Understanding Error Propagation in Deep Learning Neural Network (DNN) Accelerators and Applications

Guanpeng (Justin) Li,

Karthik Pattabiraman

Siva Kumar Sastry Hari,
Michael Sullivan, Tim Tsai,
Joel Emer, Stephen Keckler

My Research

- **Building error resilient and secure software systems**
- **Three main areas:**
  - Error Resilience Techniques [DSN'18A[DSN'18B]][SC'17] [DSN'17][SC'16][DSN'16][DSN'15][DSN'14][DSN'13][DSN'12]
  - Software Reliability Engineering [ICSE'18][ASE'17][ICSE'16] [ICSE'15][ICSE'14A][ICSE'14B][ASE'14][ASE'15][ESEM'13]
  - IoT Security [FSE'17][ACSAC'16][EDCC'15][HASE'14]
- **This talk**
  - Error Resilience Techniques

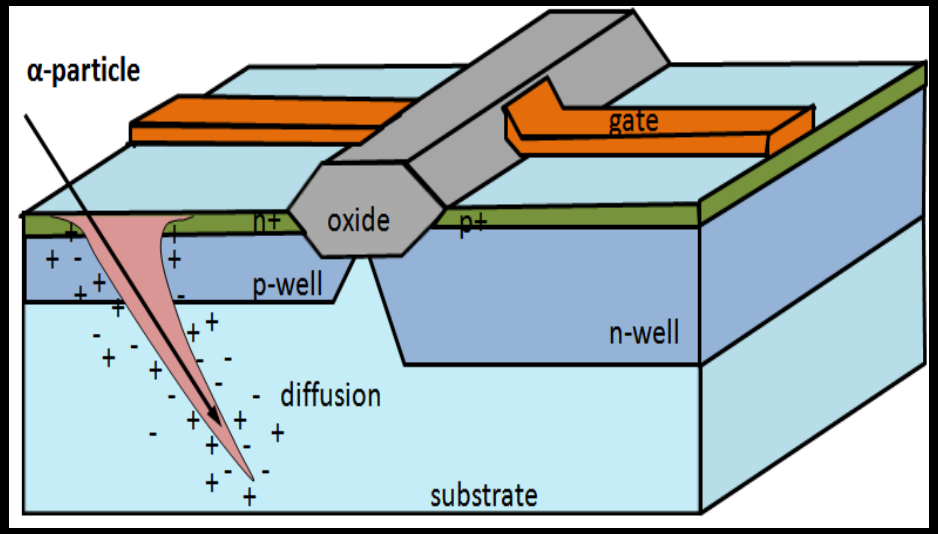
Motivation

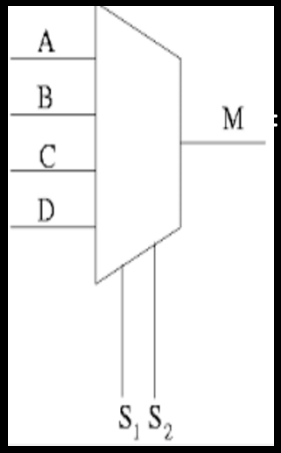
- **Neural network applications are widely deployed nowadays**
  - Deep learning neural network (DNN): Robots, Cars, Data centers
- **DNN accelerators are crucial**
  - High throughput for real-time inferencing
  - Nvidia NVDLA and Google TPU

Motivation

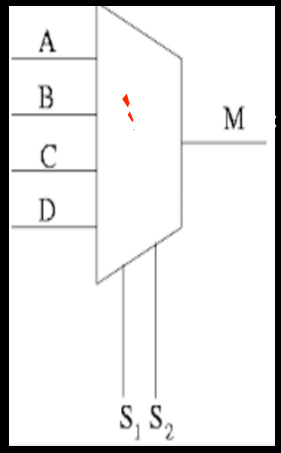
- **DNN applications are widely deployed in safety critical applications**
  - Self-driving cars – specialized accelerators for real-time processing
- **Silent Data Corruptions (SDCs)**
  - Results in wrong prediction of DNN application
  - Safety standard requires SoC FIT<10 overall (ISO 26262)

