Designing and Training Neural Networks for Analog In-Sensor Deployment: A Hardware-Aware Analysis

(Invited Paper)

[α]Mark Horton, [α]Haoxuan Shan, [α]James Kiessling, [β]Huanrui Yang, [α]Yiran Chen, and [α]Hai "Helen" Li
[α]*Dept. of Electrical and Computer Engineering, Duke University, Durham, USA*
[β]*University of Arizona, Tucson, USA*
[α]{mark.horton,haoxuan.shan, james.kiessling, yiran.chen, hai.li}@duke.edu
[β]{huanruiyang}@arizona.edu

***Abstract*—Edge AI and IoT applications demand ultra-low latency and energy efficiency, but these goals are often undermined by the costs of digitizing and transmitting data. Analog in-sensor (AIS) hardware architectures address this bottleneck by enabling analog processing directly within the sensor, minimizing digitization and data movement. However, AIS deployments face key challenges including stringent power, performance, and area constraints, susceptibility to hardware-induced noise and variations, and accuracy degradation from operating on unprocessed sensor outputs rather than refined image data. We address these challenges through a software-driven, hardware-aware analysis that distills actionable design guidance for AIS-optimized convolutional neural networks (CNNs). Drawing on prior literature and our own empirical studies, we derive design recommendations for AIS-friendly network topologies, training recipes that jointly improve noise and quantization robustness, and strategies for effective learning from emulated raw sensor data without a digital image signal processing (ISP) pipeline. This analysis provides insight into hardware-aware software-based techniques that complement cutting-edge circuit and architecture-level approaches, helping advance the limits of high-performance AIS systems.**

***Index Terms*—analog integrated circuits, artificial neural networks, CMOS image sensors, convolutional neural networks, edge computing**

I. Introduction

Emerging autonomous vehicles, wearables, medical devices, Internet of Things (IoT) systems, and other edge devices increasingly demand real-time AI inference on high-bandwidth data streams captured at the sensor. Privacy, security, network reliability, bandwidth, and latency constraints often make server-based inference impractical. Even on-device solutions that pair sensors with local GPU accelerators can struggle to meet the stringent throughput, latency, and power budgets of these lightweight platforms. Highly optimized compute-in-memory (CIM) approaches alleviate some data-movement costs by colocating weight storage and multiply-accumulate (MAC) operations, but in vision workloads they still incur substantial overhead from high-precision digitization and activation transport. For example, in the ISAAC CIM architecture [1], analog-to-digital converters (ADCs) account for 58% of power and 31% of area, with the actual analog computing memristor array comprising a tiny fraction of remaining power and area.

This work was funded in part by National Science Foundation NSF 2112562, NSF 2233808, NSF 2332744, and DARPA Project W912CG25CA001. (Corresponding author: mark.horton@duke.edu)

This leads us to the AIS paradigm. AIS accelerators are neuromorphic in nature, embedding computation directly into the sensor fabric and processing analog activation signals locally without costly analog-to-digital and digital-to-analog conversions (DACs). By integrating computation into the analog sensor array itself, much like synaptic processing in biological nervous systems, AIS systems inherit the hallmark energy efficiency of neuromorphic architectures [2], achieving locality of computation that minimizes both data movement and conversion overhead.

Another useful lens for understanding AIS is to compare it with the more established CIM approach to CNN acceleration. CIM architectures co-locate weight storage and MAC operations in tightly integrated memory crossbars to minimize weight movement, embedding weights directly alongside compute units and exploiting Kirchhoff's and Ohm's laws to perform matrix–vector multiplications in place. AIS systems apply a similar locality principle, but with respect to activations: rather than moving captured pixel signals to a separate compute array, the MAC operations are relocated into the pixel array itself. This allows each photodiode's output to be multiplied and accumulated at the point of capture, substantially reducing activation transport and eliminating the need for early activation digitization, as illustrated in Figure 1. In this sense, AIS extends CIM concepts to address the challenges of end-to-end processing directly at the sensor level.

