Improved classical simulation of quantum circuits

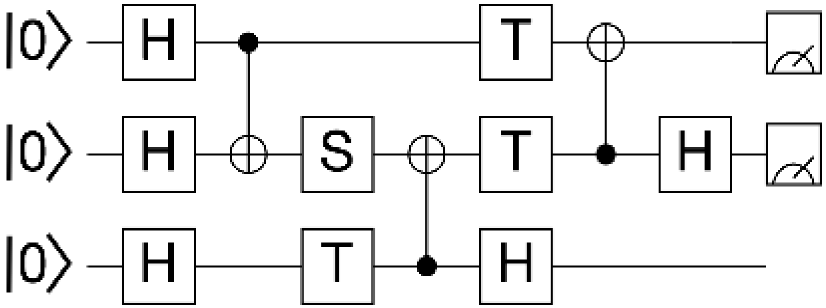
Sergey Bravyi and David Gosset

IBM

PRL 116, 250501 (2016)

QIP 2017
Seattle

Clifford+T circuits

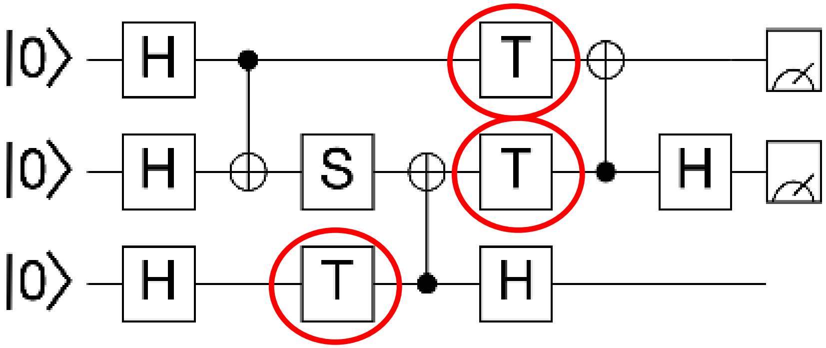


$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

$$\underbrace{S = T^2 \qquad H \qquad CNOT}_{\text{Clifford gates}}$$

- ✓ Computational universality
- ✓ Efficient gate synthesis
- ✓ Fault-tolerant realization

Clifford+T circuits



$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

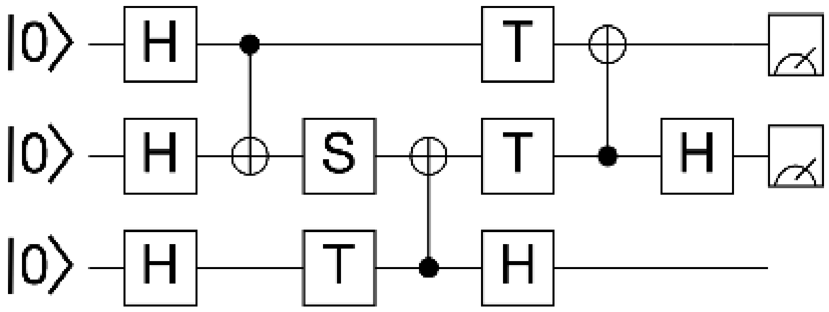
$$\underbrace{S = T^2 \quad H \quad CNOT}_{\text{Clifford gates}}$$

T-count: number of T-gates

This talk: classical simulation algorithms for Clifford+T circuits with a small T-count

Motivation

- Fault-tolerant T gates are expensive
- Verification of small quantum computers
- Understand conditions for “quantum supremacy”



Qubits	n
Clifford gates	c
T-count	t

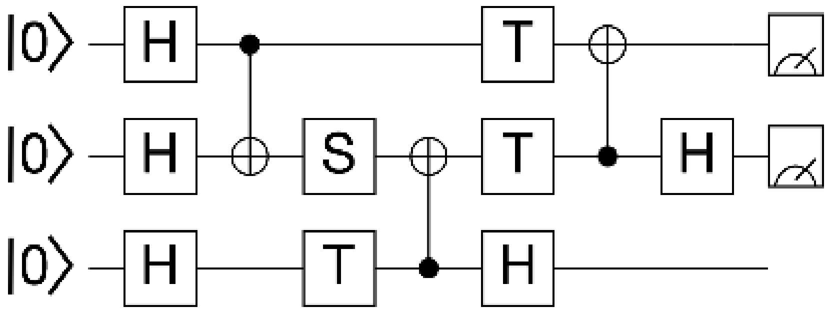
Universal simulators: store a complete description of a quantum state as a complex vector of size 2^n

Runtime: $2^n(c+t)$

Limited to a small number of qubits $n \leq 30 - 40$

Can simulate any gate set.

Wecker and Svore (2014) : LIQUi|⟩
Smelyanskiy, Sawaya, Aspuru-Guzik (2016) : qHIPSTER
Steiger, Haner, Troyer (2016) : ProjectQ



Qubits	n
Clifford gates	c
T-count	t

Stabilizer simulators: simulate Clifford gates using the Gottesman-Knill theorem.

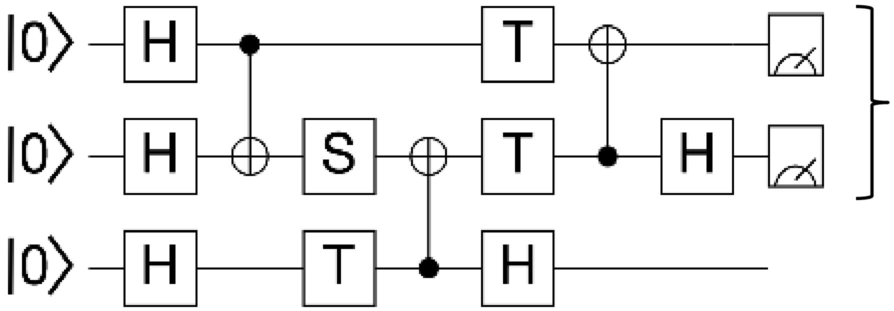
Runtime: $2^{O(t)} \cdot poly(c, n)$

Limited to a **small T-count**

How small ?

How practical ?

Aaronson and Gottesman (2004) : $2^{O(t)}$ Pauli frames
Garcia, Markov, Cross (2014) : $2^{O(t)}$ stabilizer states
Howard and Campbell (2016) : $2^{O(t)}$ Monte Carlo samples



$x \in \{0,1\}^w$

$P_{out}(x)$

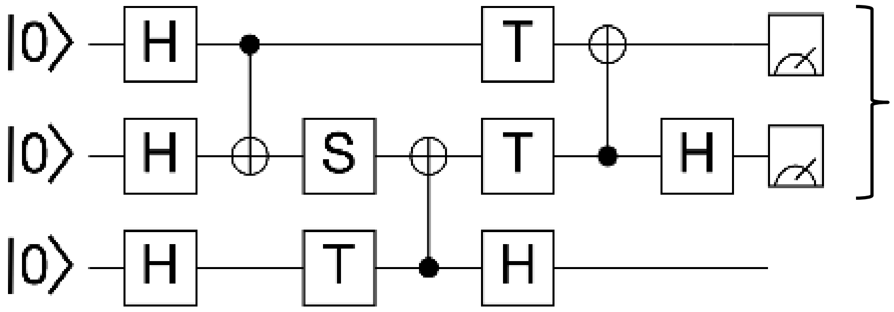
Task 1 (strong simulation):

Approximate probability $P_{out}(x)$ within a relative error ϵ for a given measurement outcome x

$$poly(n,c,t) + 2^{0.47t} \cdot t^3 \epsilon^{-2}$$

simulator runtime

Remark: strong simulation is hard even for a quantum computer.



$x \in \{0,1\}^w$

$P_{out}(x)$

w - output width

Task 2 (weak simulation):

Sample x from a probability distribution which is ϵ–close to $P_{out}(x)$ in the L_1-norm

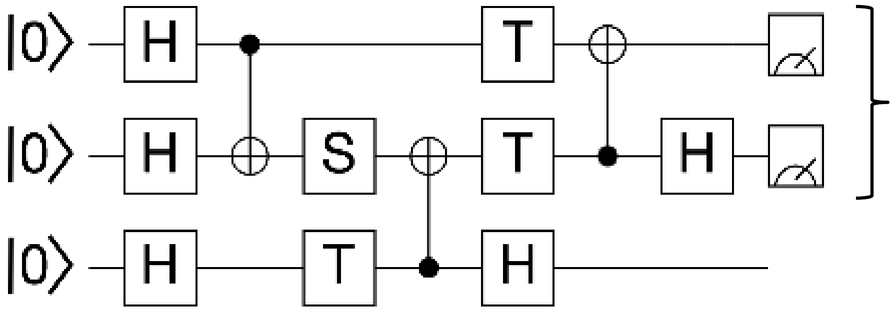
$$poly(n,c,t) + 2^{0.23t} \cdot t^3 w^4 \epsilon^{-5}$$

simulator runtime

practical ?

