arXiv:2608.02700v1 [cs.LG] 3 Aug 2026

NANQ: Noise-Floor-Aware Mixed-Precision Non-Uniform Quantization for Analog Compute-in-Memory

Yizhe Chen[1]*, Wenshuai Yao[2]*, Saiya Wang[1], Yuannuo Feng[1], Wenbo Qi[3], Kechao Tang[2], Ngai Wong[3], Wenyong Zhou[3]†, Wang Kang[1]†

[1]School of Integrated Circuit Science and Engineering, Beihang University, Beijing, China
[2]School of Integrated Circuits, Peking University, Beijing, China
[3]Department of Electrical and Computer Engineering, The University of Hong Kong, Hong Kong SAR, China

Abstract

Analog compute-in-memory (CIM) enables energy-efficient neural network inference, but device variation and read noise can severely degrade low-bit quantized models. Existing CIM-oriented quantization methods mainly minimize ideal quantization error, ignoring the hardware noise floor and thus causing inefficient precision allocation. We propose NANQ, a noise-aware mixed-precision non-uniform quantization framework for analog CIM. NANQ models magnitude-dependent weight noise from measured responses of an eFlash CIM array and converts the noise profile into an adaptive quantization density, assigning finer resolution to low-noise regions while avoiding ineffective precision in noise-dominated regions. It further assigns layer-wise bit-widths by identifying each layer's precision saturation point under hardware noise using a unified threshold. On-chip experiments on an eFlash CIM SoC show that, under 2-bit weight-magnitude quantization, NANQ improves vision-model accuracy by 8.05 percentage points and reduces language-model PPL by 54.7% on average over PowerQuant. Mixed-precision NANQ captures most of the gains obtainable from additional quantization resources with only 3.2–3.8 equivalent bits.

Introduction

The exponential growth in deep neural network (DNN) complexity has driven the need for energy-efficient inference accelerators (Liu et al. 2025). Analog compute-in-memory (CIM) architectures have emerged as a promising solution, leveraging in-situ computation within memory arrays to achieve significant improvements in energy efficiency and computational density compared to traditional digital processors (Mao et al. 2025; Chen et al. 2025). By storing neural network weights directly in memory cells and performing matrix–vector multiplications through analog operations, CIM systems can reduce data movement overhead and enable massively parallel computation (Guo et al. 2017).

However, analog CIM implementations face fundamental challenges that limit their practical deployment (Wang et al. 2025). Device variations, thermal noise, and nonlinearities inherent in analog circuits introduce computational errors that can severely degrade neural network accuracy (Liu et al. 2019). These noise sources create discrepancies between the expected weights and the weights programmed in CIM arrays (Wang et al. 2023; Xiang et al. 2019).

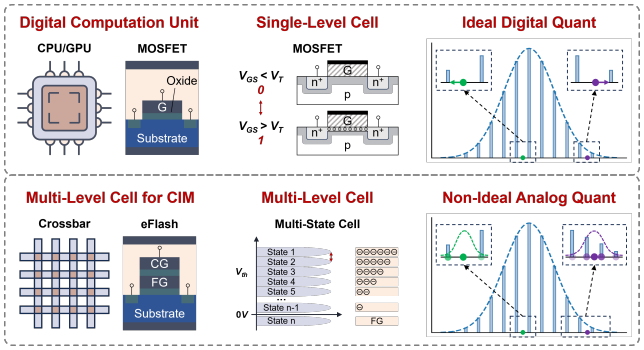

Figure 1: Fixed quantization levels spread into noisy distributions on analog hardware, causing diminishing returns with higher precision.

Due to the discrete nature of memory cell states and the limited precision of analog sensing circuits, CIM architectures inherently require model quantization (Bai et al. 2023; Sun et al. 2024; Chen et al. 2024). Previous CIM-oriented quantization schemes have fully explored the potential of uniform quantization, including mixed-precision quantization and extreme binary or ternary quantization, but have less explored non-uniform quantization, which is ideal for handling the distribution shifts caused by hardware non-idealities.

As shown in Figure 1, during the ideal digital quantization process, continuous weight values are mapped to the nearest discrete quantization levels on either side. However, after being programmed on analog CIM, each quantization level broadens into a noisy distribution whose width depends on the hardware noise strength.

To address these challenges, we propose noise-aware non-uniform quantization (NANQ), a hardware-aware framework for robust DNN inference on analog CIM accelerators. NANQ jointly considers noise variation across weight re-

*These authors contributed equally.
†Corresponding author.

Copyright © 2027, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

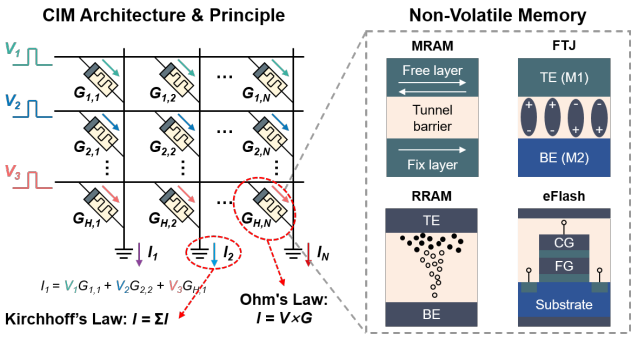

Figure 2: Schematic of an analog compute-in-memory array based on nonvolatile memory (NVM-CIM).

gions and network layers to optimize intra-layer quantization levels and inter-layer bit-widths. The main contributions are summarized as follows:

- We analyze the interaction between quantization error and chip-measured magnitude-dependent weight noise in analog CIM, showing that the benefit of increasing bit-width saturates once quantization error falls below the hardware noise floor.
- We propose NANQ, a training-free non-uniform quantization method that converts a hardware noise profile into an adaptive quantization density and constructs quantization boundaries by cumulative-density partitioning.
- We introduce a noise-aware mixed-precision allocation strategy that determines each layer's saturation bit-width from hardware-aware loss curves using a unified threshold, avoiding redundant precision without retraining or combinatorial search.

