2025 Design, Automation & Test in Europe Conference (DATE 2025)

NORA: Noise-Optimized Rescaling of LLMs on Analog Compute-in-Memory Accelerators

Yayue Hou*, Hsinyu Tsai†, Kaoutar El Maghraoui‡, Tayfun Gokmen‡, Geoffrey W. Burr†, Liu Liu*
*Rensselaer Polytechnic Institute, Troy, USA
†IBM Research, San Jose, USA, ‡IBM Research, Yorktown Heights, USA
*{houy4, liu.liu}@rpi.edu, †{htsai, gwburr}@us.ibm.com, ‡{kelmaghr, tgokmen}@us.ibm.com

***Abstract*—Large Language Models (LLMs) have become critical in AI applications, yet current digital AI accelerators suffer from significant energy inefficiencies due to frequent data movement. Analog compute-in-memory (CIM) accelerators offer a potential solution for improving energy efficiency but introduce non-idealities that can degrade LLM accuracy. While analog CIM has been extensively studied for traditional deep neural networks, its impact on LLMs remains unexplored, particularly concerning the large influence of Analog CIM non-idealities. In this paper, we conduct a sensitivity analysis on the effects of analog-induced noise on LLM accuracy. We find that while LLMs demonstrate robustness to weight-related noise, they are highly sensitive to quantization noise and additive Gaussian noise. Based on these insights, we propose a noise-optimized rescaling method to mitigate LLM accuracy loss by shifting the non-ideality burden from the sensitive input/output to the more resilient weight. Through rescaling, we can implement the OPT-6.7b model on simulated analog CIM hardware with less than 1% accuracy loss from the floating-point baseline, compared to a much higher loss of around 30% without rescaling.**

I. Introduction

Large Language Models (LLMs) have become essential for many AI-driven daily tasks. However, the increasing size of LLMs, such as OPT [37] with up to 175B parameters and LLaMA-3.1 [5] with up to 405B parameters, leads to challenges in data movement between computing cores and memory. To break the memory wall, computing in memory (CIM), which avoids intensive data transfer by directly executing matrix-vector multiplications (MVM) in memory devices, has been applied to DNN acceleration [2], [8], [17]–[20], [29], [34], [35].

Among diverse CIM devices, non-volatile memory (NVM) attracts much attention in recent years. By exploiting Ohm's law and Kirchhoff's law, the time complexity of matrix-vector multiplication has been decreased from $O(n)$ to $O(1)$. Moreover, NVM-based CIM achieves high energy efficiency thanks to low access energy and high storage density [32].

However, directly mapping LLMs on analog CIM tiles leads to unacceptable accuracy degradation. First, input and output activations on CIM devices have to be quantized into lower precision compared with digital cores due to the energy and area constraints of high-resolution Analog/Digital converters [2], [26]. Unlike conventional Deep Neural Networks (DNNs), during the intrinsic A/D conversion process, LLMs suffer from severe accuracy degradation due to the outliers in activations [4], i.e., those activations with much larger magnitude than others. Second, several non-idealities exist due to analog components used in analog CIM [14], which has a significant influence on the Transformer-based LLM quality [28].

Hardware-aware training has been studied to achieve comparable model accuracy in analog CIM with digital counterparts. Prior studies on hardware-aware training consider a diverse set of non-idealities such as quantization noise, weight drifting, programming noise, IR drop, device non-linearity, and additive noise [11], [28]. However, most previous works [11]–[13], [28] require hardware-aware training, which is non-trivial, if not prohibitive for LLMs with large number of parameters. In addition, those HWA training methods only focus on conventional DNNs without considering the properties of LLMs. For instance, most LLMs have larger input ranges caused by outliers compared with conventional DNNs. Hence, the noise management and bound management in previous works [7], [14], [28] could become less effective in LLMs due to the input distribution, as shown in Figure 1, and will be further discussed in Section II and Section III.

In this paper, we first conduct a sensitivity study on analog CIM non-idealities across a set of LLMs. **Our key observation** (as shown in Figure 1 right top) is that *LLMs are sensitive to some input/output-related non-idealities while generally resilient to weight-related non-idealities.* Specifically, we find that additive Gaussian noise at the input/output interface introduced by analog components and DAC/ADC quantization noise has the most significant contribution to the accuracy degradation of LLMs. On the other hand, weight-related short-term noises such as weight-read noise have nearly no influence on LLMs performance in analog CIM. Furthermore, weight programming [3], [21], [22], as well as weight-related long-term non-ideality compensation techniques [14], [28], are well studied.

To this end, we propose a **N**oise-**O**ptimized **R**escaling method on LLM weights and activations for **A**nalog CIM accelerators, namely NORA. Our method can move the "non-ideality burden" from sensitive input/output to resilient weight, as shown in Figure 1 (right bottom). We introduce an additional component into scaling factors of weight and input to adjust the weight and input range of LLMs, which can not only mitigate quantization noise but also increase the robustness of Analog CIM to other non-idealities. Our method applied at post-training time can achieve near lossless model accuracy, without costly fine-tuning, on LLMs such as OPT [37], Mistral [10],

This work was supported in part by IBM-RPI Future of Computing Research Collaboration and the National Science Foundation under Grant No. 2442271.

