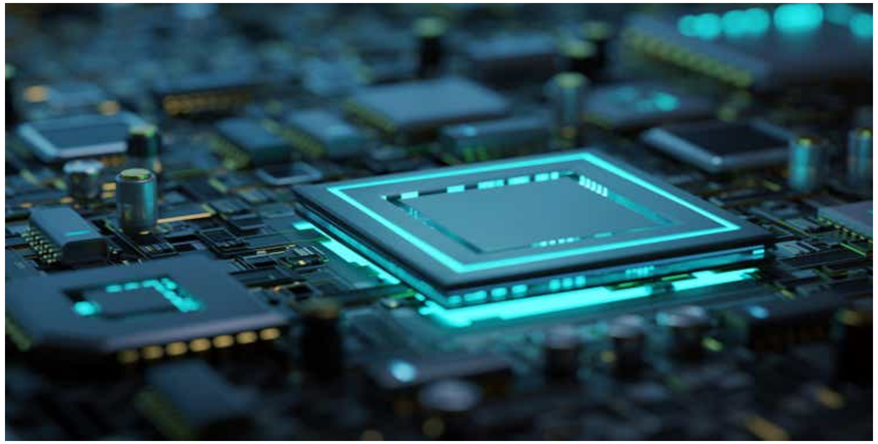
Characterizing Deep Learning Neural Network Failures between Algorithmic Inaccuracy and Transient Hardware Fault

Sabuj Laskar, Md Hasanur Rahman, Bohan Zhang, Guanpeng Li

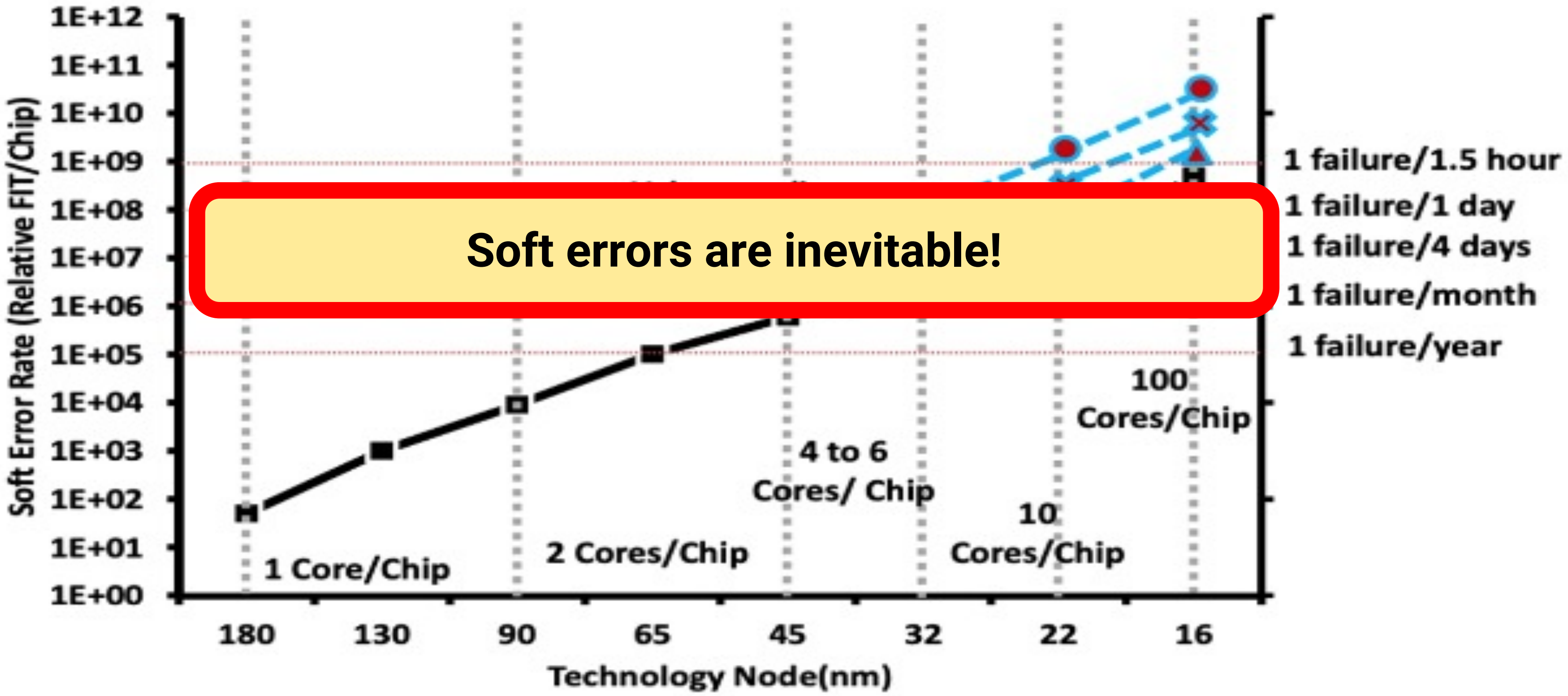
IOWA

Motivation

- DNN has been increasingly deployed in many areas
  - Computer vision, NLP, autonomous vehicles (AVs)
- DNN reliability becomes important
  - ISO 26262 safety standard requires no more than 10 FIT (10 failures in every 10^9 hours)



Soft Error



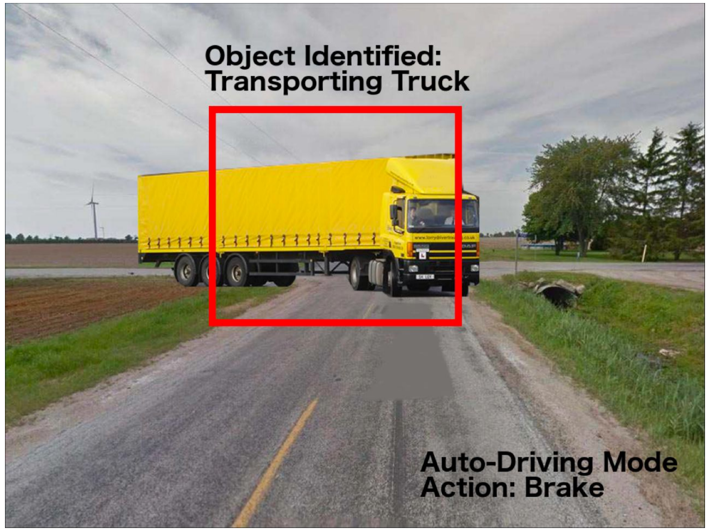
Taken from [1]

IOWA

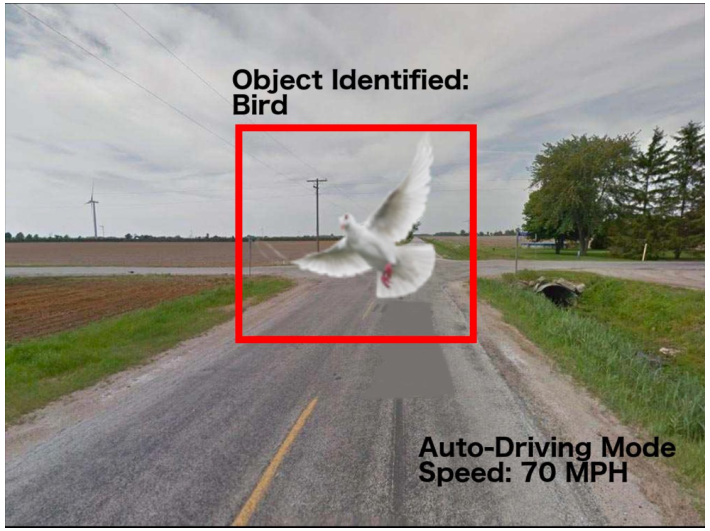
[1] Design of Low-Cost Reliable and Fault-Tolerant 32-Bit One Instruction Core for Multi-Core Systems

Consequences of Error Propagation in DNNs

- Single-bit fault[2] → Misclassification of image



Fault-free prediction label: Truck



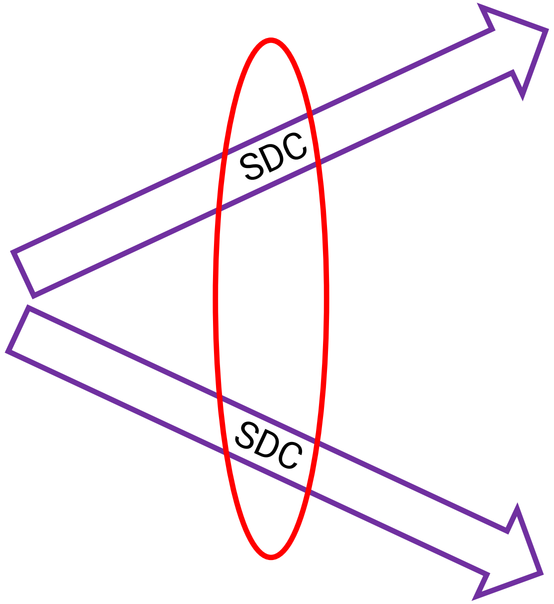
Faulty predicted label: Bird

- Reliability assessment: hardware vs software level
  - Software implemented fault injection (FI) simulation has lower cost