Soft Errors





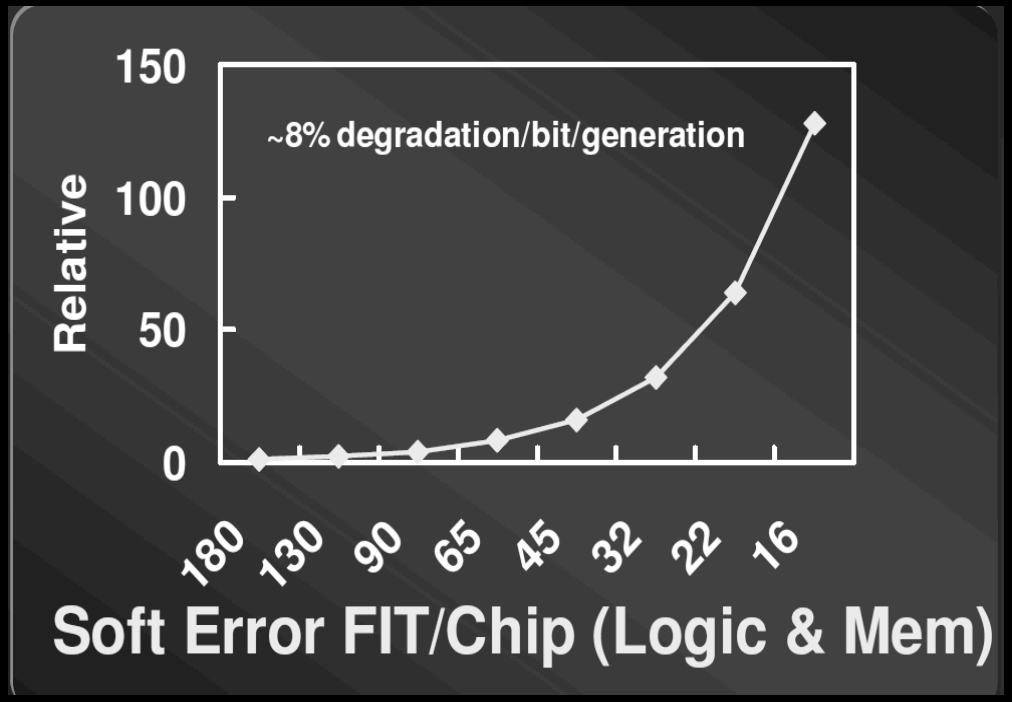
= 0001



= 0101

Soft Error Problem

- Soft errors are increasing in computer systems



Source: Shekar Borkar (Intel) - Stanford talk

Current Solutions

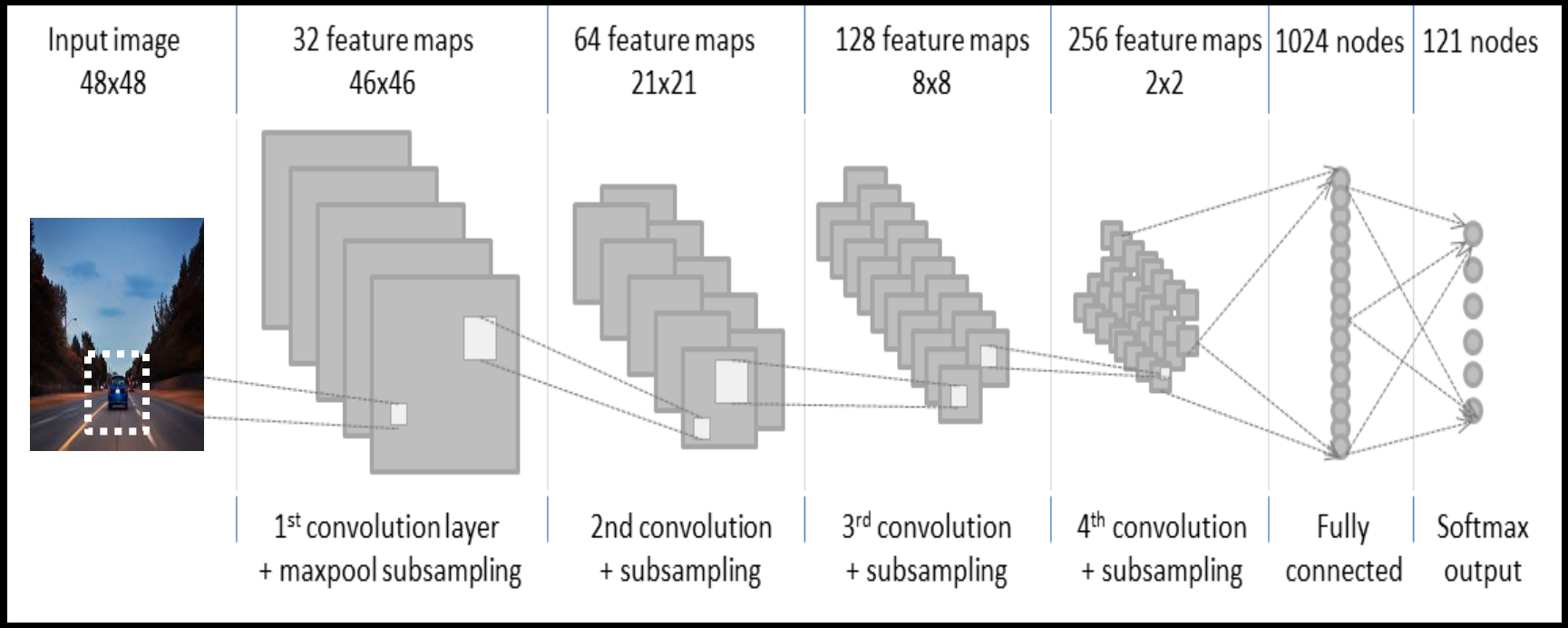
- **Traditional Solutions**
  - DMR for all latches in execution units
  - ECC/Parity on all storage elements

Incurs high overhead

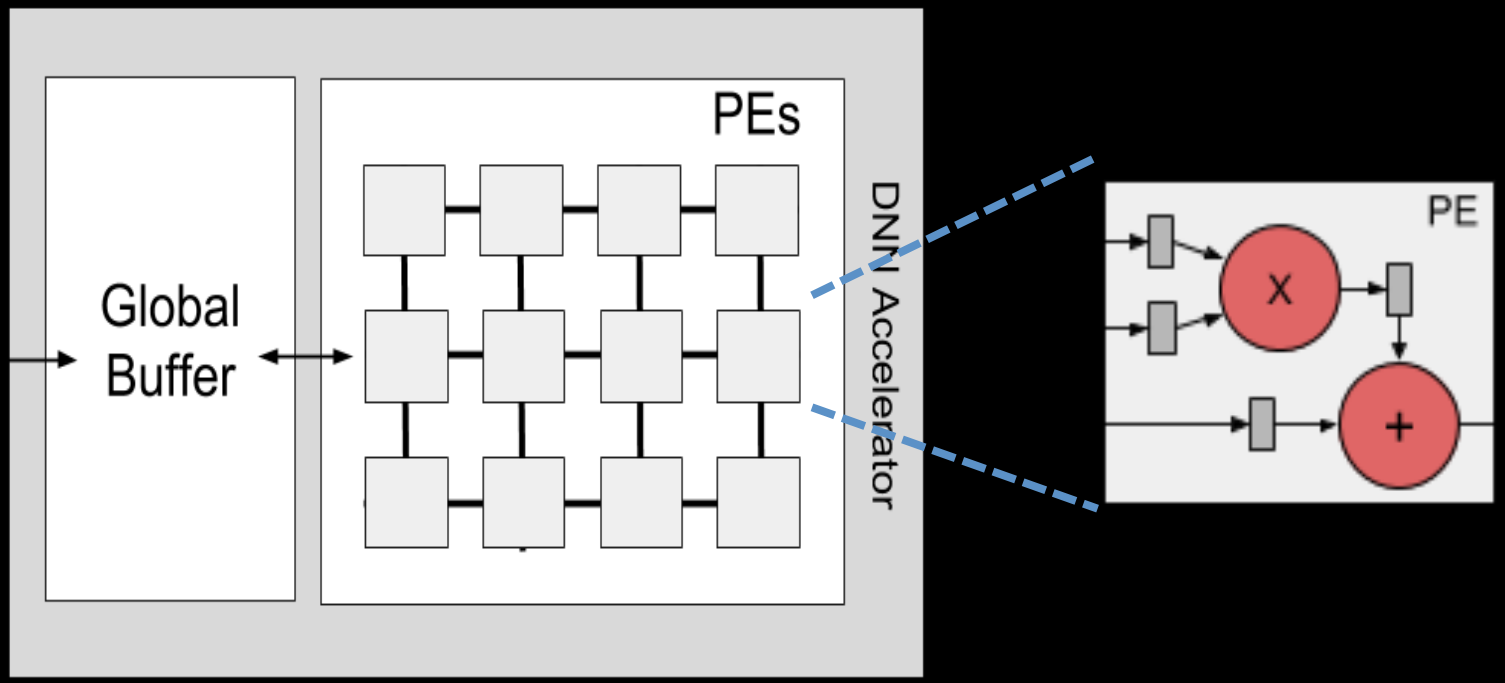
- **Recent Work**
  - Generic micro-architectural solutions
  - DNN-algorithm agnostic

