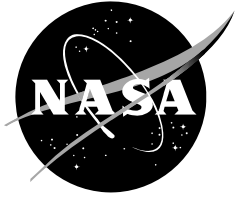


NASA/TM–20240006221



EdgeCortix SAKURA-I Machine-Learning, PCIe Accelerator SEE Proton Test

Seth Roffe

Scott Stansberry

Jeffrey Grosman

Jeffry Milrod

Manish Sinha

Uzzal Podder

Stan Crow

January 2024

NASA STI Program Report Series

The NASA STI Program collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- Help desk contact information:

<https://www.sti.nasa.gov/sti-contact-form/> and select the "General" help request type.

NASA/TM–20240006221



EdgeCortex SAKURA-I Machine-Learning, PCIe Accelerator SEE Proton Test

*Seth Roffe
Goddard Space Flight Center, Greenbelt, MD*

*Scott Stansberry
Science Systems and Applications (SSAI), Inc., Lanham MD*

*Jeffrey Grosman
EdgeCortex, Kawasaki, Kanagawa, Japan*

*Jeffrey Milrod
EdgeCortex, Kawasaki, Kanagawa, Japan*

*Manish Sinha
EdgeCortex, Kawasaki, Kanagawa, Japan*

*Uzzal Podder
EdgeCortex, Kawasaki, Kanagawa, Japan*

*Stan Crow
EdgeCortex, Kawasaki, Kanagawa, Japan*

*Test Date: 1/28/2024
Report Date: 3/18/2024*

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, MD 20771

January 2024

Acknowledgments

This work was sponsored by NASA Electronic Parts and Packaging (NEPP) Program and supported by the Office of the Under Secretary of Defense (OUSD) Trusted and Assured Microelectronics Program.

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA STI Program
Mail Stop 148
NASA's Langley Research Center
Hampton, VA 23681-2199

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

This report is available in electronic form at
<https://nepp.nasa.gov/>

1. Introduction and Purpose

To enable autonomy in space, machine-learning and computer vision applications become invaluable for sensor processing. However, these algorithms are computationally complex and unfeasible for many embedded central processing units (CPUs) and usually require external chips, such as graphics processing units (GPUs) or accelerators specific to the application. In power-constrained systems, GPUs tend to consume more power than is acceptable, so lower-power accelerators have shown promise to provide the performance needed under spacecraft constraints. For radiation engineers, developing methodologies that can properly test CPUs, GPUs, and accelerators, and enable comparisons between them remains a necessary complication to solve as the devices become more complex. The methodology in this test aims to be a start in developing a baseline single-event effect (SEE) test for client-device machine learning accelerators, in that they do not host their own operating system.

This experiment characterizes SEEs and data error susceptibility of the EdgeCortex SAKURA-I machine-learning accelerator. The device was monitored for single event upsets (SEUs) under 200-MeV protons at the Massachusetts General Hospital (MGH) Francis H. Burr Proton Therapy Center. The SAKURA-I board accelerates machine-learning inference applications on a host computer through a PCIe connection. For the purposes of this experiment, the YOLOv5 object detection model was used as the primary application during the test.

2. Test Result Summary

The SAKURA-I card did not experience any destructive SEEs during the 200-MeV proton irradiation with a flux reaching $\sim 10^9 \frac{p}{cm^2s}$. Tolerable errors were observed in the confidence scores of the objects detected in the image. Errors were determined to be tolerable when they still correctly identify the objects within the image, just with a slightly different size of bounding box. The errors in the confidence scores appeared to recover upon the next inference, potentially due to data refreshing from the DRAM not under irradiation. Additional testing is needed to determine what is likely the cause of the upsets within the system. More analysis and testing on other models, such as image classification models will also be required to further classify upsets in the SAKURA-I chip. Testing other model types, such as image classification or image segmentation will give a better understanding of how different model architectures can affect the reliability results.

3. Device Description

The device-under-test (DUT) was the SAKURA-I card, a PCIe ASIC accelerator designed to accelerate inference on machine-learning applications on the edge. It contains 20 MB of on-chip memory and 16 GB of external LPDDR4. The interface for the card connects to a host PC via a PCIe 3.0 x16 slot. Further details can be seen in Table 1. A picture of the card used in the test can be seen in Figure 1.

Table 1: SAKURA-I Card Details [1]

Part	SAKURA-I Edge Accelerator
REAG ID	24-001
Manufacturer	EdgeCortex
Cache	20MB
External Memory	16 GB LPDDR4
Reported Performance	40 TOPS
Reported Power	10 – 12 W
Interface	PCIe Gen 3.0 x16

