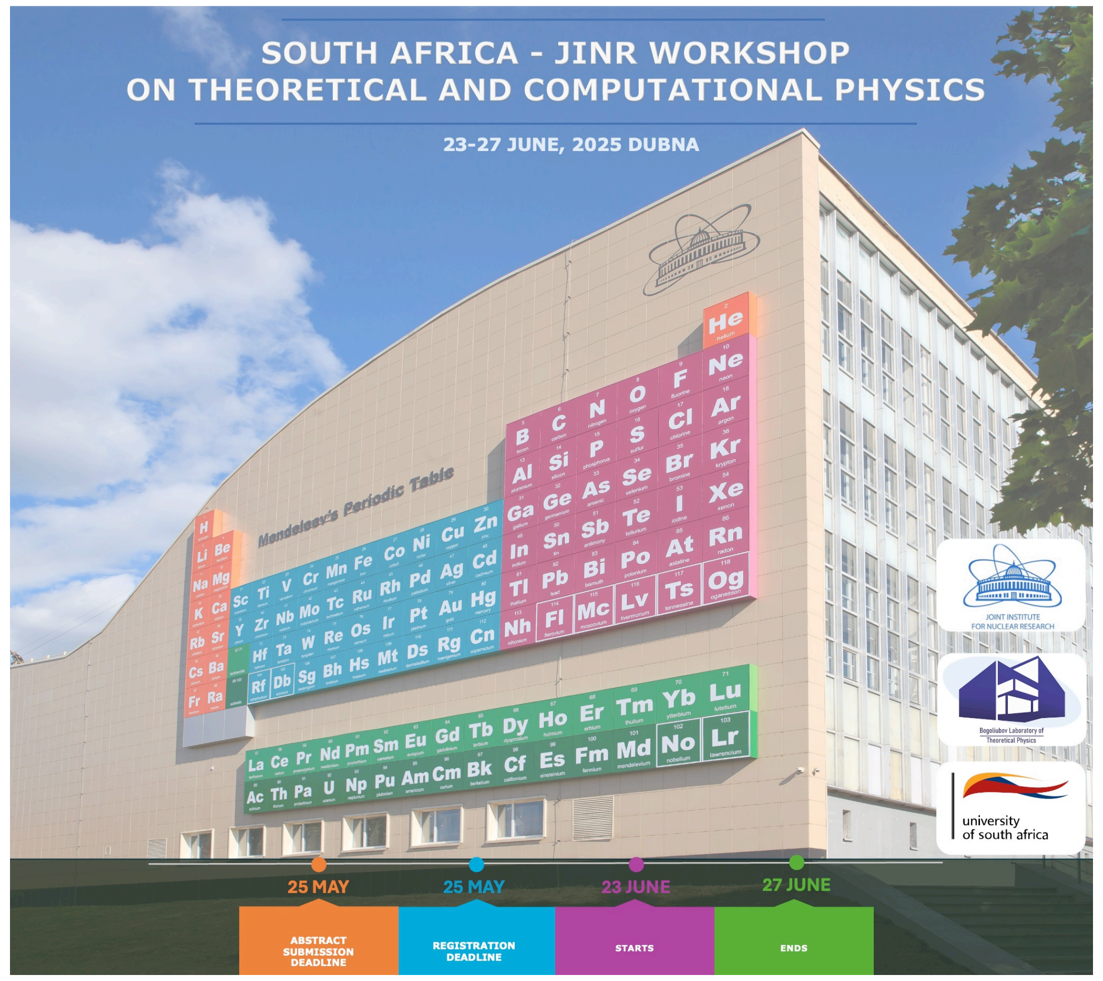
UNIVERSITY OF KWAZULU-NATAL™
INYUVESI YAKWAZULU-NATALI

NITheCS
National Institute for
Theoretical and Computational Sciences

An Introduction to Quantum Computing and Quantum Machine Learning with Quantum Parametric Circuits



SOUTH AFRICA - JINR WORKSHOP
ON THEORETICAL AND COMPUTATIONAL PHYSICS

23-27 JUNE, 2025 DUBNA

25 MAY ABSTRACT SUBMISSION DEADLINE
25 MAY REGISTRATION DEADLINE
23 JUNE STARTS
27 JUNE ENDS

Mendeleev's Periodic Table
JOINT INSTITUTE FOR NUCLEAR RESEARCH
Bogoliubov Laboratory of Theoretical Physics
university of south africa

Ilya Sinayskiy

24 June 2025

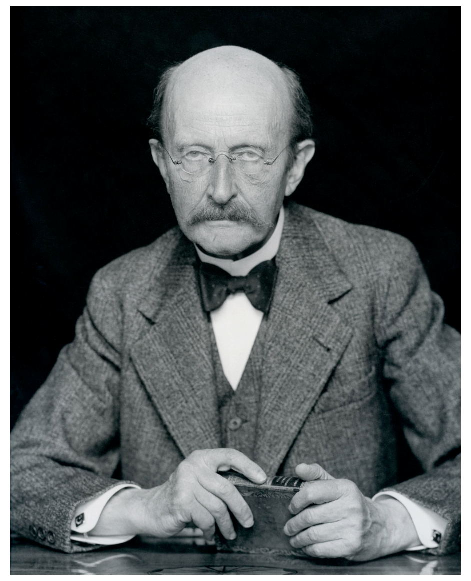
SA QuTI
South African Quantum Technology Initiative

Outline

- History of Quantum Computing
- Universal error correcting fault -tolerant quantum computers vs NISQ devices
- Basic of Quantum Computations
- QPC examples VQE/VQLS/QAOA/QCNN
- Application of architectural search to Ansatz design

Quantum Computers

Quantum Mechanics:
early 1900



Computer Science:
1930

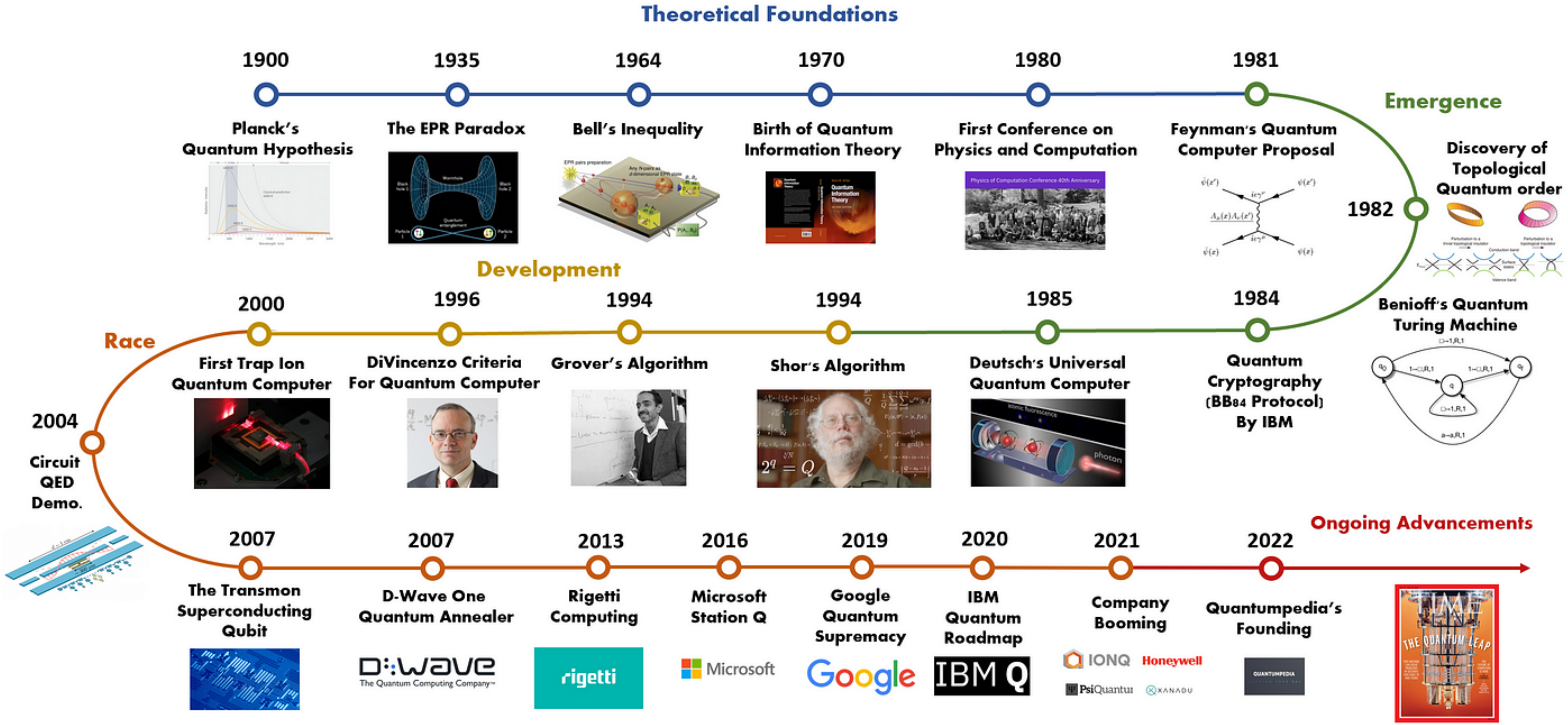
Richard P. Feynman

Yuri I. Manin (Юрий Манин)

Yu. I. Manin, “Computable and Non-Computable,” Sovetskoe Radio, Moscow, 1980, 128 p.

R.P. Feynman, Simulating physics with computers, Internat. J. Theoret. Phys. 21 (6–7) (1982) 467–488, http://dx.doi.org/10.1007/bf02650179.

Quantum Computers



Theoretical Foundations

1900 Planck's Quantum Hypothesis

1935 The EPR Paradox

1964 Bell's Inequality

1970 Birth of Quantum Information Theory

1980 First Conference on Physics and Computation

1981 Feynman's Quantum Computer Proposal

Emergence

1982 Discovery of Topological Quantum order

1984 Benioff's Quantum Turing Machine

1984 Quantum Cryptography (BB84 Protocol) By IBM

Development

1985 Deutsch's Universal Quantum Computer

1994 Shor's Algorithm

1994 Grover's Algorithm

1996 DiVincenzo Criteria For Quantum Computer

2000 First Trap Ion Quantum Computer

Race

2004 Circuit QED Demo.

2007 The Transmon Superconducting Qubit

2007 D-Wave One Quantum Annealer

2013 Rigetti Computing

2016 Microsoft Station Q

2019 Google Quantum Supremacy

2020 IBM Quantum Roadmap

2021 Company Booming

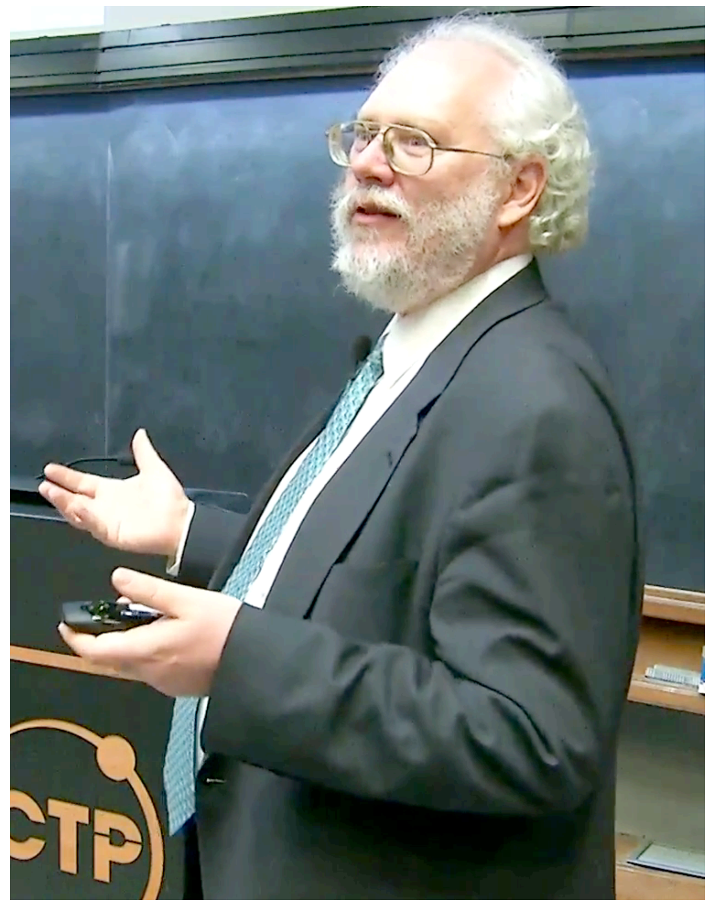
2022 Quantumpedia's Founding

Ongoing Advancements

https://quantumpedia.uk/a-brief-history-of-quantum-computing-e0bbd05893d0

Quantum Computers

Shor's Algorithm



Shor, P.W. (1994). "Algorithms for quantum computation: Discrete logarithms and factoring". Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 124–134. doi:10.1109/sfcs.1994.365700.

