Quantum computing with a reasonable overhead

Simplified quantum compiling with complex gate distillation

Guillaume Duclos-Cianci & David Poulin

Équipe de recherche sur la physique de l'information quantique
Département de Physique
Université de Sherbrooke

Sydney Quantum Information Theory Workshop
Coogee, January 2014

Outline

Outline

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

$$\Rightarrow |E_t\rangle = \epsilon V_t |\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

How to rotate a qubit by 0.23π?

- Along the z axis: just wait for $t = 0.23\pi\hbar/J$ seconds.
- Along the x axis: just Rabi pulse the qubit for $t = 0.23\pi\hbar/A$ seconds.
- General by Euler angle decomposition.

How about errors?

- Algorithm uses 10^6 qubits and has dept 10^8.
- There are 10^{14} occasions to pick up errors

$$|\psi_t\rangle = |\phi_t^0\rangle + |E_t\rangle = (U_t + \epsilon V_t)(|\phi_{t-1}^0\rangle + |E_{t-1}\rangle)$$

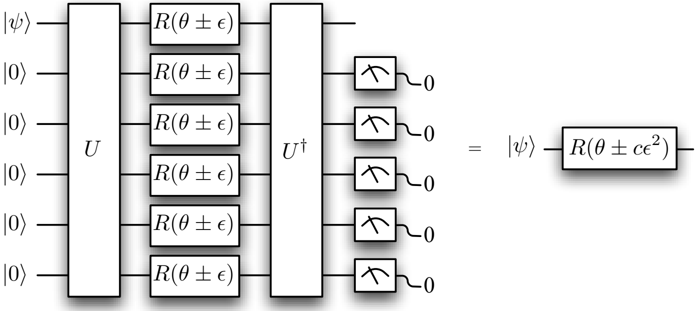
$$\Rightarrow |E_t\rangle = \epsilon V_t|\phi_{t-1}^0\rangle + (U_t + \epsilon V_t)|E_{t-1}\rangle$$

$$\Rightarrow |E_t| \leq \epsilon + |E_{t-1}|$$

- The final error is proportional to the number of gates (identity).
- Each gate requires accuracy $\ll 10^{-14}$.

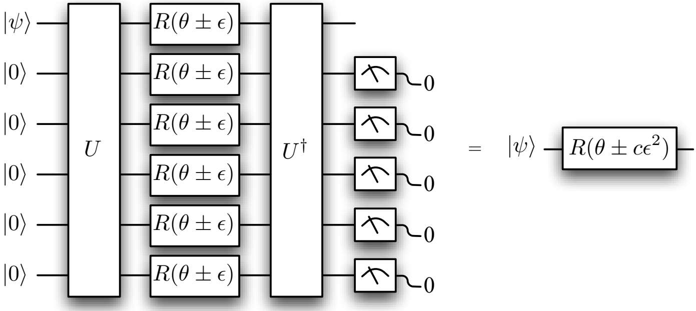
Outline

Transversal gates



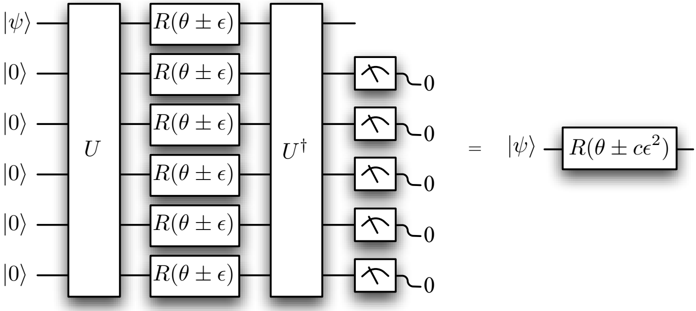
- This only works for very special angles θ.
- Error can be further suppressed by iterating (concatenation).
- We can realize the Clifford group this way: CNOT, H, P = $Z^{1/2}$.
- U itself is Clifford, so iterations don't introduce more errors.
- Not a universal gate set.
- Slightly more general setting admits T= $Z^{1/4}$, universal.

Transversal gates



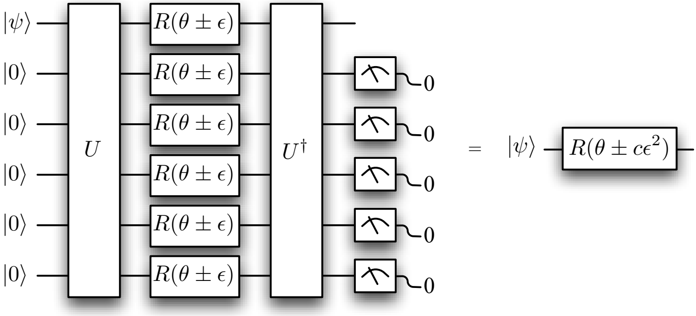
- This only works for very special angles θ.
- Error can be further suppressed by iterating (concatenation).
- We can realize the Clifford group this way: CNOT, H, P = $Z^{1/2}$.
- U itself is Clifford, so iterations don't introduce more errors.
- Not a universal gate set.
- Slightly more general setting admits T= $Z^{1/4}$, universal.

Transversal gates



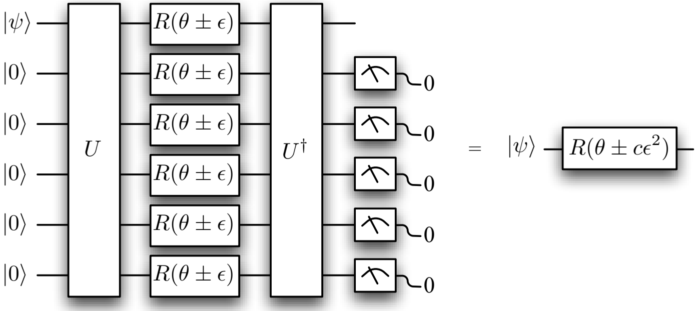
- This only works for very special angles θ.
- Error can be further suppressed by iterating (concatenation).
- We can realize the Clifford group this way: CNOT, H, P = $Z^{1/2}$.
- U itself is Clifford, so iterations don't introduce more errors.
- Not a universal gate set.
- Slightly more general setting admits T= $Z^{1/4}$, universal.

Transversal gates



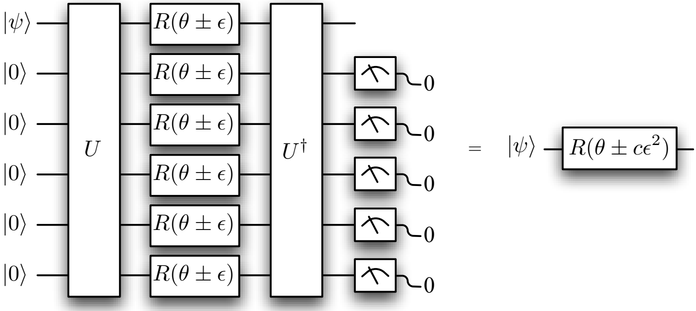
- This only works for very special angles θ.
- Error can be further suppressed by iterating (concatenation).
- We can realize the Clifford group this way: CNOT, H, P = Z$^{1/2}$.
- U itself is Clifford, so iterations don't introduce more errors.
- Not a universal gate set.
- Slightly more general setting admits T= Z$^{1/4}$, universal.

Transversal gates



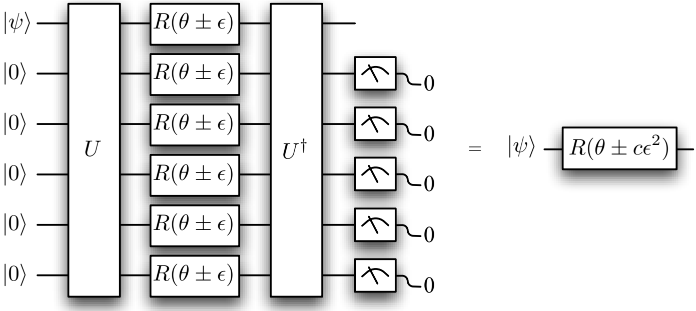
- This only works for very special angles θ.
- Error can be further suppressed by iterating (concatenation).
- We can realize the Clifford group this way: CNOT, H, P = $Z^{1/2}$.
- U itself is Clifford, so iterations don't introduce more errors.
- Not a universal gate set.
- Slightly more general setting admits T= $Z^{1/4}$, universal.

Transversal gates



- This only works for very special angles θ.
- Error can be further suppressed by iterating (concatenation).
- We can realize the Clifford group this way: CNOT, H, P = $Z^{1/2}$.
- U itself is Clifford, so iterations don't introduce more errors.
- Not a universal gate set.
- Slightly more general setting admits T= $Z^{1/4}$, universal.

Transversal gates

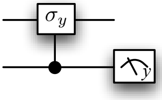


- This only works for very special angles θ.
- Error can be further suppressed by iterating (concatenation).
- We can realize the Clifford group this way: CNOT, H, P = $\mathrm{Z}^{1/2}$.
- U itself is Clifford, so iterations don't introduce more errors.
- Not a universal gate set.
- Slightly more general setting admits T= $\mathrm{Z}^{1/4}$, universal.

State injection

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|Y_\theta\rangle = \cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}|1\rangle$$



$$\cos\frac{\theta}{2}|0\rangle|\psi\rangle + \sin\frac{\theta}{2}|1\rangle\sigma_y|\psi\rangle$$

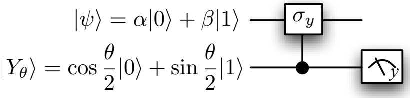
$$= \cos\frac{\theta}{2}\frac{|i\rangle + |-i\rangle}{\sqrt{2}}|\psi\rangle - i\sin\frac{\theta}{2}\frac{|i\rangle - |-i\rangle}{\sqrt{2}}\sigma_y|\psi\rangle$$

- Measure $|i\rangle$: $(\cos\frac{\theta}{2} - i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(-\theta)|\psi\rangle$
- Measure $|-i\rangle$: $(\cos\frac{\theta}{2} + i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(\theta)|\psi\rangle$

Can realize a rotation around y of angle $\pm\theta$ given state $|Y_\theta\rangle$ and Clifford operations.

- An error $\theta \pm \epsilon$ in the state $|Y_\theta\rangle$ translate into an error ϵ in the gate.

State injection



$$\cos\frac{\theta}{2}|0\rangle|\psi\rangle + \sin\frac{\theta}{2}|1\rangle\sigma_y|\psi\rangle$$

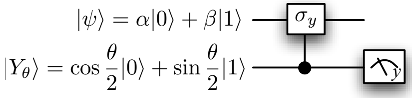
$$= \cos\frac{\theta}{2}\frac{|i\rangle + |-i\rangle}{\sqrt{2}}|\psi\rangle - i\sin\frac{\theta}{2}\frac{|i\rangle - |-i\rangle}{\sqrt{2}}\sigma_y|\psi\rangle$$

- Measure $|i\rangle$: $(\cos\frac{\theta}{2} - i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(-\theta)|\psi\rangle$
- Measure $|-i\rangle$: $(\cos\frac{\theta}{2} + i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(\theta)|\psi\rangle$

Can realize a rotation around y of angle $\pm\theta$ given state $|Y_\theta\rangle$ and Clifford operations.

- An error $\theta \pm \epsilon$ in the state $|Y_\theta\rangle$ translate into an error ϵ in the gate.

State injection



$$\cos\frac{\theta}{2}|0\rangle|\psi\rangle + \sin\frac{\theta}{2}|1\rangle\sigma_y|\psi\rangle$$

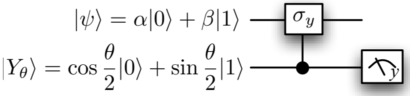
$$= \cos\frac{\theta}{2}\frac{|i\rangle + |-i\rangle}{\sqrt{2}}|\psi\rangle - i\sin\frac{\theta}{2}\frac{|i\rangle - |-i\rangle}{\sqrt{2}}\sigma_y|\psi\rangle$$

- Measure $|i\rangle$: $(\cos\frac{\theta}{2} - i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(-\theta)|\psi\rangle$
- Measure $|-i\rangle$: $(\cos\frac{\theta}{2} + i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(\theta)|\psi\rangle$

Can realize a rotation around y of angle $\pm\theta$ given state $|Y_\theta\rangle$ and Clifford operations.

- An error $\theta \pm \epsilon$ in the state $|Y_\theta\rangle$ translate into an error ϵ in the gate.

State injection



$$\cos\frac{\theta}{2}|0\rangle|\psi\rangle + \sin\frac{\theta}{2}|1\rangle\sigma_y|\psi\rangle$$

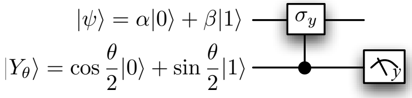
$$= \cos\frac{\theta}{2}\frac{|i\rangle + |-i\rangle}{\sqrt{2}}|\psi\rangle - i\sin\frac{\theta}{2}\frac{|i\rangle - |-i\rangle}{\sqrt{2}}\sigma_y|\psi\rangle$$

- Measure $|i\rangle$: $(\cos\frac{\theta}{2} - i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(-\theta)|\psi\rangle$
- Measure $|-i\rangle$: $(\cos\frac{\theta}{2} + i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(\theta)|\psi\rangle$

Can realize a rotation around y of angle $\pm\theta$ given state $|Y_\theta\rangle$ and Clifford operations.

- An error $\theta \pm \epsilon$ in the state $|Y_\theta\rangle$ translate into an error ϵ in the gate.

