Panacea: Novel DNN accelerator using Accuracy-preserving Asymmetric Quantization and Energy-saving Bit-slice Sparsity

Dongyun Kam[1], Myeongji Yun[1], Sunwoo Yoo[1], Seungwoo Hong[1], Zhengya Zhang[2], Youngjoo Lee[3]

[1]POSTECH, [2]University of Michigan, [3]KAIST

Outline

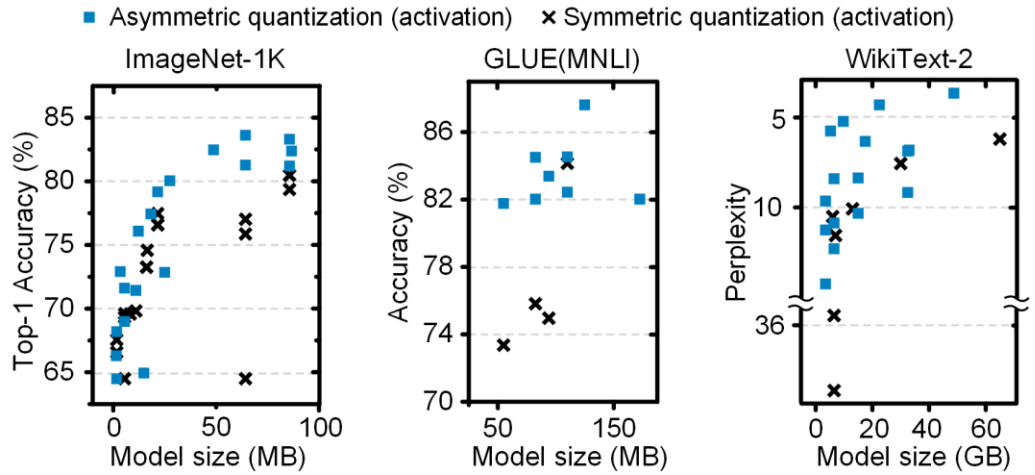
- Introduction
- Motivation
- Proposed work
- Experimental results
- Conclusion

Introduction

- **Energy-efficient DNN Inference in resource-limited devices**
  - Quantization enables INT GEMMs [1].
    - Symmetric quantization to weights for no additional overhead
    - Asymmetric quantization to activations for preserving accuracy

Symmetric quantization | Asymmetric quantization

$$\begin{aligned}\mathbf{W}\mathbf{x}+\mathbf{b} &\approx s_{\mathrm{W}}(\mathbf{W}_{\mathrm{int}})\,s_{\mathrm{x}}(\mathbf{x}_{\mathrm{uint}}-zp_{\mathrm{x}})+\mathbf{b}\\ &= s_{\mathrm{W}}s_{\mathrm{x}}(\mathbf{W}_{\mathrm{int}}\mathbf{x}_{\mathrm{uint}}-zp_{\mathrm{x}}\mathbf{W}_{\mathrm{int}}\mathbf{1}^{K\times 1}+\mathbf{b}_{\mathrm{int}})\\ &= s_{\mathrm{W}}s_{\mathrm{x}}(\mathbf{W}_{\mathrm{int}}\mathbf{x}_{\mathrm{uint}}+\hat{\mathbf{b}}_{\mathrm{int}}),\end{aligned}$$

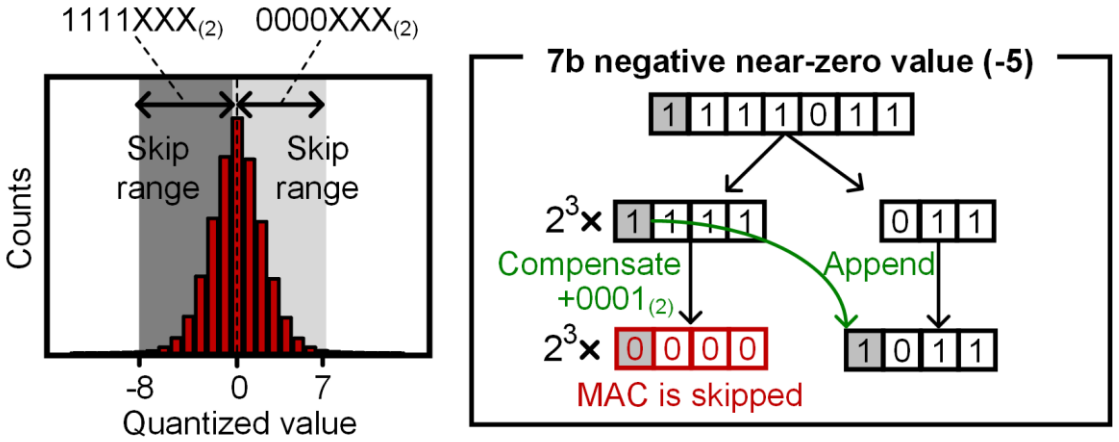


[1] M. Nagel et al., *Qualcomm White Paper*

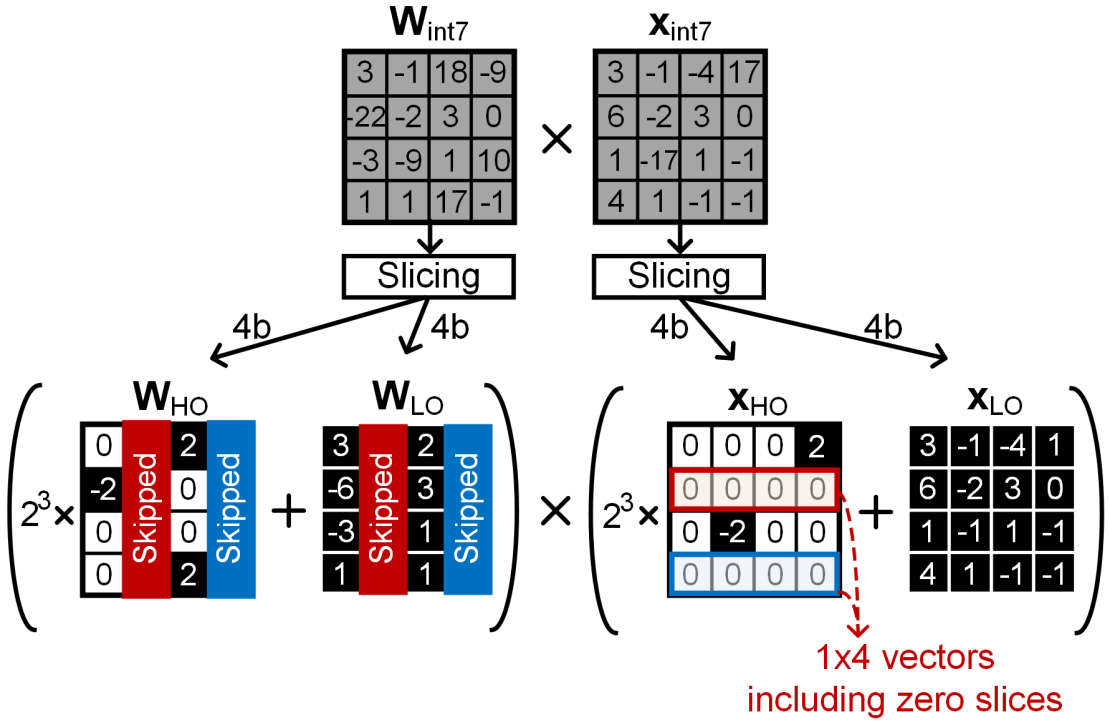
Introduction

- **Additional optimization:** Bit-slice GEMM
  - There are a lot of near-zero values for quantized weights (**W**) and activations (**X**).
  - Bit-slicing enables slice-level sparsity in High-Order (HO) slices.

7b → two 4b slices [53]



[53] D. Im et al., *MICRO 2023*



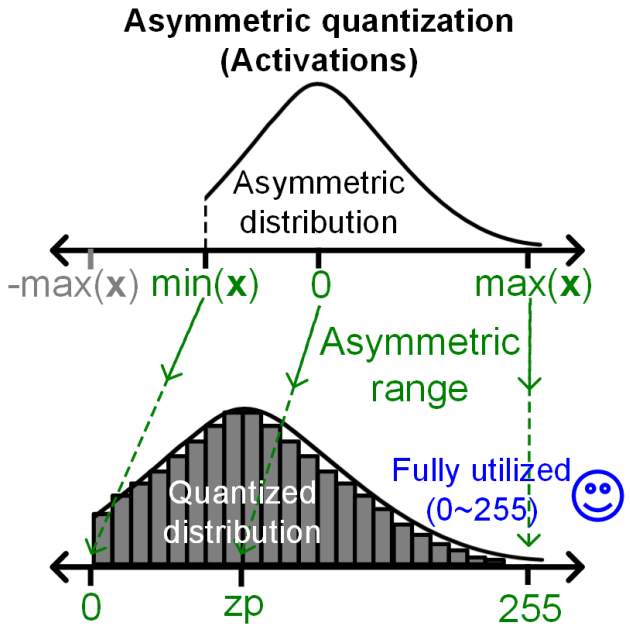
Outline

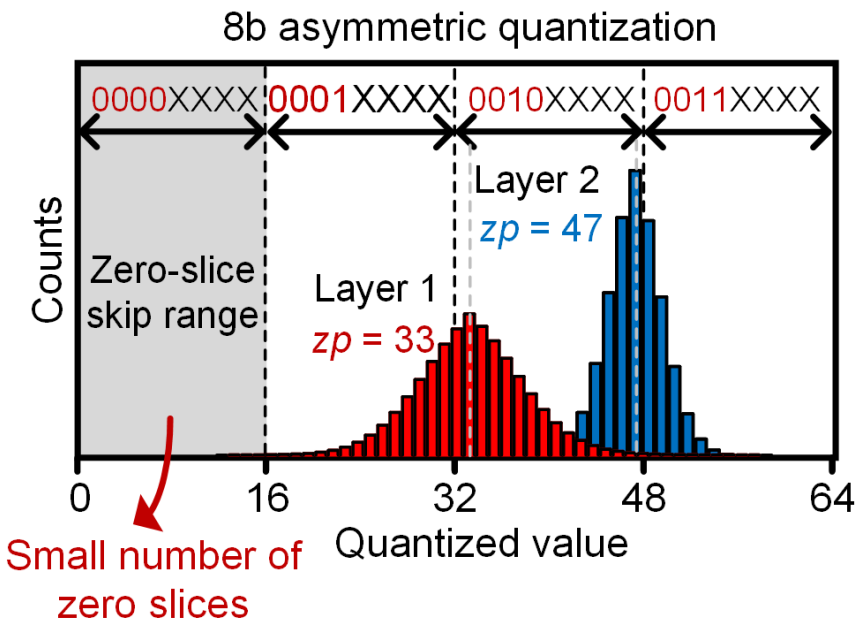
- Introduction
- **Motivation**
- Proposed work
- Experimental results
- Conclusion

Motivation

Challenges in previous works

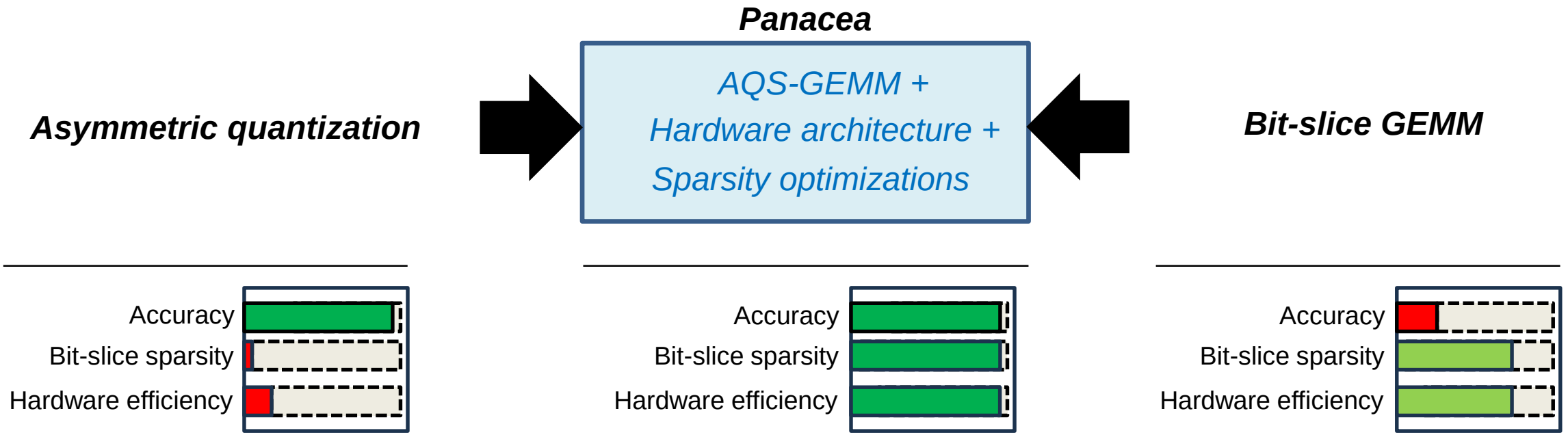
- Asymmetric quantization produces nonzero values in HO slices due to *zero point*.





