

QuST: Optimizing Quantum Neural Network Against Spatial and Temporal Noise Biases

Tingting Li¹, Liqiang Lu¹, Ziming Zhao¹, *Member, IEEE*, Ziqi Tan¹, Siwei Tan¹,
and Jianwei Yin¹, *Member, IEEE*

Abstract—Quantum neural networks (QNNs) hold immense potential for complex tasks by harnessing quantum entanglement and superposition, such as physics simulation, artificial intelligence, and cryptography. However, the presence of quantum noise, stemming from hardware imperfections and environmental interactions, significantly reduces their practical performance. Moreover, the noise varies from different devices and shifts over time, necessitating continuous retraining models to chase and cater to the evolving noise, leading to high-computation costs. In this article, we present QuST, a novel QNN robust training framework designed to handle the noise in a once-and-for-all manner, which can tackle both spatial and temporal biases to maintain the QNN model accuracy under ever-changing noise conditions. Our approach consists of three key components. First, we propose a metric called circuit sequence correctness (CSC) to characterize QNN circuit reliability in noisy environments. Then, we model CSC as a training weight to incorporate loss integration and utilize KL divergence to align noise inference with noise-free inference, thereby improving anti-noise capabilities. Furthermore, we introduce multiscale noise-aware training to enhance the model's noise tolerance at various noise magnitudes. We conduct experiments on MNIST and fashion-MNIST datasets, along with 190-day historical noise simulations and one case study on 7 real IBMQ quantum computers. The results demonstrate 8.1%–15.1% and 9.1%–11.45% accuracy improvements in temporal and spatial dimensions, respectively. Additionally, we conduct ablation experiments to validate the effectiveness of the QuST's key components. The results demonstrate that QuST consistently sustains high accuracy without retraining, even under changing noise conditions, and exhibits minimal loss of accuracy as noise levels increase.

Index Terms—Noise-resilient model, quantum computing, quantum neural networks (QNNs).

I. INTRODUCTION

WITH the development of quantum computing, quantum neural network (QNN) serves as the intersection of quantum computing and artificial intelligence, which has attracted much attention in recent years. QNN mainly consists

Received 27 February 2024; revised 20 July 2024; accepted 12 September 2024. Date of publication 1 October 2024; date of current version 21 March 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFF0905200; in part by the National Natural Science Foundation of China under Grant 62472374 and Grant 61825205; and in part by Zhejiang Pioneer (Jianbing) under Project 2023C01036. This article was recommended by Associate Editor R. Wille. (Corresponding authors: Liqiang Lu; Jianwei Yin.)

The authors are with the College of Computer Science, Zhejiang University, Hangzhou 310027, China (e-mail: liti2020@zju.edu.cn; liqianglu@zju.edu.cn; zhaoziming@zju.edu.cn; tanziqi@zju.edu.cn; siweitian@zju.edu.cn; juyjw@zju.edu.cn).

Digital Object Identifier 10.1109/TCAD.2024.3471949

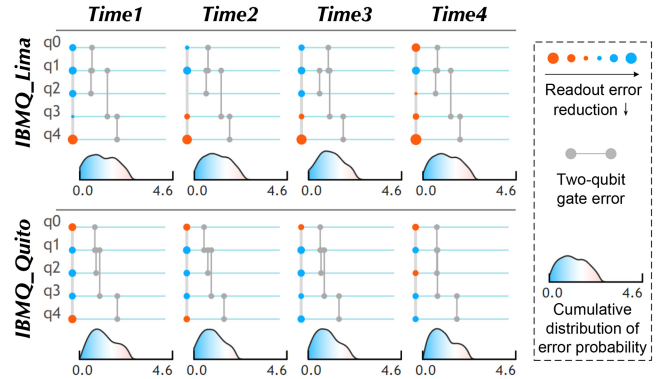


Fig. 1. Noise visualization [17] of readout and gate errors.

of parameterized quantum gates to constitute circuits across various fields, such as quantum chemistry [1], quantum natural language processing [2], and quantum adversarial learning [3]. Quantum computing is considered promising to accelerate classical tasks through quantum superposition and entanglement to perform calculations [3], [4], [5]. Similar to the classical neural networks, the constructed QNN model needs to be trained with embedded data to handle specific machine learning tasks on a quantum computer (QC). However, QCs are currently in the noisy intermediate-scale quantum (NISQ) era [6], wherein quantum operations and qubit states are susceptible to quantum noise, leading to imprecise calculations.

Quantum noise poses a significant challenge to the practicability of quantum computing. Therefore, researchers have proposed several methods to alleviate the quantum noise, including quantum error correction [7], quantum error mitigation [8], [9], [10], [11], quantum compiler optimizations [12], [13], etc. For the domain of QNN error mitigation, EQC [14] employs distributed training, QuCAD [15] and PCOAST [16] compresses circuit gates based on noise configurations, and QuantumNAT [4] injects noise and post-measurement normalization during QNN training to improve performance in specific noise environments. However, when confronted with noise variations, conventional approaches often involve retraining to adjust to the new noise rather than fundamentally aiming to train a robust model capable of resisting evolving noise environments, which is time-consuming and labor-intensive. For training a robust model that can be adapted to changing noise in a once-and-for-all manner, we summarize the following three primary challenges.

1) *Quantum Noise Exhibits Variations Across Different Times and Devices*: Quantum noise is not set in stone. For one thing, the noise presentation of various quantum devices is different depending on quantum fabrication technology, intrinsic qubit changes, environmental conditions, and so on. For another, even for the same quantum device, the quantum noise exhibits diversity over time because of the interaction with the environment. We term them *spatial bias* and *temporal bias*, respectively. In Fig. 1, we visualize the quantum noise of two IBMQ computers across four days based on VACSEN [17], the probability of readout and gate errors vary with different times and devices. Although practitioners regularly adjust parameters to calibrate machines, noise cannot be eliminated and often shows a random distribution in a certain interval. Specifically, in Fig. 2(a), we randomly selected 10 days to test on IBMQ Quito to execute MNIST-4, noise fluctuations over time lead to variation in fidelity. This results in a maximum reduction of 30% in QNN accuracy, we refer to this time-related variation as *temporal bias*. In Fig. 2(b), we deploy the same QNN model on four devices to test Fashion-4, “D1–D4” are denoted as IBMQ Quito, Lima, Belem, and Manila, respectively. The results reveal fidelity changes due to varying noise, and the model’s accuracy is reduced by up to 14%, we characterize this noise variation across devices as *spatial bias*.

2) *Existing Metrics Cannot Accurately Characterize the QNN Model Performance Under Noisy Environments*: For example, we provide a specific case in Fig. 2, conventional noise characterization metrics fidelity [18] does not exhibit a consistently positive correlation with the QNN model accuracy, such as Fig. 2(a) the fidelity is 0.95 and 0.981, while accuracy is 0.77 and 0.73 corresponding to $T7$ and $T3$. Through performance evaluation and Fidelity calculation with sampling 10 days, we observe that model accuracy and fidelity do not show a good positive correlation. Especially in subgraph b), the relationship between accuracy and fidelity exhibits a low correlation (refer to the changing trend of the two curves). Therefore, we need a metric that accurately profiles the QNN accuracy in various noisy environments.

3) *Traditional Training Methods Cannot Effectively Deal With the Diversity of Noise*: To resist noise, a strawman design would be to continuously retrain a set of QNN model parameters to adapt to new noise instances. However, this approach is neither efficient nor practical, as noise is unpredictable and constantly changing, as mentioned in the challenge 1). Therefore, instead of continually adapting to the evolving noise environment, we aim to develop noise-resilient QNNs that exhibit robustness against noise in a once-and-for-all manner.

To cope with these issues, we introduce QuST, a novel framework for optimizing QNNs against spatial and temporal noise biases. Specifically, we propose a novel metric to depict the circuit sequence correctness (CSC) under noisy

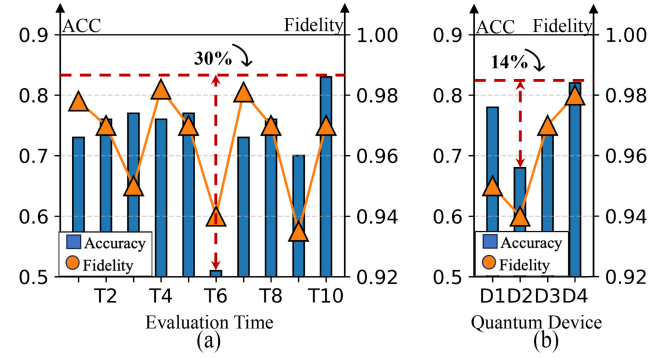


Fig. 2. Evaluate fidelity and QNN accuracy across temporal and spatial biases: (a) Assess the model at different times on IBMQ Quito and (b) evaluate the model across different devices.

environments, which aims to quantize the effect of circuit errors on QNN performance, echo back challenge 2). Then, we design a suite of training pipelines for QuST, including noise-free/aware circuit execution, utilizing CSC as weight coefficient, and multiscale noise-aware training to cater to challenge 3). Through such a training process, we could realize the fault-tolerant QNN model against noise, even if there are spatial and temporal biases, to cope with the challenge 1). Overall, this article makes three key contributions.

- 1) We carefully examine and analyze the effect that quantum noise brings for QNN models in practice, identifying two key problems: a) “spatial bias” for noise variations across devices and b) “temporal bias” for noise variations over time.
- 2) We propose QuST, a novel framework consisting of a suite of training pipelines to cope with ever-changing noises. We also propose CSC, a new noise evaluation metric that addresses the shortcomings of existing metrics in describing QNN performance.
- 3) We conduct experiments on three circuits and two datasets, using 190-day historical noise from seven IBMQ QCs. Results show that QuST significantly outperforms existing methods against ever-changing noises without retraining, maintaining its effectiveness across temporal and spatial noise variations. We also perform ablation experiments to verify the effectiveness, stability, and scalability of QuST.

II. BACKGROUND AND RELATED WORK

A. Quantum Neural Networks

Classical neural networks are algorithmic models inspired by the human brain, capable of being trained to recognize patterns in data and solve complex problems [19]. Representative structures consist of a series of interconnected nodes (neurons) arranged in a layered structure. These networks learn by adjusting their parameters through machine learning or deep learning training strategies [20]. In classical neural networks, the input and output are represented as neurons and the connections between these neurons are defined by weights that require training. In contrast, QNNs combine quantum computing with classical neural networks based on the principles of

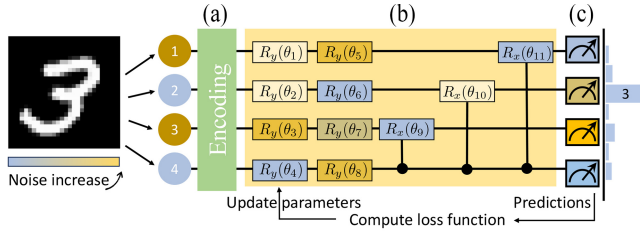


Fig. 3. QNN model architecture for MNIST classification task. (a) Encoding. (b) Trainable quantum circuit. (c) Measurement.

quantum mechanics [21]. The concept of QNNs first appeared in publications by Subhash in 1995 [22], motivated by the desire to connect investigations in the field of neuroscience with the characteristics of quantum computation. QNNs typically refer to variational or parameterized quantum circuits, which are mostly composed of a series of parameterized gates stacked together. The specific structure is shown in Fig. 3. QNNs utilize qubit states for input and output, with weights corresponding to quantum gate parameters within the quantum circuit. Each gate parameter represents a qubit operation based on principles of quantum mechanics.

