Reliable Memristive Neural Network Accelerators Based on Early Denoising and Sparsity Induction

Anlan Yu, Ning Lyu, Wujie Wen, Zhiyuan Yan

Speaker: Anlan Yu
Electrical and Computer Engineering Department, Lehigh University, USA
December 13, 2021

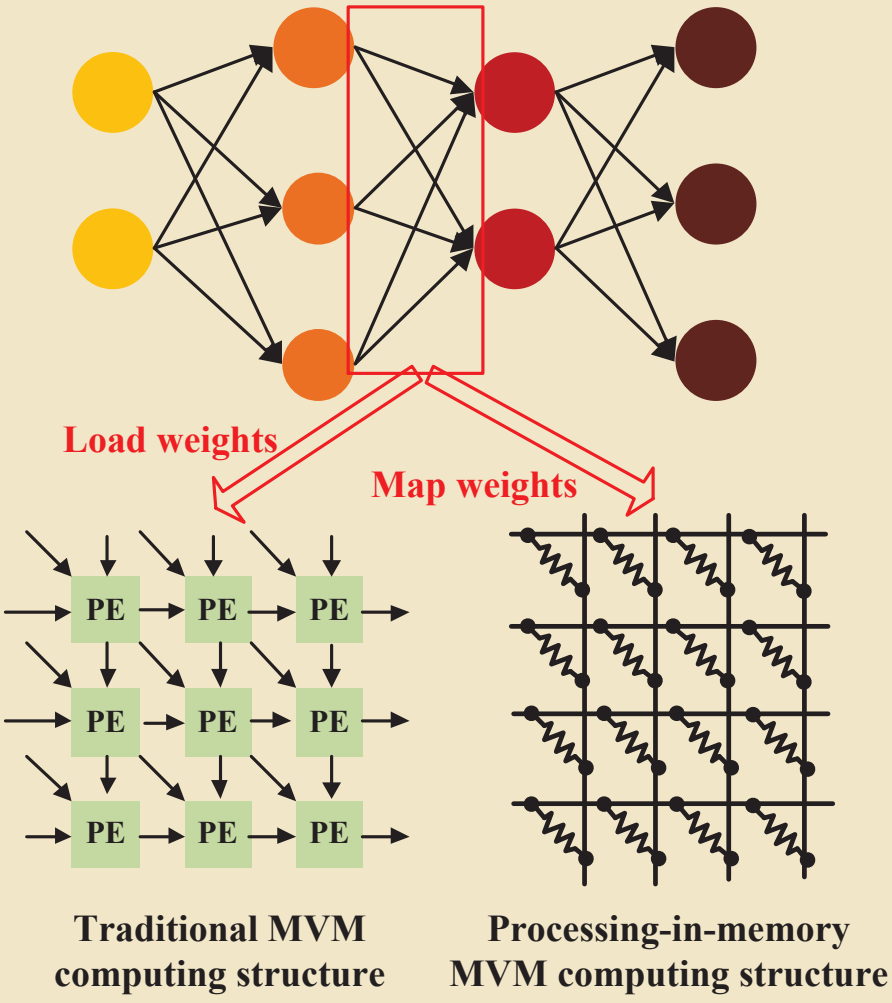
Speaker Bio

Anlan Yu

I received BE degree in information science and engineering department, Southeast University, Nanjing, China in 2018. I'm currently pursuing PhD degree in Department of Electrical and Computer Engineering, Lehigh University, USA. My research interest is about increasing reliability of memristive DNN accelerator.

Outline

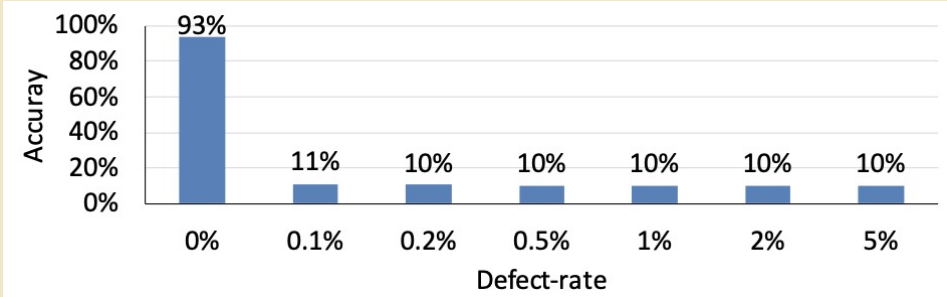
Introduction



Implementing DNN on hardware

1. Traditional MVM computing structure
   - Intensive MVM computation
   - Huge data movement
2. Processing-in-memory (PIM): memristive crossbar
   - No data movement
   - Computation in analog domain