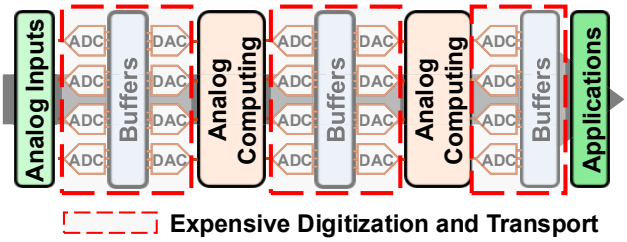
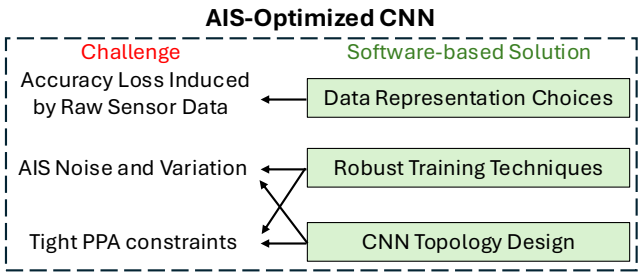


Fig. 1: AIS systems aim to minimize hardware costs by minimizing ADCs, DACs, buffers, and data buses.

In this work, we conduct a software and algorithm-focused analysis of CNN design for AIS deployment, complementing innovations in AIS circuits and architectures. We focus on the in-sensor image classification problem, a critical domain for AIS techniques and one where empirical studies can yield broadly applicable design insights. Drawing on prior work and our own experiments, we distill actionable guidance in three areas:

1) AIS-optimized network topology design,
2) training strategies that jointly improve robustness to noise



II. Background and Related Work

A. Analog In-Sensor Hardware

Although they all share some common design principles, a wide range of AIS CNN accelerators have been proposed. AnalogNet [3], built on the SCAMP-5 architecture [4], realizes a single in-sensor current-mode convolution layer with binarized outputs. [5] implements an in-sensor convolution operator with analog inputs and outputs using a conventional current-mode MAC design, paving the way for full-CNN implementations. The P^{2}M architecture [6] extends this direction with in-pixel weight storage, enabling computation both in-pixel and in-memory, and demonstrates multiple consecutive operations (convolution, batch normalization, ReLU) common in CNNs. Other works realize analog in-sensor operations with various emerging technologies including memristors [7], ferroelectrics [8], and spiking neurons [9]. Broadly, these approaches co-locate computation with the pixel array to apply locality to activations, thereby minimizing off-sensor data transport. We visualize a standard computer vision AIS pipeline in Figure 2.

B. Sensor Raw Image Processing

In AIS systems, the conventional ISP is often reduced or eliminated to save area and energy, sending raw sensor data directly to the CNN. While this design choice avoids costly digital processing, it also changes the statistical properties of the input data, which can degrade accuracy of CNNs trained and evaluated on this data.

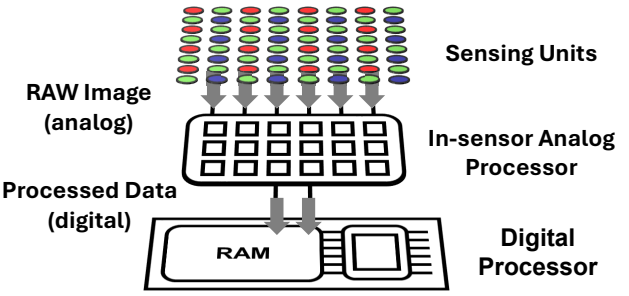


Fig. 2: An illustration of an integrated AIS vision system.

A conventional ISP converts raw Bayer sensor data into a display-ready RGB format through stages such as noise reduction, demosaicing, and color transformation from sensor RGB space to standard RGB. Modern ISPs, especially in mobile devices, have become increasingly sophisticated, but their high area and energy costs make them less attractive for AIS deployments [10]. "Vision mode" pipelines proposed in [10] remove the ISP entirely, streaming raw sensor outputs directly to downstream vision models, which is the scenario we consider here.

To emulate such pipelines, we generate RAW-format counterparts for our datasets using the Configurable and Reversible Imaging Pipeline (CRIP) [11]. CRIP supports bidirectional conversion between RAW and RGB formats and can selectively disable individual ISP stages. This flexibility allows us to isolate and quantify the accuracy impact of specific ISP components, enabling a controlled analysis of minimal-processing AIS pipelines. In Section IV-D, we apply this framework to evaluate CNN performance under different configurations of key ISP stages.

C. Analog Computing Noise and Variation

Unlike the deterministic and precise execution of digital processors, analog computing is inherently susceptible to various forms of variation and noise. The specifics of the noise and variation profile vary with device category, circuit design, and manufacturing process. These effects can manifest as noise which is correlated or uncorrelated, additive or multiplicative, or some combination thereof as described in [12]. Further, we may also consider weight noise rather than activation noise or non-Gaussian noise among other options.

