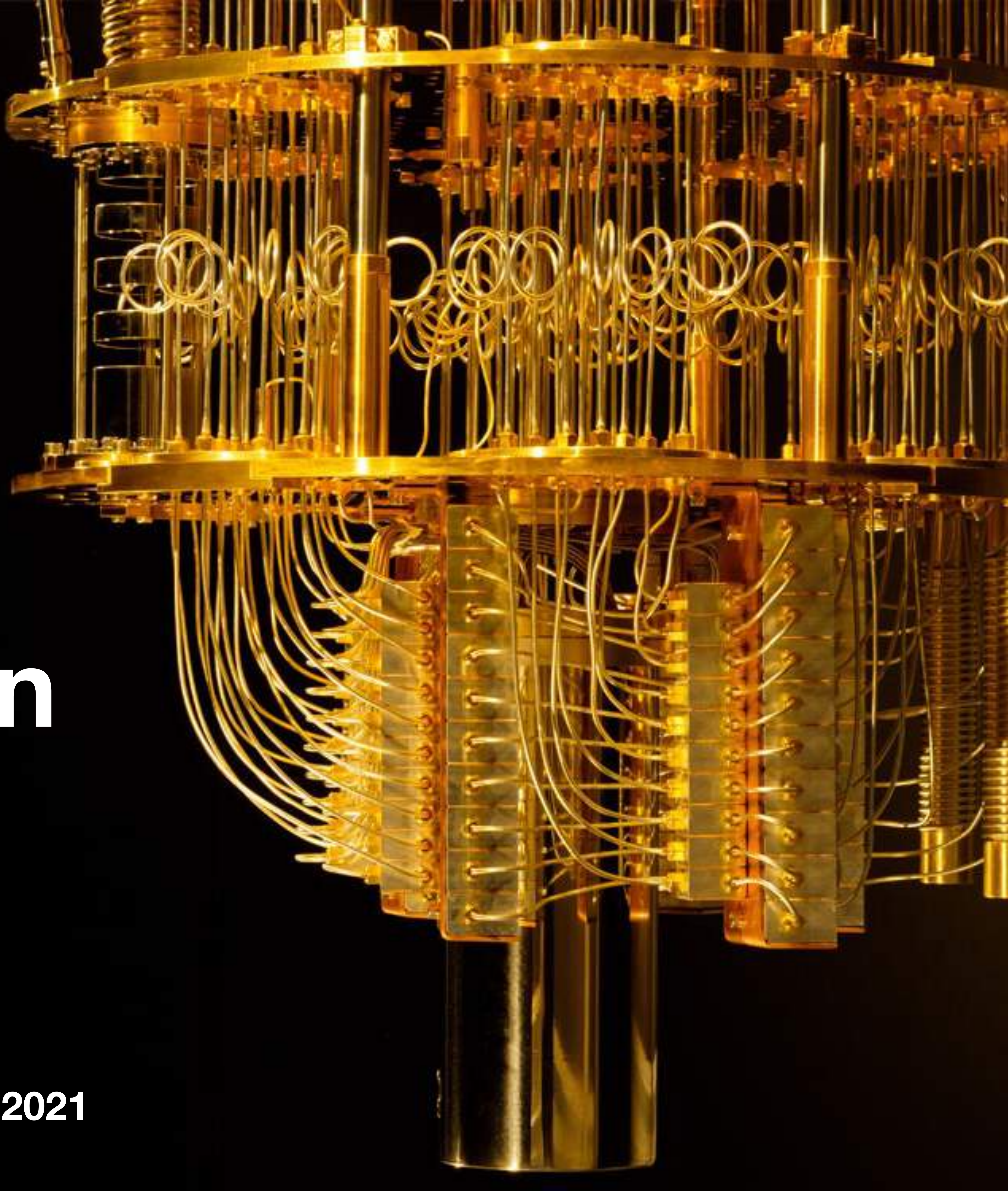
IBM Quantum devices for research in quantum information



Matteo Rossi
University of Turku (FI)

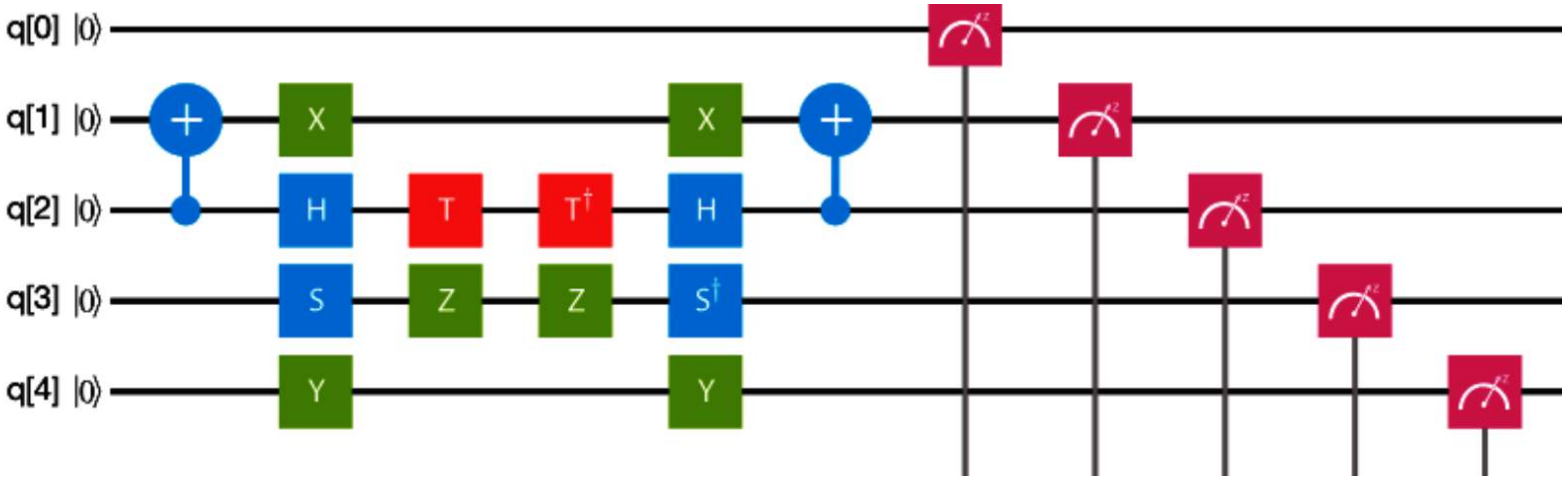
Milano
16 march 2021

Overview

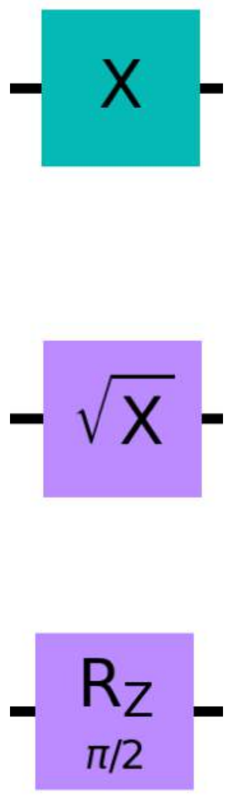
- Intro: Gate-based quantum computers and IBM Q
- Tutorial: Typical usage of IBM Q
- Applications:
  - Simulating open quantum systems (collisional models)
  - POVMS (in near-term algorithms)

Introduction

Gate-based quantum computers



A set of **basis gates** allows for **universal quantum computation.**

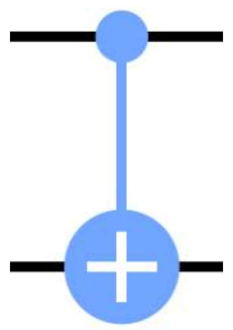


$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\frac{1}{2}\begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}$$

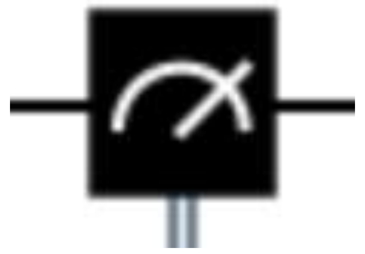
$$\begin{pmatrix} e^{-i\frac{\lambda}{2}} & 0 \\ 0 & e^{i\frac{\lambda}{2}} \end{pmatrix}$$

CNOT



$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Projective measurement

ρ  $0,1$

Gate-based quantum computers

Superconducting circuits

Google rigetti IBM Q™

Trapped ions

IONQ Honeywell

Photonic

XANADU

Cloud providers

aws Microsoft Azure

IBM Q Experience

Now IBM Quantum

- Launched in 2016 (a 5-qubit device controlled from the browser)
- Now counts 25 quantum devices from 1 to 65 qubits
- Devices are accessible from the cloud, using the browser or the Qiskit SDK
- Free registration gives immediate access to 8 devices and simulators
- Universities can join the IBM Q Network and get priority access.
- With research proposals (and lots of bureaucracy) access to restricted devices

Quantum services

View the status and details of IBM Quantum's systems and simulators, and track which are available to you.

How to cite

Systems

ibmq_casablanca
System status: Online
Processor type: Falcon r4
7 Qubits 32 Quantum volume

ibmq_bogota
System status: Online
Processor type: Falcon r4
5 Qubits 32 Quantum volume

ibmq_santiago
System status: Online
Processor type: Falcon r4
5 Qubits 32 Quantum volume

ibmq_rome
System status: Paused - In use
Processor type: Falcon r4
5 Qubits 32 Quantum volume

ibmq_athens
System status: Online
Processor type: Falcon r4
5 Qubits 32 Quantum volume

ibmq_belem
System status: Online
Processor type: Falcon r4
5 Qubits 16 Quantum volume

ibmq_quito
System status: Online
Processor type: Falcon r4
5 Qubits 16 Quantum volume

ibmq_16_melbourne
System status: Online
Processor type: Canary r1.1
15 Qubits 8 Quantum volume

ibmq_lima
System status: Online
Processor type: Falcon r4
5 Qubits 8 Quantum volume

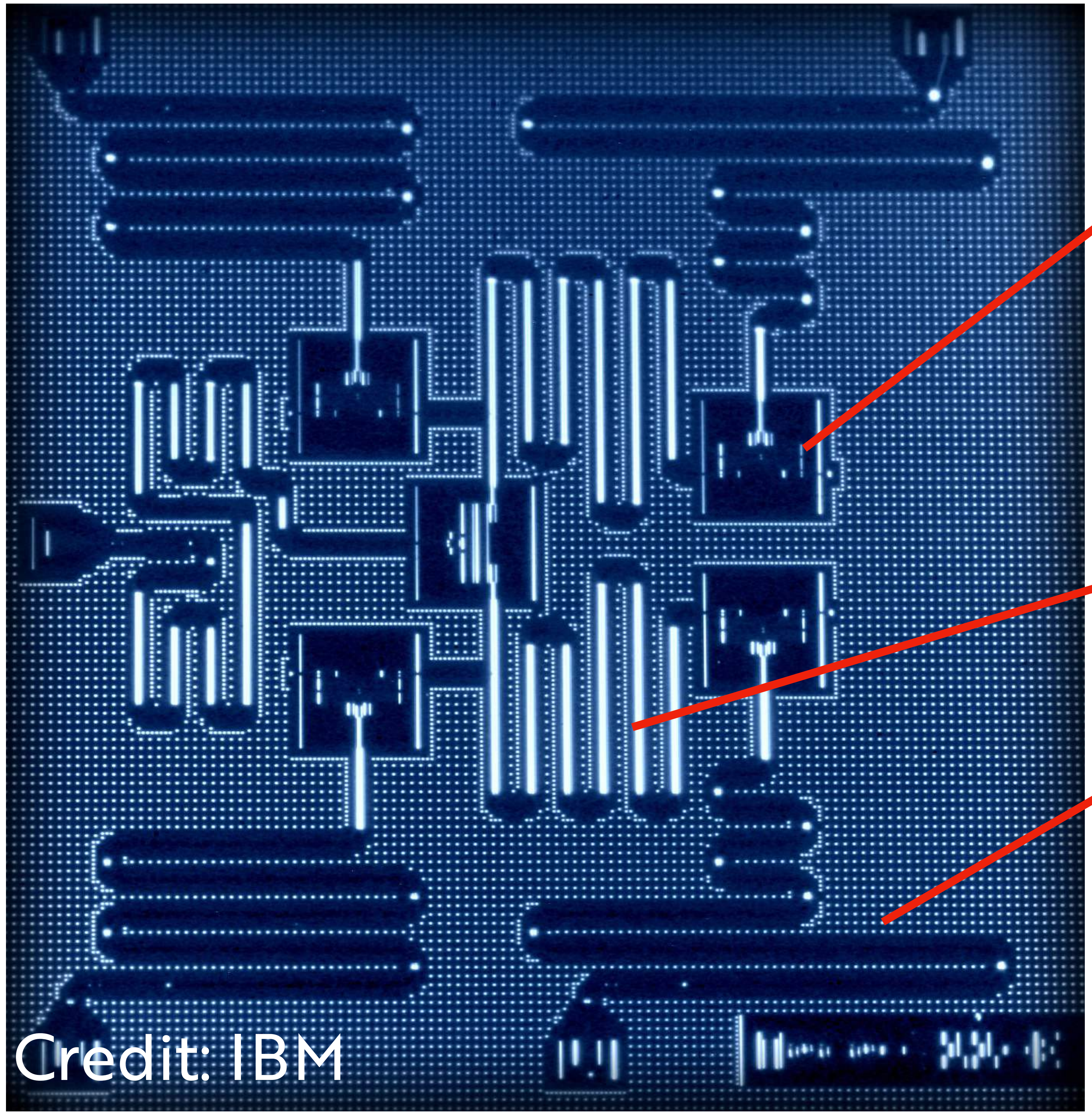
ibmq_5_yorktown
System status: Online
Processor type: Canary r1
5 Qubits 8 Quantum volume

ibmq_armonk
System status: Online
Processor type: Canary r1.2
1 Qubit 1 Quantum volume

Simulators

ibmq_qasm_simulator
Simulator status: Online
Simulator type: General, context-aware
32 Qubits

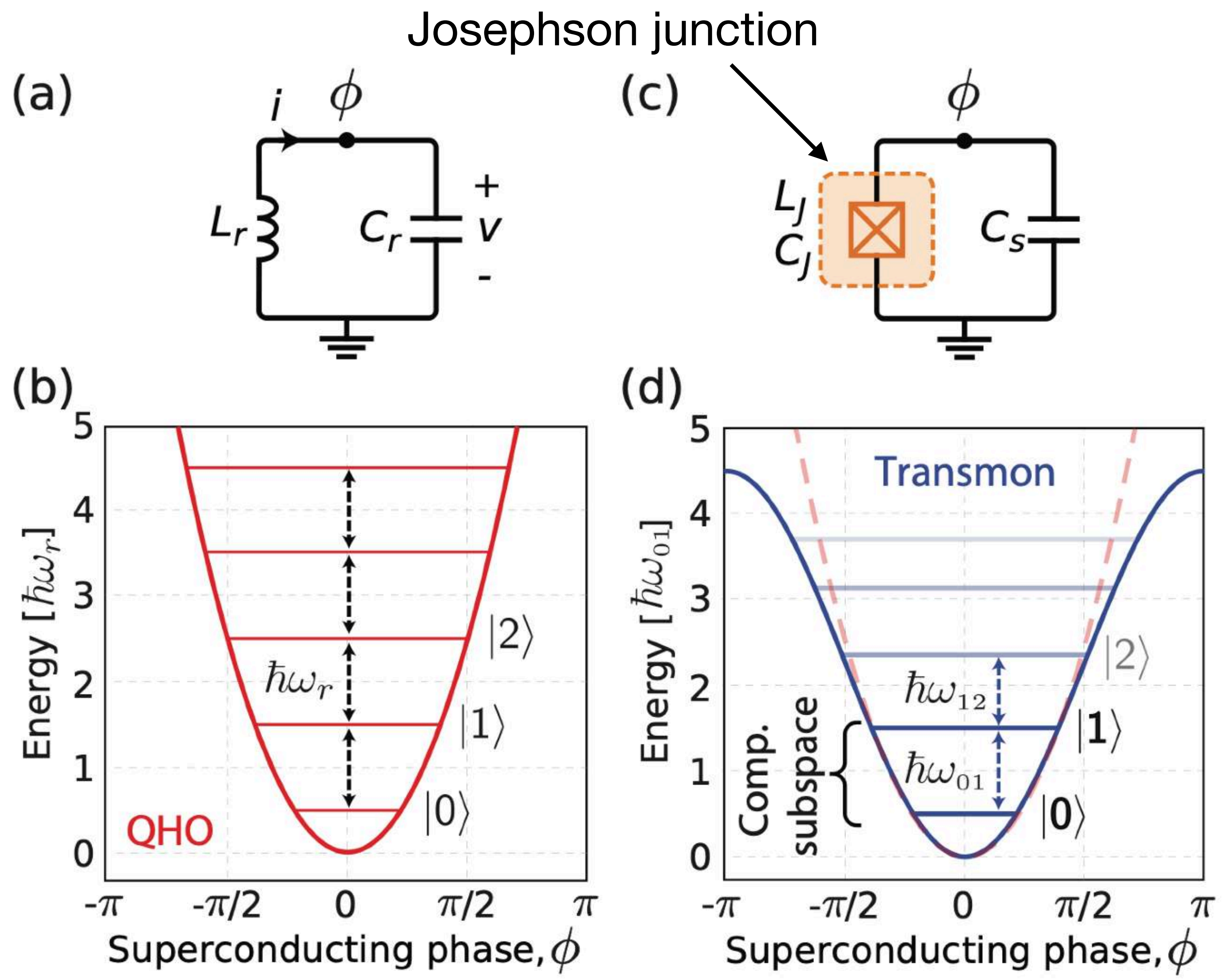
Superconducting qubits



Transmon qubits

Superconducting microwave resonators

- Single-qubit rotations
- Qubit-qubit bus
- Readout operations



arXiv:1904.06560

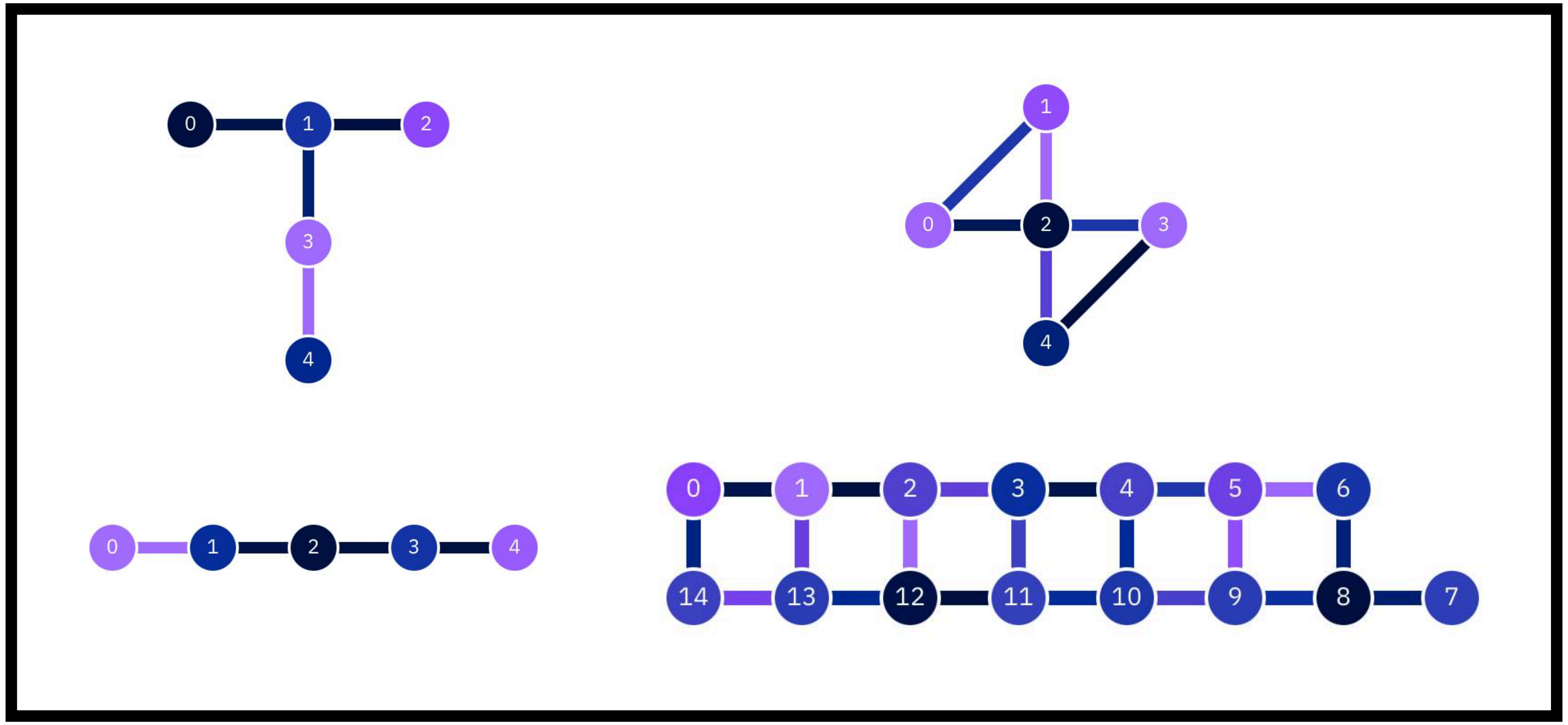
$\omega_{01} \sim 5\,\mathrm{GHz}$ ⟷ $240\,\mathrm{mK}$

All qubits have different frequencies

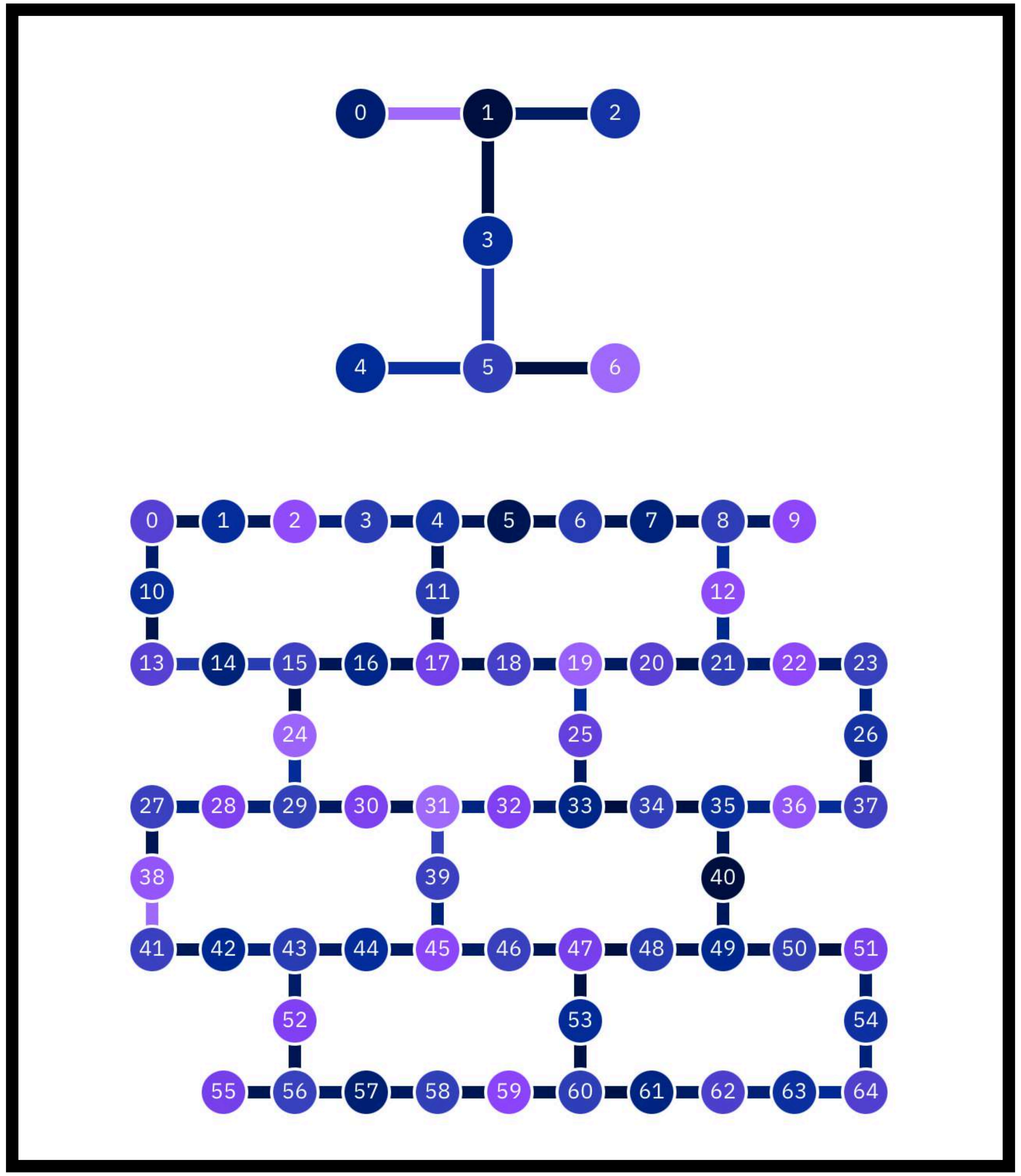
This allows to perform entangling gates using cross-resonance

