Virtual Qubits from Classical Computation

John Smolin

IBM TJ Watson Research Center

Sergey Bravyi

Graeme Smith

Virtual Qubits from Classical Computation

John Smolin

IBM TJ Watson Research Center

Graeme Smith

Sergey Bravyi

Classical Computers

Classical computers are advanced

Quantum Computers

Quantum computers are hard to build

Quantum computers are hard to simulate

- Naive simulation of a quantum circuit of L gates:
- “Just” keep track of a 2^n length vector
- Elementary gates are one-qubit or CNOTs
- Each gate is sparse, so do 2×2^n multiplications per gate
- L gates, so $L \times 2 \times 2^n$ total multiplications
- Note: $10^9 = 2^{30}$ and $10^{15} = 2^{50}$

Hybrid Classical-Quantum computation

+

50+50 = 100?

50+50 = 50?

Adding Virtual Qubits

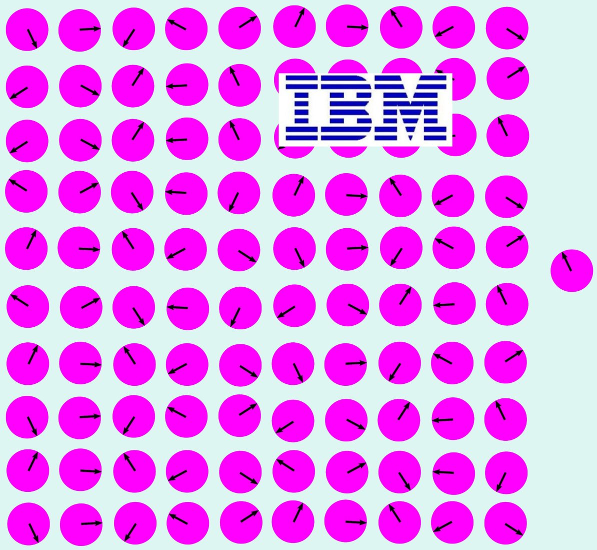
- Given a small quantum computer and a large classical computer, can we emulate having a slightly bigger quantum computer?

Adding Virtual Qubits

- Given a small quantum computer and a large classical computer, can we emulate having a slightly bigger quantum computer?
- Yes, sometimes we can!

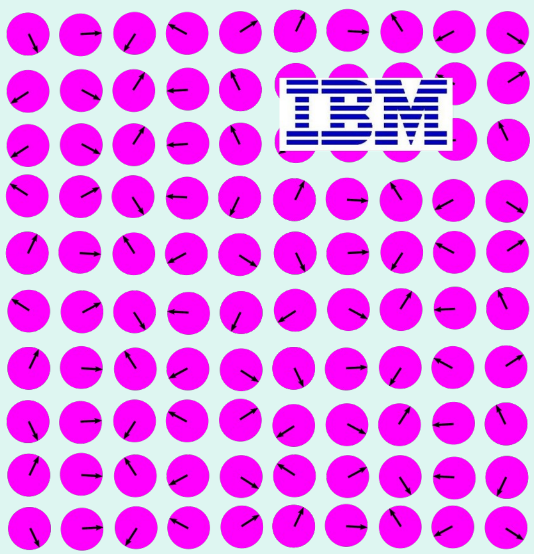
Suppose I want to run an algorithm that requires 101 qubits

If I have a 101-qubit quantum computer, fantastic!



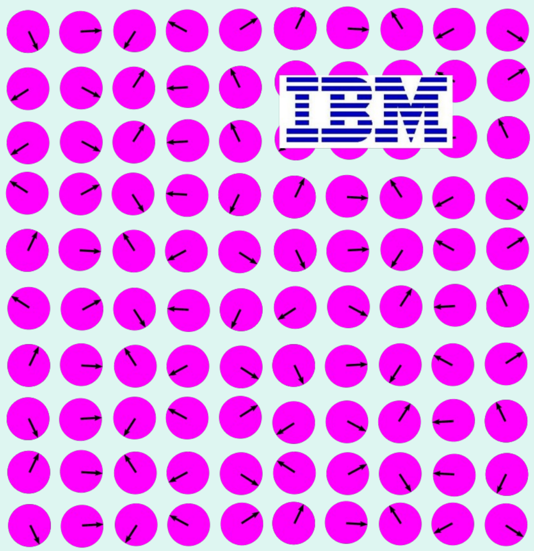
Suppose I want to run an algorithm that requires 101 qubits

If I have a 100-qubit quantum computer, disaster?!

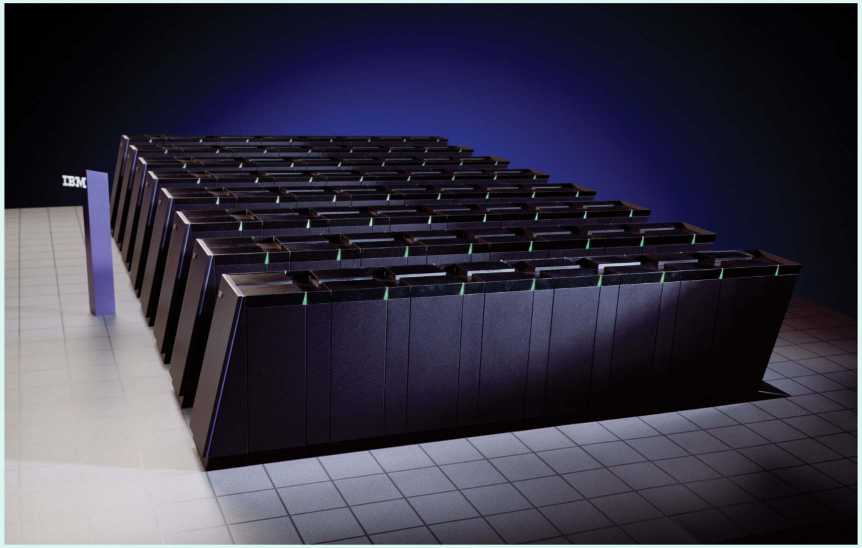


Suppose I want to run an algorithm that requires 101 qubits

If I have a 100-qubit quantum computer, disaster?!

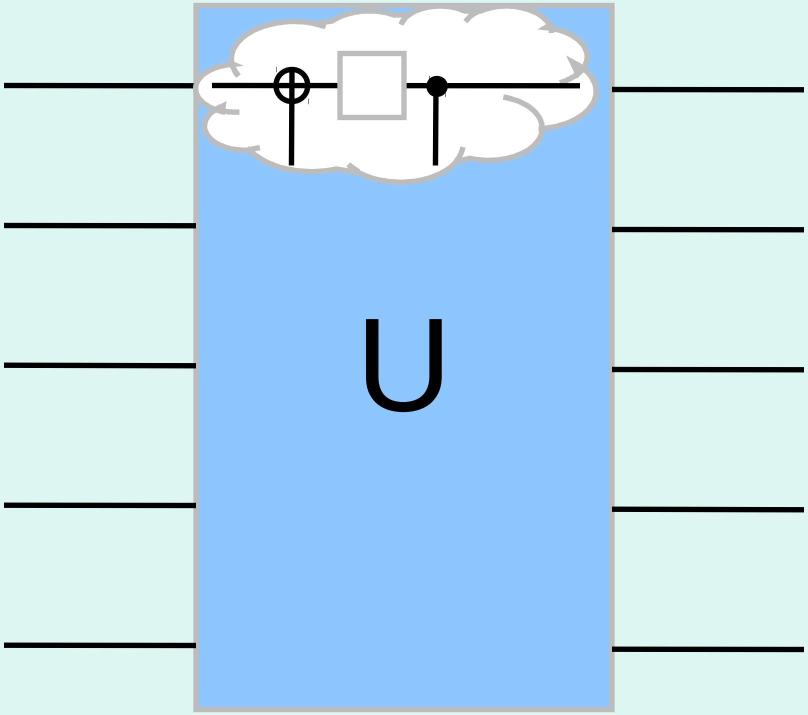


+



Can I use classical computation to save the day?

