Where to cut: Efficient ADC quantization for analog in-memory computing with discrete values

Johannes Leugering*, Shashank Bansal*, Zhaoyi Liu†, Jiahao Song*, Bouchaib Cherif‡, Gert Cauwenberghs*
*Dept. of Bioengineering, † Dept. of Electrical and Computer Engineering
University of California, San Diego, La Jolla, CA, USA
‡ Northrop Grumman, Falls Church, VA, US
Email: *†{jleugering, shbansal, zhl101, jis090, gcauwenberghs}@ucsd.edu, ‡bouchaib.cherif@ngc.com

***Abstract*—Many proposed in-memory-computing systems use analog memristive crossbars to compute matrix-vector products over discrete domains. This yields analog outputs distributed around discrete values across a wide nominal range. Lossless quantization of this range requires costly high-precision analog-to-digital converters (ADCs), which limits the applicability of this approach. But typical results are highly concentrated in a small central region; hence, an ADC with lower resolution that only operates in this central region can achieve almost full accuracy at a fraction of the cost. In this paper, we explore how to appropriately choose ADC resolution and the covered region of interest, specifically for low-precision applications in approximate in-memory-computing. Our results reveal two distinct strategies: ADCs with sufficient resolution should (at least) capture the region of interest without loss, whereas lower-resolution ADCs should space their levels just enough to cover the region of interest. We argue that using this scheme could drastically improve power efficiency and thus scalability of compute-in-memory architectures.**

***Index Terms*—ADC, in-memory-computing, quantization**

I. Introduction

Matrix-vector-multiplications (MVMs) are a crucial operation for many compute-intensive tasks like Deep Learning or physics simulations. Analog compute-in-memory (CIM) approaches promise to reduce power consumption by performing such MVM operations within the memory itself, e.g. by storing a (stationary) matrix W in a memristive crossbar [1], [2], where each element is programmed into the conductance of a corresponding nonvolatile memory cell (see Fig. 1(a)). The (discrete) input vectors x are then applied column-wise, e.g. encoded into distinct driving voltages, and the results are read out row-wise as the (analog) output vector $\tilde{y}$, e.g. in the current domain. Typically, these outputs then need to be digitized by analog-to-digital converters (ADC) for post-processing or routing. In such a mixed-signal system, the conversion to and from analog signals is typically a major energy sink [3]; thus minimizing the ADC power is crucial.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) program Massive Cross-Correlation (MAX) through Air Force Research Laboratory Contract No. FA8650-23-C-7306. The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

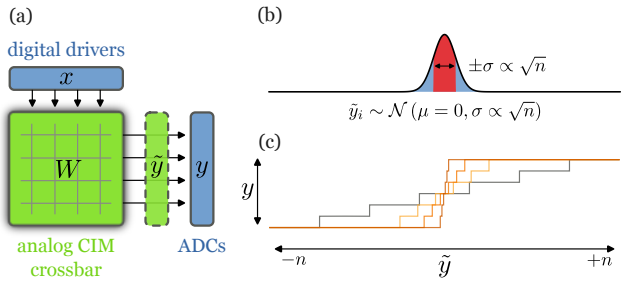

Fig. 1. (a) An example of an analog compute-in-memory system that multiplies the (bipolar) vector x with the (bipolar) matrix W. The analog output $\tilde{y}$ is quantized to yield the discrete result y. (b) The central limit theorem states that $\tilde{y}$ is approximately normally distributed with standard deviation $\propto \sqrt{n}$. (c) The ADC has evenly distributed quantization levels. The ADC can cover the entire domain of $\tilde{y}$ (gray line) or only a sub-region (colored lines).