On-chip evaluations across CNNs, ViTs, and language models show that NANQ achieves the best quantized result in 38 of 45 model–bit-width configurations, with particularly consistent gains at low precision. Under matched quantization resources, mixed-precision NANQ further outperforms all evaluated baselines across the tested models.

Related Work

Analog CIM and Device Non-Idealities As illustrated in Figure 2, analog CIM maps neural network weights to device conductance states and converts input activations into word-line voltages. The resulting cell currents are accumulated along each column, enabling parallel matrix–vector multiplication in the analog domain. This computing principle can be implemented using magnetic, resistive, and floating-gate memory devices.

Practical devices, however, deviate from ideal linear computation. Process variation, stochastic programming, oxide defects, and charge trapping introduce quasi-static cell-to-cell conductance variation that persists across inference operations (Georgiev et al. 2014). Nonlinear weight encoding, input-voltage conversion, and device I–V characteristics further produce systematic input- and weight-dependent

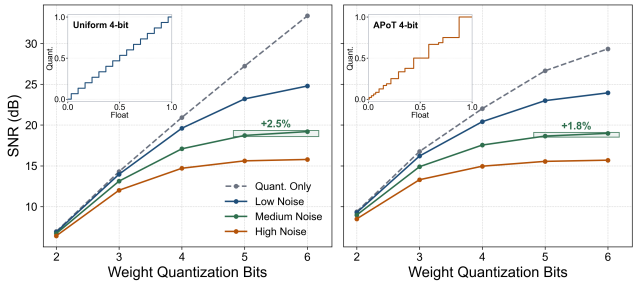

Figure 3: Output SNR versus weight quantization bit-width under different noise conditions.

distortions, which are repeatable under fixed operating conditions and can be partially mitigated through calibration (Serm et al. 2025). In addition, thermal, flicker, and peripheral readout noise cause temporal fluctuations that vary across inference passes (Feng et al. 2026). Together, these non-idealities cause the accumulated column outputs to deviate from ideal matrix–vector multiplication.

Non-Uniform Quantization Non-uniform quantization improves low-bit representation by redistributing quantization levels over the weight range. APoT constructs structured levels using sums of powers of two to better represent long-tailed weight distributions (Li et al. 2020), while PowerQuant searches layer-wise power transformations for data-free non-uniform quantization (Yvinec et al. 2023). SqueezeLLM further uses second-order sensitivity to construct non-uniform codebooks and preserves sensitive outliers with sparse high-precision representations (Kim et al. 2023). Nevertheless, these methods primarily optimize weight distributions, digital reconstruction error, or model sensitivity, without considering the magnitude-dependent hardware noise of analog CIM.

Mixed-Precision Quantization Mixed-precision quantization assigns different bit-widths to network layers to better balance model accuracy and resource consumption. PNMQ jointly optimizes parametric non-uniform quantization grids and layer-wise bit-widths under a specified compression budget without retraining (Chikin and Antiukh 2022). OMPQ uses network orthogonality as an efficient proxy for layer importance and determines the bit-width configuration through linear programming (Ma et al. 2023). More recently, InfoQ measures the global impact of layer-wise quantization on information flow and formulates precision allocation as an integer linear programming problem (Akbulut et al. 2026). Although these methods efficiently capture layer-dependent precision requirements, their allocation criteria are derived from ideal digital quantization and do not account for the hardware noise floor of analog CIM.

Methodology

Motivation

As shown in Figure 3, conventional quantization determines quantization levels primarily by minimizing ideal weight-domain reconstruction error. After the quantized weights are

mapped to physical conductance states, however, each level is further perturbed by magnitude-dependent hardware noise. Once the local quantization error falls below the hardware noise floor, finer quantization provides little effective improvement. Consequently, quantization boundaries derived solely from ideal reconstruction error may waste resolution in noise-dominated regions while providing insufficient resolution in more reliable regions. This mismatch motivates adaptively redistributing intra-layer quantization levels according to the measured hardware noise profile.

We further examine whether different layers benefit equally from increasing weight precision. Using uniform quantization, we evaluate each layer independently under hardware noise. Let $\mathcal{L}_l(b)$ denote the validation loss when layer l uses b-bit weight precision, and let N_l denote its number of weights. Since increasing the precision of this layer by one bit introduces N_l additional weight bits, we define the marginal loss reduction per added weight bit as

$$G_l(b) = \frac{\max(\mathcal{L}_l(b) - \mathcal{L}_l(b+1), 0)}{N_l}. \tag{1}$$

As shown in Figure 4, different layers exhibit substantially different marginal gains at the same bit-width, while the gain of each layer generally decreases as precision increases. Some layers approach saturation at low precision, whereas others continue to benefit from additional bits. This heterogeneity is consistently observed in both CNNs and LLMs, indicating that a uniform bit-width may allocate redundant precision to already saturated layers. This observation motivates identifying a layer-specific saturation bit-width and avoiding redundant precision allocation to layers whose hardware-aware gains have already diminished.