Shor, Peter W. (October 1997). "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". SIAM Journal on Computing. ***26*** *(5): 1484–1509. arXiv:quant-ph/9508027. doi:10.1137/S0097539795293172. 2337707.*

Shor's Code - QEC

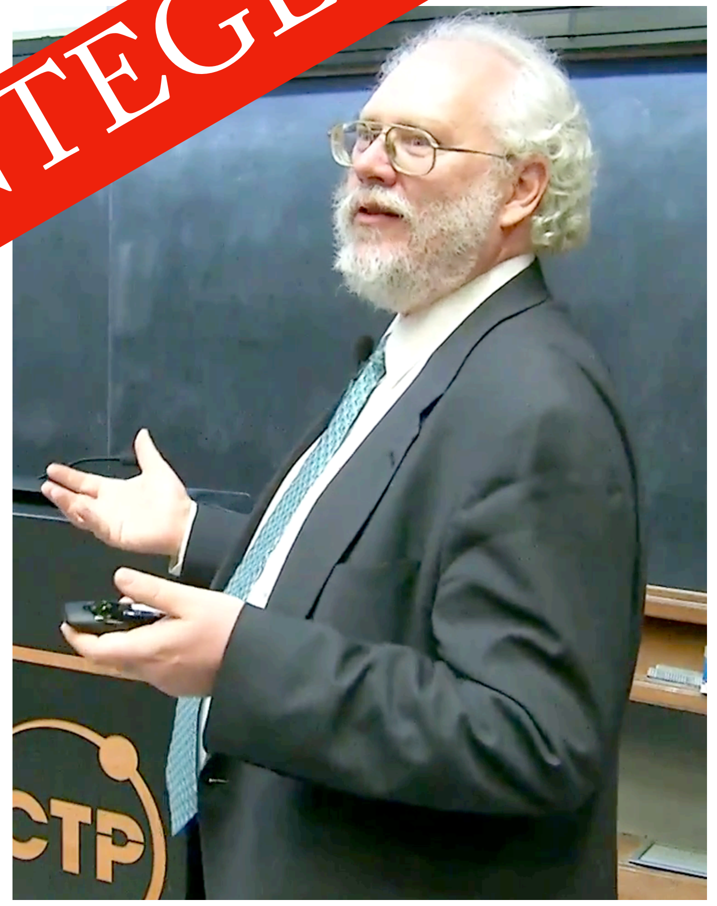
Shor, Peter W. (1995). "Scheme for reducing decoherence in quantum computer memory". Physical Review A. ***52*** *(4): R2493 – R2496*

Fault-tolerant quantum computation

Shor, Peter W. (1997). "Fault-tolerant quantum computation" quant-ph/9605011;
37th Symposium on Foundations of Computing, IEEE Computer Society Press, 1996, pp. 56-65

Quantum Computers

Shor's Algorithm



Shor, P.W. (1994). "Algorithms for quantum computation: Discrete logarithms and factoring". Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 124–134. doi:10[illegible] sfcs.1994.365700.

*Shor, Peter W. (October 1997). "Polynomial-Time Algorithms for Prime Factor[illegible]d Discrete Logarithms on a Quantum Computer". SIAM Journal on Computing. **26** [illegible]–1509. arXiv:quant-ph/9508027. doi:10.1137/S0097539795293172. 23[illegible]*

Shor's Code - QEC

*Shor, Peter W. (1995). "Scheme f[illegible] decoherence in quantum computer memory". Physical Review A. **52** (4): R2493 – R2496*

Fault-t[illegible]nt quantum computation

Shor, [illegible] (1997). "Fault-tolerant quantum computation" quant-ph/9605011;
[illegible]posium on Foundations of Computing, IEEE Computer Society Press, 1996, pp. 56-65

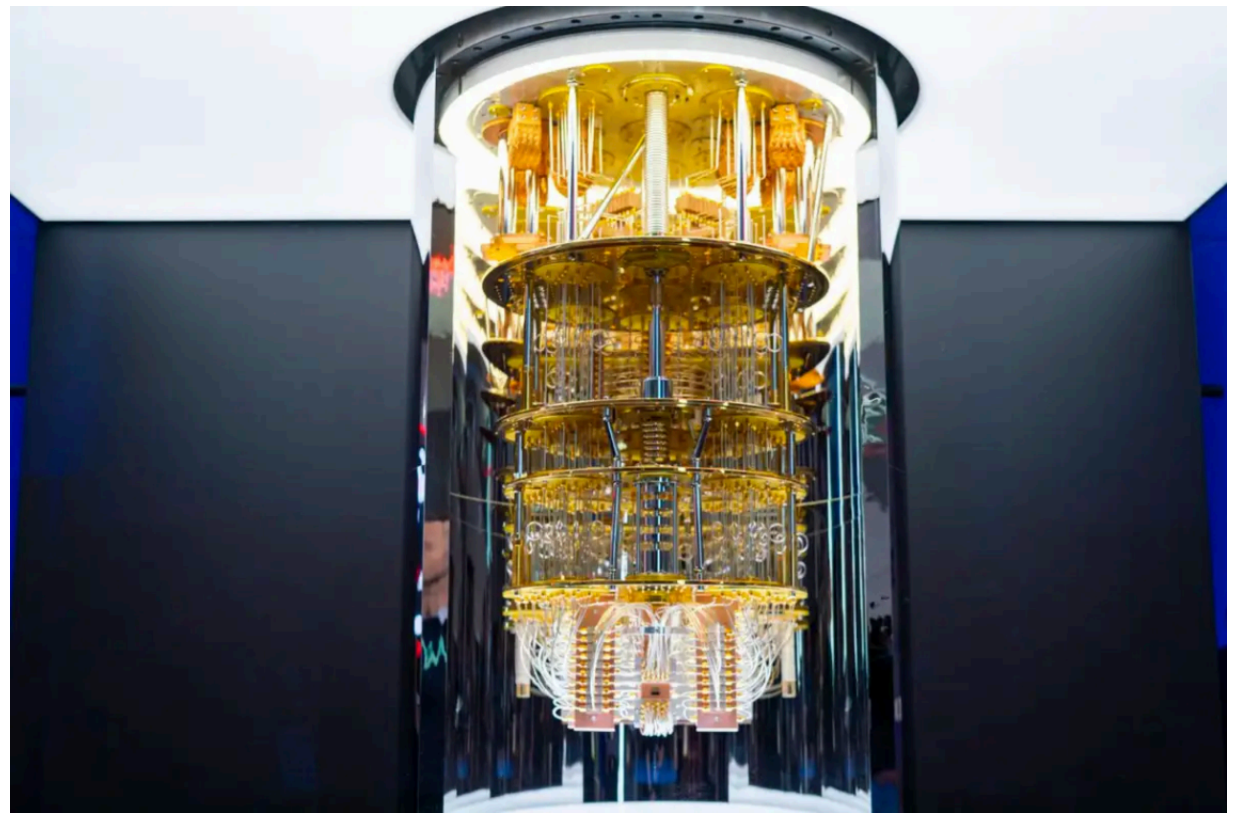
NEED MILLIONS PHYSICAL QUBITS TO FACTORISE 2048bit INTEGER

Quantum Computers - Current Status (NISQ)

Google - Willow Chip (2024)

105 qubit + some QEC

IBM - Condor Chip (2025)



1121 qubit

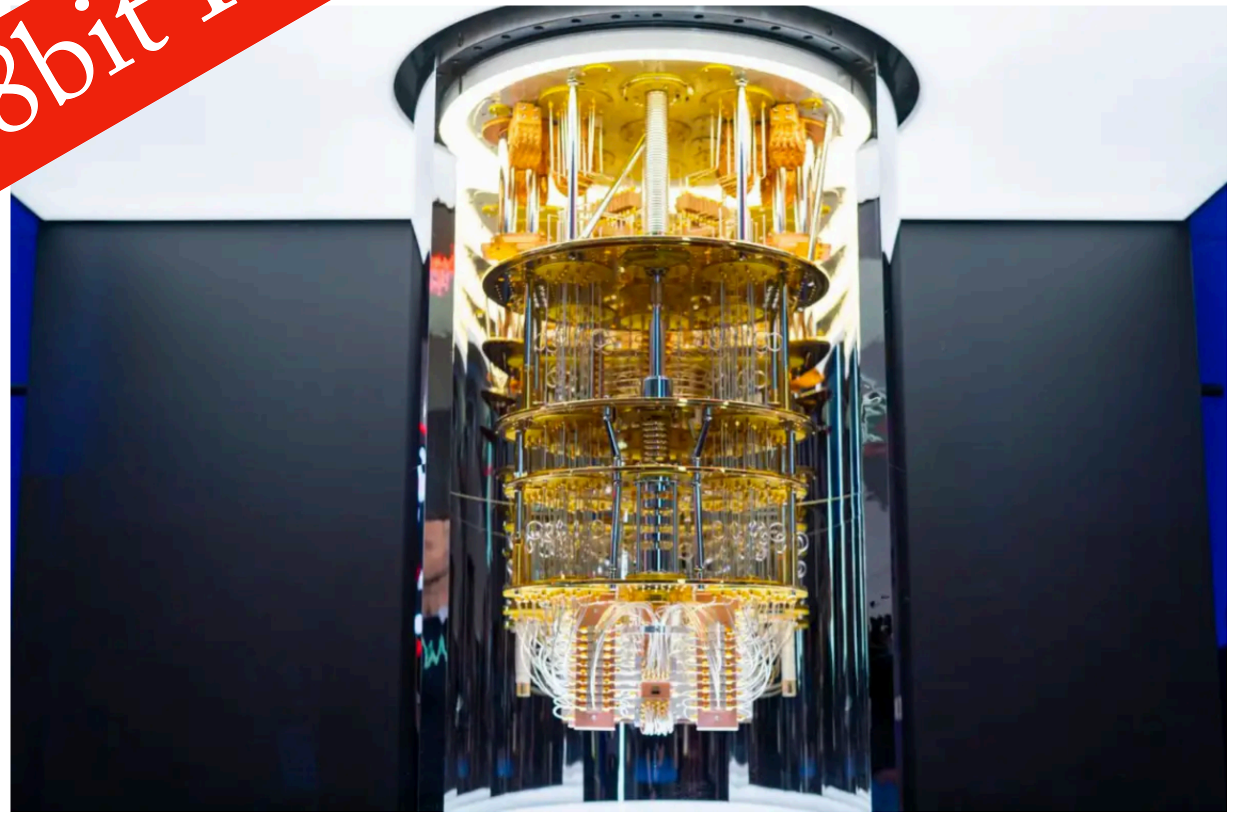
Quantum Computers - Current Status

FAR AWAY FROM FACTORING 2048bit INTEGER

Google - Willow Chip (2024)

105 qubit + [illegible] QEC

IBM - C[illegible]dor Chip (2025)



1121 qubit

Math of Quantum Computing on a single page

- Qubit is superposition of 0 and 1: $\alpha_0 |0\rangle + \alpha_1 |1\rangle \in \mathbb{C}^2$
- n-qubit system: superposition of all n-bit strings:

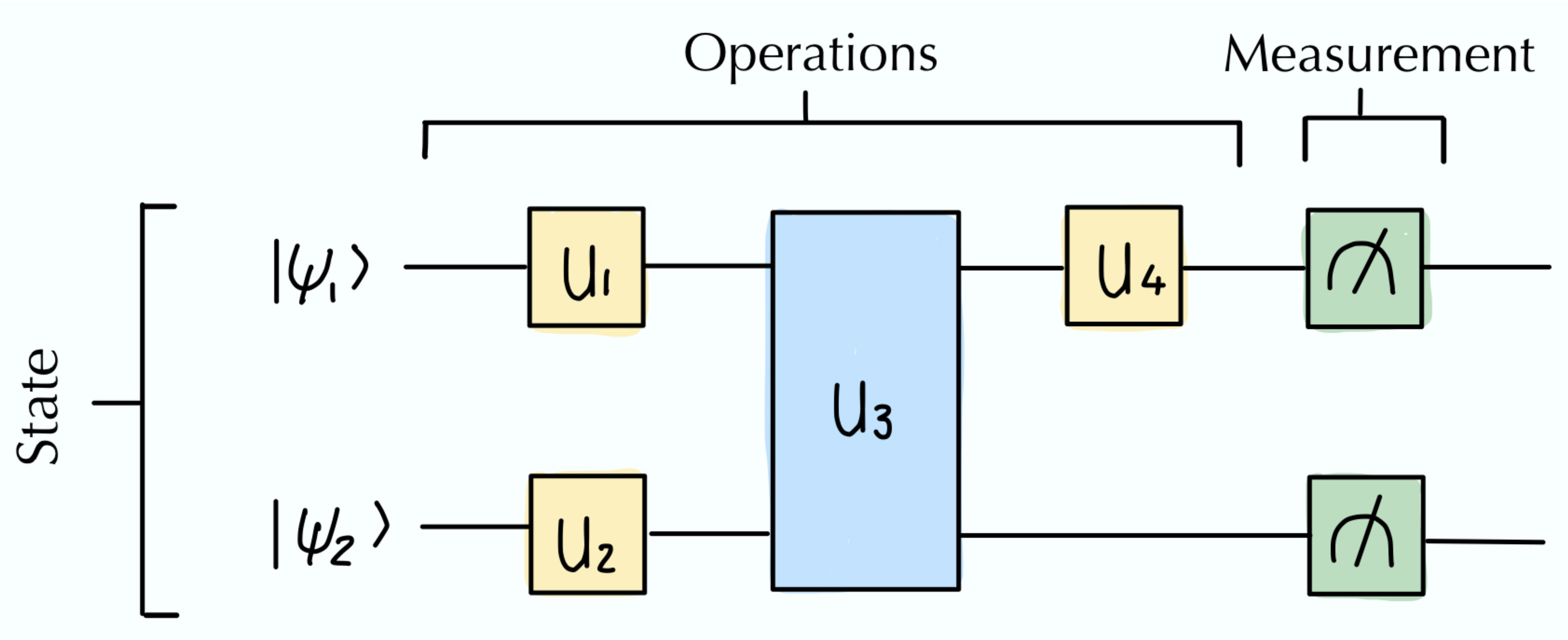
$$\sum_{x \in \{0,1\}^n} \alpha_x |x\rangle \in \mathbb{C}^{2^n}$$

