QuantumNAS: Noise-Adaptive Search for Robust Quantum Circuits

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

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

Torch
Quantum

HPCA 2022

Outline

- Overview
- Background
- QuantumNAS
- Evaluation
- TorchQuantum Library
- Conclusion

Outline

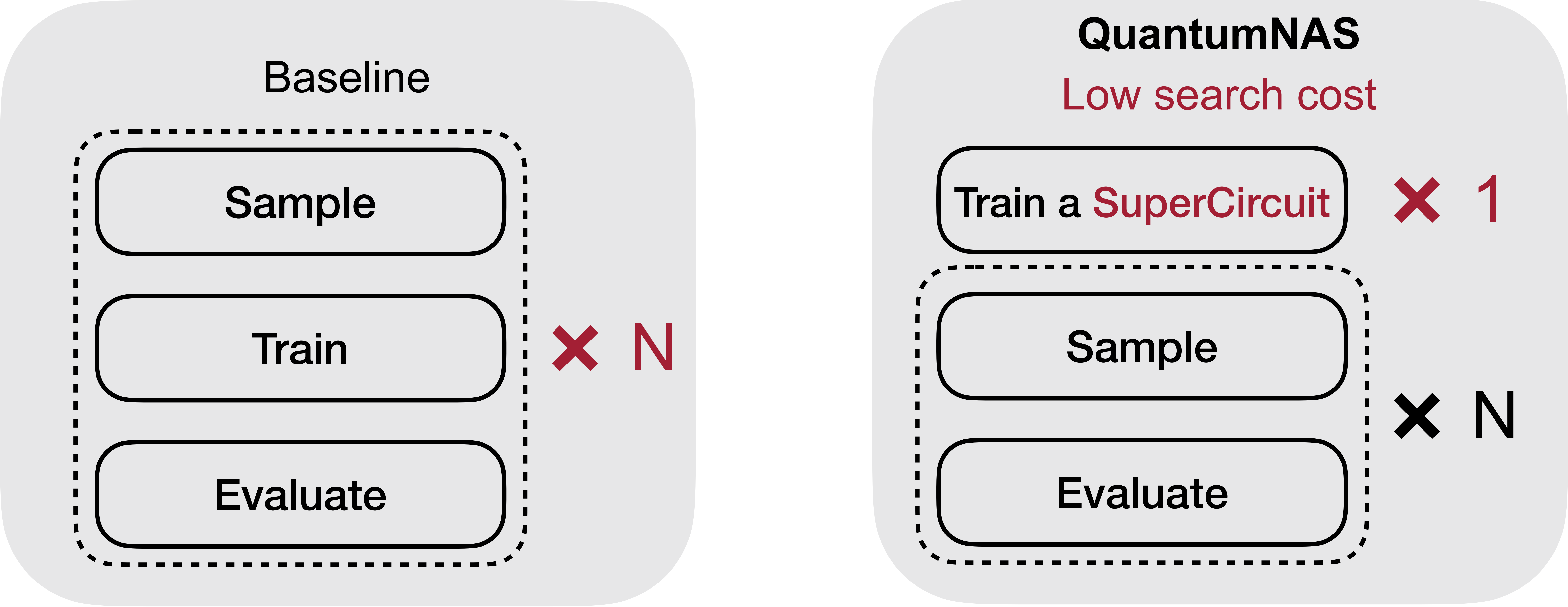
- Overview
- Background
- QuantumNAS
- Evaluation
- TorchQuantum Library
- Conclusion

Overview — QuantumNAS: Noise-Adaptive Search

- Quantum circuits are noisy
  - More gates: higher capacity, but also higher noise

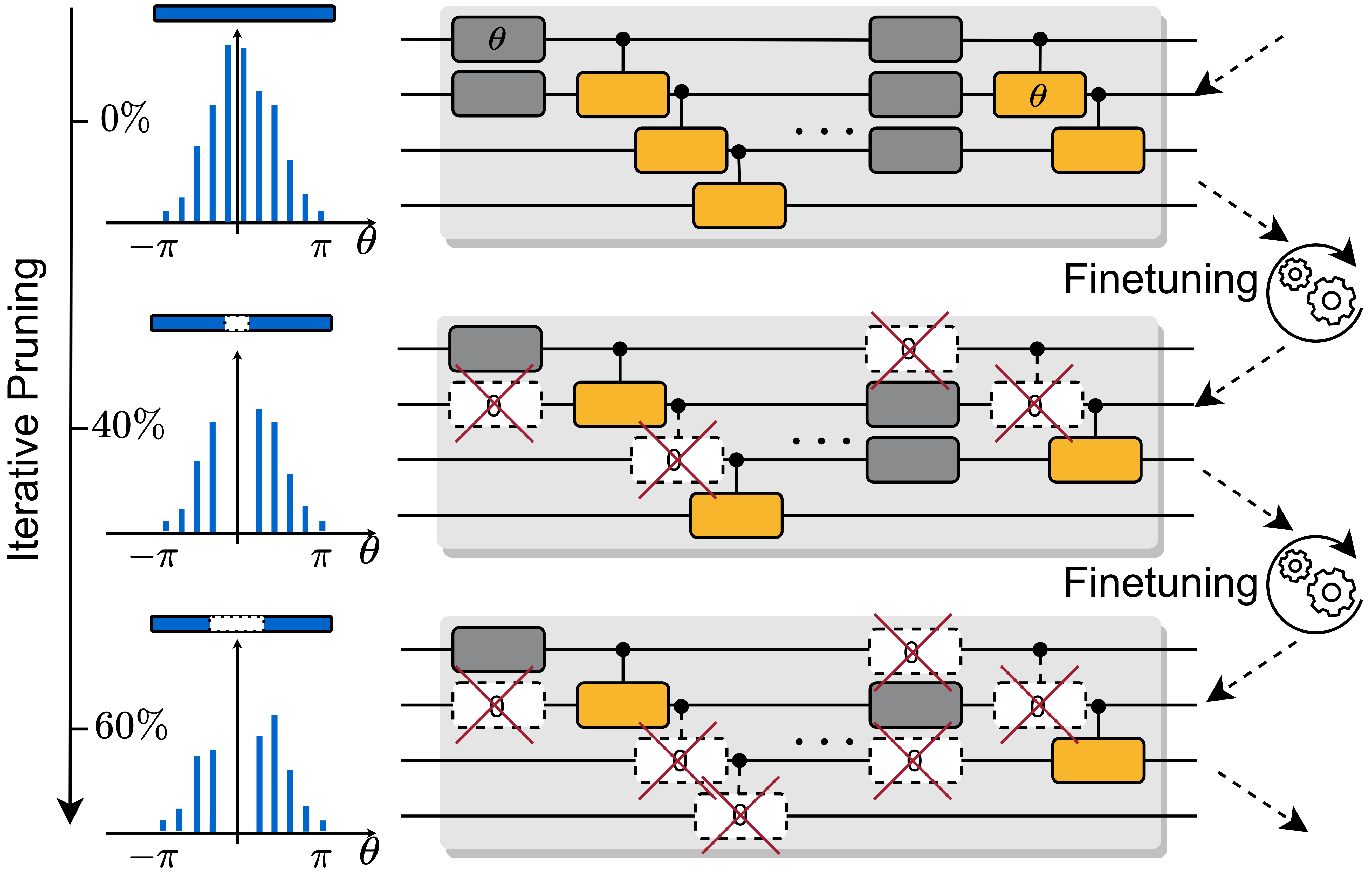
Overview — QuantumNAS: Noise-Adaptive Search

- Quantum circuits are noisy
  - More gates: higher capacity, but also higher noise
- Need to **search** for noise-robust circuit architecture
  - Naive search: train **each** possible circuit individually
  - QuantumNAS: train all circuits at **once,** amortize training cost



Overview —
Iterative Quantum Gate Pruning

- Gates with small parameter have small impact on results
- **Iteratively prune** the gates with small parameters and fine-tune the remaining ones

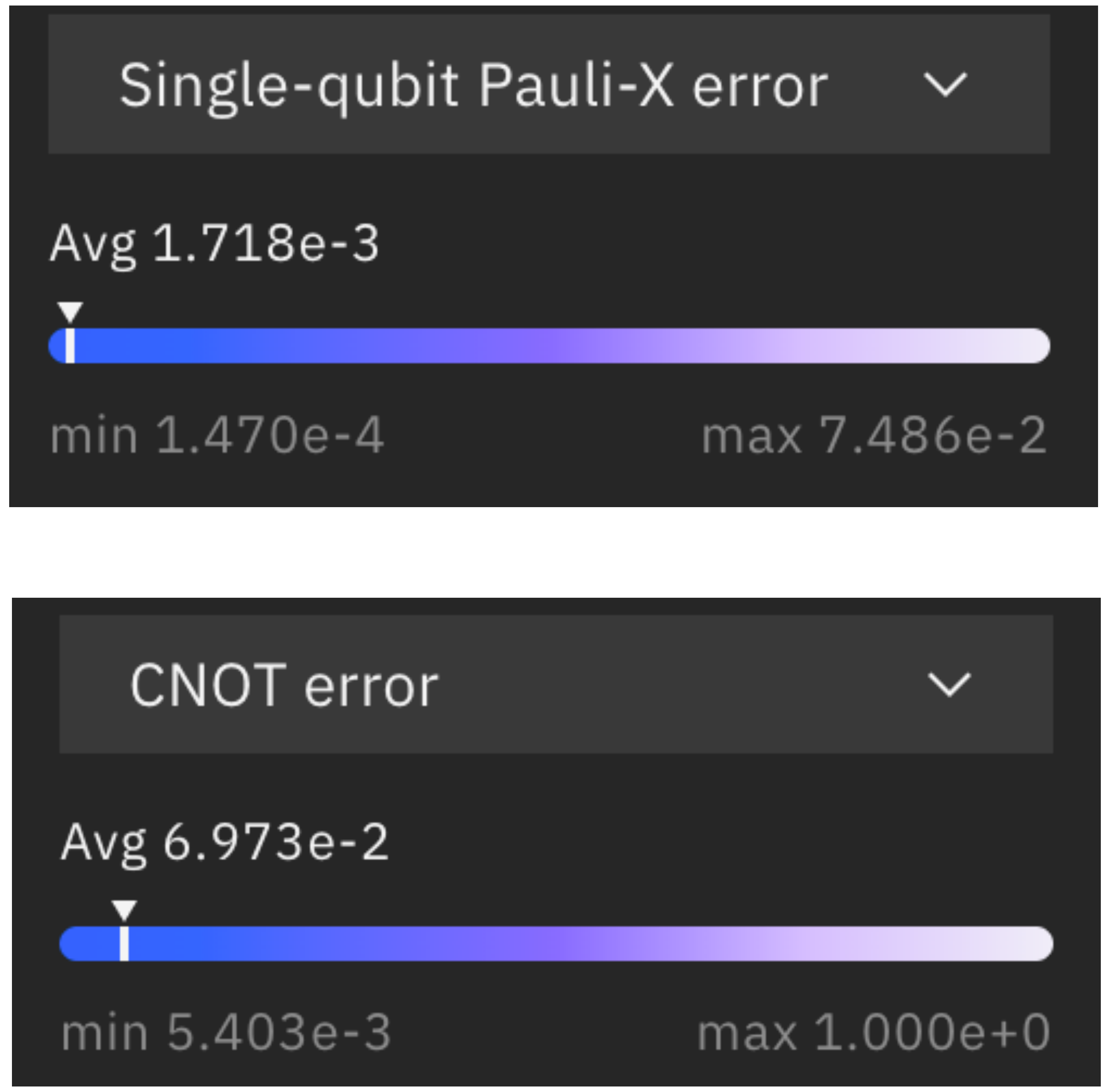


Outline

- Overview
- Background
- QuantumNAS
- Evaluation
- TorchQuantum Library
- Conclusion

