

# HAP: Accelerating DNNs with Resolution-Preserved Quantization by Harnessing Adaptive-Precision

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

{erjing1, jhan8}@ualberta.ca, {xinkuang, honglan}@sjtu.edu.cn, liulb@tsinghua.edu.cn

**Abstract**—Reducing the precision in post-training quantization can cause catastrophic accuracy loss in Deep Neural Networks, especially when compressing the activations. To address this problem, we present a novel adaptive-precision quantization (APQ) and accelerator design that achieves lossless activation compression by exploiting the inherent coding redundancy. Compared to existing APQ methods, this design can be generalized to implement asymmetric quantization, making it particularly suitable for activations. The accelerator offers a practical solution to mitigate the computational workload imbalance problem incurred by variable precision. A dual-precision quantization scheme further provides the flexibility to trade off accuracy and performance.

## I. INTRODUCTION

Post-training quantization (PTQ) avoids expensive training processes and is effective in accelerating Deep Neural Networks (DNNs) [1]–[6]. Since quantizer capacity decreases exponentially as precision reduces, adaptive-precision quantization (APQ) leverages the coding redundancy in binary representations to reduce bit-width without sacrificing resolution [7]–[10]. From an information theory perspective, we argue that APQ is better suited for activations than for weights, as activations’ pronounced long-tailed distributions lead to greater exploitable redundancy in quantized values, as evidenced by the Shannon entropy in Fig. 1. In addition, this factor makes activations less robust to aggressive PTQ [11]–[13], further motivating APQ as an alternative. However, since APQ only reduces bit-width for near-zero values, it is more effective for symmetric quantization (SQ) than for the asymmetric variant (AQ), as illustrated in Fig. 2(a)-(b). Consequently, it fails to account for the prevalent asymmetry in activations and the coding space is unnecessarily wasted [14]–[16]. Moreover, the APQ of activations incurs varying and imbalanced computational workloads, making it challenging to harvest the theoretical performance gain.

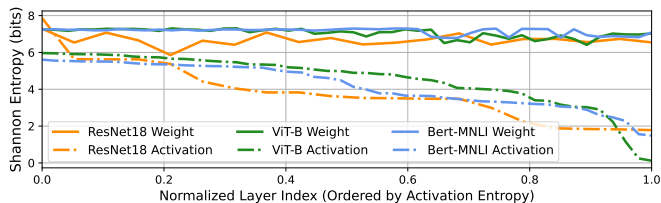


Fig. 1. Layer-wise Shannon Entropy of 8-bit ResNet18, ViT-Base, and Bert.

To bridge this gap, we present a novel PTQ and accelerator design, aiming to **H**arness **A**daptive-**P**recision (HAP) for DNN inference. HAP adopts 8-bit AQ for activations and employs a novel APQ method, dynamic asymmetric re-quantization (DAR), for compression. DAR circumvents the limitations

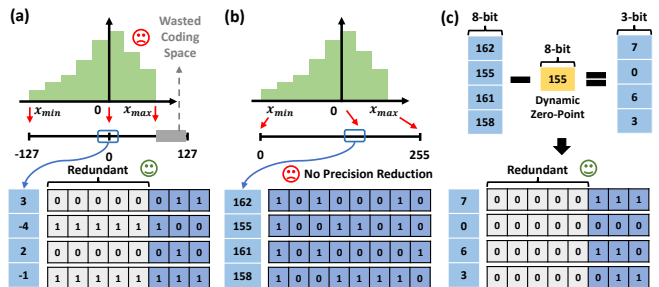


Fig. 2. Applying APQ to a data group sampled from an asymmetric distribution. In the scenarios of: (a) SQ; (b) AQ; (c) AQ + DAR.

of existing APQs with a group-wise dynamic zero-point and adopts an intra-channel grouping strategy to further exploit value locality. Based on DAR, we design an efficient bit-serial accelerator that balances workloads by reordering activation groups. For weights, HAP provides a dual-precision vulnerable channel protection scheme (VCP) to support 4- and 8-bit PTQ while preserving the flexibility to trade off accuracy and performance. Experiments demonstrate superior accuracies on typical DNNs and a  $2.55\times$  speedup over existing accelerators.

