IEEE/ACM
2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN
ICCAD
41st Edition

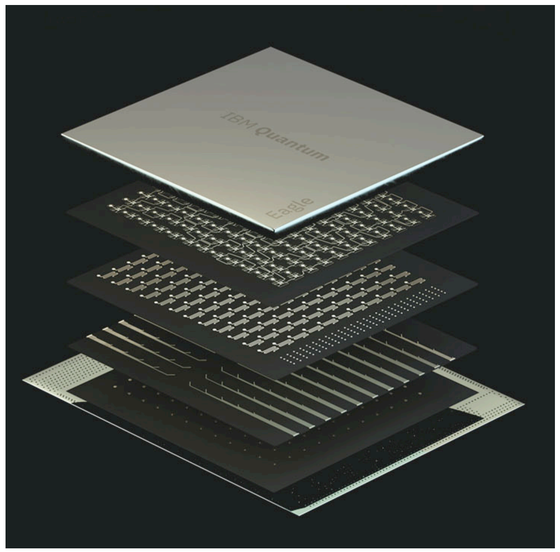
TorchQuantum Case Study for Robust Quantum Circuits

Hanrui Wang[1], Zhiding Liang[2], Jiaqi Gu[3], Zirui Li[4], Yongshan Ding[5], Weiwen Jiang[6], Yiyu Shi[2], David Z. Pan[3], Frederic T. Chong[7], Song Han[1]

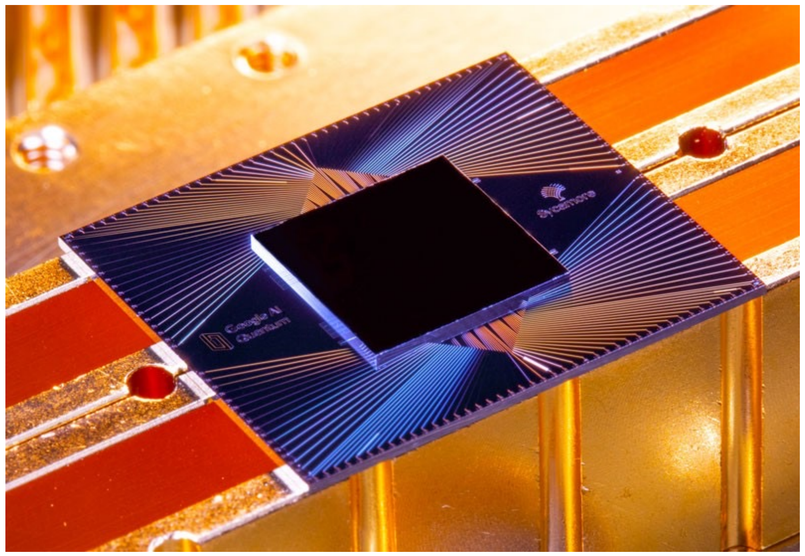
[1]MIT, [2]Notre Dame, [3]UT Austin, [4]Rutgers, [5]Yale, [6]GMU, UChicago[7]

Quantum Computing

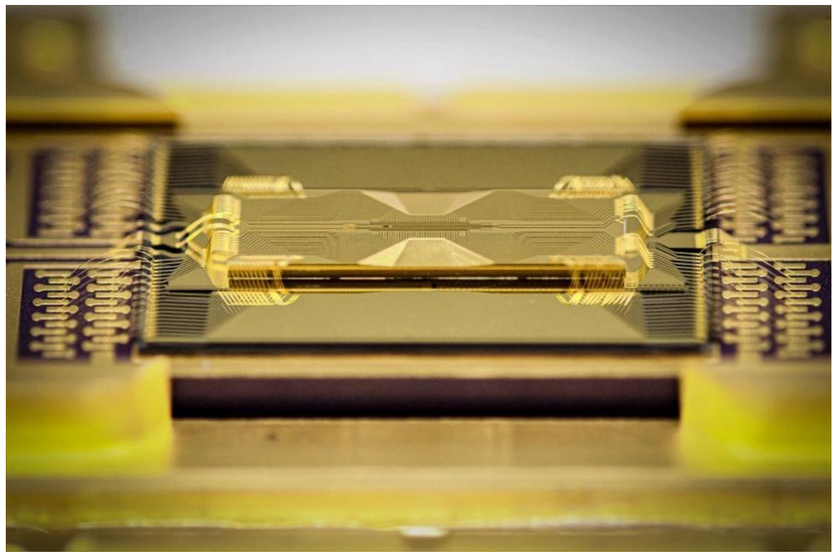
- Fast progress of quantum devices
- Different technologies
  - Superconducting, trapped ion, neutral atom, photonics, etc.



IBM
127 Qubit



Google
53 Qubits



IonQ
32+ Qubits

Torch Quantum

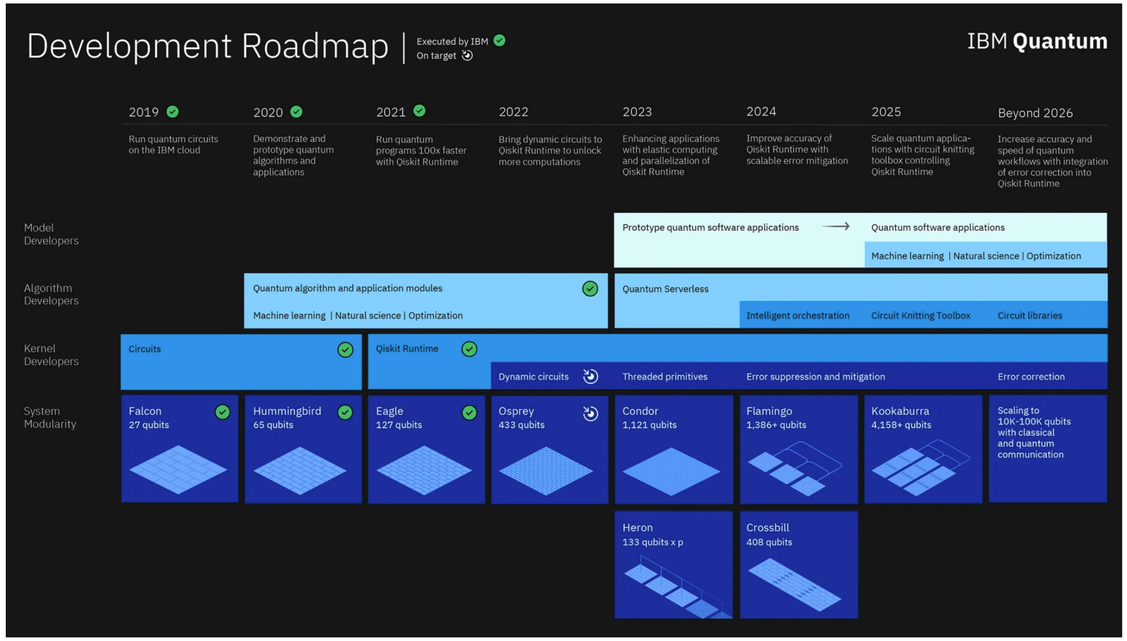
Materials at: https://torchquantum.org

MIT HAN LAB

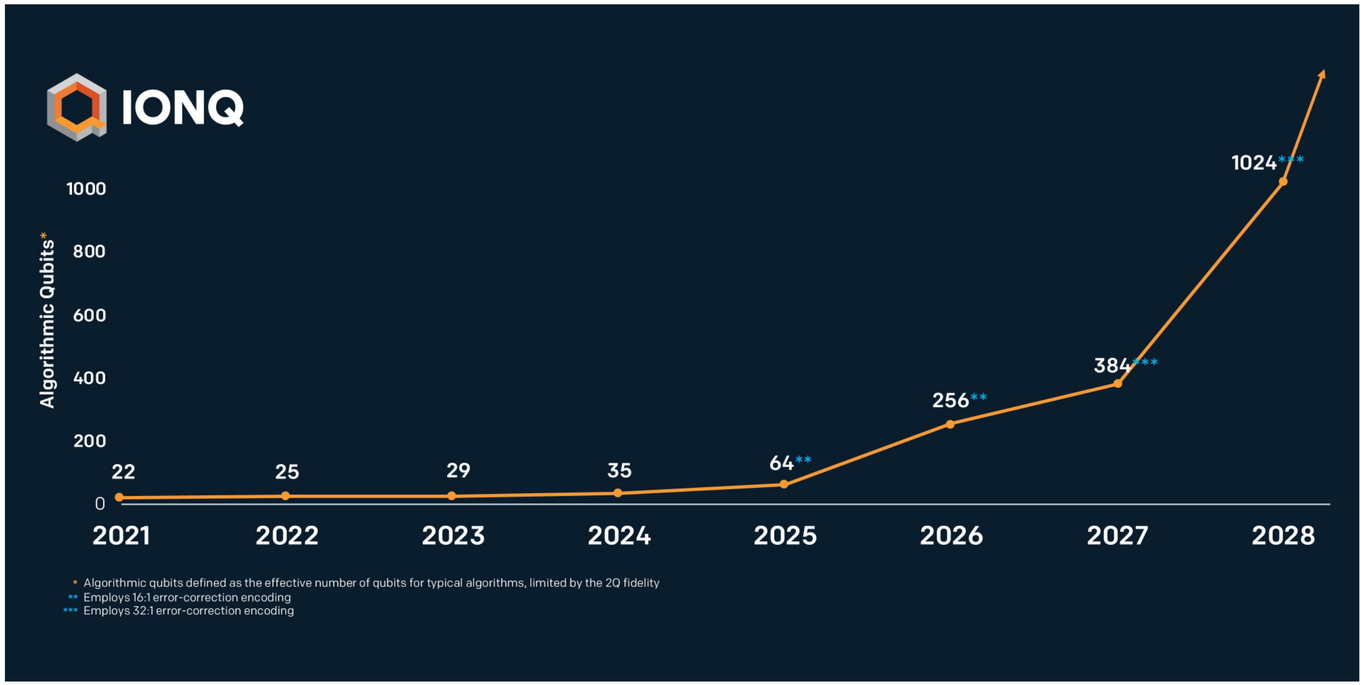
2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Quantum Computing

- The number of qubits increases exponentially over time
- The computing power increases exponentially with the number of qubits
- => “doubly exponential” rate



IBM Roadmap



IonQ Roadmap

Torch Quantum

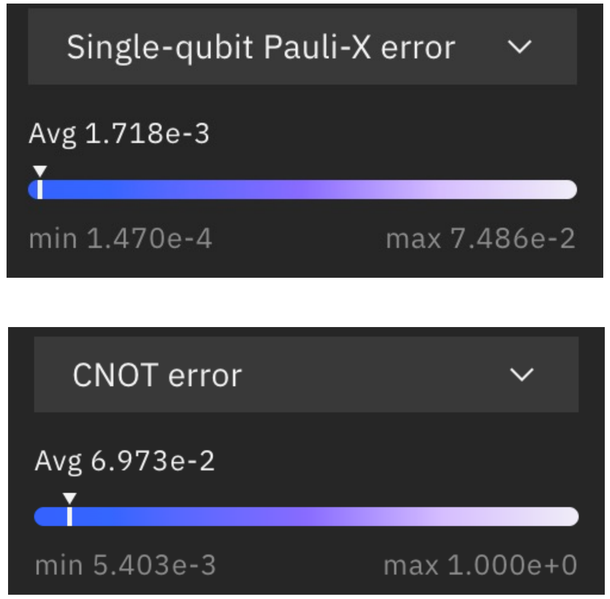
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

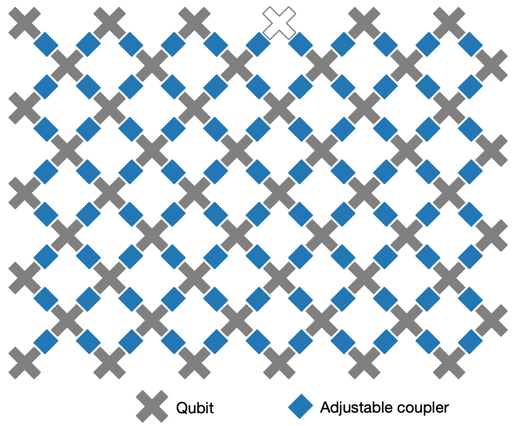
Quantum Computing in NISQ Era

- Noisy Intermediate-Scale Quantum (NISQ)
  - **Noisy**: qubits are sensitive to environment; quantum gates are unreliable
  - **Limited number of qubits**: tens to hundreds of qubits
  - **Limited connectivity:** no all-to-all connections



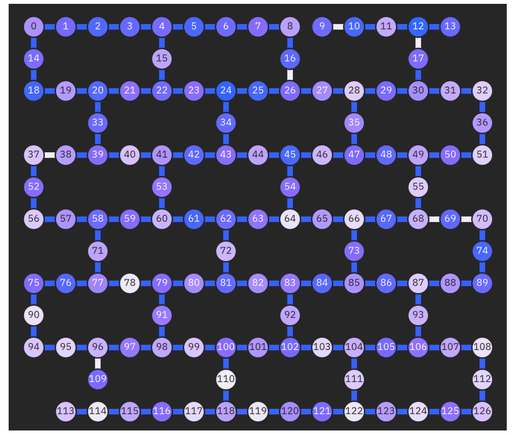
IBMQ Gate Error Rate

https://quantum-computing.ibm.com/



Google Sycamore 53Q

https://www.nature.com/articles/s41586-019-1666-5



IBM Washington 127Q

https://quantum-computing.ibm.com/

Torch Quantum

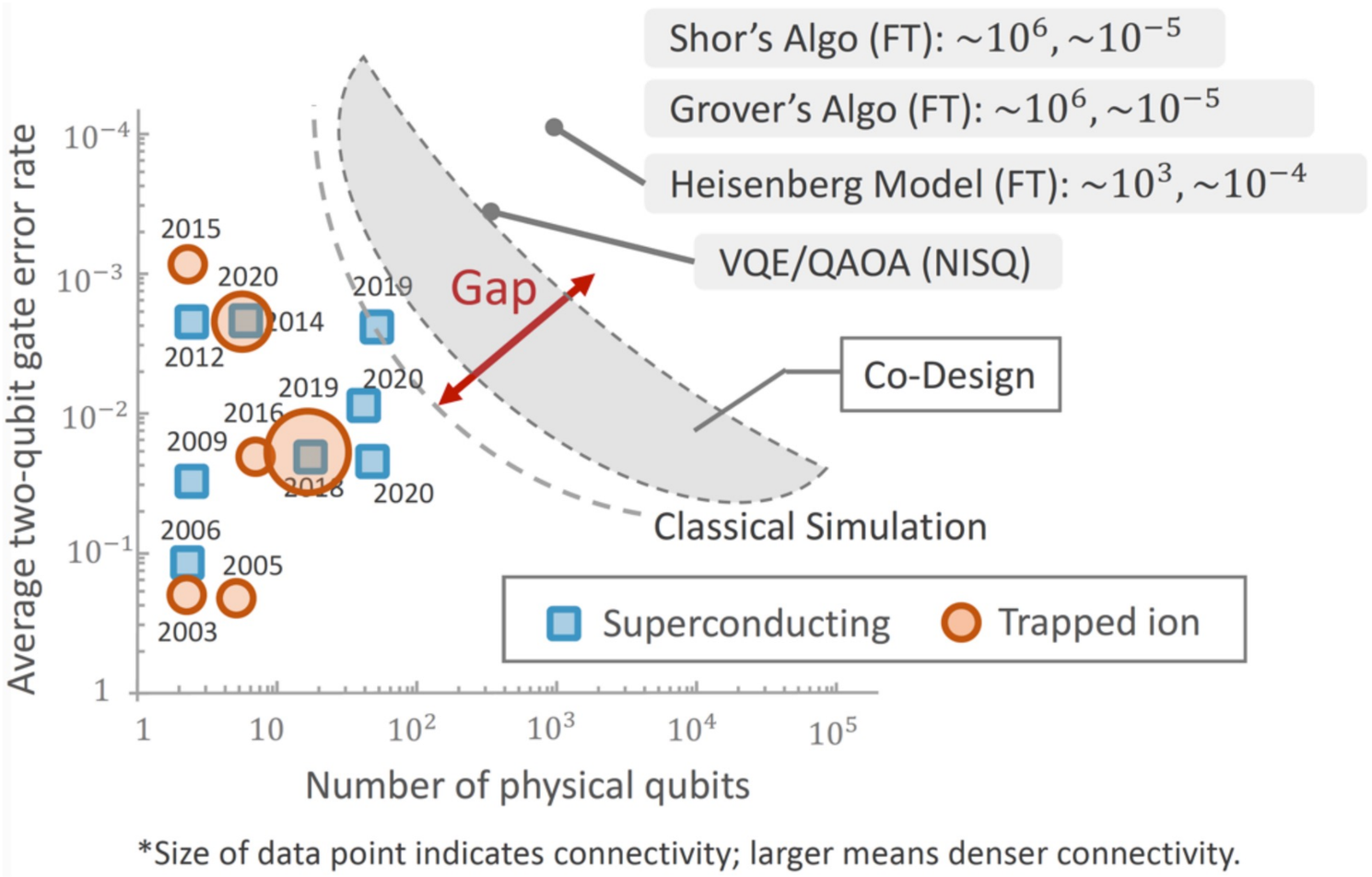
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

A Large Gap between Powerful Quantum Algorithms and Current Devices

- Close the gap with **machine learning** and **hardware-aware algorithm design**



*Size of data point indicates connectivity; larger means denser connectivity.

Fred Chong, https://indico.fnal.gov/event/47147/attachments/138920/174186/Chong-JTFI-2021.pdf18.pdf

Materials at: https://torchquantum.org

Torch Quantum

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Good Infrastructure is Critical

- To enable ML-assisted hardware-aware quantum algorithm design
- Need a simulation framework on classical computer
  - Fast speed
  - Convenience: PyTorch native
  - Portable between different frameworks with common Quantum IR
  - Scalable
  - Analyze circuit **behavior**
  - Study **noise** impact
  - Develop **ML model** for quantum optimization

TorchQuantum Library

- A fast library for classical simulation of quantum circuit in **PyTorch**
  - Automatic **gradient** computation for training parameterized quantum circuit
  - **GPU-accelerated** tensor processing with batch mode support
  - **Density matrix and state vector** simulators
  - **Dynamic computation graph** for easy debugging
  - Easy construction of **hybrid classical and quantum** neural networks
  - **Gate** level and **pulse** level simulation support
  - **Converters** to other frameworks such as IBM Qiskit
  - And so on…

Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations 
- 1.3 TQ Use Examples 
- 1.4 QNN Compression

Section 2

Use TorchQuantum on <u>Pulse</u> Level Optimization

- 2.1 Quantum Optimal Control
- 2.2 Variational Pulse Learning 

Section 3

Use TorchQuantum on <u>Gate</u> Level Optimization

- 3.1 QuantumNAS: Ansatz Search and Gate Pruning 
- 3.2 QuantumNAT: Noise Injection and Quantization
- 3.3 QOC: On-Chip Training
- 3.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations
- 1.3 TQ Use Examples
- 1.4 QNN Compression

Section 2

Use TorchQuantum on Pulse Level Optimization

- 2.1 Quantum Optimal Control
- 2.2 Variational Pulse Learning

Section 3

Use TorchQuantum on Gate Level Optimization

- 3.1 QuantumNAS: Ansatz Search and Gate Pruning
- 3.2 QuantumNAT: Noise Injection and Quantization
- 3.3 QOC: On-Chip Training
- 3.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

IEEE/ACM ICCAD 2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Quantum Bit

- Quantum Bit (Qubit)
  - Statevector: contains 2^n complex numbers for n qubit system
  - The square sum of magnitude of 2^n numbers are 1
  - 1 qubit:

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \quad a_0, a_1 \in \mathbb{C}$$

$$|a_0|^2 + |a_1|^2 = 1$$

  - 2 qubits:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad a_0, a_1, a_2, a_3 \in \mathbb{C}$$

$$|a_0|^2 + |a_1|^2 + |a_2|^2 + |a_3|^2 = 1$$

Quantum Bit

- Classical bits represented in statevector
  - Classical 0: $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
  - Classical 1: $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
  - An arbitrary quantum states: $\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = a_0 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + a_1 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Quantum Gates

- Qubit gates: operations on one qubit or multiple qubits
- The qubit gates can be represented with matrix format with dimension $2^n \times 2^n$
- All gate matrices are unitary matrices: the conjugate transpose is the same as its inverse
- Single qubit gates:
  - Not (X) gate:

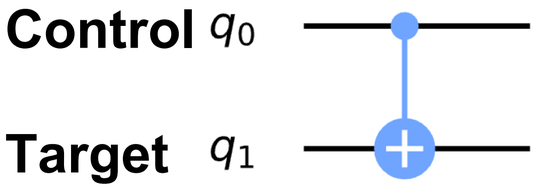
$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

  - Parameterized gate: Rotation X (RX) with parameter theta

$$RX(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

Quantum Gates

- 2-qubit gates:
  - Controlled Not (CNOT) gate:



$$CNOT = \begin{array}{c} \\ 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{cccc} \text{Output} & & & \\ 00 & 01 & 10 & 11 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

  - Controlled Rotation X (CRX) gate

$$CRX(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ 0 & 0 & -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

Quantum Gates

- Applying a gate to qubits is performing matrix-vector multiplication between the gate matrix and statevector
  - Apply an X gate to classical state 0, we get 1

$$X \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

  - Apply an CNOT gate to state 10, we get 11

$$CNOT \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations
- 1.3 TQ Use Examples
- 1.4 QNN Compression

Section 2

Use TorchQuantum on Pulse Level Optimization

- 3.1 Quantum Optimal Control
- 3.2 Variational Pulse Learning

Section 3

Use TorchQuantum on Gate Level Optimization

- 2.1 QuantumNAS: Ansatz Search and Gate Pruning
- 2.2 QuantumNAT: Noise Injection and Quantization
- 2.3 QOC: On-Chip Training
- 2.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

TQ for Statevector simulation

- Performing matrix-vector multiplication between the gate matrix and statevector
  - The tq.QuantumDevice stores the statevectors
    - `q_dev = tq.QuantumDevice(n_wires=5)`
  - Two ways of applying quantum gates: method 1:
    - `import torchquantum.functional as tqf`
    - `tqf.h(q_dev, wires=1)`

TQ for Statevector simulation

- Performing matrix-vector multiplication between the gate matrix and statevector
  - The tq.QuantumDevice stores the statevectors
    - `q_dev = tq.QuantumDevice(n_wires=5)`
  - Two ways of applying quantum gates: method 2:
    - `h_gate = tq.H()`
    - `h_gate(q_dev, wires=3)`

TQ for Statevector simulation

- Performing matrix-vector multiplication between the gate matrix and statevector
  - The tq.QuantumState class can also store the statevectors
    - `q_state = tq.QuantumState(n_wires=5)`
  - Three ways of applying quantum gates
    - `import torchquantum.functional as tqf`
    - `tqf.h(q_state, wires=1)`
    - `h_gate = tq.H()`
    - `h_gate(q_state)`
    - `q_state.h()`
    - `q_state.rx(wires=1, params=0.2 * np.pi)`

TQ for Statevector simulation

- Performing matrix-vector multiplication between the gate matrix and statevector
- The implementation of statevector and quantum gates are using the native data structure in PyTorch

```
_state = torch.zeros(2 ** self.n_wires, dtype=C_DTYPE)
_state[0] = 1 + 0j
```

```
'cnot': torch.tensor([[1, 0, 0, 0],
                      [0, 1, 0, 0],
                      [0, 0, 0, 1],
                      [0, 0, 1, 0]], dtype=C_DTYPE),
```

TQ for Density Matrix simulation

- Performing matrix-vector multiplication between the gate matrix and statevector
  - The tq.QuantumState class can also store the statevectors
    - `q_mat = tq.DesityMatrix(n_wires=5)`
  - Applying quantum gates
    - `q mat.h()`
    - `q_mat.rx(wires=1, params=0.2 * np.pi)`

```
class DensityMatrix(nn.Module):

    def __init__(self,n_wires: int,
                 bsz: int = 1):
        """Init function for DensityMatrix class(Density Operator)
        Args:
            n_wires (int): how many qubits for the densityMatrix.
        """
        super().__init__()

        self.n_wires=n_wires
        """
        For example, when n_wires=3
        matrix[001110] denotes the index of |001><110|=|index1><index2|
        Set Initial value the density matrix of the pure state |00...00>
        """
        _matrix = torch.zeros(2 ** (2*self.n_wires), dtype=C_DTYPE)
        _matrix[0] = 1 + 0j
        _matrix = torch.reshape(_matrix, [2]*(2*self.n_wires))
        self.register_buffer('matrix', _matrix)


        repeat_times = [bsz] + [1] * len(self.matrix.shape)
        self._matrix = self.matrix.repeat(*repeat_times)
        self.register_buffer('matrix', self._matrix)

        """
        Whether or not calculate by states
        """
        self._calc_by_states=True
```

Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Dynamic computation graph

- Performing matrix-vector multiplication between the gate matrix and statevector
  - The tq.QuantumState class can also store the statevectors
    - `q_state = tq.QuantumState(n_wires=5)`
    - `q_state.h(wires=1)`
    - `print(q_state)`
    - `q_state.sx(wires=3)`
    - `print(q_state)`

Batch Mode Tensorized Processing

- Performing matrix-vector multiplication between the gate matrix and statevector
  - The tq.QuantumState class can also store the statevectors
    - `q_state = tq.QuantumState(n_wires=5, bsz=64)`

GPU Support

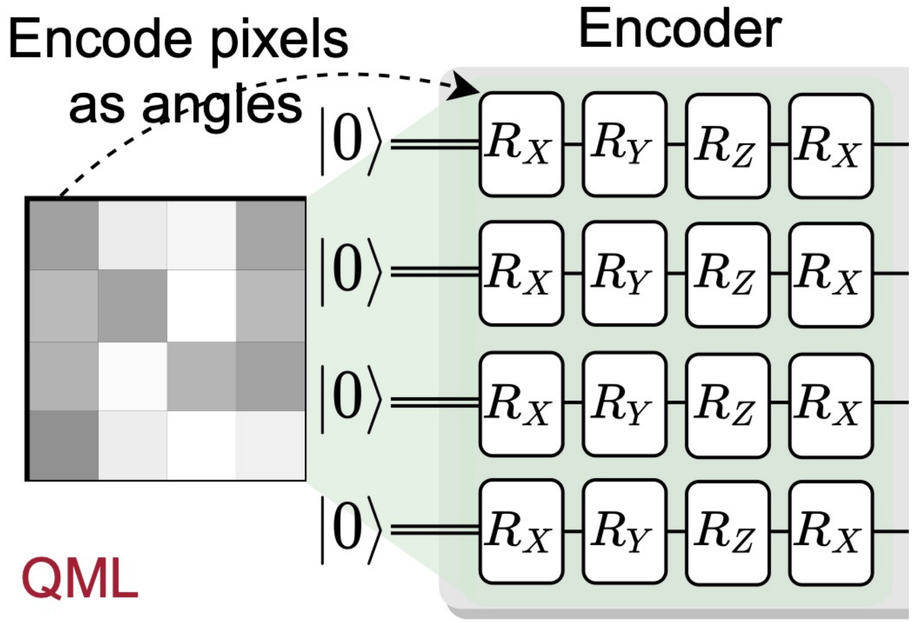
- Performing matrix-vector multiplication between the gate matrix and statevector
  - The tq.QuantumState class can also store the statevectors
    - `q_state = tq.QuantumState(n_wires=5, bsz=64)`
    - `q_state.to(torch.device('cuda'))`
  - Leverage the fast GPU support

Automatic Gradient Computation

- Performing matrix-vector multiplication between the gate matrix and statevector
  - The tq.QuantumState class can also store the statevectors
    - `q_state = tq.QuantumState(n_wires=2)`
    - `target_quantum_state = torch.tensor([0, 0, 0, 1])`
    - `loss = 1 - (q_state.get_states_1d()[0] @ target_quantum_state).abs()`
    - `loss.backward()`

Encoding Classical Data to Quantum various encoder support

- `tq.AmplitudeEncoder()` encodes the classical values to the amplitude of quantum statevector
- `tq.PhaseEncoder()` encodes the classical values using the rotation gates



Construct a Circuit Class

- Construct a class for circuit model
- `tq.QuantumModule` class
- In the __init__ function, create all the gates that will be used
- In the forward function, specify how the gates will be used in the circuit

```
Class q_model(tq.QuantumModule)
    def __init__():
        self.n_wires = 2
            self.rx_0 = tq.RX(has_params=True, trainable=True)
            self.ry_0 = tq.RY(has_params=True, trainable=True)
    def forward(q_dev):
        self.rx_0(q_dev)
        self.ry_0(q_dev)
```

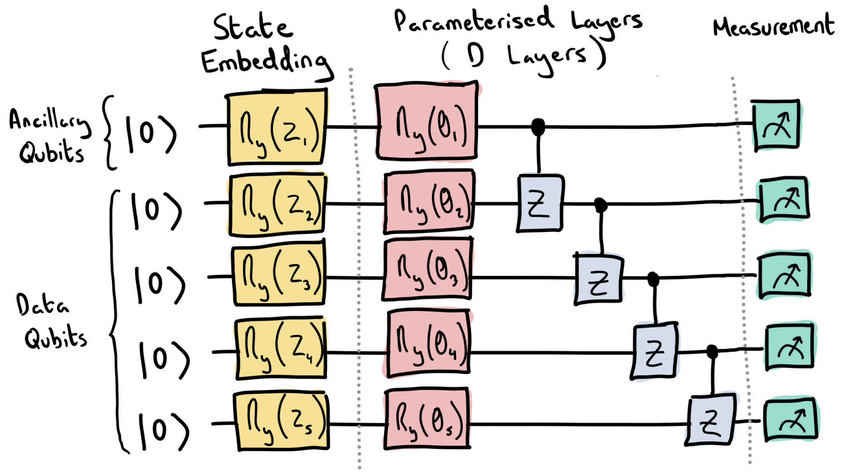
Conversion tq model to other frameworks

- Convert the tq.QuantumModule to other frameworks such as Qiskit
- `from torchquantum.plugins.qiskit_plugin import tq2qiskit`
- `circ = tq2qiskit(q_dev, q_model)`
- `circ.draw('mpl')`

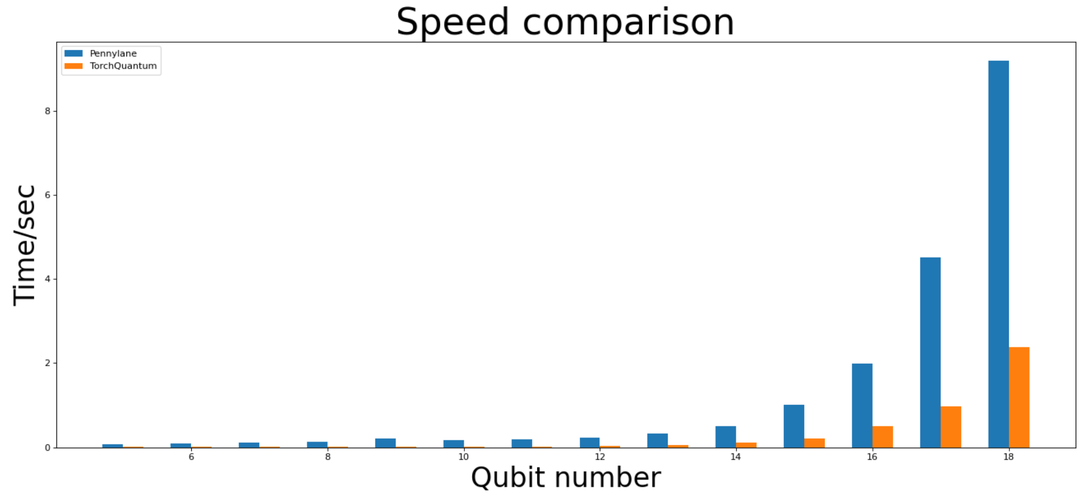
Easy Deployment on Real Quantum Machines

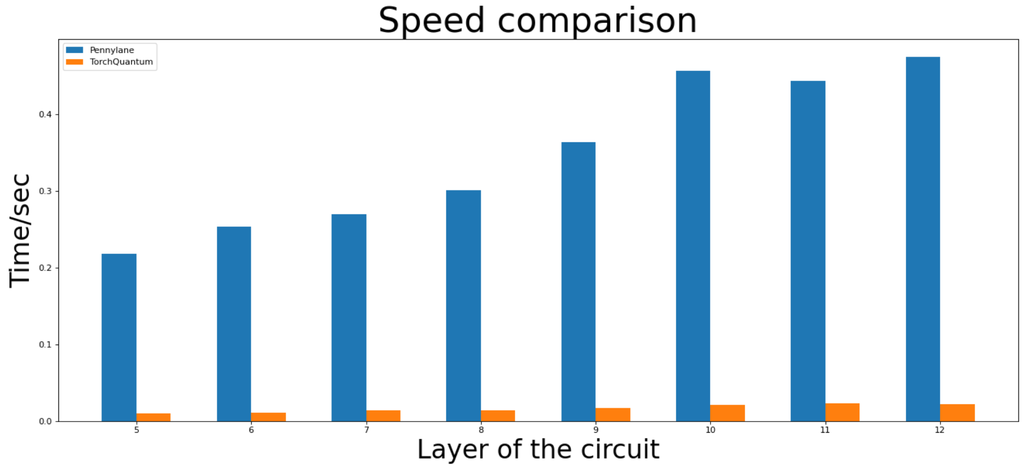
- Convert the tq.QuantumModule to other frameworks such as Qiskit
- `from torchquantum.plugins.qiskit_plugin import tq2qiskit`
- `from torchquantum.plugins.qiskit_processor import QiskitProcessor`
- `processor = QiskitProcessor(use_real_qc=False, max_jobs=1)`
- `circ = tq2qiskit(q_dev, model)`
- `circ.measure_all()`
- `res = processor.process_ready_circs(q_dev, [circ])`

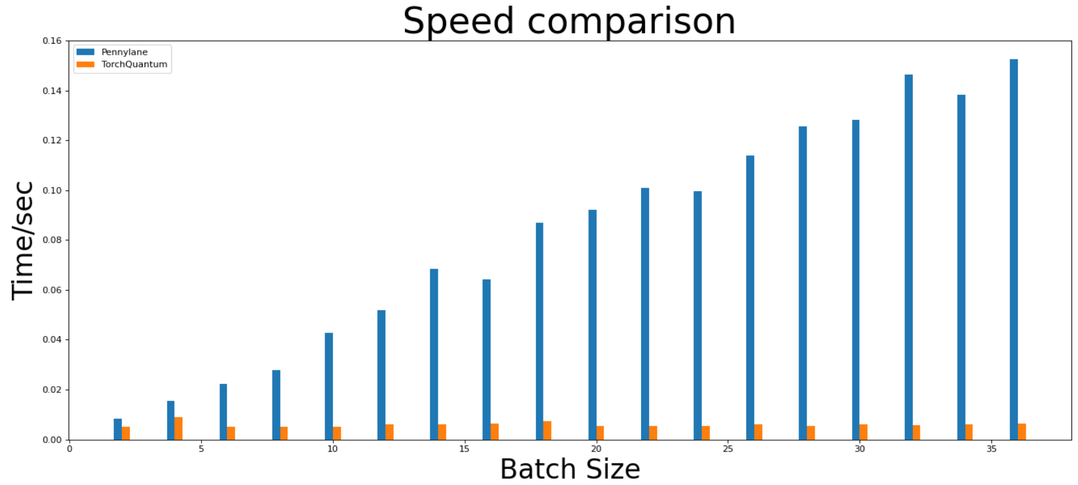
Compare the Speed of TQ and Pennylane



https://pennylane.ai/qml/_images/qcircuit.jpeg







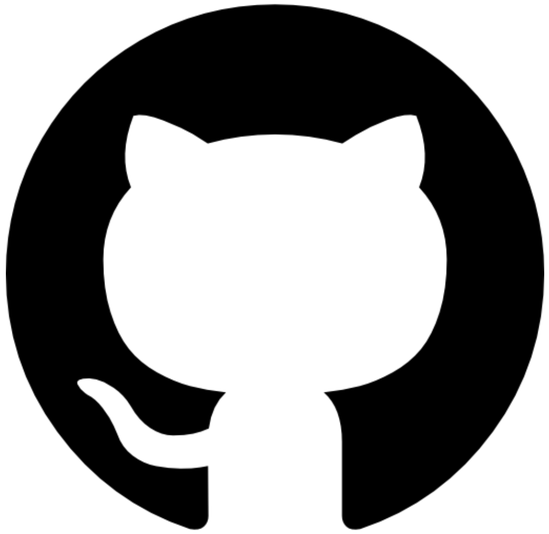
Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Hands-On Section
1.2 TorchQuantum Operations



Torch Quantum

Materials at: https

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations
- 1.3 TQ Use Examples
- 1.4 QNN Compression

Section 2

Use TorchQuantum on Pulse Level Optimization

- 3.1 Quantum Optimal Control
- 3.2 Variational Pulse Learning

Section 3

Use TorchQuantum on Gate Level Optimization

- 2.1 QuantumNAS: Ansatz Search and Gate Pruning
- 2.2 QuantumNAT: Noise Injection and Quantization
- 2.3 QOC: On-Chip Training
- 2.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

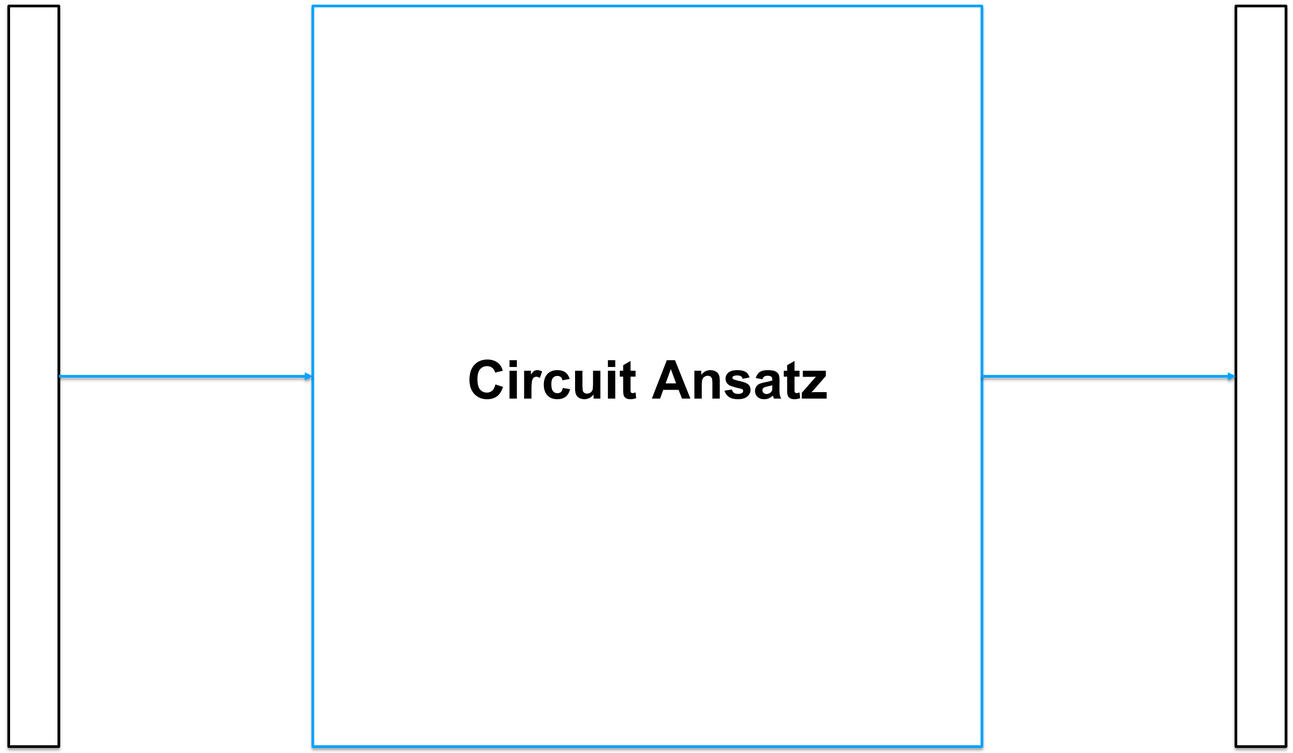
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

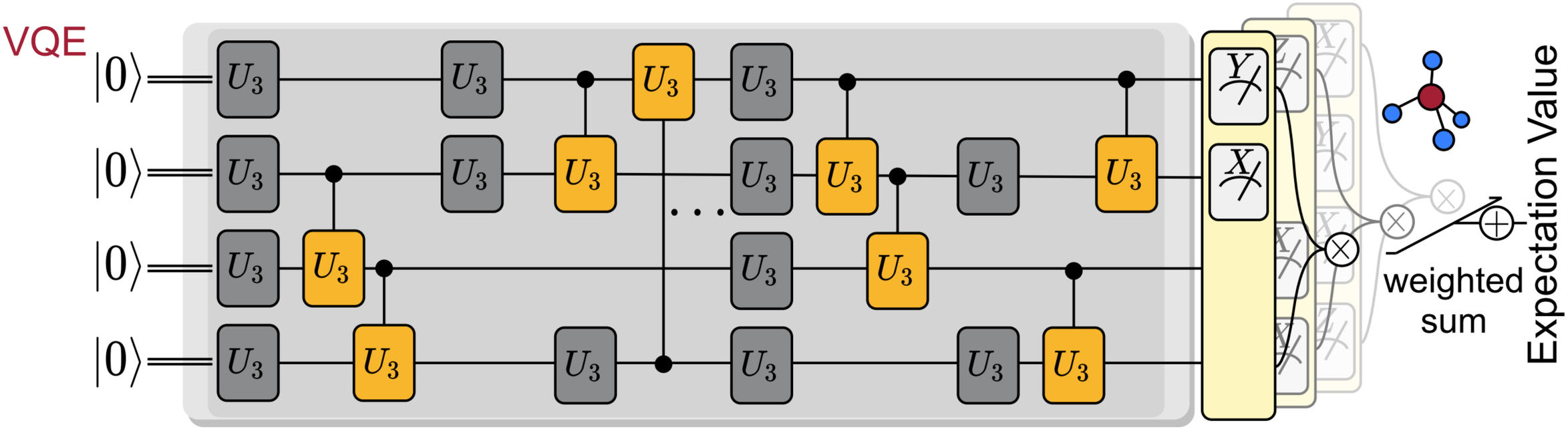
State Preparation with Parameterized Circuit