In Fig. 3, the QNN model comprises a) quantum encoding, b) trainable quantum circuit, and c) measurement, forming a cascade of quantum operations. Encoding maps data (e.g., classical data) to quantum states, the trainable quantum circuit applies gate operations¹ on qubits, and the circuit's output is obtained through measurement. Model inference results are derived using the *Softmax* function. During training, gate parameters of the trainable quantum circuit are updated iteratively through loss function computation and backpropagation. QNN models are used for various tasks, such as QuantumNAT [4] and QOC [23] for image classification, QSAN [2] for natural language processing, and QGAN [3] for generative adversarial networks.

B. Quantum Noises

Current NISQ quantum devices are susceptible to various noise sources [24], such as thermal fluctuations, electromagnetic interference, imperfections in quantum gates, and environmental interactions [25], [26], [27]. These factors cause fluctuations in qubit phase and amplitude, ultimately hindering qubit accuracy and reliability improvements. We can model these quantum noises² that affect the calculation accuracy of QCs into two categories.

- 1) *Readout errors*, often consist of state preparation and measurement (SPAM) errors [14] that would arise due to imperfect state preparation and quantum state

measurement processes. These errors can result in discrepancies between the intended states $|0\rangle$ and $|1\rangle$, causing incorrect distinctions between these fundamental states. According to Qiskit [21], readout errors are applied to all measurements. In simpler terms, errors during the preparation and measurement of a quantum state can cause confusion between the binary states $|0\rangle$ and $|1\rangle$.

- 2) *Gate errors*, stemming from hardware and environmental influences, leading to imperfect gate operations, can summarize into dephasing errors, thermal relaxation errors, depolarizing errors [28], and so on. Followed by Qiskit [21], we model single-qubit gate errors consisting of depolarizing error and thermal relaxation error, the two-qubit gate errors consisting of a 2-qubit depolarizing error, and a 1-qubit thermal relaxation error on each pair of qubits. These errors cause information loss, reduced coherency, and inaccuracies in quantum computations, collectively known as gate errors.

For the workflow of QNN, quantum encoding³ and circuit execution are often affected by gate errors, while measurements mainly suffer from readout errors. Furthermore, we provide mathematical definitions of readout errors and gate errors for noise modeling in Section III-A.

C. QNNs Noise Mitigation

Considering quantum noise greatly limits the development of quantum computing superiority, a series of error mitigation methods [8], [29] have been proposed. Some representative works involve zero-noise extrapolation (ZNE) [10], probabilistic error cancellation (PEC) [8], quantum error correcting codes (QECC) [7], [30], etc. These technologies essentially investigate error mitigation at a high level and are a general and universal process in the quantum computing field. However, recognizing the limitations of these approaches within specific domains, some recent works have turned their attention to developing noise mitigation schemes specifically tailored for QNNs. EQC [14] employs distributed training to allocate quantum tasks across multiple quantum devices to improve the training speed. QOC [23] proposes probabilistic gradient pruning to identify gradients with potentially large errors and then remove them. QuCAD [15] uses a compression-aided framework to configure compression circuit gates based on the given noise to improve QNN accuracy. QuantumNAT [4] uses noise injection during the QNN training process, post-measurement normalization, and quantization to optimize measurement, which enhances performance in a specific noisy environment and realizes state-the-of-art (SOTA) robustness against quantum noises. Nonetheless, these methods all encounter a common challenge in adapting to noise changes over time and devices. They tend to cope with these biases through retraining, which is resource-intensive and time-consuming. To achieve more effective noise mitigation, we propose QuST, a QNN training framework that enables noise-resilient capability against spatial and temporal biases in an inline manner.

³Gate-based encoding is primarily affected by gate errors, while amplitude encoding may be more susceptible to quantum state preparation errors.

¹Gate operations like the quantum gate $R_y(\theta_1)$ in Fig. 3(b), meaning they rotate the θ_1 angle across the y-axis. These gate parameters can be trained.

²Our noise modeling aligns with Qiskit [21], incorporating device parameters, calibration data, gate times, and so on. Specifically, device parameters include qubits frequency and temperature, and device calibration data includes the gate infidelity, readout error, and T_1 , T_2 relaxation time. These various noise sources are abstracted into local error channels: readout error, depolarizing error, and thermal relaxation error. These error channels are then integrated into the readout error model and gate error model. The above noise modeling process considers each qubit's coherence times (T_1 and T_2), operating frequency, and readout error rate. These noise models are simplified approximations of the actual dynamics of a QC.

III. QUANTUM NOISE FORMULATION

In this section, we formalize the problems of noise modeling and noise biases. Specifically, we formulate noise biases into spatial and temporal biases.

A. Noise Modeling

In Section II-B, following the methodology of Qiskit [21], we model various noise sources uniformly as the readout error model and the gate error model.⁴ These noise models can simulate the dynamic noise characteristics of quantum devices. The mathematical formulation of noise modeling is as follows.

- 1) *Readout Error Model*: Readout error refers to the errors in qubit SPAM that result in the inability to correctly prepare or distinguish between the states $|0\rangle$ and $|1\rangle$. It is also known as SPAM errors [14]. The readout error of each qubit is represented as a 2×2 probability matrix, such as for the i -qubit readout error matrix R_i

$$R_i = \begin{bmatrix} P_{i,0} & 1 - P_{i,0} \\ 1 - P_{i,1} & P_{i,1} \end{bmatrix} \quad (1)$$

which means the probability of correctly measuring $|0\rangle$ is $P_{i,0}$ and $|1\rangle$ is $P_{i,1}$. In contrast, the probability of wrongly measuring $|0\rangle$ to $|1\rangle$ is $(1 - P_{i,0})$ and $|1\rangle$ to $|0\rangle$ is $(1 - P_{i,1})$. For instance, the readout error of the first qubit can be represented by a 2×2 probability matrix, as follows:

$$R_1 = \begin{bmatrix} 0.9358 & 0.0642 \\ 0.0164 & 0.9836 \end{bmatrix}. \quad (2)$$

The probability of correctly measuring $|0\rangle$ is 0.9358 and $|1\rangle$ is 0.9836, and the probability of incorrect measurement is 0.0642 and 0.0164, respectively. If we have an ideal output refer to $P(0) = 0.4, P(1) = 0.6$, the noisy readout output

$$\begin{aligned} P'(0) &= 0.4 * 0.9358 + 0.6 * 0.0164 \approx 0.384 \\ P'(1) &= 0.4 * 0.0642 + 0.6 * 0.9836 \approx 0.616. \end{aligned} \quad (3)$$

- 2) *Gate Error Model*: We unify the modeling of various noise factors into gate noise, which mainly involves quantum operation errors stemming from hardware imperfections, environmental interactions, decoherence, and other factors, leading to inaccurate execution of quantum operations, such as insufficient or excessive qubit rotation. For example, a single-qubit gate error could involve a bit-flip error (Pauli- X), phase-flip error (Pauli- Z), bit-phase-flip error (Pauli- Y), and reset error [27]. A two-qubit gate error consists of many sets of identity crosses with Pauli operators

$$\begin{aligned} 1 - q \text{ gate error: } & \{X, Y, Z, \text{reset}\} \\ 2 - q \text{ gate error: } & \{IX, IY, IZ, XI, XX, XY, XZ, YI, YX, \\ & YY, YZ, ZI, ZX, ZY, ZZ, \text{reset}\}. \end{aligned} \quad (4)$$

⁴These noise models considered each qubit's coherence times (T_1 and T_2), operating frequency, and readout error rate. First, various noise sources are abstracted into the following three error channels: 1) readout error; 2) depolarizing error; and 3) thermal relaxation error. These error channels are then integrated into the readout error model and the gate error model.

Specifically, the process of gate error modeling refers to first executing the ideal operation and then randomly sampling from the noise operation set with special error probabilities, then inserting it after the original ideal gate, i.e., executing the ideal and sampled noise gates. For example, if the gate error probability of Pauli- X gate is $\{\text{"id"}: 0.9986, \text{"y"}: 0.0003, \text{"z"}: 0.0007, \text{and "reset"}: 0.0004\}$, where id is an identity gate, refers to the ideal gate. Next, if the "z" error gate is sampled, it will execute "Pauli- X +Pauli- Z " gates; if the "reset" error is sampled, the quantum state will be reset to the initial state.

B. Spatial and Temporal Biases Definitions

Since readout and gate errors are closely related to imperfect hardware operation and environmental coupling, leading to a non-Markovian process of random evolution. This stochastic noise severely damages the accuracy of quantum computation. Due to the unpredictable nature of noise, practitioners must regularly conduct quantum noise calibration. However, current calibration techniques can only mitigate noise to a certain extent but cannot eliminate it. Following calibration, noise is adjusted within an acceptable range with its maximum boundary determined, yet its specific distribution and magnitude remain random. This directly leads to the existence of *noise biases*. In practical scenarios, we abstract noise biases into two types: 1) *spatial bias* and 2) *temporal bias*, depicted as follows.

- 1) *Spatial Bias*: For different devices, it is natural to have noise bias due to the different manufacturing processes, hardware operational performance, and environmental coupling. Specifically, for readout error, the probability matrix varies across different devices and environments. For gate error, the probability of different flip types is spatially diverse. *Spatial bias* refers to this noise discrepancy on different devices. Formally, consider two QCs D_1 and D_2 , their noise profiles are different at the same time T_1 , i.e.,

$$\mathcal{N}(D_1, T_1) \neq \mathcal{N}(D_2, T_1). \quad (5)$$

- 2) *Temporal Bias*: The noise bias also exists at different times for the same quantum device, hence imperfect calibration processes and non-Markovian noise variations. In other words, for readout error, the probability matrix varies at different time points. For gate error, the probability of different flip types is temporally diverse. Formally, for QCs D_1 , *temporal bias* means the noise presentations are different at T_1 and T_2 , i.e.,

$$\mathcal{N}(D_1, T_1) \neq \mathcal{N}(D_1, T_2). \quad (6)$$

IV. METRIC OF CIRCUIT SEQUENCE CORRECTNESS

To characterize the effect of noise on the execution results of QNN circuits (to cope with the second challenge in Section I), we define a novel metric, CSC, which aims to intuitively reflect the reliability of correct execution of QNN models under different noise environments. The quantum circuit is composed of a series of cascades of quantum gate operations, and errors may occur on each gate operation (due to imperfect physical manufacturing techniques and environmental interactions).

