THE CHIPS TO SYSTEMS CONFERENCE
SPONSORED BY: CEDA IEEE Council on Electronic Design Automation, SIGDA ACM

HAP: Efficient Quantization Harnessing Adaptive Precision for DNN Hardware Acceleration

Erjing Luo*, Xinkuang Geng†, Honglan Jiang†, Leibo Liu‡, Jie Han*

*Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada

†Department of Micro-Nano Electronics, Shanghai Jiao Tong University, Shanghai, China

‡School of Integrated Circuits, Tsinghua University, Beijing, China

Email: {erjing1, jhan8}@ualberta.ca

Content

- Introduction
- Quantization Solution
- Accelerator Design
- Experiments
- Conclusions

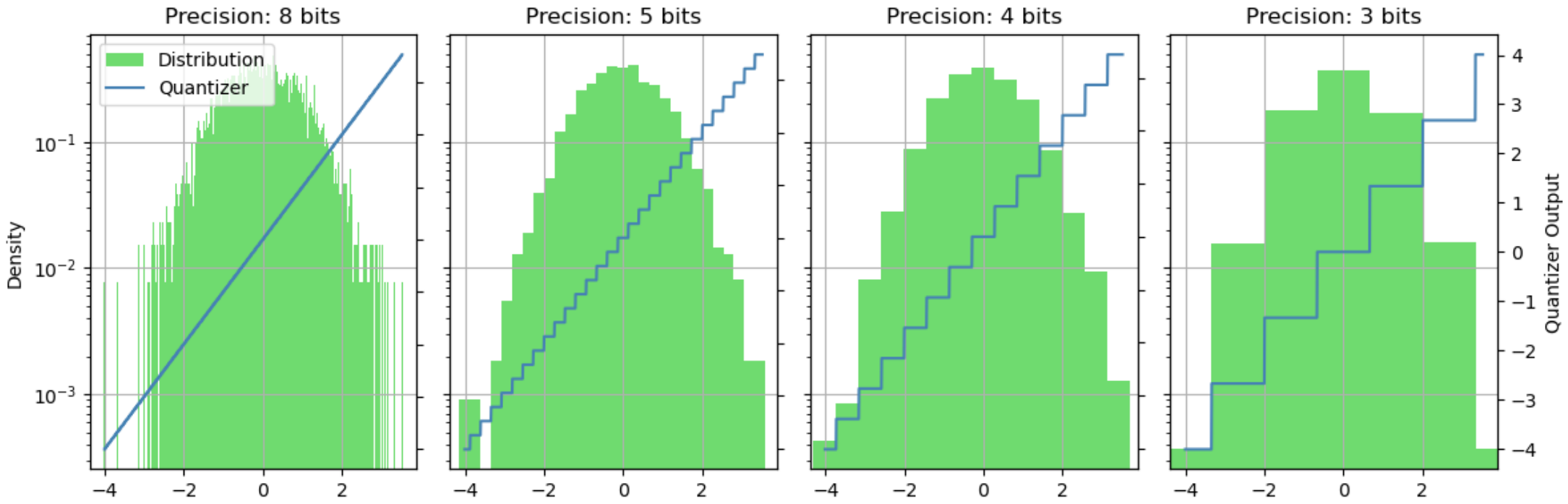
SPONSORED BY: CEDA IEEE Council on Electronic Design Automation, sigda acm

Section 1 – Introduction

- Background
- Motivations
- Challenges
- Contributions

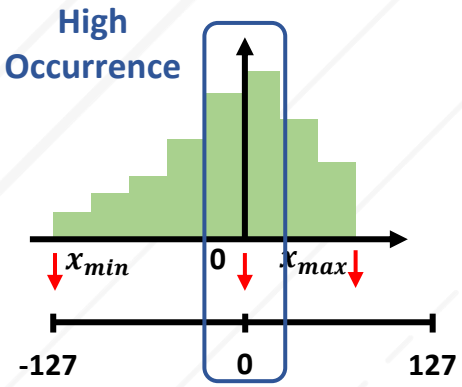
Fundamental Limitation of PTQ

- **Post-training quantization (PTQ)** has been widely used to accelerate Deep Neural Networks (DNNs) [1-6].
- However, PTQ faces fundamental limits, as the **quantization resolution** decreases exponentially with precision.

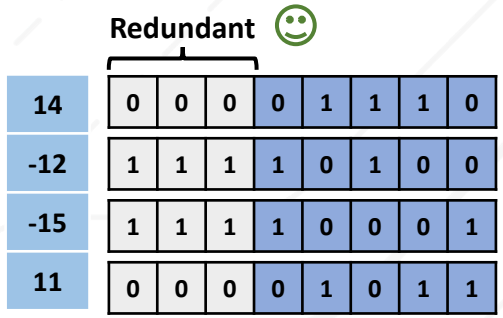


Adaptive-Precision Quantization

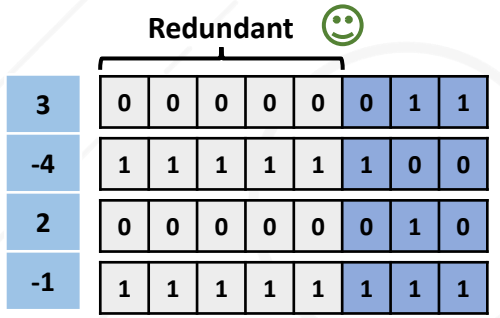
- **Key idea:** encoding bit-width < quantization precision.
- Values with smaller magnitudes require less encoding bit-width.
- **Adaptive-precision quantization (APQ)** adjusts bit-width for localized data without sacrificing resolution [8-10].