State injection



$$\cos\frac{\theta}{2}|0\rangle|\psi\rangle + \sin\frac{\theta}{2}|1\rangle\sigma_y|\psi\rangle$$

$$= \cos\frac{\theta}{2}\frac{|i\rangle + |-i\rangle}{\sqrt{2}}|\psi\rangle - i\sin\frac{\theta}{2}\frac{|i\rangle - |-i\rangle}{\sqrt{2}}\sigma_y|\psi\rangle$$

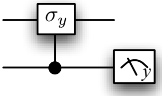
- Measure $|i\rangle$: $(\cos\frac{\theta}{2} - i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(-\theta)|\psi\rangle$
- Measure $|-i\rangle$: $(\cos\frac{\theta}{2} + i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(\theta)|\psi\rangle$

Can realize a rotation around y of angle $\pm\theta$ given state $|Y_\theta\rangle$ and Clifford operations.

- An error $\theta \pm \epsilon$ in the state $|Y_\theta\rangle$ translate into an error ϵ in the gate.

State injection

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$
$$|Y_\theta\rangle = \cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}|1\rangle$$



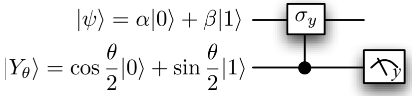
$$\cos\frac{\theta}{2}|0\rangle|\psi\rangle + \sin\frac{\theta}{2}|1\rangle\sigma_y|\psi\rangle$$
$$= \cos\frac{\theta}{2}\frac{|i\rangle + |-i\rangle}{\sqrt{2}}|\psi\rangle - i\sin\frac{\theta}{2}\frac{|i\rangle - |-i\rangle}{\sqrt{2}}\sigma_y|\psi\rangle$$

- Measure $|i\rangle$: $(\cos\frac{\theta}{2} - i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(-\theta)|\psi\rangle$
- Measure $|-i\rangle$: $(\cos\frac{\theta}{2} + i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(\theta)|\psi\rangle$

Can realize a rotation around y of angle $\pm\theta$ given state $|Y_\theta\rangle$ and Clifford operations.

- An error $\theta \pm \epsilon$ in the state $|Y_\theta\rangle$ translate into an error ϵ in the gate.

State injection



$$\cos\frac{\theta}{2}|0\rangle|\psi\rangle + \sin\frac{\theta}{2}|1\rangle\sigma_y|\psi\rangle$$

$$= \cos\frac{\theta}{2}\frac{|i\rangle + |-i\rangle}{\sqrt{2}}|\psi\rangle - i\sin\frac{\theta}{2}\frac{|i\rangle - |-i\rangle}{\sqrt{2}}\sigma_y|\psi\rangle$$

- Measure $|i\rangle$: $(\cos\frac{\theta}{2} - i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(-\theta)|\psi\rangle$
- Measure $|-i\rangle$: $(\cos\frac{\theta}{2} + i\sin\frac{\theta}{2}\sigma_y)|\psi\rangle = R_y(\theta)|\psi\rangle$

Can realize a rotation around y of angle $\pm\theta$ given state $|Y_\theta\rangle$ and Clifford operations.

- An error $\theta \pm \epsilon$ in the state $|Y_\theta\rangle$ translate into an error ϵ in the gate.

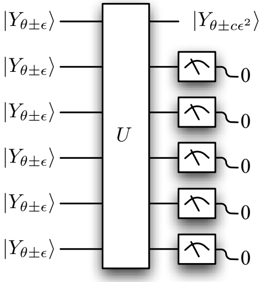
Magic state distillation

How to get accurate states $|Y_\theta\rangle$?

- U is Clifford.
- This only works for very special angles θ.
- With Clifford operations, all we need is $\theta = \pi/8$ to get T = $Z^{1/4}$, universal.

Magic state distillation

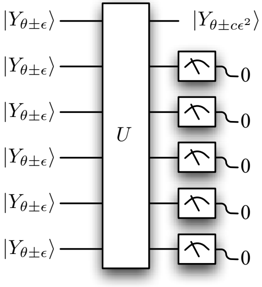
How to get accurate states $|Y_\theta\rangle$?



- U is Clifford.
- This only works for very special angles θ.
- With Clifford operations, all we need is $\theta = \pi/8$ to get T = $Z^{1/4}$, universal.

Magic state distillation

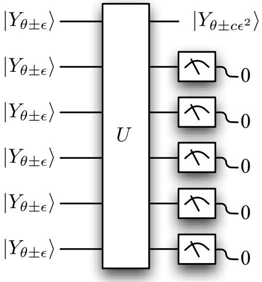
How to get accurate states $|Y_\theta\rangle$?



- U is Clifford.
- This only works for very special angles θ.
- With Clifford operations, all we need is $\theta = \pi/8$ to get T = $Z^{1/4}$, universal.

Magic state distillation

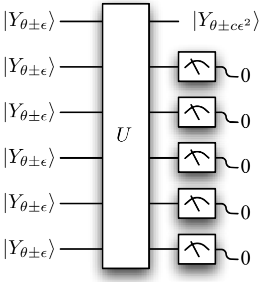
How to get accurate states $|Y_\theta\rangle$?



- U is Clifford.
- This only works for very special angles θ.
- With Clifford operations, all we need is $\theta = \pi/8$ to get T = Z$^{1/4}$, universal.

Magic state distillation

How to get accurate states $|Y_\theta\rangle$?



- U is Clifford.
- This only works for very special angles θ.
- With Clifford operations, all we need is $\theta = \pi/8$ to get T = $Z^{1/4}$, universal.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

- This is the number of noisy $|Y_{\pi/8}\rangle$ states needed to distill one high-accuracy $|Y_{\pi/8}\rangle$.
- It is assumed that Clifford operations are noiseless.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

- This is the number of noisy $|Y_{\pi/8}\rangle$ states needed to distill one high-accuracy $|Y_{\pi/8}\rangle$.
- It is assumed that Clifford operations are noiseless.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

- This is the number of noisy $|Y_{\pi/8}\rangle$ states needed to distill one high-accuracy $|Y_{\pi/8}\rangle$.
- It is assumed that Clifford operations are noiseless.

Outline

Compilation

How to rotate a qubit by 0.23π?

- Get a universal set of gates CNOT, H, P = Z$^{1/2}$, T = Z$^{1/4}$.
  - Because they are "transversal" in the code.
  - By distilling $|Y_{\pi/8}\rangle$.
- Compile: R(0.23π) $\approx$HTHPTPTHTHPTPTPTHPTHPHTPHTPTH
- Precision δ requires $\mathcal{O}(\log^c \frac{1}{\delta})$ gates (Solovay-Kitaev).
- Hidden constant in $\mathcal{O}$ are huge.

Compilation

How to rotate a qubit by 0.23π?

- Get a universal set of gates CNOT, H, P = $Z^{1/2}$, T = $Z^{1/4}$.
  - Because they are "transversal" in the code.
  - By distilling $|Y_{\pi/8}\rangle$.
- Compile: R(0.23π) $\approx$HTHPTPTHTHPTPTPTHPTHPHTPHTPTH
- Precision δ requires $\mathcal{O}(\log^c \frac{1}{\delta})$ gates (Solovay-Kitaev).
- Hidden constant in $\mathcal{O}$ are huge.

Compilation

How to rotate a qubit by 0.23π?

- Get a universal set of gates CNOT, H, P = Z$^{1/2}$, T = Z$^{1/4}$.
  - Because they are "transversal" in the code.
  - By distilling $|Y_{\pi/8}\rangle$.
- Compile: R(0.23π) $\approx$HTHPTPTHTHPTPTPTHPTHPHTPHTPTH
- Precision δ requires $\mathcal{O}(\log^c \frac{1}{\delta})$ gates (Solovay-Kitaev).
- Hidden constant in $\mathcal{O}$ are huge.

Compilation

How to rotate a qubit by 0.23π?

- Get a universal set of gates CNOT, H, P = $Z^{1/2}$, T = $Z^{1/4}$.
  - Because they are "transversal" in the code.
  - By distilling $|Y_{\pi/8}\rangle$.
- Compile: R(0.23π) $\approx$HTHPTPTHTHPTPTPTHPTHPHTPHTPTH
- Precision δ requires $\mathcal{O}(\log^c \frac{1}{\delta})$ gates (Solovay-Kitaev).
- Hidden constant in $\mathcal{O}$ are huge.

Compilation

How to rotate a qubit by 0.23π?

- Get a universal set of gates CNOT, H, P = $Z^{1/2}$, T = $Z^{1/4}$.
  - Because they are "transversal" in the code.
  - By distilling $|Y_{\pi/8}\rangle$.
- Compile: R(0.23π) $\approx$HTHPTPTHTHPTPTPTHPTHPHTPHTPTH
- Precision δ requires $\mathcal{O}(\log^c \frac{1}{\delta})$ gates (Solovay-Kitaev).
- Hidden constant in $\mathcal{O}$ are huge.

Compilation

How to rotate a qubit by 0.23π?

- Get a universal set of gates CNOT, H, P = Z$^{1/2}$, T = Z$^{1/4}$.
  - Because they are "transversal" in the code.
  - By distilling $|Y_{\pi/8}\rangle$.
- Compile: R(0.23π) $\approx$HTHPTPTHTHPTPTPTHPTHPHTPHTPTH
- Precision δ requires $\mathcal{O}(\log^c \frac{1}{\delta})$ gates (Solovay-Kitaev).
- Hidden constant in $\mathcal{O}$ are huge.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

Precision	# T gates
10^{-3}	28
10^{-5}	132
10^{-7}	670
10^{-10}	3,284
10^{-15}	14,312
10^{-22}	74,162
10^{-33}	347,388
10^{-51}	1,692,692

Kliuchnikov, Maslov, & Mosca

- Need precision 10^{-14}
- Compiled logical gates will use 10^4 T gates
- Each of these T gates must have accuracy $\frac{10^{-14}}{10^4} = 10^{-18}$

Overhead = $14,312 \times 697 = 9,975,464$

Clifford operation cost not accounted.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

Precision	# T gates
10^{-3}	28
10^{-5}	132
10^{-7}	670
10^{-10}	3,284
10^{-15}	14,312
10^{-22}	74,162
10^{-33}	347,388
10^{-51}	1,692,692

Kliuchnikov, Maslov, & Mosca

- Need precision 10^{-14}
- Compiled logical gates will use 10^4 T gates
- Each of these T gates must have accuracy $\frac{10^{-14}}{10^4} = 10^{-18}$

Overhead $= 14,312 \times 697 = 9,975,464$
Clifford operation cost not accounted.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

Precision	# T gates
10^{-3}	28
10^{-5}	132
10^{-7}	670
10^{-10}	3,284
10^{-15}	14,312
10^{-22}	74,162
10^{-33}	347,388
10^{-51}	1,692,692

Kliuchnikov, Maslov, & Mosca

- Need precision 10^{-14}
- Compiled logical gates will use 10^4 T gates
- Each of these T gates must have accuracy $\frac{10^{-14}}{10^4} = 10^{-18}$

Overhead $= 14,312 \times 697 = 9,975,464$
Clifford operation cost not accounted.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

Precision	# T gates
10^{-3}	28
10^{-5}	132
10^{-7}	670
10^{-10}	3,284
10^{-15}	14,312
10^{-22}	74,162
10^{-33}	347,388
10^{-51}	1,692,692

Kliuchnikov, Maslov, & Mosca

- Need precision 10^{-14}
- Compiled logical gates will use 10^4 T gates
- Each of these T gates must have accuracy $\frac{10^{-14}}{10^4} = 10^{-18}$

Overhead $= 14,312 \times 697 = 9,975,464$

Clifford operation cost not accounted.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

Precision	# T gates
10^{-3}	28
10^{-5}	132
10^{-7}	670
10^{-10}	3,284
10^{-15}	14,312
10^{-22}	74,162
10^{-33}	347,388
10^{-51}	1,692,692

Kliuchnikov, Maslov, & Mosca

- Need precision 10^{-14}
- Compiled logical gates will use 10^4 T gates
- Each of these T gates must have accuracy $\frac{10^{-14}}{10^4} = 10^{-18}$

Overhead $= 14,312 \times 697 = 9,975,464$
Clifford operation cost not accounted.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

Precision	# T gates
10^{-3}	28
10^{-5}	132
10^{-7}	670
10^{-10}	3,284
10^{-15}	14,312
10^{-22}	74,162
10^{-33}	347,388
10^{-51}	1,692,692

Kliuchnikov, Maslov, & Mosca

- Need precision 10^{-14}
- Compiled logical gates will use 10^4 T gates
- Each of these T gates must have accuracy $\frac{10^{-14}}{10^4} = 10^{-18}$

Overhead $= 14,312 \times 697 = 9,975,464$

Clifford operation cost not accounted.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

Precision	# T gates
10^{-3}	28
10^{-5}	132
10^{-7}	670
10^{-10}	3,284
10^{-15}	14,312
10^{-22}	74,162
10^{-33}	347,388
10^{-51}	1,692,692

Kliuchnikov, Maslov, & Mosca

- Need precision 10^{-14}
- Compiled logical gates will use 10^4 T gates
- Each of these T gates must have accuracy $\frac{10^{-14}}{10^4} = 10^{-18}$

Overhead $= 14,312 \times 697 = 9,975,464$
Clifford operation cost not accounted.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

Precision	# T gates
10^{-3}	28
10^{-5}	132
10^{-7}	670
10^{-10}	3,284
10^{-15}	14,312
10^{-22}	74,162
10^{-33}	347,388
10^{-51}	1,692,692

Kliuchnikov, Maslov, & Mosca

- Need precision 10^{-14}
- Compiled logical gates will use 10^4 T gates
- Each of these T gates must have accuracy $\frac{10^{-14}}{10^4} = 10^{-18}$

Overhead = $14,312 \times 697 = 9,975,464$

Clifford operation cost not accounted.

