Optimize Quantum Learning on Near-Term Noisy Quantum Computers

Presenter: Zhirui Hu

Email : zhu2@gmu.edu

George Mason University

Electrical and Computer Engineering

Introduction

Why Quantum Computing?

- **Applications**

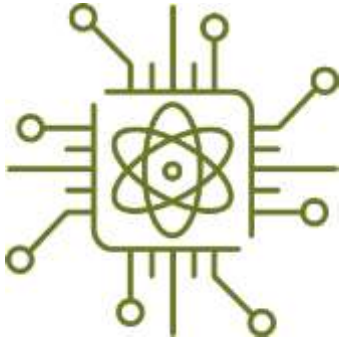
Molecular Simulation

Cybersecurity

Drug / Electronic discovery

Financial modeling / forecasting

Traffic optimization



- **Algorithms**

QML

QFT

QNN

Grover's

VQE

QAOA

Method	Speedup
Bayesian inference [24,25]	$O(\sqrt{N})$
Online perceptron [26]	$O(\sqrt{N})$
Least-squares fitting [27]	$O(\log N)$
Classical Boltzmann machine [28]	$O(\sqrt{N})$
Quantum Boltzmann machine [29,30]	$O(\log N)$
Quantum PCA [22]	$O(\log N)$
Quantum support vector machine [23]	$O(\log N)$
Quantum reinforcement learning [31]	$O(\sqrt{N})$

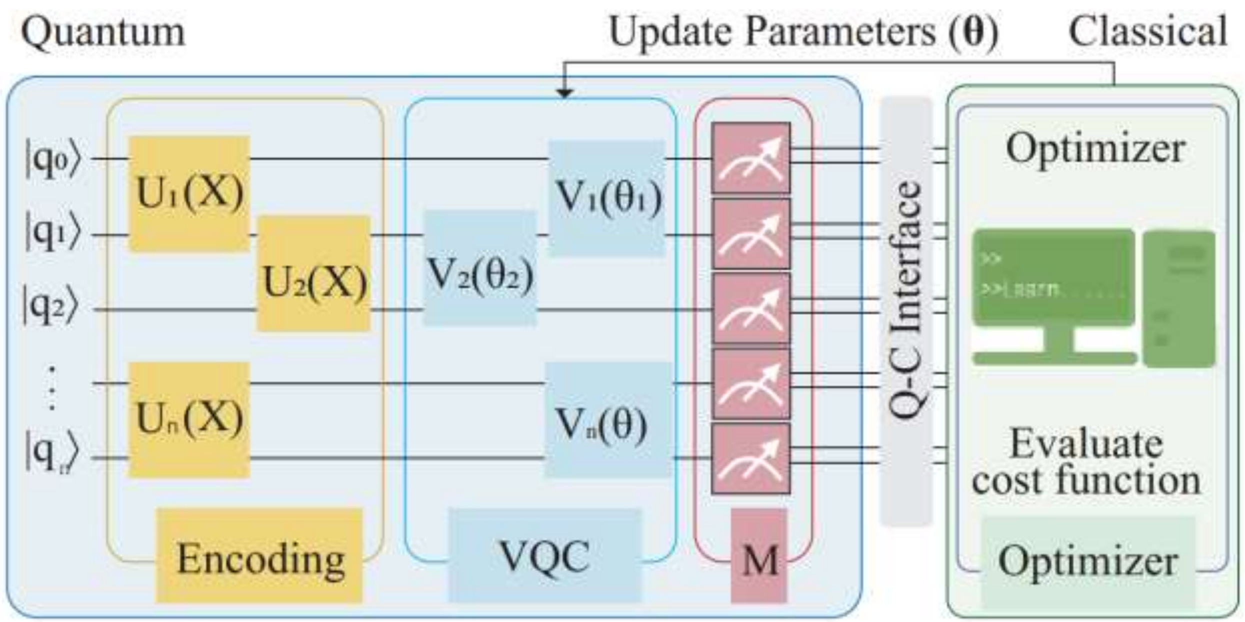
- **Advantages of quantum computing**

Parallelism

Entanglement

Background: Quantum Learning Basics &Property

Quantum Learning Basics



1. **Define a learning problem and the loss function**
2. **Select a learning model**
3. **Training a model**

Training on quantum computer :

- advantages:

 - potential on qubit scalability

- disadvantages:

 - difficult to optimize parameters
 - training time

Training on noisy simulator:

- advantages:

 - accurate gradient-based method

- disadvantages:

 - limitation on qubit scalability
 - difficult to simulate noise

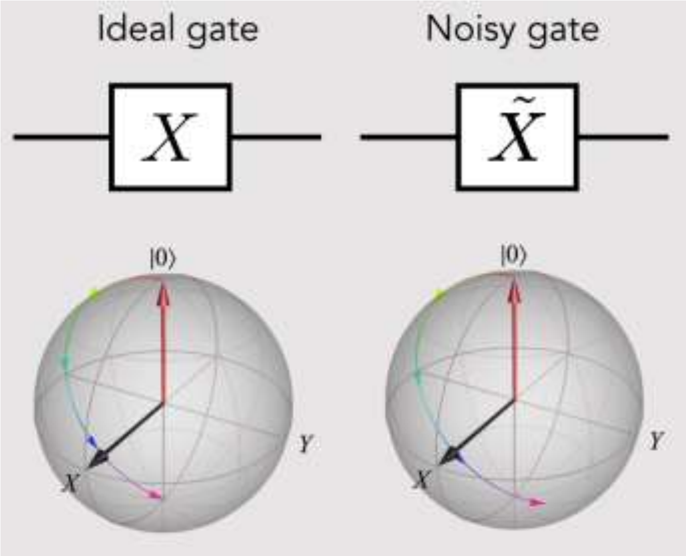
Background: Noise on Quantum Device

Take superconducting quantum computer as a case study

Quantum Noise Source

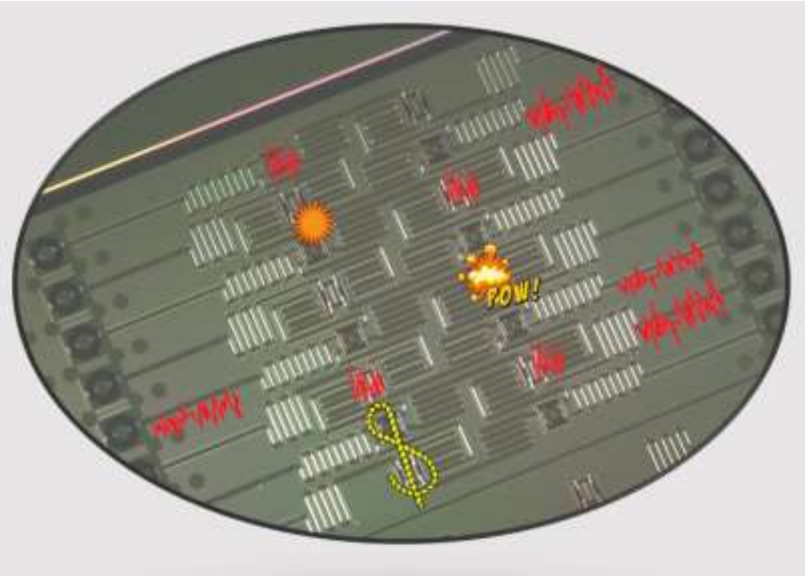
Source:

- **Control**



$$X = R_X(\pi)$$
$$\tilde{X} := R_X(\pi + \epsilon)$$

- **Environment**



- Charge noise
- Magnetic flux noise
- Crosstalk
- ...

ref Zlatko K. Minev, Introduction to Quantum Noise

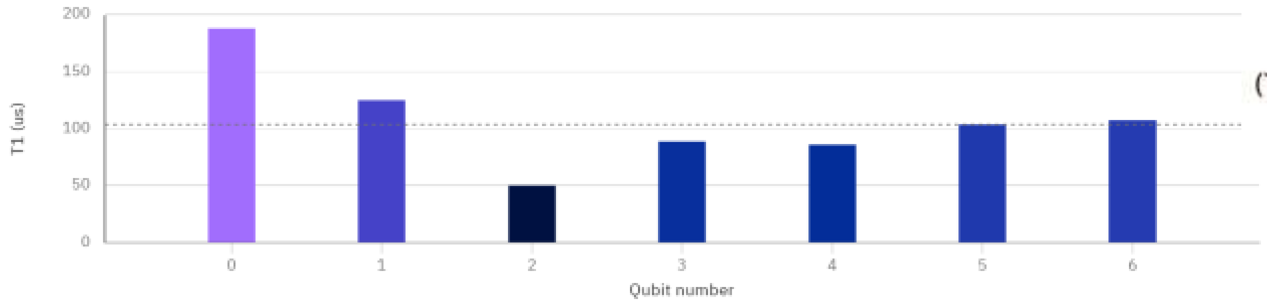
Quantum Noise Modeling

Characterization of noise sources and how they impact a given quantum system.

