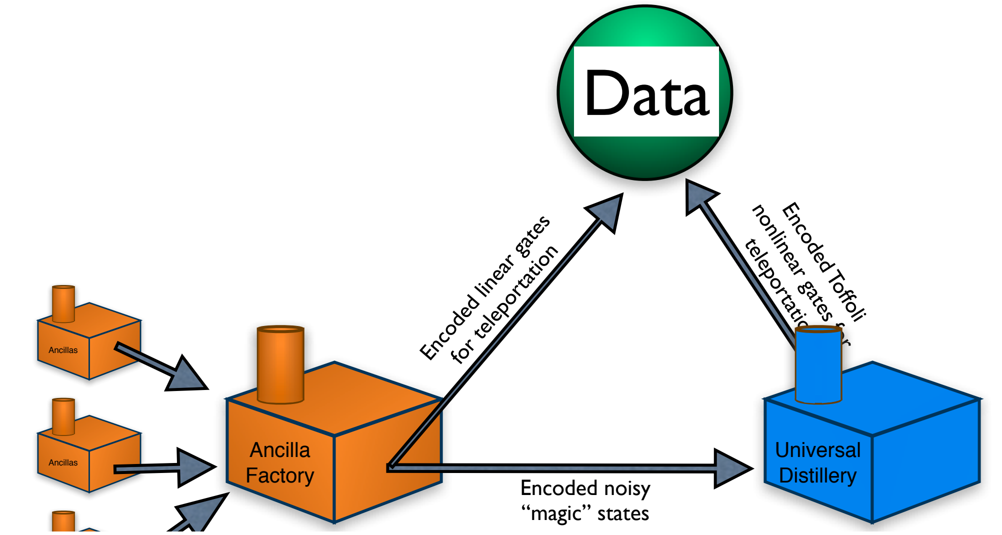
Making quantum computers fault tolerant



Ancilla factories

Universal distillery

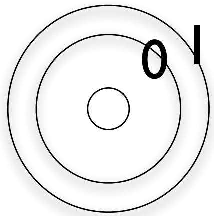
Ben Reichardt
Caltech

Motivations for quantum information processing

- Quantum computing (QC)
  - Extended Church-Turing Thesis: Anything physically efficiently computable can be computed efficiently on my laptop
  - QC: Extended Church-Turing Thesis is false; there are exponentially-faster algorithms (for interesting problems) by using quantum mechanics
- Cryptography
  - Breaks RSA public-key cryptosystem
  - Gives unconditionally secure key distribution
- Simulation & modeling
  - for quantum devices,
  - chemistry,
  - materials (high-T superconductors, new states of matter?)
- Quantum sensing
  - Precise measurement and lithography
  - Atomic clocks
- Basic science
  - Investigate measurement/decoherence, quantum/classical boundary
  - Test qu. mechanics on new scales

(but no free lunch…)

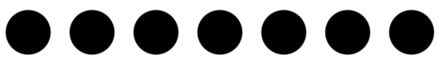
Quantum information

- "Qubit": 

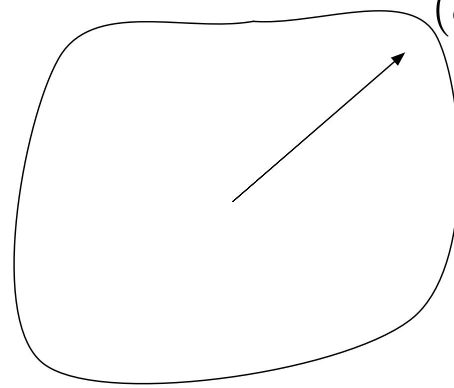
$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \alpha_0|0\rangle + \alpha_1|1\rangle$$

$$|\alpha_0|^2 + |\alpha_1|^2 = 1$$

- State of n qubits = unit vector in $\mathbf{C}^{2^n}$

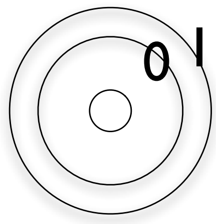


$(\alpha_x)_{x\in\{0,1\}^n}$



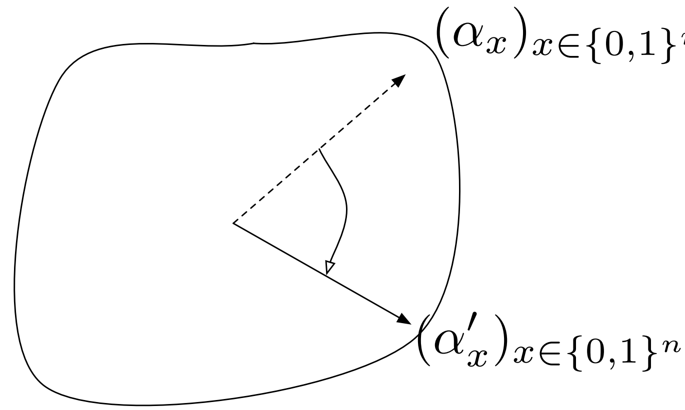
Quantum information

- "Qubit":

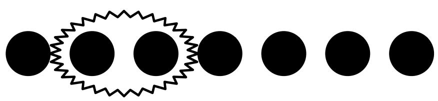


$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \alpha_0|0\rangle + \alpha_1|1\rangle$$

$$|\alpha_0|^2 + |\alpha_1|^2 = 1$$



- State of n qubits = unit vector in $\mathbf{C}^{2^n}$



- Computation by local gates, rotate the state vector

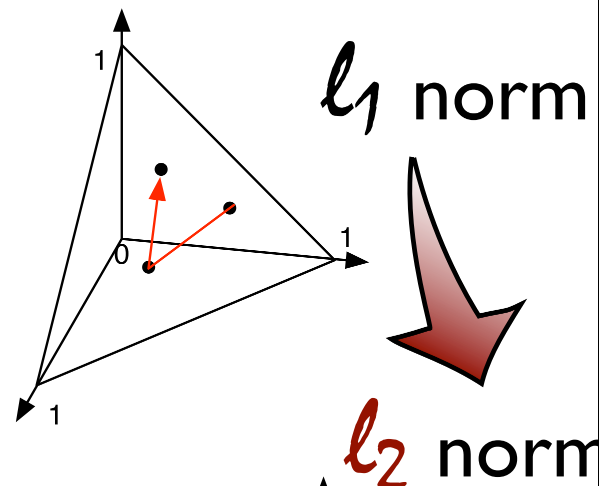
- Observing/measuring system collapses it to a *single* classical bitstring x
  - No exponential parallelism
  - Have to "finesse" the quantum system to output the classical information you want

Classical information processing

- Classical state is a vector of probabilities:

$$\{p_x\}_{x\in\{0,1\}^n} \quad p_x \geq 0 \quad \sum_x p_x = 1$$

- Valid operations are stochastic maps

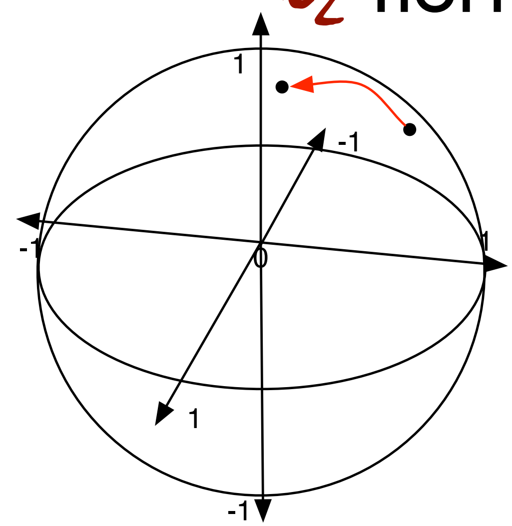


Quantum information processing

- Quantum state is also a vector

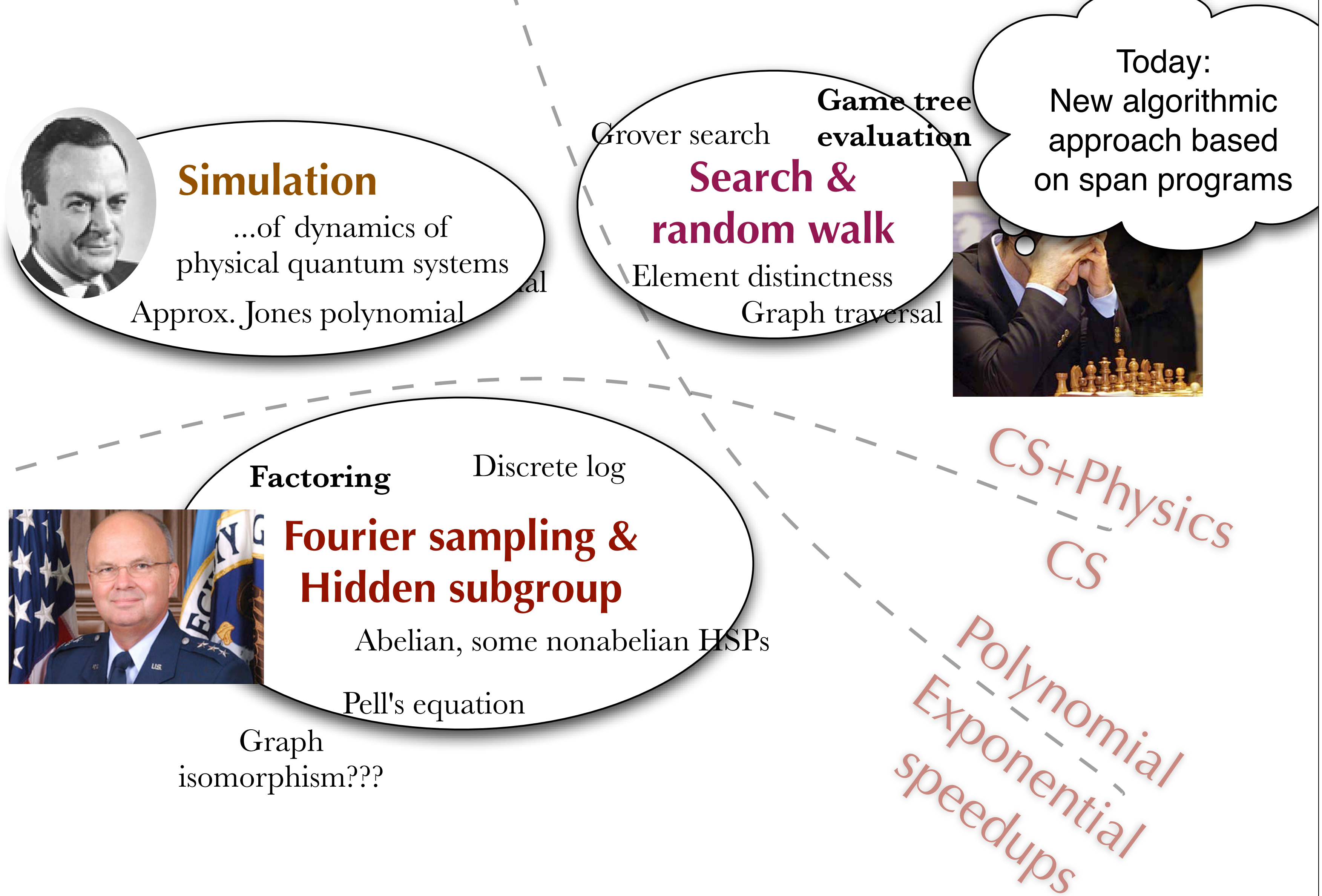
$$\{\alpha_x\}_{x\in\{0,1\}^n} \quad \sum_x |\alpha_x|^2 = 1$$

- Valid operations are rotations (unitaries)



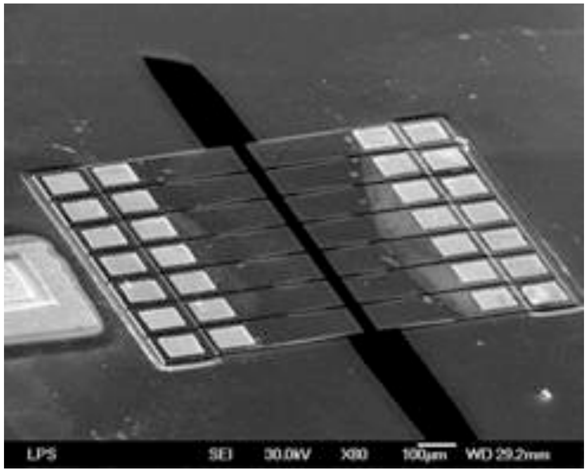
The universe is quantum mechanical but it looks classical because of noise…

Quantum algorithms



Quantum computing in 2008

- **Ion traps**
  - can trap and cool 16-18 qubits
  - can entangle 6-8 qubits in a trap
  - microfabrication of trap arrays on chips, dealing with increased noise



[SHOMSM'05]

  - in next 2-3 years may be able to compute with 40-60 qubits
  - challenges: controlling thousands of traps with dozens of detection channels and lasers along the surface of the chip…

- **Superconducting qubits**
  - 2 qubit local interactions becoming routine
  - nonlocal movement & interactions now possible

  - noise levels seem promising…

- **Other technologies:**
  - Photonic qubits, quantum dots…

- Scaling these systems is a major engineering challenge

- But the basic technologies have been proven,
  there are intermediate rewards
- And there are no known fundamental difficulties, except…

Common obstacle is noise!

- Physically reasonable noise rates are ~1% error per gate, or *maybe* 0.1%

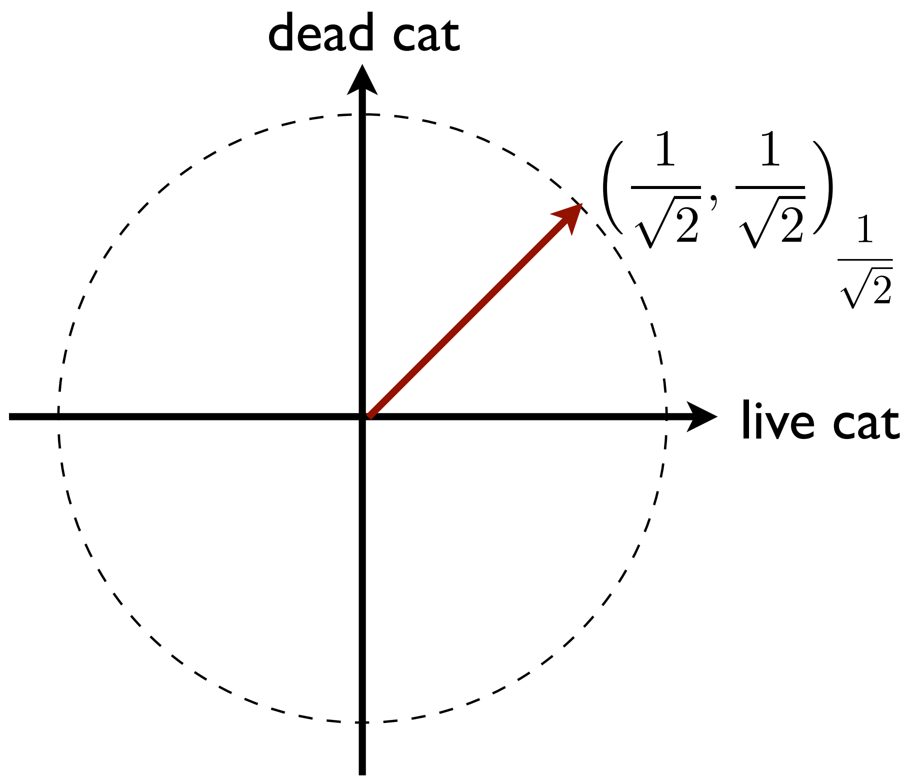
  ∴ Only 100 operations before an error can occur and propagate through the system

- Factoring a 2048-bit number uses
  - 6×10^{11} gates on
  - 10,000 qubits
  - Need error $\lesssim 1/10^{12}$ per gate

K-bit number:
72 K^3 gates versus $e^{K^{1/3}}$ classically
5 K qubits

Noise is fundamental problem for quantum computers: entangled systems are fragile

- Schrödinger's cat:



i.e.

$$\frac{1}{\sqrt{2}}\left(|\text{live cat}\rangle + |\text{dead cat}\rangle\right)$$

