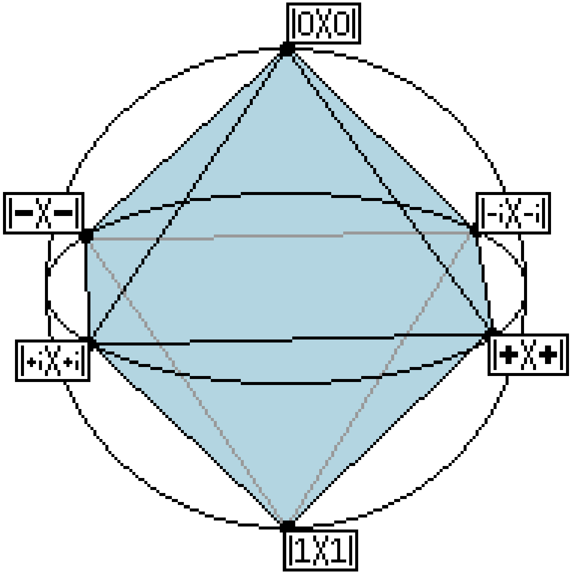
Simulating Quantum Circuits by Shuffling Paulis

Patrick Rall

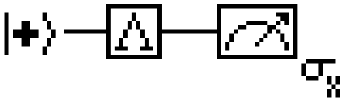
Aug 21, 2018

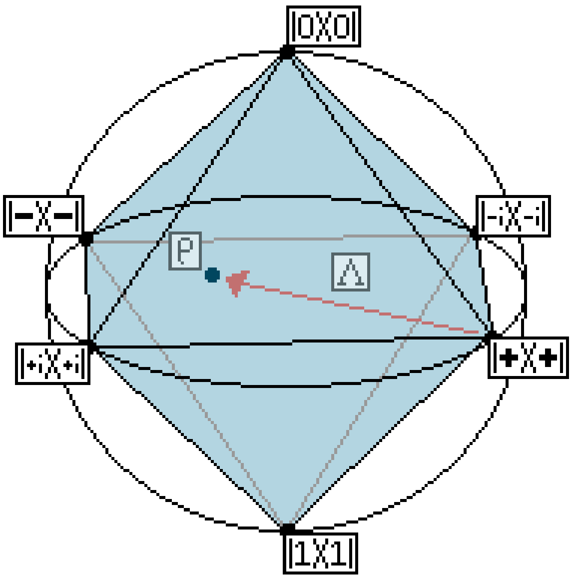
The University of Texas at Austin
Quantum Information Center

Shuffling Stabilizer States

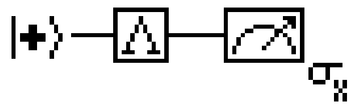


- Bennink et al. Phys. Rev. A 95, 062337
- A simple circuit:





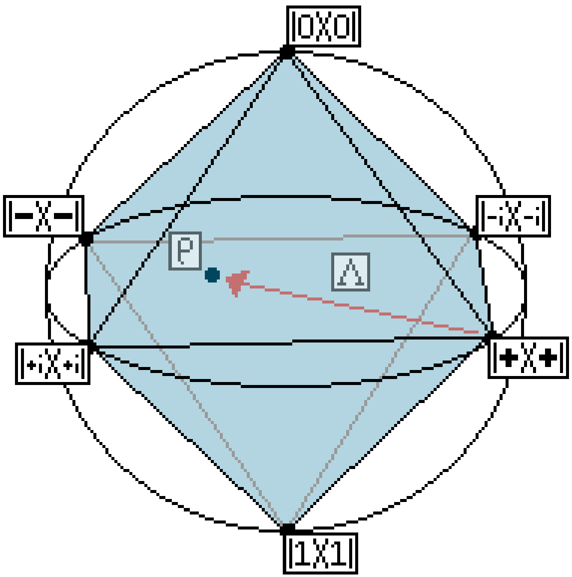
- Bennink et al. Phys. Rev. A 95, 062337
- A simple circuit:



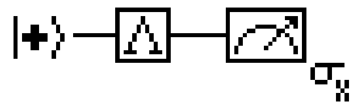
$$\Lambda(|+\rangle\langle+|) = \rho$$

$$= 0.3\,|+\rangle\langle+| + 0.7\,|-\rangle\langle-|$$

Shuffling Stabilizer States



- Bennink et al. Phys. Rev. A 95, 062337
- A simple circuit:



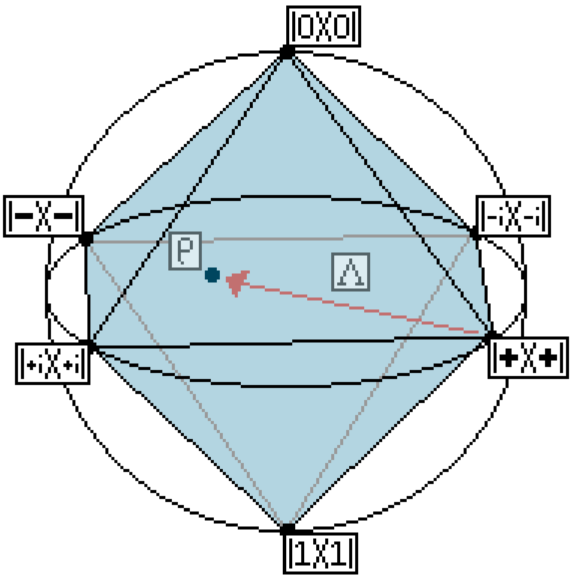
$$\Lambda(|+\rangle\langle+|) = \rho$$

$$= 0.3\,|+\rangle\langle+| + 0.7\,|-\rangle\langle-|$$

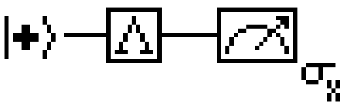
- Sample:

$$\hat{\rho} = \begin{cases} |+\rangle\langle+| & \text{with prob. } 0.3 \\ |-\rangle\langle-| & \text{with prob. } 0.7 \end{cases}$$

Shuffling Stabilizer States



- Bennink et al. Phys. Rev. A 95, 062337
- A simple circuit:



$$\Lambda(|+\rangle\langle+|) = \rho$$

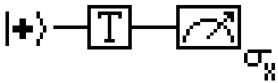
$$= 0.3\,|+\rangle\langle+| + 0.7\,|-\rangle\langle-|$$

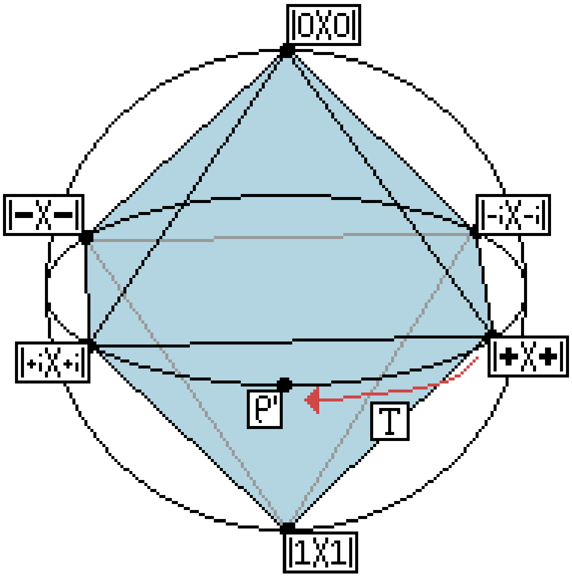
- Sample:

$$\hat{\rho} = \begin{cases} |+\rangle\langle+| & \text{with prob. } 0.3 \\ |-\rangle\langle-| & \text{with prob. } 0.7 \end{cases}$$

- $\mathrm{Tr}\left(\hat{\rho}\sigma_X\right)$ estimates $\mathrm{Tr}\left(\rho\sigma_X\right)$

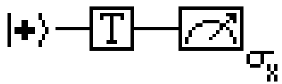
Shuffling Stabilizer States (cont.)