d-sparse circuit



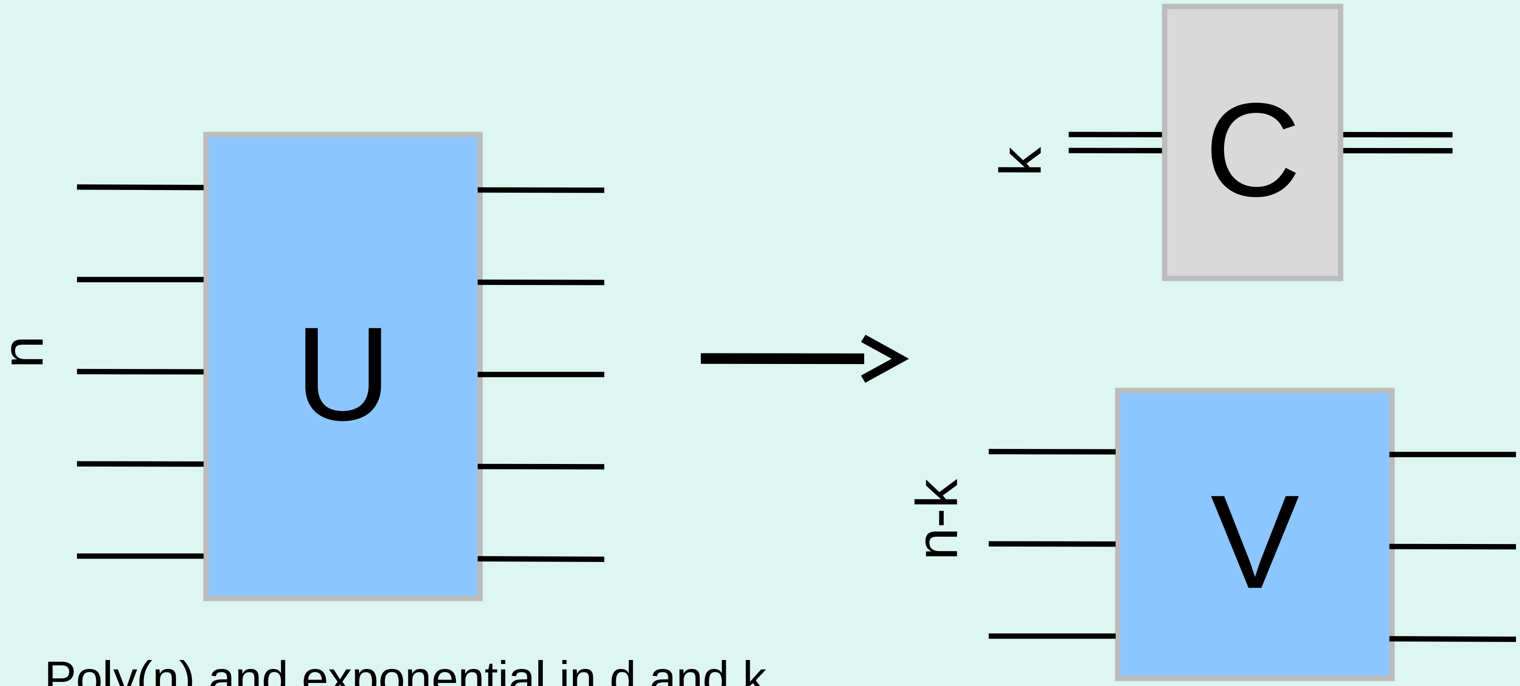
A circuit is d-sparse if each qubit participates in at most d 2-qubit gates

d-sparse circuits are nontrivial

- Includes depth-d circuits and more
- Classically hard unless BQP is contained in AM. Depth d=3 easy, d=4 hard [Terhal-Divincenzo 2002].
- Instantaneous Quantum Computation [Bremner-Jozsa-Sheppard 2010] **commuting Hamiltonians hard to sample**
- There are log-depth circuits for Shor’s algorithm (BUT!) [Cleve-Watrous 2000]

Main Result

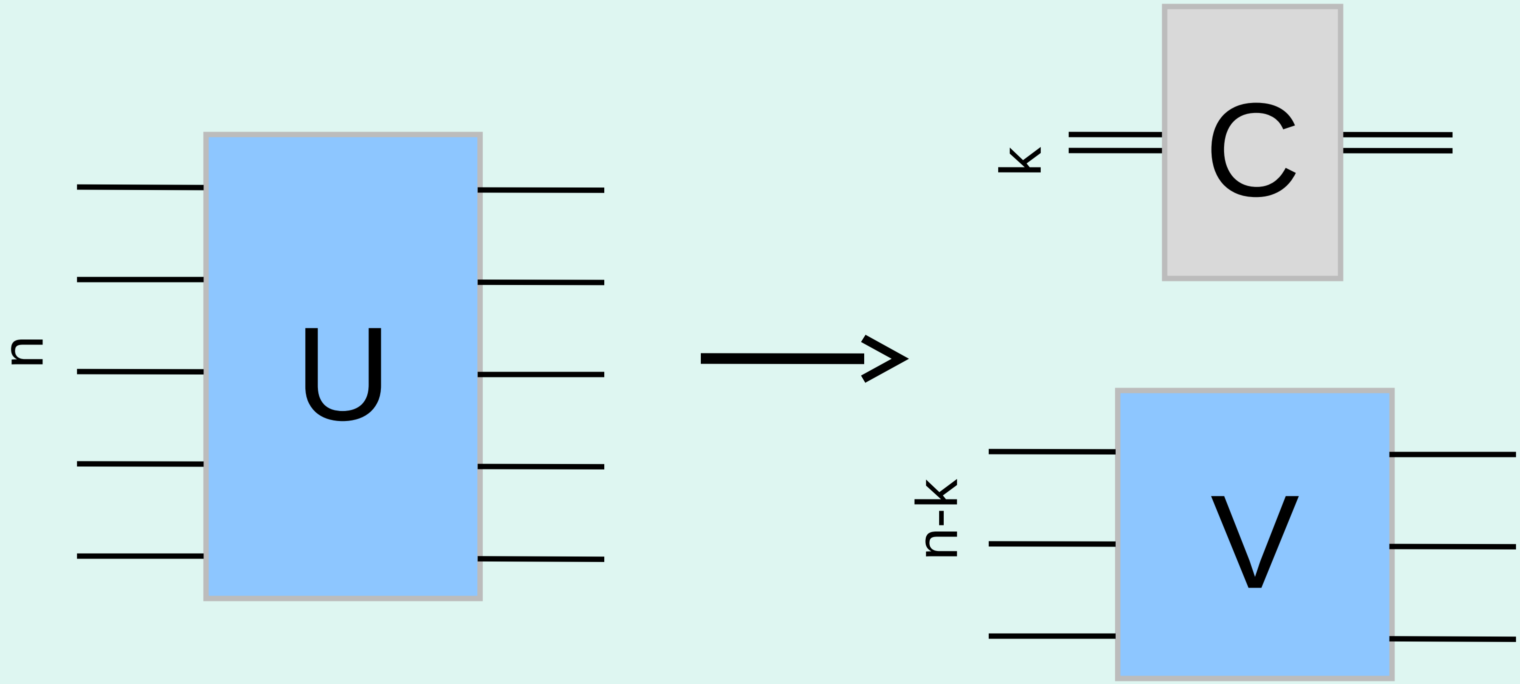
Theorem 1. *Suppose $n \geq kd+1$. Then any d-sparse quantum computation on $n+k$ qubits can be simulated by a $(d+3)$-sparse quantum computation on n qubits repeated $2^{O(kd)}$ times and a classical processing which takes time $2^{O(kd)}$poly(n).*



