Evaluating Artificial Neural Networks with Realistic Memristive Variability

R. Romero-Zaliz[1][0000−0003−4500−9781] and JB. Roldán[2][0000−0003−1662−6457]

[1] Research Centre for Information and Communication Technologies (CITIC-UGR), Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), Dept. of Computer Science and A.I., University of Granada, Spain
[2] Dept. of Electronics and Computer Technology, Faculty of Sciences, Institute Carlos I for Theoretical and Computational Physics, University of Granada, Spain
rocio@ugr.es**, jroldan@ugr.es

Abstract. This work presents a hardware-aware evaluation framework for artificial neural networks under realistic variability conditions using experimentally derived characteristics from memristive devices. Unlike conventional approaches based on synthetic or Gaussian noise, we employ statistical distributions obtained from HfO_2-based resistive memory, where conductance variability and post-programming drift follow time-dependent log-logistic behavior. Network weights are mapped to conductance levels, enabling a physically grounded modeling of quantization and variability effects. Results show that moderate quantization (5-6 levels) preserves acceptable performance, while very low resolutions (2-3 levels) significantly degrade accuracy. Experimentally derived noise exhibits a dual effect: high variability reduces performance, whereas moderate noise can improve generalization, acting as an implicit regularizer. Temporal drift does not cause statistically significant degradation when appropriate quantization mappings are used. These findings highlight the importance of hardware-algorithm co-design and support the feasibility of robust ANN deployment on memristive platforms under realistic physical constraints.

Keywords: Neural Networks · Quantization · Memristive Devices · Noise Injection.

1 Introduction

Efforts to run artificial neural networks (ANNs) on emerging hardware platforms have exposed how fragile many standard evaluation practices really are. Most assessments still assume high numerical precision and either ignore noise altogether or treat it as a simple Gaussian perturbation. That picture breaks down quickly when memristive and resistive memory technologies enter the discussion. These devices are appealing for neuromorphic computing because of their energy efficiency and scalability [12, 16], but they bring along behaviors that are

** Corresponding author

hard to avoid. Variability from device to device, stochastic switching, and conductance drift over time are all part of the normal operation of a memristive device. Gaussian noise models, popular in machine learning, do not capture this reality [8, 10]. The result is a mismatch between what algorithms appear to do in simulation and what hardware can actually deliver.

Recent experimental studies on memristive devices have shown that conductance variability across resistance states follows statistical distributions that are shaped by the physics of filament formation and rupture, which are non-Gaussian [10]. When ANN performance is evaluated using oversimplified noise assumptions, the predictions fail to accurately predict performance in real hardware settings [19]. Consequently, there is a growing need for methodologies that incorporate experimentally grounded models of device behavior into algorithmic analysis [6].

Here we develop a hardware-aware evaluation framework that ties experimental device characterization directly to neural network behavior. Trained floating-point weights are first translated into discrete conductance targets, after which noise sampled from experimentally fitted distributions is injected at the device level. This noise is not arbitrary; its statistics are taken from measurements and evolve over time, allowing drift to enter the analysis in a way that mirrors observed memristive behavior [10]. Quantization and variability are therefore treated as physical constraints rather than numerical afterthoughts.

The methodology is applied to a broad set of fully connected network configurations. MNIST and Fashion-MNIST datasets serve as test cases, not as benchmarks to be optimized, but as controlled settings in which the effects of limited resolution, noise amplitude, and temporal evolution can be separated and compared [10]. This setup makes it possible to see how performance changes in time, precision is reduced, or variability increases, and where the assumptions commonly made at the algorithmic level begin to break down.

2 Background and Related Work

ANN Quantization. Quantization cuts down the cost of neural networks by forcing weights and/or activations onto a small set of discrete values. In practice standard quantization approaches include uniform and non-uniform schemes, as well as symmetric and asymmetric mappings [4, 3]. A consistent message across the literature is that moderate precision reduction often creates lighter models. In those cases energy consumption significant drops, while maintaining accuracy within a tolerable range, particularly when quantization-aware training or careful post-training calibration is applied [4]. Pushing precision further, though, quickly exposes limits. Most analyses focus on algorithmic efficiency and treat storage and computation as stable and noise-free. That assumption breaks when working with emerging memory technologies, where variability and state dependence are part of normal operation. When these effects are ignored, quantization strategies that look benign in software may behave quite differently once weights

are stored in physical devices [3].