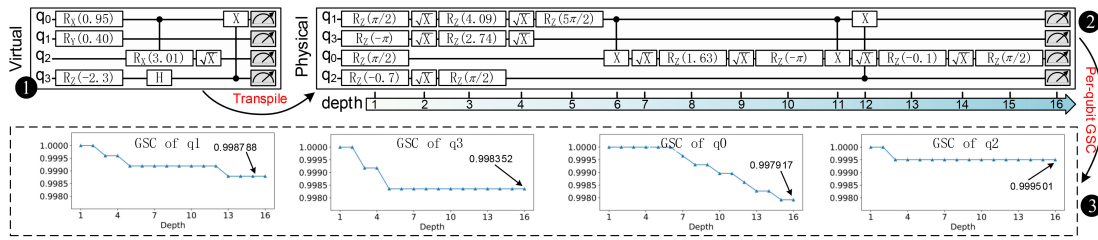


Fig. 4. Illustrative explanation of CSC/GSC calculation based on *IBMQ Quito*. First, the virtual circuit needs to be transpile (1→2) to the physical circuit. Then, we calculate the GSC for each qubit under 8 July 2021 noise, plotting in each subplot in ③. Finally, calculate the CSC score by averaging the GSC of all qubits.

So-called CSC refers to the correct probability cumulative multiplication of the gate operations (e.g., the single-qubit gate, double-qubit gate, multiqubit gate) according to the circuit sequence. Given a noise profile \mathcal{N}_p that contains specific gate error probabilities⁵ [31]. Consider a quantum circuit \mathcal{C} operating on n qubits in a noisy environment characterized by noise profile \mathcal{N}_p . The CSC formal definition is as follows:

$$\text{CSC}(\mathcal{C}, \mathcal{N}_p) = \sum_{i=0}^n \frac{\text{GSC}_i(\mathcal{C}, \mathcal{N}_p)}{n} \quad (7)$$

where GSC_i denotes the sequence of gate correctness of the i th qubit,⁶ ranging from [0, 1]. Then CSC is the average GSC of all qubits across the circuit, which is calculated as follows:

$$\text{GSC}_i(\mathcal{C}, \mathcal{N}_p) = \prod_{j=0}^{\text{len}(G_i)} \left(1 - \mathcal{N}_p[G_i^j]\right) \times (1 - \mathcal{N}_p(R_i)) \quad (8)$$

where the qubit q_i contains the sequential gate operations list G_i , readout error $\mathcal{N}_p(R_i)$, and gate error $\mathcal{N}_p[G_i^j]$. $\text{len}(G_i)$ represents the number of gate operations in G_i . To validate CSC's effectiveness, we compared the noise evaluation metrics regarding their correlation with accuracy, detailed in Section VI.

To make the CSC calculation process clearer, we draw the illustrative explanation in Fig. 4, whose noise profile from *IBMQ Quito* on 8 July 2021. On the top part of the figure, the process (1→2) displays the circuit transpile process [32] from the virtual circuit to the physical deployable circuit, to match the quantum chip topology. The transpile process includes qubit mapping [33], [34], [35], [36], gate decomposition, physical circuit optimization, etc. Among them, qubit mapping⁷ means placing the virtual circuit on physical

qubits to match the chip topology of the QC, e.g., *virtual-to-physical* = {1:3, 3:2, 0:1, 2:0}. Gate decomposition means translating the virtual circuit into a series of basis gates, making it executable on the physical device. The basis gates on *IBMQ Quito* include {RZ, SX, X, CX}, where SX denotes \sqrt{X} . Subsequently, we can calculate the CSC (2→3) reference (7) based on the physical circuit. For the bottom of Fig. 4 shows the GSC for 4 qubits (according to the order after qubit mapping, i.e., [1, 3, 0, 2]), in which the calculation of each qubit is independent. We can observe that the GSC gradually becomes smaller as the circuit depth increases, which means that deeper circuits tend to be more susceptible to noise. Specifically, q_0 has the most gates (maximum depth) and its GSC is the smallest, which is 0.997917. In this case, the GSC is 0.997917, 0.998788, 0.999501, and 0.998352 for q_0 , q_1 , q_2 , and q_3 , respectively. So the CSC of the circuit is

$$\text{CSC}(\mathcal{C}, \mathcal{N}_p) = (0.997917 + 0.998788 + 0.999501 + 0.998352)/4 = 0.9986395. \quad (9)$$

V. DESIGN OF QUST

Initially, we introduce noise injection and circuit execution to enhance the model's awareness of quantum noise. Subsequently, we leverage CSC to estimate confidence levels under different noise conditions, utilizing these metrics as training weights to enhance the model's dynamic anti-noise capability. To concurrently strengthen the model's capacity to handle diverse noise environments, we introduce multiscale noise-aware training, enabling simultaneous sensing of multiple noise levels during each iteration. Ultimately, through the loss integration, we integrate noise-free and multiscale noise-aware training losses, facilitating model parameter updates. The QuST training pipeline is shown in Algorithm 1, consisting primarily of three parts: 1) noise-free inference; 2) noise inference; and 3) integrated loss function, followed by updating model parameters. The total loss function is divided into two parts: 1) cross-entropy is used in the noise-free training to enhance the model's accuracy and 2) KL divergence is used to measure the noise loss in noise-aware training, measuring the difference between noisy inference and noise-free model inference. This forces the model to approximate noise-free inference even in noisy conditions, thereby enhancing its robustness. Ultimately, by integrating cross-entropy and KL divergence losses, the model achieves higher-classification performance and greater robustness.

⁵Error probabilities, noise profile \mathcal{N}_p contains error information for each gate and measurement for each qubit. For example, $\mathcal{N}_p[SX_0] = 0.000347$ refers to the error probability of single-qubit gate SX in q_0 is 0.0347%, $\mathcal{N}_p[R_2] = 0.043900$ represents the error probability of measurement in q_2 is 4.39%, and $\mathcal{N}_p[CX_{(1,3)}] = 0.010173$ means the error probability of double-qubit gate CX (whose control qubit is q_1 and the target qubit is q_3) is 1.0173%. Noteworthy, the error probability of multiqubit circuits could be asymmetric, e.g., it is possible that $\mathcal{N}_p[CX_{(1,3)}] \neq \mathcal{N}_p[CX_{(3,1)}]$.

⁶Note that for 1-qubit gates, the fidelity is directly multiplied by the corresponding qubit's error rate. For 2-qubit gates, only the error rate of the qubit involved in the controlled gate is accumulated.

⁷Qubit mapping will be performed when the physical topology cannot match the requirements of the virtual circuit. For instance, the virtual circuit includes a two-qubit gate $CX_{(3,1)}$, but there is no physical connection between q_1 and q_3 on the chip. Then, the SWAP gate will be introduced to exchange the contents through additional qubits, making it possible to implement $CX_{(3,1)}$. Hence, efficient qubit mapping minimizes the need for introducing SWAP gates when implementing the virtual circuit.

Algorithm 1 Training Pipelines of QuST

Input: The training dataset \mathcal{D} , the noise dataset $\mathcal{S} = \{N_1, N_2, \dots, N_m\}$ characterized by spatial and temporal biases across m days.

Output: The robust QuST QNN model M

- 1: Initialize model parameters for M
- 2: **for** $(x, y) \in (\mathcal{X}, \mathcal{Y})$ in \mathcal{D} **do**
- 3: # Noise-free training setting
- 4: $\hat{y}_{\text{ideal}} = M(x, \emptyset) \leftarrow \emptyset$ denotes noise-free inference
- 5: Compute the noise-free loss
- 6: $L_{\text{ideal}} = \text{cross_entropy}(\hat{y}_{\text{ideal}}, y)$
- 7: Sampling C_n items from noise dataset \mathcal{S} :
- 8: $\mathcal{S}_{\text{sub}} = \{N_1, N_2, \dots, N_{C_n}\}$
- 9: # Multi-scale noise-aware training setting
- 10: Initialize the multi-scale noise-aware loss $L_{\text{noise}} = 0$
- 11: **for** i in $\text{range}(C_n)$ **do**
- 12: $\hat{y}_{\text{noise}} = M(x, N_i) \leftarrow$ noise-aware inference
- 13: $\text{CSC}(N_i) \leftarrow$ Calculate the CSC of N_i
- 14: Compute noisy loss l_i with CSC as the weight:
- 15: $l_i = \text{CSC}(N_i) \times \text{KL}(\hat{y}_{\text{ideal}}, \hat{y}_{\text{noise}})$
- 16: **end for**
- 17: # Integrate multi-scale noise-aware into loss function
- 18: Accumulate and average the multiple noise-aware loss
- 19: $L_{\text{noise}} = \frac{1}{C_n} \times \sum_{i=0}^{C_n} l_i$
- 20: Combine Overall Loss L_{overall}
- 21: $L_{\text{overall}} = L_{\text{ideal}} + L_{\text{noise}}$
- 22: Update QNN model parameters
- 23: $M = \text{back_propagation}(M, L_{\text{overall}})$
- 24: **end for**
- 25: **return** The trained M

A. Noise-Free and Noise-Aware Circuit Execution

Noise-free circuit execution means that all quantum operations are executed correctly, referring to the error probability 0%. Conversely, *noise-aware circuit execution* implies the presence of a random probability for potential errors in quantum operations. Specifically, within the QNN model architecture, the model entity is designed to incorporate the *Noise Model* attribute, corresponding to a noise configuration. Therefore, during a specific gate operation, a randomly sampled probability determines whether it is performed correctly. Gate operations involve sampling from a gate set consisting of an identity gate (representing the correct gate) and potential error gates. Sampling the identity gate implies noise-free execution, whereas sampling an error gate entails execution of both the correct and error gates. By assigning historical noise data from the real quantum machines to the gate and measurement operations⁸ and subsequently performing model inference, we call this random execution process *noise-aware circuit execution*. Using these two circuit executions in the QNN model inference, the mathematical formalization is expressed as follows. Let M denote a QNN classifier aiming to achieve $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $x \in \mathcal{X} \in \mathbb{R}^d$ represents the input, and $\mathcal{Y} = \{1, 2, \dots, C\}$ (where C

denotes the number of classification categories). Let y denote the ground-truth label of x , and the model is deemed to have classified correctly when $f(x) = y$. The term noise-free inference refers to

$$\hat{y}_{\text{ideal}} = M(x, \emptyset) \quad (10)$$

where \emptyset denotes a noise-free environment. We use the cross-entropy function to calculate the noise-free loss part L_{ideal}

$$L_{\text{ideal}} = \text{Cross_Entropy}(\hat{y}_{\text{ideal}}, y). \quad (11)$$

The noise-aware inference for the i th noise profile as follows:

$$\hat{y}_{\text{noise}} = M(x, N_i). \quad (12)$$

B. Integrate CSC to Loss Function

Since quantum noise evolves randomly, resulting in a constantly changing noise magnitude, it damages the performance of QNN models. To enhance the ability of the model to dynamically adapt to different magnitudes of quantum noise, we utilize the noise evaluation metric CSC as a training weight for the loss function. This assigns higher confidence to smaller noise levels and lower confidence to higher-noise levels. This strategy is to sustain the overall training performance and prevent terrible noise scenarios from affecting the training process. The calculation of the single loss for noisy inference is as follows: (Algorithm 1 lines 12–15)

$$l_i = \text{CSC}(N_i) \times \text{KL}(\hat{y}_{\text{ideal}}, \hat{y}_{\text{noise}}) \quad (13)$$

where the Kullback–Leibler divergence [37] is used to evaluate the distribution similarity between noise-aware inference \hat{y}_{noise} and noise-free inference \hat{y}_{ideal} , improving the model's noise resilience. Overall, our objective is to maintain high accuracy in noise-free or small noise conditions and mitigate accuracy deterioration in the presence of significant noise. To achieve this goal of enabling the model to perceive and respond to various noise levels, we assign different training weights to different noise levels. Utilizing CSC as a training weight aligns with our intentions. When encountering substantial noise, the CSC value tends to decrease, thereby reducing the loss function contribution weight. This training strategy prevents extremely terrible noise scenarios from damaging the overall training effectiveness. As the saying goes, “One bad apple spoils the whole bunch.”

