ORIGINAL ARTICLE

Noise-resilient photonic neural networks through adaptive quantization

Emilio Paolini · Lorenzo De Marinis · Peter Seigo Kincaid · Luca Valcarenghi · Giampiero Contestabile · Ioannis Roumpos · Miltiadis Moralis-Pegios · Nikos Pleros · Nicola Andriolli

Received: 17 July 2025 / Accepted: 3 February 2026 / Published online: 7 March 2026
© The Author(s) 2026

Abstract

Photonic neural networks have emerged as a promising solution to overcome limitations of traditional hardware for neuromorphic computations, offering advantages in bandwidth, latency, and power efficiency. However, their performance is constrained by the limited precision of analog photonic computing, which is affected by inherent noise sources such as thermal and shot noise, and distortions. These effects degrade the photonic neural network performance, reducing the bit resolution achievable in photonic hardware typically to 2–4 bits. Traditional quantization strategies fail to account for these noise contributions, resulting in a substantial accuracy loss during inference. This paper introduces an adaptive quantization method called Adaptive-Quantization Photonic-Aware Neural Network (AQ-PANN) to address the challenges posed by different noise sources in analog photonic hardware. The proposed method uses a learnable step size quantization scheme to achieve high accuracy and stability under varying noise levels, introducing a scheme that unifies quantization step adaptation with noise injection exactly where photonic distortions arise. This design incurs only a minor training-time overhead, as it involves learning a small number of per-layer quantization step sizes and does not affect inference. Experimental evaluations on three commonly used test datasets (MNIST, SVHN, and CIFAR-10) with different bit resolutions show the robustness of AQ-PANN. On MNIST, an accuracy drop of only 2% was observed from low to high noise levels in a 4-bit configuration, while traditional DoReFa quantization suffered a 29% drop. For the SVHN dataset, AQ-PANN obtained a mean accuracy of 92% under high noise with 4-bit quantization, outperforming DoReFa by over 45%. On CIFAR-10, AQ-PANN maintained close to 60% accuracy under high noise in the 4-bit configuration, whereas DoReFa and PACT both collapsed below 40%. These results highlight the effectiveness of AQ-PANN in sustaining model performance across different noise intensities, enabling practical photonic neural network deployment.

Keywords Deep learning · Photonic neural network · Analog computing · Quantization · Noise

1 Introduction

In recent years Neural Network (NN) models have proven effective in dealing with numerous practical applications, ranging from computer vision [1] to network security [2, 3]. However, in each of these domains, Deep Neural Networks (DNNs) have obtained high accuracy by leveraging very large and deep models, e.g., which, however, require up to 100 s of ExaOps of computation during training and terabytes of storage [4]. Hence, research efforts have looked into software and hardware solutions to replace traditional hardware, which cannot keep up with this exponential trend [5]. Software solutions include NN model shrinking and compression [6], while hardware solutions involve developing dedicated computing platforms, such as neuromorphic architectures [7], which are designed to mimic NN structures and reduce power consumption while enhancing speed [8].

Among hardware-based solutions, analog photonic computing is a promising alternative, taking advantage of high bandwidth, low latency, and low power consumption of photonics [9]. Recently, a fully integrated photonic DNN was demonstrated [10] capable of ultrafast, coherent processing and in-situ training. Multiple implementations of similar systems have also been reported, such as the work discussed in [11] on photonic matrix processors leveraging advanced silicon photonics technology, and in [12] on hybrid photonic-electronic systems designed for scalable deep learning tasks. These systems highlight the growing interest in photonic neural networks, which aim to leverage photonic hardware for executing neural network operations. Additionally, a Photonic Electronic Multiply-Accumulate Neuron (PEMAN) architecture has been recently introduced and experimentally demonstrated [13, 14]. This architecture performs Time Division Multiplexing (TDM) multiplications in the photonic domain and handles accumulation in the electronic domain using analog integrator circuitry. In addition, it also applies a nonlinearity within its low-speed Analog to Digital Conversion (ADC). Such architecture is designed to perform all essential functions of an artificial neuron, offering a balance between multiplication speed, energy efficiency, and accuracy. PEMAN-based approaches have already been discussed, validating the proposed architecture in both healthcare [15] and cybersecurity use-cases [16].

Despite its advantages, when deployed in real-world scenarios, these analog photonic circuits are prone to various types of noise and distortions, restricting the number of distinguishable values. Consequently, typical resolutions after analog-to-digital conversion range between 2 and 4 bits, leading to potential performance degradation. On the architectural side, there are limitations on input representations, fan-in per neuron, and size of convolutional kernels [17]. Consequently, addressing these physical layer constraints is crucial for advancing the practical deployment of photonic neural networks. Both the NN architecture and the training process need to be tailored to account for the inherent noise and the underlying hardware limitations [18, 19]. By integrating the characteristics and behavior of photonic devices into the learning phase, it is possible to enable a Photonic-Aware Neural Network (PANN) architecture [8] to account for photonic hardware limitations directly during model optimization and progress toward optics-informed DL models [20].

In this paper, we extend the PANN training scheme proposed in [8] by introducing Adaptive-Quantization Photonic-Aware Neural Networks (AQ-PANN), a training framework that makes photonic neural models resilient to the noise of underlying photonic hardware. AQ-PANN integrates quantization-aware training with a noise-injection mechanism specifically tailored to the characteristics of analog photonic hardware, enabling robust inference under low-bitwidth and high-noise conditions. First, we formulate a learnable step-size quantizer whose parameter s is co-optimized with the measured variance of Additive White Gaussian Noise (AWGN) produced by optical hardware. Second, we reorganize the quantization-aware training pipeline by injecting AWGN immediately after each photonic Multiply-Accumulate (MAC), rather than after entire layers as in prior models, reflecting the actual accumulation flow in photonic neuromorphic systems. Finally, we demonstrate that AQ-PANN achieves state-of-the-art robustness on MNIST, Street-View House Numbers (SVHN), and CIFAR-10 benchmarks, maintaining high accuracy across noise intensities and quantization regimes. Importantly, AQ-PANN does not modify the underlying PANN architecture nor the inference pipeline, and its additional cost is confined to training, where a small set of per-layer quantization step sizes is jointly optimized with the network weights.

This contribution is distinct from existing strategies: DoReFa and pure Learned Step Size Quantization (LSQ) focus on quantization alone without explicitly incorporating noise characteristics; PACT adapts activation ranges but leaves the quantization bins unchanged; and activation stretching modifies nonlinearities but does not co-train the quantization intervals. By contrast, AQ-PANN integrates all three aspects, i.e., quantization, noise placement, and gradient balancing, within a single framework, which we show to be critical for robustness in photonic systems. Specifically, the main contributions of this work can be summarized as follows:

- We introduce AQ-PANN, a photonic-aware training framework that jointly accounts for low-bit quantization and hardware-induced noise in analog photonic neural networks.
- We propose a learnable step-size quantization mechanism that co-trains quantization intervals with network weights, allowing the effective resolution to adapt to different noise levels and bitwidth constraints.
- We explicitly model noise at the photonic MAC level during training, reflecting where distortions physically arise in interferometric and electro-photonic computing pipelines.
- We incorporate a gradient-balancing strategy that stabilizes the joint optimization of weights and quantization parameters under low-precision and noisy operating regimes.
- We validate the proposed approach across multiple datasets and noise conditions, showing that the joint treatment of quantization, noise placement, and optimization is critical to achieving robust photonic inference.

2 Related work

2.1 Background: neuromorphic optics

The neuromorphic photonic community has followed several parallel paths, each exploiting different physical properties of light to mimic neural dynamics. Early demonstrations of optical plasticity relied on solitonic and photorefractive effects, where light beams could inscribe self-written waveguides acting as adaptive interconnections. These solitonic neural networks offered a proof of concept for all-optical memory and associative recognition [21, 22].

A second direction has leveraged phase-change materials integrated with silicon photonics. Here, non-volatile switching and analog tunability allow photonic circuits to store and update synaptic weights directly in hardware, enabling in-memory optical computing with long-term retention, though at the expense of write energy and endurance [23].

In parallel, spiking photonic neurons have been realized using semiconductor lasers, and particularly vertical-cavity surface-emitting lasers (VCSELs). These devices reproduce excitable neural dynamics at sub-nanosecond timescales, enabling GHz-rate processing and reservoir-style architectures suited for ultrafast image and signal classification [24, 25].

