Title: Fast estimation of outcome probabilities for quantum circuits

Speakers: Hakop Pashayan

Series: Perimeter Institute Quantum Discussions

Date: March 31, 2021 - 4:00 PM

URL: http://pirsa.org/21030043

Abstract: We present two classical algorithms for the simulation of universal quantum circuits on n qubits constructed from c instances of Clifford gates and t arbitrary-angle Z-rotation gates such as T gates. Our algorithms complement each other by performing best in different parameter regimes. The Estimate algorithm produces an additive precision estimate of the Born rule probability of a chosen measurement outcome with the only source of run-time inefficiency being a linear dependence on the stabilizer extent (which scales like â‰ˆ1.17^t for T gates). Our algorithm is state-of-the-art for this task:

as an example, in approximately 25 hours (on a standard desktop computer), we estimated the Born rule probability to within an additive error of 0.03, for a 50 qubit, 60 non-Clifford gate quantum circuit with more than 2000 Clifford gates. The Compute algorithm calculates the probability of a chosen measurement outcome to machine precision with run-time O(2^tâˆ’r(tâˆ’r)t) where r is an efficiently computable, circuit-specific quantity. With high probability, r is very close to min{t,nâˆ’w} for random circuits with many Clifford gates, where w is the number of measured qubits. Compute can be effective in surprisingly challenging parameter regimes, e.g., we can randomly sample Clifford+T circuits with n=55, w=5, c=105 and t=80 T-gates, and then compute the Born rule probability with a run-time consistently less than 104 seconds using a single core of a standard desktop computer. We provide a C+Python implementation of our algorithms.

