ProGIP: Protecting Gradient-based Input Perturbation Approaches for Out-of-distribution Detection From Soft Errors

Speaker: Sumedh Shridhar Joshi

Chair: Dr. Aviral Shrivastava

Members: Dr. Aman Arora

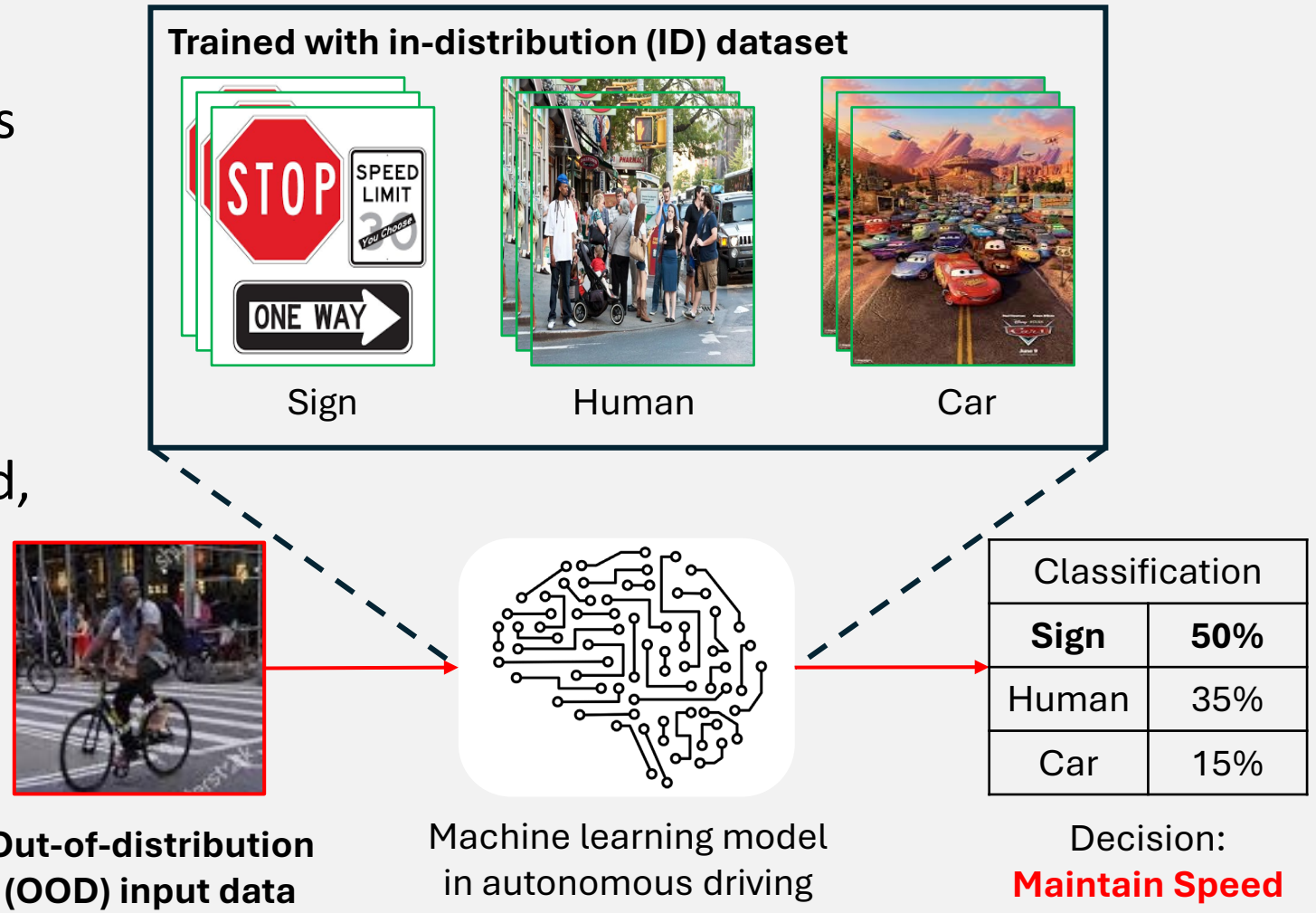
Dr. Hokeun Kim

Introduction

- Introduction
- Background
- Related Works
- ProGIP: Observations
- ProGIP: Methodology
- Experimental Setup
- Experimental Results
- Conclusion

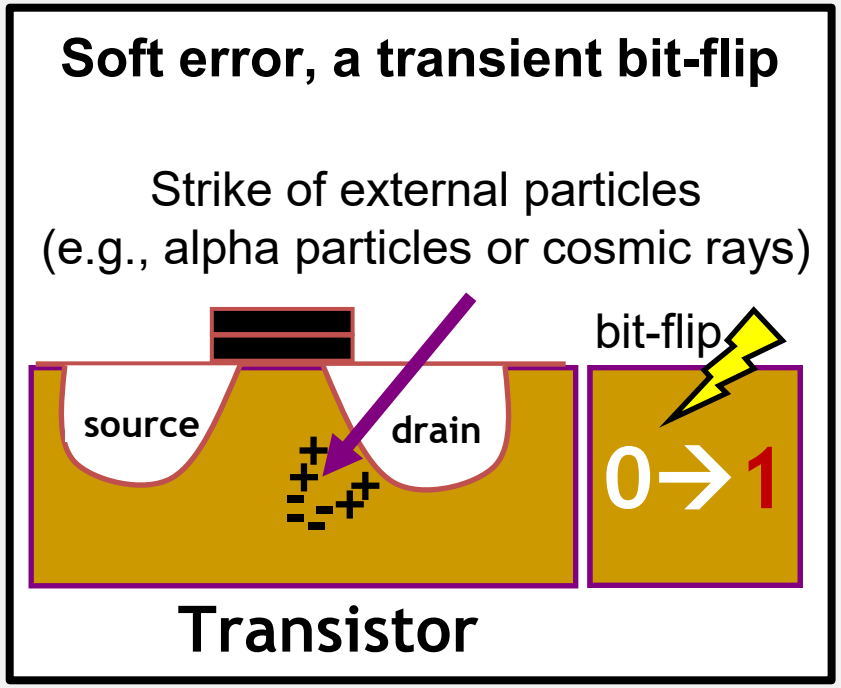
What is OOD and why is OOD Detection Important?

- OOD data is often referred to as “unknown" data.
- Since a model is not intended for the unknown inputs, it is important to detect OODs.
- The input must not be classified, and the control should be transferred to human.