Mapping Data to Quantization Domain



Example Data Group 1

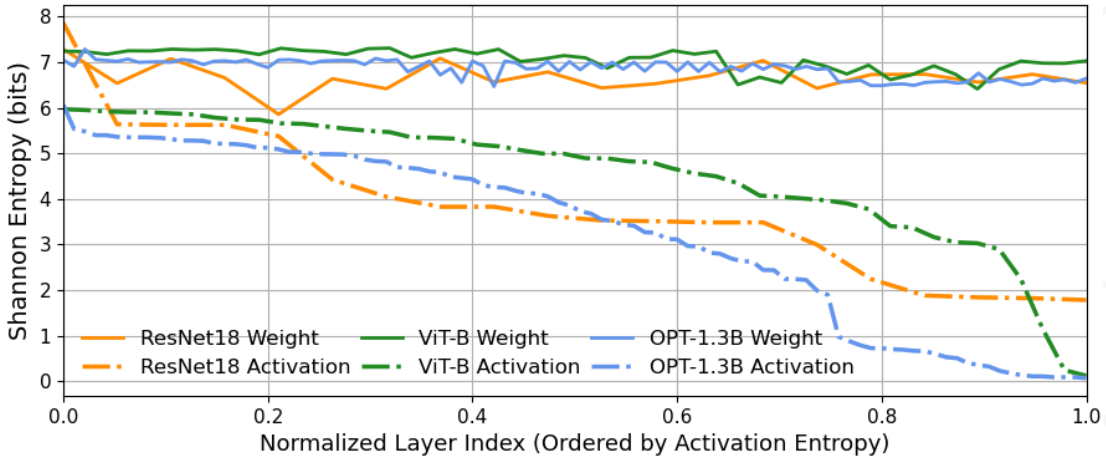


Example Data Group 2

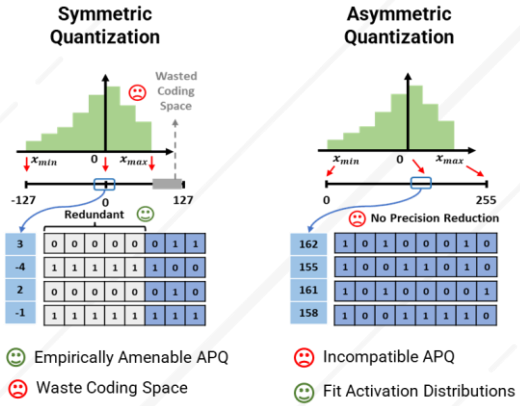
Motivations and Challenges

APQ is Better Suited for Activations than for Weights

- PTQ for activations is less robust [4-6].
- Activations exhibit more exploitable redundancy in their binary representations.

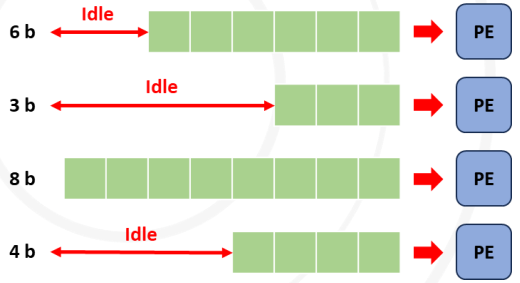


Challenges in Applying APQ to Activations



APQ is typically not applicable for asymmetric quantization (AQ) that is preferred by activations [11,12].

APQ leads to imbalanced computational workloads in parallel architectures.



Contributions

HAP – A Quantization Algorithm and Accelerator Design to Harness Adaptive-Precision

a) Dynamic Asymmetric Re-Quantization

- General APQ for activations
- Flexible for workload balancing
- Lossless compression for 8-bit activations

b) Vulnerable Channel Protection

- Channel-level dual-precision (4/8-bit) weight quantization
- Accuracy – performance trade-off

c) Accelerator Design

- Fine-grained bit-serial arithmetic
- Outer-product reordering engine to match precision at run time

d) Experiments

- Diversified evaluations across CNN, ViT, and LLM models and benchmarks
- Superior accuracy and performance

Section 2 – Quantization Solution

- Dynamic Asymmetric Quantization (DAR)
- Dataflow for DAR
- Vulnerable Channel Protection (VCP)

Dynamic Asymmetric Re-Quantization

- **Problem:** data mapping is shifted by the zero-point Z in asymmetric quantization

$$q = clip(\left\lfloor \frac{x}{\Delta} \right\rceil + Z; 0, 2^B - 1)$$

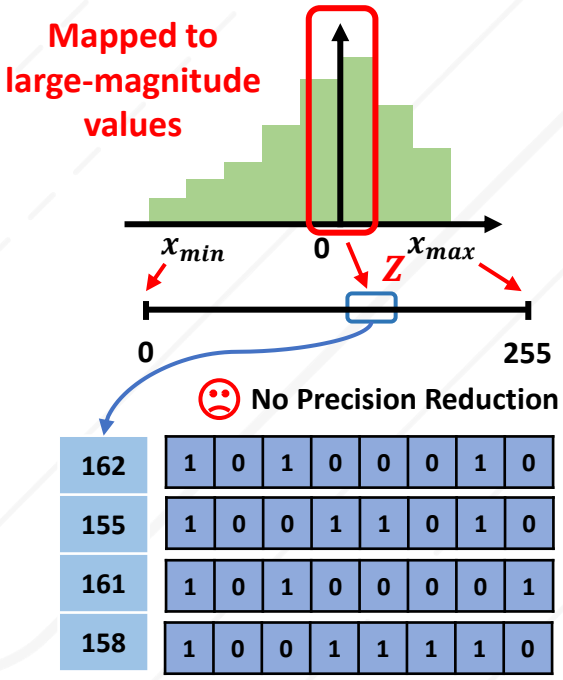
- **Solution:** subtract each data group with its minimum to restore precision adaptability

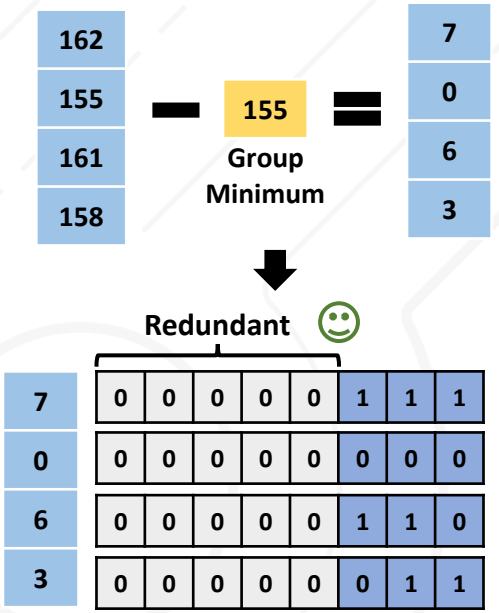
Group Range:

$$[q_{glb}, q_{gub}] \rightarrow [0, q_{gub} - q_{glb}]$$

Required Bit-width:

$$\lceil log_2(q_{gub} + 1) \rceil \rightarrow \lceil log_2(q_{gub} - q_{glb} + 1) \rceil$$