# Fast estimation of outcome probabilities for quantum circuits

Hakop Pashayan, Oliver Reardon-Smith, Kamil Korzekwa, Stephen Bartlett

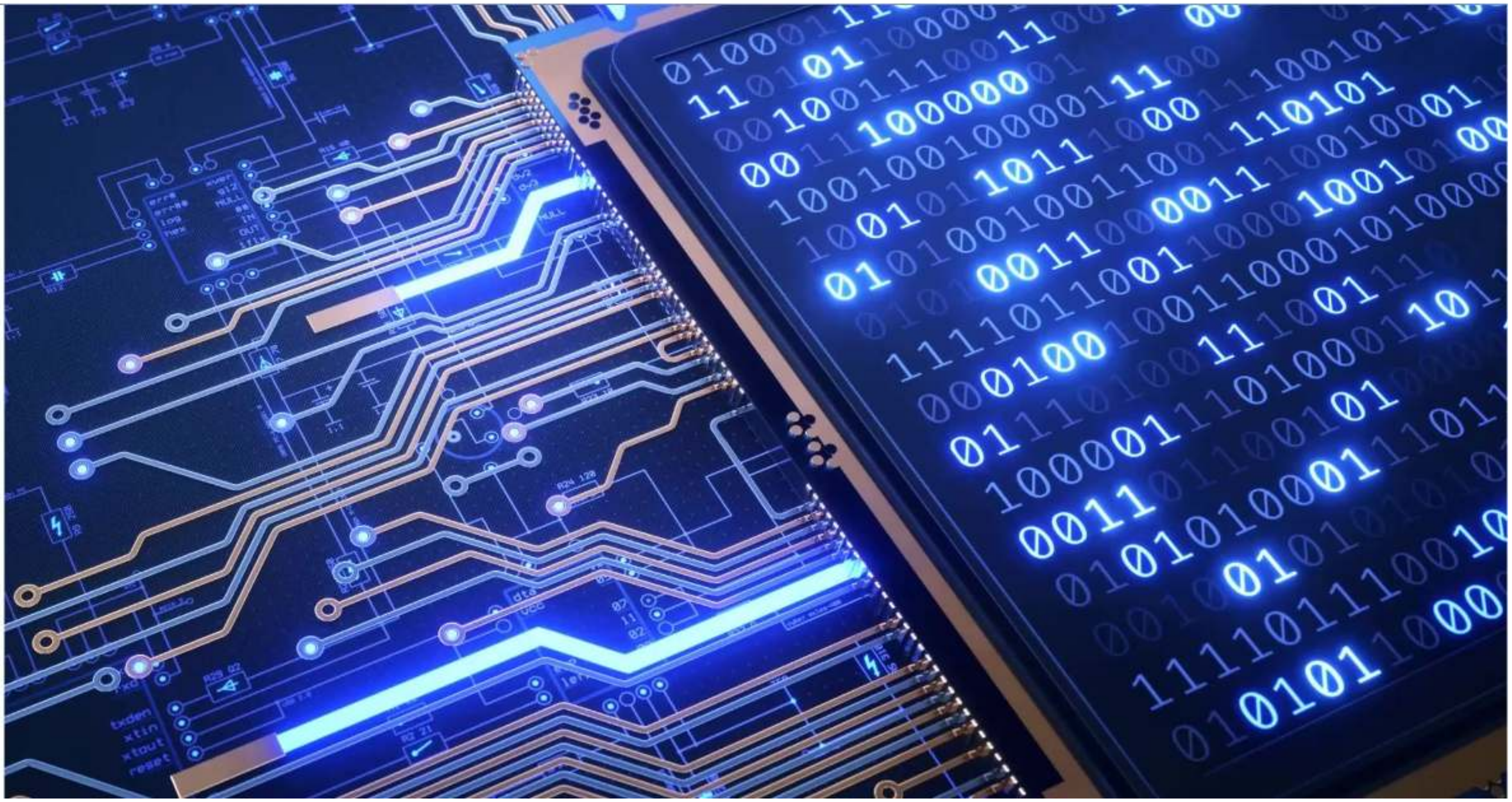
arXiv: 2101.12223

UNIVERSITY OF WATERLOO | IQC Institute for Quantum Computing

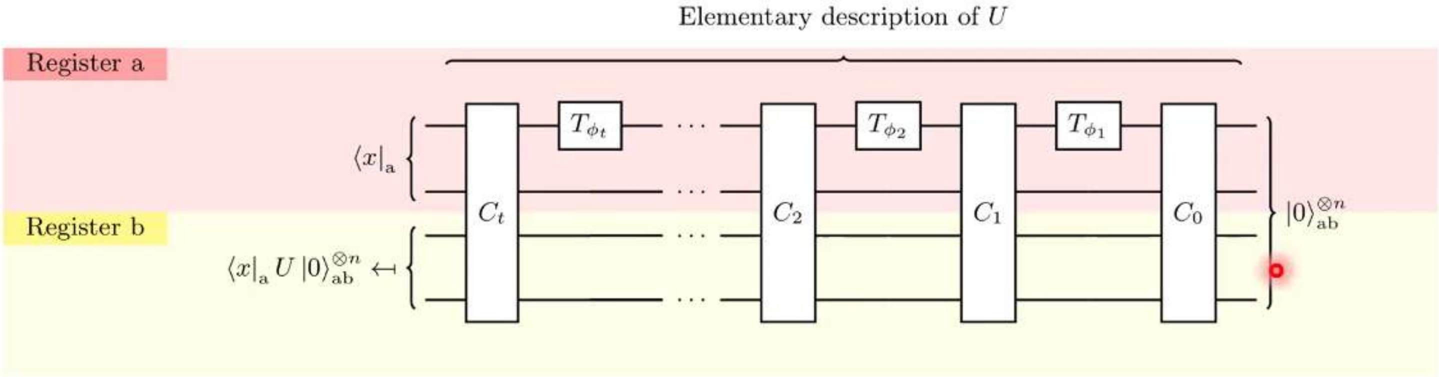
PERIMETER INSTITUTE

JAGIELLONIAN UNIVERSITY IN KRAKÓW

THE UNIVERSITY OF SYDNEY



# The task: Born rule probability estimation



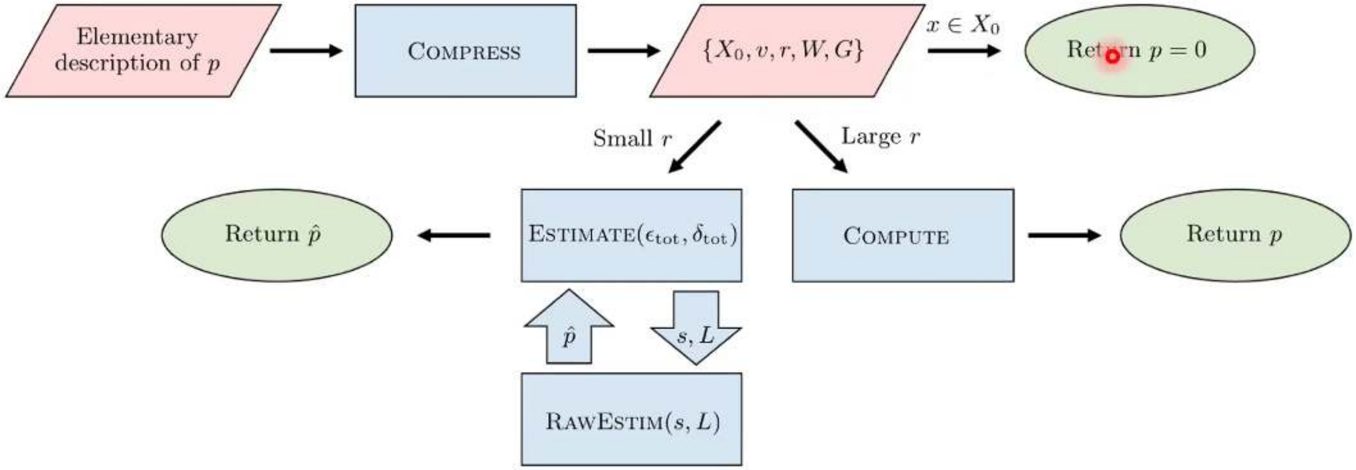
- Allowed gates: Cliffords + $T_\phi$ where $\phi \in (0, \pi/2)$ can be arbitrary.

$$T_\phi \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}$$

- For a given $x \in \{0,1\}^w$, project first $w$ qubits in computational basis state $|x\rangle$.
- Want **classical** estimate of $p := \left\| \langle x|_{\mathrm{a}} U |0\rangle_{\mathrm{ab}}^{\otimes n} \right\|_2^2$.

arXiv: 2101.12223

# Overview



Elementary description of $p$

COMPRESS

$\{X_0, v, r, W, G\}$

$x \in X_0$

Return $p = 0$

Small $r$

Large $r$

Return $\hat{p}$

ESTIMATE$(\epsilon_{\text{tot}}, \delta_{\text{tot}})$

COMPUTE

Return $p$

$\hat{p}$

$s, L$

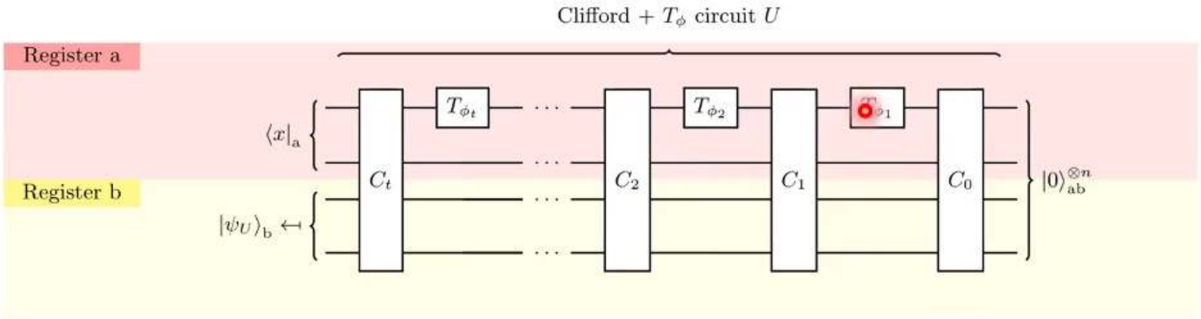
RAWESTIM$(s, L)$

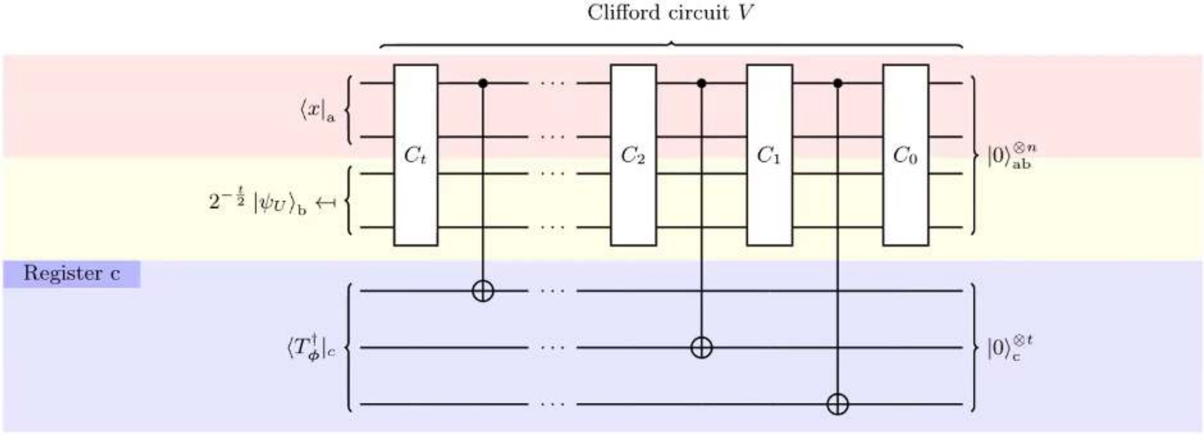
Code for algorithms availible at: https://github.com/or1426/Clifford-T-estimator and coming soon to `Qiskit`

1. COMPRESS algorithm
2. COMPUTE algorithm
3. RAWESTIM algorithm
4. ESTIMATE algorithm

arXiv: 2101.12223

# COMPRESS algorithm: Gadgetize



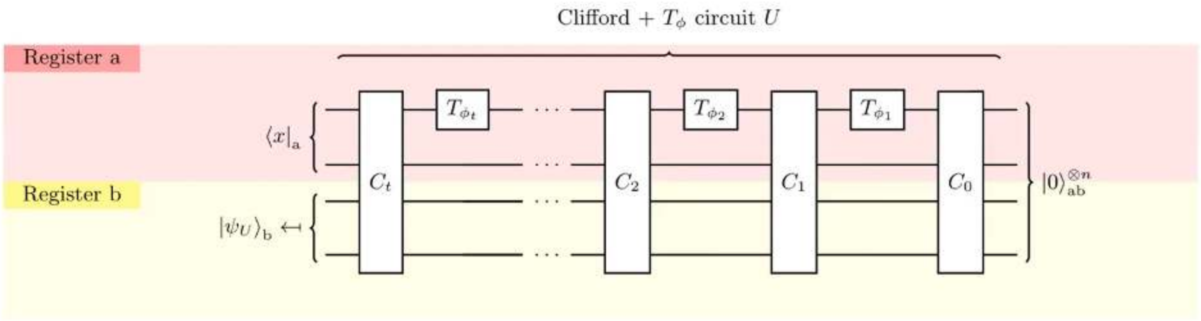


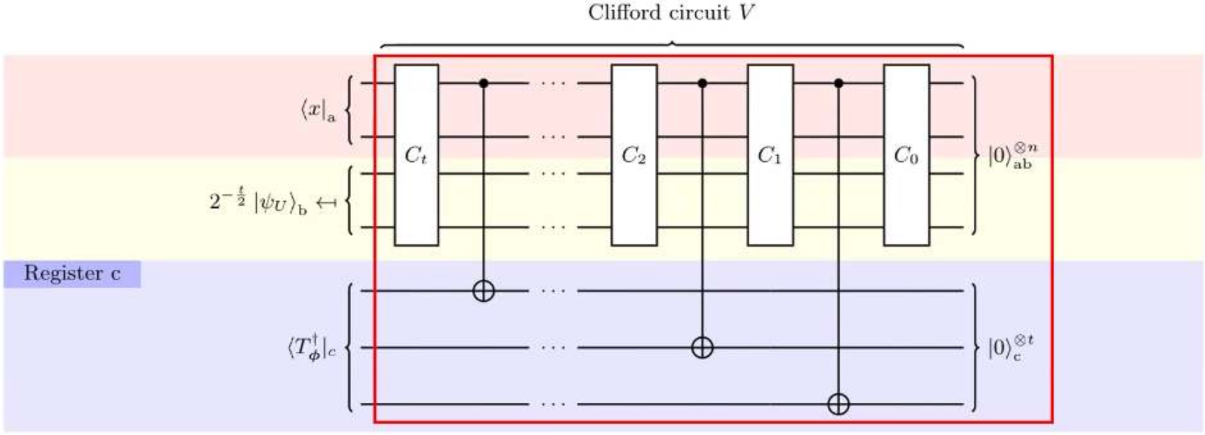
$$p = \| |\psi_U\rangle \|_2^2$$
$$= 2^t \mathrm{Tr}\left(\Pi_{G''} |x\rangle\langle x|_{\mathrm{a}} |T^\dagger_{\boldsymbol{\phi}}\rangle\langle T^\dagger_{\boldsymbol{\phi}}|_{\mathrm{c}}\right)$$