Theoretical Analysis

We analyze why increasing the quantization bit-width provides diminishing returns under analog hardware noise. For a weight region around w, the total error variance can be approximated as

$$\sigma^2_{\text{tot}}(b, w) = \sigma^2_q(b, w) + \sigma^2_h(w), \tag{2}$$

where $\sigma^2_q(b, w)$ and $\sigma^2_h(w)$ denote the local quantization-error variance and hardware noise variance, respectively. Assuming a local quantization step $\Delta_b(w)$ at bit-width b, the quantization-error variance can be approximated as

$$\sigma^2_q(b, w) \approx \frac{\Delta^2_b(w)}{12} \propto 2^{-2b}. \tag{3}$$

The corresponding output SNR is therefore expressed as

$$\text{SNR}(b, w) = \frac{P_{\text{sig}}(w)}{\sigma^2_q(b, w) + \sigma^2_h(w)}, \tag{4}$$

where $P_{\text{sig}}(w)$ denotes the signal power associated with the considered weight region.

Increasing b exponentially reduces $\sigma^2_q(b, w)$ but does not reduce the hardware noise term. Once $\sigma^2_q(b, w) \ll \sigma^2_h(w)$, the SNR approaches $P_{\text{sig}}(w)/\sigma^2_h(w)$, and additional bits provide little improvement. Moreover, weight regions or network layers with stronger hardware noise reach this precision-saturation point at lower bit-widths. Therefore, quantization resolution should be concentrated in relatively reliable weight regions, while additional bits should be assigned only to layers that still exhibit meaningful marginal gains.

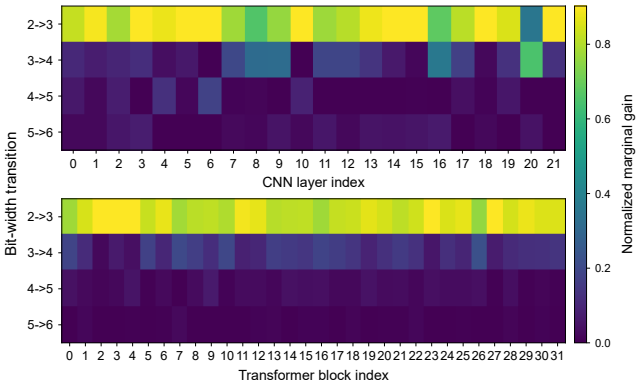

Figure 4: Layer-wise marginal gains from increasing weight precision under analog CIM noise: ResNet-20 on CIFAR-100 (Krizhevsky, Hinton et al. 2009) (top) and Llama-3.2-3B on WikiText-2 (Merity et al. 2016) (bottom).

This analysis establishes the precision-saturation effect but does not uniquely determine the optimal quantization density. Motivated by this observation, we adopt a lightweight inverse-noise density in the following section.

Adaptive Quantization Algorithm

As illustrated in Figure 5, NANQ converts the magnitude-dependent hardware noise profile into a non-uniform quantization density, which is then used to construct noise-adaptive quantization boundaries.

Figure 5(a) and (b) illustrate the normalized weight-magnitude domain and its corresponding noise-intensity profile, respectively. We characterize this profile using a noise strength function $\sigma(w)$, where $\sigma(w) \geq 0$ represents the relative noise intensity at magnitude w. For the evaluated eFlash-based CIM SoC, $\sigma(w)$ is obtained directly from repeated on-chip readback measurements at different programmed weight magnitudes. Here, $\sigma(w)$ provides a device-characterization prior for constructing the quantizer rather than serving as an explicit inference-noise simulator. Quantizer calibration and end-to-end evaluation are performed using real-chip responses. This formulation does not impose a specific analytical form on the underlying noise model and can therefore accommodate different analog CIM devices and technologies. Based on $\sigma(w)$, we define the inverse quantization-density function shown in Figure 5(c):

$$\rho(w) = \frac{1}{(\sigma(w) + \epsilon)^\gamma}, \tag{5}$$

where $\gamma \geq 0$ controls the adaptation strength and $\epsilon > 0$ ensures numerical stability. Through this inverse relationship, low-noise regions receive a higher quantization density, whereas fewer quantization resources are assigned to regions dominated by hardware noise.