Motivation

- **Our solution: Algorithm-Hardware co-design**
  - Asymmetrically Quantized bit-Slice GEMM (**AQS-GEMM**) skips nonzero HO slices.
  - ***Panacea*** is a hardware architecture to compute AQS-GEMM efficiently.
  - Two sparsity optimizations further enhance AQS-GEMM efficiency.



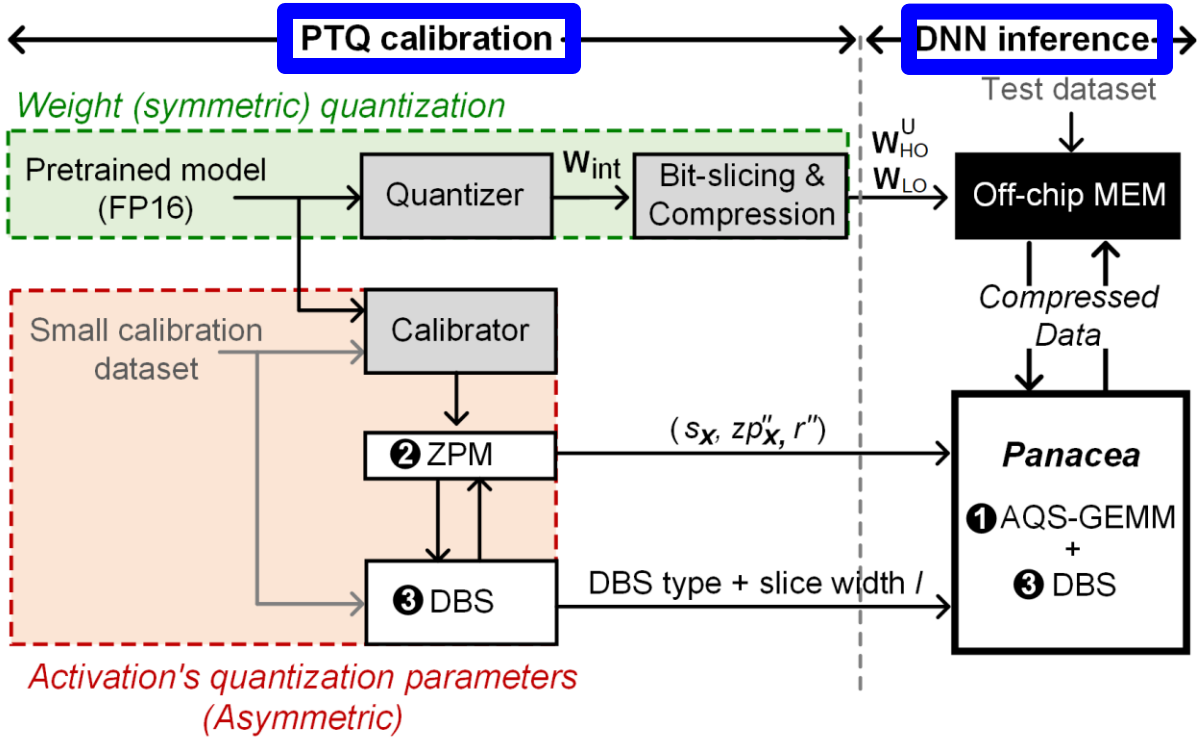
Outline

- Introduction
- Motivation
- **Proposed work**
- Experimental results
- Conclusion

Proposed work

Overview of our framework

- During DNN inference, ❶ enables bit-slice GEMM with asymmetric quantization.
- During PTQ calibration, ❷ + ❸ increases slice-level sparsity for energy-efficiency.

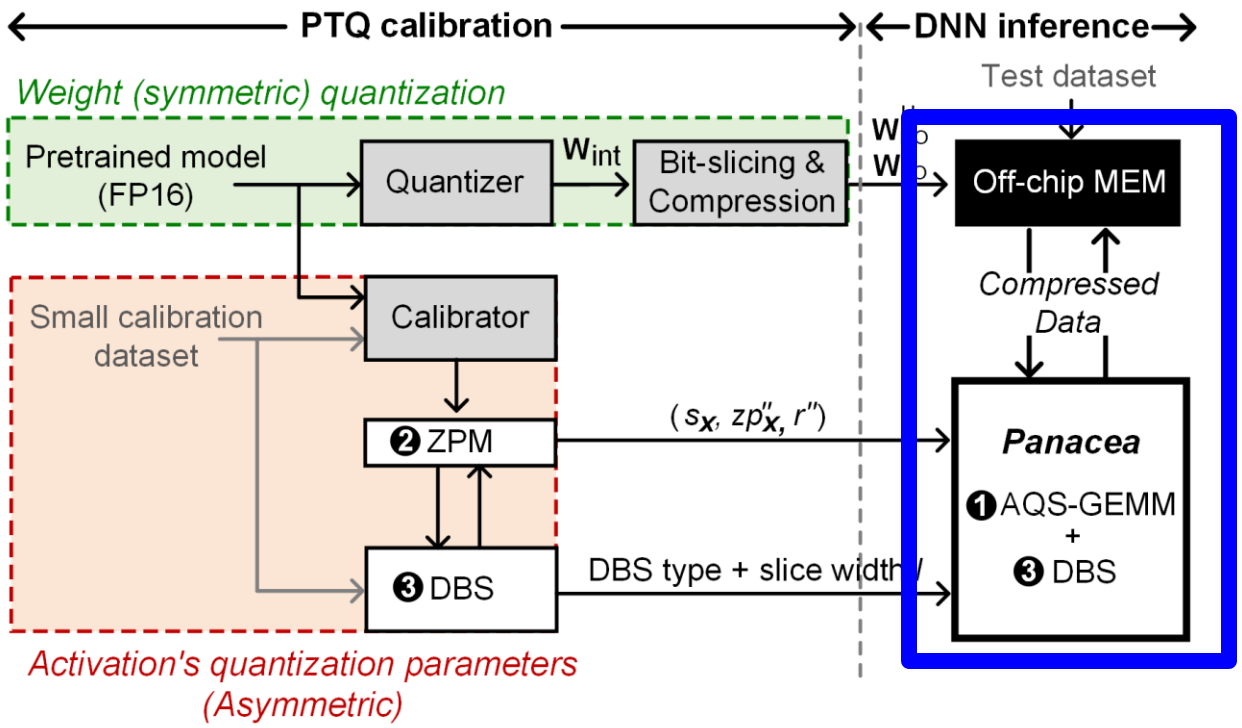


Panacea: Novel DNN accelerator using accuracy-preserving asymmetric quantization and energy-saving bit-slice sparsity

Proposed work

Overview of our framework

- During DNN inference, ❶ enables bit-slice GEMM with asymmetric quantization.
- During PTQ calibration, ❷ + ❸ increases slice-level sparsity for energy-efficiency.



Proposed work

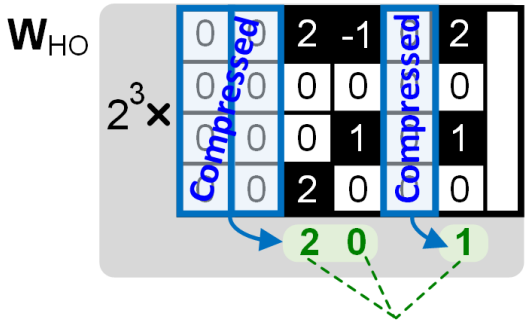
❶ AQS-GEMM: Asymmetrically Quantized bit-Slice GEMM

- 1. Compressing HO slices
  - $r = zp_{HO}$: A frequent nonzero HO slice in $\mathbf{x}_{\text{uint}}$

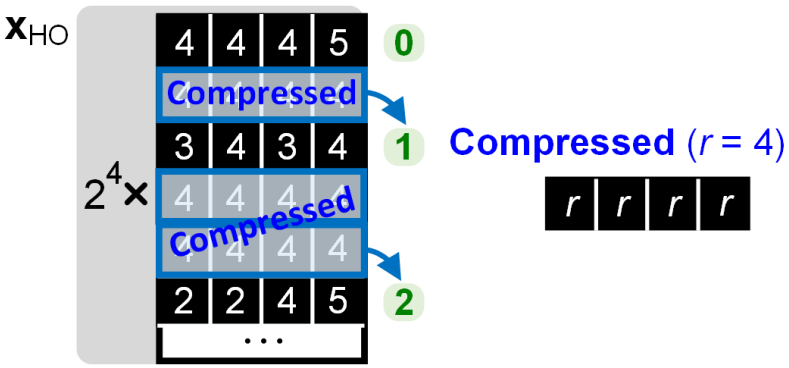
AQS-GEMM

$$\mathbf{Wx} + \mathbf{b} \approx s_W s_x(\mathbf{W}_{\text{int}}\mathbf{x}_{\text{uint}} + \hat{\mathbf{b}}_{\text{int}}) = s_W s_X((\mathbf{W}_{\text{HO}} + \mathbf{W}_{\text{LO}})(\mathbf{x}_{\text{HO}} + \mathbf{x}_{\text{LO}}) + \hat{\mathbf{b}}_{\text{int}})$$

Zero-slice's compression during calibration



Nonzero-slice's compression during inference



Proposed work

- ❶ **AQS-GEMM: Asymmetrically Quantized bit-Slice GEMM**
  - 2. Skipping MAC operations for compressed slices

AQS-GEMM

$$\mathbf{Wx}+\mathbf{b} \approx s_W s_x\left(\mathbf{W}_{int}\mathbf{x}_{uint}+\hat{\mathbf{b}}_{int}\right) = s_W s_X((\mathbf{W}_{HO}\mathbf{x}_{HO}+\mathbf{W}_{LO}\mathbf{x}_{HO}+\mathbf{W}_{HO}\mathbf{x}_{LO}+\mathbf{W}_{LO}\mathbf{x}_{LO})+\hat{\mathbf{b}}_{int})$$

Reformulation for nonzero skipping

$$(\mathbf{W}_{HO}+\mathbf{W}_{LO})\mathbf{x}_{HO} = (\mathbf{W}_{HO}+\mathbf{W}_{LO})\left(\mathbf{x}_{HO}^{\text{Uncompressed}}+\mathbf{x}_{HO}^{\text{Compressed}}\right)$$

→ Not skippable due to $\mathbf{x}_{HO}^{\text{Compressed}} \neq \mathbf{0}$

Proposed work

- ❶ **AQS-GEMM: Asymmetrically Quantized bit-Slice GEMM**
  - 2. Skipping MAC operations for compressed slices

AQS-GEMM

$$\mathbf{Wx}+\mathbf{b} \approx s_W s_x\left(\mathbf{W}_{\text{int}}\mathbf{x}_{\text{uint}}+\hat{\mathbf{b}}_{\text{int}}\right) = s_W s_X((\mathbf{W}_{\text{HO}}\mathbf{x}_{\text{HO}}+\mathbf{W}_{\text{LO}}\mathbf{x}_{\text{HO}}+\mathbf{W}_{\text{HO}}\mathbf{x}_{\text{LO}}+\mathbf{W}_{\text{LO}}\mathbf{x}_{\text{LO}})+\hat{\mathbf{b}}_{\text{int}})$$

Reformulation for nonzero skipping

$$(\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\mathbf{x}_{\text{HO}} = (\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\left(\mathbf{x}_{\text{HO}}^{\text{Uncompressed}}+\mathbf{x}_{\text{HO}}^{\text{Compressed}}\right)$$

