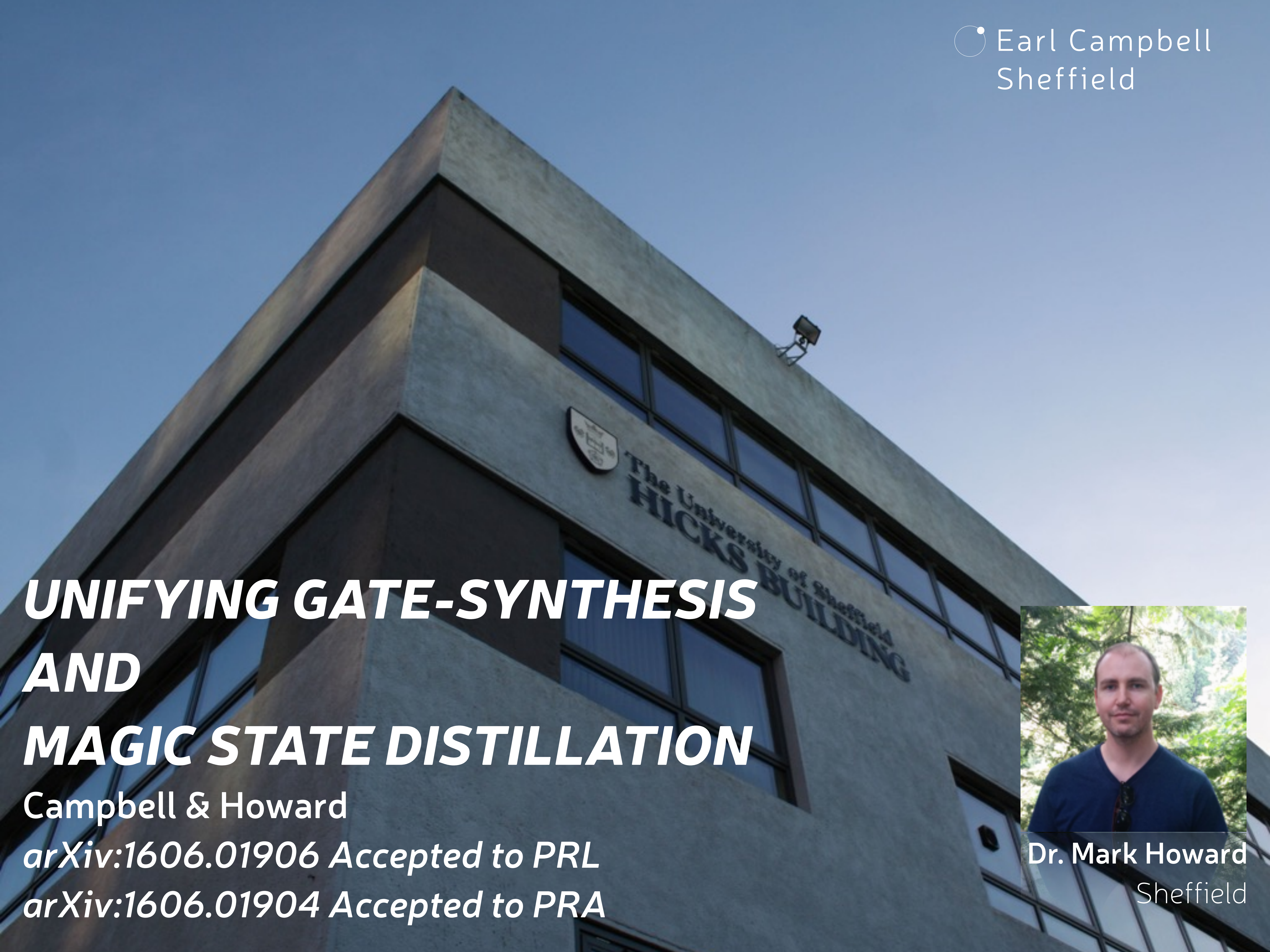



Earl Campbell
Sheffield

# *UNIFYING GATE-SYNTHESIS AND MAGIC STATE DISTILLATION*

Campbell & Howard

*arXiv:1606.01906 Accepted to PRL*
*arXiv:1606.01904 Accepted to PRA*

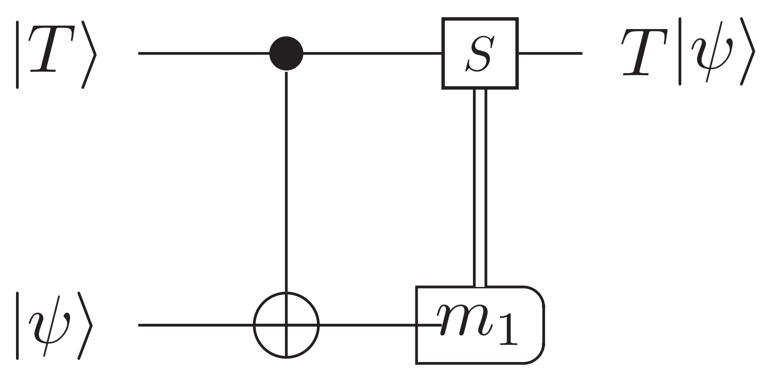
**Dr. Mark Howard**
Sheffield

**1.** Build quantum memories with reliable Clifford gates

**2.** Distill T-magic states

$$|T\rangle \propto |0\rangle + e^{i\pi/4}|1\rangle \qquad |T\rangle \propto T|+\rangle$$

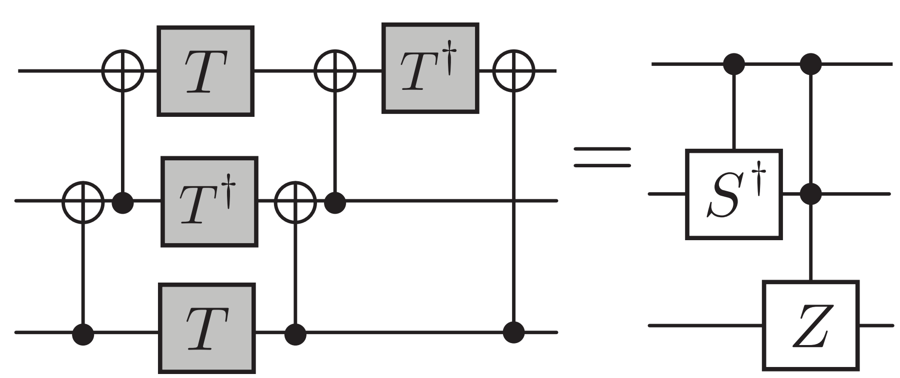
**3.** Inject to get T-gate




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

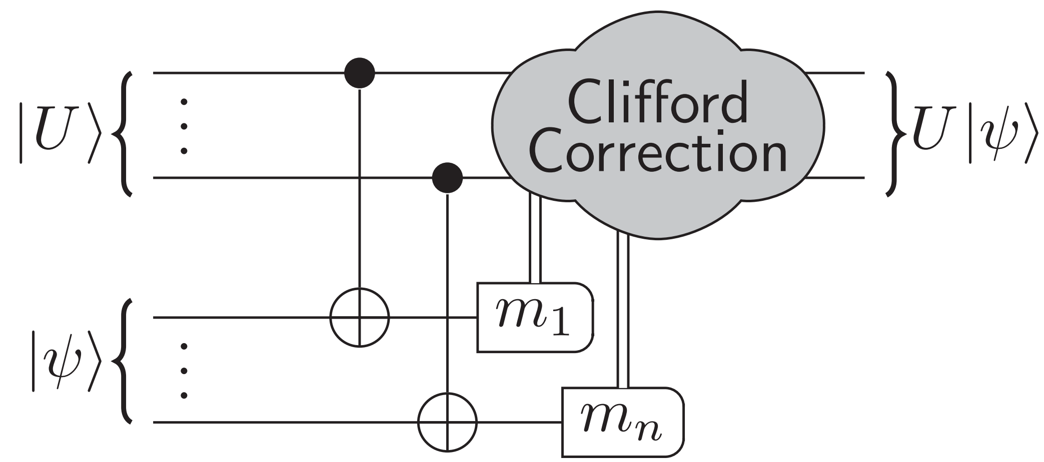
**4.** Compose Clifford+T gates to synthesize circuit

**e.g**



**Generalised state injection**

Gottesman and Chuang, *Nature* **402**, 390-393 (1999)




where $|U\rangle = U|+\rangle^{\otimes k}$ is an exotic magic state

For diagonal gates in 3rd level of Clifford hierarchy

$UPU^{\dagger} = \mathcal{C}$ $\mathcal{C} :=$ Clifford group $\mathcal{P} :=$ Pauli group

examples later!

