Title: Threshold lower bounds for Knill\'s Fibonacci scheme

Date: Jun 16, 2007 10:00 AM

URL: http://pirsa.org/07060060

Abstract:

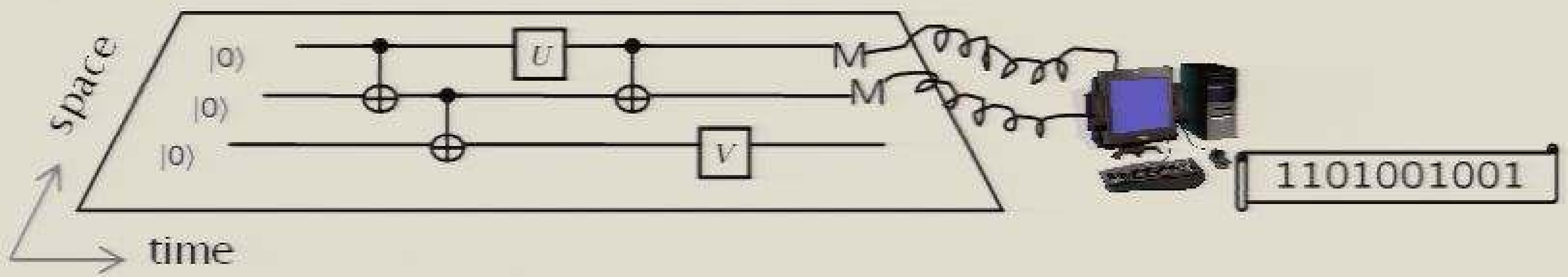
the plan

the basics	- level reduction - error-correction by teleportation	10min
Fibonacci scheme	- distance-2 codes and flags - the error model, and the 10^{-3} lower bound	20min
questions	- message-passing decoding for computation - FT with diagonal gates, and biased noise	15min

the plan

the basics	- level reduction - error-correction by teleportation	10min
Fibonacci scheme	- distance-2 codes and flags - the error model, and the 10^{-3} lower bound	20min
questions	- message-passing decoding for computation - FT with diagonal gates, and biased noise	15min

level reduction



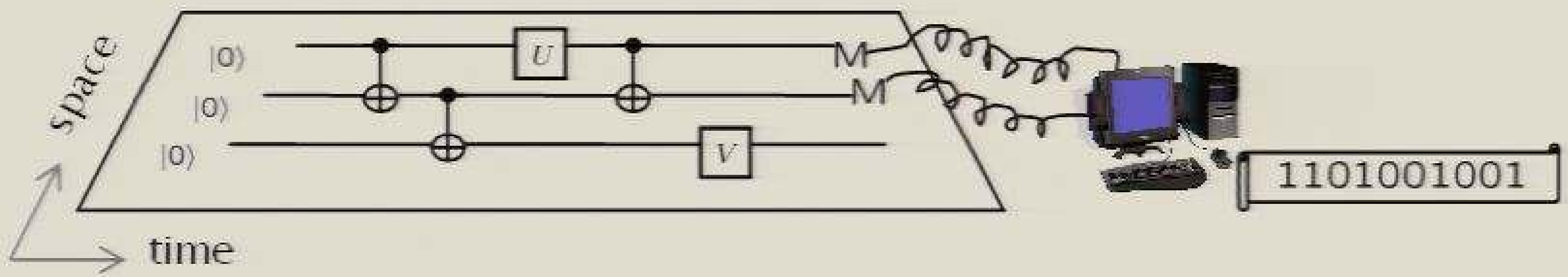
– write each real, noisy gate as $U_{\text{real}} = (I + F)U_{\text{ideal}}$.

– open parentheses to get a sum over "fault paths."

our goal: show that *bad* fault paths, i.e., those that lead to different final statistics than the ideal computation, occur with probability arbitrarily close to 0 if noise is sufficiently weak.

def. noise is *adversarial stochastic* if the probability of fault paths with faults at r specific noisy gates is at most ε^r for some constant *strength* ε.

level reduction



– write each real, noisy gate as $U_{\text{real}} = (I + F)U_{\text{ideal}}$.

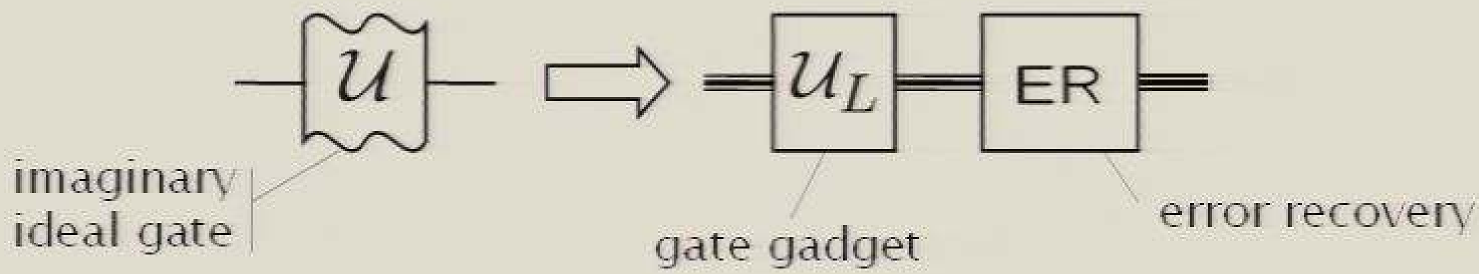
– open parentheses to get a sum over "fault paths."

> our goal: show that *bad* fault paths, i.e., those that lead to different final statistics than the ideal computation, occur with probability arbitrarily close to 0 if noise is sufficiently weak.

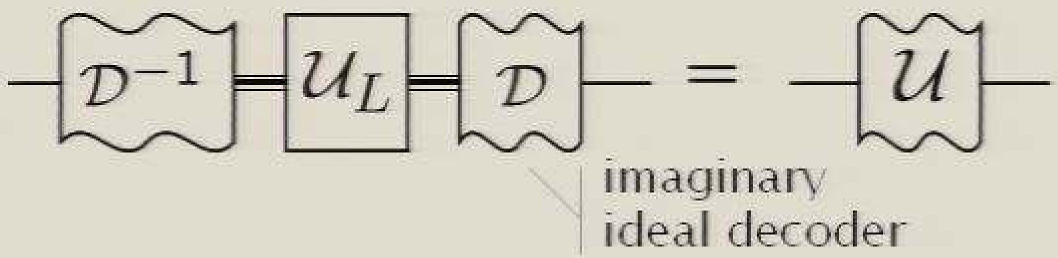
<u>def.</u> noise is *adversarial stochastic* if the probability of fault paths with faults at r specific noisy gates is at most ε^r for some constant *strength* ε.

