Genetic Algorithm-Driven IMC Mapping for CNNs Using Mixed Quantization and MLC FeFETs

Alptekin Vardar*, Franz Müller*, Gonzalo Cuñarro*, Nellie Laleni*,
Nandakishor Yadav*, and Thomas Kämpfe*†
*Fraunhofer IPMS, Dresden, Germany †TU Braunschweig, Braunschweig, Germany
Email: alptekin.vardar@ipms.fraunhofer.de, thomas.kaempfe@ipms.fraunhofer.de

***Abstract*—Ferroelectric Field-Effect Transistors (FeFETs) are emerging as a highly promising non-volatile memory (NVM) technology for in-memory computing architectures, thanks to their low power consumption and non-volatility. These characteristics make FeFETs particularly well-suited for convolutional neural networks (CNNs), especially in power-constrained environments where minimizing the memory footprint is critical for improving both area efficiency and energy consumption. Two effective strategies for reducing memory requirements are quantization and the use of multi-level cell (MLC) configurations in NVMs. This work proposes a solution that combines mixed quantization schemes with FeFET-based MLC and single-level cell (SLC) configurations to balance memory usage and accuracy. Given the large hyperparameter space introduced by these combinations, we employ a genetic algorithm to efficiently explore and identify Pareto-optimal solutions, allowing flexible adaptation to various application-specific requirements. Our approach achieves significant improvements in both memory efficiency and performance, reducing memory usage by 50% while sacrificing only 3% accuracy compared to the 8-bit ResNet baseline. After a single epoch of retraining, the accuracy matches the baseline while fully retaining the memory savings. Additionally, when compared to the 4-bit baseline, a 46% memory reduction is achieved with virtually no loss in accuracy.**

***Index Terms*—FeFET, Neural Networks, Non-Volatile Memory, Edge Computing, In-Memory Computing, Hyperparameter Optimization, MLC, Genetic Algorithm**

I. Introduction

The increasing demand for Artificial Intelligence (AI) applications, especially in edge computing, has created a pressing need for efficient memory solutions that can operate under stringent resource constraints [1]. Edge devices must balance performance and power consumption within limited hardware capabilities. These constraints have made memory footprint optimization a critical design consideration for energy-efficient computation [2]. Non-volatile memory (NVM) technologies have emerged as a promising approach to address these challenges, offering the ability to retain data without a continuous power supply, which significantly reduces energy consumption. Among the various NVM candidates, technologies such as Resistive RAM (RRAM) [3], Magnetoresistive RAM [4], and Phase-Change Memory (PCM) [5] have been explored for their suitability in edge AI applications. Ferroelectric Field-Effect Transistors (FeFETs) are another emerging NVM technology with intrinsic low power consumption and fast switching speeds, making them attractive for in-memory computing architectures [6] [7] [8].

Despite the advantages offered, the challenge of minimizing memory footprint remains significant, impacting not only energy consumption but also the area and cost of these systems. Two established strategies for reducing memory demands are the use of quantization techniques and Multi-Level Cell (MLC) configurations. Quantization reduces the precision of Convolutional Neural Network (CNN) weights, thereby lowering memory usage while trading off some level of accuracy [9]. MLC configurations in NVMs, on the other hand, allow multiple bits to be stored per memory cell, reducing the memory area but introducing higher error rates due to reduced noise margins between stored levels [10]. Combining quantization and MLC strategies, however, creates an exponentially expanding design space, making manual evaluation of all possible configurations impractical. Each combination involves unique trade-offs between memory savings and accuracy, requiring careful optimization to meet the specific needs of the application.

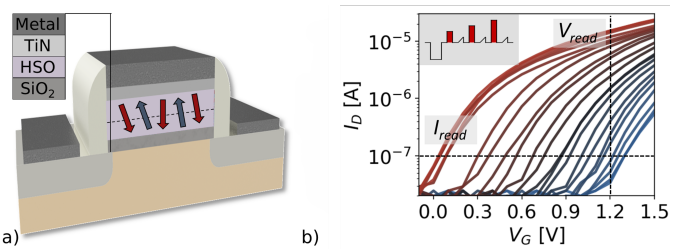

Fig. 1: a) Schematic of the FeFET device; (b) Transfer characteristics of the FeFET device following different programming voltage pulses. [11]

In this paper, we propose a design automation strategy to address the complexity of mixed MLC/SLC configurations and quantization schemes. By conducting sensitivity analysis on deeper quantizations and errors at the layer and bit levels, we reduce the hyperparameter search space into a manageable form. To efficiently explore this space, we employ a genetic algorithm to identify Pareto-optimal solutions, allowing flexible trade-offs between memory efficiency and accuracy based on application needs. Using a FeFET-based architecture as a case study, we demonstrate the effectiveness of our approach in optimizing memory footprint while maintaining accuracy. Most errors introduced by the MLC configurations either require no retraining, or can achieve target accuracy with just one additional epoch.

978-3-9826741-0-0/DATE25/© 2025 EDAA

Specifically, our contributions are as follows:

- We introduce a sensitivity analysis method that transforms the hyperparameter space of quantization and cell level MLC combinations into a tractable form, allowing for more efficient optimization of memory footprint while balancing accuracy.
- We propose a genetic algorithm-based framework to efficiently explore the large hyperparameter space created by mixed quantization levels and MLC/SLC configurations.
- Demonstration of Pareto-optimal solutions using FeFET-based designs, offering flexible trade-offs between memory savings and accuracy with minimal retraining.

