DE-C3: Dynamic Energy-Aware Compression for Computing-In-Memory-Based Convolutional Neural Network Acceleration

Speaker: Guan-Wei Wu

Advisor: Prof. An-Yeu (Andy) Wu

Department of Electrical Engineering,

National Taiwan University, Taipei, Taiwan

Date: 2023/09/07

ACCESS IC LAB

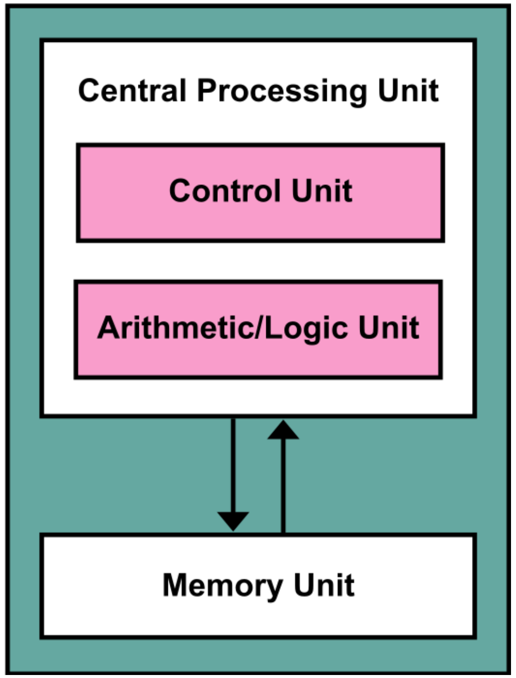
Outline

- Background
- Introduction of Computing-In-Memory (CIM)
- Challenges of Recent CIM-Based Compression
- Proposed DE-C3 Framework
- Experimental Results
- Conclusions

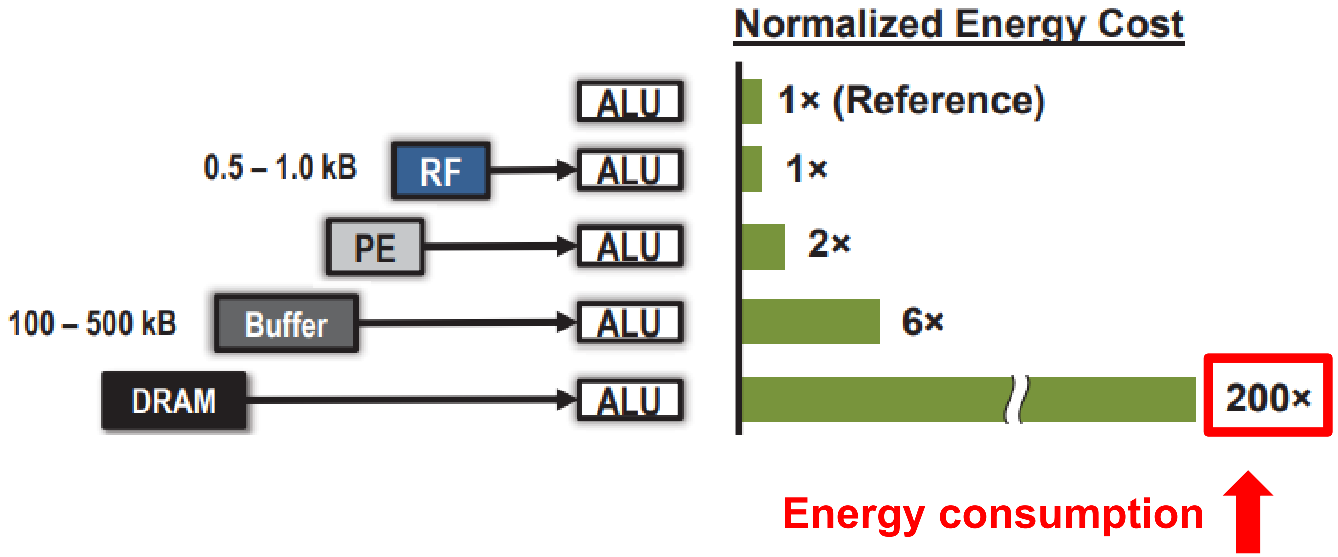
Background

Background

- Von Neumann Architecture
  - Processing unit and memory unit are separated
- CNNs require frequent data transfers between processing unit and memory unit



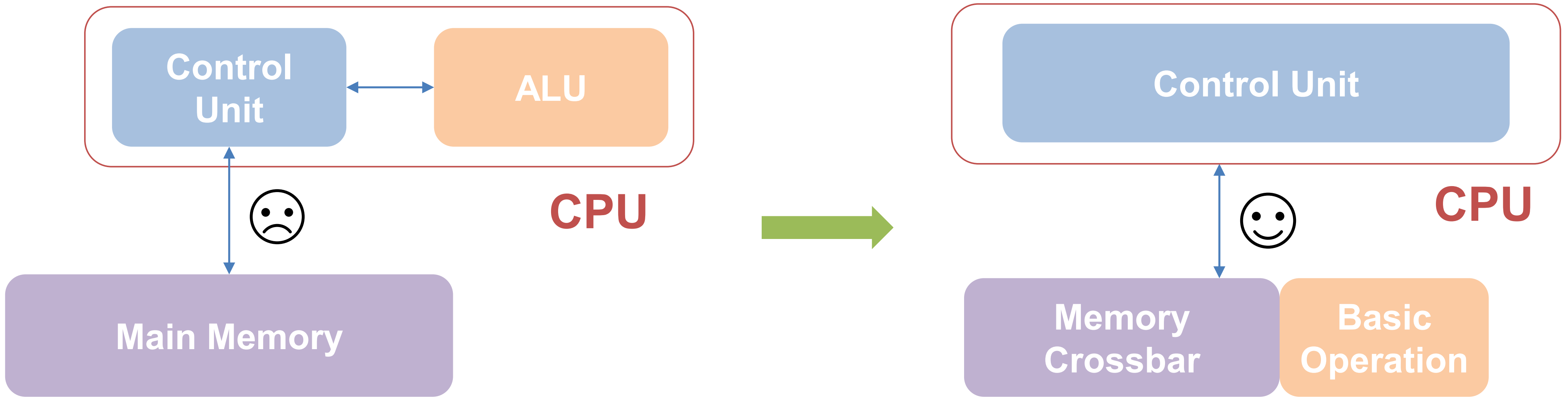
[1]



[2]

Background

- Computing-In-Memory (CIM)
  - Weights of CNNs are stored in memory crossbar
  - Memory crossbar provides basic operations



Introduction of Computing-In-Memory (CIM)

