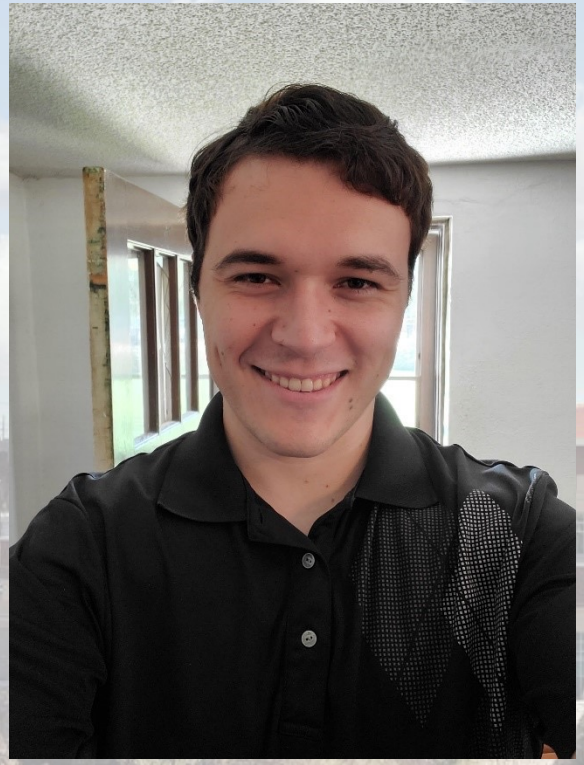
Microprocessor Reliability Enhancement Under Ionizing Radiation using Performance Counters

Antonio Teijeiro, MSEE
Advisor: Eric MacDonald, Ph.D.
University of Texas at El Paso
(915) 205-0648
aeteijeiro@gmail.com

Personal Background



- BSEE, MSEE from UTEP
- Year-Round Radiation Effects Experimentation Intern at Sandia National Laboratories
- Doctorate research on performance counter-based microprocessor reliability enhancement in high radiation environments

Introduction

Theoretical Foundations

System Design

Experimentation

Results

Discussion

Conclusion and Future Work

The Problem
Processing Power in Space

- Increased processing power required for global security and scientific advancement.
  - LEO image processing
  - Artificial Intelligence (AI)
  - Robotics for extreme environments
  - Space exploration

 [1]

[1]https://www.cpushack.com/2012/08/14/spacecraft-processors-mars-curiosity/rad750/

Radiation Effects

Solutions

- Computational redundancy
  - Duplication of work for error detection, triplication for error correction [1]

- Novel algorithms, additional machine learning layers promising but require refactoring [2], [3], [4].
  - Algorithm/layer must be tailored to a particular problem.
  - Require access to source code.

Radiation Effects

Proposed Solution

- Utilize hardware performance counters for error detection [5]
  - 2702 possible events to monitor
  - 16 performance counters
  - Practical limitation to ≤ 10 at a time.
  - Used for quality control, GPU software profiling

