Variance-Aware Noisy Training: Hardening DNNs against Unstable Analog Computations

Xiao Wang (✉), Hendrik Borras, Bernhard Klein, and Holger Fröning

Hardware and Artificial Intelligence Lab, Institute of Computer Engineering,
Heidelberg University, Germany
{xiao.wang,hendrik.borras,bernhard.klein,holger.froening}@ziti.uni-heidelberg.de

Abstract. The disparity between the computational demands of deep learning and the capabilities of compute hardware is expanding drastically. Although deep learning achieves remarkable performance in countless tasks, its escalating requirements for computational power and energy consumption surpass the sustainable limits of even specialized neural processing units, including the Apple Neural Engine and NVIDIA TensorCores. This challenge is intensified by the slowdown in CMOS scaling.

Analog computing presents a promising alternative, offering substantial improvements in energy efficiency by directly manipulating physical quantities such as current, voltage, charge, or photons. However, it is inherently vulnerable to manufacturing variations, nonlinearities, and noise, leading to degraded prediction accuracy. One of the most effective techniques for enhancing robustness, Noisy Training, introduces noise during the training phase to reinforce the model against disturbances encountered during inference. Although highly effective, its performance degrades in real-world environments where noise characteristics fluctuate due to external factors such as temperature variations and temporal drift.

This study underscores the necessity of Noisy Training while revealing its fundamental limitations in the presence of dynamic noise. To address these challenges, we propose Variance-Aware Noisy Training, a novel approach that mitigates performance degradation by incorporating noise schedules which emulate the evolving noise conditions encountered during inference. Our method substantially improves model robustness, without training overhead. Through experiments on image classification tasks in dynamic noise environments, we demonstrate a significant increase in robustness, from 79.3% with conventional Noisy Training to 97.6% with Variance-Aware Noisy Training on CIFAR-10 and from 32.4% to 99.7% on Tiny ImageNet.

Keywords: noisy training · noisy computations · analog computing · robustness · neural networks.

1 Introduction

Deep neural networks (DNNs) have driven remarkable advancements in a wide array of machine learning applications, from computer vision and natural language, speech and signal processing. These breakthroughs are largely enabled by digital compute platforms, such as graphics processing units (GPUs) or specialized accelerators, which offer high throughput and flexibility. However, as DNNs grow in scale and are increasingly deployed in energy-constrained environments, the quest for more efficient hardware solutions becomes paramount. In addition, Complementary Metal-Oxide-Semiconductor (CMOS) technology scaling is stuttering, thus alternative approaches to maintain performance scaling have to be found. Analog computing architectures replace discrete quantities with continuous ones, leveraging the inherent properties of physical systems to perform computations efficiently. These architectures enhance computational efficiency, reduce data movement overhead, and enable highly parallel multiply-accumulate (MAC) operations, while their most significant advantage lies in superior energy efficiency [8,36].

Analog accelerators leverage the intrinsic physical properties of existing and emerging device technologies—such as analog CMOS-based computing [34], photonic computing [11], resistive random-access memory (ReRAM), phase-change memory (PCM), and other non-volatile devices—to perform approximate MAC operations based on physical quantities such as charge, even directly in the memory array [41]. These architectures can significantly cut down power consumption and latency, surpassing many of their digital counterparts [6,1]. However, these advantages come at the cost of increased susceptibility to analog non-idealities and noise. Factors such as device variations, thermal fluctuations, mismatch, drift, and aging can degrade both the performance and reliability of analog DNN implementations [36,14,35].

From a machine learning perspective, various works have reported that adding small amounts of noise to the training data can improve generalization [2,15,22], thus acting as a form of data augmentation. Noise injection is often also referred to as "distortion" or "jitter", in particular in early works. Besides injecting such (usually Gaussian) noise to input variables, there are similar methods on adding noise to other parts of a neural architecture, including weights [15], gradients [32] and activations [18]. However, this applies only to small noise levels and is limited to training, while inference remains noise-free. Thus analog hardware noise has the potential to distort intermediate activations and weights, undermining the model's inference accuracy if left unmitigated. Consequently, there is substantial interest in techniques that preserve DNN accuracy under noise. One of the most widely studied methods is Noisy Training, where noise is intentionally injected during the training process to emulate the hardware imperfections encountered during inference [43,20]. Exposure to noise from the outset enables the model to adapt its parameters, enhancing robustness to real-world noise variations. From a broader perspective, there are adjacent works in multiple directions: on adversarial effects to improve the robustness of DNNs [38], on Noisy Training to

introduce sparsity in the activation space [4], as well as on using noise in physical computations as a source of stochasticity [5].

While Noisy Training has been shown to be successful in various use cases, it is important to note that its efficacy strongly depends on the fidelity with which one can replicate the true hardware noise characteristics during training. When there is a mismatch—e.g., in distribution (statistical shape), amplitude (magnitude), or temporal correlation (noise in real hardware sometimes changes over time)—between training noise and inference noise, DNNs often fail to generalize the learned robustness and may suffer a decrease in prediction quality. It is important to emphasize that in analog devices, maintaining a constant noise level is highly uncommon. Even under controlled laboratory conditions, stabilizing noise over time presents significant challenges. Environmental factors such as temperature fluctuations, electromagnetic induction, and various timing effects inherently influence noise characteristics. Given these dynamic influences, it is reasonable to assume that noise in analog hardware is not static but evolves over the device's operational lifetime. This variability underscores the necessity of considering time-dependent noise models when designing robust deep learning systems for analog accelerators. This observation raises an important research question: How precisely must one capture the analog hardware's noise characteristics during training to ensure robust inference? Addressing this question is non-trivial, given that the noise profiles in analog circuits can evolve over time due to changing environmental conditions, device aging, or even variations in operating modes.

