DependableHD: A Hyperdimensional Learning Framework for Edge-oriented Voltage-scaled Circuits

Dehua Liang
Osaka University

Hiromitsu Awano
Kyoto University

Noriyuki Miura
Osaka University

Jun Shiomi
Osaka University

Outline

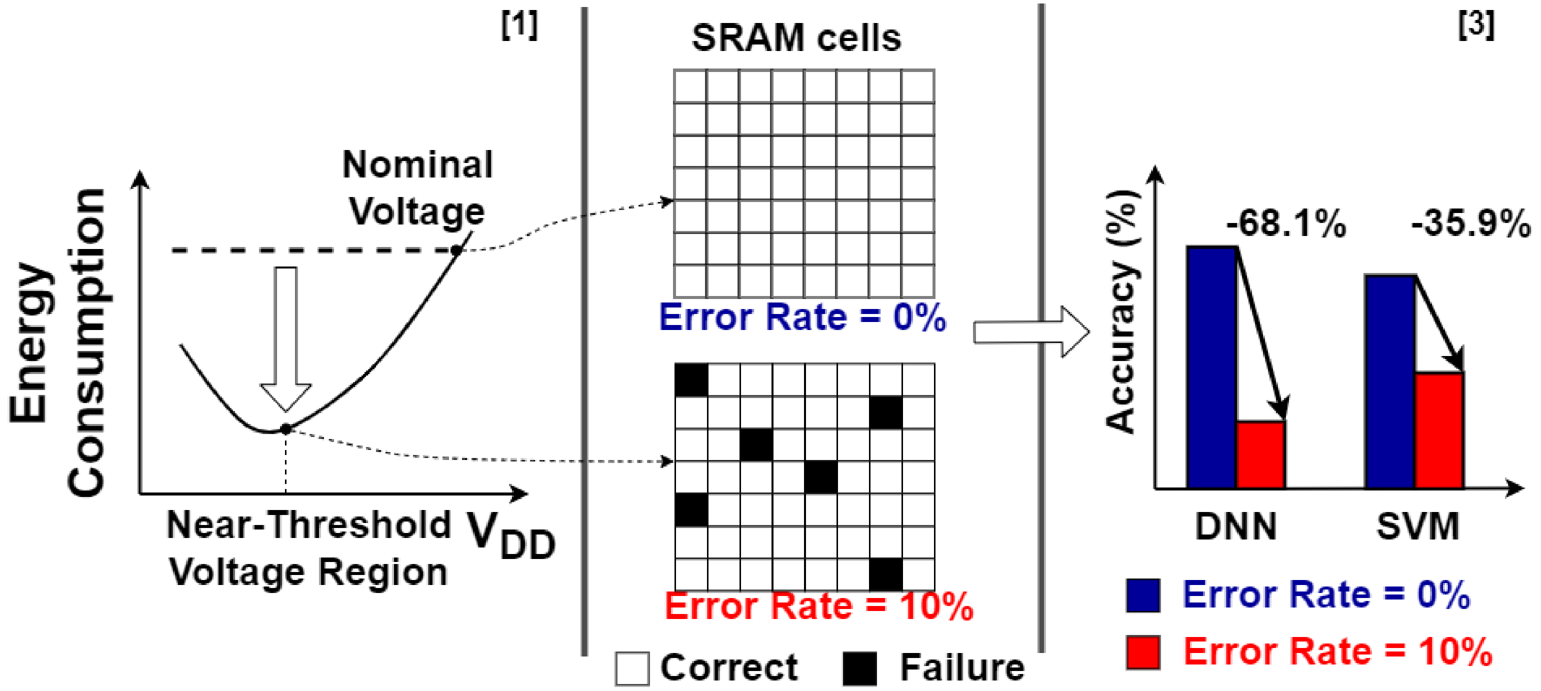
- Background (page 3-5)
  - Voltage Scaling
  - HyperDimensional Computing (HDC)
- Objective and Contributions (page 6)
- Preliminary (page 7)
- Proposed *DependableHD* Framework (page 8 - 11)
  - Margin Enhancement
  - Random Noise Injection
- Experimental Results (page 12 - 14)
- Conclusions (page 15)

Background — Energy efficient and robust computing paradigm is demanded

- Applications raise the demand for a more energy efficient and robust model.
  - Traditional DNNs brings high energy consumption and sensitivity.
  - Voltage Scaling is a promising technique to minimize energy consumption.
  - Brain-inspired HyperDimensional Computing (HDC) shows potential to provide high robustness.