Naively, one could design the ADC for loss-less conversion across the entire domain of $\tilde{y}$, i.e. each level y of the ADC corresponds to exactly one (noise-free) output $\tilde{y}$ of the crossbar. This requires an ADC resolution of least $\log_2(N)$ bit, where N is the vector length. But in many applications, for example in Deep Learning, this naive approach has proven to be inefficient, because a certain degree of errors is acceptable, and the values of $\tilde{y}$ are non-uniformly distributed and tightly clustered around a mean value. Several recent studies have shown more efficient solutions by placing the ADC's quantization levels in a smaller central region of interest where most of the probability mass resides. These studies have proposed different heuristic or data-driven approaches to determine the right size of this region of interest, e.g. using a range of ± 60 [4] or ± 120 [5] for a domain of 256 or 512 values, respectively; using a range of $\pm 3\sigma_{\tilde{Y}}$ [6] or $\pm 4\sigma_{\tilde{Y}}$ [7] that scales with the distribution's standard deviation; or optimizing the size of the ADC's region as a hyperparameter during DNNs training [8]–[10]; or using a stochastic encoding [11]. See [7] for a detailed discussion of quantization in CIM architectures and references therein.

In this paper, we address the question of how to optimally chose the region of interest systematically from an information

theoretical perspective and using numerical simulations. The results provide guidance to CIM circuit and system designers for improving the energy-efficiency of MVM quantization.

II. Information Theory Background

We assume an input $\tilde{Y}$ with (discrete) probability mass function $p_{\tilde{Y}}$. We use the lower-case symbols y and $\tilde{y}$ for concrete samples of the random variables Y and $\tilde{Y}$. An ADC with resolution ρ bit produces the discrete output $y \in \{0, \dots, 2^\rho - 1\}$ if its input (with input-referred noise ε) falls within the corresponding thresholds $\theta_y \le \tilde{y} + \varepsilon \le \theta_{y+1}$. Its quantization function Q can thus be written as a step-function:

$$y = Q(\tilde{y} + \varepsilon) = \sum_{y=0}^{2^\rho - 1} \begin{cases} y & \text{if } \theta_y \le \tilde{y} + \varepsilon < \theta_{y+1} \\ 0 & \text{otherwise} \end{cases}$$

For a given input $\tilde{y}$ and thresholds θ, we get y with probability

$$p_{Y|\tilde{Y}}(y|\tilde{y}, \theta) = P(Q(\tilde{y} + \varepsilon) = y) = \int_{\theta_y}^{\theta_{y+1}} p_\varepsilon(\varepsilon - \tilde{y}) d\varepsilon.$$

The total probability of observing the output y is then

$$p_Y(y|\theta) = \sum p_{\tilde{Y}}(\tilde{y}) p_{Y|\tilde{Y},\theta}(y|\tilde{y}, \theta).$$

Information Bottleneck Principle

We can assess the accuracy of the ADC by the mutual information $I(Y;\tilde{Y})$ between its noise-free inputs $\tilde{Y}$ and its quantized output Y (see [12]):

$$\begin{aligned} \mathrm{I}(Y;\tilde{Y}) &= \sum_{y,\tilde{y}} p_{Y,\tilde{Y}}(y,\tilde{y}|\theta) \log_2 \left(\frac{p_{Y,\tilde{Y}}(y,\tilde{y}|\theta)}{p_Y(y|\theta) p_{\tilde{Y}}(\tilde{y}|\theta)} \right) \\ &= \sum_{\tilde{y}} p_{\tilde{Y}}(\tilde{y}) \sum_y p_{Y|\tilde{Y}}(y|\tilde{y}, \theta) \\ &\quad \cdot \Big(\log_2 p_{Y|\tilde{Y}}(y|\tilde{y}, \theta) - \log_2 p_Y(y|\theta) \Big) \end{aligned} \tag{1}$$

The entropy H, also called effective number of bits (ENOB), measures the amount of information encoded in a random variable, and imposes an upper limit on the mutual information:

$$\begin{aligned} H(\tilde{Y}) &= -\sum_{\tilde{y}} p_{\tilde{Y}}(\tilde{y}) \log_2 p_{\tilde{Y}}(\tilde{y}) \\ H(Y) &= -\sum_y p_Y(y|\theta) \log_2 p_Y(\tilde{y}|\theta) \end{aligned}$$

