qmlsys.mit.edu

QuantumNAS: Noise-aware search and training for robust quantum circuits

Hanrui Wang
MIT HAN Lab

[HPCA'22] QuantumNAS: Noise-adaptive search for robust quantum circuits

[DAC'22] QuantumNAT: Quantum Noise-Aware Training with Noise Injection, Quantization and Normalization

[DAC'22] QOC: Quantum On-Chip Training with Parameter Shift and Gradient Pruning

Torch Quantum

2022/05/12 @ Tongyang Li Group

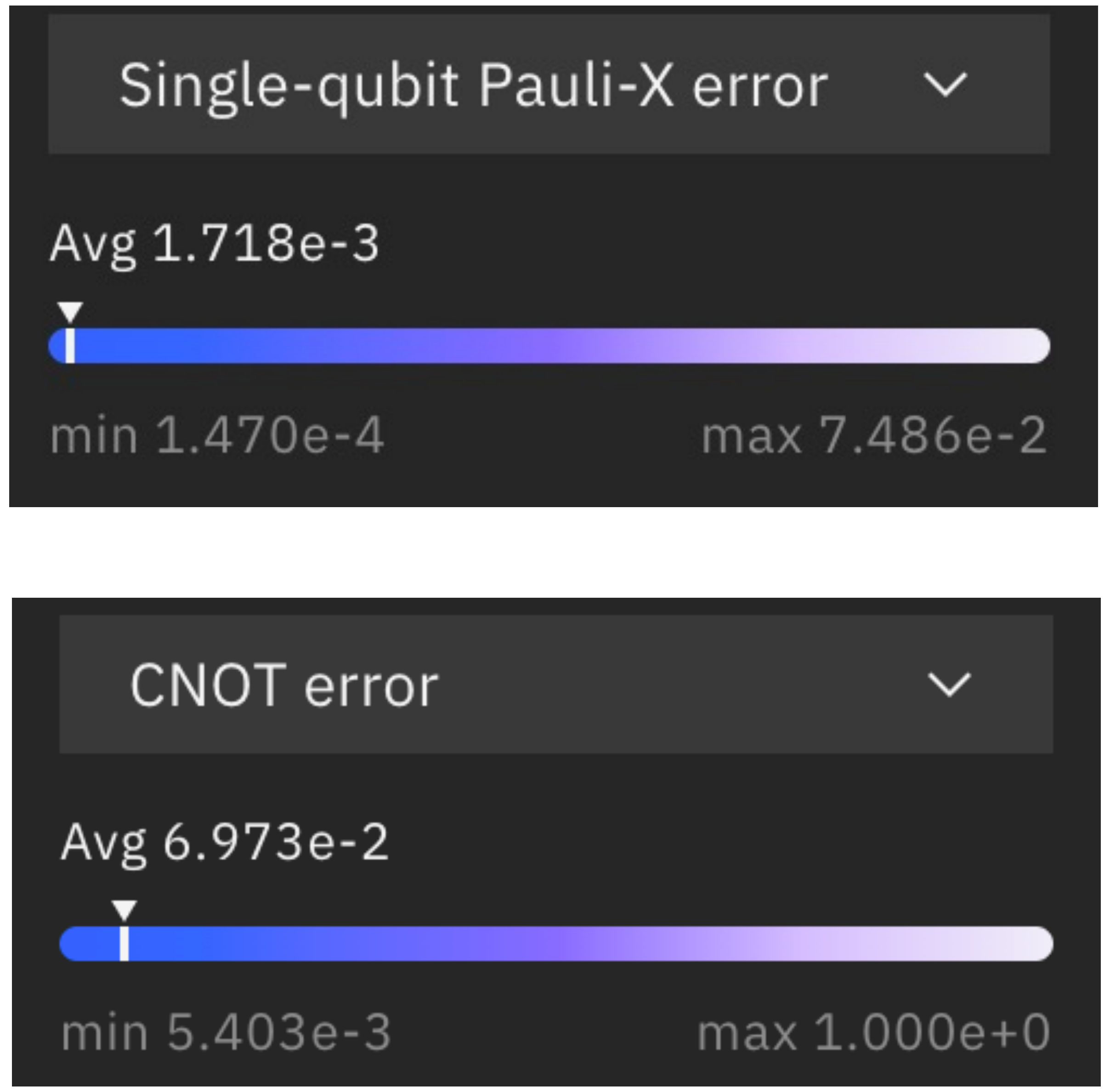
MIT HAN LAB

Outline

- Background
- QuantumNAS
- TorchQuantum Library
- Conclusion

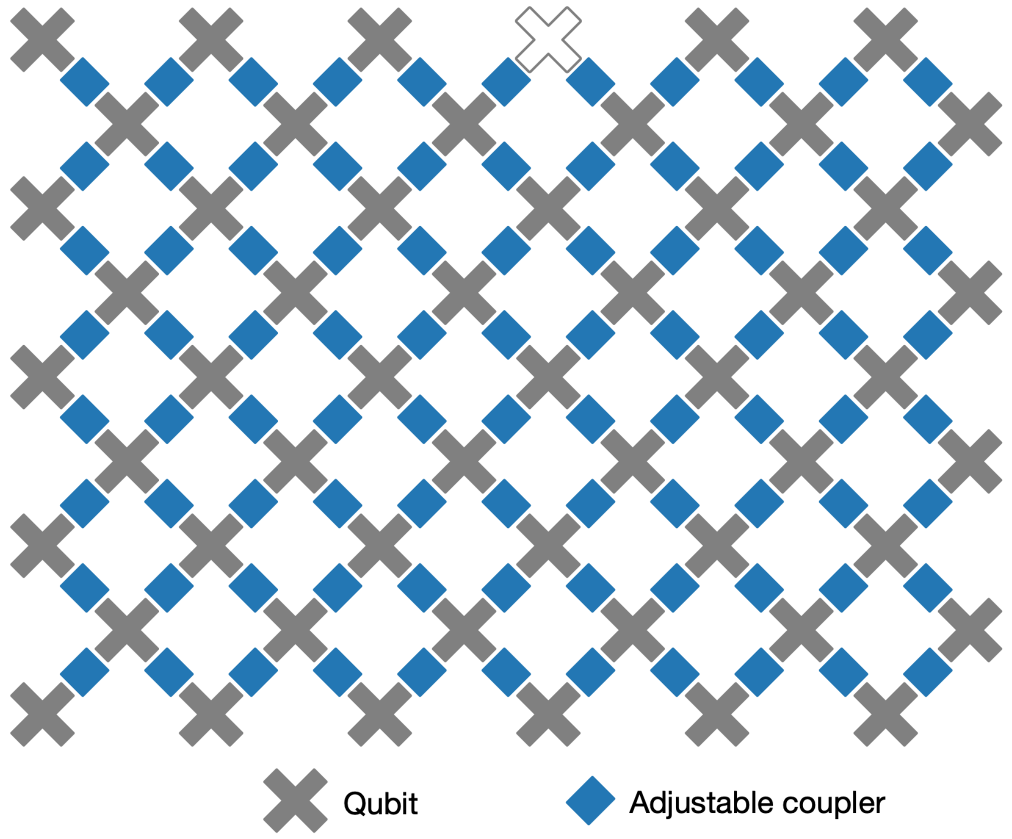
NISQ Era

- Noisy Intermediate-Scale Quantum (NISQ)
  - **Noisy**: qubits are sensitive to environment; quantum gates are unreliable
  - **Limited number** of qubits: tens to hundreds of qubits



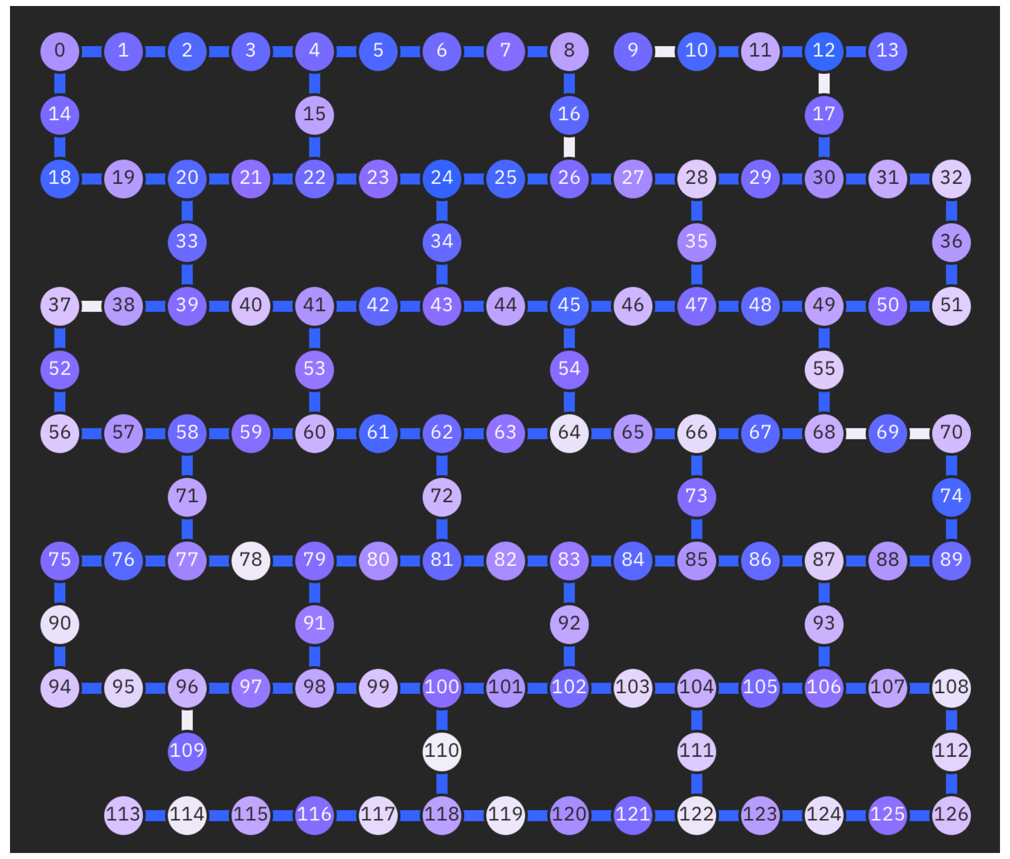
Gate Error Rate

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



Google Sycamore

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

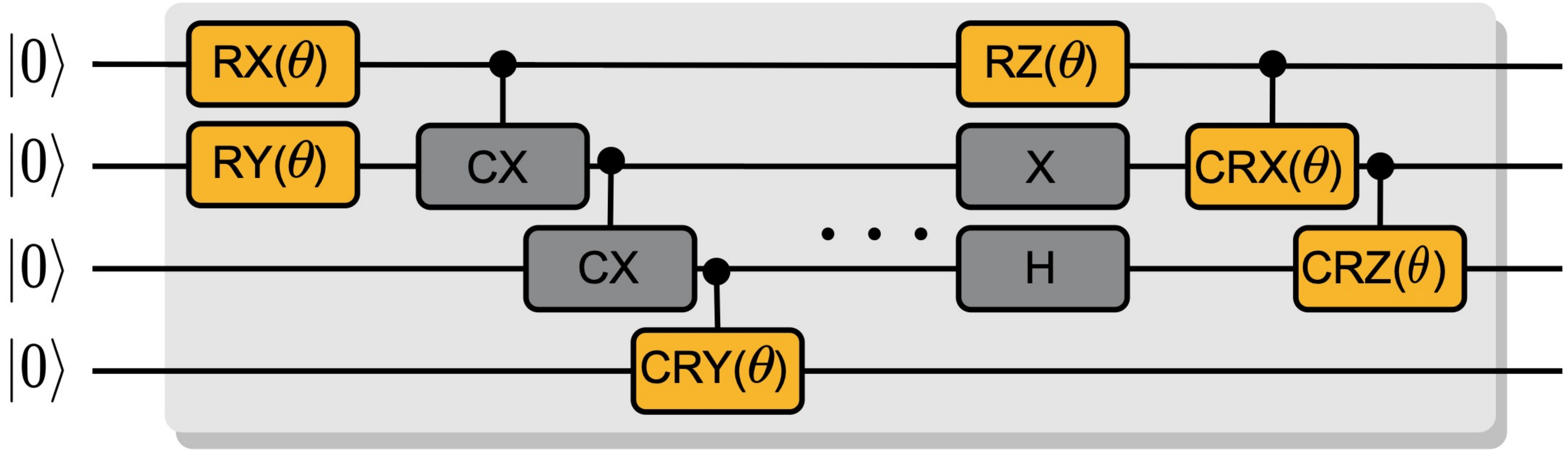


IBM Washington

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

Parameterized Quantum Circuits

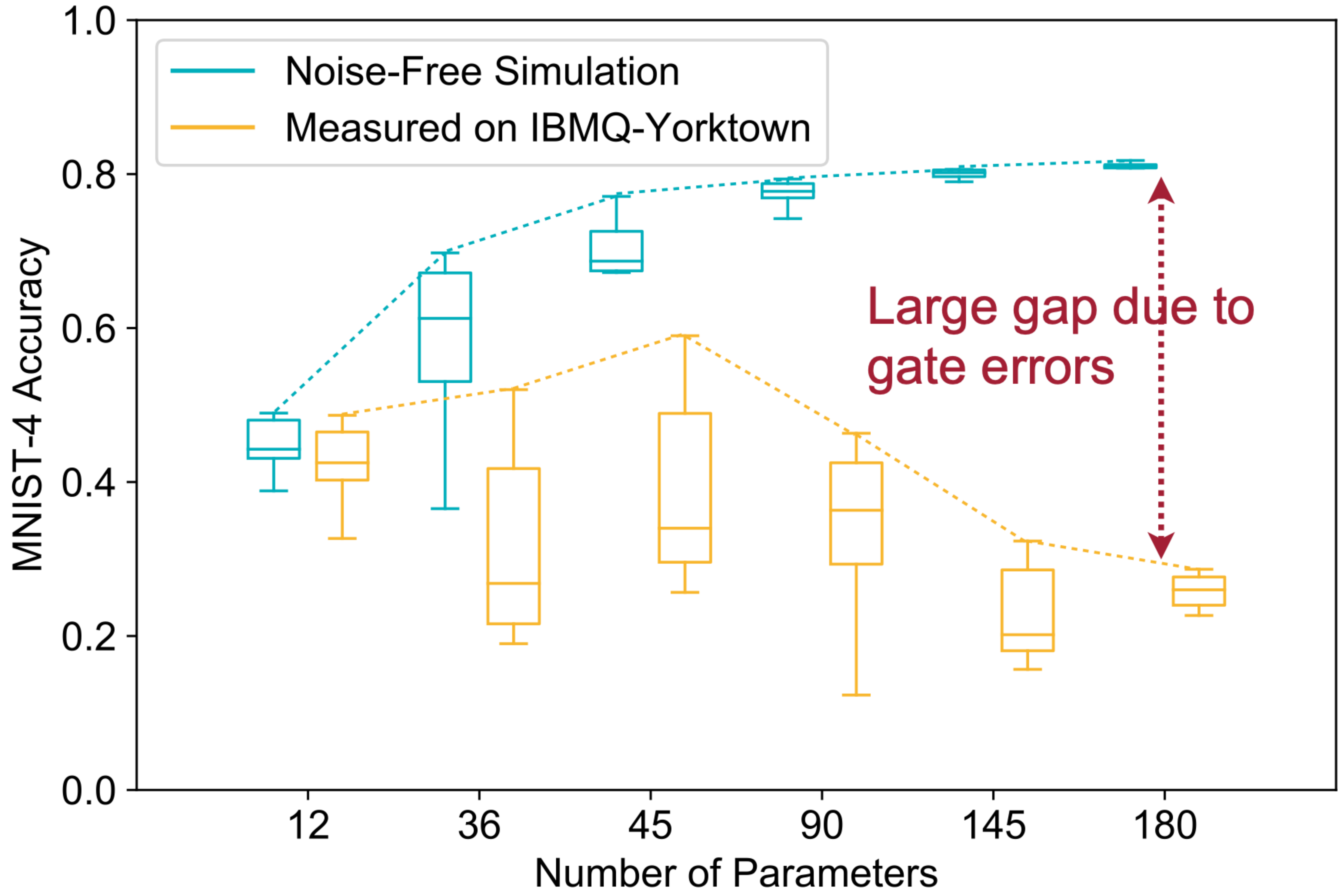
- Parameterized Quantum Circuits (PQC)
- Quantum circuit with fixed gates and parameterized gates



- PQCs are commonly used in **hybrid classical-quantum models** and show promises to achieve quantum advantage
  - Variational Quantum Eigensolver (VQE)
  - Quantum Neural Networks (QNN)
  - Quantum Approximate Optimization Algorithm (QAOA)