$$|T^\dagger_{\boldsymbol{\phi}}\rangle := |T^\dagger_{\phi_1}\rangle \otimes \ldots \otimes |T^\dagger_{\phi_t}\rangle, \qquad |T^\dagger_{\phi}\rangle := \frac{1}{\sqrt{2}}(|0\rangle + \exp(-i\phi)|1\rangle)$$

arXiv: 2101.12223

# COMPRESS algorithm: Gadgetize



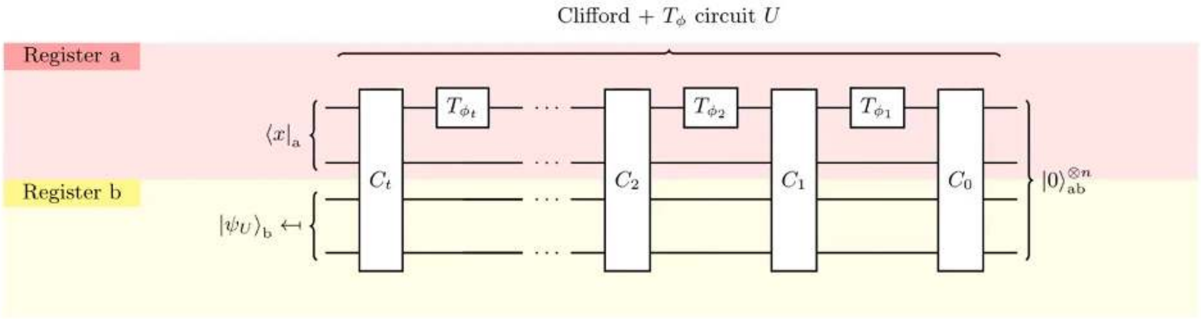


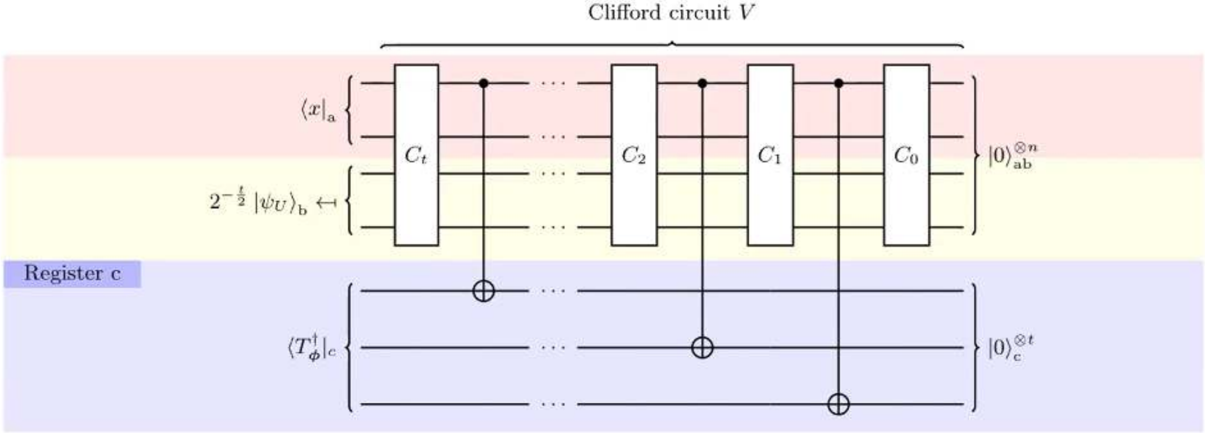
$$p = \| |\psi_U\rangle \|_2^2$$
$$= 2^t \mathrm{Tr}\left(\underline{\Pi_{G''}}\, |x\rangle\langle x|_{\mathrm{a}}\, |T^\dagger_{\boldsymbol{\phi}}\rangle\langle T^\dagger_{\boldsymbol{\phi}}|_{\mathrm{c}}\right)$$

$$|T^\dagger_{\boldsymbol{\phi}}\rangle := |T^\dagger_{\phi_1}\rangle \otimes \ldots \otimes |T^\dagger_{\phi_t}\rangle, \qquad |T^\dagger_{\phi}\rangle := \frac{1}{\sqrt{2}}\left(|0\rangle + \exp(-i\phi)\,|1\rangle\right)$$

arXiv: 2101.12223

# COMPRESS algorithm: Constrain



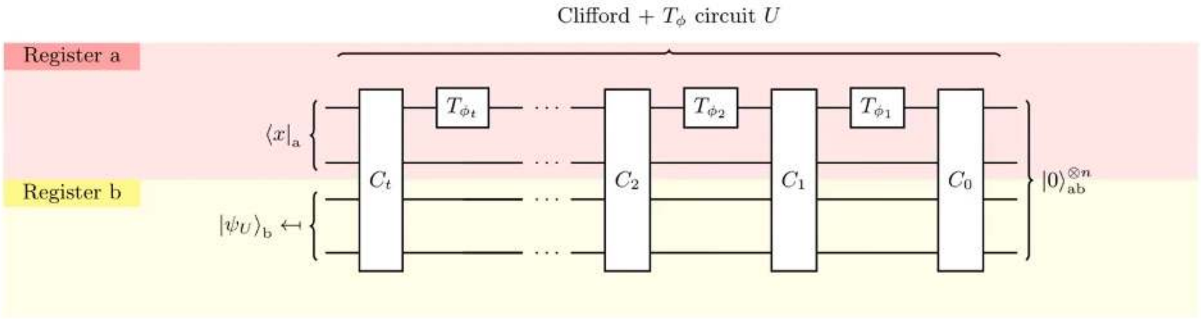


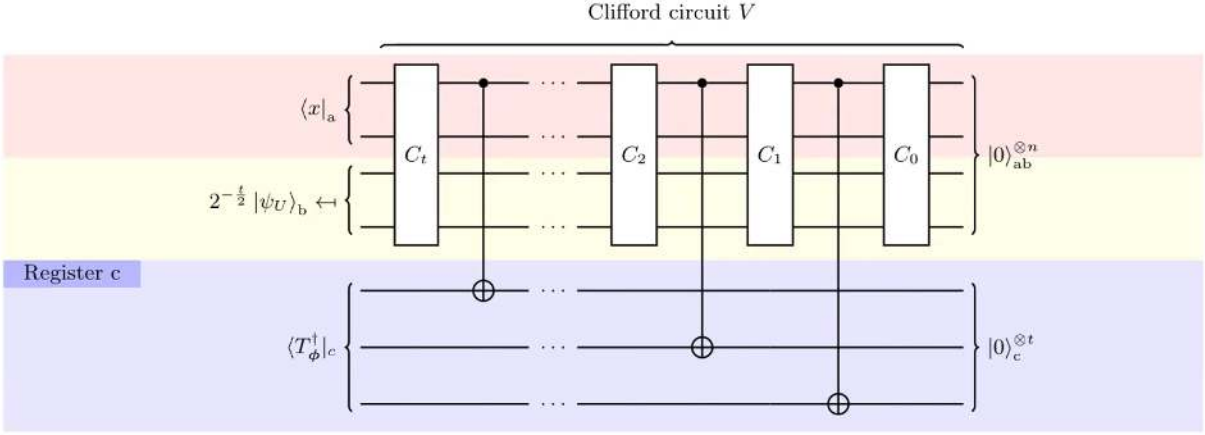
$$
\begin{aligned}
p &= \| |\psi_U\rangle \|_2^2 \\
&= 2^t \mathrm{Tr}\left(\Pi_{G''} |x\rangle\langle x|_a \, |T_\phi^\dagger\rangle\langle T_\phi^\dagger|_c\right) \\
&= 2^{t-r+v-w} \mathrm{Tr}\left(\underline{\Pi_G} |T_\phi^\dagger\rangle\langle T_\phi^\dagger|\right)
\end{aligned}
$$