C. QNN Training Against Dynamic Noise

The QC undergoes frequent calibration and physical environmental interactions, resulting in noise characteristic changes across temporal and spatial biases. In Fig. 5, the noise space is affined in a 3-D hyperplane, the 3-dim vector N_i represents a noise model for a specific day, and the big blue sphere represents the upper boundary of the noise, i.e., the maximum allowable error range. If the noise vector exceeds the blue sphere boundary, the QC will be recalibrated. After calibration, errors are suppressed into the red sphere, i.e., the lower-error bound or infimum sphere. Fig. 5(b) shows that even after recalibration, noticeable disparities persist among different noise models N_i . However, most existing methods,

⁸We explain the implementation details of loading the noise configuration from files in Section VII-A, which is based on the *Qiskit* library.

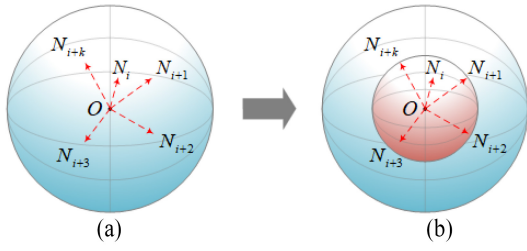


Fig. 5. Quantum noise visualization sphere: (a) Shows the calibrated single-day noise model as a vector N_i from different days and (b) red sphere represents a noise infimum sphere comprising multiple noise model vectors, while the blue sphere indicates recalibration with maximum error tolerance limits.

e.g., QuantumNAT [4] and QuCAD [15] only consider noise from a single day and heavily depend on the noise conditions during training. Once the noise changes, these methods may experience a sharp decline in model performance. Therefore, they cannot effectively handle dynamic noise in real-world quantum devices. To address this issue and improve generalizability across various noise scenarios, we introduce multiscale noise-aware training to combat dynamic noise. It contains the following steps:

Noise Sampling: We use noise data from the QCs as our noisy dataset. To enhance the model's noise resilience, we randomly sample C_n items of noise data for each training epoch, thereby expanding the noise perception scope through multiscale noise-aware training. Formulaically, each day's noise data is denoted as a noise vector N_i , forming a noise dataset $\mathcal{S} = \{N_1, N_2, \dots, N_m\}$ where $m \in \mathbb{R}$ represents the number of days. During each training epoch, we randomly sample C_n noise vectors from noise dataset \mathcal{S} to constitute a noise subset $\mathcal{S}_{\text{sub}} = \{N_1, N_2, \dots, N_{C_n}\}$ for each training iteration. Importantly, we do not assume any specific noise distribution or analyze noise patterns. Instead, we quantify how noise affects a QNN model under different QC configurations and use this as a perturbation factor in model training to enhance robustness.

Multiscale Noise-Aware Loss Function: To further improve the model's robustness against spatial and temporal biases of various noise magnitudes, we custom a multiscale noise-aware loss function. We propose injecting noise from multiple days to collaboratively update the model gradient to expand the noise adaptation space of the model (Algorithm 1 lines 9–16). That is, to fit the model to the entire infimum sphere, which is depicted in the red sphere in Fig. 5(b). Specifically, during each training iteration, randomly sample C_n noise profiles from the noise dataset of temporal and spatial biases. Next, perform noise-aware circuit inference, then accumulate and average the multiple noisy inference loss L_{noise}

$$L_{\text{noise}} = \frac{1}{C_n} \times \sum_{i=0}^{C_n} l_i \quad (14)$$

where $l_i = \text{CSC}(N_i) \times \text{KL}(\hat{y}_{\text{ideal}}, \hat{y}_{\text{noise}})$ utilize KL divergence (Kullback–Leibler divergence) to measure the difference between model outputs distributions [38] (N_i refers to the noise of i th sampling). We use it to quantify the difference between noisy and noise-free inference results, aiming to make the noisy inference results closer to the noise-free

(ideally) inference results, which correspond to lines 11–16 in Algorithm 1. Then calculate the overall loss function by summing noise-free and noise inference, which is expressed as follows:

$$\begin{aligned} L_{\text{overall}} &= L_{\text{ideal}} + L_{\text{noise}} \\ &= L_{\text{ideal}} + \frac{1}{C_n} \sum_{i=0}^{C_n} \text{CSC}(N_i) \times \text{KL}(\hat{y}_{\text{ideal}}, \hat{y}_{\text{noise}}). \end{aligned} \quad (15)$$

Based on such loss function design, the model can maintain high-noise-free classification performance (i.e., small L_{ideal}) and improve the robustness of the model against various noises (i.e., small L_{noise} , tends to reduce the effect of noise on the model accuracy). During each training iteration, the model parameters are jointly updated using both the noise-free loss L_{ideal} and the multiscale noise-aware loss L_{noise} (Algorithm 1 lines 17–23). Overall, our objective is to increase the QNN model elasticity against ever-changing noises. The goal of multiscale noise-aware training is to tailor the model to the error bounds across the entire range of acceptable errors, as depicted inside the red sphere in Fig. 5(b).

VI. ADVANTANGES OF QUST

A. Expressiveness of CSC Metric

To assess the influence of noise on QNN circuit execution and use it as a confidence weight for noise-injected training, enabling the model to adapt to various noise levels. Therefore, we introduce a novel metric named CSC, which measures the reliability of a circuit execution in noisy environments, as explained in Section IV. To confirm that CSC provides a more accurate characterization of QNN circuit execution accuracy in noisy environments, this article introduces two popular metrics for comparison.

- 1) *Fidelity*, which is defined as a quantitative measure of the closeness between two quantum states [18]. For example, comparing the quantum states prepared by the circuit in noise-free and noisy configurations, thereby estimating the reliability of executing the circuit with noise.
- 2) *PST fidelity* (Probability of Successful Trials fidelity) is an optimization version [18] of Fidelity that mitigates the exponential simulation cost of noise-free quantum states. It computes the ratio of the invariant quantum state (all zeros) by running a circuit and its reverse. PST fidelity uses Monte Carlo simulation to assess the reliability of the circuit in noisy environments by determining the ratio of correct results.

To evaluate the relationship between three different noise characterization metrics (Fidelity, PST fidelity, CSC) with the QNN model accuracy, we perform MNIST-4 on IBMQ Quito noise simulation using the “RX+RZ+CRX” circuit as shown in Fig. 6, and the correlation evaluation results. The results are shown in Fig. 7, where the black dashed line represents the ideal result as a reference, where Metric is equal to Accuracy. As observed in the orange line in Fig. 7, PST fidelity (orange line) is ineffective in characterizing model accuracy, with model accuracy ranging from 0.5–0.9, but PST fidelity shows minimal variation. Conversely, both Fidelity and

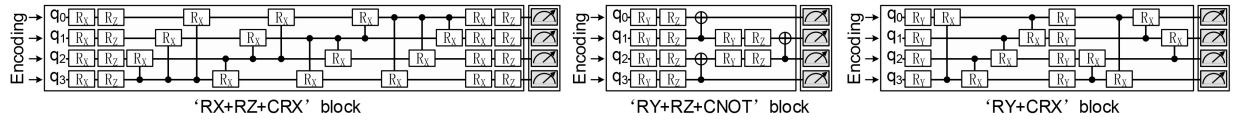


Fig. 6. Three circuits block in our evaluation and the QNN model is cascaded based on two repeated blocks.

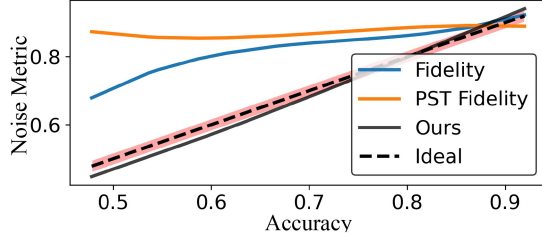


Fig. 7. Compared QuST and other metrics with accuracy.

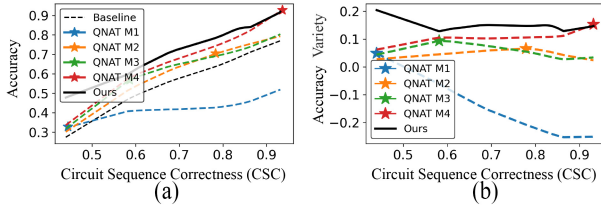


Fig. 8. Depict “model accuracy and CSC” among different training methods, t has consistently robust accuracy. The CSC value indicates the noise level, with smaller values indicating higher-noise levels. a) Noisy Evaluation, b) QuantumNAT VS Ours QuST.

CSC showcase a positive correlation with model accuracy, and CSC shows stronger alignment and sensitivity to noise fluctuations. This discrepancy arises because CSC derives from the cumulative calculation of correct circuit sequence execution probabilities, offering a direct reflection of circuit reliability. In contrast, Fidelity and PST primarily account for disparities in circuit output results and may introduce estimation bias due to their limited awareness of the intermediate gate sequence. Overall, the ranking of the positive correlation between characterization metrics and model accuracy is $CSC > Fidelity > PST$ fidelity. Hence, considering CSC effectively represents model accuracy, we can incorporate it into the loss calculation to guide the model training process.

B. Multiscale Noise-Aware Adaptation

We further investigate the correlation between CSC and QNN accuracy across single-noise perception and multiscale noise-aware perception training methods. The experimental results are shown in Fig. 8. We conduct a 190-day comparative experiment using the “RX+RZ+CRX” circuit for MNIST-4 classification, utilizing noise simulation from IBMQ Quito, which involves three methods: 1) baseline (noise-free); 2) QuantumNAT (abbreviated as QNAT, using single noise-injection training method); and 3) QuST (multiscale noise-aware perception). Notably, QuantumNAT employed multiple CSC magnitudes for training, with the corresponding training model indicated by a five-pointed star, which QuantumNAT M1–M4 with $CSC = [0.44, 0.582, 0.769, 0.932]$ and correspond to dates [19 October 2021, 29 December 2021, 7 September 2021, and

5 November 2021]. Fig. 8(a) shows the model accuracy across various CSC scales, while b) illustrates the accuracy improvement achieved by QuantumNAT (QNAT) and QuST over the baseline, indicating enhanced anti-noise capabilities (subtracting the accuracy of baseline). In Fig. 8(a), the black dashed line represents the baseline. Notably, as CSC values increase along the x -axis, model accuracy consistently improves, serving as a reference for accuracy during CSC variations. For a detailed analysis, we first examine the performance of QuantumNAT M1–M4. The CSC value used in model training is represented by a five-pointed star. We have valuable insights.

