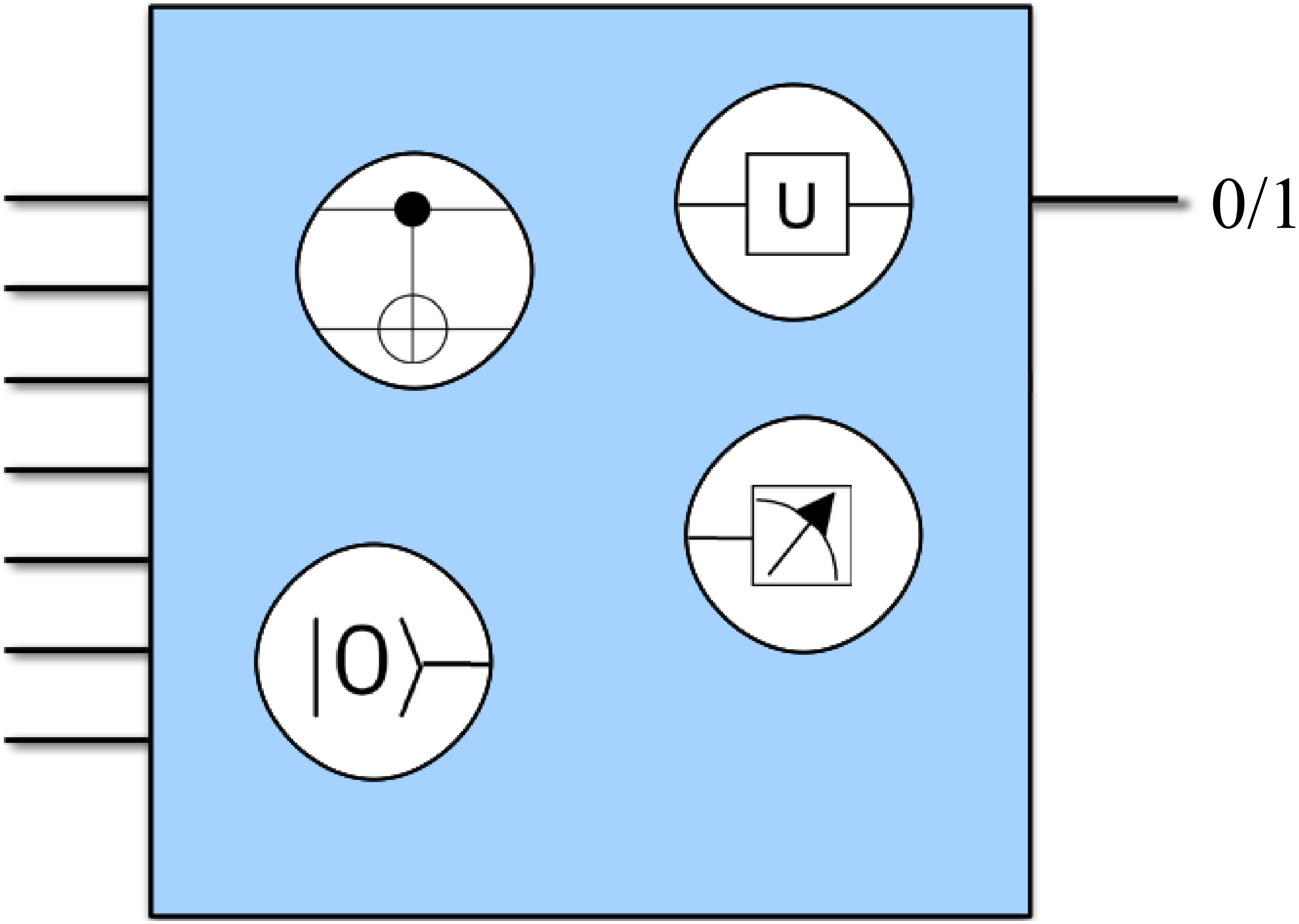
Rigorous fault-tolerance thresholds

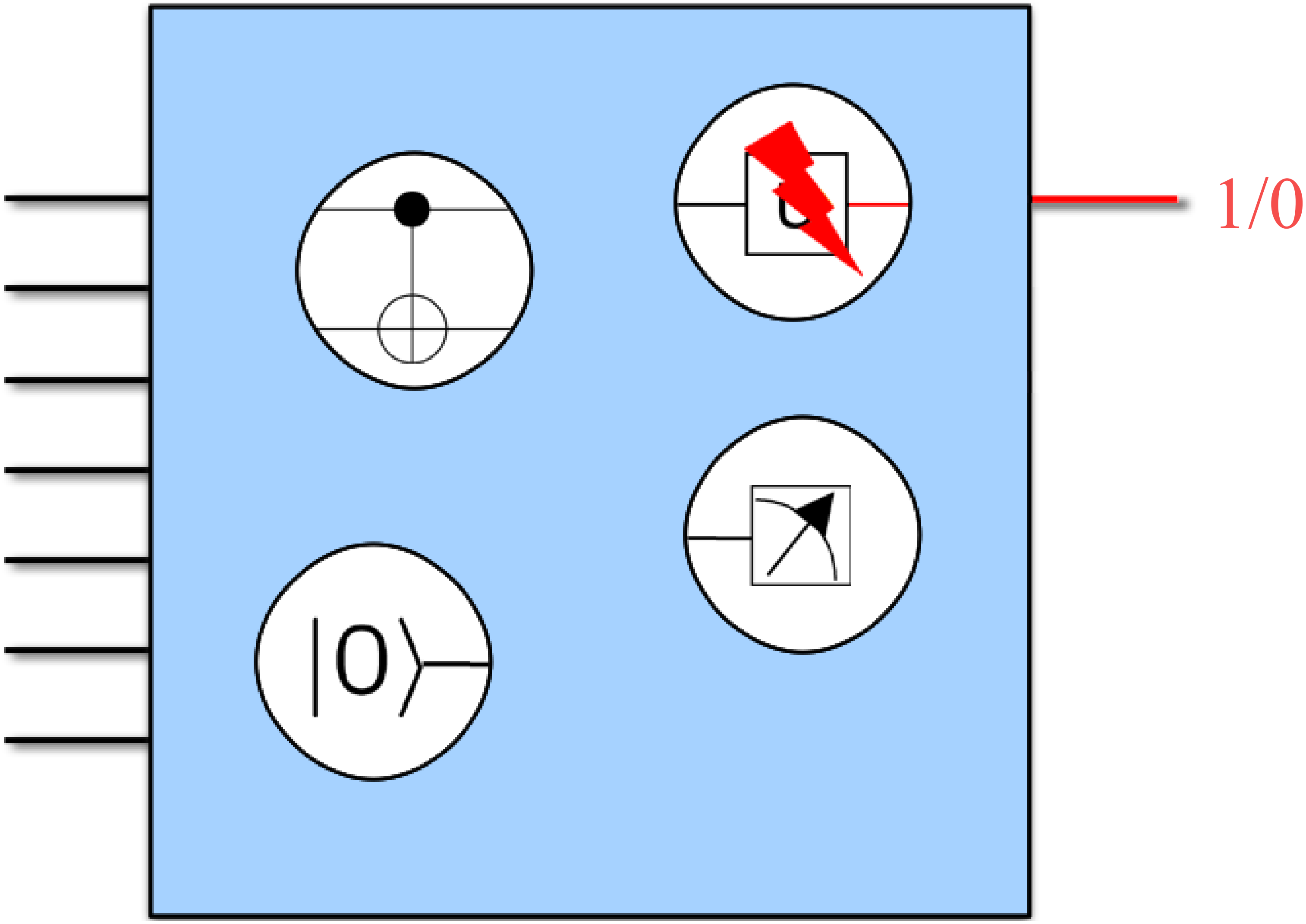
Ben Reichardt

UC Berkeley

N gate circuit

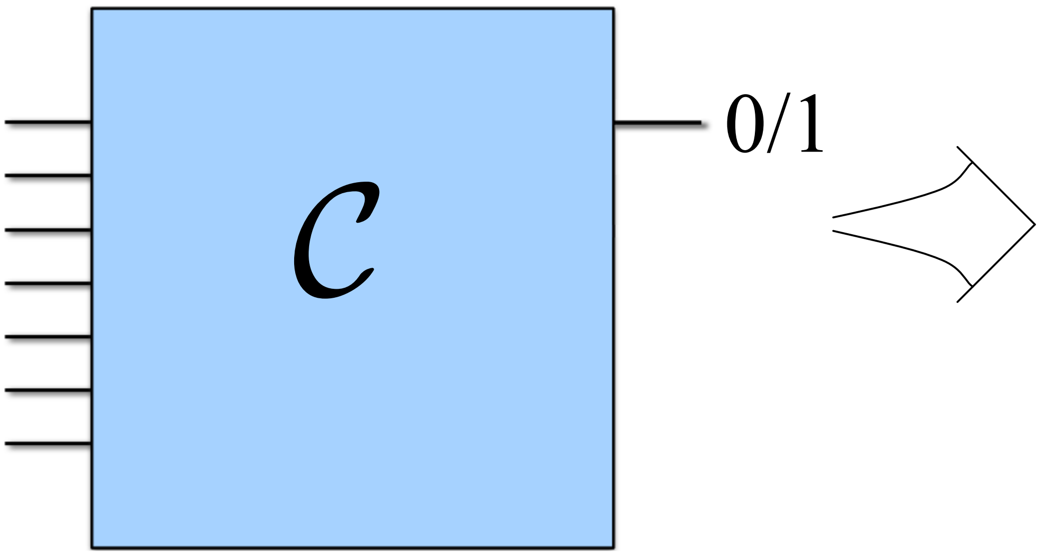


N gate circuit ⇒ Need error $\ll 1/N$



Quantum fault-tolerance problem

– Classical fault-tolerance: Von Neumann (1956)

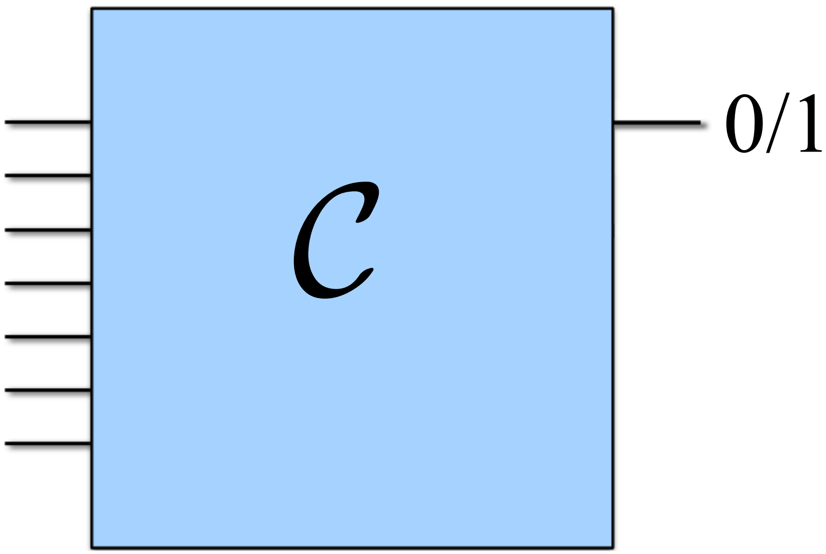


Fault-tolerant, larger $\mathcal{C}$

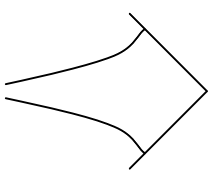
- High tolerable noise
- Low overhead

Quantum fault-tolerance problem

– Classical fault-tolerance: Von Neumann (1956)



Important problem!



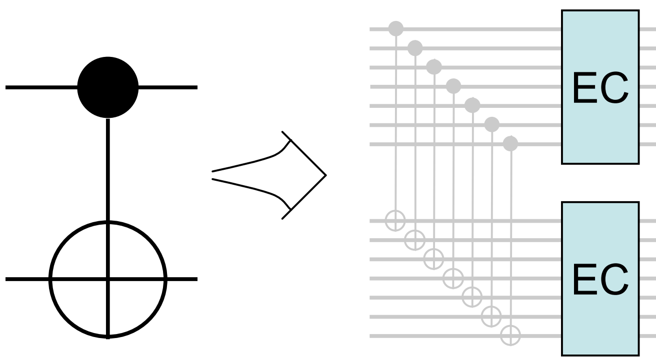
Fault-tolerant, larger

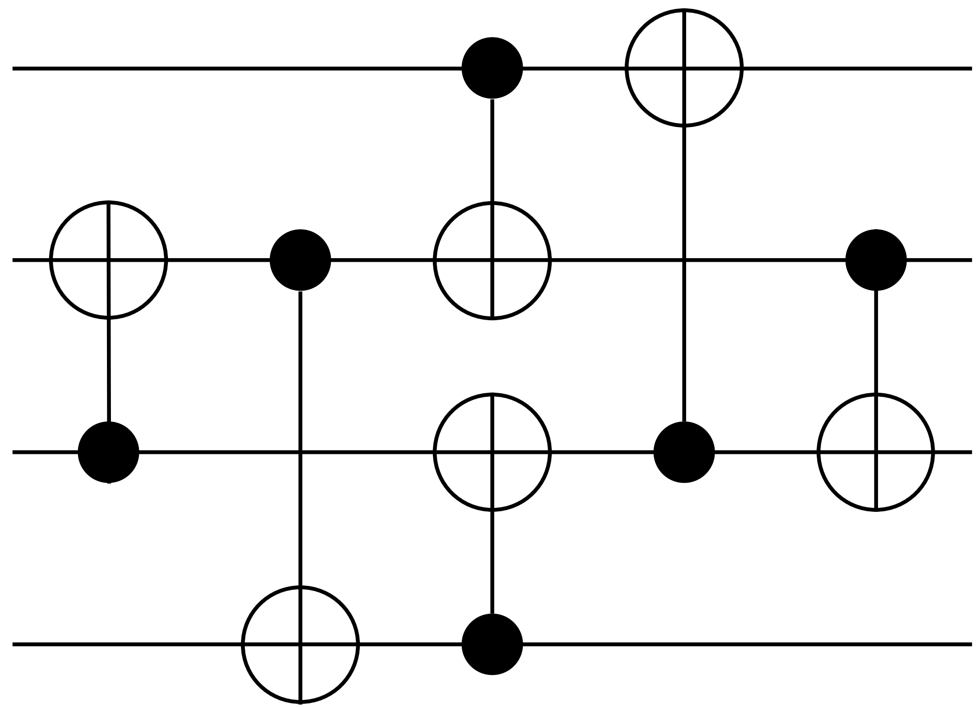
$\mathcal{C}$

- High tolerable noise
- Low overhead

Intuition

- Work on encoded data
- Correct errors to prevent spread
- Concatenate procedure for arbitrary reliability

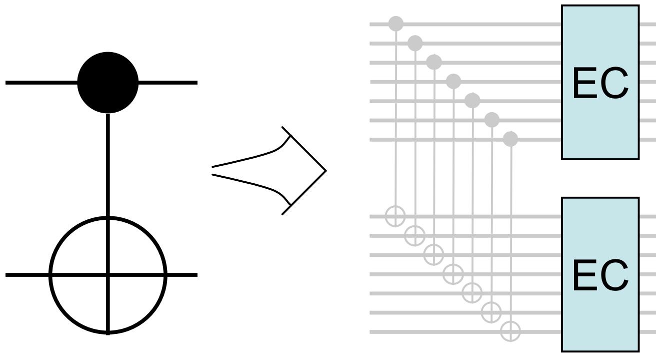


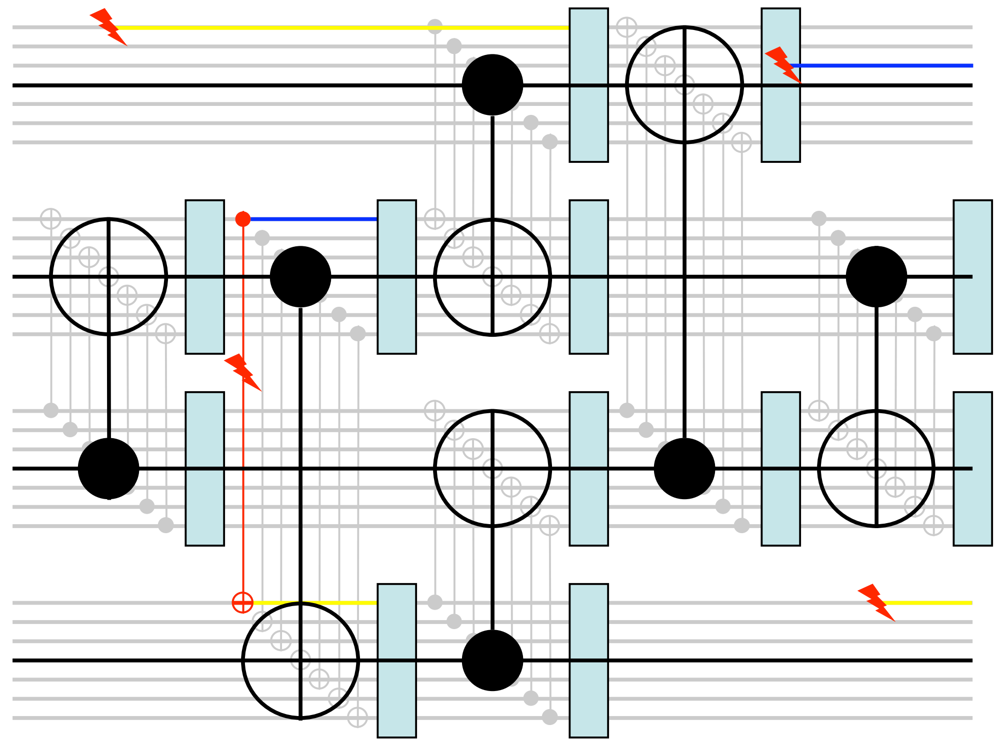


- Quantum fault-tolerance: Shor (1996)
  - Using poly(log N)-sized code, tolerate 1/poly(log N) error
- Aharonov & Ben-Or ('97), Kitaev ('97), Knill-Laflamme-Zurek ('97)
  - Using concatenated constant-sized code, tolerate constant error