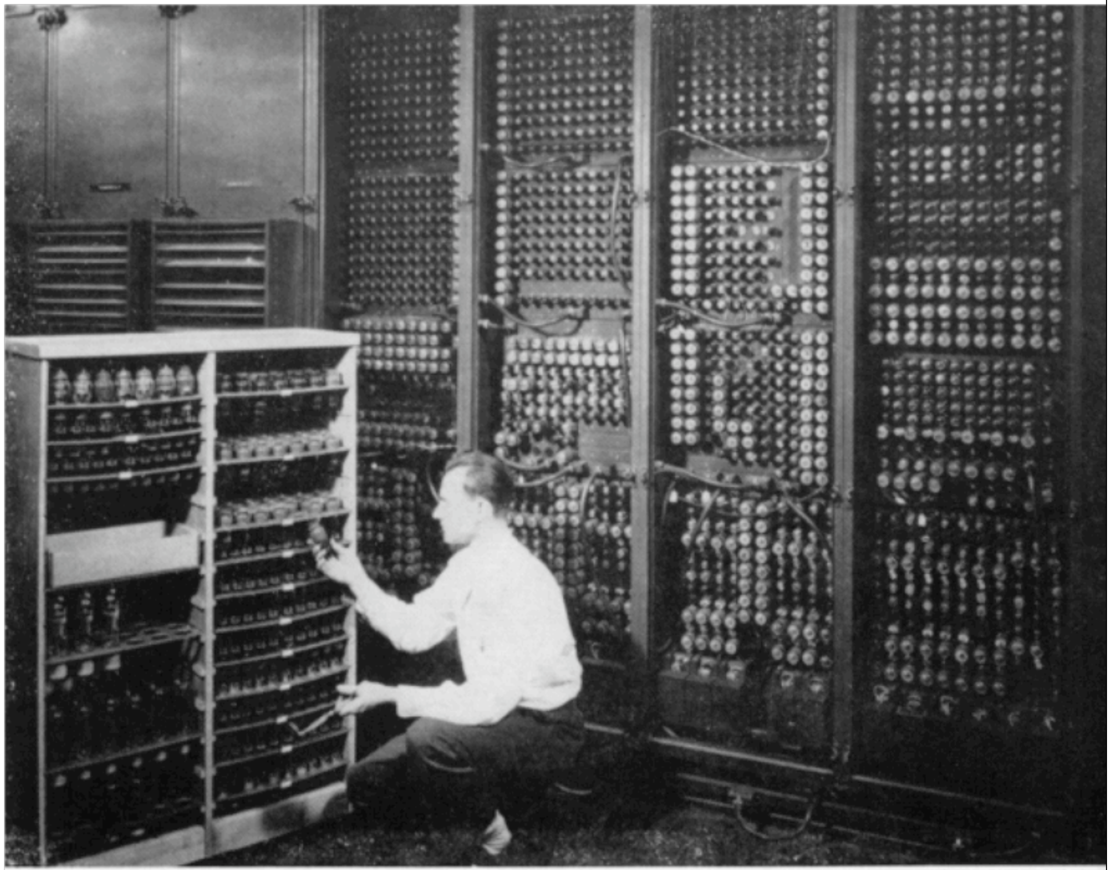
- "Both dead and alive," in superposition; but collapses to one or the other when observed
- A single stray photon can collapse it — and also analogous states in a quantum computer
- Physically reasonable noise rates are ~1% error per gate, or *perhaps* 0.1%

How to deal with noise?

1. Engineering
   - Not enough— noise is fundamental in quantum systems
2. Fault tolerance

   [Von Neumann ’56]

   - Enough to engineer the noise rate beneath a constant threshold,
   - Then effective noise rate can be decreased arbitrarily (and efficiently) using error-correcting codes



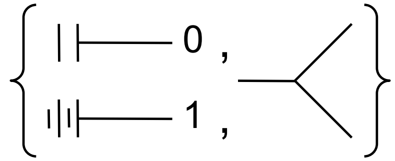
Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

Classical fault tolerance

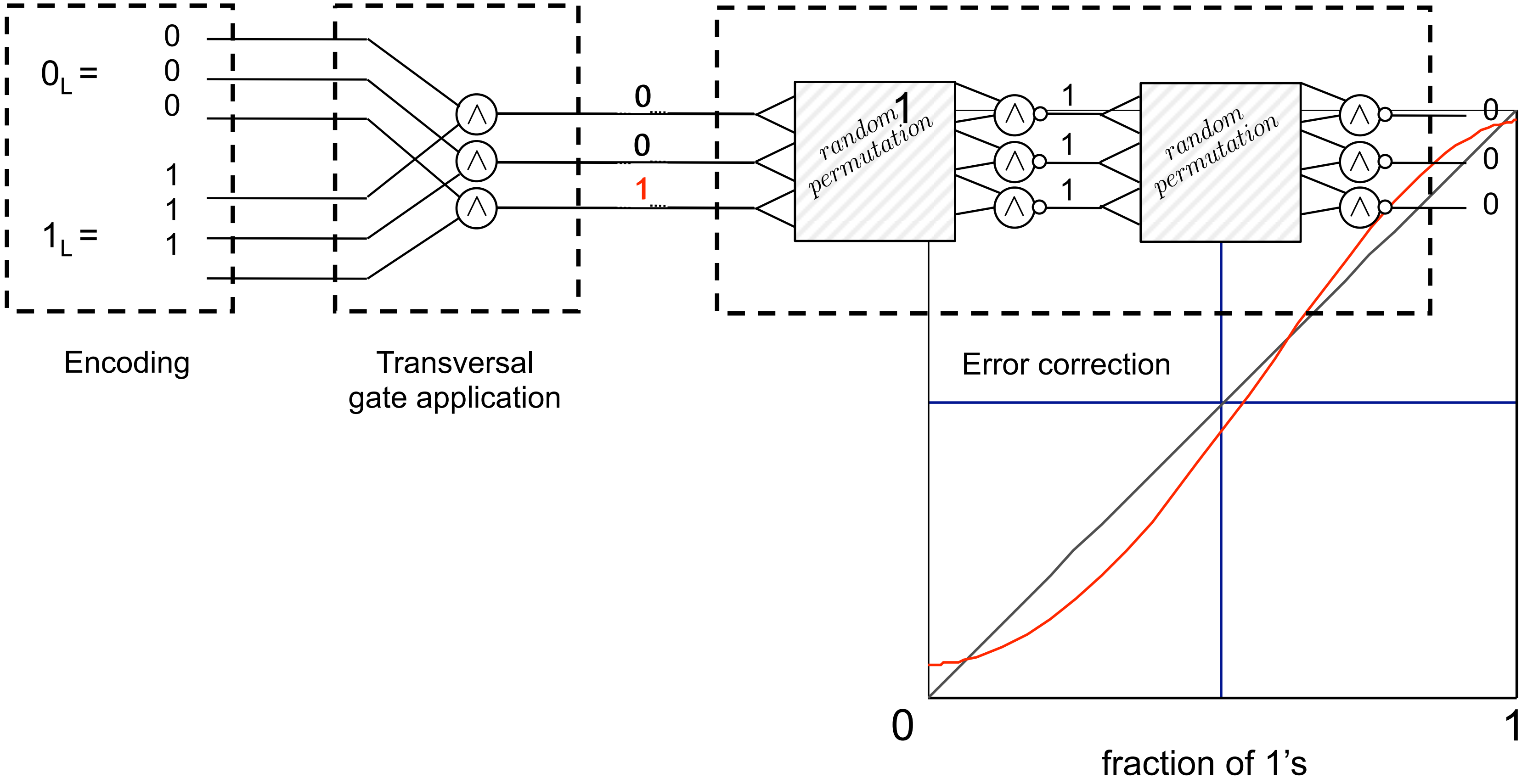
[Von Neumann '56]

Make fault-tolerant a circuit consisting of a universal set of operations, some faulty:

Perfect op's:



Faulty op's: {AND, NOT}



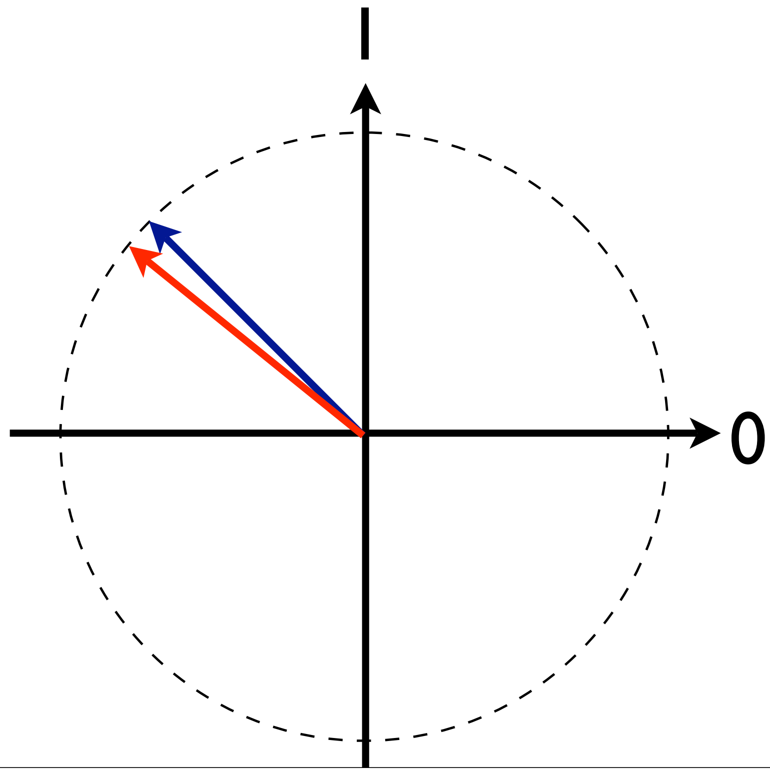
Encoding

Transversal gate application

Error correction

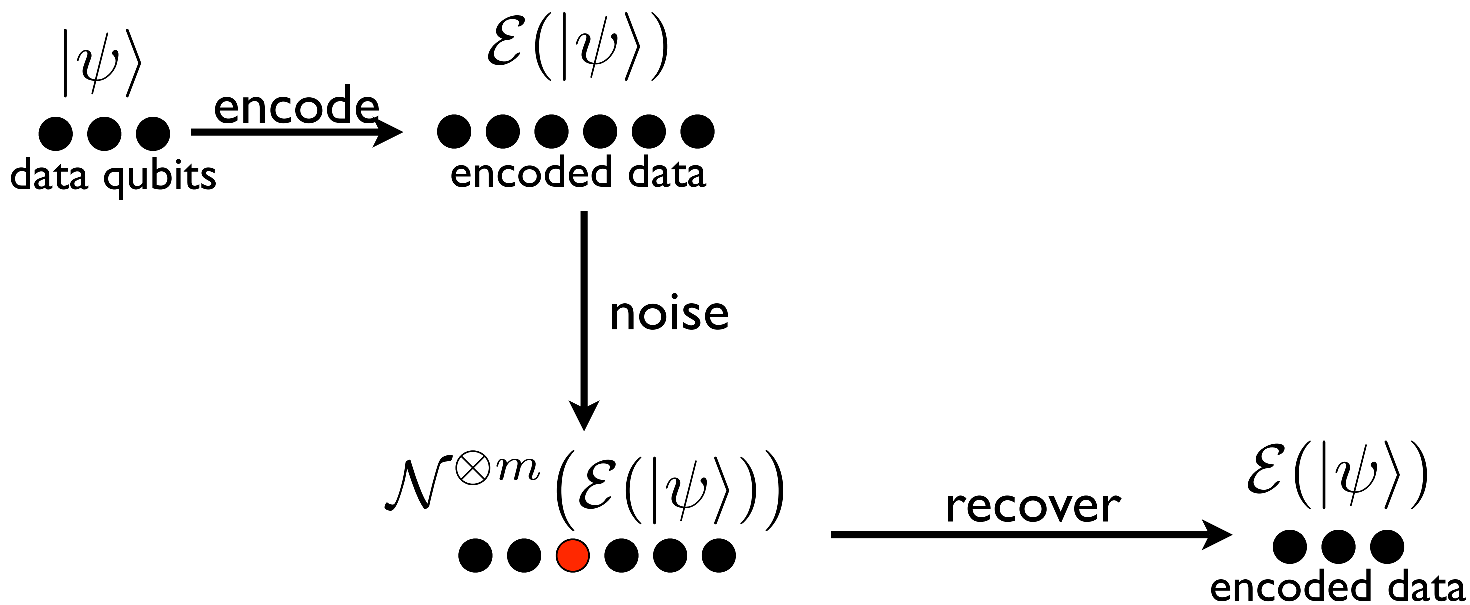
What's different quantumly?

- Quantum problems:
    - Quantum states are continuous, not discrete—need to protect against continuous errors
    - No-cloning theorem: Can't copy a quantum state $|\psi\rangle \mapsto |\psi\rangle|\psi\rangle$, so no immediate analog of the repetition code $0 \mapsto 0^n, 1 \mapsto 1^n$
- But quantum ECCs do exist! [Shor '95]



Operational def. of QECC

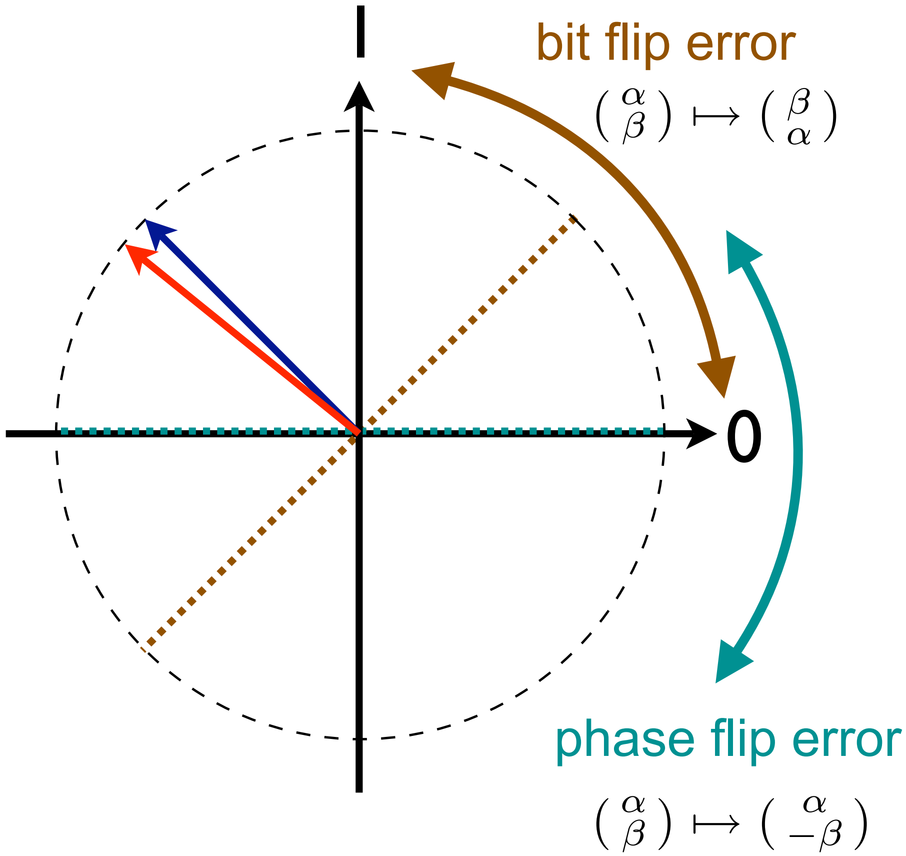
- Quantum problems:
    - Quantum states are continuous, not discrete—need to protect against continuous errors
    - No-cloning theorem: Can't copy a quantum state $|\psi\rangle \mapsto |\psi\rangle|\psi\rangle$, so no immediate analog of the repetition code $0 \mapsto 0^n, 1 \mapsto 1^n$
- But quantum ECCs do exist! [Shor '95] Operationally,



Quantum error-correcting codes exist

- Although quantum states are continuous, correcting a *discrete set* of errors (bit and phase flips) suffices
- Based on classical linear ECCs: QECC comes from *two* linear ECCs

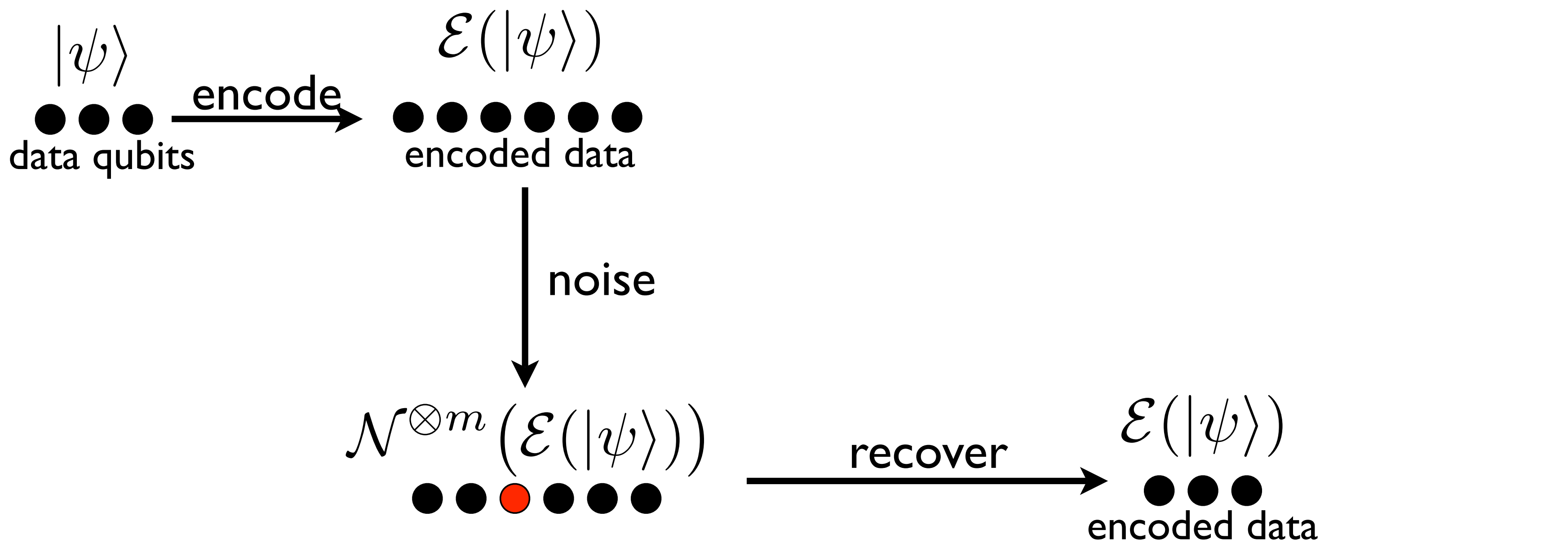
(one for bit flips, one for phase flips)