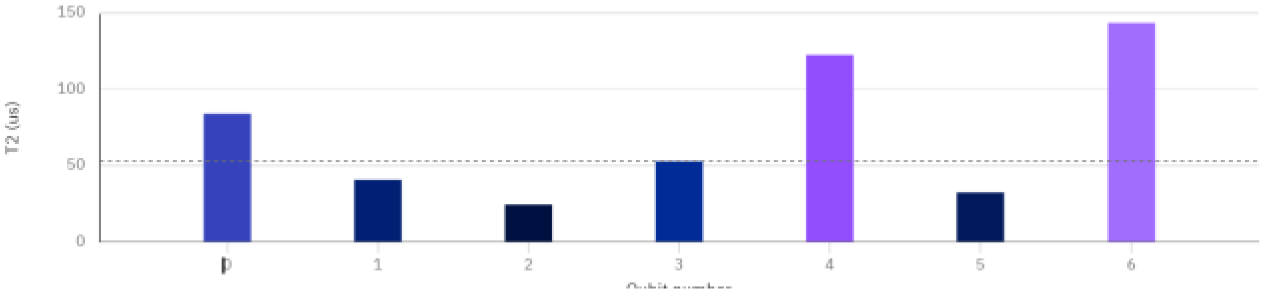
IBM, T1, T2 (Decoherent noise)