Poly(n) and exponential in d and k

Main Result

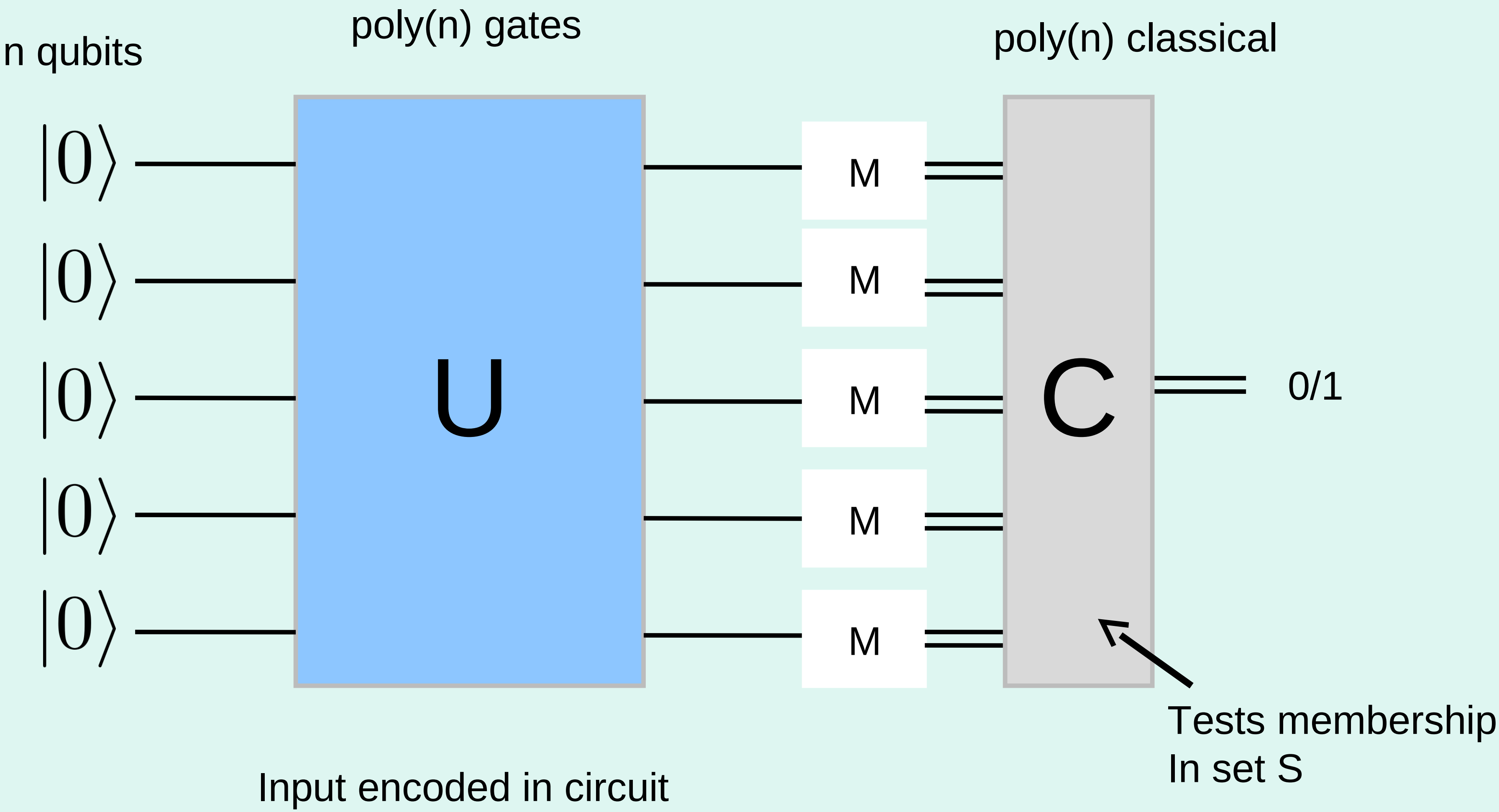
Theorem 1. *Suppose $n \geq kd+1$. Then any d-sparse quantum computation on $n+k$ qubits can be simulated by a $(d+3)$-sparse quantum computation on n qubits repeated $2^{O(kd)}$ times and a classical processing which takes time $2^{O(kd)} poly(n)$.*



Poly(n) and exponential in d and k

If d and k are O(1) then runs in poly(n) time
Direct classical simulation takes O(2^n)

Standard quantum computation with classical postprocessing



Main Result

Theorem 1. *Suppose $n \geq kd+1$. Then any d-sparse quantum computation on $n+k$ qubits can be simulated by a $(d+3)$-sparse quantum computation on n qubits repeated $2^{O(kd)}$ times and a classical processing which takes time $2^{O(kd)} poly(n)$.*

First Idea: Fix each of the qubits to be removed, and evaluate each branch

Problem 1) qubits change each time you touch them

Problem 2) Branches have to be able to interfere

Main Result

Theorem 1. *Suppose $n \geq kd + 1$. Then any d-sparse quantum computation on $n + k$ qubits can be simulated by a $(d + 3)$-sparse quantum computation on n qubits repeated $2^{O(kd)}$ times and a classical processing which takes time $2^{O(kd)} poly(n)$.*

Main Result

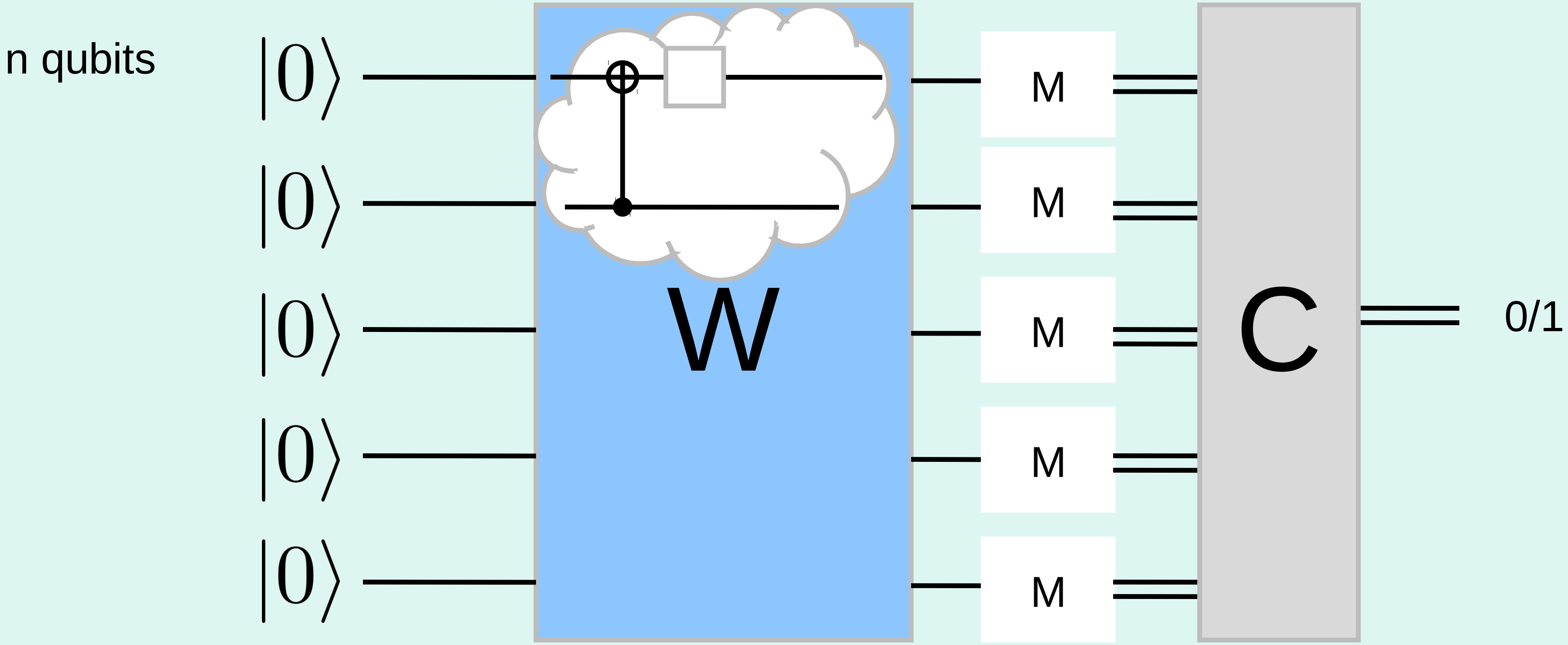
Theorem 1. *Suppose $n \geq kd+1$. Then any d-sparse quantum computation on $n+k$ qubits can be simulated by a $(d+3)$-sparse quantum computation on n qubits repeated $2^{O(kd)}$ times and a classical processing which takes time $2^{O(kd)} poly(n)$.*