Introduction

Theoretical Foundations

System Design

Experimentation

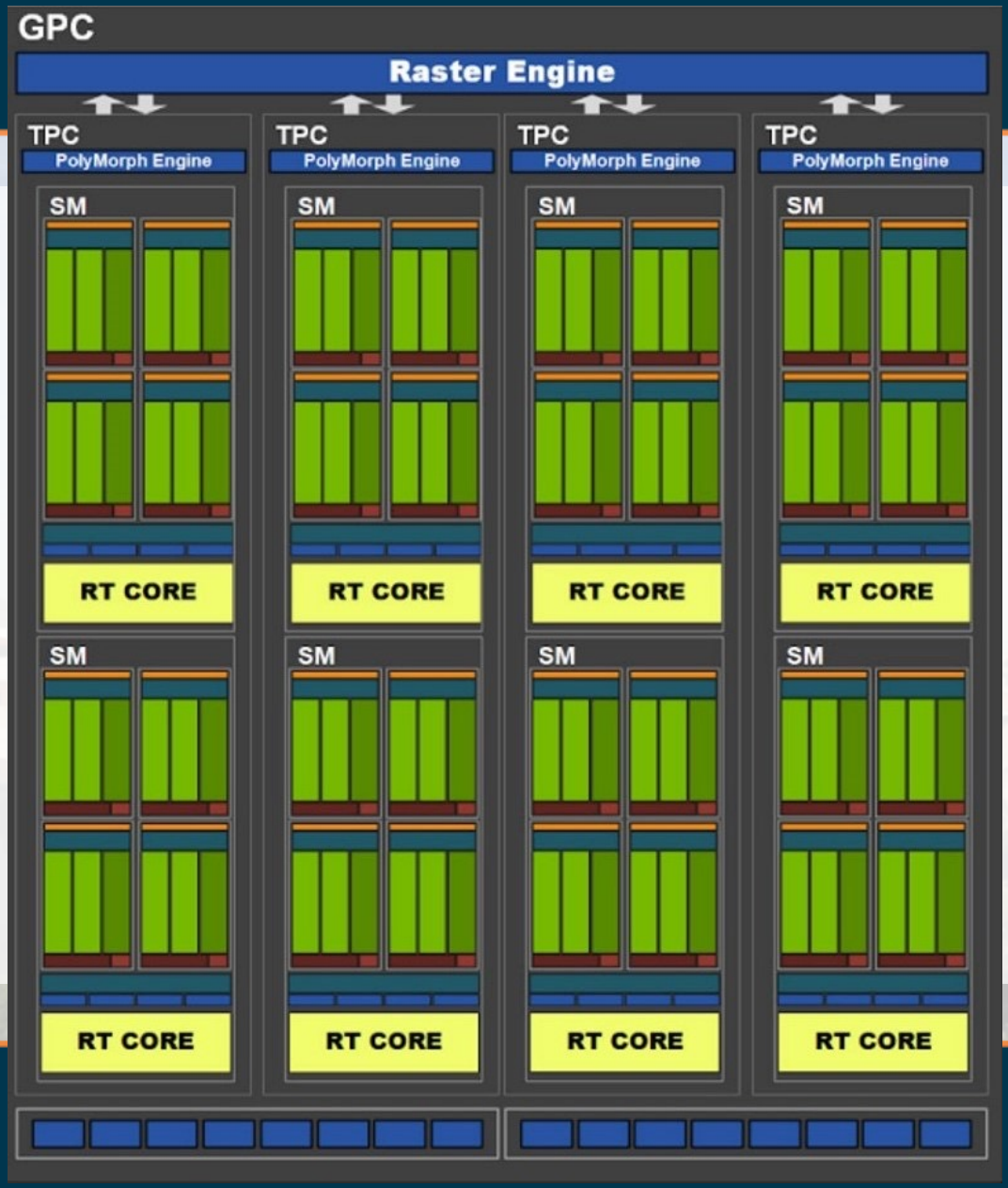
Results

Discussion

Conclusion and Future Work

GPU microarchitecture

Full GPU



[1]

[1]https://www.nvidia.com/content/PDF/nvidia-ampere-ga-102-gpu-architecture-whitepaper-v2.pdf

