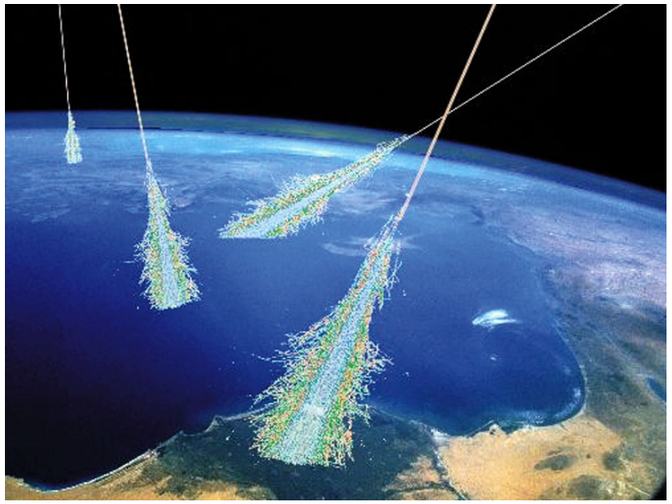
HYBRID FAULT MITIGATION FOR NEURAL NETWORKS BASED ON DIRECTIONAL AND POSITIONAL BIT SENSITIVITY

Wilfread GUILLEME, Angeliki KRITIKAKOU, Youri HELEN, Cédric KILLIAN, Daniel CHILLET

November 19, 2025 – **BITFLIP by DGA (ECW) –** Rennes, France

Reliability challenges in neural network applications

- **Cosmic rays**



https://apod.nasa.gov/apod/ap060814.html

- **Reliability of embedded systems**

© DinosoftLabs, Flaticon

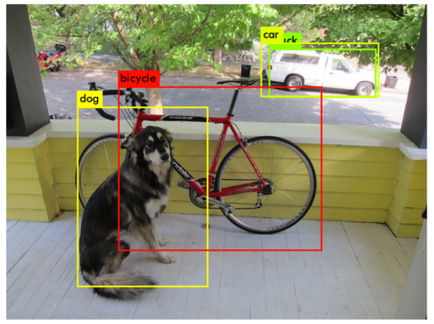
- **Fault consequence…**

Cosmic particles can change elections and cause planes to fall through the sky, scientists warn

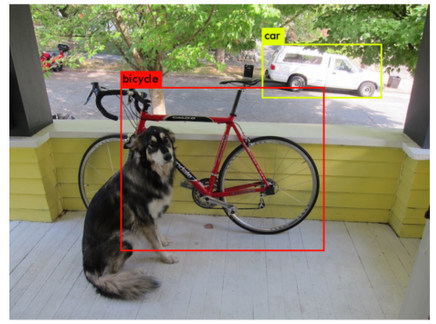
INDEPENDENT

Ian Johnston Science Correspondent in Boston • Friday 17 February 2017 16:40 GMT

- **…On a neural network [1]**



(a) Golden Prediction



(c) Unsafe Observed faults Prediction - strong corruption of prediction result

[1] Bosio et al. (IEEE LATS, 2019). A reliability analysis of a deep neural network.

Reliability challenges in neural network applications

- **Resource-aware protection of WEIGHTS and ACTIVATIONS**
  - Selective TMR (Triple Modular Redundancy) on the most vulnerable layers [2] or channels [3]
  - ECC (e.g., Hamming code) applied to most significant bits [4]
- **Behavior-based hardening**
  - Fault masking through inherent neural network robustness [5]
  - Activation clipping to limit error propagation [6]
- **Evaluation strategies**
  - Exhaustive evaluation is impractical for large models
  - Statistical fault injection methods to efficiently estimate model vulnerability [7] [8]