MATLAB implementation; 5 CPU hours:

$$t \approx 50, \quad n \approx 50, \quad c \approx 1000, \quad w = 1, \quad \epsilon \approx 10\%$$



$x \in \{0,1\}^w$

$P_{out}(x)$

w - output width

Task 2 (weak simulation):

Sample x from a probability distribution which is ϵ-close to $P_{out}(x)$ in the L_1-norm

$$poly(n,c,t) + 2^{0.23t} \cdot t^3 w^4 \epsilon^{-5}$$

simulator runtime

$$poly(n,c,t) + 2^{0.23t} \cdot t^3 w^3 \epsilon^{-3}$$

simulator runtime

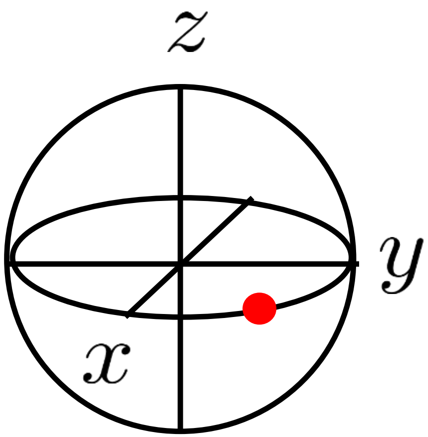
unpublished

Outline of the simulation algorithm

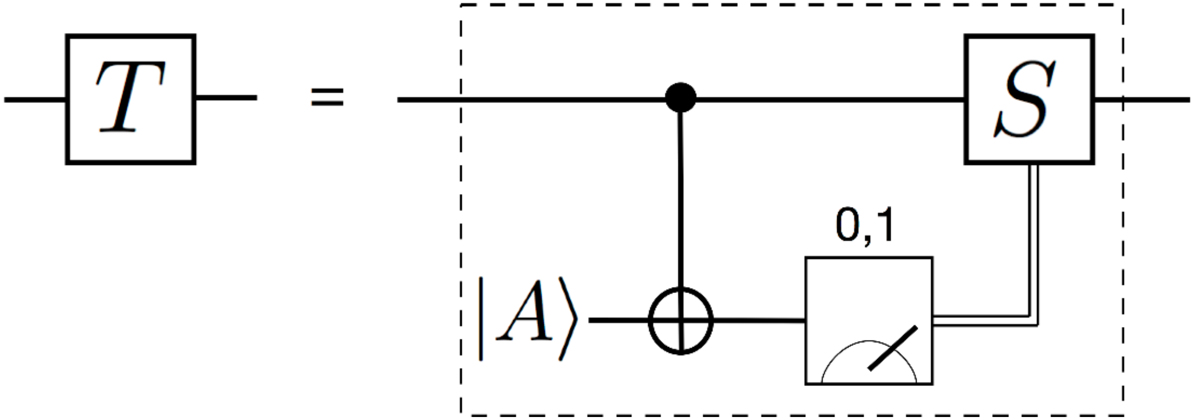
- Magic states and gadgetized circuits
- Stabilizer rank of magic states
- Fast norm estimation algorithm

$$\boxed{|A\rangle = (\,|0\rangle + e^{i\pi/4}|1\rangle\,)/\sqrt{2}}$$

"magic" state



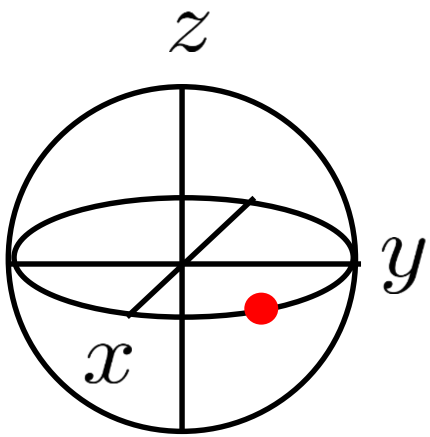
T-gadget



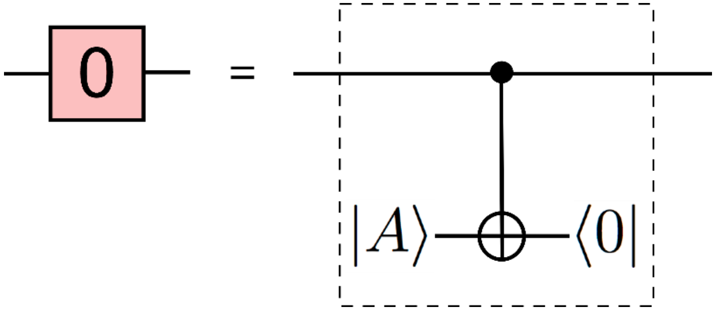
$$\Pr(0) = \Pr(1) = \frac{1}{2}$$

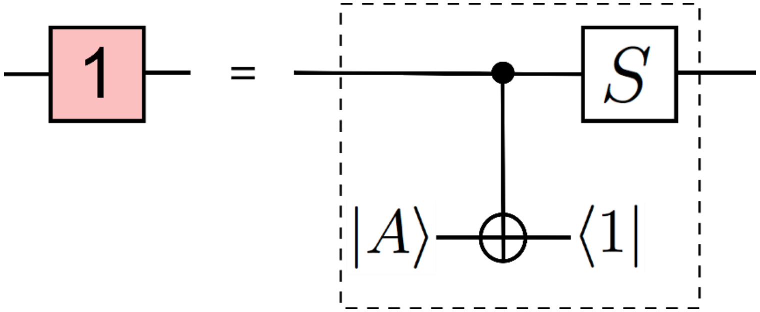
$$\boxed{|A\rangle = (\,|0\rangle + e^{i\pi/4}|1\rangle\,)/\sqrt{2}}$$

"magic" state

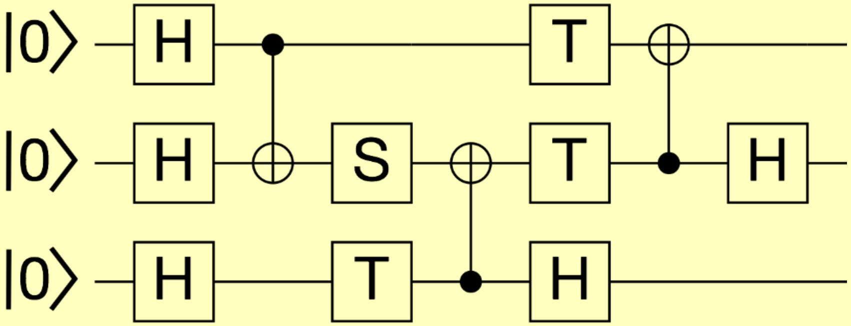


Postselected T-gadgets (implement $T/\sqrt{2}$)



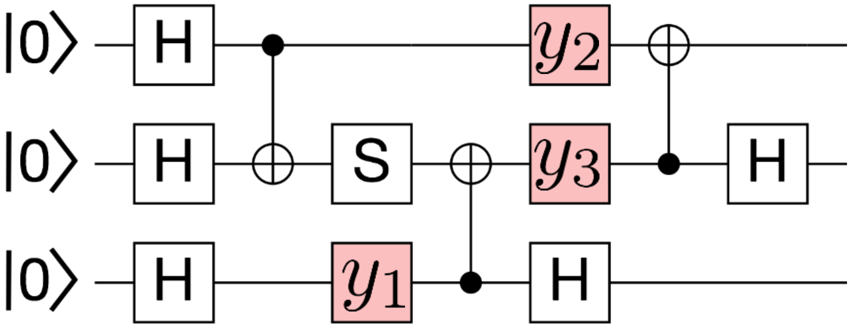


Clifford+T circuit
U



$U|0^n\rangle$

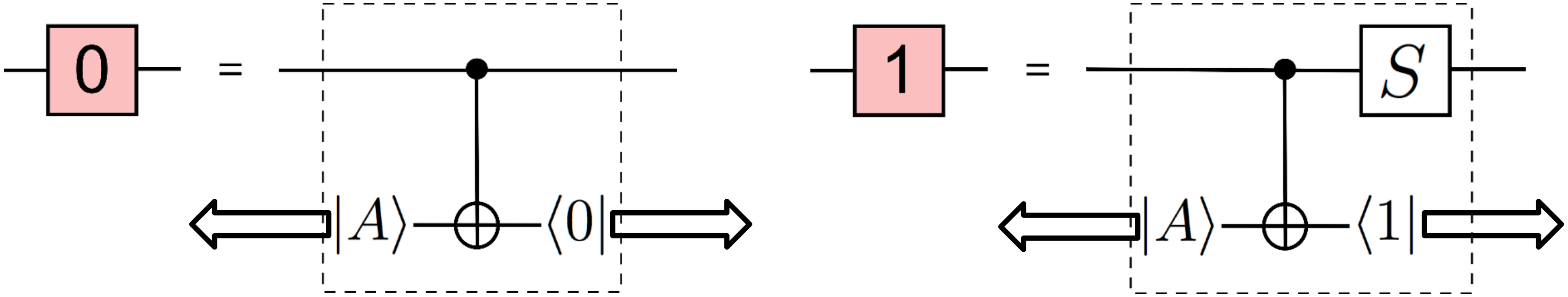
$=$



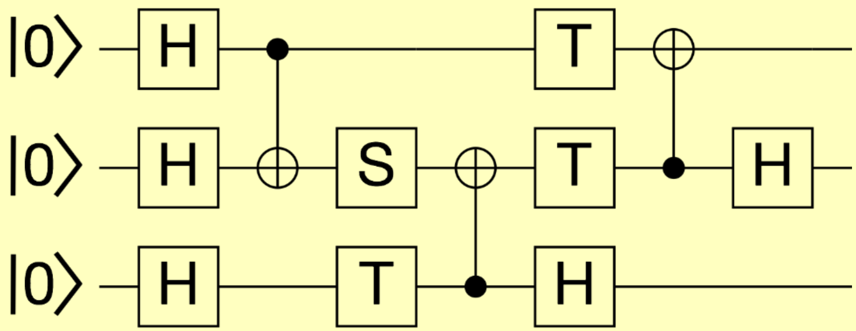
$U|0^n\rangle$

$\forall\ y_i \in \{0,1\}$

Postselected T-gadgets (implement $T/\sqrt{2}$)