Connectivity layout

Free

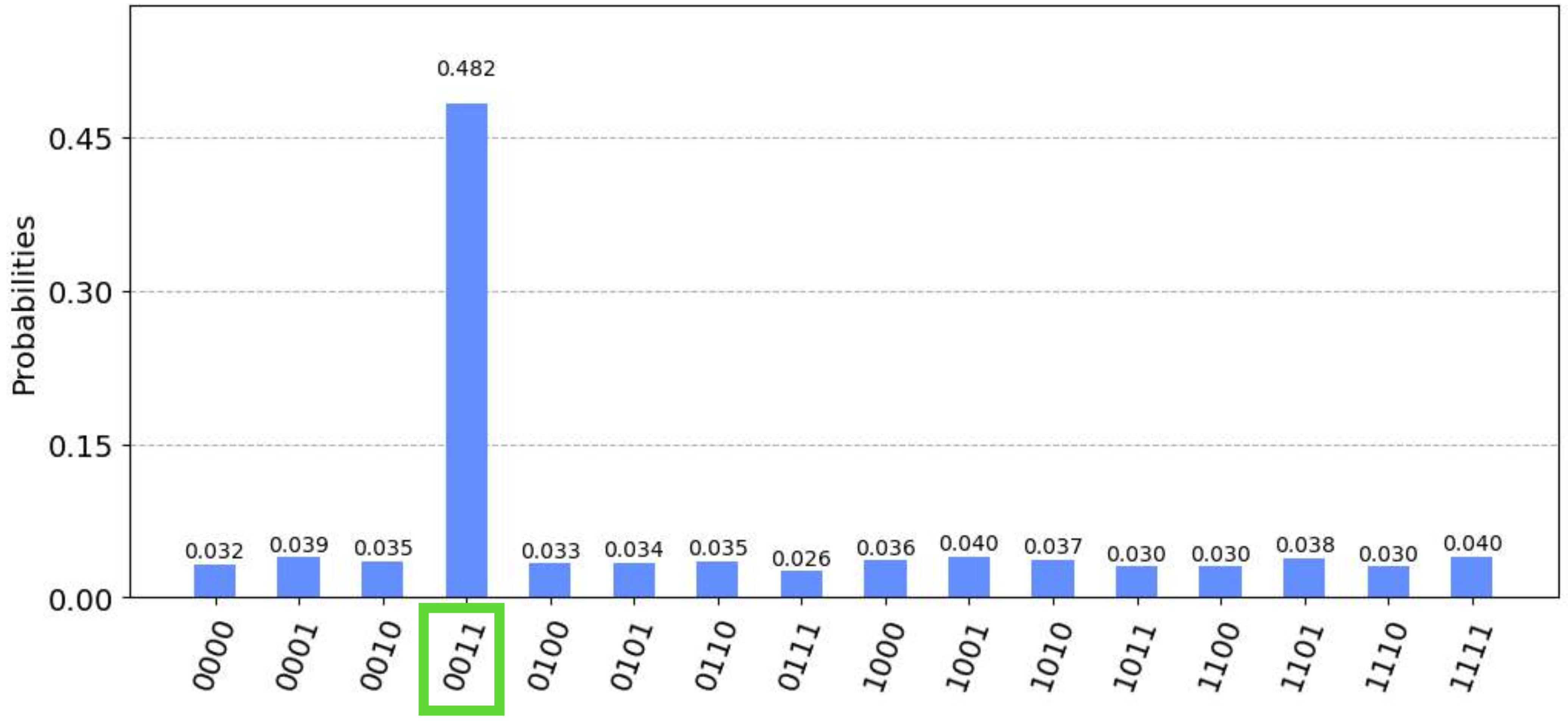


$$$ or research agreements

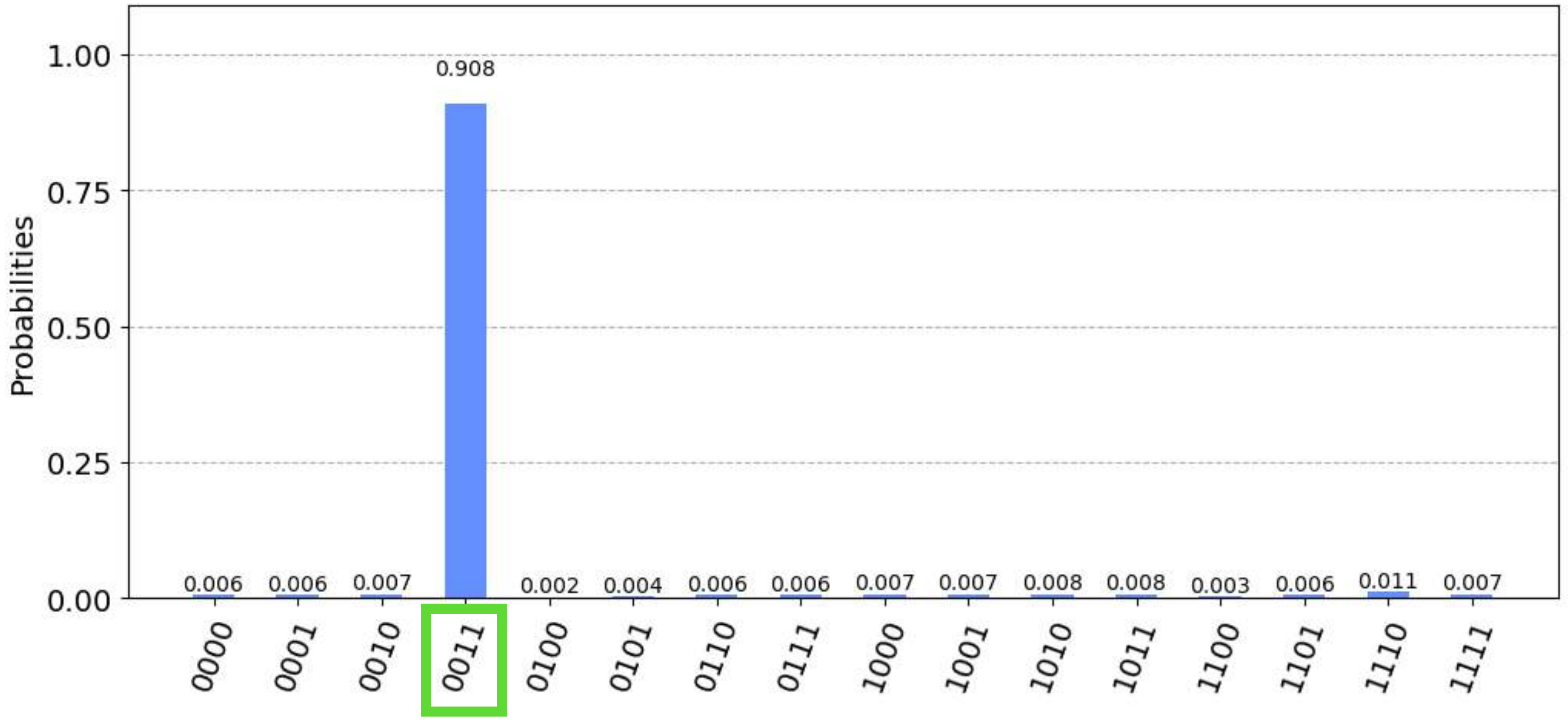


Grover search

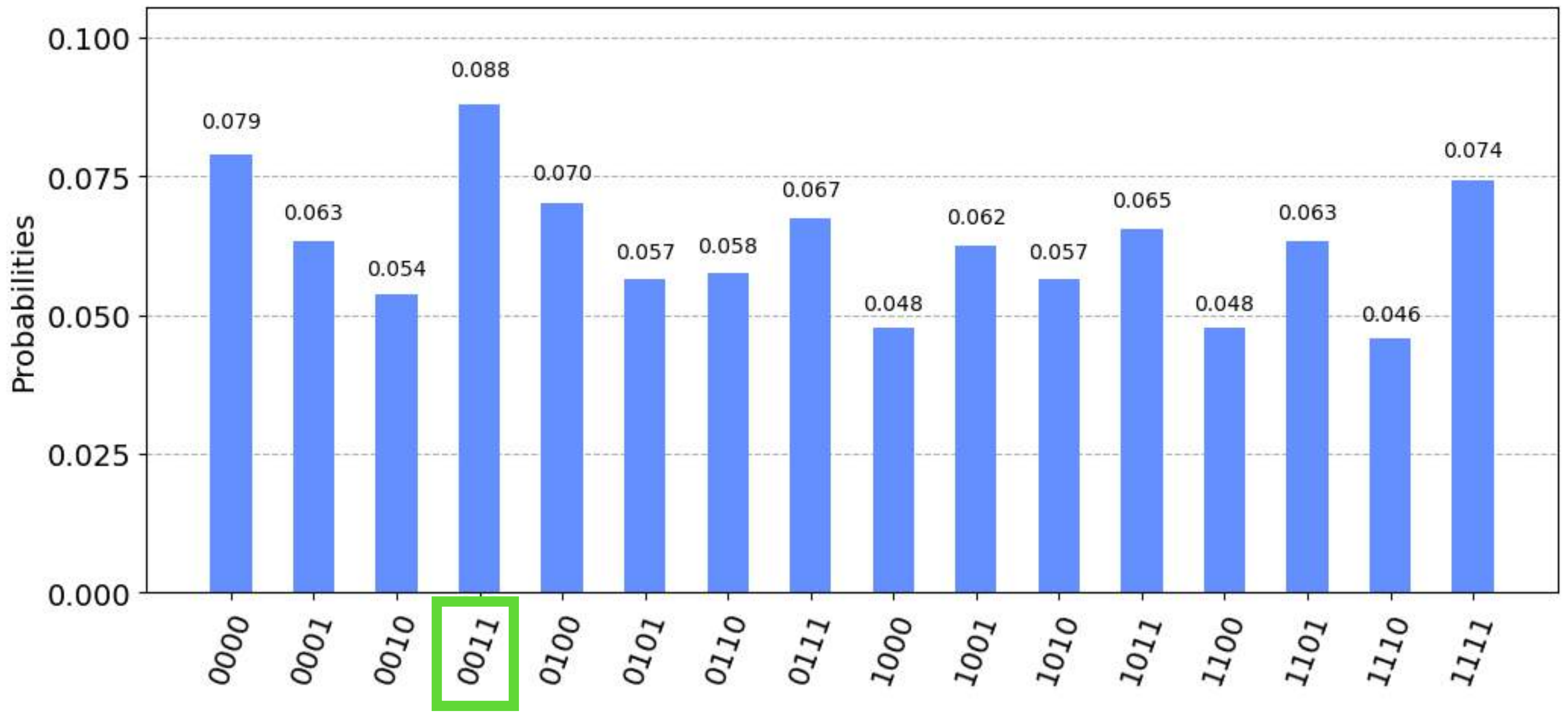
Simulation, 1 iteration



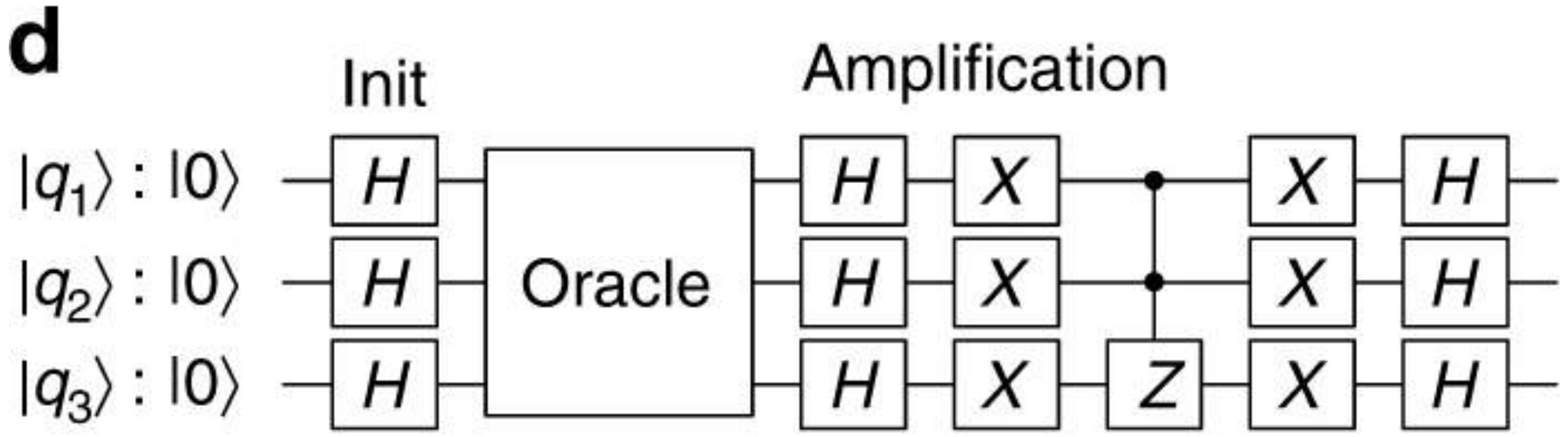
Simulation, 2 iterations



ibmq_bogota



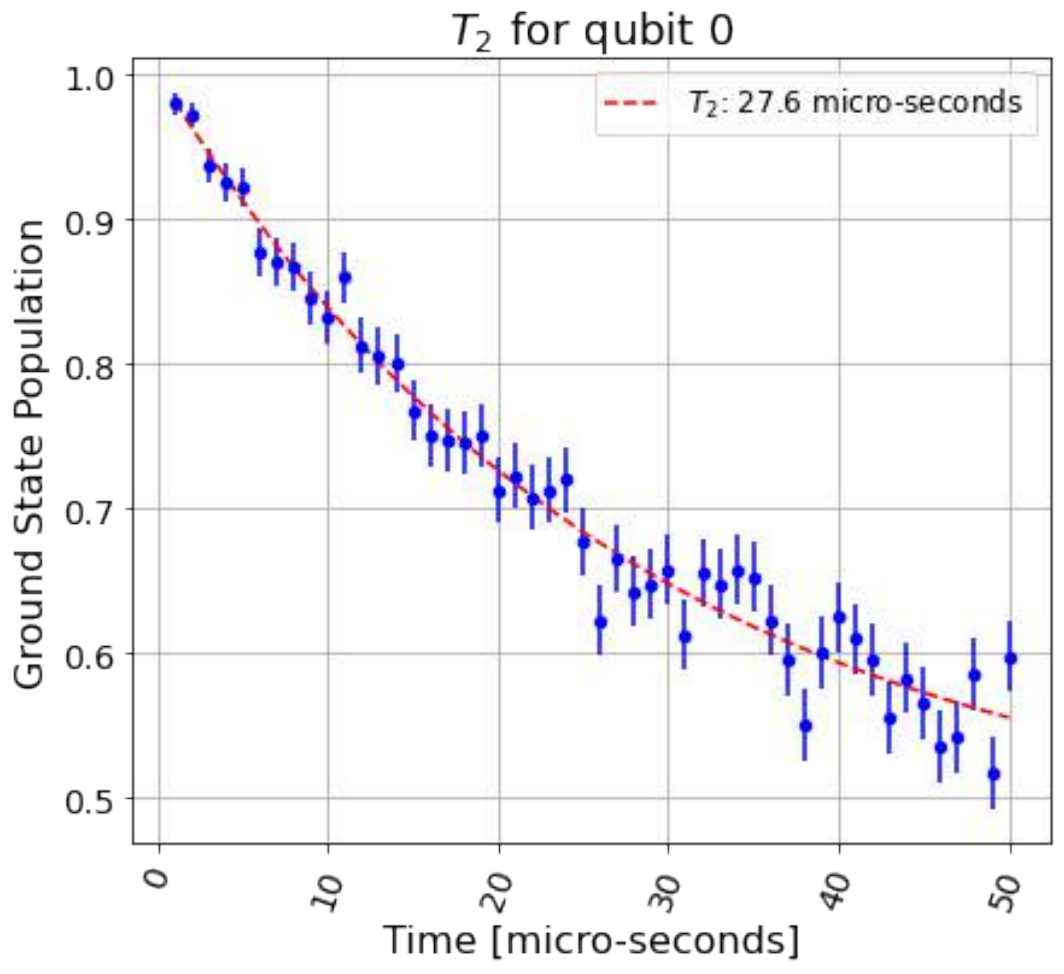
d

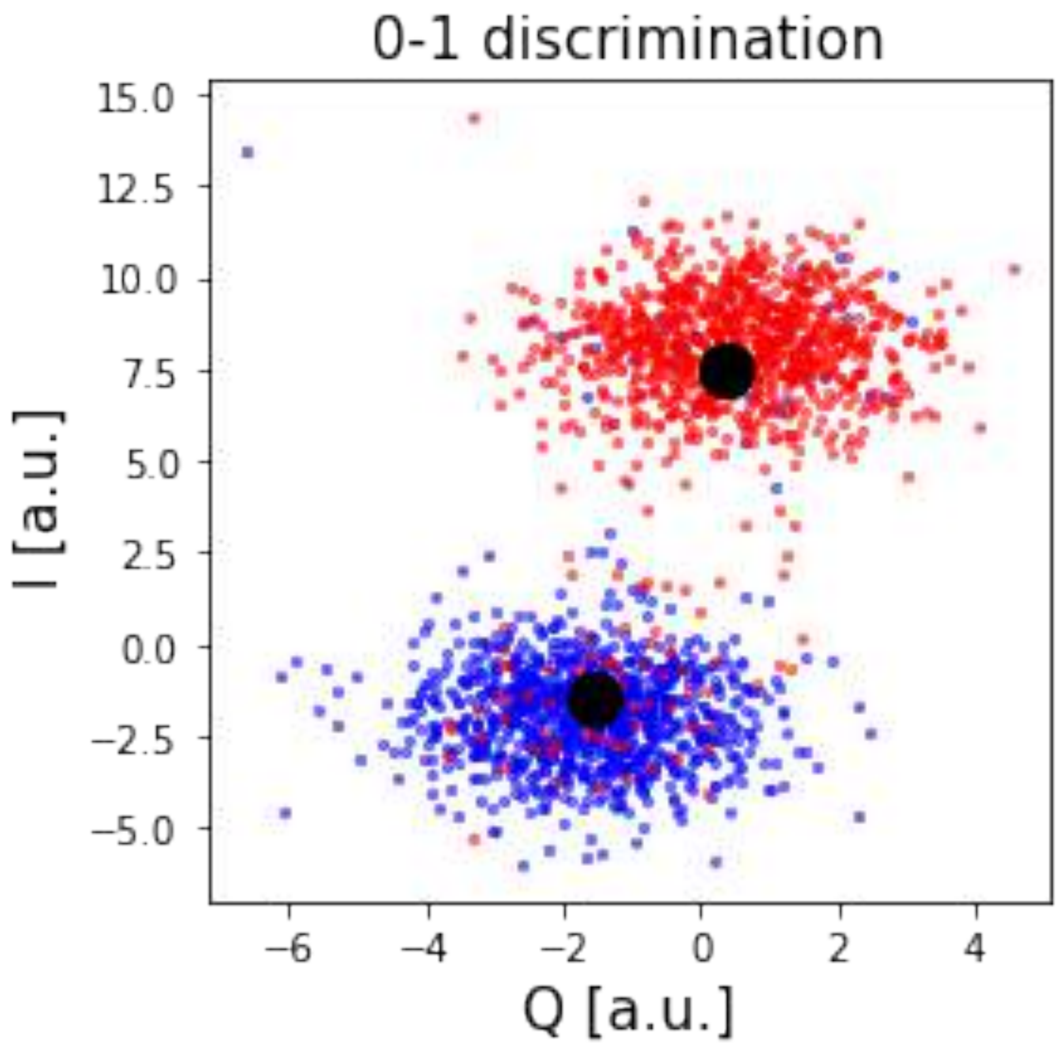


Noise

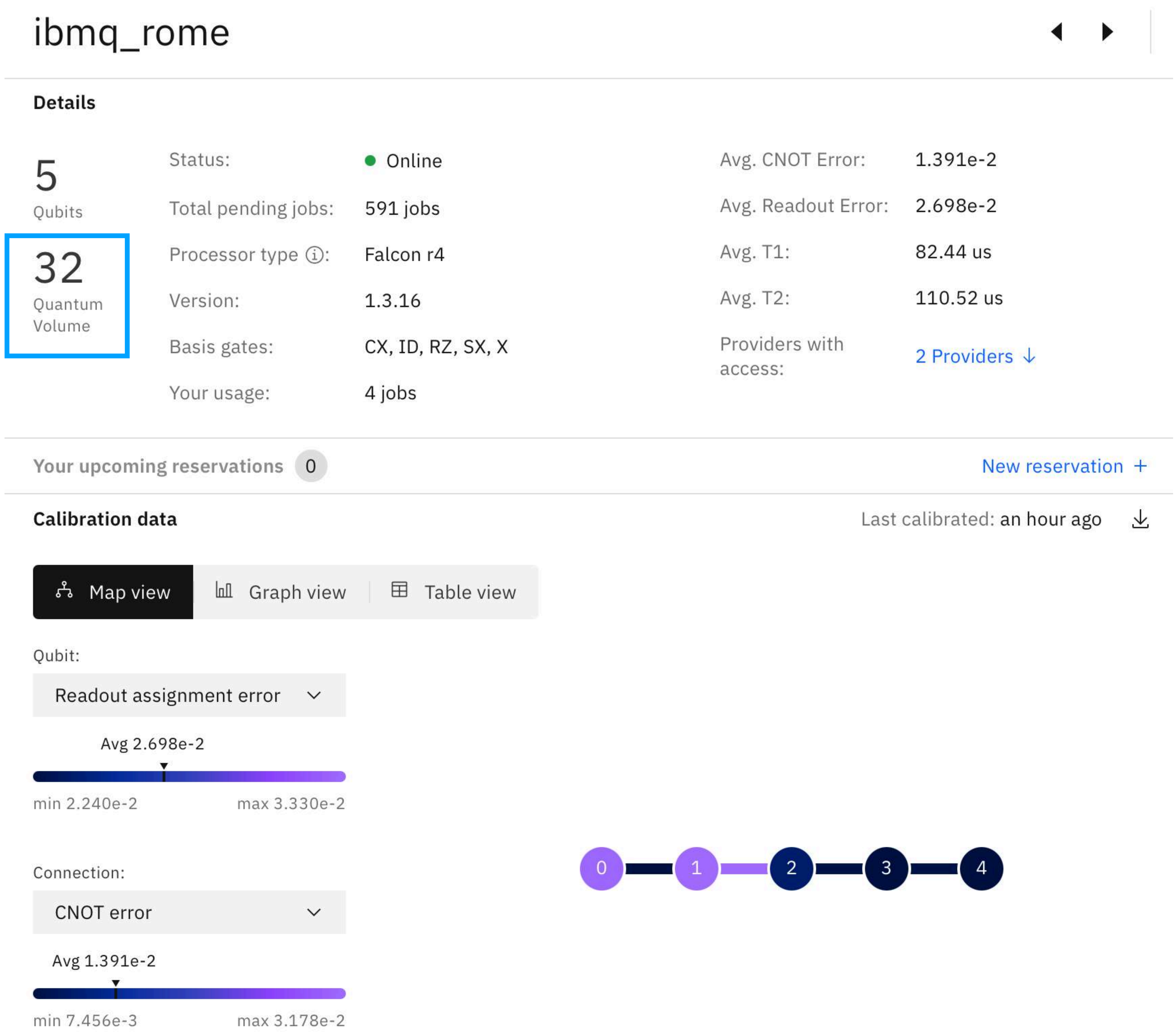
Quantum gate-based devices are characterized by various sources of noise

- Finite qubit coherence times T_1, T_2 (interaction with the environment)
- Gate error rates (imperfect pulses, interaction with env., crosstalk)
- Measurement errors (discrimination errors, imperfect pulses, crosstalk)



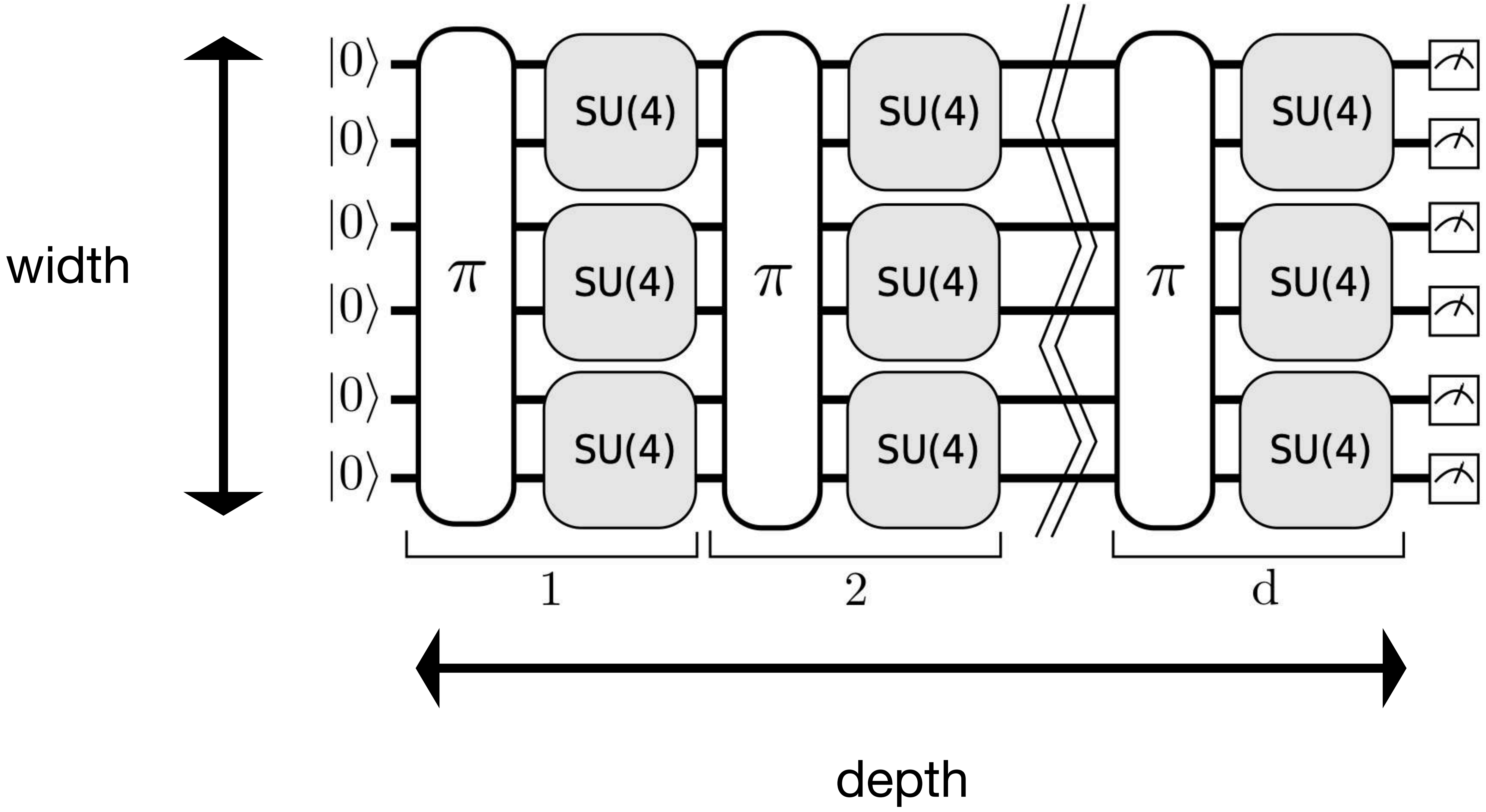


Noise



Qubit	Frequency (GHz)	T1 (µs)	T2 (µs)	√x (sx) error	Single-qubit Pauli-X error	Readout assignment error	CNOT error
Q0	4.969	100.76	81.91	2.371E-04	2.371E-04	3.300E-02	cx0_1: 7.456e-3
Q1	4.77	69.82	70.9	3.060E-04	3.060E-04	3.330E-02	cx1_2: 3.178e-2 cx1_0: 7.456e-3
Q2	5.015	86.34	154.96	5.442E-04	5.442E-04	2.380E-02	cx2_3: 8.190e-3 cx2_1: 3.178e-2
Q3	5.259	56.68	89.22	3.032E-04	3.032E-04	2.240E-02	cx3_4: 8.219e-3 cx3_2: 8.190e-3
Q4	4.998	98.57	155.61	3.035E-04	3.035E-04	2.240E-02	cx4_3: 8.219e-3

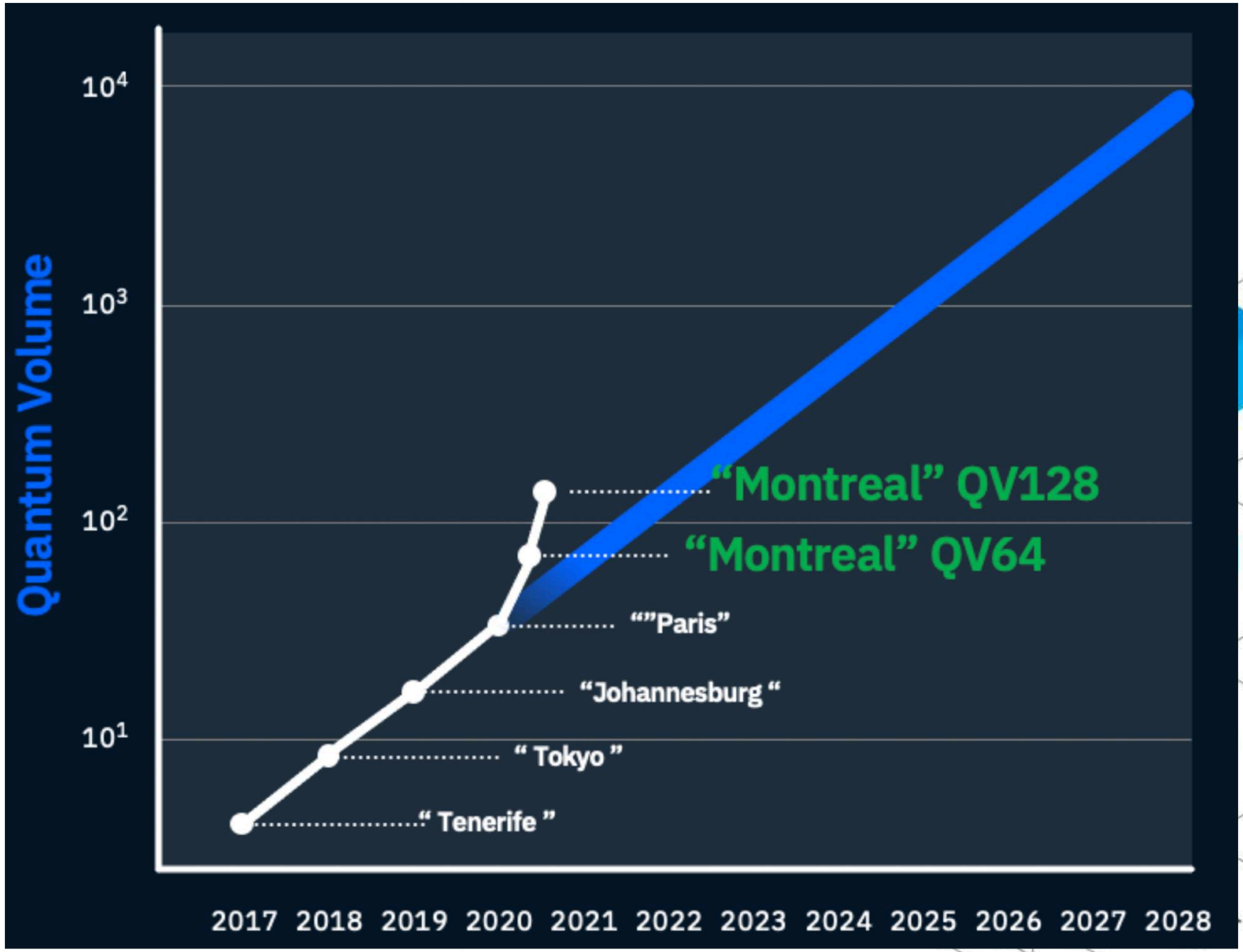
Quantum volume

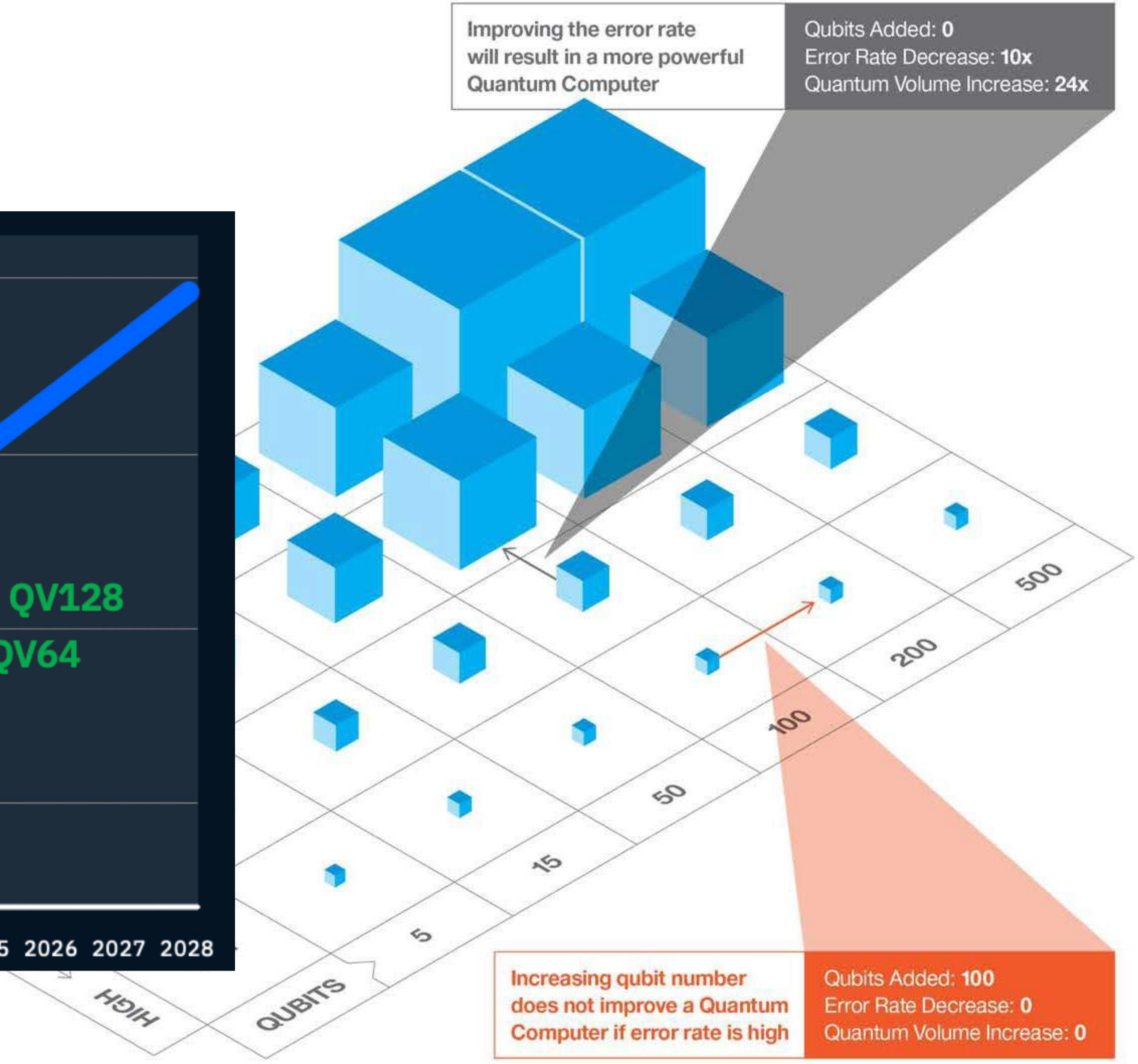


- The number of qubits is irrelevant if the depth of the circuit is limited by noise
- Limited connectivity effectively increases the depth (qubit swaps required)