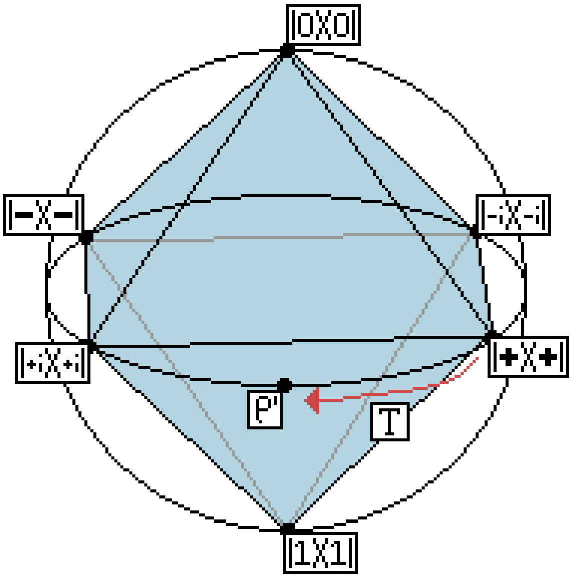




- What if output is not a stabilizer mixture, e.g. $T\,|+\rangle\langle+|\,T^\dagger = \rho'$?

Shuffling Stabilizer States (cont.)



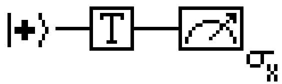


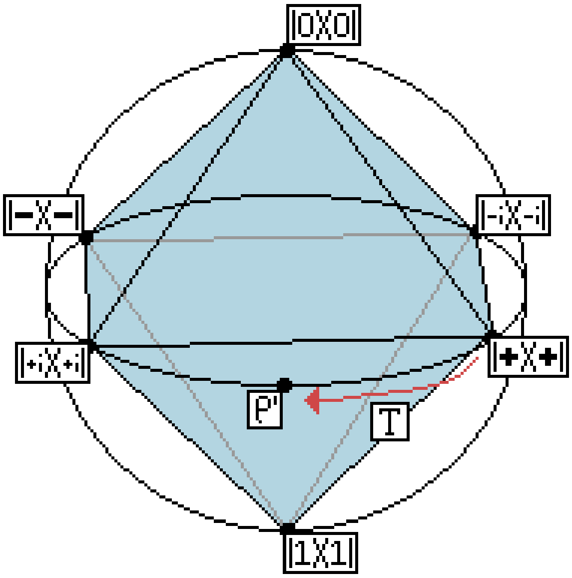
- What if output is not a stabilizer mixture, e.g. $T\,|+\rangle\langle+|\,T^\dagger = \rho'$?
- Can *still* write:

$$\rho' = \frac{\sqrt{2}+2}{4\sqrt{2}}\,|+\rangle\langle+| + \frac{\sqrt{2}+2}{4\sqrt{2}}\,|+i\rangle\langle+i|$$

$$+\frac{\sqrt{2}-2}{4\sqrt{2}}\,|-\rangle\langle-| + \frac{\sqrt{2}-2}{4\sqrt{2}}\,|-i\rangle\langle-i|$$

Shuffling Stabilizer States (cont.)





- What if output is not a stabilizer mixture, e.g. $T\left|+\right\rangle\left\langle+\right| T^\dagger = \rho'$?
- Can *still* write:

$$\rho' = \frac{\sqrt{2}+2}{4\sqrt{2}}\left|+\right\rangle\left\langle+\right| + \frac{\sqrt{2}+2}{4\sqrt{2}}\left|+i\right\rangle\left\langle+i\right|$$

$$+\frac{\sqrt{2}-2}{4\sqrt{2}}\left|-\right\rangle\left\langle-\right| + \frac{\sqrt{2}-2}{4\sqrt{2}}\left|-i\right\rangle\left\langle-i\right|$$

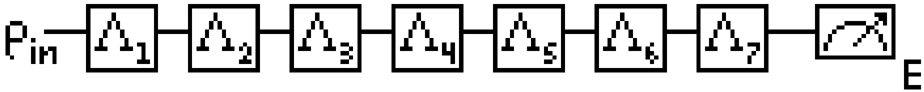
$$\mathcal{R}(\rho') = \sum_{|\phi\rangle} |q_{|\phi\rangle}| = \sqrt{2}$$

- Sample:

$$\hat{\rho} = \begin{cases} \mathcal{R}(\rho')\left|+\right\rangle\left\langle+\right| & \text{with prob. } |q_{|+\rangle}|/\mathcal{R}(\rho') \\ \mathcal{R}(\rho')\left|+i\right\rangle\left\langle+i\right| & \text{with prob. } |q_{|+i\rangle}|/\mathcal{R}(\rho') \\ \mathcal{R}(\rho')\left|-\right\rangle\left\langle-\right| & \text{with prob. } |q_{|-\rangle}|/\mathcal{R}(\rho') \\ \mathcal{R}(\rho')\left|-i\right\rangle\left\langle-i\right| & \text{with prob. } |q_{|-i\rangle}|/\mathcal{R}(\rho') \end{cases}$$

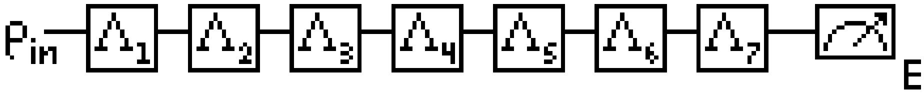
Runtime cost

- Repeated iteration:



Runtime cost

- Repeated iteration:

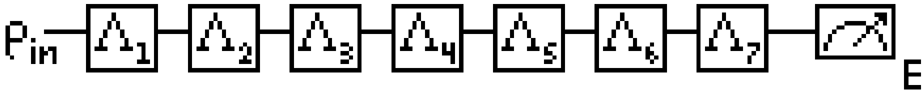


- Upper bound on sample magnitude:

$$|\text{sample}| \leq \mathcal{R}(\rho_{\text{in}}) \cdot \prod_i \max_{|\phi\rangle} \mathcal{R}\Big(\Lambda_i(|\phi\rangle\langle\phi|)\Big) \cdot \max_{|\phi\rangle} \text{Tr}\left(|\phi\rangle\langle\phi|\, E\right)$$

Runtime cost

- Repeated iteration:



- Upper bound on sample magnitude:

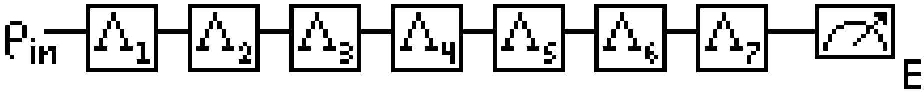
$$|\text{sample}| \leq \mathcal{R}(\rho_{\text{in}}) \cdot \prod_i \max_{|\phi\rangle} \mathcal{R}\Big(\Lambda_i(|\phi\rangle\langle\phi|)\Big) \cdot \max_{|\phi\rangle} \text{Tr}\left(|\phi\rangle\langle\phi|\, E\right)$$

- Hoeffding's inequality gives runtime: $\Pr(\text{error} \geq \varepsilon) \leq \delta$ if

$$\text{samples needed} \geq \frac{1}{\varepsilon^2} \log \frac{1}{\delta} \cdot 4 \max |\text{sample}|^2$$

Runtime cost

- Repeated iteration:



- Upper bound on sample magnitude:

$$|\text{sample}| \leq \mathcal{R}(\rho_{\text{in}}) \cdot \prod_i \max_{|\phi\rangle} \mathcal{R}\Big(\Lambda_i(|\phi\rangle\langle\phi|)\Big) \cdot \max_{|\phi\rangle} \text{Tr}\left(|\phi\rangle\langle\phi|\, E\right)$$

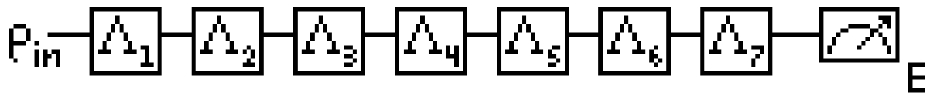
- Hoeffding's inequality gives runtime: $\Pr(\text{error} \geq \varepsilon) \leq \delta$ if

$$\text{samples needed} \geq \frac{1}{\varepsilon^2} \log \frac{1}{\delta} \cdot 4 \max |\text{sample}|^2$$

- Calculating $\mathcal{R}$ is hard. See Markus Heinrich's talk tomorrow!