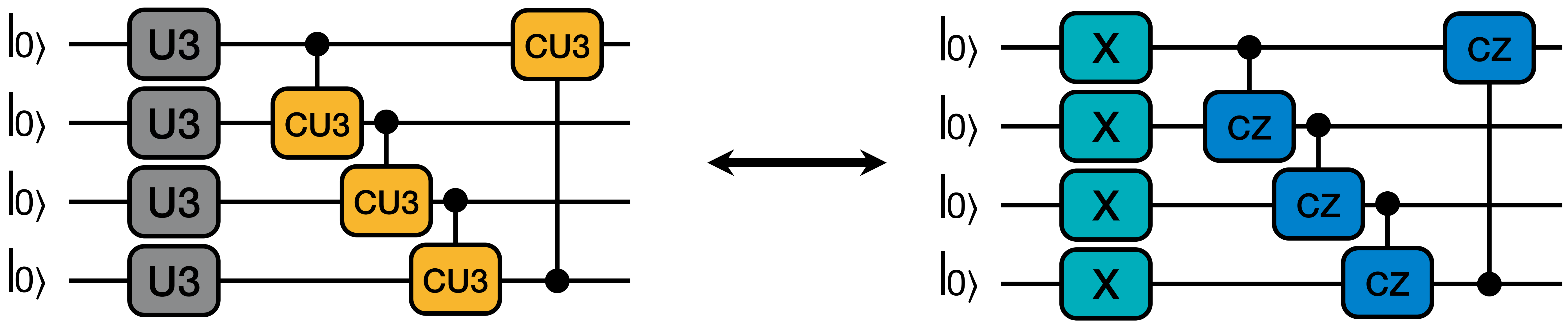
Challenges of PQC — Noise

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



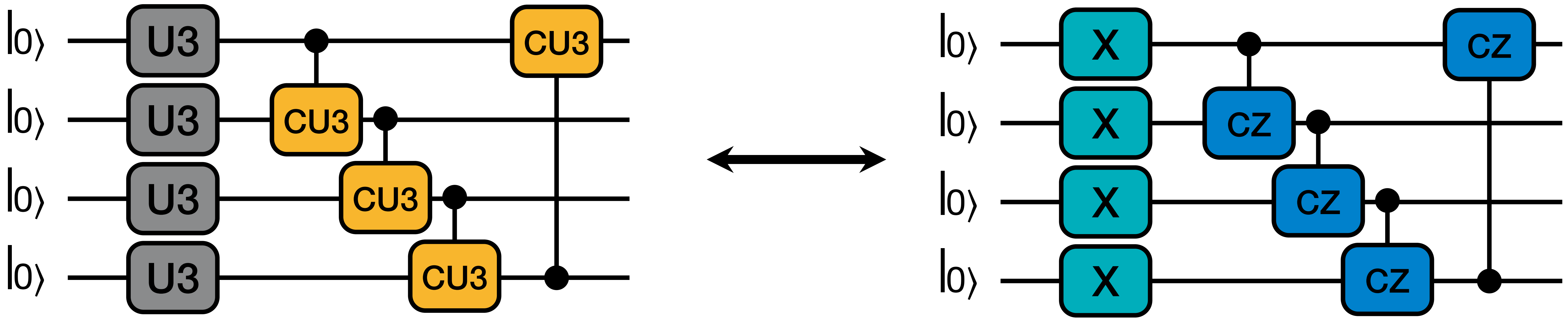
Challenges of PQC — Large Design Space

- Large design space for circuit architecture
  - Type of gates

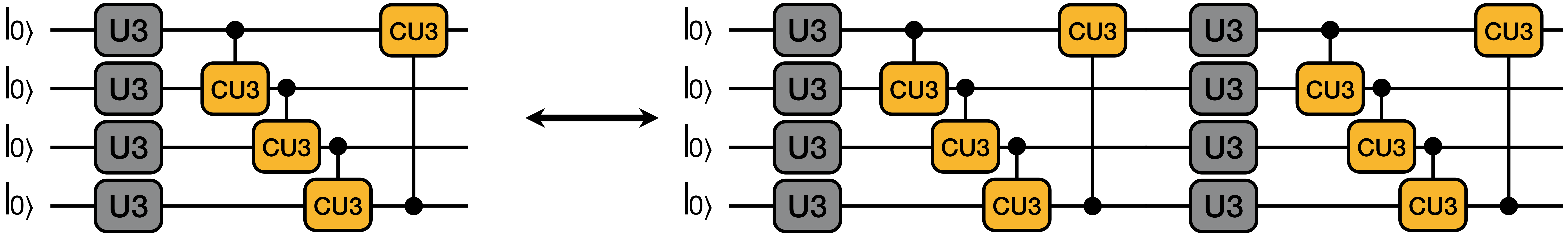


Challenges of PQC — Large Design Space

- Large design space for circuit architecture
  - Type of gates

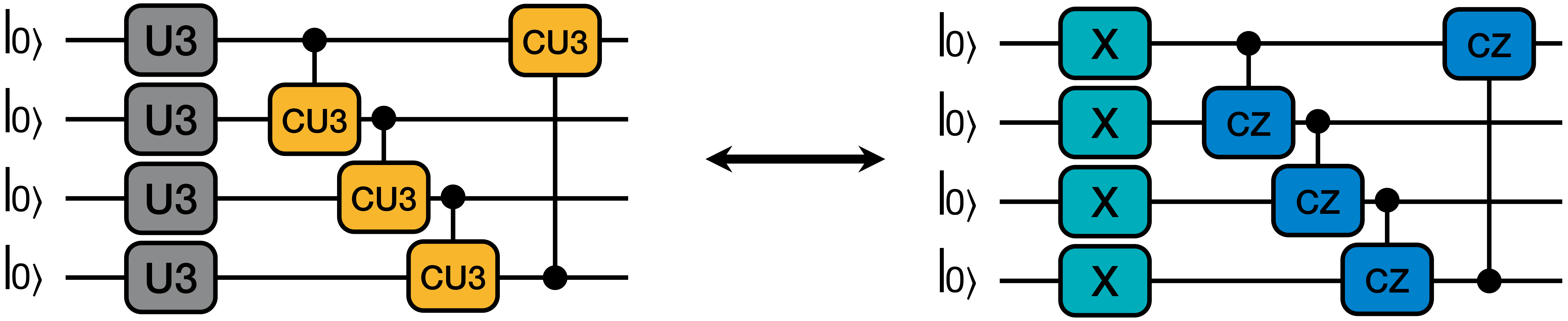


  - Number of gates

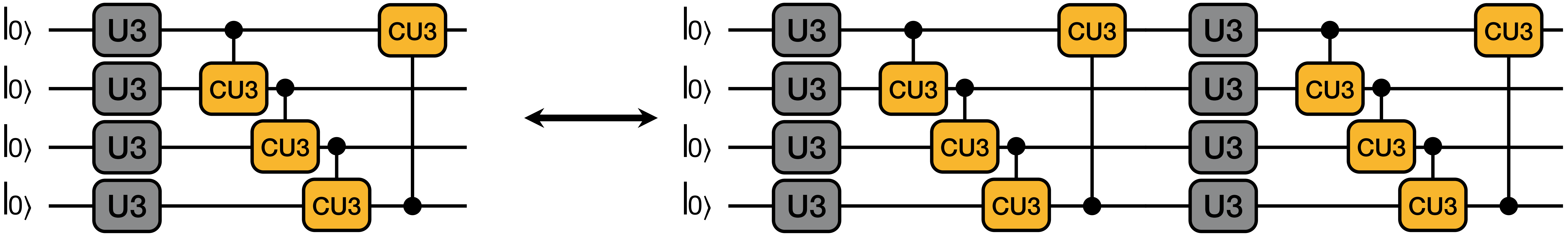


Challenges of PQC — Large Design Space

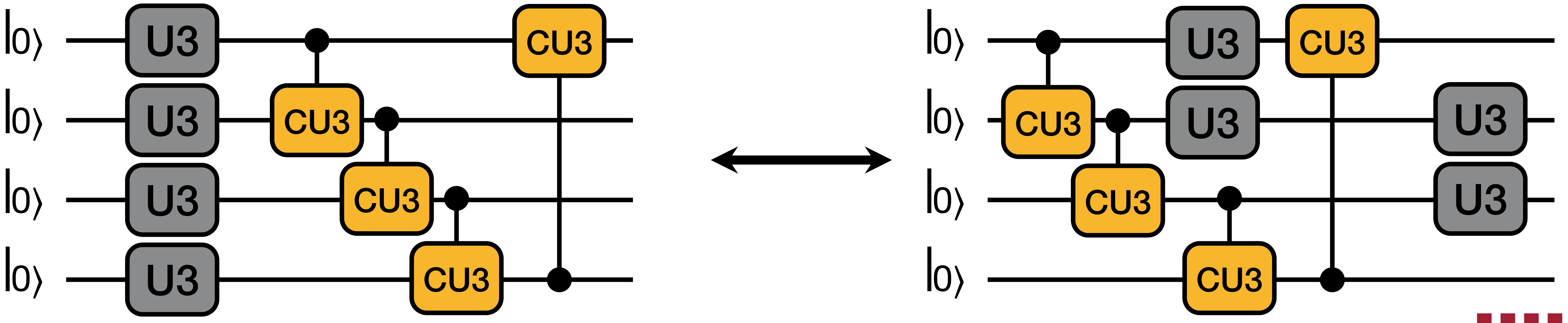
- Large design space for circuit architecture
  - Type of gates



  - Number of gates



  - Position of gates



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

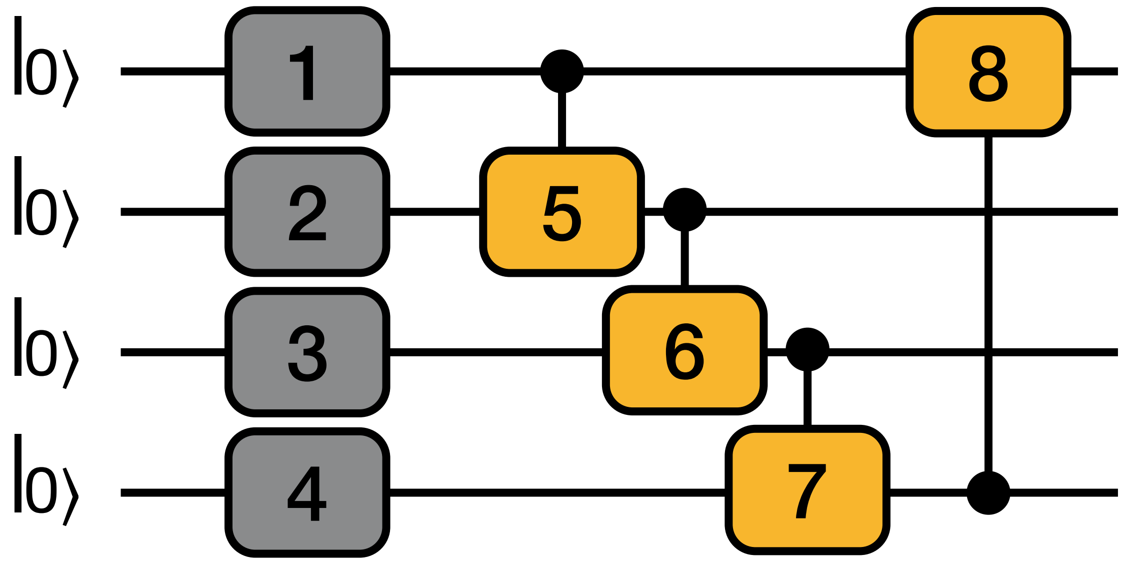
SuperCircuit & SubCircuit

- 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

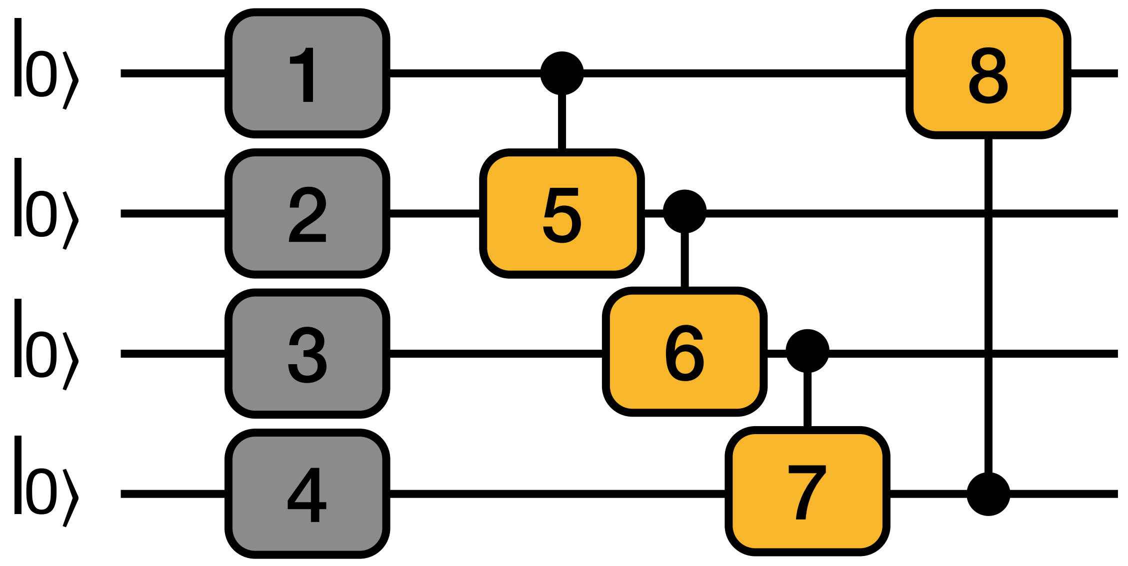
- SuperCircuit: the circuit with the **largest** number of gates in the design space
  - Example: SuperCircuit in U3+CU3 space



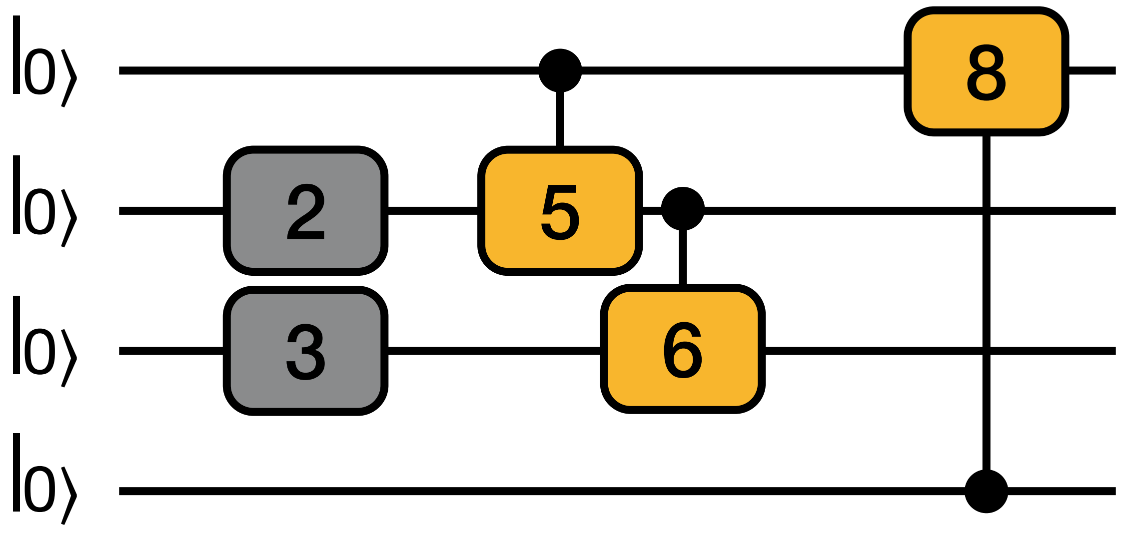
SuperCircuit & SubCircuit

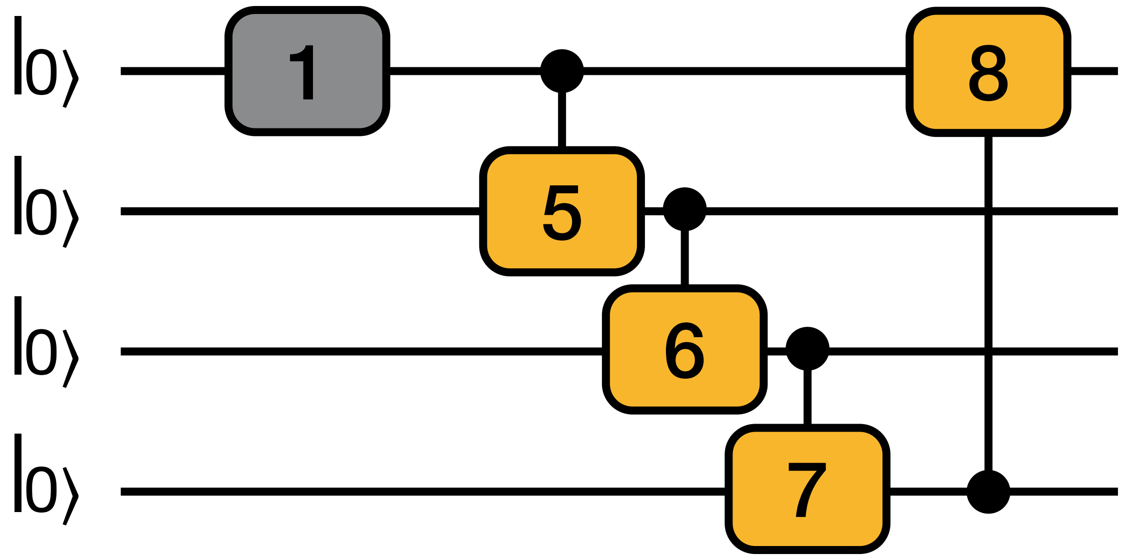
- Firstly construct a design space. For example, a design space of maximum 4 U3 in the first layer and 4 CU3 gates in the second layer

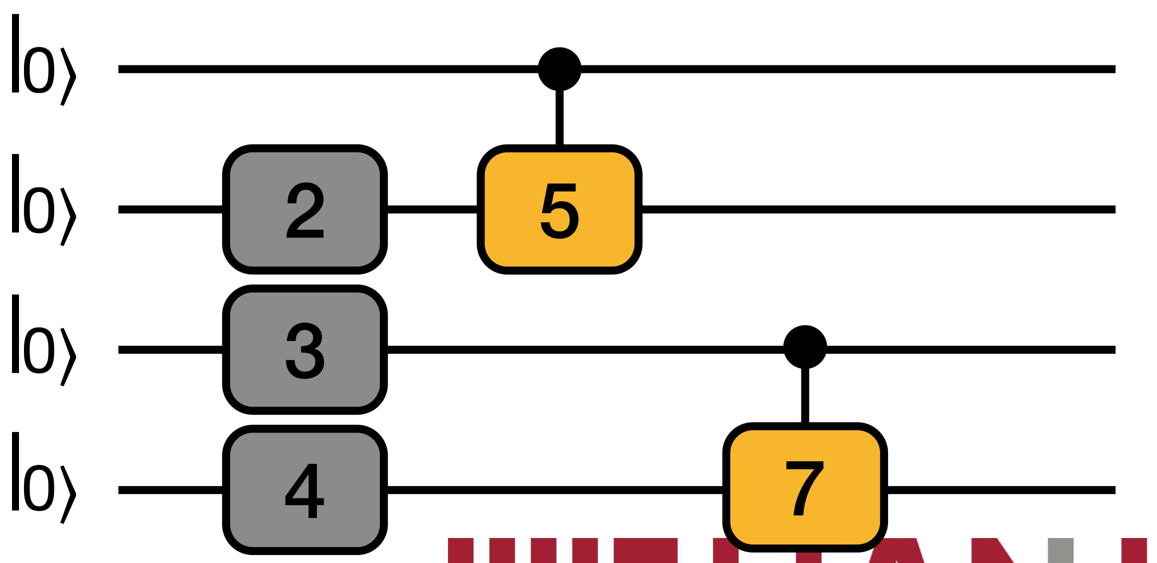
- SuperCircuit: the circuit with the **largest** number of gates in the design space
  - Example: SuperCircuit in U3+CU3 space



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







SuperCircuit Construction

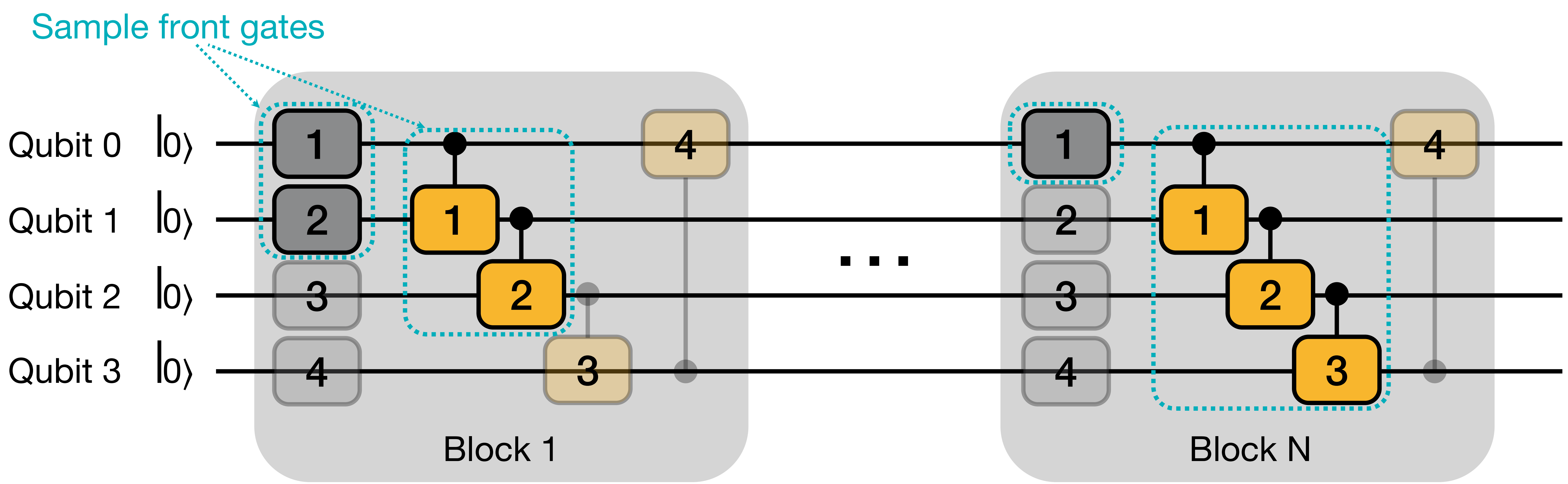
- Why use a SuperCircuit?
  - Enables **efficient** search of architecture candidates without training each
  - SubCircuit inherits parameters from SuperCircuit
  - With **inherited** parameters, we find some good SubCircuits, we find that they are also good SubCircuits with parameters **trained from-scratch**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

