QuantumNAT: Quantum Noise-Aware Training with Noise Injection, Quantization and Normalization

[1]Hanrui Wang, [2]Jiaqi Gu, [3]Yongshan Ding, [4]Zirui Li, [5]Frederic T. Chong, [2]David Z. Pan, [1]Song Han

[1]Massachusetts Institute of Technology, [2]University of Taxes at Austin, [3]Yale University, [4]Shanghai Jiao Tong University, [5]University of Chicago

https://qmlsys.mit.edu

ABSTRACT

Parameterized Quantum Circuits (PQC) are promising towards quantum advantage on near-term quantum hardware. However, due to the large quantum noises (errors), the performance of PQC models has a severe degradation on real quantum devices. Take Quantum Neural Network (QNN) as an example, the accuracy gap between noise-free simulation and noisy results on IBMQ-Yorktown for MNIST-4 classification is over 60%. Existing noise mitigation methods are *general* ones without leveraging unique characteristics of PQC; on the other hand, existing PQC work *does not* consider noise effect. To this end, we present QuantumNAT, a PQC-specific framework to perform noise-aware optimizations in both training and inference stages to improve robustness. We experimentally observe that the effect of quantum noise to PQC measurement outcome is a linear map from noise-free outcome with a scaling and a shift factor. Motivated by that, we propose *post-measurement normalization* to mitigate the feature distribution differences between noise-free and noisy scenarios. Furthermore, to improve the robustness against noise, we propose *noise injection* to the training process by inserting quantum error gates to PQC according to realistic noise models of quantum hardware. Finally, *post-measurement quantization* is introduced to quantize the measurement outcomes to discrete values, achieving the denoising effect. Extensive experiments on 8 classification tasks using 6 quantum devices demonstrate that QuantumNAT improves accuracy by up to 43%, and achieves over 94% 2-class, 80% 4-class, and 34% 10-class classification accuracy measured on real quantum computers. The code for construction and noise-aware training of PQC is available in the TorchQuantum library.

1 INTRODUCTION

Quantum Computing (QC) is a new computational paradigm that can be exponentially faster than classical counterparts in various domains. Parameterized Quantum Circuits (PQC) are circuits containing trainable weights and are promising to achieve quantum advantages in current devices. Among various PQCs, Quantum Neural Network (QNN) is a popular algorithm in which a network of parameterized quantum gates are constructed and trained to embed data and perform certain ML tasks on a quantum computer, similar to the training and inference of classical neural networks.

Currently we are in the Noisy Intermediate Scale Quantum (NISQ) stage, in which quantum operations suffer from a high error rate of 10^{-2} to 10^{-3}, much higher than CPUs/GPUs (10^{-6} FIT). The quantum errors unfortunately introduces detrimental influence on PQC

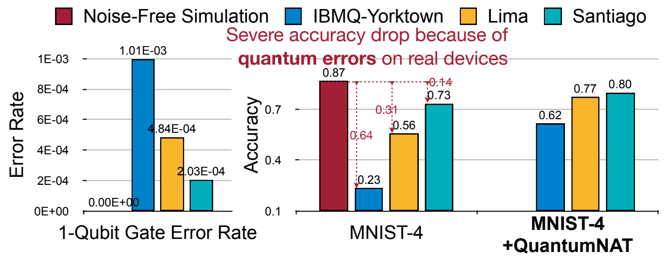

Figure 1: *Left:* Current quantum hardware has much larger error rates (around 10^{-3}) than classical CPUs/GPUs. *Right:* Due to the errors, PQC (QNN) models suffer from severe accuracy drops. Different devices have various error magnitudes, leading to distinct accuracy. These motivate QuantumNAT, a *hardware-specific noise-aware* PQC training approach to improve robustness and accuracy.

accuracy. Figure 1 shows the single-qubit gate error rates and the measured accuracy of classification tasks with QNN on different hardware. Three key observations are: (1) Quantum error rates (10^{-3}) are much larger than classical CMOS devices' error rates (10^{-6} failure per 10^9 device hours). (2) Accuracy on real hardware is significantly degraded (up to 64%) compared with noise-free simulation. (3) The same QNN on different hardware has distinct accuracy due to different gate error rates. IBMQ-Yorktown has a five times larger error rate than IBMQ-Santiago, and higher error causes lower accuracy.

Researchers have proposed noise mitigation techniques [16, 21] to reduce the noise impact. However, they are *general methods* without considering the unique characteristics of PQC, and can only be applied to PQC inference stage. On the other hand, existing PQC work [3, 8] does not consider the noise impact. This paper proposes a PQC-specific noise mitigation framework called QuantumNAT that optimizes PQC robustness in *both training and inference* stages, boosts the *intrinsic robustness* of PQC parameters, and improves accuracy on *real quantum machines*.

QuantumNAT comprises a three-stage pipeline. The first step, *post-measurement normalization* normalizes the measurement outcomes on each quantum bit (qubit) across data samples, thus removing the quantum error-induced distribution shift. Furthermore, we inject noise to the PQC training process by performing *error gate insertion*. The error gate types and probabilities are obtained from hardware-specific realistic quantum noise models provided by QC vendors. During training, we iteratively sample error gates, insert them to PQC, and updates weights. Finally, *post-measurement quantization* is further proposed to reduce the precision of measurement outcomes from each qubit and achieve a denoising effect.