level reduction

idea: choose a code of distance $2t+1$ to encode computation in

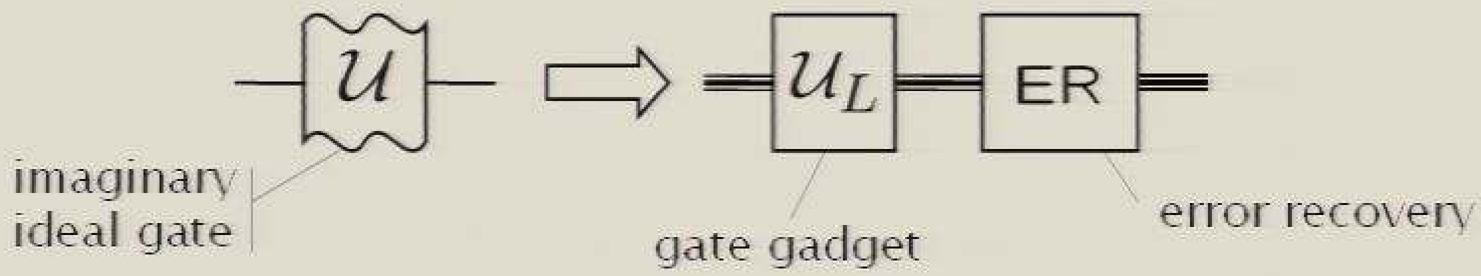


such that encoding is *correct*, i.e., if there were no noise

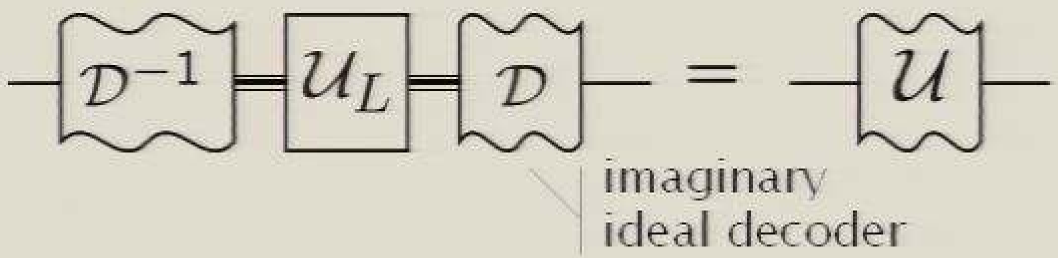


level reduction

idea: choose a code of distance $2t+1$ to encode computation in

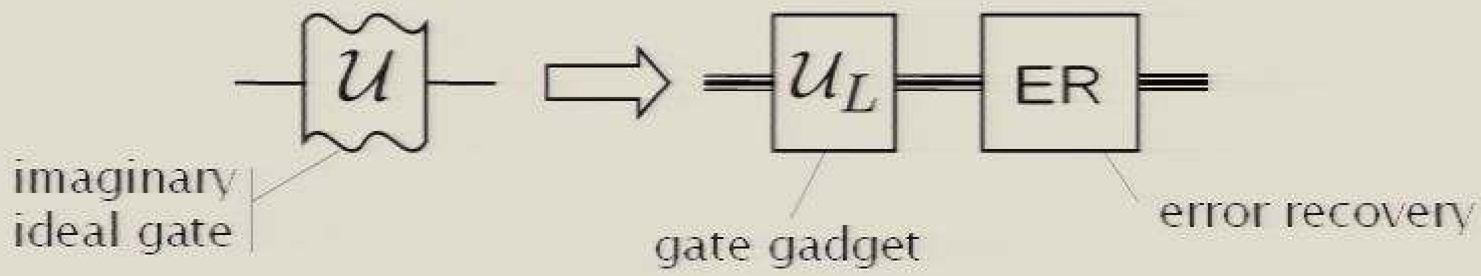


such that encoding is *correct*, i.e., if there were no noise

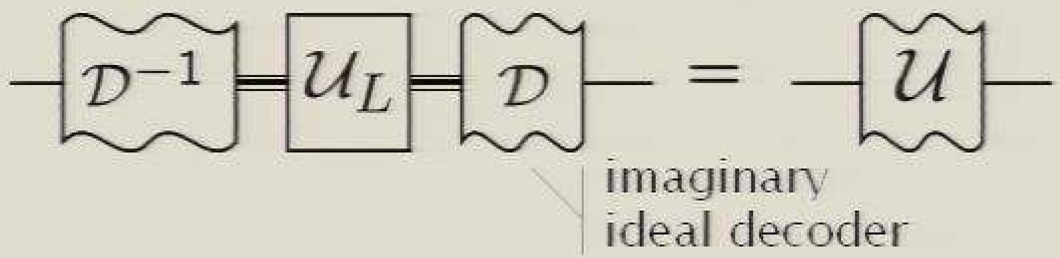


level reduction

idea: choose a code of distance $2t+1$ to encode computation in

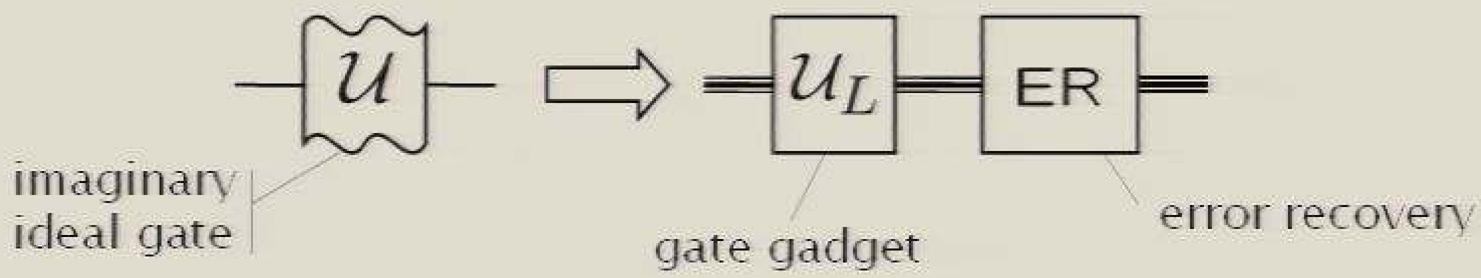


such that encoding is *correct*, i.e., if there were no noise

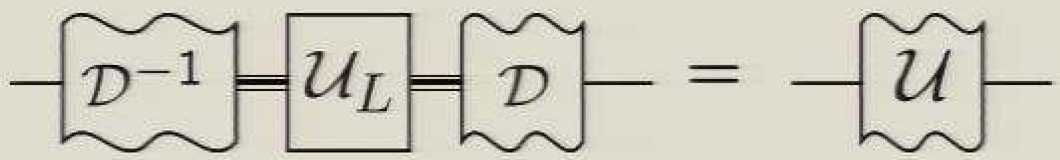


level reduction

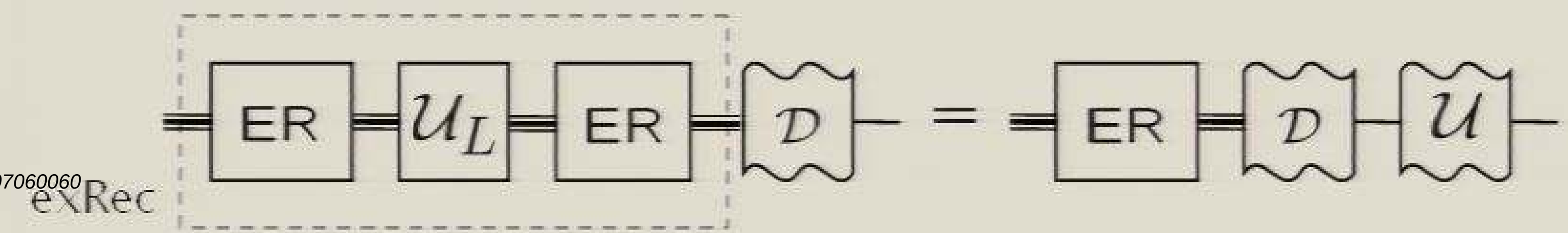
idea: choose a code of distance $2t+1$ to encode computation in



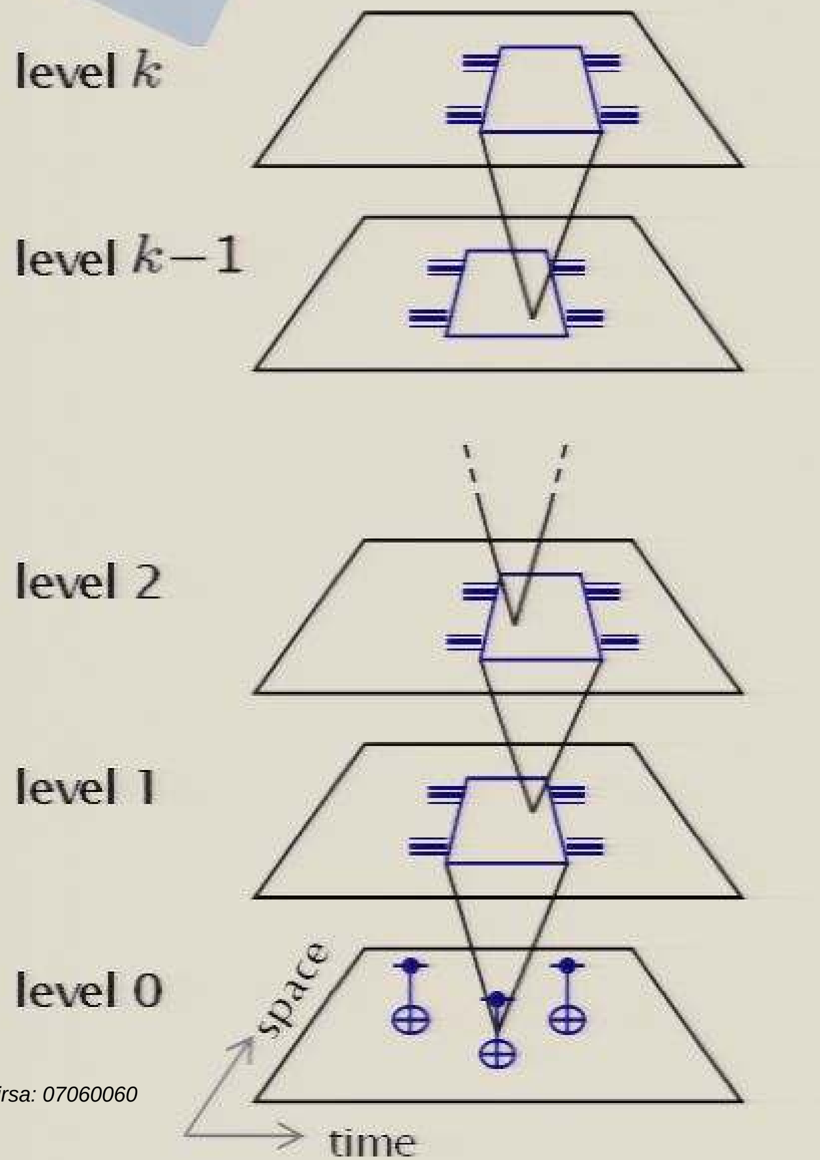
such that encoding is *correct*, i.e., if there were no noise



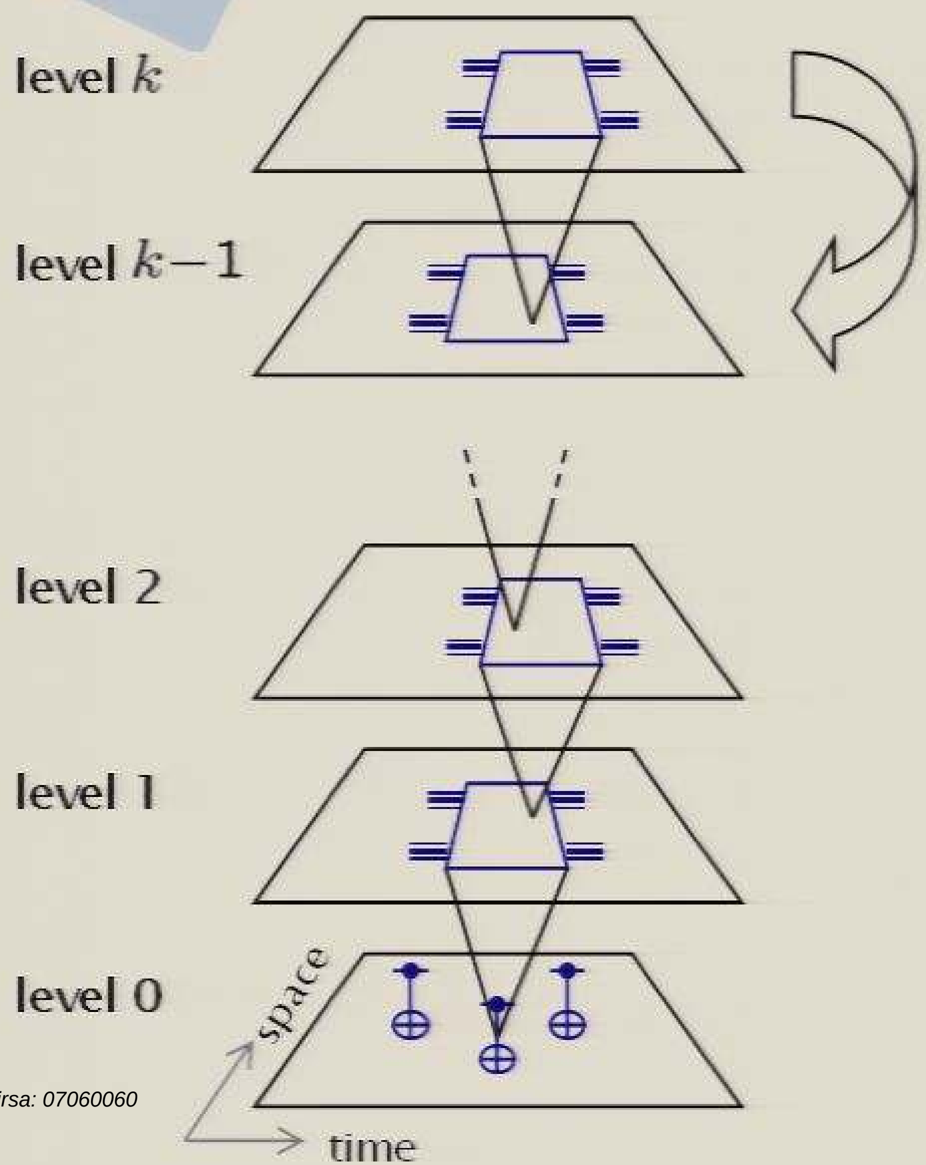
<u>lemma.</u> the simulation inside an exRec with $\leq t$ faults is correct, i.e.,



level reduction



level reduction



<u>lemma (level reduction).</u>

a level–k circuit subject to local noise of strength ε produces the same output statistics as a level–($k-1$) circuit subject to local noise of strength