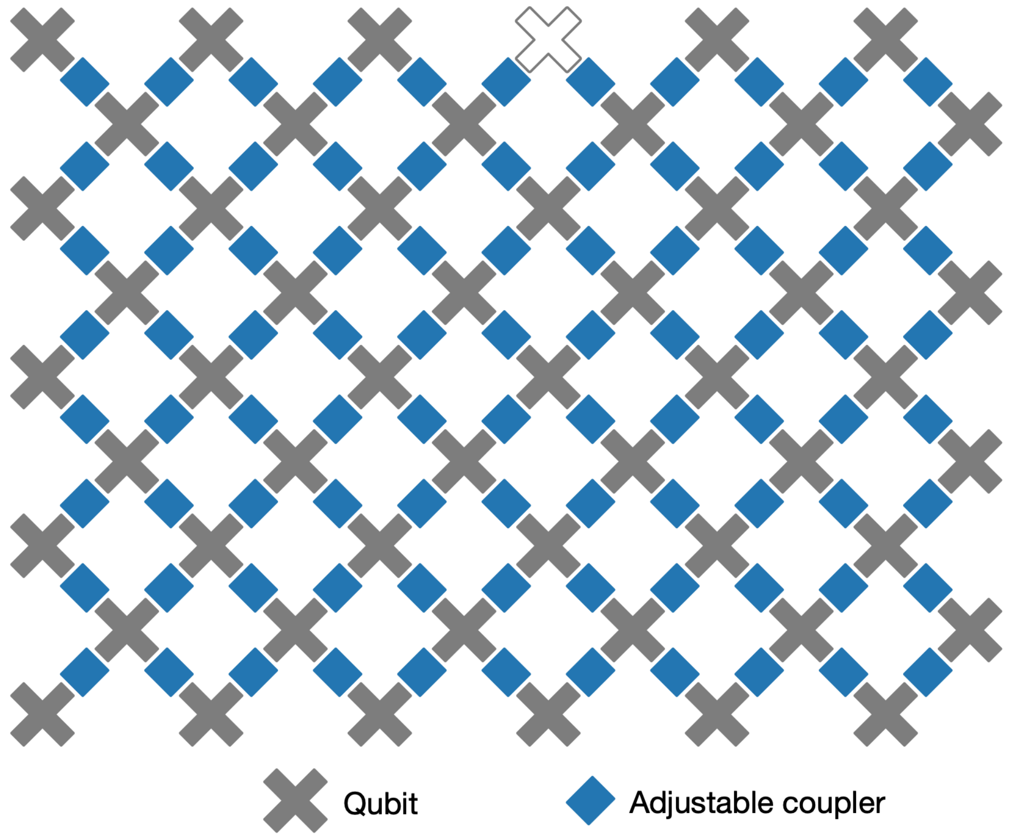
NISQ Era

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



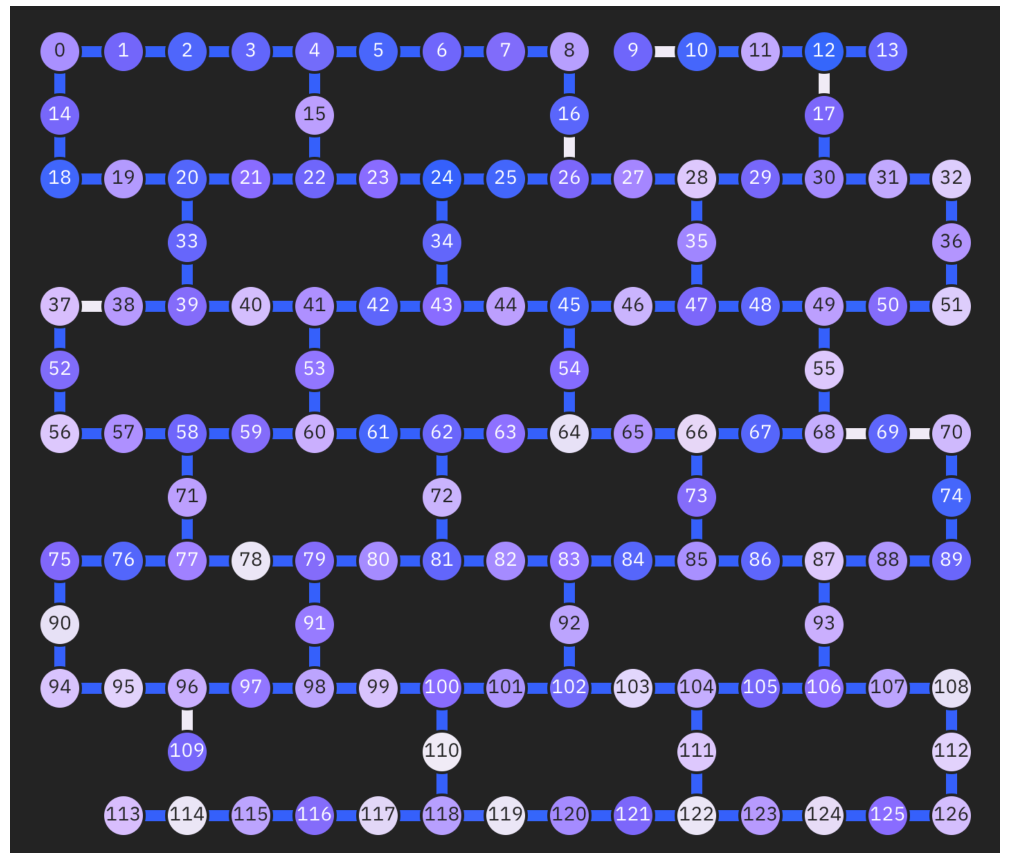
Gate Error Rate

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



Google Sycamore

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

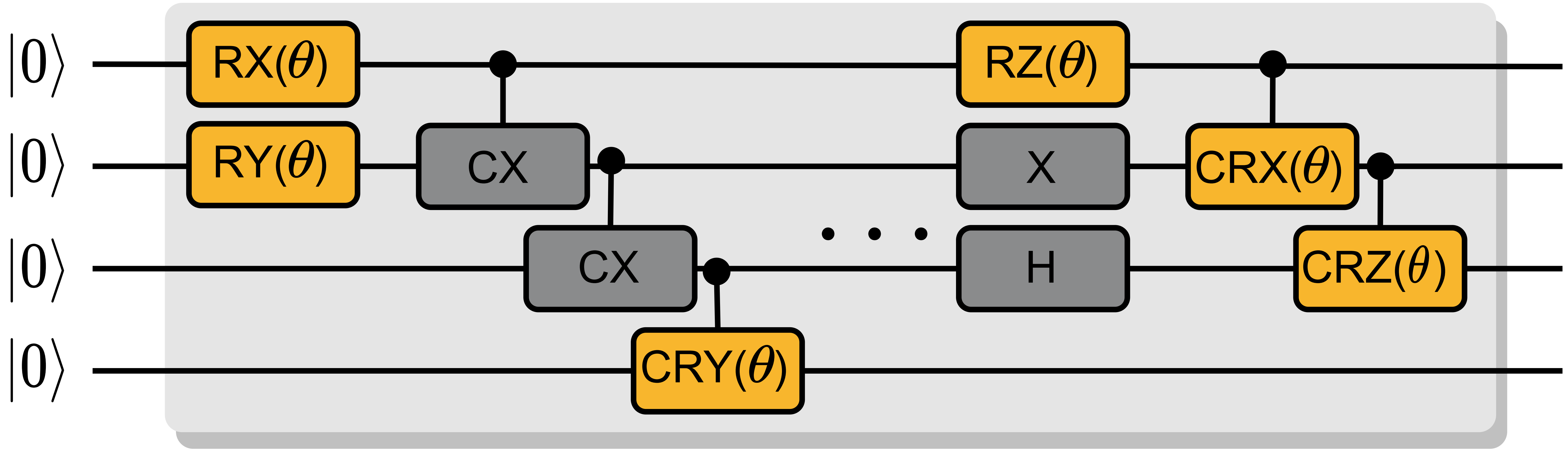


IBM Washington

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

Parameterized Quantum Circuits

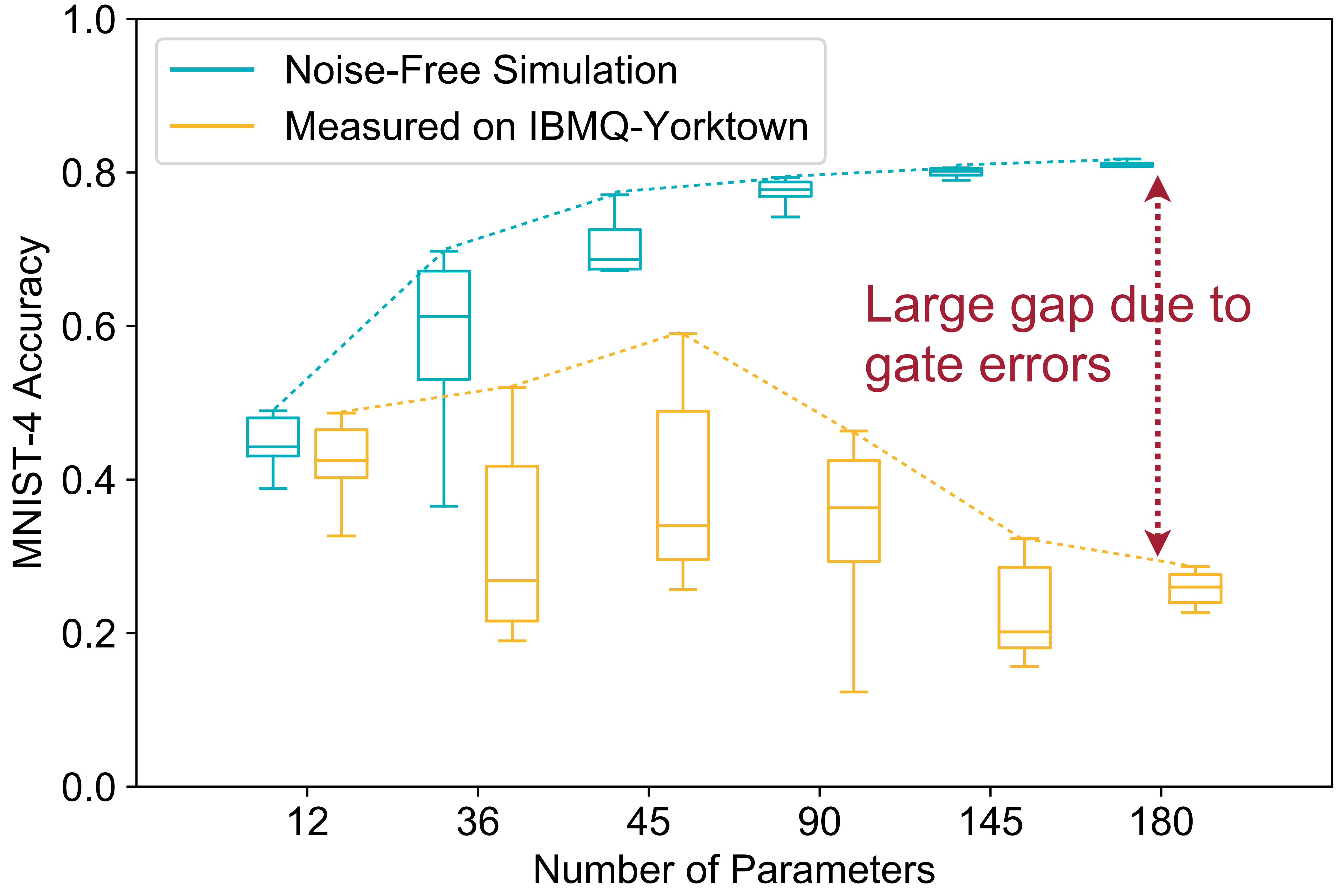
- Parameterized Quantum Circuits (PQC)
- Quantum circuit with fixed gates and parameterized gates



- 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)

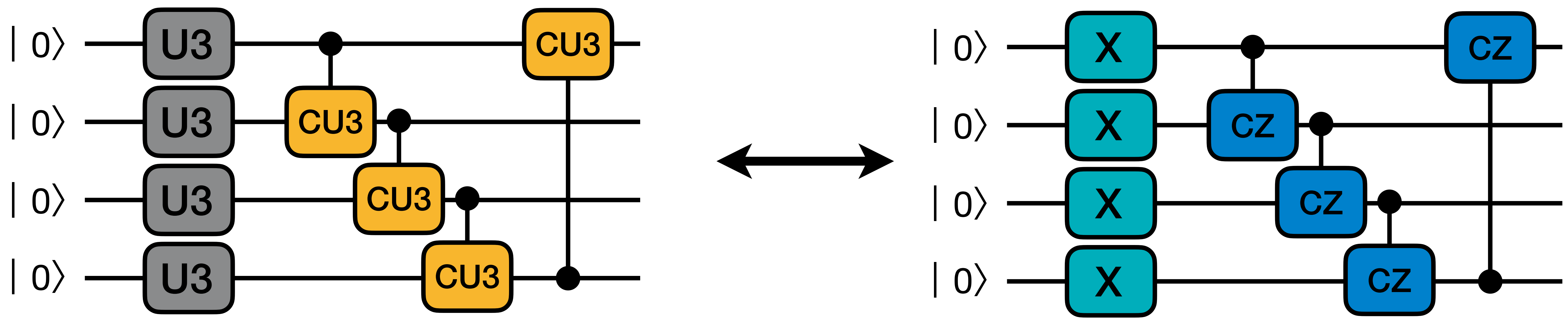
Challenges of PQC — Noise

- Noise **degrades** the Parameterized Quantum Circuit (PQC) reliability
- More parameters increase the noise-free accuracy but degrade the measured accuracy
- Under same #parameters, measured accuracy of different circuit architecture (ansatz) varies a lot
- Therefore, circuit architecture is critical



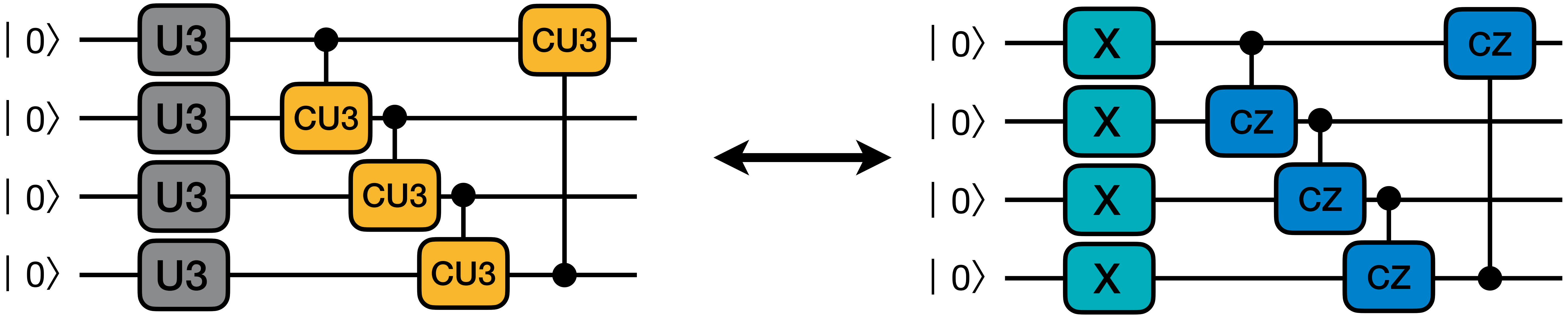
Challenges of PQC — Large Design Space

- Large design space for circuit architecture
  - Type of gates

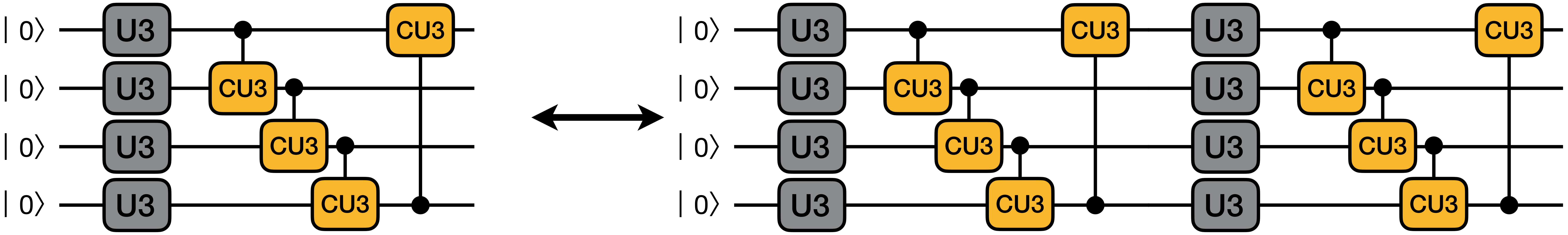


Challenges of PQC — Large Design Space

- Large design space for circuit architecture
  - Type of gates

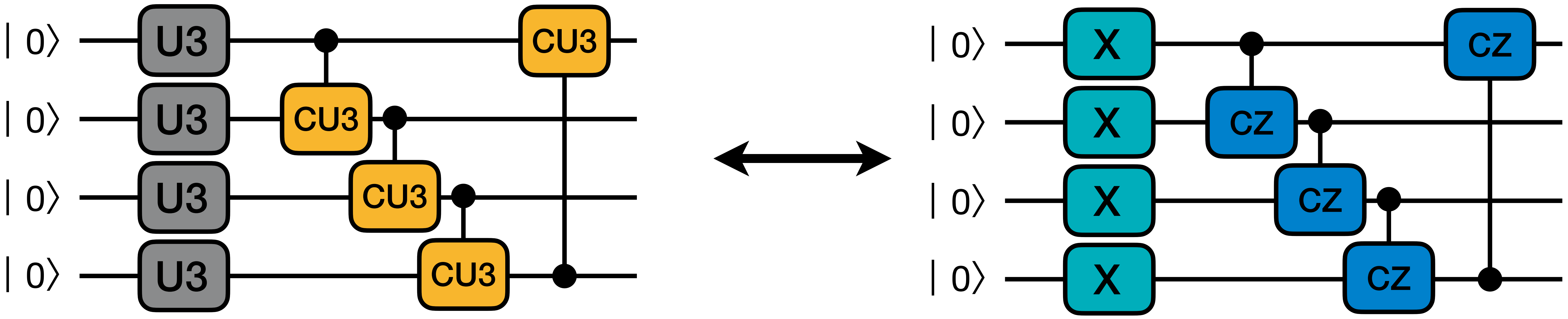


  - Number of gates

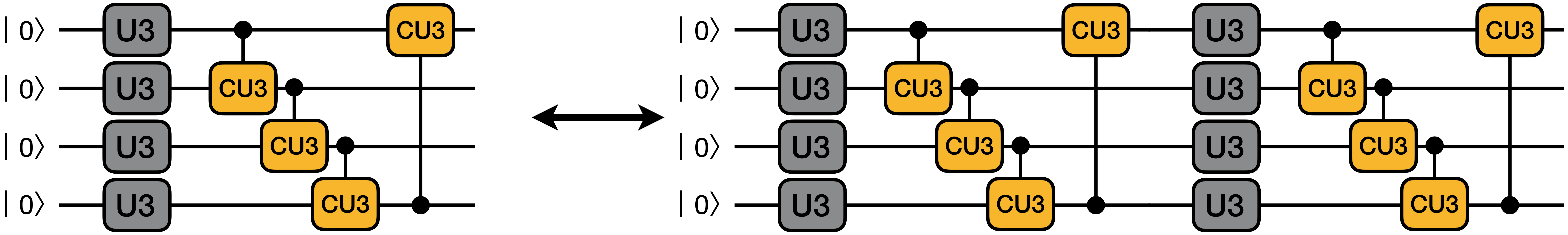


Challenges of PQC — Large Design Space

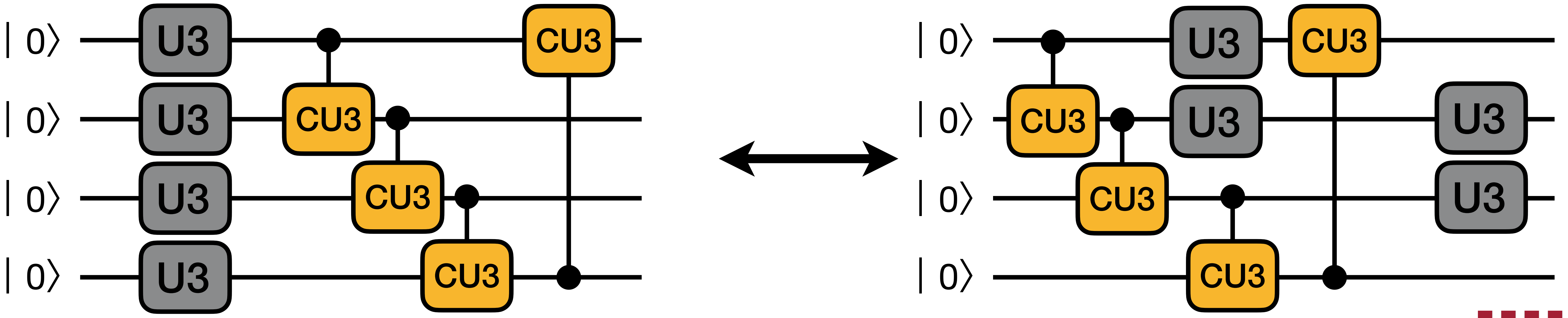
- Large design space for circuit architecture
  - Type of gates



  - Number of gates



  - Position of gates



Outline

- Overview
- Background
- QuantumNAS
- Evaluation
- TorchQuantum Library
- Conclusion

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: Decouple the Training and Search

Naive Search

```
For q_devices:
    For search episodes: // meta controller
        For circuit training iterations:
            update_parameters(); Expensive
        If good_circuit: break;
```

QuantumNAS: Decouple the Training and Search

Naive Search

For q_devices:
    For search episodes: // meta controller
        For circuit training iterations:
            update_parameters(); **Expensive**
        If good_circuit: **break;**

=>

QuantumNAS

For ***SuperCircuit*** training iterations: **Expensive**
    update_parameters(); **training**

decouple

For q_devices:
    For search episodes: **search**
        sample from ***SuperCircuit***; **Light-Weight**
        If good_circuit: **break;**
        //no training

QuantumNAS

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

QuantumNAS

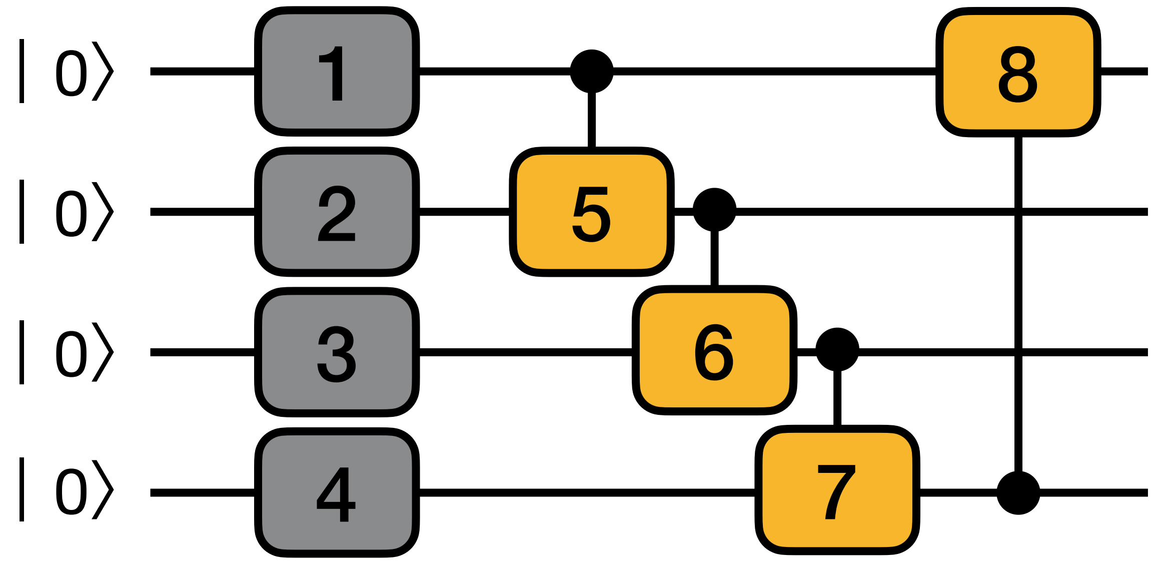
- SuperCircuit Construction and Training
- Noise-Adaptive Evolutionary Co-Search of SubCircuit and Qubit Mapping
- Train the Searched SubCircuit
- Iterative Quantum Gate Pruning

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
  - Contains 2^8 = 256 candidates

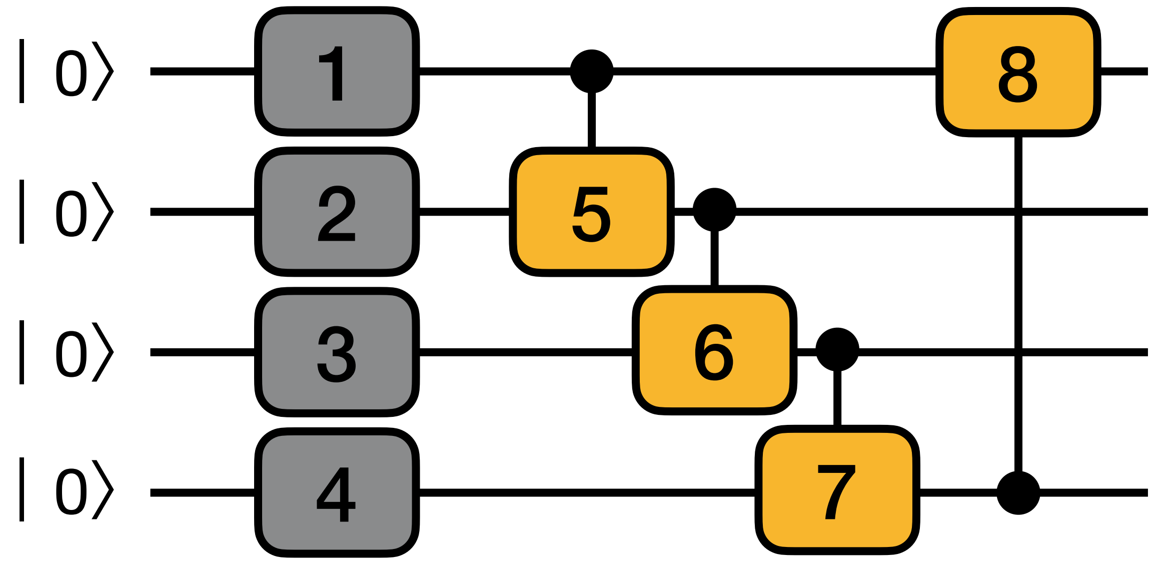
SuperCircuit & SubCircuit

- 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
  - Contains 2^8 = 256 candidates
- SuperCircuit: the circuit with the **largest** number of gates in the design space
  - Example: SuperCircuit in U3+CU3 space