Better: Express acceptance probability as convenient contraction of tensors such that

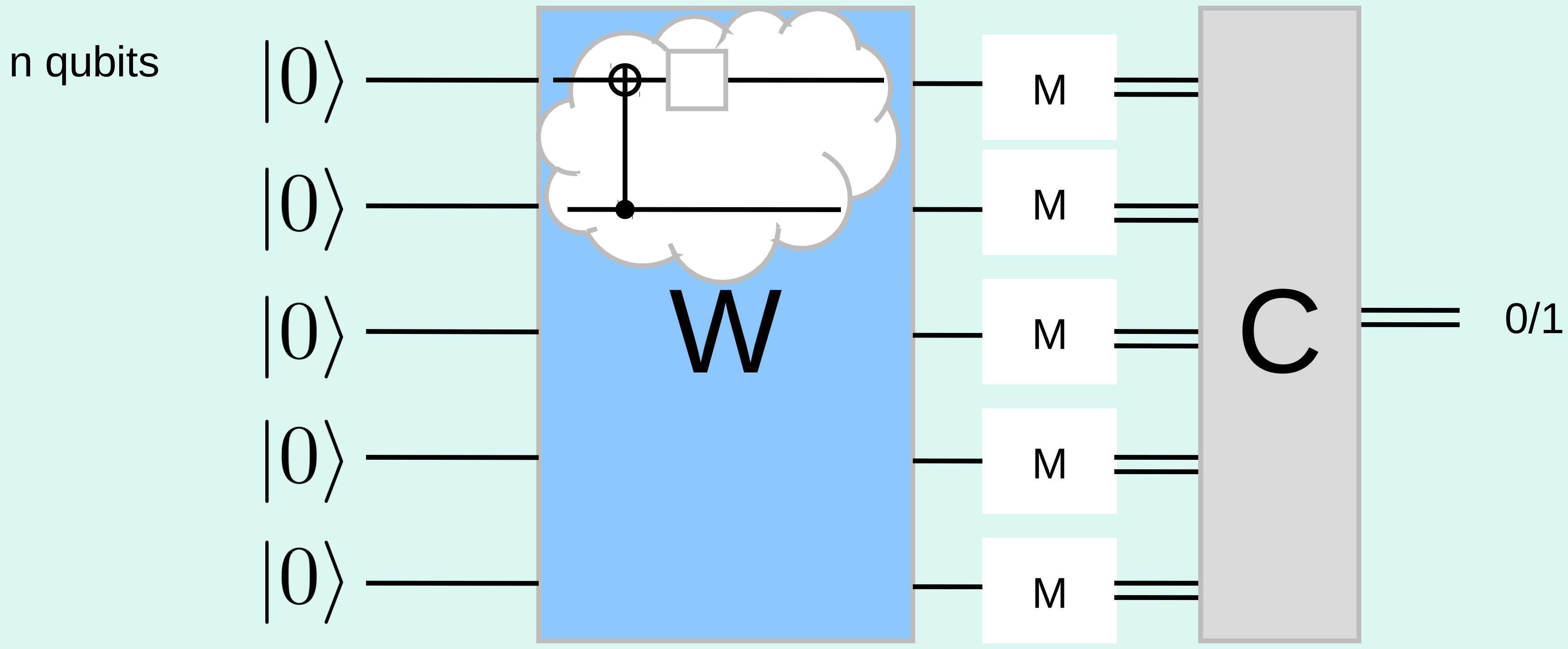
1) Entries of tensors can be evaluated on n-k qubits and

