Basic approaches to quantum computing

Mathematical introductory tutorial

Evgeniy O. Kiktenko

SIRIUS SCHOOL 2020
September 15

Slides available online: github.com/EvgeniyKiktenko/RQC-MIPT-q-inf-proc-course/Slides

Outline

0) Basics of formalism
1) Universal quantum computing
2) NISQ devices
3) Adiabatic quantum computing
4) Variational quantum computing

Basics of formalism

Basics of formalism

From bits to qubits

$$0 \leftrightarrow |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad 1 \leftrightarrow |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$101 \leftrightarrow |101\rangle \equiv |1\rangle \otimes |0\rangle|\otimes 1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{array}{c} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \end{array} \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}$$

$$\mathrm{bin}('101') = 5$$

$$\boxed{\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a \begin{bmatrix} c \\ d \end{bmatrix} \\ b \begin{bmatrix} c \\ d \end{bmatrix} \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}}$$

Basics of formalism

Quantum states

$$??? \leftrightarrow |\psi\rangle = C_{000}|000\rangle + C_{001}|001\rangle + \ldots C_{111}|111\rangle = \begin{bmatrix} C_{000} \\ C_{001} \\ C_{010} \\ C_{011} \\ C_{100} \\ C_{101} \\ C_{110} \\ C_{111} \end{bmatrix}$$

$$C_{\mathbf{x}} \in \mathbb{C}, \qquad \sum_{\mathbf{x}\in\{0,1\}^3} |C_{\mathbf{x}}|^2 = 1$$

For n qubits # of complex coefficients = 2^n ($2^{1024} \approx 1.8 \times 10^{308}$)

Basics of formalism

Measurement in the computational basis: back to classical bits

$$|\psi\rangle = \begin{bmatrix} C_{000} \\ C_{001} \\ C_{010} \\ C_{011} \\ C_{100} \\ C_{101} \\ C_{110} \\ C_{111} \end{bmatrix} \rightarrow \text{bitstring } \mathbf{x} \in \{0,1\}^3 \text{ with probability } |\langle \mathbf{x}|\psi\rangle|^2 = |C_{\mathbf{x}}|^2$$

- Measurement of qubit = sampling bit string from distribution determined by this state
- The state «collapses» after the measurement (one need to re-prepair the state for another sampling)

$$\overset{\text{«ket»}}{|\phi\rangle} = \begin{bmatrix} a \\ b \end{bmatrix} \leftrightarrow \overset{\text{«bra»}}{\langle\phi|} = |\phi\rangle^\dagger = (|\phi\rangle^*)^T = [a^* \quad b^*]$$

$$\overset{\text{«bracket»}}{\langle\chi|\theta\rangle} \equiv \overset{\text{«bra»}}{\langle\chi|} \cdot \overset{\text{«ket»}}{|\theta\rangle}$$

Basics of formalism

Evolution of quantum states

$$\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = -iH(t)|\psi(t)\rangle$$

$$|\psi(t)\rangle = U(t)|\psi(t=0)\rangle$$

($U(t) = e^{-iHt}$ for time-independent H)

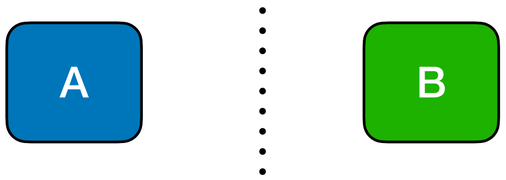
$U := U(t=\tau)$ – quantum gate

$H \sim H + \alpha\mathbf{1} \quad U \sim e^{i\phi}U$

each U has an "analog" with unit determinant (a matrix from $\mathbf{SU}$ group)

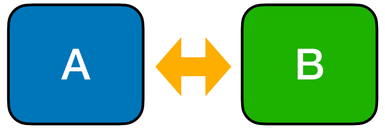
Basics of formalism

Local and non-local operations



$$H = H_A + H_B = H_A \otimes \mathbf{1} + \mathbf{1} \otimes H_B$$

$$U = (U_A \otimes \mathbf{1})(\mathbf{1} \otimes U_B) = U_A \otimes U_B$$



$$H = H_A + H_B + V = H_A \otimes \mathbf{1} + \mathbf{1} \otimes H_B + V$$

$$U \neq U_A \otimes U_B$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a\begin{bmatrix} e & f \\ g & h \end{bmatrix} & b\begin{bmatrix} e & f \\ g & h \end{bmatrix} \\ c\begin{bmatrix} e & f \\ g & h \end{bmatrix} & d\begin{bmatrix} e & f \\ g & h \end{bmatrix} \end{bmatrix} = \begin{bmatrix} ae & af & be & bf \\ ag & ah & fg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{bmatrix}$$

Universal quantum computing

Universal quantum computing

General structure of every quantum algorithm

Universal quantum computer = machine which allows you to apply a $2^n \times 2^n$ unitary constructed from 2×2 and 4×4 unitaries to vector $[1,0,\ldots,0]^{\mathrm{T}}$ and sample random variable from the probability distribution determined by the resulting vector.

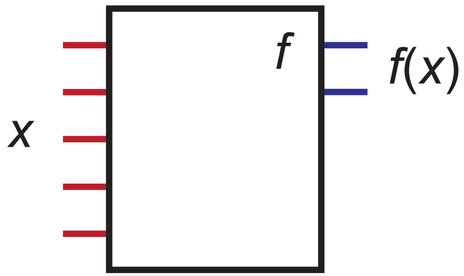
The procedure

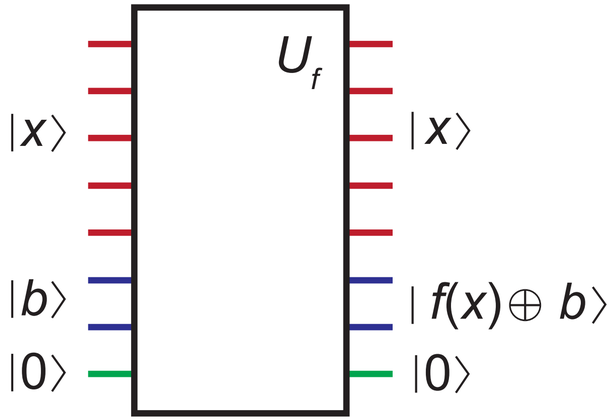
1. Design a n-qubit unitary U you want to implement ($n \leq n_{\text{available}}$)
2. Decompose U into sequence $G_M \ldots G_1$ of gates which your quantum computer can perform
3. Apply $G_M \ldots G_1$ to the initial qubit state $|0\ldots0\rangle$
4. Obtain some random bit-string $\mathbf{x} \in \{0,1\}^n$ by measuring the obtained state $|\psi\rangle = G_M \ldots G_1 |0\ldots0\rangle$ ($\Pr(\mathbf{x}) = |\langle \mathbf{x} | \psi \rangle|^2$)
5. ???? (repeat the sampling / update unitary / make post-processing / etc)
6. Profit!!!

Universal quantum computing

How to implement arbitrary Boolean function *f* with reversible unitary *U*

$f : \{0,1\}^N \to \{0,1\}^M$

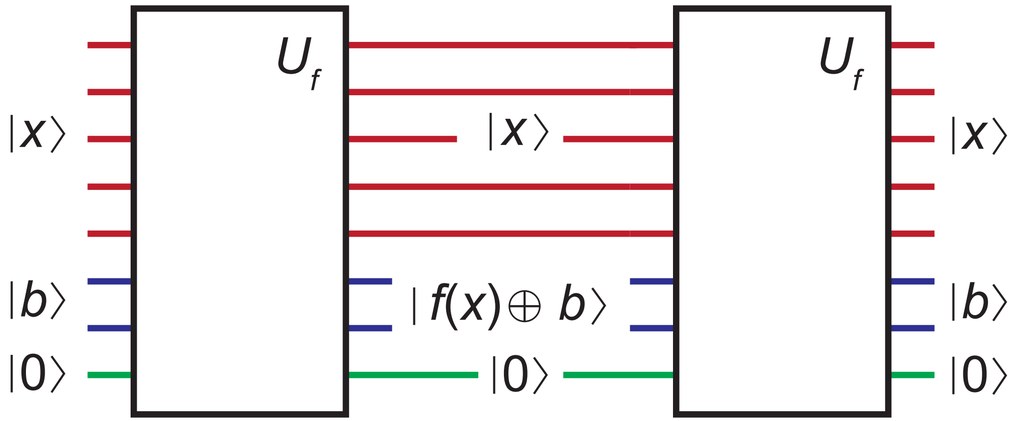




$x \in \{0,1\}^N, \quad b \in \{0,1\}^M$

$(a_0, \ldots a_{M-1}) \oplus (b_0, \ldots b_{M-1})$

$\equiv (a_0 \oplus b_0, \ldots a_{M-1} \oplus b_{M-1})$



$$0 \oplus 0 = 0$$
$$1 \oplus 0 = 1$$
$$0 \oplus 1 = 1$$
$$1 \oplus 1 = 0$$

$$U_f\left(C_1 |x_1 t_1\rangle + C_2 |x_2 t_2\rangle\right)$$
$$= C_1 U_f |x_1 t_1\rangle + C_2 U_f |x_2 t_2\rangle)$$

Universal quantum computing

Zoo of quantum gates

Pauli-X (X)	X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	
Pauli-Y (Y)	Y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	
Pauli-Z (Z)	Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	
Phase (S, P)	S	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$	
$\pi/8$ (T)	T	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$	
Controlled Not (CNOT, CX)			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)	Z		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)			$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

$U \sim e^{i\phi}U$ but $C[U] \nsim C[e^{i\phi}U]$!!

$(C[W] = |0\rangle\langle 0| \otimes \mathbf{1} + |1\rangle\langle 1| \otimes W)$

Universal quantum computing

Solovay-Kitaev theorem [arXiv:quant-ph/0505030]

Definition 1 *An* ***instruction set*** $\mathcal{G}$ *for a d-dimensional qudit is a* finite *set of quantum gates satisfying:*

1. *All gates* $g \in \mathcal{G}$ *are in* $SU(d)$*, that is, they are unitary and have determinant* 1*.*
2. *For each* $g \in \mathcal{G}$ *the inverse operation* $g^\dagger$ *is also in* $\mathcal{G}$*.*
3. $\mathcal{G}$ *is a universal set for* $SU(d)$*, i.e., the group generated by* $\mathcal{G}$ *is dense in* $SU(d)$*. This means that given any quantum gate* $U \in SU(d)$ *and any accuracy* $\epsilon > 0$ *there exists a product* $S \equiv g_1 \dots g_m$ *of gates from* $\mathcal{G}$ *which is an* ϵ*-approximation to* U*.*

Theorem 1 (Solovay-Kitaev) *Let* $\mathcal{G}$ *be an instruction set for* $SU(d)$*, and let a desired accuracy* $\epsilon > 0$ *be given. There is a constant* c *such that for any* $U \in SU(d)$ *there exists a finite sequence* S *of gates from* $\mathcal{G}$ *of length* $O(\log^c(1/\epsilon))$ *and such that* $d(U, S) < \epsilon$*.*

Different proofs of the theorem give different values for c; the proof we describe gives $c \approx 3.97$. We discuss some other proofs (and their corresponding values of c) in Section 6.

Universal quantum computing

Solovay-Kitaev theorem [arXiv:quant-ph/0505030]

Definition 1 *An* ***instruction set*** $\mathcal{G}$ *for a d-dimensional qudit is a* finite *set of quantum gates satisfying:*

1. *All gates* $g \in \mathcal{G}$ *are in* $SU(d)$*, that is, they are unitary and have determinant* 1.

2. *For each* $g \in \mathcal{G}$ *the inverse operation* $g^\dagger$ *is also in* $\mathcal{G}$.

If you have some «universal set of gates» then any unitary can be decomposed with any desirable accuracy ϵ «efficiently»:

gates ~ $O(\log^{3.97}(1/\epsilon))$.

Theorem 1 (Solovay-Kitaev) *Let* $\mathcal{G}$ *be an instruction set for* $SU(d)$*, and let a desired accuracy* $\epsilon > 0$ *be given. There is a constant* c *such that for any* $U \in SU(d)$ *there exists a finite sequence* S *of gates from* $\mathcal{G}$ *of length* $O(\log^c(1/\epsilon))$ *and such that* $d(U, S) < \epsilon$.

Different proofs of the theorem give different values for c; the proof we describe gives $c \approx 3.97$. We discuss some other proofs (and their corresponding values of c) in Section 6.