Quantum error-correcting codes exist

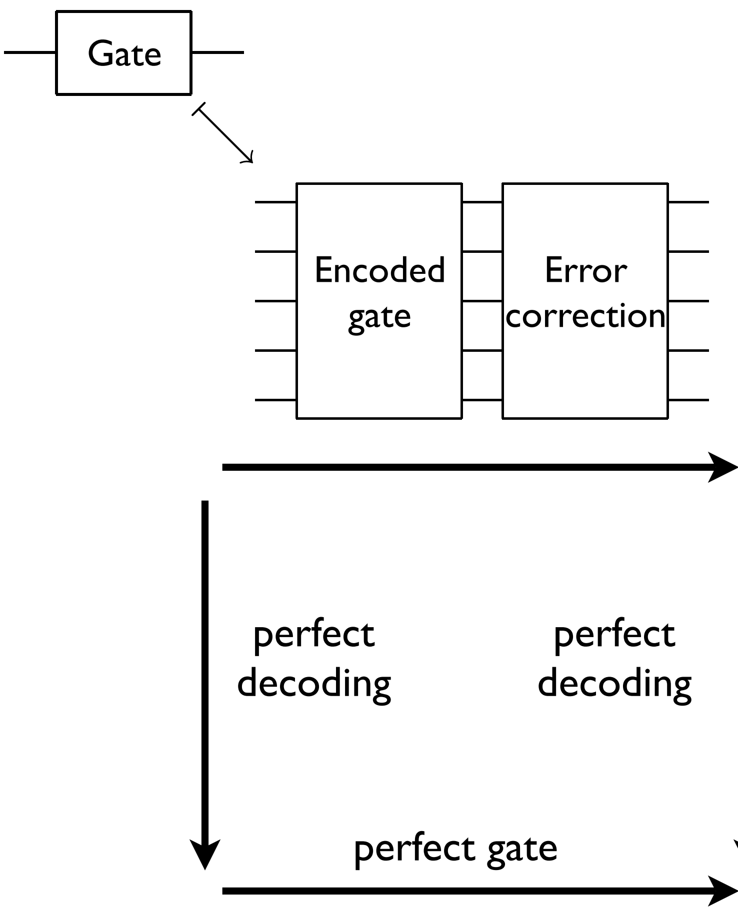
- Although quantum states are continuous, correcting a *discrete set* of errors (bit and phase flips) suffices
- Based on classical linear ECCs: QECC comes from *two* linear ECCs

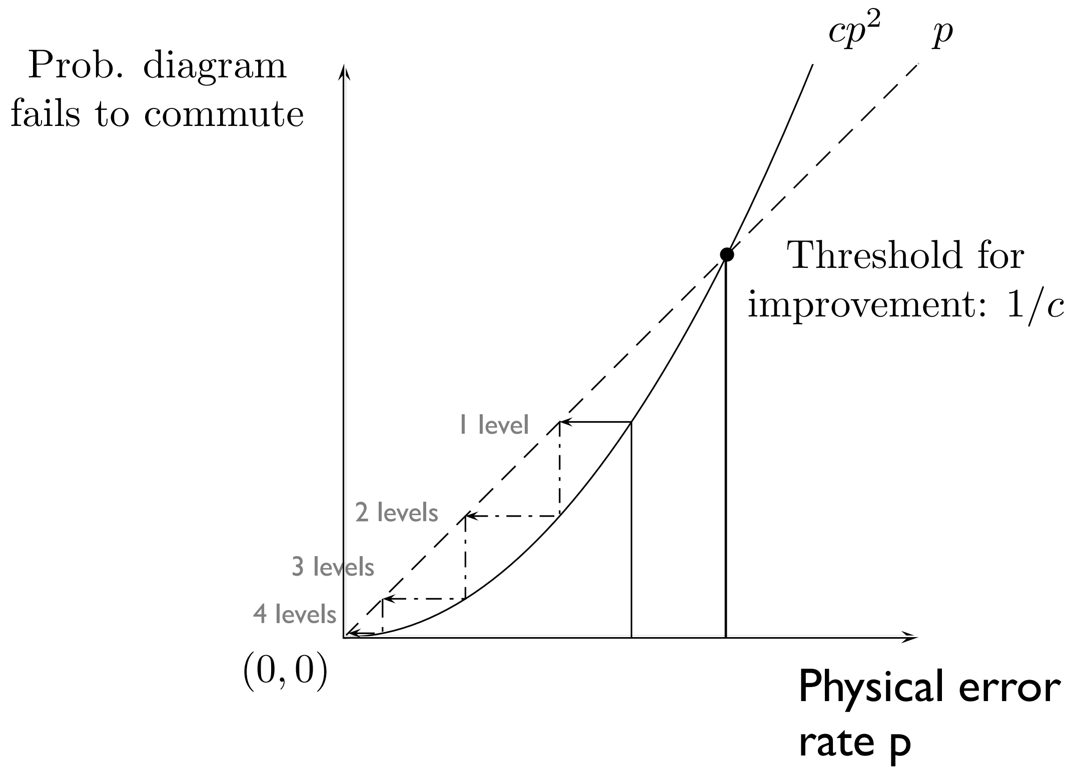
  (one for bit flips, one for phase flips)
- How can we *use* these codes?
  - Need operations as well as memory
  - Error recovery must be resilient to faults during recovery
  - How to encode into them in the first place?? (qu problem)



Fault-tolerance intuition

- Compile ideal circuit into "fault-tolerant" (noise-resistant) version, starting with small QECC:





- Concatenate (i.e., repeat) for arbitrarily improved reliability (so arb^ly^ long calcs), if starting below a constant noise threshold
- Problem: Noise model at encoded level is not the same as the physical noise model!

Abridged History of Quantum Fault Tolerance

- 1996-97: First fault-tolerance results: QECCs, threshold proofs Shor, Steane, Calderbank, Aharonov, Ben-Or, Kitaev, Knill, Laflamme, Zurek, …
  - Proved existence of some *positive* tolerable noise rate using concatenated qu. codes of distance ≥5
  - No explicit lower bounds on tolerable noise rate, but *estimates* were 10^{-6}-10^{-5} noise per gate
  - Moral: Fault tolerance makes quantum computers plausible in the real world

Abridged History of Quantum Fault Tolerance

Proofs

- 1997: Aharonov/Ben-Or, Kitaev: Prove positive tolerable noise rate for codes of distance d≥5
- 2005: R, Aliferis/Gottesman/Preskill: First explicit numerical threshold lower bounds, threshold for distance-3 codes

Estimates & simulations

- 2002: Steane: Correct bit flip errors all at once, and then phase flip errors all at once
  - based on simulations, estimates 3×10^{-3} tolerable noise rate per gate

Simulations using distance-3 codes

- Basic estimates:
  - Aharonov & Ben-Or '97
  - Gottesman '97
  - Knill-Laflamme-Zurek '98
  - Preskill '98
- Optimized estimates:
  - Zalka '97
  - R '04
  - Svore-Cross-Chuang-Aho '05
- 2D locality constraint
  - Szkopek et al '04
  - Svore-Terhal-DiVincenzo '05

Abridged History of Quantum Fault Tolerance

Proofs

- 1997: Aharonov/Ben-Or, Kitaev: Prove positive tolerable noise rate for codes of distance d≥5
- 2005: R, Aliferis/Gottesman/Preskill: First explicit numerical threshold lower bounds, threshold for distance-3 codes

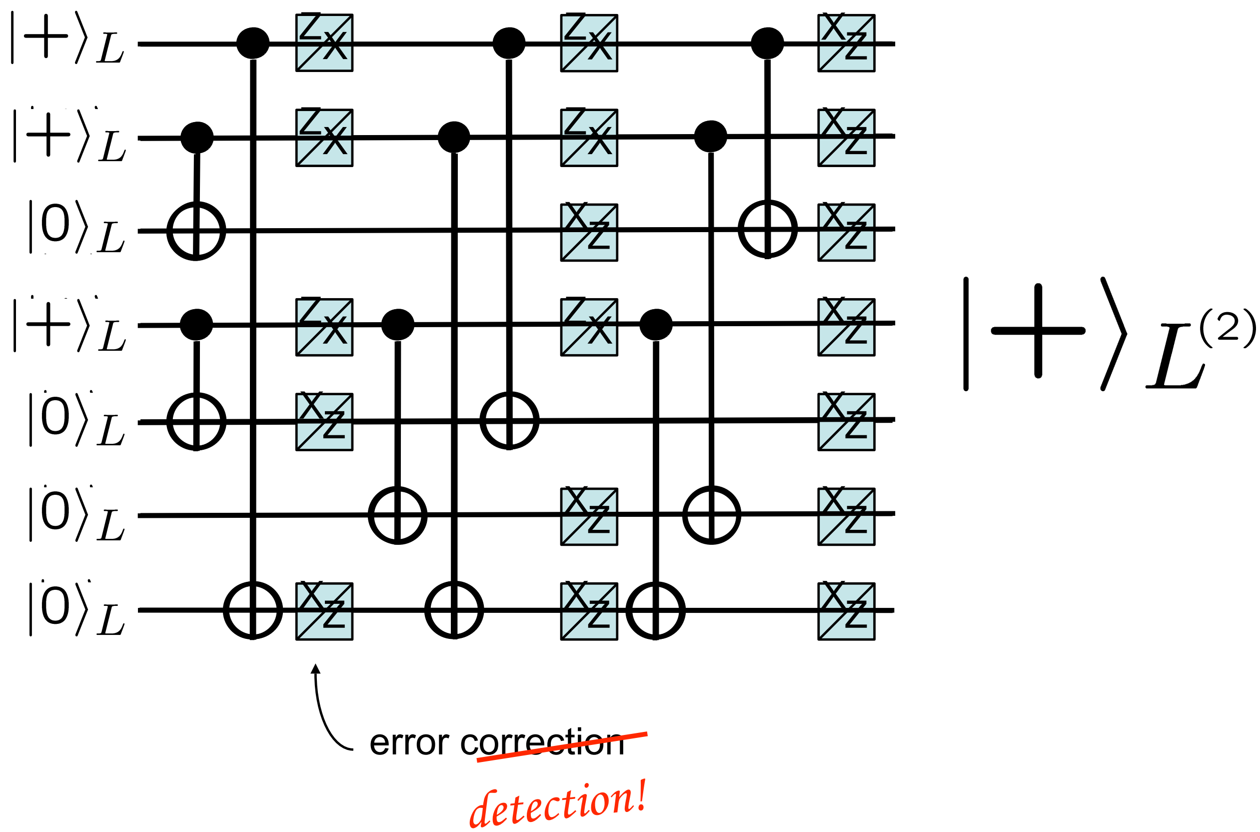
Estimates & simulations

- 2002: Steane: Correct bit flip errors all at once, and then phase flip errors all at once
  - based on simulations, estimates 3×10^{-3} tolerable noise rate per gate

- **Postselection**

Improved threshold result [R '04]

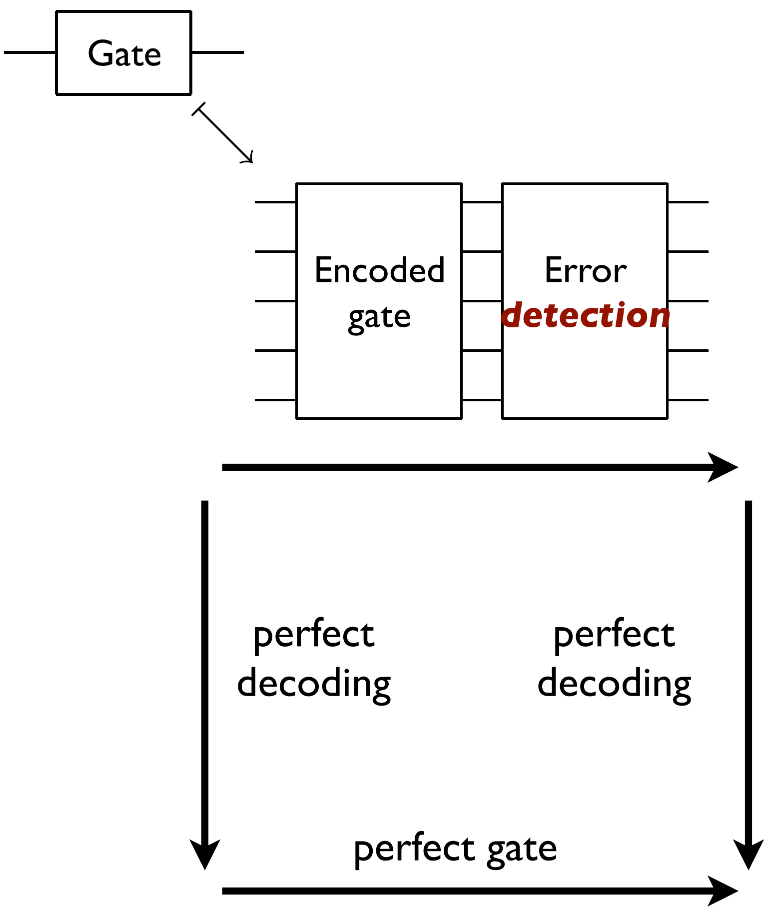
- Modification of standard error correction scheme increases *estimated* threshold 3x, to almost 1%.

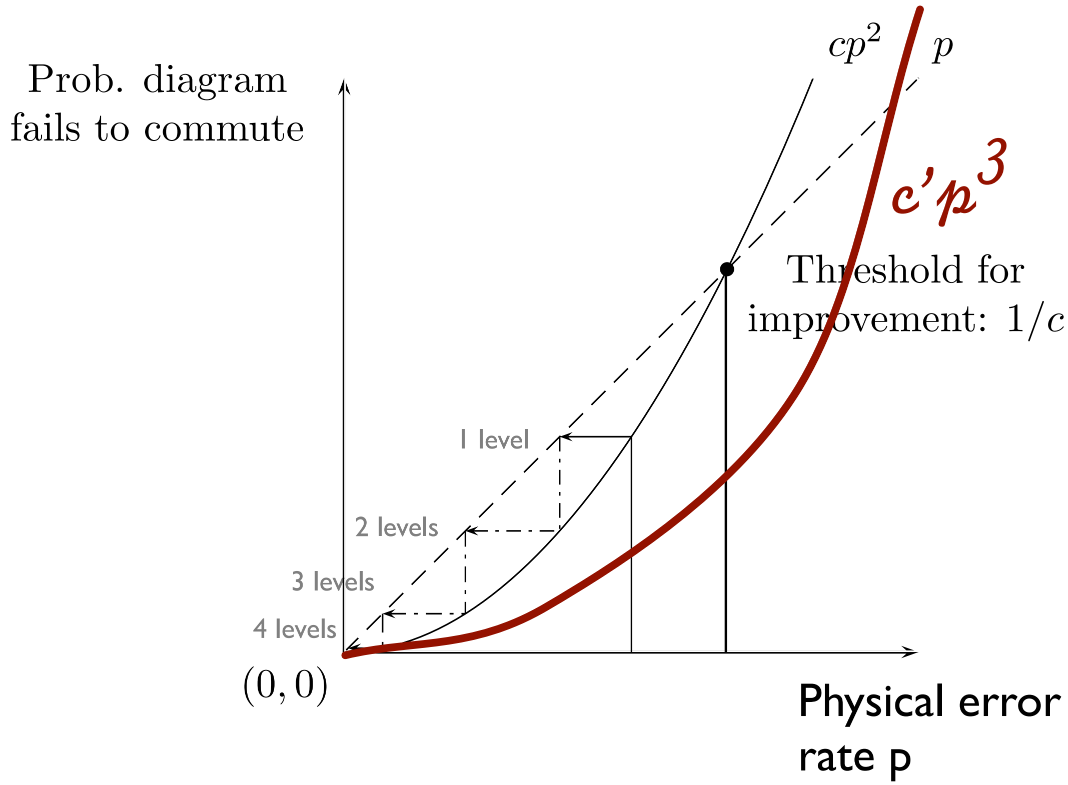


$|+\rangle_L$
$|+\rangle_L$
$|0\rangle_L$
$|+\rangle_L$
$|0\rangle_L$
$|0\rangle_L$
$|0\rangle_L$
$|+\rangle_{L^{(2)}}$
error correction
detection!

