STOCHASTIC DATA-DRIVEN HARDWARE RESILIENCE TO EFFICIENTLY TRAIN INFERENCE MODELS FOR STOCHASTIC HARDWARE IMPLEMENTATIONS

Bonan Zhang (bonanz@princeton.edu),
Lung-Yen Chen, and Naveen Verma

Princeton University

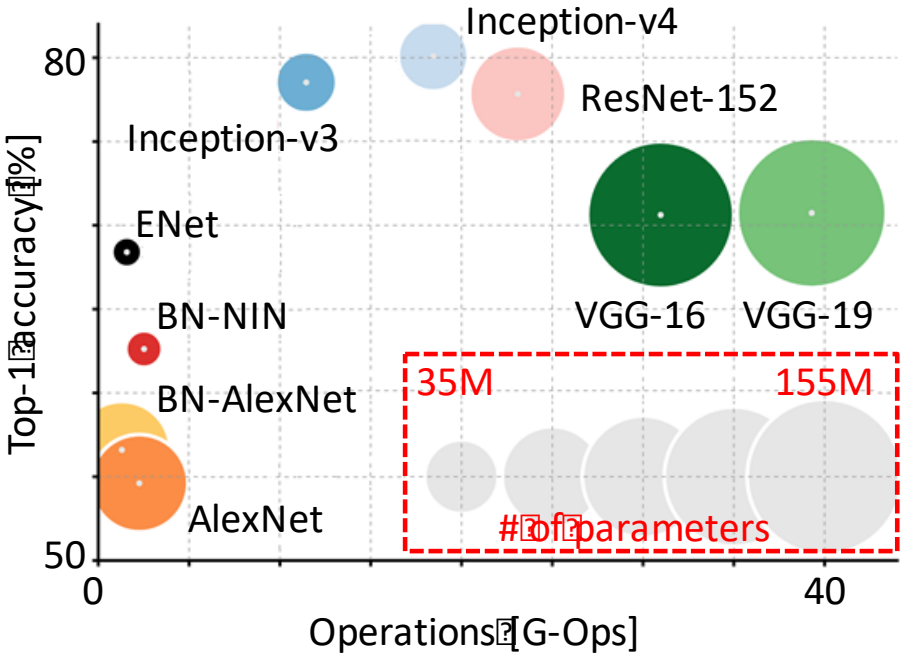
ICASSP 2019

Outline

- **Deep-learning inference drives systems to energy limits**
  - Energy-aggressive technologies/architectures affected by variations
- **Background**
  - Data-driven hardware resilience (DDHR)
  - Error Adaptive Classifier Boosting (EACB)
- **3. <u>This work:</u> Stochastic DDHR (S-DDHR)**
  - Training to a stochastic distribution of variation-affected hardware
- **4. Simulations and Results**
  - Stochastic model for MRAM-based in-memory computing
  - Train once for all variation-affected hardware
  - < 3% accuracy loss (CIFAR-10) vs. variation-free hardware
- **5. Conclusions**

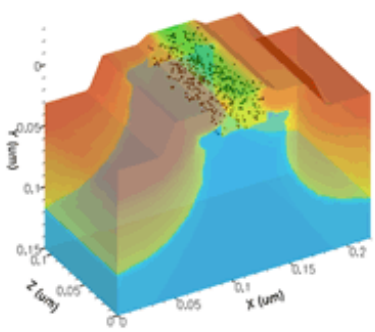
Pushing Energy-efficiency of DL Systems

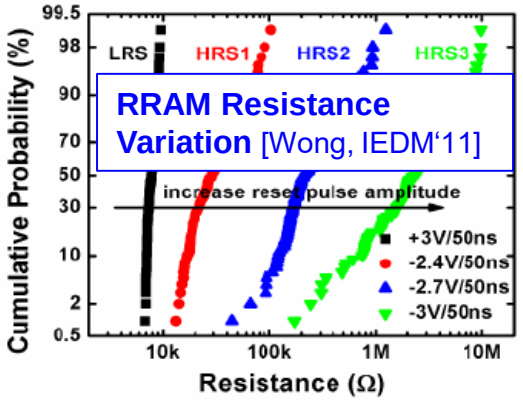
- **Energy-constrained applications limited in adoption of DL**

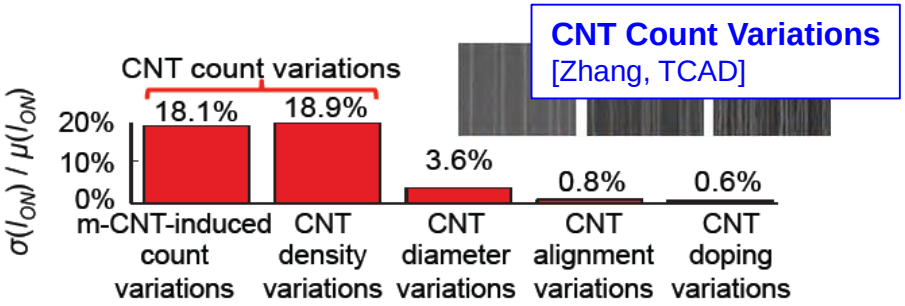


[*Canziani, Arxiv16'*]

- **Variation increasingly prominent in advanced/emerging technologies**





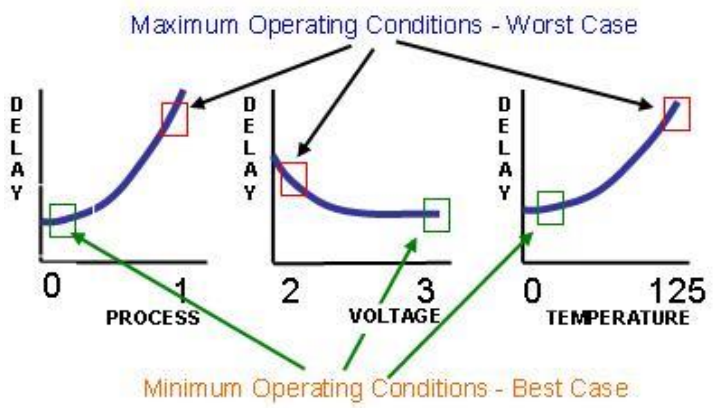


The statistical nature of machine learning algorithms open up new opportunities for energy efficiency and throughput.