[2] Libano et al. (IEEE TNS, 2018). Selective hardening for neural networks in FPGAs.
[3] Bertoa et al. (IEEE D&T, 2022). Fault-tolerant neural network accelerators with selective TMR.
[4] Traiola et al. (IEEE ETS, 2023). HarDNNing: a machine-learning-based framework for fault tolerance assessment and protection of DNNs.
[5] Burel et al. (IEEE TDMR, 2022). Mozart+: Masking outputs with zeros for improved architectural robustness and testing of dnn accelerators.
[6] Hoang et al. (IEEE DATE, 2020). Ft-clipact: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation
[7] Leveugle et al. (IEEE DATE, 2009). Statistical fault injection: Quantified error and confidence.
[8] Ruospo et al. (IEEE DATE, 2023). Assessing convolutional neural networks reliability through statistical fault injections.

SFI4NN: Statistical Fault Injection for Neural Networks

- **AlexNet [9] sensitivity analysis**
  - 8-bit fixed-point model
  - CIFAR-10 **[10]**, 10k test images

- **Exhaustive fault injection**
  - 28,505,792 **WEIGHTS** × 8-bit × 10k images
    → **2,280,463,360,000 faults**

Layer type		Number of data	Output shape	Number of parameters
INPUT	Input-0	3,072	[3, 32, 32]	0
CONV-1	Conv-1	57,600	[64, 30, 30]	1,728
	ReLu-2	57,600	[64, 30, 30]	0
	Pool-3	14,400	[64, 15, 15]	0
CONV-2	Conv-4	32,448	[192, 13, 13]	110,592
	ReLu-5	32,448	[192, 13, 13]	0
	Pool-6	6,912	[192, 6, 6]	0
CONV-3	Conv-7	13,824	[384, 6, 6]	663,552
	ReLu-8	13,824	[384, 6, 6]	0
CONV-4	Conv-9	9,216	[256, 6, 6]	884,736
	ReLu-10	9,216	[256, 6, 6]	0
CONV-5	Conv-11	9,216	[256, 6, 6]	589,824
	ReLu-12	9,216	[256, 6, 6]	0
	Pool-13	2,304	[256, 3, 3]	0
FC-1	Flat-14	2,304	[2 304]	0
	Lin-15	4,096	[4 096]	9,437,184
	ReLu-16	4,096	[4 096]	0
FC-2	Lin-17	4,096	[4 096]	16,777,216
	ReLu-18	4,096	[4 096]	0
FC-3	Lin-19	10	[10]	40,960
Total		**289,994**		**28,505,792**

[9] Krizhevsky et al. (NeurIPS, 2012).
Imagenet classification with deep convolutional neural networks.

[10] Krizhevsky et al. (Univ. of Toronto, 2009).
Learning multiple layers of features from tiny images.

SFI4NN: Statistical Fault Injection for Neural Networks

Statistical Fault Injection (SFI) equation: original from [7], adapted for neural networks in [8]

$$n(i,l) = \frac{N(i,l)}{1 + e^2 \cdot \frac{N(i,l)-1}{t^2 \cdot p(i) \cdot (1-p(i))}}$$

- **N → exhaustive set of possible fault locations**
- **n → subset of fault to inject (randomly)**
  - Bit position ***i***
  - Layer ***l***
- **Sample size depending on:**
  - Tolerated error margin ***e***
  - Confidence level ***t***
  - Probability ***p***, related to the bit position

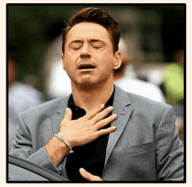
[7] Leveugle et al. (IEEE DATE, 2009). Statistical fault injection: Quantified error and confidence.
[8] Ruospo et al. (IEEE DATE, 2023). Assessing convolutional neural networks reliability through statistical fault injections.