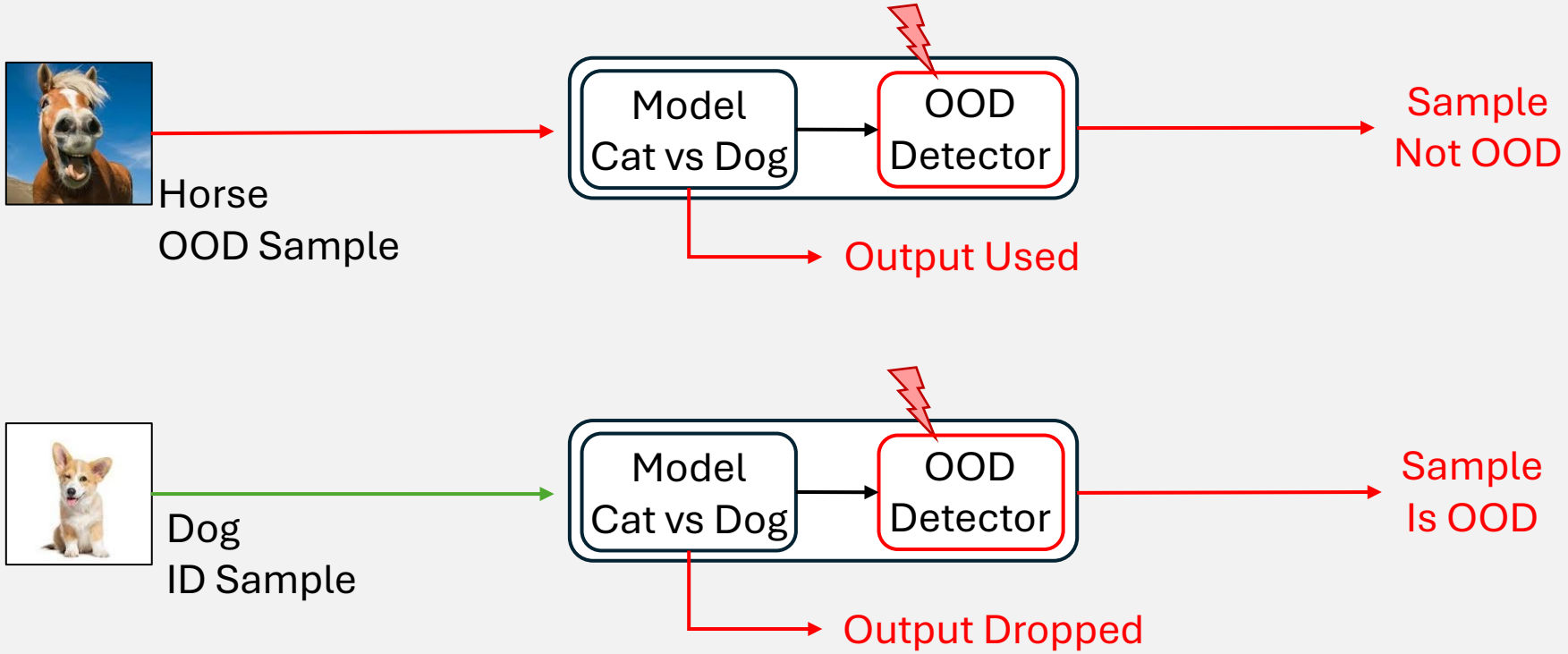


Soft Errors and their Effect on OOD Detection

- A soft error is a transient fault resulting in a temporary bit-flip error of the transistor induced by external sources such as alpha particles, thermal neutrons, or cosmic rays.
- Soft errors can cause misclassification which is dangerous in a safety-critical system.
- Soft errors can lead to incorrect identification of in-distribution (ID) samples as OOD samples or vice versa.



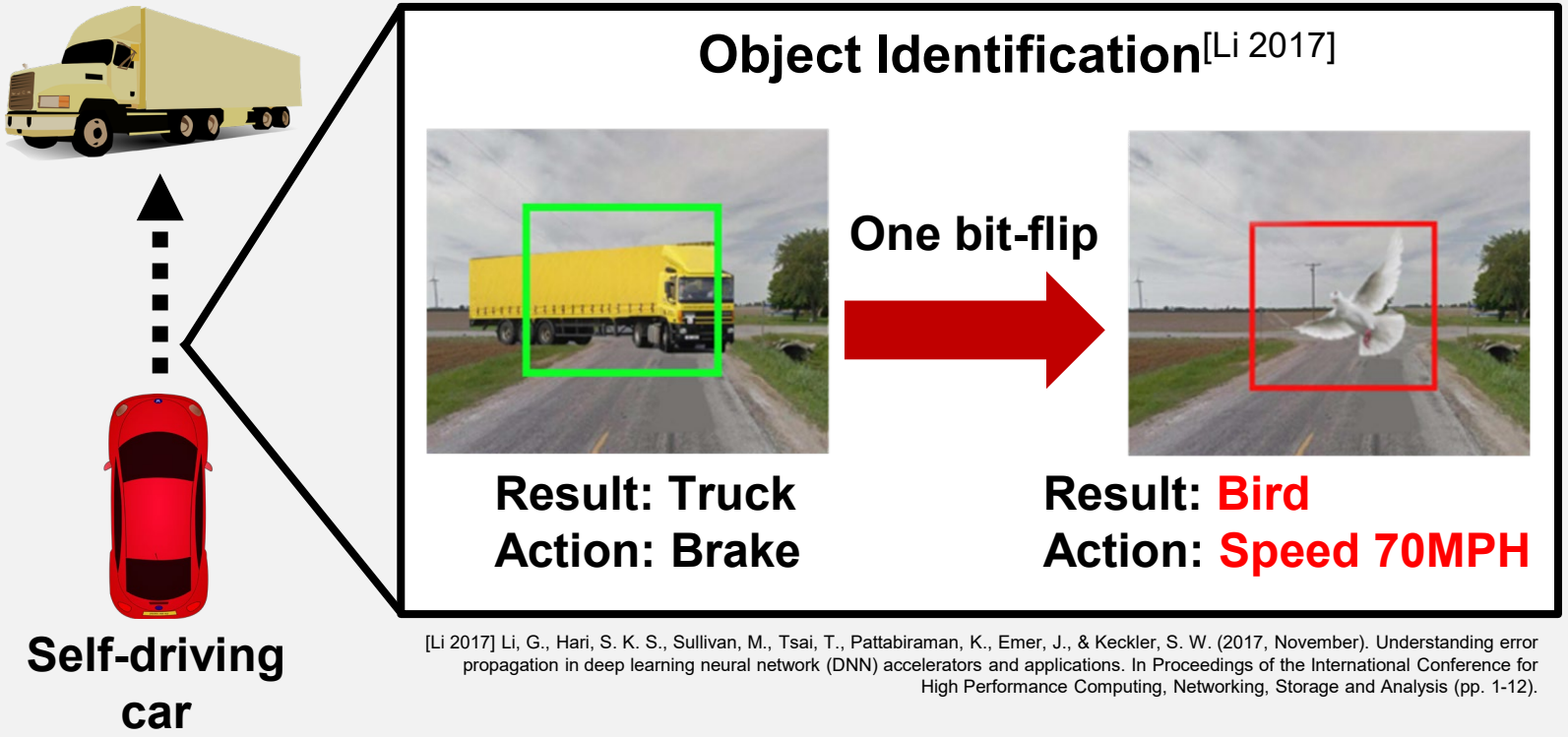
Soft Errors Cause ID/OOD Detection Failure



Soft errors can lead to incorrect identification of ID samples as OOD samples or vice versa.

Soft Errors Cause Classification Failure

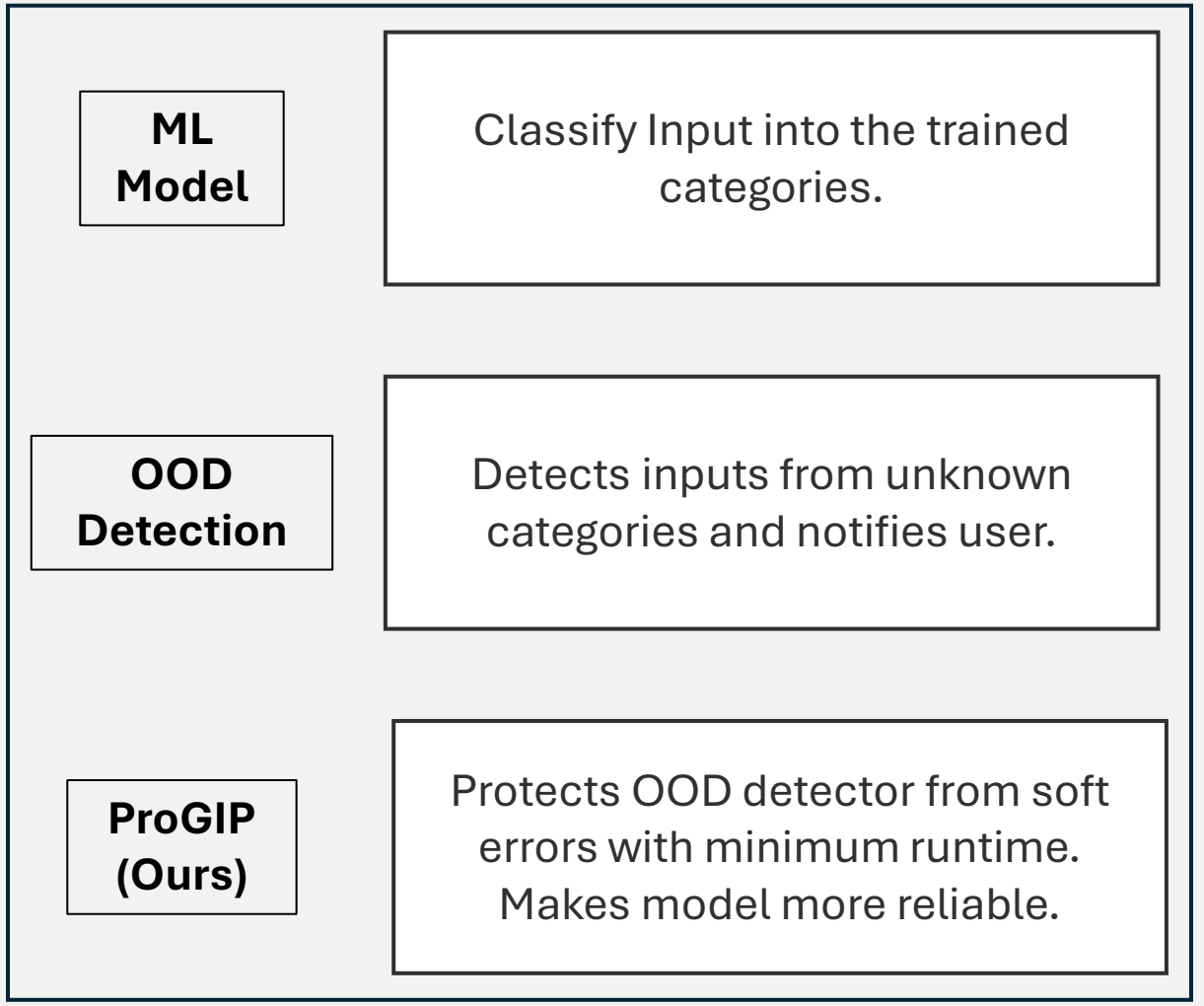
Soft errors can cause misclassification.