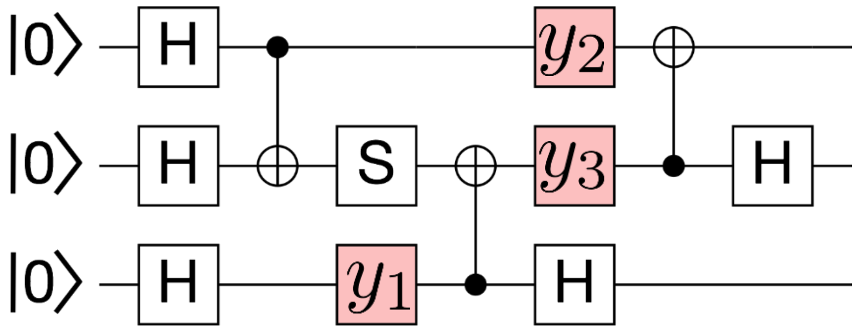


Clifford+T circuit
U



$U|0^n\rangle$

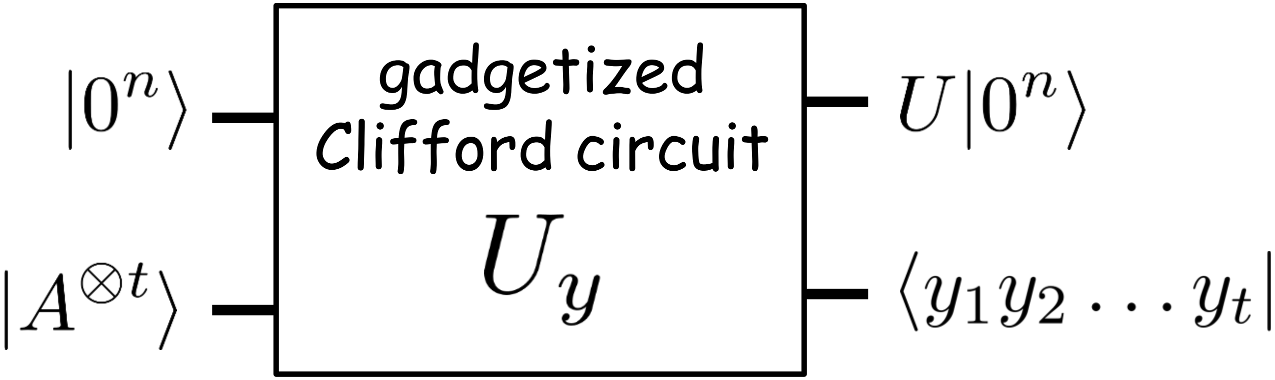
$=$



$U|0^n\rangle$

$\forall\ y_i \in \{0,1\}$

$=$



$|0^n\rangle$

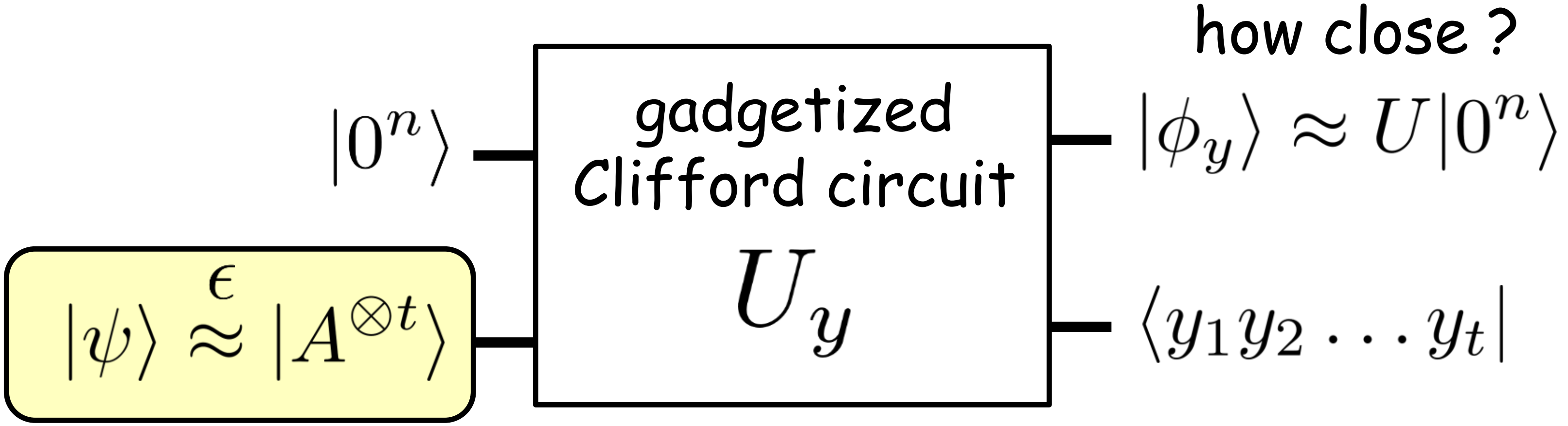
$|A^{\otimes t}\rangle$

gadgetized
Clifford circuit
U_y

$U|0^n\rangle$

$\langle y_1 y_2 \ldots y_t|$

Approximate gadgetized circuit:

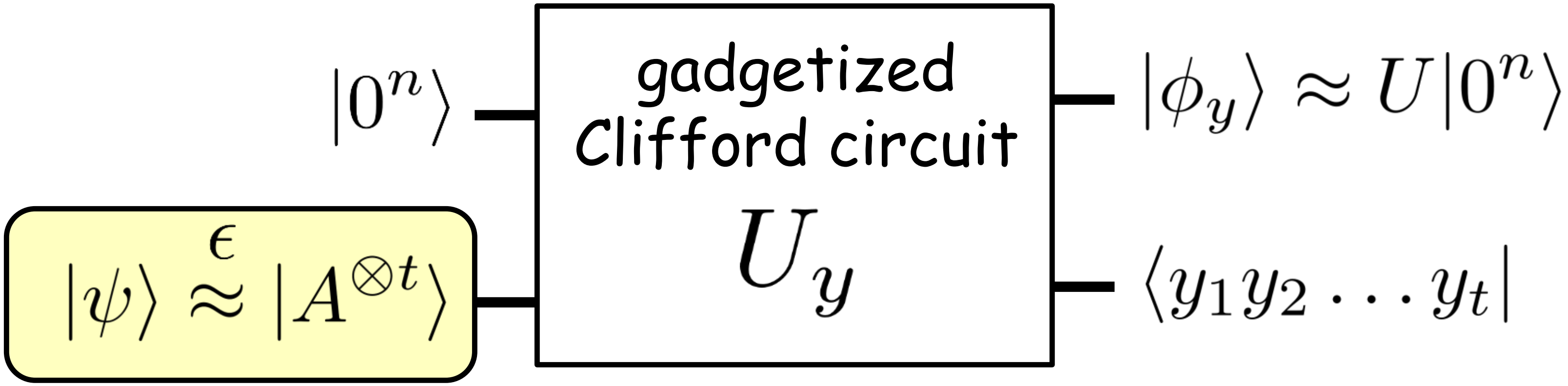


Lemma The approximate gadgetized circuit with random y simulates an approximate target circuit:

$$\| U|0^n\rangle\langle 0^n|U^\dagger - \frac{1}{2^t} \sum_{y\in\{0,1\}^t} |\phi_y\rangle\langle\phi_y| \|_1 \le O(\epsilon)$$

$$\epsilon \equiv \|\psi - A^{\otimes t}\|$$

Approximate gadgetized circuit:



$|0^n\rangle$

gadgetized Clifford circuit U_y

$|\phi_y\rangle \approx U|0^n\rangle$

$\langle y_1 y_2 \ldots y_t|$

$|\psi\rangle \overset{\epsilon}{\approx} |A^{\otimes t}\rangle$

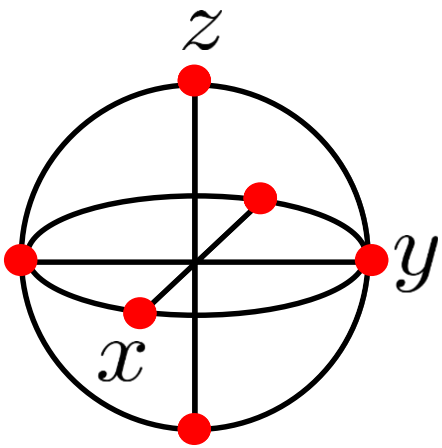
Corollary: weak simulation of Clifford+T circuits reduces to strong simulation of random postselective Clifford circuits with the initial state $|\psi\rangle \approx |A^{\otimes t}\rangle$

Our method: try to choose ψ as a linear combination of a few stabilizer states; use Gottesman-Knill theorem.