Potential physical sources include thermal noise, shot noise, and flicker noise. Endurance-related issues such as temporal conductance drift can degrade accuracy over time, further challenging the long-term reliability of analog computations [13]. Fabrication imperfections introduce process variations, which together with voltage and temperature fluctuations (PVT), complicate precise analog operation.

Because many of these effects are hardware-dependent and may vary across devices and operating conditions, both circuit-level design and algorithm-level robustness must be considered. The latter can help ensure reliable outputs

despite the presence of noise, complementing hardware mitigation techniques.

D. Quantization and Sparsity

Quantization and pruning are two widely used techniques for compressing deep neural networks. Quantization maps each weight in a model to a lower precision representation. It is typically performed either through quantization-aware training (QAT), which simulates quantized inference during training using a straight-through estimator [14], or through post-training quantization (PTQ), which derives scaling factors from a small calibration dataset.

Pruning, in contrast, removes weights by setting them to zero so they no longer contribute to computation. This can be done in an unstructured manner, where individual weights are driven to zero during training [15], or in a structured manner, where entire filters, channels, or other weight groups are zeroed simultaneously [16].

While these techniques can improve efficiency in digital systems, they are critical for analog computing. Pruning may be used to deactivate analog components, resulting in power savings. Analog accelerators typically support much lower weight precision than GPU-based floating-point implementations. Quantization reduces PPA costs for storage, data movement, ADCs/DACs, and analog MAC units, depending on the circuit design. In addition, quantized models often show greater robustness to small noise perturbations due to reduced sensitivity to minor signal variations [17].

E. Robust Training

Robust training methods aim to produce models resilient to noise in inputs, weights, or labels, thereby improving stability under challenging conditions. One notable robust training method is Sharpness-Aware Minimization (SAM) [18], which perturbs weights along the gradient ascent direction to encourage flatter loss landscapes. Models trained with SAM show impressive generalization across a range of difficult tasks such as classification with noisy labels or adversarially perturbed images.

Robust training techniques have also been adapted to address the unique challenges of analog deep neural networks. For example, [19] proposes a noise-eliminating training scheme for memristor-based crossbar systems, and [20] introduces device-variation-aware training by profiling noise distributions specific to analog memory. [21] combines such device-aware noise profiling with SAM-based optimization to further enhance robustness in analog deployments. SAM is particularly appealing for analog neural networks because it improves robustness without requiring prior knowledge or explicit modeling of the hardware noise distribution.

HERO [17] extends SAM and reformulates quantization as a robustness problem, demonstrating that robustness-oriented training can improve quantized accuracy. Notably, under certain perturbation schemes, quantizing to lower bit precisions can even increase robustness [22].

F. Neural Architecture Search

Neural Architecture Search (NAS) is a powerful technique that automates the process of identifying high-performing neural network architectures that meet specific accuracy, efficiency, and resource constraints. Numerous NAS frameworks and algorithms have been developed to balance model accuracy with computational efficiency, targeting improvements in metrics such as model size, inference speed, and energy consumption [23], [24].

While widely used for digital platforms such as CPUs and GPUs, where metrics like FLOPs, parameter count, and inference time can be directly measured, applying NAS to analog accelerators introduces unique challenges. In analog systems, network accuracy can vary with noise distribution and magnitude, and architectures differ in their robustness to such perturbations [25]. In addition, modeling efficiency is more complex: metrics like power and area must include the cost of frequent digital-to-analog and analog-to-digital conversions, which add significant overhead even if they do not change MAC counts. Consequently, NAS for analog accelerators must be both variation-aware and hardware-aware [23].

Several recent works extend conventional NAS to incorporate hardware-specific factors. NACIM [24] performs cross-layer co-exploration of neural architecture, quantization, and hardware design for CIM systems, integrating area cost and device variation into the evaluation. AnalogNAS [13], targeting analog in-memory computing, expands the search space to include temporal drift and finds that wider networks are more robust to analog non-idealities. Interestingly, AnalogNAS also observes that optimal architectures may be dispersed in the search space, making local search strategies common in conventional NAS works less effective.

NAS principles are directly relevant to the topology and PPA trade-offs analyzed in Section IV-B and Section IV-C, where design choices influence both noise tolerance and hardware cost.

III. Methods

We design our experiments to evaluate algorithmic strategies for deploying convolutional neural networks (CNNs) in analog in-sensor (AIS) hardware, under realistic noise and power–performance–area (PPA) constraints.