[Li 2017] Li, G., Hari, S. K. S., Sullivan, M., Tsai, T., Pattabiraman, K., Emer, J., & Keckler, S. W. (2017, November). Understanding error propagation in deep learning neural network (DNN) accelerators and applications. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-12).

Problem Statement

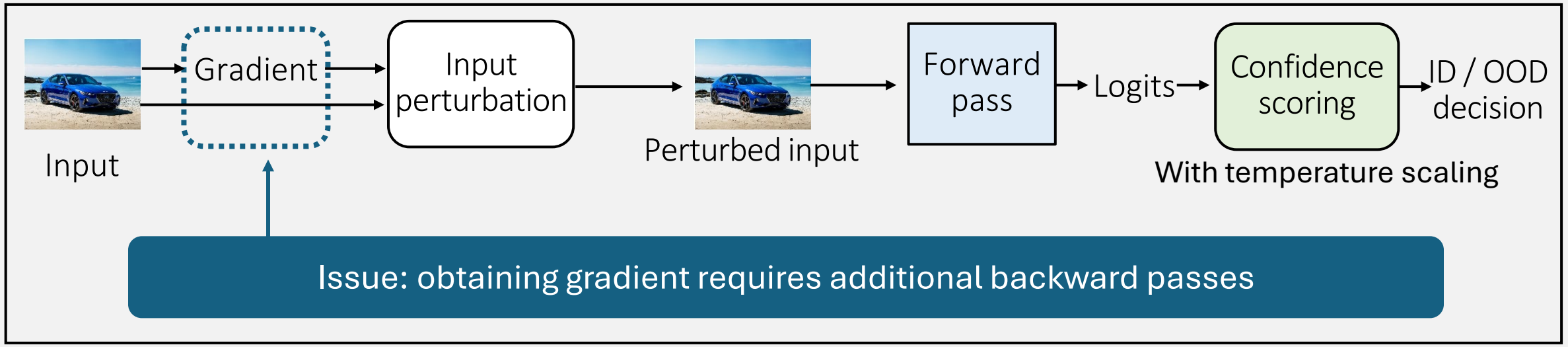
- **Detect soft errors** in systems that detect OODs to improve holistic reliability of the system.
- Differentiate between soft error and OOD to better the deployment of models.
- Soft errors are not very commonly occurring so the method to detect them should be light-weight.



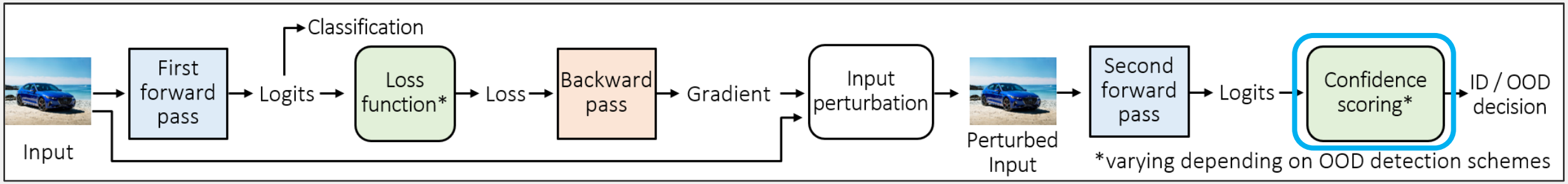
Background

- Introduction
- Background
- Related Works
- ProGIP: Observations
- ProGIP: Methodology
- Experimental Setup
- Experimental Results
- Conclusion

Gradient Based Input Perturbation OOD Detection: Overview



ODIN (Liang et. al.)



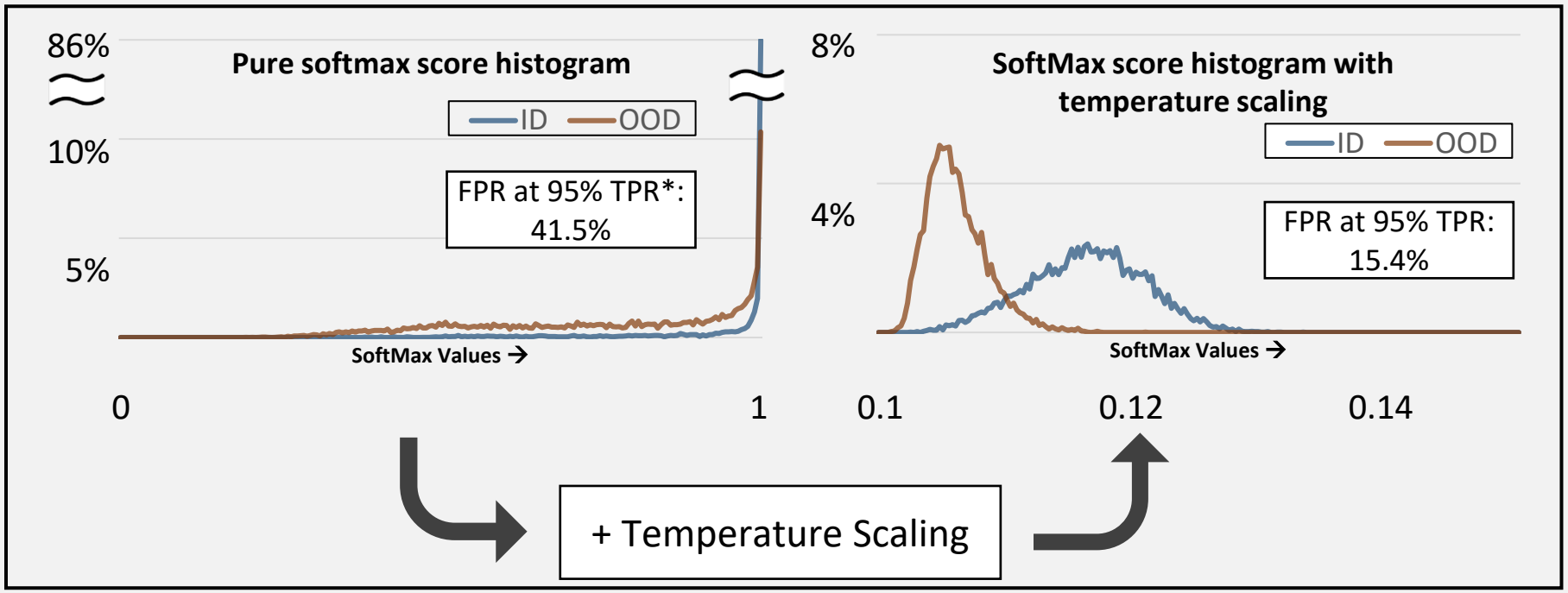
- Distinguish between ID and OOD using the perturbed input & temperature scaled SoftMax output.
- $S(xi;T) = \frac{e^{f(xi/T)}}{\sum_{j=1}^{N} e^{f(xj/T)}}$

Where:

- x: test sample,
- xi : max logit from network,
- N : total classes,
- T : Temperature (hyperparameter),
- $f(x)$: pre-trained features of the SoftMax based neural classifier.