Runtime cost

- Repeated iteration:



- Upper bound on sample magnitude:

$$|\text{sample}| \leq \mathcal{R}(\rho_{\text{in}}) \cdot \prod_i \max_{|\phi\rangle} \mathcal{R}\Big(\Lambda_i(|\phi\rangle\langle\phi|)\Big) \cdot \max_{|\phi\rangle} \text{Tr}\left(|\phi\rangle\langle\phi|\, E\right)$$

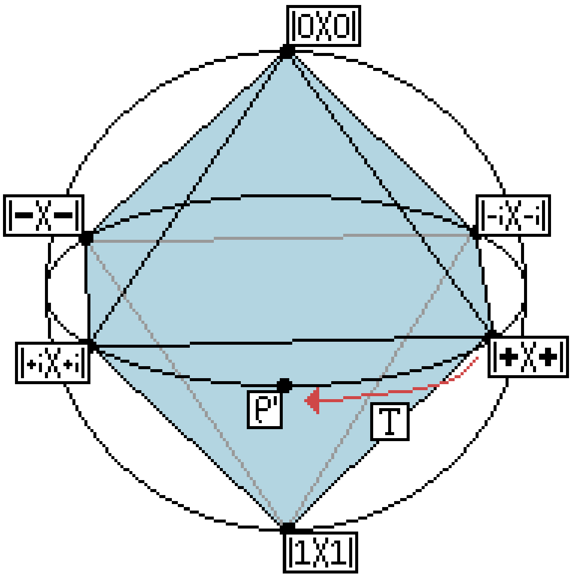
- Hoeffding's inequality gives runtime: $\Pr(\text{error} \geq \varepsilon) \leq \delta$ if

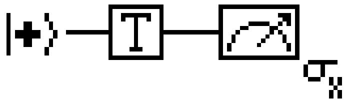
$$\text{samples needed} \geq \frac{1}{\varepsilon^2} \log \frac{1}{\delta} \cdot 4 \max |\text{sample}|^2$$

- Calculating $\mathcal{R}$ is hard. See Markus Heinrich's talk tomorrow!
- Lower bound $\mathcal{D}$ (aka st-norm): Phys. Rev. Lett. 118, 090501

$$\mathcal{R}(\rho) \geq \mathcal{D}(\rho) = 2^{-n} \sum_{\sigma_i} |\text{Tr}\left(\sigma_i \rho\right)|$$

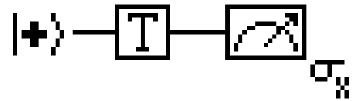
Shuffling Paulis





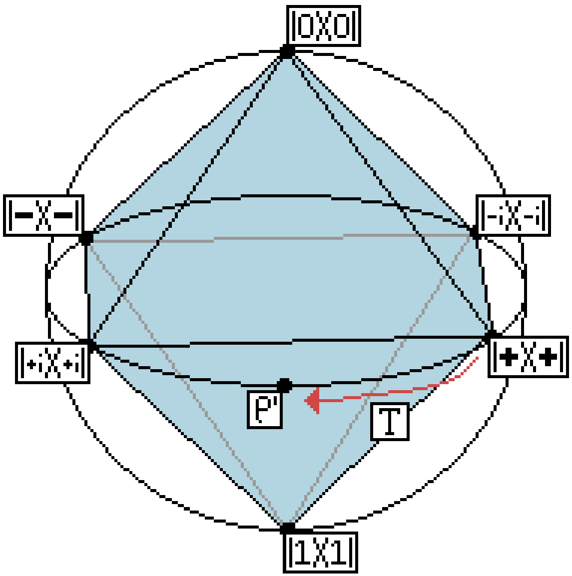
- Decompose input: $|+\rangle\langle+| = \frac{\sigma_X + \sigma_I}{2}$

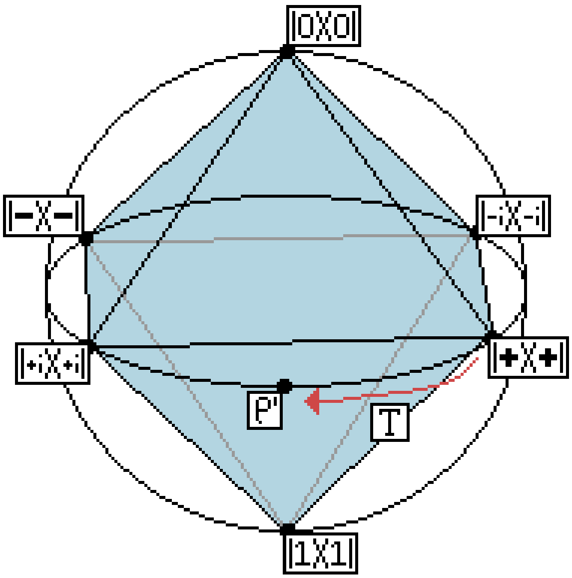
Shuffling Paulis

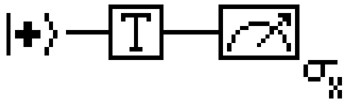


- Decompose input: $|+\rangle\langle+| = \frac{\sigma_X + \sigma_I}{2}$

$$\hat{\rho} = \begin{cases} \sigma_X & \text{with prob. } 1/2 \\ \sigma_I & \text{with prob. } 1/2 \end{cases}$$







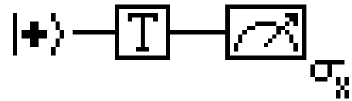
- Decompose input: $|+\rangle\langle+| = \frac{\sigma_X + \sigma_I}{2}$

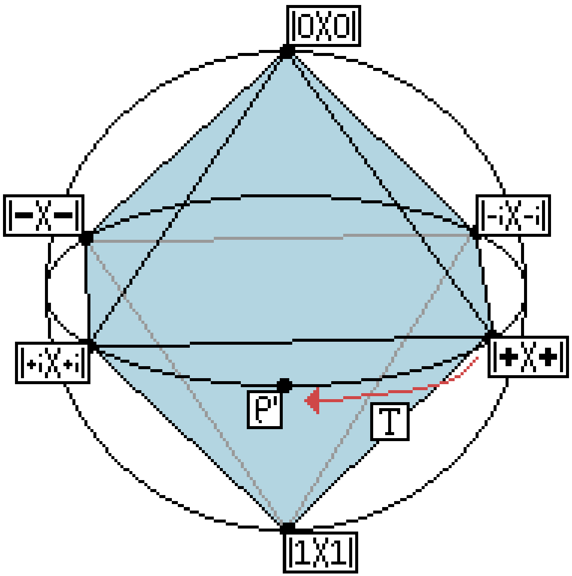
$$\hat{\rho} = \begin{cases} \sigma_X & \text{with prob. } 1/2 \\ \sigma_I & \text{with prob. } 1/2 \end{cases}$$

- Decompose $T\hat{\rho}T^\dagger$:

$$T\sigma_I T^\dagger = \sigma_I; \qquad T\sigma_X T^\dagger = \frac{\sigma_X + \sigma_Y}{\sqrt{2}}$$

Shuffling Paulis





- Decompose input: $|+\rangle\langle+| = \frac{\sigma_X + \sigma_I}{2}$

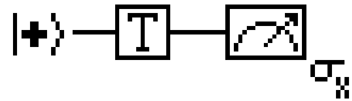
$$\hat{\rho} = \begin{cases} \sigma_X & \text{with prob. } 1/2 \\ \sigma_I & \text{with prob. } 1/2 \end{cases}$$