A second challenge arises when perfectly matching the real hardware noise in training is either impractical or impossible. While techniques like hardware-in-the-loop training can provide more accurate noise profiles [33], they may be expensive or time-consuming to implement. In practical implementations, only approximations or partial knowledge of noise statistics may be available. This raises a key question: How can training algorithms ensure robustness when training noise only partially matches deployment conditions?

In response to these challenges, research has increasingly focused on rigorous noise modeling and robust training schemes that can handle real-world analog non-idealities [3,24,26,33,43,20]. Understanding how noise affects different DNN layers and accumulates through network depth is crucial for mitigating its impact. Sophisticated training techniques—ranging from gradient-based noise modeling to Bayesian approaches—have been proposed to enhance model reliability under noisy conditions [43]. Ultimately, the goal is to facilitate a new generation of DNN accelerators that can achieve cutting-edge performance while maintaining a significantly lower energy footprint.

In this paper, we systematically explore the interplay between Noisy Training and real-world dynamic noise environments. Specifically:

- **Quantifying Noise Mismatch:** We study how varying degrees of mismatch between the noise injected during training and the real noise encountered in inference affect the final model accuracy.

- **Evaluation of Robustness Techniques:** We evaluate the effectiveness of robustness techniques, including Noisy Training, Quantization, and Perturbation on weights, in dynamic noise environments characteristic of analog hardware. Our results indicate that Noisy Training significantly outperforms the alternative methods, establishing it as the most reliable baseline for robustness in such settings.
- **Strategies for Imperfect Noise Knowledge:** We propose a novel training procedure designed to mitigate the adverse effects of partial or inaccurate noise assumptions: *Variance-Aware Noisy Training (VANT)*
- **Guidelines for Robust Analog DNN Training:** Drawing on our theoretical and experimental findings, we offer practical guidelines on how to tailor *VANT* to diverse analog hardware setups.

By addressing these facets, we contribute to the broader effort of understanding and optimizing DNNs for analog hardware deployment. Our results demonstrate that carefully designed Noisy Training enables robust energy-efficient inference, even under non-ideal and time-varying hardware noise. Ultimately, we aim to offer both theoretical and practical contributions that inform the design of next-generation hardening methods for DNNs on analog accelerators.

2 Related Work

Neural network robustness is a critical research area, addressing threats such as adversarial attacks, compression errors, and computational noise. Noise injection plays a central role both as an evaluation metric and a training technique to enhance resilience. This section overviews research on robustness, noise injection, and Noisy Training in analog computing.

Quantization and Robustness Quantization, a prevalent technique in hardware-efficient deep learning, reduces numerical precision and thereby affects model robustness. Prior research has extensively examined its impact on adversarial resilience. For example, studies have demonstrated that adversarial robustness exhibits a non-monotonic relationship with bit-width, indicating that increased precision does not always enhance robustness [13]. Similarly, findings suggest that quantization can improve resilience against adversarial attacks while incurring minimal accuracy loss [10]. To mitigate error amplification that exacerbate adversarial perturbations, methods such as controlling the Lipschitz constant during quantization have been proposed [29]. Further analyses have investigated quantization effects across various neural architectures, revealing that highly complex models can recover from severe weight quantization through retraining, whereas smaller models experience greater performance degradation [37].

Perturbation and Robustness By perturbing the weights of a DNN, SGD can find regions within the parameter space, which are more robust in general. Or inversely: When injecting noise into a DNN, its predictions are more likely to

be correct if the DNN is trained towards a robust loss region. Motivated by the relationship between the loss landscape sharpness and generalization, SAM [12] seeks parameter values whose entire neighborhoods maintain consistently low training loss. Moreover, other research has incorporated pertutbation on both weights and inputs, improving the robustness against adversarial attacks [40].

Noise Injection for Robustness Noise injection has long been recognized as an effective strategy to improve generalization in machine learning models [31,17]. Early works explored its utility in mitigating overparameterization, comparing it with techniques such as weight decay and early stopping [23]. With the rise of adversarial attacks, noise injection evolved into a robust defense mechanism alongside adversarial training [16]. Various approaches have been proposed, including globally injected additive Gaussian noise [21] and ensembles leveraging layer-wise noise injection [30]. However, they often assume static noise distributions, overlooking dynamic variations in real-world scenarios.

Noisy Training in Analog Computing Unlike adversarial perturbations that affect input sensitivity, noisy analog hardware primarily introduces stochasticity into internal computations, particularly affecting neural network weights and dot product calculations. Studies have modeled non-volatile memory noise as an additive zero-mean i.i.d. Gaussian noise term on model weights, demonstrating the benefits of injecting similar noise during training and extending robustness via knowledge distillation [43]. Other research has incorporated memristor perturbation models to simulate drift in neural network weights, capturing long-term instability in analog devices and proposing architecture search and layer-specific dropout to increase robustness against drifts [42].

Dependent on the considered hardware implementation noise might be dominant in different parts of the accelerator and thus occur at different positions in the computations. Techniques have been developed to address noise from both weight readout [43,42] and subsequent computations, such as injecting noise at the output activation level [3]. Thereby, latter accounts for accumulated noise and extends further by introducing layer-specific noise to evaluate robustness and learning dynamics [3].

Additionally, Noisy Training approaches have been extended to exploit noise as an inherent feature of analog computing systems, enhancing adversarial robustness and supporting stochastic inference [7,39,5].

Despite their effectiveness, most Noisy Training strategies assume static noise characteristics. This limits real-world deployment, where noise fluctuates due to temperature changes, voltage instability, and device aging. Variance-Aware Noisy Training addresses this by integrating dynamic noise schedules that reflect realistic inference-time variations.