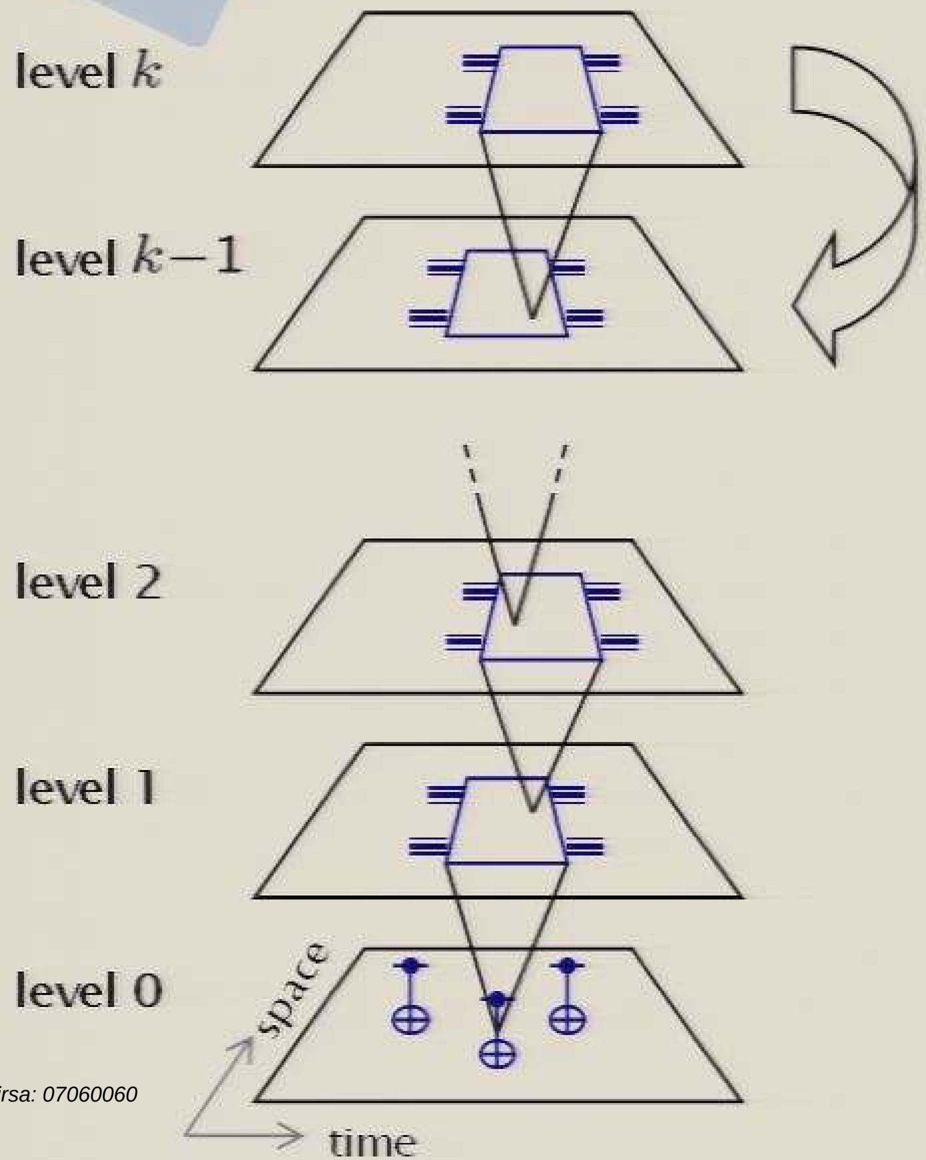
$$\varepsilon^{(1)} \equiv \varepsilon_0 \times \left(\frac{\varepsilon}{\varepsilon_0}\right)^{t+1} .$$

idea: each fault path in the level– k circuit can be *simulated* by a fault path acting before the last encoding step—encoding just changes the noise strength.

local noise model is stable !
can repeat k times to find

$$\varepsilon^{(k)} \equiv \varepsilon_0 \times \left(\frac{\varepsilon}{\varepsilon_0}\right)^{(t+1)^k} .$$

level reduction



lemma (level reduction).

a level–k circuit subject to local noise of strength ε produces the same output statistics as a level–$(k-1)$ circuit subject to local noise of strength

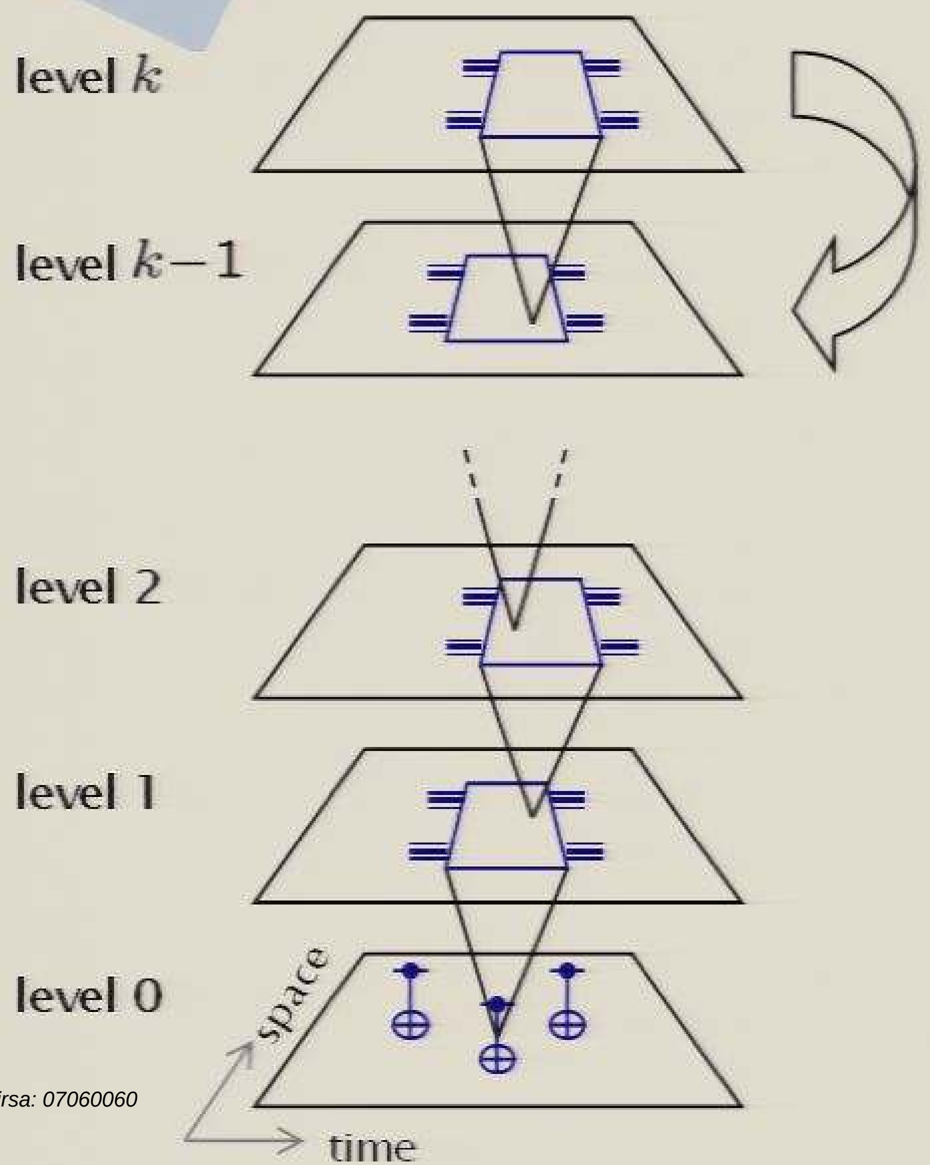
$$\varepsilon^{(1)} \equiv \varepsilon_0 \times \left(\frac{\varepsilon}{\varepsilon_0}\right)^{t+1} .$$

idea: each fault path in the level–k circuit can be *simulated* by a fault path acting before the last encoding step—encoding just changes the noise strength.

local noise model is stable !
can repeat k times to find

$$\varepsilon^{(k)} \equiv \varepsilon_0 \times \left(\frac{\varepsilon}{\varepsilon_0}\right)^{(t+1)^k} .$$

level reduction



<u>lemma (level reduction).</u>

a level–k circuit subject to local noise of strength ε produces the same output statistics as a level–$(k-1)$ circuit subject to local noise of strength

$$\varepsilon^{(1)} \equiv \varepsilon_0 \times \left(\frac{\varepsilon}{\varepsilon_0}\right)^{t+1} .$$

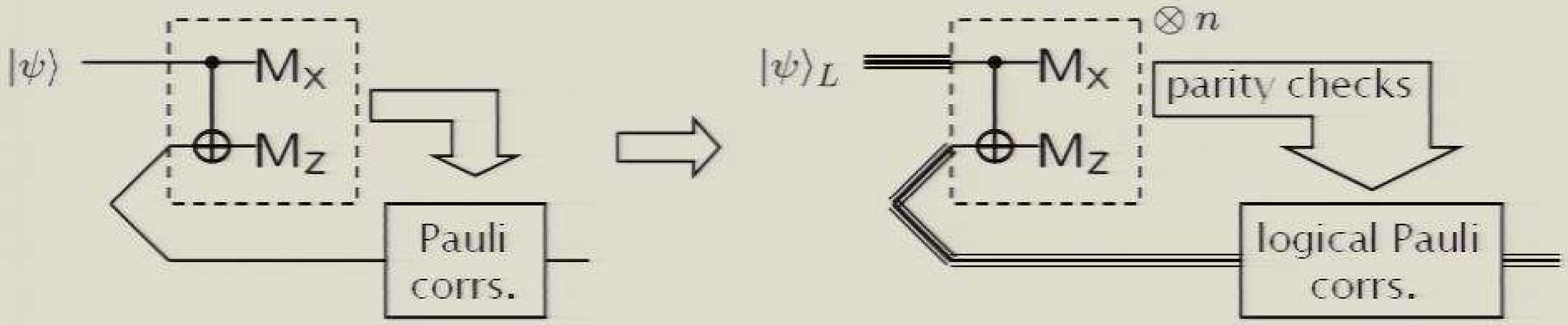
idea: each fault path in the level– k circuit can be *simulated* by a fault path acting before the last encoding step—encoding just changes the noise strength.