GPU microarchitecture

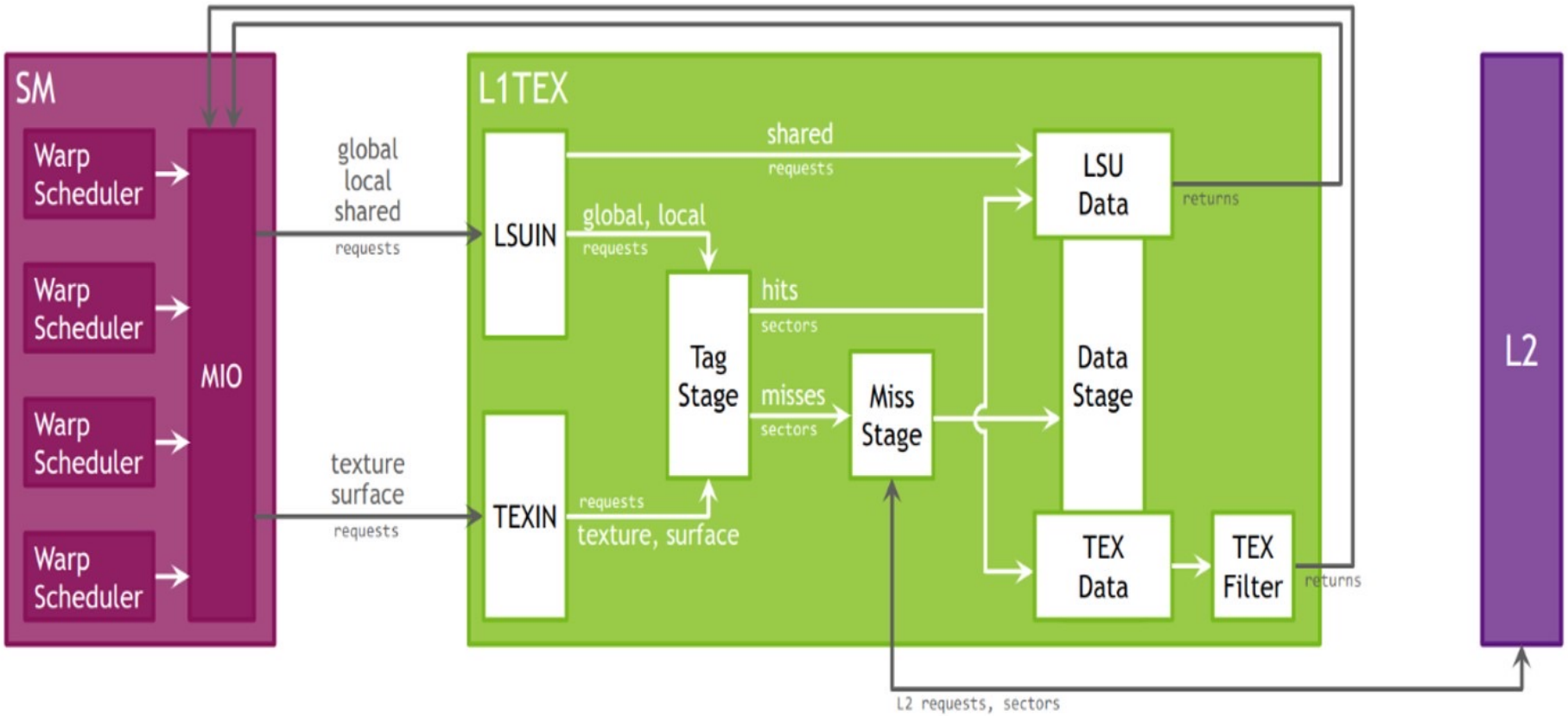
Single SM

[1]

[1]https://www.nvidia.com/content/PDF/nvidia-ampere-ga-102-gpu-architecture-whitepaper-v2.pdf

GPU microarchitecture

Low-Level Memory Hierarchy



[1]https://docs.nvidia.com/nsight-compute/ProfilingGuide/index.html#metrics-structure

Hardware Profiling

Events, Counters, and Metrics

- *Event* refers to some hardware condition in the GPU.
- *Counters* are special registers which measure event *activity.*
- *Metrics* are …
- 2702 hardware events available for monitoring the GA102
  - 4 rollups per event [6]
    - sum, average, min, max
    - 18 sub-metrics appliable to each rollup
  - 17 possible units
  - 15 possible subunits
  - 13 possible pipestages

Hardware Profiling
Events, Counters, and Metrics

- *Event:*

  smsp__pipe_fma_cycles_active

- *Counters:*

  %pm0 to %pm7

  %pm0_64 to %pm7_64

- *Metrics:*

  smsp__pipe_fma_cycles_active.sum

Introduction

Theoretical Foundations

System Design

Experimentation

Results

Discussion

Conclusion and Future Work

The Benchmark
PAPI integration with GPU4S

- Focusing on the matrix_multiplication_bench benchmark within GPU4S [7].
- PAPI is an open-source performance counter configuration and reading tool.
- Counters are read once before and once after kernel launch.