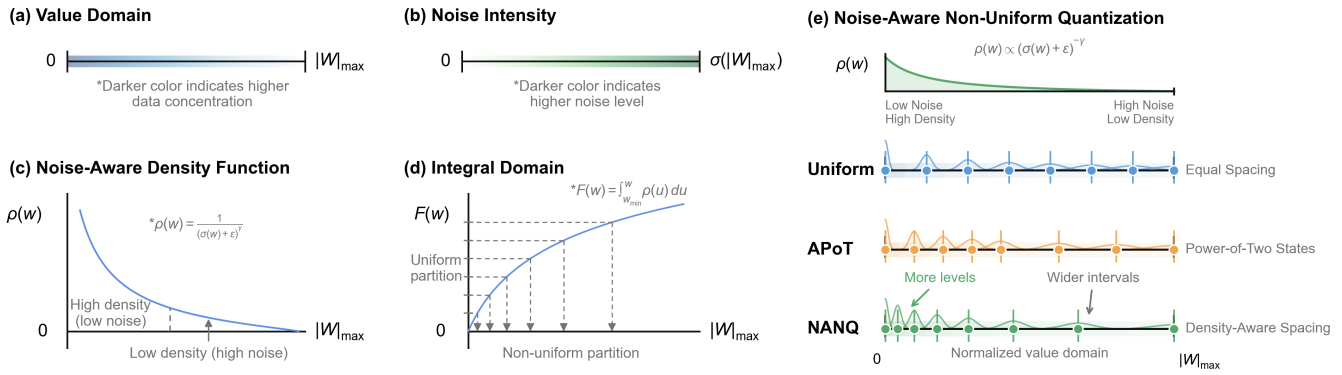

Figure 5: Overview of our NANQ method. Building on existing quantization methods, NANQ adjusts quantization points based on the noise sensitivity metric of values prior to quantization.

The parameters γ and ϵ provide flexible control over the quantization-level distribution. When $\gamma = 0$, the density becomes constant, and the method reduces to uniform quantization regardless of the noise characteristics. As γ increases, the density becomes more sensitive to noise variation, strengthening the redistribution between high-noise and low-noise regions. Meanwhile, ϵ prevents numerical instability and excessive concentration of quantization levels in regions where $\sigma(w)$ approaches zero.

To transform the continuous density into discrete quantization intervals, we compute the cumulative density shown in Figure 5(d):

$$F(w) = \int_0^w \rho(u)\, du. \tag{6}$$

Given Q magnitude intervals determined by the selected magnitude bit-width, the boundaries $\{e_i\}$ are obtained by equally partitioning the cumulative density:

$$F(e_i) = \frac{i}{Q} F(w_{\max}), \qquad i = 0, 1, \ldots, Q. \tag{7}$$

The reconstruction level and signed quantized weight are then defined as

$$q_i = \frac{e_i + e_{i+1}}{2}, \qquad \widehat{w} = \operatorname{sign}(w) q_i, \quad |w| \in [e_i, e_{i+1}). \tag{8}$$

The empirical weight distribution in Figure 5(a) is shown only to visualize parameter concentration and is not used to construct $\rho(w)$. Since each interval contains the same cumulative noise-aware density, low-noise regions receive narrower intervals and finer resolution, whereas high-noise regions are represented using wider intervals.

As summarized in Figure 5(e), uniform quantization uses equally spaced levels, while APoT remains constrained by predefined power-of-two combinations. In contrast, NANQ directly converts the hardware noise profile into density-aware quantization spacing. This CDF-based construction preserves the fixed budget of Q intervals and generates noise-adaptive boundaries without explicit iterative boundary optimization.

Noise-Aware Mixed-Precision Allocation

Motivated by the layer-wise differences in precision saturation under hardware noise, we use a unified saturation threshold to independently determine the weight precision of each layer. Consider a network with L quantized layers and a candidate magnitude-bit set $\mathcal{B} = \{b_{\min}, \ldots, b_{\max}\}$, where the sign bit is excluded. For layer l and candidate bit-width b, we apply NANQ quantization and hardware noise only to this layer while keeping all other layers in full precision. The resulting noisy quantized weight is denoted by

$$\widehat{W}_l^{b,\gamma,\xi} = \widetilde{Q}_{\mathrm{NANQ}}(W_l; b, \gamma, \xi), \tag{9}$$

where $\gamma \in \Gamma$ controls the adaptation of the quantization levels to the hardware-noise profile, and ξ is sampled from the measured noise distribution p_{hw}. We then define the minimum hardware-aware loss of layer l at bit-width b as

$$E_l(b) = \min_{\gamma \in \Gamma} \mathbb{E}_{\xi \sim p_{\mathrm{hw}}} \left[\mathcal{L}\big(f(W_{-l}, \widehat{W}_l^{b,\gamma,\xi}); \mathcal{D}_{\mathrm{cal}}\big) \right], \tag{10}$$

where $\mathcal{D}_{\mathrm{cal}}$ denotes the calibration set and W_{-l} represents the full-precision weights of all layers except layer l. The corresponding optimal noise-adaptation parameter is denoted by $\gamma_l^*(b)$.

Finite calibration data and stochastic noise sampling may introduce local fluctuations into $\{E_l(b)\}_{b \in \mathcal{B}}$. We therefore fit a monotonically non-increasing loss curve, denoted by $\widetilde{E}_l(b)$. The fraction of recoverable loss reduction remaining beyond bit-width b is then defined as

$$R_l(b) = \frac{\widetilde{E}_l(b) - \widetilde{E}_l(b_{\max})}{\widetilde{E}_l(b_{\min}) - \widetilde{E}_l(b_{\max}) + \epsilon}. \tag{11}$$

This normalization removes the scale differences among layers and measures the relative benefit that remains when increasing the precision from b to $b_{\max}$. Given a global saturation threshold τ, the bit-width of layer l is selected as

$$b_l^*(\tau) = \min \{ b \in \mathcal{B} \mid R_l(b) \le \tau \}. \tag{12}$$

For example, $\tau = 0.1$ indicates that the selected bit-width has already captured at least 90% of the recoverable loss