Refine Model Calibration Using Temperature Scaling

- $S(x_i; T) = \frac{e^{f(xi/T)}}{\sum_{j=1}^{N} e^{f(xj/T)}}$



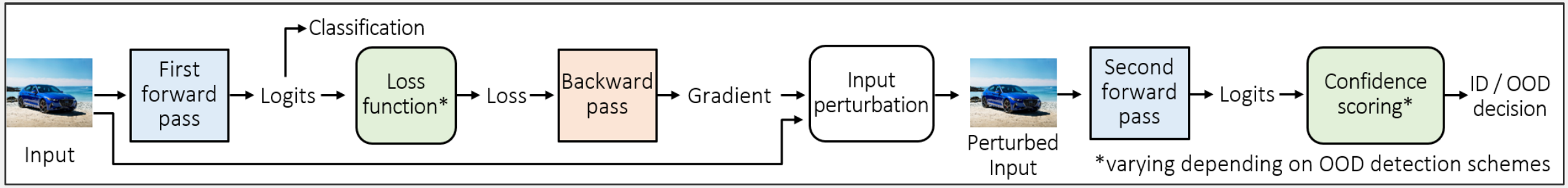
- Temperature Scaling calibrates the SoftMax outputs.
- Model Calibration: The process of adjusting a model's predicted probabilities to ensure they align with the true likelihood of outcomes.
- FPR (False Positive Rates) @ 95% TPR (True Positive Rates): The fraction of misclassified out-of-distribution samples, from 34.7% to 4.3%, when 95% of in-distribution images are correctly classified.

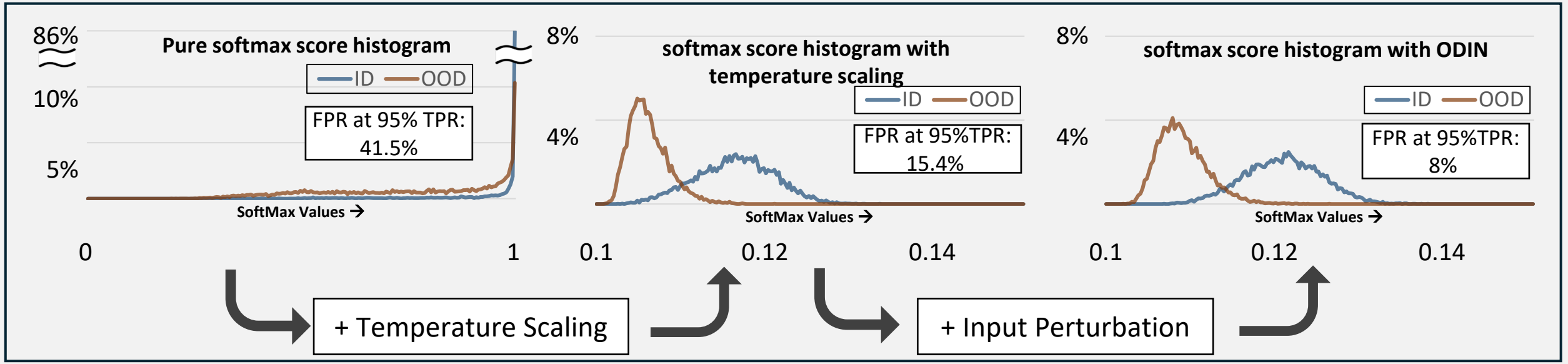
Input Perturbation for GIP based OOD Detection

$$\tilde{x} = x - \varepsilon * sign\left(-\nabla x \log S\left(x;\ T\right)\right)$$

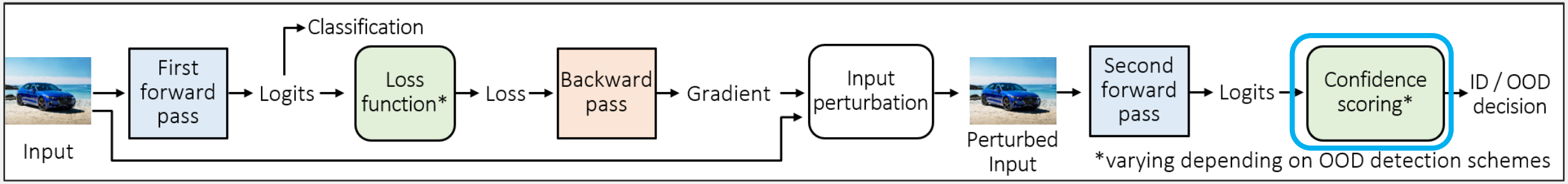
- Where:
  - x: Input
  - $\tilde{x}$: Perturbed Input
  - ε: Perturbation Magnitude
  - $\nabla x \log S\left(x;\ T\right)$: Gradient of scoring function w.r.t. sample input.
  - $S(x;\ T)$: Maximum SoftMax Probability
- The perturbation can have stronger effect on the in-distribution images than that on out-of-distribution images, making them more separable.

Impact after combining Temperature Scaling and Input Perturbation



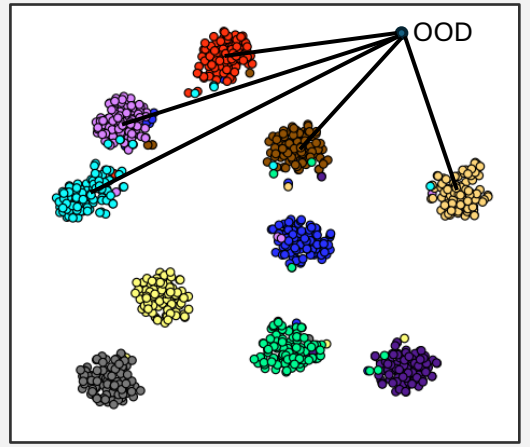


Mahalanobis (K. Lee, et. al.)



- Calculates the Mahalanobis distance between the logits generated with perturbed input and the mean of logits generated with ID inputs for each class. It utilizes the negative of the maximum distance value as a confidence score.

- $M(x) = \max_c \left\{ -\left((f(x) - \hat{\mu}_c)^T \hat{\Sigma}^{-1} (f(x) - \hat{\mu}_c) \right) \right\}$ where,
  - x : test sample,
  - c : the index of the closest class,
  - $\widehat{\mu c}$:the sampled mean of the class c,
  - $f(x)$: pre-trained features of the SoftMax based neural classifier,
  - $\hat{\Sigma}$: the covariance matrix.



K. Lee, et. al. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," Advances in neural information processing systems, vol. 31, 2018.

Related Works

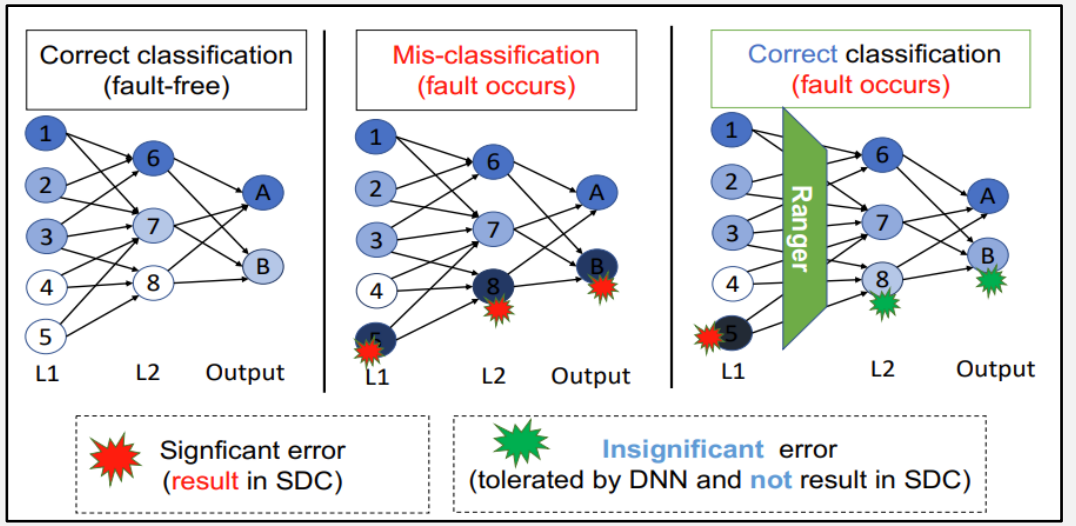
- Introduction
- Background
- Related Works
- ProGIP: Observations
- ProGIP: Methodology
- Experimental Setup
- Experimental Results
- Conclusion

Methods that Detect Soft Errors as well as OODs

- There is no direct method that address soft error and OOD detection problem in a single solution.
- Some soft error detection solutions can be extended on OOD detection flow.

Solutions Extended to Protect OOD Detection System from Soft Errors

- Ranger
  - Checks the abnormal values of all activation function layers and following max pool, average pool, and reshape layers, via built-in hook methods of PyTorch.
  - The Ranger to correct faults by changing the abnormal values.
- Open-Set Recognition: An Inexpensive Strategy to Increase DNN Reliability
  - Treats faults as anomaly.
- FACER: A Universal Framework for Detecting Anomalous Operation of Deep Neural Networks
  - Considers fault and OODs as anomaly.
  - Requires additional data for both fault as well as OOD detection.