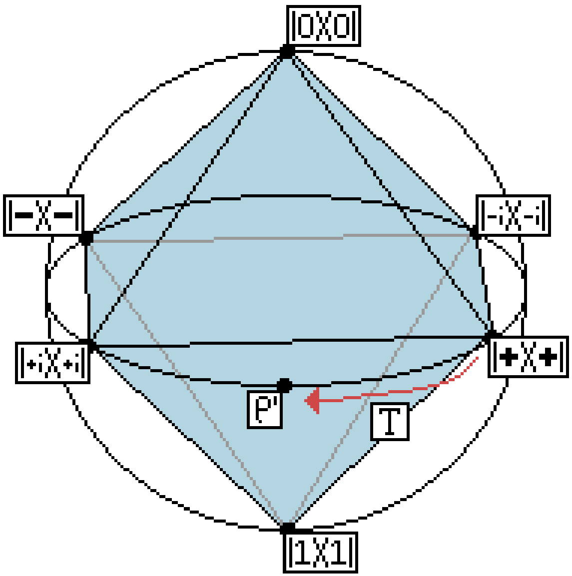
- Decompose $T\hat{\rho}T^\dagger$:

$$T\sigma_I T^\dagger = \sigma_I; \qquad T\sigma_X T^\dagger = \frac{\sigma_X + \sigma_Y}{\sqrt{2}}$$

$$\mathcal{D}(T\sigma_X T^\dagger) = \frac{1}{2}\sum_i \left| \mathrm{Tr}\left(\sigma_i T \sigma_X T^\dagger\right) \right| = \frac{2}{\sqrt{2}} = \sqrt{2}$$

Shuffling Paulis





- Decompose input: $|+\rangle\langle+| = \frac{\sigma_X + \sigma_I}{2}$

$$\hat{\rho} = \begin{cases} \sigma_X & \text{with prob. } 1/2 \\ \sigma_I & \text{with prob. } 1/2 \end{cases}$$

- Decompose $T\hat{\rho}T^\dagger$:

$$T\sigma_I T^\dagger = \sigma_I; \qquad T\sigma_X T^\dagger = \frac{\sigma_X + \sigma_Y}{\sqrt{2}}$$

$$\mathcal{D}(T\sigma_X T^\dagger) = \frac{1}{2}\sum_i \left|\mathrm{Tr}\left(\sigma_i T\sigma_X T^\dagger\right)\right| = \frac{2}{\sqrt{2}} = \sqrt{2}$$

- Sample:

$$\hat{\rho} = \begin{cases} \mathcal{D}(T\sigma_X T^\dagger)\sigma_X & \text{w.p. } (1/\sqrt{2})/\mathcal{D}(T\sigma_X T^\dagger) \\ \mathcal{D}(T\sigma_X T^\dagger)\sigma_Y & \text{w.p. } (1/\sqrt{2})/\mathcal{D}(T\sigma_X T^\dagger) \end{cases}$$

Outline for the rest of the talk

1. Properties of $\mathcal{D}$ and $\mathcal{R}$
2. Hyper-Octahedral States
3. Discarding Qubits
4. Performance in Practice
5. A Cartoon

Properties of $\mathcal{D}$ and $\mathcal{R}$

- If ρ is a stabilizer state:

$$\mathcal{D}(\rho) \leq 1 \qquad \mathcal{R}(\rho) = 1$$

Properties of $\mathcal{D}$ and $\mathcal{R}$

- If ρ is a stabilizer state:

$$\mathcal{D}(\rho) \leq 1 \qquad \mathcal{R}(\rho) = 1$$

- Shuffling Paulis can efficiently simulate Clifford circuits without ever writing down a stabilizer state.

Properties of $\mathcal{D}$ and $\mathcal{R}$

- If ρ is a stabilizer state:

$$\mathcal{D}(\rho) \leq 1 \qquad \mathcal{R}(\rho) = 1$$

- Shuffling Paulis can efficiently simulate Clifford circuits without ever writing down a stabilizer state.
- For all ρ:

$$\mathcal{D}(\rho) \geq 2^{-n} \qquad \mathcal{R}(\rho) \geq 1$$

Properties of $\mathcal{D}$ and $\mathcal{R}$

- If ρ is a stabilizer state:

$$\mathcal{D}(\rho) \leq 1 \qquad \mathcal{R}(\rho) = 1$$

- Shuffling Paulis can efficiently simulate Clifford circuits without ever writing down a stabilizer state.
- For all ρ:

$$\mathcal{D}(\rho) \geq 2^{-n} \qquad \mathcal{R}(\rho) \geq 1$$

- Shuffling Paulis can be faster than shuffling stabilizer states: Highly mixed states with $\mathcal{D}(\rho) \ll 1$ *improve* runtime!

Hyper-Octahedral States

- Families of states:
  - Stabilizer mixtures:

$$\mathcal{R}(\rho) = 1$$

  - Hyper-octahedral states:

$$\mathcal{D}(\rho) \leq 1$$

  - Magic states:

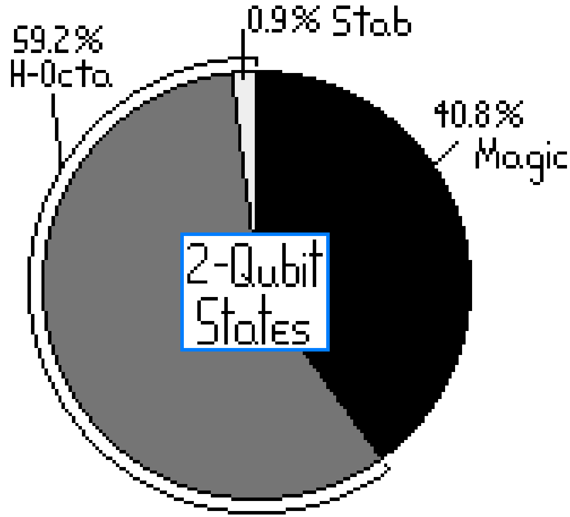
$$\mathcal{D}(\rho) > 1, \mathcal{R}(\rho) > 1$$

Hyper-Octahedral States

- Families of states:
  - Stabilizer mixtures:
    $$\mathcal{R}(\rho) = 1$$
  - Hyper-octahedral states:
    $$\mathcal{D}(\rho) \leq 1$$
  - Magic states:
    $$\mathcal{D}(\rho) > 1, \mathcal{R}(\rho) > 1$$
- Observation: In *multi-qubit systems* there exist hyper-octahedral states that are not stabilizer mixtures!

Hyper-Octahedral States

- Families of states:
  - Stabilizer mixtures:
    $$\mathcal{R}(\rho) = 1$$
  - Hyper-octahedral states:
    $$\mathcal{D}(\rho) \leq 1$$
  - Magic states:
    $$\mathcal{D}(\rho) > 1, \mathcal{R}(\rho) > 1$$



- Observation: In *multi-qubit systems* there exist hyper-octahedral states that are not stabilizer mixtures!
- There are a lot of these. (Based on 100 000 random 2-Qubit states)

Hyper-Octahedral States

- Families of states:
  - Stabilizer mixtures:

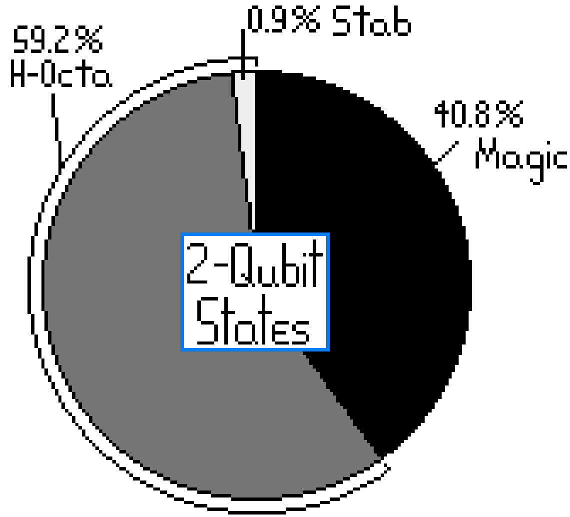
    $$\mathcal{R}(\rho) = 1$$

  - Hyper-octahedral states:

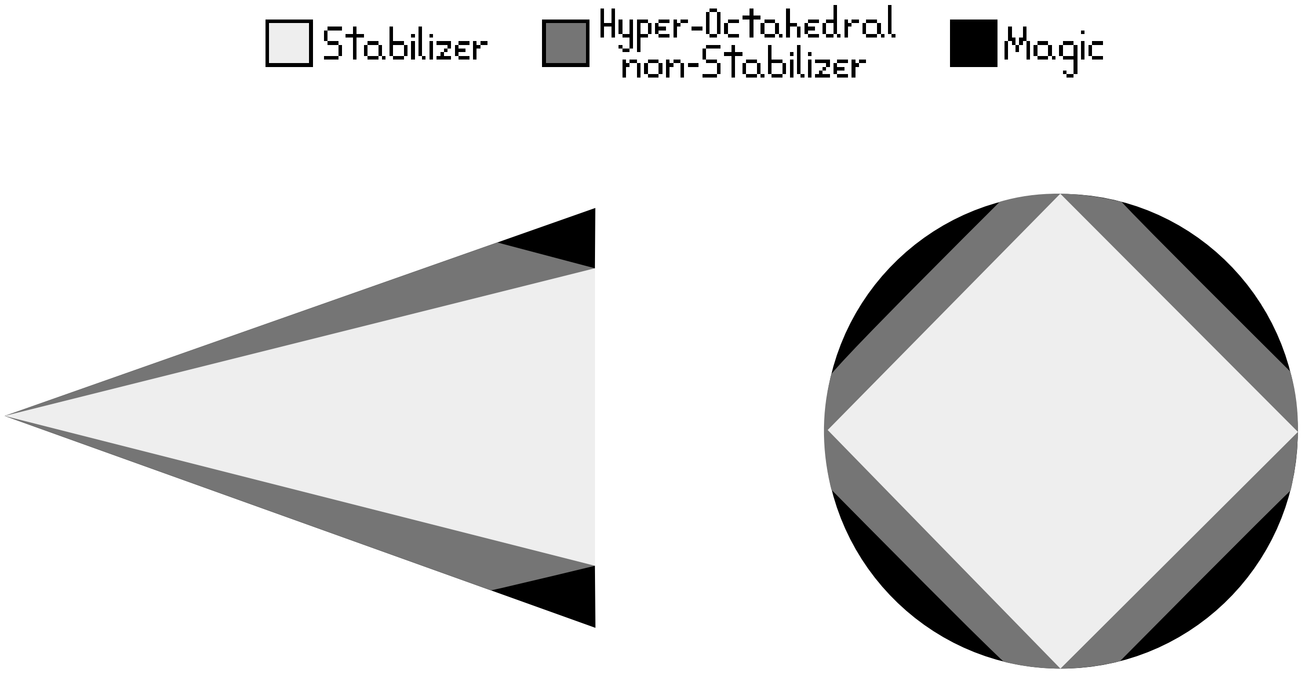
    $$\mathcal{D}(\rho) \leq 1$$

  - Magic states:

    $$\mathcal{D}(\rho) > 1, \mathcal{R}(\rho) > 1$$



- Observation: In *multi-qubit systems* there exist hyper-octahedral states that are not stabilizer mixtures!
- There are a lot of these. (Based on 100 000 random 2-Qubit states)
- Previously known to exist for odd-dimensional systems
  e.g. New J. Phys. 15 039502, doi:10.1038/nature13460



$$\rho(x,y) = \frac{\sigma_{II}}{4} + x\sigma_{ZZ} + y(\sigma_{XX} + \sigma_{XY} + \sigma_{YX} - \sigma_{YY})$$

$$\rho(x,y) = \frac{\sigma_{II} + 0.8\sigma_{ZZ}}{4} + x(\sigma_{XX} - \sigma_{YY}) + y(\sigma_{XY} + \sigma_{YX})$$

Discarding Qubits

- Runtime $\sim \mathcal{O}\left(\max |\text{sample}|^2\right)$

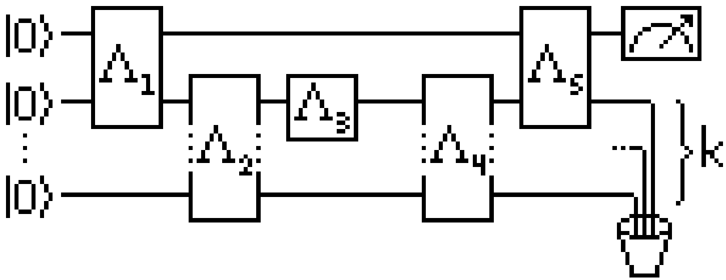
$$|\text{sample}| \leq \mathcal{D}(\rho_{\text{in}}) \cdot \prod_j \max_{\sigma_i} \mathcal{D}\Big(\Lambda_j(\sigma_i)\Big) \cdot \max_{\sigma_i} \text{Tr}\,(\sigma_i E)$$

Discarding Qubits

- Runtime $\sim \mathcal{O}\left(\max|\text{sample}|^2\right)$

$$|\text{sample}| \leq \mathcal{D}(\rho_{\text{in}}) \cdot \prod_j \max_{\sigma_i} \mathcal{D}\Big(\Lambda_j(\sigma_i)\Big) \cdot \max_{\sigma_i} \text{Tr}\left(\sigma_i E\right)$$

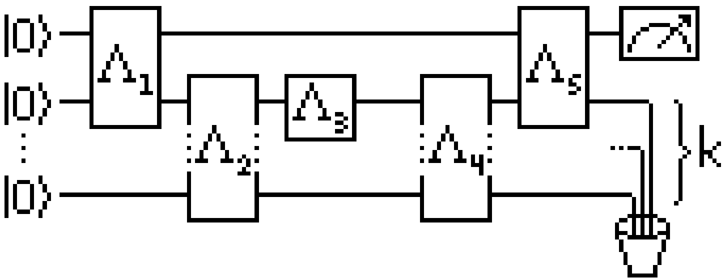
- Circuit with k discarded ancillas:



- Runtime $\sim \mathcal{O}\left(\max|\text{sample}|^2\right)$

$$|\text{sample}| \leq \mathcal{D}(\rho_{\text{in}}) \cdot \prod_j \max_{\sigma_i} \mathcal{D}\Big(\Lambda_j(\sigma_i)\Big) \cdot \max_{\sigma_i} \text{Tr}\left(\sigma_i E\right)$$

- Circuit with k discarded ancillas:

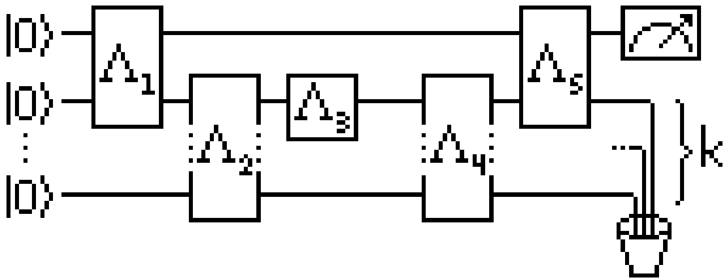


- $E = |0\rangle\langle 0| \otimes \sigma_I^{\otimes k}$ so $\max_{\sigma_i} \text{Tr}\left(\sigma_i E\right) = 2^k$, which can be large!

Discarding Qubits (contd.)

- We can usually recover polynomial runtime in # of qubits.
- Use back-propagation with $\Lambda^{-1}(E)$:

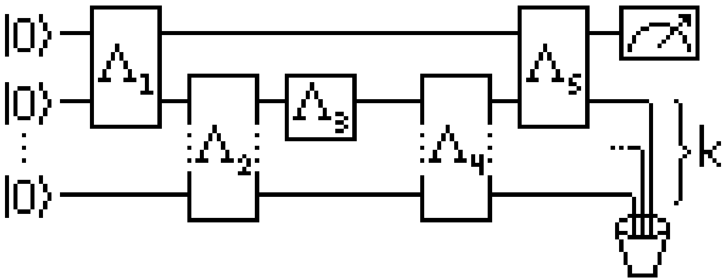
$$\mathrm{Tr}\left(\Lambda(\rho)E\right) = \mathrm{Tr}\left(\rho\Lambda^{-1}(E)\right)$$



Discarding Qubits (contd.)

