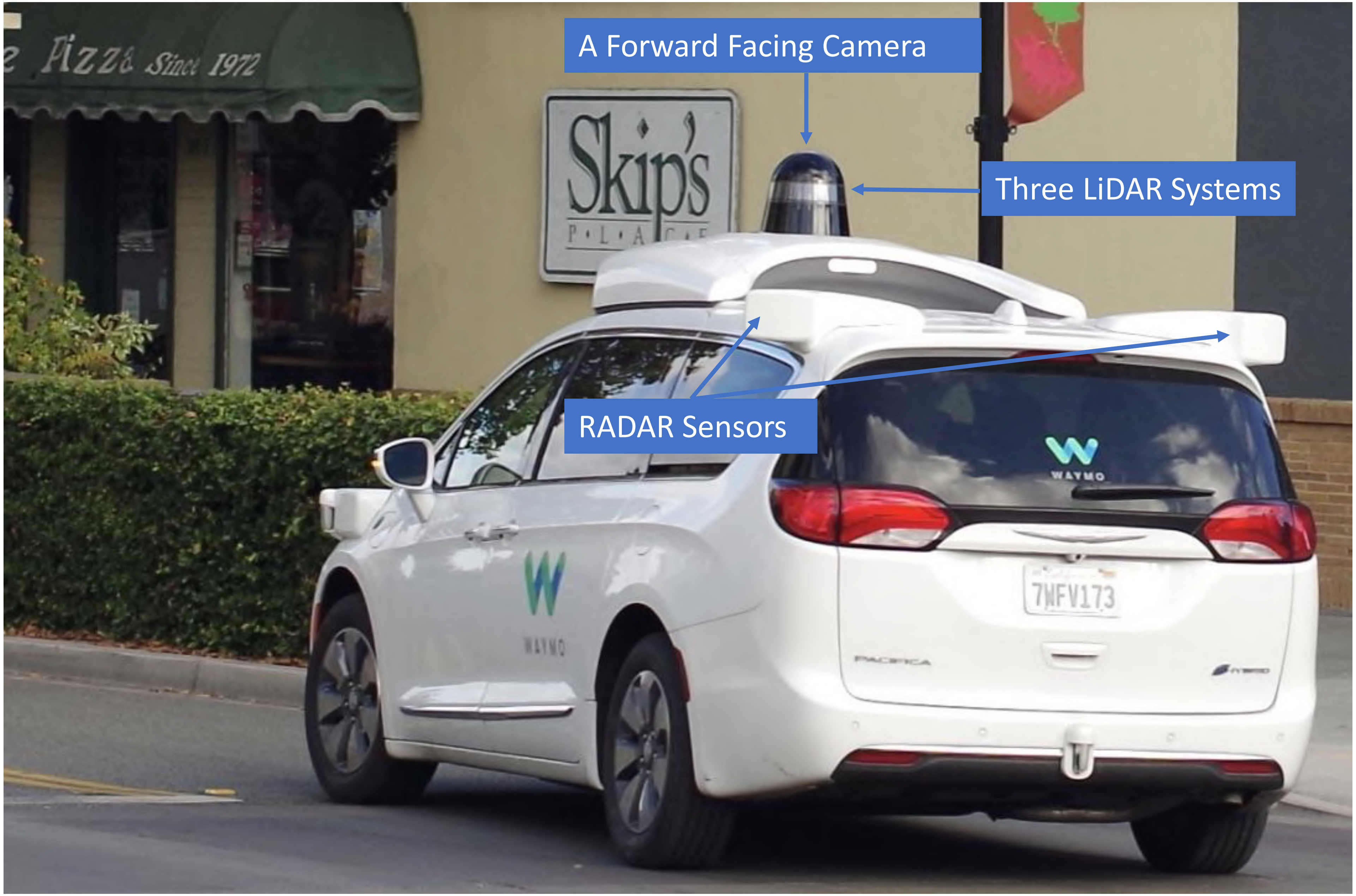
The Road to Resilient Autonomous Cars is Paved with Testability & Redundancy

Nirmal R. Saxena
NVIDIA
Jan 5, 2020

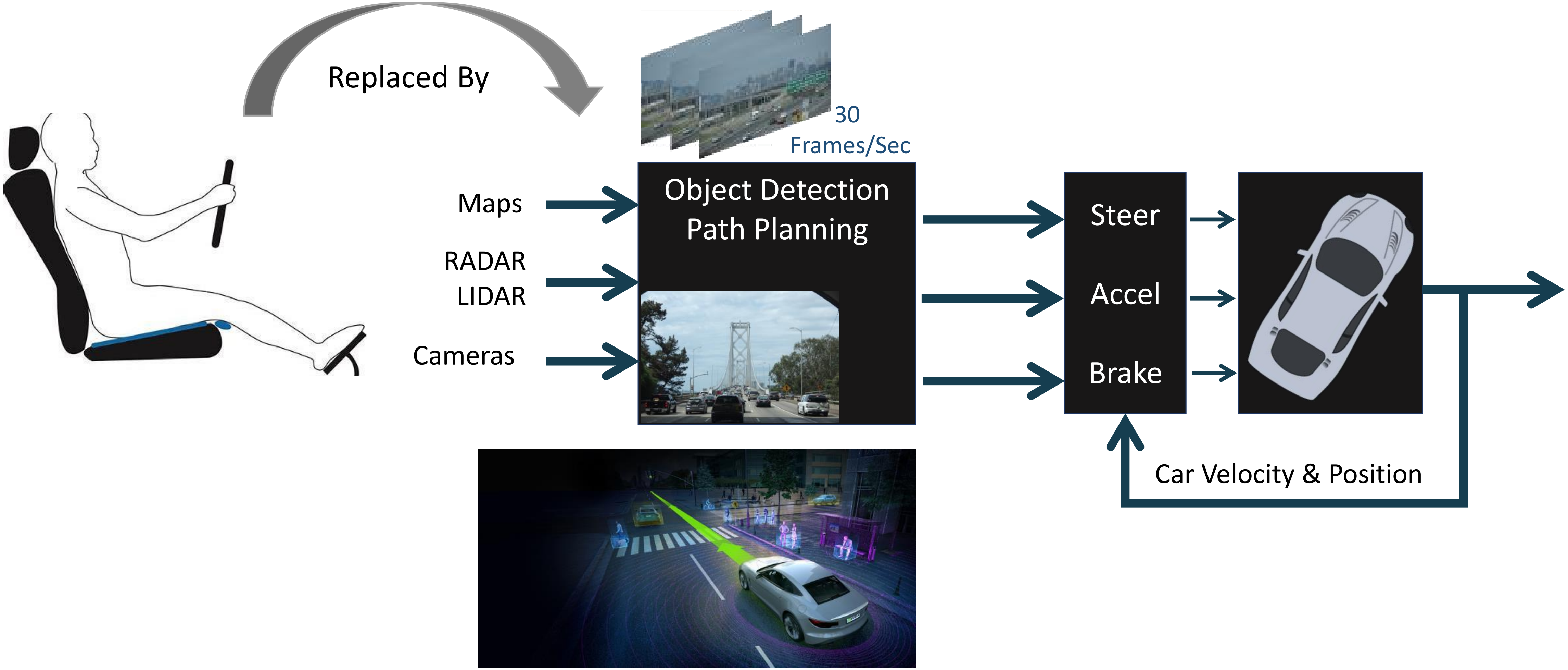
NVIDIA

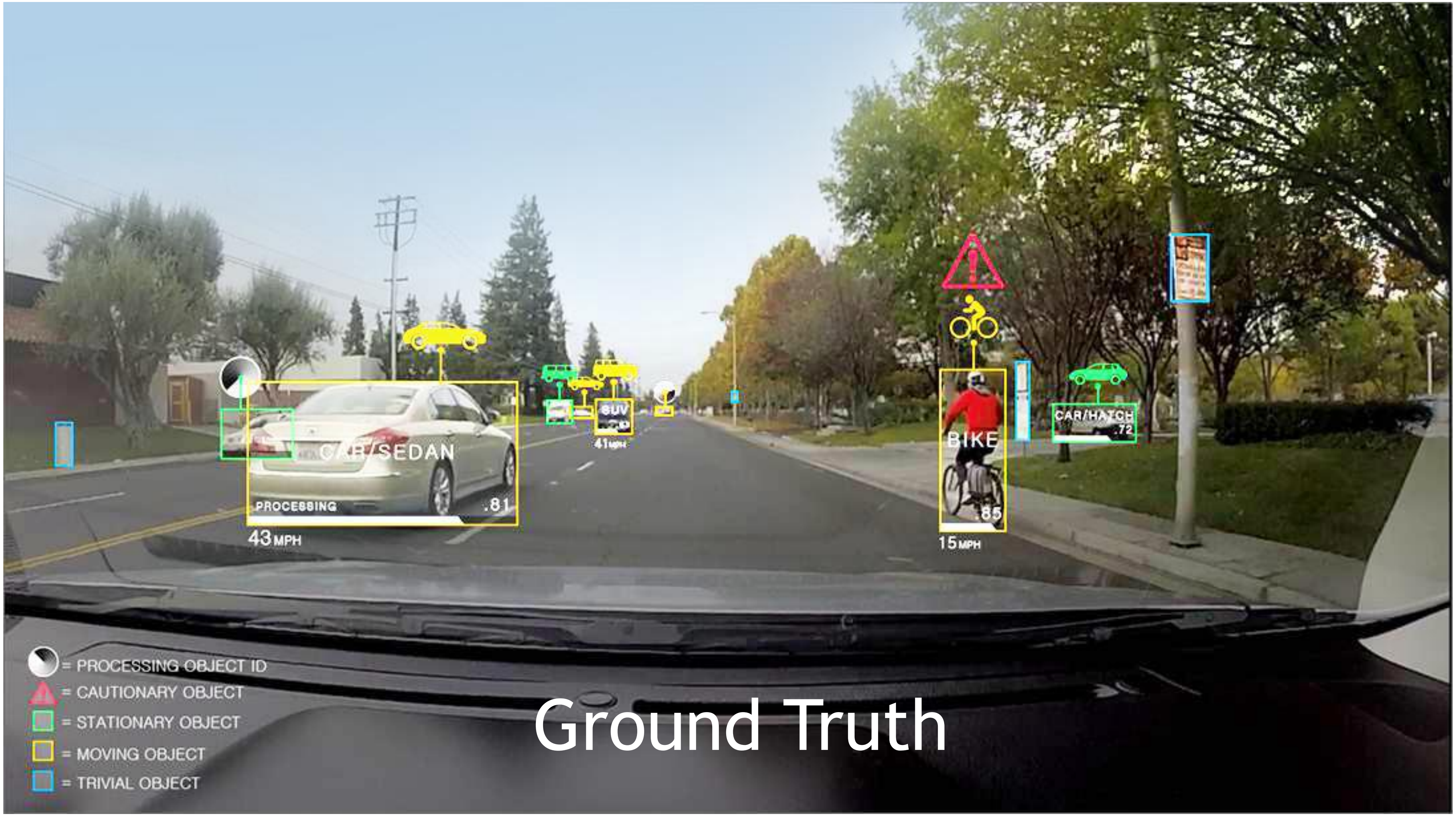
Tutorial Flow

- Auto Safety Standard
  - Driverless Car Model
  - Resiliency & Testability Requirements
- Testability Evaluation, Solutions & Challenges
  - Transient & Permanent Faults
  - Use Case Application Resiliency Characteristics (AVF)
  - Permanent Fault Coverage & Availability Challenges
- Road to Resiliency
  - Reliability Models
  - Latent Fault Coverage
  - Need for Diversity– Systematic Faults


A Forward Facing Camera
Three LiDAR Systems
RADAR Sensors
Pizza Since 1972
Skip's
P L A C E
WAYMO
7MFV173
PACIFICA

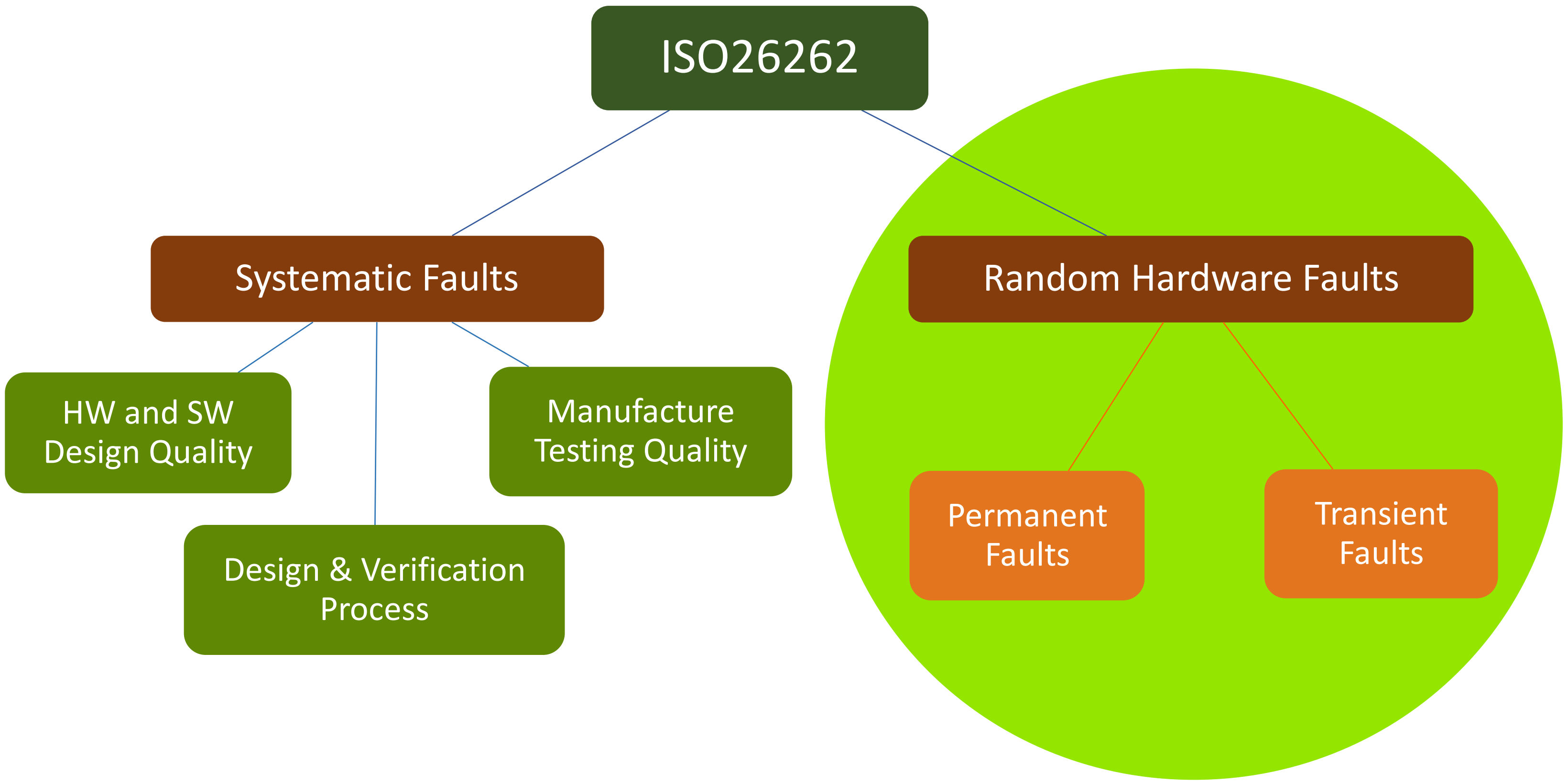
Control System Model– Autonomous Car




CAR/SEDAN
PROCESSING
.81
43 MPH
SUV
41 MPH
BIKE
.85
15 MPH
CAR/HATCH
.72
= PROCESSING OBJECT ID
= CAUTIONARY OBJECT
= STATIONARY OBJECT
= MOVING OBJECT
= TRIVIAL OBJECT
Ground Truth

Object Detection & Path Planning

ISO26262 Auto Safety Specification



Random Hardware Faults Requirement

Hardware Random Fault Metrics	ASIL B	ASIL C	ASIL D
Permanent Fault Coverage (SPFM)	90%	97%	99%
Transient Fault Coverage (SPFM)	90%	97%	99%
Latent Fault Coverage (LFM)	60%	80%	90%
Hardware Failure Probability (PMHF)	100FIT $\leq 10^{-7}/hr$	100FIT $\leq 10^{-7}/hr$	10FIT $\leq 10^{-8}/hr$

FIT = Failures in Time, Time = 10^9 Hours. 1 FIT = 10^{-9} *failures/hour*

ASIL	Automotive Safety Integrity Level
SPFM	Single Point Fault Metric
LFM	Latent Fault Metric
PMHF	Probabilistic Metric for Hardware Failures

FIT Failure Rate Model

λ failure rate: *failures* in an hour

FIT: *failures in time* (time=10^9 hours)

$$\lambda = FIT\ x\ 10^{-9}$$

<u>Arrival of failure events</u>

Follow exponential distribution at **constant rate** λ

<u>Poisson's Model</u>

$$Probability\ of\ n\ failures\ in\ time\ t = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

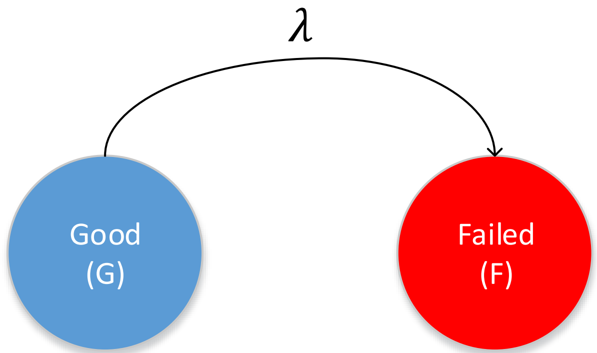
Exponential Distribution

$R(t)$: Probability of no failure up to time t

$$R(t) = e^{-\lambda t} \cong 1 - \lambda t, \lambda t < \epsilon$$

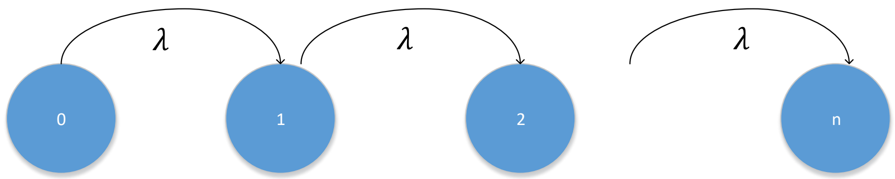
$F(t)$: Probability of a failure in time $\leq t$

$$F(t) = 1 - e^{-\lambda t} \cong \lambda t, \lambda t < \epsilon$$



$$\frac{dP_G(t)}{dt} = -\lambda P_G(t), \mathrm{P_G}(0) = 1$$

$$\int \frac{dP_G(t)}{P_G(t)} = \ln\left(P_G(t)\right) = -\int \lambda dt = -\lambda t$$



$$e^{-\lambda t} \qquad \lambda t e^{-\lambda t} \qquad \frac{(\lambda t)^2}{2!} e^{-\lambda t} \qquad \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

ISO262 Fault Definitions and Raw Failure Rates

Safe Faults:

- Fault whose occurrence will not cause violation of a *safety goal*.
  - Example– Fault in the unused logic of a processor element.
  - This type of fault is characterized by a fault rate λ_S.

Single-Point Faults:

- Faults not detected because of the absence of a checking mechanism
  - Have the potential to cause a safety violation.
- This type of fault is characterized by a fault rate λ_{SPF}.

ISO262 Fault Definitions and Raw Failure Rates

Residual Faults:

- Faults not detected by Checking Mechanism (Safety Mechanism).
  - Have the potential to cause a safety violation.
  - This type of fault is characterized by a fault rate λ_{RF}.

Multi-Point Faults:

- Multiple Independent Faults that in Combination
  - Have the potential to cause a safety violation.
  - Subset of these faults are either detected (checker) or perceived (user)
- Characterized by a fault rates $\lambda_{MPF,det}$ and $\lambda_{MPF,per}$.