Background — Voltage Scaling

- Voltage Scaling is a classic technique to minimize energy consumption.



A) 4.7× **energy consumption reduction** by voltage scaling[1]

B) 65nm SRAM cells error rate increases from 10^{-7} to 4% when supply voltage scaled from nominal value to 500mV[2]

C) The **memory failure** result in serious and unacceptable performance degradation[3]

[1] Jain, Shailendra, et al., "A 280mV-to-1.2V wide-operating-range IA-32 processor in 32nm CMOS," in ISSCC, 2012.
[2] Dreslinski, R. G. et al., "Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits," in Proc. IEEE, 2010.
[3] P. Poduval et al., "Hyperdimensional Self-Learning Systems Robust to Technology Noise and Bit-Flip Attacks," in ICCAD, 2021.

Background — Hyperdimensional Computing

- Brain-inspired Hyperdimensional Computing (HDC) is a promising alternative computing paradigm in a light-weight and robust approach.

- Advantage:

✓ Fast and efficient learning process.

✓ Hardware friendly and light-weight.

✓ **High robustness**.

- Related Works & Challenges:

☐ Ref [4] focuses on the robustness during extra feature extraction phase.

☐ High precision elements lead to higher accuracy but also weaker robustness[5].

☐ The robustness in Ref [5, 6] is superior than DNNs, but **NOT ENOUGH** for aggressive voltage scaling strategy.

[4] P. Poduval et al., "StocHD: Stochastic Hyperdimensional System for Efficient and Robust Learning from Raw Data," in DAC, 2021.
[5] A. Rahimi et al., "A Robust and Energy-Efficient Classifier Using Brain-Inspired Hyperdimensional Computing," in ISLPED, 2016.
[6] A. Hernandez-Cane et al., "OnlineHD: Robust, Efficient, and Single-Pass Online Learning Using Hyperdimensional System," in DATE, 2021.

Objective & Contributions

- **Objective:** Use robust HDC to handle voltage-scaling induced memory failure.

- **Main Contributions in *DependableHD*:**
  - **New concept for HDC learning to improve robustness**

    Proposed margin enhancement and random noise injection techniques.
  - **High robustness**

    1.22% accuracy loss under 10% error rate, **11.2x improvement**.
  - **Energy Reduction**