2) Contraction can be done efficiently

Removing a qubit: simplest case

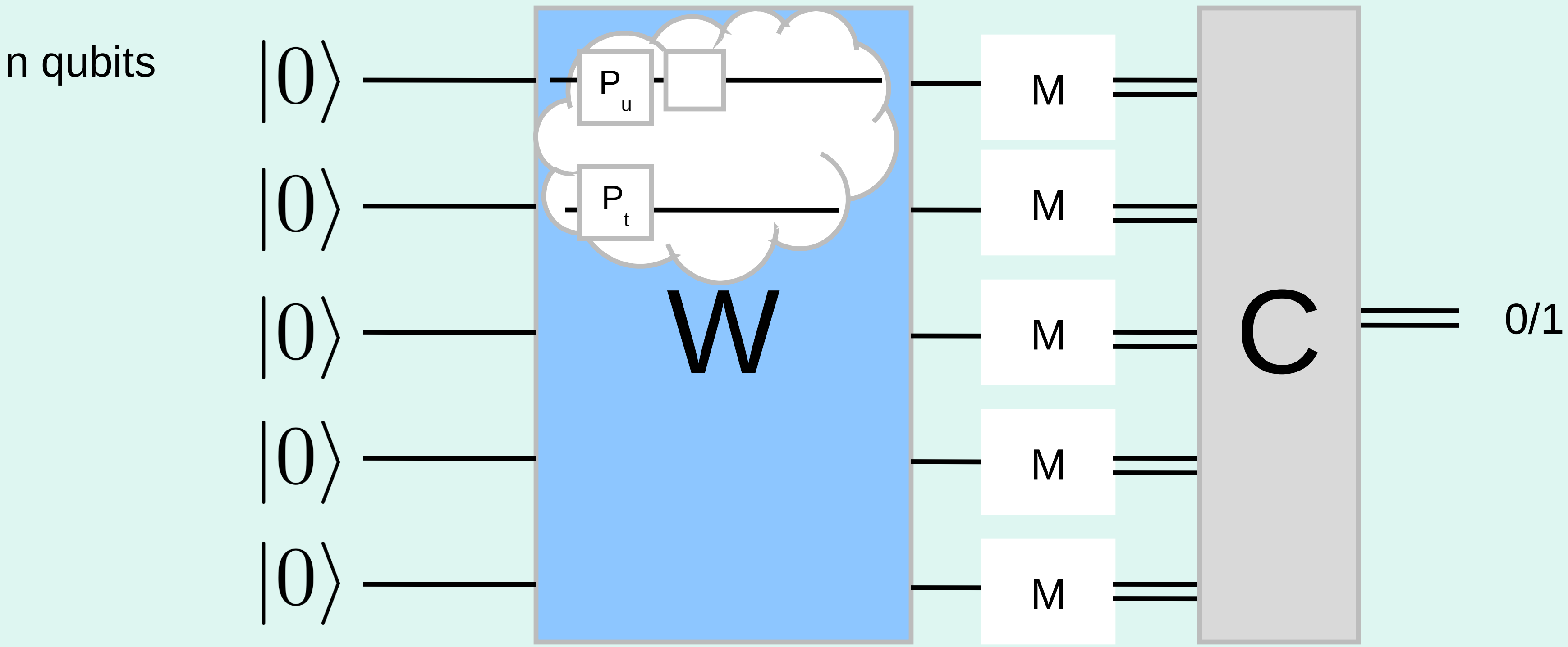


Removing a qubit: simplest case



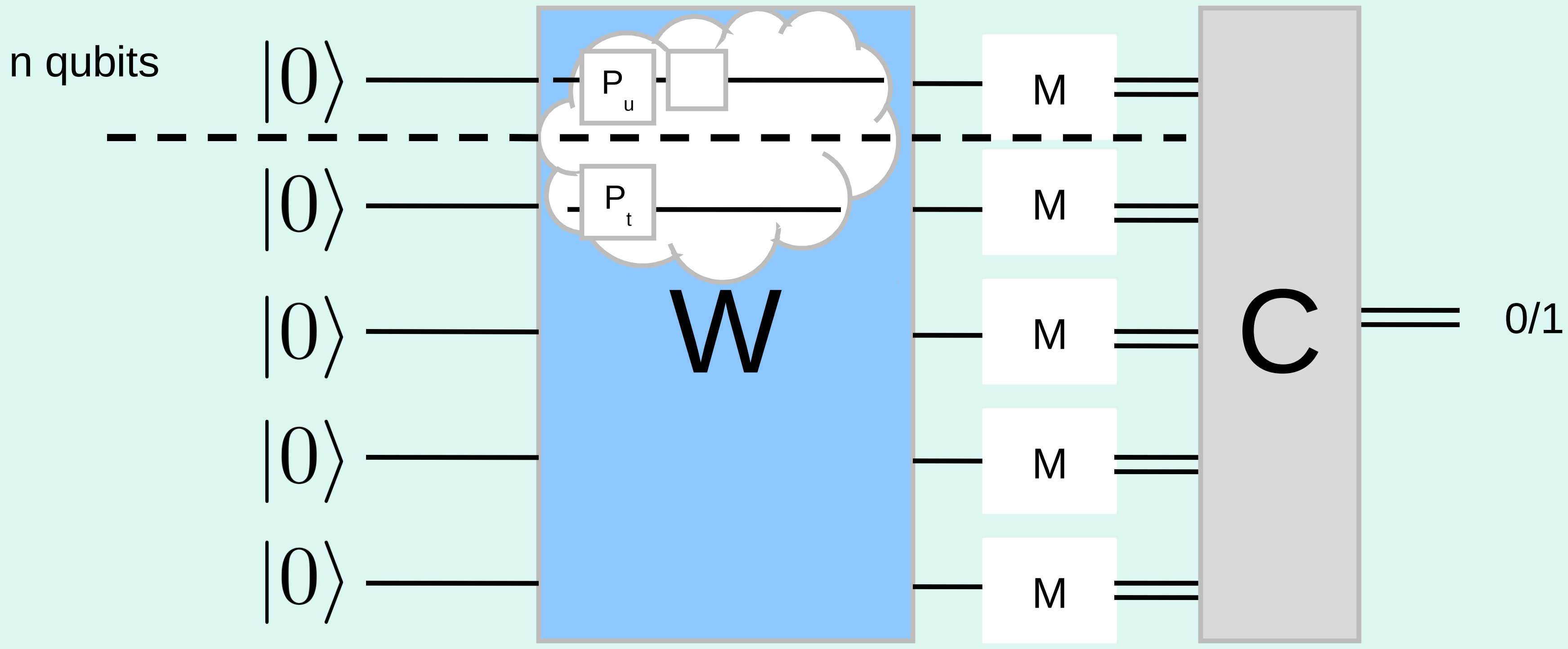
We can expand the CNOT in the Pauli basis CNOT = $\sum_{u,t} c_{u,t} P_u P_t$