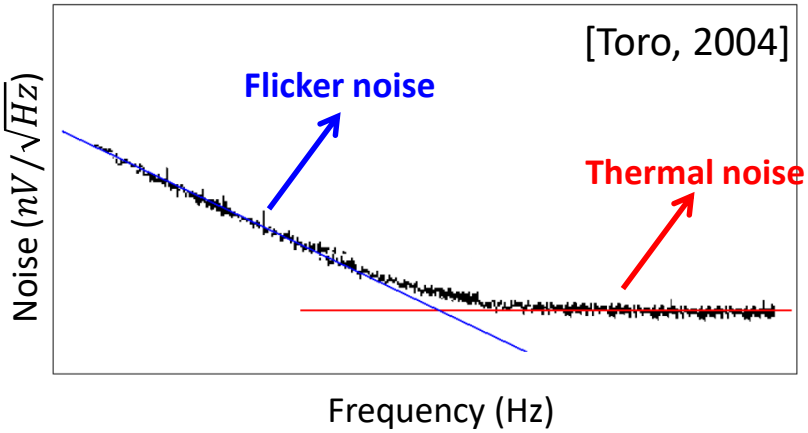
Hardware Variability

Types of variations

- **Static: e.g., process variations**

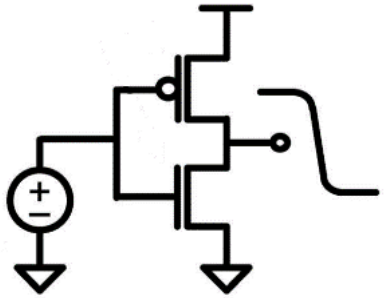


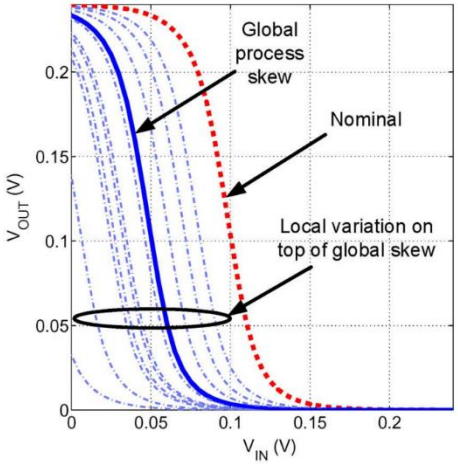
- **Dynamic: electronic noise**



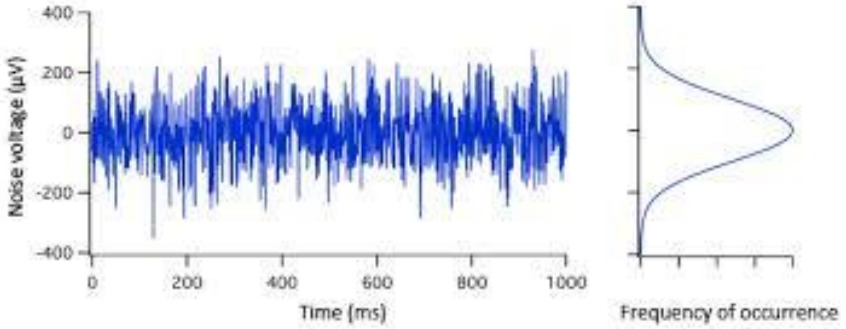
Effects on circuit functionality

- **Fixed error in given circuit's operation**

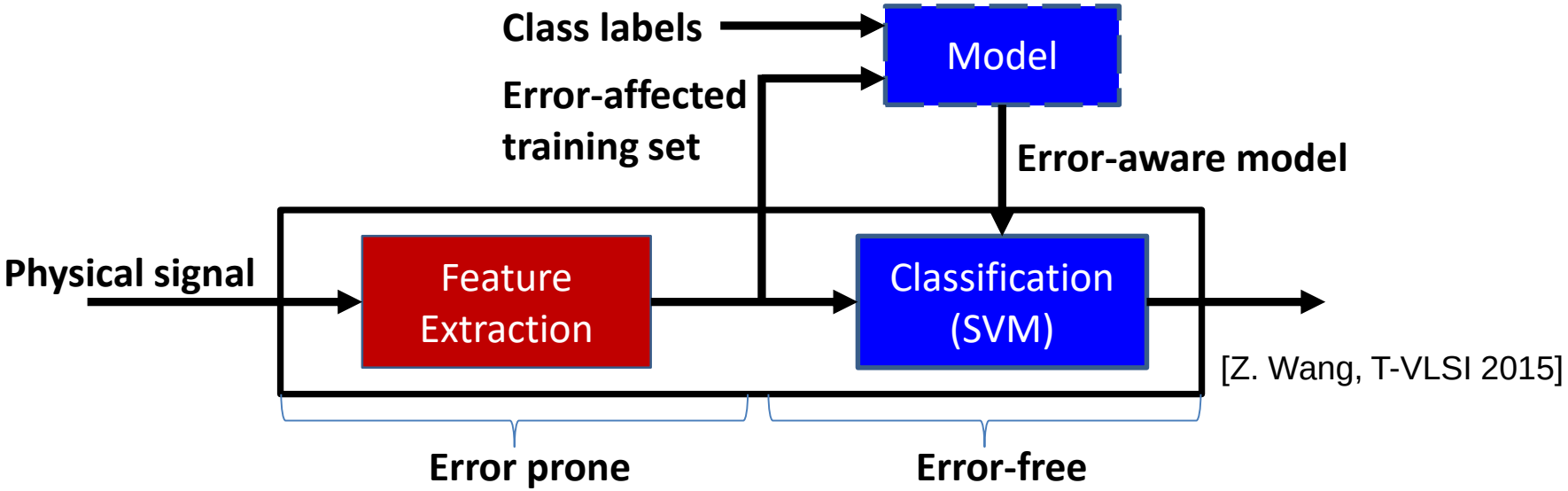




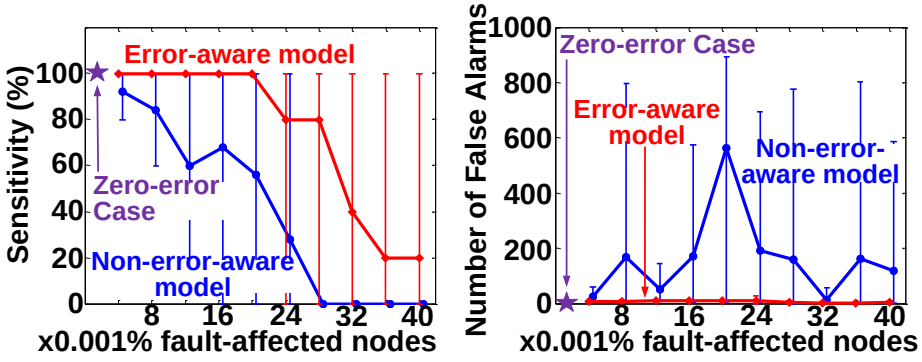
- **Continuously-changing error in given circuit's operation**



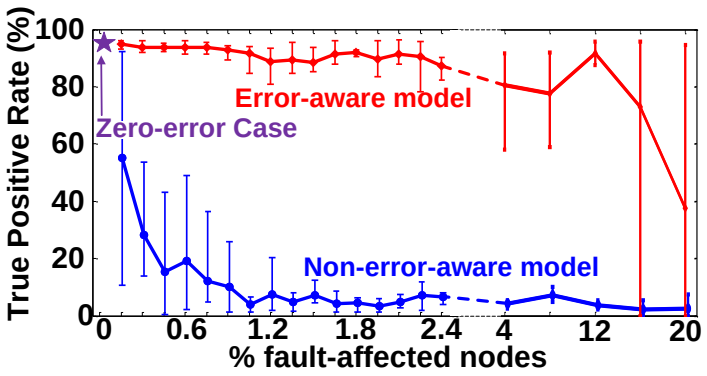
Data-driven Hardware Resilience (DDHR)



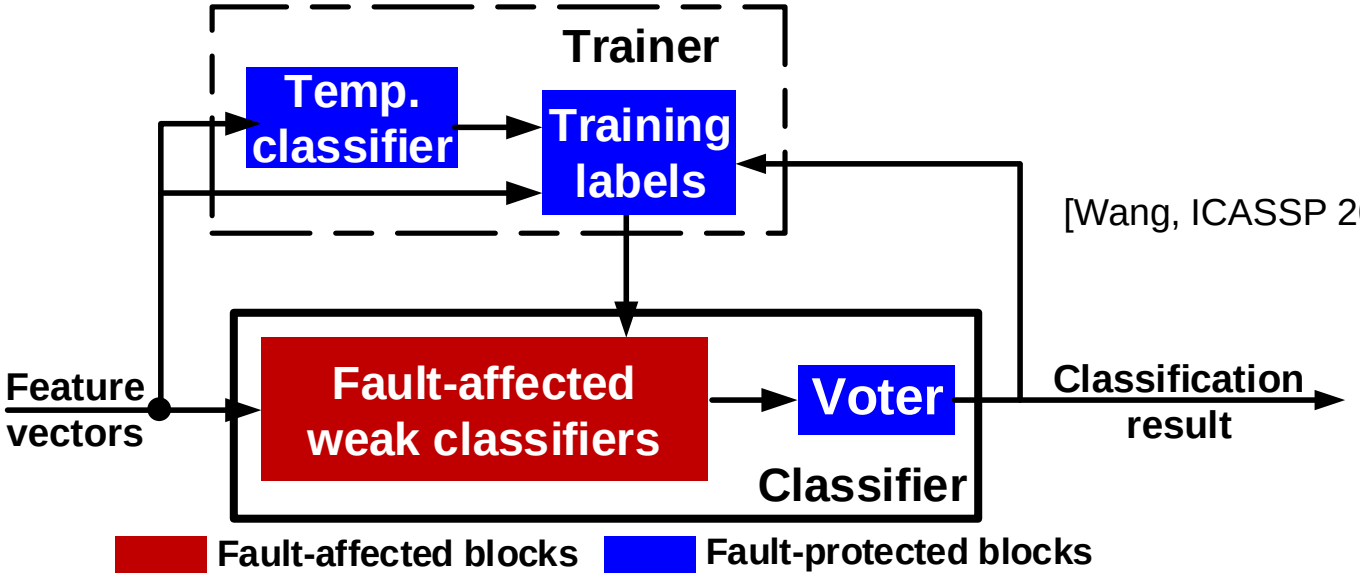
Ex. 1: EEG-based Seizure Detector (BER 10%~45%)