Memristive DNN accelerator suffers from non-ideal effects![1]

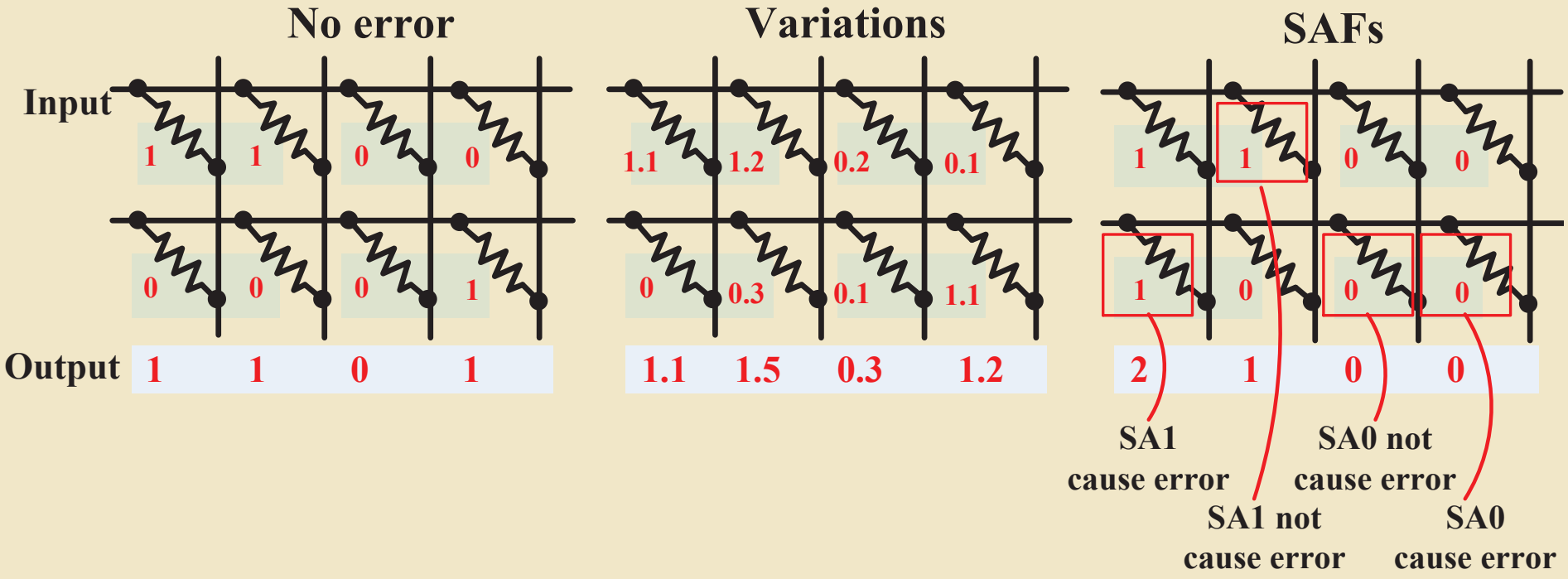


[1] Fateme S Hosseini et al. "Tolerating Defects in Low-Power Neural Network Accelerators Via Retraining-Free Weight Approximation". In: *ACM Transactions on Embedded Computing Systems (TECS)* 20.5s (2021), pp. 1–21.

Introduction

Types of errors for memristive crossbar

1. **Variations**: additive white Gaussian noise (AWGN) on the current output.
2. **Stuck-at-faults (SAFs)**: memristor cells are stuck at certain conductance levels and cannot be programmed.