- We can usually recover polynomial runtime in # of qubits.
- Use back-propagation with $\Lambda^{-1}(E)$:

$$\mathrm{Tr}\left(\Lambda(\rho)E\right) = \mathrm{Tr}\left(\rho\Lambda^{-1}(E)\right) \qquad \Lambda^{-1}(E) = 2^{-n}\sum_{\sigma_i}\sigma_i \mathrm{Tr}\left(\Lambda(\sigma_i)E\right)$$



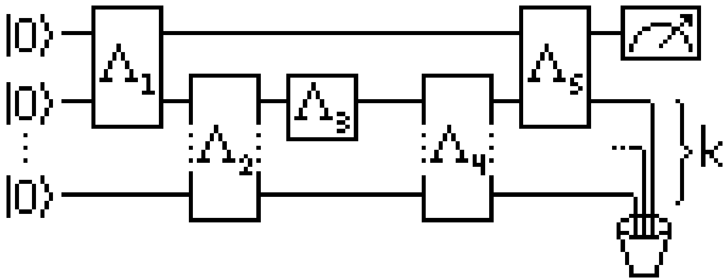
Discarding Qubits (contd.)

- We can usually recover polynomial runtime in # of qubits.
- Use back-propagation with $\Lambda^{-1}(E)$:

$$\mathrm{Tr}\left(\Lambda(\rho)E\right) = \mathrm{Tr}\left(\rho\Lambda^{-1}(E)\right) \qquad \Lambda^{-1}(E) = 2^{-n}\sum_{\sigma_i}\sigma_i \mathrm{Tr}\left(\Lambda(\sigma_i)E\right)$$

- Runtime:

$$|\text{sample}| \leq \mathcal{D}(E)\cdot\prod_j \max_{\sigma_i}\mathcal{D}\left(\Lambda_j^{-1}(\sigma_i)\right)\cdot \max_{\sigma_i}\mathrm{Tr}\left(\sigma_i\rho_{\text{in}}\right)$$



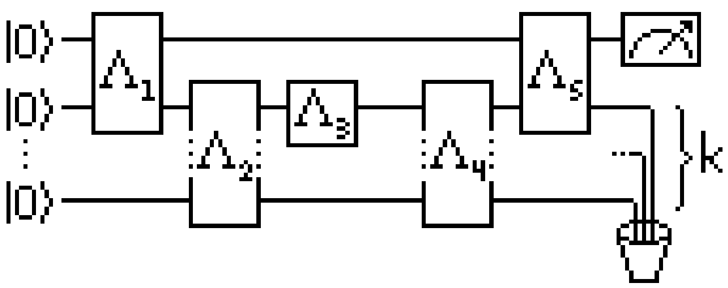
Discarding Qubits (contd.)

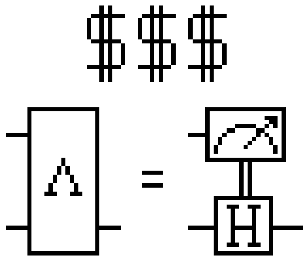
- We can usually recover polynomial runtime in # of qubits.
- Use back-propagation with $\Lambda^{-1}(E)$:

$$\mathrm{Tr}\left(\Lambda(\rho)E\right) = \mathrm{Tr}\left(\rho\Lambda^{-1}(E)\right) \qquad \Lambda^{-1}(E) = 2^{-n}\sum_{\sigma_i}\sigma_i \mathrm{Tr}\left(\Lambda(\sigma_i)E\right)$$

- Runtime:

$$|\text{sample}| \leq \mathcal{D}(E)\cdot\prod_j \max_{\sigma_i}\mathcal{D}\left(\Lambda_j^{-1}(\sigma_i)\right)\cdot\max_{\sigma_i}\mathrm{Tr}\left(\sigma_i\rho_{\text{in}}\right)$$





Performance in Practice

- *Preliminary* data on Texas Maverick supercomputer:
  10 NVIDIA Tesla K40 GPUs take ≈ 40 billion samples in ≈ 30 min.

Performance in Practice

- *Preliminary* data on Texas Maverick supercomputer:
  10 NVIDIA Tesla K40 GPUs take $\approx$ 40 billion samples in $\approx$ 30 min.
- 16 qubit supremacy circuits with depth $\rightarrow$16, T count $\rightarrow$40
  Nothing impressive – I can definitely do larger cases.

Performance in Practice

- *Preliminary* data on Texas Maverick supercomputer:
  10 NVIDIA Tesla K40 GPUs take $\approx$ 40 billion samples in $\approx$ 30 min.
- 16 qubit supremacy circuits with depth $\rightarrow$16, T count $\rightarrow$40
  Nothing impressive – I can definitely do larger cases.
- Added depolarizing noise to each single-qubit gate. $\mathcal{D} = \frac{1+f}{2} \leq 1$

Performance in Practice

- *Preliminary* data on Texas Maverick supercomputer:
  10 NVIDIA Tesla K40 GPUs take $\approx$ 40 billion samples in $\approx$ 30 min.
- 16 qubit supremacy circuits with depth $\rightarrow$16, T count $\rightarrow$40
  Nothing impressive – I can definitely do larger cases.
- Added depolarizing noise to each single-qubit gate. $\mathcal{D} = \frac{1+f}{2} \leq 1$
- Analyzed distribution directly for better error bound.

Performance in Practice

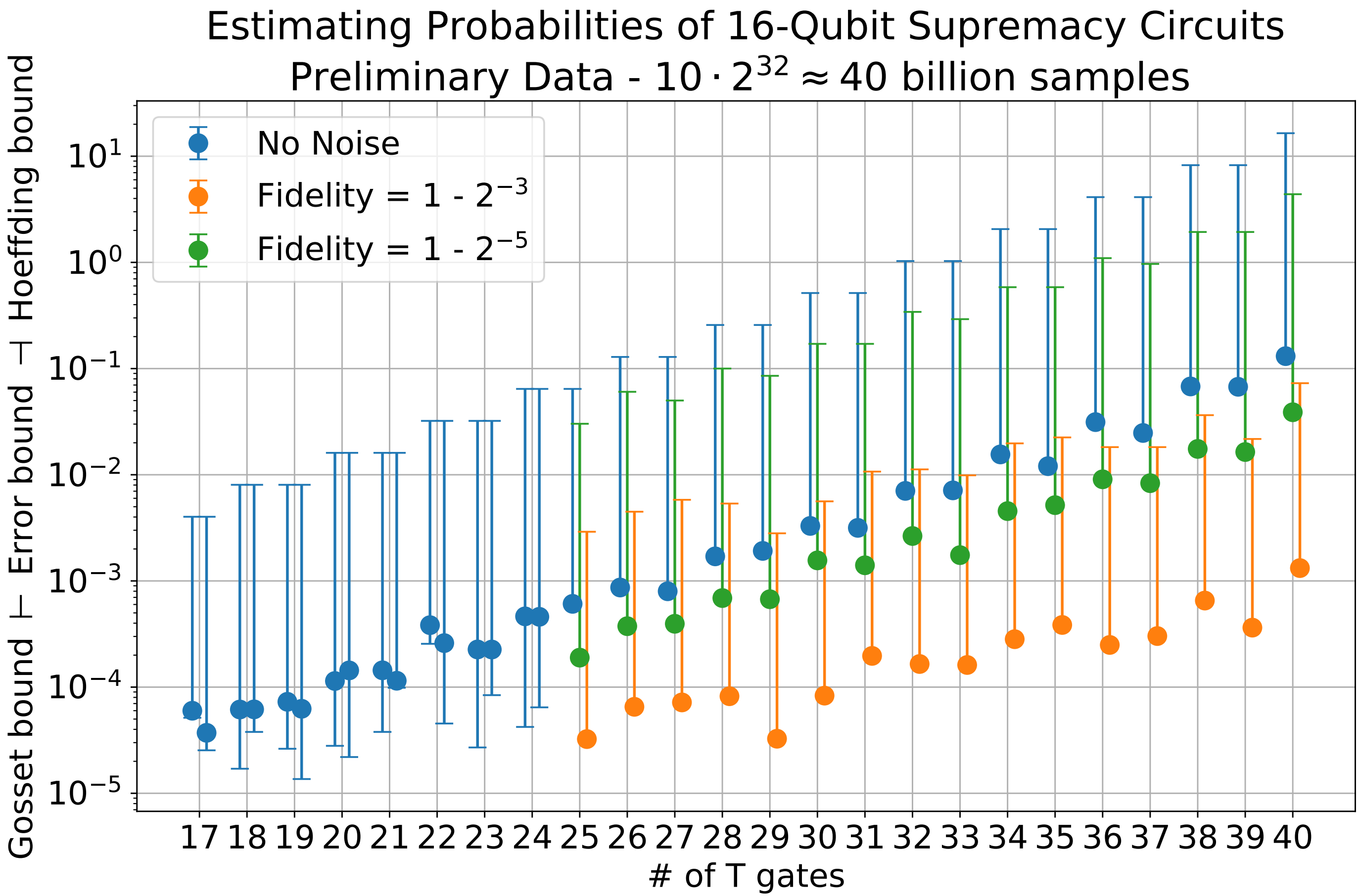
- *Preliminary* data on Texas Maverick supercomputer:
  10 NVIDIA Tesla K40 GPUs take $\approx$ 40 billion samples in $\approx$ 30 min.
- 16 qubit supremacy circuits with depth $\to$16, T count $\to$40
  Nothing impressive – I can definitely do larger cases.
- Added depolarizing noise to each single-qubit gate. $\mathcal{D} = \frac{1+f}{2} \leq 1$
- Analyzed distribution directly for better error bound.
- Verified smaller cases using stabilizer-rank techniques.
  Phys. Rev. Lett. 116, 250501, also see Dr. Earl Campbell's talk tomorrow!