CUDA compilation

- Compilation to PTX code, JIT compiled to SASS code on the GPU, then to executable.

- Process is broken up by passing the -dryrun and –keep options to the nvcc line.
  - Error injection instructions are inserted into PTX code, prior to the SASS generation step.

CUDA compilation

```
mov.u32      %r47, %ntid.y;
mov.u32      %r48, %ctaid.y;
mul.lo.s32   %r2, %r48, %r47;
mov.u32      %r3, %tid.y;
add.s32      %r4, %r2, %r3;
setp.ge.u32  %p1, %r1, %r41;
setp.ge.u32  %p2, %r4, %r43;
or.pred      %p3, %p1, %p2;
@%p3 bra     $L__BB0_9;
```

```
mov.u32      mystoreCTAx, %ctaid.x;
mov.u32      mystoreCTAy, %ctaid.y;
mov.u32      mystorewarpID, %warpid;
setp.lt.u32  mycondition, mystoreCTAx,1;
@mycondition setp.eq.u32  mycondition, mystoreCTAy,0;
@mycondition setp.eq.u32  mycondition, mystorewarpID,0;
@mycondition xor.pred     %p3, %p3, 1;
@%p3 bra     $L__BB0_9;
```

Introduction

Theoretical Foundations

System Design

Experimentation

Results

Discussion

Conclusion and Future Work

Hypothesis

- Upsets affecting L0 instruction cache, LSU, or ALU hardware can incite cache misses
  - Will be detectable from miss-stage, frame buffer, and DRAM activity.
- Upsets affecting the FMA pipeline will be reflected in abnormal FMA activity.
- Upsets affecting the ADU will be reflected in abnormal FMA activity

GPU microarchitecture

Single SM

Procedure

1. L0 cache, LSU, and ALU are all targeted with 2^20 XOR in the destination address used in the store instruction.
   - L0 cache targeted at the warp level.
   - LSU targeted at the quarter-warp level
   - ALU targeted at the lane level.
2. FMA targeted by the addition of 100,000 to thread index calculation results at the warp level.
3. ADU targeted by logical XOR with predicate guard condition used for early thread exits.
4. Perform 1000 golden, 1000 injected runs.
5. Noise removal, apply machine learning

Introduction

Theoretical Foundations

System Design

Experimentation

Results

Discussion

Conclusion and Future Work

Best Metrics
SEU

TABLE I. BEST SDC DETECTION RESULTS

Targeted Hardware	Metric	3-fold CV Accuracy	LOF Accuracy
L0 i-Cache	lts__t_requests_aperture_device_evict_normal_lookup_miss.sum	[0.96701649 0.97601199 0.96696697]	91.55%
LSU	lts__t_requests_aperture_device_evict_normal_lookup_miss.sum	[0.93553223 0.93703148 0.95495495]	72.6%
ALU	lts__t_requests_aperture_device_evict_normal_lookup_miss.sum	[0.86656672 0.87556222 0.87987988]	49.95%
FMA	sm__pipe_fma_cycles_active.sum	[1. 1. 1.]	100.0
ADU	sm__pipe_fma_cycles_active.sum	[1. 1. 1.]	100.0

Hypothesized Metrics
SEU

TABLE II. LESS SENSITIVE METRICS (SEU)