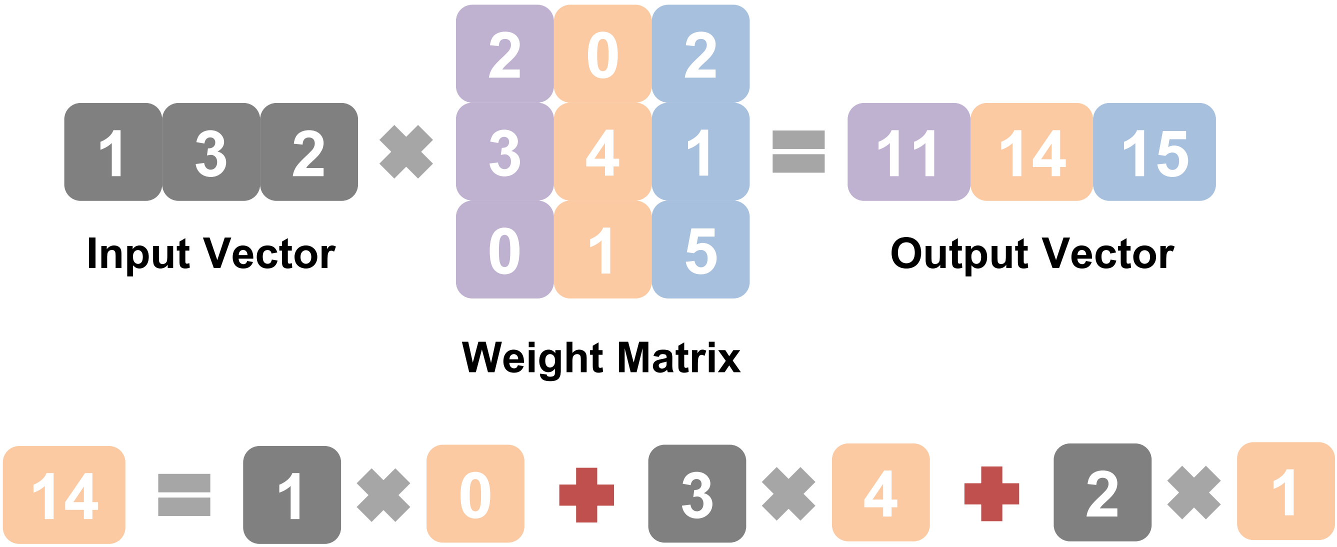
Introduction of Computing-In-Memory (CIM)

- Most operations in CNNs are vector-matrix multiplications (VMMs)

$$\underbrace{\begin{bmatrix}1 & 3 & 2\end{bmatrix}}_{\text{Input Vector}} \times \underbrace{\begin{bmatrix}2 & 0 & 2\\ 3 & 4 & 1\\ 0 & 1 & 5\end{bmatrix}}_{\text{Weight Matrix}} = \underbrace{\begin{bmatrix}11 & 14 & 15\end{bmatrix}}_{\text{Output Vector}}$$

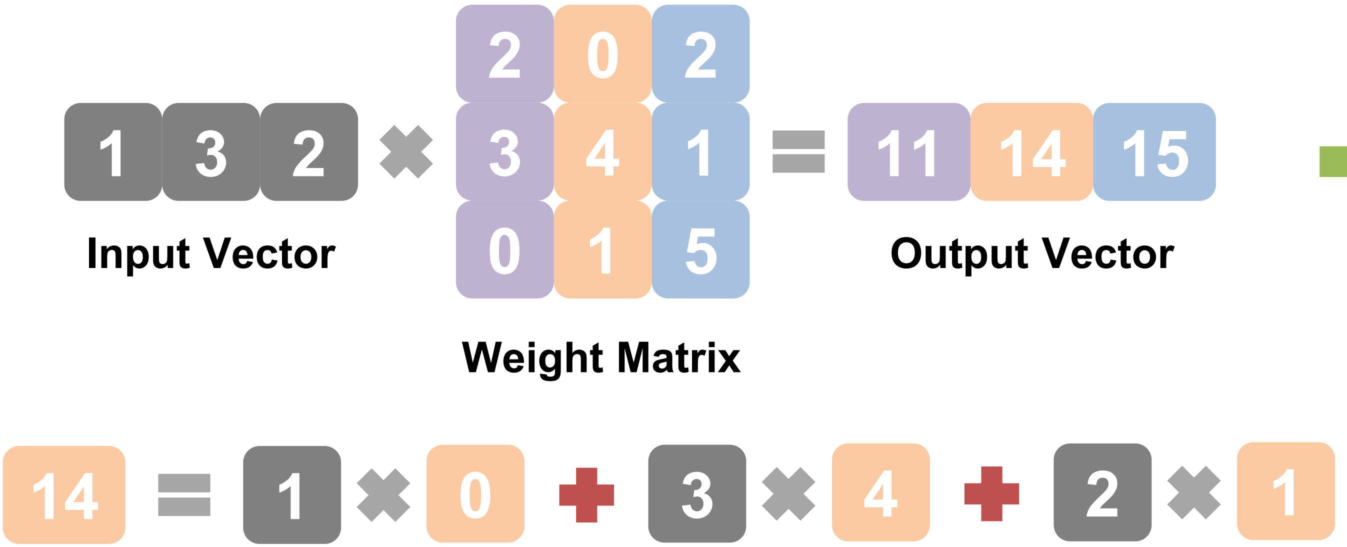
Introduction of Computing-In-Memory (CIM)

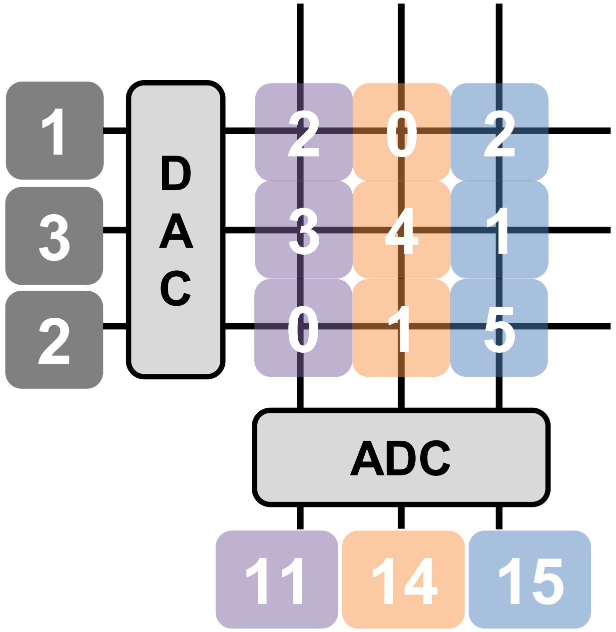
- Most operations in CNNs are vector-matrix multiplications (VMMs)
  - Essentially perform multiply-and-accumulate (MAC) operations



Introduction of Computing-In-Memory (CIM)