- Measurement: see outcome $x \in \{0,1\}^n$ with probability $|\alpha_x|^2$
- Unitary transformation: matrix that preserves the length of the vector of amplitudes. Gates: unitaries on 1 qubit

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

- or on 2 qubits, CNOT: $|a, b\rangle \mapsto |a, a \oplus b\rangle$ $CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
- Combine simultaneous gates via tensor product, combine sequential gates via matrix product

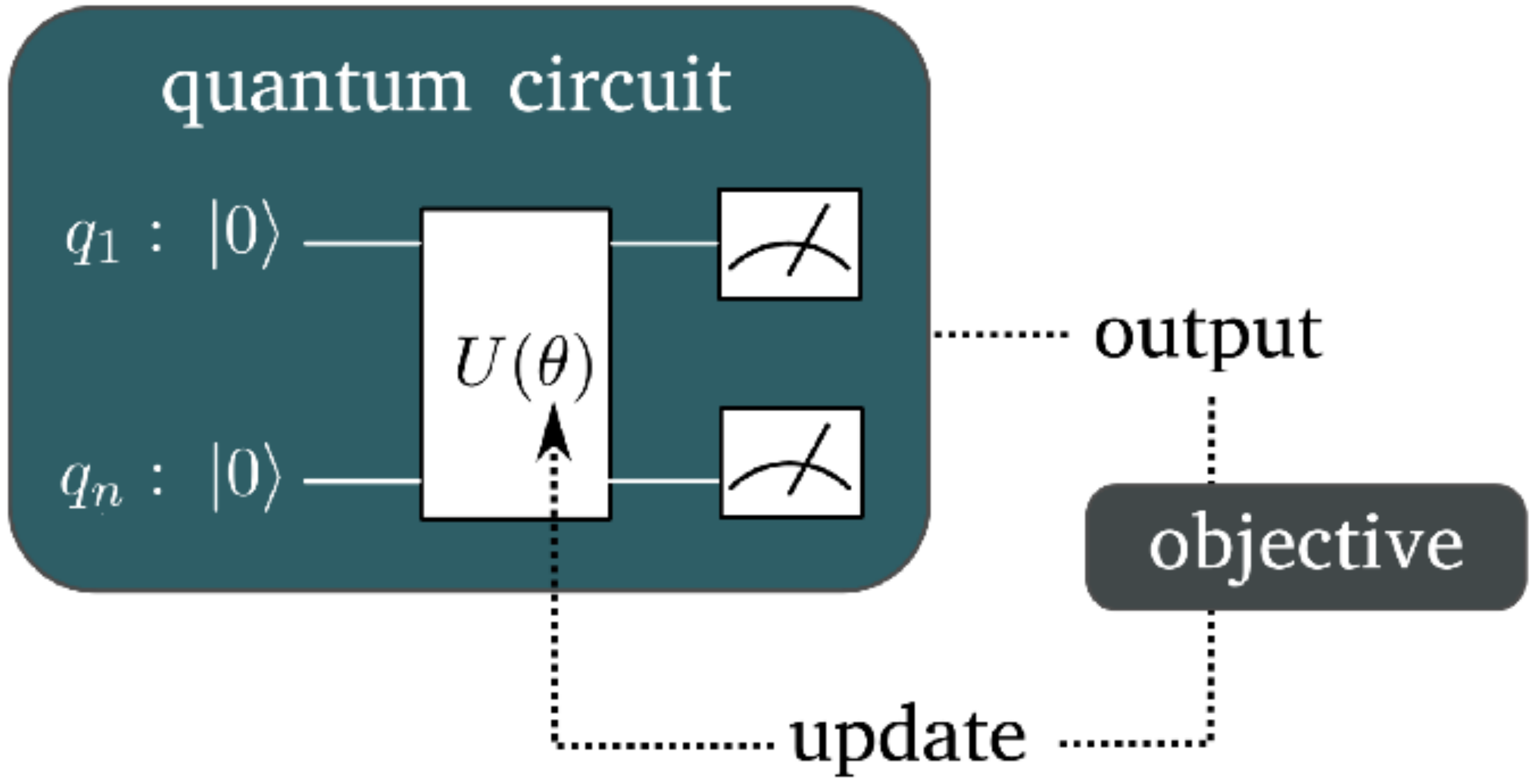
Circuit Model of Quantum Computations



Circuit for Quantum Computer

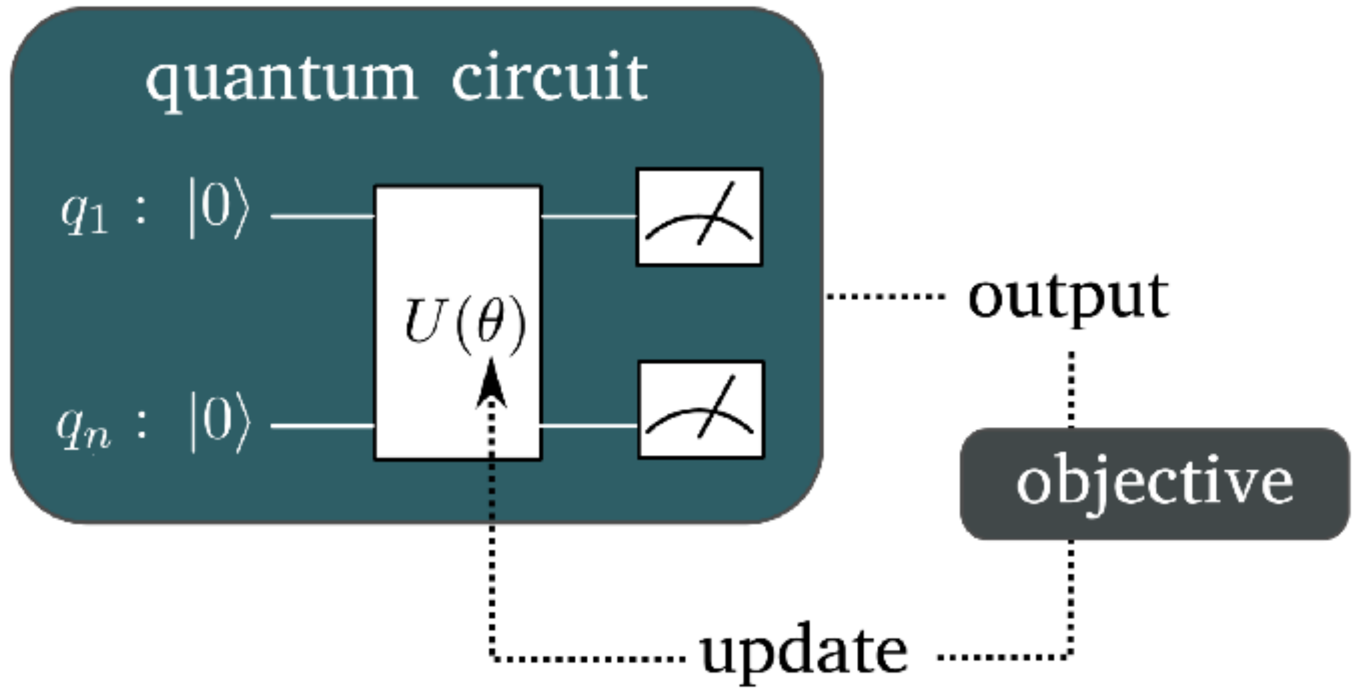
Quantum Computing with NISQ devices

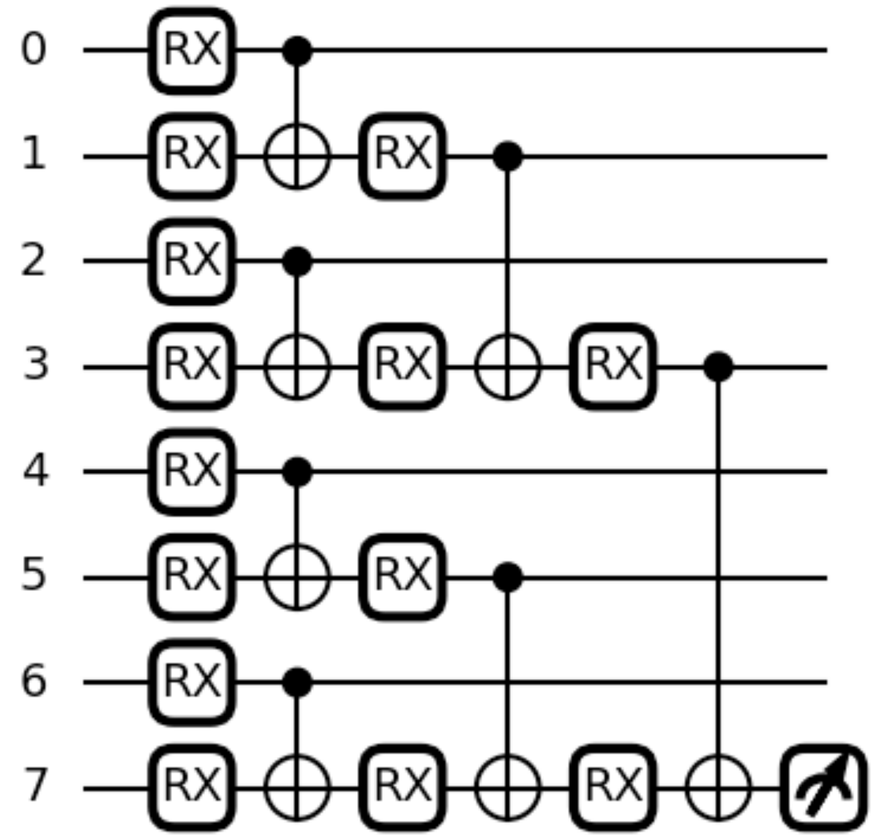
Quantum Parametric Circuit



Quantum Computing with NISQ devices

Quantum Parametric Circuit

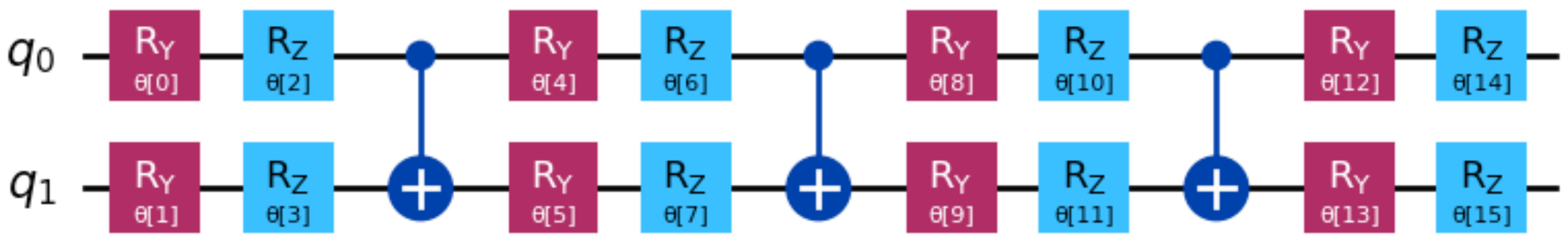




$$RX(\theta) = \exp\left(-i\frac{\theta}{2}X\right) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

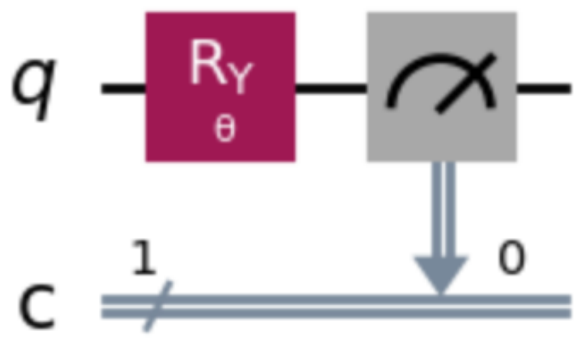
$$RZ(\phi) = \exp\left(-i\frac{\phi}{2}Z\right) = \begin{pmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{i\frac{\phi}{2}} \end{pmatrix}$$

$$RY(\theta) = \exp\left(-i\frac{\theta}{2}Y\right) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