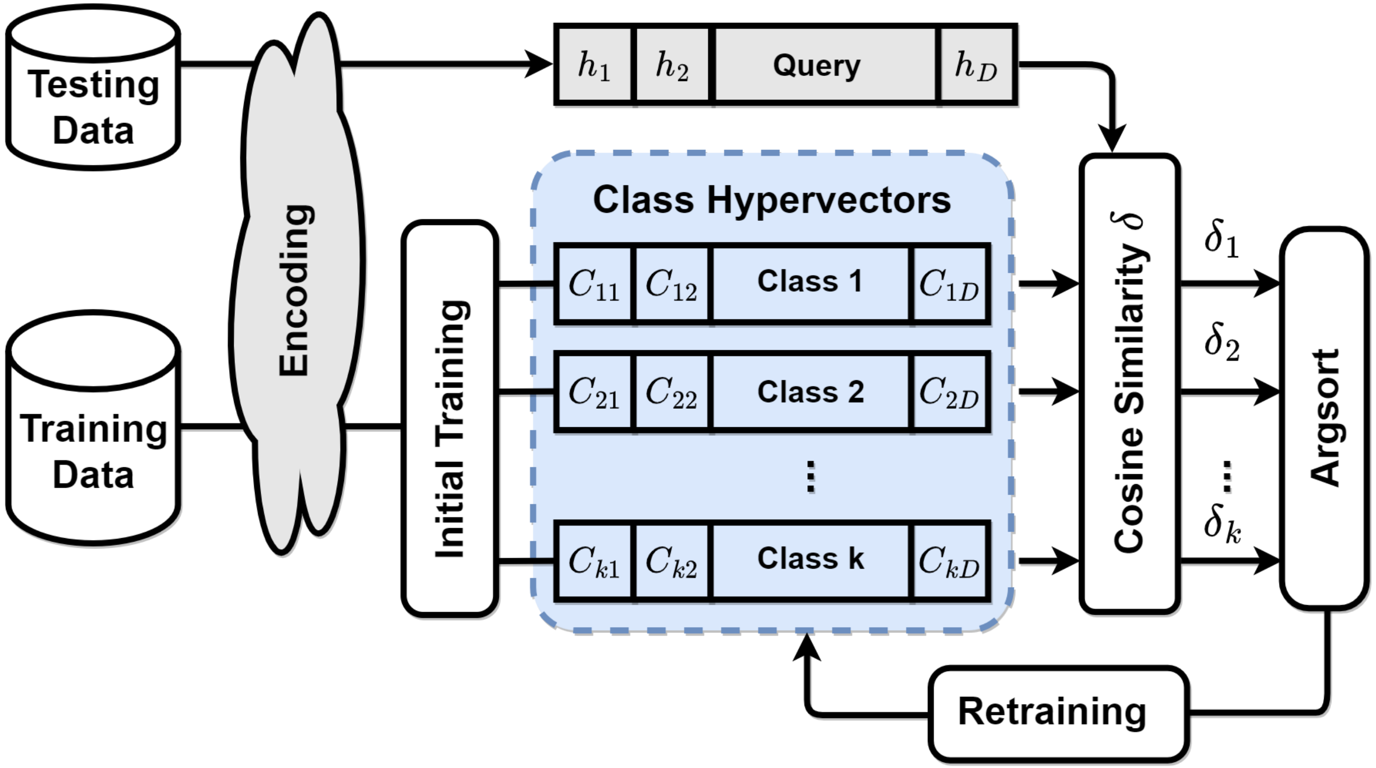
    Support the systems to scale down V_{DD} from 400mV to 300mV.

    50.4% energy consumption reduction.

Preliminary of HDC

Encoding:

- Encode the training/testing objects to long size vectors (D=10k), called *hypervectors* $\vec{H}$



Initial Training:

- Adds all training hypervectors from the same class i to generate class hypervector $\vec{C}_i$.

Similarity Measurement & Inference:

- For hypervectors with non-binarized elements, check the Cosine similarity $\delta_i = \delta(\vec{H}, \vec{C}_i)$ between testing hypervector $\vec{H}$ and class hypervectors $\vec{C}$.

Preliminary of HDC

Conventional Retraining:

Step 1 - For the target label l, check the similarity δ_l

Step 2 - Find the highest $\delta_r = Max(\delta_i)$ where $i \neq l$

Step 3 - When $\delta_l - \delta_r > 0$, the prediction is right. No update for the model.