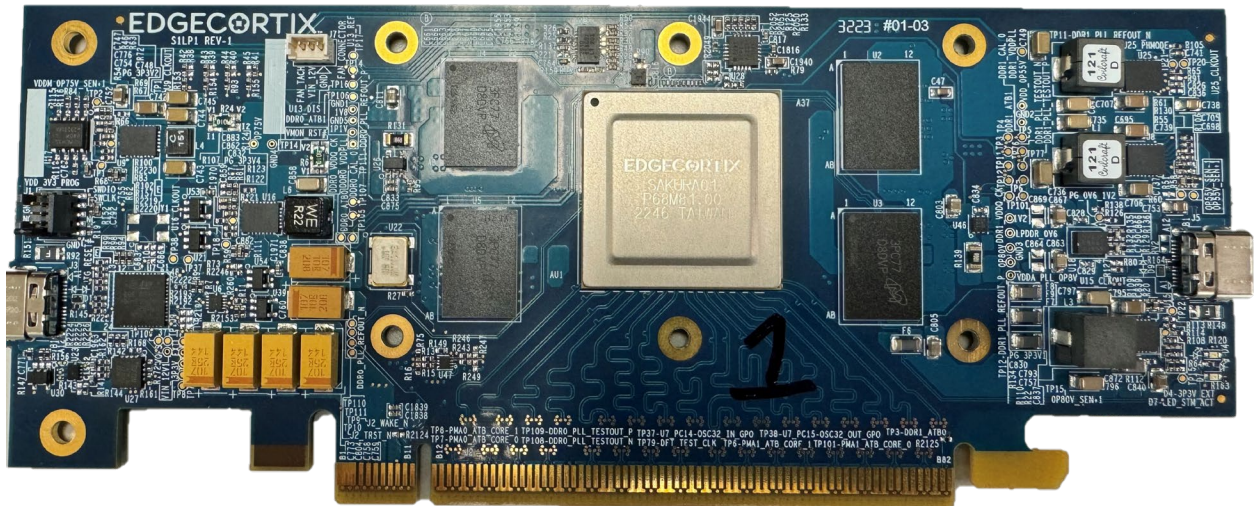


Figure 1: Picture of the Sakura-I Card

4. Test Setup

The SAKURA-I Card was mounted and clamped at normal incidence to the end of the beam line and connected to a host PC running Ubuntu 20.04 via a 0.5 m PCIe extension cable. A one-inch collimator was used to narrow the beam to roughly the size of the main chip. Two images of the test setup can be seen in Figure 2.



Figure 2. Test setup of the radiation test.

5. Test Facility

Facility:	Massachusetts General Hospital Burr Proton Therapy Center
Type of Radiation:	Proton
Energy:	200 MeV
Flux:	$\sim 10^7 - 10^9 \frac{p}{cm^2s}$

6. Test Conditions

Temperature:	Room Temperature
In-Air or Vacuum:	In-air
Supply Voltages:	12 V

7. Test Methods and Procedures

This section covers the methodology used in this experiment. Details about the model choices and how runs were defined are discussed herein.

7.1. Model Selection

The model chosen was the YOLOv5 object detection model with images selected from the Common Objects in Context (COCO) 80 dataset, which contains 80 classes [2]. To control the input data to the model, only 5 images were selected due to the number of objects that would be detected in each one. By reducing the inputs to a small number, the output vectors will be consistent between runs without introducing the additional data size of having too many input vectors. However, by having more than one input image, any input dependence on the output score reliability can be measured. These two points keep the test realistic to a real-world case, where there would likely be inferences on only one image at a time, while still analyzing how different confidences affect the results. The images used for this test can be seen in Figure 3. An example of a correctly classified outcome can be seen in Figure 4.

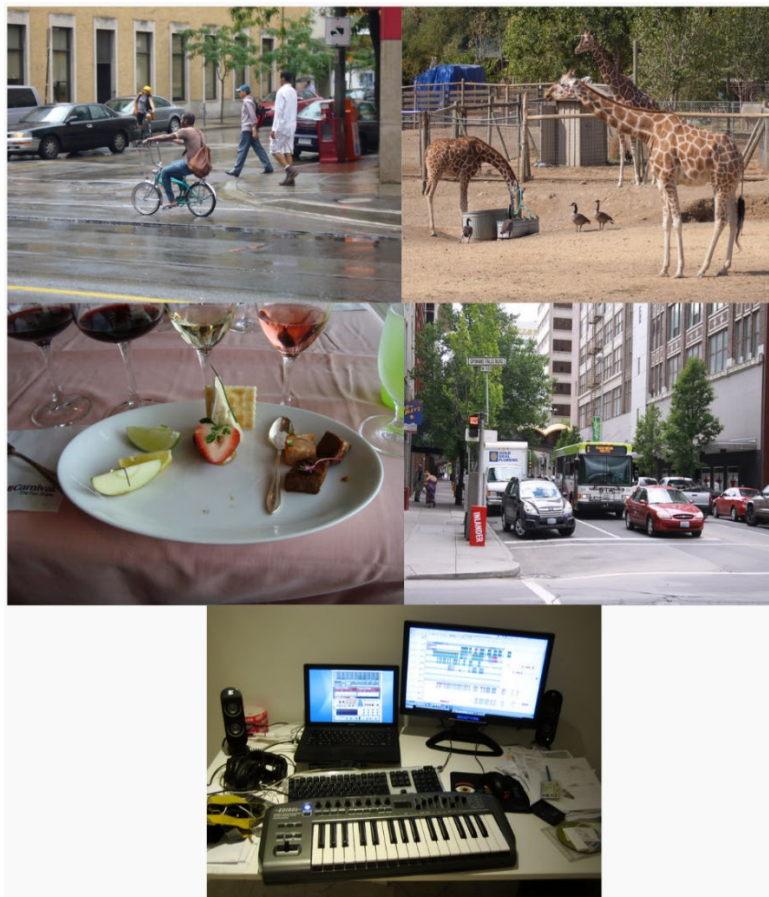


Figure 3. Images used from the COCO dataset.