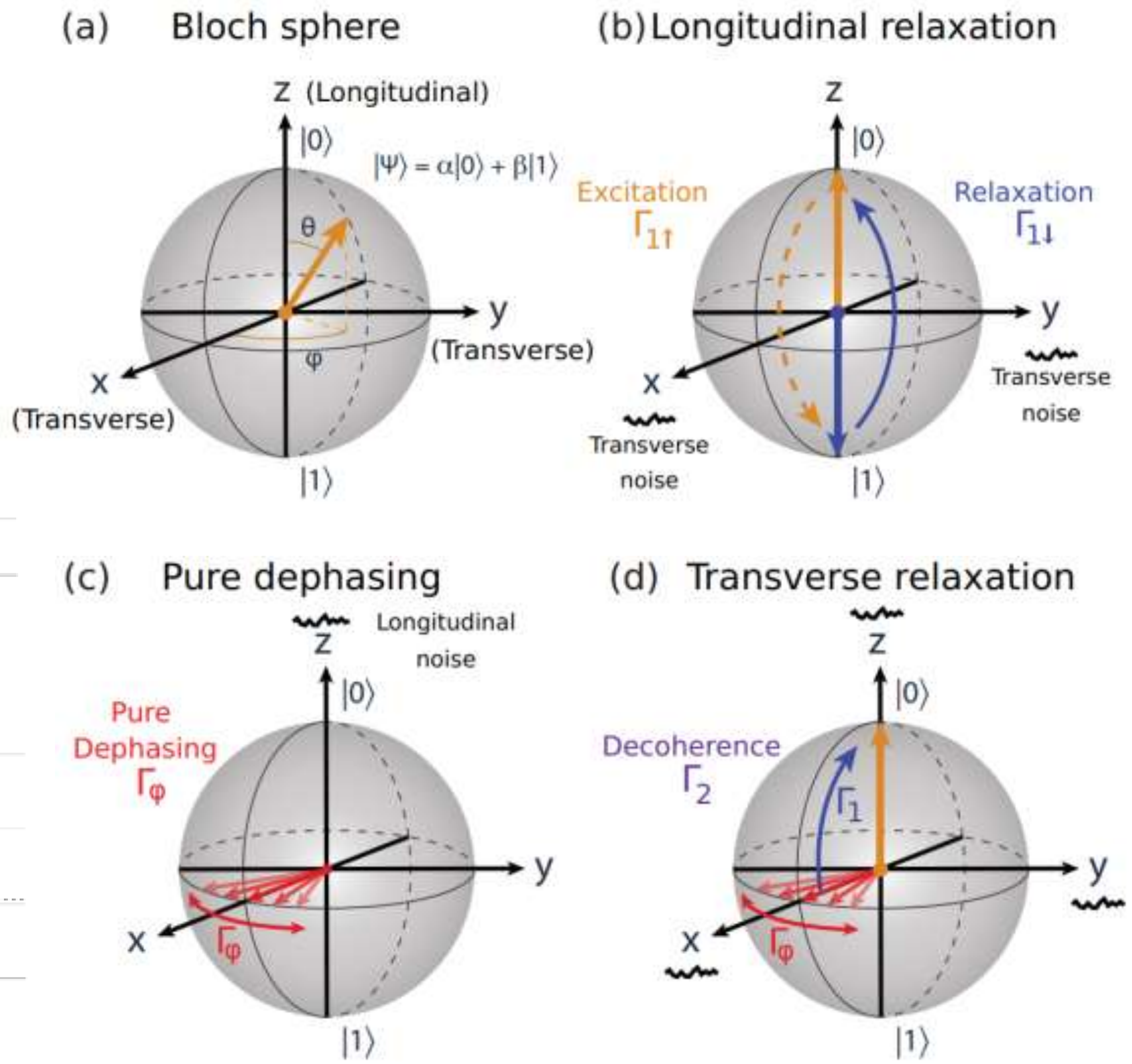
ibm_oslo OpenQASM 3

thermal relaxation time



dephasing time

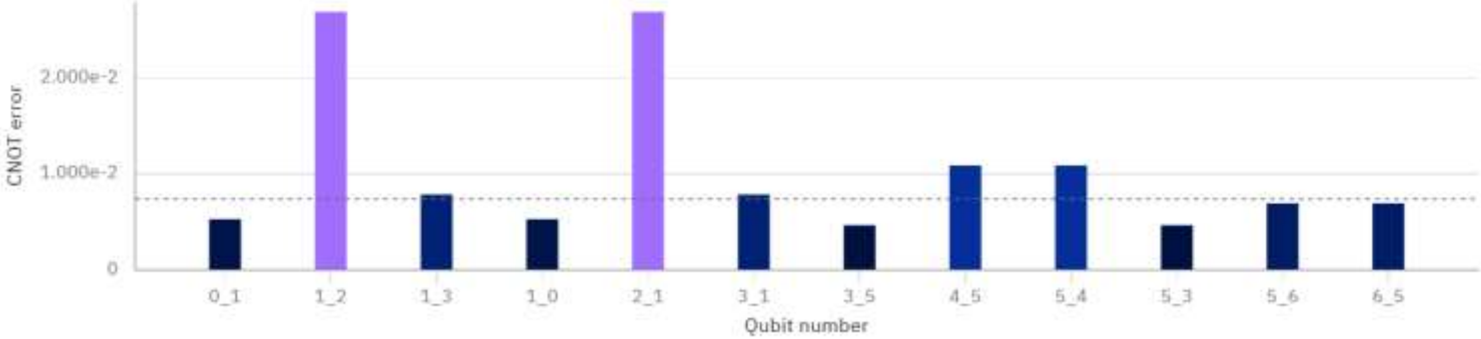


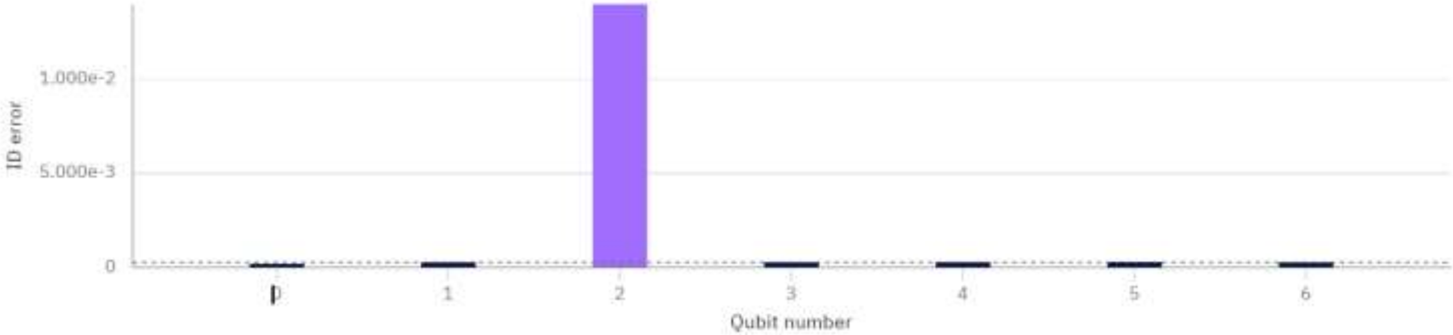


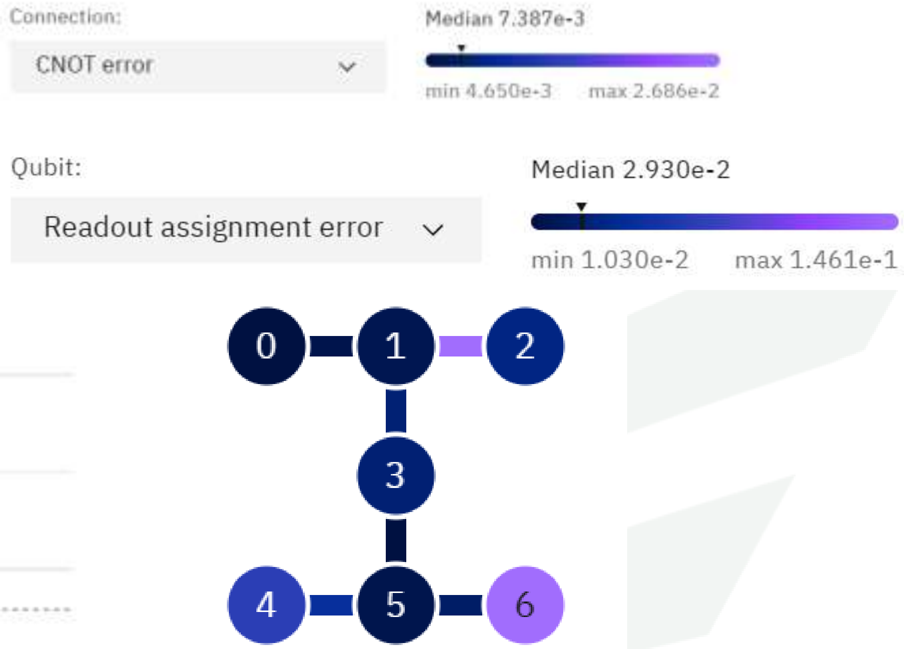
Quantum Noise Modeling

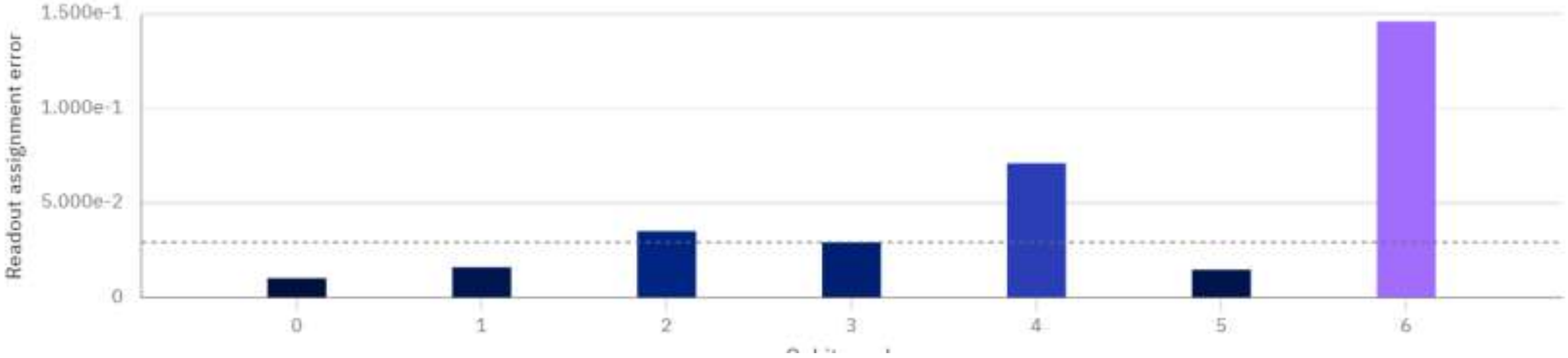
IBM, Gate error and readout error

ibm_oslo OpenQASM 3



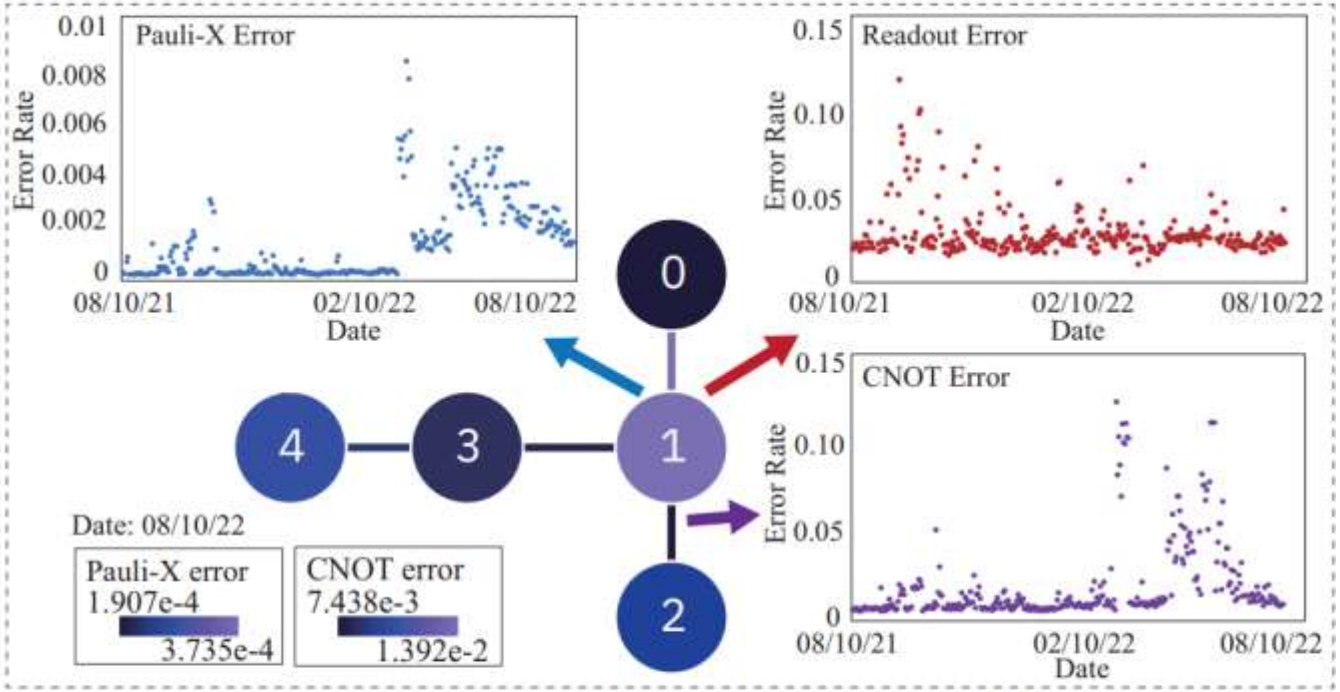






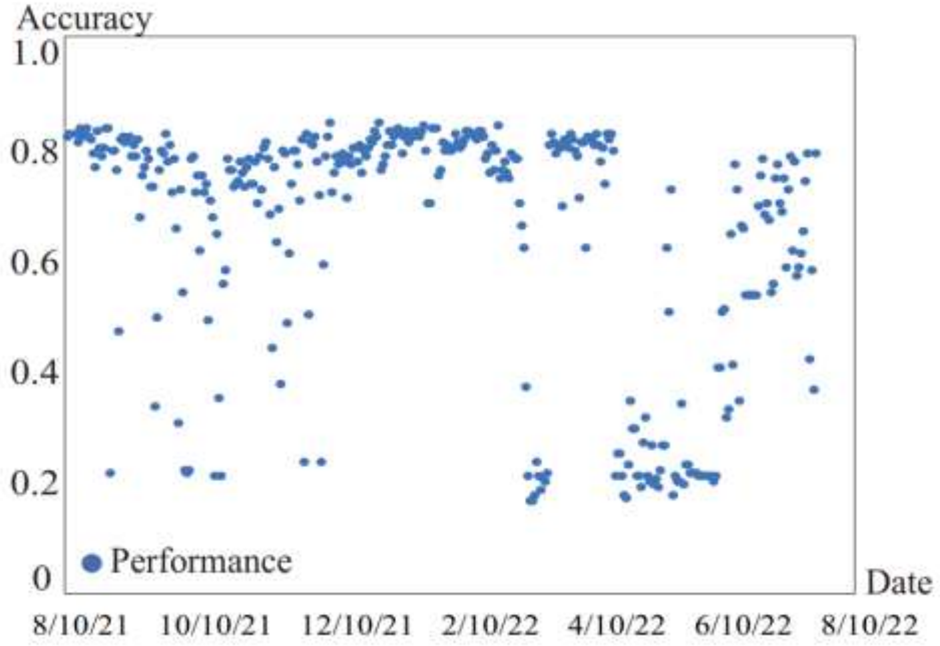
Fluctuating Quantum Noise

- **Fluctuating noise on quantum device**



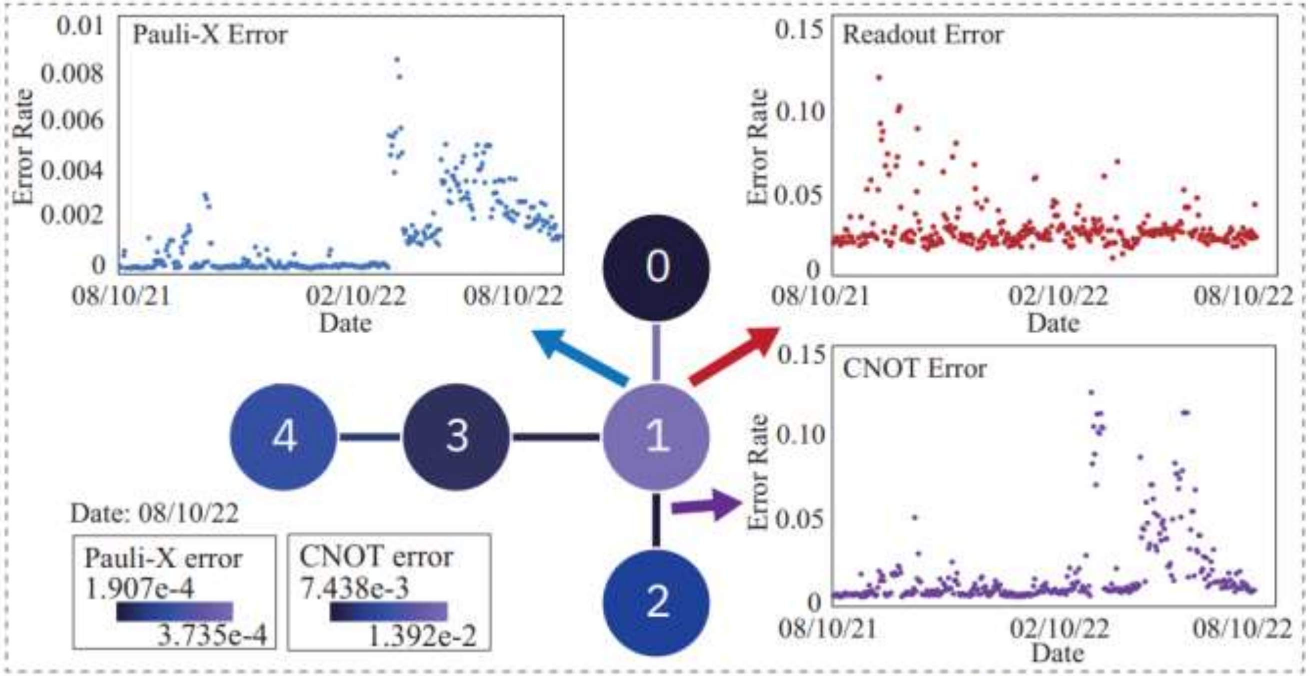
(a) Fluctuating quantum noise on real quantum computer (1-year long daily profiling)

- **Fluctuating accuracy**



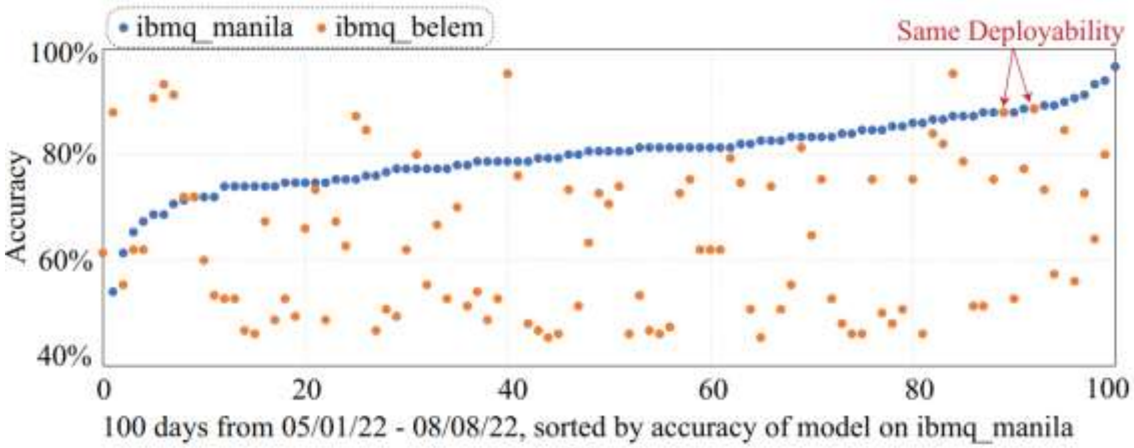
Quantum Noise heterogeneous

- **Temporal**



(a) Fluctuating quantum noise on real quantum computer (1-year long daily profiling)