$$
|T_{\boldsymbol{\phi}}^\dagger\rangle := |T_{\phi_1}^\dagger\rangle \otimes \ldots \otimes |T_{\phi_t}^\dagger\rangle, \qquad |T_\phi^\dagger\rangle := \frac{1}{\sqrt{2}}(|0\rangle + \exp(-i\phi)|1\rangle)
$$

arXiv: 2101.12223

# COMPRESS algorithm: Construct


Clifford + $T_\phi$ circuit $U$


Clifford circuit $V$

$$
\begin{aligned}
p &= \big\| |\psi_U\rangle \big\|_2^2 \\
&= 2^t \mathrm{Tr}\left(\Pi_{G''}\, |x\rangle\langle x|_{\mathrm{a}}\, |T^\dagger_{\boldsymbol{\phi}}\rangle\langle T^\dagger_{\boldsymbol{\phi}}|_{\mathrm{c}}\right) \\
&= 2^{t-r+v-w}\mathrm{Tr}\left(\Pi_G |T^\dagger_{\boldsymbol{\phi}}\rangle\langle T^\dagger_{\boldsymbol{\phi}}|\right) \\
&= 2^{t-r+v-w}\left\| \langle 0|^{\otimes t-r}\, W |T^\dagger_{\boldsymbol{\phi}}\rangle \right\|_2^2
\end{aligned}
$$

$$
|T^\dagger_{\boldsymbol{\phi}}\rangle := |T^\dagger_{\phi_1}\rangle \otimes \ldots \otimes |T^\dagger_{\phi_t}\rangle, \qquad |T^\dagger_{\phi}\rangle := \frac{1}{\sqrt{2}}\left(|0\rangle + \exp(-i\phi)\,|1\rangle\right)
$$

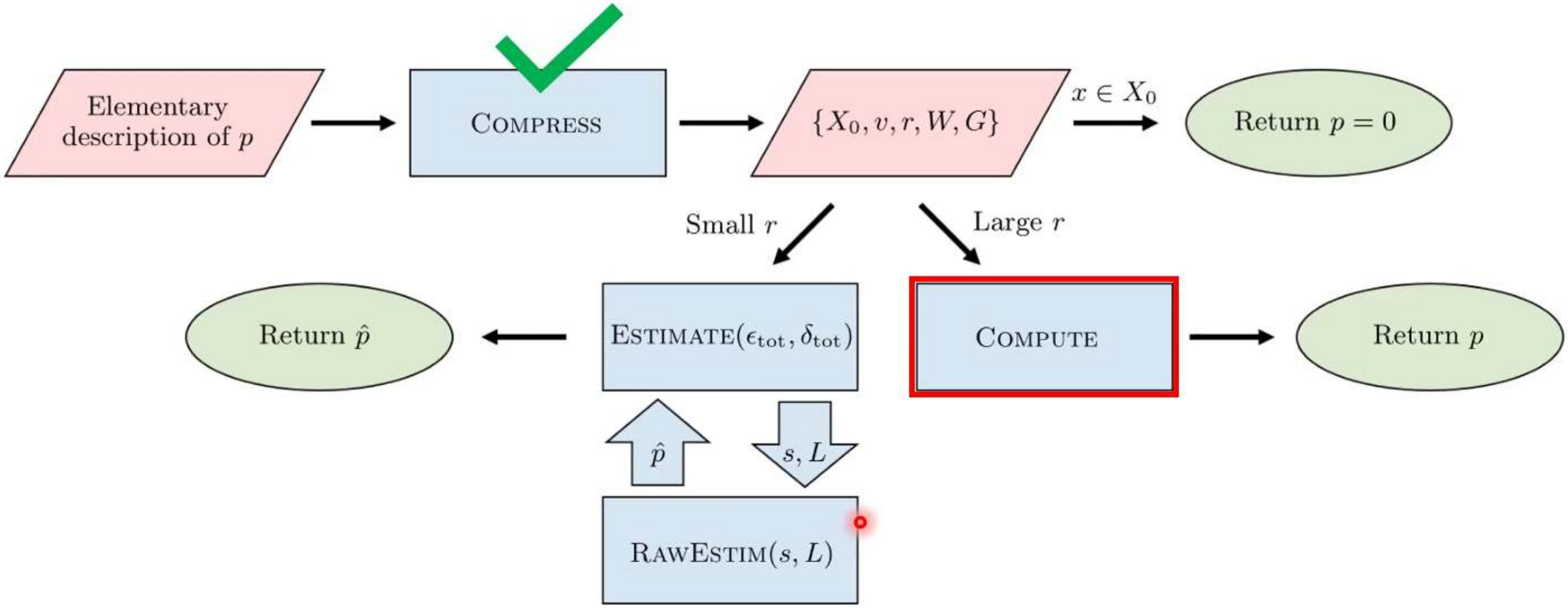
arXiv: 2101.12223

# COMPRESS algorithm

Given an 'elementary description of $p$', COMPRESS **efficiently** outputs:

- The locations of some deterministic outcome qubits: $\{q_1, \ldots, q_v\} \subseteq [w]$ and specifes the outcome
- Stabilizer generating set $G \in \mathcal{G}(t, t-r)$ $\qquad p = 2^{t-r+v-w}\mathrm{Tr}\left(\Pi_G |T_{\boldsymbol{\phi}}^\dagger\rangle\langle T_{\boldsymbol{\phi}}^\dagger|\right)$
- $r \in \{0, 1, \ldots, \min\{t, n-w\}\}$
- Elementary description of Clifford circuit $W$ $\qquad p = 2^{t-r+v-w}\left\|\langle 0|^{\otimes t-r} W |T_{\boldsymbol{\phi}}^\dagger\rangle\right\|_2^2$

# Checkpoint!



Elementary description of $p$

COMPRESS

$\{X_0, v, r, W, G\}$

$x \in X_0$

Return $p = 0$

Small $r$

Large $r$

Return $\hat{p}$

ESTIMATE($\epsilon_{\text{tot}}, \delta_{\text{tot}}$)

COMPUTE

Return $p$

$\hat{p}$

$s$, $L$

RAWESTIM($s$, $L$)

# COMPUTE algorithm

- Exactly computes $p$ in time $\tau_{\textsc{Compute}} = O(2^{t-r}(t-r)t)$.
- Using $G = \{g_1, \ldots, g_{t-r}\}$ from COMPRESS:

$$p = 2^{t-r+v-w}\mathrm{Tr}\left(\Pi_G |T_{\phi}^{\dagger}\rangle\langle T_{\phi}^{\dagger}|\right) = 2^{v-w}\langle T_{\phi}^{\dagger}|\prod_{i=1}^{t-r}(I+g_i)|T_{\phi}^{\dagger}\rangle$$

- $r \in \{0, 1, \ldots, r_{\max}\}$ where $r_{\max} = \min\{t, n-w\}$.
- $r$ concentrates near $r_{\max}$ for high Clifford count random circuits.

$n = 55$
$w = 5$
$t = 80$
$c = 10^5$
runtime $\leq 2$ hrs

arXiv: 2101.12223

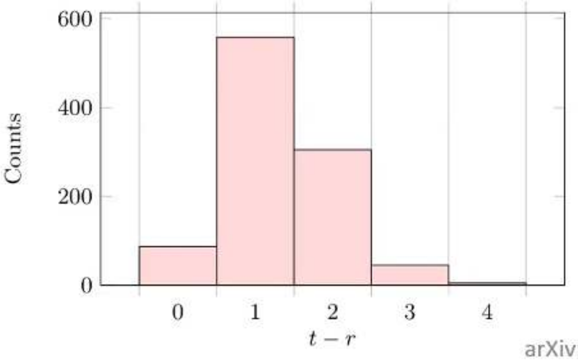
# COMPUTE algorithm

- Exactly computes $p$ in time $\tau_{\textsc{Compute}} = O(2^{t-r}(t-r)t)$.
- Using $G = \{g_1, \ldots, g_{t-r}\}$ from COMPRESS:

$$p = 2^{t-r+v-w}\mathrm{Tr}\left(\Pi_G |T_{\boldsymbol{\phi}}^{\dagger}\rangle\langle T_{\boldsymbol{\phi}}^{\dagger}|\right) = 2^{v-w}\langle T_{\boldsymbol{\phi}}^{\dagger}| \prod_{i=1}^{t-r}(I+g_i)|T_{\boldsymbol{\phi}}^{\dagger}\rangle$$

- $r \in \{0, 1, \ldots, r_{\max}\}$ where $r_{\max} = \min\{t, n-w\}$.
- $r$ concentrates near $r_{\max}$ for high Clifford count random circuits.