local noise model is stable !
can repeat k times to find

$$\varepsilon^{(k)} \equiv \varepsilon_0 \times \left(\frac{\varepsilon}{\varepsilon_0}\right)^{(t+1)^k} .$$

error-correction by teleportation

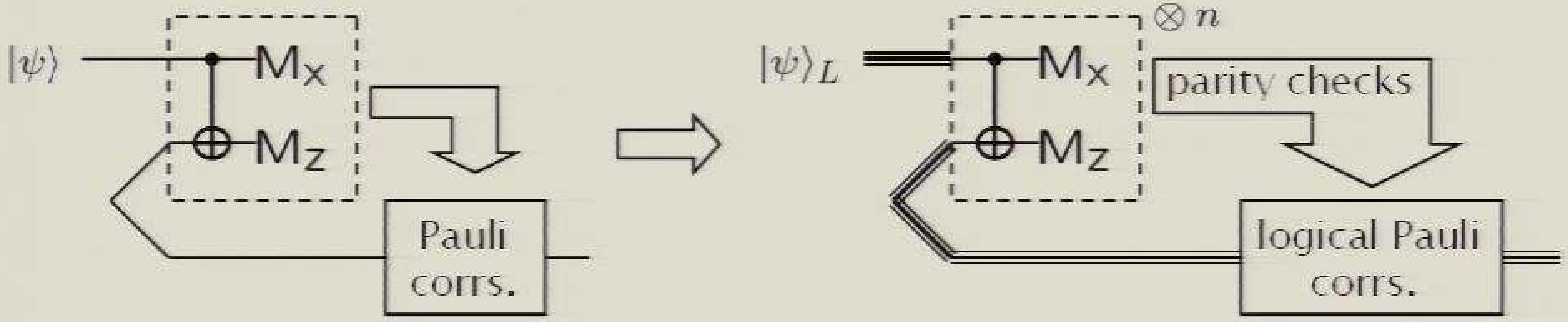
– encode teleportation using an n–qubit stabilizer code



idea: errors in the input cause will flip some measurement outcomes;
if errors are correctable, we can still infer the ideal logical Pauli corrs.

error-correction by teleportation

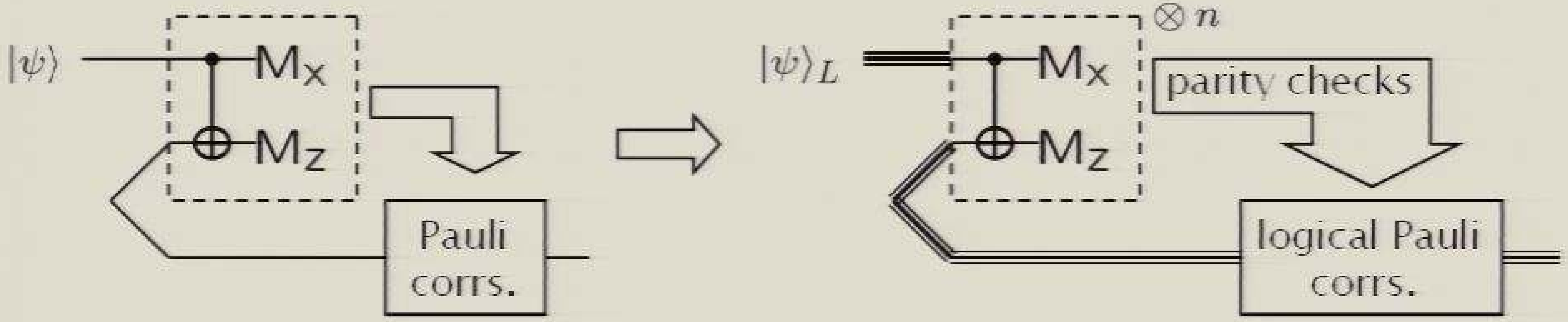
– encode teleportation using an n–qubit stabilizer code



idea: errors in the input cause will flip some measurement outcomes;
if errors are correctable, we can still infer the ideal logical Pauli corrs.

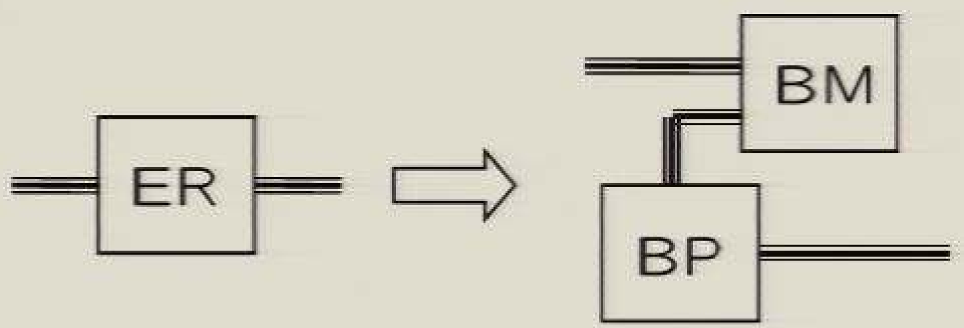
error-correction by teleportation

– encode teleportation using an n–qubit stabilizer code



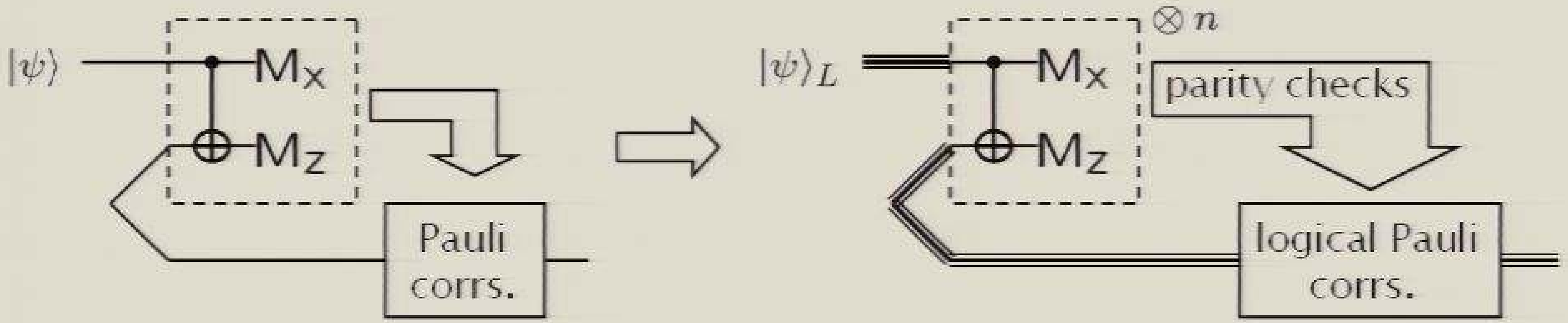
idea: errors in the input cause will flip some measurement outcomes;
if errors are correctable, we can still infer the ideal logical Pauli corrs.

– schematically,



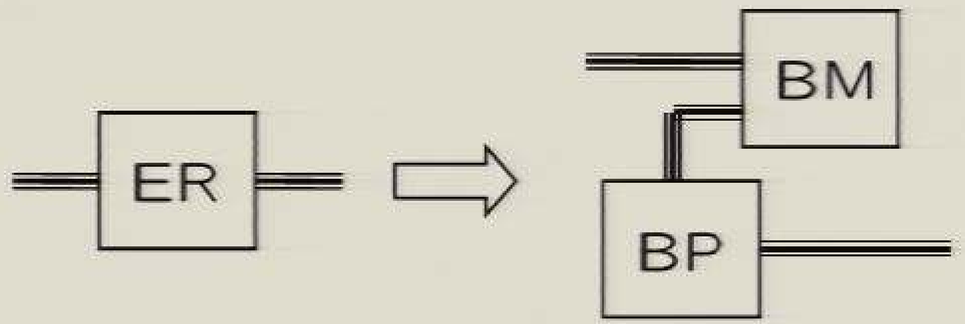
error-correction by teleportation

– encode teleportation using an n–qubit stabilizer code



idea: errors in the input cause will flip some measurement outcomes;
if errors are correctable, we can still infer the ideal logical Pauli corrs.

– schematically,



the Fibonacci or C_4/C_6 scheme

encode using concatenated distance–2 codes, and when you *detect* an error at one coding level, either abort and start again anew or try to *correct* it at the next level.

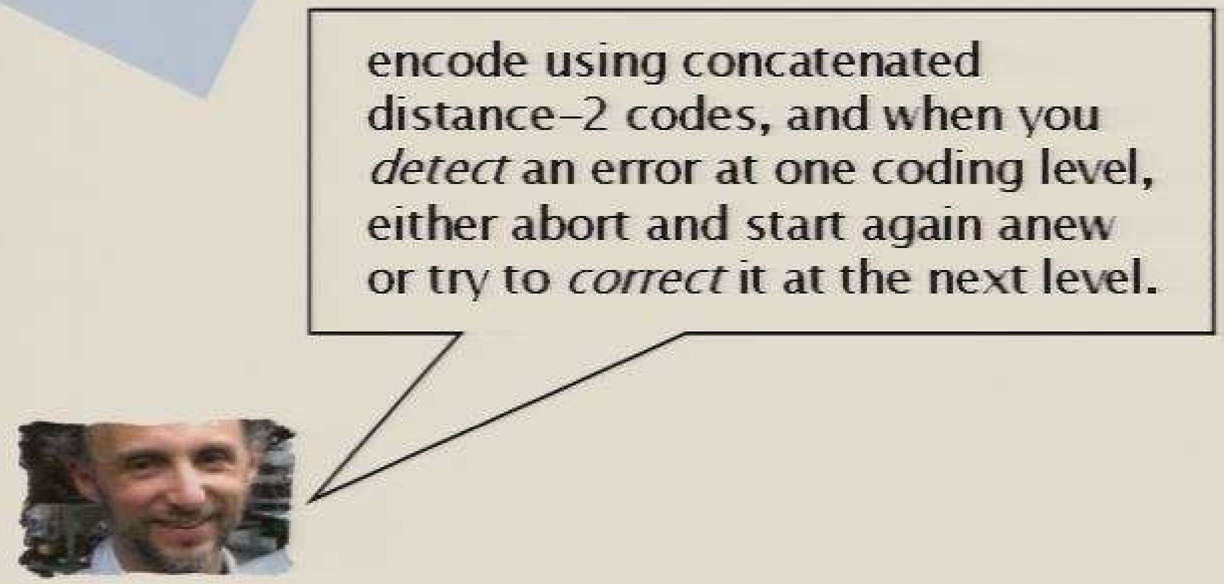
why distance–2 codes? Because circuit is so simple (~50 gates in CNOT–exRec).

the Fibonacci or C_4/C_6 scheme

> encode using concatenated distance-2 codes, and when you *detect* an error at one coding level, either abort and start again anew or try to *correct* it at the next level.

why distance-2 codes? Because circuit is so simple (~50 gates in CNOT-exRec).