In this paper, we are mainly using *QNNs as benchmarks* but the techniques can also be applied to *other PQCs*. The contributions of QuantumNAT are as follows:

- A systematic pipeline to mitigate noise impact: post-measurement normalization noise injection and post-measurement quantization.

DAC '22, July 10–14, 2022, San Francisco, CA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9142-9/22/07.
https://doi.org/10.1145/3489517.3530400

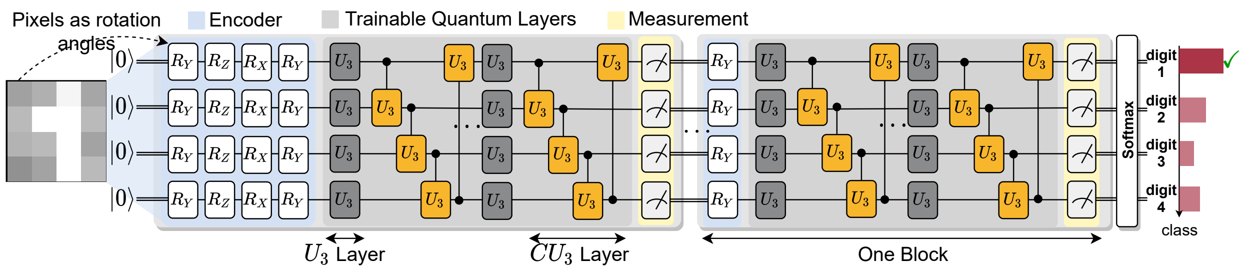

Figure 2: Quantum Neural Networks Architecture. QNN has multiple blocks, each has an encoder to encode classical values to quantum domain, quantum layers with trainable weights, and a measurement layer that obtains classical values.

- Extensive experiments on 8 ML tasks with 5 different design spaces on 6 quantum devices show that QuantumNAT can improve accuracy by up to 42%, 43%, 23% for 2-class, 4-class and 10-class classification tasks and demonstrates over 94%, 80% and 34% accuracy for 2-, 4-, and 10-classifications on real quantum hardware.
- The code for construction and noise-aware training of PQC is available at the TorchQuantum library. It is an convenient infrastructure to query noise models from QC providers such as IBMQ, extract noise information, perform training on CPU/GPU and finally deploy on real QC.

2 BACKGROUND AND RELATED WORK

QML and QNN. Quantum machine learning explores performing ML tasks on quantum devices. The path to *quantum advantage* on QML is typically provided by the quantum circuit's ability to generate and estimate highly complex kernels, which would otherwise be intractable to compute with conventional computers. They have been shown to have potential speed-up over classical counterparts in various tasks, including metric learning, data analysis, and principal component analysis. Quantum Neural Networks is one type of QML models using variational quantum circuits with trainable parameters to accomplish feature encoding of input data and perform complex-valued linear transformations thereafter. Various theoretical formulations for QNN have been proposed such as quantum Boltzmann machine [1] and quantum classifier [3, 17, 18, 20], etc.

Quantum error mitigation. As the error forms the bottleneck of the quantum area. Researchers have developed various error mitigation techniques [21]. Extrapolation methods [16] perform multiple measurements of a quantum circuit under different error rates and then extrapolate the ideal measurement outcomes when there is no noise. [11] trains PQC using RL with noisy simulator. QuantumNAT is fundamentally different from existing methods: (i) Prior work focuses on low-level numerical correction in inference only; QuantumNAT embraces more optimization freedom in both *training and inference.* It improves the intrinsic robustness and statistical fidelity of *PQC parameters.* (ii) PQC has a good built-in error-tolerance which motivates QuantumNAT's post-measurement quantization to reduce the numerical precision of intermediate results while preserving accuracy. (iii) QuantumNAT has a small overhead (<2%), while others introduce high measurements, circuit complexity cost, etc. We show that existing extrapolation method is orthogonal to QuantumNAT in Section 4.

Quantization and noise injection of classical NN. To improve NN efficiency, extensive work has been explored to trim down redundant bit representation in NN weights and activations [4, 19]. Though low-precision quantization limits the model capacity, it can improve the generalization and robustness [12]. An intuitive explanation is that quantization corrects errors by value clamping, thus avoiding cascaded error accumulation. Moreover, by sparsifying the parameter space, quantization reduces the NN complexity as a regularization mechanism that mitigates potential overfitting issues. Similarly, injecting noises into neural network training is demonstrated to help obtain a smoothed loss landscape for better generalization [14].

3 NOISE-AWARE PQC TRAINING

We use QNN as the benchmark PQC in this work. Figure 2 shows the QNN architecture. The inputs are classical data such as image pixels, and the outputs are classification results. The QNN consists of multiple blocks. Each has three components: encoder encodes the classical values to quantum states with rotation gates such as RY; trainable quantum layers contain parameterized gates that can be trained to perform certain ML tasks; measurement part measures each qubit and obtains a classical value. The measurement outcomes of one block are passed to the next block. For the MNIST-4 example in Figure 2, the first encoder takes the pixels of the down-sampled 4× 4 image as rotation angles θ of 16 rotation gates. The measurement results of the last block are passed through a Softmax to output classification probabilities. QuantumNAT overview is in Figure 3.