978-3-9826741-0-0/DATE25/© 2025 EDAA

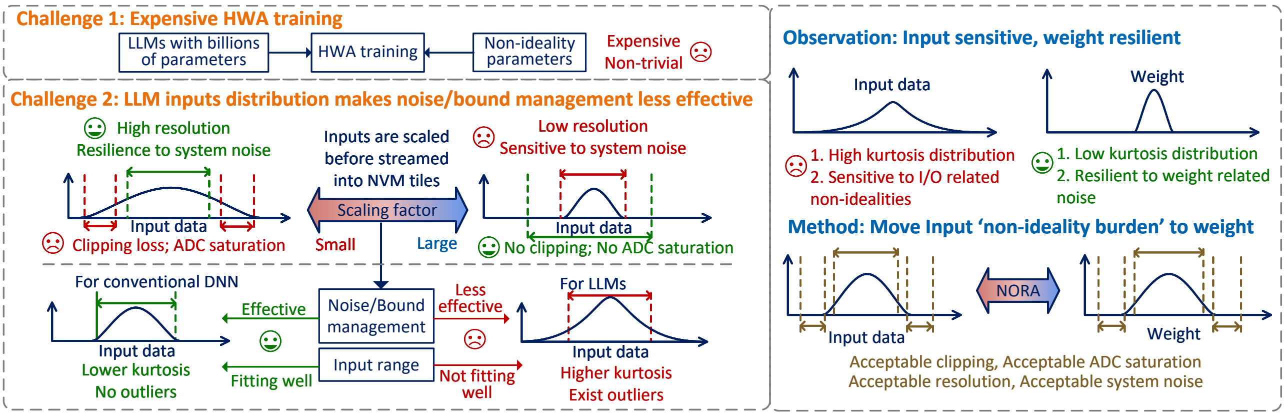

Challenge 1: Expensive HWA training
LLMs with billions of parameters
HWA training
Non-ideality parameters
Expensive
Non-trivial
Challenge 2: LLM inputs distribution makes noise/bound management less effective
High resolution
Resilience to system noise
Input data
Clipping loss; ADC saturation
Inputs are scaled before streamed into NVM tiles
Scaling factor
Small
Large
Low resolution
Sensitive to system noise
Input data
No clipping; No ADC saturation
For conventional DNN
Effective
Noise/Bound management
Less effective
For LLMs
Input data
Lower kurtosis
No outliers
Fitting well
Input range
Not fitting well
Input data
Higher kurtosis
Exist outliers
Observation: Input sensitive, weight resilient
Input data
Weight
1. High kurtosis distribution
2. Sensitive to I/O related non-idealities
1. Low kurtosis distribution
2. Resilient to weight related noise
Method: Move Input 'non-ideality burden' to weight
NORA
Input data
Weight
Acceptable clipping, Acceptable ADC saturation
Acceptable resolution, Acceptable system noise

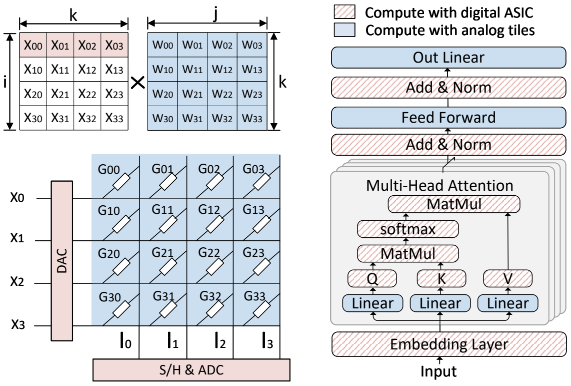
Fig. 1: Overview of our method. Challenge 1: HWA training in other works is expensive and non-trivial for LLMs; Challenge 2: Several trade-off solutions in conventional DNNs do not work well in LLMs. Our observation is that LLMs are sensitive to input/output related noise but resilient to weight related noise. Based on this observation, we propose NORA, a noise-optimized rescaling method for analog CIM, which can move the 'accuracy burden' on I/O to weight.

and LLaMA [30]. Specifically, we perform a comprehensive sensitivity study of several LLMs on major Analog CIM non-idealities. Our findings suggest that LLMs are more sensitive to input/output noise while resilient to weight noise. Then, our proposed noise-optimized rescaling method can move the "non-ideality burden" from dynamically streamed data (input and output) to statically mapped data (weights) and achieve nearly lossless performance.

II. Background

Analog compute-in-memory (CIM) exploits Ohm's law and Kirchhoff's law to execute general matrix-vector multiplication (GEMV) by accumulating the output current of non-volatile memory (NVM) devices, such as resistive random-access memory (ReRAM) and phase-change memory (PCM). As shown in Fig. 2, weights are programmed as the conductance of NVM cells using a write-verify memory programming process. During GEMV computation, input vectors are converted into analog signals or bit streams and flow into wordlines. The output current at the bitlines, which could be described as $I_{out} = V_{intput} \times G_{weight}$, are then quantized by analog-digital converters (ADC) and saved for further operations. The time complexity of GEMV decreases from $O(n)$ to $O(1)$ in Analog CIM. Though NVM cells could be reprogrammed, the programming of NVM devices is too expensive [17], [34]. Hence, for transformer models, the self-attention is deployed on digital tiles or digital cores [9], [18], [20]. As shown in Figure 2, the linear layers are programmed on PCM tiles, and the normalization, activation function, and self-attention are all deployed on digital compute units.

