MITIGATING MULTIPLE SINGLE-EVENT UPSETS DURING DEEP NEURAL NETWORK INFERENCE USING FAULT-AWARE TRAINING

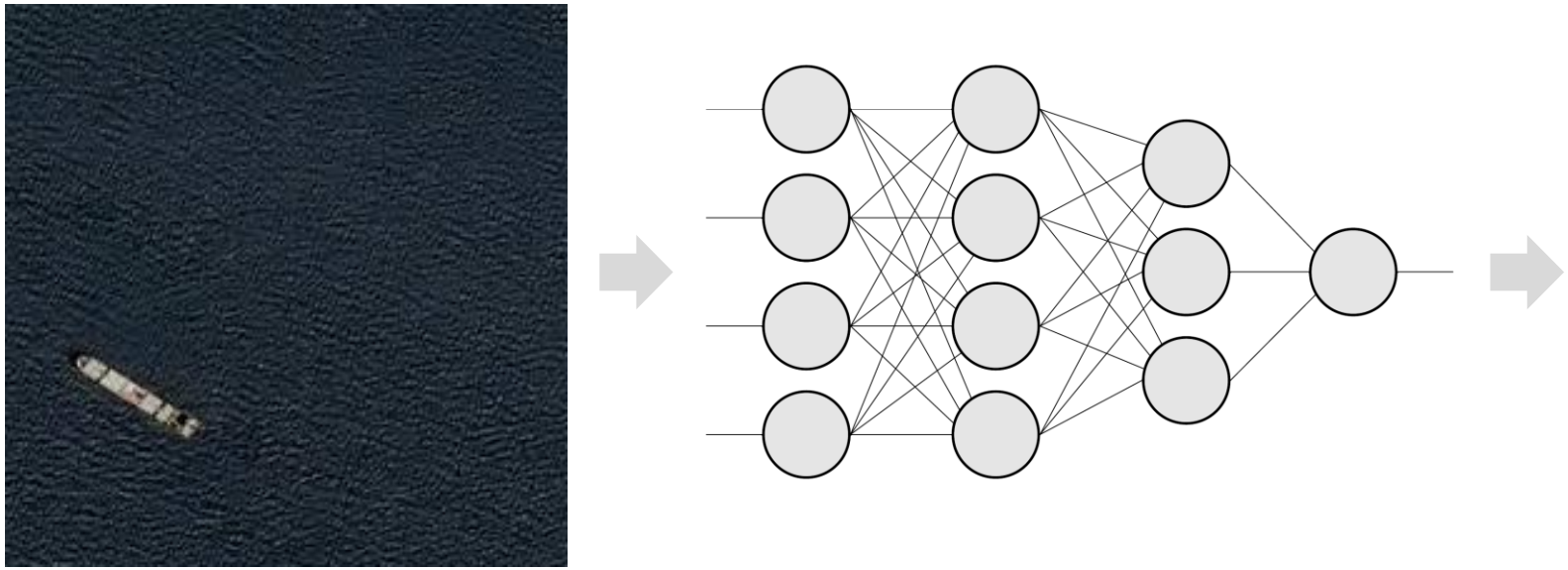
TWEPP 2024

Toon Vinck, Gert Dekkers, Jeffrey Prinzie, Naïn Jonckers and Peter Karsmakers

KU LEUVEN

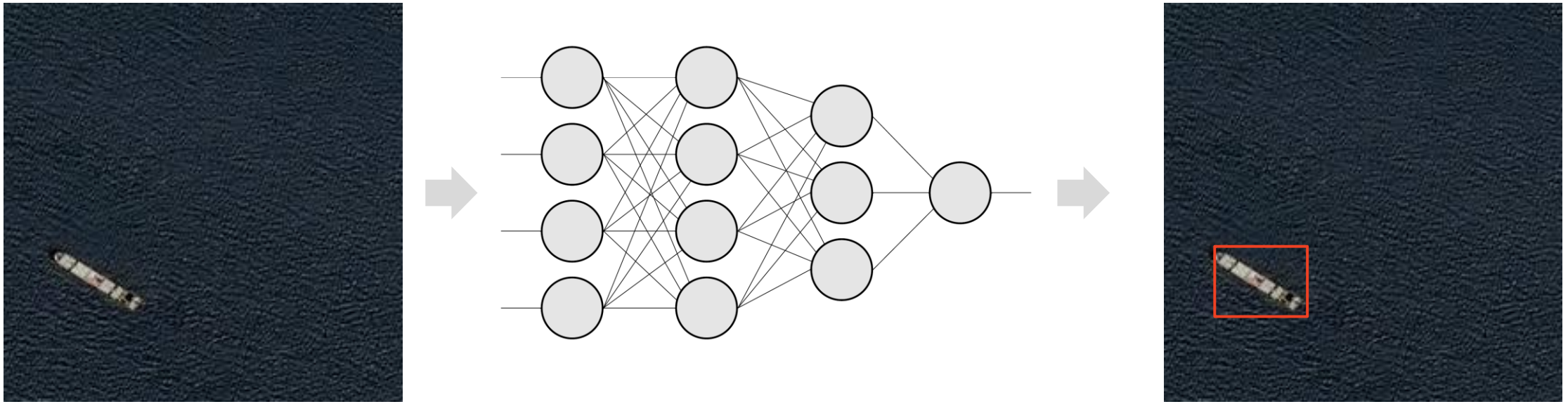
MAGICS

Single Event Upsets (SEUs) in DNNs



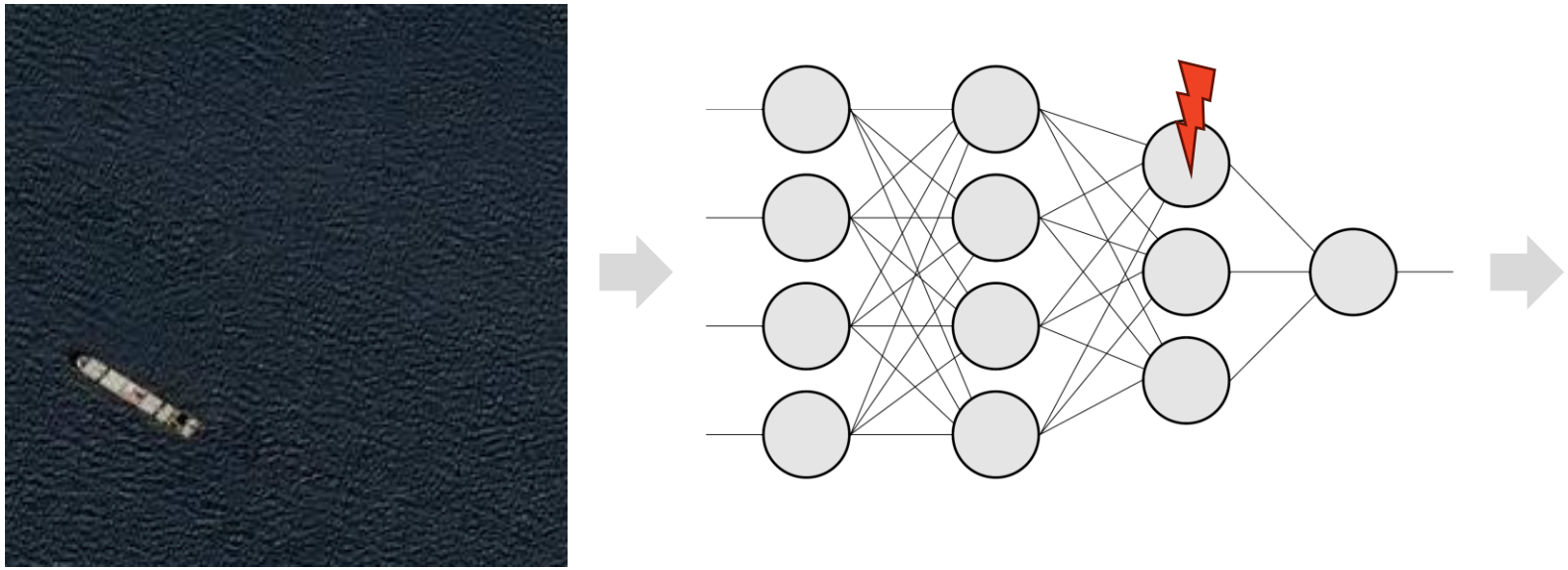
inversion Jeff Faudi, M. Airbus Ship Detection Challenge 2018. https://kaggle.com/competitions/airbus-ship-detection.