Targeted Hardware	Metric	3-fold CV Accuracy	LOF Accuracy
L0 i-Cache	l1tex__t_bytes_pipe_lsu_lookup_miss.sum	[0.79910045 0.77961019 0.75825826]	69.8%
L0 i-Cache	fbpa__dram_write_bytes.sum	[0.48275862 0.48125937 0.51201201]	48.8%
LSU	l1tex__t_bytes_pipe_lsu_lookup_miss.sum	[0.63568216 0.63118441 0.63963964]	53.45%
LSU	fbpa__dram_write_bytes.sum	[0.48575712 0.47826087 0.503003]	51.05%
ALU	l1tex__t_bytes_pipe_lsu_lookup_miss.sum	[0.57721139 0.5892054 0.56606607]	49.1%
ALU	fbpa__dram_write_bytes.sum	[0.55922039 0.51124438 0.51951952]	50.25%

Hypothesized Metrics
MEU (Repeated SEU)

TABLE III. LESS SENSITIVE METRICS, REPEATED SEU

Targeted Hardware	Metric	3-fold CV Accuracy	LOF Accuracy
L0 i-Cache	l1tex__t_bytes_pipe_lsu_lookup_miss.sum	[1. 1. 1.]	93.85%
L0 i-Cache	fbpa__dram_write_bytes.sum	[0.98650675 0.98350825 0.98048048]	91.6%
LSU	l1tex__t_bytes_pipe_lsu_lookup_miss.sum	[1. 1. 1.]	98.85%
LSU	fbpa__dram_write_bytes.sum	[0.95352324 0.93853073 0.95195195]	86.75%
ALU	l1tex__t_bytes_pipe_lsu_lookup_miss.sum	[0.98650675 0.988006 0.996997]]	98.35%
ALU	fbpa__dram_write_bytes.sum	[0.7946027 0.79610195 0.80930931]	70.3.%