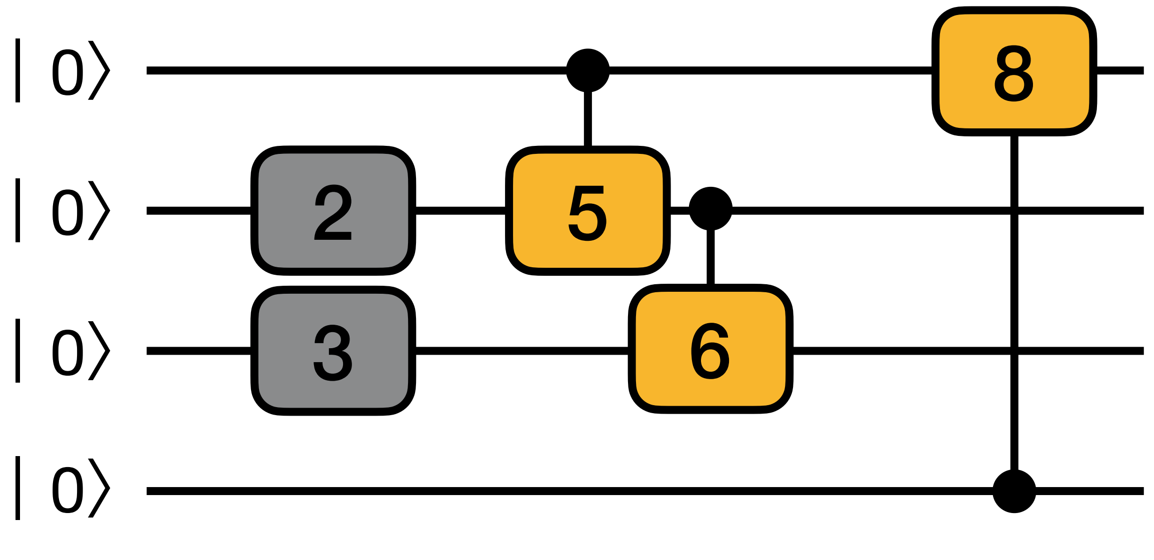


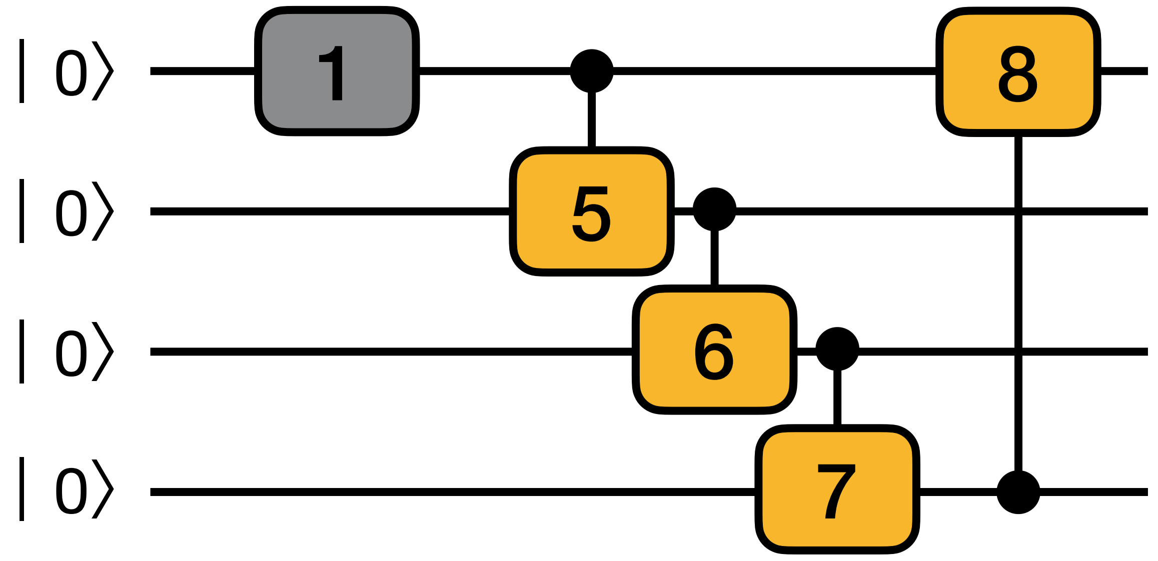
SuperCircuit & SubCircuit

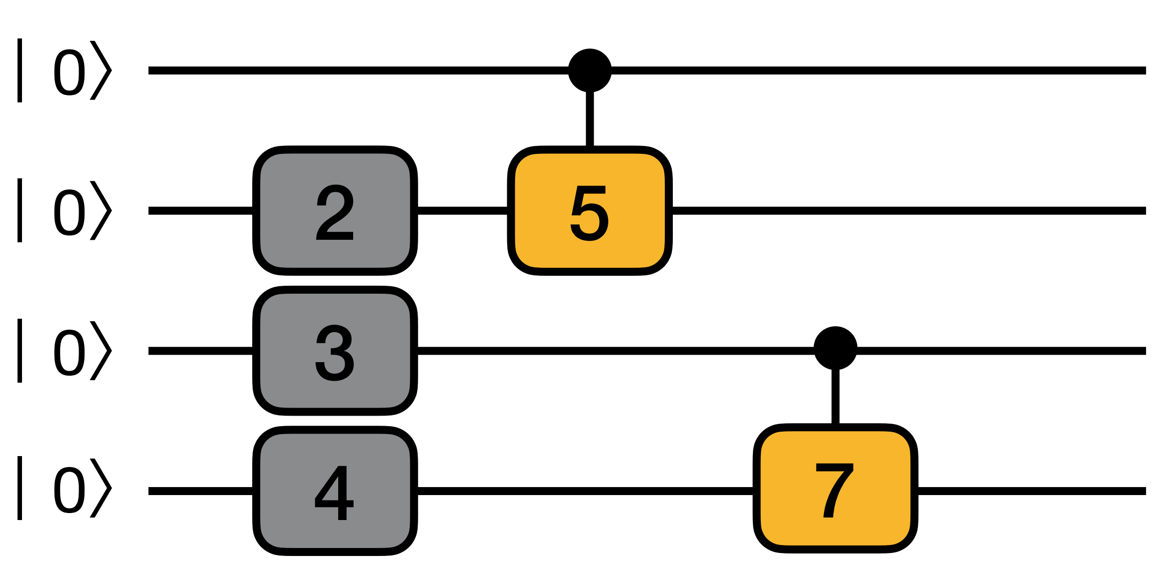
- Firstly construct a design space. For example, a design space of maximum 4 U3 in the first layer and 4 CU3 gates in the second layer
  - Contains 2^8 = 256 candidates
- SuperCircuit: the circuit with the **largest** number of gates in the design space
  - Example: SuperCircuit in U3+CU3 space



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







SuperCircuit Construction

- Why use a SuperCircuit?
  - It enables **efficient** search of circuit architecture candidates with no need of training each of them individually
  - For one SubCircuit candidate, we can directly inherit parameters from SuperCircuit and consider that the SubCircuit can operate **as if it is trained individually from scratch**

SuperCircuit Construction

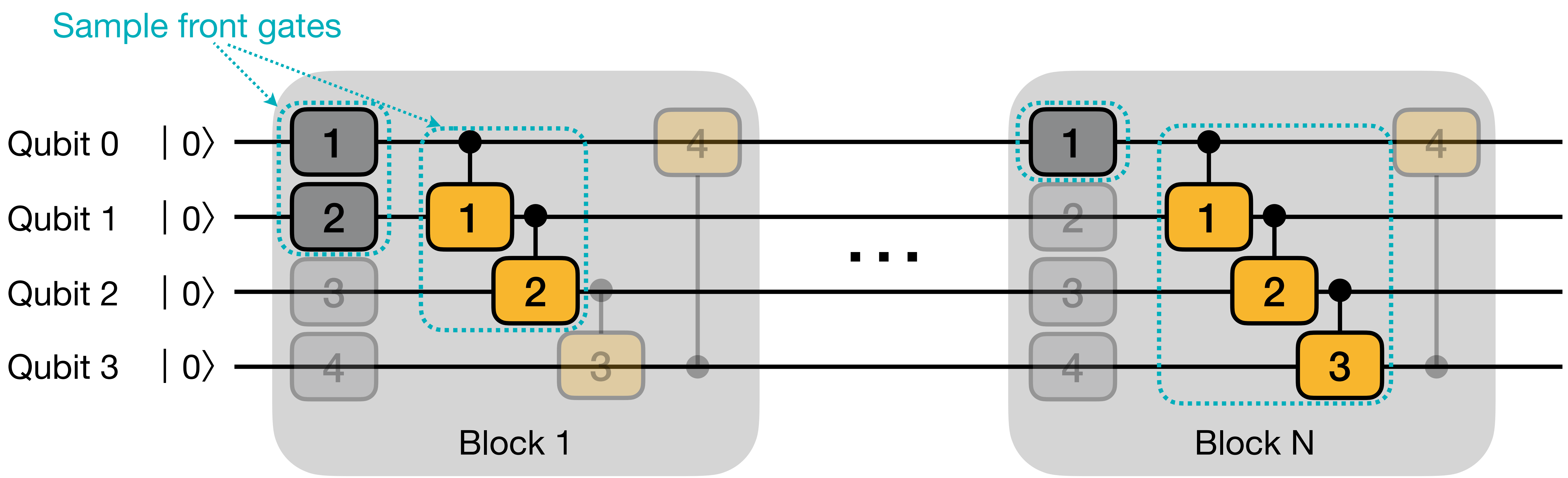
- Why use a SuperCircuit?
  - It enables **efficient** search of circuit architecture candidates with no need of training each of them individually
  - For one SubCircuit candidate, we can directly inherit parameters from SuperCircuit and consider that the SubCircuit can operate **as if it is trained individually from scratch**
- Need to prevent interference of SubCircuits from each other