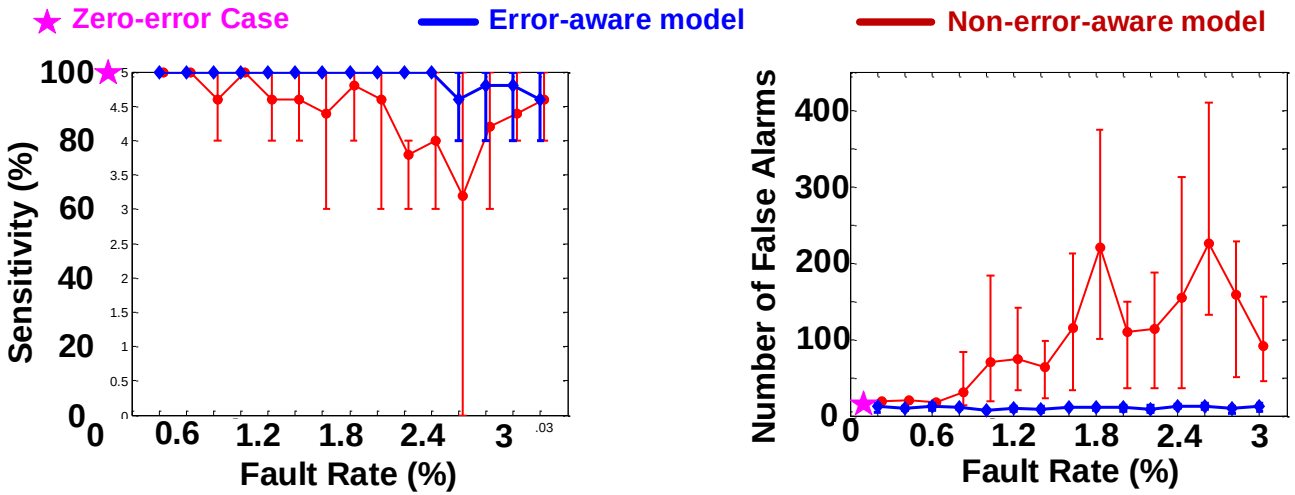
Ex. 2: ECG-based Arrhythmia Detector (BER 20%~50%)



Error Adaptive Classifier Boosting (EACB)