Intuition

- Work on encoded data
- Correct errors to prevent spread
- Concatenate procedure for arbitrary reliability





- Quantum fault-tolerance: Shor (1996)
  - – Using poly(log N)-sized code, tolerate 1/poly(log N) error
- Aharonov & Ben-Or (‘97), Kitaev (‘97), Knill-Laflamme-Zurek (‘97)
  - – Using concatenated constant-sized code, tolerate constant error

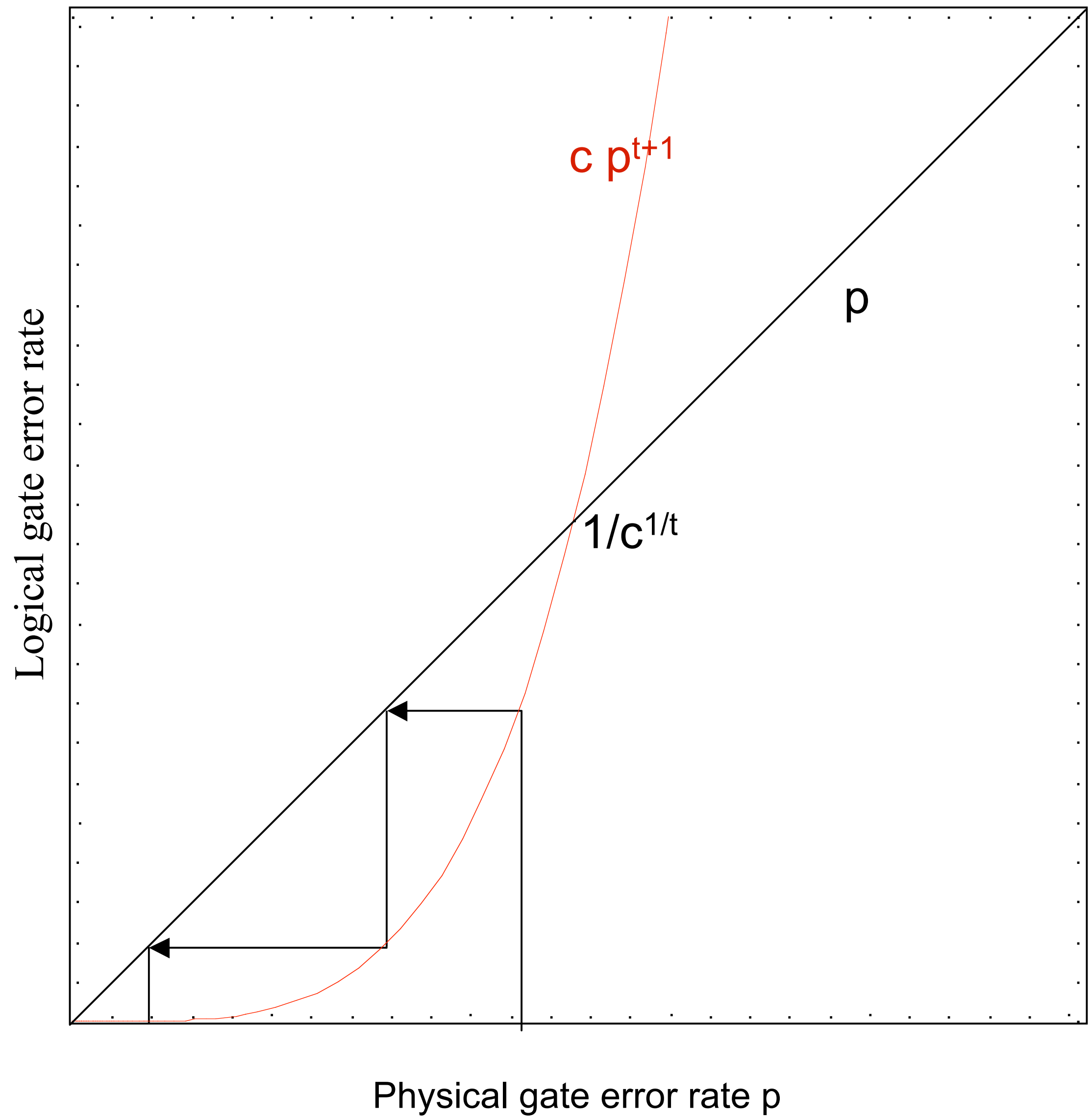
Concatenation

- N gate circuit
  - $\Rightarrow$ Want error $\ll 1/N$

- m-qubit, t-error correcting code

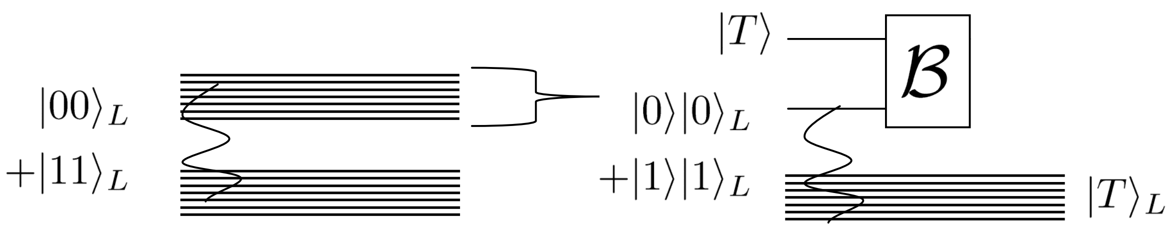
Probability of error	Physical bits per logical bit
p	1
$c\,p^{t+1}$	m
$\sim p^{(t+1)^2}$	m^2
$p^{(t+1)^3}$	m^3

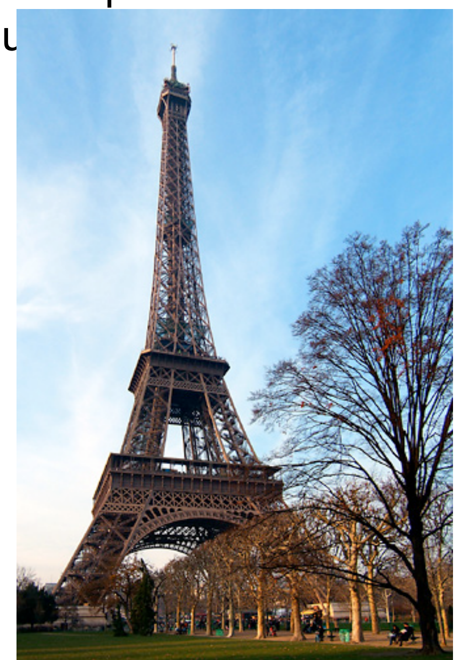
O(log log N) concatenations
poly(log N) physical bits / logical

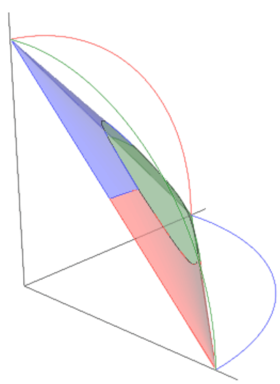


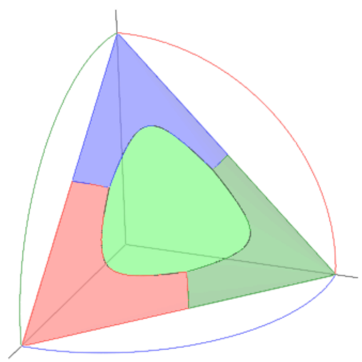
Recent results

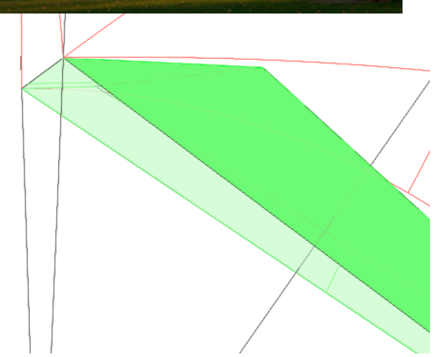
- Magic states distillation [Bravyi & Kitaev '04, Knill '04]
  - Universality method, related to best current threshold upper bounds
  - Reduction Universal fault-tolerance ⇝ Stabilizer op. fau











Recent results

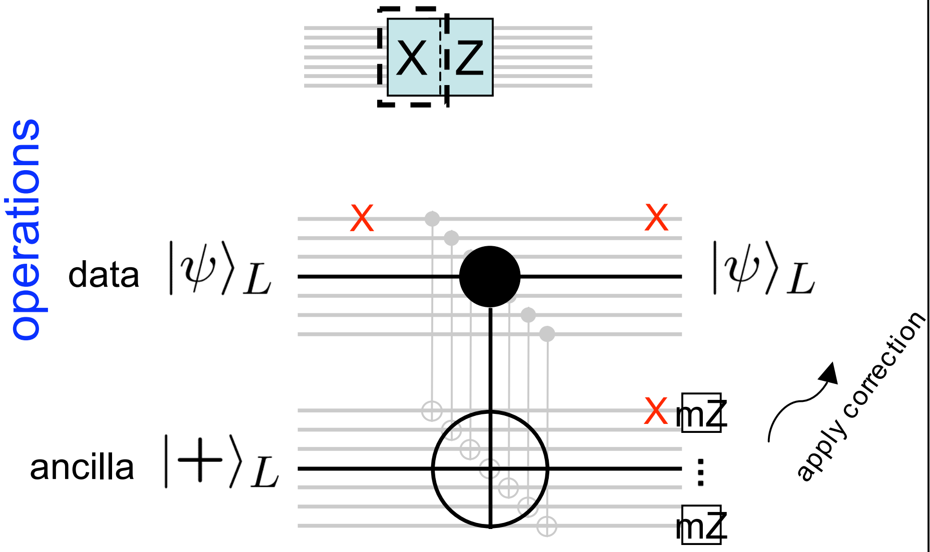
- Magic states distillation [Bravyi & Kitaev '04, Knill '04]
  - Universality method, related to best current threshold upper bounds
  - Reduction from FT universality to FT stabilizer operations
- Optimized fault-tolerance schemes: [Knill '03]
  - Erasure error threshold is 1/2 for Bell measurements
  - [Knill '05]: > ~~5%~~ estimated threshold for depolarizing noise
    1% with substantial but more reasonable overhead

Fault-tolerance threshold myth:

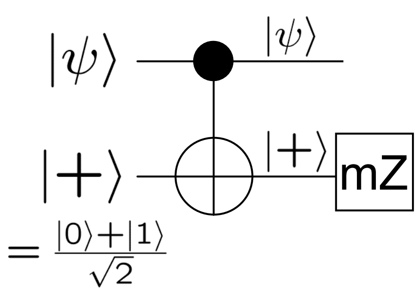
Threshold is all that counts.
Maximize the threshold at all costs.