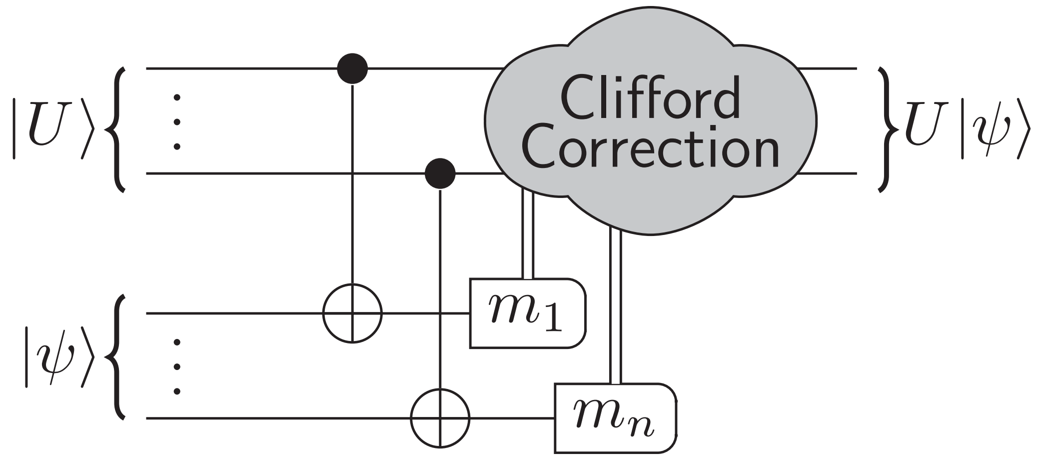
**1.** Build quantum memories with reliable Clifford gates

**2.** Distill exotic magic states

? ? ? ? ? ?

$$|U\rangle = U|+\rangle^{\otimes k}$$

**3.** Inject to large chunks of circuits




**4.** Compose into larger circuits (less synthesis required)

1 **Review background ideas**

2 **Preparation of $|U\rangle$ magic states**

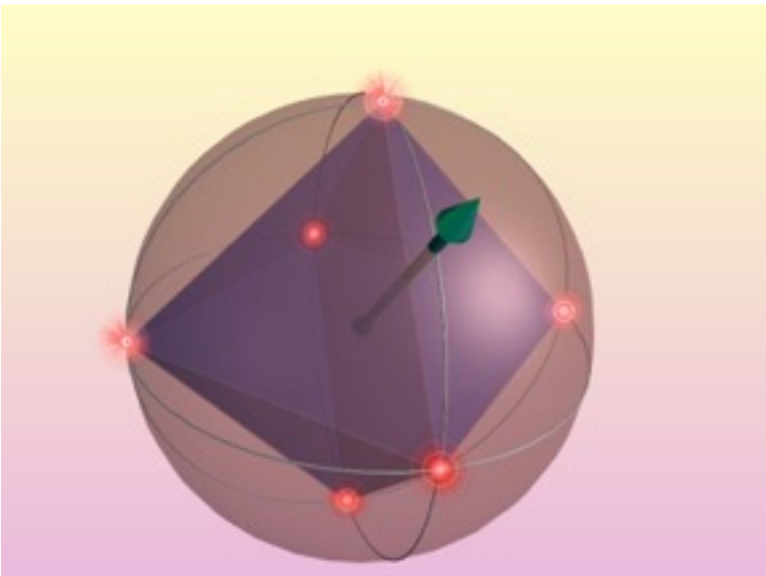
3 **Resource comparison (**factor ~3 better**)**

4 **Bonus results on gate synthesis**

# Background

Gate synthesis

**Clifford group:** "easy" to implement in many quantum codes **Cost: 1$**


The Hadamard

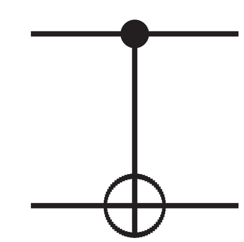
$$HZH^\dagger = X$$
$$HXH^\dagger = Z$$
$$HYH^\dagger = -Y$$

**Generated by**

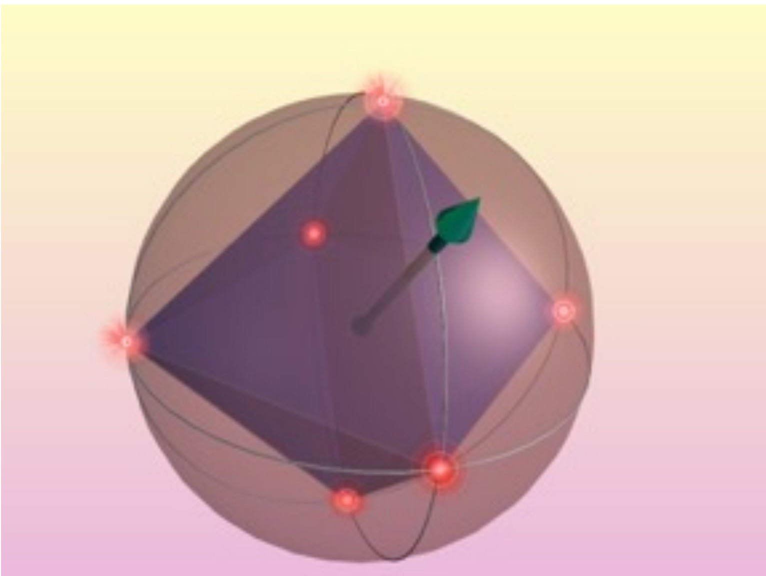
Hadamard $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ $\boxed{H}$

S-gate $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ $\boxed{S}$

Control-NOT $C_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$



**Clifford group:** "easy" to implement in many quantum codes **Cost: 1$**



The Hadamard

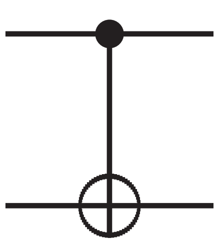
**Generated by**

Hadamard $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ $\boxed{H}$

S-gate $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ $\boxed{S}$

$$\begin{aligned} HZH^\dagger &= X \\ HXH^\dagger &= Z \\ HYH^\dagger &= -Y \end{aligned}$$

Control-NOT $C_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

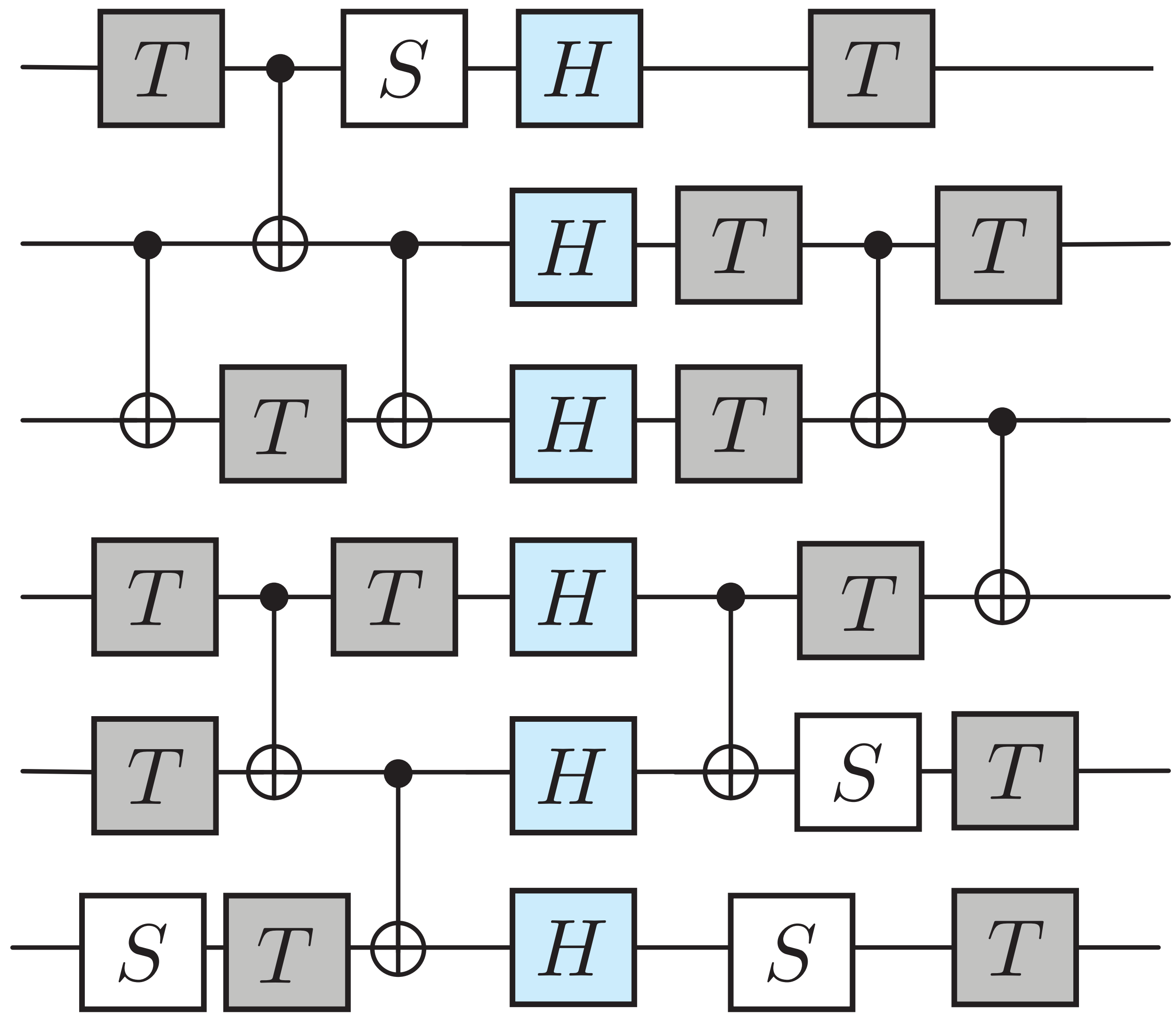


**T gate:** "harder" to implement, via expensive magic state distillation

**cost: 230$-500$**

nonClifford phase gate $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ $\boxed{T}$