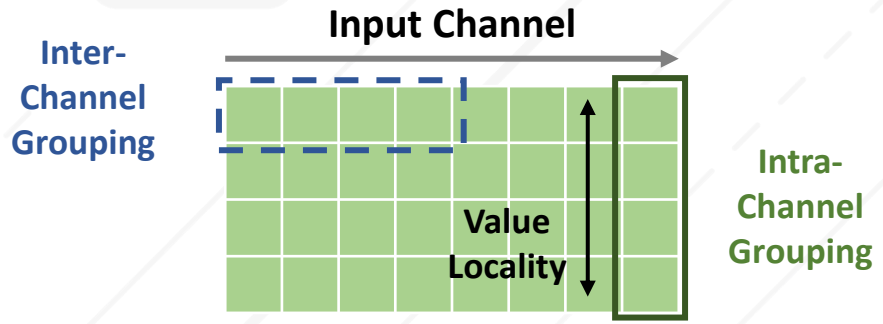




Dynamic Asymmetric Re-Quantization

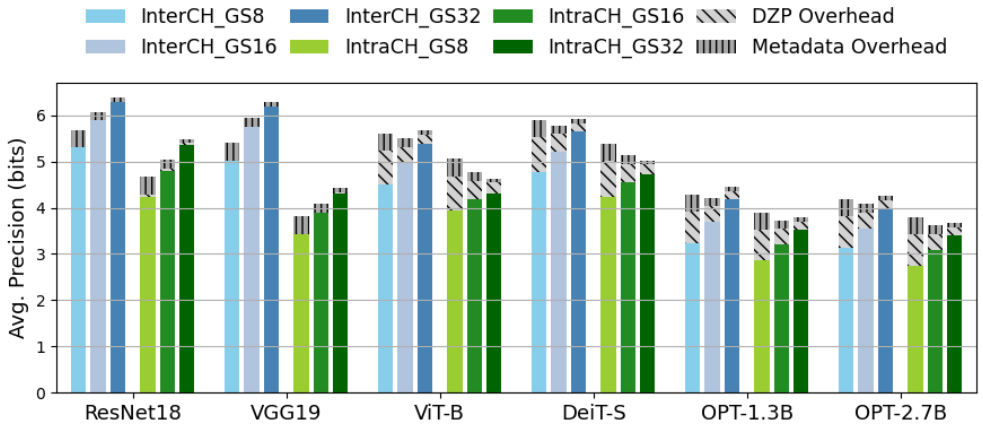
- **Dynamic Asymmetric Re-quantization (DAR)** : additionally introduces a per-group dynamic zero-point $\tilde{Z} = q_{glb}$

$$\tilde{q} = q - \tilde{Z} = clip(\left\lfloor \frac{x}{\Delta} \right\rceil + Z - \tilde{Z}; 0, 2^{\tilde{B}} - 1)$$



- **Intra-Channel Grouping**. Data within an input channel witness value locality [1, 2, 13] which results in lower average bit-width.

- Avg. bit-width: **4.39** bits (GS=16)



GEMM Decomposition Based on DAR

- DAR reformulates the general matrix-multiplication (GEMM) in DNNs into three terms.

$$y_{i,j} \approx \Delta^a \Delta^w \sum_{k=0}^{K-1} (\tilde{q}_{i,k}^a + \tilde{Z}_k^a - Z^a) q_{k,j}^w$$

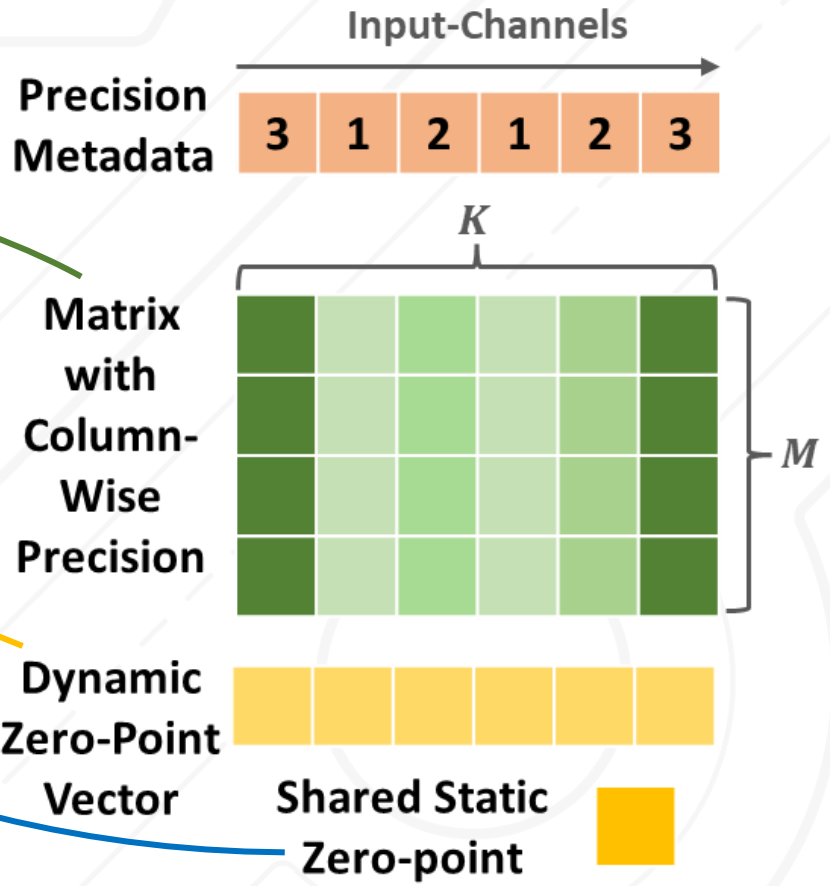
$$= \Delta^a \Delta^w (P_A + P_D + P_S)$$

where

$$P_A = \sum_{k=0}^{K-1} \tilde{q}_{i,k}^a q_{k,j}^w \,, P_D = \sum_{k=0}^{K-1} \tilde{Z}_k^a q_{k,j}^w \,, P_S = -\sum_{k=0}^{K-1} Z^a q_{k,j}^w$$