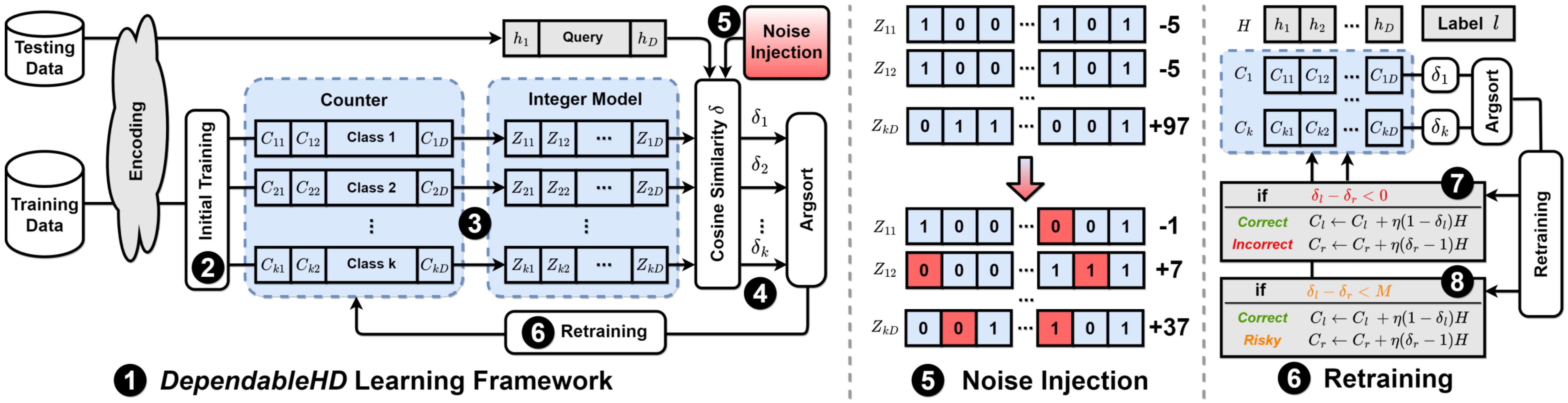
Step 4 - When $\delta_l - \delta_r < 0$, the prediction is wrong. Update the model as:

if	$\delta_l - \delta_r < 0$
Correct	$C_l \leftarrow C_l + \eta(1 - \delta_l)H$
Incorrect	$C_r \leftarrow C_r + \eta(\delta_r - 1)H$

- For the **correct** class l, add H to the class hypervector C_l
  - $\delta_l \uparrow$
- For the **incorrect** class r, subtract H from the class hypervector C_r
  - $\delta_r \downarrow$

Proposed — Overview of *DependableHD*

Compared to the traditional HDC, our *DependableHD* is a combination of (❽) Margin enhancement and (❺) Random noise injection during the retraining phase.



Such strategy is capable for most popular HDC algorithms, and no extra hardware cost during the inference is required.

Proposed — Margin Enhancement

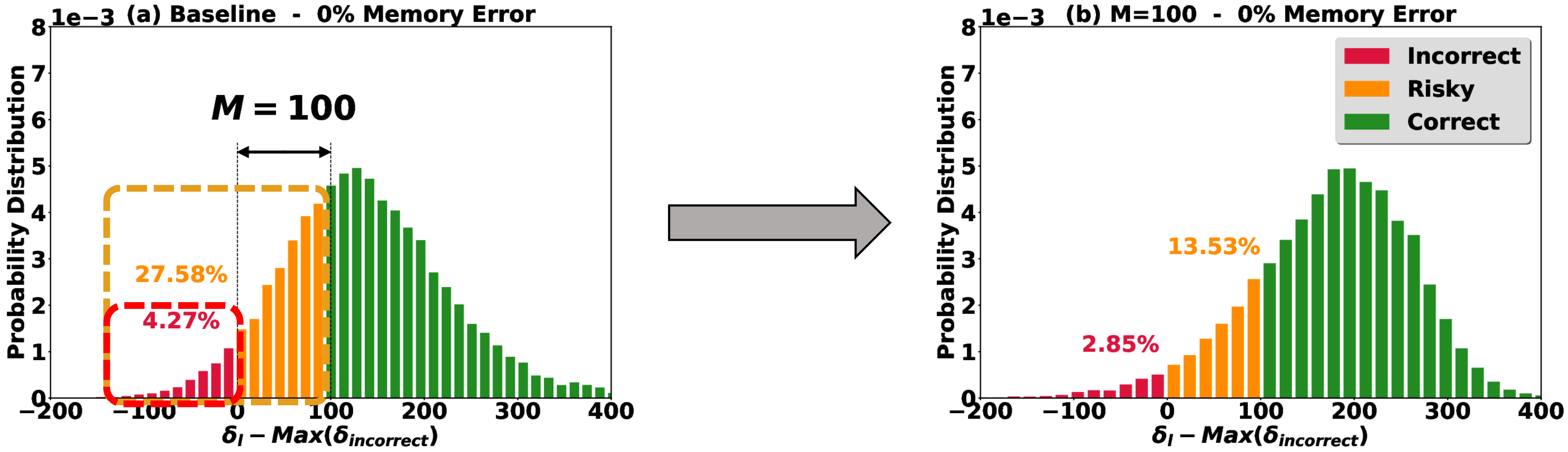
Baseline: Utilize the training samples with incorrect prediction to retrain the model.