II. Background

A. MLC in HSO FeFET Devices

In FeFETs, data storage is achieved by modulating the polarization state of the ferroelectric layer through the application of a high gate voltage. This polarization controls charge carriers in the channel, altering the device's electrical properties. A positive gate voltage accumulates charge carriers, turning the transistor on, while a negative voltage depletes the channel and turns it off, as shown in Fig. 1a. This reversible polarization enables non-volatile data storage, making FeFETs ideal for memory and computational tasks. The FeFETs used in this study are MFIS structures (450×450 nm^2) fabricated with GlobalFoundries' 28 nm HKMG technology, utilizing a Silicon-doped HfO_2 (HSO) ferroelectric layer and a SiO_2 interface layer, which are critical for switching behavior.

MLC capability in FeFETs is enabled by gradual polarization of ferroelectric domains at varying angles, allowing multiple bits per cell. However, the direct field-coupling between the HSO layer and the channel introduces current percolation paths [12], [13], which, along with device scaling and large ferroelectric grains, causes variation between devices. To maintain performance, advanced write-verify schemes are employed for precise control, as depicted in Fig. 1b.

FeFETs are written by applying a linearly increasing gate voltage (V_G) to the wordlines (WLs) while keeping the sourceline (SL) and bitline (BL) at 0V. A write-verify scheme alternates set-pulses and readouts to analyze the I_D-V_G curves and ensure proper I_{on}/I_{off} ratios and threshold voltage (V_T) separation. After resetting with a -5V pulse, devices are incrementally written to target states, ensuring accurate state control and error mapping.

In the crossbar used for the measurements, the FeFETs are organized in an AND-connected array configuration with 9 wordlines (WL) and 7 sourcelines (SL) or bitlines (BL), resulting in 63 FeFETs per array. The layout of this configuration is illustrated in Fig. 2a. Each BL/SL pair is accessed in parallel through individual Source-Measure Units (SMUs), enabling precise control and measurement of device characteristics across the array [11]. A random 2-bit pattern with uniformly distributed states was programmed into the arrays. Readout was performed at $V_{WL} = 1.4V$ and $V_{BL} = 1V$. A total of 25 arrays, each containing 63 FeFETs, were programmed,

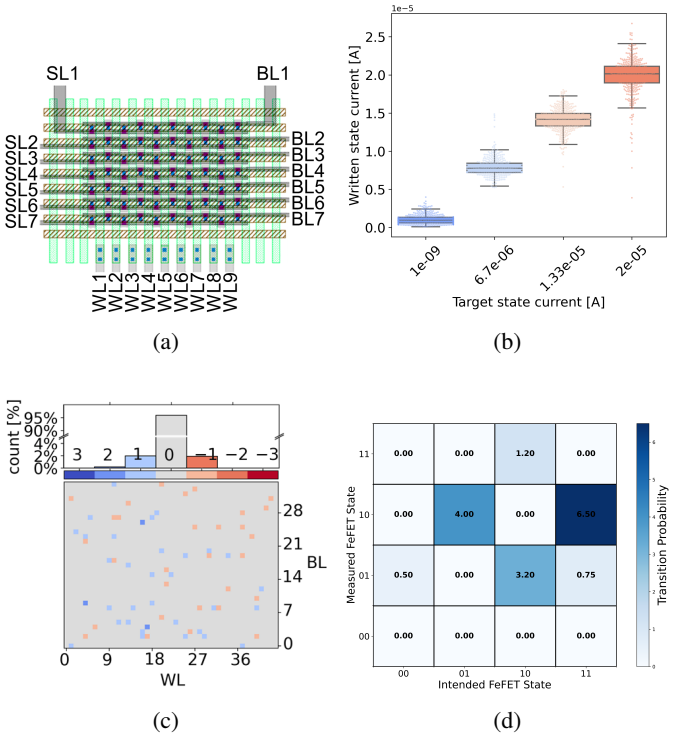

Fig. 2: (a) The layout of 63 FeFETs arranged in an AND-connected array [11], (b) Box plots comparing the written states to the targeted states [14], (c) A BER map of the wafer showing individual bit error locations and their respective distances from their target [11], and (d) A heat map illustrating the distribution of MLC error states.

and their states were verified. The resultant currents for the four target levels are shown in Fig. 2b [14]. Verification revealed an average bit error rate (BER) of 4%, with error rates varying across states due to inherent stability differences. A map of individual bit error locations and their respective distances from their target is shown in Fig. 2c. Importantly, most errors occurred between neighboring states, indicating minimal state transitions. This suggests that many errors involve small deviations. A detailed heat map of error flows and magnitudes is provided in Fig. 2d, offering further insights into the spatial distribution and impact of errors across the array. This characterization serves as the baseline for simulations presented in subsequent sections.

B. Quantization in Convolutional Neural Networks

Quantization is essential for adapting neural networks to hardware constraints, particularly in energy- and memory-limited environments like edge devices. By reducing data precision—typically from 32-bit floating-point to lower-precision formats such as 8-bit or 4-bit integers—quantization reduces both computational complexity and memory usage, enabling efficient deployment in resource-constrained systems.