Steane-type error correction

Physical operations



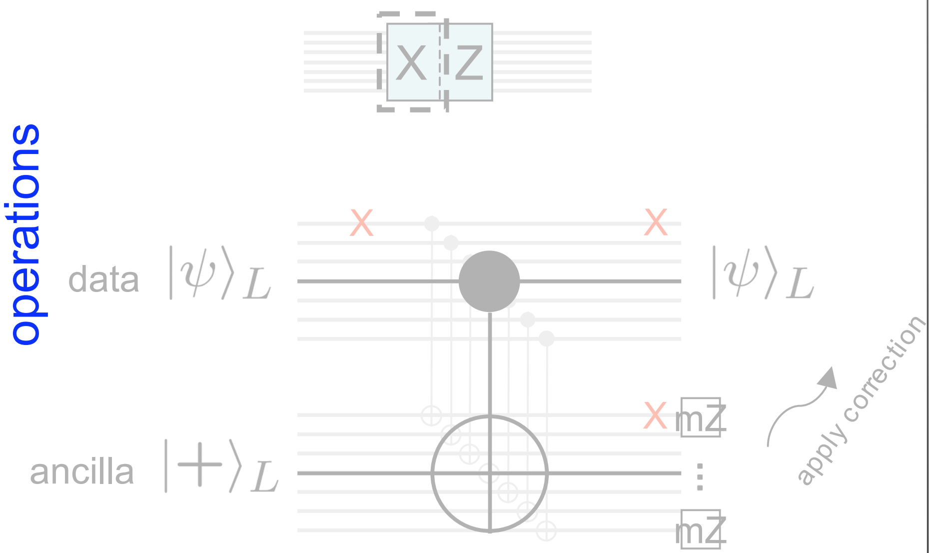
Logical operations

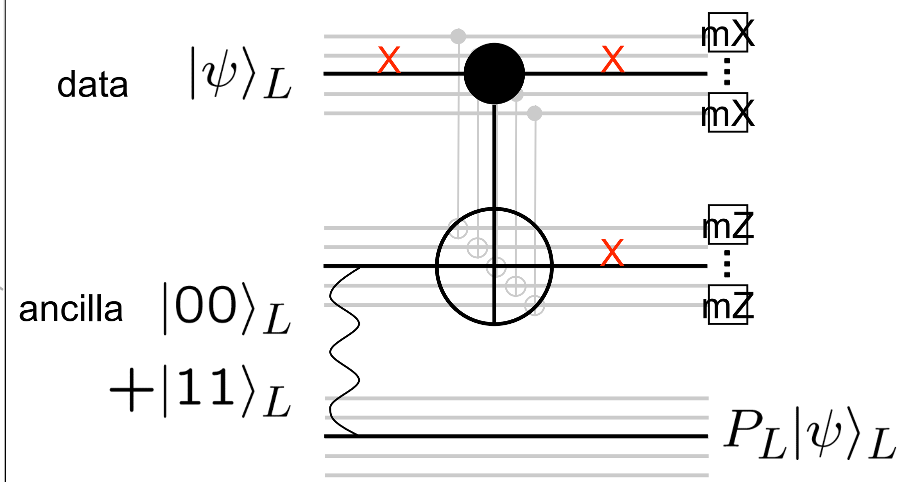


Steane-type error correction

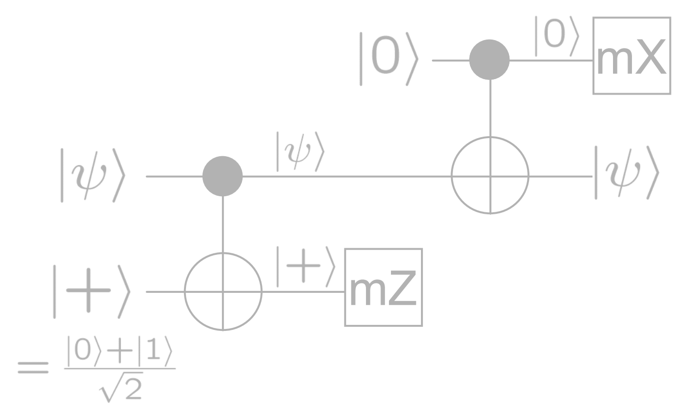
Knill-type error correction

Physical operations

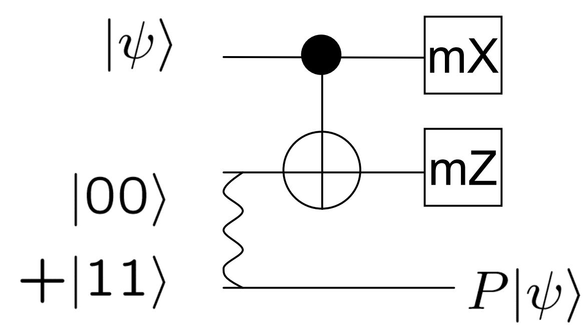




Logical operations

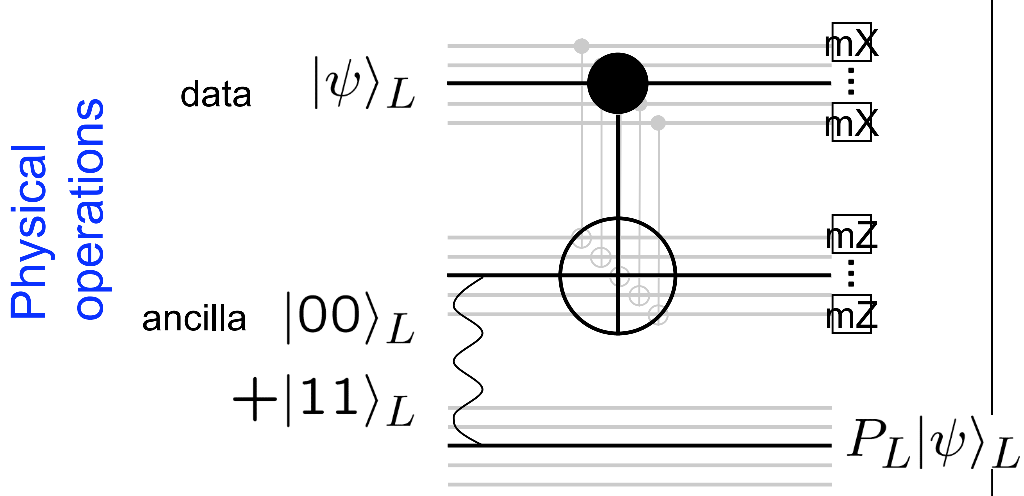


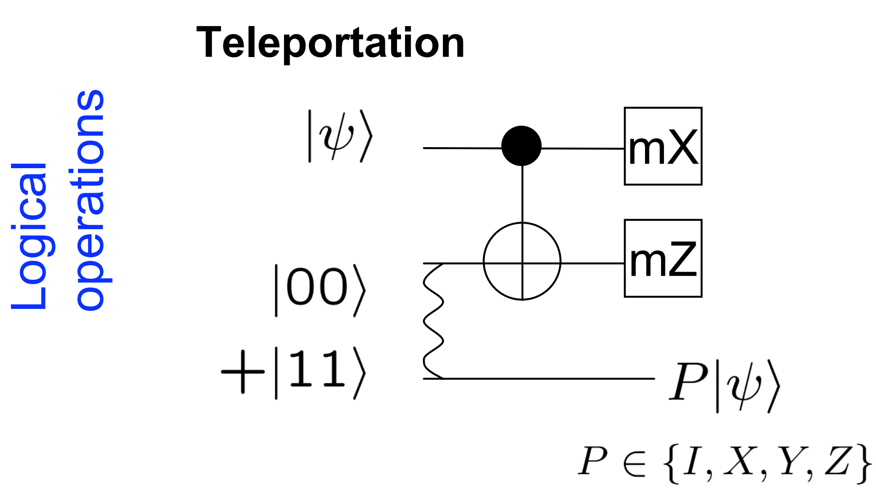
Teleportation



$P \in \{I, X, Y, Z\}$

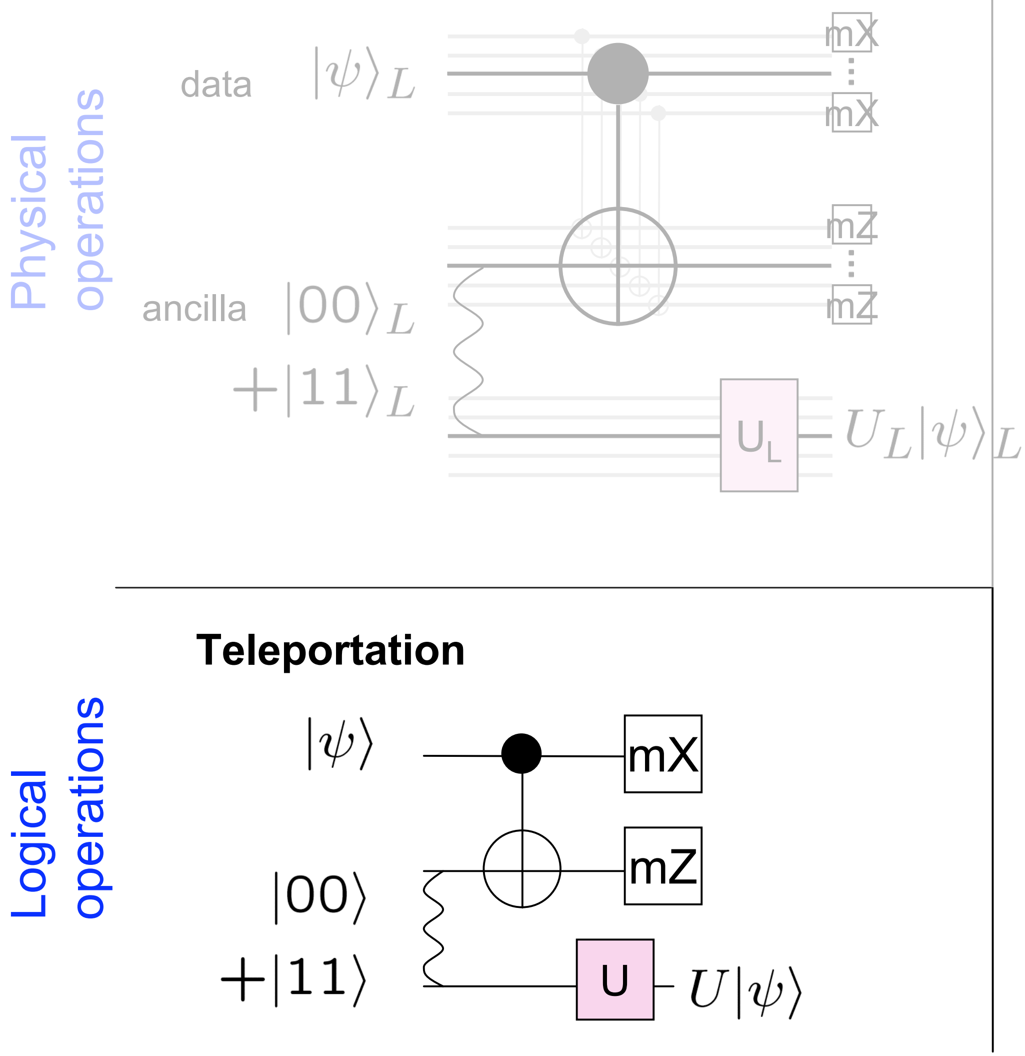
Knill-type error correction





- Advantages
  - Efficient
  - Technical advantage: Reduces blockwise independence to encoded Bell state

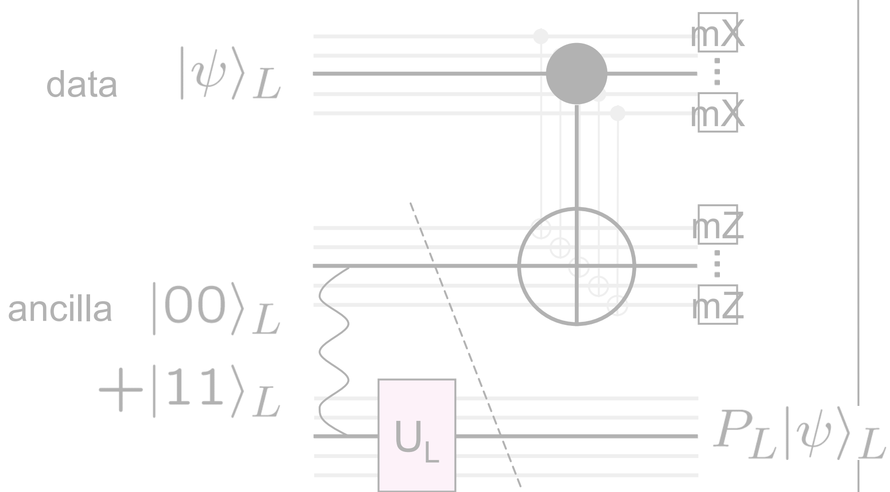
Knill-type correction + computation



- Advantages
  - Efficient
  - Technical advantage: Reduces blockwise independence to encoded Bell state

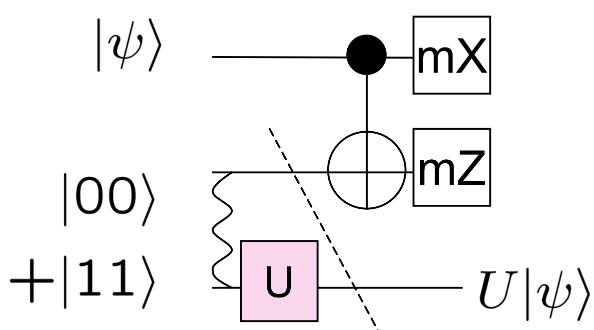
Knill-type correction + computation

Physical operations



Logical operations

Teleportation

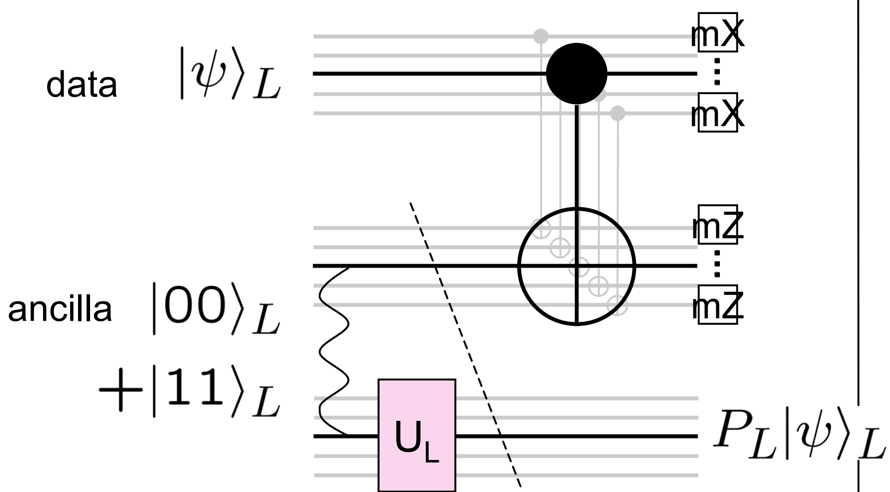


- Advantages
  - Efficient
  - Technical advantage: Reduces blockwise independence to encoded Bell state

Knill-type correction + computation

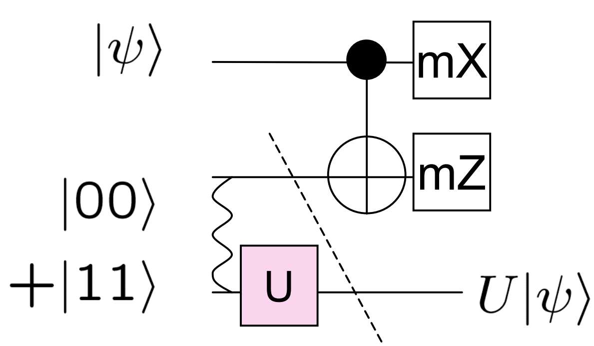
+ Distance-two code
+ Postselection

Physical operations



Logical operations

Teleportation



- Advantages
  - Efficient
  - Technical advantage: Reduces blockwise independence to encoded Bell state
  - Allows for more checking
- Disadvantages
  - High overhead at high error rates with error detection
  - Renormalization penalty requires stronger control over error distribution
  - No threshold has been proved to exist

Main issues

- Bounded dependencies
  - Between different blocks
  - In time
  - Between bit errors and logical errors
- Example:

$|0\rangle_L$ w/ prob. 1-q $|1\rangle_L$ w/ prob. q

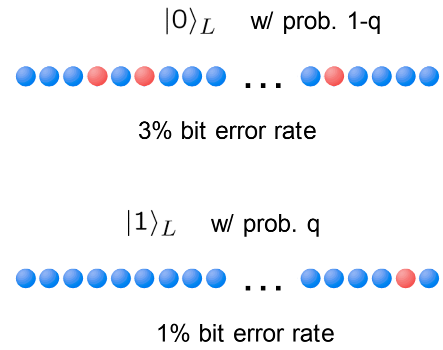
3% bit error rate   1% bit error rate

accepted w/ prob. $(1\text{-}q)\ .97^n$ $q\ .99^n$

$\Rightarrow$ Probability of logical error increases exponentially!

Main issues

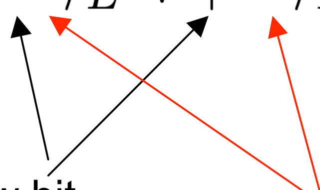
- Bounded dependencies
  - Between bit & logical errors



Monotonicity?

want encoded Bell pair: $|00\rangle_L + |11\rangle_L$

get: $|01\rangle_L + |10\rangle_L$

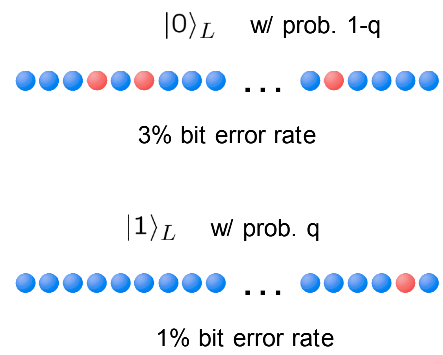


monotonicity $\Rightarrow$ low bit error rate, high bit error rate

But! $|01\rangle_L + |10\rangle_L$

Main issues

- Bounded dependencies
  - Between bit & logical errors



Monotonicity?

(repetition code)

$$|0\rangle_{L(k)}$$

$$|0\rangle_{L(k-1)}\ |0\rangle_{L(k-1)}\ |{\color{red}1}\rangle_{L(k-1)}$$

$$|0\rangle_{L(k-2)}\ |{\color{red}1}\rangle_{L(k-2)}\ |{\color{red}1}\rangle_{L(k-2)}$$

Recent results (continued)

- Magic states distillation [Bravyi & Kitaev '04, Knill '04]
  - Universality method, related to best current threshold upper bounds
  - Reduction from FT universality to FT stabilizer operations
- Optimized fault-tolerance schemes: [Knill '03]
  - Erasure error threshold is 1/2 for Bell measurements
  - [Knill '05]: > ~~5%~~ estimated threshold for depolarizing noise
    1% with substantial but more reasonable overhead
- Improved threshold *proofs*
  - Aliferis/Gottesman/Preskill '05: 2.7×10^{-5} } more efficient
  - R. '05: $< 1.4 \times 10^{-5}$ } distance three
  - Ouyang, R. (unpublished): 10^{-4}

Distance-3 code thresholds

- Basic estimates
  - Aharonov & Ben-Or (1997)
  - Knill-Laflamme-Zurek (1998)
  - Preskill (1998)
  - Gottesman (1997)
- Optimized estimates
  - Zalka (1997)
  - R. (2004)
  - Svore-Cross-Chuang-Aho (2005)
- 2-dimensional locality constraint
  - Szkopek et al (2004)
  - Svore-Terhal-DiVincenzo (2005)

- But no constant threshold was even proven to exist for distance-3 codes!
  - Aharonov & Ben-Or proof only works for codes of distance at least 5
- Today: Threshold for distance-3 codes

Dist-2 code threshold & threshold gap

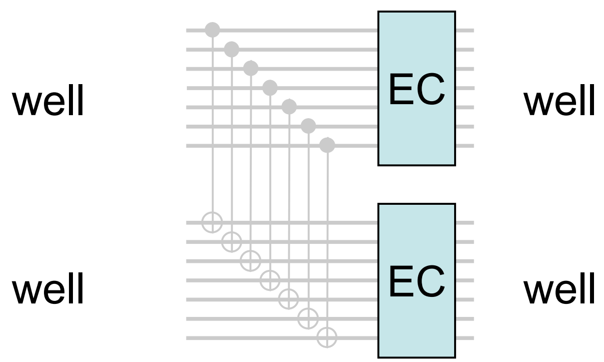
- Knill (2005) has highest threshold estimate ~5%
  - … Albeit with large constant overhead (more reasonable at 1%)
  - Again, no threshold has been proved to exist

- Gaps between proven and estimated thresholds
  - Estimates are as high as ~5%
  - Aliferis-Gottesman-Preskill (2005): 2.6×10^{-5}

- Caveat: Small codes aren't necessarily the most efficient
  - Steane ('03) found 23-qubit Golay code had higher threshold (based on simulations), particularly with slow measurements
  - 23-qubit Golay code proven: 10^{-4}

Distance-three code threshold proof intuition

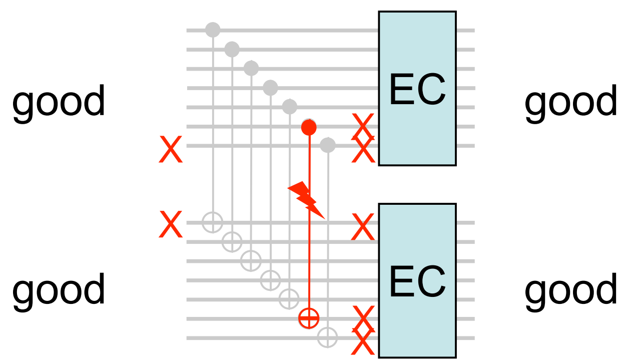
- **Idea:** Maintain inductive invariant of wellness. (A block is well "if it has at most one unwell subblock, and that only rarely.")