Introduction

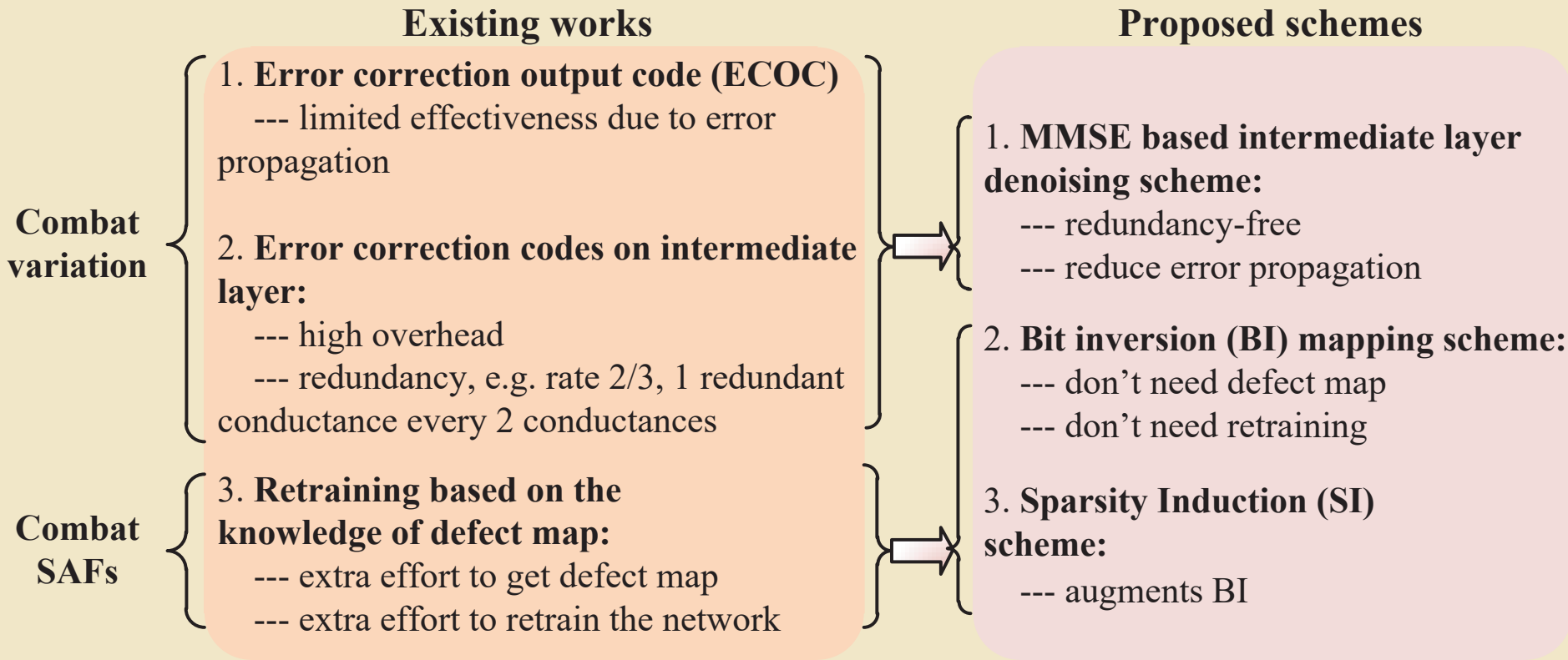


Figure: Existing works[2][3][4] vs our work

[2] Tao Liu et al. "A fault-tolerant neural network architecture". In: *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE. 2019, pp. 1–6.

[3] Qiuwen Lou et al. "Embedding error correction into crossbars for reliable matrix vector multiplication using emerging devices". In: *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*. 2020, pp. 139–144.

[4] Zhezhi He et al. "Noise injection adaption: End-to-end ReRAM crossbar non-ideal effect adaption for neural network mapping". In: *Proceedings of the 56th Annual Design Automation Conference*. 2019, pp. 1–6.