[2] *Understanding Error Propagation in Deep Learning Neural Network (DNN) Accelerators and Applications, SC'17*

Previous Works Only Consider SDC

Bus



Cab

Albatross

IOWA

[2] *Understanding Error Propagation in Deep Learning Neural Network (DNN) Accelerators and Applications, SC'17*

Not All Misclassifications Are Equal

Similar action for AVs

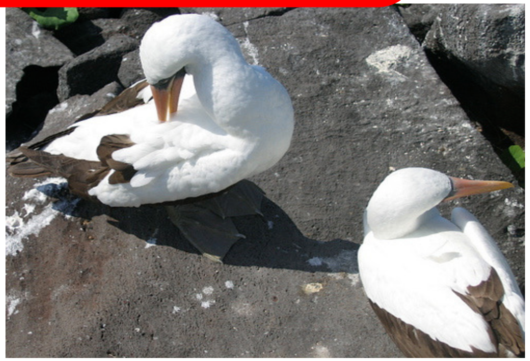
lassification

Bus

Need a different metric to differentiate safety critical misclassifications from non-safety critical ones

Misclassification

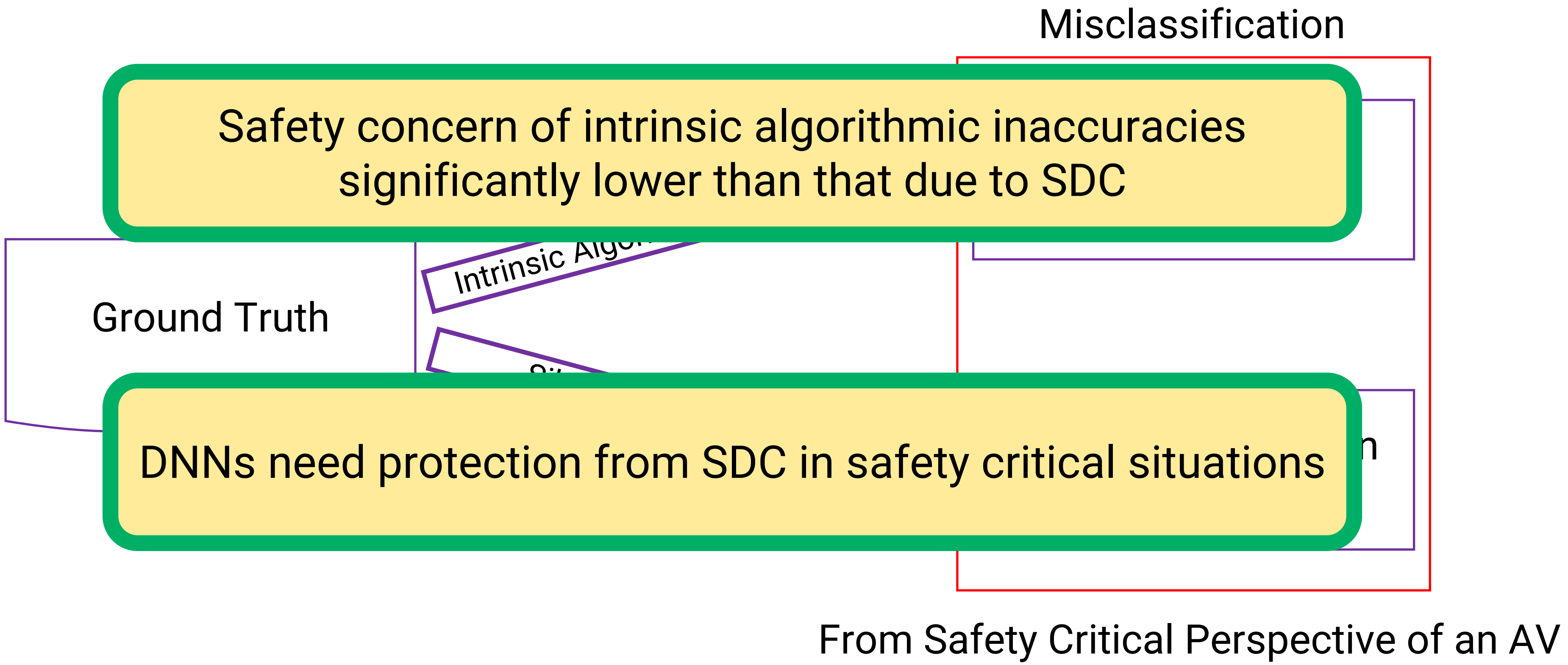
Different action for AVs



Albatross

IOWA

Our Hypothesis



Safety concern of intrinsic algorithmic inaccuracies significantly lower than that due to SDC

DNNs need protection from SDC in safety critical situations

Existing DNN Reliability Measurement Tools

TensorFI[3]

- A fault injector for TensorFlow applications
- Specifically, for TensorFlow 1 applications

TensorFI 2[4]

- A fault injector for
- This only supports sequential models

Need Support to inject faults in non-sequential DNN models with TensorFlow 2

Most DNN models are non-sequential

- Sequential: VGG16, VGG19
- Non-Sequential: ResNet50, ResNet101, GoogleNet, Xception, DenseNet121, DeseNet169, MobileNet

IOWA

[3] Tensorfi: A configurable fault injector for tensorflow applications, ISSREW'18
[4] https://github.com/DependableSystemsLab/TensorFI2

Our Contributions

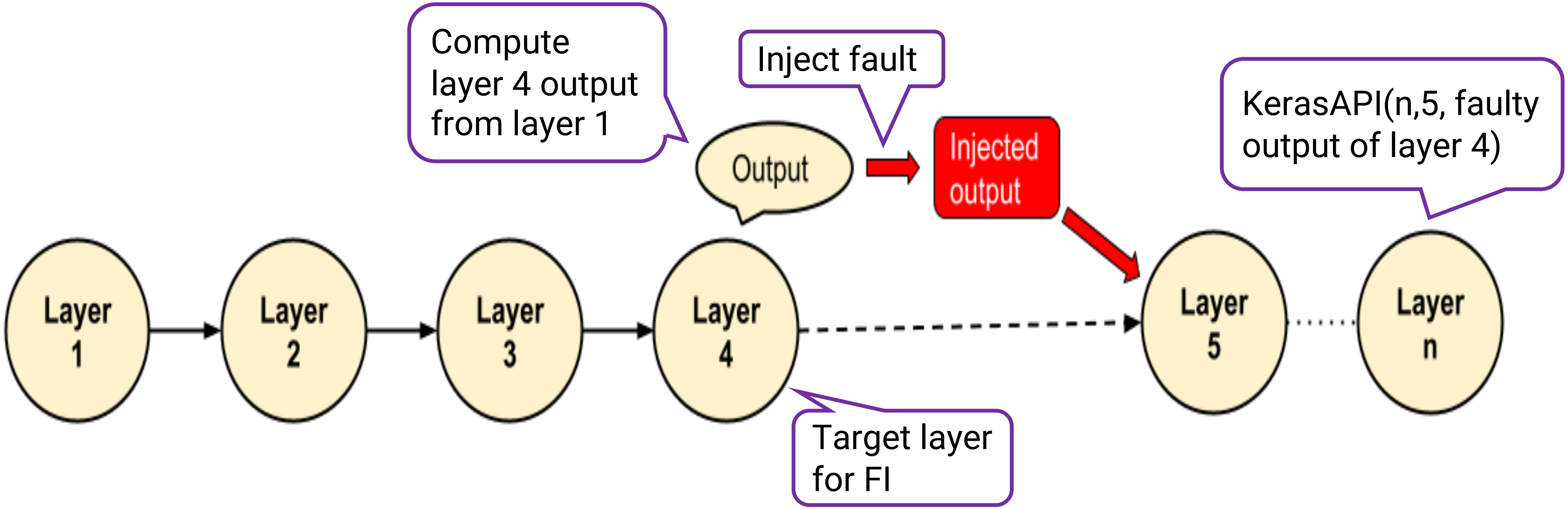
- Developed open-source tool, TensorFI+, to support FI in non-sequential DNN models
- Proposed new metrics to differentiate safety critical misclassifications from the perspective of AVs
- Analyzed why DNNs need protection from SDC in safety critical situations