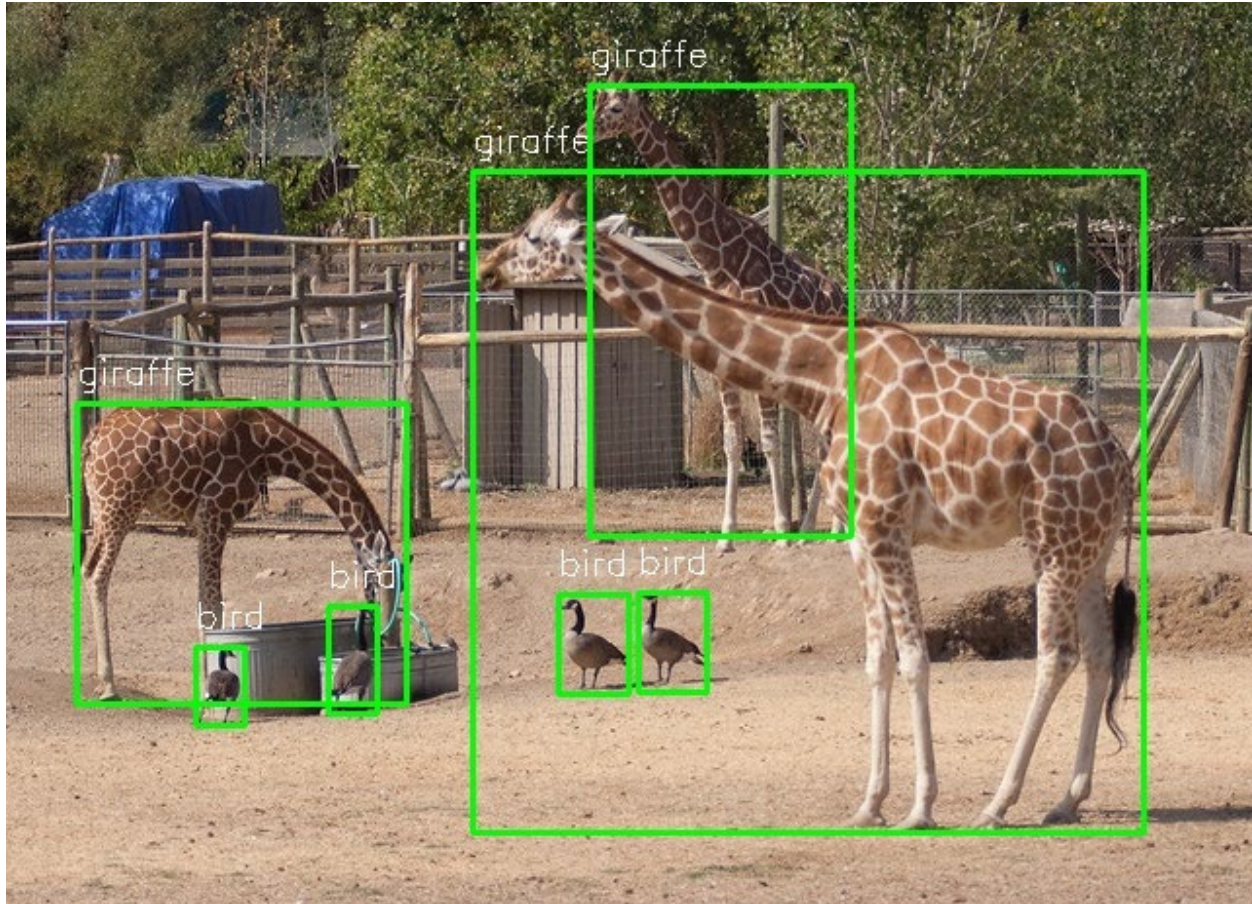


Figure 4. Output of object detection algorithm for a single image

7.2. Run Methodology

The device memory was programmed with a known pattern, irradiated, and then evaluated for single-event upsets. Several test conditions were used in this test with varying experimental controls. These tests help understand the nature of any upsets seen within the SAKURA-I chip. These experiments were the following:

1. Running repeated inferences without any data transfers with the host beyond the initial setup with one input image. This allows us to observe any degradation in the output, if any, when inferences are run without constantly updating model parameters from the host.
2. Running repeated inferences with a reloading of the model parameters after a specified number of iterations. This allows us to observe if there is a recovery from recent upsets, if any, upon a refresh from the host PC.
3. Running repeated inferences without any data transfers with the host PC beyond the initial setup with five input images. This allows us to observe if there is any input dependence to the upsets seen.

Since the host PC was not in the beam path, and any critical or relevant data is passed to the SAKURA DUT on a run start, the PC was only rebooted when there was a system hang, or communication with the card was upset. At the beginning of each new beam run, all model data and configuration were sent from the host PC to the DUT. The beam was powered on simultaneously with a run start.