Error-*detection*-based fault-tolerance

- Compile ideal circuit into "fault-tolerant" (noise-resistant) version, starting with small QECC:

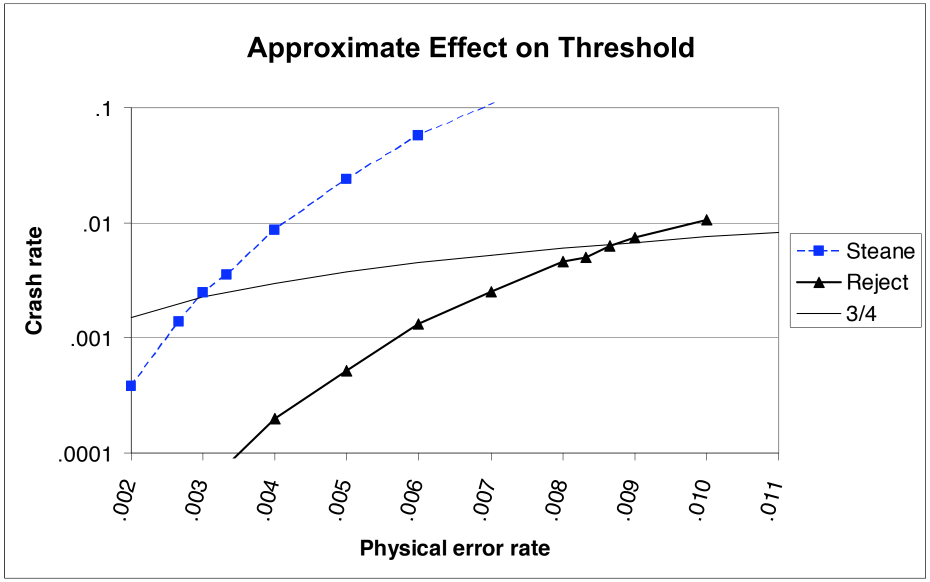


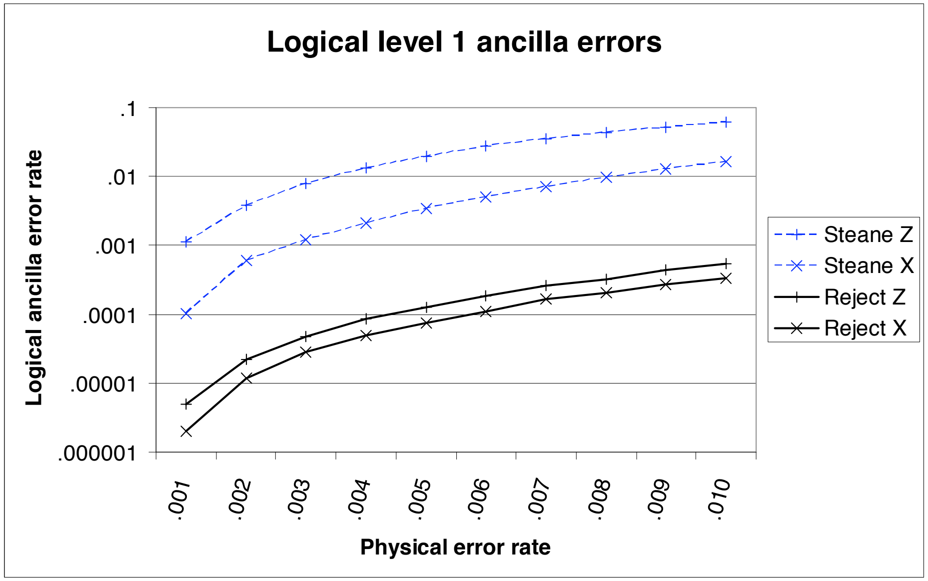


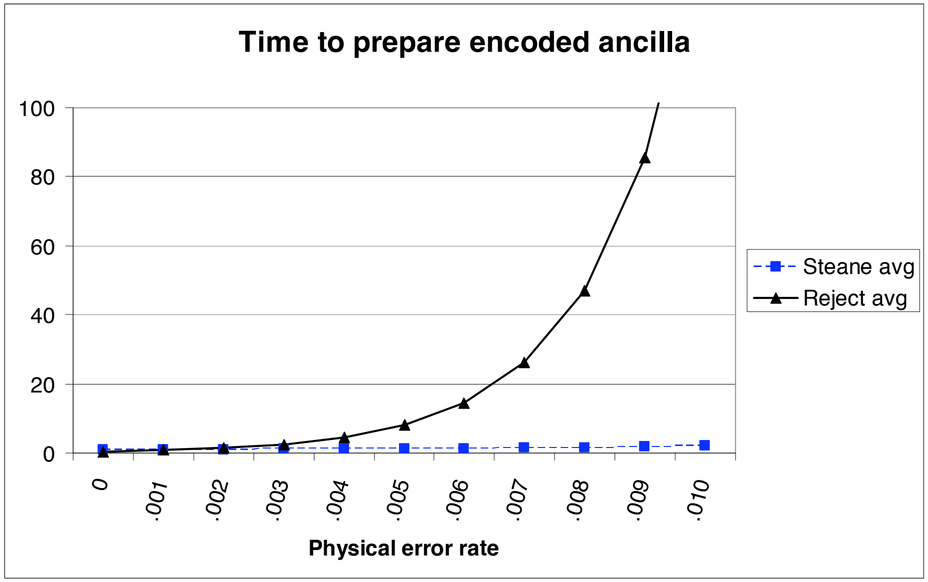
- In simulations, tolerates much higher noise rates than error-correction-based FT schemes
- But (previously) no proven *positive* threshold!

Effect of **postselection** in ancilla preparation

[R '04]







Abridged History of Quantum Fault Tolerance

Proofs

- 1997: Aharonov/Ben-Or, Kitaev: Prove positive tolerable noise rate for codes of distance d≥5
- 2005: R, Aliferis/Gottesman/Preskill: First explicit numerical threshold lower bounds, threshold for distance-3 codes

Estimates & simulations

- 2002: Steane: Correct bit flip errors all at once, and then phase flip errors all at once
  - based on simulations, estimates 3×10^{-3} tolerable noise rate per gate

- **Postselection**

Improved threshold result [R '04]

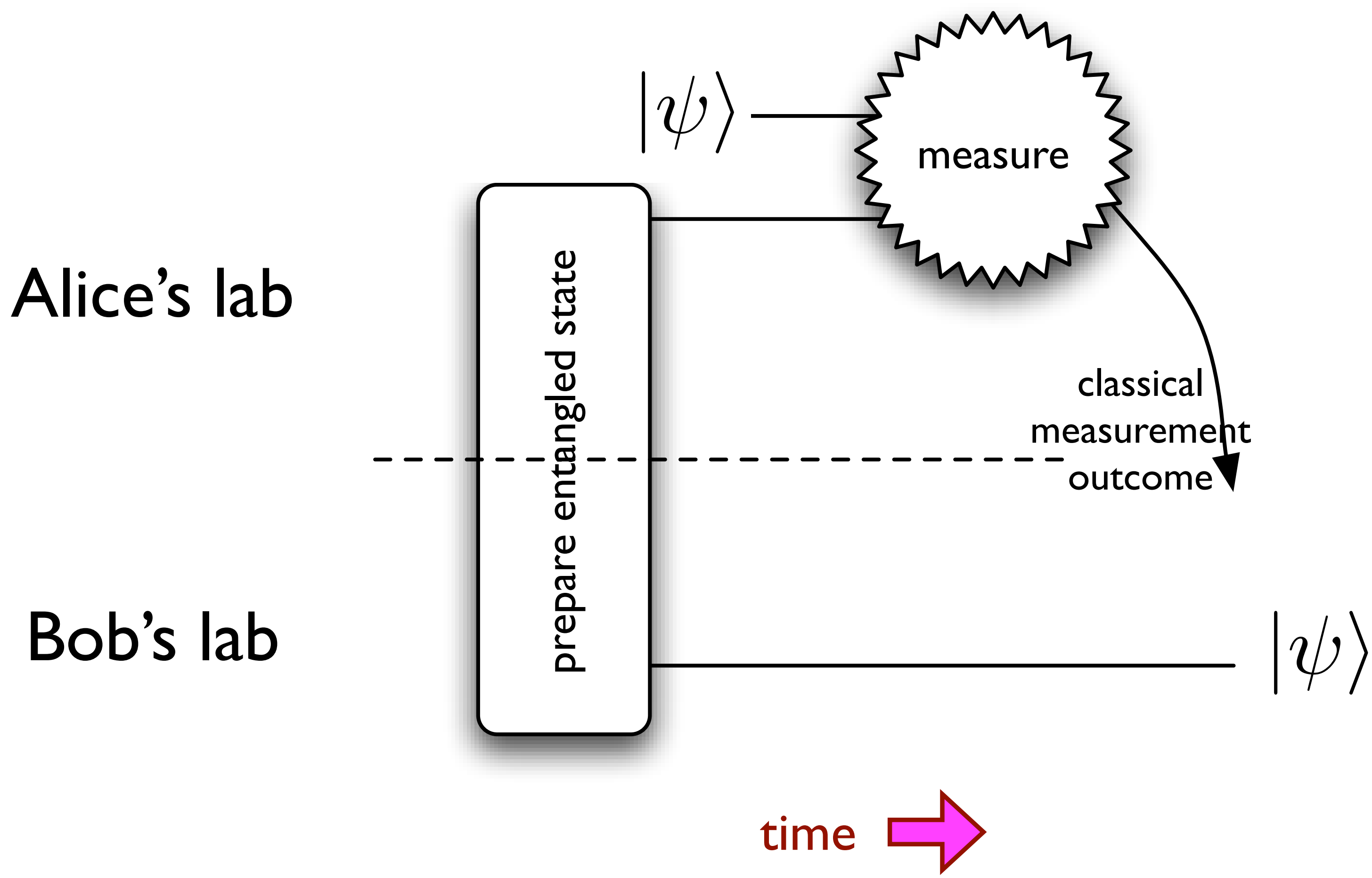
- Modification of standard error correction scheme increases *estimated* threshold 3x, to almost 1%.

- **Postselection + Teleportation**

Knill's threshold result [Knill '04]

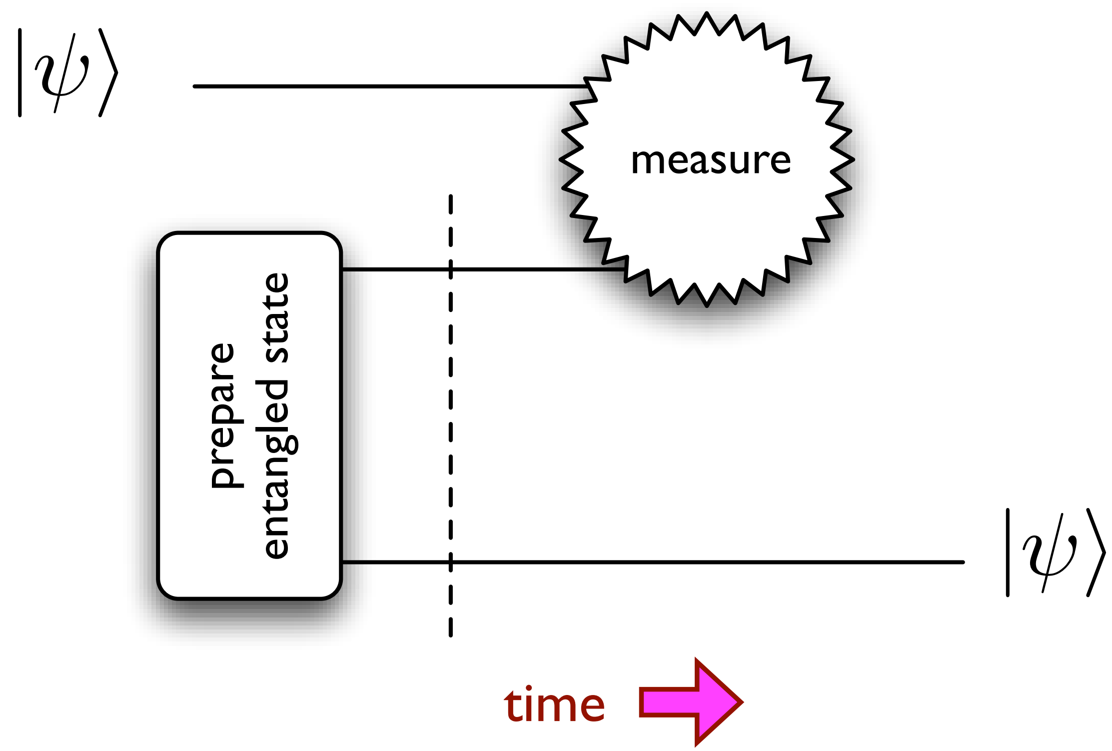
- *Estimated* 3-6% threshold for independent depolarizing errors.

Quantum teleportation



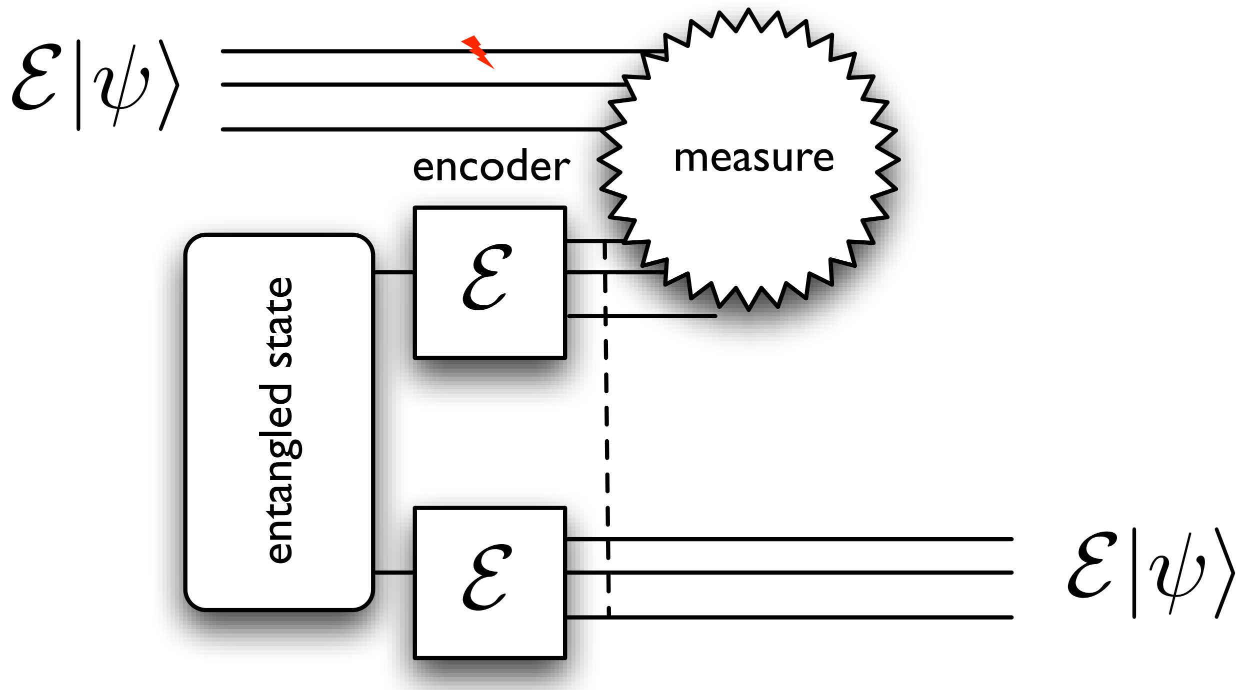
Applying teleportation to fault tolerance