TensorFI+ Development

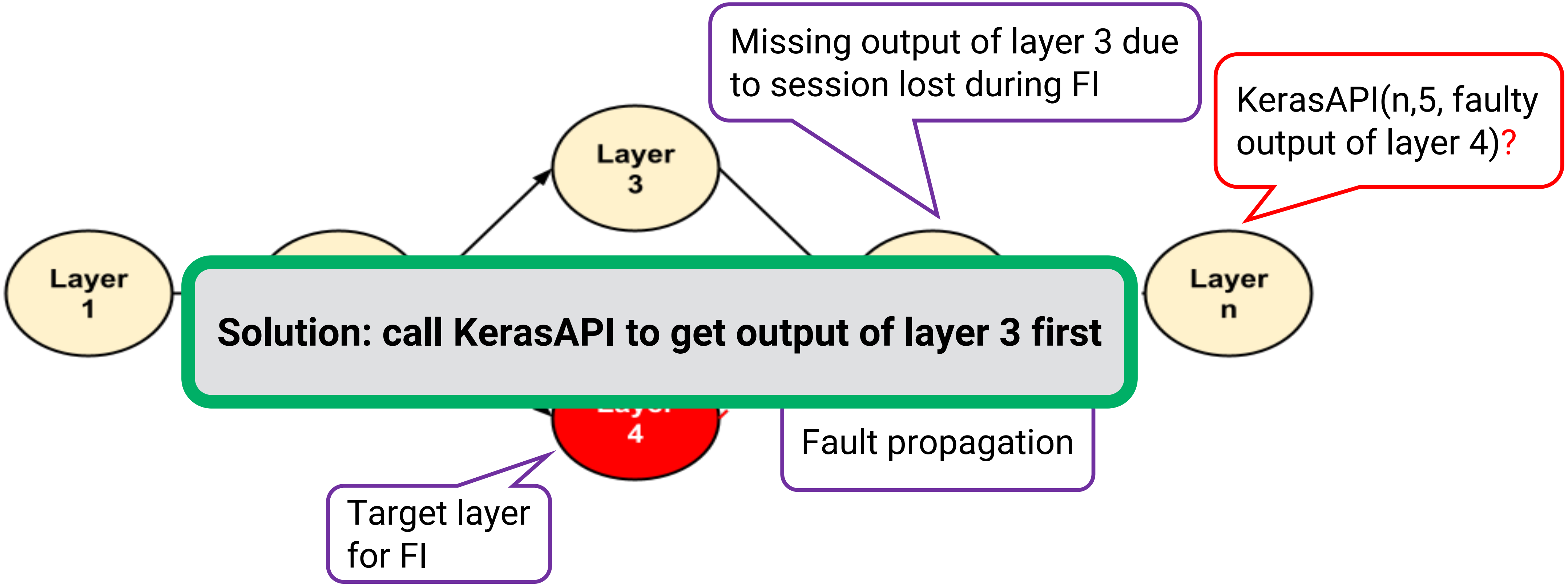
Keras Execution Flow Changes with TensorFI+

- Operators' structure changes in TensorFlow 2 are not allowed
- Need Keras API for fault injection and propagation
  - Output (layer D) = KerasAPI(Destination layer D, Source layer S, Input values of S)
  - KerasAPI call to get output of target layer t
  - Random bit fip of output of layer t
  - Previous session gone, need API calls to propagate faulty output to final layer

FI in a Sequential Model

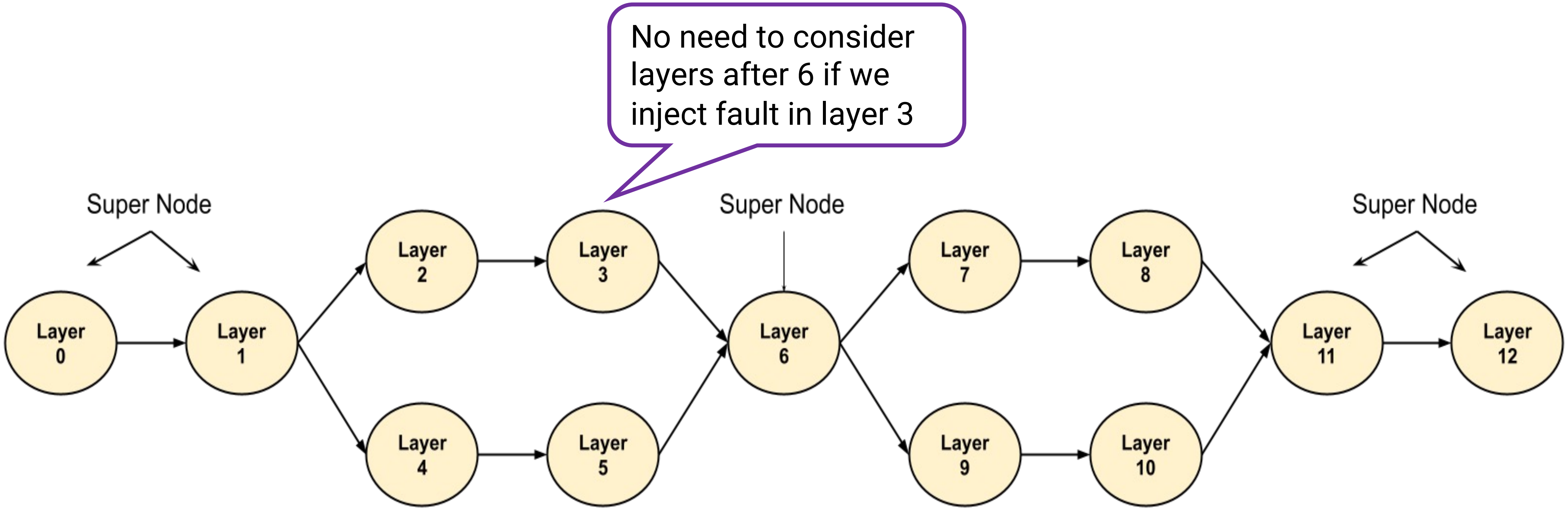


Issues in FI in Non-sequential Model



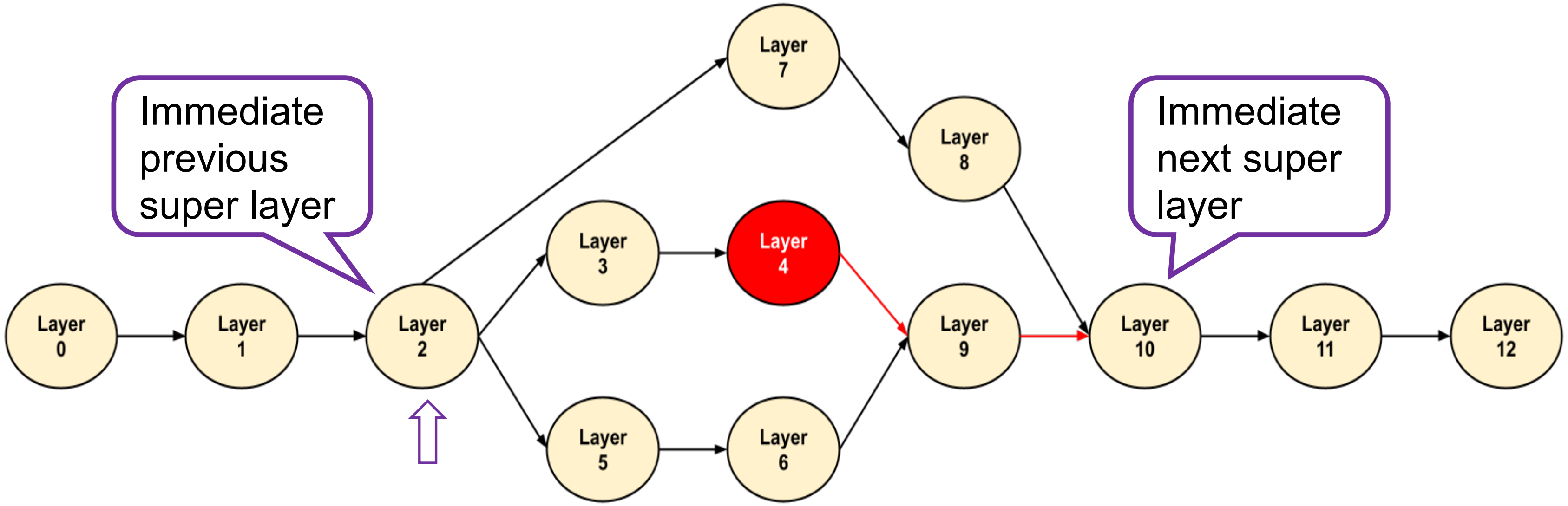
Solution: call KerasAPI to get output of layer 3 first