Though LLM inference could benefit from the performance speedup and energy efficiency of GEMV in Analog CIM accelerators, there are challenges in retaining the model accuracy due to the diverse non-idealities of NVM devices. Non-idealities modeled by us are listed in Table I. According to the position where non-idealities exist, we divide them into two types: tile non-idealities and IO non-idealities. IO non-idealities mostly appear in the I/O interface, such as A/D converters. They could lead to large accuracy degradation and are hard to compensate. Tile non-idealities are non-idealities that appear on analog tiles and could have less influence on LLM performance or could be easily compensated by applying some simple techniques.



(a) NVM tile (b) Transformer

Fig. 2: (a) NVM-based CIM tile, which consists of CIM devices and A/D interface; (b) Transformer structure. Shadow part are executed on analog tiles, others are executed on digital units.

TABLE I: Major I/O and tile non-idealities modeled

Category	Noise	Type
IO non-idealities	ADC noise	Quantization noise
	DAC noise	Quantization noise
	Additive output noise	System Gaussian noise
	Additive input noise	System Gaussian noise
	S-shape nonlinearity	Device Nonlinearity
Tile non-idealities	Programming noise	Weight fabrication non-ideality
	Short-term Read noise	Cyle-by-cycle read variance
	IR-drop	Wire resistance non-ideality

A. *IO Non-idealities*

IO non-idealities appear when activations pass through the tiles. The input and output interface of the NVM tiles has A/D converters that are expensive to keep in high precision. Hence, ADC and DAC precision are typically set to less than 9 bits, which means input would be quantized before GEMV and output would be quantized after GEMV. In addition, mix-signal devices also introduce Gaussian noise, which could be modeled as additive noise. Typically, before streamed into DACs, the input data is scaled by a linear factor α as $\mathbf{x}/\alpha$ in previous work [7], [28]. As shown in Figure 1 (left center), smaller α ensures data resolution during quantization and provides a higher signal-to-noise ratio (SNR) but could lead to clipping of input outliers and output saturation. On the contrary, a larger α ensures minimum input clipping and mitigates output saturation but decreases the data resolution and SNR. Previous work [14] has proposed noise management and bound management to balance this trade-off. By dynamically adjusting linear factor α, noise management targets the balance between avoiding input clipping and maximum resolution, while bound management targets the balance between fixed input range and output clipping (ADC saturation). These complex dynamic adjustment techniques present good accuracy calibration in conventional models evaluated in previous work [14], [28]. However, bound and noise management become less effective in LLMs whose inputs have a large kurtosis caused by outliers. Even though an optimal α is achieved, the data resolution loss and clipping loss are both unacceptable. Hence, applying optimization for LLMs to retain the model accuracy during inference is crucial.

B. *Tile Non-idealities*

Take PCM as an example. PCM weight mapping is based on material properties that allow PCM cells to achieve intermediate stages between the fully amorphous state with high resistance and the fully crystalline state with low resistance [15]. Ideally, weights and PCM conductance follow Equation 1:

$$g_{ij} = g_{max} \frac{w_{ij}}{max(|\mathbf{w}_i|)} \tag{1}$$

However, in reality, PCM conductance has programming noise and drifts over time. The conductance read over time could be described as:

$$g_{read}(t_i) = g_0(t_i) + n_g(t_i) \tag{2}$$

t_i represents the time from programming to read, $g_0(t_i)$ represents the noise free conductance and $n_g(t_i)$ represents the long-term read noise (also known as $1/f$ read noise) [23]. In addition to long-term read noise, there also exists short-term read noise which is the variance when reading weight values for multiple times cycle-by-cycle. Moreover, the voltage along one column (i.e., bitline) can drop along the wire due to IR-drop from wire resistance. According to previous research, programming noise, read noises, weight drift and IR-drop are less significant in transformer models [28]. Furthermore, IR-drop and drift could be simply compensated [28].

III. Motivation

Given the diverse non-ideality types inherent to Analog CIM hardware, we perform a systematic sensitivity study to examine how different noise sources impact LLM accuracy. Unlike traditional DNNs, where prior research has provided insight into their sensitivity to analog-induced noise [28], LLMs introduce new challenges due to their more complex distributions, particularly in activations, which often contain outliers. LLMs, such as OPT [37], LLaMA [30], and Mistral [10], exhibit distinct characteristics – such as activation outliers – that set them apart from conventional DNN models, necessitating a different approach when deploying them on the NVM tiles in Analog CIM accelerators.

A. *Sensitivity Study on Analog CIM-induced Noises for LLMs*

In this study, we analyze the unique properties of LLM weight and activation distributions in the context of analog CIM-induced noises. To the best of our knowledge, this is the first systematic investigation of LLM sensitivity to such hardware-induced noise. Inspired by earlier work on DNNs [28], we apply various non-idealities at different levels to characterize their impact on LLM accuracy, as shown in Figure 3. Our results reveal key insights into how LLMs behave under analog noise conditions.

1) Observation on IO non-idealities: As shown in Figure 3 (a) and (b). We observe that OPT models are very sensitive to A/D quantization noise. Whereas, Mistral and LLaMA models show better robustness to A/D conversion noise compared with OPT models. We discuss this in more detail on this phenomenon in Section V. In addition, as shown in Figure 3 (c) and (d), compared with other non-idealities, additive systematic Gaussian noises have the most significant influence on all the models' performance. Though Mistral and LLaMA models are more robust against additive noises, the accuracy drops are still unacceptable.

2) Observation on tile non-idealities and S-shape non-linearity: As shown in Figure 3 (e), (f), (g) and (h), most models exhibit good robustness to IR-drop, short-term read noise, S-shape non-linearity, and programming noise (i.e., tile non-idealities and S-shape non-linearity). There is nearly no accuracy drop when scaling up those non-idealities.