- **Spatial**



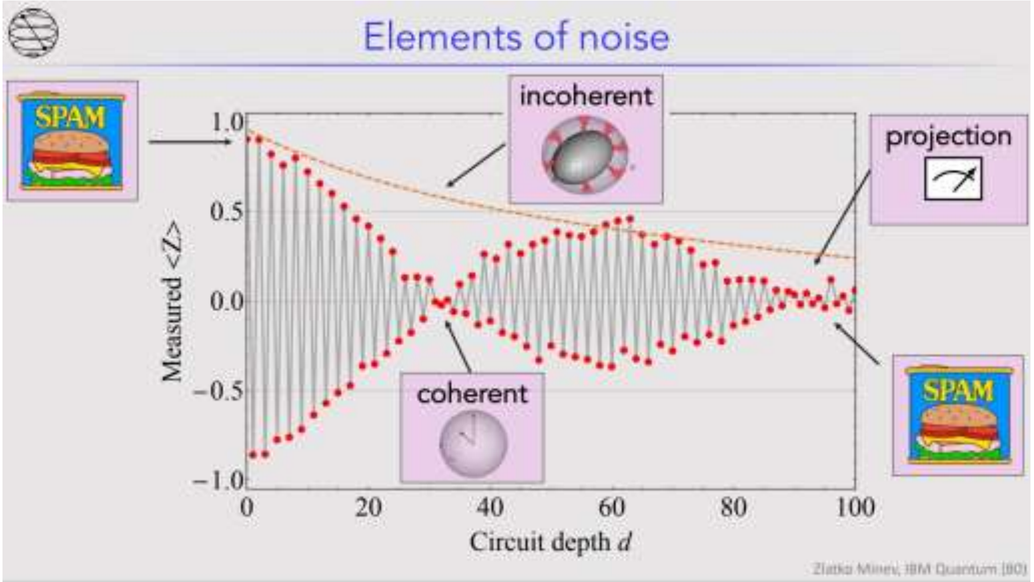
Quantum Neural Network Compression

Motivation

Why Compression in QNN?

From noise perspective:

- As the gates becomes more, the control error will be accumulated, and the result will be divergent.
- As time grows, the deconherent error will become severe.
- Even if noise can be learnt in the parameter, the noise is extremely random and varying, which will damage the accuracy.



Ref: Zlatko K. Minev, IBM Quantum .

Motivation

Why Compression in QNN?

From optimization perspective

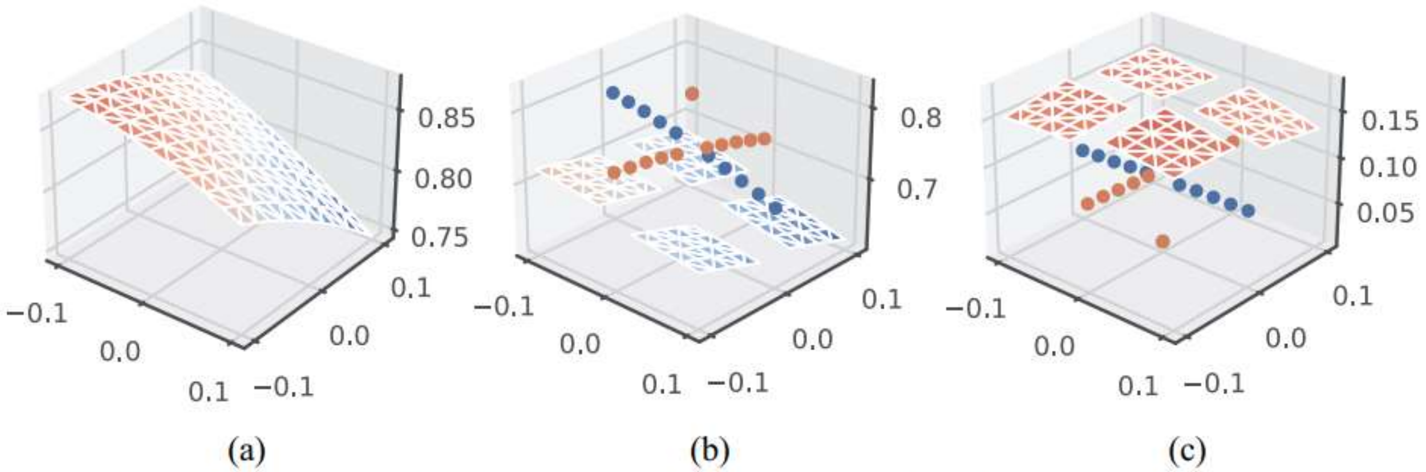
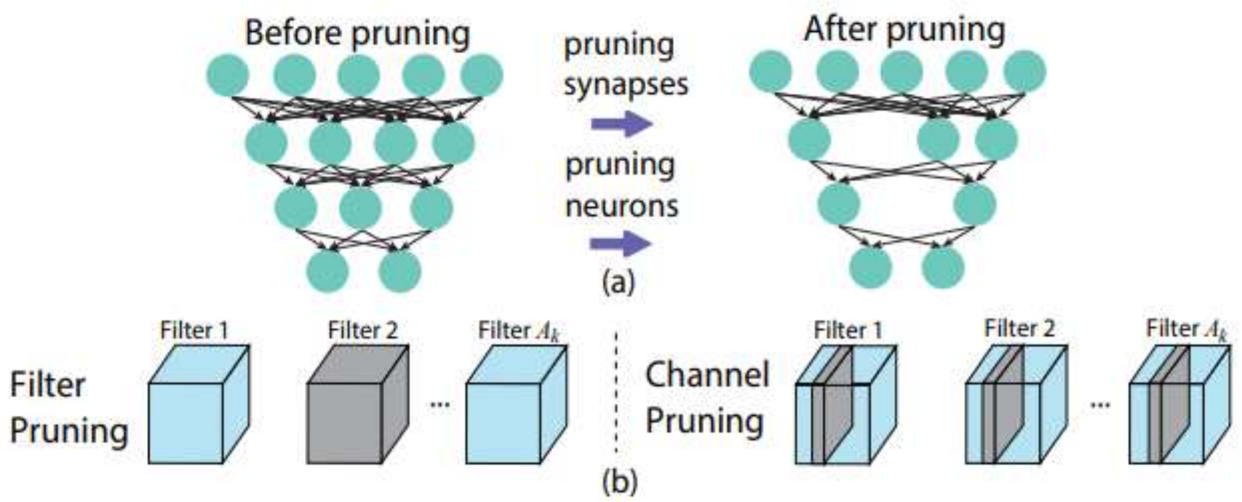


Fig. 3. Noise-aware training may miss optimal solution: (a) Optimization surface of 2-parameter VQC under noise free environment. (b) Optimization surface of the same VQC under a noisy environment. (c) Difference between (a) and (b).

Technique: Classical Pruning

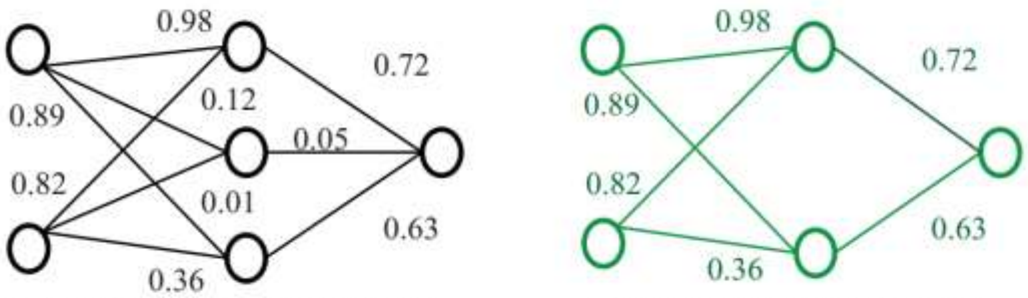


(a) Non-structured weight pruning and (b) two types of structured weight pruning.

ref: PatDNN: Achieving Real-Time DNN Execution on Mobile Devices with Pattern-based Weight Pruning