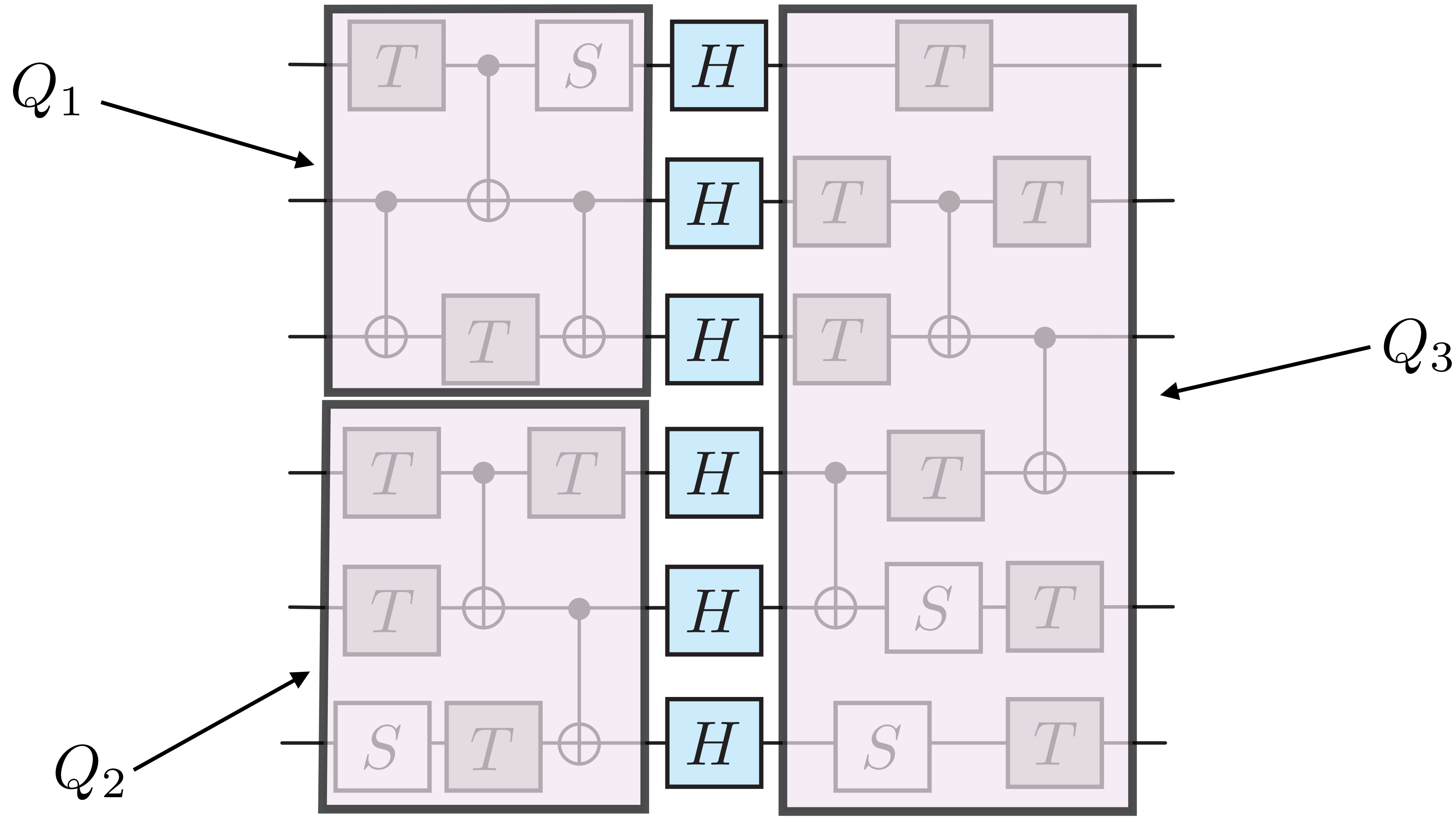
standard paradigm




must supply

$$|T\rangle^{\otimes 13} = |T\rangle|T\rangle|T\rangle|T\rangle|T\rangle|T\rangle|T\rangle|T\rangle|T\rangle|T\rangle|T\rangle|T\rangle|T\rangle$$

our paradigm




**Several papers on this class of subcircuits**

Selinger *Phys. Rev. A* **87**, 042302 (2013)

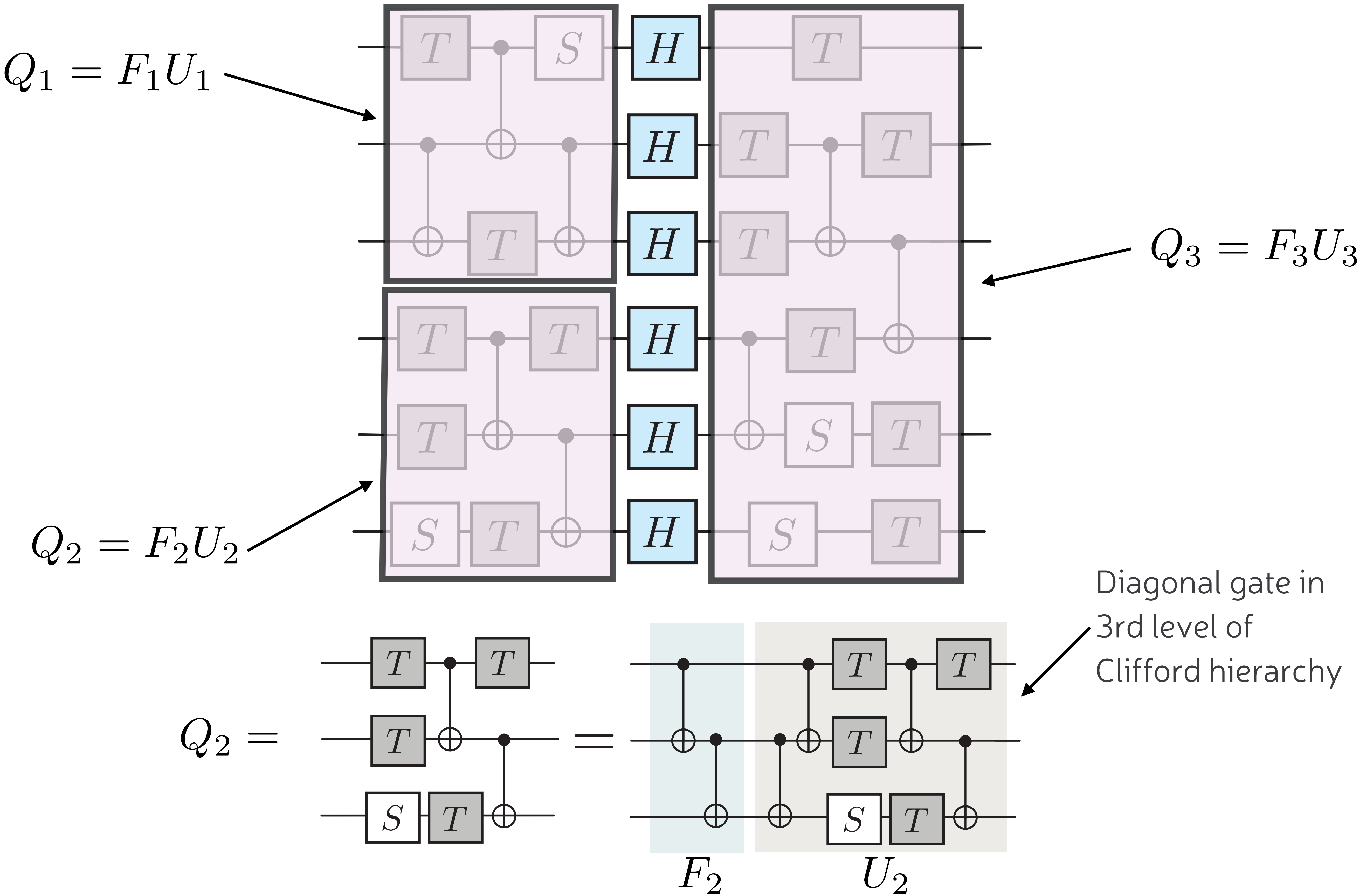
Amy, Maslov, Mosca, Roetteler *IEEE* **32** 818 (2013)

Amy, Maslov, Mosca *IEEE* **33** 1476 (2014)

Amy and Mosca arXiv:1601.07363 (2016)