ISO262 Fault Definitions and Raw Failure Rates

Latent Fault:

- A Multi-Point Fault Neither Perceived Nor Detected
  - By itself has no potential to cause a safety violation
  - In combination with another fault has the potential to cause safety violation.
- Characterized by a fault rate $\lambda_{MPF,lat}$

$$\lambda = \lambda_S + \underbrace{\lambda_{SPF} + \lambda_{RF} + \lambda_{MPF,lat}}_{\text{SDC– Silent Data Corruption}} + \underbrace{\lambda_{MPF,det} + \lambda_{MPF,per}}_{\text{DUE– Detected Uncorrected Error}}$$

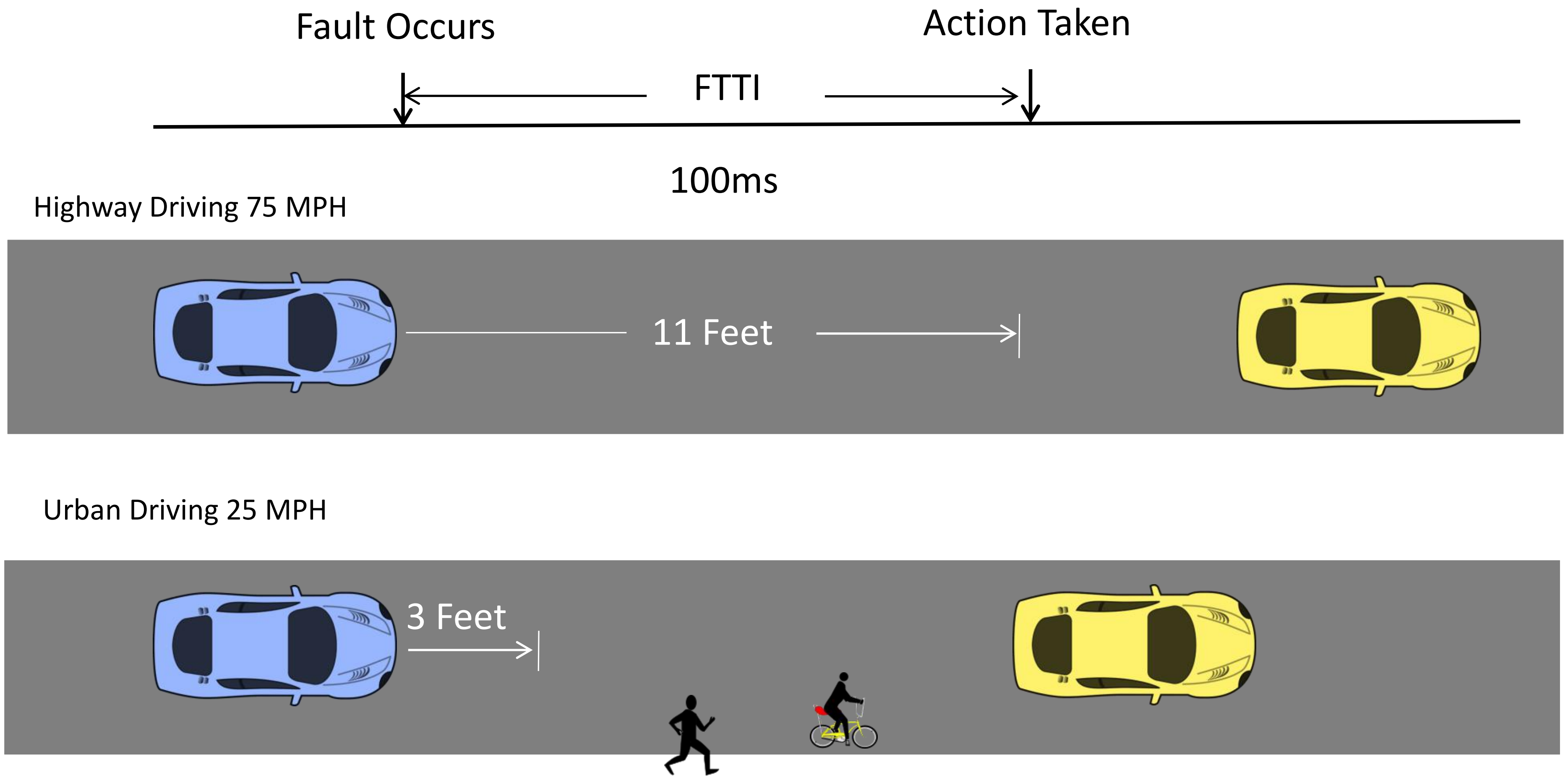
SPFM & LFM

$$SPFM = 1 - \frac{\sum(\lambda_{SPF} + \lambda_{RF})}{\sum \lambda}$$

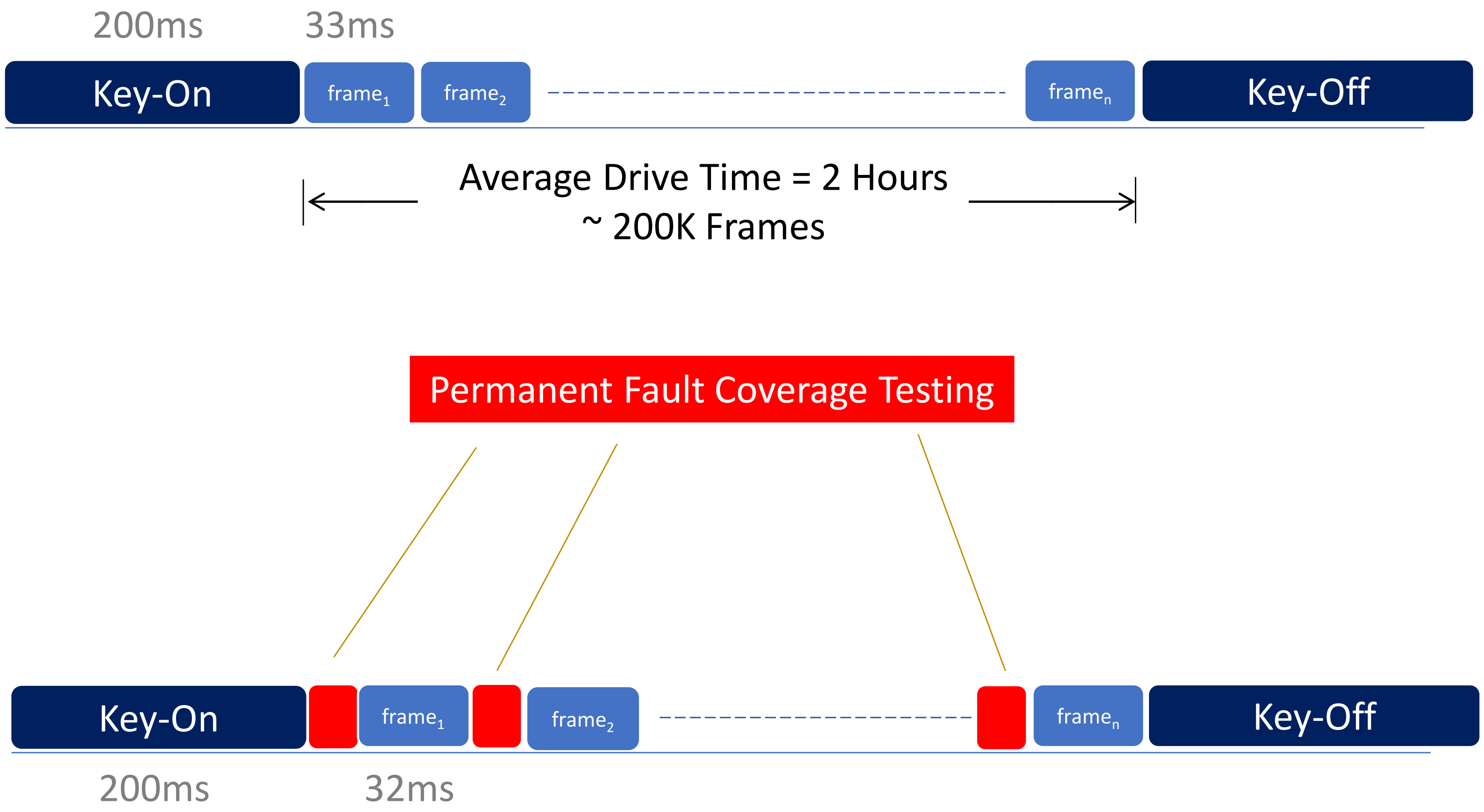
$$LFM = 1 - \frac{\sum \lambda_{MPF,lat}}{\sum(\lambda - \lambda_{SPF} - \lambda_{RF})}$$

Fault Tolerant Time Interval (FTTI)

ISO26262 does not Quantify FTTI

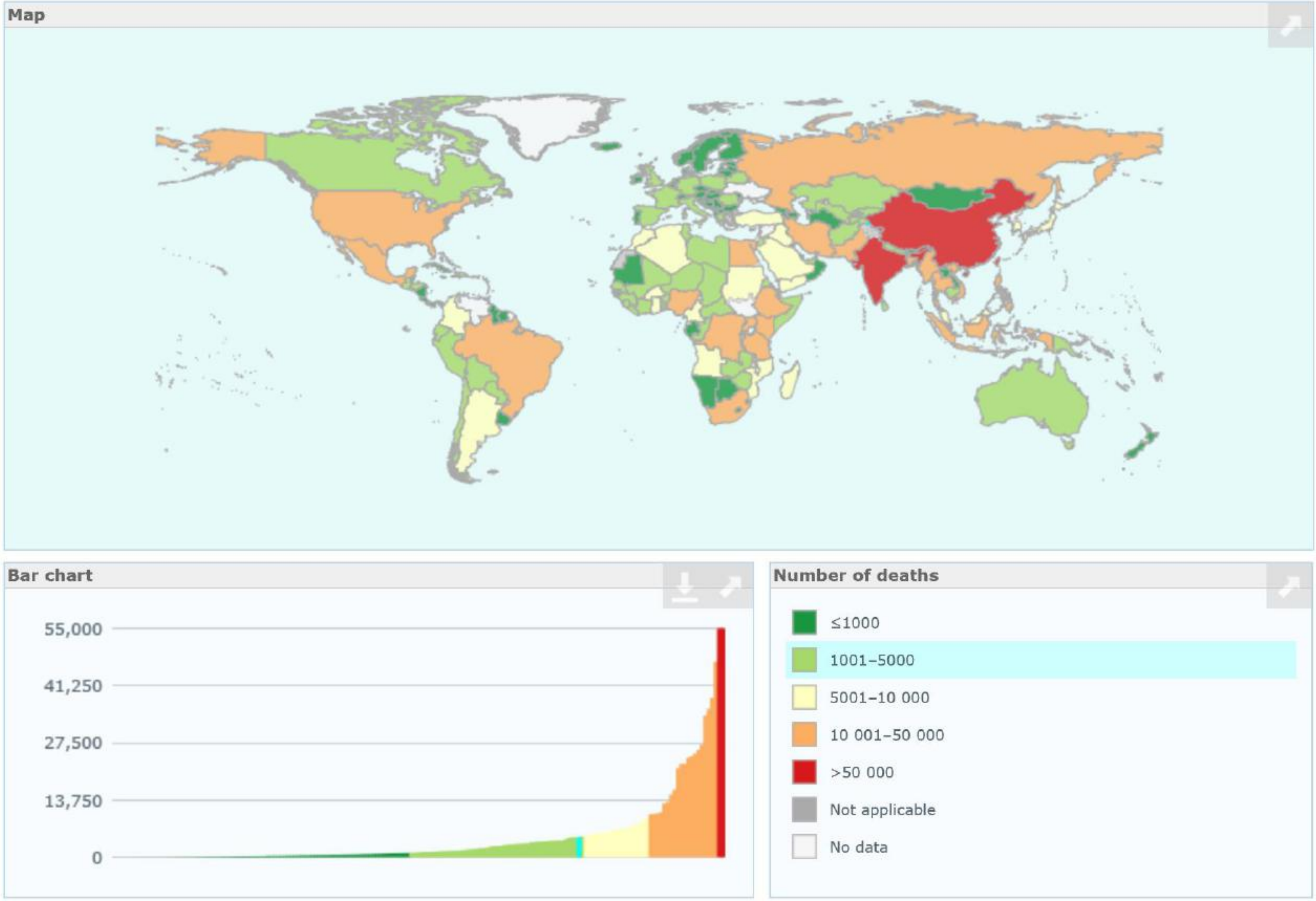


Key-On, Drive-Time, Key-Off



1.25 Million Road Traffic Deaths Globally in 2013

https://en.wikipedia.org/wiki/List_of_countries_by_traffic-related_death_rate



Accident Statistics– US

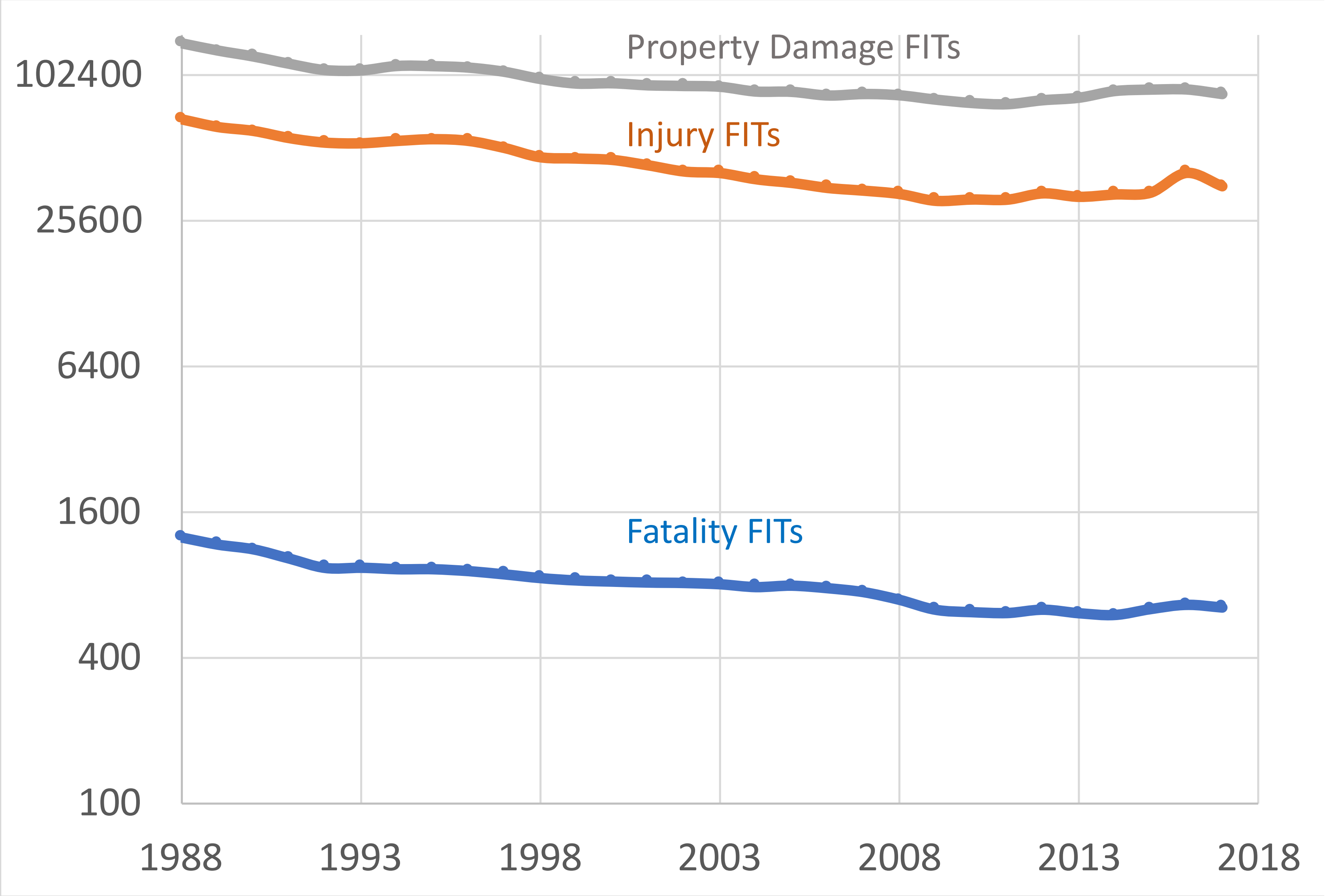
Reference: National Highway Traffic Safety Administration (NHTSA): www.nhtsa.gov

Description	2013 Statistics	2015 Statistics
Fatal Crashes	**30,057**	**35,092**
Driver Related Fatal Crashes	**10,076**	**10,265**
Non-Fatal Crashes	**5,657,000**	**6,263,834**
Number of Registered Vehicles	**269,294,000**	**281,312,446**
Licensed Drivers	**212,160,000**	**218,084,465**
Vehicle Miles Travelled	**2,988,000,000,000**	**3,095,373,000,000**
Fatal Crash Rate in FITs	**250 – 500**	**283 - 566**
Non-Fatal Crash Rate in FITs	**46K – 92K**	**51K – 102K**
ASIL D 10 FITs is ~ 50x Improvement over Fatal Crash Rate & 4 Orders of Improvement in Non-Fatal CR FITs		

Economic Cost of Traffic Crashes (2010) $242 Billion

Google Non-Fatal Crash FIT Rate = 150K

FIT Range Distribution in the US– 1988 through 2017



2013 Accident Statistics– India

Ministry of Transportation (India)
https://en.wikipedia.org/wiki/List_of_countries_by_traffic-related_death_rate

Description	Statistics
People Killed in Fatal Crashes	238,562
Driver Related Fatal Crashes	
Total Number of Accidents	497,686
Number of Registered Vehicles	160,000,000
Licensed Drivers	
Vehicle Miles Travelled	
Number of People Killed per 100,000 Vehicles (India)	130
Number of People Killed per 100,000 Vehicles (US)	12
Fatality Rate Compared to US is 10x	

Biggest Impact in Developing Countries

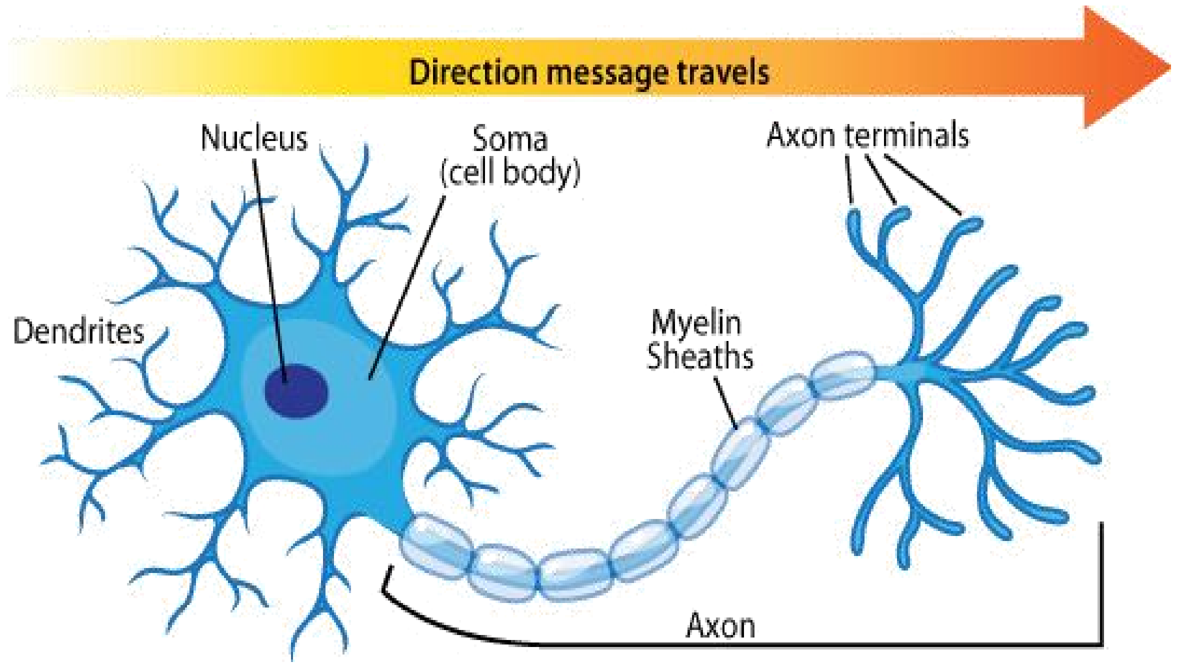
Deep Learning (DL) Accuracy vs. Resilience?

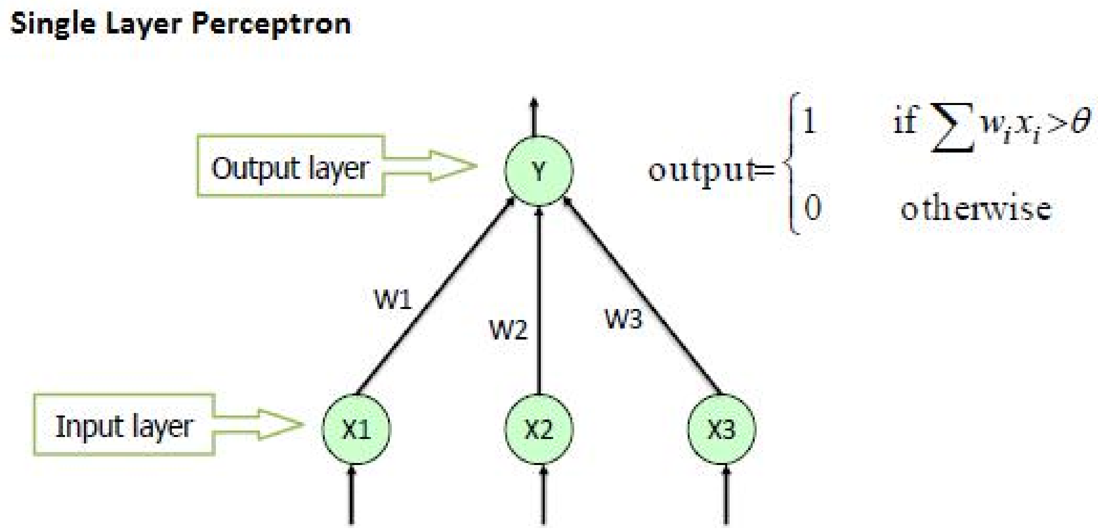
Why Require $\leq$ 1 Failure in 10^8 Hours (= 10^{13} $Frames$)

when

99% Object Detection Accuracy is Equivalent to 1 Missed Frame in 10^2 Frames?

Perceptron– A Compute Model of Neuron

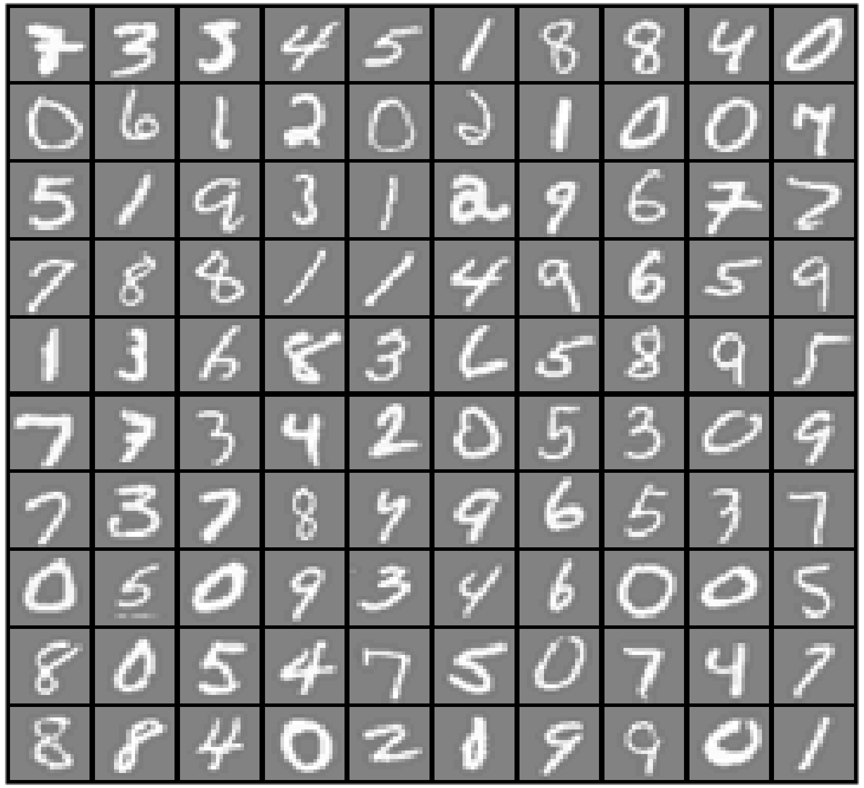




$Input\ Vector\ (X_1, X_2, X_3)$
$Weight\ Vector\ (W_1, W_2, W_3)$
$Dot\ Product = W_1X_1 + W_2X_2 + W_3X_3$

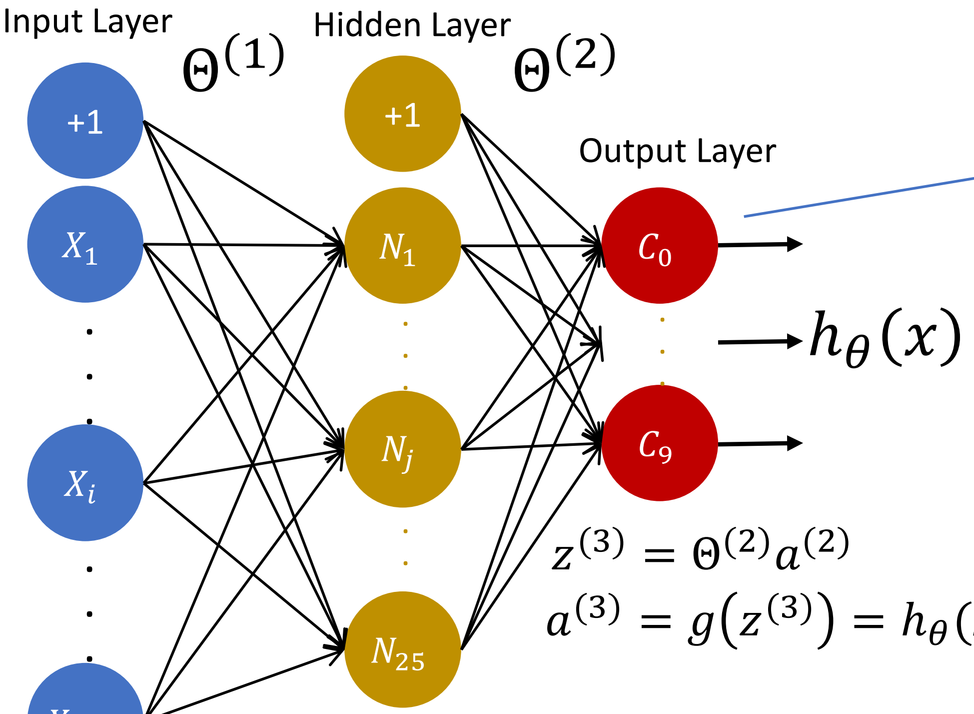
Handwritten Digit Recognition Dataset

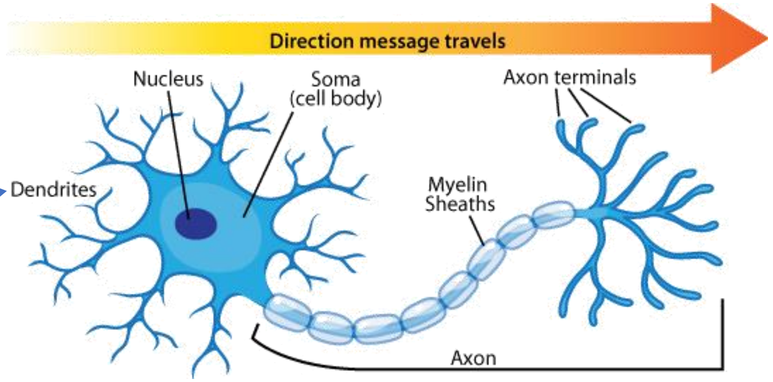
- 5000 Training Examples
- Each Digit 20 x 20 Pixels
  - Flattened to 400 Elements
- Each Pixel Greyscale Shading
  - Floating Point Number
- Supervised Learning