Processed Online

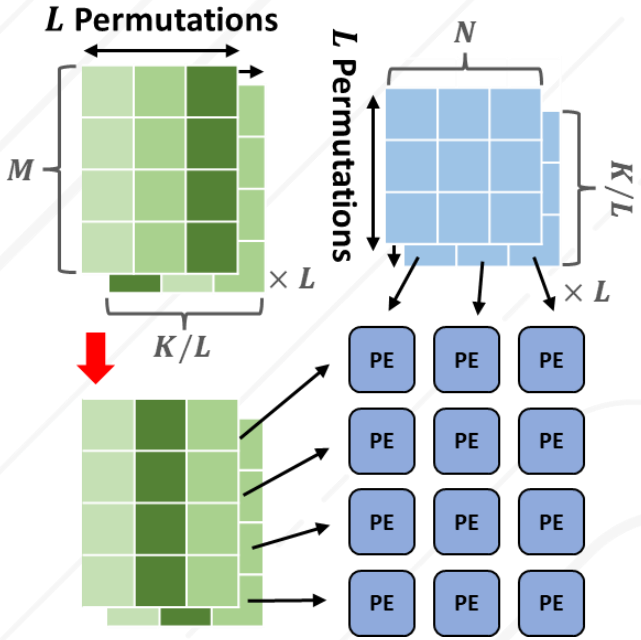
Processed Offline



Dataflow for DAR

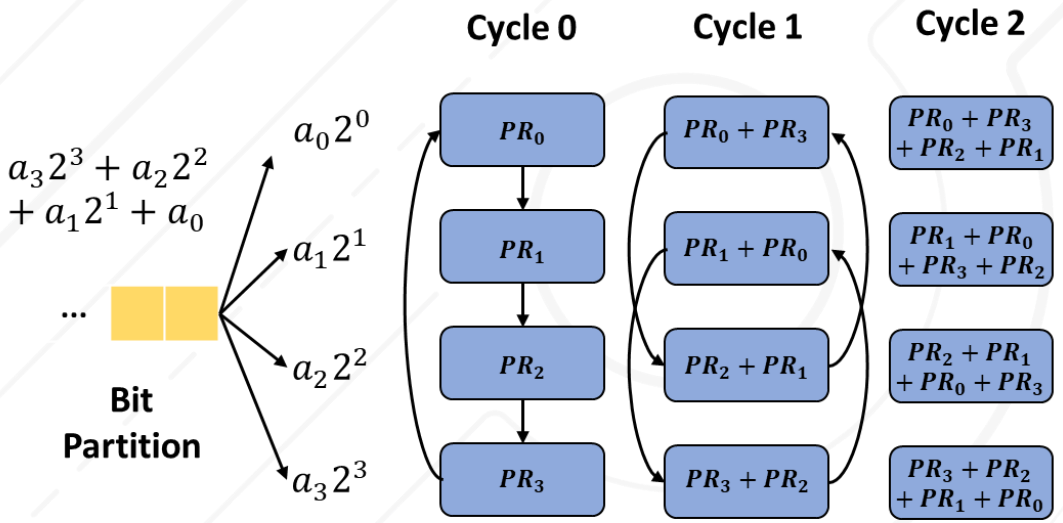
Outer-Product Reordering

- Bit-serial output-stationary dataflow
- Permute outer-product to match precision



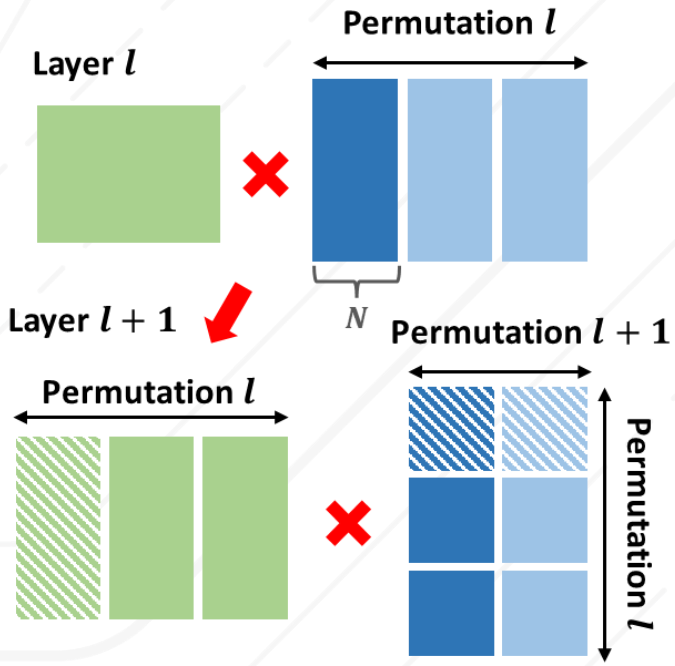
Architecture Reuse for DZP-Term

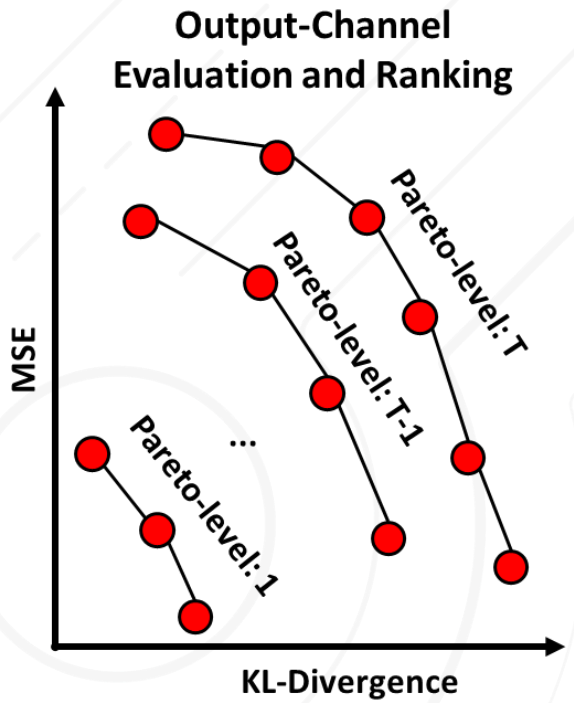
- Partition each bit significance to expand the DZP vector as a bit matrix
- Aggregate partial results with logarithmic complexity