Table of Contents

Proposed Reliable Crossbar Computing Schemes

1. Crossbar output denoising scheme based on MMSE

Linear MMSE estimation:

$$a_{opt}, \mathbf{b}_{opt} = \arg\min_{a,b} \mathbb{E}[(\mathbf{J}_+ - a\mathbf{J}_{+,n} - \mathbf{b})^2], \tag{1}$$

MMSE based denoising:

$$\hat{\mathbf{J}}_+ = a_{opt}\mathbf{J}_{+,n} + \mathbf{b}_{opt}, \tag{2}$$

- Linear transformation of output current
- Low computational complexity – multiplication and addition
- Parameters: statistical features, no online computation

Table of Contents

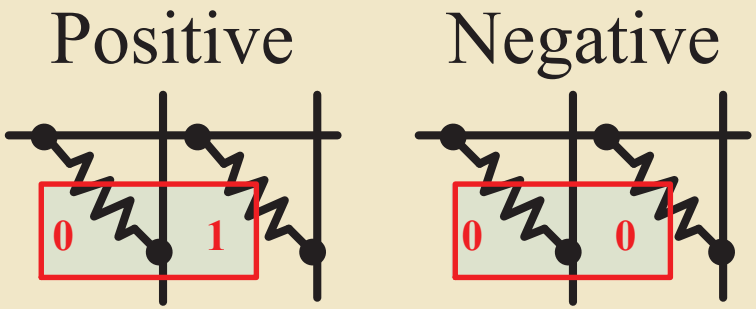
Proposed Reliable Crossbar Computing Schemes

2. Bit inversion (BI) mapping scheme

$e_1 \approx 5e_0$ in practice[5] $\Longrightarrow$ Intentionally increase percentage of 1s on the crossbar.

Example: a weight element is 1 and it's represented by two bits. Each cell is stuck at 0 and 1 with probability e_0 and e_1.

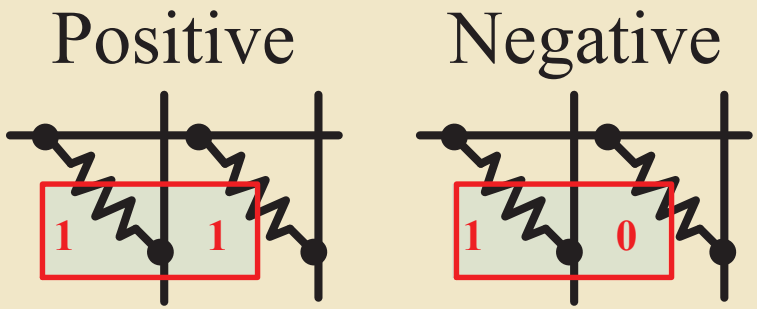
Traditional mapping scheme:



$1 - 0 = 1$

Error probability: $e_t \approx 3e_1 + e_0$

BI mapping scheme:



$3 - 2 = 1$

Error probability: $e_b \approx e_1 + 3e_0$

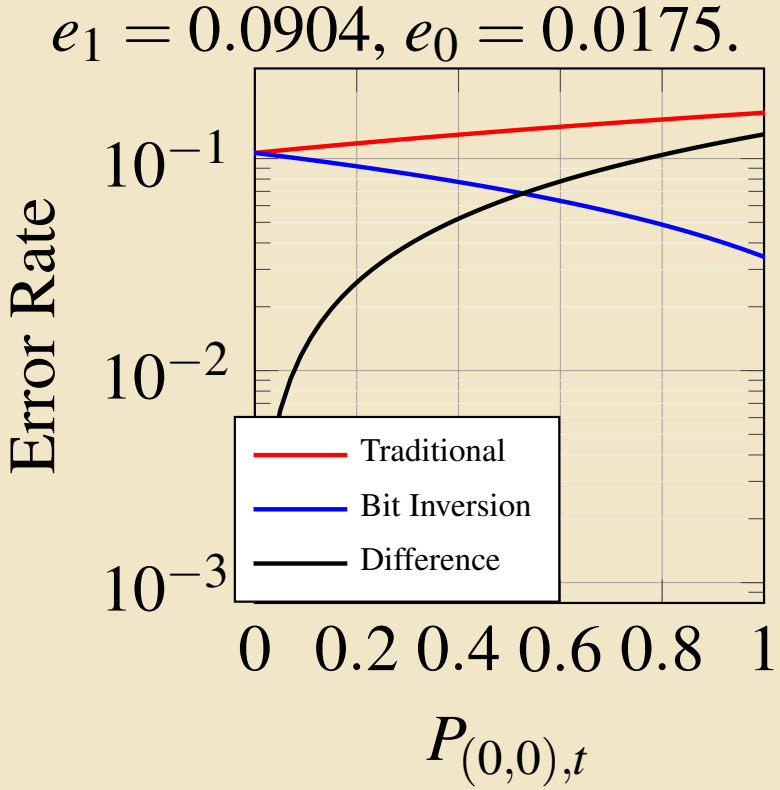
As long as $e_1 > e_0$, $e_t - e_b = 2e_1 - 2e_0 > 0$.