For each run, the number of inferences was defined by a command line argument on start. Additionally, a command line argument was used to define the number of model parameter reloads if the run was following experiment (2). In the case of condition (3), a separate script was run that used additional input images. For experiments (1) and (2), the image with the giraffes and geese were used, shown in Figure 4. All images in Figure 3 were used in experiment (3). After a run, the fluence was recorded along with the output confidence scores for all objects detected within the image. Additionally, the confidence score for all 80 classes within the dataset for each object were also recorded to observe any changes within any other class score, even if it was not the predicted output.

7.3. Data Analysis

To analyze the data, the confidence scores of the top scoring class for each detected object was plotted against the inference iteration number. This gives a timeline of how the confidence scores change over time. The model should be deterministic without any upsets. In other words, without radiation, the plot of confidence score vs inference iteration number should be a straight, horizontal line in a golden case for each detected object. An example of this deterministic behavior can be seen in Figure 5, which was run prior to the radiation experiment.

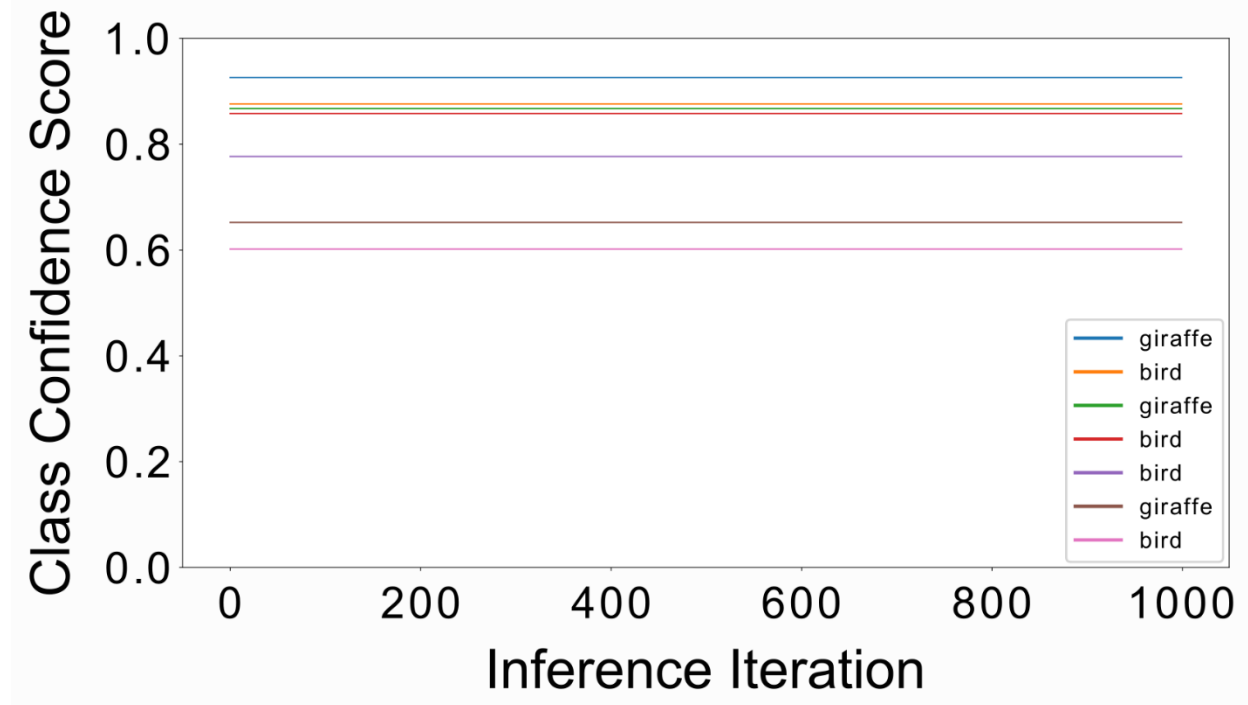


Figure 5. Golden confidence scores for one image

The confidence scores were monitored for any upsets or deviations from the expected outcome. Additionally, the bounding boxes for each object were observed for any changes in accuracy. An error was considered “tolerable” if the output class was what was expected without radiation, even if the confidence score had changed. Any deviation from these golden scores is considered an upset, even if they were tolerable errors.

8. Test Results

The SAKURA card was irradiated with only one tune at 200 MeV protons, and classification of observed upsets were made. Upsets were seen in the form of a drop or change in confidence score of the output classes. No destructive effects were observed for the entirety of the radiation test, though heavy-ion testing will be needed to perform a true survival test. Single-event functional interrupts (SEFIs) were often seen and defined as losses in the communication with the device. These errors likely consisted of either upsets within the PCIe protocol hardware in the card or a system hang which led to a loss in data telemetry.

For most of the upsets seen, any change in the confidence score was immediately recovered to its expected state on the next iteration, and therefore were mostly tolerable upsets. This is likely due to model data being transferred to the on-chip memory from the DDR memory, which was not under irradiation, on the start of every iteration, even if there are no data transfers with the host PC. These temporary upsets seemed tolerable and did not seem to impact the overall accuracy of the model across the inference iterations. However, more research and data into this would need to be collected before understanding the effects fully. An example of a run with observed upsets can be seen in Figure 6.