Universal quantum computing

Common universal gate sets

Single-qubit gates

$$R_x(\theta) = e^{-i\sigma_x\theta/2} = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

$$R_y(\theta) = e^{-i\sigma_y\theta/2} = \begin{bmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

$$R_z(\theta) = e^{-i\sigma_y\theta/2} = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

Two-qubit gates

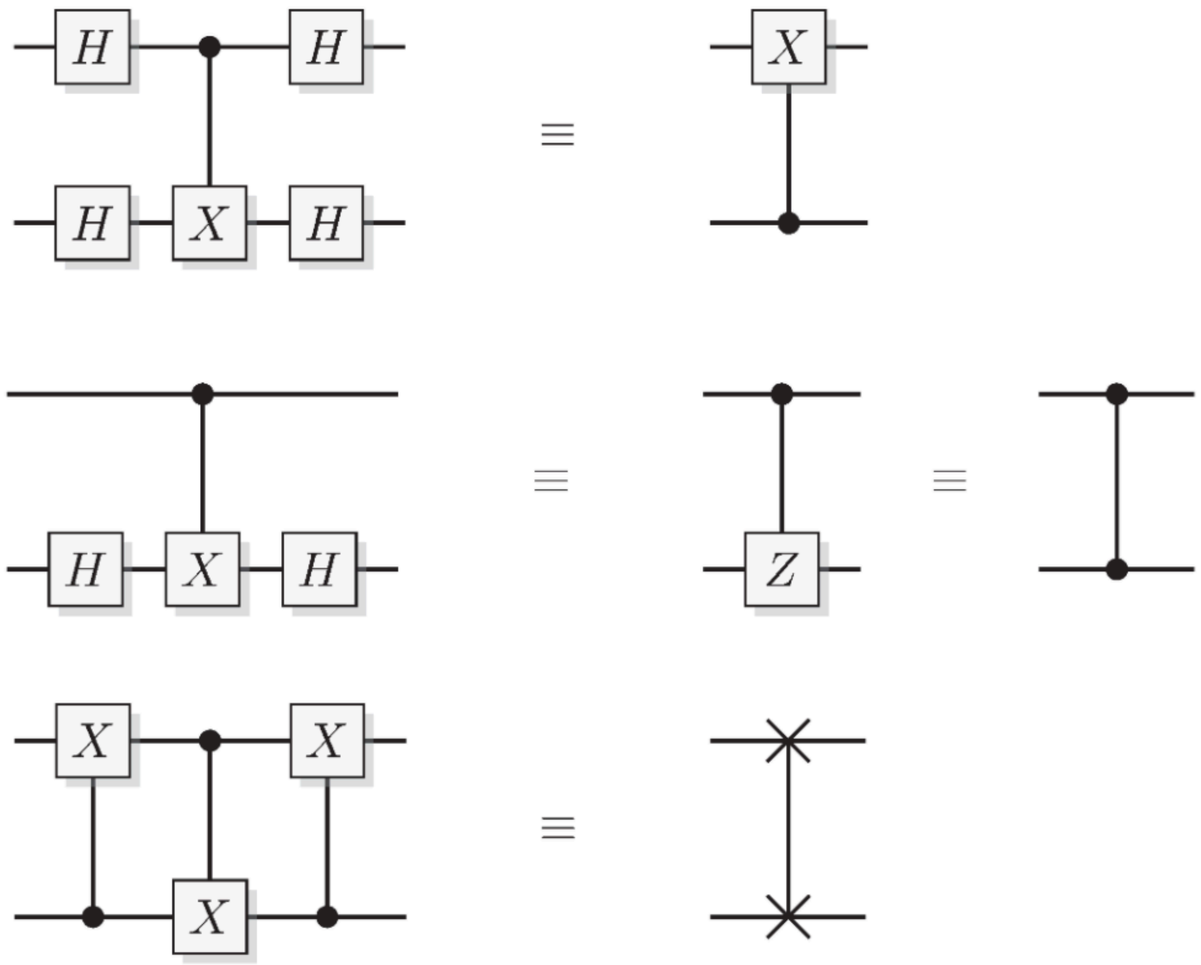
$$CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{MS} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & i \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ i & 0 & 0 & 1 \end{bmatrix}$$

$$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad \text{iSWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

with single-qubit gates

Universal quantum computing

Some useful gate identities



N. Johansson, J.-Å. Larsson, Entropy 21(8), 800 (2019)

Universal quantum computing

Super-conducting processor from IBM

ibmq_16_melbourne v2.3.0

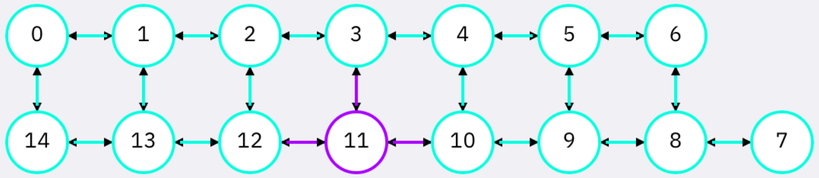
online

Queue: 18 jobs

Providers with access:

ibm-q/open/main

Qubits Connectivity



Single-qubit U2 error rate

4.962e-4 — 5.090e-2

CNOT error rate

1.198e-2 — 1.000e+0

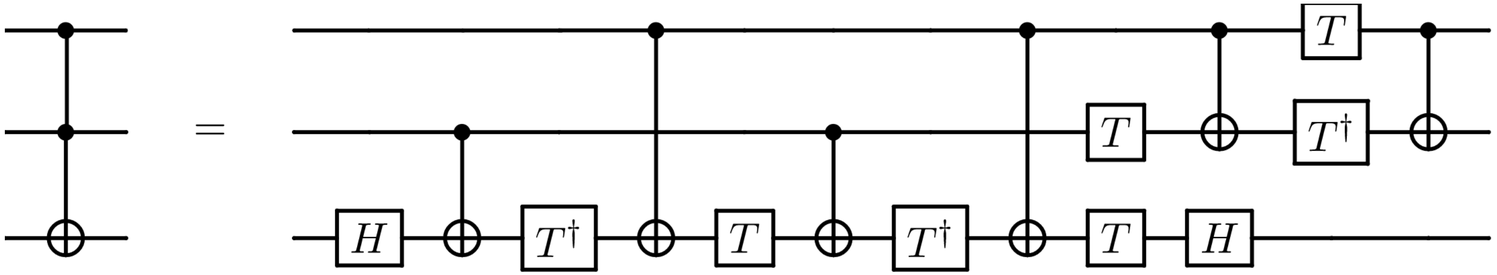
Download Calibrations

Qubits 15	Quantum Volume 8
Online since Nov 06, 2018	Basis gates id, u1, u2, u3, cx
Maximum shots 8192	Maximum circuits 75
Last calibration update Sep 02, 2020 1:06 PM	

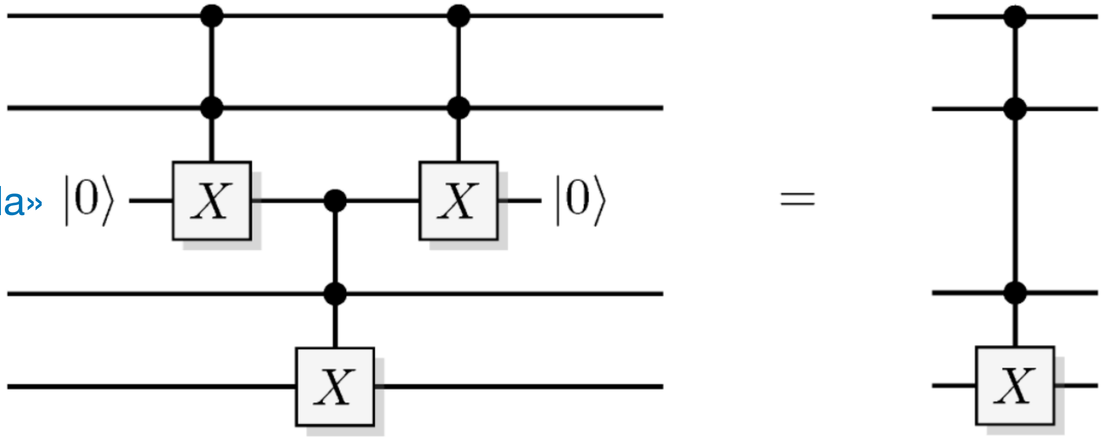
quantum-computing.ibm.com

Universal quantum computing

Using ancillary qubits for gate decomposition



«Clean ancilla»



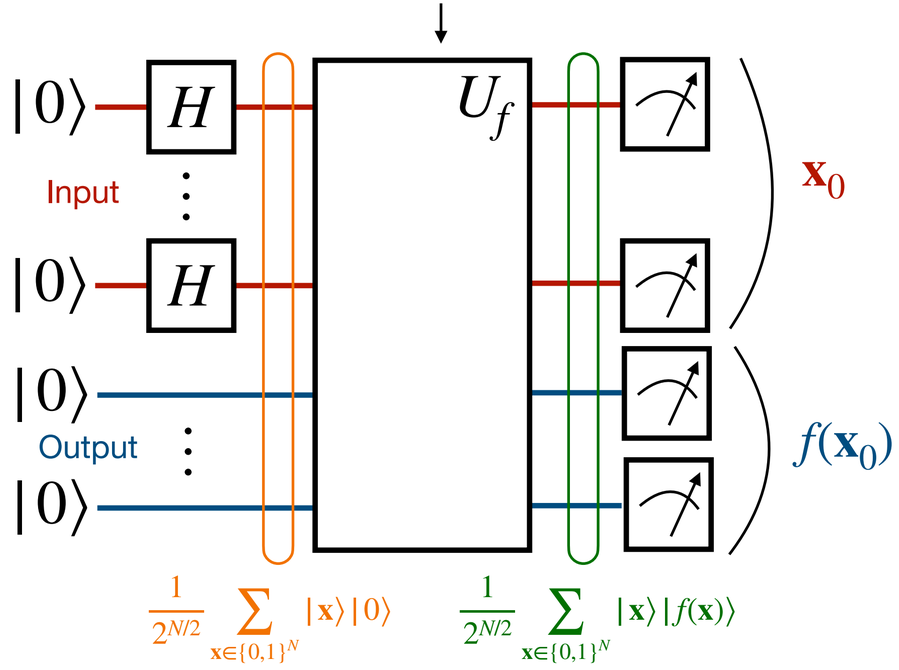
Ancillary qubits can be substituted by qu**D**its [Phys. Rev. A 101, 022304 (2020)]
Ask Nastya Nikolaeva near her poster «Quantum computing with qudits» about details!

Universal quantum computing

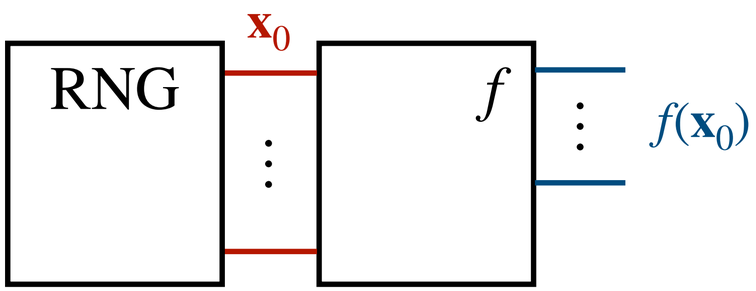
How quantum computer doesn't work

$$f : \{0,1\}^N \to \{0,1\}^M$$

Computes f for all 2^N possible inputs simultaneously!