- Use parameterized circuit to prepare initial quantum states



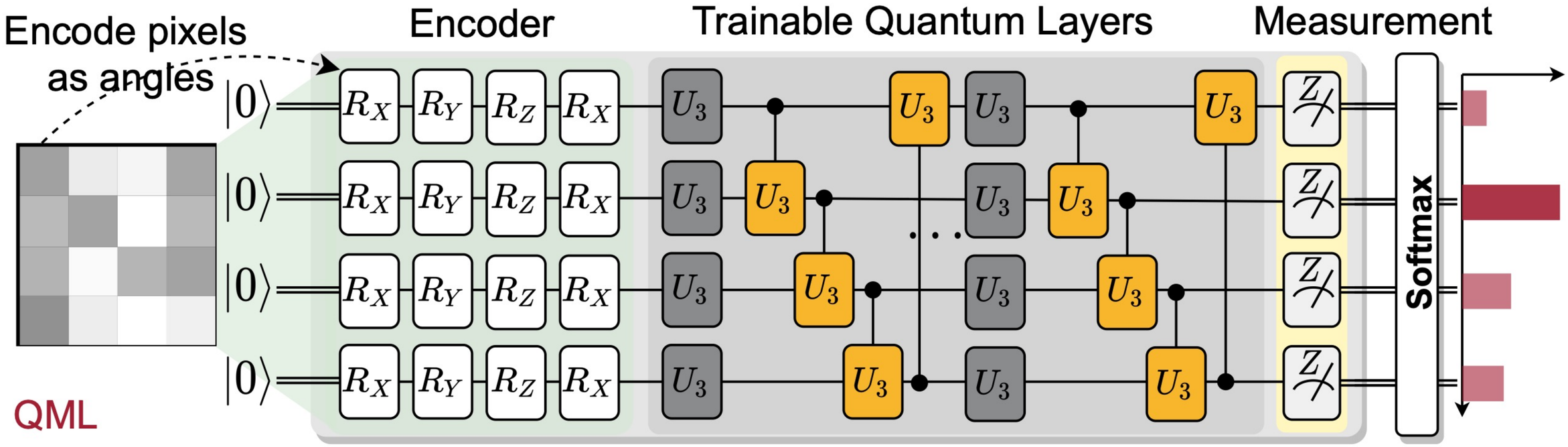
Variational Quantum Eigensolver

- VQE: Finds the ground state energy of molecule Hamiltonian



Quantum Neural Networks for MNIST Classification

- Encode the classical values using phase encoding
- Then trainable quantum layers
- Finally measurement layers



TorchQuantum Hub

- QML leaderboard: dataset, accuracy, implementation etc
- Continuously adding new results

Table

MNIST Leaderboard

Digit Set	Name	abbr.	Models	Accura...	Rollup	Paper	Simulat...
	Centroid	Centroid	a Quantum Centroid	86	arxiv:1401.21	✅Quantur	Simulation
	Centroid	Centroid	a Quantum Centroid	91	doi:10.1007/s	✅Quantur	Simulation
	QNN (Nearest	QNN	a Quantum Nearest	nearly 100	arxiv:1401.21	✅Quantur	Simulation
	QNN (Nearest	QNN	a Quantum Nearest	95	doi:10.1007/s	✅Quantur	Simulation
	QKNN (k-Nea	QKNN	a Quantum K-Neare	98	doi:10.1007/s	✅Quantur	Simulation
	QC	QC	a Circuit-centric Qu	96.7	arXiv:1804.00	✅Circuit-c	Simulation
{0, 1}	QRNN	QRNN	a Quantum Recurre	99.2±0.2	arXiv:2006.14	✅Recurre	Simulation
{0, 1}	ensemble of 4	ensemble of 4	ensemble of 4	99.6±0.16	arXiv:2006.14	✅Recurre	Simulation
	QNN (VQE) (1	QNN (17 qubi	a Quantum Neural N	98%	arXiv:1802.06	✅Classific	Simulation
{3, 6}	QNN (VQE) (1	QNN (17 qubi	a Quantum Neural N	98%	arXiv:2006.14	✅Recurre	Simulation
full MNST							

Models

Paper	Model	abbr.
Quantum algorithms for s	a Quantum Centroid	Centroid
✅Quantum Algorithms fo	a Quantum Nearest Neighbor	QNN
✅Quantum Algorithm for	a Quantum K-Nearest Neighbor	QKNN
✅Circuit-centric quantun	a Circuit-centric Quantum Classifier	QC
✅Classification with Qua	a Quantum Neural Network (17 qubits) (belongs to VQE	QNN (17 qubits
✅An initialization strateg	a Quantum Neural Network (2-12 qubits) (belongs to V	QNN (2-12 qub
	a Quantum Recurrent Neural Network	QRNN
	ensemble of 4	ensemble of 4

+ New

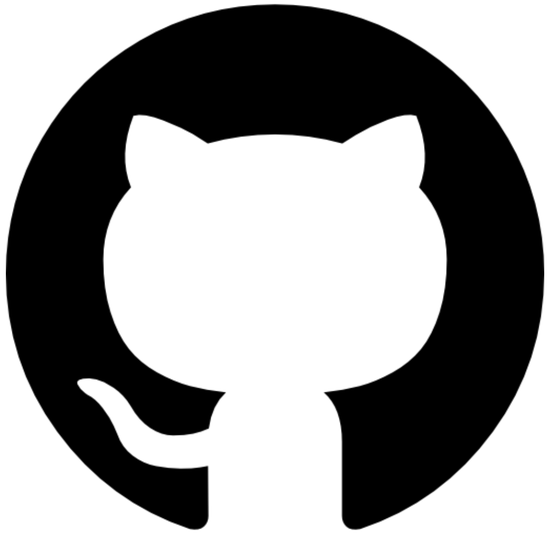
Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Hands-On Section
1.3 TQ Use Examples



Torch Quantum

Materials at: https

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations
- 1.3 TQ Use Examples
- 1.4 QNN Compression

Section 2

Use TorchQuantum on Pulse Level Optimization

- 3.1 Quantum Optimal Control
- 3.2 Variational Pulse Learning

Section 3

Use TorchQuantum on Gate Level Optimization

- 2.1 QuantumNAS: Ansatz Search and Gate Pruning
- 2.2 QuantumNAT: Noise Injection and Quantization
- 2.3 QOC: On-Chip Training
- 2.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

IEEE/ACM ICCAD 2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Quantum Neural Network Compression

Zhirui Hu[1], Peiyan Dong[2], Zhepeng Wang[1], Youzuo Lin[3], Yanzhi Wang[2], Weiwen Jiang[1]

[1] George Mason University

[2] Northeastern University

[3] Los Alamos National Laboratory

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations
- 1.3 TQ Use Examples
- 1.4 QNN Compression

Section 2

Use TorchQuantum on Pulse Level Optimization

- 2.1 Quantum Optimal Control
- 2.2 Variational Pulse Learning

Section 3

Use TorchQuantum on Gate Level Optimization

- 2.1 QuantumNAS: Ansatz Search and Gate Pruning
- 2.2 QuantumNAT: Noise Injection and Quantization
- 2.3 QOC: On-Chip Training
- 2.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

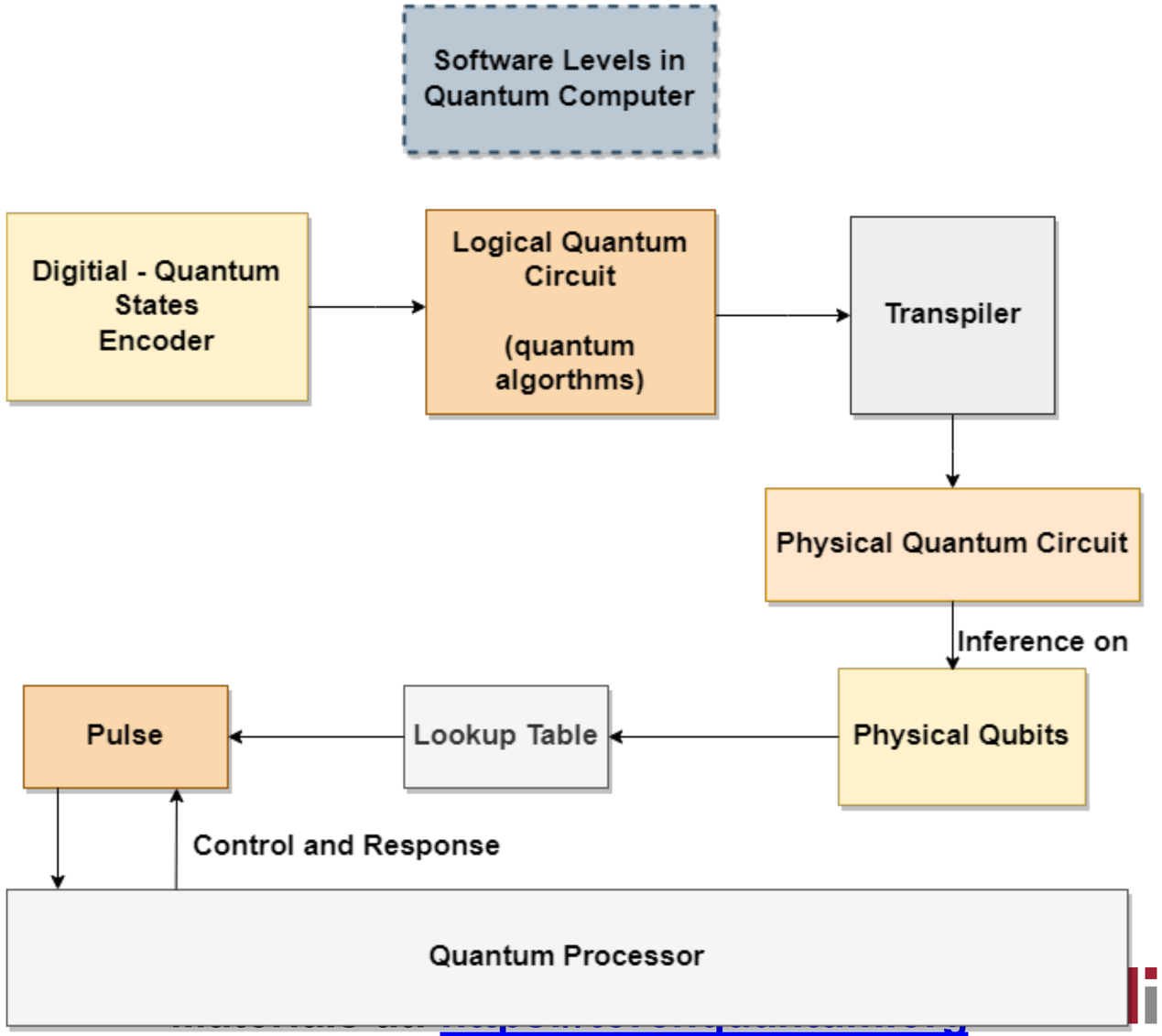
Materials at: https://torchquantum.org

MIT HAN LAB

IEEE/ACM ICCAD 2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Brief Overview of Workflow

- When we want to execute a quantum program on a quantum computer, we need to compile it first:



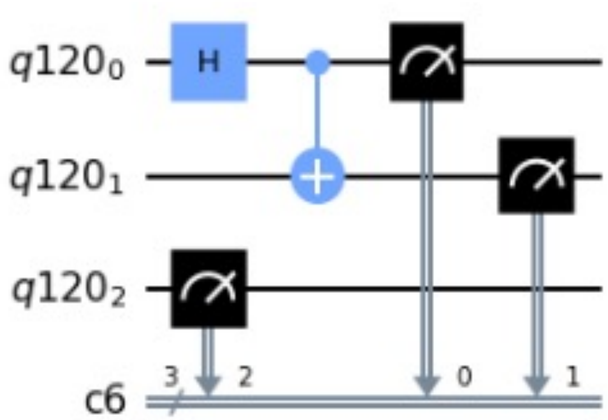
Torch Quantum

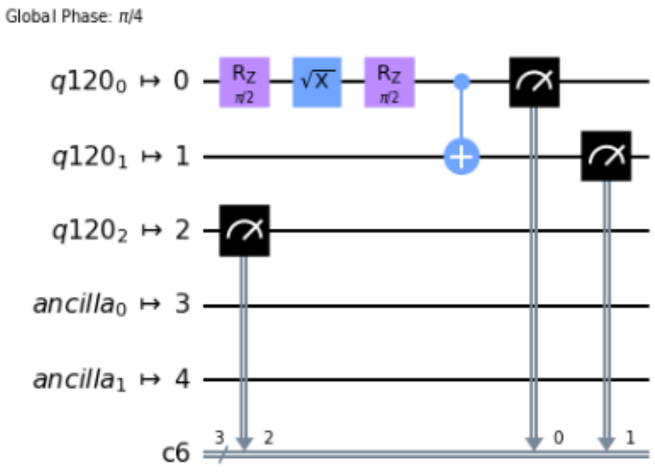
MIT HAN LAB

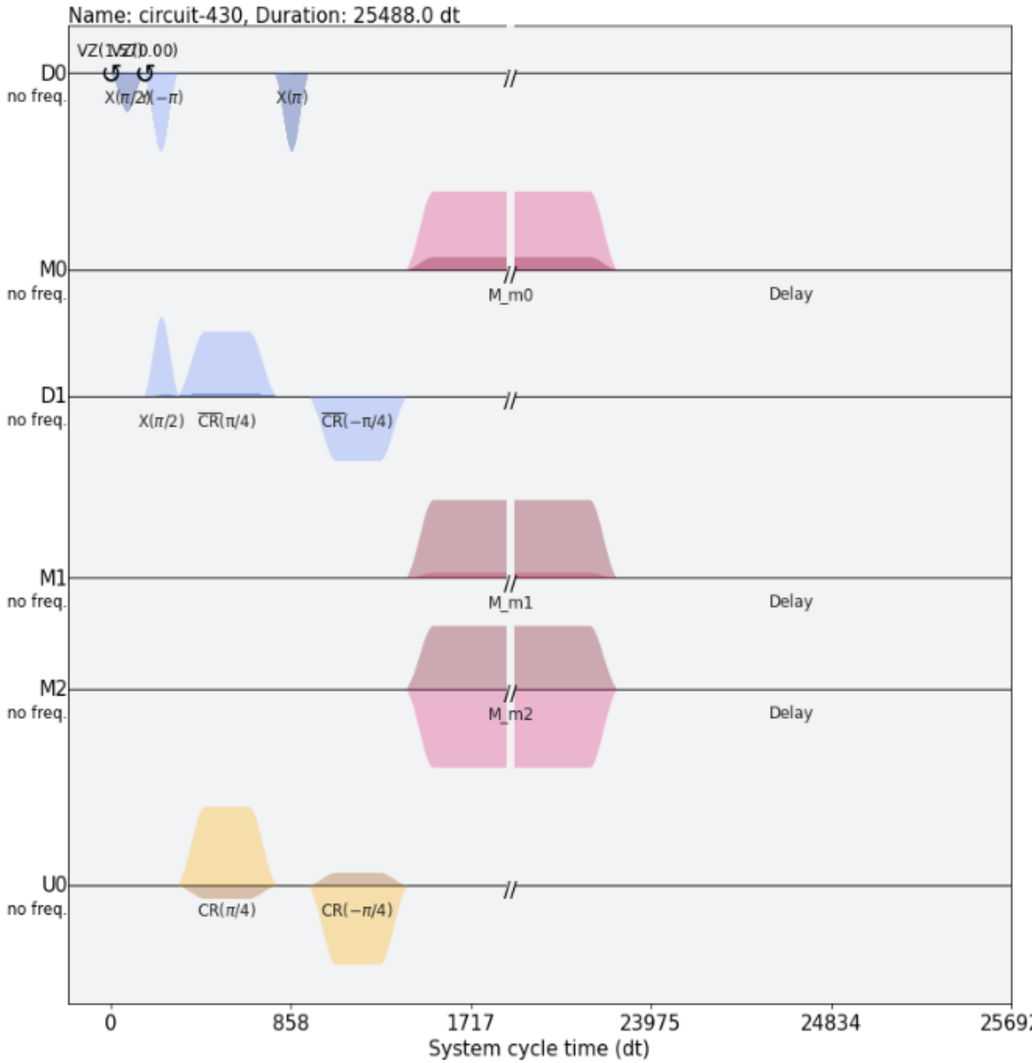
2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Pulse-Based and Gate-Based

- Gate- and Pulse-level are two different abstraction layers.





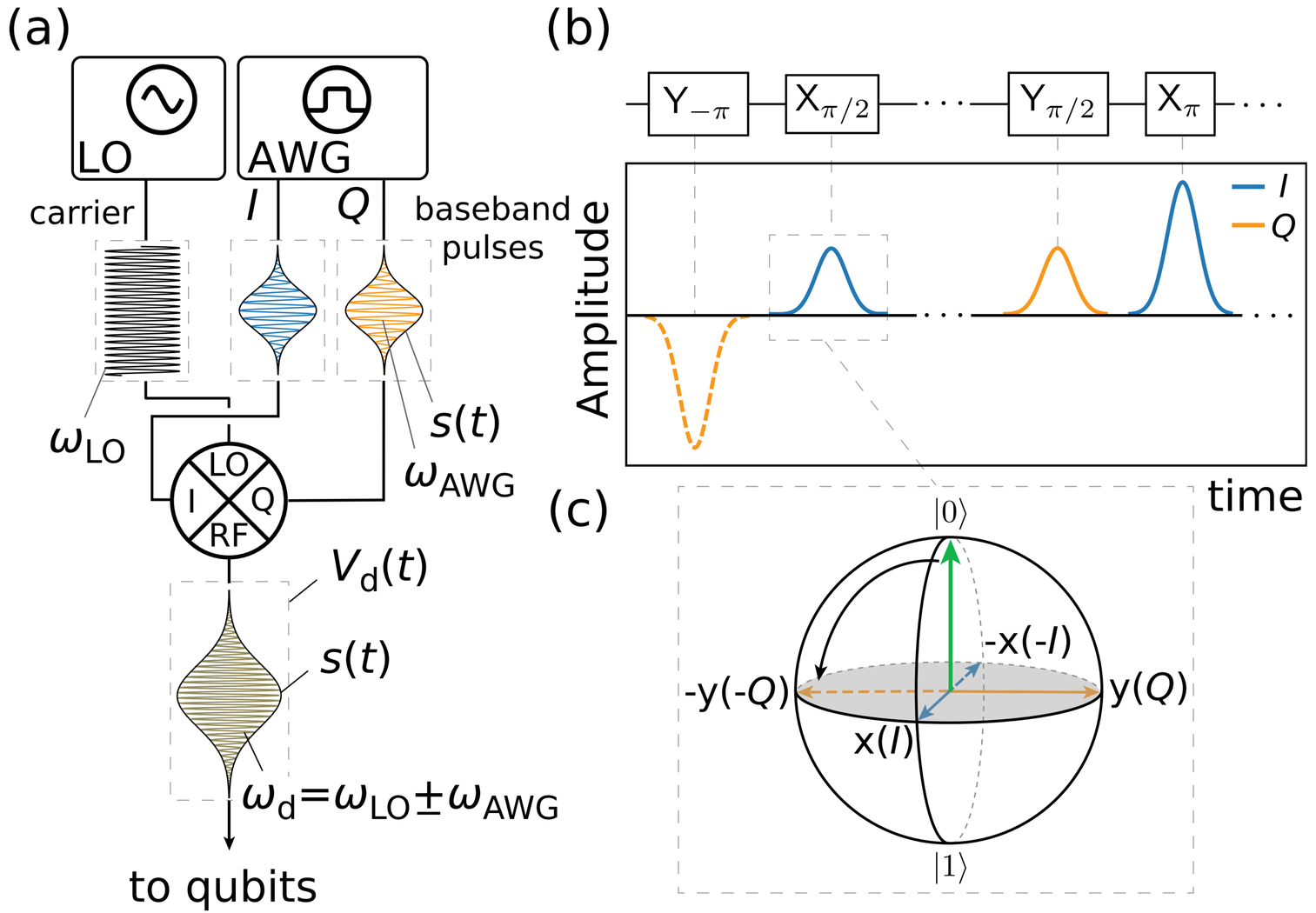


Torch Quantum

s at: h

Pulse-Based and Gate-Based

- The microwave is used to control the quantum bits.

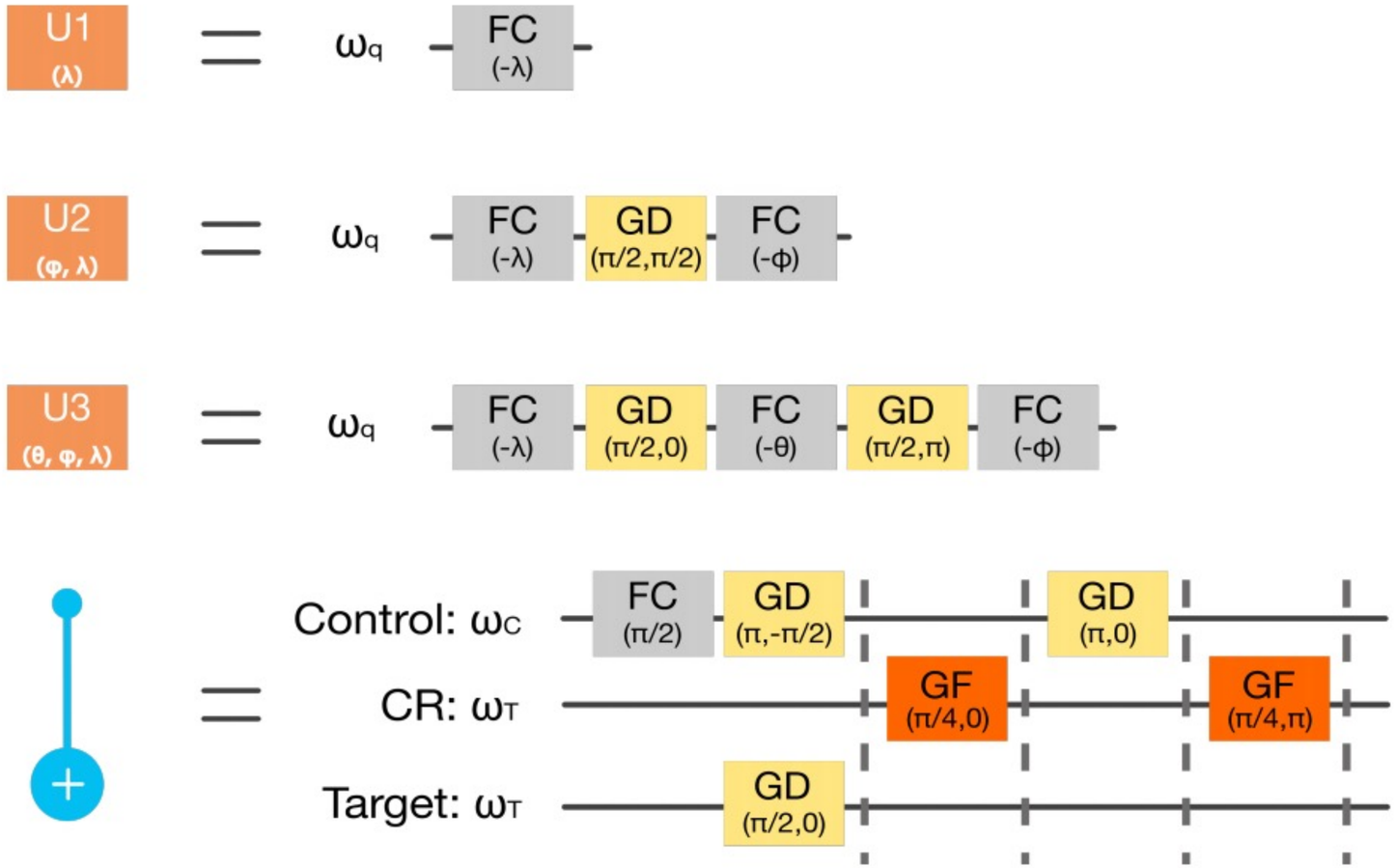


Krantz, Philip, et al. "A quantum engineer's guide to superconducting qubits." Applied Physics Reviews 6.2 (2019): 021318.

Torch Quantum

Gate-based Compilation

- Each gate corresponds to a pulse. One-by-one concatenation of all pulses realize the function of many gates.

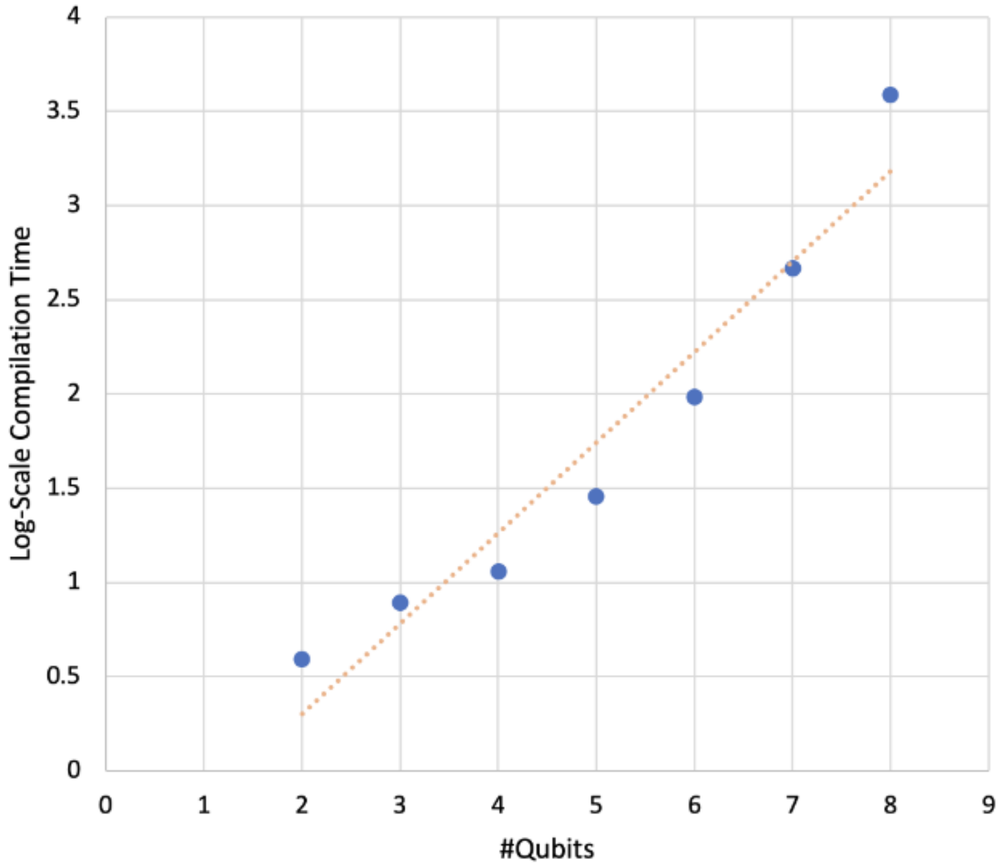


Torch Quantum

Materials at: https://torchquantum.org

Quantum Optimal Control(QOC)

- QOC is commonly used to generate pulses (from target unitary matrices to pulses)
- QOC is computationally expensive



Torch Quantum

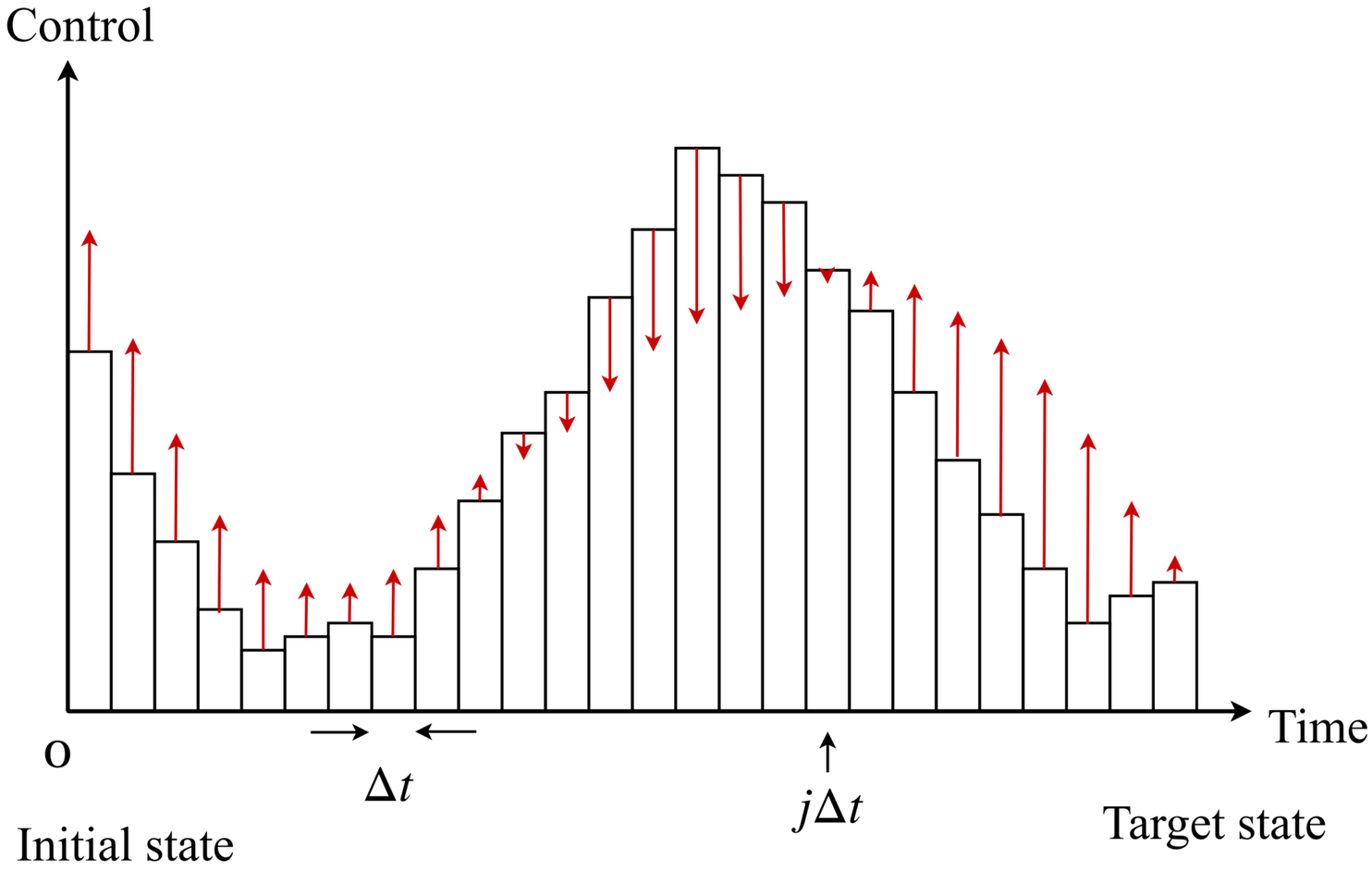
Materials at: https://torchquantum.org

MIT HAN LAB

ICCAD 2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Quantum Optimal Control (QOC)

- As shown in the figure, the control pulse is divided into multiple time slots. The amplitude of each time slot will be adjusted through optimization.



Shi, Yunong, et al. "Optimized compilation of aggregated instructions for realistic quantum computers." 2019

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations
- 1.3 TQ Use Examples
- 1.4 QNN Compression

Section 2

Use TorchQuantum on Pulse Level Optimization

- 2.1 Quantum Optimal Control
- 2.2 Variational Pulse Learning

Section 3

Use TorchQuantum on Gate Level Optimization

- 2.1 QuantumNAS: Ansatz Search and Gate Pruning
- 2.2 QuantumNAT: Noise Injection and Quantization
- 2.3 QOC: On-Chip Training
- 2.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

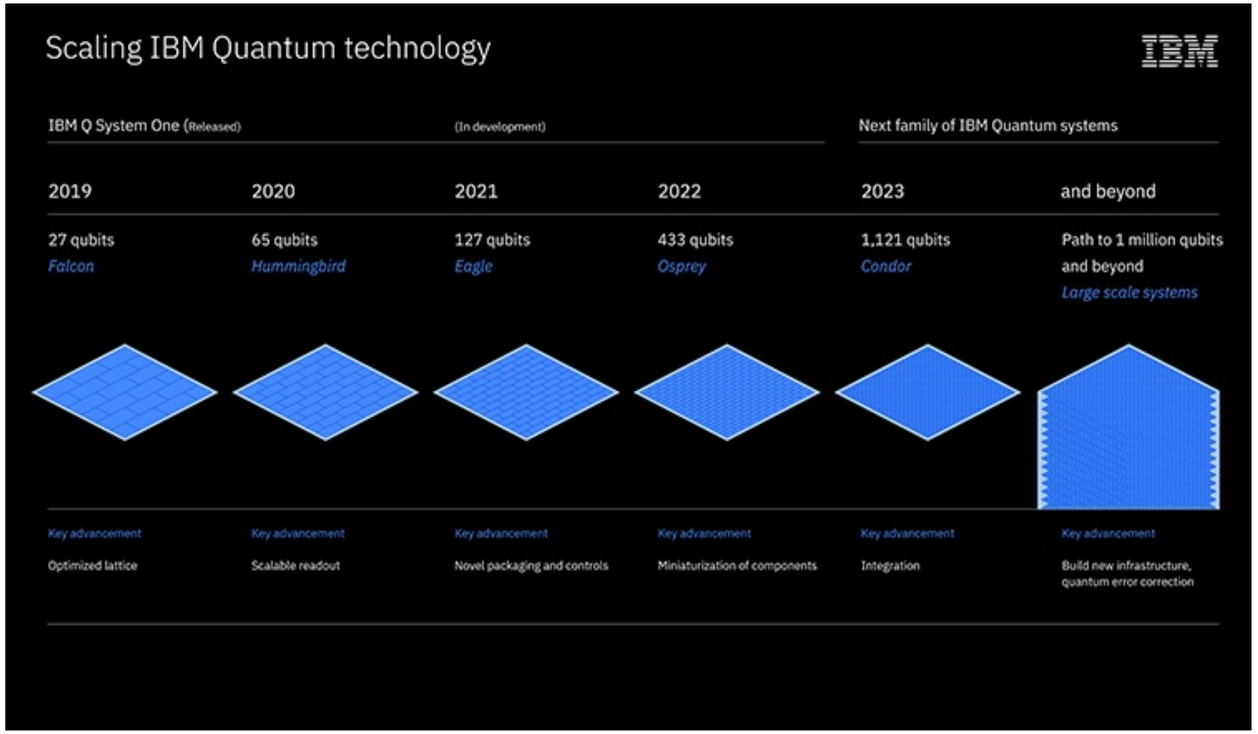
Materials at: https://torchquantum.org

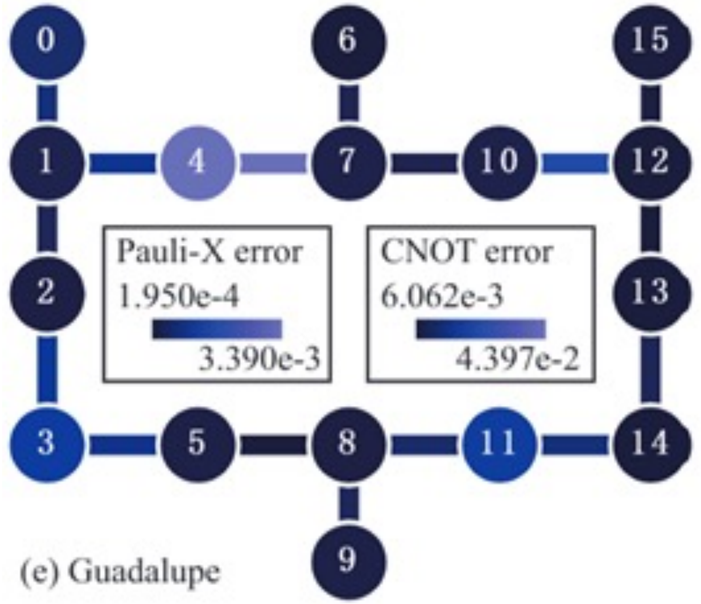
MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Variational Quantum Pulse Learning (VQP)

- Coherence time is limited in NISQ machines, which means we cannot perform a huge quantum circuit with enough depth and width in NISQ machines.
- Noise and compilation overhead seriously affect the performance of a quantum program.





Error rate on a bit in CMOS Device error rate is about 10^{-15}

But error rate on a quantum bit reaches 10^{-4} to 10^{-2}

Torch Quantum

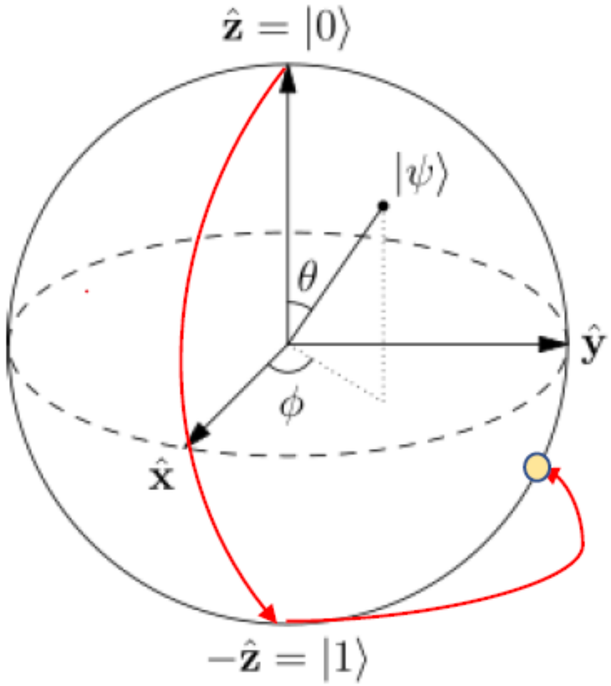
Materials at: https://torchquantum.org

IFTHANLab

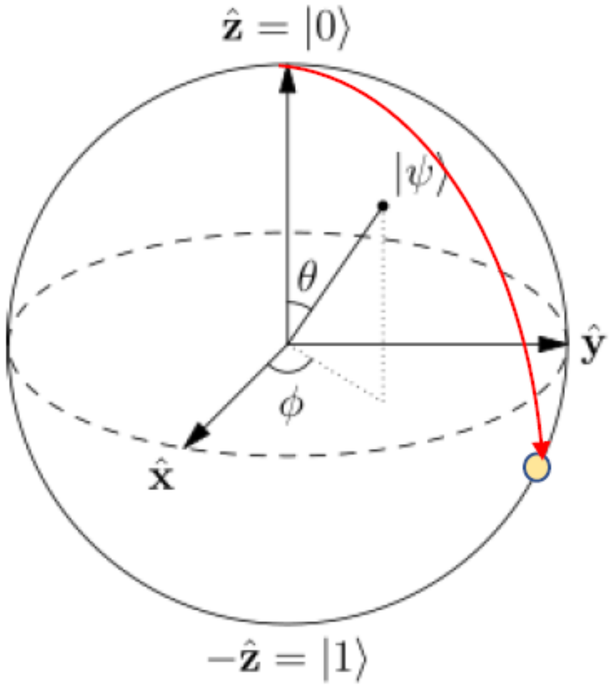
2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Motivation and Challenges

- Gate level optimization generate high redundancy.



a)



b)

Torch Quantum

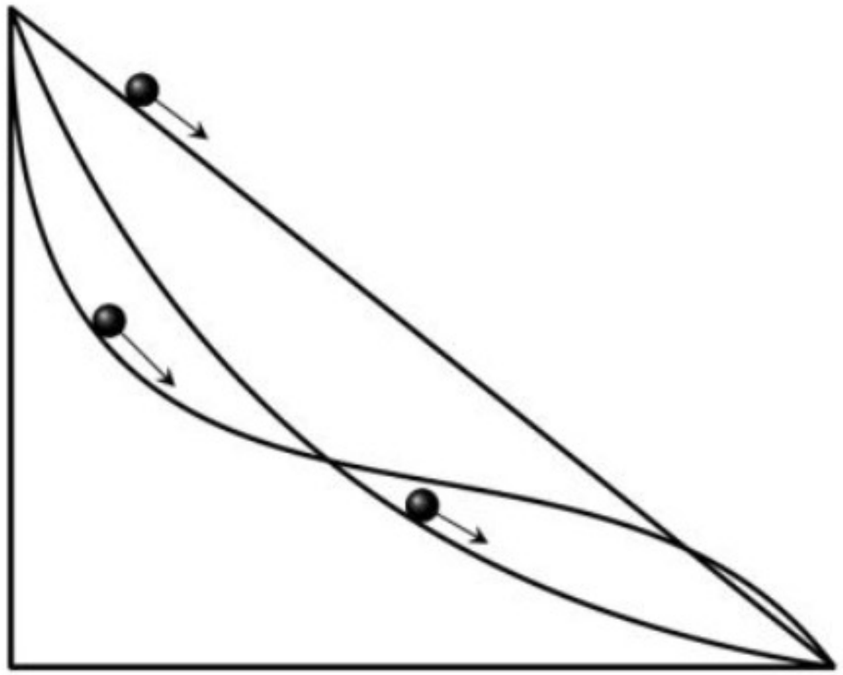
Materials at: https://torchquantum.org

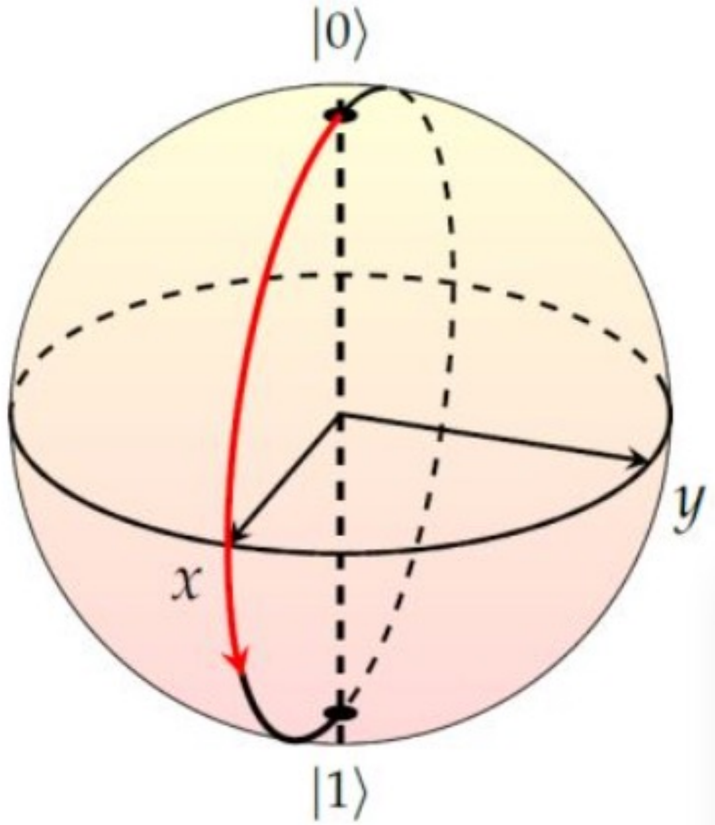
MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Motivation and Challenges

- QOC is limited to few qubits since it is computationally expensive.
- Most works demonstrate QOC on quantum simulators, however, it is hard to be evaluated on NISQ machines.