Credits: Coursera Machine Learning Class, Prof. Andrew Ng, Stanford University

Handwritten Digit Recognition
Neural Network



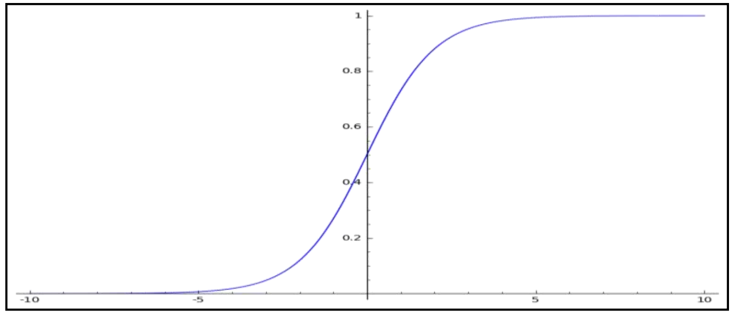


$$z^{(3)} = \Theta^{(2)} a^{(2)}$$
$$a^{(3)} = g\left(z^{(3)}\right) = h_\theta(x)$$

$$a^{(1)} = x$$

$$z^{(2)} = \Theta^{(1)} a^{(1)}$$
$$a^{(2)} = g(z^{(2)})$$

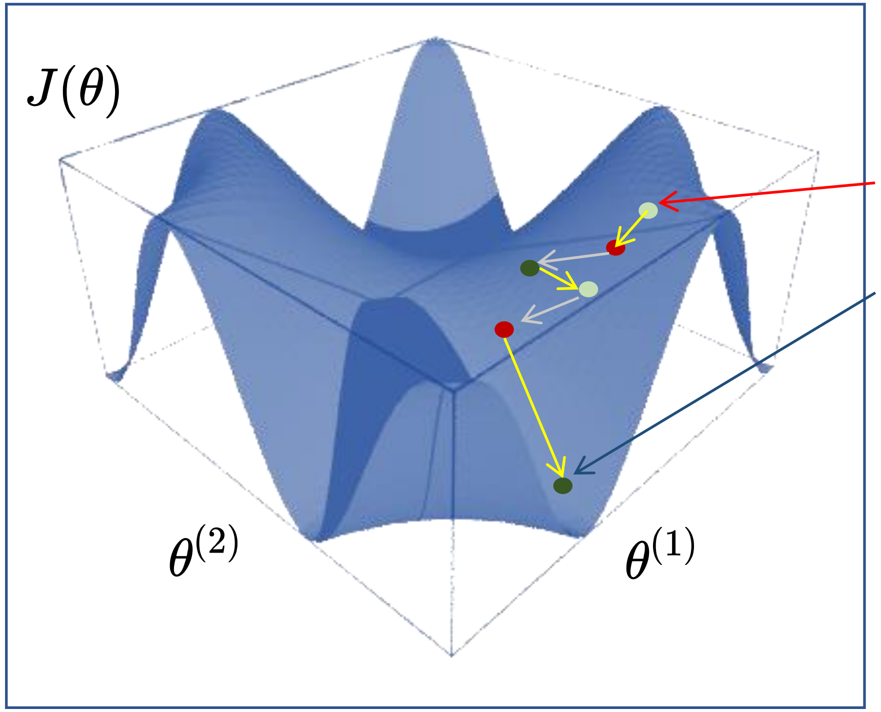
$$sigmoid(z) = g(z) = \frac{1}{1 + e^{-z}}$$



Gradient Descent Algorithm

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m=5000}[-y^{(i)}log(h_\theta(x^{(i)})) - (1-y^{(i)})log(1-h_\theta(x^{(i)}))]$$ Cost Function

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

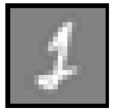


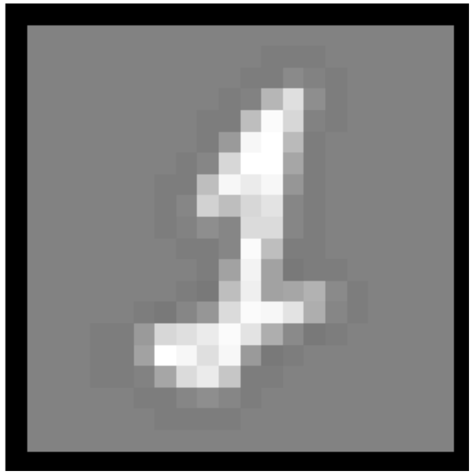
$h_\theta(x)\ far\ away\ from\ y$

$h_\theta(x)\ close\ to\ y$

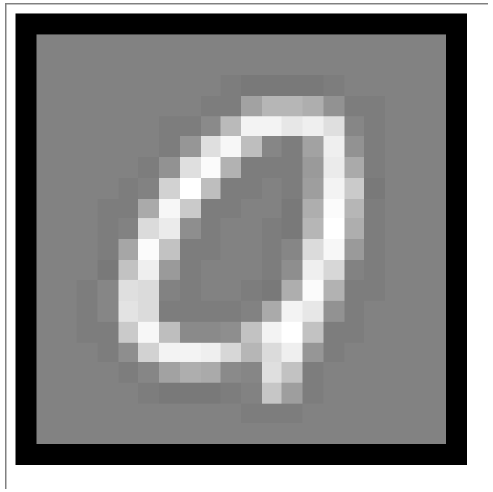
50 Iterations, 20mins > 95% Accuracy
400 Iterations, 3hrs > 99% Accuracy

Test Examples– Resilient Learning



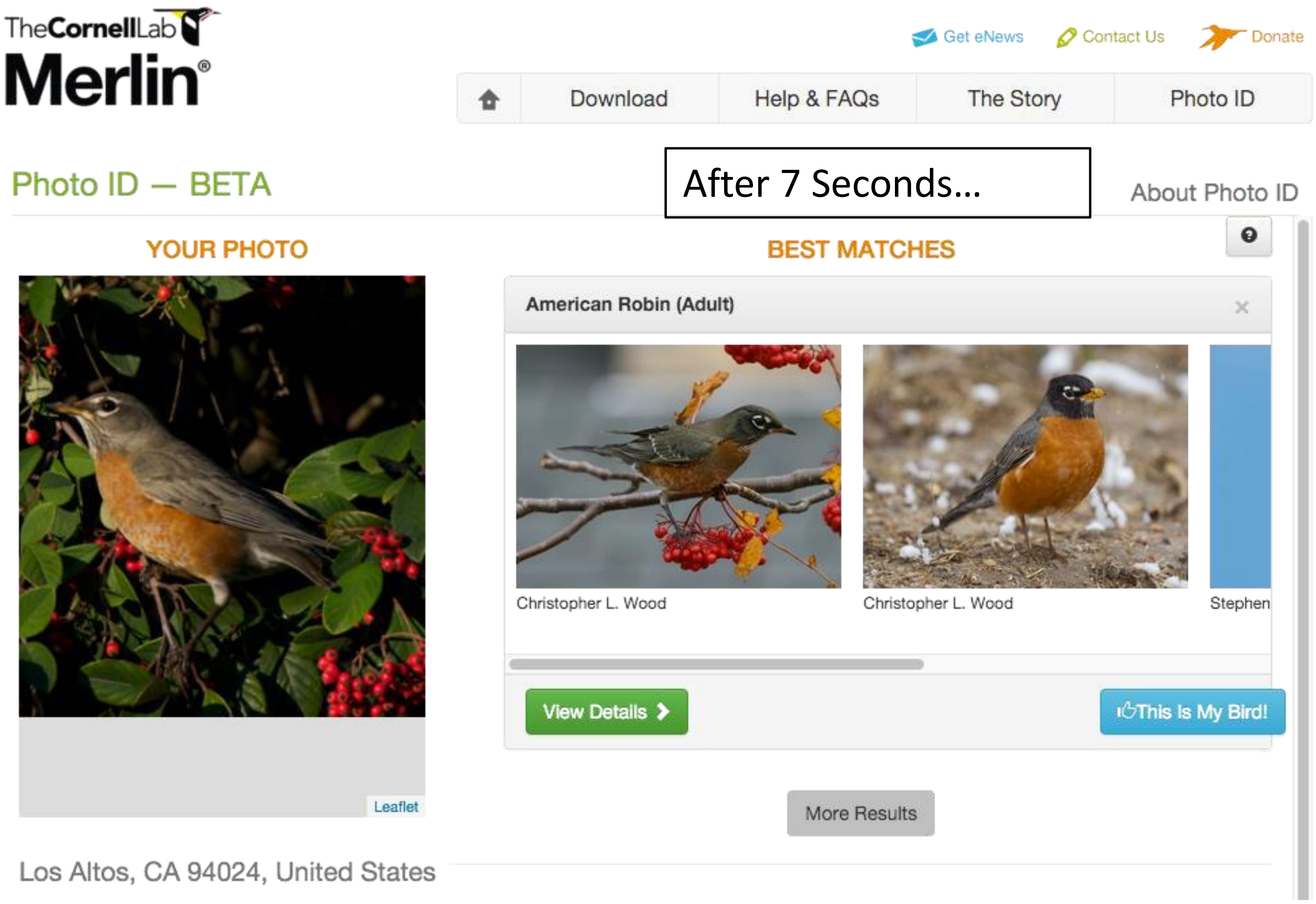


Labeled as 2 but Detected as 1

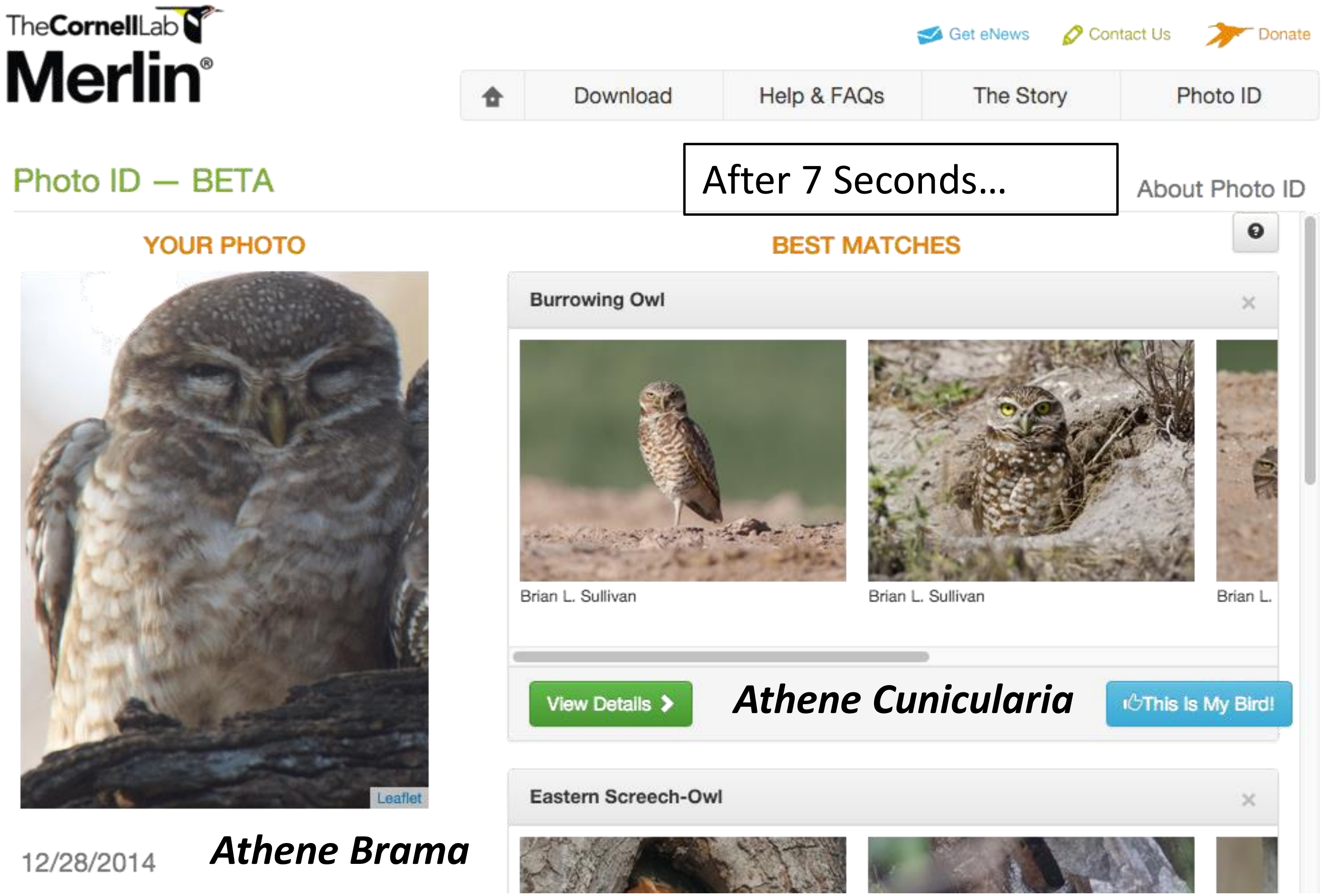


Labeled as 9 but Detected as 0

Deep Learning– North American Bird-ID



Spotted-Owlet in Rajasthan– North India



What Processing Power Per Frame is Needed?

Discounting Network Time

- Image Classification Takes 6 Secs

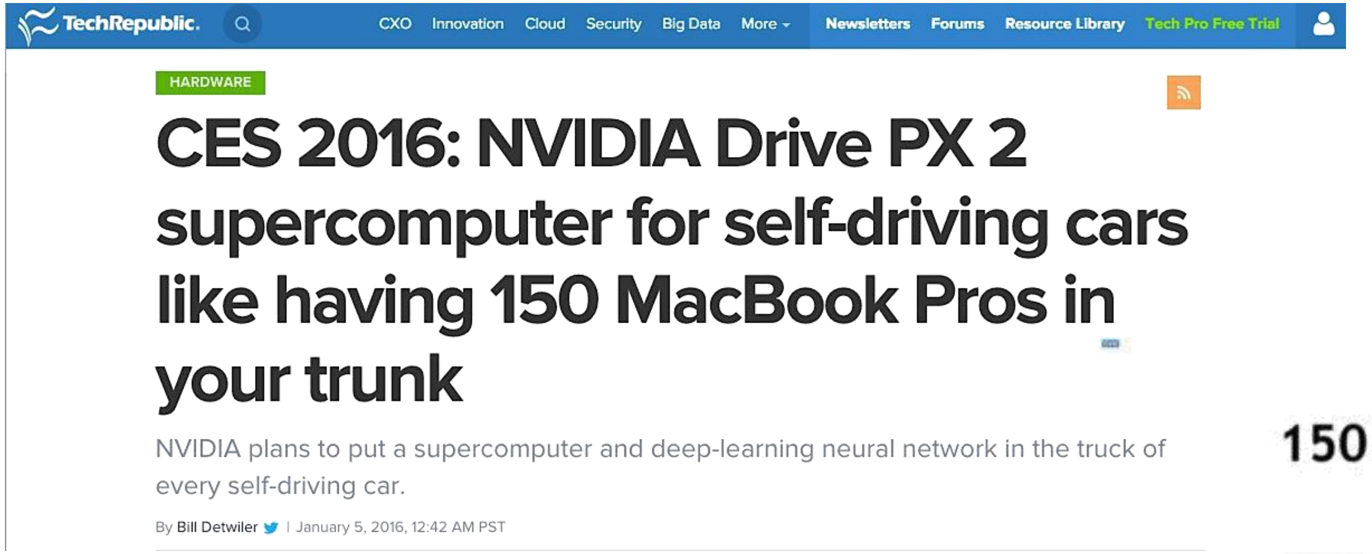
Merlin Bird-ID Hosted on AWS

- Possibly uses Single Xeon Server

To Classify Image in 33ms

- Need 6000ms/33ms = 180 Xeons…

Supercomputer in a Car



TechRepublic. CXO Innovation Cloud Security Big Data More Newsletters Forums Resource Library Tech Pro Free Trial

HARDWARE

CES 2016: NVIDIA Drive PX 2 supercomputer for self-driving cars like having 150 MacBook Pros in your trunk

NVIDIA plans to put a supercomputer and deep-learning neural network in the truck of every self-driving car.

By Bill Detwiler | January 5, 2016, 12:42 AM PST

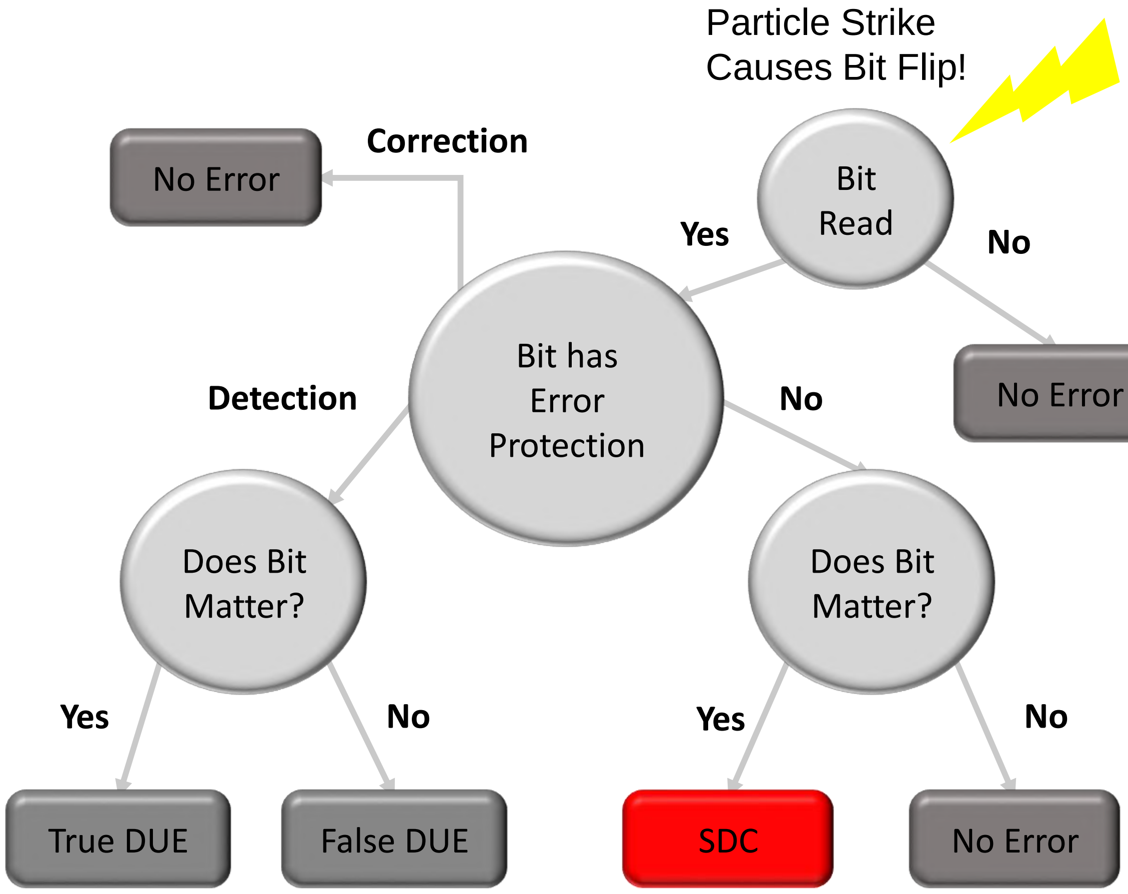
NVIDIA Drive-PX Pegasus GTC-Europe-2017

https://www.anandtech.com/show/11913/nvidia-announces-drive-px-pegasus-at-gtc-europe-2017-feat-nextgen-gpus

Architectural Vulnerability Factor (AVF)

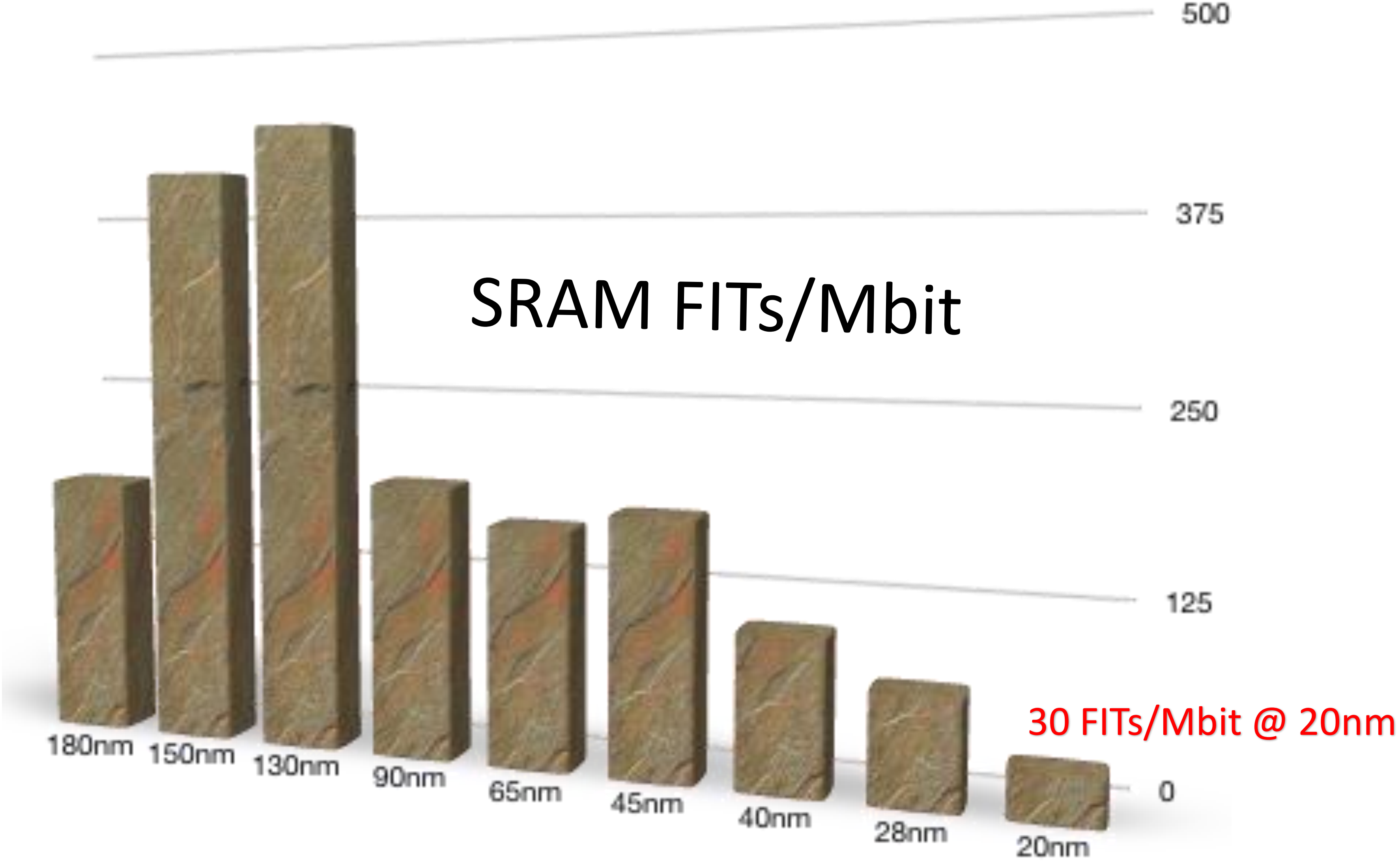
SDC AVF

- Bit Error Results in Corrupted Output
- DUE AVF
- Bit Error Detected and Signaled
- **Low AVF → <u>Architectural Fault Avoidance</u>**
- AVF Function Of
- Design Structure
- Application's Static & Dynamic Behavior