Memristive Devices and Variability. Resistive memory technologies have attracted sustained attention for neuromorphic and in-memory computing, particularly memristive devices based on transition metal oxides such as HfO_2 [14] due to their ability to represent synaptic weights directly as conductance levels, without the need for digital encoding. In practice, these devices support multiple stable states, which makes analog storage feasible but also introduces some complications. On the one hand, variability, since switching events are inherently stochastic, and both device-to-device dispersion and cycle-to-cycle fluctuations are commonly observed [8, 13]. On the other hand, time after programming, since conductance does not remain fixed but drifts gradually, sometimes over long timescales, altering the stored value [9]. Experimental reports suggest that the distributions are often heavy-tailed, and their parameters evolve with the time elapsed since programming [9].

Noise Modeling in ANNs. Noise has long been introduced into artificial neural networks as a way to probe robustness, improve generalization, or mimic hardware behavior. Common approaches include additive or multiplicative Gaussian perturbations applied during training and/or inference. Historically, these strategies have been linked to regularization effects, limiting overfitting and pushing models toward solutions that transfer better to unseen data [1]. Many studies implicitly assume Gaussian statistics, however, Gaussian-based simulations may underestimate the impact of extreme deviations or fail to reproduce realistic error patterns. Simplified noise models tend to downplay extreme events and produce error patterns that look cleaner than what hardware actually delivers. More recent efforts have started to move away from these abstractions bringing models closer to device reality, but it remains partial [2, 18]. Quantization effects are often treated in isolation, while temporal drift is ignored or folded into static noise terms. Capturing both, especially as they evolve after programming, is still uncommon. How much accuracy is lost by these shortcuts, and when it actually matters at the system level, is not always clear.

3 Methodology

This study follows a two-level experimental design that links device-level measurements directly to ANN evaluation. Quantization and noise are not introduced as abstract perturbations but are derived from experimental statistics reported in [10]. That work analyzes post-programming conductance drift in HfO_2-based memristive devices across several resistance states (HRS and LRS1–LRS6) and time points (After Switching (AS)[1], 10 min, 60 min, and 24 h), showing that conductance is well described by log-logistic distributions whose parameters evolve over time.

[1] The immediate state measured right after the programming pulse.

Those fitted distributions provide the basis for weight modeling here. Floating-point ANN weights are treated as ideal target conductances, which are first quantized onto a finite set of stable levels, following the experimentally observed low-resistance states. Noise, which is connected to the cycle-to-cycle variability, is then injected following the same distributions measured in hardware to mimic the variability from device dispersion and temporal drift.

Evaluation proceeds in three stages: a floating-point baseline without any quantization or noise; a quantization-only pass where weights are discretized onto experimentally observed conductance levels; and a final stage where those quantized weights are further perturbed using the measured noise statistics, repeated at each post-programming time point.

3.1 ANN Architecture Configuration

We studied different basic ANN architectures on two well-known datasets: MNIST [7] and Fashion-MNIST [17]. The experimental setup considers a collection of fully connected ANN architectures (Table 1), varying in both the number of hidden layers and the number of neurons per layer[2]. The preference for fully connected models and standard benchmark datasets over more complex configurations such as convolutional neural networks (CNN) is intentional, as it provides a well-known and controlled experimental environment that enables isolating the effects of device-level variability without introducing additional complexity from advanced architectures or data heterogeneity. Although CNNs have been experimentally implemented in memristive hardware [18], such implementations are typically limited to simplified architectures, thus we only include a simple 3-layer CNN in our study.

3.2 Weight Quantization

We apply quantization by reducing the number of discrete representation levels used to encode ANN weights during inference. We evaluate several quantization granularities (2, 3, 4, and 5 levels) and compare them against a full-precision floating-point baseline.

Because experiments provide six distinct conductance levels, reducing the number of quantization levels allows multiple valid configurations. We construct these configurations by enumerating all possible combinations of the experimentally measured low-resistance states (LRS). For the 2-level case, we consider all pairwise combinations of the six states, which yields 15 configurations spanning from (LRS1, LRS2) to (LRS5, LRS6). For 3 levels, we include all triplets, resulting in 20 configurations, such as (LRS1, LRS2, LRS3) through (LRS4, LRS5, LRS6). Using the same approach, we form 15 configurations for 4 levels by selecting all subsets of four states, and 6 configurations for 5 levels, each excluding

[2] All ANN experiments are conducted independently on an AMD EPYC 7313P 16-Core Processor with a NVIDIA RTX A5000 GPU.

