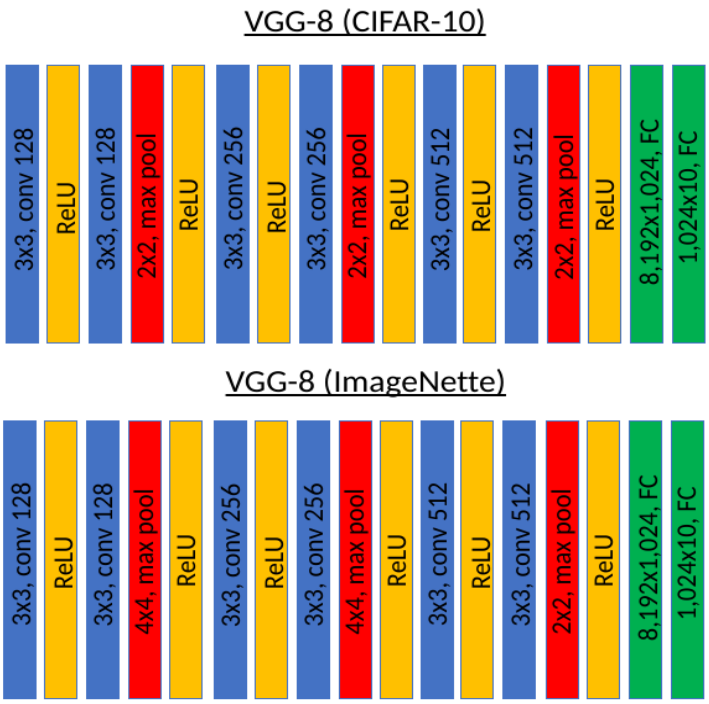


Fig. 15. The VGG-8 architecture is nearly identical to VGG-16, although it is shallower with only eight weight layers as opposed to sixteen.

3.1.1.3 ResNet-18

The ResNet-18 model is another highly popular model which gained fame after winning 1st place in image classification at the 2015 ILSVRC competition. The model contains eighteen weight layers which comprise four "residual learning blocks." Each block of the network contains both convolutional layers and shortcut connections to future blocks. The shortcut connections allow for information to flow from the initial layers to the last layers in one hop which helps alleviate the effects of overfitting during training.

This technique was shown to enable even deeper networks than VGG. The full architecture is shown in Fig. 16.