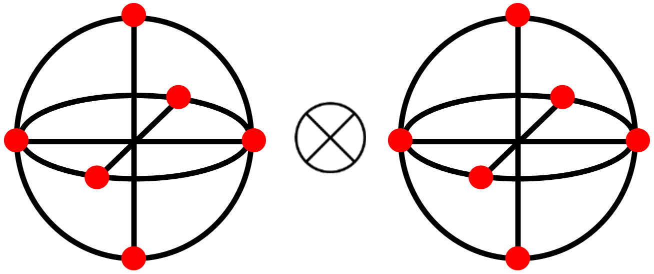
- Magic states and gadgetized circuits
- Stabilizer rank of magic states
- Fast norm estimation algorithm

Stabilizer states: $|\psi\rangle = \underset{\uparrow}{U}|0^n\rangle$

Clifford unitary



1 qubit: 6 states



$$(I \otimes {\color{red}U})(|00\rangle + |11\rangle)/\sqrt{2}$$

2 qubits: 60 states

$\mathcal{S}_n$ set of all n-qubit stabilizer states

$$|\mathcal{S}_n| \sim 2^{0.5n^2}$$

Stabilizer rank $\chi(\psi)$: smallest χ such that ψ can be written as a linear combination of χ stabilizer states:

$$|\psi\rangle = \sum_{a=1}^{\chi} c_a |\psi_a\rangle, \quad \psi_a \in \mathcal{S}_n$$

Stabilizer rank $\chi_\epsilon(\psi)$: smallest χ such that ψ is ϵ-close to a linear combination of χ stabilizer states:

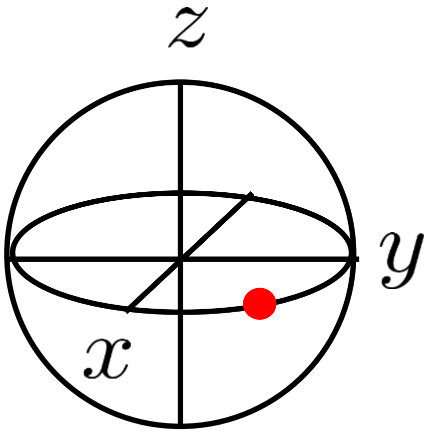
$$\|\, |\psi\rangle - \sum_{a=1}^{\chi} c_a |\psi_a\rangle \,\| \le \epsilon, \qquad \psi_a \in \mathcal{S}_n$$

Example: magic state

$$\boxed{|A\rangle \sim |0\rangle + e^{i\pi/4}|1\rangle}$$

$$\chi(A) = 2$$

$$\chi(A^{\otimes 2}) = ?$$



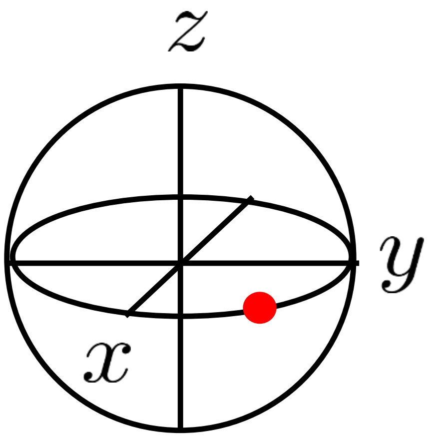
$$|A^{\otimes 2}\rangle \sim \underbrace{(|00\rangle + i|11\rangle)}_{\text{stabilizer state}} + e^{i\pi/4}\underbrace{(|01\rangle + |10\rangle)}_{\text{stabilizer state}}$$

Example: magic state

$$\boxed{|A\rangle \sim |0\rangle + e^{i\pi/4}|1\rangle}$$

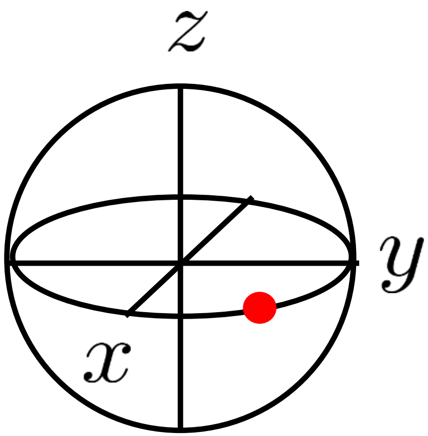
$$\chi(A) = 2$$

$$\chi(A^{\otimes 2}) = 2$$



$$|A^{\otimes 2}\rangle \sim \underbrace{(|00\rangle + i|11\rangle)}_{\text{stabilizer state}} + e^{i\pi/4}\underbrace{(|01\rangle + |10\rangle)}_{\text{stabilizer state}}$$

Example: magic state $\boxed{|A\rangle \sim |0\rangle + e^{i\pi/4}|1\rangle}$



$$\chi(A) = 2$$

$$\chi(A^{\otimes 2}) = 2$$

$$\left.\begin{aligned} \chi(A^{\otimes 3}) &\le 3 \\ \chi(A^{\otimes 4}) &\le 4 \\ \chi(A^{\otimes 5}) &\le 6 \\ \chi(A^{\otimes 6}) &\le 7 \end{aligned}\right\}$$

numerical search

Why should we care ?

SB, Smith, Smolin, Phys. Rev. X, 6, 021034 (2016)

Implications for simulation of Clifford+T circuits:

Strong simulator's runtime:

$$poly(n,c,t) + \chi(A^{\otimes t}) \cdot t^3 \epsilon^{-2}$$

Weak simulator's runtime:

$$poly(n,c,t) + \chi_\epsilon(A^{\otimes t}) \cdot t^3 w^4 \epsilon^{-3}$$

Sub-exponential upper bounds on the stabilizer rank of magic states gives sub-exponential classical simulator for constant depth Clifford+T circuits !

Best known upper bounds

$$\boxed{\chi(A^{\otimes n}) \leq 7^{n/6} \approx 2^{0.47n}}$$

SB, Smith, Smolin (2016)

Proof:

$$\chi(A^{\otimes 6}) \leq 7$$

$$\chi(\psi \otimes \phi) \leq \chi(\psi)\chi(\phi)$$

$$\boxed{\chi_\epsilon(A^{\otimes n}) \leq \epsilon^{-2} \cos\left(\pi/8\right)^{-2n} \approx \epsilon^{-2} \cdot 2^{0.23n}}$$

new result

$$\boxed{\chi_\epsilon(A^{\otimes n}) \le \epsilon^{-2} \cos{(\pi/8)}^{-2n} \approx \epsilon^{-2} \cdot 2^{0.23n}}$$

sketch of the proof:

$$R \equiv \frac{1}{\sqrt{2}} \left[\begin{array}{cc} 1 & e^{-i\pi/4} \\ 1 & e^{i\pi/4} \end{array} \right] \qquad \left. \begin{array}{c} R|0\rangle \\ R|1\rangle \end{array} \right\}$$

stabilizer states

Approximate $A^{\otimes n}$ by subset states:

$$|\psi_M\rangle \sim R^{\otimes n} \sum_{x\in M} |x\rangle \qquad M \subseteq \{0,1\}^n$$

Stabilizer rank: $\chi(\psi_M) \le |M|$

Simple facts

$$|M| = 2^n \quad \Rightarrow \quad |\psi_M\rangle = |A^{\otimes n}\rangle$$

$$|M| = 1 \quad \Rightarrow \quad |\langle A^{\otimes n}|\psi_M\rangle| = \max_{\psi\in\mathcal{S}_n} |\langle A^{\otimes n}|\psi\rangle|$$

$$\boxed{\chi_\epsilon(A^{\otimes n}) \le \epsilon^{-2}\cos{(\pi/8)}^{-2n} \approx \epsilon^{-2}\cdot 2^{0.23n}}$$

sketch of the proof:

$$R \equiv \frac{1}{\sqrt{2}}\left[\begin{array}{cc} 1 & e^{-i\pi/4} \\ 1 & e^{i\pi/4} \end{array}\right]$$

$$\left.\begin{array}{c} R|0\rangle \\ R|1\rangle \end{array}\right\} \text{stabilizer states}$$

Approximate $A^{\otimes n}$ by subset states:

$$|\psi_M\rangle \sim R^{\otimes n}\sum_{x\in M}|x\rangle \qquad M \subseteq \{0,1\}^n$$

Lemma Suppose $M \subseteq \{0,1\}^n$ is a random linear subspace with a fixed dimension k.

$$2^k \ge \epsilon^{-2}(\cos{(\pi/8)})^{-2n} \Longrightarrow \|A^{\otimes n} - \psi_M\| \le \epsilon$$