Cross, Bishop, Sheldon, Nation, Gambetta Phys. Rev. A **100**, 032328

Quantum volume



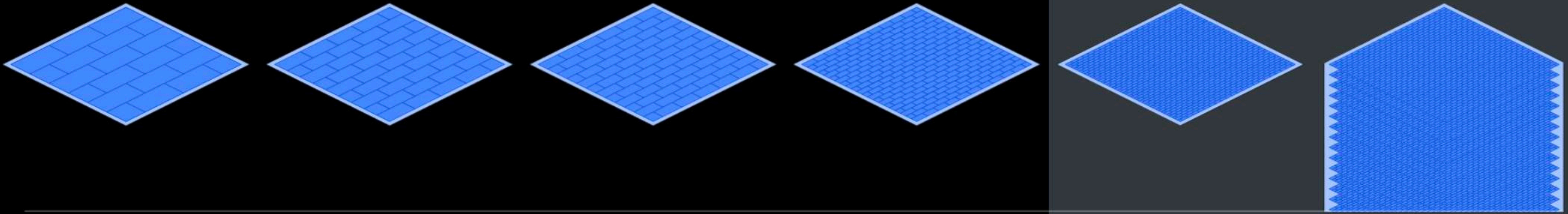


Source:
IBM Research

Scaling IBM Quantum technology

IBM **Quantum**

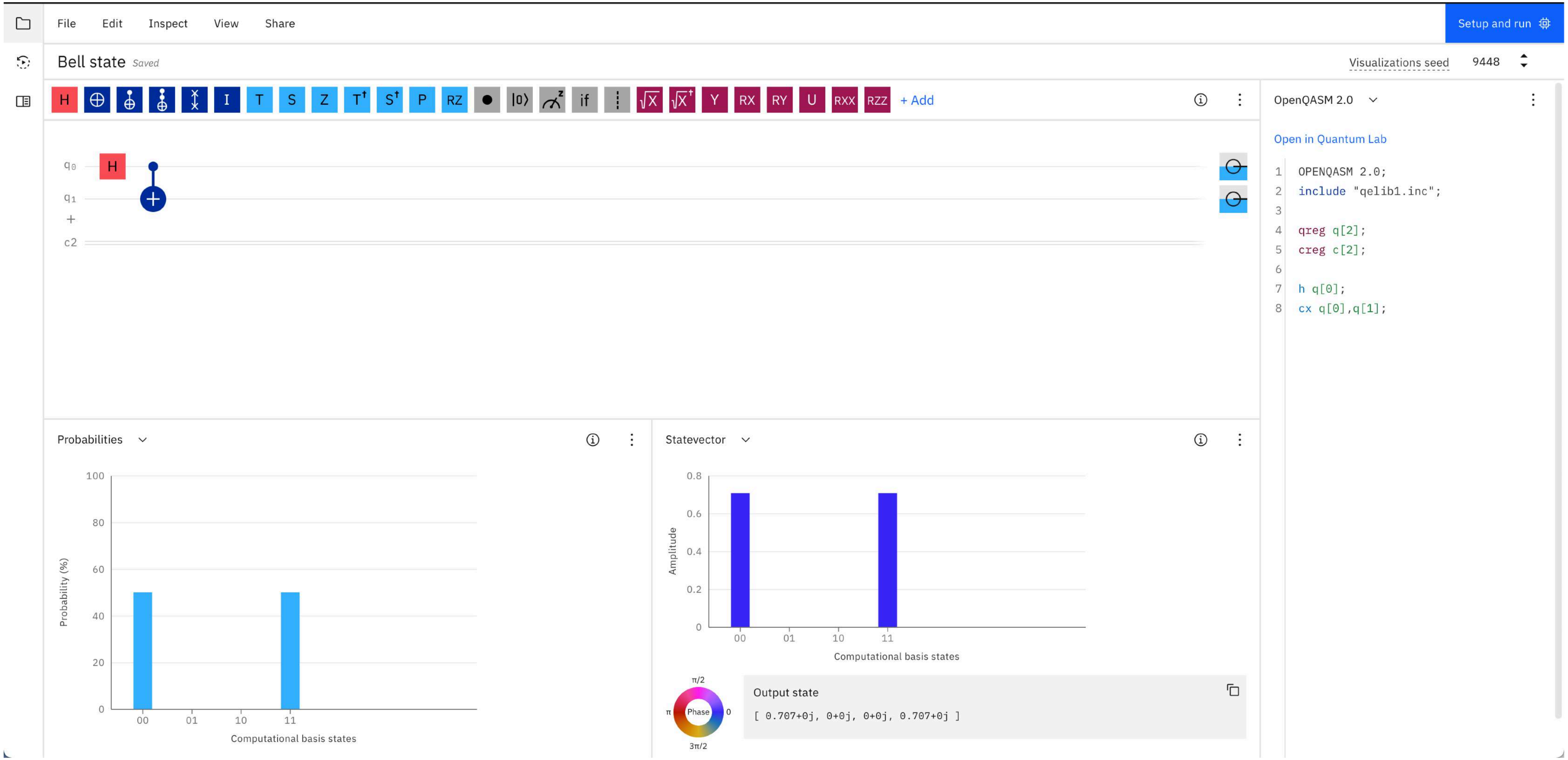
Released		In development		Next family of IBM Quantum systems	
2019	9/1/2020	2021	2022	2023	and beyond
27 qubits *Falcon*	65 qubits *Hummingbird*	127 qubits *Eagle*	433 qubits *Osprey*	1,121 qubits *Condor*	Path to 1 million qubits and beyond *Large scale systems*



Key advancement	Key advancement	Key advancement	Key advancement	Key advancement	
Optimized lattice	Scalable readout	Novel packaging and controls	Miniaturization of components	Integration	

Tutorial

How to use IBM Q devices



Qiskit

qiskit 0.24.0

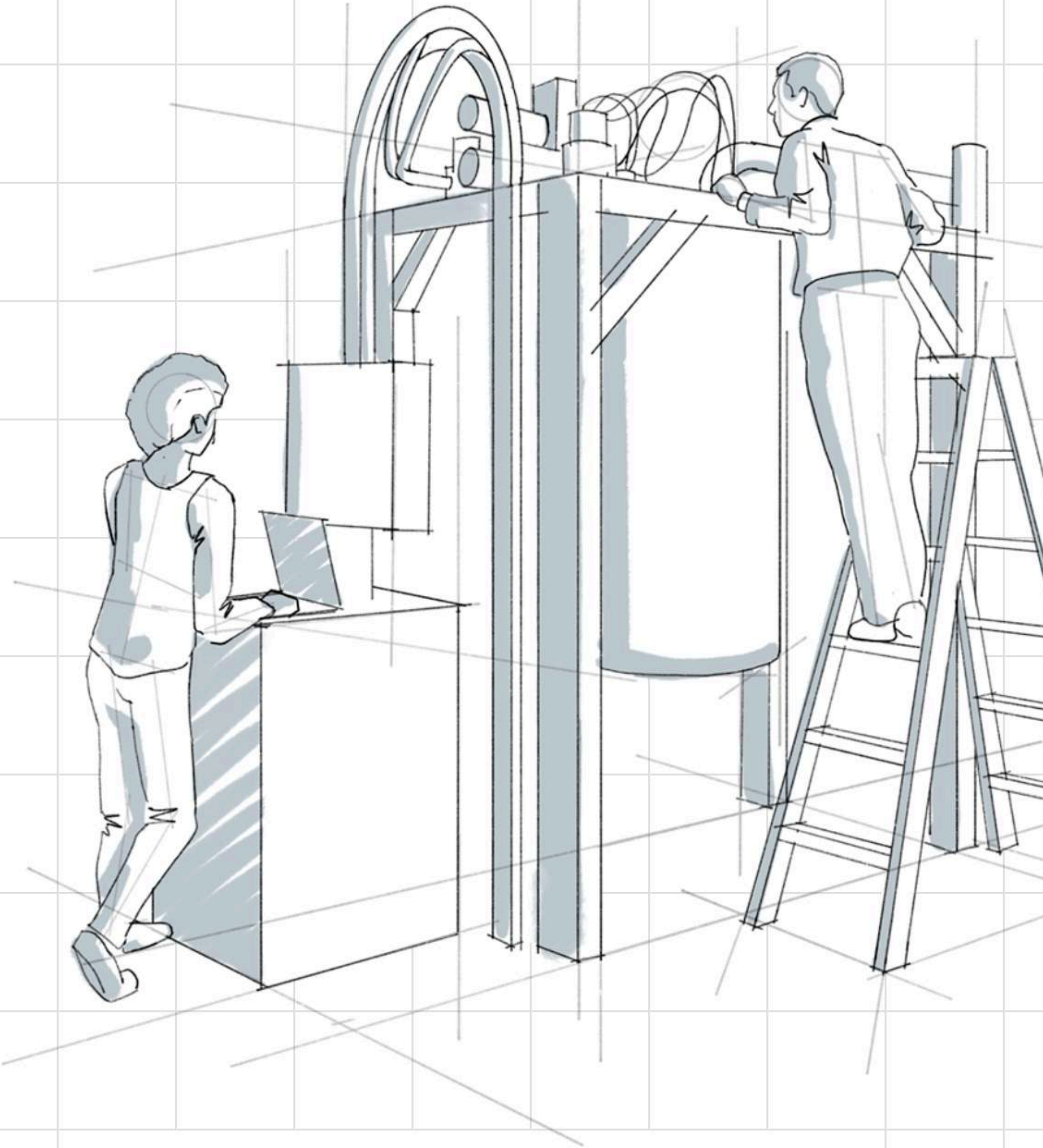
see release notes

Open-Source Quantum Development

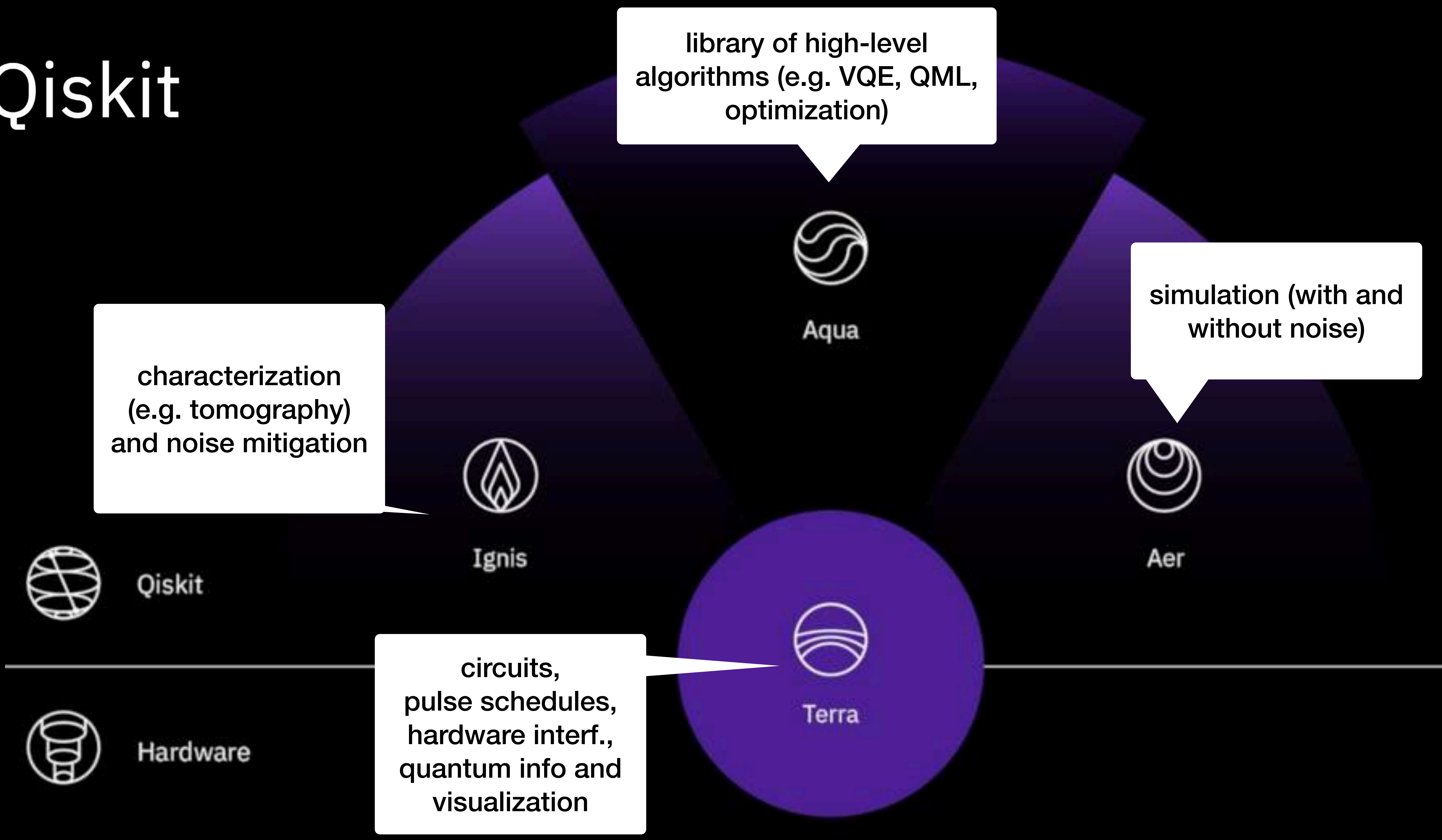
Qiskit [kiss-kit] is an open source SDK for working with quantum computers at the level of pulses, circuits and application modules.

Qiskit [quiss-kit] is an open source SDK for working with quantum computers at the level of pulses, circuits and application modules.

Get started



Qiskit


library of high-level algorithms (e.g. VQE, QML, optimization)
Aqua
simulation (with and without noise)
characterization (e.g. tomography) and noise mitigation
Ignis
Aer
Qiskit
Terra
Hardware
circuits, pulse schedules, hardware interf., quantum info and visualization

Qiskit textbook

https://qiskit.org/textbook

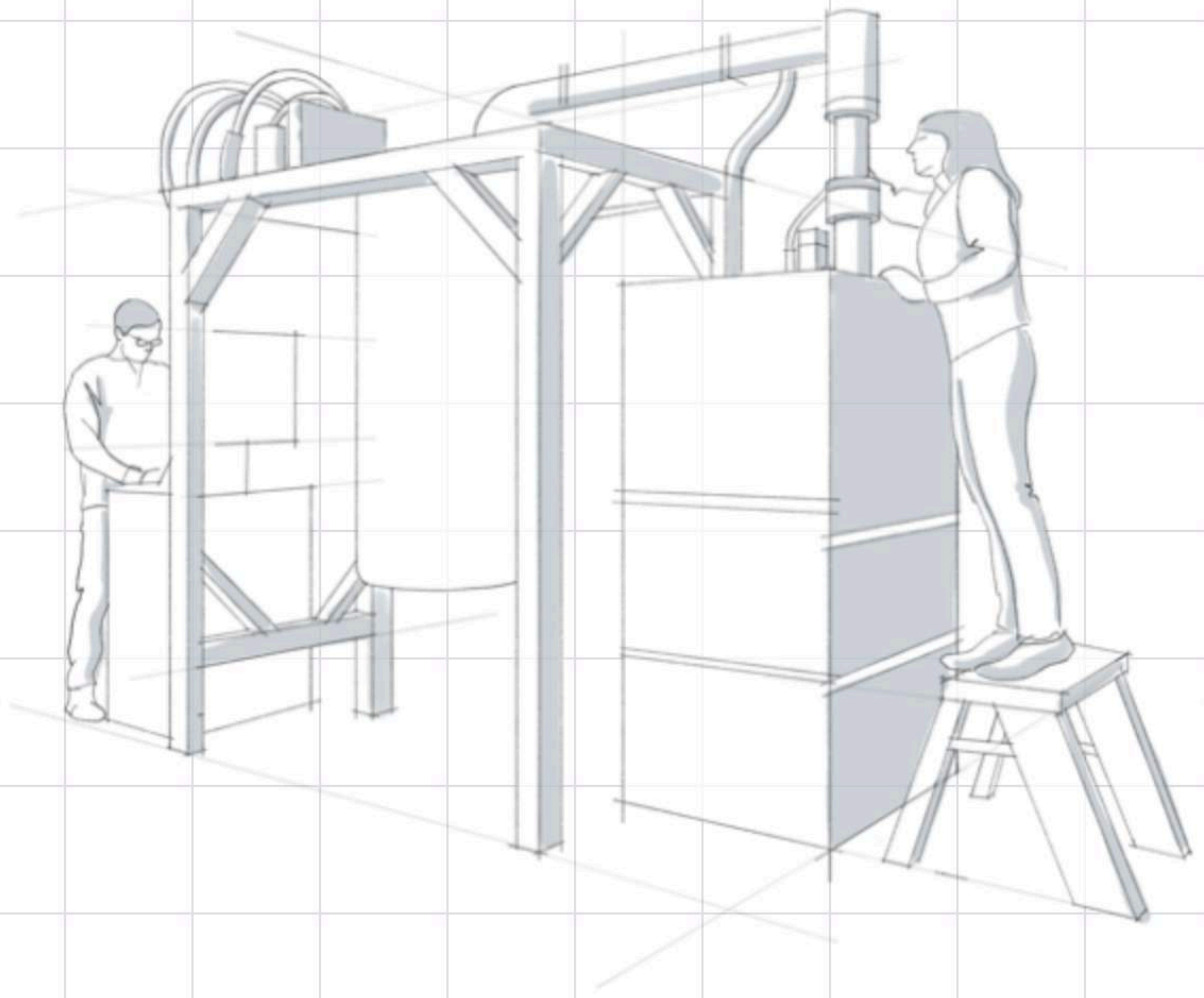
Learn Quantum Computation using Qiskit

Learn Quantum Computation using Qiskit

Greetings from the Qiskit Community team! This textbook is a university quantum algorithms/computation course supplement based on Qiskit to help learn:

1. The mathematics behind quantum algorithms
2. Details about today's non-fault-tolerant quantum devices
3. Writing code in Qiskit to implement quantum algorithms on IBM's cloud quantum systems

Read the textbook →



Qiskit Hello world!

```
pip install qiskit
```

```python
from qiskit import IBMQ, Aer, execute
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
```

Circuit definition:

```python
qr = QuantumRegister(2)
cr = ClassicalRegister(2)
qc = QuantumCircuit(qr, cr)
qc.h(0)
qc.cx(0, 1)
qc.measure(0, 0)
qc.measure(1, 1)
qc.draw(output='mpl', initial_state=True)
```

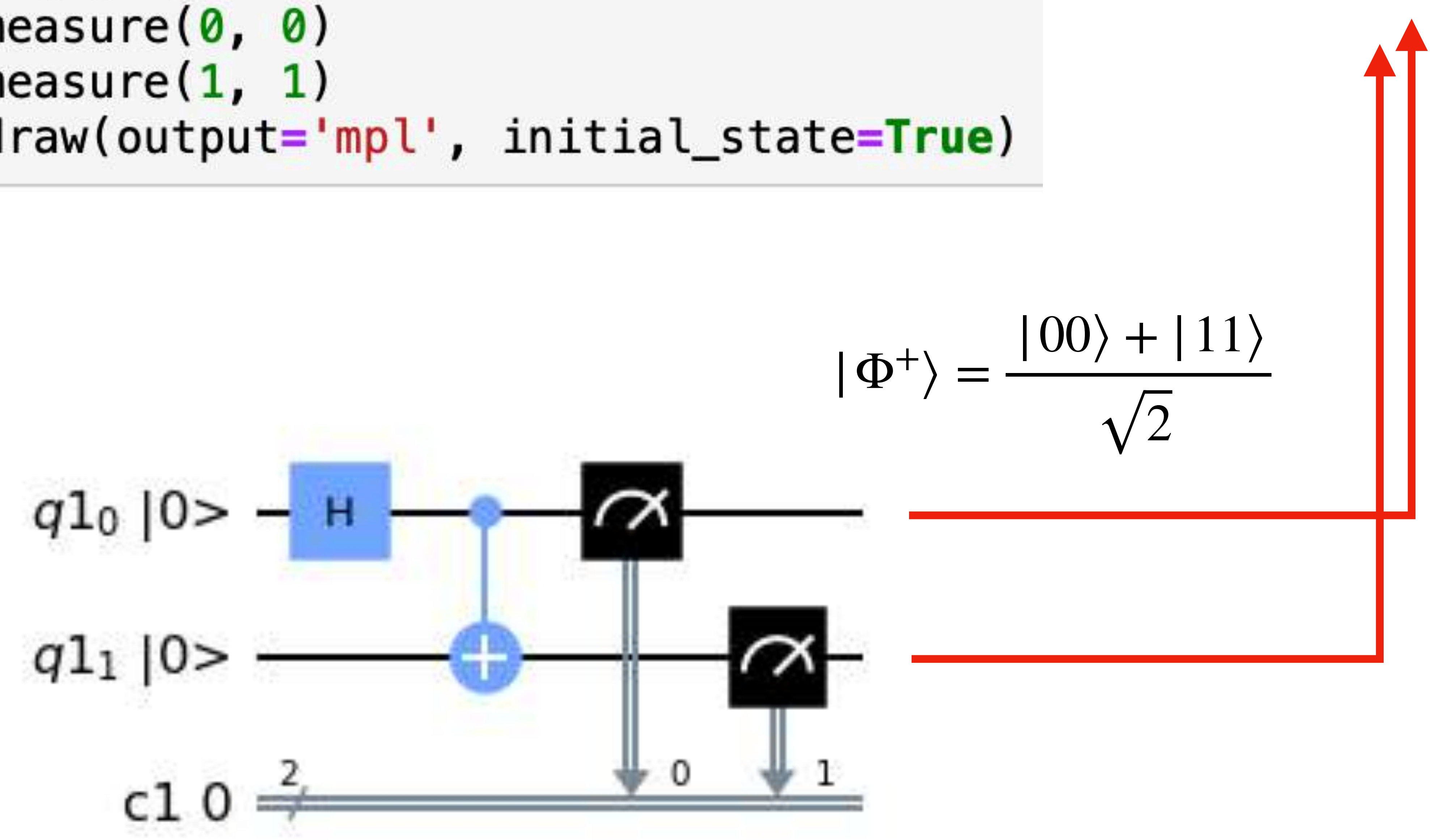
Simulation:

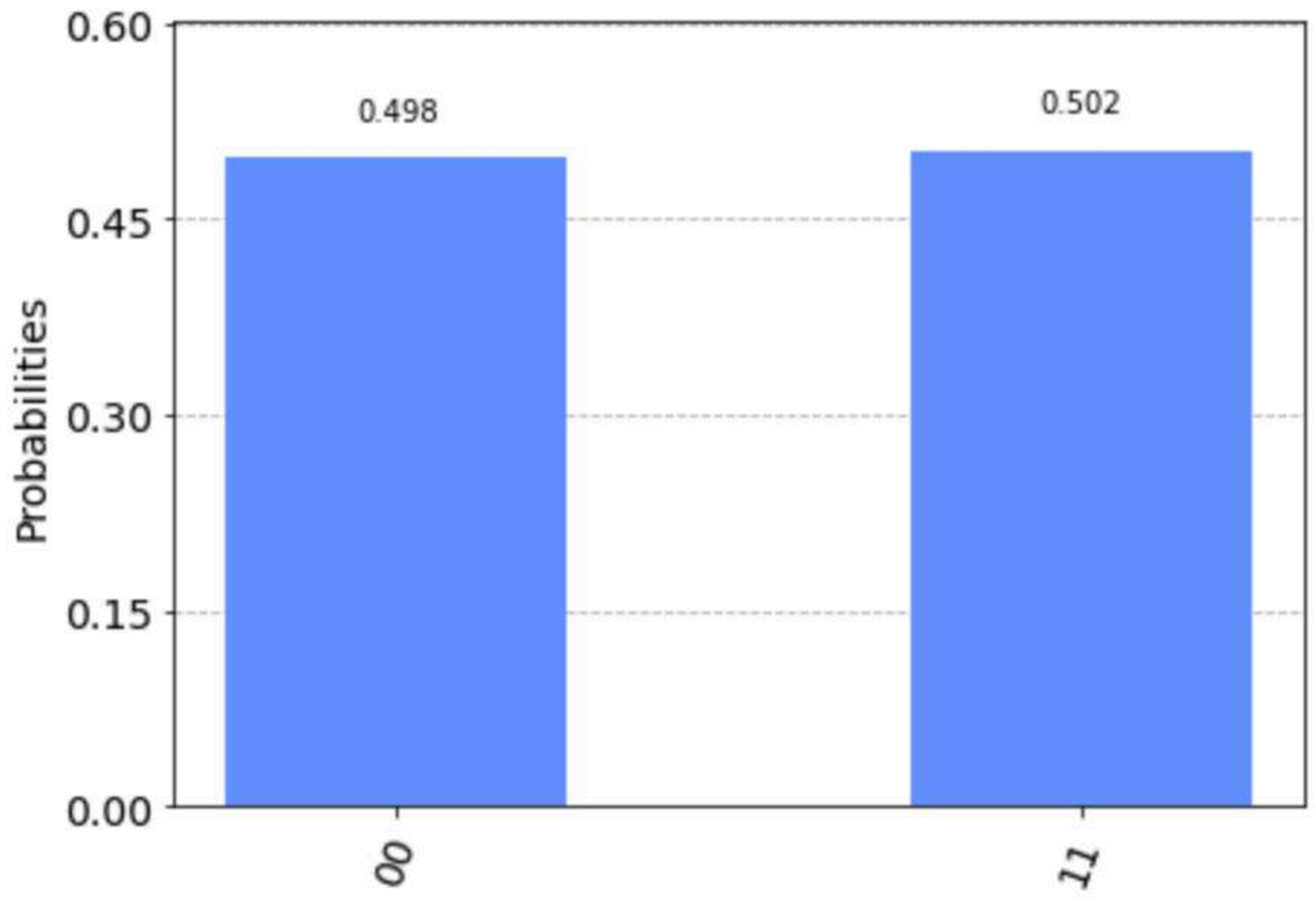
```python
job = execute(qc, Aer.get_backend('qasm_simulator'), shots=8192)
result = job.result()
result.get_counts()
```

```
{'00': 4076, '11': 4116}
```

LSB on the right!!!!