3 Neural Networks and Noisy Environments

We begin with an overview of the datasets and model architectures used in our work, followed by a comprehensive analysis of existing methods and their

characteristics. Additionally, we describe the methodology for simulating a noisy analog environment and outline our approach to evaluating robustness.

Datasets, Models and Experimental Setup In order to establish the effectiveness of our proposed method, experiments are performed for different networks and datasets. For the initial and comprehensive evaluation, we perform image classification on CIFAR-10 [25], CINIC-10 [9] and Tiny ImageNet [27]. For CIFAR-10, we evaluate two model architectures: LeNet-5 [28] and ResNet-18 [19]. While for Tiny ImageNet and CINIC-10, LeNet-5 is undersized and thus we focus on ResNet-18 and ResNet-50.

LeNet-5 and ResNet-18 use initial learning rates of 0.001 and 0.01 on CIFAR-10, respectively, while both ResNet-18 and ResNet-50 use a learning rate of 0.001 on Tiny ImageNet and CINIC-10. All models are trained with Adam and cosine learning rate decay, using a batch size of 128 for 400 epochs. [1]

3.1 Global Noise Injection and Noisy Training

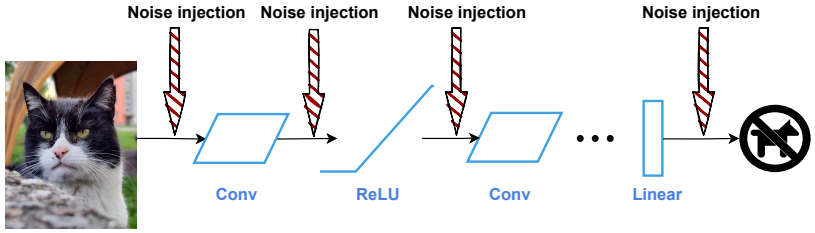


Fig. 1. Global noise injection in a DNN. Noise is applied to activations between layers.

To simulate the noisy environment present in analog hardware, we inject noise at a global level during model computation. In this work, we follow the *Walking Noise* [3] methodology, which focuses on injecting noise at the activations. We consider additive Gaussian noise, due to its widespread occurrence in natural processes and its demonstrated effectiveness in previous works on noise injection [43]. To inject noise without bias, we sample with zero mean, i.e. $\mathcal{N}(0, \sigma)$, with σ being the standard deviation of the noise. A schematic of the noise injection is shown in Figure 1. By varying the noise level σ during inference, we assess the model's robustness under noisy environments of different intensities.

Noise injection during training has been shown to significantly improve network accuracy under noisy computations [24,43]. We also evaluate performance of standard Noisy Training by injecting noise in the forward pass during the training procedure. Figure 2 reveals the impact of noise injection to accuracy for

[1] The code is available at: `https://github.com/HAWAIILAB/VANT`

models trained with and without noise injection. When a model is trained with the same noise level it encounters during inference, it typically achieves optimal accuracy. By connecting these optimal points, we obtain the dashed curve, which represents the best achievable performance using Noisy Training at each noise level. We assume the dashed curve thus to be the theoretical upper bound in terms of robustness and accuracy for any given noise level.

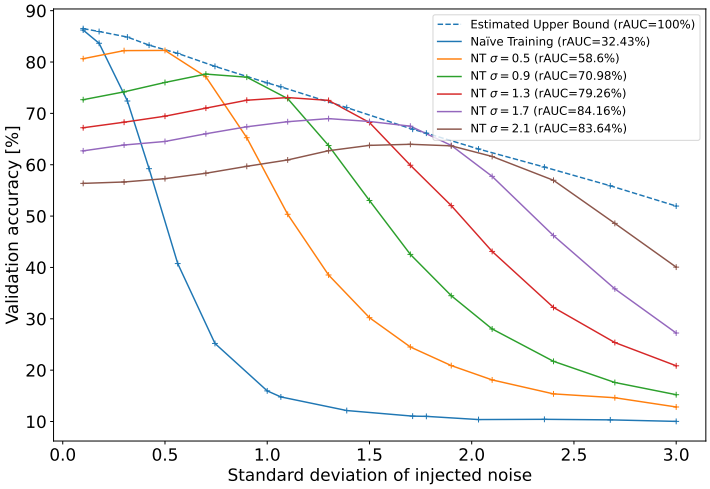


Fig. 2. Accuracy degradation under noise. NT: Models trained with Noisy Training (NT) at noise of σ. Dashed line indicates the upper bound, when using the same noise for training and inference. ResNet-18 on CIFAR-10.

3.2 Evaluation: Quantifying Robustness under Noise

The standard deviation of the noise σ under test is selected from the range $[0.1, 3.0]$. This selection is based on previous findings [24], which report noise levels on analog hardware to fall within this interval. This ensures that our robustness assessment is both relevant and practical for deployment scenarios involving noisy analog computations.

To quantify the robustness under noisy computation, we utilize the Area Under the Curve (AUC) as primary performance metric. However, directly comparing AUC values can be misleading, as accuracy is influenced by factors such as model complexity and dataset difficulty. Moreover, absolute AUC values do not directly indicate how close a method's performance is to the upper bound. To provide a fair comparison, we use the **relative AUC percentage (rAUC)**, defined as:

$$\text{rAUC} = \frac{\text{AUC of the method}}{\text{AUC of the upper bound curve}}[\%]$$

This metric directly indicates how close a method is to the best possible result.

4 Noisy Training: Strong but Not Flawless

As shown, training with noise injection significantly enhances robustness compared to a baseline model trained without noise. However, it is important to understand how other robustness-enhancing methods contribute to overall performance. In the following, we explore these methods as well as the limitations of Noisy Training.