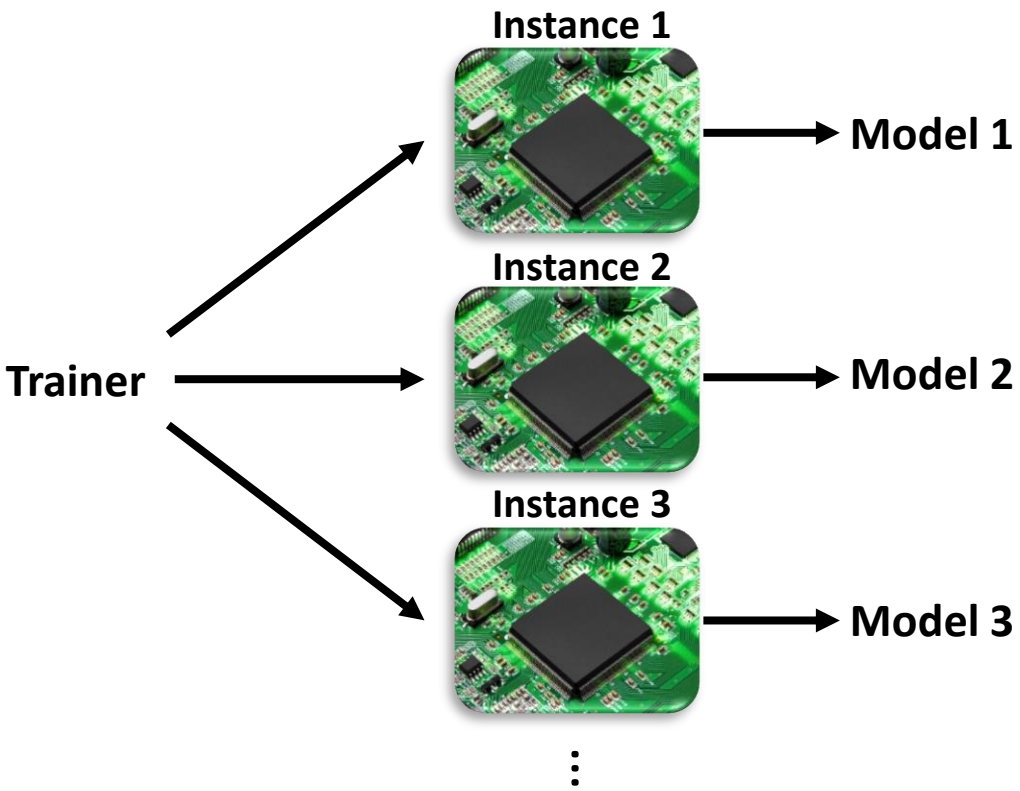


Ex: EEG-based Seizure Detection

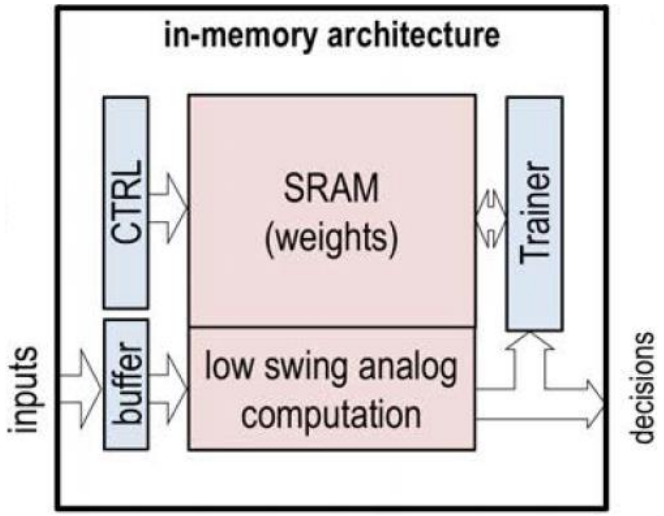


Limitation

- **Require training complexity for each instance of hardware**



Ex: Embedded training hardware

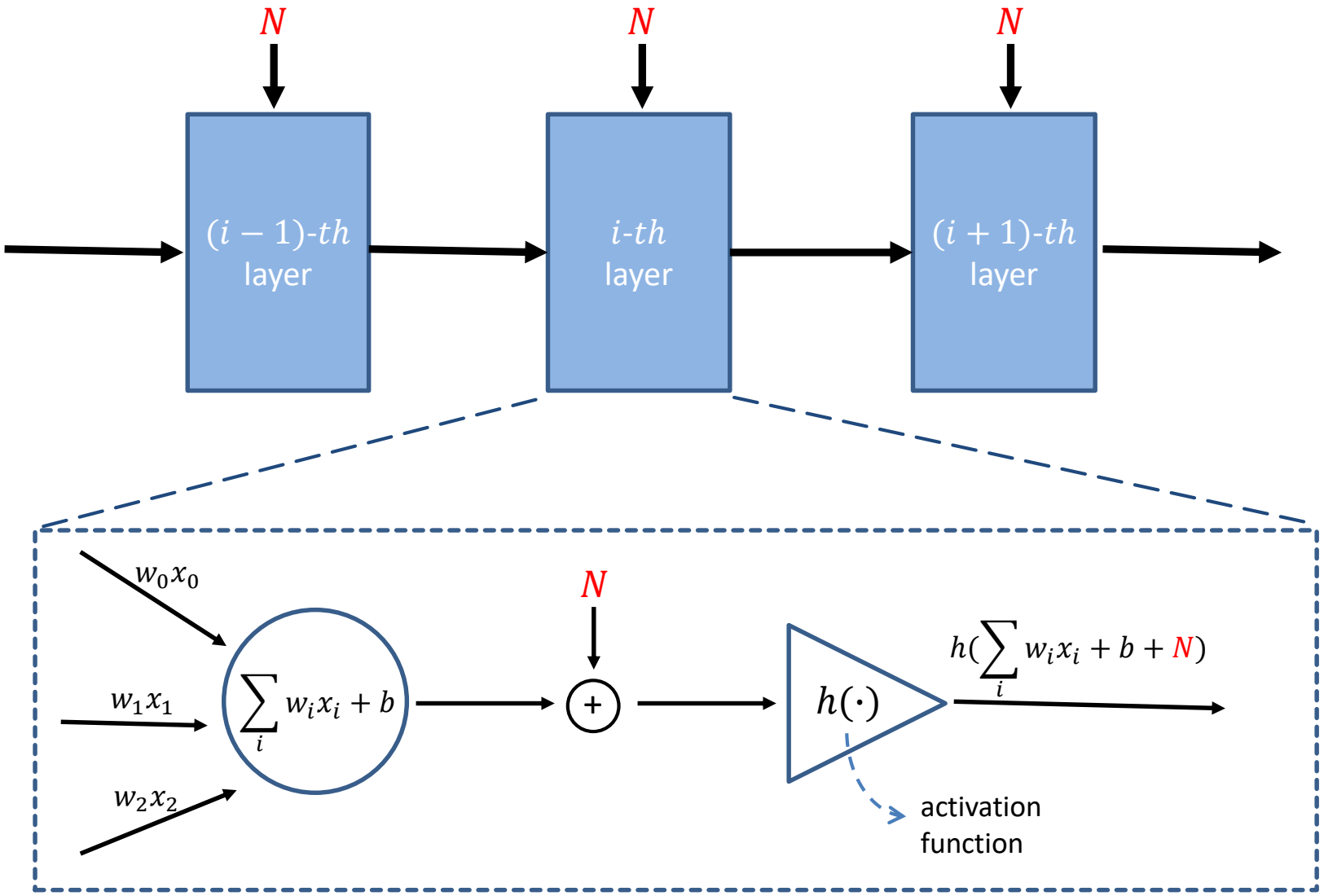


[S. Gonugondola, *ISSCC'*18]

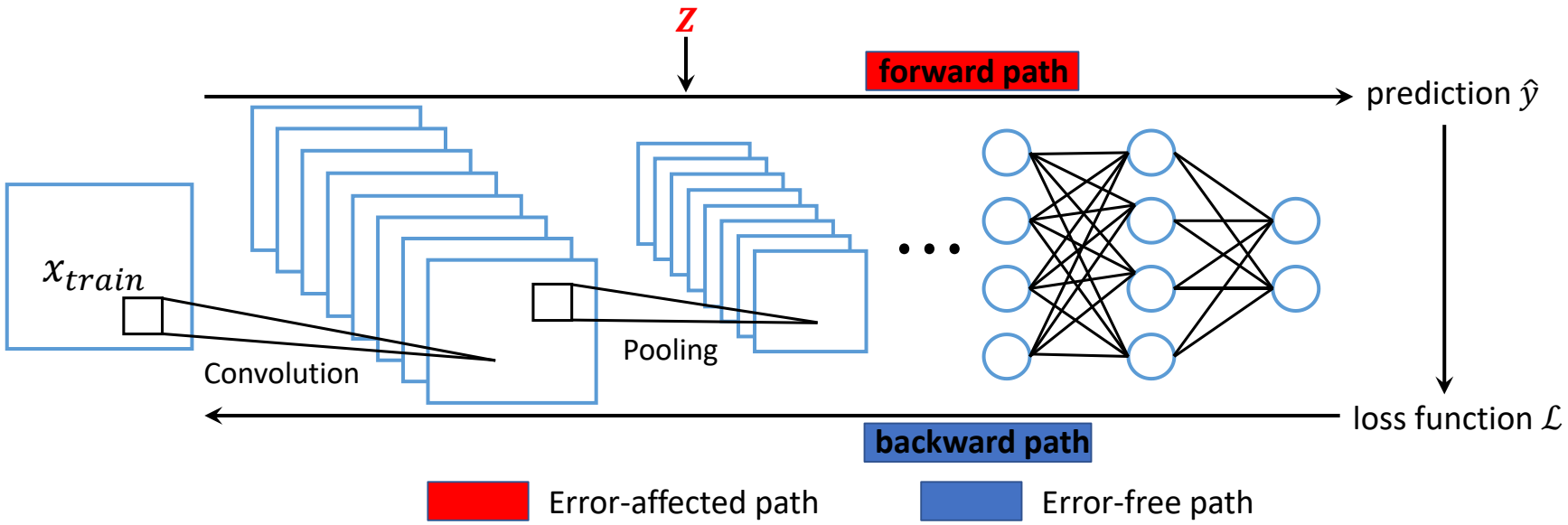
Objective: no training complexity with each hardware instance

Stochastic Neural-network Training

Stochastic training model:



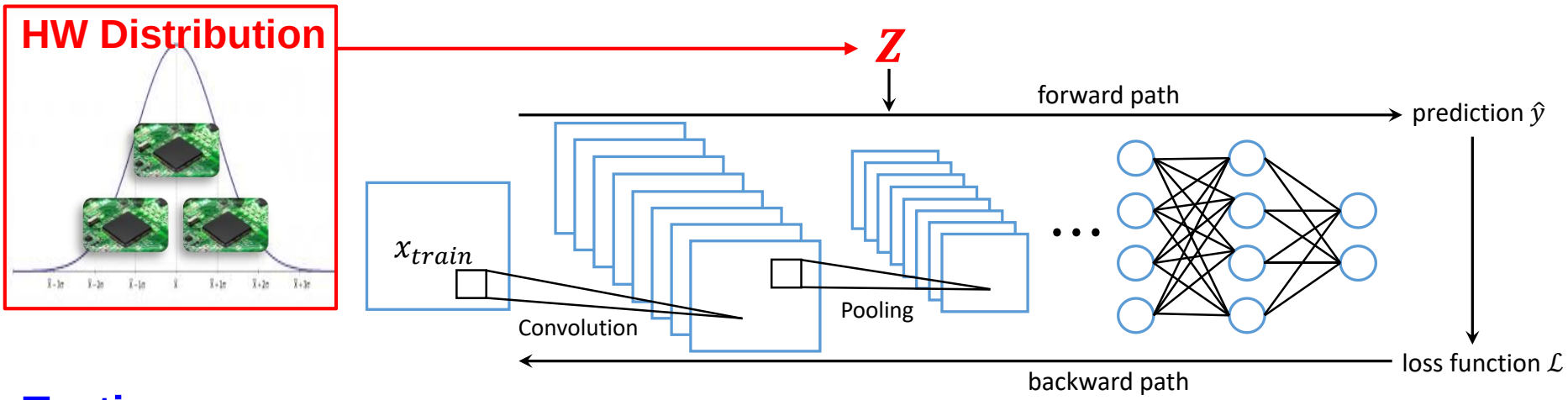
Stochastic DDHR (S-DDHR)



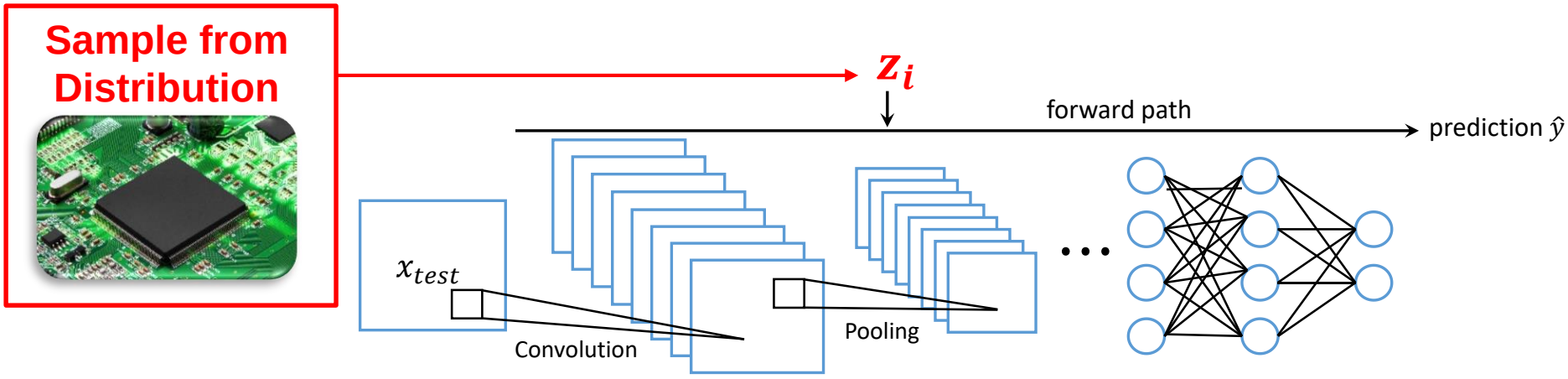
- **Introduce randomness to the forward path**
- ***Z*** **is a random variable representing hardware variation**
- **Both feature extraction stages (conv layers) and classification stages (fc layers) are error-affected**