1. Encoding   2. Decoding   3. Error correction   4. Computation



Applying teleportation to fault tolerance

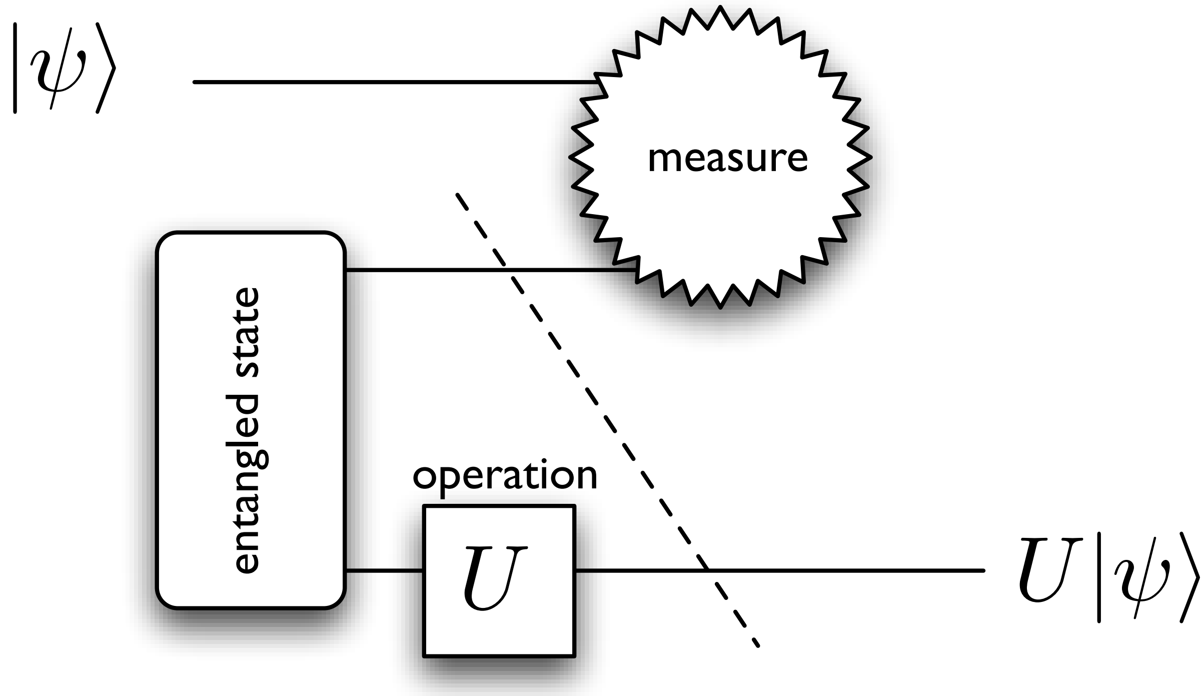
1. Error correction 2. Computation



* decoding measurements using classical computer

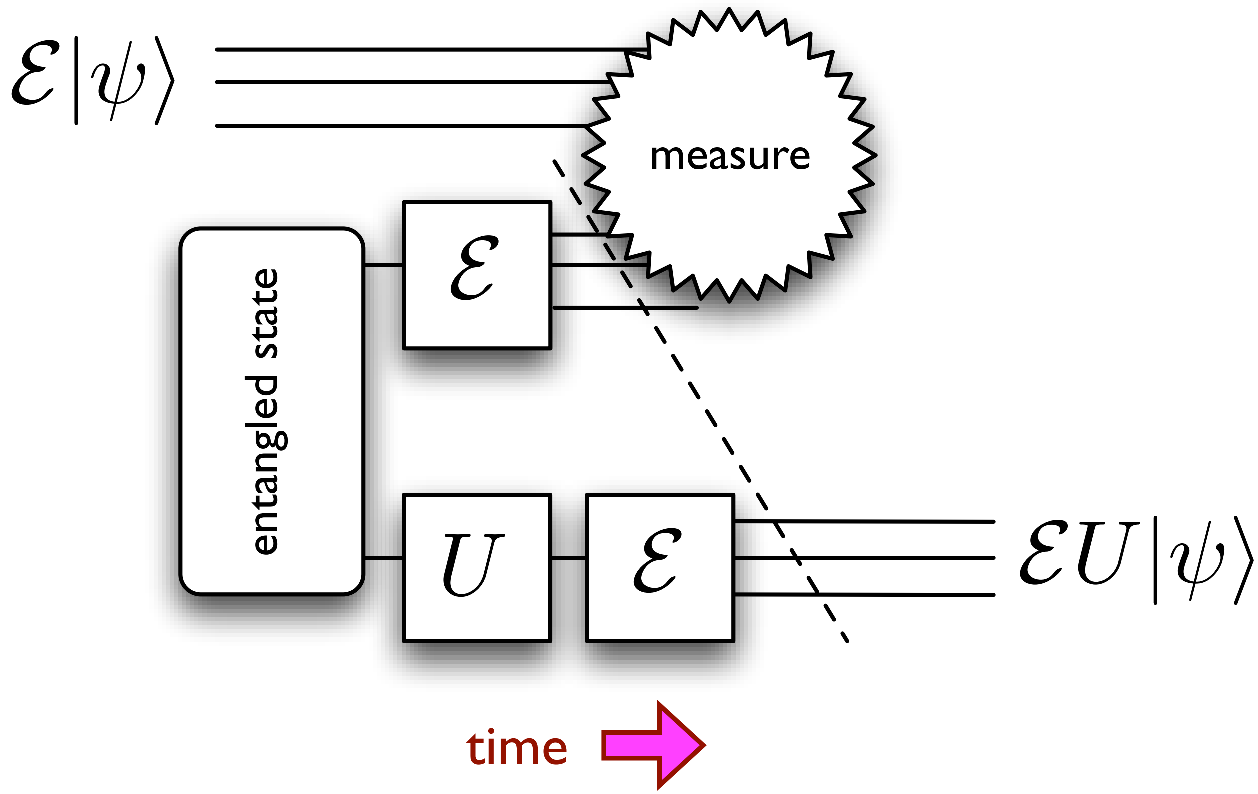
Applying teleportation to fault tolerance

1. Error correction **2. Computation**



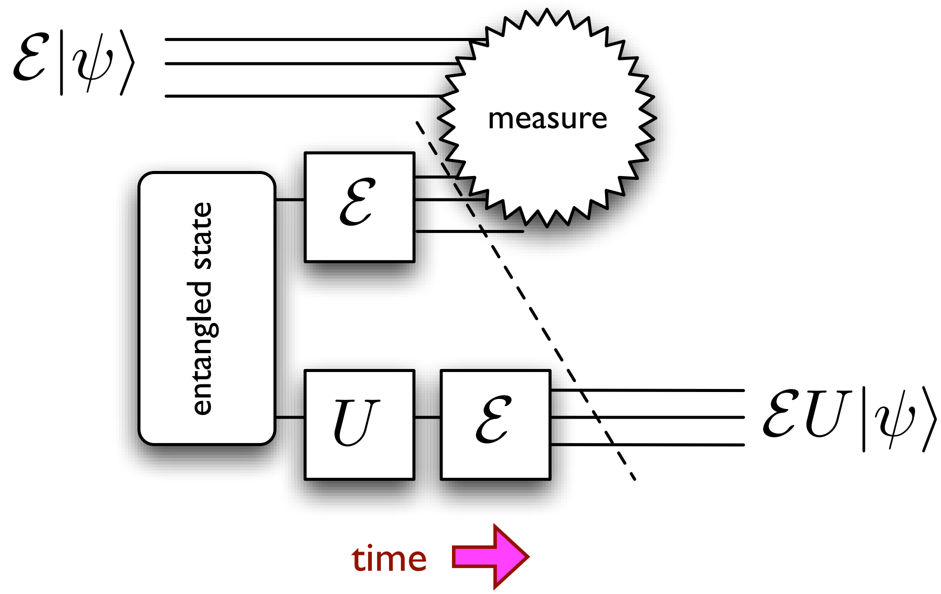
* operation U should be "C_2"

Applying teleportation to fault tolerance
Error correction + Encoded Computation



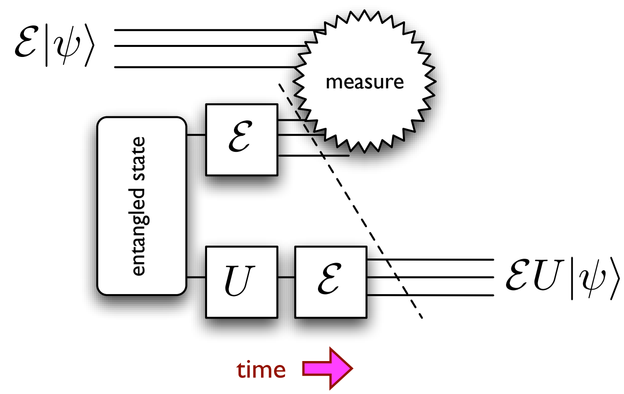
- Teleportation allows for correcting bit flip errors, phase flip errors, *and* doing one step of computation all at once.
- (Provided that we can prepare reliably the necessary resource states.)

Teleported EC + encoded computation



- Teleportation allows for correcting bit flip errors, phase flip errors, *and* doing one step of computation all at once.
- (Provided that we can prepare reliably the necessary resource states.)
- **Note:** We can prepare very good ancilla states, e.g., throwing away all ancillas with any detected errors ("postselection"). We wouldn't want to throw away the data—but the data is isolated from the ancilla state.

Teleported EC + encoded computation



- Teleportation allows for correcting bit flip errors, phase flip errors, *and* doing one step of computation all at once.
- (Provided that we can prepare reliably the necessary resource states.)
- **Note:** We can prepare very good ancilla states, e.g., throwing away all ancillas with any detected errors ("postselection"). We wouldn't want to throw away the data—but the data is isolated from the ancilla state.

Quantum disadvantages

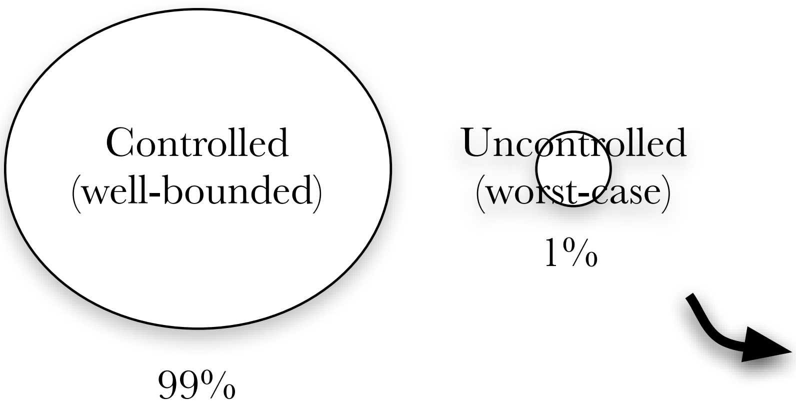
- States are continuous (i.e., analog)
- No-cloning theorem

Quantum advantage!

- Quantum teleportation allows isolating the data from errors

- **Problem:** Although Knill *estimated* tolerable noise rate was 3-6%, proofs could not show that postselection-based schemes tolerated any noise at all!

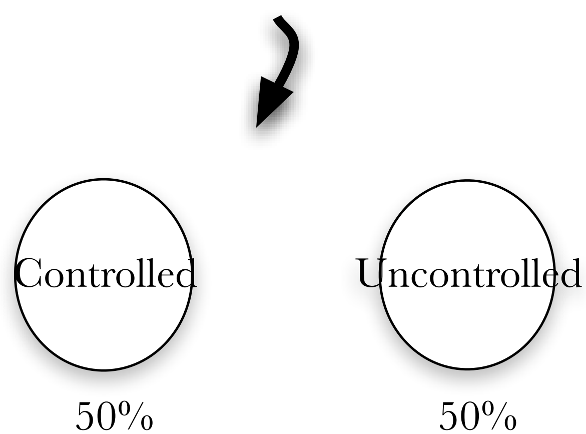
Renormalization frustrates previous proofs



Most of the time, errors are detected — but (counterintuitively) survival probability for uncontrolled portion could be much *higher*

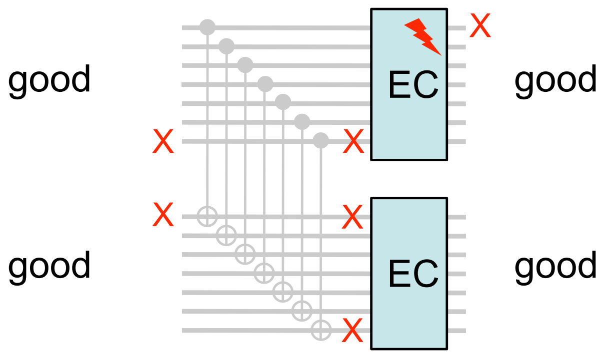
Proofs based on controlling events most of the time, with occasional failures

Uncontrolled fraction of probability mass increases exponentially after renormalizing!

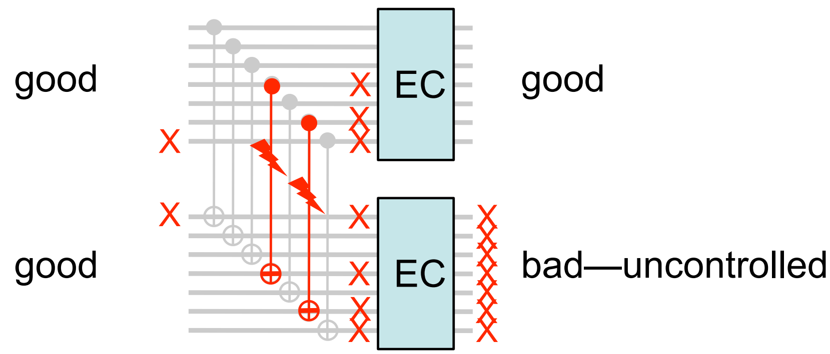


Intuition for Aharonov/Ben-Or's proof

- **Idea:** Maintain inductive invariant of goodness. (A level-k block is good "if it has at most one bad level-(k-1) subblock.")



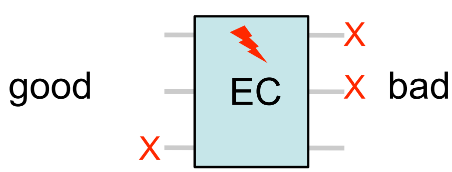
(assuming one level k-1 error, m≥7)



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

- **Problems:**
  - Inefficient analysis: Logical error rate for a distance-d code drops as c $p^{(d-1)/2}$ instead of c $p^{(d+1)/2}$

    ∴ Can't hope for very good rigorous lower bounds on the noise threshold
  - No threshold at all for concatenated d=3 codes, or for postselection-based schemes

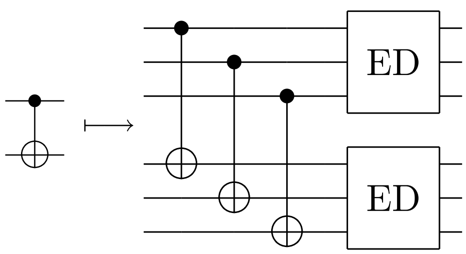


(one level k-1 error is already too many)

- **Problem:** Although Knill *estimated* tolerable noise rate was 3-6%, proofs could not show that postselection-based schemes tolerated any noise at all!
    - Renormalizing the error distribution leads to bad correlations.

Results [R '06]

- *Existence* of tolerable noise rates for many fault-tolerance schemes, including:
    - Schemes based on error-*detecting* codes, not just ECCs (Knill-type)
    - Distance-3 codes, and more efficient "Fibonacci"-type schemes (d=2 codes)



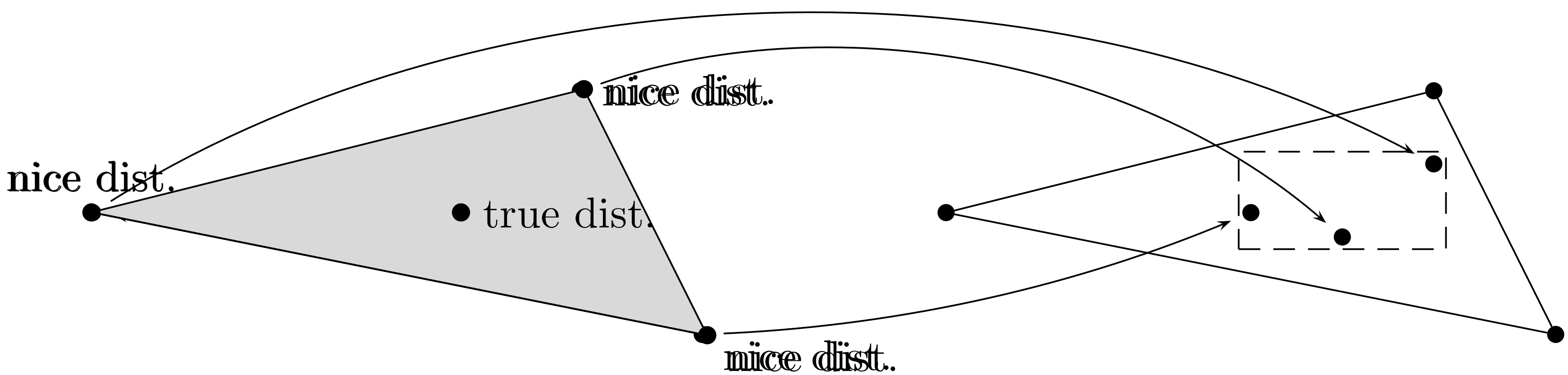
- Tolerable threshold *lower bounds**
    - 0.1% simultaneous depolarization noise†
    - 1.1%, if error model known *exactly*

* Subject to minor numerical caveats

† Versus .02% best lower bound for error-correction-based FT scheme [Aliferis, Cross 2006]

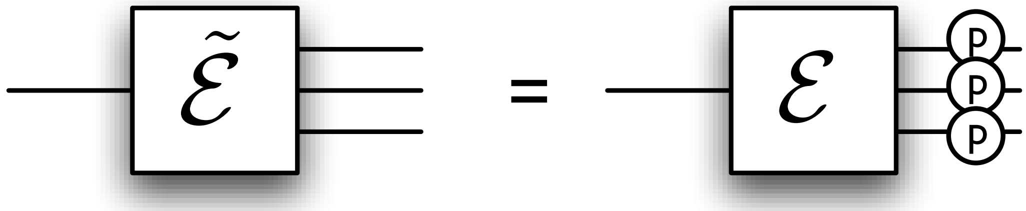
Techniques

- Main new technique is to maintain close control over the distribution of errors in the quantum computer
  (Previous threshold proofs had used a “worst-case” criterion for error behavior that blew up during renormalization.)
  - Rewrite true error distribution as a mixture of nearby distributions whose error distributions lack nasty correlations



Bitwise-independent noise is nice...