→ Not skippable due to $\mathbf{x}_{\text{HO}}^{\text{Compressed}} \neq \mathbf{0}$

$$= (\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\left(\mathbf{x}_{\text{HO}}^{\text{Uncompressed}}+r\times\mathbf{J}^{\text{Compressed}}\right)$$

Additional memory access and index matching

Proposed work

❶ AQS-GEMM: Asymmetrically Quantized bit-Slice GEMM

- 2. Skipping MAC operations for compressed slices

AQS-GEMM

$$\mathbf{Wx}+\mathbf{b} \approx s_W s_x\left(\mathbf{W}_{\text{int}}\mathbf{x}_{\text{uint}}+\hat{\mathbf{b}}_{\text{int}}\right) = s_W s_X((\mathbf{W}_{\text{HO}}\mathbf{x}_{\text{HO}}+\mathbf{W}_{\text{LO}}\mathbf{x}_{\text{HO}}+\mathbf{W}_{\text{HO}}\mathbf{x}_{\text{LO}}+\mathbf{W}_{\text{LO}}\mathbf{x}_{\text{LO}})+\hat{\mathbf{b}}_{\text{int}})$$

Reformulation for nonzero skipping

$$(\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\mathbf{x}_{\text{HO}} = (\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\left(\mathbf{x}_{\text{HO}}^{\text{Uncompressed}}+\mathbf{x}_{\text{HO}}^{\text{Compressed}}\right)$$

→ Not skippable due to $\mathbf{x}_{\text{HO}}^{\text{Compressed}} \neq \mathbf{0}$

$$= (\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\left(\mathbf{x}_{\text{HO}}^{\text{Uncompressed}}+r\times\mathbf{J}^{\text{Compressed}}\right)$$

$$= (\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\left(\mathbf{x}_{\text{HO}}^{\text{Uncompressed}}+r\times(\mathbf{1}^{\text{K}\times N}-\mathbf{J}^{\text{Uncompressed}})\right)$$

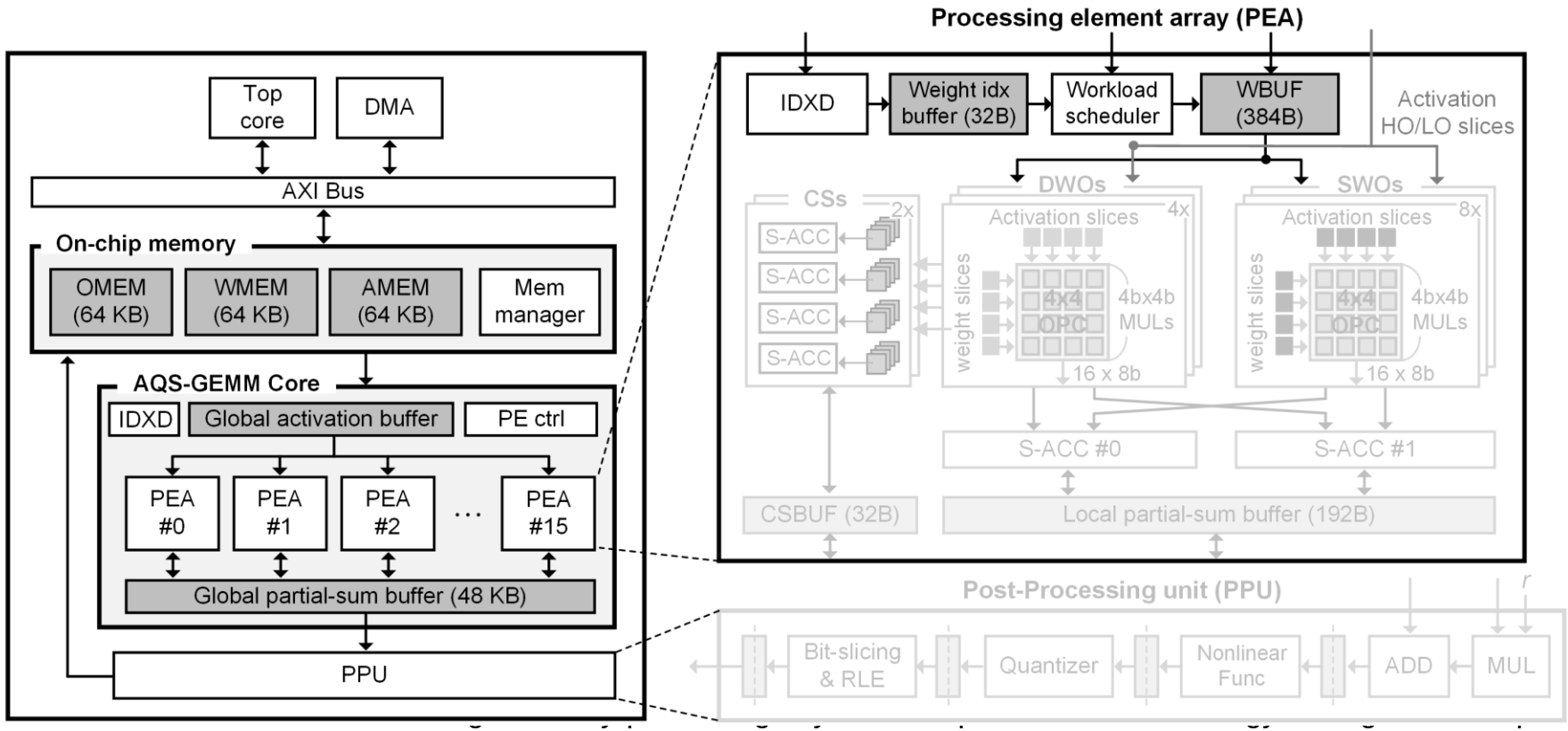
$$= (\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\mathbf{x}_{\text{HO}}^{\text{Uncompressed}} - r(\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\mathbf{J}^{\text{Uncompressed}} + \mathbf{b}'$$

Same index matching & data reuse ➔ **Efficient compensation term**

Proposed work

❶ *Panacea*'s hardware architecture

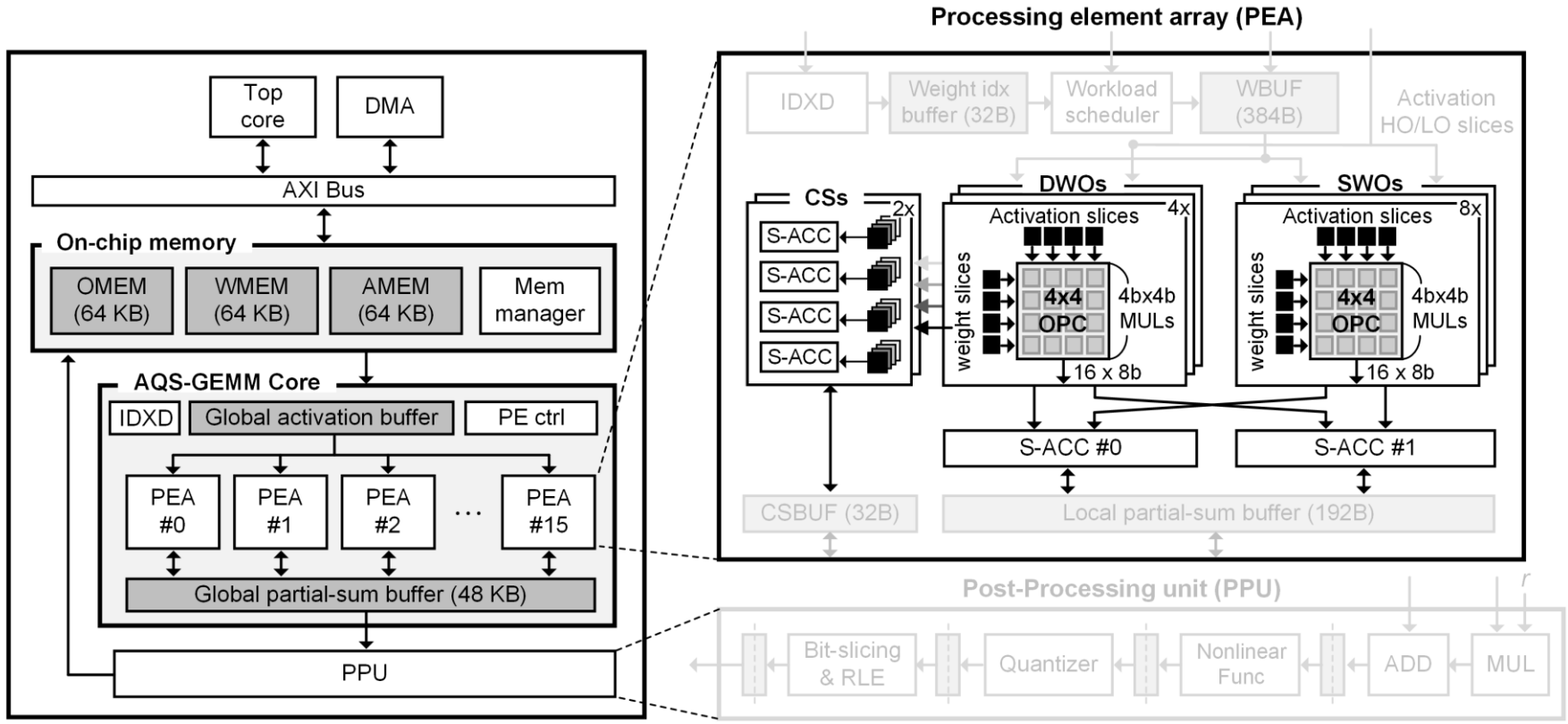
- Each PEA decodes RLE indices for uncompressed weight slices.
- The workload scheduler produces sparse/dense workloads for the AQS-GEMM.



Proposed work

❶ *Panacea*'s hardware architecture

- Each PEA dedicates two types of outer-product operators (OPC).
  - Dynamic Workload Operators (**DWOs**): Processing sparse-related bit-slice computations
  - Static Workload Operators (**SWOs**): Processing dense-related bit-slice computations