Tolerable upsets were seen in the form of small spikes in the confidence scores. Occasionally, an upset could be seen as an increase in the output confidence score, as seen in the blue box in Figure 6 on the pink, bird classification. Interestingly, when there were upsets that change the confidence scores, it appeared to consistently change in magnitude on subsequent errors. In other words, the output changed in the exact same magnitude every time there was an upset in a run. This can be seen especially well in the yellow box for the brown, giraffe confidence line in Figure 6.

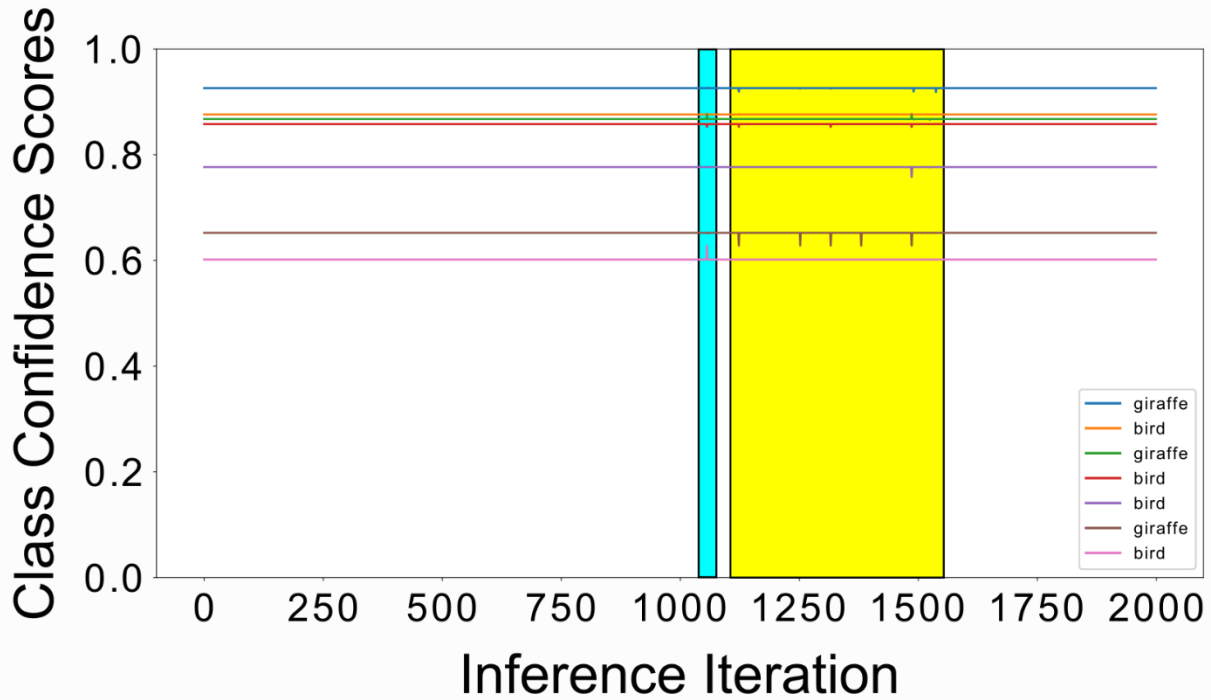


Figure 6. Example of recovered, temporary upsets. (Blue) Example of an observed upset that raised the confidence score. (Yellow) Example of magnitude drop consistency between upsets.

The majority of observed upsets were tolerable and recovered on the next iteration. However, persistent upsets were also observed. Persistent errors remained for the remainder of the run, after a memory refresh from the host PC, and even persisted to the next run without a reset in between. Some persistent errors observed were also tolerable in that they did not appear to have any misclassifications within the image, even with a change in the confidence score. An example of this kind of upset, along with the bounding box associated with a permanent upset can be seen in Figure 7. The dashed, vertical line is the specific inference that is shown in the image. The bounding boxes shown in red are the upset boxes, while the ones in blue are what they were expected to be. Even though the box sizes are different, they still correctly predict the object to an acceptable level of accuracy. It was only when the PC was power cycled, and thus the card was restarted entirely, that the system would recover. Additionally, the isolated, temporary upsets still occur even within the period where there was a persistent error. Further research and analysis into this behavior is needed to understand any sources or causes.