A. Dataset and Preprocessing

We use the CIFAR-10 dataset [26] which contains 60,000 32×32 pixel RGB images across 10 classes, split into 50,000 training and 10,000 test samples. All images are normalized using the dataset mean and standard deviation. For experiments simulating raw sensor input Section IV-D, we generate RAW-format dataset variants using the Configurable and Reversible Imaging Pipeline (CRIP) [11] as described in Section II-B.

B. Model Architecture

Our baseline model is a modified VGG-13 CNN, chosen for its straightforward feed-forward topology and absence of skip connections, which are nontrivial to implement in AIS hardware. The architecture is adapted to replace standard ReLU activations with smooth softplus functions, more closely approximating analog-friendly nonlinearities achievable with saturated-transistor or diode-based circuits [27].

C. Noise Modeling

To represent analog non-idealities, we inject additive Gaussian activation noise at the output of each convolutional layer during evaluation. Unless otherwise stated, noise is drawn from $\mathcal{N}(0, \sigma^2)$, with σ^2 specified for each experiment. We find that our results generalize well across diverse noise types, including spatially correlated, uncorrelated, and multiplicative activation noise, though sensitivity to specific profiles can vary.

IV. Analysis of Algorithmic Strategies

A. Overview

We designed our experimental analysis to address three key challenges in deploying CNNs to analog in-sensor (AIS) hardware:

1) maintaining accuracy under hardware-induced noise and process variation,
2) meeting tight power–performance–area (PPA) constraints, and
3) operating effectively on minimally processed sensor outputs.

To study these challenges, we evaluate:

- how network topology influences both noise tolerance (Section IV-B) and PPA characteristics (Section IV-C),
- training strategies that jointly improve robustness to noise and quantization (Section IV-E), and
- data representation approaches for learning directly from raw sensor signals (Section IV-D).

The following subsections present empirical findings in each challenge area, linking algorithmic design choices to AIS-specific constraints.

B. Noise Tolerance and CNN Topology

Our experiments reveal that noise tolerance varies substantially across CNN topologies, and that (MAC) fan-in is a key factor. Using the NAS-Bench-101 search space [28], we identified a set of ten architectures with similar runtime and clean accuracy. Under additive Gaussian activation noise $N(0, 0.16)$, their relative clean accuracies correlated poorly with noisy accuracies (Pearson $r = 0.321$), indicating that clean performance is not a reliable predictor of robustness, and there is a non-trivial relationship between CNN topology and noise tolerance.

To isolate the role of fan-in, we compared a VGG-13 baseline to sparsely connected variants with each convolution output channel receiving input from only 50% or 25% of the baseline model's input channels. All models had approximately the same MAC count and parameter count. While clean accuracies were similar, lower fan-in variants degraded more rapidly as noise magnitude increased (Figure 3). This finding generalized across all tested noise profiles except pure multiplicative noise, under which the models saw similar performance degradation.

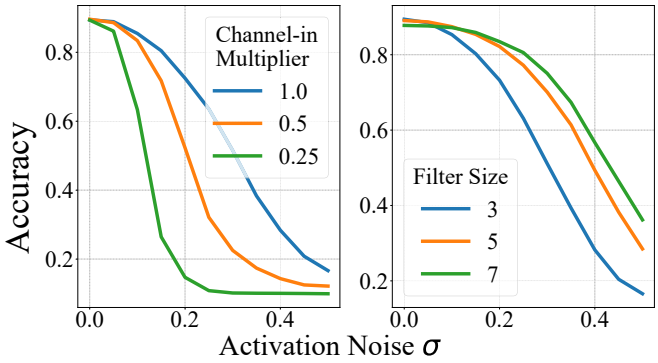


Fig. 3: Accuracy of architectures with varying MAC fan-in under different levels of activation noise. In our experiments, lower fan-in (e.g., via sparse connectivity or smaller filters) reduced noise tolerance, while higher fan-in moderated accuracy degradation.

We observed a complementary trend when increasing filter size from 3×3 to 5×5 and 7×7. Although larger filters slightly reduced clean accuracy, their higher fan-in slowed performance degradation under noise. This finding generalized across all tested noise profiles. Prior work [13] and our own experiments with MobileNet, an architecture comprised of very low fan-in depthwise separable convolutions, further confirm that architectures with small fan-in convolutions suffer disproportionately from noise and variation.

C. CNN Topology and PPA Trends

Convolutional neural network (CNN) topologies interact with AIS hardware constraints in ways that differ significantly from conventional digital deployments. A key observation is the distinction between spatial-bound and channel-bound regions of the network. Early layers typically operate on large activation maps with relatively few channels (spatial-bound), while deeper layers operate on smaller spatial dimensions but more channels (channel-bound). Figure 4 illustrates this trend for a simple CNN with 3×3 stride-1 convolutions, doubling channels at each downsampling stage. Under this very common channel growth paradigm, although the number of MAC operations per stage remains constant, activation counts shrink and parameter counts grow with depth.