For the resolution ρ we have $I(Y;\tilde{Y}) \le \min(H(\tilde{Y}), \rho)$. Initially, we assume bounded (and hence negligible) noise for $\mathrm{Var}[\varepsilon] \to 0$. Then Y is just a deterministic function of $\tilde{Y}$ and we have the special case:

$$\begin{aligned} \mathrm{I}(Y;\tilde{Y}) &= H(Y|\theta) \\ \text{where } p_Y(y|\theta) &= \sum_{\tilde{y} \in \{\lceil \theta_s \rceil, \dots, \lfloor \theta_{s+1} \rfloor\}} p_{\tilde{Y}}(\tilde{y}) \end{aligned} \tag{2}$$

For a continuous distribution, the information bottleneck principle states that to maximize mutual information, the ADC's quantization function should approximate the (non-linear!) cumulative distribution function (CDF) of $\tilde{y}$. Here $\tilde{y}$ is discrete (albeit subject to noise) and we are looking for quantization with evenly-spaced thresholds, thus finding the optimal spacing α between thresholds θ becomes a combinatorial optimization problem.

Here, we assume MVM operations on vectors of length N, where the individual x_i and W_i are each independent and identically distributed with zero mean and standard deviations σ_x and σ_W, respectively. For large N, the product $\tilde{y}_i = \sum_{j=0}^{n} W_{ij} x_j$ approaches a normal distribution with standard deviation $\sigma_{\tilde{y}} = \sigma_x \sigma_W \sqrt{n}$ according to the central limit theorem (see Fig. 1(b)). Hence, the actual information content of $\tilde{Y}$ approaches

$$\mathrm{ENOB}(n) = \frac{1}{2} \log_2 \left(\frac{e\pi}{2} \sigma_{\tilde{y}}^2 n \right) \approx 1.05 + \log_2(\sigma_{\tilde{y}}) + \frac{1}{2} \log_2(n).$$

III. Choosing the optimal resolution

First, we'd like to choose a suitable resolution ρ. Following our information theoretic approach, we are primarily interested in two metrics: first in absolute terms, how closely the ADC's output comes to a loss-less encoding, i.e. how close its mutual information (MI) is to the ENOB of the input distribution; what loss is acceptable depends heavily on the application. Second, we are interested in the ADCs bit-efficiency, i.e. how many bits of MI it can achieve for the bits of its nominal resolution. As a concrete example, assume $x_i \in \{-1, 1\}$ and $W_{ij} \in \{-1, 1\}$ are i.i.d. uniform bipolar random variables and $n = 256$, ignoring noise for now ($\varepsilon = 0$). Then $b = 1$ and $\sigma_x = \sigma_W = \sigma_{\tilde{y}} = 1$, and thus $\mathrm{ENOB}(256) \approx 5.05$. We now compare the naive, loss-less quantization function that uniformly covers the entire domain of $\tilde{y}$, with a lower resolution alternative that places the thresholds in an optimally chosen central region of interest.

A. Full range quantization

Following the naive approach, we'd choose the resolution $\rho_{\text{unif}}(n) = b + \log_2(n) = 9$ to cover all 257 possible values of $\tilde{y}$ uniformly. Fig. 2(c) shows the quantization function Q of the naive solution overlaid on the CDF of $\tilde{Y}$. This one-to-one mapping results in the same normal distribution for Y as it does for $\tilde{Y}$ (see panel (f)). Hence,

$$I(Y;\tilde{Y}) = H(Y) = H(\tilde{Y}) \approx 5.05 \ll 9 = \rho_{\text{unif}}.$$

Here, we get a bit-efficiency of $\frac{5.05}{9} = 0.56$. Note that as N increases, this resolution grows twice as fast as the ENOB, so as $\lim_{n\to\infty} \frac{\mathrm{ENOB}(n)}{\rho_{\text{unif}}(n)} = \frac{1}{2}$. In other words, up to half the ADC's bits might be superfluous!