Within this scenario, the present work focuses on interferometric and electro-photonic pipelines, such as meshes of Mach–Zehnder interferometers and PEMAN-style hybrids [13]. In these systems, matrix multiplications are carried out in the photonic domain, while accumulation and detection occur electronically. Their main limitation is the susceptibility to noise and distortions immediately after the photonic multiply–accumulate stage, which constrains the effective bit precision to a few levels. It is within this context that AQ-PANN is introduced: a training framework specifically designed to take into accounts these hardware constraints by injecting noise at the point where it physically arises and adapting quantization during learning to preserve accuracy under low-bit, high-noise conditions.

2.2 State of the art

This section focuses on prior works that are most relevant to the design choices of AQ-PANN. In particular, we discuss approaches that explicitly address hardware-induced noise and low-precision constraints during training, as these aspects are central to robust inference in analog photonic neural systems. Accordingly, we emphasize works that model noise during training [18] or investigate quantization effects in NNs [26], as these directly inform the noise-aware and quantization-aware components of AQ-PANN.

As discussed by Kirtas et al. [18], counteracting noise in the training process of NNs improves their performance, obtaining models robust to various impairments arising from the underlying photonic hardware. For instance, in the presence of input, weight, and activation noise, the proposed noise-aware initialization method achieved a Cohen's κ statistic of 0.1566 for a noise variance $\sigma^2 = 0.1$, outperforming the baseline methods, which included Xavier and He initialization techniques without noise-aware training. Even for a higher noise variance $\sigma^2 = 0.5$, the method retained performance with a Cohen's κ of 0.1, highlighting its ability to enhance robustness in noisy environments.

In this context, several methods have been proposed to mitigate the effects of noise on NN computations. For instance, Mourgias-Alexandris et al. [27] explores the impact of various non-deterministic noise sources within photonic hardware by modeling them as AWGN. The mean and standard deviation of the AWGN were calibrated based on the experimentally measured noise characteristics of the photonic circuit. By incorporating these noise parameters directly into the training process, the proposed noise-aware training model significantly improved the noise resilience of the neural network, achieving 99.3% accuracy on the MNIST dataset under experimental conditions even with a noise standard deviation of $\sigma = 0.4$. This demonstrates the effectiveness of noise-aware training in maintaining high performance in the presence of realistic photonic impairments.

Still in the context of counteracting noise during training, Paolini et al. [28] proposed an activation stretching strategy that adjusts the range of activation functions to enhance noise resilience in PANNs. By widening the activation intervals, this method reduces the likelihood of noise-induced errors, especially in low-resolution photonic hardware. The results indicate that the proposed stretching strategy leads to a significant improvement in classification accuracy for photonic neuromorphic systems.

A noise-aware training approach has been proposed by Passalis et al. [29] for recurrent photonic networks aimed at enhancing robustness. The approach includes an adaptive initialization technique, by adjusting the initial weight distributions to account for the expected noise. Results demonstrate how the network learns to be more resilient, ultimately improving performance and reducing sensitivity to noise during inference.

While AQ-PANN builds on core ideas from quantization-aware training [26] and noise modeling in photonic systems [18], its formulation introduces several components specifically adapted to the constraints of low-bit-width photonic hardware. First, it incorporates a gradient scaling factor g, which balances the update dynamics between the learnable step size s and the layer weights w, promoting stable convergence even in high-noise, low-precision regimes. Second, the AWGN block is placed directly after the photonic MAC operation, within the quantization-aware forward path, reflecting the point of degradation typically introduced by optical modulators and detectors. These design choices allow AQ-PANN to maintain accuracy under conditions where conventional quantization or noise-aware training schemes degrade, making it well-suited to practical photonic neuromorphic implementations.

Overall, existing strategies such as noise-aware initialization or activation stretching provide useful robustness gains, but they remain limited because they abstract noise at a coarse level and do not reflect how errors actually arise in photonic MAC pipelines. For ease of comparison, Table 1 summarizes the main design dimensions of these approaches, highlighting differences in quantization strategy, noise awareness, and photonic-specific modeling.

AQ-PANN addresses this gap by jointly learning the quantization step size and injecting noise at the precise point where photonic distortions occur. In doing so, it captures both the quantization dynamics and the physical

Table 1 Comparison of AQ-PANN with representative quantization and noise-aware training methods for photonic neural networks

Method	Learned Step Size	Noise-Aware	Per-MAC Noise	Photonic-Aware Design
DoReFa [30]	✗	✗	✗	✗
PACT [31]	✗	✗	✗	●
Activation Stretching [28]	✗	✓	✗	✓
Noise-Aware Training [18]	✗	✓	✗	●
LSQ [26]	✓	✗	✗	✗
AQ-PANN (this work)	✓	✓	✓	✓

✓ feature explicitly supported; ● feature partially or indirectly addressed; ✗ feature not supported

degradation mechanism, making it more faithful to hardware behavior and more resilient under low-bitwidth constraints.

3 Adaptive-Quantization photonic-aware neural networks (AQ-PANN)

Photonic hardware introduces distinctive challenges for quantized neural networks, particularly due to the presence of analog noise and low-resolution constraints. In this context, AQ-PANN introduces a quantization-aware training scheme where the additive white AWGN, representative of typical impairments in photonic front-ends, is injected immediately after the MAC operation. This placement more accurately captures the signal distortion introduced by physical hardware components such as modulators and photodetectors, and its impact on the quantized output. The remainder of this section formalizes the components of this approach, including the forward computation, the quantization strategy, and the step size update rule.

3.1 Noise-aware PANN

There are many effects which contribute to noise in short-range photonic systems used for neuromorphic applications, those stemming from optical signal generation and detection are laser relative intensity noise (RIN), thermal noise, and shot noise [27]. Additionally, the ADC and Digital to Analog Conversion (DAC) conversion processes introduce additional noise, and the nonlinear transfer functions of the modulators further distort the signal.

These processes inherent to photonic hardware ultimately limit the achievable bit precision, and degrade the inference accuracy. As demonstrated in [27], incorporating noise-aware training techniques tailored to the hardware noise characteristics, modeled as AWGN, can improve resilience to these impairments, enabling robust photonic neural networks even under experimentally measured noise levels.

3.2 Modelling noise in PANN

A generic NN layer can be described as $y = \varphi[f(w, x) + b]$, with full-precision weights w, input x, linear operation of the layer f, output activation function φ, and bias b.

When considering noise, the output of a layer is represented by a probability distribution. As discussed in [18], the noise sources of the basic building blocks of a photonic neuron can be modeled as additive white Gaussian noise (AWGN). Hence, the output of the linear part of the layer can be defined as:

$$\Pi = f(w, x) + \mathcal{N}(0, \sigma^2),$$

with $\mathcal{N}(0, \sigma^2)$ representing the AWGN process with zero mean and variance σ^2.

Focusing on the training phase, [5] showed that direct quantization of floating-point parameters results in a severe accuracy loss due to the mismatch between the continuous weight updates during training and the fixed-point constraints of photonic hardware during inference. Hence, a quantization-aware training scheme should be

exploited. The details of our approach, which builds upon the LSQ method [26], are described in the following, where we address these challenges and illustrate the noise-aware quantization process.

The per-MAC AWGN model adopted in this work is a first-order abstraction that captures the dominant fast fluctuations introduced by optical shot noise, thermal noise, and electronic readout. More complex impairments are certainly present in practical photonic systems, including quasi-static drifts, thermal crosstalk across interferometers, laser relative-intensity noise (RIN), and modulator nonlinearities such as clipping and transfer-curve curvature. These effects are not strictly additive and can in principle break the equivalence with a pure Gaussian model. However, prior studies have shown that AWGN is sufficient to reproduce the main degradation trends of photonic neural networks under low-bit quantization [18, 27], while non-Gaussian effects typically manifest as slow bias shifts or bounded nonlinear distortions that can be compensated by calibration or system-level control.

For this reason we restrict our analysis to the AWGN case, which offers a tractable yet realistic framework for assessing the robustness of quantization schemes.

In the presence of quasi-static drift or multiplicative noise, distortions would mainly appear as persistent offsets in the computed values or as noise whose strength depends on the signal magnitude, rather than as independent random perturbations at each operation. In such regimes, AQ-PANN is expected to generalize by adapting the learned quantization step sizes accordingly. Extensions to explicitly incorporate multiplicative or correlated noise processes, or to explore non-uniform quantization strategies tailored to these perturbations, are therefore left as complementary directions and reserved for future experimental validation.

3.3 Quantization process

The AQ-PANN approach adopts a quantization method that introduces a trainable step size s for each layer, dynamically adjusting the quantization granularity based on data distribution and noise levels.