$10^3$ RCs
$n = 100$
$w = 20$
$t = 80$
$c = 10^5$



arXiv: 2101.12223

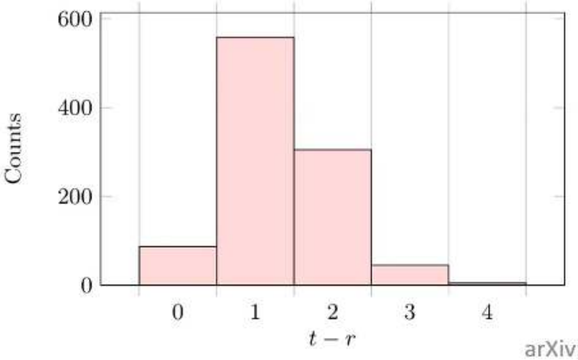
# COMPUTE algorithm

- Exactly computes $p$ in time $\tau_{\text{COMPUTE}} = O(2^{t-r}(t-r)t)$.
- Using $G = \{g_1, \ldots, g_{t-r}\}$ from COMPRESS:

$$p = 2^{t-r+v-w}\mathrm{Tr}\left(\Pi_G |T_{\boldsymbol{\phi}}^\dagger\rangle\langle T_{\boldsymbol{\phi}}^\dagger|\right) = 2^{v-w}\langle T_{\boldsymbol{\phi}}^\dagger|\prod_{i=1}^{t-r}(I+g_i)|T_{\boldsymbol{\phi}}^\dagger\rangle$$

- $r \in \{0, 1, \ldots, r_{\max}\}$ where $r_{\max} = \min\{t, n-w\}$.
- $r$ concentrates near $r_{\max}$ for high Clifford count random circuits.

$10^3$ RCs
$n = 100$
$w = 20$
$t = 80$
$c = 10^5$

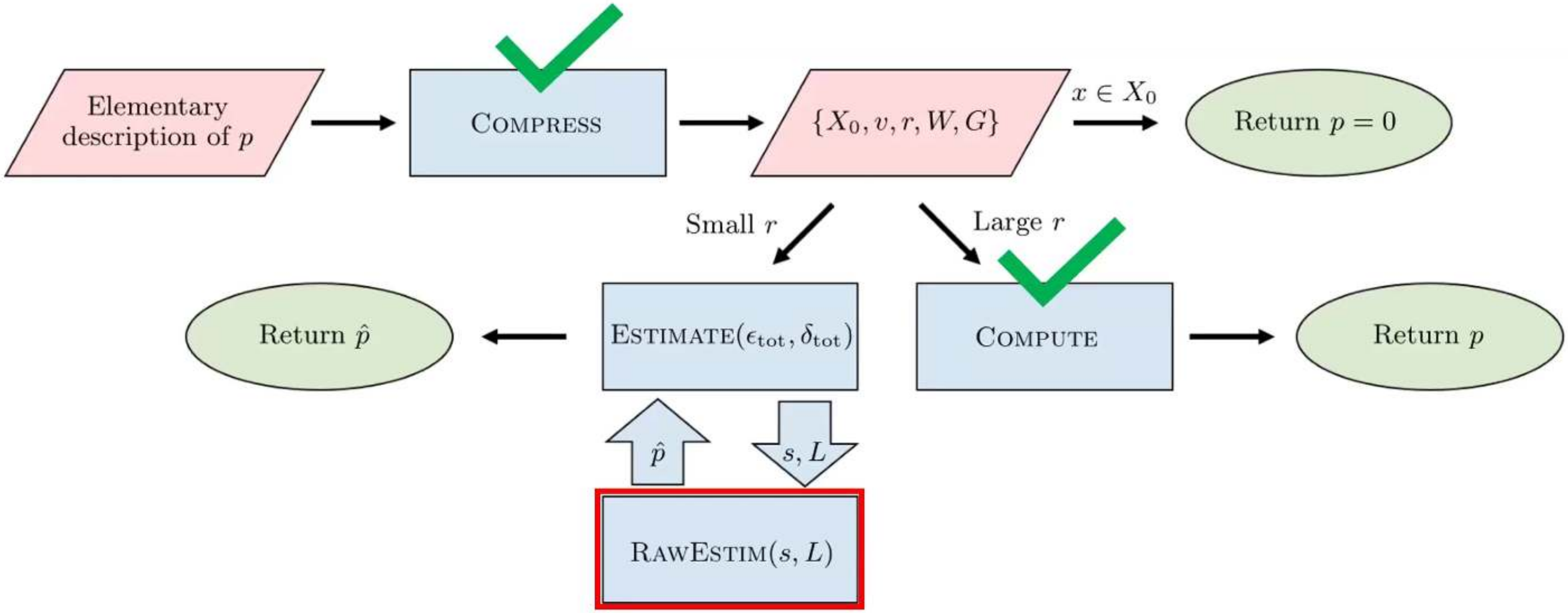


**Note:** This says 'typically' $r \approx r_{\max}$.
But 'typically' $p \approx 2^{-w}$.
When do we get practical simulation for 'non-uniform' $p$?

**Open:** Intersting applications for COMPUTE?

arXiv: 2101.12223

# Checkpoint!


Elementary description of $p$

COMPRESS

$\{X_0, v, r, W, G\}$

$x \in X_0$

Return $p = 0$

Small $r$

Large $r$

Return $\hat{p}$

ESTIMATE$(\epsilon_{\text{tot}}, \delta_{\text{tot}})$

COMPUTE

Return $p$

$\hat{p}$

$s, L$

RAWESTIM$(s, L)$

arXiv: 2101.12223

# RawEstim algorithm: Concentration

$$p = \left\| 2^{\frac{t-r+v-w}{2}} \langle 0|^{\otimes t-r} W |T_{\phi}^{\dagger}\rangle \right\|_2^2 = \| |\mu\rangle \|_2^2$$

- For $y \in \{0,1\}^t$, we define unnormalized stabilizer states $|\psi(y)\rangle$ and product probability distribution $q(y)$ s.t.

$$\sum_y q(y) \, |\psi(y)\rangle = |\mu\rangle \, .$$

- Concentration: The $s$-sample average $|\overline{\psi}\rangle = \frac{1}{s}\sum_{i=1}^{s} |\psi_i\rangle$ concentrates around $|\mu\rangle$:

$$\Pr\left( \left\| |\overline{\psi}\rangle\langle\overline{\psi}| - |\mu\rangle\langle\mu| \right\|_1 \geq \epsilon \right) \leq 2e^2 \exp\left( \frac{-s(\sqrt{p+\epsilon} - \sqrt{p})^2}{2(\sqrt{m} + \sqrt{p})^2} \right),$$

  where $p = \| |\mu\rangle \|_2^2$ and $m \geq \| |\psi(y)\rangle \|_2^2$.

- We show that $\xi^* \geq \| |\psi(y)\rangle \|_2^2$.

arXiv: 2101.12223

# RawEstim algorithm: Concentration

$$p = \left\| 2^{\frac{t-r+v-w}{2}} \langle 0|^{\otimes t-r} W|T_{\phi}^{\dagger}\rangle \right\|_2^2 = \| |\mu\rangle \|_2^2$$

- For $y \in \{0,1\}^t$, we define unnormalized stabilizer states $|\psi(y)\rangle$ and product probability distribution $q(y)$ s.t.

$$\sum_y q(y) \, |\psi(y)\rangle = |\mu\rangle \, .$$

- Concentration: The $s$-sample average $|\overline{\psi}\rangle = \frac{1}{s}\sum_{i=1}^{s} |\psi_i\rangle$ concentrates around $|\mu\rangle$:

$$\Pr\left( \left\| |\overline{\psi}\rangle\langle\overline{\psi}| - |\mu\rangle\langle\mu| \right\|_1 \geq \epsilon \right) \leq 2e^2 \exp\left( \frac{-s(\sqrt{p+\epsilon} - \sqrt{p})^2}{2(\sqrt{m} + \sqrt{p})^2} \right),$$

  where $p = \| |\mu\rangle \|_2^2$ and $m \geq \| |\psi(y)\rangle \|_2^2$.