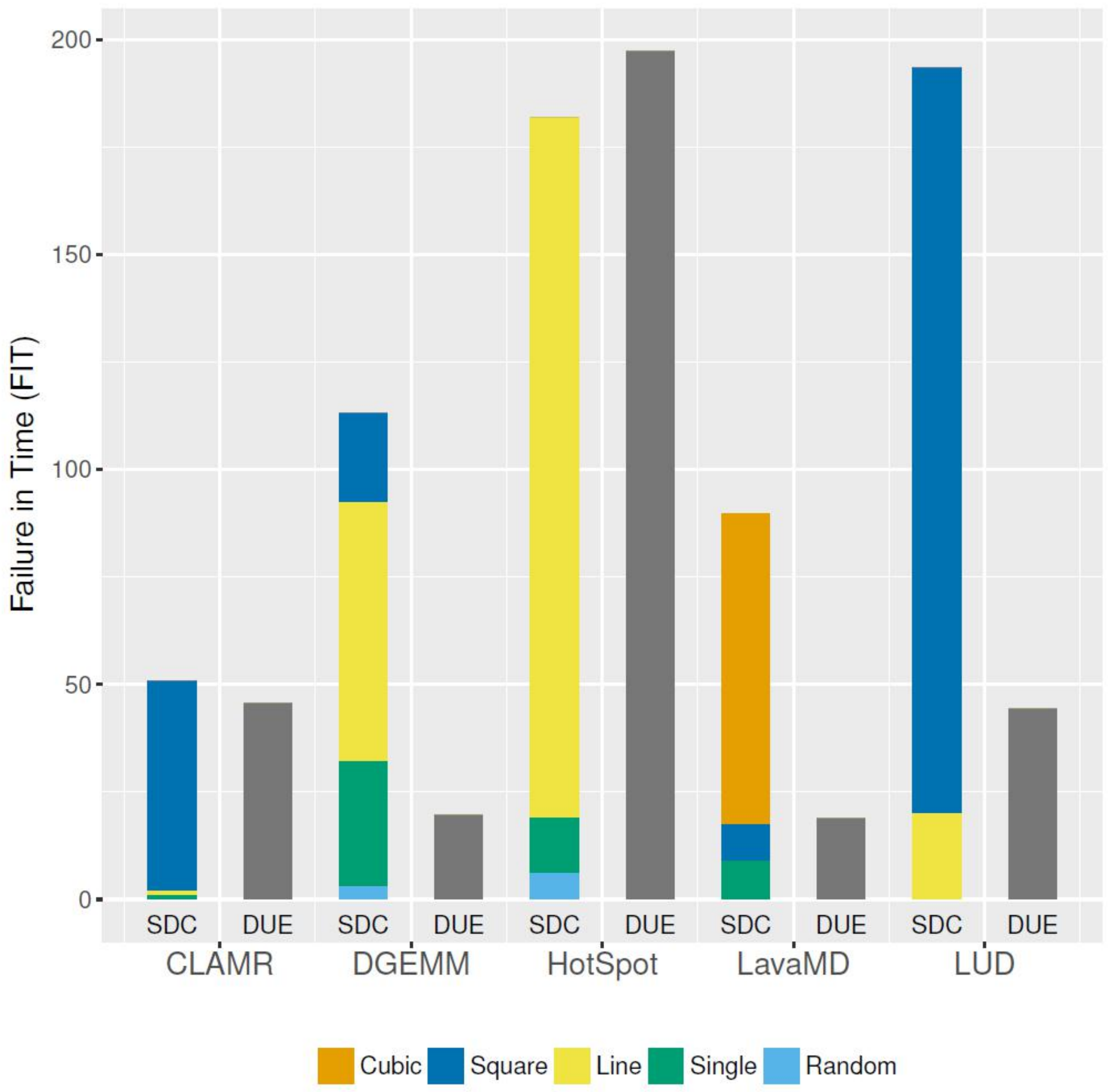


$$\text{PMHF} = (1 - \text{SPFM})\,\lambda$$

$$\boldsymbol{PMHF = AVF_{SDC}\lambda = SDC\ FITS}$$



Source: Xilinx Device Reliability Report (11/17)– UG116

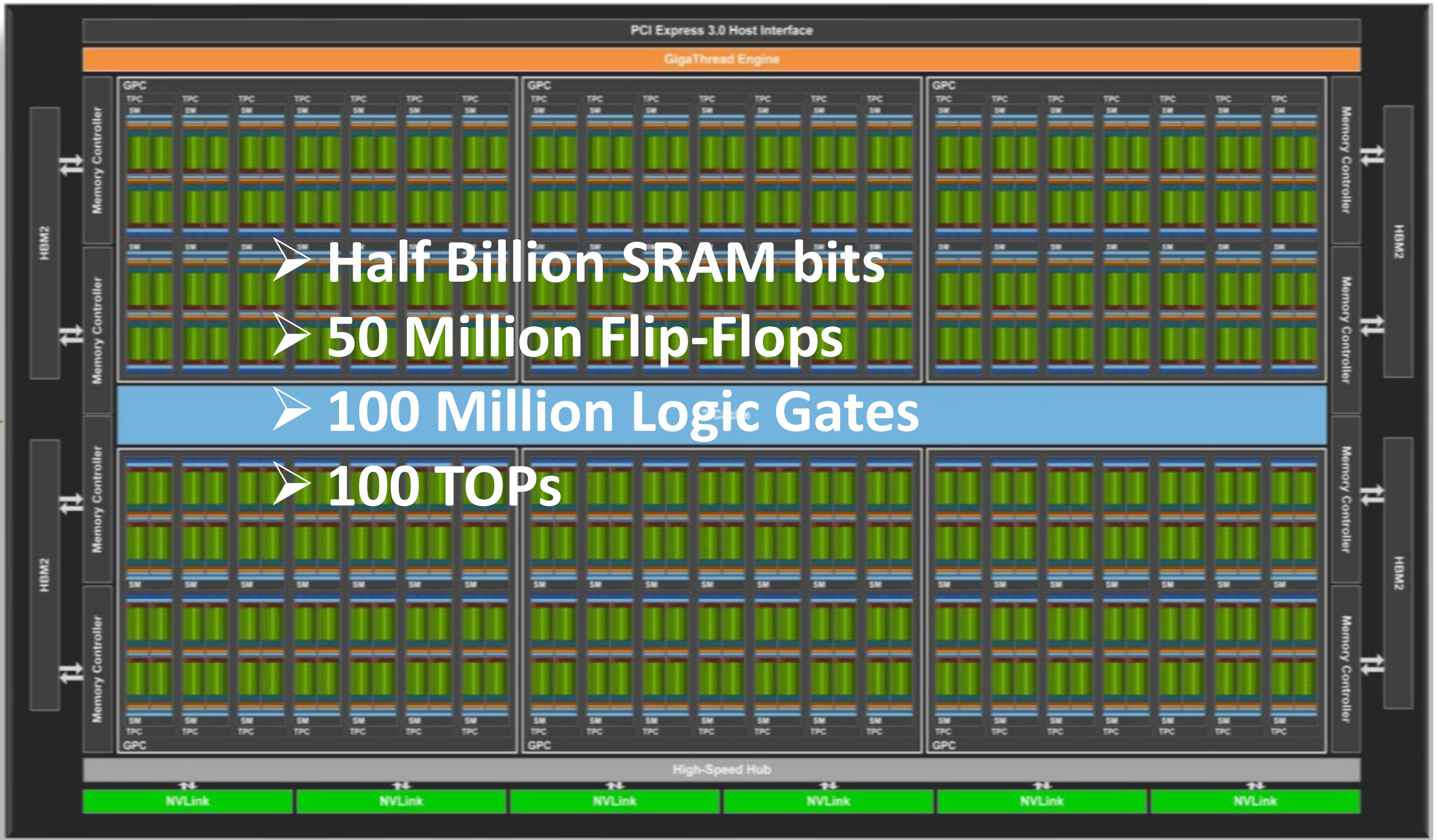


Knights Corner
Xeon-Phi Measured
Max SDC FIT = 193

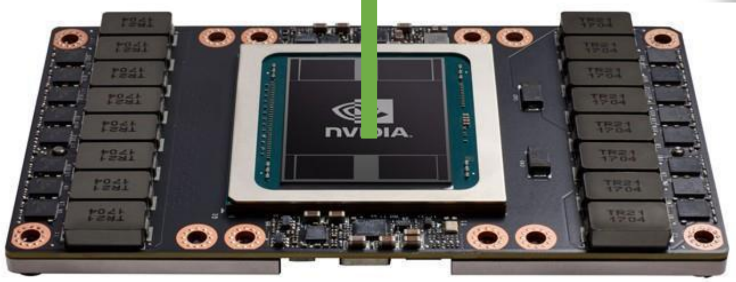
(Assuming 5000 Raw FITs)
Derived SPFM is ~ 96%

SC17, November 12-17, 2017, Denver, CO, USA D. Oliveira, L. Pilla, N. DeBardeleben, S. Blanchard, H. Quinn et al.

A Hypothetical SPFM & PMHF Projection



- Half Billion SRAM bits
- 50 Million Flip-Flops
- 100 Million Logic Gates
- 100 TOPs

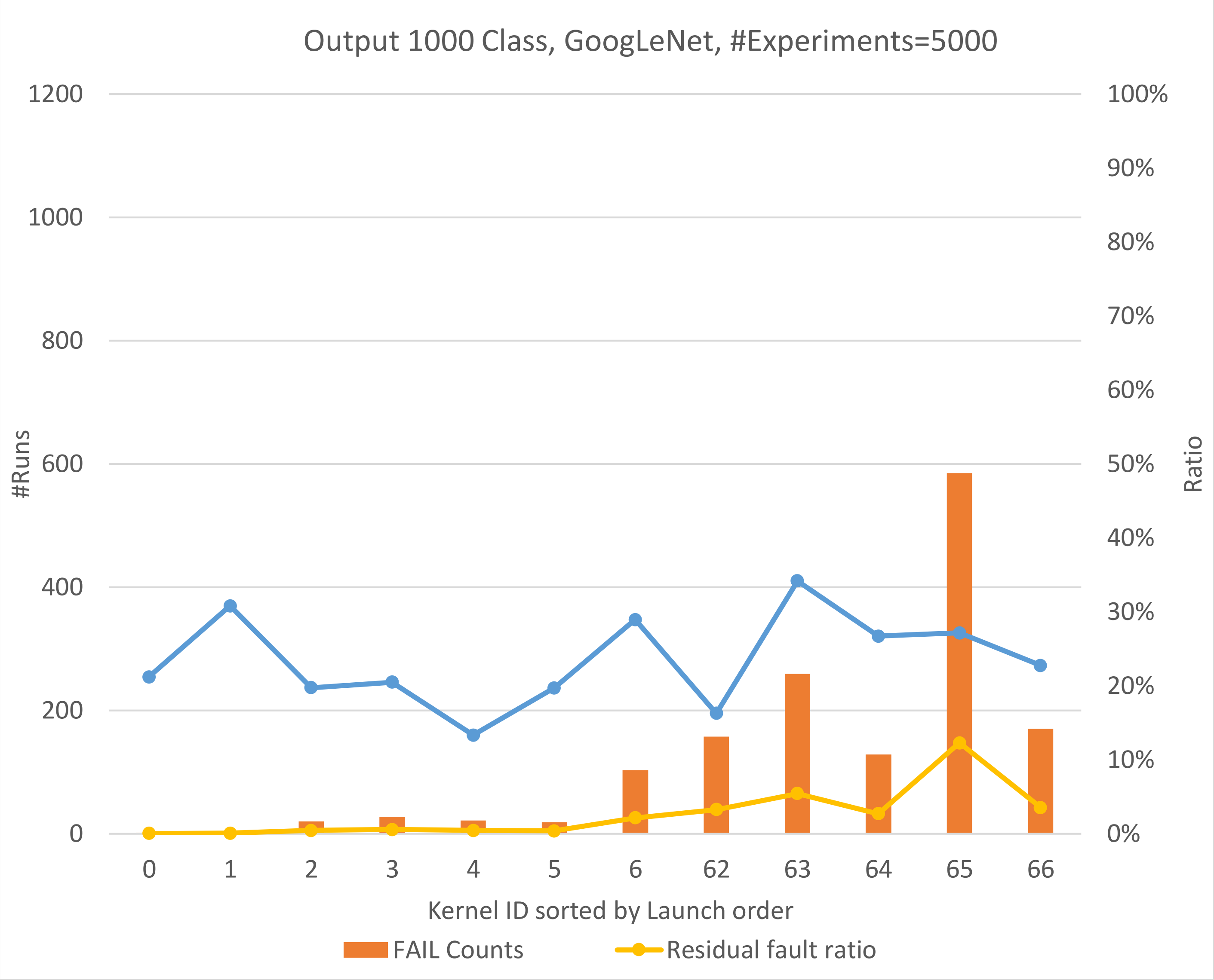


> 50 FITs @ 99% SPFM

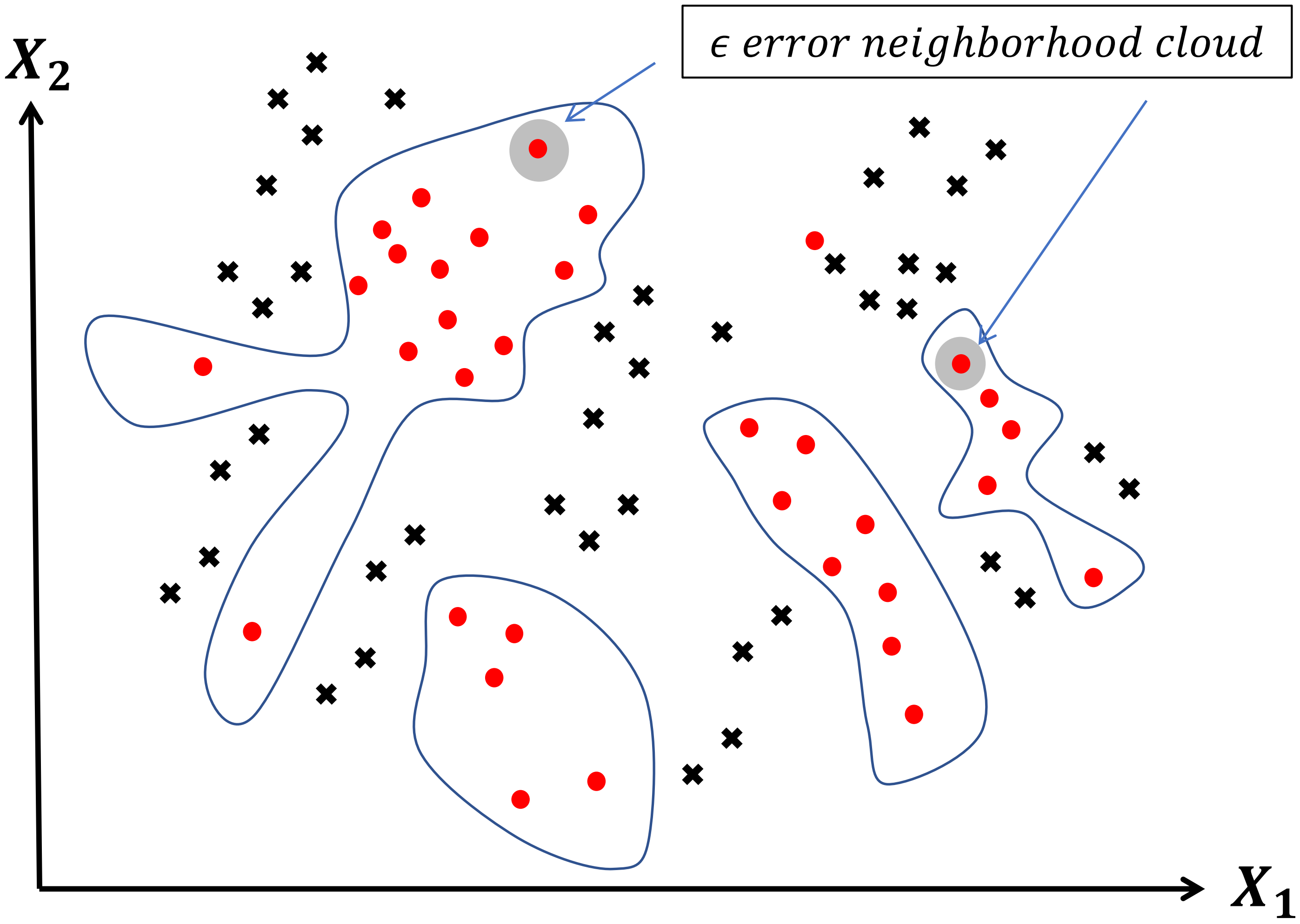
Intrinsic Application Resilience

Less Than 1% Average SDC AVF in DL Classification

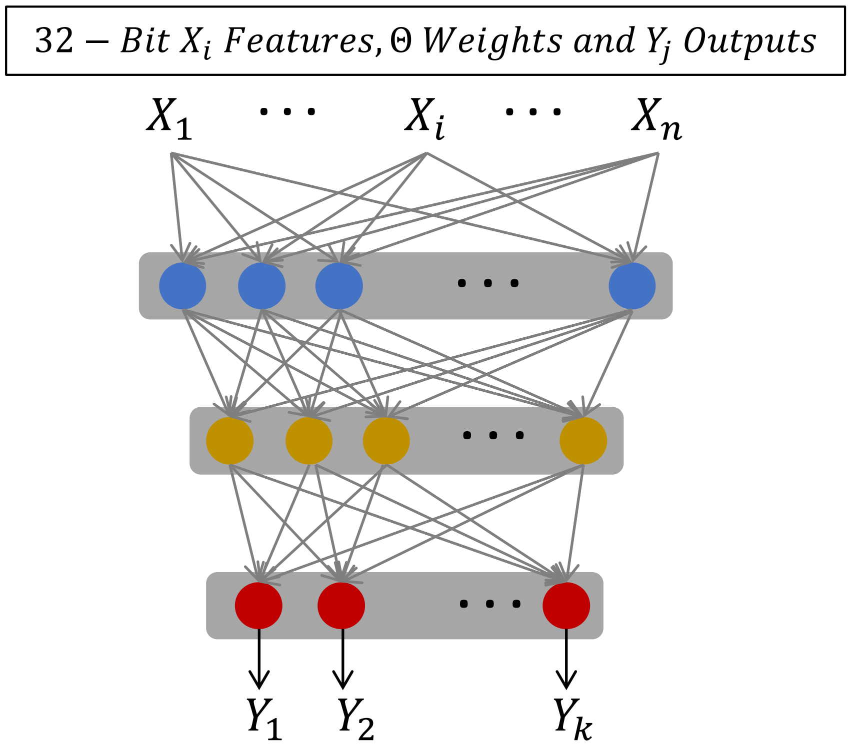
- GTC-2017 Conference
  - Richard Bramley
- GIE GoogLeNet
  - 67 Kernels for 67 Network Layers
- Faults in Latter Kernels
  - Generally Higher SDC AVF
- Weighted Average Safeness > 99 %
  - $Safeness = 1 - AVF_{SDC}$

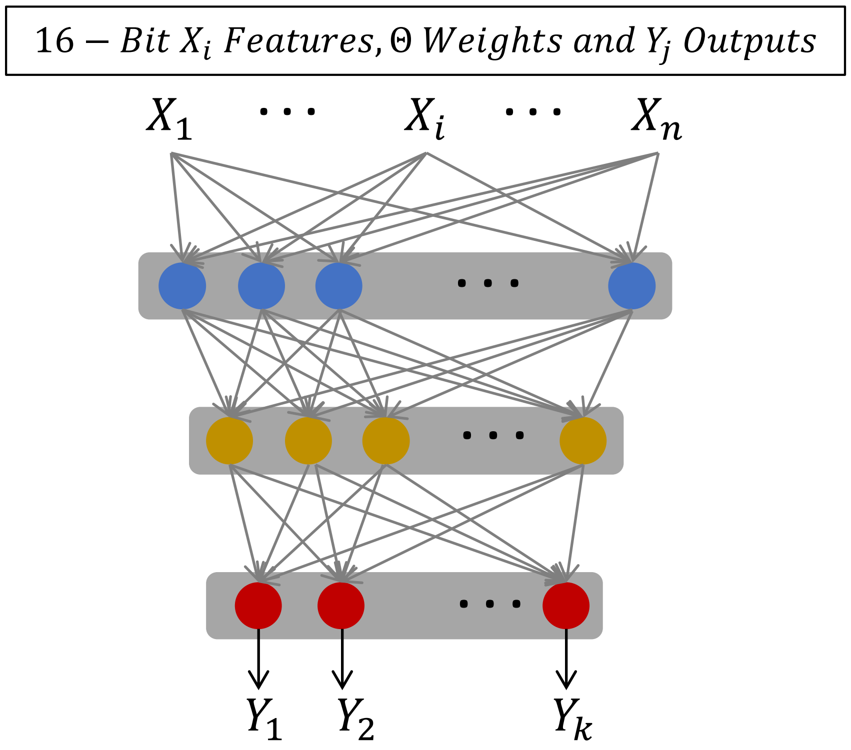


AVF for Feature X_i Error– Very Low



Higher DL Performance through Reduced Precision





2X
Performance
Improvement!!

DL Resilience with Reduced Precision?