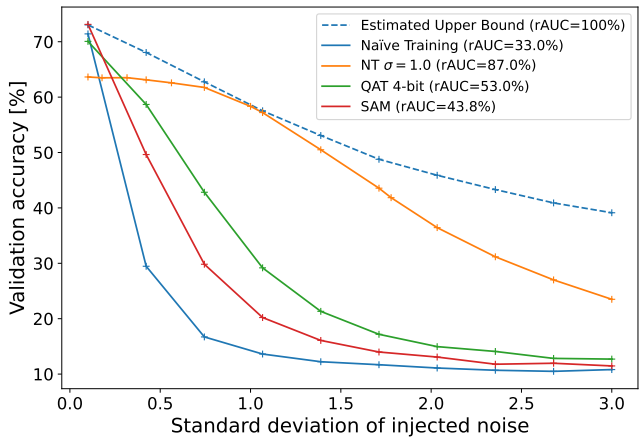


Fig. 3. Comparison of typical hardening methods for LeNet-5 on CIFAR-10.

4.1 The Importance of Noisy Training

We begin with the assumption that we do not have prior knowledge of noise characteristics in analog hardware and explore how to mitigate its impact. We consider quantization and the generalization method Sharpness-Aware Minimization (SAM) [12] as potential countermeasures.

Intuitively, quantization introduces quantization error during computation, which may however increase the stability of DNNs when subjected to noise perturbations. To evaluate the impact of quantization techniques on the robustness of neural networks against computational noise, we test a quantized network in a noisy environment. For our evaluations we employ quantization-aware training (QAT)[2]. Figure 3 shows that a model quantized to 4-bit outperforms the baseline model. We further evaluate the impact of the perturbation method SAM,

[2] Our implementation is based on Brevitas (https://github.com/Xilinx/brevitas)

as described in section 2. As shown in Figure 3, while SAM offers improvements it remains less effective than quantization.

However, neither quantization nor SAM can match the performance of Noisy Training in enhancing robustness. This is largely due to the fundamental characteristics of noise in analog hardware: it is pervasive, affecting not only the input but also internal computations; its magnitude can be substantial; and, critically, it accumulates as signals propagate through the neural network. These factors highlight that an effective countermeasures must account for the specific noise properties of the hardware.

4.2 Limitations of Noisy Training

While Noisy Training is essential for robustness, a key challenge remains: the noise characteristics of analog hardware can fluctuate over time due to environmental factors such as temperature variations. Additionally, different hardware units usually exhibit notable variations in noise levels. We define the noise level present in a specific hardware instance at the time of measurement as σ_{train}.

This raises an important question: even if a model is trained under a specific noise level, how well does it generalize when the on-device noise deviates from the training conditions? To explore this, we train the model under a fixed noise level and then evaluate its performance across different noise strengths. The orange curve in Figure 3 illustrates the performance of LeNet-5 trained with $\sigma_{\text{train}} = 1.0$. As expected, the model achieves optimal performance when the noise level matches the training condition. However, as the noise deviates from $\sigma_{\text{train}} = 1.0$, accuracy declines, highlighting sensitivity to mismatched noise levels.

This observation leads to a crucial conclusion: Noisy Training is only effective when the noise characteristics are precisely known. If the noise level during training does not align with the actual noise encountered during deployment, the model's robustness can be significantly compromised. This leads to the central research question of this work:

> How can we train models that remain robust across an entire fleet of devices, each potentially exhibiting different noise strengths over time?

5 Beating the odds: Variance-Aware Noisy Training

In order to address the previously presented shortcomings of Noisy Training we present a novel training technique, *Variance-Aware Noisy Training (VANT)* which is more robust against unstable noise settings.

5.1 Methodology: Variance-Aware Noisy Training

The central assumption behind standard (stable) Noisy Training is that one can model the accelerator's noise perfectly, in particular, that it will remain constant over time and devices. This way gradient descent adjusts a given DNN to the

characteristics of a given accelerator. Centrally missing however is any treatment of variation in the noise. We thus extend Noisy Training as follows:

$$\begin{aligned} x &\sim \mathcal{N}(0, \sigma_{\text{var}}), \\ \sigma_{\text{var}} &\sim \mathcal{N}(\alpha \cdot \sigma_{\text{train}}, \theta). \end{aligned} \tag{1}$$

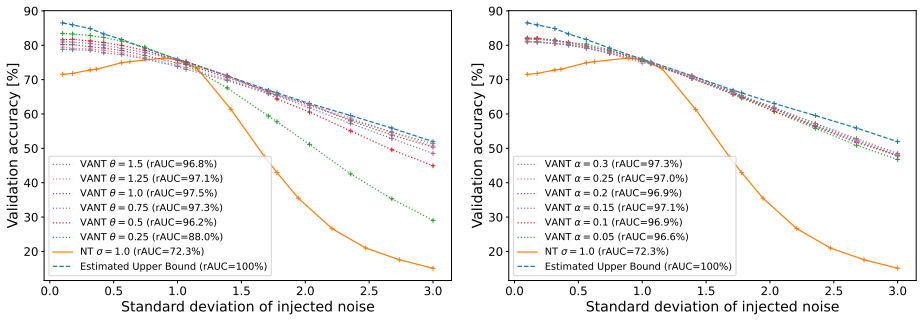
Here σ_{train} is an extrinsic parameter, representing the known noise characteristic of a given hardware target. *VANT* additionally introduces two parameters: θ adjusts Noisy Training to the time variations of a given accelerator, while α is a calibration parameter for σ_{train}. Looking forward to Sections 5.2 and 5.3, we note that *VANT* is rather insensitive to α, while θ strongly depends on the chosen σ_{train}.