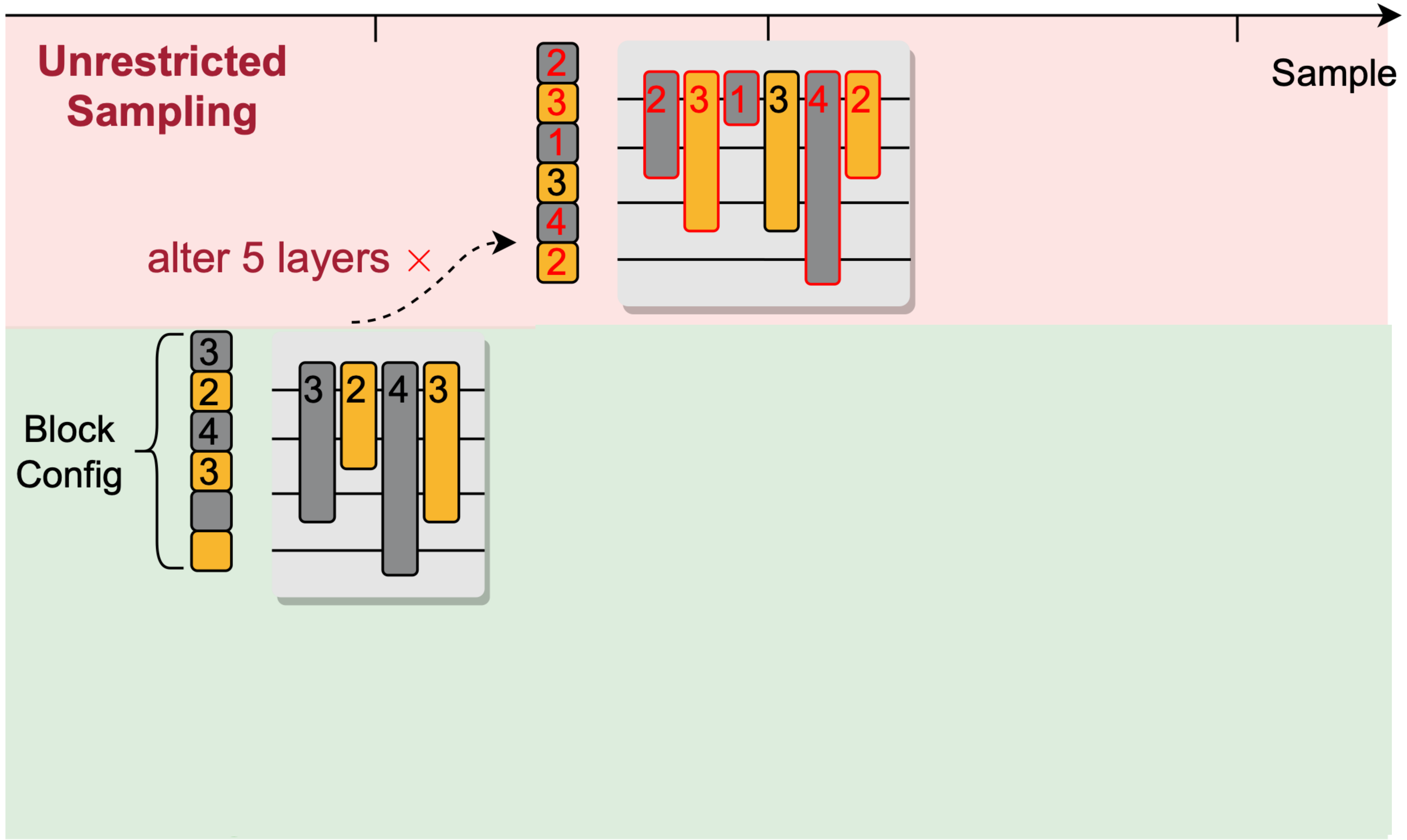
Front Sampling

- During sampling, we first sample total number of blocks, then sample gates within each block
  - Front sampling: Only the **front** several blocks and **front** several gates can be sampled to make SuperCircuit training more stable



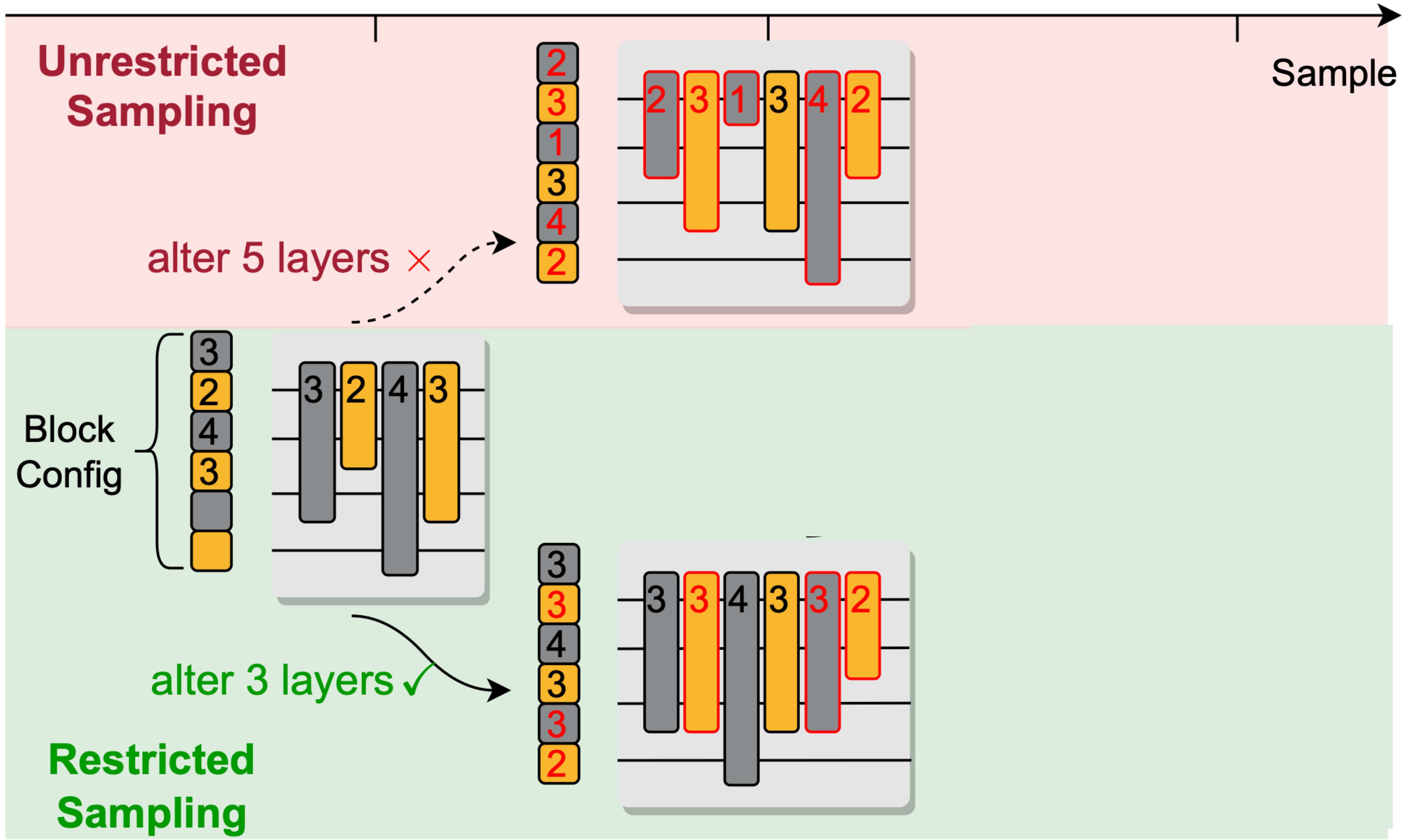
Restricted Sampling

- Restricted Sampling:
  - Restrict the difference between SubCircuits of two consecutive steps
  - For example: restrict to at most 4 different layers



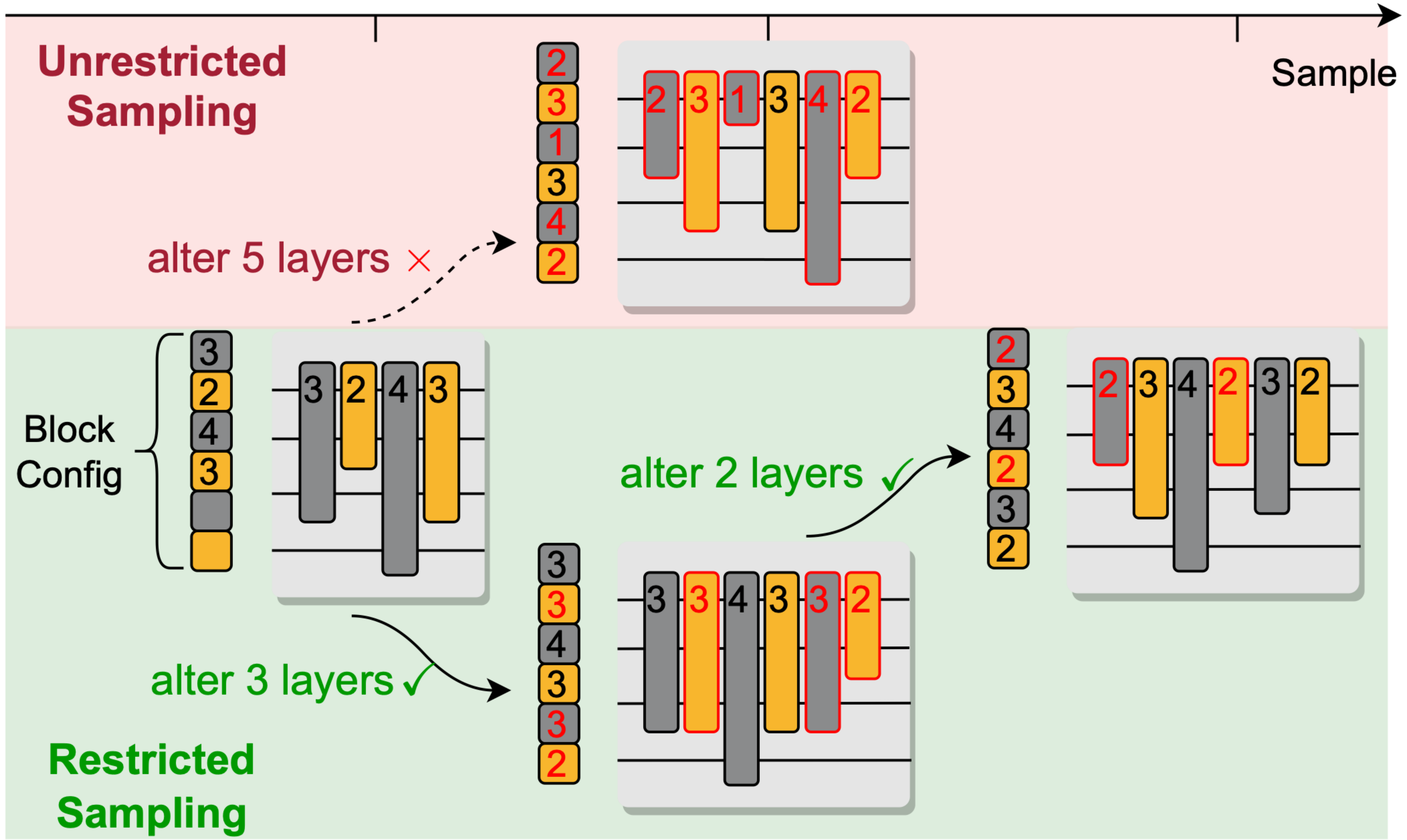
Restricted Sampling

- Restricted Sampling:
  - Restrict the difference between SubCircuits of two consecutive steps
  - For example: restrict to at most 4 different layers



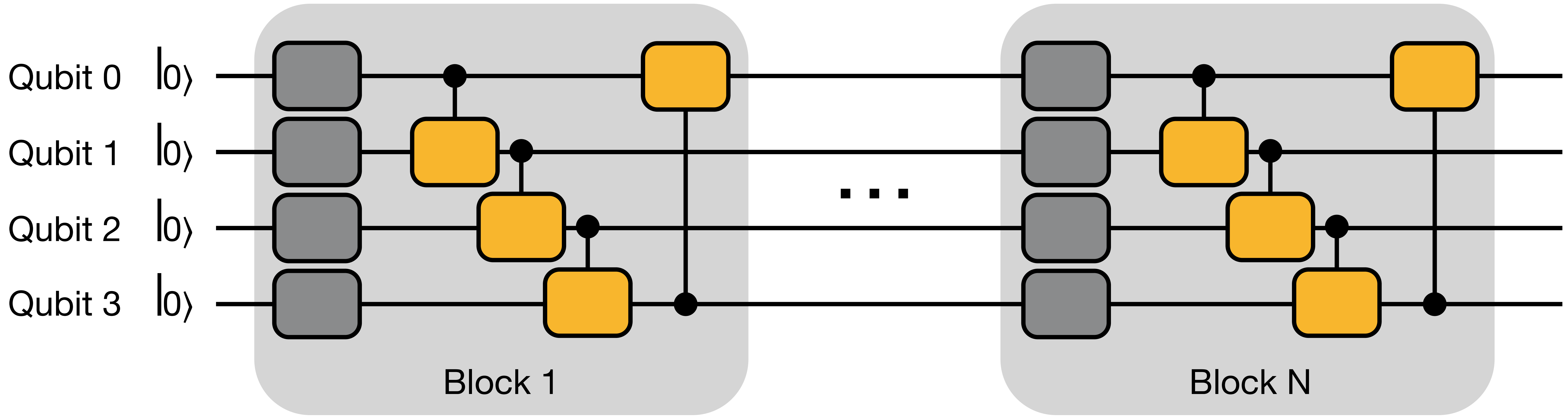
Restricted Sampling

- Restricted Sampling:
  - Restrict the difference between SubCircuits of two consecutive steps
  - For example: restrict to at most 4 different layers



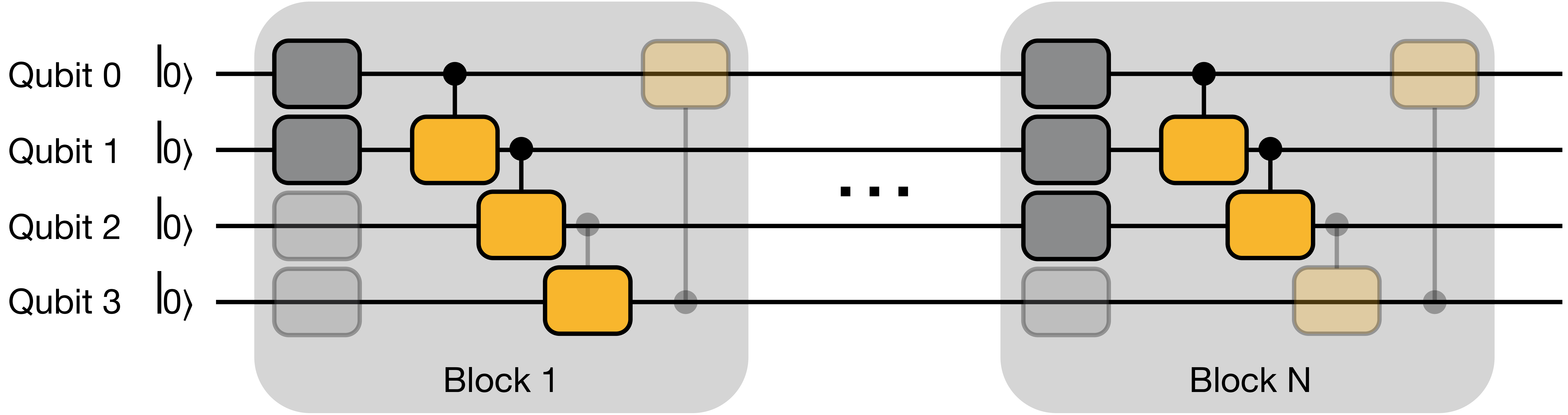
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



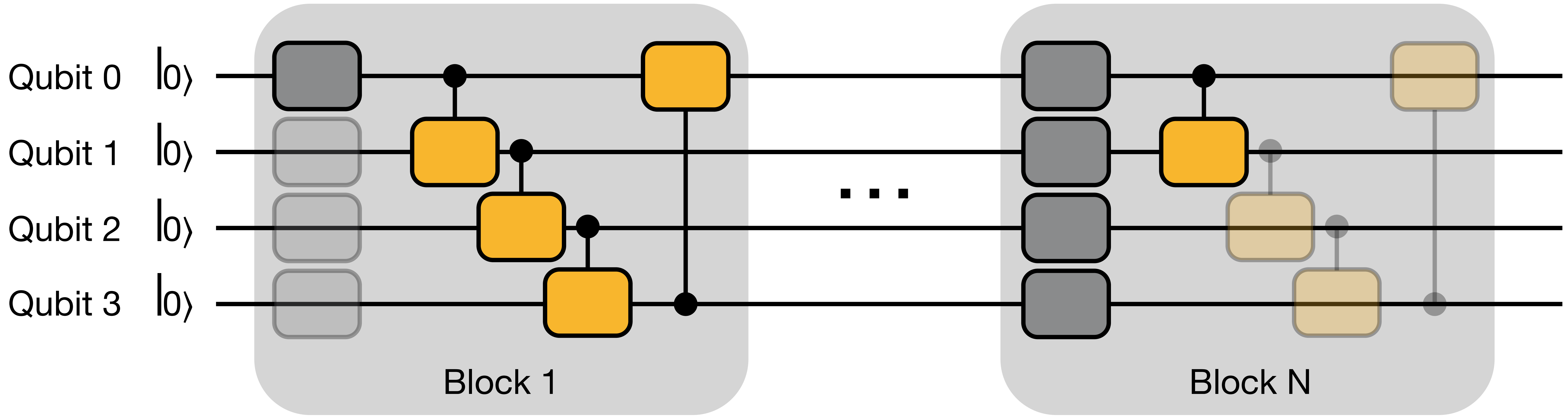
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



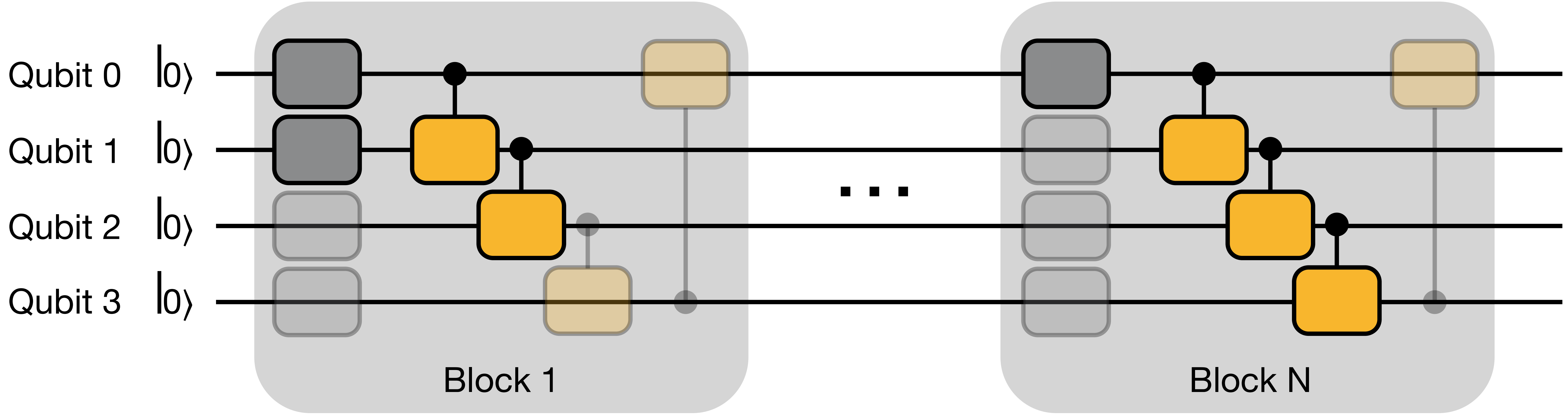
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