What's new: Control *probability distribution* of errors, not just error states.

Aharonov/Ben-Or-style proof intuition

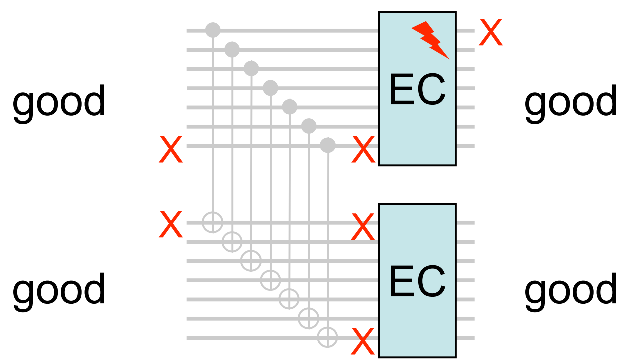
- **Idea:** Maintain inductive invariant of goodness. (A block is good "if it has at most one bad subblock.")



(assuming one level k-1 error, m≥7)

Aharonov/Ben-Or-style proof intuition

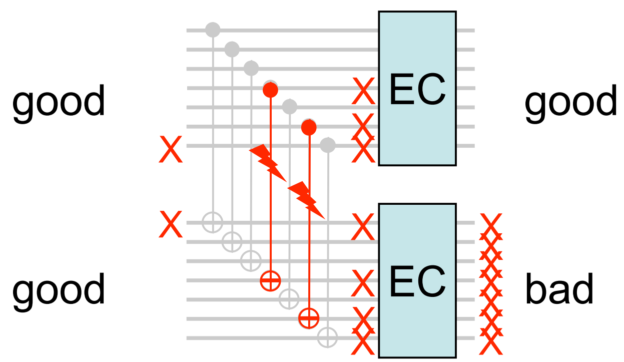
- **Idea:** Maintain inductive invariant of goodness. (A block is good "if it has at most one bad subblock.")



(assuming one level k-1 error, m≥7)

Aharonov/Ben-Or-style proof intuition

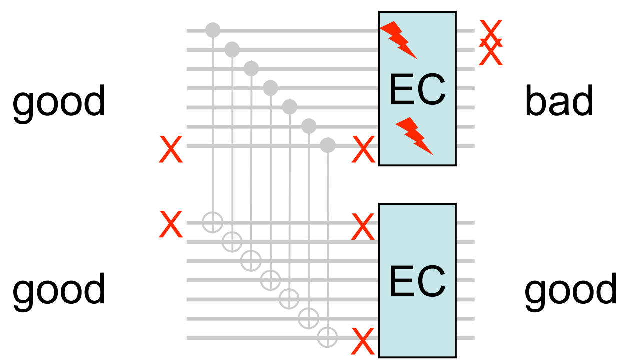
- **Idea:** Maintain inductive invariant of goodness. (A block is good “if it has at most one bad subblock.”)



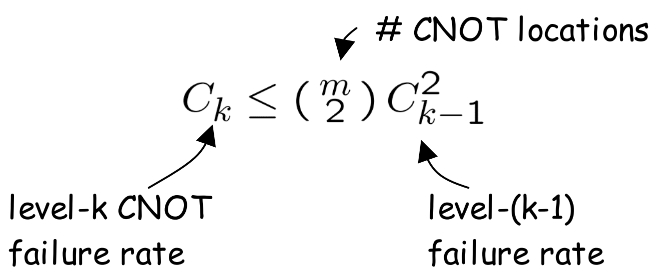
(two level k-1 errors, m=7)

Aharonov/Ben-Or-style proof intuition

- **Idea:** Maintain inductive invariant of goodness. (A block is good "if it has at most one bad subblock.")



(two level k-1 errors)

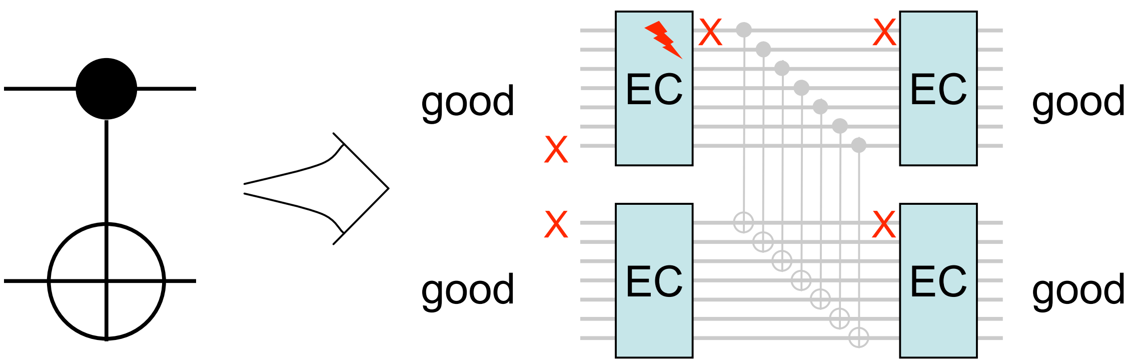


$$C_k \leq \binom{m}{2} C_{k-1}^2$$

Aharonov/Ben-Or-style proof intuition

- **Idea:** Maintain inductive invariant of goodness. (A block is good "if it has at most one bad subblock.")

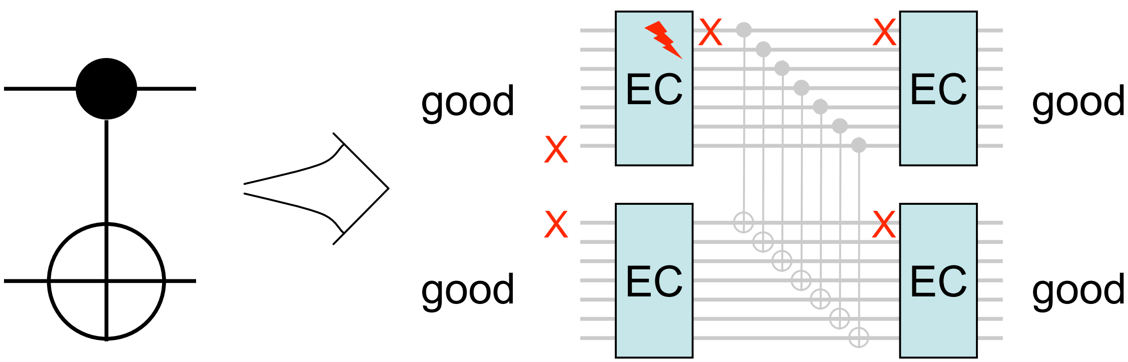
For distance-5 code:



Aharonov/Ben-Or-style proof intuition

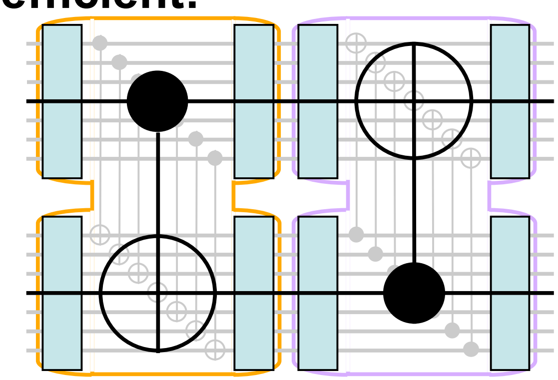
- **Idea:** Maintain inductive invariant of goodness. (A block is good "if it has at most one bad subblock.")

For distance-5 code:



- **Inefficient:**

1.

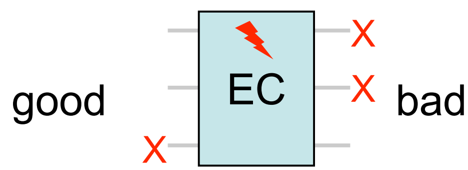


2. $p \to \binom{m}{2} p^2$ not cp^3 (distance = 5)

3. No threshold for concatenated distance-three codes.

Aharonov/Ben-Or-style proof intuition

- **Idea:** Maintain inductive invariant of goodness. (A block is good “if it has at most one bad subblock.”)
- Why not for distance-three codes?



(one level k-1 error is already too many)

- **New idea:** Most blocks should have no bad subblocks. Maintain inductive invariant of a controlled probability distribution of errors: “wellness.” (A block is well “if it only rarely has a bad subblock.”)

Proof overview

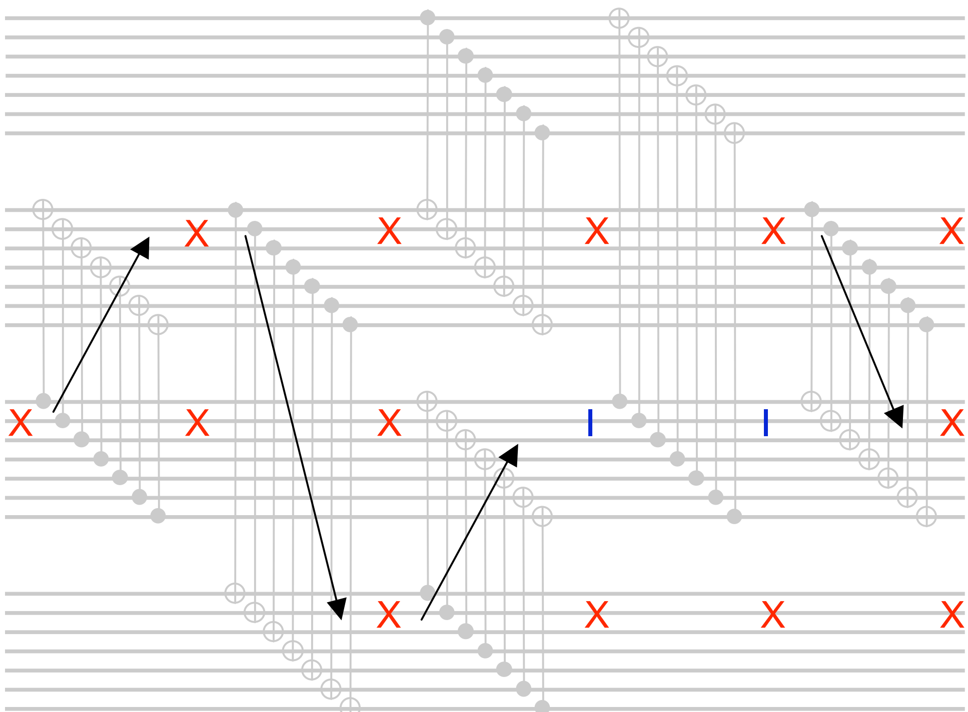
- Def: Error states (resolve $|01\rangle + |10\rangle$ ambiguity)
- Def: Relative error states (encoded CNOT must work even on erroneous input)
- **Def: good block**
- **Def: "well" block**
- Distance-3 code threshold setup
- Def: Logical success and failure
- Distance-3 code threshold proof

Def: Error states

- **Problem:** Different errors are equivalent, so it is ambiguous which bit is in error $|01\rangle + |10\rangle$
- **Solution:** Track errors from their introduction

Def: Error states

- **Tracking errors**



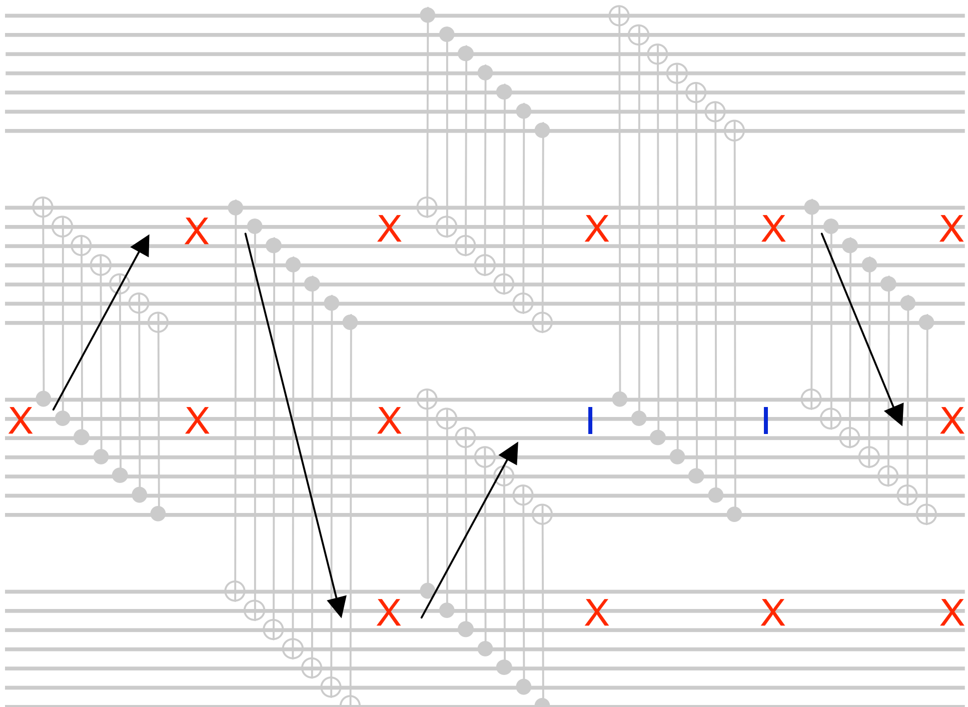
- **Problem:** Different errors are equivalent, so it is ambiguous which bit is in error

$$|01\rangle + |10\rangle$$

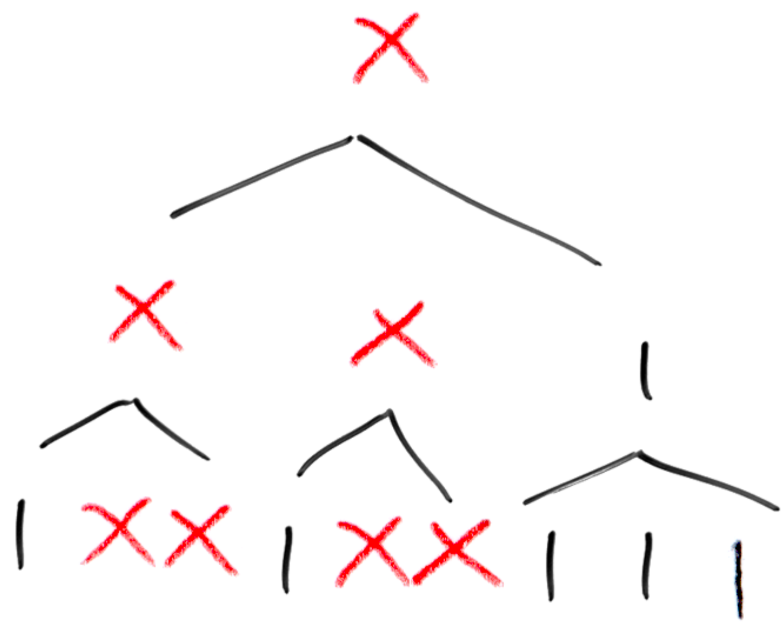
- **Solution:** Track errors from their introduction