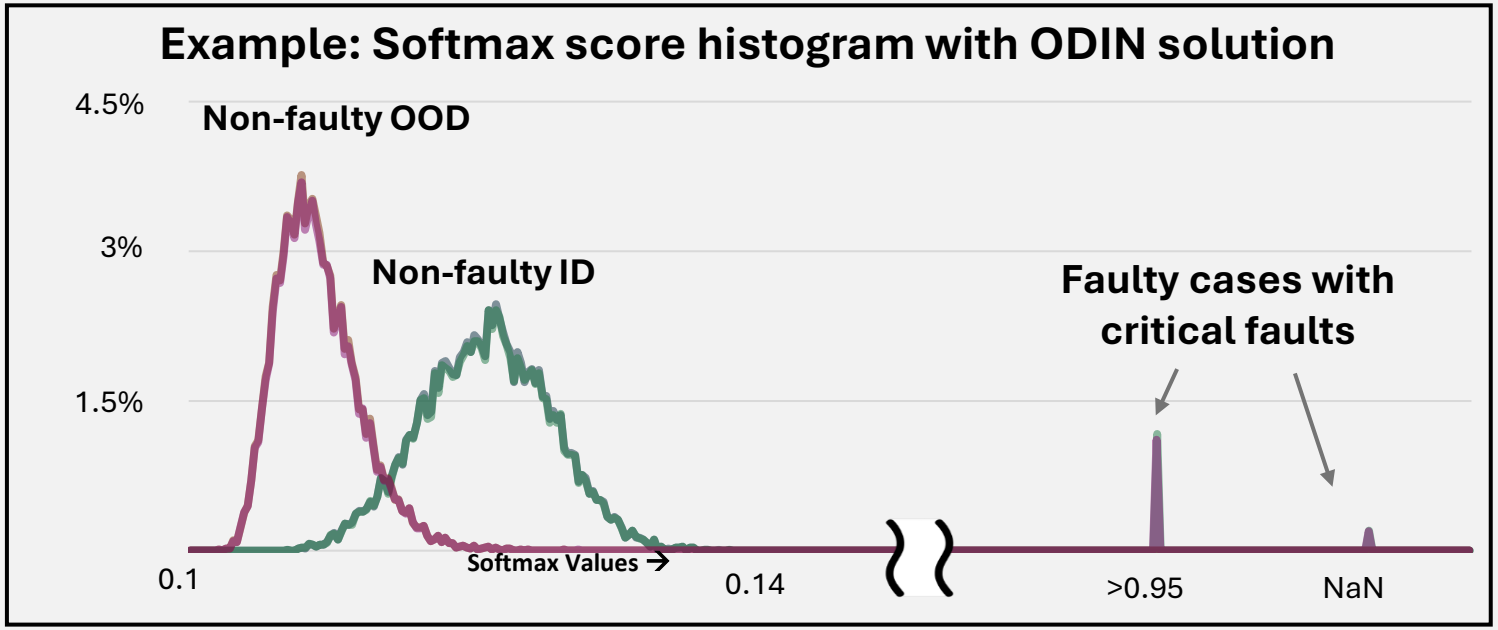
Z. Chen, et. al. "A low-cost fault corrector for deep neural networks through range restriction," 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Net-works (DSN). IEEE, pp. 1–13.

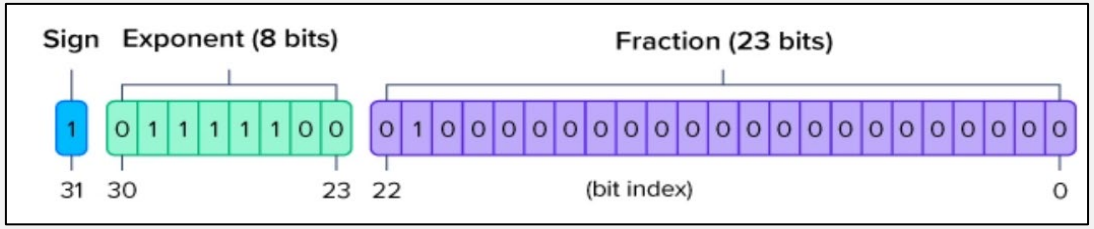
ProGIP: Observations

- Introduction
- Background
- Related Works
- ProGIP: Observations
- ProGIP: Methodology
- Experimental Setup
- Experimental Results
- Conclusion

Observing the Impact of Soft Errors on Classification & OOD Detection Outcomes

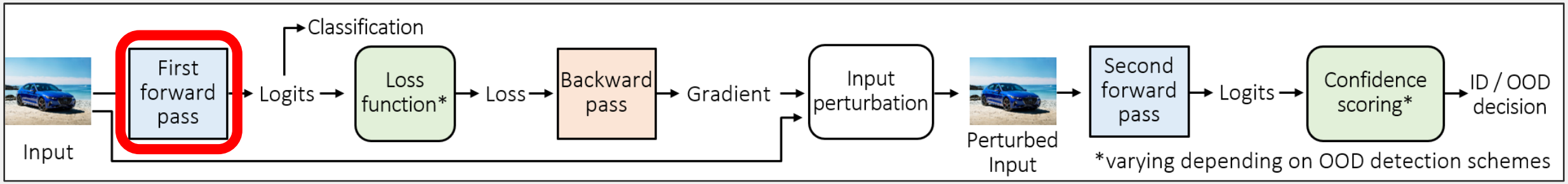
- A soft error in the model with GIP solution can change the result of **classification** or **ID/OOD detection**
- Faults on different passes show different symptoms effects.
- Faults on higher bits lead to critical faults.

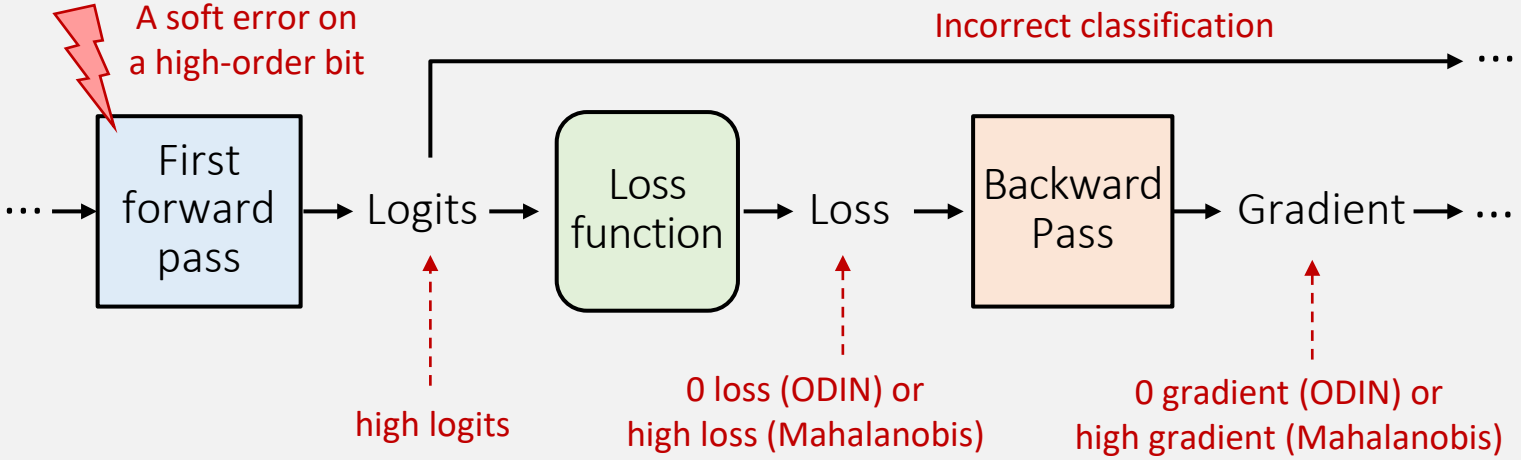




https://www.exxactcorp.com/blog/hpc/what-is-fp64-fp32-fp16

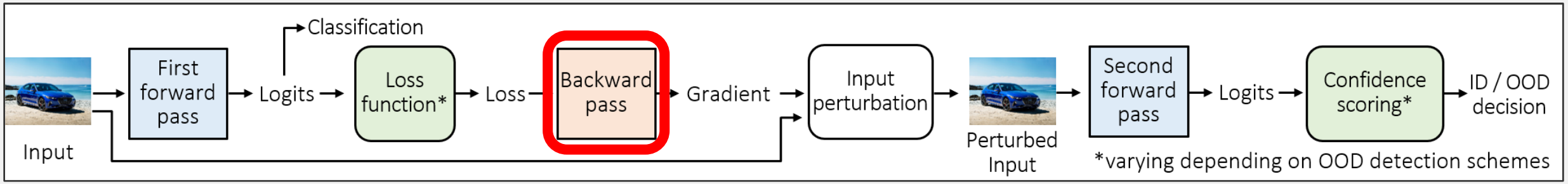
Signatures of Errors in the First Forward Pass After Fault Injection