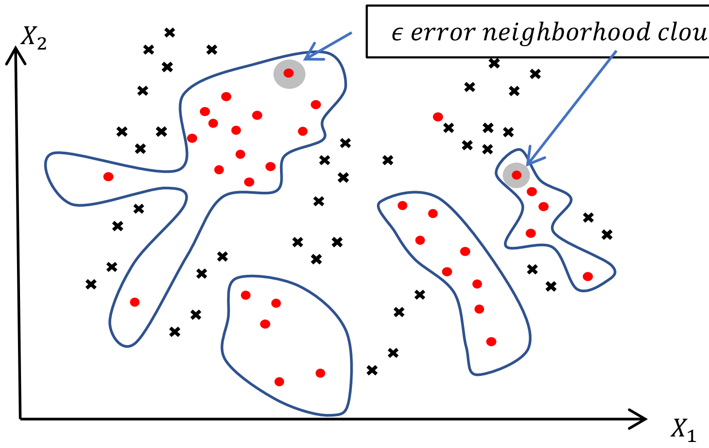
$32 - bit\ Precision\ and\ X_i > 2^{30}$

01|[29:23]	[22:0]

$16 - bit\ Precision\ and\ X_i > 2^{14}$

01|[13:7]	[6:0]

$$\boxed{\frac{\Delta X_i}{X_i} = \epsilon < 0.01}$$



Precision	Vulnerable Bits (Average)	Vulnerable Fraction (Average)	Raw FITs/Word (Relative)	Effective FITs/Word (Relative)
int32	22	68.75%	2	1.375
int16	14	87.50%	1	0.875
fp32	21	65.63%	2	1.313
fp16	12	75.00%	1	0.750

Resiliency Gets Better with Reduced Precision

DL Resilience for Control-Flow Faults?

- Neural Networks Implemented as Program Code
- Errors in Control-Flow
  - Program Counter, Instruction Bits
- SDC-AVF in the Range 20% to 40%
  - Requires Parity Protection & Self-Checking Code
- Recovery Strategy– Detect and Retry
  - Works for Transient Errors

Resiliency of Automotive Object Detection Networks on GPU Architectures

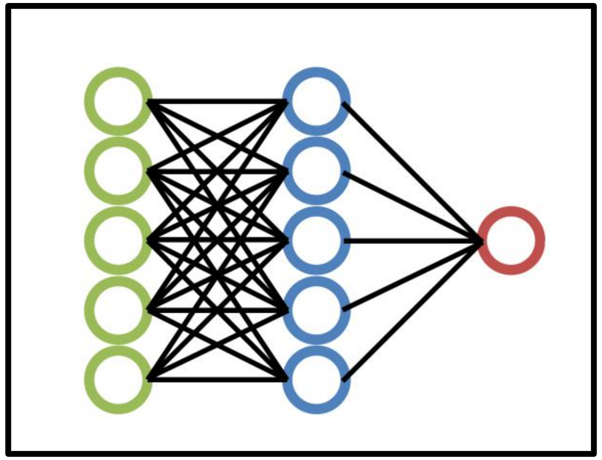
ITC 2019

Atieh Lotfi, Saurabh Hukerikar,
Keshav Balasubramanian, Paul Racunas,
Nirmal Saxena, Richard Bramley,
Yanxiang Huang

NVIDIA

What we already know …

Image classification networks are somewhat resilient to transient faults

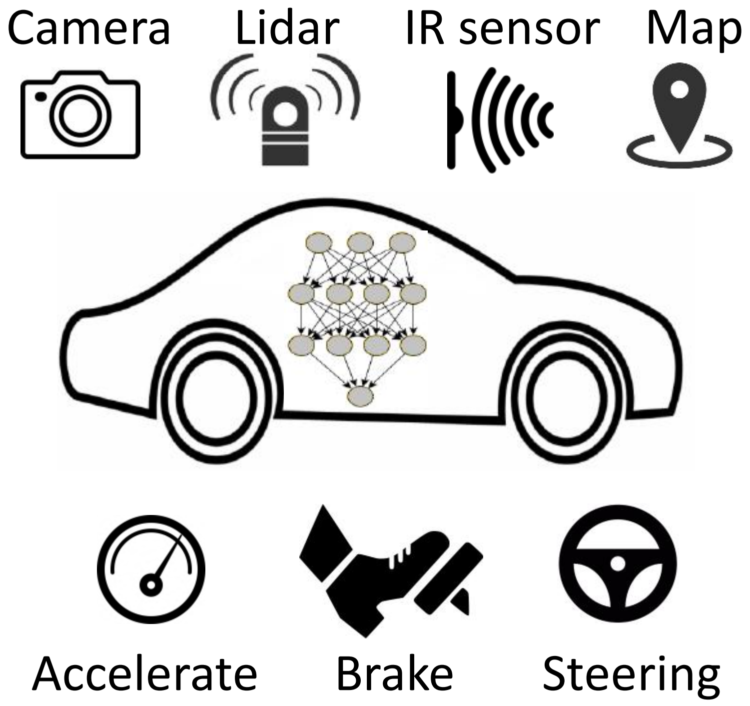


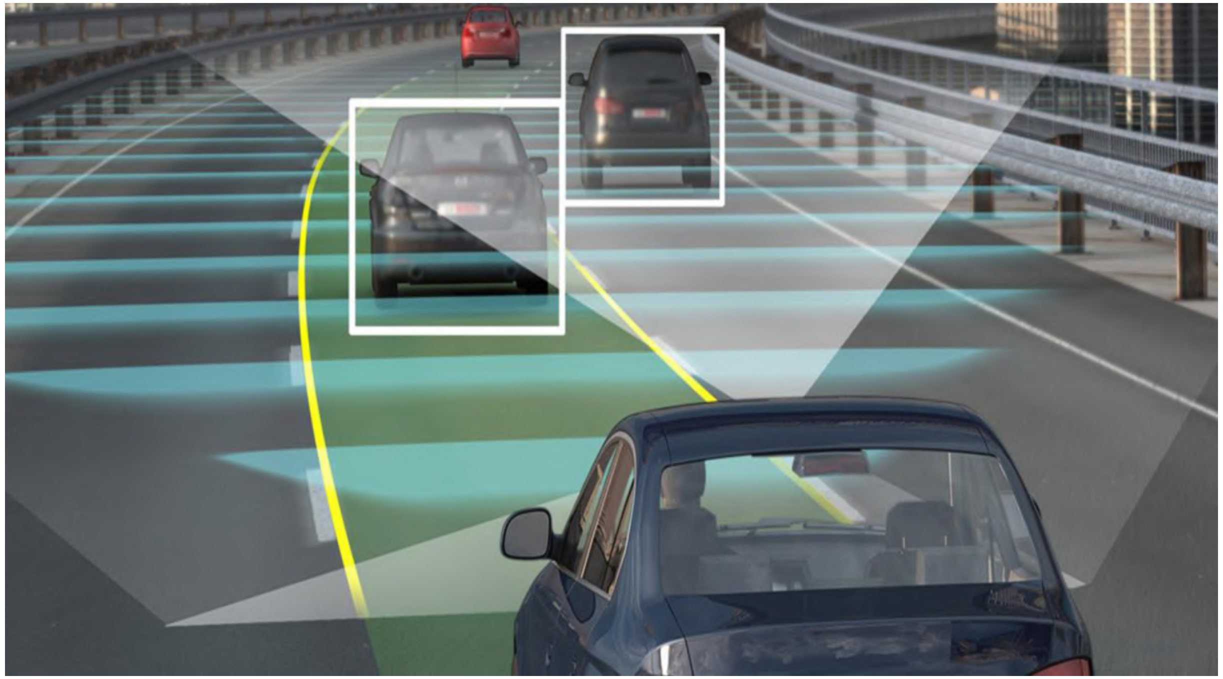
Car (Yes/No)
Pedestrian (Yes/No)
Traffic Sign (Yes/No)

Are **object detection** networks resilient to random hardware faults?

Object Detection Networks for Autonomous Driving

Object Detection: Image Classification + Object Localization





Path Planning and Navigation

Parallel architectures including GPUs are well-suited to accelerate DNNs

Safety in Autonomous Driving

ISO26262 failure rate requirement: 10 FIT for ASIL D compliance

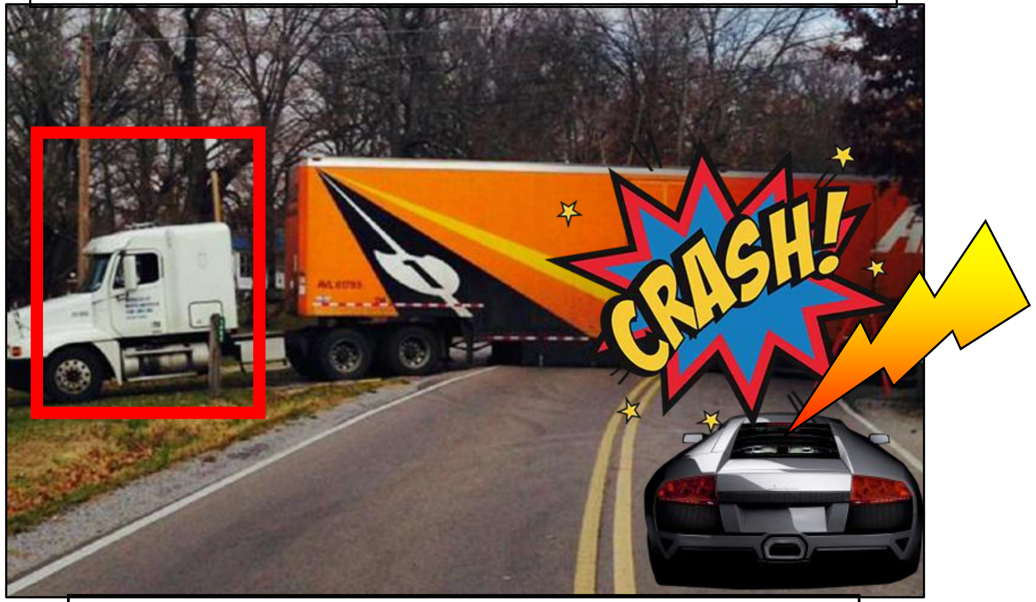
Random Hardware Faults in Automotive Object Detection networks

Does it violate safety goals?

Truck detected

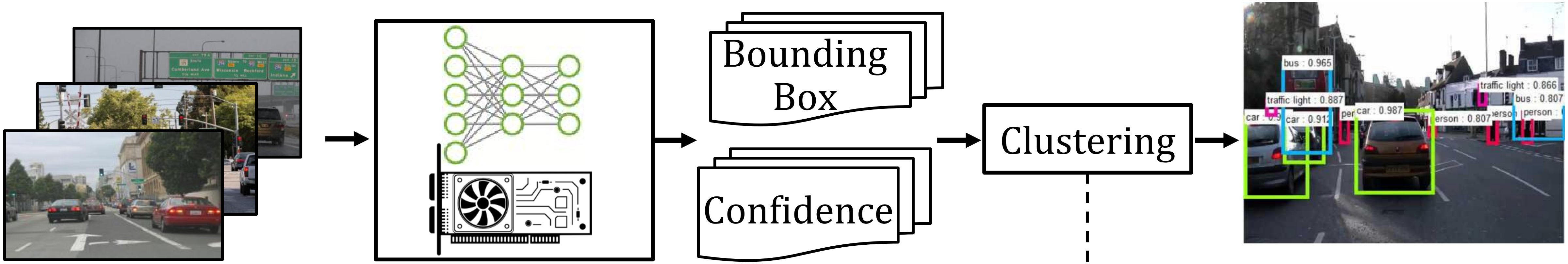
Action: Brake

Incorrect location detected



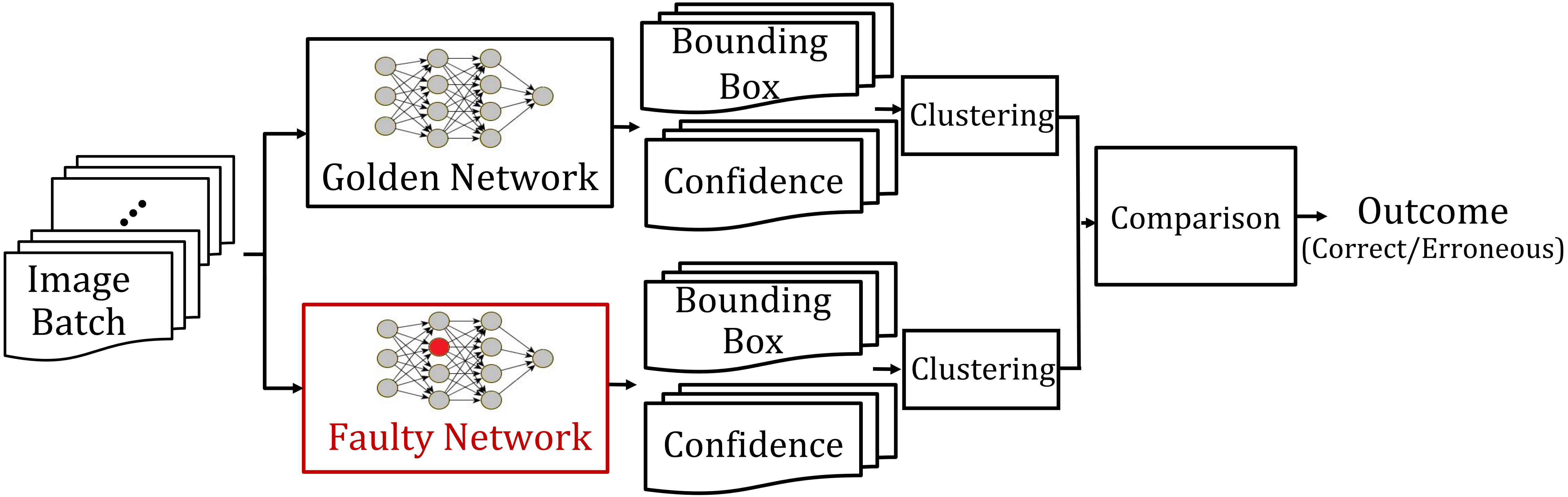
Action: Drive at 60 MPH

Object Detection Inference Networks



1. Discarding bounding boxes with low confidence
2. Clustering bounding boxes that have enough overlap:
   - $\dfrac{\text{Area of intersection}}{\text{Area of union}} > \textit{Threshold}$
   - Sum of confidence > *Confidence_level*

Fault Vulnerability Evaluation in Object Detection Networks



Fault Injection Outcome Comparison

For each detected object in the golden network

The same object is found in faulty network?

No

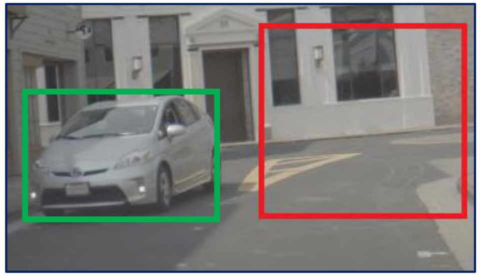
Yes

Enough overlap between bounding boxes in golden and faulty network

No

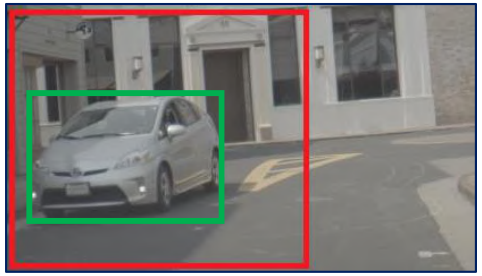
Yes

SDC



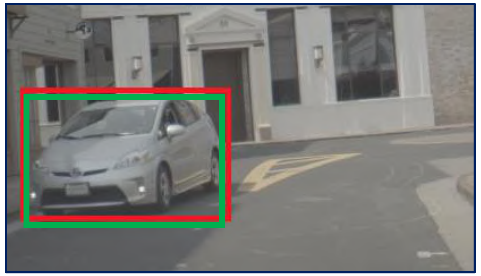
Unsafe

Inclusion



Safe / Unsafe

Masked



Safe

SDC: Silent Data Corruption

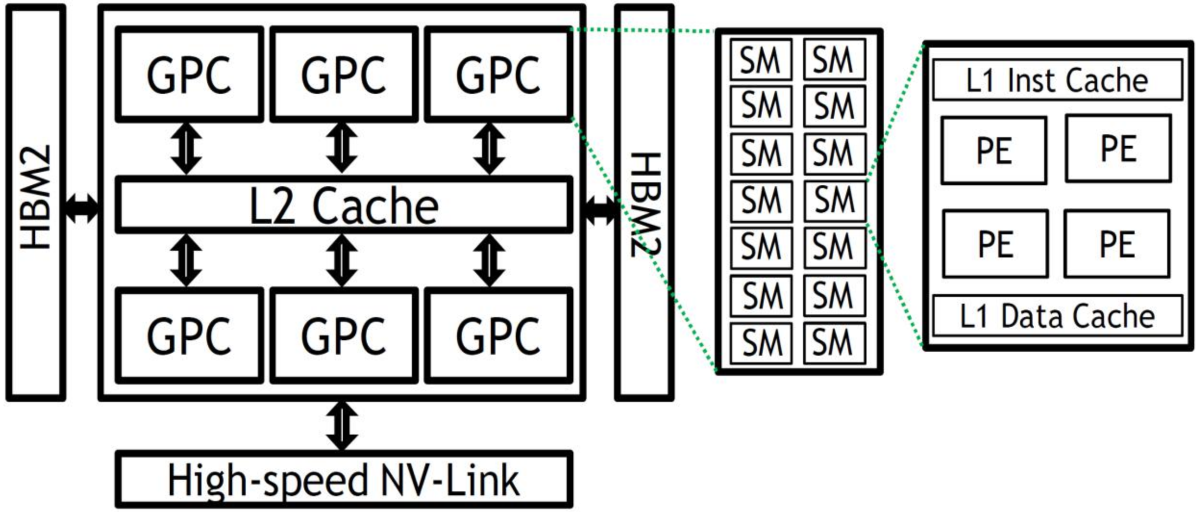
Bounding box in golden network

Bounding box in faulty network

Platform

- Automotive object detection network from NVIDIA DRIVE™ platform
  - TensorRT framework

- Inference on NVIDIA Volta Family GPU
  - HBM2
    - ECC
  - On-chip SRAMs
    - ECC or Parity



Transient Fault Injection

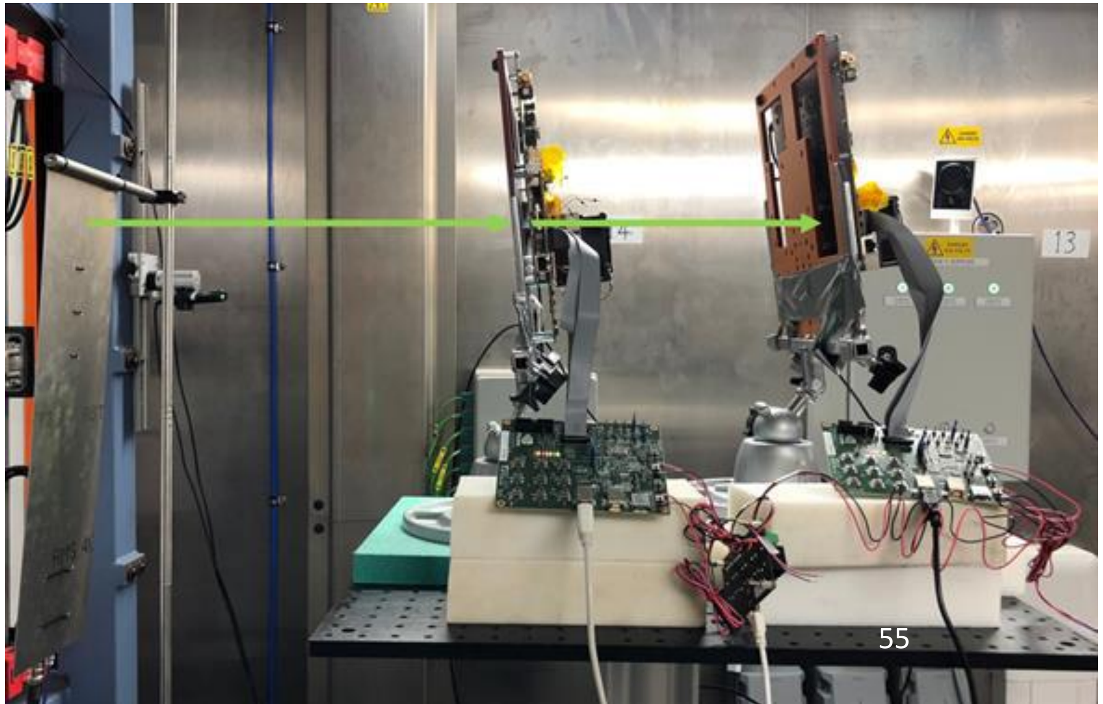
Accelerated Neutron Beam Testing

- Radiation experiments beam testing campaigns
  - Weapons Neutrons Research @ LANSCE
  - ChipIR microelectronics @ Rutherford Appleton Laboratory
- 2000 years of exposure to terrestrial neutron flux