=



Be careful about «quantum parallelism»!

$$H^{\otimes N}|0\rangle^{\otimes N} = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)^{\otimes N} = \frac{1}{2^{N/2}} \sum_{\mathbf{x}\in\{0,1\}^N} |\mathbf{x}\rangle$$

Universal quantum computing

How quantum computer does work: Grover's algorithm

$$\mathbf{x} \rightarrow \boxed{\;f\;} \rightarrow f(\mathbf{x})$$

x	f(**x**)
«000»	0
«001»	0
«010»	0
«011»	0
w = «100»	1
«101»	0
«110»	0
«111»	0

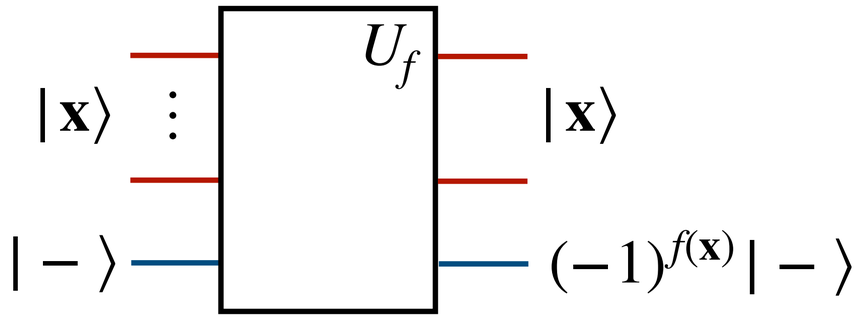
Problem: find **w** with minimal number queries to *f*

Classical solution («brute-force»): $O(2^n)$

Quantum solution (Grover): $O(2^{n/2})$

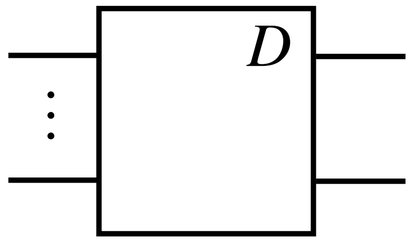
Universal quantum computing

Phase request & diffusion operator



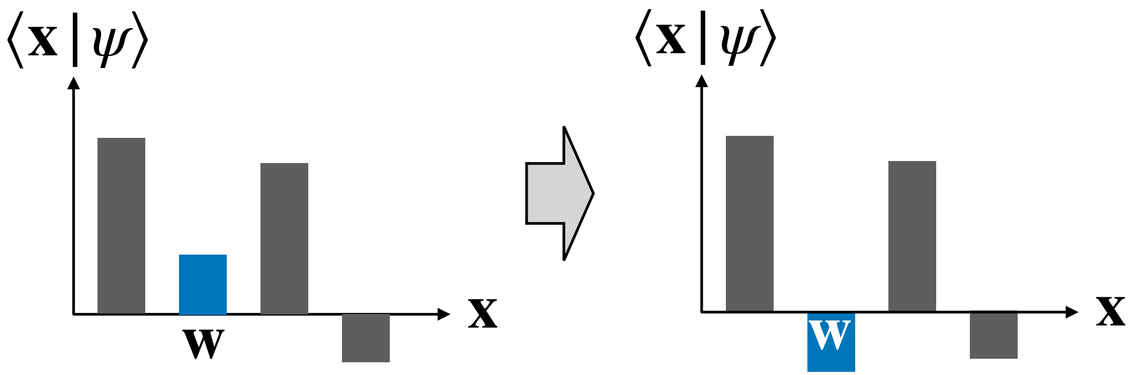
$$|-\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

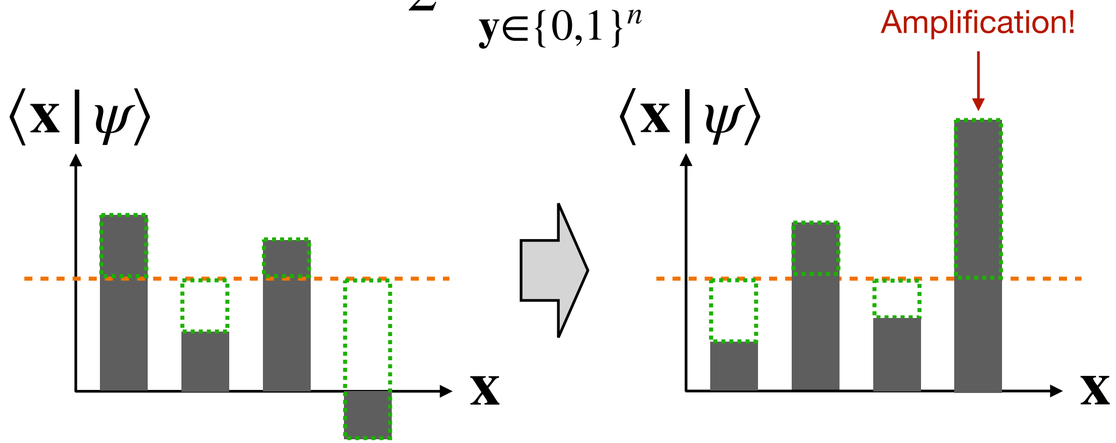
Diffusion operator = rotation around mean



$$\langle \mathbf{x}|\psi\rangle \to 2C_{\text{mean}} - \langle \mathbf{x}|\psi\rangle$$

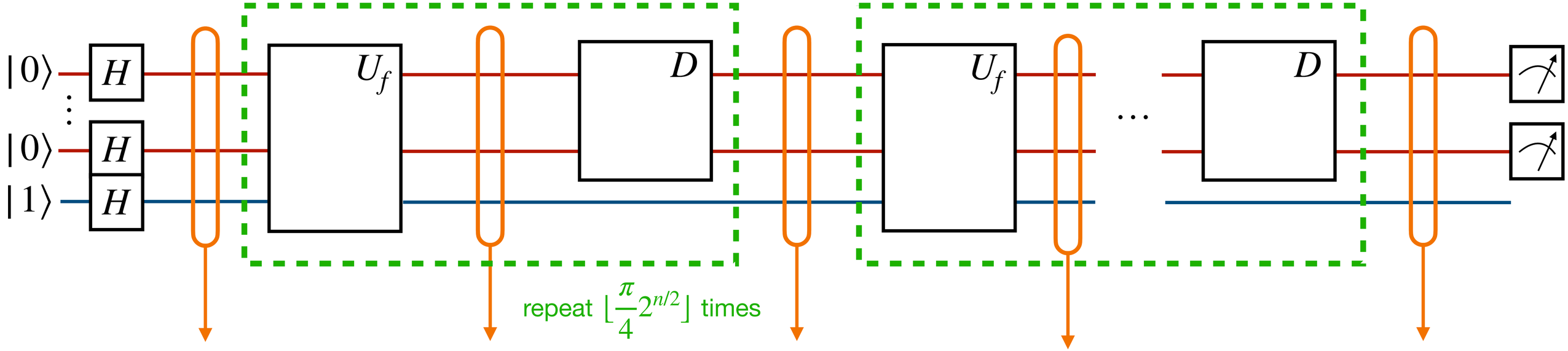
$$C_{\text{mean}} = \frac{1}{2^n} \sum_{\mathbf{y}\in\{0,1\}^n} \langle \mathbf{y}|\psi\rangle$$

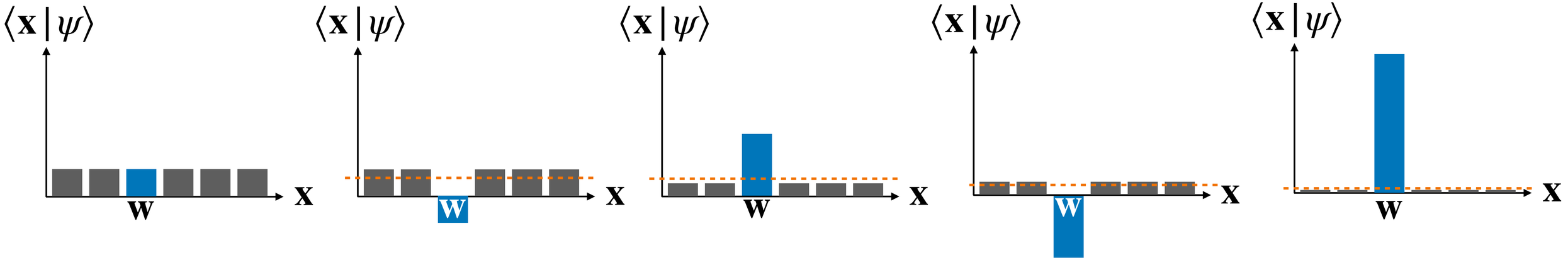




Universal quantum computing

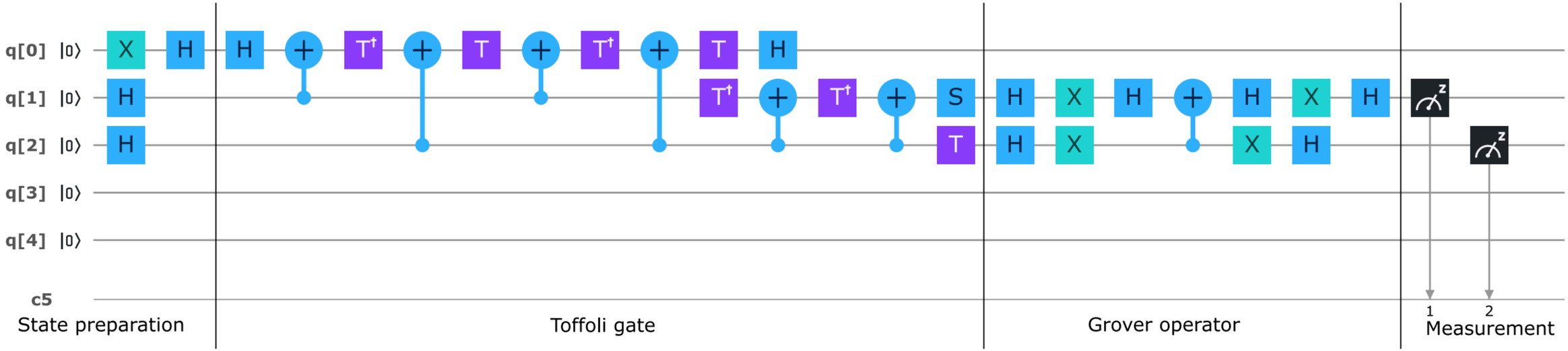
Amplitude amplification in Grover's algorithm





Universal quantum computing

Resulting 3 qubit circuit of Grover's algorithm for *n* = 2

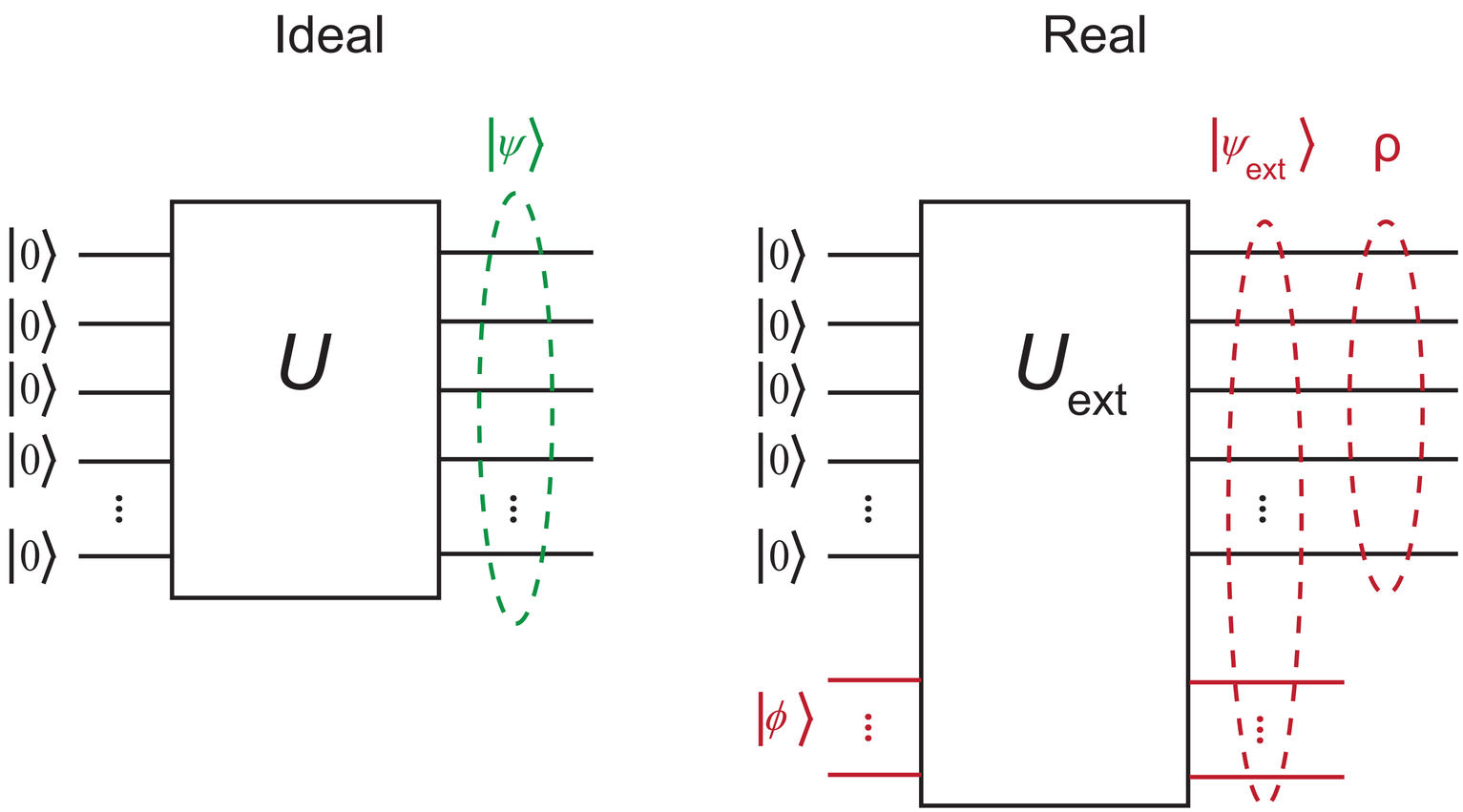


«Quantum Algorithm Implementations for Beginners» arXiv:1804.03719

Noisy Intermediate-Scale Quantum (NISQ) devices

NISQ devices

Decoherence problem

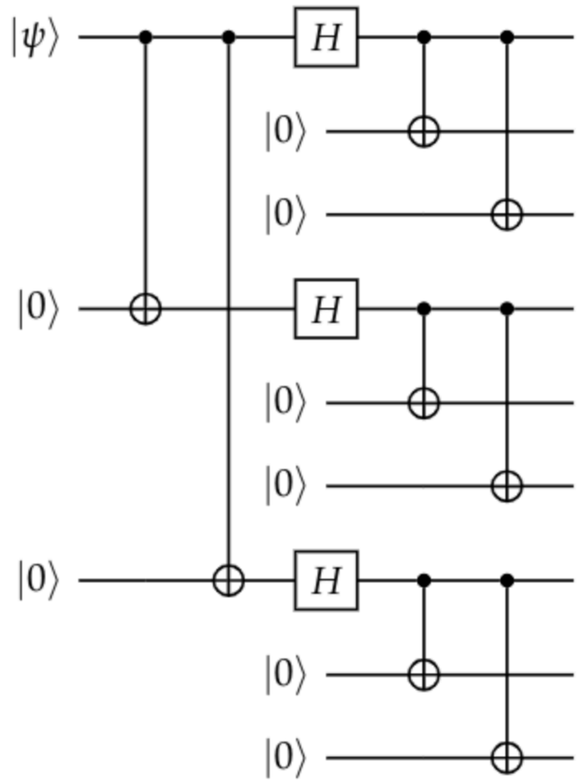


NISQ devices

Error correcting codes

General idea: encode quantum information in **non-local states** in order to tolerate **local errors**