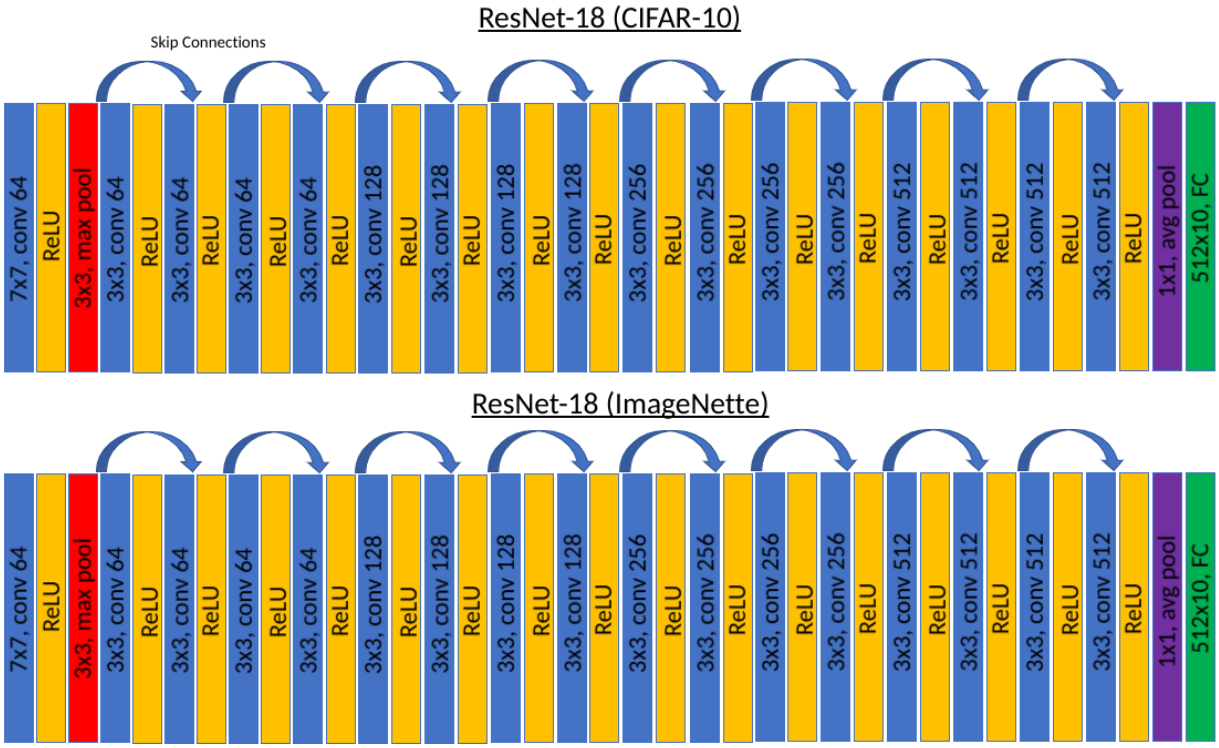


Fig. 16. The ResNet-18 architecture consists of a regular feed-forward pipeline like VGG-16 and VGG-8, but also has skip connections between convolutional blocks. ResNet-18 is the deepest model of the three and only has one max-pool and one average-pool layer before a fully connected layer for classification.

3.1.2 Recurrent Neural Networks (RNN)

3.1.2.1 LSTM

Although basic RNNs have shown good performance on some time-dependent tasks, their performance is known to be limited by a few major issues such as the vanishing gradient problem in training and the short-term memory problem in inference. In other words, basic RNNs are difficult to train, and they "forget" temporal dependencies in the dataset quickly

which hinders their performance on tasks that require longer-term context memory like speech recognition and NLP. To solve this problem, many variants of the RNN architecture have been explored in the literature. One of the most popular RNN variants is the Long Short-Term Memory (LSTM) architecture. The LSTM architecture consists of multiple LSTM cells connected sequentially, where each LSTM cell has three information pathways known as the input gate, output gate, and forget gate. The forget gate decides what information in the cell is important to keep or forget, the input gate decides what information should enter long term memory, and the output gate decides what information should be passed to the next LSTM cell or network layer. This carefully regulated cell structure allows an LSTM network to "remember" information for longer periods of time than the basic RNN architecture and helps avoid the vanishing gradient problem that commonly plagues RNNs. Therefore, due to its superior properties, an LSTM network with 512 hidden cell units was used for testing the effects of TID on RNNs in this experiment. The full architecture is shown in Fig. 17.

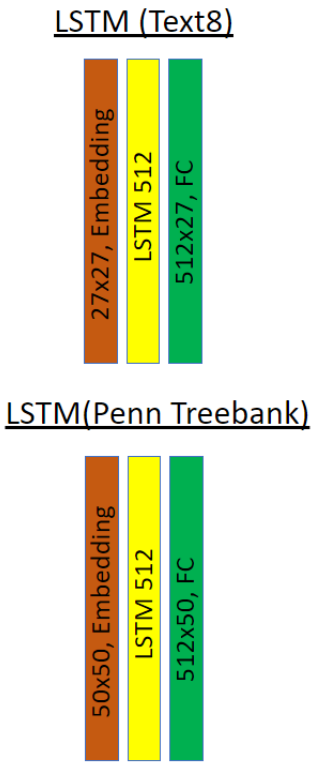


Fig. 17. The LSTM architecture consists of an embedding layer which learns to group input words of similar meaning, an LSTM layer with 512 hidden units for context learning, and then a fully connected layer to predict the next character. The Text8 dataset has 27 unique characters whereas Penn Treebank has 50.