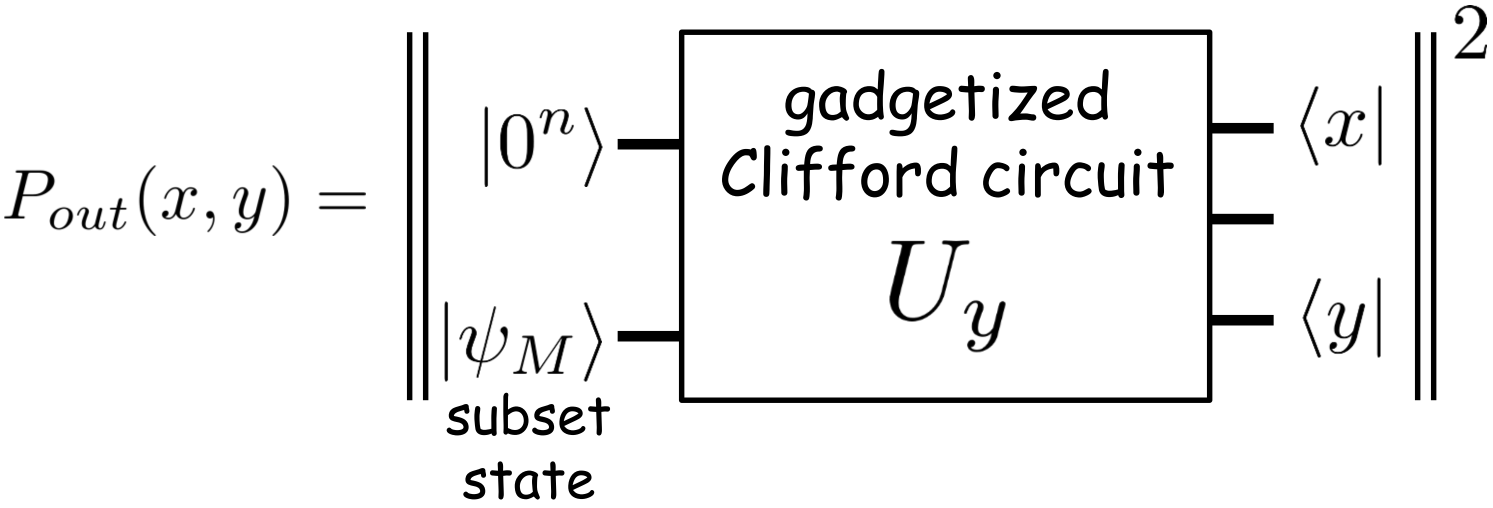
w.h.p.

- Magic states and gadgetized circuits
- Stabilizer rank of magic states
- Fast norm estimation algorithm

Reminder: weak simulation of Clifford+T circuits reduces to strong simulation of random postselective Clifford circuits with the initial state $|\psi\rangle \approx |A^{\otimes t}\rangle$

Reminder: weak simulation of Clifford+T circuits reduces to strong simulation of random postselective Clifford circuits with the initial state $\underset{\substack{\uparrow \\ \text{subset state}}}{|\psi_M\rangle} \approx |A^{\otimes t}\rangle$

Reminder: weak simulation of Clifford+T circuits reduces to strong simulation of random postselective Clifford circuits with the initial state $|\psi_M\rangle \approx |A^{\otimes t}\rangle$

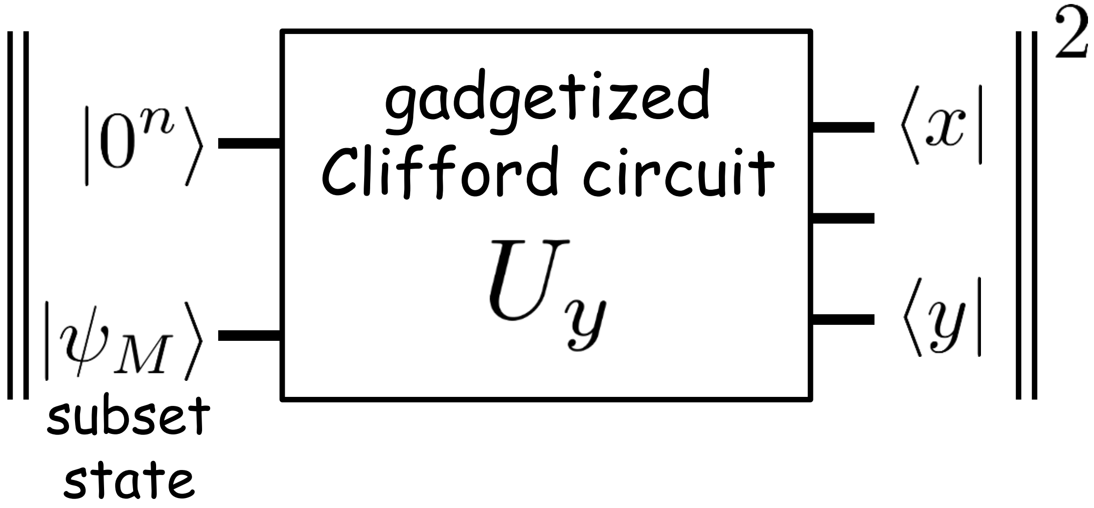


$x \in \{0,1\}^w$: output

$y \in \{0,1\}^t$: postselection

Want: compute $P_{out}(x,y)$ with a small relative error

$P_{out}(x,y) =$



Gottesman-Knill preprocessing:

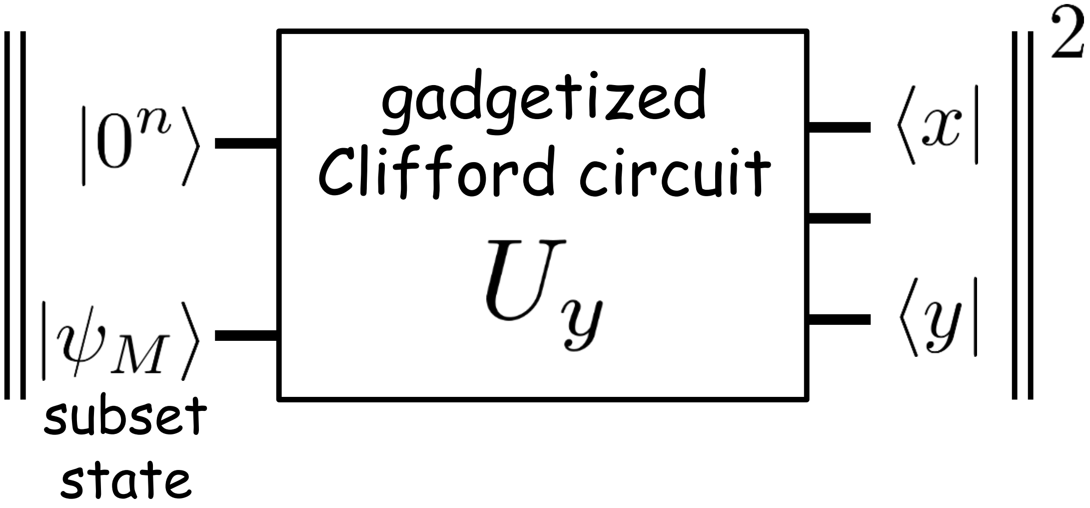
$$\boxed{P_{out}(x,y) = 2^k \cdot \| \Pi |\psi_M\rangle \|^2}$$

$k \in \mathbb{Z}$

Π - postselective Clifford circuit (gates=projectors)

We can compute k and Π in time $poly(n,c,t)$

$P_{out}(x, y) =$



Gottesman-Knill preprocessing:

$$\boxed{P_{out}(x, y) = 2^k \cdot \|\Pi|\psi_M\rangle\|^2}$$

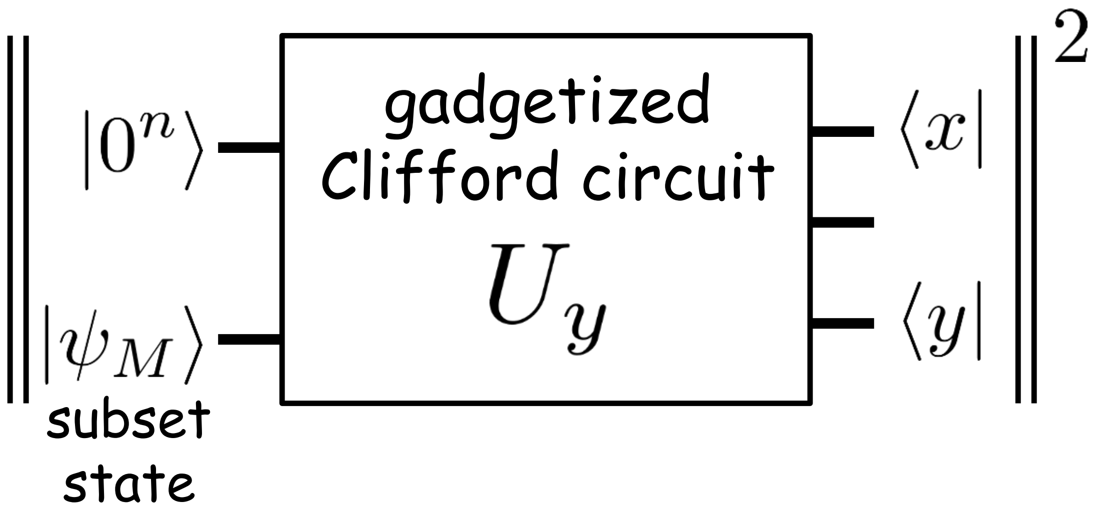
$k \in \mathbb{Z}$

Π - postselective Clifford circuit (gates=projectors)

$\Pi|\psi_M\rangle$ - sum of roughly $2^{0.23t}$ stabilizer states

(since Π maps stabilizer states to stabilizer states)

$P_{out}(x,y) =$



Gottesman-Knill preprocessing:

$$\boxed{P_{out}(x,y) = 2^k \cdot \|\Pi|\psi_M\rangle\|^2}$$

$k \in \mathbb{Z}$

Π - postselective Clifford circuit (gates=projectors)

$\Pi|\psi_M\rangle$ - sum of roughly $2^{0.23t}$ stabilizer states

Want: norm estimation algorithm for states with VERY LARGE stabilizer rank

Stabilizer states: algorithmic tools

problem	runtime
Inner Product Given $\psi, \phi \in \mathcal{S}_n$ Compute $\langle \psi \| \phi \rangle = 2^{-p/2} e^{i\pi m/4}$	$O(n^3)$
Pauli Measurement Given $\psi \in \mathcal{S}_n$ $P \in \mathrm{Pauli}(n)$ Compute $\| \phi \rangle = (1/2)(I + P) \| \psi \rangle \in \mathcal{S}_n$	$O(n^2)$
Random Stabilizer State Generate uniform random $\psi \in \mathcal{S}_n$	$O(n^2)$

Stabilizer states: algorithmic tools

problem	runtime
Inner Product Given $\psi, \phi \in \mathcal{S}_n$ Compute $\langle\psi\|\phi\rangle = 2^{-p/2} e^{i\pi m/4}$	$O(n^3)$
Pauli Measurement Given $\psi \in \mathcal{S}_n$ $P \in \mathrm{Pauli}(n)$ Compute $\|\phi\rangle = (1/2)(I+P)\|\psi\rangle \in \mathcal{S}_n$	$O(n^2)$
Random Stabilizer State Generate uniform random $\psi \in \mathcal{S}_n$	$O(n^2)$ new ?