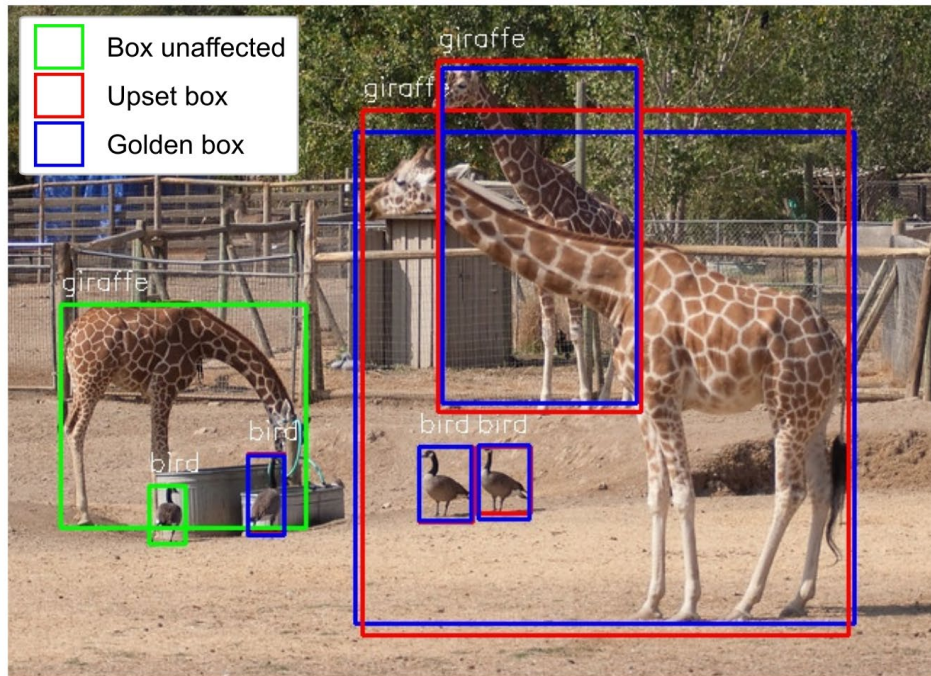
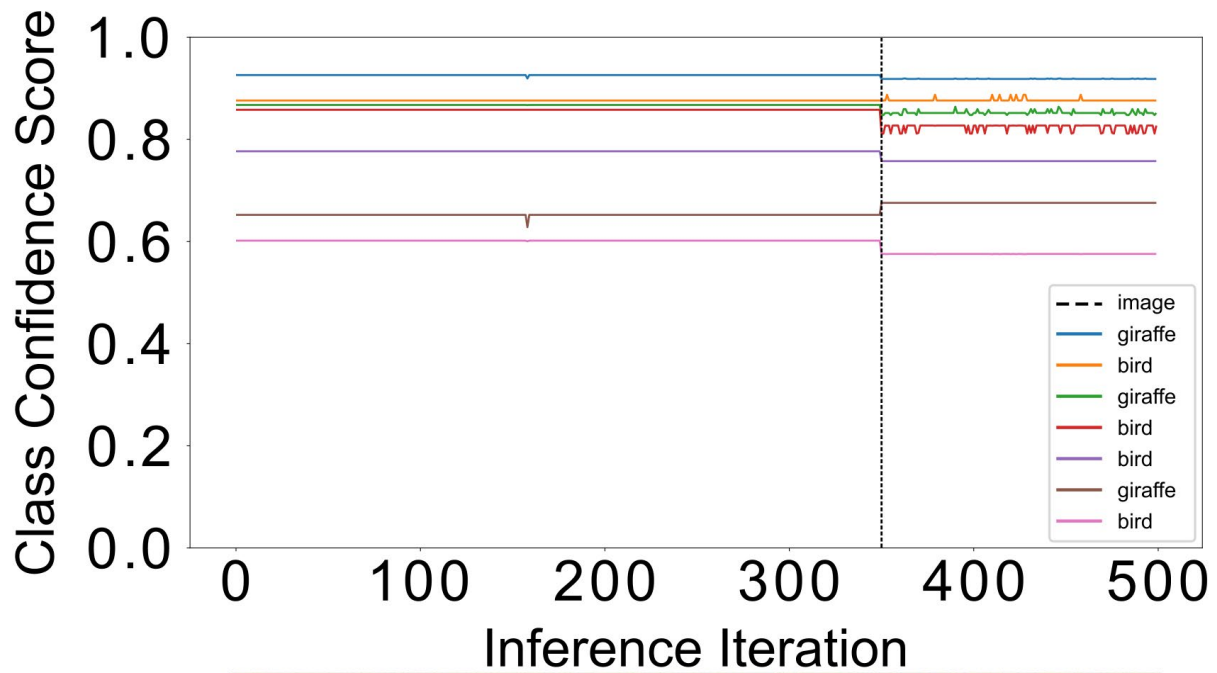


Figure 7. Example of a run with persistent errors. The dashed, vertical line indicates the inference number that is shown in both images.

There were also upsets that lead to a failure of the model. These failures are in the form of additional predictions of objects that are incorrect on top of the correct predictions, objects that are detected with the incorrect class prediction, or the failure of the model to predict objects in the image at all. These failures appeared both as temporary, non-persistent upsets that were recovered on the next inference and as persistent errors that required a reboot of the host PC to recover. An example of the temporary misprediction can be seen in Figure 8, where two of the

birds were classified as both “hot dog” and “bird,” marked in red. The persistent errors showed several incorrect predictions with very high confidence scores in random spots. Additionally, the added bounding boxes and predictions are different every inference after the first error. Finally, this error seems to be consistent among all the images used. Two examples of this persistent error can be seen in Figure 9.