The quantization process maps input data v into a scaled and clipped quantized representation, denoted as $\bar{v}$. This process consists of two main steps. First, the intermediate, integer-scaled quantization $\bar{v}$ is computed as:

$$\bar{v} = \left\lfloor \text{clip}\left(\frac{v}{s}, -Q_N, Q_P\right)\right\rceil,$$

where $\text{clip}(x, a, b)$ restricts x to the interval $[a, b]$, Q_N and Q_P define the lower and upper quantization bounds, and s is the trainable step size. This step constrains the quantized data to an effective precision range, mitigating the effects of noise.

The final quantized output v_q is then obtained by scaling $\bar{v}$ back by s:

$$v_q = \bar{v} \cdot s.$$

When noise is included, the output of the layer becomes:

$$y = \varphi\left[f(w_q, x_q) + \mathcal{N}(0, \sigma^2)\right],$$

where $w_q = v_q(w)$ and $x_q = v_q(x)$ represent the quantized weights and inputs, and $\varphi(\cdot)$ is the activation function. If activations are also quantized, the resulting output is:

$$y_q = v_q(y).$$

To address noise in activations, the AQ-PANN quantizes them as part of the layer's output without requiring an additional quantization step, ensuring the process remains streamlined. As illustrated in Fig. 1, noise can cause the layer output to fluctuate across quantization boundaries. While values falling within the correct interval (green region) are preserved, even small perturbations may push them into an adjacent interval (gray region), leading to incorrect computations.

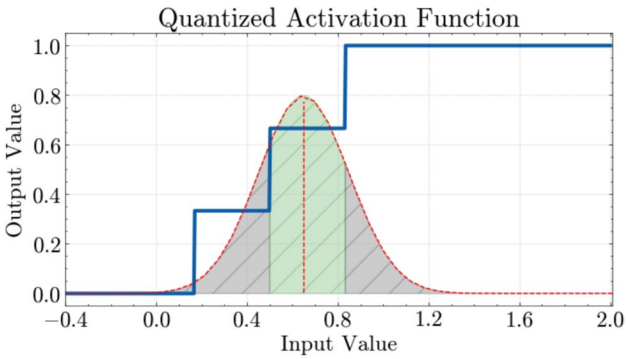


Fig. 1 Impact of noise in the quantization process. The layer output may remain in the proper quantization interval (green area in the figure), or it may fluctuate to an undesired quantized interval (grey area), resulting in a wrong computation.

The AQ-PANN method dynamically adjusts the quantization interval by learning s during training. This dynamic adjustment eliminates the need for manually tuning noise thresholds, as the learned s compensates for variations in data distribution and noise levels. By scaling the quantization process layer by layer, AQ-PANN provides stability and precision under noisy conditions. Unlike previous work [28], which employs a stretching strategy for nonlinear activation functions to enhance noise resilience by increasing the probability of outputs falling within the correct quantization interval, the AQ-PANN approach directly incorporates a trainable step size s, enabling dynamic adjustment of quantization granularity during training. Furthermore, it is worth noting that adjusting the quantization intervals of the activation function does not impact the final accuracy because the quantization-aware scheme, applied during training to mitigate quantization-induced losses, effectively accounts for this modification [5].

3.4 Gradient calculation and straight-through estimator

To enable learning of s during training, the AQ-PANN method defines a gradient for v_q with respect to s that accurately reflects transitions across quantization bins. This is crucial because small changes in s near bin boundaries can cause shifts in the quantized output, which in turn could affect the model accuracy. The gradient of v_q with respect to s is defined as follows:

$$\frac{\partial v_q}{\partial s} = \begin{cases} -\frac{v}{s} + \left\lfloor \frac{v}{s} \right\rceil & \text{if } -Q_N < \frac{v}{s} < Q_P, \\ -Q_N & \text{if } \frac{v}{s} \leq -Q_N, \\ Q_P & \text{if } \frac{v}{s} \geq Q_P. \end{cases} \tag{1}$$

Using the STE [33] for the gradient approximation, AQ-PANN handles the discontinuities caused by rounding. This gradient formulation enables continuous updates to s, allowing quantization boundaries to dynamically adapt to the data distribution. This adaptability to local data characteristics helps the quantization process remain robust and accurate even in presence of AWGN.

A schematic of the full approach is presented in Fig. 2, which summarizes the layer-wise flow and the per-MAC noise placement. The AQ-PANN method leverages this dynamic adjustment of the step size s during training to enhance noise resilience. By learning s, AQ-PANN mitigates quantization errors caused by AWGN, ensuring higher accuracy during inference.

3.5 Balancing step size and weight updates

A critical aspect of the AQ-PANN approach is maintaining balance between the updates to s and the updates to other parameters, especially the weights w in each layer. Overly aggressive updates to s can destabilize the training process or overshoot optimal quantization levels. Conversely, conservative updates can slow convergence and reduce quantization efficiency. To achieve balanced updates, AQ-PANN introduces a scaling factor g to ensure that the average magnitude of the update of s is comparable to that of w. This balance is achieved by defining a ratio R that measures relative update magnitudes:

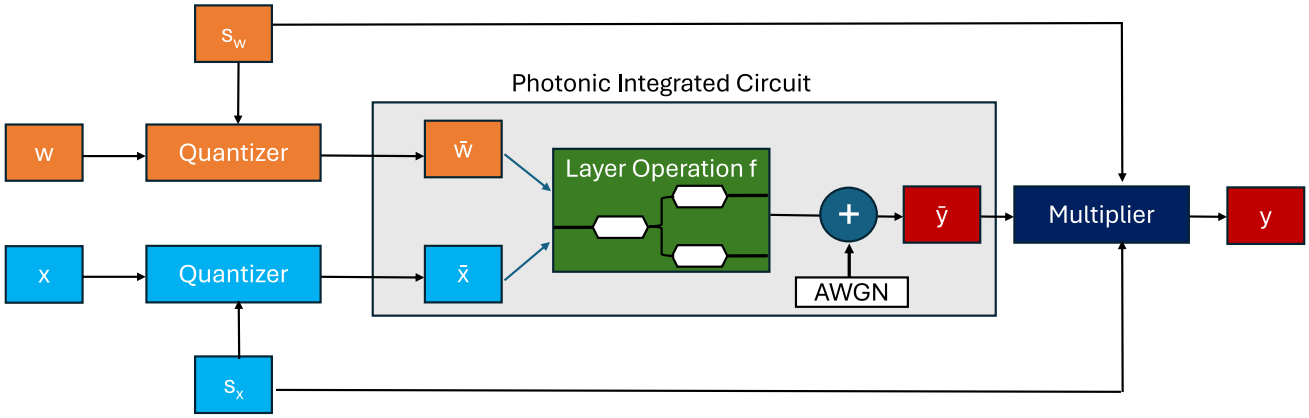


Fig. 2 AQ-PANN layer-wise training flow. Inputs x and weights w are quantized with learnable step sizes s_x, s_w, then processed by a photonic MAC (e.g., MZI mesh [32]). Each quantizer applies three operations: normalization by s, rounding and clipping to integer levels, and a final multiplication (multiplier) by s to rescale the output back to the original domain. AWGN, representative of distortions introduced by photonic components, is added after the MAC. The Straight-Through Estimator (STE) allows gradients to flow through the quantizers, and step sizes are updated during training. The same pipeline applies across all layers.

$$R = \frac{\frac{\partial L/\partial s}{s}}{\frac{\|\partial L/\partial w\|}{\|w\|}}, \tag{2}$$

where L is the loss function, w represents the weights, and $\|\cdot\|$ denotes the ℓ_2-norm. Ideally, R should be close to 1, ensuring proportionate updates. To achieve this, AQ-PANN sets the gradient scale g for s as:

$$g = \frac{1}{\sqrt{N_W Q_P}}, \tag{3}$$

where N_W is the number of weights in a layer and Q_P is the upper quantization limit. This scaling factor stabilizes the training by controlling the magnitude of s updates relative to weight updates, preventing s from diverging excessively during optimization [26].

4 Experiments and results

4.1 Experimental setup

Concerning the experiments, we focused on three image-classification use-cases, exploiting MNIST[34], Street View House Numbers (SVHN) [35], and CIFAR-10 dataset [36]. This choice is motivated by the fact that we want to validate the proposed method with models with varying architectural complexity, with the PANN architecture for the MNIST scenario having fewer parameters than the ones used in the SVHN and CIFAR-10 experiments. Furthermore, in the experiments we have considered both PACT [31] and DoReFa quantizers [30] as a benchmark in order to have a comparison with other methodologies used in PANN training [8]. Additionally, AQ-PANN is compared with the previous work [28], a noise-resilience approach that modifies the activation functions through a stretching strategy.