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$





Qiskit Hello world!

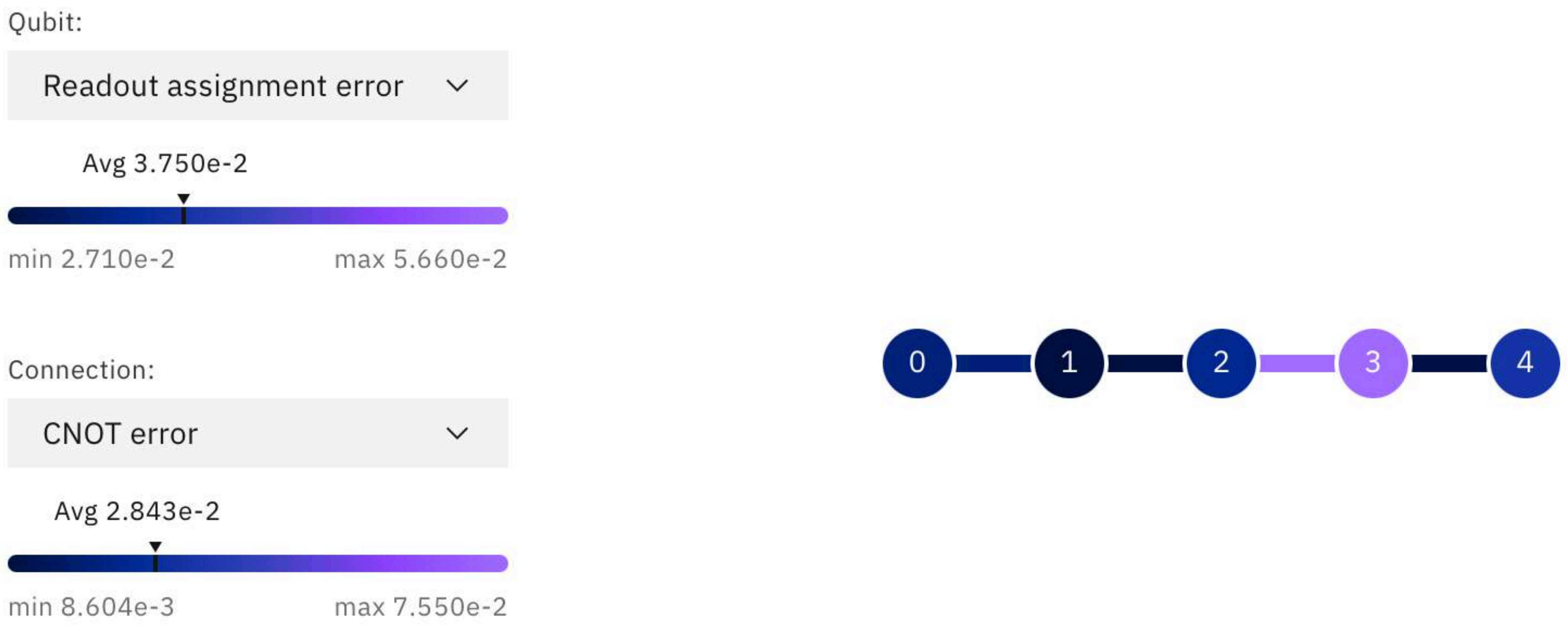
Execution on a real device

```
IBMQ.load_account()
provider = IBMQ.get_provider(hub='ibm-q-research', group='uni-turku-4', project='main')
backend = provider.get_backend('ibmq_bogota')
```

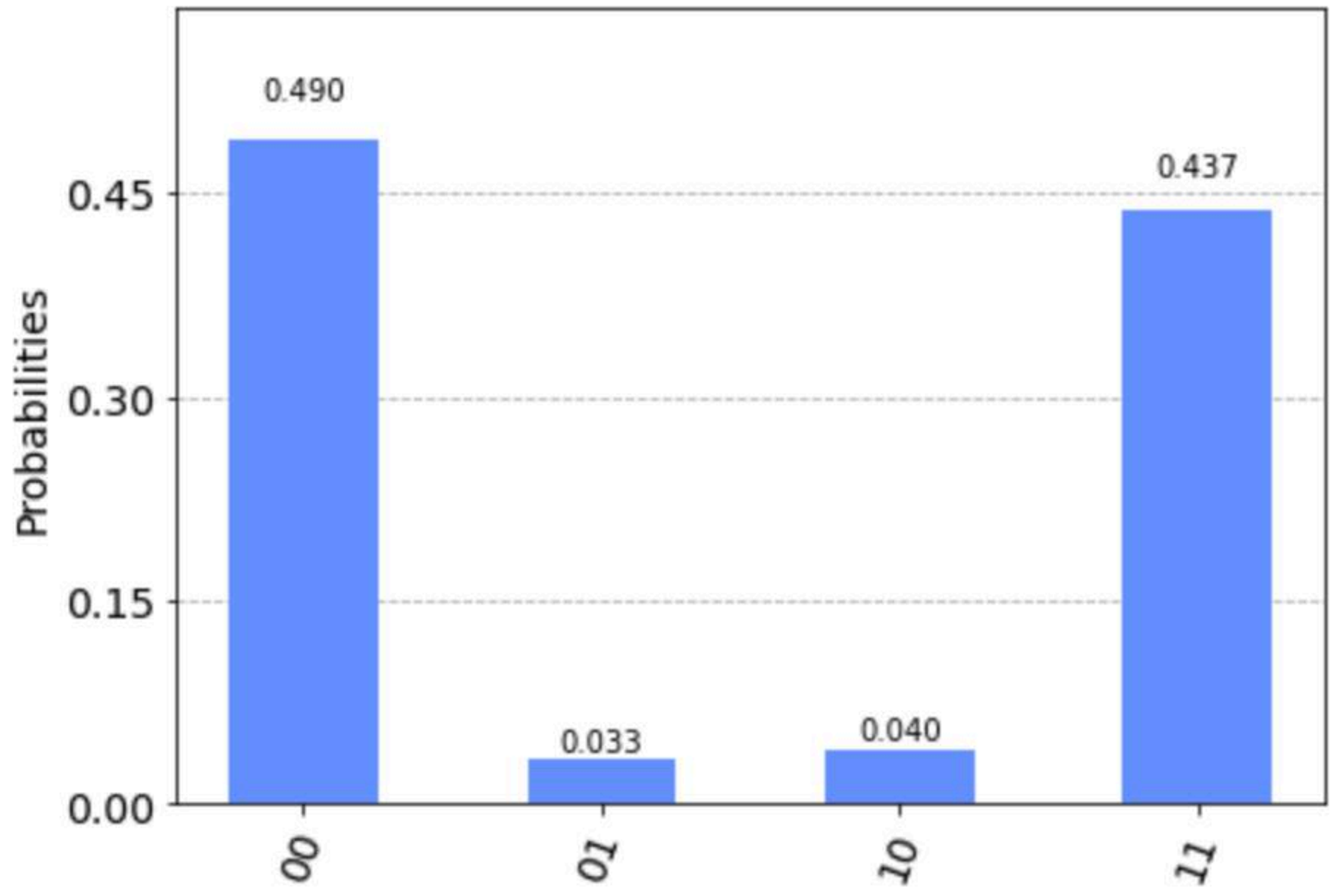
```
job = execute(qc, backend, shots=8192)
```

```
job.status()
```

```
<JobStatus.DONE: 'job has successfully run'>
```



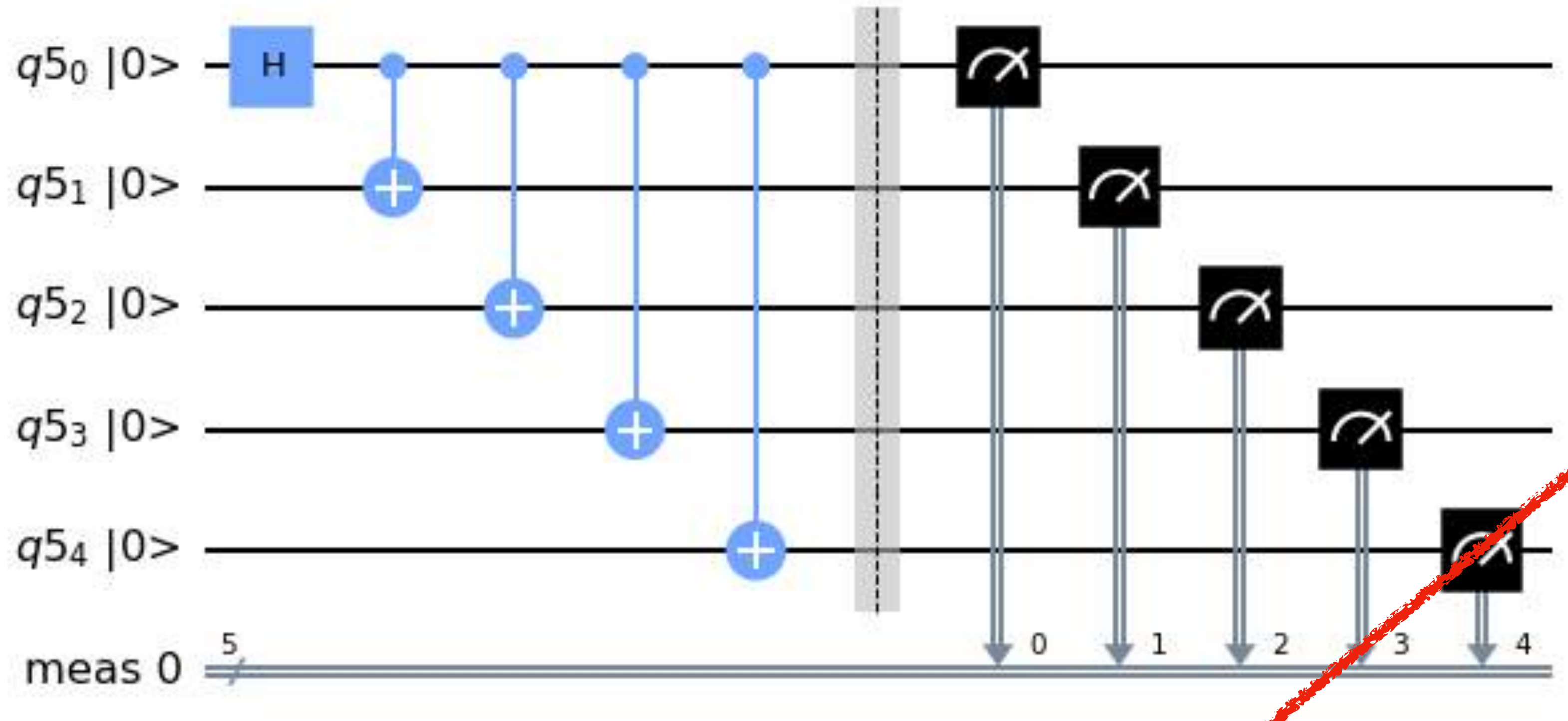

```
result = job.result()
plot_histogram(result.get_counts())
```

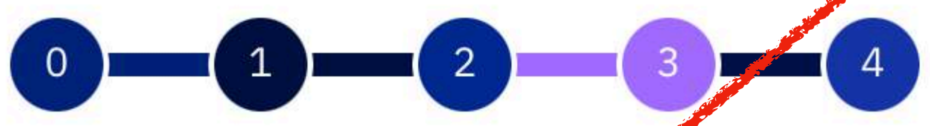


Improving your results

```
qr = QuantumRegister(5)
qc = QuantumCircuit(qr)

qc.h(0)
for i in range(4):
    qc.cx(0, i + 1)

qc.measure_all()
```

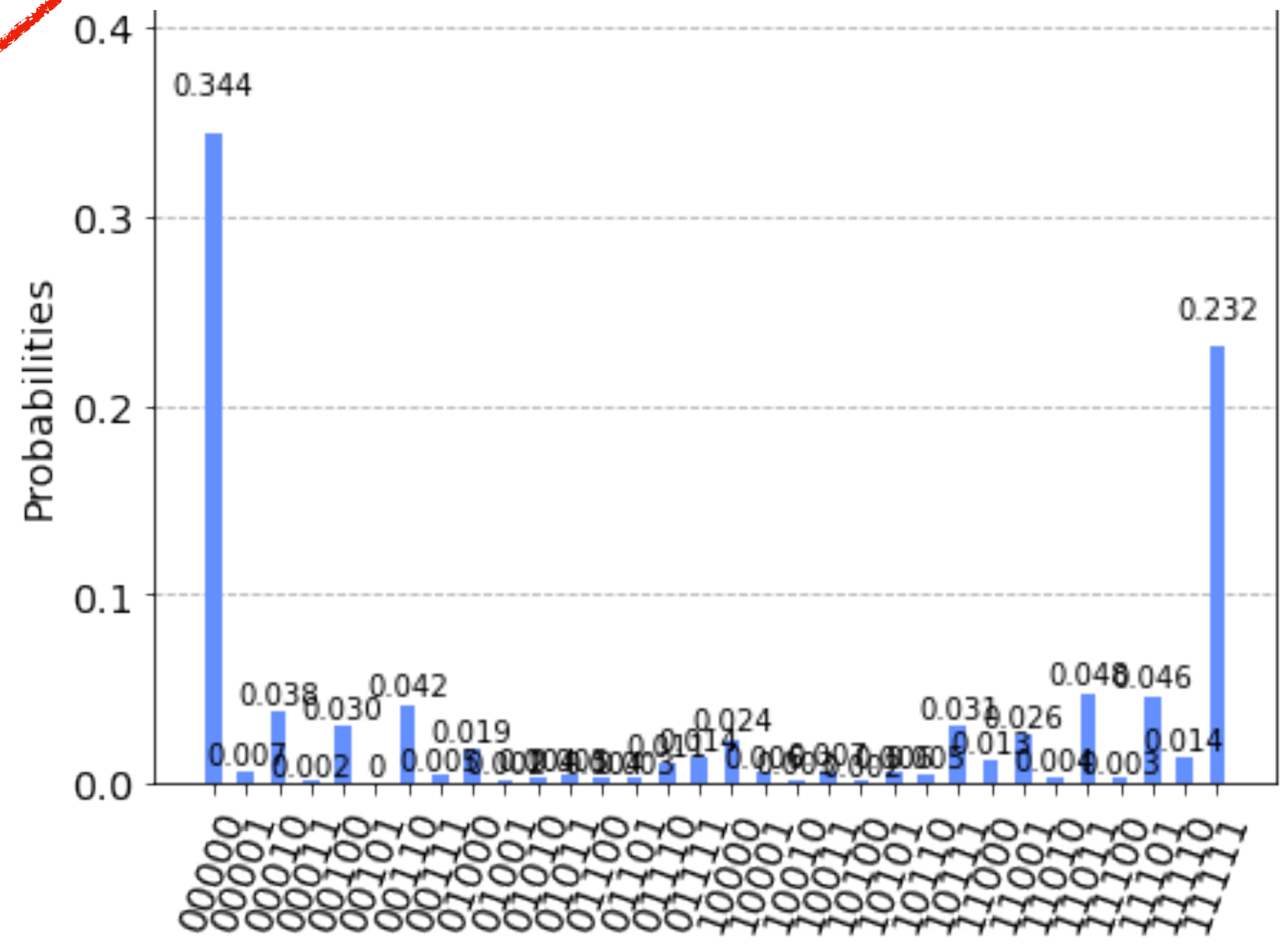
$$\frac{|00000\rangle + |11111\rangle}{\sqrt{2}}$$



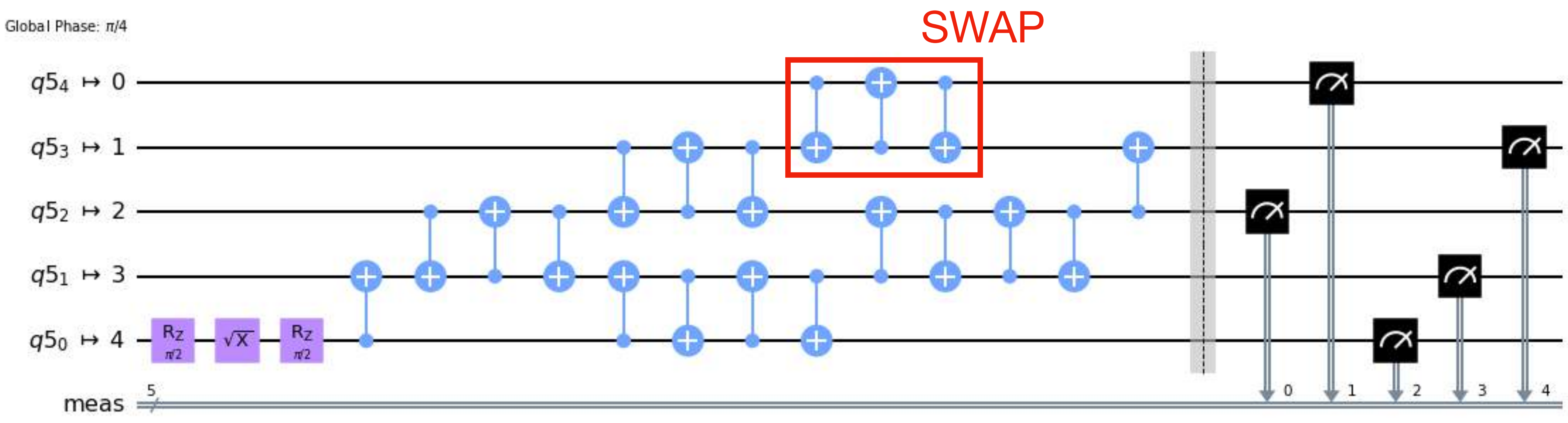


ibmq_bogota

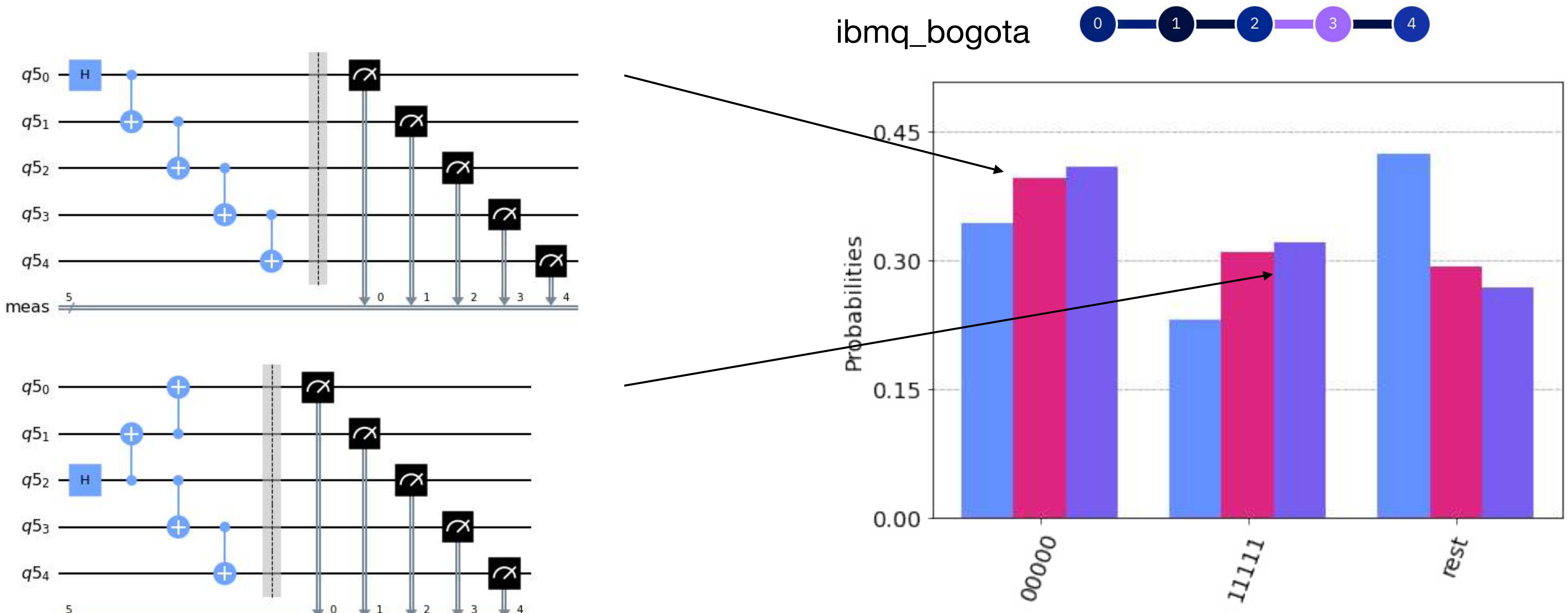
```
job = execute(qc, backend, shots=8192)
```



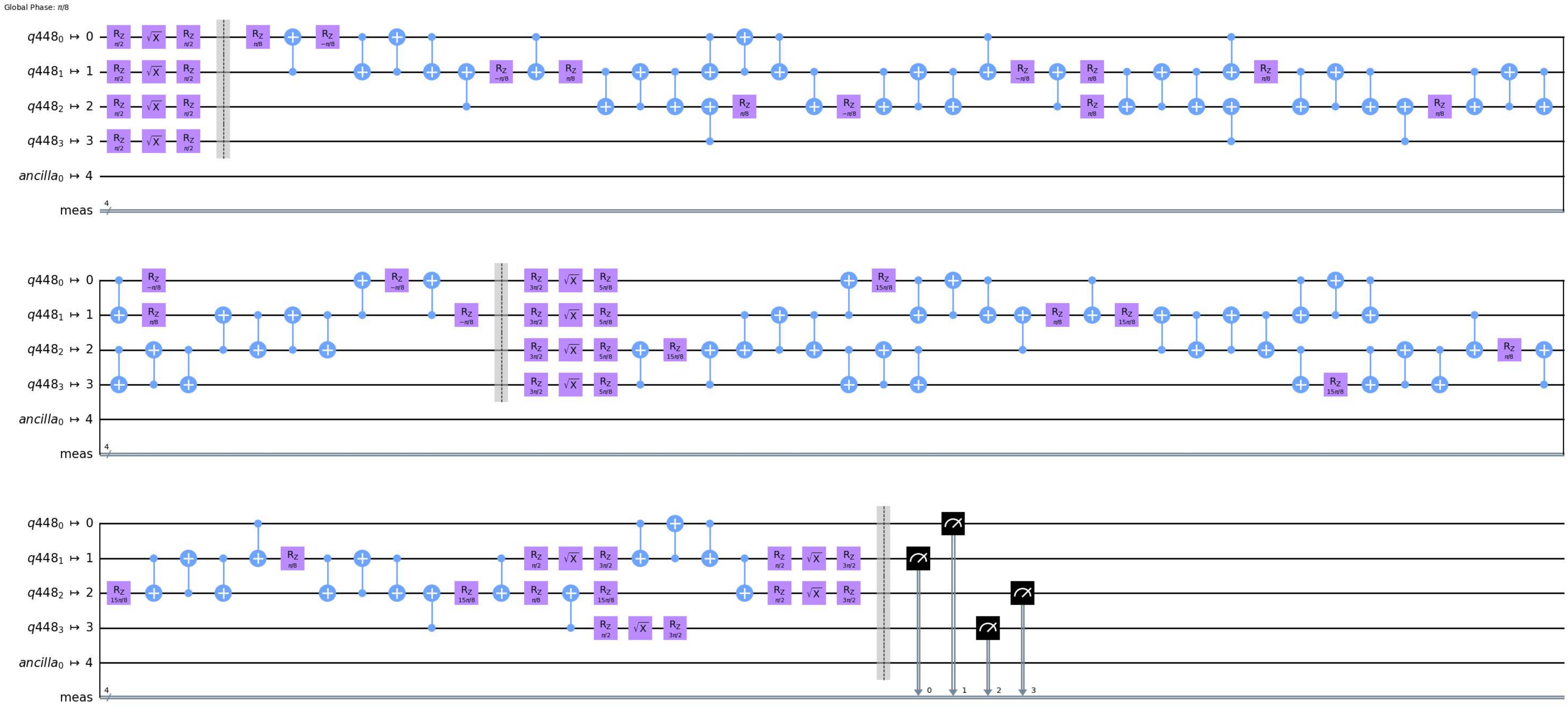

```
from qiskit import transpile
tqc = transpile(qc, backend, optimization_level=1)
```



Improving your results



Grover search



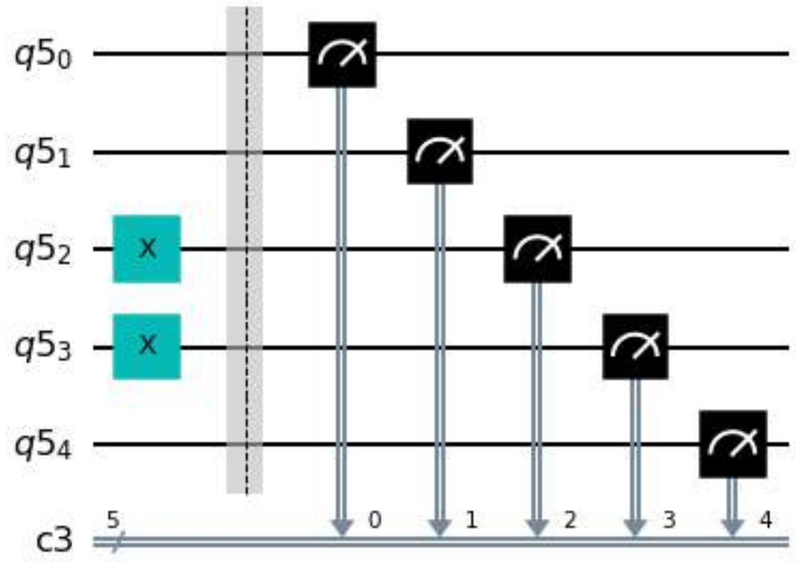
Improving your results

Measurement noise mitigation

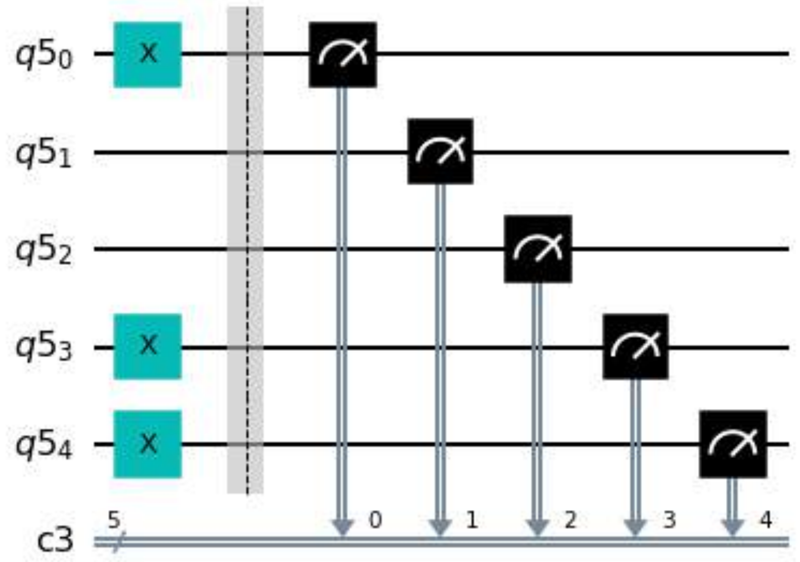
Prepares all 2^n bitstrings and measures them

```
## Mitigation
# Import measurement calibration functions
from qiskit.ignis.mitigation.measurement import (complete_meas_cal,
                                                  CompleteMeasFitter)

meas_calibs, state_labels = complete_meas_cal(qr=qr,
                                              qubit_list=range(5),
                                              circlabel='mcal')

calib_job = qiskit.execute(meas_calibs, backend=backend, shots=8192)
```