Vulnerable Channel Protection

- **Vulnerable channel protection (VCP):** target 4-bit PTQ while upgrading vulnerable output-channels (VCs) to 8-bit PTQ
- Cluster VCs and apply orthogonal permutation to ensure correct computations.
- Empirically identify VCs with a Pareto-based multi-metric evaluation





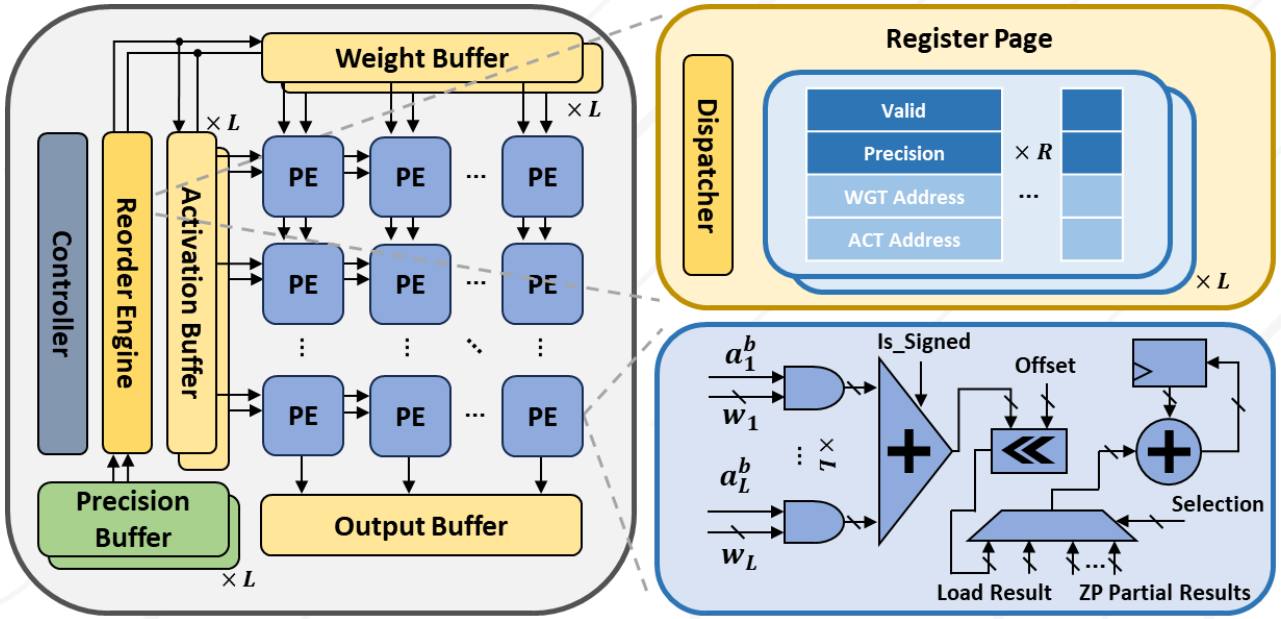
SPONSORED BY: CEDA IEEE Council on Electronic Design Automation sigda acm

Section 3 – Accelerator Design

- Overview
- Reorder Engine

Architecture Overview

- 16×32 processing engines (PEs).
- Multiplications are serialized as 4-bit additions.
- PE concurrency: 16 multiplications.
- Cache-like reorder engine to balance computational workloads.
- Pipeline: Precision Buffer -> Reorder Engine -> Activation/Weight Buffer -> Array -> Output Buffer