We considered 2, 3, and 4 bits as possible bitwidths due to PANN resolution constraint, as discussed in Sec. 1. Higher resolutions can be achieved with a lower operation speed due to the underlying photonic hardware [16]. Additionally, different AWGN noise levels have been tested simulating different noise conditions. Specifically, we considered a noise standard deviation σ of 0.1, 0.3, and 0.5, corresponding to low, medium, and high noise

intensity, respectively. These levels were chosen to reflect realistic operational scenarios [14]. In the following, results and discussions on MNIST and SVHN are reported.

In the experimental evaluation, AQ-PANN is compared against three baseline methods: DoReFa [30], PACT [31], and the activation stretching approach [28]. These methods were selected because they represent complementary strategies for handling low-precision and noise effects, ranging from fixed-step quantization (DoReFa), to learnable activation clipping (PACT), to photonic-specific noise mitigation through linear reshaping (stretching).

It is worth emphasizing that all baselines are trained under identical conditions (same architectures, optimizers, and bitwidths), ensuring a fair comparison. Under these controlled settings, the relative differences can therefore be attributed to the distinct design choices of each method. As will be shown, AQ-PANN consistently exhibits smoother accuracy decay across noise levels compared to DoReFa, PACT, and stretching, highlighting the effectiveness of its adaptive quantization strategy.

4.2 MNIST

4.2.1 Architecture and training setup

The network used for MNIST experiment leverages three convolutional layers, each one followed by a batch normalization (for stable and faster training) and an average pooling layer. The latter is preferred over max pooling due to its linearity, which makes it more suitable for implementation in photonic systems [8]. The convolutional layer starts with 8 filters, increasing to 16 in the second layer, and expanding further to 64 in the third layer. After the convolutional part, the PANN architecture exploits a series of Fully-Connected (FC) layers, designed to progressively decrease the number of neurons, starting from 192, followed by 164 and 128 neurons in subsequent layers. The final layer has 10 neurons to perform classification on MNIST digits. This architecture fulfills all PANN architectural constraints [8]. The models have been trained for 30 epochs and a batch size of 256, exploiting Adam as optimizer with a learning rate of 10^{-3}. Since the dataset is perfectly balanced, the accuracy metric is used to evaluate the performance of the PANN architecture with confidence intervals at 95% confidence level computed by repeating both the training and the testing procedures 10 times. Table 2 reports a concise summary of the MNIST PANN architecture and the associated training setup.

4.2.2 Results

Results for 2, 3, and 4 bitwidth configurations are depicted in Fig. 3a, b, and c, respectively. Across 2-bit, 3-bit, and 4-bit configurations, all the methods, i.e., AQ-PANN, DoReFa, PACT, and the 0.75 stretching approach, show varying degrees of resilience to noise as σ increases from 0.1 to 0.5. At 2 bits, AQ-PANN experiences the smallest accuracy drop (from about 94.4% to 91.5%), while the stretching approach decreases moderately (from around 89.4% to 85.5%), and DoReFa falls most sharply (from roughly 90.5% down to 58.2%). In addition, the low bitwidth prevents PACT from exploiting its clipping mechanism effectively, leading to poor performance at 2 bits.

Table 2 MNIST PANN architecture and training setup

Component	Configuration
Input	28×28 grayscale
Conv stack	3 conv layers with filters [8, 16, 64]
Normalization	BatchNorm after each conv
Pooling	Average pooling (instead of max pooling)
FC stack	$192 \rightarrow 164 \rightarrow 128 \rightarrow 10$
Epochs/Batch size	30 epochs / batch 256
Optimizer/LR	Adam / 10^{-3}
Metric	Accuracy (95% CI over 10 runs)

 https://doi.org/10.1007/s00521-026-11984-6

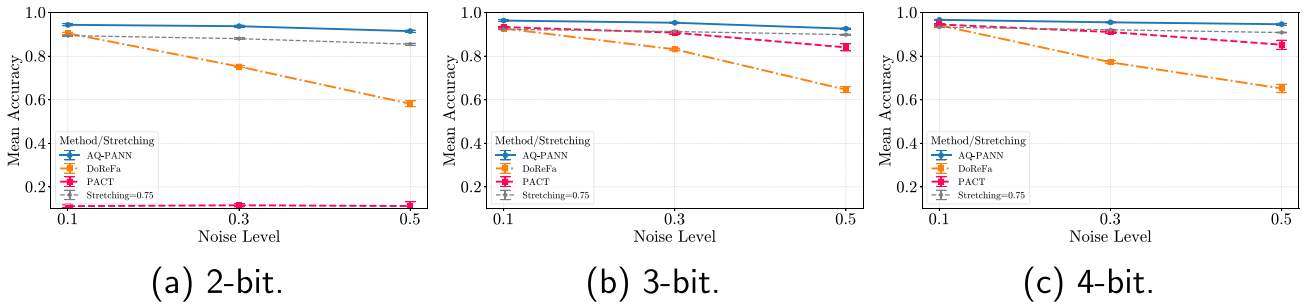
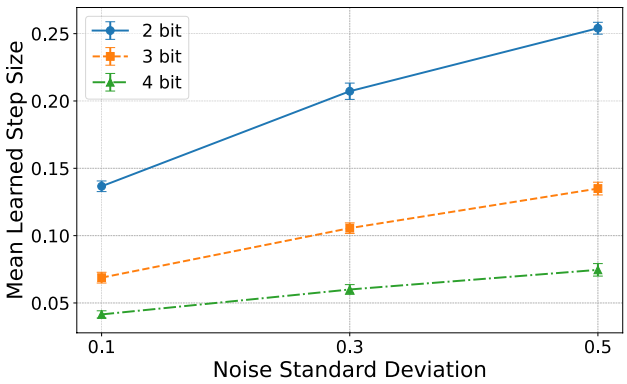


Fig. 3 MNIST accuracy results.

Fig. 4 Learned step size vs Noise Standard Deviation for MNIST.



Moving to 3 bits improves overall accuracies: AQ-PANN decreases from about 96.3% to 93%, the stretching method from 92.3% to 90%, DoReFa from 92.7% to 64.6%, and PACT from 93.3% to 84%.

Finally, at 4 bits, the highest noise robustness is observed, with AQ-PANN maintaining roughly 96.6% to 94.6%, the stretching method about 93.3% to 90.8%, PACT from 94.6% to 85.2% and DoReFa showing a steeper decline (from near 94.2% to 65.2%). In every case, AQ-PANN consistently exhibits superior noise resilience, while the stretching approach offers moderate stability, and DoReFa remains the most affected as noise levels rise.

Across all bitwidth configurations, AQ-PANN demonstrates a consistent and controlled reduction in accuracy as noise increases, with accuracy drops remaining around 2-4% from low to high noise levels in each bitwidth scenario. This stability highlights AQ-PANN's ability to predictably mitigate the impact of noise across quantization depths. In contrast, DoReFa exhibits a more variable response, with substantial accuracy drops at lower bitwidths (e.g., 32% at 2 bits), which do not decrease as bitwidth increases. This highlights that DoReFa's resilience to noise remains inconsistent across configurations, unlike AQ-PANN, which demonstrates both higher accuracy and greater stability in noise handling at all bitwidth levels. The stretching approach, while less robust than AQ-PANN, shows moderate stability across noise levels, providing a reliable but less optimized alternative. PACT, by contrast, is largely ineffective at 2 bits but becomes more competitive at 3 and 4 bits, where its clipping mechanism can better exploit the increased quantization levels.

4.2.3 Learned step size analysis

Another analysis was carried out, regarding the learned step size s in our AQ-PANN method. Figure 4 depicts the learned step size for the second convolutional layer of the PANN architecture.

For the 2-bit configuration the learned step size is around 0.13 when the noise is low, reaching approximately 0.26 at the highest noise level ($\sigma = 0.5$). This steady increase in step size indicates that the model is adjusting to

larger gradients, which likely helps to stabilize the training under higher noise by coarsening the weight updates to maintain effective learning. In the 3-bit setup we observe a similar trend with slightly smaller step sizes from 0.07 with low noise to about 0.14 for $\sigma = 0.5$. Finally, for the 4-bit configuration the learned step size is even smaller, from around 0.04 with low noise and increasing modestly to about 0.08 at the highest noise level.

The observed decrease in the learned step size s as the bitwidth increases arises from the model's optimization process during training. Although s is a learned parameter in our method, the model adapts it to effectively utilize the available quantization levels determined by the bitwidth. Specifically, as the bitwidth b increases, the number of quantization levels $L = 2^b$ increases exponentially, providing the potential for finer quantization. To take advantage of this increased resolution within the fixed value range R, the model learns a smaller s, resulting in narrower quantization intervals. This adjustment allows the AQ-PANN method to represent data with greater precision, capturing subtle variations crucial for enhancing performance.