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

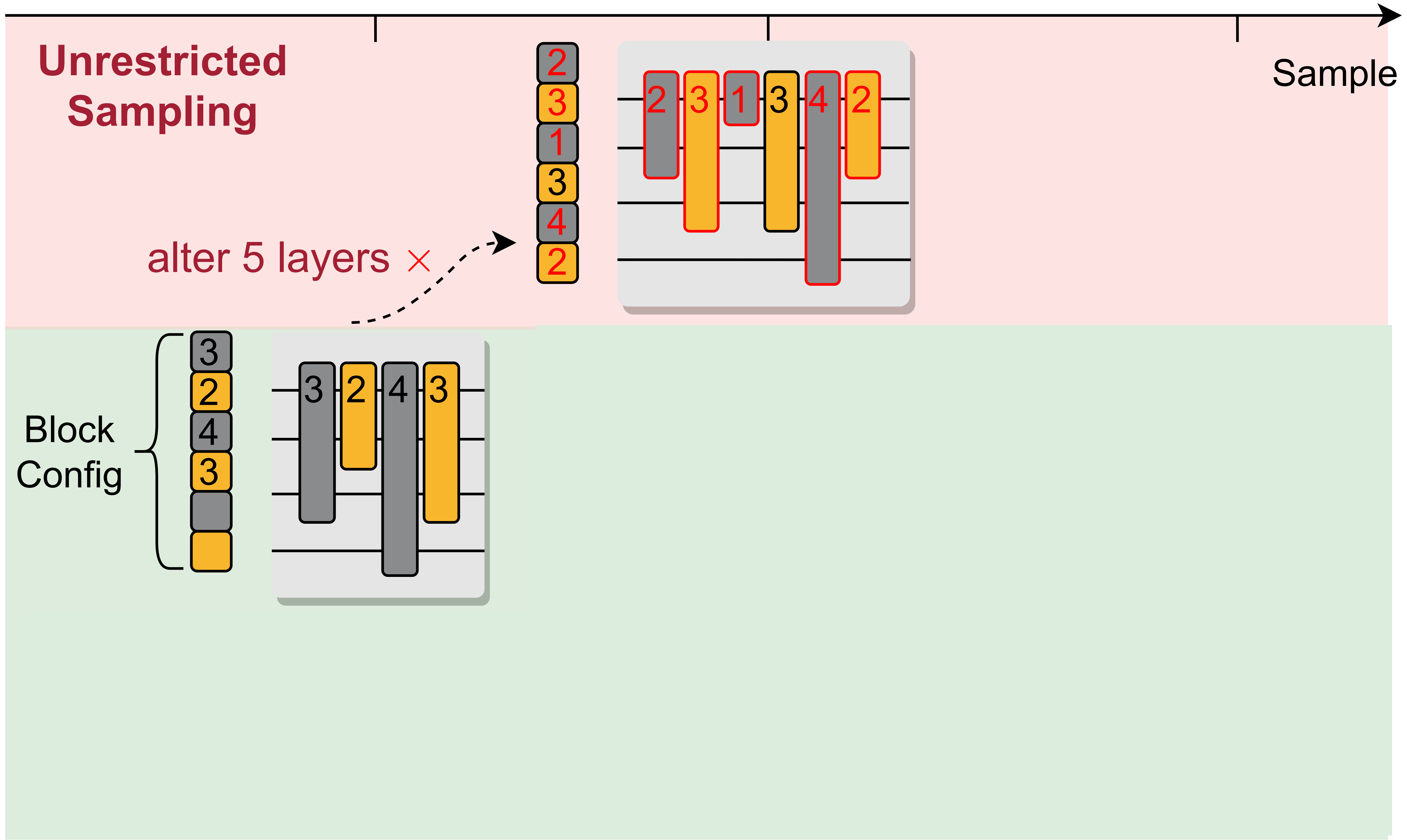
Front Sampling

- 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



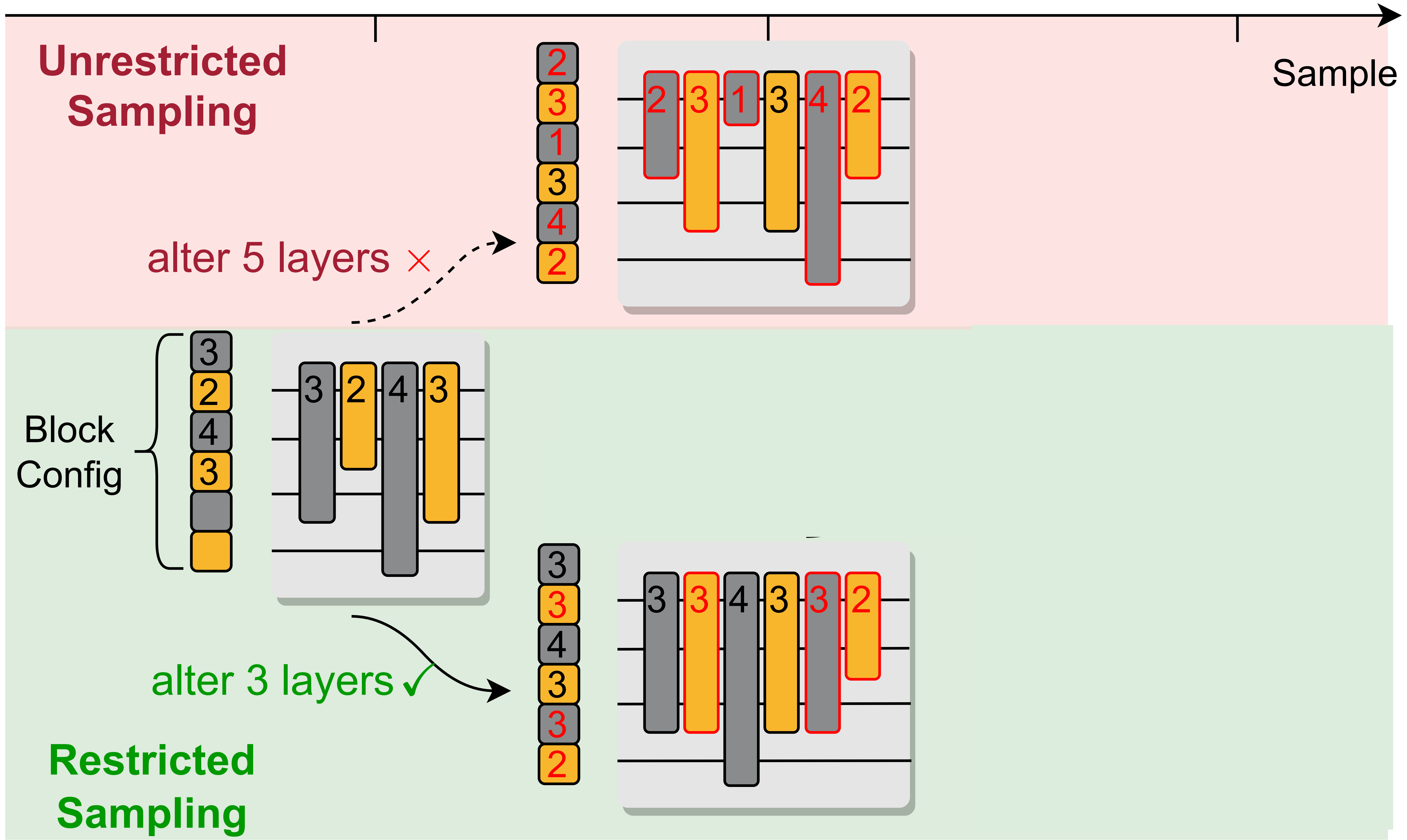
Restricted Sampling

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



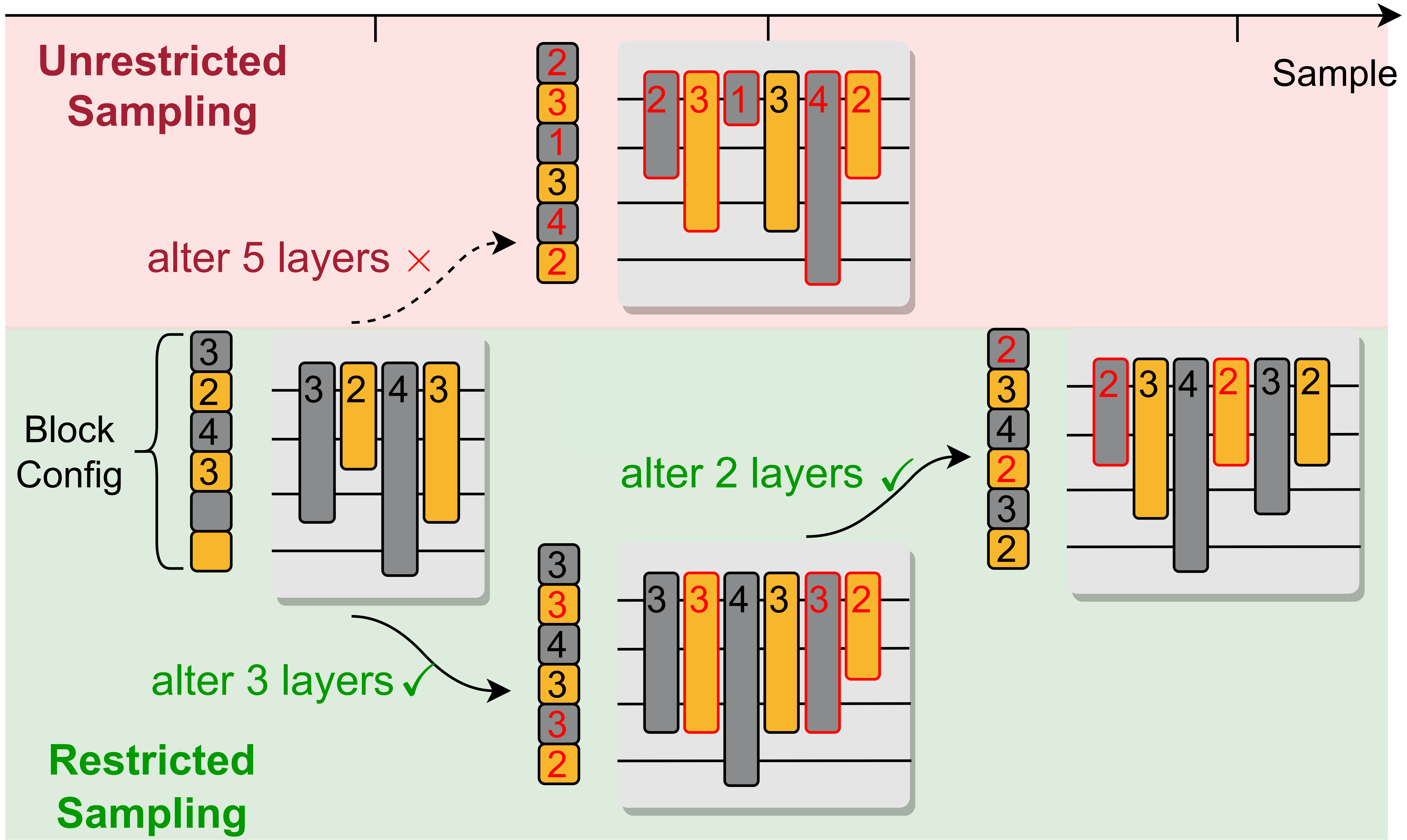
Restricted Sampling

- Restricted Sampling:
  - Restrict the difference between two consecutively sampled SubCircuits
  - For example: restrict to at most 4 different layers



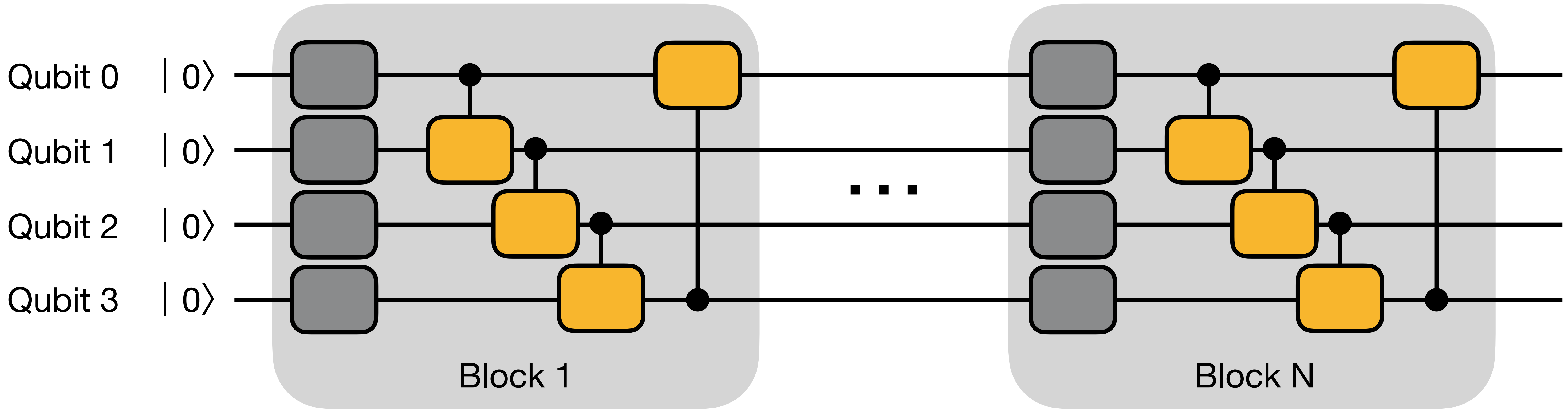
Restricted Sampling

- Restricted Sampling:
  - Restrict the difference between two consecutively sampled SubCircuits
  - For example: restrict to at most 4 different layers



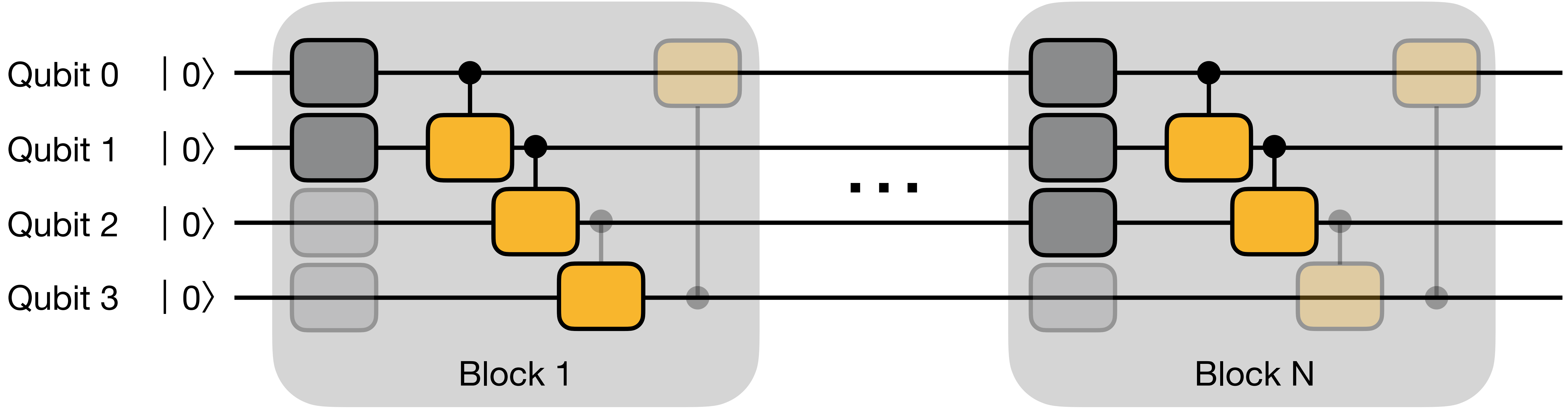
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



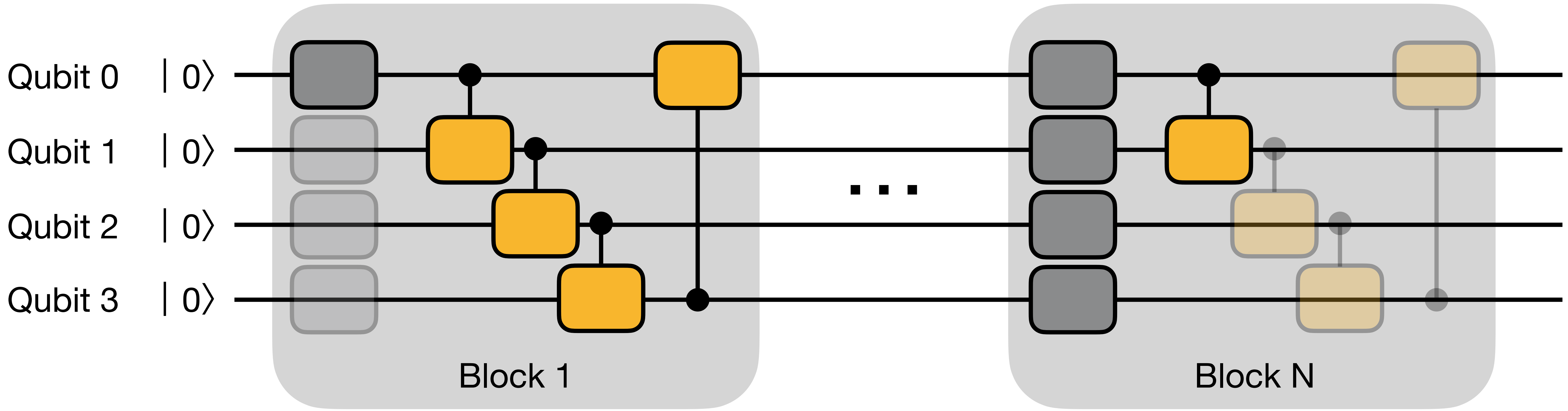
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



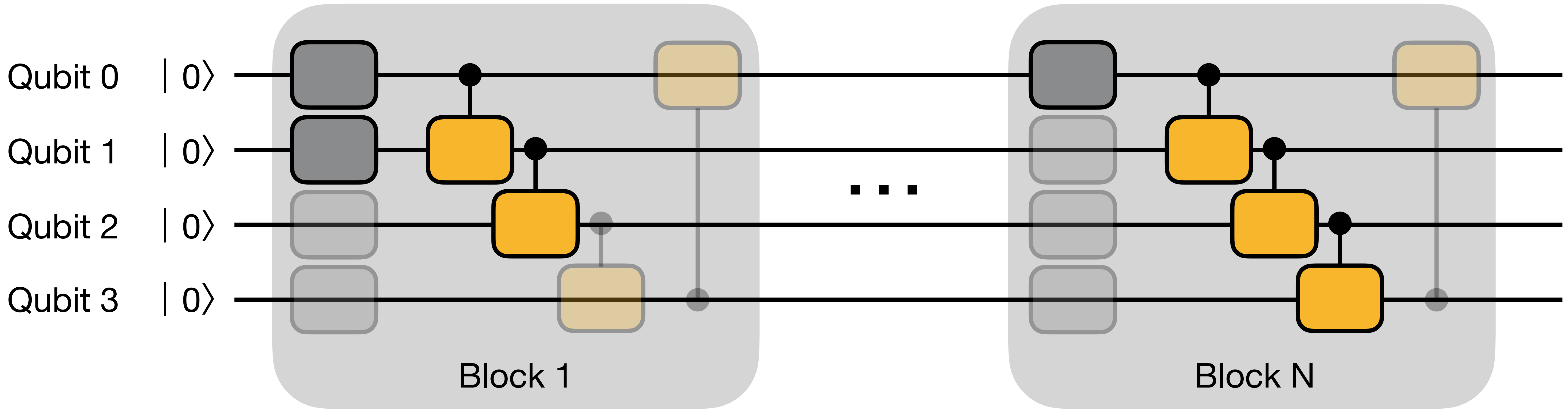
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



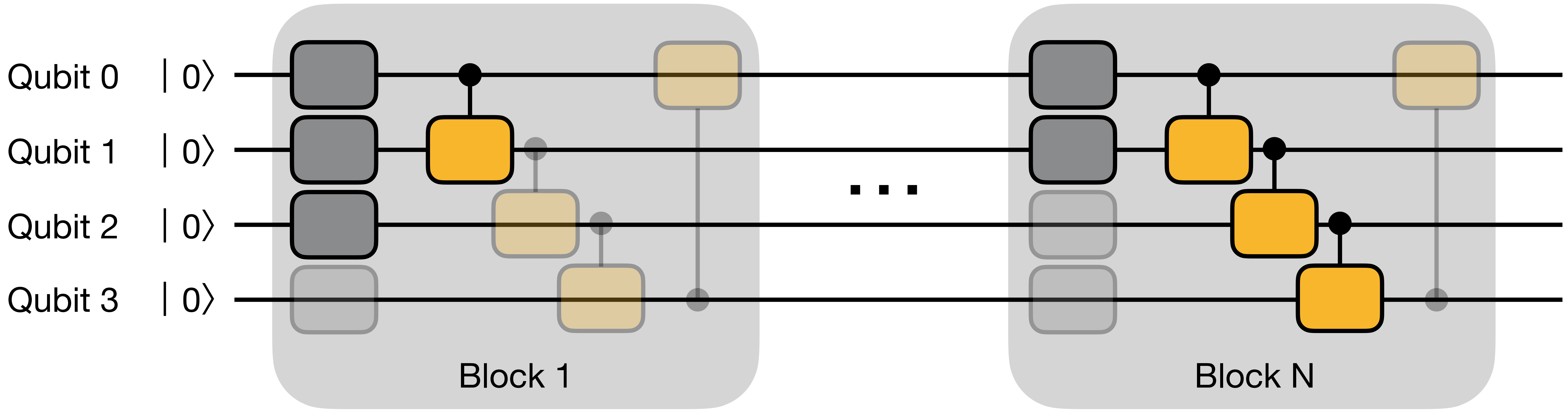
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



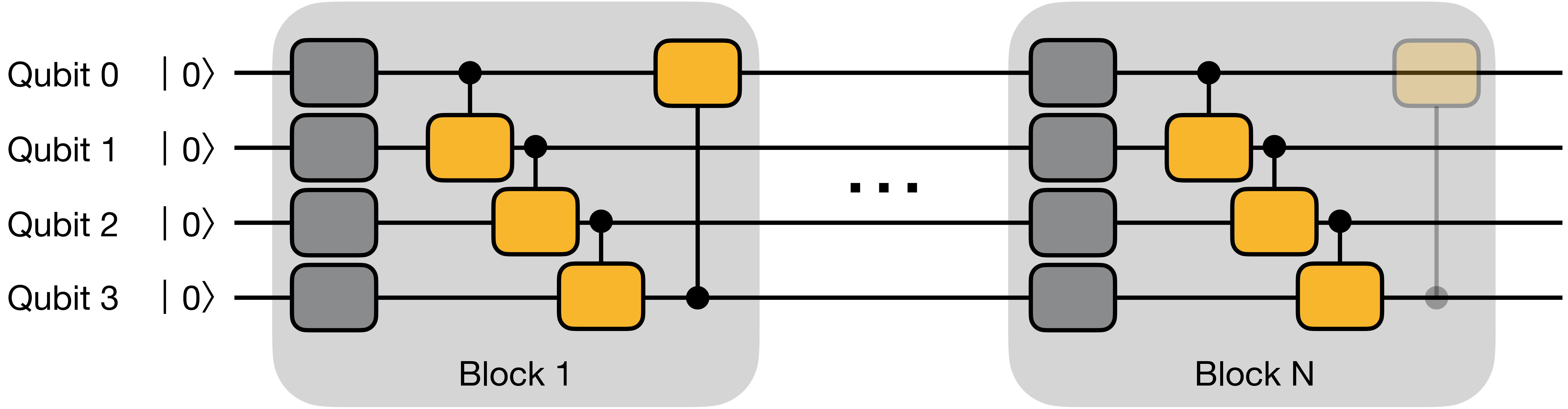
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



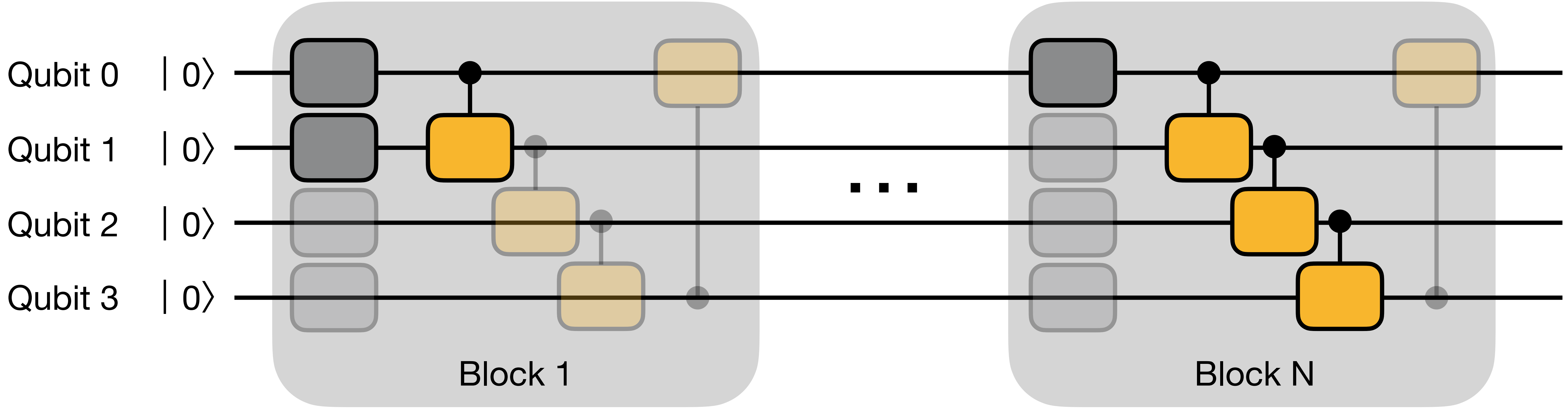
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



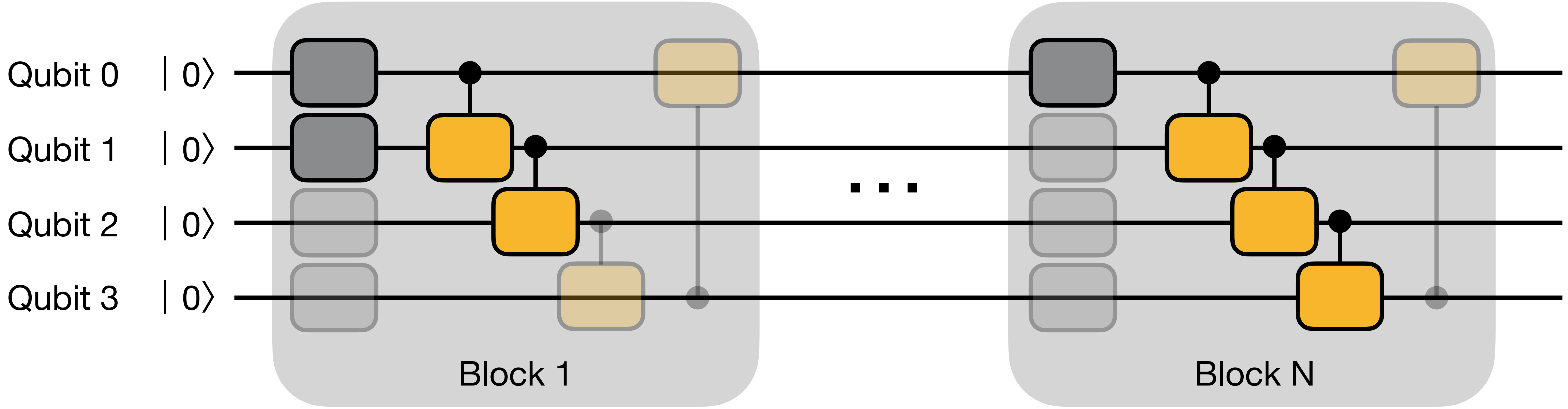
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