Table 1: Summary of architectures considered for both the MNIST and the Fashion-MNIST experiments. The last architecture is a 3-layer CNN (32–64–128 filters, 3x3 kernels) followed by a 128–10 ANN.

Number of Layers	Neurons per Layer	Total Number of Neurons
1	8	8
1	16	16
1	32	32
1	64	64
1	128	128
2	8	16
2	16	32
2	32	64
2	64	128
2	128	256
3	8	24
3	16	48
3	32	96
3	64	192
3	128	384
4	8	32
4	16	64
4	32	128
4	64	256
4	128	512
2	16–8	24
2	32–16	48
2	32–8	40

N. of Layers	Neurons per Layer	Total N. of Neurons
2	64–32	96
2	64–16	80
2	64–8	72
2	128–64	192
2	128–32	160
2	128–16	144
2	128–8	136
3	32–16–8	56
3	64–32–16	112
3	64–32–8	104
3	64–16–8	88
3	128–64–32	224
3	128–64–16	208
3	128–64–8	200
3	128–32–16	176
3	128–32–8	168
3	128–16–8	152
4	64–32–16–8	120
4	128–64–32–16	240
4	128–64–16–8	216
4	128–32–16–8	184
4	128–64–32–8	232
5	32–64–128–128–10^3	362

one LRS. The 6-level case uses the complete set (LRS1–LRS6). This enumeration covers all quantization subsets derived from the available resistance states.

Following previous work on quantization schemes, we perform weight quantization using asymmetric uniform quantization [5, 15] applied to the trained floating-point weights. We first determine the minimum and maximum weight values to define the effective dynamic range. Then, we linearly rescale the weights to the discrete interval $[0, n-1]$, where n denotes the number of quantization levels. This procedure preserves the original dynamic range by adapting the quantization step size to the weight distribution. Finally, we round the rescaled values to the nearest integer, obtaining uniformly spaced discrete levels.

We assess quantization performance without noise using a reference configuration at each quantization level: for 2 levels (LRS1, LRS2), for 3 levels (LRS1, LRS2, LRS3), for 4 levels (LRS1, LRS2, LRS3, LRS4), for 5 levels (LRS1, LRS2, LRS3, LRS4, LRS5) and for 6 levels (LRS1, LRS2, LRS3, LRS4, LRS5, LRS6). This configuration allows direct comparison with the non-quantized model. We randomly split the dataset into training and test subsets using an 80%-20% ratio

and analyze accuracy metrics to examine the trade-off between representational efficiency and performance loss.

3.3 Noise Injection

To simulate conductance fluctuations, we introduce noise with varying strength. We define temporal checkpoints (AS, 10 min, 60 min, and 24 h) that match those used in the experimental characterization, allowing us to directly link conductance drift over time to changes in algorithmic performance associated with different shape parameters of the log-logistic fits.

To evaluate the robustness of the quantized models, we inject artificial noise into the weights of ANNs and CNN at inference time using a multiplicative noise strategy [11]. For each quantization level i, we apply multiplicative noise independently to all weights w assigned to that level (bin_i). We draw random samples from a log-logistic distribution parameterized by the shape c_{LRS_i} and scale s_{LRS_i} obtained from experimental conductance fits. Let

$$r \sim \text{log-logistic}(c_{LRS_i}, s_{LRS_i}), \tag{1}$$

denote a random variable sampled from this distribution. We normalize the samples by the scale parameter,

$$m = \frac{r}{s_{LRS_i}}, \tag{2}$$

which yields a dimensionless quantity. We then center and rescale the normalized samples to enforce zero mean and unit variance,