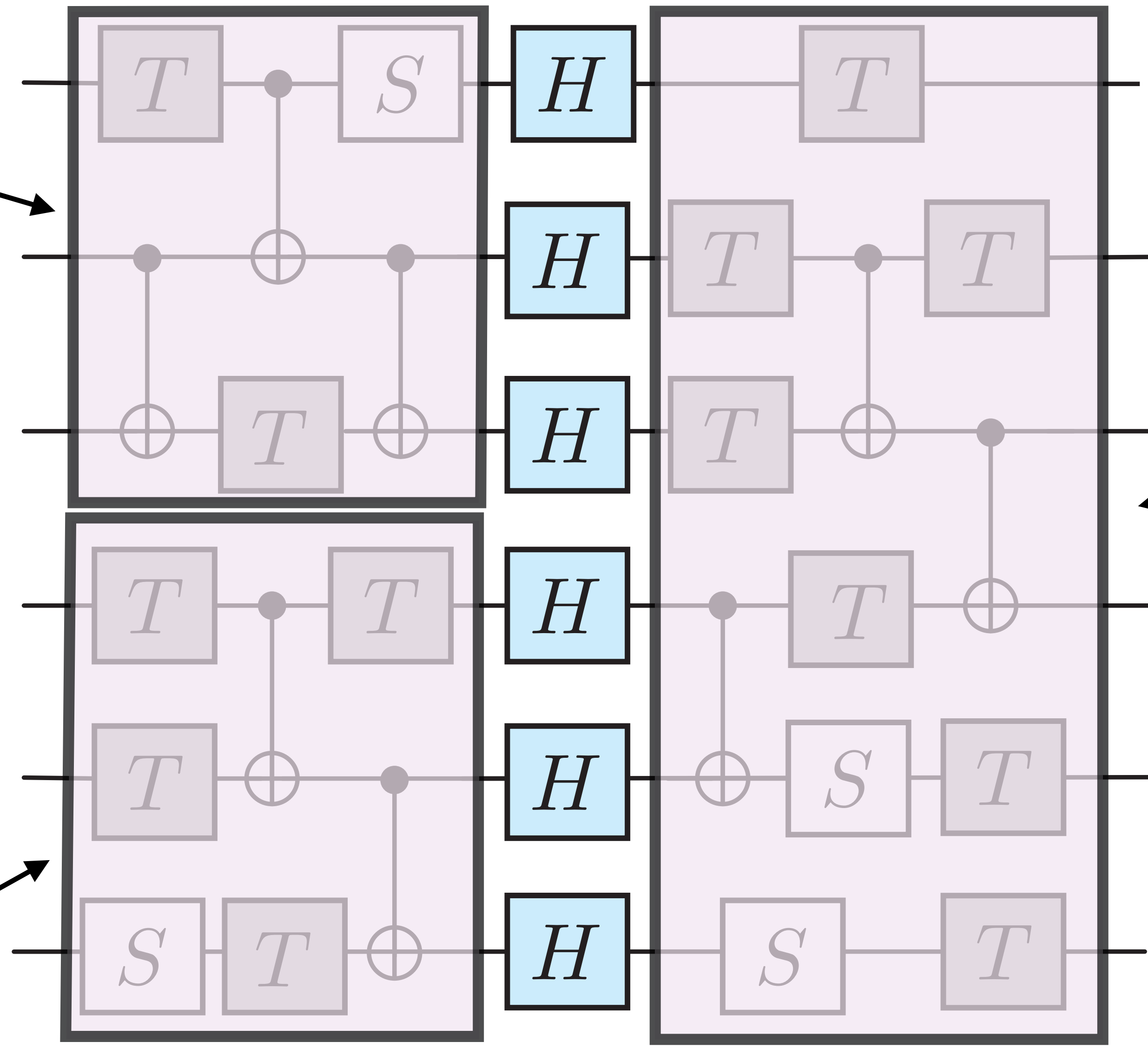
.... and more I am less familiar with!

our paradigm



our paradigm

$$Q_1 = F_1 U_1$$



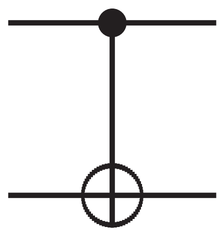

$$Q_3 = F_3 U_3$$

$$Q_2 = F_2 U_2$$

must supply

$$|U_1\rangle|U_2\rangle|U_3\rangle$$

## Example 0

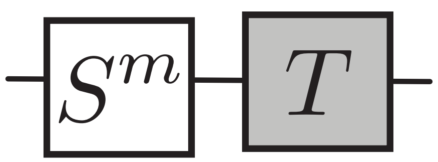


$$U|x_1, x_2\rangle = |x_1, x_1 \oplus x_2\rangle$$

mod 2 math

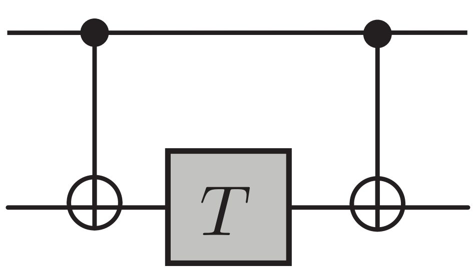
| $x_1$ | $x_2$ | $x_1 \oplus x_2$ |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 0 |

## Example 1



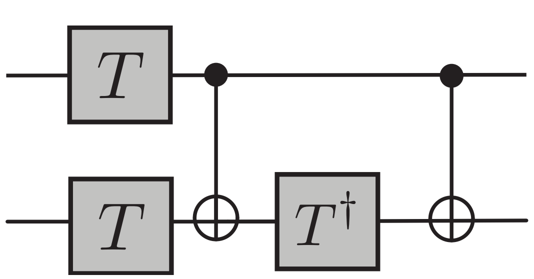

$$U|x_1\rangle = \omega^{(2m+1)x_1}|x_1\rangle$$ with $\omega = e^{i\pi/4}$

## Example 2

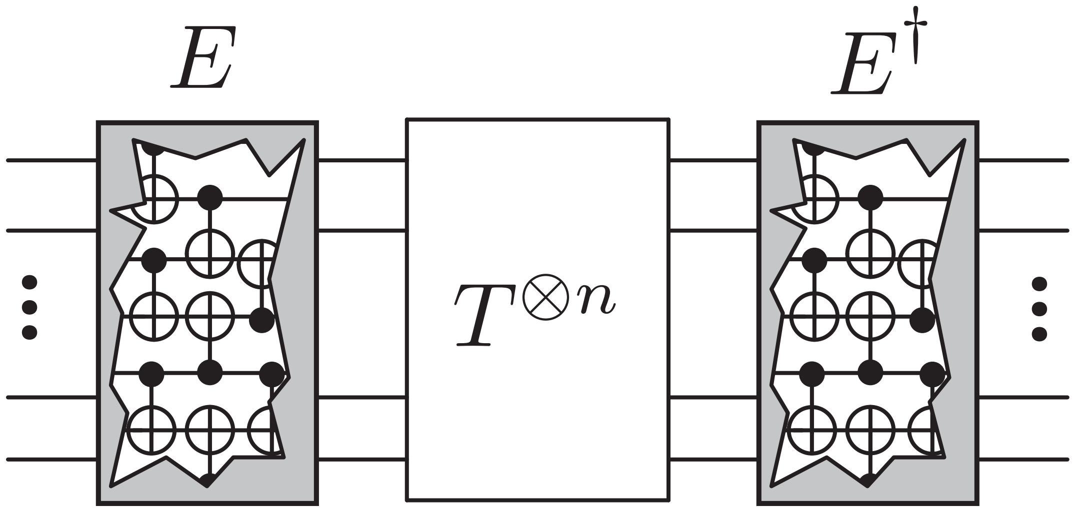



$$U|x_1, x_2\rangle = \omega^{x_1 \oplus x_2}|x_1, x_2\rangle$$

## Example 3




$$U|x_1, x_2\rangle = \omega^{x_1 + x_2 - x_1 \oplus x_2}|x_1, x_2\rangle$$




$$E|\vec{x}\rangle = |J^T\vec{x}\rangle$$

$J$ linearly independent and binary

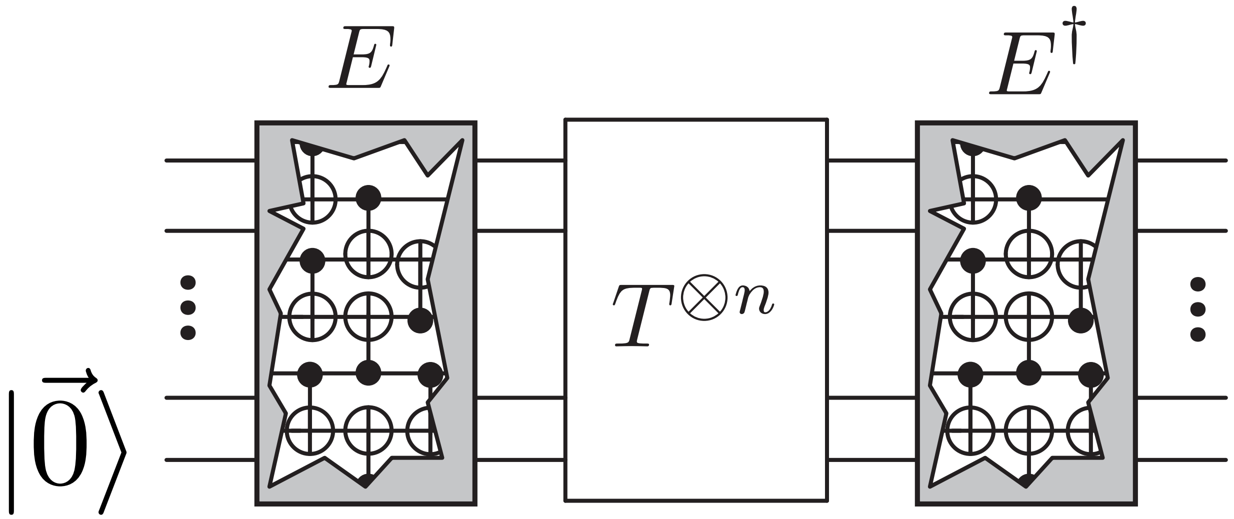
$$E^\dagger T^{\otimes n} E|\vec{x}\rangle = \omega^{|J^T\vec{x}|}|\vec{x}\rangle$$