B. Region-of-interest quantization

Contrast this with an ADC with the much lower resolution $\rho_{\text{opt}} = 4$, but thresholds that are uniformly placed in a smaller "region of interest" – covering less than an eighth of the domain of $\hat{y}$. Here, the ADC thresholds are uniformly spaced around the center of the domain with a fixed increment α and offset β, resulting in $2^{\rho_{\text{opt}}} - 2$ uniformly spaced central

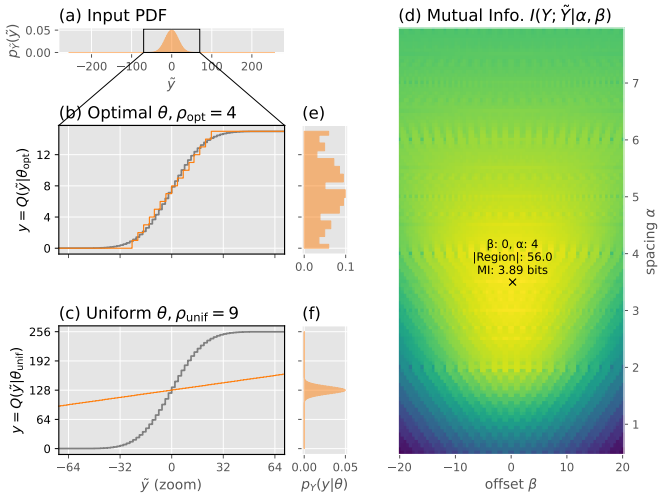

Fig. 2. Two quantizations for $N = 256$. (a) Distribution of the input $\tilde{Y}$. (b, c) Quantization functions for two ADC with resolutions $\rho_{\text{opt}} = 4$ (uniformly spaced in an optimally chosen range) and $\rho_{\text{unif}} = 9$ (lossless encoding of the entire domain of $\tilde{Y}$). (d) MI as a function of scale and offset parameters α, β. The optimal choice for ρ_{opt} is marked. (e,f) The corresponding output distributions for both ADCs.

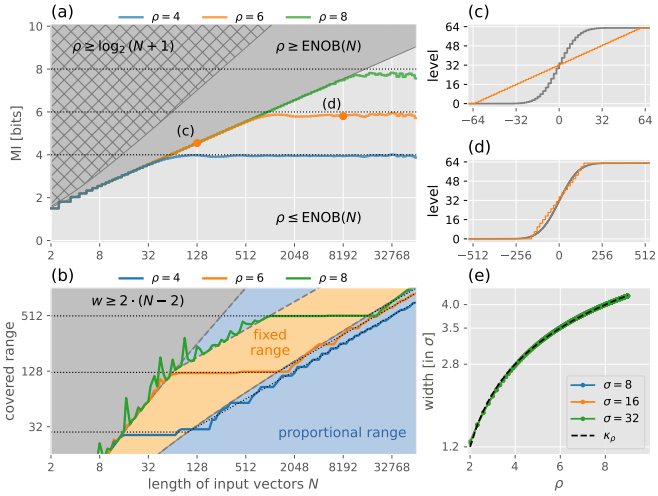

Fig. 3. Optimal ADC range as a function of vector length N. (a) MI for three ADCs with resolutions $\rho \in \{4, 6, 8\}$ (solid lines). (b) The covered range between smallest and largest threshold for the optimal parameters of each ρ as a function of N. (c, d) Optimal quantization functions for $\rho = 6$ and $N \in \{128, 8192\}$. (e) Proportionality constant κ_ρ as a function of ρ.

bins and two larger bins for the clipped extreme values. We use grid search to find the optimal parameters α and β to maximize the MI according to Eqn. (2). Fig. 2(d) shows the calculated mutual information for each combination of parameters. Since the possible values of $\tilde{y}$ are all even integers and hence spaced $\delta = 2$ apart, the optimal quantization with $\alpha = 4$ maps two consecutive values of $\tilde{y}$ to the same symbol y. Note that the discrete nature of the problem introduces many local optima that pose a challenge for optimization and defy a closed-form solution. Fig. 2(b) shows the resulting quantizer Q, which approximates the CDF in its central linear region of around ± 32, and clips outside that region. This step function closely follows the CDF, resulting in a much more uniformly distributed distribution of Y, and a mutual information of 3.89. At a nominal resolution of 4 bit, we thus get a much better bit-efficiency of $\frac{3.89}{4} = 0.97$.