## II. METHODOLOGY

### A. DAR: Dynamic Asymmetric Re-quantization

AQ maps a variable  $x$  from an interval  $[x_{lb}, x_{ub}]$  into  $[0, 2^B - 1]$  and rounds  $(\lfloor \cdot \rfloor)$  it to a  $B$ -bit integer  $q$ ,

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

where  $\Delta = \frac{x_{ub} - x_{lb}}{2^B - 1}$  is the scaling factor, the zero-point  $Z = -\lfloor \frac{x_{lb}}{\Delta} \rfloor$  is the output when  $x \approx 0$ , and  $\text{clip}(q; \alpha, \beta)$  further limits  $q$  to the nearest boundary if it exceeds  $[\alpha, \beta]$ . Due to the presence of  $Z$ , near-zero values are mapped to integers with large magnitudes, thus reducing the effectiveness of APQ. To address this, we subtract the minimum from each quantized data group  $[q_{glb}, q_{gub}]$  (i.e.  $\tilde{q} = q - q_{glb}$ ) to shift it to  $[0, q_{gub} - q_{glb}]$  with smaller magnitudes, such that the bit-width of  $\tilde{q}$  can be reduced to  $\tilde{B} = \lceil \log_2(q_{gub} - q_{glb} + 1) \rceil$ . Essentially, this method incurs an extra per-group dynamic zero-point (DZP)  $\tilde{Z} = q_{glb}$ ,

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

In practice, as some layers naturally satisfy  $Z \approx 0$ , we enable DZP based on profiling. Moreover, as there exists value locality within activation channels [1]–[3], we adopt an intra-channel grouping strategy (Fig. 3(a)). Fig. 4 profiles the average precision of DAR on DNNs. The group size  $GS$  is set to 16, which leads to an average precision of 4.5 bits.

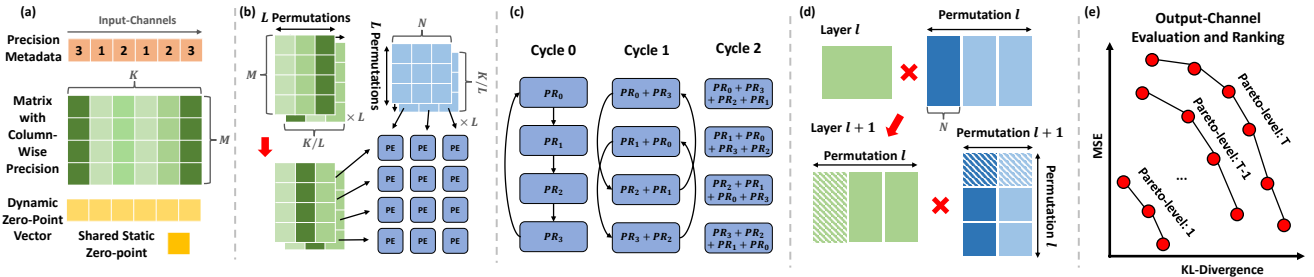


Fig. 3. Illustrative examples with  $M = GS = 4$ ,  $N = 3$ ,  $K = 6$ ,  $L = 2$ , and  $B^a = 4$ . (a) Intra-channel DAR; (b) Outer-product Reordering Dataflow; (c) partial result aggregation; (d) compile-time VC clustering and orthogonal permutation; (e) Pareto-based weight channel evaluation and ranking.

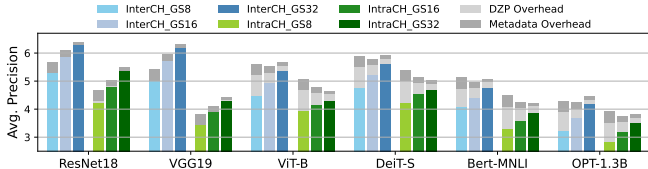


Fig. 4. Average precision for DAR, under different settings of grouping strategies (intra-channel or inter-channel) and group sizes.

### B. GEMM Accelerator Based on DAR

Based on DAR, we design an accelerator to process the General Matrix Multiplications (GEMMs) in DNNs (Fig. 3(a)-(c)). Particularly, we consider a GEMM tile with an activation matrix  $A \in \mathbb{R}^{M \times K}$  and a weight matrix  $W \in \mathbb{R}^{K \times N}$ , where  $M = GS$ . In this case,  $A$  can be decomposed into three components: a matrix with variable precisions in the columns, a DZP vector, and a static zero-point. Assuming SQ for weights, the quantized dot-product can be formulated as

$$y_{i,j} \approx \Delta^a \Delta^w \sum_{k=0}^{K-1} (\tilde{q}_{i,k}^a + \tilde{Z}_k^a - Z^a) q_{k,j}^w \quad (3)$$

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

where the superscripts  $a$  and  $w$  indicate activations and weights,

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

We focus on  $P_A$  and  $P_D$ , as  $P_S$  can be pre-computed.