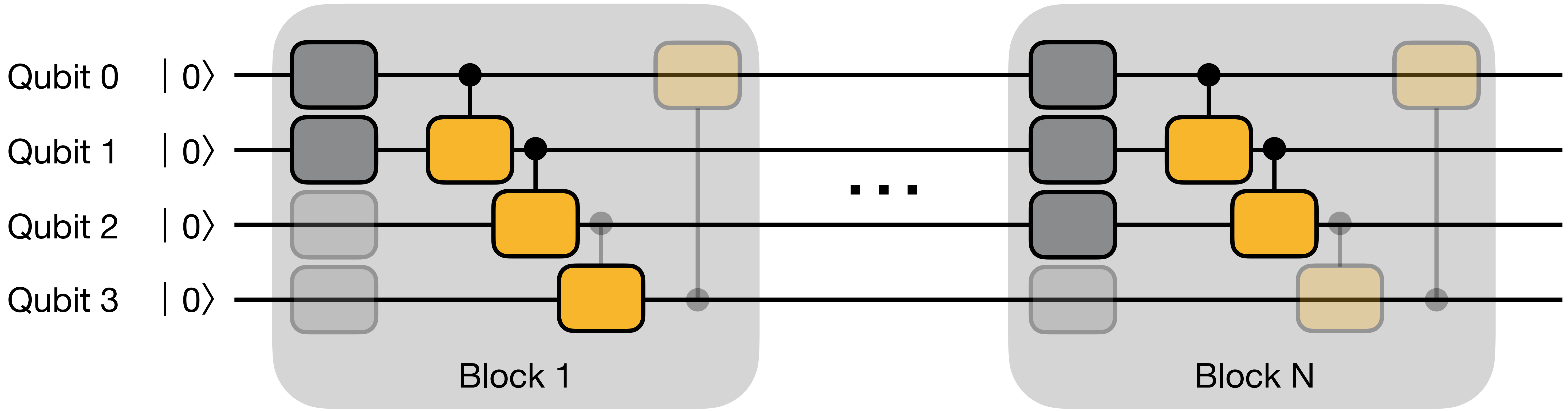
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



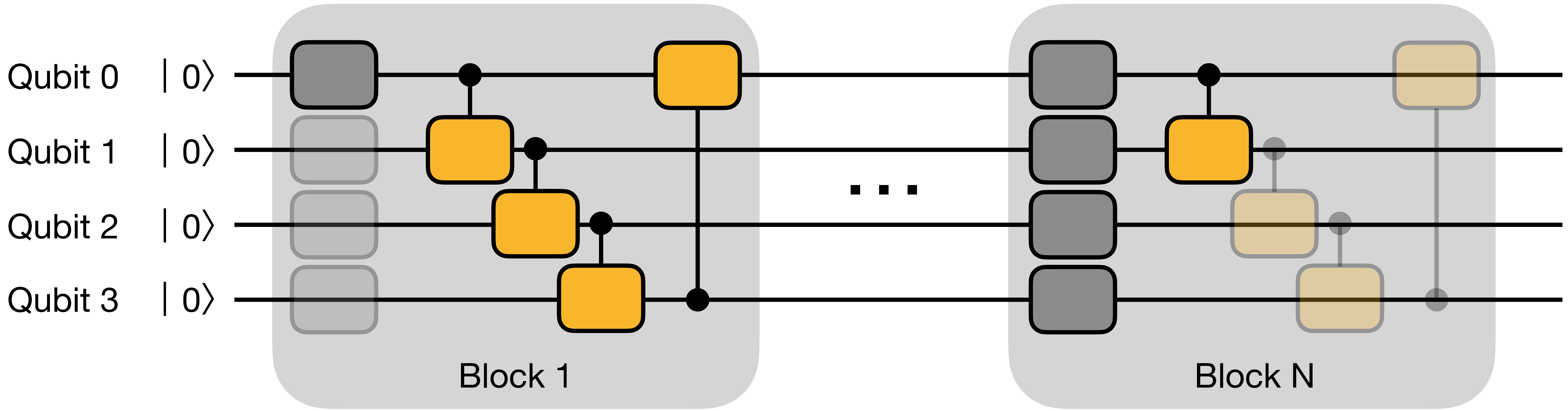
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



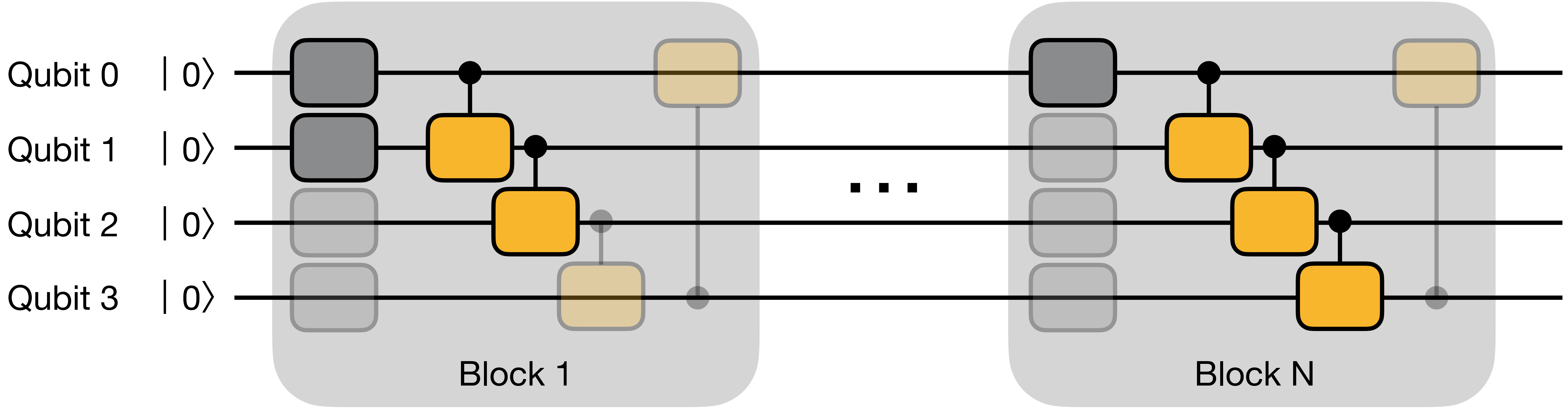
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



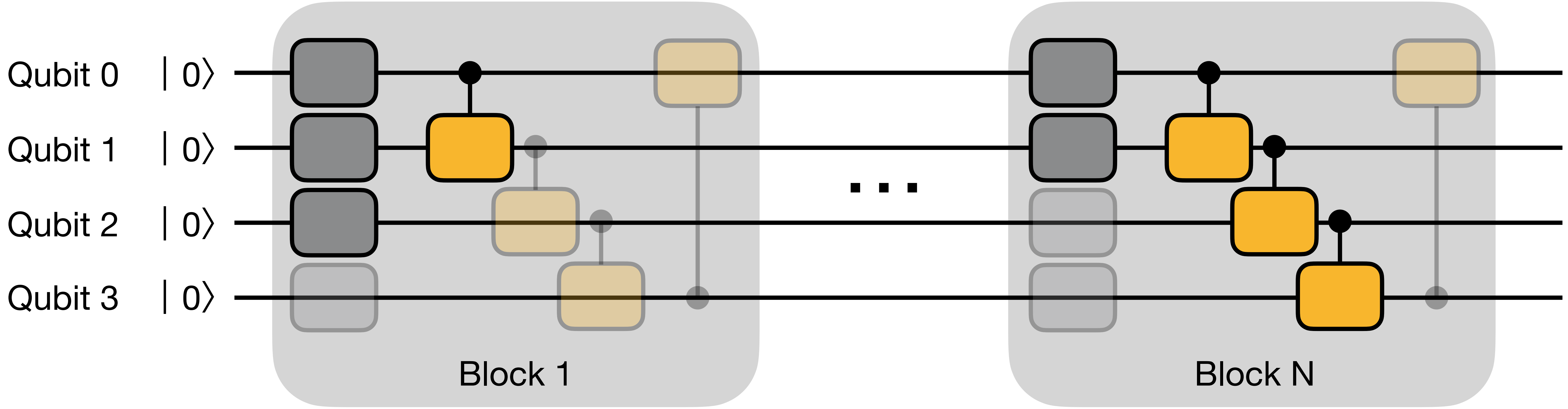
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



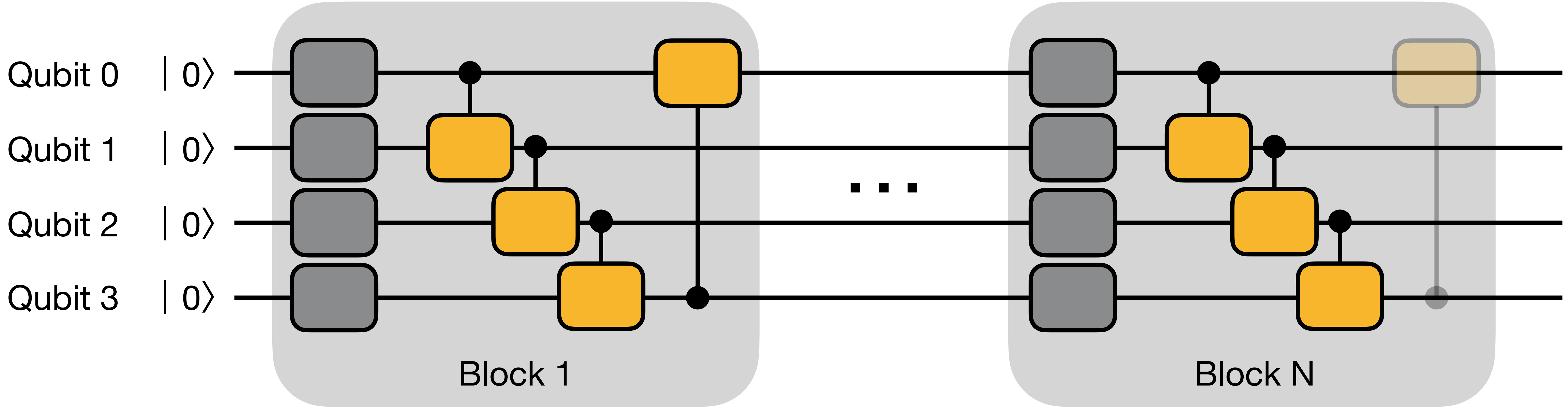
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



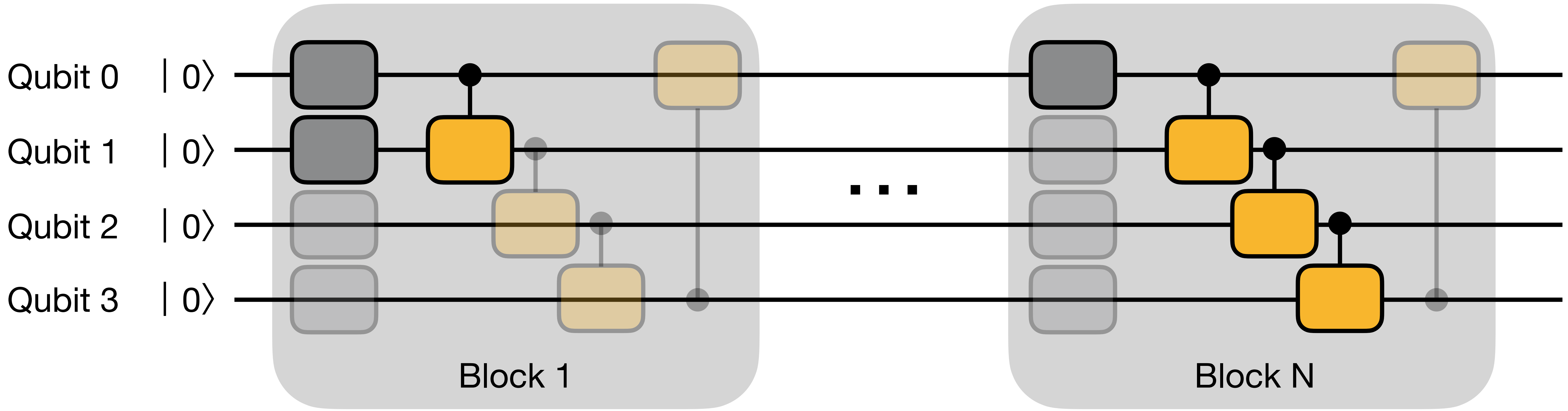
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



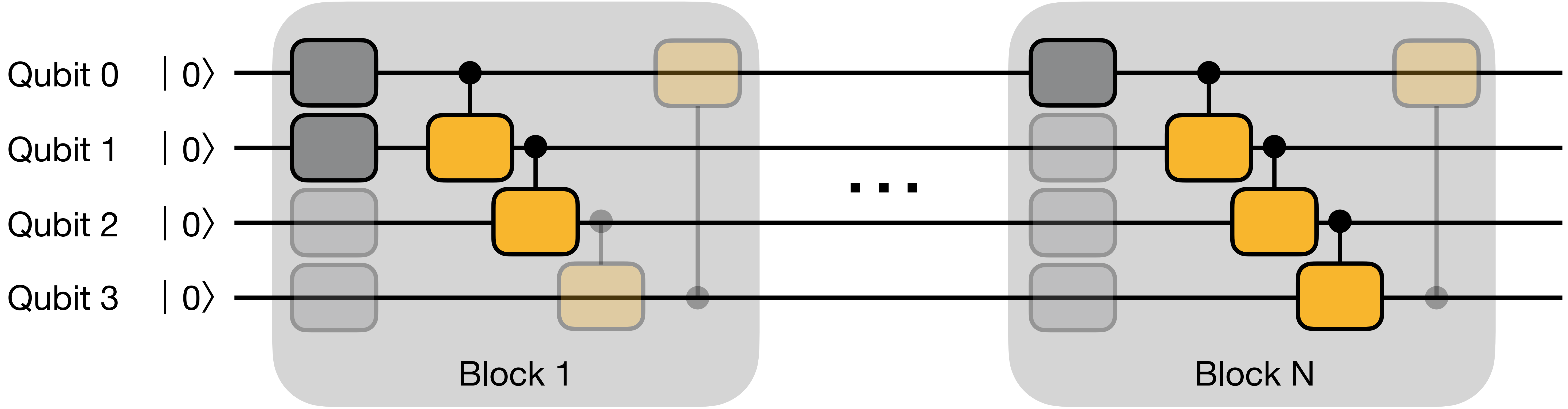
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



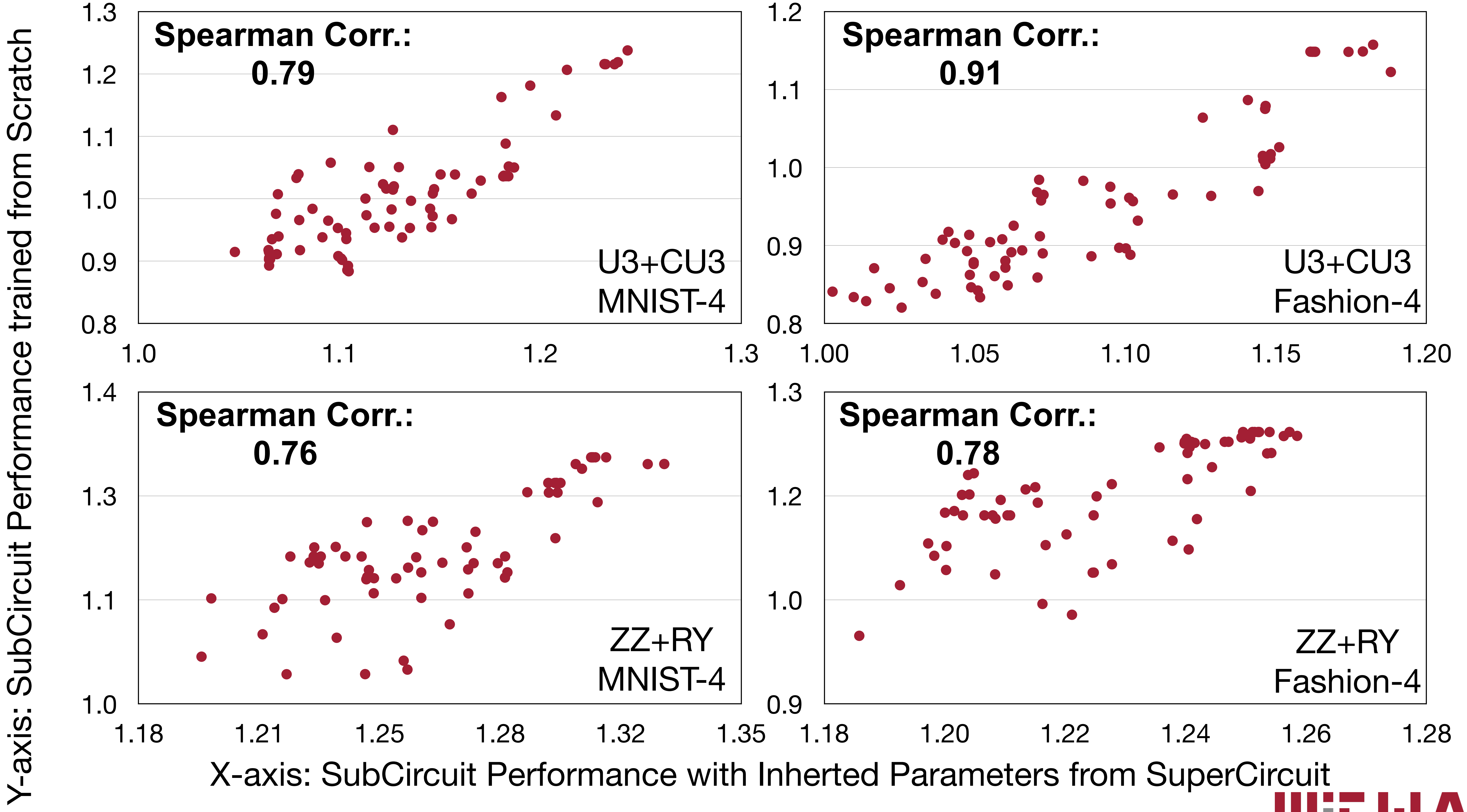
Train SuperCircuit for Multiple Steps

- In one SuperCircuit Training step: Sample and Train



How Reliable is the SuperCircuit?