Cost for physical noise rate 1%

Precision	# inputs
10^{-3}	5
10^{-5}	17
10^{-6}	28
10^{-8}	87
10^{-10}	139
10^{-12}	261
10^{-15}	436
10^{-18}	697
10^{-23}	1309
10^{-29}	2181
10^{-38}	3632
10^{-45}	6543

Meier, Eastin, & Knill

Precision	# T gates
10^{-3}	28
10^{-5}	132
10^{-7}	670
10^{-10}	3,284
10^{-15}	14,312
10^{-22}	74,162
10^{-33}	347,388
10^{-51}	1,692,692

Kliuchnikov, Maslov, & Mosca

- Need precision 10^{-14}
- Compiled logical gates will use 10^4 T gates
- Each of these T gates must have accuracy $\frac{10^{-14}}{10^4} = 10^{-18}$

Overhead = $14,312 \times 697 = 9,975,464$
Clifford operation cost not accounted.

Outline

High level state distillation

Our approach

- High level magic-state distillation $|Y_k\rangle = |Y_\theta\rangle$ with $\theta = 2\pi/2^k$.
- Corresponding rotations $R_k = R(2\pi/2^k)$.
    - T = R_3.
- Note that $R_{k-1}Z|Y_k\rangle = |Y_k\rangle$.
- Need R_{k-1} to distil gates $|Y_k\rangle$ (Gottesman-Chuang).
- Compiling becomes trivial (next slide).

- Landahl & Cesare used Reed-Muller codes to distill $|Y_k\rangle$
- We use the 4-qubit code in a similar way as Meier, Eastin, & Knill.

High level state distillation

Our approach

- High level magic-state distillation $|Y_k\rangle = |Y_\theta\rangle$ with $\theta = 2\pi/2^k$.
- Corresponding rotations $R_k = R(2\pi/2^k)$.
  - T = R_3.
- Note that $R_{k-1}Z|Y_k\rangle = |Y_k\rangle$.
- Need R_{k-1} to distil gates $|Y_k\rangle$ (Gottesman-Chuang).
- Compiling becomes trivial (next slide).

- Landahl & Cesare used Reed-Muller codes to distill $|Y_k\rangle$
- We use the 4-qubit code in a similar way as Meier, Eastin, & Knill.

High level state distillation

Our approach

- High level magic-state distillation $|Y_k\rangle = |Y_\theta\rangle$ with $\theta = 2\pi/2^k$.
- Corresponding rotations $R_k = R(2\pi/2^k)$.
  - T = R_3.
- Note that $R_{k-1}Z|Y_k\rangle = |Y_k\rangle$.
- Need R_{k-1} to distil gates $|Y_k\rangle$ (Gottesman-Chuang).
- Compiling becomes trivial (next slide).

- Landahl & Cesare used Reed-Muller codes to distill $|Y_k\rangle$
- We use the 4-qubit code in a similar way as Meier, Eastin, & Knill.

High level state distillation

Our approach

- High level magic-state distillation $|Y_k\rangle = |Y_\theta\rangle$ with $\theta = 2\pi/2^k$.
- Corresponding rotations $R_k = R(2\pi/2^k)$.
  - T = R_3.
- Note that $R_{k-1}Z|Y_k\rangle = |Y_k\rangle$.
- Need R_{k-1} to distil gates $|Y_k\rangle$ (Gottesman-Chuang).
- Compiling becomes trivial (next slide).

- Landahl & Cesare used Reed-Muller codes to distill $|Y_k\rangle$
- We use the 4-qubit code in a similar way as Meier, Eastin, & Knill.

High level state distillation

Our approach

- High level magic-state distillation $|Y_k\rangle = |Y_\theta\rangle$ with $\theta = 2\pi/2^k$.
- Corresponding rotations $R_k = R(2\pi/2^k)$.
  - T = R_3.
- Note that $R_{k-1}Z|Y_k\rangle = |Y_k\rangle$.
- Need R_{k-1} to distil gates $|Y_k\rangle$ (Gottesman-Chuang).
- Compiling becomes trivial (next slide).

- Landahl & Cesare used Reed-Muller codes to distill $|Y_k\rangle$
- We use the 4-qubit code in a similar way as Meier, Eastin, & Knill.

High level state distillation

Our approach

- High level magic-state distillation $|Y_k\rangle = |Y_\theta\rangle$ with $\theta = 2\pi/2^k$.
- Corresponding rotations $R_k = R(2\pi/2^k)$.
  - T = R_3.
- Note that $R_{k-1}Z|Y_k\rangle = |Y_k\rangle$.
- Need R_{k-1} to distil gates $|Y_k\rangle$ (Gottesman-Chuang).
- Compiling becomes trivial (next slide).

- Landahl & Cesare used Reed-Muller codes to distill $|Y_k\rangle$
- We use the 4-qubit code in a similar way as Meier, Eastin, & Knill.

High level state distillation

Our approach

- High level magic-state distillation $|Y_k\rangle = |Y_\theta\rangle$ with $\theta = 2\pi/2^k$.
- Corresponding rotations $R_k = R(2\pi/2^k)$.
  - T = R_3.
- Note that $R_{k-1}Z|Y_k\rangle = |Y_k\rangle$.
- Need R_{k-1} to distil gates $|Y_k\rangle$ (Gottesman-Chuang).
- Compiling becomes trivial (next slide).

- Landahl & Cesare used Reed-Muller codes to distill $|Y_k\rangle$
- We use the 4-qubit code in a similar way as Meier, Eastin, & Knill.

High level state distillation

Our approach

- High level magic-state distillation $|Y_k\rangle = |Y_\theta\rangle$ with $\theta = 2\pi/2^k$.
- Corresponding rotations $R_k = R(2\pi/2^k)$.
  - T = R_3.
- Note that $R_{k-1}Z|Y_k\rangle = |Y_k\rangle$.
- Need R_{k-1} to distil gates $|Y_k\rangle$ (Gottesman-Chuang).
- Compiling becomes trivial (next slide).

- Landahl & Cesare used Reed-Muller codes to distill $|Y_k\rangle$
- We use the 4-qubit code in a similar way as Meier, Eastin, & Knill.

High level state distillation

Our approach

- High level magic-state distillation $|Y_k\rangle = |Y_\theta\rangle$ with $\theta = 2\pi/2^k$.
- Corresponding rotations $R_k = R(2\pi/2^k)$.
  - T = R_3.
- Note that $R_{k-1}Z|Y_k\rangle = |Y_k\rangle$.
- Need R_{k-1} to distil gates $|Y_k\rangle$ (Gottesman-Chuang).
- Compiling becomes trivial (next slide).

- Landahl & Cesare used Reed-Muller codes to distill $|Y_k\rangle$
- We use the 4-qubit code in a similar way as Meier, Eastin, & Knill.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.000111010111000010100011110101111$$

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.00011101011100001010001111010111$$

$$\uparrow$$

$$R(2\pi/2^{32})$$

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.00011101011100001010001111010111$$

$$R(2\pi/2^{32}) \begin{cases} + \\ - \end{cases}$$

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.00011101011100001010001111010111$$

$$\uparrow$$

$$R(2\pi/2^{32}) \begin{cases} + \\ - \end{cases}$$

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.00011101011100001010001111010111$$

$R(2\pi/2^{32}) \begin{cases} + \\ - \end{cases}$

$R(2\pi/2^{31}) \begin{cases} + \\ - \end{cases}$

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.00011101011100001010001111010111$$

$$R(2\pi/2^{32}) \begin{cases} + \\ - \end{cases}$$

$$R(2\pi/2^{31}) \begin{cases} + \\ - \end{cases}$$

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.000111010111000010100011110\overset{10}{\cancel{01}}11$$

$$\uparrow\uparrow$$

$$R(2\pi/2^{32}) \begin{cases} + \\ - \end{cases}$$

$$R(2\pi/2^{31}) \begin{cases} + \\ - \end{cases}$$

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.0001110101110000101000111101\overset{10}{\cancel{01}}11$$

$R(2\pi/2^{32})$ $\begin{cases} + \\ - \end{cases}$

$R(2\pi/2^{31})$ $\begin{cases} + \\ - \end{cases}$

Nothing

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.0001110101110000101000111101\overset{10}{\cancel{01}}11$$

$$R(2\pi/2^{32}) \begin{cases} + \\ - \end{cases}$$

$$R(2\pi/2^{31}) \begin{cases} + \\ - \end{cases}$$

Nothing

$$R(2\pi/2^{29}) \begin{cases} + \\ - \end{cases}$$

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.0001110101110000101000111101\overset{10}{\cancel{01}}11$$

$$R(2\pi/2^{32}) \begin{cases} + \\ - \end{cases}$$

$$R(2\pi/2^{31}) \begin{cases} + \\ - \end{cases}$$

Nothing

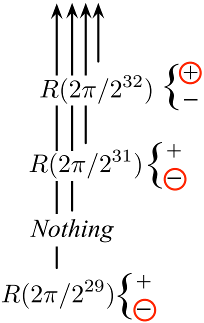
$$R(2\pi/2^{29}) \begin{cases} + \\ - \end{cases}$$

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.00011101011100001010001111010111$$

010



- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.0001110101110000101000111\overset{010}{\cancel{101}}11$$

$R(2\pi/2^{32}) \begin{cases} + \\ - \end{cases}$

$R(2\pi/2^{31}) \begin{cases} + \\ - \end{cases}$

Nothing

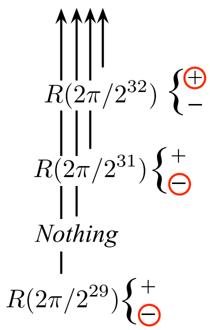
$R(2\pi/2^{29}) \begin{cases} + \\ - \end{cases}$

Clifford (deterministic)

- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

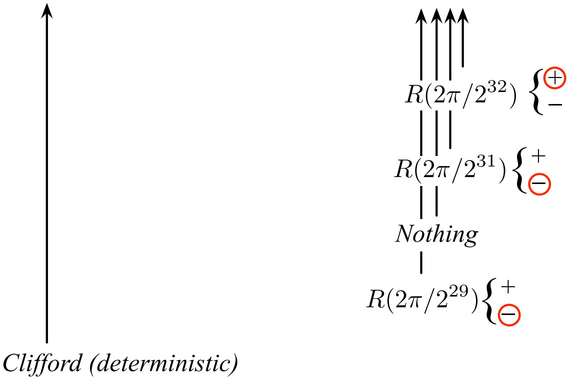
$$0.23\pi = 2\pi \times 0.0001110101110000101000111\overset{010}{\cancel{101}}11$$



- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

How to rotate a qubit by 0.23π?

$$0.23\pi = 2\pi \times 0.00011101011100001010001111\overset{010}{10}10111$$



- Rotate to precision 2^k with $k+1$ of these rotations.
- Get any single-qubit rotation by Euler angles decomposition.

Twirl

- For a qubit basis $\{|\psi\rangle, |\overline{\psi}\rangle\}$, twirl $T|\psi\rangle = |\psi\rangle$ and $T|\overline{\psi}\rangle = -|\overline{\psi}\rangle$
- Twirl makes matrices diagonal in $\{|\psi\rangle, |\overline{\psi}\rangle\}$ basis

$$\frac{1}{2}(\rho + T\rho T^{\dagger}) = \langle\psi|\rho|\psi\rangle \cdot |\psi\rangle\langle\psi| + \langle\overline{\psi}|\rho|\overline{\psi}\rangle \cdot |\overline{\psi}\rangle\langle\overline{\psi}|$$

- Twirl for $\{|0\rangle, |1\rangle\}$ is $T_0 = Z$.
- Twirl for $\{|Y_k\rangle, |\overline{Y}_k\rangle\}$ is $T_k = R_{k-1}Z$
- We can perform T_k with a $|Y_{k-1}\rangle$ state.

Twirl

- For a qubit basis $\{|\psi\rangle, |\overline{\psi}\rangle\}$, twirl $T|\psi\rangle = |\psi\rangle$ and $T|\overline{\psi}\rangle = -|\overline{\psi}\rangle$
- Twirl makes matrices diagonal in $\{|\psi\rangle, |\overline{\psi}\rangle\}$ basis

$$\frac{1}{2}(\rho + T\rho T^\dagger) = \langle\psi|\rho|\psi\rangle \cdot |\psi\rangle\langle\psi| + \langle\overline{\psi}|\rho|\overline{\psi}\rangle \cdot |\overline{\psi}\rangle\langle\overline{\psi}|$$

- Twirl for $\{|0\rangle, |1\rangle\}$ is $T_0 = Z$.
- Twirl for $\{|Y_k\rangle, |\overline{Y}_k\rangle\}$ is $T_k = R_{k-1}Z$
- We can perform T_k with a $|Y_{k-1}\rangle$ state.

Twirl

- For a qubit basis $\{|\psi\rangle, |\overline{\psi}\rangle\}$, twirl $T|\psi\rangle = |\psi\rangle$ and $T|\overline{\psi}\rangle = -|\overline{\psi}\rangle$
- Twirl makes matrices diagonal in $\{|\psi\rangle, |\overline{\psi}\rangle\}$ basis

$$\frac{1}{2}(\rho + T\rho T^\dagger) = \langle\psi|\rho|\psi\rangle \cdot |\psi\rangle\langle\psi| + \langle\overline{\psi}|\rho|\overline{\psi}\rangle \cdot |\overline{\psi}\rangle\langle\overline{\psi}|$$

- Twirl for $\{|0\rangle, |1\rangle\}$ is $T_0 = Z$.
- Twirl for $\{|Y_k\rangle, |\overline{Y}_k\rangle\}$ is $T_k = R_{k-1}Z$
- We can perform T_k with a $|Y_{k-1}\rangle$ state.