Solution: Super Layer



- Super layers are not part of any branch
- Any layer after a super layer is not dependent on any layer prior to super layer

Simulation of FI with TensorFI+

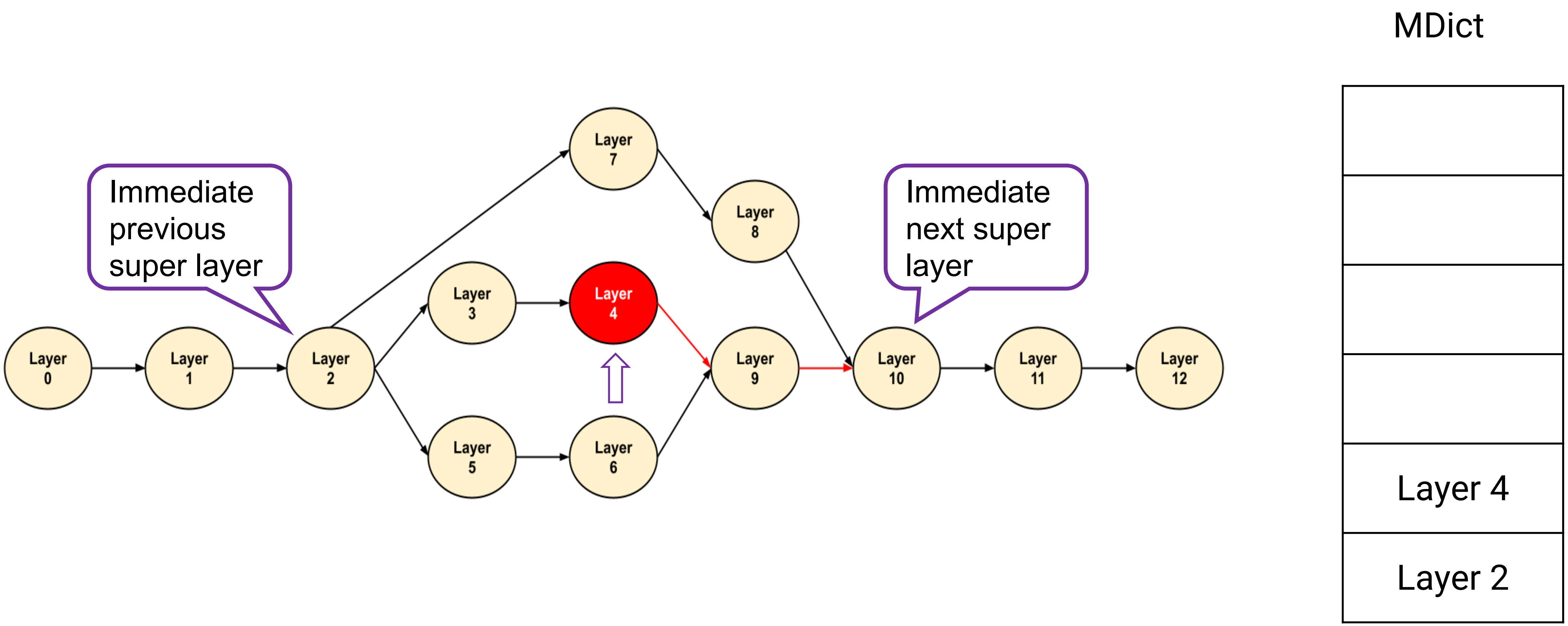


MDict

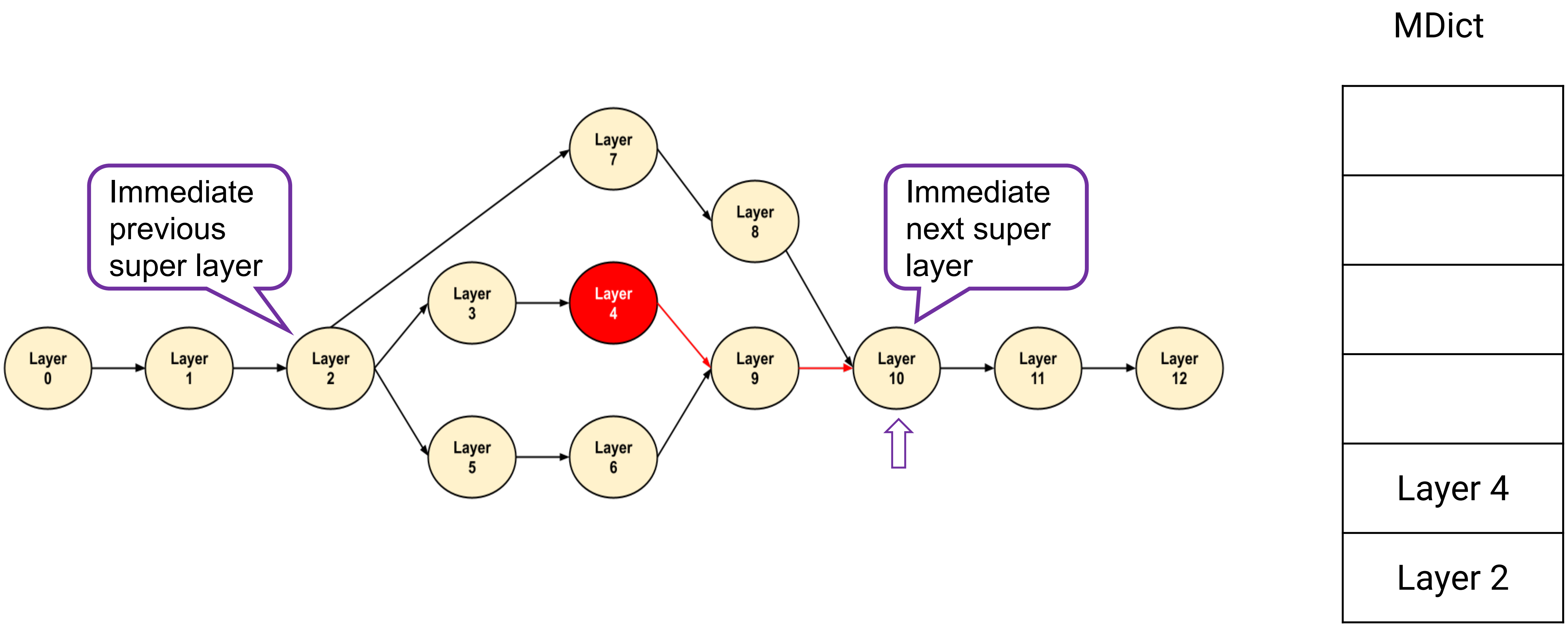
Layer 2

IOWA

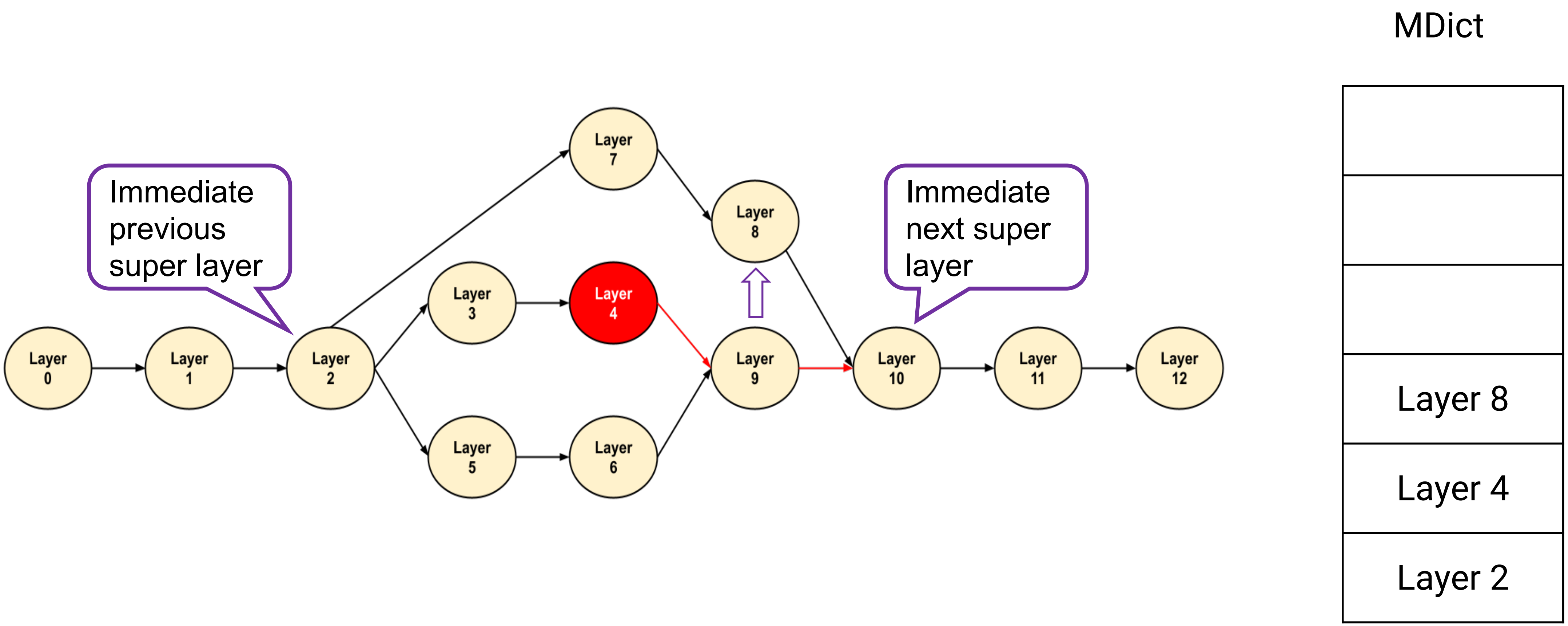
Simulation of FI with TensorFI+



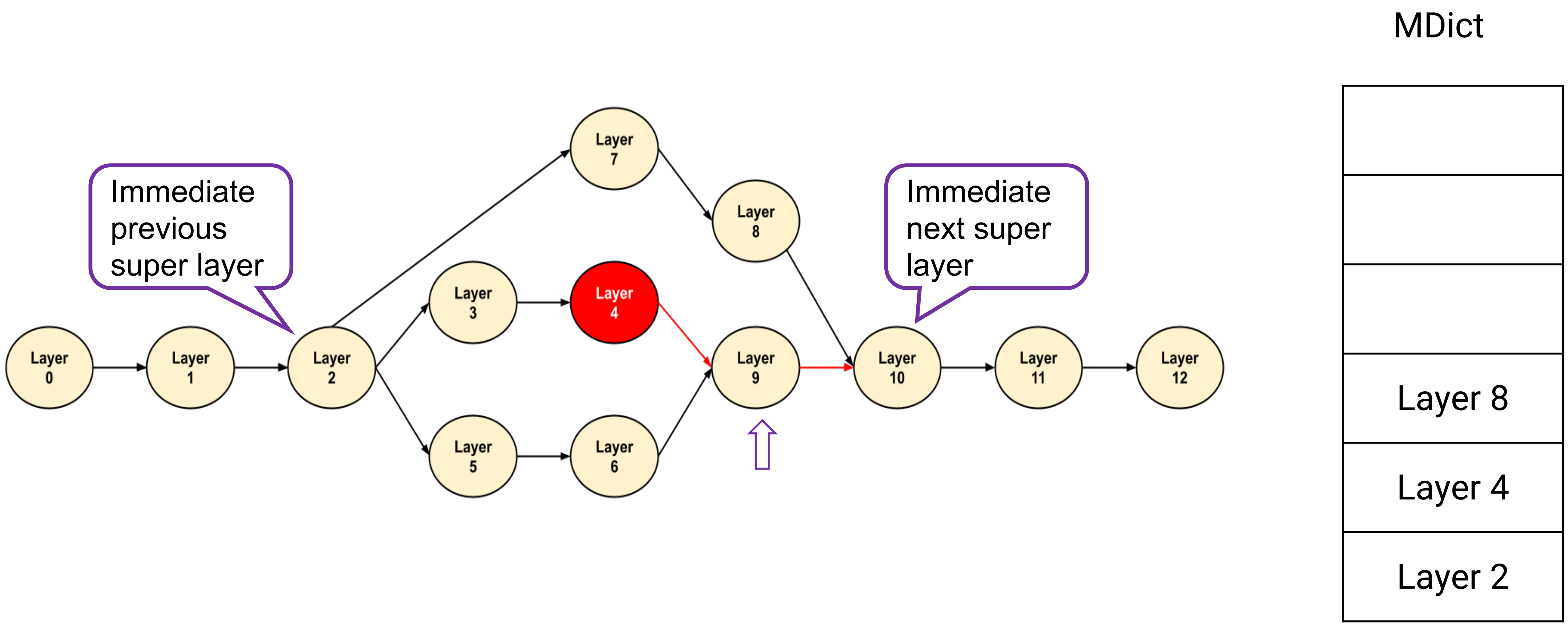
Simulation of FI technique of TensorFI+



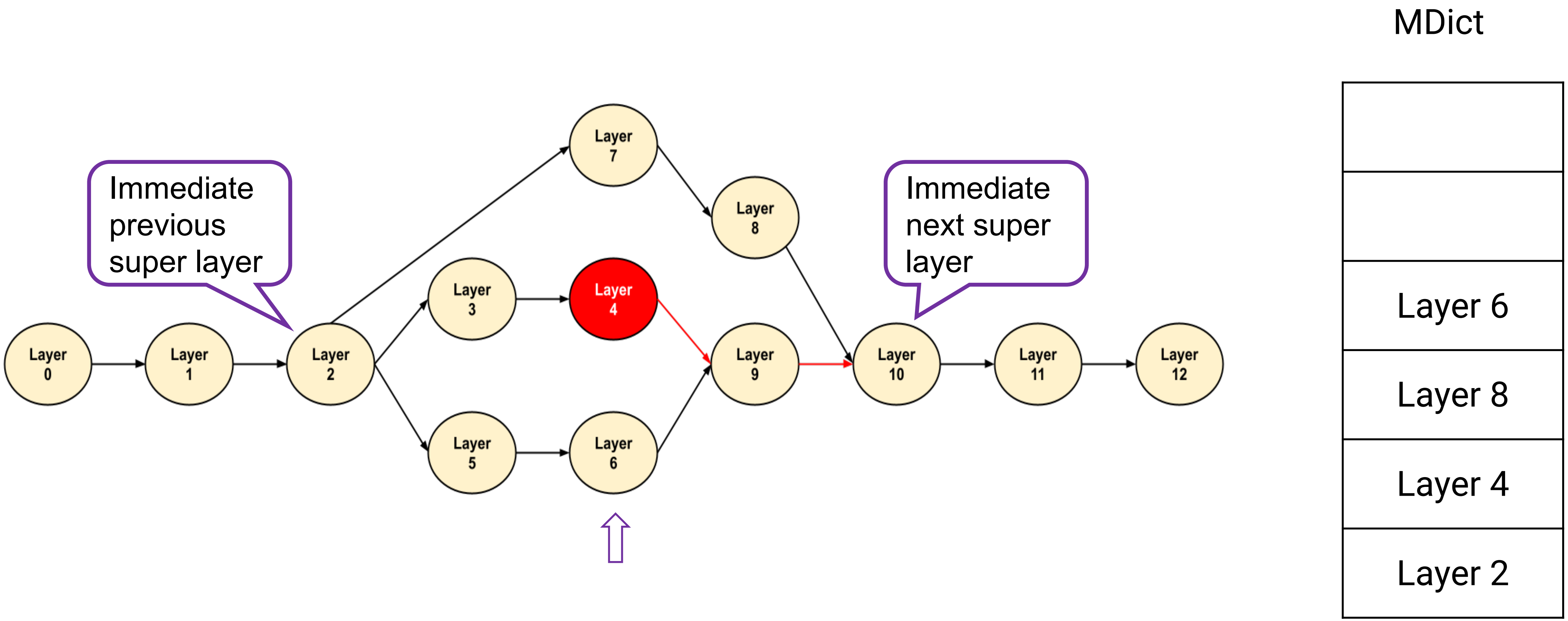
Simulation of FI technique of TensorFI+



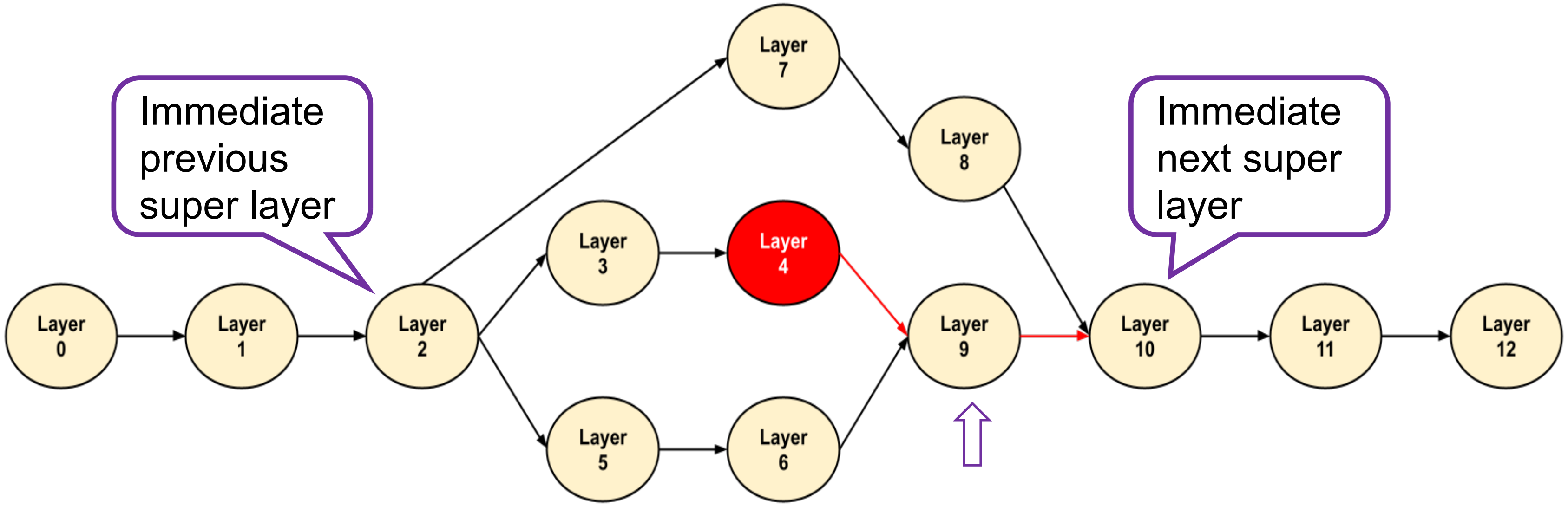
Simulation of FI technique of TensorFI+



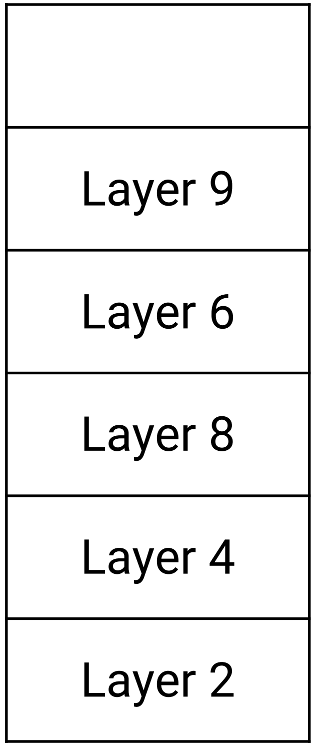