Shor's code = quantum analog of the repetition code (0→000, 1→111)



$$\alpha|0\rangle+\beta|1\rangle \rightarrow \alpha(|000\rangle+|111\rangle)\otimes(|000\rangle+|111\rangle)\otimes(|000\rangle+|111\rangle)+$$
$$\beta(|000\rangle-|111\rangle)\otimes(|000\rangle-|111\rangle)\otimes(|000\rangle-|111\rangle)$$

Bit-flip error:

$$\alpha(|000\rangle+|111\rangle)\otimes(|010\rangle+|101\rangle)\otimes(|000\rangle+|111\rangle)+$$
$$\beta(|000\rangle-|111\rangle)\otimes(|010\rangle-|101\rangle)\otimes(|000\rangle-|111\rangle)$$

Phase-flip error:

$$\alpha(|000\rangle-|111\rangle)\otimes(|010\rangle+|101\rangle)\otimes(|000\rangle+|111\rangle)+$$
$$\beta(|000\rangle+|111\rangle)\otimes(|010\rangle-|101\rangle)\otimes(|000\rangle-|111\rangle)$$

NISQ devices

Recent experimental results in the quantum error correction

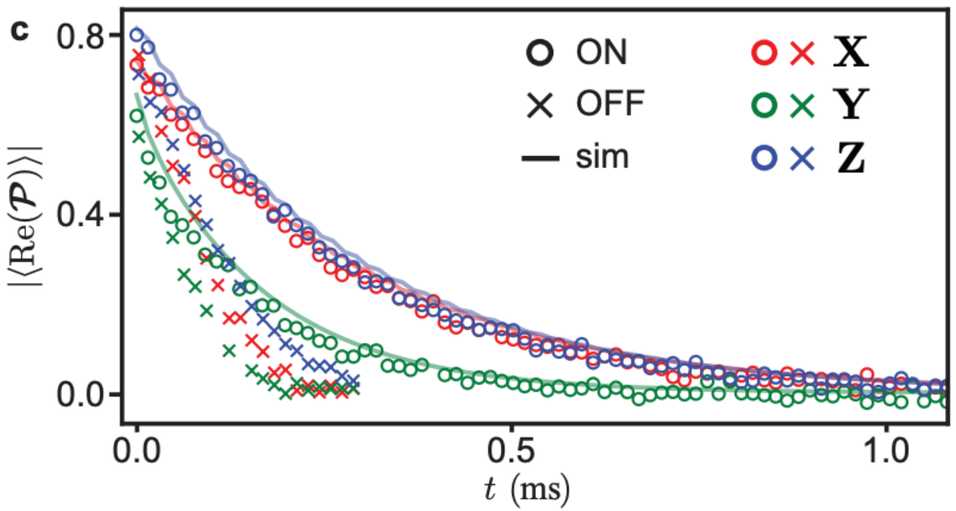
Article

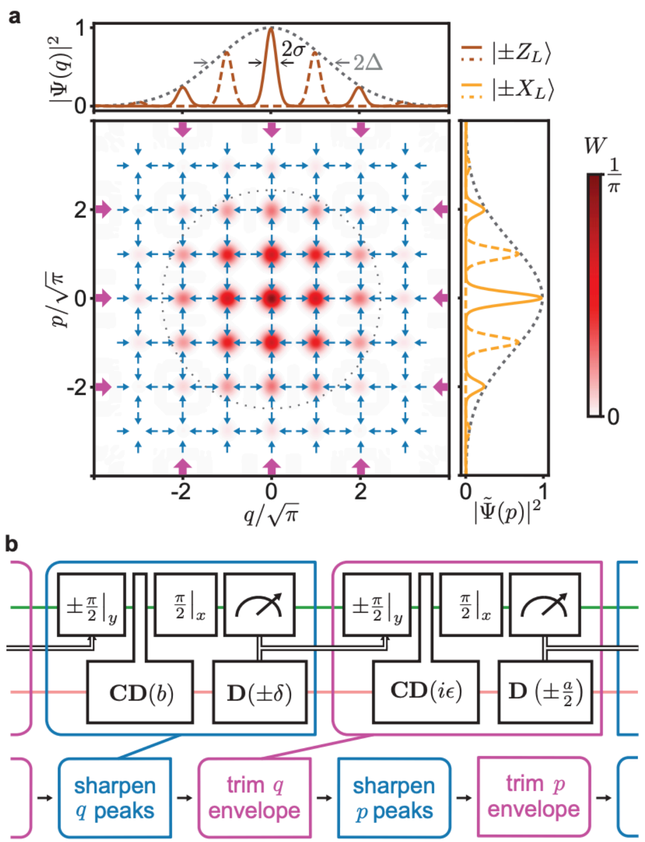
Quantum error correction of a qubit encoded in grid states of an oscillator

https://doi.org/10.1038/s41586-020-2603-3
Received: 8 November 2019
Accepted: 12 June 2020
Published online: 19 August 2020
Check for updates

P. Campagne-Ibarcq[1,3,6], A. Eickbusch[1,6], S. Touzard[1,4,6], E. Zalys-Geller[1], N. E. Frattini[1], V. V. Sivak[1], P. Reinhold[1], S. Puri[1], S. Shankar[1,5], R. J. Schoelkopf[1], L. Frunzio[1], M. Mirrahimi[2] & M. H. Devoret[1]

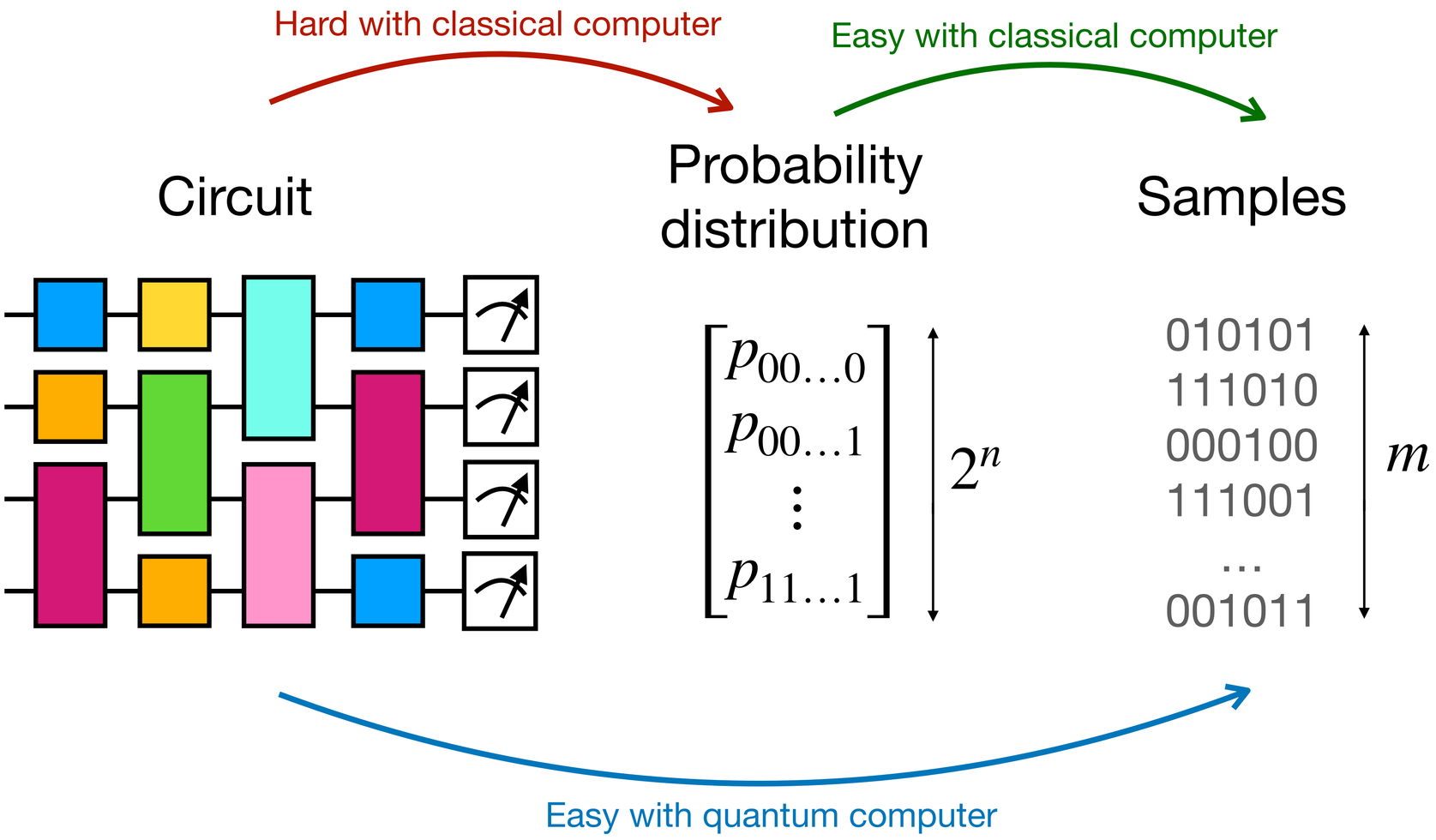
The accuracy of logical operations on quantum bits (qubits) must be improved for quantum computers to outperform classical ones in useful tasks. One method to





NISQ devices

Once again about basic idea



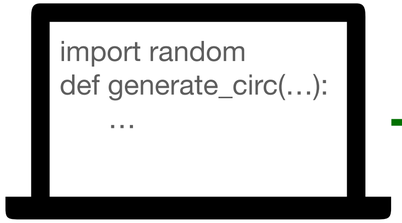
NISQ devices

Introducing additional layer of randomness

Hard with classical computer

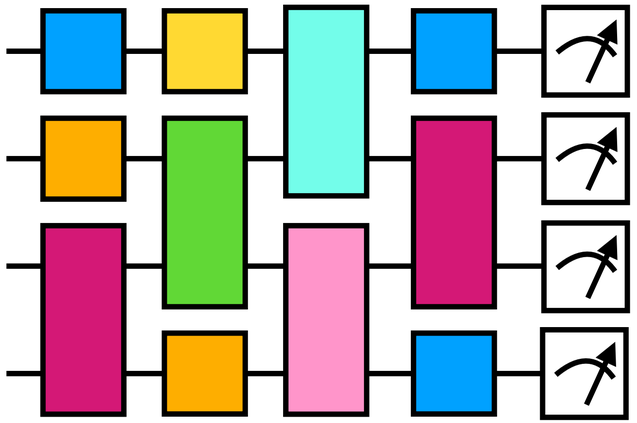
Can be verified!

Circuit generator



Easy with classical computer

Random circuit



Random probability distribution*

$$\begin{bmatrix} p_{00\ldots0} \\ p_{00\ldots1} \\ \vdots \\ p_{11\ldots1} \end{bmatrix} \updownarrow 2^n$$

Random samples from random distribution

010101
111010
000100
111001
...
001011

$\updownarrow m$

Easy with quantum computer
But is it correct??

*Mind the decoherence!