3.2 Datasets

The datasets used for training the CNNs are comprised of thousands of images of ten different classes. Specifically, two popular image datasets were used to train the CNNs for image classification: CIFAR-10 and ImageNette. The images from these datasets are of assorted sizes, resolutions, and classes. Each image class refers to a different main subject of the image (e.g., dog, horse, airplane, truck, French horn, golf ball).

The datasets used for training the RNN are comprised of millions of English characters and words. Specifically, two popular language datasets were used to train the

RNN for character sequence prediction: Text8 and Penn Treebank. The sentences from these datasets come from various encyclopaedia, news, and literature sources and are commonly used for evaluating models on sequence labelling tasks.

3.2.1 *CIFAR-10*

The CIFAR-10 dataset [20] was prepared by the Canadian Institute for Advanced Research and contains 60,000 color images with 32×32 resolution in 10 different classes. There are 6,000 images of each class with the 10 different classes representing airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. The images in CIFAR-10 are very low resolution and are a labelled subset of the 80 Million Tiny Images dataset. CIFAR-10 is often used for quickly prototyping neural networks for image classification tasks.

3.2.2 *ImageNette*

The ImageNette dataset [21] was prepared by Fast.ai, a non-profit research group focused on deep learning and artificial intelligence. The images in ImageNette are a labelled subset of the popular ImageNet dataset, which contains over 14,000,000 images and is used in the annual ILSVRC competition to determine the state-of-the-art DNNs for image classification and object detection. ImageNette contains 14,000 color images of varying sizes in 10 different classes. The 10 classes are tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, and parachute. Since each image in the dataset has a different size, our PyTorch code applies bilinear interpolation to resize the images to 224×224 resolution before passing them to the networks.

3.2.3 Text8

The Text8 dataset [22] was prepared by Matt Mahoney. It is a 100 MB subset of "clean" English text (A-Z characters only) from the first 10^9 bytes of a dump of Wikipedia from 2006. It is commonly used for prototyping sequence based DNN models.

3.2.4 Penn Treebank

The English Penn Treebank (PTB) dataset [23] was prepared by the Penn Treebank Project team. It consists of a selection of 98,732 stories from the Wall Street Journal over a three-year period. The dataset includes over 7,000,000 words and 50 unique characters. PTB is commonly used for NLP research.

3.3 Methodology

There are three main steps to simulating the effects of TID on a DNN implemented with HfO_2 RRAM. First, the neural network must be trained and quantized to the bit precision of the RRAM hardware. Then, analog shift is applied to each weight of the network according to observed trends in the experimental TID data. Finally, digital shift is applied to a small percentage of weights to simulate the outlier RRAM cells which flip to other states. Each step of the process is explained more thoroughly in this section.

3.3.1 *Quantization*

The first step in the methodology is to make the DNN models compatible with the RRAM device cell precision. In this study, an RRAM chip with 2-bit MLC precision was used. Therefore, the weights of the neural networks must also be quantized to 2-bit values.

This quantization step was performed for each CNN model using the WAGE framework from [10]. WAGE constrains weights (W), activations (A), gradients (G) and errors (E) among all the layers of a CNN to low bit width integers in both training and inference. Therefore, a 2-8-8-8 W/A/G/E configuration was used in this work to isolate the effects of TID on weights while still keeping fairly high precision for activations, gradients, and errors which could be computed in a neuromorphic computing architecture with analog circuitry. For the RNN models, the Trained Ternary Quantization (TTQ) method from [15] is used. Quantizing RNNs is known to be a more challenging task since quantization naturally augments the exploding/vanishing gradient problem which RNNs already struggle with at full 32-bit precision. However, TTQ has shown good performance for low bit width RNN performance on language modelling tasks [14].