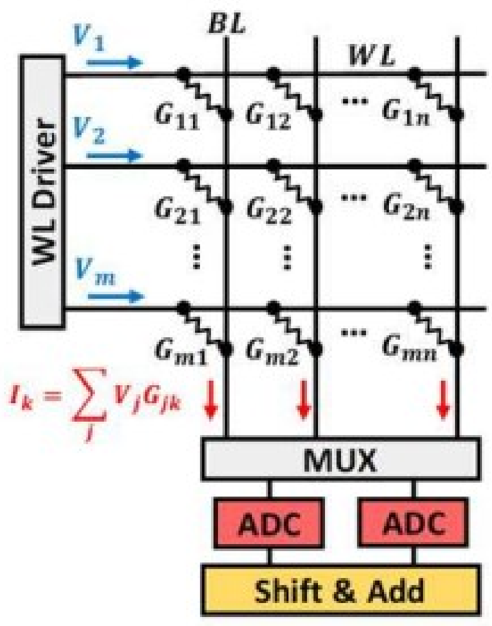
- Most operations in CNNs are vector-matrix multiplications (VMMs)
  - Essentially perform multiply-and-accumulate (MAC) operations
- CIM parallelly computes VMMs by performing MAC in crossbar
  - Reduce energy consumption





Introduction of Computing-In-Memory (CIM)

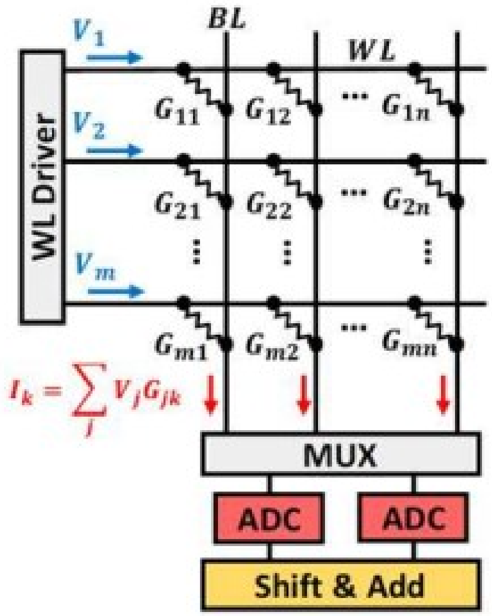
- Mapping from CNNs to CIM architecture
  - Input voltage, weight conductance, output current

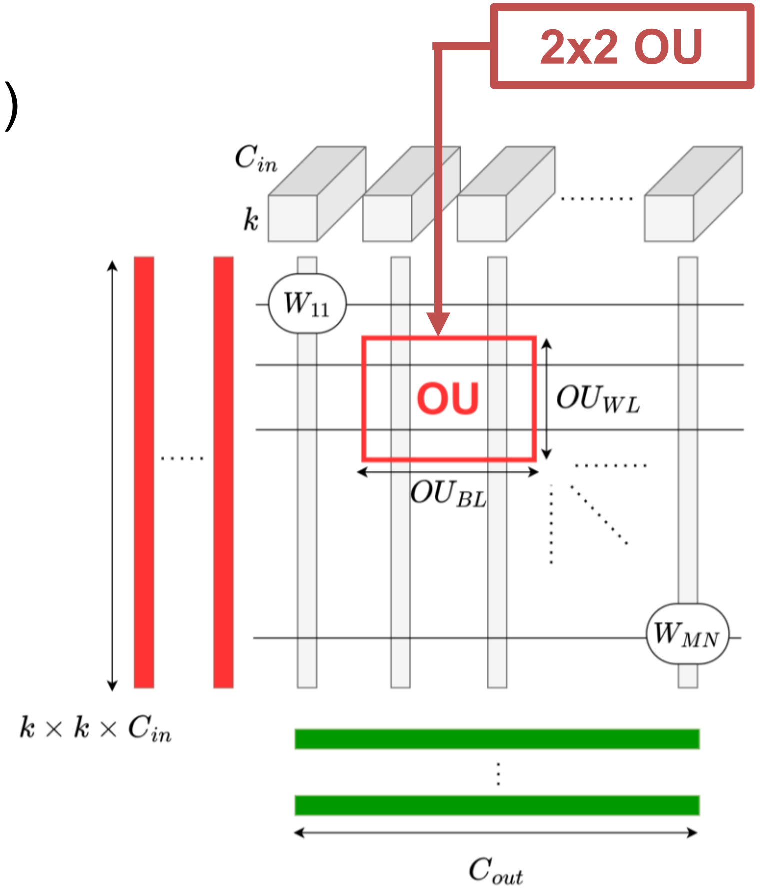


[3]

Introduction of Computing-In-Memory (CIM)

- Mapping from CNNs to CIM architecture
  - Input voltage, weight conductance, output current
- Typical CIM architecture and operation units (OUs)



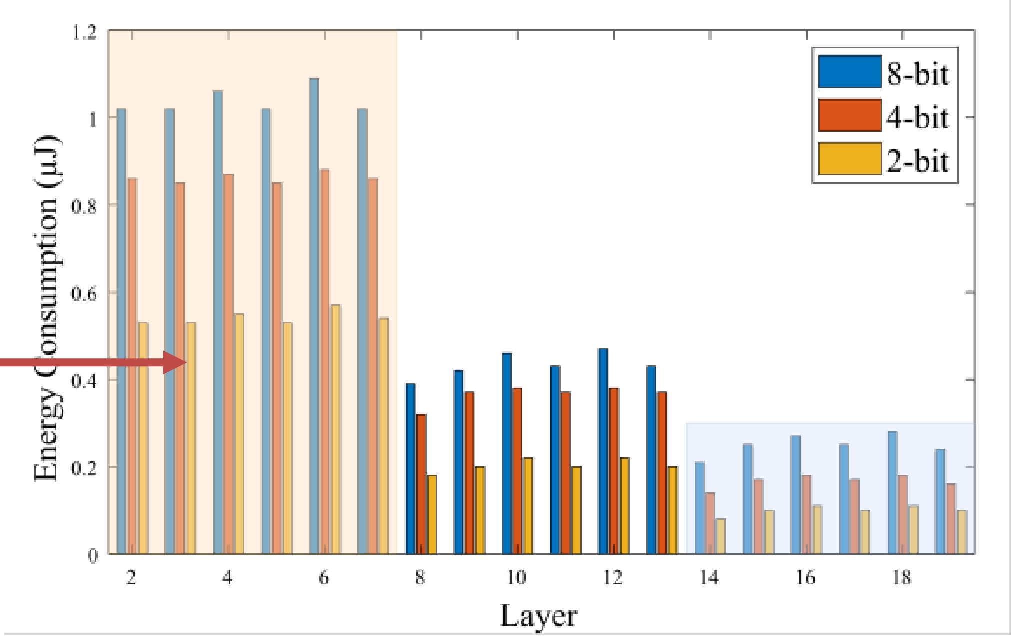


Challenges of Recent CIM-Based Compression

Challenges of Recent CIM-Based Compression

- Energy consumption per each layer can be very different
  - Larger size of feature maps in shallower layers
  - Smaller size of weight matrices in shallower layers