- Inherited parameters from SuperCircuit can provide accurate relative performance

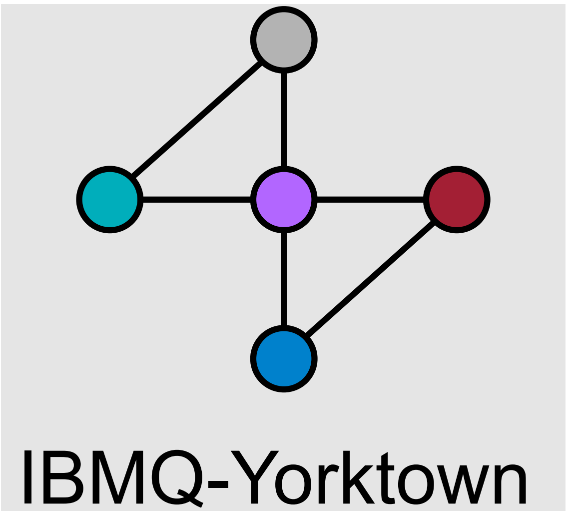


QuantumNAS

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

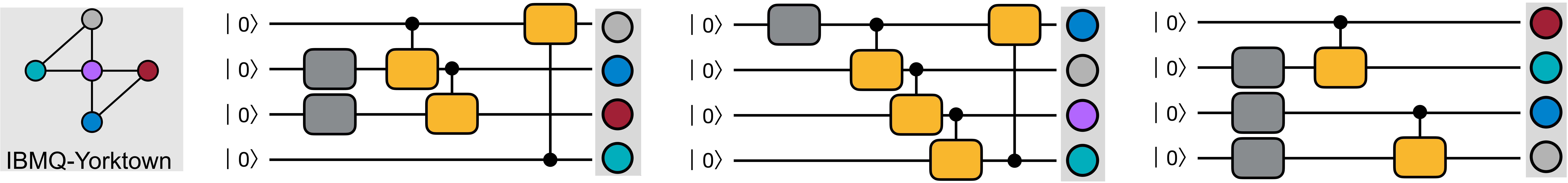
Noise-Adaptive Evolutionary Co-Search

- 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

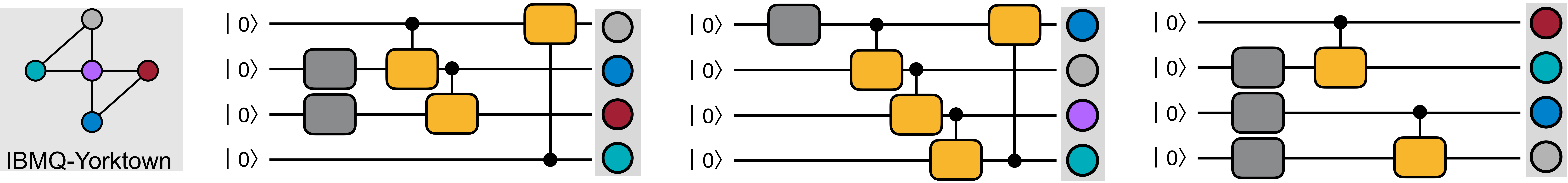
SubCircuit and Qubit Mapping pairs →

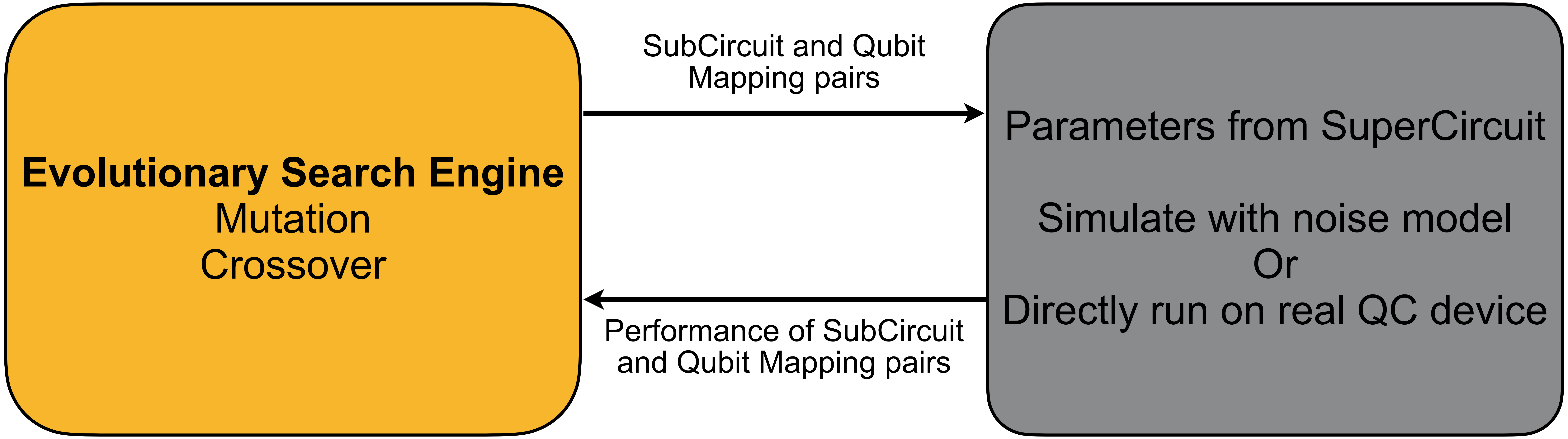
Parameters from SuperCircuit

Simulate with noise model
Or
Directly run on real QC device

Noise-Adaptive Evolutionary Co-Search

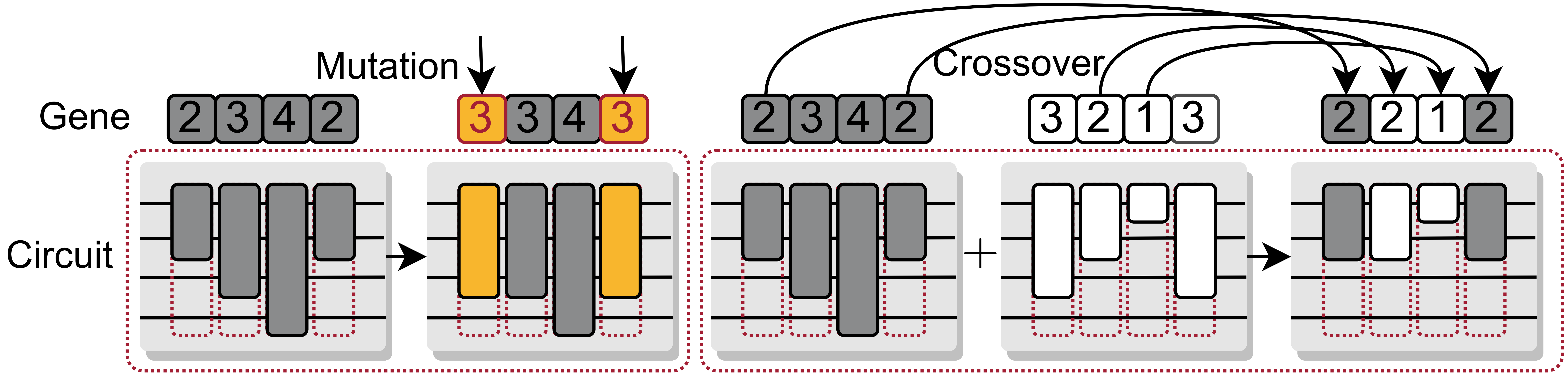
- Search the best SubCircuit and its qubit mapping on target device





Mutation and Crossover

- Mutation and crossover create new SubCircuit candidates



QuantumNAS

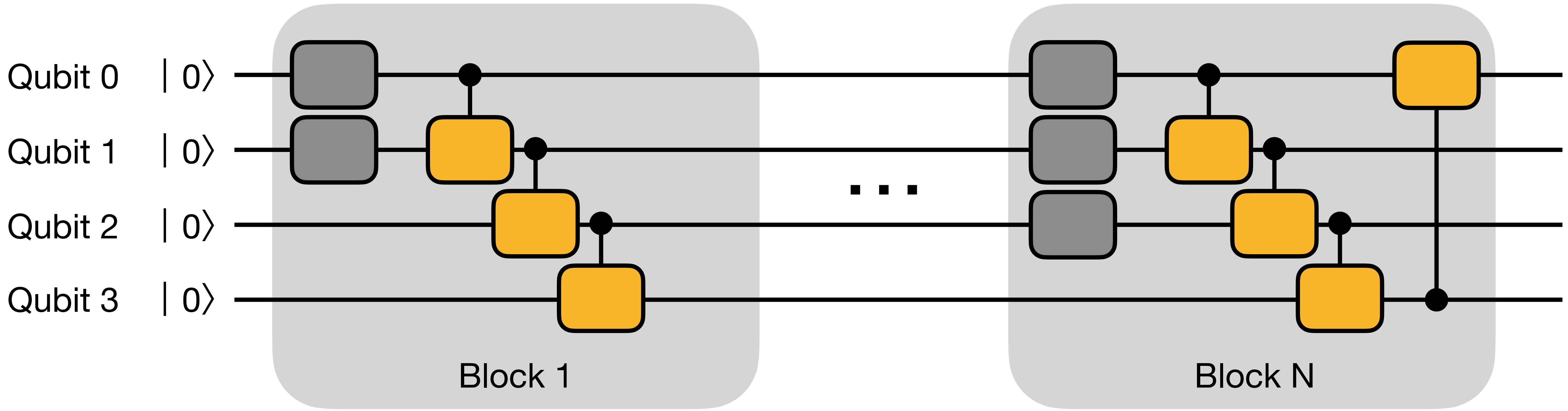
- SuperCircuit Construction and Training
- Noise-Adaptive Evolutionary Co-Search of SubCircuit and Qubit Mapping
- Train the Searched SubCircuit
- Iterative Quantum Gate Pruning

QuantumNAS

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

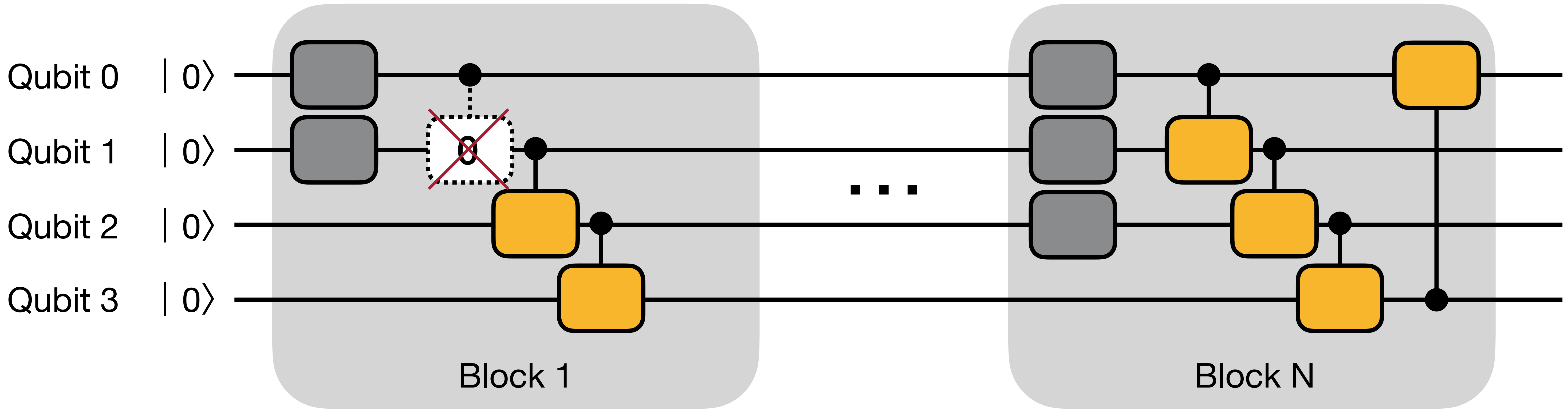
Iterative Pruning

- 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



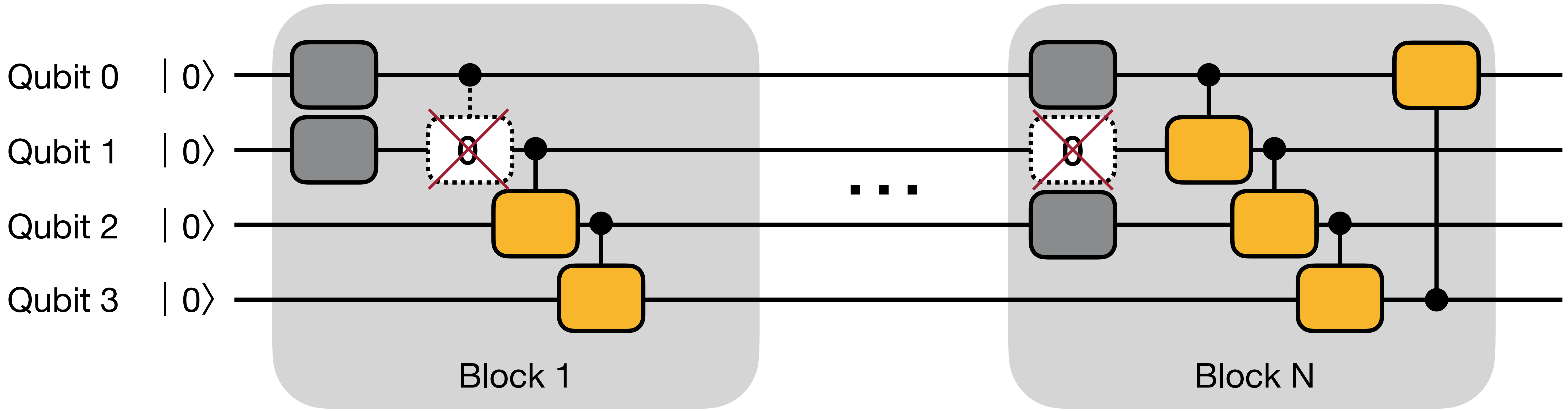
Iterative Pruning

- 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



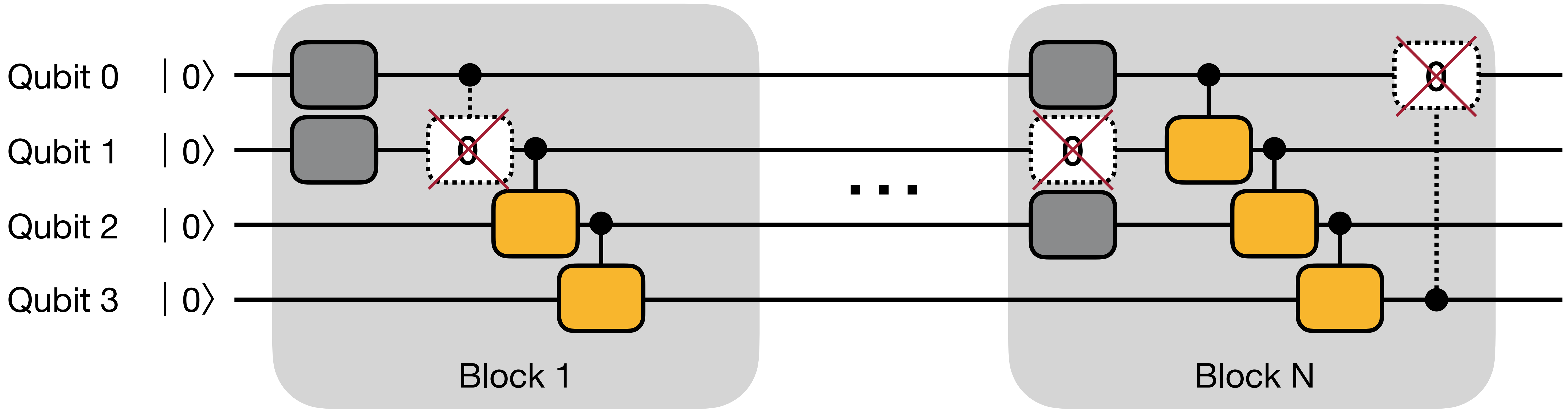
Iterative Pruning

- 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



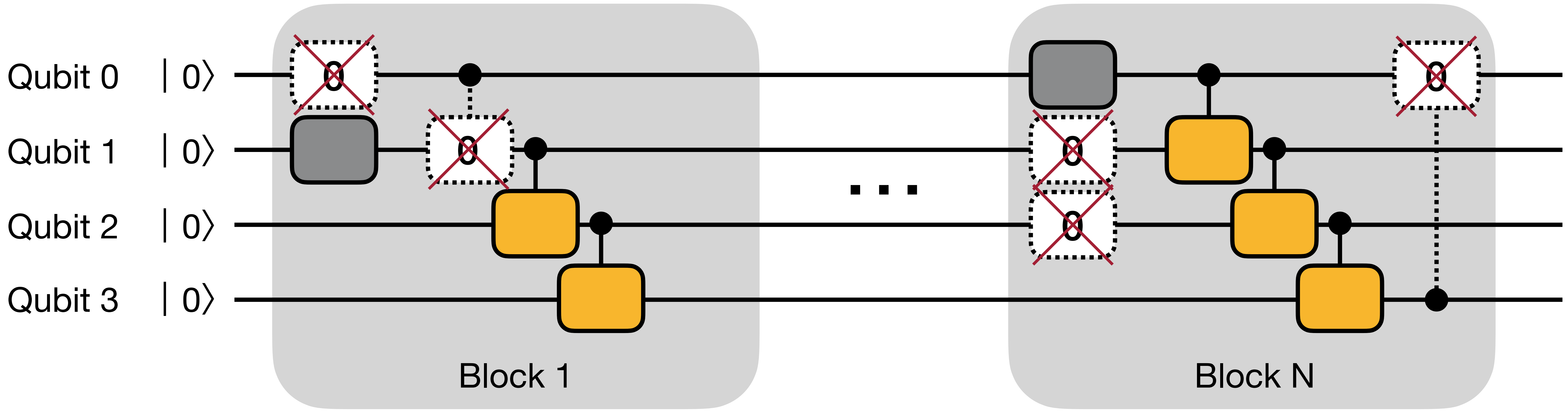
Iterative Pruning

- 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



Outline

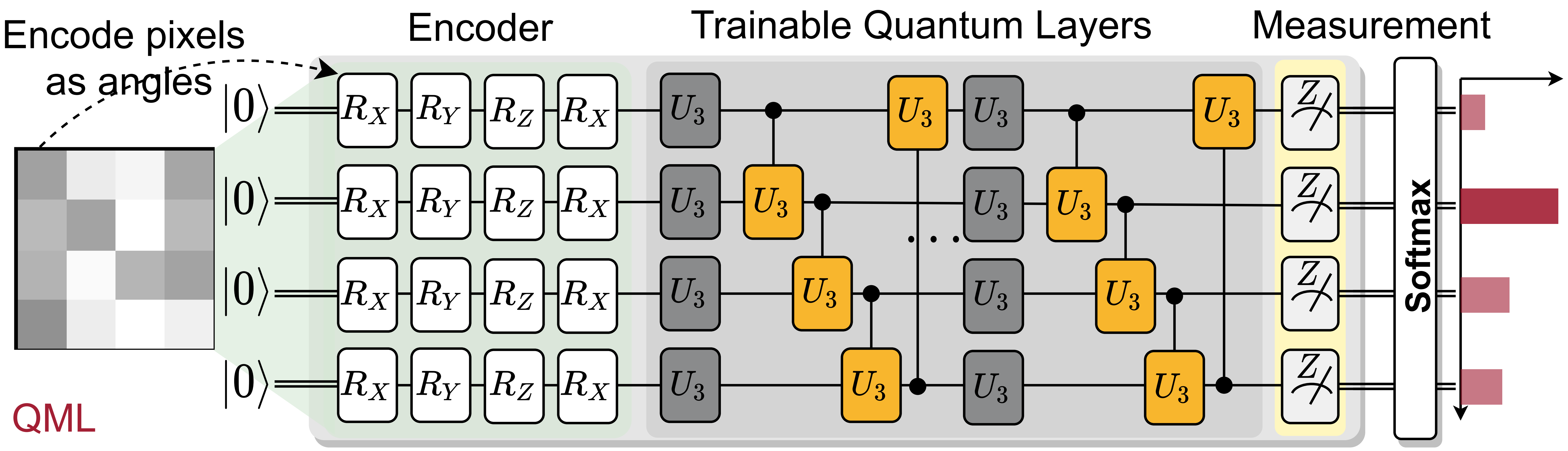
- Overview
- Background
- QuantumNAS
- Evaluation
- TorchQuantum Library
- Conclusion