$$\hat{r} = \frac{m - \bar{m}}{\text{std}(m)}, \tag{3}$$

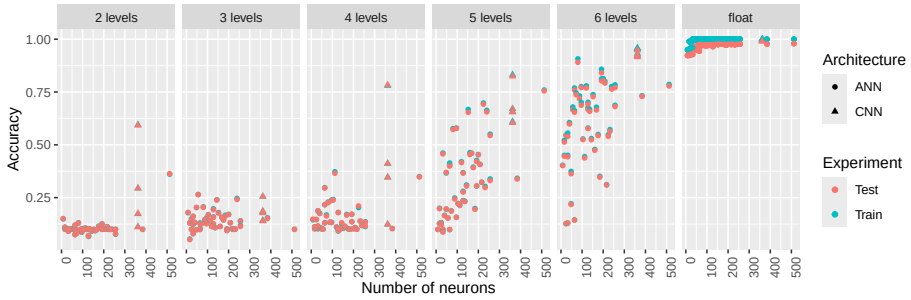
where $\bar{m}$ denotes the mean of m. We construct the multiplicative perturbation factor as

$$\alpha = 1 + \lambda \hat{r}, \tag{4}$$

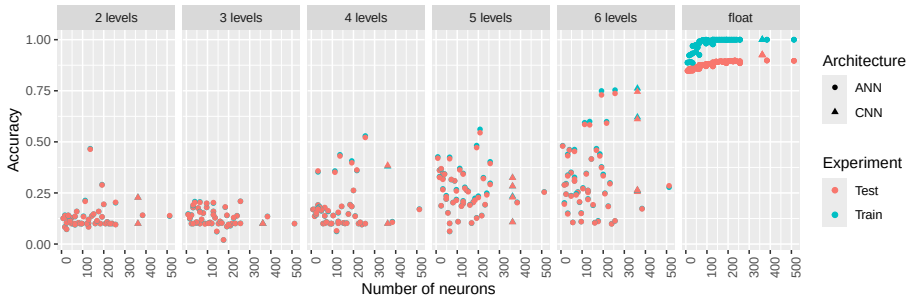
with λ controlling the noise amplitude. We evaluate several noise strengths, using $\lambda = 1, 3, 5$, and 10. Finally, we obtain the noisy weight as

$$\hat{w} = w \cdot \alpha. \tag{5}$$

This formulation centers the perturbation around unity, preserves the sign and relative magnitude of the quantized weights, and introduces variability that follows the same statistics observed in experimentally measured conductance fluctuations and drift.



(a) MNIST



(b) Fashion-MNIST

Fig. 1: Training and test accuracy as a function of the total number of neurons under different weight quantization resolutions. Results are shown for 2-6 quantization levels and compared with the full-precision (float) baseline.

4 Results and Discussion

Overall Performance Trends under Quantization. We can infer from the observed results a clear and systematic effect of weight quantization on ANN performance for both MNIST and Fashion-MNIST datasets. As the number of quantization levels decreases, both training and test accuracy drop relative to the full-precision baseline. The magnitude of this drop varies depending on the network architecture and on the specific choice of quantization levels (Fig. 1). At higher resolutions (5-6 levels) several quantized configurations retain acceptable accuracy and in some cases approach full-precision performance. As a consequence, networks can tolerate a substantial reduction in numerical precision, as long as we select an appropriate mapping between weights and quantization levels. Extremely low resolutions, by contrast, fail to reach competitive accuracy regardless of the selected configuration. On Fashion-MNIST dataset, we also observe mild overfitting (Fig. 1(b)), which we attribute to the deliberately oversized architectures used to probe noise sensitivity across configurations. Our goal here is to measure how noise impacts on the performance of the ANN, rather than

achieving the best accuracy. The CNN showed slightly better performance than the full ANNs in some configurations across both datasets.

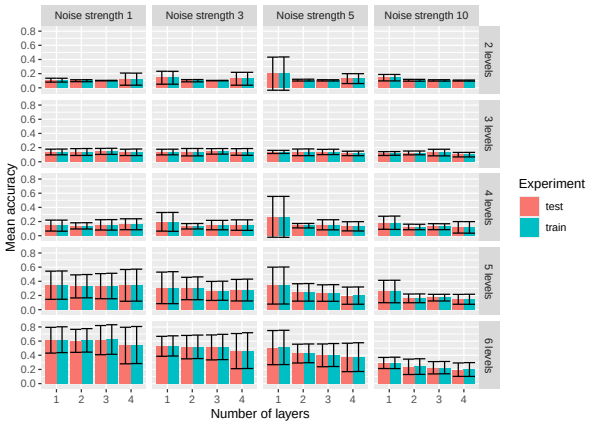
Effect of Noise in Quantization Level Combinations. Quantization at a fixed resolution does not define a single model. Because multiple experimentally measured LRS states are available, each resolution admits several valid configurations. We initially expected that configurations spanning a wider conductance range would perform better than those clustered within a narrow region. But, different LRS combinations at the same resolution lead to different performance variations. Statistical analysis indicates that these differences are not significant (Kruskal–Wallis $\chi^2 = 1.5258$, $p = 0.9101$ for the ANN experiments).