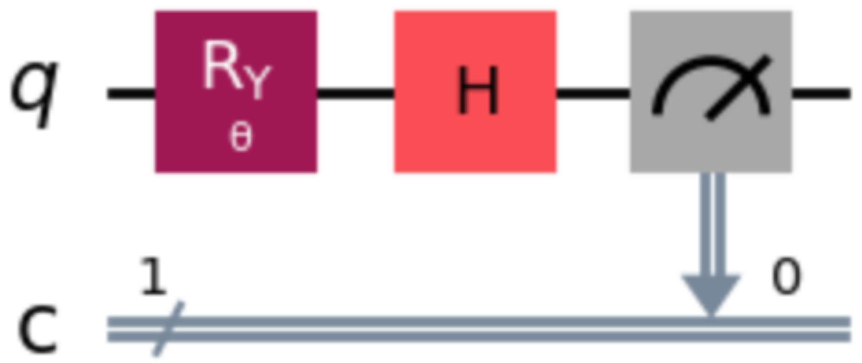


QPC for Ground State of a Two Level System

$$\hat{H}_{\text{TLS}} = \omega \hat{Z} + \lambda \hat{X} \qquad \text{Ansatz } |\psi(\theta)\rangle = \hat{R}_y(\theta)|0\rangle$$



$$\langle\psi(\theta)|\hat{Z}|\psi(\theta)\rangle = \cos\theta$$

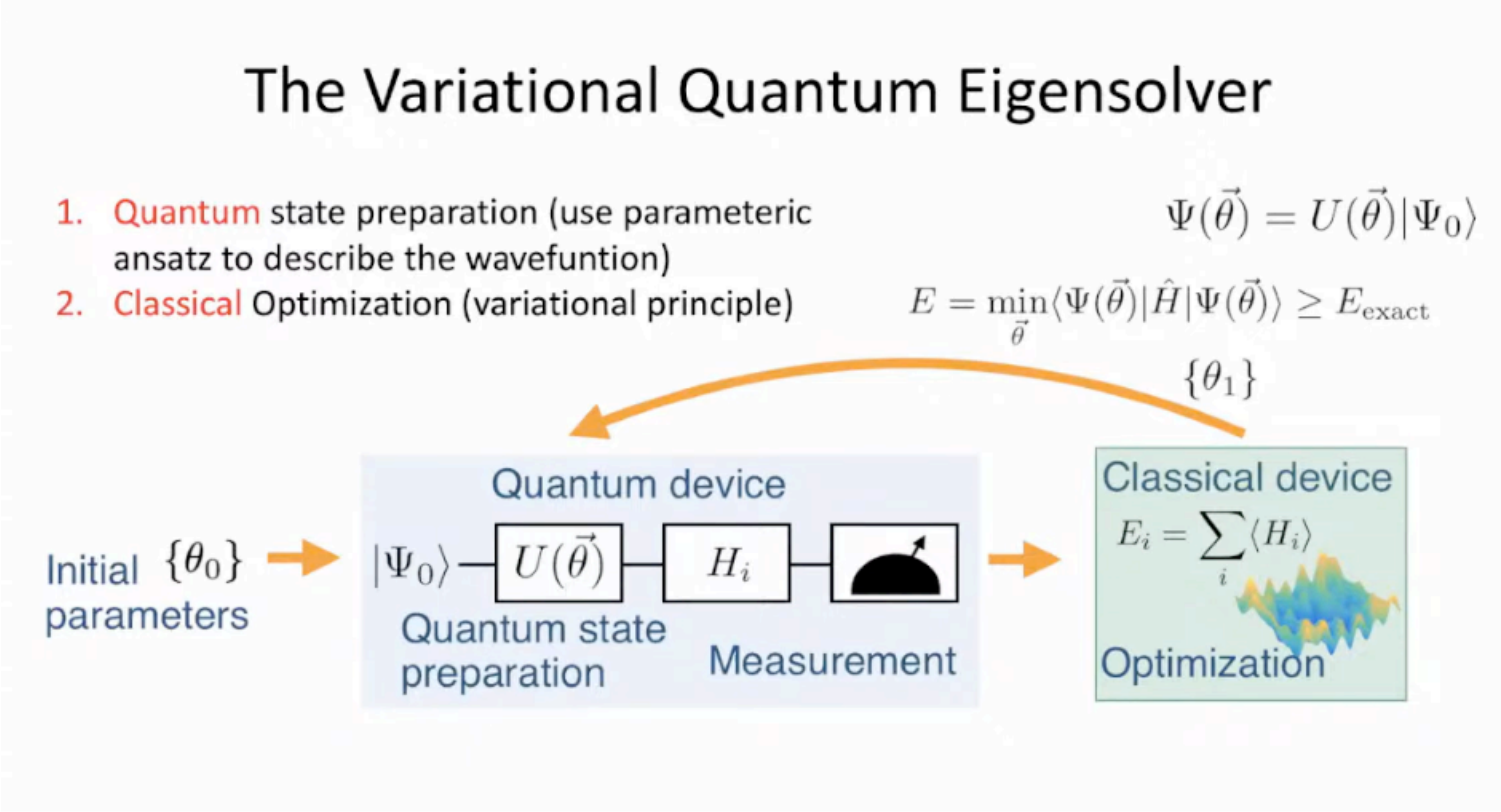


$$\langle\psi(\theta)|\hat{H}\hat{Z}\hat{H}|\psi(\theta)\rangle = \langle\psi(\theta)|\hat{X}|\psi(\theta)\rangle = \sin\theta$$

$$\langle\psi(\theta)|\hat{H}_{\text{TLS}}|\psi(\theta)\rangle = \omega\cos\theta + \lambda\sin\theta$$

$$\min_{\theta}(\langle\psi(\theta)|\hat{H}_{\text{TLS}}|\psi(\theta)\rangle) = -\sqrt{\omega^2 + \lambda^2}$$

General case: QPC for Ground State Estimation



The Variational Quantum Eigensolver

1. Quantum state preparation (use parameteric ansatz to describe the wavefuntion) $\Psi(\vec{\theta}) = U(\vec{\theta})|\Psi_0\rangle$
2. Classical Optimization (variational principle) $E = \min_{\vec{\theta}}\langle\Psi(\vec{\theta})|\hat{H}|\Psi(\vec{\theta})\rangle \geq E_{\text{exact}}$

$\{\theta_1\}$

Initial parameters $\{\theta_0\}$

Quantum device

$|\Psi_0\rangle$ — $U(\vec{\theta})$ — H_i

Quantum state preparation

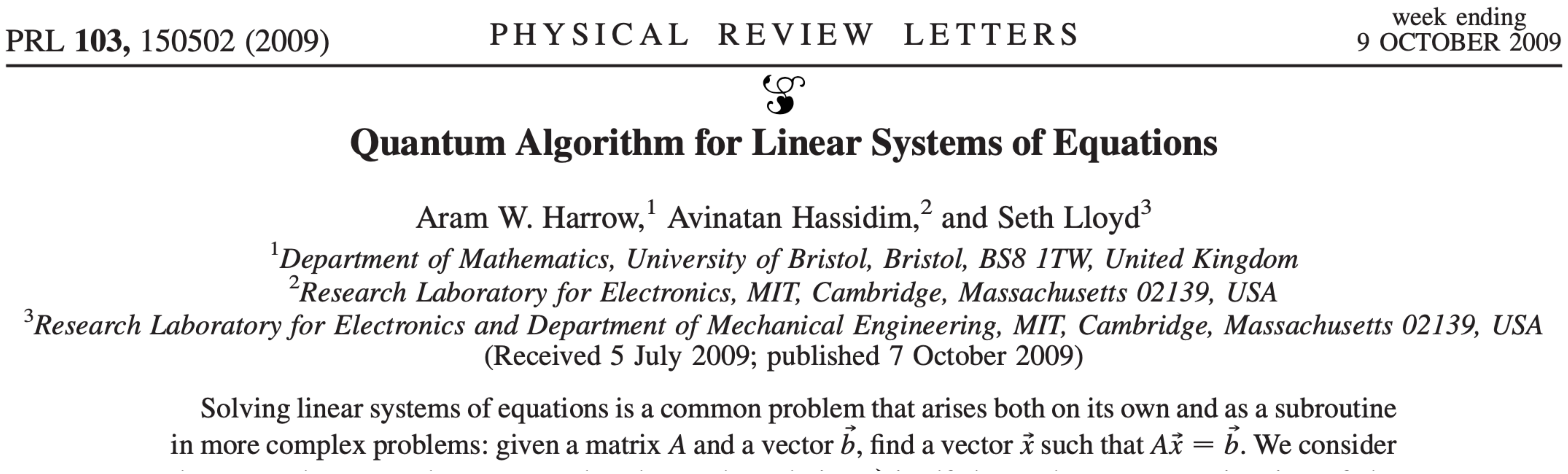
Measurement

Classical device

$E_i = \sum_i \langle H_i\rangle$

Optimization

Solving Systems of Linear Equations with QPC



PRL **103,** 150502 (2009) PHYSICAL REVIEW LETTERS week ending 9 OCTOBER 2009

Quantum Algorithm for Linear Systems of Equations

Aram W. Harrow,[1] Avinatan Hassidim,[2] and Seth Lloyd[3]

[1] *Department of Mathematics, University of Bristol, Bristol, BS8 1TW, United Kingdom*
[2] *Research Laboratory for Electronics, MIT, Cambridge, Massachusetts 02139, USA*
[3] *Research Laboratory for Electronics and Department of Mechanical Engineering, MIT, Cambridge, Massachusetts 02139, USA*

(Received 5 July 2009; published 7 October 2009)

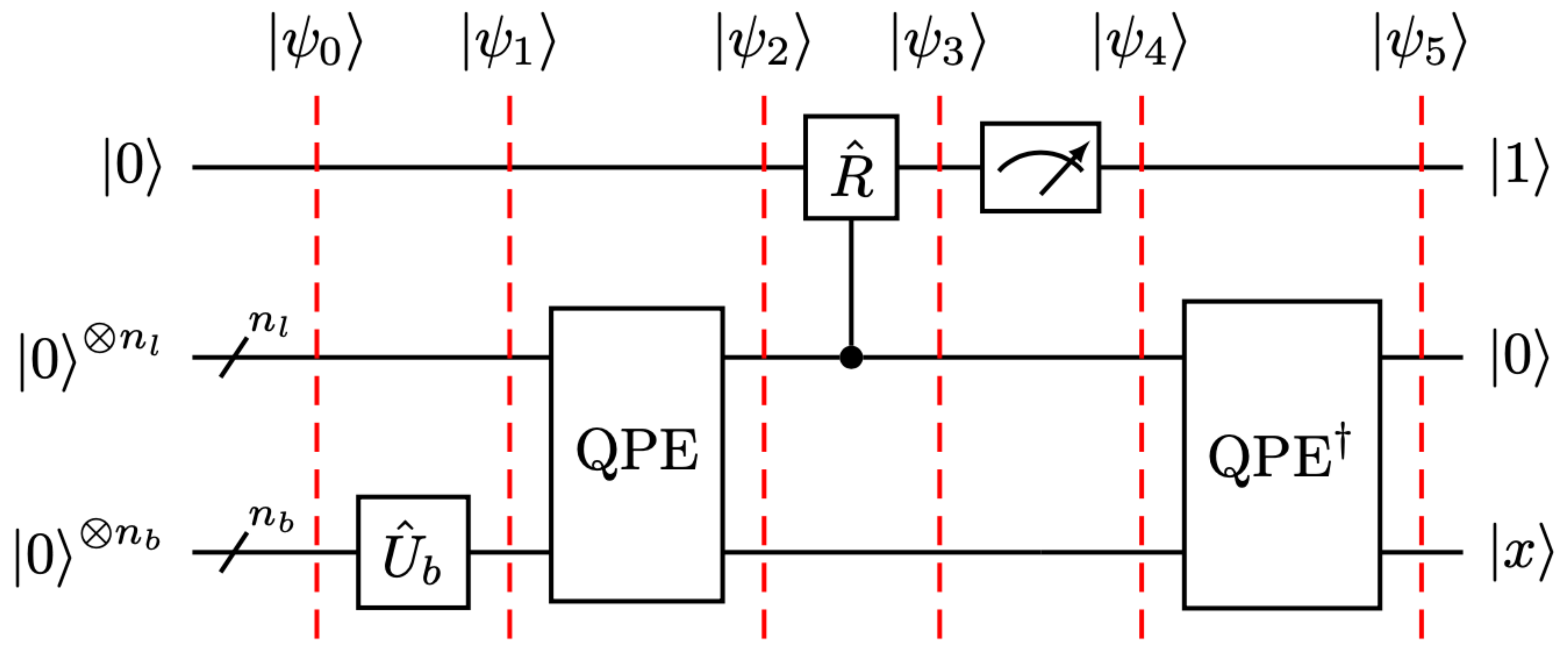
Solving linear systems of equations is a common problem that arises both on its own and as a subroutine in more complex problems: given a matrix A and a vector $\vec{b}$, find a vector $\vec{x}$ such that $A\vec{x} = \vec{b}$. We consider

Classical algorithms scales as $N\sqrt{\kappa}$.