IV. Choosing the Optimal Range

To better understand how the optimal range changes with ρ and $\sigma_{\tilde{y}} \propto \sqrt{N}$, we sweep N from 1 to 2^{16} and use the grid-search procedure explained above to compute the best achievable mutual information for three different ADC configurations with resolutions $\rho \in \{4, 6, 8\}$. The resulting MI closely follows the theoretical upper bound $\min(\text{ENOB}(N), \rho)$ in each case (see Fig. 3(a)).

As we vary N, the optimization procedure finds different optimal values of the scaling parameter α, which determines the range $w = \alpha(2^\rho - 2)$ between the ADC's smallest and largest threshold. Looking at the covered range in Fig. 3(b), the relationship between N and w is surprisingly complex. When possible, each ADC covers the entire domain of $\tilde{Y}$ one-to-one, so the covered range grows linearly with N. At some point, here empirically determined to be $N = 64$, the optimal range w begins to scale with the much lower rate $16\sigma_{\tilde{Y}} \propto \sqrt{N}$. The cause of this effect is not clear at present. Increasing N further, w hits the maximum range that the ADC's resolution permits without incurring losses. At this point, the values (in the center) of the distribution are uniquely represented by one corresponding level of the ADC. When N is increased beyond this point, the scaling parameter α and hence the range of the ADC remains constant, as long as w is larger than some critical value $\hat{w}$, which we formally define to be the (minimal) *region of interest*. As N is increased further, $\hat{w}$ and w grow as $w \approx \hat{w} = \kappa_\rho \sigma_{\tilde{y}} \propto \sqrt{N}$.

To determine the constant of proportionality κ_ρ, we consider the limiting case as $N \to \infty$, and the distribution of $\tilde{Y}$ approaches a continuous normal distribution. Then, $\hat{w}$ can be computed numerically for this normal distribution and a given resolution ρ. Normalizing the result by the distribution's standard deviation yields κ_ρ. κ_ρ is invariant to the standard deviation of the distribution, but it depends nonlinearly on ρ. Fig. 3(e) show κ as a function ρ. For example, for $N \to \infty$, the region of interest of an ADC with resolution $\rho = 8$ is $\hat{w} \approx 4$, which fits earlier observations [7], whereas for $\rho = 4$ we get $\hat{w} \approx 2.8$.

In summary, we see two distinct "strategies" to maximize MI: If the ADC's resolution ρ is large enough to uniquely resolve (at least) every possible value in the region of interest $\hat{w}$, then the increments α should be fixed accordingly. Conversely, if the ADC's resolution is lower, then α should be increased such that $w \geq \hat{w}$. For example, for $N = 256 \Rightarrow \sigma_{\tilde{Y}} = 16$ and $\rho = 6$ we get $\hat{w} = \kappa_\rho \sigma_{\tilde{Y}} \approx 3.5 \cdot 16 = 56$ from Fig. 3(e), which the ADC can (more than) cover with $\alpha = \delta = 2$; thus $w = \alpha(2^\rho - 2) = 124 \gg \hat{w}$.