Based on the output-stationary dataflow in [9], we propose outer-product reordering dataflow (ORD) to efficiently process  $P_A$ . Essentially, we fold both matrices into  $K/L$  iterations to concurrently process  $L$  pairs of outer-products on a  $M \times N$  bit-serial processing engine (PE) array, where we empirically set  $M = 16$ ,  $N = 32$ , and  $L = 16$ . Subsequently, ORD matches the precision of the  $L$  activation columns to reduce the waiting time for the heaviest workload, which is achieved by permuting the outer-product pairs within each folded sub-matrix (Fig. 3(b)). We implement ORD as a cache-like reorder engine that accesses data via permuted addresses. A major challenge is that the matching conditions become more stringent as  $L$  increases. Fortunately, we find that applying relaxation to the match conditions can effectively mitigate this problem.

To reuse the PE array for  $P_D$ , the DZP vector is decomposed bit-wise and expanded as a bit matrix. In our assumptions,  $M$  is a multiple of  $B^a$ . Hence, we map  $M/B^a$  DZP vectors onto the array in each execution. After computation, the additional aggregation paths between every  $B^a$  PEs in a column enable the summation of the partial results with logarithmic complexity.

TABLE I  
ACCURACY (%) & PERPLEXITY (FOR OPT-1.3B)

Prec. (W/A)	8/8	4/8	8/4	5/5	4.6/4~	5.6/4~
CNNs	SQuant [17]					
ResNet18 $\uparrow$	69.65	68.49	63.41	68.09	68.84	<b>69.21</b>
VGG19 $\uparrow$	74.16	73.61	69.22	72.95	73.91	<b>74.09</b>
ViTs	NoisyQuant [12]					
ViT-B $\uparrow$	84.31	81.62	72.42	79.04	82.98	<b>83.47</b>
DeiT-S $\uparrow$	79.24	76.50	48.32	61.05	78.26	<b>78.84</b>
NLP Models	SmoothQuant [11]					
Bert-Base $\uparrow$	84.62	83.82	73.92	75.31	84.65	<b>84.74</b>
OPT-1.3B $\downarrow$	16.77	230.58	1243.69	55.14	25.85	<b>18.84</b>

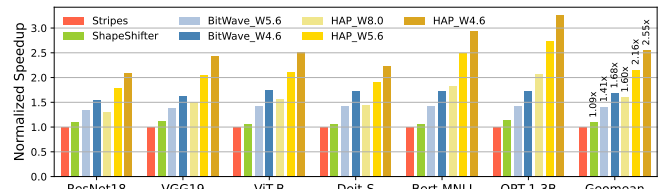


Fig. 5. Performance comparisons (normalized to Stripes [18]).

### C. VCP: Vulnerable Channel Protection

HAP targets 4-bit PTQ for weights, while enhancing accuracy by upgrading potentially vulnerable output-channels (VCs) to 8 bits. In hardware, an 8-bit operand is processed as two serial 4-bit ones. To keep weight precision consistent across the PEs, similar to [10], we limit the number of VCs to a multiple of the array column count  $N$ , and cluster them at compile-time. To ensure correctness in subsequent computations, this effect is counteracted by an orthogonal permutation of the weight matrix in the next layer. Such an operation is applicable in most DNN layers that follow one-path structures. A few exceptional cases, including Query-Key-Value projections and residual connections, are additionally handled by constraints and explicit permutations. Since there is no unified vulnerability metric, we employ a Pareto-based multi-metric evaluation (MSE and KL-Divergence) to empirically rank and identify VCs.

## III. EXPERIMENTS

**Accuracy.** Table I compares the accuracy with recent PTQ solutions [11], [12], [17] on a variety of DNN benchmarks: CNNs and ViTs on ImageNet-1K, Bert-Base on GLUE-MNLI, and OPT-1.3B on WikiText-2. HAP maintains activations in 8-bit precision and leverages APQ for compression, thus preserving most of the accuracy. VCP further enhances 4-bit quantization, leading to superior accuracy results at 4 or 5 bits.

**Performance.** Fig. 5 illustrates the performance comparisons with three bit-serial accelerators, Stripes [18], ShapeShifter [7], and BitWave [9], based on a customized cycle-accurate simulator. HAP achieves an average speedup of  $1.60\times$  (HAP\_W8.0),  $2.16\times$  (HAP\_W5.6), and  $2.55\times$  (HAP\_W4.6), respectively.