Performance in Practice

- *Preliminary* data on Texas Maverick supercomputer:
  10 NVIDIA Tesla K40 GPUs take $\approx$ 40 billion samples in $\approx$ 30 min.
- 16 qubit supremacy circuits with depth $\rightarrow$16, T count $\rightarrow$40
  Nothing impressive – I can definitely do larger cases.
- Added depolarizing noise to each single-qubit gate. $\mathcal{D} = \frac{1+f}{2} \leq 1$
- Analyzed distribution directly for better error bound.
- Verified smaller cases using stabilizer-rank techniques.
  Phys. Rev. Lett. 116, 250501, also see Dr. Earl Campbell's talk tomorrow!

- Question: Is there an algorithm that:
  - is fast for Clifford circuits,
  - supports noisy channels,
  - *and* gives multiplicative error?

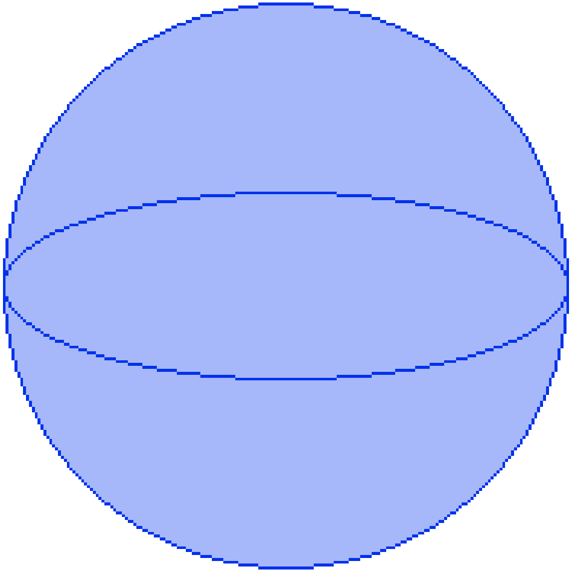
Performance in Practice (contd.)



A Cartoon

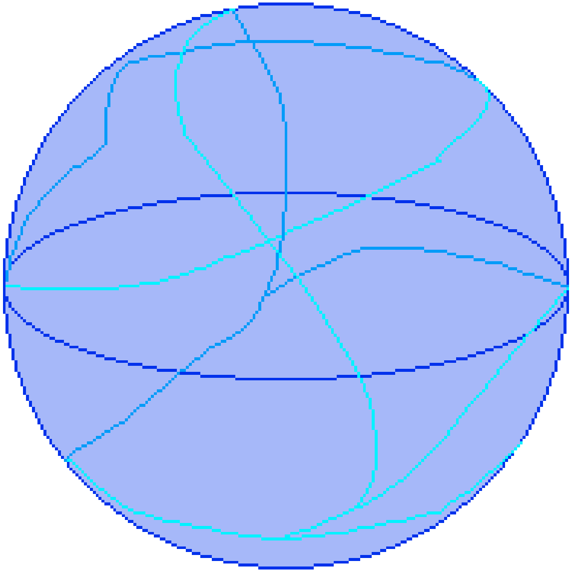
- Special thanks to:
  - Dr. Scott Aaronson
  - Dr. James Troupe
  - Daniel Liang

A Cartoon



- Pure states $\rho^2 = \rho$ are on a hyper-sphere...
- Special thanks to:
  - Dr. Scott Aaronson
  - Dr. James Troupe
  - Daniel Liang

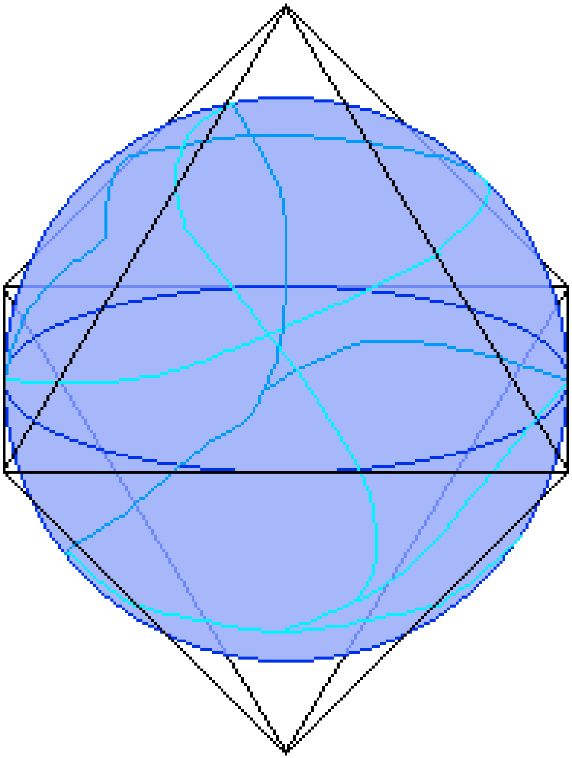
A Cartoon



- Pure states $\rho^2 = \rho$ are on a hyper-sphere...
  ... but there aren't enough to cover it!

- Special thanks to:
  - Dr. Scott Aaronson
  - Dr. James Troupe
  - Daniel Liang

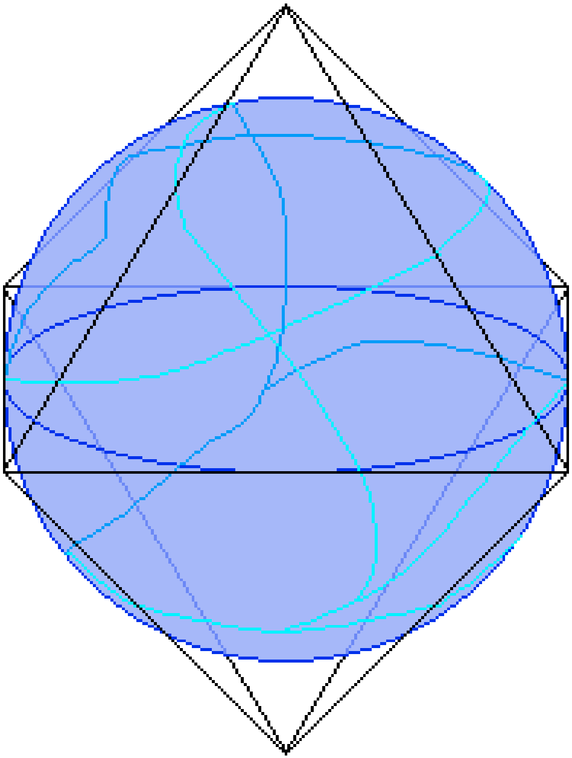
A Cartoon



- Pure states $\rho^2 = \rho$ are on a hyper-sphere...
  ... but there aren't enough to cover it!
- $\mathcal{D}(\rho) \leq 1$ defines a hyper-octahedron.