Nonoptimal for DNN systems

Deep learning Neural Network (DNN)



DNN Accelerator Architecture (e.g., Eyeriss – MIT)



Goal

- **Understand error propagation in DNN accelerators - fault injection**
  - Quantification
  - Characterization

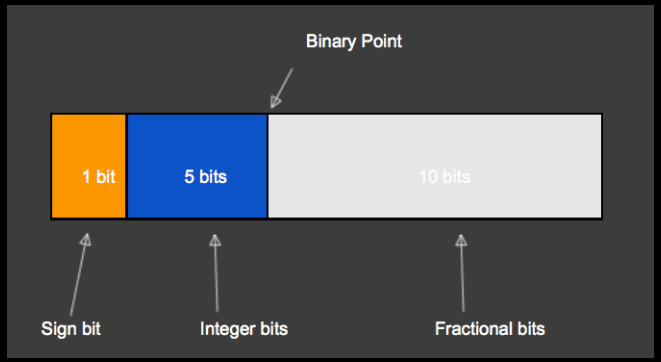
- **Based on the insights, mitigate failures:**
  - Efficient way to detect errors
  - Hardware: Selective duplication
  - Software: Symptom-based detection

Fault Injection: Parameters

- **DNNs**

Network	Dataset	No. of Output Candidates	Topology
ConvNet	CIFAR-10	10	3 CONV + 2 FC
AlexNet	ImageNet	1,000	5 CONV(with LRN) + 3 FC
CaffeNet	ImageNet	1,000	5 CONV(with LRN) + 3 FC
NiN	ImageNet	1,000	12 CONV

- **Data Types**
  - Fixed Point (FxP): 16-bit and 32-bit
  - Float Point (FP): Full- and half-precision

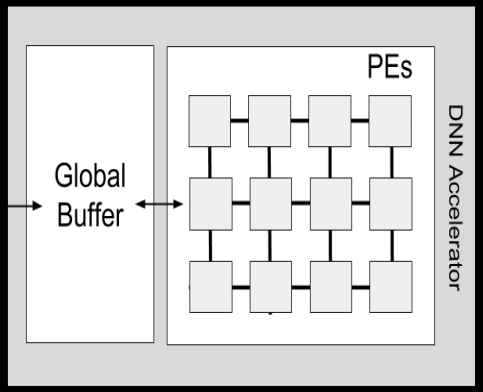


Fault Injection Study: Setup

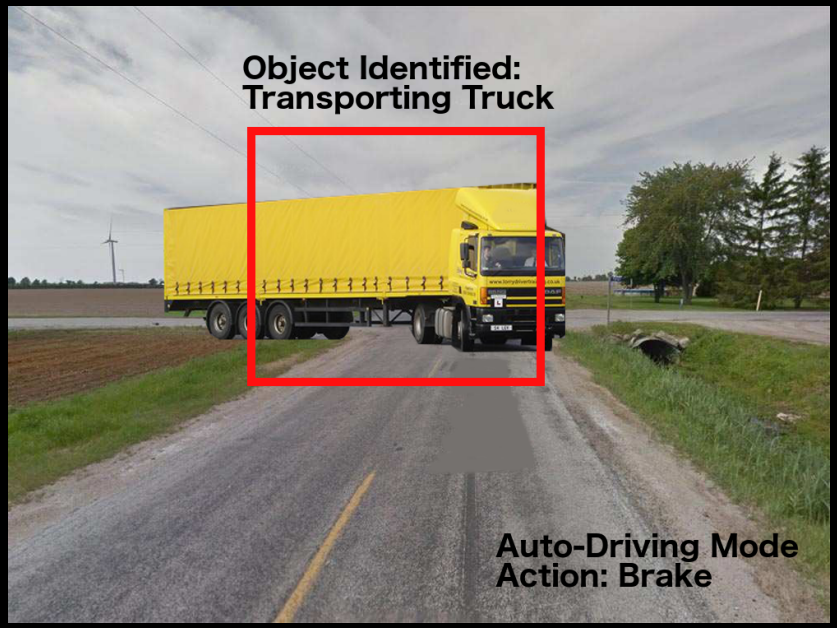
- **Fault Injection**
  - 3,000 random faults per each latch in each layer
- **Simulator**
  - DNN simulation in Tiny-CNN in C
  - Fault injections at C line code