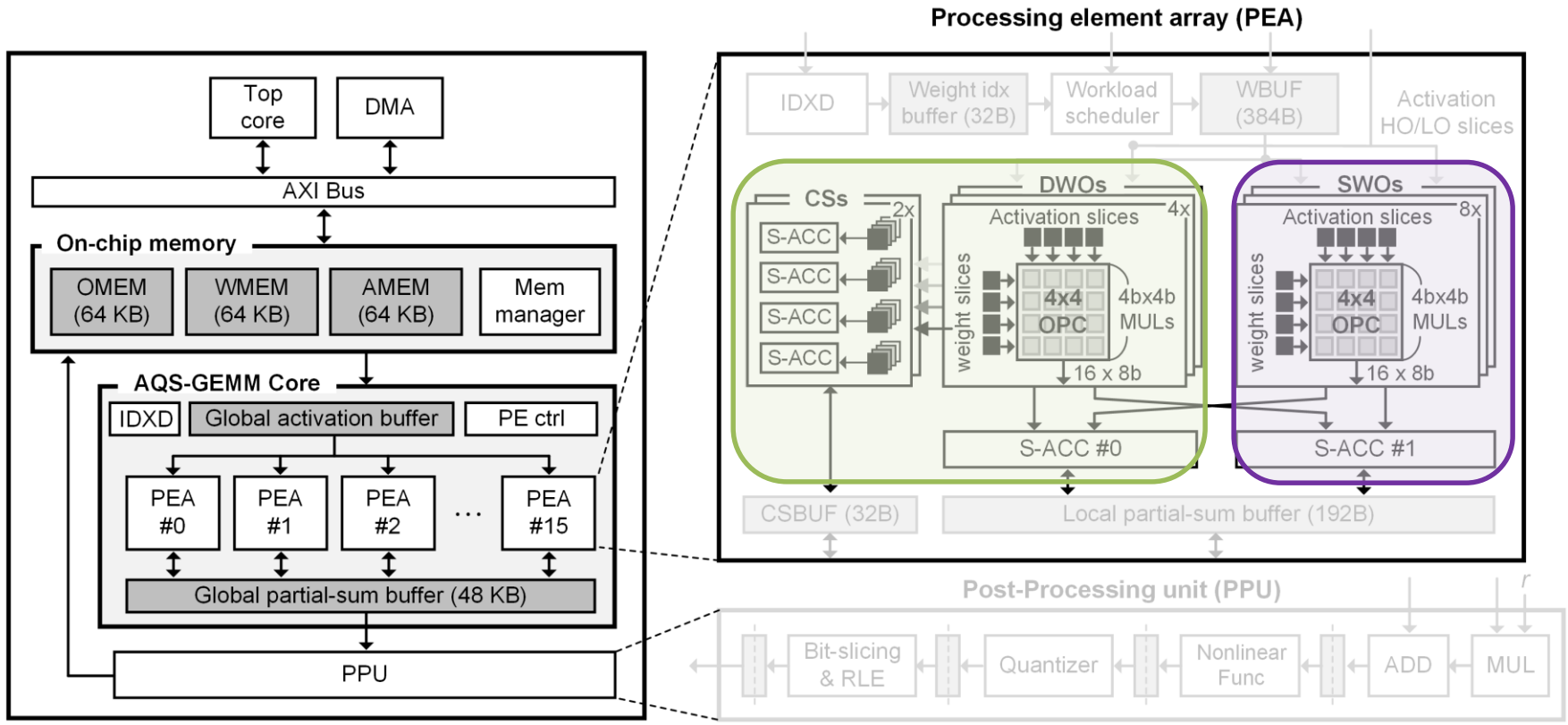


Proposed work

❶ *Panacea*'s hardware architecture

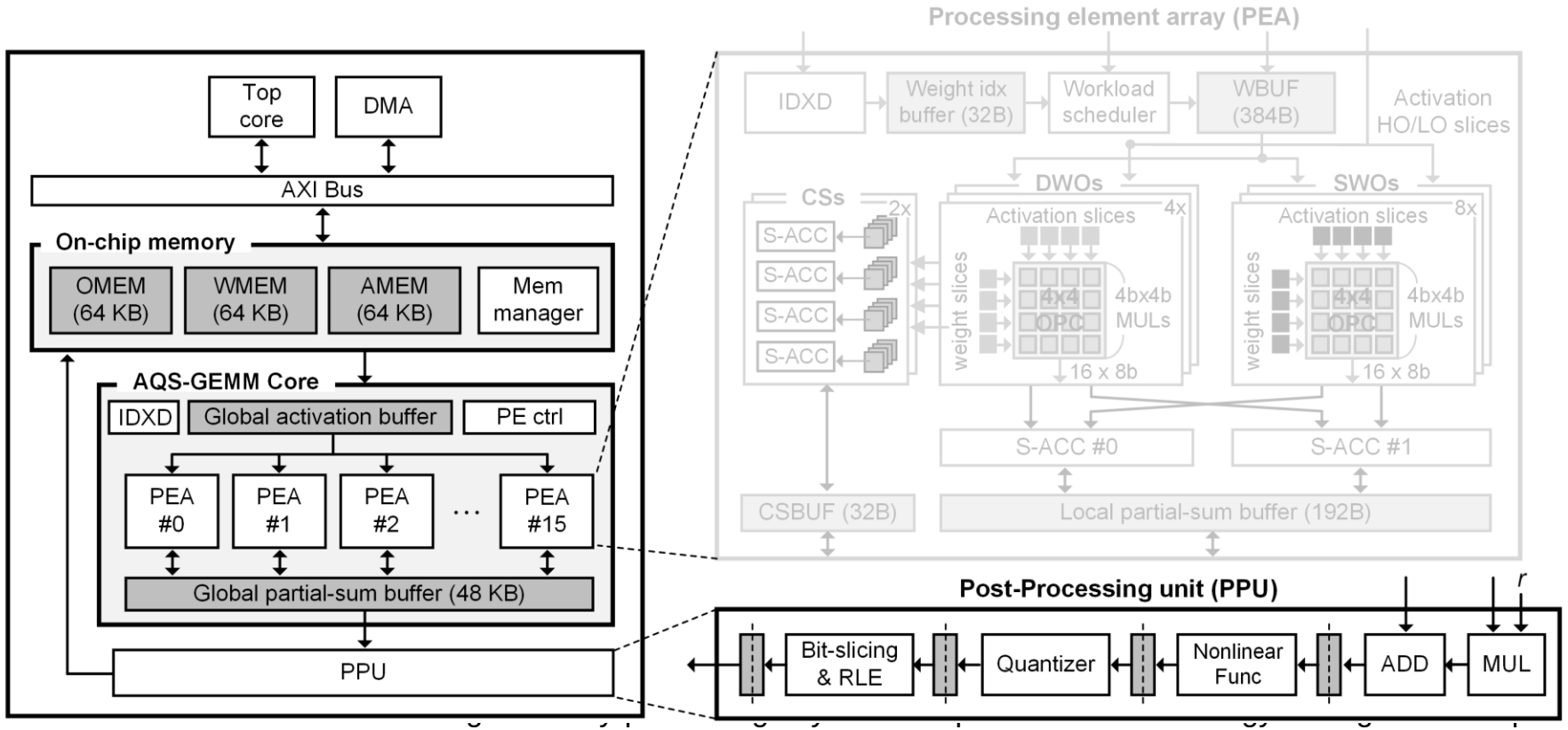
$$\mathbf{W}\mathbf{x}+\mathbf{b} \approx s_W s_x\left(\mathbf{W}_{\text{int}}\mathbf{x}_{\text{uint}}+\hat{\mathbf{b}}_{\text{int}}\right) = s_W s_X((\mathbf{W}_{\text{HO}}\mathbf{x}_{\text{HO}}+\mathbf{W}_{\text{LO}}\mathbf{x}_{\text{HO}}+\mathbf{W}_{\text{HO}}\mathbf{x}_{\text{LO}}+\mathbf{W}_{\text{LO}}\mathbf{x}_{\text{LO}})+\hat{\mathbf{b}}_{\text{int}})$$

DWOs + Compensators (CSs) SWOs



Proposed work

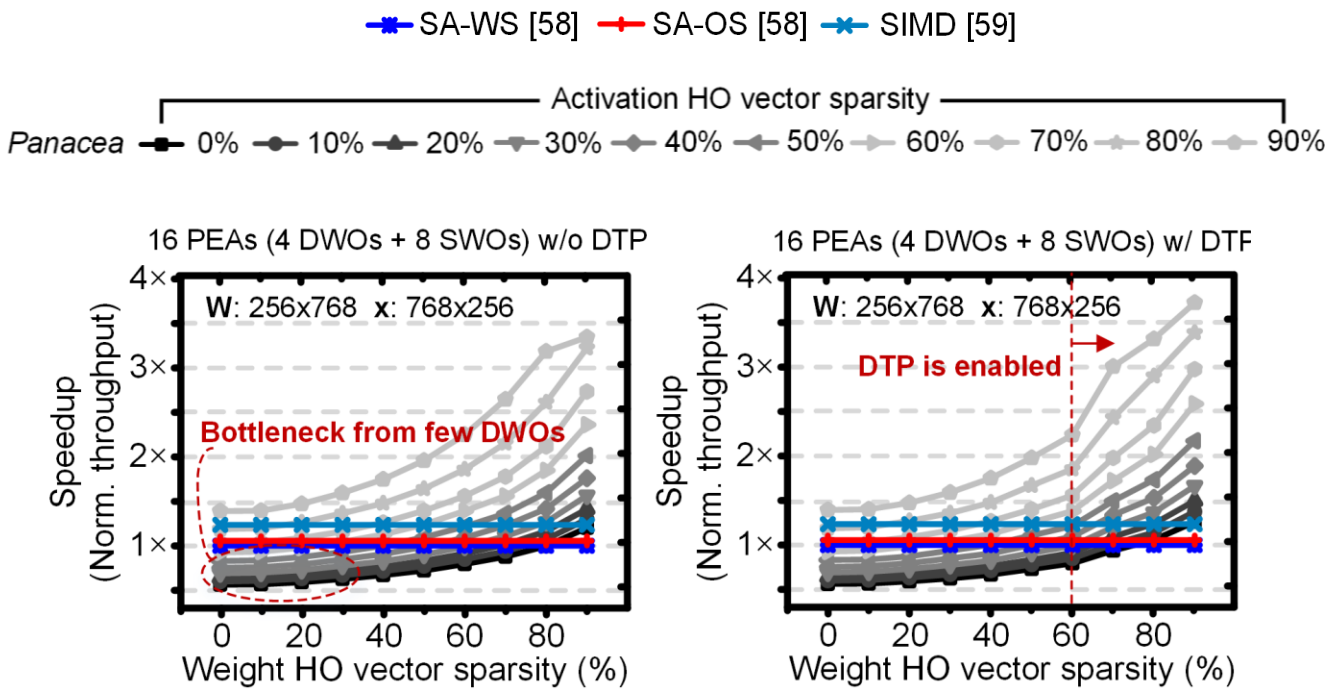
- ❶ ***Panacea*'s hardware architecture**
  - The Post-Processing Unit (PPU) consists of multiple
  - It compresses nonzero slices frequently observed.



Proposed work

❶ Throughput evaluation on slice-level sparsity

- 16 PEAs, each of which has 4 DWOs and 8 SWOs.
- **Double tile processing (DTP)** allows each PEA to process two weight tiles at high sparsity**,** addressing the low utilization of DWOs.

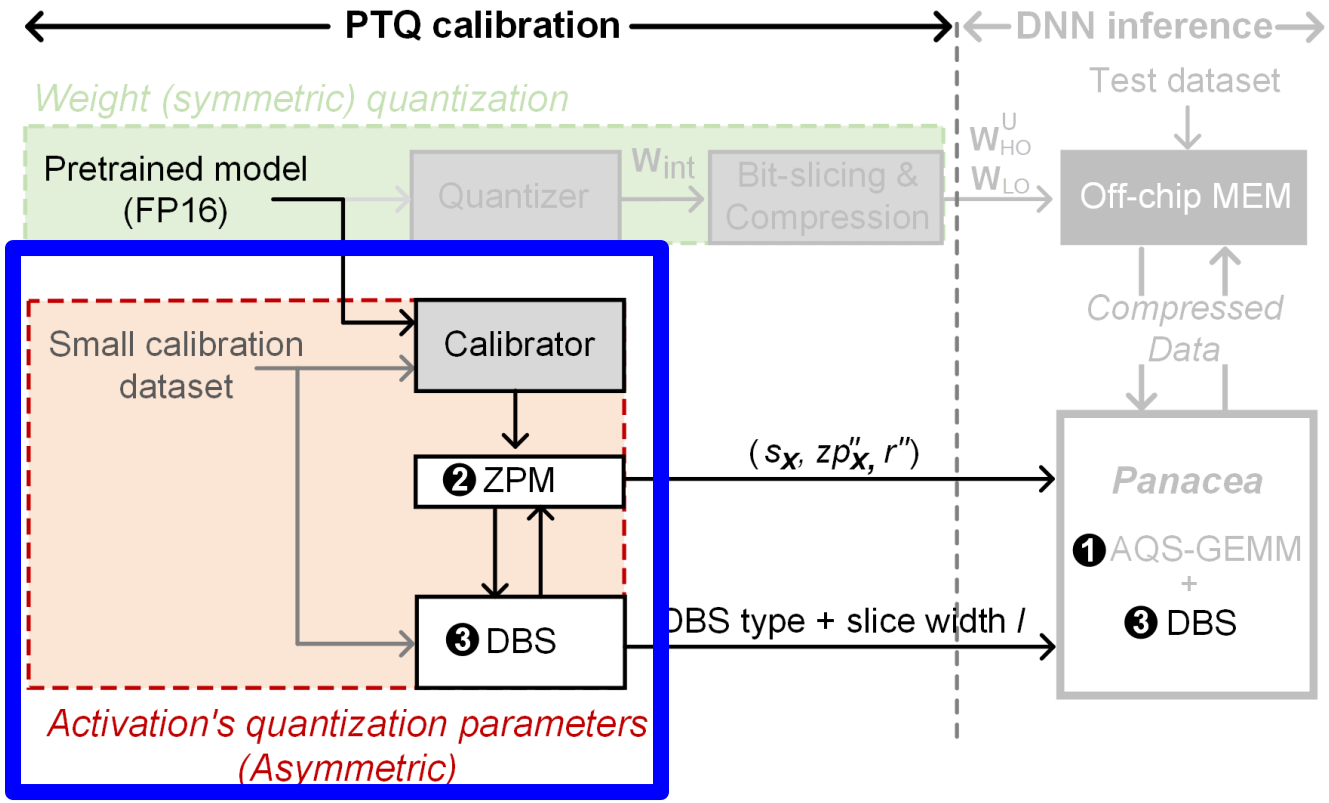


[58] B. Asgari et al., *ICCD 2020*
[59] B. Keller et al., *JSSC* 2023

Proposed work

Overview of our framework

- During inference, ❶ enables bit-slice GEMM with asymmetric quantization.
- During calibration, ❷ + ❸ increases slice-level sparsity for energy-efficiency.