if	$\delta_l - \delta_r < 0$
Correct	$C_l \leftarrow C_l + \eta(1-\delta_l)H$
Incorrect	$C_r \leftarrow C_r + \eta(\delta_r - 1)H$

Margin Enhancement: Utilize the training samples with incorrect and risky prediction to retrain the model.

if	$\delta_l - \delta_r < M$
Correct	$C_l \leftarrow C_l + \eta(1-\delta_l)H$
Risky	$C_r \leftarrow C_r + \eta(\delta_r - 1)H$

➢The percentage of risky prediction is reduced from 27.58% to 13.53%



Proposed — Margin Enhancement

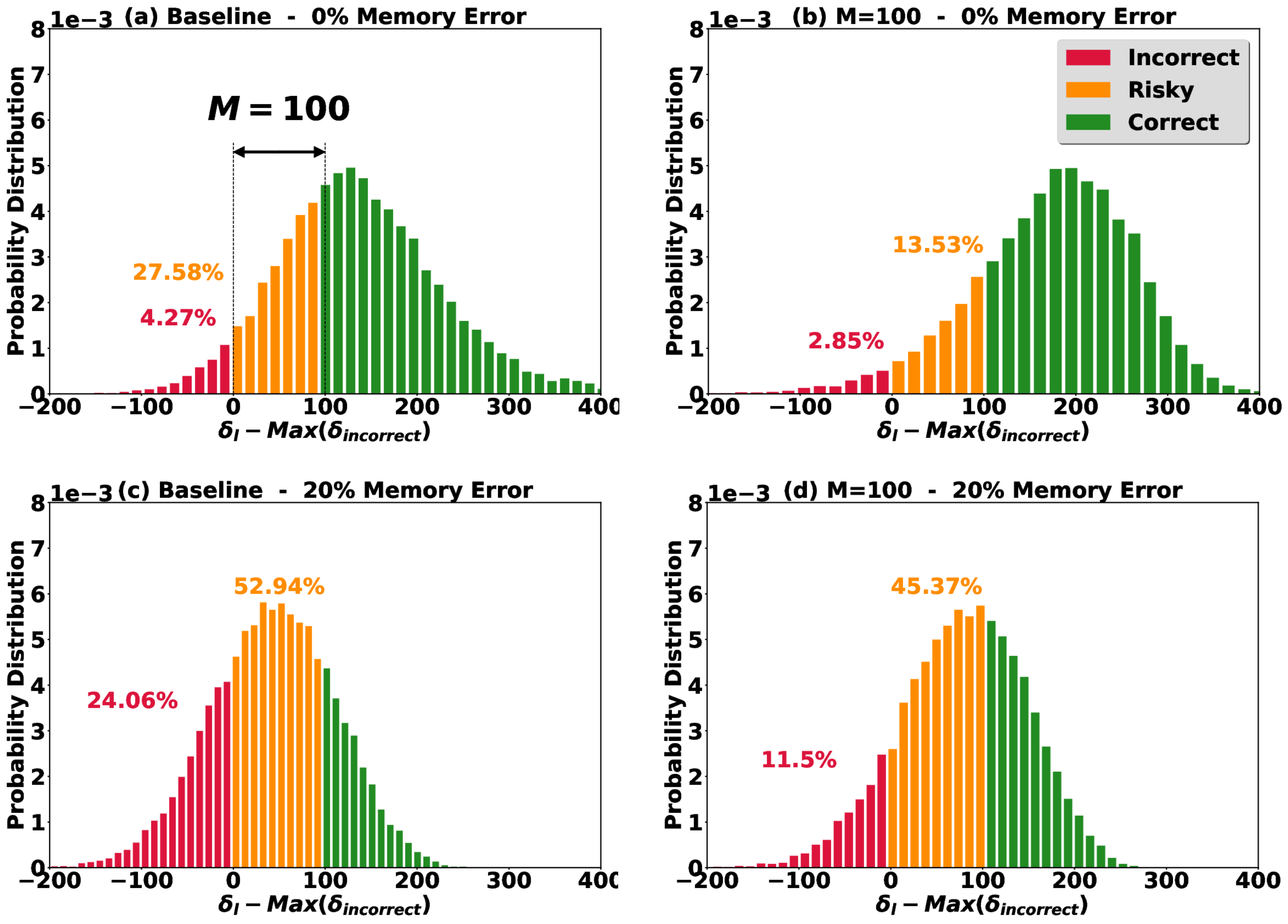


Fig.(a) → Fig.(c):

- The risky predictions tend to becomes incorrect prediction

Fig.(a) → Fig.(b):

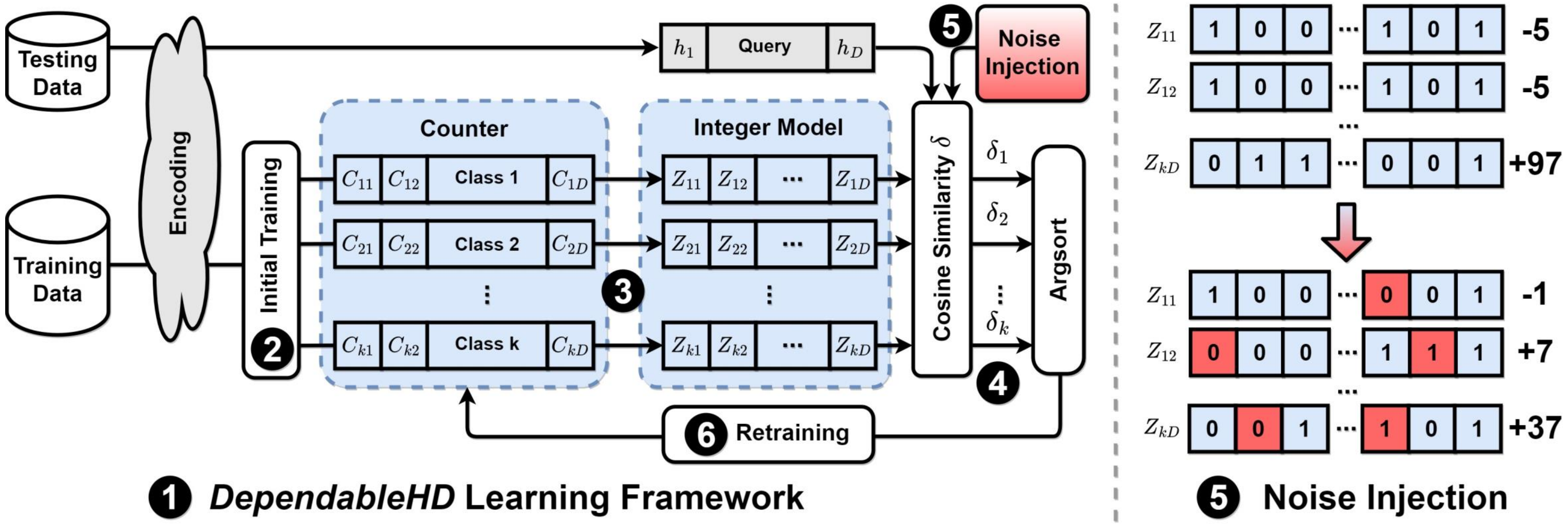
- The percentage of risky prediction is reduced

Fig.(c) and (d)