One caveat to the training process is that both WAGE and TTQ quantization with 2-bit precision actually produce symmetric ternary weights centered at 0 (i.e., (-x, 0, +x)) instead of utilizing the full range of quaternary values that are possible with 2 bits (i.e., (-x, -y, +y, +x)). This is because prior research has shown that symmetric ternary weights tend to perform better than binary and quaternary weights [10, 14]. WAGE clamps the weights between [-0.5, +0.5] and scales them down in later layers of the network by a constant scalar value in order to simplify batch normalization and prevent exploding/vanishing gradients during backpropagation. Since now the weight values are quantized to (-0.5, 0, +0.5) but there are 4 cell states in the RRAM hardware, two mapping schemes were devised: the first scheme is to map States 1/2/3 to (-0.5, 0, +0.5) and the second scheme is to map States 2/3/4 to (-0.5, 0, +0.5). These mapping schemes will be referred to as Scheme A and Scheme B respectively. For consistency, Scheme A and

Scheme B are also used in evaluating the RNN with TTQ quantization. It's worth noting that in a physical hardware design, negative weights can still be represented by using the middle state as a reference and performing differential read-out in the practical circuits.

3.3.2 Analog Weight Updates

In order to simulate the "analog" effects of TID exposure on RRAM-based neural networks, the conductance drift and conductance fluctuation phenomena previously discussed in Section 2 must be incorporated into the weights of the neural network. This weight update process is summarized by the following two steps:

1. For each weight in the network, sample a ΔG from the Gaussian distribution:

$$\Delta G = \mathcal{N}(\mu(\Delta G),\, \sigma(\Delta G)^2) \tag{2}$$

   where μ(ΔG) and σ(ΔG) are the drift and fluctuation parameters at 1 Mrad (Si) corresponding to the associated cell state for that weight.

2. Map ΔG → ΔW according to the rule:

$$G_{ref} = \frac{G_{max} + G_{min}}{2} \tag{3}$$

$$W = \left(G - G_{ref}\right) * weight\ scale \tag{4}$$

   where weight scale is a constant scalar used by WAGE and TTQ to scale down the weight magnitudes in each layer, simplifying batch normalization and achieving better training stability.

Table 3 shows the conductance drift and fluctuation parameters for each of the four cell states at 1 Mrad (Si).

3.3.3 Digital Weight Updates

In order to simulate the "digital" effects of TID radiation on RRAM memory arrays, the bit flipping phenomenon previously described in Section 2 must also be incorporated into the weights of the neural network. The outlier transition probabilities for each cell state at 1 Mrad (Si) are shown in Table 2. The error caused by RRAM cell conductances shifting to other levels was simulated in the DNN models by modifying random weights in each layer of the network so as to fit the same distribution of state transitions. For example, the experimental data showed that 0.146% of the cells transitioned from HRS to LRS. Taking that number as the probability of HRS → LRS transitions, 0.146% of the HRS weights in the DNN were changed to LRS weights. This process is repeated for all observed state transitions.

3.4 Results

3.4.1 CNN Models

For experimental validation, each CNN model was trained on each image dataset for 300 epochs and the baseline pre-irradiation inference accuracy for each of the DNN models was recorded. Then, each CNN was evaluated ten times under a simulated TID exposure of 1 Mrad (Si). During each trial, the weights of the baseline CNN were modified according to the TID simulation methodology using weight mapping Scheme A. Then the baseline CNN was modified according to the TID simulation methodology using weight mapping

Scheme B. The results of the ten trials for each CNN on each dataset were averaged and the resulting accuracy degradations were calculated. Table 4 shows the experimental results for each model on the CIFAR-10 dataset along with the number of weights in each model. Table 5 shows the experimental results for each model on the ImageNette dataset along with the number of weights in each model.