## REFERENCES

- [1] E. Yvinec, A. Dapogny, M. Cord, and K. Bailly, “Spiq: Data-free per-channel static input quantization,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 3869–3878.
- [2] J. Moon, D. Kim, J. Cheon, and B. Ham, “Instance-aware group quantization for vision transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 16 132–16 141.
- [3] C. Xue, C. Zhang, X. Jiang, Z. Gao, Y. Lin, and G. Sun, “Oltron: Algorithm-hardware co-design for outlier-aware quantization of llms with inter-/intra-layer adaptation,” in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, ser. DAC ’24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: <https://doi.org/10.1145/3649329.3656221>
- [4] X. Geng, S. Liu, J. Jiang, K. Jiang, and H. Jiang, “Compact powers-of-two: An efficient non-uniform quantization for deep neural networks,” in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2024, pp. 1–6.
- [5] C. Guo, C. Zhang, J. Leng, Z. Liu, F. Yang, Y. Liu, M. Guo, and Y. Zhu, “Ant: Exploiting adaptive numerical data type for low-bit deep neural network quantization,” in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2022, pp. 1414–1433.
- [6] C. Guo, J. Tang, W. Hu, J. Leng, C. Zhang, F. Yang, Y. Liu, M. Guo, and Y. Zhu, “Olive: Accelerating large language models via hardware-friendly outlier-victim pair quantization,” in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ser. ISCA ’23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: <https://doi.org/10.1145/3579371.3589038>
- [7] A. D. Lascorz, S. Sharify, I. Edo, D. M. Stuart, O. M. Awad, P. Judd, M. Mahmoud, M. Nikolic, K. Siu, Z. Poulos, and A. Moshovos, “Shapeshifter: Enabling fine-grain data width adaptation in deep learning,” in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO ’52. New York, NY, USA: Association for Computing Machinery, 2019, p. 28–41. [Online]. Available: <https://doi.org/10.1145/3352460.3358295>
- [8] L. Liu, Z. Xu, Y. He, Y. Wang, H. Li, X. Li, and Y. Han, “Drift: Leveraging distribution-based dynamic precision quantization for efficient deep neural network acceleration,” in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, ser. DAC ’24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: <https://doi.org/10.1145/3649329.3655986>
- [9] M. Shi, V. Jain, A. Joseph, M. Meijer, and M. Verhelst, “Bitwave: Exploiting column-based bit-level sparsity for deep learning acceleration,” in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2024, pp. 732–746.
- [10] Y. Chen, J. Meng, J.-s. Seo, and M. S. Abdelfattah, “Bbs: Bi-directional bit-level sparsity for deep learning acceleration,” in *2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2024, pp. 551–564.
- [11] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, “SmoothQuant: Accurate and efficient post-training quantization for large language models,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 38 087–38 099. [Online]. Available: <https://proceedings.mlr.press/v202/xiao23c.html>
- [12] Y. Liu, H. Yang, Z. Dong, K. Keutzer, L. Du, and S. Zhang, “Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 20 321–20 330.
- [13] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, “Awq: Activation-aware weight quantization for on-device llm compression and acceleration,” in *Proceedings of Machine Learning and Systems*, P. Gibbons, G. Pekhimenko, and C. D. Sa, Eds., vol. 6, 2024, pp. 87–100.
- [14] D. Kam, M. Yun, S. Yoo, S. Hong, Z. Zhang, and Y. Lee, “Panacea: Novel dnn accelerator using accuracy-preserving asymmetric quantization and energy-saving bit-slice sparsity,” in *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2025, pp. 701–715.
- [15] X. Geng, S. Liu, L. Liu, J. Han, and H. Jiang, “Quq: Quadruplet uniform quantization for efficient vision transformer inference,” in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, ser. DAC ’24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: <https://doi.org/10.1145/3649329.3656516>
- [16] Y. Chen, A. F. AbouElhamayed, X. Dai, Y. Wang, M. Andronic, G. A. Constantinides, and M. S. Abdelfattah, “Bitmod: Bit-serial mixture-of-datatype llm acceleration,” in *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2025, pp. 1082–1097.
- [17] C. Guo, Y. Qiu, J. Leng, X. Gao, C. Zhang, Y. Liu, F. Yang, Y. Zhu, and M. Guo, “SQuant: On-the-fly data-free quantization via diagonal hessian approximation,” in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=JXhROKNZzOc>
- [18] P. Judd, J. Albericio, T. Hetherington, T. M. Aamodt, and A. Moshovos, “Stripes: Bit-serial deep neural network computing,” in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2016, pp. 1–12.