Simulation of FI technique of TensorFI+



Simulation of FI technique of TensorFI+



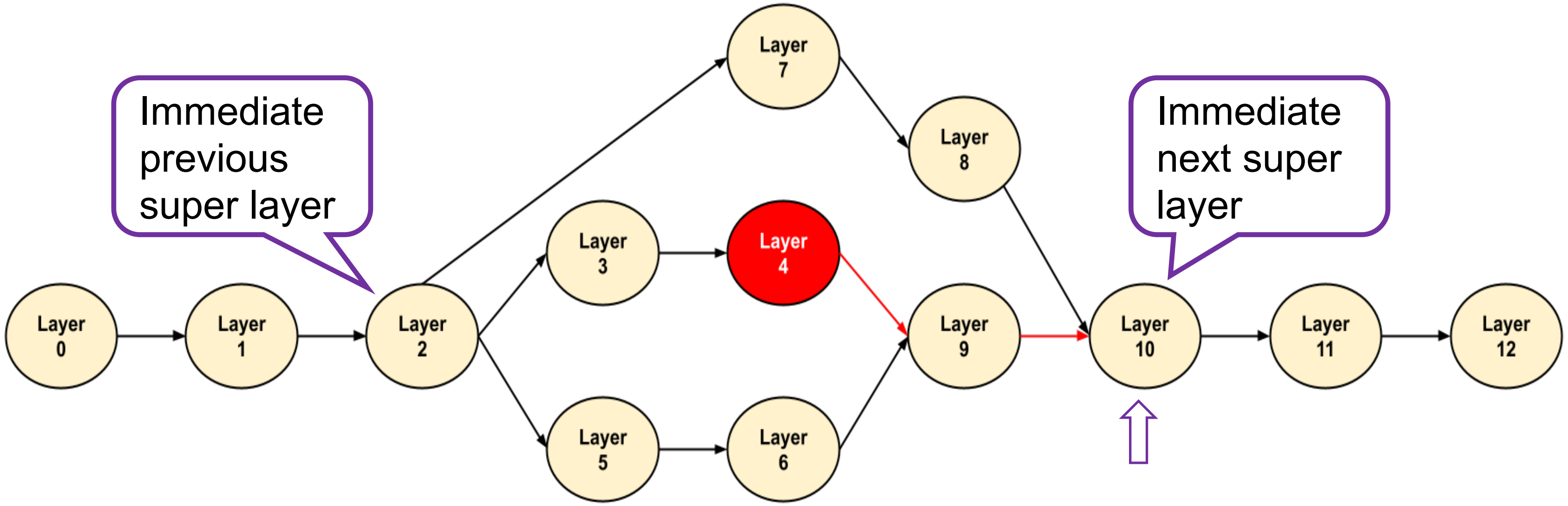
MDict



Simulation of TensorFI+

MDict

Layer 10
Layer 9
Layer 6
Layer 8
Layer 4
Layer 2



👉 Finally Compute the output of layer 12 using only one KerasAPI(12, 10, inputs(10)) call

Metrics to Differentiate Safety Critical Misclassification

Overview: Steps to Define Our Metrics

- Create several groups based on similarity of objects
- Organize all the groups into two supergroups based on safety concern
- Define two metrics to measure whether a misclassification is safety critical or not.