the Fibonacci or C_4/C_6 scheme



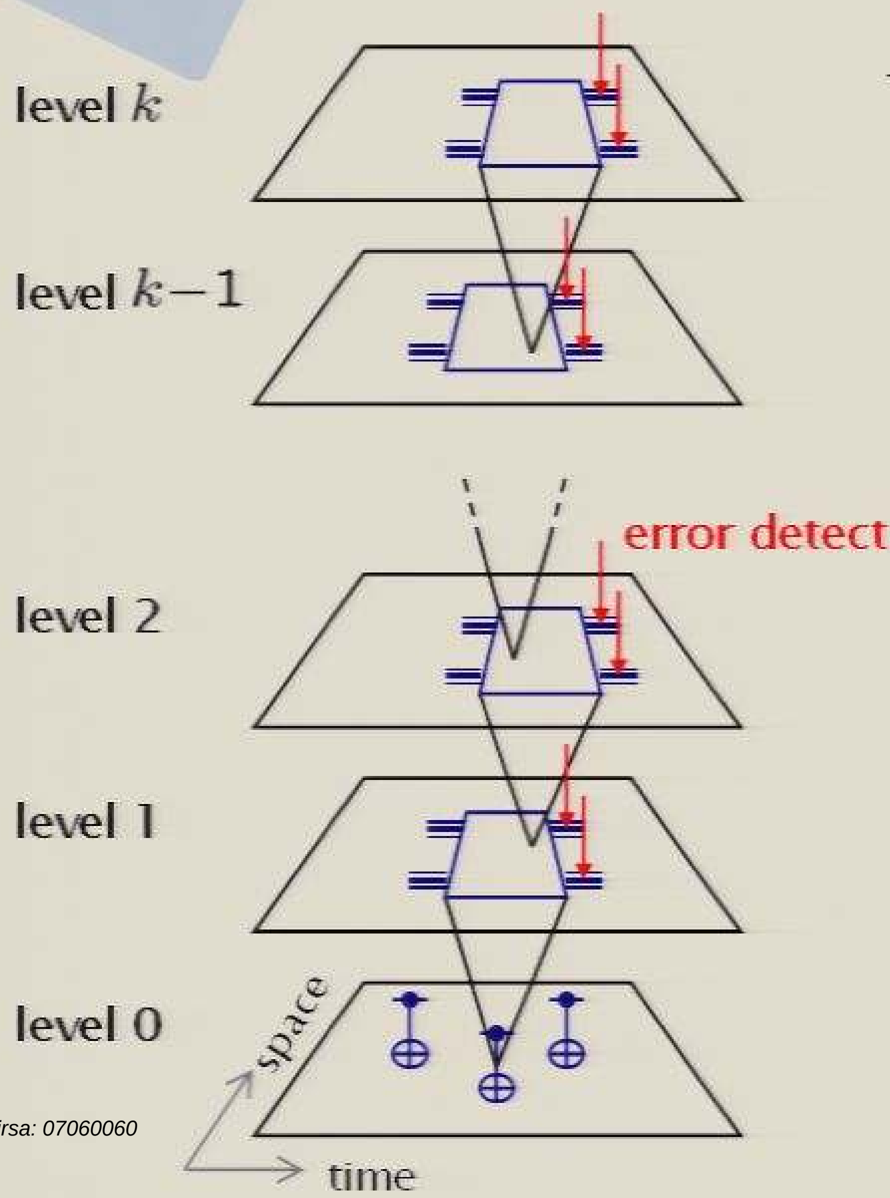
why distance-2 codes? Because circuit is so simple (~50 gates in CNOT-exRec).

- with postselection the threshold is ~3%, while with error correction, ~1%. (based on numerical simulations for independent stochastic Pauli noise)

- analytic lower bounds for the postselection scheme give threshold ~0.1% (for independent stochastic noise)

- interestingly, postselection does *not* work for *adversarial* stochastic noise.

the Fibonacci or C_4/C_6 scheme



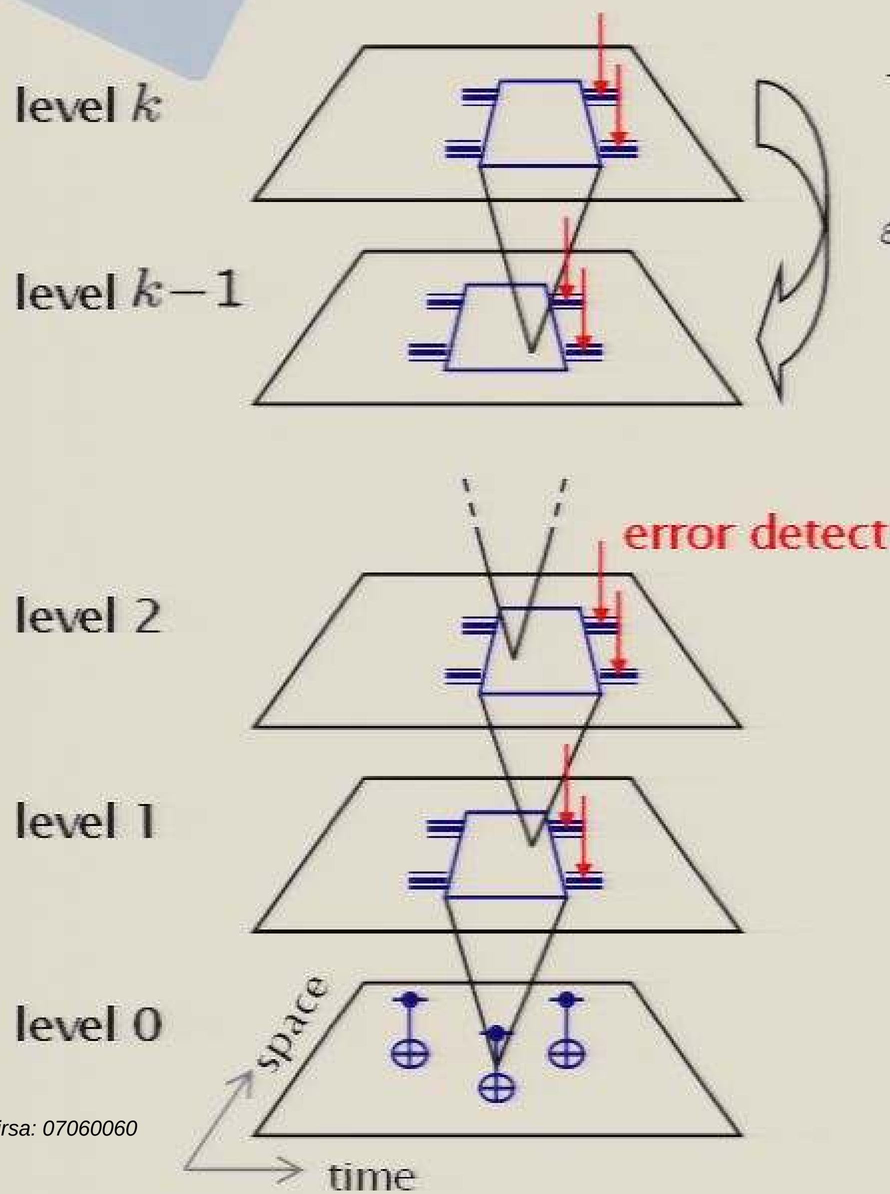
– why Fibonacci?

$$\varepsilon^{(1)}_{\neg f} = O(\varepsilon^2) \qquad \varepsilon^{(1)}_{f} = O(\varepsilon)$$

$$\varepsilon^{(0)}_{\neg f} = \varepsilon \qquad \varepsilon^{(0)}_{f} = 0$$

no located faults (flags) at the physical level

the Fibonacci or C_4/C_6 scheme



– why Fibonacci?

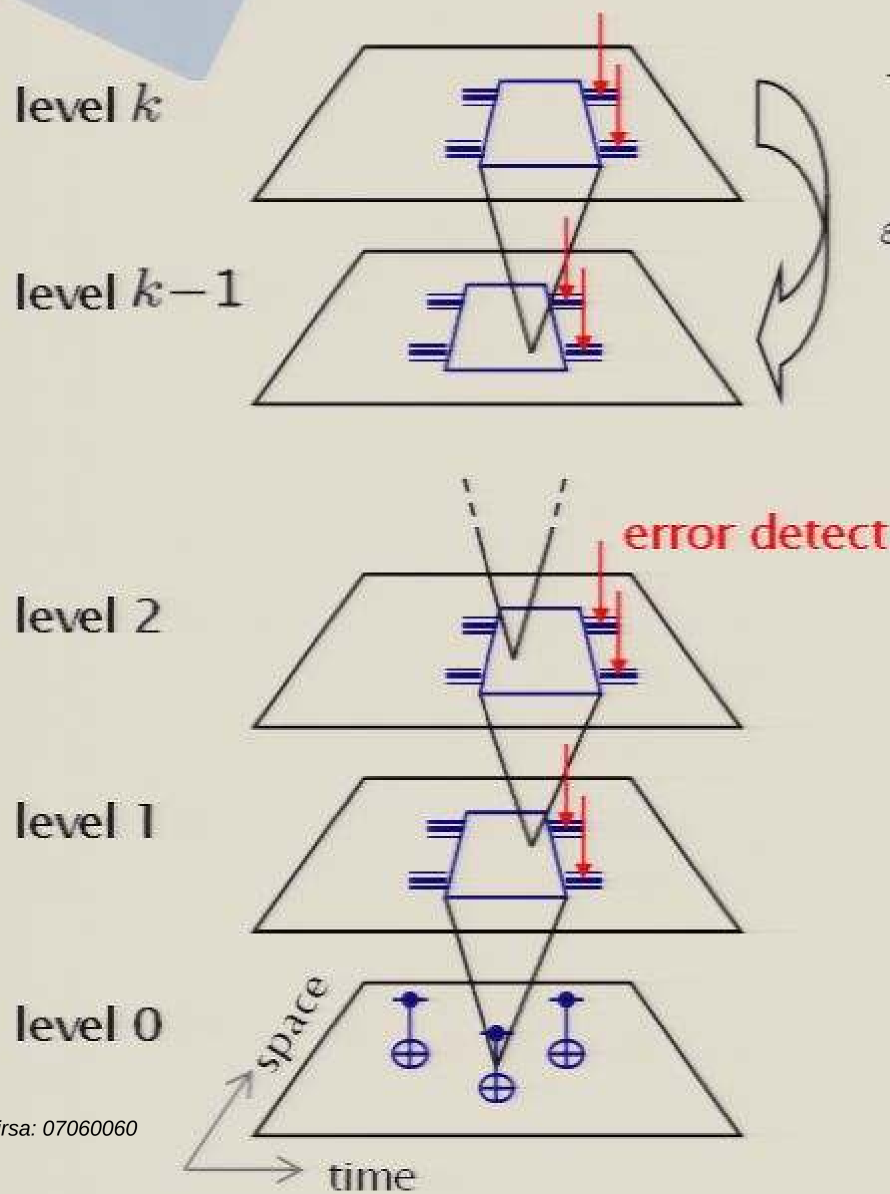
$$\varepsilon_{\neg f}^{(k)} = O\left(\varepsilon_f^{(k-1)}\varepsilon_{\neg f}^{(k-1)}\right)\ ,\ \ \varepsilon_f^{(k)} = O\left(\varepsilon_{\neg f}^{(k-1)}\right)$$

$$\varepsilon_{\neg f}^{(1)} = O(\varepsilon^2) \qquad \varepsilon_f^{(1)} = O(\varepsilon)$$

$$\varepsilon_{\neg f}^{(0)} = \varepsilon \qquad \varepsilon_f^{(0)} = 0$$

no located faults (flags)
at the physical level

the Fibonacci or C_4/C_6 scheme



– why Fibonacci?