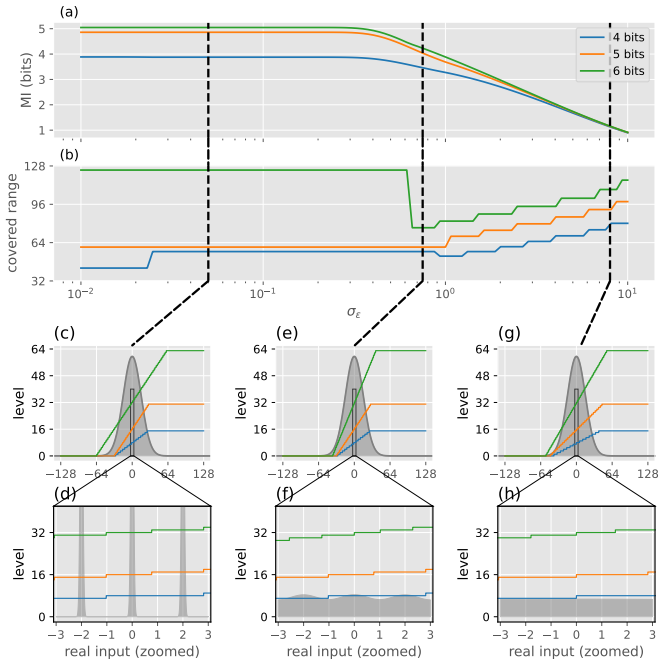

Fig. 4. The optimal quantization as a function of noise amplitude σ_ε for $N = 256$. (a) MI for $\rho \in \{4, 5, 6\}$. (b) The covered range with optimally chosen parameters for each ρ. (c, e, g) Distribution of unquantized results (gray fill) and quantization function for each ρ (solid lines) at $\sigma_\varepsilon \in \{0.05, 0.75, 8\}$. (d, f, h) Corresponding zoomed-in views of the center region.

V. Understanding the Influence of Noise

Our analysis above assumed bounded noise ε that could be neglected. But a real compute-in-memory system is likely subject to (possibly unbounded) noise. To account for this, we use the more general Eqn. (1) to estimate MI, which includes the noise term ε. We now sweep the noise amplitude σ_ε over a wide range for three different ADC resolutions $\rho_i \in \{4, 5, 6\}$ while keeping $N = 256$ constant.

For low noise levels, the results match our earlier observations: ADCs with $\rho_2 = 5$ and $\rho_3 = 6$ can cover (more than) their respective regions of interest $\hat{w}_2 \approx 3.2 \cdot 16 \approx 51$ and $\hat{w}_3 \approx 3.5 \cdot 16 = 56$ with $\alpha = \delta = 2$ without loss, resulting in $w_2 = 64$ and $w_3 = 128$. The resolution $\rho_1 = 4 \ll \text{ENOB}(N) = 5.05$ is too small for that, so a larger $\alpha \approx 4$ is needed to cover $\hat{w} \approx 2.8 \cdot 16 \approx 45$.

At around $\sigma_\varepsilon \approx 0.4$, performance begins to degrade as the noisy measurements $\tilde{y} + \varepsilon$ increasingly blend into a single normal distribution. At around $\sigma_\varepsilon \approx 0.7$, the distribution loses its apparent discrete characteristics, and the optimal threshold placement changes abruptly $\rho = 6$. As it has more than twice the levels needed to cover $\hat{w}$ without loss, the optimal solution now places two thresholds between each possible value of $\tilde{y}$, instead reducing the covered range. Note that this trade-off is worthwhile (only) in the high-noise case, where these additional levels can help to reduce ambiguity.

As the noise amplitude increases further, the covered ranges increase with the logarithm of σ_ε - an effect likely due to a reduction in signal-to-noise ratio within the region of interest[1].

VI. Discussion

We propose that designers of compute-in-memory systems exploit the central limit theorem and design their ADCs with a resolution close to the distribution's ENOB and a reduced range that focuses on a central region-of-interest. This can obviate almost half the ADCs bits when compared to a full-resolution implementation. Our systematic analysis found that the optimally covered range scales proportionally to the ADC's input's standard deviation, but with a constant of proportionality κ_ρ that itself depends on the chosen resolution ρ. If ρ is large enough, the optimal solution is to cover (at least) each bin in the central region by a uniquely corresponding ADC level; otherwise, it is best to stretch the ADC's range to fully cover the region of interest.