Example

$$J = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \qquad J^T\vec{x} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_1 \oplus x_2 \oplus x_3 \end{pmatrix}$$

$$|J^T\vec{x}| = x_1 + x_2 + (x_1 \oplus x_2 \oplus x_3)$$

number variables = number qubits =number of terms = number of T-gates




$$E|\vec{x}\rangle|\vec{0}\rangle = |A^T\vec{x}\rangle$$

$A$ linearly independent and binary

$$E^\dagger T^{\otimes n} E|\vec{x}\rangle|\vec{0}\rangle = \omega^{|A^T\vec{x}|}|\vec{x}\rangle|\vec{0}\rangle$$

Example

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \quad A^T\vec{x} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_1 \oplus x_2 \end{pmatrix}$$

$$|A^T\vec{x}| = x_1 + x_2 + x_1 \oplus x_2$$

**row(A)**=number variables = number qubits
**col(A)**=number of terms = number of T-gates

## Clifford equivalence

$$U_f = C_1 U_g C_2 \quad \Longleftrightarrow \quad f(\vec{x}) \sim_c g(\vec{x})$$

For all third level diagonal gates

$$U|\vec{x}\rangle = \omega^{f(\vec{x})}|\vec{x}\rangle \quad \text{with} \quad |\vec{x}\rangle = |x_1, x_2, \ldots, x_n\rangle$$

there exists some **A**

$$f(\vec{x}) \sim_c |A^T\vec{x}|$$

where

$$\mathrm{col}(A) = \tau(U) = \text{optimal unitary synthesis cost}$$

# Background

Magic state distillation

The magic T state $|T\rangle \propto |0\rangle + e^{i\pi/4}|1\rangle$

**Protocols for distillation**

Bravyi, Kitaev, *Phys. Rev. A* **71** 022316 (2005)
Meier, Eastin, Knill, *QIC* **13** 0195 (2013)
Bravyi, Haah, *Phys. Rev. A* **86** 052329 (2012)
Jones, *Phys. Rev. A* **87**, 042305 (2013)

**3k+8** inputs
$\epsilon$ error rate

**Bravyi-Haah** →

**k** outputs
$\sim (1+3k)\epsilon^2 + O(\epsilon^3)$

**Triorthogonal matrices** key technical feature of Bravyi-Haah.

**n :** input magic states

$$G = \left(\frac{K}{S}\right)$$

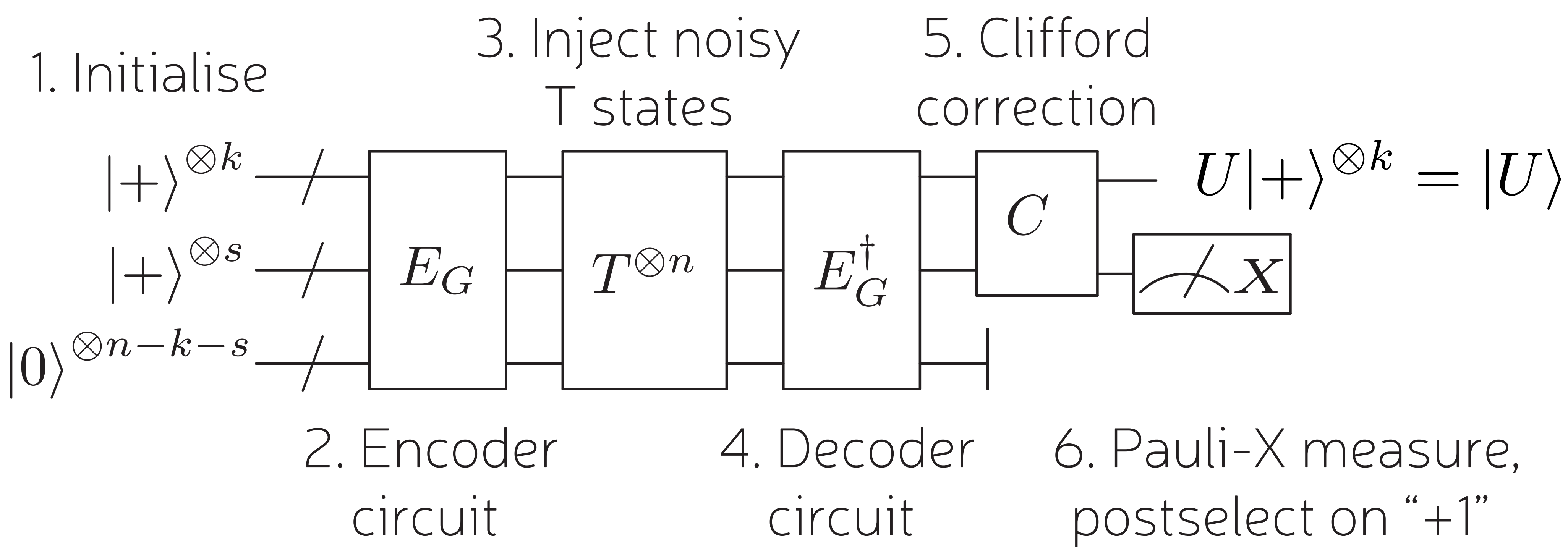
**k :** input magic states

**s :** "checks" for noise

Triorthogonality in our language

$$|K^T\vec{x} \oplus S^T\vec{y}| \sim_c x_1 + x_2 + \ldots + x_k$$

# Bravyi-Haah protocol




where encoder acts as

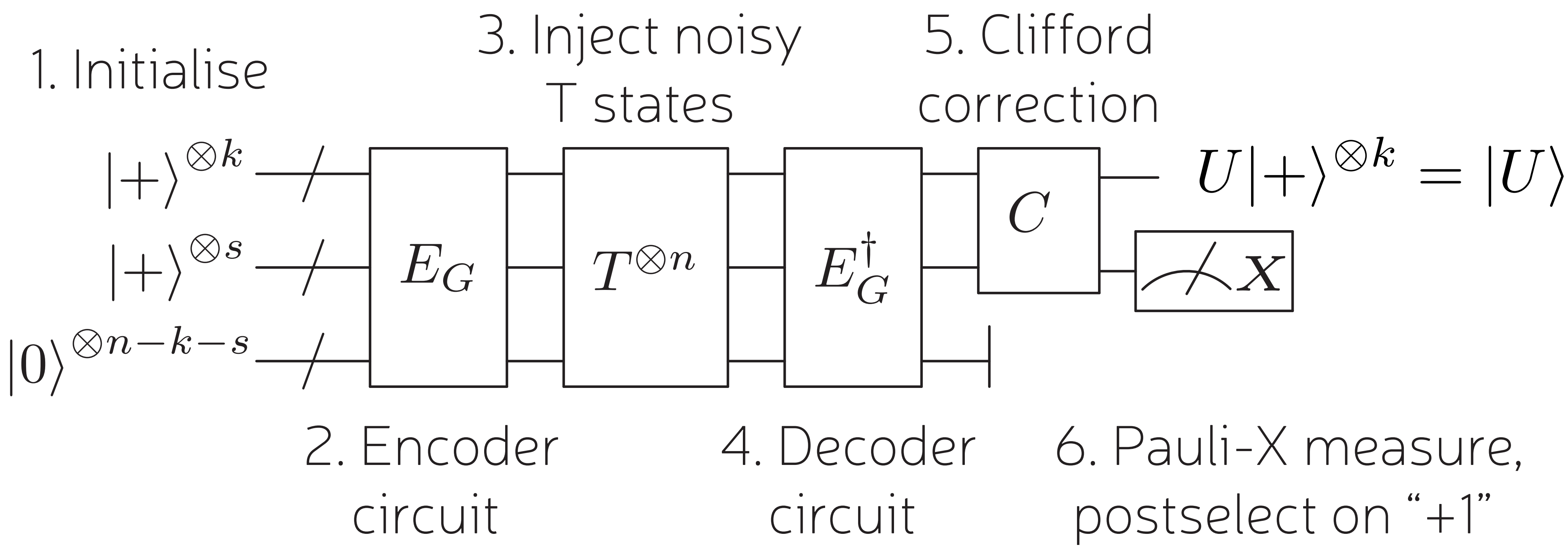
$$E_G|\vec{x}\rangle|\vec{y}\rangle|\vec{0}\rangle = |K^T\vec{x} \oplus S^T\vec{y}\rangle \qquad G = \left(\frac{K}{S}\right)$$

**Synthillation (*verb*).**
*To perform synthesis and distillation in a single step. Origin: a portmanteau of these two processes.*