During training σ_{var} is then sampled for each input image, while additively injected noise (x) is sampled for each activation. All sampling and thus noise injection only applies during the forward pass of gradient descent training.

5.2 Experiments on CIFAR-10

In order to evaluate how the parameters of *VANT* behave, we run initial evaluations on CIFAR-10. In a later step we then evaluate how well *VANT* transfers to a more complex dataset, when utilizing the same parameters obtained here.

Initially we evaluate α and θ individually to explore their general behavior (Figure 4). For any setting of α and θ, the robustness (rAUC) of *VANT* is sig-



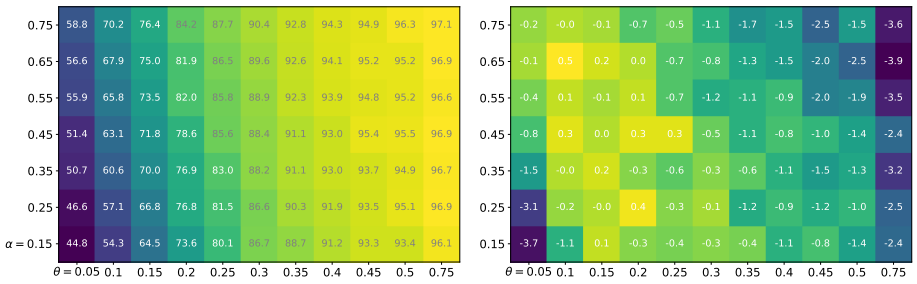
(a) Constant $\alpha = 0.3$ (b) Constant $\theta = 0.75$

Fig. 4. Exploration of *VANT* hyperparameters for ResNet-18 on CIFAR-10, when plotted over the injected noise. Here compared to Noisy Training (NT) at $\sigma_{\text{train}} = 1.0$.

nificantly better compared to standard Noisy Training. And results are similar for LeNet-5. While variations in α appear to have little impact on the overall robustness as seen in Figure 4(b), θ plays a significant role in both the overall robustness and shape of the curve, see Figure 4(a). Furthermore, in Figure 4(a) it also becomes apparent that the method does not necessarily preserve the peak

accuracy of standard Noisy Training at σ_{train}. The variation of noise, θ, effectively increases the maximum noise observed, which shifts the point of optimal accuracy for *VANT*. Thus, during the evaluation of α and θ this influence needs to be considered. This is accomplished by measuring how close the accuracy at σ_{train} is to standard Noisy Training, since ideally *VANT* should preserve all advantages of Noisy Training. In order to quantify this behavior, a new metric is introduced: *Preserved Accuracy*. It measures by how much the accuracy of *VANT* and standard Noisy Training differ at the noise injection point of σ_{train}. Ideally a setting of α and θ can be found, which keeps this metric at zero or higher, perfectly preserving the accuracy of standard Noisy Training.

To further identify the dependency of the robustness (rAUC) and preserved accuracy on α and θ, a grid scan for both parameters across a wide range is performed in Figure 5. For the robustness we primarily observe that θ plays a



(a) Robustness (rAUC %) (b) Preserved Accuracy

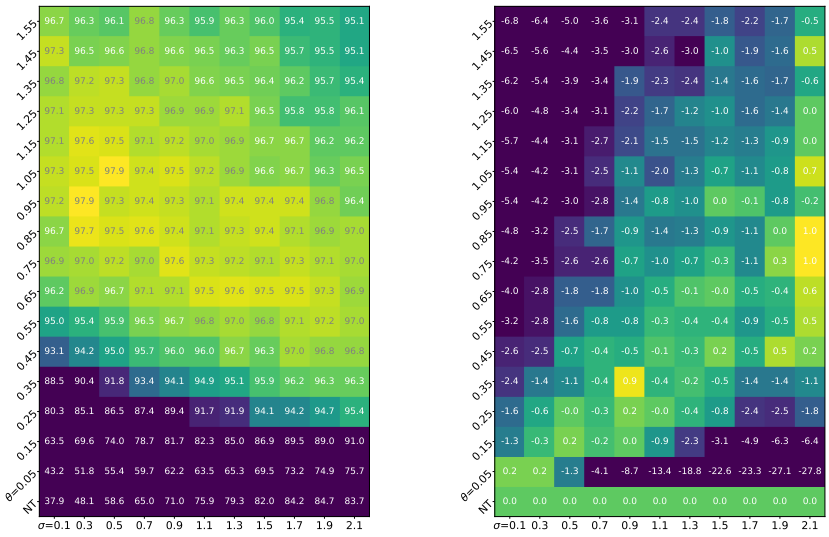
Fig. 5. Heatmap of quality metrics for *VANT* with ResNet-18 on CIFAR-10, when varying both hyperparameters of *VANT*, while keeping $\sigma_{\text{train}} = 0.4$ constant.

strictly monotonic role in improving robustness. We further note that α similarly monotonically increases the robustness. While this effect appears to suggest that one should simply increase both α and θ, this is not the case. Instead the maximally achievable robustness is bounded by the preserved accuracy, as this value should stay at zero or larger in Figure 5(b). Notably a sweet spot becomes visible for finding an optimal set of parameters for *VANT*. While this sweet spot is well constrained in θ, it is relatively broad for α.

Selecting the best set of α and θ parameters is done as follows:

1. Select all sets of α and θ, for which the preserved accuracy is above 0, ensuring parity to Noisy Training.
2. Sub-select α and θ for which the robustness is maximized.

In the case of Figure 5, we select $\alpha = 0.45$ and $\theta = 0.25$ as the optimal parameters.