- We show that $\xi^* \geq \| |\psi(y)\rangle \|_2^2$.

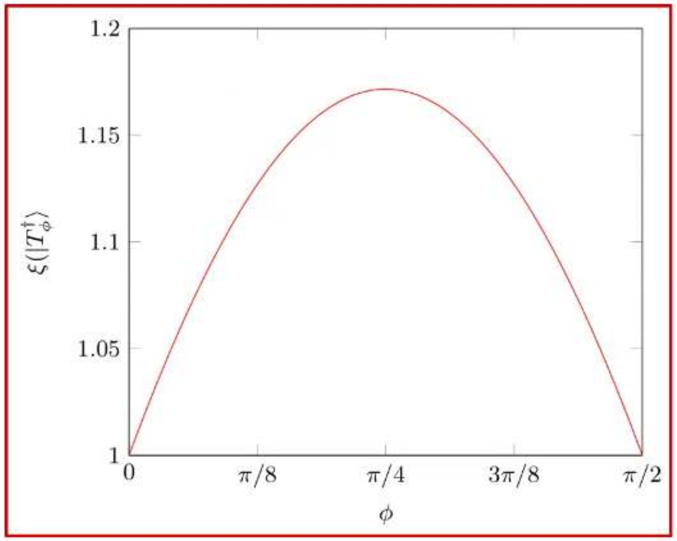
T. P. Hayes, A large-deviation inequality for vector-valued martingales, Combinatorics, Probability and Computing (2005).

# RawEstim algorithm: FNE and CH-form

- $\hat{p}$ is estimate of $\left\| |\overline{\psi}\rangle \right\|_2^2$ using $L$ iterations of the BG fast norm estimation algorithm.

- CH-form [BBCCGH] + tricks used to compute $|\psi(y)\rangle \propto \langle 0|^{\otimes t-r} W |\tilde{y}\rangle$
  - Instead of $|\tilde{y}\rangle \mapsto W|\tilde{y}\rangle$ we use $W\left|\tilde{0}^t\right\rangle \mapsto W|\tilde{y}\rangle$
  - Factorization of CH-form: $CHF(|0\rangle \otimes |\varphi\rangle) \mapsto CHF(|\varphi\rangle)$

- Stabilizer extent: $\xi^* := \xi(|T^\dagger_{\phi_1}\rangle) \times \ldots \times \xi(|T^\dagger_{\phi_t}\rangle) < 1.2^t$.



S. Bravyi and D. Gosset, Improved classical simulation of quantum circuits dominated by Clifford gates, Phys. Rev. Lett. **116**, 250501 (2016).

S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, Simulation of quantum circuits by low-rank stabilizer decompositions, Quantum **3**, 181 (2019).

arXiv: 2101.12223

# $\textsc{RawEstim}$ algorithm: performance

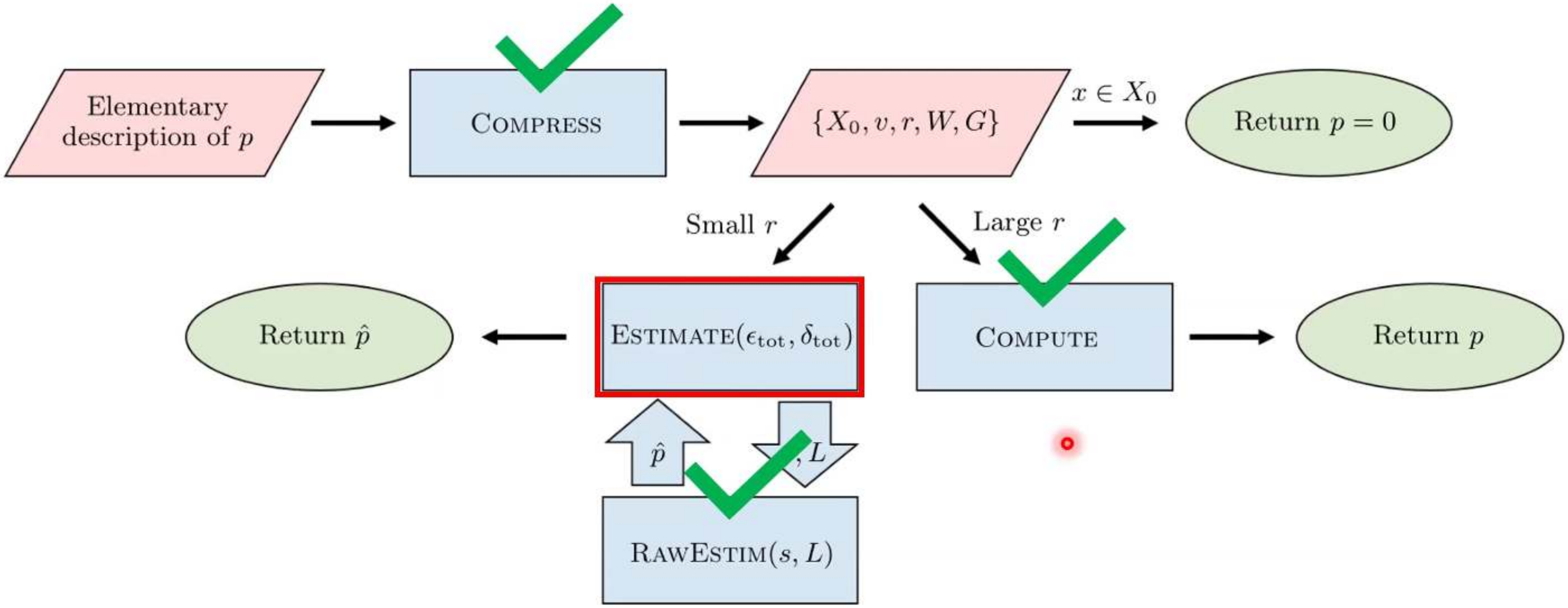
- For given $s, L \in \mathbb{N}^+$ computes $\hat{p}$ in time $\tau_{\textsc{RawEstim}} = O(st^2(t-r) + sLr^3)$ s.t. $\forall \epsilon_{\text{tot}} > 0, \epsilon \in (0, \epsilon_{\text{tot}})$:

$$\Pr\left(|\hat{p} - p| \geq \epsilon_{\text{tot}}\right) \leq 2e^2 \exp\left(-s\frac{(\sqrt{p+\epsilon} - \sqrt{p})^2}{2(\sqrt{\xi^*} + \sqrt{p})^2}\right) + \exp\left(-L\left(\frac{\epsilon_{\text{tot}} - \epsilon}{p+\epsilon}\right)^2\right)$$
$$=: \delta(p, \epsilon_{\text{tot}}, \epsilon, s, L).$$

- $\delta(p, \epsilon_{\text{tot}}, \epsilon, s, L)$ depends on unknown $p$
- Can UB $\delta(p, \epsilon_{\text{tot}}, \epsilon, s, L)$ by setting $p = 1$ but this is very conservative.

arXiv: 2101.12223

# Checkpoint!



Elementary description of $p$

COMPRESS

$\{X_0, v, r, W, G\}$

$x \in X_0$

Return $p = 0$

Small $r$

Large $r$

Return $\hat{p}$

ESTIMATE$(\epsilon_{\text{tot}}, \delta_{\text{tot}})$

COMPUTE

Return $p$

$\hat{p}$

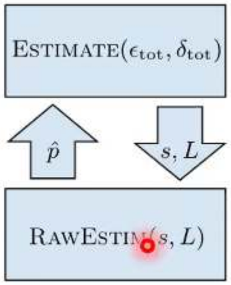
$, L$

RAWESTIM$(s, L)$

arXiv: 2101.12223

# ESTIMATE algorithm

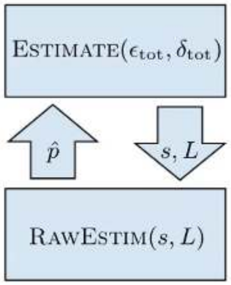
- ESTIMATE purpose: $\Pr\left(|\hat{p}-p| \geq \epsilon_{\mathrm{tot}}\right) \leq \delta_{\mathrm{tot}}$
- ESTIMATE will use RAWESTIM as a sub-routine