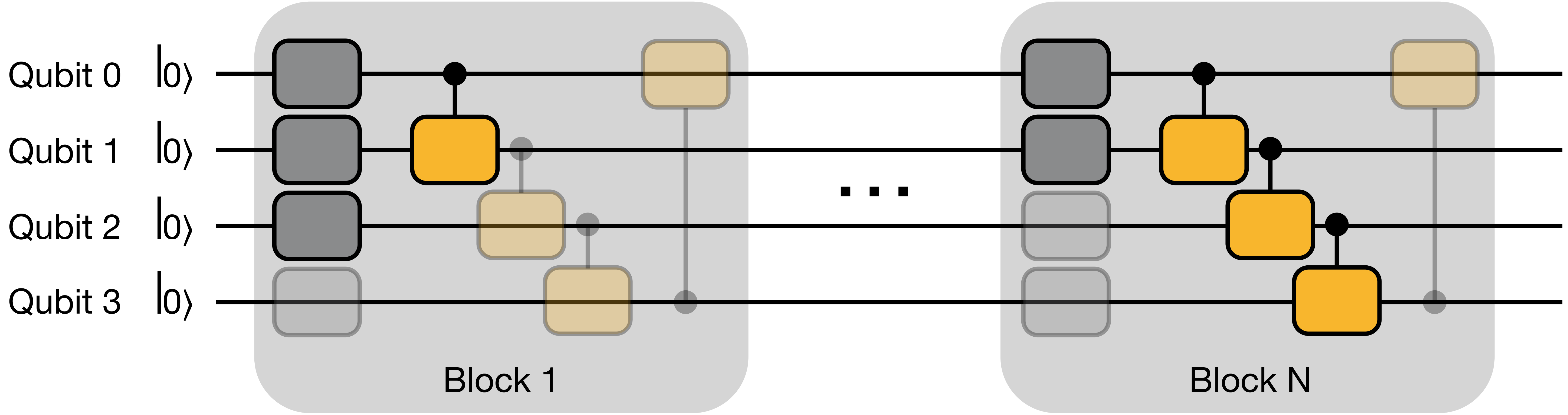
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



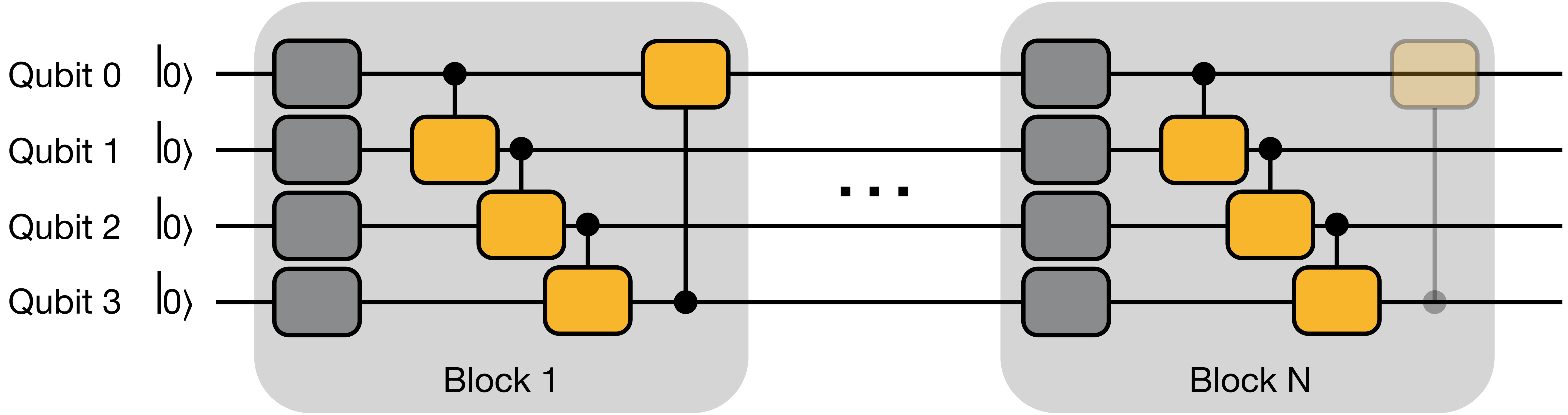
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



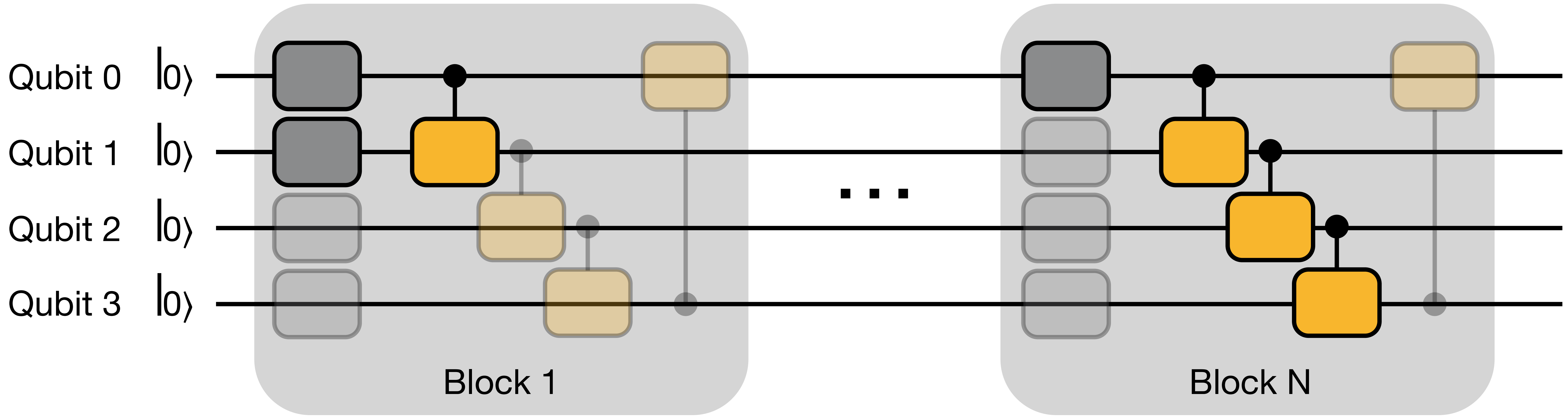
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



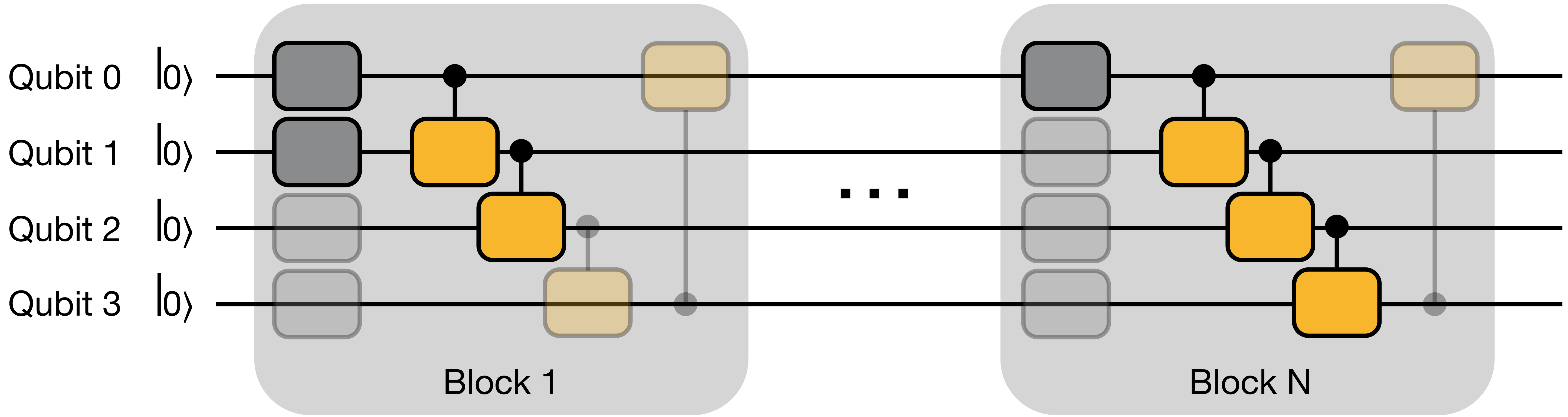
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



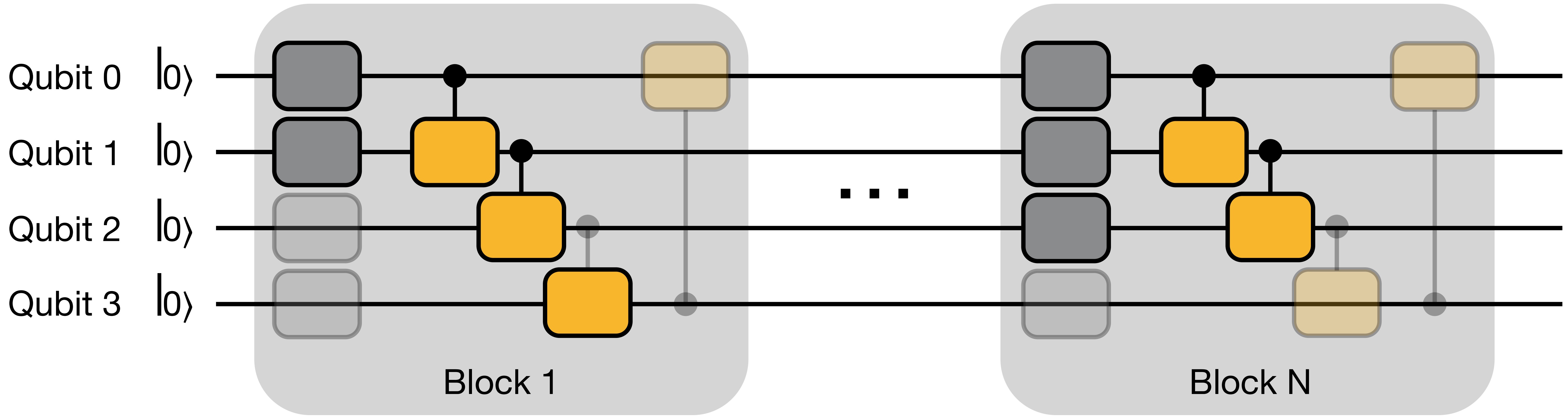
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



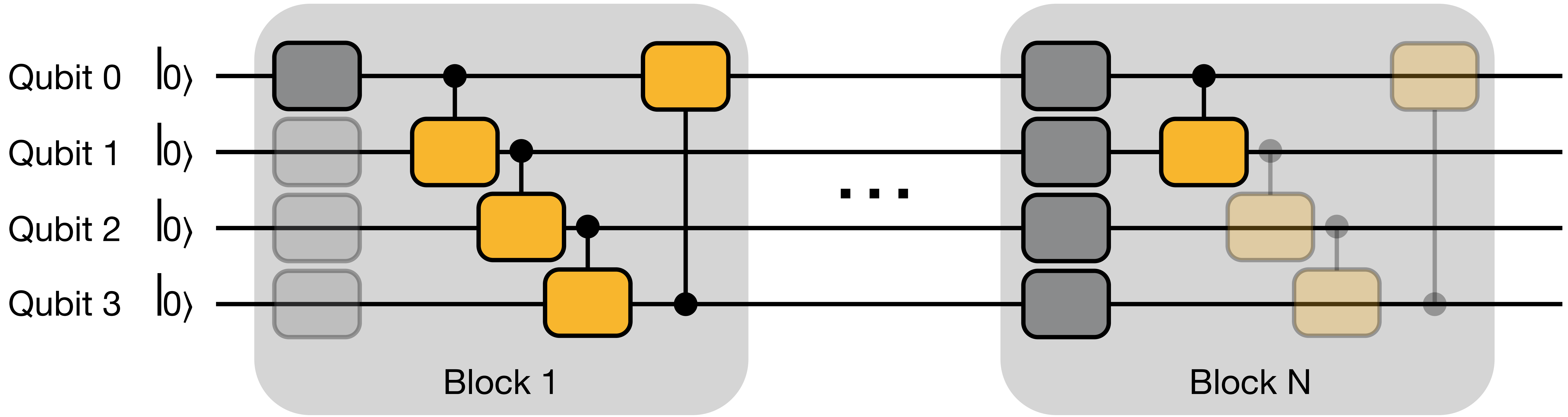
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



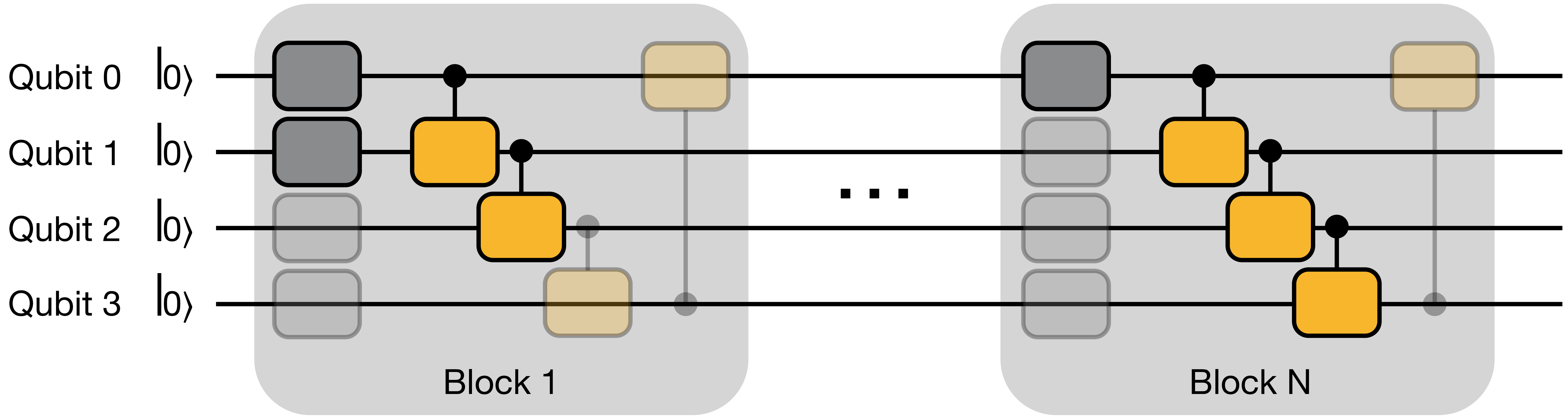
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



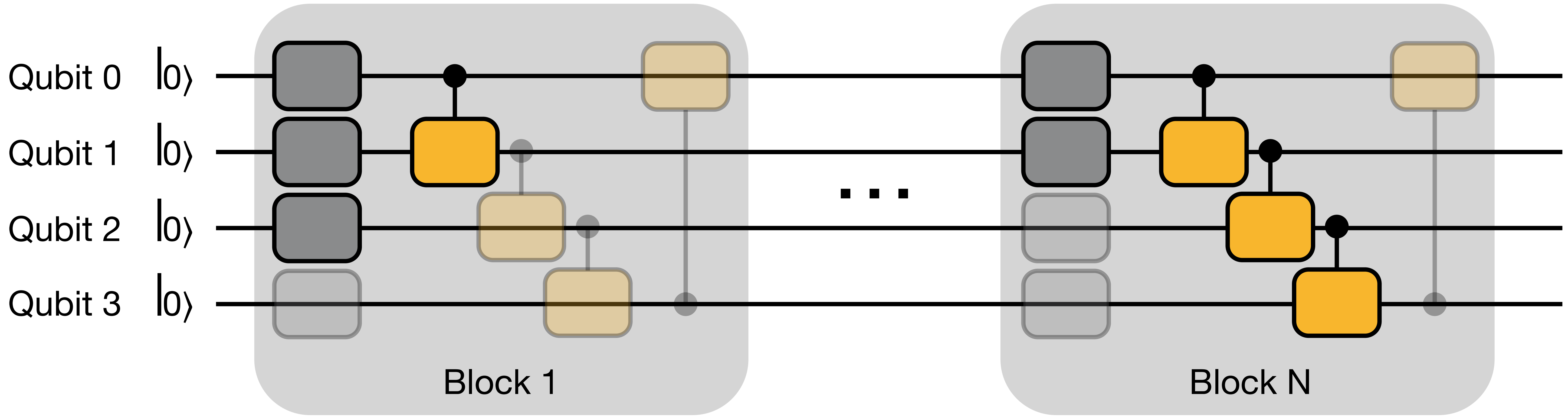
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



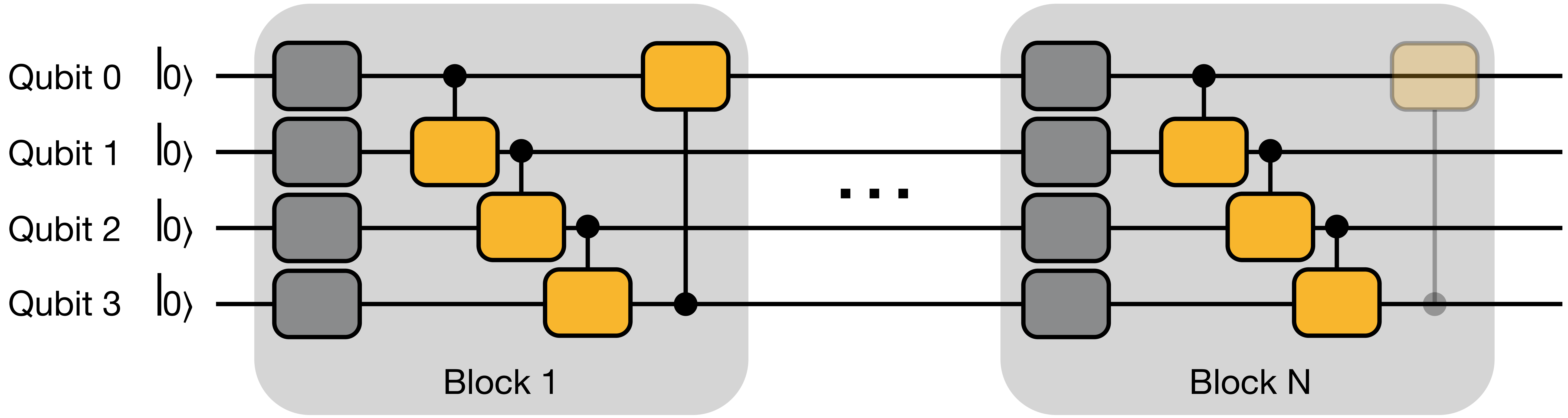
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



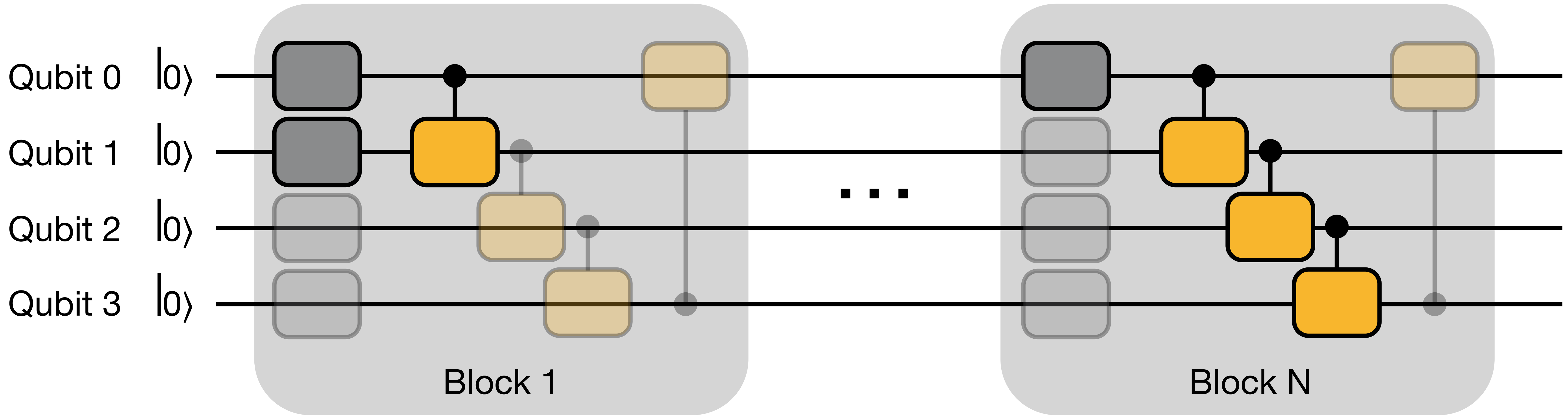
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