Torch Quantum

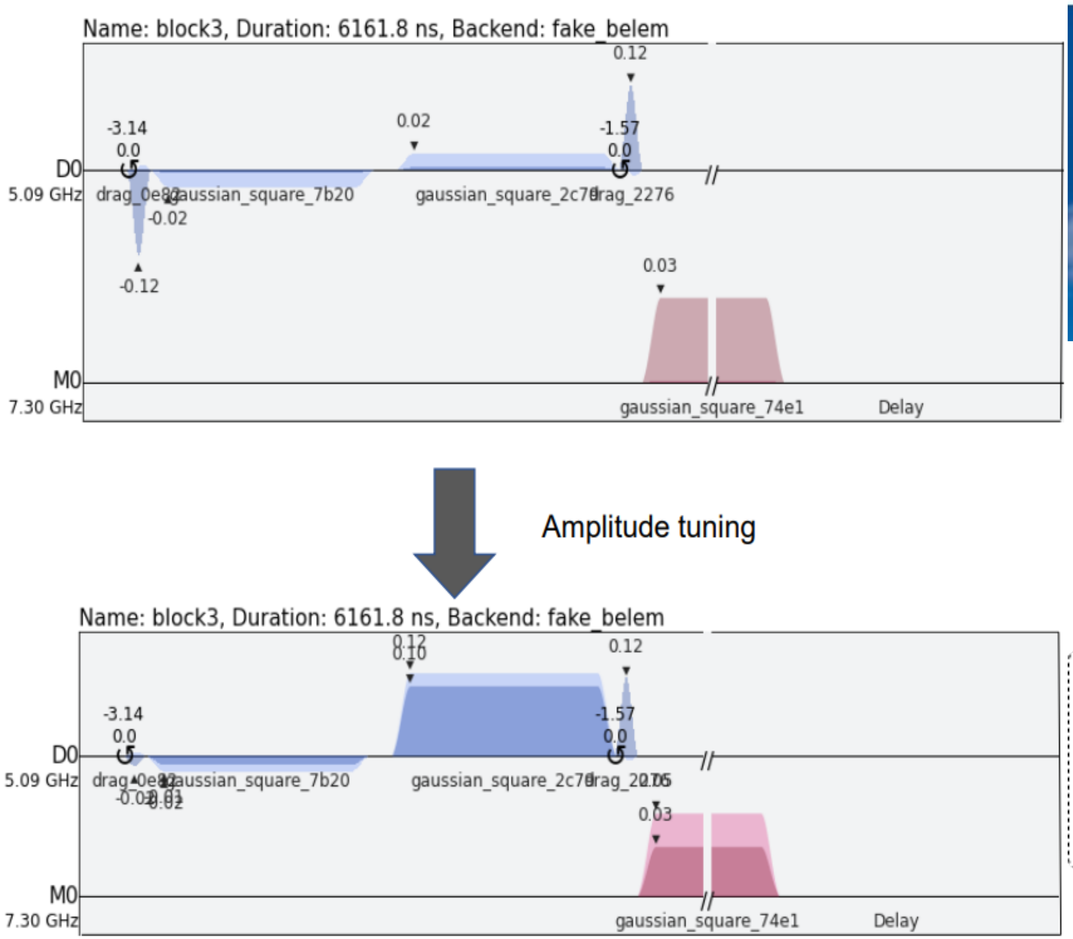
Materials at: https://torchquantum.org

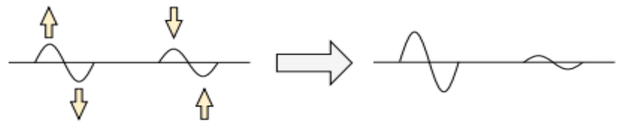
MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Attempt on Quantum Neural Network

- Can we find an intermediate-level approach between the gate level operations and quantum optimal control?
- Can v





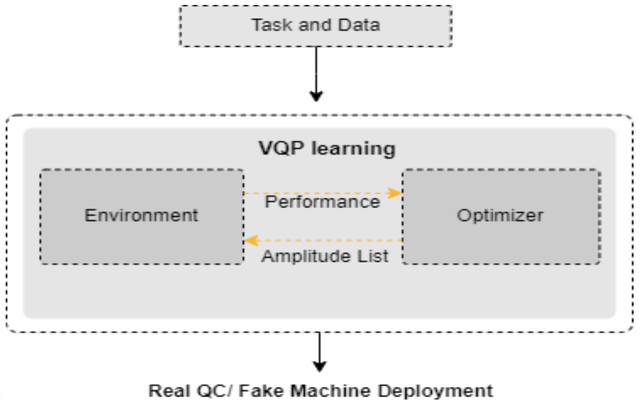


Fig. 2: An example QNN that uses VQC for QML tasks.

Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Methodology

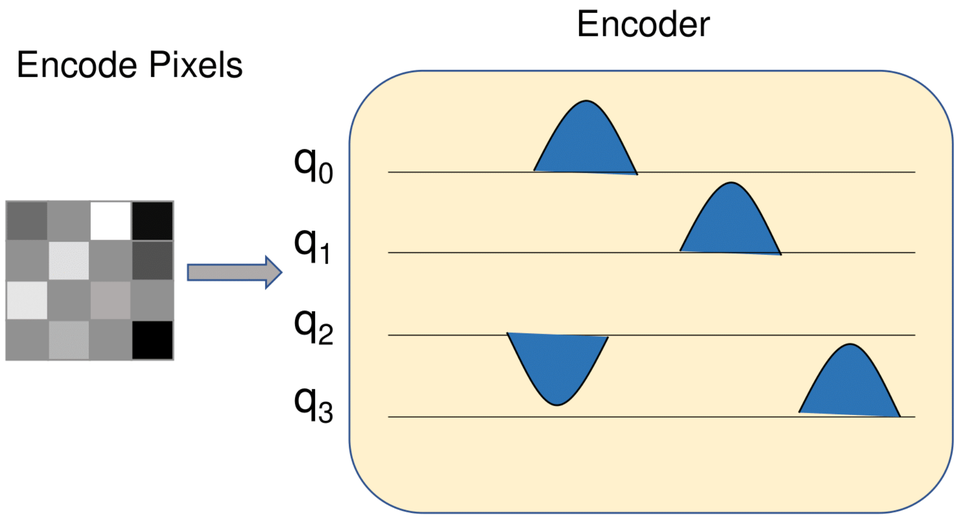
1. Enable a Novel Encoder on Pulse-level

2. Intermediate Layer Pulse Learning

3. Noise-Robustness Pulse Learning

Enable a Novel Encoder on Pulse-level

- Encode digital numbers to parameters on pulse.
- Future perspective: If we can achieve efficient encoding on pulse level?



Torch Quantum

Methodology

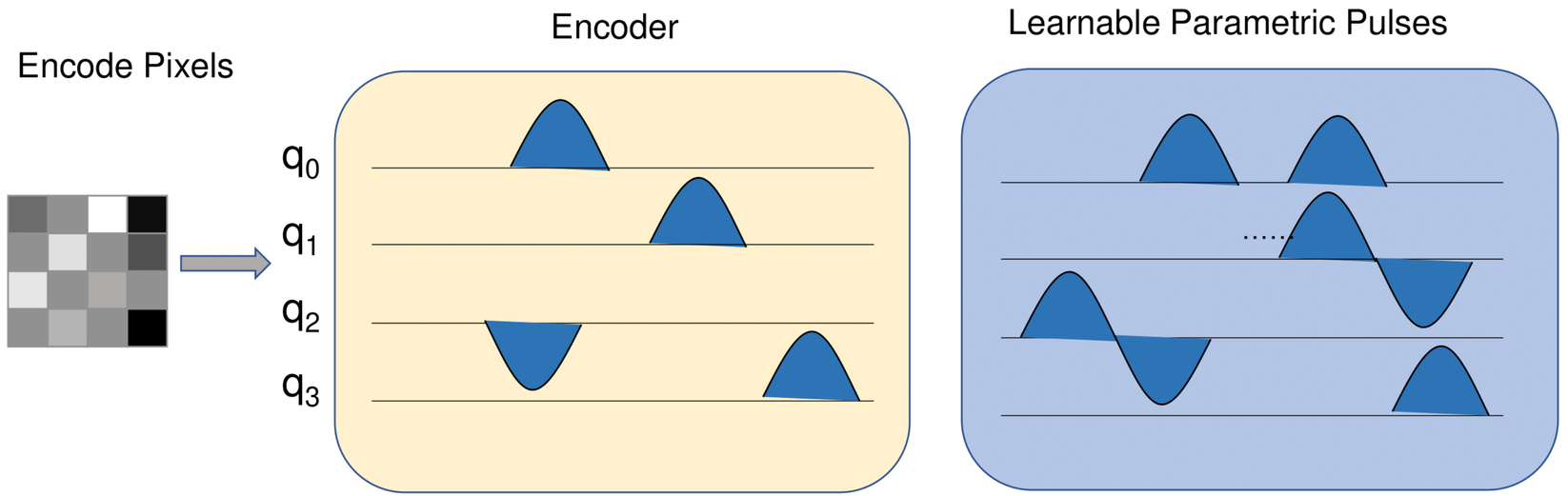
1. Enable a Novel Encoder on Pulse-level

2. Intermediate Layer Pulse Learning

3. Noise-Robustness Pulse Learning

Intermediate Layer Pulse Learning

- An intermediate level pulse learning between quantum optimal control and gate optimization scheme with good tradeoff between performance and training cost



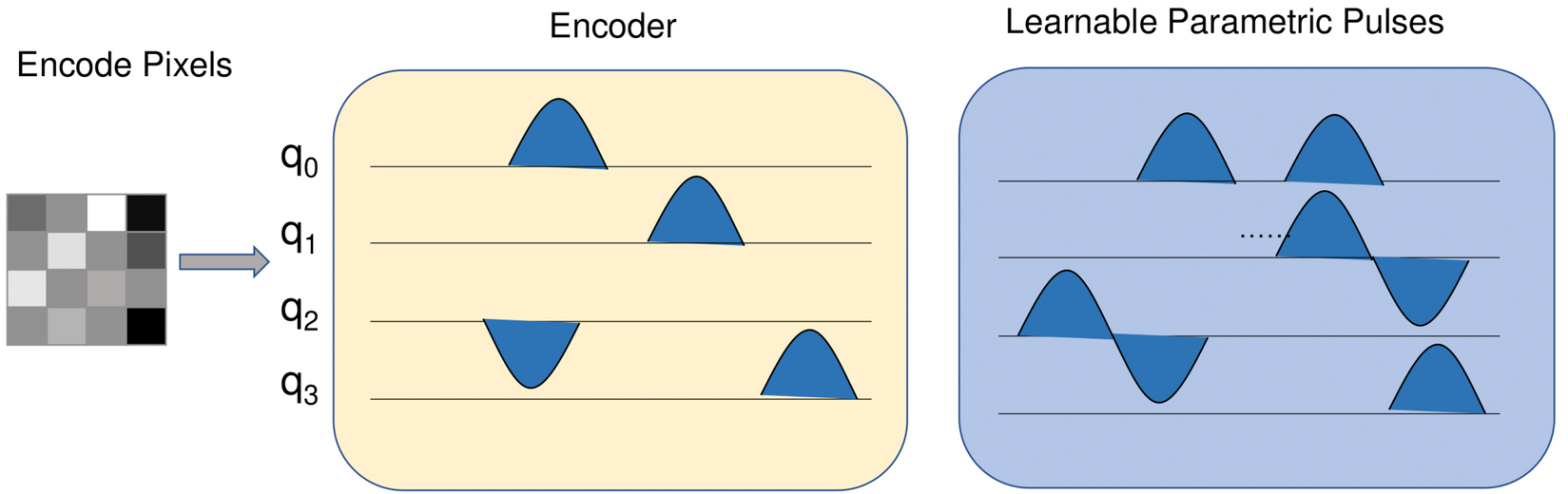
Torch Quantum

LAIb

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Intermediate Layer Pulse Learning

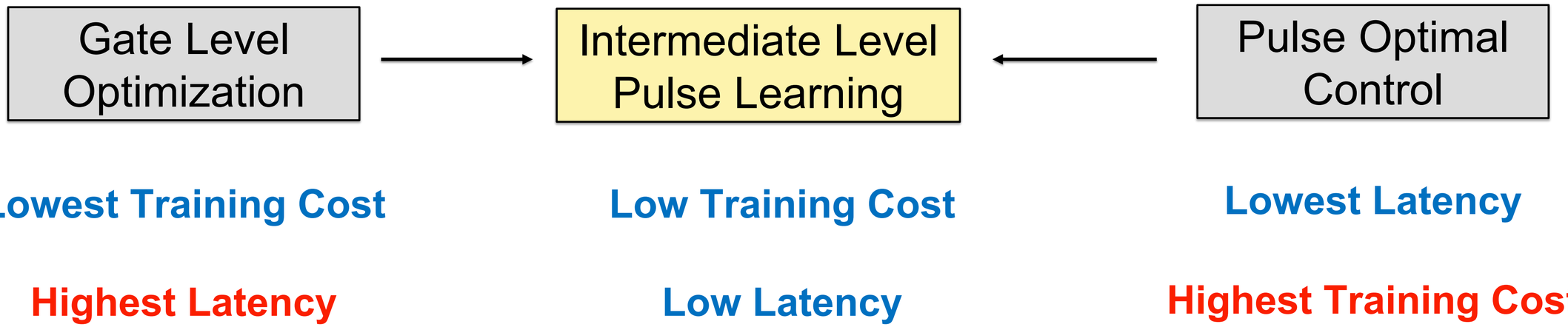
- An intermediate level pulse learning between quantum optimal control and gate optimization scheme with good tradeoff between performance and training cost



Torch Quantum

Intermediate Layer Pulse Learning

- An intermediate level pulse learning between quantum optimal control and gate optimization scheme with good tradeoff between performance and training cost



Methodology

1. Enable a Novel Encoder on Pulse-level

2. Intermediate Layer Pulse Learning

3. Noise-Robustness Pulse Learning

Optimization Framework

Algorithm 1: Variational Pulse BO Learning

Data: ρ, χ, M, D

// ρ is the amplitude list, χ is the search bound, D consists of x_i and y_i, M is the Gaussian Process Regression model.

$D \leftarrow$InitPulses(ρ, χ);

for $i \leftarrow |D|$ *to* N_total **do**

  // iterative optimization

  $p(y|x, D) \leftarrow FitModel(M, D)$;

  // Acquisition function actively searches for the next optimized amplitudes.

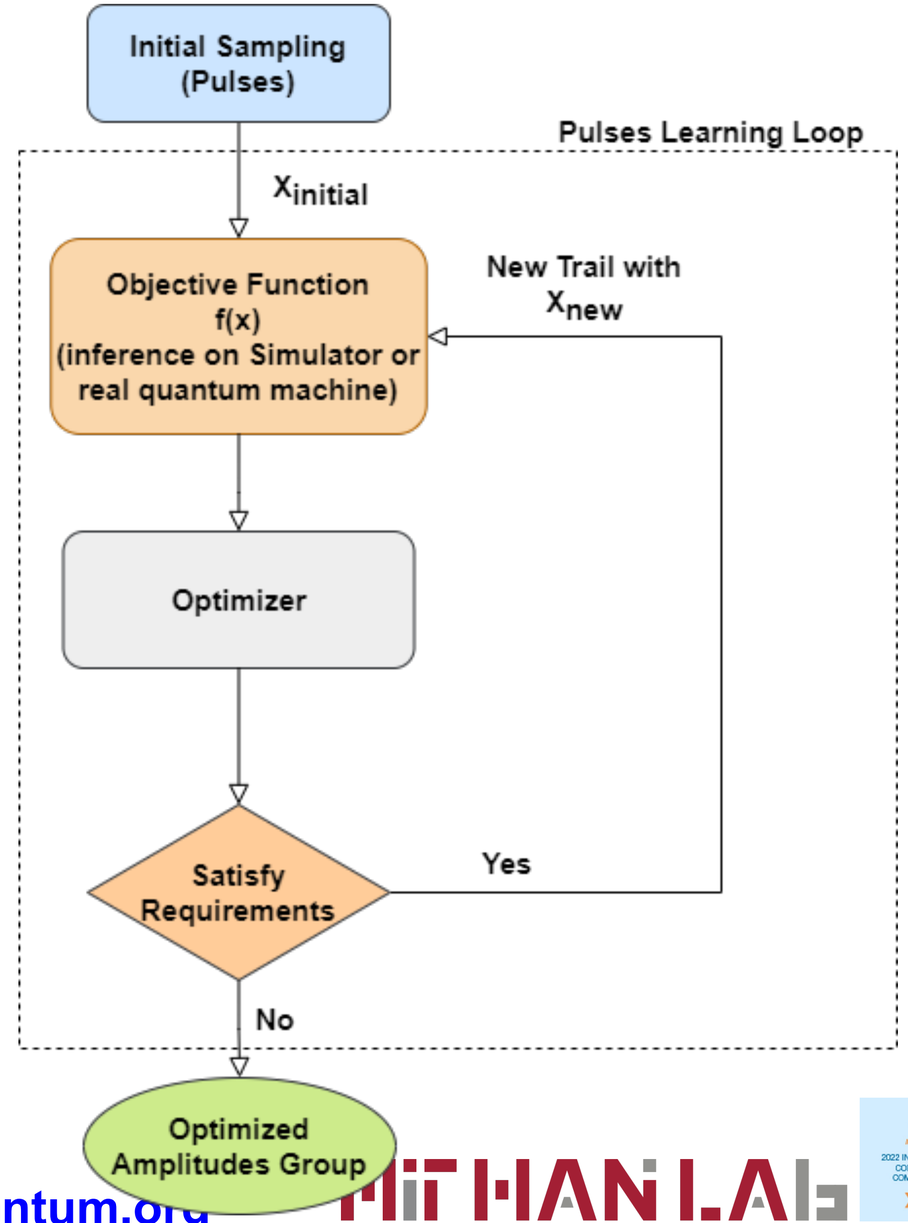
  $x_i \leftarrow argmin_{x \in \chi} S(x, p(y|x, D))$;

  // Calculate corresponding error rate by processing in quantum machine.

  $y_i \leftarrow f(x_i)$;

  $D \leftarrow D \bigcup (x_i, y_i)$;

end



Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Pulse based is better than Gate based

- Pulse based method has higher accuracy than gate based with same depth/latency.

Model	# of Gates	Accuracy
Gate Based	9	0.62
Pulse Based	**9**	**0.71**
Gate Based	12	0.68

Benefits Observed from VQP

Form of CX Gate	Noise Simulator (Quito)	Noise Simulator (Belem)	Noise Simulator (Jakarta)
CRX(pi) gate	26832.0dt	32016.0dt	26832.0dt
CX gate	25136.0dt	27728.0dt	25136.0dt

Advantage in specific gate

Model	# of Gate	Time Duration	
		Ibmq_Jakarta	Noise Simulator (Belem)
VQP	**9**	**40816.0dt**	**45168.0dt**
VQC*	12	58896.0dt	58768.0dt
VQP_transpiled	**11**	**32368.0dt**	**32816.0dt**
VQC*_transpiled	17	53008.0dt	46192.0dt

Advantage in general circuit

Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Challenge for Pulse Learning

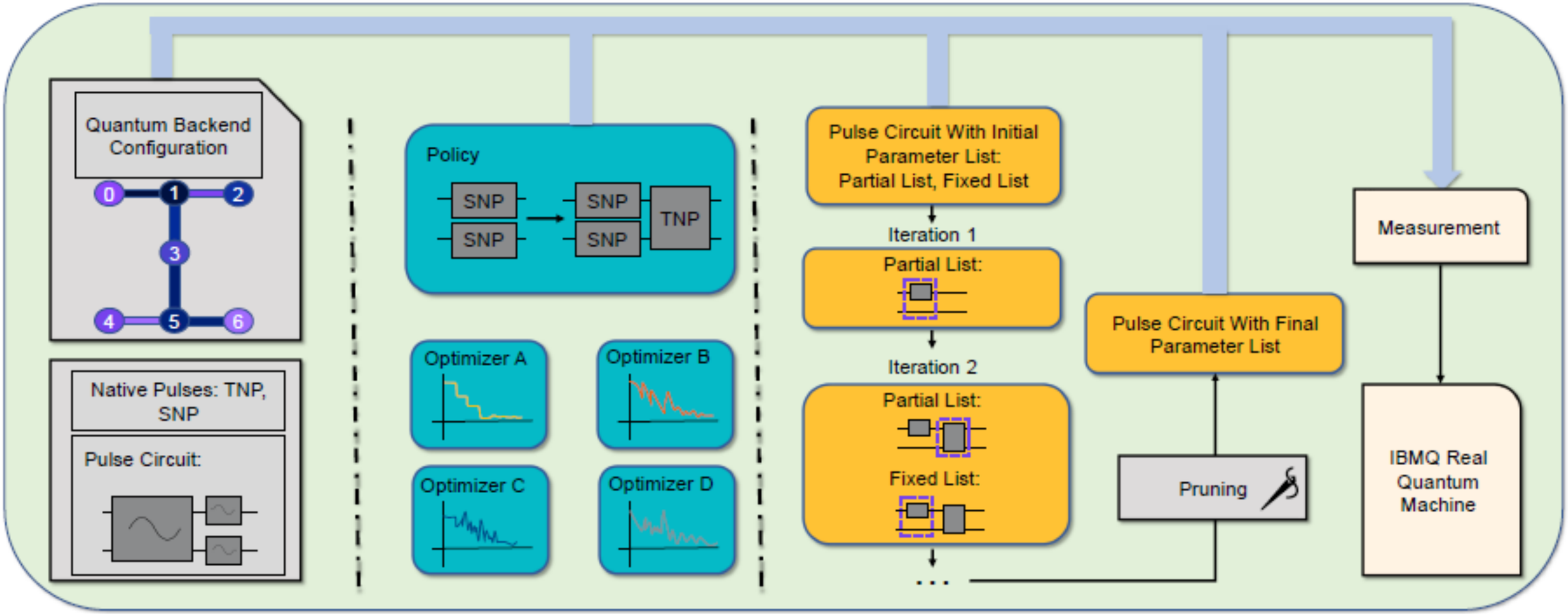
- Non-gradient-based optimizer has randomness when parameter in high dimensional space.
- Qiskit pulse simulator is not efficient, e.g., need around 3 mins to finish a 9-gate circuit.

Model	# of Gates	Accuracy
Gate Based	9	0.62
Pulse Based	**9**	**0.71**
Gate Based	12	0.68

Model	# of Gates	Accuracy
VQP	9	0.71
VQC with gradient	9	0.73
VQC* with gradient	12	0.77

Pulse Ansatz on VQAs

- Can we find a hardware efficient ansatz at the pulse level?
- What is a good pulse ansatz?

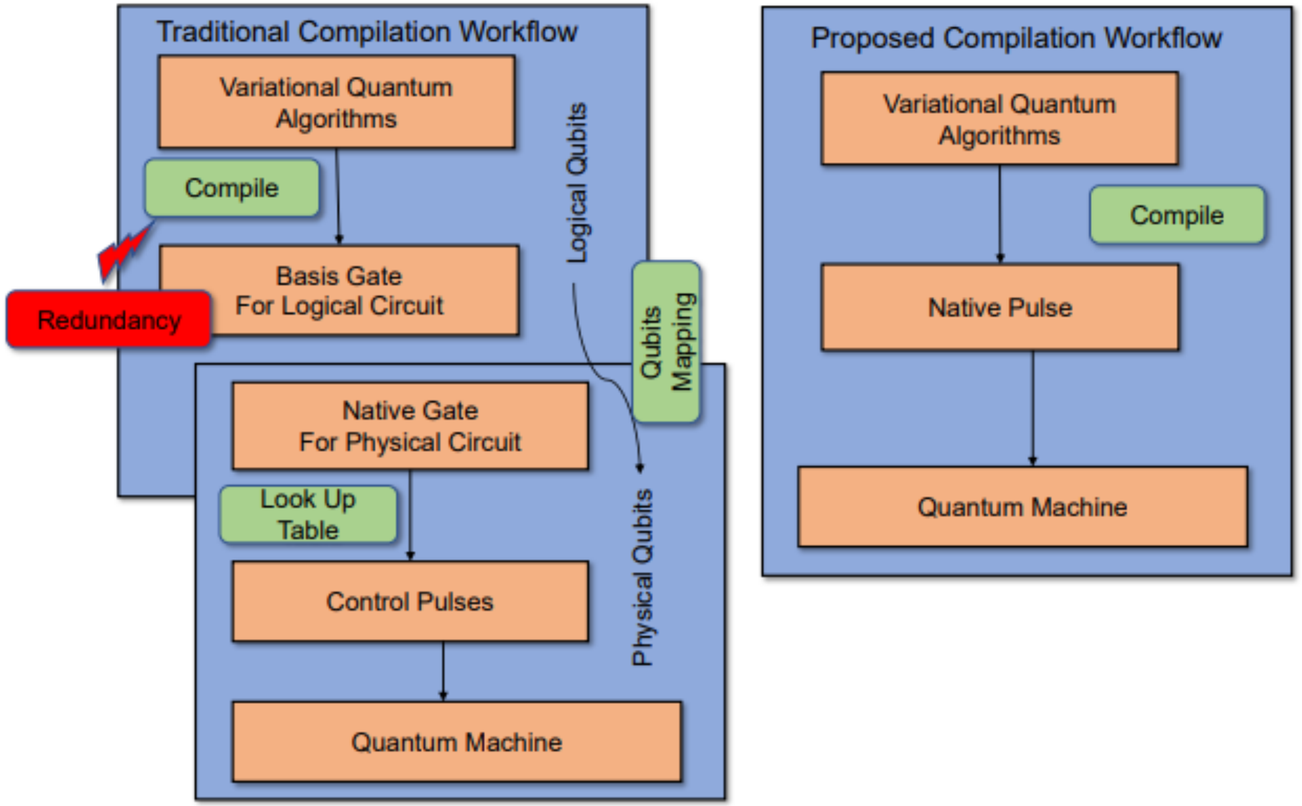


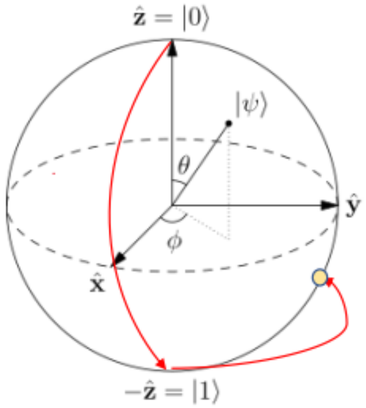
Torch Quantum

Materials at: https://torchquantum.org

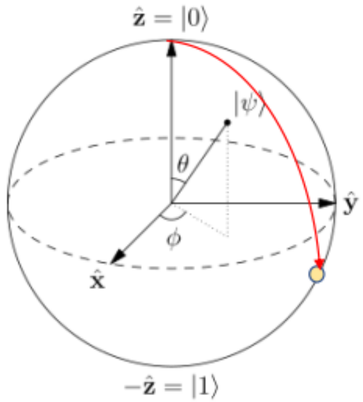
Benefits of Pulse Ansatz

- Source of Advantages:
- Enable flexible and efficient compilation workflow on pulse-level.





a)



b)

Torch Quantum

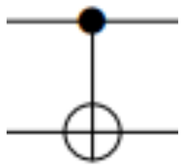
Materials at: https://torchquantum.org

Benefits of Pulse Ansatz

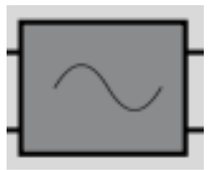
- Source of Advantages:
- Two-qubit pulse is tunable, whereas two-qubit gates have few flexibility.

COMPARISON OF TRAINABILITY FOR DIFFERENT PULSE CIRCUITS AND GATE CIRCUITS ON IBMQ_JAKARTA.

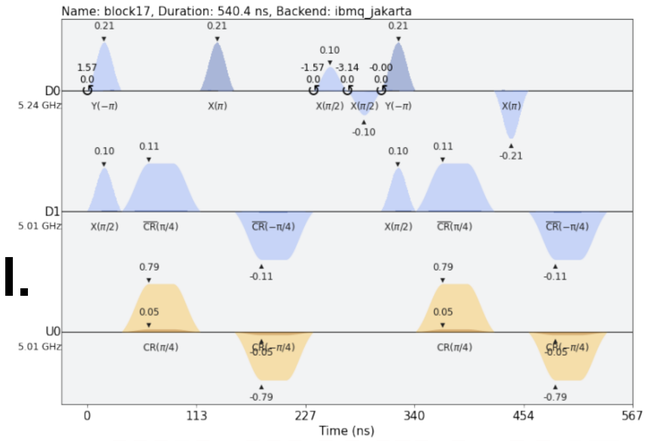
Operations	Circuit Level	Molecule Bond Length	Reference Energy	VQE (H_2) Result	Duration(on ibmq_jakarta)
SNP	Pulse Circuit	0.1Å	2.710H	4.380H	71.1ns
TNP	Pulse Circuit	0.1Å	2.710H	2.927H	163.6ns
SNP	Pulse Circuit	0.75Å	-1.137H	-0.549H	71.1ns
TNP	Pulse Circuit	0.75Å	-1.137H	-1.032H	163.6ns
TNP + SNP	Pulse Circuit	0.75Å	-1.137H	-1.036H	234.7ns
Two Gate Ansatz	Gate Circuit	0.75Å	-1.137H	-0.534H	341.3ns



✗**Nothing can do with two qubits gate**



✓ **Parameters are tunable on control channel.**

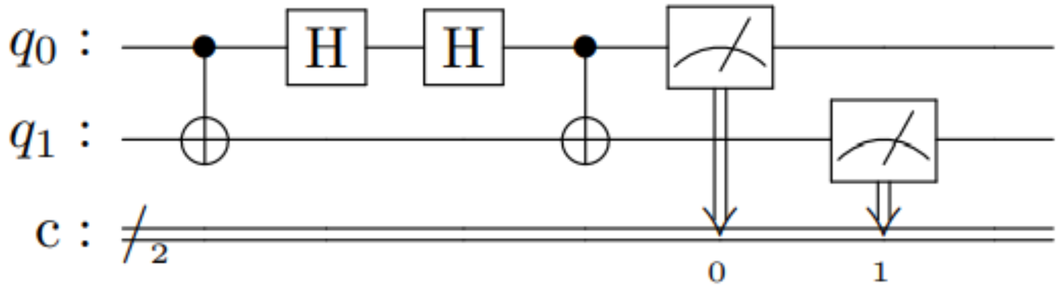


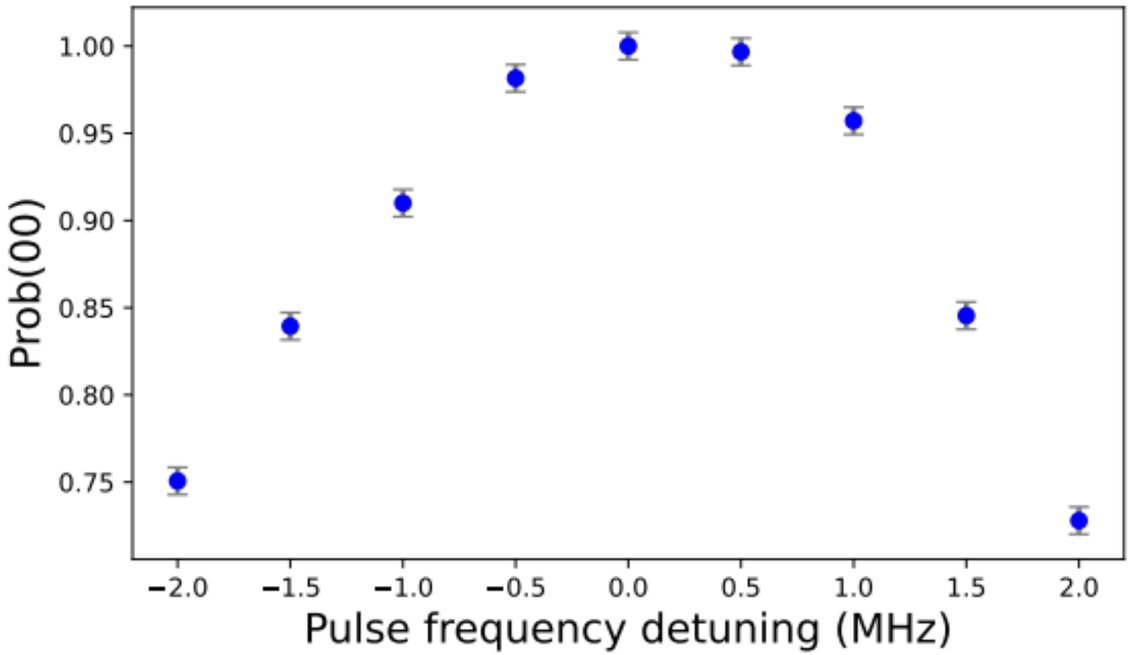
Torch Quantum

Materials at: https://torchquantum.org

Benefits of Pulse Ansatz

- Source of Advantages:
- Capability to tune frequency on pulse-level





Torch Quantum

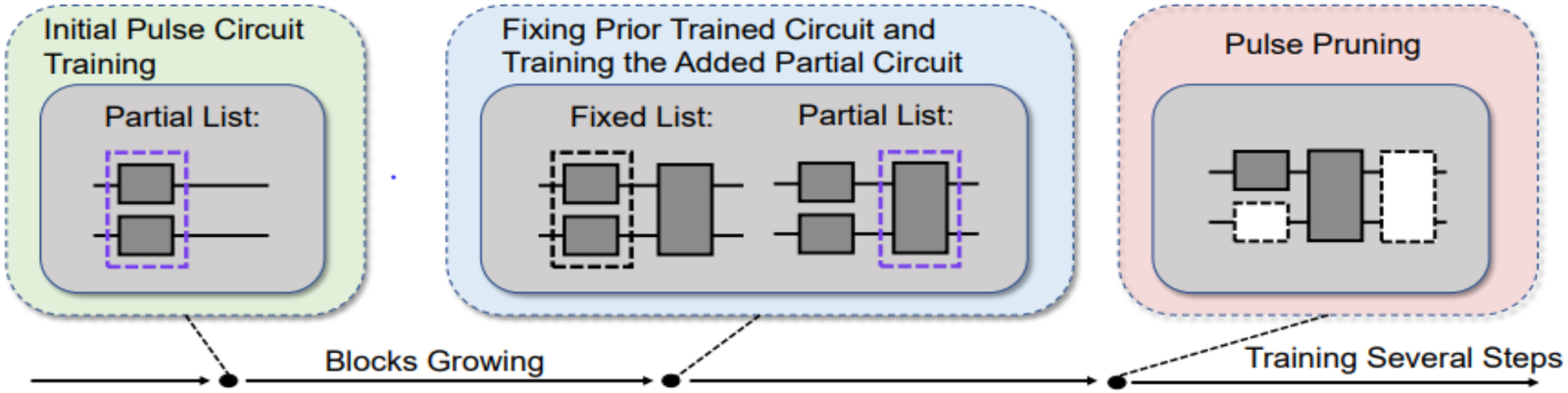
Materials at: https://torchquantum.org

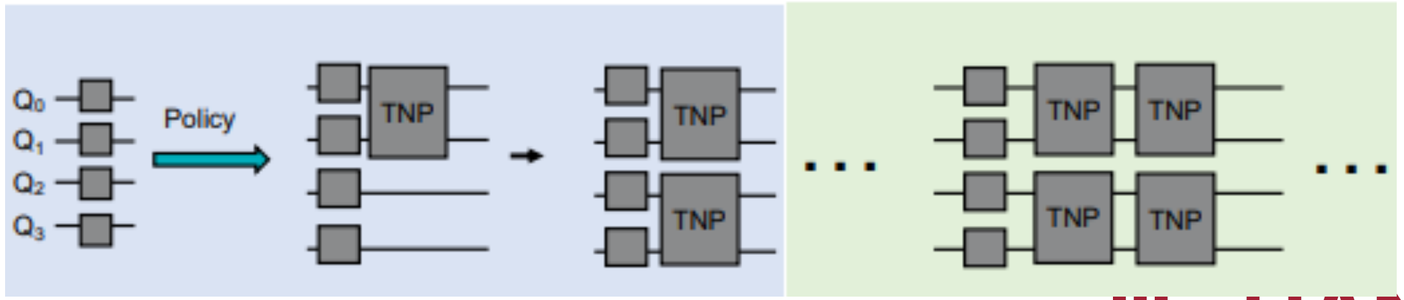
MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Framework and Evaluation

- Progressive learning fix the problem that non gradient optimizer cannot hold high dimensional parameters, and progressive approaching to target point is also fit the quantum speed limit (QSL) theory.





Torch Quantum

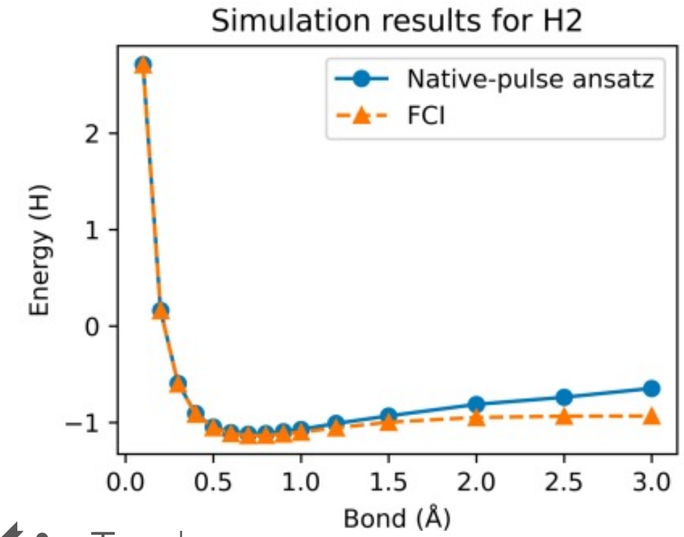
Materials at: https://torchquantum.org

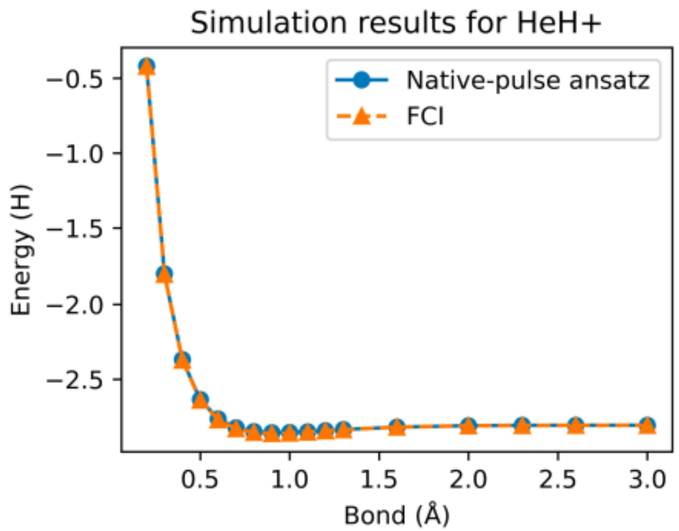
MIT HAN LAB

Framework and Evaluation

COMPARISON OF DURATION, PULSE COUNTS, AND ESTIMATED ENERGY OF GATE ANSATZES AND THE NATIVE-PULSE ANSATZ GENERATED BY PAN ON NISQ MACHINES.

Model	Ansatz Level	Qubits	Duration	Single-Qubit Pulse Count	Multi-Qubit Pulse Count	Molecule	Energy	Reference Energy
Random Genrated Ansatz	Gate Ansatz	2	682.7ns	16	2	H_2	-0.853	-1.137
RealAmplitude Ansatz [2]	Gate Ansatz	2	376.9ns	12	1	H_2	-0.974	-1.137
QuantumNAS [75]	Gate Ansatz	2	682.7ns	16	2	H_2	-1.033	-1.137
PAN	Pulse Ansatz	2	71.1ns	3	0	H_2	-1.100	-1.137
RealAmplitude Ansatz	Gate Ansatz	2	753.8ns	24	2	$HeH+$	-2.691	-2.863
PAN	Pulse Ansatz	2	199.1ns	1	1	$HeH+$	-2.866	-2.863
QuantumNAS	Gate Ansatz	6	7296.0ns	40	12	LiH	-6.914	-7.882
PAN	Pulse Ansatz	4	199.1ns	4	2	LiH	-7.590	-7.882





97.3% reduction in ansatz duration compared to QuantumNAS.

Reduce duration by 73.6% compared to RealAmplitude Ansatz while maintaining ansatz performance.

Torch Quantum

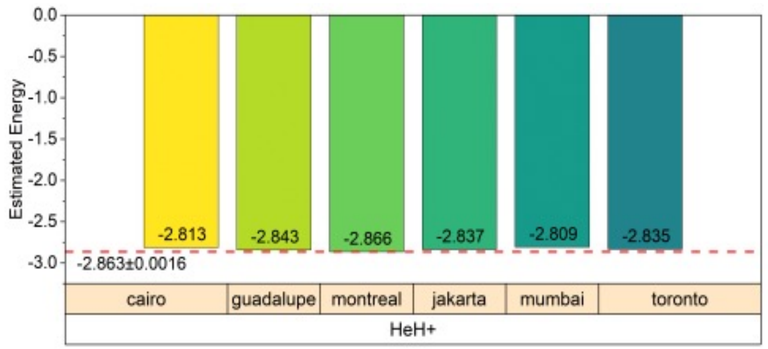
Materials at: https://torchquantum.org

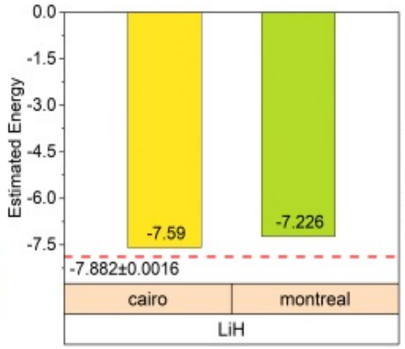
MIT HAN LAB

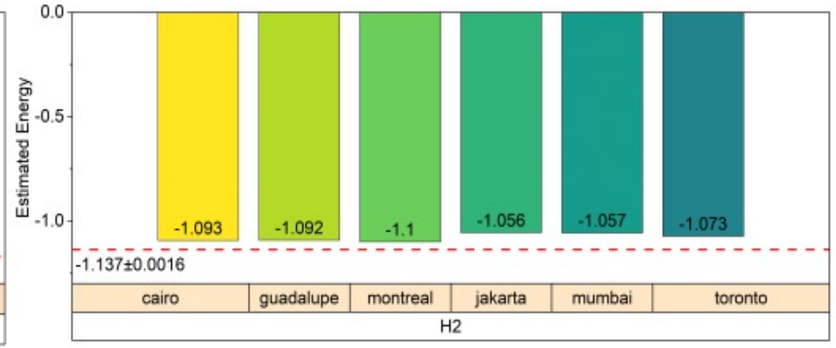
2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

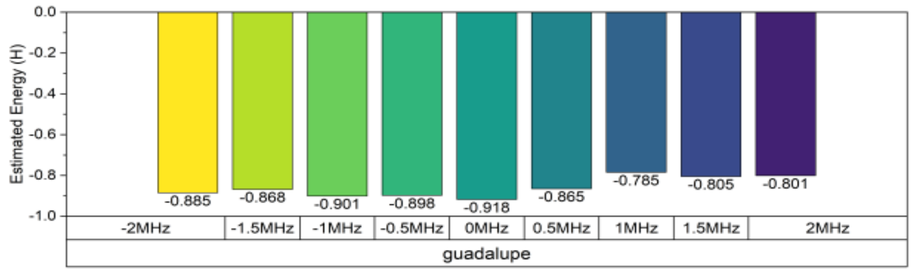
Framework and Evaluation

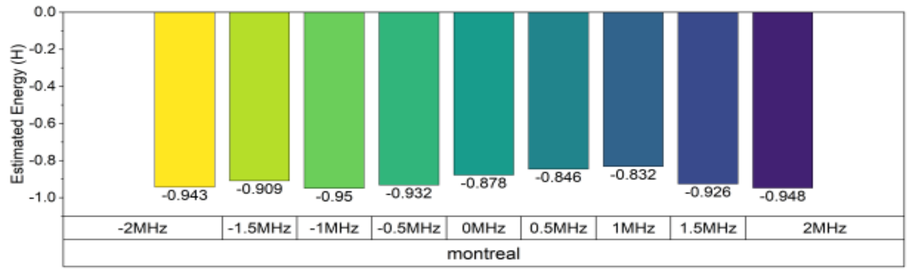
- **HeH+**: average accuracy 99.336%, with 99.895% being the highest achievable accuracy. The absolute difference in energy is 0.003H, close to the requirement of computational chemistry error (0.0016H) with only a toy model.