Introduction

Theoretical Foundations

System Design

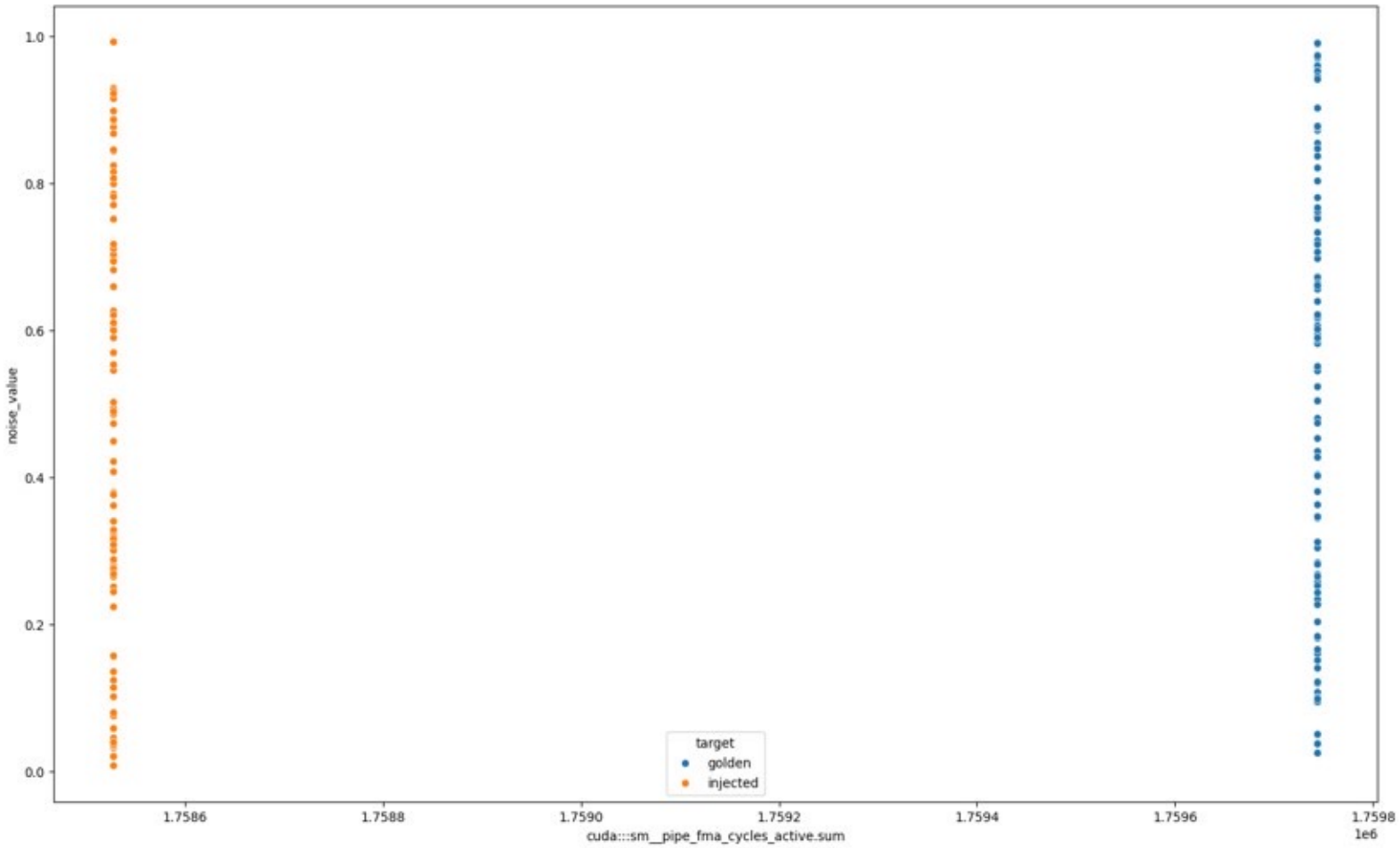
Experimentation

Results

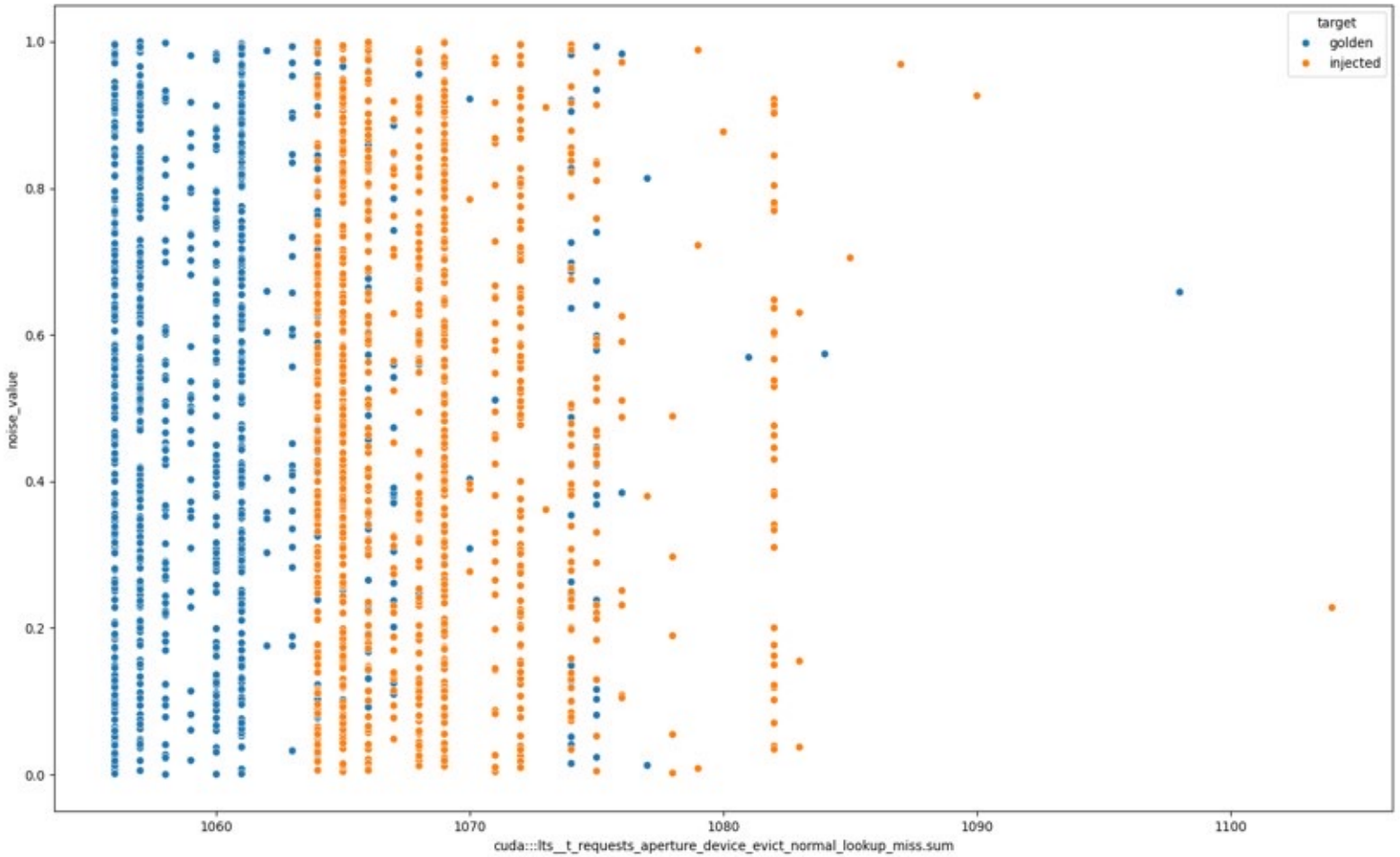
Discussion

Conclusion and Future Work

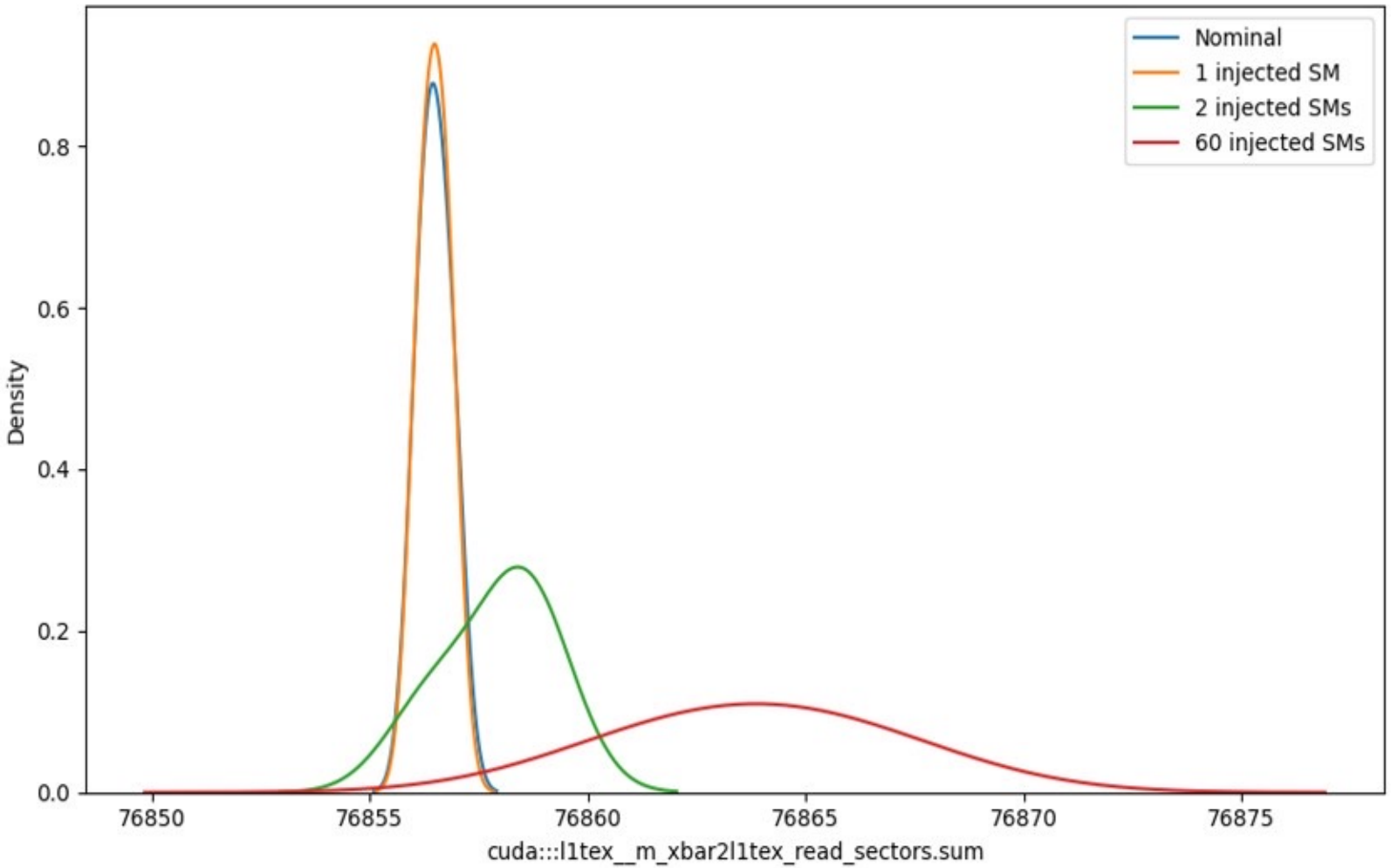
Overfitting Analysis



Overfitting Analysis



Overfitting Analysis



Introduction

Theoretical Foundations

System Design

Experimentation

Results

Discussion

Conclusion and Future Work

Conclusion

- Certain hardware is more difficult, but not impossible to monitor
- LOF classification is useful for error detection training without fault injection.
- Perfect accuracy is possible for deterministic metrics

Future Work

1. Expansion to remaining hardware
2. Consider regression to older GPU with CUPTI compatibility
3. Test under low levels of radiation
4. Apply in conjunction with progressive SM deactivation

Questions?