3.1 Post-Measurement Normalization

Measurement outcome shift due to quantum noises. Through extensive experiments, we find that the quantum noises apply a linear transformation to the measurement outcome y of a QNN for the input x. This can be formulated as $f(y_x) = \gamma y_x + \beta_x$, where (1) $\gamma \in [-1, 1]$ is an input-independent constant scaling factor, (2) β_x is an input-dependent shift. By analyzing the noise distribution, we observe that the changes in measurement results can often be *compensated by proper post-measurement normalization* across input batches. The method is most powerful when applied on a *small batch* of input data $\mathbf{x} = \{x_1, \ldots, x_m\}$. For small noises, γ is close to 1, and β_i is close to 0 for all $i \in [m]$. Therefore, the distribution of noisy measurement results undergoes a constant scaling by $\gamma \leq 1$ and a small shift by each β_i. In the small-batch regime when $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_m\}$ has small variance, the distribution is shifted by its mean $\beta = \mathbb{E}[\boldsymbol{\beta}]$. Since the input-dependent shifts can be approximated as their average value, i.e., $f(y_i) \approx \gamma y_i + \beta$, our normalization method can effectively compensate for such noise.

Post-measurement normalization. Based on the analysis above, we propose *post-measurement normalization* to offset the distribution scaling and shift. For each qubit, we collect its measurement results on a batch of inputs, compute their mean and std., then make the distribution of each qubit across the batch *zero-centered* and of *unit variance.* This is performed during both training and inference. During training, for a batch of measurement results: $\boldsymbol{y} = \{y_1, \ldots, y_m\}$, the normalized results are $\hat{y}_i = (y_i - \mathbb{E}[\boldsymbol{y}])/\sqrt{\text{Var}(\boldsymbol{y})}$. For noisy inference,

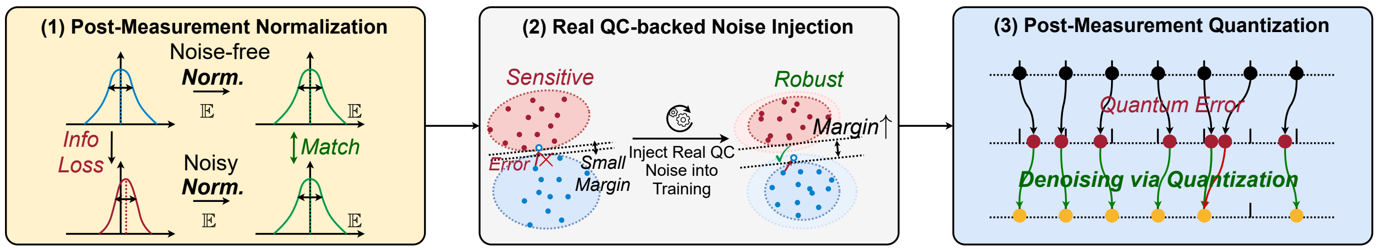
(1) Post-Measurement Normalization
Noise-free
Norm.
Info Loss
Noisy
Norm.
Match
(2) Real QC-backed Noise Injection
Sensitive
Robust
Error
Small Margin
Inject Real QC Noise into Training
Margin↑
(3) Post-Measurement Quantization
Quantum Error
Denoising via Quantization

Figure 3: QuantumNAT Overview. (1) Post-measurement normalization matches the distribution of measurement results between noise-free simulation and real QC. (2) Based on realistic noise models, noise-injection inserts *quantum error gates* to the training process to increase the classification margin between classes. (3) Measurement outcomes are further quantized for denoising.

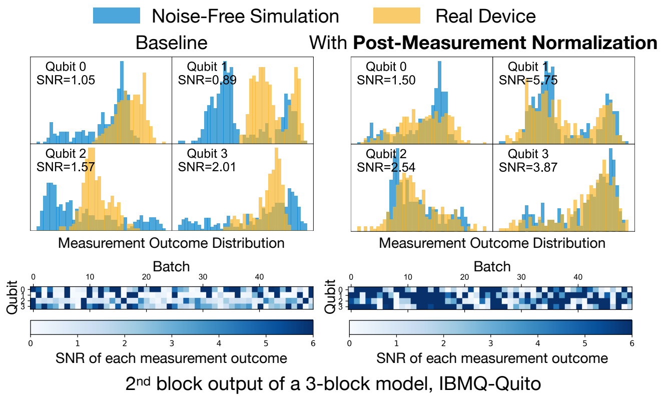

Figure 4: Post-measurement normalization reduces the distribution mismatch between noise-free simulation and noisy results, thus improving the Signal-to-Noise Ratio (SNR).

we correct the error as $\widehat{f(y_i)} = (f(y_i) - \mathbb{E}[f(\boldsymbol{y})])/\sqrt{\text{Var}(f(\boldsymbol{y}))} = ((\gamma y_i + \beta) - (\gamma\mathbb{E}[\boldsymbol{y}] + \beta))/\sqrt{\gamma^2\text{Var}(\boldsymbol{y})} = \hat{y}_i$.