Single Event Upsets (SEUs) in DNNs



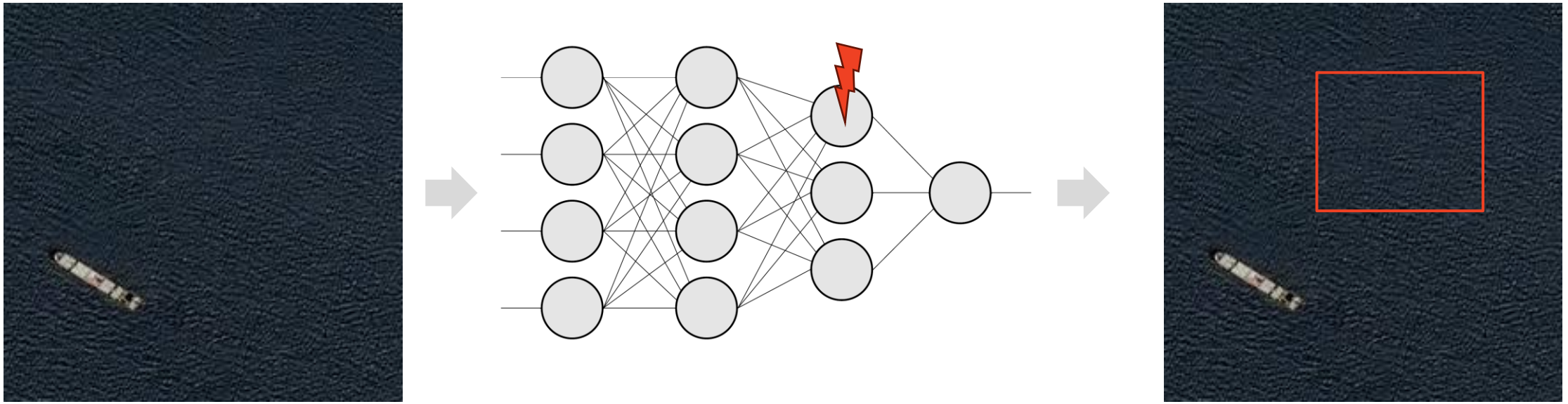
inversion Jeff Faudi, M. Airbus Ship Detection Challenge 2018. https://kaggle.com/competitions/airbus-ship-detection.

Single Event Upsets (SEUs) in DNNs



inversion Jeff Faudi, M. Airbus Ship Detection Challenge 2018. https://kaggle.com/competitions/airbus-ship-detection.

Single Event Upsets (SEUs) in DNNs



inversion Jeff Faudi, M. Airbus Ship Detection Challenge 2018. https://kaggle.com/competitions/airbus-ship-detection.

How do SEUs impact DNNs?

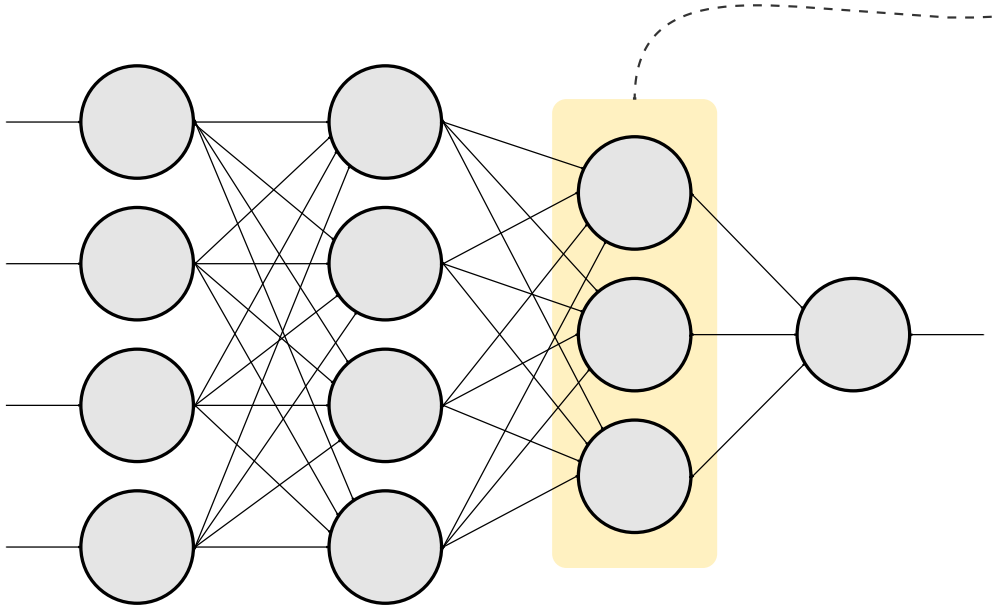
Can **Fault Aware Training** (FAT) improve the DNNs' robustness to SEUs?

KU LEUVEN

How do SEUs impact DNNs?

Can **Fault Aware Training** (FAT) improve the DNNs' robustness to SEUs?

DNN: Matrix multiplications



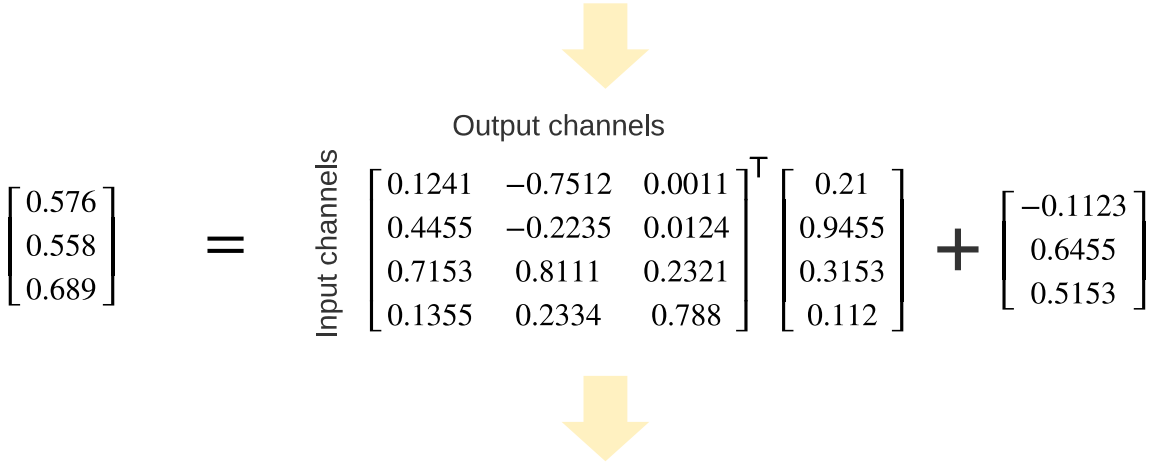
$$\mathbf{y} = \mathbf{W}^T\mathbf{x} + \mathbf{b}$$

$$\begin{bmatrix} 0.576 \\ 0.558 \\ 0.689 \end{bmatrix} = \begin{matrix} \\ \text{Input channels} \end{matrix} \overset{\text{Output channels}}{\begin{bmatrix} 0.1241 & -0.7512 & 0.0011 \\ 0.4455 & -0.2235 & 0.0124 \\ 0.7153 & 0.8111 & 0.2321 \\ 0.1355 & 0.2334 & 0.788 \end{bmatrix}^T} \begin{bmatrix} 0.21 \\ 0.9455 \\ 0.3153 \\ 0.112 \end{bmatrix} + \begin{bmatrix} -0.1123 \\ 0.6455 \\ 0.5153 \end{bmatrix}$$