TABLE 4. TID SIMULATION RESULTS ON THE CIFAR-10 DATASET

Model	Baseline	Post-Rad Scheme A	Post-Rad Scheme B	Acc. Deg. Scheme A	Acc. Deg. Scheme B	# Network Weights
VGG-8	93.15%	92.69%	92.62%	-0.46%	-0.53%	12,973,440
VGG-16	88.91%	87.70%	88.17%	-1.21%	-0.74%	14,715,584
ResNet-18	72.28%	66.95%	23.66%	-5.33%	-48.62%	11,172,032

TABLE 5. TID SIMULATION RESULTS ON THE IMAGENETTE DATASET

Model	Baseline	Post-Rad Scheme A	Post-Rad Scheme B	Acc. Deg. Scheme A	Acc. Deg. Scheme B	# Network Weights
VGG-8	62.39%	61.68%	30.06%	-0.71%	-32.33%	30,274,944
VGG-16	89.32%	88.92%	87.80%	-0.40%	-1.52%	134,289,088
ResNet-18	87.54%	86.73%	33.99%	-0.81%	-53.55%	11,172,032

It should be noted that not all baseline accuracies are of similar value. Achieving state-of-the-art (>90%) accuracy on a dataset is a difficult undertaking which requires extensive testing, validation, architecture modification, and hyperparameter tuning. Different models tend to require different sets of parameters for maximum training efficacy. Larger models also tend to take longer to train but have higher upper limits on their accuracies. Since all models in this experiment were trained for 300 epochs using similar batch sizes and learning rates for variable control, the classification accuracies differ. For VGG-8, we also look at Top-1 accuracy while the other models use Top-5. However, for the purposes of this research, we are more interested in the TID-induced accuracy degradation effect than the maximum theoretical performance of each model.

3.4.2 RNN Models

The same experimental setup was used for training and evaluating the RNNs, however 100 epochs were used to train the RNNs on each of the text datasets and instead of classification accuracy, average character perplexity (PPL) is used as the metric for RNN performance. Lower PPL means the model's predicted next character fits the language model (i.e., the character probability distribution) better. Table 6 shows the experimental results for the LSTM model on the Text8 dataset along with the number of weights in the model. Table 7 shows the experimental results for the LSTM model on the Penn Treebank dataset along with the number of weights in each model.

TABLE 6. TID SIMULATION RESULTS ON THE TEXT8 DATASET

Model	Baseline PPL	Post-Rad Scheme A	Post-Rad Scheme B	PPL Incr. Scheme A	PPL Incr. Scheme B	# Network Weights
LSTM	3.49	3.61	10.51	+0.12	+7.02	2,232,859

TABLE 7. TID SIMULATION RESULTS ON THE PENN TREEBANK DATASET

Model	Baseline PPL	Post-Rad Scheme A	Post-Rad Scheme B	PPL Incr. Scheme A	PPL Incr. Scheme B	# Network Weights
LSTM	2.75	2.86	8.27	+0.11	+5.52	2,338,866

3.5 Discussion

3.5.1 Comparison Between Architectures

Overall, each of the CNN models showed good resilience to TID-induced weight changes up to 1 Mrad (Si) with at least one of the weight mapping schemes. In particular, VGG-8 and VGG-16 showed the highest resiliency to TID effects, exhibiting accuracy degradations of less than 1% in most cases. ResNet-18 also showed some resiliency with Scheme A, having experienced an average accuracy degradation of 5.33% on the CIFAR-10 dataset and 0.81% on the ImageNette dataset. However, ResNet-18 showed significant degradation on the order of 50% with both datasets when using Scheme B.

The ResNet architecture mainly differs from VGG in that it includes skip connections between previous layers and future layers in the residual learning blocks (Fig 17). I hypothesize that these skip connections make the ResNet architecture more unstable to TID radiation effects. With the weight error injection methodology presented in this thesis, each weight of the network is slightly shifted to account for the TID effects. However, with skip connections, weight errors in earlier layers will also be propagated to later layers, thus creating a compounding effect that could accumulate in the classification layers and amplify the accuracy degradation. In the case of the VGG architecture, no such skip connections exist, and each layer is independent of the previous/following layers. Therefore, errors in one layer are isolated to just that layer and compounding effects are minimized.