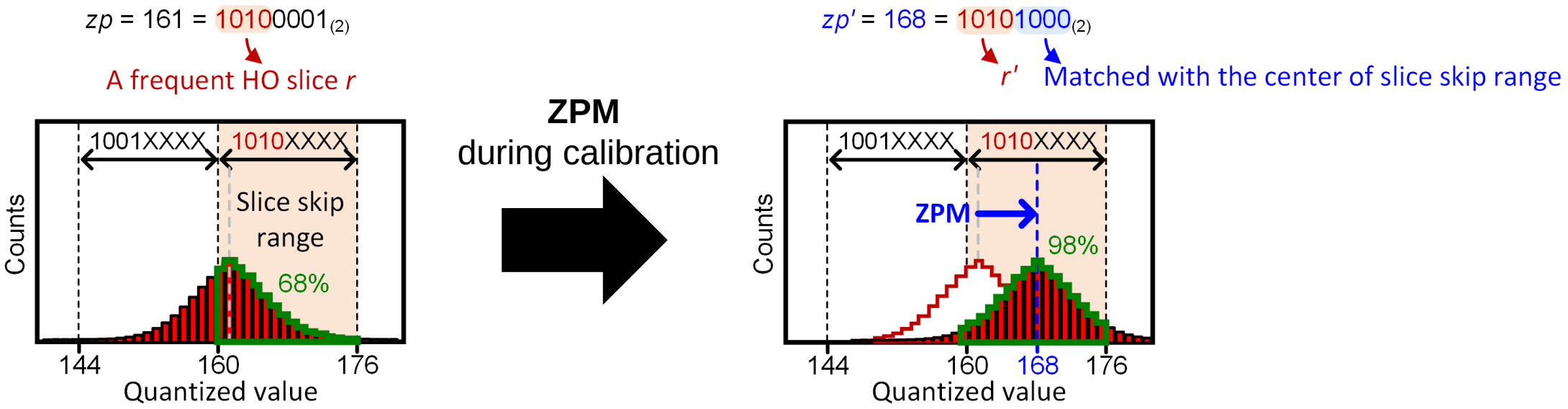
Panacea: Novel DNN accelerator using accuracy-preserving asymmetric quantization and energy-saving bit-slice sparsity

Proposed work

❷ Enhancing slice-sparsity : Zero-point manipulation (ZPM)

- Matching the center of quantized distribution with the center of skip range

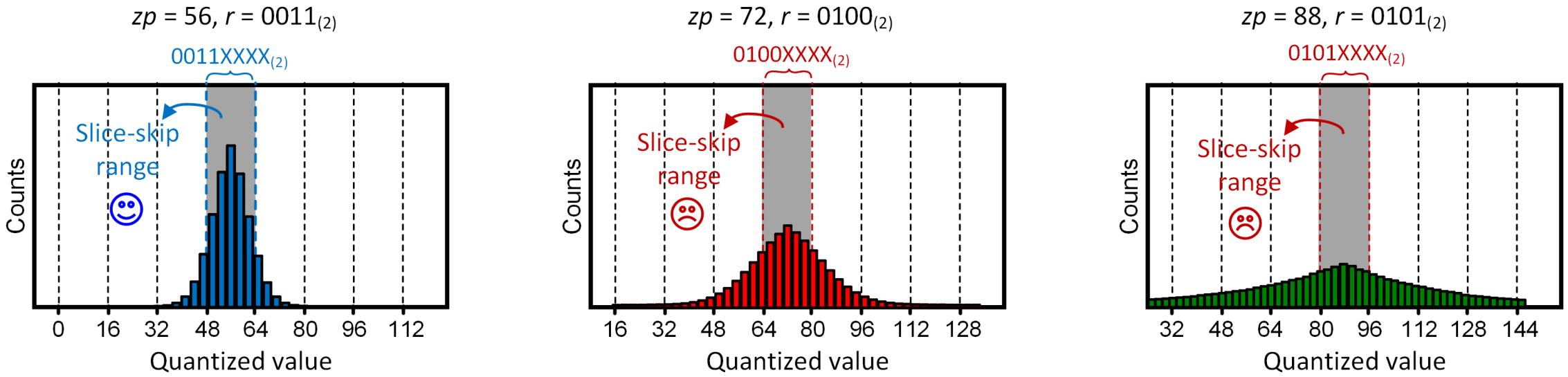
$zp' = 2^l \lfloor zp/2^l \rceil + 2^{l-1}$, where l indicates the bit-width of LO slice and $r' = r$



Proposed work

❸ Enhancing slice-sparsity : Distribution-based slicing (DBS)

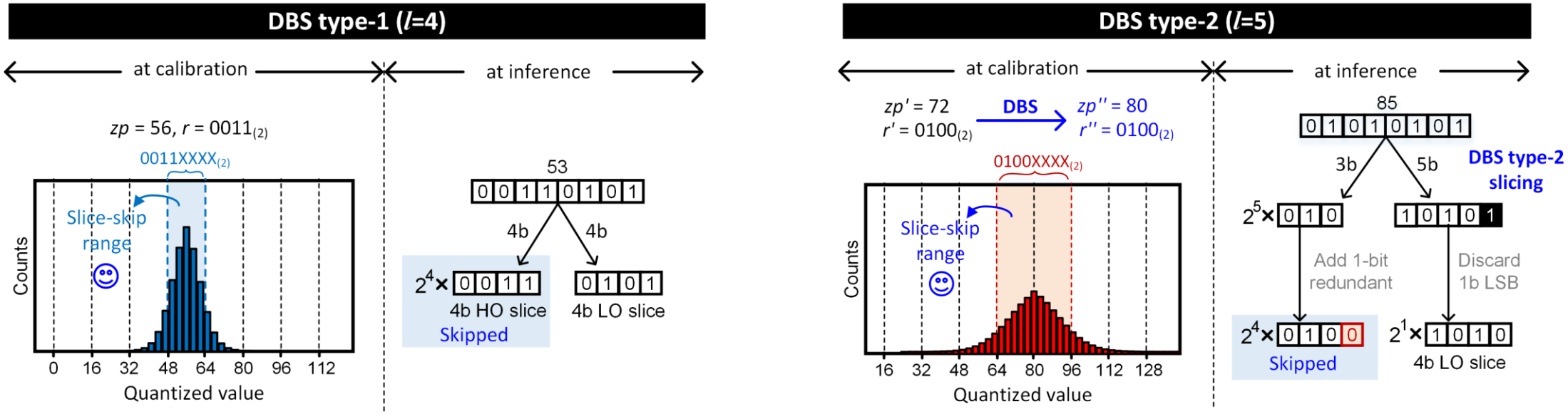
- Some activation's distributions spread over a wide range.
  - Fewer quantized values fall within the skip range (low slice sparsity).



Proposed work

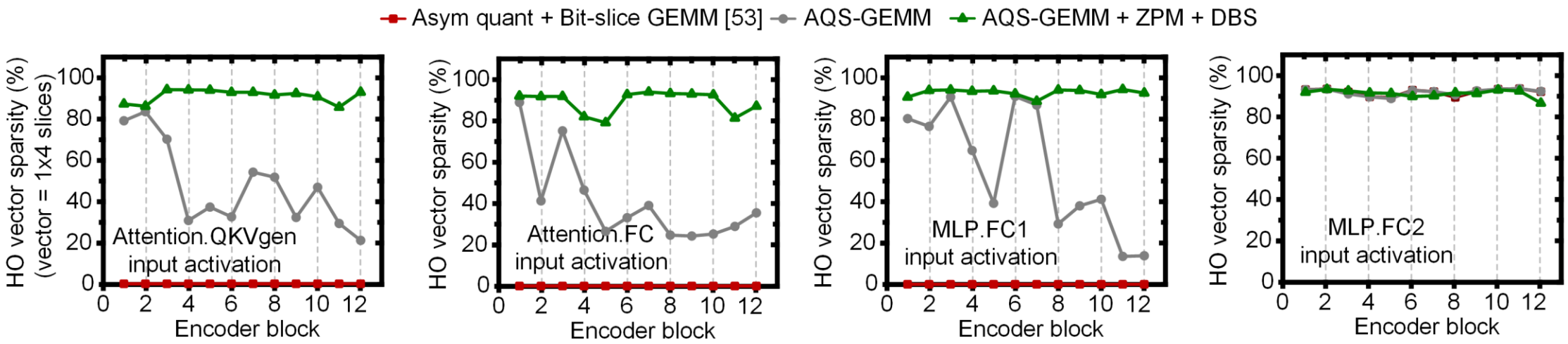
❸ Enhancing slice-sparsity : Distribution-based slicing (DBS)

- Categorizing distributions into three DBS types
- Applying different bit slicing rules to different DBS types
- Keeping the bit-width of slices at 4 bits without additional hardware



Proposed work

- **❷ + ❸ Slice-vector sparsity evaluation on DNN models**
  - Asymmetric quantization produces nonzero HO slices.
  - The AQS-GEMM utilizes slice-vector sparsity caused by frequent nonzero slices.

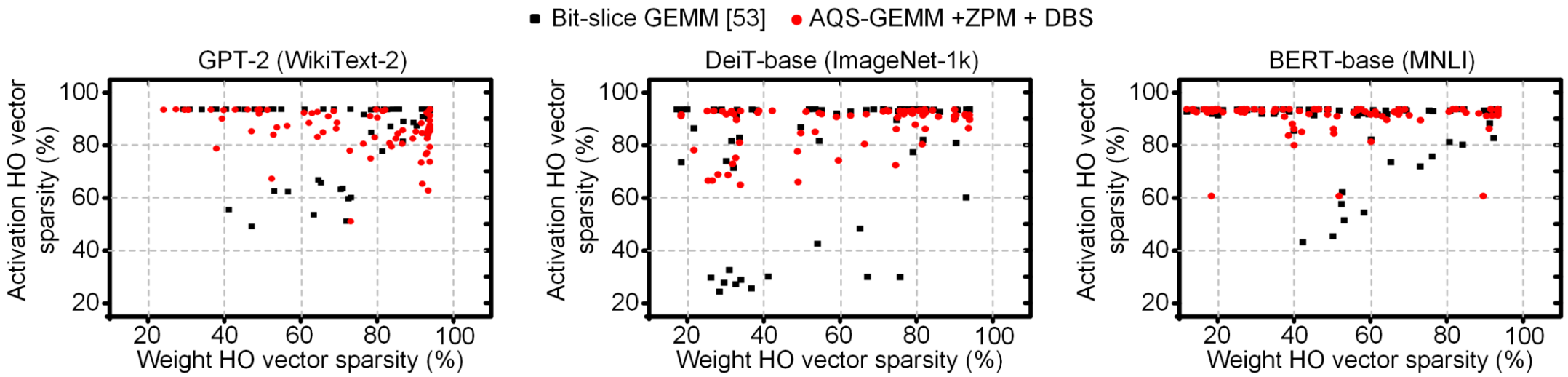


[53] D. Im et al., *MICRO 2023*

Proposed work

❷ + ❸ Slice-vector sparsity evaluation on DNN models

- During inference, we evaluate the slice-vector sparsity for all weights and activations.
- The proposed methods effectively increases slice-vector sparsity.