On the other hand, the observed increase in the learned step size s with higher noise levels reflects the model's adaptive mechanism to maintain robustness in the presence of noise. Higher noise levels introduce greater variability in the input signals, which can cause small perturbations that may push values across quantization boundaries, leading to quantization errors and degraded performance. By increasing the step size s, the model effectively widens the quantization intervals, reducing the number of quantization levels within the fixed range. This coarser quantization makes the model less sensitive to small fluctuations caused by noise because larger intervals mean that minor changes are less likely to result in a different quantized value.

4.3 SVHN

4.3.1 Architecture and training setup

In the SVHN dataset experiments, a deeper PANN architecture compared to the one used for MNIST was used, reflecting the increased complexity of this classification task.

The model comprises 8 convolutional layers, each followed by batch normalization for stable and faster training. Even in this scenario, average pooling layers are used instead of max pooling due to their linearity. The PANN architecture starts with 16 filters in the first convolutional layer. The number of filters then doubles in each of the next three layers, increasing to 32, 64, and 128 filters, respectively. Following this pattern, the subsequent layers continue to expand with 164 filters in the fifth layer, 192 in the sixth, and 256 in the seventh layer. The eighth and final convolutional layer contains 200 filters. Following the convolutional layers, the PANN architecture incorporates a series of FC layers to reduce the dimensionality before classification. The first FC layer reduces the feature size from 200 to 92 neurons and is followed by batch normalization to maintain training stability. The second FC layer further reduces the size from 92 to 10 neurons, corresponding to the ten classes in the SVHN dataset. The training parameters are the same as those used in the MNIST experiment.

Since the dataset is not perfectly balanced, a class weighting method has been applied during training and the F1-score is used to evaluate the performance of the model, with confidence intervals at 95% confidence level

Table 3 SVHN PANN architecture and training setup

Component	Configuration
Input	32×32 RGB
Conv stack	8 convolutional layers with filters [16, 32, 64, 128, 164, 192, 256, 200]
Normalization	Batch normalization after each convolution
Pooling	Average pooling (photonic-friendly)
FC stack	$200 \rightarrow 92 \rightarrow 10$
Epochs/Batch size	30 epochs / batch size 256
Optimizer/Learning rate	Adam / 10^{-3}
Evaluation metric	F1-score (95% CI over 10 runs)

computed by repeating 10 times the training procedure. The SVHN PANN architecture and training hyperparameters are summarized in Table 3.

4.3.2 Results

Results for 2, 3, and 4 bitwidth configurations are depicted in Fig. 5a, b, and c, respectively.

Across the 2-bit, 3-bit, and 4-bit configurations, the three methods, namely AQ-PANN, DoReFa, and the 0.75 stretching, exhibit distinct trends in noise resilience as σ increases from 0.1 to 0.5. At 2 bits, AQ-PANN demonstrates the strongest robustness, with accuracy dropping only slightly from 92.6% to 91.3%. The stretching method shows moderate resistance, decreasing from 88.5% to 76.8%, whereas DoReFa is heavily impacted, falling sharply from 81.6% to 38.8%. Even in this case, PACT proves to be ineffective at 2 bits, maintaining an F1-score of $\approx 12\%$ across all noise configurations.

In the 3-bit scenario, both DoReFa and the stretching improve in handling noise, but the gap between them remains evident. AQ-PANN retains its lead, with accuracy decreasing from 92.9% to 91.7%. PACT achieves intermediate performance: it remains competitive at low noise levels (around 87% F1-score) but exhibits a sharp decline as the noise intensity increases, going down to about 12%. The stretching method appears less robust with respect to the the 2-bit scenario, moving from 89.6% to 81.4%, while DoReFa suffers a more pronounced drop, sliding from 87.4% to 44.2%.

With 4 bits, the overall noise resistance reaches its peak, particularly for AQ-PANN, which remains highly stable, reducing from 93.3% to 92%. PACT performs strongly in this regime at low noise (around 89%), but its effectiveness rapidly vanishes as noise grows, with accuracy collapsing to about 12% at the highest noise level. The stretching approach decreases slightly from 91% to 82.5%. DoReFa continues to show the steepest decline, dropping from 88% to 46.8%.

Across all bitwidth configurations, AQ-PANN demonstrates a consistent and controlled reduction in accuracy as noise increases, with accuracy drops remaining between 1 and 2% moving from low to high noise levels in each bitwidth scenario. This stability highlights AQ-PANN's ability to mitigate the impact of noise across quantization depths in a predictable way. The stretching method is less robust than AQ-PANN, being a viable alternative in scenarios where moderate noise resilience is sufficient. PACT, on the other hand, proves effective only under low-noise conditions and with 3-4 bits, but collapses sharply as noise grows, indicating limited applicability in realistic high-noise regimes. In contrast, DoReFa exhibits a more variable response, with substantial accuracy drops at lower bitwidths that decrease only slightly as bitwidth increases, suggesting that DoReFa's resilience to noise lacks the steady improvement seen with AQ-PANN. This difference underscores AQ-PANN's advantage in maintaining both high accuracy and stable noise resistance as bitwidth increases.

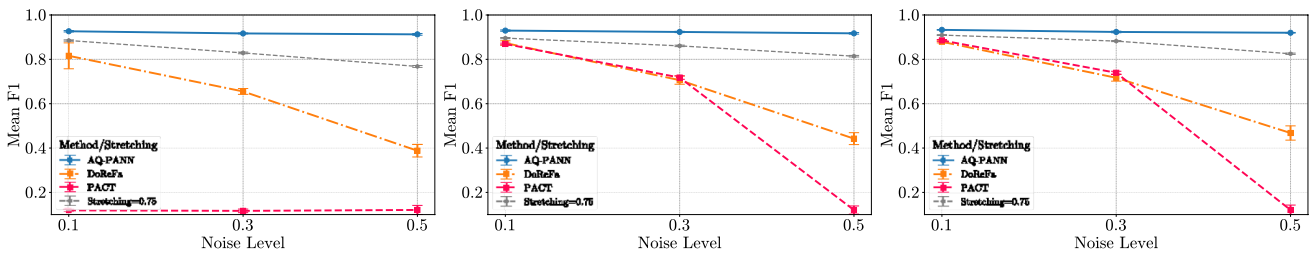


Fig. 5 SVHN F1-Score results.

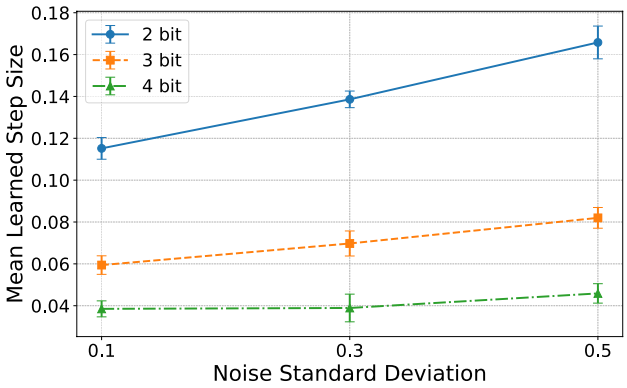


Fig. 6 Learned step size vs Noise standard deviation for SVHN.

Table 4 CIFAR-10 PANN architecture and training setup

Component	Configuration
Input	32×32 RGB
Conv stack	6 convolutional layers with filters [32, 32, 64, 64, 128, 128]
Normalization	Batch normalization after each convolution
Pooling	Average pooling (photonic-friendly)
FC stack	$128 \rightarrow 10$
Epochs/Batch size	100 epochs / batch size 256
Optimizer/Learning rate	Adam / 10^{-3}
Evaluation metric	Accuracy (95% CI over 10 runs)

4.3.3 Learned step size analysis

The learned step size s is reported in Fig. 6. In the 2-bit configuration, the learned step size begins around 0.11 at low noise levels and steadily rises to approximately 0.16 at a noise level of 0.5, suggesting that the model adjusts to larger gradients to stabilize training under higher noise by coarsening weight updates. In the 3-bit setup a similar pattern appears with slightly smaller step sizes, starting close to 0.06 in the absence of noise and increasing to about 0.08 as noise reaches 0.5. For the 4-bit configuration the learned step size is even smaller and remains relatively stable, ranging from about 0.039 to 0.045, indicating that higher bitwidth provides finer granularity and less need for adjustment in response to noise.

The same observations made for the MNIST dataset are valid here: (i) as bitwidth increases, the model benefits from finer quantization levels, allowing it to maintain or enhance performance with greater precision; (ii) as noise levels rise, the model adapts by coarsening quantization, thus enhancing robustness and reducing sensitivity to small fluctuations due to noise.