Twirl

- For a qubit basis $\{|\psi\rangle, |\overline{\psi}\rangle\}$, twirl $T|\psi\rangle = |\psi\rangle$ and $T|\overline{\psi}\rangle = -|\overline{\psi}\rangle$
- Twirl makes matrices diagonal in $\{|\psi\rangle, |\overline{\psi}\rangle\}$ basis

$$\frac{1}{2}(\rho + T\rho T^{\dagger}) = \langle\psi|\rho|\psi\rangle \cdot |\psi\rangle\langle\psi| + \langle\overline{\psi}|\rho|\overline{\psi}\rangle \cdot |\overline{\psi}\rangle\langle\overline{\psi}|$$

- Twirl for $\{|0\rangle, |1\rangle\}$ is $T_0 = Z$.
- Twirl for $\{|Y_k\rangle, |\overline{Y}_k\rangle\}$ is $T_k = R_{k-1}Z$
- We can perform T_k with a $|Y_{k-1}\rangle$ state.

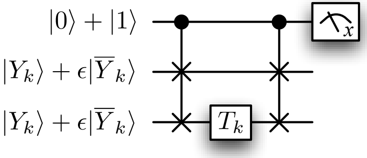
Twirl

- For a qubit basis $\{|\psi\rangle, |\overline{\psi}\rangle\}$, twirl $T|\psi\rangle = |\psi\rangle$ and $T|\overline{\psi}\rangle = -|\overline{\psi}\rangle$
- Twirl makes matrices diagonal in $\{|\psi\rangle, |\overline{\psi}\rangle\}$ basis

$$\frac{1}{2}(\rho + T\rho T^\dagger) = \langle\psi|\rho|\psi\rangle \cdot |\psi\rangle\langle\psi| + \langle\overline{\psi}|\rho|\overline{\psi}\rangle \cdot |\overline{\psi}\rangle\langle\overline{\psi}|$$

- Twirl for $\{|0\rangle, |1\rangle\}$ is $T_0 = Z$.
- Twirl for $\{|Y_k\rangle, |\overline{Y}_k\rangle\}$ is $T_k = R_{k-1}Z$
- We can perform T_k with a $|Y_{k-1}\rangle$ state.

Swap test



$$(|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle \xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle$$

$$\xrightarrow{T_k} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle$$

$$\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle$$

$$= (|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle$$

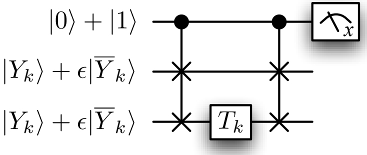
$$(|0\rangle + |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle \xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|\overline{Y}_k\rangle$$

$$\xrightarrow{T_k} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|Y_k\rangle|\overline{Y}_k\rangle$$

$$\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|\overline{Y}_k\rangle|Y_k\rangle$$

$$= (|0\rangle - |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle$$

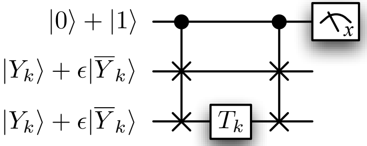
Swap test



$$
\begin{aligned}
(|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{T_k} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&= (|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle
\end{aligned}
$$

$$
\begin{aligned}
(|0\rangle + |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{T_k} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|\overline{Y}_k\rangle|Y_k\rangle \\
&= (|0\rangle - |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle
\end{aligned}
$$

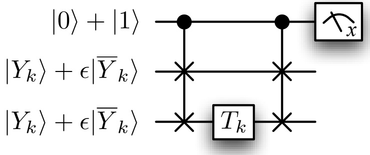
Swap test



$$
\begin{aligned}
(|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{T_k} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&= (|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle
\end{aligned}
$$

$$
\begin{aligned}
(|0\rangle + |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{T_k} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|\overline{Y}_k\rangle|Y_k\rangle \\
&= (|0\rangle - |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle
\end{aligned}
$$

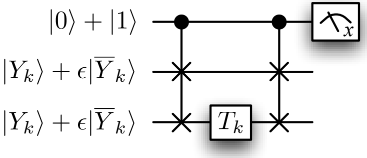
Swap test



$$
\begin{aligned}
(|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{T_k} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&= (|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle
\end{aligned}
$$

$$
\begin{aligned}
(|0\rangle + |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{T_k} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|\overline{Y}_k\rangle|Y_k\rangle \\
&= (|0\rangle - |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle
\end{aligned}
$$

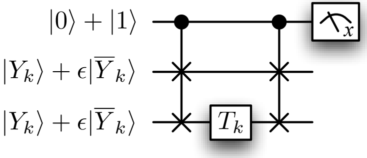
Swap test



$$
\begin{aligned}
(|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{T_k} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&= (|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle
\end{aligned}
$$

$$
\begin{aligned}
(|0\rangle + |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{T_k} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|\overline{Y}_k\rangle|Y_k\rangle \\
&= (|0\rangle - |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle
\end{aligned}
$$

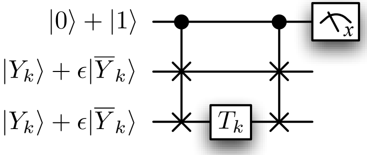
Swap test



$$\begin{aligned}
(|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{T_k} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&= (|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle
\end{aligned}$$

$$\begin{aligned}
(|0\rangle + |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{T_k} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|\overline{Y}_k\rangle|Y_k\rangle \\
&= (|0\rangle - |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle
\end{aligned}$$

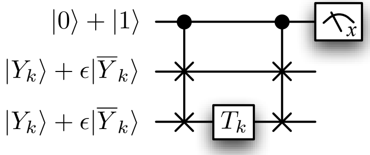
Swap test



$$
\begin{aligned}
(|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{T_k} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&= (|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle
\end{aligned}
$$

$$
\begin{aligned}
(|0\rangle + |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{T_k} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|\overline{Y}_k\rangle|Y_k\rangle \\
&= (|0\rangle - |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle
\end{aligned}
$$

Swap test



$$
\begin{aligned}
(|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{T_k} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|Y_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|Y_k\rangle \\
&= (|0\rangle + |1\rangle)|Y_k\rangle|Y_k\rangle
\end{aligned}
$$

$$
\begin{aligned}
(|0\rangle + |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle &\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle + |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{T_k} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|Y_k\rangle|\overline{Y}_k\rangle \\
&\xrightarrow{SWAP} |0\rangle|\overline{Y}_k\rangle|Y_k\rangle - |1\rangle|\overline{Y}_k\rangle|Y_k\rangle \\
&= (|0\rangle - |1\rangle)|\overline{Y}_k\rangle|Y_k\rangle
\end{aligned}
$$

Swap test

- Get outcome $+$ with probability $1 - 2\epsilon$.
- Given this outcome, state is $|Y_k\rangle|Y_k\rangle + \epsilon^2|\overline{Y}_k\rangle|\overline{Y}_k\rangle$.
- Get outcome $-$ with probability 2ϵ, reject the state.

Distillation

Can quadratically increase the fidelity of $|Y_k\rangle$ using T_k and controlled SWAP gate.

- Works as well for incoherent noise $|Y_k\rangle\langle Y_k| + \epsilon|\overline{Y}_k\rangle\langle\overline{Y}_k|$.
  - This can be obtained from any state by twirling.
- How to perform T_k?
  - Answer: By injecting distilled $|Y_{k-1}\rangle$ states.
- How to perform controlled-SWAP without introducing more errors?
  - Not a Clifford operation.

Swap test

- Get outcome $+$ with probability $1-2\epsilon$.
- Given this outcome, state is $|Y_k\rangle|Y_k\rangle + \epsilon^2|\overline{Y}_k\rangle|\overline{Y}_k\rangle$.
- Get outcome $-$ with probability 2ϵ, reject the state.

Distillation

Can quadratically increase the fidelity of $|Y_k\rangle$ using T_k and controlled SWAP gate.

- Works as well for incoherent noise $|Y_k\rangle\langle Y_k| + \epsilon|\overline{Y}_k\rangle\langle\overline{Y}_k|$.
  - This can be obtained from any state by twirling.
- How to perform T_k?
  - Answer: By injecting distilled $|Y_{k-1}\rangle$ states.
- How to perform controlled-SWAP without introducing more errors?
  - Not a Clifford operation.

Swap test

- Get outcome $+$ with probability $1-2\epsilon$.
- Given this outcome, state is $|Y_k\rangle|Y_k\rangle + \epsilon^2|\overline{Y}_k\rangle|\overline{Y}_k\rangle$.
- Get outcome $-$ with probability 2ϵ, reject the state.

Distillation

Can quadratically increase the fidelity of $|Y_k\rangle$ using T_k and controlled SWAP gate.

- Works as well for incoherent noise $|Y_k\rangle\langle Y_k| + \epsilon|\overline{Y}_k\rangle\langle\overline{Y}_k|$.
  - This can be obtained from any state by twirling.
- How to perform T_k?
  - Answer: By injecting distilled $|Y_{k-1}\rangle$ states.
- How to perform controlled-SWAP without introducing more errors?
  - Not a Clifford operation.

Swap test

- Get outcome $+$ with probability $1-2\epsilon$.
- Given this outcome, state is $|Y_k\rangle|Y_k\rangle + \epsilon^2|\overline{Y}_k\rangle|\overline{Y}_k\rangle$.
- Get outcome $-$ with probability 2ϵ, reject the state.

Distillation

Can quadratically increase the fidelity of $|Y_k\rangle$ using T_k and controlled SWAP gate.

- Works as well for incoherent noise $|Y_k\rangle\langle Y_k| + \epsilon|\overline{Y}_k\rangle\langle\overline{Y}_k|$.
  - This can be obtained from any state by twirling.
- How to perform T_k?
  - Answer: By injecting distilled $|Y_{k-1}\rangle$ states.
- How to perform controlled-SWAP without introducing more errors?
  - Not a Clifford operation.

Swap test

- Get outcome $+$ with probability $1 - 2\epsilon$.
- Given this outcome, state is $|Y_k\rangle|Y_k\rangle + \epsilon^2|\overline{Y}_k\rangle|\overline{Y}_k\rangle$.
- Get outcome $-$ with probability 2ϵ, reject the state.

Distillation

Can quadratically increase the fidelity of $|Y_k\rangle$ using T_k and controlled SWAP gate.

- Works as well for incoherent noise $|Y_k\rangle\langle Y_k| + \epsilon|\overline{Y}_k\rangle\langle\overline{Y}_k|$.
  - This can be obtained from any state by twirling.
- How to perform T_k?
  - Answer: By injecting distilled $|Y_{k-1}\rangle$ states.
- How to perform controlled-SWAP without introducing more errors?
  - Not a Clifford operation.

Swap test

- Get outcome $+$ with probability $1 - 2\epsilon$.
- Given this outcome, state is $|Y_k\rangle|Y_k\rangle + \epsilon^2|\overline{Y}_k\rangle|\overline{Y}_k\rangle$.
- Get outcome $-$ with probability 2ϵ, reject the state.

Distillation

Can quadratically increase the fidelity of $|Y_k\rangle$ using T_k and controlled SWAP gate.

- Works as well for incoherent noise $|Y_k\rangle\langle Y_k| + \epsilon|\overline{Y}_k\rangle\langle\overline{Y}_k|$.
  - This can be obtained from any state by twirling.
- How to perform T_k?
  - Answer: By injecting distilled $|Y_{k-1}\rangle$ states.
- How to perform controlled-SWAP without introducing more errors?
  - Not a Clifford operation.

Swap test

- Get outcome $+$ with probability $1-2\epsilon$.
- Given this outcome, state is $|Y_k\rangle|Y_k\rangle + \epsilon^2|\overline{Y}_k\rangle|\overline{Y}_k\rangle$.
- Get outcome $-$ with probability 2ϵ, reject the state.

Distillation

Can quadratically increase the fidelity of $|Y_k\rangle$ using T_k and controlled SWAP gate.

- Works as well for incoherent noise $|Y_k\rangle\langle Y_k| + \epsilon|\overline{Y}_k\rangle\langle\overline{Y}_k|$.
  - This can be obtained from any state by twirling.
- How to perform T_k?
  - Answer: By injecting distilled $|Y_{k-1}\rangle$ states.
- How to perform controlled-SWAP without introducing more errors?
  - Not a Clifford operation.

Swap test

- Get outcome $+$ with probability $1 - 2\epsilon$.
- Given this outcome, state is $|Y_k\rangle|Y_k\rangle + \epsilon^2|\overline{Y}_k\rangle|\overline{Y}_k\rangle$.
- Get outcome $-$ with probability 2ϵ, reject the state.

Distillation

Can quadratically increase the fidelity of $|Y_k\rangle$ using T_k and controlled SWAP gate.

- Works as well for incoherent noise $|Y_k\rangle\langle Y_k| + \epsilon|\overline{Y}_k\rangle\langle\overline{Y}_k|$.
  - This can be obtained from any state by twirling.
- How to perform T_k?
  - Answer: By injecting distilled $|Y_{k-1}\rangle$ states.
- How to perform controlled-SWAP without introducing more errors?
  - Not a Clifford operation.

Swap test

- Get outcome $+$ with probability $1-2\epsilon$.
- Given this outcome, state is $|Y_k\rangle|Y_k\rangle + \epsilon^2|\overline{Y}_k\rangle|\overline{Y}_k\rangle$.
- Get outcome $-$ with probability 2ϵ, reject the state.

Distillation

Can quadratically increase the fidelity of $|Y_k\rangle$ using T_k and controlled SWAP gate.

- Works as well for incoherent noise $|Y_k\rangle\langle Y_k| + \epsilon|\overline{Y}_k\rangle\langle\overline{Y}_k|$.
  - This can be obtained from any state by twirling.
- How to perform T_k?
  - Answer: By injecting distilled $|Y_{k-1}\rangle$ states.
- How to perform controlled-SWAP without introducing more errors?
  - Not a Clifford operation.

Swap test

- Get outcome $+$ with probability $1-2\epsilon$.
- Given this outcome, state is $|Y_k\rangle|Y_k\rangle + \epsilon^2|\overline{Y}_k\rangle|\overline{Y}_k\rangle$.
- Get outcome $-$ with probability 2ϵ, reject the state.