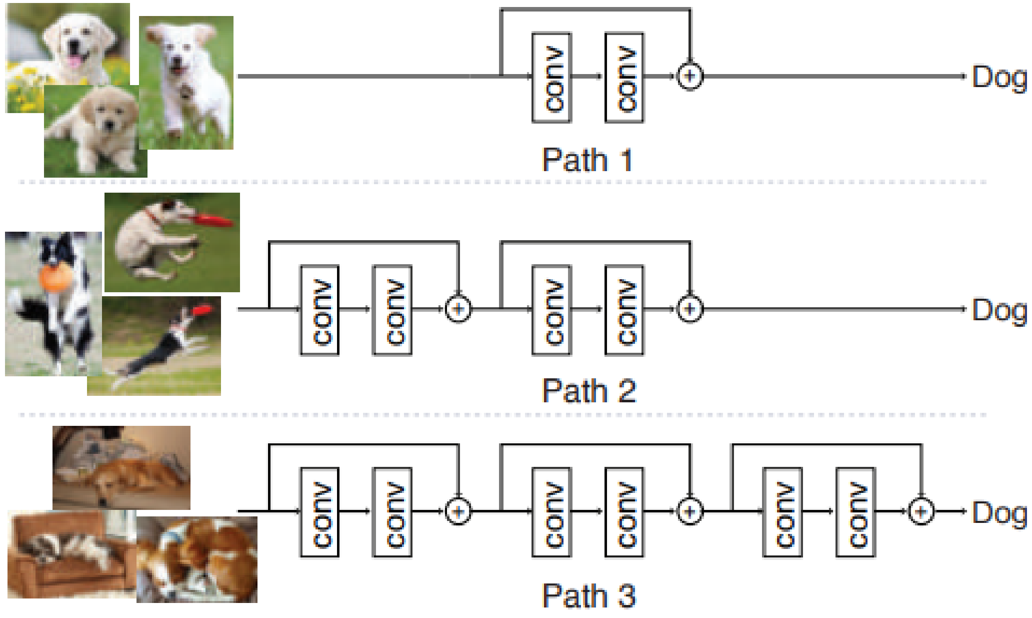
Larger Energy Cost
Smaller # of OUs



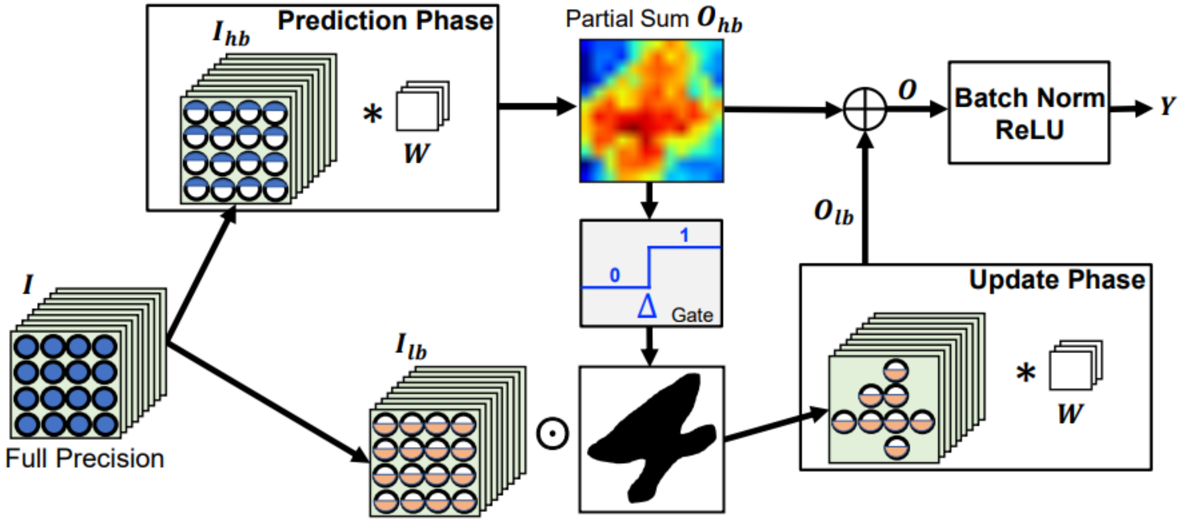
An OU in ℓ^{th} layer consumes:
$$\frac{E^{\ell}}{\#\ of\ OUs}\ (\mu J)$$

Challenges of Recent CIM-Based Compression

- Identify the input-dependent redundancy and detect critical input component
- Apply dual-path processing for dynamic inference
  - DE-C3 provides dual-precision quantization and pruning compression modes



[4]