$$\varepsilon_{\neg f}^{(k)} = O\left(\varepsilon_f^{(k-1)}\varepsilon_{\neg f}^{(k-1)}\right), \quad \varepsilon_f^{(k)} = O\left(\varepsilon_{\neg f}^{(k-1)}\right)$$

$$\varepsilon_{\neg f}^{(3)} = O(\varepsilon^5) \qquad \varepsilon_f^{(1)} = O(\varepsilon^3)$$

$$\varepsilon_{\neg f}^{(2)} = O(\varepsilon^3) \qquad \varepsilon_f^{(1)} = O(\varepsilon^2)$$

$$\varepsilon_{\neg f}^{(1)} = O(\varepsilon^2) \qquad \varepsilon_f^{(1)} = O(\varepsilon^1)$$

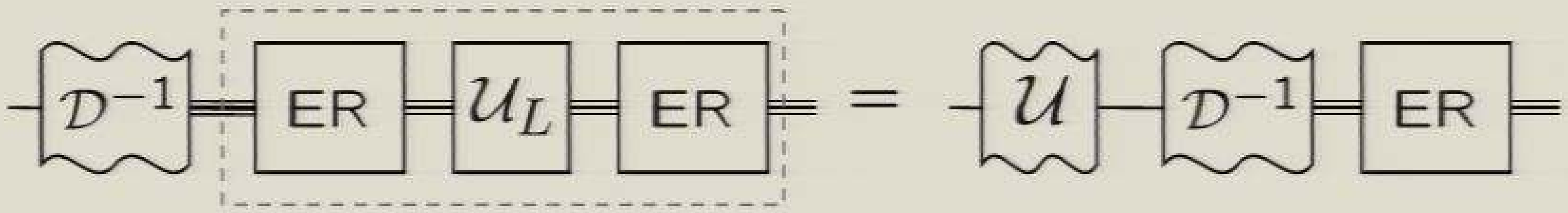
$$\varepsilon_{\neg f}^{(0)} = \varepsilon = \varepsilon^1 \qquad \varepsilon_f^{(0)} = 0$$

no located faults (flags)
at the physical level

the Fibonacci or C_4/C_6 scheme

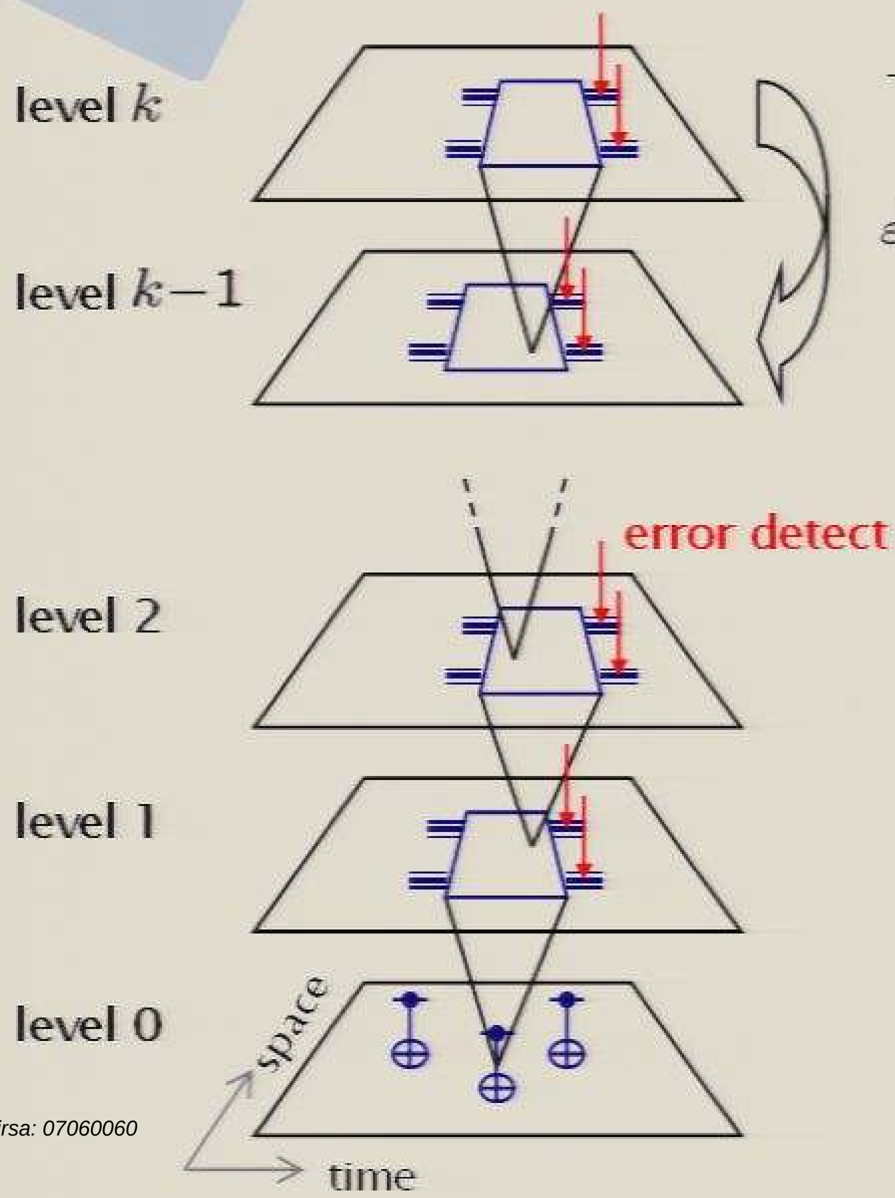
– how should we level reduce? When we use postselection, we may say

<u>lemma.</u> the simulation inside a postselected exRec with ≤ 1 fault is correct, i.e.,



– but in the Fibonacci scheme, the ideal encoder must also know about flags...

the Fibonacci or C_4/C_6 scheme



– why Fibonacci?

$$\varepsilon_{\neg f}^{(k)} = O\left(\varepsilon_f^{(k-1)}\varepsilon_{\neg f}^{(k-1)}\right)\ ,\ \ \varepsilon_f^{(k)} = O\left(\varepsilon_{\neg f}^{(k-1)}\right)$$

$$\varepsilon_{\neg f}^{(3)} = O(\varepsilon^5) \qquad \varepsilon_f^{(1)} = O(\varepsilon^3)$$

$$\varepsilon_{\neg f}^{(2)} = O(\varepsilon^3) \qquad \varepsilon_f^{(1)} = O(\varepsilon^2)$$

$$\varepsilon_{\neg f}^{(1)} = O(\varepsilon^2) \qquad \varepsilon_f^{(1)} = O(\varepsilon^1)$$

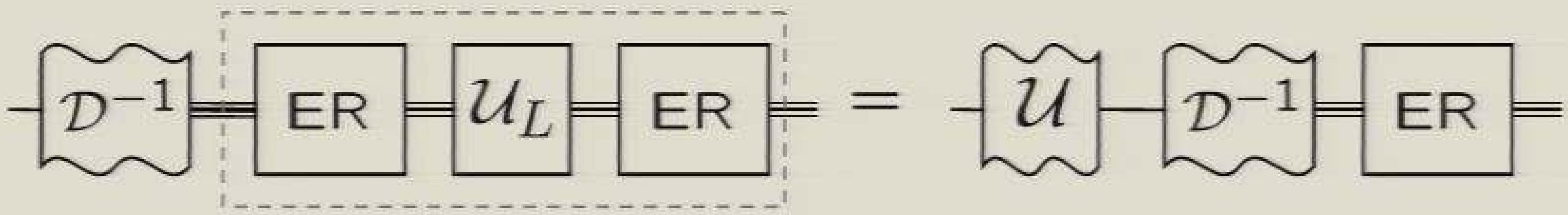
$$\varepsilon_{\neg f}^{(0)} = \varepsilon = \varepsilon^1 \qquad \varepsilon_f^{(0)} = 0$$

no located faults (flags)
at the physical level

the Fibonacci or C_4/C_6 scheme

– how should we level reduce? When we use postselection, we may say

<u>lemma.</u> the simulation inside a postselected exRec with ≤ 1 fault is correct, i.e.,

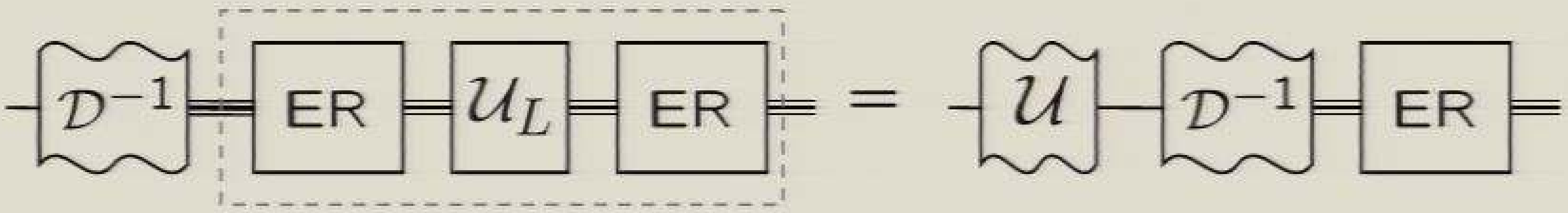


– but in the Fibonacci scheme, the ideal encoder must also know about flags…

the Fibonacci or C_4/C_6 scheme

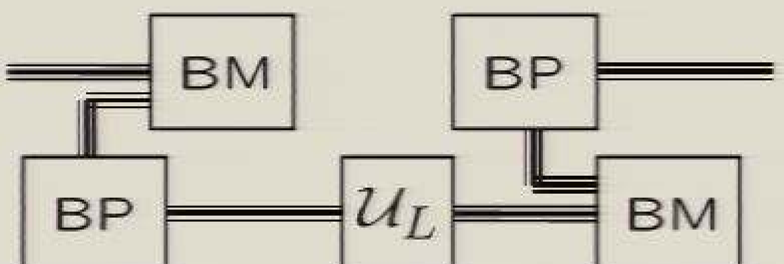
– how should we level reduce? When we use postselection, we may say

<u>lemma.</u> the simulation inside a postselected exRec with ≤ 1 fault is correct, i.e.,



– but in the Fibonacci scheme, the ideal encoder must also know about flags…

– instead, in our exRecs we can take all noise to be in the BMs



so, we may say that level reduction results in a faulty gate if no logical error occurs in the *trailing* BMs.

the Fibonacci or C_4/C_6 scheme

– we use the 4–qubit code stabilized by $(\sigma_X)^{\otimes 4}$ and $(\sigma_Z)^{\otimes 4}$.