…

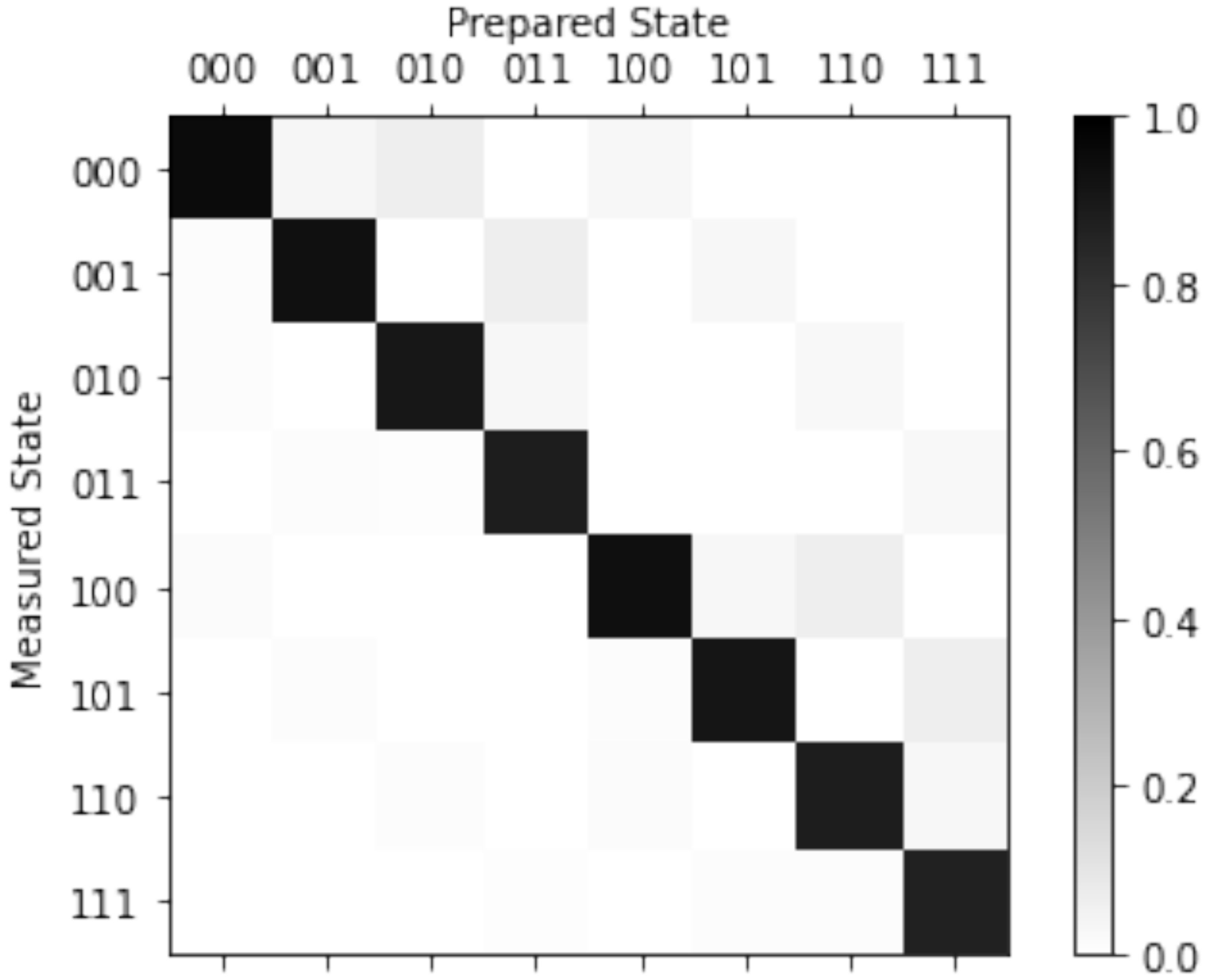


…



Builds a confusion matrix

```
meas_fitter = CompleteMeasFitter(cal_results,
                                 state_labels,
                                 qubit_list=range(5),
                                 circlabel='mcal')
# Plot the calibration matrix
meas_fitter.plot_calibration()
```

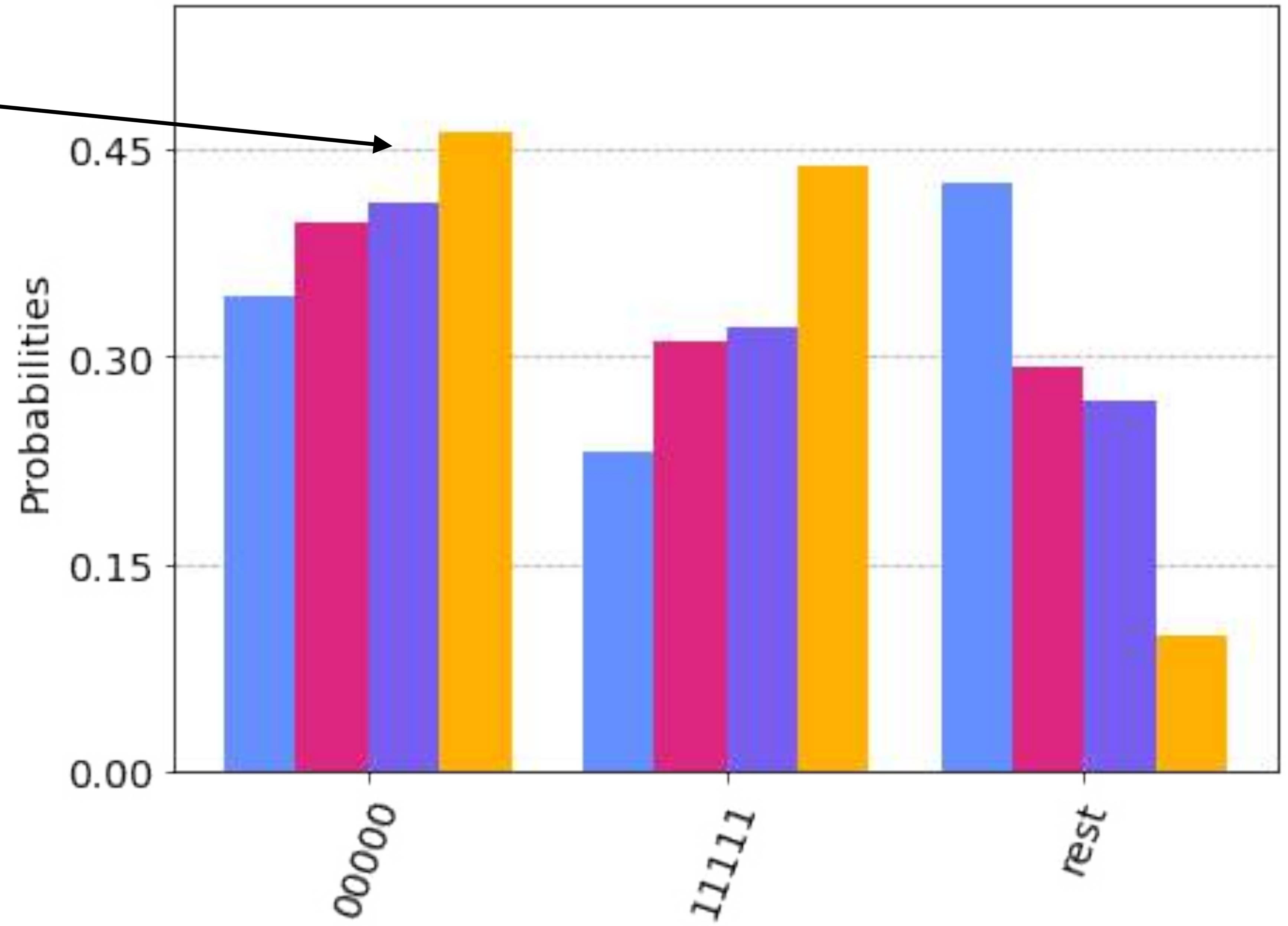


Improving your results

Measurement noise mitigation

Applies the inverse of the confusion matrix to the results

```
meas_filter = meas_fitter.filter
mitigated_result = meas_filter.apply(result)
```



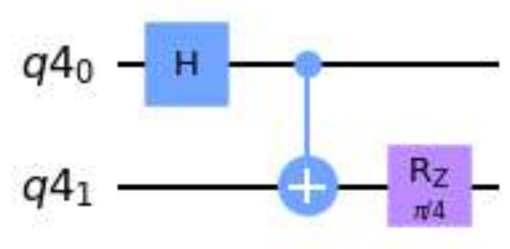
For short circuits the readout error is quite relevant!

n-qubit tomography

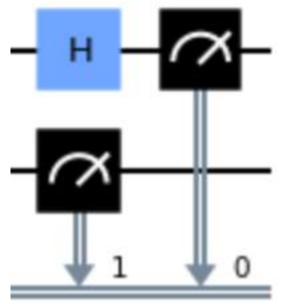
Reconstruct the quantum state ρ

```
# Tomography
from qiskit.ignis.verification.tomography import state_tomography_circuits
from qiskit.ignis.verification.tomography import StateTomographyFitter
```

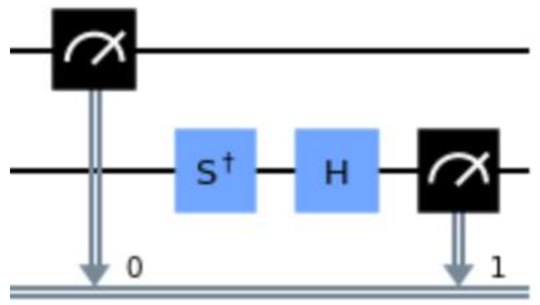
```
# Construct the tomography circuits by passing the initial circuit
# and a list of the qubits of interest
tomo_circuits = state_tomography_circuits(qc, [qr[0], qr[1]])
```



$\langle \sigma_x \otimes \sigma_z \rangle$



$\langle \sigma_z \otimes \sigma_y \rangle$

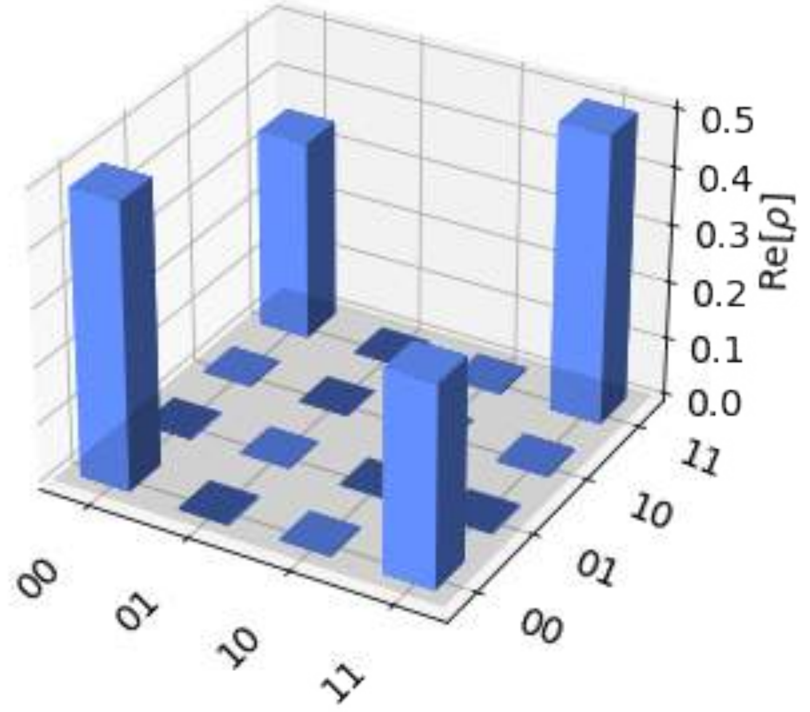


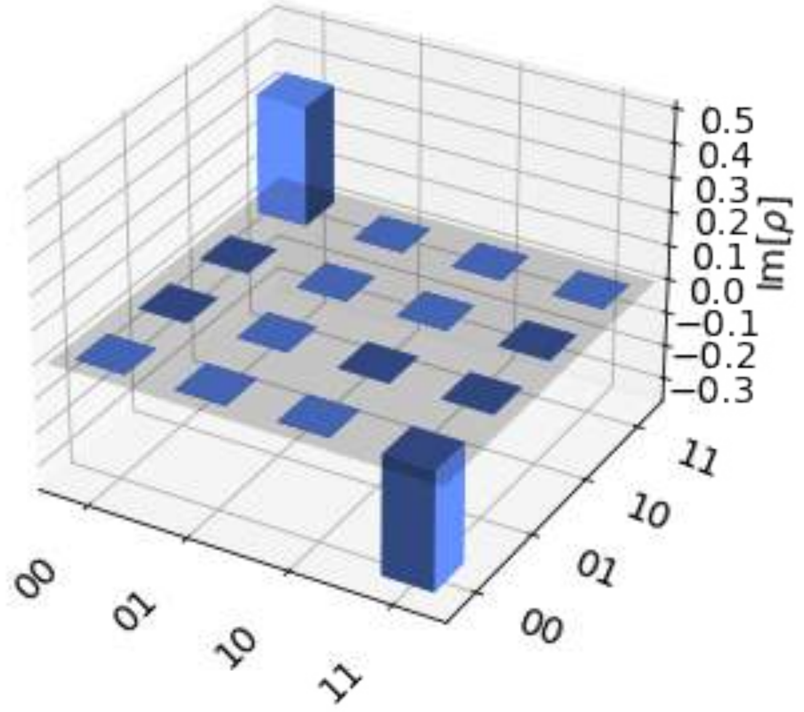
Total: 3^N circuits

```
# The fitter takes the result of the list of tomography circuits
# and reconstructs the density matrix using maximum likelihood
tomo_fitter = StateTomographyFitter(result, tomo_circuits)
rho = tomo_fitter.fit()
```

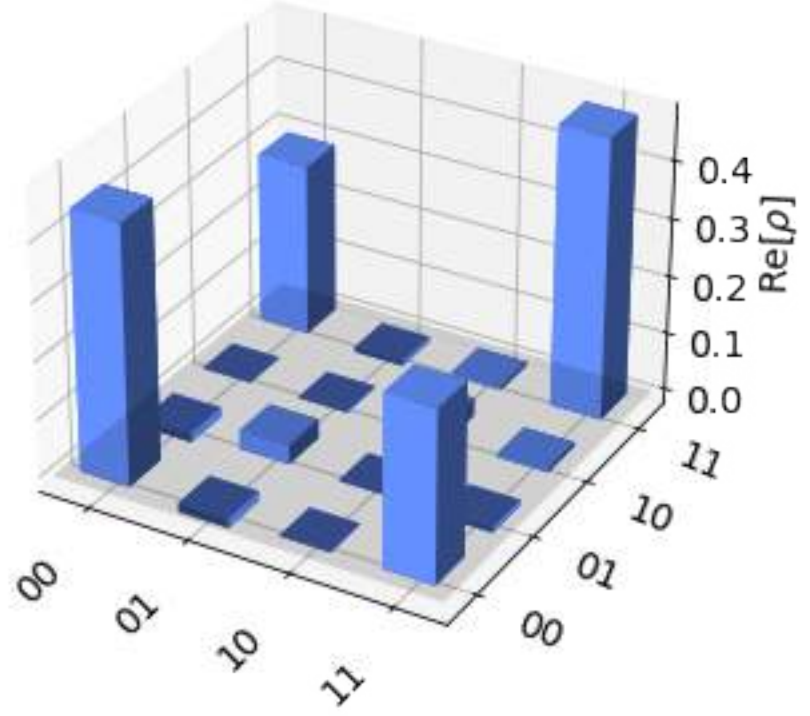
Also process tomography!

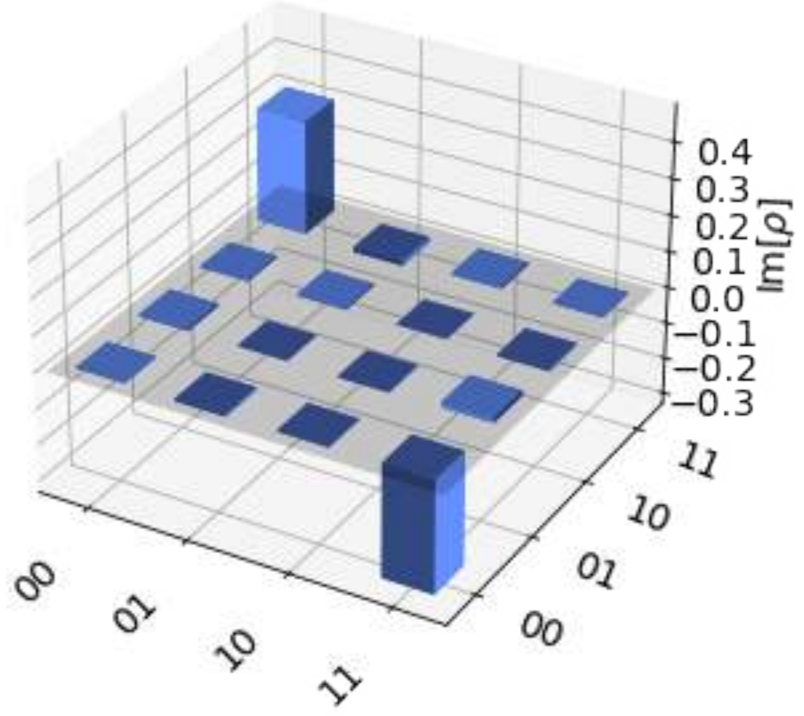
Simulation





Experiment





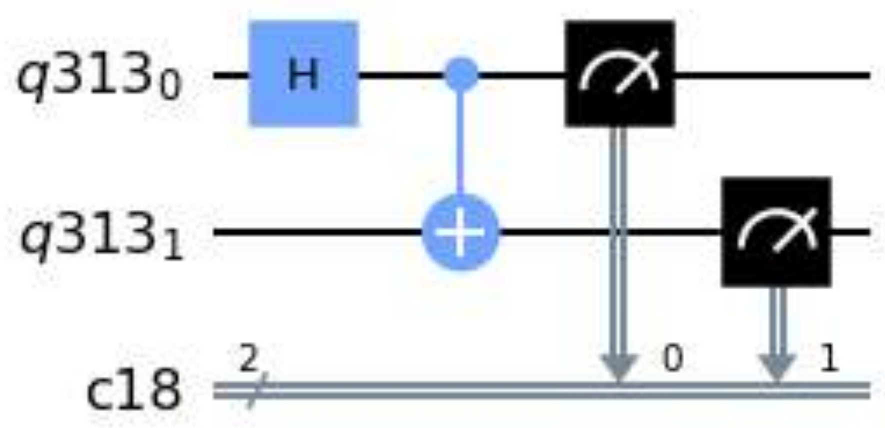
Fidelity: 0.92

Take home messages

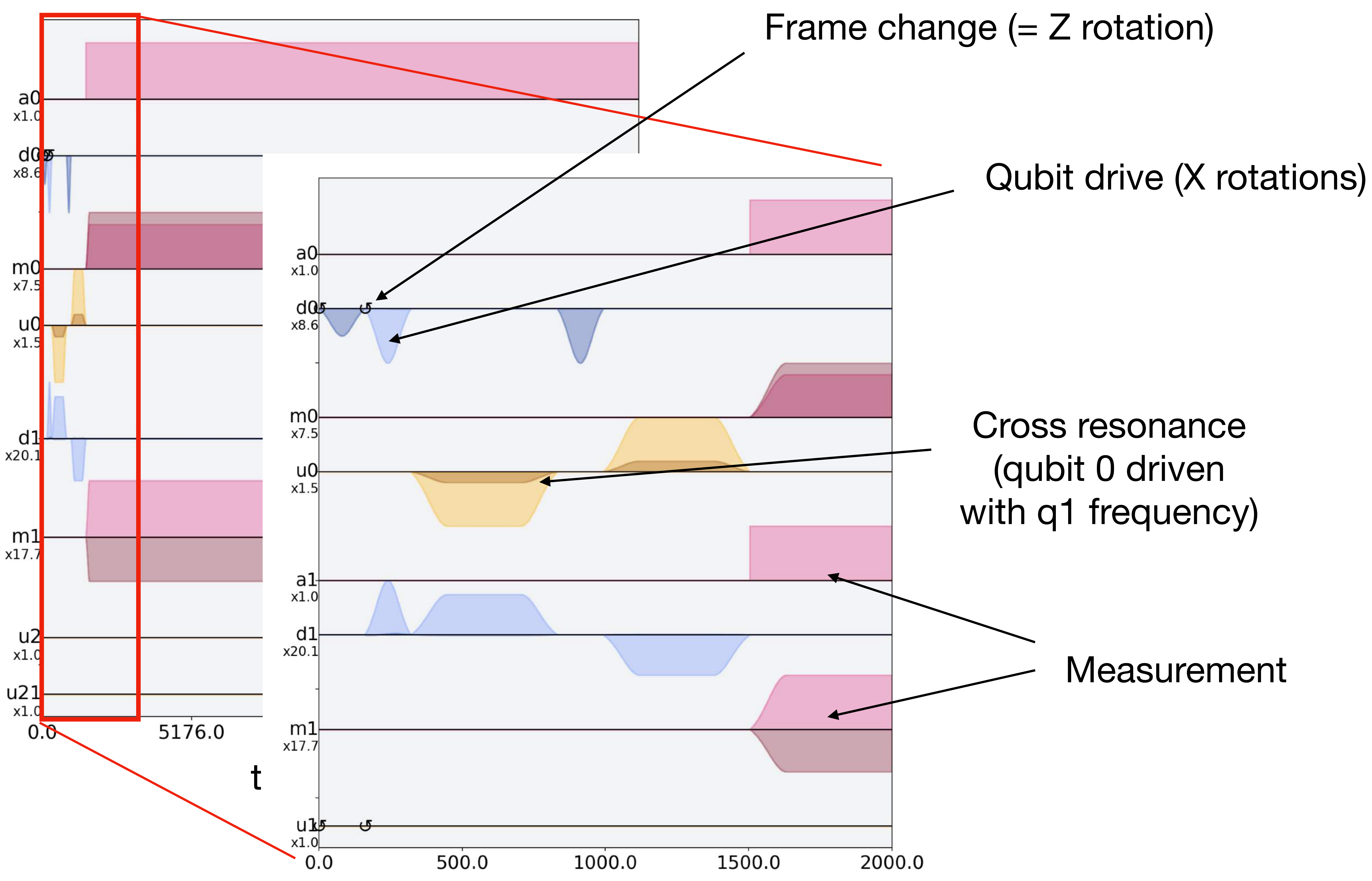
- Noise and **connectivity layout** pose limitations to the size of circuits
- **Do not trust the compilers**! They are still rudimentary
- **Handmade short circuits** that take into account the connectivity layout can give good experimental resuts
- `qiskit.ignis` gives tools such as **measurement noise mitigation** and state and process **tomography** to improve and analyze the results

Pulses

```
from qiskit.compiler import schedule
tqc = transpile(qc, backend)
sched = schedule(tqc, backend)
```

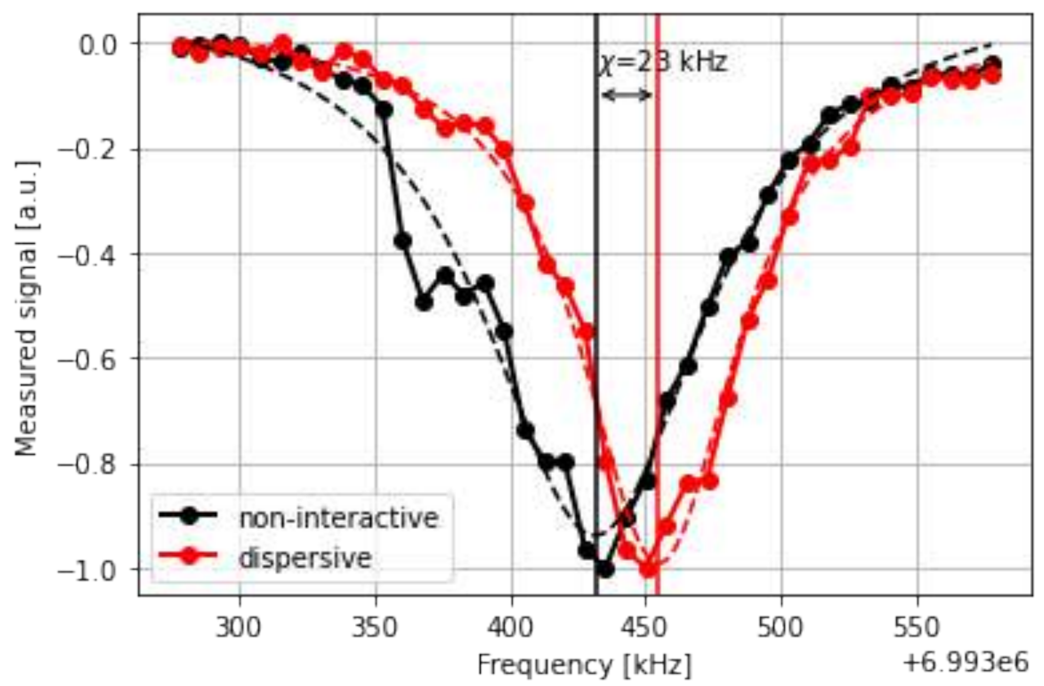


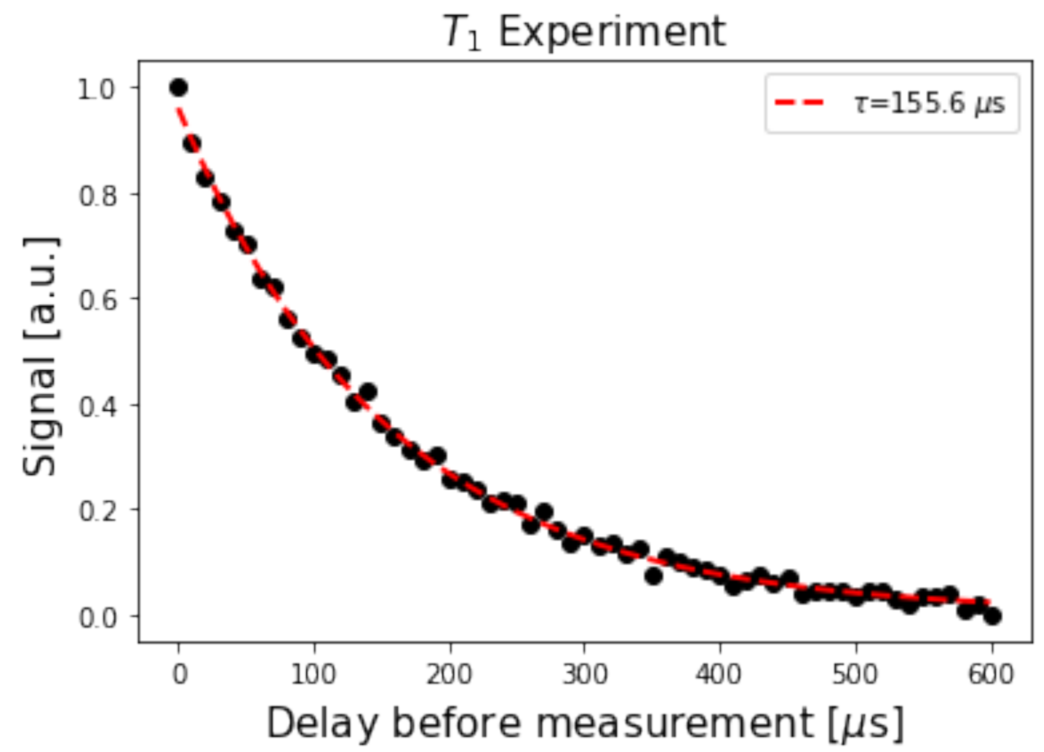
$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$



Pulses

- Access to higher energy levels $|0\rangle, |1\rangle, |2\rangle, \ldots$
- Studying qubit-cavity interaction with JC Hamiltonians
- Characterizing qubit noise, frequency etc.
- Might be very interesting for projects in open quantum systems, metrology, parameter estimation…
- Tutorials on the qiskit textbook