Stabilizer states: algorithmic tools

problem	runtime
Norm Estimation Given $\lvert\phi\rangle = \sum_{a=1}^{\chi} c_a \lvert\phi_a\rangle, \quad \phi_a \in \mathcal{S}_n$ Estimate $\lVert\phi\rVert$ within relative error δ	$O(\chi n^3 \delta^{-2})$ Linear in χ !

Stabilizer states: algorithmic tools

problem	runtime
Norm Estimation Given $\vert\phi\rangle = \sum_{a=1}^{\chi} c_a \vert\phi_a\rangle, \quad \phi_a \in \mathcal{S}_n$ Estimate $\Vert\phi\Vert$ within relative error δ	$O(\chi n^3 \delta^{-2})$ Linear in χ !

Brute-force algorithm: $O(\chi^2 n^3)$

$$\|\phi\|^2 = \langle\phi|\phi\rangle = \sum_{a,b=1}^{\chi} \bar{c}_a c_b \langle\phi_a|\phi_b\rangle$$

compute each term in time $O(n^3)$

Stabilizer states: algorithmic tools

problem	runtime
Norm Estimation Given $\vert\phi\rangle = \sum_{a=1}^{\chi} c_a \vert\phi_a\rangle, \quad \phi_a \in \mathcal{S}_n$ Estimate $\Vert\phi\Vert$ within relative error δ	$O(\chi n^3 \delta^{-2})$ Square-root speedup !

Brute-force algorithm: $O(\chi^2 n^3)$

$$\|\phi\|^2 = \langle\phi|\phi\rangle = \sum_{a,b=1}^{\chi} \bar{c}_a c_b \langle\phi_a|\phi_b\rangle$$

compute each term in time $O(n^3)$

$$|\phi\rangle = \sum_{a=1}^{\chi} c_a |\phi_a\rangle, \quad \phi_a \in \mathcal{S}_n \qquad\qquad \|\phi\| =?$$

Key idea: compute inner products between ϕ and random stabilizer states

$$\xi \equiv 2^n \cdot |\langle\psi|\phi\rangle|^2$$

$\psi \in \mathcal{S}_n$ random uniformly distributed

$$\underbrace{\mathbb{E}(\xi) = \|\phi\|^2 \qquad \mathrm{Var}(\xi) = \frac{2^n - 1}{2^n + 1}\|\phi\|^4 \approx \|\phi\|^4}_{\text{Use the fact that } S_n \text{ is a 2-design}}$$

$$|\phi\rangle = \sum_{a=1}^{\chi} c_a |\phi_a\rangle, \quad \phi_a \in \mathcal{S}_n \qquad \|\phi\| =?$$

Key idea: compute inner products between ϕ and random stabilizer states

$$\xi \equiv 2^n \cdot |\langle\psi|\phi\rangle|^2$$

$\psi \in \mathcal{S}_n$ random uniformly distributed

$$\mathbb{E}(\xi) = \|\phi\|^2 \qquad \mathrm{Var}(\xi) = \frac{2^n - 1}{2^n + 1}\|\phi\|^4 \approx \|\phi\|^4$$

We get unbiased estimator of the norm $||\phi||^2$ with a constant relative error !

$$|\phi\rangle = \sum_{a=1}^{\chi} c_a |\phi_a\rangle, \quad \phi_a \in \mathcal{S}_n \qquad\qquad \|\phi\| = ?$$

Key idea: compute inner products between ϕ and random stabilizer states

$$\xi \equiv 2^n \cdot |\langle\psi|\phi\rangle|^2$$

$\psi \in \mathcal{S}_n$ random uniformly distributed

$$\langle\psi|\phi\rangle = \sum_{a=1}^{\chi} c_a \langle\psi|\phi_a\rangle$$

each term can be computed in time $O(n^3)$

One can compute ξ in time $\underbrace{O(n^2)}_{\text{random state generation}} + \underbrace{O(\chi n^3)}_{\text{inner products}} = O(\chi n^3)$

$$\xi \equiv 2^n \cdot |\langle \psi | \phi \rangle|^2$$

$\psi \in \mathcal{S}_n$ random uniformly distributed

Unbiased estimator of the norm $||\phi||^2$ with a constant relative error. We need relative error δ

Monte Carlo: generate $K \sim \delta^{-2}$ samples $\xi_1, \xi_2, \ldots, \xi_K$.
Compute the average

$$X = K^{-1} \sum_{a=1}^{K} \xi_a$$

Chebyshev $\Longrightarrow$

$$(1-\delta)\|\phi\|^2 \le X \le (1+\delta)\|\phi\|^2$$

w.h.p.

Overall running time: $O(\chi n^3 K) = O(\chi n^3 \delta^{-2})$

Want to compute:

$$\boxed{P_{out}(x,y) = 2^k \cdot \|\Pi|\psi_M\rangle\|^2}$$

Π – postselective Clifford circuit (gates=projectors)

$|\psi_M\rangle$ – sum of roughly $2^{0.23t}$ stabilizer states

$$\begin{tikzcd}
\|\Pi|\psi_M\rangle\|^2 \arrow[d, Rightarrow, "\text{norm estimation}"] & \boxed{\textbf{workflow}} & \langle\phi'|\psi_M\rangle \\
\begin{array}{c}\langle\phi|\Pi|\psi_M\rangle \\ \phi \in \mathcal{S}_t \\ \text{random}\end{array} \arrow[rr, Rightarrow, "\text{Pauli measurement}"'] & & |\phi'\rangle = \Pi|\phi\rangle \in \mathcal{S}_t \arrow[u, Rightarrow, "2^{0.23t}\ \text{inner products}"']
\end{tikzcd}$$

Implementation and benchmarking

How to choose benchmark circuits ?

Desiderata:

1. Deterministic output:

$$U|0^n\rangle = |s_1, s_2, \ldots, s_n\rangle$$

2. Small T-count ($t \le 50$)

3. Large number of qubits ($n \ge 40$)

Hidden shift quantum algorithm for bent functions

Roetteler (2010)

Hidden shift quantum algorithm for bent functions

Roetteler (2010)

$$f \ : \ \{0,1\}^n \rightarrow \{+1,-1\}$$

Hidden shift quantum algorithm for bent functions

Roetteler (2010)

$$f \; : \; \{0,1\}^n \rightarrow \{+1,-1\}$$

Reminder: f is a bent function if both f and the Hadamard transform of f take values ± 1

Hidden shift quantum algorithm for bent functions

Roetteler (2010)

$$f\ :\ \{0,1\}^n \rightarrow \{+1,-1\}$$

$s \in \{0,1\}^n$ hidden shift string

Hidden shift quantum algorithm for bent functions

Roetteler (2010)

$$f\ :\ \{0,1\}^n \rightarrow \{+1,-1\}$$

$s \in \{0,1\}^n$ hidden shift string

Oracles:

$$x \longrightarrow \boxed{O} \longrightarrow f(x)$$

$$x \longrightarrow \boxed{O_s} \longrightarrow \hat{f}(x \oplus s)$$

Hidden shift quantum algorithm for bent functions Roetteler (2010)

$$f \,:\, \{0,1\}^n \rightarrow \{+1,-1\}$$

$s \in \{0,1\}^n$ hidden shift string

Oracles:

$$x \rightarrow \boxed{O} \rightarrow f(x)$$

$$x \rightarrow \boxed{O_s} \rightarrow \hat{f}(x \oplus s)$$

Promise:

f is a bent function

Hidden shift quantum algorithm for bent functions

Roetteler (2010)

$$f\ :\ \{0,1\}^n \rightarrow \{+1,-1\}$$

$s \in \{0,1\}^n$ hidden shift string

Oracles:

$$x \longrightarrow \boxed{O} \longrightarrow f(x)$$

$$x \longrightarrow \boxed{O_s} \longrightarrow \hat{f}(x \oplus s)$$

Promise:

f is a bent function

Problem: find s by making as few queries to the oracles as possible

Hidden shift quantum algorithm for bent functions Roetteler (2010)

$$f\ :\ \{0,1\}^n \rightarrow \{+1,-1\}$$

$s \in \{0,1\}^n$ hidden shift string

Oracles:

$$x \longrightarrow \boxed{O} \longrightarrow f(x)$$

$$x \longrightarrow \boxed{O_s} \longrightarrow \hat{f}(x \oplus s)$$

Promise:

f is a bent function

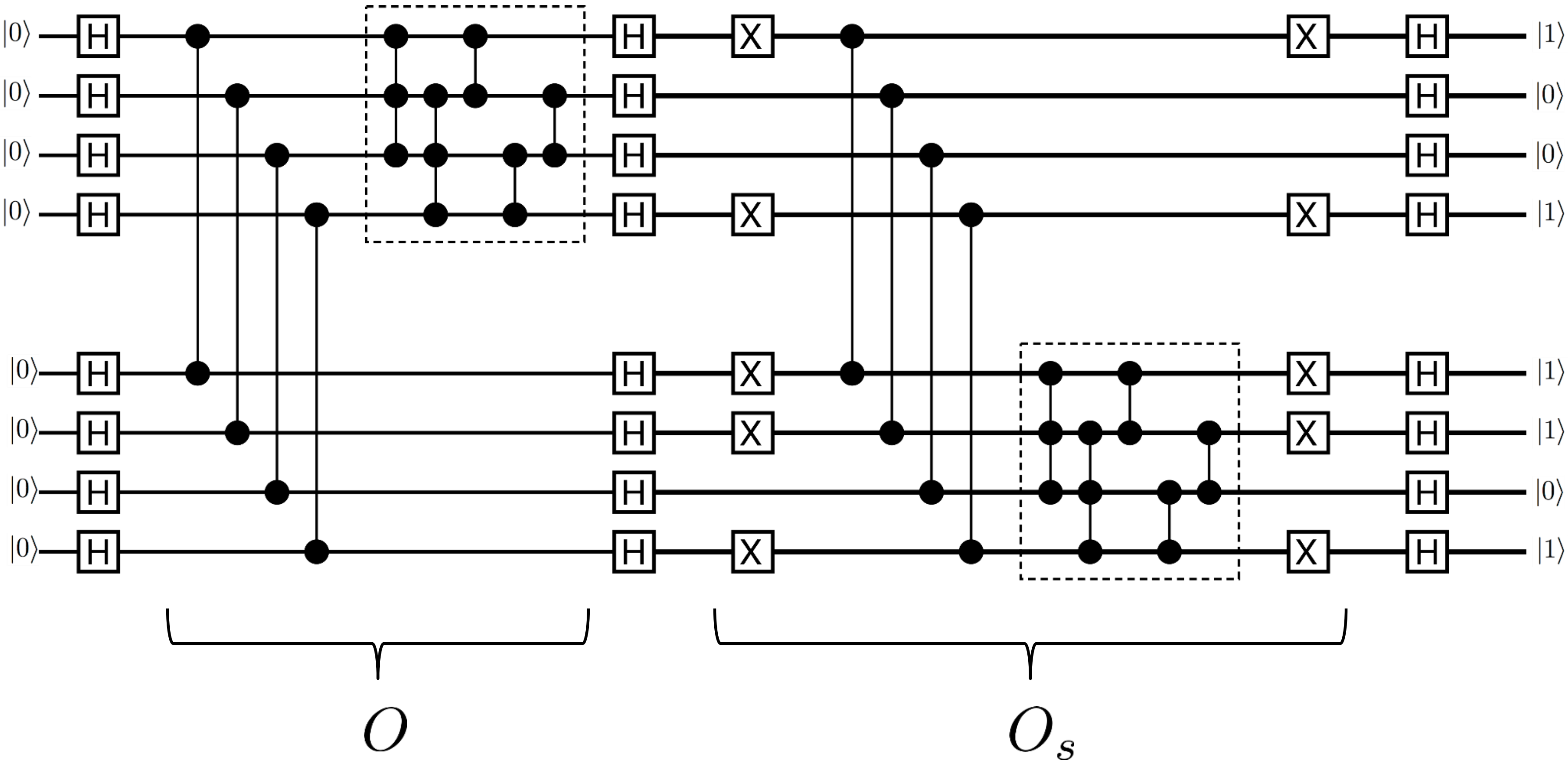
Problem: find s by making as few queries to the oracles as possible

Quantum algorithm: two queries

$$|s\rangle = H^{\otimes n} O_s H^{\otimes n} O H^{\otimes n} |0^n\rangle$$

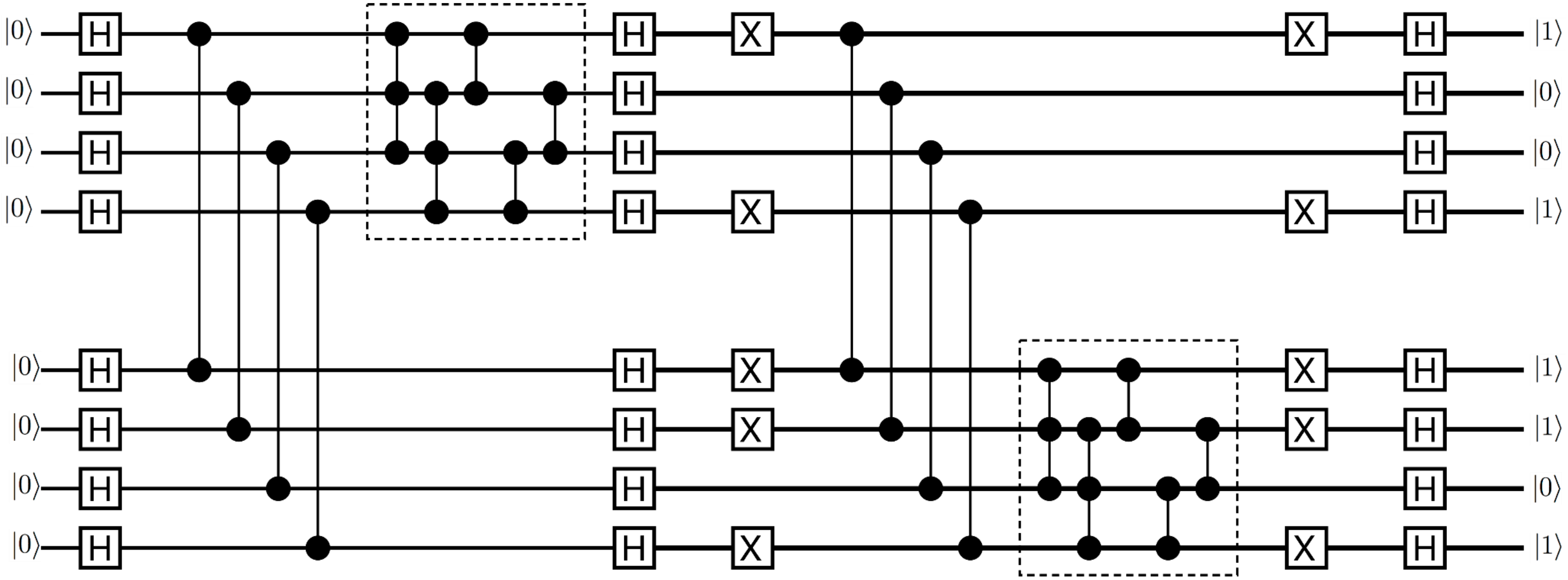
Classical algorithms: $\Omega(n)$ queries