- Margin enhancement reduces the incorrect prediction

➢ Robustness improved

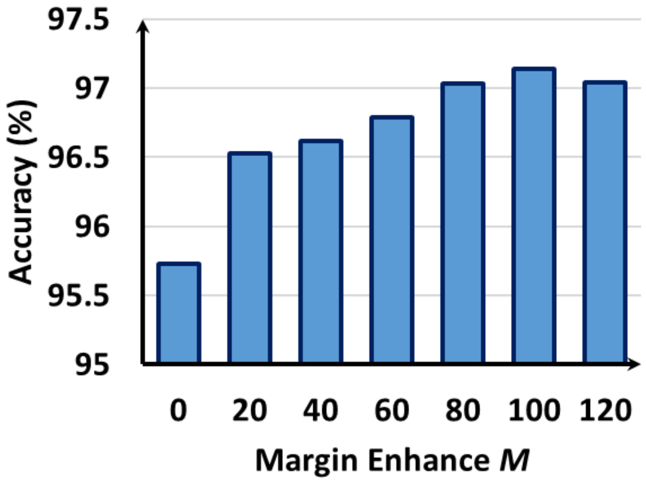
Proposed — Random Noise Injection

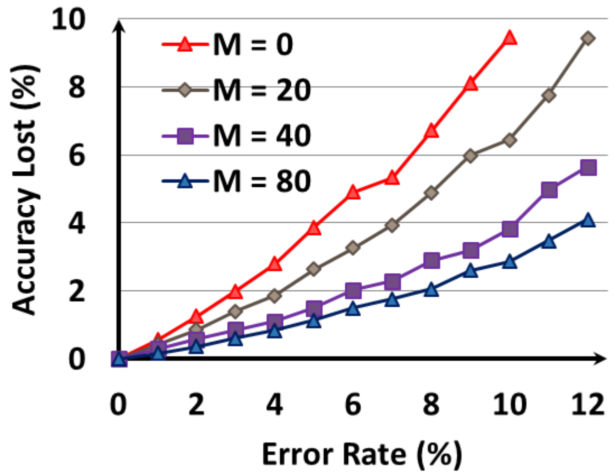


❺ **Random Noise Injection**: Inject bit-flipping error (error rate = ***R***) during the retraining

Experiment — Impact of parameters

Margin Enhancement Level *M*





With the increase of ***M*** in the range of 0~100

- Accuracy is slightly improved.
- The strengthen of prediction in risky samples benefits the model.