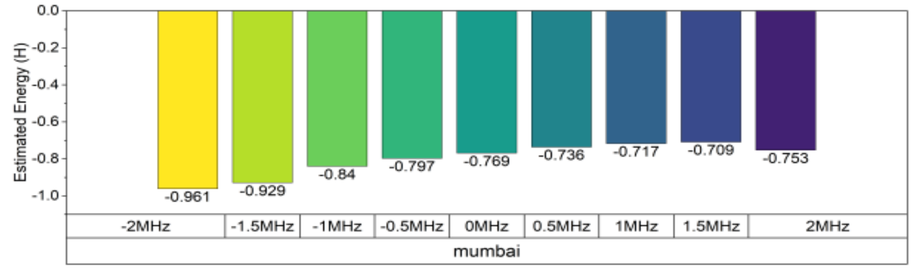


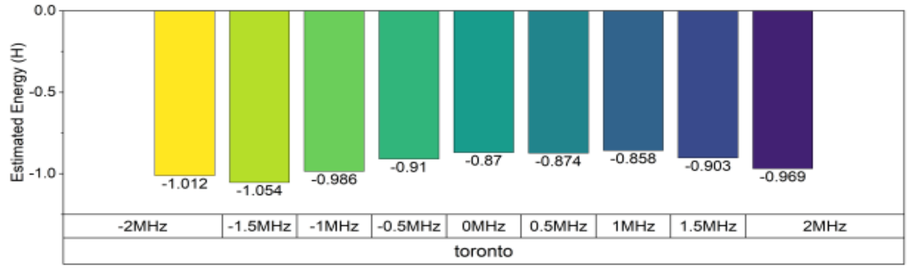












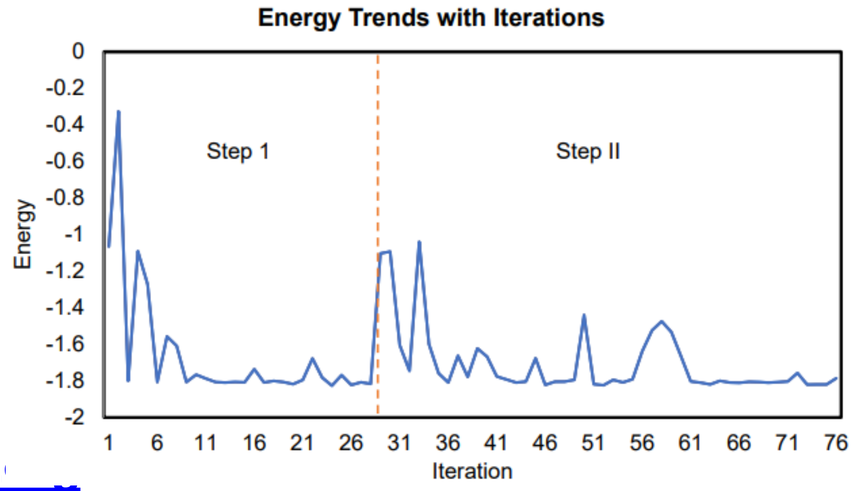
Torch Quantum

Framework and Evaluation

- Verify the effectiveness of progressive learning scheme.

RESULTS OF ESTIMATED ENERGY FOR MOLECULES IN DIFFERENT STEPS

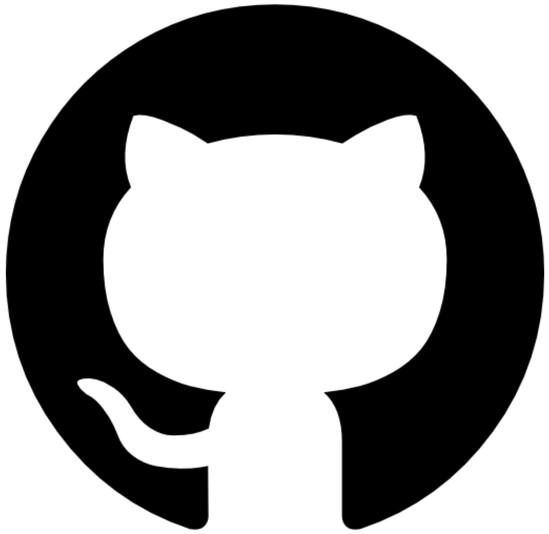
Model		Cairo	Montreal	Toronto	NISQ machine Avg	Simulator	FCI
$H2$	Step I	-1.093 (3.870%)	-1.087 (4.398%)	-1.073 (5.629%)	-1.084 (4.661%)	-1.121 (1.407%)	-1.137
	Step II	-1.107 (2.639%)	-1.110 (2.375%)	-1.073 (5.629%)	-1.097 (3.518%)	-1.123 (1.231%)	-1.137
	Inaccuracy Reduction	31.83%	46.00%	0.000%	24.52%	12.51%	-
$HeH+$	Step I	-2.813 (1.746%)	-2.845 (0.663%)	-2.820 (1.485%)	-2.826 (1.292%)	-2.855 (0.279%)	-2.863
	Step II	-2.833 (1.047%)	-2.866 (0.105%)	-2.834 (1.013%)	-2.844 (0.664%)	-2.856 (0.244%)	-2.863
	Inaccuracy Reduction	40.03%	84.16%	31.78%	48.61%	12.54%	-



Torch Quantum

Materials at: https://torchquantum.

Hands-On Section
2.2 Variational Pulse Learning



Torch Quantum

Materials at: https

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations
- 1.3 TQ Use Examples
- 1.4 QNN Compression

Section 2

Use TorchQuantum on Pulse Level Optimization

- 2.1 Quantum Optimal Control
- 2.2 Variational Pulse Learning

Section 3

Use TorchQuantum on Gate Level Optimization

- 3.1 QuantumNAS: Ansatz Search and Gate Pruning
- 3.2 QuantumNAT: Noise Injection and Quantization
- 3.3 QOC: On-Chip Training
- 3.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

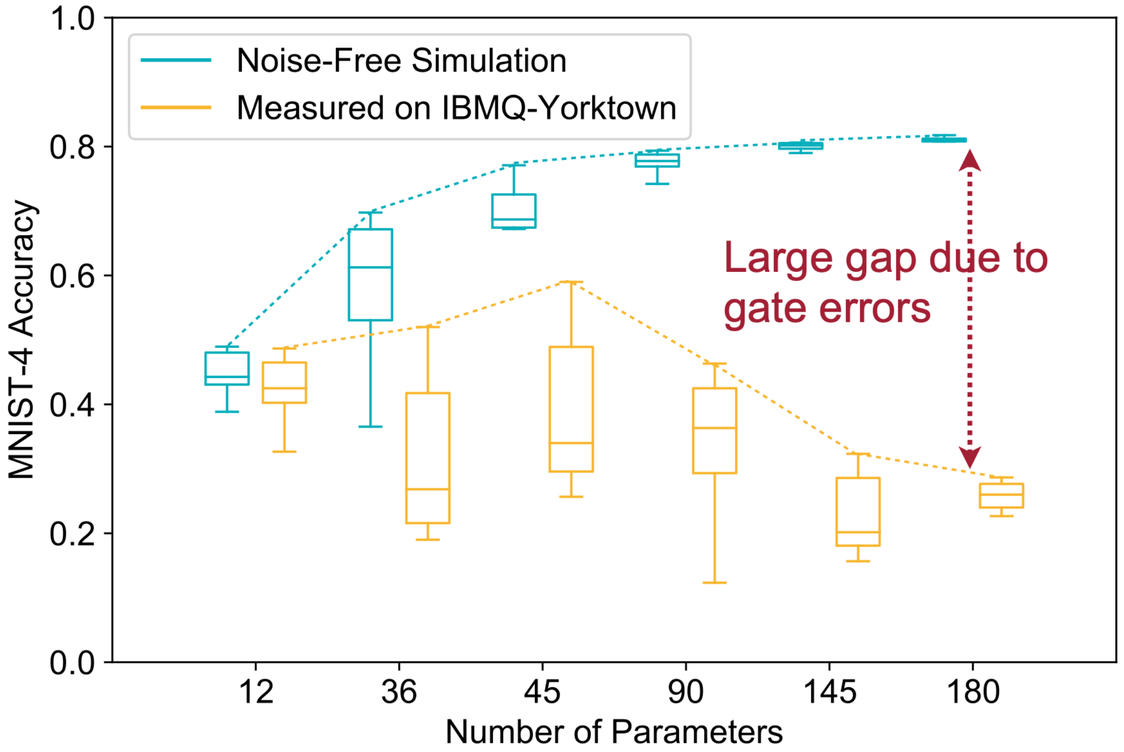
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Challenges of PQC — Noise

- Noise **degrades** PQC reliability
- More parameters increase the noise-free accuracy but degrade the measured accuracy
- Therefore, circuit architecture is critical



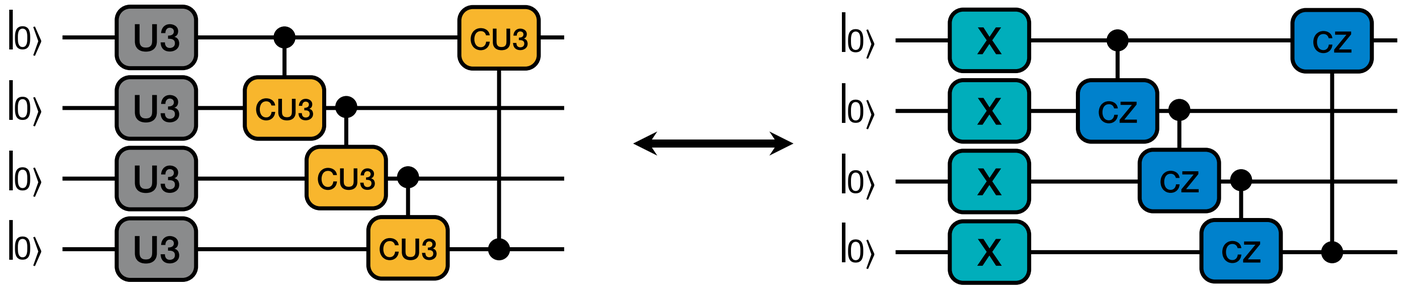
Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

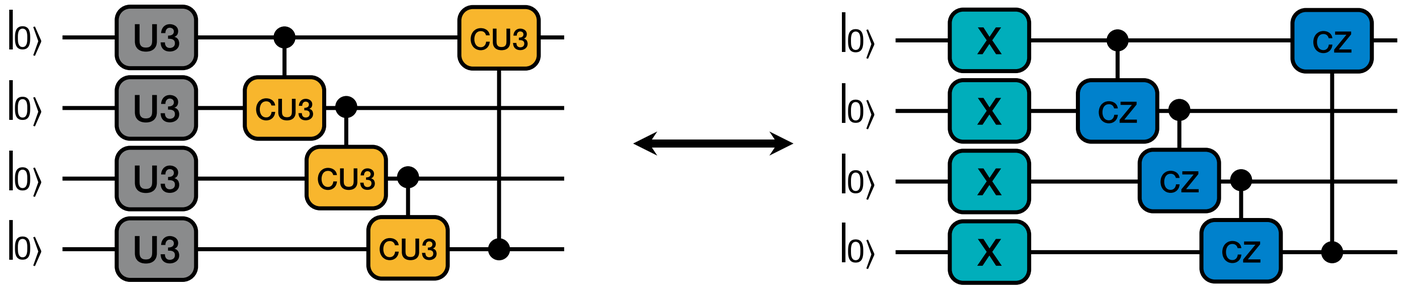
Challenges of PQC — Large Design Space

- Large design space for circuit architecture
  - Type of gates

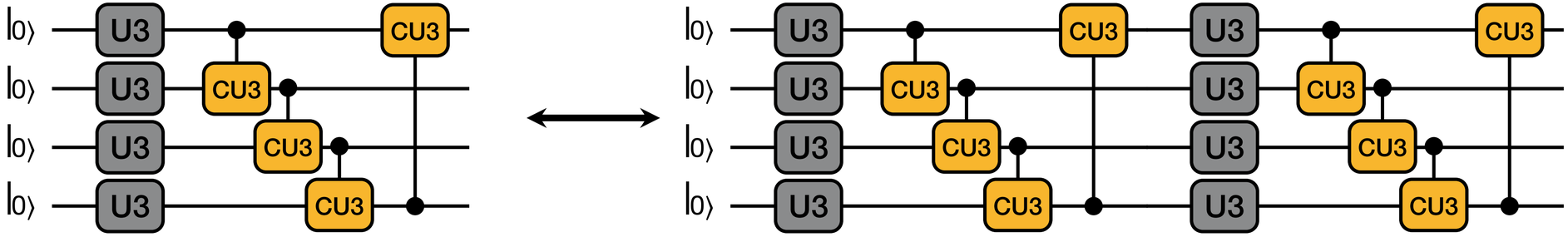


Challenges of PQC — Large Design Space

- Large design space for circuit architecture
  - Type of gates

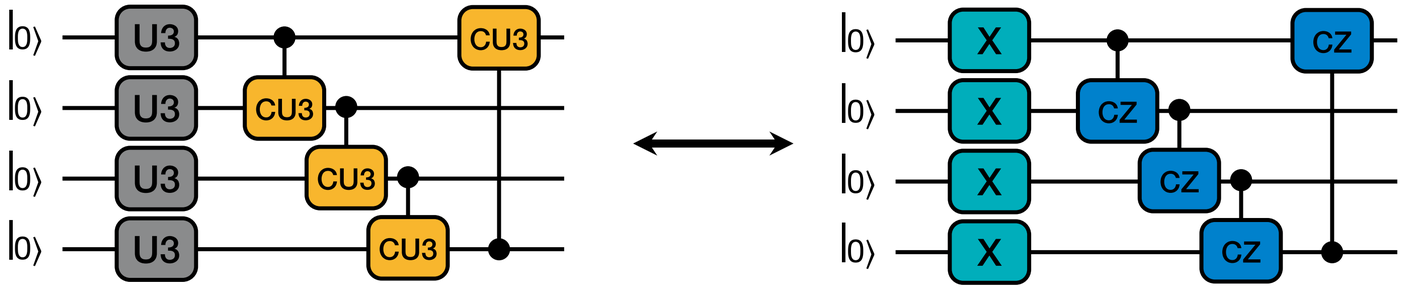


  - Number of gates

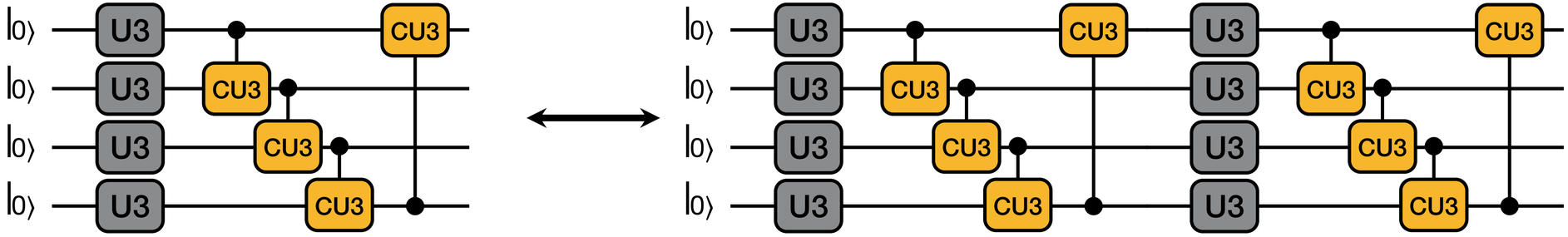


Challenges of PQC — Large Design Space

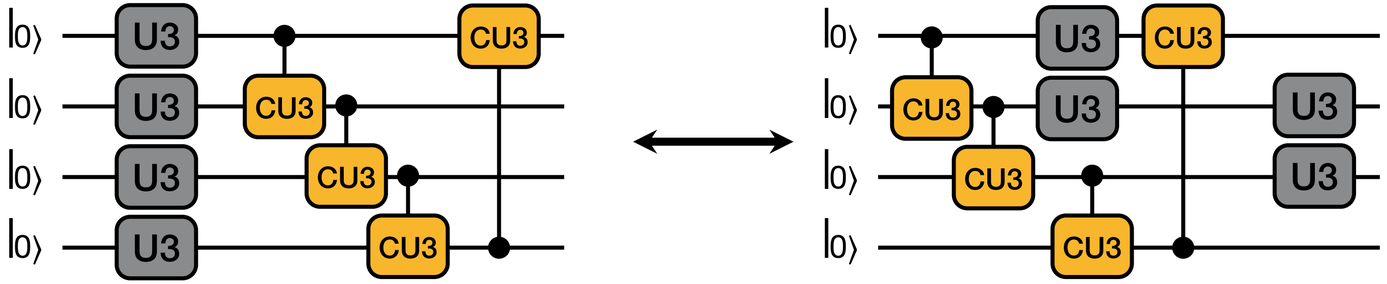
- Large design space for circuit architecture
  - Type of gates



  - Number of gates



  - Position of gates



Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Goal of QuantumNAS

Automatically & efficiently search for noise-robust quantum circuit

Train one “SuperCircuit”, providing parameters to many “SubCircuits”

Solve the challenge of large design space

(1) Quantum noise feedback in the search loop
(2) Co-search the circuit architecture and qubit mapping

Solve the challenge of large quantum noise

QuantumNAS

- SuperCircuit Construction and Training
- Noise-Adaptive Evolutionary Co-Search of SubCircuit and Qubit Mapping
- Train the Searched SubCircuit
- Iterative Quantum Gate Pruning

QuantumNAS

- SuperCircuit Construction and Training
- Noise-Adaptive Evolutionary Co-Search of SubCircuit and Qubit Mapping
- Train the Searched SubCircuit
- Iterative Quantum Gate Pruning

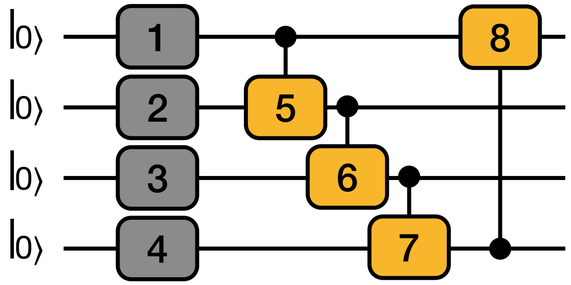
SuperCircuit & SubCircuit

- Firstly construct a design space. For example, a design space of maximum 4 U3 in the first layer and 4 CU3 gates in the second layer

SuperCircuit & SubCircuit

- Firstly construct a design space. For example, a design space of maximum 4 U3 in the first layer and 4 CU3 gates in the second layer

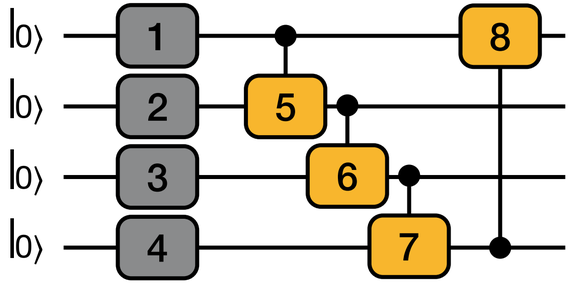
- SuperCircuit: the circuit with the **largest** number of gates in the design space
  - Example: SuperCircuit in U3+CU3 space



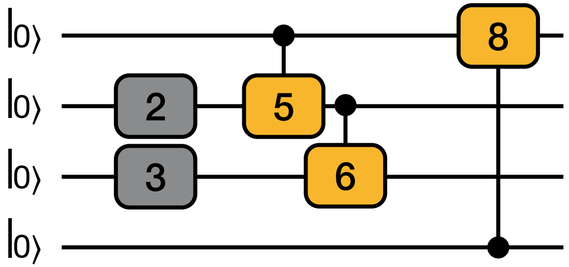
SuperCircuit & SubCircuit

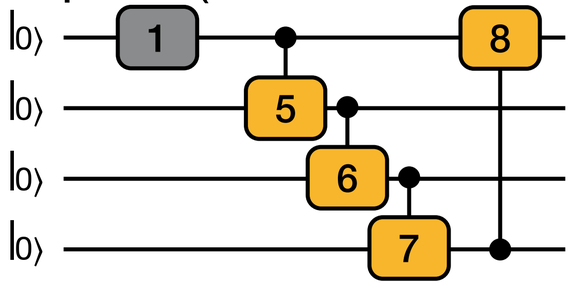
- Firstly construct a design space. For example, a design space of maximum 4 U3 in the first layer and 4 CU3 gates in the second layer

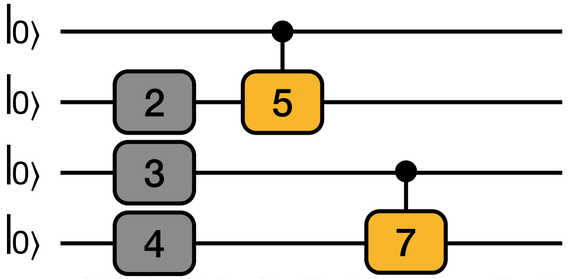
- SuperCircuit: the circuit with the **largest** number of gates in the design space
  - Example: SuperCircuit in U3+CU3 space



- Each candidate circuit in the design space (called SubCircuit) is a **subset** of the SuperCircuit







Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

SuperCircuit Construction

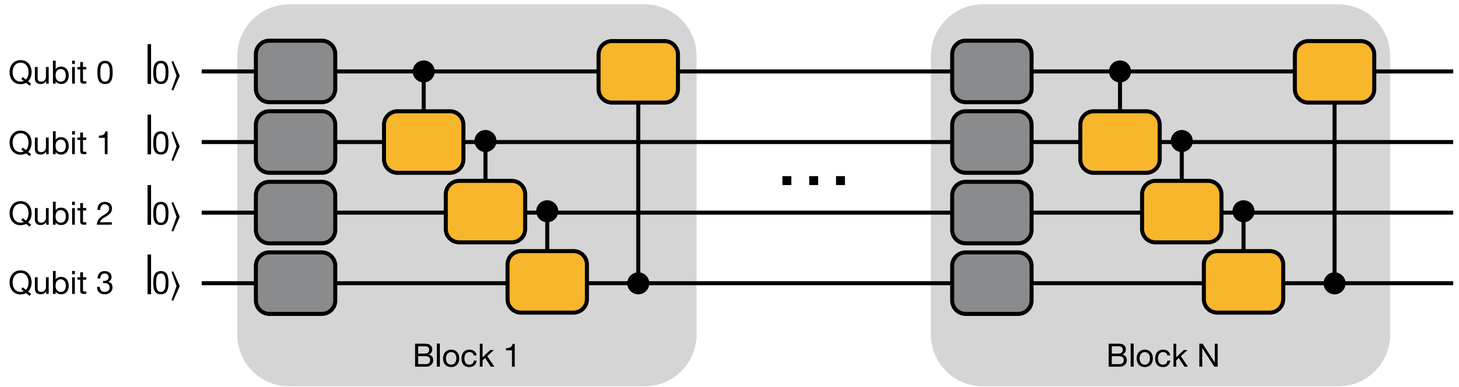
- Why use a SuperCircuit?
  - Enables **efficient** search of architecture candidates without training each
  - SubCircuit inherits parameters from SuperCircuit
  - With **inherited** parameters, we find some good SubCircuits, we find that they are **also good SubCircuits** with parameters **trained from-scratch** individually

SuperCircuit Training

- In one SuperCircuit Training step:
  - Sample a gate subset of SuperCircuit (a SubCircuit)
    - Front Sampling and Restricted Sampling
  - Only use the subset to perform the task and updates the parameters in the subset
  - Parameter updates are cumulative across steps

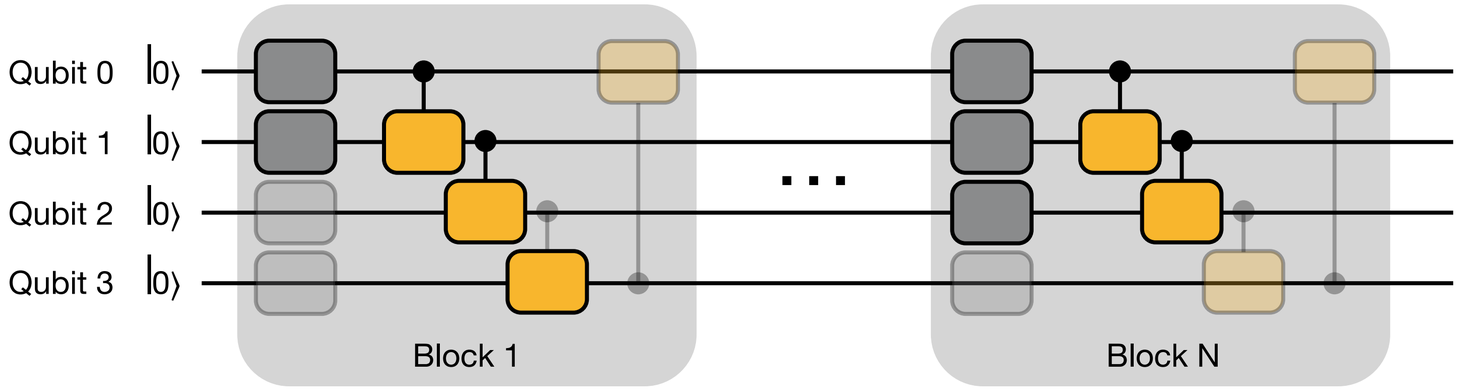
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



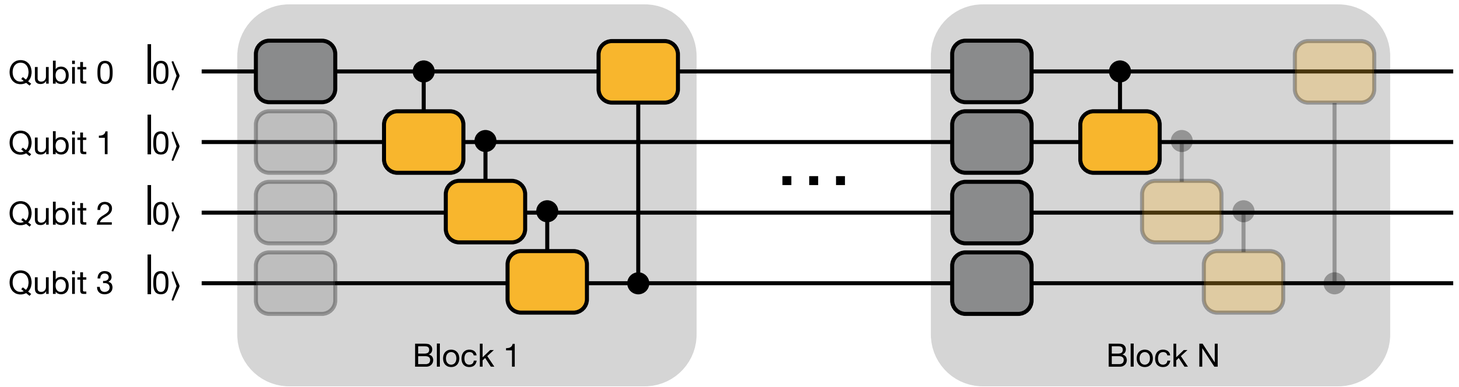
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



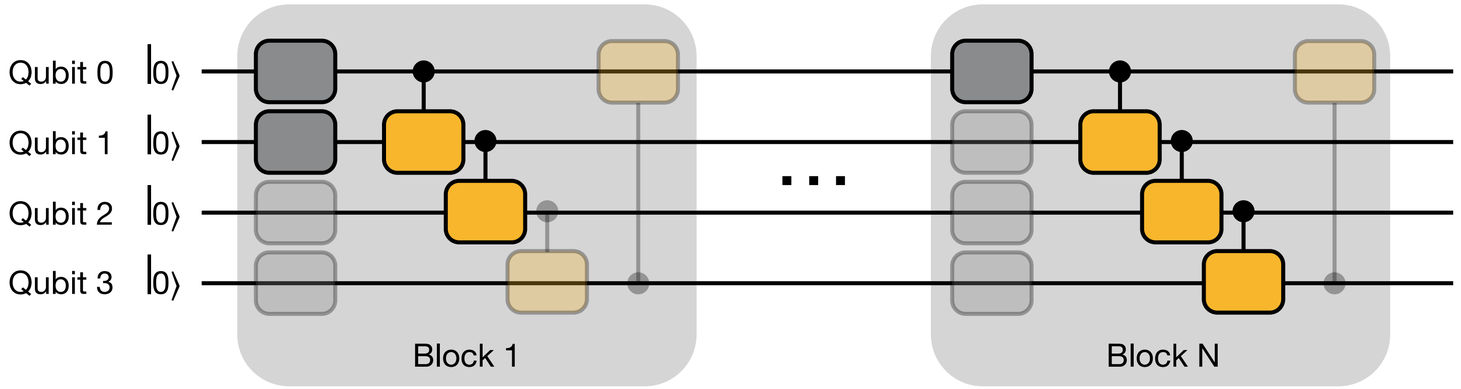
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



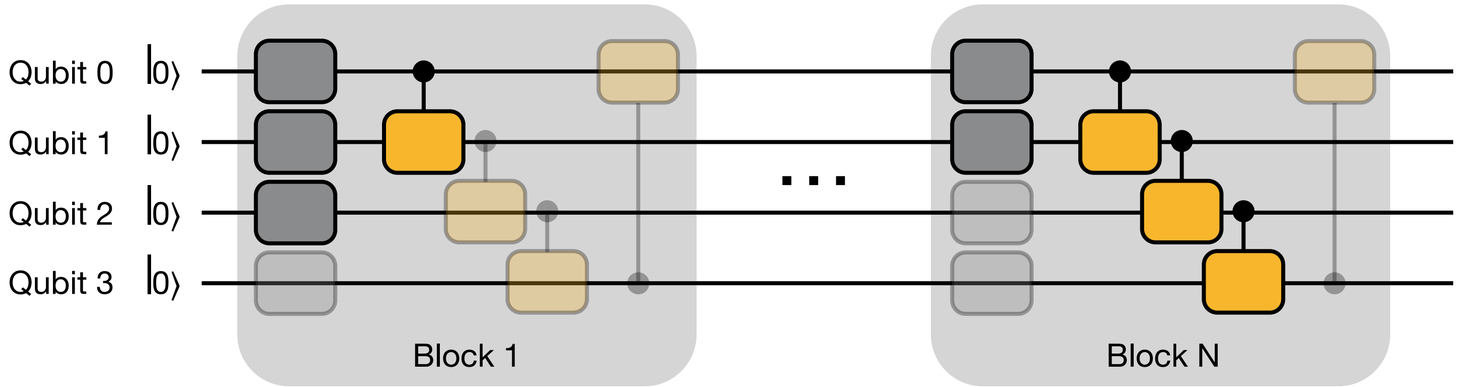
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



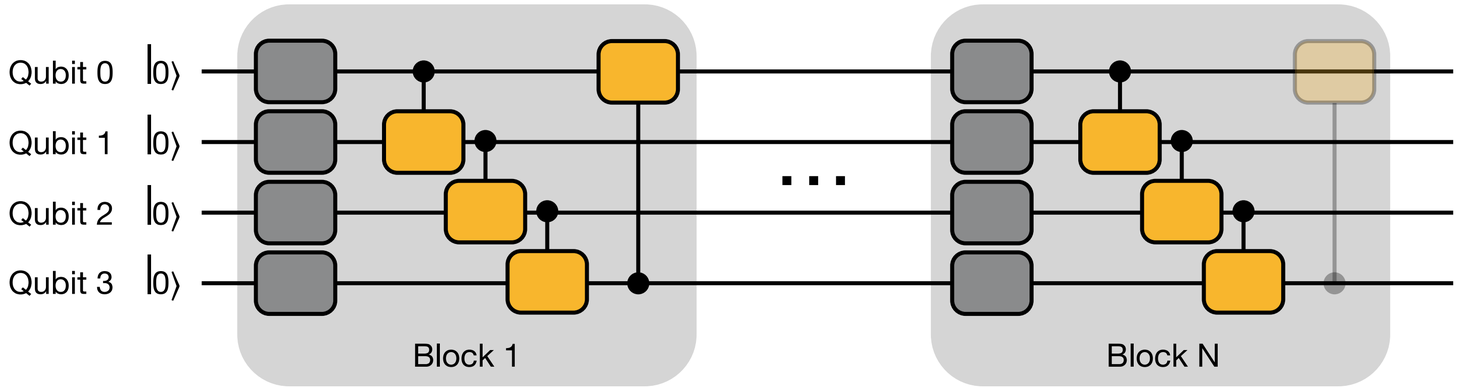
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



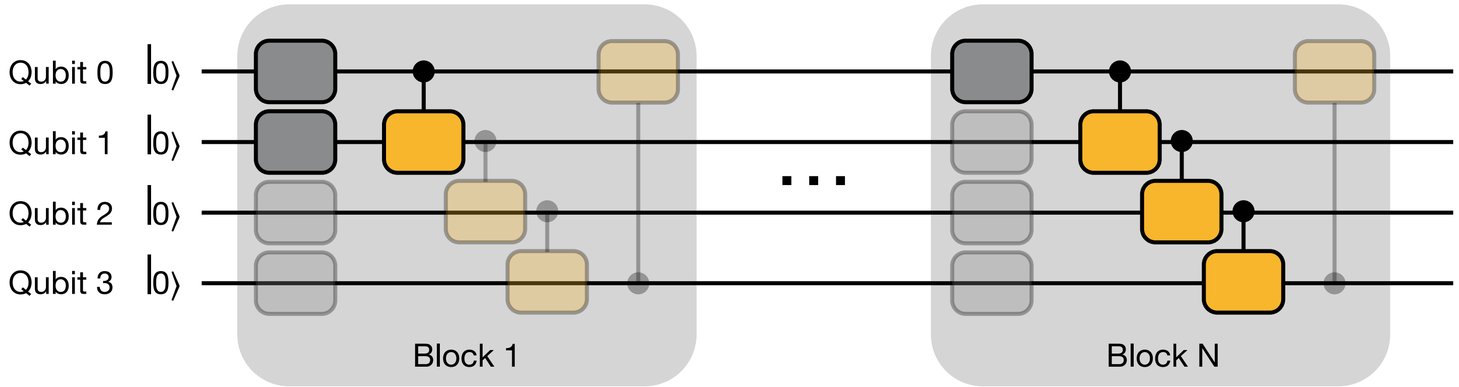
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



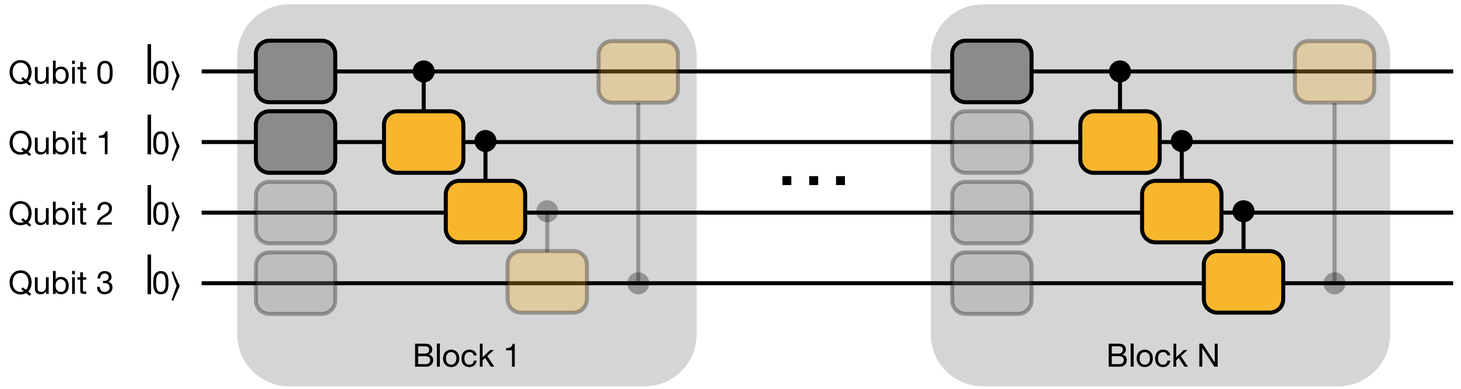
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



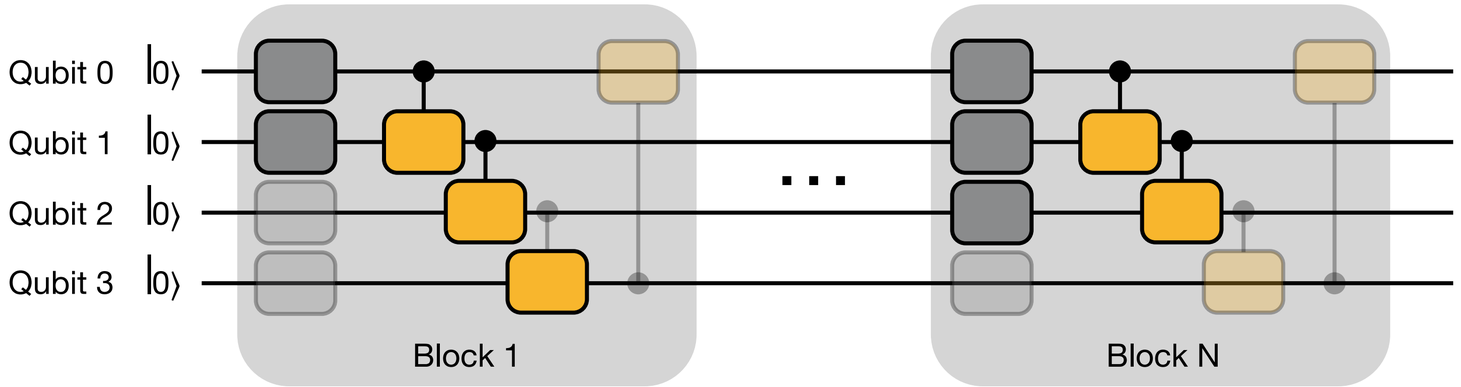
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



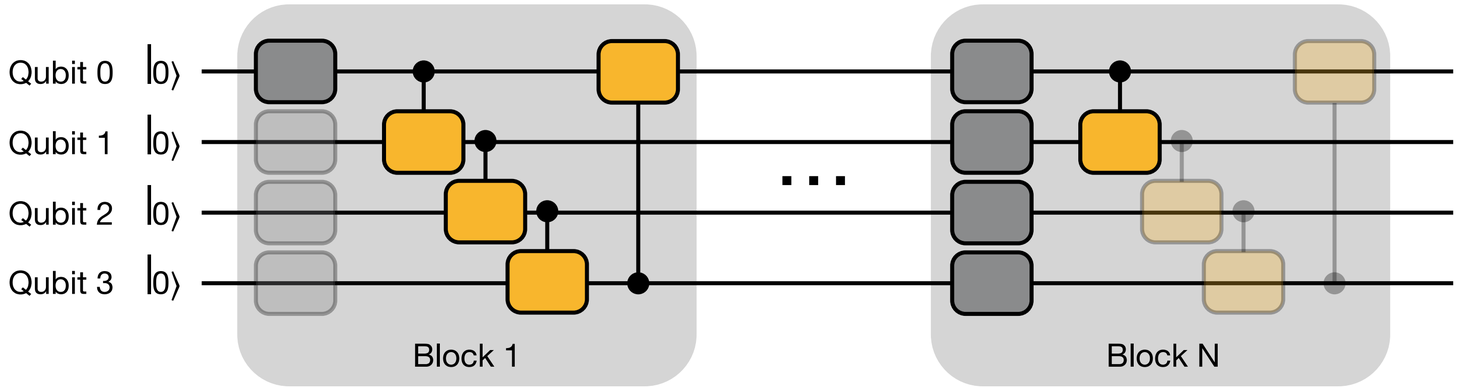
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



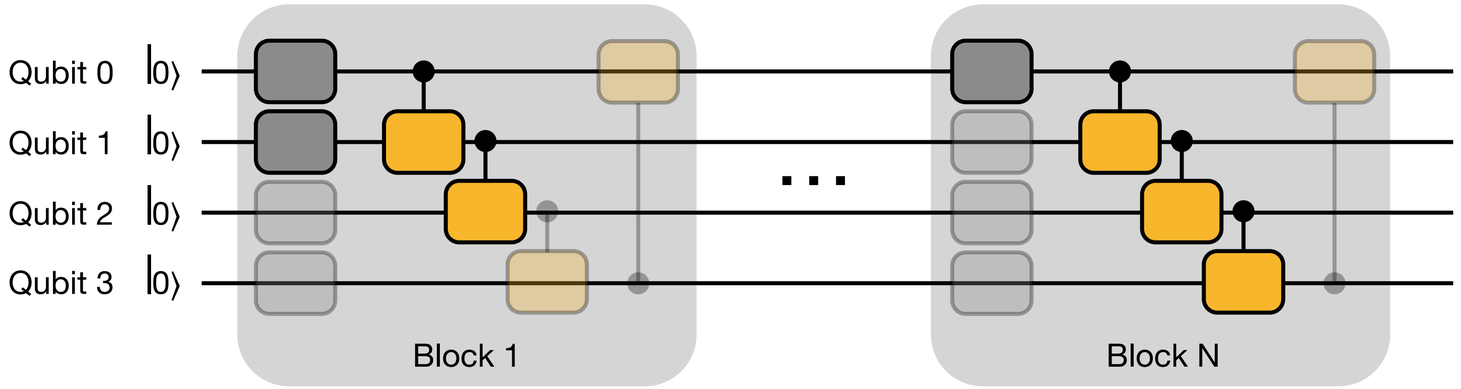
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



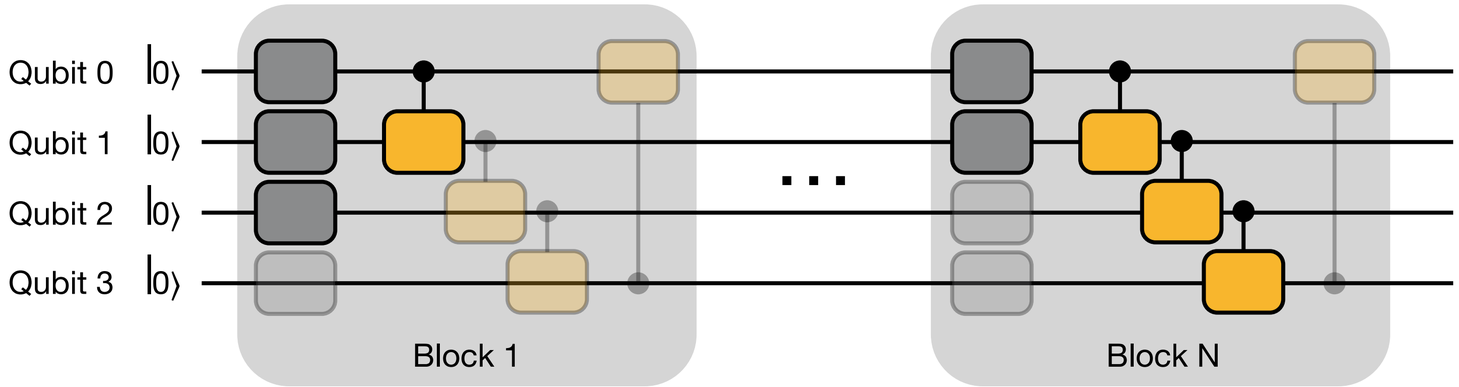
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