- RAWESTIM gives: $\Pr\left(|\hat{p}-p| \geq \epsilon_{\mathrm{tot}}\right) \leq \delta(p, \epsilon_{\mathrm{tot}}, \epsilon, s, L)$
- Defn: $\epsilon^{*}(p, \delta_{\mathrm{tot}}, \mathcal{T})$

# ESTIMATE algorithm

- ESTIMATE purpose: $\Pr\left(|\hat{p} - p| \geq \epsilon_{\text{tot}}\right) \leq \delta_{\text{tot}}$
- ESTIMATE will use RAWESTIM as a sub-routine



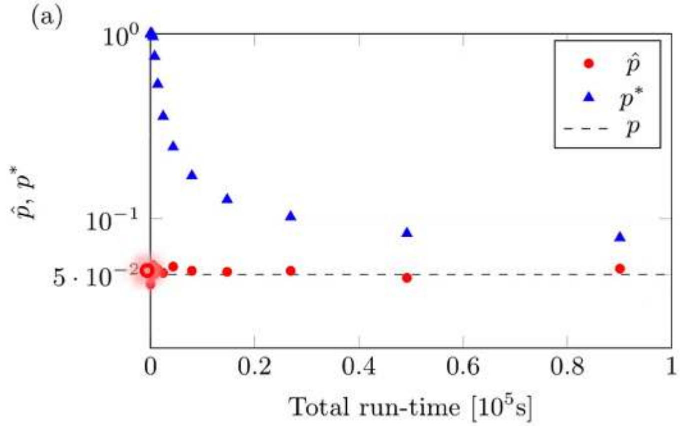
- RAWESTIM gives: $\Pr\left(|\hat{p} - p| \geq \epsilon_{\text{tot}}\right) \leq \delta(p, \epsilon_{\text{tot}}, \epsilon, s, L)$
- Defn: $\epsilon^*(p, \delta_{\text{tot}}, \mathcal{T})$ is the minimal achievable additive error s.t. $\tau_{\textsc{RawEstim}}(s, L) \leq \mathcal{T}$ and $\text{FP} \leq \delta_{\text{tot}}$
- $\Pr\left(|\hat{p} - p| \geq \epsilon^*(p, \delta_{\text{tot}}, \mathcal{T})\right) \leq \delta_{\text{tot}}$
- $\epsilon^*(p, \delta_{\text{tot}}, \mathcal{T})$ is monotone increasing in $p$

arXiv: 2101.12223

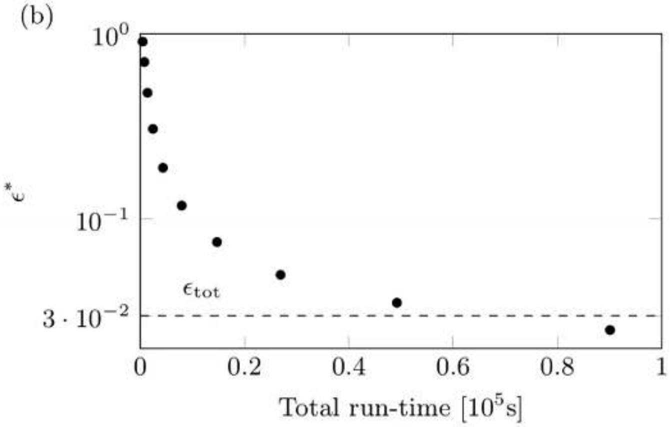
# ESTIMATE algorithm

- Idea: for $k = 1, 2, \ldots$, compute $\epsilon^*(p, \delta_{\text{tot}}, 2^k \mathcal{T}_0)$ until we find $\epsilon^*(p, \delta_{\text{tot}}, 2^k \mathcal{T}_0) \leq \epsilon_{\text{tot}}$
- Idea: for $k = 1, 2, \ldots$, compute $\epsilon^*_k := \epsilon^*(p^*_{k-1}, \delta_k, 2^k \mathcal{T}_0)$ until we find $\epsilon^*(p^*, \delta_k, 2^k \mathcal{T}_0) \leq \epsilon_{\text{tot}}$
- $p^*_0 = 1$, $p^*_k = \hat{p}_k + \epsilon^*_k$ and
- $\delta_k := \frac{6}{\pi^2 k^2} \delta_{\text{tot}}$ ensures $\sum_k \delta_k = \delta_{\text{tot}}$

(a)



(b)

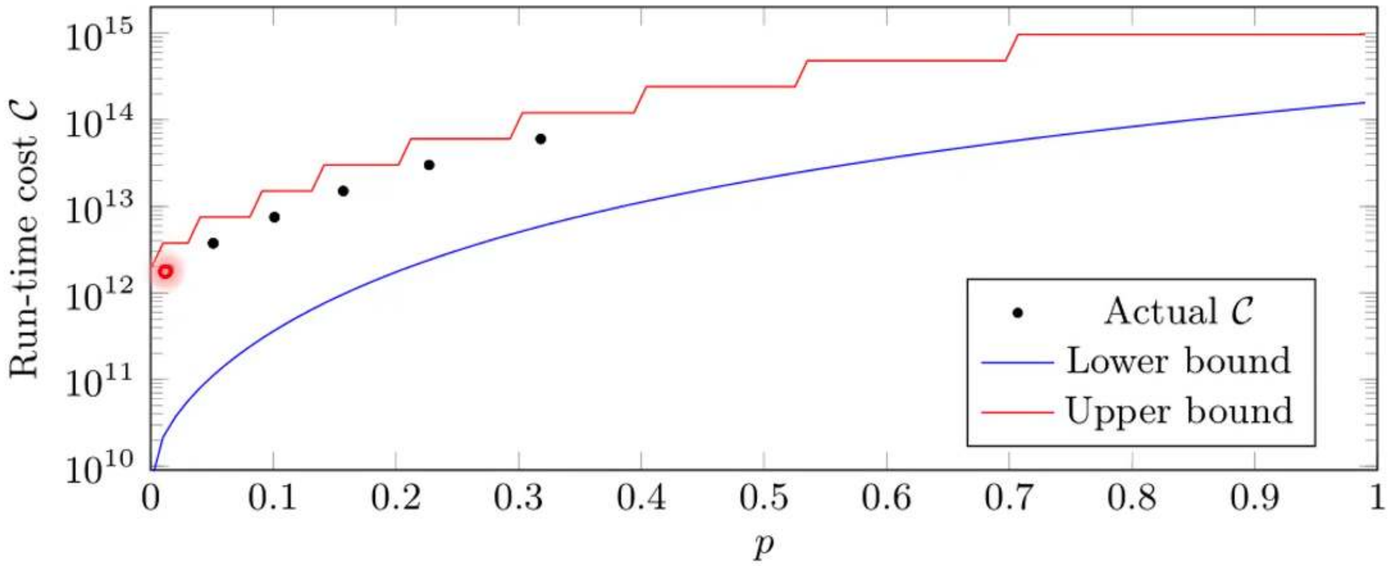


$n = 50$
$w = 8$
$t = 52 + 8$
$r = 10$
$c = 2000$
$\xi^* = \xi(|T\rangle^{52})$
$\delta_{\text{tot}} = 10^{-3}$

arXiv: 2101.12223

# ESTIMATE algorithm

- ESTIMATE gives: $\Pr\left(|\hat{p} - p| \geq \epsilon_{\text{tot}}\right) \leq \delta_{\text{tot}}$
- Runtime depends on $p$ and is probabilistic.
- RUNTIME gives a probabilistic upper bound of $\tau_{\text{ESTIMATE}}$ for any possible $p$.