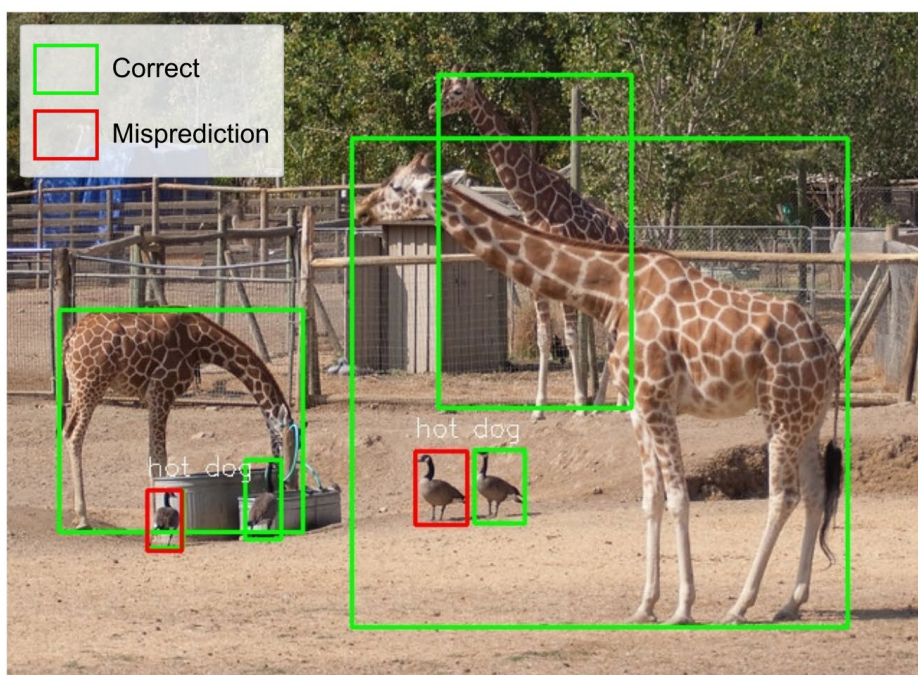
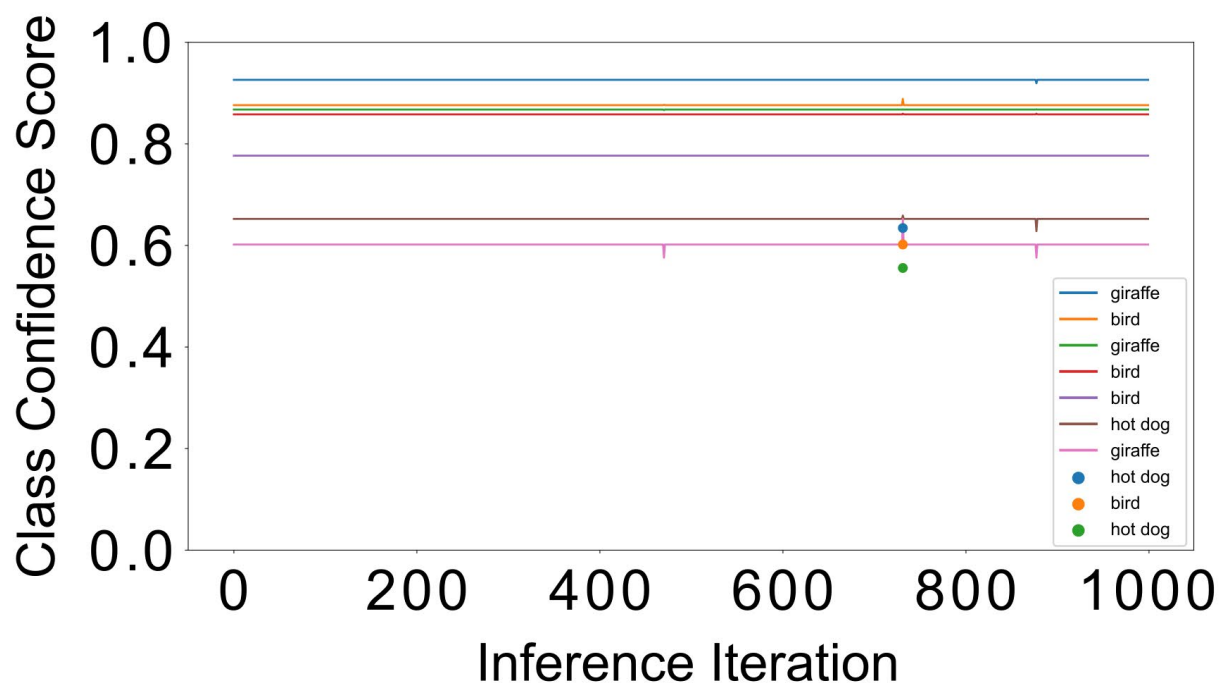