Impact of Noise Injection. When we introduce noise derived from experimentally fitted log-logistic distributions, the experiments show different outcomes in performance. Higher noise strengths and lower quantization resolutions typically reduce peak accuracy (Fig. 2). At the same time, noise does not always negatively affect the network, in several architectures and quantization regimes, noisy models match or slightly exceed the accuracy of their noise-free counterparts. This effect appears at moderate noise levels, for example when comparing noise-free models with those using $\lambda = 1$ (Fig. 3). This is consistent with what one would expect from implicit regularization: the noise prevents the network from committing too strongly to the trained weights. On the one hand, deeper architectures respond more strongly to aggressive quantization and noise, likely because perturbations affect a larger number of parameters. On the other hand, gains in robustness do not scale linearly with size. Beyond a certain point in ANNs, adding neurons only reduces accuracy, especially under strong noise. However, CNNs tend to exhibit a superior robustness to noise than ANNs.

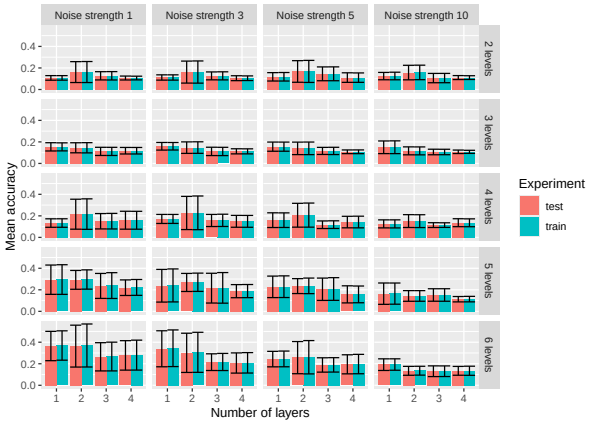
Temporal Effects and Post-Programming Drift. We track performance across the four time points studied (AS, 10 min, 60 min, and 24 h) and observe that post-programming drift introduces additional, yet structured, variability. We use the AS condition as a baseline, while later checkpoints reflect the gradual redistribution of conductance values observed experimentally (Fig. 4). To determine whether performance evolves over time, we apply a Friedman test to the best-performing LRS configuration at each quantization level, ANN architecture, and time point. The analysis of ANNs shows no statistically significant differences across time points, both for training ($\chi^2 = 1.78$, $p = 0.620$) and test ($\chi^2 = 1.28$, $p = 0.735$). These results indicate that, once we select the optimal configuration for a given setting, performance remains largely stable across different LRS combinations despite temporal drift.

5 Conclusions and Future Work

In this work we develop a hardware-aware framework to evaluate ANNs that explicitly accounts for non-ideal effects measured in memristive devices. We con-