Removing a qubit: simplest case

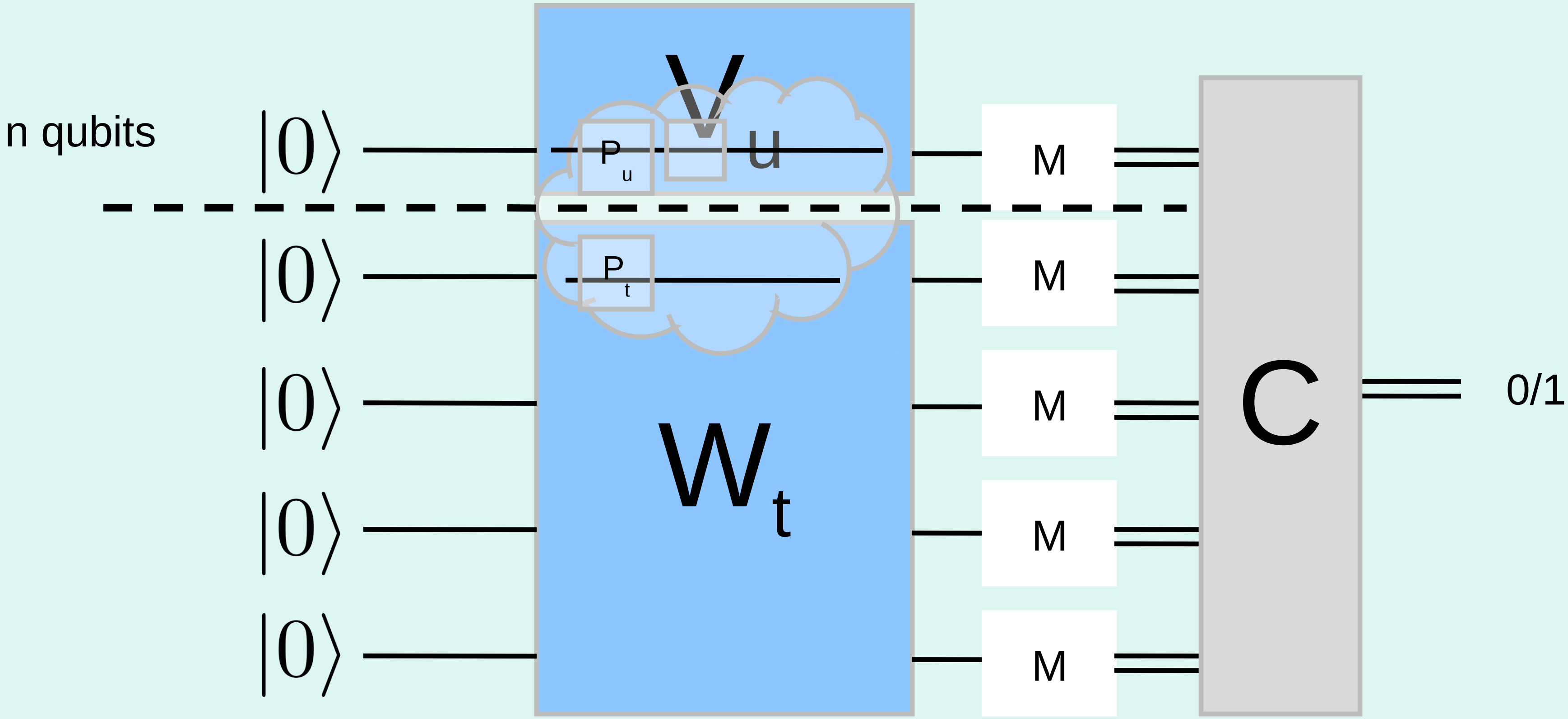


We can expand the CNOT in the Pauli basis CNOT = $\sum_{u,t} c_{u,t} P_u P_t$

Removing a qubit: simplest case

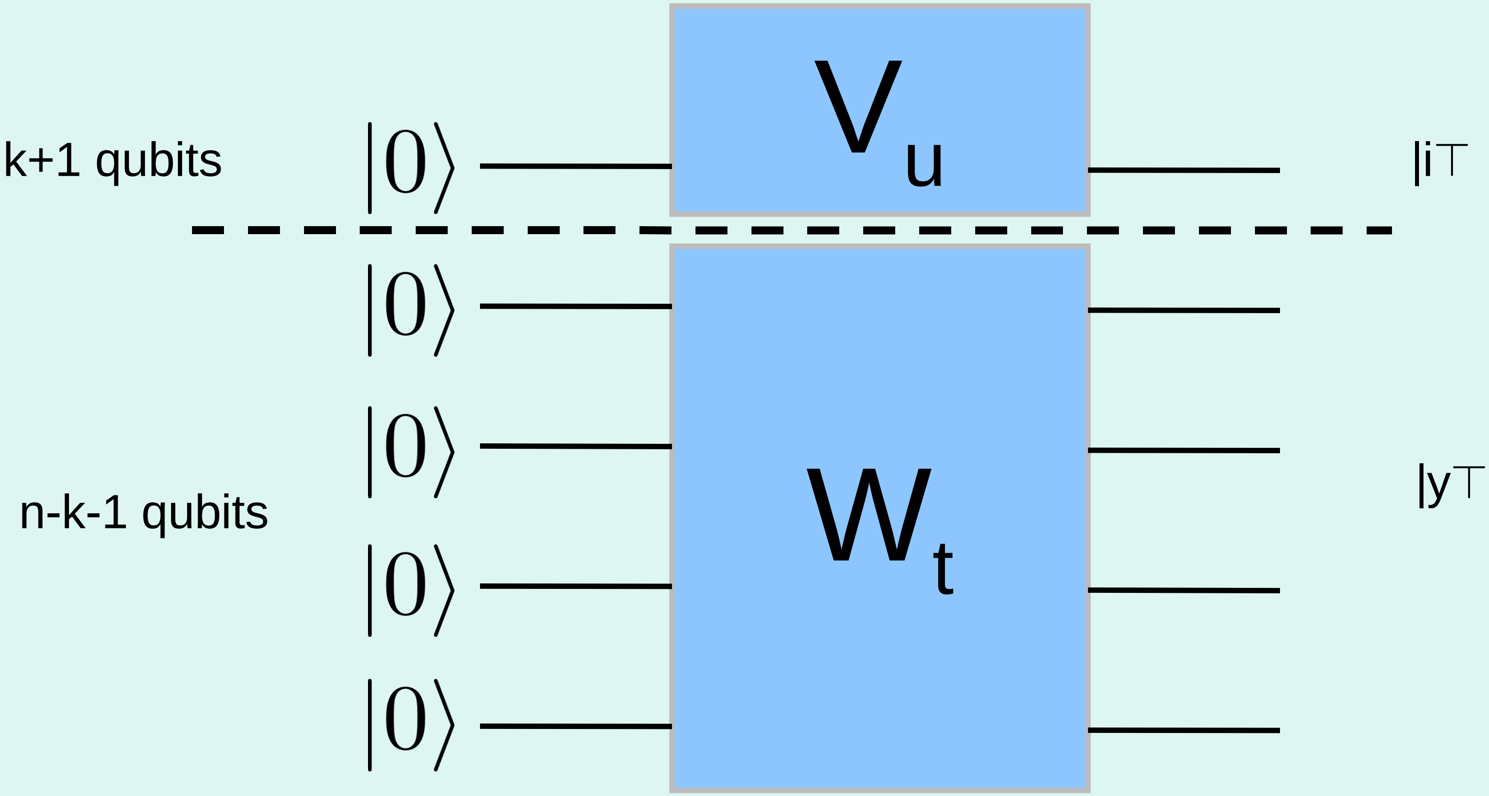


We can expand the CNOT in the Pauli basis CNOT = $\sum_{u,t} c_{u,t} P_u P_t$



We can expand the CNOT in the Pauli basis CNOT = $\sum_{u,t} c_{u,t} P_u P_t$

Acceptance prob in terms of these:



Let’s evaluate them!

Lemma: contraction of acceptance probability

$$\Pi_{\mathbf{acc}}(W) = \sum_{\mathbf{x} \in S} |\langle x|W|0^n\rangle|^2.$$

This can be further decomposed as

$$\Pi_{\mathbf{acc}}(W) = \sum_{\alpha,\beta,i} A^i_{\alpha\beta} B^i_{\alpha\beta},$$

where $\alpha, \beta = 1 \ldots 16^{(k+1)d}$, $i \in \{0,1\}^{k+1}$,

$$A^i_{\alpha\beta} = C_\alpha \overline{C}_\beta \langle i|V_\alpha|0^{k+1}\rangle \overline{\langle i|V_\beta|0^{k+1}\rangle}$$
$$B^i_{\alpha\beta} = \sum_{y \in T_i} \langle y|W_\alpha|0^{n-k-1}\rangle \overline{\langle y|W_\beta|0^{n-k-1}\rangle}$$

with V_α *and* W_α *are* d*-sparse circuits on* $k+1$ *and* $n-k-1$ *qubits, respectively, and*

$$T_i = \left\{ y \in \{0,1\}^{n-k-1} | (i,y) \in S \right\}.$$

Lemma: contraction of acceptance probability

Proof:

$$
\begin{aligned}
\Pi_{\mathbf{acc}}(W) &= \sum_{x\in S} |\langle x|W|0^n\rangle|^2 \\
&= \sum_{x\in S} \langle x|W|0^n\rangle \overline{\langle x|W|0^n\rangle} \\
&= \sum_{i\in\{0,1\}^{k+1}} \sum_{y\in T_i} \langle i|\langle y|W|0^n\rangle \overline{\langle i|\langle y|W|0^n\rangle} \\
&= \sum_{i\in\{0,1\}^{k+1}} \sum_{y\in T_i} \sum_{\alpha,\beta=1}^{16^{(k+1)d}} C_\alpha \bar{C}_\beta \langle i|\langle y|V_\alpha \otimes W_\alpha|0^n\rangle \overline{\langle i|\langle y|V_\beta \otimes W_\beta|0^n\rangle} \\
&= \sum_{i\in\{0,1\}^{k+1}} \sum_{\alpha,\beta=1}^{16^{(k+1)d}} C_\alpha \bar{C}_\beta \langle i|V_\alpha|0^{k+1}\rangle \overline{\langle i|V_\beta|0^{k+1}\rangle} \left[\sum_{y\in T_i} \langle y|W_\alpha|0^{n-k-1}\rangle \overline{\langle y|W_\beta|0^{n-k-1}\rangle}\right] \\
&= \sum_{\alpha,\beta,i} A^i_{\alpha\beta} B^i_{\alpha\beta},
\end{aligned}
$$

Lemma III.3. *Let W_α and W_β be poly(n)-sized unitaries on n qubits, and let $T \subset \{0,1\}^n$ be a set for which membership can be tested efficiently on a classical computer. Then,*

$$B_{\alpha\beta} = \sum_{y\in T} \langle y|W_\alpha|0^n\rangle\langle 0^n|W_\beta^\dagger|y\rangle \tag{19}$$

can be estimated to within ϵ by $O\left(\frac{1}{\epsilon^2}\log(1/\epsilon)\right)$ quantum computations that take poly(n) time on a quantum computer with $n+1$ qubits.

Proof. First note that, letting $|\varphi_\alpha\rangle = W_\alpha|0^n\rangle$ and $|\varphi_\beta\rangle = W_\beta|0^n\rangle$ we can efficiently prepare

$$\frac{1}{\sqrt{2}}\left(|0\rangle|\varphi_\alpha\rangle + |1\rangle|\varphi_\beta\rangle\right)$$

and, by applying a hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ to the first register, we get the state

$$\frac{1}{2}\left(|0\rangle\left[|\varphi_\alpha\rangle + |\varphi_\beta\rangle\right] + |1\rangle\left[|\varphi_\alpha\rangle - |\varphi_\beta\rangle\right]\right).$$

Measuring in the computational basis, and labeling the first bit b and the next n bits y, we find that

$$\Pr(b=0, y\in T) = \frac{1}{4}\left(\sum_{y\in T}\left(|\langle y|\varphi_\alpha\rangle|^2 + |\langle y|\varphi_\beta\rangle|^2\right) + 2\mathbf{Re}\left[\sum_{y\in T}\langle y|\varphi_\alpha\rangle\langle\varphi_\beta|y\rangle\right]\right).$$