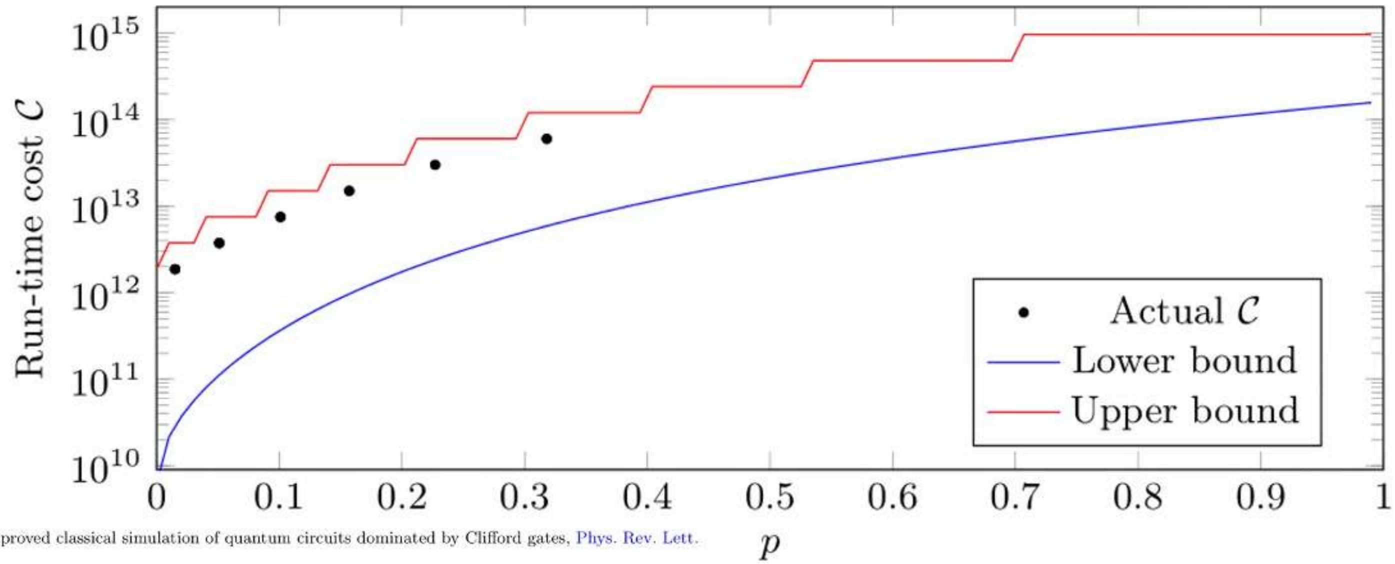


$n = 40$
$w = 8$
$t = 32 + 8$
$r = 8$
$c = 2000$
$\xi^* = \xi(|T\rangle^{32})$
$\epsilon_{\text{tot}} = 0.05$
$\delta_{\text{tot}} = 10^{-3}$
$\delta_{\text{UB}} = 0.05$

arXiv: 2101.12223

# ESTIMATE algorithm

- ESTIMATE gives: $\Pr\left(|\hat{p} - p| \geq \epsilon_{\text{tot}}\right) \leq \delta_{\text{tot}}$
- Runtime depends on $p$ and is probabilistic.
- RUNTIME gives a probabilistic upper bound of $\tau_{\text{ESTIMATE}}$ for any possible $p$.



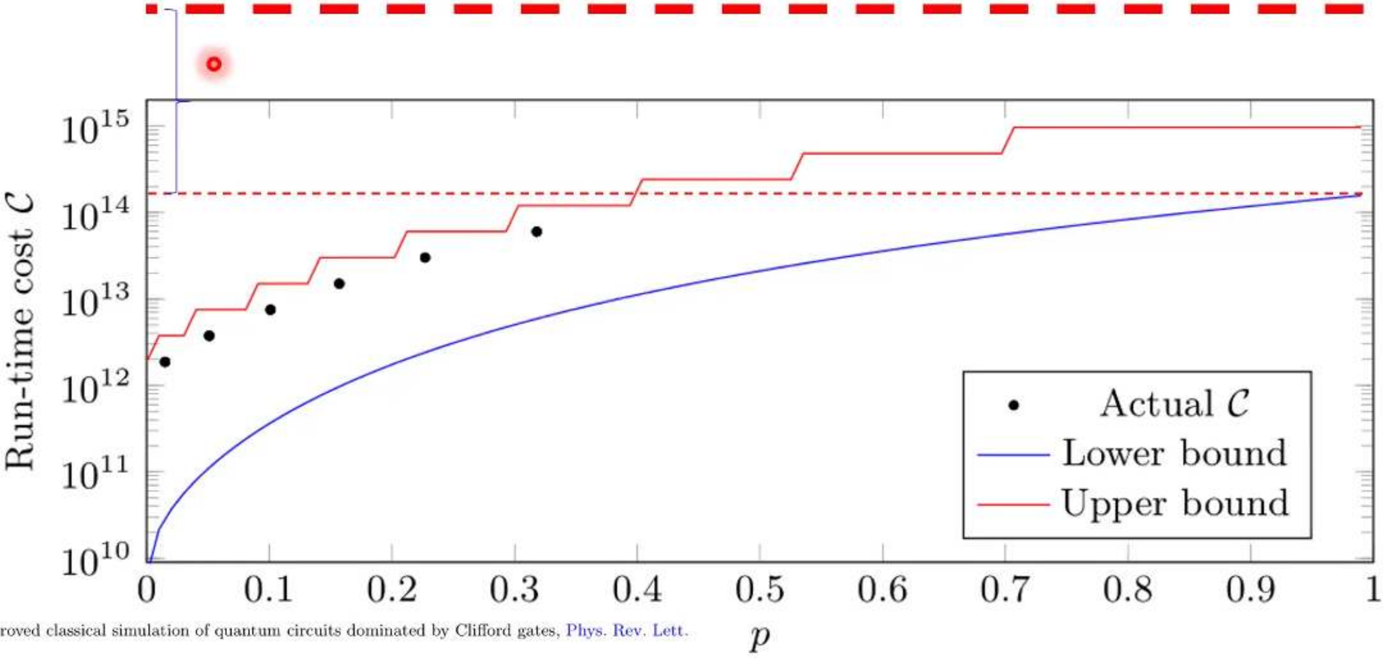
$n = 40$
$w = 8$
$t = 32 + 8$
$r = 8$
$c = 2000$
$\xi^* = \xi(|T\rangle^{32})$
$\epsilon_{\text{tot}} = 0.05$
$\delta_{\text{tot}} = 10^{-3}$
$\delta_{\text{UB}} = 0.05$

[1] S. Bravyi and D. Gosset, Improved classical simulation of quantum circuits dominated by Clifford gates, Phys. Rev. Lett. **116**, 250501 (2016).
[2] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, Simulation of quantum circuits by low-rank stabilizer decompositions, Quantum **3**, 181 (2019).

arXiv: 2101.12223

# ESTIMATE algorithm

- ESTIMATE gives: $\Pr\left(|\hat{p} - p| \geq \epsilon_{\text{tot}}\right) \leq \delta_{\text{tot}}$
- Runtime depends on $p$ and is probabilistic.
- RUNTIME gives a probabilistic upper bound of $\tau_{\text{ESTIMATE}}$ for any possible $p$.



$n = 40$
$w = 8$
$t = 32 + 8$
$r = 8$
$c = 2000$
$\xi^* = \xi(|T\rangle^{32})$
$\epsilon_{\text{tot}} = 0.05$
$\delta_{\text{tot}} = 10^{-3}$
$\delta_{\text{UB}} = 0.05$

[1] S. Bravyi and D. Gosset, Improved classical simulation of quantum circuits dominated by Clifford gates, Phys. Rev. Lett. **116**, 250501 (2016).
[2] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, Simulation of quantum circuits by low-rank stabilizer decompositions, Quantum **3**, 181 (2019).

arXiv: 2101.12223

# Outlook

- Can we predict properties of COMPRESS output?
- Practical applications for COMPUTE?
- Can we apply some of these techniqes to improve sampling algorithms?
- Can we adapt ESTIMATE to the mixed state formalism?

# Outlook

- Can we predict properties of COMPRESS output?
- Practical applications for COMPUTE?
- Can we apply some of these techniqes to improve sampling algorithms?
- Can we adapt ESTIMATE to the mixed state formalism?

## Thank you!

arXiv: 2101.12223