Figure 4 compares the noise-free measurement result distribution of 4 qubits (blue) with their noisy counterparts (yellow) for MNIST-4. Qualitatively, we can clearly observe that the post-measurement normalization reduces the mismatch between two distributions. Quantitatively, we adopt signal-to-noise ratio, $SNR = \|\boldsymbol{A}\|_2^2/\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\|_2^2$, the inverse of relative matrix distance (RMD), as the metric. The SNR on each qubit and each individual measurement outcome is clearly improved. Though similar, it is different from Batch Normalization [7] as the testing batch uses its own statistics instead of that from training, and there is no trainable affine parameter.

3.2 Quantum Noise Injection

Although the normalization above mitigates error impacts, we can still observe small discrepancies on each individual measurement outcome, which degrade the accuracy. Therefore, to make the QNN model robust to those errors, we propose noise injection to the training process.

Quantum error gate insertion. As introduced in Section 2, different quantum errors can be approximated by Pauli errors via Pauli Twirling. The effect of Pauli errors is the random insertion of Pauli X, Y, and Z gates to the model with a probability distribution $\mathcal{E}$. How to compute $\mathcal{E}$ is out of the scope of this work. But fortunately, we can directly obtain it from the realistic device noise model provided by quantum hardware manufacturers such as IBMQ. The noise model specifies the probability $\mathcal{E}$ for different gates on each qubit. For single-qubit gates, the error gates are inserted *after* the original gate. For two-qubit gates, error gates are inserted after the gate on *one or both* qubits. For example, the SX gate on qubit 1 on IBMQ-Yorktown device has $\mathcal{E}$ as {X: 0.00096, Y: 0.00096, Z: 0.00096, None: 0.99712}. When 'None' is sampled, we will not insert any gate. The same gate on different qubits or different hardware will have up 10× probability difference. As in Figure 5, during training, for each QNN gate, we sample error gates based on $\mathcal{E}$ and insert it after the original gate. A new set of error gates is sampled for each training step. In reality, the QNN is compiled to the basis gate set of the quantum hardware (e.g., X, CNOT, RZ, CNOT, and ID) before performing gate insertion and training. We will also scale the probability distribution by a constant *noise factor* T and scale the X, Y, Z probability by T during sampling. T factor explores the trade-off between adequate noise injection and training stability. Typical T values are in the range of [0.5, 1.5]. The gate insertion overhead is typically less than 2%.

Readout noise injection. Obtaining classical values from qubits is referred as readout/measurement, which is also error-prone. The realistic noise model provides the statistical readout error in the form of a 2× 2 matrix for each qubit. For example, the qubit 0 of IBMQ-Santiago has readout error matrix [[0.984, 0.016], [0.022, 0.978]] which means the probability of measuring a $|0\rangle$ as 0 is 0.984 and as 1 is 0.016. We emulate the readout error effect during training by changing the measurement outcome. For instance, originally $P(0) = 0.3, P(1) = 0.7$, the noise injected version will be $P'(0) = 0.3 \times 0.984 + 0.7 \times 0.022 = 0.31, P'(1) = 0.7 \times 0.978 + 0.3 \times 0.016 = 0.69$.

Direct perturbation. Besides gate insertion, we also experimented with directly perturbing measurement outcomes or rotation angles as noise sources. For outcome perturbation, with benchmarking samples from the validation set, we obtain the error Err distribution between the noise-free and noisy measurement results and compute the mean μ_{Err} and std σ_{Err}. During training, we directly add noise with Gaussian distribution $\mathcal{N}(\mu_{Err}, \sigma_{Err}^2)$ to the normalized measurement outcomes. Similarly, for rotation angle perturbation, we add Gaussian noise to the angles of all rotation gates in QNN and make the effect of rotation angle Gaussian noise on measurement outcomes similar to real QC noise. We show in Section 4 that the gate insertion method is better than direct perturbations.

3.3 Post-Measurement Quantization

Finally, we propose post-measurement quantization on the normalized results to further denoise the measurement outcomes. We first clip the outcomes to $[p_{min}, p_{max}]$, where p are pre-defined thresholds, and then perform uniform quantization. The quantized values are later passed to the next block's encoder. Figure 6 shows one real

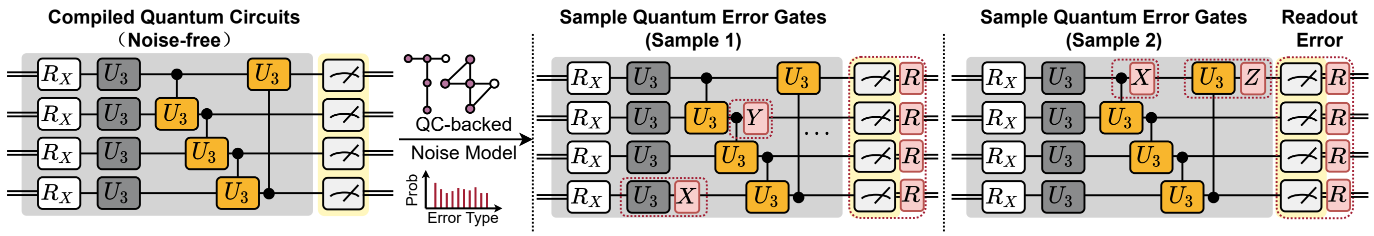

Figure 5: Noise injection via error gate insertion. X, Y, Z are sampled Pauli error gates. R is the injected readout error. Probabilities for gate insertion are obtained from real device noise models.