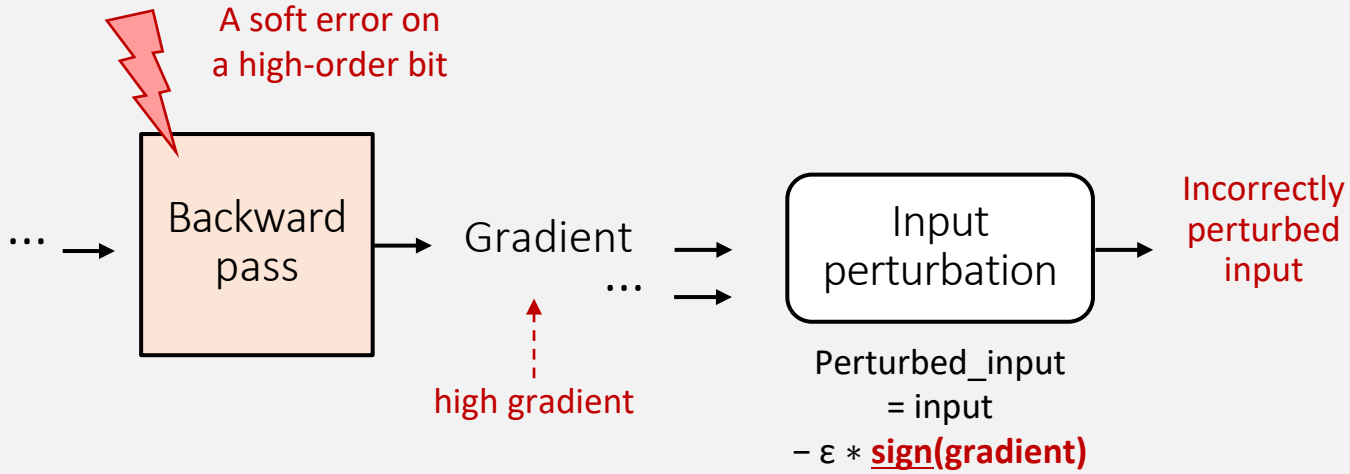




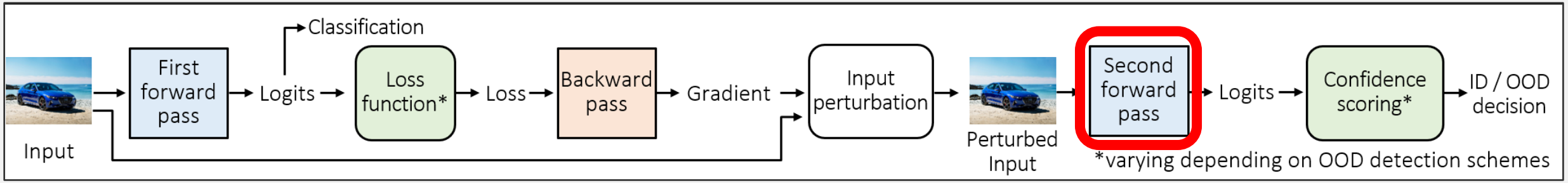
Errors in first forward pass leads to classification failure.

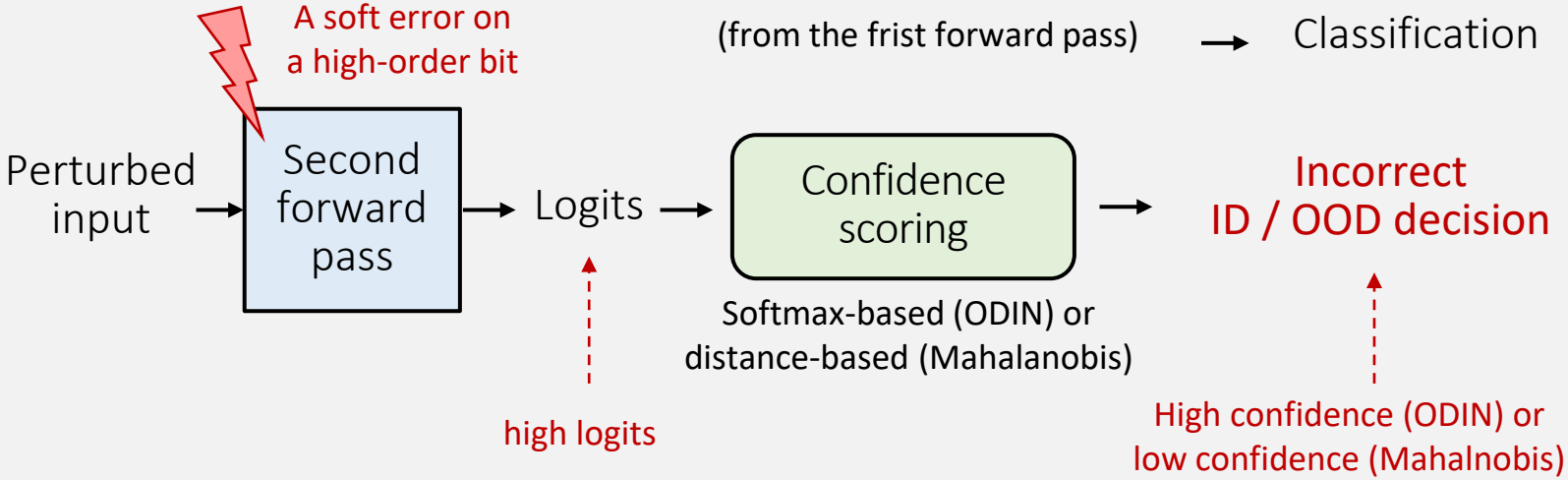
Signatures of Errors in the Backward Pass After Fault Injection





Signatures of Errors in the Second Forward Pass After Fault Injection



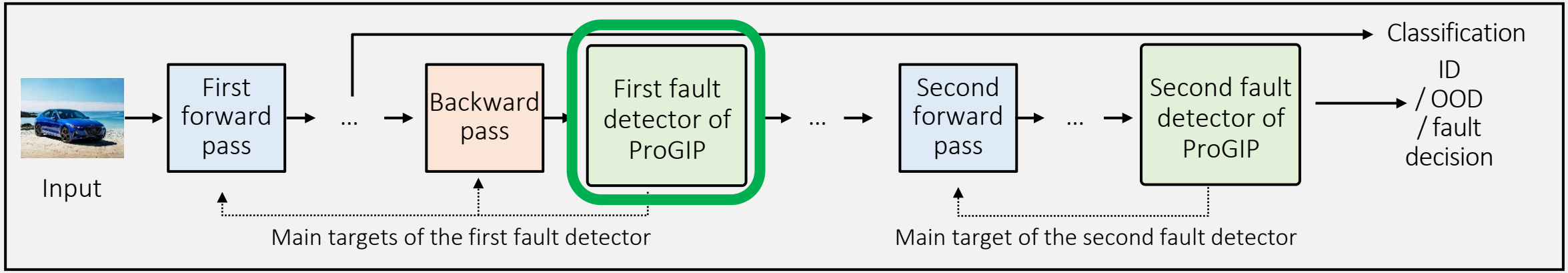


Errors in second forward pass leads to incorrect ID/OOD detection failure.