$$y_0 = w_{00}x_0 + w_{10}x_1 + w_{20}x_2 + w_{30}x_3 + b_0$$

DNN: Matrix multiplications

$$\mathbf{y} = \mathbf{W}^T\mathbf{x} + \mathbf{b}$$



$$\begin{bmatrix} 0.576 \\ 0.558 \\ 0.689 \end{bmatrix} = \begin{bmatrix} 0.1241 & -0.7512 & 0.0011 \\ 0.4455 & -0.2235 & 0.0124 \\ 0.7153 & 0.8111 & 0.2321 \\ 0.1355 & 0.2334 & 0.788 \end{bmatrix}^T \begin{bmatrix} 0.21 \\ 0.9455 \\ 0.3153 \\ 0.112 \end{bmatrix} + \begin{bmatrix} -0.1123 \\ 0.6455 \\ 0.5153 \end{bmatrix}$$

$$y_0 = w_{00}x_0 + w_{10}x_1 + w_{20}x_2 + w_{30}x_3 + b_0$$

DNN: Matrix multiplications

Input

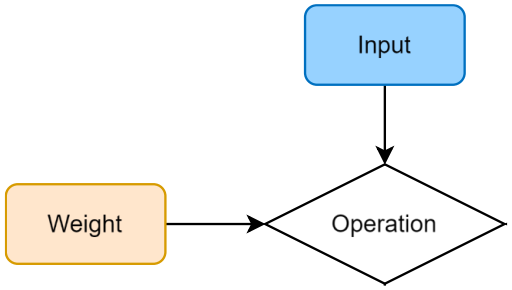
$$\mathbf{y} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$$

Output channels

$$\begin{bmatrix} 0.576 \\ 0.558 \\ 0.689 \end{bmatrix} = \text{Input channels} \begin{bmatrix} 0.1241 & -0.7512 & 0.0011 \\ 0.4455 & -0.2235 & 0.0124 \\ 0.7153 & 0.8111 & 0.2321 \\ 0.1355 & 0.2334 & 0.788 \end{bmatrix}^T \begin{bmatrix} 0.21 \\ 0.9455 \\ 0.3153 \\ 0.112 \end{bmatrix} + \begin{bmatrix} -0.1123 \\ 0.6455 \\ 0.5153 \end{bmatrix}$$

$$y_0 = w_{00} x_0 + w_{10} x_1 + w_{20} x_2 + w_{30} x_3 + b_0$$

DNN: Matrix multiplications

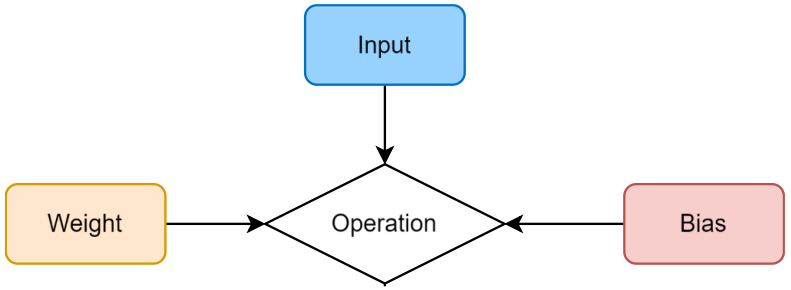


$$\mathbf{y} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$$

$$\begin{bmatrix} 0.576 \\ 0.558 \\ 0.689 \end{bmatrix} = \begin{matrix} \text{Input channels} \\ \end{matrix} \overset{\text{Output channels}}{\begin{bmatrix} 0.1241 & -0.7512 & 0.0011 \\ 0.4455 & -0.2235 & 0.0124 \\ 0.7153 & 0.8111 & 0.2321 \\ 0.1355 & 0.2334 & 0.788 \end{bmatrix}^T} \begin{bmatrix} 0.21 \\ 0.9455 \\ 0.3153 \\ 0.112 \end{bmatrix} + \begin{bmatrix} -0.1123 \\ 0.6455 \\ 0.5153 \end{bmatrix}$$

$$y_0 = w_{00} x_0 + w_{10} x_1 + w_{20} x_2 + w_{30} x_3 + b_0$$

DNN: Matrix multiplications

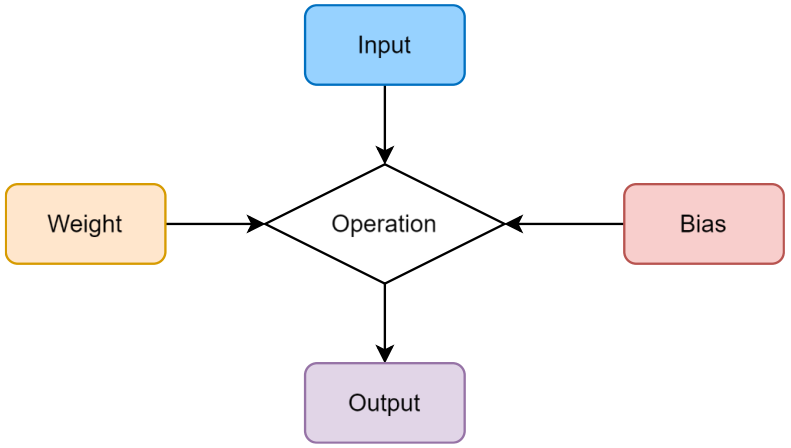


$$\mathbf{y} = \mathbf{W}^T\mathbf{x} + \mathbf{b}$$

$$\begin{bmatrix} 0.576 \\ 0.558 \\ 0.689 \end{bmatrix} = \begin{matrix} \text{Input channels} \\ \end{matrix} \overset{\text{Output channels}}{\begin{bmatrix} 0.1241 & -0.7512 & 0.0011 \\ 0.4455 & -0.2235 & 0.0124 \\ 0.7153 & 0.8111 & 0.2321 \\ 0.1355 & 0.2334 & 0.788 \end{bmatrix}^T} \begin{bmatrix} 0.21 \\ 0.9455 \\ 0.3153 \\ 0.112 \end{bmatrix} + \begin{bmatrix} -0.1123 \\ 0.6455 \\ 0.5153 \end{bmatrix}$$

$$y_0 = w_{00}x_0 + w_{10}x_1 + w_{20}x_2 + w_{30}x_3 + b_0$$

DNN: Matrix multiplications



$$\mathbf{y} = \mathbf{W}^T\mathbf{x} + \mathbf{b}$$

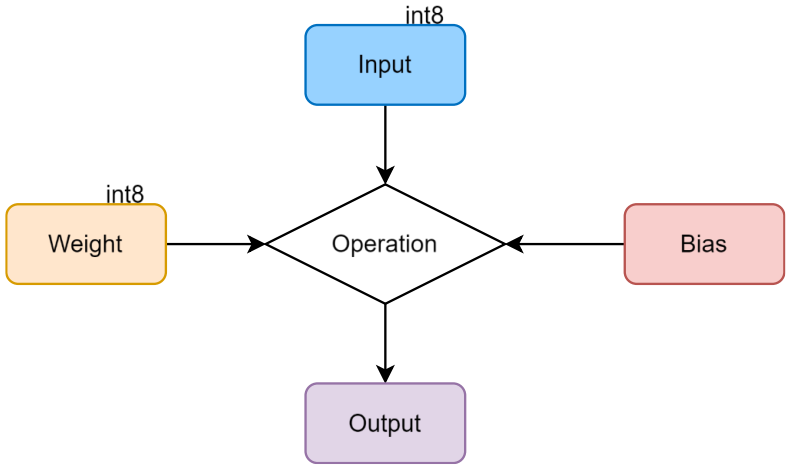
$$\begin{bmatrix} 0.576 \\ 0.558 \\ 0.689 \end{bmatrix} = \begin{bmatrix} 0.1241 & -0.7512 & 0.0011 \\ 0.4455 & -0.2235 & 0.0124 \\ 0.7153 & 0.8111 & 0.2321 \\ 0.1355 & 0.2334 & 0.788 \end{bmatrix}^T \begin{bmatrix} 0.21 \\ 0.9455 \\ 0.3153 \\ 0.112 \end{bmatrix} + \begin{bmatrix} -0.1123 \\ 0.6455 \\ 0.5153 \end{bmatrix}$$