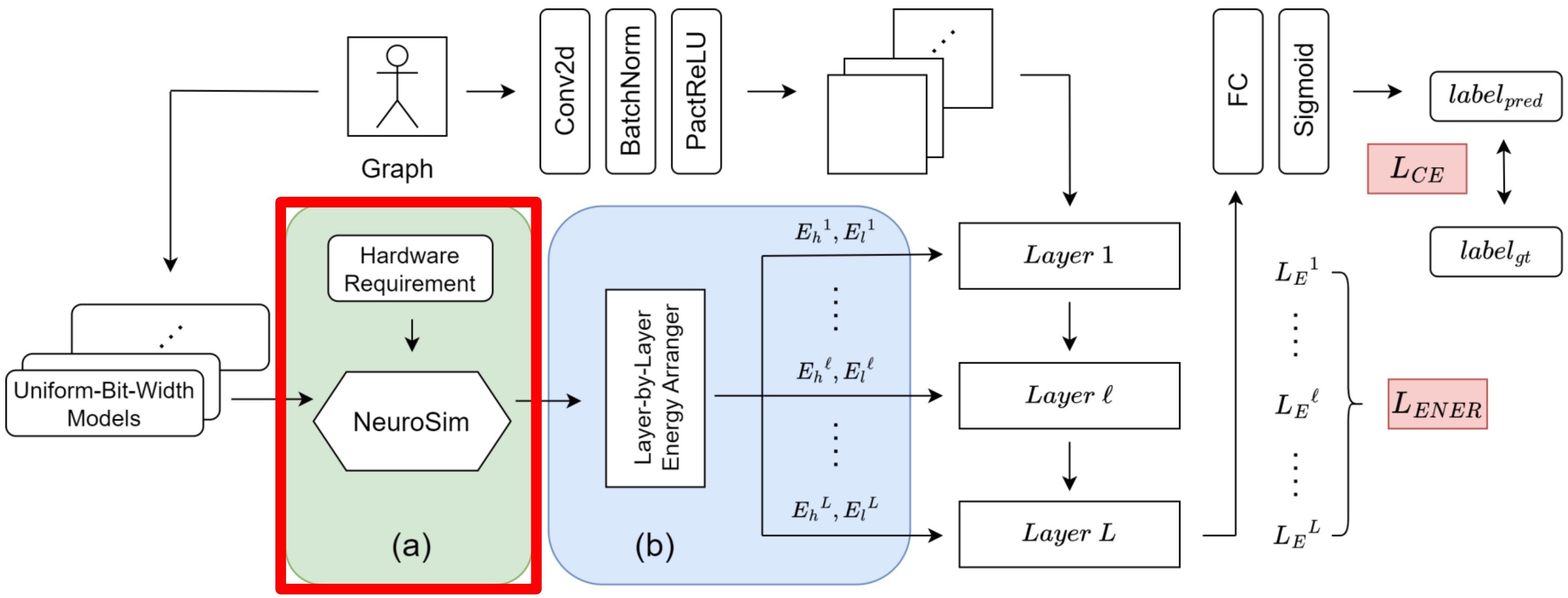


[5]

Proposed DE-C3 Framework

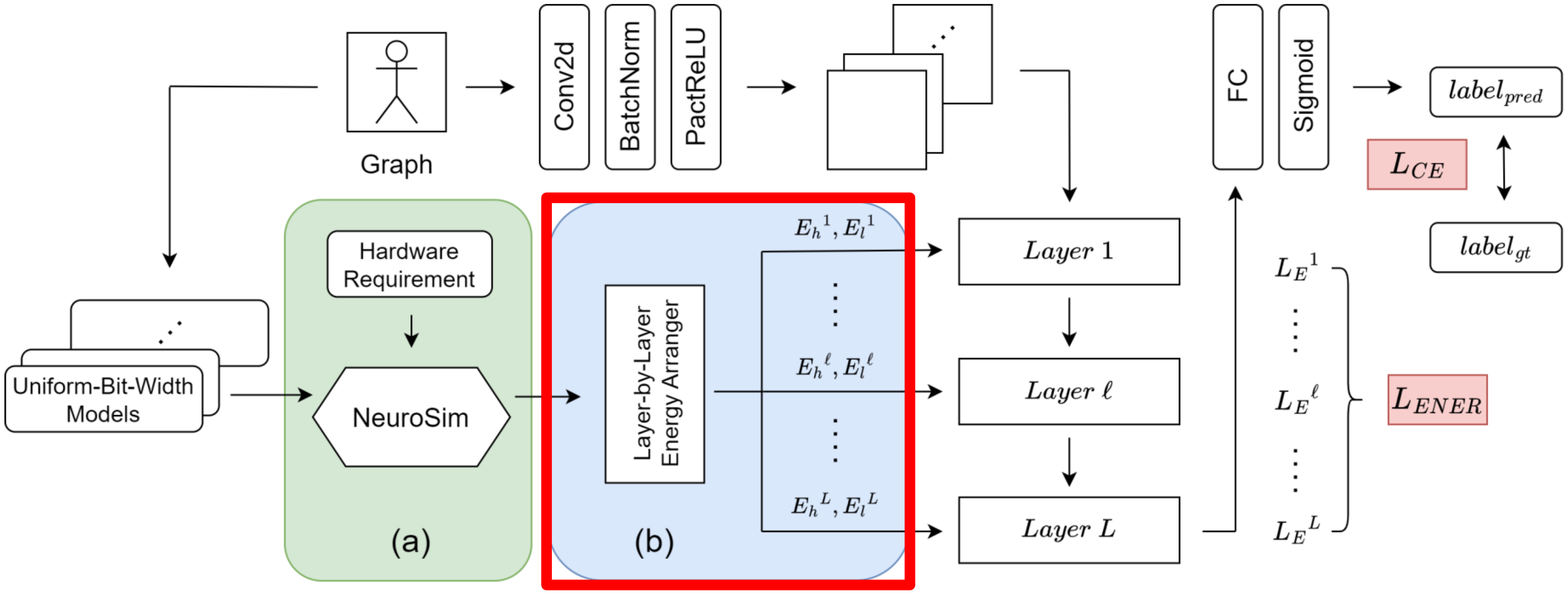
Proposed DE-C3 Framework

- DNN+NeuroSim estimates layer-wise energy consumption for models trained on different bit width under CIM architecture