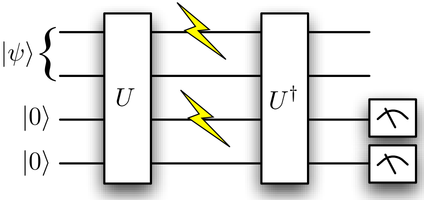
Distillation

Can quadratically increase the fidelity of $|Y_k\rangle$ using T_k and controlled SWAP gate.

- Works as well for incoherent noise $|Y_k\rangle\langle Y_k| + \epsilon|\overline{Y}_k\rangle\langle\overline{Y}_k|$.
  - This can be obtained from any state by twirling.
- How to perform T_k?
  - Answer: By injecting distilled $|Y_{k-1}\rangle$ states.
- How to perform controlled-SWAP without introducing more errors?
  - Not a Clifford operation.

4 qubit code

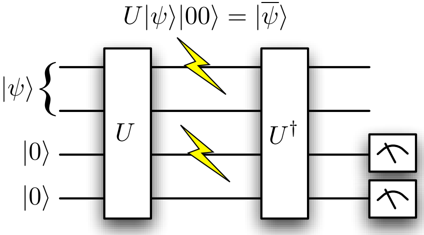
$$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$$



$$\overline{Z}_1 \equiv UZ_1U^\dagger = ZIIZ$$
$$\overline{X}_1 \equiv UX_1U^\dagger = XXII$$
$$\overline{Z}_2 \equiv UZ_2U^\dagger = XIIX$$
$$\overline{X}_2 \equiv UX_2U^\dagger = ZZII$$
$$S_1 \equiv UZ_3U^\dagger = ZZZZ$$
$$S_2 \equiv UZ_4U^\dagger = XXXX$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \to \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

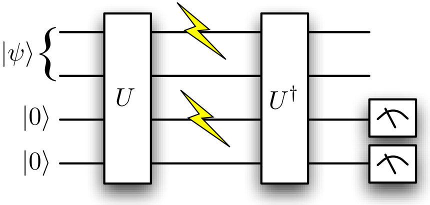


$$\overline{Z}_1 \equiv UZ_1U^\dagger = ZIIZ$$
$$\overline{X}_1 \equiv UX_1U^\dagger = XXII$$
$$\overline{Z}_2 \equiv UZ_2U^\dagger = XIIX$$
$$\overline{X}_2 \equiv UX_2U^\dagger = ZZII$$
$$S_1 \equiv UZ_3U^\dagger = ZZZZ$$
$$S_2 \equiv UZ_4U^\dagger = XXXX$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \to \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

$$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$$

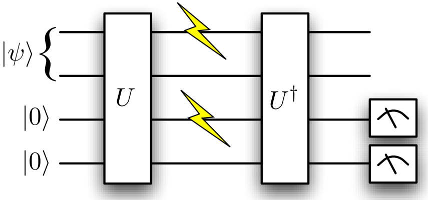


$$\begin{aligned}
\overline{Z}_1 &\equiv UZ_1U^\dagger = ZIIZ \\
\overline{X}_1 &\equiv UX_1U^\dagger = XXII \\
\overline{Z}_2 &\equiv UZ_2U^\dagger = XIIX \\
\overline{X}_2 &\equiv UX_2U^\dagger = ZZII \\
S_1 &\equiv UZ_3U^\dagger = ZZZZ \\
S_2 &\equiv UZ_4U^\dagger = XXXX
\end{aligned}$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \to \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

$$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$$

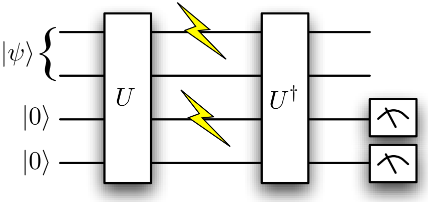


$$\begin{aligned}
\overline{Z}_1 &\equiv UZ_1U^\dagger = ZIIZ \\
\overline{X}_1 &\equiv UX_1U^\dagger = XXII \\
\overline{Z}_2 &\equiv UZ_2U^\dagger = XIIX \\
\overline{X}_2 &\equiv UX_2U^\dagger = ZZII \\
S_1 &\equiv UZ_3U^\dagger = ZZZZ \\
S_2 &\equiv UZ_4U^\dagger = XXXX
\end{aligned}$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \to \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1 H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1 H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2 H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2 H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

$$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$$

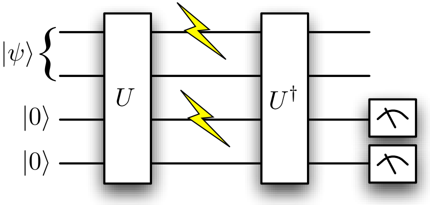


$$\begin{aligned}
\overline{Z}_1 &\equiv UZ_1U^\dagger = ZIIZ\\
\overline{X}_1 &\equiv UX_1U^\dagger = XXII\\
\overline{Z}_2 &\equiv UZ_2U^\dagger = XIIX\\
\overline{X}_2 &\equiv UX_2U^\dagger = ZZII\\
S_1 &\equiv UZ_3U^\dagger = ZZZZ\\
S_2 &\equiv UZ_4U^\dagger = XXXX
\end{aligned}$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \to \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

$$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$$

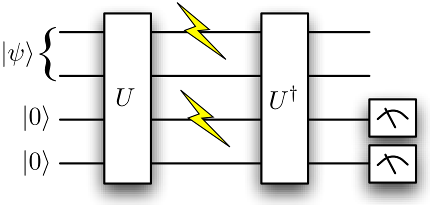


$$\overline{Z}_1 \equiv UZ_1U^\dagger = ZIIZ$$
$$\overline{X}_1 \equiv UX_1U^\dagger = XXII$$
$$\overline{Z}_2 \equiv UZ_2U^\dagger = XIIX$$
$$\overline{X}_2 \equiv UX_2U^\dagger = ZZII$$
$$S_1 \equiv UZ_3U^\dagger = ZZZZ$$
$$S_2 \equiv UZ_4U^\dagger = XXXX$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
    - Error rate $\epsilon \to \epsilon^2$.
- Apply H to all qubits maps S to itself.
    - $H^{\otimes 4}\overline{Z}_1H^{\otimes 4} = XIIX = \overline{Z}_2$
    - $H^{\otimes 4}\overline{X}_1H^{\otimes 4} = ZZII = \overline{X}_2$
    - $H^{\otimes 4}\overline{Z}_2H^{\otimes 4} = ZIIZ = \overline{Z}_1$
    - $H^{\otimes 4}\overline{X}_2H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$

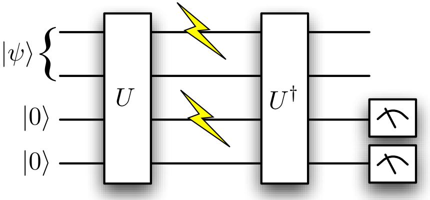


$$\begin{aligned}
\overline{Z}_1 &\equiv UZ_1U^\dagger = ZIIZ\\
\overline{X}_1 &\equiv UX_1U^\dagger = XXII\\
\overline{Z}_2 &\equiv UZ_2U^\dagger = XIIX\\
\overline{X}_2 &\equiv UX_2U^\dagger = ZZII\\
S_1 &\equiv UZ_3U^\dagger = ZZZZ\\
S_2 &\equiv UZ_4U^\dagger = XXXX
\end{aligned}$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \to \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$

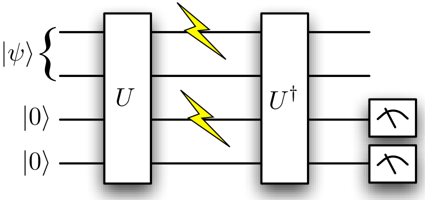


$$\overline{Z}_1 \equiv UZ_1U^\dagger = ZIIZ$$
$$\overline{X}_1 \equiv UX_1U^\dagger = XXII$$
$$\overline{Z}_2 \equiv UZ_2U^\dagger = XIIX$$
$$\overline{X}_2 \equiv UX_2U^\dagger = ZZII$$
$$S_1 \equiv UZ_3U^\dagger = ZZZZ$$
$$S_2 \equiv UZ_4U^\dagger = XXXX$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \to \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

$$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$$

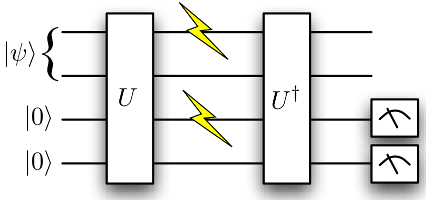


$$\overline{Z}_1 \equiv UZ_1U^\dagger = ZIIZ$$
$$\overline{X}_1 \equiv UX_1U^\dagger = XXII$$
$$\overline{Z}_2 \equiv UZ_2U^\dagger = XIIX$$
$$\overline{X}_2 \equiv UX_2U^\dagger = ZZII$$
$$S_1 \equiv UZ_3U^\dagger = ZZZZ$$
$$S_2 \equiv UZ_4U^\dagger = XXXX$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \rightarrow \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1 H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1 H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2 H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2 H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$

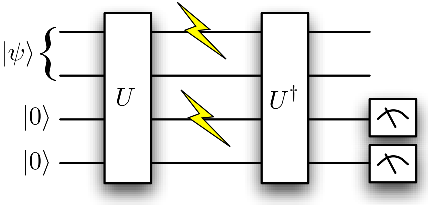


$$\overline{Z}_1 \equiv UZ_1U^\dagger = ZIIZ$$
$$\overline{X}_1 \equiv UX_1U^\dagger = XXII$$
$$\overline{Z}_2 \equiv UZ_2U^\dagger = XIIX$$
$$\overline{X}_2 \equiv UX_2U^\dagger = ZZII$$
$$S_1 \equiv UZ_3U^\dagger = ZZZZ$$
$$S_2 \equiv UZ_4U^\dagger = XXXX$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \rightarrow \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

$$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$$

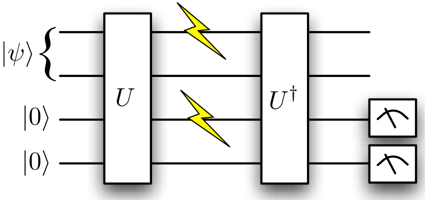


$$\begin{aligned}
\overline{Z}_1 &\equiv UZ_1U^\dagger = ZIIZ \\
\overline{X}_1 &\equiv UX_1U^\dagger = XXII \\
\overline{Z}_2 &\equiv UZ_2U^\dagger = XIIX \\
\overline{X}_2 &\equiv UX_2U^\dagger = ZZII \\
S_1 &\equiv UZ_3U^\dagger = ZZZZ \\
S_2 &\equiv UZ_4U^\dagger = XXXX
\end{aligned}$$

- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \to \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

4 qubit code

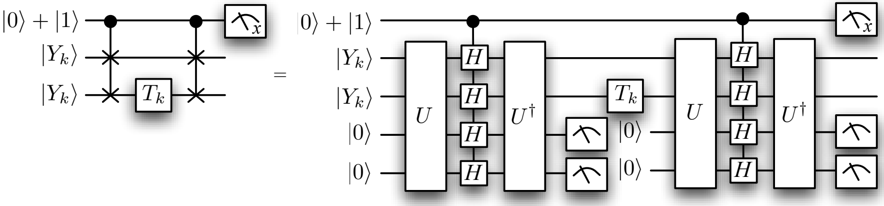
$$U|\psi\rangle|00\rangle = |\overline{\psi}\rangle$$



$$\overline{Z}_1 \equiv UZ_1U^\dagger = ZIIZ$$
$$\overline{X}_1 \equiv UX_1U^\dagger = XXII$$
$$\overline{Z}_2 \equiv UZ_2U^\dagger = XIIX$$
$$\overline{X}_2 \equiv UX_2U^\dagger = ZZII$$
$$S_1 \equiv UZ_3U^\dagger = ZZZZ$$
$$S_2 \equiv UZ_4U^\dagger = XXXX$$

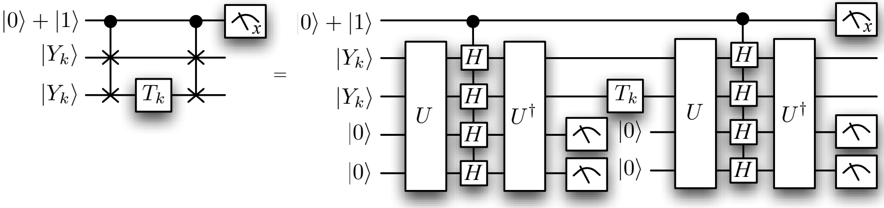
- U is Clifford.
- $Z|0\rangle = |0\rangle \Leftrightarrow S|\overline{\psi}\rangle = |\overline{\psi}\rangle$.
- This code detects all single-qubit errors.
  - Error rate $\epsilon \to \epsilon^2$.
- Apply H to all qubits maps S to itself.
  - $H^{\otimes 4}\overline{Z}_1 H^{\otimes 4} = XIIX = \overline{Z}_2$
  - $H^{\otimes 4}\overline{X}_1 H^{\otimes 4} = ZZII = \overline{X}_2$
  - $H^{\otimes 4}\overline{Z}_2 H^{\otimes 4} = ZIIZ = \overline{Z}_1$
  - $H^{\otimes 4}\overline{X}_2 H^{\otimes 4} = XXII = \overline{X}_2$
- Transversal Hadamard realizes SWAP.