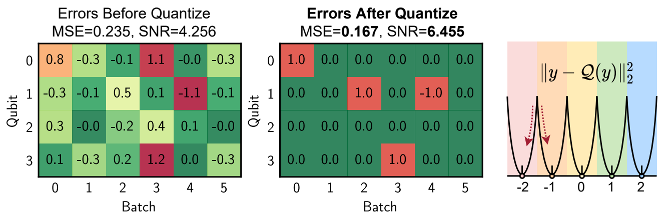

Figure 6: *Left*: Error maps before and after post-measurement quantization. Most errors can be corrected. *Right*: 5-level quantization buckets with a quadratic penalty loss.

example from Fashion-4 on IBMQ-Santiago with five quantization levels and $p_{min} = -2, p_{max} = 2$. The left/middle matrices show the error maps between noise-free and noisy outcomes before/after quantization. Most errors can be corrected back to zero with few exceptions of being quantized to a wrong centroid. The MSE is reduced from 0.235 to 0.167, and the SNR is increased from 4.256 to 6.455. We also add a loss term $||y - Q(y)||_2^2$ to the training loss, as shown on the right side, to encourage outcomes to be near to the quantization centroids to improve error tolerance and reduce the chance of being quantized to a wrong centroid. Besides improving robustness, quantization also reduces the control complexity of rotation gates.

4 EXPERIMENTS

4.1 Experiment Setups

Datasets. We conduct experiments on 8 classification tasks including MNIST [10] 10-class, 4-class (`0, 1, 2, 3`). and 2-class (`3, 6`); Vowel 4-class (`hid, hId, had, hOd`); Fashion [22] 10-class, 4-class (`t-shirt/top, trouser, pullover, dress`), and 2-class (`dress, shirt`), and CIFAR [9] 2-class (`frog, ship`). MNIST, Fashion, and CIFAR use 95% images in 'train' split as training set and 5% as the validation set. Due to the limited real QC resources, we use the first 300 images of 'test' split as test set. Vowel-4 dataset (990 samples) is separated to train:validation:test = 6:1:3 and test with the whole test set. MNIST and Fashion images are center-cropped to 24×24; and then down-sample to 4×4 for 2- and 4-class, and 6×6 for 10-class; CIFAR images are converted to grayscale, center-cropped to 28× 28, and down-sampled to 4× 4. All down-samplings are performed with average pooling. For vowel-4, we perform feature principal component analysis (PCA) and take 10 most significant dimensions.
QNN models. QNN models for 2 and 4-class use 4 qubits; 10-class uses 10. The first quantum block's encoder embeds images and vowel features. For 4×4 images, we use 4 qubits and 4 layers with 4 RY, 4 RX, 4 RZ, and 4 RY gates in each layer, respectively. There are in total 16 gates to encode the 16 classical values as the rotation angles. For 6×6 images, 10 qubits and 4 layers are used with 10 RY, 10 RX, 10 RZ, and 6 RY gates in each layer, respectively. 10 vowel features, uses 4 qubits and 3 layers with 4 RY, 4 RX, and 2 RZ gates on each layer for encoding. For trainable quantum layers, we use U3 and CU3 layers interleaved as in Figure 2 except for Table 2. For measurement, we measure the expectation values on Pauli-Z basis and obtain a value [-1, 1] from each qubit. The measurement outcome goes through post-measurement normalization and quantization and is used as rotation angles for RY gates in the next block's encoder. After the last block, for two-classifications, we sum the qubit 0 and 1, 2 and 3 measurement outcomes, respectively, and use Softmax to get probabilities. For 4 and 10-class, Softmax is directly applied to measurement outcomes.
Quantum hardware and compiler configurations.. We use IBMQ quantum computers via Qiskit [6] APIs. We study 6 devices, with #qubits from 5 to 15 and Quantum Volume from 8 to 32. We also employ Qiskit for compilation. The optimization level is set to 2 for all experiments. All experiments run 8192 shots. The noise models we used are off-the-shelf ones updated by IBMQ team.

4.2 Main Results

QNN results. We experiment with four different QNN architectures on 8 tasks running on 5 quantum devices to demonstrate QuantumNAT's effectiveness. For each benchmark, we experiment with noise factor $T = \{0.1, 0.5, 1, 1.5\}$ and quantization level among $\{3, 4, 5, 6\}$ and select one out of 16 combinations with the lowest loss on the *validation set* and test on the *test set.* Normalization and quantization are not applied to the last block's measurement outcomes as they are directly used for classification. As in Table 1, QuantumNAT consistently achieves the highest accuracy on 26 benchmarks. The third bars of Athens are unavailable due to its retirement. On average, normalization, noise injection and quantization improve accuracy by 10%, 9%, and 3%, respectively. A larger model does not necessarily have higher accuracy. For example, Athens' model is 7.5× larger than Yorktown with higher noise-free accuracy. However, due to more errors introduced by the larger model, the real accuracy is lower.
Performance on different design spaces. In Table 2, we evaluate QuantumNAT on different QNN design spaces. Specifically, the trainable quantum layers in one block of 'ZZ+RY' [13] space contains one layer of ZZ gate, with ring connections, and one RY layer. 'RXYZ' [15] space has five layers: $\sqrt{\text{H}}$, RX, RY, RZ, and CZ. 'ZX+XX' [3] space has two layers: ZX and XX. 'RXYZ+U1+CU3' [5] space, according to their random circuit basis gate set, has 11 layers in the order of RX, S, CNOT, RY, T, SWAP, RZ, H, $\sqrt{\text{SWAP}}$, U1 and CU3. We conduct experiments on MNIST-4 and Fashion-2 on 2 devices. In 13 settings out of 16, QuantumNAT achieves better accuracy. Thus, QuantumNAT is a general technique agnostic to QNN model size and design space.
Scalability. When classical simulation is infeasible, we can move the the noise-injected training to real QC using techniques such as *parameter shift* [2]. In this case, the training cost is *linearly* scaled with qubit number. Post-measurement normalization and quantization are also *linearly* scalable because they are performed on the