- 1) Excessive noise in the noise injection method does not improve the model’s noise resilience, instead, it introduces considerable randomness, making it challenging to train the model effectively. For instance, QuantumNAT M1 with $CSC = 0.44$ is the largest noisy configuration among M1–M4 and still performs poorly even in small noise-magnitude environments.
- 2) Comparing the above insights, we can observe that trained with medium-scale noise, such as QuantumNAT M2–M4, exhibit better-noise adaptability. This improvement is most prominent when the test noise magnitude closely matches the training noise magnitude, as evident in Fig. 8(b) of the QuantumNAT five-pointed star at the peak of each curve. However, their ability to adapt to noise is limited when confronted with noise magnitudes that are different from the training noise magnitude. This highlights that the noise adaptability of QuantumNAT is heavily reliant on the specific noise configuration during training, as its objective is to achieve optimal performance under the given training noise conditions.
- 3) When comparing QuST, represented as the black line in Fig. 8, our approach incorporates a CSC weighting mechanism tailored to evaluate different noise levels. In scenarios with small noise, the model assigns higher confidence (i.e., CSC) to the correct circuit execution and adjusts training loss weights accordingly. This prevents excessive noise from degrading model performance. This weighting mechanism enables the QNN model to consistently maintain high accuracy in various small noise conditions and minimize accuracy degradation when dealing with substantial noise challenges.

Overall, the experimental results highlight the strong correlation between QuantumNAT’s noise resilience and the magnitude of the training noise. Optimal performance occurs when the noise magnitude matches the training noise, enhancing the model’s anti-noise capability. Specifically, excessive noise negatively affects accuracy, while overly small noise provides limited resilience improvement. In contrast, our QuST framework employs weighted noise environment evaluation

and incorporates multiscale noise perception, enabling it to combat noise across various scales effectively.

VII. EVALUATION

In this section, we comprehensively evaluate the model's anti-noise ability across temporal and spatial biases in noise magnitudes and conduct real QC experiments and ablation studies.

A. Experiment Setup

Platform: We train the QNN model using noisy simulations only once and validate it on the real IBMQ machines: *Quito*, *Lima*, *Belem*, and *Manila*, with a 4-qubit QNN model. For simulation validation, we utilize 190 days of historical noise data from these machines, and the results are averaged. We implement QuST using TorchQuantum [4] for noise simulation and batch experiments, benefiting from fast GPU support and efficient batch sample training compared to other libraries.

Datasets: We use the popular image classification dataset MNIST [39] and Fashion-MNIST. The MNIST contains 70K 28×28 grayscale images of handwritten digits from 0 to 9, and the Fashion dataset contains 70K 28×28 grayscale images of 10 classes of clothing, which include t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. Specifically, we mainly conduct MNIST-4 (0, 1, 2, 3) and Fashion-4 (t-shirt/top, trouser, pullover, dress). We split the dataset 90% of the images for training and validation, and the remaining 10% consists of our test set. Before feeding the QNN model, images will be centrally cropped from 28×28 to 24×24 , then down-sampled to 4×4 with average pooling, the same as QuantumNAT [4]. To construct our noise dataset, we refer to the VACSEN dataset [17], which encompasses calibration data collected over 190 days, spanning from July 2021 to January 2022, across 11 IBMQ QCs. The dataset contains multiple physical hardware properties (e.g., decoherence time, error rate, etc.) collected from the IBM Quantum platform [40]. We mainly selected 4 devices from this VACSEN, including *Quito*, *Lima*, *Belem*, and *Manila*.

Experiment Setting: Our experiments utilize calibration data from 4 IBMQ QCs over 190 days. We set the noise factor $F = 1$, and the final result is the average of $m = 190$ days of noise simulation tests without specific annotation. QuST set samples $C_n = 3$. In Section VII-F testing different noise magnitudes, we set noise factor $F = \{0, 0.5, 1, 2\}$, and include four scenarios: 1) “No noise;” 2) “Read” noise; 3) “Gate” noise; and 4) “Read and Gate” noise.

QNN Model Implementation: Fig. 6 depicts our QNN model circuit architecture, we use amplitude encoding, trainable quantum layers use “RX+RZ+CRX,” “RY+RZ+CNOT,” and “RY+CRX” circuit templates shown in Fig. 6, and measurement adopts the Pauli-Z basis. In this article, we construct our quantum layer by cascading two blocks for each circuit. Finally, the measurement results apply a Softmax function to output the model prediction results, and then calculate the loss function and update quantum layers' trainable parameters. We use qiskit [21] for circuit transpile, setting optimization –

TABLE I
TEMPORAL BIAS EVALUATION ON MNIST-4 AND FASHION-4

| Data | Circuit | Noise | Baseline | QuantumNAT | Ours |
|-----------|--------------|-------|----------|----------------|-----------------------|
| MNIST-4 | ‘RX+RZ+CRX’ | ✗ | 0.869 | 0.824 (↓0.045) | 0.864 (↓0.005) |
| | | ✓ | 0.664 | 0.723 (↑0.059) | 0.805 (↑0.141) |
| | ‘RY+RZ+CNOT’ | ✗ | 0.842 | 0.801 (↓0.041) | 0.835 (↓0.007) |
| | | ✓ | 0.631 | 0.707 (↑0.076) | 0.782 (↑0.151) |
| | ‘RY+CRX’ | ✗ | 0.854 | 0.819 (↓0.035) | 0.848 (↓0.006) |
| | | ✓ | 0.659 | 0.716 (↑0.057) | 0.793 (↑0.134) |
| Fashion-4 | ‘RX+RZ+CRX’ | ✗ | 0.865 | 0.840 (↓0.025) | 0.861 (↓0.004) |
| | | ✓ | 0.690 | 0.729 (↑0.039) | 0.811 (↑0.121) |
| | ‘RY+RZ+CNOT’ | ✗ | 0.850 | 0.832 (↓0.018) | 0.849 (↓0.001) |
| | | ✓ | 0.702 | 0.733 (↑0.031) | 0.783 (↑0.081) |
| | ‘RY+CRX’ | ✗ | 0.872 | 0.838 (↓0.034) | 0.857 (↓0.015) |
| | | ✓ | 0.698 | 0.711 (↑0.013) | 0.796 (↑0.098) |

level = 0, which simply translates multiqubit gates into basis gates supported by IBMQ devices and matches the quantum chip topology without performing noise optimization. Noise simulations using IBMQ calibration data to calculate gate and readout error probabilities.

Baselines: We compare QuST with two training methods: 1) baseline (noise-free) and 2) QuantumNAT (noise-injection, aka QNAT). To ensure fairness, QuantumNAT is trained on four different days with corresponding CSC scores: [0.44, 0.582, 0.769, 0.932], which correspond to dates [19 October 2021, 29 December 2021, 7 September 2021, and 5 November 2021], with models denoted as “QNAT M1–M4”. Notably, “QNAT M1” exhibits significant underperformance due to adverse noise conditions. Therefore, except for the experiment discussed in Section VI on single-noise and multi-noise adaptation, we use the average accuracy of “QNAT M2–M4” as the accuracy of the QuantumNAT model.

B. Evaluation of Temporal Bias

We conduct temporal bias evaluation on IBMQ *Quito* noise simulation, spanning 190 days of calibration data. Our testing involves two classification tasks (MNIST-4 and Fashion-4) across three different circuit blocks. In Table I, the performance ranking for the three circuit blocks on the two datasets is “RX+RZ+CRX” > “RY+CRX” > “RY+RZ+CNOT,” indicating that the “RX+RZ+CRX” circuit exhibits greater expressiveness. Specifically, Table I reveals that the baseline (noise-free training) performs well in noise-free environments but suffers a significant performance drop, up to 21.1% in the “RY+RZ+CNOT” circuit of the MNIST-4. We visualize the daily CSC scores of 190 days on IBMQ *Quito*, which is depicted in Fig. 9. To provide a clear representation, we conduct a 30-day random (slash noted in Fig. 9). This sampling includes various noise magnitudes from 190 days to demonstrate daily performance fluctuations using the “RX+RZ+CRX” circuit for MNIST-4 on IBMQ *Quito*. These results are illustrated in Fig. 10, the x-axis refers to a relative time sequence spanning 190 days, ranging from 7 July 2021 to 12 January 2022. For instance, the date “17 July 2021” is abbreviated as “D10,” and the y-axis corresponds to the F1-score. QuantumNAT utilizes the “QNAT M3” model, trained with the 7 September 2021 CSC(= 0.769) noise configuration, as it provides relatively stable accuracy.

Based on Fig. 10, the models perform notably well on days D85 and D127, particularly for baseline. Due to high CSC

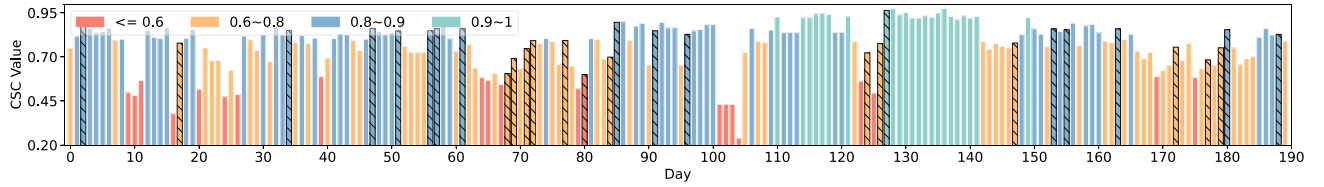


Fig. 9. CSC values from 190-day quantum noise in IBMQ Quito, with slashed bars denoting 30-day samples in Fig. 10.

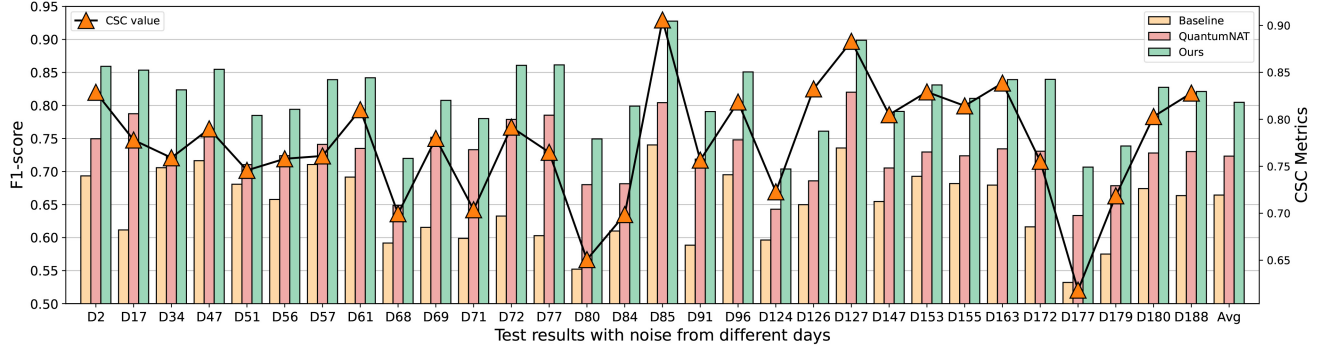


Fig. 10. Visualizing daily F1-score and CSC over 30 days, using the “RX+RZ+CRX” circuit for MNIST-4 on IBMQ Quito.