PM

# Our synthillation protocol

synthillation = synthesis + distillation




where encoder acts as

$$E_G|\vec{x}\rangle|\vec{y}\rangle|\vec{0}\rangle = |K^T\vec{x} \oplus S^T\vec{y}\rangle \qquad G = \left(\frac{K}{S}\right)$$

and **G** is built from gate synthesis matrix **A**

*width is distillation cost*

*width is T-count*

**Main result:**

Distills $|U\rangle$ provided $|K^T\vec{x} \oplus S^T\vec{y}| \sim_c f(\vec{x})$

where $f(\vec{x})$ is the function $U|\vec{x}\rangle = \omega^{f(\vec{x})}|\vec{x}\rangle$

$\epsilon \rightarrow O(\epsilon^d)$ distance ***d*** depends on $S$ $\quad p_{\mathrm{suc}} = 1 - O(\epsilon)$

**Special cases:**

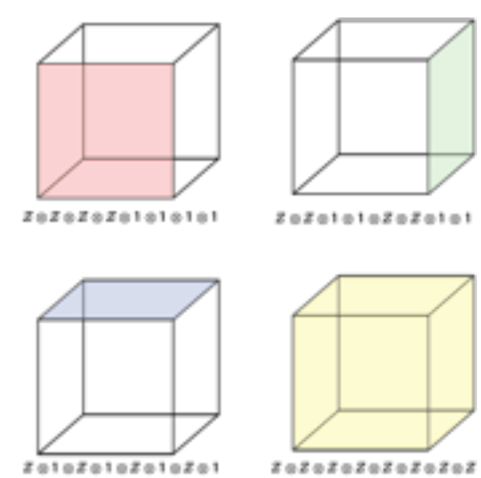
Bravyi-Haah

Distills $|U\rangle = |T\rangle^{\otimes k}$

provided

$$|K^T\vec{x} \oplus S^T\vec{y}| \sim_c x_1 + x_2 + \ldots + x_k$$

Blog post on color codes



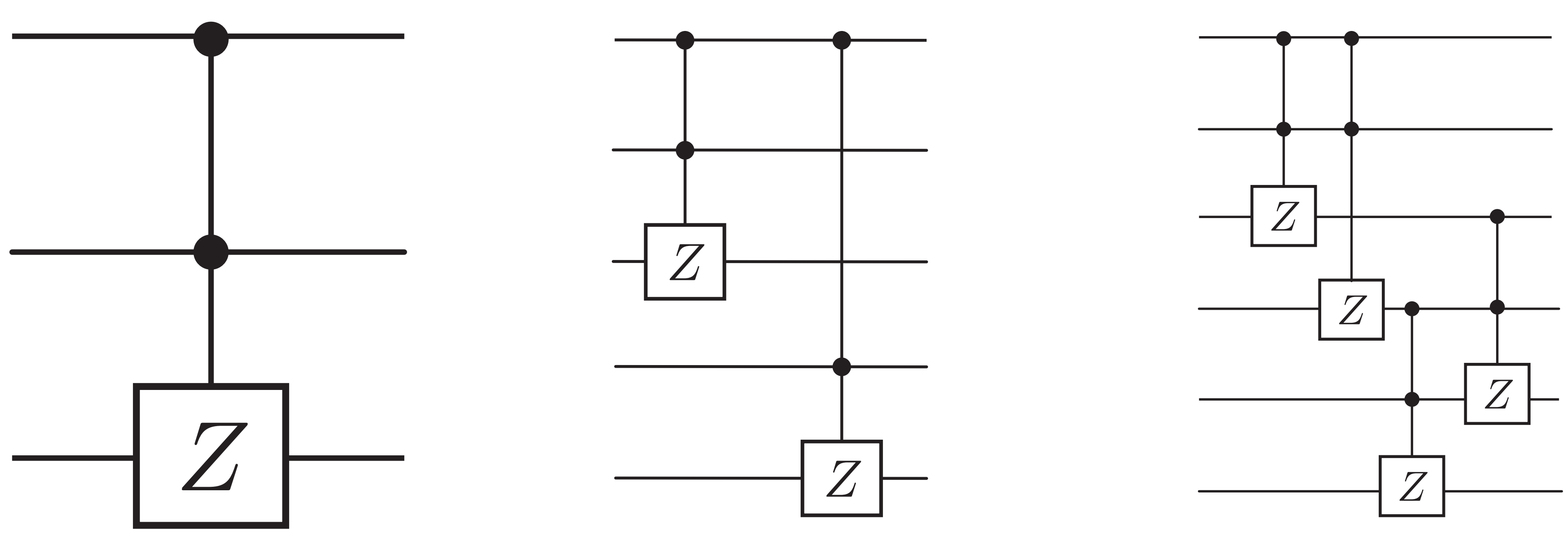
**But still** how does one find good **K** and **S**

**Special case** circuits of CCZ gates

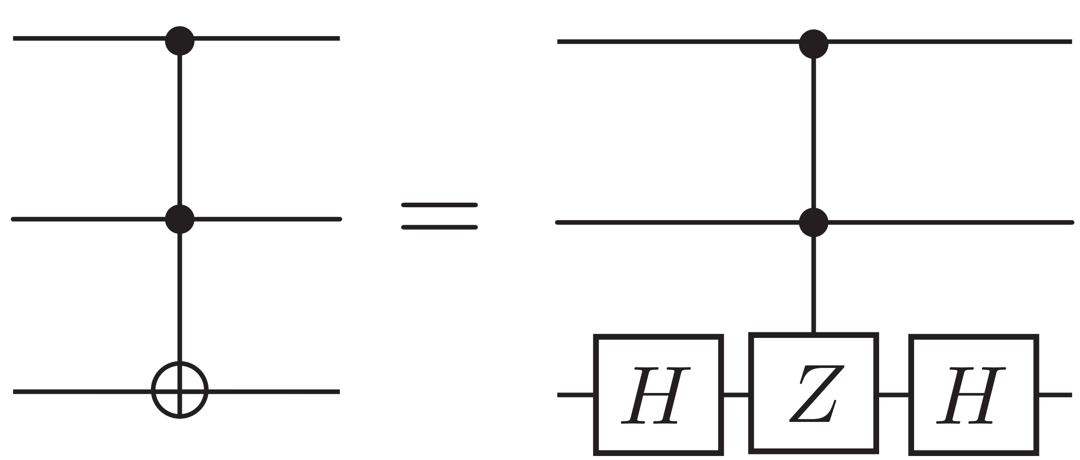
$$4x_1x_2x_3 = x_1 + x_2 + x_3 + (x_1 \oplus x_2 \oplus x_3) + 7(x_1 \oplus x_2) + 7(x_2 \oplus x_3) + 7(x_1 \oplus x_3)$$

only get phase if all $x_j$ variables equal to 1

**Examples**




Equivalence to Toffoli



**Special case** circuits of CCZ gates

assume odd width

$$G = \left( \frac{K}{S} \right) = \left( \begin{array}{cc} A & 0 \\ \hline 1 & 1 \end{array} \right)$$

**Protective layer of checks**
Give quadratic error suppression.

*Synthillation cost* = $\tau[U] + 1$

$$|A^T \vec{x}| \sim_c f(\vec{x}) \implies |K^T \vec{x} \oplus S^T \vec{y}| \sim_c f(\vec{x})$$


The gate synthesis matrix

$$|A^T\vec{x}| \sim_c f(\vec{x})$$

T-count $\tau[U] = \mathrm{col}[A]$


Narrowest matrix with $AA^T = BB^T \pmod 2$

Use Lempel (1975) solver to find efficiently

define $\mu[U] = \mathrm{col}[B]$

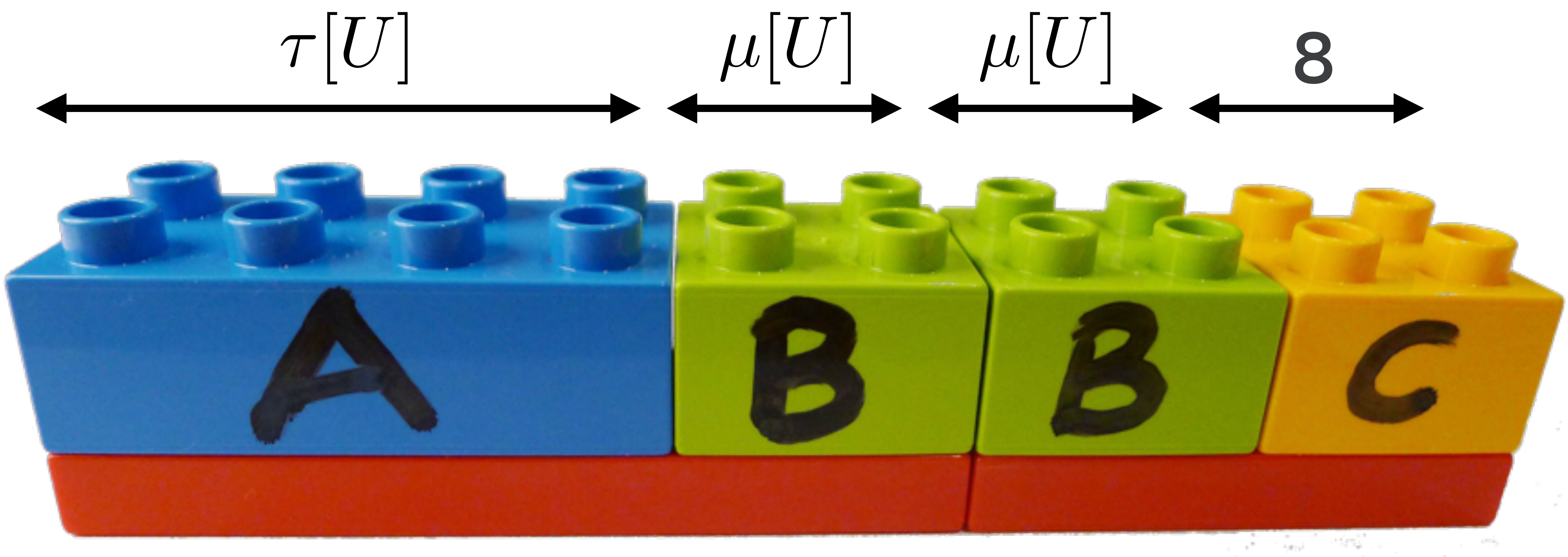
$\mu[U] \leq \tau[U]$ Always

$\mu[U] \ll \tau[U]$ For large “typical” circuits (PROOF)

$\mu[U] = 0$ For CCZ circuits.