4.4 CIFAR-10

4.4.1 Architecture and training setup

For the CIFAR-10 dataset, a PANN architecture of similar style to the MNIST and SVHN cases was employed, but adjusted to account for the complexity of the task. The network consists of six convolutional layers (two with 32 filters, two with 64 filters, and two with 128 filters), each followed by batch normalization and average pooling, and completed with two fully connected layers, the first with 128 neurons and the last with 10 neurons for classification into the CIFAR-10 classes. Training was carried out for 100 epochs with a batch size of 256, using Adam with a learning rate of 10^{-3}. Accuracy is reported as the main performance metric, with confidence intervals at 95% confidence level computed by repeating the training and testing procedures 10 times. The CIFAR-10 PANN architecture and training hyperparameters are summarized in Table 4.

 https://doi.org/10.1007/s00521-026-11984-6

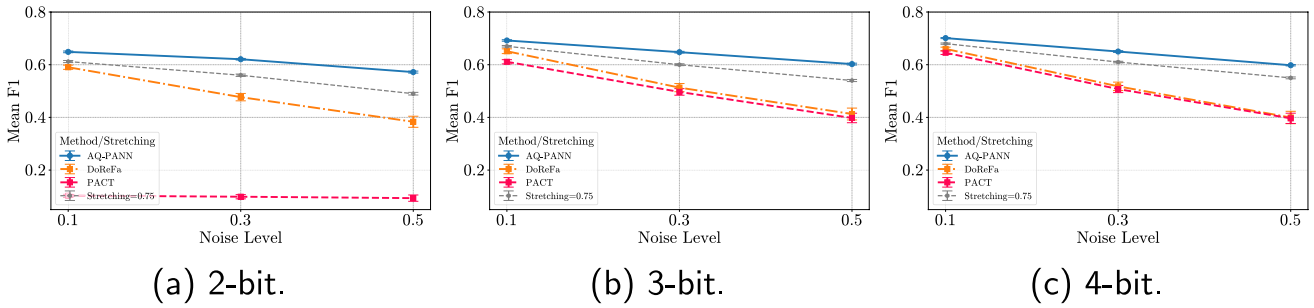


Fig. 7 CIFAR-10 accuracy results.

4.4.2 Results

Results for 2, 3, and 4 bitwidth configurations are shown in Fig. 7a, b, and c, respectively. At 2 bits, AQ-PANN proves the most resilient, dropping from about 64.9% at $\sigma = 0.1$ to 57.2% at $\sigma = 0.5$. DoReFa performs worse, decreasing from 59.0% to 38.3%, while PACT remains ineffective at this precision, staying near random performance ($\approx 10\%$). The stretching approach provides intermediate stability, maintaining results between 61.2% and 49.0% across noise levels.

Moving to 3 bits, AQ-PANN maintains the best stability, decreasing smoothly from 69.2% to 60.2%. DoReFa drops more steeply from 65.1% to 41.3%, and PACT follows a similar pattern, starting at 61.1% under low noise and falling to 39.8% at the highest noise. Stretching performs moderately well in this case, with results ranging from 67.0% to 54.0%, offering a compromise between the robustness of AQ-PANN and the instability of DoReFa and PACT.

Finally, in the 4-bit case, AQ-PANN remains clearly superior, with accuracy reducing only from 70.1% to 59.8%. DoReFa starts close at 66.0% but drops below to 40%, while PACT shows competitive performance at low noise (64.5%) yet drops significantly to 39.6% under strong noise. The stretching strategy provides intermediate performance, decreasing from 68.0% to 55.0% as noise increases.

Across all bitwidths, AQ-PANN demonstrates the smoothest and most predictable accuracy decay as noise increases. In contrast, DoReFa and PACT both suffer substantial drops, particularly under higher noise. The stretching approach, while less robust than AQ-PANN, consistently shows more stability than DoReFa and PACT, confirming the same intermediate trend observed in MNIST and SVHN.

Finally, a comprehensive summary of the results obtained across MNIST, SVHN, and CIFAR-10, covering all evaluated bitwidths and noise levels, is reported in Table 5.

4.4.3 Learned step size analysis

The learned step size s for CIFAR-10 is reported in Fig. 8.

At 2 bits, s grows from about 0.06 at low noise to 0.13 at $\sigma = 0.5$. For 3 bits, it increases more modestly from 0.029 to 0.055, and for 4 bits it remains the smallest, ranging from 0.016 to 0.029.

These results confirm the same trend observed in MNIST and SVHN: higher bitwidth naturally reduces the learned step size to exploit finer quantization intervals, while larger noise levels drive the model to enlarge s to coarsen the quantization and mitigate noise-induced errors.

4.5 Weight distribution and quantization intervals

In this section, we analyze and compare the weight distributions generated by DoReFa and AQ-PANN quantization schemes. This comparison highlights the differences in the quantization levels and the resulting weight distributions, with implications for model robustness to noise. We compare weight distributions and quantization

Table 5 Complete summary across datasets, bitwidths, and noise levels

Dataset	Bitwidth	Method	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$
MNIST					
2-bit		DoReFa	90.5	75.2	58.2
		PACT	11.2	11.6	11.2
		Stretching	89.4	88.0	85.5
		AQ-PANN	944	93.7	91.5
3-bit		DoReFa	92.7	83.1	64.6
		PACT	93.3	90.8	84.0
		Stretching	92.3	91.3	89.9
		AQ-PANN	96.3	95.3	92.6
4-bit		DoReFa	94.2	77.2	65.2
		PACT	94.6	91.1	85.2
		Stretching	93.3	92.1	90.8
		AQ-PANN	96.7	95.5	94.6
SVHN					
2-bit		DoReFa	81.6	65.5	38.8
		PACT	11.9	11.7	12.1
		Stretching	88.5	82.9	76.8
		AQ-PANN	92.7	91.7	91.3
3-bit		DoReFa	87.4	70.6	44.2
		PACT	86.9	71.8	12.0
		Stretching	89.6	86.1	81.4
		AQ-PANN	93.0	92.3	91.7
4-bit		DoReFa	87.9	71.7	46.8
		PACT	88.5	73.9	12.1
		Stretching	91.0	88.2	82.5
		AQ-PANN	93.3	92.4	92.0
CIFAR-10					
2-bit		DoReFa	59.0	47.7	38.3
		PACT	10.3	9.9	9.4
		Stretching	61.2	56.0	49.0
		AQ-PANN	64.9	62.1	57.2
3-bit		DoReFa	65.1	51.3	41.3
		PACT	61.1	49.7	39.8
		Stretching	67.0	60.0	54.0
		AQ-PANN	69.2	64.8	60.2
4-bit		DoReFa	66.0	51.8	40.0
		PACT	64.5	50.8	39.6
		Stretching	68.0	61.0	55.0
		AQ-PANN	70.1	65.0	59.8

intervals for the second convolutional layer of the MNIST PANN architecture for DoReFa and AQ-PANN in Fig. 9. The weight distribution is shown for both the initial and final models, representing the beginning and the end of the training process.

DoReFa quantization, as shown in Fig. 9a, applies a fixed set of quantization intervals uniformly across the entire range from −1 to 1. Unlike AQ-PANN, which adapts its intervals to the distribution of the weights, DoReFa's intervals are evenly spaced and do not adjust based on where the majority of weight values lie. The uniform spacing in DoReFa means that, under noisy conditions, small perturbations in weight values are more likely to push weights across these fixed boundaries. This misalignment between quantization intervals and the actual weight distribution impacts DoReFa's noise resilience, leading to degradation in accuracy as noise increases. Consequently, even a minor noise can lead to significant reductions in accuracy because DoReFa's fixed intervals lack the flexibility needed to accommodate weight shifts without crossing quantization boundaries.