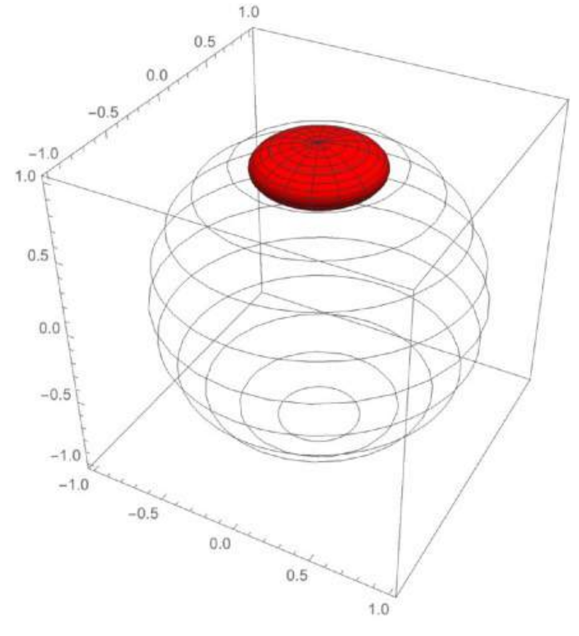


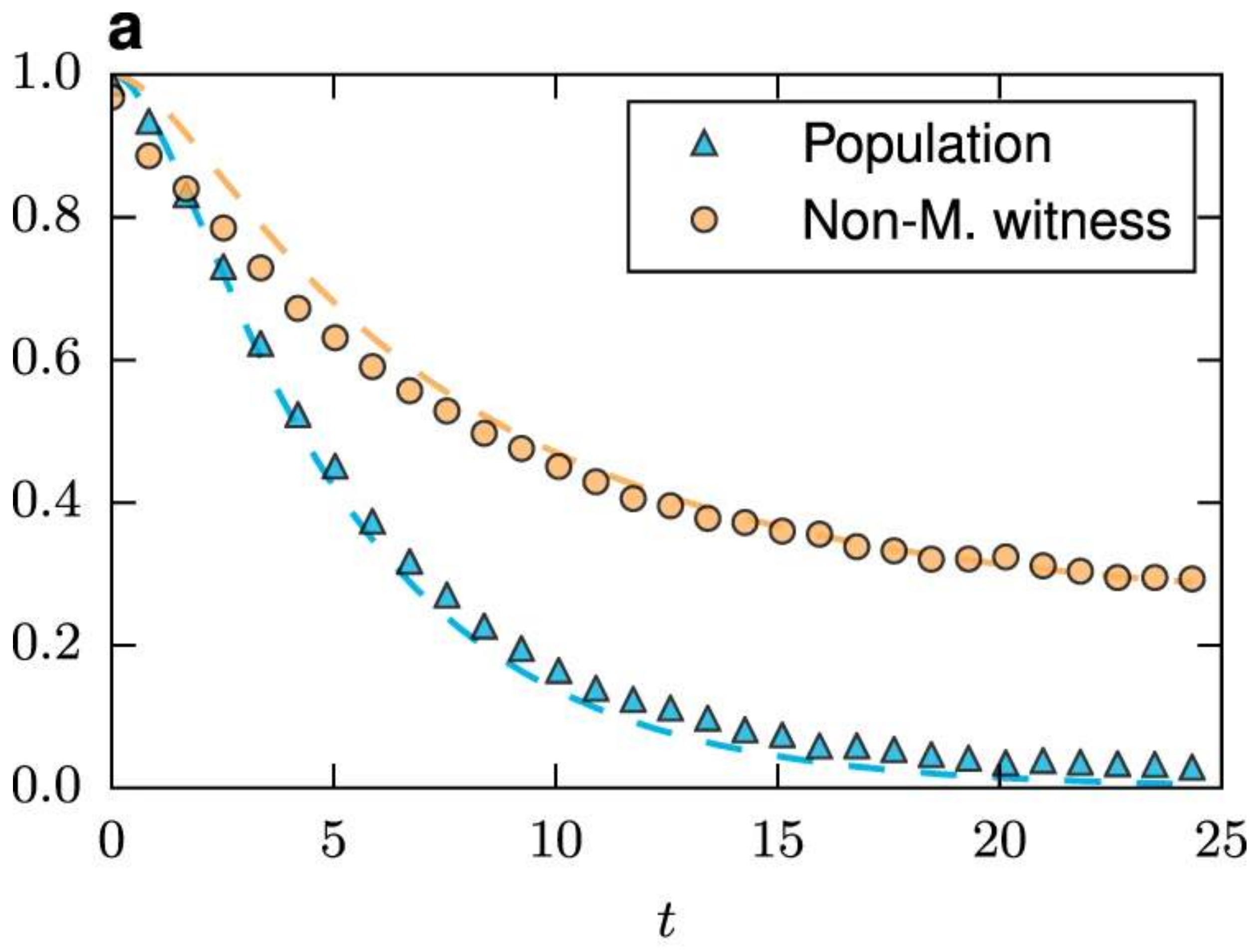


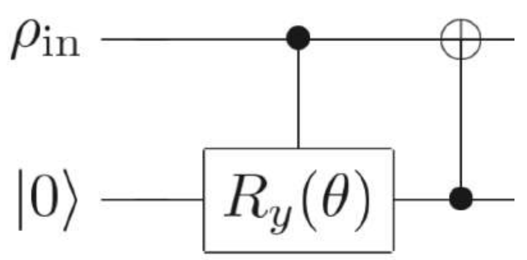
Example applications

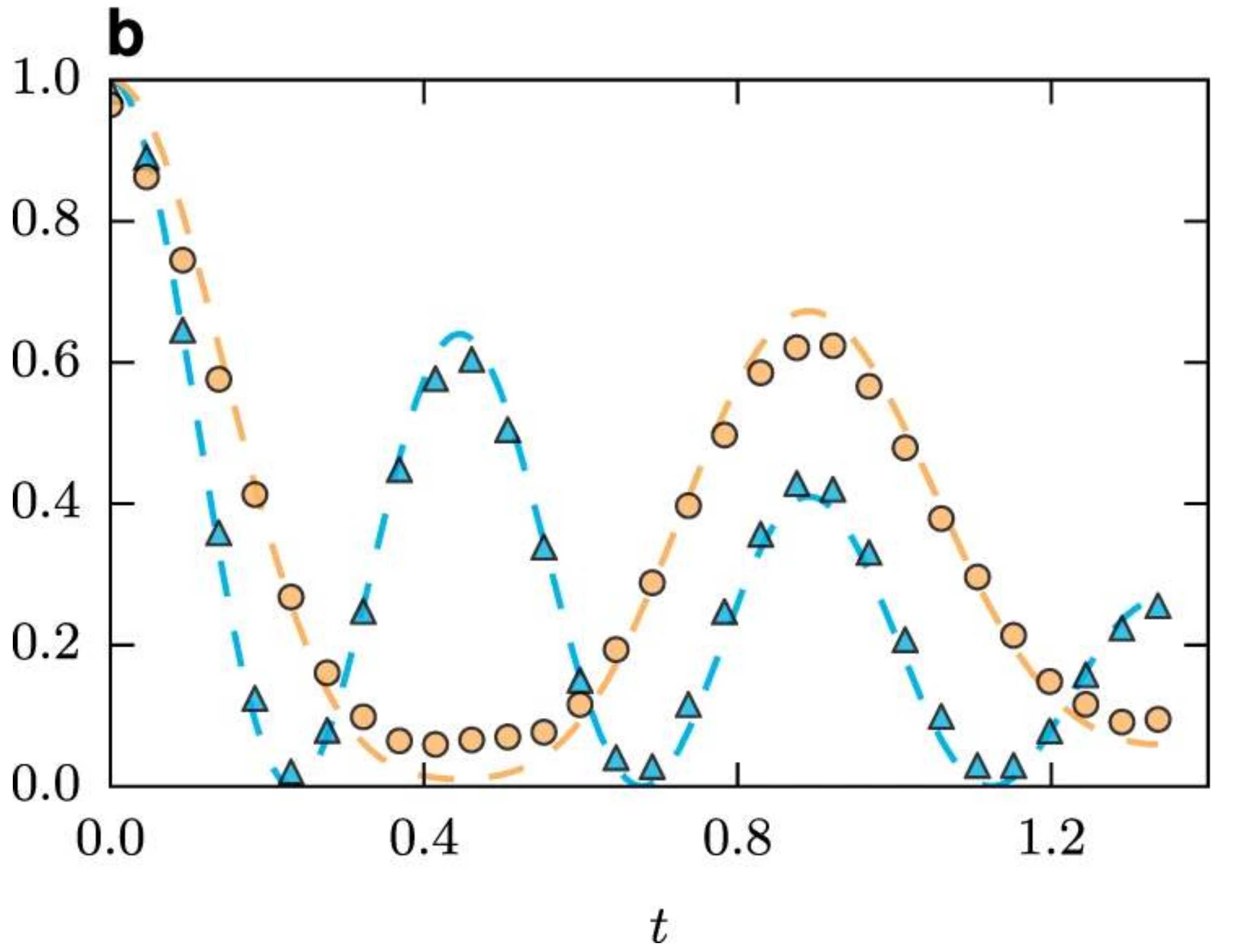
Simulating open quantum systems

Amplitude damping

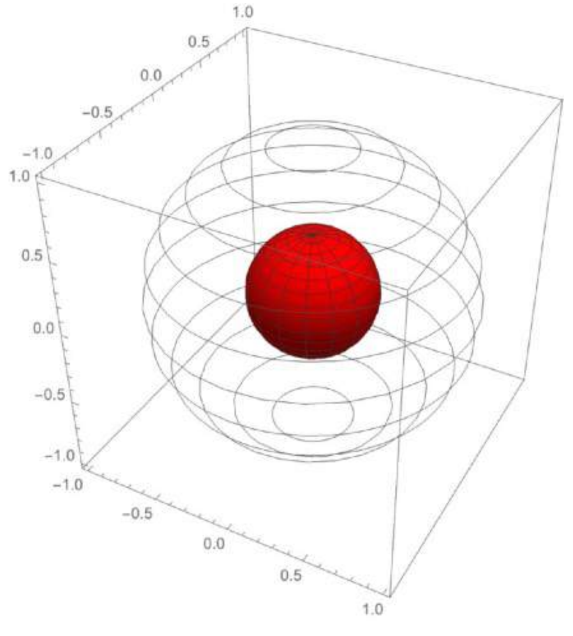


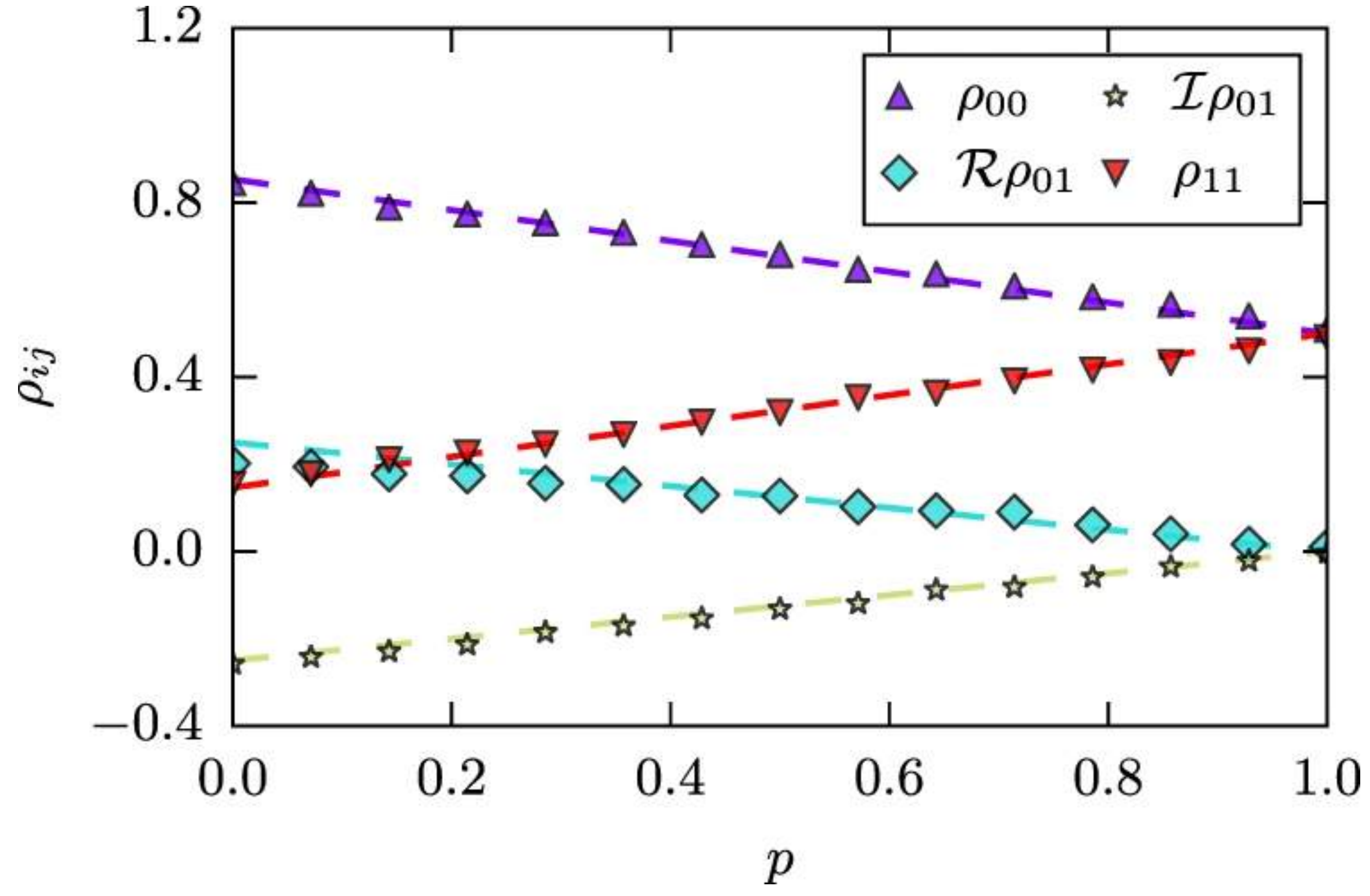


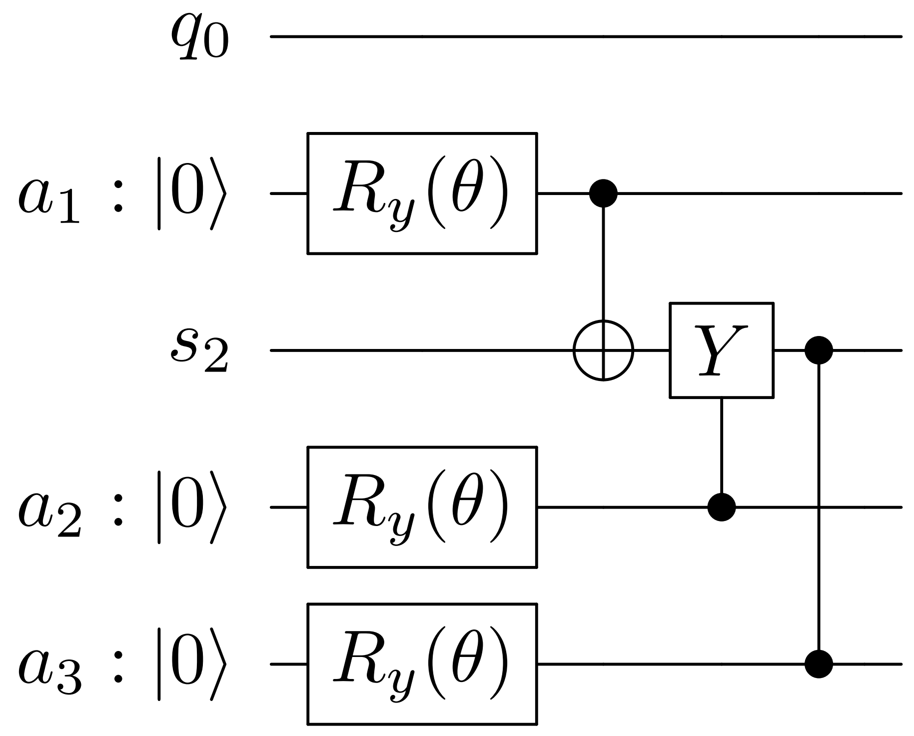




Depolarizing channel

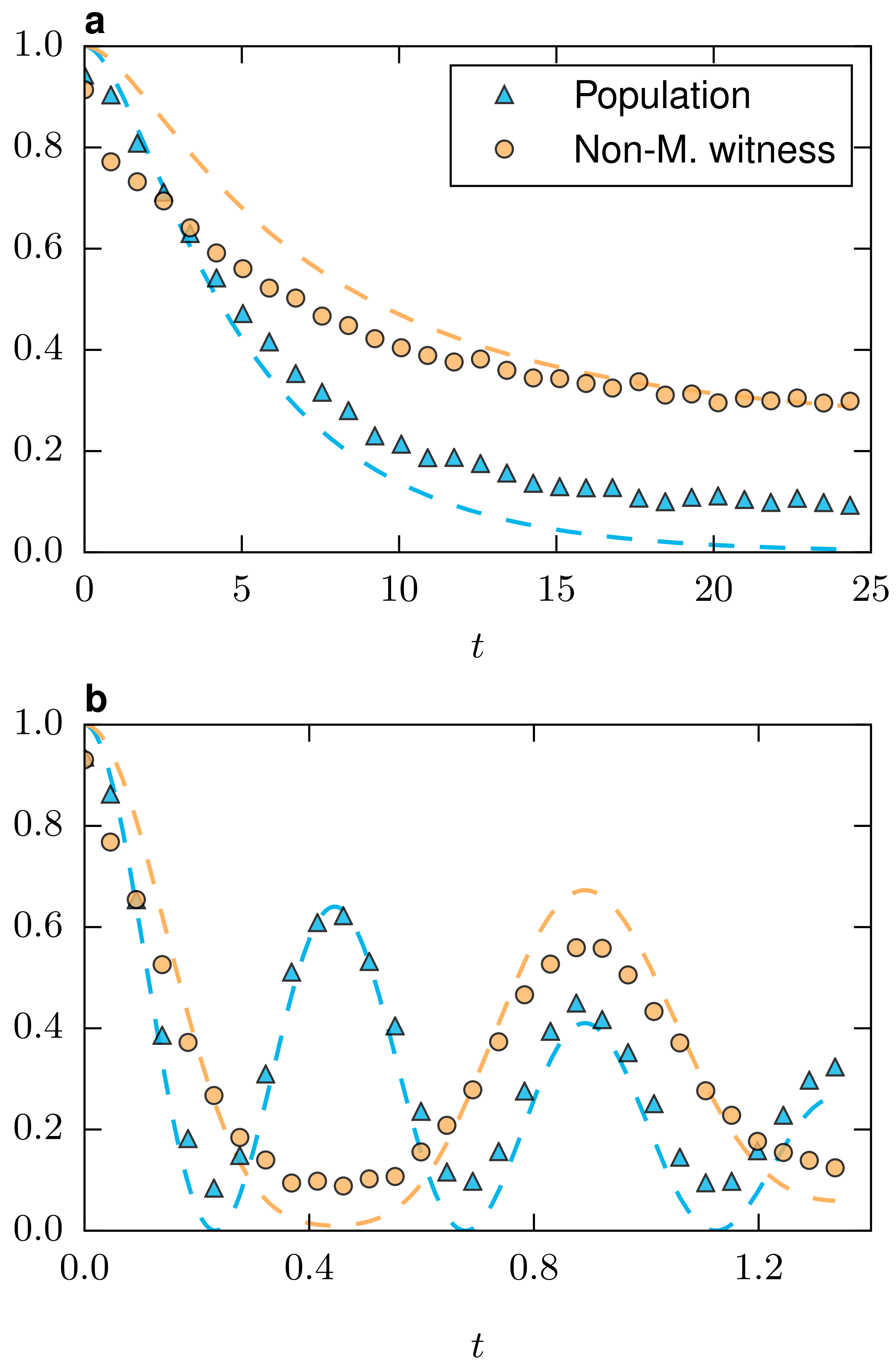




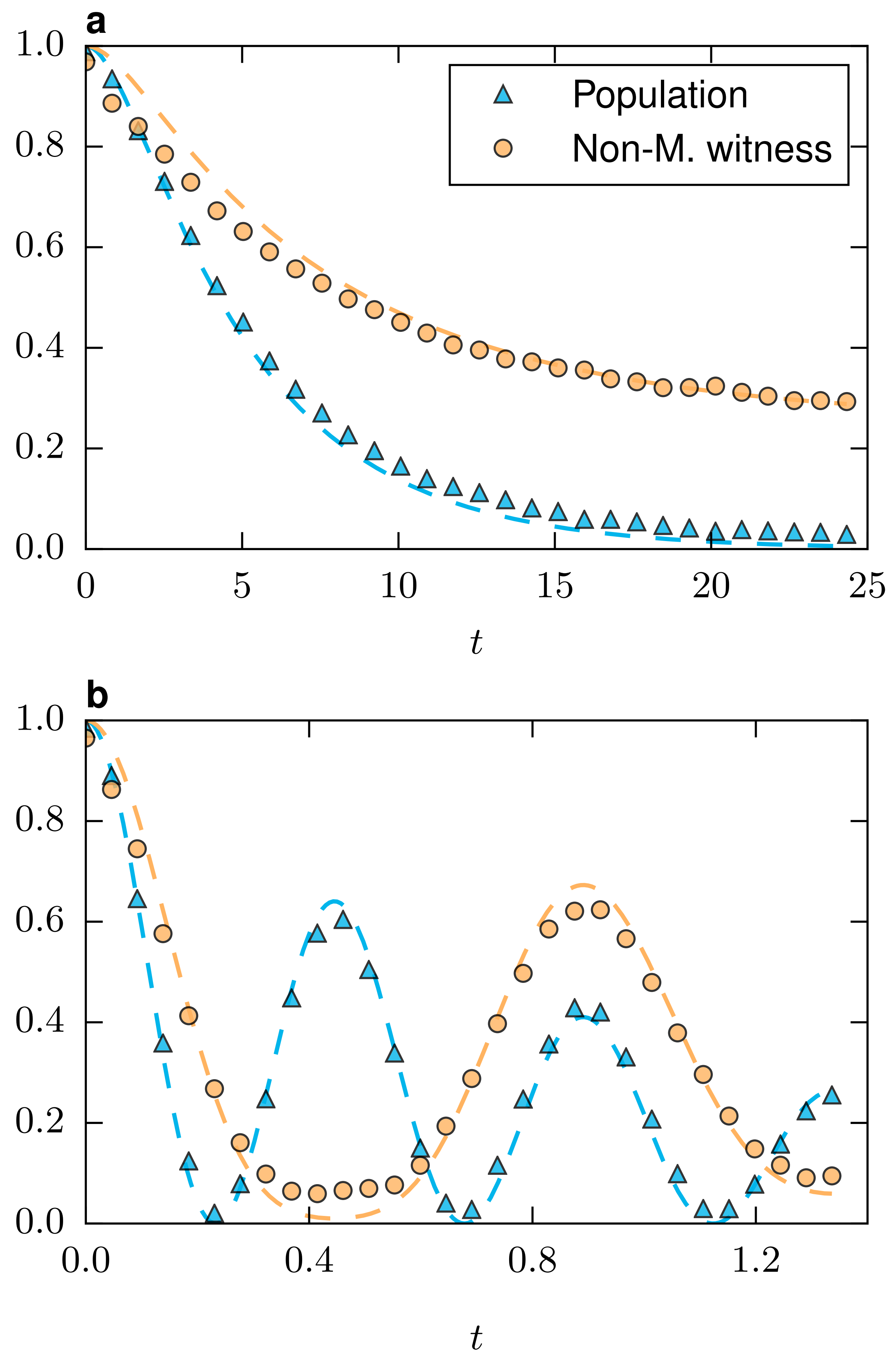


García-Pérez, Rossi, Maniscalco npj Quantum Information 6, 1 (2020)