References

1. Iturbe, X., Keymeulen, D., Ozer, E., Yiu, P., Berisford, D., Hand, K., & Carlson, R. (2015). An integrated SOC for Science Data Processing in next-generation space flight instruments avionics. *2015 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*. https://doi.org/10.1109/vlsi-soc.2015.7314405.
2. Santos, F. F., Pimenta, P. F., Lunardi, C., Draghetti, L., Carro, L., Kaeli, D., & Rech, P. (2019b). Analyzing and increasing the reliability of convolutional neural networks on gpus. *IEEE Transactions on Reliability*, *68*(2), 663–677. https://doi.org/10.1109/tr.2018.2878387
3. Sabena, D., Reorda, M. S., Sterpone, L., Rech, P., & Carro, L. (2013). On the evaluation of soft-errors detection techniques for gpgpus. *2013 8th IEEE Design and Test Symposium*. https://doi.org/10.1109/idt.2013.6727092
4. So, H., Didehban, M., Shrivastava, A., & Lee, K. (2019). A software-level redundant multithreading for soft/hard error detection and recovery. *2019 Design, Automation & Test in Europe Conference & Exhibition*.

References

5. Guerrero-Balaguera, J.-D., Condia, J. E., & Reorda, M. S. (2021). Using hardware performance counters to support infield GPU testing. 2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS). https://doi.org/10.1109/icecs53924.2021.9665511
6. *Kernel Profiling Guide*. NVIDIA Documentation Hub - NVIDIA Docs. (n.d.-a). https://docs.nvidia.com/nsight-compute/ProfilingGuide/index.html#metrics-structure
7. Kosmidis, L., Rodriguez, I., Jover, Á., Alcaide, S., Lachaize, J., Abella, J., Notebaert, O., Cazorla, F. J., & Steenari, D. (2020). GPU4S: Embedded gpus in space - latest project updates. *Microprocessors and Microsystems*, *77*, 103143. https://doi.org/10.1016/j.micpro.2020.103143