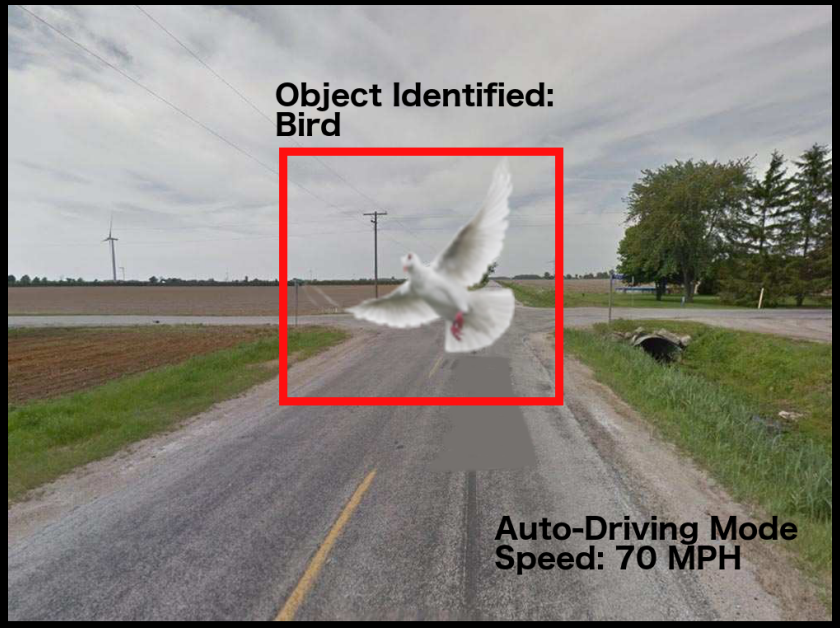
```
...
foreach layer:
  ...
  foreach weight:
    ...
    foreach input:
      ...
      R_L2.2 = inject_fault(R_L2.2)
      R_L3 = R_L2.2 + R_L5
      ...
...
```

- **Fault Model**
  - Transient single bit-flip
  - Execution Units: Latches
  - Storage: buffer SRAM, scratch pad, REG



Silent Data Corruption (SDC) Consequences





A single bit-flip error → misclassification of image by the DNN

Characterization: Research Questions

- RQ1: What are SDC rates in different DNNs using different data types?
- RQ2: Which bits are sensitive to SDCs in different data types?
- RQ3: How do errors affect values that result in SDCs?
- RQ4: How does an error propagate layer by layer?

SDC Types

SDC1:

- Mismatch between winners in faulty and fault-free execution

SDC5:

- Winner is not in top 5 predictions in the faulty execution

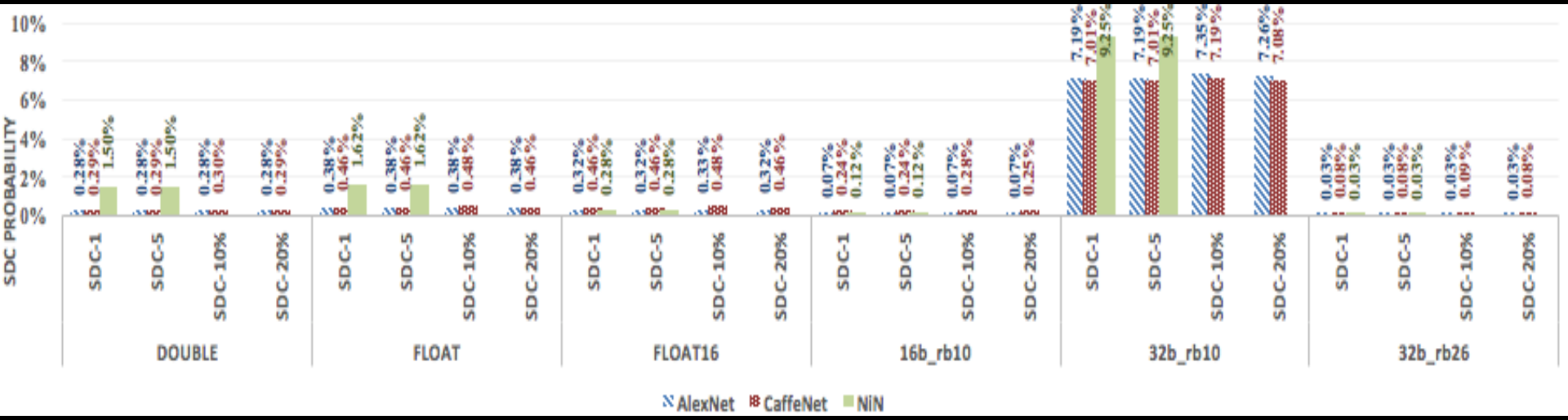
SDC10%:

- Confidence of the winner drops more than 10%

SDC20%:

- Confidence of the winner drops more than 20%

RQ1: SDC in DNNs



1.All SDCs defined have similar SDC probabilities

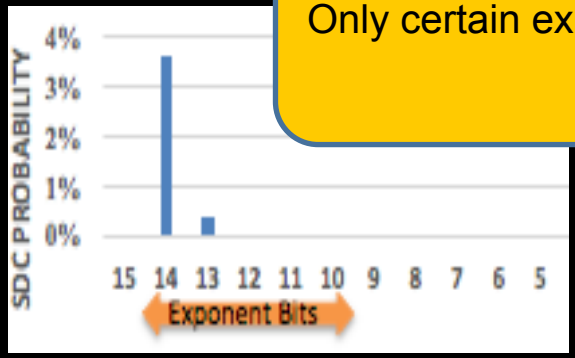
2.SDC probabilities are different in different DNNs

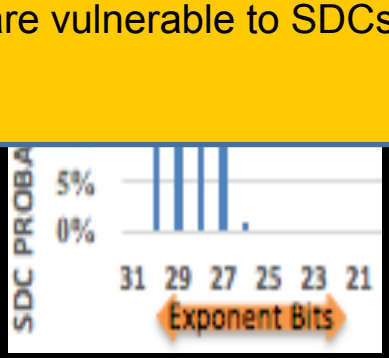
3.SDC probabilities vary a lot using different data types

RQ2: Bit Sensitivity

FP data types:

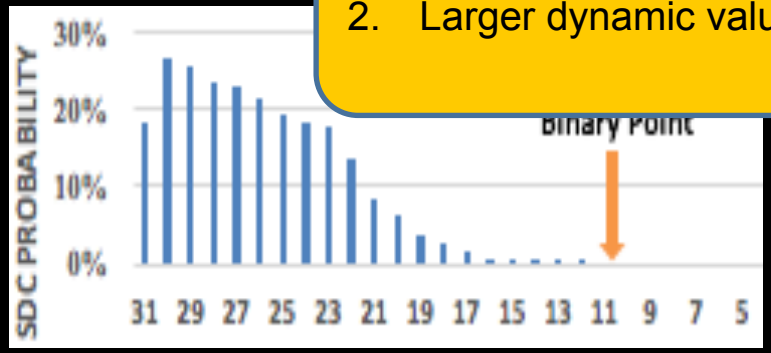
Only certain exponent bits are vulnerable to SDCs