Stochastic DDHR (S-DDHR)

Training:



Testing:



Comparison between Different Models

Cross entropy loss for the t^{th} training sample:

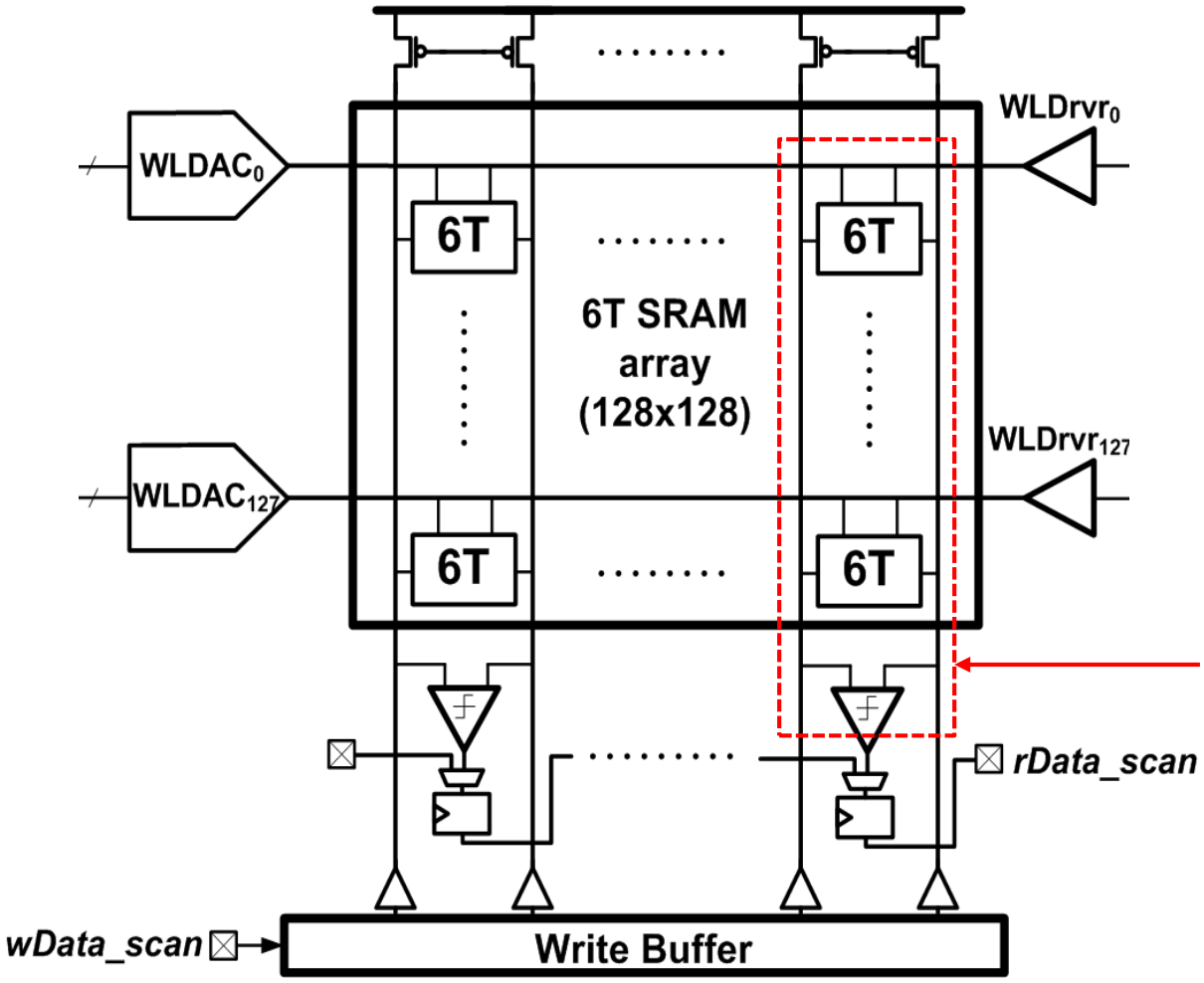
1. **Variability-free model:**
   - $\mathcal{L}^{t}_{Ideal} = -\sum_{i=1}^{C} y_i^t \log f(x^t, \theta)$
2. **DDHR model:**
   - $\mathcal{L}^{t}_{DDHR} = -\sum_{i=1}^{C} y_i^t \log f(x^t, \theta, z_i)$
3. **Training with a fixed instance of hardware:**
   - $\mathcal{L}^{t}_{Fixed} = -\sum_{i=1}^{C} y_i^t \log f(x^t, \theta, z_\emptyset)$
4. **S-DDHR model:**
   - $\mathcal{L}^{t}_{S-DDHR} = -\sum_{i=1}^{C} y_i^t \log f(x^t, \theta, Z)$

Assume:
- C: number of classes
- θ: model parameter
- Z: random variable
- z_i and $z_\emptyset$: different samples

Variants of S-DDHR model:

1. **Parametric S-DDHR (PS-DDHR)**
   - Employs an analytical distribution for Z
   - Compute distribution parameters for each layer
2. **Approximate S-DDHR (AS-DDHR)**
   - Fixed distribution parameters for all layers

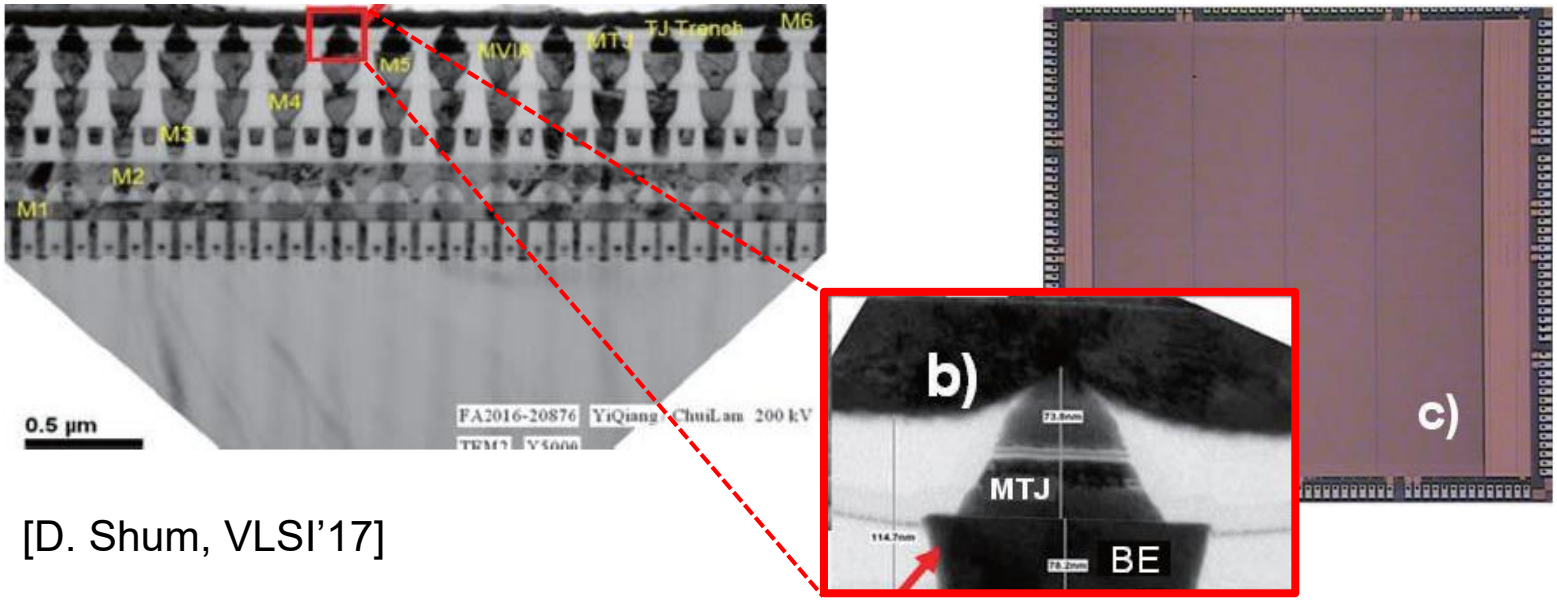
Evaluation: in-memory computing



Binarized Neuron → large-dynamic range

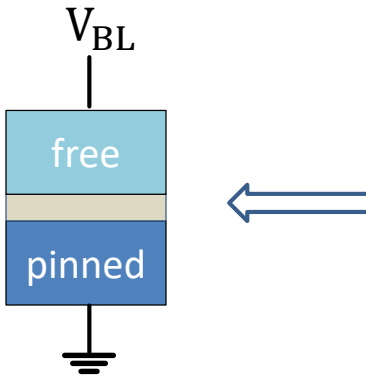
[J. Zhang, *JSSC*'17]