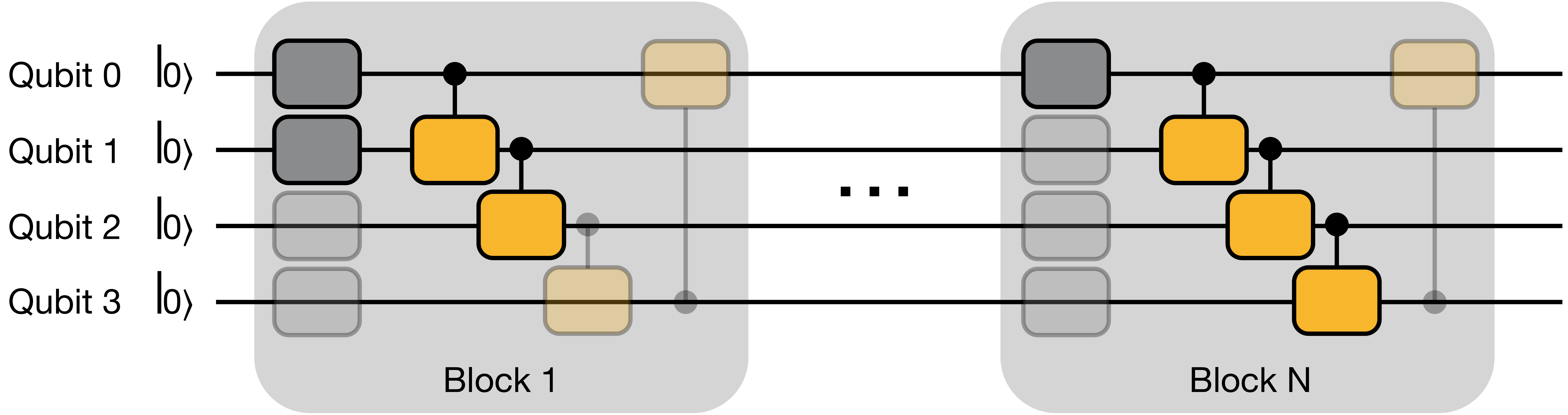
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train

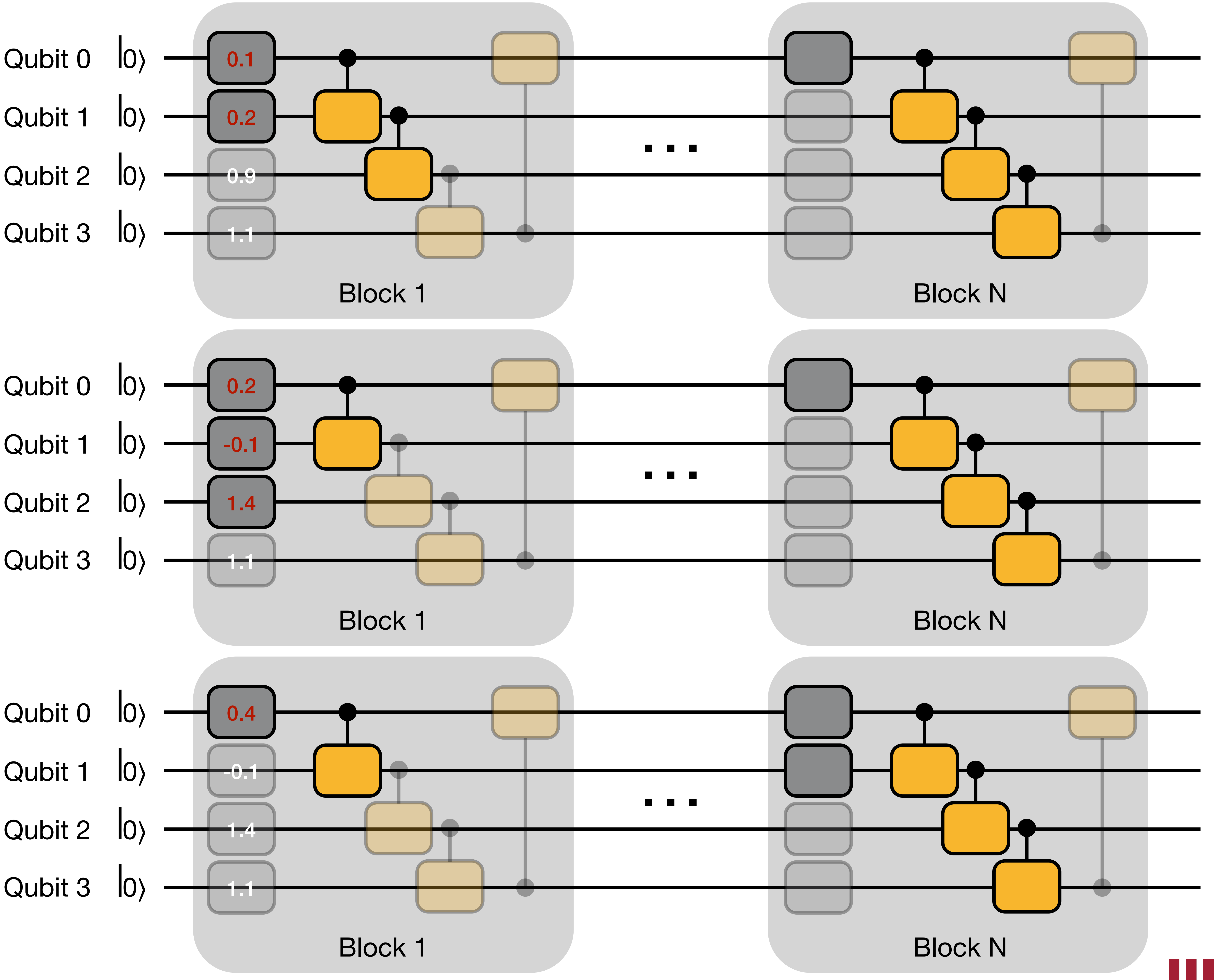


Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train

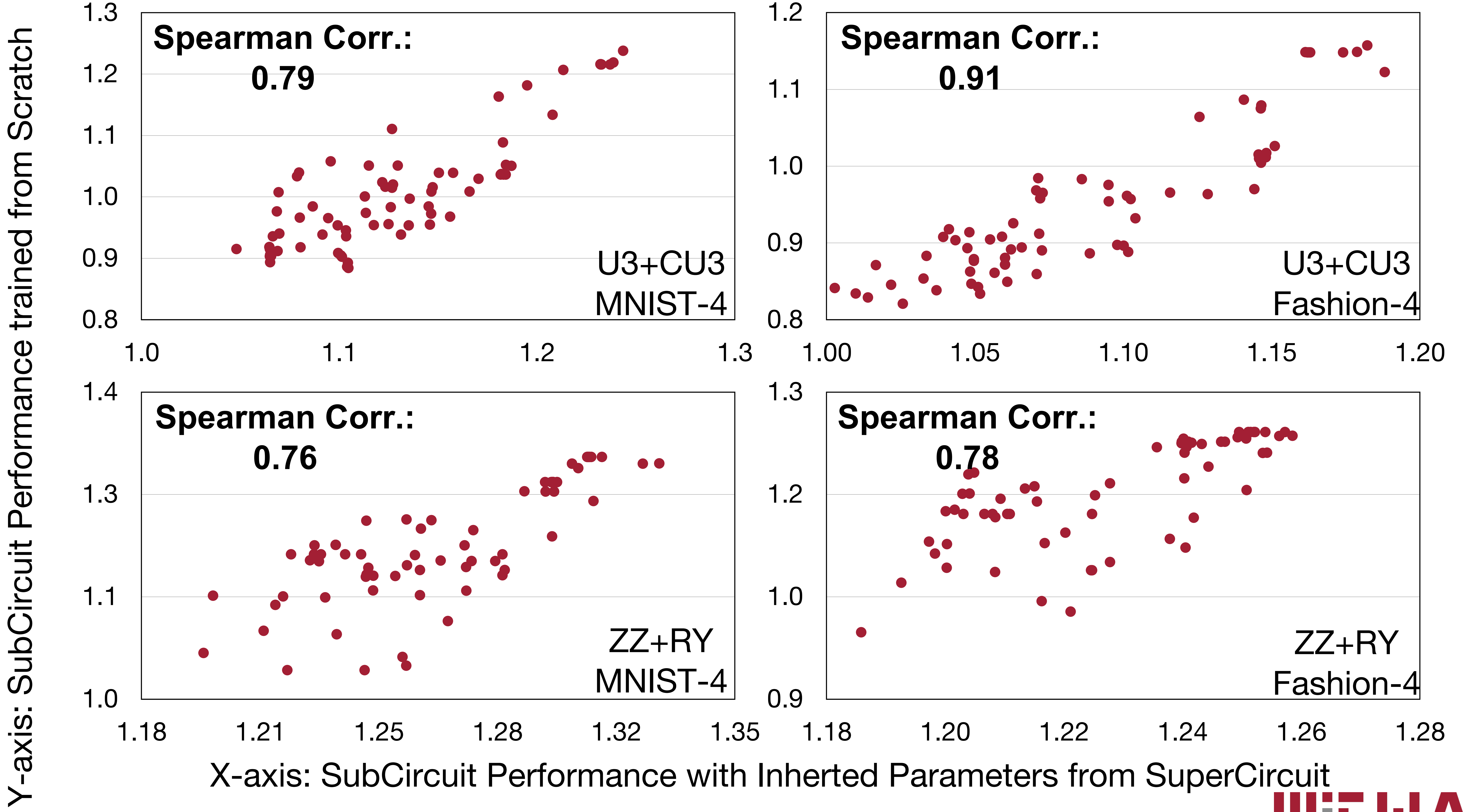


Train SuperCircuit for Multiple Steps



How Reliable is the SuperCircuit?

- Inherited parameters from SuperCircuit can provide accurate relative performance



Torch Quantum

qmlsys.mit.edu

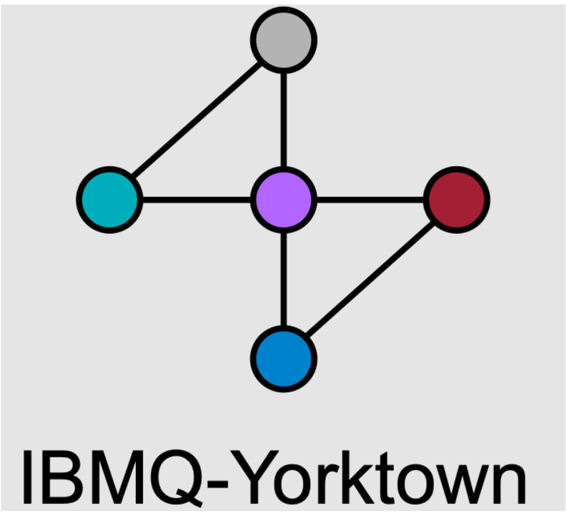
MIT HAN LAB

QuantumNAS

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

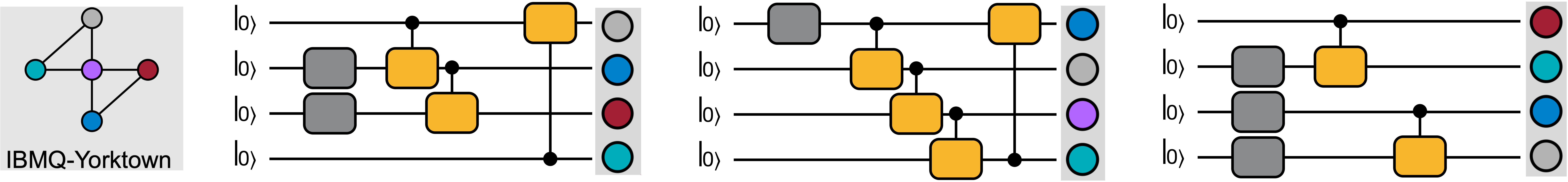
Noise-Adaptive Evolutionary Co-Search

- 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

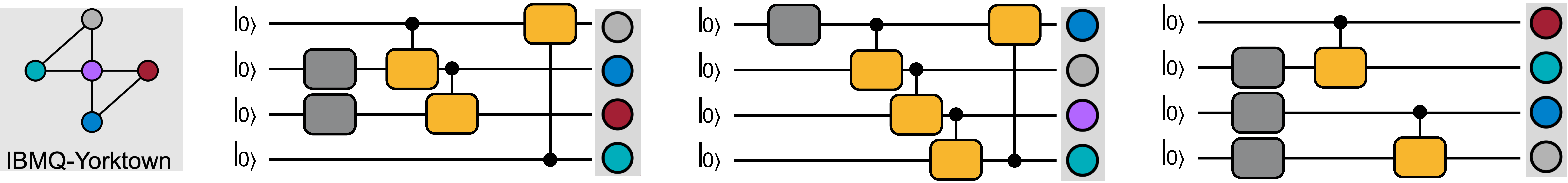
SubCircuit and Qubit Mapping pairs →

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



Evolutionary Search Engine
Mutation
Crossover

SubCircuit and Qubit Mapping pairs →

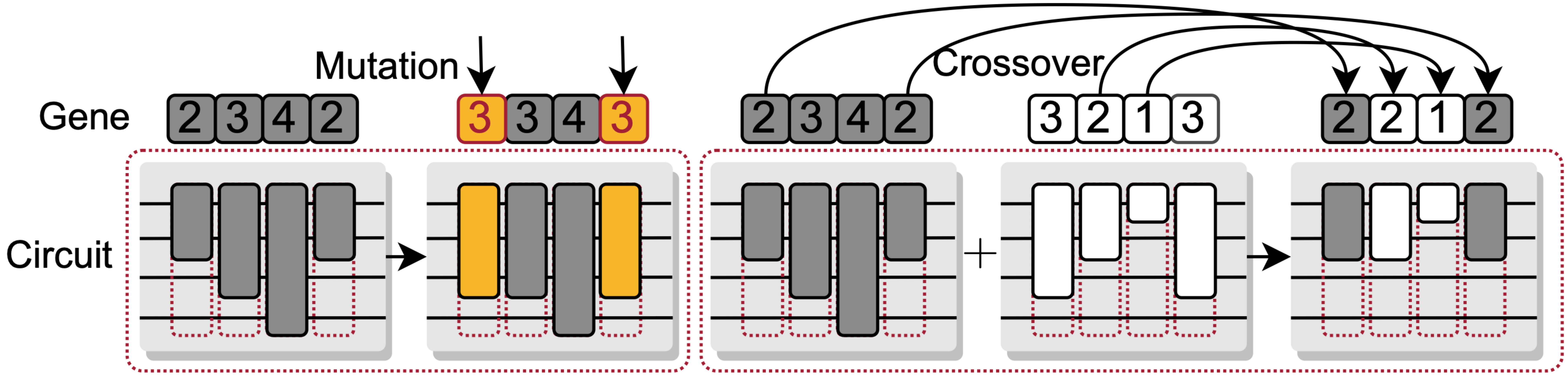
← Performance of SubCircuit and Qubit Mapping pairs

Parameters from SuperCircuit

Simulate with noise model
Or
Directly run on real QC device

Mutation and Crossover

- Mutation and crossover create new SubCircuit candidates



QuantumNAS

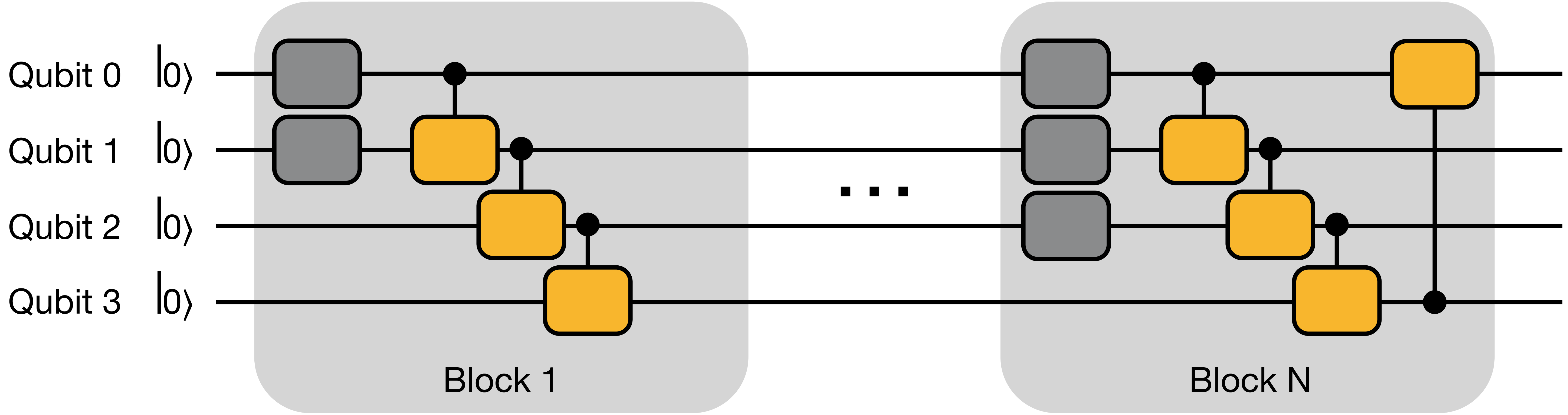
- SuperCircuit Construction and Training
- Noise-Adaptive Evolutionary Co-Search of SubCircuit and Qubit Mapping
- Train the Searched SubCircuit
- Iterative Quantum Gate Pruning