SFI4NN: Statistical Fault Injection for Neural Networks

- **AlexNet sensitivity analysis**
  - 8-bit fixed-point model
  - CIFAR-10, 10k test images

- **Exhaustive fault injection**
  - 28,505,792 **WEIGHTS** × 8-bit × 10k images
    → **2,280,463,360,000 faults**

SFI



- **Statistical fault injection**
  - 10% error margin, 90% confidence level
    → **14,690,000 faults**

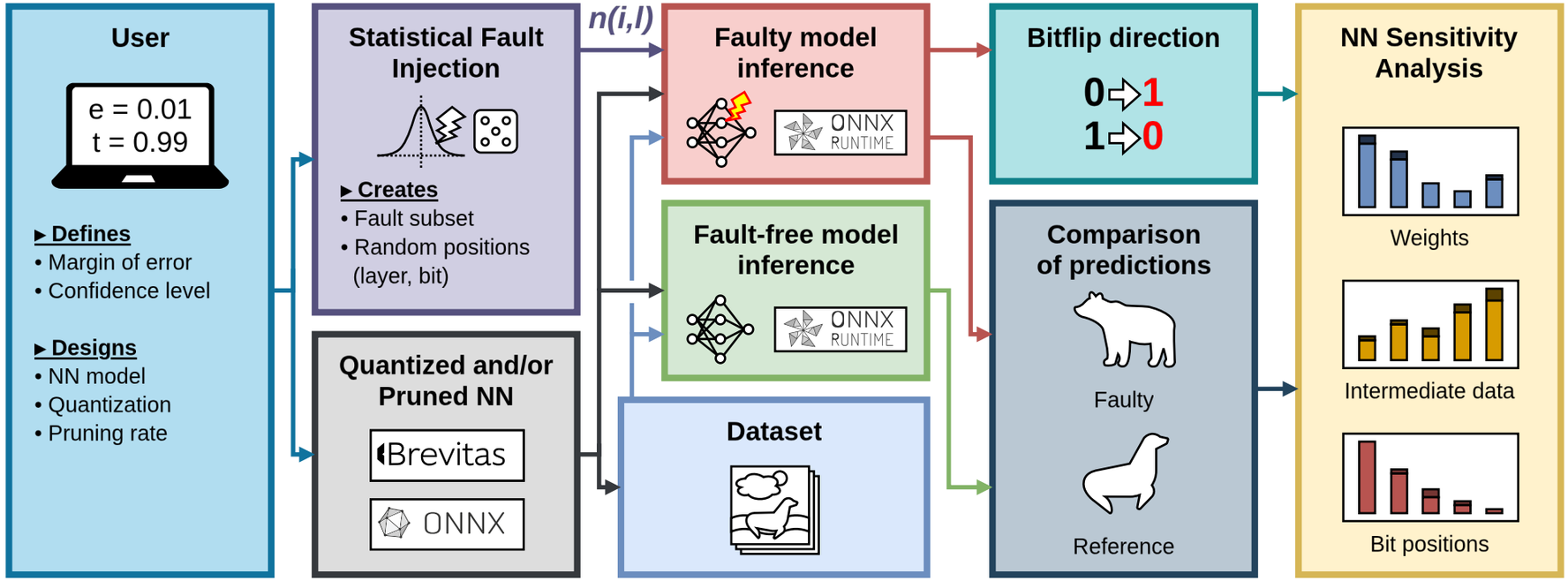
Layer type		Number of data	Output shape	Number of parameters
INPUT	Input-0	3,072	[3, 32, 32]	0
CONV-1	Conv-1	57,600	[64, 30, 30]	1,728
	ReLu-2	57,600	[64, 30, 30]	0
	Pool-3	14,400	[64, 15, 15]	0
CONV-2	Conv-4	32,448	[192, 13, 13]	110,592
	ReLu-5	32,448	[192, 13, 13]	0
	Pool-6	6,912	[192, 6, 6]	0
CONV-3	Conv-7	13,824	[384, 6, 6]	663,552
	ReLu-8	13,824	[384, 6, 6]	0
CONV-4	Conv-9	9,216	[256, 6, 6]	884,736
	ReLu-10	9,216	[256, 6, 6]	0
CONV-5	Conv-11	9,216	[256, 6, 6]	589,824
	ReLu-12	9,216	[256, 6, 6]	0
	Pool-13	2,304	[256, 3, 3]	0
FC-1	Flat-14	2,304	[2 304]	0
	Lin-15	4,096	[4 096]	9,437,184
	ReLu-16	4,096	[4 096]	0
FC-2	Lin-17	4,096	[4 096]	16,777,216
	ReLu-18	4,096	[4 096]	0
FC-3	Lin-19	10	[10]	40,960
Total		**289,994**		**28,505,792**