Def: Error states

- **Tracking errors**

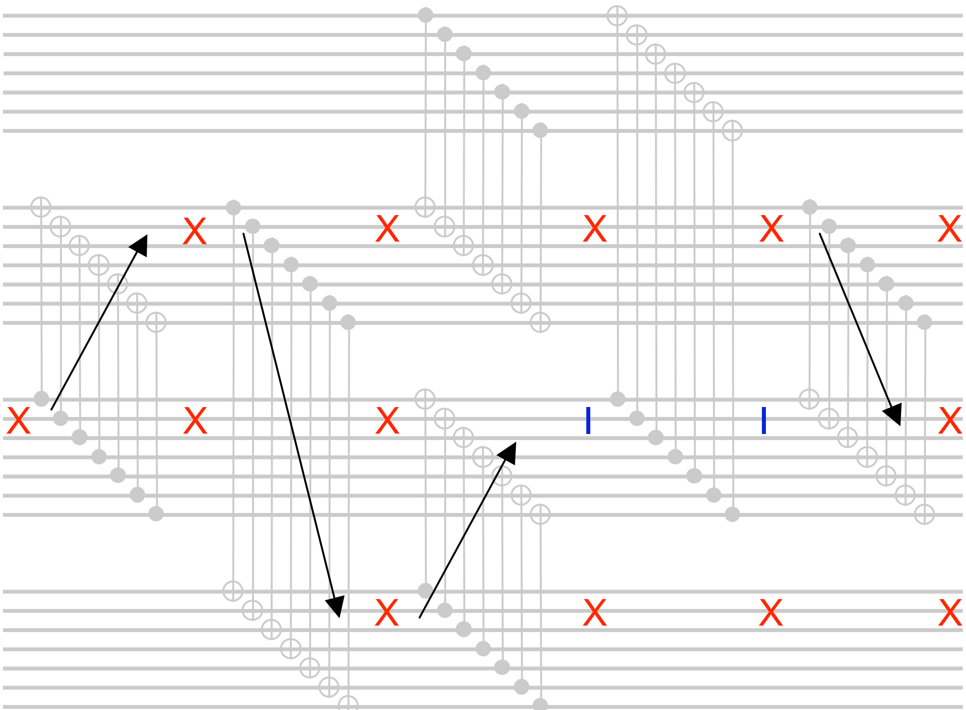


- **Block error states:** ideal recursive decoding

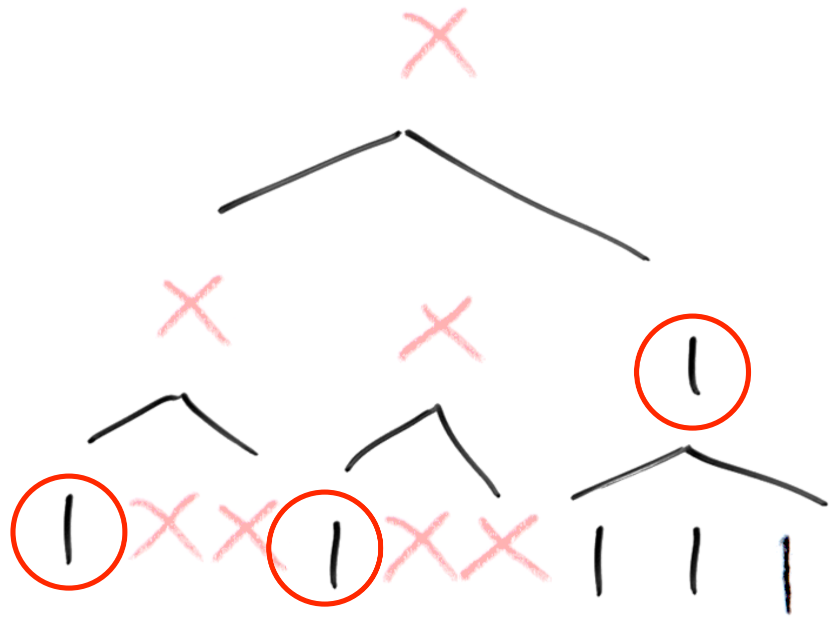


Def: *Relative* Error states

- **Tracking errors**

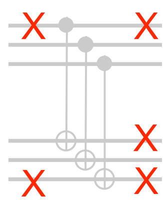


- **Block error states:** ideal recursive decoding
- **Relative error states**

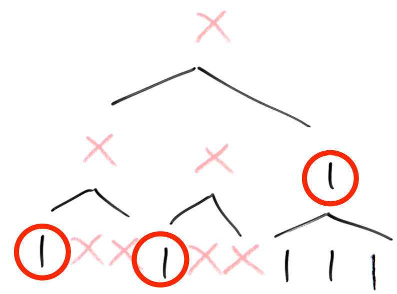


Def: good

- **Tracking errors**



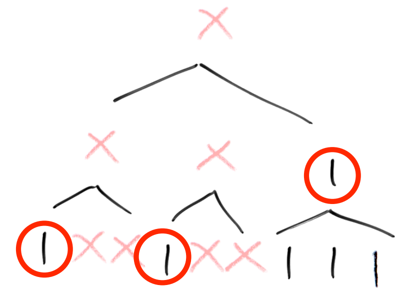
- **Block error states:** ideal recursive decoding
- **Relative error states**



- **Def:** A block_k is good_k if it has at most one subblock_{k-1} either in relative error or not good_{k-1} itself.
(Every bit [$\equiv \text{block}_0$] is good_0.)

good examples

- **Relative error states** based on ideal recursive decoding
- A **good** block has at most one subblock either in relative error or bad.

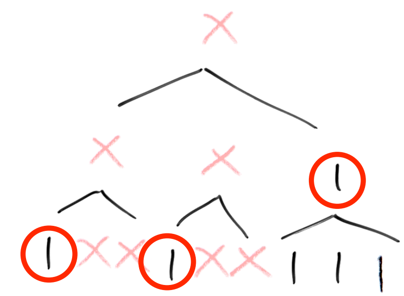


good

bad

good examples

- **Relative error states** based on ideal recursive decoding
- A **good** block has at most one subblock either in relative error or bad.

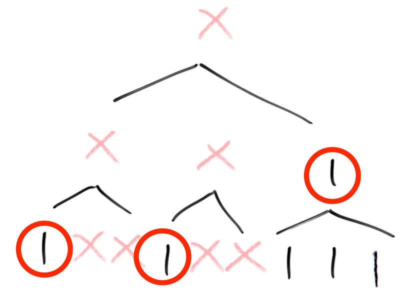


good

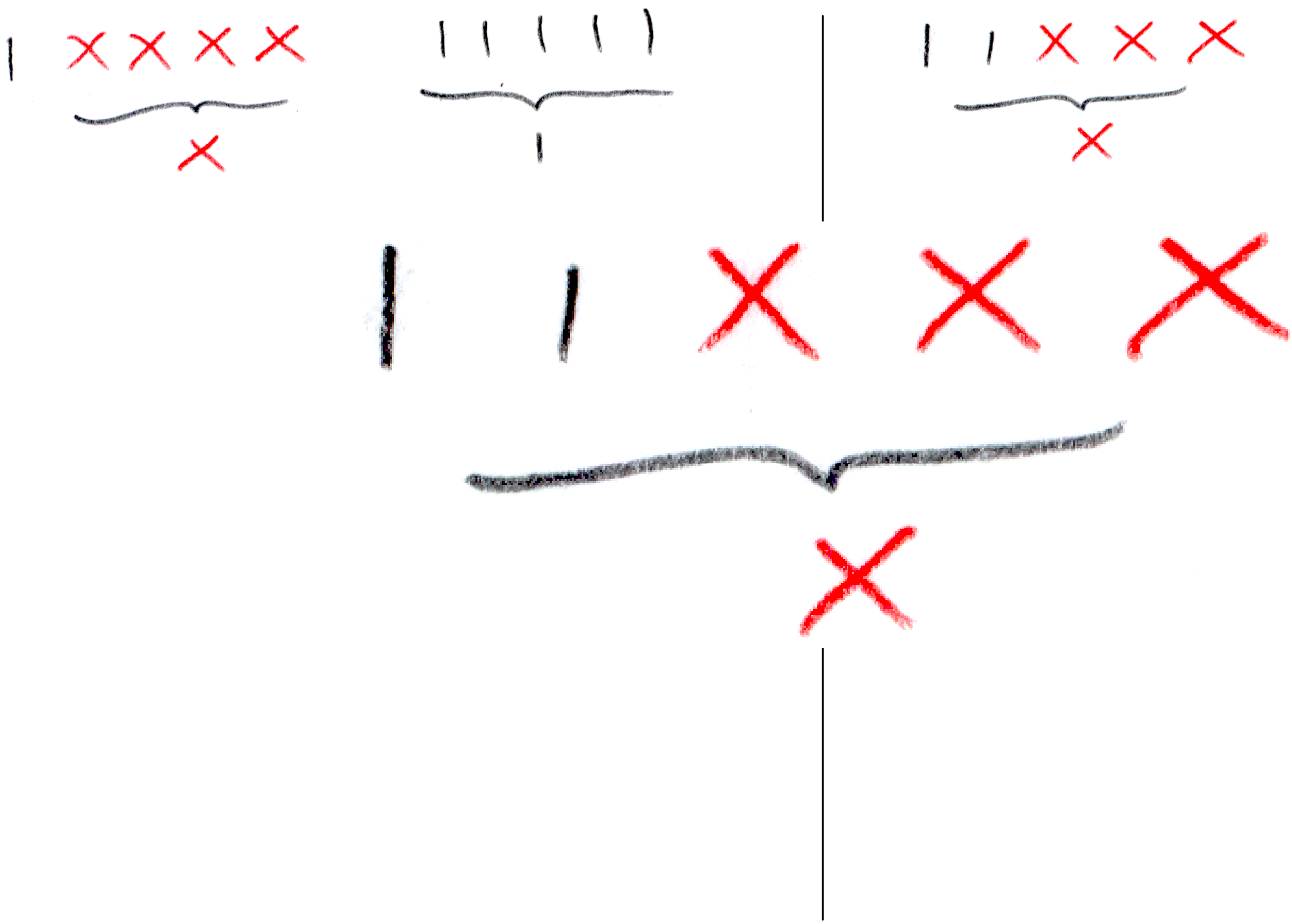
bad

good examples

- **Relative error states** based on ideal recursive decoding
- A **good** block has at most one subblock either in relative error or bad.



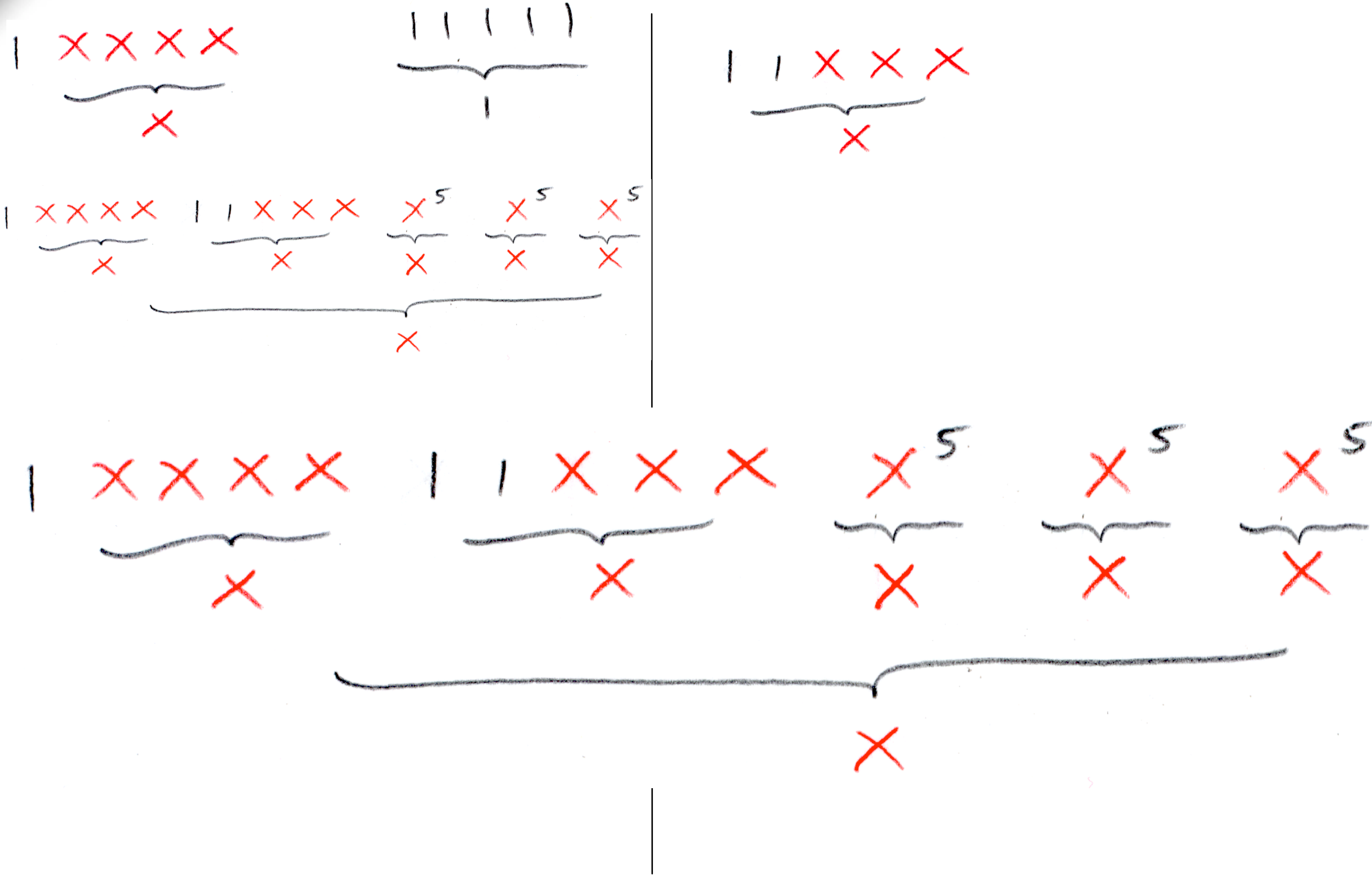
good | bad



good

(at most one subblock either in relative error or bad)

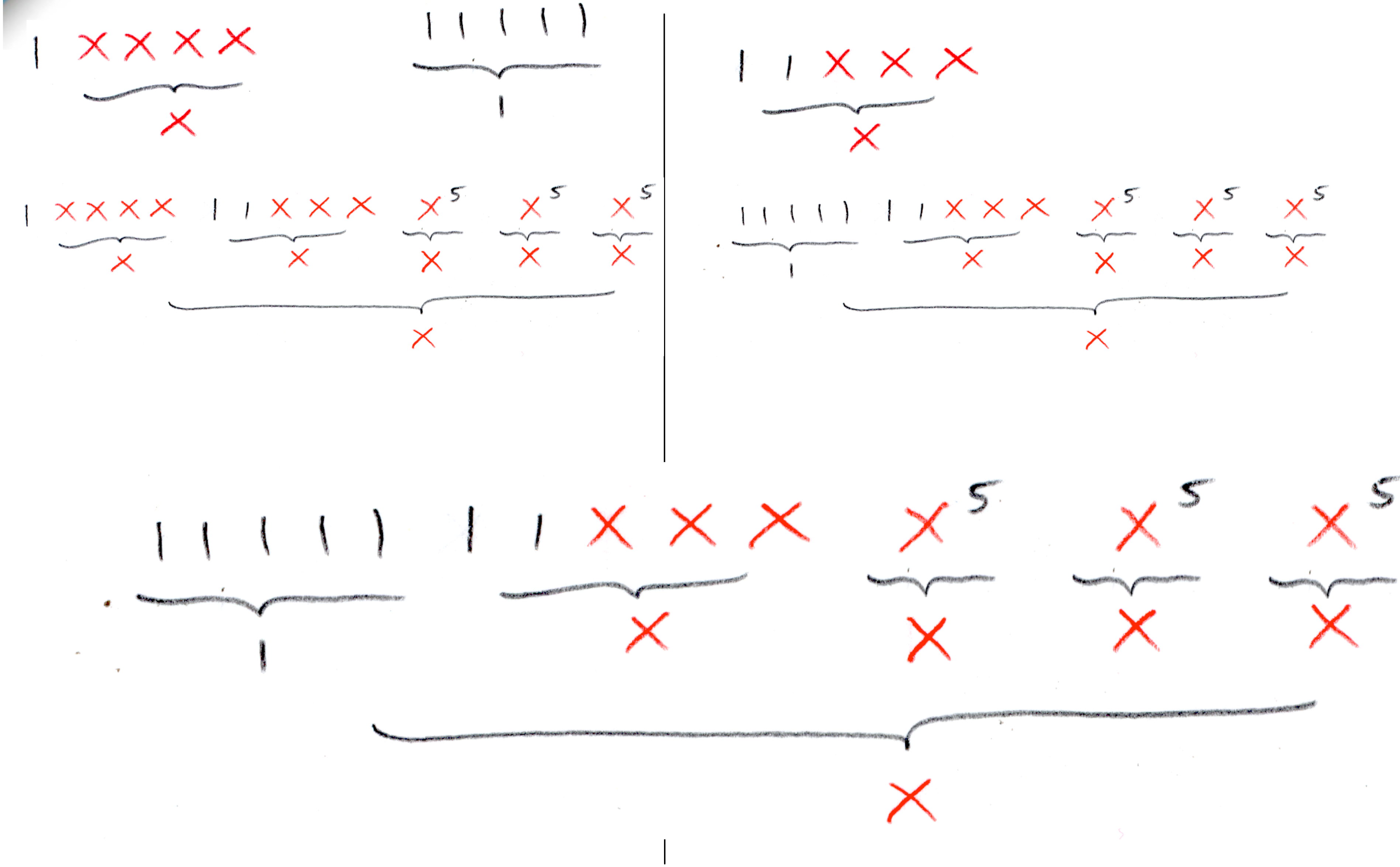
bad



good

(at most one subblock either in relative error or bad)