[53] D. Im et al., *MICRO 2023*

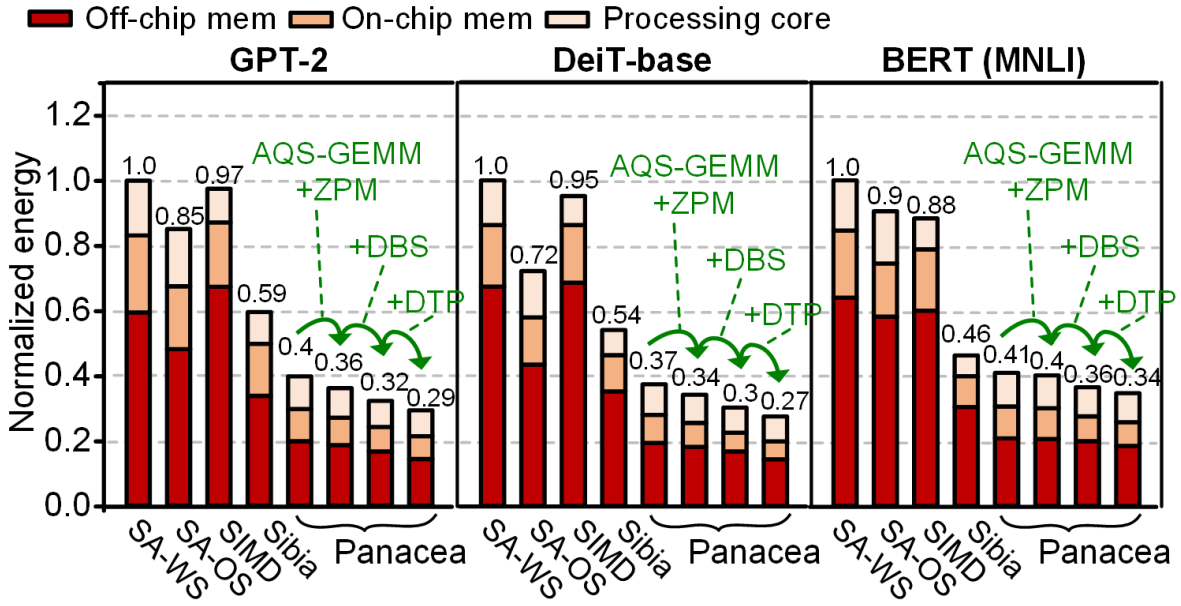
Outline

- Introduction
- Motivation
- Proposed work
- **Experimental results**
- Conclusion

Experimental Results

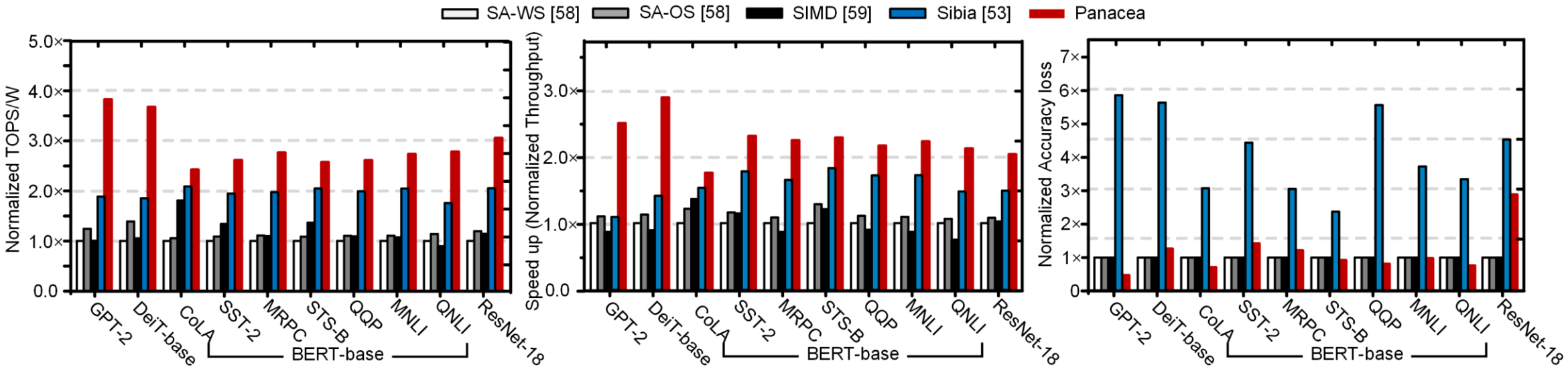
Accelerator's energy consumption

- We evaluate the recent accelerators (Systolic array (SA), SIMD, *Sibia*, *Panacea*).
  - *Panacea* reduces energy consumption due to the reduction of external memory accesses, on-chip memory accesses, and MAC operations.



Experimental Results

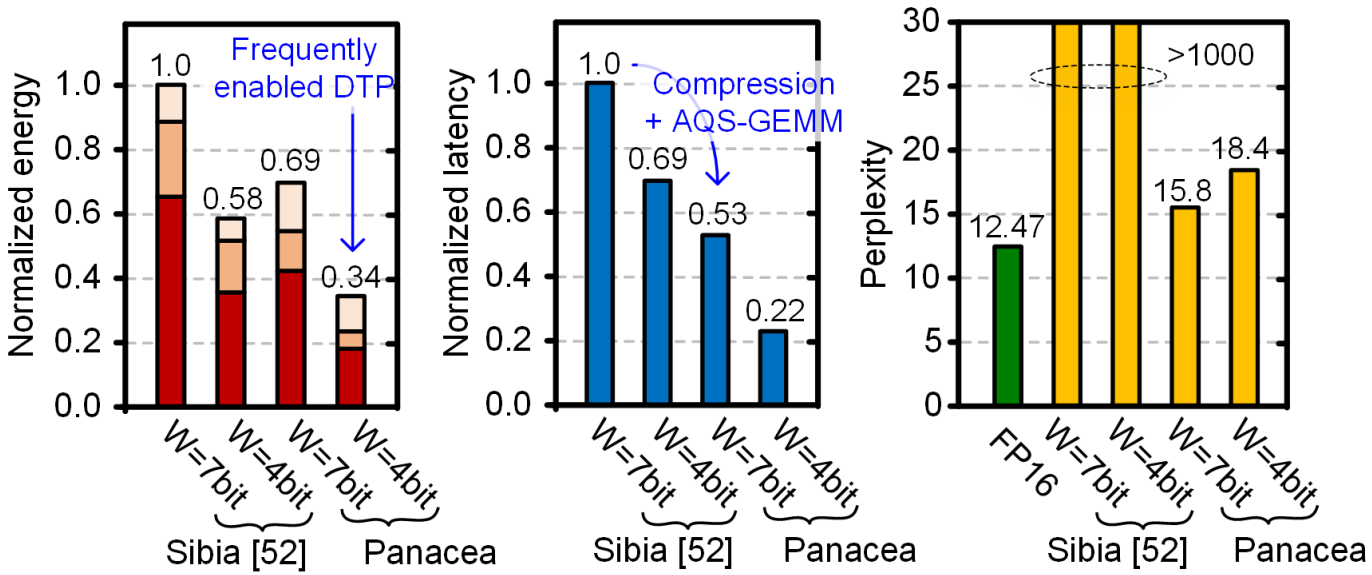
- **Accelerator's performance for DNN models**
  - Accelerators: Systolic arrary (SA), SIMD (dense), *Sibia,* and *Panacea*
  - Evaluation metrics: TOPS/W, Throughput, and Accuracy loss.
  - Models : DeiT-base, GPT-2, BERT-base, and RestNet-18



[53] D. Im et al., *MICRO 2023* [58] B. Asgari et al., *ICCD 2020* [59] B. Keller et al., *JSSC* 2023

Experimental Results

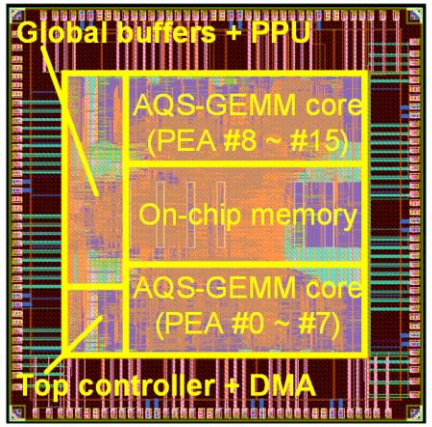
- **Accelerator's performance with low-bit weight quantization**
  - The bit-slice accelerators (*Sibia & Panacea*) are scalable for lower bit-precisions.
    - *Panacea* can split (3n+4)-bit weight into (n+1) 4b slices (n≥0).
    - Using 4b weight quantization achieves better hardware performance than 7b quantization.



Experimental Results

Implementation results

- *Panacea* is implemented in 28nm CMOS technology.
- It achieves better energy-efficiency than previous bit-slice designs, while providing better perplexity on GPT-2 due to asymmetric quantization.



Design	***Panacea***	*Sibia* [53]	*LUTein* [56]
Technology	28nm*	28nm**	28nm**
Frequency (MHz)	250	250	250
# of 4bx4b MUL	3072	1536	2048
Overall area (mm^2)	2.11	-	-
Core area (mm^2)	1.31	1.069	0.724
Throughput (TOPS)	1.268	0.770	0.907
Efficiency (TOPS/W)	12.5	7.65	10.3
Perplexity (GPT-2)	27.6	103.4***	103.4***

*FD-SOI technology **CMOS technology ***7-bit sym.quantization for all activations

Outline

- Introduction
- Motivation
- Proposed work
- Experimental results
- **Conclusion**

Conclusion

- **In summary, Panacea is a novel DNN accelerator that:**
  - ✓ Efficiently integrates asymmetric quantization with bit-slice sparsity.
    ✓ Optimizes data reuse with workload-specific operators.
    ✓ Employs specialized dataflows for enhanced performance.
    ✓ Achieves strong energy efficiency and inference performance across various DNN models.

Thank you for listening

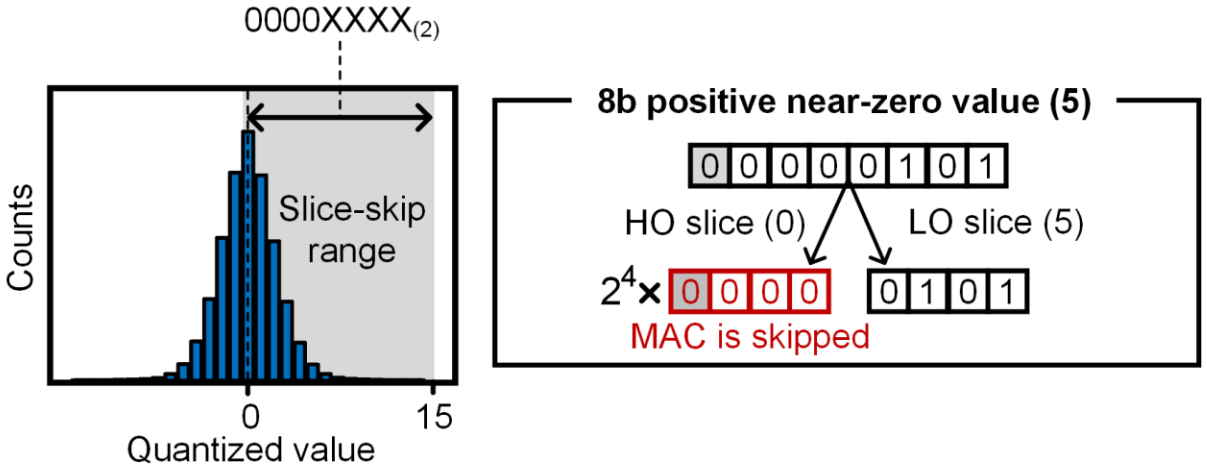
Dongyun Kam, Ph.D.
Postdoctoral Researcher, POSTECH, Republic of Korea
rkaehddbs@postech.ac.kr

Appendix

Introduction

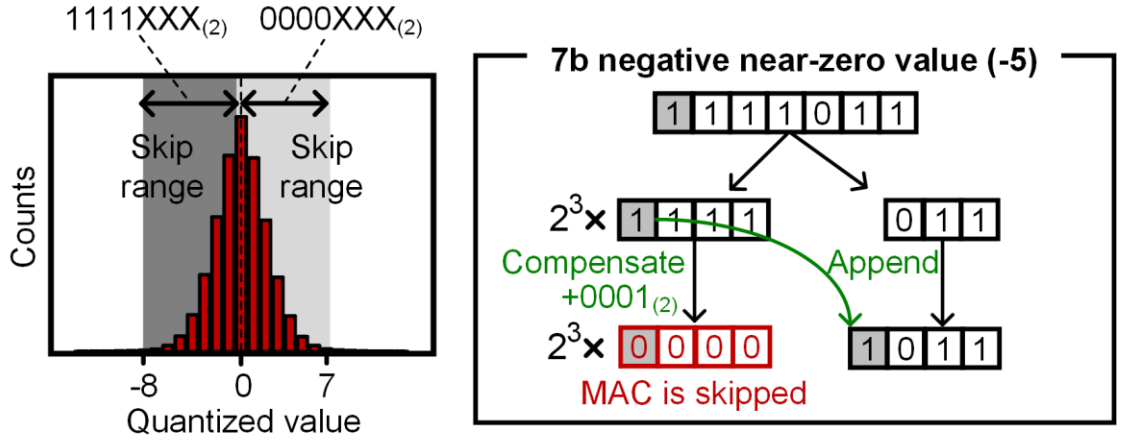
- **Additional optimization: Bit-slicing**
  - There are a lot of near-zero values for quantized weights (**W**) and activations (**X**).
  - Bit-slicing enables slice-level sparsity in High-Order (HO) slices.