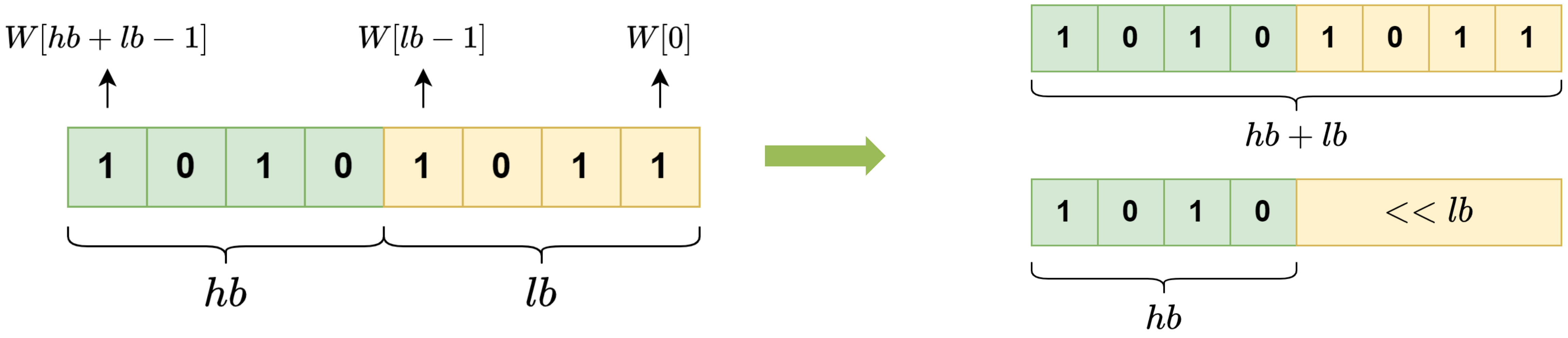
Proposed DE-C3 Framework

❖ A pair of bit width pair is given to each layer manually by the arranger



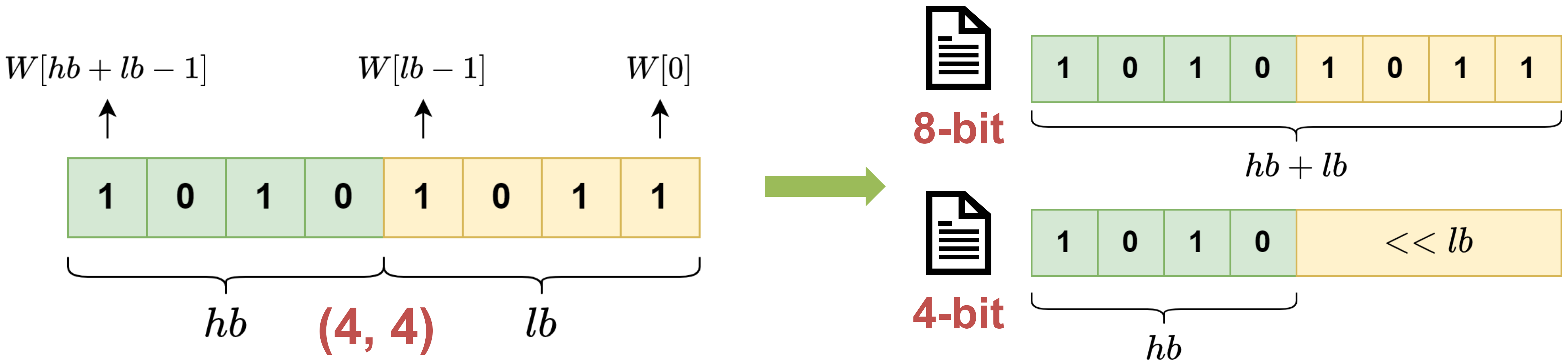
Proposed DE-C3 Framework

- Each bit width pair corresponds to two energy values of different precisions from DNN+NeuroSim



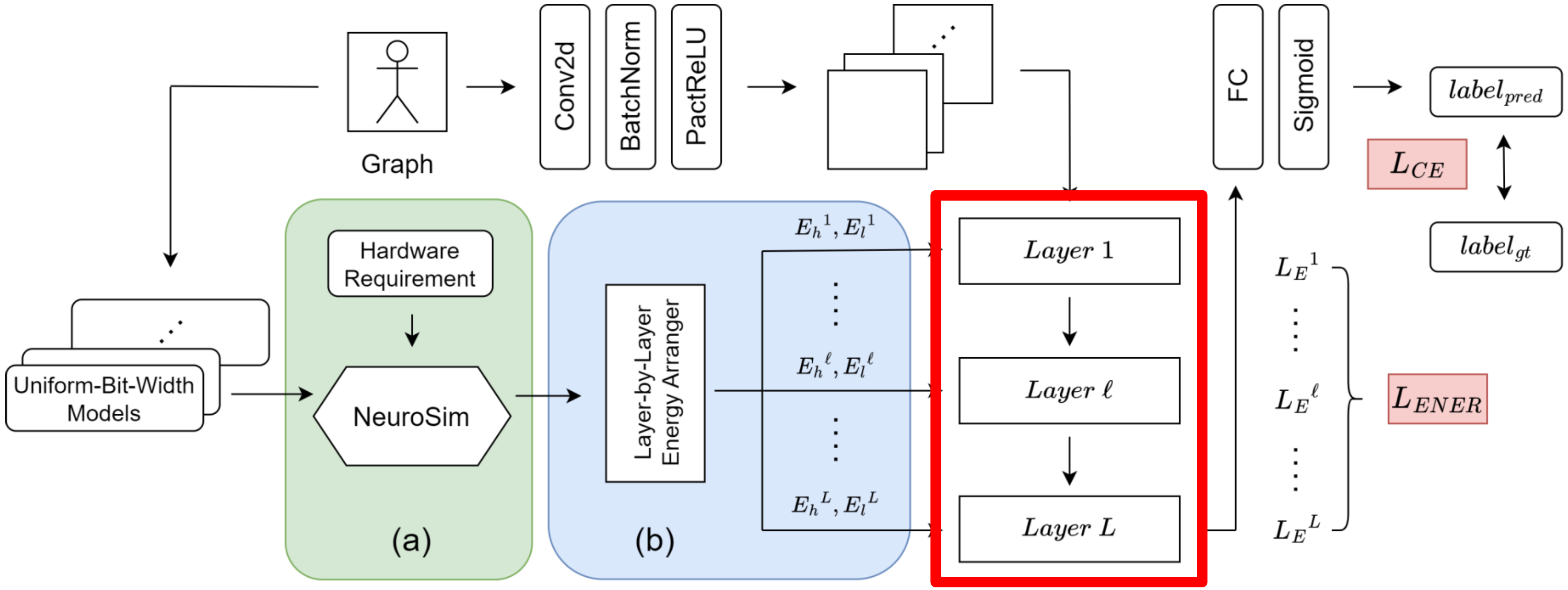
Proposed DE-C3 Framework

- Each bit width pair corresponds to two energy values of different precisions from DNN+NeuroSim