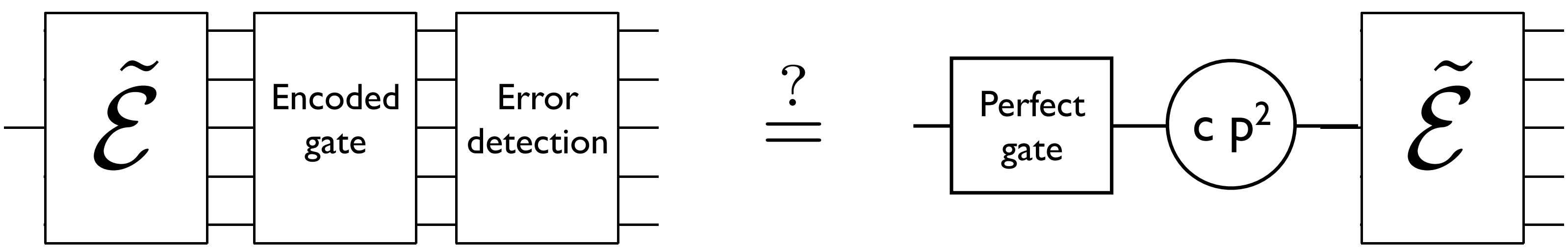
- **Def:** Noisy encoder = perfect encoder, followed by bitwise-indep. noise at rates ≤p.



(tool for analysis—such encoders don't actually exist)

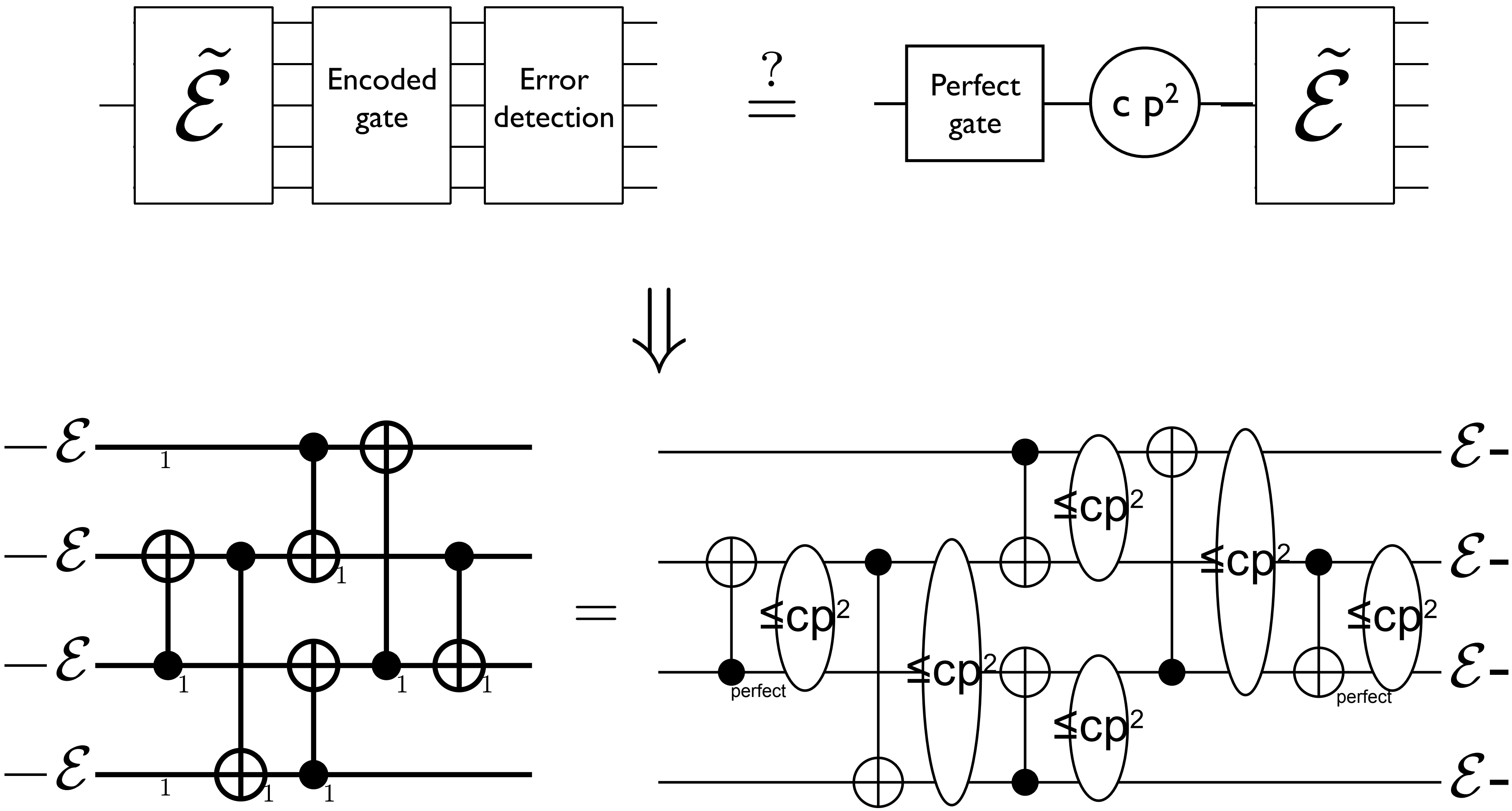
Induction claim?

(much stronger than Aharonov/Ben-Or's claim)



bitwise-independent errors preceding encoded gate

bitwise-independent errors following perfect gate, plus quadratically suppressed independent logical errors

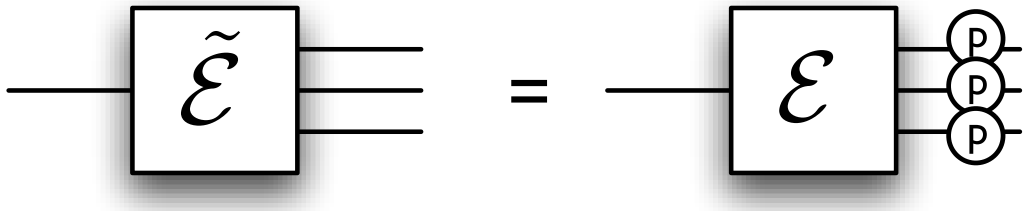


encoded FT circuit

- Since the error model is preserved (level-one logical errors have the same form as physical errors), the analysis can be repeated to give a threshold

Bitwise-independent noise is nice...

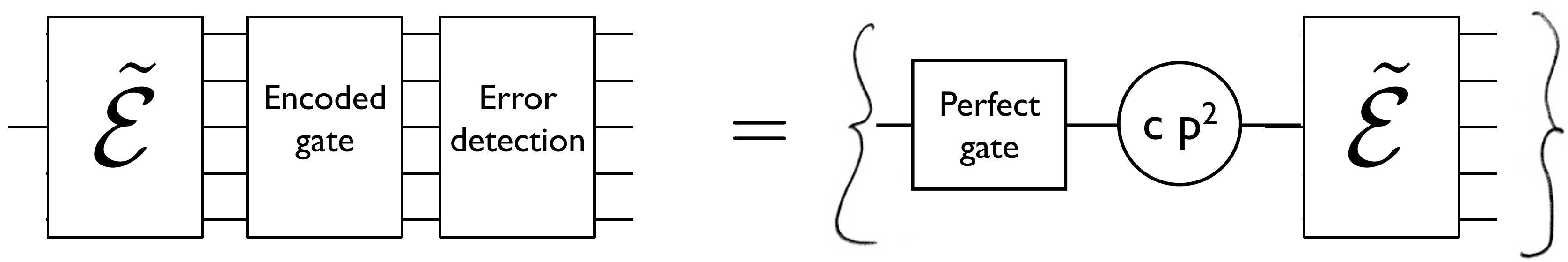
- **Def:** Noisy encoder = perfect encoder, followed by bitwise-indep. noise at rates ≤p.



(tool for analysis—such encoders don't actually exist)

Induction claim

(much stronger than Aharonov/Ben-Or's claim)

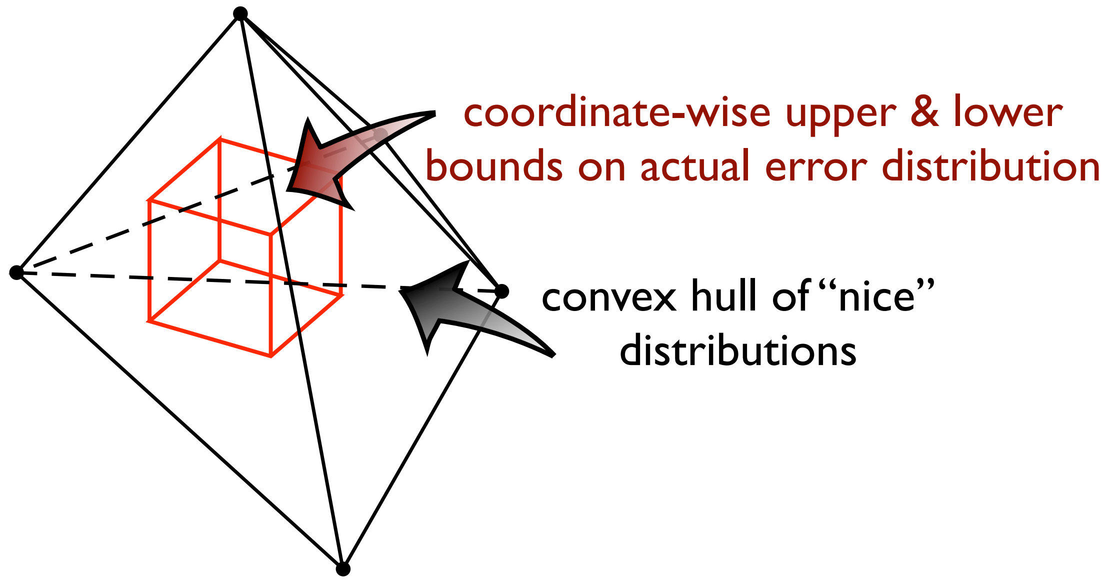


bitwise-independent errors preceding encoded gate

bitwise-independent errors following perfect gate, plus quadratically suppressed independent logical errors

Details in proving that mixing works

- Numerical approach (for numerical threshold lower bounds)

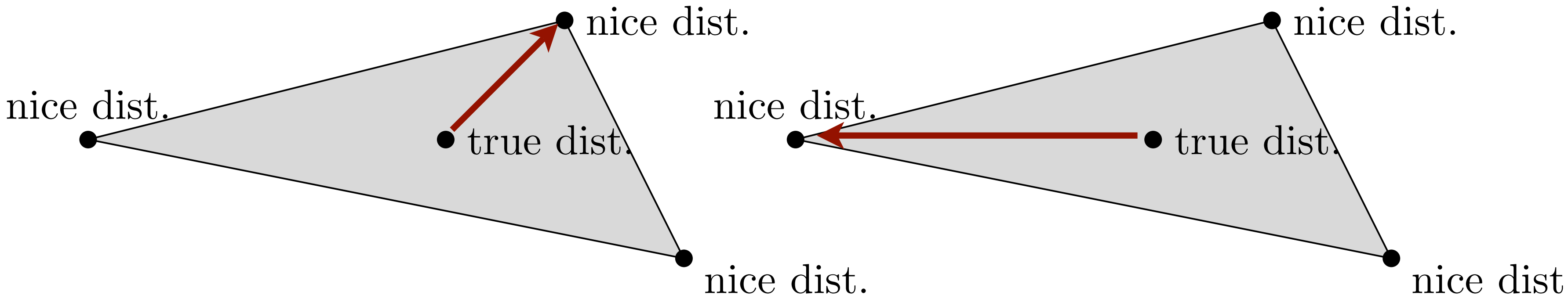


- Existence argument (for threshold existence proofs):
  - characterize convex hull of dit-wise independent distributions (a simplex)
  - "pull back" actual distribution onto distribution on dits

- Must also obtain universality — CNOT and similar "linear" gates can be efficiently simulated on a classical computer. Need a nonlinear operation (AND or Toffoli). Use "magic states distillation."

Conclusions

- Conclusion: Mixing argument shows that concatenation works to reduce errors. Error events are correlated, but error correlations do *not* explode.
- Correlations manifest themselves as asymmetries in the conditional error models



—violates a key assumption of Knill, that all gates have symmetrical failure models, *at all levels of concatenation*

- With postselection, gate error rates that are asymmetrically too low can be just as bad as error rates that are too high
- Are Knill's simulations too optimistic?

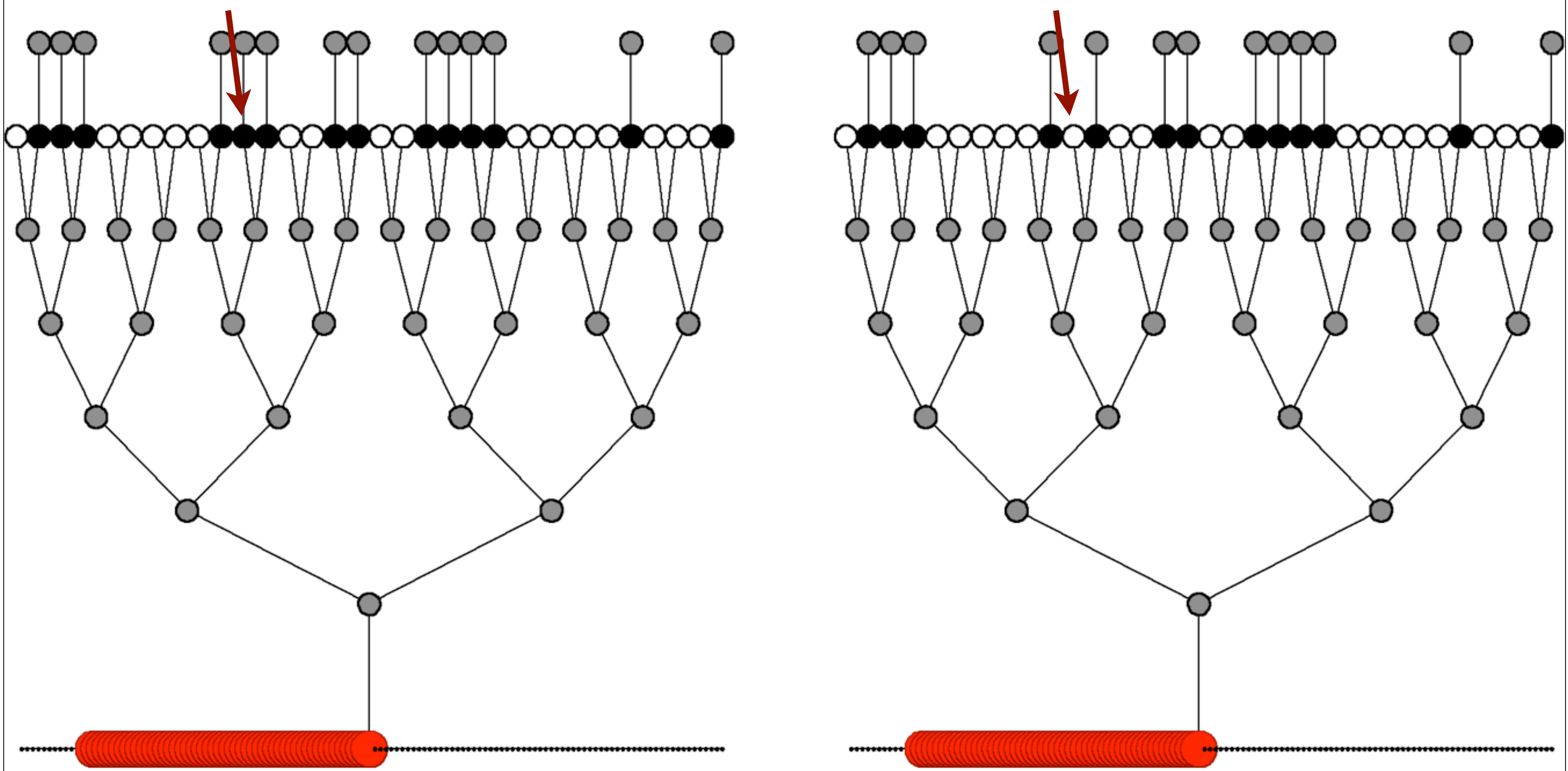
More fault-tolerance work...