Sample $\frac{1}{\epsilon^2}\log(1/\epsilon)$ times to get ϵ approximation

Also measure $\sum_{y\in T}\left(|\langle y|\varphi_\alpha\rangle|^2 + |\langle y|\varphi_\beta\rangle|^2\right)$ to get $\mathbf{Re}\left[\sum_{y\in T}\langle y|\varphi_\alpha\rangle\langle\varphi_\beta|y\rangle\right]$

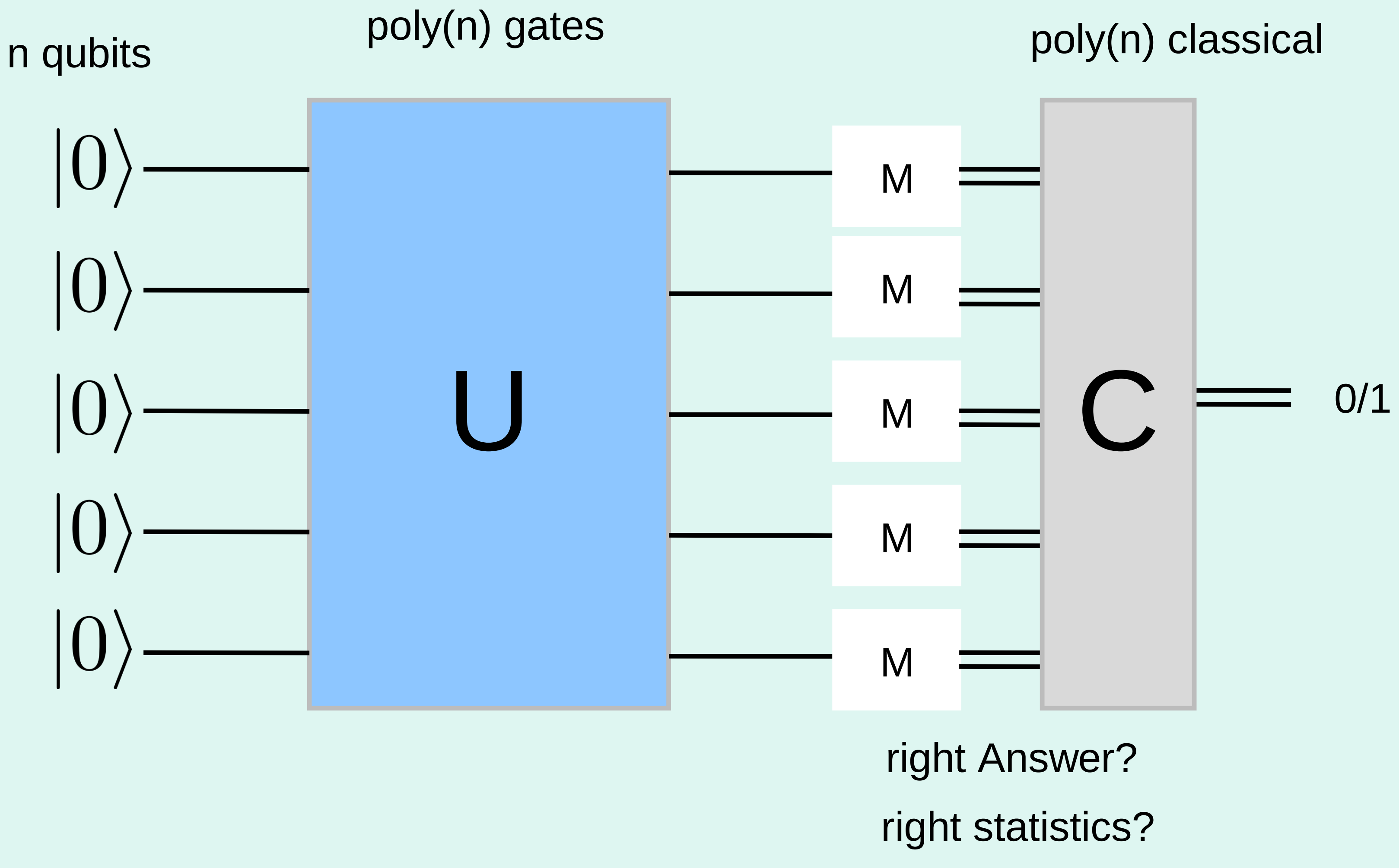
Get imaginary part by starting with $\frac{1}{\sqrt{2}}\left(|0\rangle|\varphi_\alpha\rangle + i|1\rangle|\varphi_\beta\rangle\right)$

Trivial lower "bound"

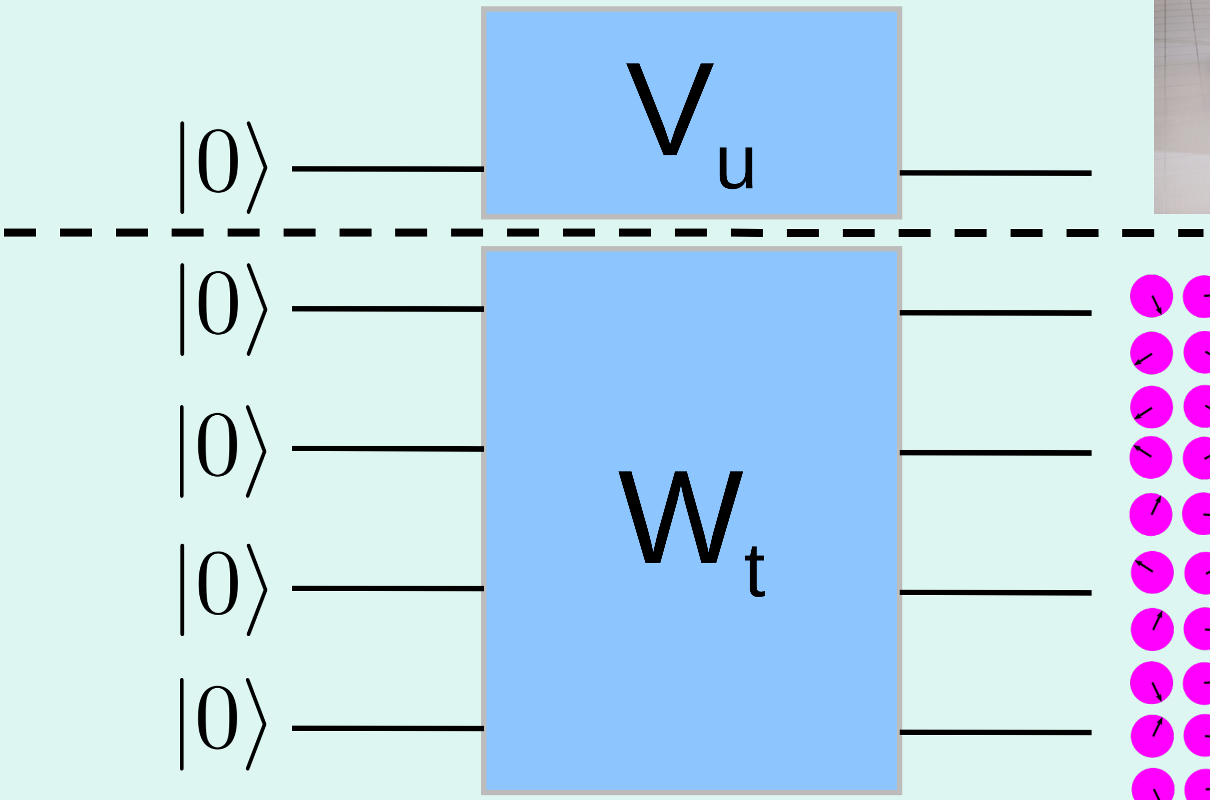
- No unconditional results---it's unknown whether quantum computer can be efficiently simulated classically (though such a simulation is "implausible")
- Assume L-gate QC takes 2 x L x 2^n classical computation time
- Suppose can emulate L-gate n qubit computation by performing a larger number of L-gate quantum computations b on n-1 qubits plus poly-time classical comp.
- By iterating, we end up with a c^{n-1} single-qubit computations with L gates, which can be done classically in time L b^{n-1} gates, so b>2.
- Our algorithm gives somewhat larger $b = O(8^{k+1}\ 16^{3(1+1)d} / 2^2)$