– any nontrivial syndrome is ambiguous;
we decide by convention about recovery and flag to the next coding level

<u>flagging rules.</u>

if we measure $\begin{cases} (\sigma_X)^{\otimes 4} = -1 \\ (\sigma_Z)^{\otimes 4} = -1 \end{cases}$, we say $\begin{cases} \sigma_Z \otimes I^{\otimes 3} \\ \sigma_X \otimes I^{\otimes 3} \end{cases}$ happened and raise a flag.

we also raise a flag if exRec contains ≥ 2 flagged operations.

the Fibonacci or C_4/C_6 scheme

– we use the 4–qubit code stabilized by $(\sigma_X)^{\otimes 4}$ and $(\sigma_Z)^{\otimes 4}$.

– any nontrivial syndrome is ambiguous;
we decide by convention about recovery and flag to the next coding level

<u>flagging rules.</u>

if we measure $\begin{cases} (\sigma_X)^{\otimes 4} = -1 \\ (\sigma_Z)^{\otimes 4} = -1 \end{cases}$, we say $\begin{cases} \sigma_Z \otimes I^{\otimes 3} \\ \sigma_X \otimes I^{\otimes 3} \end{cases}$ happened and raise a flag.

we also raise a flag if exRec contains ≥ 2 flagged operations.

the Fibonacci or C_4/C_6 scheme

– we use the 4–qubit code stabilized by $(\sigma_X)^{\otimes 4}$ and $(\sigma_Z)^{\otimes 4}$.

– any nontrivial syndrome is ambiguous;
we decide by convention about recovery and flag to the next coding level

<u>flagging rules.</u>

if we measure $\begin{cases}(\sigma_X)^{\otimes 4} = -1 \\ (\sigma_Z)^{\otimes 4} = -1\end{cases}$, we say $\begin{cases}\sigma_Z \otimes I^{\otimes 3} \\ \sigma_X \otimes I^{\otimes 3}\end{cases}$ happened and raise a flag.

we also raise a flag if exRec contains ≥ 2 flagged operations.

– recall noise model.

<u>def.</u> noise is *adversarial stochastic* if the probability of fault paths with faults at r specific noisy gates is at most ε^r for some constant *strength* ε.

the Fibonacci or C_4/C_6 scheme

– first level reduction.

prob of a faulty unflagged op.: $\varepsilon_{\neg f}^{(1)} \leq C_2\,\varepsilon^2$

prob of a faulty flagged op.: $\varepsilon_{f}^{(1)} \leq C_1\,\varepsilon + D_1\,\varepsilon^2$

prob of a flag: $f^{(1)} \leq C_0\,\varepsilon$

– now, noise model includes flagged operations which occur with prob $f^{(1)}$ and fail with prob $\varepsilon_f^{(1)}$.

the Fibonacci or C_4/C_6 scheme

– first level reduction. prob of a faulty unflagged op.: $\varepsilon_{\neg f}^{(1)} \leq C_2\,\varepsilon^2$

prob of a faulty flagged op.: $\varepsilon_{f}^{(1)} \leq C_1\,\varepsilon + D_1\,\varepsilon^2$

prob of a flag: $f^{(1)} \leq C_0\,\varepsilon$

– now, noise model includes flagged operations which occur with prob $f^{(1)}$ and fail with prob $\varepsilon_f^{(1)}$.

– after the k–th level reduction,

$$\varepsilon_{\neg f}^{(k)} \leq \max\{\underbrace{C_2'\left(\varepsilon_{\neg f}^{(k-1)}\right)^2}_{\text{no flags}},\ \underbrace{\varepsilon_f^{(k-1)} \cdot N\,\varepsilon_{\neg f}^{(k-1)} + C_2'\left(\varepsilon_{\neg f}^{(k-1)}\right)^2}_{\text{one flag}}\}$$

the Fibonacci or C_4/C_6 scheme

– first level reduction.

prob of a faulty unflagged op.: $\varepsilon_{\neg f}^{(1)} \leq C_2\,\varepsilon^2$

prob of a faulty flagged op.: $\varepsilon_{f}^{(1)} \leq C_1\,\varepsilon + D_1\,\varepsilon^2$

prob of a flag: $f^{(1)} \leq C_0\,\varepsilon$

– now, noise model includes flagged operations which occur with prob $f^{(1)}$ and fail with prob $\varepsilon_f^{(1)}$.

– after the k–th level reduction,

$$\varepsilon_{\neg f}^{(k)} \leq \max\{\underbrace{C_2'\left(\varepsilon_{\neg f}^{(k-1)}\right)^2}_{\text{no flags}},\ \underbrace{\varepsilon_f^{(k-1)}\cdot N\,\varepsilon_{\neg f}^{(k-1)} + C_2'\left(\varepsilon_{\neg f}^{(k-1)}\right)^2}_{\text{one flag}}\}$$

$$\varepsilon_{f}^{(k)} \leq \max\{\underbrace{C_1'\varepsilon_{\neg f}^{(k-1)} + D_1'\left(\varepsilon_{\neg f}^{(k-1)}\right)^2}_{\text{no flags}},\ \underbrace{C_0'\,\varepsilon_{\neg f}^{(k-1)}}_{\text{one flag}},\ \underbrace{\left(f^{(k-1)}\right)^2}_{\text{at least two flags}}\}$$

$$f^{(k)} \leq \max\{\underbrace{C_0'\,\varepsilon_{\neg f}^{(k-1)}}_{\text{no flags}},\ \underbrace{\varepsilon_f^{(k-1)} + C_0'\,\varepsilon_{\neg f}^{(k-1)}}_{\text{one flag}},\ \underbrace{\left(f^{(k-1)}\right)^2}_{\text{at least two flags}}\}$$

the Fibonacci or C_4/C_6 scheme

– first level reduction.

prob of a faulty unflagged op.: $\varepsilon_{\neg f}^{(1)} \leq C_2\,\varepsilon^2$

prob of a faulty flagged op.: $\varepsilon_{f}^{(1)} \leq C_1\,\varepsilon + D_1\,\varepsilon^2$

prob of a flag: $f^{(1)} \leq C_0\,\varepsilon$

– now, noise model includes flagged operations which occur with prob $f^{(1)}$ and fail with prob $\varepsilon_f^{(1)}$.

– after the k–th level reduction,

$$\varepsilon_{\neg f}^{(k)} \leq \max\{\underbrace{C_2'\left(\varepsilon_{\neg f}^{(k-1)}\right)^2}_{\text{no flags}}\ ,\ \underbrace{\varepsilon_f^{(k-1)}\cdot N\,\varepsilon_{\neg f}^{(k-1)} + C_2'\left(\varepsilon_{\neg f}^{(k-1)}\right)^2}_{\text{one flag}}\}$$

$$C_2' \to f^{(k-1)}C_2' + \text{corr.}$$

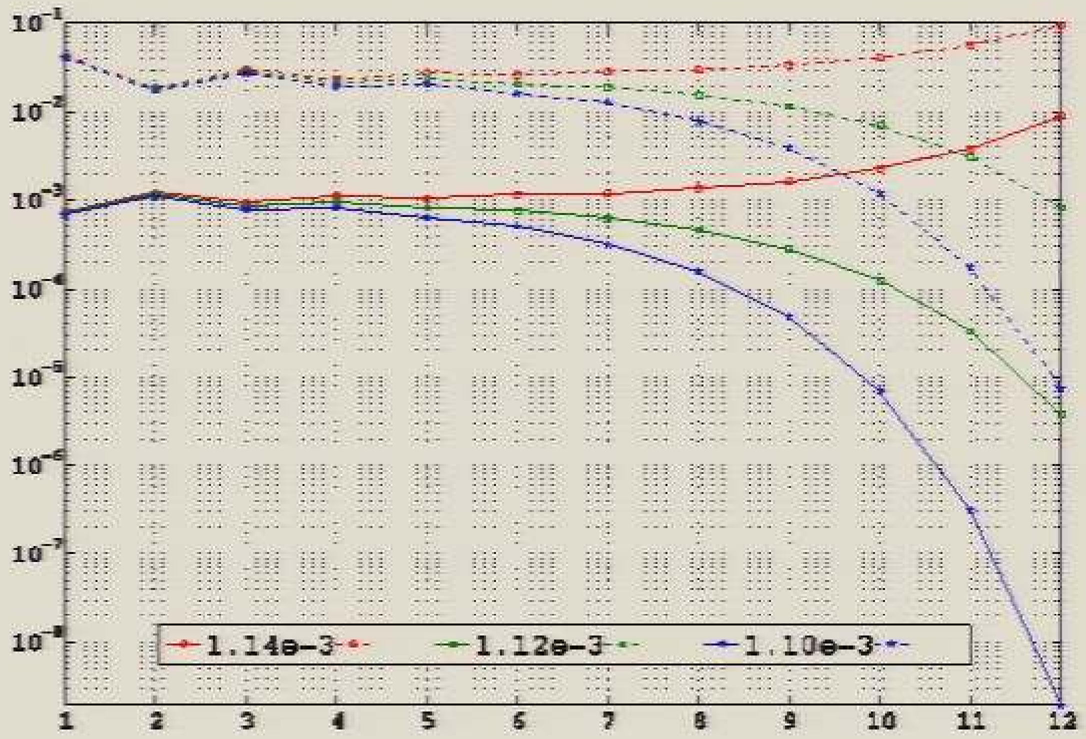
$$\varepsilon_{f}^{(k)} \leq \max\{\underbrace{C_1'\varepsilon_{\neg f}^{(k-1)} + D_1'\left(\varepsilon_{\neg f}^{(k-1)}\right)^2}_{\text{no flags}}\ ,\ \underbrace{C_0'\,\varepsilon_{\neg f}^{(k-1)}}_{\text{one flag}}\ ,\ \underbrace{\left(f^{(k-1)}\right)^2}_{\text{at least two flags}}\}$$

$$C_0' \to f^{(k-1)}C_0' + \text{corr.}$$

$$f^{(k)} \leq \max\{\underbrace{C_0'\,\varepsilon_{\neg f}^{(k-1)}}_{\text{no flags}}\ ,\ \underbrace{\varepsilon_f^{(k-1)} + C_0'\,\varepsilon_{\neg f}^{(k-1)}}_{\text{one flag}}\ ,\ \underbrace{\left(f^{(k-1)}\right)^2}_{\text{at least two flags}}\}$$

the Fibonacci or C_4/C_6 scheme

– counting to get constants for CNOT-exRec, iterating, and plotting



– so for "CSS operations," i.e., 0, + prep., X and Z meas., H and CNOT, 1.12×10^{-3} is below the threshold.

– for universalily, we also need the $S \equiv \exp(-i\frac{\pi}{4}\sigma z)$ and $T \equiv S^{1/2}$.

the Fibonacci or C_4/C_6 scheme

– for $\begin{cases} S \equiv \exp(-i\frac{\pi}{4}\sigma z) \\ T \equiv S^{1/2} \end{cases}$, it is sufficient to prepare $\begin{cases} |+i\rangle \propto |0\rangle + i|1\rangle \\ |H\rangle = \cos\left(\frac{\pi}{4}\right)|0\rangle + \sin\left(\frac{\pi}{4}\right)|1\rangle \end{cases}$

– idea: prepare single–qubit copies and teleport with a Bell state that is half a qubit and half a code block.