- Experiment Design

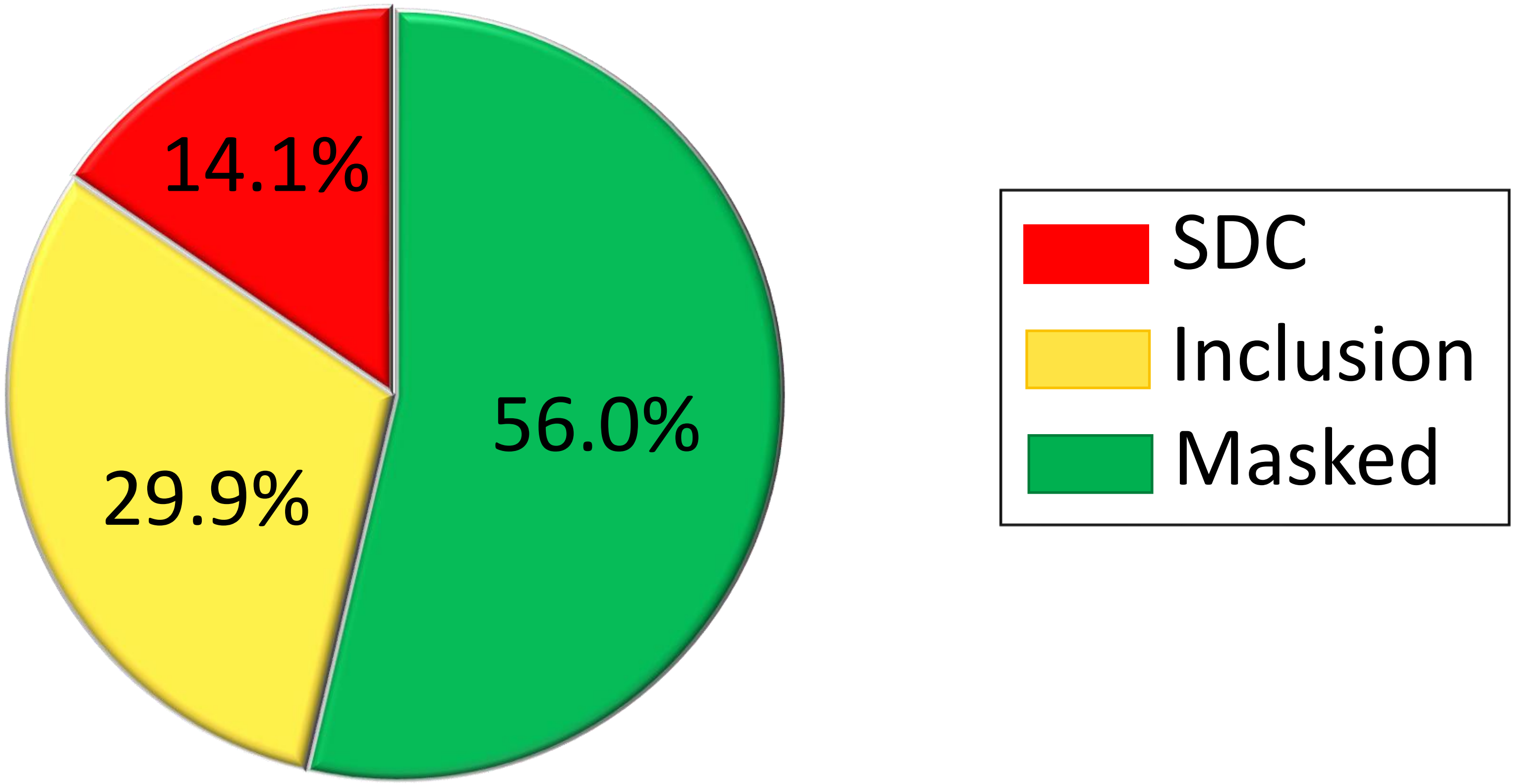
DRAM ECC	SRAM ECC
OFF	OFF
ON	OFF
ON	ON

Flight path of neutron beam



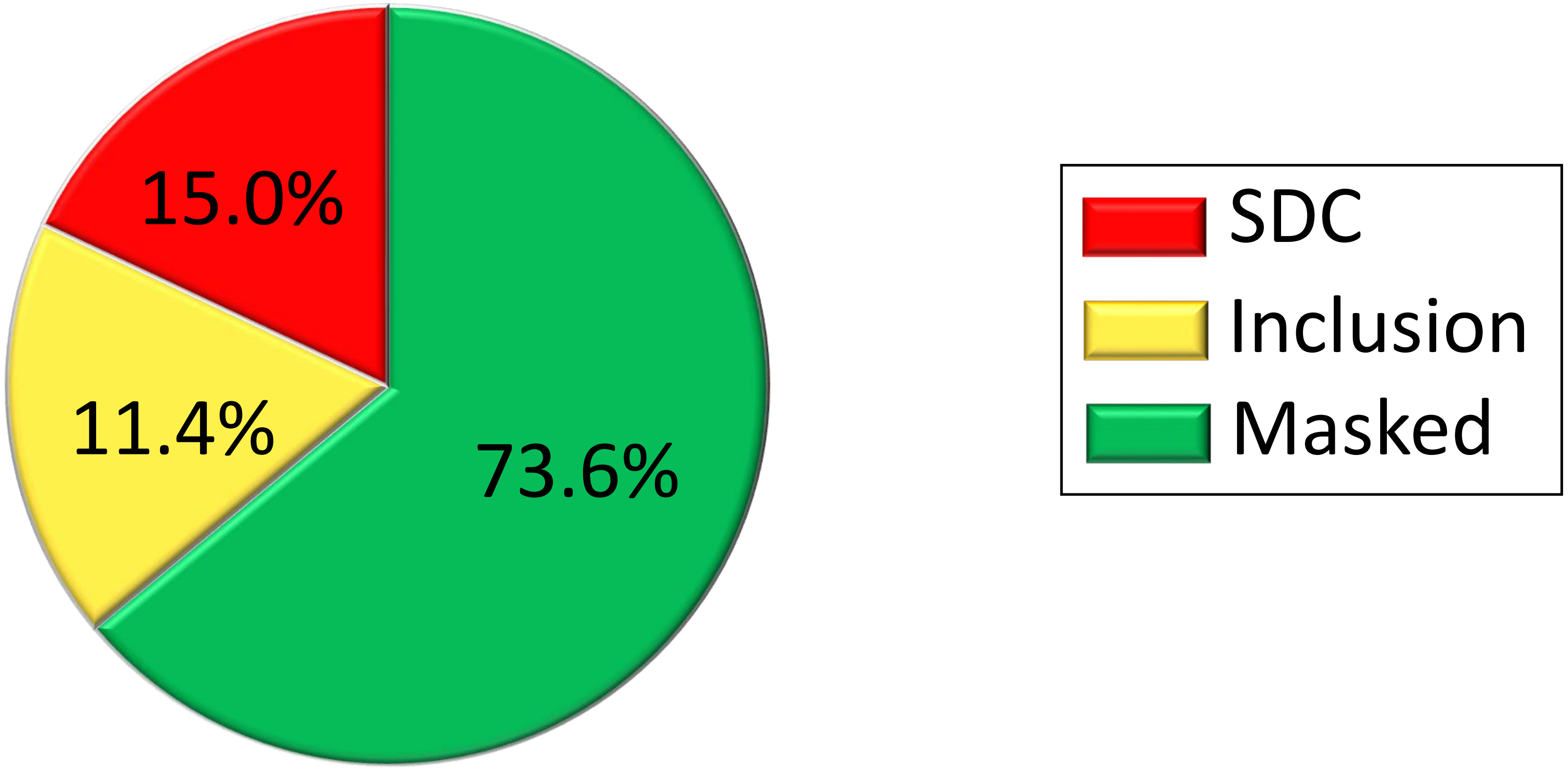
Accelerated Beam Testing Results

DRAM ECC	SRAM ECC
OFF	OFF



Accelerated Beam Testing Results

DRAM ECC	SRAM ECC
ON	OFF



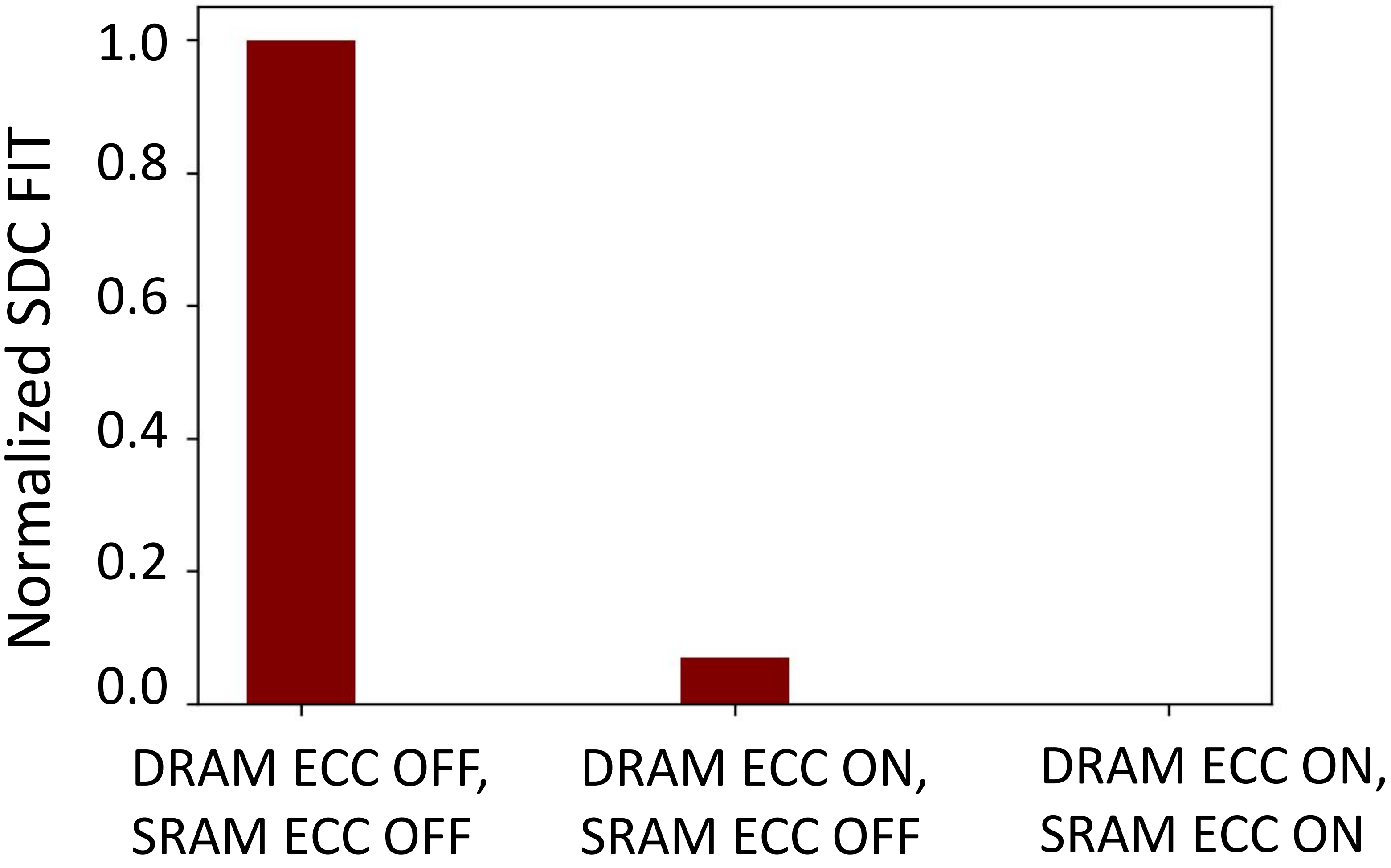
Accelerated Beam Testing Results

DRAM ECC	SRAM ECC
ON	ON

Masked
or
Detected

Zero SDC Events

Evaluation of Chip-level Protection Mechanisms in GPUs



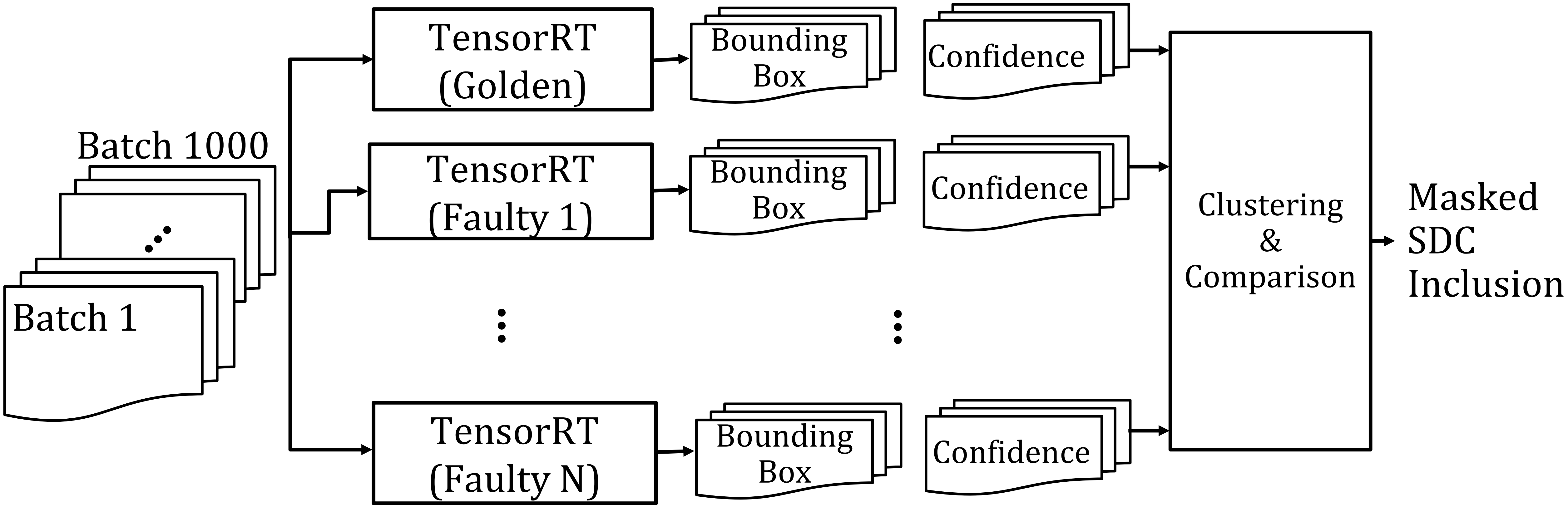
Permanent Fault Injection

Permanent Fault Injection

- Simulate injection of single-bit random permanent faults
  - Bit-flip in input image
  - Perturbing network weights

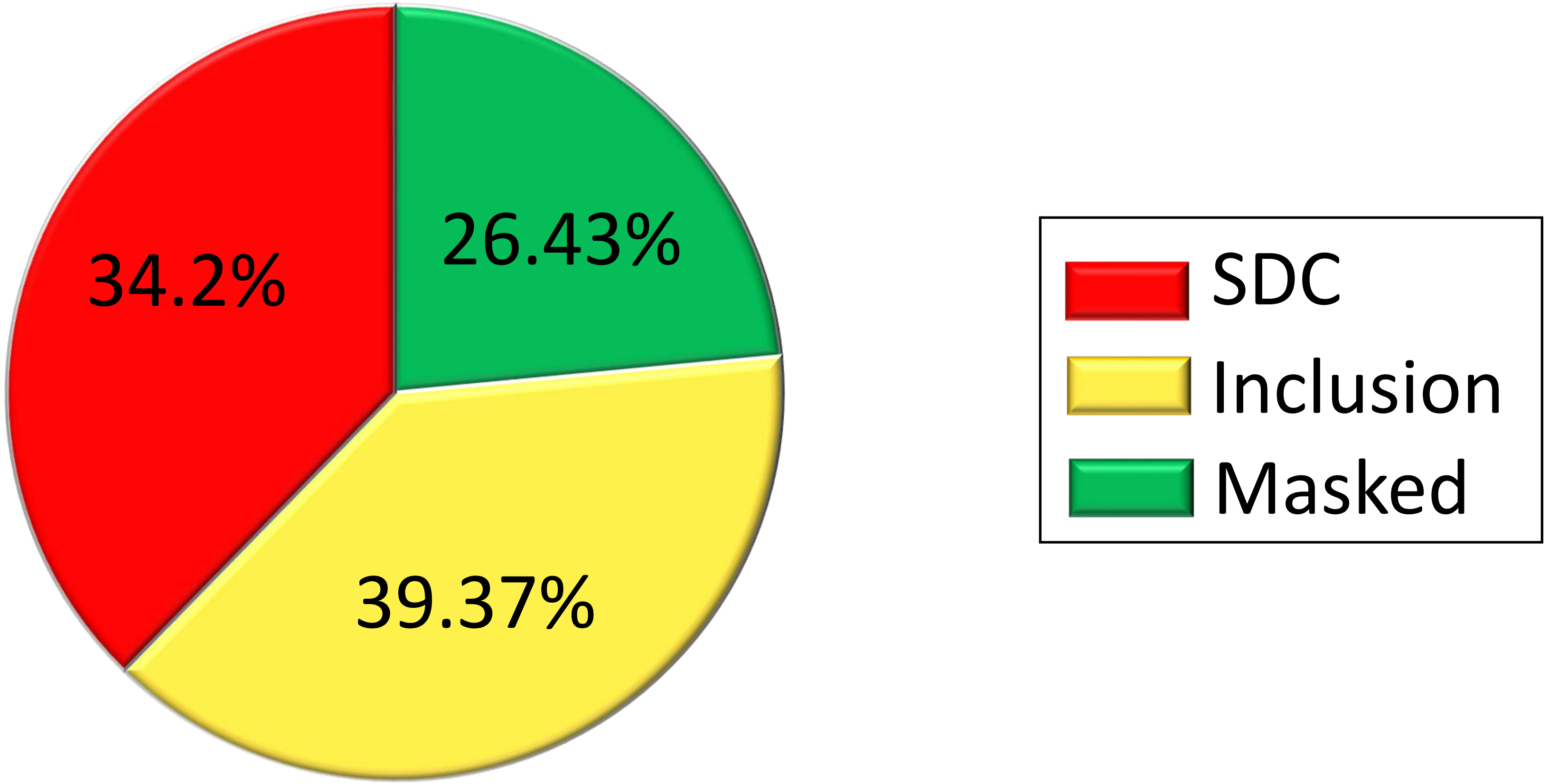
- Permanent fault experiments is a proxy

Permanent Fault injection on Network Weights



Permanent Fault Injection Results

- Faults in input batches: SDC (+ inclusion) < 1.8%
- Faults in weights:

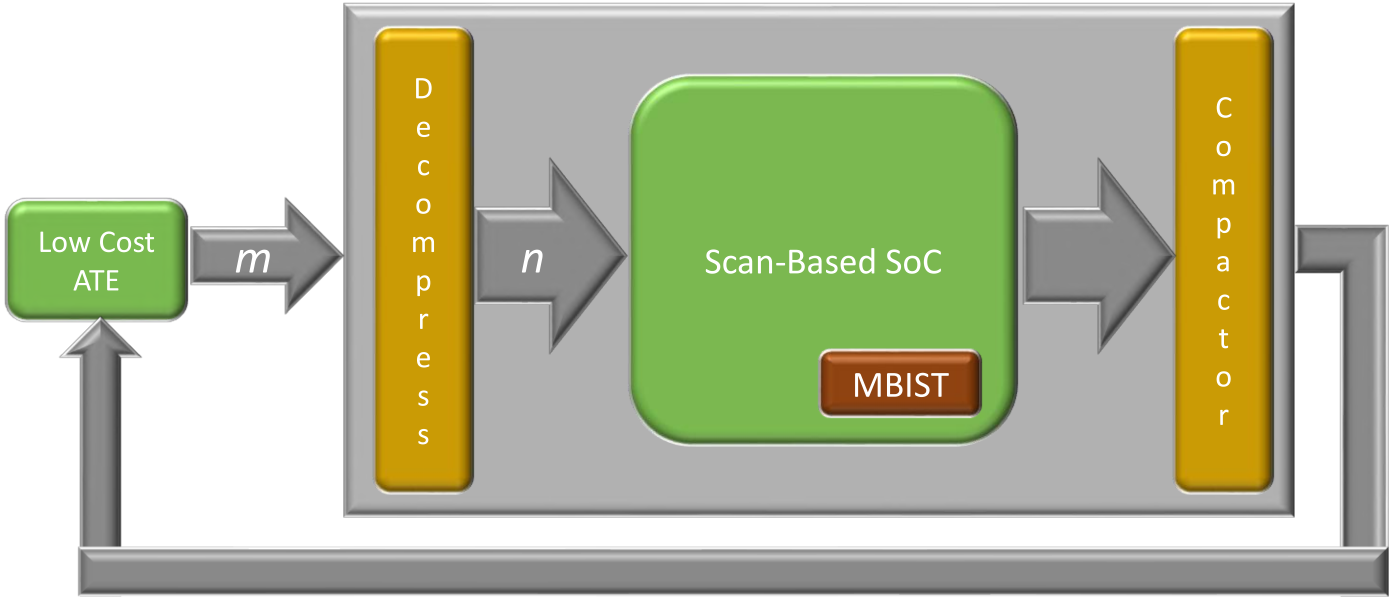


Object detection networks are vulnerable to permanent faults

Object Detection Conclusion

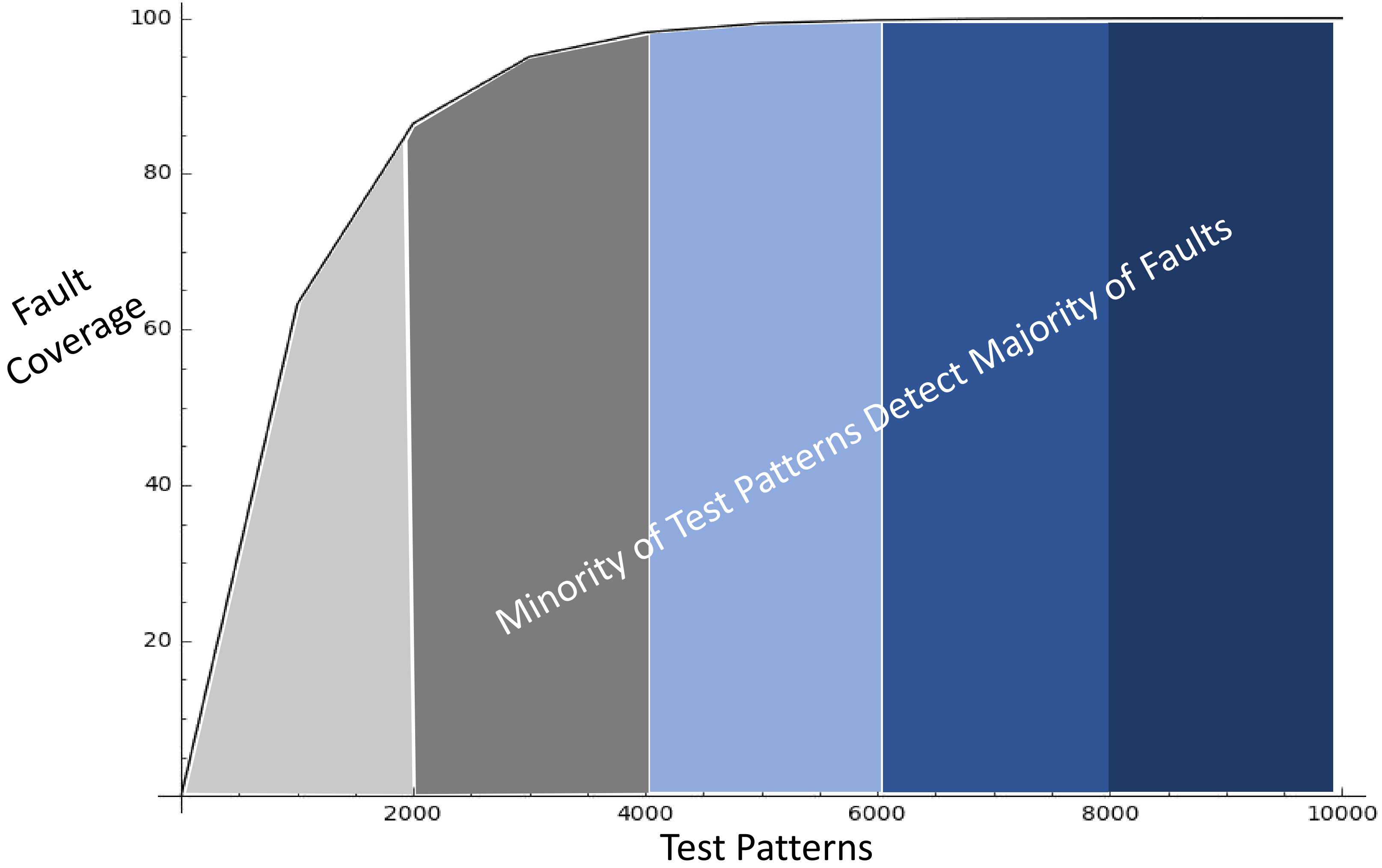
- Without protection– object detection networks show high SDC rate
  - Unlike classification networks that show resilience to transient errors
- Zero SDC with chip-level protections
  - For transient faults
- Not all permanent fault are detected by ECC/Parity:
  - Raw permanent FIT rate (hundreds) vs raw transient FIT rate (tens of thousands)
    - Offline structural tests during key-off and key-on events,
    - Online periodic tests (meeting FTTI requirement)

Leveraging Test Compression



- VLSI Test Principles and Architectures, 2006, Edited by: L-T Wang et. al.
- Chapter 6 [X. Li, K-J Lee, Nur Touba]
- [Reddy et. al. 2002] [Wûrtenberger 2004][[Jas 2003][Reda 2002][Han 2005b]
- [Chandra 2001][Krishna 2003][Rajski 2004][Hamzaoglu and Patel 1999][Li 2004]
- [Wang 2004][Wohl 2001][Das 2003][Mitra 2004]