Output channels

Input channels

$$y_0 = w_{00}x_0 + w_{10}x_1 + w_{20}x_2 + w_{30}x_3 + b_0$$

Quantized DNN



$$\mathbf{y} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$$

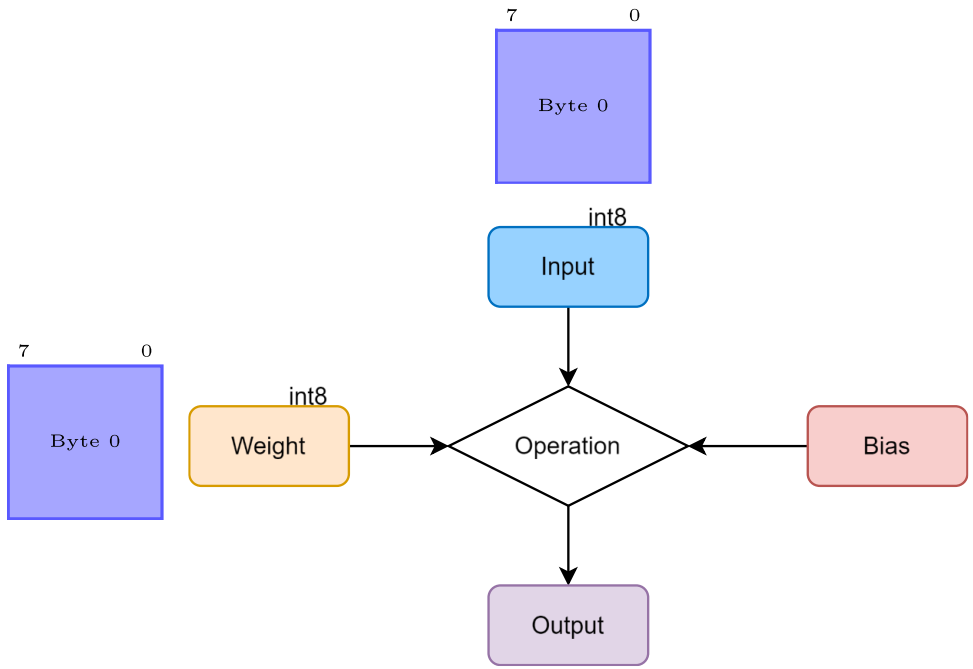
$$\begin{bmatrix} 9437 \\ 9142 \\ 11289 \end{bmatrix} = \begin{bmatrix} 16 & -96 & 0 \\ 57 & -29 & 2 \\ 92 & 104 & 30 \\ 17 & 30 & 101 \end{bmatrix}^T \begin{bmatrix} 27 \\ 121 \\ 40 \\ 14 \end{bmatrix} + \begin{bmatrix} -1840 \\ 10575 \\ 8443 \end{bmatrix}$$

Output channels

Input channels

$$y_0 = w_{00}x_0 + w_{10}x_1 + w_{20}x_2 + w_{30}x_3 + b_0$$

Quantized DNN

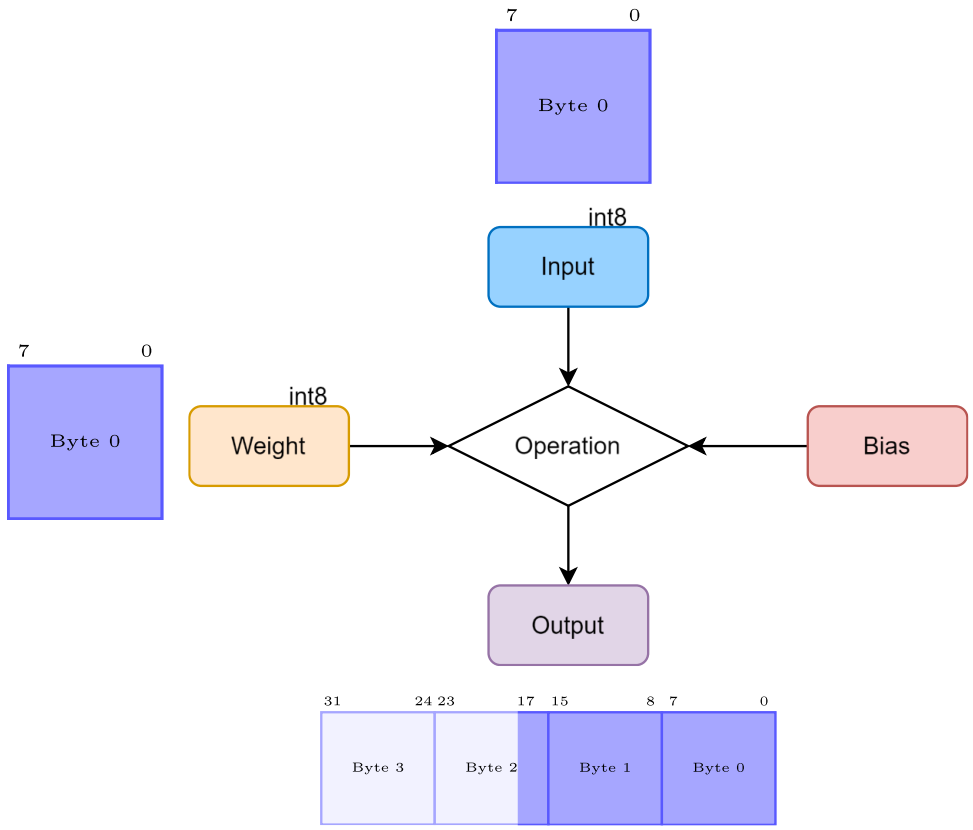


$$\mathbf{y} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$$

$$\begin{bmatrix} 9437 \\ 9142 \\ 11289 \end{bmatrix} = \begin{matrix} \text{Input channels} \\ \end{matrix} \overset{\text{Output channels}}{\begin{bmatrix} 16 & -96 & 0 \\ 57 & -29 & 2 \\ 92 & 104 & 30 \\ 17 & 30 & 101 \end{bmatrix}^T} \begin{bmatrix} 27 \\ 121 \\ 40 \\ 14 \end{bmatrix} + \begin{bmatrix} -1840 \\ 10575 \\ 8443 \end{bmatrix}$$