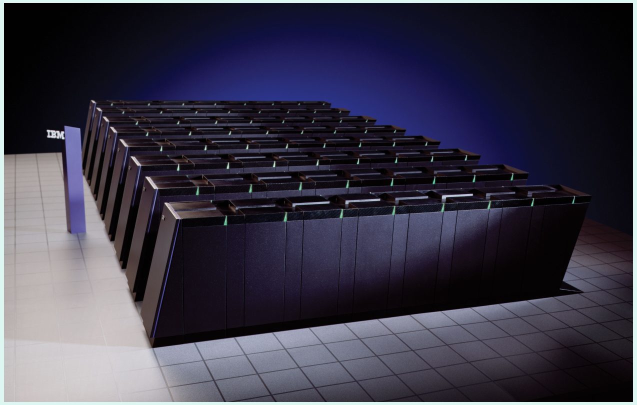
Characterizing Noise

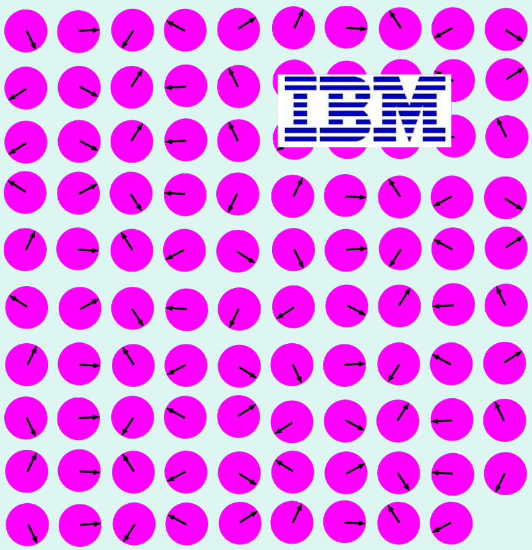
If we run a quantum computation with some noise, how do we check it's working right?



Characterizing Noise







If different, removed qubits aren't working right
If same, noise on removed qubits benign

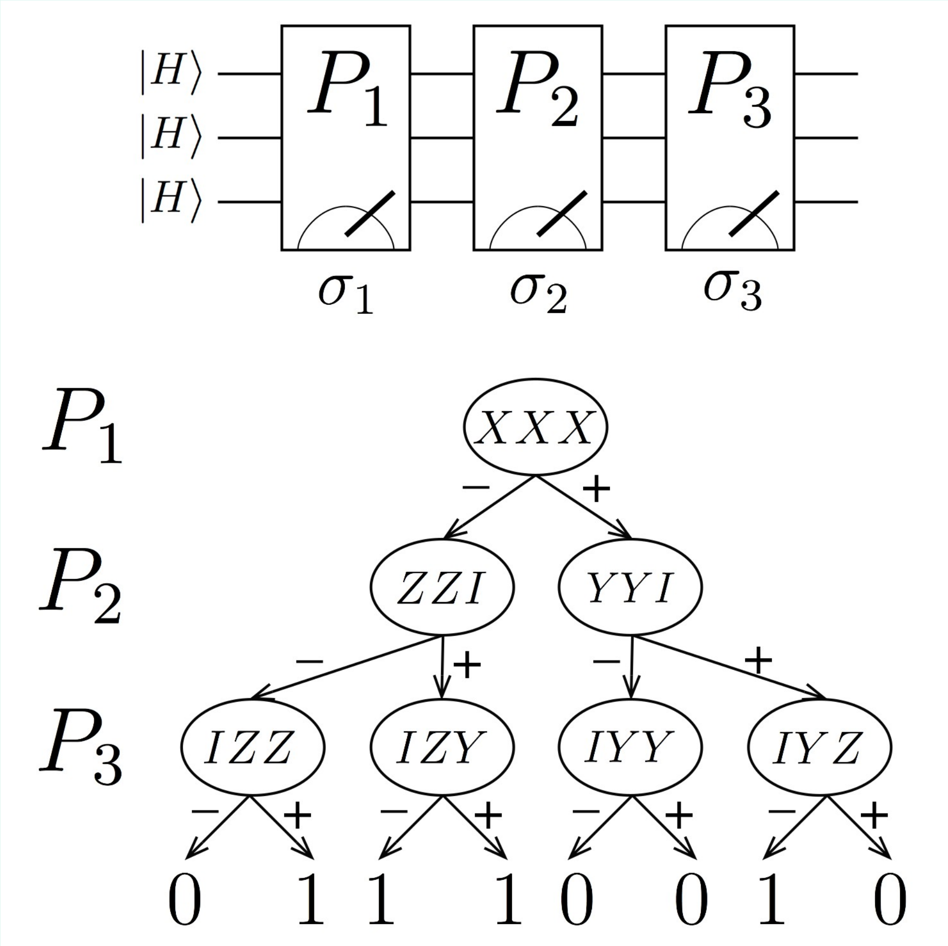
same Answer?
same statistics?

Summary

- Want to emulate an n qubit computation using n-k qubits together with a classical machine
- Can do so by expressing acceptance probability as a contraction of tensors whose entries can be estimated on n qubits
- Cost scales exponentially with k and d for a d-sparse computation.
- k scaling is expected, but d scaling may not be optimal
- Could be used for characterizing noise in quantum circuits and emulating modular quantum computations
- Most annoying open question: can you remove 1 qubit from an arbitrary poly(n) circuit for less than 2^n effort? More generally, lower bounds?

Pauli-Based Computation (PBC)

$$|H\rangle = \cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle$$ The Magic State



Theorem 2. *Any quantum computation in the circuit-based model with n qubits and poly(n) gates drawn from the Clifford+T set can be simulated by a PBC on m qubits, where m is the number of T gates, and poly(n) classical processing.*

Theorem 3. *A PBC on $n + k$ qubits can be simulated by a PBC on n qubits repeated $2^{O(k)}$ times and a classical processing which takes time $2^{O(k)}$poly(n).*

Theorem 4. *Any PBC on n qubits can be simulated classically in time $2^{\alpha n}$poly(n), where $\alpha \approx 0.94$.*

Theorem 2. *Any quantum computation in the circuit-based model with n qubits and poly(n) gates drawn from the Clifford+T set can be simulated by a PBC on m qubits, where m is the number of T gates, and poly(n) classical processing.*

Theorem 3. *A PBC on $n + k$ qubits can be simulated by a PBC on n qubits repeated $2^{O(k)}$ times and a classical processing which takes time $2^{O(k)}$poly(n).*

Theorem 4. *Any PBC on n qubits can be simulated classically in time $2^{\alpha n}$poly(n), where $\alpha \approx 0.94$.*

Note that $2^{.94} < 2$

Theorem 4. *Any PBC on n qubits can be simulated classically in time $2^{\alpha n}poly(n)$, where $\alpha \approx 0.94$.*

Proof comes from decomposing magic states as superpositions of stabilizer states

Stabilizer rank of $|\psi\rangle$ is smallest value of χ

such that $|\psi\rangle = \sum_{i=1}^{\chi} c_i |\phi_i\rangle$

Stabilizer states

Conjecture: Magic states have the lowest stabilizer rank of any non-stabilizer states

Theorem 4. *Any PBC on n qubits can be simulated classically in time $2^{\alpha n}poly(n)$, where $\alpha \approx$* 0.75

Proof comes from decomposing magic states as superpositions of stabilizer states

$$\text{Stabilizer rank of } |\psi\rangle \text{ is smallest value of } \chi$$

such that $$|\psi\rangle = \sum_{i=1}^{\chi} c_i |\phi_i\rangle$$

Stabilizer states

Conjecture: Magic states have the lowest stabilizer rank of any non-stabilizer states

See Bravyi and Gosset's talk at QIP 2017