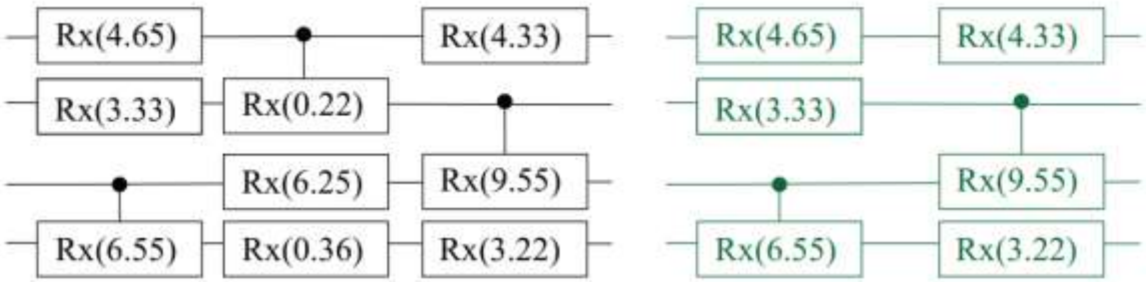
Technique: From Classical To Quantum

- Pruning in Classical ML



(a) Non-Compression Classical NN (b) Classical NN with Pruning

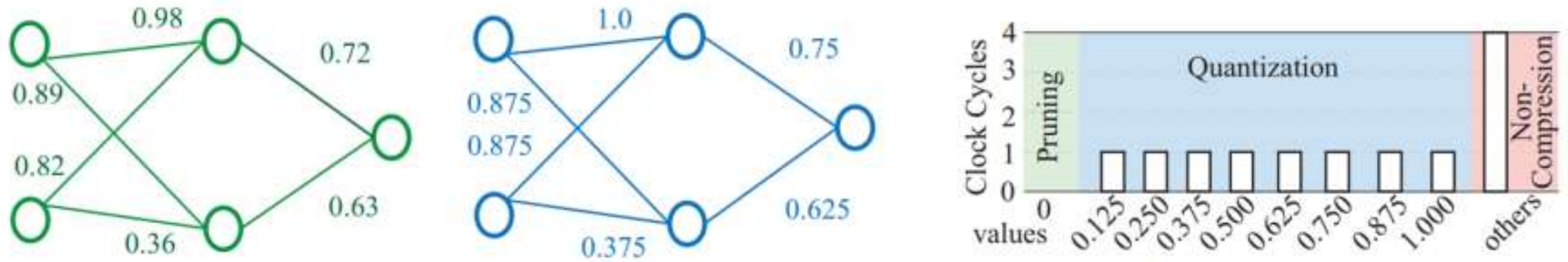
- Pruning in **Quantum ML**



(e) Non-Compression QNN (f) QNN with Pruning

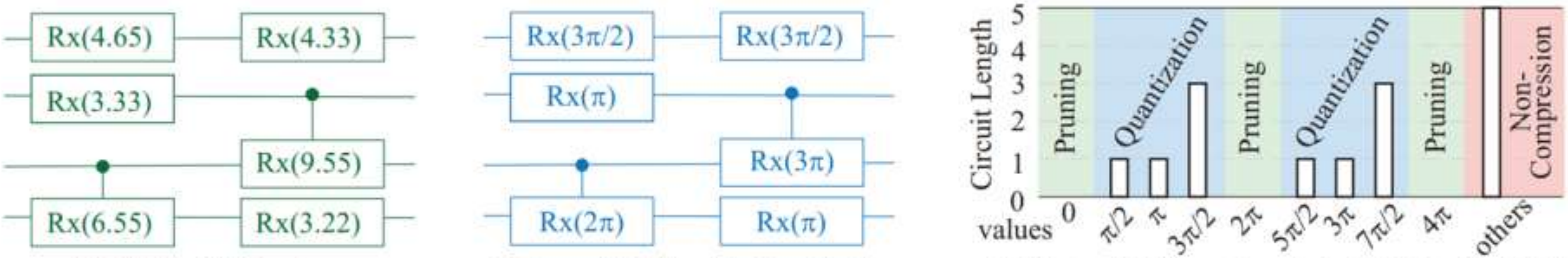
Technique: From Classical To Quantum

- Quantization in Classical ML



(b) Classical NN with Pruning (c) Pruned NN with Quantization (d) Cost of Different Levels in Classical NN

- Quantization in Quantum ML



(f) QNN with Pruning (g) Pruned QNN with Quantization (h) Cost of Different Levels in RX Gate in QNN

★

Compiler makes a difference!

Technique: LUT Construction

❑ Compression-Level Lookup Table (LUT)

A combination of pruning/quantization level called as “compression level”.

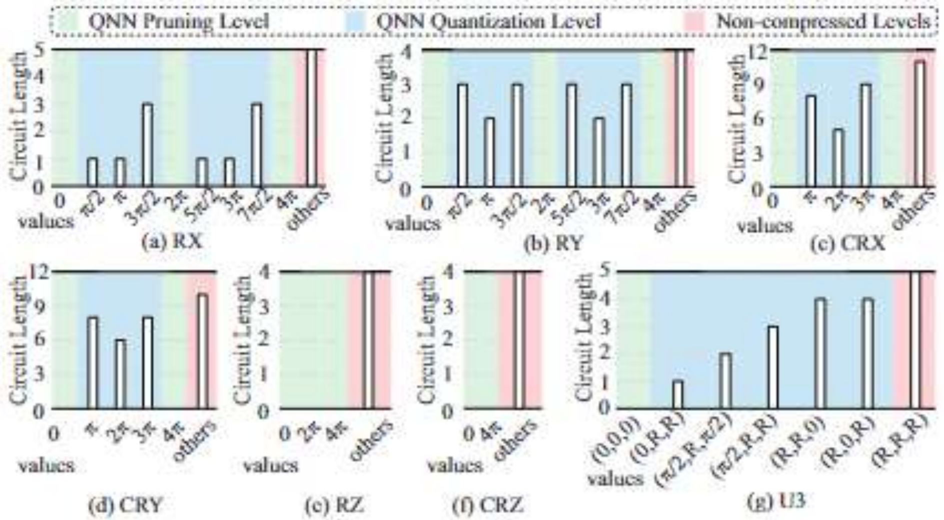


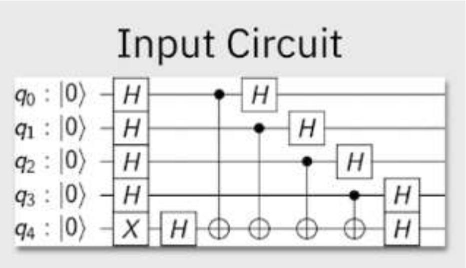
Table 1: circuit depth of compiled quantum gates on IBM quantum processors; parameters are in the range of $[0, 4\pi]$

Gate	0	π	2π	3π	4π	$\pi/2$	$3\pi/2$	$5\pi/2$	$7\pi/2$	others
RX	0	1	0	1	0	1	3	1	3	5
RY	0	2	0	2	0	3	3	3	3	4
CRX	0	8	5	9	0	11	11	11	11	11
CRY	0	8	6	8	0	10	10	10	10	10

- **Pruning:** Not only 0 can be pruned, but also 2π, 4π, etc.
- **Quantization:** Different quantization level may have different cost

How can we decide the compression level?

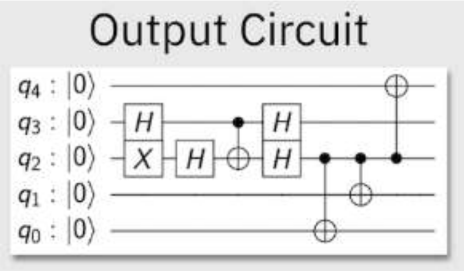
Technique: Compiler diversity



Logical Gate Opt. &Decomp.

Mapping & Routing

Physical Gate Decomp.& Opt.



basic gates for decomp.
SU(4), SU(2)
RX, RY, RZ, CRX, CRY, CRZ
U3, RXX, RYY, RZZ, RZX
....

basic gate son quantum device.
CNOT,ID,SX,X,RZ
Rx, Ry, and XX
......

Pulse assign & scheduling

Output pulse

Technique:

- **Two objective:**
  a.Maximize accuracy of classification
  b.Minimize circuit length

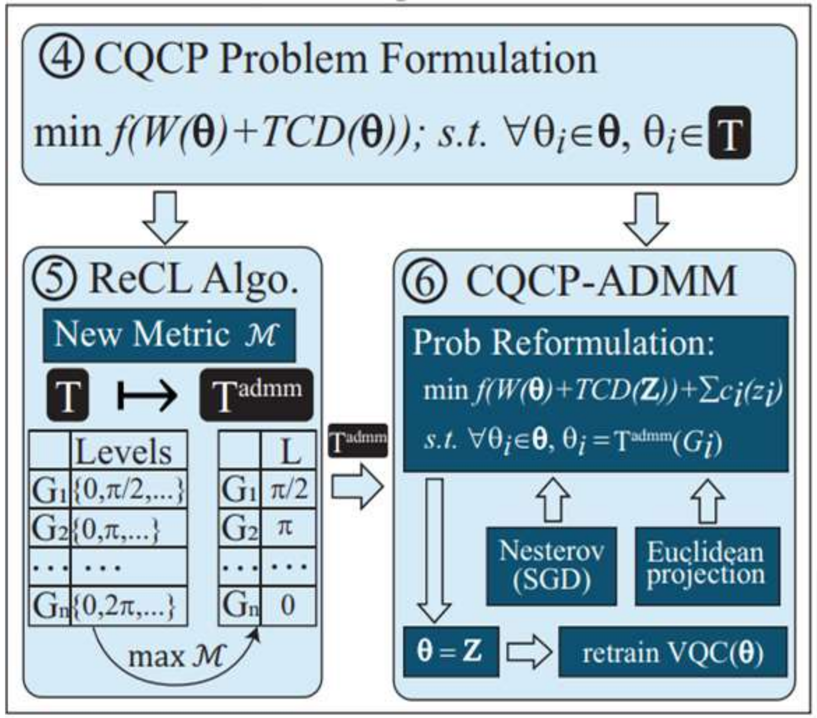
- **Quantum NN ----limited number of parameters**

- **Decide compression level for each parameters:**

  A heuristic metric: $\mathcal{M}(\theta, G_i(\gamma_{i,k})) = acc(W(\theta^{i,k})) \cdot \tau(\theta^{i,k}, \theta)$

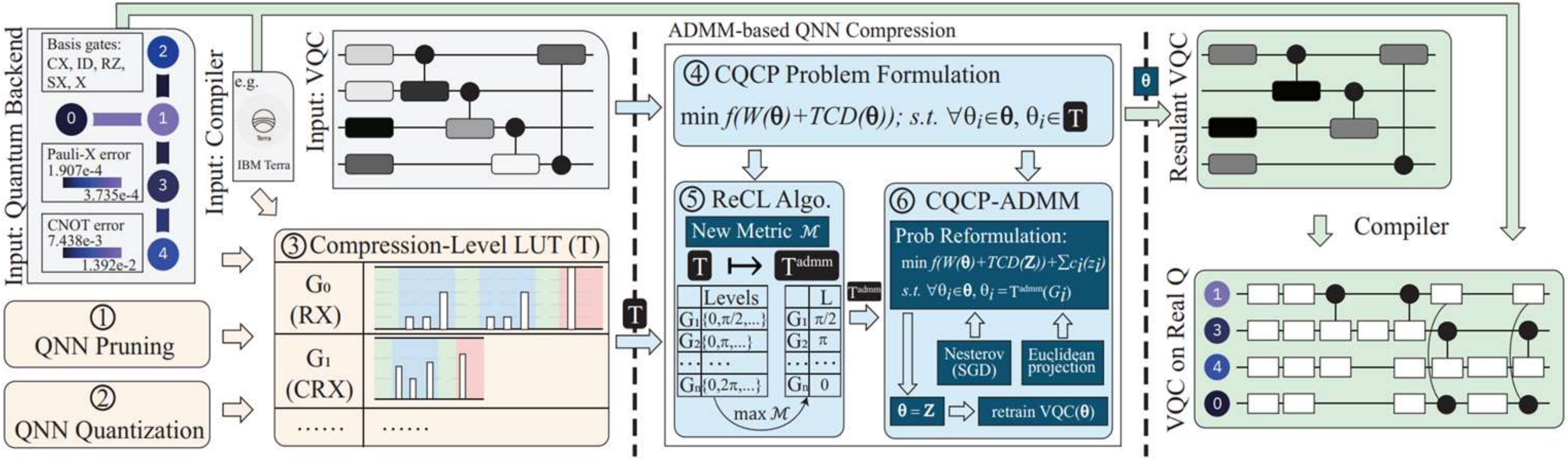
  Select the compression level of max matric