Quantization maps continuous values to discrete levels, with frameworks like QKeras [15] allowing flexible, layer-wise quantization. QKeras quantizes a floating-point number

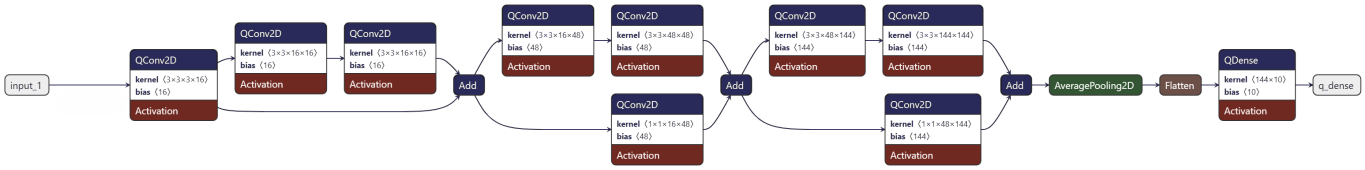

Fig. 3: Resnet-8 Architecture Used for the Demonstration

$(-1)^s 2^e m$, where s, e, and m represent the sign bit, exponent, and mantissa, using the formula:

$$2^{\text{int\_bits}+1} \times \text{clip}\left(\text{round}\left(x \times 2^{\text{bits-int}-1}\right), -2^{\text{bits}-1}, 2^{\text{bits}-1}-1\right)$$

This transformation improves both performance and memory efficiency—key for power-constrained environments. However, applying the same bit-width across all layers can be inefficient, as layers vary in sensitivity to precision loss. Mixed-precision quantization solves this by assigning different bit-widths based on sensitivity [16], [17], [18], allowing a trade-off between accuracy and resource savings. Methods like HAWQ [19] and OMPQ [20] efficiently explore this space, though they are computationally expensive, while simpler criterion-based methods allocate bit-widths based on predetermined sensitivities.

Most mixed-precision efforts focus solely on quantization, overlooking strategies like mixed MLC/SLC configurations in non-volatile memory. Combining MLC/SLC with mixed-precision quantization expands the design space by optimizing both data precision and memory configuration. While complex, this approach offers significant efficiency gains through a balanced trade-off between precision and memory footprint.

III. Layer-wise Sensitivity Analysis to Quantization and MLC Strategies

To evaluate our proposed method, we used a ResNet8 [21] model. The layer-by-layer details, including the weight matrices, are shown in Fig. 3. The model was trained on the CIFAR-10 [22] dataset. The architecture consists of 8 layers: 7 convolutional layers and 1 fully connected output layer. Although ResNet8 with FeFET is our demonstration case, the approach is easily generalizable to other architectures and non-volatile memory (NVM) devices.

After establishing a 12-bit uniform baseline for memory footprint and loss, we progressively quantized the layers to 8-bit, 4-bit, and 2-bit representations, capturing the accuracy and memory savings at each step. Skip connections were bundled with their respective layers, as we observed better performance when handled together due to their small size and minimal impact on overall memory. For each quantization level, we applied a variety of SLC and MLC configurations to explore the trade-offs between memory savings and error rates. The final fully connected(dense) layer was excluded from MLC configurations, as it exhibited poor error tolerance without subsequent layers to compensate for injected errors.

To evaluate the network's sensitivity to quantization and MLC-induced errors, we performed non-uniform error injections using hardware-driven probabilities [23], as described in Section 2B. These injections closely mimicked real-world conditions and were repeated to account for variability in the probabilistic distribution of bit flips, as detailed in Fig. 4. Loss was used as a more reliable indicator of the network's resilience to errors, offering smoother convergence and stability compared to accuracy, which can fluctuate due to small changes. Since individual weights within a layer exhibit varying sensitivities, each experimental run produced slightly different losses. To address this, we conducted 100 repetitions of each experiment and averaged the results. Interestingly, in some layers, introducing errors or deeper quantization improved accuracy by reducing overfitting, as controlled noise or precision loss can enhance generalization [24].

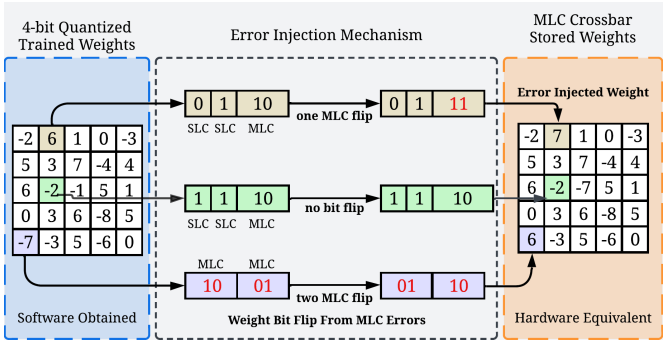

Fig. 4: Detailed MLC error injection scheme

This process allowed us to compile a detailed sensitivity profile for each layer, summarizing the trade-offs between different solutions, as shown in Table I. Each row in the table corresponds to a different layer of the ResNet8 network, while the columns represent different pairings of quantization and MLC/SLC configurations. The solution index at the bottom of the table provides a reference for each unique configuration, allowing for easy identification in subsequent discussions.