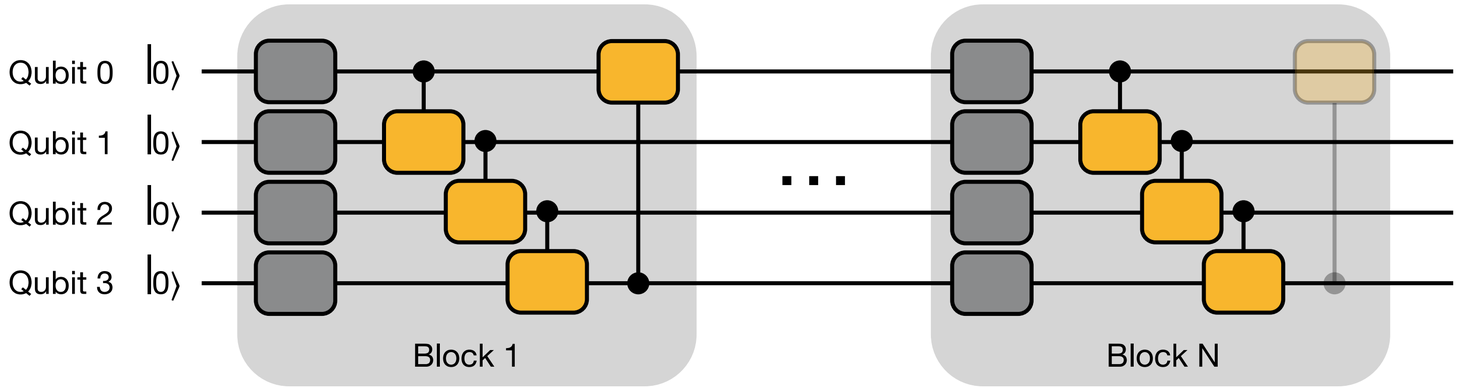
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



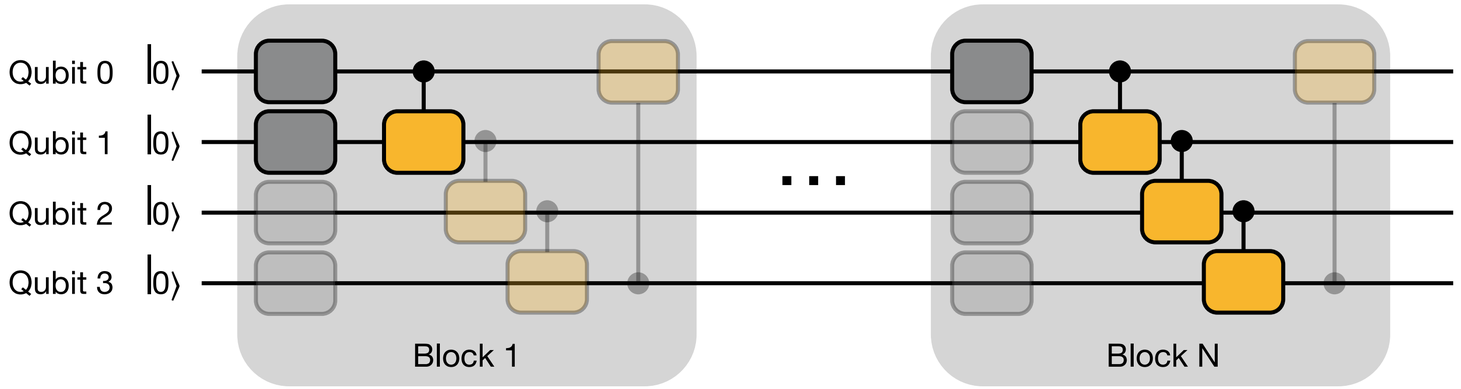
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



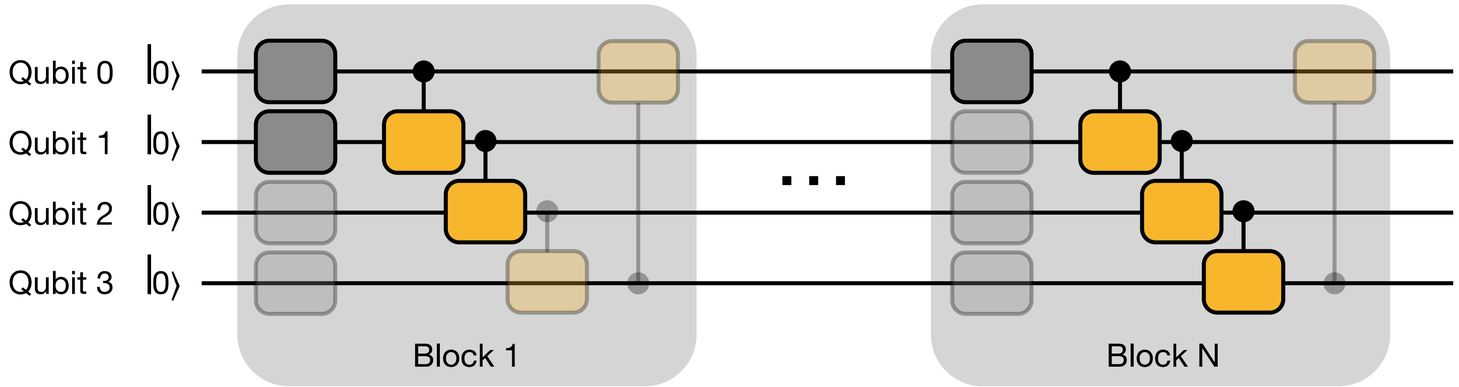
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train

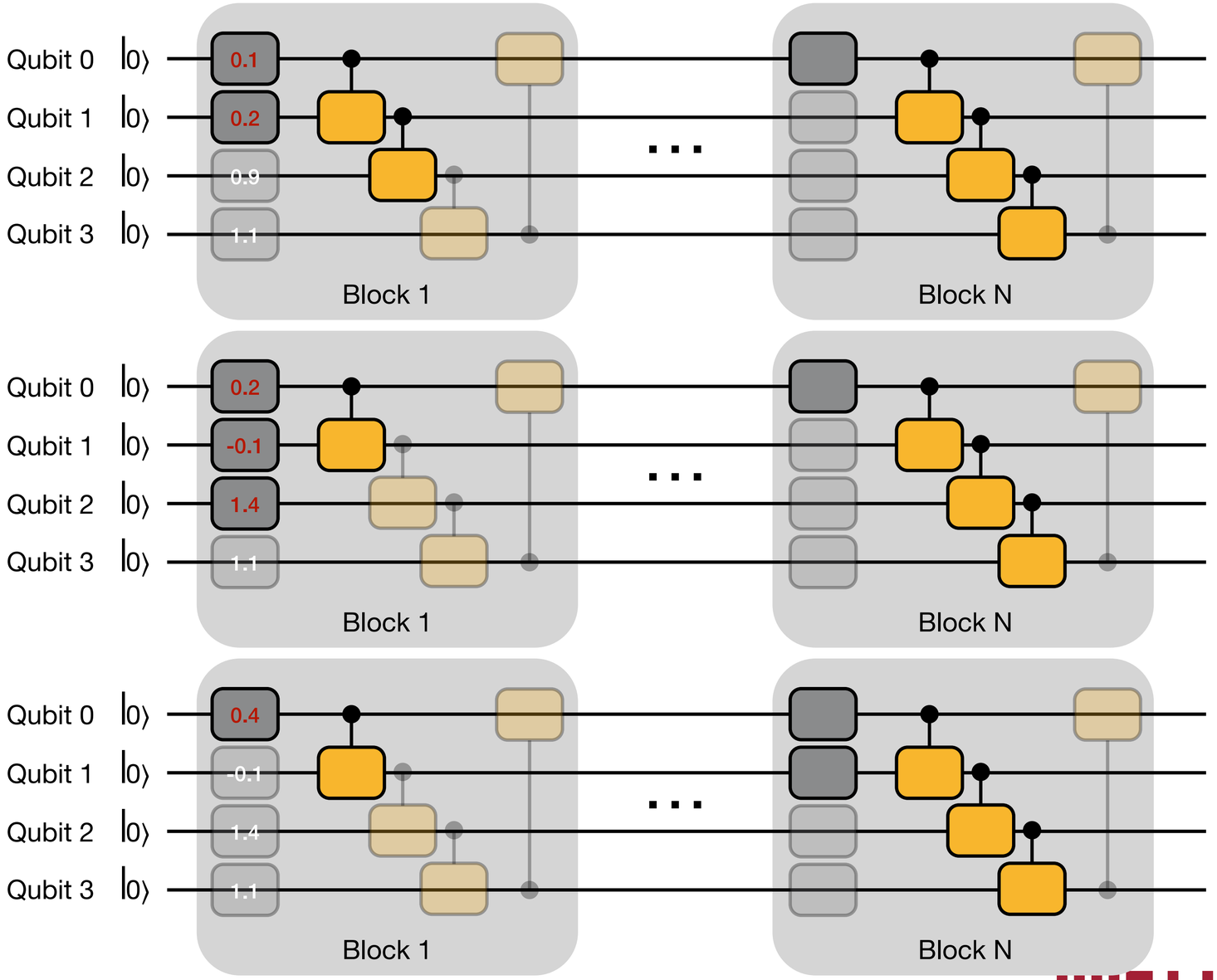


Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



Train SuperCircuit for Multiple Steps



Torch Quantum

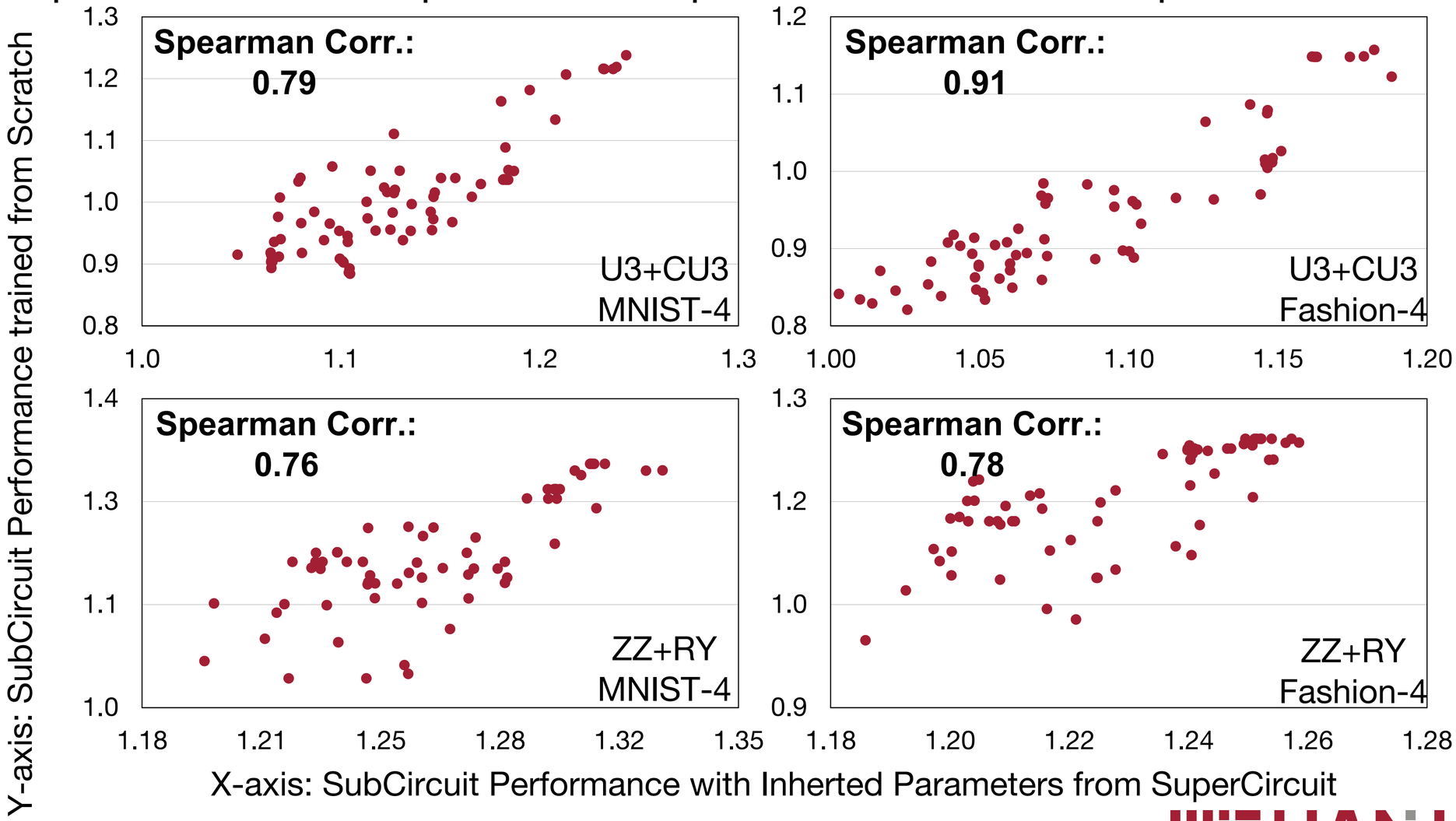
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

How Reliable is the SuperCircuit?

- Inherited parameters from SuperCircuit can provide accurate relative performance



Torch Quantum

Materials at: https://torchquantum.org

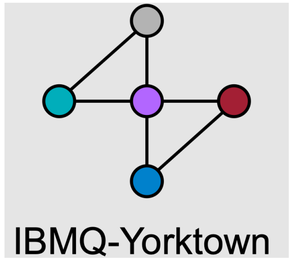
MIT HAN LAB

QuantumNAS

- SuperCircuit Construction and Training
- Noise-Adaptive Evolutionary Co-Search of SubCircuit and Qubit Mapping
- Train the Searched SubCircuit
- Iterative Quantum Gate Pruning

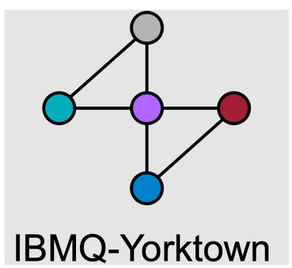
Noise-Adaptive Evolutionary Co-Search

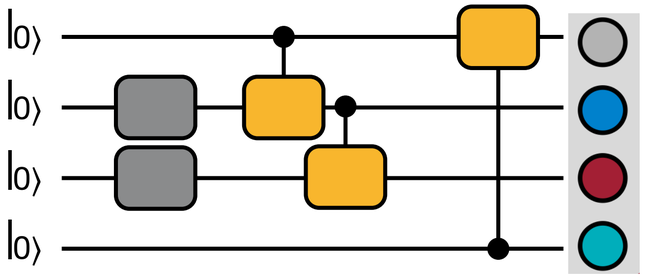
- Search the best SubCircuit and its qubit mapping on target device

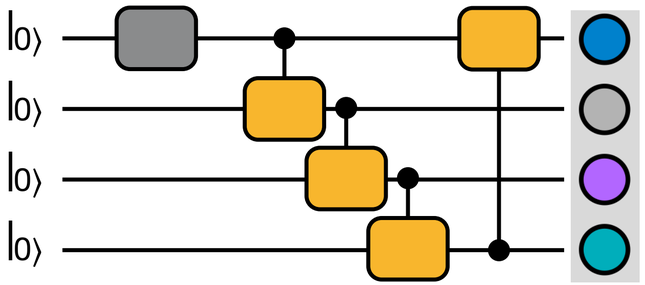


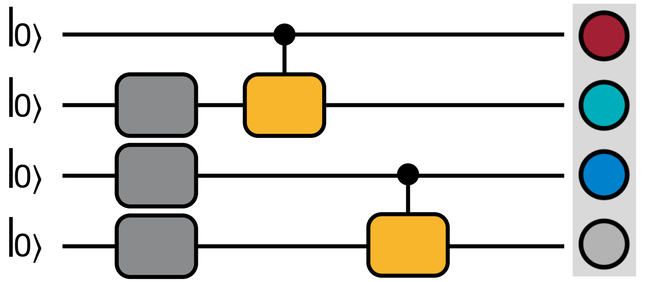
Noise-Adaptive Evolutionary Co-Search

- Search the best SubCircuit and its qubit mapping on target device









Evolutionary Search Engine
Mutation
Crossover

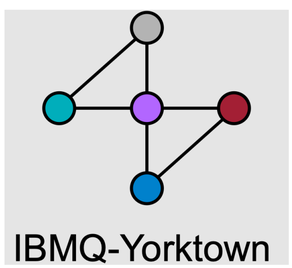
SubCircuit and Qubit Mapping pairs

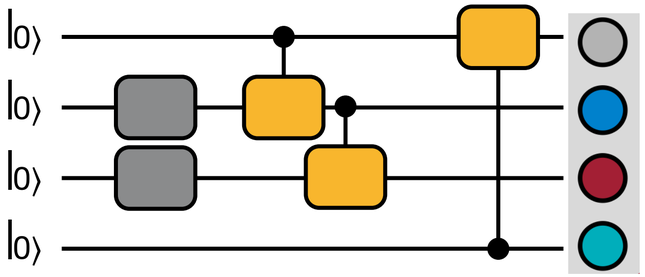
Parameters from SuperCircuit

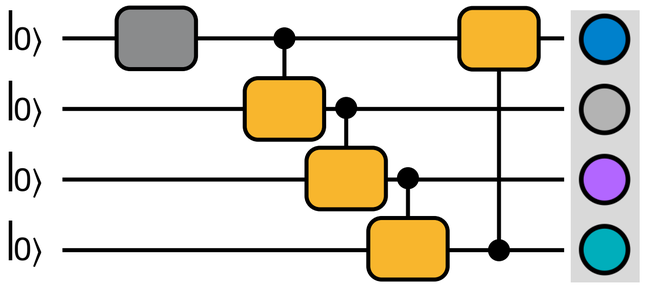
Simulate with noise model
Or
Directly run on real QC device

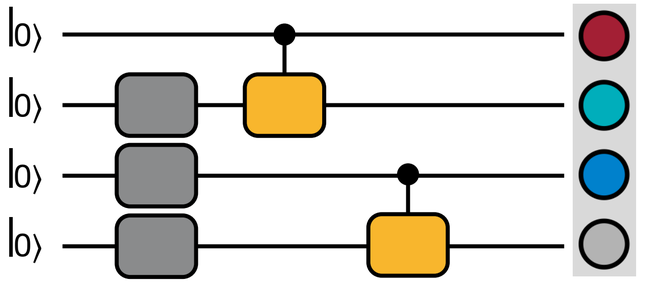
Noise-Adaptive Evolutionary Co-Search

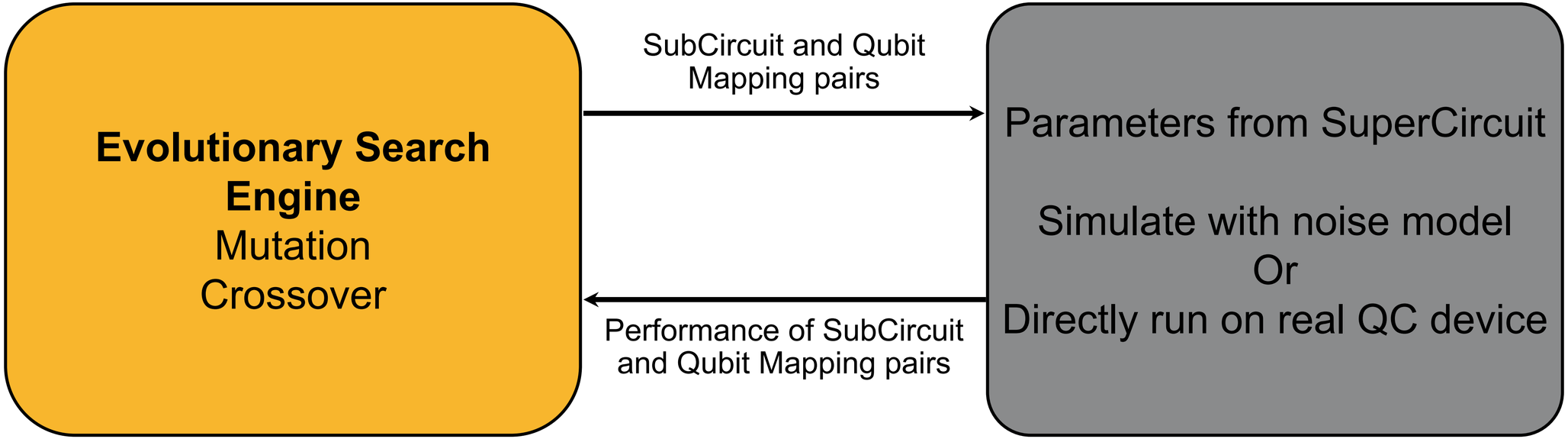
- Search the best SubCircuit and its qubit mapping on target device











QuantumNAS

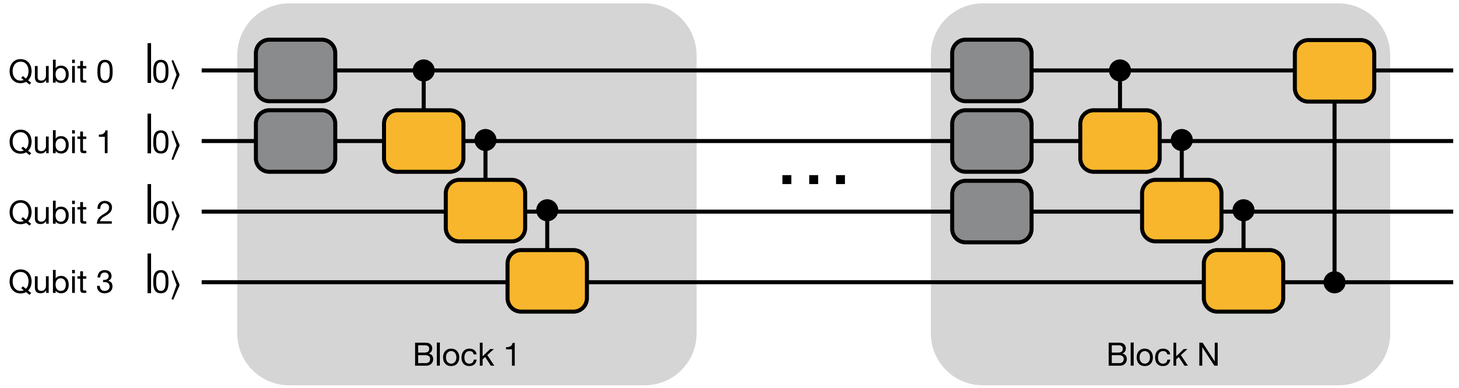
- SuperCircuit Construction and Training
- Noise-Adaptive Evolutionary Co-Search of SubCircuit and Qubit Mapping
- Train the Searched SubCircuit
- Iterative Quantum Gate Pruning

QuantumNAS

- SuperCircuit Construction and Training
- Noise-Adaptive Evolutionary Co-Search of SubCircuit and Qubit Mapping
- Train the Searched SubCircuit
- Iterative Quantum Gate Pruning

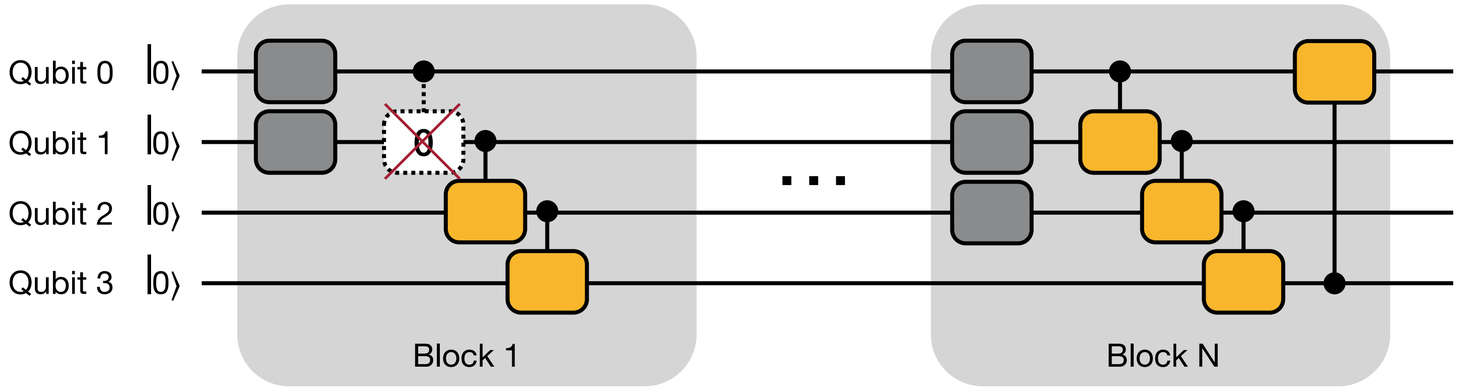
Iterative Pruning

- Some gates have parameters close to 0
  - Rotation gate with angle close to 0 has small impact on the results
- Iteratively prune small-magnitude gates and fine-tune the remaining parameters



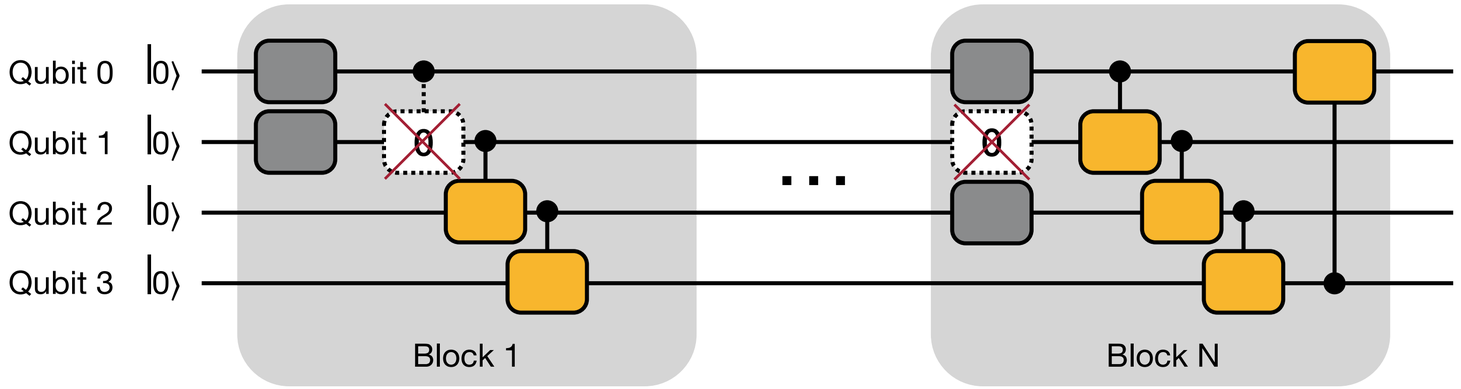
Iterative Pruning

- Some gates have parameters close to 0
  - Rotation gate with angle close to 0 has small impact on the results
- Iteratively prune small-magnitude gates and fine-tune the remaining parameters



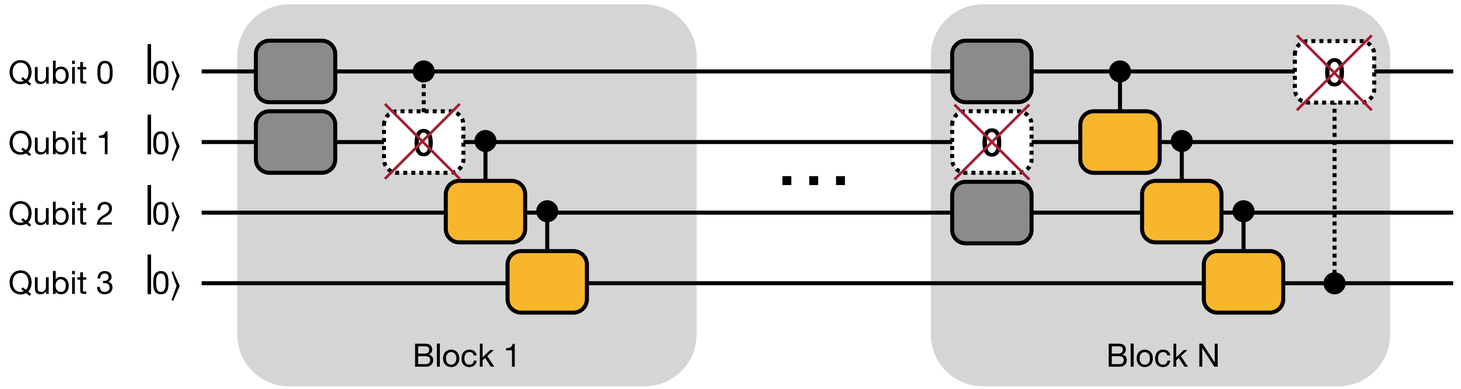
Iterative Pruning

- Some gates have parameters close to 0
  - Rotation gate with angle close to 0 has small impact on the results
- Iteratively prune small-magnitude gates and fine-tune the remaining parameters



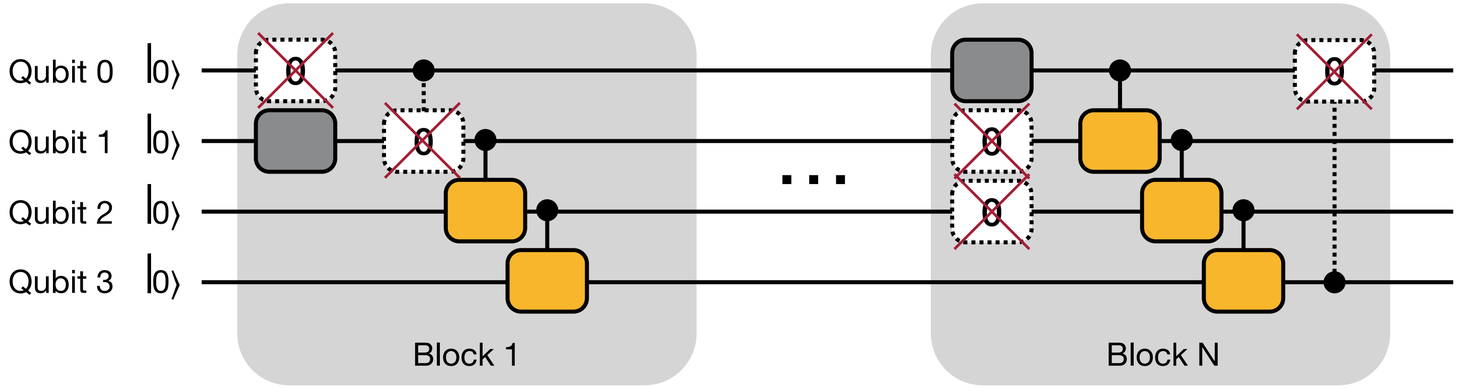
Iterative Pruning

- Some gates have parameters close to 0
  - Rotation gate with angle close to 0 has small impact on the results
- Iteratively prune small-magnitude gates and fine-tune the remaining parameters



Iterative Pruning

- Some gates have parameters close to 0
  - Rotation gate with angle close to 0 has small impact on the results
- Iteratively prune small-magnitude gates and fine-tune the remaining parameters

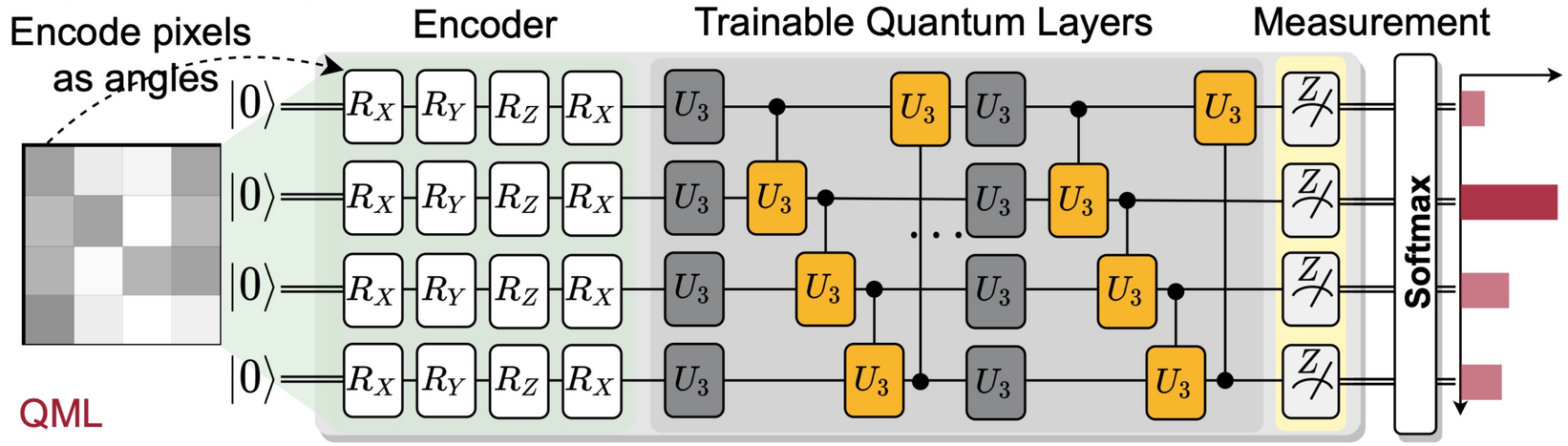


Evaluation Setups: Benchmarks and Devices

- Benchmarks
  - QML classification tasks: MNIST 10-class, 4-class, 2-class, Fashion 4-class, 2-class, Vowel 4-class
  - VQE task molecules: H2, H2O, LiH, CH4, BeH2
- Quantum Devices
  - IBMQ
  - #Qubits: 5 to 65
  - Quantum Volume: 8 to 128

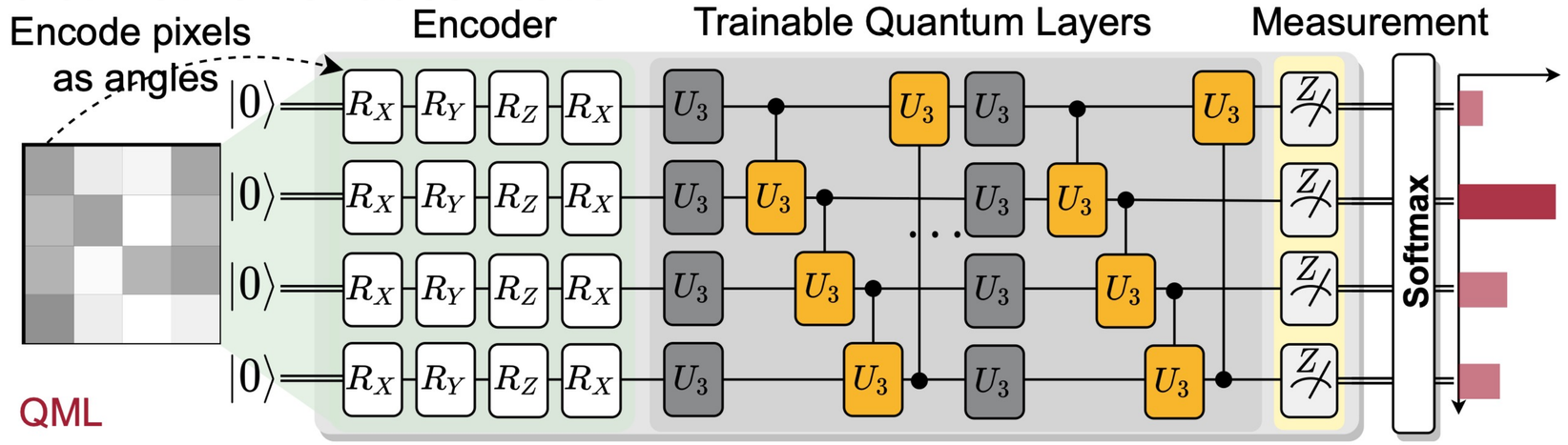
Benchmarks: QNN and VQE

- Quantum Neural Networks: classification

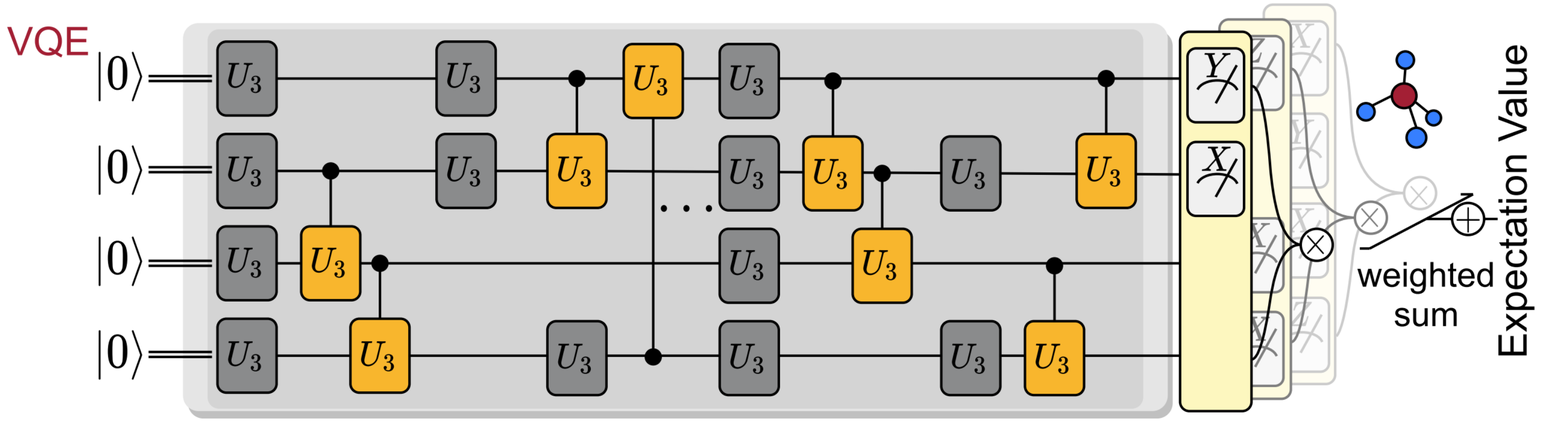


Benchmarks: QNN and VQE

- Quantum Neural Networks: classification



- Variational Quantum Eigensolver: finds the ground state energy of molecule Hamiltonian



Torch Quantum

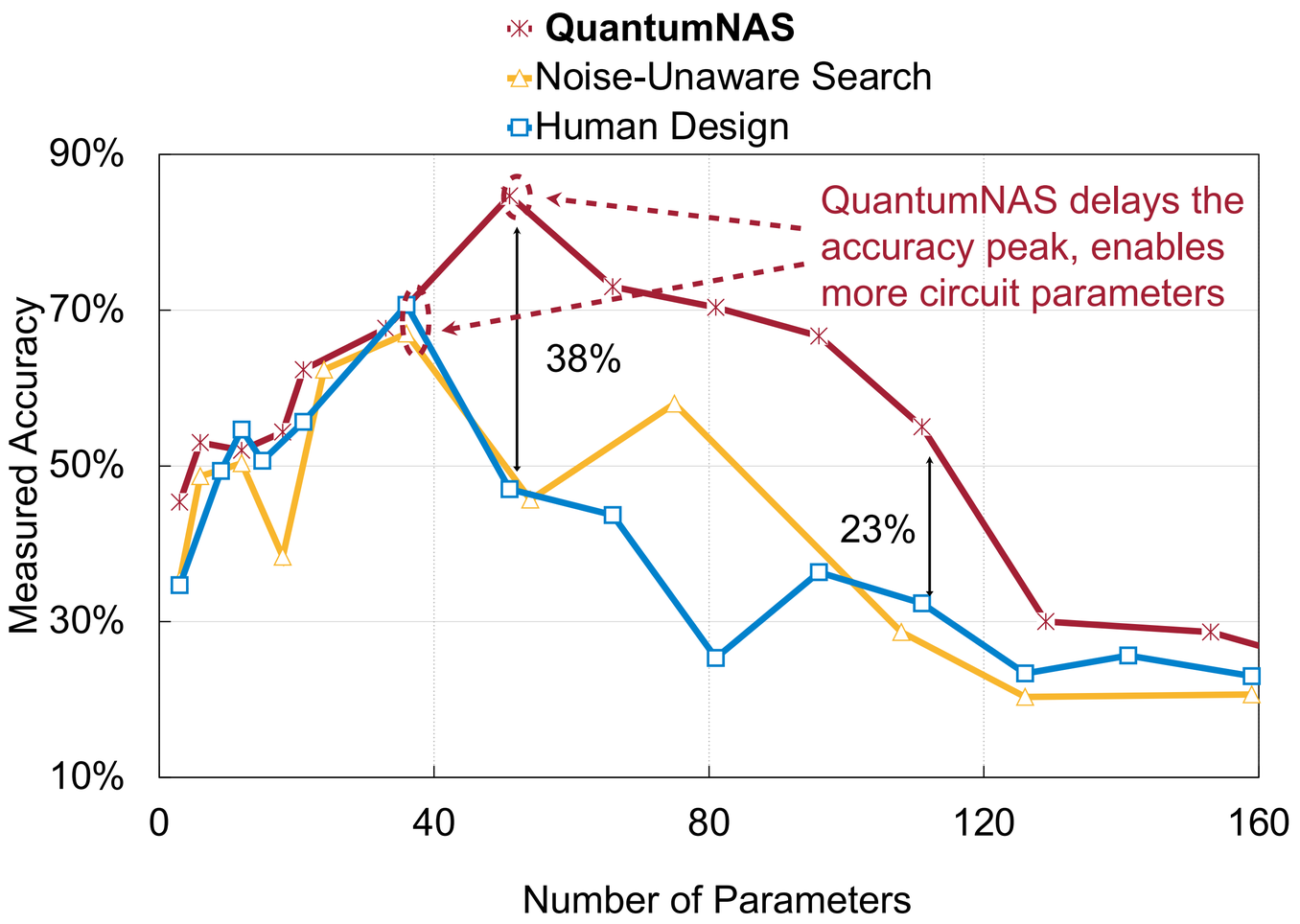
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