QuantumNAS

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

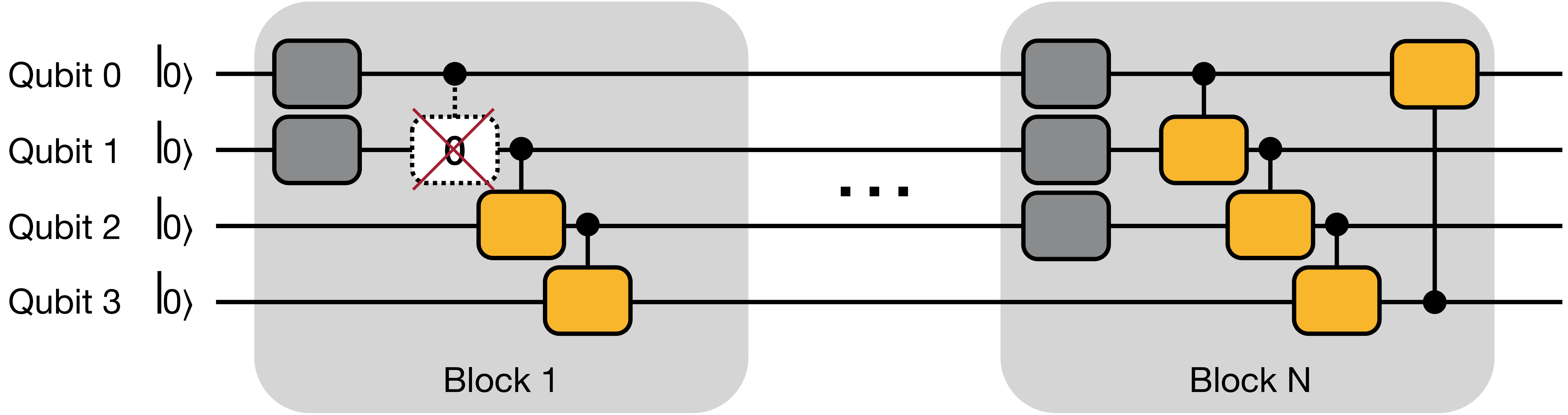
Iterative Pruning

- 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



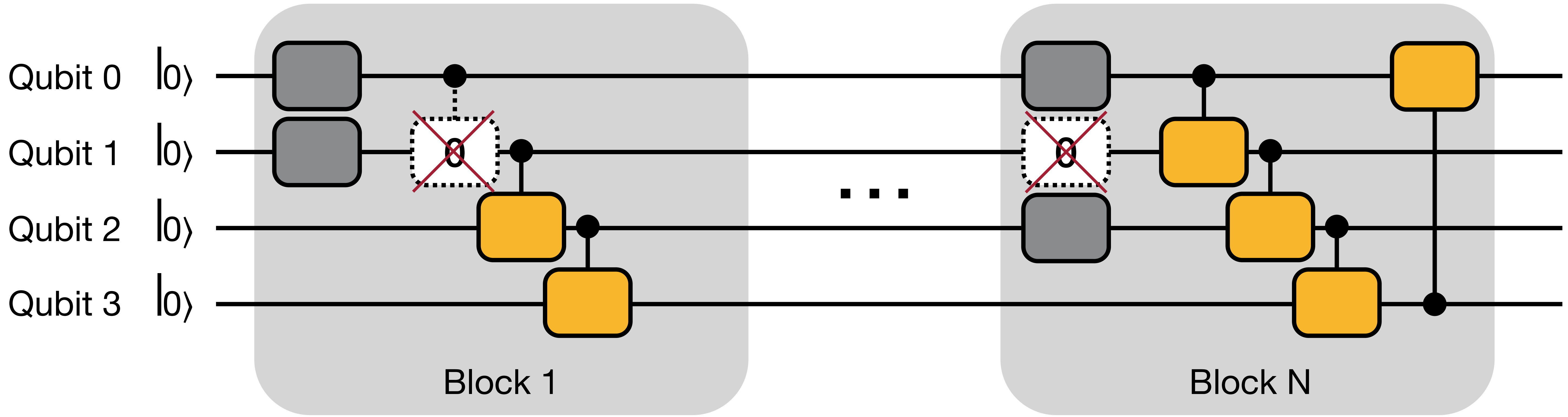
Iterative Pruning

- 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



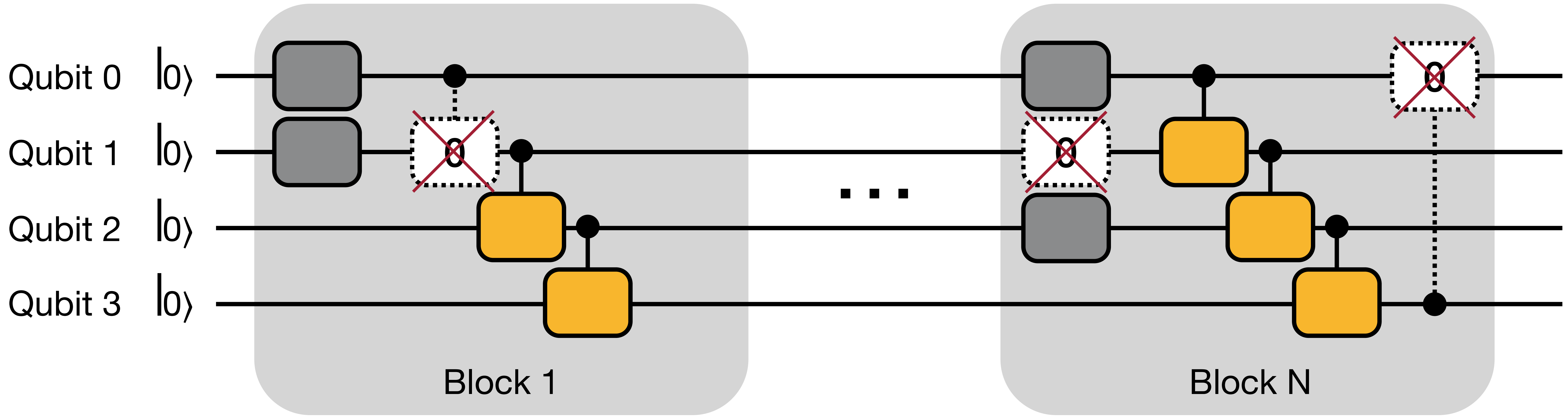
Iterative Pruning

- 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



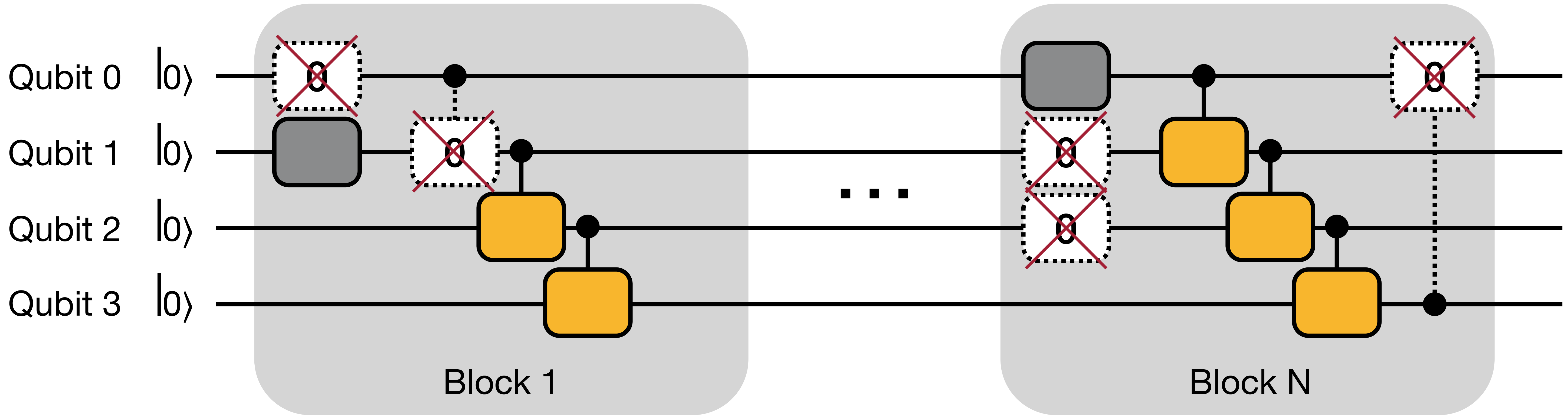
Iterative Pruning

- 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

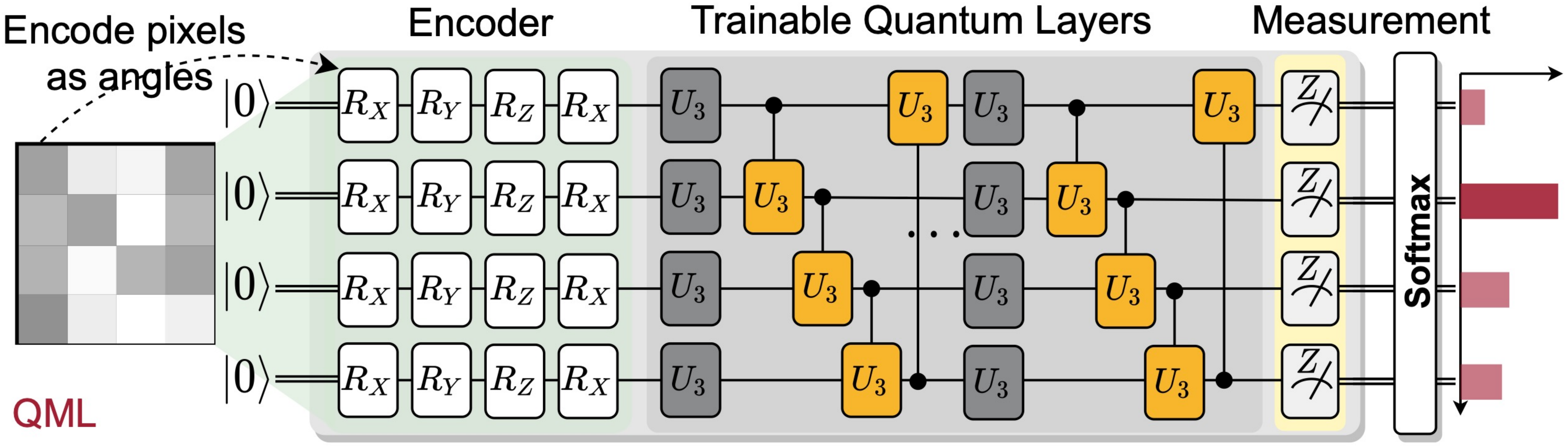


Evaluation Setups: Benchmarks and Devices

- 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

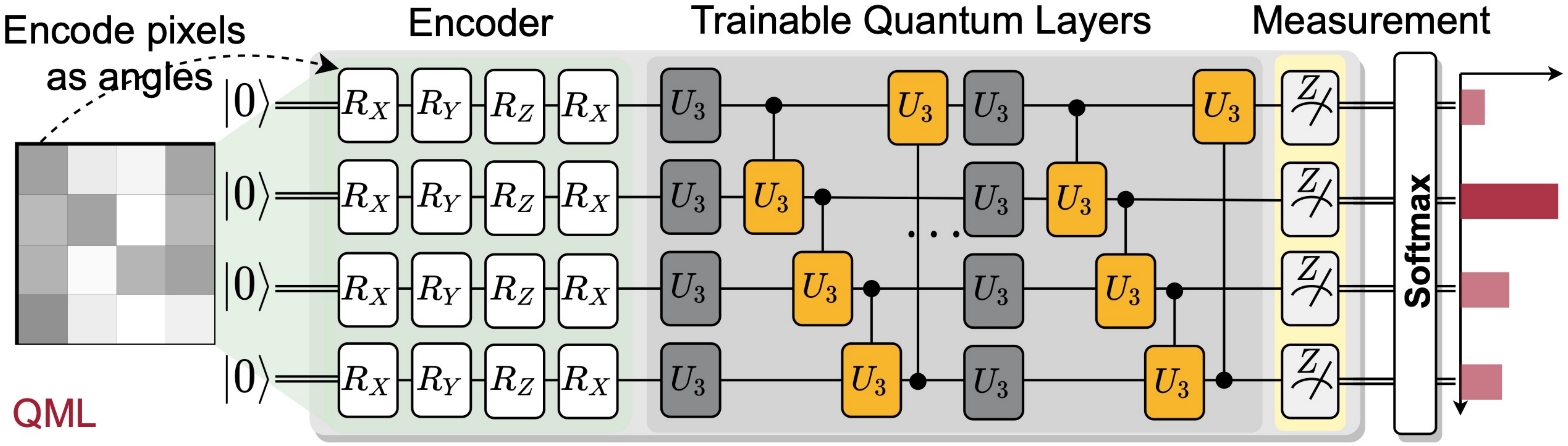
Benchmarks: QNN and VQE

- Quantum Neural Networks: classification

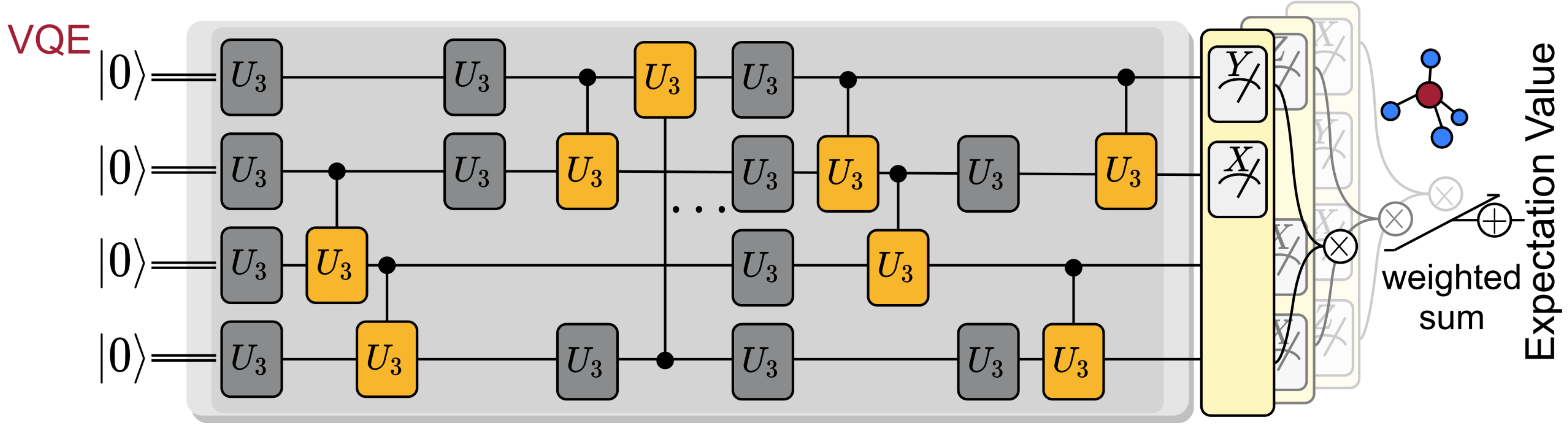


Benchmarks: QNN and VQE

- Quantum Neural Networks: classification

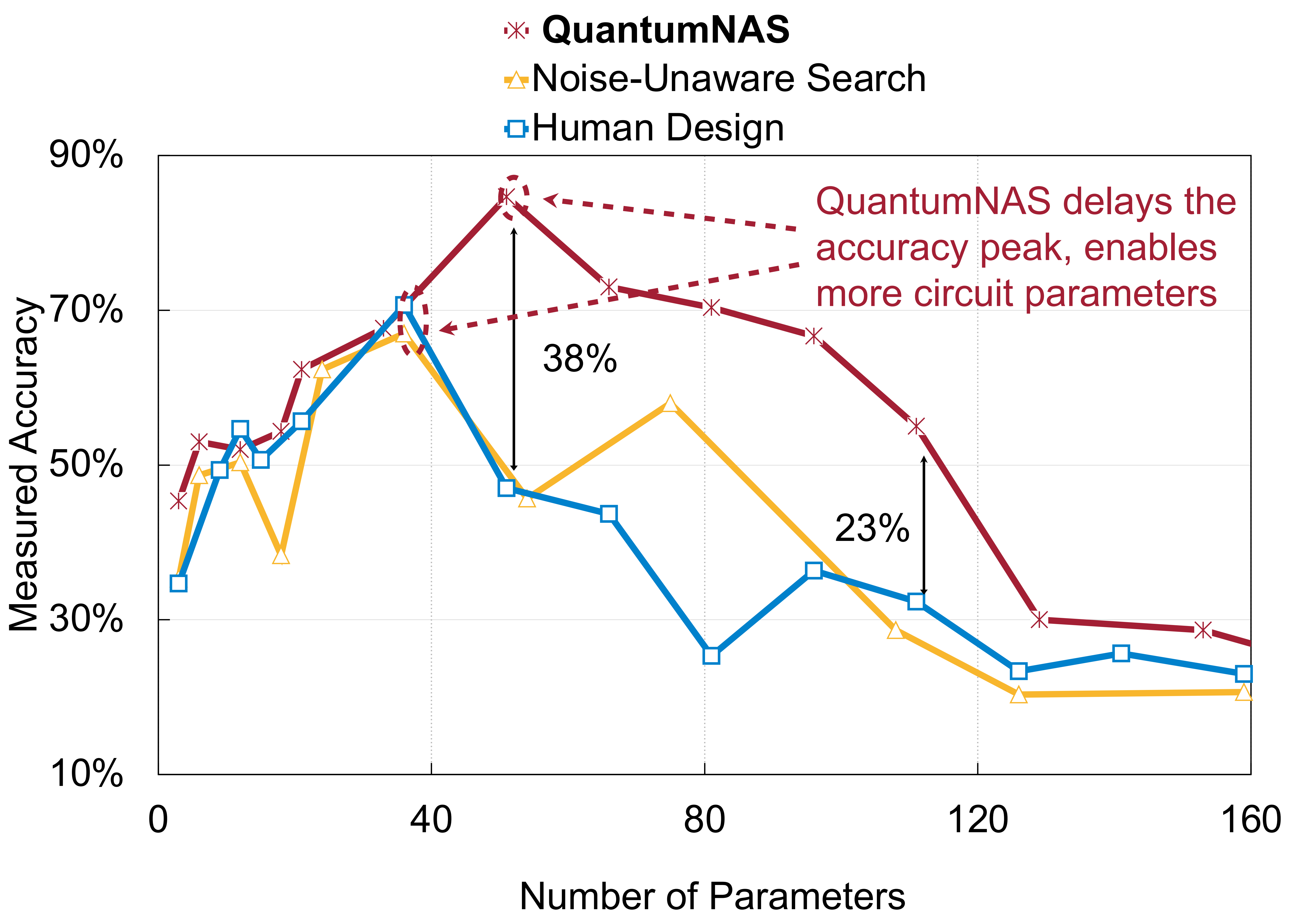


- Variational Quantum Eigensolver: finds the ground state energy of molecule Hamiltonian



QML Results

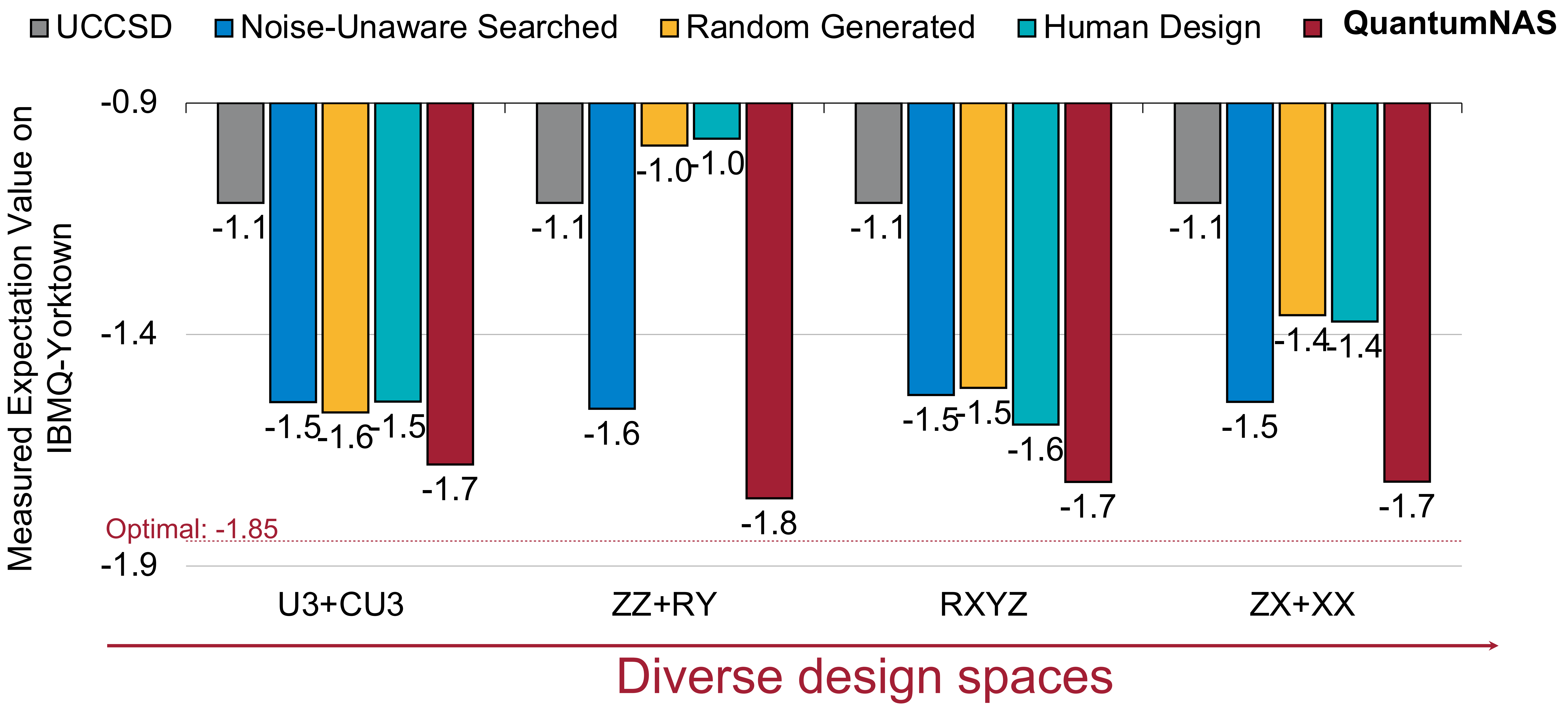
- 4-classification: MNIST-4 U3+CU3 on IBMQ-Yorktown