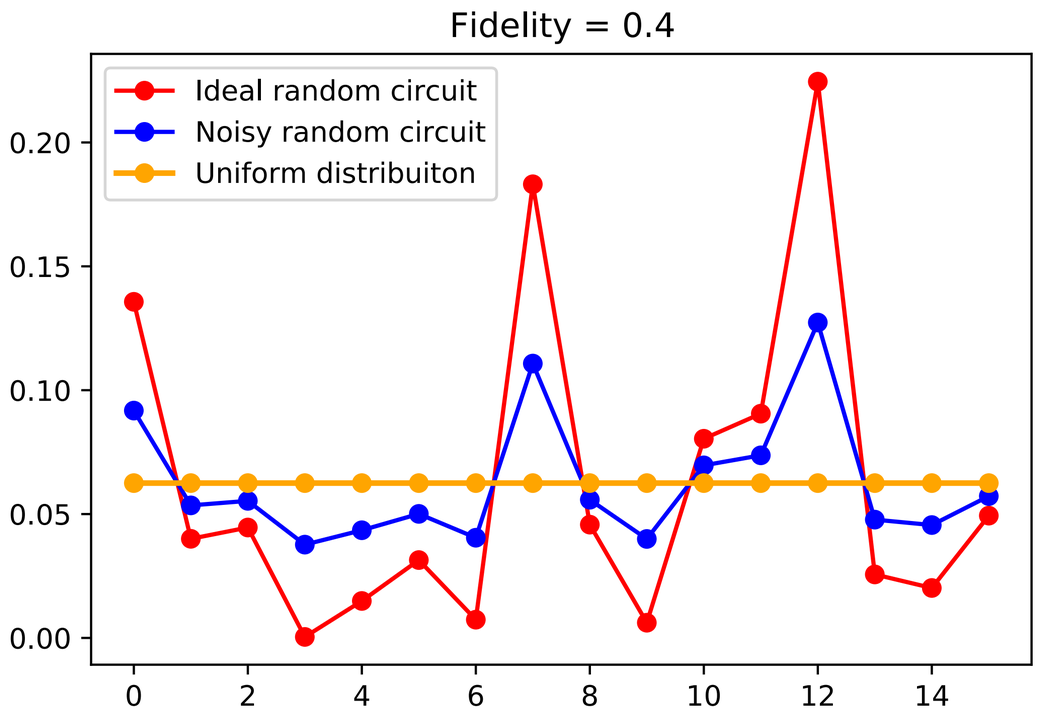
NISQ devices

Taking decoherence into account

Depolarizing noise
= «white noise»

Fidelity

$$|\psi\rangle \Rightarrow F|\psi\rangle\langle\psi| + (1-F)\frac{\mathbf{1}}{2^n}$$



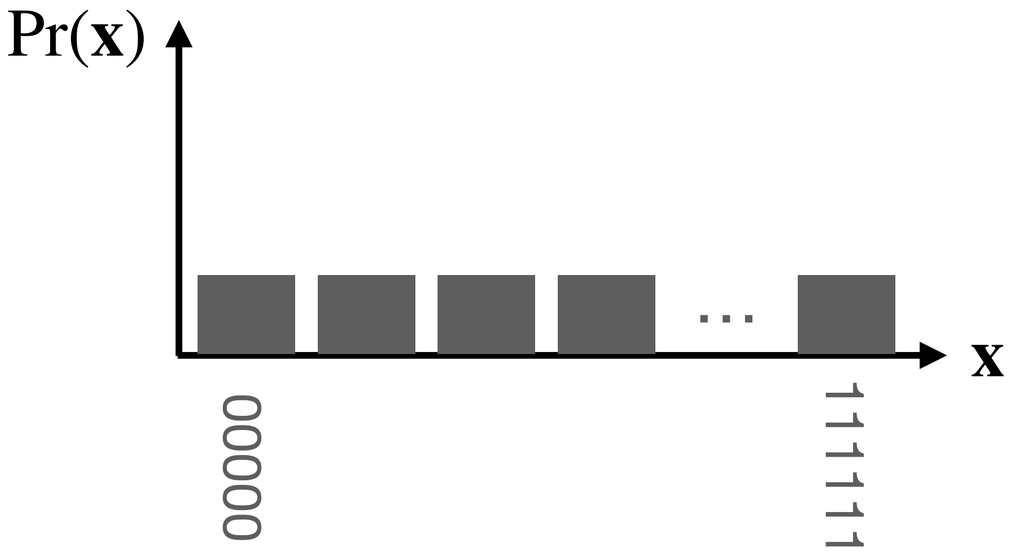
"True" fidelity: $F(|\psi\rangle, \rho) = \langle\psi|\rho|\psi\rangle$

NISQ devices

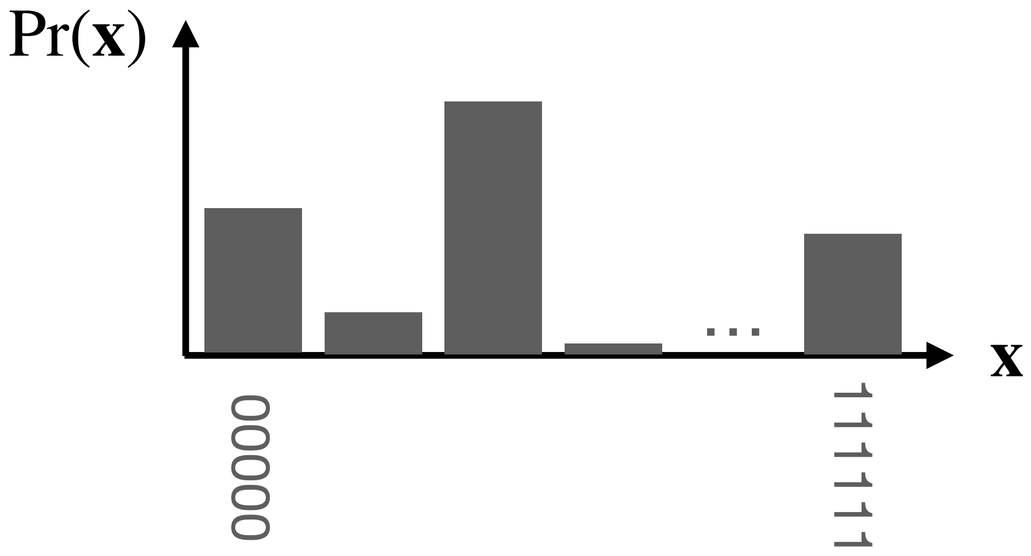
How to check distribution: statistical tests

{010101, 111010, 000100, 111001, 001011, 010101, 001111, 111000}

samples $\ll 2^n$



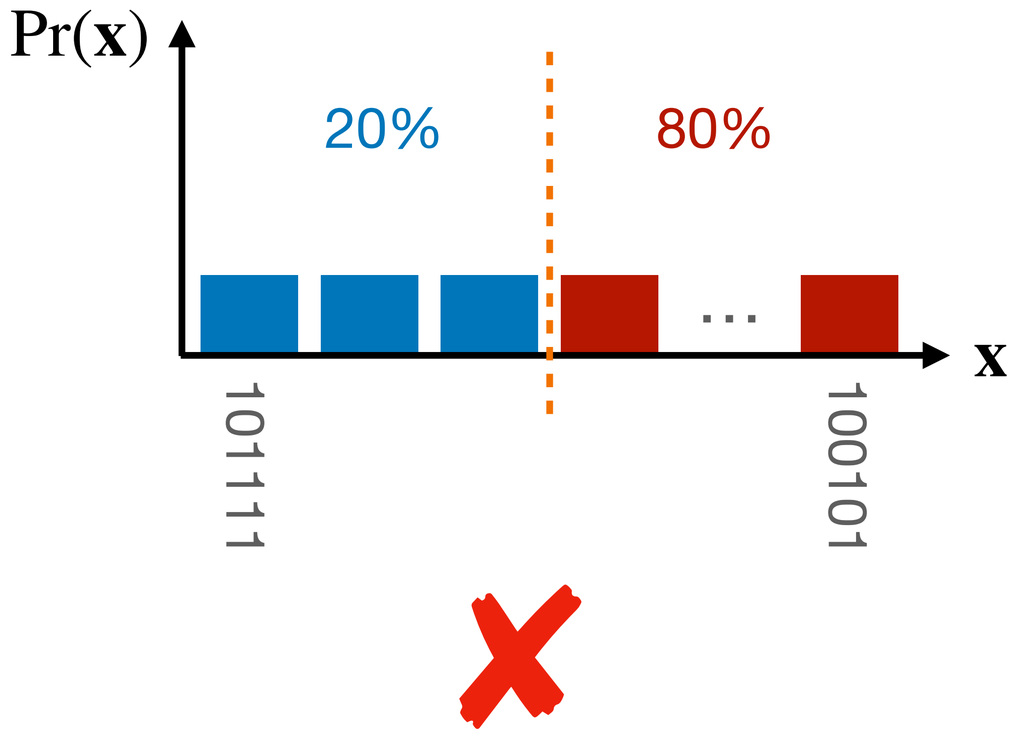
OR



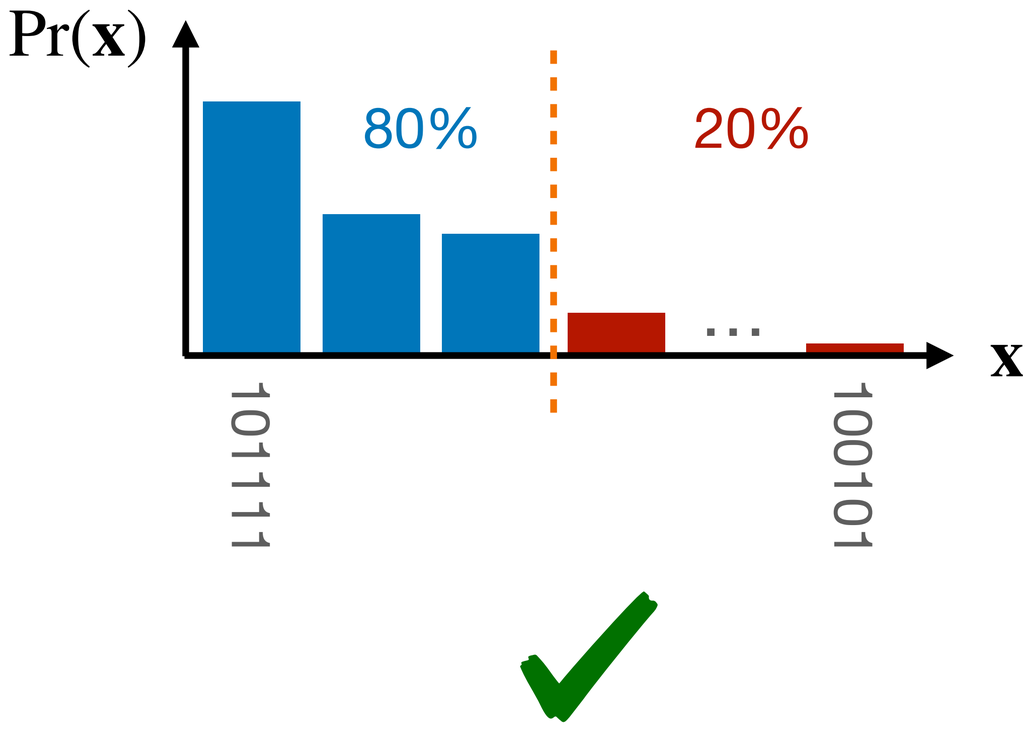
NISQ devices

How to check distribution: statistical tests

{010101, 111010, 000100, 111001, 001011, 010101, 001111, 111000}



OR



NISQ devices

More clever statistical test: linear cross-entropy benchmark (XEB)

$$\text{samples} = (\mathbf{x}_0, \mathbf{x}_1, \ldots \mathbf{x}_m), \quad m \ll 2^n$$

$$\mathscr{F}_{\mathrm{XEB}} = 2^n \langle \Pr(\mathbf{x}_i) \rangle_{\text{samples}} - 1 = \begin{cases} 1 & \text{for ideal random circuit,} \\ 0 & \text{for unifrom distribution,} \\ F & \text{for noisy random circuit,} \end{cases}$$

$$\Pr(\mathbf{x}_i) = \langle \mathbf{x}_i | U | 0 \rangle^{\otimes n}$$

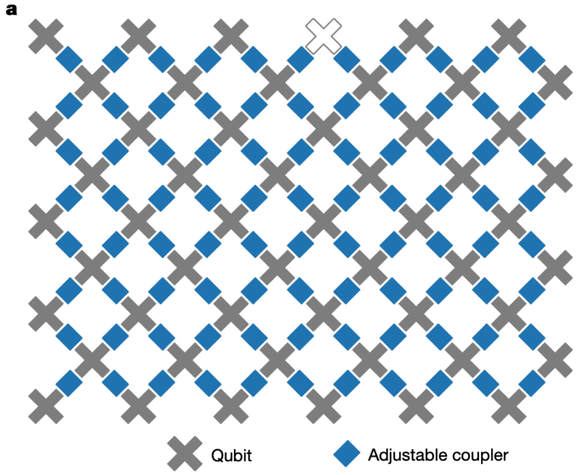
The procedure of verifying quantum advantage

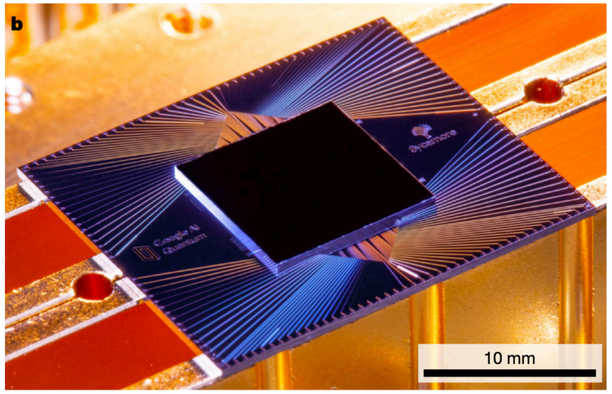
1. Generate random circuit (easy with classical computer)
2. Sample $(\mathbf{x}_0, \ldots \mathbf{x}_m)$ with quantum computer (easy with quantum computer)
3. Compute $(\Pr(\mathbf{x}_0), \ldots \Pr(\mathbf{x}_m))$ and $\mathscr{F}_{\mathrm{XEB}}$ (very difficult with classical computer for «big» n)
4. If $\mathscr{F}_{\mathrm{XEB}} > 0$ with enough confidence then profit!