TABLE II
TESTING OF SPATIAL BIAS TOWARD MNIST-4 AND FASHION-4

| Dataset | Circuit Design | Noise | Lima | | | Belem | | | Manila | | |
|-----------|----------------|-------|----------|------------|--------------|----------|------------|--------------|----------|------------|--------------|
| | | | Baseline | QuantumNAT | Ours | Baseline | QuantumNAT | Ours | Baseline | QuantumNAT | Ours |
| MNIST-4 | ‘RX+RZ+CRX’ | ✗ | 0.870 | 0.777 | 0.869 | 0.871 | 0.741 | 0.864 | 0.869 | 0.801 | 0.853 |
| | | ✓ | 0.622 | 0.754 | 0.785 | 0.620 | 0.721 | 0.770 | 0.660 | 0.736 | 0.765 |
| Fashion-4 | ‘RY+CRX’ | ✗ | 0.674 | 0.618 | 0.672 | 0.673 | 0.645 | 0.656 | 0.673 | 0.662 | 0.654 |
| | | ✓ | 0.625 | 0.650 | 0.671 | 0.620 | 0.642 | 0.662 | 0.641 | 0.643 | 0.650 |

of 0.906 and 0.880. Conversely, the baseline F1-scores for D177 and D80 are much lower due to lower-CSC values of 0.618 and 0.65 on those days. Notably, there is a substantial performance gap between QuantumNAT and baseline on days D17, D72, and D77, driven by their respective CSC values of 0.778, 0.792, and 0.760. On these days, the baseline exhibits limited effectiveness, while the proximity of QuantumNAT’s training with $CSC = 0.769$ on 7 September 2021 results in a more favorable performance. The average F1-score of QuST is $\sim 8\%$ higher than QuantumNAT. Overall, baseline accuracy tends to vary with CSC, where smaller noise magnitudes lead to better performance, while bigger noise magnitudes result in poorer performance. QuantumNAT is more inclined to adapt to noise of a comparable magnitude to the noise injected during the training process, and its performance is unstable in other noise magnitudes environments. In contrast, our model exhibits overall stability in performance, especially when dealing with better-CSC scores, because our model is designed to prioritize fitting smaller noise magnitudes. For a more comprehensive analysis of CSC and model performance, please refer to deep insights in Section VI.

C. Evaluation of Spatial Bias

For spatial bias, we use the “RX+RZ+CRX” circuit on MNIST-4 and the “RY+CRX” circuit on Fashion-4, tested on IBMQ *Lima*, *Belem*, and *Manila*. As shown in Table II, baseline achieves peak performance in noise-free tests but suffers significant degradation in noisy tests, with

accuracy losses of up to 29%. In noisy environments on MNIST-4, QuST outperforms baseline by 10.5%–16.3% and QuantumNAT by 2.9%–5%, with an average improvement of 13.9% over baseline and 3.7% over QuantumNAT. For Fashion-4, QuST outperforms baseline by 10.9%–14.6% and QuantumNAT by 2%–3.7%, with an average improvement of 14.2% over baseline and 2.6% over QuantumNAT.

D. Extended Experiments on Temporal and Spatial Biases

To further evaluate the performance of QuST, we conduct cross-testing experiments and smaller noise magnitudes evaluation.

- 1) *Cross-Testing Evaluation*: To thoroughly evaluate the effect of noise variations on model accuracy, we conducted comprehensive cross-testing using the “RX+RZ+CRX” circuit for the MNIST-4 classification on 4 IBMQ machines. The result is shown in Table III, which shows the average accuracy over a 190-day testing period. In noisy testing environments, the highest accuracy is usually achieved on the training machine, while accuracy on other machines depends on their inherent noise characteristics. In the noise test on four machines (IBMQ *Quito*, *Lima*, *Belem*, and *Manila*), compared to the baseline, QuantumNAT achieved average accuracy improvements of 5.48%, 3.48%, 5.6%, and 4.88%, while QuST achieved increased by 11.3%, 9.1%, 11.1%, and 11.45%, respectively.

TABLE III
EXTENDED EXPERIMENTS ON TEMPORAL AND SPATIAL BIASES EVALUATION OF MNIST-4 USING THE “RY+RZ+CNOT” CIRCUIT

| Train | Test Noise | Quito | | Lima | | Belem | | Manila | |
|--------|------------|------------------------------|-----------------------------------|------------------------------|-----------------------------------|------------------------------|-----------------------------------|------------------------------|-----------------------------------|
| | | \times | \checkmark | \times | \checkmark | \times | \checkmark | \times | \checkmark |
| Quito | Baseline | 0.869 | 0.657 | 0.872 | 0.694 | 0.859 | 0.667 | 0.863 | 0.615 |
| | QuantumNAT | 0.824 ($\downarrow 0.045$) | 0.723 ($\uparrow 0.066$) | 0.843 ($\downarrow 0.029$) | 0.714 ($\uparrow 0.020$) | 0.827 ($\downarrow 0.032$) | 0.721 ($\uparrow 0.054$) | 0.820 ($\downarrow 0.043$) | 0.694 ($\uparrow 0.079$) |
| | Ours | 0.864 ($\downarrow 0.005$) | 0.805 ($\uparrow 0.148$) | 0.866 ($\downarrow 0.006$) | 0.762 ($\uparrow 0.068$) | 0.852 ($\downarrow 0.007$) | 0.761 ($\uparrow 0.094$) | 0.858 ($\downarrow 0.005$) | 0.757 ($\uparrow 0.142$) |
| Lima | Baseline | 0.871 | 0.735 | 0.871 | 0.744 | 0.869 | 0.747 | 0.864 | 0.651 |
| | QuantumNAT | 0.856 ($\downarrow 0.015$) | 0.746 ($\uparrow 0.011$) | 0.845 ($\downarrow 0.026$) | 0.762 ($\uparrow 0.018$) | 0.833 ($\downarrow 0.036$) | 0.754 ($\uparrow 0.007$) | 0.847 ($\downarrow 0.017$) | 0.754 ($\uparrow 0.103$) |
| | Ours | 0.872 ($\downarrow 0.001$) | 0.824 ($\uparrow 0.089$) | 0.869 ($\downarrow 0.002$) | 0.836 ($\uparrow 0.092$) | 0.863 ($\downarrow 0.006$) | 0.796 ($\uparrow 0.049$) | 0.856 ($\downarrow 0.008$) | 0.785 ($\uparrow 0.134$) |
| Belem | Baseline | 0.871 | 0.664 | 0.869 | 0.709 | 0.871 | 0.681 | 0.863 | 0.622 |
| | QuantumNAT | 0.849 ($\downarrow 0.022$) | 0.726 ($\uparrow 0.062$) | 0.834 ($\downarrow 0.035$) | 0.727 ($\uparrow 0.018$) | 0.841 ($\downarrow 0.030$) | 0.731 ($\uparrow 0.050$) | 0.836 ($\downarrow 0.027$) | 0.716 ($\uparrow 0.094$) |
| | Ours | 0.870 ($\downarrow 0.001$) | 0.793 ($\uparrow 0.129$) | 0.868 ($\downarrow 0.001$) | 0.776 ($\uparrow 0.067$) | 0.865 ($\downarrow 0.006$) | 0.790 ($\uparrow 0.109$) | 0.857 ($\downarrow 0.006$) | 0.760 ($\uparrow 0.138$) |
| Manila | Baseline | 0.869 | 0.639 | 0.859 | 0.672 | 0.861 | 0.620 | 0.863 | 0.660 |
| | QuantumNAT | 0.842 ($\downarrow 0.027$) | 0.682 ($\uparrow 0.043$) | 0.832 ($\downarrow 0.027$) | 0.690 ($\uparrow 0.018$) | 0.832 ($\downarrow 0.029$) | 0.678 ($\uparrow 0.058$) | 0.850 ($\downarrow 0.013$) | 0.736 ($\uparrow 0.076$) |
| | Ours | 0.862 ($\downarrow 0.007$) | 0.756 ($\uparrow 0.117$) | 0.853 ($\downarrow 0.006$) | 0.758 ($\uparrow 0.086$) | 0.859 ($\downarrow 0.002$) | 0.743 ($\uparrow 0.123$) | 0.851 ($\downarrow 0.012$) | 0.792 ($\uparrow 0.132$) |

TABLE IV
CIRCUIT COST OF VIRTUAL CIRCUITS AND TRANSPILE PHYSICAL CIRCUITS ON FOUR REAL *IBMQ* DEVICES.

| Circuit | Virtual Circuit | | | <i>IBMQ</i> Quito | | | <i>IBMQ</i> Lima | | | <i>IBMQ</i> Belem | | | <i>IBMQ</i> Manila | | |
|--------------|-----------------|------|-------|-------------------|------|-------|------------------|------|-------|-------------------|------|-------|--------------------|------|-------|
| | #1-q | #2-q | depth | #1-q | #2-q | depth | #1-q | #2-q | depth | #1-q | #2-q | depth | #1-q | #2-q | depth |
| ‘RX+RZ+CRX’ | 32 | 24 | 30 | 253 | 96 | 232 | 242 | 84 | 234 | 253 | 96 | 232 | 259 | 114 | 254 |
| ‘RY+RZ+CNOT’ | 24 | 6 | 21 | 48 | 6 | 21 | 48 | 6 | 21 | 48 | 6 | 21 | 48 | 6 | 21 |
| ‘RY+CRX’ | 16 | 16 | 18 | 191 | 80 | 167 | 190 | 68 | 181 | 191 | 80 | 167 | 191 | 80 | 167 |

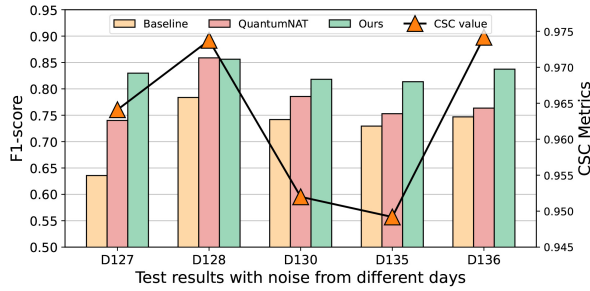


Fig. 11. Test on four small noise scenarios on *IBMQ* Quito, where QuantumNAT is trained on D128 (CSC value close to D136). The performance of baseline and QuantumNAT fluctuates significantly, whereas ours remains remarkably stable.