HHL scales as poly(log(N), k)

$$A\vec{x} = \vec{b}. \longrightarrow A|x\rangle = |b\rangle, \xrightarrow{U = e^{iAt}} |x\rangle = A^{-1}|b\rangle.$$

HHL works in fault-tolerant settings only!



Solving Systems of Linear Equations with QPC

$$|x\rangle := \frac{\sum_i x_i |i\rangle}{||\sum_i x_i |i\rangle||_2},$$
$$|b\rangle := \frac{\sum_i b_i |i\rangle}{||\sum_i b_i |i\rangle||_2}.$$

$$\mathbf{A} = \sum_{i=1}^{m} c_i A_i$$

$$\tilde{\mathbf{A}} = \begin{pmatrix} 0 & \mathbf{A} \\ \mathbf{A}^\dagger & 0 \end{pmatrix}$$

$$|\psi\rangle = A|x\rangle \qquad U|0\rangle = |b\rangle$$

$$C_G = \frac{\langle x|H_G|x\rangle}{\langle\psi|\psi\rangle},$$

$$H_G = A^\dagger (\mathbb{1} - |b\rangle\langle b|)\, A.$$

Global CF is susceptible to BP!

$$H_L = A^\dagger U \left(\mathbb{1} - \frac{1}{n}\sum_{j=1}^{n} |0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}}\right) U^\dagger A,$$
$$C_L = \frac{\langle x|H_L|x\rangle}{\langle\psi|\psi\rangle}.$$

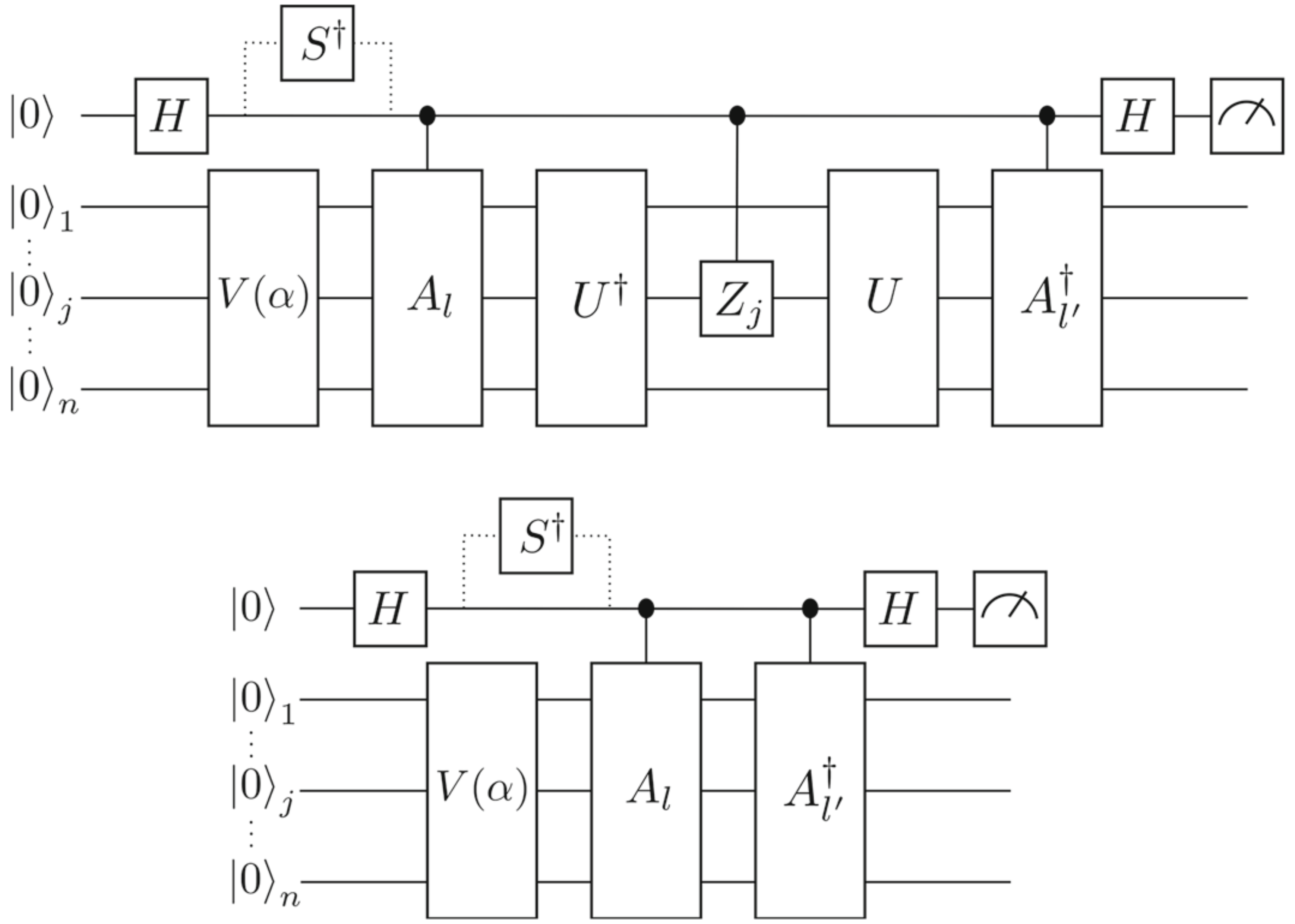


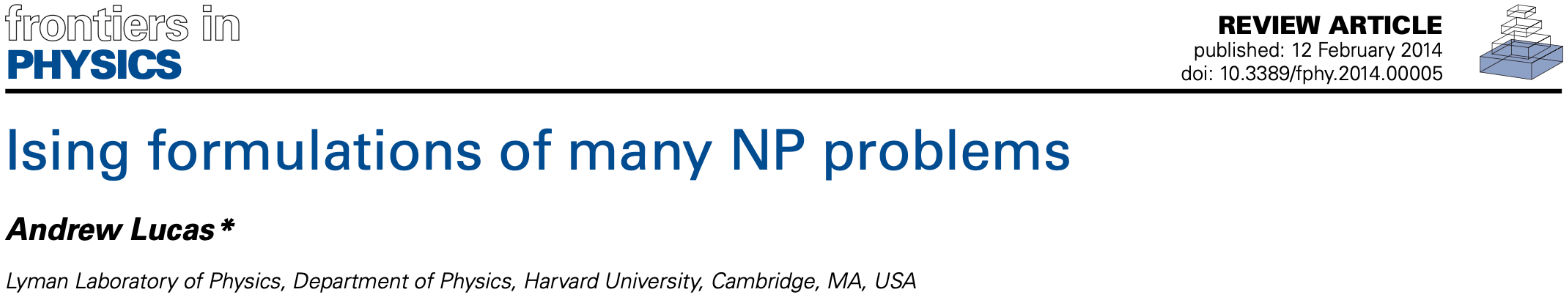
Fig. 2 Hadamard Test Circuits for cost function C_L Eq. (9). The top circuit is employed when calculating the value of the numerator $\langle x|H_L|x\rangle$, while the bottom circuit is employed when calculating the value of the denominator, $\langle\psi|\psi$. The $S^\dagger$ gate is included when calculating imaginary-valued parts of the cost function and excluded when calculating the real-valued parts, and therefore, drawn in as a dotted line. $V(\alpha)$ denotes the ansatz, A_l the l-th unitary from the linear sum $A = \sum_i c_i A_i$ and U the unitary such that $U|0\rangle = |b\rangle$. Z_j denotes a standard Z gate on the j-th qubit, and H being a standard Hadamard gate

Carlos Bravo-Prieto, Ryan LaRose, M. Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J. Coles, **Variational Quantum Linear Solver,** Quantum 7, 1188 (2023).

Pellow-Jarman, A., Sinayskiy, I., Pillay, A. *et al*. Near term algorithms for linear systems of equations. *Quantum Inf Process* **22**, 258 (2023). https://doi.org/10.1007/s11128-023-04020-2

Quantum Approximate Optimization Algorithm (QAOA)

Many problems in Optimization theory/Graph Theory/Computer Science can be formulated as a minimum energy state of classical spin glass.



frontiers in
PHYSICS

REVIEW ARTICLE
published: 12 February 2014
doi: 10.3389/fphy.2014.00005

Ising formulations of many NP problems

Andrew Lucas*

Lyman Laboratory of Physics, Department of Physics, Harvard University, Cambridge, MA, USA

Partitioning problems (number partitioning, graph partitioning)
Cliques, covering and packing problems (exact cover, set packing, vertex cover)
Satisfiability, minimal maximal matching, knapsack with integer weights, coloring problems
Hamiltonian cycles and paths, traveling salesman, tree problems…

Quantum Approximate Optimization Algorithm (QAOA)

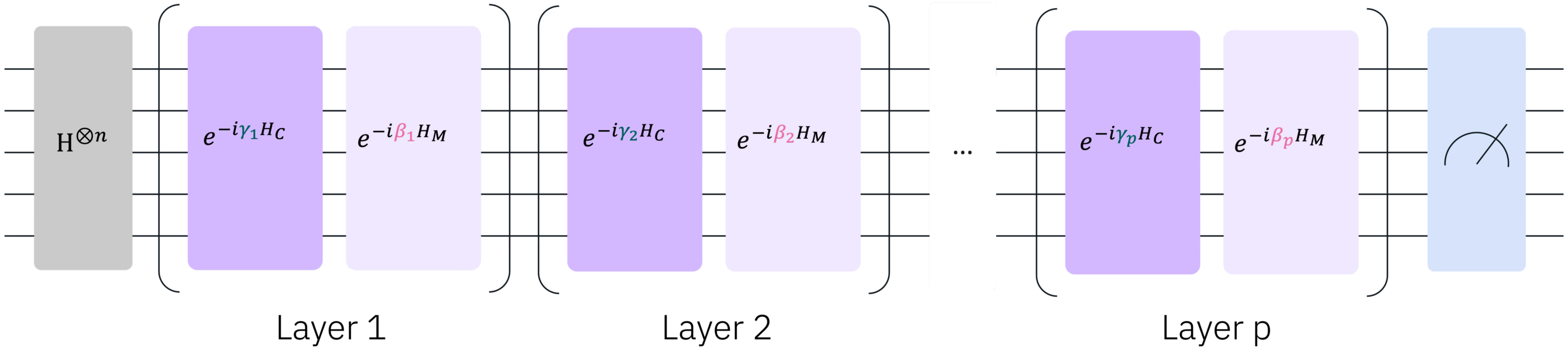
$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_P,$$