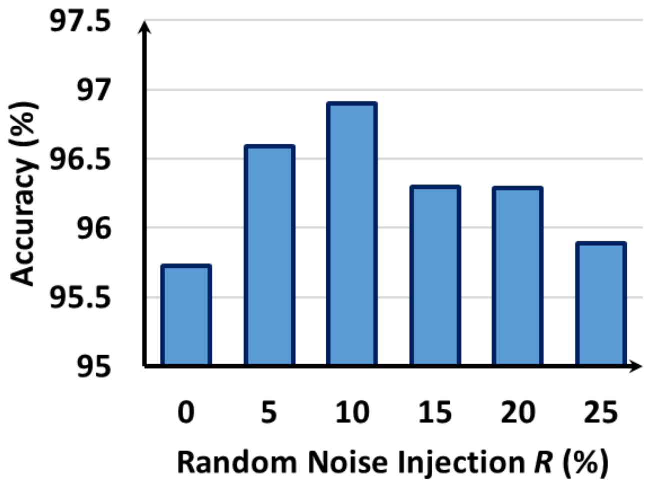
When ***M*** continues to get large

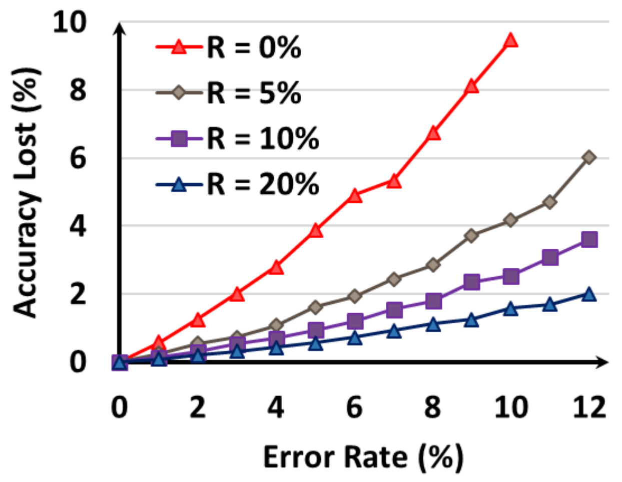
- Too much risk samples are utilized to retrain the model, results in slight accuracy draw back.
- Robustness is further improved.

➢ **There is a trade-off in tuning the margin enhancement level *M***

Experiment — Impact of parameters

Random Noise Injection Level *R*





With the increase of ***R*** in the range of 0~10%

- Accuracy is slightly improved.
- The random noise prevent the model from overfitting issues.

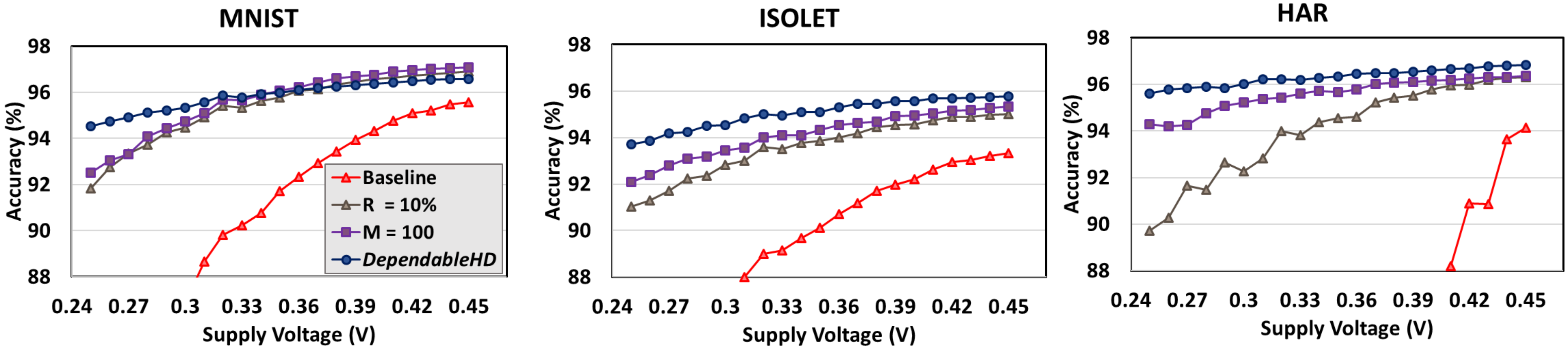
When ***R*** continues to get large

- Too much noise are injected during retraining, results in slight accuracy draw back.
- Robustness is further improved.

➢ **There is a trade-off in tuning the random noise injection level *R***

Experimental Results —Performance Comparison

Accuracy comparison under different supply voltage in 65nm SRAM.



Baseline	Reference [3]
Noise Injection	*R*=10%
Margin Enhance	*M*=100
DependableHD	*R*=10%, *M*=100

	$@V_{dd}$	Accuracy	Energy Consumption	Energy Reduction
Baseline HD [6]	400mV	91.09%	0.75pJ	–
DependableHD	300mV	95.29%	0.37pJ	50.4%

[3] P. Poduval et al., "Hyperdimensional Self-Learning Systems Robust to Technology Noise and Bit-Flip Attacks," in ICCAD, 2021.

Conclusions

- Proposed **margin enhancement** and **random noise injection** to improve the robustness of HDC model.
- 1.22% accuracy loss under 10% error rate, achieved **11.2x robustness improvement** without any extra hardware cost.
- The sufficient robustness guarantees the application of aggressive voltage scaling strategy from 400mV to **300mV**, which provides **50.4% energy reduction**.