B. *Optimization Target*

Based on our observation in Section III-A, we can break the challenges down into more specific optimization targets. First, as discussed in Section II-A, A/D quantization noise could be caused both by the clipping of outliers and the low resolution of small values. Many previous works [4], [33], [36] have proved that tightly distributed data (i.e., with lower kurtosis) is more friendly to quantization compared with long-tail distributed data (i.e., with higher kurtosis). This is because clipping could be mitigated and the resolution could be maximized. Unfortunately, as the example shown in Figure 4, most LLMs' activation distribution has high kurtosis. Hence, for LLMs that are quantization sensitive like OPT models, **the first target** is to tighten the activation distribution.

Second, for systematic Gaussian noise, which has the most significant influence on LLMs' performance, improving the SNR of the system could highly improve the data quality. Tightening input distribution can boost the signal strength and alleviate the influence of output noise, as discussed in previous work [14]. In addition, increasing the signal magnitude is also a prevalent method [7] to improve SNR. Therefore, **the second target** is when tightening the input distribution, at the same time, maximizing the achievable signal magnitude

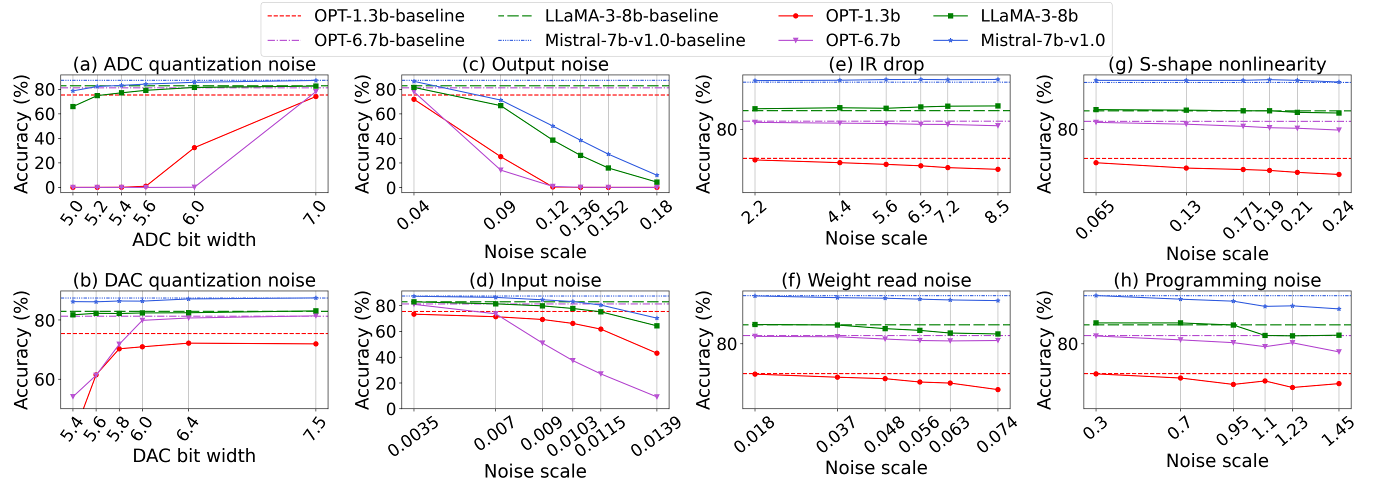


Fig. 3: Accuracy drop according to non-idealities added on LLMs. Each noise scale on the x-axis starts with a level causing 0.0001∼0.0002 MSE and ends with causing 0.0027∼0.0028 MSE compared with ideal situation on a 4096 × 4096 feature map. Under the same noise level, LLMs are all sensitive to output additive noise. OPT models are more sensitive to A/D quantization noise. All models are robust to short-term read noise, programming noise, S-shape non-linearity and IR-drop.

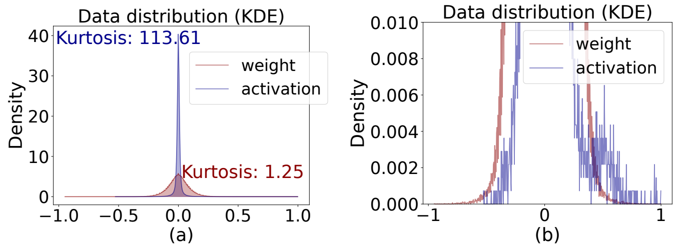


Fig. 4: (a) The normalized activation and query weight kernel density estimation (KDE) of layer 2 in Mistral-7B-v1.0 model. The kurtosis of activation is 113.61, while the kurtosis of weight is only 1.25. (b) Zoom in data distribution between the density of [0,0.01]. It is clear that on the right side of this figure, activation has several outliers that lead to a long-tail distribution.

(input/output) without leading to unacceptable input/output clipping.

IV. Method

As discussed in Section III, we aim to tighten the distribution of input and maximize the achievable signal magnitudes. Based on our observation that LLMs are sensitive to several IO non-idealities (A/D quantization noise and additive Gaussian noise) but resilient to tile non-idealities, it's easy to think of moving some "non-ideality burden" from activation quantization to weight mapping. Inspired by a low-precision quantization method [33], we could tighten input data by adding an additional component into scaling factors of Analog CIM that are applied to the input matrix before DAC conversion and weight matrix before programming. We call our method NORA – **N**oise-**O**ptimized **R**escaling method on LLM weights and activations for **A**nalog CIM accelerators.