Figure 8. Example of temporary, non-persistent misprediction

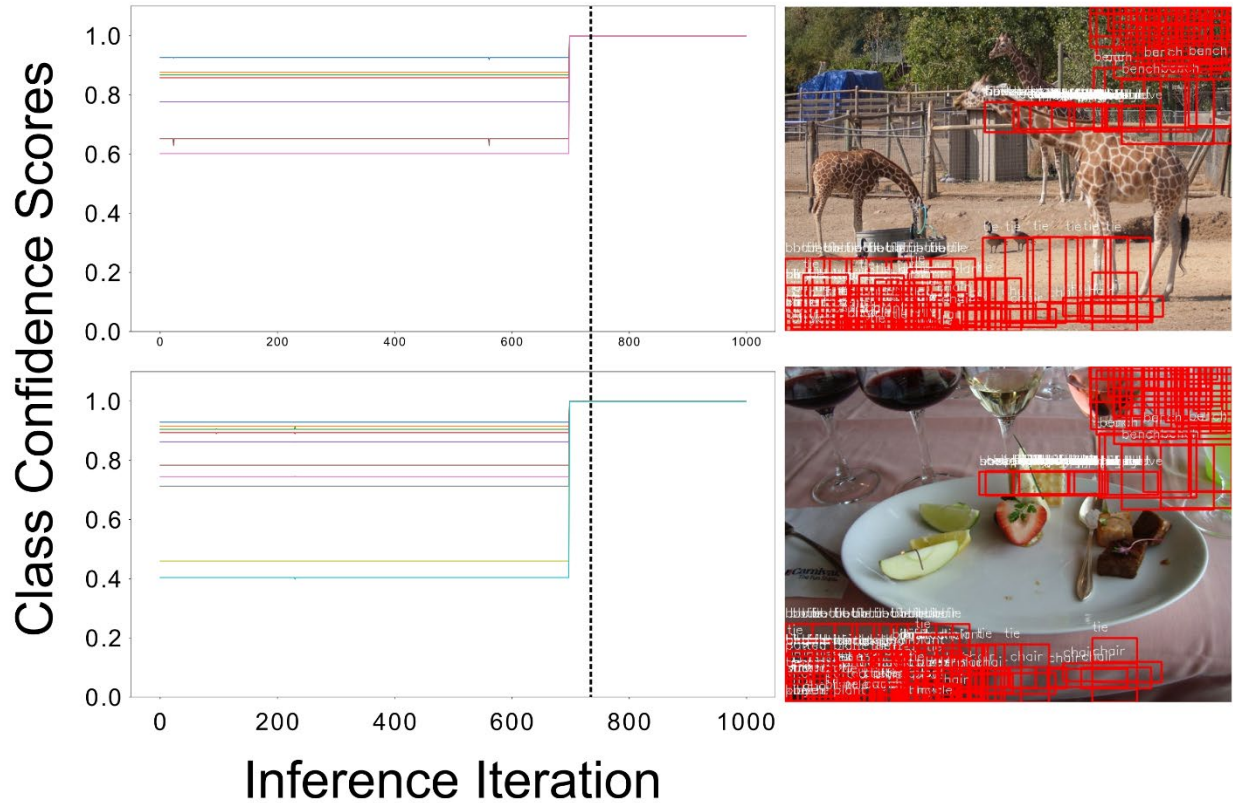


Figure 9. Persistent errors in two images with many random bounding boxes. The dashed, vertical line indicates the inference number that is shown in both images.

Out of the 126 runs that did not encounter beam or software issues, the non-persistent errors were the most common, with SEFIs being second most common. A table of the error classification frequencies can be seen in Table 2.

Table 2: Frequencies of Error Classification (N=126).

Error Classification	Frequency (N=126)
No Errors	13.5%
Non-Persistent Upsets - Tolerable	53.2%
Non-Persistent Upsets – Misprediction	0.79%
Persistent Upsets – Tolerable	3.97%
Persistent Upsets - Misprediction	2.38%
SEFI	26.2%

Both non-persistent and persistent upsets in the model can demonstrate degradation of the model parameters in memory space. In a real-world application, there will be no ground truth in an inference to show that the model is not performing as trained. However, these upsets can be detected in software by introducing images of known ground truth into the inference batch. This can be done using the golden batch refreshing method mentioned in [3]. In the case of this card, if a persistent error is detected, then a full reset will be needed to refresh the PCIe interface.

9. Conclusion

More research and data analysis is necessary to understand the upset modes of the SAKURA-I card. This report summarizes the preliminary results and classifies the observed errors seen under 200-MeV protons. Most of the SEUs observed were temporary decreases in the class confidence scores which recovered on the next inference iteration. However, some upsets saw persistent errors in the model which required a power cycle of the host PC. Additional analysis of this data is needed to understand the causes of the observed errors, and more radiation testing is needed to fully classify the upset modes of EdgeCortex SAKURA-I.

10. References

- [1] EdgeCortex, “SAKURA-I Edge AI Accelerator,” April 2024.
- [2] G. Jocher *et al.*, “ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.” Zenodo, Nov. 22, 2022. doi: [10.5281/ZENODO.3908559](https://doi.org/10.5281/ZENODO.3908559).
- [3] Garrett, Tyler, Seth Roffe, and Alan George. "Soft-Error Characterization and Mitigation Strategies for Edge Tensor Processing Units in Space." *IEEE Transactions on Aerospace and Electronic Systems* (2024).