Matrix of bounded width $\Delta \leq 11$

e.g. if **A** and **B** are even then $\Delta = 8$



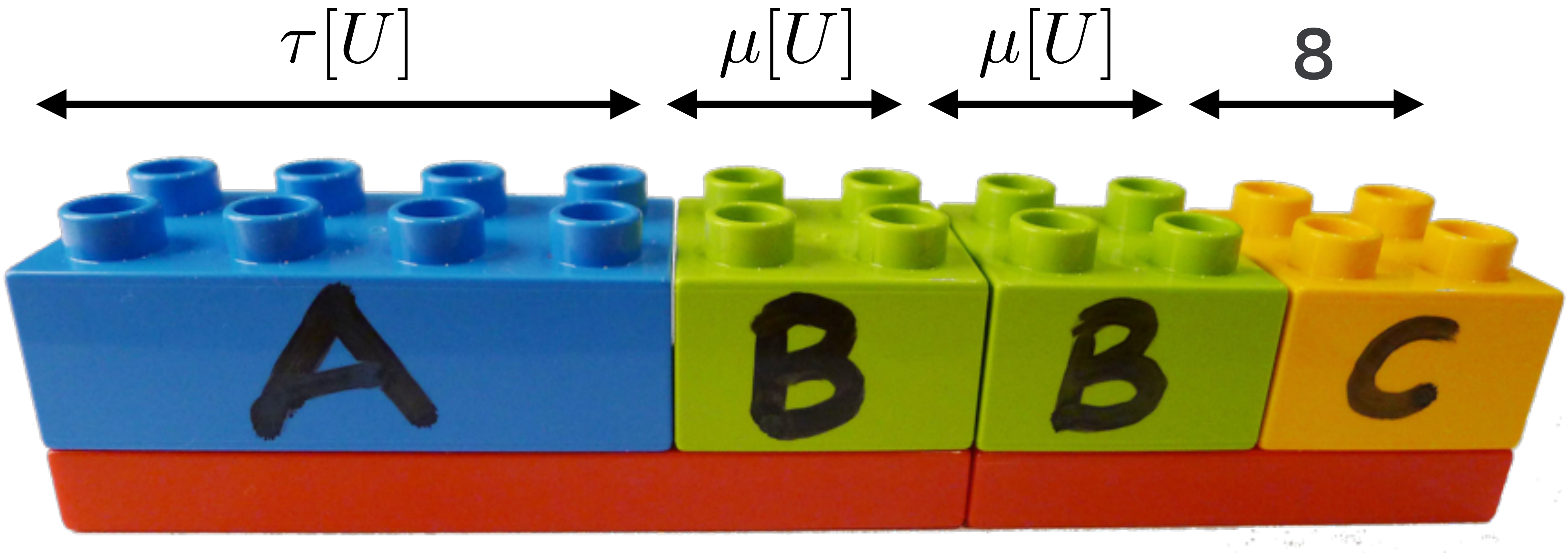

*Synthillation cost* = $\tau[U] + 2\mu[U] + 8$

Typically have $\mu[U] \ll \tau[U]$

then

*Synthillation cost* $\sim \tau[U] =$ *Synthesis cost**

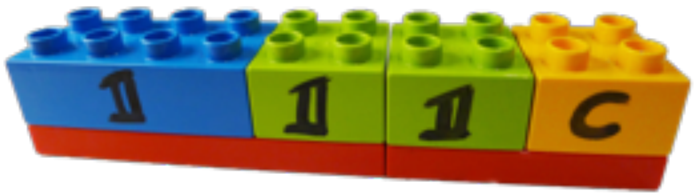
**without ancilla*




$$G = \left(\frac{K}{S}\right) = \left(\begin{array}{ccccccccccc} A & B & B & \vec{c} & \vec{c} & \vec{c} & \vec{c} & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

# Resource Comparison

**Bravyi-Haah**



may need additional distillation

distill

$3\tau[U] + 8$
$T$-states with $\epsilon$ noise

distill

$\tau[U]$ $T$-states
with $O(\epsilon^2)$ noise

synthesize

$U$ gate
with $O(\epsilon^2)$ noise

standard paradigm

distill

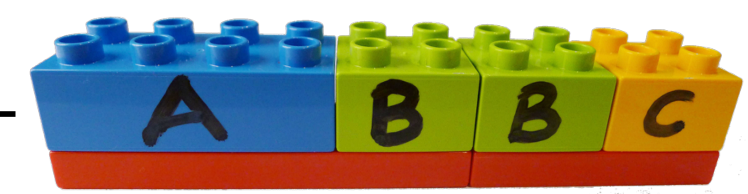
$\tau[U] + 2\mu[U] + \Delta$
$T$-states with $\epsilon$ noise

synthillate

$U$ gate
with $O(\epsilon^2)$ noise

our mindset

**Synthillation**



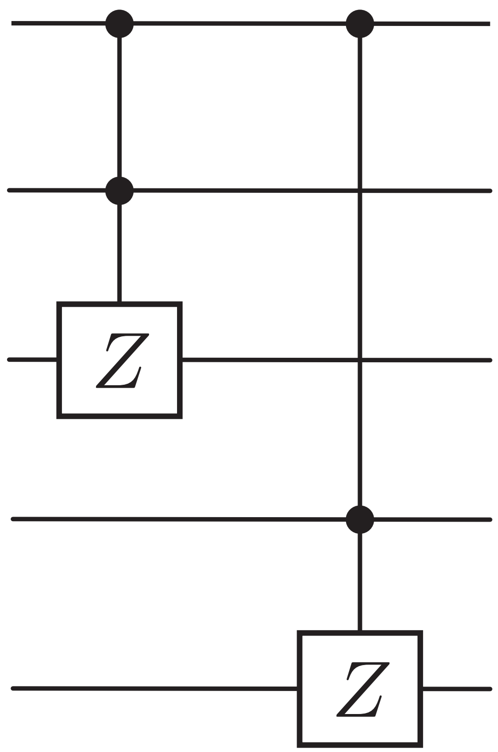

**Gate synthesis**

# Tof#

Without error suppression

Naive (noisy) circuit cost
**T-count = 14**

Optimised (noisy) circuit cost
**T-count = 11**




With error suppression

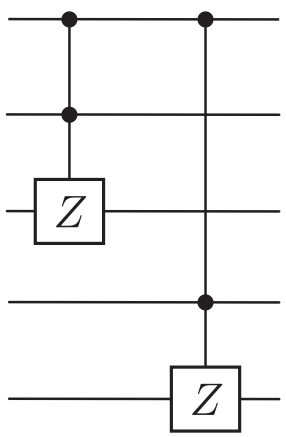
Circuit cost using distilled T states
**T-count ~ 33 +**
(quadratic error suppression)