Table 1: QuantumNAT consistently achieves the highest accuracy, with on average 22% better. 'B' for Block, 'L' for Layer.

Model	Method	MNIST-4	Fash.-4	Vow.-4	MNIST-2	Fash.-2	Cifar-2
2B×12L Santiago	Baseline	0.30	0.32	0.28	0.84	0.78	0.51
	+ Post Norm.	0.41	0.61	0.29	0.87	0.68	0.56
	+ Gate Insert.	0.61	0.70	0.44	0.93	0.86	0.57
	+ Post Quant.	**0.68**	**0.75**	**0.48**	**0.94**	**0.88**	**0.59**
2B×2L Yorktown	Baseline	0.43	0.56	0.25	0.68	0.70	0.52
	+ Post Norm.	0.57	0.60	0.38	0.86	0.72	0.56
	+ Gate Insert.	0.58	0.60	**0.45**	0.91	0.85	0.57
	+ Post Quant.	**0.62**	**0.65**	0.44	**0.93**	**0.86**	**0.60**
2B×6L Belem	Baseline	0.28	0.26	0.20	0.46	0.52	0.50
	+ Post Norm.	0.52	0.57	0.33	0.81	0.62	0.51
	+ Gate Insert.	0.52	0.60	0.37	0.84	**0.82**	0.57
	+ Post Quant.	**0.58**	**0.62**	**0.41**	**0.88**	0.80	**0.61**
3B×10L Athens	Baseline	0.29	0.36	0.21	0.54	0.46	0.49
	+ Post Norm.	0.44	0.46	0.37	0.51	0.51	0.50
	+ Gate Insert.	-	-	-	-	-	-
	+ Post Quant.	**0.56**	**0.64**	**0.41**	**0.87**	**0.64**	**0.53**

Model	Method	MNIST-10	Fash.-10	Avg.-All
2B×2L Melbo.	Baseline	0.11	0.09	0.42
	+ Post Norm.	0.08	0.12	0.52
	+ Gate Insert.	0.25	0.24	0.61
	+ Post Quant.	**0.34**	**0.31**	**0.64**

Table 2: Accuracy on different design spaces.

Design Space	MNIST-4		Fashion-2	
	Yorktown	Santiago	Yorktown	Santiago
'ZZ+RY'	**0.43**	0.57	0.80	**0.91**
+QuantumNAT	0.34	**0.60**	**0.83**	0.86
'RXYZ'	0.57	0.61	0.88	0.89
+QuantumNAT	**0.61**	**0.70**	**0.92**	**0.91**
'ZX+XX'	0.29	0.51	**0.52**	0.61
+QuantumNAT	**0.38**	**0.64**	**0.52**	**0.89**
'RXYZ+U1+CU3'	0.28	**0.25**	0.48	0.50
+QuantumNAT	**0.33**	0.21	**0.53**	**0.52**

Table 3: Scalable noise-aware training.

Machine	Bogota	Santiago	Lima
Noise-unaware	0.74	0.97	0.87
QuantumNAT	**0.79**	**0.99**	**0.90**

Table 4: Compatible with existing noise mitigation.

Method	MNIST-4	Fashion-4
Normalization only	0.78	0.81
Normalization + Extrapolation	**0.81**	**0.83**

measurement outcomes. Gradients obtained with real QC are naturally noise-aware because they are directly influenced by quantum noise. To demonstrate the practicality, we train a 2-class task with two numbers as input features [8] (Table 3). The QNN has 2 blocks; each with 2 RY and a CNOT gates. The noise-unaware baseline trains the model on classical part and test on real QC. In QuantumNAT, we train the model with parameter shift and test, both on real QC. We consistently outperform noise-unaware baselines.

Compatibility with existing noise mitigation. QuantumNAT is orthogonal to existing noise mitigation such as extrapolation method. It can be combined with post-measurement normalization (Table 4). The QNN model has 2 blocks, each with three U3+CU3 layers. For "Normalization only", the measurement outcomes of the 3-layer block are normalized across the batch dimension. For "Extrapolation + Normalization", we use extrapolation to estimate the standard deviation of noise-free measurement outcomes. We firstly train the QNN model to convergence and then repeat the 3 layers to 6, 9, 12 layers and obtain four standard deviations of measurement outcomes. Then we linearly extrapolate them to obtain noise-free std. We normalize the measurement outcomes of the 3-layer block to make their std the same as noise-free and then apply the proposed post-measurement norm. Results show that the extrapolation can further improve the QNN accuracy thus being orthogonal.