Specifically, we can start with the original GEMV operation of vector $\mathbf{x}_i$ (in matrix $\mathbf{X}$) and matrix $\mathbf{W}$ on analog tiles (take Figure 2 top-left as an example) could be described as follows:

$$y_{ij} = \alpha_i \gamma_j f_{adc}\left(\sum_k (\tilde{w}_{kj} \cdot \tilde{x}_{ik}) + \sigma_{out}\xi_{ij}\right) + \beta_{ij} \tag{3}$$

Where,

$$\tilde{w}_{kj} = f_{map}\left(\frac{w_{kj}}{\gamma_j}\right) + \sigma_w \xi_{kj} \qquad \gamma_j = \frac{max(|\mathbf{w}_j|)}{g_{max}} \tag{4}$$

$$\tilde{x}_{ik} = f_{dac}\left(\frac{x_{ik}}{\alpha_i}\right) + \sigma_{in}\xi_{ik} \qquad \alpha_i = max(|\mathbf{x}_i|) \tag{5}$$

$\mathbf{w}_j$ ($\mathbf{w}_k$) means each column (row) in $\mathbf{W}$. $\mathbf{x}_i$ ($\mathbf{x}_k$) means each row (column) in $\mathbf{X}$. $\tilde{w}_{kj}$, $\tilde{x}_{ik}$ refer to analog weight and input respectively. f_{xxx} are A/D conversion and mapping functions and σ is the variance of additive Gaussian noise. $\mathbf{x}_i$ is scaled by α_i before steamed into DAC and $\mathbf{w}_j$ is scaled by γ_j before programming. α_i and γ_j are scaling factors in Analog CIM. After finishing the GEMV and getting the digital outputs from ADCs, the outputs are scaled back by $\alpha_i\gamma_j$. In our method, we introduce a component s_k into the scaling factors γ_j and α_i when mapping weights and activations to analog tiles, as shown below:

$$\tilde{w}_{kj} = f_{map}\left(\frac{w_{kj}s_k}{\gamma'_j}\right) + \sigma_w \xi_{kj} \qquad \gamma'_j = \frac{max(|\mathbf{w}_j \odot \mathbf{s}|)}{g_{max}} \tag{6}$$

$$\tilde{x}_{ik} = f_{dac}\left(\frac{x_{ik}}{\alpha'_i s_k}\right) + \sigma_{in}\xi_{ik} \qquad \alpha'_i = max(|\mathbf{x}_i \oslash \mathbf{s}|) \tag{7}$$

$\mathbf{s}$ is vector which has the dimension same as $\mathbf{x}_i$ or $\mathbf{w}_j$, s_k is a component in vector $\mathbf{s}$. s_k varies among values in vector $\mathbf{x}_i$ and $\mathbf{w}_j$ while keeps the same among values in vector $\mathbf{x}_k$ and $\mathbf{w}_k$. By adjusting s_k, we could balance the values between vector $\mathbf{x}_i$ and vector $\mathbf{w}_j$ (For instance, for large x_{ik}, s_k could be larger). To better determine each s_k in the $\mathbf{s}$ vector, we follow the method in previous work [33] and determine s_k by $s_k = max(|\mathbf{x}_k|)^{\lambda} / max(|\mathbf{w}_k|)^{1-\lambda}$. According to previous research [4], [33], outliers in LLM activation tend to appear in some specific channels ($\mathbf{x}_k$) regardless of the input data. Hence, this component could be calculated by a small calibration dataset offline and used for all tasks [33]. After applying the component s_k, the input data distribution is tightened by slightly sacrificing the tightness of weights.

In addition, by applying s_k, the scaling factors for outputs after ADC conversion become:

$$\alpha_i'\gamma_j' = max(|\mathbf{x}_i \odot \mathbf{s}|)\frac{max(|\mathbf{w}_j \oslash \mathbf{s}|)}{g_{max}} \tag{8}$$

As shown in Equation 3, smaller $\alpha_i\gamma_j$ indicates larger output current of a single CIM bitline (the elements in the $f_{adc}()$ of Equation 3). Hence, if the scaling factors $\alpha_i'\gamma_j'$ after applying our method are smaller than the original scaling factors $\alpha_i\gamma_j$, it indicates that the output current which is going to be streamed into ADC has been increased. As discussed in previous work [14], most system Gaussian noises come from mixed-signal or analog components in ADCs. If we could increase the output current, the influence of output additive Gaussian noise could be alleviated. Fortunately, according to our experiments, which will be discussed in Section V-C, our re-scaling method can achieve smaller scaling factors in most LLMs.

V. Evaluation

We choose models: OPT with 1.3B, 2.7B, 6.7B and 13B parameters [37], LLaMA 2 with 7B parameters [30], LLaMA 3 with 8B parameters [1] and Mistral v1.0 with 7B parameters [10] for evaluation. We use the Pile [6] dataset as the calibration set to find our additional components s_k. We use word prediction task with Lambada dataset [24] to evaluate the performance of models. We build up our evaluation using PyTorch framework [25], as well as HuggingFace Transformers [31] and Datasets [16] package.

TABLE II: AIHWKIT modeling settings.