Group Formation

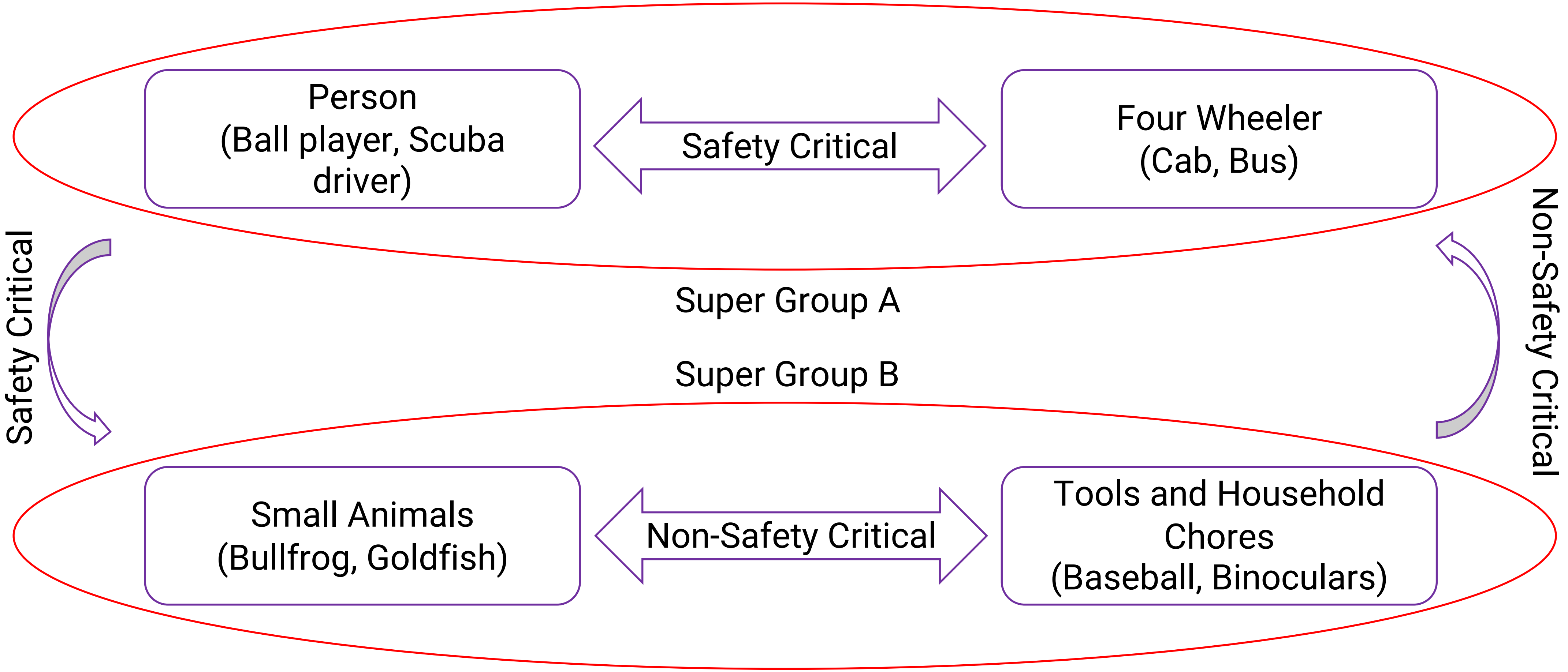
Objects:

- Ball player
- Scuba driver
- Cab
- Bus
- Bullfrog
- Goldfish
- Baseball
- Binoculars

Groups:

- Person
- Four Wheeler
- Small Animals
- Tools and Household Chores

Organize Groups into Two Supergroups



Metrics: SCM and Non-SCM Probability

- Safety Critical Misclassification(SCM) Probability
  - Original label is in Supergroup A and the predicted label is in Supergroup B
  - They are from different groups within Supergroup A
- Non-Safety Critical Misclassification(Non-SCM) Probability
  - Non-SCM probability complements to SCM probability
  - They add up to 100%.