Table 5: Post-measurement norm. improves acc. & SNR.

Quantum Devices ↓	QNN Models →	2 Blocks				4 Blocks			
		×2 Layers		×8 Layers		×2 Layers		×4 Layers	
		Acc.	SNR	Acc.	SNR	Acc.	SNR	Acc.	SNR
Santiago	Baseline	0.61	6.15	0.52	1.79	0.57	6.96	0.62	4.20
	+Norm	**0.66**	**15.69**	**0.79**	**4.85**	**0.70**	**11.36**	**0.68**	**6.55**
Quito	Baseline	0.58	6.64	0.35	1.43	0.60	3.98	0.29	1.73
	+Norm	**0.66**	**13.92**	**0.71**	**2.98**	**0.74**	**12.26**	**0.72**	**4.54**
Athens	Baseline	0.59	8.91	0.60	2.14	0.63	9.52	0.55	3.54
	+Norm	**0.64**	**20.27**	**0.78**	**3.47**	**0.74**	**14.07**	**0.69**	**6.09**

4.3 Ablation Studies

Ablation on post-measurement normalization. Table 5 compares the accuracy and signal-to-noise ratio (SNR) before and after post-measurement normalization on MNIST-4. We study 4 different QNN architectures and evaluate on 3 devices. The normalization can significantly and consistently increase SNR.

Ablation on different noise injection methods. Figure 7 compares different noise injection methods. Gaussian noise statistics for perturbations are obtained from error benchmarking. The left side shows accuracy without quantization. With different noise factors T, the gate insertion and measurement outcome perturbation have similar accuracy, both better than rotation angle perturbation. A possible explanation is that the rotation angle perturbation does not consider non-rotation gates such as X and SX. The right side further investigates the first two methods' performance with quantization. We set noise factor $T = 0.5$ and alter quantization levels. Gate insertion outperforms perturbation by 11% on average on 3 different devices and QNN models. The reason is: directly added perturbation on measurement outcomes can be easily canceled by quantization, and thus it is harder for noise injection to take effect.

Noise factor and post-measurement quantization level analysis. We visualize the QNN accuracy contours on Fashion-4 on IBMQ-Athens with different noise factors and quantization levels in Figure 8 left. The best accuracy occurs for factor 0.2 and 5 levels. Horizontal-wise, the accuracy first goes up and then goes down. This is because too few quantization levels hurt the QNN model capacity; too many levels cannot bring sufficient denoising effect. Vertical-wise, the accuracy also goes up and then down. Reason: when the noise is too small, the noise-injection effect is weak, thus cannot improve the model robustness; while too large noise makes the training process unstable and hurts accuracy.

Visualization of QNN extracted features. MNIST-2 classification result is determined by which feature is larger between the two: feature one is the sum of measurement outcomes of qubit 0 and 1; feature 2 is that of qubit 2 and 3. We visualize the two features obtained from experiments on Belem in a 2-D plane as in Figure 8 right. The blue dash line is the classification boundary. The circles/stars are samples of digit '3' and '6'. All the baseline points (yellow) huddled

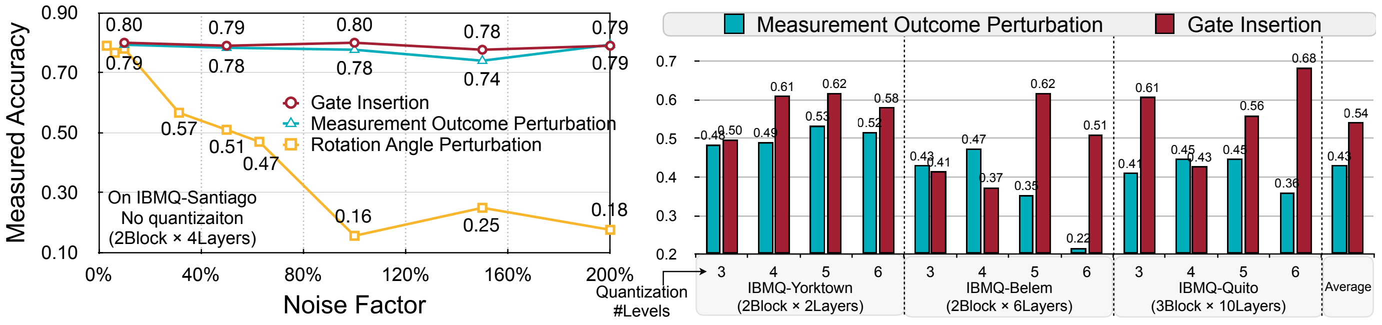
Figure 7: Ablation on different noise injection methods. *Left:* Without quantization, gate insertion and measurement perturbation performs similar, both better than rotation angle perturbation. *Right:* With quantization, gate insertion is better as perturbation effect can be canceled by quantization.

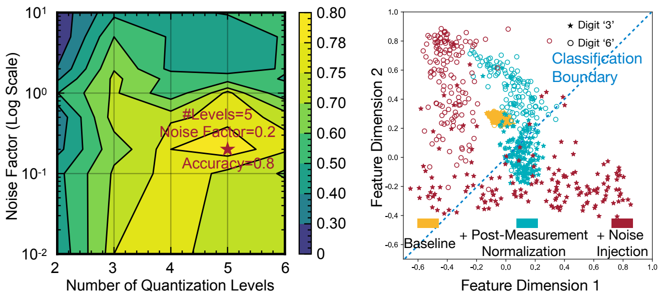
Figure 8: Left: Accuracy contours of quantization levels and noise factors. Right: Feature visualization.