SFI4NN: Statistical Fault Injection for Neural Networks

Bitflip	Binary	Decimal	Fault directionality
∅	0 0 1 1 . 1 0	3.5	Original value ☑️
1 → 0	0 0 **0** 1 . 1 0	1.5	Closer to zero ↘️
0 → 1	0 **1** 1 1 . 1 0	7.5	Away from zero ↗️
0 → 1	**1** 0 1 1 . 1 0	-4.5	Away from zero (Sign bit) ↗️

- **0-to-1 flip**
  - **Positive** value → Away from zero
  - **Negative** value → Closer to zero
- **1-to-0 flip**
  - **Positive** value → Closer to zero
  - **Negative** value → Away from zero

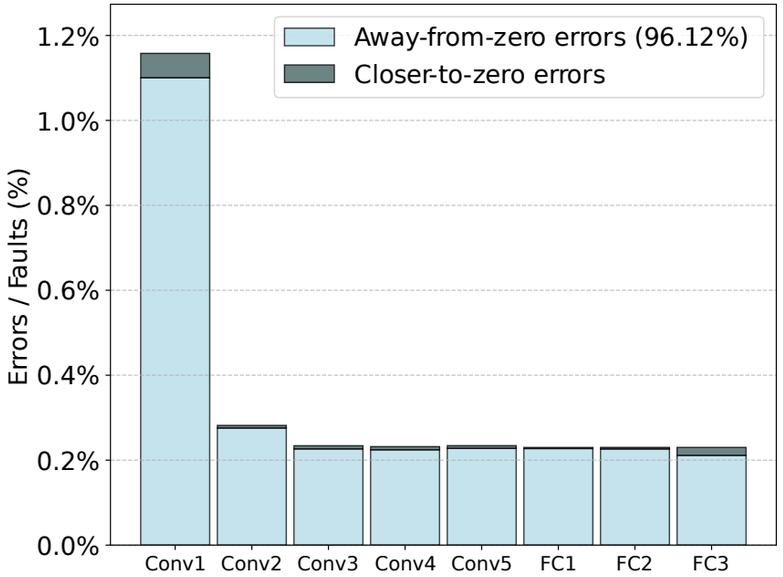
SFI4NN: Statistical Fault Injection for Neural Networks



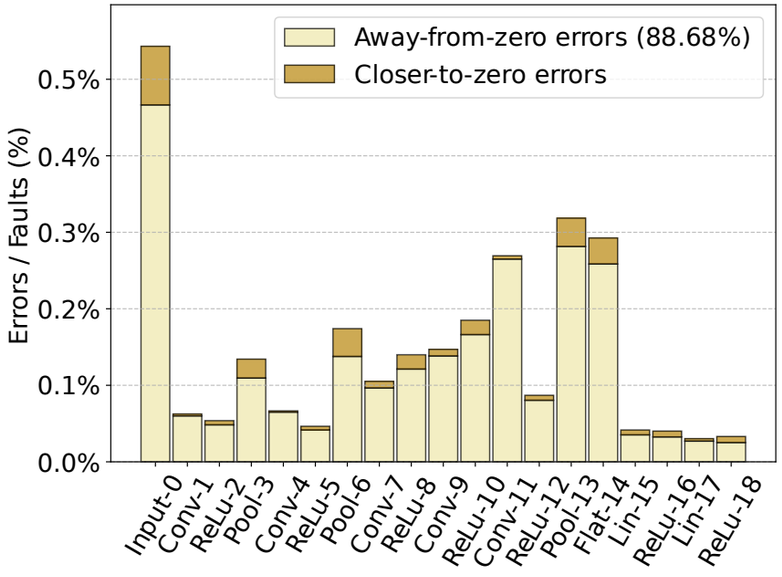
- Targets weights and intermediate data
- Provides layer and bit-level granularity
- Supports fixed-point arithmetic
- Uses random fault sampling
- Handles fault direction (0 → 1 vs. 1 → 0)
- Enables CNN sensitivity analysis