- **Leverage ADMM for two-objective optimization**



Technique: Admm-based framework

Three stages: 1. Preparation; 2. Compression; 3. Deployment



CompVQC Framework: Experiment Results

Results on Multiple IBM Quantum Computers

Datasets		Syn-Dataset-4		Syn-Dataset-16	
Compression Method		Acc. (vs. Baseline)	TCD (Speedup)	Acc. (vs. Baseline)	TCD (Speedup)
Qiskit Aer	Vanilla VQC	94%(0)	23(0)	96%(0)	51(0)
	Comp-VQC	99%(5%)	11(2.09×)	98%(2%)	23(2.22×)
IBM Q	Vanilla VQC	79%(-15%)	23(1.00×)	86%(-10%)	51(1.00×)
	Comp-VQC	99%(5%)	11(2.09×)	98%(2%)	23(2.22×)

Acc.(vs. Baseline)	ibm_lagos	ibm_perth	ibm_jakarta
Vanilla VQC(TCD=23)	79%(0)	86%(0)	92%(0)
CompVQC(TCD=11)	99%(20%)	98%(12%)	100%(8%)

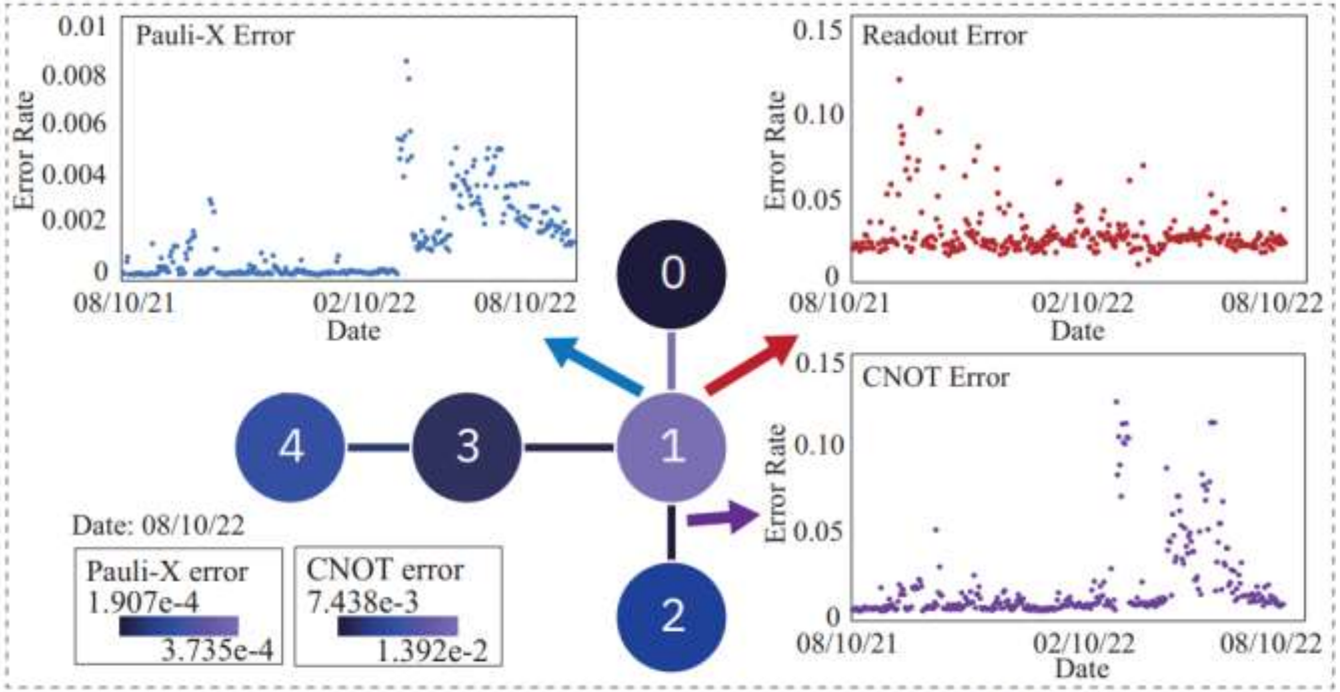
- CompVQC can reduce circuit length by 2x while the accuracy is also higher in a noisy environment.

Circuit compression can make the QNN model more robust to the noise

Compression-Aided Framework to battle againt fluctuating noise

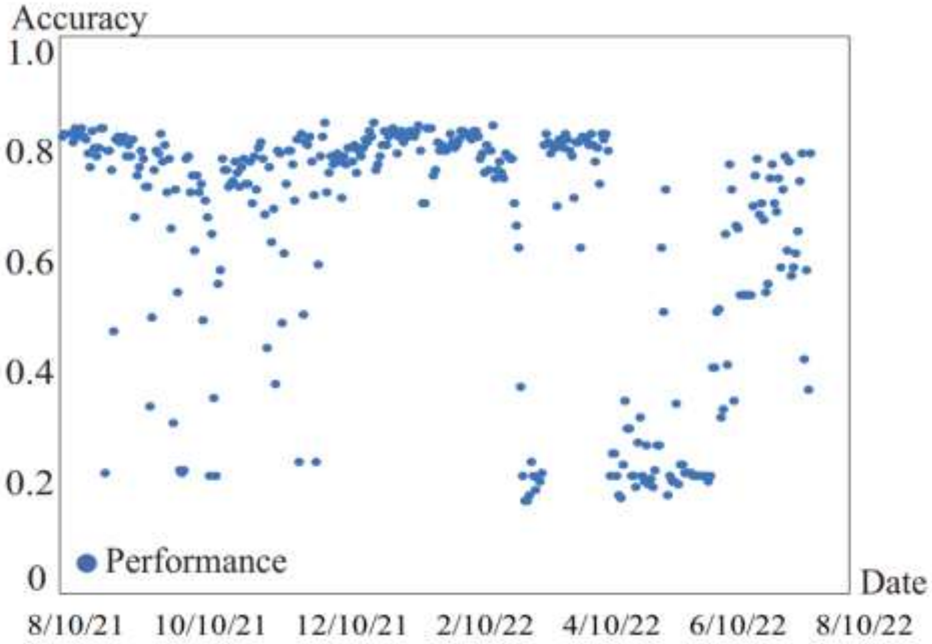
Fluctuating Quantum Noise

- **Fluctuating noise on quantum device**



(a) Fluctuating quantum noise on real quantum computer (1-year long daily profiling)

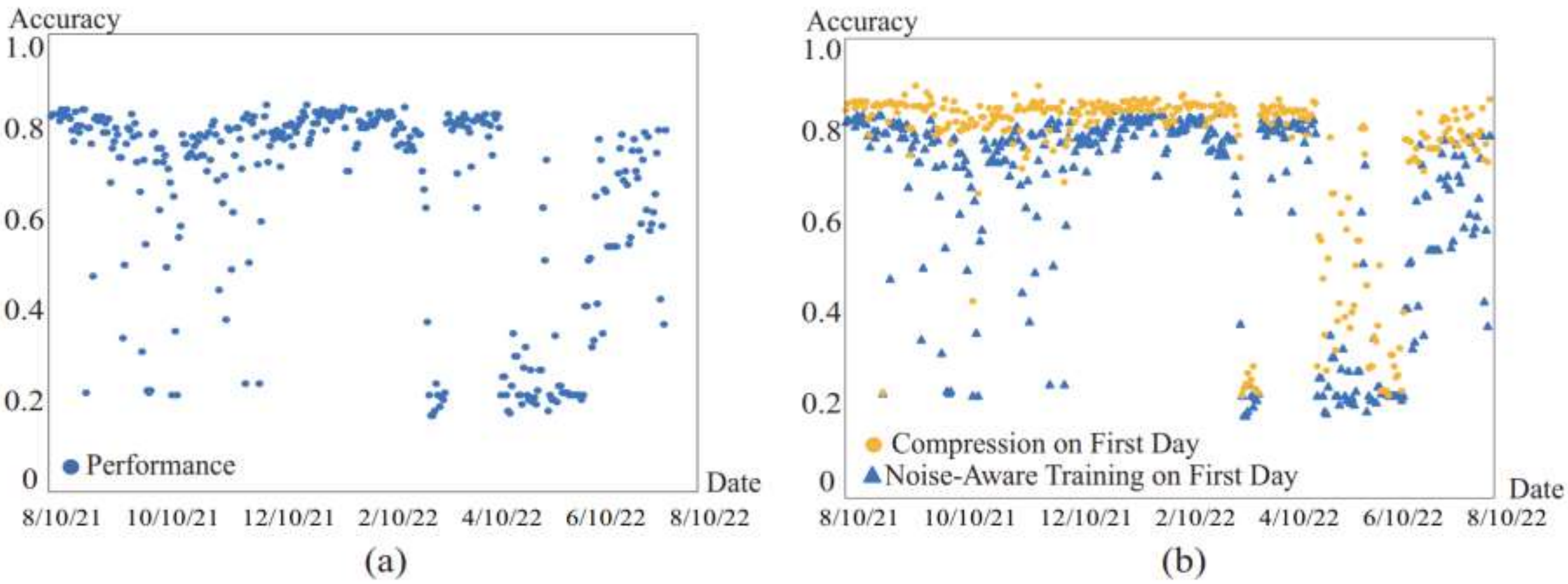
- **Fluctuating accuracy**



Fluctuating Quantum Noise

Observation: Fluctuating noise can collapse the model accuracy of a noise-aware trained QNN model

Observation: Compression can boost the performance of QNN than noise-aware training.