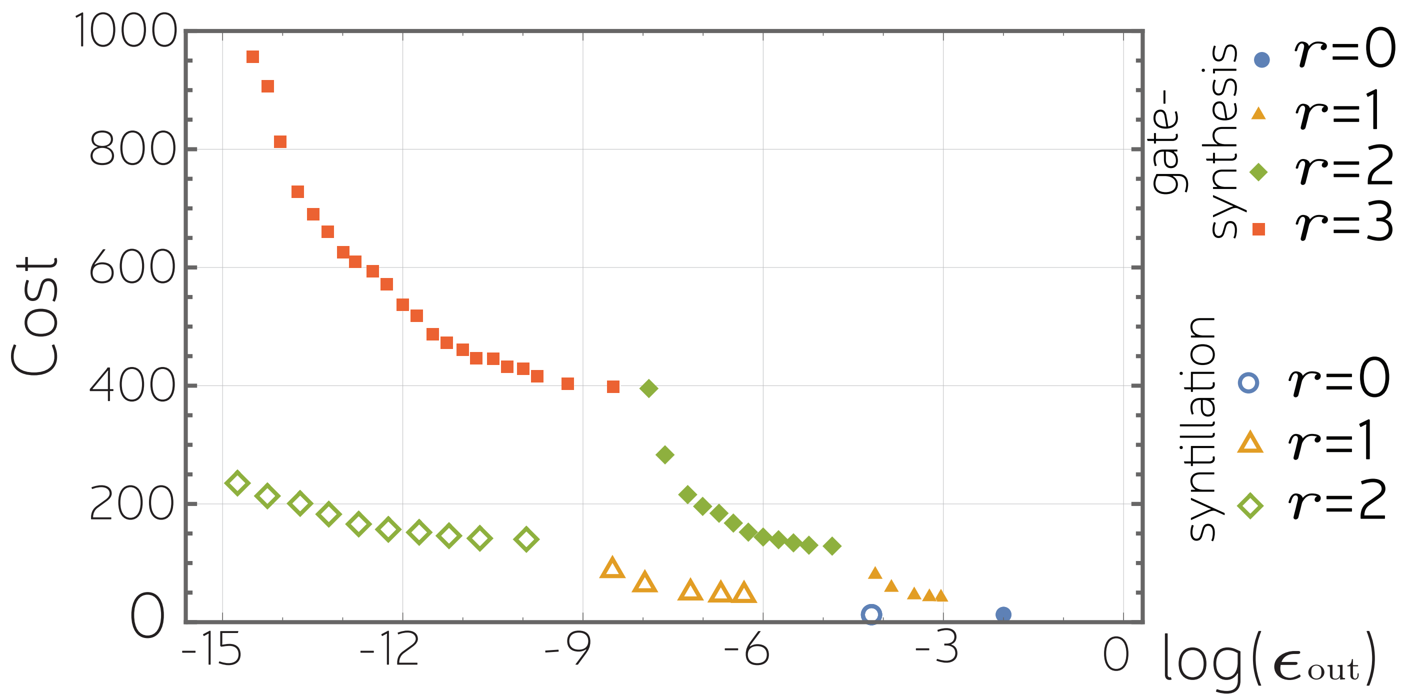
Synthillation cost
**T-count = 12**
(quadratic error suppression)

**12 << 33+**

synthillation wins!

# Resource costs: synthillation vs. standard paradigm




## Example IV .4

**0.4 Example IV**

```
Ginfo2 HASH = Case11[ATofHash];
Analyse[Ginfo2 HASH]
```

$$\left(\begin{array}{cccccccccccc} 0&0&0&1&1&0&0&0&1&1&0&0\\ 0&0&1&1&0&0&0&1&1&0&0&0\\ 0&1&1&1&1&0&1&1&1&1&0&0\\ 1&0&1&1&1&1&0&1&1&1&0&0\\ 1&1&1&1&1&0&0&0&0&0&1&0\\ \hline 1&1&1&1&1&1&1&1&1&1&1&1 \end{array}\right)$$

Linearly independent = True

well behaved = True

[[n,k,d]] = [[12,5,2]]

Rate = 0.416667

Prob Success = $1 - 12\epsilon + 132\epsilon^2 - 880\epsilon^3 + 3960\epsilon^4 - 12\,672\epsilon^5 + 29\,568\epsilon^6 - 50\,688\epsilon^7 + 63\,360\epsilon^8 - 56\,320\epsilon^9 + 33\,792\epsilon^{10} - 12\,288\epsilon^{11} + 2048\epsilon^{12}$

Error out (numerator) = $66\epsilon^2 - 660\epsilon^3 + 3450\epsilon^4 - 11\,760\epsilon^5 + 28\,192\epsilon^6 - 48\,864\epsilon^7 + 61\,320\epsilon^8 - 54\,560\epsilon^9 + 32\,736\epsilon^{10} - 11\,904\epsilon^{11} + 1984\epsilon^{12}$

Error out (normalised) = $66\epsilon^2 + 132\epsilon^3 - 3678\epsilon^4 - 15\,240\epsilon^5 + 185\,608\epsilon^6 + 1\,267\,104\epsilon^7 - 8\,358\,792\epsilon^8 - 91\,097\,536\epsilon^9 + 308\,405\,568\epsilon^{10} + O[\epsilon]^{11}$

4 (x[1] x[2] x[5] + x[3] x[4] x[5])

# Spin off ideas

new algorithms for
gate synthesis

NOT distillation

## Gate-synthesis optimisation

find best phase polynomial <-> smallest **A** matrix


$$|A_1^T \vec{x}| \sim_c f(\vec{x})$$


$$|A_2^T \vec{x}| \sim_c f(\vec{x})$$


$$|A_3^T \vec{x}| \sim_c f(\vec{x})$$

**Shown by** Amy and Mosca arXiv:1601.07363 (2016)

**Gate-synthesis optimisation** find best phase polynomial

| | T-count | Difficultly |
|---|---|---|
| Optimal solver (reed-Muller decoder) | $\leq \frac{n^2}{2} + O(n)$ optimal | Believed very hard. Practically limited to n=6 Related to tensor contraction. |
| Simple solver | $\leq O(n^3)$ usually suboptimal | Super fast |
| Our solver* for U=control-C | $\leq 2n+1$ optimal | Super fast |
| Our solver* for general problem | $\leq \frac{n^2}{2} + O(n)$ no optimality promise | Super fast |

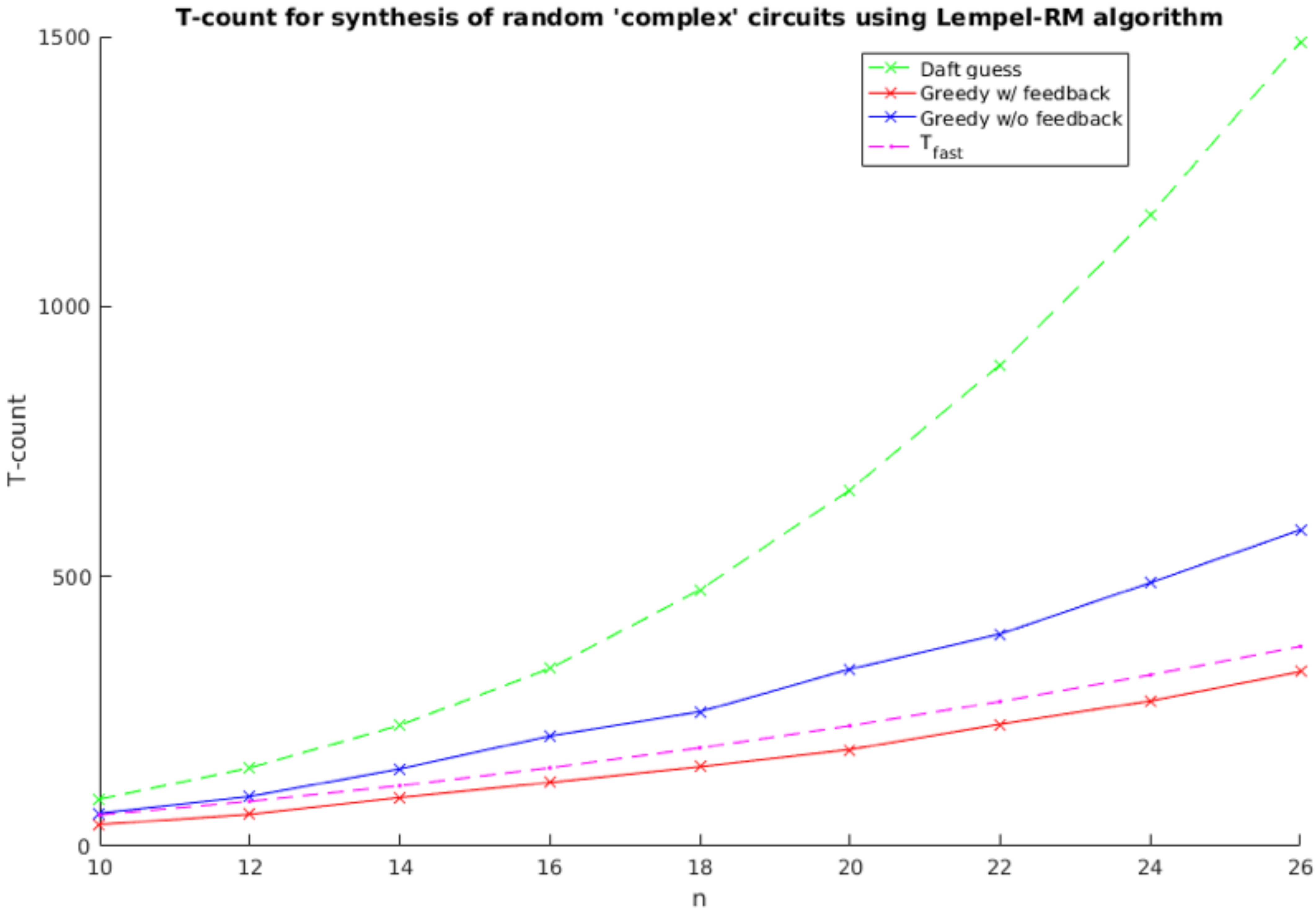
Reduce problem to factorisation

$$Q = BB^T \pmod 2$$

and use Lempel (1975)

## **Preliminary numerical data** watch this space

$O(n^3)$




$O(n^2)$

Luke Heyfron
PhD student

THANK YOU!