ProGIP: Methodology

- Introduction
- Background
- Related Works
- ProGIP: Observations
- ProGIP: Methodology
- Experimental Setup
- Experimental Results
- Conclusion

Implementing Our Solution for Enhanced Reliability – First Fault Detector



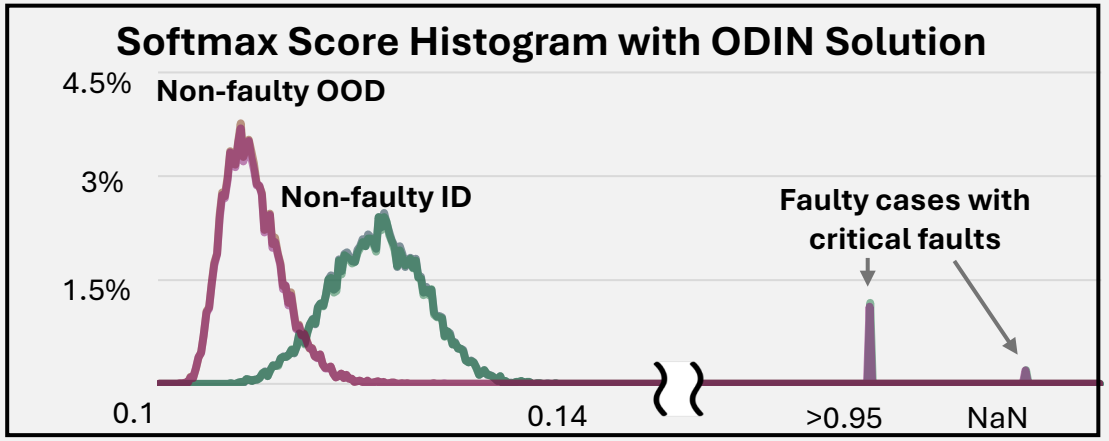
Beyond this point the signature of the faults in first forward and the backward pass disappear but their effects remain.

First fault detector of ProGIP for ODIN

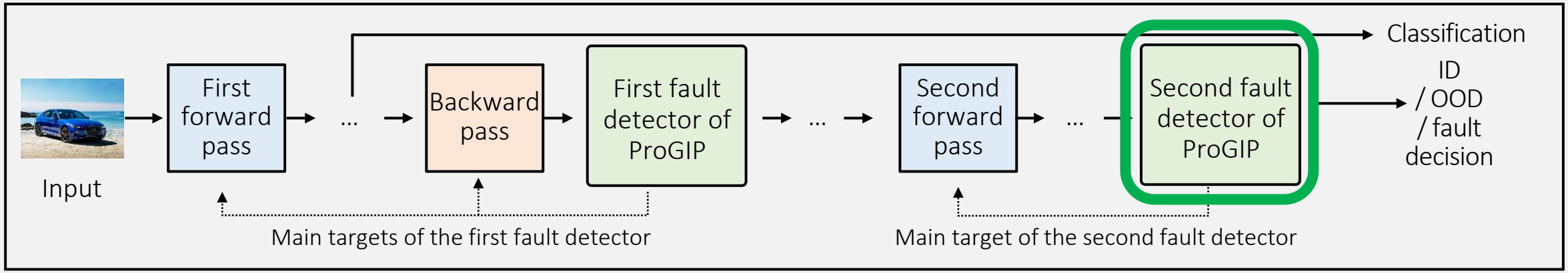
$$Faulty\ \ if\ \ \max(abs(grad)) \approx 0$$
$$Faulty\ \ if\ \max(abs(grad)) > \delta_{F1}$$
$$Fine\ \ otherwise$$

First fault detector of ProGIP for Mahalanobis

$$Faulty\ \ if\ \ \max\big(abs(grad)\big) > \delta_{F1}$$
$$Fine\ \ otherwise$$



Implementing Our Solution for Enhanced Reliability – Second Fault Detector



Second fault detector of ProGIP with ODIN

$$OOD \ \ if \ \ score_O(\tilde{x}) \leq \delta_{OOD}$$
$$ID \ \ if \ \ \delta_{OOD} < score_O(\tilde{x}) \leq \delta_{F2}$$
$$Faulty \ \ if \ \ score_O(\tilde{x}) > \delta_{F2}$$

Second fault detector of ProGIP with Mahalanobis

$$OOD \ \ if \ \ score_O(\tilde{x}) \leq \delta_{OOD}$$
$$ID \ \ if \ \ \delta_{OOD} < score_O(\tilde{x}) \leq \delta_{F2}$$
$$Faulty \ \ if \ \ score_O(\tilde{x}) > \delta_{F2}$$

Experimental Setup

- Introduction
- Background
- Related Works
- ProGIP: Observations
- ProGIP: Methodology
- Experimental Setup
- Experimental Results
- Conclusion

Experimental Setup for OOD Detection

Architectures

- DenseNet-BC
- ResNet34

Datasets

- CIFAR10 – ID
- ImageNet resize (32x 32) - OOD

OOD Detection Methods

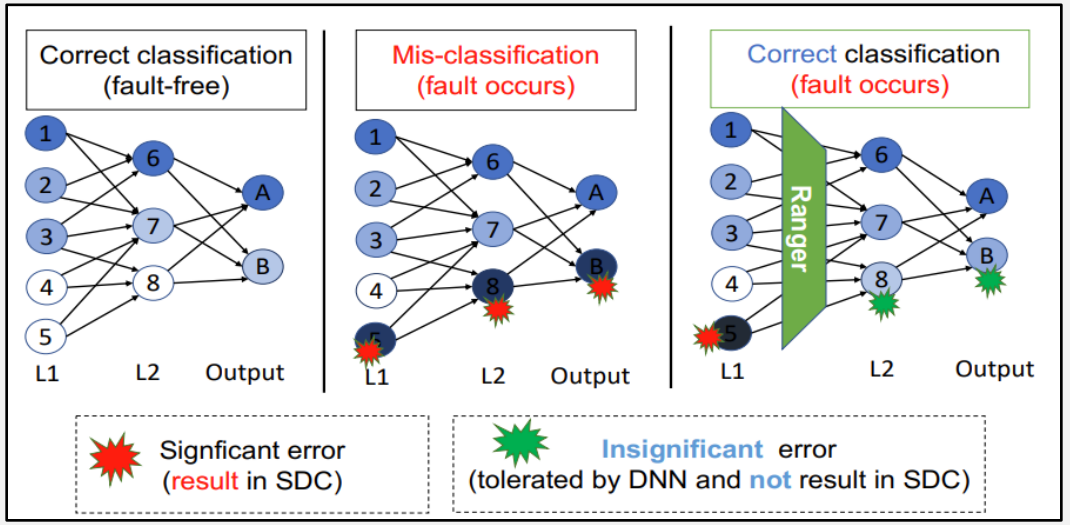
- ODIN
- Mahanalobis

Comparing Methods

- Detection-Only Ranger
- ProGIP (Ours)

Detection-Only Ranger Extended To Protect OOD Detection System from Soft Errors

- Checks the abnormal values of all activation function layers and following max pool, average pool, and reshape layers, via built-in hook methods of PyTorch.
- Our detection-only Ranger is implemented to only provide a fault detection alarm.



Z. Chen, et. al. "A low-cost fault corrector for deep neural networks through range restriction," 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Net-works (DSN). IEEE, pp. 1–13.

Base Accuracy Overview: Key Metrics

Network	OOD Detection	ID Classification Accuracy	AUROC for TPR vs FPR curve	FPR at 95% TPR
DenseNet-BC	ODIN	95.19%	98.40%	8.0%
	Mahanalobis		96.40%	15.0%
ResNet-34	ODIN	94.61%	90.30%	39.2%
	Mahanalobis		93.00%	35.2%

Implement Fault Injection with Bit-Flips in Neural Networks

- Implemented bit-flips in neural network outputs using PyTorch hooks.
- Randomly select a layer execution pass for fault injection.
- Inject faults by flipping a random bit in the selected layer's output.
- Total: 2.4 million fault injections
  - 100,000 runs × 2 networks × 2 OOD detections × 2 dataset types (ID/OOD) × 3 execution passes.