SFI4NN: Statistical Fault Injection for Neural Networks

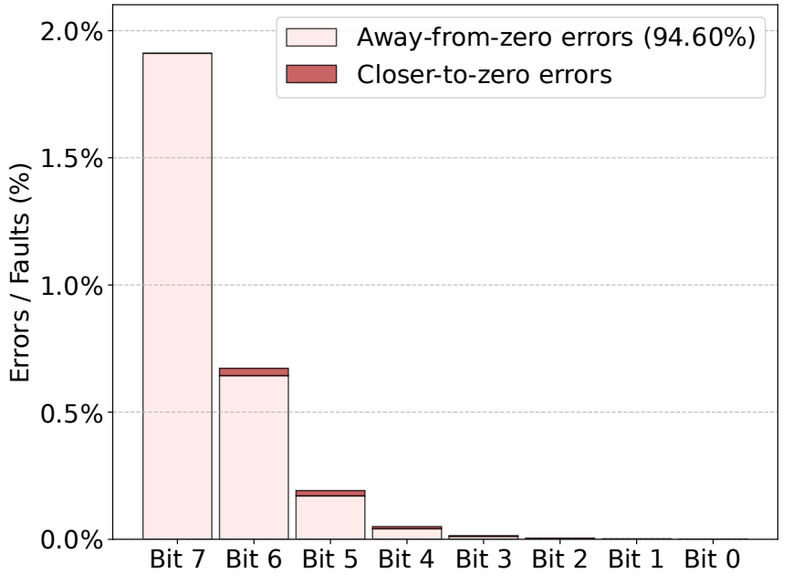
Weights (layers)



Activations (layers)



Bits position



- **AlexNet sensitivity analysis using SFI4NN**
  - 10% error margin
  - 90% confidence level
  - 8-bit fixed-point model
  - CIFAR-10, 10k test images