8b → two 4b slices [54]



[54] G. Shomron et al., *MICRO 2020*

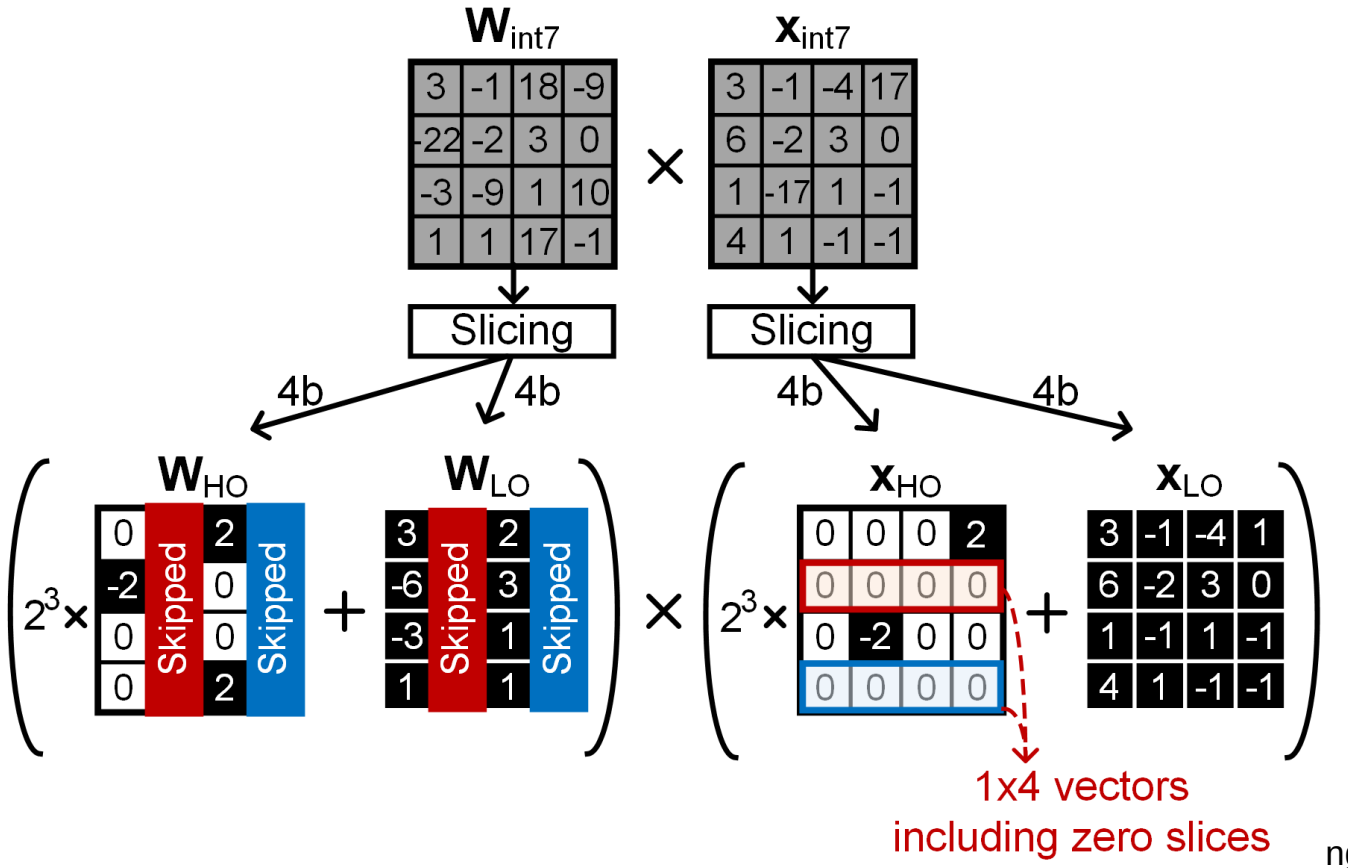
7b → two 4b slices [53]



[53] D. Im et al., *MICRO 2023*

Introduction

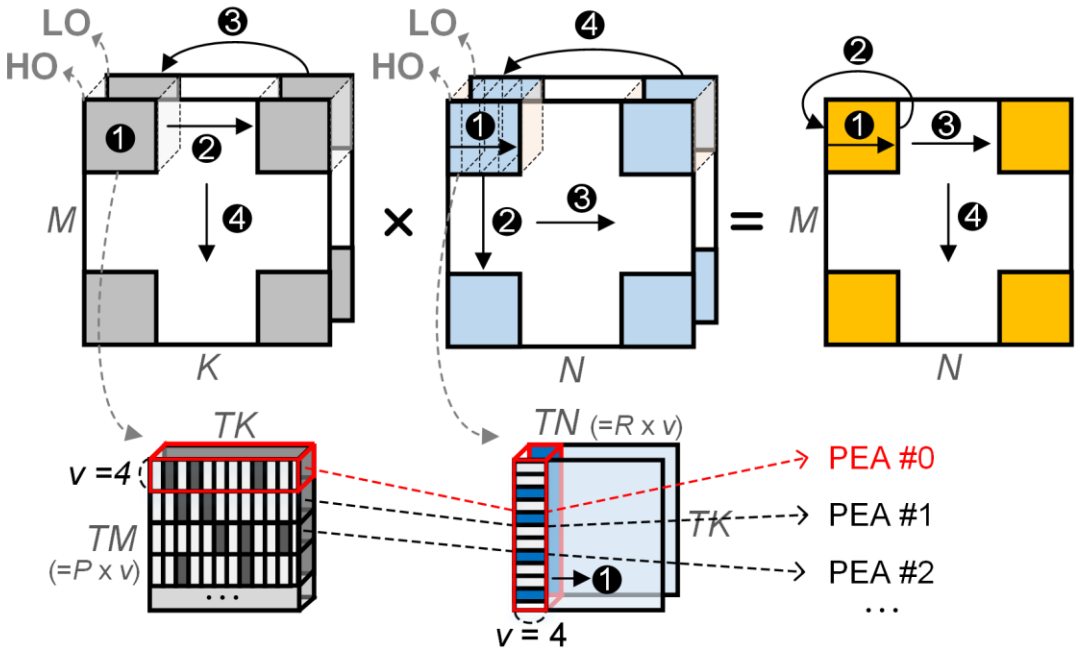
- **Bit-slice GEMM**
  - This is an example of bit-slice GEMM with symmetric quantization to **W** and **X**.
  - The bit-slice GEMM improves energy-efficiency by skipping the redundant OPs.



Proposed work

❶ *Panacea*'s dataflow

- To maximize data reuse, *Panacea* supports a tiled AQS-GEMM.
  - Output stationary in terms of EMA + Weight stationary in terms of PEAs



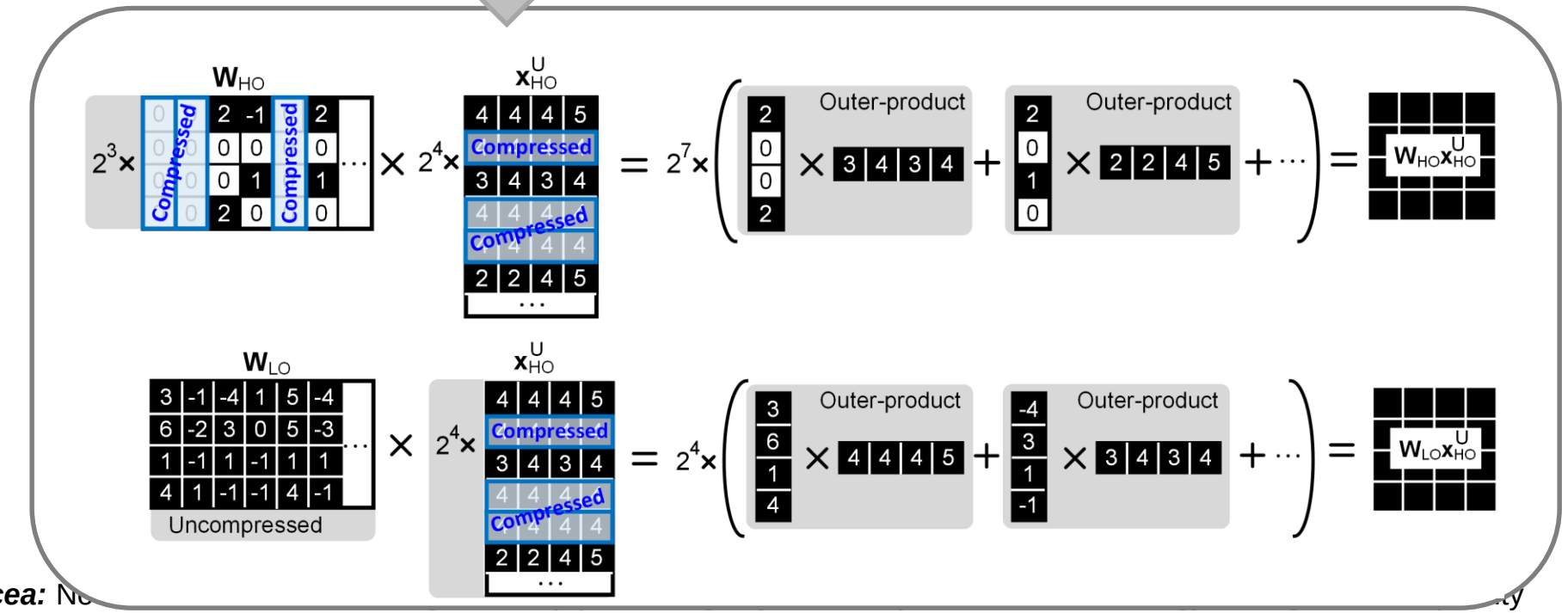
Proposed work

❶ Asymmetrically Quantized bit-Slice GEMM (AQS-GEMM)

AQS-GEMM

$$\mathbf{W}\mathbf{x}+\mathbf{b} \approx s_W s_x\left(\mathbf{W}_{\text{int}}\mathbf{x}_{\text{uint}}+\hat{\mathbf{b}}_{\text{int}}\right) = s_W s_X((\mathbf{W}_{\text{HO}}\mathbf{x}_{\text{HO}}+\mathbf{W}_{\text{LO}}\mathbf{x}_{\text{HO}}+\mathbf{W}_{\text{HO}}\mathbf{x}_{\text{LO}}+\mathbf{W}_{\text{LO}}\mathbf{x}_{\text{LO}})+\hat{\mathbf{b}}_{\text{int}})$$

$$(\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\mathbf{x}_{\text{HO}}^{\text{Uncompressed}} - r(\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\mathbf{J}^{\text{Uncompressed}} + \mathbf{b}'$$