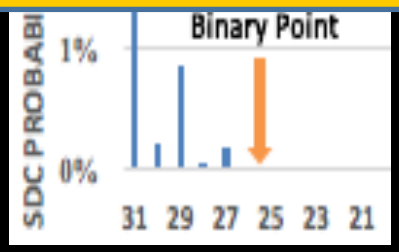




FxP data types:

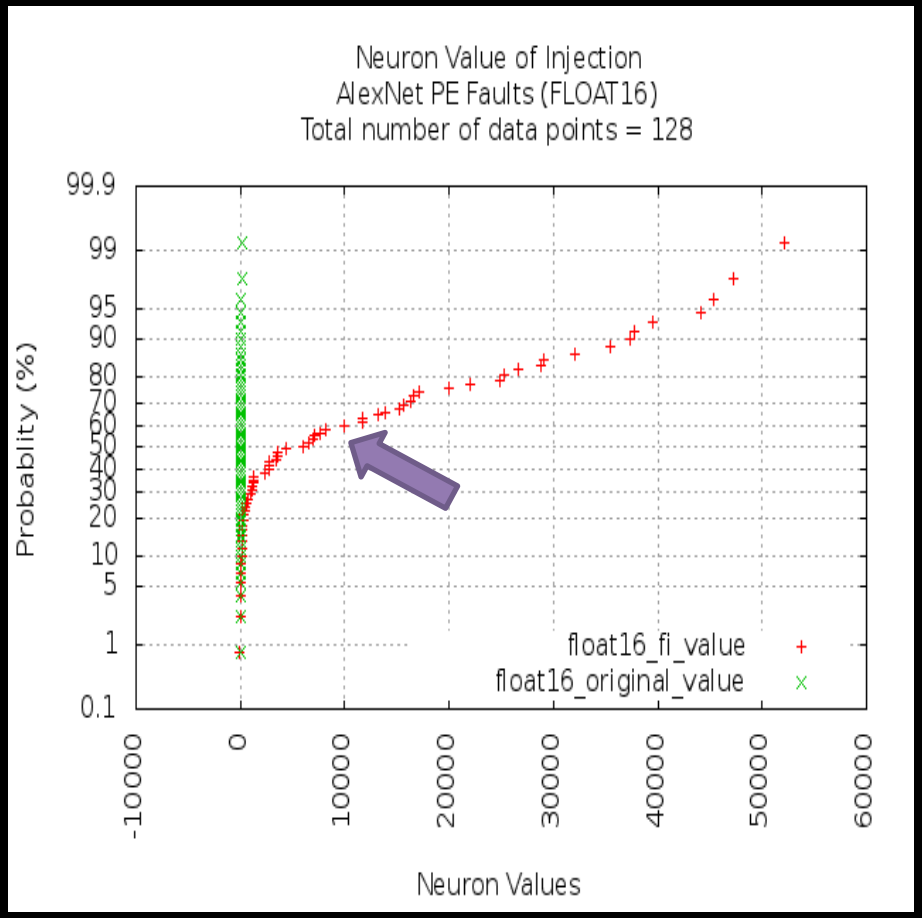
1. High-order bits are vulnerable
2. Larger dynamic value range allows more vulnerable bits