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

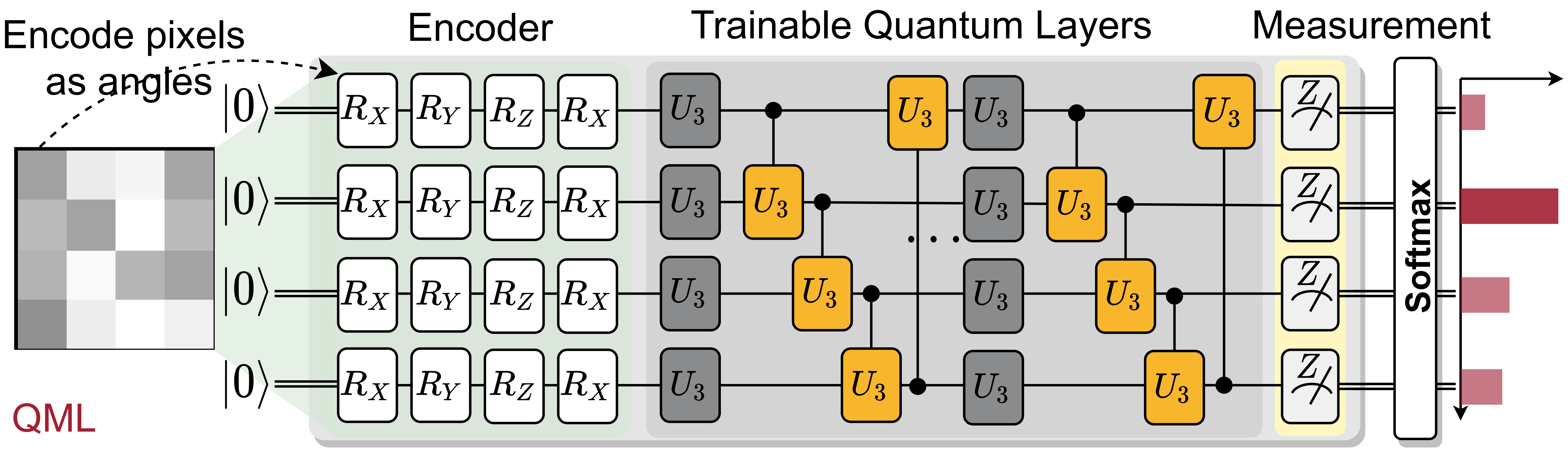
Benchmarks: QNN and VQE

- Quantum Neural Networks: classification

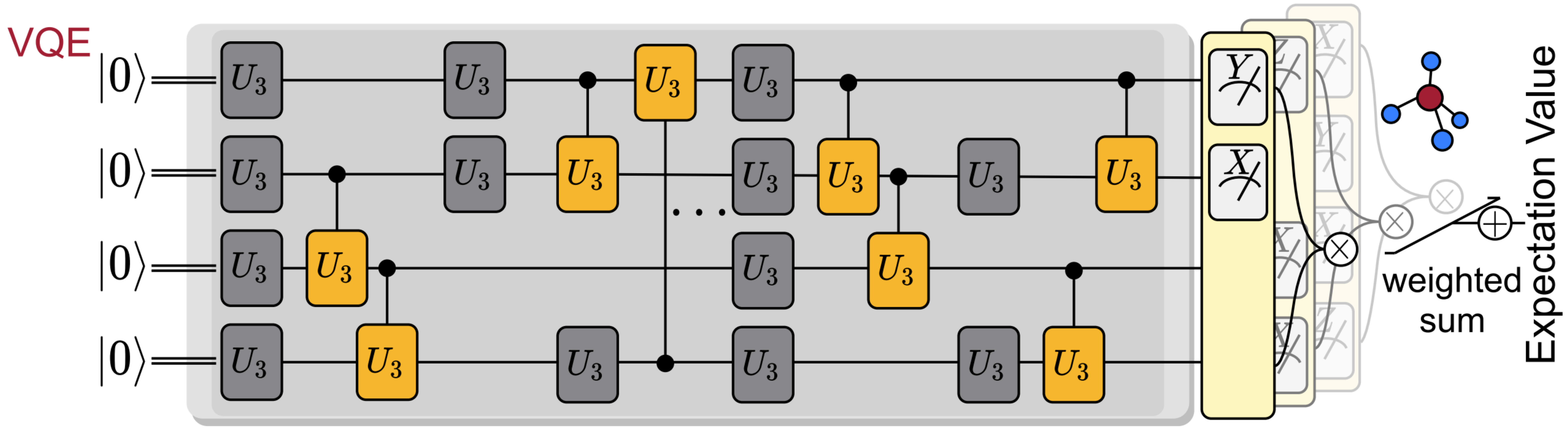


Benchmarks: QNN and VQE

- Quantum Neural Networks: classification

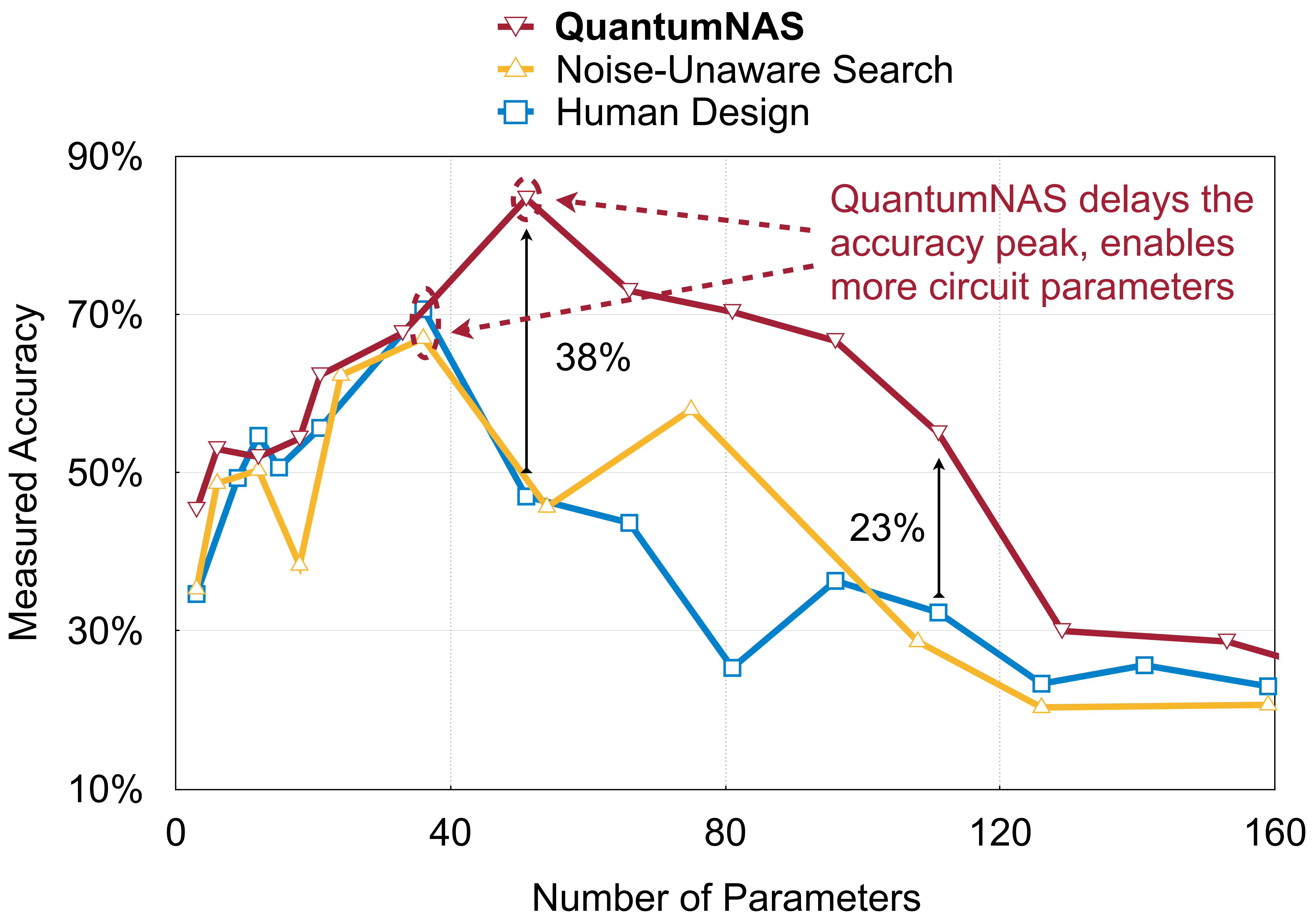


- Variational Quantum Eigensolver: finds the ground state energy of molecule Hamiltonian



QML Results

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



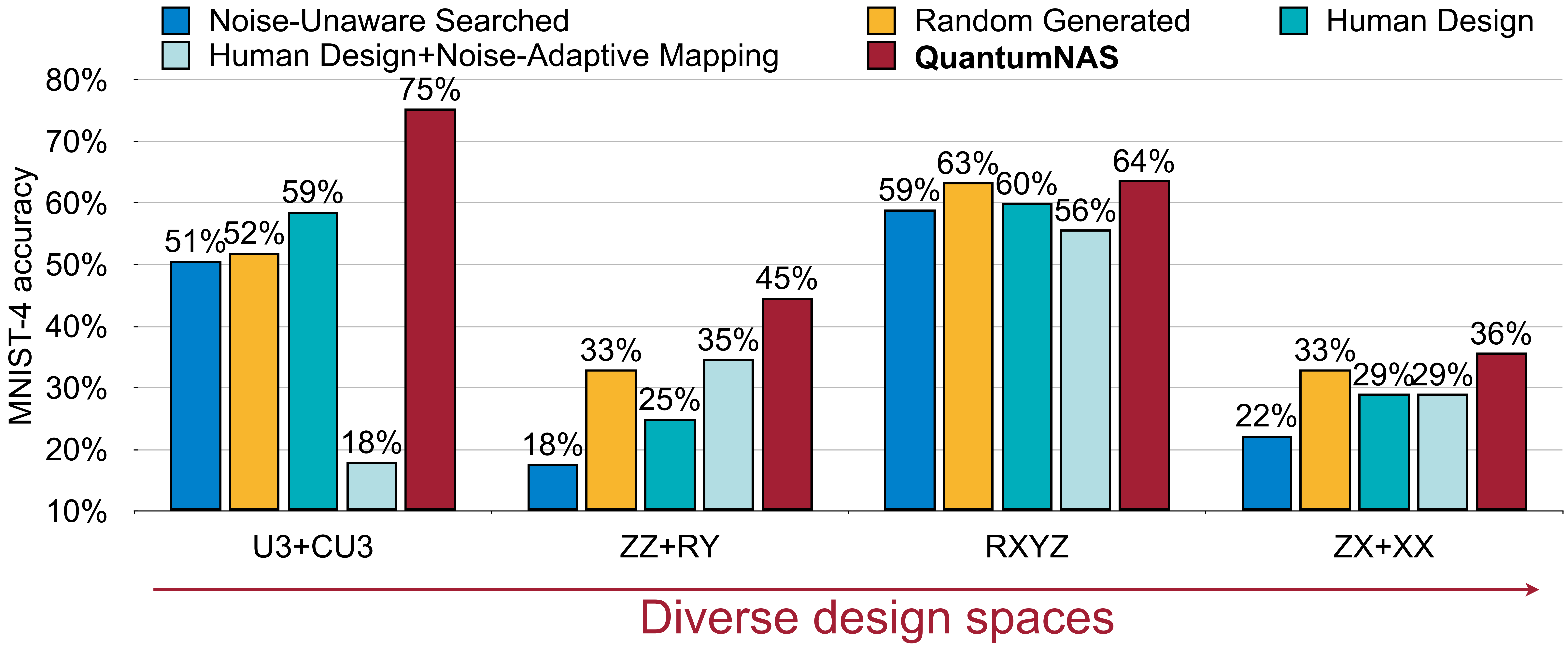
Analysis of Searched Circuit

- QuantumNAS searched circuit has fewer #gates and #params but higher accuracy

MNIST-2	Depth	#Gates	#Params	**QuantumNAS**
Human Design	64	135	36	**88%**
QuantumNAS	70	116	22	**92%**

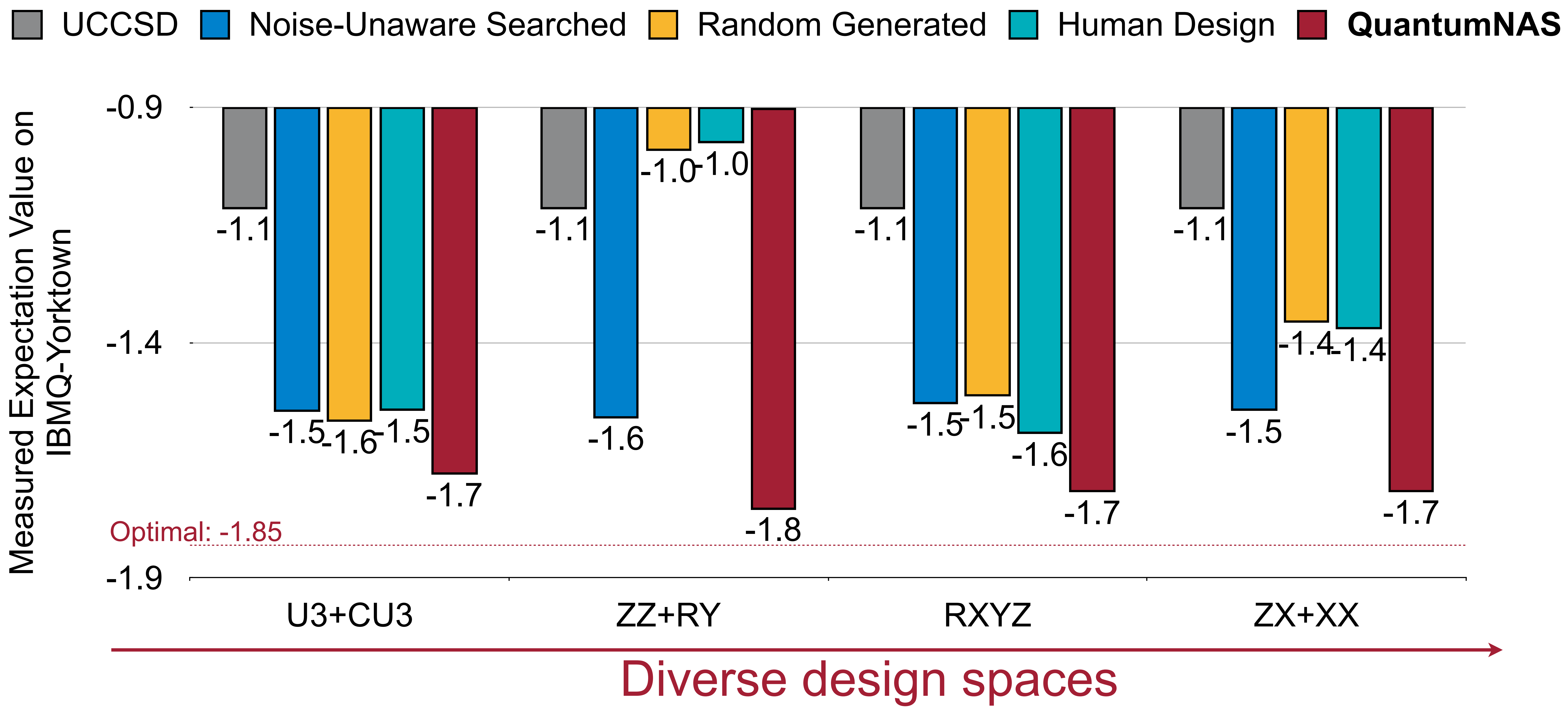
Consistent Improvements on Diverse Design Spaces

- ‘U3+CU3’, ’ZZ+RY’, ‘RXYZ’, ‘ZX+XX’ spaces with different gates



Consistent Improvements on Diverse Design Spaces

- H2 in different design spaces on IBMQ-Yorktown



Consistent Improvements on Diverse Devices

- QuantumNAS is effective for different real quantum devices
- On different 5-Qubit devices
- MNIST-4, Fashion-4, Vowel-4, MNIST-2, Fashion-2 averaged accuracy

Diverse devices

Method	Noise-Unaware Searched	Random	Human	**QuantumNAS**
Belem (5Q, 16QV)	47%	50%	67%	**76%**
Quito (5Q, 16QV)	73%	68%	74%	**79%**
Athens (5Q, 32QV)	50%	63%	68%	**77%**
Santiago (5Q, 32QV)	74%	73%	75%	**80%**