AIS systems tend to have a comparative advantage in spatial-bound regions. Large activation maps incur minimal additional cost in AIS hardware because activations remain

Feature Map Size	32x32x16	16x16x32	8x8x64
Activations (x0.5)	16384	8192	4096

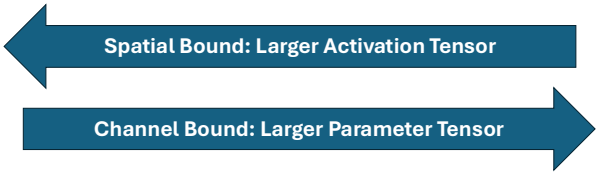


Fig. 4: We analyze 3×3 stride-1 convolutions in a simple CNN that doubles channels at each downsampling stage. While the number of MACs remains constant across depths, activation sizes decrease and parameter counts increase in deeper layers.

in the analog domain, avoiding expensive digitization and transport. In contrast, the channel-bound region can be more costly: parameters must either be stored locally in the pixel array or fetched from off-sensor memory, both of which can significantly impact PPA.

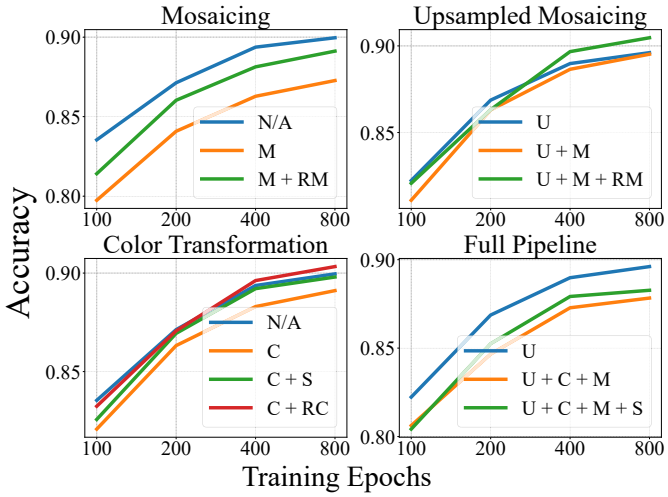
These observations suggest several design implications:

- Front-loading computation: Increasing early-stage capacity via slower channel growth, additional layers before downsampling, or early global average pooling can exploit AIS advantages in spatial-bound regions while reducing later-stage parameter costs.
- Kernel size trade-offs: As shown in Section IV-B, larger kernels can improve noise tolerance, but kernel sizes above 3×3 generally offer limited accuracy gains under clean conditions while incurring greater analog routing costs.
- Grouped and depthwise convolutions: Many popular efficient architectures like MobileNet [29] and EfficientNet [30] employ lower fan-in MACs that may be attractive for PPA, but their reduced noise averaging could diminish robustness in AIS deployments.

Taken together, these trends highlight an inherent trade-off: designs that minimize PPA in AIS hardware may also reduce tolerance to analog noise and process variation. A neural architecture search (NAS) framework tailored to AIS constraints can help navigate this trade-off by jointly optimizing accuracy, robustness, and PPA metrics. Such a search space should consider channel growth rate, skip connection usage, kernel sizes, grouped/depthwise convolution, and weight precision, potentially alongside hardware parameters such as allowable on-sensor storage and interconnect costs. While our present findings provide preliminary guidance, a comprehensive AIS-aware NAS study remains an important direction for future work.

D. Sensor Raw Data Representation

Using CRIP [11], we train and evaluate on estimated raw sensor images corresponding to CIFAR-10, focusing on two components of the raw image pipeline: mosaicing and color



Code	Processing step
N/A	No transformation
U	$2\times$ bilinear upsampling
C	Color transform (CIFAR-10 RGB $\rightarrow$ sensor RGB)
RC	Reverse color transform (sensor RGB $\rightarrow$ CIFAR-10 RGB)
S	Square root per channel
M	Mosaicing (Bayer pattern)
RM	Reverse mosaicing

Fig. 5: Accuracy impact of different raw image processing pipeline variations. Letter codes indicate processing steps (defined in the table above) applied to CIFAR-10 images before training. Images for each variation are shown in Figure 6.

transformation visualized in Figure 6. As shown in Figure 5, training on fully processed images consistently achieves higher accuracy than on their raw sensor counterparts, or on variants with only the color transform or only the mosaicing applied. The color-transformed images can be reversed losslessly back to the processed form, achieving equivalent accuracy (Figure 5). However, applying a per-channel square root to the color-transformed inputs partially recovers accuracy without reversing the transform, suggesting that expanding the dynamic range near zero improves learnability by making low-intensity variations more separable. This type of nonlinearity is also well-suited to efficient analog implementation.