(a) MNIST

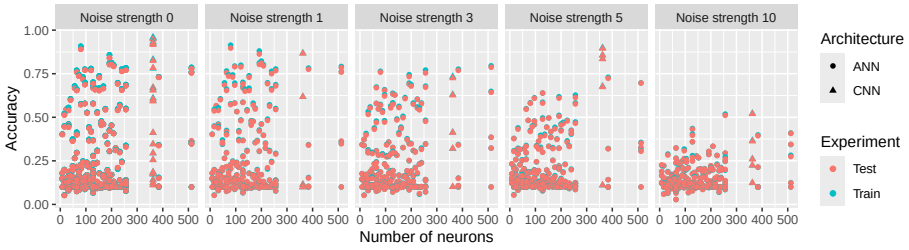


(b) Fashion-MNIST

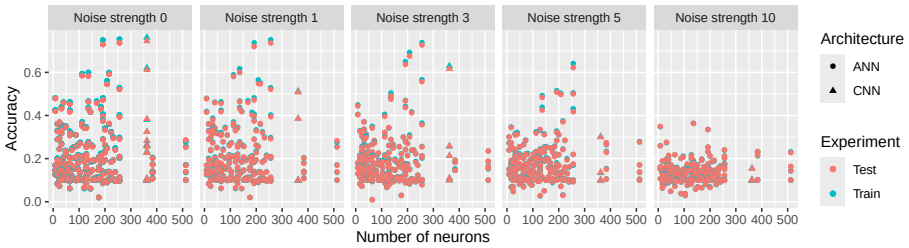
Fig. 2: Mean training and test accuracy as a function of the number of layers under noise at the AS condition using the best LRS combination. Results are shown for different quantization levels (2-6 levels) and noise strengths, with accuracies averaged including an error bar.

nect device-level statistical characterization with algorithm-level analysis, which lets us examine artificial neural networks behavior under physical constraints that standard software-only evaluations typically overlook. Although the experimental setting used does not capture the full behavior of modern deep learning models, it allows for a systematic analysis of variability-induced perturbations and establishes a baseline for future work.

Our results show that aligning weight quantization with experimentally available conductance states allows a substantial reduction in numerical precision



(a) MNIST



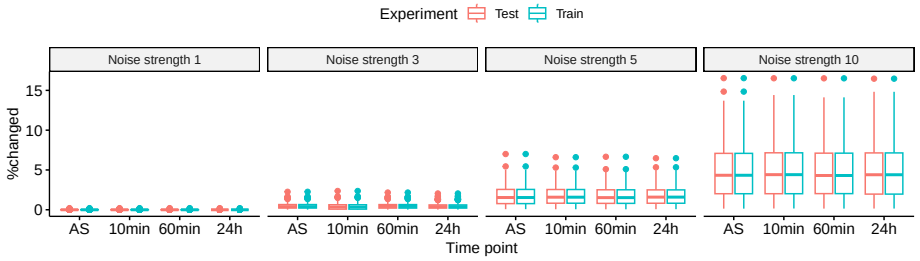
(b) Fashion-MNIST

Fig. 3: Training and test accuracy as a function of the total number of neurons under quantization with additive and multiplicative noise at the AS condition. Results are shown for increasing noise strengths, including the noise-free case, illustrating the combined effect of quantization and device-inspired noise on ANN performance.

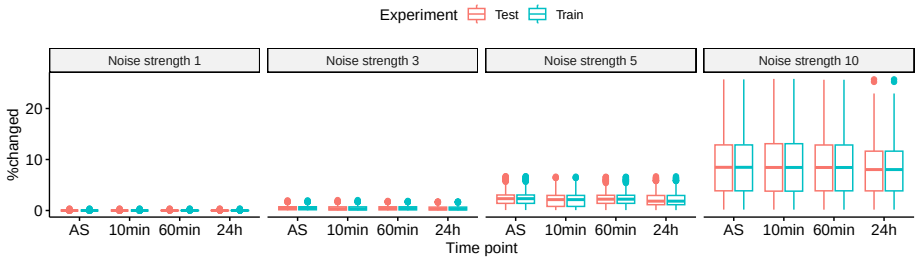
while preserving acceptable accuracy at moderate resolutions. By relying on log-logistic noise models extracted from previous work, we capture device variability more faithfully than with the Gaussian assumptions commonly used in the literature. The noise injection process leads to different performance outcomes. Excessive variability degrades performance, but moderate noise can improve generalization in some configurations, acting much like an implicit regularizer. Temporal drift, on its own, does not introduce statistically significant performance loss in appropriate quantization mapping, which suggests that stable inference does not require frequent recalibration.

These observations point toward the need to design hardware and algorithms together, rather than treating device behavior as post-design correction step. In particular, the results suggest that device characteristics such as quantization levels, variability, and drift dynamics should be explicitly incorporated during the algorithmic design phase, influencing choices such as network architecture, precision, and training strategies. Realistic physical models matter when targeting memristive and neuromorphic platforms.

In the future, we plan to extend this line of work examining more complex architectures in detail to probe scalability. We also plan to introduce quantization



(a) MNIST



(b) Fashion-MNIST

Fig. 4: Distribution of training and test accuracy across post-programming time points (AS, 10 min, 60 min, and 24 h) for different noise strengths.

and noise during training instead of restricting to inference. Finally, we will study other device technologies and variability models to test how broadly this framework applies beyond HfO_2-based systems.