Encoded SWAP test



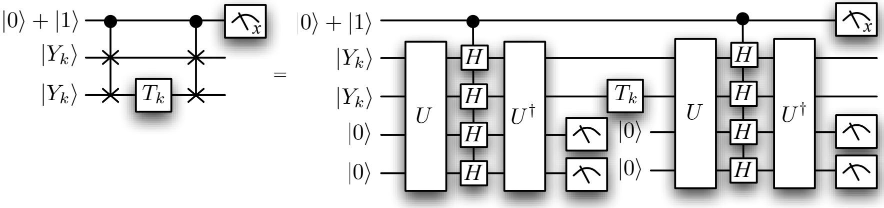
- How to realize controlled-H gates? (Not Clifford)
  - Note that $H = R(\pi/8)\, Z\, R(-\pi/8)$
  - So up to $\pi/8$ rotations, c-H = c-Z, the latter is Clifford.
  - $\pi/8$ rotations are obtained by injecting $|Y_3\rangle$.
- But $|Y_3\rangle$ gates are noisy!
  - This is OK, they are used inside an error correcting code.
  - We can use previously distilled $|Y_3\rangle$ states when distilling $|Y_k\rangle$.

Encoded SWAP test



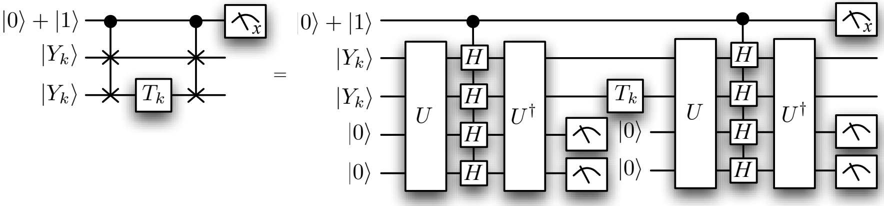
- How to realize controlled-H gates? (Not Clifford)
  - Note that $H = R(\pi/8)\, Z\, R(-\pi/8)$
  - So up to $\pi/8$ rotations, c-H = c-Z, the latter is Clifford.
  - $\pi/8$ rotations are obtained by injecting $|Y_3\rangle$.
- But $|Y_3\rangle$ gates are noisy!
  - This is OK, they are used inside an error correcting code.
  - We can use previously distilled $|Y_3\rangle$ states when distilling $|Y_k\rangle$.

Encoded SWAP test



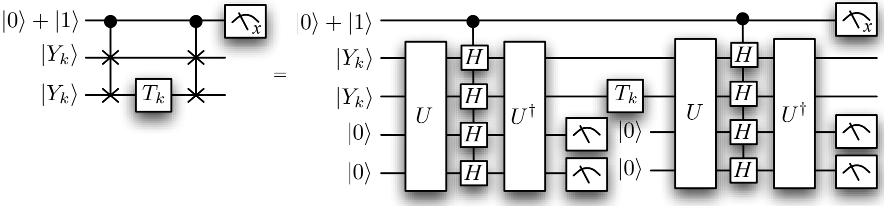
- How to realize controlled-H gates? (Not Clifford)
  - Note that $H = R(\pi/8)\, Z\, R(-\pi/8)$
  - So up to $\pi/8$ rotations, c-H = c-Z, the latter is Clifford.
  - $\pi/8$ rotations are obtained by injecting $|Y_3\rangle$.
- But $|Y_3\rangle$ gates are noisy!
  - This is OK, they are used inside an error correcting code.
  - We can use previously distilled $|Y_3\rangle$ states when distilling $|Y_k\rangle$.

Encoded SWAP test



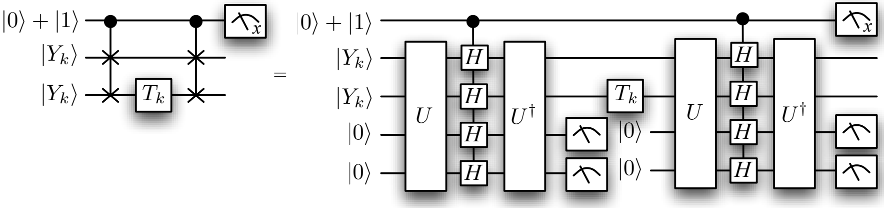
- How to realize controlled-H gates? (Not Clifford)
  - Note that $H = R(\pi/8)\, Z\, R(-\pi/8)$
  - So up to $\pi/8$ rotations, c-H = c-Z, the latter is Clifford.
  - $\pi/8$ rotations are obtained by injecting $|Y_3\rangle$.
- But $|Y_3\rangle$ gates are noisy!
  - This is OK, they are used inside an error correcting code.
  - We can use previously distilled $|Y_3\rangle$ states when distilling $|Y_k\rangle$.

Encoded SWAP test



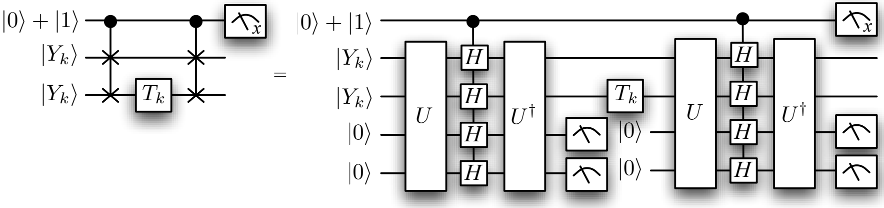
- How to realize controlled-H gates? (Not Clifford)
  - Note that $H = R(\pi/8)\, Z\, R(-\pi/8)$
  - So up to $\pi/8$ rotations, c-H = c-Z, the latter is Clifford.
  - $\pi/8$ rotations are obtained by injecting $|Y_3\rangle$.
- But $|Y_3\rangle$ gates are noisy!
  - This is OK, they are used inside an error correcting code.
  - We can use previously distilled $|Y_3\rangle$ states when distilling $|Y_k\rangle$.

Encoded SWAP test



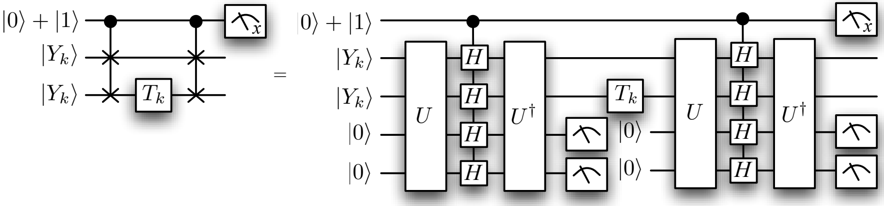
- How to realize controlled-H gates? (Not Clifford)
  - Note that $H = R(\pi/8)\, Z\, R(-\pi/8)$
  - So up to $\pi/8$ rotations, c-H = c-Z, the latter is Clifford.
  - $\pi/8$ rotations are obtained by injecting $|Y_3\rangle$.
- But $|Y_3\rangle$ gates are noisy!
  - This is OK, they are used inside an error correcting code.
  - We can use previously distilled $|Y_3\rangle$ states when distilling $|Y_k\rangle$.

Encoded SWAP test



- How to realize controlled-H gates? (Not Clifford)
  - Note that $H = R(\pi/8)\, Z\, R(-\pi/8)$
  - So up to $\pi/8$ rotations, c-H = c-Z, the latter is Clifford.
  - $\pi/8$ rotations are obtained by injecting $|Y_3\rangle$.
- But $|Y_3\rangle$ gates are noisy!
  - This is OK, they are used inside an error correcting code.
  - We can use previously distilled $|Y_3\rangle$ states when distilling $|Y_k\rangle$.

Encoded SWAP test



- How to realize controlled-H gates? (Not Clifford)
  - Note that $H = R(\pi/8)\, Z\, R(-\pi/8)$
  - So up to $\pi/8$ rotations, c-H = c-Z, the latter is Clifford.
  - $\pi/8$ rotations are obtained by injecting $|Y_3\rangle$.
- But $|Y_3\rangle$ gates are noisy!
  - This is OK, they are used inside an error correcting code.
  - We can use previously distilled $|Y_3\rangle$ states when distilling $|Y_k\rangle$.

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some “not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some “not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1-\epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some "not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some "not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1-\epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some "not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some "not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1 - \epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some “not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some “not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1-\epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some “not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some “not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1-\epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some “not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some “not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1-\epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some “not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some “not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1-\epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some "not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some "not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1-\epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some “not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some “not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1-\epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some “not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some “not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1 - \epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some “not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some “not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1-\epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some “not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some “not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1 - \epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Summary

To distill a $|Y_k\rangle$ state, we need...

- Some noisy versions of $|Y_k\rangle$.
- Some "not so noisy" versions of $|Y_{k-1}\rangle$ to implement the twirl T_k.
  - As used in the SWAP test.
  - To make the $|Y_k\rangle$ noise more diagonal if desired (not needed).
- Some "not so noisy" versions of $|Y_3\rangle$ to implement the c-H.

Given the accuracy $1-\epsilon_j$ of the inputs $|Y_j\rangle$, we can...

- Compute the accuracy of the distilled $|Y_k\rangle$.
- Compute the rejection probability (wasting these inputs)
  - Reject when the 4-qubit code detects an error.
  - Reject when the SWAP test fails.
- Knowing the rejection probability enables us to compute the average number of each component used.
- In our calculations, the accuracy of $|Y_j\rangle$ has two components (diagonal and off-diagonal).

Outline

Rule of thumb

- Distillation with perfect gates takes ϵ_k to $\epsilon'_k = c\epsilon_k^2$.
- The use of noisy $|Y_j\rangle$ to distill $|Y_k\rangle$ will deteriorate this ϵ'_k.
- How accurate should the $|Y_j\rangle$ be?
  - Having $\epsilon_j = 10^{-30}$ while attempting to distill $\epsilon'_k = 10^{-10}$ seems like an overkill, we've wasted our time obtaining very high-quality $|Y_j\rangle$.
  - If they are too noisy, distillation will be useless.
- If N_j states $|Y_j\rangle$ are used in the distillation, use $\epsilon_j \approx \epsilon'_k/N_j$.
  - This will roughly double ϵ'_k.
- This could be thoroughly optimized.

Rule of thumb

- Distillation with perfect gates takes ϵ_k to $\epsilon'_k = c\epsilon_k^2$.
- The use of noisy $|Y_j\rangle$ to distill $|Y_k\rangle$ will deteriorate this ϵ'_k.
- How accurate should the $|Y_j\rangle$ be?
  - Having $\epsilon_j = 10^{-30}$ while attempting to distill $\epsilon'_k = 10^{-10}$ seems like an overkill, we've wasted our time obtaining very high-quality $|Y_j\rangle$.
  - If they are too noisy, distillation will be useless.
- If N_j states $|Y_j\rangle$ are used in the distillation, use $\epsilon_j \approx \epsilon'_k/N_j$.
  - This will roughly double ϵ'_k.
- This could be thoroughly optimized.

Rule of thumb

- Distillation with perfect gates takes ϵ_k to $\epsilon'_k = c\epsilon_k^2$.
- The use of noisy $|Y_j\rangle$ to distill $|Y_k\rangle$ will deteriorate this ϵ'_k.
- How accurate should the $|Y_j\rangle$ be?
  - Having $\epsilon_j = 10^{-30}$ while attempting to distill $\epsilon'_k = 10^{-10}$ seems like an overkill, we've wasted our time obtaining very high-quality $|Y_j\rangle$.
  - If they are too noisy, distillation will be useless.
- If N_j states $|Y_j\rangle$ are used in the distillation, use $\epsilon_j \approx \epsilon'_k/N_j$.
  - This will roughly double ϵ'_k.
- This could be thoroughly optimized.

Rule of thumb

- Distillation with perfect gates takes ϵ_k to $\epsilon'_k = c\epsilon_k^2$.
- The use of noisy $|Y_j\rangle$ to distill $|Y_k\rangle$ will deteriorate this ϵ'_k.
- How accurate should the $|Y_j\rangle$ be?
  - Having $\epsilon_j = 10^{-30}$ while attempting to distill $\epsilon'_k = 10^{-10}$ seems like an overkill, we've wasted our time obtaining very high-quality $|Y_j\rangle$.
  - If they are too noisy, distillation will be useless.
- If N_j states $|Y_j\rangle$ are used in the distillation, use $\epsilon_j \approx \epsilon'_k / N_j$.
  - This will roughly double ϵ'_k.
- This could be thoroughly optimized.

Rule of thumb

- Distillation with perfect gates takes ϵ_k to $\epsilon'_k = c\epsilon_k^2$.
- The use of noisy $|Y_j\rangle$ to distill $|Y_k\rangle$ will deteriorate this ϵ'_k.
- How accurate should the $|Y_j\rangle$ be?
  - Having $\epsilon_j = 10^{-30}$ while attempting to distill $\epsilon'_k = 10^{-10}$ seems like an overkill, we've wasted our time obtaining very high-quality $|Y_j\rangle$.
  - If they are too noisy, distillation will be useless.
- If N_j states $|Y_j\rangle$ are used in the distillation, use $\epsilon_j \approx \epsilon'_k/N_j$.
  - This will roughly double ϵ'_k.
- This could be thoroughly optimized.

Rule of thumb

- Distillation with perfect gates takes ϵ_k to $\epsilon'_k = c\epsilon_k^2$.
- The use of noisy $|Y_j\rangle$ to distill $|Y_k\rangle$ will deteriorate this ϵ'_k.
- How accurate should the $|Y_j\rangle$ be?
  - Having $\epsilon_j = 10^{-30}$ while attempting to distill $\epsilon'_k = 10^{-10}$ seems like an overkill, we've wasted our time obtaining very high-quality $|Y_j\rangle$.
  - If they are too noisy, distillation will be useless.
- If N_j states $|Y_j\rangle$ are used in the distillation, use $\epsilon_j \approx \epsilon'_k/N_j$.
  - This will roughly double ϵ'_k.
- This could be thoroughly optimized.