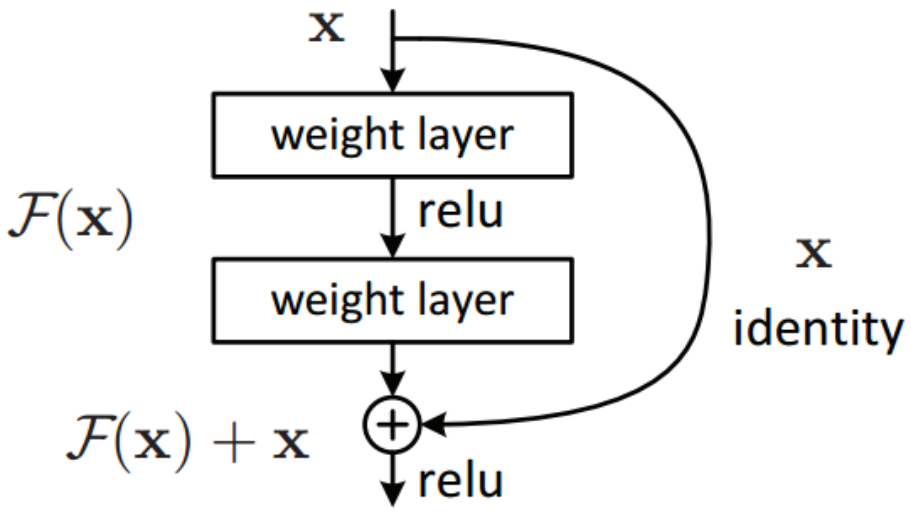


Fig. 18. Single residual block of a ResNet architecture showing the skip connection between layers [24]. TID-induced errors in the first weight layer would be compounded with errors in the second weight layer and then compounded again with errors from the previous residual block before passing the output to the next residual block.

The RNNs showed similar performance results to the CNNs despite having a different quantization method, a different inference task, and no noise-smoothing convolution operations. For Scheme A, the LSTM architecture was highly resilient to 1 Mrad (Si) of TID, only displaying a 0.11-0.12 increase in PPL for both datasets. However, the performance degradation became more severe for both datasets when using Scheme B. These results suggest that the degradation of DNN performance under TID may not depend significantly on the quantization method or the inference task (e.g., classification vs. generation). Rather, the structure of the architecture and the connection paths between neurons play a more pertinent role.

3.5.2 Comparison Between Weight Mapping Schemes

The results show that the choice of weight mapping scheme is important for minimizing inference accuracy degradation in both types of DNNs. In some cases, the difference between Scheme A and Scheme B was small, and either choice resulted in good resiliency. However, Scheme A tended to outperform Scheme B in most trials. This can likely be attributed to the stability of the represented cell states in each weight mapping scheme. In Scheme A, States 1, 2, and 3 are mapped to weights (-x, 0, +x). State 1 had the lowest amount of conductance drift and did not cause much fluctuation in its associated weights. State 2, the most unstable state, was relegated to influencing the 0 weights of the network. Therefore, despite having the widest ΔG distribution, State 2's fluctuations about 0 may preserve symmetry in a way that has minimal impact on the neural network inference. Meanwhile in Scheme B, States 2, 3, and 4 are mapped to weights (-x, 0, +x). State 1 is no longer considered in this mapping, and instead State 4 is used. State 4 contributed more fluctuation to its associated weights while the unstable intermediate

States 2 and 3 were still being used. Previous experimental results indicated TID-induced conductance drift is the more dominant force in accuracy degradation compared to TID-induced bit flips. Thus, Scheme B is naturally more unstable and displays larger accuracy degradation on average.

3.5.3 Comparison Between Model Sizes

The last variable of interest is the model size/depth. The impact of model size on TID-resiliency is less conclusive in this data and requires a more thorough investigation. However, some of the data suggest that larger models with higher numbers of weights are more resilient to TID effects. For example, VGG-16 is the largest model with ~15 million weights for the CIFAR-10 dataset and ~134 million weights for the ImageNette dataset. It was also the most resilient to TID-induced weight changes across both datasets and weight mapping schemes, averaging only about 1% accuracy degradation. This observation might be explained by the higher degree of redundancy that larger models have with their weights and neuron connections. With more connections, a significant shift in one weight due to intra-state fluctuation or inter-state bit flipping has less of a total effect. However, it should be noted that LSTM, the smallest model with just ~2 million weights, still displayed high resiliency on both datasets with Scheme A. Smaller models being less resilient may be a trend, but it is not a rule.