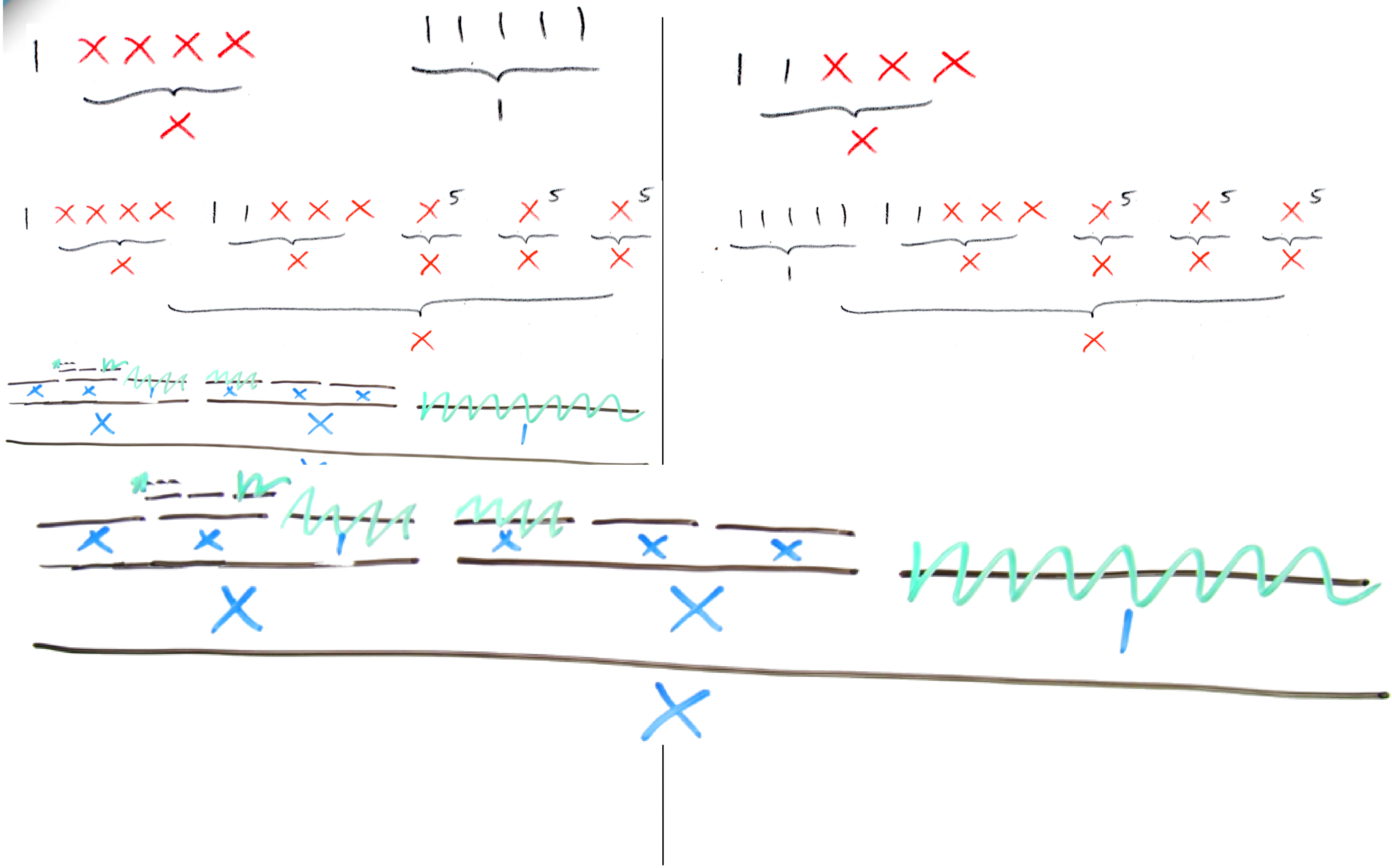
bad



good

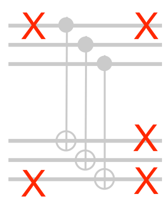
(at most one subblock either in relative error or bad)

bad

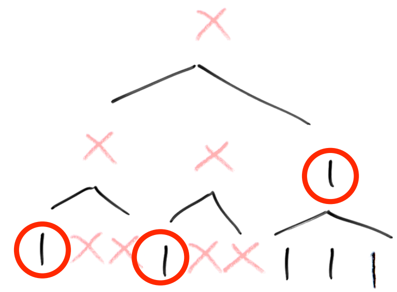


Def: well

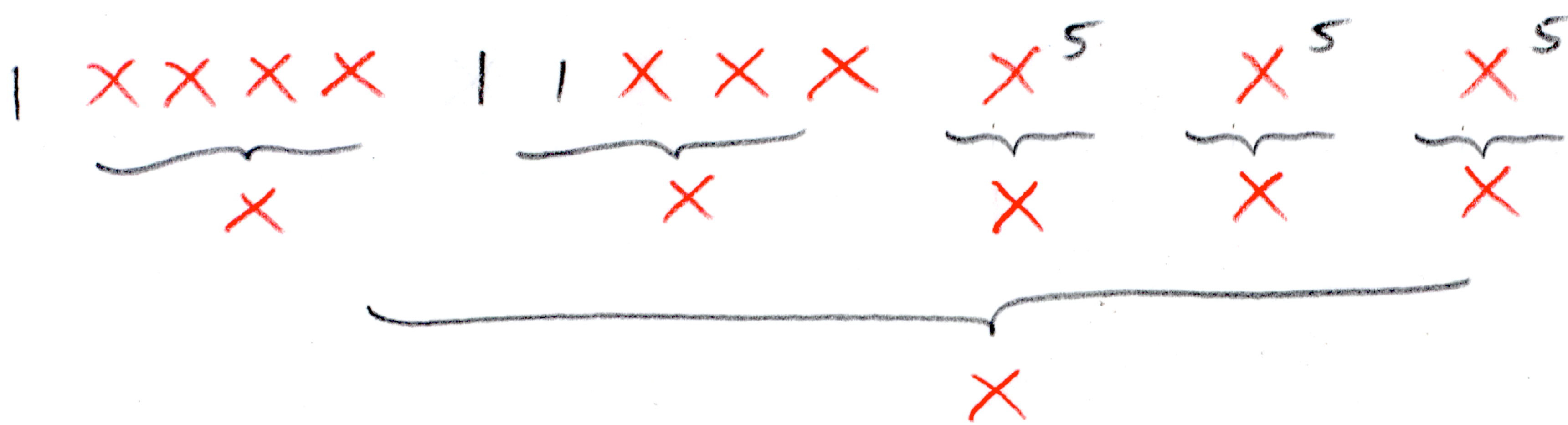
- **Tracking errors**



- **Block error states:** ideal recursive decoding
- **Relative error states**

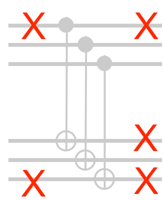


- **Def:** A block_k is good_k if it has at most one subblock_{k-1} either in relative error or not good_{k-1} itself.
(Every bit [$\equiv \text{block}_0$] is good_0.)

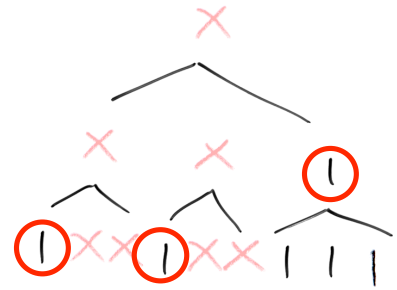


Def: well

- **Tracking errors**



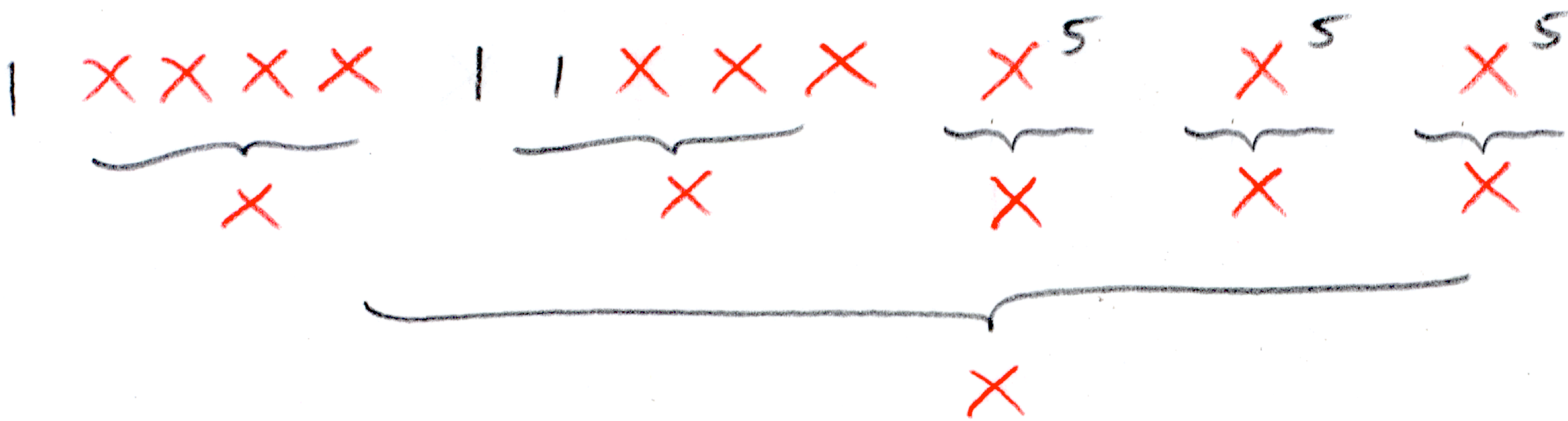
- **Block error states:** ideal recursive decoding
- **Relative error states**



- **Def:** A block_k is $\text{well}_k(p_1,\ldots,p_k)$ if it has at most one subblock_{k-1} either in relative error or not $\text{well}_{k-1}(p_1,\ldots,p_{k-1})$ itself.

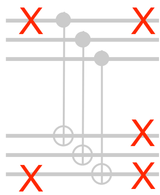
  Additionally, the probability of such a subblock, conditioned on the block's state and the state of all bits in other blocks, is $\leq p_k$.

(Every bit [$\equiv \text{block}_0$] is well_0.)

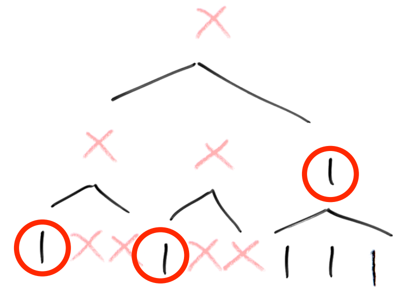


Def: well

- **Tracking errors**

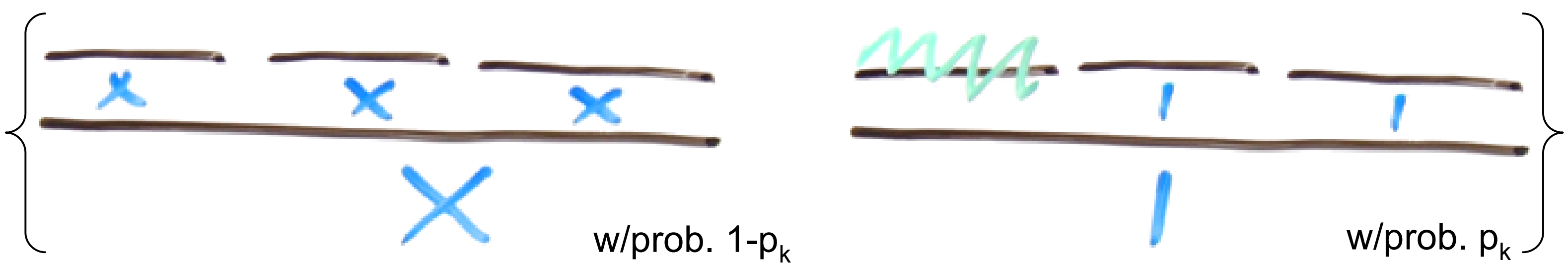


- **Block error states:** ideal recursive decoding
- **Relative error states**



- **Def:** A block_k is $\text{well}_k(p_1,\ldots,p_k)$ if it has at most one subblock_{k-1} either in relative error or not $\text{well}_{k-1}(p_1,\ldots,p_{k-1})$ itself.
  Additionally, the probability of such a subblock, conditioned on the block's state and the state of all bits in other blocks, is $\leq p_k$.
  (Every bit [$\equiv \text{block}_0$] is well_0.)

- **Note:** *Conditioned* on block's state, e.g.,



is *not* 1-well.

Dist-3 code setup

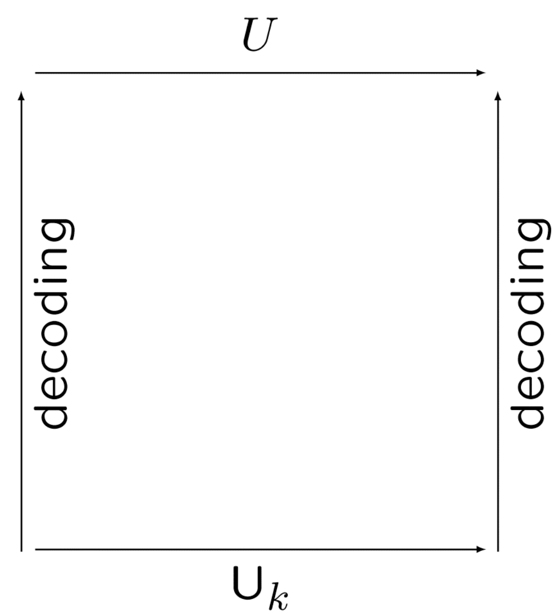
- **Base noise model:** $CNOT_0$ gates fail with X errors independently w/ prob. p

- **Claim C_k** ($CNOT_k$)**:** On success:
  - $Well_k(b_1,\ldots,b_k)$ inputs ⇒ $well_k(b_1,\ldots,b_k)$ outputs, and logical CNOT
  - Arbitrary inputs ⇒ $well_k(b_1,\ldots,b_k)$ outputs, and possibly incorrect logical effect

  Failure prob. ≤ C_k (C_0 = p).