Rule of thumb

- Distillation with perfect gates takes ϵ_k to $\epsilon'_k = c\epsilon_k^2$.
- The use of noisy $|Y_j\rangle$ to distill $|Y_k\rangle$ will deteriorate this ϵ'_k.
- How accurate should the $|Y_j\rangle$ be?
  - Having $\epsilon_j = 10^{-30}$ while attempting to distill $\epsilon'_k = 10^{-10}$ seems like an overkill, we've wasted our time obtaining very high-quality $|Y_j\rangle$.
  - If they are too noisy, distillation will be useless.
- If N_j states $|Y_j\rangle$ are used in the distillation, use $\epsilon_j \approx \epsilon'_k/N_j$.
  - This will roughly double ϵ'_k.
- This could be thoroughly optimized.

Rule of thumb

- Distillation with perfect gates takes ϵ_k to $\epsilon'_k = c\epsilon_k^2$.
- The use of noisy $|Y_j\rangle$ to distill $|Y_k\rangle$ will deteriorate this ϵ'_k.
- How accurate should the $|Y_j\rangle$ be?
  - Having $\epsilon_j = 10^{-30}$ while attempting to distill $\epsilon'_k = 10^{-10}$ seems like an overkill, we've wasted our time obtaining very high-quality $|Y_j\rangle$.
  - If they are too noisy, distillation will be useless.
- If N_j states $|Y_j\rangle$ are used in the distillation, use $\epsilon_j \approx \epsilon'_k/N_j$.
  - This will roughly double ϵ'_k.
- This could be thoroughly optimized.

Initial accuracy

- Standard assumption: $|Y_3\rangle$ is initially prepared with accuracy 1%, and distilled to any desired accuracy.
- How well should we assume $|Y_j\rangle$ can be prepared?
- Does it make sense to prepare $|Y_{10}\rangle \approx |0\rangle + 2^{-10}|1\rangle$ to accuracy 1%?
- May as well prepare $|0\rangle$.

Assumption on initial accuracy

- We use the scheme of Meier, Eastin, & Knill to prepare $|Y_3\rangle$ assuming an initial error of 1%.
- For all other $|Y_k\rangle$, $k > 3$, we initially prepare $|0\rangle \approx |Y_k\rangle$, which we can do to great accuracy (Clifford).
- As always, additional cost for Clifford operations is not accounted.

Initial accuracy

- Standard assumption: $|Y_3\rangle$ is initially prepared with accuracy 1%, and distilled to any desired accuracy.
- How well should we assume $|Y_j\rangle$ can be prepared?
- Does it make sense to prepare $|Y_{10}\rangle \approx |0\rangle + 2^{-10}|1\rangle$ to accuracy 1%?
- May as well prepare $|0\rangle$.

Assumption on initial accuracy

- We use the scheme of Meier, Eastin, & Knill to prepare $|Y_3\rangle$ assuming an initial error of 1%.
- For all other $|Y_k\rangle$, $k > 3$, we initially prepare $|0\rangle \approx |Y_k\rangle$, which we can do to great accuracy (Clifford).
- As always, additional cost for Clifford operations is not accounted.

Initial accuracy

- Standard assumption: $|Y_3\rangle$ is initially prepared with accuracy 1%, and distilled to any desired accuracy.
- How well should we assume $|Y_j\rangle$ can be prepared?
- Does it make sense to prepare $|Y_{10}\rangle \approx |0\rangle + 2^{-10}|1\rangle$ to accuracy 1%?
- May as well prepare $|0\rangle$.

Assumption on initial accuracy

- We use the scheme of Meier, Eastin, & Knill to prepare $|Y_3\rangle$ assuming an initial error of 1%.
- For all other $|Y_k\rangle$, $k > 3$, we initially prepare $|0\rangle \approx |Y_k\rangle$, which we can do to great accuracy (Clifford).
- As always, additional cost for Clifford operations is not accounted.

Initial accuracy

- Standard assumption: $|Y_3\rangle$ is initially prepared with accuracy 1%, and distilled to any desired accuracy.
- How well should we assume $|Y_j\rangle$ can be prepared?
- Does it make sense to prepare $|Y_{10}\rangle \approx |0\rangle + 2^{-10}|1\rangle$ to accuracy 1%?
- May as well prepare $|0\rangle$.

Assumption on initial accuracy

- We use the scheme of Meier, Eastin, & Knill to prepare $|Y_3\rangle$ assuming an initial error of 1%.
- For all other $|Y_k\rangle$, $k > 3$, we initially prepare $|0\rangle \approx |Y_k\rangle$, which we can do to great accuracy (Clifford).
- As always, additional cost for Clifford operations is not accounted.

Initial accuracy

- Standard assumption: $|Y_3\rangle$ is initially prepared with accuracy 1%, and distilled to any desired accuracy.
- How well should we assume $|Y_j\rangle$ can be prepared?
- Does it make sense to prepare $|Y_{10}\rangle \approx |0\rangle + 2^{-10}|1\rangle$ to accuracy 1%?
- May as well prepare $|0\rangle$.

Assumption on initial accuracy

- We use the scheme of Meier, Eastin, & Knill to prepare $|Y_3\rangle$ assuming an initial error of 1%.
- For all other $|Y_k\rangle$, $k > 3$, we initially prepare $|0\rangle \approx |Y_k\rangle$, which we can do to great accuracy (Clifford).
- As always, additional cost for Clifford operations is not accounted.

Initial accuracy

- Standard assumption: $|Y_3\rangle$ is initially prepared with accuracy 1%, and distilled to any desired accuracy.
- How well should we assume $|Y_j\rangle$ can be prepared?
- Does it make sense to prepare $|Y_{10}\rangle \approx |0\rangle + 2^{-10}|1\rangle$ to accuracy 1%?
- May as well prepare $|0\rangle$.

Assumption on initial accuracy

- We use the scheme of Meier, Eastin, & Knill to prepare $|Y_3\rangle$ assuming an initial error of 1%.
- For all other $|Y_k\rangle$, $k > 3$, we initially prepare $|0\rangle \approx |Y_k\rangle$, which we can do to great accuracy (Clifford).
- As always, additional cost for Clifford operations is not accounted.

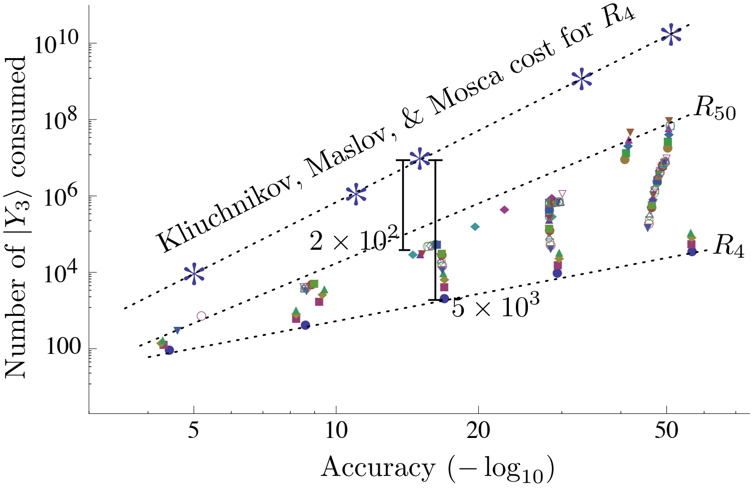
Initial accuracy

- Standard assumption: $|Y_3\rangle$ is initially prepared with accuracy 1%, and distilled to any desired accuracy.
- How well should we assume $|Y_j\rangle$ can be prepared?
- Does it make sense to prepare $|Y_{10}\rangle \approx |0\rangle + 2^{-10}|1\rangle$ to accuracy 1%?
- May as well prepare $|0\rangle$.

Assumption on initial accuracy

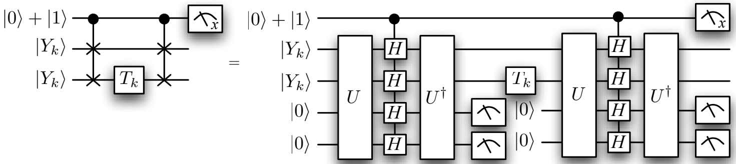
- We use the scheme of Meier, Eastin, & Knill to prepare $|Y_3\rangle$ assuming an initial error of 1%.
- For all other $|Y_k\rangle$, $k > 3$, we initially prepare $|0\rangle \approx |Y_k\rangle$, which we can do to great accuracy (Clifford).
- As always, additional cost for Clifford operations is not accounted.

Results: cost of realizing R_k



Outline

High rate generalization



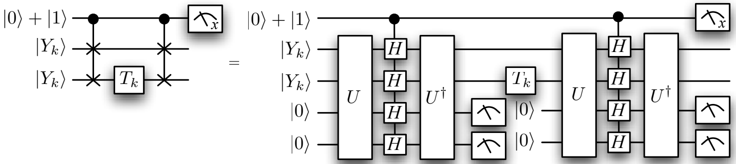
- Requires 4 c-H gates to distill 2 states $|Y_k\rangle$ (per round).
- Gates c-H are realized by injecting states $|Y_3\rangle$: rate 1/2.
- Higher rate encoding?
- Use $[[n = 2m+2, k = 2m, d = 2]]$ with the property that $H^{\otimes n}$ SWAPs qubit j with $m + j$.

$$S_0 = \prod_i X_i, \quad S_1 = \prod_i Z_i$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} Z_i, \quad \overline{X}_j = X_{2j+1}X_{2j+2} \qquad \text{for } j = 0, 1, \ldots m-1$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} X_i, \quad \overline{X}_j = Z_{2j+1}Z_{2j+2} \quad \text{for } j = m, m+1, \ldots 2m-1$$

High rate generalization



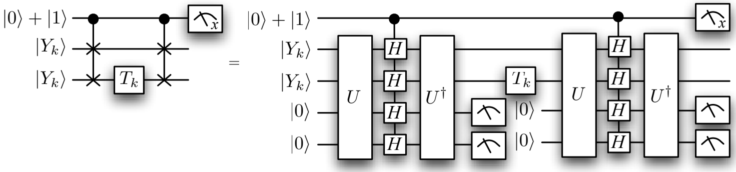
- Requires 4 c-H gates to distill 2 states $|Y_k\rangle$ (per round).
- Gates c-H are realized by injecting states $|Y_3\rangle$: rate 1/2.
- Higher rate encoding?
- Use $[[n = 2m+2, k = 2m, d = 2]]$ with the property that $H^{\otimes n}$ SWAPs qubit j with $m + j$.

$$S_0 = \prod_i X_i, \quad S_1 = \prod_i Z_i$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} Z_i, \quad \overline{X}_j = X_{2j+1}X_{2j+2} \qquad \text{for } j = 0, 1, \ldots m-1$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} X_i, \quad \overline{X}_j = Z_{2j+1}Z_{2j+2} \quad \text{for } j = m, m+1, \ldots 2m-1$$

High rate generalization



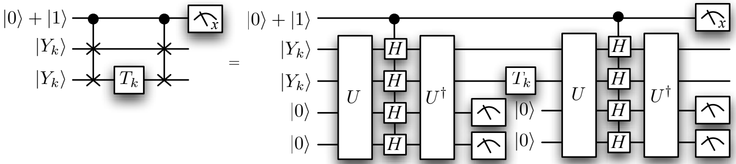
- Requires 4 c-H gates to distill 2 states $|Y_k\rangle$ (per round).
- Gates c-H are realized by injecting states $|Y_3\rangle$: rate 1/2.
- Higher rate encoding?
- Use $[[n = 2m + 2, k = 2m, d = 2]]$ with the property that $H^{\otimes n}$ SWAPs qubit j with $m + j$.

$$S_0 = \prod_i X_i, \quad S_1 = \prod_i Z_i$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} Z_i, \quad \overline{X}_j = X_{2j+1} X_{2j+2} \quad \text{for } j = 0, 1, \ldots m-1$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} X_i, \quad \overline{X}_j = Z_{2j+1} Z_{2j+2} \quad \text{for } j = m, m+1, \ldots 2m-1$$

High rate generalization



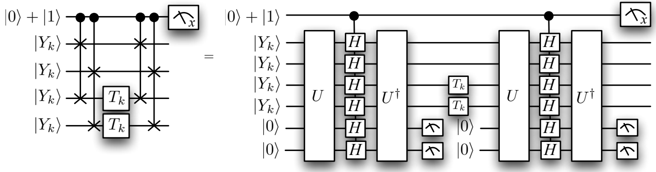
- Requires 4 c-H gates to distill 2 states $|Y_k\rangle$ (per round).
- Gates c-H are realized by injecting states $|Y_3\rangle$: rate 1/2.
- Higher rate encoding?
- Use $[[n = 2m+2, k = 2m, d = 2]]$ with the property that $H^{\otimes n}$ SWAPs qubit j with $m+j$.

$$S_0 = \prod_i X_i, \quad S_1 = \prod_i Z_i$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} Z_i, \quad \overline{X}_j = X_{2j+1}X_{2j+2} \qquad \text{for } j = 0, 1, \ldots m-1$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} X_i, \quad \overline{X}_j = Z_{2j+1}Z_{2j+2} \quad \text{for } j = m, m+1, \ldots 2m-1$$

High rate generalization



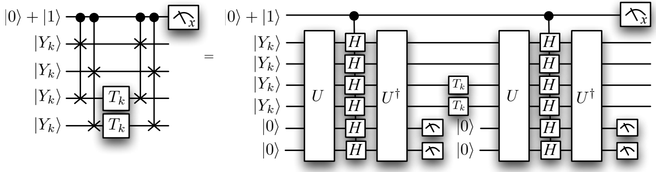
- Requires 4 c-H gates to distill 2 states $|Y_k\rangle$ (per round).
- Gates c-H are realized by injecting states $|Y_3\rangle$: rate 1/2.
- Higher rate encoding?
- Use $[[n = 2m + 2, k = 2m, d = 2]]$ with the property that $H^{\otimes n}$ SWAPs qubit j with $m + j$.