QML Results

- 4-classification: MNIST-4 U3+CU3 on IBMQ-Yorktown



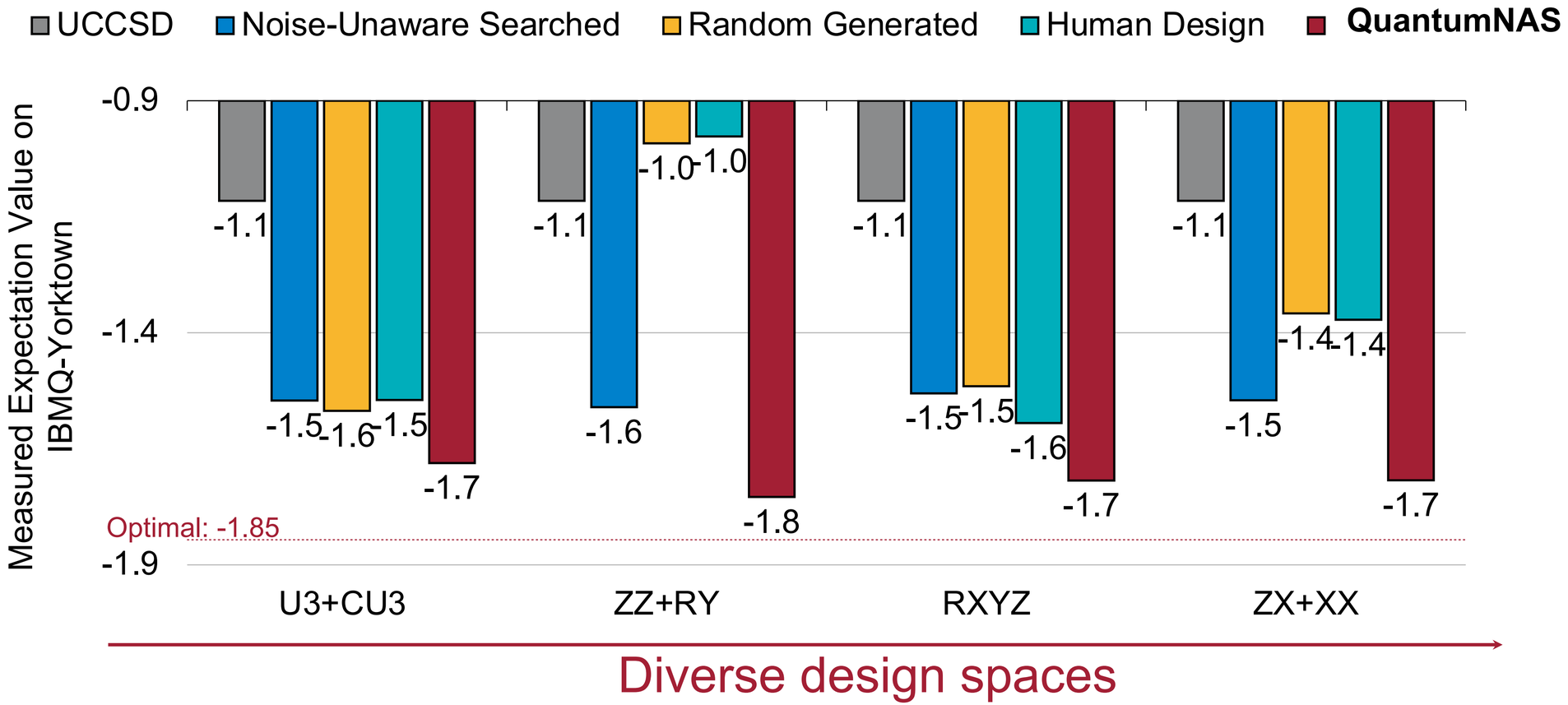
Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

Consistent Improvements on Diverse Design Spaces

- H2 in different design spaces on IBMQ-Yorktown



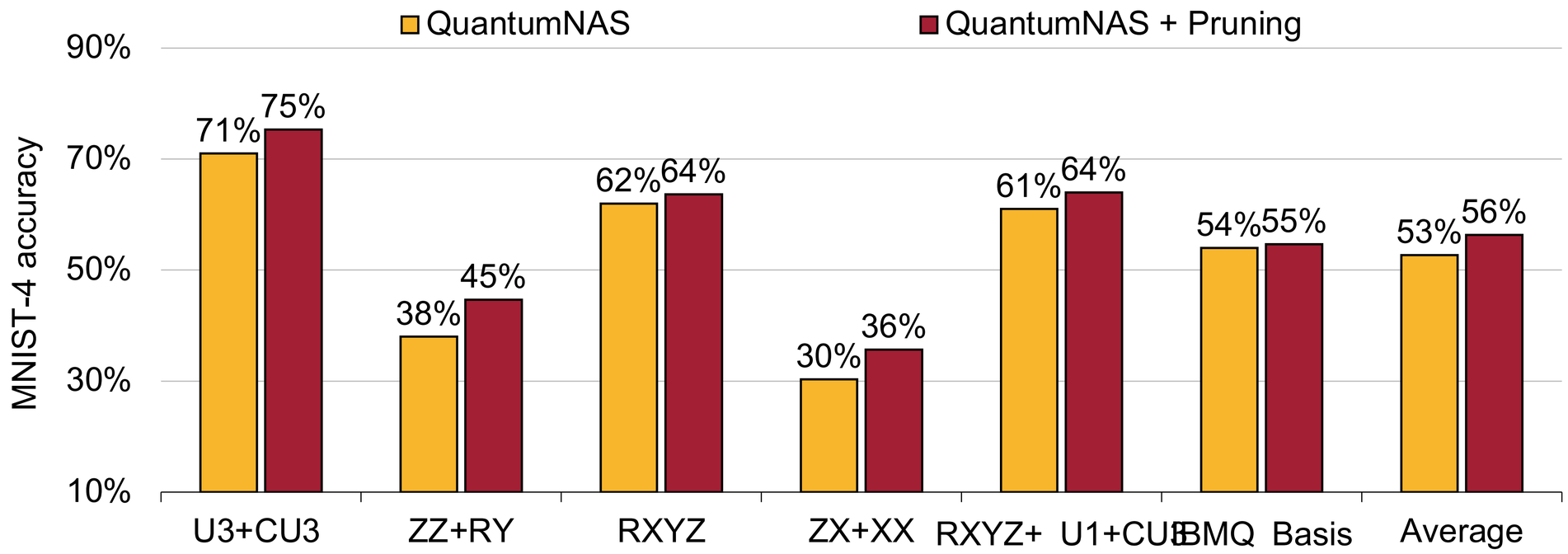
Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

Effective of Quantum Gate Pruning

- For MNIST-4, Quantum gate pruning improves accuracy by 3% on average



Torch Quantum

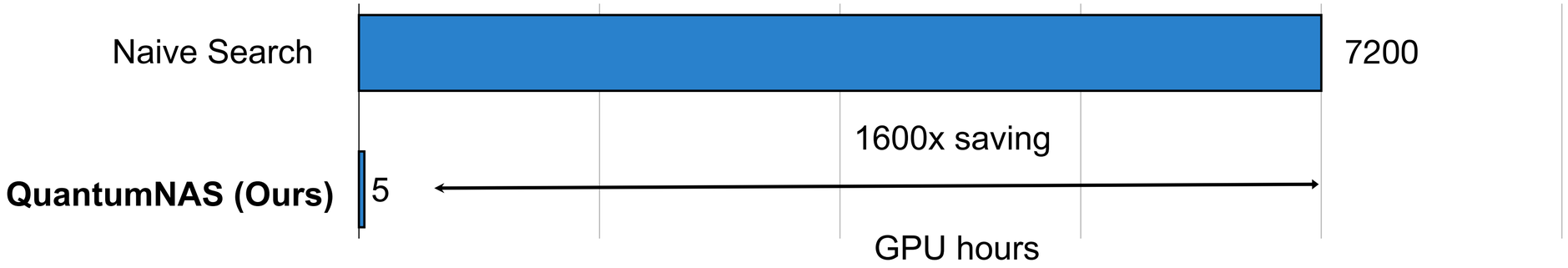
Materials at: https://torchquantum.org

MIT HAN LAB

Time Cost with TorchQuantum

- On 1 Nvidia Titan RTX 2080 ti GPU

#qubits \ Step	SuperCircuit Training	Noise-Adaptive Co-search	SubCircuit Training	Deployment on Real QC
4 Qubits	0.5h	3h	0.5h	0.5h
15 Qubits	5h	5h	5h	1h
21 Qubits	20h	10h	15h	1h

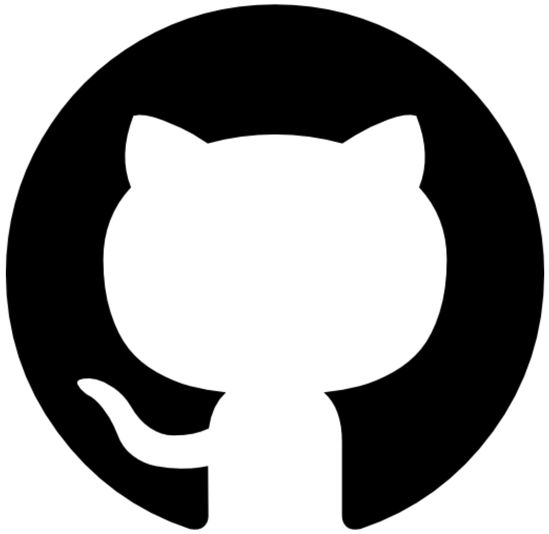


Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

Hands-On Section
3.1 QuantumNAS



Torch Quantum

Materials at: https

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations
- 1.3 TQ Use Examples
- 1.4 QNN Compression

Section 2

Use TorchQuantum on Pulse Level Optimization

- 3.1 Quantum Optimal Control
- 3.2 Variational Pulse Learning

Section 3

Use TorchQuantum on Gate Level Optimization

- 3.1 QuantumNAS: Ansatz Search and Gate Pruning
- 3.2 QuantumNAT: Noise Injection and Quantization
- 3.3 QOC: On-Chip Training
- 3.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

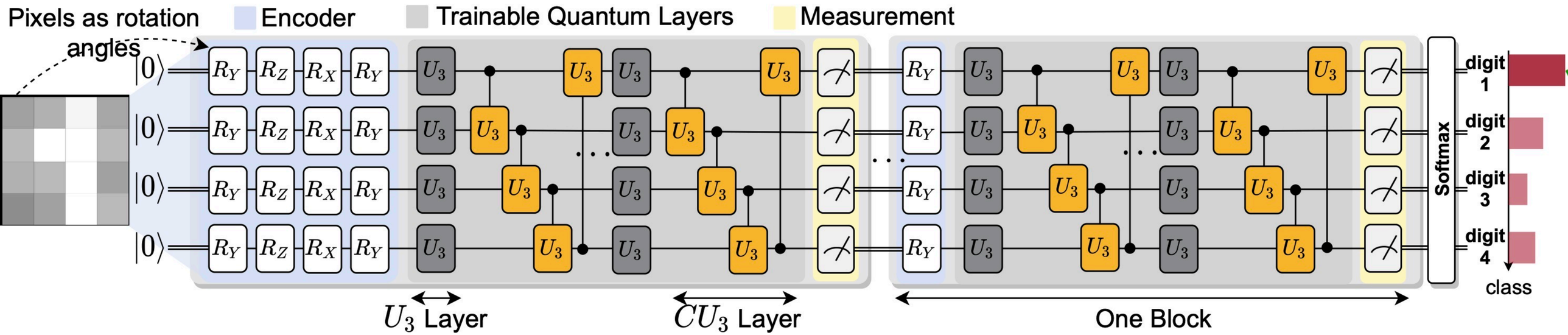
2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

QuantumNAS vs. QuantumNAT

- **QuantumNAS** finds noise robust circuit **architecture**
- **QuantumNAT** finds noise robust circuit **parameters**

PQC Circuit in QuantumNAT

- QNN with multiple nodes
  - Encoder
  - Trainable Quantum Layers
  - Measurements



Torch Quantum

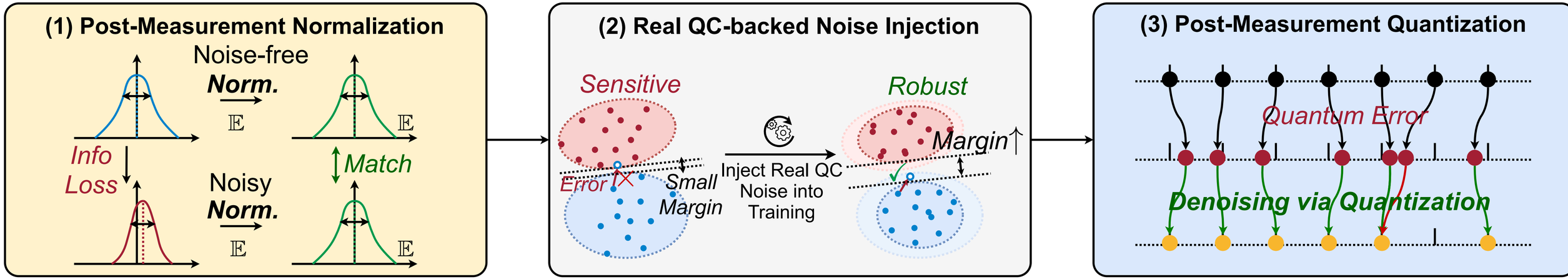
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Three Techniques in QuantumNAT

- QuantumNAT:
  - Normalization: mitigate noise impact
  - Noise injection: make the parameters aware of noise
  - Quantization: mitigate noise impact



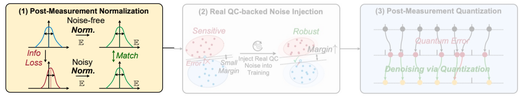
Torch Quantum

Materials at: https://torchquantum.org

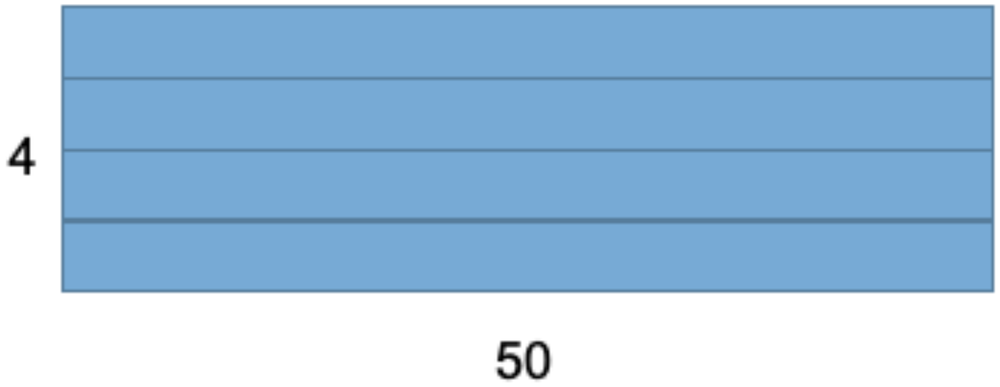
MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

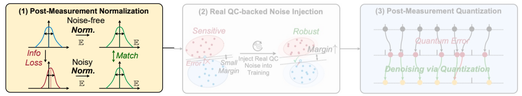
Post-Measurement Normalization



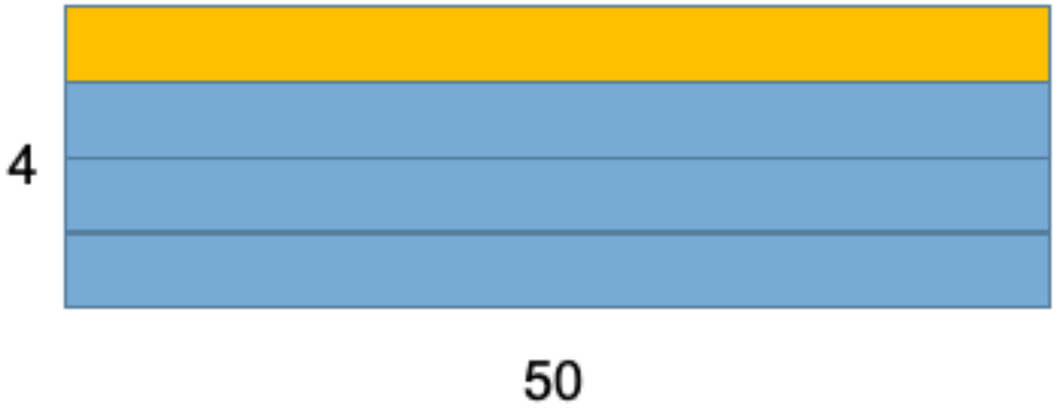
- Normalize the measurement outcome
  - Along the **batch** dimension
  - For example, we train the 4-qubit PQC with batch=50 then we have results as 50 * 4 values



Post-Measurement Normalization



- Normalize the measurement outcome
  - Along the **batch** dimension
  - For example, we train the 4-qubit PQC with batch=50 then we have results as 50 * 4 values
    - Compute the mean and std on of the measurement outcome on each qubit across batch dim

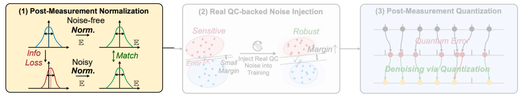


Torch Quantum

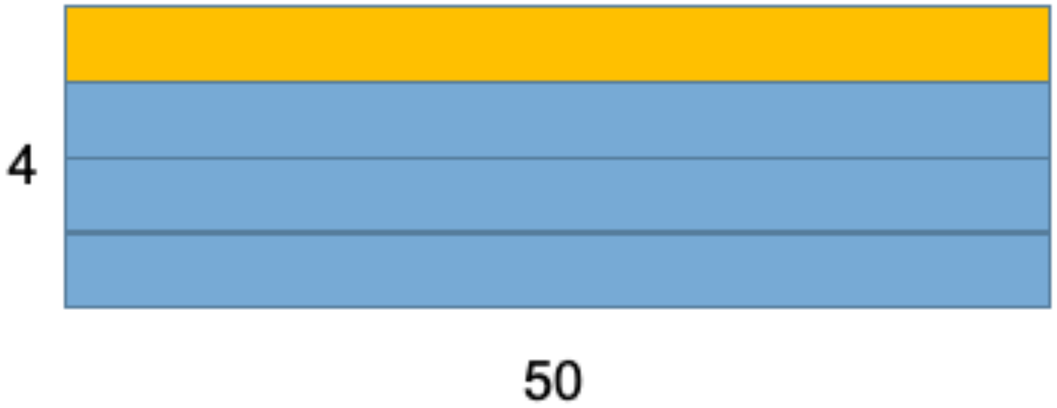
THAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Post-Measurement Normalization



- Normalize the measurement outcome
  - Along the **batch** dimension
  - For example, we train the 4-qubit PQC with batch=50 then we have results as 50 * 4 values
    - Compute the mean and std on of the measurement outcome on each qubit across batch dim
    - Normalize the measurement outcome with the computed mean and std



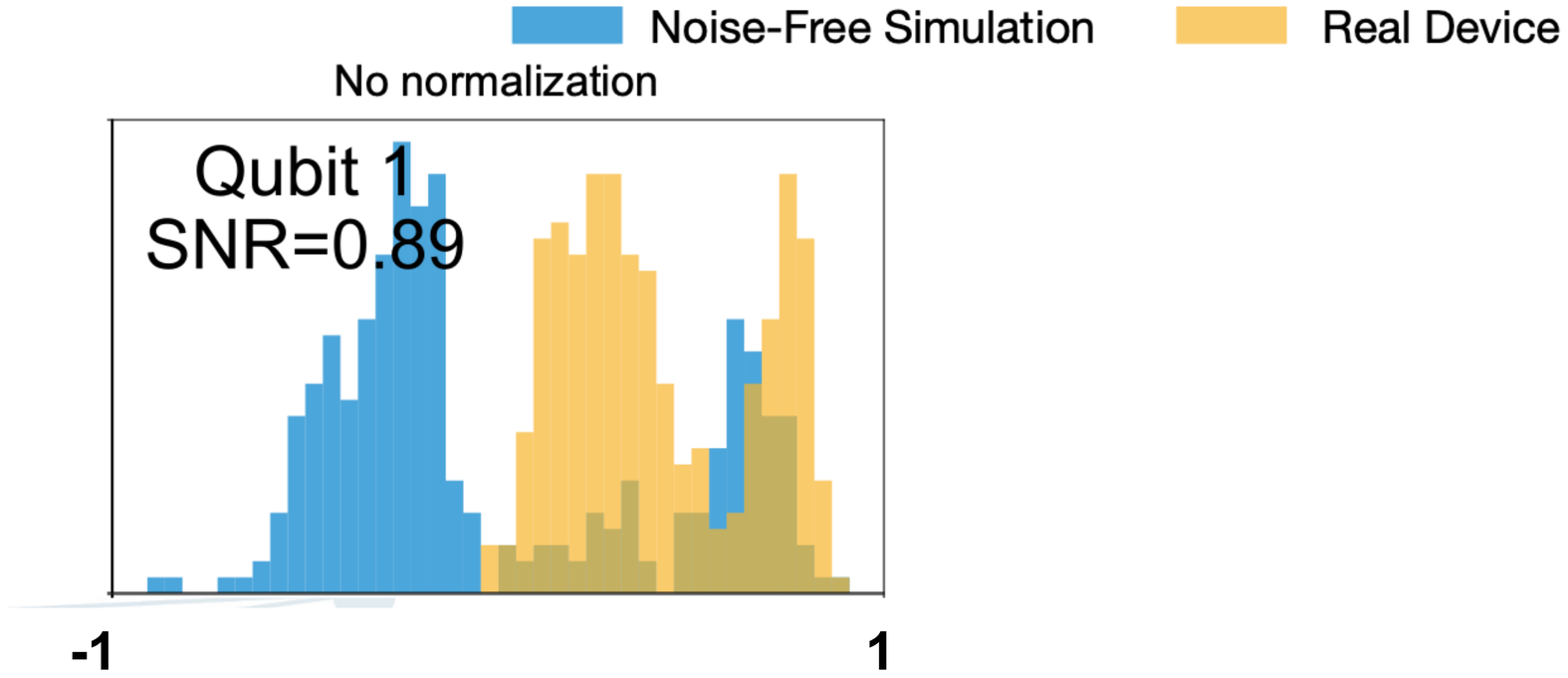
Torch Quantum

THAN LAB

IEEE/ACM ICCAD 2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Post-Measurement Normalization

- Normalize the measurement outcome
  - Along the **batch** dimension
- Measurement outcome distribution of 50 quantum circuits:



Torch Quantum

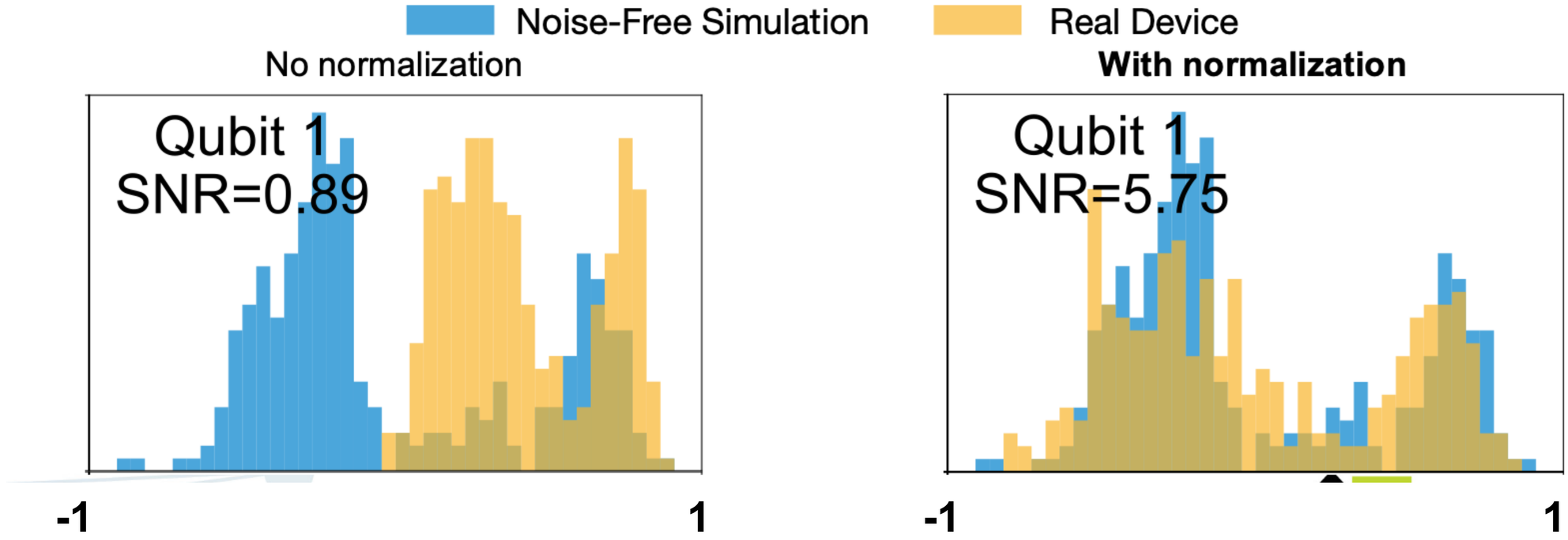
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Post-Measurement Normalization

- Normalize the measurement outcome
  - Along the **batch** dimension
- Measurement outcome distribution of 50 quantum circuits:



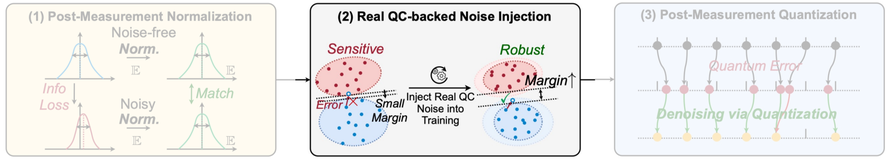
Torch Quantum

Materials at: https://torchquantum.org

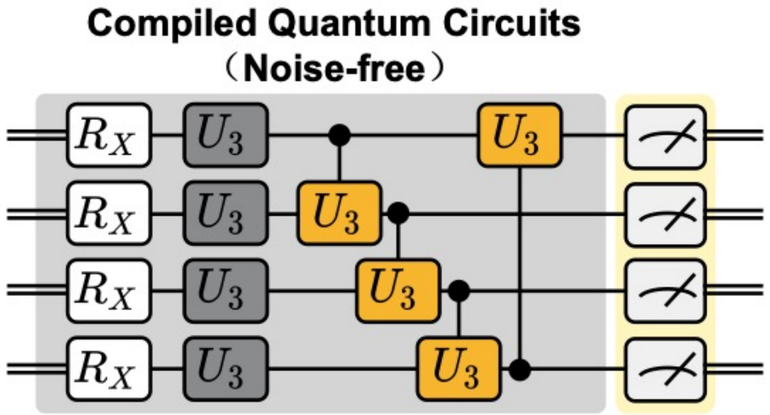
MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Noise Injection



- Inject noise during training on classical simulator
  - Pauli error
  - Readout error



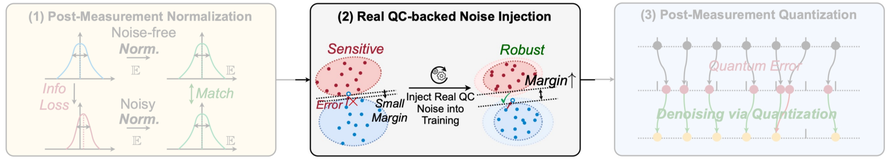
Torch Quantum

Materials at: https://torchquantum.org

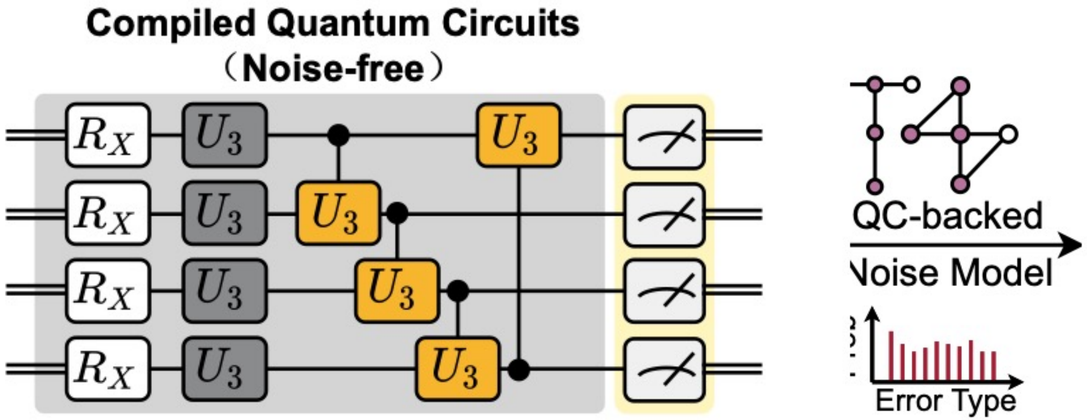
MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Noise Injection



- Inject noise during training on classical simulator
  - Pauli error
  - Readout error



Pauli error: SX gate: {X: 0.00096, Y: 0.00096, Z: 0.00096, None: 0.99712}

Readout Error Matrix:

0.984, 0.016

0.022, 0.978

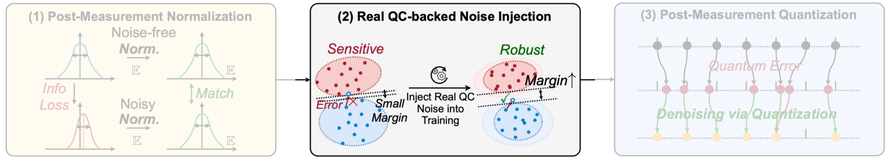
Materials at: https://torchquantum.org

Torch Quantum

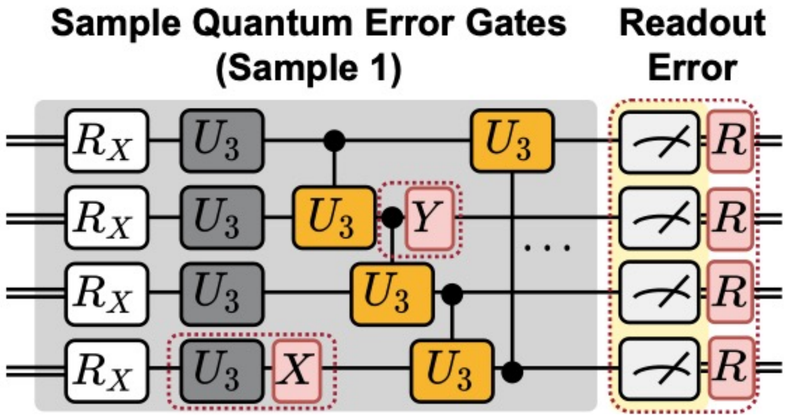
MIT HAN LAB

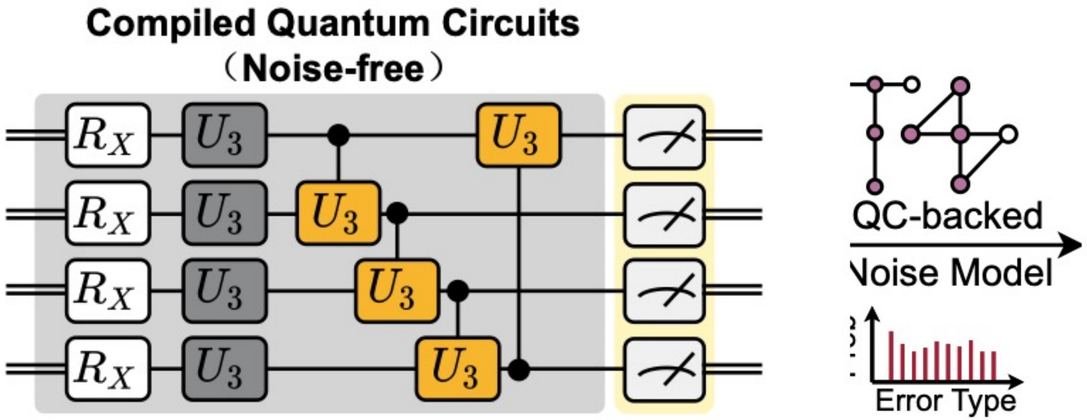
2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Noise Injection



- Inject noise during training on classical simulator
  - Pauli error
  - Readout error





Pauli error: SX gate: {X: 0.00096, Y: 0.00096, Z: 0.00096, None: 0.99712}

Readout Error Matrix: **0.984, 0.016**
0.022, 0.978

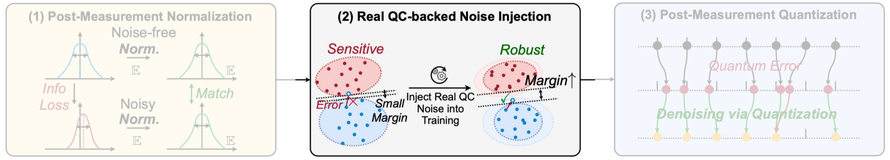
Materials at: https://torchquantum.org

Torch Quantum

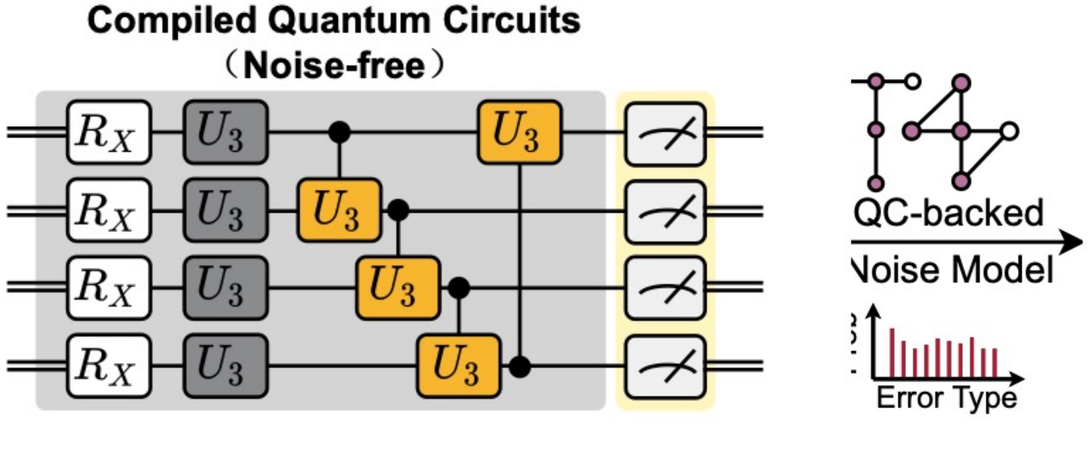
MIT HAN LAB

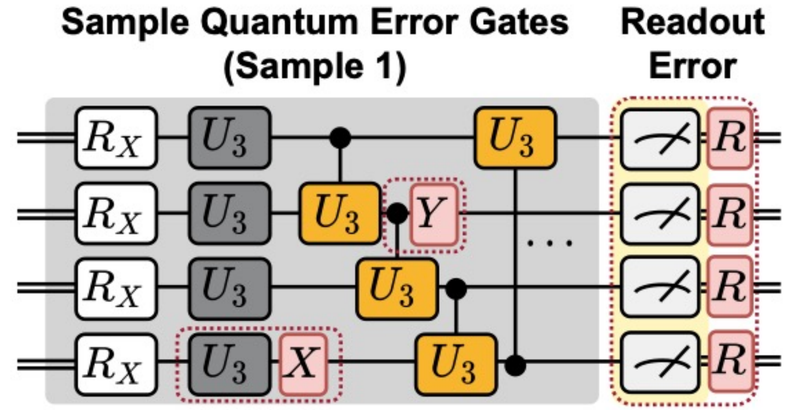
2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

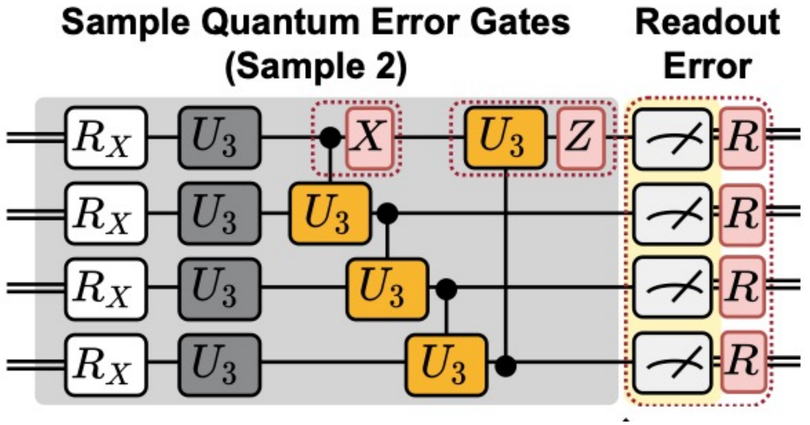
Noise Injection



- Inject noise during training on classical simulator
  - Pauli error
  - Readout error







Pauli error: SX gate: {X: 0.00096, Y: 0.00096, Z: 0.00096, None: 0.99712}

Readout Error Matrix: 0.984, 0.016
0.022, 0.978

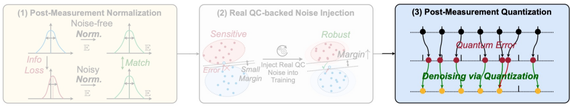
Materials at: https://torchquantum.org

Torch Quantum

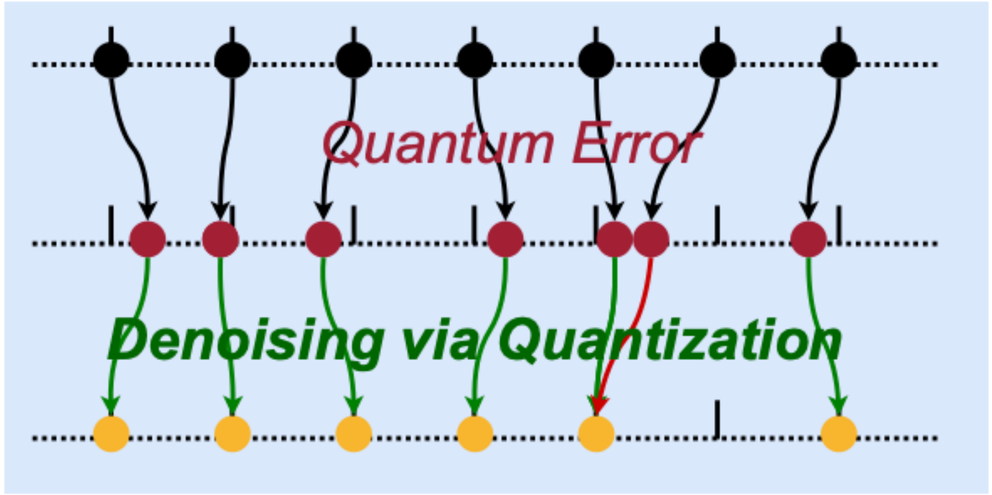
MIT HAN LAB

ICCAD 2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Post-Measurement Quantization



- Quantize measurement outcomes
  - Denoising effect
  - Small errors will be mitigated



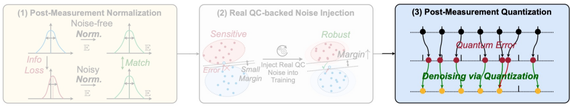
Torch Quantum

Materials at: https://torchquantum.org

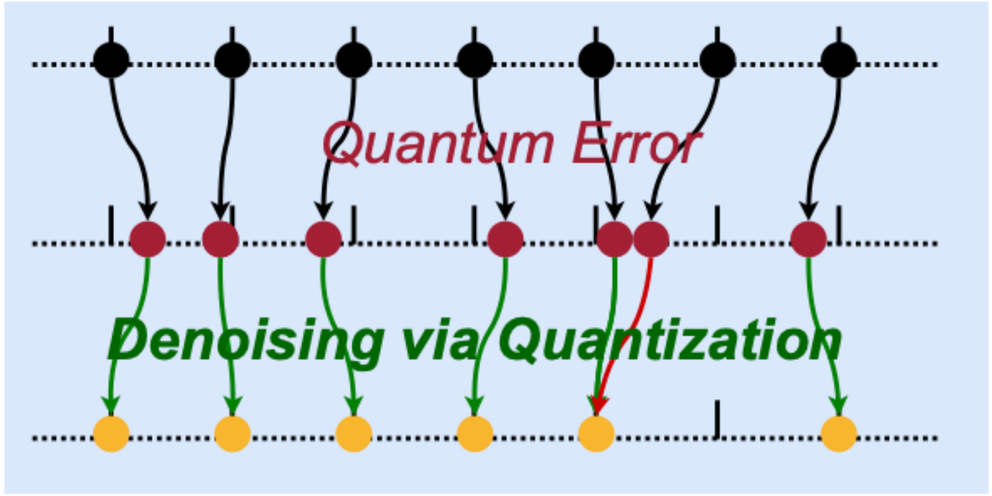
MIT HAN LAB

IEEE/ACM ICCAD 2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

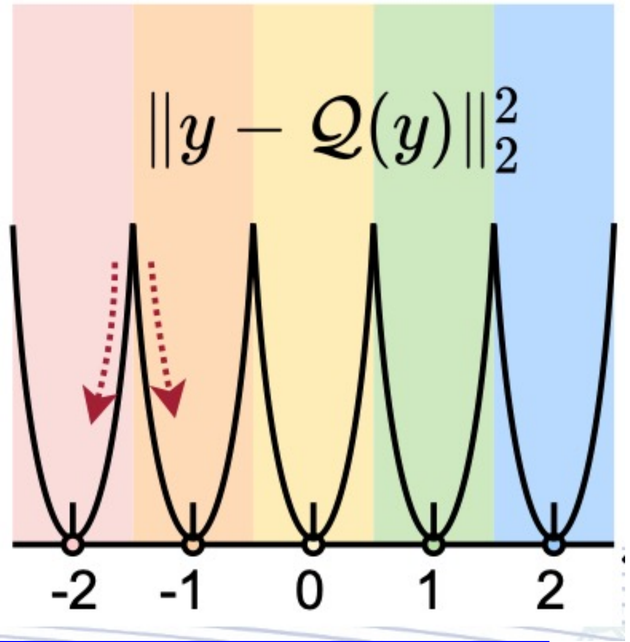
Post-Measurement Quantization



- Quantize measurement outcomes
  - Denoising effect
  - Small errors will be mitigated



- **Loss** term to encourage measurement outcomes to be close to **centroids**



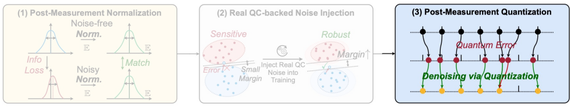
Torch Quantum

Materials at:

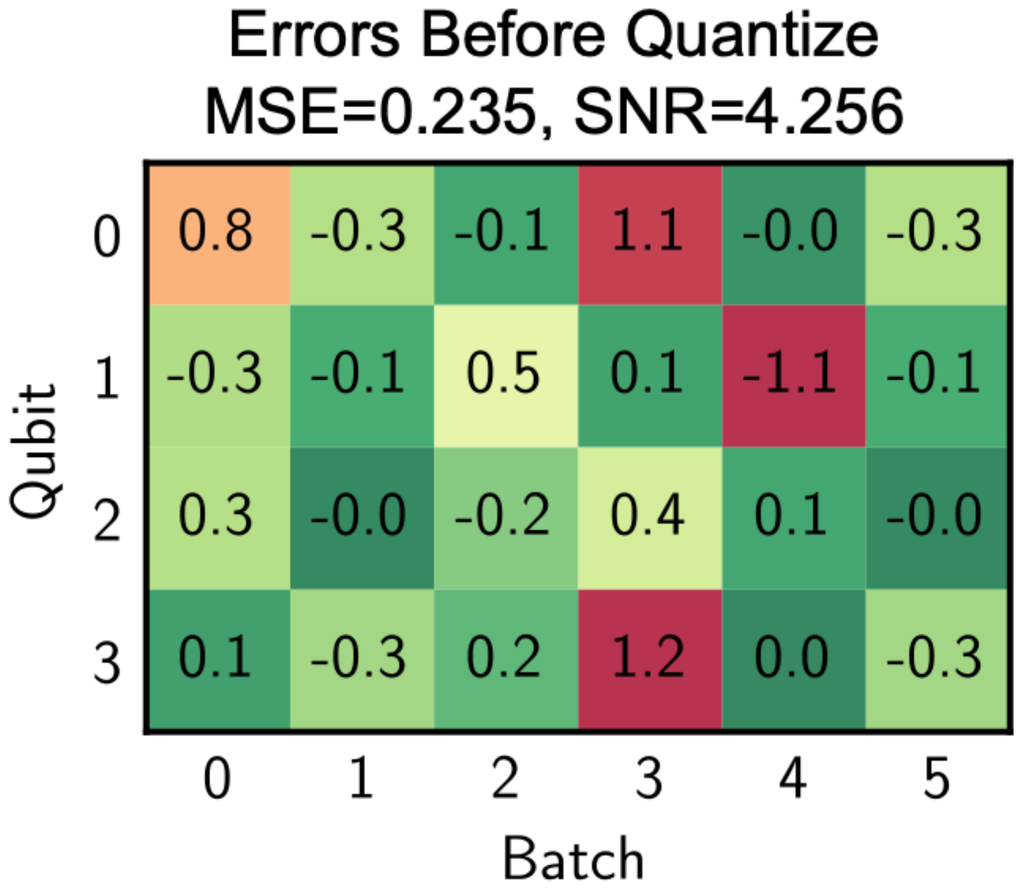
MIT HAN LAB

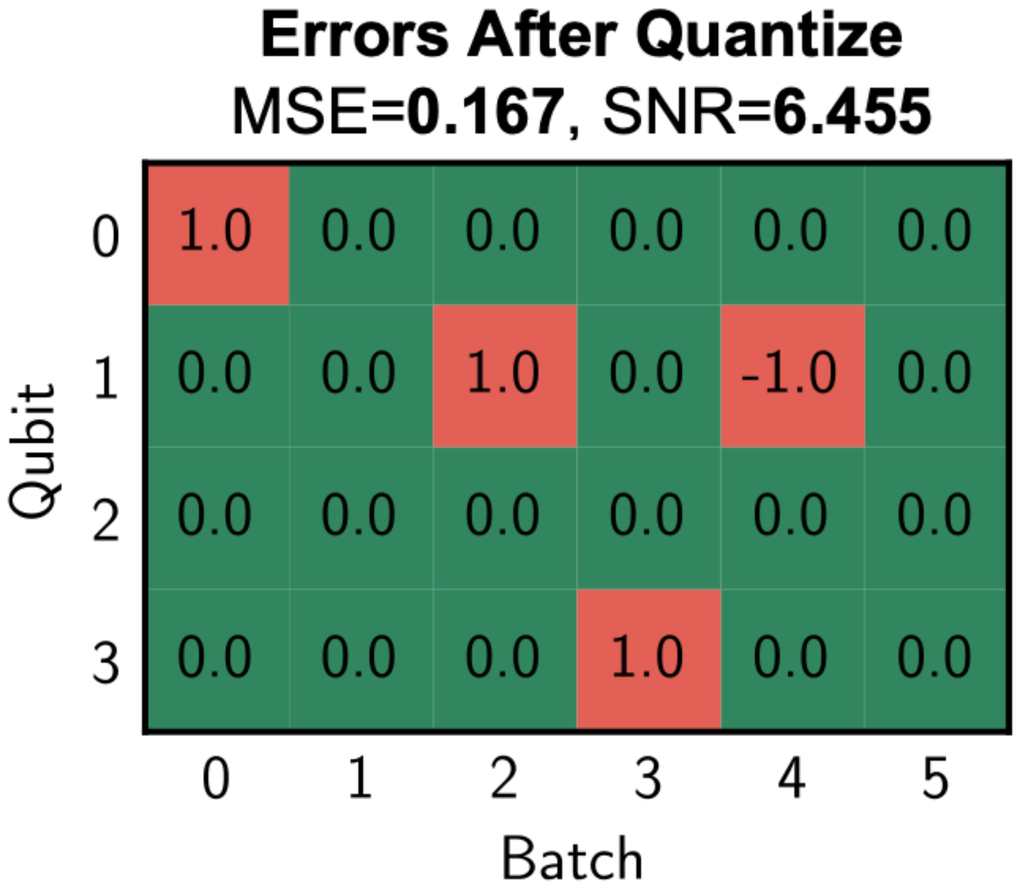
2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Post-Measurement Quantization



- Quantization reduces errors and improves SNR





Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

IEEE/ACM ICCAD 2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Evaluation

- Benchmarks
  - Quantum Machine Learning task:
    - MNIST 10-class, 4-class, 2-class
    - Fashion MNIST 10-class, 4-class, 2-class
    - Vowel 4-class
    - Cifar-2 class
- Quantum Devices
  - IBMQ
  - #Qubits: 5 to 15
  - Quantum Volume: 8 to 32

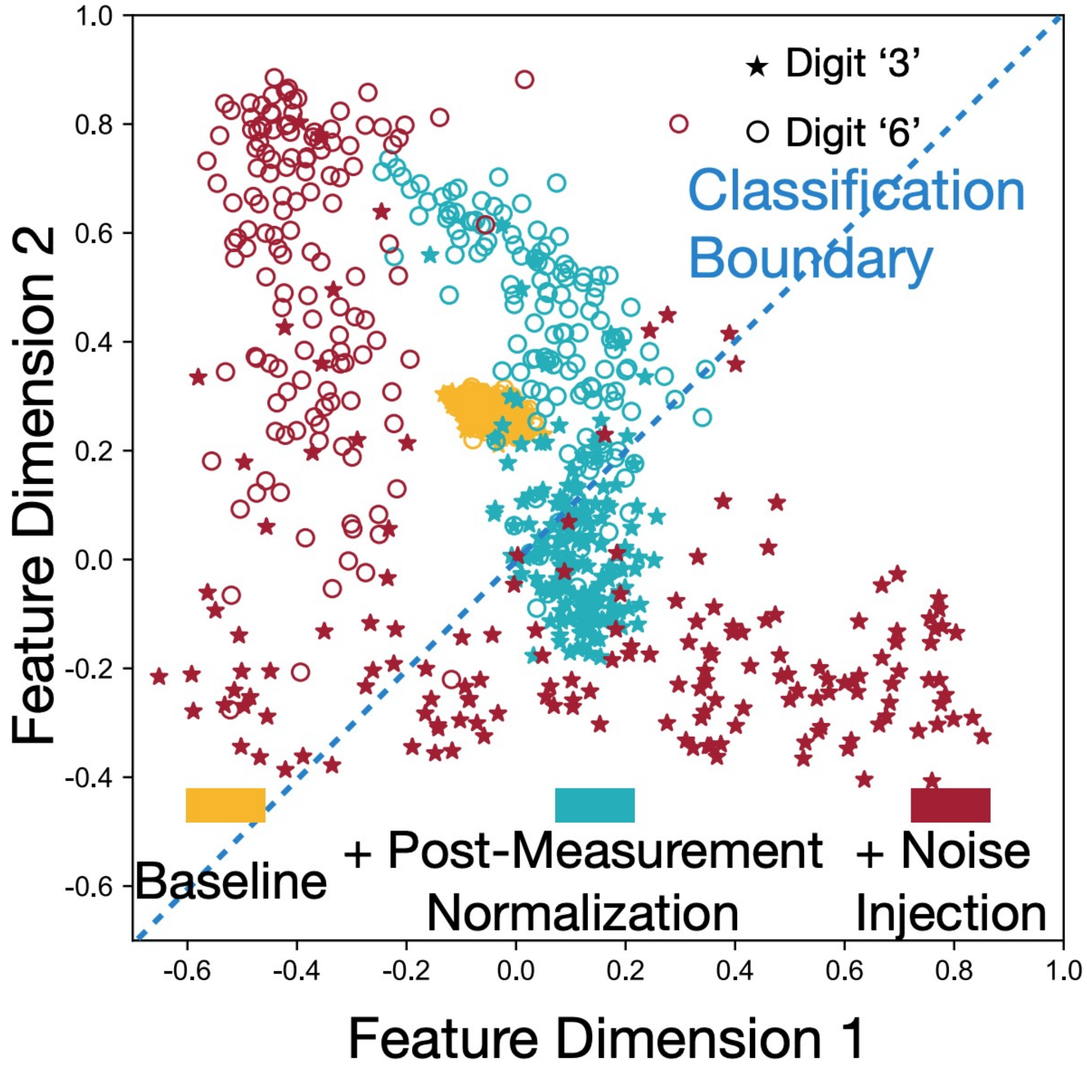
Consistent Improvements on Various Benchmarks

- On IBMQ santiago

Method	MNIST-4	FMNIST-4	Vowel-4	MNIST-2	FMNIST-2	Cifar-2
Baseline	0.30	0.32	0.28	0.84	0.78	0.51
+ Normalization	0.41	0.61	0.29	0.87	0.68	0.56
+Noise Injection	0.61	0.70	0.44	0.93	0.86	0.57
+ Quantization	**0.68**	**0.75**	**0.48**	**0.94**	**0.88**	**0.59**

Visualization

- QuantumNAT stretches the distribution of features
  - MNIST-2 classification task



Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Noise Model in TQ

```
noise_model_tq = tq.NoiseModelTQ(
        noise_model_name='ibmq_quito’,
        factor=10,
        add_thermal=True
    )
```

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations
- 1.3 TQ Use Examples
- 1.4 QNN Compression

Section 2

Use TorchQuantum on Pulse Level Optimization

- 3.1 Quantum Optimal Control
- 3.2 Variational Pulse Learning

Section 3

Use TorchQuantum on Gate Level Optimization

- 3.1 QuantumNAS: Ansatz Search and Gate Pruning
- 3.2 QuantumNAT: Noise Injection and Quantization
- 3.3 QOC: On-Chip Training
- 3.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

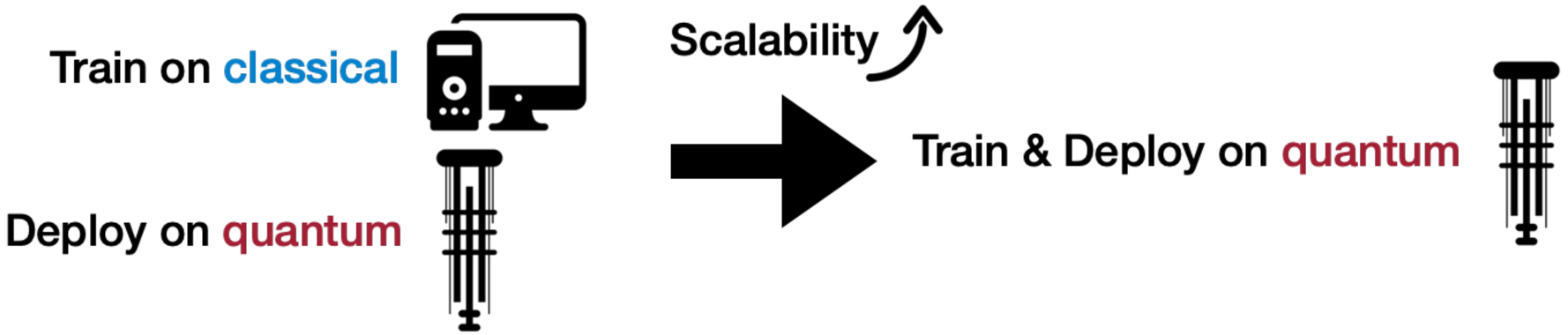
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

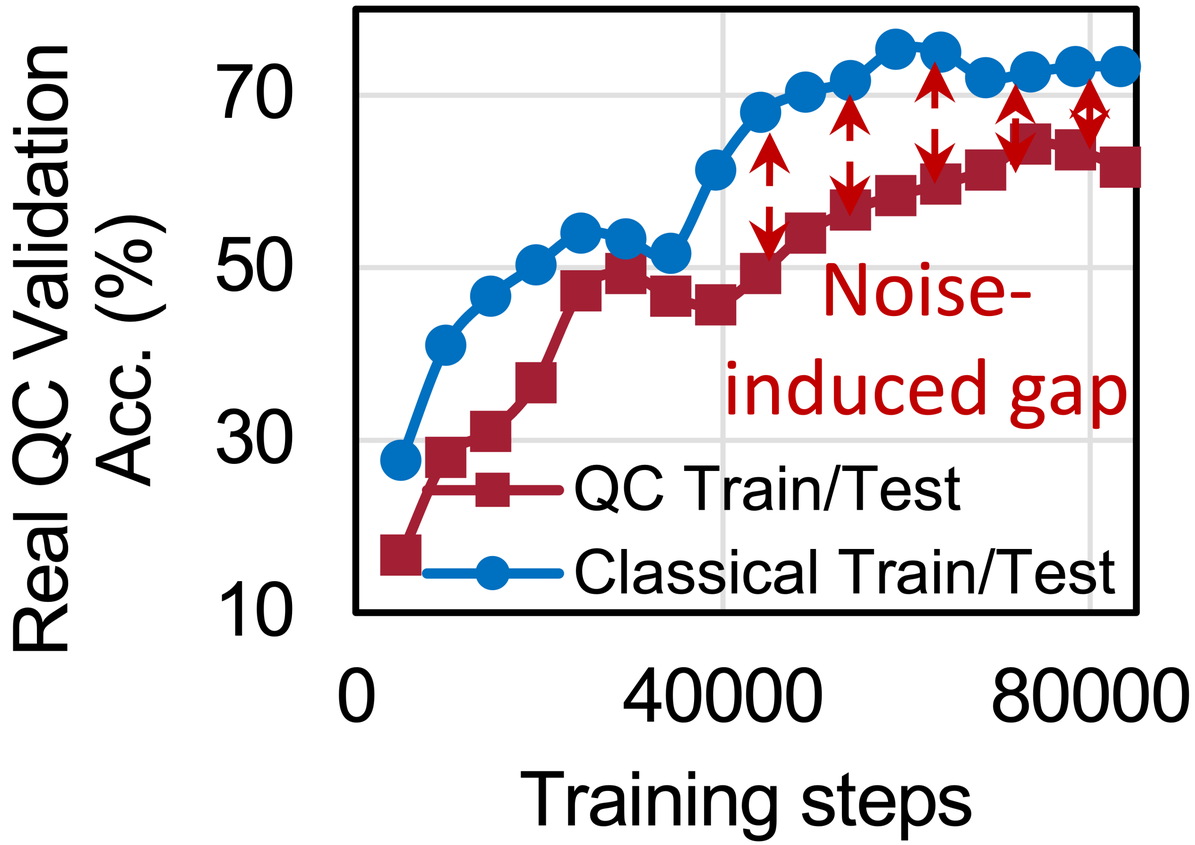
QOC for High Scalability

- How to further improve the scalability of PQC training?
- Train the parameters directly on real quantum machine



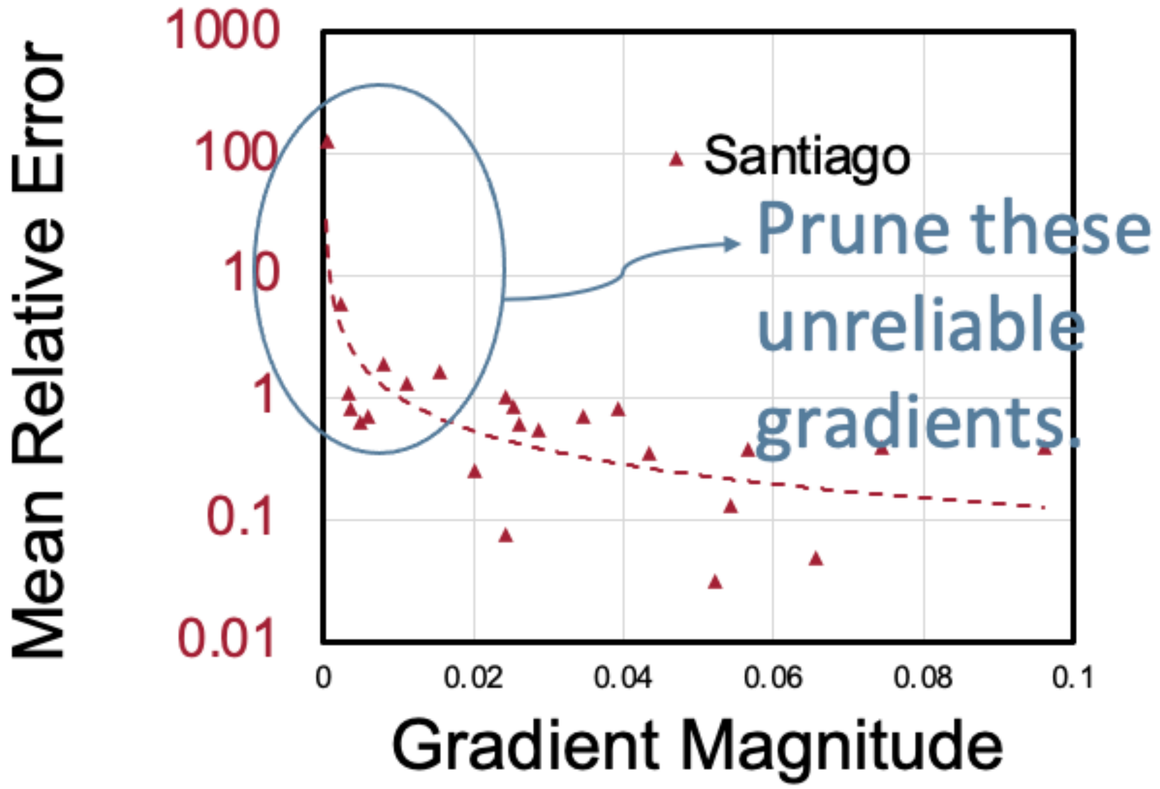
Challenge of On-chip Training: noise

- Noise **reduces reliability** of on-chip computed gradients



Challenge of On-chip Training: noise

- Noise **reduces reliability** of on-chip computed gradients
- **Small** magnitude gradients have **large** relative errors



Torch Quantum

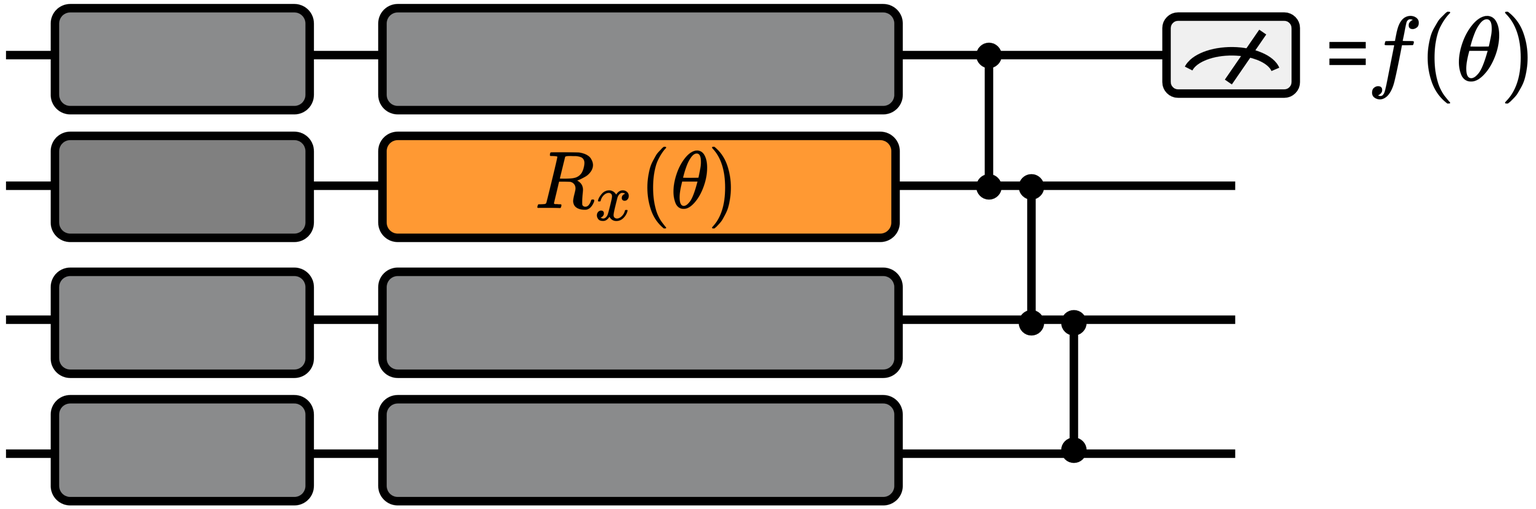
Materials at: https://torchquantum.org

MIT HAN LAB

ICCAD 2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Parameter Shift Rules

- Calculate the gradient of θ w.r.t. $f(\theta)$.

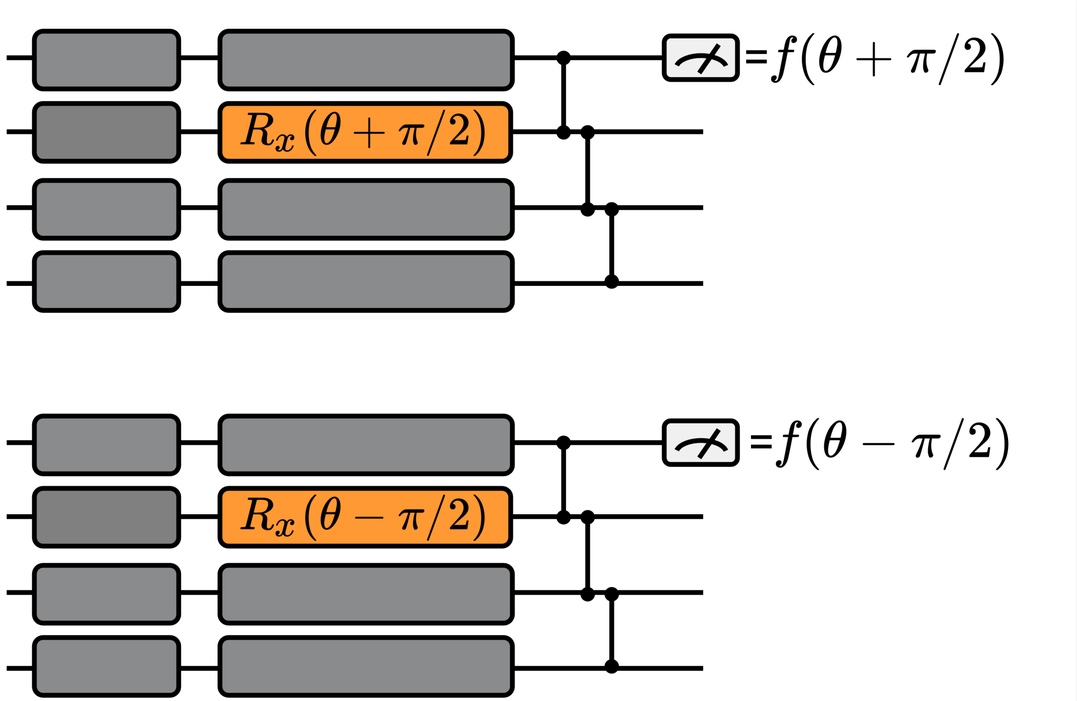


Parameter Shift Rules

- Calculate the gradient of θ w.r.t. $f(\theta)$.

Parameter Shift Rules

- Shift θ twice



$$\frac{\partial}{\partial\theta} f(\theta) = \frac{1}{2}\left(f\left(\theta + \frac{\pi}{2}\right) - f\left(\theta - \frac{\pi}{2}\right)\right)$$

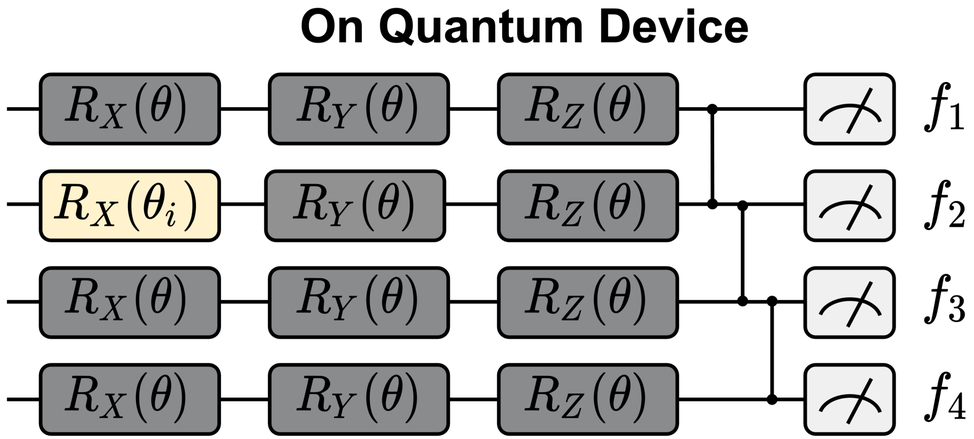
Parameter Shift Rules

- This formula is valid to all rotation gates
  - RZ, RY, RX, RXX, RZZ
- One gradient requires two runs on real quantum machine

$$\frac{\partial}{\partial\theta} f(\theta) = \frac{1}{2}\left(f\left(\theta + \frac{\pi}{2}\right) - f\left(\theta - \frac{\pi}{2}\right)\right)$$

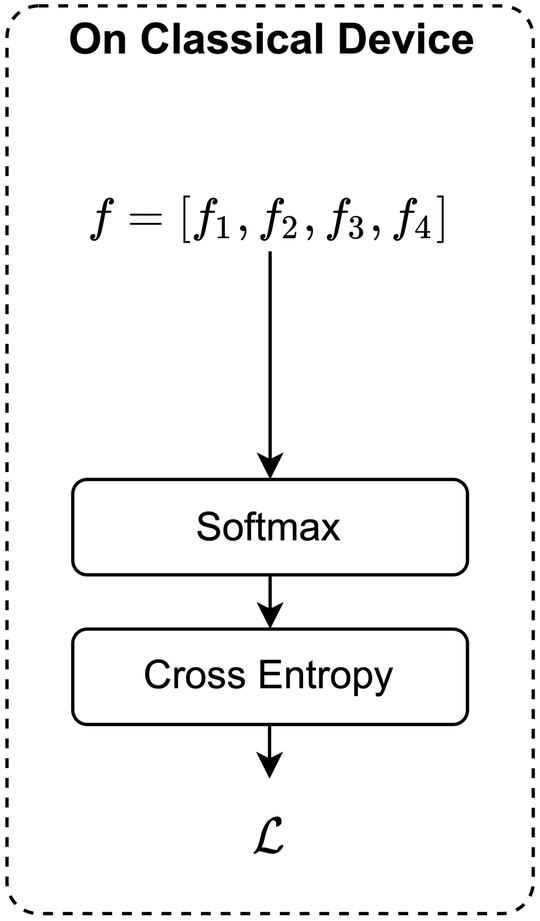
Calculate Gradients of PQC

- Step 1: Run on QC without shift to obtain f



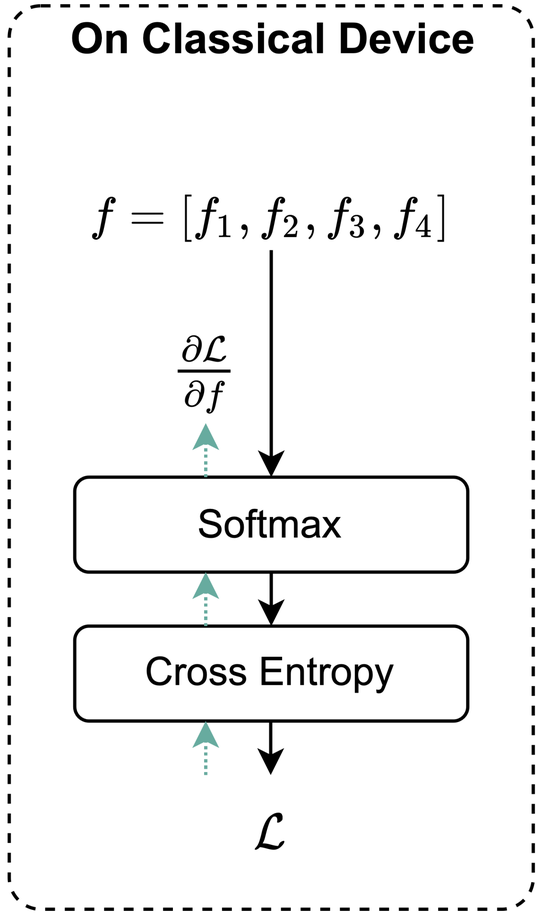
Calculate Gradients of PQC

- Step 2: Further forward to get $Loss$



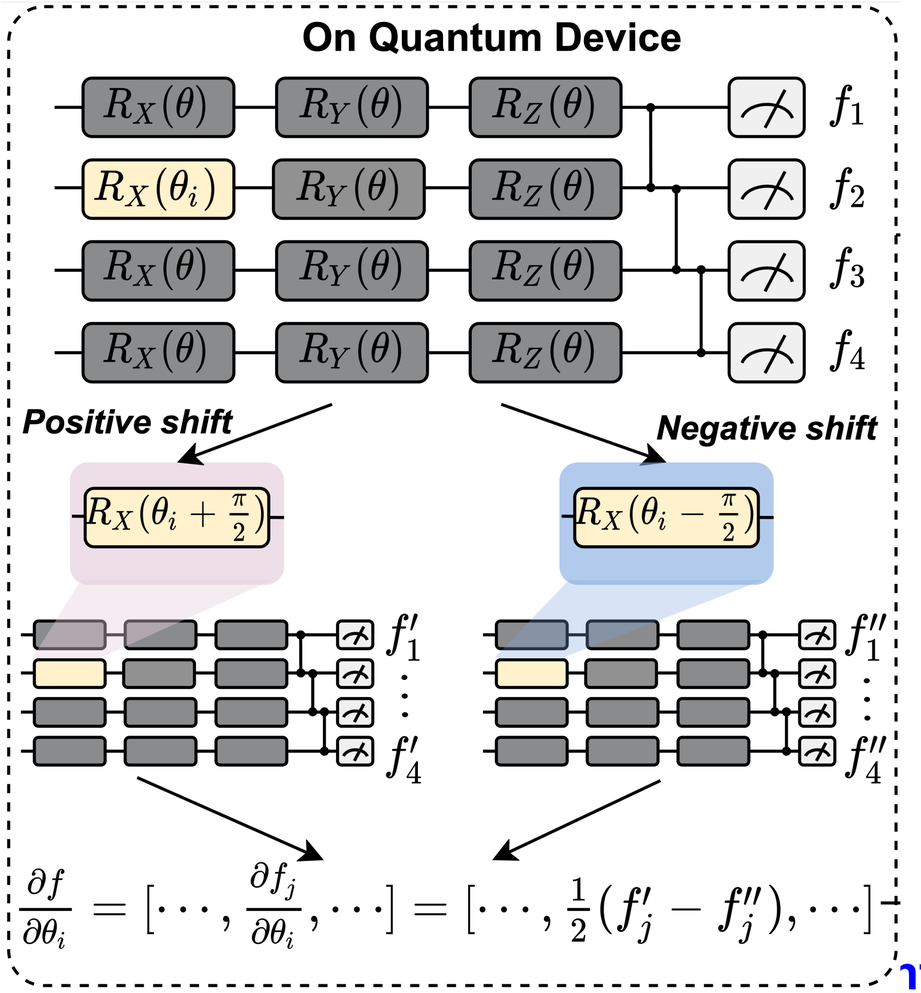
Calculate Gradients of PQC

- Step 3: Backpropagation to calculate $\frac{\partial Loss}{\partial f(\theta)}$



Calculate Gradients of PQC

- Step 4: Shift twice and run on QC to calculate $\frac{\partial f(\theta)}{\partial \theta_i}$



Torch Quantum

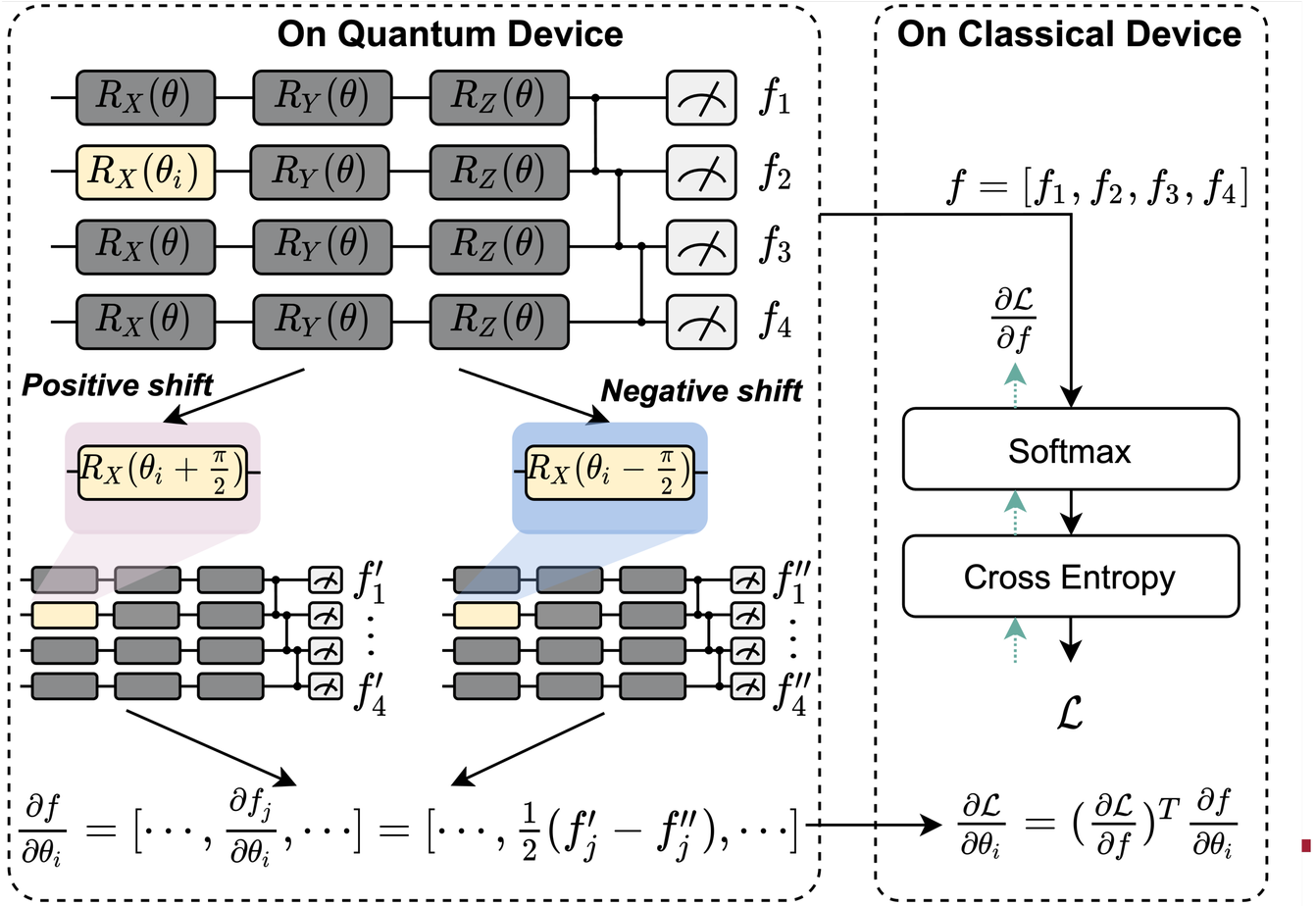
ntum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Calculate Gradients of PQC

- Step 5: By Chain Rule: $\frac{\partial Loss}{\partial f(\theta)}\frac{\partial f(\theta)}{\partial \theta_i} = \frac{\partial Loss}{\partial \theta_i}$, sum over 4 passes (4 qubits)
- Only **forward** on quantum device



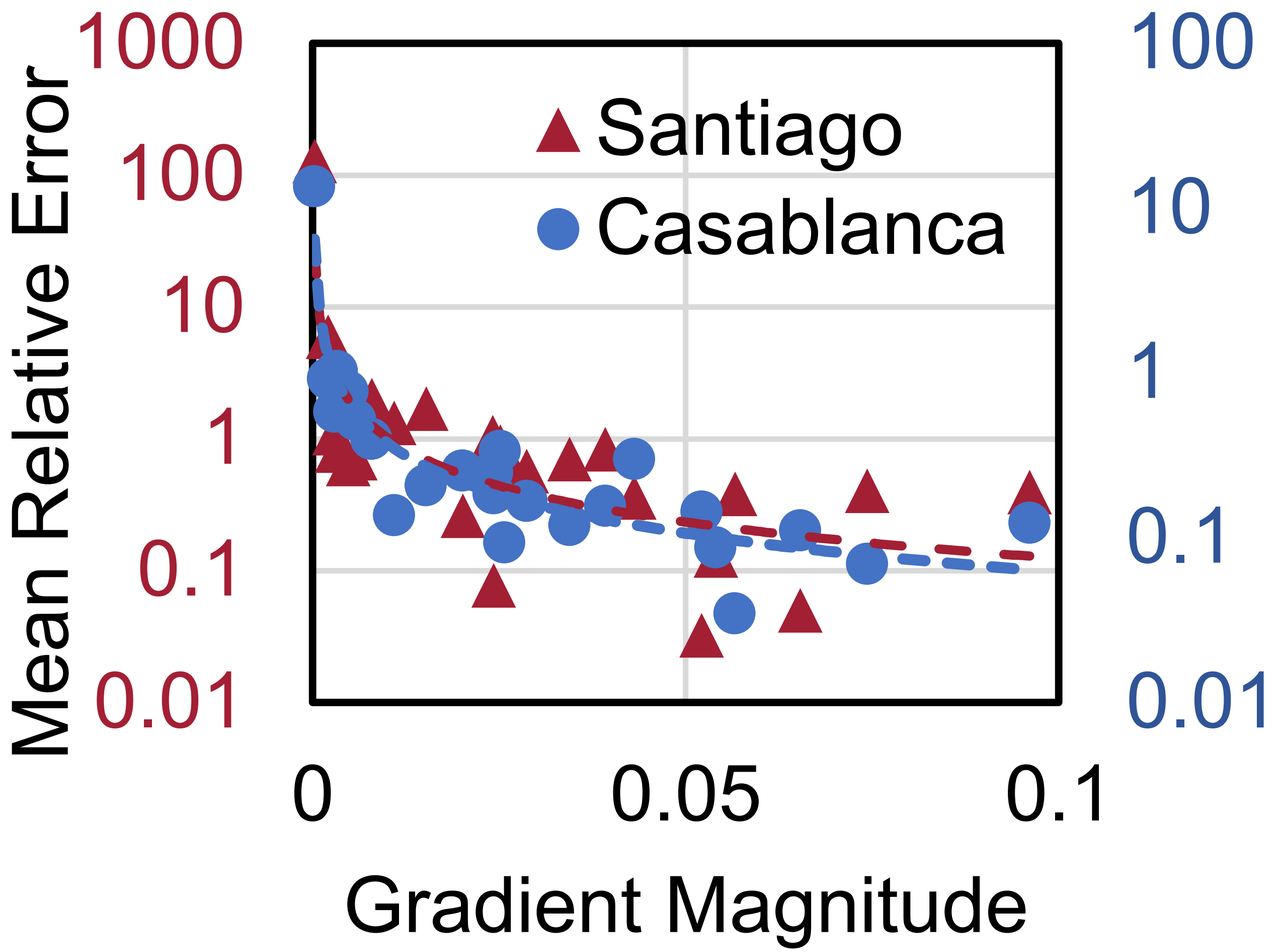
Torch Quantum

HANLAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN ICCAD 41st Edition

Probabilistic Gradient Pruning

- **Small** magnitude gradients have **large** relative errors



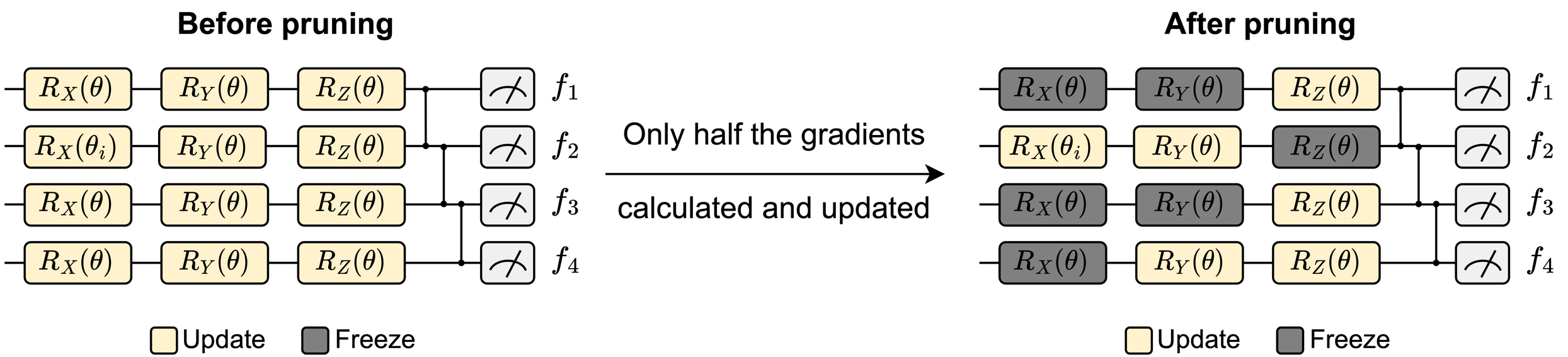
Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Probabilistic Gradient Pruning



Torch Quantum

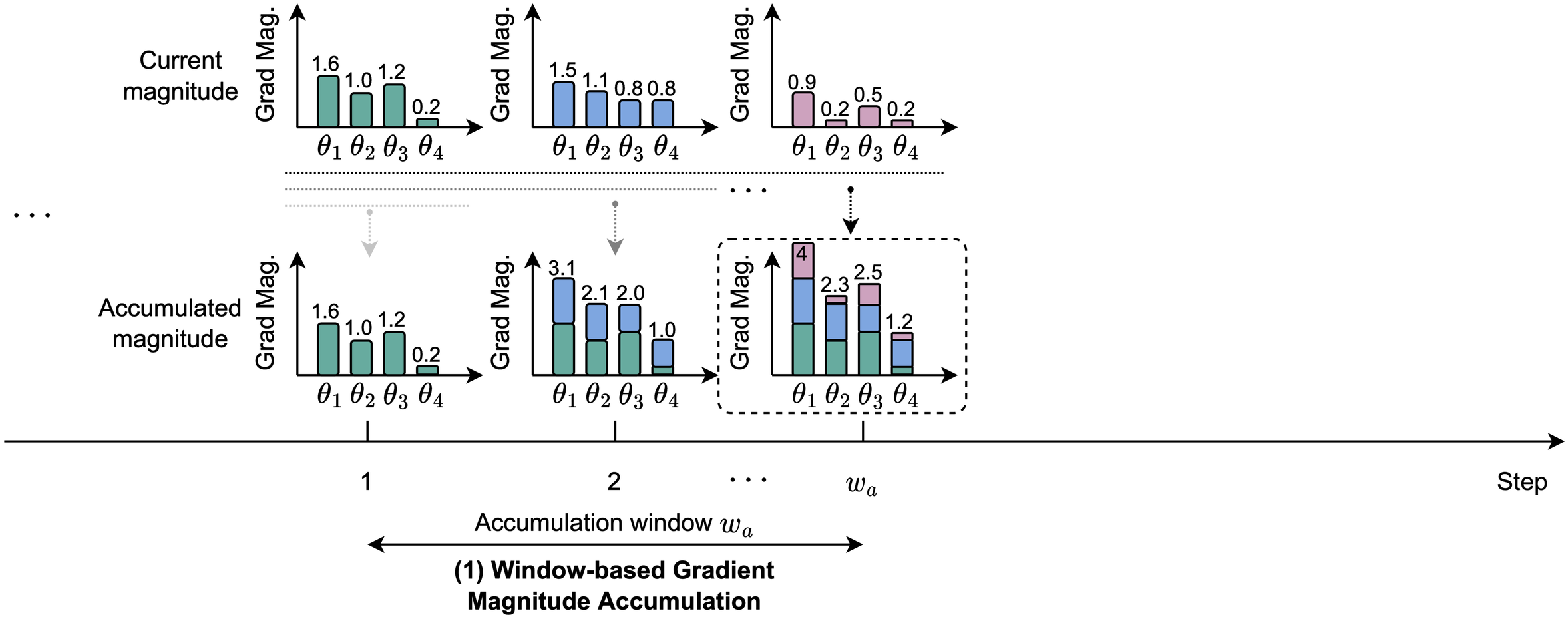
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