 https://doi.org/10.1007/s00521-026-11984-6

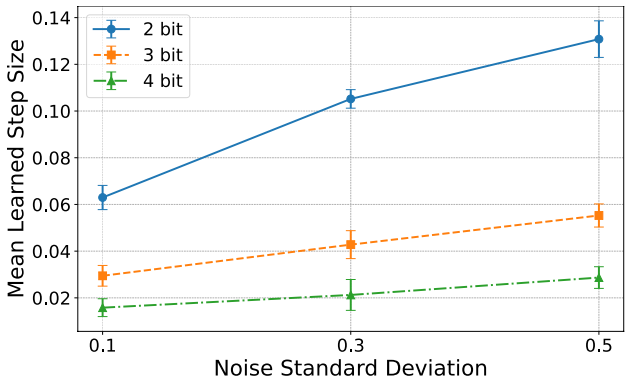
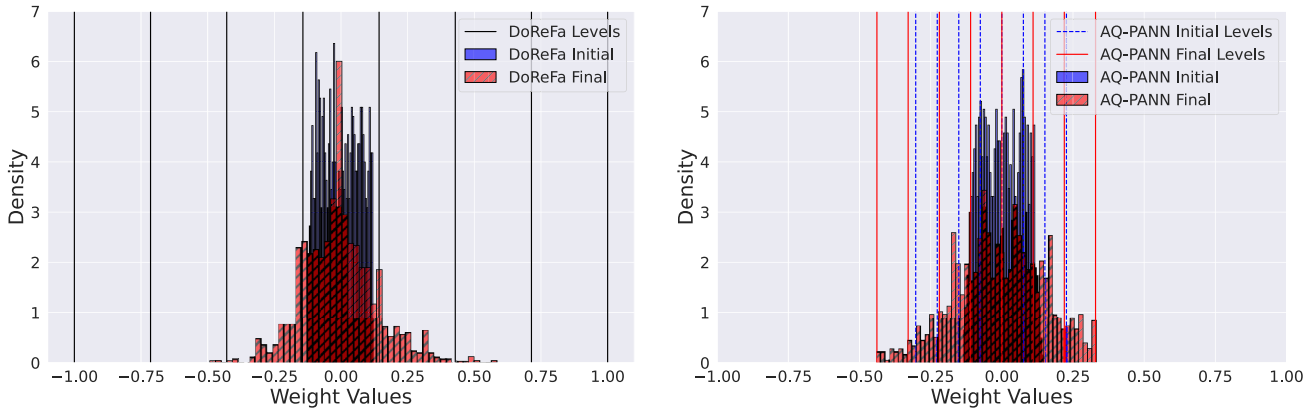


Fig. 8 Learned step size vs Noise Standard Deviation for CIFAR-10.



(a) DoReFa quantization. (b) AQ-PANN quantization.

Fig. 9 Weight distribution and quantization intervals for the second convolutional layer of the MNIST PANNs architecture.

In contrast, AQ-PANN quantization, reported in Fig. 9b, adapts its quantization intervals to the actual distribution of weights, concentrating intervals within a narrower range, approximately from −0.4 to 0.3. This range aligns closely with the natural distribution of the weights, capturing the majority of weight values within its bounds. Unlike DoReFa, AQ-PANN's range is tailored to where weights are most densely packed, particularly around zero.

The noise resilience of AQ-PANN can be attributed to its learnable quantization intervals, which are adjusted dynamically during training to account for noise. AQ-PANN can adaptively adjust the spacing of its intervals to match both the weight distribution and the expected noise patterns. As a result, the intervals are positioned strategically to stabilize weights against the typical shifts caused by noise, especially in high-density regions near zero. In contrast, fixed-interval schemes like DoReFa lack this adaptability, as they do not adjust to the noise characteristics during training, resulting in higher quantization errors under similar conditions. The learnable and noise-aware intervals of AQ-PANN enable the model to maintain higher accuracy and stability by proactively shaping the quantization levels to withstand expected noise, giving it a significant advantage in noisy, low-bitwidth scenarios.

4.6 Physical interpretation of learned step size

The changes in the learned step size s provide an interpretable link between AQ-PANN and the physical constraints of photonic hardware. Across all evaluated datasets, s consistently increases with the noise standard deviation, reflecting the need to widen quantization intervals as the effective noise floor of the photonic front-end

grows. For instance, in the MNIST experiment, the learned step size in a representative convolutional layer increases from approximately $s \approx 0.13$ at low noise ($\sigma = 0.1$) to $s \approx 0.26$ at high noise ($\sigma = 0.5$) in the 2-bit configuration, corresponding to a doubling of the quantization interval width. A similar trend is observed for SVHN, where s rises from about 0.11 to 0.16 under the same noise increase, and for CIFAR-10, where s grows from roughly 0.06 to 0.13 at 2 bits.

From a hardware perspective, this behavior is consistent with the limited effective number of bits achievable in analog photonic systems, where higher noise levels caused by shot noise, thermal noise, and ADC uncertainty reduce the number of reliably distinguishable signal levels. By enlarging the quantization step size, AQ-PANN reduces the probability that noise-induced fluctuations push signals across quantization boundaries, thereby stabilizing inference under noisy operating conditions.

Conversely, when the available bitwidth increases, AQ-PANN systematically converges to smaller step sizes, exploiting the finer quantization granularity enabled by improved resolution. For example, in the MNIST case, the learned step size decreases from values on the order of 0.13–0.26 at 2 bits to approximately 0.04–0.08 at 4 bits across the same noise range, while CIFAR-10 exhibits a reduction from $s \approx 0.06$–0.13 at 2 bits to $s \approx 0.016$ –0.029 at 4 bits. These values indicate that AQ-PANN automatically adapts its quantization scale to match the effective signal-to-noise ratio supported by the photonic hardware.

Overall, the learned step size provides a direct indication of how the quantization resolution is adjusted during training to accommodate the noise levels and bitwidth limitations imposed by the photonic hardware. This behavior highlights the photonics-aware nature of AQ-PANN, as the training process naturally aligns the numerical representation with the operating regime of the underlying analog photonic system.

5 Conclusion

In this work, we proposed a noise-resilient quantization-aware training strategy for PANNs, leveraging a dynamically learned step size quantization method to enhance model robustness against typical noise sources in analog photonic hardware. The proposed method, AQ-PANN, combines three key components: (i) a learnable step size jointly optimized with weight and activation quantization, (ii) a gradient-balancing rule to stabilize training under low-bitwidth constraints, and (iii) a training-time injection of additive noise after each photonic MAC to accurately reflect hardware noise. Together, these elements allow the network to adjust its precision and quantization scale in response to the estimated noise variance.

AQ-PANN showed significant improvements over traditional DoReFa quantization, particularly in low-bitwidth scenarios. For instance, on the MNIST dataset, AQ-PANN maintained a mean accuracy of 91.5% in a 2-bit configuration under high noise conditions (noise standard deviation of 0.5), compared to only 65.2% achieved by DoReFa. Similarly, on the SVHN dataset, AQ-PANN achieved a mean accuracy of 91.3% in the 2-bit configuration under high noise, while DoReFa only reached 38.9%. The CIFAR-10 experiment further confirms the robustness of our approach also on more complex datasets: AQ-PANN showed a controlled reduction in accuracy from about 69% at low noise to 60% at the highest noise level for the 4-bit configuration. We also tested AQ-PANN against the 0.75 stretching method, which widens the activation function range to mitigate noise-induced errors. While stretching offers moderate stability, it consistently lags behind AQ-PANN in all tested noise and bitwidth configurations, highlighting the efficacy of our adaptive step size approach.

These results demonstrate the effectiveness of our adaptive quantization strategy in maintaining accuracy under various noise levels, making PANNs viable for deployment in noisy environments. By equipping PANNs models with greater adaptability, we pave the way for energy-efficient, high-speed neuromorphic computing, enabling intelligent systems that can process information while adjusting to their operating environments. Future work will explore adaptive quantization schemes that can autonomously adjust bitwidth and step size based on real-time noise conditions, potentially leading to self-optimizing NNs that adapt to varying operational environments.

In addition, we plan to validate AQ-PANN on photonic hardware prototypes, using measured noise profiles and device impairments to further corroborate the simulation results.

Acknowledgements This work has been sponsored in part by the CHIPS-JU SMARTY project (GA 101140087), funded by the Key Digital Technologies Joint Undertaking and its members, including top up funding by the Italian MUR. It is also supported by the European Union under Italy's National Recovery and Resilience Plan (NRRP) of NextGenerationEU, ("Telecommunica tions of the Future" partnership PE00000001- program "RESTART"), and by the Italian MUR under the FoReLab Project (Departments of Excellence).

Funding Open access funding provided by Scuola Superiore Sant'Anna within the CRUI-CARE Agreement.