Scalable to Large #Qubits

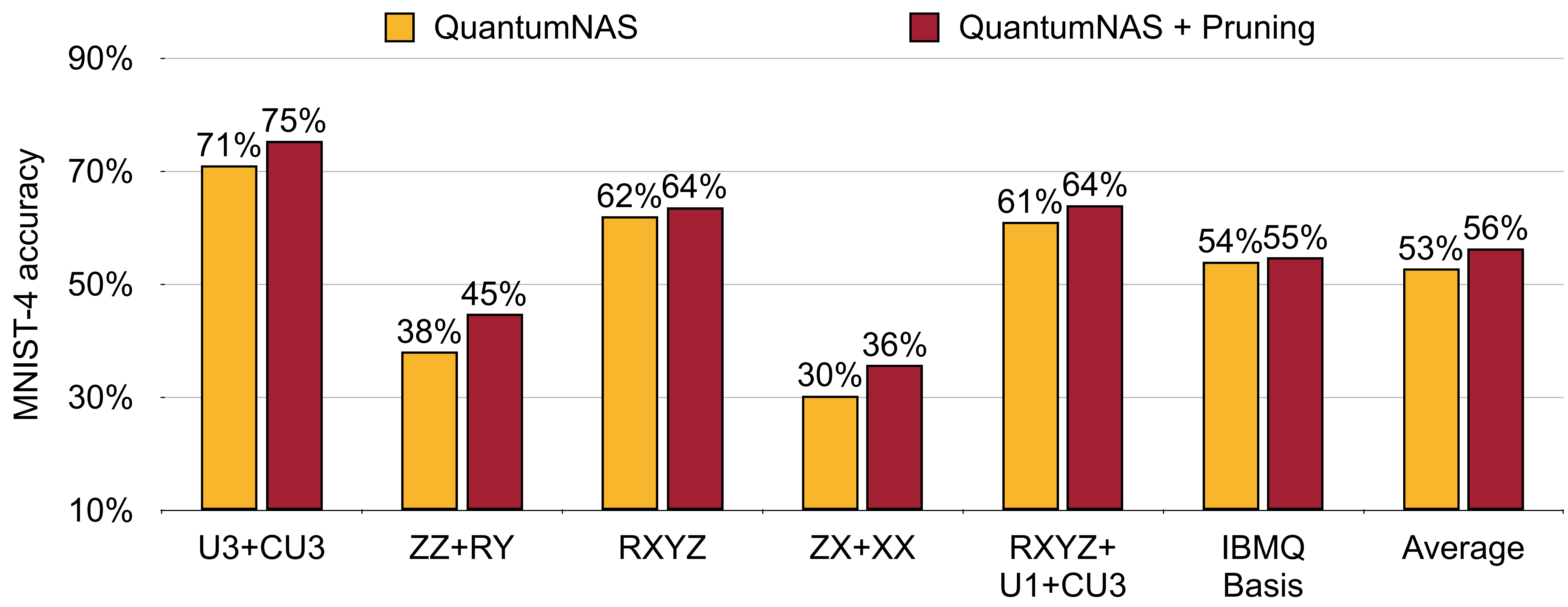
- On large devices
- MNIST-10 accuracy

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%**

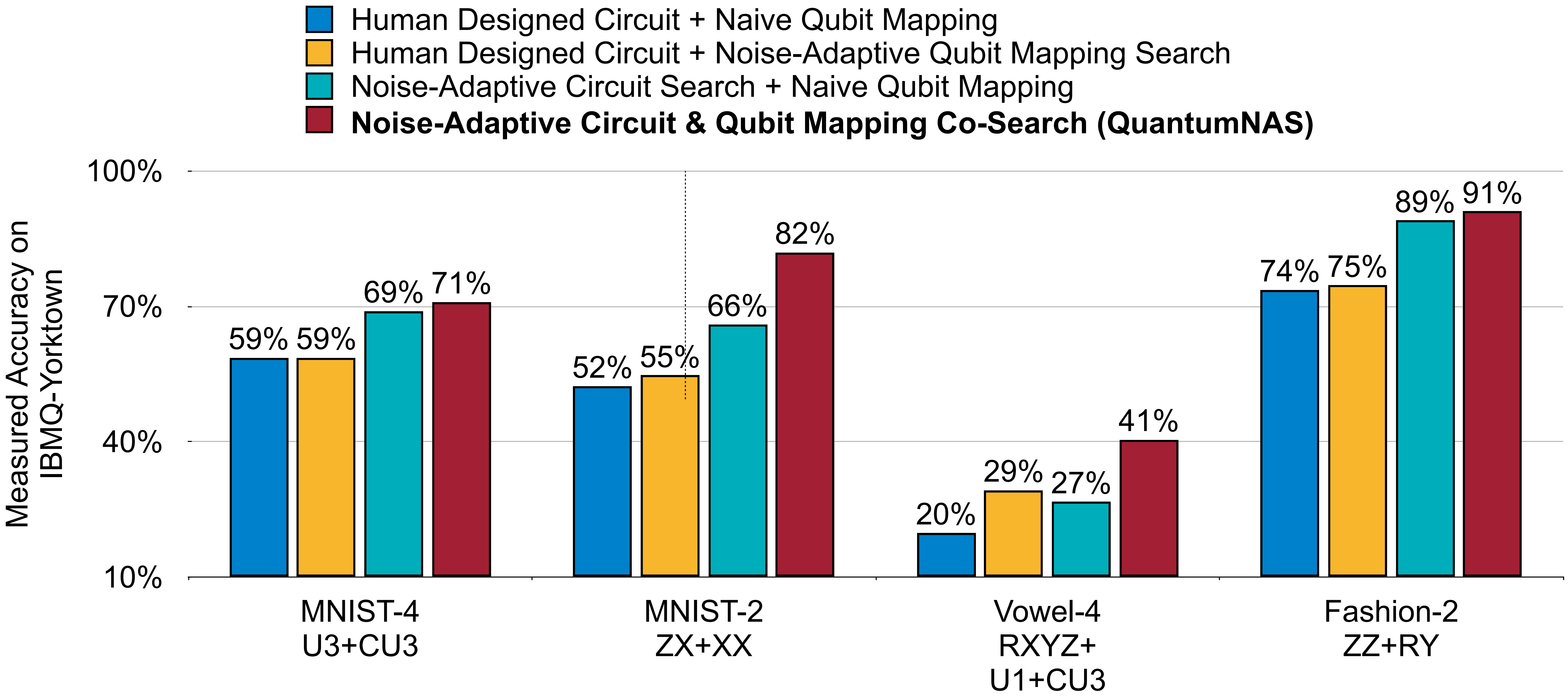
Effective of Quantum Gate Pruning

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



Effect of Circuit & Qubit Mapping Co-Search

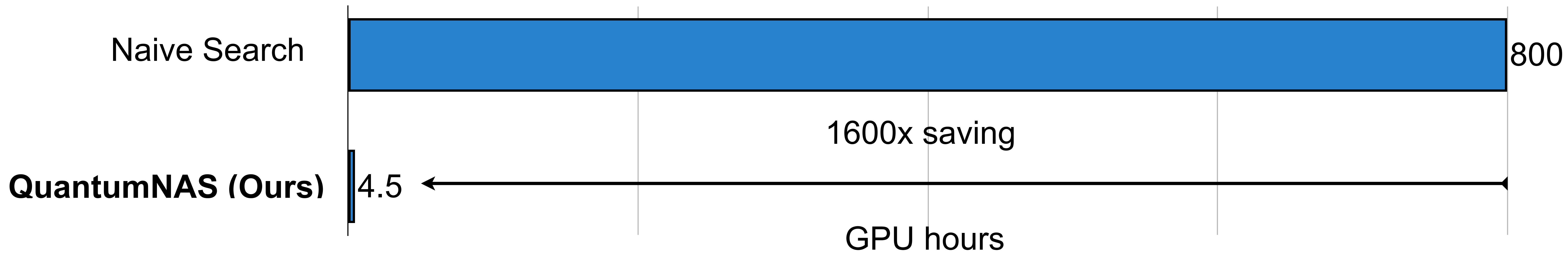
- Co-search is better than only search architecture/mapping



Time Cost

- On 1 Nvidia Titan RTX 2080 ti GPU

Step / #qubits	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

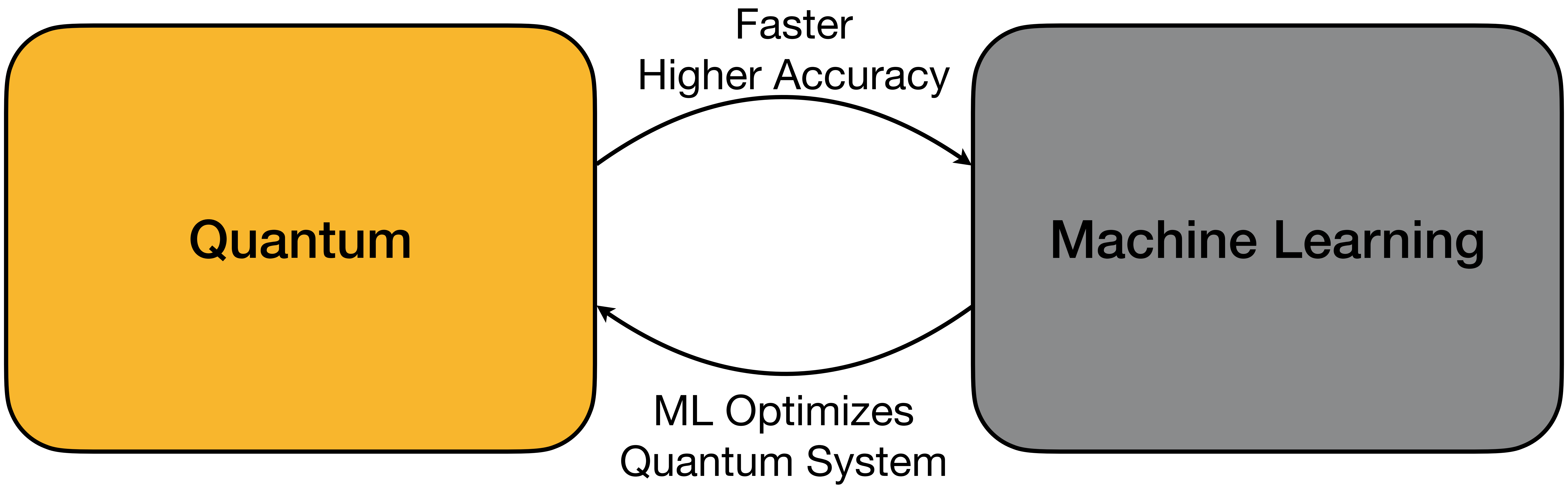


Outline

- Quick Overview
- Background
- QuantumNAS
- Evaluation
- TorchQuantum Open-source
- Conclusion

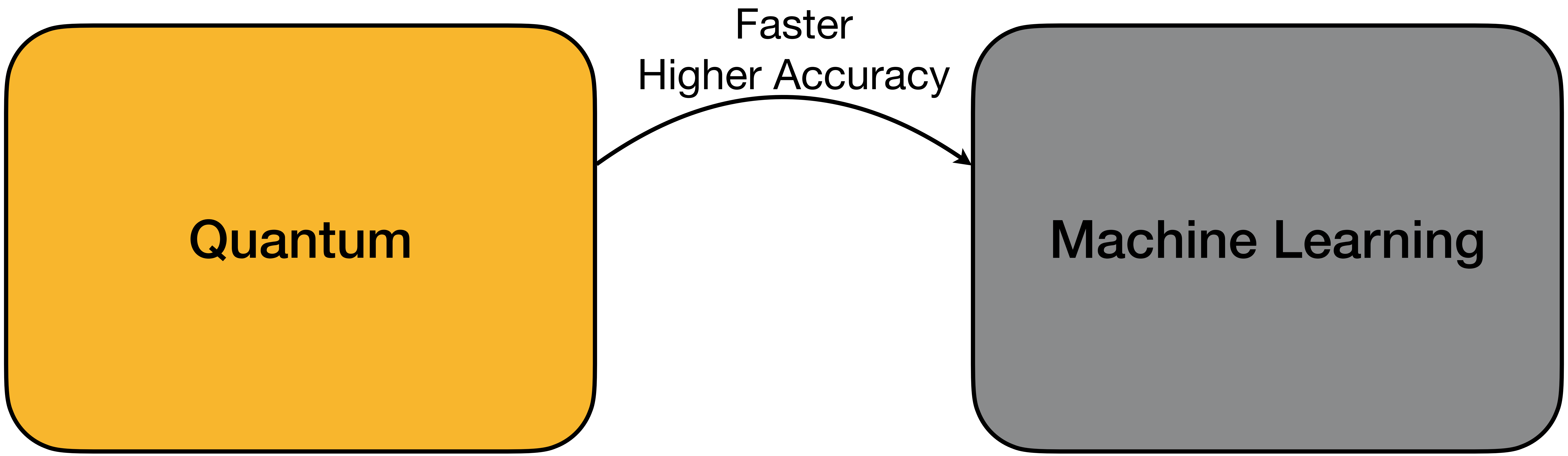
Open-source: TorchQuantum

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



Open-source: TorchQuantum

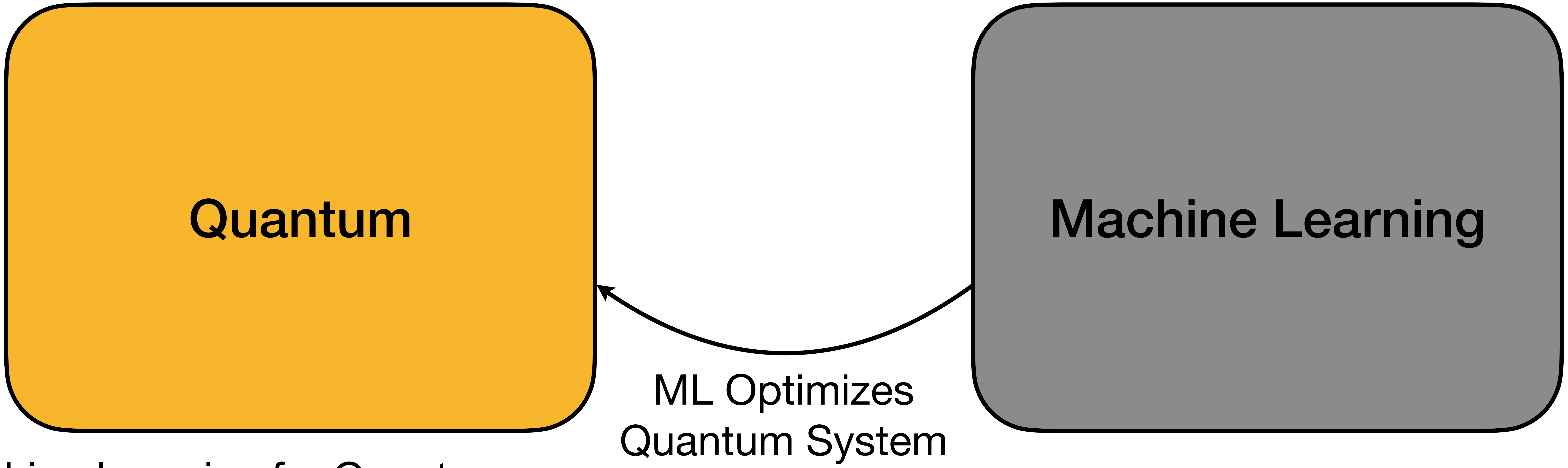
- TorchQuantum — An open-source library for interdisciplinary research of quantum computing and machine learning
- https://github.com/mit-han-lab/torchquantum



- 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



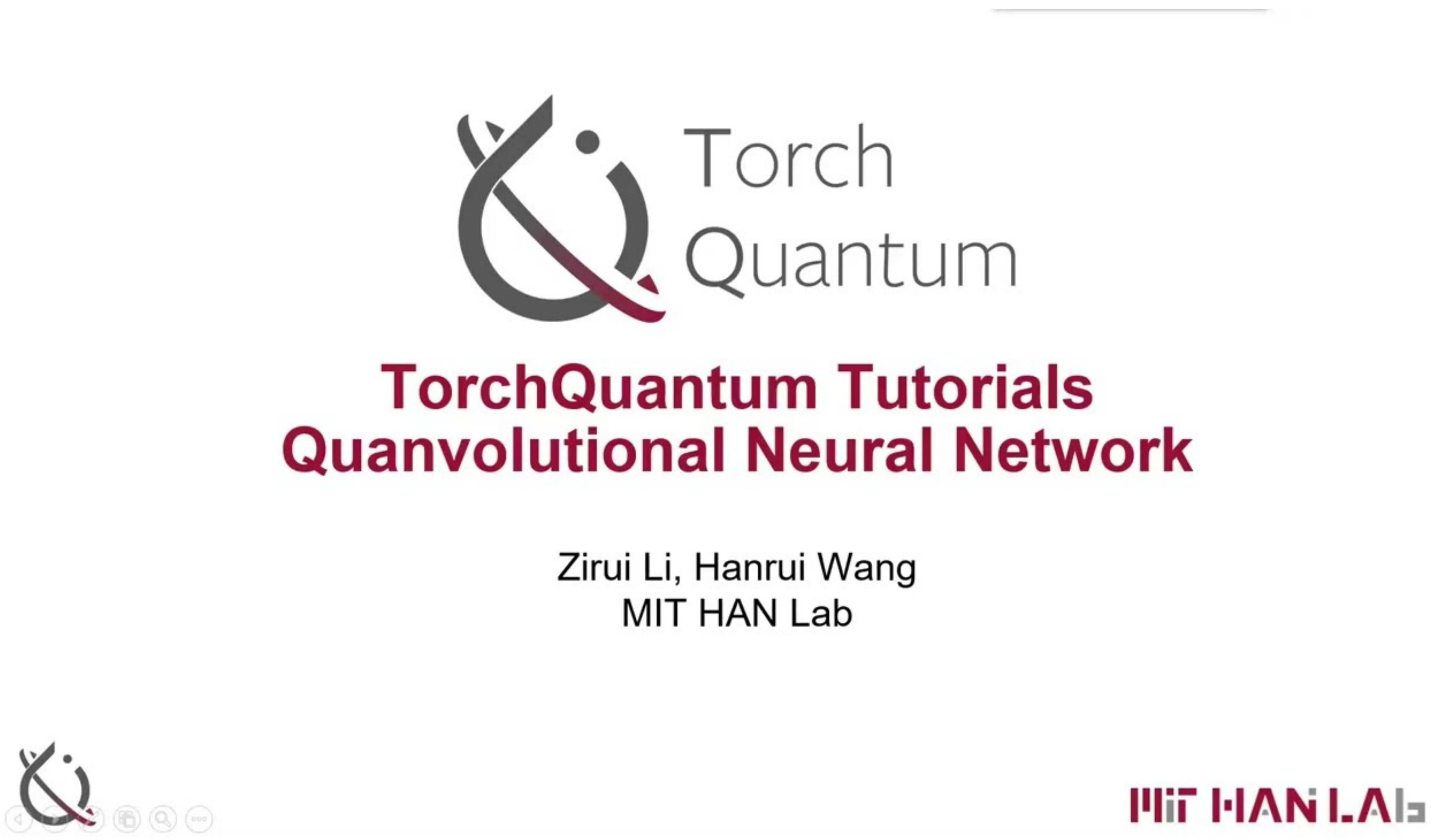
- Machine Learning for Quantum
  - ML for quantum compilation (qubit mapping, unitary synthesis)
  - …

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

Examples and tutorials

- Tutorial Colab and videos



Outline

- Overview
- Background
- QuantumNAS
- Evaluation
- TorchQuantum Library
- Conclusion

Conclusion

- **QuantumNAS** exploits **SuperCircuit-**based co-search for most **noise-robust** circuit architecture and qubit mapping
- Iterative **quantum gate pruning** to further remove redundant gates
- Improves MNIST 2-class accuracy from 88% to **95%**,10-class from 15% to **32%**
- Save search cost by over **1,000 times**
- Open-sourced **TorchQuantum** library for Quantum + ML research

Torch
Quantum

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

qmlsys.mit.edu

qmlsys.mit.edu