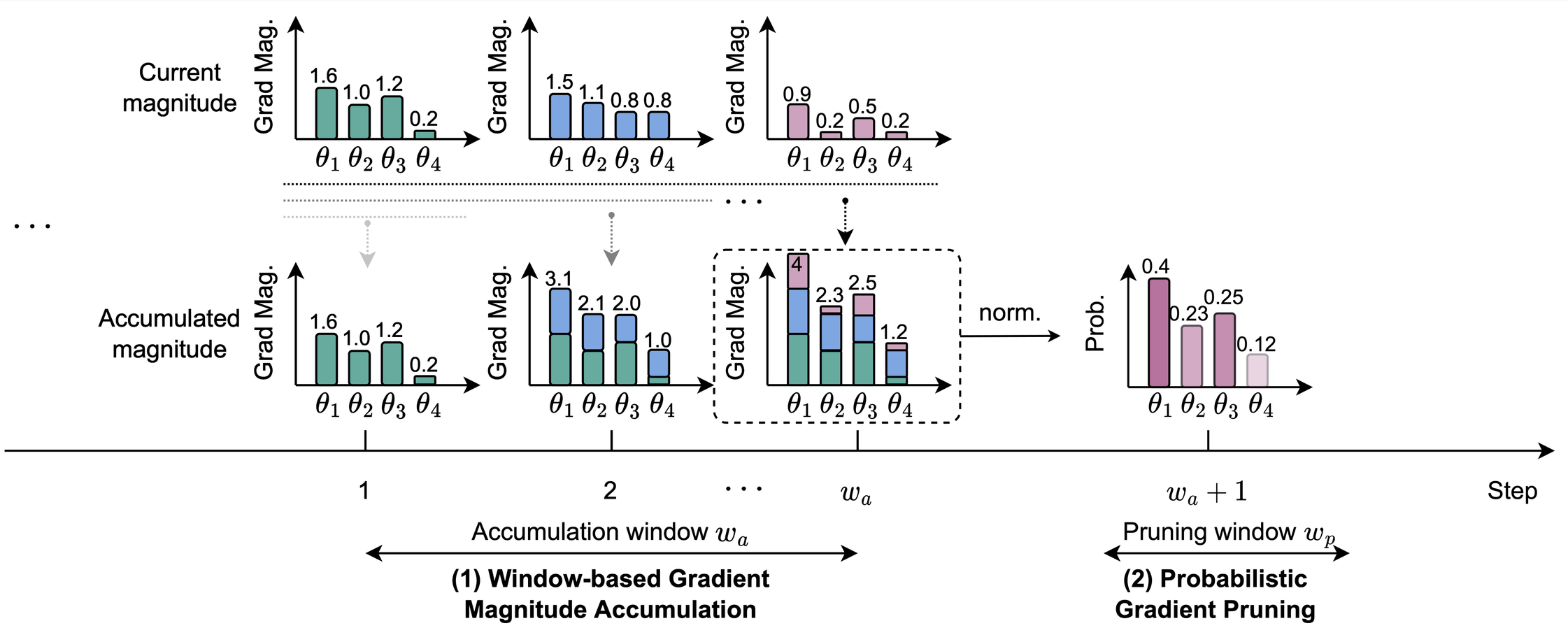
Accumulation Window

- Keep a record of accumulated gradient magnitude



Accumulation Window

- Normalize the accumulated gradient magnitude to a probability distribution



Torch Quantum

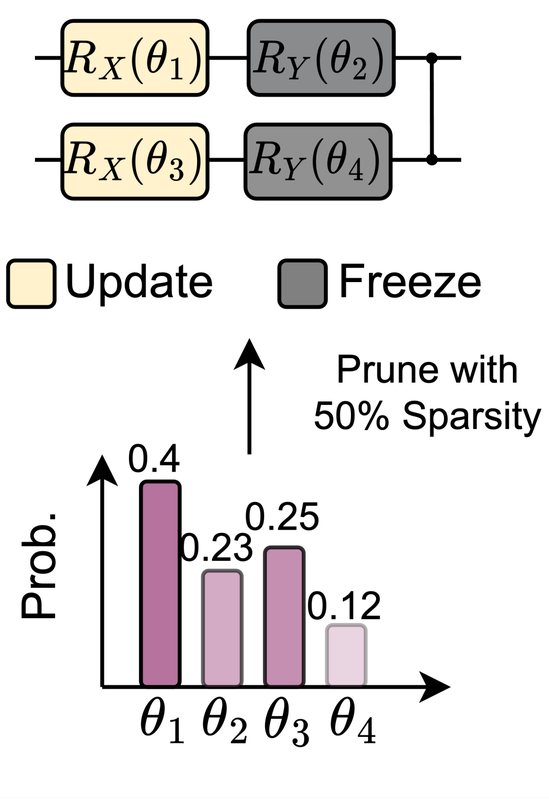
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Accumulation Window

- Prune the calculation of some gradients according to the probability distribution



Evaluation

- Benchmarks
  - Quantum Machine Learning task: MNIST 4-class, 2-class, Fashion MNIST 4-class, 2-class, Vowel 4-class
  - Variational Quantum Eigensolver task: H2 molecule
- Quantum Devices
  - IBMQ
  - #Qubits: 5 to 7
  - Quantum Volume: 8 to 32
- Circuit architecture
  - RZZ+RY, RXYZ+CZ, RZX+RXX

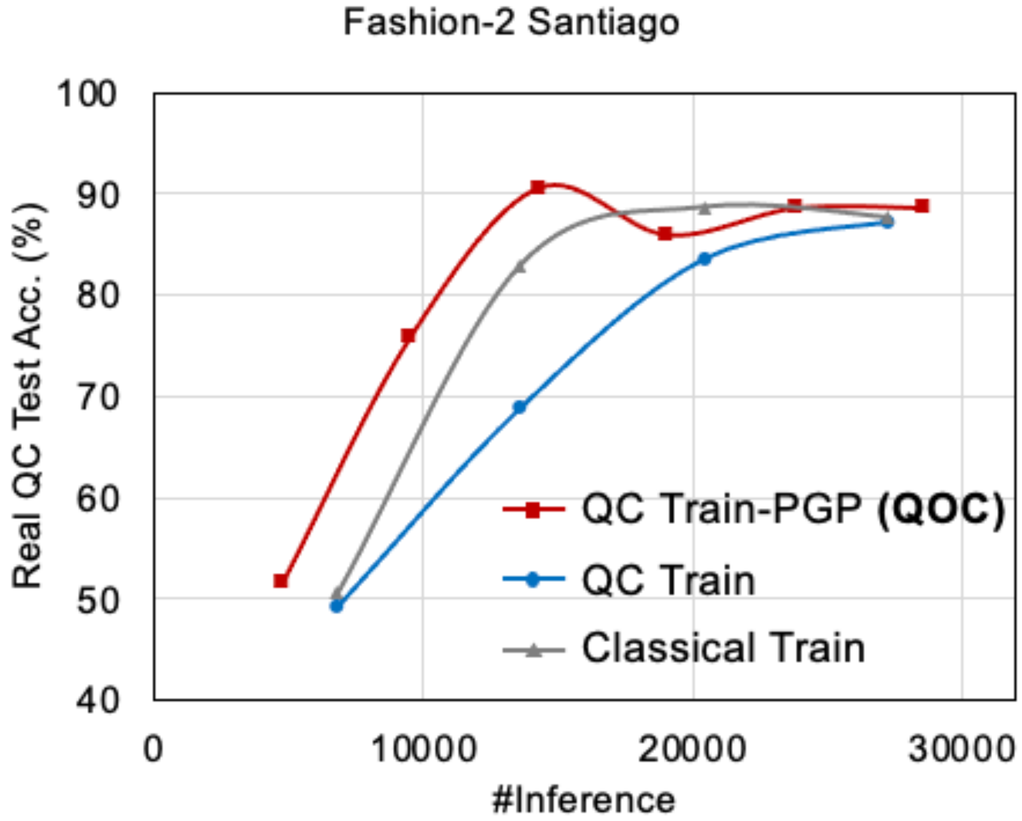
QNN results

- QOC achieved similar results to classical simulation

Method	Acc.	MNIST-4 Jarkata	MNIST-2 Jarkata	Fashion-4 Manila	Fashion-2 Santiago	Vowel-4 Lima
Classical-Train	Simu.	0.61	0.88	0.73	0.89	0.37
Classical-Train	QC	0.59	0.79	0.54	0.89	0.31
QC-Train		0.59	0.83	0.49	0.84	0.34
QC-Train-PGP		**0.64**	**0.86**	**0.57**	**0.91**	**0.36**

QNN Training Curves

- Classical Train: Train on classical simulator and test on real QC
- QC Train: Train and test the model on real QC
- QC Train-PGP (**QOC**): Train and test on real QC with gradient pruning



Materials at: https://torchquantum.org

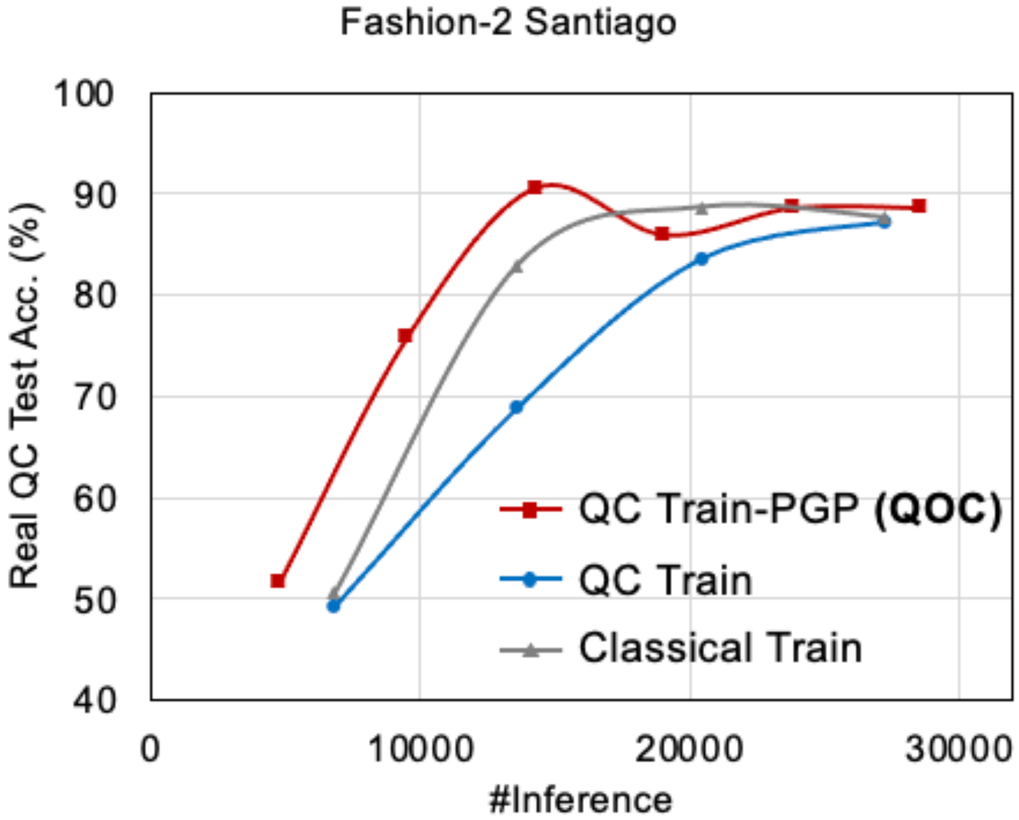
Torch Quantum

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

QNN Training Curves

- Gradient pruning can brings **2%~4%** accuracy improvements
- Pruning **accelerates convergence** with **2x** training time reduction



Materials at: https://torchquantum.org

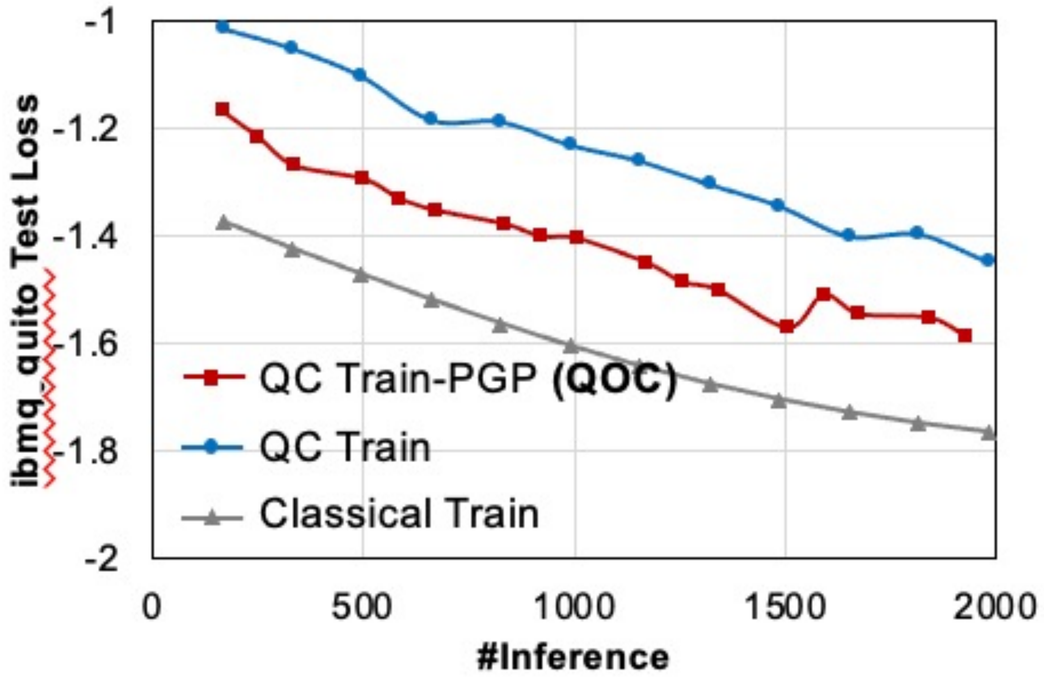
Torch Quantum

MIT HAN LAB

2022 International Conference on Computer-Aided Design

VQE Training Curves

- Gradient pruning can **reduce the gap** between quantum and classical



Torch Quantum

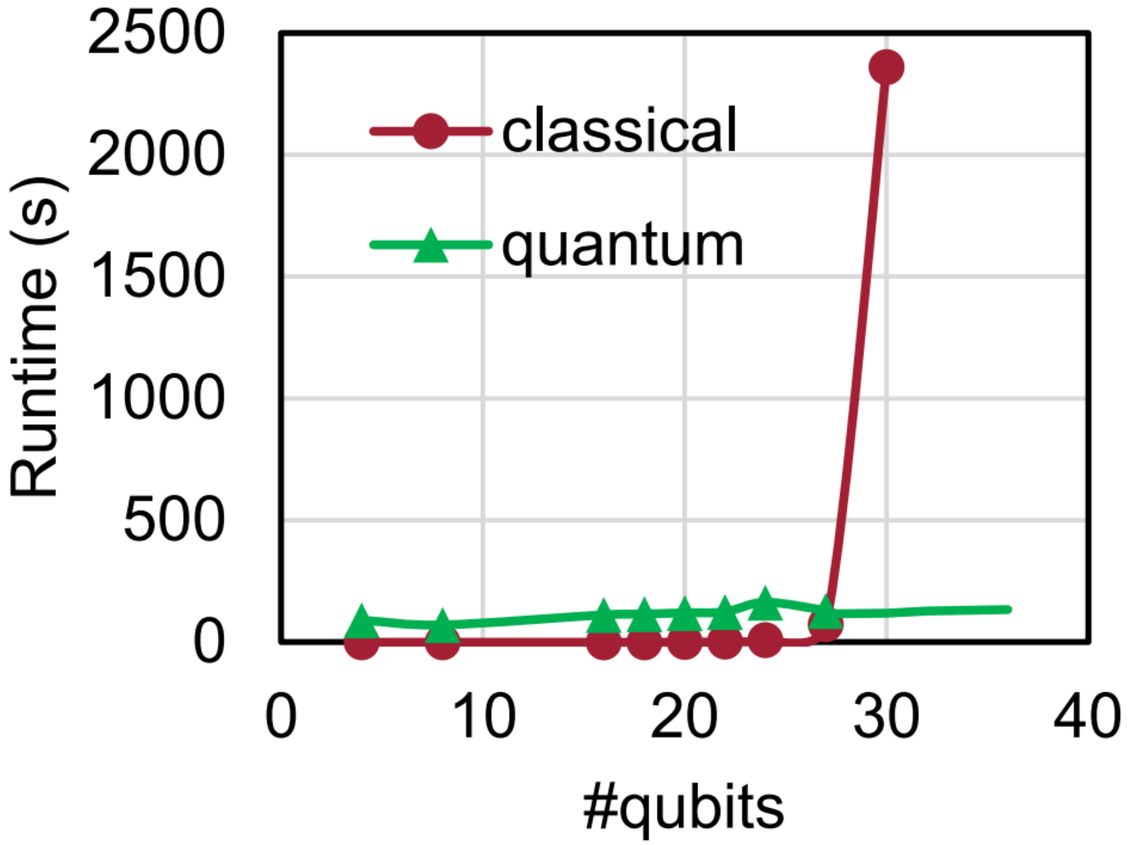
Materials at: https://torchquantum.org

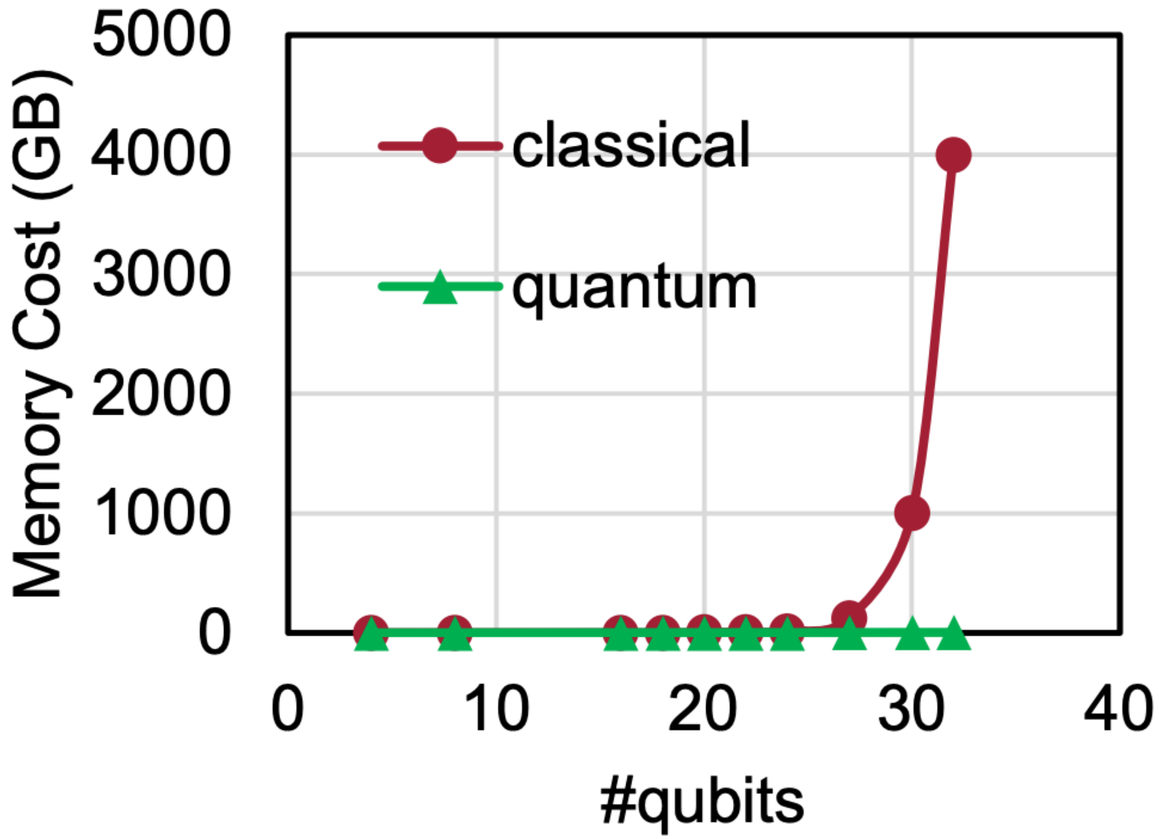
MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Scalability Improvements

- On-chip Training is scalable





Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations
- 1.3 TQ Use Examples
- 1.4 QNN Compression

Section 2

Use TorchQuantum on Pulse Level Optimization

- 3.1 Quantum Optimal Control
- 3.2 Variational Pulse Learning

Section 3

Use TorchQuantum on Gate Level Optimization

- 3.1 QuantumNAS: Ansatz Search and Gate Pruning
- 3.2 QuantumNAT: Noise Injection and Quantization
- 3.3 QOC: On-Chip Training
- 3.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

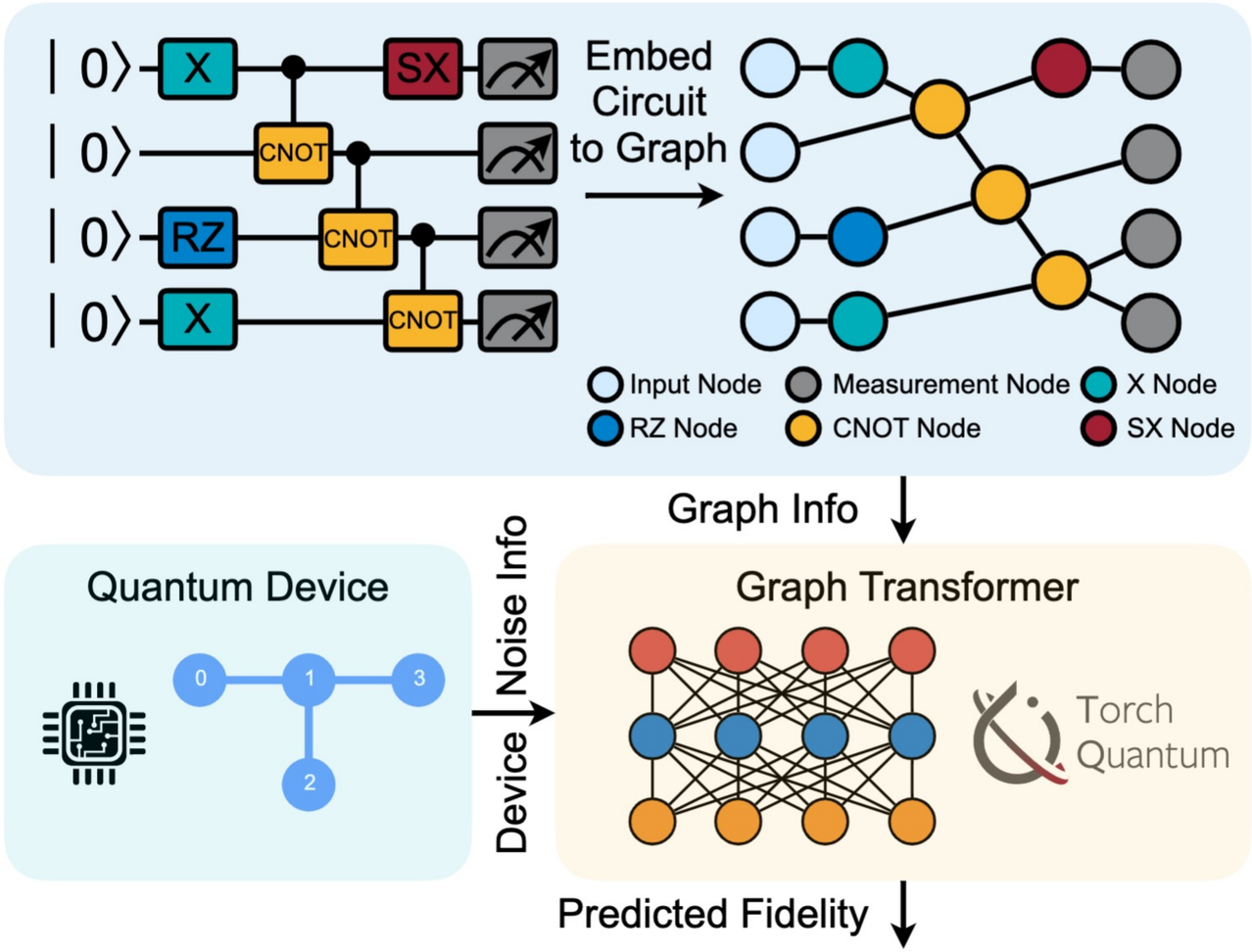
Materials at: https://torchquantum.org

MIT HAN LAB

IEEE/ACM ICCAD 2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Transformer for Quantum Circuit Reliability Prediction

- Use the circuit graph information



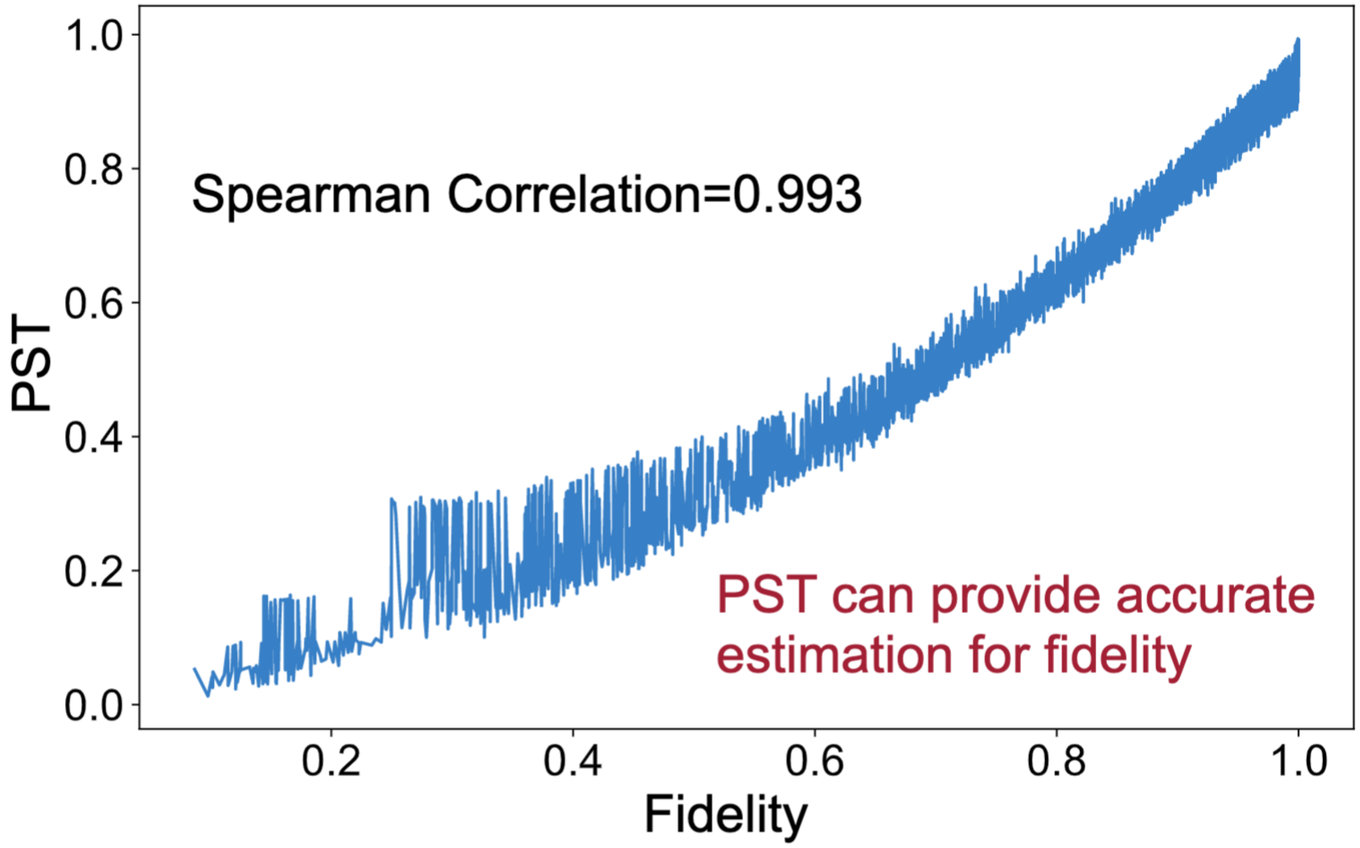
Torch Quantum

Materials at: https://torchquantum.org

Transformer for Quantum Circuit Reliability Prediction

- Use PST as the metrics

$$PST = \frac{\#Trials\ with\ output\ same\ as\ initial\ state}{\#Total\ trials}$$



Torch Quantum

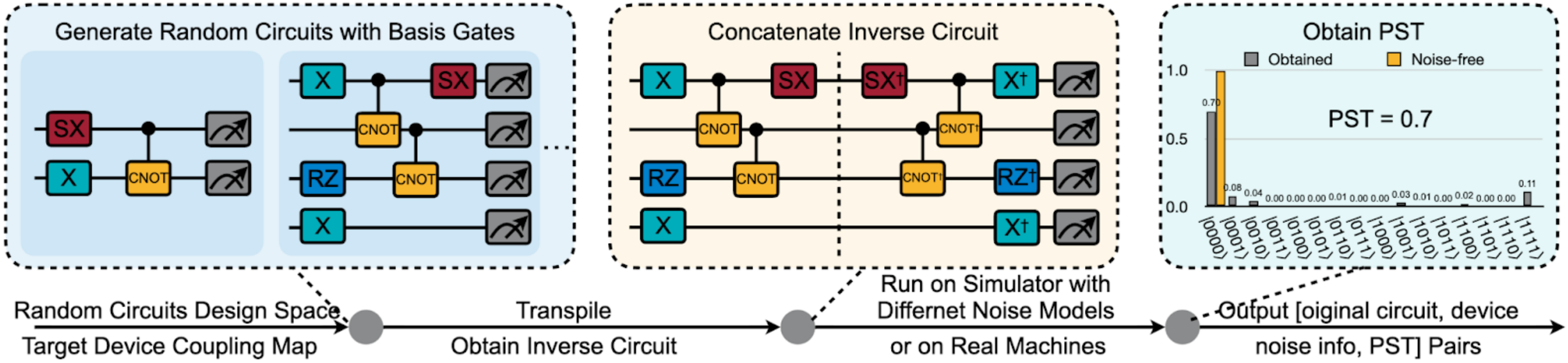
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Dataset collection

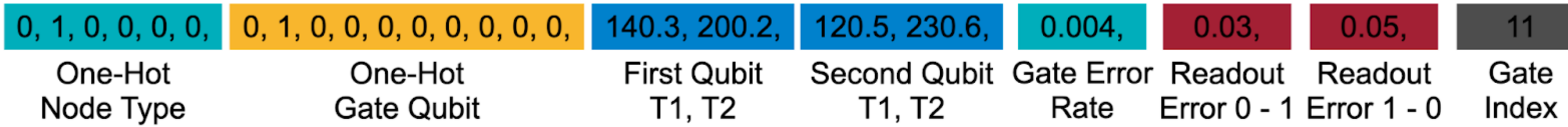
- Collect dataset on real machine / noisy simulator



Torch Quantum

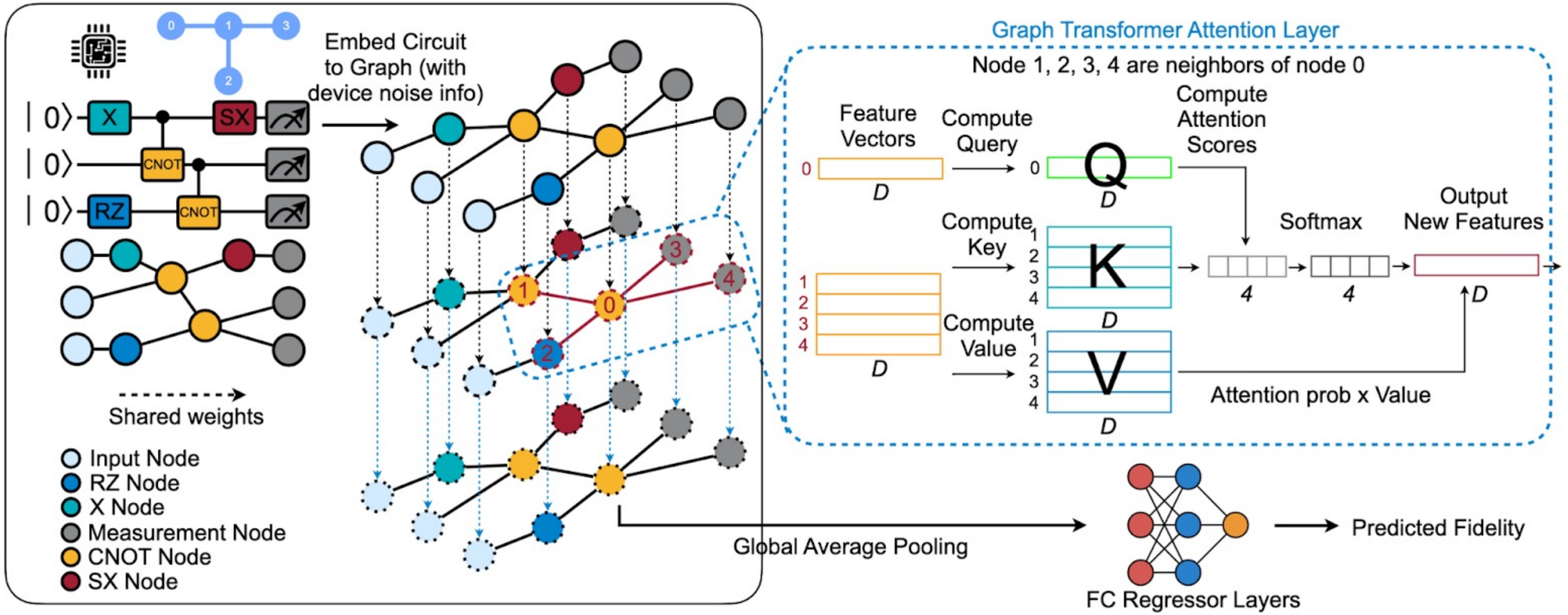
Materials at: https://torchquantum.org

Features on each node



Graph Transformer

- Graph Transformer



Torch Quantum

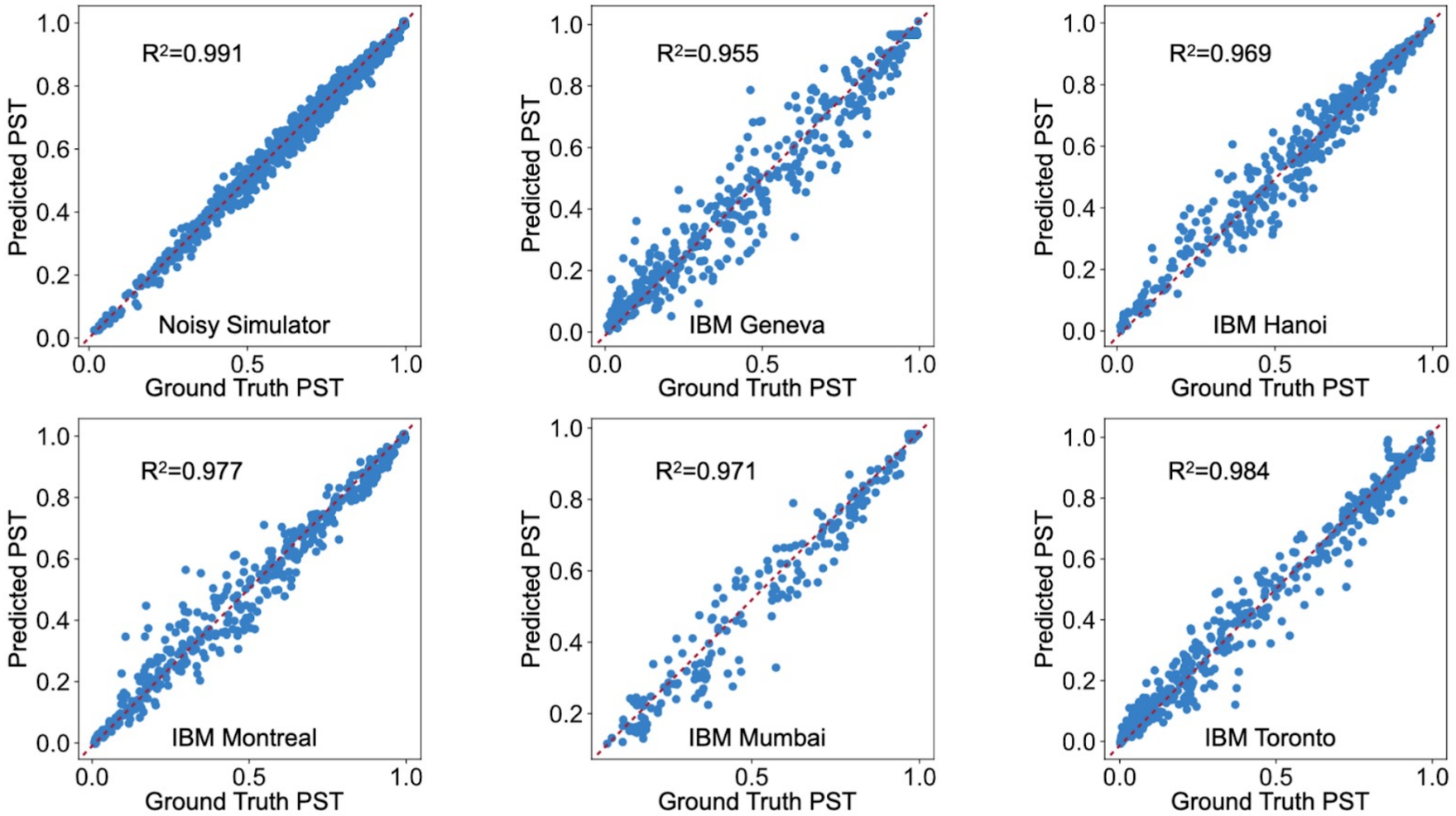
Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Evaluation

- On random generated circuit



Torch Quantum

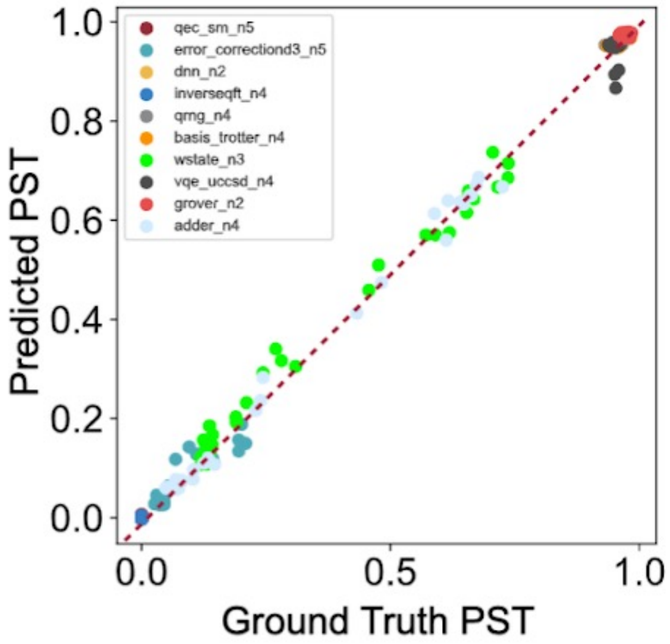
Materials at: https://torchquantum.org

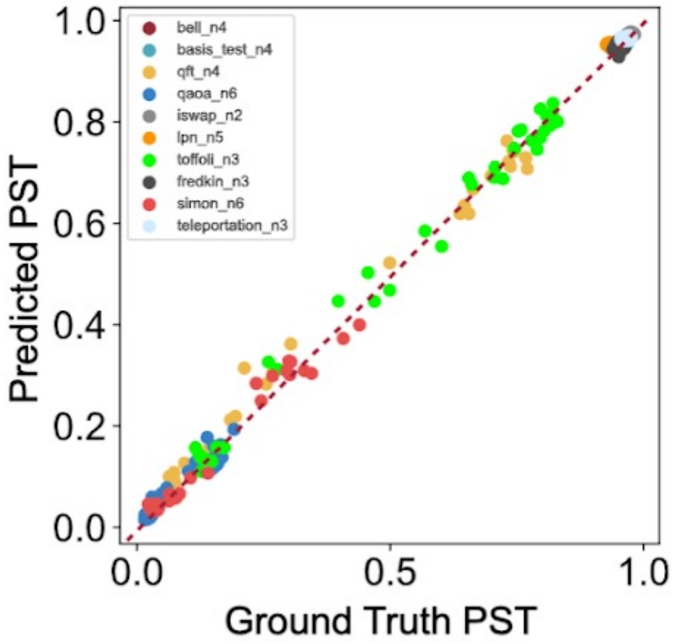
IBM QUANTUM

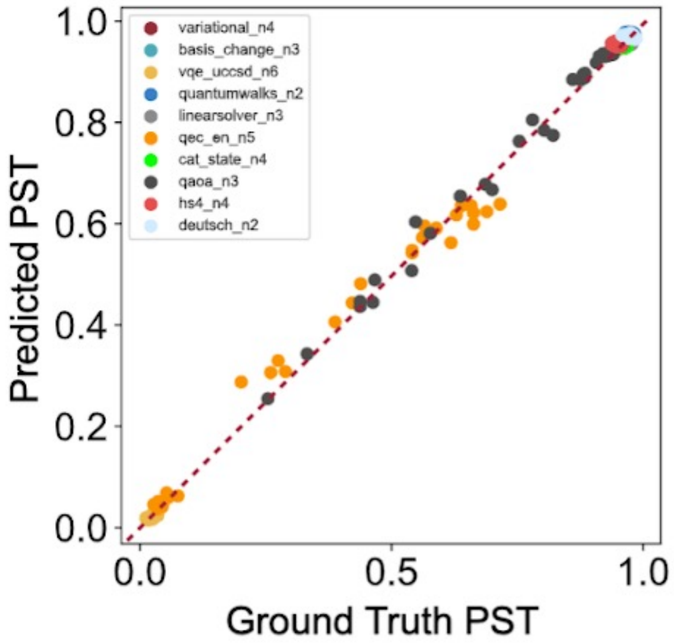
2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Evaluation

- Circuits from quantum algorithms







Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

TorchQuantum Tutorial Outline

Section 1

TorchQuantum Basic Usage

- 1.1 Quantum Basics
- 1.2 TQ Operations 
- 1.3 TQ Use Examples 
- 1.4 QNN Compression

Section 2

Use TorchQuantum on <u>Pulse</u> Level Optimization

- 2.1 Quantum Optimal Control
- 2.2 Variational Pulse Learning 

Section 3

Use TorchQuantum on <u>Gate</u> Level Optimization

- 3.1 QuantumNAS: Ansatz Search and Gate Pruning 
- 3.2 QuantumNAT: Noise Injection and Quantization
- 3.3 QOC: On-Chip Training
- 3.4 Transformer for Quantum Circuit Reliability Prediction

Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

2022 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 41st Edition

Thank you for listening!

Torch Quantum

https://github.com/mit-han-lab/torchquantum

qmlsys.mit.edu