$$H_0 = -h_0 \sum_{i=1}^{N} \sigma_i^x,$$

$$H(s_1, \ldots, s_N) = -\sum_{i<j} J_{ij} s_i s_j - \sum_{i=1}^{N} h_i s_i.$$

We go from the ground state of the driver's Hamiltonian

To the ground state of the problem Hamiltonian



QAOA example: Minimum Vertex Cover

Minimize: $\sum_{i \in V} x_i$

Subject to: $x_i + x_j \geq 1, \quad \forall (i,j) \in E$

and: $x_i \in \{0, 1\}, \quad \forall i \in V$

$$H_C = A \sum_{(u,v) \in E} (1 - x_u)(1 - x_v) + B \sum_{v \in V} x_v$$

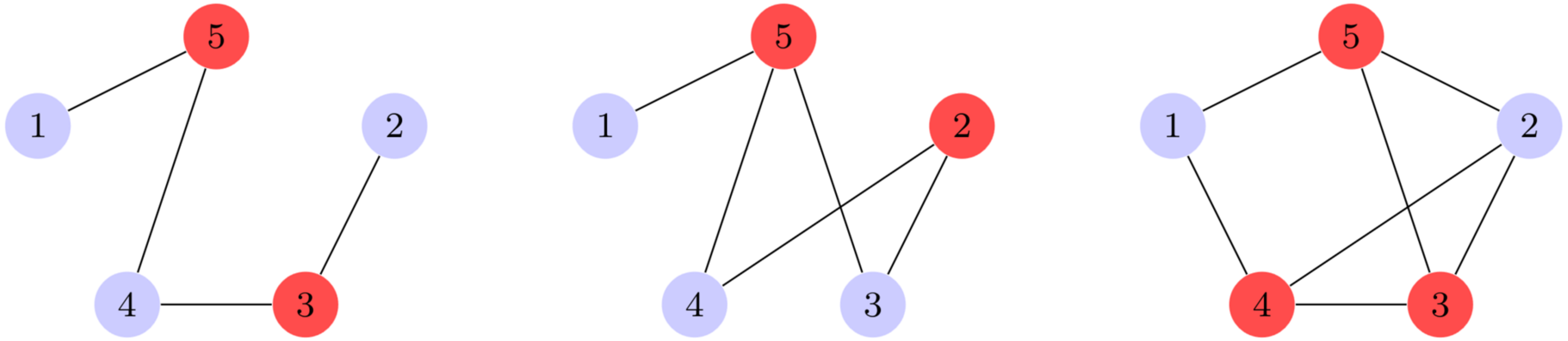
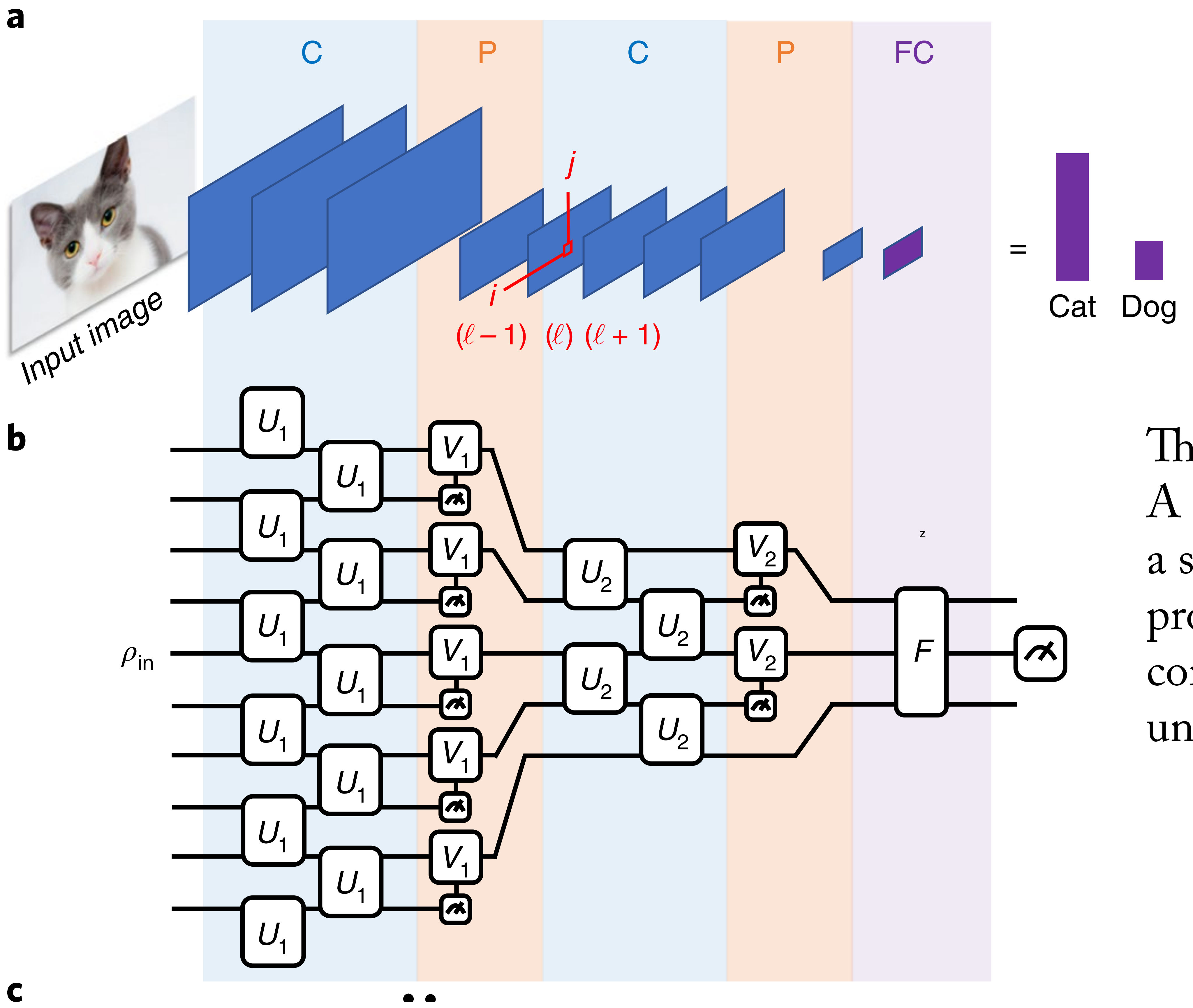


Figure 1. In the three graphs above, the red nodes show the set of vertices forming each graph's respective minimum vertex cover. Each edge in the graph under consideration must have, at least one vertex in the cover. The cover forms the minimum cover of a graph, when it contains the fewest number of vertices, whilst ensuring each edge is still incident to at least one.

A Pellow-Jarman, S McFarthing, IS, DK Park, A Pillay, F Petruccione, The effect of classical optimizers and Ansatz depth on QAOA performance in noisy devices. *Sci Rep* **14**, 16011 (2024)

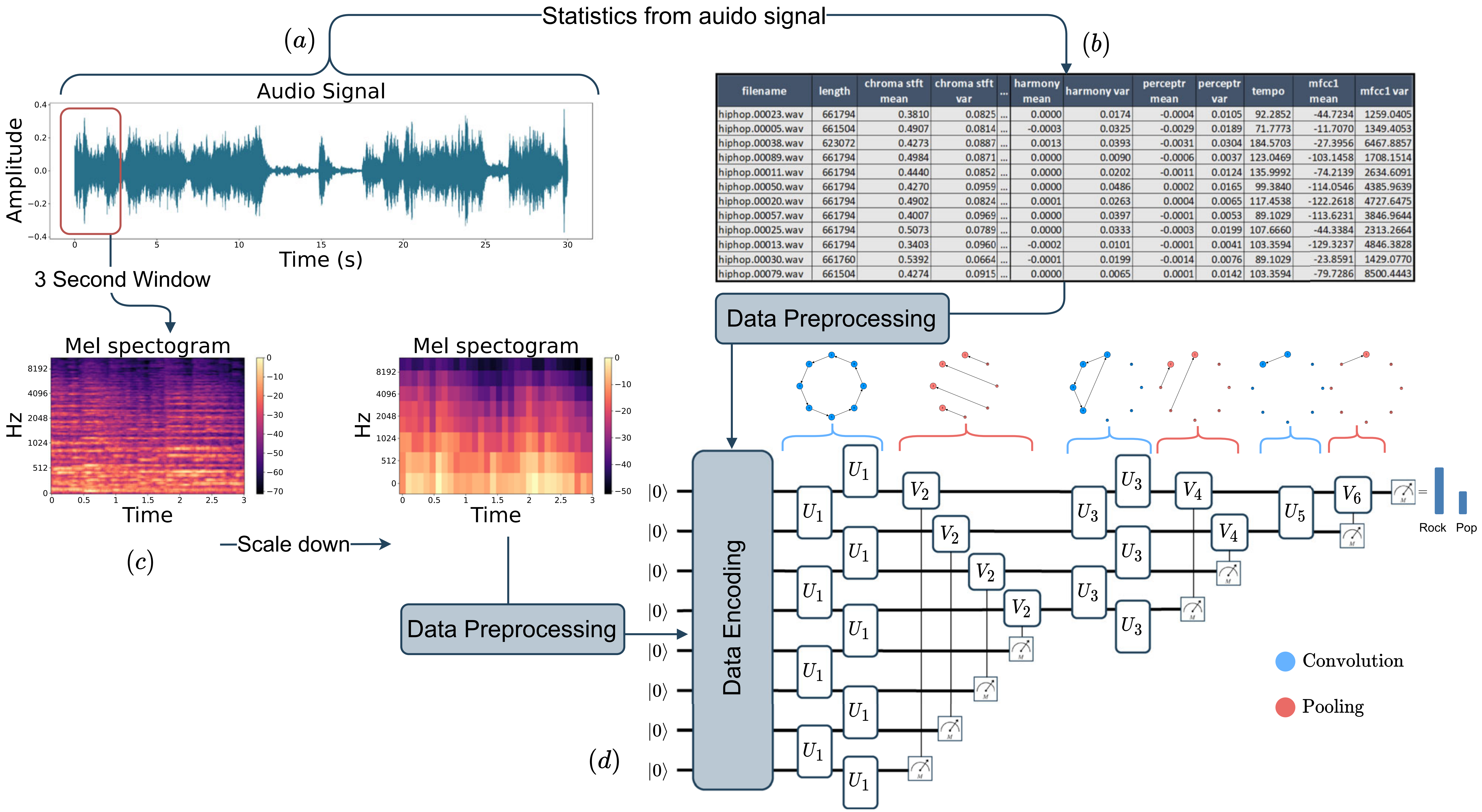
Quantum Convolutional Neural Networks



The concept of QCNNs. (a) Simplified illustration of classical CNNs. A sequence of image-processing layers transforms an input image into a series of feature maps (blue rectangles) and finally into an output probability distribution (purple bars). C, convolution; P, pooling; FC, fully connected. (b) QCNNs inherit a similar layered structure. Boxes represent unitary gates or measurement with feed-forwarding.

Cong, I., Choi, S. & Lukin, M.D. Quantum convolutional neural networks. *Nat. Phys.* **15**, 1273–1278 (2019).

Quantum Convolutional Neural Networks

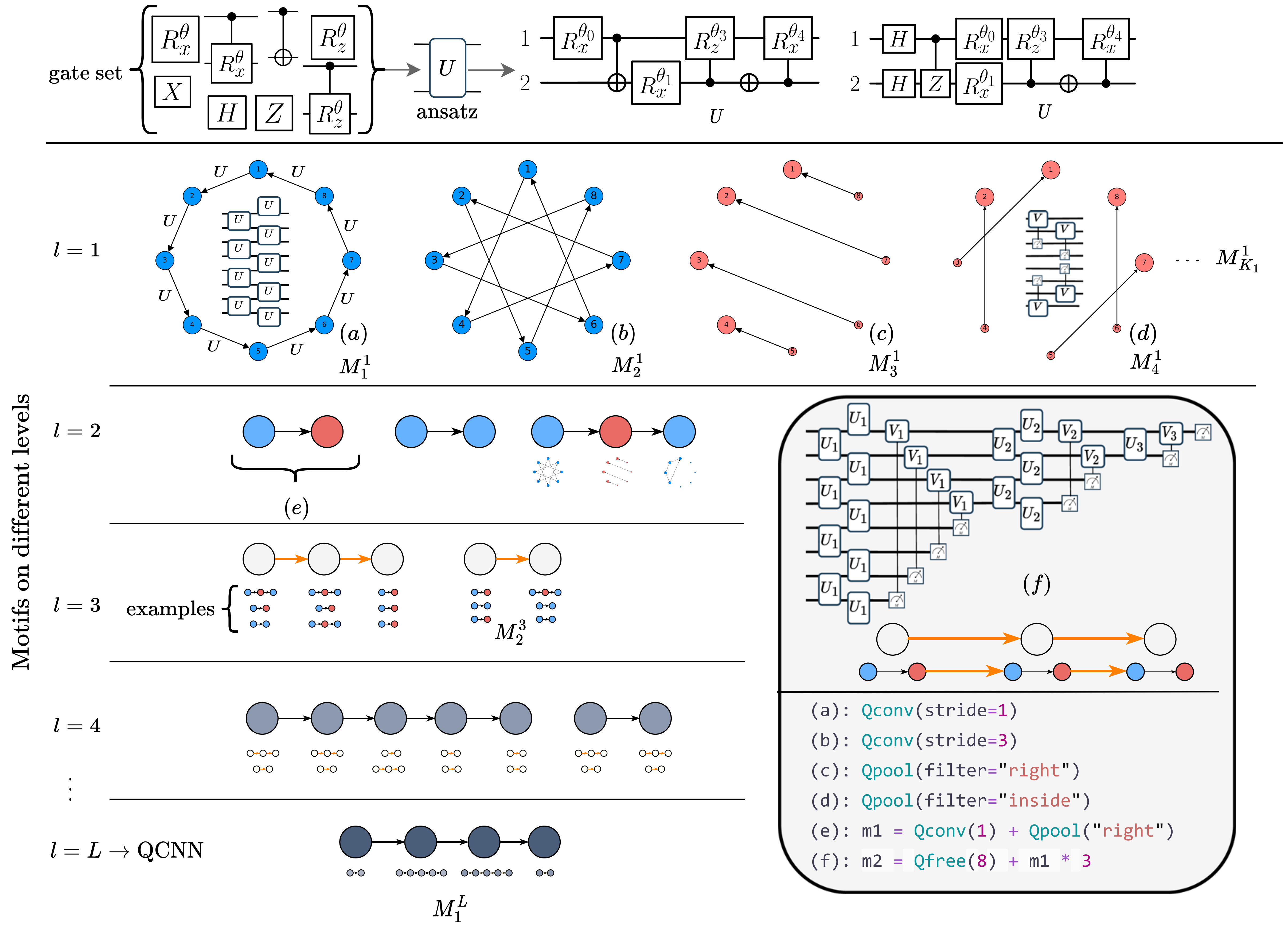


filename	length	chroma stft mean	chroma stft var	...	harmony mean	harmony var	perceptr mean	perceptr var	tempo	mfcc1 mean	mfcc1 var
hiphop.00023.wav	661794	0.3810	0.0825	...	0.0000	0.0174	-0.0004	0.0105	92.2852	-44.7234	1259.0405
hiphop.00005.wav	661504	0.4907	0.0814	...	-0.0003	0.0325	-0.0029	0.0189	71.7773	-11.7070	1349.4053
hiphop.00038.wav	623072	0.4273	0.0887	...	0.0013	0.0393	-0.0031	0.0304	184.5703	-27.3956	6467.8857
hiphop.00089.wav	661794	0.4984	0.0871	...	0.0000	0.0090	-0.0006	0.0037	123.0469	-103.1458	1708.1514
hiphop.00011.wav	661794	0.4440	0.0852	...	0.0000	0.0202	-0.0011	0.0124	135.9992	-74.2139	2634.6091
hiphop.00050.wav	661794	0.4270	0.0959	...	0.0000	0.0486	0.0002	0.0165	99.3840	-114.0546	4385.9639
hiphop.00020.wav	661794	0.4902	0.0824	...	0.0001	0.0263	0.0004	0.0065	117.4538	-122.2618	4727.6475
hiphop.00057.wav	661794	0.4007	0.0969	...	0.0000	0.0397	-0.0001	0.0053	89.1029	-113.6231	3846.9644
hiphop.00025.wav	661794	0.5073	0.0789	...	0.0000	0.0333	-0.0003	0.0199	107.6660	-44.3384	2313.2664
hiphop.00013.wav	661794	0.3403	0.0960	...	-0.0002	0.0101	-0.0001	0.0041	103.3594	-129.3237	4846.3828
hiphop.00030.wav	661760	0.5392	0.0664	...	-0.0001	0.0199	-0.0014	0.0076	89.1029	-23.8591	1429.0770
hiphop.00079.wav	661504	0.4274	0.0915	...	0.0000	0.0065	0.0001	0.0142	103.3594	-79.7286	8500.4443

Fig. 1 Machine-learning pipeline for music genre classification. The machine-learning pipeline we implemented for music genre classification. Given an audio signal of a song (**a**), we generate two forms of data: tabular (**b**) and image (**c**). Each form has data preprocessing applied before being encoded into a quantum state (**d**). The QCNN circuit shown in (**d**) favours Principal Component Analysis (PCA) because qubits are pooled from bottom to top, and principal components are encoded from top to bottom. This architecture is an instance of the reverse binary tree family that we generated with our framework.