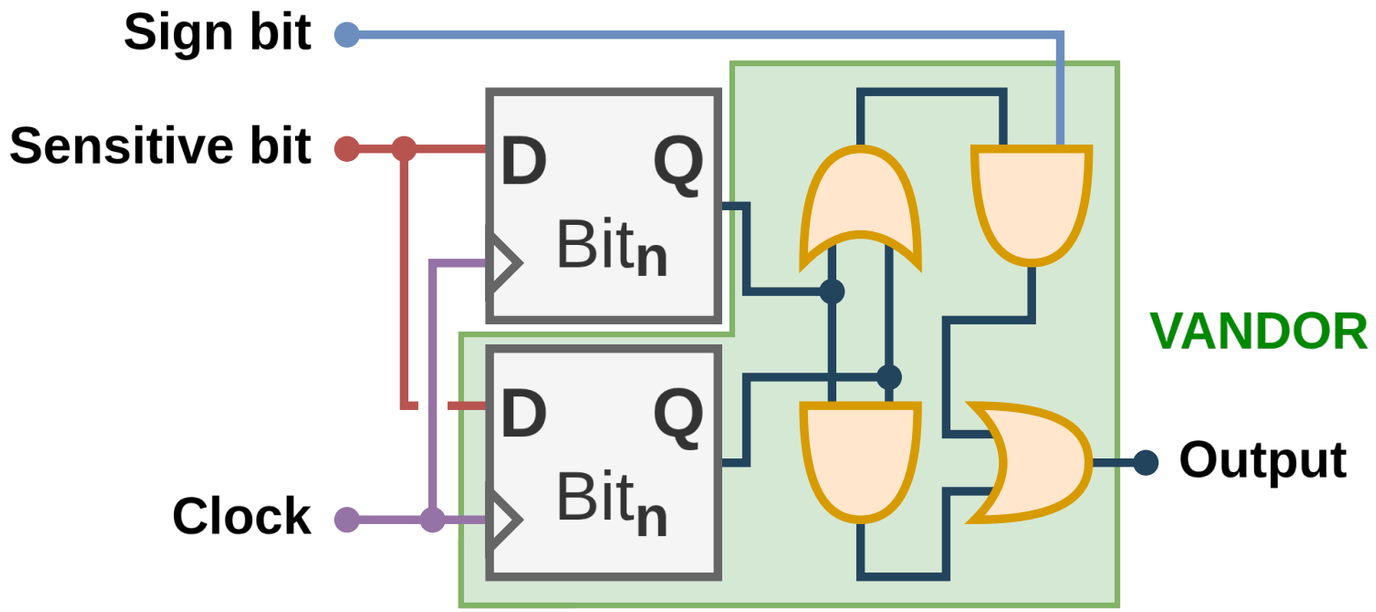
Most NN mispredictions are caused by away-from-zero errors

VANDOR: Voter block with AND/OR gates [11]

- **Closer-to-zero faults**
  - Have a lower impact on neural network predictions
  - Selecting the value closer to zero can reduce fault effects
- **Duplication**
  - Detects faults when duplicated data differ
  - Correction is not possible (no majority voting as in TMR)
- **VANDOR concept**
  - Duplicates data and selects the value closer to zero upon fault detection
  - Enables lightweight, bit-level fault mitigation

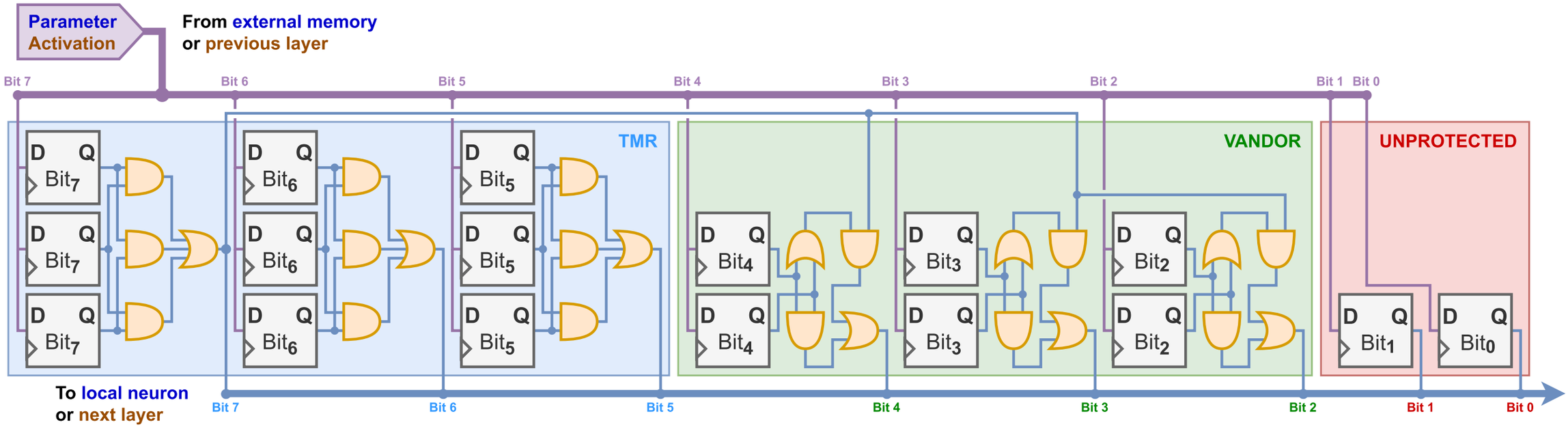
[11] Guillemé et al. (IEEE IOLTS, 2024). VANDOR: Mitigating SEUs into Quantized Neural Networks.