Simulating open quantum systems

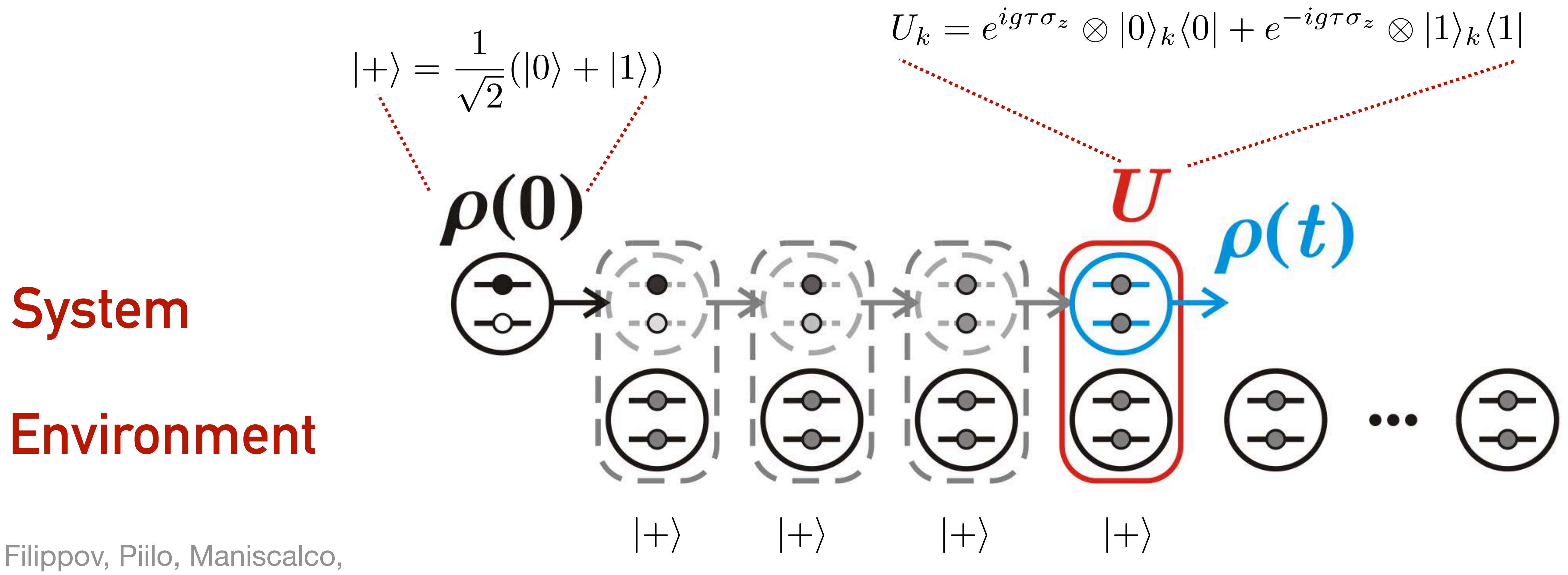


1st submission

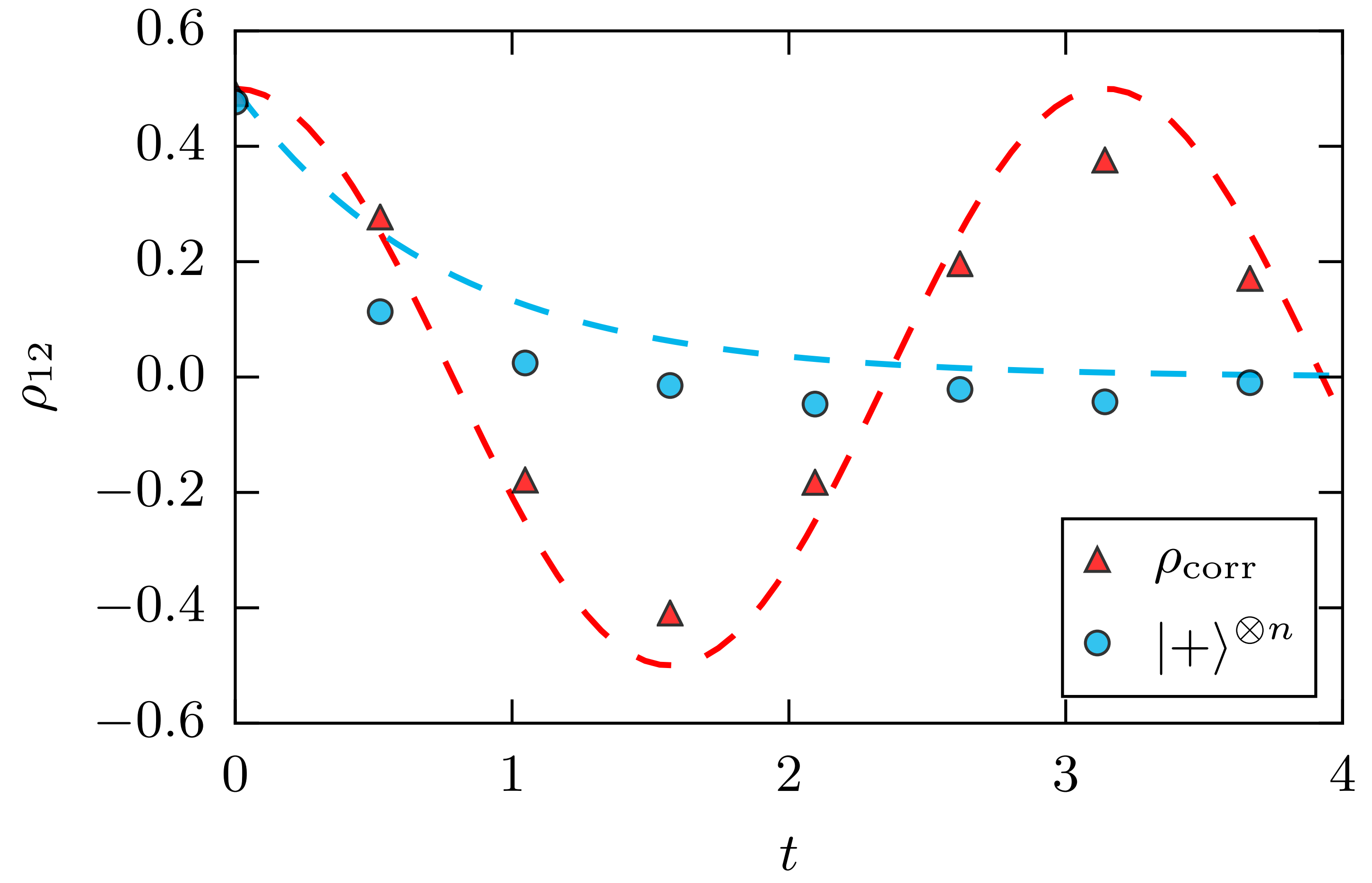


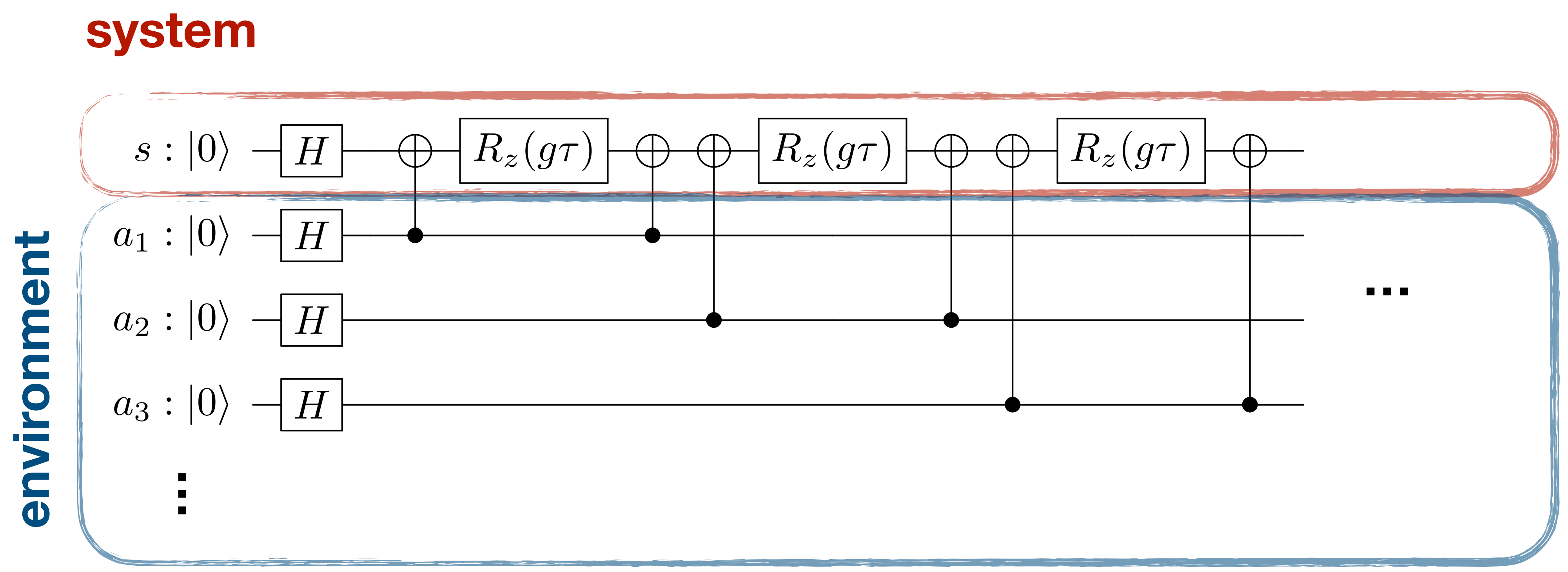
resubmission

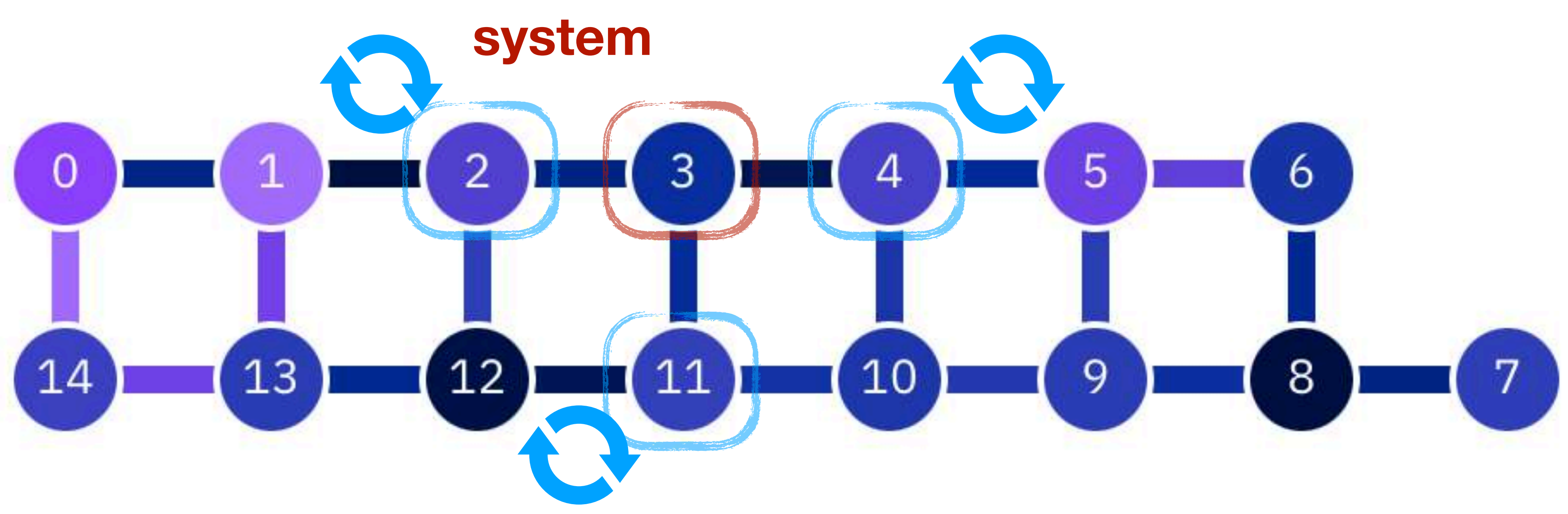
Collisional models

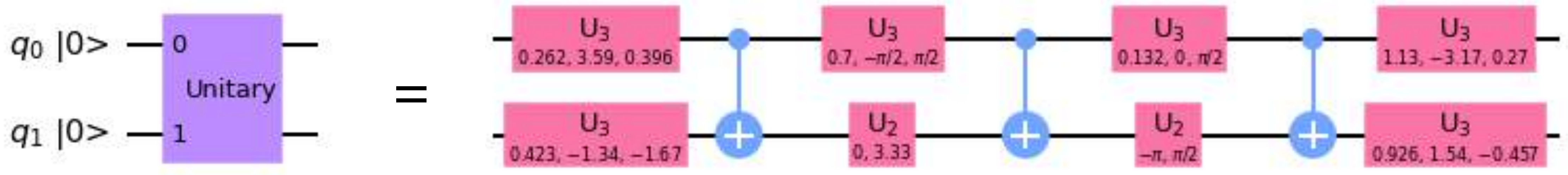


Filippov, Piilo, Maniscalco,
Ziman, PRA 96, 032111 (2017)





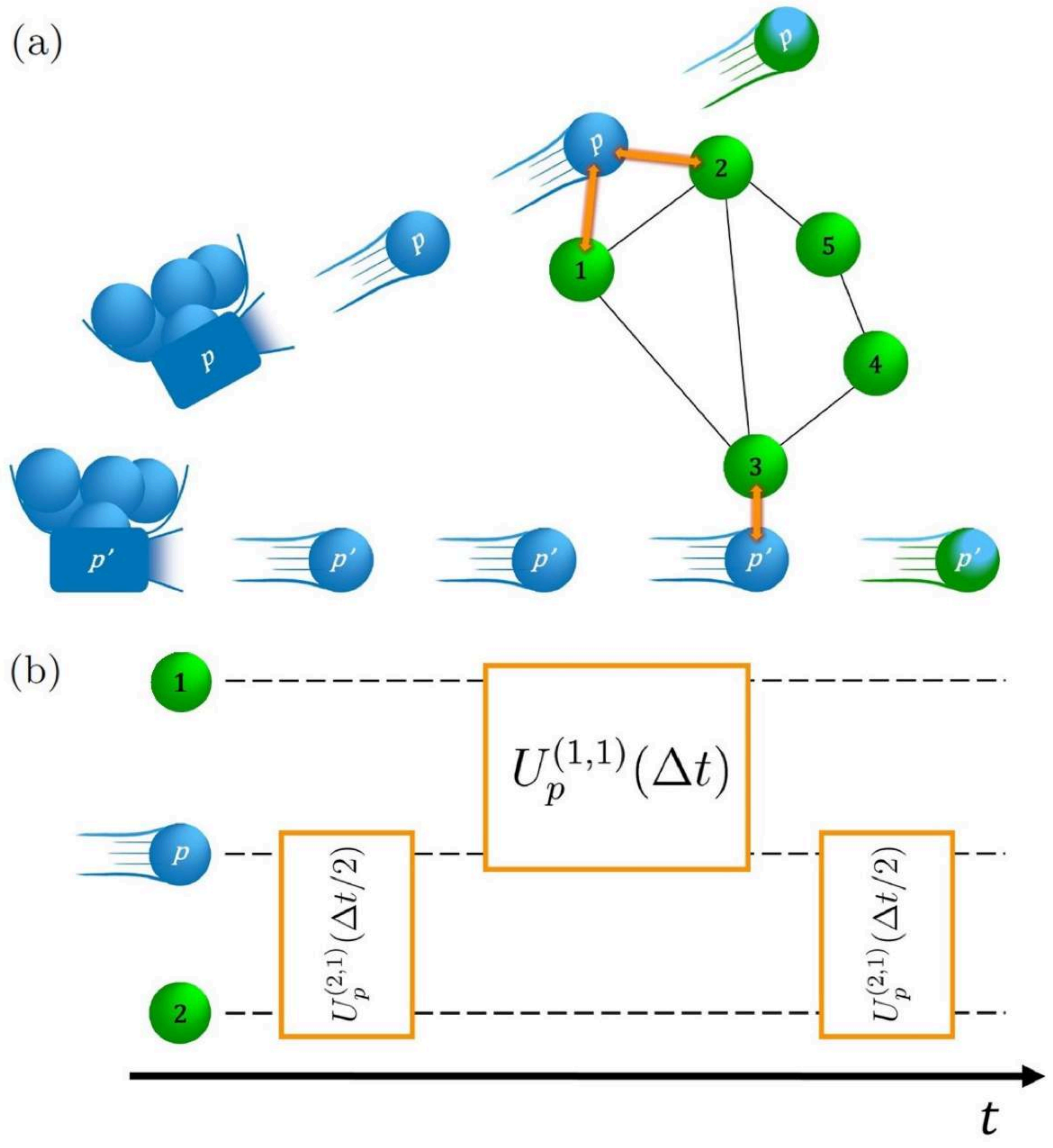




García-Pérez, Rossi, Maniscalco npj Quantum Information 6, 1 (2020)

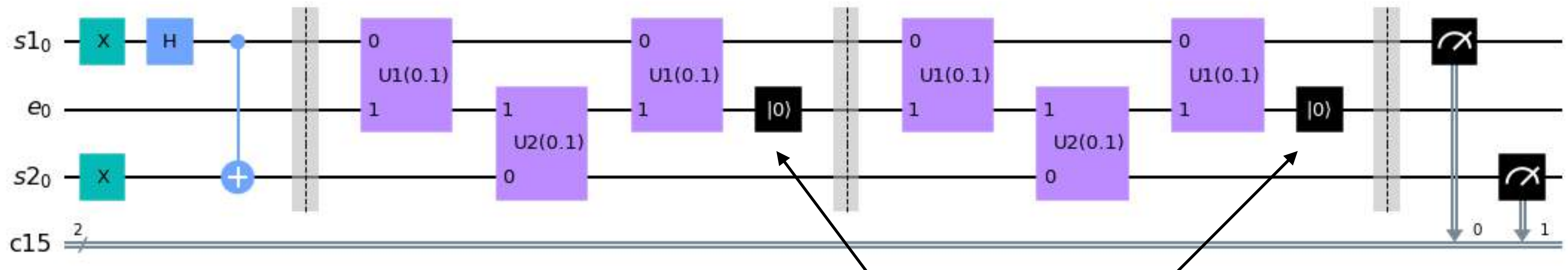
Collisional model

- IBM introduced conditional qubit resets in Nov 2020.
- An X gate is applied if the qubit is measured in $|1\rangle$
- We can reset the state of neighbouring ancillas: no need to swap them with fresh ones!
- We are trying to use this for multipartite collisional models



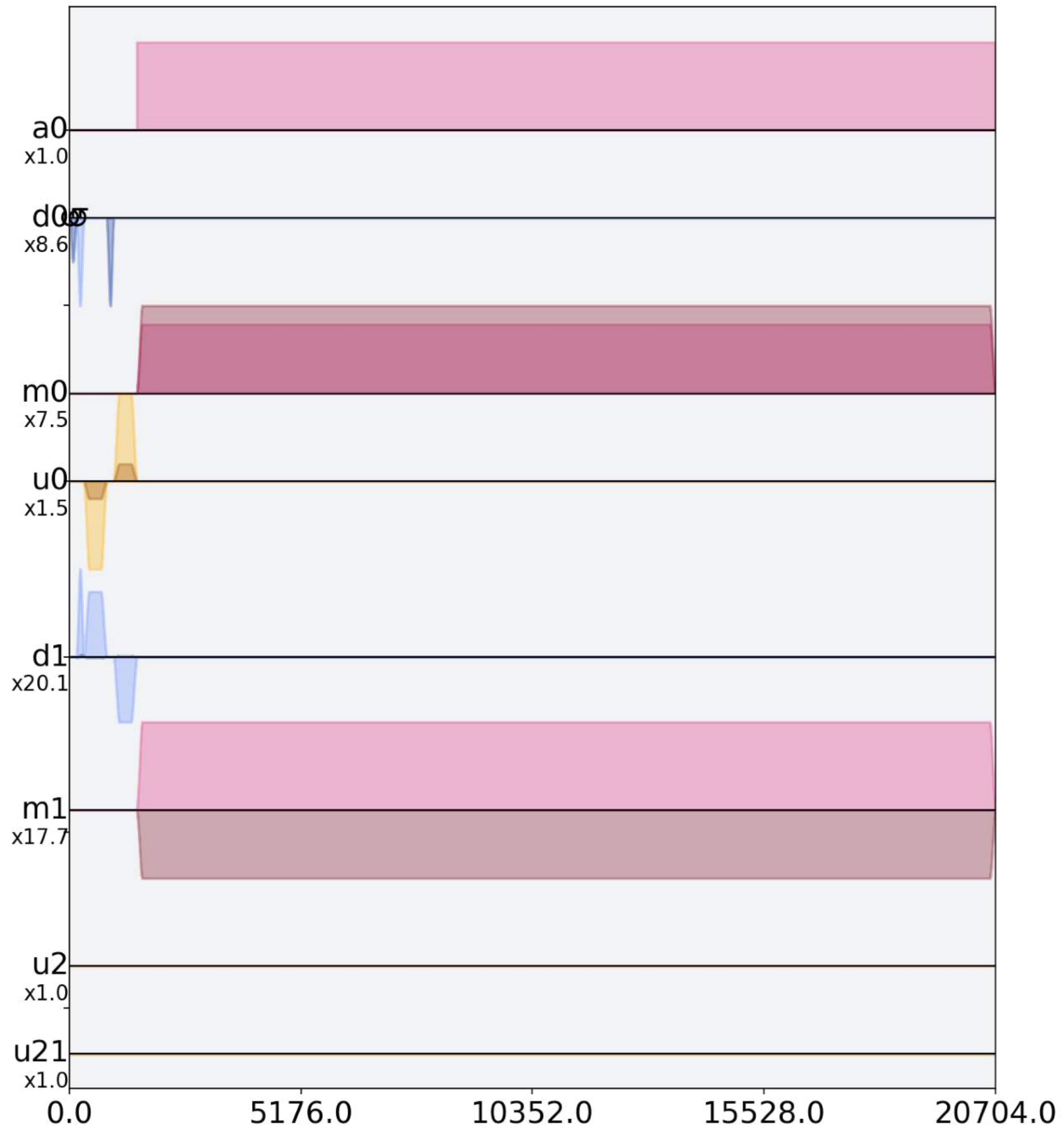
Marco Cattaneo, De Chiara, Maniscalco, Zambrini, Giorgi, arXiv:2010.13910

Collisional models

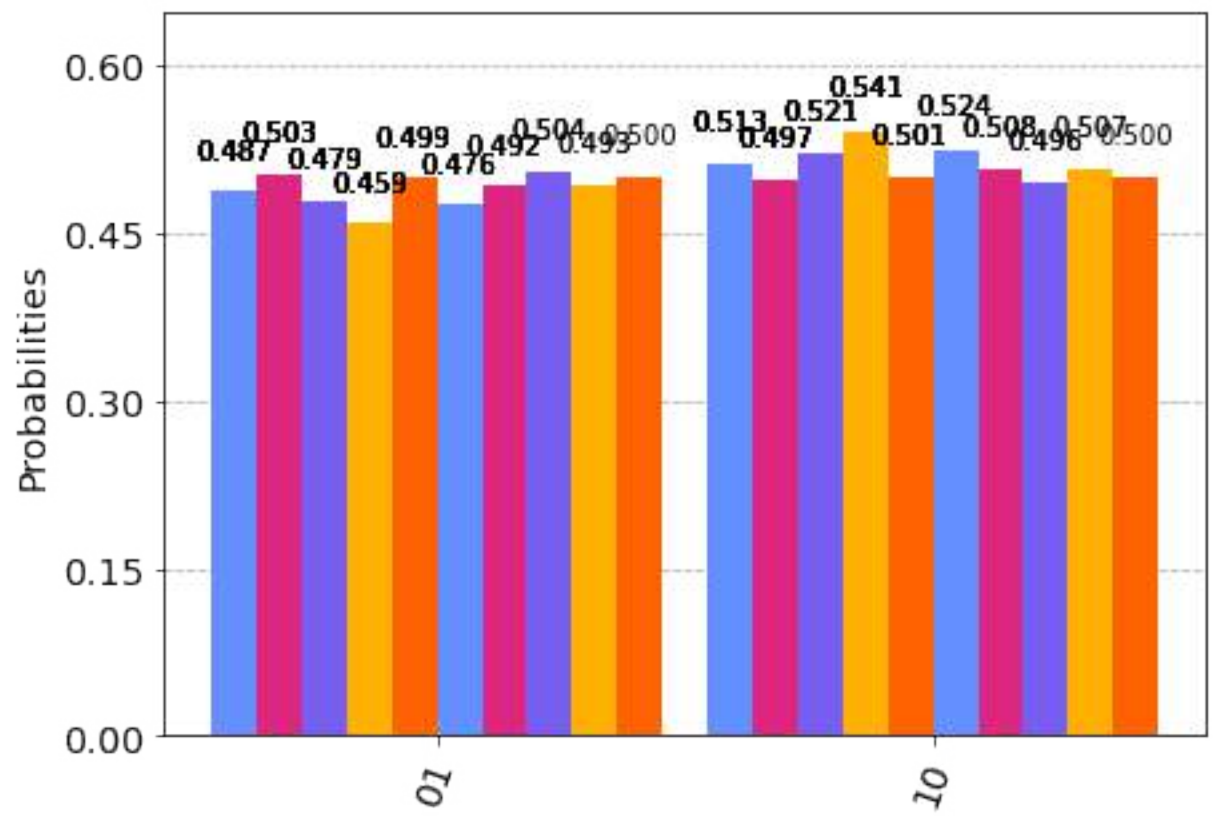


Reset

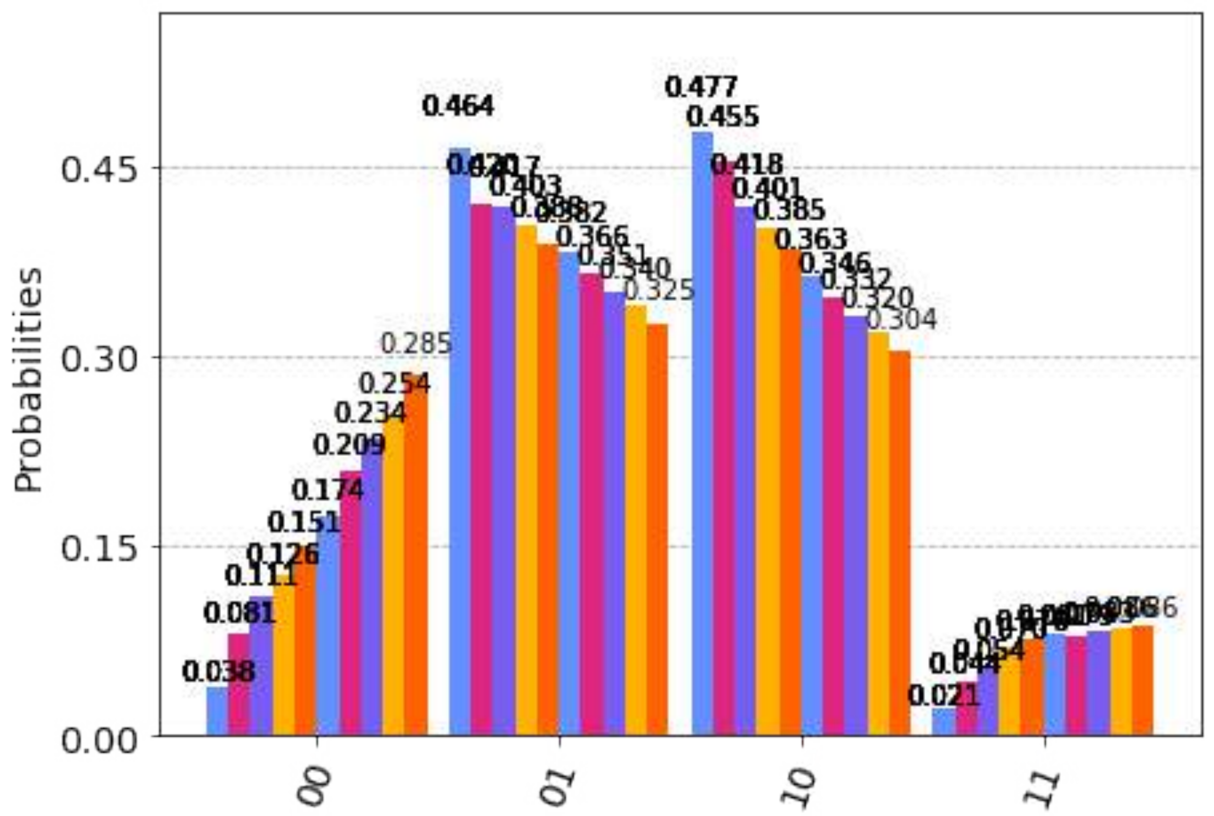
$$|\psi_-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$



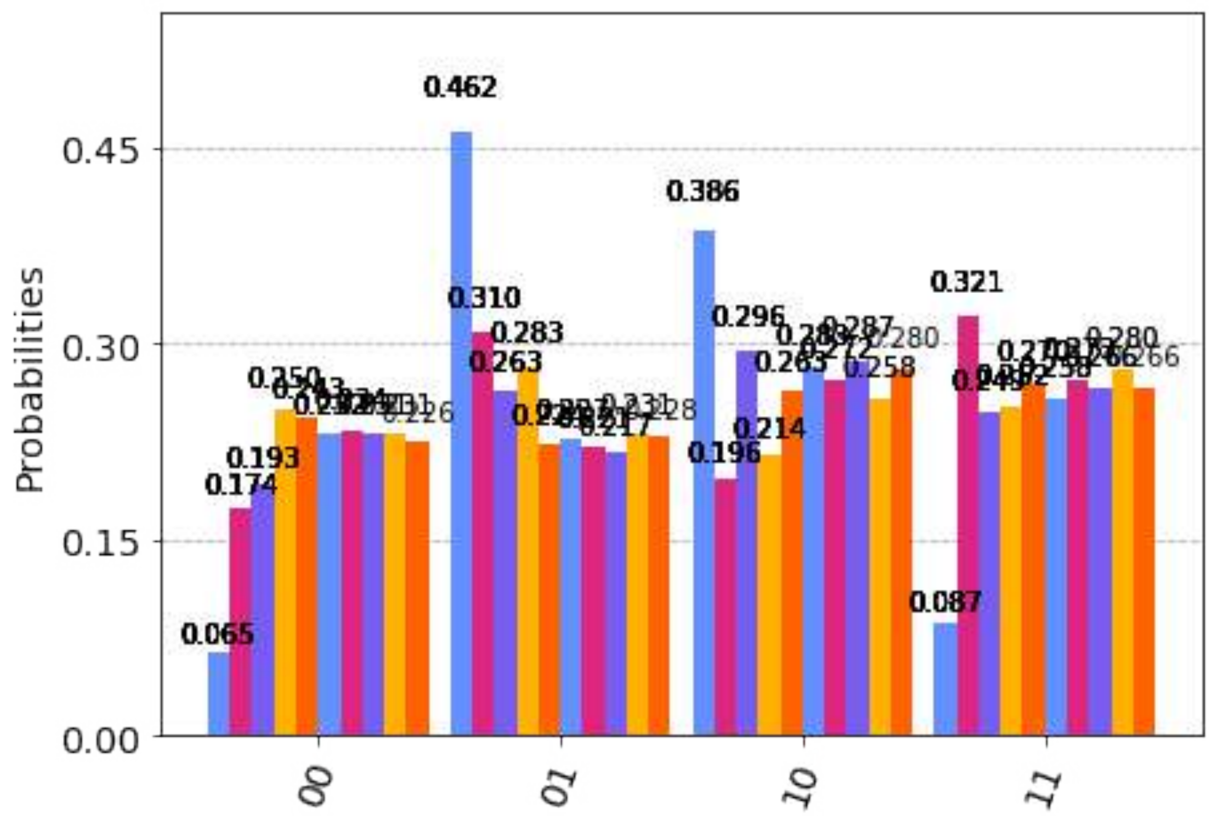
Noiseless simulation



Noisy simulation

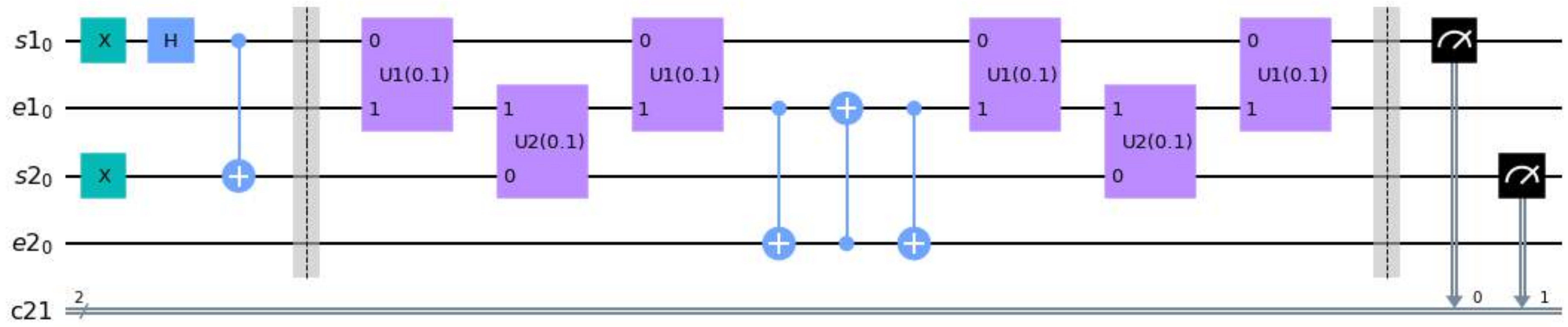


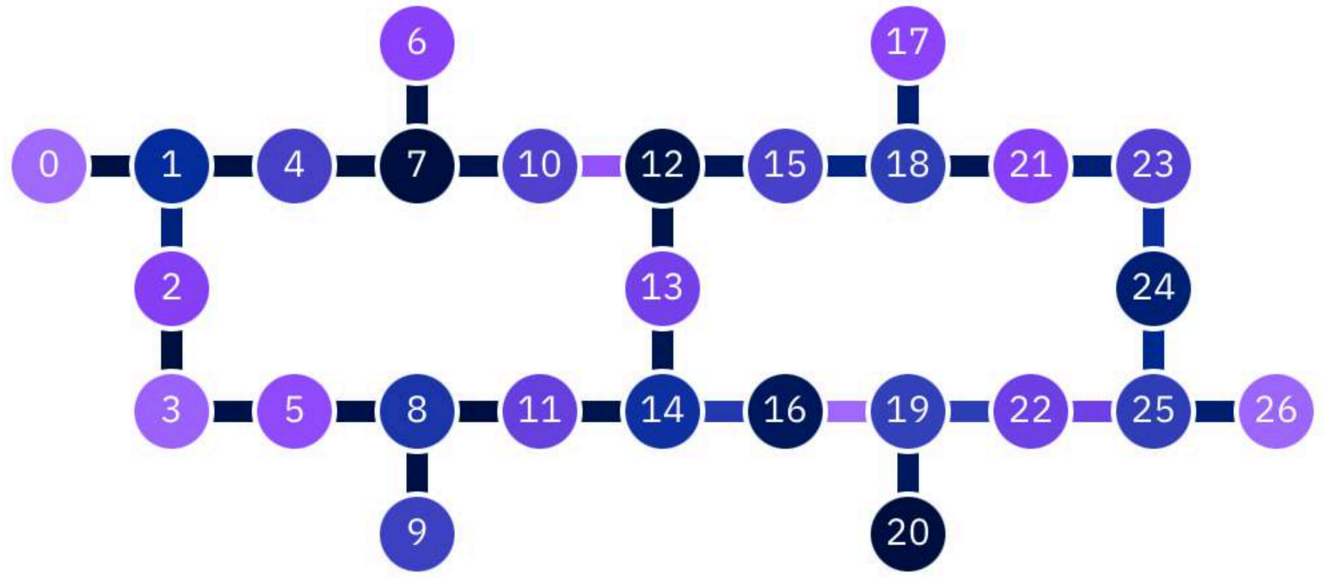
ibmq_bogota



Marco Cattaneo et al. (in progress)