qmlsys.mit.edu

Consistent Improvements on Diverse Design Spaces

- H2 in different design spaces on IBMQ-Yorktown



Torch Quantum

qmlsys.mit.edu

MIT HAN LAB

Scalable to Large #Qubits

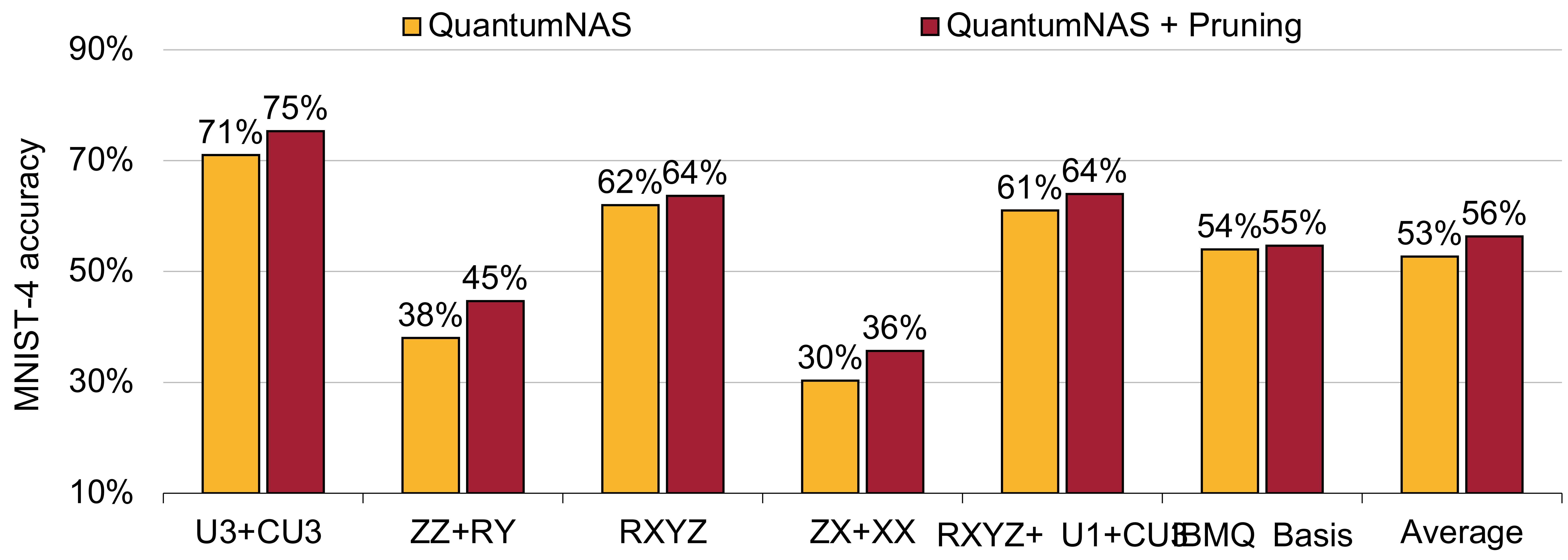
- On large devices
- MNIST-10 accuracy

More Qubits

Method	Noise-Unaware Searched	Random	Human	**QuantumNAS**
Melbourne (15Q, 8QV, use 15Q)	11%	10%	15%	**32%**
Guadalupe (16Q, 32QV, use 16Q)	14%	12%	10%	**15%**
Montreal (27Q, 128QV, use 21Q)	13%	7%	14%	**16%**
Manhattan (65Q, 32QV, use 21Q)	11%	11%	15%	**18%**

Effective of Quantum Gate Pruning

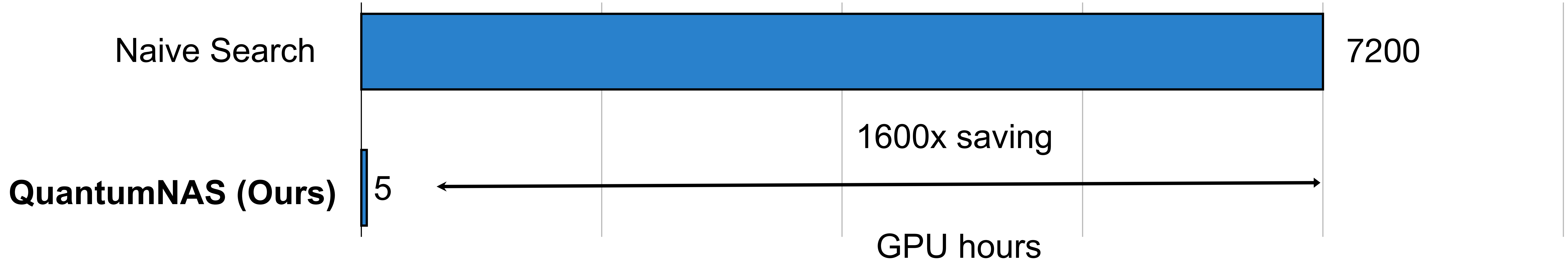
- For MNIST-4, Quantum gate pruning improves accuracy by 3% on average



Time Cost

- 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

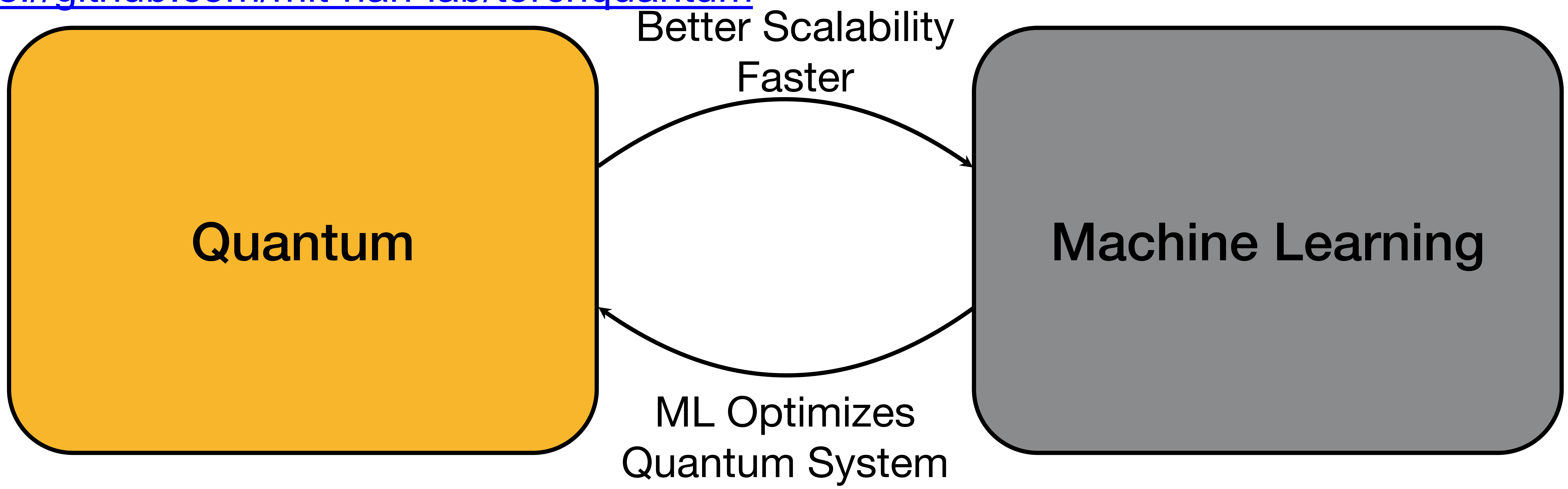


Torch
Quantum

Open-source: TorchQuantum

- TorchQuantum — An open-source library for interdisciplinary research of quantum computing and machine learning

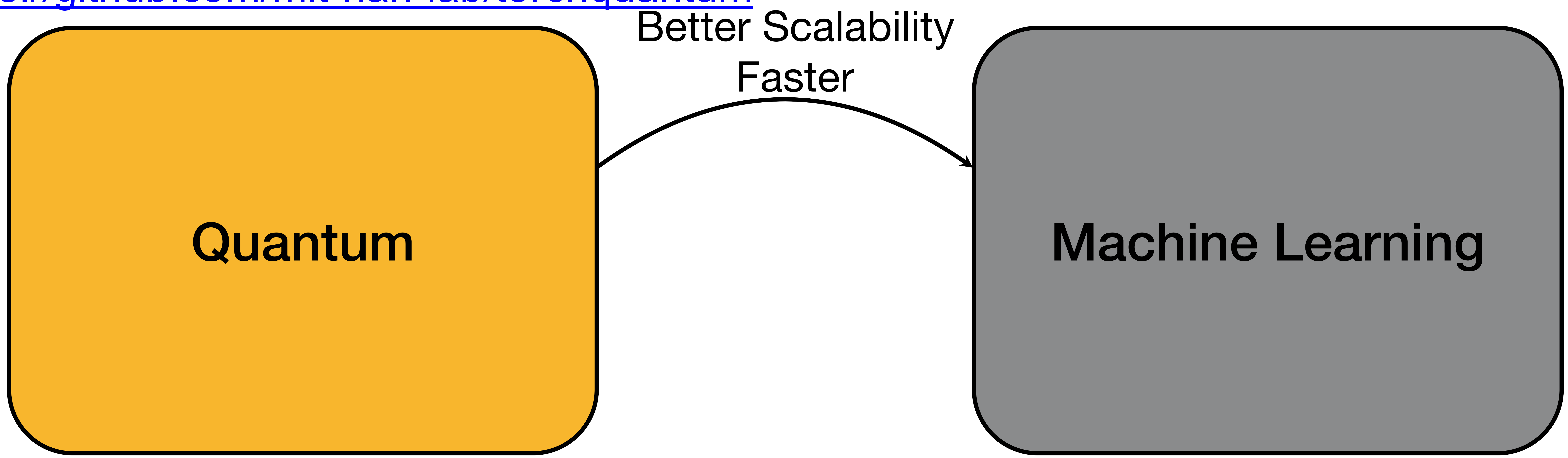
  https://github.com/mit-han-lab/torchquantum



Open-source: TorchQuantum

- TorchQuantum — An open-source library for interdisciplinary research of quantum computing and machine learning

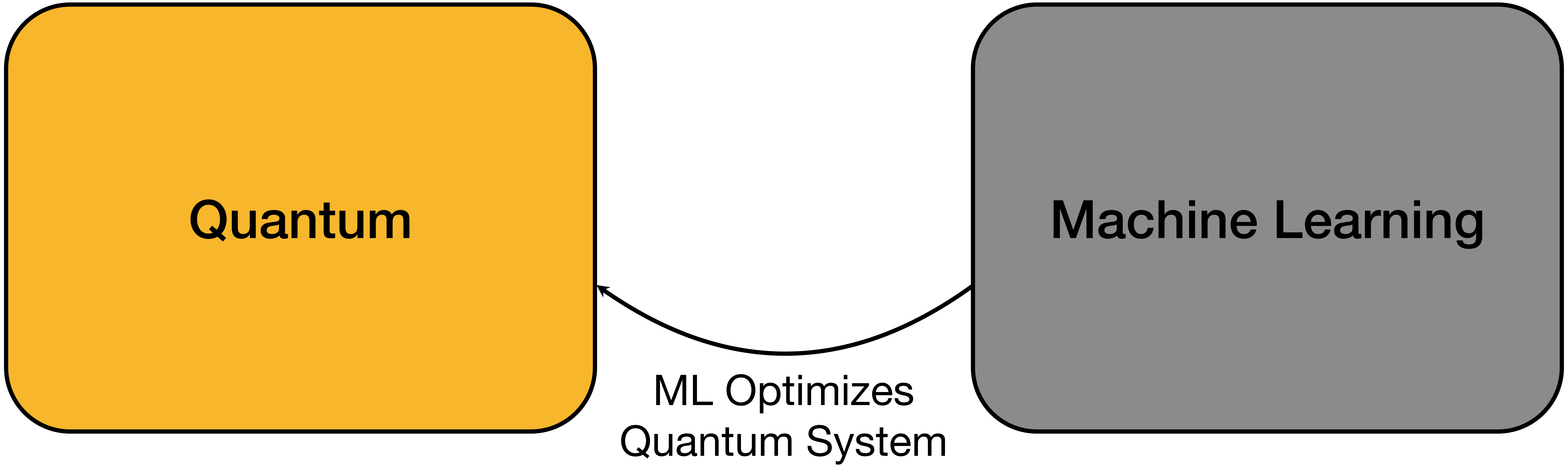
  https://github.com/mit-han-lab/torchquantum



- Quantum for Machine learning
  - Quantum neural networks
  - Quantum kernel methods

Open-source: TorchQuantum

- TorchQuantum — An open-source library for interdisciplinary research of quantum computing and machine learning
- https://github.com/mit-han-lab/torchquantum



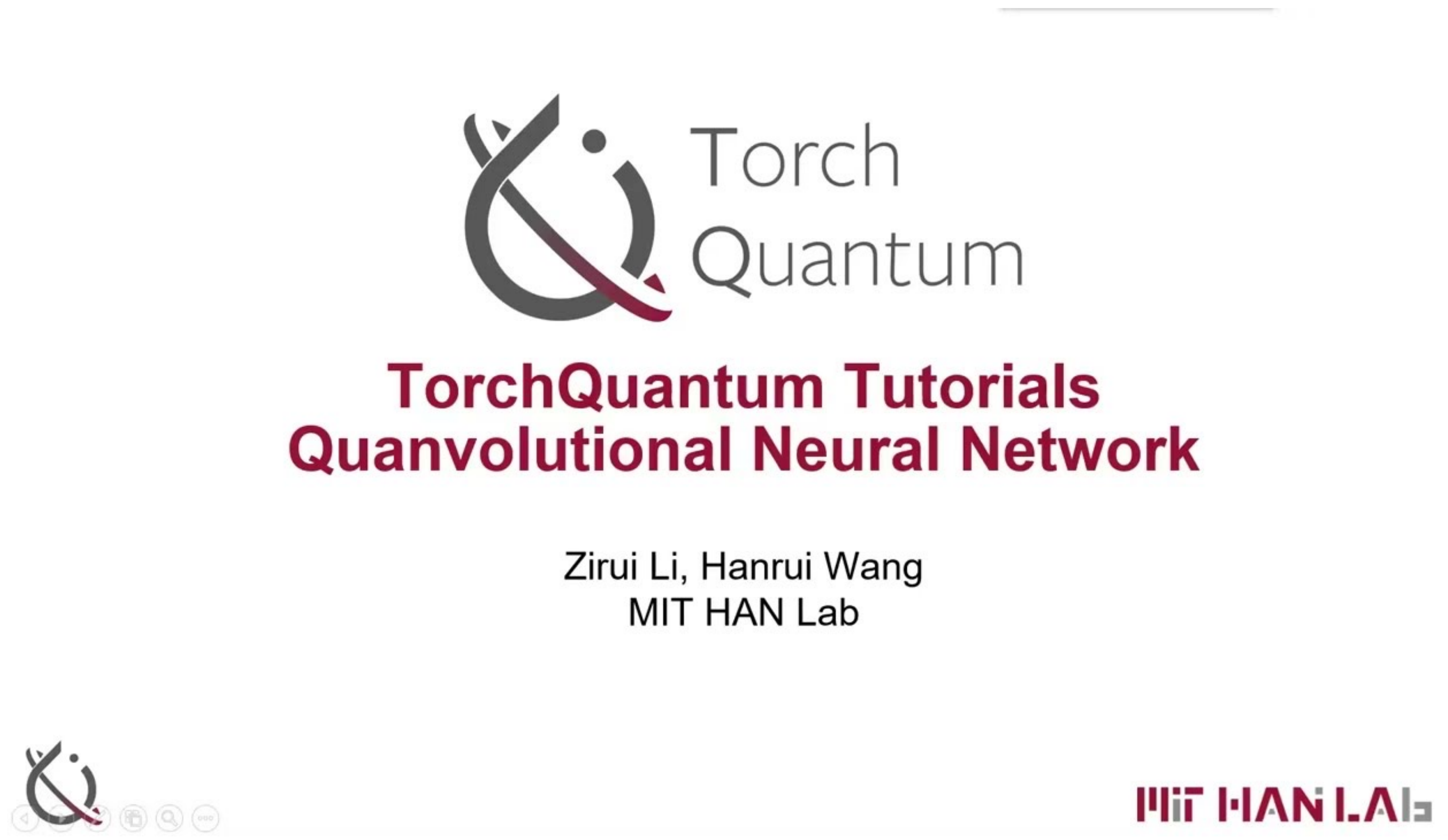
- Machine Learning for Quantum
  - ML for quantum compilation (qubit mapping, unitary synthesis)
  - …

TorchQuantum

- Features
  - Easy construction of **parameterized quantum circuits** such as Quantum Neural Networks in PyTorch
  - Support **batch mode inference and training** on GPU/CPU, supports highly-parallelized training
  - Support **easy deployment** on real quantum devices such as IBMQ
  - Provide tutorials, videos and example projects of QML and using ML to optimize quantum computer system problems
- Will be on QCE (Quantum Computing Engineering) tutorials

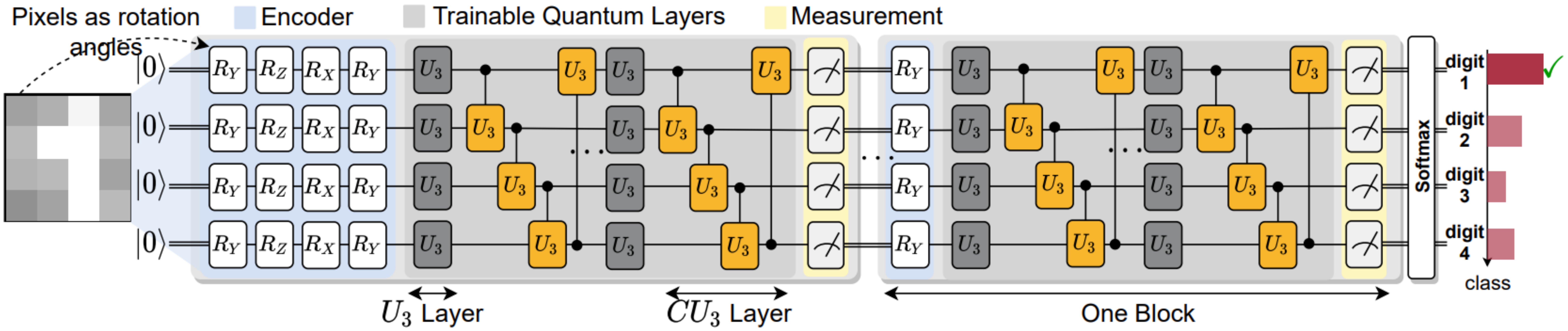
Examples and tutorials

- Tutorial Colab and videos



QuantumNAT

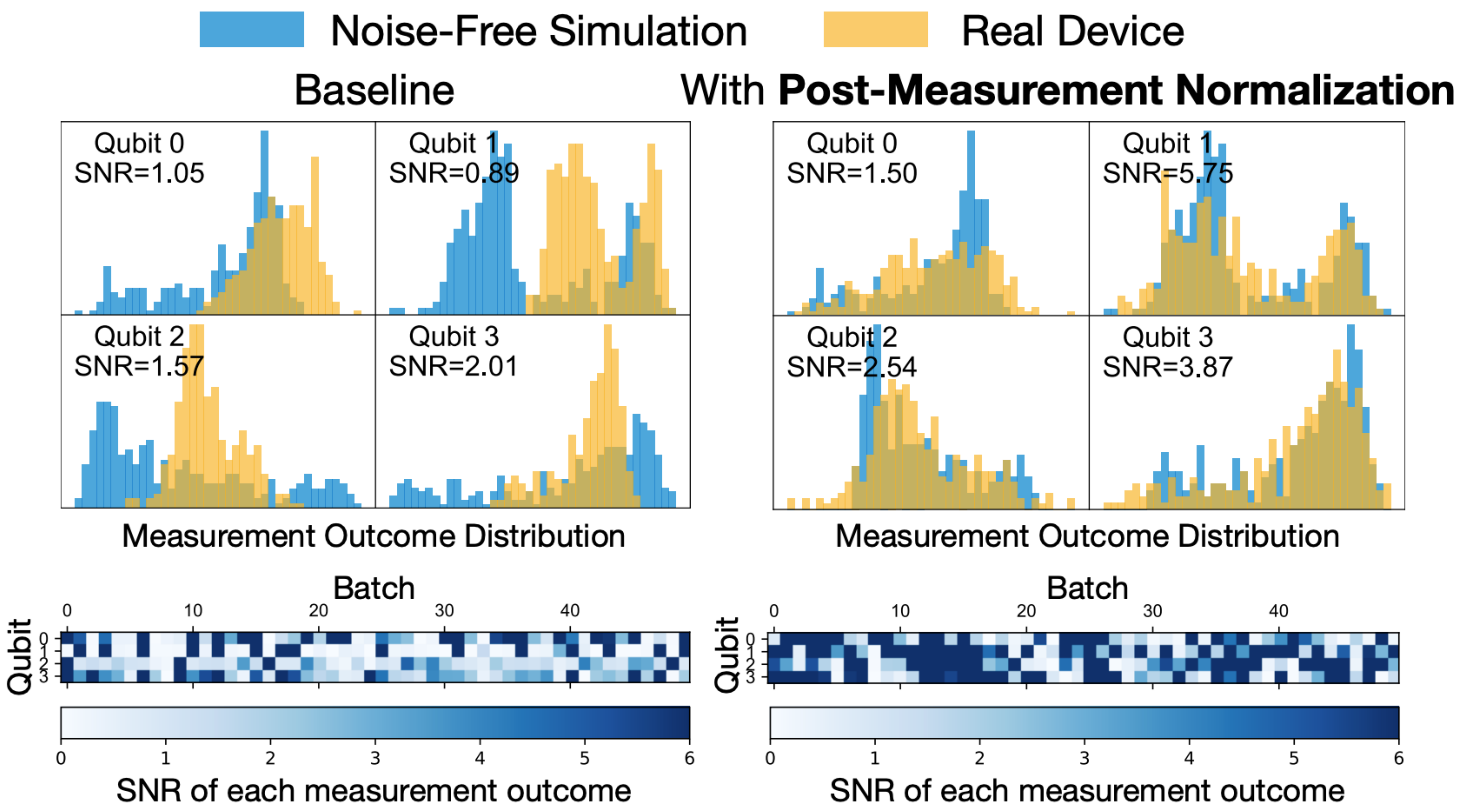
- **QuantumNAS: find the circuit architecture robust to noise**
- **QuantumNAT: further make the parameter robust to noise**