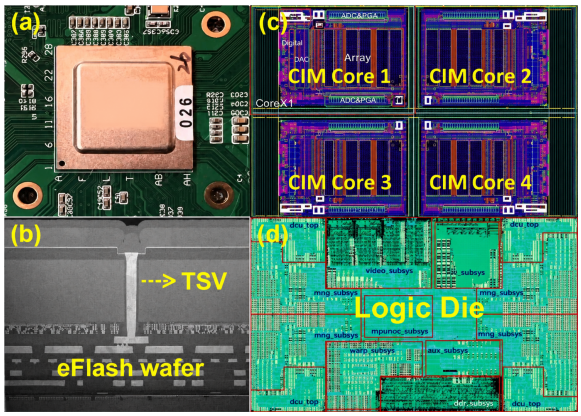

Figure 6: Photographs of the 3D eFlash CIM SoC. (a) Packaged SoC and test board; (b) TEM cross-section of the 3D stack; (c) Flash die layout; (d) Logic die layout.

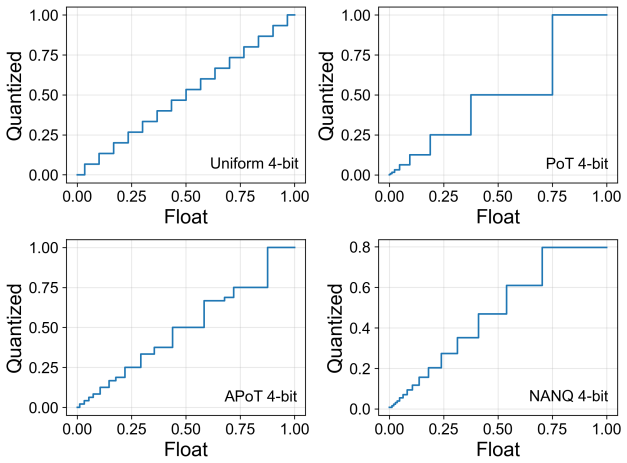

Figure 7: Comparison of quantization steps across different methods.

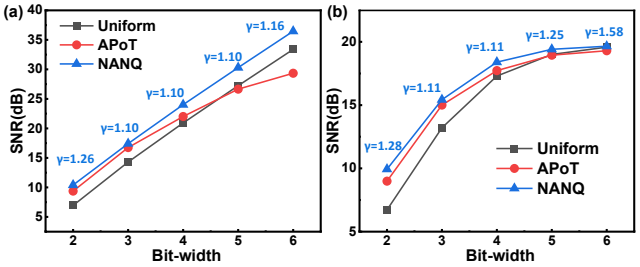

Figure 8: Single-MAC output SNR under (a) quantization only and (b) on-chip quantization with hardware noise.

reduction between $b_{\min}$ and $b_{\max}$. If the total recoverable reduction of a layer is negligible, we directly set $b_l^* = b_{\min}$ to avoid assigning redundant precision to an already saturated layer.

The final configuration assigns each layer l the pair $\big(b_l^*(\tau), \gamma_l^*(b_l^*(\tau))\big)$, where a smaller τ generally results in higher precision. To account for cross-layer error accumulation, we evaluate a small set of candidate thresholds on the full model and select the point beyond which further decreasing τ provides negligible end-to-end improvement. The resulting resource cost is reported using the equivalent average bit-width

$$B_{\text{eq}}(\tau) = \frac{\sum_{l=1}^{L} N_l b_l^*(\tau)}{\sum_{l=1}^{L} N_l}, \tag{13}$$

where N_l is the number of weights in layer l. This procedure requires only layer–bit evaluations and a few full-model validations, without retraining or combinatorial bit-width search.

Experiments

Experiment Setup

We conduct end-to-end evaluations on representative CNN and Transformer models using the eFlash CIM SoC shown in Figure 6. For CNNs, convolutional and fully connected layers are mapped onto the physical CIM array, while QKV projection and FFN layers are mapped for ViTs and LLMs. Layers exceeding the on-chip capacity are partitioned into tiles and executed sequentially, whereas all remaining operations, including batch normalization and softmax, are performed on an external NPU. Weights are quantized per channel, while activations are quantized per token.

Experiment Results

MAC-Level Quantization Analysis As shown in Figure 7, uniform quantization distributes levels evenly across the entire range, which does not match the bell-shaped weight distribution and may waste levels in low-density regions. PoT (Zhou et al. 2017) concentrates quantization levels near zero, but increasing the bit-width mainly adds levels to the small-magnitude region, leaving limited resolution for larger values. APoT improves the coverage by combining multiple power-of-two terms, yet its level distribution is still restricted by predefined structures. In contrast, NANQ introduces only one hyperparameter to flexibly reshape the quantization levels according to the magnitude-dependent hardware noise profile. It remains training-free and requires only a few inference trials for parameter search, making it more suitable for noisy analog CIM inference.

As shown in Figure 8, NANQ achieves higher single-MAC output SNR, particularly under low-bit on-chip inference. Since neural network weights are typically concentrated near zero and the measured on-chip noise profile increases approximately with $|W|$, the small-magnitude region simultaneously exhibits a higher weight density and a lower noise level. NANQ therefore allocates more quantization levels to this region. Under the quantization-only setting, such allocation utilizes the limited quantization levels more effectively and reduces the dominant quantization error. With on-chip hardware noise, NANQ further avoids ineffective precision allocation to large-magnitude regions with stronger noise, leading to a more pronounced robustness advantage at low bit-widths.