– noise is dominated by the accuracy of decoding a block to a qubit,

$$\varepsilon_{\neg f}^{\text{dec},(k)} \leq 3\% \qquad \varepsilon_{f}^{\text{dec},(k)} \leq 70\%$$

– if we repeat in parallel, with high probability we have an "unflagged" ancilla and its noise is well below the distillation thresholds (~10%)

the Fibonacci or C_4/C_6 scheme

– for $\begin{cases} S \equiv \exp(-i\frac{\pi}{4}\sigma_Z) \\ T \equiv S^{1/2} \end{cases}$, it is sufficient to prepare $\begin{cases} |+i\rangle \propto |0\rangle + i|1\rangle \\ |H\rangle = \cos\left(\frac{\pi}{4}\right)|0\rangle + \sin\left(\frac{\pi}{4}\right)|1\rangle \end{cases}$

– idea: prepare single–qubit copies and teleport with a Bell state that is half a qubit and half a code block.

– noise is dominated by the accuracy of decoding a block to a qubit,

$$\varepsilon_{\neg f}^{\text{dec},(k)} \le 3\% \qquad \varepsilon_{f}^{\text{dec},(k)} \le 70\%$$

– if we repeat in parallel, with high probability we have an "unflagged" ancilla and its noise is well below the distillation thresholds (~10%)

– so 1.12×10^{-3} is the threshold of universal quantum computation for adversarial stochastic noise! *(no geometry, fast meas., robust PCs)*

– we didn't use postselection, but we used ~10 coding levels; one logical qubit is encoded in a block of ~4^{10}~10^6 physical qubits (C_4/C_6 scheme is more efficient, but is too complex for me and is still bad).

the Fibonacci or C_4/C_6 scheme

– for $\begin{cases} S \equiv \exp(-i\frac{\pi}{4}\sigma z) \\ T \equiv S^{1/2} \end{cases}$, it is sufficient to prepare $\begin{cases} |+i\rangle \propto |0\rangle + i|1\rangle \\ |H\rangle = \cos\left(\frac{\pi}{4}\right)|0\rangle + \sin\left(\frac{\pi}{4}\right)|1\rangle \end{cases}$

– idea: prepare single–qubit copies and teleport with a Bell state that is half a qubit and half a code block.

– noise is dominated by the accuracy of decoding a block to a qubit,

$$\varepsilon_{\neg f}^{\text{dec},(k)} \leq 3\% \qquad \varepsilon_{f}^{\text{dec},(k)} \leq 70\%$$

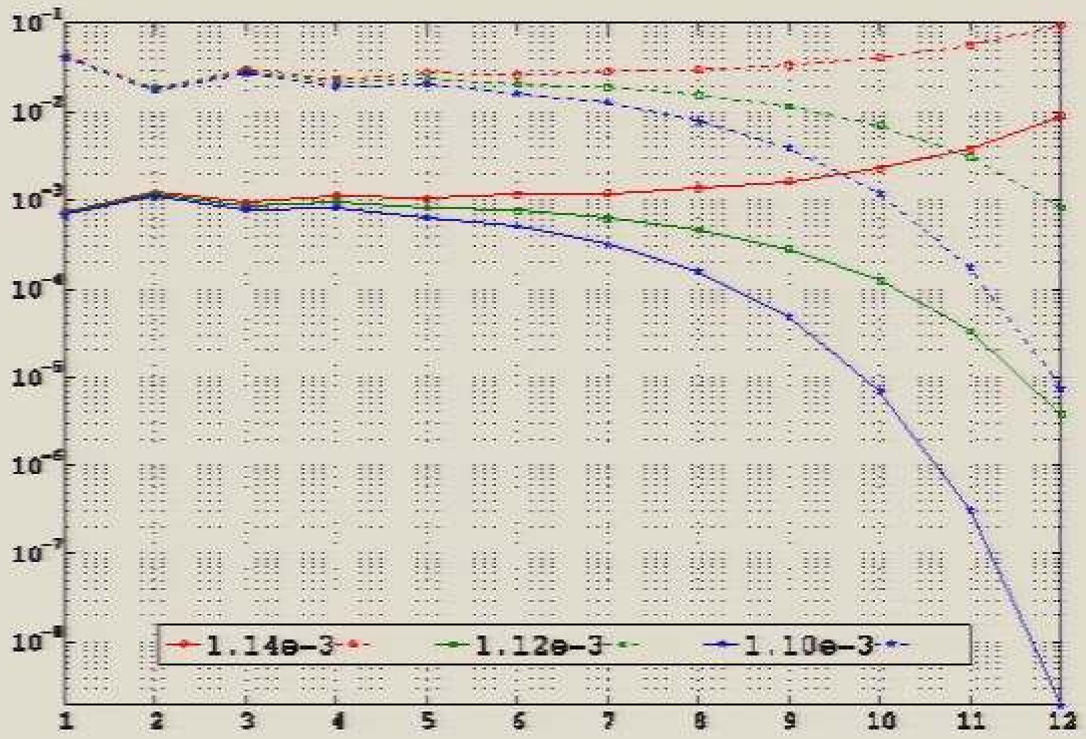
– if we repeat in parallel, with high probability we have an "unflagged" ancilla and its noise is well below the distillation thresholds (~10%)

– so 1.12×10^{-3} is the threshold of universal quantum computation for adversarial stochastic noise! *(no geometry, fast meas., robust PCs)*

– we didn't use postselection, but we used ~10 coding levels; one logical qubit is encoded in a block of $\sim 4^{10} \sim 10^{6}$ physical qubits (C_4/C_6 scheme is more efficient, but is too complex for me and is still bad).

the Fibonacci or C_4/C_6 scheme

– counting to get constants for CNOT–exRec, iterating, and plotting



– so for "CSS operations," i.e., 0, + prep., X and Z meas., H and CNOT, 1.12×10^{-3} is below the threshold.

– for universalily, we also need the $S \equiv \exp(-i\frac{\pi}{4}\sigma z)$ and $T \equiv S^{1/2}$.

the Fibonacci or C_4/C_6 scheme

– for $\begin{cases} S \equiv \exp(-i\frac{\pi}{4}\sigma z) \\ T \equiv S^{1/2} \end{cases}$, it is sufficient to prepare $\begin{cases} |+i\rangle \propto |0\rangle + i|1\rangle \\ |H\rangle = \cos\left(\frac{\pi}{4}\right)|0\rangle + \sin\left(\frac{\pi}{4}\right)|1\rangle \end{cases}$

– idea: prepare single–qubit copies and teleport with a Bell state that is half a qubit and half a code block.

– noise is dominated by the accuracy of decoding a block to a qubit,

$$\varepsilon_{\neg f}^{\text{dec},(k)} \leq 3\% \qquad \varepsilon_{f}^{\text{dec},(k)} \leq 70\%$$

– if we repeat in parallel, with high probability we have an "unflagged" ancilla and its noise is well below the distillation thresholds (~10%)

– so 1.12×10^{-3} is the threshold of universal quantum computation for adversarial stochastic noise! *(no geometry, fast meas., robust PCs)*

– we didn't use postselection, but we used ~10 coding levels; one logical qubit is encoded in a block of $\sim 4^{10} \sim 10^6$ physical qubits (C_4/C_6 scheme is more efficient, but is too complex for me and is still bad).

message-passing decoding for computation

– say we use an [[n,1,d=2t+1]] code, we concatenate k times, and level reduce naively (i.e., decode iteratively); then,

$$\varepsilon^{(k)} \equiv \varepsilon_0 \times \left(\frac{\varepsilon}{\varepsilon_0}\right)^{(t+1)^k}$$

– to achieve accuracy 1/L we need an overhead (i.e., # physical ops./logical op.)

$$A^k = (t+1)^{k \log_{t+1} A} = O\left((\log L)^{\log_{t+1} A}\right)$$

where A is # of locs in largest Rec ($\log_2 81 \approx 6.4$ for d=3 BS code).

message-passing decoding for computation

– say we use an [[n,1,d=2t+1]] code, we concatenate k times, and level reduce naively (i.e., decode iteratively); then,

$$\varepsilon^{(k)} \equiv \varepsilon_0 \times \left(\frac{\varepsilon}{\varepsilon_0}\right)^{(t+1)^k}$$

– to achieve accuracy 1/L we need an overhead (i.e., # physical ops./logical op.)

$$A^k = (t+1)^{k \log_{t+1} A} = O\left((\log L)^{\log_{t+1} A}\right)$$

where A is # of locs in largest Rec ($\log_2 81 \approx 6.4$ for d=3 BS code).

– can we do better?

with the toric code in 4D, no concatenation, and a local recovery procedure, Ahn & Preskill can get an overhead $O\left((\log L)^4\right)$.

message-passing decoding for computation

- say we use an [[n,1,d=2t+1]] code, we concatenate k times, and level reduce naively (i.e., decode iteratively); then,

$$\varepsilon^{(k)} \equiv \varepsilon_0 \times \left(\frac{\varepsilon}{\varepsilon_0}\right)^{(t+1)^k}$$

- to achieve accuracy 1/L we need an overhead (i.e., # physical ops./logical op.)

$$A^k = (t+1)^{k \log_{t+1} A} = O\left((\log L)^{\log_{t+1} A}\right)$$

where A is # of locs in largest Rec ($\log_2 81 \approx 6.4$ for d=3 BS code).

- can we do better?

with the toric code in 4D, no concatenation, and a local recovery procedure, Ahn & Preskill can get an overhead $O\left((\log L)^4\right)$.

- a concatenated [[n,1,d=2t+1]] code has distance $\geq d^k$

so, scaling could be $\varepsilon^{(k)} \propto \varepsilon^{d^k/2}$, and the overhead $O\left((\log L)^{\log_d A}\right)$.

note: $\log_3 81 = 4$, which echoes Ahn & Preskill's scaling!

- is this possible to get by level reducing using a "flag"–passing algorithm?

references

threshold theorem & level reduction

PA, Gottesman, and Preskill, quant-ph/0504218,

and quant-ph/0703264,

and my thesis, quant-ph/0703230

Error-correction by teleportation

Knill, quant-ph/0410199

& the Fibonacci scheme

Reichardt, thesis, quant-ph/0612004

message-passing algorithms

Poulin, quant-ph/0606126

Rahn, Doherty, Mabuchi, quant-ph/0206061

Microsoft PowerPoint - [FTQCII_june2007.ppt]

references

threshold theorem & level reduction

PA, Gottesman, and Preskill, quant-ph/0504218,
and quant-ph/0703264,


and my thesis, quant-ph/0703230

Error-correction by teleportation

Knill, quant-ph/0410199 & the Fibonacci scheme

Reichardt, thesis, quant-ph/0612004

message-passing algorithms

Poulin, quant-ph/0606126

Rahn, Doherty, Mabuchi, quant-ph/0206061

Slide 36 of 36 Default Design English (U.S.)