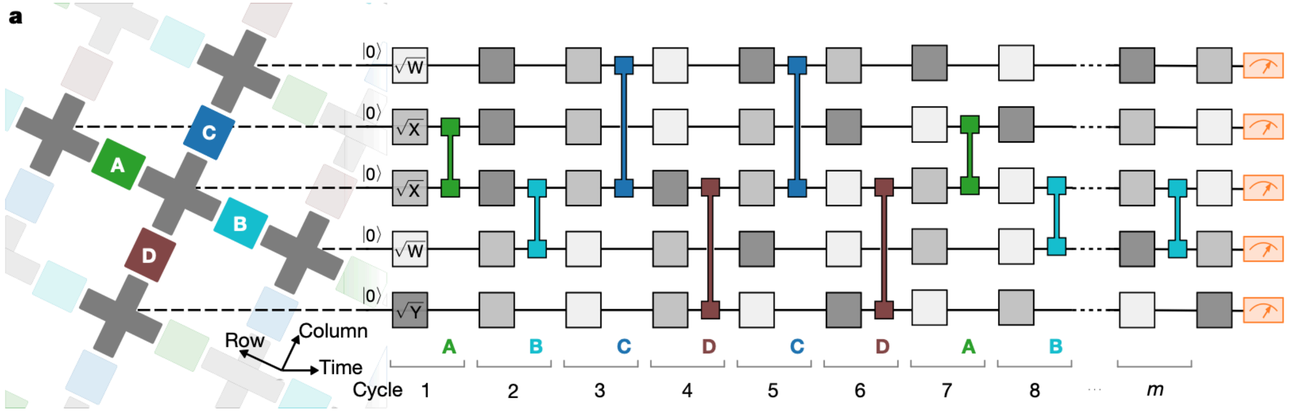
NISQ devices

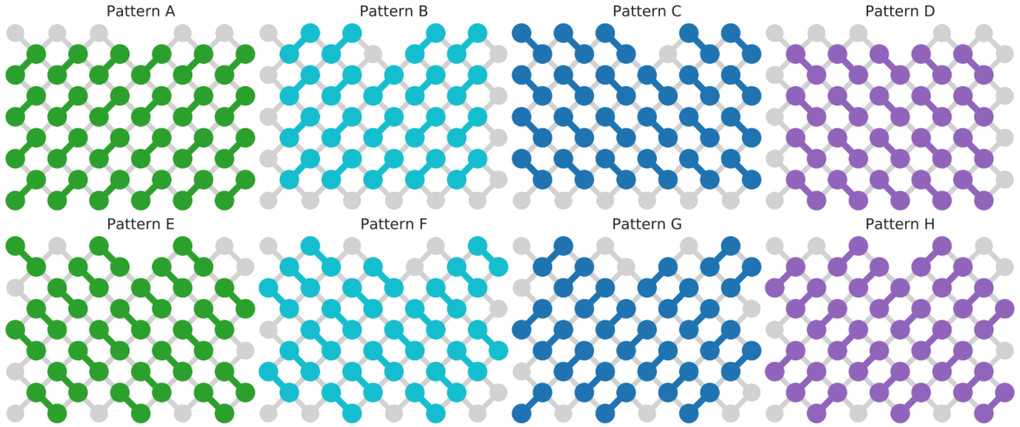
Demonstrating quantum advantage by Google

[F. Arute, K. Arya, [...] John M. Martinis, Nature **574**, 505–510 (2019)]





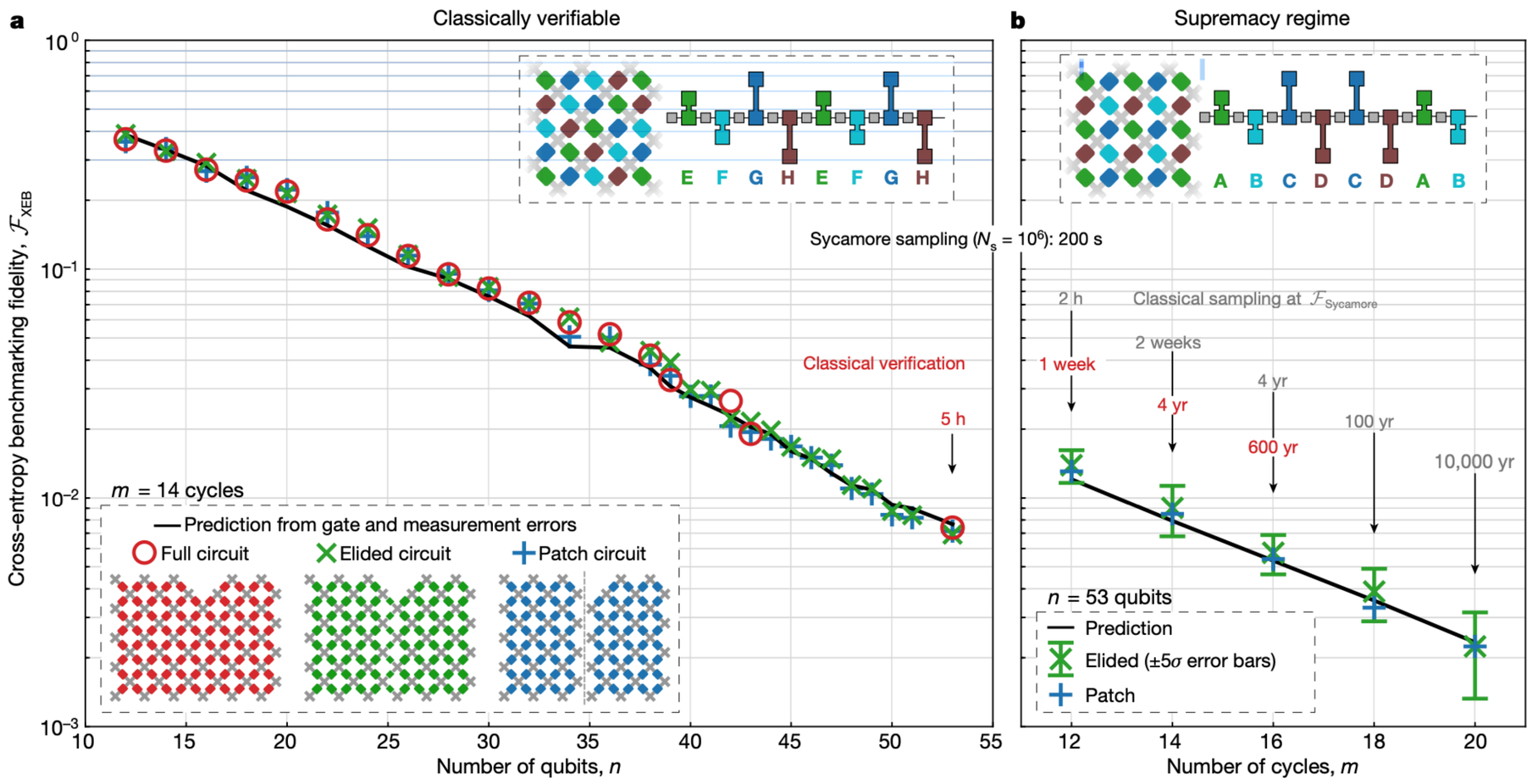




NISQ devices

Demonstrating quantum advantage by Google

[F. Arute, K. Arya, [...] John M. Martinis, Nature **574**, 505–510 (2019)]



NISQ devices

Python notebook on XEB tests

master RQC-MIPT-q-inf-proc-course / Python / XEB_playground.ipynb Go to file

EvgeniyKiktenko Updating notebook ... Latest commit 253f664 11 seconds ago History

1 contributor

351 lines (351 sloc) | 60.3 KB Raw Blame

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.linalg
```

In [2]:
```
# Some basic operators
sigma = np.array([np.eye(2, dtype=complex) for _ in range(4)])
sigma[1] = np.array([[0,1],[1,0]])
sigma[2] = np.array([[0,-1j],[1j,0]])
sigma[3] = np.array([[1,0],[0,-1]])

def rx_gate(theta):
    return scipy.linalg.expm(-1j * sigma[1] * theta / 2)

def ry_gate(theta):
    return scipy.linalg.expm(-1j * sigma[2] * theta / 2)

def rz_gate(theta):
    return scipy.linalg.expm(-1j * sigma[3] * theta / 2)