$$y_0 = w_{00}x_0 + w_{10}x_1 + w_{20}x_2 + w_{30}x_3 + b_0$$

Quantized DNN

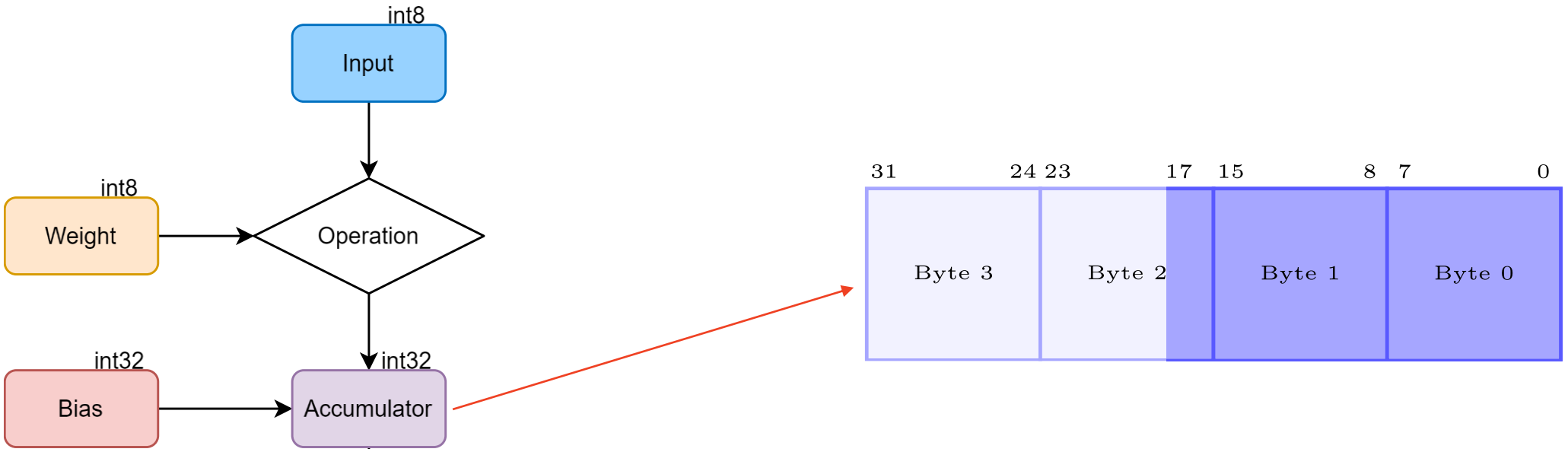


$$\mathbf{y} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$$

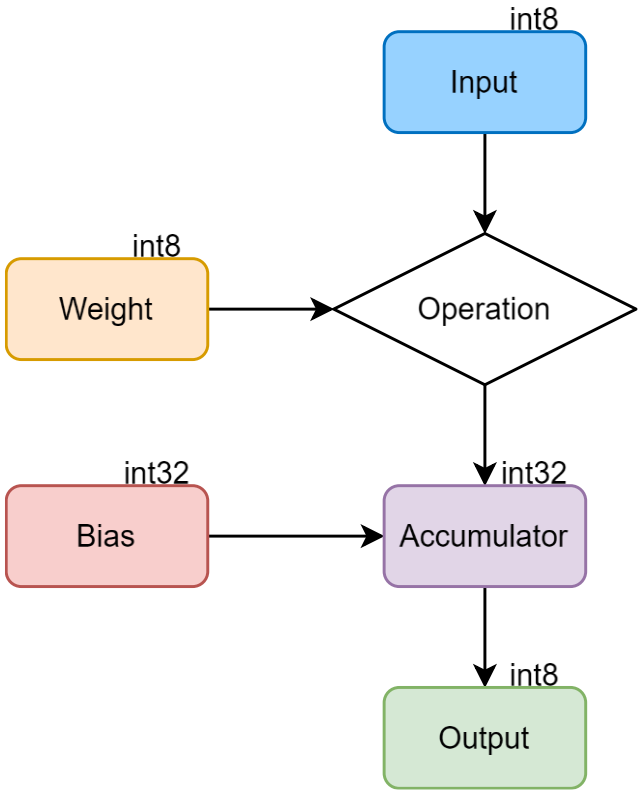
$$\begin{bmatrix} 9437 \\ 9142 \\ 11289 \end{bmatrix} = \begin{matrix} \\ \text{Input channels} \end{matrix} \overset{\text{Output channels}}{\begin{bmatrix} 16 & -96 & 0 \\ 57 & -29 & 2 \\ 92 & 104 & 30 \\ 17 & 30 & 101 \end{bmatrix}^T} \begin{bmatrix} 27 \\ 121 \\ 40 \\ 14 \end{bmatrix} + \begin{bmatrix} -1840 \\ 10575 \\ 8443 \end{bmatrix}$$

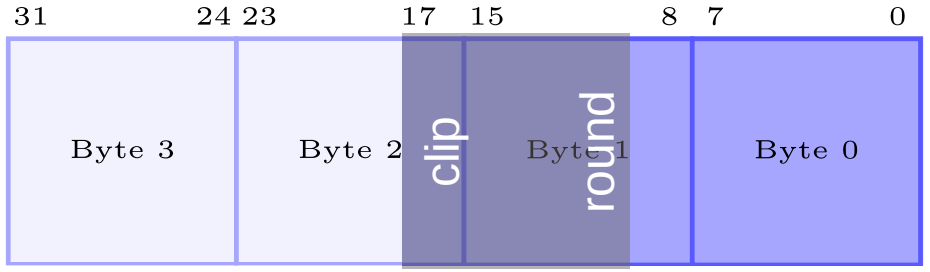
$$y_0 = w_{00}x_0 + w_{10}x_1 + w_{20}x_2 + w_{30}x_3 + b_0$$

Quantized DNN



Quantized DNN

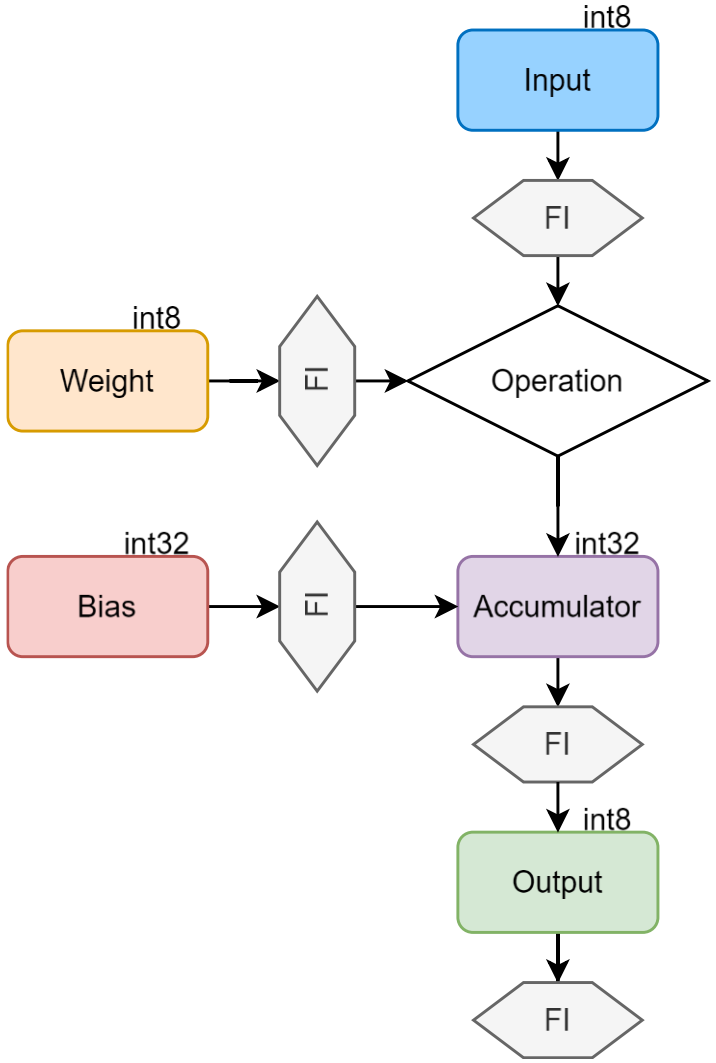




How do SEUs impact DNNs?

Can **Fault Aware Training** (FAT) improve the DNNs' robustness to multiple SEUs?