In each cell, two metrics are presented: the percentage relative memory reduction (%ΔMemory) and the percentage change in loss (%ΔLoss) compared to the baseline model. A smaller loss value indicates better model performance, while a larger memory saving value reflects greater efficiency. For example, in layer two under solution index 6, where 4-bit quantization is applied with SLC for the 2 most significant bits and MLC for the 2 least, the tradeoff between memory and loss can be found.

	Quantized 8					Quantized 4			Quantized 2	
No. MLC / Layer	4	3	2	1	0	2	1	0	1	0
1	0.099 / 157.00	0.087 / 5.83	0.074 / -0.25	0.062 / -0.56	0.050 / -0.57	0.124 / 71.42	0.111 / 2.37	0.099 / 0.59	0.136 / 10.06	0.124 / 1.86
2	0.528 / 131.05	0.462 / 2.79	0.396 / -1.24	0.330 / -1.26	0.264 / -1.14	0.661 / 20.45	0.595 / -1.74	0.528 / -1.97	0.727 / 5.65	0.661 / 1.65
3	0.528 / 226.51	0.462 / 4.77	0.396 / -2.72	0.330 / -3.08	0.264 / -3.17	0.661 / 47.76	0.595 / 0.51	0.528 / -1.10	0.727 / 5.89	0.661 / 1.67
4	1.585 / 295.07	1.387 / 22.78	1.189 / -0.35	0.991 / -0.64	0.793 / -0.34	1.982 / 100.50	1.784 / 7.10	1.585 / -0.09	2.180 / 24.88	1.982 / 6.48
5	4.933 / 567.63	4.316 / 43.12	3.699 / -0.19	3.083 / -1.87	2.466 / -2.01	6.166 / 119.72	5.549 / 12.14	4.933 / 2.75	6.782 / 54.54	6.166 / 22.96
6	14.269 / 569.74	12.486 / 19.55	10.702 / -0.43	8.918 / -1.78	7.135 / -1.93	17.836 / 328.78	16.053 / 19.35	14.269 / 0.25	19.620 / 145.81	17.836 / 19.36
7	44.393 / 108.49	38.844 / 8.18	33.295 / -0.06	27.746 / 0.35	22.197 / 0.53	55.491 / 24.10	49.942 / 6.10	44.393 / 4.11	61.040 / 40.31	55.491 / 20.44
8	- / -	- / -	- / -	- / -	-0.40 / 0.16	- / -	- / -	-0.10 / 0.33	- / -	2.24 / 0.40
Solution Index:	0	1	2	3	4	5	6	7	8	9

TABLE I: Percentage Relative Loss and Memory Reduction(%ΔMemory / %ΔLoss) Comparison for Quantization Levels and MLC/SLC Configurations Across All 8 Layers

IV. Pareto Optimal Mapping with a Genetic Algorithm

A genetic algorithm (GA) is a search heuristic inspired by natural selection, designed to explore large and complex search spaces efficiently. It mimics evolutionary processes such as selection, crossover, and mutation to iteratively improve a population of potential solutions. GAs are particularly effective for optimization problems where the solution space is too vast for traditional methods [25], [26].

In our work, the GA is applied to optimize the trade-off between memory savings and accuracy through mixed quantization and MLC/SLC configurations across the layers of the ResNet8 model. Each possible configuration, as represented by the solution indexes in Table I, serves as a "gene" in the GA's search process. These configurations encode the quantization level and MLC/SLC pairing for each layer, allowing the GA to explore combinations that minimize memory usage while maintaining acceptable accuracy.

To enhance the effectiveness of the GA, we treat mutation rates and crossover rates as evolvable parameters, allowing them to adapt alongside the solutions as genes. Mutation introduces random changes to maintain diversity, while crossover combines traits from two individuals to create new solutions. Tournament size, which defines how many individuals compete for selection in each generation, and the elite rate, which controls how many top-performing solutions are preserved unchanged, are periodically adjusted throughout the optimization process. Initially, smaller tournament sizes and elite rates are used to promote diversity, enabling the algorithm to explore a broad range of potential solutions. As the algorithm progresses and starts refining promising candidates, both tournament size and elite rate increase, ensuring that the best solutions are retained and further refined. This adaptive structure allows the GA to balance exploration and exploitation effectively, starting with a wide search of the solution space and gradually focusing on the most successful configurations as convergence nears.

After the population evolves through adaptive mechanisms, the GA applies Pareto optimization to identify a set of Pareto-optimal solutions. Since memory and accuracy are conflicting objectives, no single solution optimizes both. A solution is Pareto-optimal if improving one objective, such as memory savings, necessarily sacrifices the other, like accuracy. Instead of converging to a single outcome, the GA generates a range of trade-offs, each representing a different balance between memory efficiency and accuracy. These solutions offer flexibility, allowing users to select configurations that prioritize either memory savings or accuracy, based on specific application requirements. By navigating the trade-offs between quantization levels and MLC/SLC configurations, the GA ensures that the final set of solutions is diverse and practical for real-world use.

V. Results and Evaluation

In this section, we analyze the results of the GA, focusing on key parameters and the numerical analysis of the trade-offs between memory efficiency and accuracy found in the Pareto-optimal solutions.