Benchmark & Experimental Setup

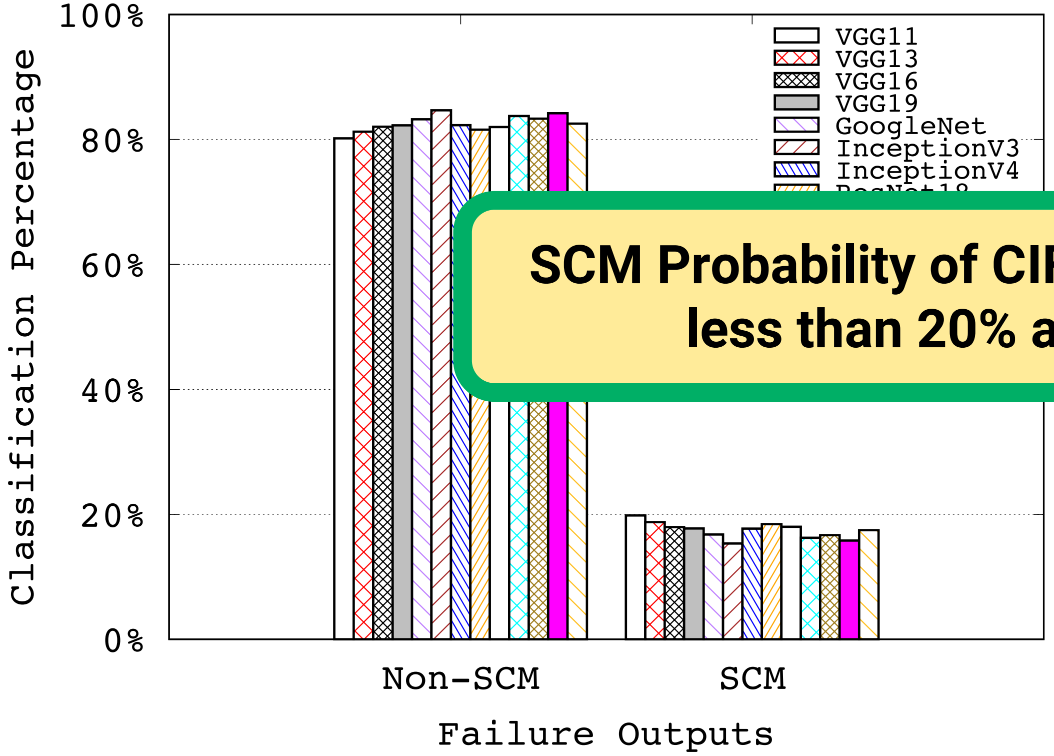
- Demonstrated on 30 popular DNN models
  - VGGNets, ResNets, DenseNets
- 2 open-source widely used datasets
  - CIFAR-100, ImageNet
- 3000 random fault injections per DNN model
- Measured SDC, SCM and Non-SCM probability in the evaluation

Results: SDC rates

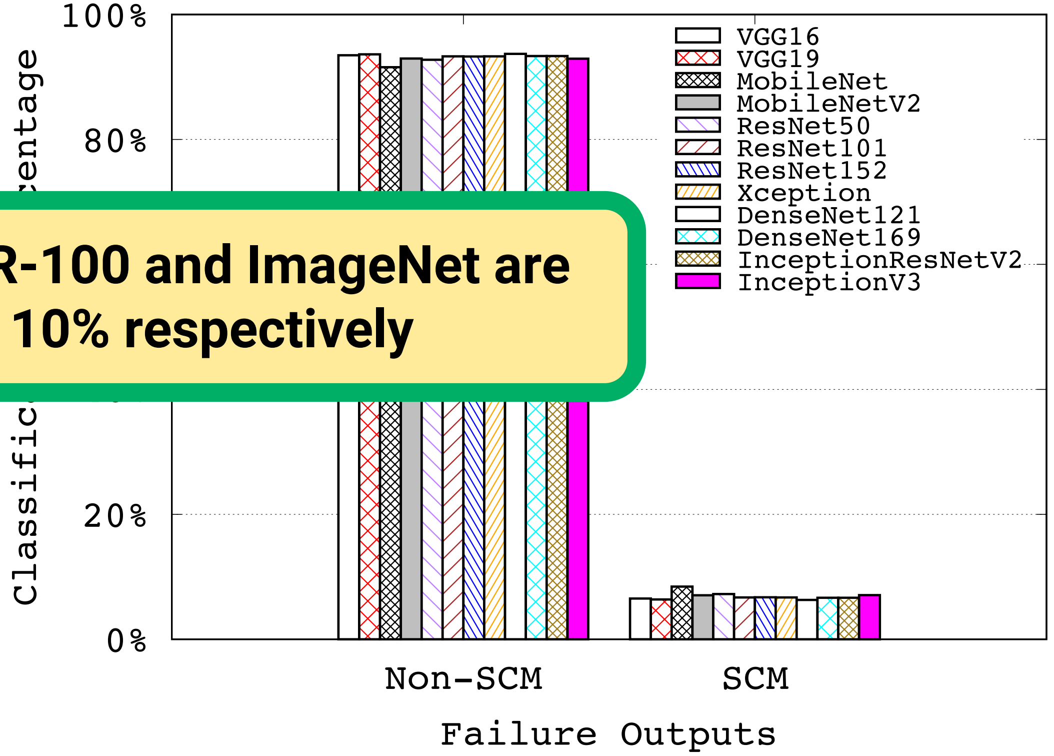
Dataset	Model	Top-1 Accuracy	SDC Rate
ImageNet	VG16(Sequential)	71.18%	3.53%
	ResNet50(Non-sequential)	74.76%	1.43%
	DenseNet121(Non-sequential)	75.04%	1.20%
CIFAR-100	VGG19(Sequential)	71.53%	1.23%
	GoogleNet(Non-sequential)	76.70%	1.57%
	Xception(Non-sequential)	77.96%	2.00%

SDC rates range from 0.53% to 2.07% (error bars range from 0.10% to 2.95%) across different non-sequential DNN models