- Magic states distillation (for teleporting into a universal gate set) [R'05, '06]
- Optimizing ancilla verification [R'06]
- Higher-order-accurate composite pulses, based on quantum search algorithm, for eliminating noise *without encoding* [R'05]

Open questions in fault tolerance

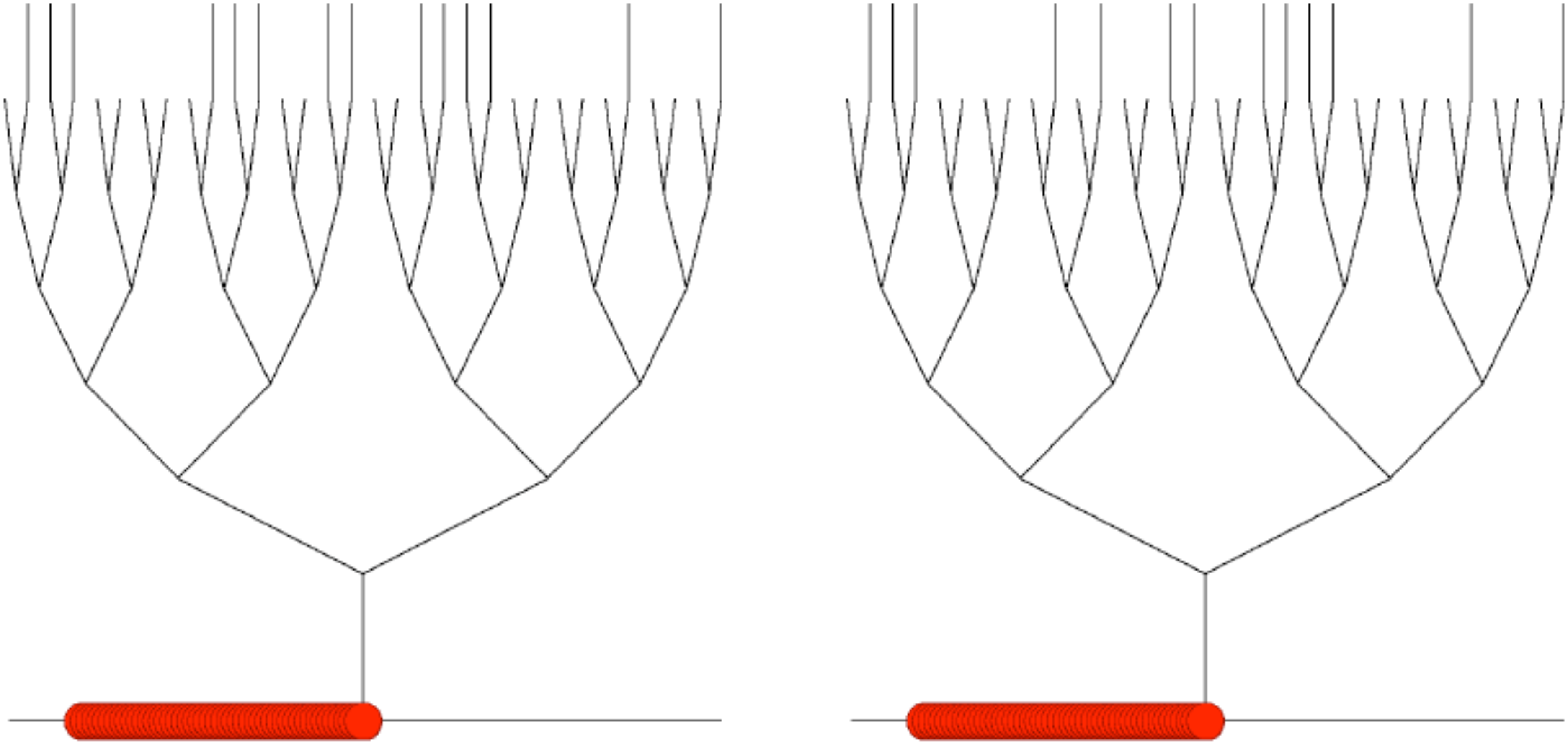
- Quantum computers need fault-tolerance techniques if they are to scale, but…
- Current FT schemes are not good enough
  - Need to increase tolerable noise rate, reduce overhead
  - So far, the biggest improvements have come *not* from optimizations or customizations, but rather from new quantum concepts that unify. The next division to remove is the code concatenation levels.
- Foundations:
  - Extend applicability of threshold proofs
  - Improve threshold upper bounds
- Connecting full-blown fault-tolerance schemes to implementations
  - Specialized, low-level error prevention (e.g., composite pulses, DFSs)

[Farhi, Goldstone, Gutmann '07]

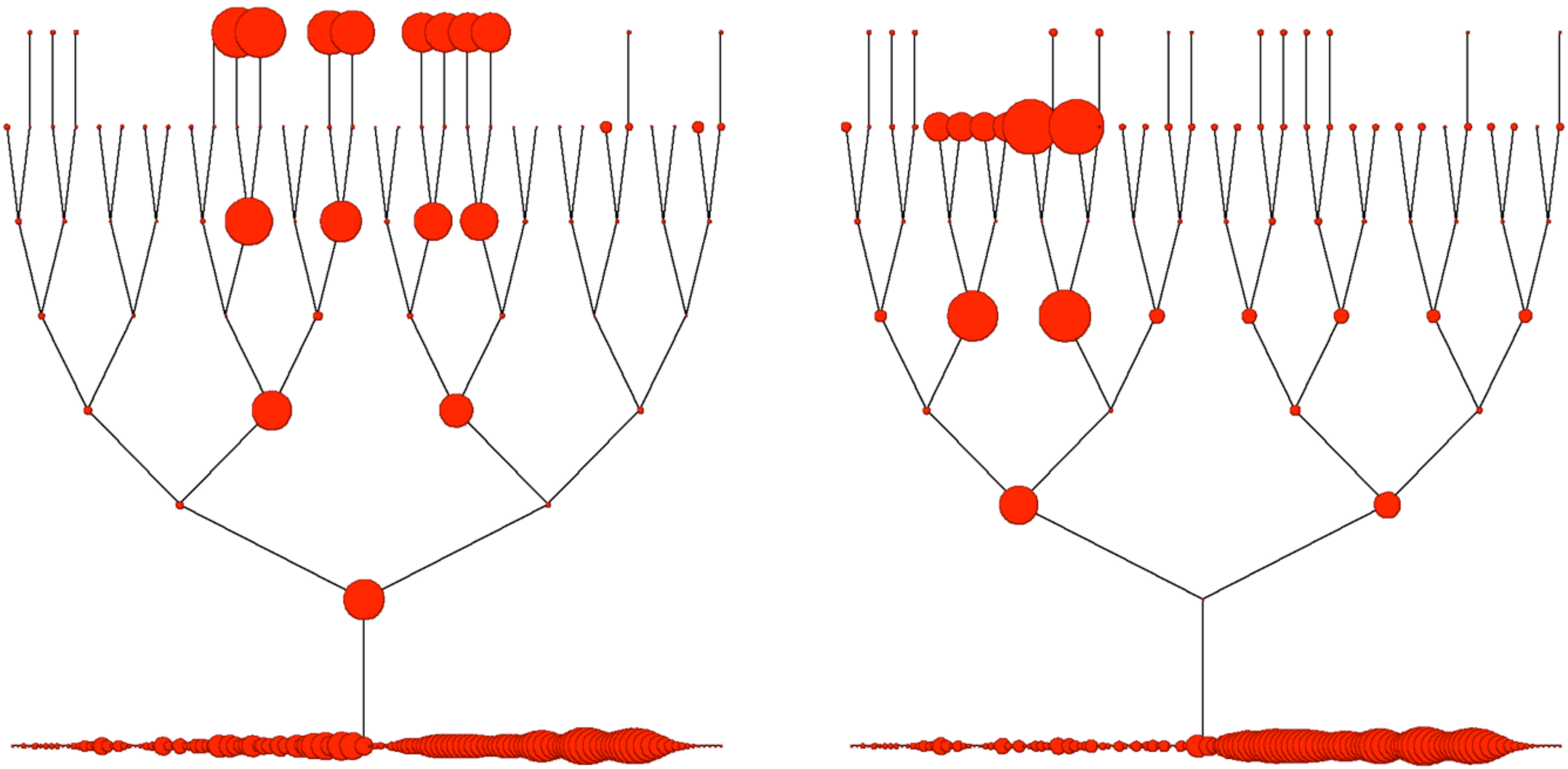


Scatter a wave against the tree…

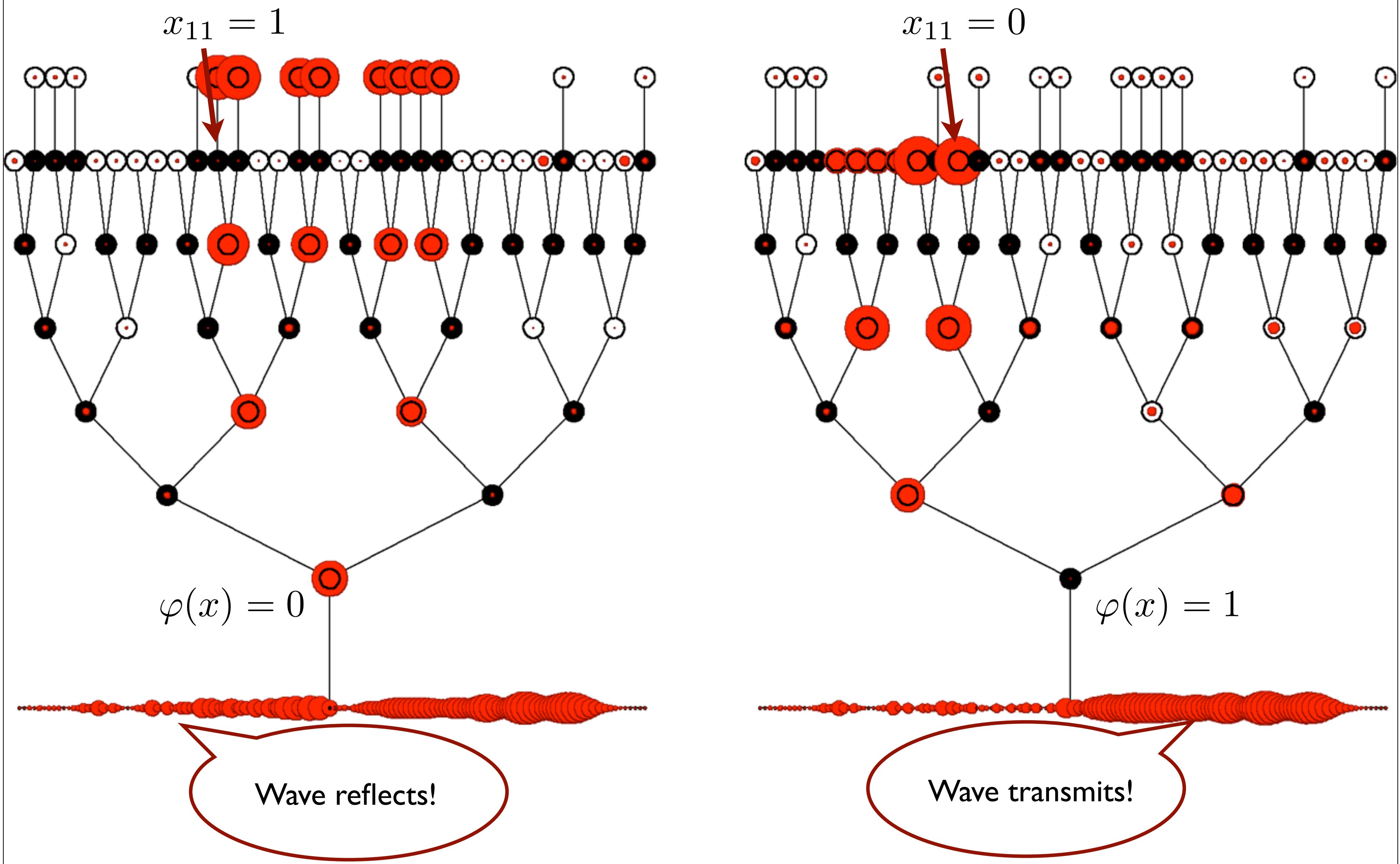
FGG quantum walk $|\psi_t\rangle = e^{iA_G t}|\psi_0\rangle$



FGG quantum walk $|\psi_t\rangle = e^{iA_G t}|\psi_0\rangle$

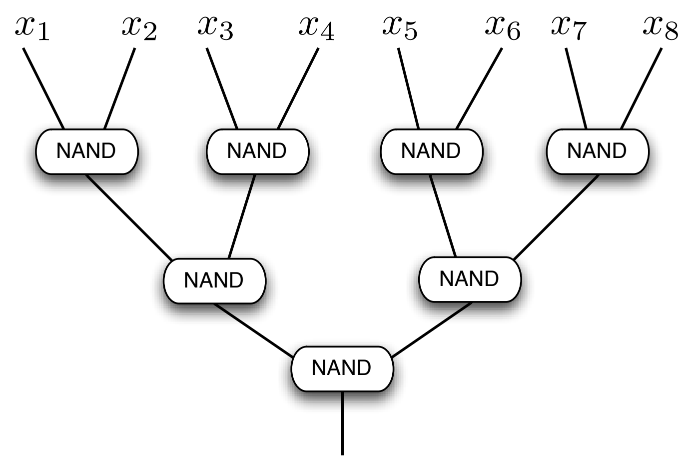


FGG quantum walk $|\psi_t\rangle = e^{iA_G t}|\psi_0\rangle$



Farhi, Goldstone, Gutmann ‘07 algorithm

- **Theorem** ([FGG ‘07, CCJY ‘07]): A balanced binary NAND formula can be evaluated in time $N^{\frac{1}{2}+o(1)}$.



Questions:

1. Why does it work?
2. How does it connect to what we know already?
3. How does it generalize?
4. What kinds of problems can we hope to solve with this technique?

Farhi, Goldstone, Gutmann ‘07 algorithm

- **Theorem** ([FGG ‘07, CCJY ‘07]): A balanced binary NAND formula can be evaluated in time $N^{½+o(1)}$.

Questions:

1. Why does it work?
2. How does it connect to what we know already?
3. How does it generalize?
4. What kinds of problems can we hope to solve with this technique?

Answers:

“span programs” [Karchwer/Wig. ‘93]

formula evaluation problem over extended gate sets

Farhi, Goldstone, Gutmann '07 algorithm

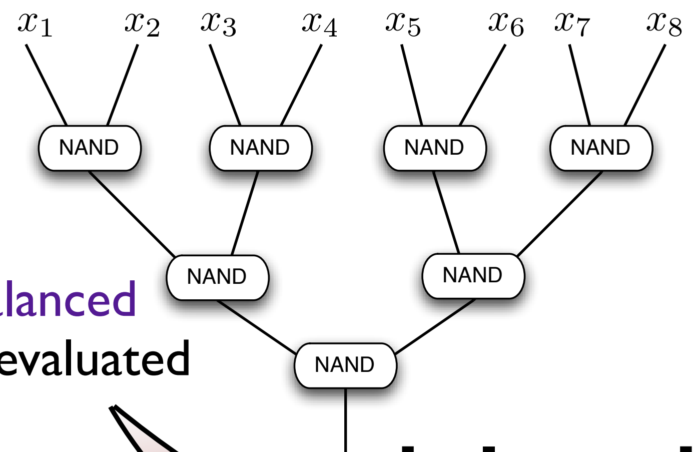
- **Theorem** ([FGG '07, CCJY '07]): A balanced binary NAND formula can be evaluated in time $N^{1/2+o(1)}$.

Questions:	Answers:
1. Why does it work?	
2. How does it connect to what we know already?	"span programs" [Karchwer/Wig. '93]
3. How does it generalize?	
4. What kinds of problems can we hope to solve with this technique?	formula evaluation problem over extended gate sets
5. No, really, *WHY* does it work?	???

[FGG ‘07] algorithm

- **Theorem** ([FGG ‘07, CCJY ‘07]): A balanced binary AND-OR formula can be evaluated in time $N^{½+o(1)}$.

  Analysis by scattering theory.



balanced, more gates

unbalanced AND-OR

[ACRŠZ ‘07] algorithm

- **Theorem**:
  - An “approximately balanced” AND-OR formula can be evaluated with $O(\sqrt{N})$ queries (optimal!).
  - A general AND-OR formula can be evaluated with $N^{½+o(1)}$ queries.