Noise	Value
in_res (ADC quantization steps)	7bit (128)
out_res (DAC quantization steps)	7bit (128)
out_noise (system additive noise deviation)	0.04
ir_drop (noise scale)	1.0
w_noise (short-term weight noise)	0.0175
tile_size	512×512

We use the analog in-memory hardware acceleration kit (AIHWKIT) [27] to evaluate model accuracy on Analog CIM tiles. We convert all nn.Linear layers of models into AnalogLinear layers defined in AIHWKIT while keeping the normalization, activation function, and self-attention operation in the original digital format. Following previous work [14], [28], we list our settings of AIHWKIT in Table II. All other simulation parameters are set as default. Normalization, activation functions, and self-attention are executed on digital units with full precision.

A. Overall Model Accuracy

Applying the setting in Table II, we get the following results:

a) Results for OPT models: As shown in Figure 5 (a), naively deploying the linear layers onto analog tiles could lead to catastrophic accuracy degradation. OPT-2.7b could even have an over 40% accuracy drop. After applying our method, the accuracy of models can be effectively retained. For OPT-6.7b and OPT-13b, we can achieve less than 1% accuracy loss.

b) Results for other LLMs: The performance of our method on LLaMA and Mistral models is shown in Table III. Our method can also achieve good accuracy on other LLMs besides OPT models: LLaMA-3-8B and LLaMA-2-7B can achieve less than 1.6% accuracy loss; and Mistral-7B-v1.0 can even achieve less than 1% accuracy loss. Overall, our method can effectively retain the accuracy of all tested models in Analog CIM.

TABLE III: NORA accuracy for LLaMA and Mistral models

Model	Setting	Lambada acc(%)
LLaMA-2 7B [30]	Our method	87.99
	Digital Full precision	89.04
LLaMA-3 8B [1]	Our method	81.33
	Digital Full precision	82.92
Mistral 7B v1.0 [10]	Our method	86.55
	Digital Full precision	87.41

B. Analysis on Noise Optimization

To analyze the ability of our method to mitigate different non-idealities, We scale up each non-ideality to the same level (i.e., cause the same mean square error) independently with other non-idealities set into the ideal situation (i.e., same as digital full-precision) and evaluate the performance of naive Analog CIM and our method. The results are shown in Figure 5 (b) and (c). From the results, we can find that:

First, for models sensitive to quantization noise (like OPT models), our method can effectively mitigate the accuracy drop due to both DAC and ADC quantization. Our method can recover nearly 10% accuracy drop caused by DAC quantization and recover nearly 75% accuracy drop caused by ADC quantization in model OPT-6.7B, which means our method can not only avoid input clipping, but also effectively mitigate output ADC saturation.

Second, for additive Gaussian (input/output) noises that significantly impact accuracy of all LLMs, our method also presents good improvements. For more sensitive OPT models, we can recover 60% to 70% accuracy drop caused by additive output noise and 5% to 60% accuracy drop caused by additive input noise. For models that are more resilient, e.g., LLaMA and Mistral, to additive noise, i.e., models do not show large accuracy drop due to additive noise, we can still present significant accuracy recovery. This indicates that our method can increase SNR in Analog CIM systems.

C. Analysis on Data Distribution and output current

To better support our analysis in Section V-B, we calculate the kurtosis of inputs and weights of each layer in LLMs. As shown in Figure 6 (a) and (b), for OPT-6.7B, which is sensitive to quantization noise, the kurtosis of the input data decreases greatly with a slight increase in weight kurtosis after applying our method. For quantization-resilient models like LLaMA and Mistral, our method could still greatly decrease the input kurtosis for the first several layers. Furthermore, as discussed in Section IV, smaller $\alpha_i\gamma_j$ indicates larger output current streamed into ADC. As shown in Figure 6 (c), after applying our method, $\alpha_i\gamma_j g_{max}$ is reduced (g_{max} is constant and multiplied to the left only for evaluation convenience), providing a larger output current and a higher SNR.

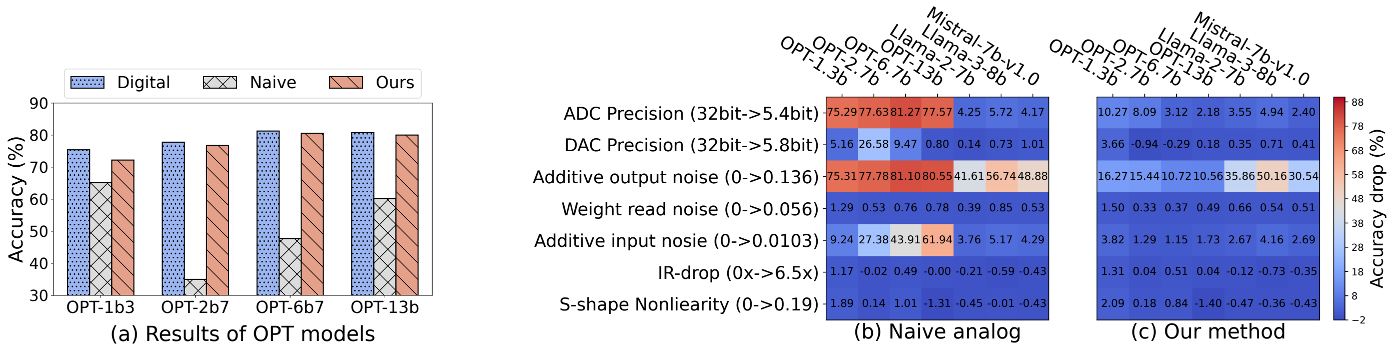


Fig. 5: (a) Accuracy of OPT models [37] on Lambada dataset [24] under: Digital full precision (NVIDIA H100), Naive analog settings (Table II) and our method. (b) Noise mitigation of our method when setting all noise under the situation: for a 4096× 4096 analog tile, the noise could cause a mean square error (MSE) between 0.0015 and 0.0016 compared with ideal results.