Mosaicing, in contrast, produces accuracy losses that cannot be fully reversed by demosaicing (M vs. M+RM in Figure 5), consistent with information loss from spatial subsampling. We find that oversampling the image by $2\times$ in each dimension before mosaicing, then average-pooling back to 32×32 immediately before training, substantially reduces the loss relative to mosaicing at the final resolution (compare U vs. U+M and U+M+RM in Figure 5). This indicates that a Bayer grid matched exactly to the target resolution contains less recoverable spatial information than an RGB image of

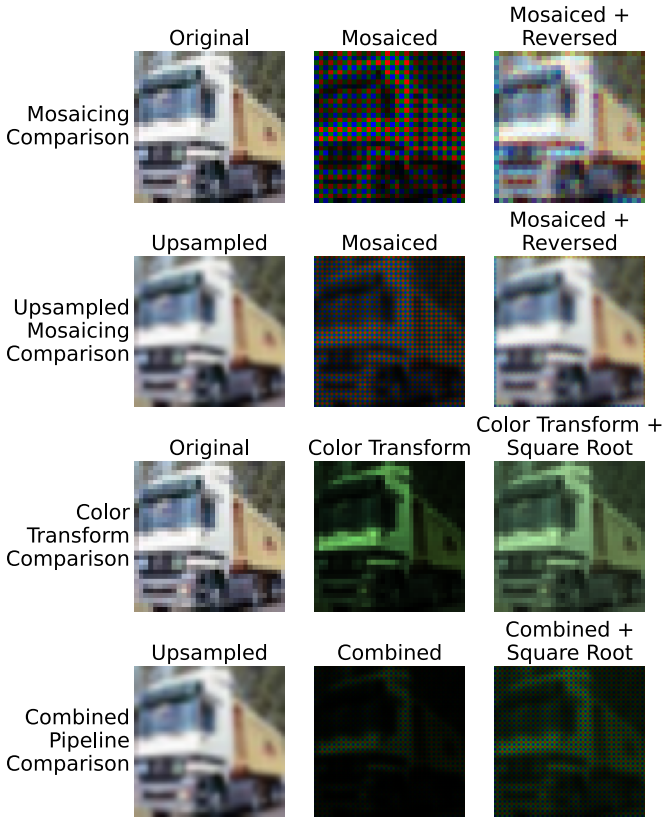


Fig. 6: CIFAR-10 image shown after various processing stages. "Mosaiced" applies a Bayer pattern to a processed image. "Color Transform" remaps visual RGB values to their sensor-space equivalents. "Combined" applies the color transform followed by mosaicing. "Square Root" applies a per-channel square root to pixel intensities.

the same size, and that expanding the number of sensors in the Bayer pattern can help preserve fine-scale cues for learning.

These results suggest that AIS systems may benefit from lightweight analog transformations that increase near-zero dynamic range, and from provisioning a pixel sensor density above the desired output resolution when PPA budgets allow.

E. Robust Training with SAM

We find that training with SAM yields a dual benefit in AIS settings, as illustrated in Figure 7. First, SAM markedly reduces accuracy degradation across a range of noise levels. We found that SAM produced robustness gains against all tested noise profiles despite the absence of any noise-model-specific tuning. This broad applicability stems from SAM's formulation: by explicitly seeking parameter updates that minimize loss under an adversarial, worst-case perturbation of the weights, SAM implicitly prepares the network to withstand a wide spectrum of perturbations, including those not seen during training. As a result, SAM-trained models are well-suited for deployment across

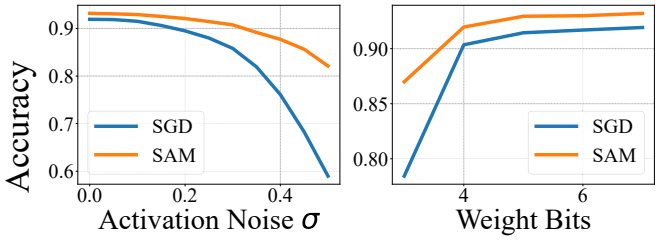


Fig. 7: SAM fine-tuning with a magnitude 0.5 perturbation improves tolerance to both activation noise and low-bit-precision quantization compared to an SGD-trained baseline.