Fault Injection



Simulate SEUs:

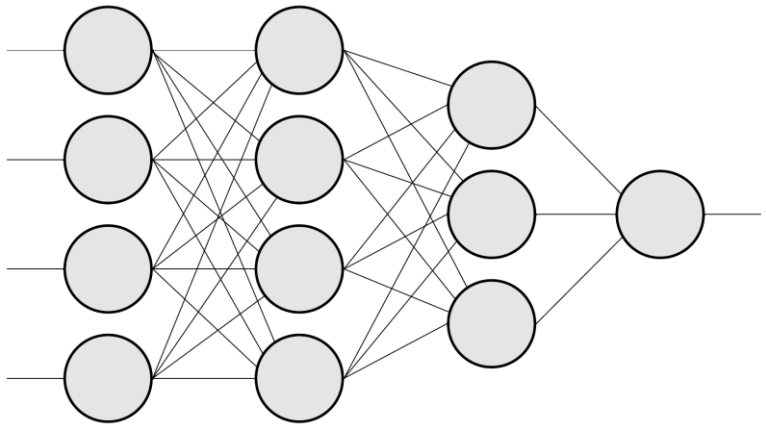
Design **Fault Injection tool** in **PyTorch**

- Flip randomly selected bits in model

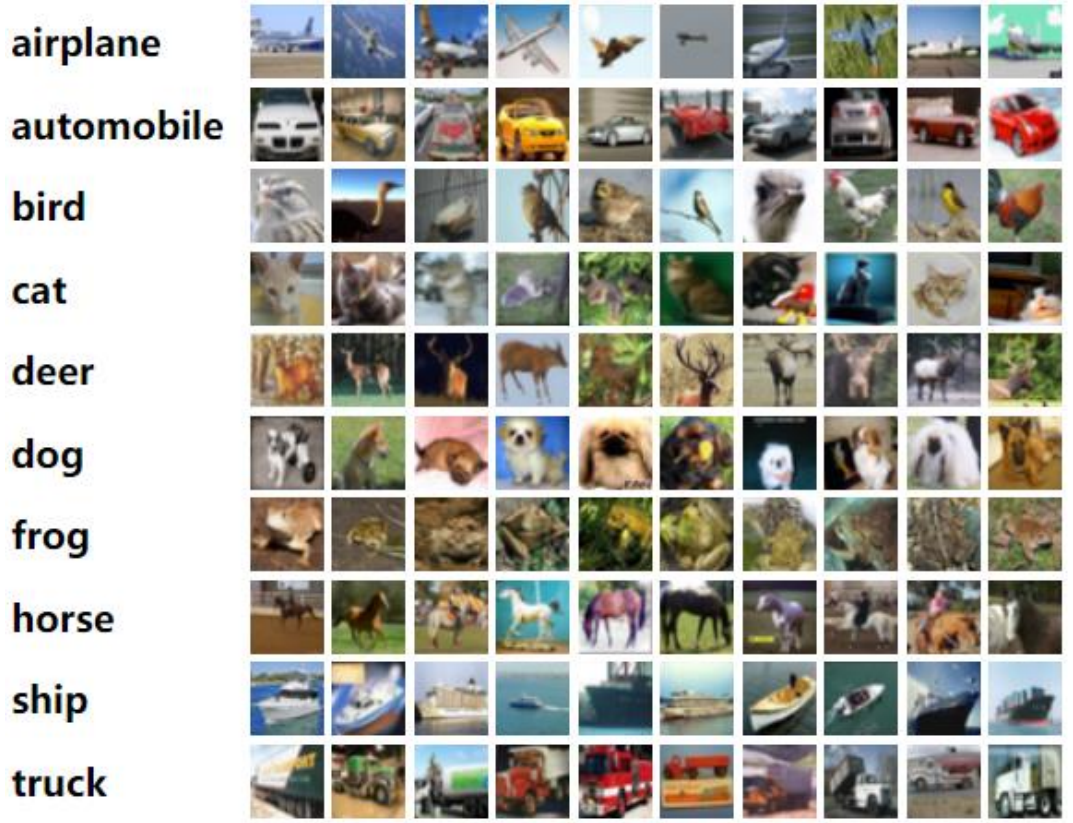
Experiment

KU LEUVEN

MobileNetV2

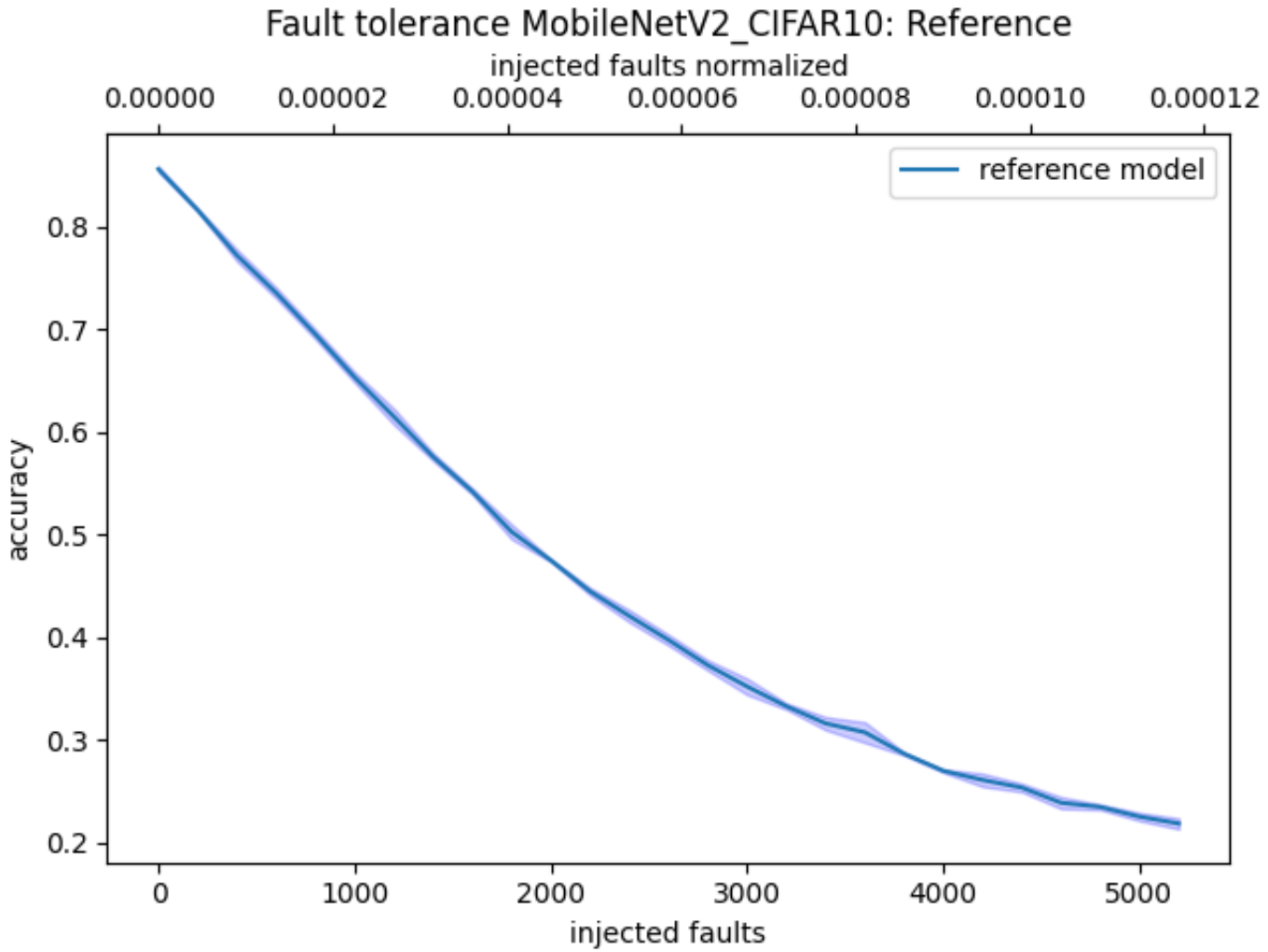


CIFAR10

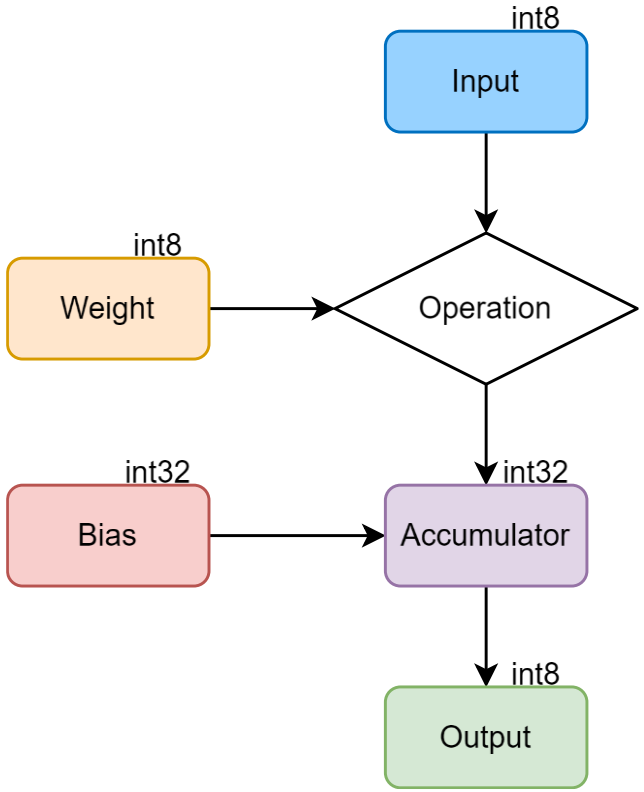


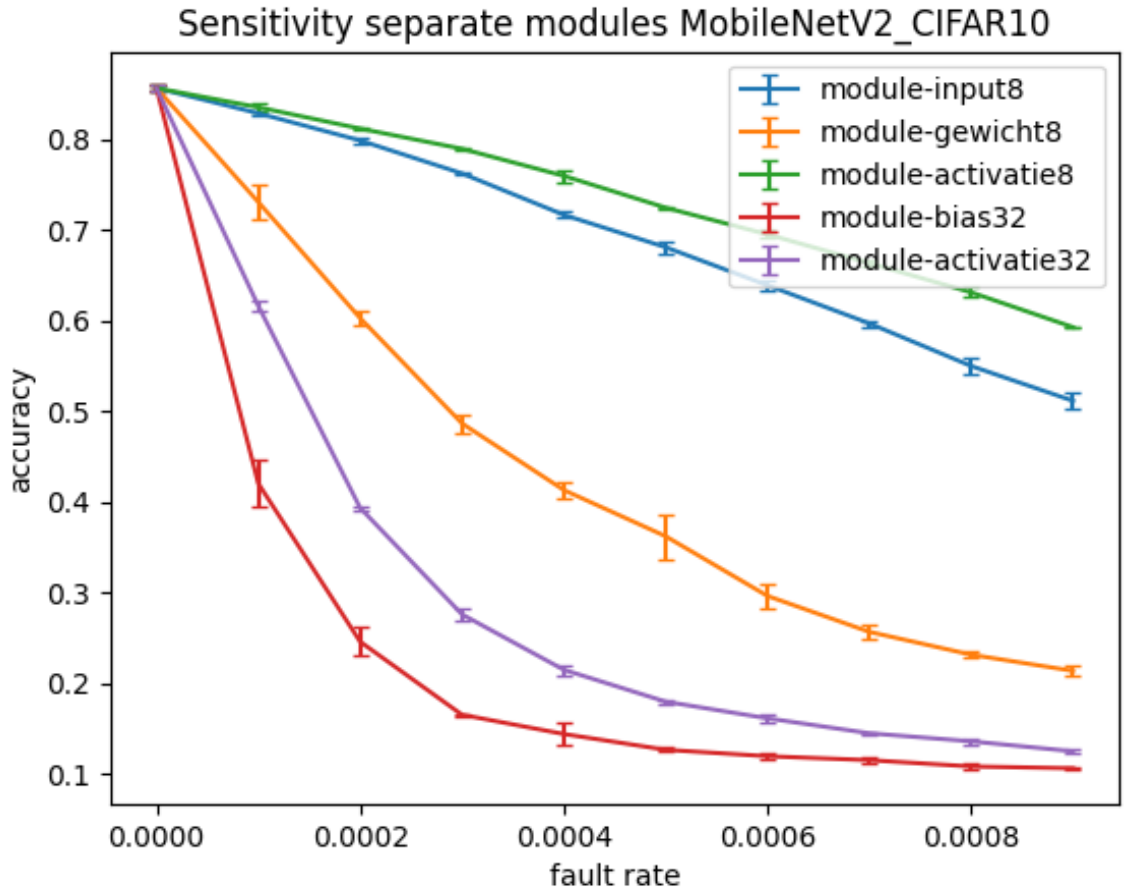
Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images (2009).