cz_gate = np.eye(4); cz_gate[3,3] = -1

print(f'CZ gate:\n{cz_gate}')
```

```
CZ gate:
[[ 1.  0.  0.  0.]
 [ 0.  1.  0.  0.]
 [ 0.  0.  1.  0.]
 [ 0.  0.  0. -1.]]
```

NISQ devices

Google VS IBM

https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/

IBM and a device described by Google in a paper published in the journal *Nature*. In the paper, it is argued that their device reached "quantum supremacy" and that "a state-of-the-art supercomputer would require approximately 10,000 years to perform the equivalent task." *We argue that an ideal simulation of the same task can be performed on a classical system in 2.5 days and with far greater fidelity*. This is in fact a

F. Arute, K. Arya, [...] John M. Martinis, Nature 574, 505–510 (2019)

past few years[42–50]. We expect that lower simulation costs than reported here will eventually be achieved, but we also expect that they will be consistently outpaced by hardware improvements on larger quantum processors.

Adiabatic quantum computing

Adiabatic quantum computing

General idea

$$H_t = (1 - p_t)H_0 + p_t H_1$$

Easy to prepare

$$H_0 |\psi_{\mathrm{gr}}^{(0)}\rangle = E_{\mathrm{gr}}^{(0)} |\psi_{\mathrm{gr}}^{(0)}\rangle$$

Contains solution of the problem

$$H_1 |\psi_{\mathrm{gr}}^{(1)}\rangle = E_{\mathrm{gr}}^{(1)} |\psi_{\mathrm{gr}}^{(1)}\rangle$$

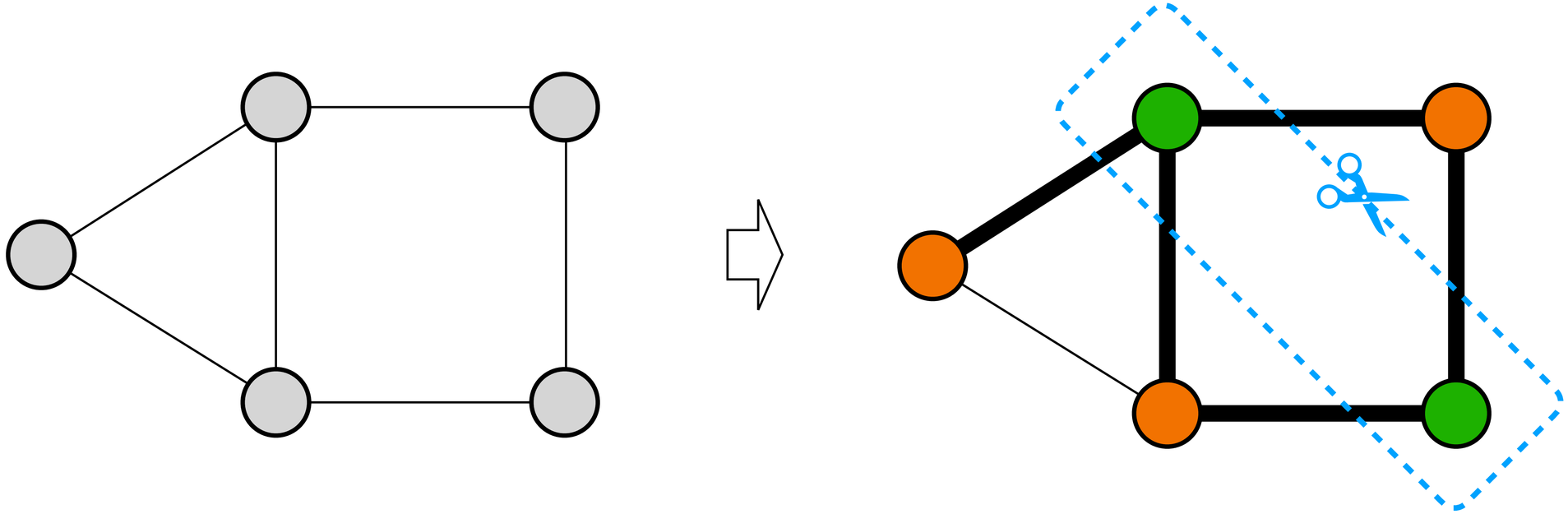
$$\frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = -iH_t |\psi(t)\rangle$$

«**Adiabatic theorem**»: if p_t goes «slowly enough» from 0 to 1 then $|\psi_{\mathrm{gr}}^{(0)}\rangle \to |\psi_{\mathrm{gr}}^{(1)}\rangle$.

Adiabatic quantum computing

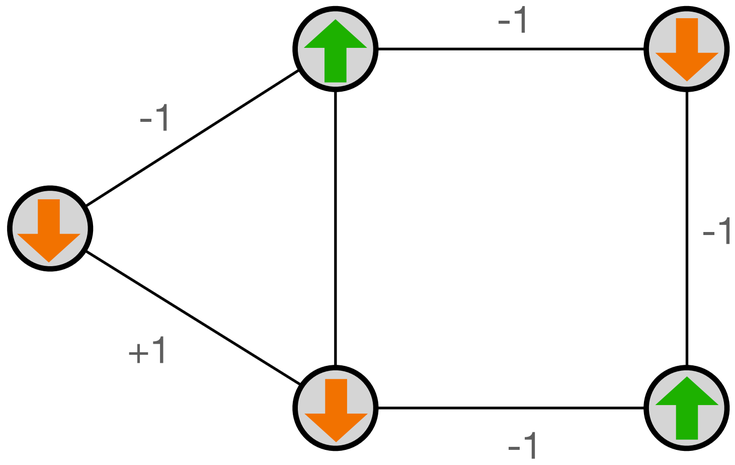
How ground state can solve a problem: MAX-CUT case

MAX-CUT problem: Given a graph *G* and an integer *k*, determine whether there is a cut of size at least *k* in *G.*
Known to be NP-complete!



Adiabatic quantum computing

From abstract graphs to physical spins



$$H_1 = \sum_{<i,j>\in G} \sigma_z^{(i)}\sigma_z^{(j)}$$

$$\langle \uparrow |_i \langle \uparrow |_j \sigma_z^{(i)}\sigma_z^{(j)} | \uparrow \rangle_i | \uparrow \rangle_j = \langle \downarrow |_i \langle \downarrow |_j \sigma_z^{(i)}\sigma_z^{(j)} | \downarrow \rangle_i | \downarrow \rangle_j = 1$$

$$\langle \uparrow |_i \langle \downarrow |_j \sigma_z^{(i)}\sigma_z^{(j)} | \uparrow \rangle_i | \downarrow \rangle_j = \langle \downarrow |_i \langle \uparrow |_j \sigma_z^{(i)}\sigma_z^{(j)} | \downarrow \rangle_i | \uparrow \rangle_j = -1$$

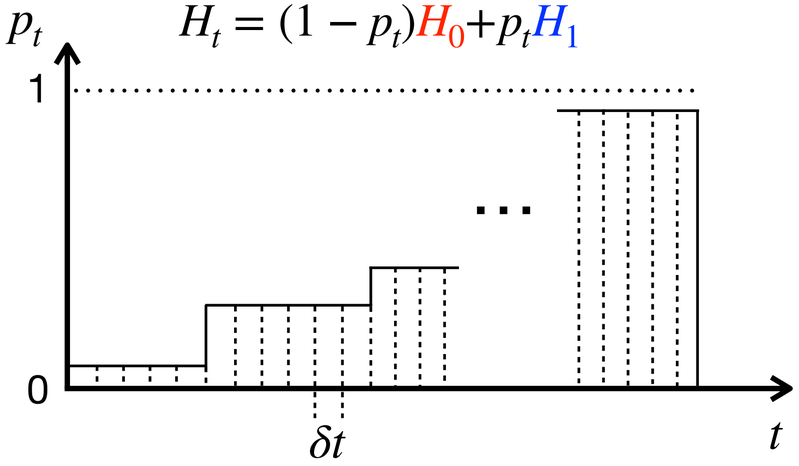
Ground state of H_1 is exactly the solution of the MAX-CUT problem!

$$\boxed{\sigma_z^{(i)}\sigma_z^{(j)} \equiv \underbrace{\mathbf{1} \otimes \ldots \otimes \mathbf{1} \otimes \overbrace{\sigma_z}^{i} \otimes \ldots \otimes \sigma_z}_{j} \otimes \ldots \otimes \mathbf{1}}$$

Adiabatic quantum computing

Discretizing adiabatic evolution

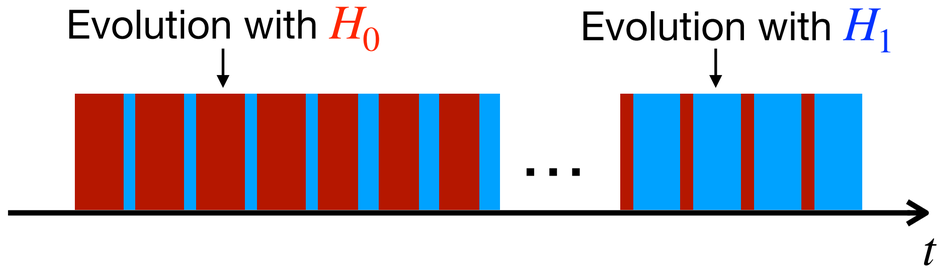
$H_t = (1 - p_t)H_0 + p_t H_1$



$$U = \prod_{j=1}^{N} e^{-i[(1-p_{t_j})H_0\delta t + p_{t_j}H_1\delta t]}$$

$$e^{A+B} = \lim_{n\to\infty} (e^{A/n}e^{B/n})^n \approx (e^{A/N}e^{B/N})^N$$ ⟵ Lie–Trotter product formula

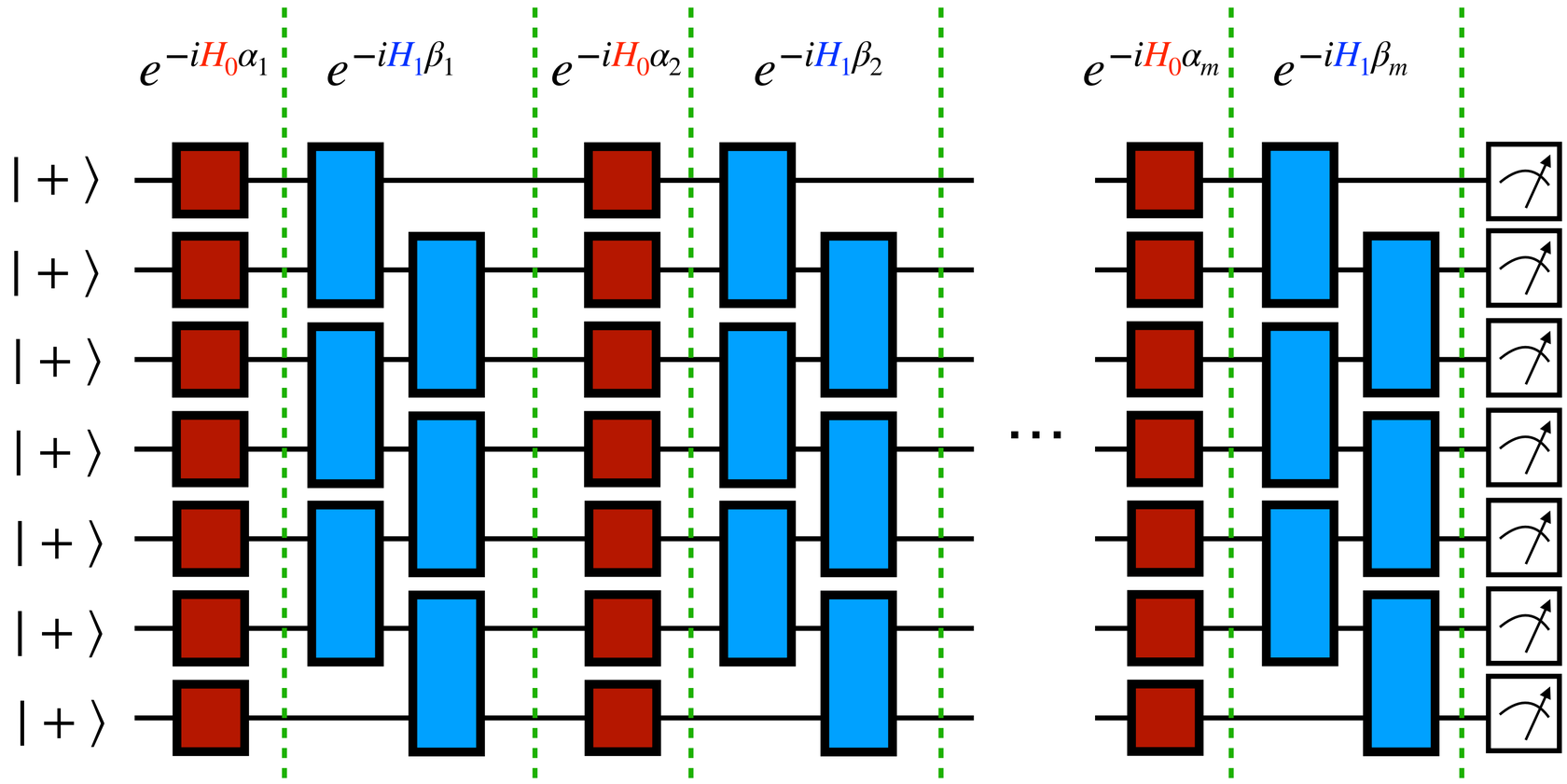
$$U \approx \prod_{j=1}^{N} e^{-i(1-p_{t_j})H_0\delta t} e^{-ip_{t_j}H_1\delta t} = \prod_{j=1} e^{-H_0 t_j^{(0)}} e^{-H_0 t_j^{(1)}}$$ ⟵ «Trotterization»



$$e^{\Delta(A+B)} = e^{\Delta A}e^{\Delta B}e^{-\frac{\Delta^2}{2}[A,B]}e^{\frac{\Delta^3}{6}\cdots} \neq e^{\Delta A}e^{\Delta B} \text{ for } [A,B] \neq 0$$

Adiabatic quantum computing

Approximation with variational ansatz



Provides solution if $\{\alpha_i\}_{i=1}^m$, $\{\beta_i\}_{i=1}^m$ are «good» and m is large enough

$$H_0 = -\sum_i \sigma_x^{(i)} \rightarrow |\psi_{\mathrm{gr}}^{(0)}\rangle = |+\rangle^{\otimes n}$$

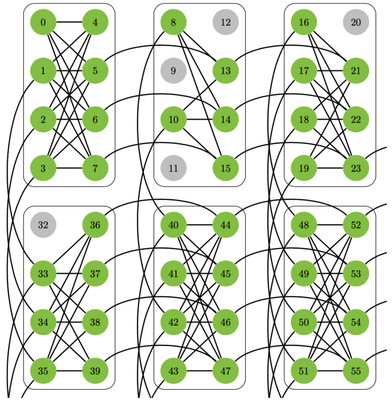
Adiabatic quantum computing

What about universality?

Example of a universal target Hamiltonian: $$H_1 = \sum_i A_i \sigma_z^{(i)} + \sum_i B_i \sigma_x^{(i)} + \sum_{i,j} C_{i,j} \sigma_x^{(i)} \sigma_x^{(j)} + \sum_{i,j} D_{i,j} \sigma_z^{(i)} \sigma_z^{(j)}$$

«Every quantum algorithm that provides a speedup in the circuit model can in principle be implemented with up to polynomial overhead in AQC»*

Quantum annealer = «noisy restricted adiabatic quantum computer»