(a) Robustness (rAUC %) (b) Preserved Accuracy

Fig. 6. Heatmap of quality metrics for *VANT* with ResNet-18 on CIFAR-10, when varying θ and the reference hardware noise σ_{train}. While keeping $\alpha = 0.45$ constant, as it is largely invariant. NT: Noisy Training as baseline at the bottom.

Since different analog accelerators require different initial noise levels, we now explore how *VANT* behaves for different σ_{train}. As *VANT* is largely invariant to α we fix it to 0.45, as a middle ground for the sweet spot from Figure 5(b).

Similarly to Figure 5, Figure 6 shows both robustness and preserved accuracy. However, in this case the x-axis denotes the change in σ_{train}, e.g. different hardware accelerators, and the y-axis explores the behavior of θ. Again, there is a trade-off to be made between the preserved accuracy and the robustness.

Notably, θ shows a broad optimum for the robustness. However, when following the procedure for selecting the best θ as stated in the steps above, then θ is tightly confined by an approximately linear relationship between σ_{train} and θ, which is approximately: $\theta = 0.4 \cdot \sigma_{\text{train}}$

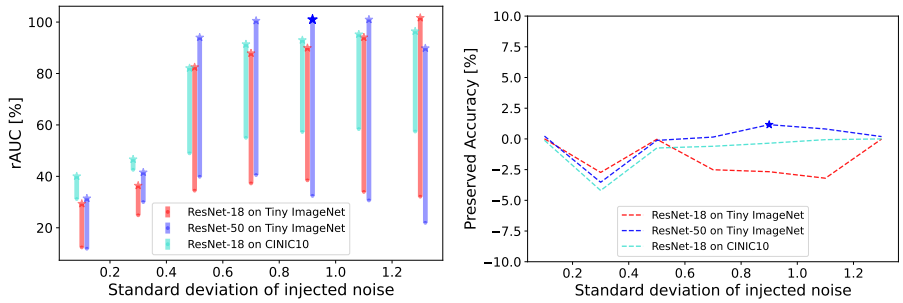
Table 1. Quality metrics for *VANT* with ResNet-18 on CIFAR-10 for the optimal θ and α, when varying the reference hardware noise σ_{train}.

σ_{train}	0.1	0.3	0.5	0.7	0.9	1.1	1.3	1.5	1.7	1.9	2.1
θ	0.05	0.05	0.25	0.35	0.35	0.45	0.65	0.65	0.95	0.75	0.85
Preserved Accuracy	0.2	0.2	0	-0.4	0.9	-0.1	-0.1	0	-0.1	0.3	1.0
rAUC [%] NT	37.9	48.1	58.6	65.0	71.0	75.9	79.3	82.0	84.2	84.7	83.7
rAUC [%] VANT	43.2	51.8	86.5	93.4	94.1	96.0	**97.6**	97.5	97.5	97.1	97.0

However, for all following experiments, we choose θ as described with the steps above, as these are more accurate given the ground truth data available. These can be found in Table 1. All selected settings for *VANT* simultaneously preserves the accuracy of standard Noisy Training and strictly improve the robustness, as visible by comparing them to the bottom row in Figure 6.

5.3 Generalizing to Complex Data: CINIC-10 & Tiny ImageNet

In the following we explicitly test for two properties of *VANT*: How well it transfers to more complex datasets and if the parameters of *VANT* additionally show stability across varying architectures. As such all settings for α and θ for the following experiments are chosen from the optimal results of the previous section, shown in Table 1.



(a) Robustness (rAUC) (b) Preserved Accuracy

Fig. 7. Quality metrics for *VANT* with ResNet-18/50 on CINIC-10 and Tiny ImageNet at varying σ_{train}. For the robustness (rAUC): The dot at the bottom of the bars represents standard Noisy Training, while the star at the top of the bar represents *VANT*.

As a naturally more complex architecture we investigate ResNet-50. On the dataset side we increase complexity in two steps: CINIC-10 and Tiny ImageNet.

Results for both quality metrics under consideration are shown in Figure 7. Looking first at the preserved accuracy, one can observe that the accuracy is generally preserved across datasets and models. A notable exception is $\sigma = 0.3$, where the preserved accuracy drops across all experiments and increases in robustness are also low. We postulate that for this specific setting θ was ill-chosen. Further investigating the robustness in Figure 7(a), we observe that *VANT* improves the robustness for all models, datasets and strengths of injected noise. However the effect is not consistent across the whole range of injected noise. *VANT* provides less significant robustness improvements for noise strengths of $\sigma_{\text{train}} < 0.5$. The underlying reason is that in order to achieve high accuracy under low noise, regions of higher noise are less prominently sampled. This results in subpar performance for much of the investigated range by the rAUC metric. Notably this behavior is largely inherited from Noisy Training.

For $\sigma_{\text{train}} \geq 0.5$, however, *VANT* is considerably more robust. Interestingly the largest improvements can be found on Tiny ImageNet with ResNet-50, the most complex dataset and model investigated, where the rAUC increases dramatically from 32.4% to 99.7% at $\sigma_{\text{train}} = 0.9$. Nonetheless, siginificant improvements are also visible for ResNet-18 and CINIC-10.

Concluding we find, that *VANT* shows good generalization across both datasets and models. While the performance of a set of chosen hyper-parameters remains consistent at the same time.

6 Summary

The increasing computational demands of modern deep learning models pose significant challenges for conventional digital CMOS technology, which is approaching fundamental scaling limits. To address this bottleneck, alternative computing paradigms have gained attention, with analog computing emerging as a promising candidate. However, analog accelerators introduce new challenges, particularly due to inherent noise that can degrade model performance.