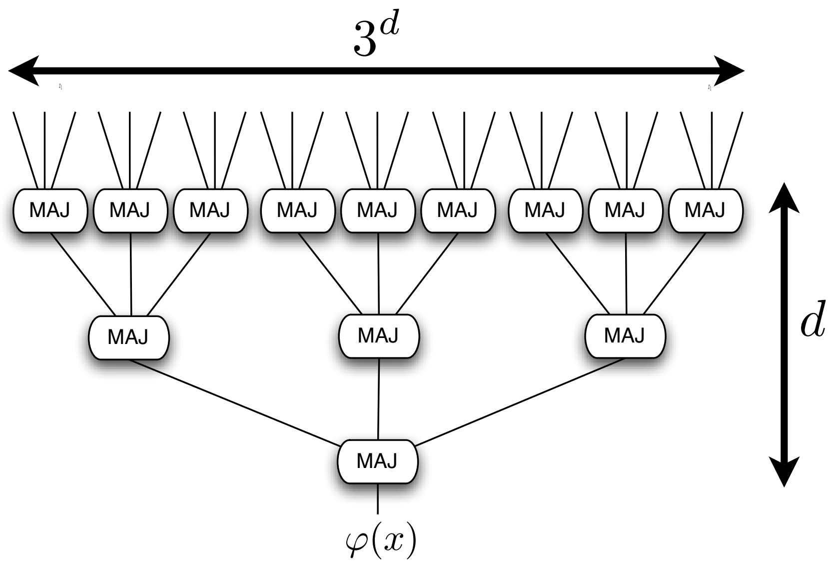
Running time is $N^{½+o(1)}$ in each case, after preprocessing.

[RŠ ‘08] algorithm

- **Theorem:** A balanced (“adversary-bound-balanced”) formula φ over a gate set including all three-bit gates (and more…) can be evaluated in O(ADV(φ)) queries (optimal!).

  (Some gates, e.g., AND, OR, PARITY, can be unbalanced—but not most!)

Recursive 3-bit majority tree



- Best quantum lower bound is
  $\Omega(\mathrm{ADV}(\varphi) = 2^d)$ [LLS'05]

- Expand majority into {AND, OR} gates:
  $$\mathrm{MAJ}_3(x_1, x_2, x_3) = (x_1 \wedge x_2) \vee (x_3 \wedge (x_1 \vee x_2))$$

∴ {AND, OR} formula size is ≤ 5^d

∴ $O(\sqrt{5^d}) = O(2.24^d)$-query algorithm

[ACRSZ '07]

[RŠ '08] algorithm

- **Theorem:** A balanced ("adversary-bound-balanced") formula φ over a gate set including all three-bit gates (and more…) can be evaluated in O(ADV(φ)) queries (optimal!).

  (Some gates, e.g., AND, OR, PARITY, can be unbalanced—but not most!)

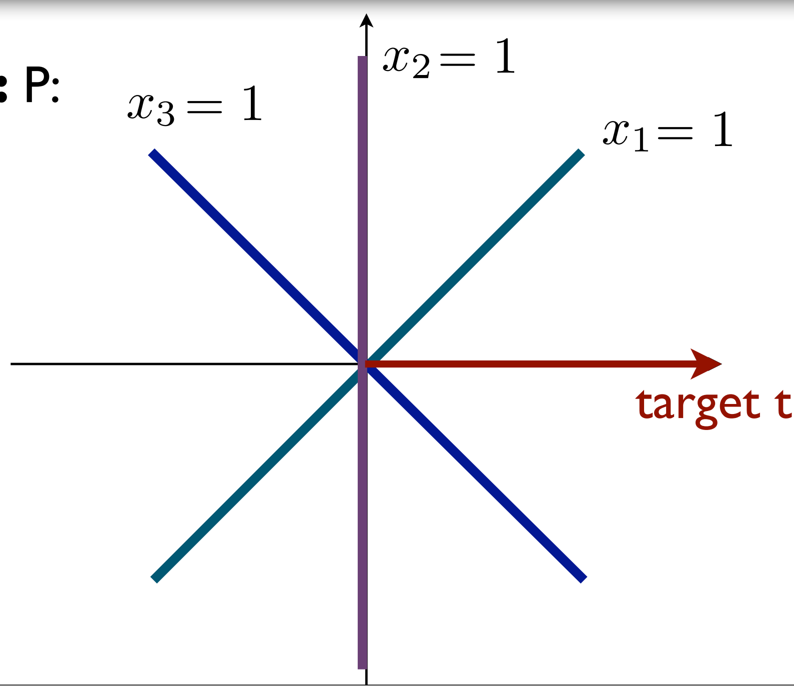
- New: $O(2^d)$-query quantum algorithm

Span program definition

- **Def:** An n-bit span program P is*: [Karchmer, Wigderson '93]
  - A target vector t in vector space V over **C**,
  - n input subspaces, one for each bit

Span program P computes $f_P: \{0,1\}^n \to \{0,1\}$,
$f_P(x) = 1 \Leftrightarrow$ t lies in the span of { subspace i : x_i=1}

- **Ex.:** P:

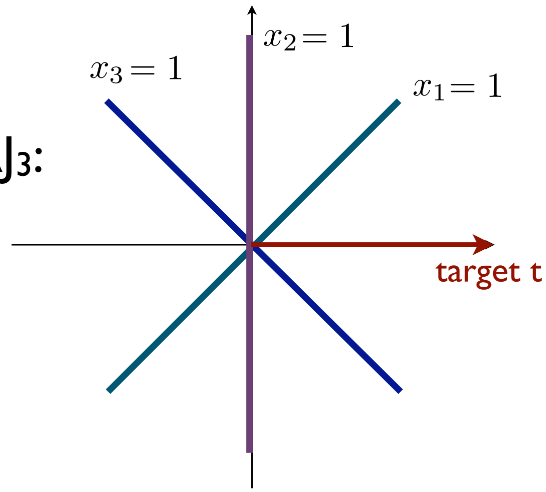


➡ $f_P = MAJ_3$

* *Not* the general def.

Span program P

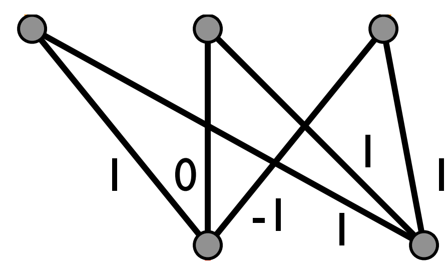
E.g., MAJ3:



Matrix

$$t = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & -1 \\ 1 & 1 & 1 \end{pmatrix}$$

$x_1 = 1$ $\quad x_2 = 1$ $\quad x_3 = 1$



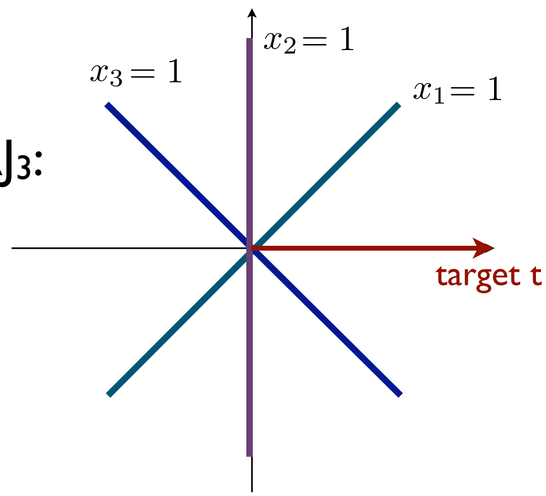
output edge

For a given x, add edges above those inputs evaluating to false.

Weighted bipartite graph

Span program P

E.g., MAJ_3:



Matrix

$$t = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & -1 \\ 1 & 1 & 1 \end{pmatrix}$$

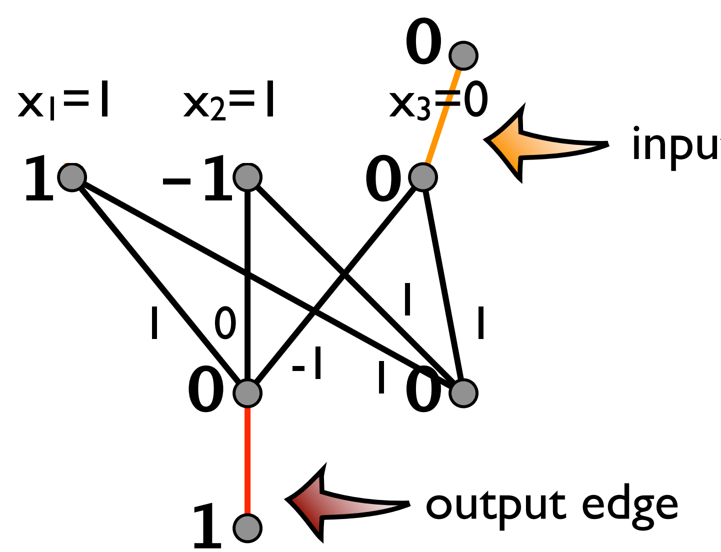
$x_1 = 1$ $\quad x_2 = 1$ $\quad x_3 = 1$

λ=0 eigenvector computes P

For a given x, add edges above those inputs evaluating to false.

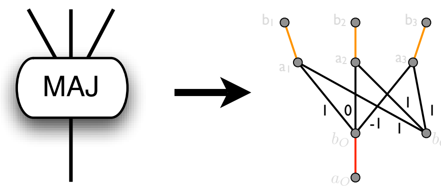
Thm: $f_P(x) = 1 \Leftrightarrow \exists$ eigenvalue-0 eigenvector supported on bottom vertex.

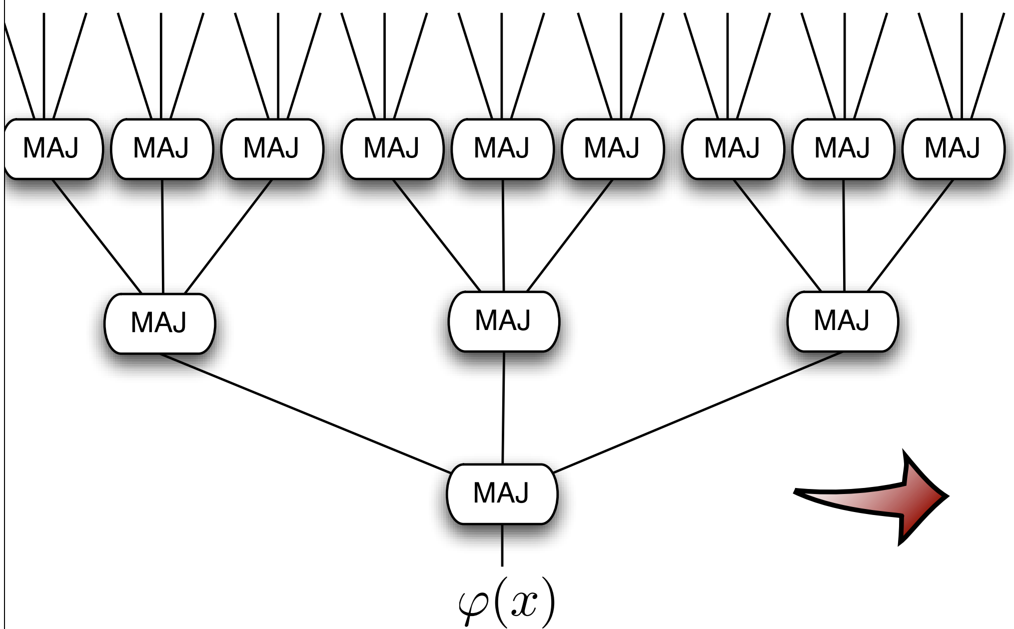
E.g., $x_1 = 1$ $\quad x_2 = 1$ $\quad x_3 = 0$

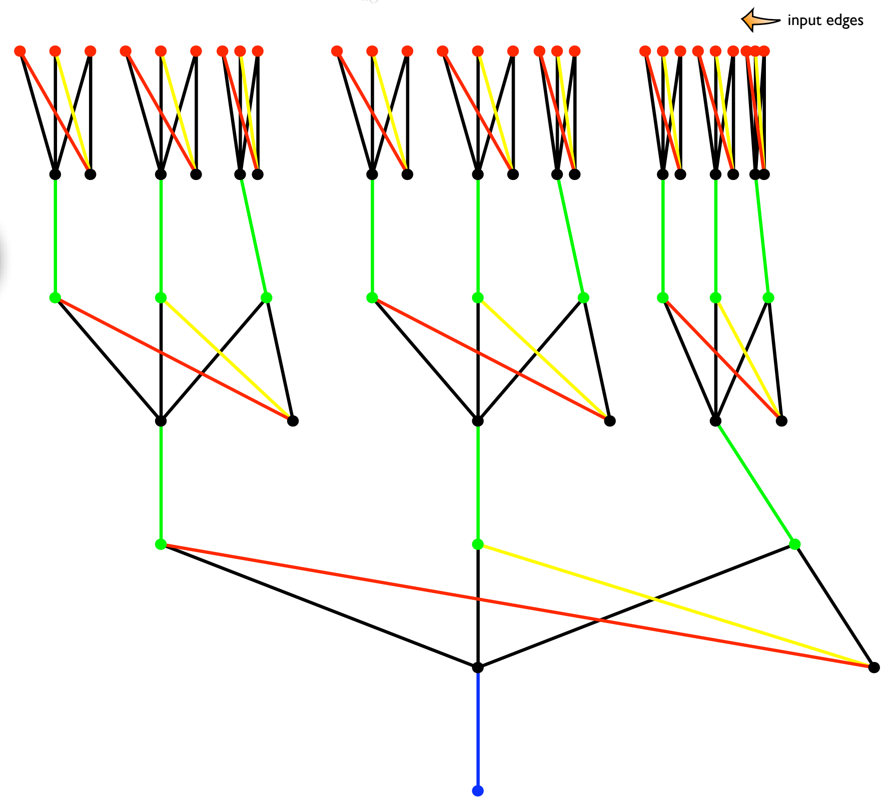


Weighted bipartite graph

Recursive MAJ_3

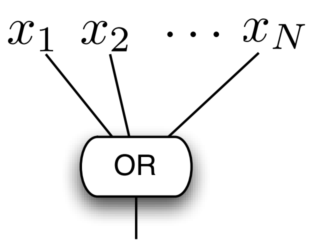
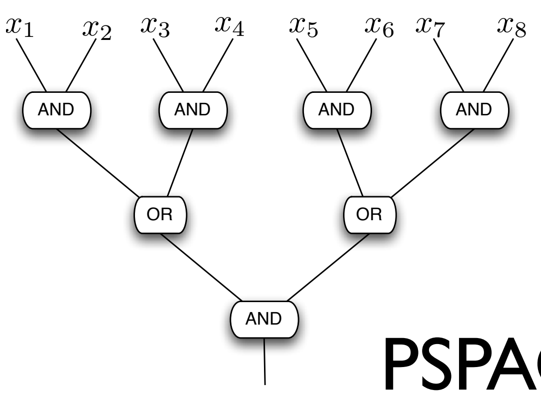
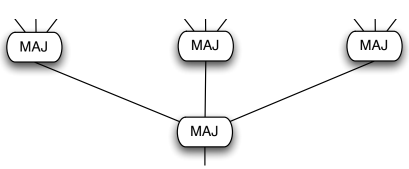






- ## Main Theorem:
  - $\varphi(x)=1 \Rightarrow A_{G(x)}$ has $\lambda=0$ eigenstate with $\Omega(1)$ support on the root.
  - $\varphi(x)=0 \Rightarrow A_{G(x)}$ has no eigenvectors overlapping the root with $|\lambda|<1/2^d$.

$\Rightarrow O(2^d)$-query (optimal!) recursive MAJ_3 evaluation algorithm

	Classical	Quantum
 NP	$\Theta(N)$	$\Theta(\sqrt{N})$ [Grover '96]
Balanced AND-OR  PSPACE	$\Theta(N^{0.753\ldots})$ (fan-in two) [S'85, SW'86, S'95]	$\Theta(\sqrt{N})$ [FGG, AC**R**ŠZ '07]
General read-once AND-OR ⋮	$\Omega(N^{0.51})$ [HW'91] Conj.: $\Omega(D(f)^{0.753\ldots})$ [SW '86]	$\Omega(\sqrt{N})$, $\sqrt{N} \cdot 2^{O(\sqrt{(\log N)})}$ [BS '04] [AC**R**ŠZ '07]
Balanced MAJ3 	$\Omega((7/3)^d)$, $O((2.6537\ldots)^d)$ [JKS '03]	$\Theta(2^d = N^{\log_3 2})$ and much more… [**R**Š '08]

Open:

- Extensions to larger gate sets…
- Unbalaned formulas over more gates…
- Why do span programs work so well? Connection to adversary lower bounds $ADV(f) \leq ADV^{\pm}(f)$?

Open ?: More quantum algorithms based on span programs?

- Our quantum algorithm evaluates span programs. We've applied it by building a large span program by composing small ones for all the gates.
- New framework for developing quantum algorithms: Are there interesting quantum algorithms based directly on large span programs? (E.g., graph problems, Perfect Matching, …) [notion of quantum recursion]

Bird's-eye view of quantum computing