Permanent Fault Coverage– Power Law



Permanent Fault Coverage Challenges

- Test Time < 3 millisecond
  - Fraction of Frame Time to Reduce Testing Overhead (< 10%)
- Periodic Test Power Usage
- Fast Context Switch
- Run-Time Process and Offline Structural Test
- Periodic Software Test as an Alternative
- Solves the Context-Switch Problem
- Coverage Evaluation Still an Issue (Hard to Meet 99% @FTTI)

Comments on Achievable SPFM Coverage

- ECC/Parity Achieves 100% Transient and Single Memory Element Permanent Faults
  - Not All SRAMs/Flip-Flops/Latches Can Be Parity Protected
  - Inefficient to Protect Logic Gates with Parity Protection
- Technical Approach to Address Uncovered Faults
  - SDC AVF for Applications with Natural Resilience
  - Augment Applications with Concurrent Error Detection (ABFT)
  - Periodic Software Diagnostic Tests to Meet FTTI for Permanent Faults
- 99% Permanent Fault Coverage Still NP-Complete Problem
  - Let Alone 99% @FTTI

Road to Resiliency

Redundant Execution– One Solution to Achieve > 99% SPFM (Internal Redundancy)

Detect & Retry Does Not Work for Permanent Faults

Error Signals Still Needed

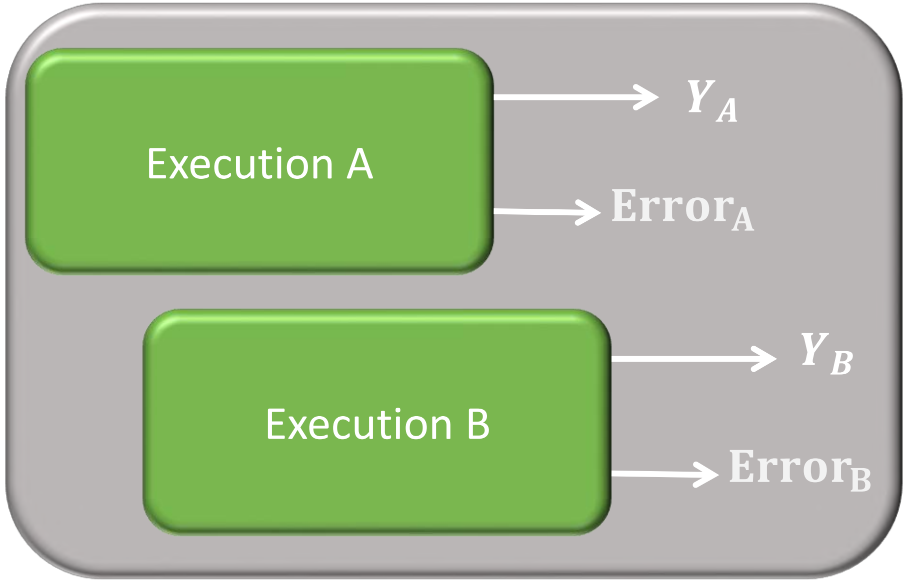
- Single-Point Fault Tolerance

Similar to Erasure Codes

- Mirrored RAID
- Identify Correct Copy

Execution Instances

- On Non-Overlapping Hardware



Markov Chain Analysis– Need Physical Redundancy

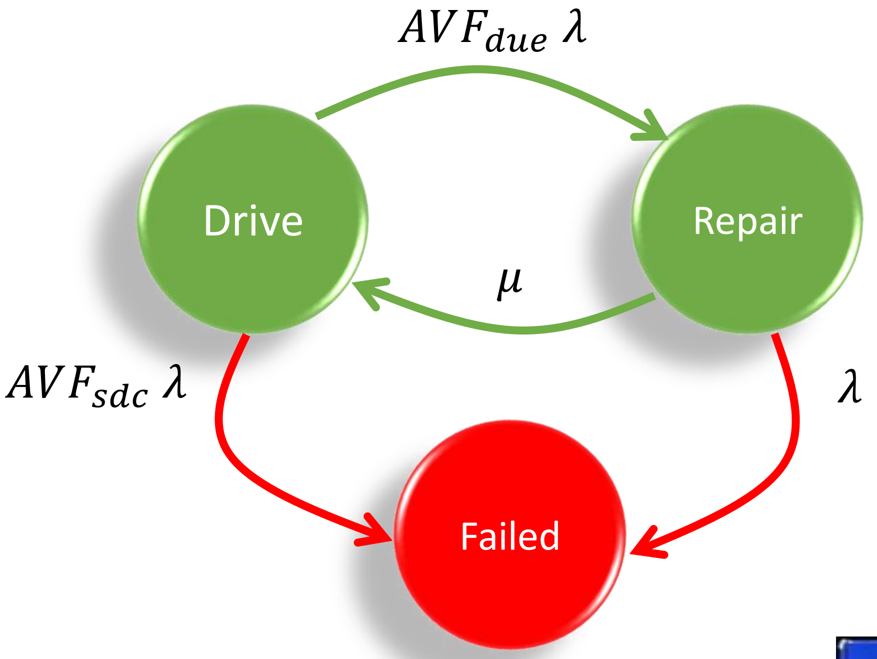
Availability is Important Here

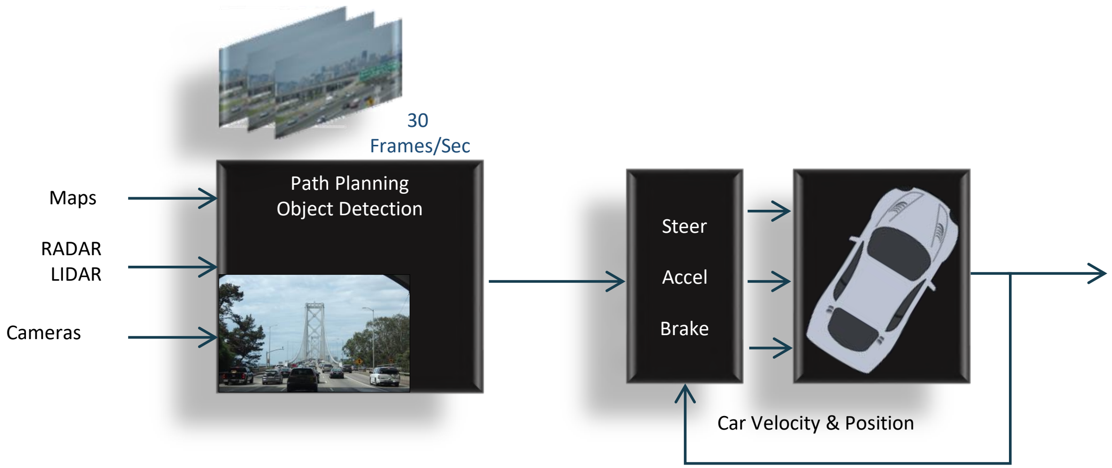
For Driverless Car

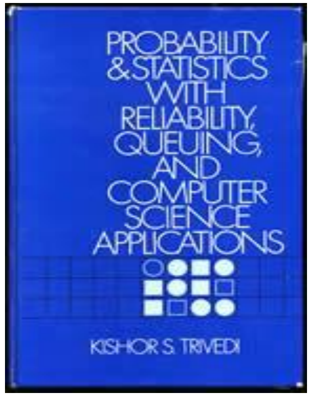
Loss of Frames => Loss of Life

For 3 Frame-Tolerance, Need

$$\frac{1}{\mu} < 100ms$$







Dual Redundant System

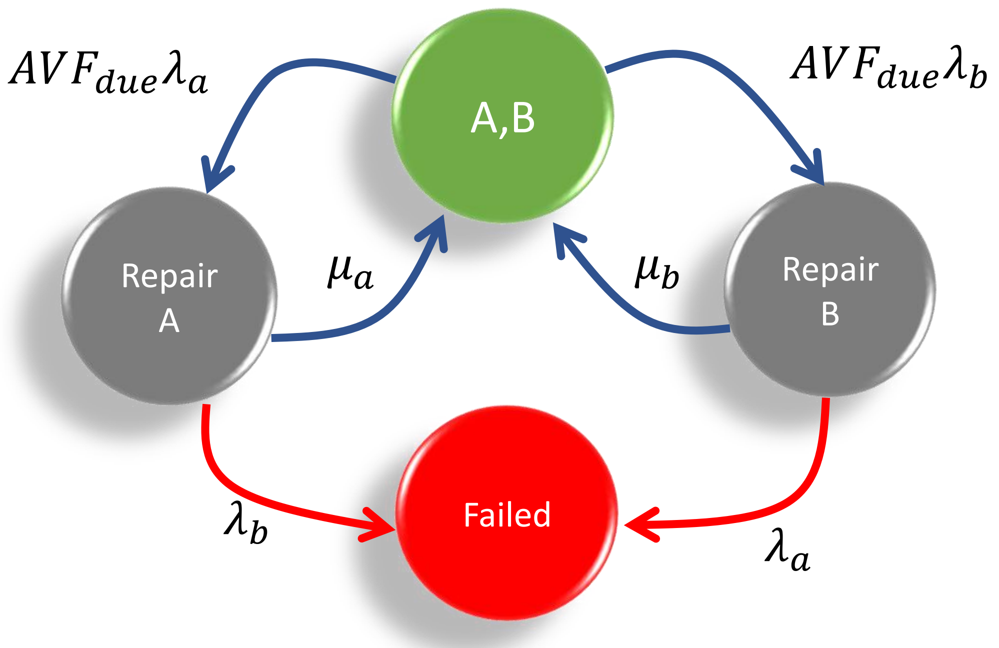
Relaxed Constraints on Repair Rate

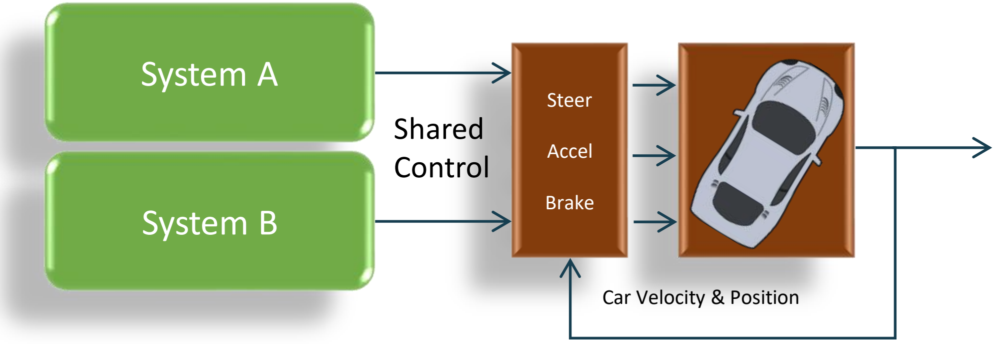
$$\frac{1}{\mu_a} < \frac{1}{\lambda_b}$$

$$\frac{1}{\mu_b} < \frac{1}{\lambda_a}$$

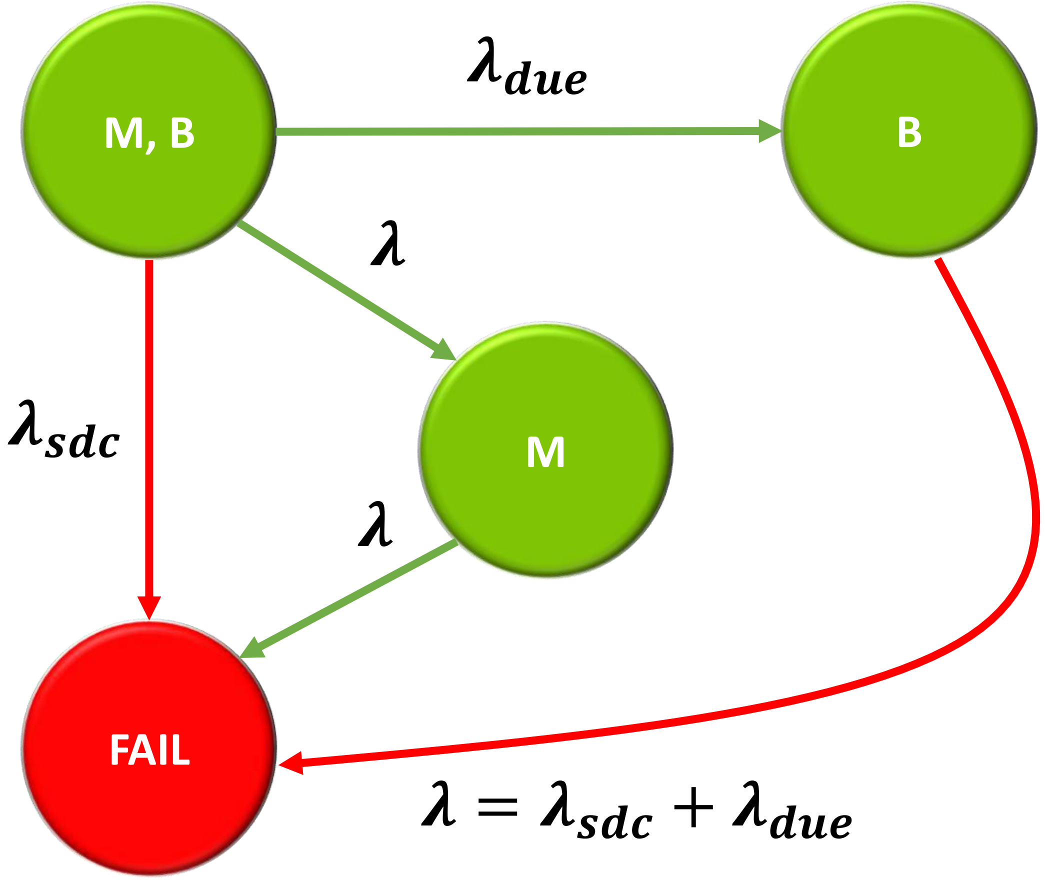
$\frac{1}{\lambda_a}$ *or* $\frac{1}{\lambda_b}$ in the order 1000's of hours

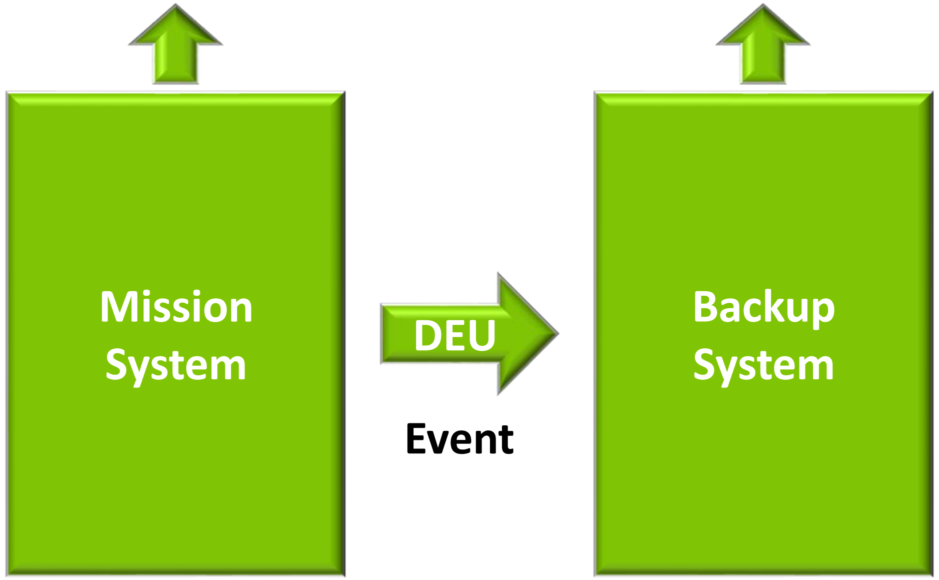
Repair can wait till the next Key-Off Event





Backup Standby Model– Markov Chain





Probability of Backup Markov Chain States

$$Probability\ of\ being\ in\ M, B\ state, P_{m,b}(t) = e^{-2\lambda t}$$

$$Probability\ of\ being\ in\ B\ state, P_b(t) = \frac{\lambda_{due}}{\lambda}\left(e^{-\lambda t} - e^{-2\lambda t}\right)$$

$$Probability\ of\ being\ in\ M\ state, P_m(t) = e^{-\lambda t} - e^{-2\lambda t}$$

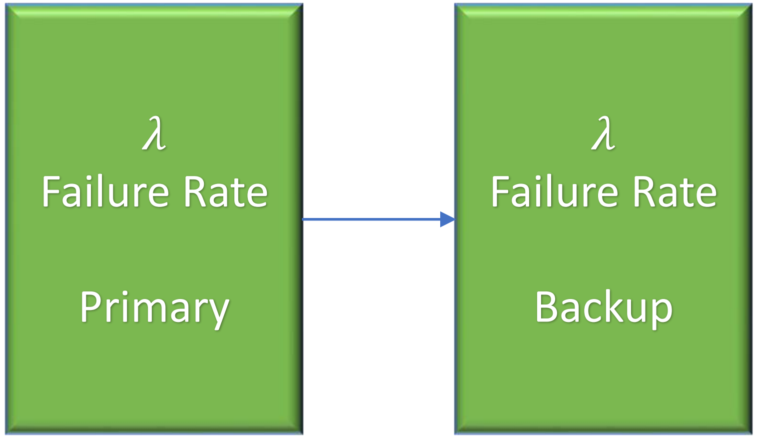
$$Probability\ of\ being\ in\ Fail\ State, F(t) = 1 - \left(\frac{\lambda + \lambda_{due}}{\lambda}\right)e^{-\lambda t} + \frac{\lambda_{due}}{\lambda}e^{-2\lambda t}$$

$$\boldsymbol{MTTF = \int_0^\infty t\frac{dF(t)}{dt}dt = \frac{1}{\lambda} + \frac{\lambda_{due}}{2\lambda^2}\ asymtotically\ approaches\ \frac{3}{2\lambda}\ (when\ \lambda_{sdc} = 0)}$$

1.5x Gain in MTTF over Simplex or 1.5x Reduction in Effective Failure Rate over an infinite drive time

Is MTTF Sufficient to Distinguish Two Systems?

Duplex System



$$Duplex\ MTTF = \frac{3}{2\lambda}$$

$$\neq$$

Simplex System

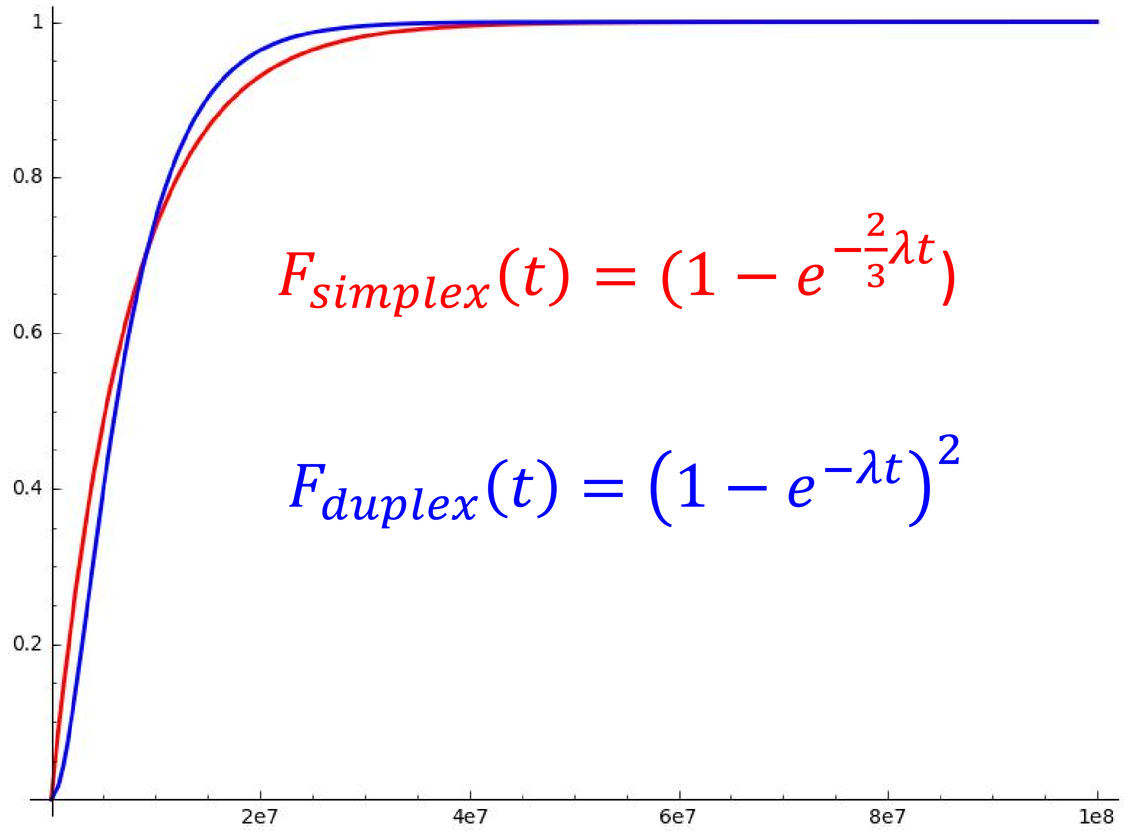
$$Simplex\ MTTF = \frac{3}{2\lambda}$$

Failure Probability Reduction metric as a function of mission time distinguishes various redundant systems [Mitra, Saxena, McCluskey 2004]. A related work was cited in **ISO DIS 26262-11:2016(E)**

S. Mitra, N.R. Saxena, and E.J. McCluskey, "Efficient Design Diversity Estimation for Combinational Circuits," *IEEE Trans. Comp.*, Vol. 53, Issue 11, pp. 1,483-1,492, Nov. 2004

S. Mitra, N.R. Saxena and E.J. McCluskey, "Common-Mode Failures in Redundant VLSI Systems: A Survey," *IEEE Trans. Reliability*, Special Issue on Fault-Tolerant VLSI Systems, Vol. 49, Issue 3, pp. 285-295, Sept. 2000.