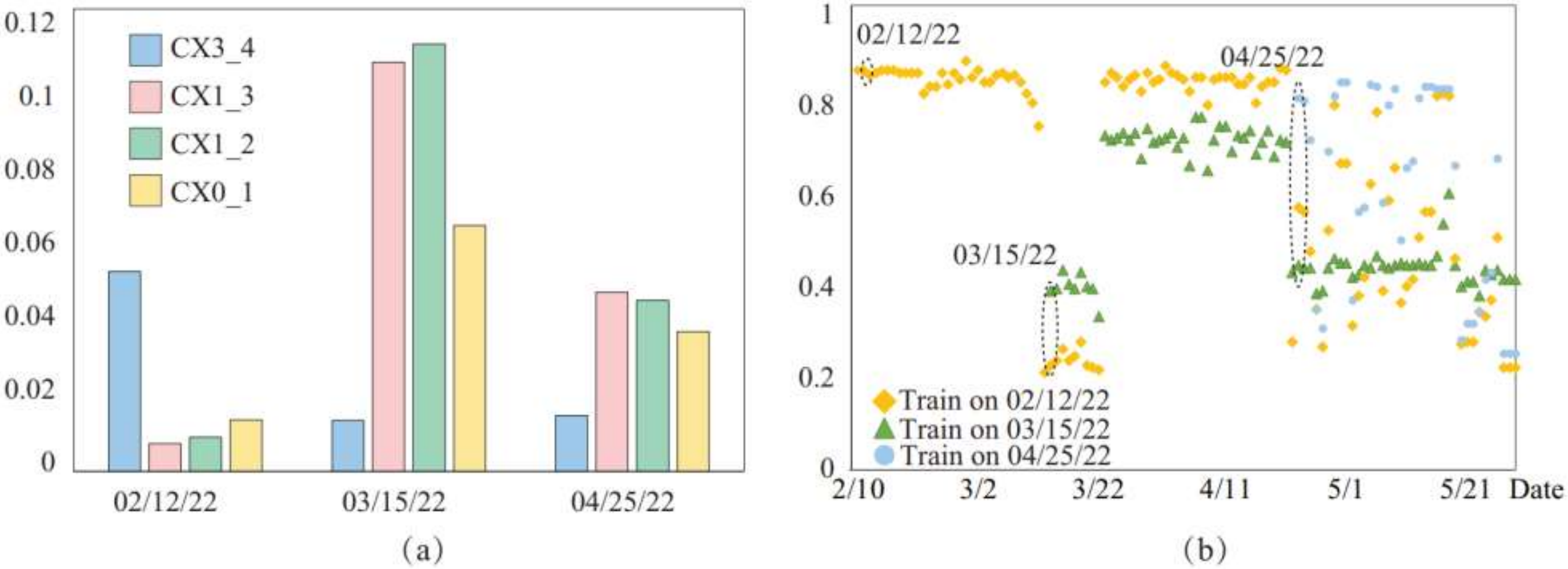


The accuracy of QNN on 4-class MNIST from August 2021 to August 2022 on IBM backend belem using Qiskit Simulation.

Battle Against Fluctuating Quantum Noise

Observation 1: Models Compressed on different noise levels (dates) have different performance on the same day

Observation 2: Models Compressed on one noise level have different performance on different days



➡ ① **Noise aware compression**

Noise-aware compression

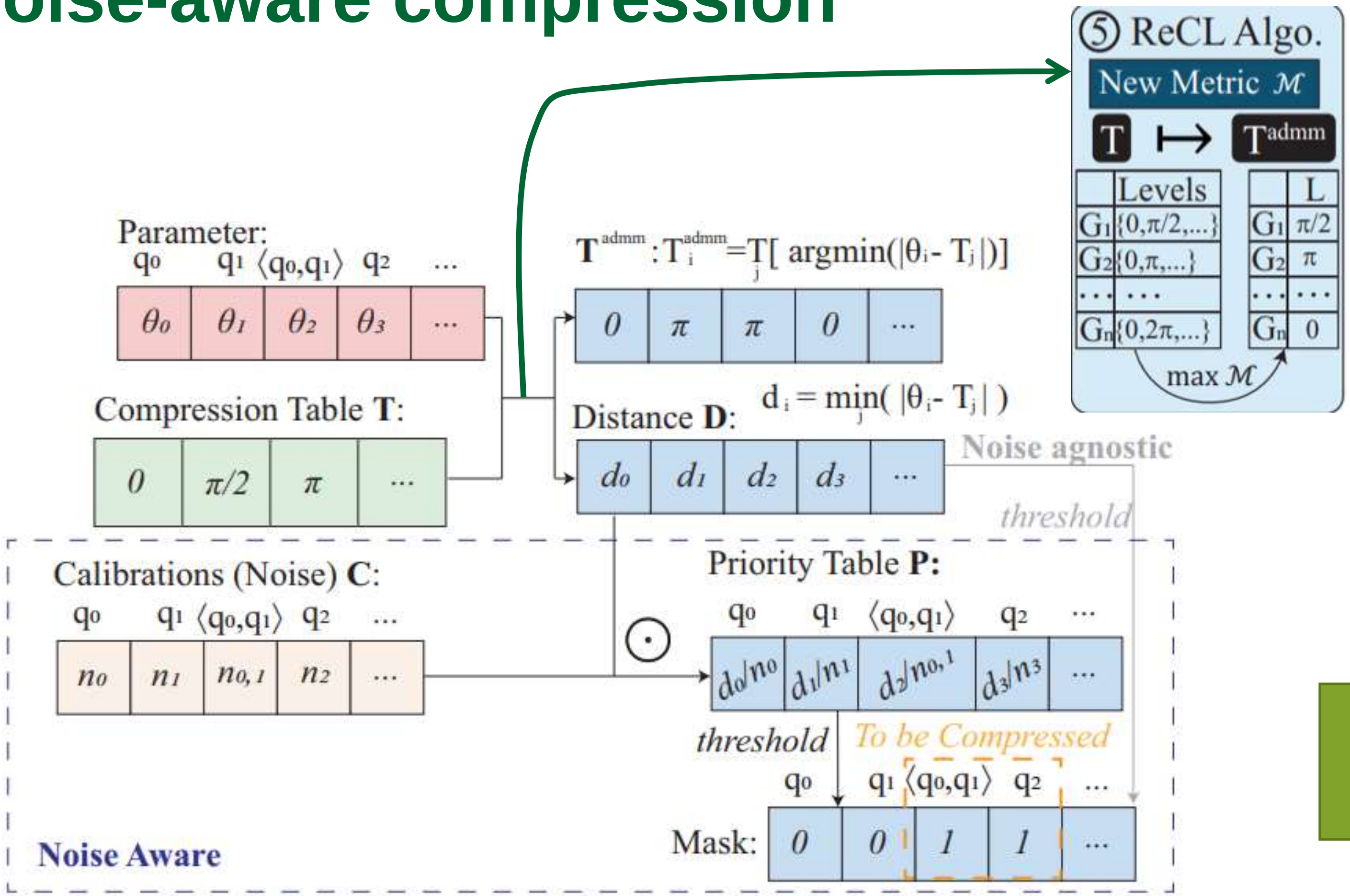
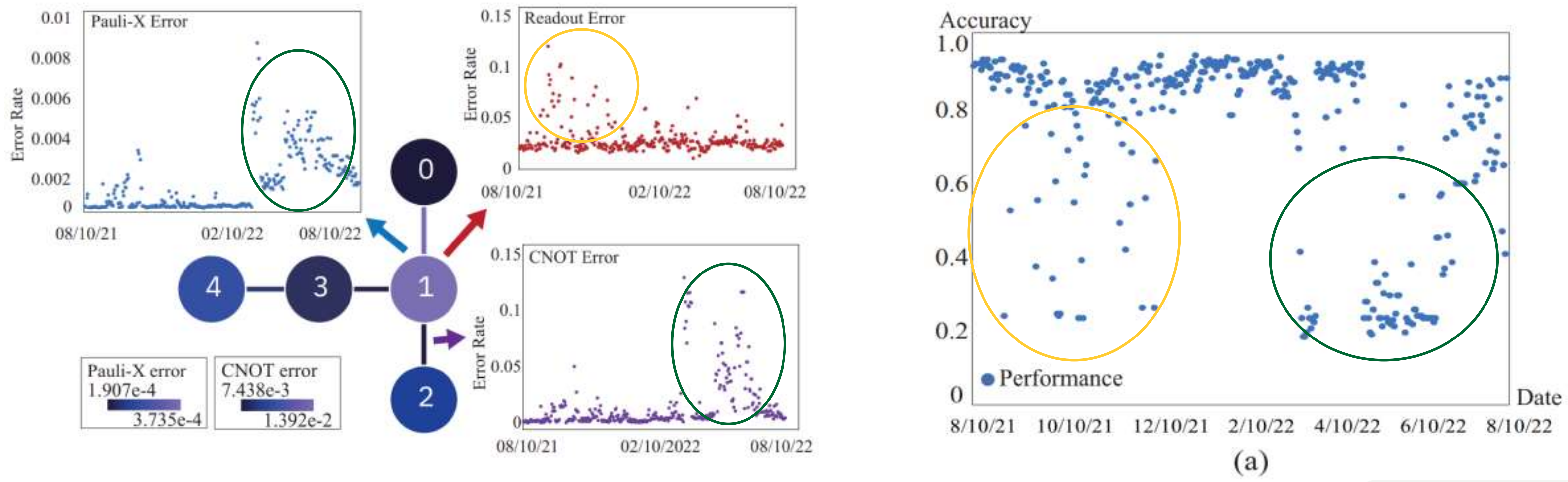


Fig. 6. Noise-aware mask generation in ADMM process.

Weight → Weight Mask to be pruned

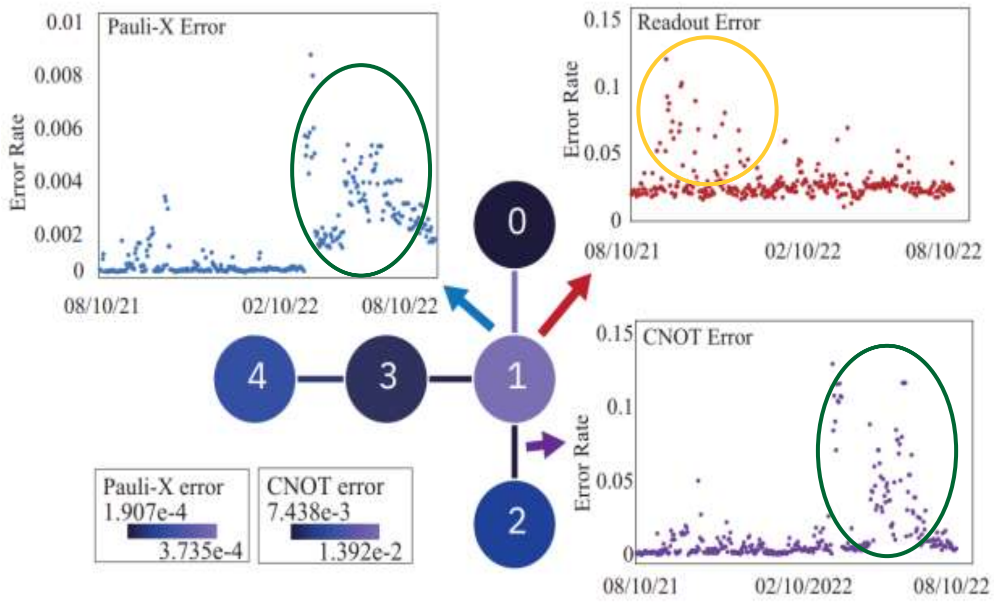
Model Repository Construction



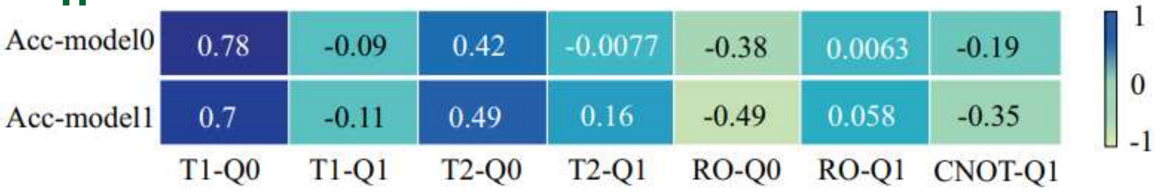
The cost of noise-aware compression everyday is very large.