VANDOR: Voter block with AND/OR gates [11]



- Triplicated sign bit enables valid correction
- One additional D Flip-Flop
- Mismatch between sensitive and duplicated bits → fault detection
- Acts as an AND gate for positives (sign bit = 0), and as anOR gate for negatives (sign bit = 1)

$T_3V_3U_2$ implementation for an 8-bit data word



- Triplication on the most significant bits
- VANDOR on intermediate bits
- Least significant bits left Unprotected
- Each layer can have a different level of TVU protection

Cost of TVU strategies in DFFs per 8-bit data word

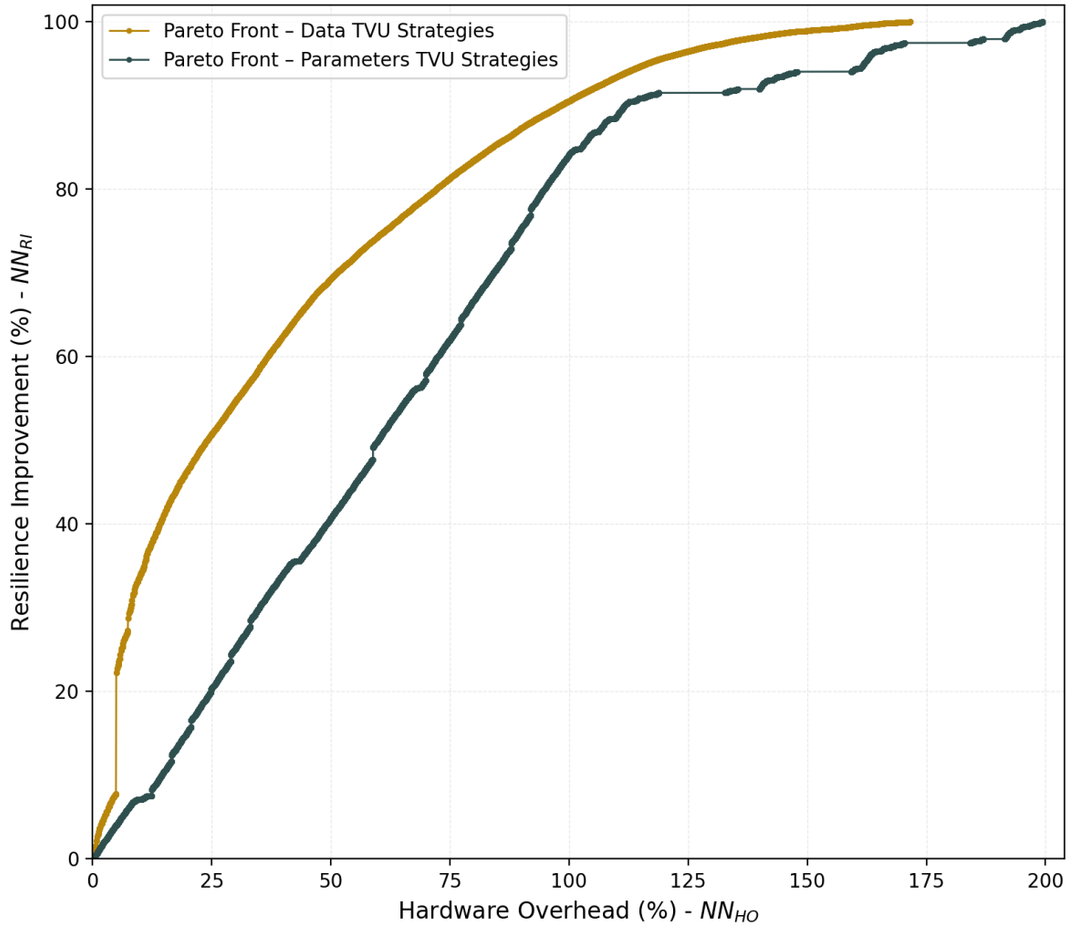
Layer protection TVU strategy	TMR / VANDOR / UNPROTECTED	DFF Overhead per Data
T0-V0-U8	Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0	+0 DFF
T1-V7-U0	Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0	+9 DFF
T3-V3-U2	Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0	+9 DFF
T8-V0-U0	Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0	+16 DFF