[5] Lerong Chen et al. "Accelerator-friendly neural-network training: Learning variations and defects in RRAM crossbar". In: *Design, Automation Test in Europe Conference Exhibition (DATE)*. 2017, pp. 19–24.

Proposed Reliable Crossbar Computing Schemes

2. Bit inversion (BI) mapping scheme – theoretical analysis

Error probability difference between two schemes.



$$
\begin{aligned}
& e_{diff} \\
& = e_t - e_b \\
& = 2P_{(0,0),t}(e_1 - e_0)(1 - e_1 - e_0).
\end{aligned}
\tag{3}
$$

- e_1: ratio of SA1.
- e_0: ratio of SA0.

If $e_1 > e_0$, $e_{diff} >= 0$.

Table of Contents

Proposed Reliable Crossbar Computing Schemes

3. Sparsity induction (SI) scheme

Sparsity induction (SI) scheme

- Use L1 regularization during training – reduce error rate caused by SAFs.
- Remove MVM results of all zero internal nodes and feature maps – eliminate error propagation.

Table of Contents

Evaluation

1. Experimental Settings

- Crossbar size: 128×128.
- Quantization precision: $N_w = 16$ bits and $N_i = 8$ bits.
- $e_1 = 0.0904$ and $e_0 = 0.0175$[5].
- Variations modeled as AWGN with variance σ_n^2.

Table: Experimental Settings

Network	Dataset	Accuracy	Configuration
MLP	MNIST	98.81%	$784 - 256 - 256 - 256 - 10$
Lenet 5	MNIST	99.19%	$28 \times 28 - 6c5 - 2s - 16c5 - 2s - 120 - 84 - 10$
Alexnet	CIFAR10	71.77%	$32 \times 32 - 64c11 - 2s - 192c5 - 2s - 384c3 - 256c3 - 256c3 - 2s - 10$

[5] Lerong Chen et al. "Accelerator-friendly neural-network training: Learning variations and defects in RRAM crossbar". In: *Design, Automation Test in Europe Conference Exhibition (DATE)*. 2017, pp. 19–24.