[DAC'22] Wang et, at, QuantumNAT: Quantum Noise-Aware Training with Noise Injection, Quantization and Normalization

Post-Measurement Normalization

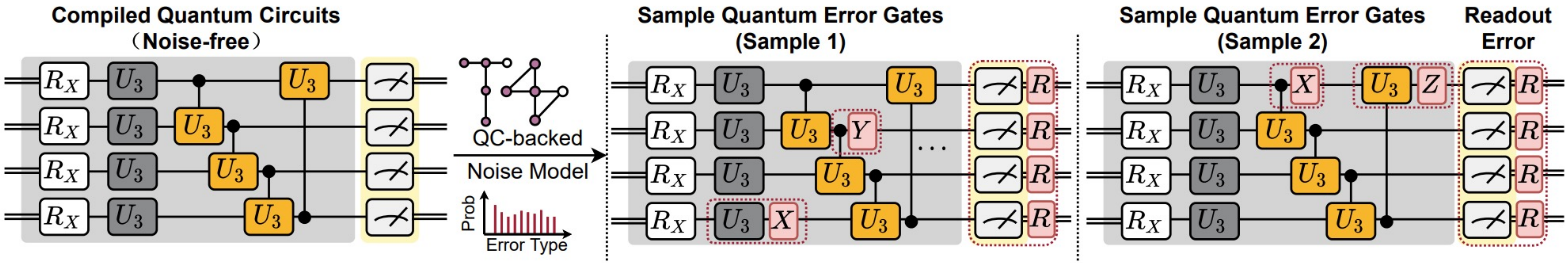
- **Normalize the measurement outcomes on the batch dimension**



2nd block output of a 3-block model, IBMQ-Quito

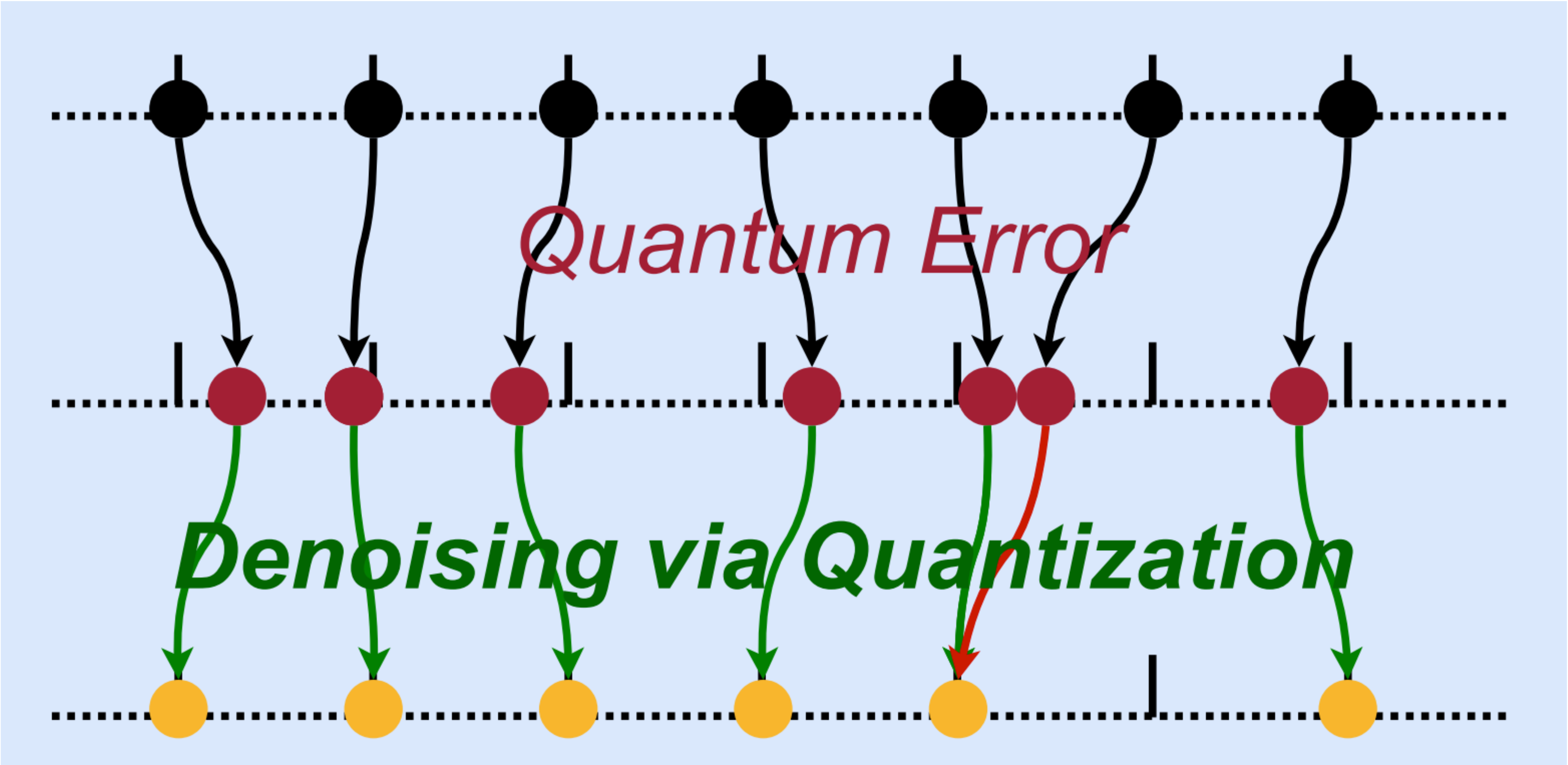
Noise Injection

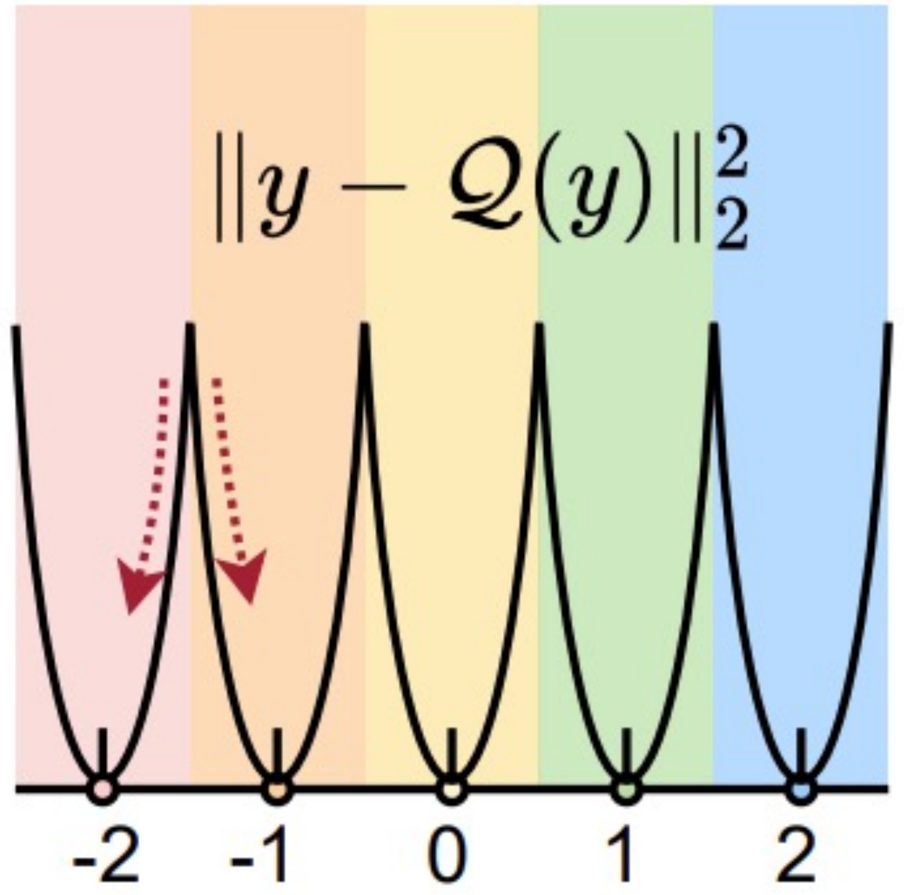
- **Insert noise gate during training, according to the noise model**
- **For each step, sample new positions for noise gates**



Post-Measurement Quantization

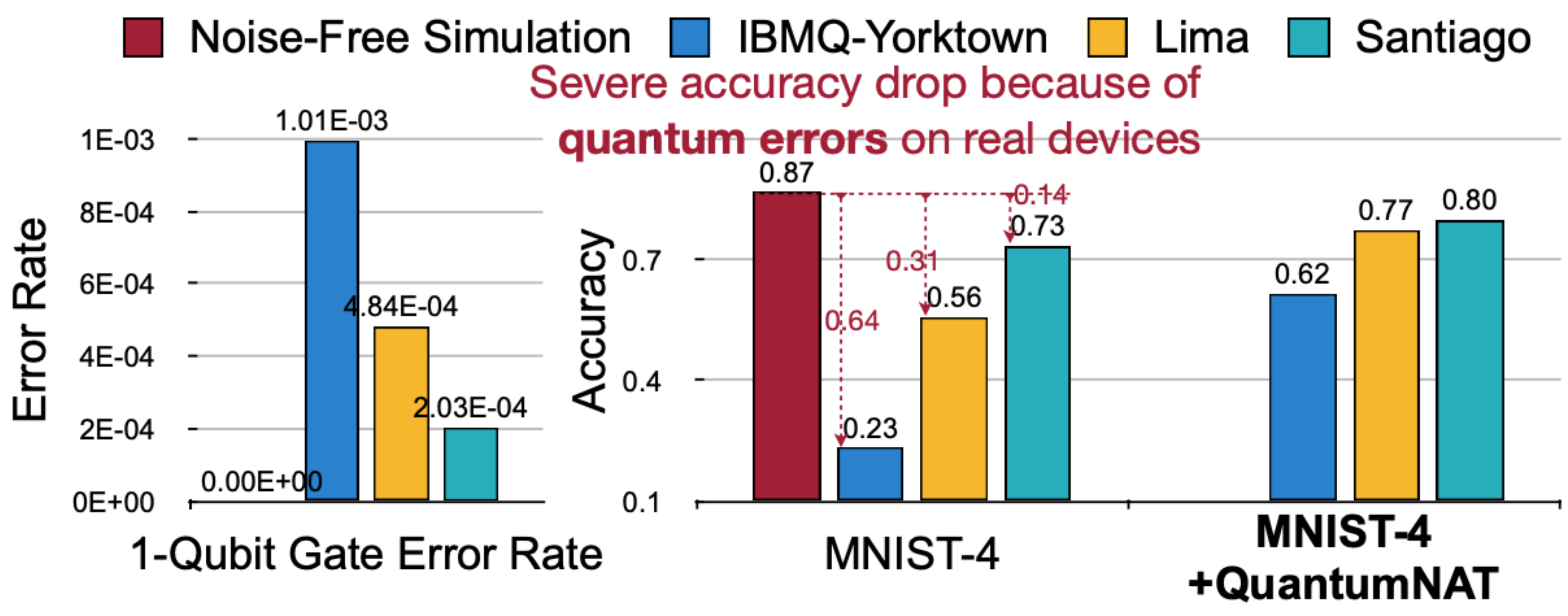
- **Quantization provides denoising effects**
- **Quadratic penalty loss to encourage measurement outcomes close to quantization centroids**

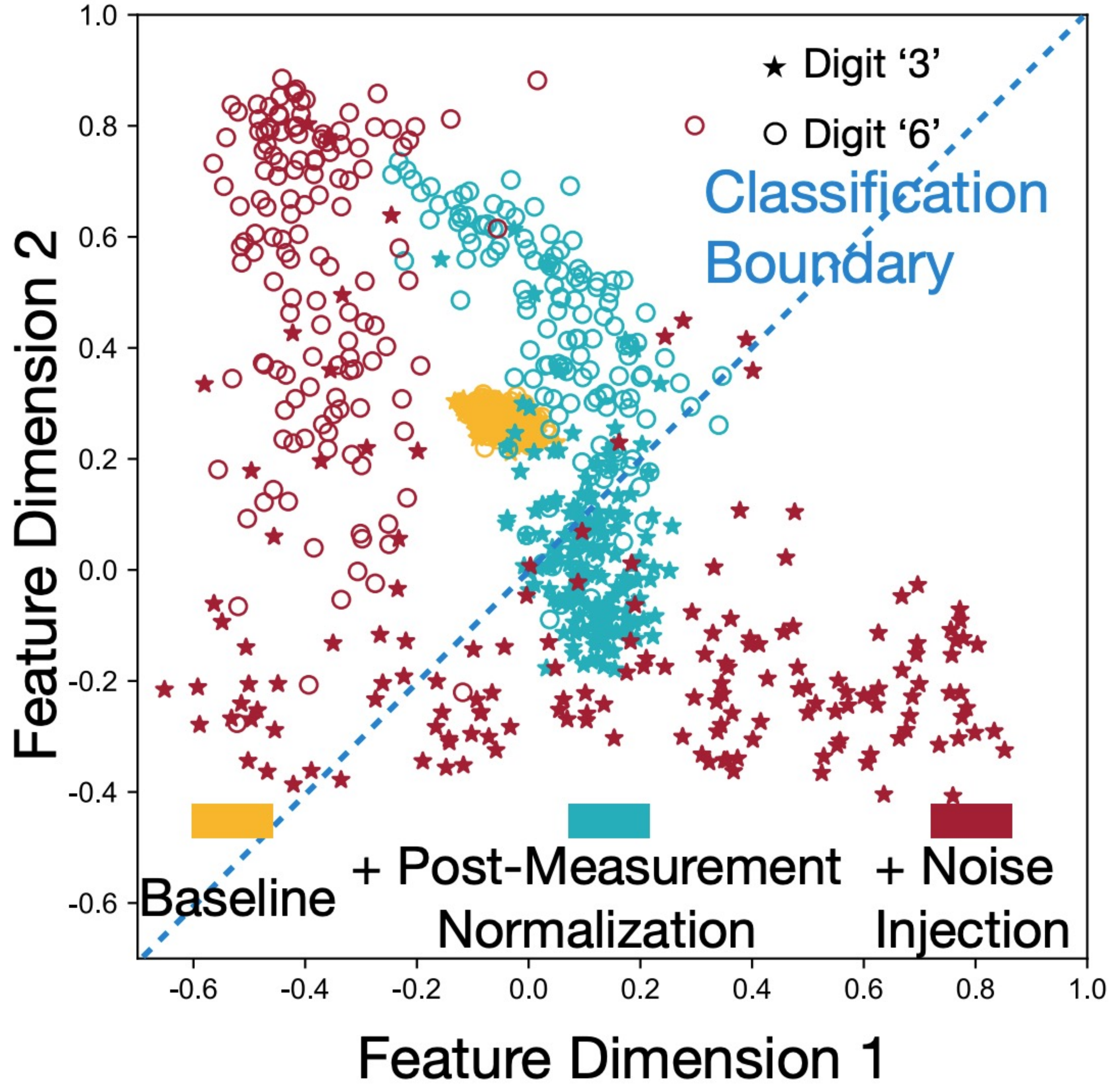




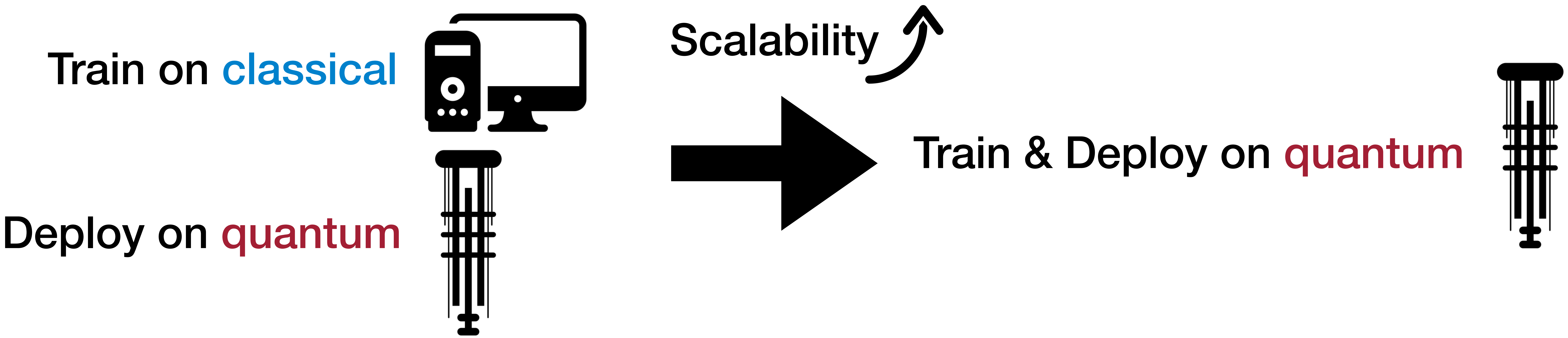
Evaluation

- **On classification tasks with QNN**





QOC (Quantum On-Chip Training)



[DAC'22] Wang et, at, QOC: Quantum On-Chip Training with Parameter Shift and Gradient Pruning

Thank you for listening!

- Take home
  - **TorchQuantum:** fast open-source **library** for quantum ML system
  - **QuantumNAS & QuantumNAT:** framework to search for **noise-robust** circuit architecture and train parameters
  - **QOC: train on quantum**, test on quantum

Torch Quantum

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

qmlsys.mit.edu