Data availability The datasets generated during the current study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

Ethical approval Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E (2018) Deep learning for computer vision: a brief review. Comput Intell Neurosci 2018(1):7068349. https://doi.org/10.1155/2018/7068349
2. He K, Kim DD, Asghar MR (2023) Adversarial machine learning for network intrusion detection systems: a comprehensive survey. IEEE Commun Surv Tutor 25(1):538–566. https://doi.org/10.1109/COMST.2022.3233793
3. Paolini E, Valcarenghi L, Maggiani L, Andriolli N (2024) Real-time network packet classification exploiting computer vision architectures. IEEE Open J Commun Soc 5:1155–1166. https://doi.org/10.1109/OJCOMS.2024.3363082
4. Wang N, Choi J, Brand D, Chen C-Y, Gopalakrishnan K (2018) Training deep neural networks with 8-bit floating point numbers. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. NIPS'18, pp. 7686–7695. Curran Associates Inc., Red Hook, NY, USA
5. Paolini E, De Marinis L, Cococcioni M, Valcarenghi L, Maggiani L, Andriolli N (2022) Photonic-aware neural network: a fixed-point emulation of photonic hardware. In: 2022 27th OptoElectronics and Communications Conference (OECC) and 2022 International Conference on Photonics in Switching and Computing (PSC), pp. 01–03. https://doi.org/10.23919/OECC/PSC53152.2022.9850019
6. Deng L, Li G, Han S, Shi L, Xie Y (2020) Model compression and hardware acceleration for neural networks: a comprehensive survey. Proc IEEE 108(4):485–532. https://doi.org/10.1109/JPROC.2020.2976475
7. Li R, Gong Y, Huang H, Zhou Y, Mao S, Wei Z, Zhang Z (2024) Photonics for neuromorphic computing: fundamentals, devices, and opportunities. Adv Mater 37(2):2312825
8. Paolini E, De Marinis L, Cococcioni M, Valcarenghi L, Maggiani L, Andriolli N (2022) Photonic-aware neural networks. Neural Comput Appl 34(18):15589–15601
9. Nahmias MA, De Lima TF, Tait AN, Peng H-T, Shastri BJ, Prucnal PR (2019) Photonic multiply-accumulate operations for neural networks. IEEE J Sel Top Quantum Electron 26(1):1–18
10. Bandyopadhyay S, Sludds A, Krastanov S, Hamerly R, Harris N, Bunandar D, Streshinsky M, Hochberg M, Englund D (2024) Single-chip photonic deep neural network with forward-only training. Nat Photonics 18(12):1335–1343. https://doi.org/10.1038/s41566-024-01567-z
11. Lin Z, Shastri BJ, Yu S, Song J, Zhu Y, Safarnejadian A, Cai W, Lin Y, Ke W, Hammood M et al (2024) 120 GOPS Photonic tensor core in thin-film lithium niobate for inference and in situ training. Nat Commun 15(1):9081

12. Hu Y, Song Y, Zhu X, Guo X, Lu S, Zhang Q, He L, Franken C, Powell K, Warner H et al (2024) Integrated lithium niobate photonic computing circuit based on efficient and high-speed electro-optic conversion. arXiv preprint arXiv:2411.02734
13. De Marinis L, Catania A, Castoldi P, Contestabile G, Bruschi P, Piotto M, Andriolli N (2022) A codesigned integrated photonic electronic neuron. IEEE J Quantum Electron 58(5):1–10. https://doi.org/10.1109/JQE.2022.3177793
14. Roumpos I, Marinis LD, Kovaios S, Kincaid PS, Paolini E, Tsakyridis A, Moralis-Pegios M, Berciano M, Ferraro F, Bode D, Srinivasan SA, Pantouvaki M, Andriolli N, Contestabile G, Pleros N, Vyrsokinos K (2024) Silicon integrated photonic-electronic neuron for noise-resilient deep learning. Opt Express 32(20):34264–34274. https://doi.org/10.1364/OE.532306
15. Paolini E, De Marinis L, Contestabile G, Gupta S, Maggiani L, Andriolli N (2023) Validation of Photonic Neural Networks in Health Scenarios. In: 2023 International Conference on Photonics in Switching and Computing (PSC). https://doi.org/10.1109/PSC57974.2023.10297132
16. Paolini E, Civerchia F, De Marinis L, Valcarenghi L, Maggiani L, Andriolli N (2022) Photonic-aware Neural Networks for Packet Classification in Beyond 5G Networks. In: 2022 13th International Conference on Network of the Future (NoF). https://doi.org/10.1109/NoF55974.2022.9942486
17. Shi B, Calabretta N, Stabile R (2020) Deep neural network through an InP SOA-based photonic integrated cross-connect. IEEE J Sel Top Quantum Electron 26(1):1–11. https://doi.org/10.1109/JSTQE.2019.2945548
18. Kirtas M, Passalis N, Mourgias-Alexandris G, Dabos G, Pleros N, Tefas A (2023) Robust architecture-agnostic and noise resilient training of photonic deep learning models. IEEE Trans Emerg Top Comput Intell 7(1):140–149. https://doi.org/10.1109/TETCI.2022.3182765
19. Tsakyridis A, Moralis-Pegios M, Giamougiannis G, Kirtas M, Passalis N, Tefas A, Pleros N (2024) Photonic neural networks and optics-informed deep learning fundamentals. APL Photonics 9(1):011102. https://doi.org/10.1063/5.0169810
20. Kirtas M, Oikonomou A, Passalis N, Mourgias-Alexandris G, Moralis-Pegios M, Pleros N, Tefas A (2022) Quantization-aware training for low precision photonic neural networks. Neural Netw 155:561–573
21. Bile A (2024) Solitonic neural networks. Springer Nature, Cham
22. Bile A, Tari H, Fazio E (2022) Episodic memory and information recognition using solitonic neural networks based on photorefractive plasticity. Appl Sci 12(11):5585
23. Li T, Li Y, Wang Y, Liu Y, Liu Y, Wang Z, Miao R, Han D, Hui Z, Li W (2023) Neuromorphic photonics based on phase change materials. Nanomaterials 13(11):1756
24. Robertson J, Kirkland P, Alanis JA, Hejda M, Bueno J, Di Caterina G, Hurtado A (2022) Ultrafast neuromorphic photonic image processing with a vcsel neuron. Sci Rep 12(1):4874
25. Owen-Newns D, Robertson J, Hejda M, Hurtado A (2022) Ghz rate neuromorphic photonic spiking neural network with a single vertical-cavity surface-emitting laser (vcsel). IEEE J Sel Top Quantum Electron 29(2: Optical Computing):1–10
26. Esser SK, McKinstry JL, Bablani D, Appuswamy R, Modha DS (2019) Learned step size quantization. arXiv preprint arXiv:1902.08153
27. Mourgias-Alexandris G, Moralis-Pegios M, Tsakyridis A, Simos S, Dabos G, Totovic A, Passalis N, Kirtas M, Rutirawut T, Gardes F et al (2022) Noise-resilient and high-speed deep learning with coherent silicon photonics. Nat Commun 13(1):5572
28. Paolini E, De Marinis L, Valcarenghi L, Maggiani L, Andriolli N (2024) Activation stretching for tackling noise in photonic aware neural networks. In: 2024 Optical Fiber Communications Conference and Exhibition (OFC), pp. 2–13. https://doi.org/10.1364/OFC.2024.Th2A.13
29. Passalis N, Kirtas M, Mourgias-Alexandris G, Dabos G, Pleros N, Tefas A (2021) Training noise-resilient recurrent photonic networks for financial time series analysis. In: 2020 28th European Signal Processing Conference (EUSIPCO), pp. 1556–1560. https://doi.org/10.23919/Eusipco47968.2020.9287649
30. Zhou S, Wu Y, Ni Z, Zhou X, Wen H, Zou Y (2016) Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160
31. Choi J, Wang Z, Venkataramani S, Chuang PI-J, Srinivasan V, Gopalakrishnan K (2018) Pact: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085
32. Tian Y, Zhao Y, Liu S, Li Q, Wang W, Feng J, Guo J (2022) Scalable and compact photonic neural chip with low learning-capability-loss. Nanophotonics 11(2):329–344
33. Yin P, Lyu J, Zhang S, Osher S, Qi Y, Xin J (2019) Understanding straight-through estimator in training activation quantized neural nets. arXiv preprint arXiv:1903.05662
34. LeCun Y, Bottou L, Bengio Y, Haffner P (2002) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
35. Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY et al (2011) Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, vol. 2011, p. 7. Granada
36. Krizhevsky A, Nair V, Hinton G CIFAR-10 (Canadian Institute for Advanced Research)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Emilio Paolini[1] · Lorenzo De Marinis[1] · Peter Seigo Kincaid[1] · Luca Valcarenghi[1] · Giampiero Contestabile[1] · Ioannis Roumpos[2] · Miltiadis Moralis-Pegios[3] · Nikos Pleros[3] · Nicola Andriolli[4]

✉ Emilio Paolini
emilio.paolini@santannapisa.it

[1] Telecommunications, Computer Engineering, and Photonics Institute, Scuola Superiore Sant'Anna, Pisa, Italy

[2] Wireless, Photonic Systems, Networks Group, Department of Physics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

[3] Wireless, Photonic Systems, Networks Group, Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

[4] Department of Information Engineering (DII), University of Pisa, Pisa, Italy