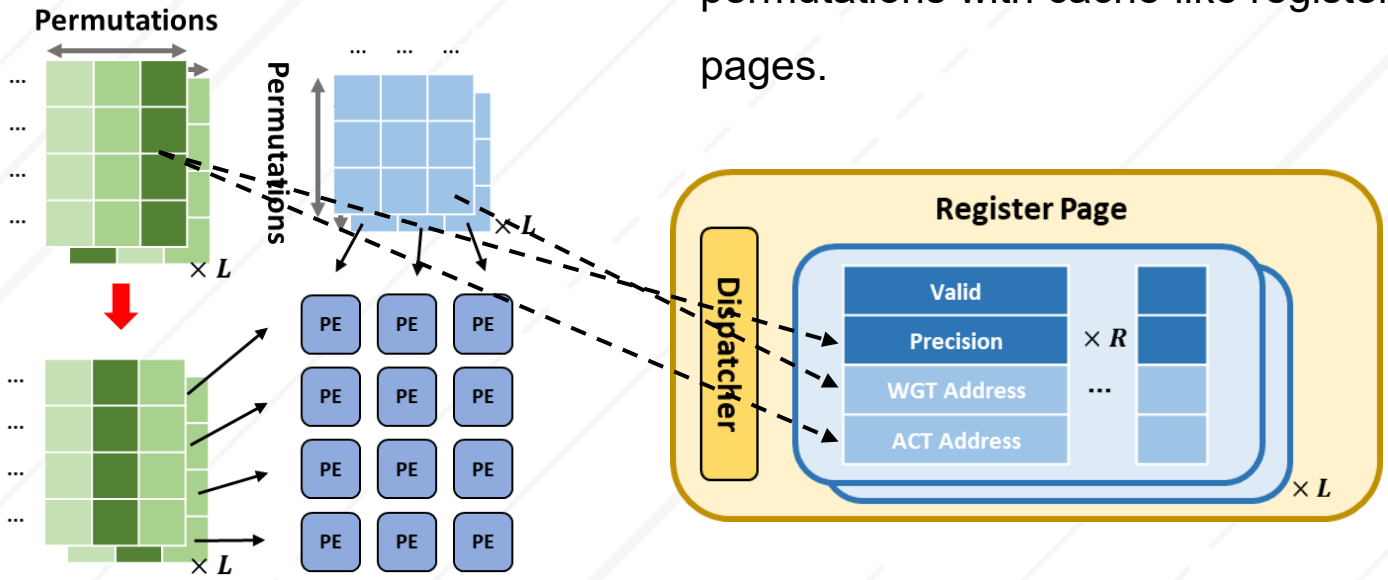


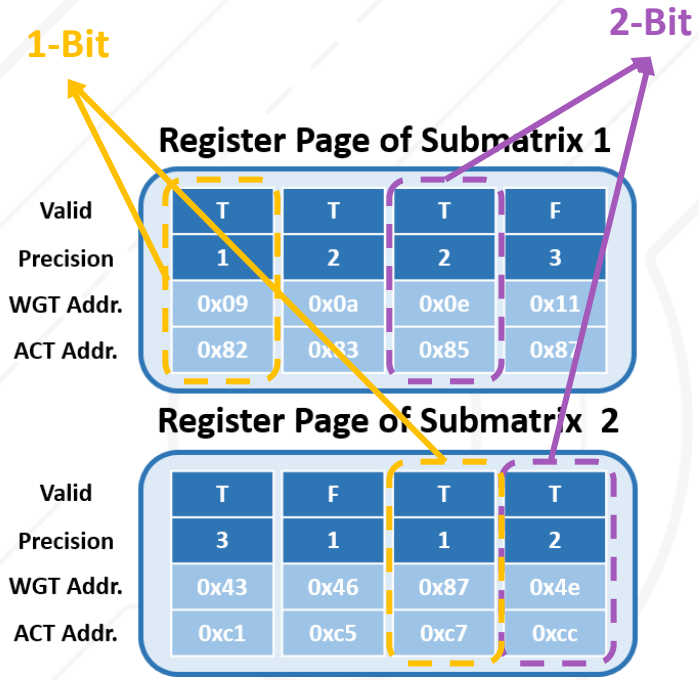
Reorder Engine

Goal: matching the precisions of the L activation groups through reordering the outer-products in GEMM

Key Idea: assuming an activation or a weight group are stored under the same address in buffers, the outer-products are reordered via address permutations with cache-like register pages.

An example of reordering with $L = 2$ and $R = 4$. Both 1-bit and 2-bit are matched across different register pages.



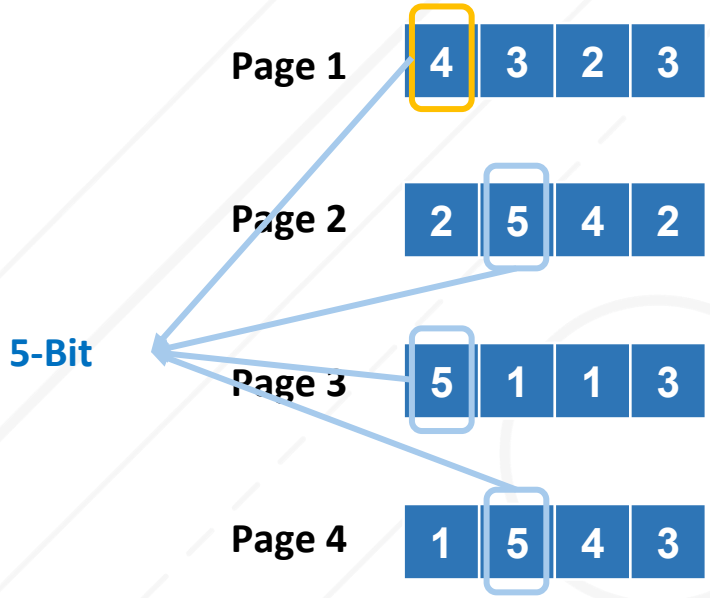


Reorder Engine

Problem: exact matching becomes infeasible with more register pages but limited page sizes.

Cannot Detect a Matched Precision across Pages ☹

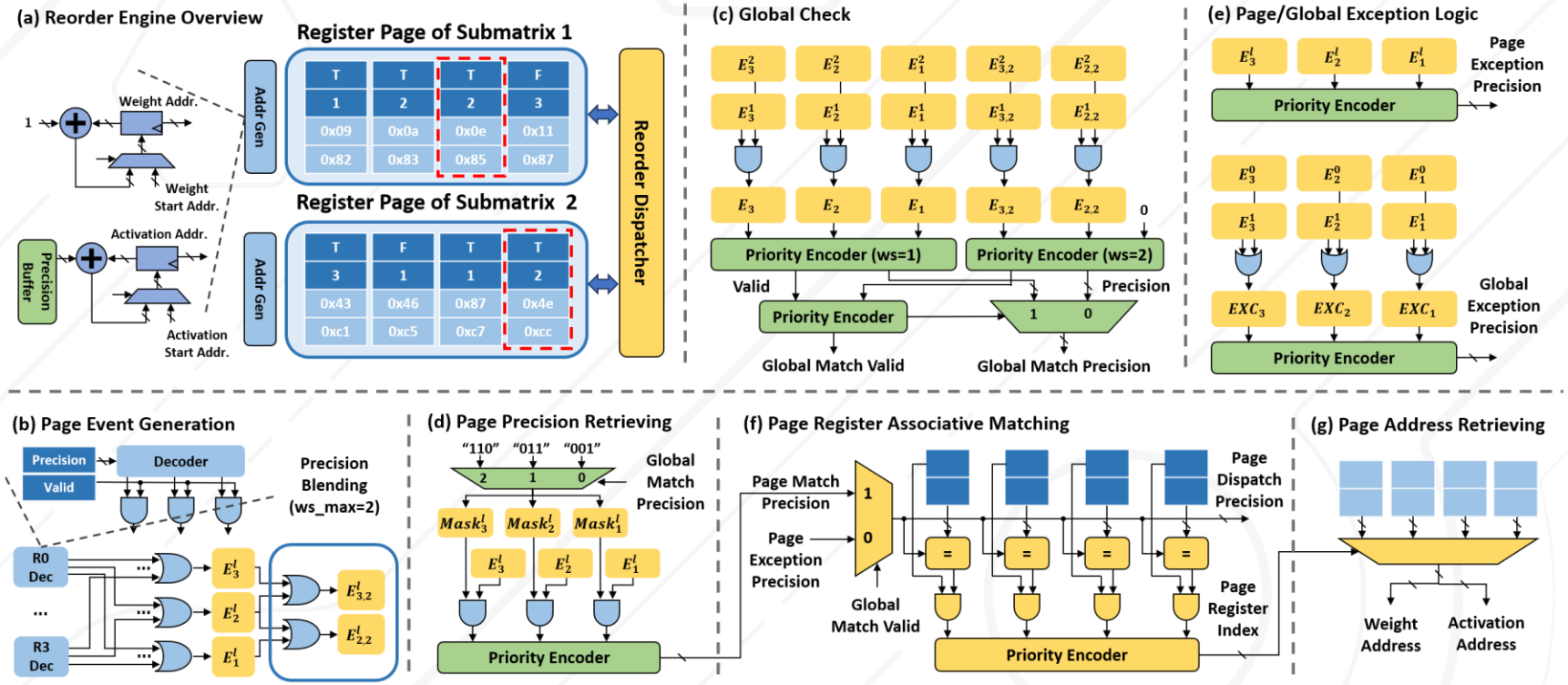
Intuition: match rate can be effectively improved with relaxation on the match conditions