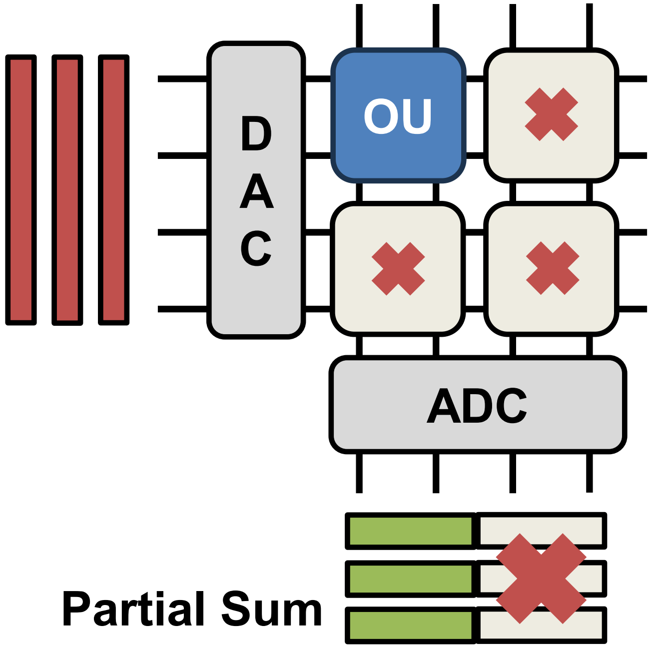
Proposed DE-C3 Framework

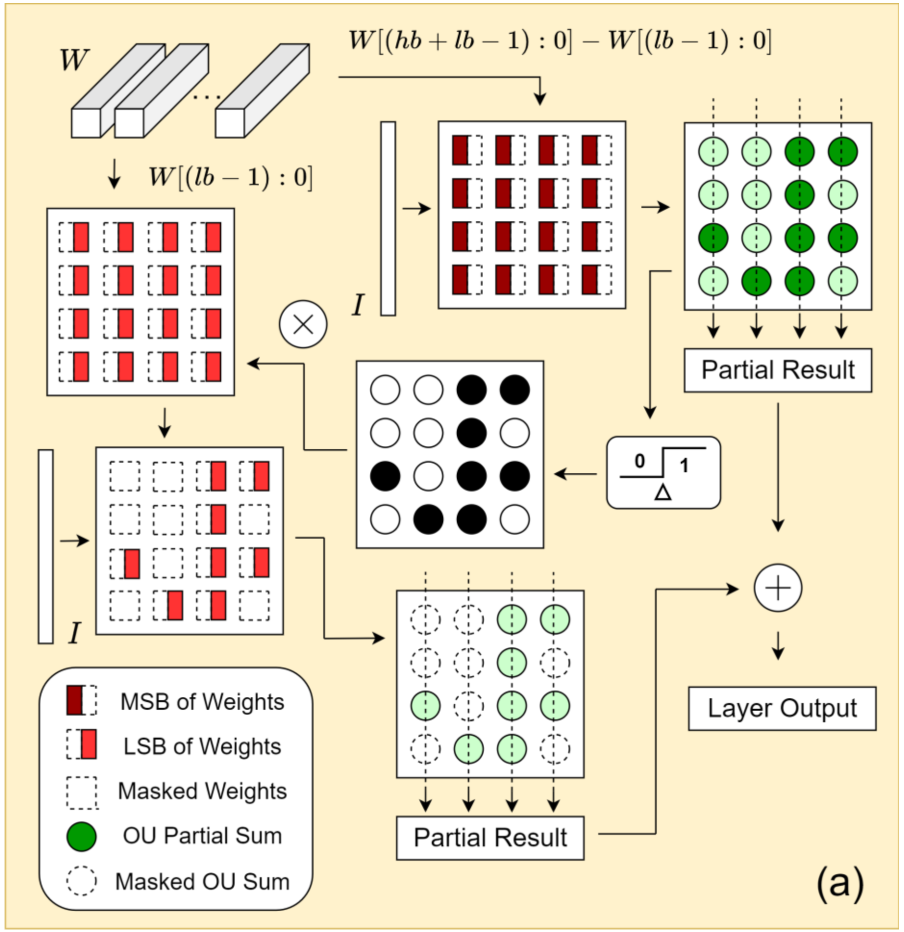
- Each DE-C3 layer takes the input feature map and the energy information into account



Proposed DE-C3 Framework

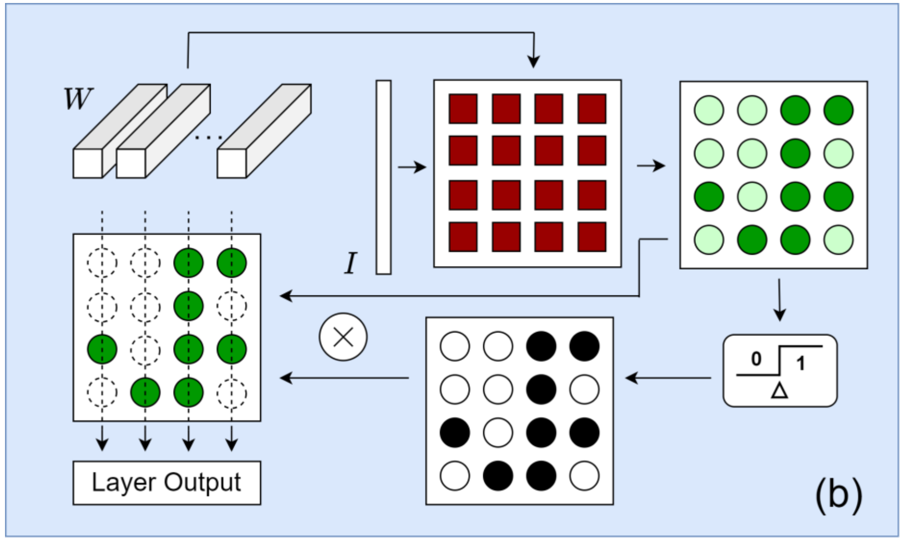
- The weights are divided into the MSB and LSB parts
- The mask filters OU regions related to importance partial sums on weight matrix





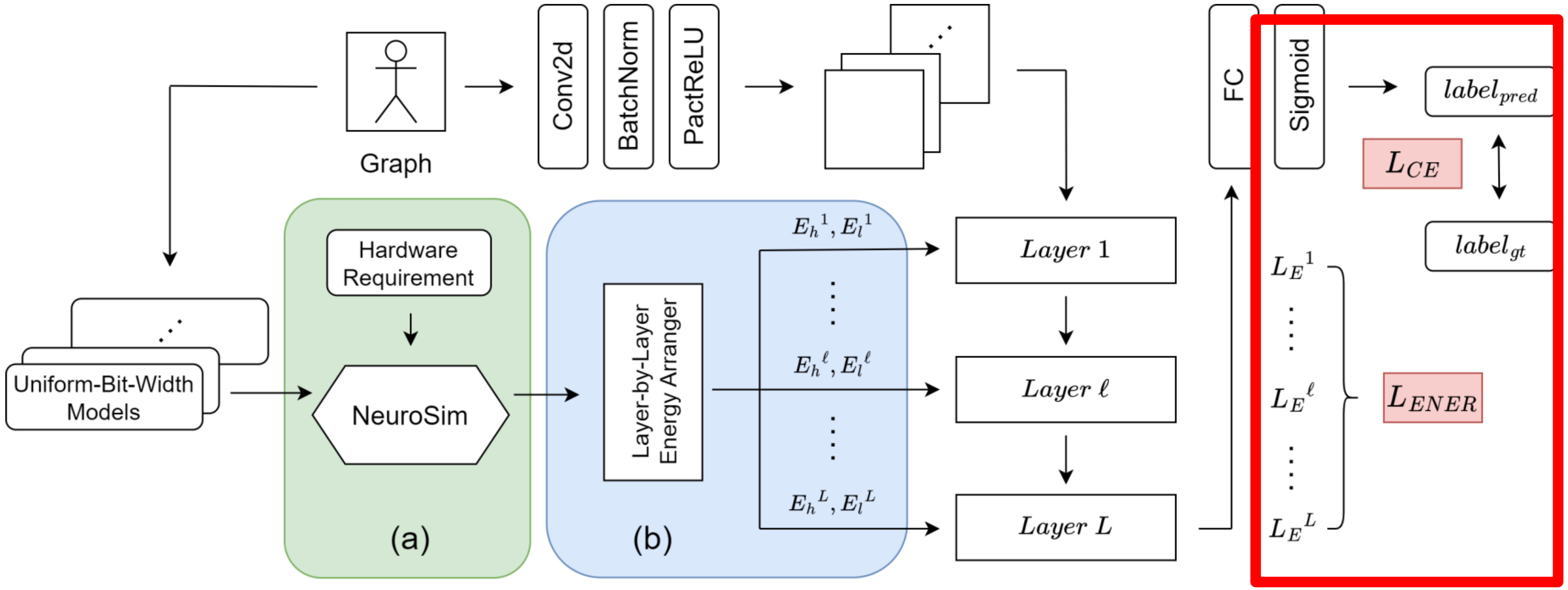
Proposed DE-C3 Framework

- A special case occurs when lb in bit width pair (hb, lb) is 0
- There is no LSB part of weights in this mode and the mask will be directly applied on the MSB partial sums to obtain the final output