Acknowledgments. This study was funded by Consejería de Universidad, Investigación e Innovación de la Junta de Andalucía (`DGP_PIDI_2024_00563`).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bishop, C.M.: Training with noise is equivalent to Tikhonov regularization. Neural Computation **7**(1), 108–116 (1995)
2. Esmanhotto, E., Hirtzlin, T., Bonnet, D., Castellani, N., Portal, J.M., Querlioz, D., Vianello, E.: Experimental demonstration of multilevel resistive random access memory programming for up to two months stable neural networks inference accuracy. Advanced Intelligent Systems **4**(11), 2200145 (2022)
3. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M.W., Keutzer, K.: A survey of quantization methods for efficient neural network inference. arXiv preprint arXiv:2103.13630 (2021)

4. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: Training neural networks with low precision weights and activations. Journal of Machine Learning Research **18**(187), 1–30 (2017)
5. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2704–2713 (2018)
6. Lanza, M., Wong, H.S., Pop, E., Ielmini, D., Strukov, D., Regan, B., Larcher, L.: Recommended methods to study resistive switching devices. Advanced Electronic Materials **5**(1), 1800143 (2019)
7. LeCun, Y., Cortes, C., Burges, C.J.: The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/ (1998)
8. Maldonado, D., Acal, C., Ortiz, H., Navas-Gomez, F., Cantudo, A., Wenger, C., Pérez, E., Roldán, J.: Variability in HfO_2-based memristors under pulse operation. Microelectronic Engineering **304**, 112445 (2026)
9. Maldonado, D., Baroni, A., Aldana, S., Pérez, E.: Kinetic Monte Carlo simulation analysis of the conductance drift in multilevel HfO_2-based RRAM devices. Nanoscale **16**(40), 19021–19033 (2024)
10. Maldonado, D., Acal, C., Ortiz, H., Aguilera, A.M., Ruiz-Castro, J.E., Cantudo, A., Baroni, A., Dorai Swamy Reddy, K., Pechmann, S., Uhlmann, M., Wenger, C., Pérez, E., Roldán, J.B.: A comprehensive statistical study of the post-programming conductance drift in HfO_2-based memristive devices. Materials Science in Semiconductor Processing **196**, 109668 (2025)
11. Nalisnick, E., Anandkumar, A., Smyth, P.: A scale mixture perspective of multiplicative noise in neural networks. arXiv preprint arXiv:1506.03208 (2015)
12. Prezioso, M., Merrikh-Bayat, F., Hoskins, B., Adam, G.C., Likharev, K.K., Strukov, D.B.: Training and operation of an integrated neuromorphic network based on metal-oxide memristors. Nature **521**, 61–64 (2015)
13. Roldán, J.B., Maldonado, D., Aguilera-Pedregosa, C., Alonso, F.J., Xiao, Y., Shen, Y., Zheng, W., Yuan, Y., Lanza, M.: Modeling the variability of au/ti/h-bn/au memristive devices. IEEE Transactions on Electron Devices (2022)
14. Roldán, J.B., Miranda, E., Maldonado, D., Mikhaylov, A.N., Agudov, N.V., Dubkov, A.A., et al.: Variability in resistive memories. Advanced Intelligent Systems **5**, 2200338 (2023)
15. Romero-Zaliz, R., Cantudo, A., Pérez, E., Jimenez-Molinos, F., Wenger, C., Roldan, J.B.: An analysis on the architecture and the size of quantized hardware neural networks based on memristors. Electronics **10**(24), 3141 (2021)
16. Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R., Eleftheriou, E.: Memory devices and applications for in-memory computing. Nature Nanotechnology **15**, 529–544 (2020)
17. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
18. Yao, P., Wu, H., Gao, B., Tang, J., Zhang, Q., Zhang, W., Yang, J.J., Qian, H.: Fully hardware-implemented memristor convolutional neural network. Nature **577**(7792), 641–646 (2020)
19. Yousuf, O., Hossen, I., Daniels, M.W., Lueker-Boden, M., Dienstfrey, A.M., Adam, G.C.: Device modeling bias in ReRAM-based neural network simulations. IEEE Journal on Emerging and Selected Topics in Circuits and Systems **13**(1), 382–394 (2023)