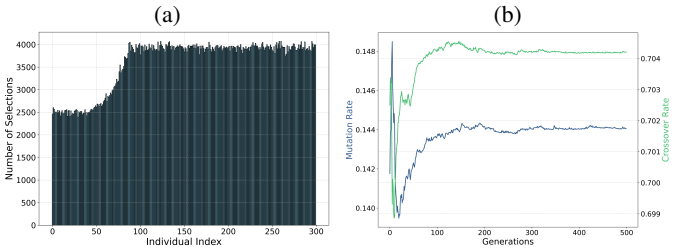

Fig. 5: (a) Cumulative Selection Pressure (b) Mutation and Crossover Evolution

The Cumulative Selection Pressure graph in Fig. 5a visualizes how frequently individuals from the population are selected for reproduction. Some individuals with higher performance are selected more frequently, but the overall distribution remains broad enough to ensure that diversity is preserved, preventing premature convergence and allowing for further exploration of potential improvements. The Mutation and Crossover Rate Evolution graph shown in Fig. 5b illustrates how these rates adapt as the GA progresses. Initially, both fluctuates, encouraging broad exploration of the solution space. As the GA approaches convergence, both rates stabilize at their optimal values. This convergence reflects the transition from

early-stage exploration to later-stage exploitation, where the algorithm focuses on refining the most promising solutions rather than exploring entirely new possibilities.

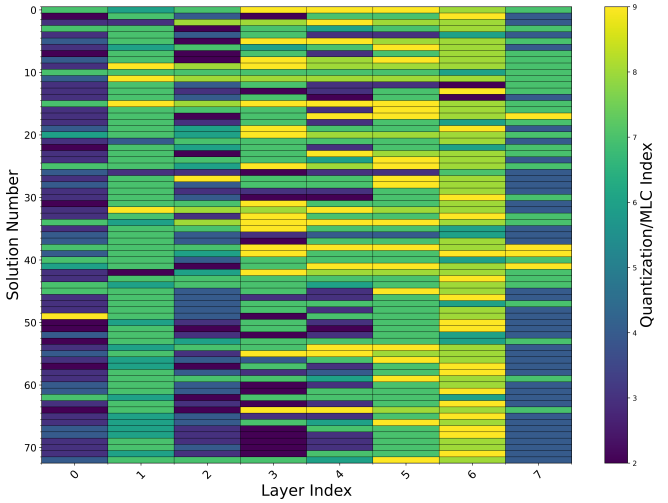

Fig. 6: Heatmap of Quantization/MLC Configurations Across Layers for Pareto-Optimal Solutions

The Solution Heatmap in Fig.6 visualizes the decision variables across Pareto-optimal solutions, offering insight into frequently selected configurations of quantization levels or MLC/SLC setups, across layers. Each solution on the y-axis represents a different Pareto-optimal configuration, while the x-axis corresponds to the layer indices. The color gradient indicates the configuration chosen for each layer in the respective solutions. Although a range of quantization/MLC pairings are possible, patterns emerge based on each layer's sensitivity and memory demands. Certain layers consistently favor specific configurations, reflecting how their trade-offs between memory and accuracy shape the overall network's optimization. This demonstrates that layer-specific constraints play a key role in determining the most likely solutions.

Finally, the Pareto Front graph in Fig. 7 displays the set of Pareto-optimal solutions discovered by the GA, visually representing the trade-offs between memory footprint and accuracy. This overview allows users to identify the best solutions that meet their specific constraints, enabling informed decision-making depending on application priorities. The graph can be divided into three distinct regions. In Region A, significant memory footprint improvements are achieved with little to no impact on accuracy, making it ideal for applications where accuracy is critical, but some memory savings are still desirable. Region B represents the most competitive set of solutions, where careful selection of configurations is necessary. Here, users can find optimal trade-offs between memory savings and accuracy, and even slight adjustments in quantization or MLC configurations can lead to measurable benefits. Finally, Region C is where memory reductions are highest, suited for highly constrained systems. However, the cumulative effect of deeper quantization levels and multiple MLC configurations results in a substantial penalty in accuracy, making it suitable only

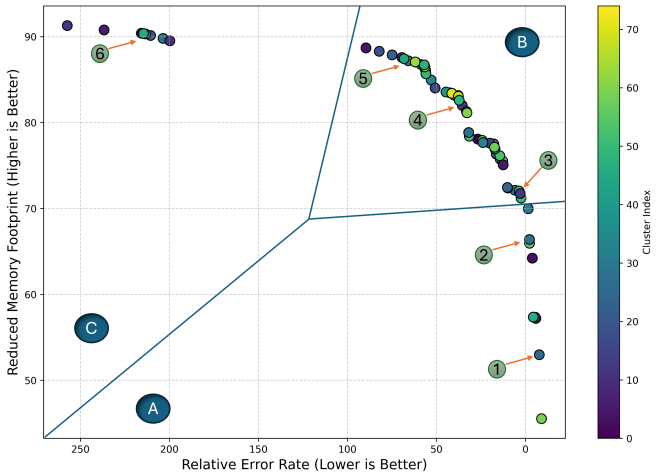

Fig. 7: Optimal Memory and Accuracy Trade-offs Across Pareto Solutions

for cases where memory constraints outweigh performance concerns.