Proposed DE-C3 Framework

❖ The overall objective function includes two parts of loss functions



Proposed DE-C3 Framework

- The loss function contains L_{CE} and L_{NENR}
- L_{CE} represents the cross-entropy loss between predicted and ground truth labels, while L_{NENR} represents the sum of layer-wise energy cost

- $\boxed{L_{ENER}} = \sum_{l=1}^{L} \boxed{{L_E}^l}$ $\boxed{{L_E}^l = {E_l}^l + ({E_h}^l - {E_l}^l) \times (1 - spar^l)}$

- $L_{total} = L_{CE} + \alpha \times \boxed{L_{ENER}}$

Proposed DE-C3 Framework

- The loss function contains L_{CE} and L_{NENR}
- L_{CE} represents the cross-entropy loss between predicted and ground truth labels, while L_{NENR} represents the sum of layer-wise energy cost

- $L_{ENER} = \sum_{l=1}^{L} L_E{}^l$ $\quad L_E{}^l = E_l{}^l + (E_h{}^l - E_l{}^l) \times (1 - spar^l)$

- $L_{total} = L_{CE} + \alpha \times L_{ENER}$

α acts as a controller

Experimental Results

Experimental Setting

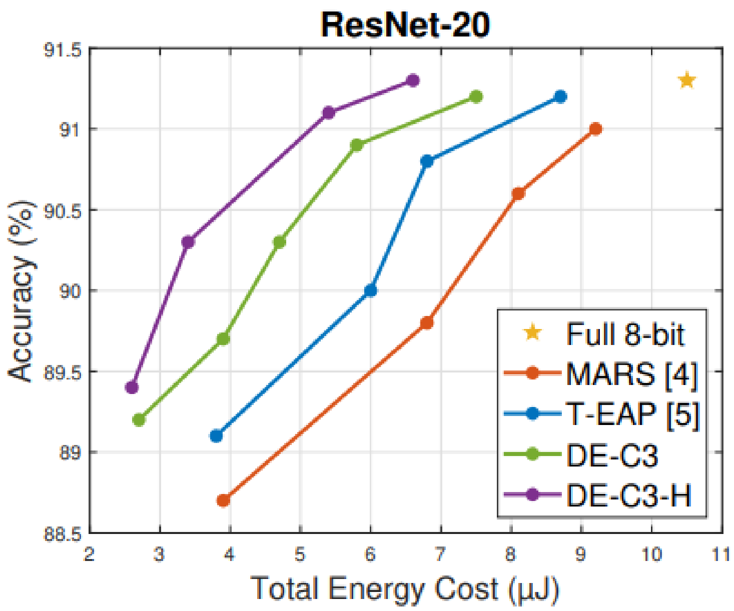
Hardware Setting	
Simulator	NeuroSim v1.3
OU Size	16
ADC Resolution	[4, 5, 6]
Bit Width	[8, 6, 4, 2, 1, 0]
Technology Node	22nm ReRAM

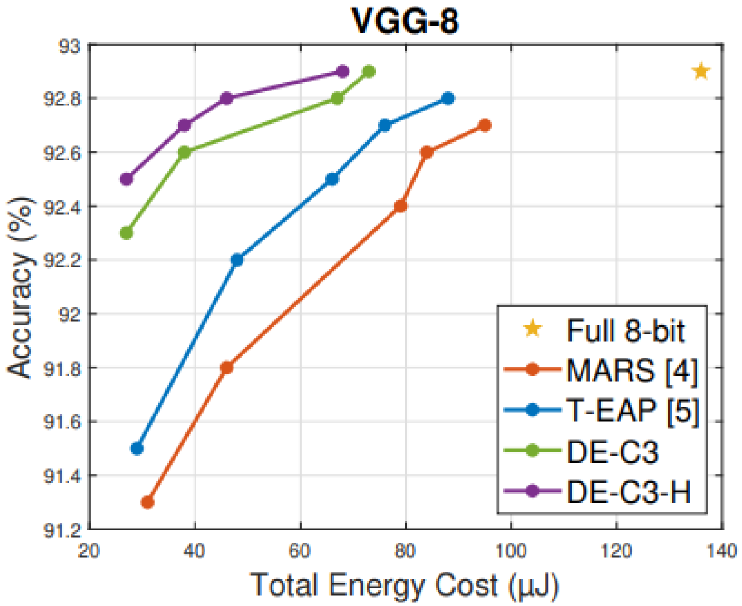
Software Setting	
Learning Rate	0.1
Batch Size	128
α (L_{ENER} portion)	[1e-7, 1e-8, 1e-9]
Epoch	100
Optimizer	SGD
Scheduler	$lr \times 0.1$ at 50^{th}, 75^{th} epoch

	OU-Based	Energy-Aware	Dynamic	Hybrid Compression
MARS	O	X	X	X
T-EAP	O	O	X	X
DE-C3	O	O	O	O

Experimental Results

- ❖ The performance of accuracy-energy trade-off
- ❖ Pareto front with a more upper-left position performs better





Conclusions

Conclusions

- CIM enables basic MAC operations to complete VMMs on the memory crossbar and break the memory wall bottleneck
- Our proposed DE-C3 framework leverages layer-wise energy consumption in dual-path processing to enable dynamic inference
  - The energy consumption can be reduced up to 2.3× for ResNet-20 and 3.4× for VGG-8
- More details can be found in our paper

References

[1] https://en.wikipedia.org/wiki/Von_Neumann_architecture

[2] T.-J. Yang, Y.-H. Chen, J. Emer, and V. Sze, “A method to estimate the energy consumption of deep neural networks,” in *2017 51st asilomar conference on signals, systems, and computers*. IEEE, 2017, pp. 1916–1920.

[3] C.-Y. Chang, Y.-C. Chuang, K.-C. Chou, and A.-Y. Wu, “T-eap: Trainable energy-aware pruning for nvm-based computing-in-memory architecture,” in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2022, pp. 78–81.

[4] Z. Wu, T. Nagarajan, A. Kumar, S. Rennie, L. S. Davis, K. Grauman, and R. Feris, “Blockdrop: Dynamic inference paths in residual networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8817–8826.

[5] Y. Zhang, R. Zhao, W. Hua, N. Xu, G. E. Suh, and Z. Zhang, “Precision gating: Improving neural network efficiency with dynamic dual-precision activations,” *arXiv preprint arXiv:2002.07136*, 2020.

This is the end of my presentation
Thanks for your listening

Q & A