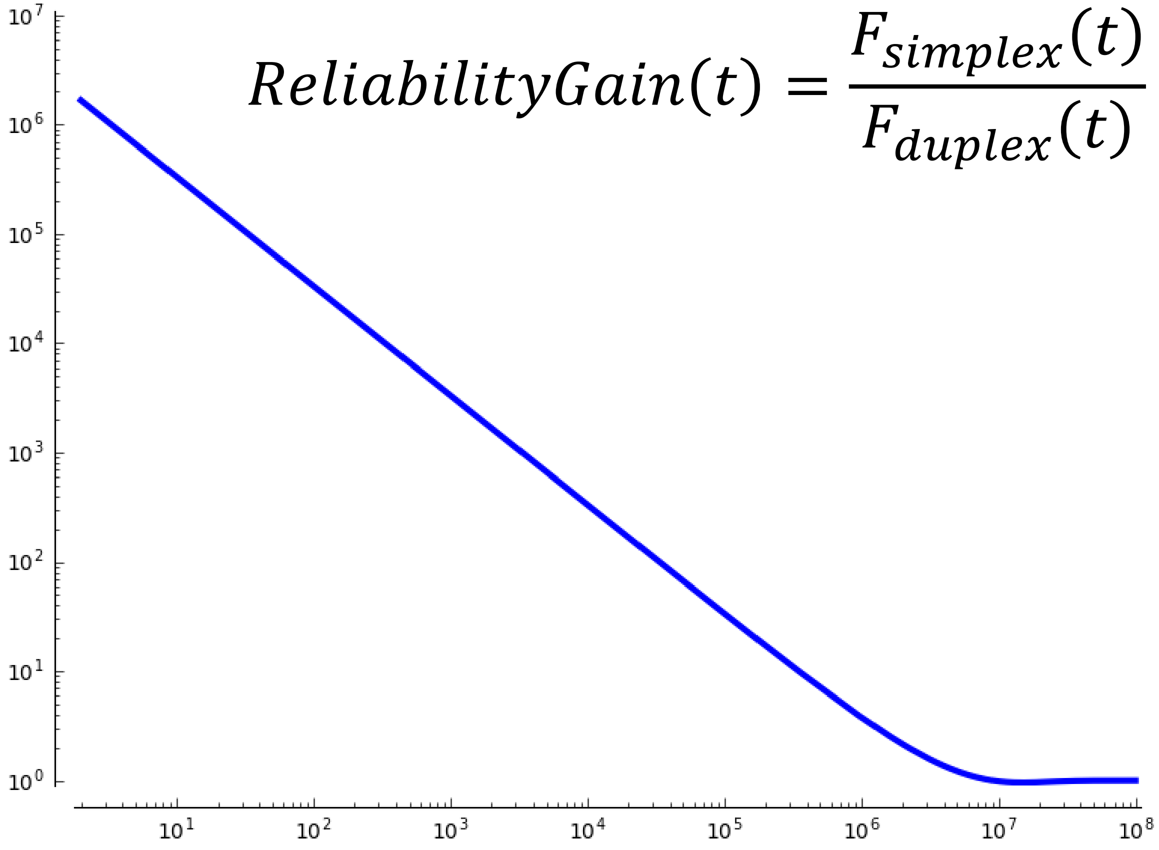
Reliability Gain with Perfect Duplex $\times 10^6$ *in 2 Hour Drive Time*



$$F_{simplex}(t) = (1 - e^{-\frac{2}{3}\lambda t})$$

$$F_{duplex}(t) = (1 - e^{-\lambda t})^2$$

Drive Time in Hours

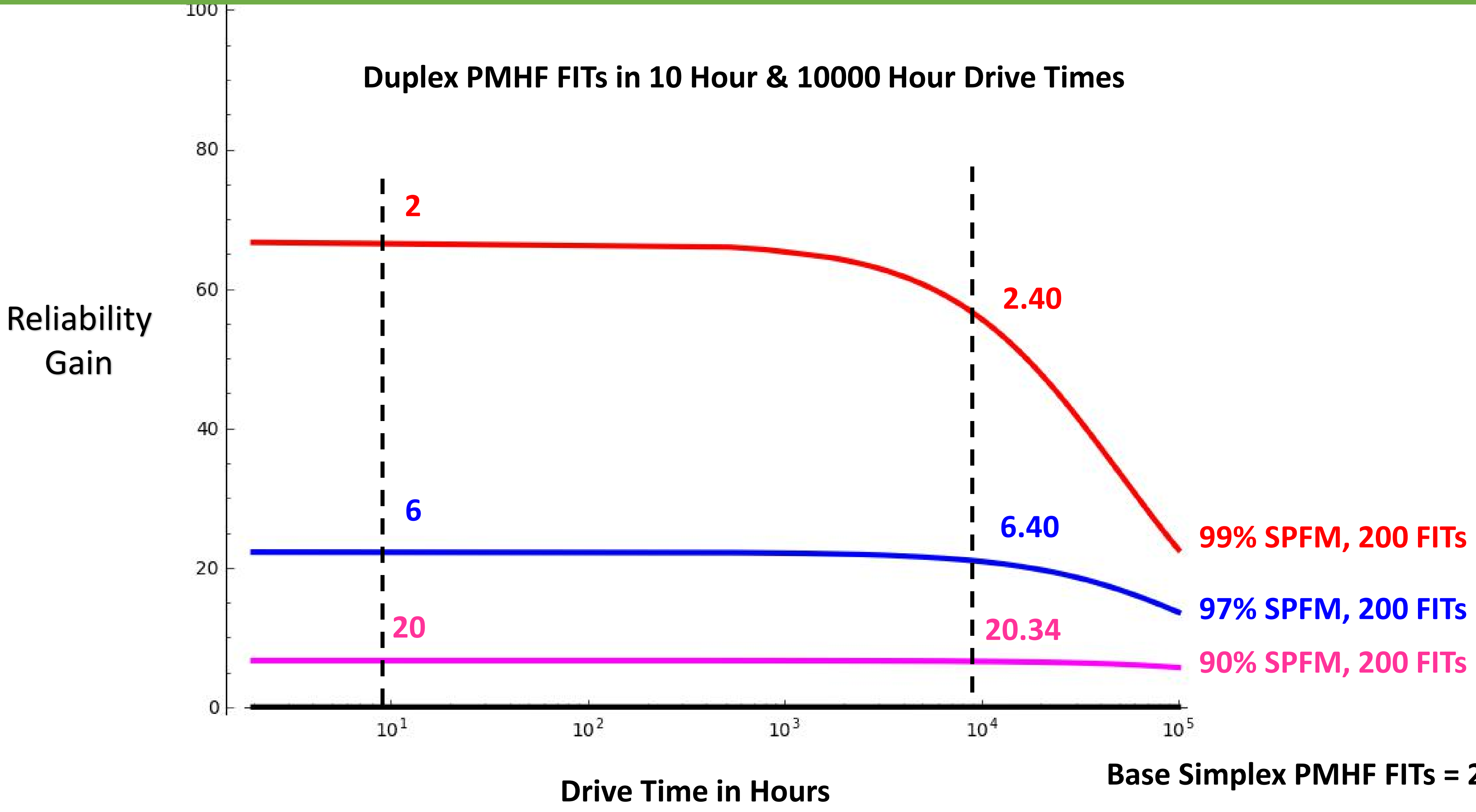


$$ReliabilityGain(t) = \frac{F_{simplex}(t)}{F_{duplex}(t)}$$

Drive Time in Hours

$$\lambda = 200\ FITs$$

Back-Up Standby Model– SPFM Sensitivity

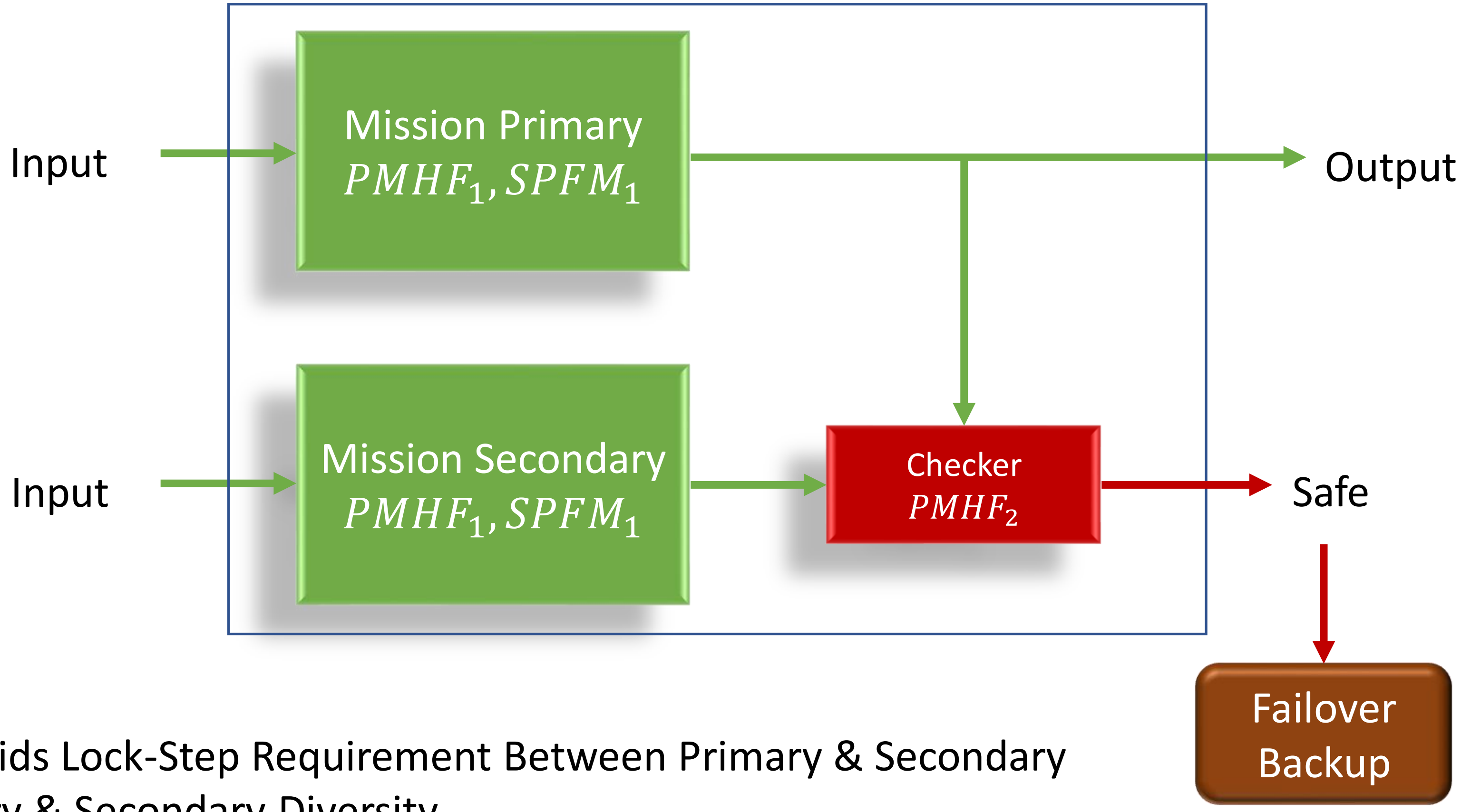


Base Simplex PMHF FITs = 200

Duplex System with Decoupled Checker

Probability Drive System Fails == Mission Primary Fails & Checker Fails

$$(1 - e^{-PMHF_1 \times T/10^9})(1 - e^{-PMHF_2 \times T/10^9})$$



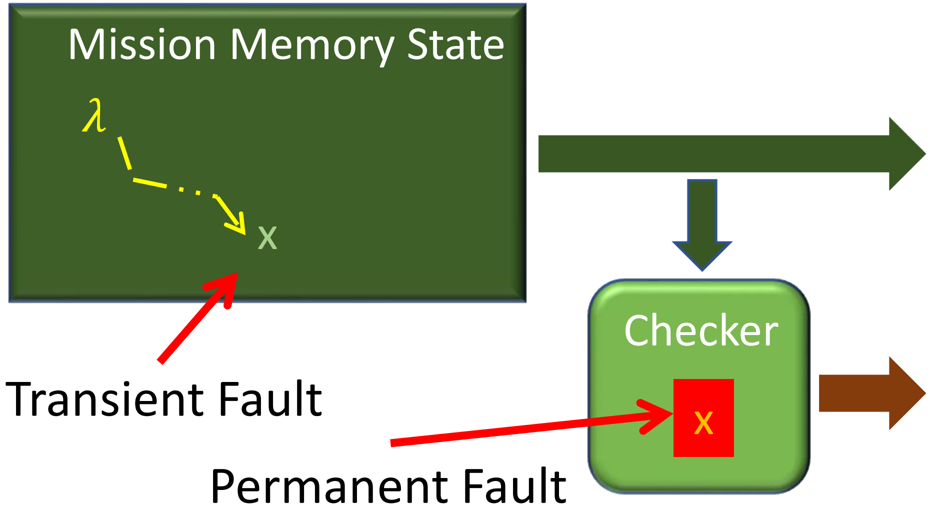
Decoupled Checker Avoids Lock-Step Requirement Between Primary & Secondary

- Important for Primary & Secondary Diversity

Duplex with Decoupled Checker– SPFM Sensitivity

Mission Raw Failure Rate (FITs)	Mission SPFM	Mission PMHF	Checker PMHF	Drive System PMHF (MT = 1Hr)	Drive System PMHF (MT = 10Hrs)	Drive System PMHF (MT = 1000 Hrs)
1000	50%	500	10	0.0003	0.003	0.3
2000	60%	800	100	0.0006	0.006	0.6
4000	60%	1600	200	0.003	0.03	3.0
8000	70%	2400	500	0.006	0.06	6.0
10000	50%	5000	500	0.025	0.25	**25.0**

Latent Fault Metric– LFM



Percentage of **Fault-Secure** Permanent Faults in the Checker

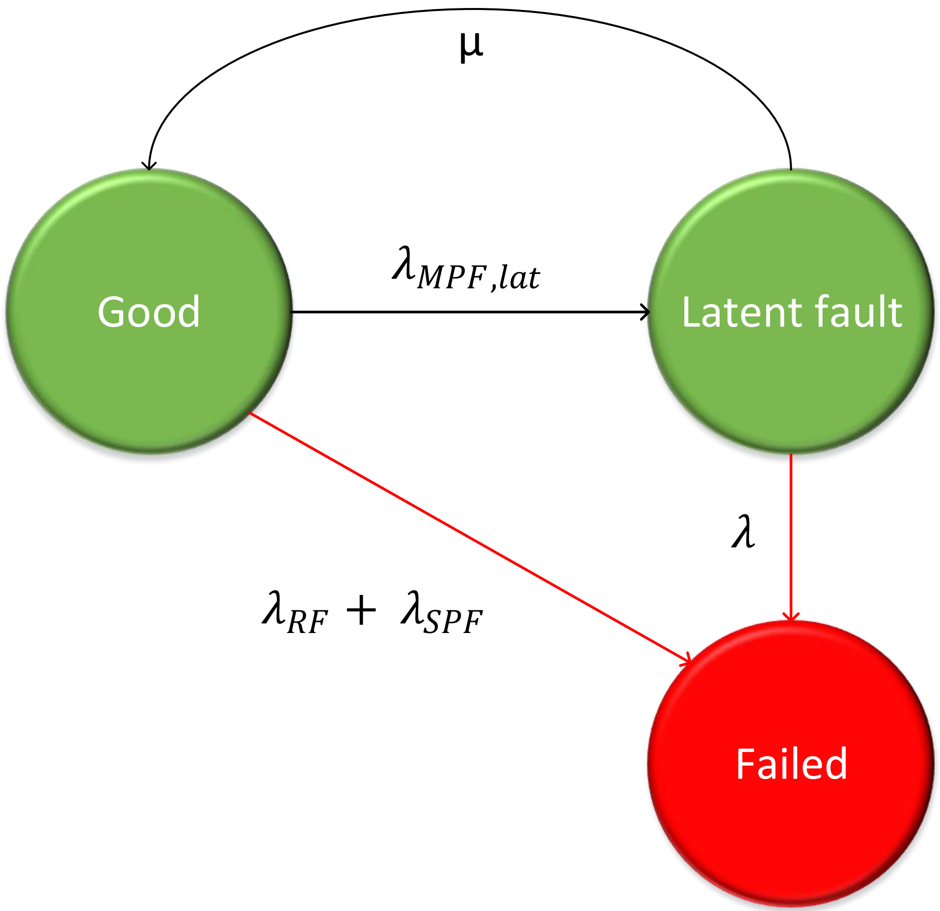
How to Detect Latent Fault?

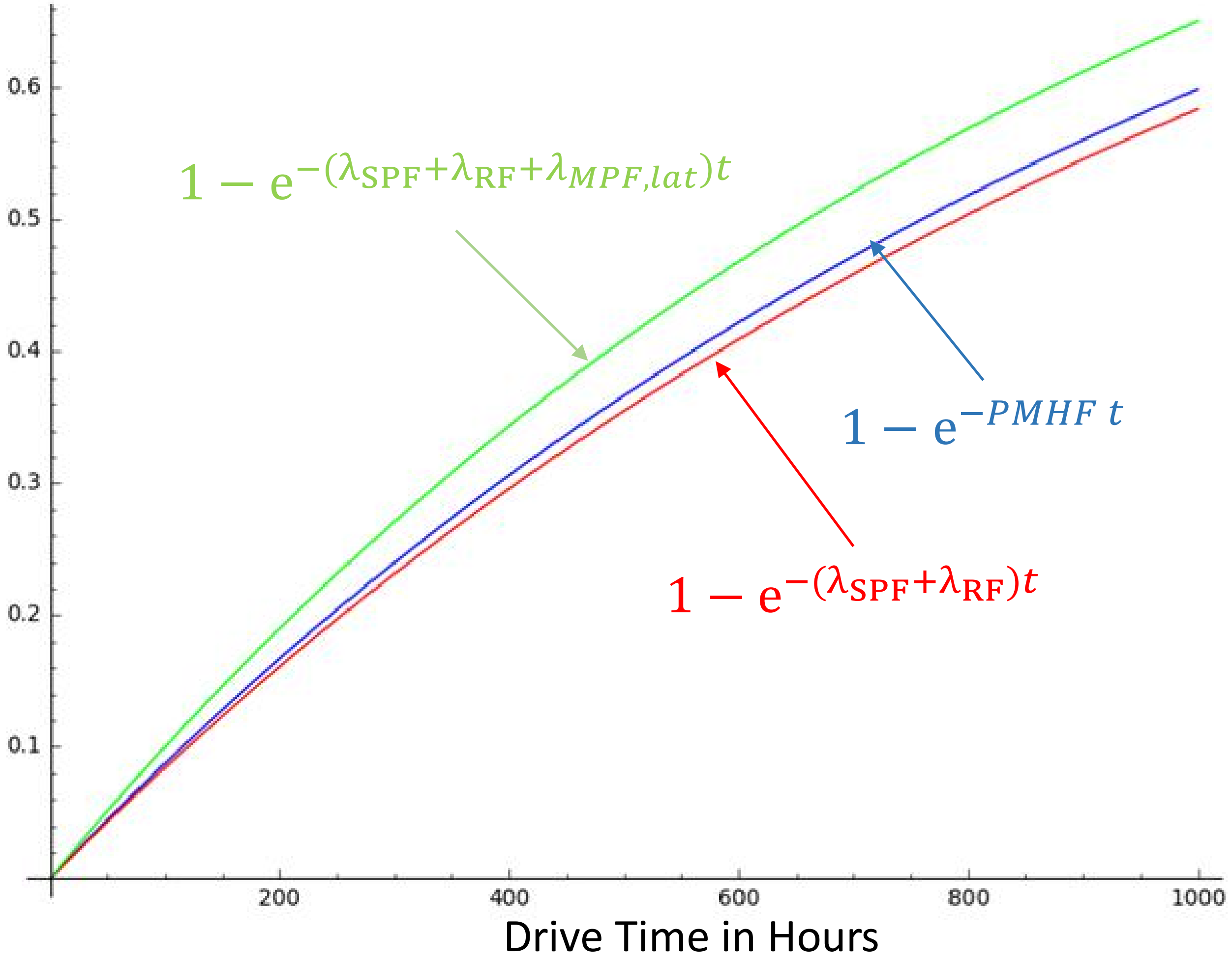
- Use Permanent Fault Tests– Works Only During Periodic Tests
    - Not an Issue as MTTI is Drive Time
- Self-Checking Checker– Works During Run-Time
- Software Based Checker– Use Algorithm Based Fault Tolerance (ABFT)

- Totally Self-Checking Circuits [Andersen & Metze 1973]
- [Ashjaee & Reddy 1976] and ABFT [Huang & Abraham]

Relationship of LFM and PMHF?

$$PMHF = \lambda_{RF} + \lambda_{SPF} + f\left(\lambda_{RF} + \lambda_{SPF}, \lambda_{MPF,lat}, \mu, \lambda\right) = AVF_{SDC}\lambda + f\left(AVF_{SDC}\lambda, \lambda_{MPF,lat}, \mu, \lambda\right)$$





$$\lambda_{RF} + \lambda_{SPF} \leq PMHF \leq \lambda_{RF} + \lambda_{SPF} + \lambda_{MPF,lat}$$

What is the Current FIT Rate for Systematic Faults?

Systematic Faults	Observed Bug Rate	FIT Rate
Hardware Design Faults	**3 Bugs in 48 Years**	**7000**
Software Design Faults	**1 Bug Every Year**	**100000**

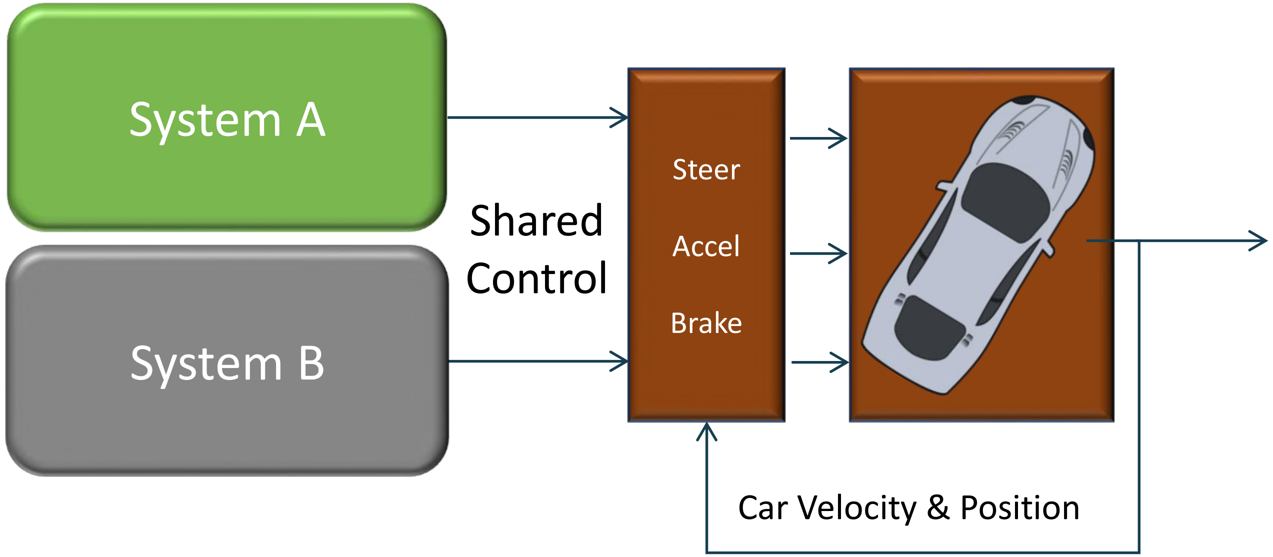
Mitigating Factors

- Automotive Environment is More Constrained

Hardware Design Quality– Need Three Orders of Improvement

Software Design Quality– Need Four Orders of Improvement

Design Diversity



Coping with Systematic Hardware and Software Design Errors

- [Siewiorek et. al. 1978] (byte reversal copies C.mmp processor)
- [Sedmak and Liebergot 1980] (complementary function diversity in VLSI)
- [Chen and Avizienis 1978] (N-version programming, SIFT software implemented fault-tolerance)
- [Horning et. al 1974] (Recovery Blocks) [Patel] RESO Technique
- [Amman and Knight 1987] (Data Diversity)
- [McCluskey, Saxena, Mitra 1998] Diversity for Reconfigurable Logic & Quantifying Diversity

Conclusions

PMHF Metric is the Only Metric that Matters

- ASIL Compliance of SPFM Coverage Metric is Neither Necessary Nor Sufficient

Road to Resiliency ⇒ Dual Physical Redundancy

- Concurrent Permanent Fault Testing
  - SPFM 100% @FTTI for Hardware Random Faults
- Higher Availability During Drive Time (Mission Time)
  - Almost Zero PMHF for Drive Times Less than 100 Hours

Systematic Faults

- Rigorous Testing and Validation

  Need 3-to-4 Orders of Improvement

- Physical Redundancy with Design Diversity