From these regions, six representative solutions (labeled 1-6) were selected to demonstrate the trade-offs between memory efficiency and accuracy. These points cover a range of configurations and illustrate how mixed strategies compare to traditional fixed-precision approaches. Detailed comparisons are shown in Table II. Each scenario was run 400 times, and the averages were taken.

Solutions 1 through 4 demonstrate that substantial memory savings can be achieved with minimal accuracy loss. For example, Solutions 3 and 4 provide significant memory reductions while maintaining acceptable accuracy compared to the 4-bit baseline, making them ideal when a fixed 4-bit configuration is too memory-intensive, but the accuracy of a 2-bit setup is insufficient. Solution 3 reduces memory usage by nearly 17% compared to the 4-bit configuration, while still achieving 84.5% accuracy. Similarly, Solution 4 cuts memory by almost 46%, maintaining an accuracy of 80.6%. These solutions offer a balanced approach for systems where fixed-precision options fail to meet both memory and accuracy demands, highlighting the flexibility and efficiency of mixed quantization and MLC configurations.

Next, we introduce the Single Epoch Error Compensation Process, designed to mitigate accuracy loss caused by MLC errors. In mixed-signal systems, the mapping of weights to memory cells is critical for maintaining synchronization, and adjusting this mapping or the pipelining structure to account for errors can disrupting system performance. By using the program-verify scheme, as described in section 2A, we can identify weights compromised by MLC-induced errors. To address these errors, a single-epoch retraining process is performed, during which the erroneous weights remain untouched, as they cannot be reliably updated, while the healthy weights are selectively adjusted to compensate. This retraining lasts for only one epoch and involves minimal

Solution Number	Variables for Each Layer	Memory (Kilo Devices)	Accuracy w/o Retraining	Accuracy w/ Retraining
Full 8	[4, 4, 4, 4, 4, 4, 4, 4]	2325.1k	89%	-
Full 4	[7, 7, 7, 7, 7, 7, 7, 7]	1162.6k	86.8%	-
Full 2	[9, 9, 9, 9, 9, 9, 9, 9]	581.3k	78.5%	-
Solution 1	[4, 6, 3, 2, 3, 7, 2, 4]	1640k	87%	89.5%
Solution 2	[4, 7, 3, 7, 3, 2, 6, 4]	1172.3k	86%	89%
Solution 3	[3, 6, 3, 9, 7, 7, 6, 7]	961.1k	84.5%	88.4%
Solution 4	[2, 6, 2, 7, 2, 7, 8, 7]	628.1k	80.6%	86.4%
Solution 5	[7, 6, 7, 7, 6, 9, 8, 7]	433.7k	61%	83.8%
Solution 6	[7, 6, 9, 8, 9, 8, 8, 9]	321.8k	18.8%	77%

TABLE II: Selected Pareto Optimal solutions with Corresponding Variables, Memory, and Accuracy Before and After Compensation Training

weight updates, offering a computationally efficient solution that avoids changes to the mapping or scheduling pipeline, making it well-suited for mixed-signal systems.

After the retraining, Solutions 1 and 2 demonstrate significant memory savings while achieving performance comparable to or even surpassing the 8-bit baseline. Solution 1, with a 29.5% memory reduction, improves its accuracy to 89.5%, slightly outperforming the 8-bit baseline's 89%. Similarly, Solution 2 achieves a 49.5% memory reduction and a post-retraining accuracy of 89%, matching the 8-bit network. The slight improvement in accuracy can be attributed to reduced overfitting, as quantization and MLC-induced noise help the model generalize better. Compared to the 4-bit baseline, Solution 3, which reduces memory by 17%, reaches 88.4% accuracy after retraining, greatly surpassing the 4-bit baseline accuracy of 86.8%. Likewise, Solution 4 achieves an aggressive 46% memory reduction over the already deeply quantized 4-bit version, recovering from 80.6% to 86.4% accuracy after retraining, making it nearly as accurate as the 4-bit configuration. Solutions 5 and 6 represent the most extreme memory-saving configurations, aimed at systems with highly constrained resources. After retraining, Solution 5 achieves 83.8% accuracy with a 25% memory reduction compared to the 2-bit configuration, while Solution 6 reaches 77% accuracy with a 45% memory reduction compared to the 2-bit version. These configurations offer comparable or better accuracy and significant memory savings over the standard 2-bit setup, making them ideal for systems requiring extreme memory efficiency while maintaining acceptable performance.

Lastly, the box plots in Fig. 8 illustrate the accuracy distributions across 400 runs for all six solutions, both before and after compensation training. Prior to retraining, there is significant variation in accuracy between runs, largely due to the random nature of error injection, which is particularly evident in configurations with aggressive memory savings. However, after a single epoch of compensation training, both accuracy and stability are noticeably improved, as reflected by the reduced variance in the lower plot. This correction scheme restores accuracy and ensures consistent performance, essential for the reliability and stability of real-world applications.