Results: Fault Free Inference



CIFAR-100

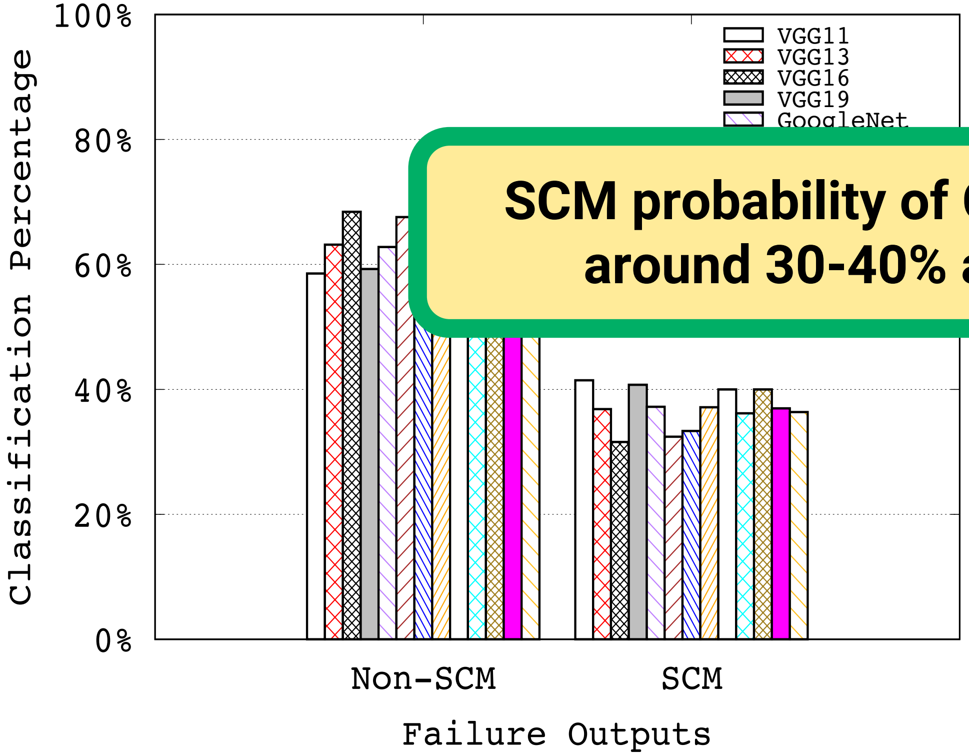


ImageNet

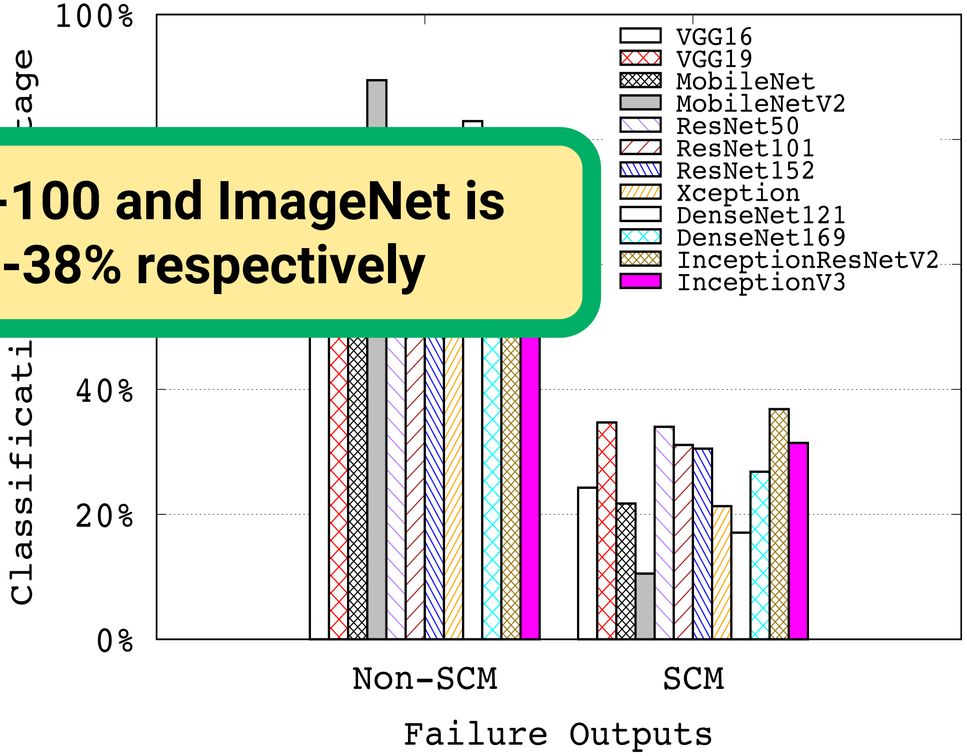
SCM Probability of CIFAR-100 and ImageNet are less than 20% and 10% respectively

IOWA

Results: FI in Correctly Classified Images



CIFAR-100

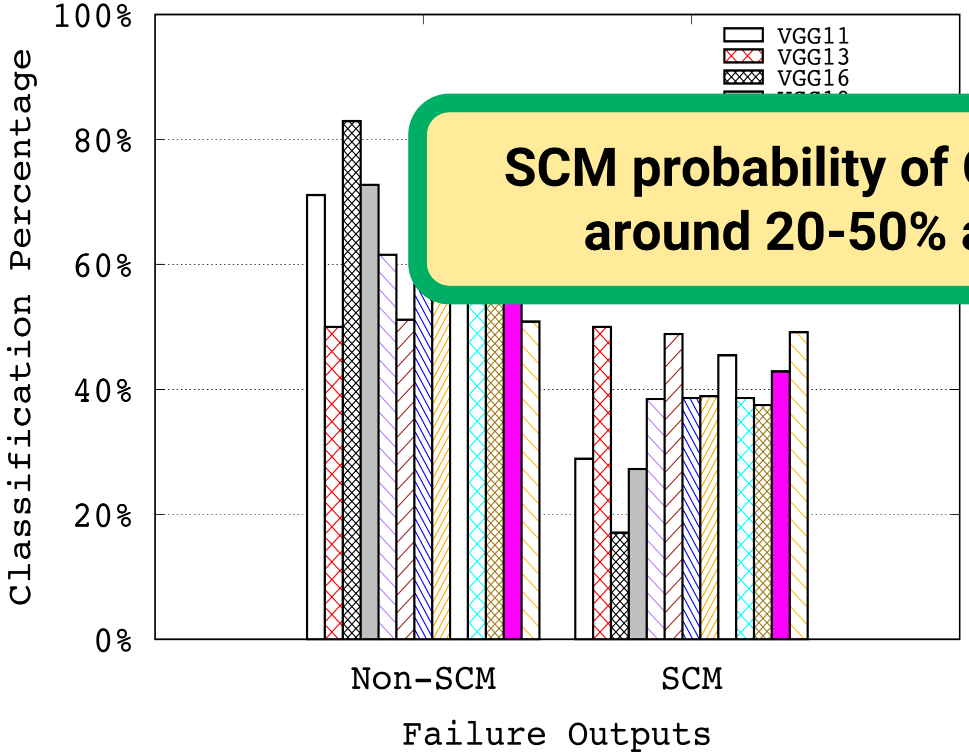


ImageNet

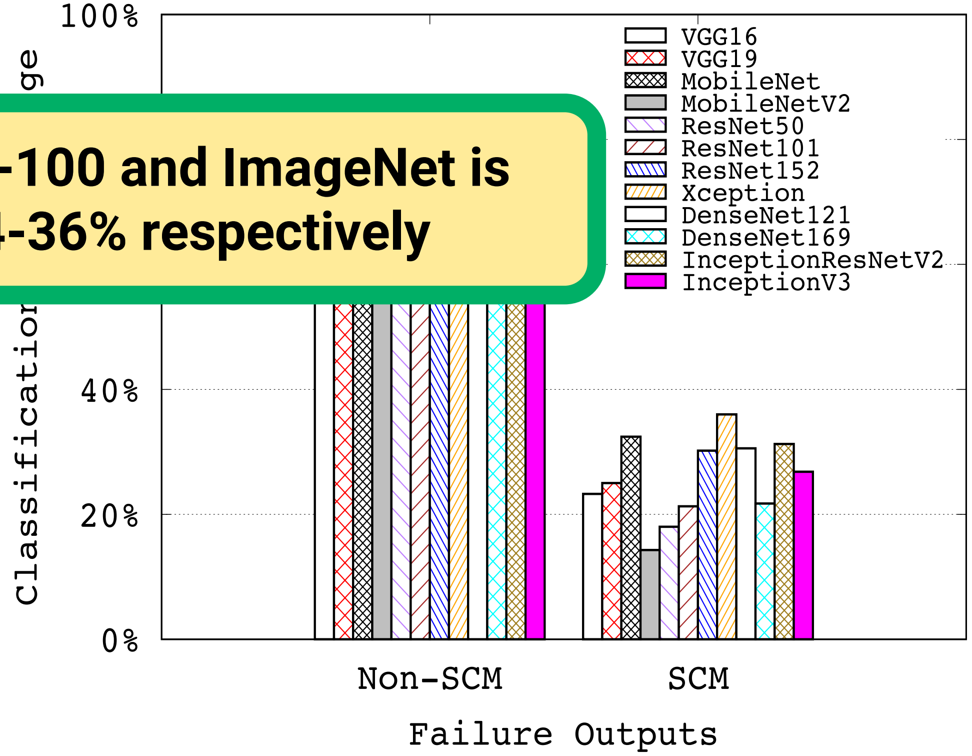
SCM probability of CIFAR-100 and ImageNet is around 30-40% and 10-38% respectively

IOWA

Results: FI in Misclassified Images



CIFAR-100



ImageNet

SCM probability of CIFAR-100 and ImageNet is around 20-50% and 14-36% respectively

Conclusion

- Built a FI tool, TensorFI+, for both sequential and non-sequential DNN resilience evaluation
- We introduce two new metrics to differentiate safety critical misclassifications.
- SCM probability is much higher with FI compared to fault free inference
  - Shows the necessity of protecting DNN models from SDC.
- Our code is open source at https://github.com/sabuj7177/characterizing_DNN_failures

Sabuj Laskar
University of Iowa

IOWA