Collisional models

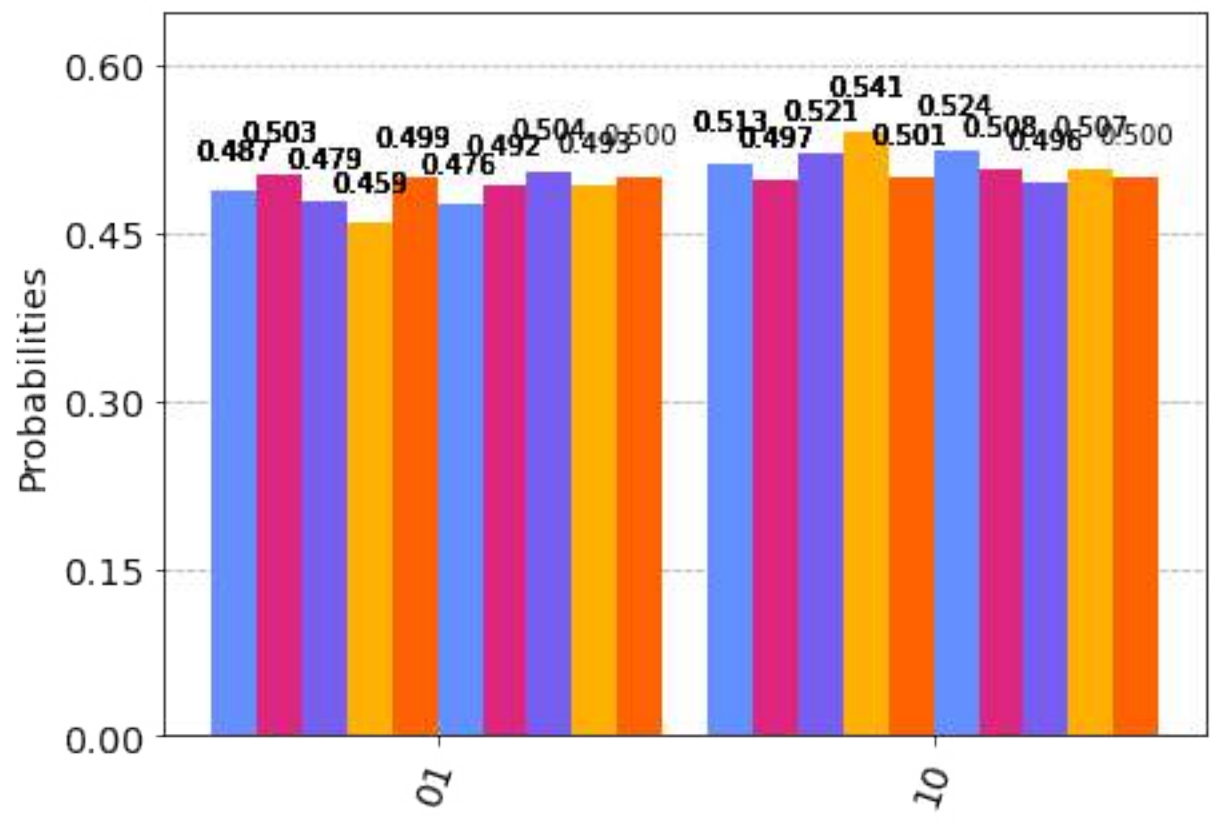


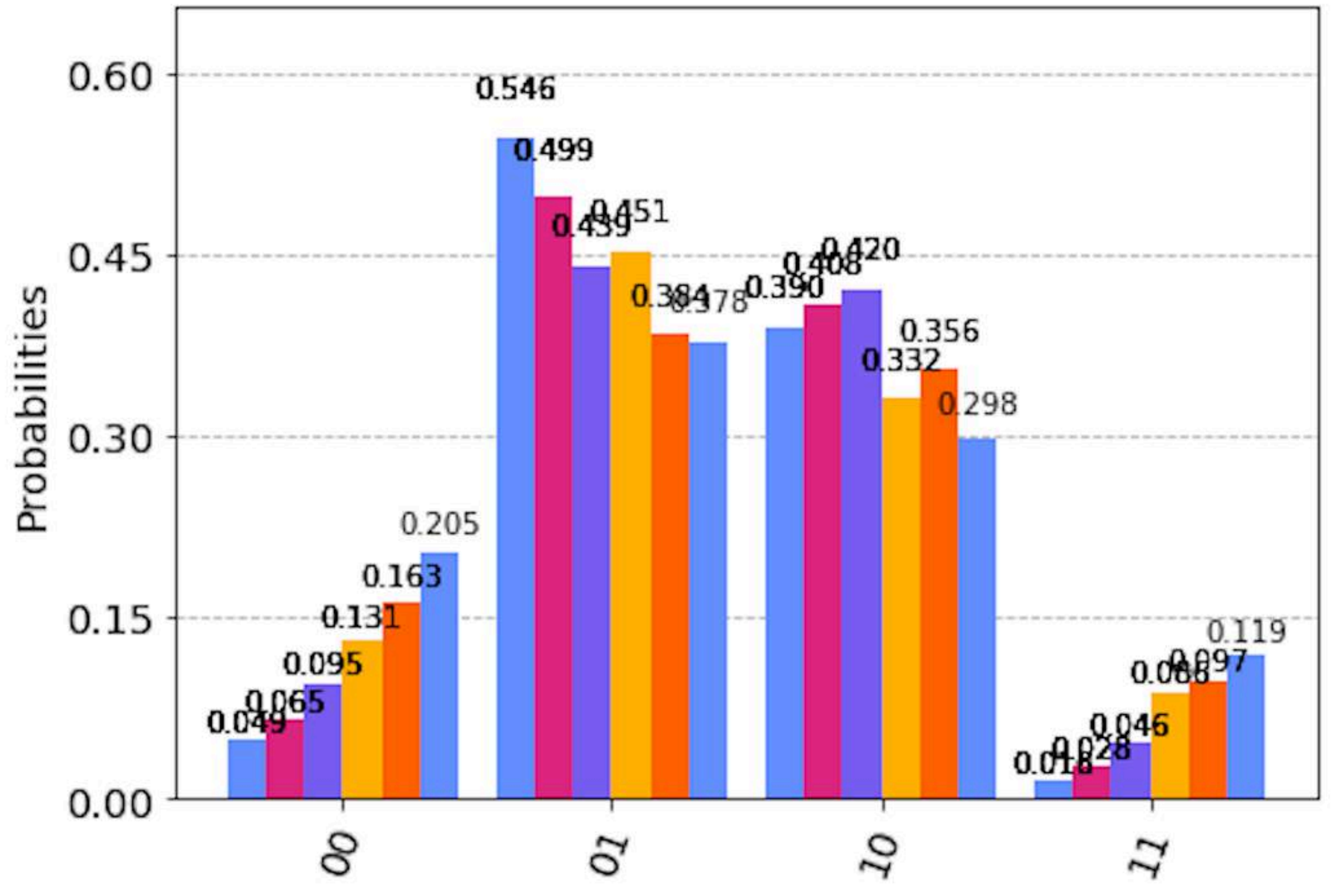


$$|\psi_-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

ibmq_toronto

Noiseless simulation





Marco Cattaneo et al. (in progress)

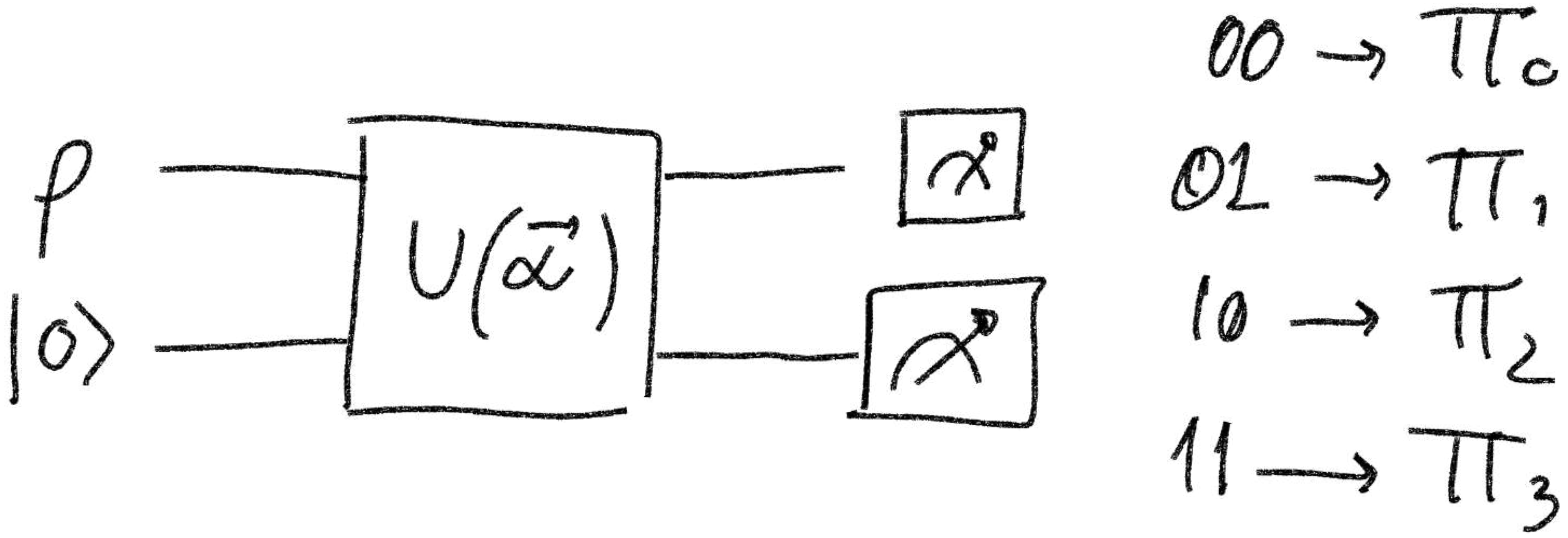
POVMs

- Generalized measurement with positive-valued operators

$$\{\Pi_0, \Pi_1, \Pi_2, \Pi_3\} \qquad\qquad \sum_i \Pi_i = \mathbb{1}$$

If they span $\mathscr{L}(\mathscr{H})$ it is informationally complete (d^2 operators)

- Can be implemented using dilation

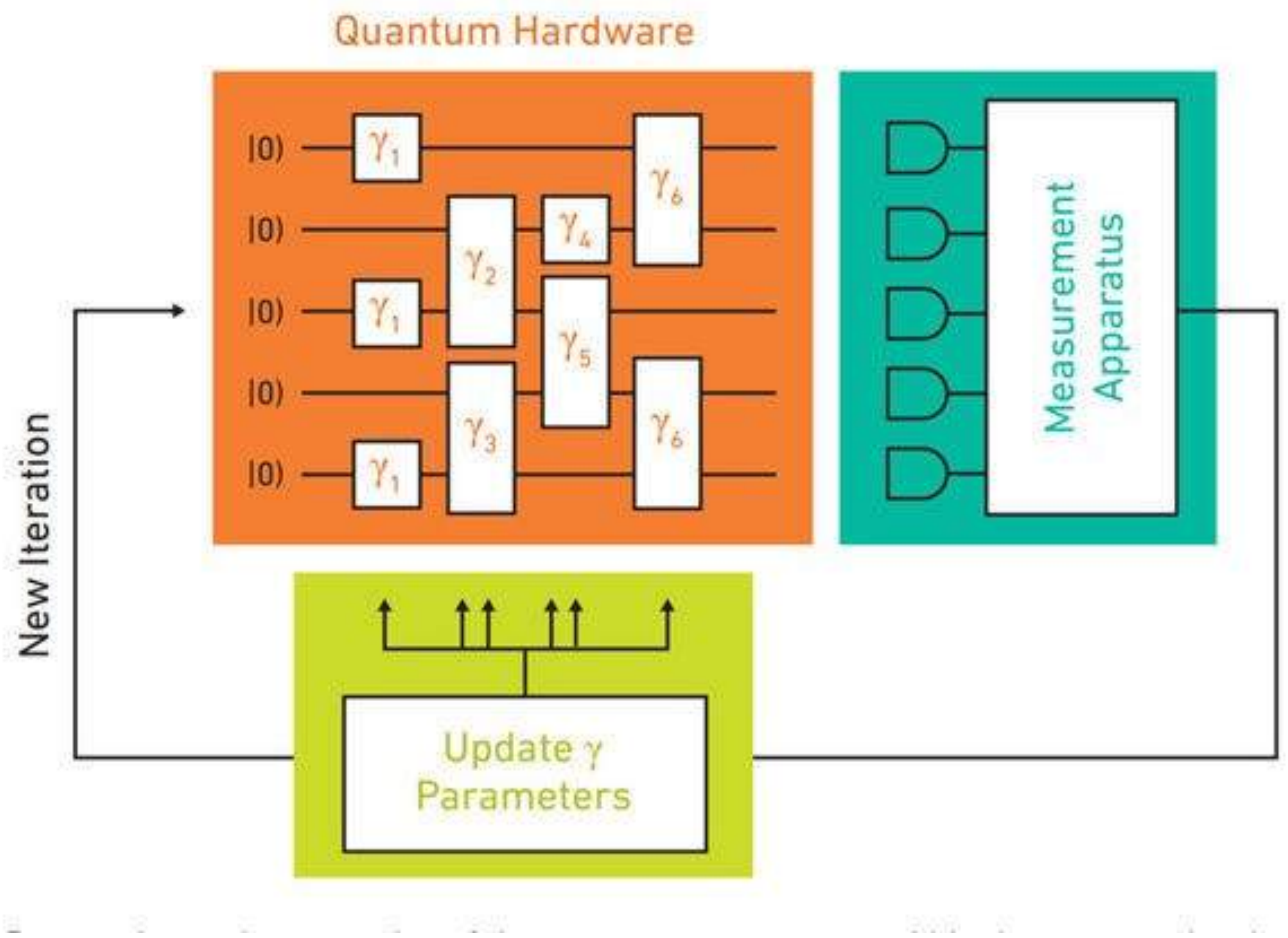


$$\{\Pi_i(\vec{\alpha})\} \longleftrightarrow U(\vec{\alpha})$$

POVMs and VQE

- Variational Quantum Eigensolver: find an approximate ground state energy

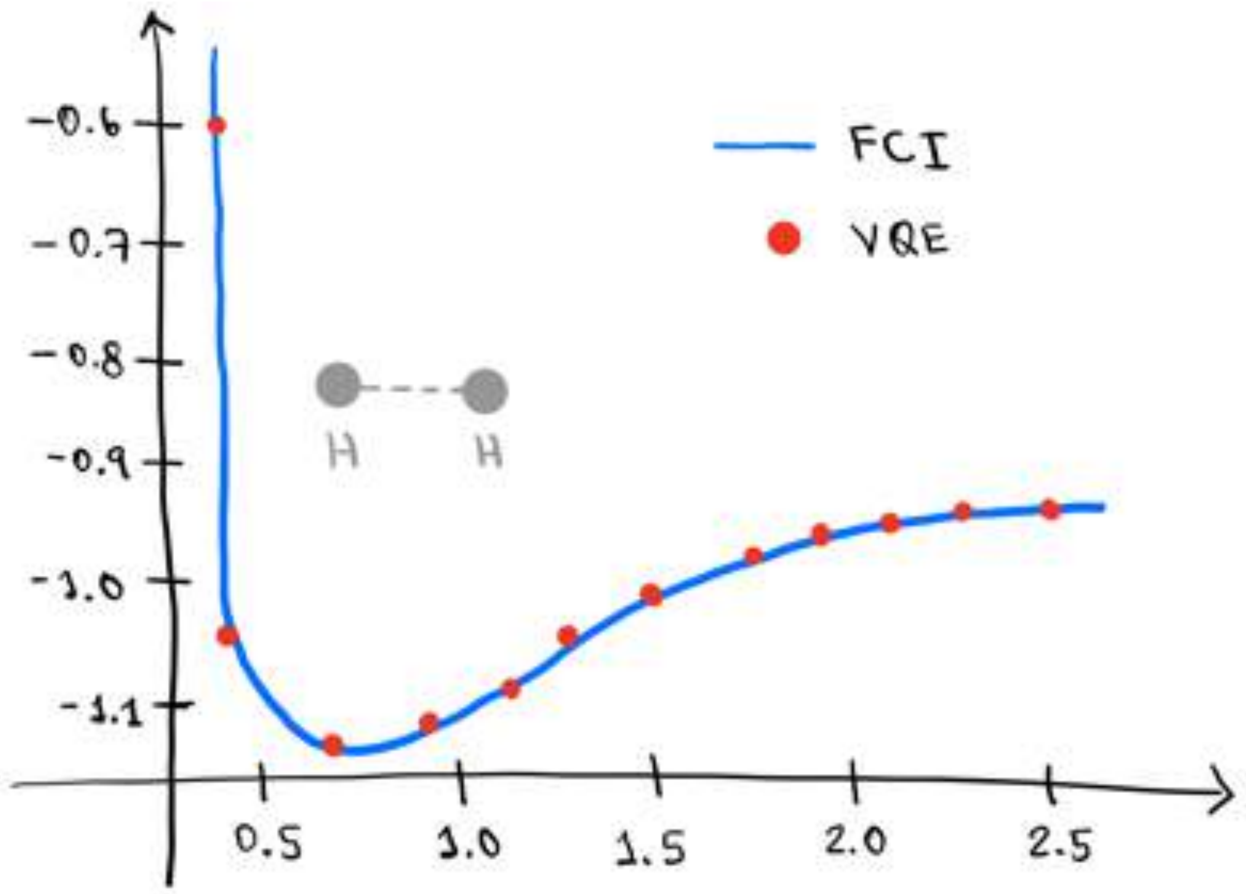
$$\lambda_{min} \leq \lambda_\theta \equiv \langle \psi(\theta) | H | \psi(\theta) \rangle$$



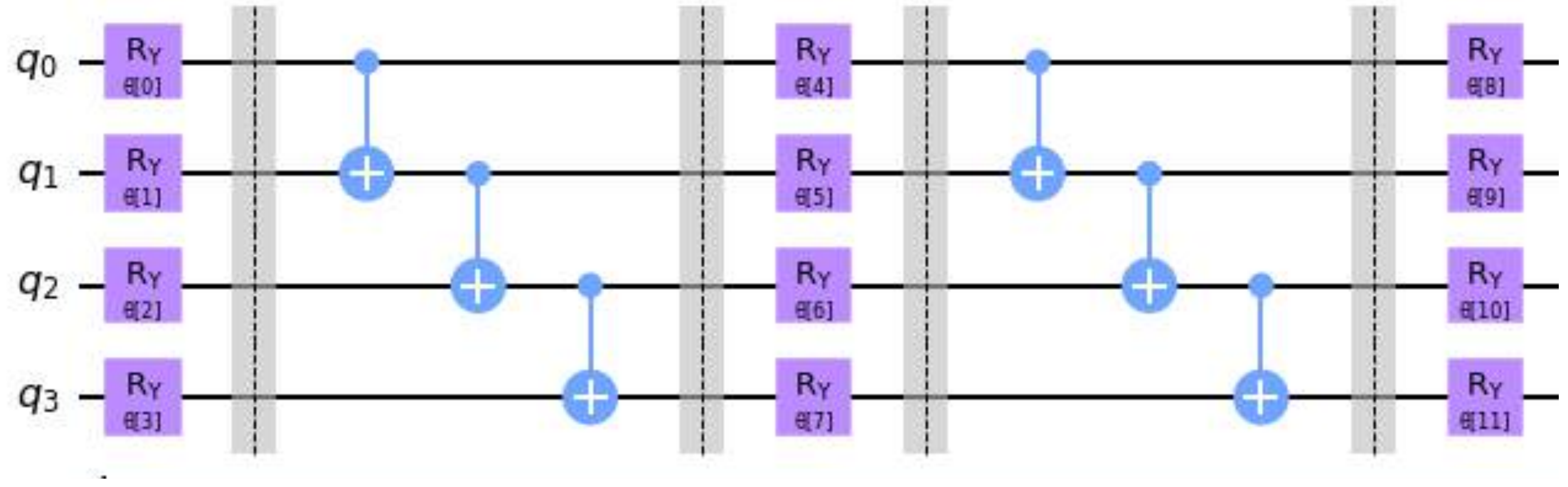
- Hybrid quantum-classical algorithm:
  - Quantum computer: state preparation and measurement
  - Classical computer: parameter optimization

Turku + IBM Research Zurich arXiv:2104.00569

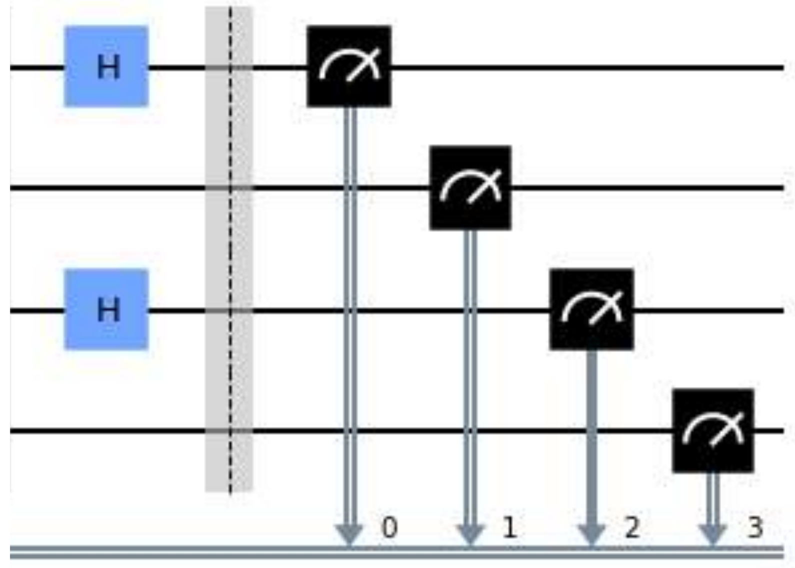
POVMs and VQE



We choose an ansatz state that depends on parameters $\vec{\theta}$



$\psi(\vec{\theta})$



We evaluate the expectation value of the energy

$P_1 = \sigma_x \otimes \sigma_z \otimes \sigma_x \otimes \sigma_z$

$$\langle H \rangle_{\psi(\theta)} = \sum_k c_k \langle P_k \rangle_{\psi(\theta)}$$

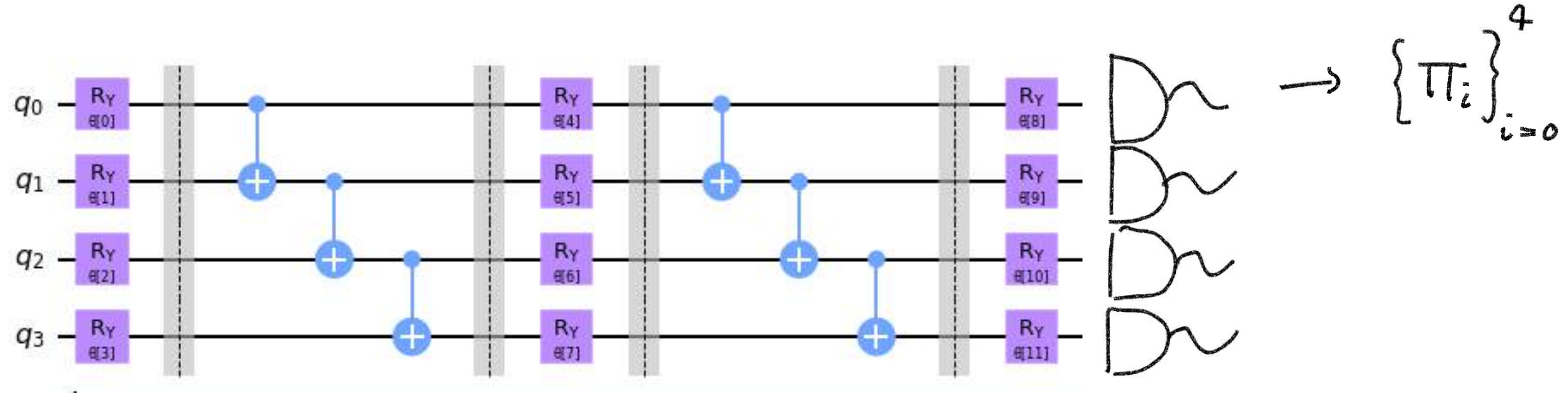
Measurement problem: the number of Pauli strings to be measured grows very badly with the size of the Hamiltonian, and hence the number of shots required to reach chemical accuracy

Solutions: grouping of Pauli strings that can be measured at once (with marginalization), machine learning, etc…

Turku + IBM Research Zurich arXiv:2104.00569

POVMs and VQE

Our proposal: perform local IC-POVM on each qubit



$$\langle H \rangle_{\psi(\theta)} = \sum_k \langle P_k \rangle_{\psi(\theta)} = \sum_{\vec{m}} w_{\vec{m}} p_{\vec{m}}$$

$\vec{m}$ is a sequence of outcomes

$w_{\vec{m}}$ is the corresponding weight in H (can be calculated efficiently)

The expectation value of the energy is simply a Montecarlo sampling

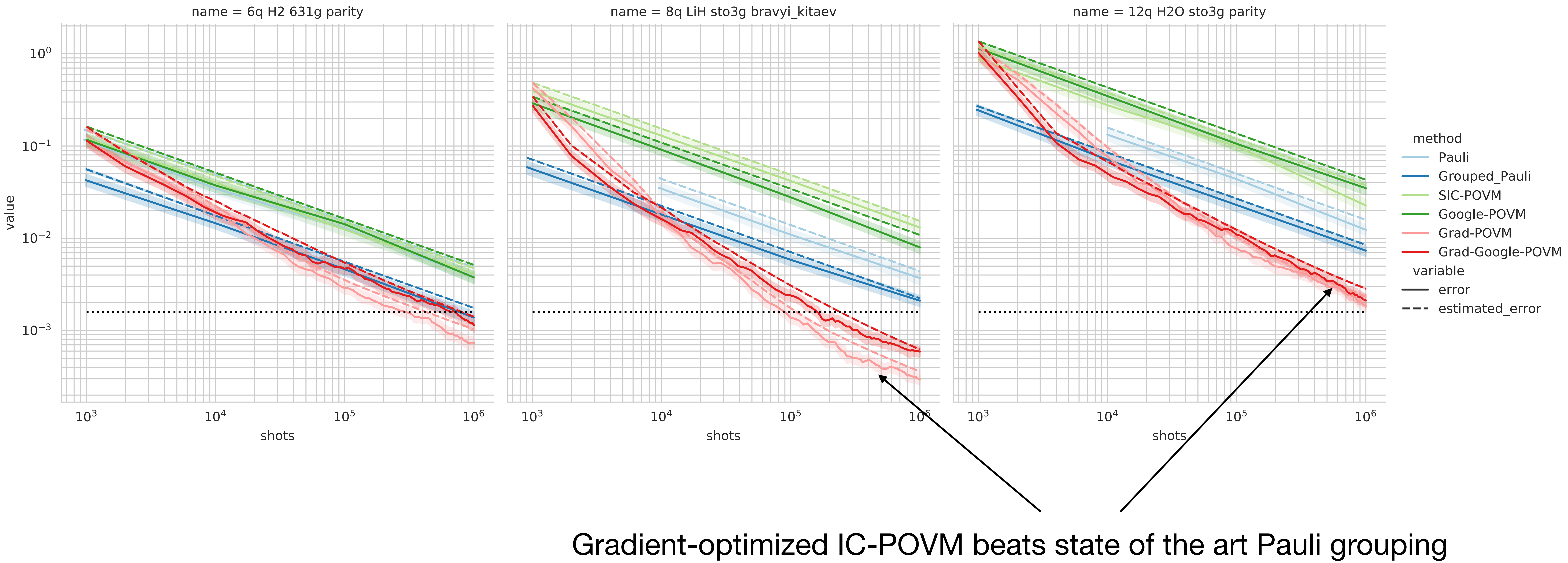
Moreover, we have informationally complete data (useful for e.g. partial tomography, quantum subspace expansion etc.)

The POVM operators can be optimized to increase the precision!

Turku + IBM Research Zurich arXiv:2104.00569

POVMs and VQE

Results



Turku + IBM Research Zurich arXiv:2104.00569

Recap

- Introduction to the limitations of superconducting quantum devices
- How to get the best results from IBM Q devices
- Two research examples in OQS and mesurements

- The devices improve fast. Things that may not work now may work in a few months
- New possiblities offered by the pulse level control

Thank you!