Example for $n = 8$



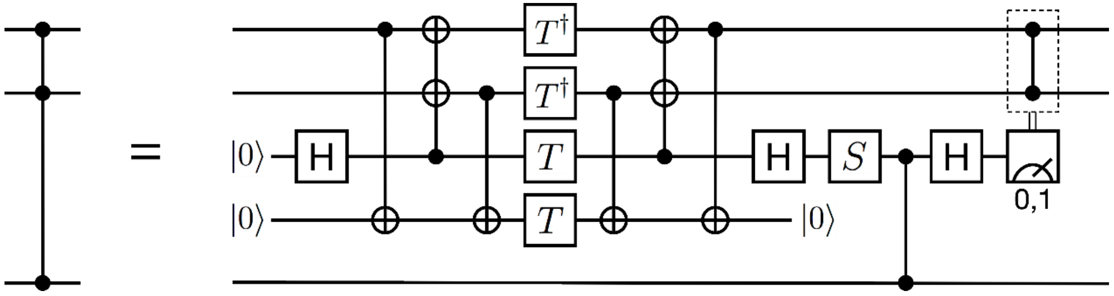
Hidden shift string: $s = 10011101$

Example for $n = 8$

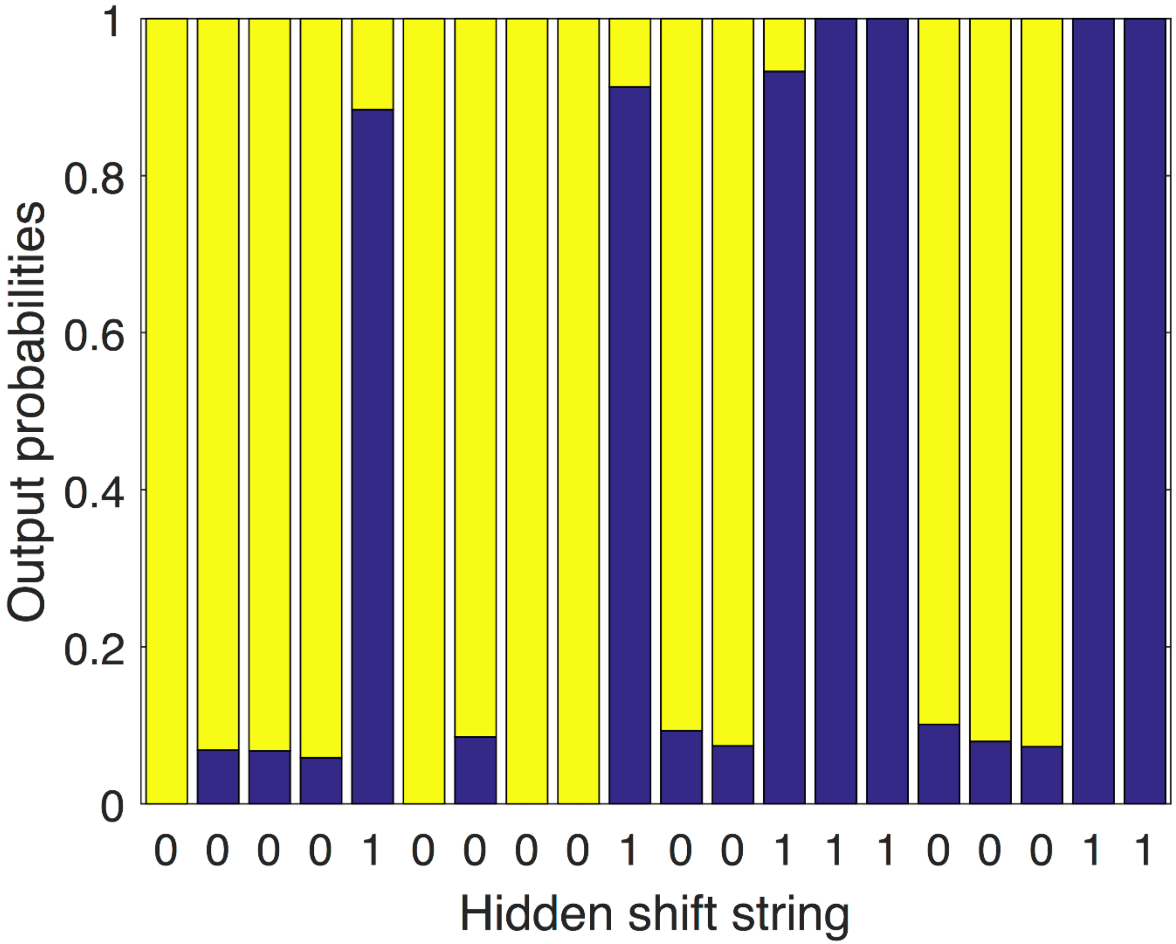


CCZ gadget:

Jones (2013)

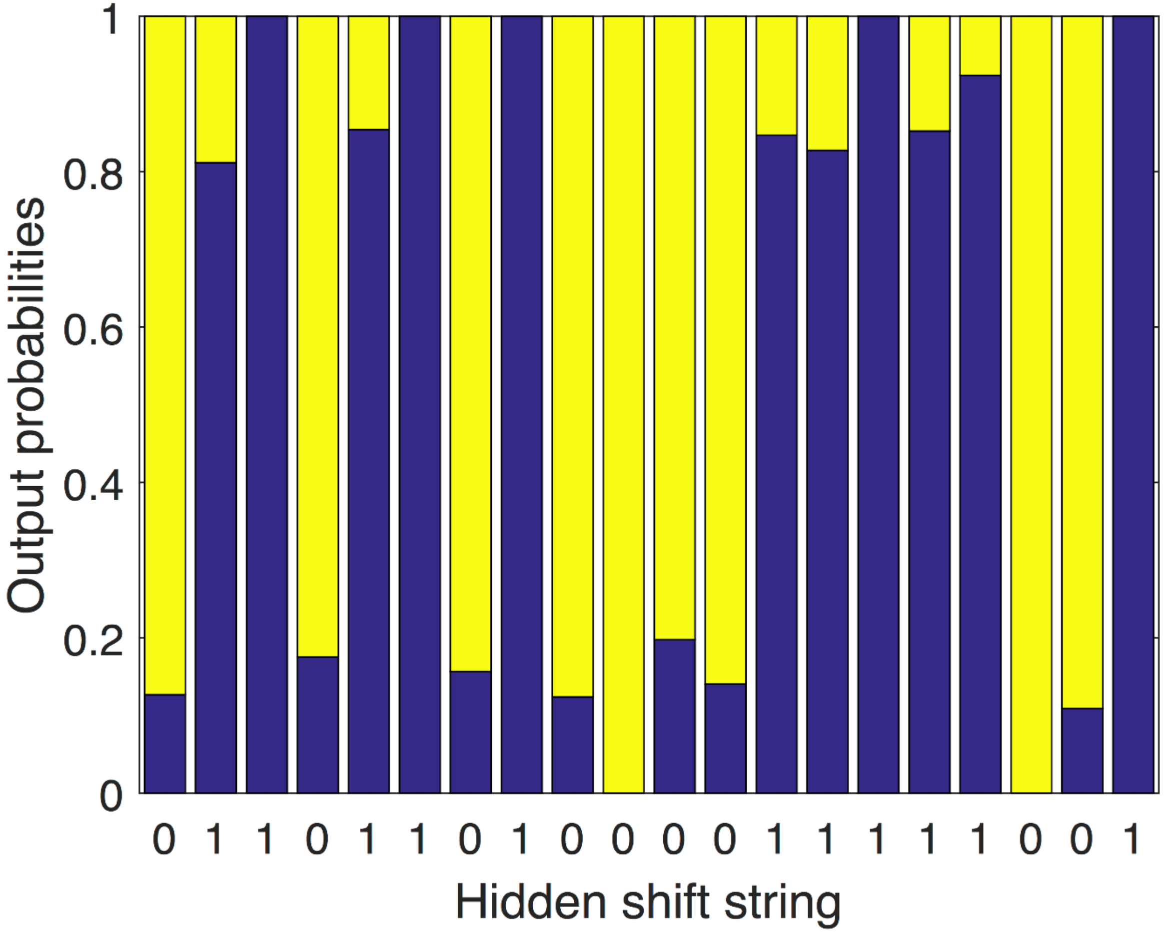


Numerical results



$$n = 40, \quad t = 40, \quad c \sim 10^3$$

Numerical results

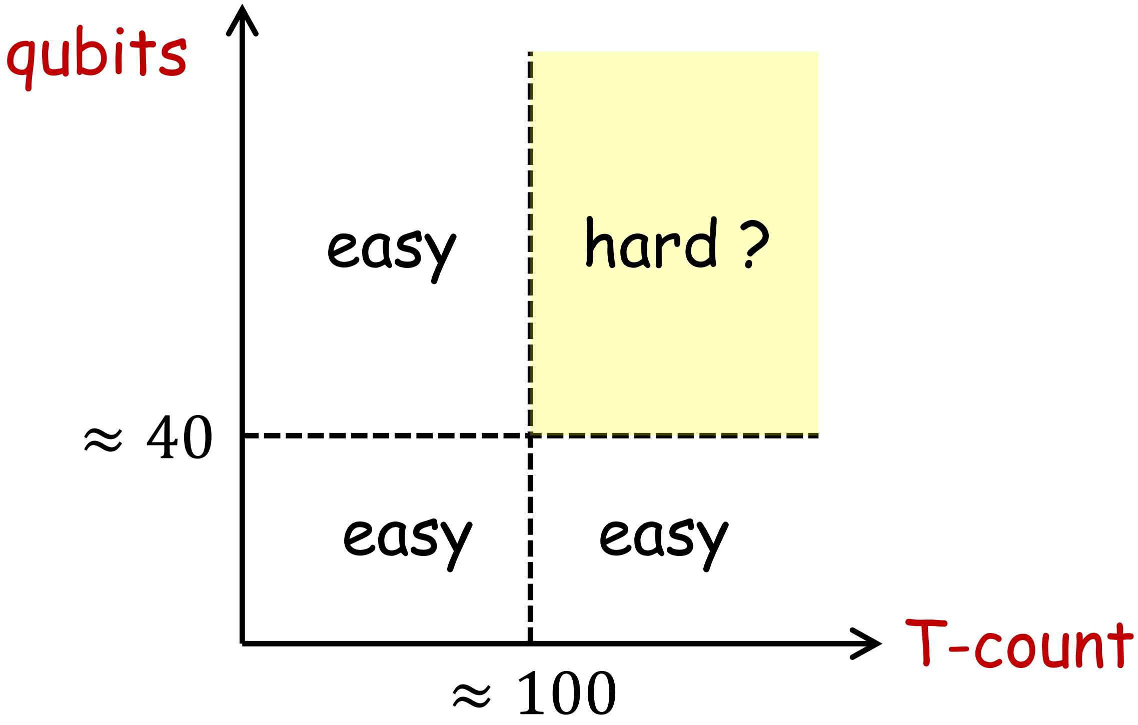


$$n = 40, \quad t = 48, \quad c \sim 10^3$$

Summary

New algorithmic tools for stabilizer-based simulators:

- Random gadgetized circuits
- Low-rank stabilizer approximations of magic states
- Fast norm estimation



Open problems and future work

- More efficient (parallel) implementation
- Better upper/lower bounds on the stabilizer rank
- Stabilizer rank reduction algorithms