In this work, we first examine the challenges associated with training DNNs under these imperfect conditions. We observe that while techniques such as quantization and SAM contribute to improved model robustness, they fall short of the robustness provided by Noisy Training. However, Noisy Training itself has critical limitations: although it can achieve high peak accuracy, it exhibits poor generalization when subjected to variations in noise levels, as typically encountered in analog hardware. Specifically, standard Noisy Training tends to overfit to a particular noise configuration, leading to suboptimal performance when the noise characteristics shift due to factors such as temperature fluctuations and hardware aging—common occurrences in analog accelerators.

To address this robustness gap, we propose *Variance-Aware Noisy Training (VANT)*, an extension of standard Noisy Training that explicitly accounts for temporal variations in noise. *VANT* incorporates an additional term which models the expected evolution of the noise environment over time. It thus enhances the generalization capabilities of DNNs under real-world deployment conditions, where noise characteristics are dynamic rather than fixed.

Empirical evaluations demonstrate the effectiveness of *VANT* in improving robustness across different noise regimes and dataset complexities. For instance, under typical analog noise conditions, *VANT* increases robustness from 79.3% to 97.6% on CIFAR-10. On the more challenging Tiny ImageNet dataset, VANT similarly yields significant gains, improving performance from 32.4% to 99.7%.

In summary, our findings highlight a crucial principle for deploying DNNs on noisy analog hardware: it is not sufficient to account solely for the immediate noise environment during training; rather, it is essential to model the temporal evolution of noise over time. By adopting a more comprehensive approach that considers the dynamic nature of hardware noise, *VANT* represents a significant step toward enabling robust deep learning models on fleets of analog accelerators.

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

References

1. Ambrogio, S., Narayanan, P., Tsai, H., Shelby, R.M., Boybat, I., di Nolfo, C., Sidler, S., Giordano, M., Bodini, M., Farinha, N.C.P., Killeen, B., Cheng, C., Jaoudi, Y., Burr, G.W.: Equivalent-accuracy accelerated neural-network training using analogue memory. Nature **558**(7708), 60–67 (Jun 2018). https://doi.org/10.1038/s41586-018-0180-5
2. Bishop, C.M.: Training with noise is equivalent to tikhonov regularization. Neural Computation **7**(1), 108–116 (1995). https://doi.org/10.1162/neco.1995.7.1.108
3. Borras, H., Klein, B., Fröning, H.: Walking Noise: On layer-specific robustness of neural architectures against noisy computations and associated characteristic learning dynamics. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. ECML-PKDD (2024). https://doi.org/10.1007/978-3-031-70359-1_3
4. Bricken, T., Schaeffer, R., Olshausen, B., Kreiman, G.: Emergence of sparse representations from noise. In: International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 3148–3191. PMLR (23–29 Jul 2023), https://proceedings.mlr.press/v202/bricken23a.html
5. Brückerhoff-Plückelmann, F., Borras, H., Klein, B., Varri, A., Becker, M., Dijkstra, J., Brückerhoff, M., Wright, C.D., Salinga, M., Bhaskaran, H., Risse, B., Fröning, H., Pernice, W.: Probabilistic photonic computing with chaotic light. Nature Communications **15**(1), 10445 (Dec 2024), https://doi.org/10.1038/s41467-024-54931-6
6. Burr, G.W., Shelby, R.M., Sebastian, A., Kim, S., Kim, S., Sidler, S., Virwani, K., Ishii, M., Narayanan, P., Fumarola, A., et al.: Neuromorphic computing using non-volatile memory. Advances in Physics: X **2**(1), 89–124 (2017). https://doi.org/10.1080/23746149.2016.1259585
7. Cappelli, A., Ohana, R., Launay, J., Meunier, L., Poli, I., Krzakala, F.: Adversarial robustness by design through analog computing and synthetic gradients. In: IEEE International Conference on Acoustics, Speech and Signal Processing (2022). https://doi.org/10.1109/ICASSP43922.2022.9746671
8. Chi, P., Li, S., Xu, C., Zhang, T., Zhao, J., Liu, Y., Wang, Y., Xie, Y.: PRIME: A novel processing-in-memory architecture for neural network computation in reram-based main memory. In: International Symposium on Computer Architecture (ISCA) (2016). https://doi.org/10.1109/ISCA.2016.13
9. Darlow, L.N., Crowley, E.J., Antoniou, A., Storkey, A.J.: CINIC-10 is not imagenet or CIFAR-10. CoRR **abs/1810.03505** (2018), http://arxiv.org/abs/1810.03505
10. Duncan, K., Komendantskaya, E., Stewart, R., Lones, M.: Relative robustness of quantized neural networks against adversarial attacks. In: International Joint Conference on Neural Networks. IJCNN, IEEE (2020). https://doi.org/10.1109/IJCNN48605.2020.9207596
11. Feldmann, J., Youngblood, N., Karpov, M., Gehring, H., Li, X., Stappers, M., Le Gallo, M., Fu, X., Lukashchuk, A., Raja, A.S., Liu, J., Wright, C.D., Sebastian,