Def: Logical failure

- **Def:** Logical operation U_k on one or more blocks$_k$ has the correct logical effect if the diagram commutes:

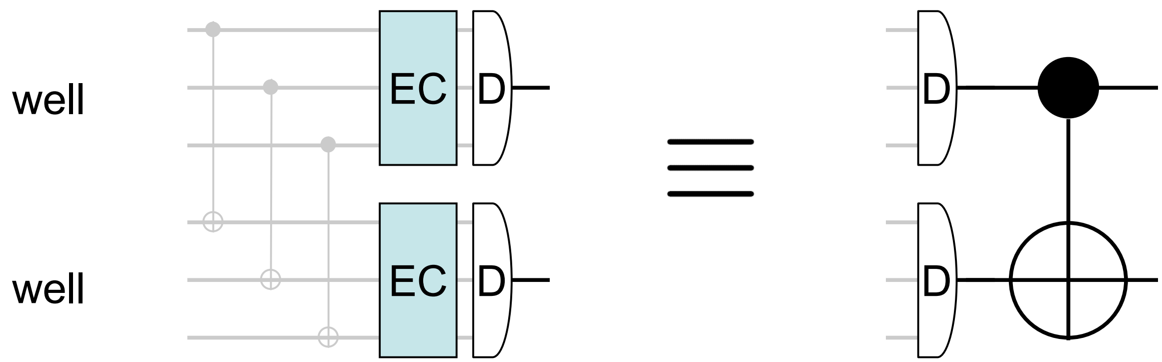


- U_k has a possibly incorrect logical effect if the same diagram commutes but with $P \circ U$ on the top arrow, where P is a Pauli operator or Pauli product on the involved blocks.

Dist-3 code setup

- **Claim C_k** (CNOT$_k$)**:** On success:
  - Well inputs ⇒ well outputs, and logical CNOT
  - Arbitrary inputs ⇒ well outputs

Failure prob. ≤ C_k (C_0 = p).



- **Claim B_k** (Correction$_k$)**:** On success:
  - Well$_k$(b_1,…,b_k) input ⇒ well$_k$(b_1,…,b_k) output, and no logical effect
  - Arbitrary input ⇒ well$_k$(b_1,…,b_k) output

Failure prob. ≤ B_k (B_0 = 0).

Additionally, if all but one of the input subblocks$_{k-1}$ are well$_{k-1}$(b_1,…,b_{k-1}), then with probability at least 1-B_k' there is no logical effect and the output is well$_k$(b_1,…,b_k).

Dist-3 code threshold proof

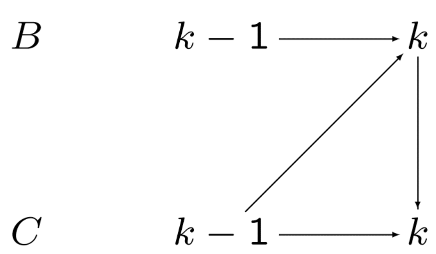
- **Two operations:**

  B. Error correction
  C. (Logical) CNOT gate

- **Two indexed claims:**

  B_k Error correction$_k$ success except w/ prob. $\leq B_k$
  C_k CNOT$_k$ success except w/ prob. $\leq C_k$

- **Proofs by induction:** Implications:



$$B_k = O\left((B_{k-1} + C_{k-1})^2\right)$$

$$C_k = O\left(B_k + C_{k-1}^2\right)$$

Dist-3 code threshold proof

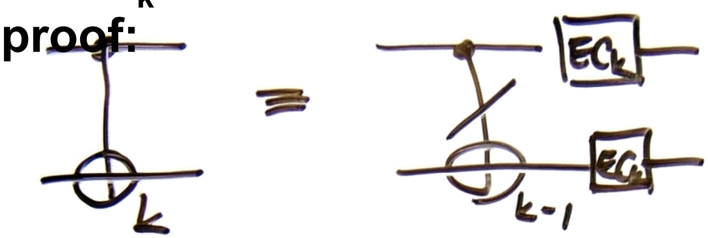
- **Claim B_k** (Correction$_k$)**:** On success:
  - Well$_k$(b_1,…,b_k) input ⇒ well$_k$(b_1,…,b_k) output, no logical effect
  - Arbitrary input ⇒ well$_k$(b_1,…,b_k) output

  Failure prob. ≤ B_k (B_0 = 0).

  Additionally, if all but one of input subblocks$_{k-1}$ are well$_{k-1}$(b_1,…,b_{k-1}), then w/ prob. ≥ 1-B_k', output is well$_k$(b_1,…,b_k) and no logical effect.
- **Claim C_k** (CNOT$_k$)**:** On success:
  - Well inputs ⇒ well outputs, and logical CNOT
  - Arbitrary inputs ⇒ well outputs

  Failure prob. ≤ C_k (C_0 = p).

CNOT$_k$ proof:



- Assume input blocks are well$_k$(b_1,…,b_k). Declare failure if either Correction$_k$ fails, or if there are two level k-1 failures.

$$C_k \equiv \Big(2B_k + (nC_{k-1})(2B'_k) + \binom{n}{2} C_{k-1}^2\Big) + 2b_k(2B'_k + nC_{k-1}) + b_k^2$$

- On success, transverse CNOTs$_{k-1}$ implement the correct logical effect (but possibly correlate errors). The successful Corrections$_k$ have no logical effect but restore wellness (bounded dependencies).

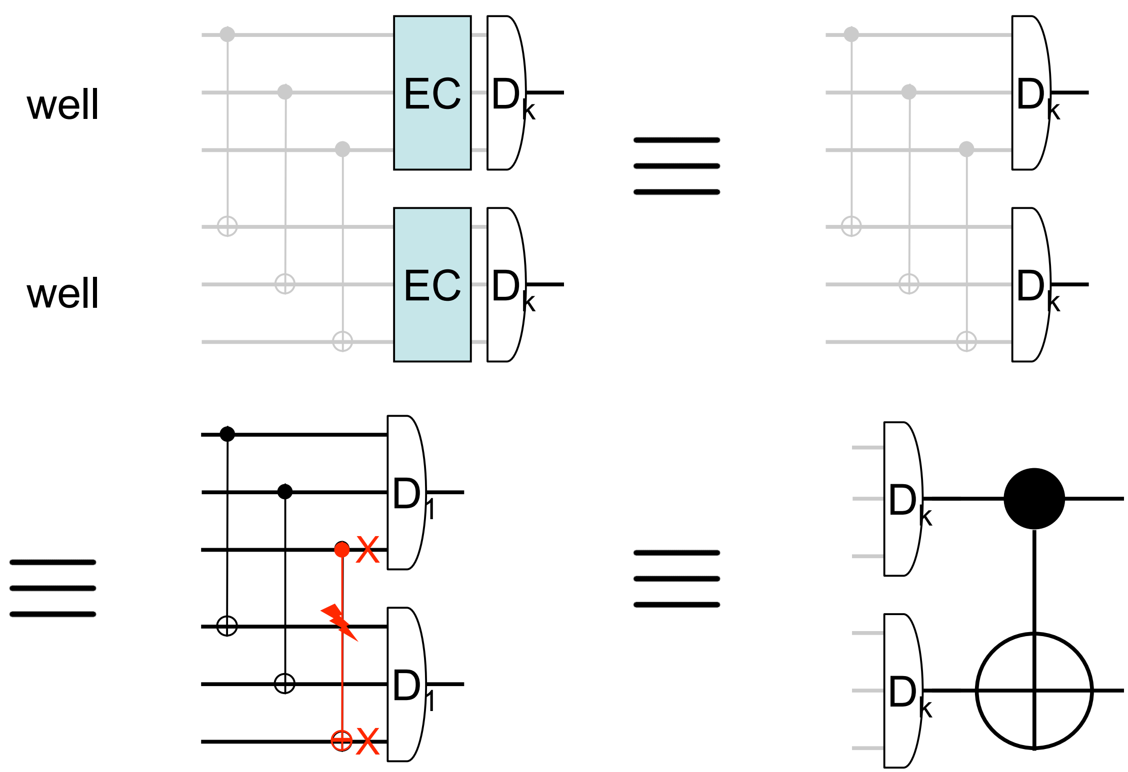
Dist-3 code threshold proof

- **Claim C_k** (CNOT$_k$)**:** On success:
  - Well inputs ⇒ well outputs, and logical CNOT
  - Arbitrary inputs ⇒ well outputs

  Failure prob. ≤ C_k (C_0 = p).

CNOT$_k$ proof: Failure if either Correction$_k$ fails, or if there are two level k-1 failures.

Success: transverse CNOTs$_{k-1}$ implement correct logical effect. Corrections$_k$ have no logical effect.

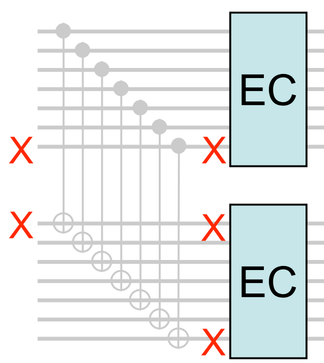


□

Aliferis-Gottesman-Preskill threshold intuition

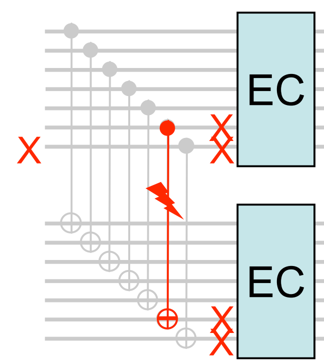
- **Aharonov & Ben-Or Idea:** Maintain inductive invariant of (1-)goodness. (A block is good “if it has at most one bad subblock.”)
- Two ways it can fail with distance-three codes:

1.



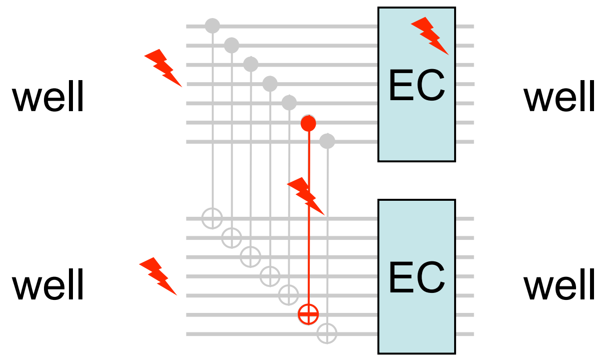
Both input blocks have a bad subblock.

2.



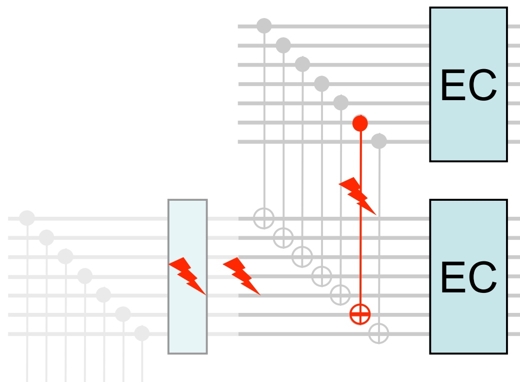
One input block has a bad subblock, and an additional error occurs.

Aliferis-Gottesman-Preskill threshold intuition



- A/B: Maintain ‘good’ness — two faults in rectangle cause logical failure (d≥5)
- R: Maintain ‘well’ness — two faults in rectangle or well input cause logical failure

Aliferis-Gottesman-Preskill threshold intuition



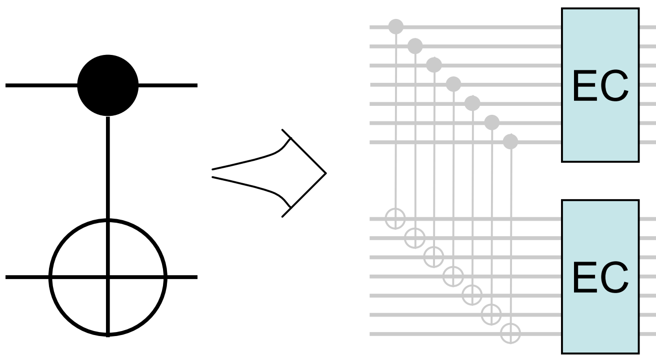
- A/B: Maintain 'good'ness — two faults in rectangle cause logical failure (d≥5)

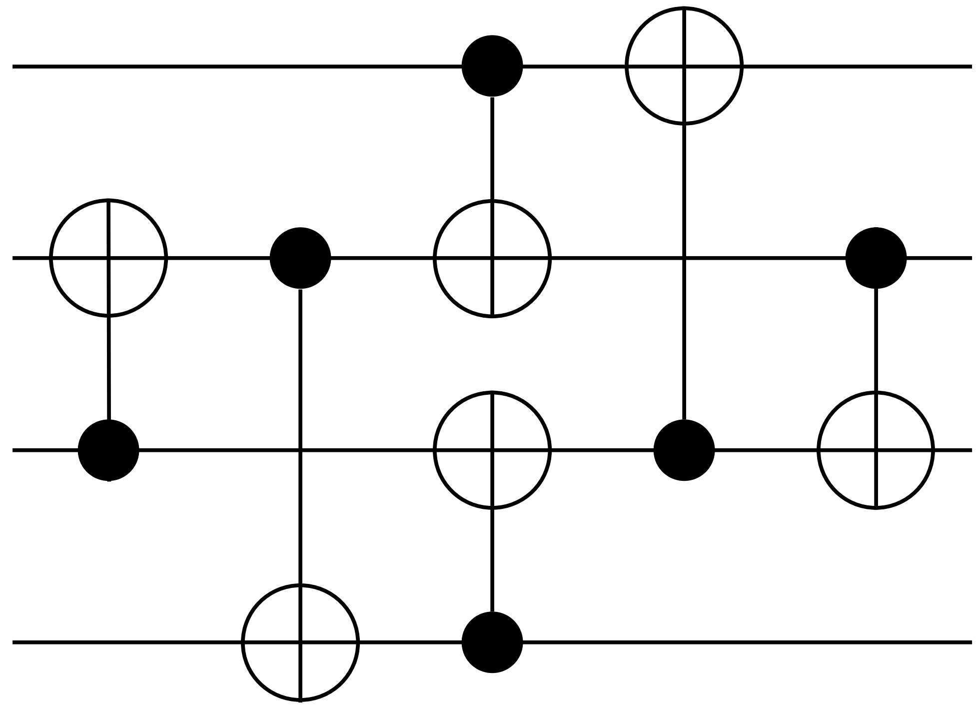
- R: Maintain 'well'ness — two faults in rectangle or well input cause logical failure

…errors in input come from errors in
the preceding error correction…

- A/G/P: two faults in *extended* (overlapping) rectangle cause logical failure

Aliferis-Gottesman-Preskill threshold intuition



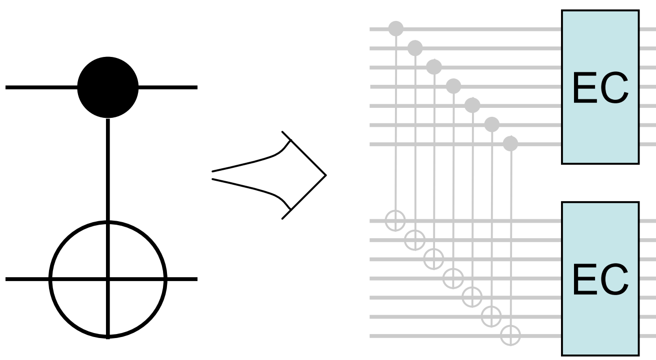


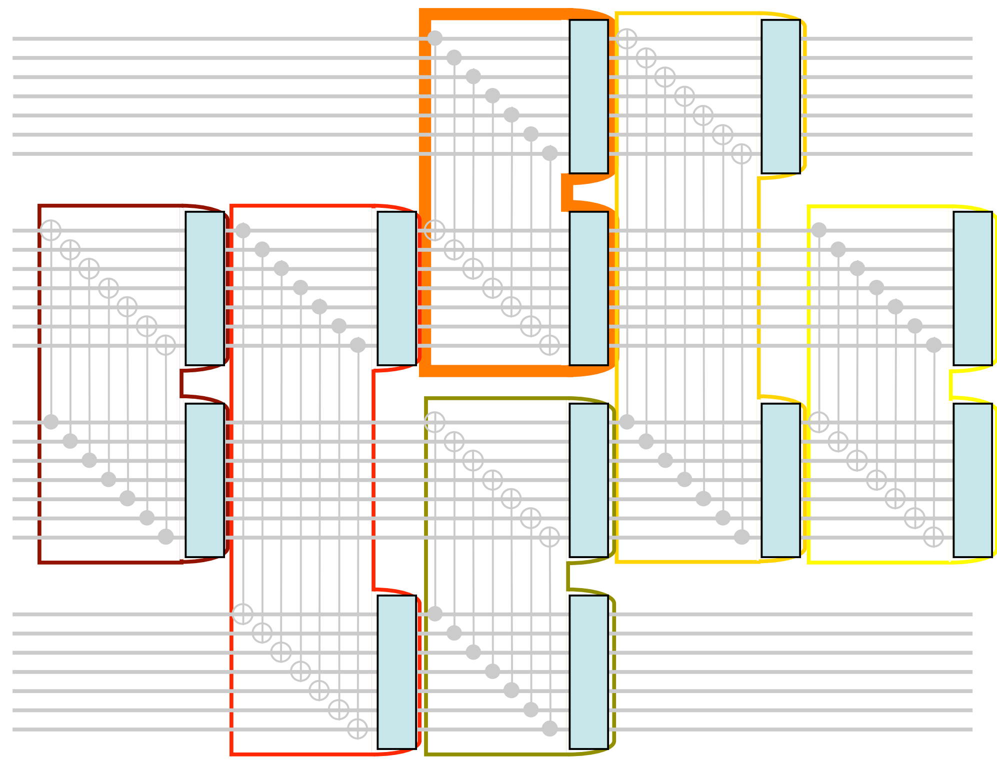
- A/B: Maintain 'good'ness — two faults in rectangle cause logical failure (d≥5)
- R: Maintain 'well'ness — two faults in rectangle or well input cause logical failure