M Lourens, IS, DK Park, C Blank and F Petruccione. "Hierarchical quantum circuit representations for neural architecture search", *npj Quantum Inf* **9**, 79 (2023)

Quantum Convolutional Neural Networks



```
(a): Qconv(stride=1)
(b): Qconv(stride=3)
(c): Qpool(filter="right")
(d): Qpool(filter="inside")
(e): m1 = Qconv(1) + Qpool("right")
(f): m2 = Qfree(8) + m1 * 3
```

Fig. 2 Hierarchical quantum circuit representation. An overview of our architectural representation for QCNNs. From a given set of gates, we build two-qubit unitary ansatzes. The representation then captures design motifs M_k^l on different levels l of the hierarchy. On the lowest level $l = 1$, we define primitives which act as building blocks for the architecture. For example, a convolution operation with stride one is encoded as the directed graph M_1^1. The directed graph M_3^1 is a pooling operation that measures the bottom half of the circuit. Combined, they form the level two motif (e): a convolution-pooling unit M_1^2. Higher-level motifs consist of combinations of lower-level motifs up until the final level $l = L$, which contains only one motif M_1^L, the complete QCNN architecture. M_1^L is a hierarchy of directed graphs fully specifying how to spread the unitary ansatzes across the circuit. The two lines of code (e) and (f) show the power of this representation as it is all that is required to create the entire QCNN circuit from Fig. 1d. The code comes from the Python package we implemented based on the work of this paper. It facilitates dynamic QCNN creation and search space design.

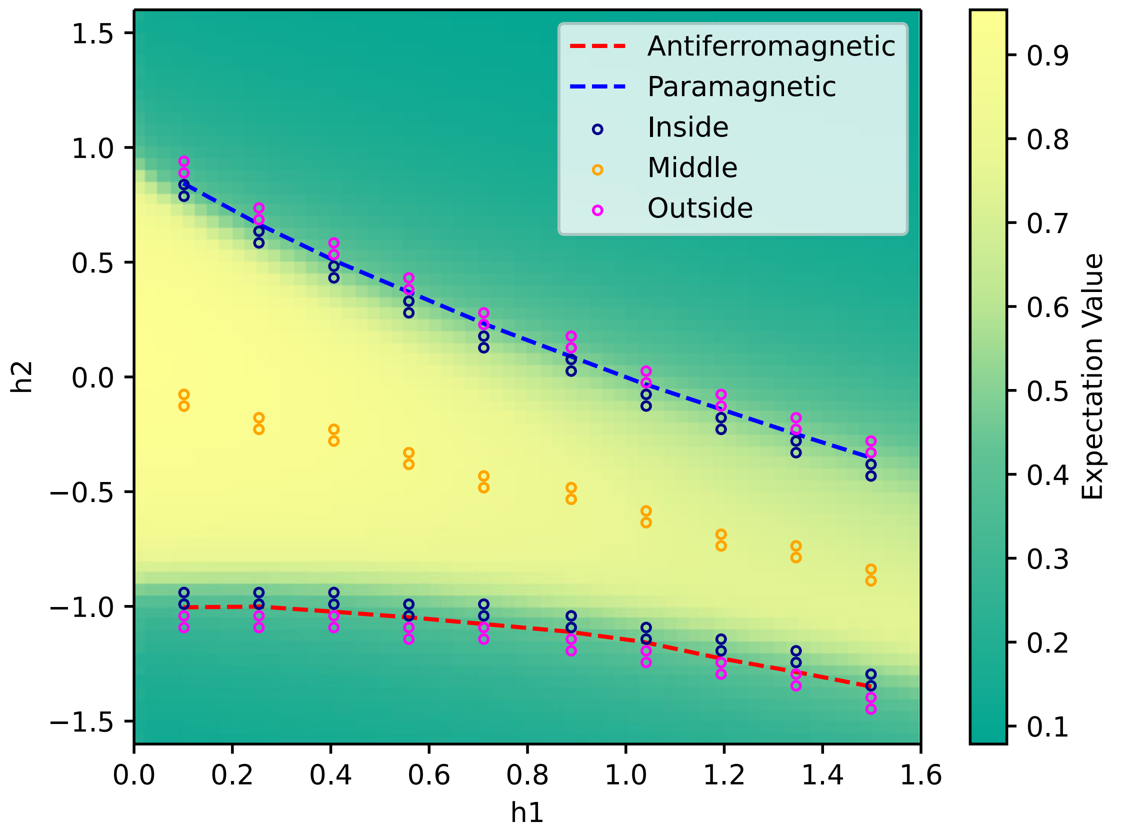


Fig. 5 Quantum phase recognition result. Expectation values for the circuit found via evolutionary search for a system of $N = 15$ spins. Points represent a test set of 64×64 ground states for various h_1 and h_2 values of the Hamiltonian, $J = 1$. The inside, middle and outside points were used to evaluate an architecture's fitness during search. The same colour scale as in[5] is used to facilitate comparison.

Table 3. Performance of architecture found with an evolutionary search.

Metric	Reference	Found
Number of parameters	1308	**11**
Sample complexity (inside)	61.523	**36.079**
Sample complexity (middle)	**10.992**	13.253
MSE (outside)	**0.164**	0.167

Different performance metrics (lower is better) for the 15-qubit QCNN from ref. [5] and the architecture found via evolutionary search. The best performing architecture for each metric is highlighted in bold. Sample complexity represents the expected number of measurements required to be 95% confident that the ground state is in the SPT phase (non-zero expectation value). Metrics are calculated on a set of points in the test set, where inside refers to SPT points near the phase boundary, outside to non-SPT points near the phase boundary and middle to points in between, as shown in Fig. 5.

M Lourens, IS, DK Park, C Blank and F Petruccione. "Hierarchical quantum circuit representations for neural architecture search", *npj Quantum Inf* **9**, 79 (2023)

Surprising Application: Ansatz Search

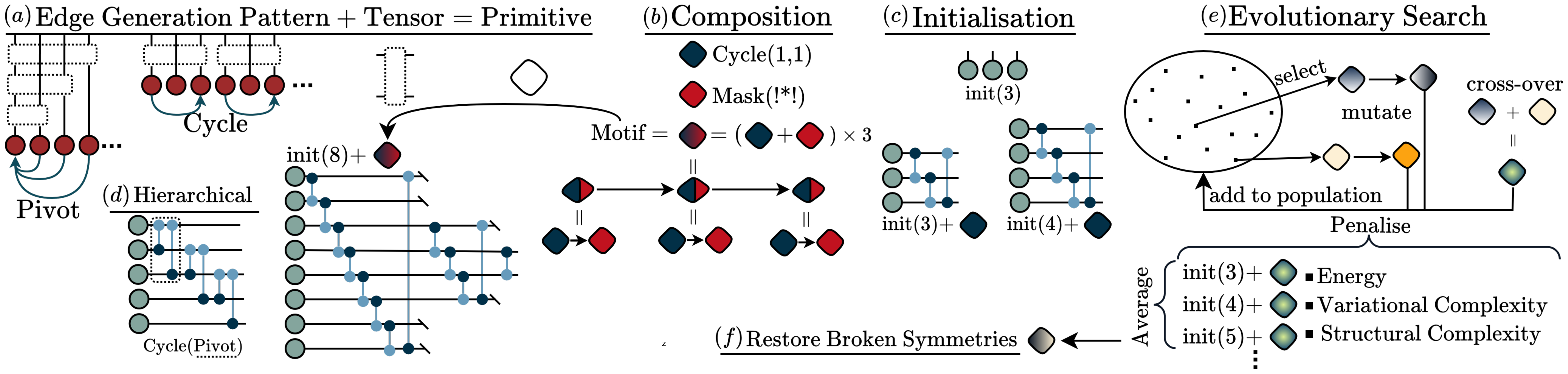


FIG. 1. Overview of our method, a domain-specific-language (DSL) enables ansatz generation via an evolutionary algorithm. (a) A primitive is an edge generation pattern associated with a tensor. (b) Composition: Sequences of primitives form motifs; sequences of motifs form higher-level motifs. (c) Specifying the number of nodes generates edges, and the associated tensor is repeated and connected to each edge, forming a tensor network. (d) A specified network, being itself a tensor, can again be associated with an edge generation pattern to form a new primitive. (e) The evolutionary algorithm mutates and crosses over motifs each generation. (f) Once the ansatz is found, broken symmetries are restored.

Matt Lourens, IS, Johannes N. Kriel, and Francesco Petruccione, Generating Generalised Ground-State Ansatzes from Few-Body Examples, arXiv:2503.00497 (2025)

Surprising Application: Ansatz Search

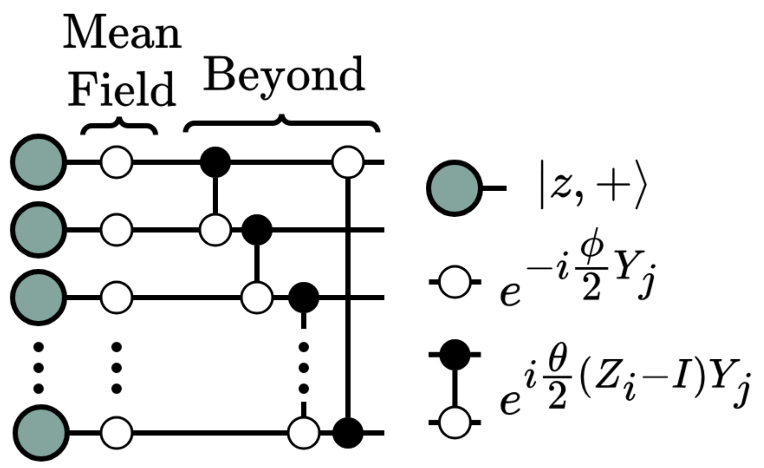


FIG. 2. The ansatz generated by our method for the LMG and TFIM models.

$$|\theta,\phi\rangle = \left(\prod_{k=0}^{N-1} C^{\theta}_{k,k+1} R^{\theta}_{k+1}\right)\left(\prod_{j=0}^{N-1} R^{\phi}_{j}\right)|z,+\rangle^{\otimes N}$$

$$C^{\theta}_{ij} = e^{i\frac{\theta}{2}Z_iY_j} \qquad \text{and} \qquad R^{\theta}_{j} = e^{-i\frac{\theta}{2}Y_j}.$$