$$H_1 = \sum_i A_i \sigma_z^{(i)} + \sum_{i,j} B_{i,j} \sigma_z^{(i)} \sigma_z^{(j)} \longleftarrow \text{Ising type}$$

- Finite temperature effects
- Not truly adiabatic regime
- A number of runs is usually performed to solve a problem

www.dwavesys.com

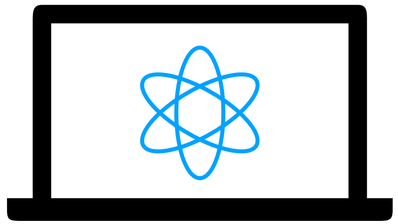
*See [Rev. Mod. Phys. 90, 015002 (2018)]

Variational quantum computing

Variational quantum computing

General idea: combining quantum and classical computing

Quantum processor



Samples

Classical computer

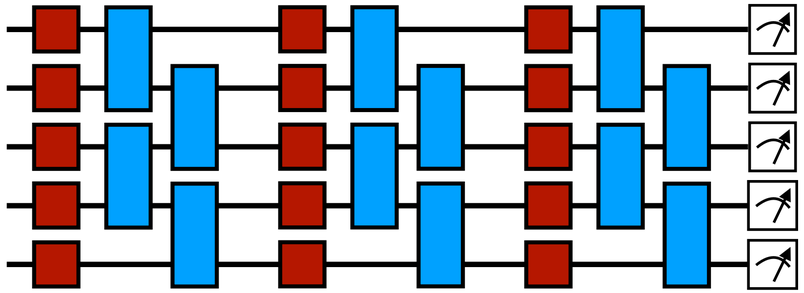
```
def cost_func(…):
    …
```

1) Compute cost function
2) Update parameters (use your favorite classical optimization algorithm — Nelder-Mead, BFGS, …)

Updated parameters $(\alpha_1, \beta_1, \ldots \alpha_m, \beta_m)$

Use as input

Parametrized circuit
(«hardware-efficient» ansatz)



$e^{-iH_0\alpha_1} e^{-iH_1\beta_1} \qquad e^{-iH_0\alpha_m} e^{-iH_1\beta_m}$

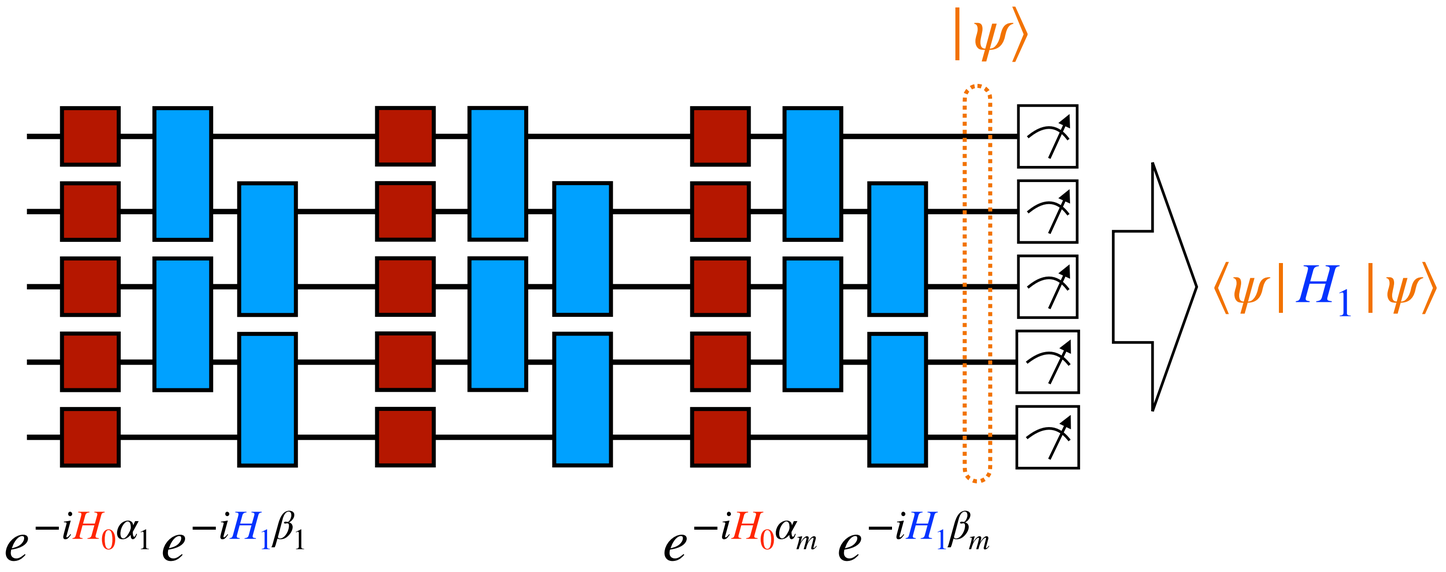
$\boxed{\#\text{ parameters } \ll \dim\mathscr{H} = 2^n}$

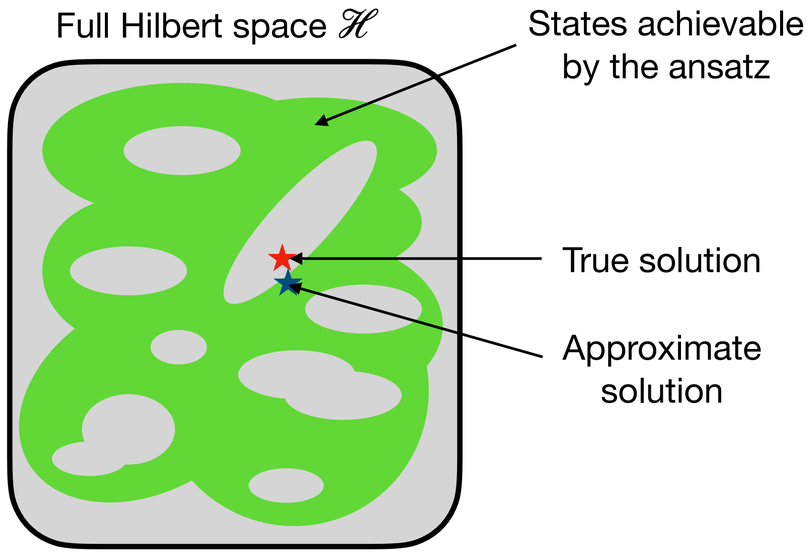
Variational quantum computing

Quantum approximate optimization algorithm (QAOA)

Problem: find $x_0 \in \{0,1\}^n$ s.t. $f(x_0) \to \min$ for given $f : \{0,1\}^n \to \mathbb{Z}$.

$$f(x) = \langle x | H_1 | x \rangle \quad H_1 = \sum_{i,j} g_{i,j} \sigma_z^{(i)} \sigma_z^{(j)} + \sum_i h_i \sigma_z^{(i)}$$



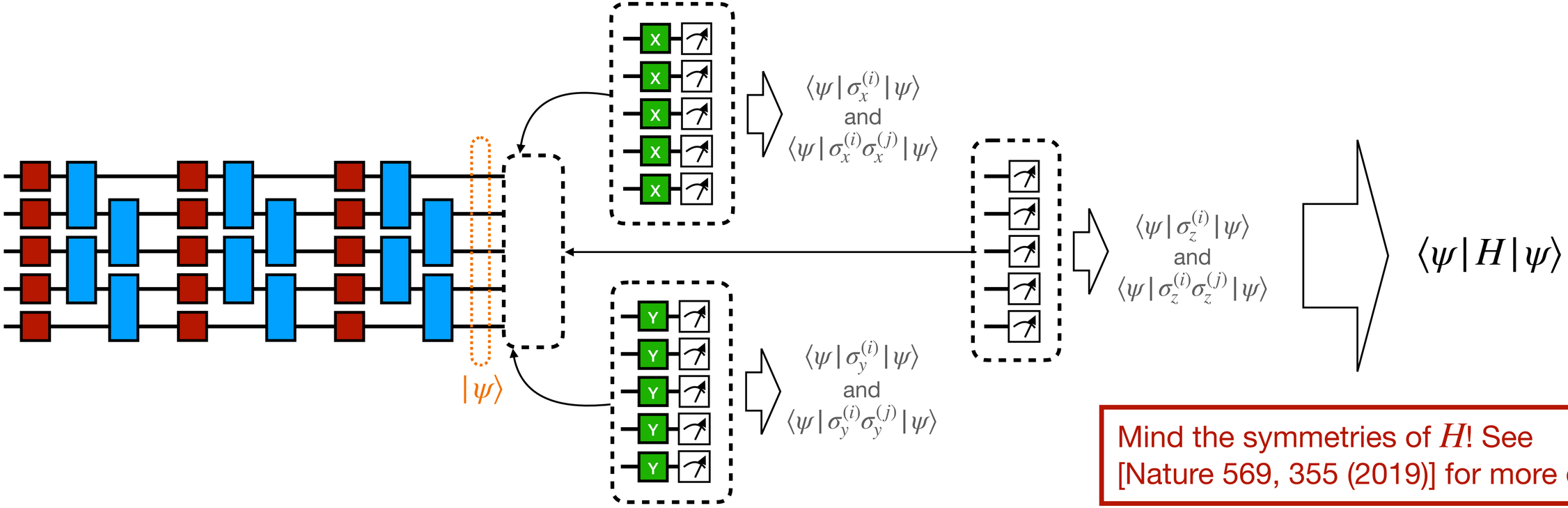


Variational quantum computing

Variational quantum state eigensolver (VQE)

Problem: find $|\psi_0\rangle$ s.t. $\langle\psi_0|H|\psi_0\rangle \to \min$ for given H.

$$H = \sum_{i,j,\alpha} g_{i,j}\sigma_\alpha^{(i)}\sigma_\alpha^{(i)} + \sum_{i,\beta} h_{i,\beta}\sigma_\beta^{(i)} \Rightarrow \langle\psi|H|\psi\rangle = \sum_{i,j,\alpha} g_{i,j}\langle\psi|\sigma_\alpha^{(i)}\sigma_\alpha^{(j)}|\psi\rangle + \sum_{i,\beta} h_{i,\beta}\langle\psi|\sigma_\beta^{(i)}|\psi\rangle$$



Mind the symmetries of H! See [Nature 569, 355 (2019)] for more details.

Variational quantum computing

What about universality?

PHYSICAL REVIEW LETTERS **124,** 090504 (2020)

Reachability Deficits in Quantum Approximate Optimization

V. Akshay,* H. Philathong, M. E. S. Morales, and J. D. Biamonte
Deep Quantum Laboratory, Skolkovo Institute of Science and Technology, 3 Nobel Street, Moscow 121205, Russia

(Received 25 July 2019; revised manuscript received 24 October 2019; accepted 3 February 2020; published 5 March 2020)

The quantum approximate optimization algorithm (QAOA) has rapidly become a cornerstone of contemporary quantum algorithm development. Despite a growing range of applications, only a few results have been developed towards understanding the algorithm's ultimate limitations. Here we report that QAOA exhibits a strong dependence on a problem instances constraint to variable ratio—this problem density places a limiting restriction on the algorithms capacity to minimize a corresponding objective function (and hence solve optimization problem instances). Such *reachability deficits* persist even in the absence of barren plateaus and are outside of the recently reported level-1 QAOA limitations. These findings are among the first to determine strong limitations on variational quantum approximate optimization.

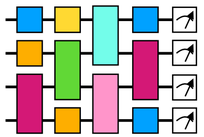
DOI: 10.1103/PhysRevLett.124.090504

Conclusion

Conclusion

Take-home messages

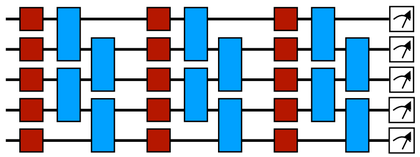
Universal quantum computing — very cool but still very far from reaching (no sufficient error-correction, too few physical qubits). Nevertheless, first «quantum advantage» demonstrations with NISQ devices reported.



Adiabatic quantum computing — also proved to be universal but still no full-fledged implementation demonstrated. Nevertheless, first quantum annealers are available.

$$H_t = (1 - p_t)H_0 + p_t H_1$$

Variational quantum computing — can be launched on NISQ devices but deeper study of limitation is required. Nevertheless, interesting application to optimization and quantum simulation demonstrated.



A lot of work for you to be done! :-)

«Time to pretend that this was interesting» (c)

Slides: github.com/EvgeniyKiktenko/RQC-MIPT-q-inf-proc-course/Slides
Anonymous feedback form: https://forms.gle/gju1tiz6MTKBR66EA
Questions: e.kiktenko@rqc.ru

(c) Yaroslav Zolotarev at inner RQC workshop on «Quantum autoencoders for quantum information compression»