2) *Evaluate Small Noise Magnitudes*: To evaluate the performance under small noise scenarios, we selected four days with minimal noise magnitudes (CSC: 0.949–0.974) for the MNIST-4 classification task on *IBMQ* Quito, where QuantumNAT is trained on D128 (CSC \approx 0.9737). The results are shown in Fig. 11, QuantumNAT achieved its best-F1 score of 0.859 on training day D128. However, when tested on D136, which is of a similar noise level (CSC \approx 0.9742), the F1-score dropped to 0.764, a decrease of 9.52%. In contrast, our method showed only a 1.89% difference in performance between D128 and D136. This indicates that even in the presence of small noise, QuantumNAT’s training method cannot maintain high performance across different noise conditions. Specifically, a model trained on D128 can adapt to the small noise of D128 but cannot adapt to the small noise of D136. Conversely, our multiscale training method ensures adaptability to different noise distributions, consistently maintaining good performance across the four small noise test days. This consistency underscores the original intent behind the design of QuST. We also perform evaluations on

TABLE V
TEST MODEL PERFORMANCE IN REAL QUANTUM MACHINES

| Dataset | Test Model | Quito | Lima | Belem | Manila |
|-----------|------------|--------------|--------------|--------------|--------------|
| MNIST-4 | Baseline | 0.410 | 0.550 | 0.520 | 0.451 |
| | QuantumNAT | 0.636 | 0.682 | 0.553 | 0.615 |
| | Ours | 0.713 | 0.747 | 0.702 | 0.711 |
| Fashion-4 | Baseline | 0.503 | 0.567 | 0.433 | 0.531 |
| | QuantumNAT | 0.652 | 0.698 | 0.656 | 0.671 |
| | Ours | 0.702 | 0.744 | 0.679 | 0.685 |
| MNIST-2 | Baseline | 0.654 | 0.731 | 0.634 | 0.702 |
| | QuantumNAT | 0.822 | 0.855 | 0.810 | 0.847 |
| | Ours | 0.896 | 0.911 | 0.862 | 0.865 |
| Fashion-2 | Baseline | 0.655 | 0.752 | 0.633 | 0.732 |
| | QuantumNAT | 0.763 | 0.862 | 0.760 | 0.823 |
| | Ours | 0.879 | 0.906 | 0.836 | 0.855 |

different noise magnitudes, the results can be found in <https://github.com/JanusQ/QuST>.

E. Evaluation on Real-World Quantum Devices

To demonstrate the effectiveness and deployability on real QCs, we conduct a case study using the “RX+RZ+CRX” circuit to evaluate the classification accuracy of MNIST-2 and 4 and Fashion-2 and 4 on four *IBMQ* devices, including *IBMQ* Quito, Lima, Belem, and Lima. QuantumNAT is the average of QNAT M2–M4, as shown in Table V. Moreover, we calculate the circuit cost from the virtual circuit to the transpiled physical circuit, and the result in Table IV. The columns #1-q, #2-q, and depth represent the number of single-qubit gates, the number of two-qubit gates, and the depth of the circuit, respectively. In Table V, the average test accuracy of the baseline model on the four classification tasks (MNIST-4, Fashion-4, MNIST-2, Fashion-2) is 0.483, 0.509, 0.680, and 0.693, respectively. QuantumNAT is 0.622 ($\uparrow 13.9\%$), 0.669 ($\uparrow 16.1\%$), 0.833 ($\uparrow 15.3\%$), 0.802 ($\uparrow 10.9\%$) compared with baseline, QuST achieves 0.718 ($\uparrow 23.6\%$),

TABLE VI

TEST MODEL PERFORMANCE ON MNIST-4 IN LATEST REAL QUANTUM MACHINES, QNAT (OLD) MEANS QUANTUMNAT TRAIN ON IBMQ QUITO, QNAT (RE) MEANS RETRAIN ON THE CORRESPONDING MACHINE, OUR METHOD IS TRAINED ON IBMQ QUITO

| Test Time | Sherbrooke | | | | Brisbane | | | | Kyoto | | | |
|-----------|------------|-----------|----------|--------------|----------|-----------|----------|--------------|----------|-----------|--------------|--------------|
| | Baseline | QNAT(old) | QNAT(re) | Ours | Baseline | QNAT(old) | QNAT(re) | Ours | Baseline | QNAT(old) | QNAT(re) | Ours |
| 2024/6/15 | 0.487 | 0.523 | 0.733 | 0.812 | 0.493 | 0.522 | 0.687 | 0.702 | 0.506 | 0.542 | 0.754 | 0.749 |
| 2024/6/23 | 0.501 | 0.568 | 0.751 | 0.827 | 0.543 | 0.573 | 0.757 | 0.813 | 0.473 | 0.508 | 0.688 | 0.693 |

TABLE VII

ABLATION EXPERIMENTS ON CSC-WEIGHTED AND LOSS INTEGRATION USING THE “RX+RZ+CRX” CIRCUIT IN MNIST-4

| Weight Coefficients θ | 0 | 1 | CSC | 10 | 100 |
|------------------------------|-------|-------|-------|-------|-------|
| “RX+RZ+CRX” | 0.723 | 0.778 | 0.805 | 0.768 | 0.741 |

0.703 ($\uparrow 19.4\%$), 0.883 ($\uparrow 20.3\%$), 0.869 ($\uparrow 17.6\%$) compared to baseline, and 9.7%, 3.3%, 5%, and 6.7% higher than QuantumNAT, respectively. The test performance ranking within the same training category is *Lima* > *Manila* > *Quito* > *Belem*. This ranking aligns with the noise magnitudes, quantum chip topology, and other characteristics of actual QCs. Overall, our method consistently delivers the highest performance on real QCs, maintaining a noticeable performance advantage over the other two training methods.

In 2024, IBMQ unveiled new QCs boasting 127–1121 qubits, offering enhanced performance due to increased qubit count and reduced noise levels. To assess the effectiveness of QuST on these latest machines, we conducted experiments on IBM Sherbrooke, IBM Brisbane, and IBM Kyoto. We trained the model on IBMQ Quito and evaluated its performance on the three new machines. QNAT (old) refers to directly testing a QuantumNAT model trained on Quito, while QNAT (re) involves retraining the model on the target machine before testing. The results are summarized in Table VI. Thanks to our robust training methodology, leveraging weighted noise training to achieve robustness across time and variations in noise profiles allowed our method to maintain superior performance even on the different latest machines. For comparison, QuantumNAT just considers single-day noise on the single quantum machine, which demonstrates a significant performance gap (up to 21.2% on IBM Kyoto) between direct deployment and retraining. These findings further highlight the importance of training noise-resistant models like QuST.

F. Ablation Studies

We conduct ablation experiments on CSC, loss integration, and the training perception periods of noises. To verify the validity of key components, we implement the MNIST-4 classification task on IBMQ Quito using the “RX+RZ+CRX” circuit, and subsequent settings are the same if not specifically stated.

CSC and Loss Integration: We conduct ablation studies for CSC-weighted loss integration in MNIST-4 classification, using the “RX+RZ+CRX” circuit and setting the noise factor to $F = 1$. We consider four weight coefficient θ settings for loss integration: $\theta = \{0, 1, \text{CSC}, 10, 100\}$. Since CSC is incorporated into the loss function as a training weight

coefficient, wherein $\theta = 0$ represents without loss integration, $\theta = \{1, \text{CSC}, 10, 100\}$ is used to analyze the effect of different scales CSC-weighted loss integration values on the model’s accuracy. The results are shown in Table VII. The experimental results demonstrate that the introduction of the loss integration component leads to an improvement of the QNN accuracy from 1.8% to 8.2%. Different weight coefficients assigned to the loss function result in various degrees of performance enhancement. The accuracy improvements corresponding to $\theta = \{1, \text{CSC}, 10, 100\}$ are 5.5%, 8.2%, 4.5%, and 1.8%, respectively. Notably, our proposed CSC dynamic evaluation weight achieves the highest-accuracy improvement of 8.2%. In summary, loss integration in the presence of noise contributes to improving the model’s resilience to noise. Moreover, customizing different training weight coefficients for distinct noise environments facilitates better adaptation to varying noise magnitudes, thus enhancing the overall noise resistance of the model.

Multiscale Noise-Aware Training Perception Periods: For the multiscale noise-aware training, we conduct the experiments with five groups sampling periods of noise subsets, including one week, two weeks, one month, whole datasets (190 days), and whole filtered (filtered outliers CSC < 0.5 of whole datasets, remaining 180 days.). Specifically, the first three groups are sampled from 190 days. We evaluate the model performance on the MNIST-4 classification task with the “RX+RZ+CRX” circuit over 190 days and noise factor $F = 1$. Fig. 12 shows the accuracy and standard deviation with different settings. Based on our observations, we summarize three key points.

- 1) Introducing noise outliers can damage the model’s optimal accuracy. For example, training with noise outliers decreases optimal performance (e.g., 190 days perform worse than one month). In contrast, overall performance improves with filtering outliers (filtering CSC < 0.5), which prevents excessive noise from overshooting the model accuracy.
- 2) As noise accumulates, increasing the noise perception range is helpful to improve the model’s robustness. According to the tested noise types, there is little difference between no noise and read noise, as read errors tend to be more stable. Gate and read and gate significantly affect noise perception due to greater noise accumulation, achieving the best performance after filtering noise outliers.
- 3) There is a tradeoff between the sampling period and model convergence time. Shorter sampling periods result in faster convergence while extending the model’s noise perception range, which enhances robustness but requires more training time. Users can tailor the training strategy to suit their specific needs.

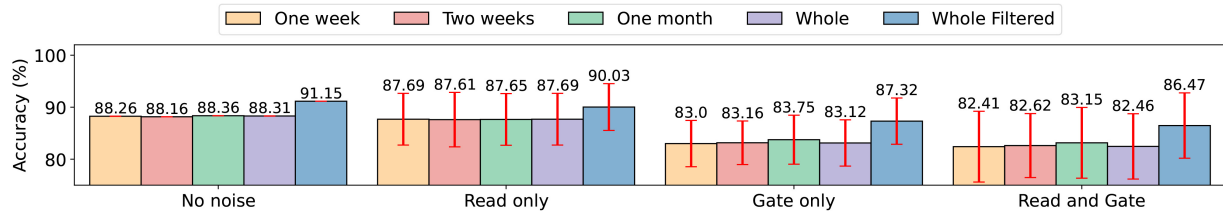


Fig. 12. Ablation on noise sample period, the whole filtered means filtered outliers (CSC < 0.5).

VIII. DISCUSSION

The Practicability of QuST: QuST is a general training method, it can be combined with existing error mitigation methods, such as ZNE [10], PEC [8], QECC [7], [30], and so on. Taking QECC as an example, the QECC reduces the probability of calculation errors by introducing additional qubits during the encoding process, detecting and repairing some errors during decoding. The QNN model perceives noise and mitigates errors in advance, which helps it exhibit better performance.