=> clustering

Model Repository Construction



correlation



$$\rho = \left|\frac{\mathrm{cov}(X,Y)}{\sigma_x \sigma_y}\right|$$

X: accuracy of days
Y: different noise data of days

$$\boldsymbol{w} = [\rho_{\{T1-Q0\}}, \rho_{\{T1-Q1\}}, \ldots]$$

- **clustering**
- **distance:** $dist^{w}_{L1}(\mathbf{c_i}, \mathbf{c_j}) = dist_{L_1}(\mathbf{w} \cdot \mathbf{c_i}, \mathbf{w} \cdot \mathbf{c_j})$

Battle Against Fluctuating Quantum Noise

Solution: Offline + Online

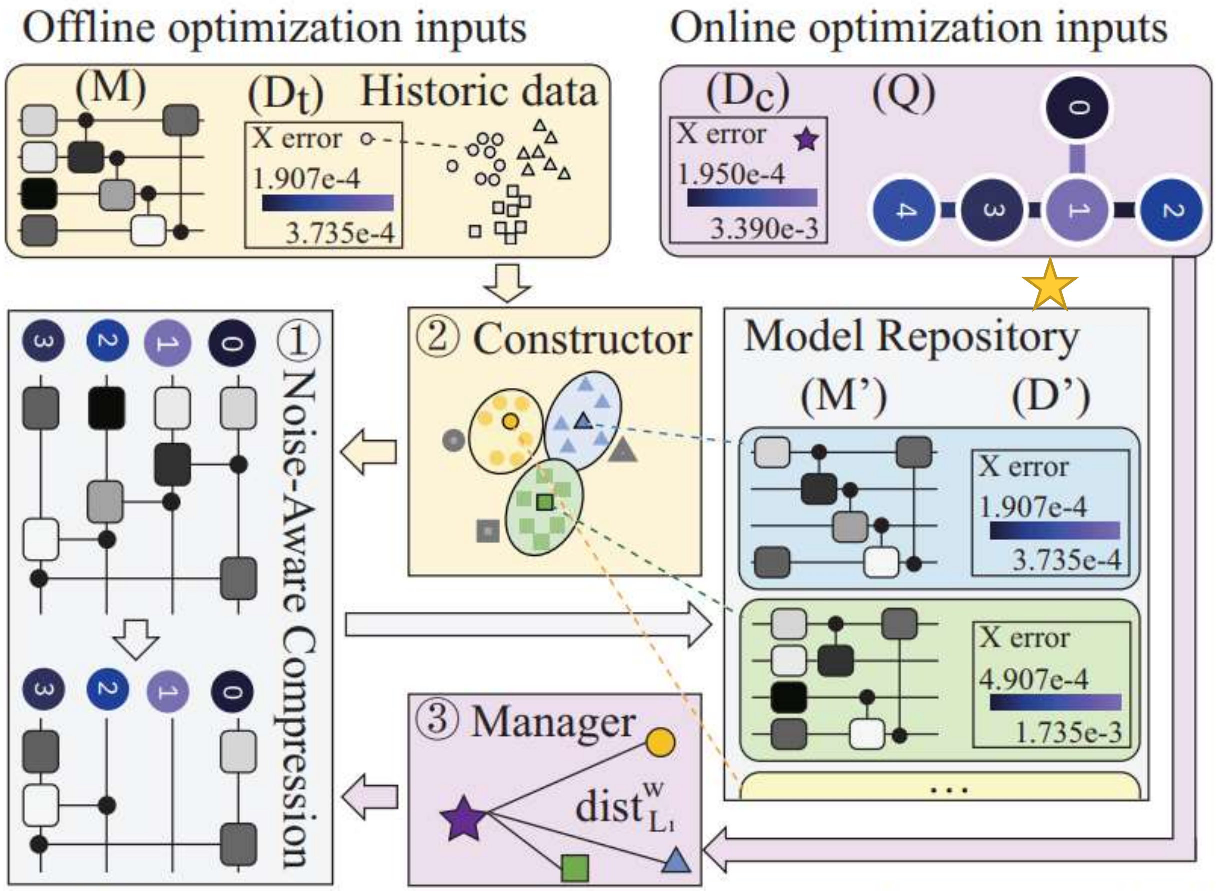


Fig. 5. Illustration of the proposed Compression-Aided Framework (QuCAD).

Offline:
Use historic data to construct a repository by clustering

Online:
① Select a model to do inference
② Maintain the repository: whether to generate new models into the model repository manager.

Battle Against Fluctuating Quantum Noise

Main experiment results

Dataset	Method	Mean Accuracy	vs. Baseline	Variance	Days over 0.8	vs. Baseline	Days over 0.7	vs. Baseline	Days over 0.5	vs. Baseline
Seismic Wave	Baseline	68.40%	0.00%	0.014	18	0	70	0	137	0
	Noise-aware Train Once [4]	68.85%	0.45%	0.014	19	1	78	8	137	0
	Noise-aware Train Everyday	68.28%	-0.11%	0.013	22	4	69	-1	138	1
	One-time Compression [15]	78.99%	10.59%	0.007	80	62	130	60	144	7
	QuCAD w/o offline	82.34%	13.95%	0.001	110	92	145	75	146	9
	QuCAD (ours)	**83.75%**	**15.36%**	**0.001**	**133**	**115**	**146**	**76**	**146**	**9**

Runing on real backend

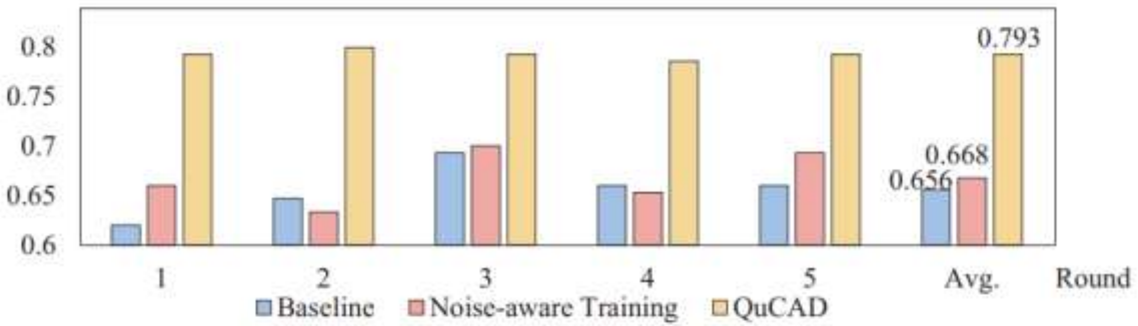


Fig. 8. On earthquake detection dataset, the performance of different approaches on the 7-qubit quantum device, ibm-jakarta.

Training Time

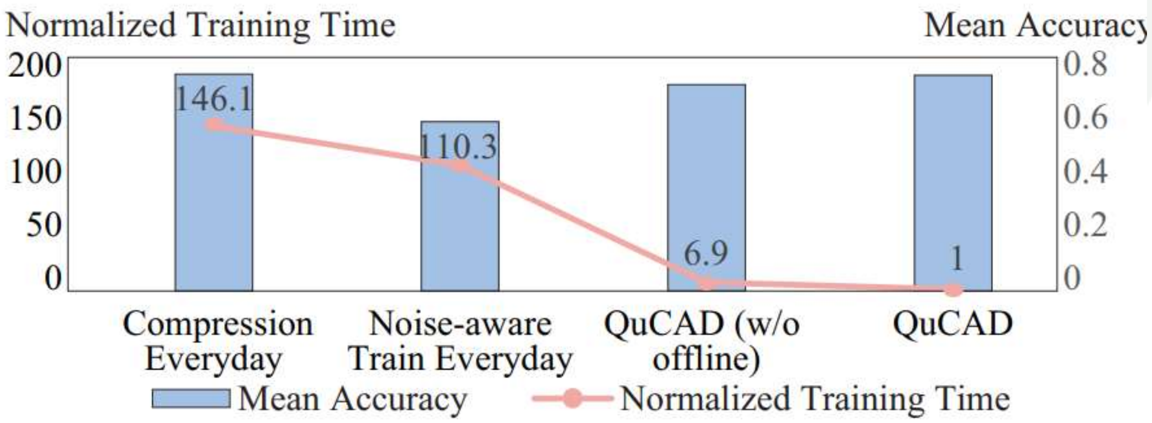


Fig. 7. The comparison of training time and accuracy.

Battle Against Fluctuating Quantum Noise

Distance Metric:

COMPARISON OF DIFFERENT CLUSTER

Method	K	Mean Acc. of Clusters	Mean Acc. of Samples
K-Means with L2	6	72.94%	78.45%
Proposed K-Means with $dist_{L1}^{w}$	6	**75.83%**	**80.68%**

Conclusion

There are two ways to optimize variational quantum circuits in NISQ era.

- **Build up a robust variational quantum circuits**
- **Efficient noise-aware adaptation**

Thanks for your attention!

zhu2@gmu.edu

George Mason University

College of Engineering and Computing

4511 Patriot Cir
Fairfax, Virginia 22030