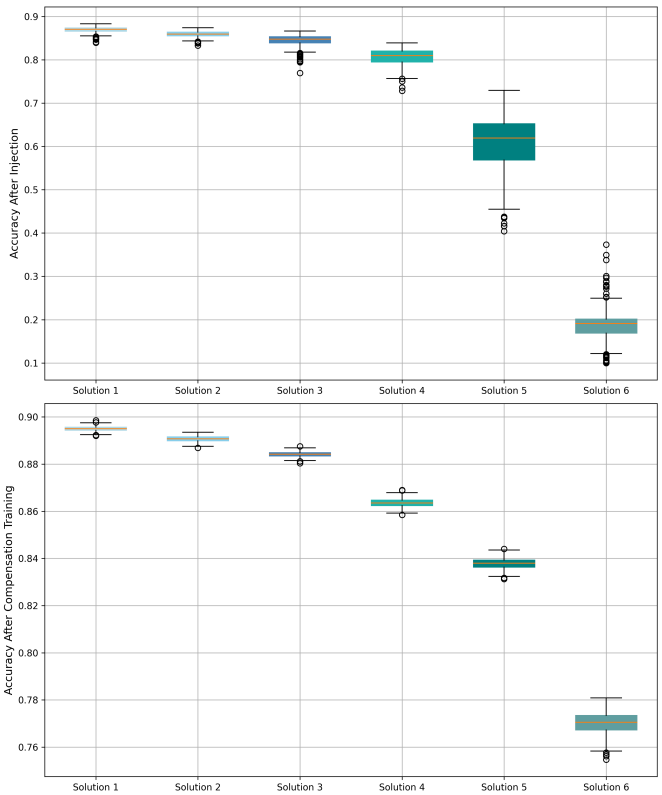

Fig. 8: Box Plot of Accuracy Distribution After Error Injection and Retraining

VI. Conclusion

This work demonstrates the potential of combining mixed quantization with MLC configurations to optimize the memory footprint of CNNs. By employing a genetic algorithm, we efficiently explore the large design space, achieving an optimal balance between memory savings and accuracy. The genetic algorithm identifies Pareto-optimal solutions that offer flexible trade-offs tailored to specific application needs. For example, our method achieves up to 50% memory savings with only a 3% accuracy reduction compared to an 8-bit baseline, and after a single epoch of retraining, the accuracy fully matches the baseline while retaining the memory reduction. Additionally, compared to a 4-bit baseline, our method delivers a 46% memory reduction with minimal accuracy loss, effectively balancing memory efficiency while reducing accuracy variability. The approach we present is adaptable to other CNN architectures and NVM technologies, making it a versatile and robust solution for memory-constrained environments.

Acknowledgements

This research was funded by the NeurOSmart project, under the auspices of the Fraunhofer-Gesellschaft and financially supported by the Federal Ministry of Education and Research (BMBF), Germany under the project "DE-TW-FeEdge" (16ME0981). We thank GlobalFondries for providing the wafers containing the FeFETs, fabricated in GlobalFoundries 28nm HKMG node.

REFERENCES

[1] X. Xu, Y. Ding, S. Hu, M. Niemier, J. Cong, Y. Hu, and Y. Shi, "Scaling for edge inference of deep neural networks," *Nature Electronics*, vol. 1, 04 2018.

[2] F. Daghero, D. J. Pagliari, and M. Poncino, "Chapter eight - energy-efficient deep learning inference on edge devices," in *Hardware Accelerator Systems for Artificial Intelligence and Machine Learning*, ser. Advances in Computers, S. Kim and G. C. Deka, Eds. Elsevier, 2021, vol. 122, pp. 247–301. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0065245820300553

[3] W. He, S. Yin, Y. Kim, X. Sun, J.-J. Kim, S. Yu, and J.-S. Seo, "2-bit-per-cell rram-based in-memory computing for area-/energy-efficient deep learning," *IEEE Solid-State Circuits Letters*, vol. 3, pp. 194–197, 2020.

[4] S. Jung, H. Lee, S. Myung, H. Kim, S. K. Yoon, S.-W. Kwon, Y. Ju, M. Kim, W. Yi, S. Han, B. Kwon, B. Seo, K. Lee, G.-H. Koh, K. Lee, Y. Song, C. Choi, D. Ham, and S. J. Kim, "A crossbar array of magnetoresistive memory devices for in-memory computing," *Nature*, vol. 601, no. 7892, pp. 211–216, 2022. [Online]. Available: https://doi.org/10.1038/s41586-021-04196-6

[5] S. Ambrogio, P. Narayanan, A. Okazaki, A. Fasoli, C. Mackin, K. Hosokawa, A. Nomura, T. Yasuda, A. Chen, A. Friz, M. Ishii, J. Luquin, Y. Kohda, N. Saulnier, K. Brew, S. Choi, I. Ok, T. Philip, V. Chan, C. Silvestre, I. Ahsan, V. Narayanan, H. Tsai, and G. W. Burr, "An analog-ai chip for energy-efficient speech recognition and transcription," *Nature*, vol. 620, no. 7975, pp. 768–775, 2023. [Online]. Available: https://doi.org/10.1038/s41586-023-06337-5

[6] K. Ni, P. Sharma, J. Zhang, M. Jerry, J. A. Smith, K. Tapily, R. Clark, S. Mahapatra, and S. Datta, "Critical role of interlayer in hf0.5zr0.5o2 ferroelectric fet nonvolatile memory performance," *IEEE Transactions on Electron Devices*, vol. 65, no. 6, pp. 2461–2469, 2018.