Table of Contents

Evaluation

2. Performance evaluation – accuracy vs noise level

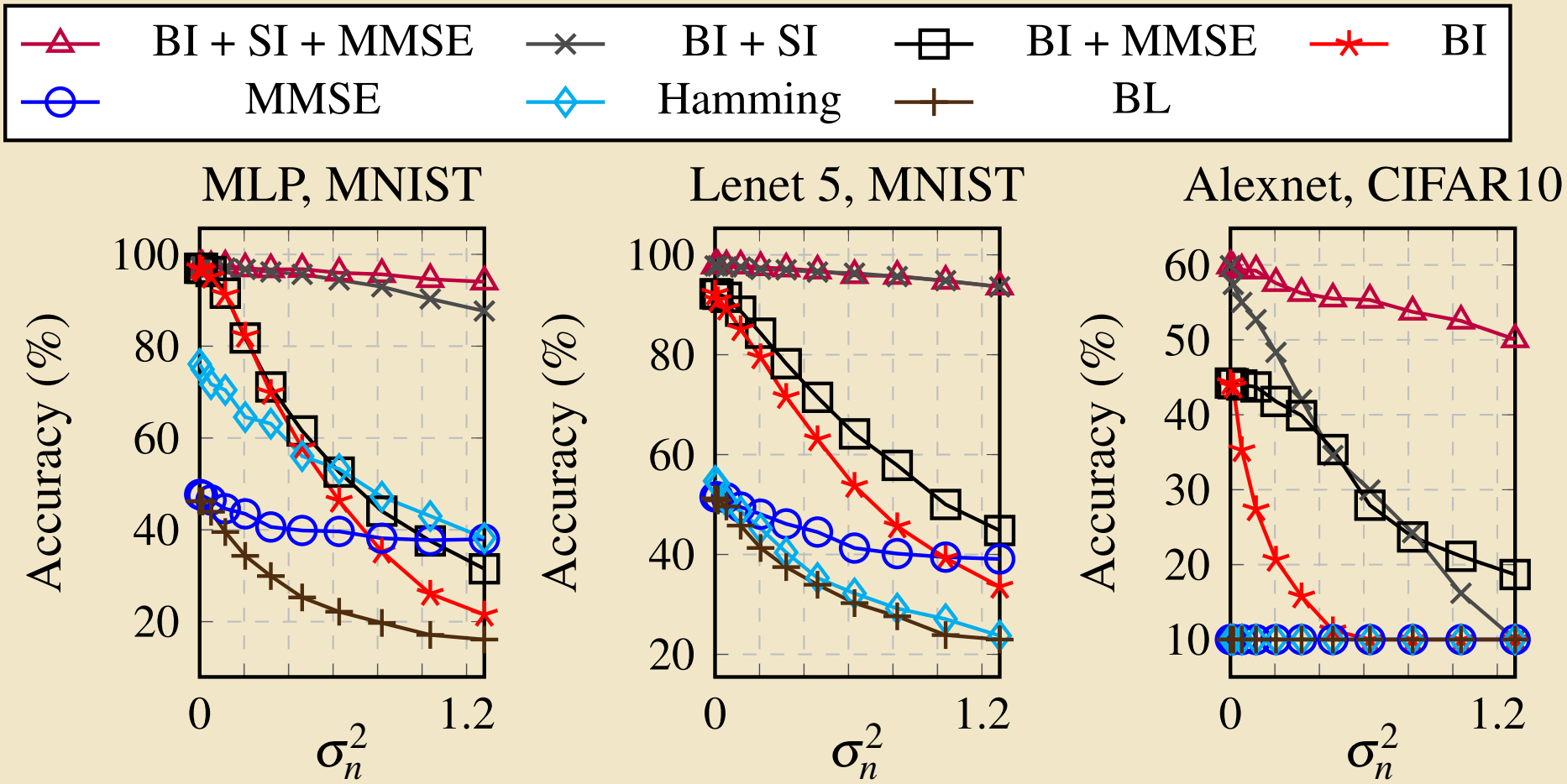
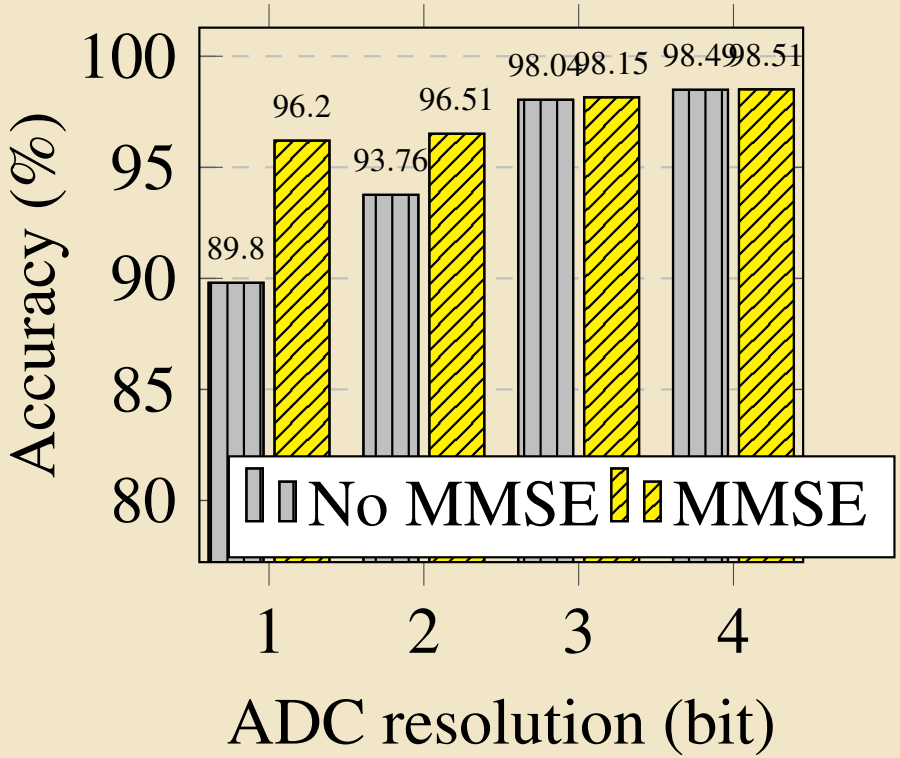