(a) Vision Models: Top-1 Accuracy (%) ↑

Family	Model	Method	Baseline	$B_m = 2$	$B_m = 3$	$B_m = 4$	$B_m = 5$	$B_m = 6$
ResNet CIFAR-100	20	Uniform	62.89	2.35	24.64	54.90	56.43	57.29
		APoT (Li et al. 2020)		4.21	46.27	55.57	55.81	56.95
		PowerQuant (Yvinec et al. 2023)		12.63	50.84	56.61	**58.14**	57.23
		NANQ		**19.84**	**53.92**	**57.24**	57.89	**57.56**
VGG CIFAR-100	11	Uniform	66.88	1.16	50.46	64.79	65.88	66.18
		APoT (Li et al. 2020)		3.70	61.59	66.28	66.04	66.13
		PowerQuant (Yvinec et al. 2023)		31.86	64.33	66.40	66.22	**66.35**
		NANQ		**56.27**	**66.72**	**66.51**	**66.45**	66.27
ViT ImageNet-1K	Tiny	Uniform	75.46	3.78	61.50	70.80	72.40	72.77
		APoT (Li et al. 2020)		**16.37**	65.57	70.61	72.15	72.54
		PowerQuant (Yvinec et al. 2023)		15.94	66.03	71.14	72.41	72.70
		NANQ		15.28	**66.58**	**71.61**	**72.67**	**72.83**
	Base	Uniform	85.11	34.73	81.86	84.20	84.54	84.67
		APoT (Li et al. 2020)		70.96	83.17	84.29	84.54	84.66
		PowerQuant (Yvinec et al. 2023)		72.18	83.51	84.36	84.58	**84.75**
		NANQ		**73.40**	**83.78**	**84.43**	**84.61**	84.71

(b) Language Models on WikiText-2: PPL ↓

Family	Model	Method	Baseline	$B_m = 2$	$B_m = 3$	$B_m = 4$	$B_m = 5$	$B_m = 6$
Pythia	410M	Uniform	16.96	7.61k	46.04	24.87	22.10	21.70
		APoT (Li et al. 2020)		615.3	32.13	24.00	22.32	21.92
		PowerQuant (Yvinec et al. 2023)		431.8	30.82	23.55	22.05	22.08
		NANQ		**294.1**	**29.73**	**23.10**	**21.84**	**21.64**
	1B	Uniform	11.61	2.54k	16.64	13.19	12.64	12.49
		APoT (Li et al. 2020)		184.1	14.55	12.95	12.65	12.50
		PowerQuant (Yvinec et al. 2023)		112.6	14.31	12.83	12.58	**12.45**
		NANQ		**66.83**	**14.10**	**12.72**	**12.53**	12.47
Llama-3.2	1B	Uniform	8.620	35.0k	43.99	11.56	10.12	9.869
		APoT (Li et al. 2020)		3.56k	16.94	10.56	10.02	9.886
		PowerQuant (Yvinec et al. 2023)		1.84k	14.72	10.41	9.950	9.840
		NANQ		**650.6**	**12.87**	**10.30**	**9.888**	**9.808**
	3B	Uniform	6.941	11.1k	13.62	8.223	7.619	7.495
		APoT (Li et al. 2020)		916.1	9.916	7.777	7.566	7.511
		PowerQuant (Yvinec et al. 2023)		482.7	9.210	7.720	**7.480**	7.489
		NANQ		**129.1**	**8.557**	**7.668**	7.512	**7.476**
OPT	1.3B	Uniform	12.56	13.4k	286.6	14.92	13.81	13.79
		APoT (Li et al. 2020)		7.62k	23.51	14.04	13.85	13.77
		PowerQuant (Yvinec et al. 2023)		4.18k	19.07	14.18	13.80	**13.75**
		NANQ		**1.54k**	**15.16**	**13.92**	**13.77**	13.76

Table 1: On-chip inference performance on (a) vision models using CIFAR-100 and ImageNet-1K (Russakovsky et al. 2015), and (b) language models using WikiText-2. Quantized results are averaged over three independent on-chip inference runs; Baseline denotes clean BF16 inference. Shaded bold values indicate the best quantized result for each model and B_m.

End-to-End Inference Evaluation Many non-uniform quantization methods require retraining or iterative optimization of thresholds and codebooks. We compare NANQ with Uniform quantization, APoT, and PowerQuant, a representative data-free non-uniform method. CIMQ and CIM^{2}PQ jointly optimize array-wise input, weight, and partial-sum precision under architecture-specific hardware objectives and are therefore not directly comparable under our weight-only evaluation setting. B_m denotes the weight-magnitude bit-width excluding the sign bit and ranges from 2 to 6 bits, while activations are fixed at INT8.

As shown in Table 1, NANQ achieves the best mean performance in 38 of the 45 model–bit-width configurations, with the largest gains at 2–3 bits. Compared with PowerQuant, at $B_m = 2$, NANQ improves the accuracy of ResNet-20, VGG-11, and ViT-Base by 7.21, 24.41, and 1.22 percentage points, respectively, and reduces language-model PPL by up to 73.3%. At $B_m = 3$, its PPL reductions over PowerQuant range from 1.5% to 20.50%, with smaller model variants generally benefiting more, suggesting that larger models are inherently more robust to low-bit perturbations. The advantage narrows at higher precision, with PowerQuant slightly outperforming NANQ in a few 5–6-bit settings, as denser quantization levels reduce the benefit of noise-aware level allocation. Overall, NANQ is particularly effective for low-bit analog CIM inference where hardware noise is dominant.