An interesting outlier is the decrease of 32% in VGG-8 for the ImageNette dataset. VGG-8 is not very small (~30 million weights) and previously showed good performance on CIFAR-10 with both weight schemes. It also does not have any architecture idiosyncrasies that could amplify the effects of weight errors. It's possible that during

training, this model converged to a relatively steep peak in its high-dimensional parameter space. If the model converged to such a peak, then a small disturbance could still have a large effect on the overall accuracy. In any case, the dynamics of radiation-induced effects on deep neural network performance are complicated and not yet well understood. More research is needed in this area to better understand this phenomenon.

CONCLUSION

In this research, a HfO_2-based multi-level RRAM test chip was analyzed under gamma ray irradiation at increasing levels of total ionizing dose radiation. A statistical model was developed from the data to capture the distribution of conductance changes, ΔG, of each RRAM cell as a result of TID radiation. Finally, the statistical model was used to inject weight errors into multiple pre-trained, 2-bit quantized DNN models in order to simulate the effects of 1 Mrad (Si) of TID radiation on DNN inference accuracy, assuming the DNNs were implemented in hardware with similar RRAM-based memory.

The results of the simulations showed that such RRAM-based neural network accelerators should be highly robust to TID-induced weight changes up to 1 Mrad (Si), exhibiting inference accuracy degradations as low as 0.40%. However, software and hardware engineers must be careful when designing and evaluating their networks under simulated TID environments. DNN architecture, model size, weight quantization method, and weight mapping scheme can all influence the resiliency of the network to TID radiation effects. Although these preliminary results are promising for the prospect of using RRAM-based neural network accelerators in radiation environments, there is still more opportunity for research in this area. In a future work, more scrutiny could be placed on the lower-level implications of weight error propagation through a network. It would also be useful to study the effects of the intermediate TID levels (36krad – 652krad) that we were not simulated in this study in order to better understand the role of the RRAM-based neural network performances across all the TID levels of interest. Finally, the effects of TID on

GANs and other types of RNN architectures like Gated Recurrent Unit (GRU) should be studied further when their training and quantization methods become more mature.

REFERENCES

[1] R. Bez, E. Camerlenghi, A. Modelli and A. Visconti, "Introduction to flash memory," *Proceedings of the IEEE*, vol. 91, no. 4, pp. 489-502, April 2003.

[2] D.N. Nguyen, C.I Lee and A.H. Johnston, “Total ionizing dose effects on flash memories,” *IEEE Radiation Effects Data Workshop*, pp. 100-103, August 1998.

[3] H. P. Wong, H. Lee, S. Yu, Y. Chen, Y. Wu, P. Chen, B. Lee, F. T. Chen, and M. Tsai, "Metal–Oxide RRAM," *Proceedings of the IEEE*, vol. 100, pp. 1951-1970, June 2012.

[4] C. Ho, S. Chang, C. Huang, Y. Chuang, S. Lim, M. Hsieh, S. Chang, and H. Liao, "Integrated HfO2-RRAM to achieve highly reliable, greener, faster, cost-effective, and scaled devices," *IEEE International Electron Devices Meeting (IEDM)*, 2017, pp. 2.6.1-2.6.4.

[5] C. Chou, Z. Lin, P. Tseng, C. Li, C. Chang, W. Chen, Y. Chih, and T. J. Chang, "An N40 256K×44 embedded RRAM macro with SL-precharge SA and low-voltage current limiter to improve read and write performance," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2018, pp. 478-480.