A., Kippenberg, T.J., Pernice, W.H.P., Bhaskaran, H.: Parallel convolutional processing using an integrated photonic tensor core. Nature **589**(7840), 52–58 (Jan 2021). https://doi.org/10.1038/s41586-020-03070-1
12. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-aware minimization for efficiently improving generalization. International Conference on Learning Representations (2021), https://openreview.net/forum?id=6Tm1mposlrM
13. Giacobbe, M., Henzinger, T.A., Lechner, M.: How many bits does it take to quantize your neural network? In: Tools and Algorithms for the Construction and Analysis of Systems. TACAS (2020). https://doi.org/10.1007/978-3-030-45237-7_5
14. Gokmen, T., Vlasov, Y.: Acceleration of deep neural network training with resistive cross-point devices: Design considerations. Frontiers in neuroscience **10**, 333 (2016). https://doi.org/10.3389/fnins.2016.00333
15. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
16. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. CoRR **abs/1412.6572** (2014), https://arxiv.org/abs/1412.6572
17. Grandvalet, Y., Canu, S., Boucheron, S.: Noise injection: Theoretical prospects. Neural Computation **9**(5) (1997). https://doi.org/10.1162/neco.1997.9.5.1093
18. Gulcehre, C., Moczulski, M., Denil, M., Bengio, Y.: Noisy activation functions. In: Balcan, M.F., Weinberger, K.Q. (eds.) International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 3059–3068. PMLR (2016), https://proceedings.mlr.press/v48/gulcehre16.html
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR (2016). https://doi.org/10.1109/CVPR.2016.90
20. He, Z., Lin, J., Ewetz, R., Yuan, J.S., Fan, D.: Noise injection adaption: End-to-end reram crossbar non-ideal effect adaption for neural network mapping. In: Design Automation Conference. DAC, ACM (2019). https://doi.org/10.1145/3316781.3317870
21. He, Z., Rakin, A.S., Fan, D.: Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019), https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00068
22. Holmstrom, L., Koistinen, P.: Using additive noise in back-propagation training. IEEE Transactions on Neural Networks **3**(1), 24–38 (1992). https://doi.org/10.1109/72.105415
23. Jiang, Y., Zur, R.M., Pesce, L.L., Drukker, K.: A study of the effect of noise injection on the training of artificial neural networks. In: International Joint Conference on Neural Networks (2009). https://doi.org/10.1109/IJCNN.2009.5178981
24. Klein, B., Kuhn, L., Weis, J., Emmel, A., Stradmann, Y., Schemmel, J., Fröning, H.: Towards addressing noise and static variations of analog computations using efficient retraining. In: ECML PKDD 2021 Workshops (2021). https://doi.org/10.1007/978-3-030-93736-2_32
25. Krizhevsky, A., Hinton, G.E.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
26. Kuhn, L., Klein, B., Fröning, H.: On the non-associativity of analog computations pp. 183–195 (2025), https://doi.org/10.1007/978-3-031-74643-7_15
27. Le, Y., Yang, X.S.: Tiny imagenet visual recognition challenge (2015), https://api.semanticscholar.org/CorpusID:16664790

28. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86** (1998). https://doi.org/10.1109/5.726791
29. Lin, J., Gan, C., Han, S.: Defensive quantization: When efficiency meets robustness. In: International Conference on Learning Representations. ICLR (2019), https://openreview.net/forum?id=ryetZ20ctX
30. Liu, X., Cheng, M., Zhang, H., Hsieh, C.J.: Towards robust neural networks via random self-ensemble. In: European Conference on Computer Vision (2018). https://doi.org/10.1007/978-3-030-01234-2_23
31. Murray, A., Edwards, P.: Enhanced mlp performance and fault tolerance resulting from synaptic weight noise during training. IEEE Transactions on Neural Networks **5**(5) (1994). https://doi.org/10.1109/72.317730
32. Neelakantan, A., Vilnis, L., Le, Q.V., Sutskever, I., Kaiser, L., Kurach, K., Martens, J.: Adding gradient noise improves learning for very deep networks. CoRR **abs/1511.06807** (2015), https://arxiv.org/abs/1511.06807
33. Neftci, E.O., Mostafa, H., Zenke, F.: Surrogate gradient learning in spiking neural networks. CoRR **abs/1901.09948** (2019), http://arxiv.org/abs/1901.09948
34. Schemmel, J., Billaudelle, S., Dauer, P., Weis, J.: Accelerated analog neuromorphic computing. CoRR **abs/2003.11996** (2020), https://arxiv.org/abs/2003.11996
35. Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R., Eleftheriou, E.: Memory devices and applications for in-memory computing. Nature Nanotechnology **15**(7), 529–544 (Jul 2020). https://doi.org/10.1038/s41565-020-0655-z
36. Shafiee, A., Nag, A., Muralimanohar, N., Balasubramonian, R., Strachan, J.P., Hu, M., Williams, R.S., Srikumar, V.: ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In: International Symposium on Computer Architecture. ISCA (2016). https://doi.org/10.1109/ISCA.2016.12
37. Sung, W., Shin, S., Hwang, K.: Resiliency of deep neural networks under quantization. CoRR **abs/1511.06488** (2015), https://arxiv.org/abs/1511.06488
38. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness May Be at Odds with Accuracy. arXiv e-prints arXiv:1805.12152 (May 2018). https://doi.org/10.48550/arXiv.1805.12152
39. Wu, C., Yang, X., Yu, H., Peng, R., Takeuchi, I., Chen, Y., Li, M.: Harnessing optoelectronic noises in a photonic generative network. Science Advances **8**(3) (2022), https://www.science.org/doi/abs/10.1126/sciadv.abm2956
40. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. Advances in Neural Information Processing Systems **33** (2020). https://doi.org/10.5555/3495724.3495973
41. Yang, J.J., Strukov, D.B., Stewart, D.R.: Memristive devices for computing. Nature Nanotech **8**(1), 13–24 (2013). https://doi.org/10.1038/nnano.2012.240
42. Ye, N., Cao, L., Yang, L., Zhang, Z., Fang, Z., Gu, Q., Yang, G.Z.: Improving the robustness of analog deep neural networks through a bayes-optimized noise injection approach. Communications Engineering **2** (2023). https://doi.org/10.1038/s44172-023-00074-3
43. Zhou, C., Kadambi, P., Mattina, M., Whatmough, P.N.: Noisy machines: Understanding noisy neural networks and enhancing robustness to analog hardware errors using distillation. CoRR **abs/2001.04974** (2020), https://arxiv.org/abs/2001.04974