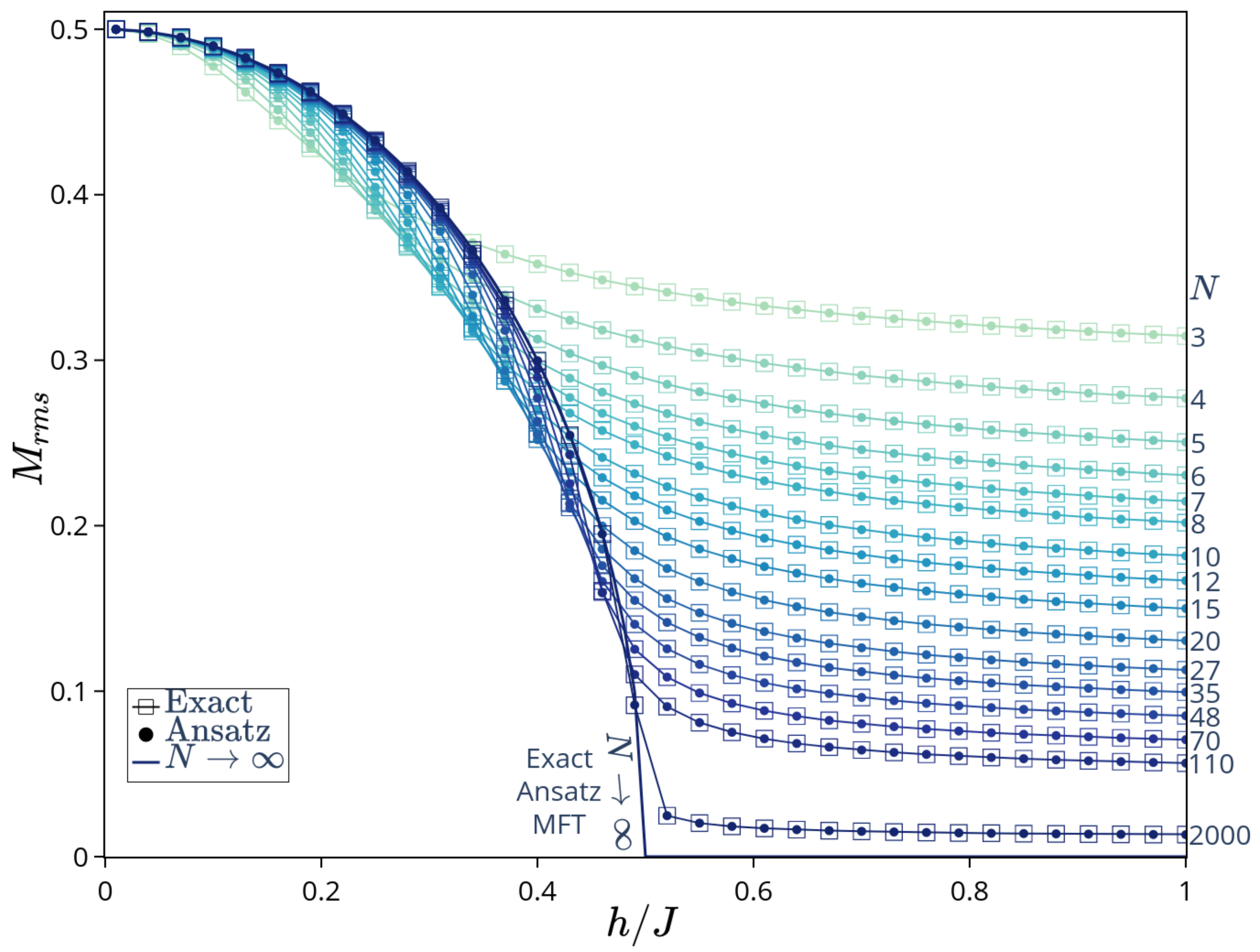


FIG. 3. RMS magnetisation of Eq. (20) vs h/J for the LMG model. Exact results compared to the symmetrised ansatz.

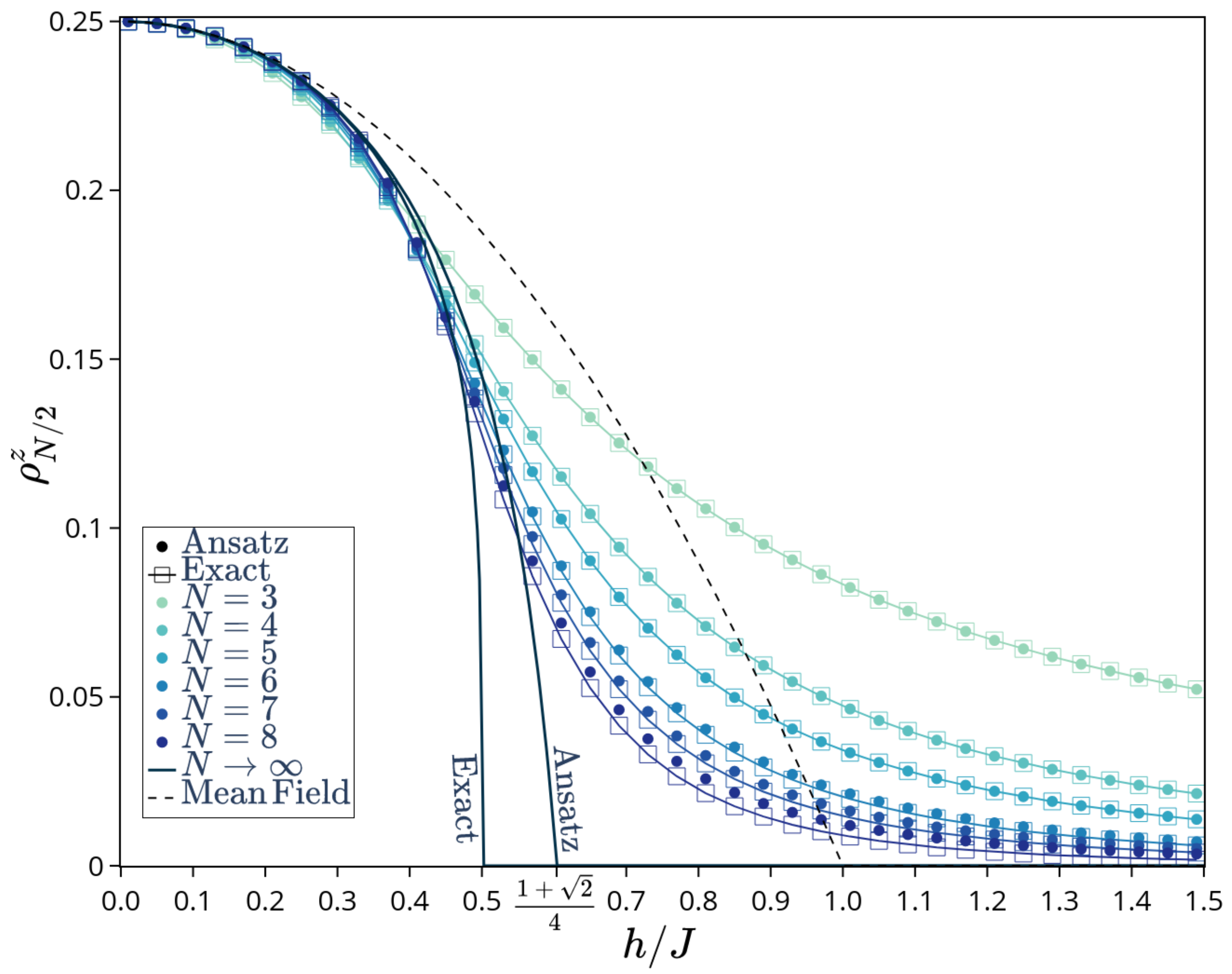


FIG. 5. TFIM long-range correlation $\rho^z_{N/2}$ vs h/J. Shows results from the symmetrised ansatz (finite N, $N \to \infty$), exact values, and the mean-field prediction.

Matt Lourens, IS, Johannes N. Kriel, and Francesco Petruccione, Generating Generalised Ground-State Ansatzes from Few-Body Examples, arXiv:2503.00497 (2025)

UNIVERSITY OF KWAZULU-NATAL™
INYUVESI YAKWAZULU-NATALI

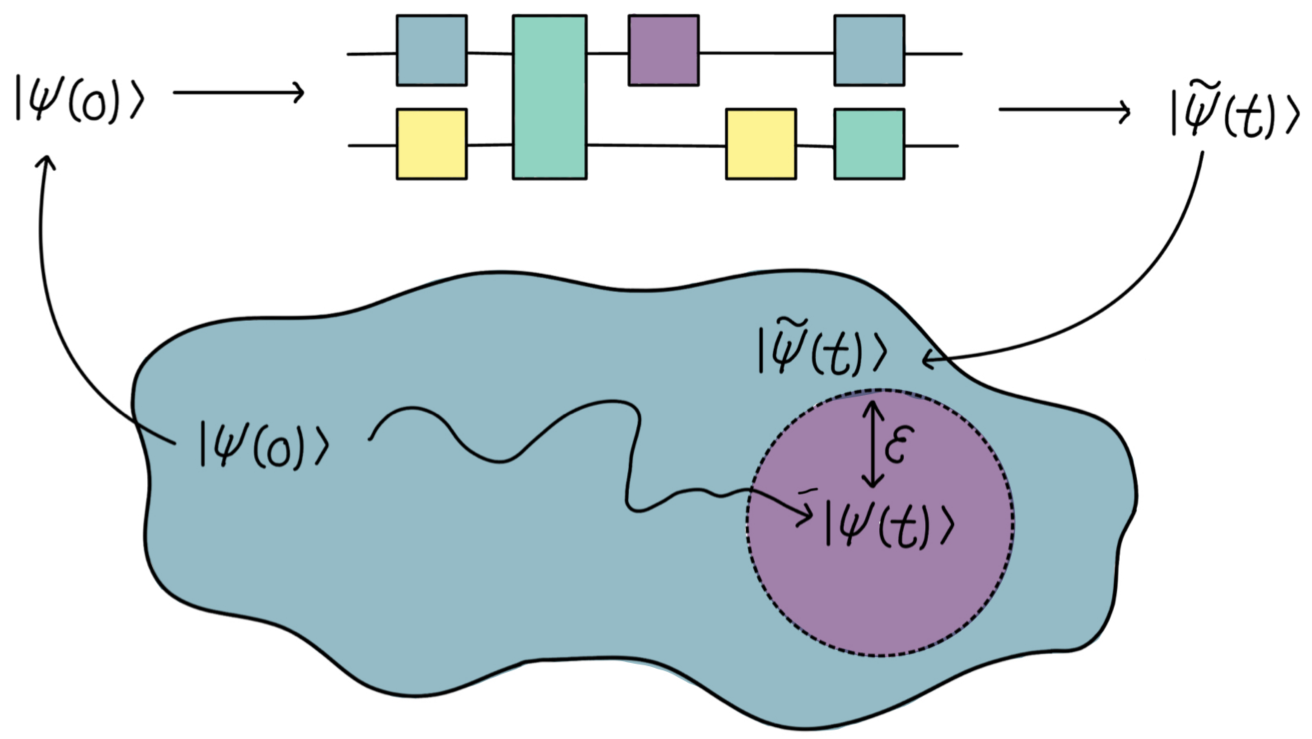
SA QuTI South African Quantum Technology Initiative

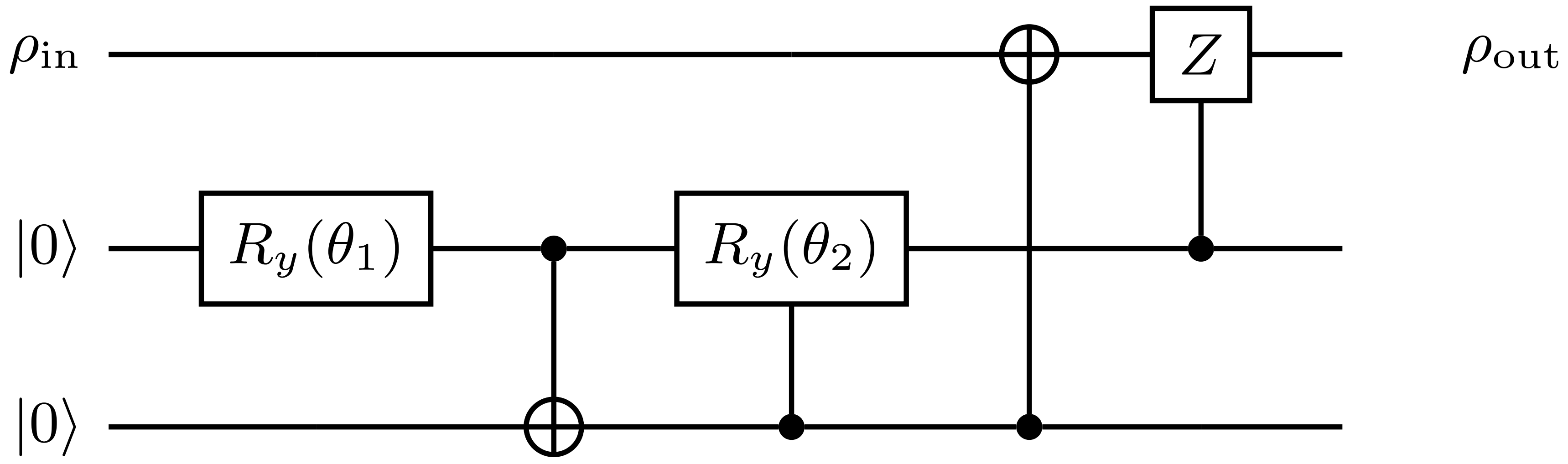
NITheCS
National Institute for Theoretical and Computational Sciences

Thank you!

Extra questions? - sinayskiy@ukzn.ac.za

quantum.ukzn.ac.za/sinayskiy


$|\psi(0)\rangle$
$|\tilde{\psi}(t)\rangle$
$|\tilde{\psi}(t)\rangle$
$|\psi(0)\rangle$
ε
$|\psi(t)\rangle$


ρ_{in}
$|0\rangle$
$|0\rangle$
$R_y(\theta_1)$
$R_y(\theta_2)$
Z
ρ_{out}