Extensibility of QuST: QuST is a robust training framework that is scalable and efficient, not limited by factors, such as qubits numbers or quantum implementation technology. It can also be integrated with optimized circuit compilation techniques to reduce circuit depth, improve QNN model accuracy, and so on. Furthermore, while this article focuses on QNN models, this approach can be extended to other quantum machine learning methods in the future and potentially applied to various research areas within quantum computing.

Limitations and Future Works: Our work has some limitations as follows.

- 1) Spatial bias in quantum systems can arise from imperfect hardware, environmental interactions, and other factors, often complex interactions. We focus on investigating the effect of noise using high-level noise models across different quantum devices, and future studies can delve deeper into spatial bias.
- 2) Regarding temporal bias, this article investigates the effects of multiple noise sampling periods. Future investigations can explore combined training schemes of different noise scales.
- 3) Develop adaptive optimization techniques, such as automatically optimizing the QNN circuit structure according to the characteristics of the quantum device (T1 (thermal relaxation time), T2 (dephasing time), and chip topology), and automatically searching for the optimal circuit combined with our error mitigation technology.

In the future, we will conduct comprehensive research on various noise sources, carefully analyze the intrinsic characteristics of distinct noise, and subsequently enhance the model's noise-resilient ability to adapt to diverse noise scenarios.

IX. CONCLUSION

In this article, we propose a noise-resilient QNN model training framework capable of adapting to temporal and spatial noise biases at multiple magnitudes without retraining. We introduce CSC as a metric for evaluating circuit reliability in

noisy environments. By utilizing CSC as a training weight, we integrate different noise configurations into the QNN model's loss function, and we employ multiscale noise-aware training techniques to expand the model's noise tolerance. Consequently, our experimental results demonstrate that our proposed noise-resilient training framework, QuST, maintains stable performance and enhances noise resilience, effectively mitigating noise biases across temporal and spatial dimensions.

REFERENCES

- [1] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, "Quantum computational chemistry," *Rev. Mod. Phys.*, vol. 92, no. 1, Mar. 2020, Art. no. 015003.
- [2] J. Shi, R. X. Zhao, W. Wang, S. Zhang, and X. Li, "QSAN: A near-term achievable quantum self-attention network," 2022, *arXiv:2207.07563*.
- [3] K. Huang et al., "Quantum generative adversarial networks with multiple superconducting qubits," *Nat. NPJ Quant. Inf.*, vol. 7, p. 165, Dec. 2021.
- [4] H. Wang et al., "QuantumNAT: Quantum noise-aware training with noise injection, quantization and normalization," in *Proc. DAC*, 2022, pp. 1–6.
- [5] J. Li, M. Alam, and S. Ghosh, "Large-scale quantum approximate optimization via divide-and-conquer," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 42, no. 6, pp. 1852–1860, Jun. 2023.
- [6] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.
- [7] J. Roffe, "Quantum error correction: An introductory guide," *Contemp. Phys.*, vol. 60, no. 3, pp. 226–245, 2019.
- [8] K. Temme, S. Bravyi, and J. M. Gambetta, "Error mitigation for short-depth quantum circuits," *Phys. Rev. Lett.*, vol. 119, no. 18, 2017, Art. no. 180509.
- [9] Y. Li and S. C. Benjamin, "Efficient variational quantum simulator incorporating active error minimization," *Phys. Rev. X*, vol. 7, no. 2, 2017, Art. no. 021050.
- [10] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, "Digital zero noise extrapolation for quantum error mitigation," in *Proc. IEEE Int. Conf. Quant. Comput. Eng. (QCE)*, 2020, pp. 306–316.
- [11] Y. Kim et al., "Evidence for the utility of quantum computing before fault tolerance," *Nature*, vol. 618, pp. 500–505, Jun. 2023.
- [12] P. Murali, D. C. McKay, M. Martonosi, and A. Javadi-Abhari, "Software mitigation of crosstalk on noisy intermediate-scale quantum computers," in *Proc. ASPLOS*, 2020, pp. 1–15.
- [13] W. Jiang et al., "A co-design framework of neural networks and quantum circuits towards quantum advantage," *Nat. Commun.*, vol. 12, p. 579, Jan. 2021.
- [14] S. Stein et al., "EQC: Ensembled quantum computing for variational quantum algorithms," in *Proc. 49th ISCA*, 2022, pp. 59–71.
- [15] Z. Hu, Y. Lin, Q. Guan, and W. Jiang, "Battle against fluctuating quantum noise: Compression-aided framework to enable robust quantum neural network," in *Proc. 60th ACM/IEEE Design Autom. Conf. (DAC)*, 2023, pp. 1–6.
- [16] J. Paykin et al., "PCOAST: A pauli-based quantum circuit optimization framework," 2023, *arXiv:2305.10966*.
- [17] S. Ruan, Y. Wang, W. Jiang, Y. Mao, and Q. Guan, "VACSSEN: A visualization approach for noise awareness in quantum computing," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 1, pp. 462–472, Jan. 2023.
- [18] S. S. Tannu et al., "Not all qubits are created equal: A case for variability-aware policies for NISQ-era quantum computers," in *Proc. ASPLOS*, 2019, pp. 987–999.
- [19] A. Brahmé, *Comprehensive Biomedical Physics*. Boston, MA, USA: Newnes, 2014.

- [20] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *J. Mach. Learn. Res.*, vol. 10, no. 1, pp. 1–40, 2009.
- [21] "IBM Qiskit." 2022. [Online]. Available: <https://qiskit.org/>
- [22] S. Kak, "On quantum neural computing," *Inf. Sci.*, vol. 83, no. 3, pp. 143–160, 1995.
- [23] H. Wang, Z. Li, J. Gu, Y. Ding, D. Z. Pan, and S. Han, "QOC: Quantum on-chip training with parameter shift and gradient pruning," in *Proc. DAC*, 2022, pp. 655–660.
- [24] T. Li, Z. Zhao, and J. Yin, "Task-driven quantum device fingerprint identification via modeling QNN outcome shift induced by quantum noise," in *Proc. Compan., Proc. ACM Web Conf.*, 2024, pp. 557–560.
- [25] H. P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems*. Oxford, U.K.: Oxford Univ. Press, 2002.
- [26] T. Patel, A. Potharaju, B. Li, R. B. Roy, and D. Tiwari, "Experimental evaluation of NISQ quantum computers: Error measurement, characterization, and implications," in *Proc. Int. Conf. High Perf. Comput., Netw., Stor. Anal.*, 2020, pp. 1–15.
- [27] K. Georgopoulos, C. Emary, and P. Zuliani, "Modeling and simulating the noisy behavior of near-term quantum computers," *Phys. Rev. A*, vol. 104, no. 6, 2021, Art. no. 062432.
- [28] T. Wu and J. Guo, "A multiscale simulation approach for germanium-hole-based quantum processor," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 1, pp. 257–265, Jan. 2023.
- [29] R. Wille, L. Burgholzer, and A. Zulehner, "Mapping quantum circuits to IBM QX architectures using the minimal number of SWAP and H operations," in *Proc. 50th DAC*, 2019, pp. 1–6.
- [30] W.-H. Tseng, C.-H. Hsu, W.-H. Lin, and Y.-W. Chang, "A bridge-based compression algorithm for topological quantum circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 12, pp. 5582–5595, Dec. 2022.
- [31] T. Grurl, J. FuSS, and R. Wille, "Noise-aware quantum circuit simulation with decision diagrams," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 3, pp. 860–873, Mar. 2023.
- [32] G. Li, Y. Ding, and Y. Xie, "Tackling the qubit mapping problem for NISQ-era quantum devices," in *Proc. ASPLOS*, 2019, pp. 1001–1014.
- [33] K.-Y. Chang and C.-Y. Lee, "Mapping nearest neighbor compliant quantum circuits onto a 2-D hexagonal architecture," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 10, pp. 3373–3386, Oct. 2022.
- [34] L. Lao, H. van Someren, I. Ashraf, and C. G. Almudever, "Timing and resource-aware mapping of quantum circuits to superconducting processors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 2, pp. 359–371, Feb. 2022.
- [35] A. Deb, G. W. Dueck, and R. Wille, "Exploring the potential benefits of alternative quantum computing architectures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 9, pp. 1825–1835, Sep. 2021.
- [36] X. Zhou, Y. Feng, and S. Li, "Supervised learning enhanced quantum circuit transformation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 2, pp. 437–447, Feb. 2023.
- [37] T. Van Erven and P. Harremos, "Rényi divergence and Kullback–Leibler divergence," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 3797–3820, Jul. 2014.
- [38] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.
- [39] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
- [40] "IBMQ quantum," IBM. 2022. [Online]. Available: <https://quantum-computing.ibm.com/>



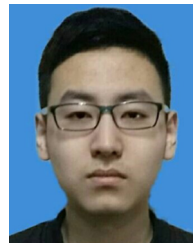
Tingting Li is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. She has published papers in conferences and journals, including HPCA, MICRO, WWW, and TIFS. Her research interests include quantum computing, quantum calibration, and quantum machine learning.



Liqiang Lu received the bachelor and Ph.D. degrees from Peking University, Beijing, China, in 2022 and 2017, respectively.

He is an Associate Professor (ZJU100 Young Professor) with the College of Computer Science, Zhejiang University, Hangzhou, China. He has authored 16 scientific publications in premier international journals and conferences in related domains, including ISCA, MICRO, HPCA, FCCM, DAC, *IEEE Micro*, and IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. His research interests include quantum computing, computer architecture, deep learning accelerators, and hardware–software codesign.

Dr. Lu also serves as a TPC Member for the premier conferences in the related domain, including ICCAD, FPT, and HPCC.



Ziming Zhao (Member, IEEE) is currently pursuing the Ph.D. degree with Zhejiang University, Hangzhou, China.

He has published more than 25 papers in international journals and conference proceedings, including SIGCOMM, WWW, INFOCOM, CCS, RTSS, TSE, TMC, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, ToN, TIFS, TDSC, COSE, ESE, and AAAI. His research interests include machine learning, traffic identification, privacy-preserving, and quantum machine learning.



Ziqi Tan is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China.

He has published papers in conferences and journals including *Nature Physics*, MICRO, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, WSDM, SIGKDD, and MM. His research interests mainly focus on quantum computing and machine learning.



Siwei Tan is currently pursuing the Ph.D. degree with the College of Computer Science, Zhejiang University, Hangzhou, China.

He has published papers in conferences and journals including ASPLOS, MICRO, HPCA, VIS, TVCG, and SCC. His research interests are quantum algorithms, computer architecture, and computer systems.



Jianwei Yin (Member, IEEE) received the Ph.D. degree in computer science from Zhejiang University (ZJU), Hangzhou, China, in 2001.

He was a Visiting Scholar with the Georgia Institute of Technology, Atlanta, GA, USA. He is currently a Full Professor with the College of Computer Science, ZJU. He has published more than 100 papers in top international journals and conferences. His current research interests include quantum computing, service computing, and business process management.

Dr. Yin is an Associate Editor of the IEEE TRANSACTIONS ON SERVICES COMPUTING.