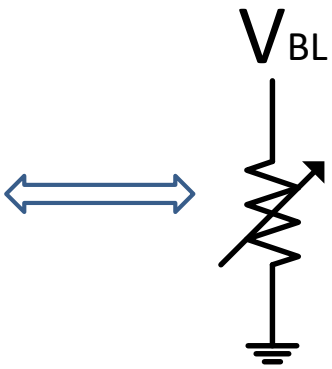
Evaluation: magnetic RAM (MRAM)



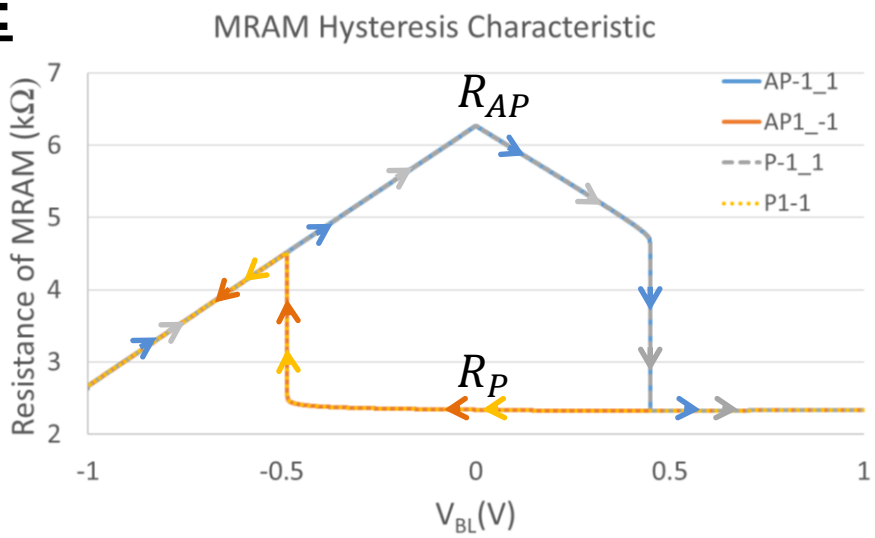
[D. Shum, VLSI'17]

Magnetic Tunneling Junction (MTJ):

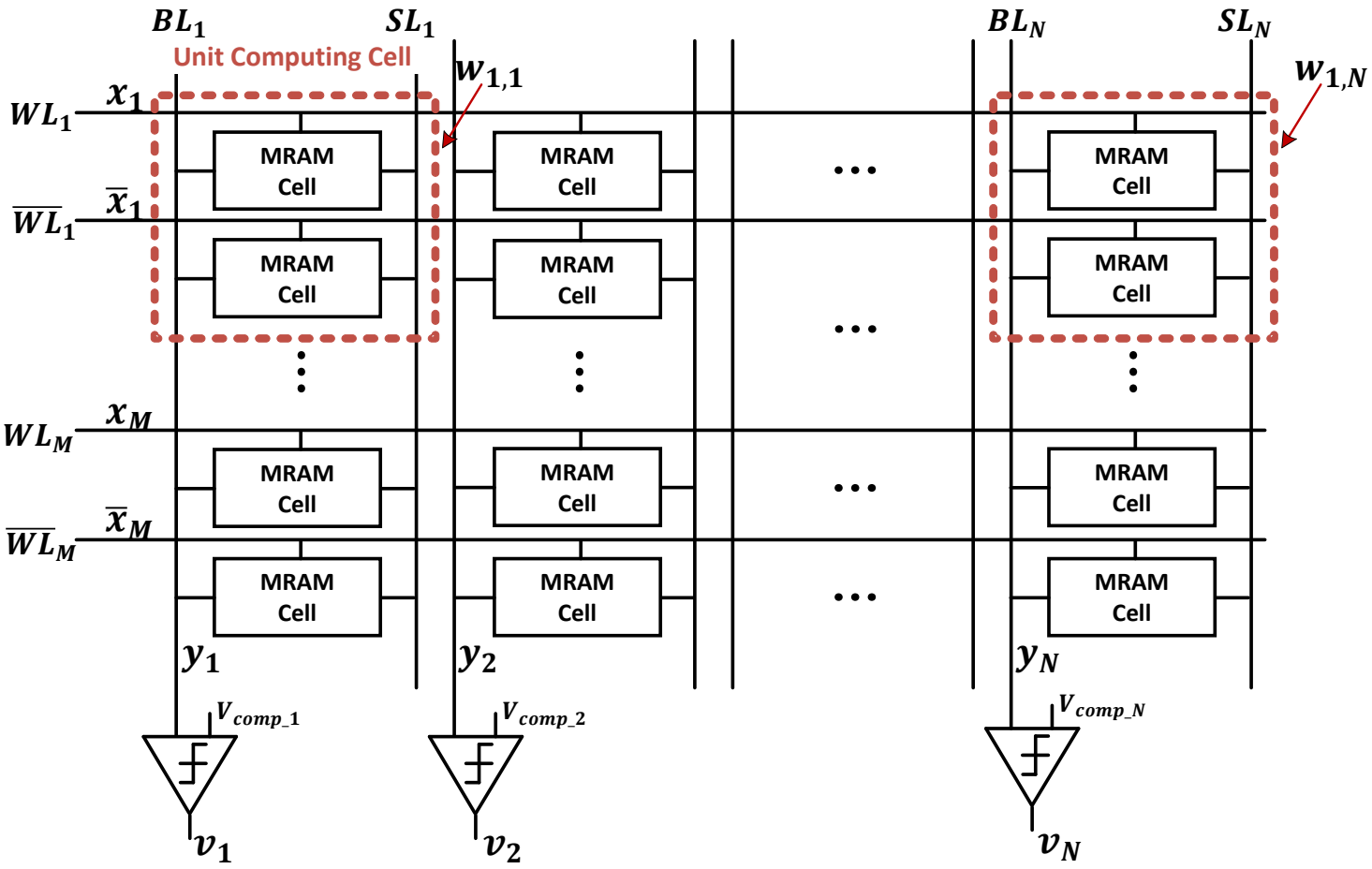




hysteresis



MRAM-based In-memory Computing Archietcture



- $W_{m,n} \in \{-1,1\}$, $x_m \in \{-1,1\}$
- Two complementary MRAM cells as a unit computing cell, with complementary word line inputs ($WL_m/\overline{WL_m}$)

Distribution Modeling

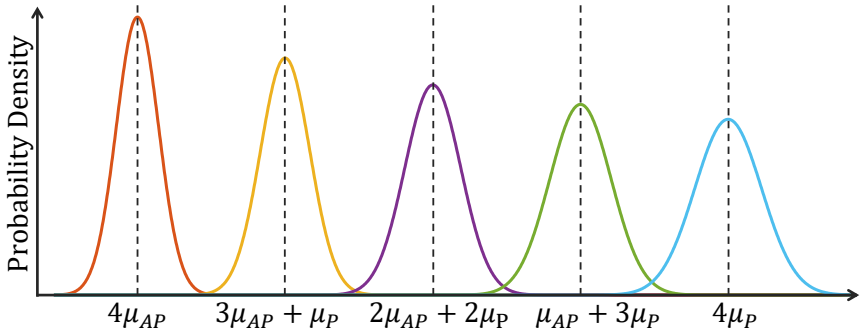
Stochastic model:

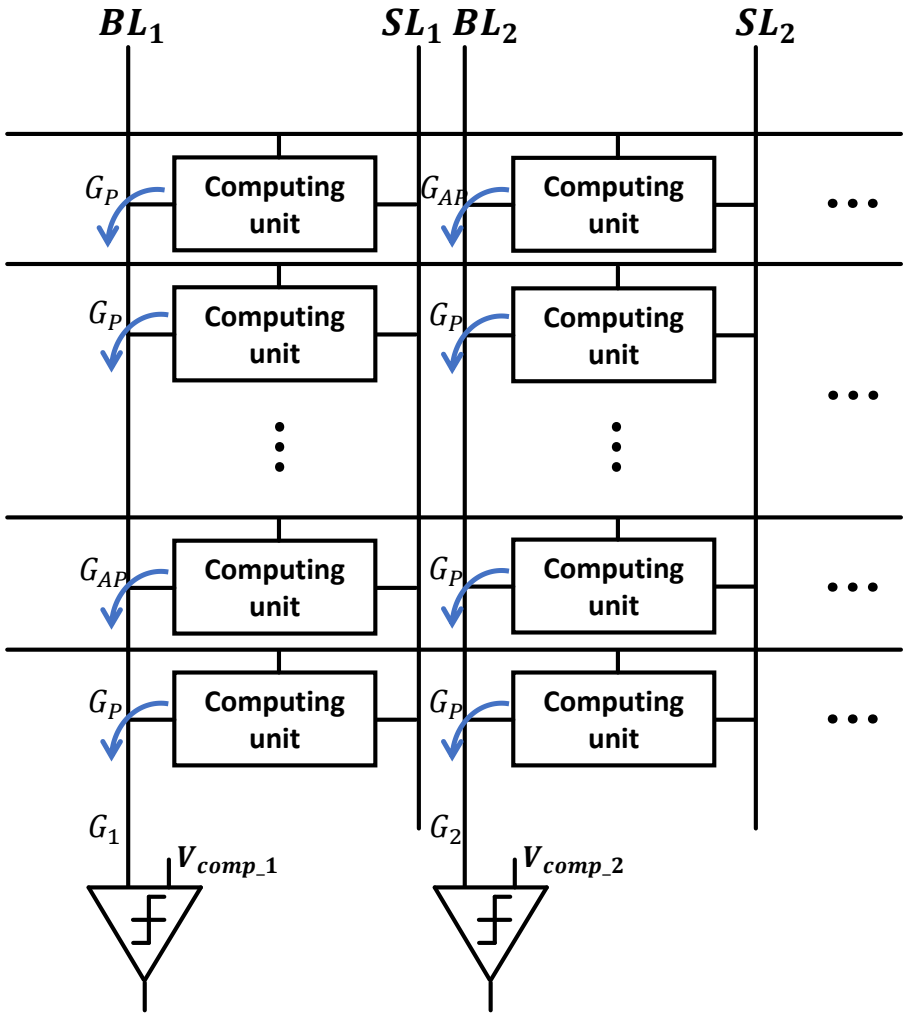
- $R_P \sim N(\mu_P, \sigma_P^2)$, $R_{AP} \sim N(\mu_{AP}, \sigma_{AP}^2)$
- $G_P \sim N(\frac{1}{\mu_P}, \frac{\sigma_P^2}{\mu_P^4})$, $G_{AP} \sim N(\frac{1}{\mu_{AP}}, \frac{\sigma_{AP}^2}{\mu_{AP}^4})$

Computation model:

- $Y = W \times X$
- $y_{m,n} = p_{1m,n} \times G_P + p_{-1m,n} \times G_{AP}$
  ($p_{1m,n}/p_{-1m,n}$: number of element-wise product terms equal to 1/-1)

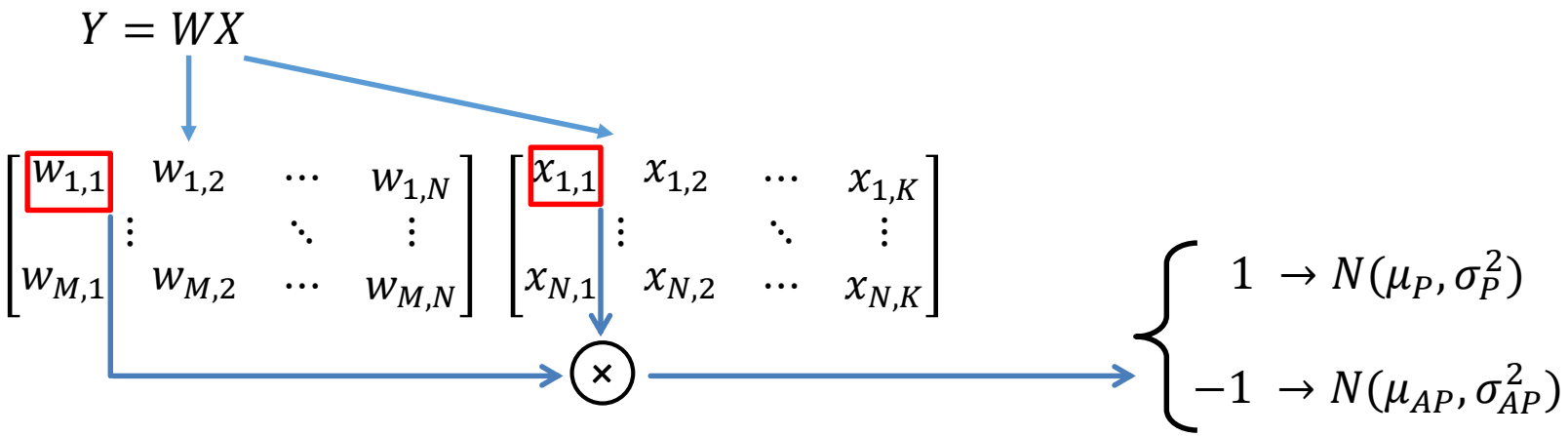
Ex: y_n with four rows:



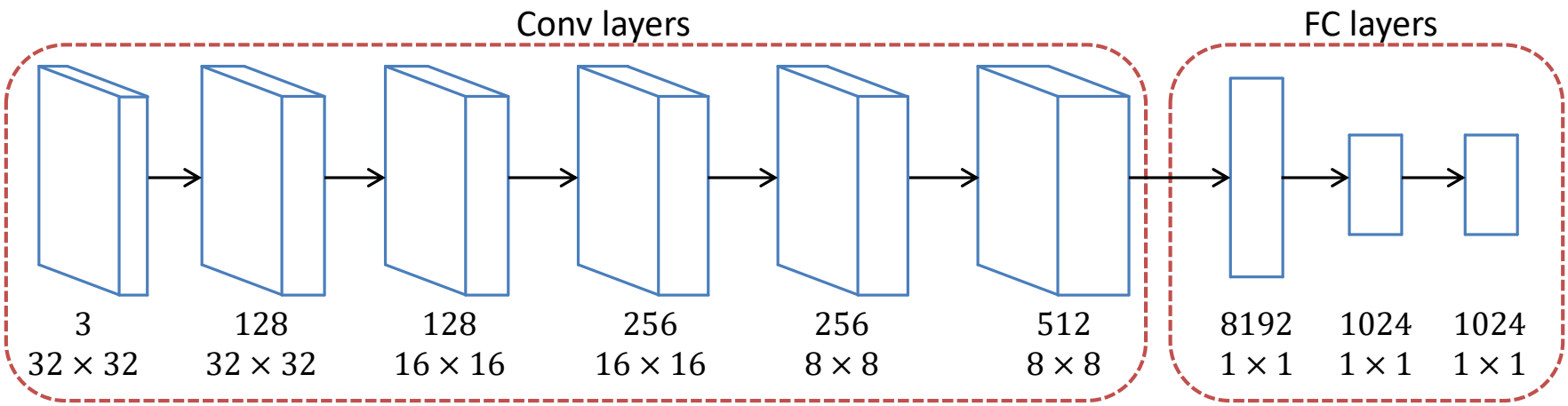


Training/Testing Framework

Implementation of matrix multiplication :



Architecture of binarized CNN based on BinaryNet:



Simulation Results

$\mathcal{L}_{DDHR}$

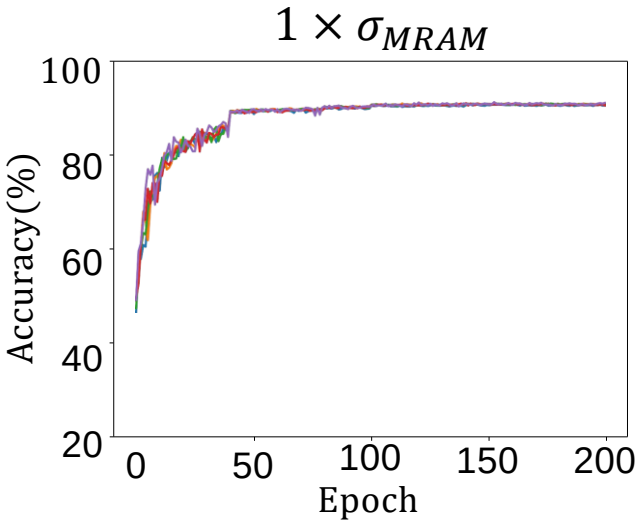
$\mathcal{L}_{PS-DDHR}$

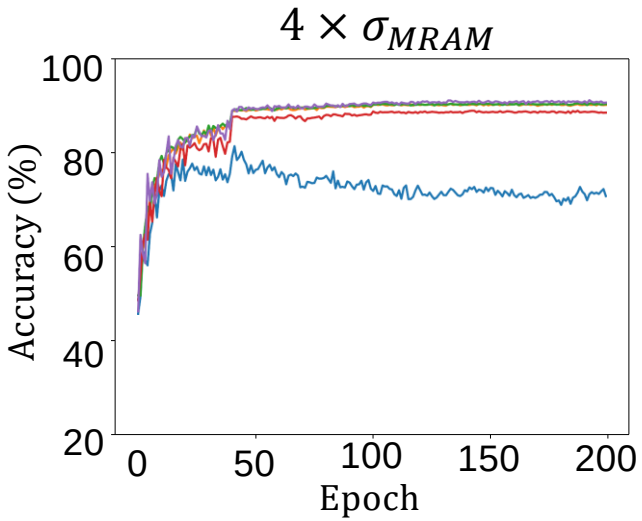
$\mathcal{L}_{AS-DDHR}$

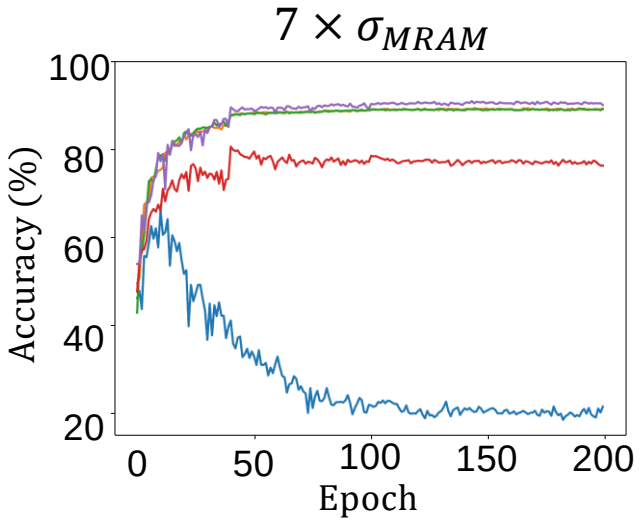
$\mathcal{L}_{Ideal}$

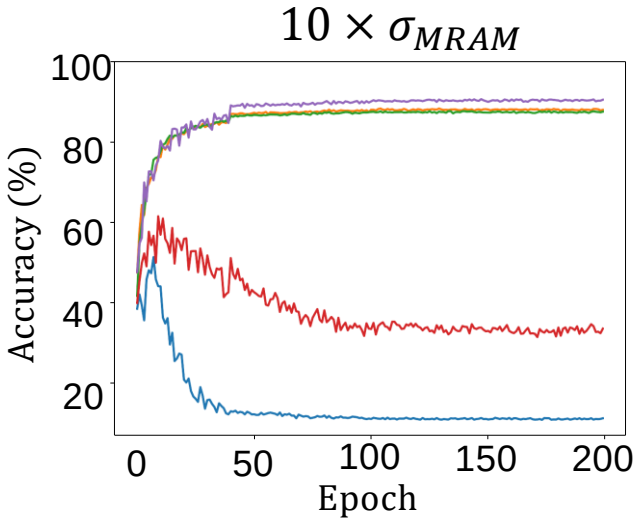
$\mathcal{L}_{Fixed}$

Variation levels normalized to actual MRAM variability σ_{MRAM}

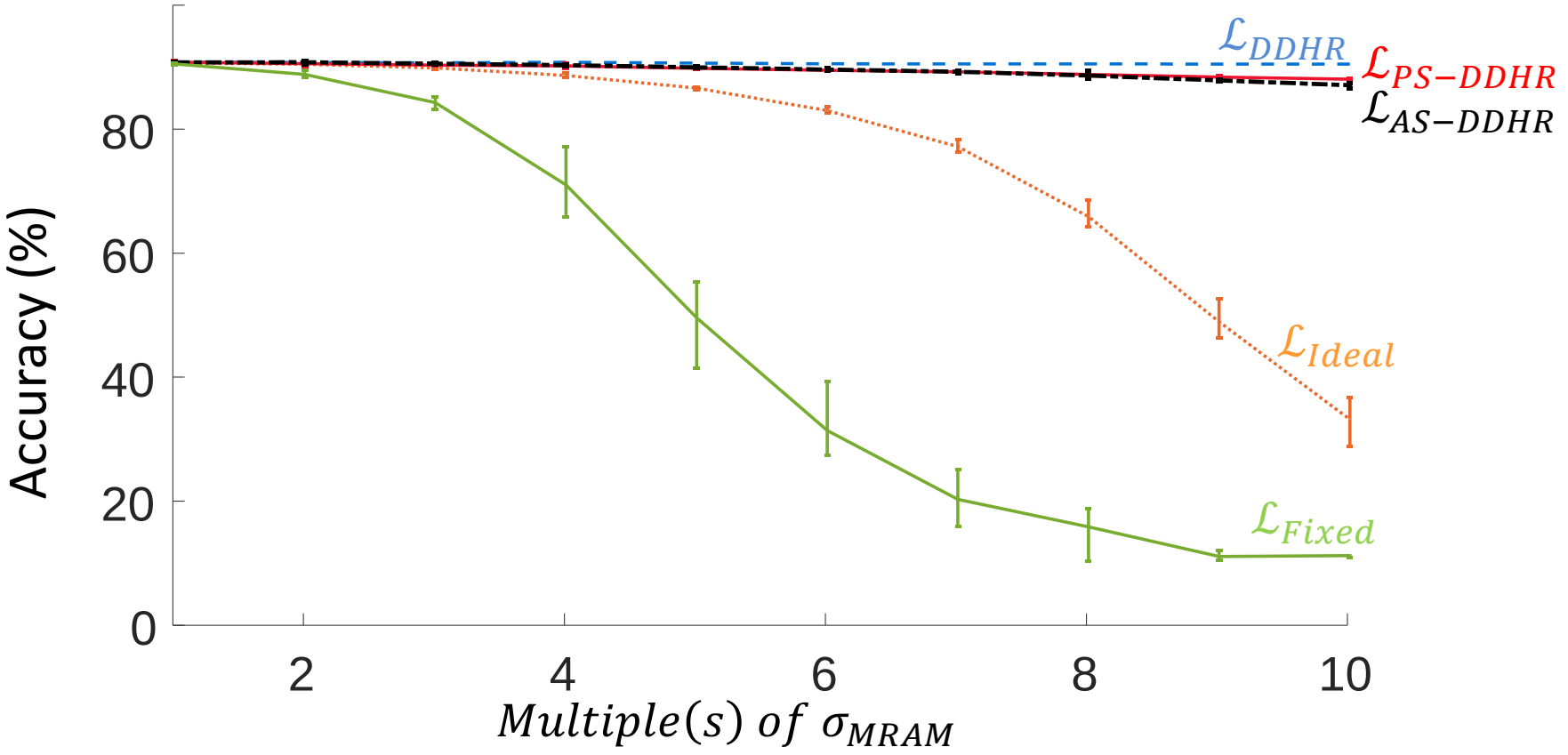








CIFAR-10 Image Classification Results



Accuracy at $10 \times \sigma_{MRAM}$:

Model	$\mathcal{L}_{DDHR}$	$\mathcal{L}_{Ideal}$	$\mathcal{L}_{Fixed}$	$\mathcal{L}_{PS-DDHR}$	$\mathcal{L}_{AS-DDHR}$
Accuracy	90.40%	33.18%	11.25%	88.11%	87.42%

Summary and Conclusion

- **Objective:**
  - **Enable machine learning inference with hardware variability**
- **Previous data-driven algorithms are limited by instance-by-instance training**
- **Stochastic DDHR (S-DDHR):**
  - **Construct stochastic model for hardware variability**
  - **Stochastic training for inference-model parameters**
- **Demonstration for BNN on CIFAR-10 dataset, implemented using MRAM-based in-memory computing architecture**
  - **Avoid instance-by-instance training cost**
  - **<3% accuracy degradation from variability-free hardware**

Acknowledgements: This material is based on research sponsored by Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under agreement number FA8650-18-2-7866.