$$S_0 = \prod_i X_i, \quad S_1 = \prod_i Z_i$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} Z_i, \quad \overline{X}_j = X_{2j+1}X_{2j+2} \qquad \text{for } j = 0, 1, \ldots m-1$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} X_i, \quad \overline{X}_j = Z_{2j+1}Z_{2j+2} \quad \text{for } j = m, m+1, \ldots 2m-1$$

High rate generalization



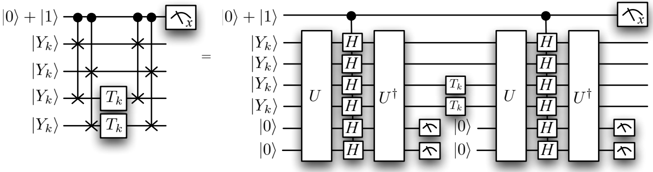
- Requires 4 c-H gates to distill 2 states $|Y_k\rangle$ (per round).
- Gates c-H are realized by injecting states $|Y_3\rangle$: rate 1/2.
- Higher rate encoding?
- Use $[[n = 2m+2, k = 2m, d = 2]]$ with the property that $H^{\otimes n}$ SWAPs qubit j with $m+j$.

$$S_0 = \prod_i X_i, \quad S_1 = \prod_i Z_i$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} Z_i, \quad \overline{X}_j = X_{2j+1}X_{2j+2} \qquad \text{for } j = 0, 1, \ldots m-1$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} X_i, \quad \overline{X}_j = Z_{2j+1}Z_{2j+2} \quad \text{for } j = m, m+1, \ldots 2m-1$$

High rate generalization



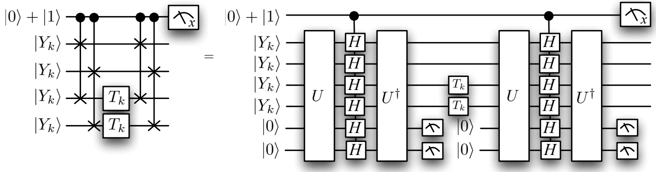
- Requires 4 c-H gates to distill 2 states $|Y_k\rangle$ (per round).
- Gates c-H are realized by injecting states $|Y_3\rangle$: rate 1/2.
- Higher rate encoding?
- Use $[[n = 2m+2, k = 2m, d = 2]]$ with the property that $H^{\otimes n}$ SWAPs qubit j with $m+j$.

$$S_0 = \prod_i X_i, \quad S_1 = \prod_i Z_i$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} Z_i, \quad \overline{X}_j = X_{2j+1}X_{2j+2} \qquad \text{for } j = 0, 1, \ldots m-1$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} X_i, \quad \overline{X}_j = Z_{2j+1}Z_{2j+2} \quad \text{for } j = m, m+1, \ldots 2m-1$$

High rate generalization



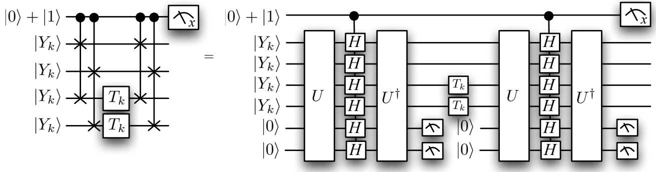
- Requires 4 c-H gates to distill 2 states $|Y_k\rangle$ (per round).
- Gates c-H are realized by injecting states $|Y_3\rangle$: rate 1/2.
- Higher rate encoding?
- Use $[[n = 2m+2, k = 2m, d = 2]]$ with the property that $H^{\otimes n}$ SWAPs qubit j with $m + j$.

$$S_0 = \prod_i X_i, \quad S_1 = \prod_i Z_i$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} Z_i, \quad \overline{X}_j = X_{2j+1}X_{2j+2} \qquad \text{for } j = 0, 1, \ldots m-1$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} X_i, \quad \overline{X}_j = Z_{2j+1}Z_{2j+2} \quad \text{for } j = m, m+1, \ldots 2m-1$$

High rate generalization



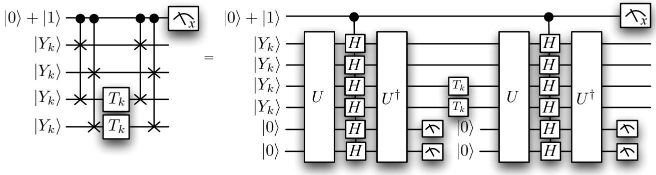
- Requires 4 c-H gates to distill 2 states $|Y_k\rangle$ (per round).
- Gates c-H are realized by injecting states $|Y_3\rangle$: rate 1/2.
- Higher rate encoding?
- Use $[[n = 2m+2, k = 2m, d = 2]]$ with the property that $H^{\otimes n}$ SWAPs qubit j with $m+j$.

$$S_0 = \prod_i X_i, \quad S_1 = \prod_i Z_i$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} Z_i, \quad \overline{X}_j = X_{2j+1}X_{2j+2} \qquad \text{for } j = 0, 1, \ldots m-1$$

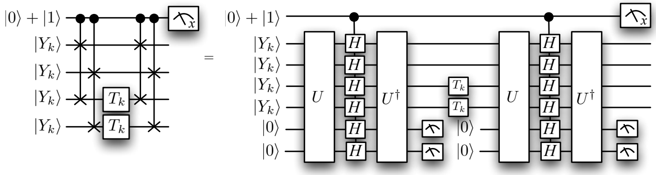
$$\overline{Z}_j = \prod_{i=0}^{2j+1} X_i, \quad \overline{X}_j = Z_{2j+1}Z_{2j+2} \quad \text{for } j = m, m+1, \ldots 2m-1$$

High rate generalization



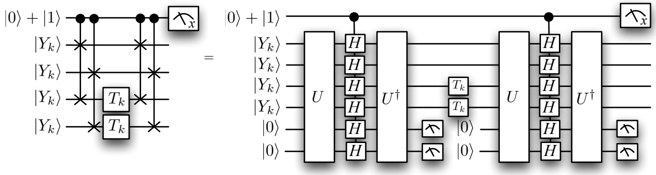
- Rate $= \frac{m}{m+1} \to 1$.
- Error suppression $\epsilon \to c\epsilon^2$
  - $c \propto m^2$ since there are more locations for failures.
- Rejection probability $p \propto m\epsilon$
  - Increase m along the distillation flow.
- Gain is more that just factor of 2 because of the recursive nature of the distillation.
- Enables to smoothly vary target accuracy.

High rate generalization



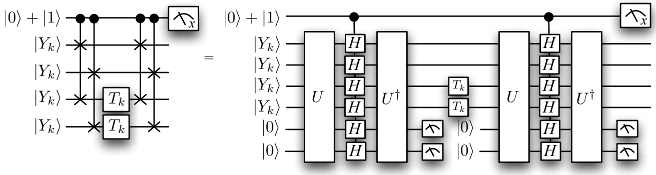
- Rate $= \frac{m}{m+1} \to 1$.
- Error suppression $\epsilon \to c\epsilon^2$
  - $c \propto m^2$ since there are more locations for failures.
- Rejection probability $p \propto m\epsilon$
  - Increase m along the distillation flow.
- Gain is more that just factor of 2 because of the recursive nature of the distillation.
- Enables to smoothly vary target accuracy.

High rate generalization



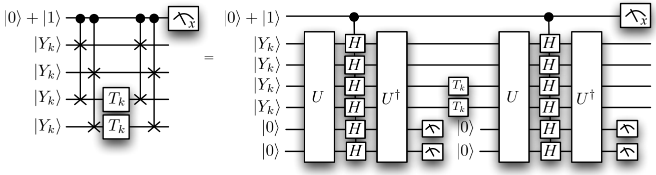
- Rate $= \frac{m}{m+1} \to 1$.
- Error suppression $\epsilon \to c\epsilon^2$
  - $c \propto m^2$ since there are more locations for failures.
- Rejection probability $p \propto m\epsilon$
  - Increase m along the distillation flow.
- Gain is more that just factor of 2 because of the recursive nature of the distillation.
- Enables to smoothly vary target accuracy.

High rate generalization



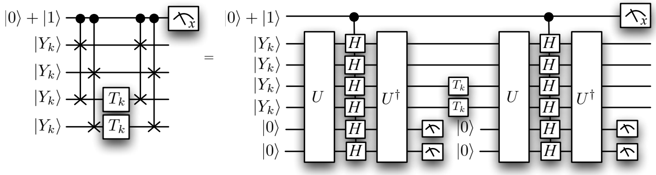
- Rate $= \frac{m}{m+1} \to 1$.
- Error suppression $\epsilon \to c\epsilon^2$
  - $c \propto m^2$ since there are more locations for failures.
- Rejection probability $p \propto m\epsilon$
  - Increase m along the distillation flow.
- Gain is more that just factor of 2 because of the recursive nature of the distillation.
- Enables to smoothly vary target accuracy.

High rate generalization



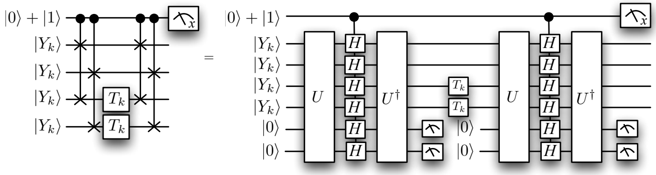
- Rate $= \frac{m}{m+1} \to 1$.
- Error suppression $\epsilon \to c\epsilon^2$
  - $c \propto m^2$ since there are more locations for failures.
- Rejection probability $p \propto m\epsilon$
  - Increase m along the distillation flow.
- Gain is more that just factor of 2 because of the recursive nature of the distillation.
- Enables to smoothly vary target accuracy.

High rate generalization



- Rate $= \frac{m}{m+1} \to 1$.
- Error suppression $\epsilon \to c\epsilon^2$
  - $c \propto m^2$ since there are more locations for failures.
- Rejection probability $p \propto m\epsilon$
  - Increase m along the distillation flow.
- Gain is more that just factor of 2 because of the recursive nature of the distillation.
- Enables to smoothly vary target accuracy.

High rate generalization



- Rate $= \frac{m}{m+1} \to 1.$
- Error suppression $\epsilon \to c\epsilon^2$
  - $c \propto m^2$ since there are more locations for failures.
- Rejection probability $p \propto m\epsilon$
  - Increase m along the distillation flow.
- Gain is more that just factor of 2 because of the recursive nature of the distillation.
- Enables to smoothly vary target accuracy.

Conclusion & Outlook

- Distillation of complex magic states to circumvent Solovay-Kitaev-like compiling.
  - Found savings of a 2-4 orders of magnitude for relevant noise regimes.
- Start with better approximations: $|0\rangle$ is very close to $|Y_k\rangle$ for $k \gg 1$.
- Add cost of Clifford gates.
  - Not simply multiplicative: required accuracy changes along the distillation flow.
- Rate 1 code for controlled SWAP.
- Use proper optimization instead of rule of thumb.

Conclusion & Outlook

- Distillation of complex magic states to circumvent Solovay-Kitaev-like compiling.
  - Found savings of a 2-4 orders of magnitude for relevant noise regimes.
- Start with better approximations: $|0\rangle$ is very close to $|Y_k\rangle$ for $k \gg 1$.
- Add cost of Clifford gates.
  - Not simply multiplicative: required accuracy changes along the distillation flow.
- Rate 1 code for controlled SWAP.
- Use proper optimization instead of rule of thumb.

Conclusion & Outlook

- Distillation of complex magic states to circumvent Solovay-Kitaev-like compiling.
  - Found savings of a 2-4 orders of magnitude for relevant noise regimes.
- Start with better approximations: $|0\rangle$ is very close to $|Y_k\rangle$ for $k \gg 1$.
- Add cost of Clifford gates.
  - Not simply multiplicative: required accuracy changes along the distillation flow.
- Rate 1 code for controlled SWAP.
- Use proper optimization instead of rule of thumb.

Conclusion & Outlook

- Distillation of complex magic states to circumvent Solovay-Kitaev-like compiling.
  - Found savings of a 2-4 orders of magnitude for relevant noise regimes.
- Start with better approximations: $|0\rangle$ is very close to $|Y_k\rangle$ for $k \gg 1$.
- Add cost of Clifford gates.
  - Not simply multiplicative: required accuracy changes along the distillation flow.
- Rate 1 code for controlled SWAP.
- Use proper optimization instead of rule of thumb.

Conclusion & Outlook

- Distillation of complex magic states to circumvent Solovay-Kitaev-like compiling.
  - Found savings of a 2-4 orders of magnitude for relevant noise regimes.
- Start with better approximations: $|0\rangle$ is very close to $|Y_k\rangle$ for $k \gg 1$.
- Add cost of Clifford gates.
  - Not simply multiplicative: required accuracy changes along the distillation flow.
- Rate 1 code for controlled SWAP.
- Use proper optimization instead of rule of thumb.

Conclusion & Outlook

- Distillation of complex magic states to circumvent Solovay-Kitaev-like compiling.
  - Found savings of a 2-4 orders of magnitude for relevant noise regimes.
- Start with better approximations: $|0\rangle$ is very close to $|Y_k\rangle$ for $k \gg 1$.
- Add cost of Clifford gates.
  - Not simply multiplicative: required accuracy changes along the distillation flow.
- Rate 1 code for controlled SWAP.
- Use proper optimization instead of rule of thumb.

Conclusion & Outlook

- Distillation of complex magic states to circumvent Solovay-Kitaev-like compiling.
  - Found savings of a 2-4 orders of magnitude for relevant noise regimes.
- Start with better approximations: $|0\rangle$ is very close to $|Y_k\rangle$ for $k \gg 1$.
- Add cost of Clifford gates.
  - Not simply multiplicative: required accuracy changes along the distillation flow.
- Rate 1 code for controlled SWAP.
- Use proper optimization instead of rule of thumb.