We further evaluate mixed-precision quantization under matched precision budgets. For each model, NANQ uses $\tau = 0.1$ to determine the layer-wise bit-widths and the corresponding target equivalent bit-width B_{eq}. Under the same budget, APoT and PowerQuant start from the minimum bit-width and greedily increase the precision of the layer with

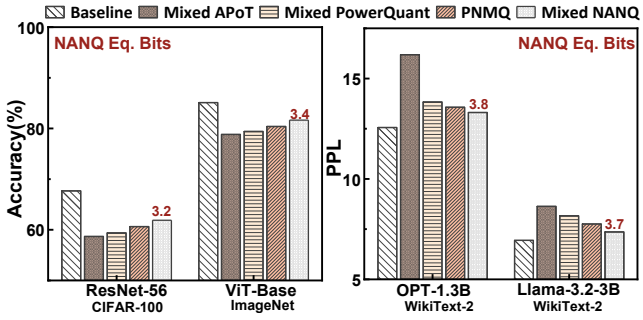

Figure 9: Comparison of mixed-precision quantization methods on vision and language models (τ = 0.10).

B_m	γ^*	$\gamma = \gamma^* + \Delta\gamma$				
		-0.2	-0.1	0	$+0.1$	$+0.2$
2	1.23	392.973 ±46.334	370.075 ±20.973	**294.091 ±16.292**	303.869 ±15.394	338.122 ±36.829
3	1.15	32.588 ±0.254	31.014 ±0.318	**29.728 ±0.286**	30.314 ±0.254	32.433 ±0.604
4	1.09	23.206 ±0.173	23.675 ±0.180	**23.102 ±0.206**	23.177 ±0.310	23.476 ±0.352
5	0.93	22.078 ±0.124	21.900 ±0.216	**21.842 ±0.153**	22.004 ±0.277	21.991 ±0.249
6	1.42	21.749 ±0.188	21.712 ±0.173	**21.642 ±0.193**	21.717 ±0.242	21.732 ±0.196

Table 2: PPL of Pythia-410M on WikiText-2 under perturbations around the selected γ^*. Results are reported as mean $\pm$ standard deviation. Lower is better.

the largest marginal gain per added weight bit until the budget is exhausted. PNMQ (Chikin and Antiukh 2022) follows its original allocation strategy under the same budget. This protocol allows each method to independently optimize its layer-wise configuration while avoiding confounding effects from different quantization resources.

As shown in Figure 9, Mixed NANQ achieves the best quantized performance on all evaluated models, with adaptively determined equivalent bit-widths ranging from 3.2 to 3.8 bits. Compared with PNMQ, NANQ improves accuracy by 1.26 percentage points on both CIFAR-100 ResNet-56 and ImageNet-1K ViT-Base, while reducing the PPL of OPT-1.3B and Llama-3.2-3B from 13.57 and 7.760 to 13.31 and 7.361, respectively. Compared with Mixed PowerQuant, NANQ improves accuracy by 2.52 and 2.23 percentage points on the two vision models and reduces PPL by 3.8% and 9.9% on the two language models. These results show that NANQ improves performance under matched resources through noise-aware quantization and layer-wise precision allocation.

Ablation Study

Table 2 evaluates the sensitivity to the selected γ^* by perturbing it within ± 0.2. The selected γ^* consistently achieves the lowest mean PPL for all B_m values, validating our parameter selection. The impact of γ is most significant at $B_m = 2$, where the PPL increases from 294.091 to as high as 392.973 and the standard deviation also becomes larger. As B_m increases, the maximum PPL degradation decreases from about 9.6% at $B_m = 3$ to below 0.5% at $B_m = 6$, indicating that γ is critical under aggressive low-bit quantization, while the method remains robust at higher bit-widths.

τ	B_{eq}	PPL ↓
0	5.86	21.87 ± 0.18
0.05	4.63	22.04 ± 0.20
0.10	**3.76**	$\mathbf{22.68 \pm 0.22}$
0.20	3.42	24.74 ± 0.27
0.30	3.16	26.31 ± 0.34
0.50	2.72	29.68 ± 0.48

Table 3: Effect of τ on the equivalent bit-width and PPL of Pythia-410M on WikiText-2. Results are reported as mean $\pm$ standard deviation.

Table 3 shows the effect of the saturation threshold τ on the equivalent bit-width and model performance. Decreasing τ from 0.1 to 0.05 and 0 increases B_{eq} from 3.76 to 4.63 and 5.86, while improving PPL only modestly from 22.68 to 22.04 and 21.87. In contrast, increasing τ to 0.2 or above further reduces the equivalent bit-width but noticeably increases both PPL and its standard deviation. These results indicate that $\tau = 0.1$ provides a favorable trade-off between model performance and equivalent precision.

Discussion and Limitations

NANQ can be extended to other multi-level analog memories through device-specific noise characterization, although substantial changes in temperature, aging, or operating conditions may require the noise profile to be updated. The reported B_{eq} is an algorithm-level precision metric rather than a direct measure of physical storage, latency, or energy savings; realizing such benefits requires hardware-aware mapping and compiler support. In addition, the current framework approximates CIM non-idealities as magnitude-dependent weight perturbations. Future work may separately model and jointly optimize quantization for weights, activations, and outputs.