Future work should verify that these results generalize beyond our simplifying assumptions of i.i.d. elements and bipolar values.

VII. Code Availability

All code used in this paper is available at: https://github.com/Integrated-Systems-Neuroengineering/optquant.

References

[1] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electronics*, vol. 1, no. 6, pp. 333–343, Jun. 2018, number: 6 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41928-018-0092-2

[2] Q. Xia and J. J. Yang, "Memristive crossbar arrays for brain-inspired computing," *Nature Materials*, vol. 18, no. 4, pp. 309–323, Apr. 2019, number: 4 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41563-019-0291-x

[3] A. S. Rekhi, B. Zimmer, N. Nedovic, N. Liu, R. Venkatesan, M. Wang, B. Khailany, W. J. Dally, and C. T. Gray, "Analog/Mixed-Signal Hardware Error Modeling for Deep Learning Inference," in *Proceedings of the 56th Annual Design Automation Conference 2019*. Las Vegas NV USA: ACM, Jun. 2019, pp. 1–6. [Online]. Available: https://dl.acm.org/doi/10.1145/3316781.3317770

[4] S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, "XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks," *IEEE Journal of Solid-State Circuits*, pp. 1–11, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8959407/

[5] Z. Jiang, S. Yin, J.-S. Seo, and M. Seok, "C3SRAM: An In-Memory-Computing SRAM Macro Based on Robust Capacitive Coupling Computing Mechanism," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 7, pp. 1888–1897, Jul. 2020, conference Name: IEEE Journal of Solid-State Circuits. [Online]. Available: https://ieeexplore.ieee.org/document/9094713

[6] H. Sun, Z. Zhu, Y. Cai, X. Chen, Y. Wang, and H. Yang, "An Energy-Efficient Quantized and Regularized Training Framework For Processing-In-Memory Accelerators," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. Beijing, China: IEEE, Jan. 2020, pp. 325–330. [Online]. Available: https://ieeexplore.ieee.org/document/9045192/

[7] S. K. Gonugondla, C. Sakr, H. Dbouk, and N. R. Shanbhag, "Fundamental Limits on Energy-Delay-Accuracy of In-Memory Architectures in Inference Applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 10, pp. 3188–3201, Oct. 2022, conference Name: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. [Online]. Available: https://ieeexplore.ieee.org/document/9598887/?arnumber=9598887

[1] If this effect was instead just due to an increase in the width of the distribution of $\tilde{y} + \varepsilon$, we'd expect a growth proportional to $\sqrt{\tilde{y}^2 + \sigma_\varepsilon^2}$.

[8] Y. Kim, H. Kim, and J.-J. Kim, "Extreme Partial-Sum Quantization for Analog Computing-In-Memory Neural Network Accelerators," *J. Emerg. Technol. Comput. Syst.*, vol. 18, no. 4, pp. 75:1–75:19, Oct. 2022. [Online]. Available: https://doi.org/10.1145/3528104

[9] S. Huang, H. Jiang, and S. Yu, "Hardware-aware Quantization/Mapping Strategies for Compute-in-Memory Accelerators," *ACM Transactions on Design Automation of Electronic Systems*, vol. 28, no. 3, pp. 1–23, May 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3569940

[10] N. Laubeuf, J. Doevenspeck, I. A. Papistas, M. Caselli, S. Cosemans, P. Vrancx, D. Bhattacharjee, A. Mallik, P. Debacker, D. Verkest, F. Catthoor, and R. Lauwereins, "Dynamic quantization range control for analog-in-memory neural networks acceleration," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 27, no. 5, Jun. 2022. [Online]. Available: https://doi.org/10.1145/3498328

[11] R. Genov and G. Cauwenberghs, "Stochastic mixed-signal vlsi architecture for high-dimensional kernel machines," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. MIT Press, 2001. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2001/file/5352696a9ca3397beb79f116f3a33991-Paper.pdf

[12] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," Apr. 2000, arXiv:physics/0004057. [Online]. Available: http://arxiv.org/abs/physics/0004057