Results

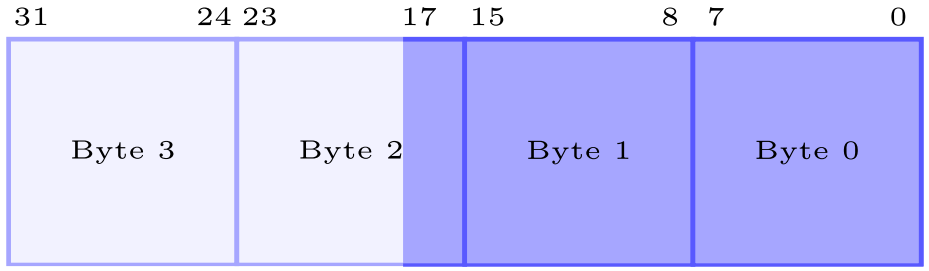


Results



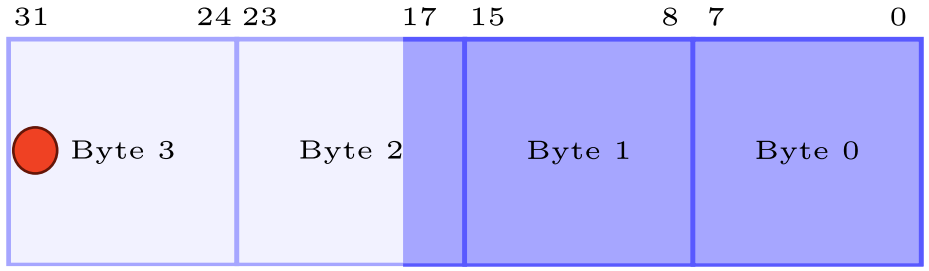


FI in 32 bit-reg



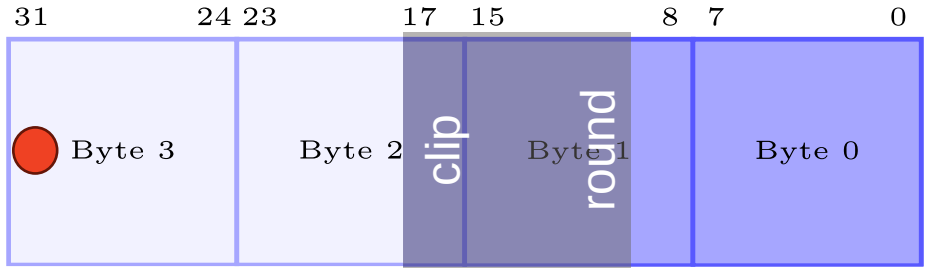
- **Bit flips** can happen in one of the **MSBs**
- **Large error** compared to original values

FI in 32 bit-reg



- **Bit flips** can happen in one of the **MSBs**
- **Large error** compared to original values