- 37 TVU strategies for an 8-bit data word
- T1V7U0 and T3V3U2 → same DFF overhead, different protection efficiency
- Embedded systems have limited hardware → full triplication not feasible

Pareto-optimal TVU strategies for an AlexNet use case

- **Total possible configurations**
  - 37 configurations per layer
  - 8 parameter layers → 37^{8}
  - 19 activation layers → 37^{19}

- **Pareto-optimal configurations**
  - **Weight** layers: 114,120
  - **Activation** layers: 4,007



Pareto-optimal TVU strategies for an AlexNet use case

Weight TVU strategies for selected NN_{HO} targets

NN_{HO} (%)	NN_{RI} (%)	CONV-1	CONV-2	CONV-3	CONV-4	CONV-5	FC-1	FC-2	FC-3
20	15.22	T8V0U0	T1V4U3	T1V5U2	T0V0U8	T1V2U5	T1V2U5	T0V0U8	T1V2U5
50	40.70	T1V6U1	T1V2U5	T1V5U2	T1V4U3	T1V3U4	T0V0U8	T1V4U3	T0V0U8
100	84.08	T0V0U8	T1V1U6	T1V5U2	T1V7U0	T1V7U0	T1V6U1	T1V6U1	T1V0U7
147.85	94.04	T8V0U0	T8V0U0	T8V0U0	T7V1U0	T7V1U0	T8V0U0	T1V7U0	T8V0U0
170.39	97.47	T8V0U0	T8V0U0	T8V0U0	T7V1U0	T7V1U0	T1V7U0	T8V0U0	T8V0U0

Weight TVU strategies for selected NN_{RI} targets

NN_{HO} (%)	NN_{RI} (%)	CONV-1	CONV-2	CONV-3	CONV-4	CONV-5	FC-1	FC-2	FC-3
14.49	10	T1V3U4	T0V0U8	T1V5U2	T0V0U8	T0V0U8	T1V1U6	T0V0U8	T1V0U7
29.89	25	T0V0U8	T1V1U6	T0V0U8	T0V0U8	T1V1U6	T1V5U2	T0V0U8	T0V0U8
59.95	50	T1V3U4	T0V0U8	T0V0U8	T0V0U8	T1V2U5	T0V0U8	T1V6U1	T1V1U6
89.66	75	T5V2U1	T1V3U4	T0V0U8	T0V0U8	T1V4U3	T1V5U2	T1V6U1	T1V0U7
111.66	90	T3V4U1	T1V5U2	T1V5U2	T1V7U0	T1V7U0	T1V7U0	T1V7U0	T0V0U8

CONCLUSION

SFI4NN framework

- Provides a statistical fault injection approach for fixed-point quantized neural networks
- Quantifies and analyzes the impact of bit-flip directionality
- Operates at layer- and bit-level granularity
- Targets both parameters and activations

TVU protection

- Selective protection strategies that account for neural network behavior
- Fine-grained protection down to individual flip-flops
- Improves reliability with minimal hardware cost

Thanks for your attention.