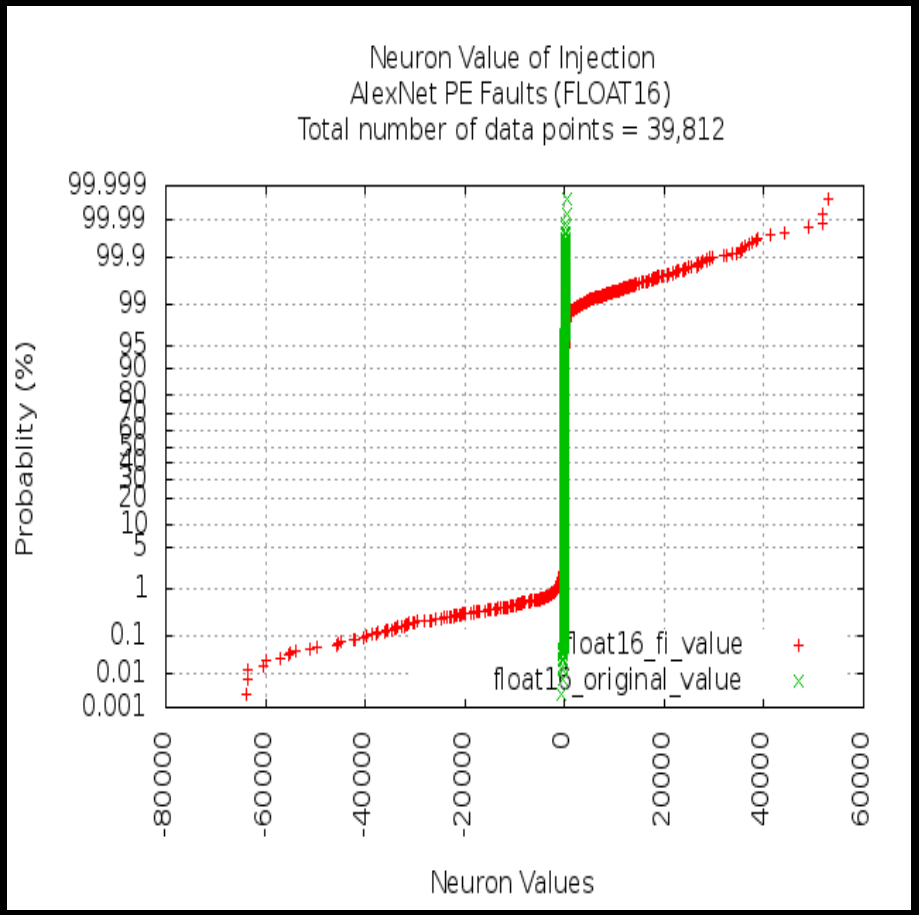


RQ3: Value Changes

AlexNet, PE Errors, Float16



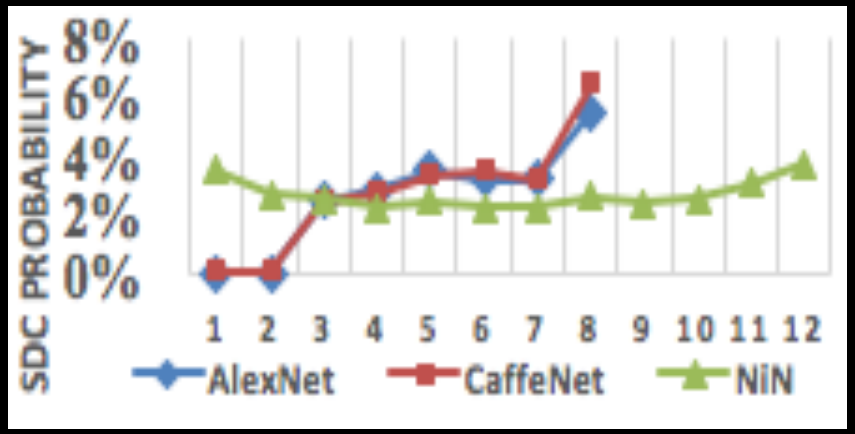
SDC

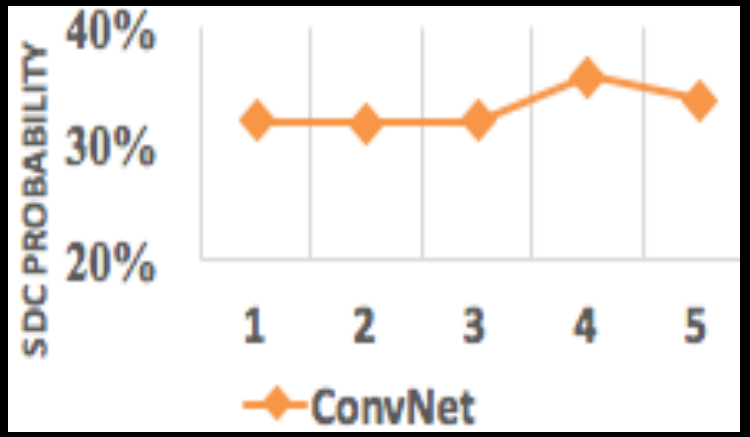


Benign

If a neuron value is changed to be a large value under a fault, it likely causes SDC

RQ4: SDC in Different Layers





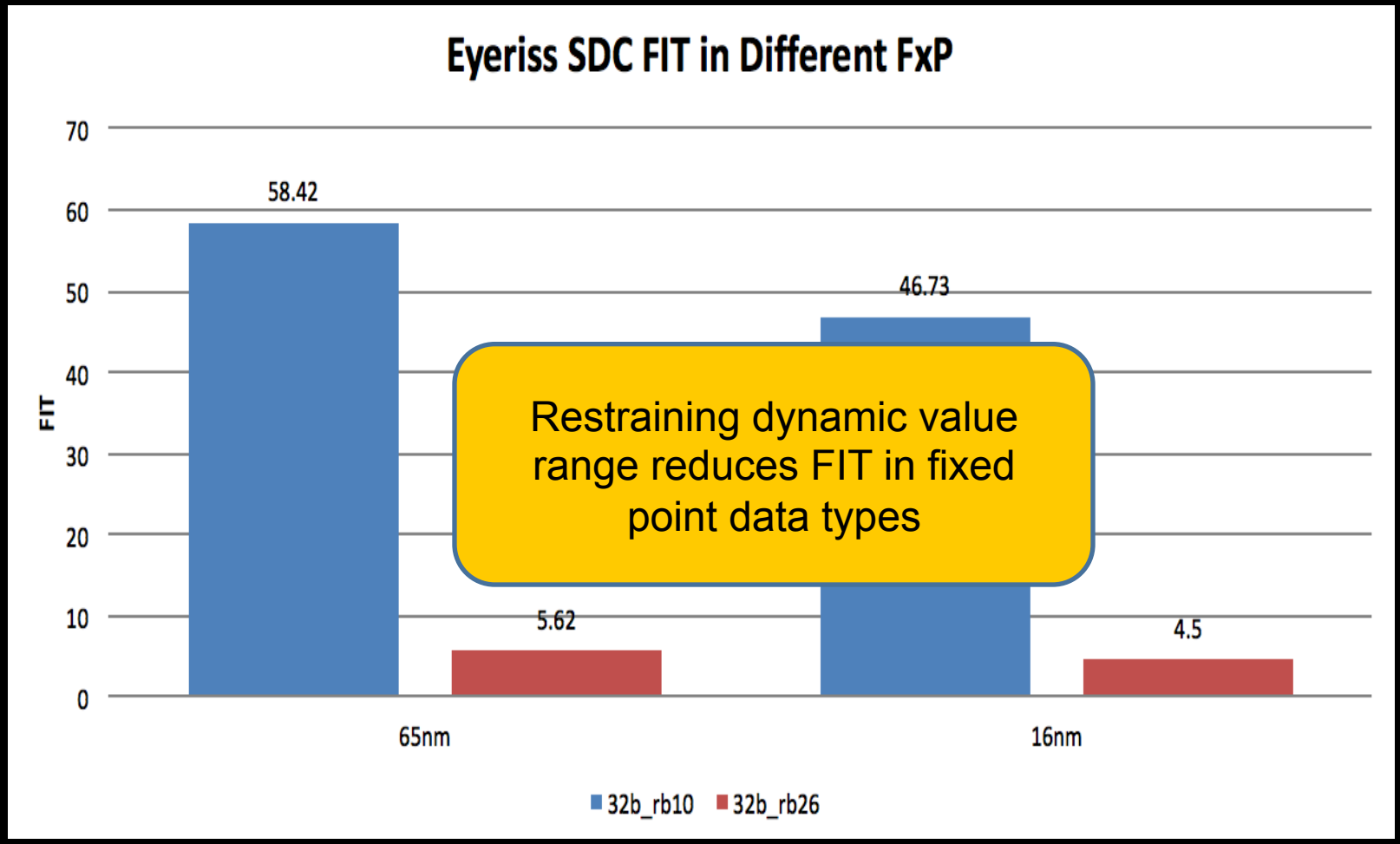
1.Layers 1&2 have lower SDC probabilities in AlexNet and CaffeNet