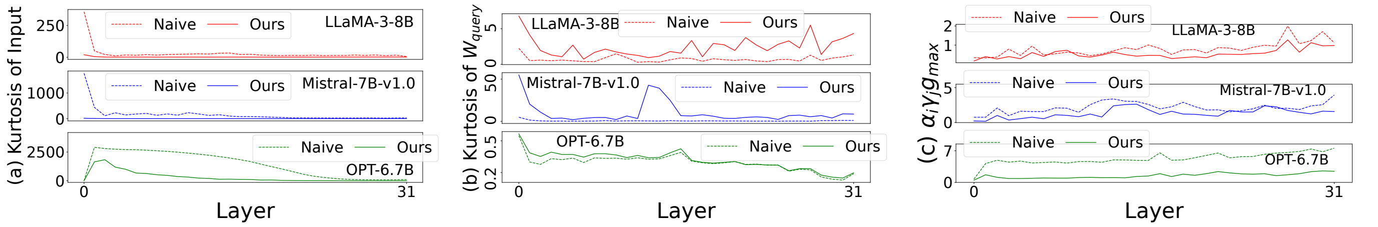


Fig. 6: Input (a) and query weight (b) kurtosis of each layer in OPT-6.7B, LLaMA-3-8B and Mistral-7B-v1.0. By applying NORA, the kurtosis of input data decreases largely with slightly increasing the weight kurtosis. For quantization resilient models, out method has more significant influence in first few layers. (c) Decreasing of scaling factors, i.e., $\alpha_i \gamma_j \cdot g_{max}$, indicates the increase of output current which provides a higher SNR. Here, g_{max} is constant and multiplied to left only for evaluation convenience.

VI. Related Work

Early work on Analog CIM accelerators mostly focus on the precision of weight-programming [3], [22], [23], which provides a mutual understanding of Analog CIM working mechanism and robust on-tile data storage. By estimating different model architectures on NVM tiles with diverse non-idealities, some studies [7], [11], [14], [28] have found that those I/O related non-idealities are also significant to accuracy drop. Hence, hardware-aware training and compensation techniques have been studied to mitigate the influence of all types of non-idealities in those works. However, the complex and expensive hardware-aware training, which is non-trivial for LLMs, is needed in all those previous works.

On LLM quantization, several methods focus on LLM quantization for digital cores. LLM.int8() [4] processes outliers separately in higher precision, however, it needs extra architecture design to support mixed-signal activation in Analog CIM tiles. SmoothQuant [33] proposed an offline method to move the quantization overhead from activation to weights by multiplying the scaling matrix on activation and weight. However, those methods are limited to LLM quantization on digital computing such as GPUs without considering the challenges and properties of Analog CIM.

VII. Conclusion

Despite the potentials of Analog Compute-in-Memory (CIM) accelerators, serving Large Language Models (LLMs) on analog CIM faces severe accuracy loss from the non-idealities. In this paper, we first perform a sensitivity study to evaluate the accuracy of several LLMs in analog CIM, and we find that LLMs are sensitive to input/output-related non-idealities while generally resilient to weight-related non-idealities. Based on this finding, we propose a noise-optimized rescaling method on LLM weights and activations for Analog CIM accelerators (NORA), which introduces an additional component to the scaling factors of Analog CIM. Our method can achieve less than 1% accuracy loss in our evaluated LLM models.

Limitations and Future Work: We also evaluated our method after drifting the weights for 1 hour and find that our method become less significant in some models. Although 1 hour is far more than the time needed to inference one batch, the influence of drifting and long-term reading noise should be considered in future work. It is worth noting that though our results and analysis are based on PCM, this method can also been extended to other NVM devices such as ReRAM. Although some NVM devices cannot provide continuous analog weights, they can achieve over 8-bit weight precision by using multiple memory cells. In addition, due to our limited resources, we have no experiments on models larger than 13b and only evaluated limited tasks. In our future work, we will try to enrich our experiments by adding additional benchmarks. Furthermore, we plan to compare with more baselines and add more ablation studies, such as per-layer evaluation, or other quantization techniques. The evaluation of power, area, and latency is also considered an essential part in the future.

References

[1] AI@Meta, "Llama 3 model card," 2024. [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[2] T. Andrulis, J. S. Emer, and V. Sze, "Raella: Reforming the arithmetic for efficient, low-resolution, and low-loss analog pim: No retraining required!" in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–16.

[3] J. Büchel *et al.*, "Programming weights to analog in-memory computing cores by direct minimization of the matrix-vector multiplication error," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2023.

[4] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale," *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 318–30 332, 2022.

[5] A. Dubey *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[6] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy, "The Pile: An 800gb dataset of diverse text for language modeling," *arXiv preprint arXiv:2101.00027*, 2020.

[7] T. Gokmen, M. Onen, and W. Haensch, "Training deep convolutional neural networks with resistive cross-point devices," *Frontiers in neuroscience*, vol. 11, p. 538, 2017.

[8] M. Imani, S. Gupta, Y. Kim, and T. Rosing, "Floatpim: In-memory acceleration of deep neural network training with high precision," in *Proceedings of the 46th International Symposium on Computer Architecture*, 2019, pp. 802–815.