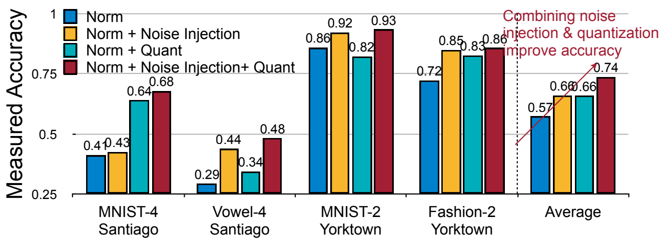
Figure 9: Ablation of applying noise injection and quantization individually or jointly.

together, and all digit '3' samples are misclassified. With normalization (green), the distribution is significantly expanded, and the majority of '3' is correctly classified. Finally, after noise injection (red), the margin between the two classes is further enlarged, and the samples are farther away from the classification boundary, thus becoming more robust.

Breakdown of accuracy gain. Figure 9 shows the performance of only applying noise-injection, only applying quantization, and both. Using two techniques individually can both improve accuracy by 9%. Combining two techniques delivers better performance with a 17% accuracy gain. This indicates the benefits of synergistically applying three techniques in QuantumNAT.

5 CONCLUSION

PQC is a promising candidate to demonstrate practical quantum advantages over classical approaches. The road to such advantage relies on: (1) the discovery of novel feature embedding that encodes classical data non-linearly, and (2) overcome the impact of quantum noise. This work focuses on the latter and show that a noise-aware training pipeline with post-measurement normalization, noise injection, and post-measurement quantization can elevate the PQC robustness against arbitrary, realistic quantum noises. We anticipate such robust PQC being useful in exploring practical QC applications.

ACKNOWLEDGMENT

We acknowledge NSF CAREER Award #1943349, MIT-IBM Watson AI Lab, Baidu Fellowship, Qualcomm Innovation Fellowship, and IBM Quantum.

REFERENCES

[1] Mohammad H Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. 2018. Quantum boltzmann machine. *Physical Review X* 8, 2 (2018).

[2] Gavin E Crooks. 2019. Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition. *arXiv preprint 1905.13311* (2019).

[3] Edward Farhi and Hartmut Neven. 2018. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002* (2018).

[4] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).

[5] Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook. 2020. Quanvolutional neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence* 2, 1 (2020), 1–9.

[6] IBM. [n. d.]. IBM Quantum. ([n. d.]). https://quantum-computing.ibm.com/

[7] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*. PMLR, 448–456.

[8] Weiwen Jiang, Jinjun Xiong, and Yiyu Shi. 2021. A co-design framework of neural networks and quantum circuits towards quantum advantage. *Nature communications* 12, 1 (2021), 1–13.

[9] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. [n. d.]. CIFAR-10 (Canadian Institute for Advanced Research). ([n. d.]).

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[11] Zhiding Liang, Zhepeng Wang, Junhuan Yang, Lei Yang, Yiyu Shi, and Weiwen Jiang. 2021. Can Noise on Qubits Be Learned in Quantum Neural Network? A Case Study on QuantumFlow. In *ICCAD*. IEEE, 1–7.

[12] Ji Lin, Chuang Gan, and Song Han. 2019. Defensive quantization: When efficiency meets robustness. *arXiv preprint arXiv:1904.08444* (2019).

[13] Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. 2020. Quantum embeddings for machine learning. *arXiv preprint 2001.03622* (2020).

[14] K. Matsuoka. 1992. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man, and Cybernetics* 22, 3 (1992), 436–440.

[15] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. 2018. Barren plateaus in quantum neural network training landscapes. *Nature communications* 9, 1 (2018), 1–6.

[16] Kristan Temme, Sergey Bravyi, and Jay M Gambetta. 2017. Error mitigation for short-depth quantum circuits. *Physical review letters* 119, 18 (2017), 180509.

[17] Hanrui Wang, Yongshan Ding, Jiaqi Gu, Yujun Lin, David Z Pan, Frederic T Chong, and Song Han. 2022. QuantumNAS: Noise-adaptive search for robust quantum circuits. *HPCA* (2022).

[18] Hanrui Wang, Zirui Li, Jiaqi Gu, et al. 2022. QOC: Quantum On-Chip Training with Parameter Shift and Gradient Pruning. *DAC* (2022).

[19] Hanrui Wang, Zhekai Zhang, and Song Han. 2021. Spatten: Efficient sparse attention architecture with cascade token and head pruning. In *HPCA*. IEEE, 97–110.

[20] Zhepeng Wang, Zhiding Liang, Shanglin Zhou, et al. 2021. Exploration of Quantum Neural Architecture by Mixing Quantum Neuron Designs. In *ICCAD*. IEEE.

[21] Robert Wille, Lukas Burgholzer, and Alwin Zulehner. 2019. Mapping quantum circuits to IBM QX architectures using the minimal number of SWAP and H operations. In *DAC*. IEEE, 1–6.

[22] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv 1708.07747* (2017).