2.SDC probability increases as layer numbers increase

Mitigation Techniques

- Data type choice
- Symptom-based Error Detection
- Selective Latch Hardening
- Algorithmic Error Resilience (Ongoing)

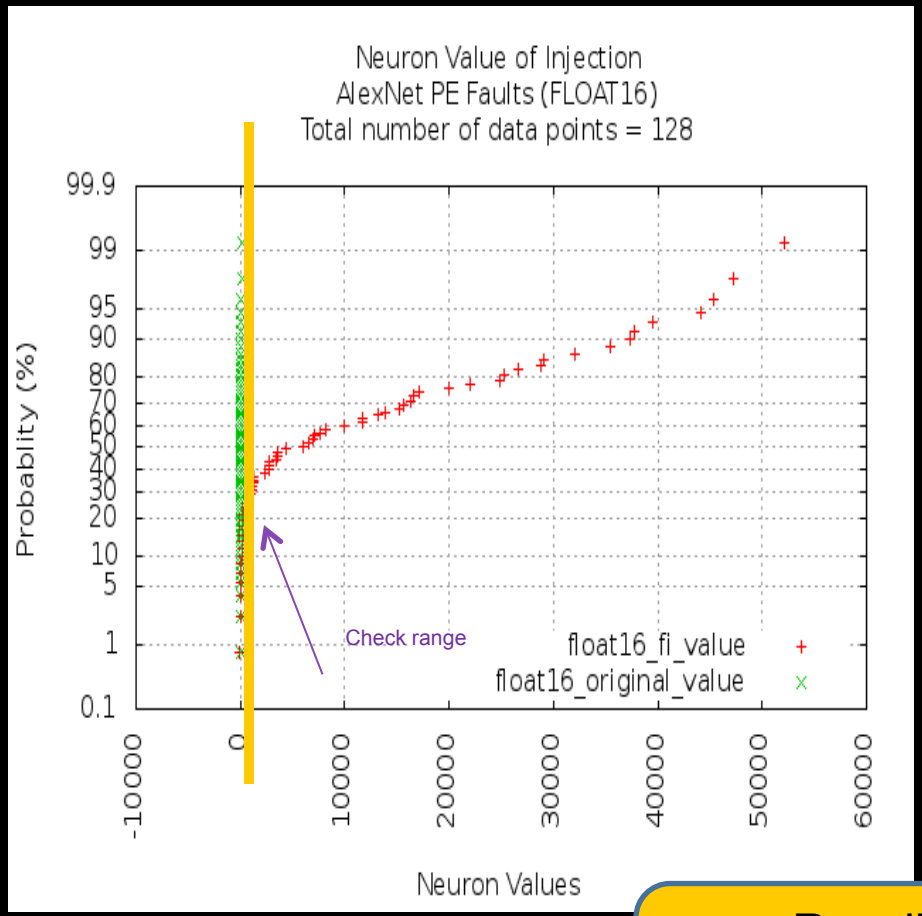
Mitgation: Data Type Choice



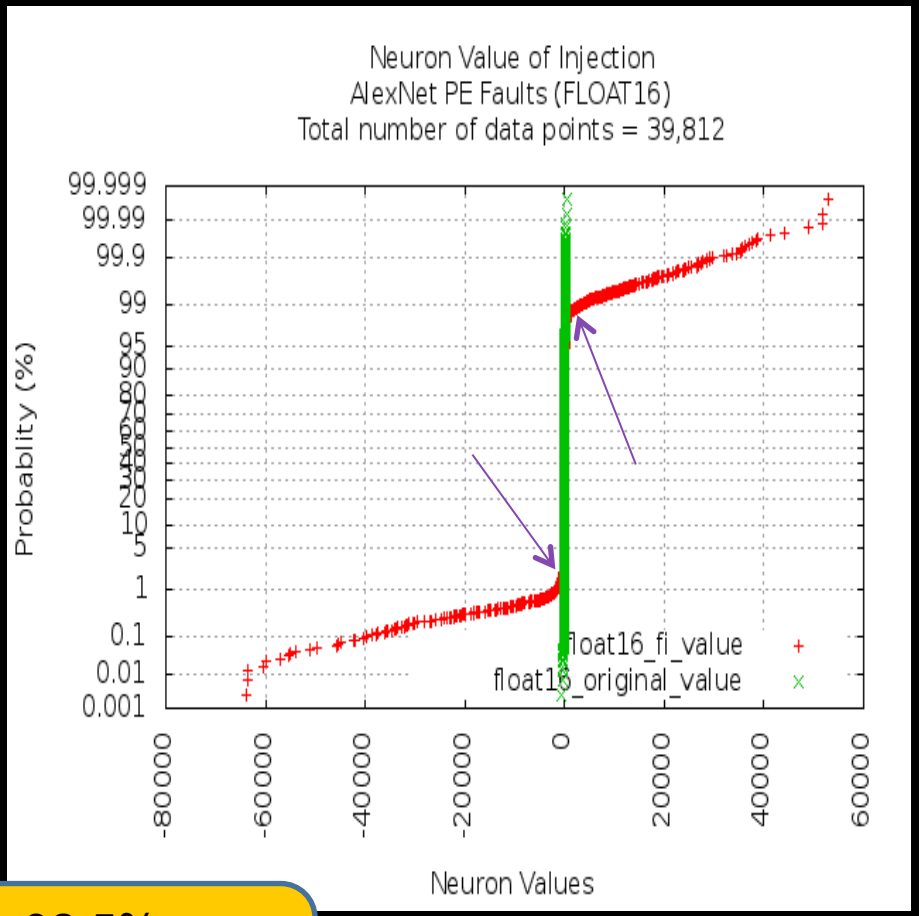
*Scaling factor = 2 by each tech. generation
All raw FIT rates are projected based on the FIT at 28nm [Neale, IEEE TNS]

Mitigation: Symptom-Based Error Detector (Software)