FI in 32 bit-reg



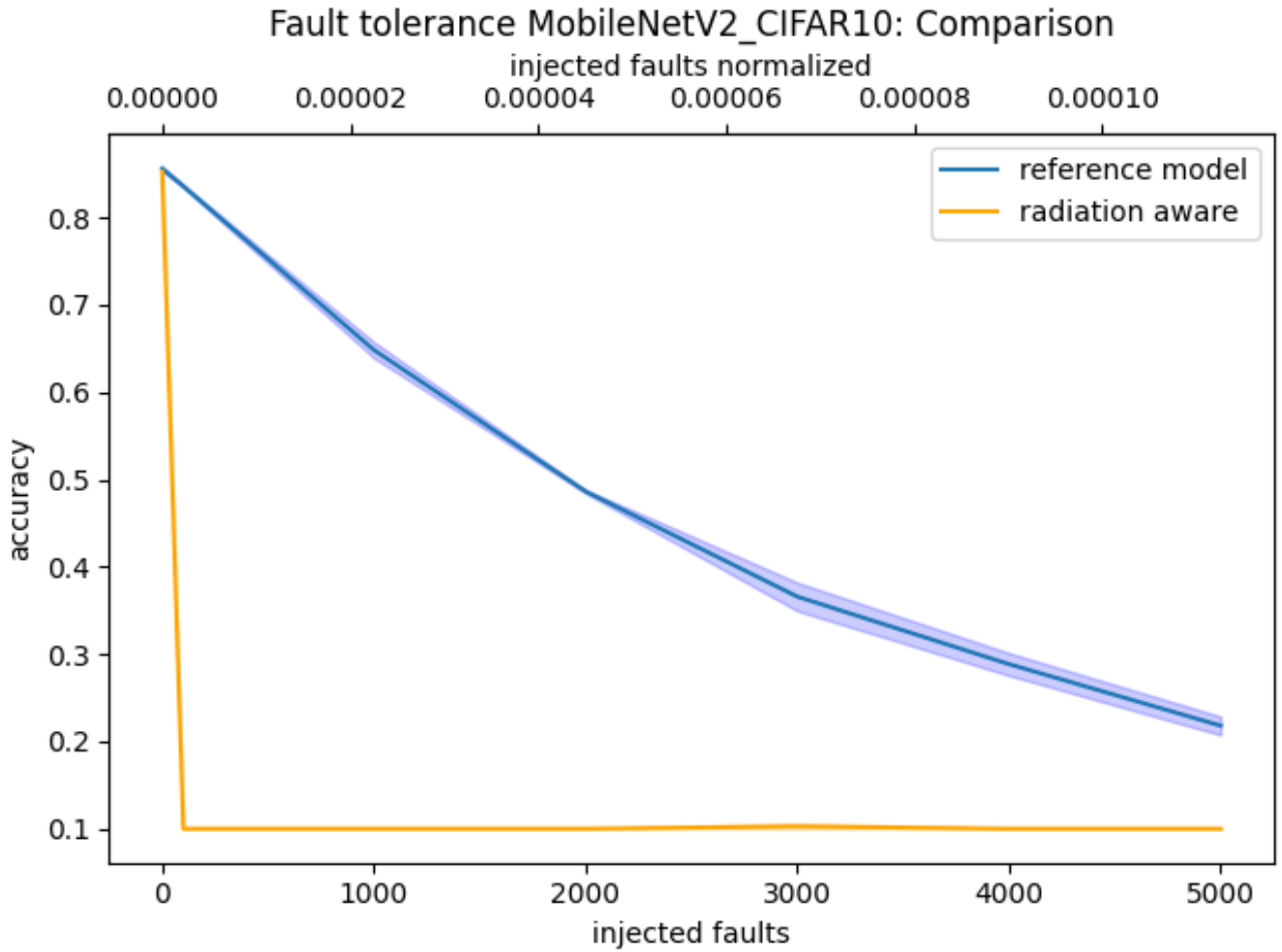
- **Bit flips** can happen in one of the **MSBs**
- **Large error** compared to original values
- **Error** will be clipped

KU LEUVEN

How do SEUs impact DNNs?

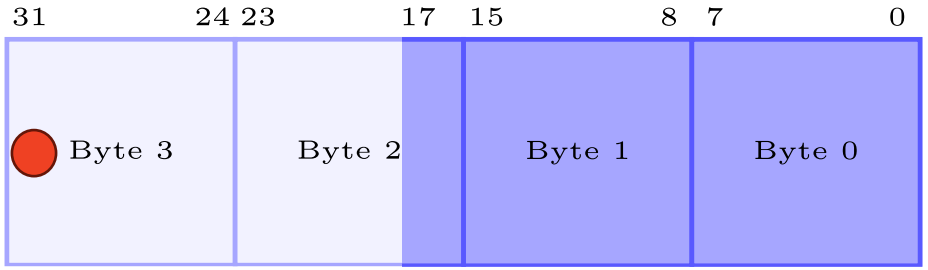
Can **Fault Aware Training** (FAT) improve the DNNs' robustness to multiple SEUs?

Results



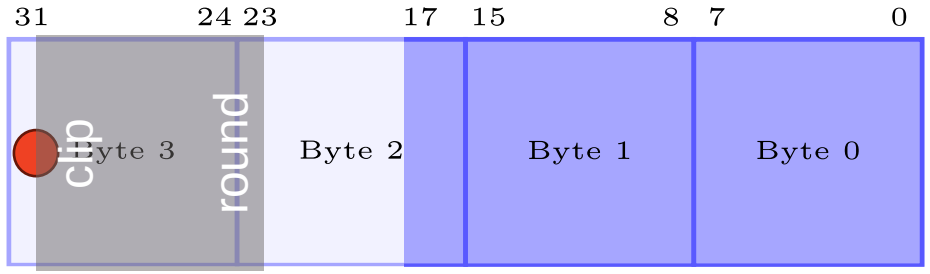
Problem

- **Dynamic scale factor during training**



Problem

- **Dynamic scale factor during training**



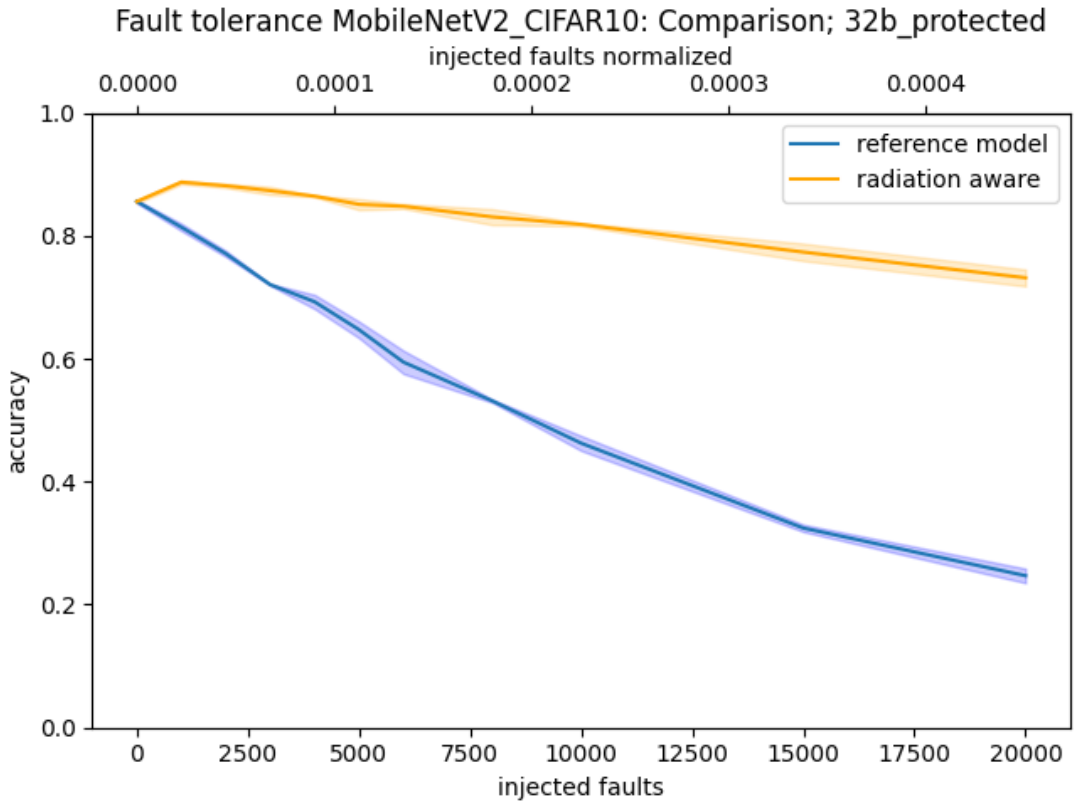
Solution

KU LEUVEN

Assume hardware protection on the 32-bit registers

Solution

Assume hardware protection on the 32-bit registers



Conclusions

- DNNs naturally have a certain tolerance to cope with multiple SEUs.
- Fault Aware Training can improve this tolerance.

- BUT, you have to be aware of the overdimensioned accumulator!