Conclusion

This work presents NANQ, a training-free, noise-aware mixed-precision non-uniform quantization framework for analog CIM. NANQ adaptively assigns quantization levels according to chip-measured magnitude-dependent weight noise and determines layer-wise bit-widths from their precision saturation under hardware noise. On-chip experiments on an eFlash CIM SoC show that NANQ significantly outperforms existing methods at low precision, while mixed-precision NANQ captures most attainable quantization gains using only 3.2–3.8 equivalent bits. These results demonstrate that explicitly accounting for the hardware noise floor enables more efficient use of limited precision in analog CIM inference.

References

Akbulut, M. E.; Shalby, H. H. Y.; Pittorino, F.; and Roveri, M. 2026. Infoq: Mixed-precision quantization via global information flow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 19598–19606.

Bai, J.; et al. 2023. CIMQ: A hardware-efficient quantization framework for computing-in-memory-based neural network accelerators. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(1): 189–202.

Chen, Y.; et al. 2024. Device Characteristic-Aware Quantization for eFlash-Based In-Memory Computing SoC. In *2024 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*, 70–71. IEEE.

Chen, Y.; et al. 2025. A Reconfigurable Heterogeneous In-Memory Computing Architecture for Variable Precision Computation: A Software-Hardware Co-Design Approach. *Moore and More*, 2(1): 13.

Chikin, V.; and Antiukh, M. 2022. Data-free network compression via parametric non-uniform mixed precision quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 450–459.

Feng, Y.; Zhou, W.; Lv, Y.; Liu, H.; Wang, G.; Liu, Z.; Wong, N.; and Kang, W. 2026. Noise-Aware Adaptive Sampling for Robust Diffusion Models on Analog Compute-in-Memory. In *2026 Design, Automation & Test in Europe Conference (DATE)*, 1–3.

Georgiev, V. P.; Markov, S.; Vilà-Nadal, L.; Busche, C.; Cronin, L.; and Asenov, A. 2014. Optimization and Evaluation of Variability in the Programming Window of a Flash Cell With Molecular Metal–Oxide Storage. *IEEE Transactions on Electron Devices*, 61(6): 2019–2026.

Guo, X.; et al. 2017. Fast, Energy-Efficient, Robust, and Reproducible Mixed-Signal Neuromorphic Classifier Based on Embedded NOR Flash Memory Technology. In *2017 IEEE International Electron Devices Meeting (IEDM)*, 6.5.1–6.5.4.

Kim, S.; Hooper, C.; Gholami, A.; Dong, Z.; Li, X.; Shen, S.; Mahoney, M. W.; and Keutzer, K. 2023. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.(2009).

Li, Y.; et al. 2020. Additive Powers-of-Two Quantization: An Efficient Non-uniform Discretization for Neural Networks. In *8th International Conference on Learning Representations (ICLR)*, 1–15.

Liu, H.; et al. 2025. Efficient Weight Mapping and Resource Scheduling on Crossbar-based Multi-core CIM Systems. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, 1–6.

Liu, T.; et al. 2019. A Fault-Tolerant Neural Network Architecture. In *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 1–6.

Ma, Y.; Jin, T.; Zheng, X.; Wang, Y.; Li, H.; Wu, Y.; Jiang, G.; Zhang, W.; and Ji, R. 2023. Ompq: Orthogonal mixed precision quantization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 9029–9037.

Mao, W.; et al. 2025. HyIMC: Analog-Digital Hybrid In-Memory Computing SoC for High-Quality Low-Latency Speech Enhancement. In *2025 Design, Automation & Test in Europe Conference (DATE)*, 1–2.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252.

Serm, T.-B.; Chen, T.-C.; Wang, P.-K.; Wu, Y.-D.; Chan, T.-Y.; Chiang, M.-H.; and Ho, Y.-H. 2025. Experimental and modeled analysis of source-line loading effects in NOR flash compute-in-memory arrays. *Applied Physics Letters*, 127(20).

Sun, S.; et al. 2024. CIM^{2}PQ: An Arraywise and Hardware-Friendly Mixed Precision Quantization Method for Analog Computing-in-Memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(7): 2084–2097.

Wang, G.; et al. 2023. A 40nm 5-16Tops/W@INT8 eFlash In-Memory Computing SoC Chip with Noise Suppression and Compensation Techniques to Improve the Accuracy. In *2023 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*, 128–129.

Wang, G.; et al. 2025. NoiseGuard: A Comprehensive Framework with Noise Modeling, Noise-Aware Training and Noise Compensation for In-Memory Computing SoC. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1–1.

Xiang, Y. C.; et al. 2019. Analog Deep Neural Network Based on NOR Flash Computing Array for High Speed/Energy Efficiency Computation. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–4.

Yvinec, E.; Dapogny, A.; Cord, M.; and Bailly, K. 2023. Powerquant: Automorphism search for non-uniform quantization. *arXiv preprint arXiv:2301.09858*.

Zhou, A.; et al. 2017. Incremental Network Quantization: Towards Lossless CNNs with Low-precision Weights. In *5th International Conference on Learning Representations (ICLR)*, 1–14.