Experimental Results

- Introduction
- Background
- Related Works
- ProGIP: Observations
- ProGIP: Methodology
- Experimental Setup
- Experimental Results
- Conclusion

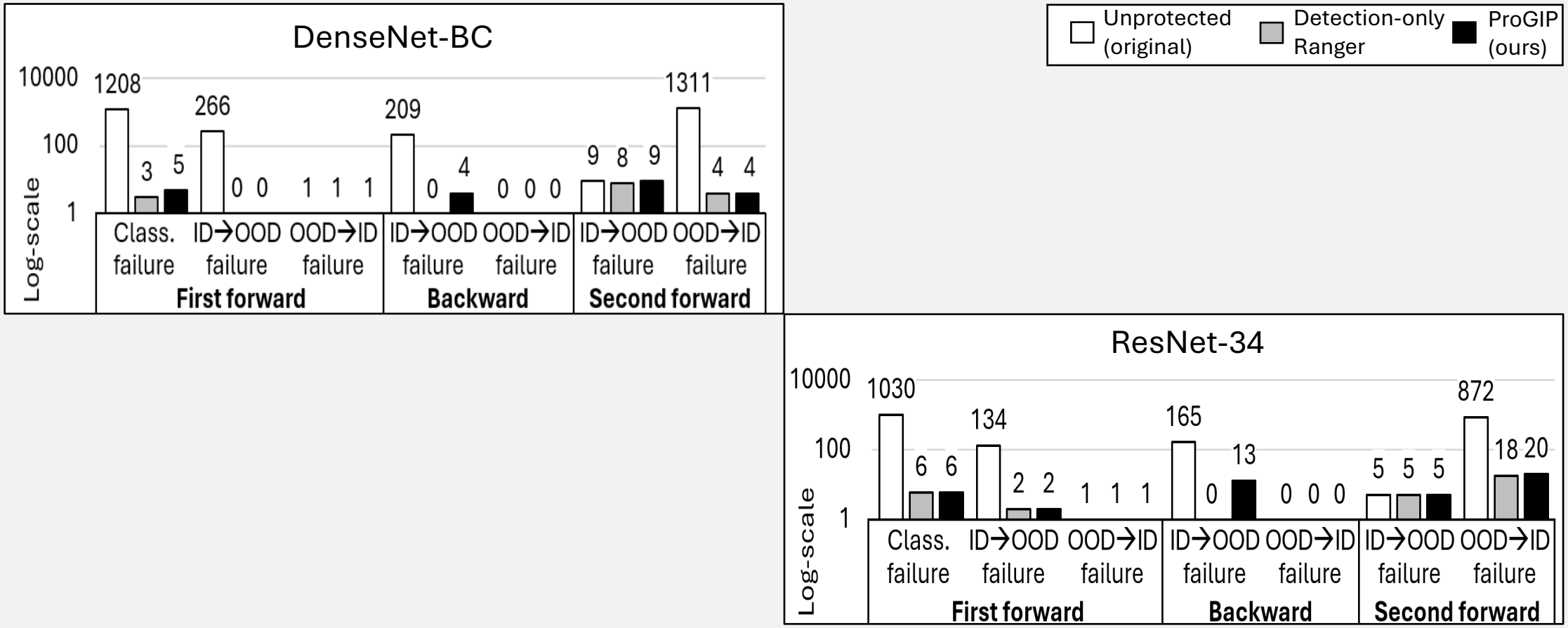
Summary of Runtime Results

Network	OOD Detection	Normalized Execution Time by Unprotected Run		
		Unprotected	Detection-only Ranger	ProGIP (Ours)
DenseNet-BC	ODIN	100%	264.79%	**100.74%**
	Mahanalobis		259.01%	**100.68%**
ResNet-34	ODIN		245.48%	**101.24%**
	Mahanalobis		234.17%	**100.56%**
Average			250.86%	**100.81%**

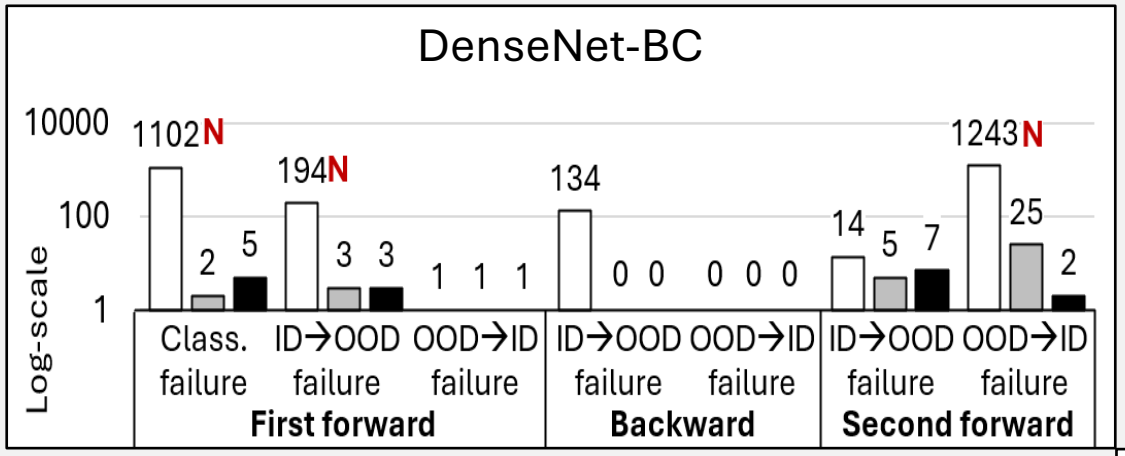
Overall Results

	Fault Injection Run	Originally Correct Run	Classification Failure	Classification Failures Detection Accuracy	ID/OOD Detection Failure	ID/OOD Failures Detection Accuracy	Overall Fault Detection Accuracy	Normalized Runtime Execution by Unprotected
Unprotected	2,400,000	2,004,945	4138	N/A	5989	N/A	N/A	100%
Detection-Only Ranger	2,400,000	2,004,945	12	99.71%	133	97.78%	98.57%	250.86%
ProGIP (ours)	**2,400,000**	**2,004,945**	**17**	**99.59%**	**111**	**98.15%**	**98.74%**	**100.81%**

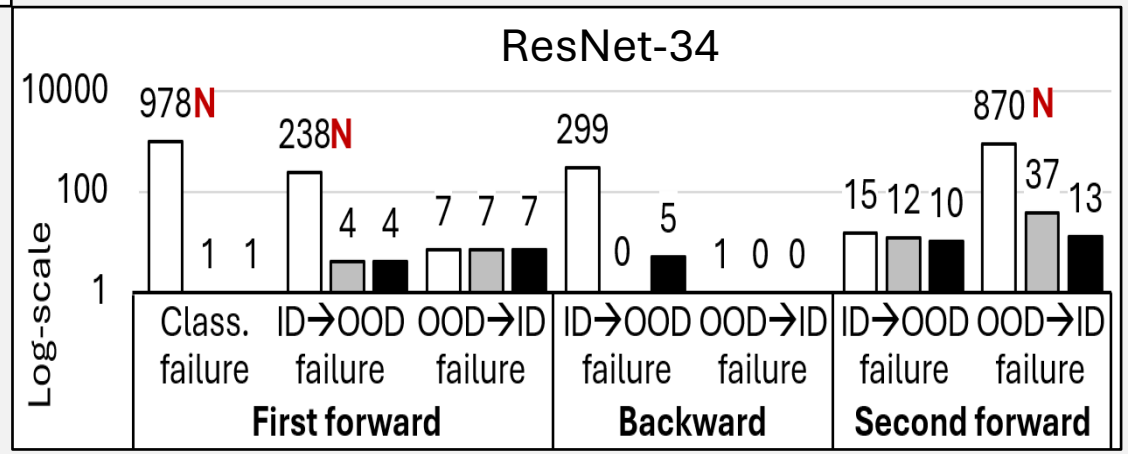
Results of ODIN Across All Passes



Results of Mahanalobis Across All Passes



N: not-a-number-dominant (NaN-dominant), which means more than 96% failures produced not-a-number outputs (confidence score for forward passes and gradient for backward pass).



Publications

- ProGIP: Protecting Gradient-based Input Perturbation Approaches for Out-of-distribution Detection From Soft Errors
  - Authors: **Sumedh Joshi**, Hwisoo So, Soyeong Park, Woobin Ko, Jinhyo Jung, Yohan Ko, Uiwon Hwang, Kyoungwoo Lee and Aviral Shrivastava
  - Design, Automation, and Test in Europe (DATE), 2025 (Submitted and under review)
- Maintaining Sanity: Algorithm-based Comprehensive Fault Tolerance for CNNs
  - Authors: Jinhyo Jung, Hwisoo So, Woobin Ko, **Sumedh Joshi**, Yebon Kim, Yohan Ko, Aviral Shrivastava, Kyoungwoo Lee
  - Design Automation Conference (DAC), 2024

Conclusion and Future Scope

- ProGIP detects soft errors alongside existing GIP based OOD detection techniques.
- Negligible runtime overhead of 0.81%.
- Achieves high detection rates of 98.74% with minimal checkpoint insertions.
- Adaptable thresholds for diverse application requirements.
- Integration with other OOD detection techniques.
- Investigation of combined defenses against multiple reliability threats.