[9] S. Jain *et al.*, "A heterogeneous and programmable compute-in-memory accelerator architecture for analog-ai using dense 2-d mesh," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 1, pp. 114–127, 2022.

[10] A. Q. Jiang *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[11] V. Joshi, M. Le Gallo, S. Haefeli, I. Boybat, S. R. Nandakumar, C. Piveteau, M. Dazzi, B. Rajendran, A. Sebastian, and E. Eleftheriou, "Accurate deep neural network inference using computational phase-change memory," *Nature communications*, vol. 11, no. 1, p. 2473, 2020.

[12] R. Khaddam-Aljameh, M. Stanisavljevic, J. F. Mas, G. Karunaratne, M. Brändli, F. Liu, A. Singh, S. M. Müller, U. Egger, A. Petropoulos *et al.*, "Hermes-core—a 1.59-tops/mm 2 pcm on 14-nm cmos in-memory compute core using 300-ps/lsb linearized cco-based adcs," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 4, pp. 1027–1038, 2022.

[13] M. Le Gallo *et al.*, "A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference," *Nature Electronics*, vol. 6, no. 9, pp. 680–693, 2023.

[14] M. Le Gallo, C. Lammie *et al.*, "Using the ibm analog in-memory hardware acceleration kit for neural network training and inference," *APL Machine Learning*, vol. 1, no. 4, 2023.

[15] M. Le Gallo and A. Sebastian, "An overview of phase-change memory device physics," *Journal of Physics D: Applied Physics*, vol. 53, no. 21, p. 213002, 2020.

[16] Q. Lhoest *et al.*, "Datasets: A community library for natural language processing," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 175–184. [Online]. Available: https://aclanthology.org/2021.emnlp-demo.21

[17] H. Li, H. Jin, L. Zheng, X. Liao, Y. Huang, C. Liu, J. Xu, Z. Duan, D. Chen, and C. Gui, "Cpsaa: Accelerating sparse attention using crossbar-based processing-in-memory architecture," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

[18] H. Li, Z. Li, Z. Bai, and T. Mitra, "Asadi: Accelerating sparse attention using diagonal-based in-situ computing," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2024, pp. 774–787.

[19] W. Li, P. Xu, Y. Zhao, H. Li, Y. Xie, and Y. Lin, "Timely: Pushing data movements and interfaces in pim accelerators towards local and in time domain," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2020, pp. 832–845.

[20] S. Liu, C. Mu, H. Jiang, Y. Wang, J. Zhang, F. Lin, K. Zhou, Q. Liu, and C. Chen, "Hardsea: Hybrid analog-reram clustering and digital-sram in-memory computing accelerator for dynamic sparse self-attention in transformer," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2023.

[21] C. Mackin *et al.*, "Optimised weight programming for analogue memory-based deep neural networks," *Nature communications*, vol. 13, no. 1, p. 3765, 2022.

[22] M. Martemucci, B. Kersting, R. Khaddam-Aljameh, I. Boybat, S. Nandakumar, U. Egger, M. Brightsky, R. L. Bruce, M. Le Gallo, and A. Sebastian, "Accurate weight mapping in a multi-memristive synaptic unit," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2021, pp. 1–5.

[23] S. Nandakumar *et al.*, "Precision of synaptic weights programmed in phase-change memory devices for deep learning inference," in *2020 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2020, pp. 29–4.

[24] D. Paperno *et al.*, "The LAMBADA dataset: Word prediction requiring a broad discourse context," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 1525–1534. [Online]. Available: http://www.aclweb.org/anthology/P16-1144

[25] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," 2019. [Online]. Available: https://arxiv.org/abs/1912.01703

[26] X. Peng, S. Huang, Y. Luo, X. Sun, and S. Yu, "Dnn+ neurosim: An end-to-end benchmarking framework for compute-in-memory accelerators with versatile device technologies," in *2019 IEEE international electron devices meeting (IEDM)*. IEEE, 2019, pp. 32–5.

[27] M. J. Rasch *et al.*, "A flexible and fast pytorch toolkit for simulating training and inference on analog crossbar arrays," in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2021, pp. 1–4.

[28] M. J. Rasch, C. Mackin *et al.*, "Hardware-aware training for large-scale and diverse deep learning inference workloads using in-memory computing-based accelerators," *Nature communications*, vol. 14, no. 1, p. 5282, 2023.

[29] A. Shafiee *et al.*, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 14–26, 2016.

[30] H. Touvron *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[31] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[32] C. Wolters, X. Yang, U. Schlichtmann, and T. Suzumura, "Memory is all you need: An overview of compute-in-memory architectures for accelerating large language model inference," *arXiv preprint arXiv:2406.08413*, 2024.

[33] 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 *International Conference on Machine Learning*. PMLR, 2023, pp. 38 087–38 099.

[34] X. Yang, B. Yan, H. Li, and Y. Chen, "Retransformer: Reram-based processing-in-memory architecture for transformer acceleration," in *Proceedings of the 39th International Conference on Computer-Aided Design*, 2020, pp. 1–9.

[35] A. Yazdanbakhsh, A. Moradifirouzabadi, Z. Li, and M. Kang, "Sparse attention acceleration with synergistic in-memory pruning and on-chip recomputation," in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2022, pp. 744–762.

[36] H. Yu, T. Wen, G. Cheng, J. Sun, Q. Han, and J. Shi, "Low-bit quantization needs good distribution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 680–681.

[37] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.