heterogeneous AIS hardware where precise noise characteristics may be unknown or may vary over time. Second, consistent with the observations in HERO [17], SAM-trained networks also exhibit reduced sensitivity to lower bit precision quantization. In AIS deployments, this combination of tolerance to both noise and reduced precision is particularly advantageous: in many MAC implementations, it enables the simultaneous lowering of weight precision and acceptance of higher intrinsic noise without significant accuracy loss. These properties directly translate into reductions in MAC area and power consumption, aligning with the stringent PPA requirements of AIS systems.

V. Discussion and Future Work

As AIS circuits and architectures advance, algorithmic techniques remain a powerful lever for improving end-to-end performance. We align network topology with AIS cost profiles, employ robustness-oriented training (such as SAM), and select sensor data representations that expand dynamic range near zero while preserving fine-scale cues. Together, these strategies enable higher accuracy and lower hardware costs in edge vision workloads.

Looking ahead, we expect significant performance gains from AIS-aware neural architecture search that explicitly incorporates robustness, quantization tolerance, and hardware costs into its optimization objectives. By seamlessly integrating hardware and neural network design choices into a search space, we can pursue optimal holistic trade-offs in AIS systems. We also see promising opportunities to incorporate on-device, variation-aware fine-tuning, enabling us to reduce model size and cost without compromising robustness. Finally, we may achieve further gains by optimizing the ISP pipeline under the constraints of efficient analog computation, ensuring it maximally supports downstream neural network accuracy.

While our study focuses on computer vision, these principles extend to other sensing modalities where computations must tolerate noise, quantization, and tight PPA budgets. We hope these findings inspire deeper integration of sensor hardware and algorithmic adaptation in the next generation of on-sensor intelligence.

REFERENCES

[1] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," ACM SIGARCH Computer Architecture News, vol. 44, no. 3, pp. 14–26, 2016.

[2] X. Liu, M. Mao, B. Liu, B. Li, Y. Wang, H. Jiang, M. Barnell, Q. Wu, J. Yang, H. Li et al., "Harmonica: A framework of heterogeneous computing systems with memristor-based neuromorphic computing accelerators," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 63, no. 5, pp. 617–628, 2016.

[3] M. Z. Wong, B. Guillard, R. Murai, S. Saeedi, and P. H. Kelly, "Analognet: Convolutional neural network inference on analog focal plane sensor processors," arXiv preprint arXiv:2006.01765, 2020.

[4] S. J. Carey, A. Lopich, D. R. Barr, B. Wang, and P. Dudek, "A 100,000 fps vision sensor with embedded 535gops/w 256× 256 simd processor array," in 2013 symposium on VLSI circuits. IEEE, 2013, pp. C182–C183.

[5] J. Zhu, B. Chen, Z. Yang, L. Meng, and T. T. Ye, "Analog circuit implementation of neural networks for in-sensor computing," in 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). IEEE, 2021, pp. 150–156.

[6] G. Datta, S. Kundu, Z. Yin, R. T. Lakkireddy, J. Mathai, A. P. Jacob, P. A. Beerel, and A. R. Jaiswal, "A processing-in-pixel-in-memory paradigm for resource-constrained tinyml applications," Scientific Reports, vol. 12, no. 1, p. 14396, 2022.

[7] B. Dang, T. Zhang, X. Wu, K. Liu, R. Huang, and Y. Yang, "Reconfigurable in-sensor processing based on a multi-phototransistor–one-memristor array," Nature Electronics, vol. 7, no. 11, pp. 991–1003, 2024.

[8] Y. Wang, Y. Cai, F. Wang, J. Yang, T. Yan, S. Li, Z. Wu, X. Zhan, K. Xu, J. He et al., "A three-dimensional neuromorphic photosensor array for nonvolatile in-sensor computing," Nano Letters, vol. 23, no. 10, pp. 4524–4532, 2023.

[9] Z. Li, Q. Zheng, Y. Chen, and H. Li, "Spikesen: Low-latency in-sensor-intelligence design with neuromorphic spiking neurons," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 70, no. 6, pp. 1876–1880, 2023.

[10] A. Ignatov, L. Van Gool, and R. Timofte, "Replacing mobile camera isp with a single deep learning model," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 536–537.

[11] M. Buckler, S. Jayasuriya, and A. Sampson, "Reconfiguring the imaging pipeline for computer vision," in The IEEE International Conference on Computer Vision (ICCV), 2017.