Figure: Accuracy comparison with SAFs and variations.

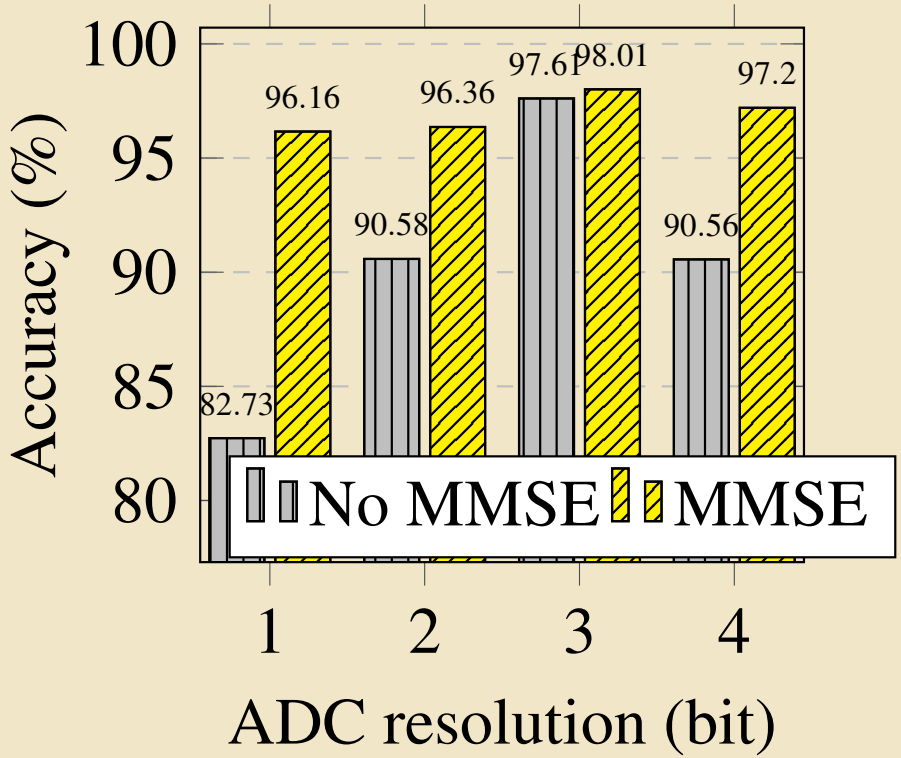
The proposed schemes improve the accuracy by $40\% - 78\%$**.**

Evaluation

2. Performance evaluation – accuracy vs ADC resolution



(a) MLP, MNIST, $\sigma_n^2 = 0$



(b) MLP, MNIST, $\sigma_n^2 = 0.4608$

Figure: Accuracy comparison with different ADC resolutions.

MMSE makes it possible to use fewer ADC bits → lower complexity.

Table of Contents

Evaluation

3. Overhead discussion

- BI: no additional hardware overhead
- SI: no additional hardware overhead
- MMSE: two extra multipliers and an adder

Potential ways to reduce overhead:

1. Applying MMSE to important layers only.
2. Using power 2 to approximate parameters.

Conclusion

- Proposed **MMSE based output denoising scheme** – first redundancy-free intermediate layer denoising.
- Proposed **BI mapping scheme**.
- Proposed **SI scheme**.