…errors in input come from errors in the preceding error correction…

- A/G/P: two faults in *extended* (overlapping) rectangle cause logical failure

Aliferis-Gottesman-Preskill threshold intuition



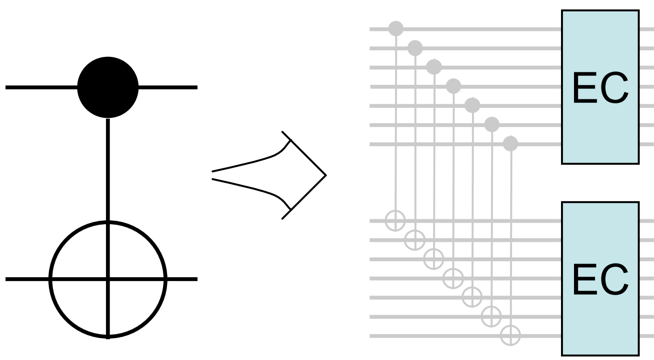


- A/B: Maintain ‘good’ness — two faults in rectangle cause logical failure (d≥5)
- R: Maintain ‘well’ness — two faults in rectangle or well input cause logical failure

…errors in input come from errors in the preceding error correction…

- A/G/P: two faults in *extended* (overlapping) rectangle cause logical failure

Aliferis-Gottesman-Preskill threshold intuition



- A/B: Maintain 'good'ness — two faults in rectangle cause logical failure (d≥5)
- R: Maintain 'well'ness — two faults in rectangle or well input cause logical failure

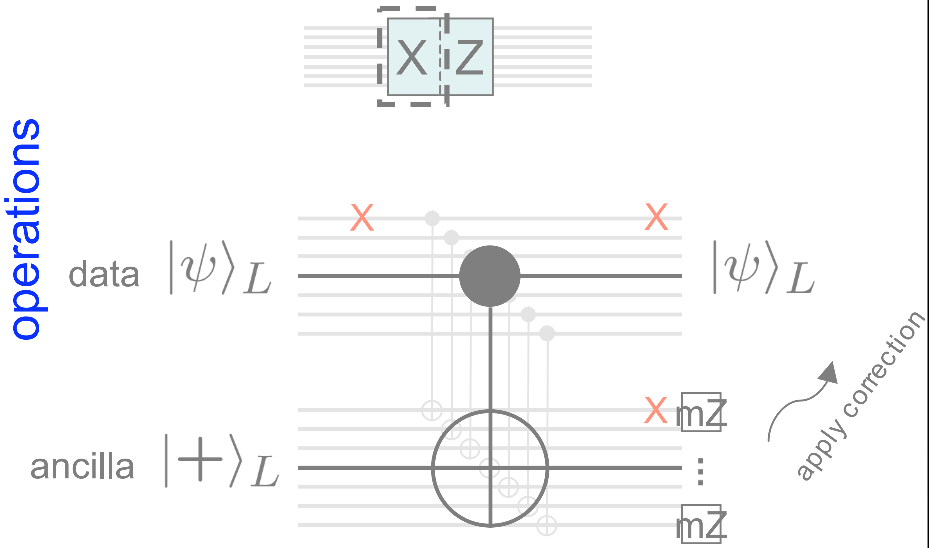
…errors in input come from errors in the preceding error correction…

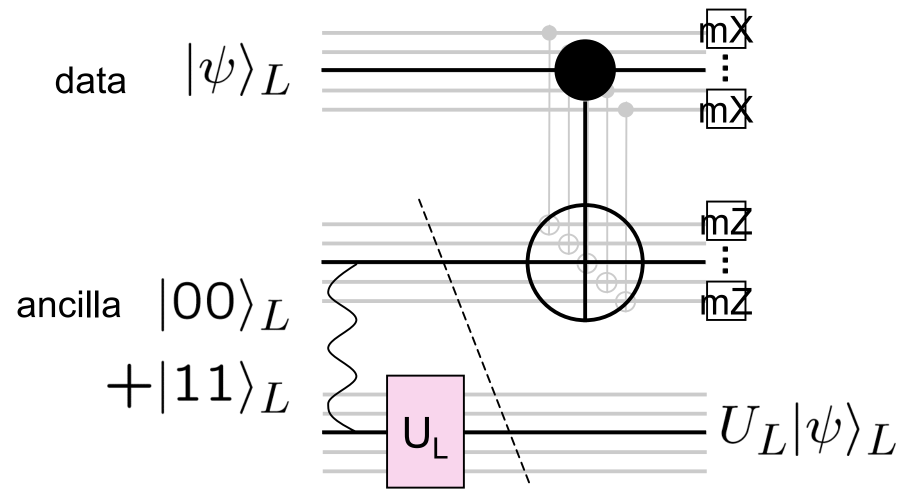
- A/G/P: two faults in *extended* (overlapping) rectangle cause logical failure

Steane-type error correction

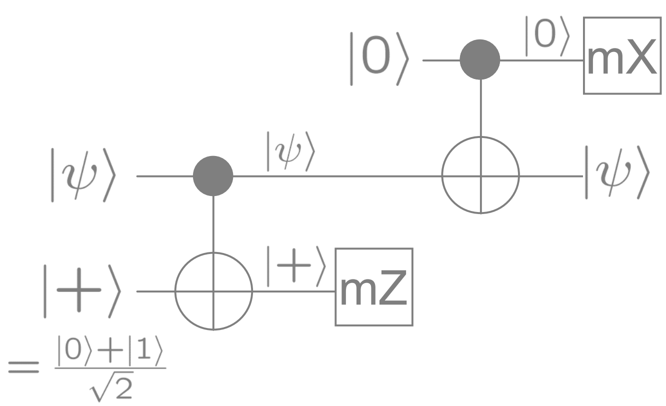
Knill-type correction + computation

Physical operations

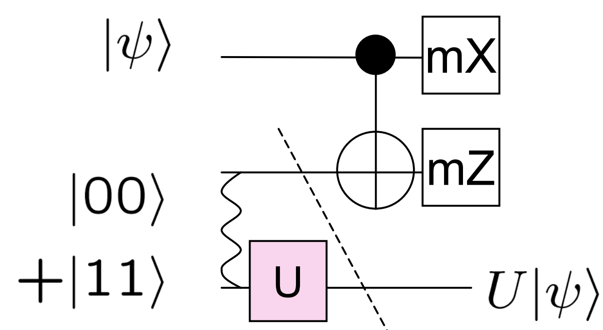




Logical operations

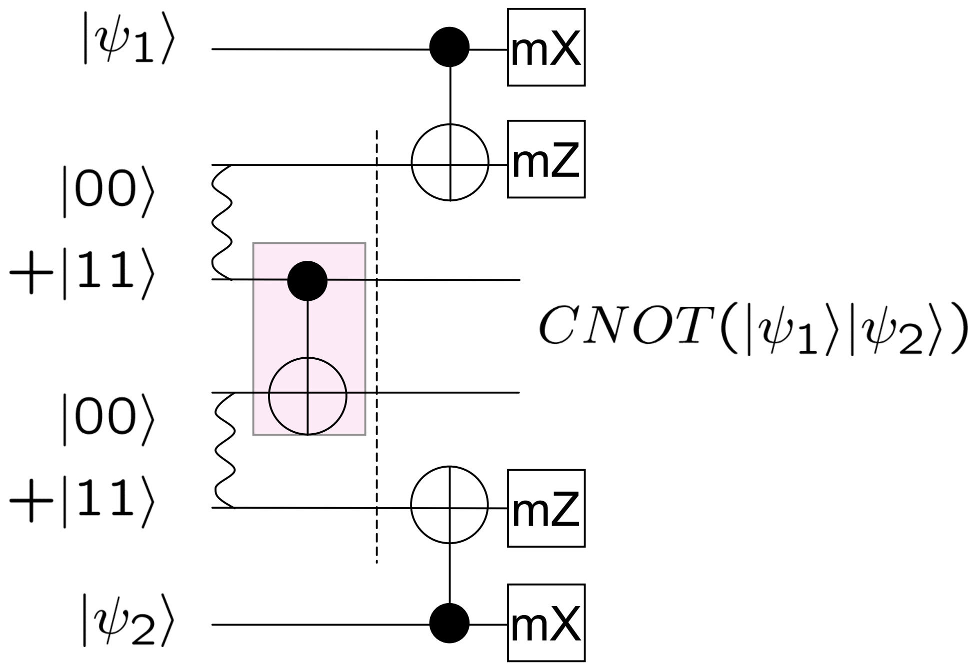


Teleportation

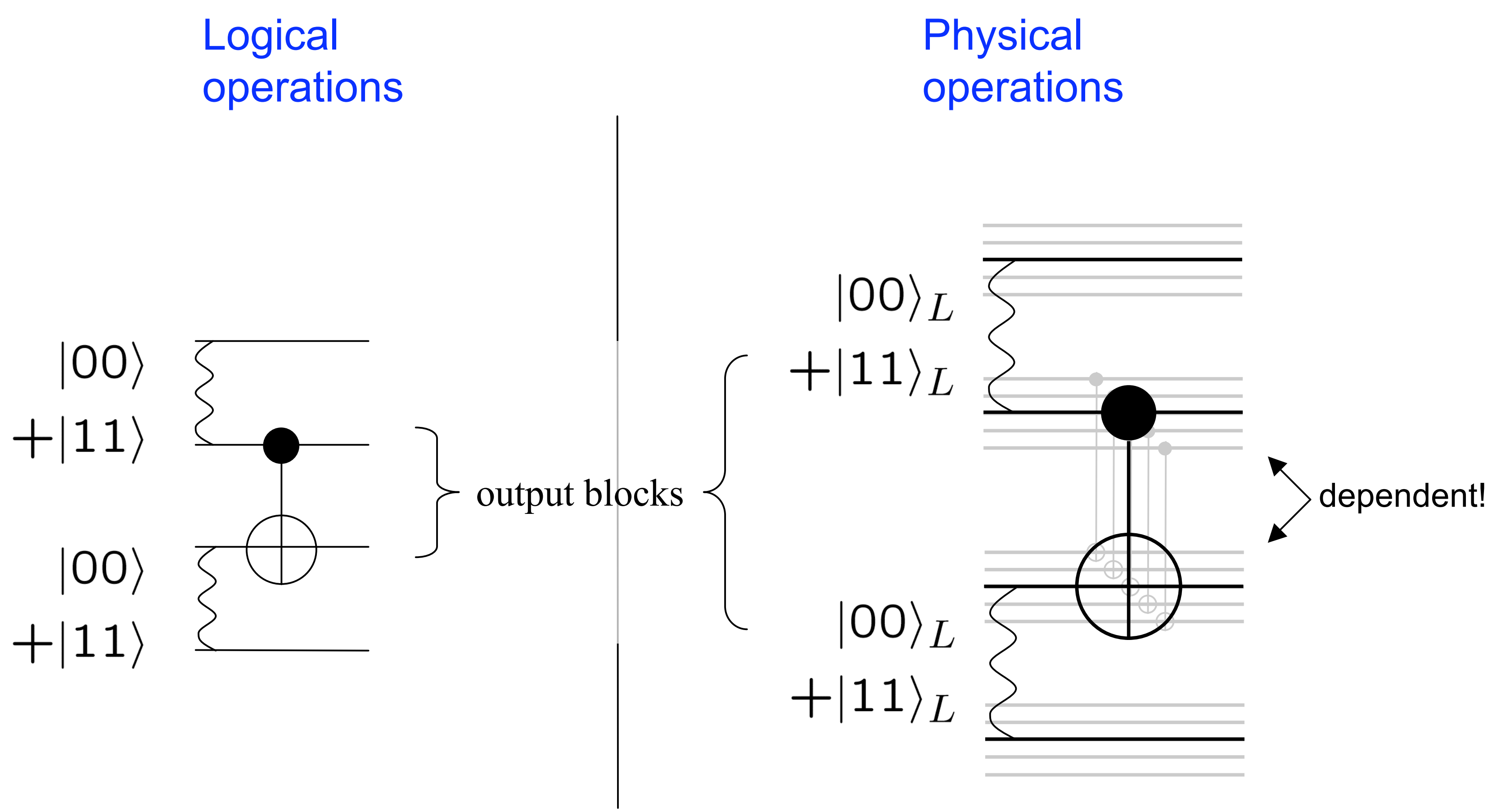


Teleporting a CNOT gate

Logical operations

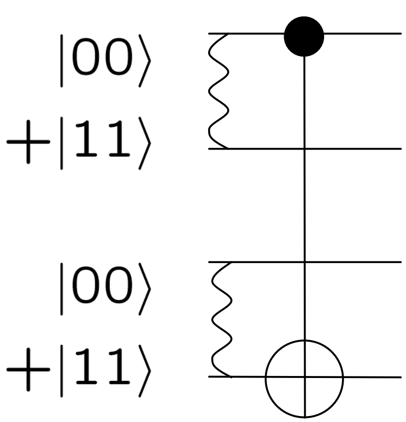


Teleporting a CNOT gate

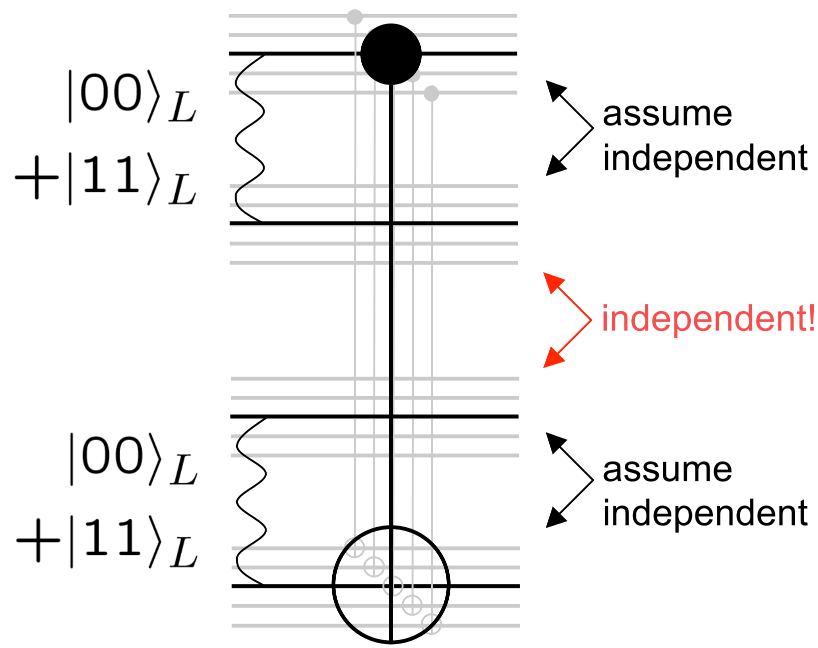


Teleporting a CNOT gate

Logical operations



Physical operations



$\Rightarrow$ Achieving independent errors on CNOT output blocks reduces to preparing encoded Bell states with block-independent errors

Unfortunately, this is impossible… But:

Summary

- New threshold proof
  - Based on bounding the *distribution* of errors in the system at each time step
  - More efficient than classical threshold proofs, leads to higher rigorous noise threshold lower bounds
  - Works for concatenated distance-three codes
- Possible extensions
  - Improved analysis of optimized standard fault-tolerance schemes

    (Ouyang, R.: 10^{-4})
  - Extend proof to work with schemes using distance-two codes and extensive postselection. Major difficulty is obtaining better control over error distribution, particularly of dependencies and of errors in the *bad* blocks.

Blank slide