AlexNet, PE Faults, Float16



SDC



Benign

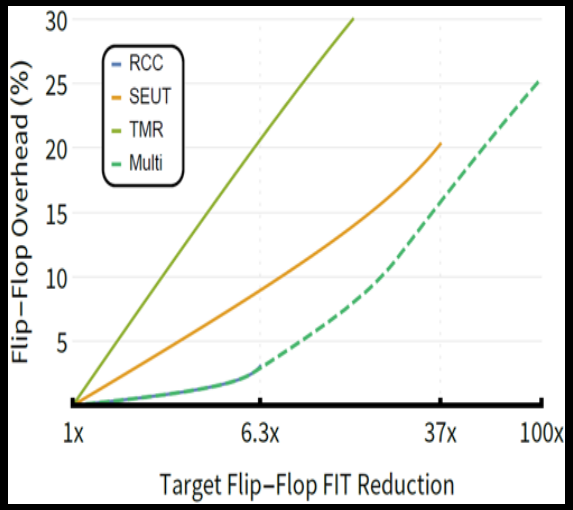
Recall: 92.5%
Precision: 90.21%
Overheads negligible

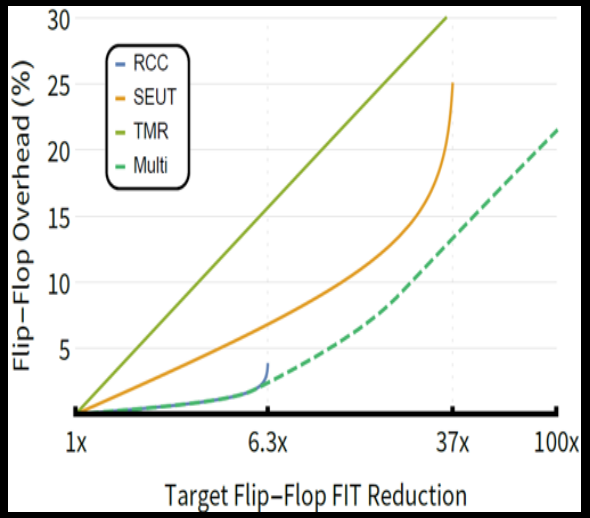
Mitigation: Selective Latch Hardening (Hardware)

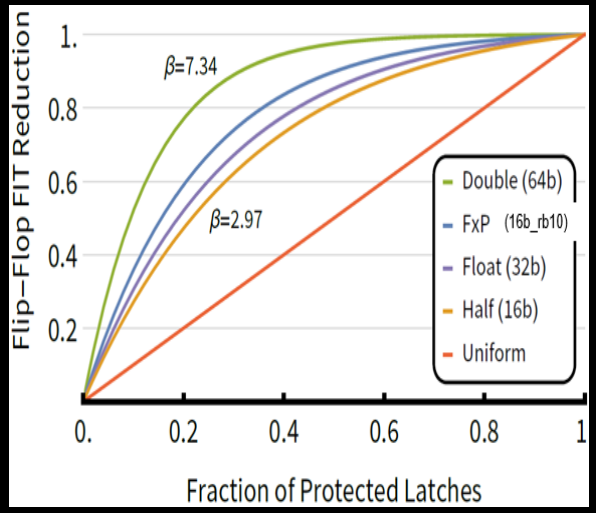
Latch hardening design choices:

Latch Type	Area Overhead	FIT Rate Reduction
Baseline	1x	1x
Strike Suppression (RCC)	1.15x	6.3x
Redundant Node (SEUT)	2x	37x
Triplicated (TMR)	3.5x	1,000,000x

~20% overhead provides 100x reduction in FIT



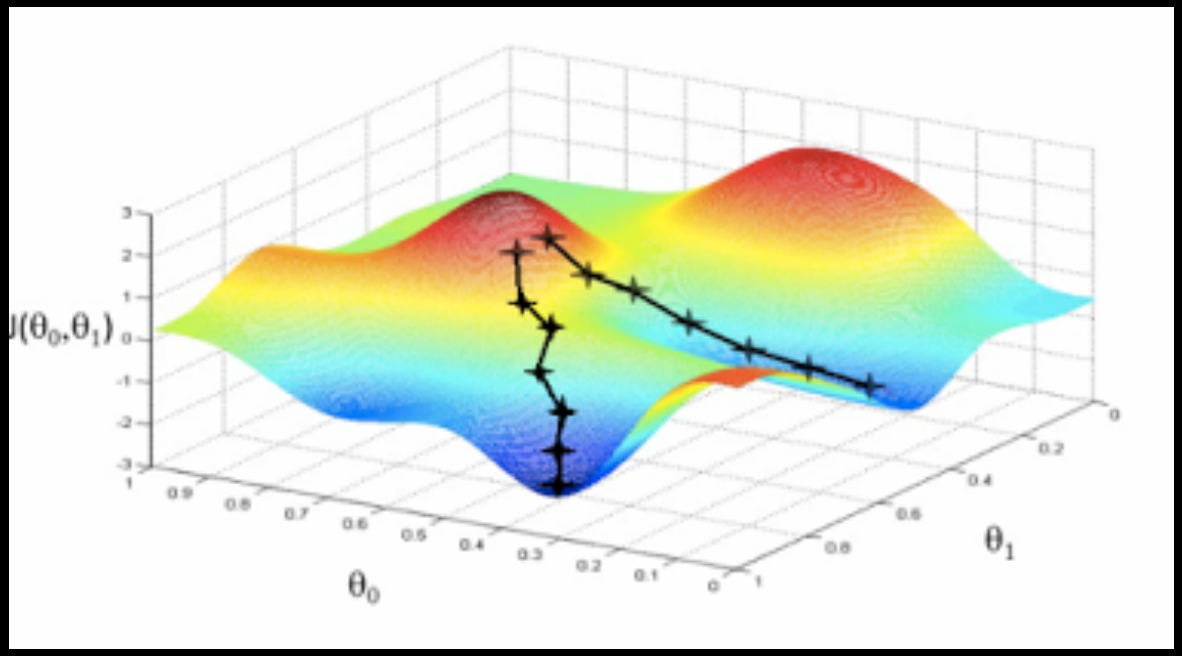




Ongoing Work: Algorithmic Resilience

Deriving ML algorithms resilient to perturbations

- Small changes → Similar outputs



Conclusions

Characterized error propagation in DNN accelerators based on data types, layers, value types & topologies

Mitigation Methods

- Choosing Restrained Data Types
- Symptom-Based Error Detection
- Selective Latch Hardening
- Algorithmic Resilience

Questions ? karthikp@ece.ubc.ca