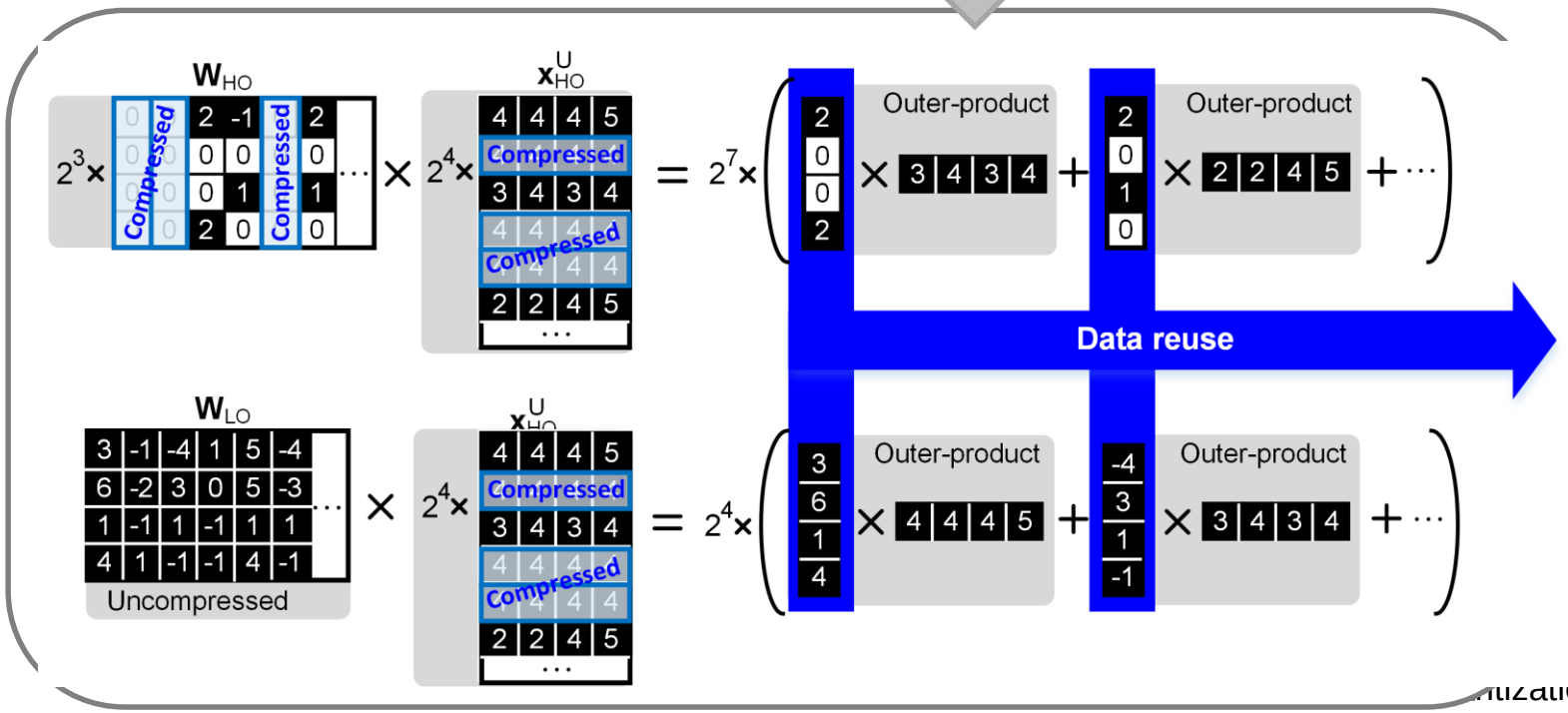
Proposed work

❶ Asymmetrically Quantized bit-Slice GEMM (AQS-GEMM)

AQS-GEMM

$$\mathbf{W}\mathbf{x}+\mathbf{b} \approx s_W s_x\left(\mathbf{W}_{\text{int}}\mathbf{x}_{\text{uint}}+\hat{\mathbf{b}}_{\text{int}}\right) = s_W s_x((\mathbf{W}_{\text{HO}}\mathbf{x}_{\text{HO}}+\mathbf{W}_{\text{LO}}\mathbf{x}_{\text{HO}}+\mathbf{W}_{\text{HO}}\mathbf{x}_{\text{LO}}+\mathbf{W}_{\text{LO}}\mathbf{x}_{\text{LO}})+\hat{\mathbf{b}}_{\text{int}})$$

$$(\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\mathbf{x}_{\text{HO}}^{\text{Uncompressed}} - r(\mathbf{W}_{\text{HO}}+\mathbf{W}_{\text{LO}})\mathbf{J}^{\text{Uncompressed}} + \mathbf{b}'$$



Efficient compensation term

$$2^4\times\left(2^3\times\left(\begin{bmatrix}2\\0\\0\\2\end{bmatrix}+\begin{bmatrix}2\\0\\1\\0\end{bmatrix}+\cdots\right)+\begin{bmatrix}3\\6\\1\\4\end{bmatrix}+\begin{bmatrix}-4\\3\\1\\-1\end{bmatrix}+\cdots\right)$$

1 x 4 vector
(r = 4)

$$\times\begin{bmatrix}4&4&4&4\end{bmatrix}$$

Proposed work

- **❶ Asymmetrically Quantized bit-Slice GEMM (AQS-GEMM)**
  - $\boldsymbol{W}_{int7} \in \mathrm{Z}^{4\times K}, \boldsymbol{x}_{uint8} \in \mathrm{Z}^{K\times 4}$
  - ρ_w, ρ_x: HO vector-level sparsities of weights and activations

TABLE I
HARDWARE WORKLOADS IN BIT-SLICE GEMM ACCELERATORS

Accel.	*Sibia* [53]	*Panacea* (AQS-GEMM core)		
Core's comput.	Bit-slice GEMMs	Bit-slice GEMMs w/o compensation	Compensation	
			In (5)	In (6)
Mul.	$32K(2-\max(\rho_x,\rho_w))$	$16K(2-\rho_x)(2-\rho_w)$	16	
Add.	$32K(2-\max(\rho_x,\rho_w))$	$16K(2-\rho_x)(2-\rho_w)$	$8K\rho_x$	$8K(1-\rho_x)$
EMA	$14K$	$4K(4-\rho_w-\rho_x)$	$8K\rho_x$	0

Straightforward version

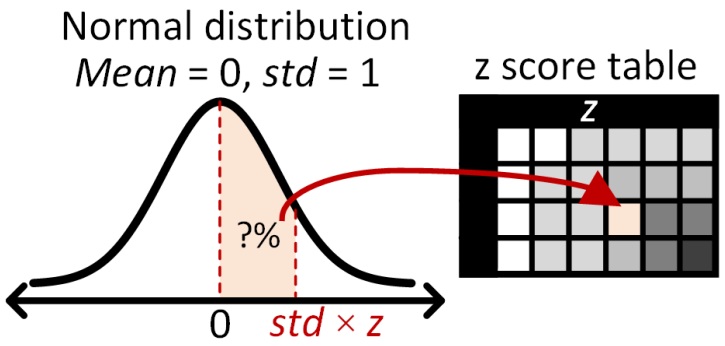
Proposed compensation

Panacea: Novel DNN accelerator using accuracy-preserving asymmetric quantization and energy-saving bit-slice sparsity

Proposed work

❸ Enhancing slice-sparsity : Distribution-based slicing (DBS)

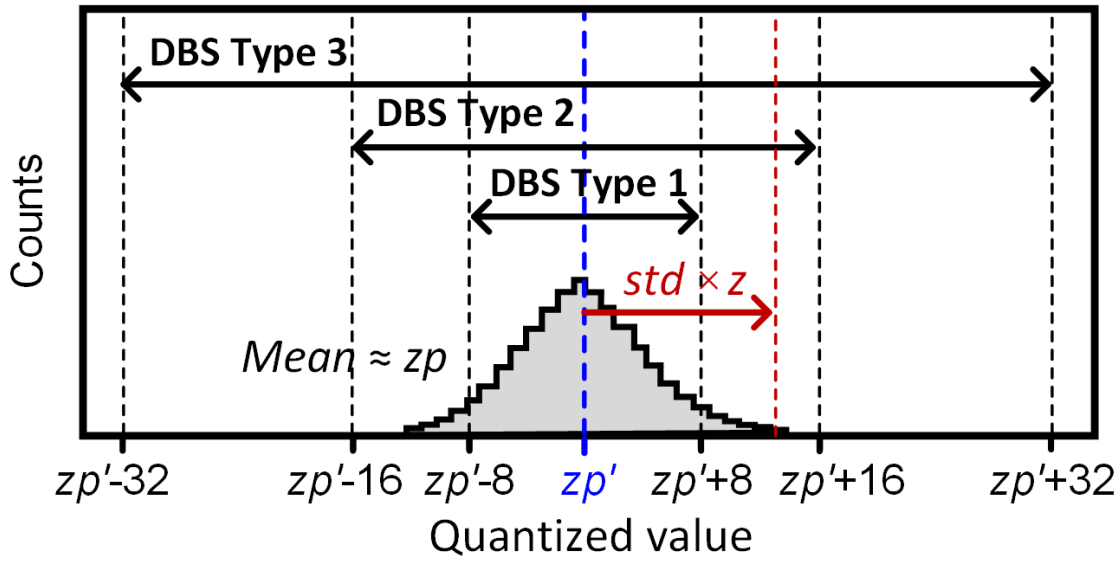
- According to *std* (observed by calibration), we define three DBS types.



DBS Type-1 : $0 \leq std \times z < 8$ (narrow distribution)

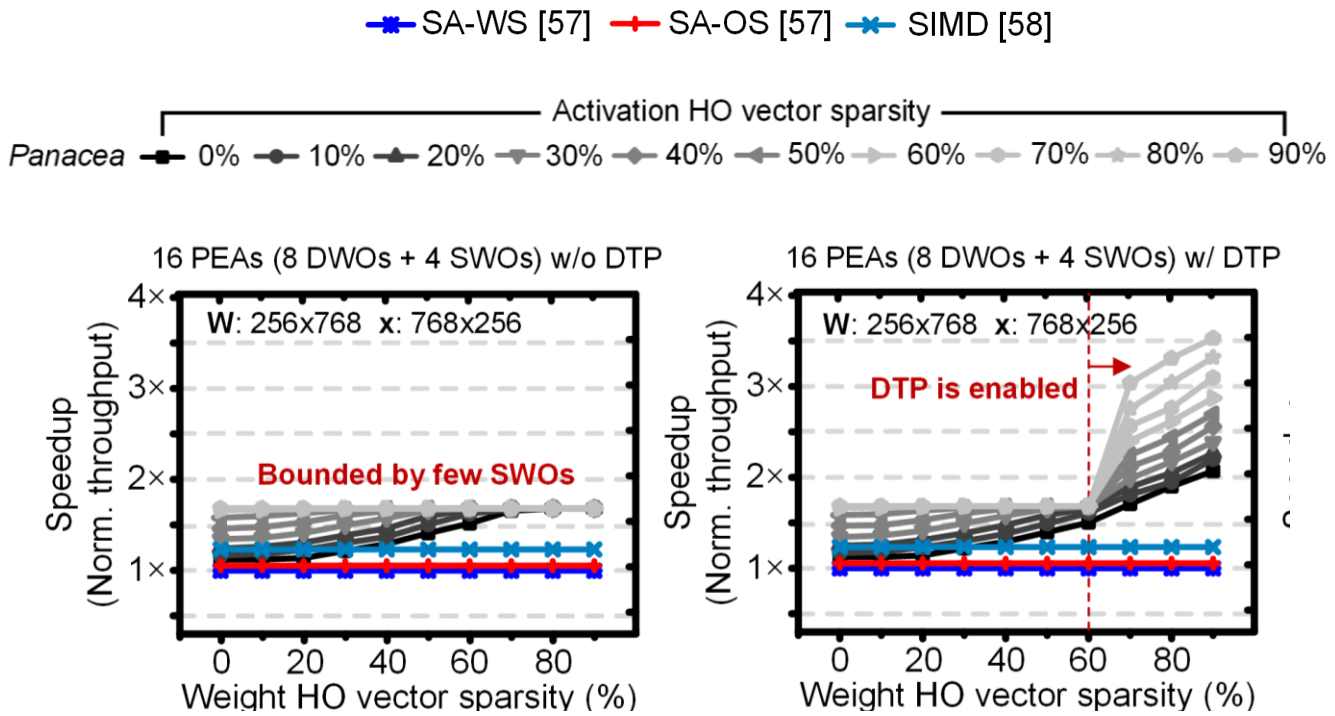
DBS type-2 : $8 \leq std \times z < 16$ (widely distribution)

DBS type-3 : $16 \leq std \times z$ (super-widely distribution)



Proposed work

- **❶ Throughput evaluation on slice-level sparsity & design options**
  - 16 PEAs, each of which 8 DWOs and 4 SWOs.
  - At low sparsity, the throughput performance is bounded by few SWOs.



Experimental Results

- **Decoupling the advantages of proposed methods**
  - These are relative energy-efficiency and perplexity for inference of OPT-2.7B.
    - (a) using different quantization methods in ***Panacea***
    - (b) using the same asymmetric quantization in two versions of ***Panacea***