Relaxation only Introduces Trivial Computation Idling ☺

Reorder Engine Datapath

- **Relaxation:** given a window size ws, precision b' is considered matched on b if it satisfies $0 \leq b - b' \leq ws$
- **Multiple levels of relaxation:** implementing multiple window sizes, $1, \cdots, ws_{max}$; prioritizing lower window sizes.
- **Exception** is triggered if there is still no matched case



SPONSORED BY: CEDA IEEE Council on Electronic Design Automation, sigda acm

Section 4 – Experiments

- Setup
- Accuracy
- Match Rate & Synthesis
- End-End Simulation

Experimental Setup

- **DNNs and Datasets:** CNNs (ResNet18, VGG19) and Vision Transformers (ViT-B, Deit-S) on ImageNet-1K, Language Models (OPT-1.3B and OPT-2.7B) on WikiText-2
- **Quantization:** SQuant [3], NoisyQuant [4], SmoothQuant [5], and OmniQuant [14]
- **Accelerators:** Stripes [7], ShapeShifter [8], and BitWave [9]
- **End-End Simulation:** cycle-accurate simulator + CACTI [15] energy simulator + ISO-compute comparison (512 8-bit multipliers)
- **Synthesis:** 65nm TSMC PDK + 500 MHz

Accuracy

- Top-1 Accuracy for CNNs and ViTs; perplexity for LLMs. Empirically set VCP precision as 4.6 and 5.6 bits
- The PTQ for activations are less robust than the PTQ for weights
- The lossless adaptive-precision quantization preserves most accuracy
- VCP enhances pure 4-bit weight quantization.

Table 1: Top-1 Accuracy (%) & Perplexity (for LLMs)

Prec. (W/A)	32/32	8/8	4/8	8/4	5/5	4.6/4~	5.6/4~
CNNs	FP		SQuant			HAP+SQuant	
ResNet18 ↑	69.76	69.65	68.49	63.41	68.09	68.84	**69.21**
VGG19 ↑	74.23	74.16	73.61	69.22	72.95	73.91	**74.09**
ViTs	FP		NoisyQuant			HAP	
ViT-B ↑	84.54	84.31	81.62	72.42	79.04	82.98	**83.47**
Deit-S ↑	79.83	79.24	76.50	48.32	61.05	78.26	**78.84**
LLMs	FP		SmoothQuant			HAP	
OPT-1.3B ↓	14.62	16.77	29.15	675.80	34.07	23.77	19.13
OPT-2.7B ↓	12.47	13.90	34.17	380.78	35.05	22.66	16.13
LLMs	FP		OmniQuant-LWC			HAP+LWC	
OPT-1.3B ↓	14.62	15.36	17.15	321.43	92.70	16.01	**15.81**
OPT-2.7B ↓	12.47	13.44	14.42	1573.49	50.90	14.29	**13.96**

Worst

Improved Accuracy

Match Rate & Synthesis Results

- Naïve PE utilization is 66.8%
- Larger R and ws_{max} improves PE utilization
- Parameter selection: $ws_{max} = 3; R = 8$

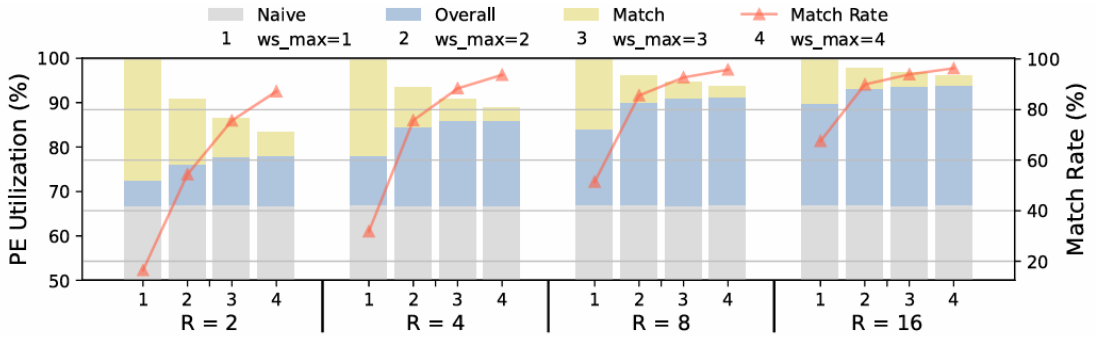


Figure 7: Match rate and PE utilization based on ViT-B.

- Relaxation is more area- and energy efficient than naively increasing register size
- The total area and energy overhead of Reorder Engine are 4.83% and 6.05%

Table 2: Core Area & Power Breakdown

Module	Area (mm^2)		Power (mW)	
PE Array	1.0034	93.72%	254.7820	92.15%
RE Datapath Base	0.0080	0.75%	0.3180	0.12%
RE Precision Blending	0.0009	0.08%	0.1710	0.06%
Register Pages	0.0428	4.00%	16.2397	5.87%
Others	0.0156	1.45%	4.9880	1.80%
Total	1.0707	100.00%	276.4987	100.00%

End-End Speedup & Energy Breakdown

- Performance gain from DAR and RE: 1.63x (HAP_W8.0)
- Further improvement with VCP: 2.25x (HAP_W5.6) and 2.65x (HAP_W4.6)