[12] N. Semenova, L. Larger, and D. Brunner, "Understanding and mitigating noise in trained deep neural networks," Neural Networks, vol. 146, pp. 151–160, 2022.

[13] H. Benmeziane, C. Lammie, I. Boybat, M. J. Rasch, M. L. Gallo, H. Tsai, R. Muralidhar, S. Niar, H. Ouarnoughi, V. Narayanan, A. Sebastian, and K. E. Maghraoui, "Analognas: A neural network design framework for accurate inference with analog in-memory computing," in IEEE International Conference on Edge Computing and Communications, EDGE 2023, Chicago, IL, USA, July 2-8, 2023, C. A. Ardagna, F. M. Awaysheh, H. Bian, C. K. Chang, R. N. Chang, F. C. Delicato, N. Desai, J. Fan, G. C. Fox, A. Goscinski, Z. Jin, A. Kobusinska, and O. F. Rana, Eds. IEEE, 2023, pp. 233–244. [Online]. Available: https://doi.org/10.1109/EDGE60047.2023.00045

[14] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013. [Online]. Available: https://arxiv.org/abs/1308.3432

[15] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in Advances in Neural Information Processing Systems, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf

[16] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in Advances in Neural Information Processing Systems, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/41bfd20a38bb1b0bec75acf0845530a7-Paper.pdf

[17] H. Yang, X. Yang, N. Z. Gong, and Y. Chen, "HERO: hessian-enhanced robust optimization for unifying and improving generalization and quantization performance," in DAC '22: 59th ACM/IEEE Design Automation Conference, San Francisco, California, USA, July 10 - 14, 2022, R. Oshana, Ed. ACM, 2022, pp. 25–30. [Online]. Available: https://doi.org/10.1145/3489517.3530678

[18] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," CoRR, vol. abs/2010.01412, 2020. [Online]. Available: https://arxiv.org/abs/2010.01412

[19] B. Liu, M. Hu, H. Li, Z.-H. Mao, Y. Chen, T. Huang, and W. Zhang, "Digital-assisted noise-eliminating training for memristor crossbar-based analog neuromorphic computing engine," in Proceedings of the 50th Annual Design Automation Conference, 2013, pp. 1–6.

[20] Y. Long, X. She, and S. Mukhopadhyay, "Design of reliable DNN accelerator with un-reliable reram," in Design, Automation & Test in Europe Conference & Exhibition, DATE 2019, Florence, Italy, March 25-29, 2019, J. Teich and F. Fummi, Eds. IEEE, 2019, pp. 1769–1774. [Online]. Available: https://doi.org/10.23919/DATE.2019.8715178

[21] C. Park, J. Jeon, and H. Cho, "Dat: Leveraging device-specific noise for efficient and robust ai training in reram-based systems," in 2023 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), 2023, pp. 289–292.

[22] M. Gorsline, J. Smith, and C. Merkel, "On the adversarial robustness of quantized neural networks," in Proceedings of the 2021 Great Lakes Symposium on VLSI, ser. GLSVLSI '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 189–194. [Online]. Available: https://doi.org/10.1145/3453688.3461755

[23] H. Benmeziane, K. E. Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, and N. Wang, "Hardware-aware neural architecture search: Survey and taxonomy," in Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, Z. Zhou, Ed. ijcai.org, 2021, pp. 4322–4329. [Online]. Available: https://doi.org/10.24963/ijcai.2021/592

[24] W. Jiang, Q. Lou, Z. Yan, L. Yang, J. Hu, X. S. Hu, and Y. Shi, "Device-circuit-architecture co-exploration for computing-in-memory neural accelerators," IEEE Trans. Computers, vol. 70, no. 4, pp. 595–605, 2021. [Online]. Available: https://doi.org/10.1109/TC.2020.2991575

[25] N. Ye, L. Cao, L. Yang et al., "Improving the robustness of analog deep neural networks through a bayes-optimized noise injection approach," Communications Engineering, vol. 2, p. 25, 2023. [Online]. Available: https://doi.org/10.1038/s44172-023-00074-3

[26] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[27] A. Ananthakrishnan and M. G. Allen, "All-passive hardware implementation of multilayer perceptron classifiers," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 9, pp. 4086–4095, 2020.

[28] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter, "Nas-bench-101: Towards reproducible neural architecture search," in International conference on machine learning. PMLR, 2019, pp. 7105–7114.

[29] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[30] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in International conference on machine learning. PMLR, 2019, pp. 6105–6114.