- Special thanks to:
  - Dr. Scott Aaronson
  - Dr. James Troupe
  - Daniel Liang

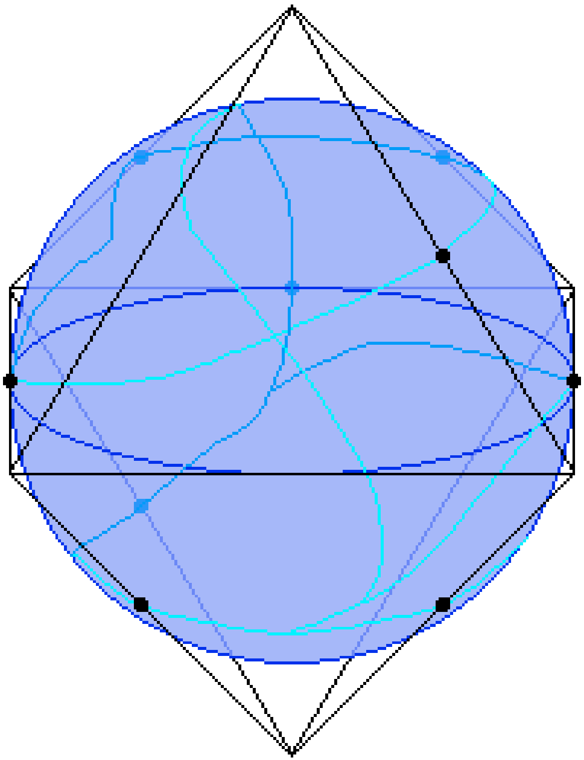
A Cartoon



- Pure states $\rho^2 = \rho$ are on a hyper-sphere... ... but there aren't enough to cover it!
- $\mathcal{D}(\rho) \leq 1$ defines a hyper-octahedron.
- Stab. states lie on $(2^n - 2)$-faces...

- Special thanks to:
  - Dr. Scott Aaronson
  - Dr. James Troupe
  - Daniel Liang

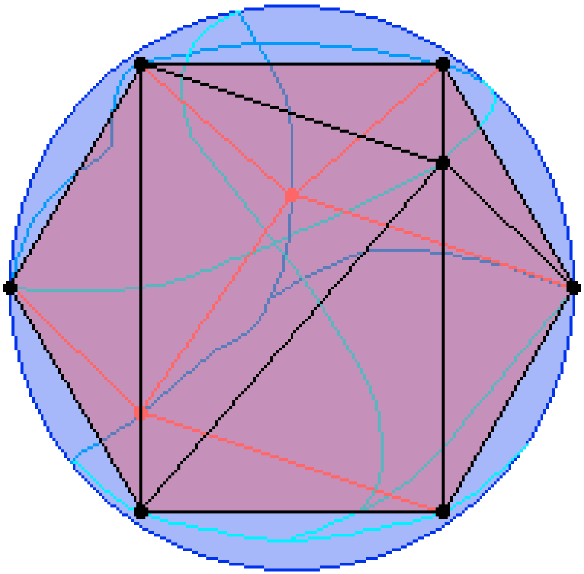
A Cartoon



- Pure states $\rho^2 = \rho$ are on a hyper-sphere...
  ... but there aren't enough to cover it!
- $\mathcal{D}(\rho) \leq 1$ defines a hyper-octahedron.
- Stab. states lie on $(2^n - 2)$-faces...
  ... but only where $\rho^2 = \rho$!

- Special thanks to:
  - Dr. Scott Aaronson
  - Dr. James Troupe
  - Daniel Liang

A Cartoon



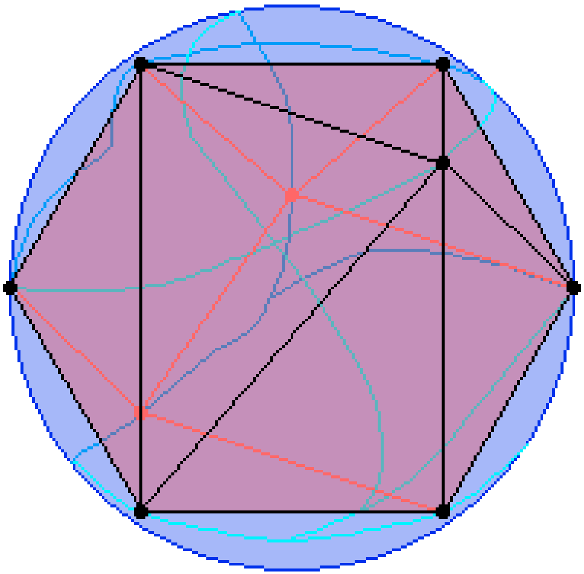
- Pure states $\rho^2 = \rho$ are on a hyper-sphere...
  ... but there aren't enough to cover it!
- $\mathcal{D}(\rho) \leq 1$ defines a hyper-octahedron.
- Stab. states lie on $(2^n - 2)$-faces...
  ... but only where $\rho^2 = \rho$!

$$\text{stabilizer polytope} =$$

$$\text{hyper-octahedron} + \left[\rho^2 = \rho\right]$$

- Special thanks to:
  - Dr. Scott Aaronson
  - Dr. James Troupe
  - Daniel Liang

A Cartoon



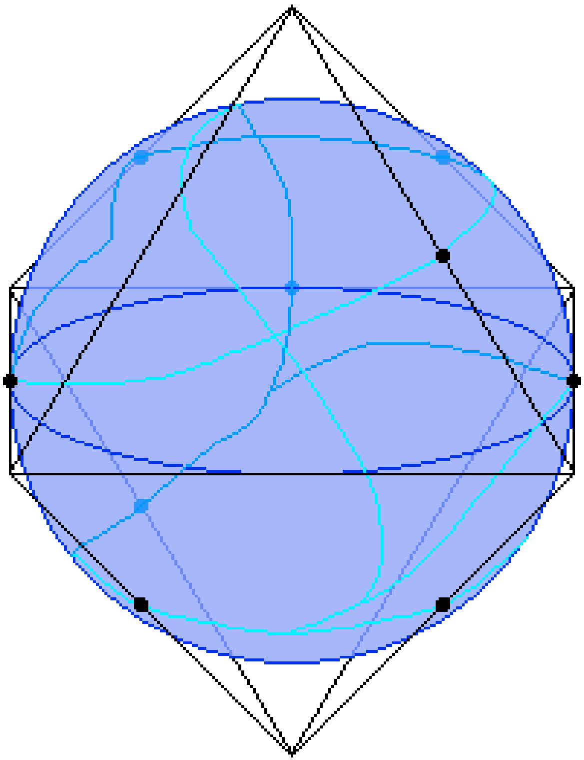
- Pure states $\rho^2 = \rho$ are on a hyper-sphere...
  ... but there aren't enough to cover it!
- $\mathcal{D}(\rho) \leq 1$ defines a hyper-octahedron.
- Stab. states lie on $(2^n - 2)$-faces...
  ... but only where $\rho^2 = \rho$!

$$\text{stabilizer polytope} =$$

$$\text{hyper-octahedron} + \left[\rho^2 = \rho\right]$$

- Special thanks to:
  - Dr. Scott Aaronson
  - Dr. James Troupe
  - Daniel Liang
- Thank you for listening!

A Cartoon



- Pure states $\rho^2 = \rho$ are on a hyper-sphere...
  ... but there aren't enough to cover it!
- $\mathcal{D}(\rho) \leq 1$ defines a hyper-octahedron.
- Stab. states lie on $(2^n - 2)$-faces...
  ... but only where $\rho^2 = \rho$!

$$\text{stabilizer polytope} =$$

$$\text{hyper-octahedron} + \left[\rho^2 = \rho\right]$$

- Special thanks to:
  - Dr. Scott Aaronson
  - Dr. James Troupe
  - Daniel Liang
- Thank you for listening!