[7] A. J. Tan, Y.-H. Liao, L.-C. Wang, N. Shanker, J.-H. Bae, C. Hu, and S. Salahuddin, "Ferroelectric hfo2 memory transistors with high-k interfacial layer and write endurance exceeding 1010 cycles," *IEEE Electron Device Letters*, vol. 42, no. 7, pp. 994–997, 2021.

[8] A. I. Khan, A. Keshavarzi, and S. Datta, "The future of ferroelectric field-effect transistor technology," *Nature Electronics*, vol. 3, no. 10, pp. 588–597, 2020. [Online]. Available: https://doi.org/10.1038/s41928-020-00492-7

[9] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, "A white paper on neural network quantization," 2021. [Online]. Available: https://arxiv.org/abs/2106.08295

[10] T. Soliman, S. Chatterjee, N. Laleni, F. Müller, T. Kirchner, N. Wehn, T. Kämpfe, Y. S. Chauhan, and H. Amrouch, "First demonstration of in-memory computing crossbar using multi-level cell fefet," *Nature Communications*, vol. 14, no. 1, p. 6348, 2023. [Online]. Available: https://doi.org/10.1038/s41467-023-42110-y

[11] F. Müller, S. De, R. Olivo, M. Lederer, A. Altawil, R. Hoffmann, T. Kämpfe, T. Ali, S. Dünkel, H. Mulaosmanovic, J. Müller, S. Beyer, K. Seidel, and G. Gerlach, "Multilevel operation of ferroelectric fet memory arrays considering current percolation paths impacting switching behavior," *IEEE Electron Device Letters*, vol. 44, no. 5, pp. 757–760, 2023.

[12] F. Müller, M. Lederer, R. Olivo, T. Ali, R. Hoffmann, H. Mulaosmanovic, S. Beyer, S. Dünkel, J. Müller, S. Müller, K. Seidel, and G. Gerlach, "Current percolation path impacting switching behavior of ferroelectric fets," in *2021 International Symposium on VLSI Technology, Systems and Applications (VLSI-TSA)*, 2021, pp. 1–2.

[13] K. Ni, S. Thomann, O. Prakash, Z. Zhao, S. Deng, and H. Amrouch, "On the channel percolation in ferroelectric fet towards proper analog states engineering," in *2021 IEEE International Electron Devices Meeting (IEDM)*, 2021, pp. 15.3.1–15.3.4.

[14] F. Müller, S. De, M. Lederer, R. Hoffmann, R. Olivo, T. Kämpfe, K. Seidel, T. Ali, H. Mulaosmanovic, S. Dünkel, J. Müller, S. Beyer, and G. Gerlach, "Multi-level operation of ferroelectric fet memory arrays for compute-in-memory applications," in *2023 IEEE International Memory Workshop (IMW)*, 2023, pp. 1–4.

[15] C. N. Coelho, A. Kuusela, S. Li, H. Zhuang, J. Ngadiuba, T. K. Aarrestad, V. Loncar, M. Pierini, A. A. Pol, and S. Summers, "Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors," *Nature Machine Intelligence*, vol. 3, no. 8, pp. 675–686, Aug. 2021. [Online]. Available: https://doi.org/10.1038/s42256-021-00356-5

[16] C. Tang, K. Ouyang, Z. Wang, Y. Zhu, W. Ji, Y. Wang, and W. Zhu, "Mixed-precision neural network quantization via learned layer-wise importance," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 259–275.

[17] N. P. Pandey, M. Nagel, M. van Baalen, Y. Huang, C. Patel, and T. Blankevoort, "A practical mixed precision algorithm for post-training quantization," in *34th British Machine Vision Conference Workshop Proceedings, BMVC Workshop 2023, Aberdeen, UK, November 20-24, 2023*. BMVA Press, 2023. [Online]. Available: https://workshops.proceedings.bmvc2023.org/InternationalWorkshoponComputationalAspectsofDeepLearning/3/CameraReady/AMP_CameraReady.pdf

[18] A. Vardar, L. Zhang, S. Hu, S. B. Jain, S. Mojumder, N. Laleni, A. Shrivastava, S. De, and T. Kämpfe, "Layer sensitivity aware cnn quantization for resource constrained edge devices," in *2022 9th International Conference on Soft Computing and Machine Intelligence (ISCMI)*, 2022, pp. 26–30.

[19] Z. Dong, Z. Yao, A. Gholami, M. Mahoney, and K. Keutzer, "Hawq: Hessian aware quantization of neural networks with mixed-precision," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 293–302.

[20] Y. Ma, T. Jin, X. Zheng, Y. Wang, H. Li, Y. Wu, G. Jiang, W. Zhang, and R. Ji, "Ompq: Orthogonal mixed precision quantization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, pp. 9029–9037, Jun. 2023. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/26084

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[22] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, 05 2012.

[23] A. Vardar, L. Zhang, S. B. Jain, S. Mojumder, N. Laleni, S. De, and T. Kämpfe, "The true cost of errors in emerging memory devices: A worst-case analysis of device errors in imc for safety-critical applications," in *2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2023, pp. 1–4.

[24] G. Tallec, E. Yvinec, A. Dapogny, and K. Bailly, "Fighting over-fitting with quantization for learning deep neural networks on noisy labels," 2023. [Online]. Available: https://arxiv.org/abs/2303.11803

[25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[26] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.