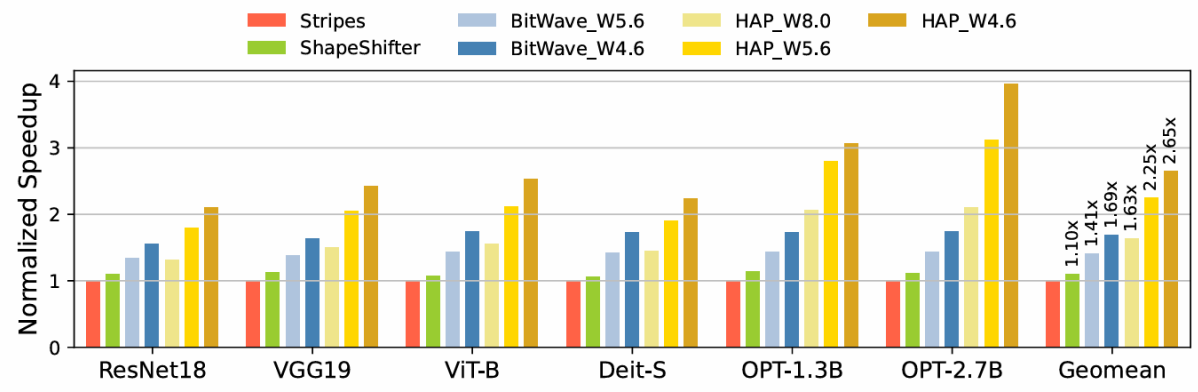


Figure 8: Normalized speedup.

- Energy reduction: 32% (HAP_W8.0), 48% (HAP_W5.6), and 55% (HAP_W4.6)

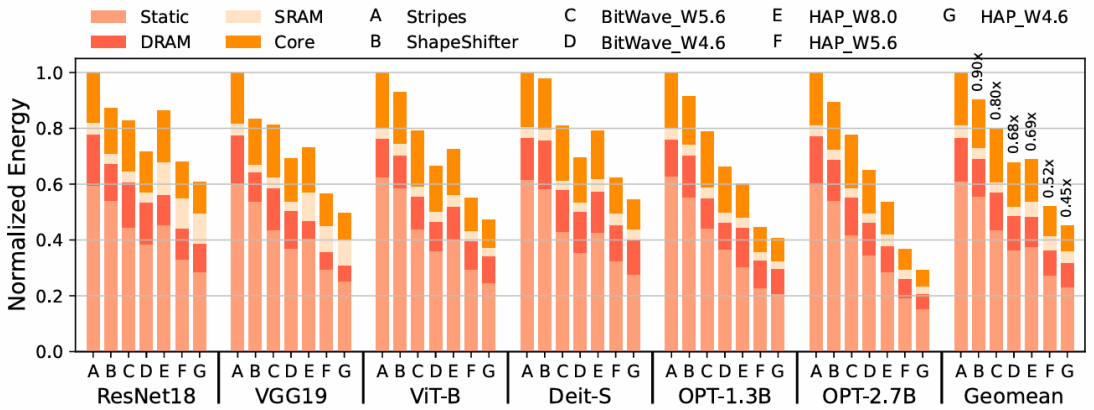


Figure 9: Normalized energy breakdown.

SPONSORED BY: CEDA IEEE Council on Electronic Design Automation, sigda acm

Section 5 – Conclusion

- Summary
- Reference

Contributions

HAP – A Quantization Algorithm and Accelerator Design to Harness Adaptive-Precision

a) Dynamic Asymmetric Re-Quantization

- General APQ for activations
- Flexible for workload balancing
- Lossless compression for 8-bit activations

b) Vulnerable Channel Protection

- Channel-level dual-precision (4/8-bit) weight quantization
- Accuracy – performance trade-off

c) Accelerator Design

- Fine-grained bit-serial arithmetic
- Outer-product reordering engine to match precision at run time

d) Experiments

- Diversified evaluations across CNN, ViT, and LLM models and benchmarks
- Superior accuracy and performance

Reference

[1] Yvinec, Edouard, et al. "Spiq: Data-free per-channel static input quantization." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2023.

[2] Moon, Jaehyeon, et al. "Instance-aware group quantization for vision transformers. " Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[3] Guo, Cong, et al. "SQuant: On-the-Fly Data-Free Quantization via Diagonal Hessian Approximation. " Proceedings of International Conference on Learning Representations. 2022

[4] Liu, Yijiang, et al. "Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[5] Xiao, Guangxuan, et al. "Smoothquant: Accurate and efficient post-training quantization for large language models." International conference on machine learning. PMLR, 2023.

[6] Lin, Ji, et al. "Awq: Activation-aware weight quantization for on-device llm compression and acceleration." Proceedings of machine learning and systems 6 (2024): 87-100.

[7] Judd, Patrick, et al. "Stripes: Bit-serial deep neural network computing." 2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2016.

[8] Lascorz, Alberto Delmás, et al. "Shapeshifter: Enabling fine-grain data width adaptation in deep learning." Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture. 2019.

[9] Shi, Man, et al. "Bitwave: Exploiting column-based bit-level sparsity for deep learning acceleration." 2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, 2024.

[10] Chen, Yuzong, et al. "BBS: Bi-directional bit-level sparsity for deep learning acceleration." 2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2024.

[11] Geng, Xinkuang, et al. "QUQ: quadruplet uniform quantization for efficient vision transformer inference." Proceedings of the 61st ACM/IEEE Design Automation Conference. 2024.

[12] Kam, Dongyun, et al. "Panacea: Novel DNN Accelerator using Accuracy-Preserving Asymmetric Quantization and Energy-Saving Bit-Slice Sparsity." 2025 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2025.

[13] Xue, Chenhao, et al. "Oltron: Algorithm-Hardware Co-design for Outlier-Aware Quantization of LLMs with Inter-/Intra-Layer Adaptation." Proceedings of the 61st ACM/IEEE Design Automation Conference. 2024.

[14] Shao, Wenqi, et al. "Omniquant: Omnidirectionally calibrated quantization for large language models." arXiv preprint arXiv:2308.13137 (2023).

[15] Balasubramonian, Rajeev, et al. "CACTI 7: New tools for interconnect exploration in innovative off-chip memories." ACM Transactions on Architecture and Code Optimization (TACO) 14.2 (2017): 1-25.

THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY:

CEDA IEEE Council on Electronic Design Automation

SIGDA ACM

Thank You!

Acknowledgement

The work at the University of Alberta was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada (Project Numbers: RES0048688, RES0051374 and RES0054326) and Alberta Innovates (Project Number: RES0053965).