[6] P. Jain, U. Arslan, M. Sekhar, B. C. Lin, L. Wei, T. Sahu, J. Alzate-vinasco, A. Vangapaty, M. Meterelliyoz, N. Strutt, A. B. Chen, P. Hentges, P. A. Quintero, C. Connor, O. Golonzka, K. Fischer, and F. Hamzaoglu, "13.2 A 3.6Mb 10.1Mb/mm2 Embedded Non-Volatile ReRAM Macro in 22nm FinFET Technology with Adaptive Forming/Set/Reset Schemes Yielding Down to 0.5V with Sensing Time of 5ns at 0.7V," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2019, pp. 212-214.

[7] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way", Medium, December 15, 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[8] Wikipedia. "Recurrent neural network", En.wikipedia.org, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Recurrent_neural_network.

[9] J. Brownlee, "A Gentle Introduction to Generative Adversarial Networks (GANs)", Machine Learning Mastery, June 17, 2019. [Online]. Available: https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/.

[10] S. Wu, G. Li, F. Chen and L. Shi, "Training and inference with integers in deep neural networks," *International Conference on Learning Representations (ICLR)*, 2018.

[11] J. Choi, S. Venkataramani, V. Srinivasan, K. Gopalakrishnan, Z. Wang and P. Chuang, "Accurate and efficient 2-bit quantized neural networks," *Conference on Systems and Machine Learning (SysML)*, 2019.

[12] Y. Li, X. Dong and W. Wang, "Additive powers-of-two quantization: An efficient non-uniform discretization of neural networks," *International Conference on Learning Representations (ICLR)*, 2020.

[13] P. Wang, D. Wang, Y. Ji, X. Xie, H. Song, X. Liu, Y. Lyu and Y. Xie, "QGAN: Quantized Generative Adversarial Networks," arXiv:1901.08263 [cs], Jan. 2019, [Online]. Available: http://arxiv.org/abs/1901.08263.

[14] L. Hou, J. Zhu, J. Kwok, F. Gao, T. Qin, and T-Y. Liu, "Normalization Helps Training of Quantized LSTM," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[15] C. Zhu, S. Han, H. Mao, W. Dally, "Trained Ternary Quantization," ArXiv:1612.01064 [cs], Feb. 2017, [Online]. Available: http://arxiv.org/abs/1612.01064.

[16] M. Imani, M. Samragh, Y. Kim, S. Gupta, F. Koushanfar and T. Rosing, RAPIDNN: In-Memory Deep Neural Network Acceleration Framework, 2018.

[17] P. Narayanan, A. Fumarola, L. L. Sanches, K. Hosokawa, S. C. Lewis, R. M. Shelby and G. W. Burr "Toward on-chip acceleration of the backpropagation algorithm using nonvolatile memory," *IBM Journal of Research and Development*, vol. 61, no. 4/5, pp. 11:1-11:11, 1 July-Sept. 2017.

[18] X. Han, A. Privat, K. E. Holbert, J.-S. Seo, S. Yu and H. J. Barnaby, "Total ionizing dose effect on multi-state HfOx-based RRAM synaptic array," *IEEE Nuclear & Space Radiation Effects Conference (NSREC)*, 2020.

[19] R. Fang, Y. Gonzalez-Velo, W. Chen, K. Holbert, M. Kozicki, H. Barnaby and S. Yu, “Total ionizing dose effect of γ-ray radiation on the switching characteristics and filament stability of HfOx resistive random access memory,” *Applied Physics Letters*, vol. 104, 183507, 2014.

[20] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” University of Toronto, 2009.

[21] J. Howard, "fastai/imagenette", GitHub, 2021. [Online]. Available: https://github.com/fastai/imagenette.

[22] M. Mahoney, "About the Test Data", Mattmahoney.net, 2011. [Online]. Available: http://mattmahoney.net/dc/textdata.

[23] "Treebank-3 - Linguistic Data Consortium", University of Pennsylvania, [Online]. Available: https://catalog.ldc.upenn.edu/LDC99T42.

[24] S. Sahoo, “Residual Blocks — Building Blocks of ResNet,” Medium, November 27, 2018. [Online]. Available: https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec.