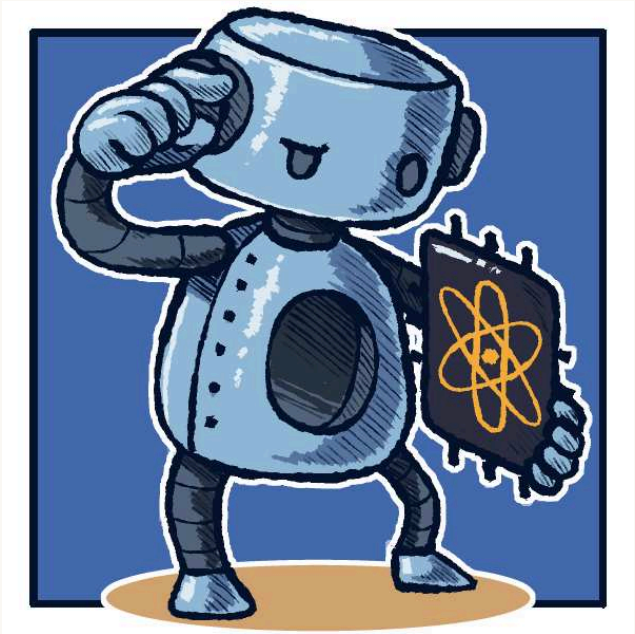
Quantum Oracle Engineering

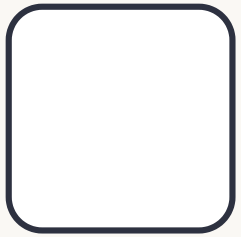
Lesson 1: A different computer



A dice game

First to 100

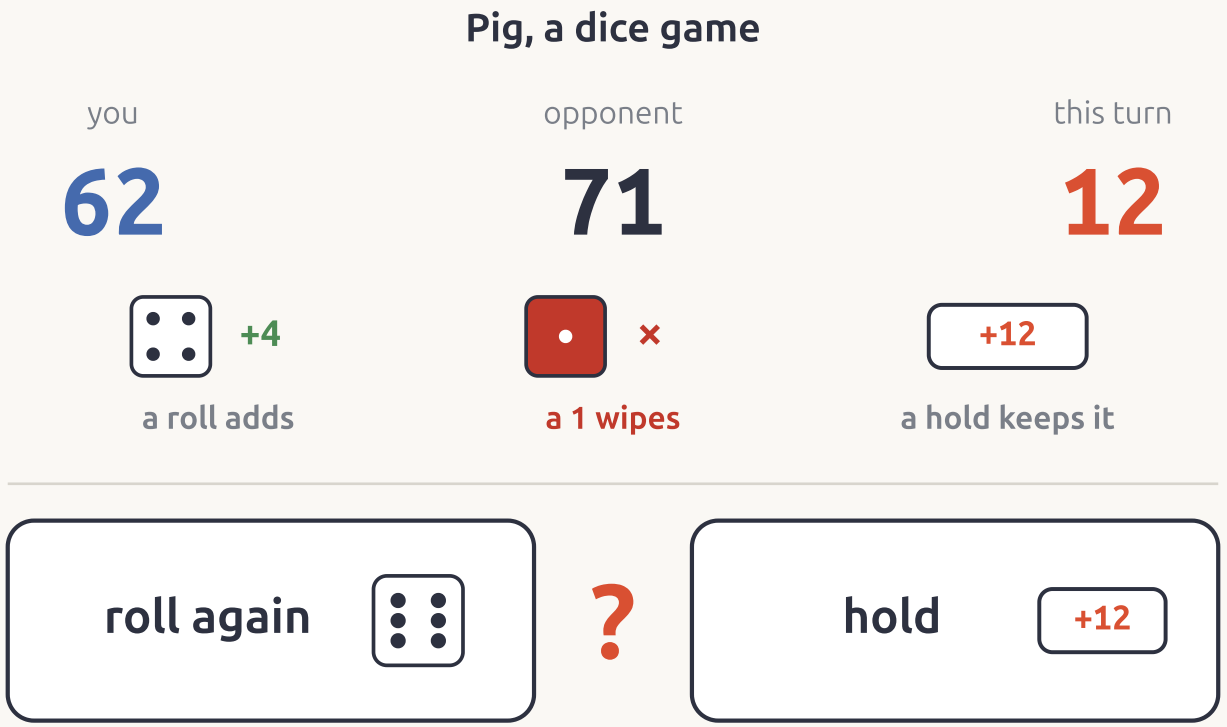
YOU	THEM
0	0



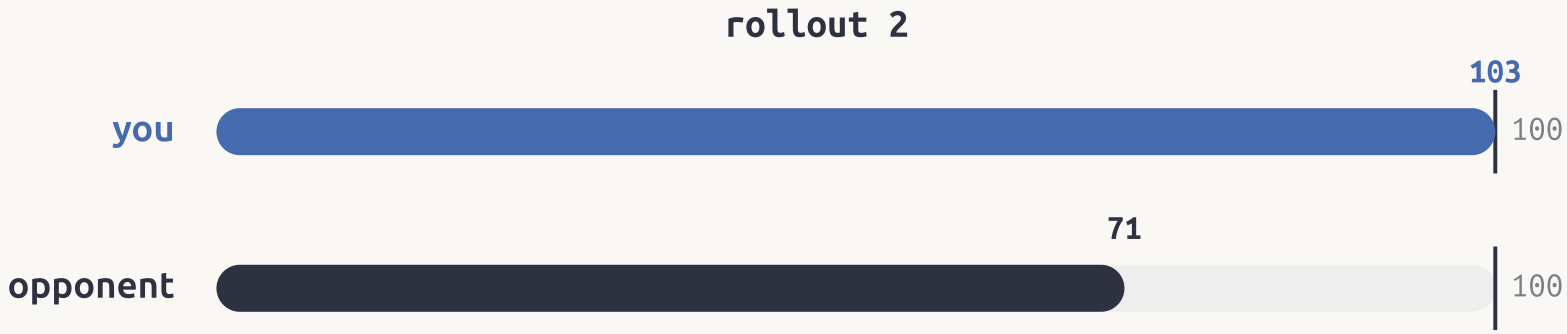
roll hold

A dice game

Which move should we choose?



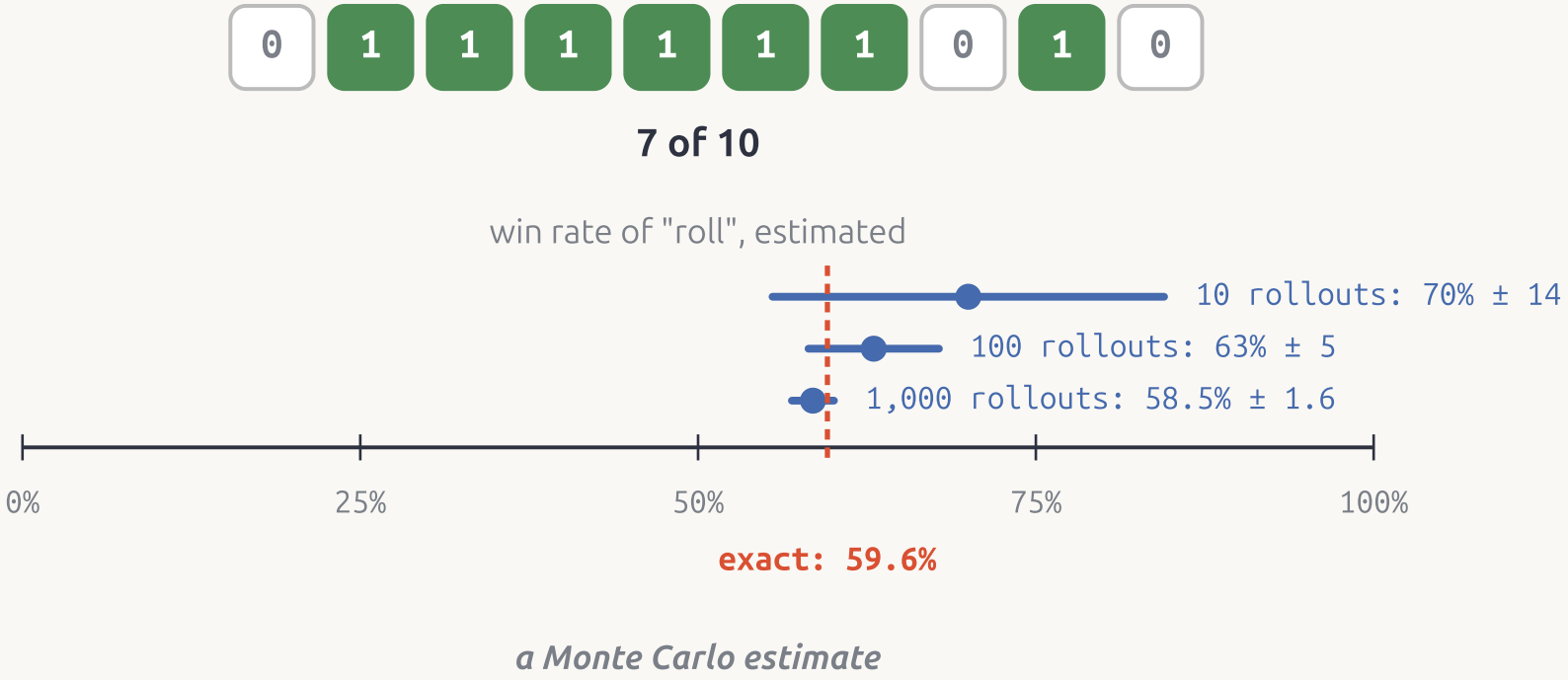
Play one possible future



you hold, +18

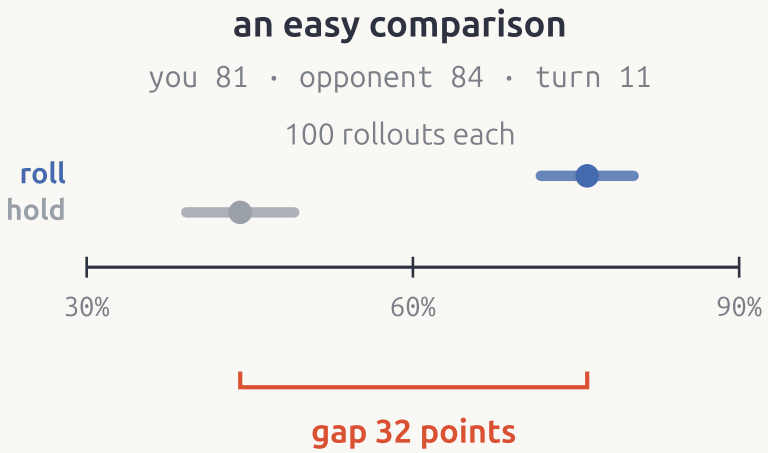
```
win = 1
```

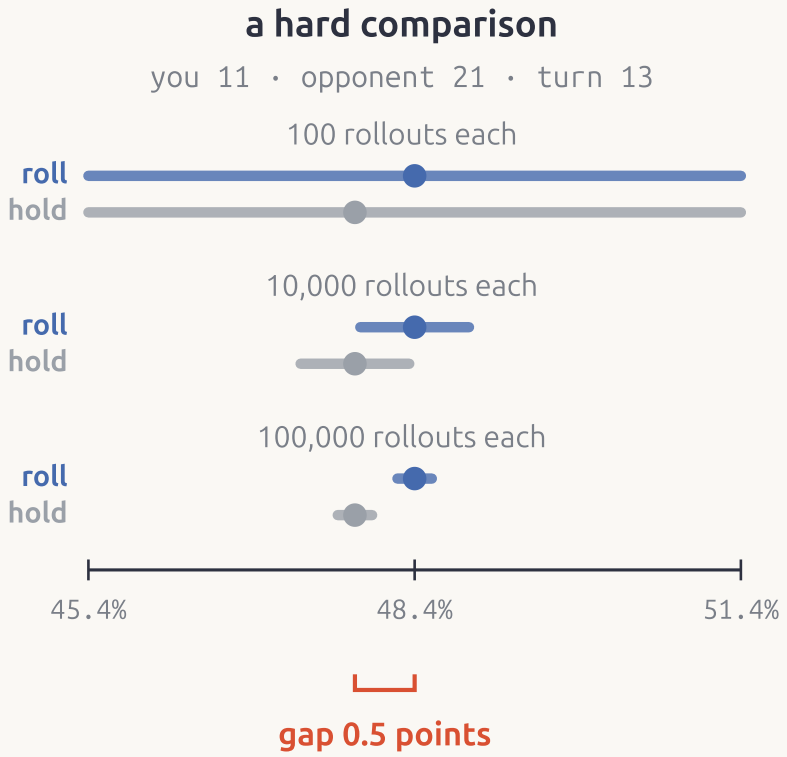
Repeat, then count wins



a Monte Carlo estimate

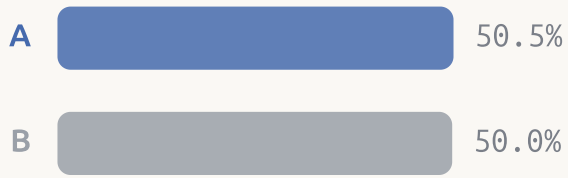
When is the answer clear enough?





Do we need the better move?

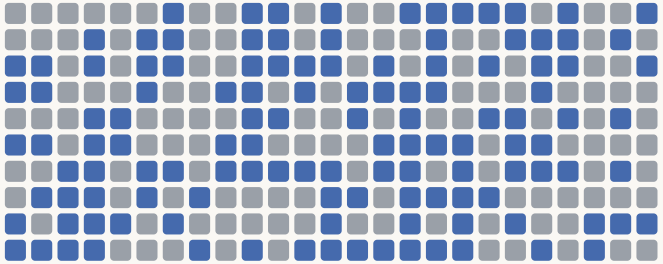
one decision



either move is fine

the worse pick costs half a point, once

a thousand decisions

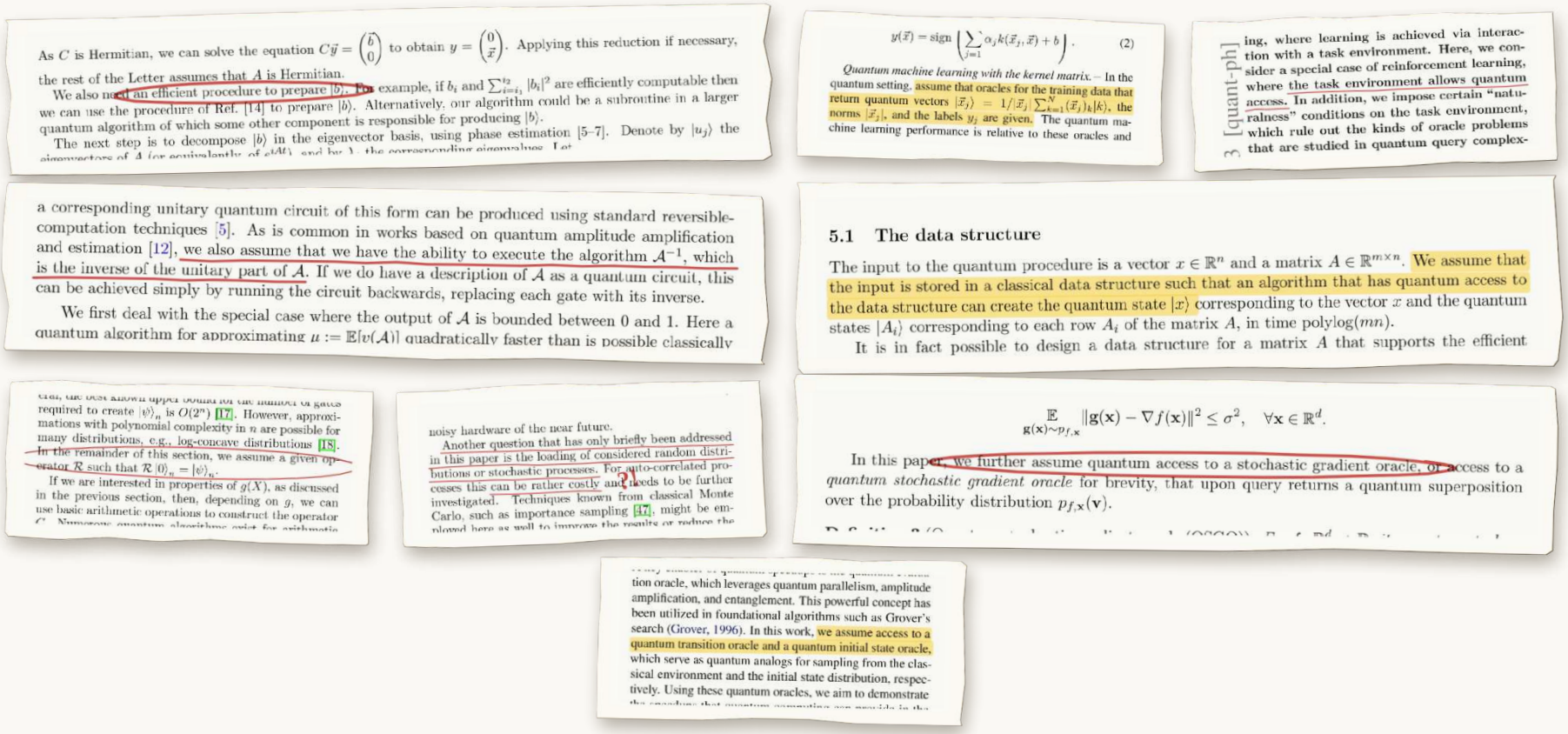


505 to 495, over a thousand games

the gap begins to matter

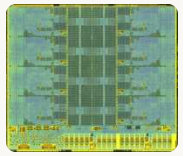
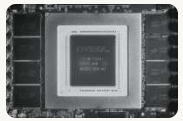
ranking two strategies, tuning a player, a tournament

“Assume oracle access.”

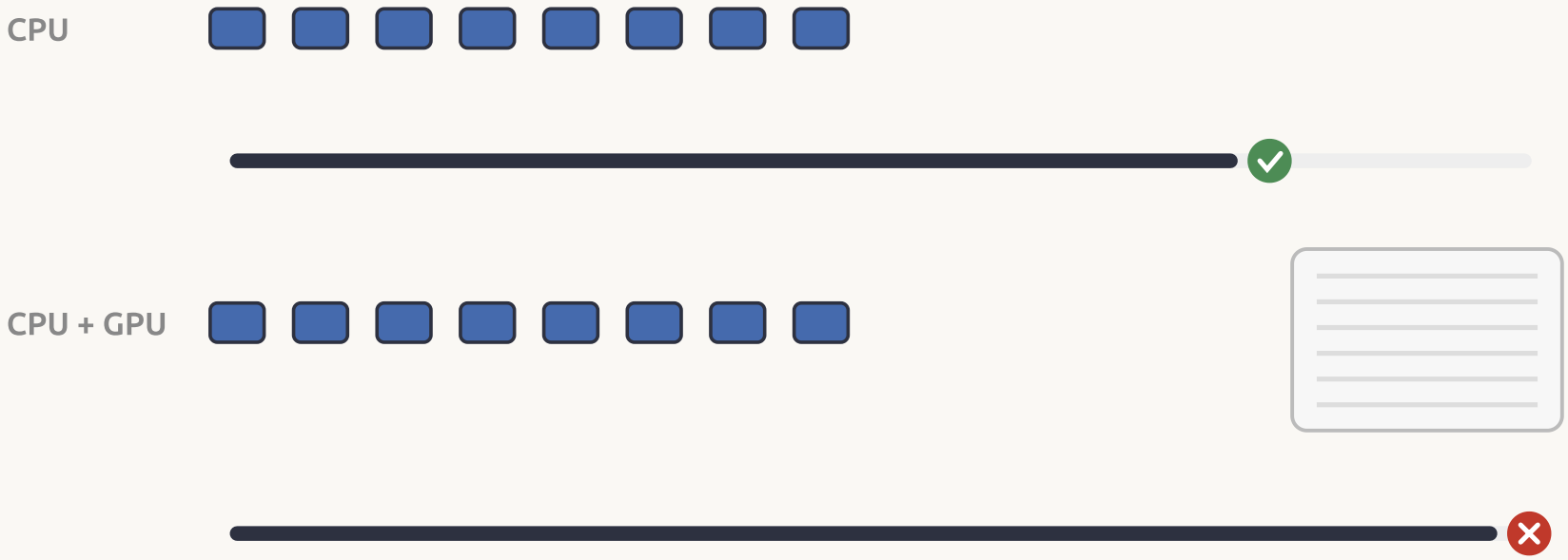


A dice game

Three cost models

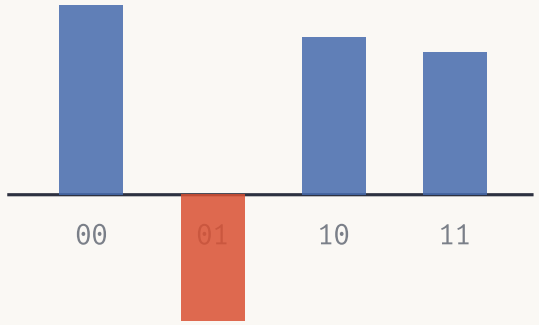
DEVICE	STRENGTH	WORKLOADS
 **CPU**	latency	branching, queries
 **GPU**	throughput	training, graphics
 **QPU**	precision, simulation, factoring	rollouts, expectations

Mismatch costs



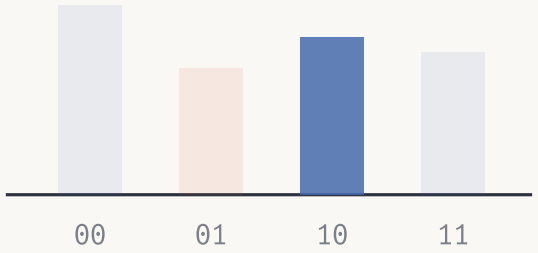
What changes on a quantum computer?

amplitudes



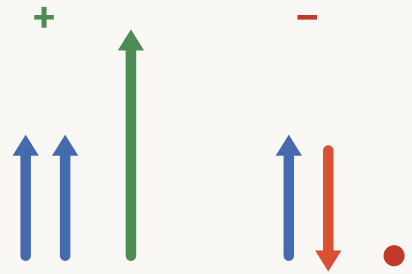
signed, and not yet odds

measure



one outcome: a rollout again

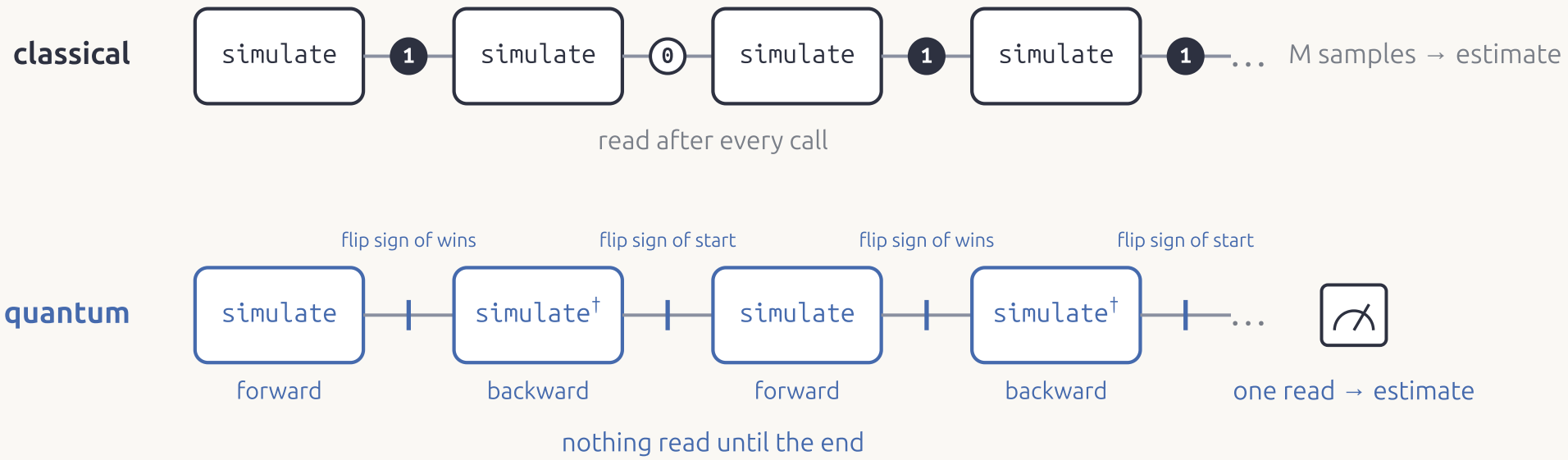
interfere first



agree: add · disagree: cancel

the odds are reshaped before anyone measures

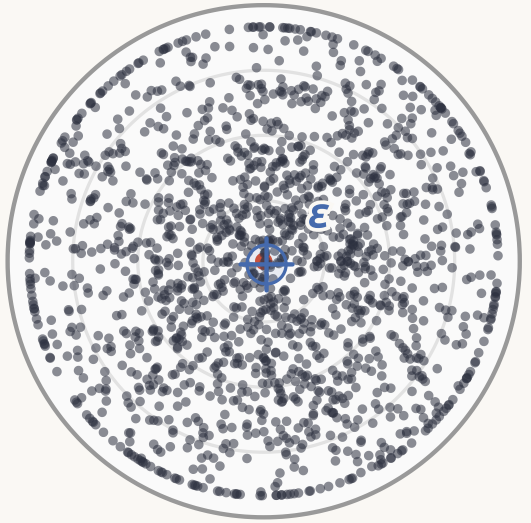
The simulator is the oracle



The classical rate

$$\underbrace{\varepsilon_{\text{classical}}(M) \sim \frac{1}{\sqrt{M}}}_{\text{error after } M \text{ samples}} \quad \Longrightarrow \quad \underbrace{M_{\text{classical}}(g) \sim \frac{1}{g^2}}_{\text{samples to resolve gap } g}$$

ε ÷2

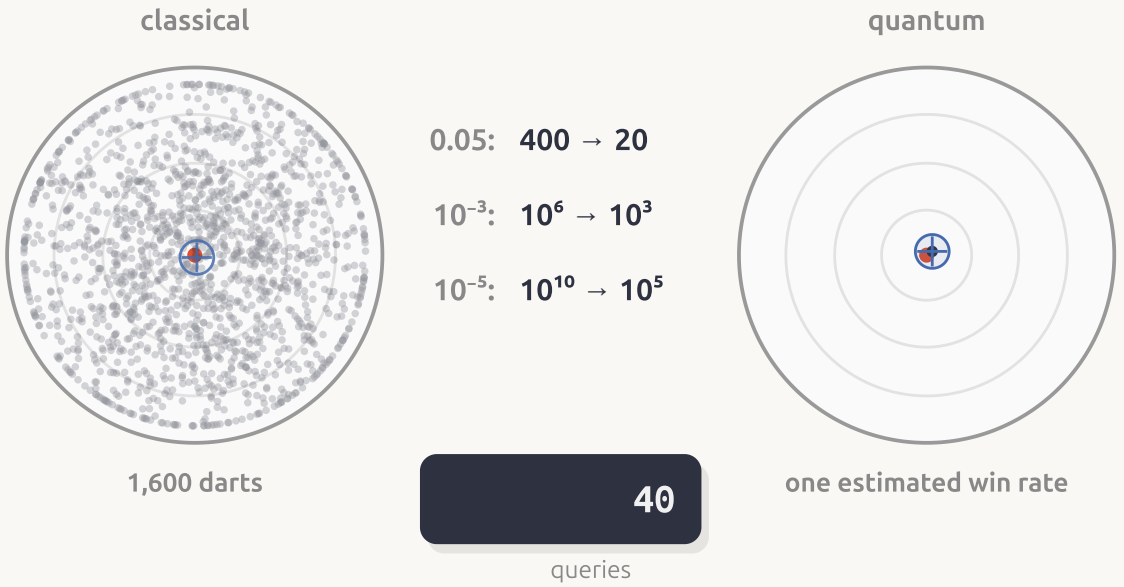


M ×4

M 1,600

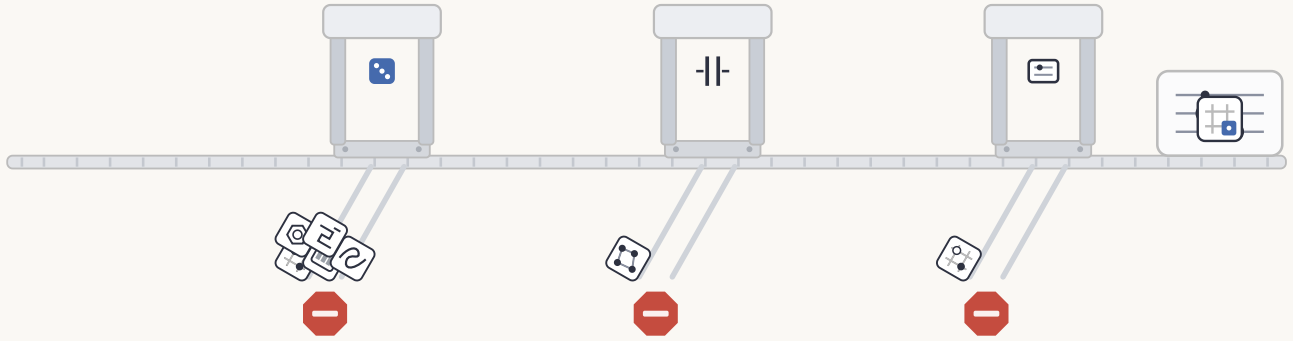
The quantum rate

$$\underbrace{\varepsilon_{\text{quantum}}(M) \sim \frac{1}{M}}_{\text{error after } M \text{ queries}} \quad \Longrightarrow \quad \underbrace{M_{\text{quantum}}(g) \sim \frac{1}{g}}_{\text{queries to resolve gap } g}$$



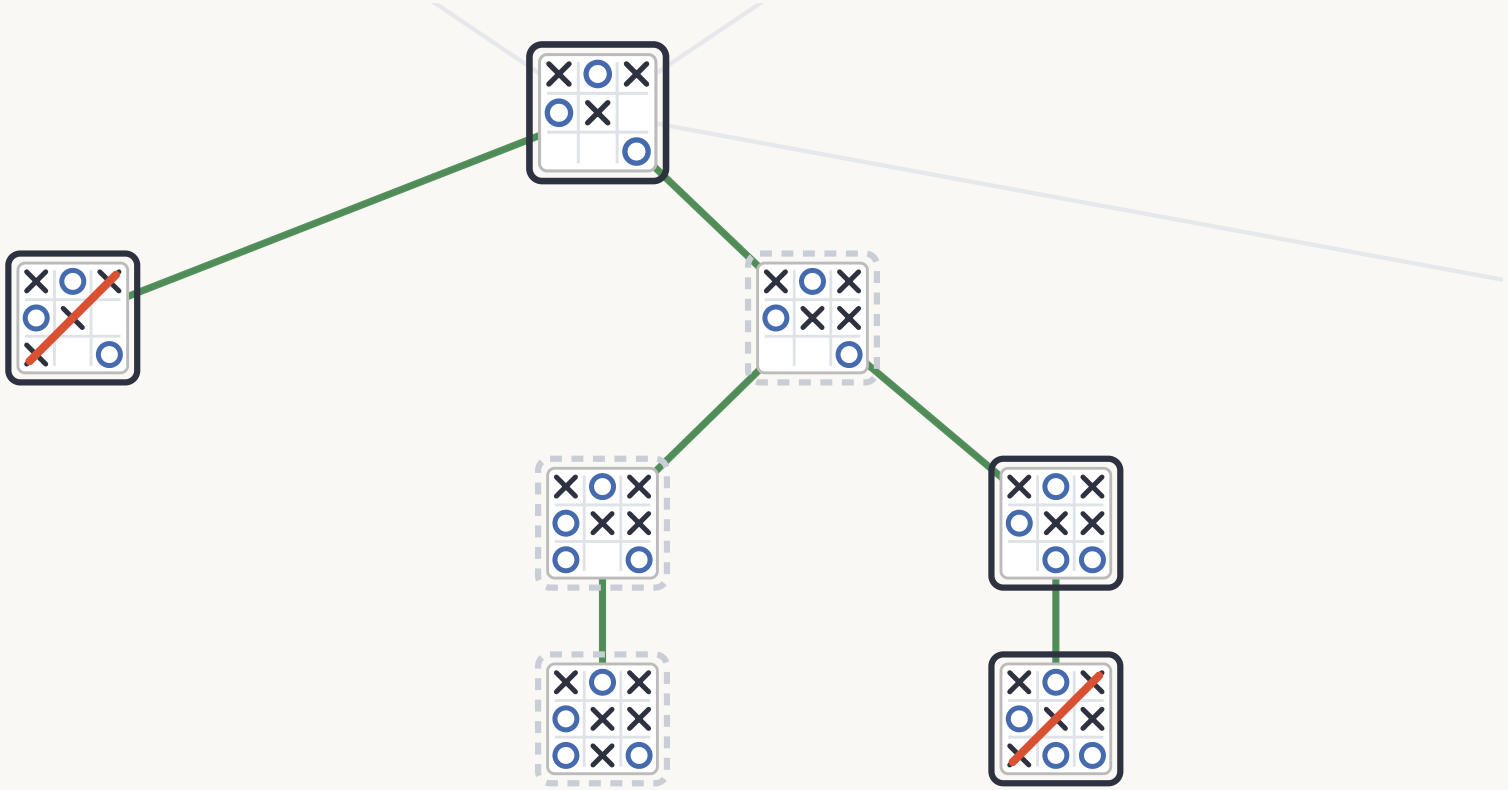
The fine print

Three questions



- does the best classical method still sample?
- is precision the bottleneck?
- is the oracle worth building?

A deterministic world



Pig, solved

roll

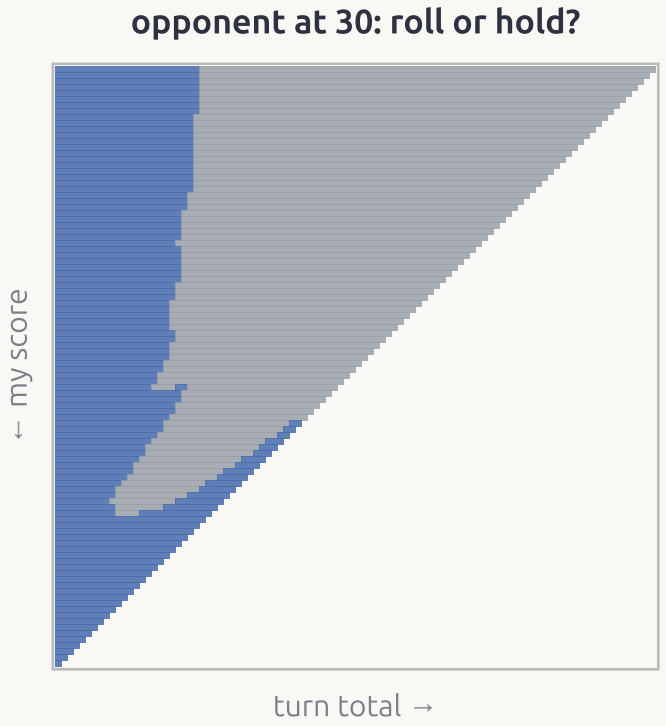
hold

505,000

positions, one table

no rollouts

Neller and Presser, 2004

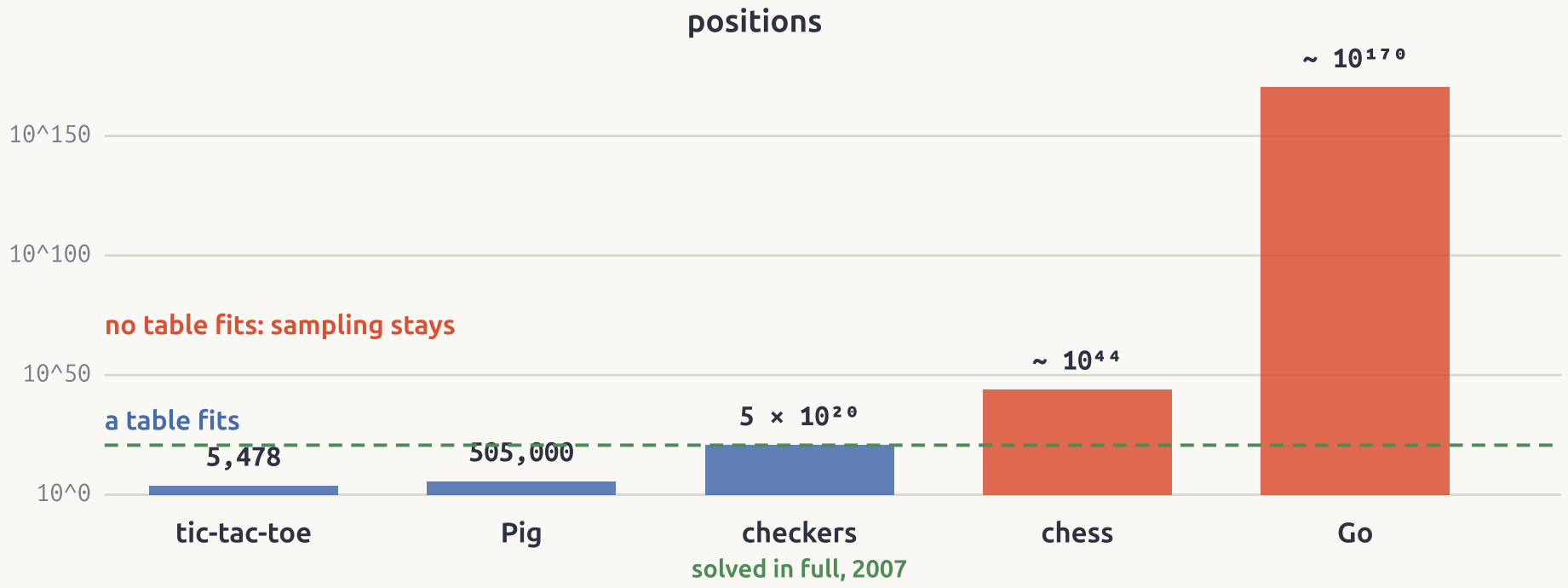


Whose dice?

in the task

in the solver

When sampling stays



The diagnostic

after the strongest classical reformulation

↓

does sampling still dominate?

yes

remaining expectation

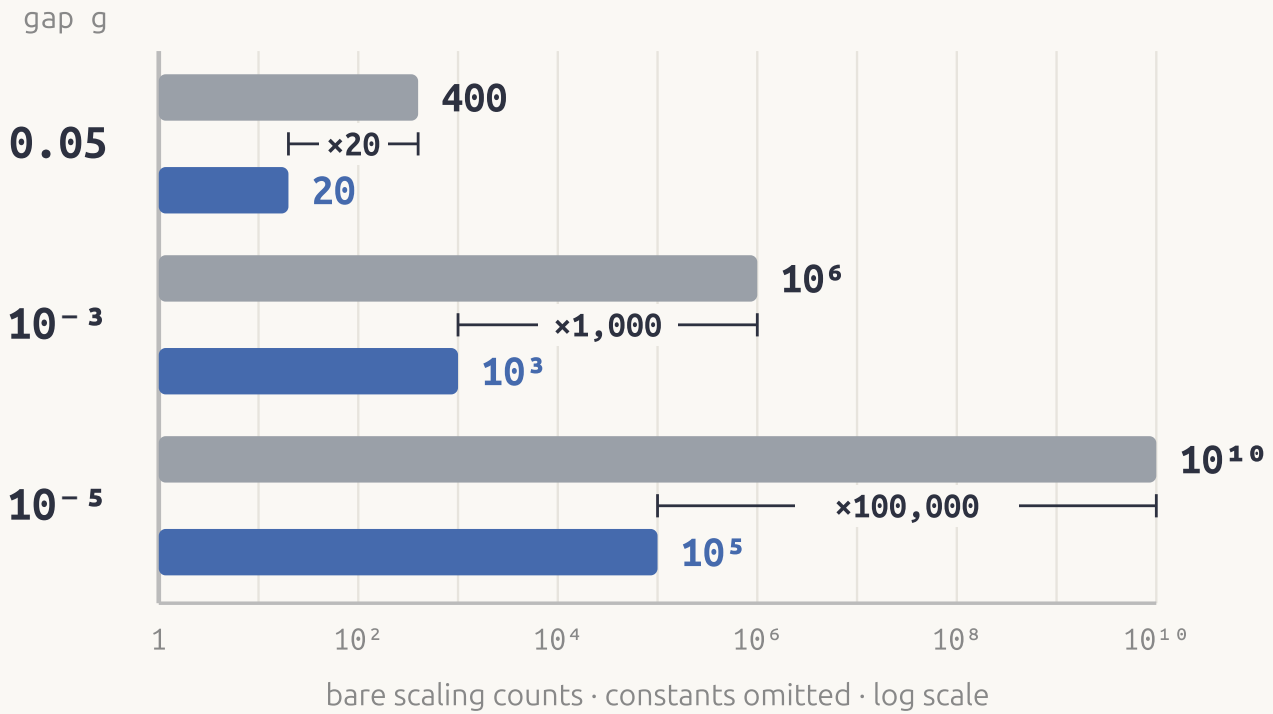
keep going

no

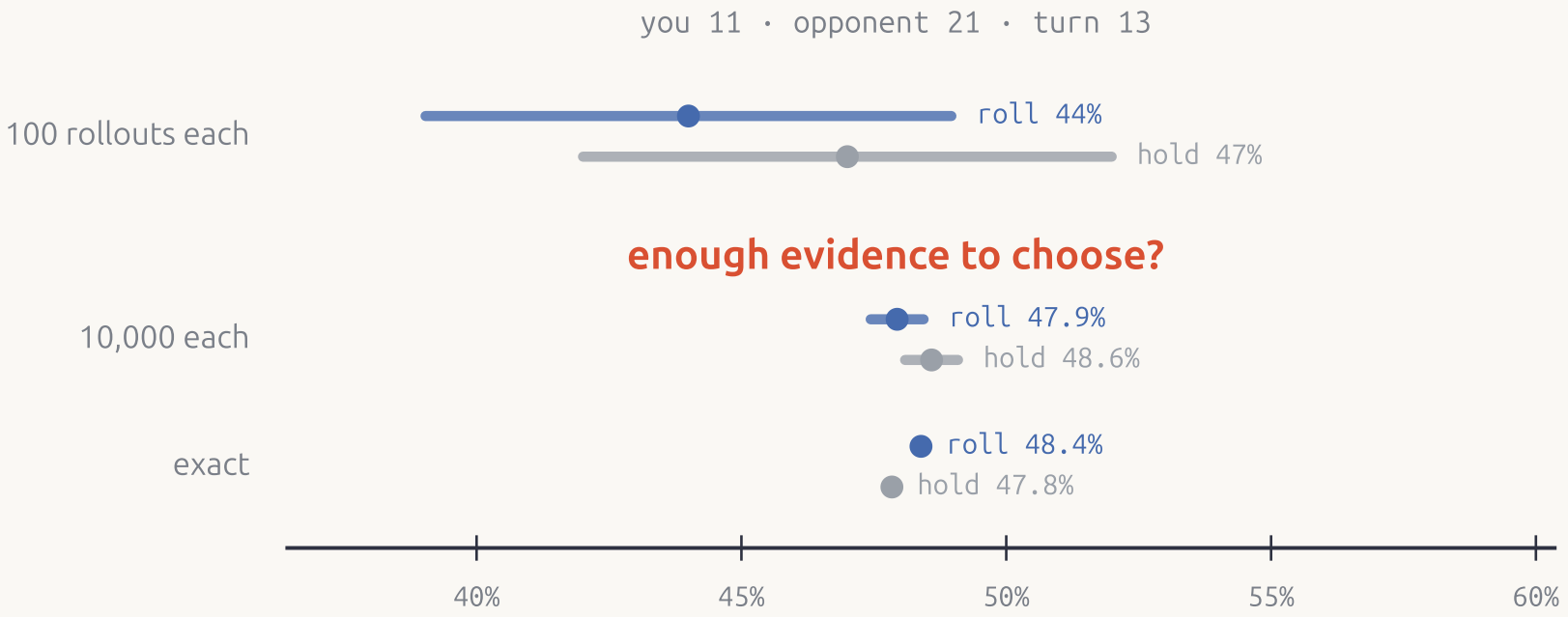
classical structure removes it

stop here

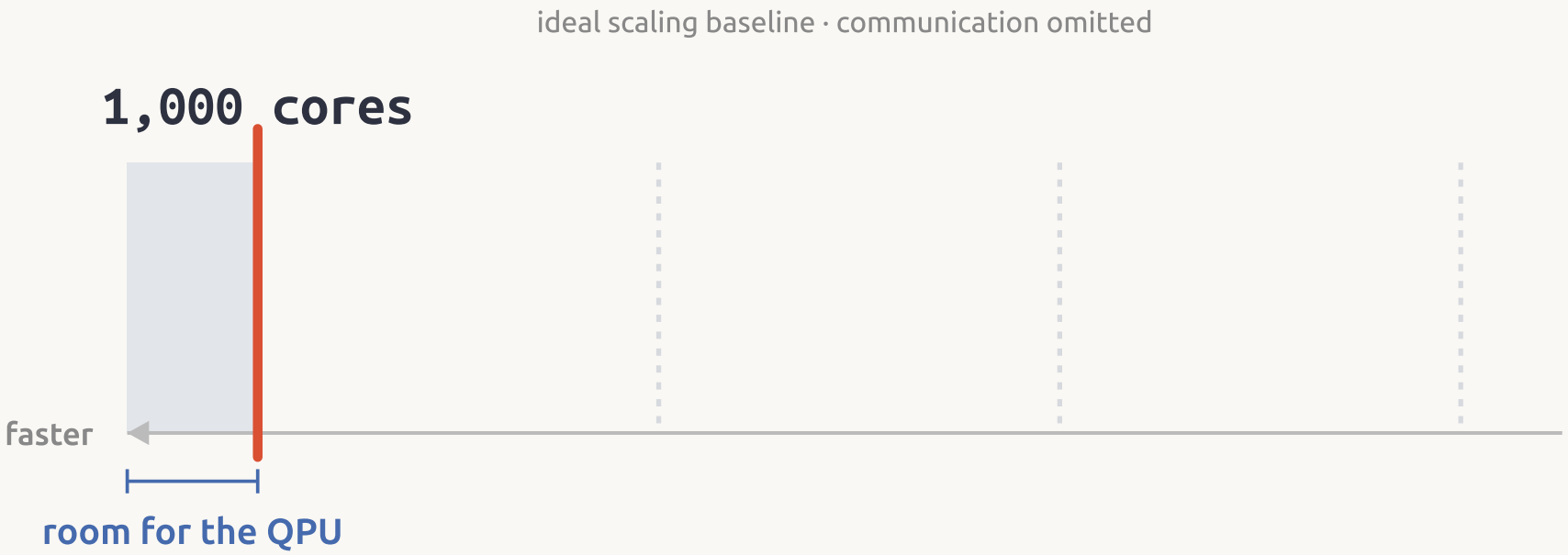
The cost, itemized



Enough evidence to choose?



The classical counterpunch



The second diagnostic

once sampling is the last resort standing

↓

does the gap drive the cost?

yes

tiny gap, samples pile up

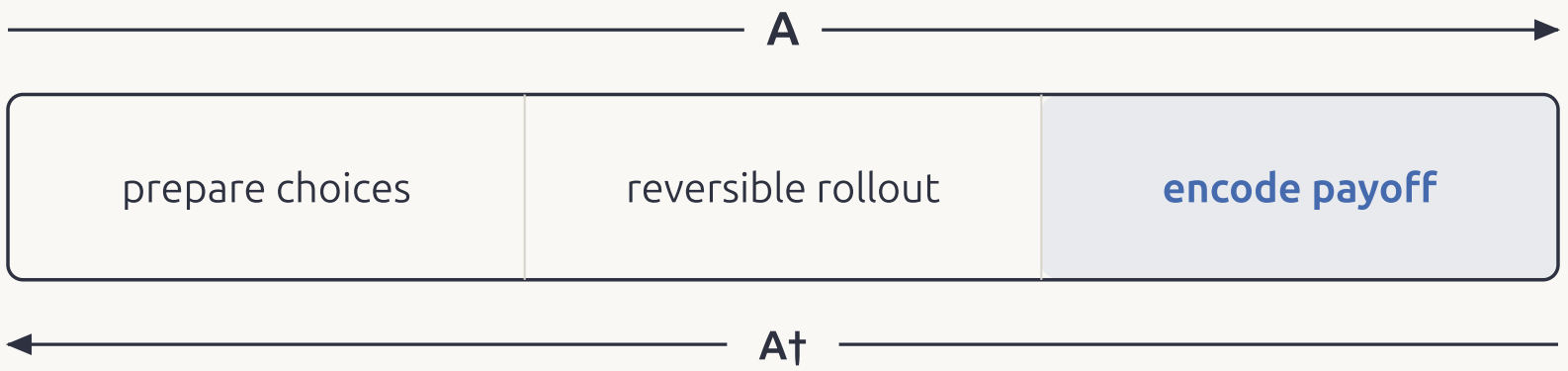
keep going

no

wide gap, rollouts stay few

stop here

The rollout query

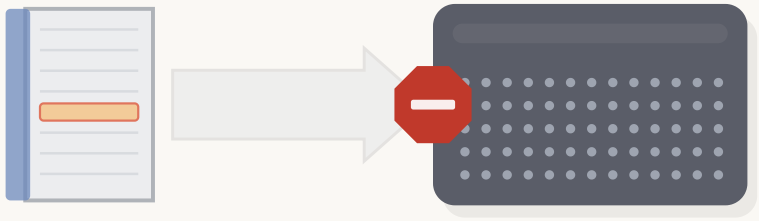


An oracle can be heavy

find one entry in 1,000,000

one-off stored data

1,000,000

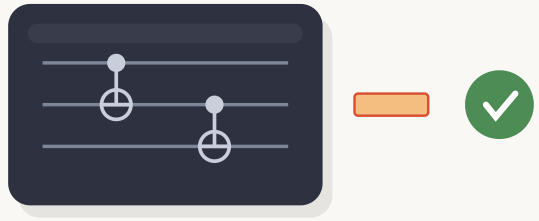


records written into gates

one-off load ≈ classical scan

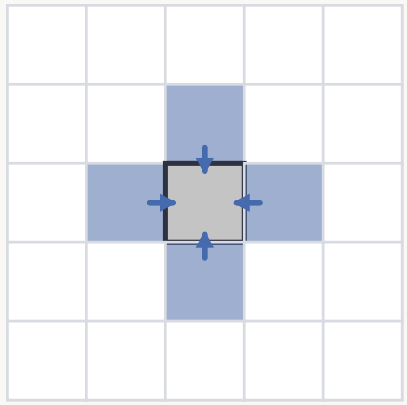
computed predicate

1,000

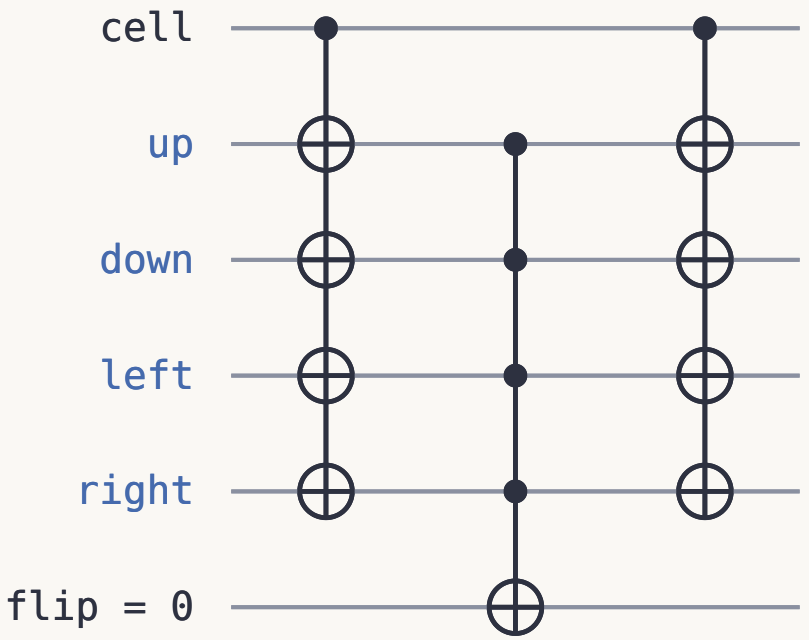


calls to the rule

Rules that compile

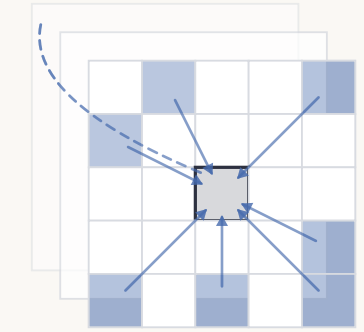


reads the cell and its four neighbours

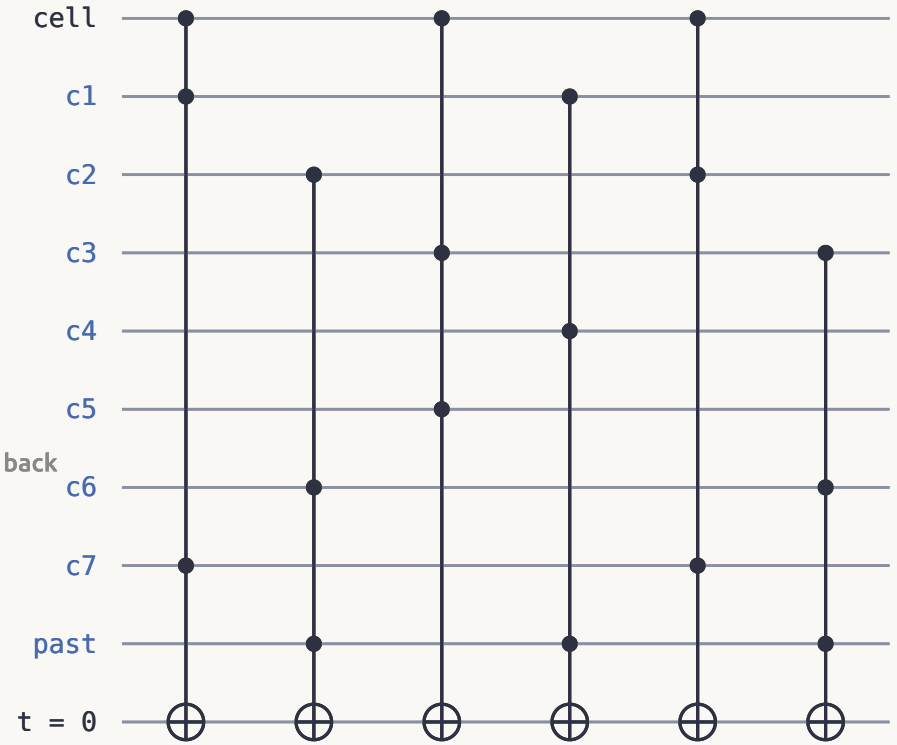


5 reads per cell, at any board size

Rules that tangle



the whole board, and one board back



every cell and the past, per cell

The third diagnostic

with a tiny gap and sampling to close it

↓

is one coherent query cheap?

yes

local rules, nothing loaded

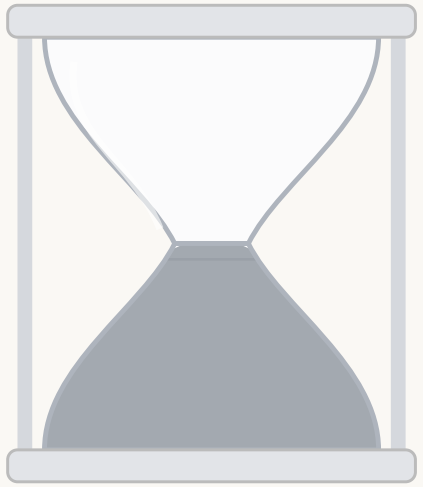
✓ worth costing out

no

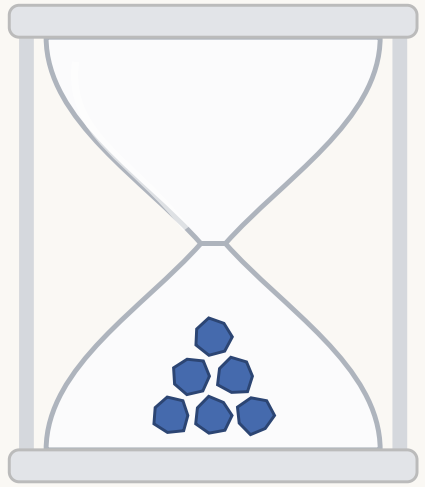
tangled rules or loaded data

⛔ stop here

Winning queries, losing the clock



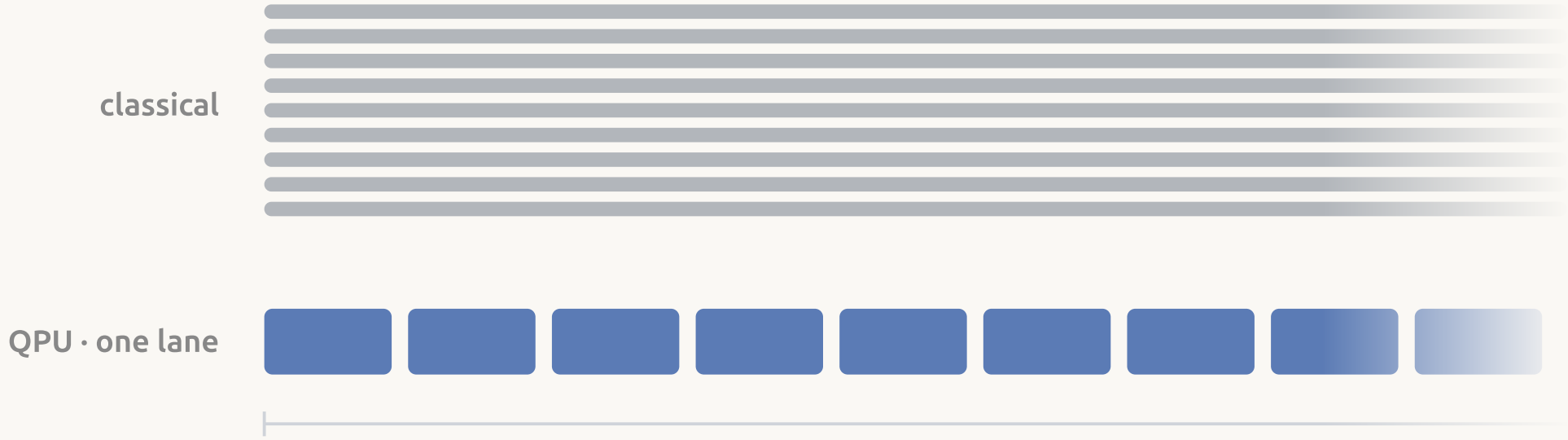
classical · samples



QPU · one query lane

Which machine finishes first?

$$\underbrace{T_{\mathrm{CPU}} \sim \frac{t_{\mathrm{roll}}}{P}\,\frac{1}{g^2}}_{\text{ideal } P\text{-way classical baseline}} \qquad \underbrace{T_{\mathrm{QPU}} \sim C\,t_{\mathrm{oracle}}\,\frac{1}{g}}_{\text{one coherent query lane}}$$



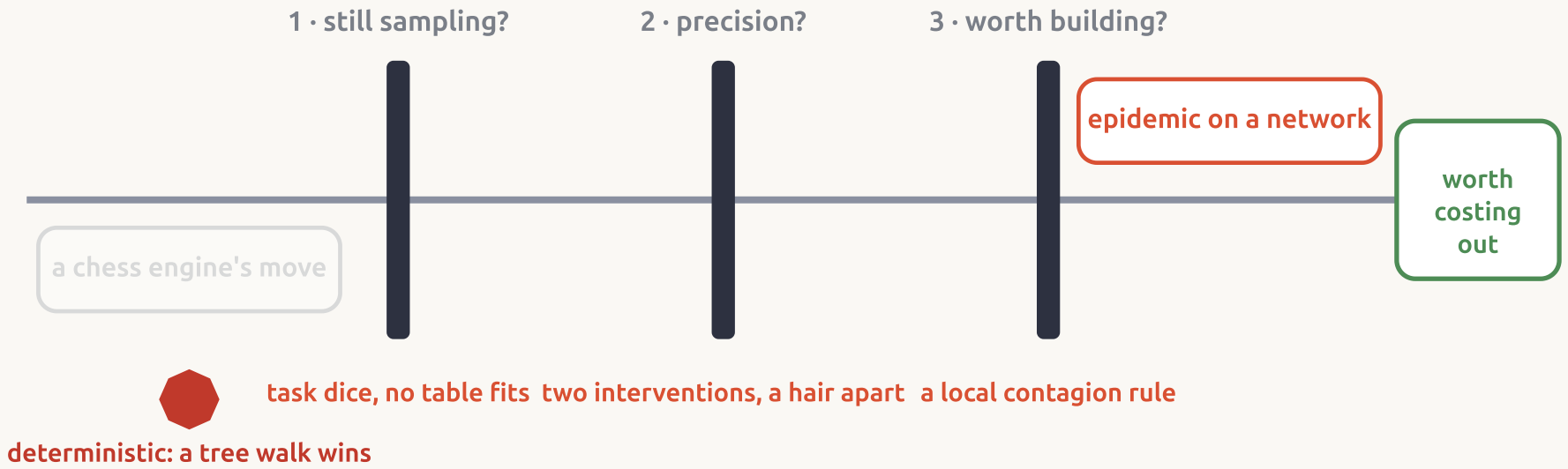
Break-even

$$\underbrace{t_{\text{oracle}}}_{\text{one query}} < \frac{\overbrace{t_{\text{roll}}}^{\text{one rollout}}}{\underbrace{C}_{\text{overhead}} \cdot \underbrace{P}_{\text{cores}} \cdot \underbrace{g}_{\text{gap}}}$$

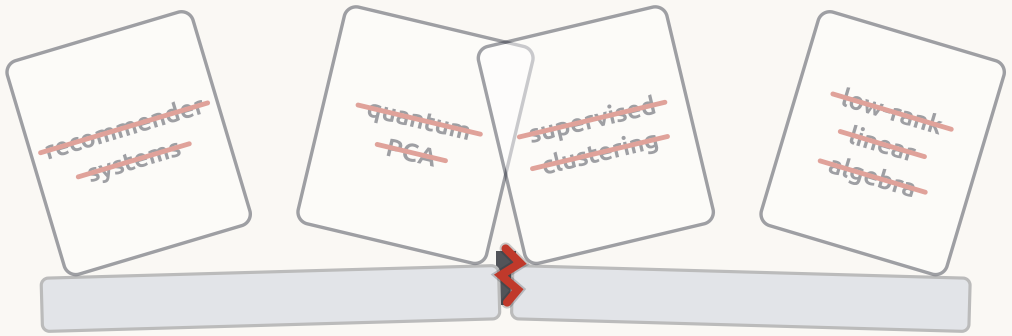
A hundred times faster

$$\underbrace{g = 0.01}_{\text{gap}} \qquad \underbrace{P = 1000}_{\text{cores}} \qquad \underbrace{C = 10}_{\text{overhead}} \qquad \Longrightarrow \qquad t_{\text{oracle}} < \frac{t_{\text{roll}}}{100}$$

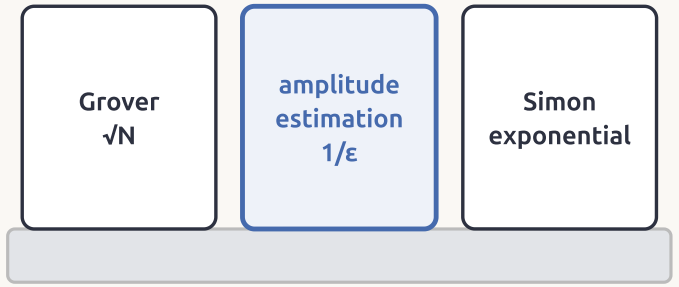
Apply the three questions



The dominoes



2018 · Tang → 2020 · general framework



A different computer

1	2	3–7	8	9–11	12
choose the problem	**prove** the opening	**build** the oracle	**trust** the pieces	**locate** the quantum	**judge** the next claim

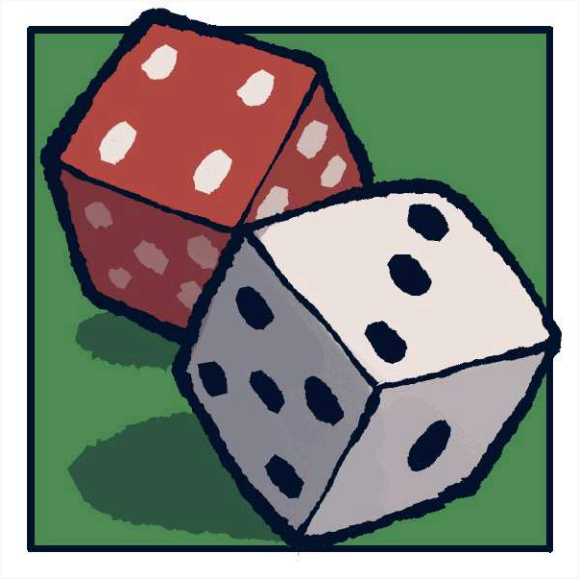
Continue to Lesson 2

shukla.io/quantum-oracle-engineering

Subscribe on the course page for each new lesson

Quantum Oracle Engineering

Lesson 2: The Monte Carlo speedup

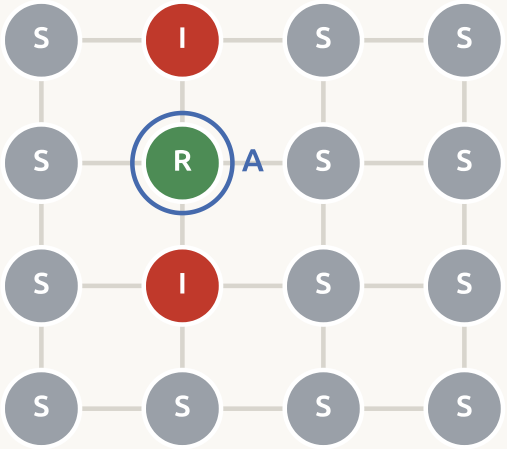


“Monte Carlo gets a quadratic quantum speedup.”

Which intervention should we choose?

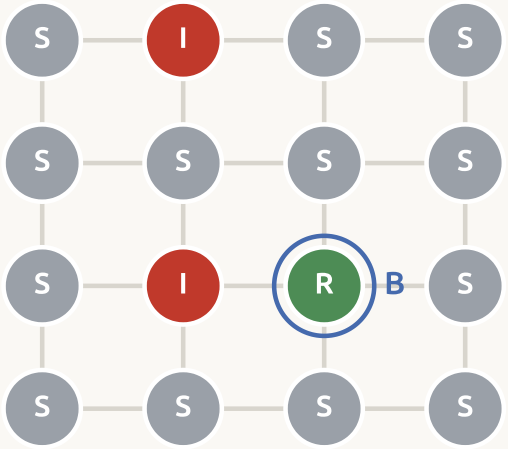
Goal: at most 2 infected after 6 steps.

Vaccinate A first



2 infected

Vaccinate B first



2 infected

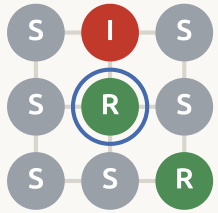
S susceptible I infected R recovered / vaccinated

Next step New future Step 0 / 6

A decision worth simulating

What does the simulator need?

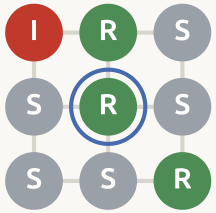
Choose



One candidate action

→

Update



Random events, H steps

→

Score

1 infected

threshold = 2

payoff 1

Objective met? 1 or 0

1 **0** **1** **1** **0** → average = 3/5

Same starting state + action. A fresh future each time.

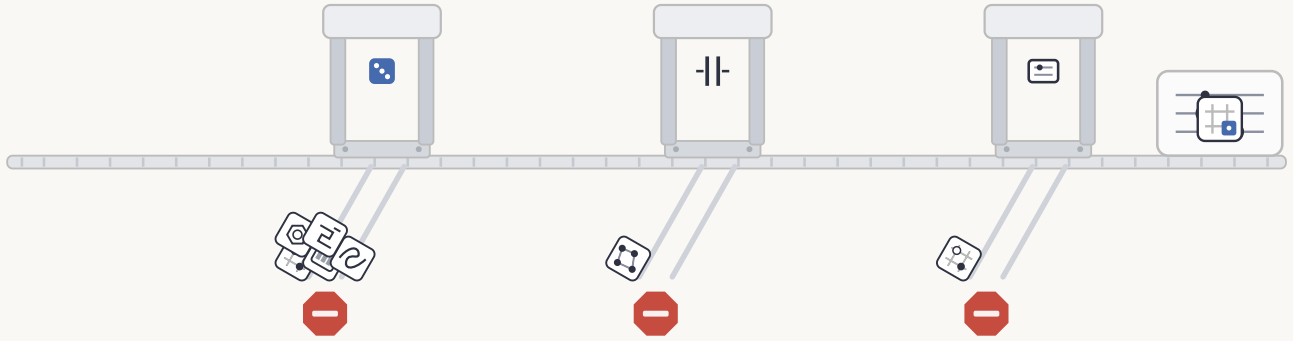
Back

4 / 4 · Repeat and average

Complete

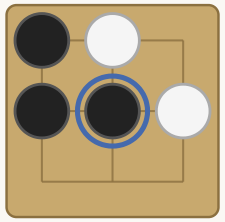
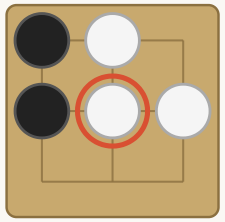
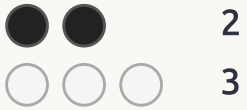
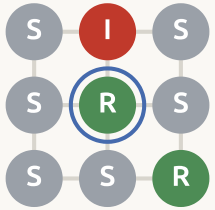
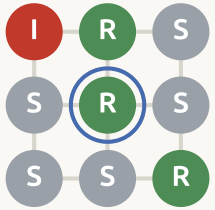
One simulated future is a rollout. Later actions follow a fixed policy.

Three questions, no free passes



- still sampling?
- precision?
- worth building?

We'll build it first in Sway

	Choose	Update	Score
Sway	 Place on an empty cell	 Roll → change color	 2 3 **payoff 0** Black > White → 1
Epidemic	 Vaccinate an S site	 Infect / recover	1 infected threshold = 2 **payoff 1** Infected ≤ threshold → 1

Back

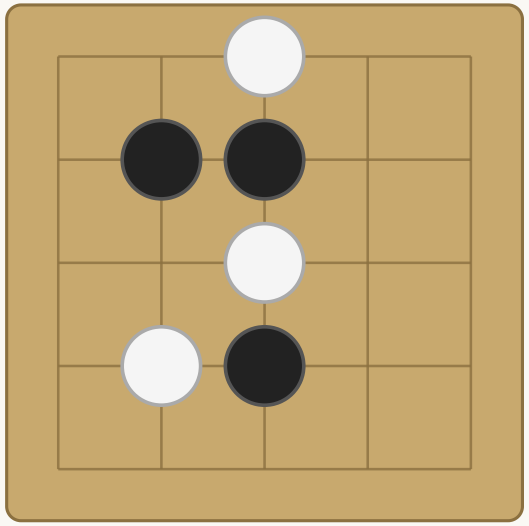
3 / 3 · Evaluate the final state

Complete

One event and illustrative final counts shown. Repeat updates to H before scoring.

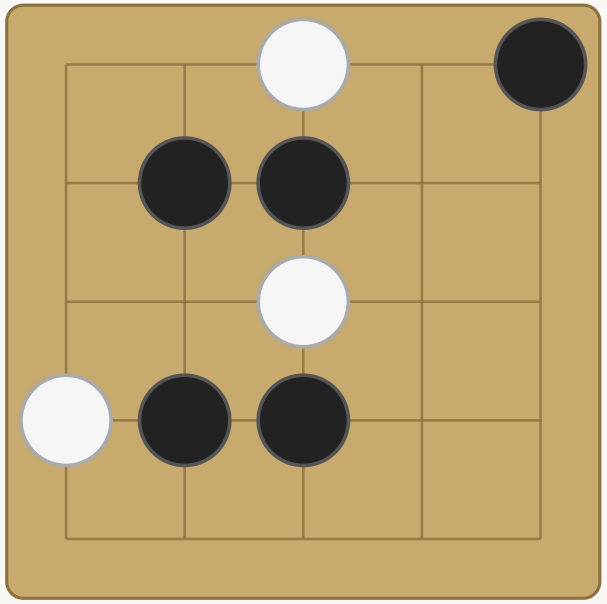
The template for most problems

Sway



5×5 · Black to move · 3 rounds left

Black, White, then the board

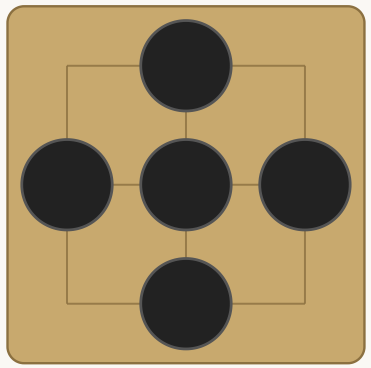


round 2 of 3

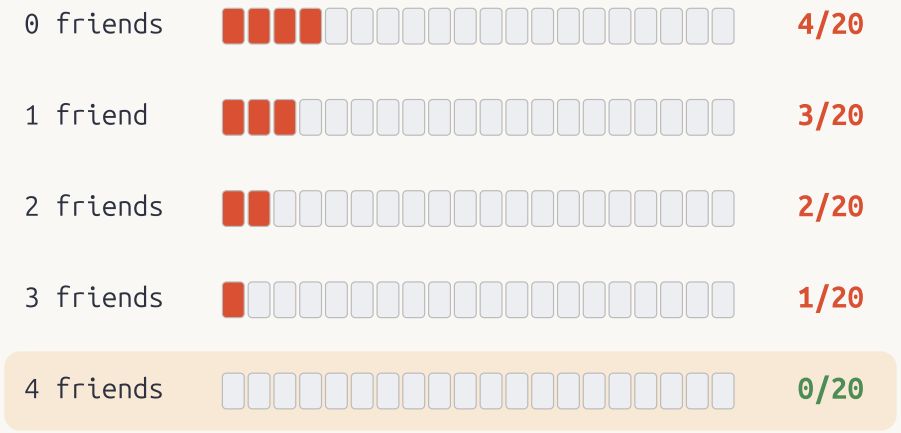
1. Black places
2. White places
3. **Sway!**

the flips land together; on to round 2

Friends make a stone harder to sway

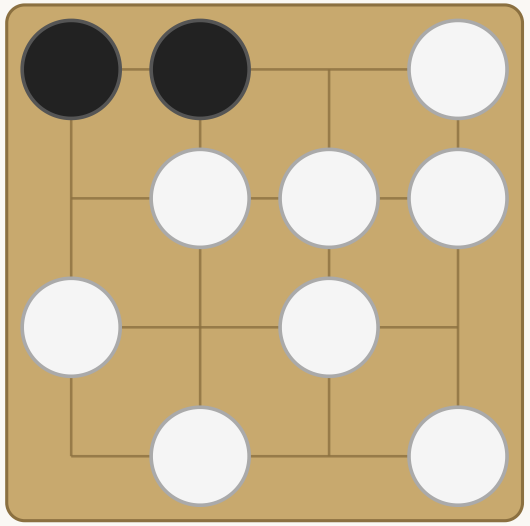


corner: ≤ 2 friends



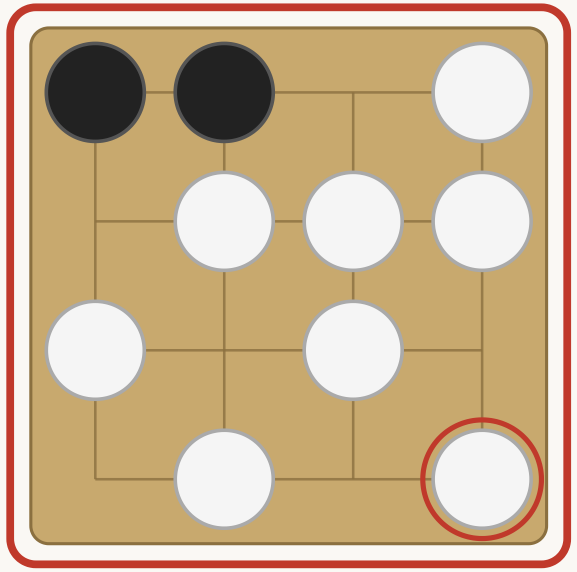
$$\Pr[\text{flip}] = \frac{4 - c}{20}$$

Everyone decides from the old board



synchronous

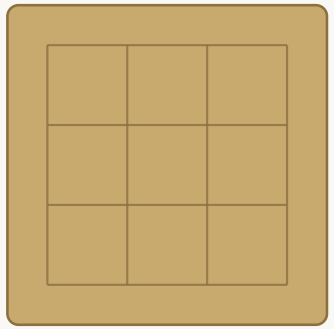
same dice



sequential

How do we score a rollout?

Black ahead

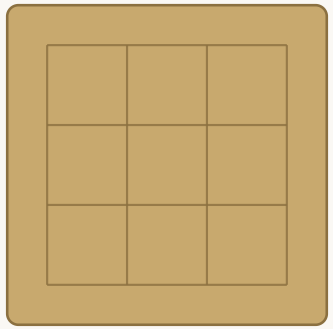


6

4

payoff 1

White ahead

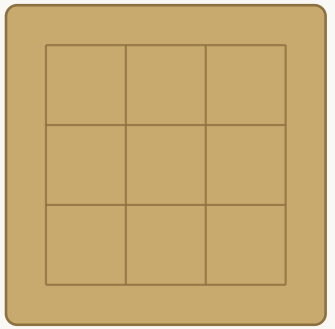


4

6

payoff 0

A tie



5

5

payoff 0

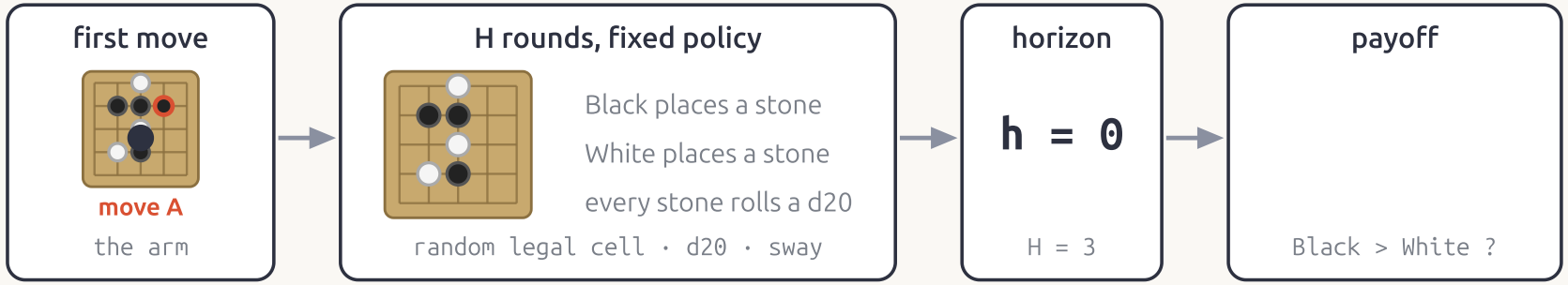
Back

4 / 4 · Reveal the tie

Complete

Our objective: win probability. Winning by 1 or 10 both return 1.

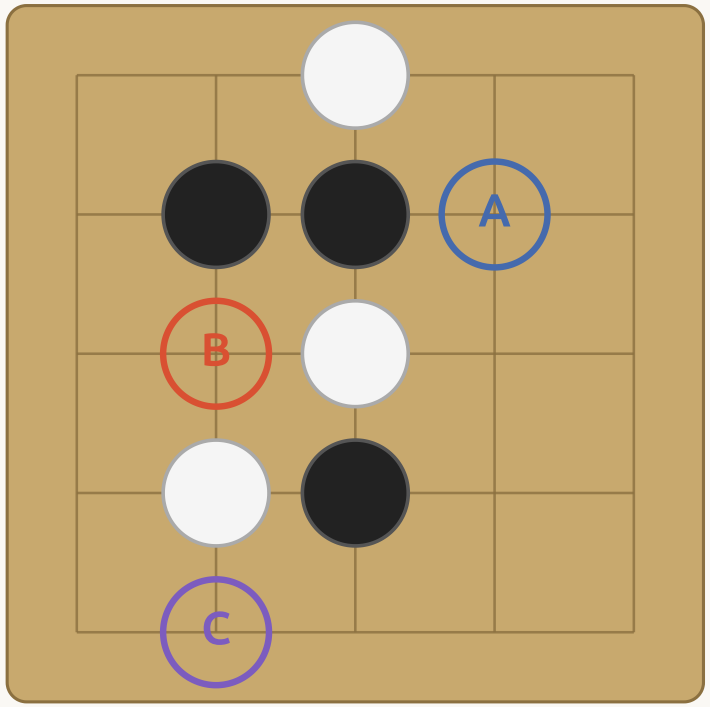
Follow one Sway rollout



running average: –

$$v_i = \Pr\left[\text{Black outnumbers White at } H \,\middle|\, \text{first move } i,\ \text{fixed policy}\right]$$

Where would you play?



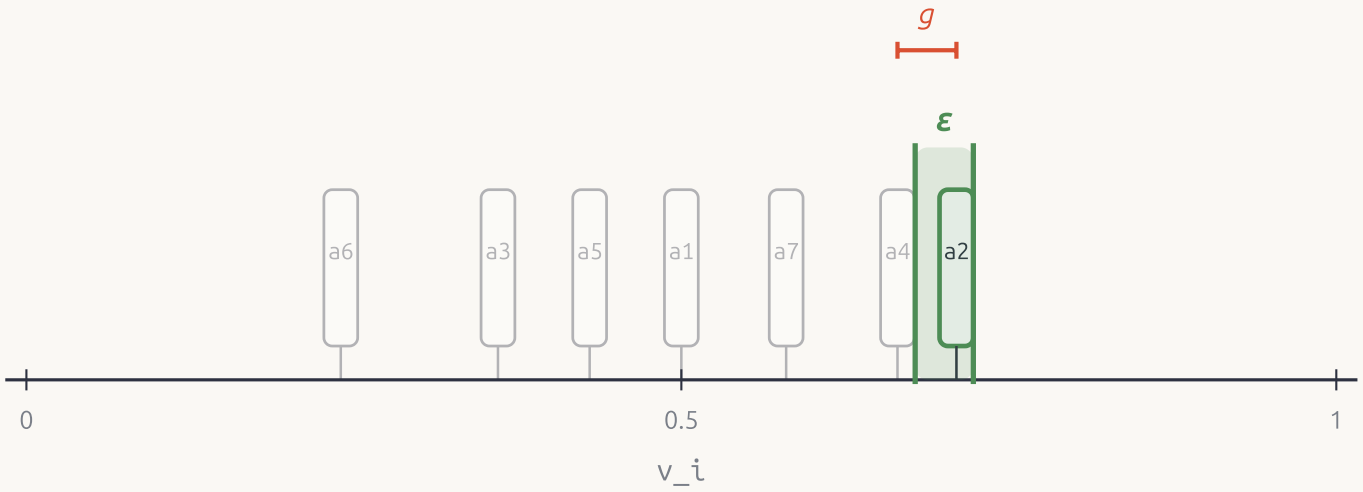
A — 1 friend — 0.449

B — 1 friend — 0.454

C — 0 friends — 0.413

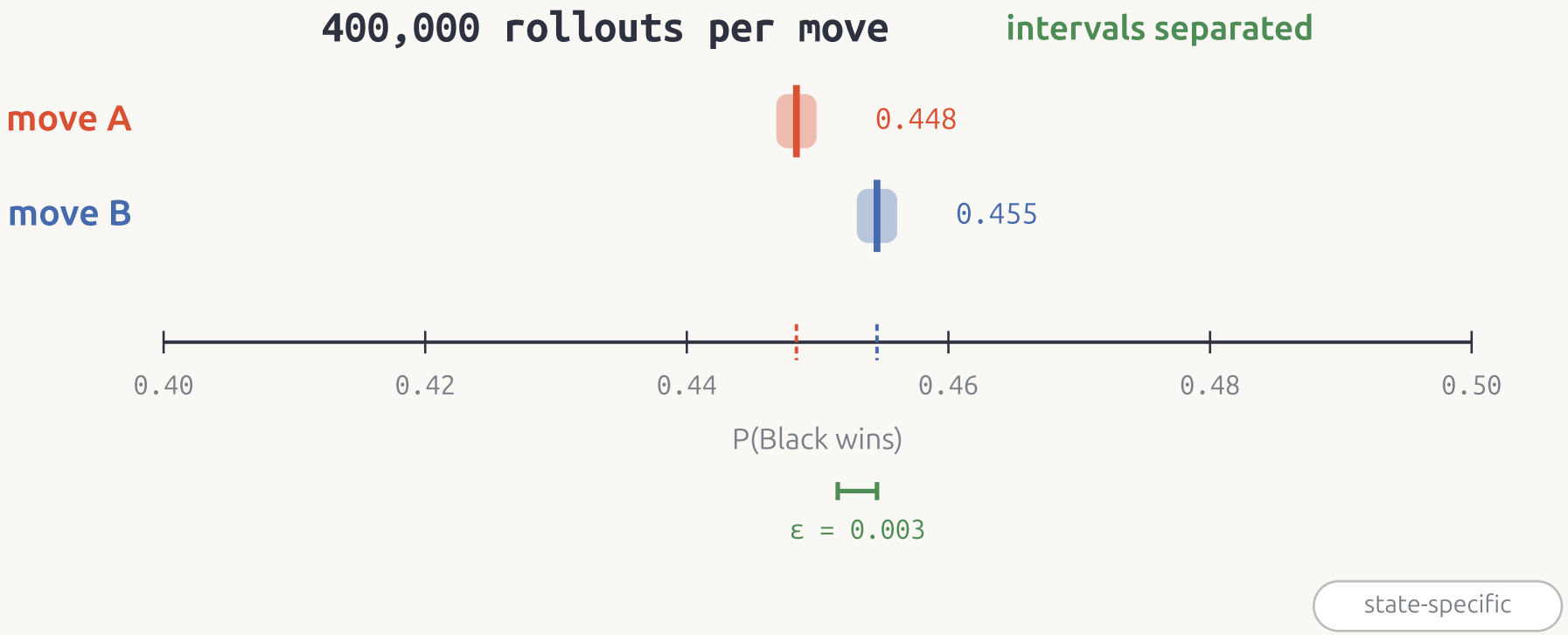
40,000 rollouts / move · 95% intervals

How good does the chosen move need to be?



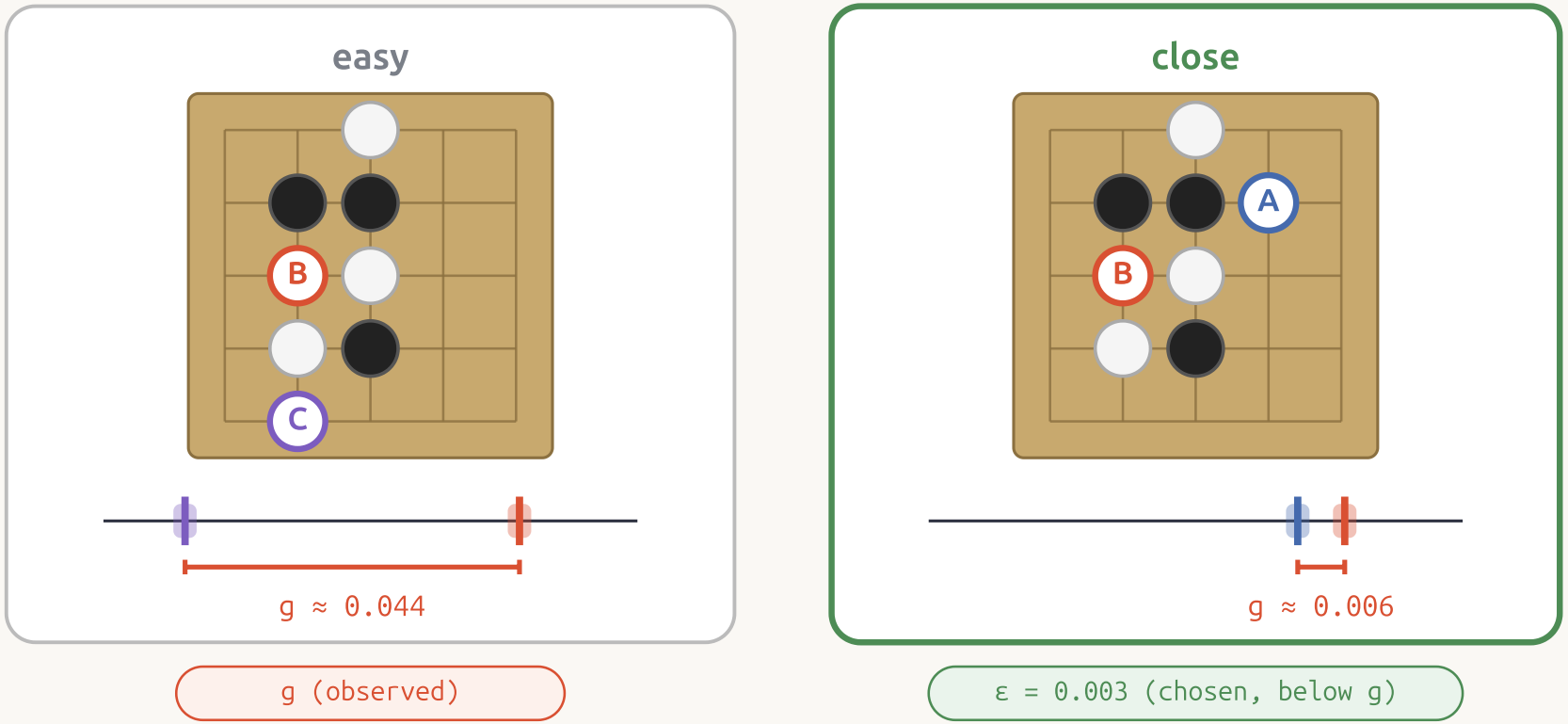
$$\Pr\left[v_{\hat{a}} \;\geq\; \max_j v_j - \varepsilon \right] \;\geq\; \frac{2}{3}$$

Close calls can be designed and measured

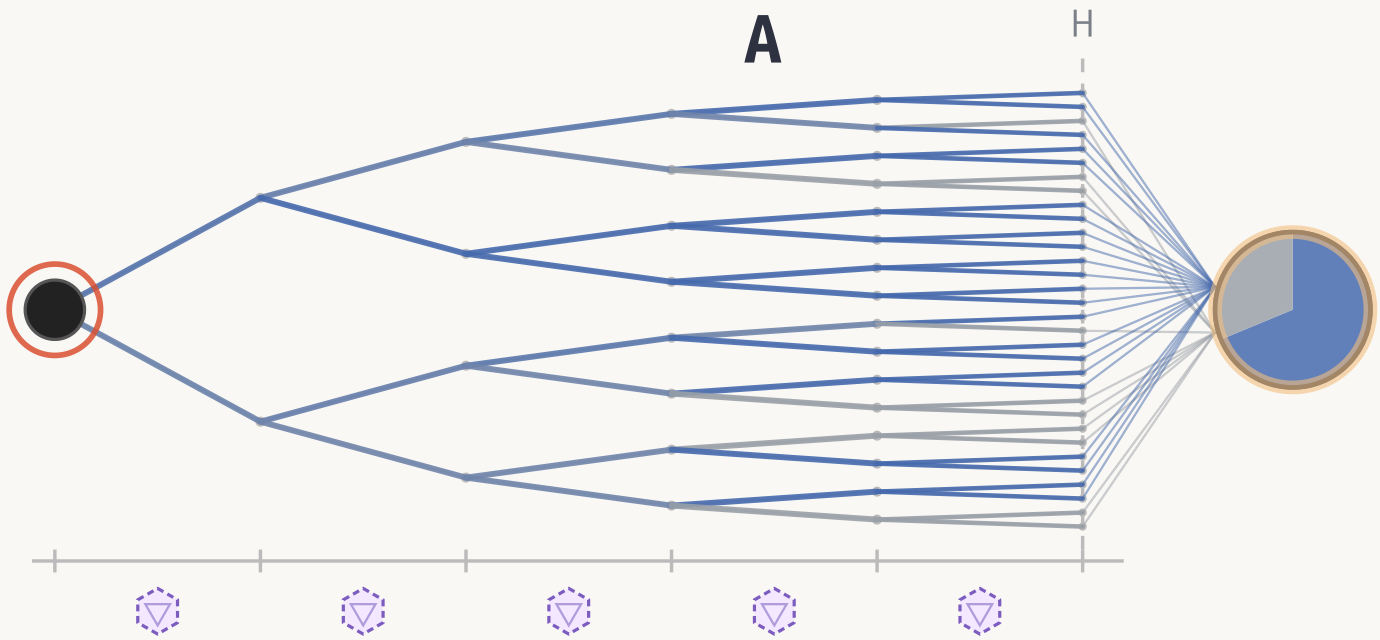


Learn the simulator

Close moves need more evidence

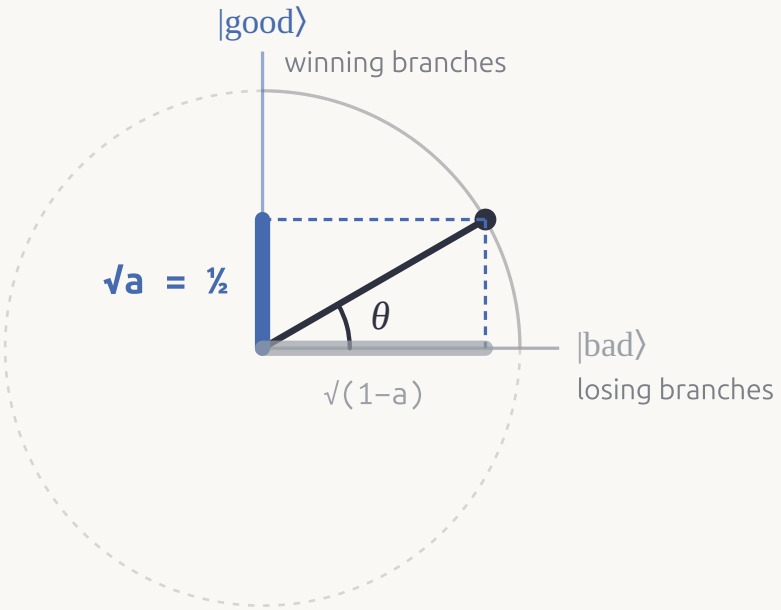


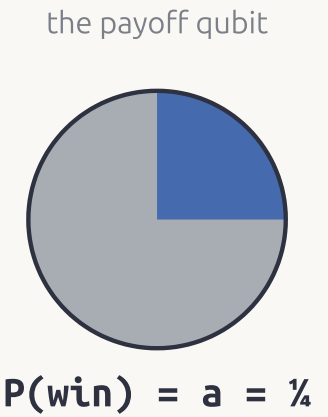
A coherent rollout keeps every branch



$$A_i\,|0\rangle = \sqrt{1-a_i}\;|\psi_{i,0}\rangle|0\rangle + \sqrt{a_i}\;|\psi_{i,1}\rangle|1\rangle \qquad \text{and } A_i^\dagger \text{ on request}$$

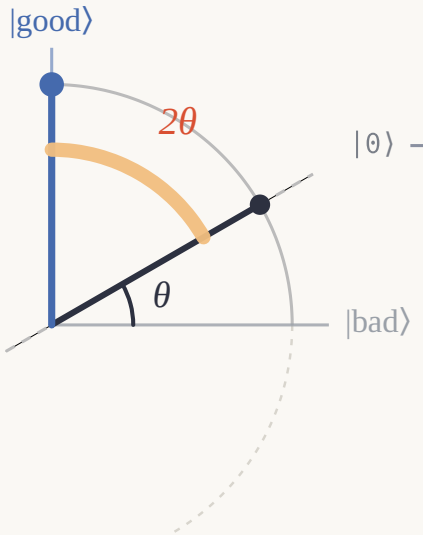
A probability becomes an angle

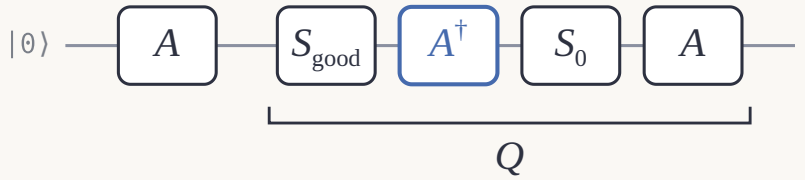




$$a = \sin^2 \theta$$

Two reflections make a rotation





$a = ¼$, $\theta = 30°$

one Q lands on |good⟩

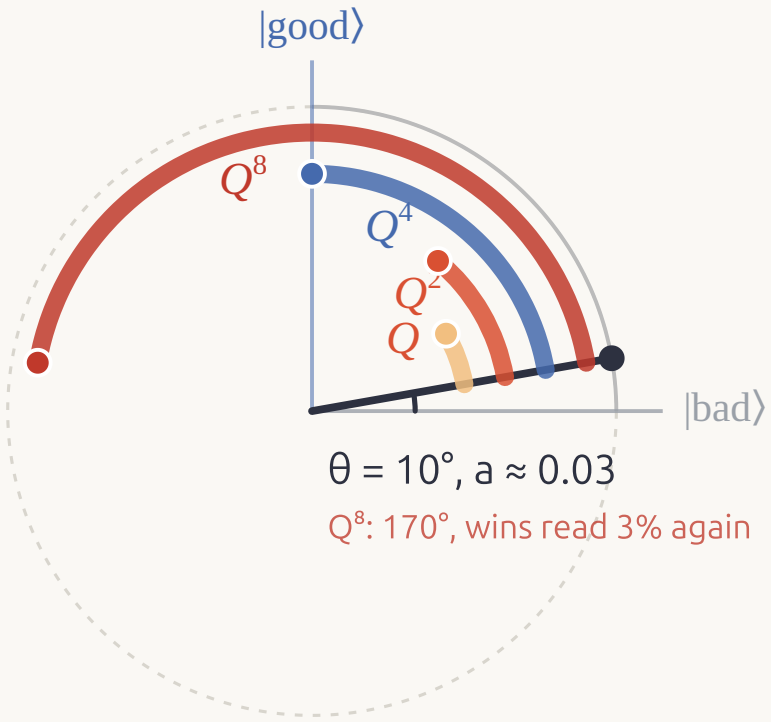
Back

4 / 4 · Read the rotation: 2θ

Complete

$$Q = -\,A\,S_0\,A^\dagger\,S_{\text{good}} \quad = \text{ rotation by } 2\theta$$

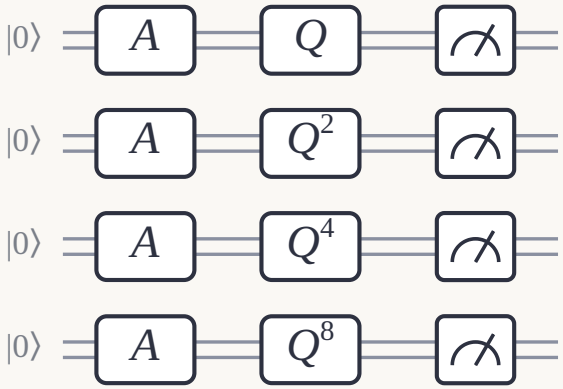
Every turn of Q adds 2θ



θ = 10°, a ≈ 0.03

Q^8: 170°, wins read 3% again

fresh preparation for each circuit



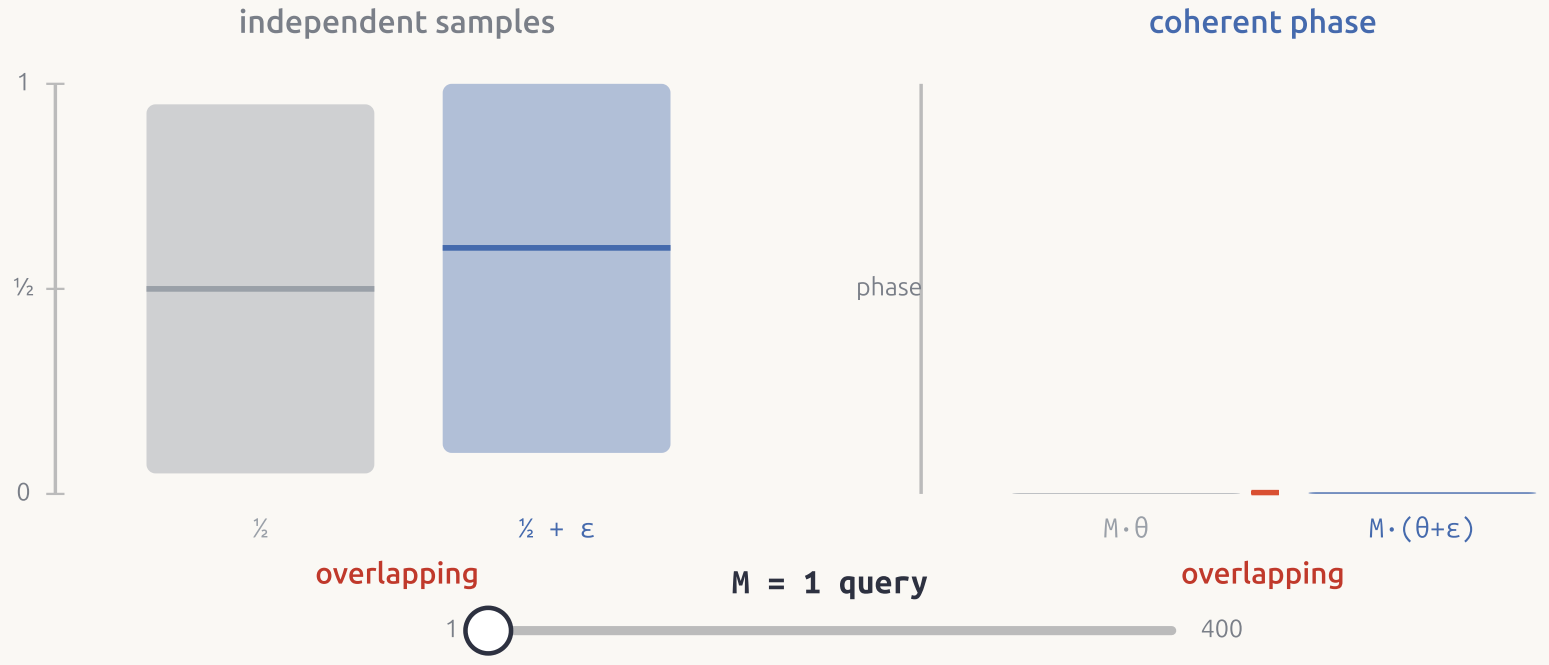
1 + 2 + 4 + 8

15

turns of Q · one A and one A† each

Back 6 / 6 · Apply Q^8: overshoot to 170° Complete

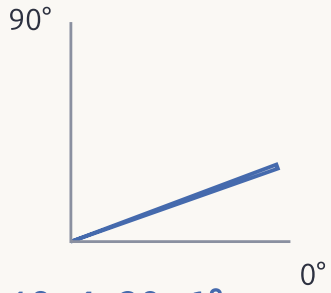
Nearby probabilities become easier to distinguish



$$\frac{\text{signal}}{\text{noise}} \sim \varepsilon\sqrt{M} \;\Rightarrow\; M = \Theta\left(\tfrac{1}{\varepsilon^2}\right)$$

$$\text{phase gap} \sim M\varepsilon \;\Rightarrow\; M = \Theta\left(\tfrac{1}{\varepsilon}\right)$$

Run, measure, refine

Possible original angles	Run	Measure	Narrow
 **19.4–20.6°** $a = \sin^2\theta$ $0.110 \le a \le 0.124$	A — →	1 0 0 0 0 0 0 0 60 / 512 ones	12.8–25.5°
	A Q →	1 1 1 1 0 1 1 0 384 / 512 ones	18.5–21.6°
	A Q^3 →	0 0 1 0 1 0 0 1 212 / 512 ones	19.4–20.6°

Interval width < 2ε: stop here for ε = 0.01.

Back 9 / 9 · Narrow the interval Complete

Fresh preparation for every shot. Measure only after the coherent sequence.

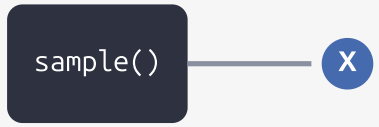
Illustrative batches · 512 shots each · Iterative Quantum Amplitude Estimation

The access contract

ACCESS CONTRACT

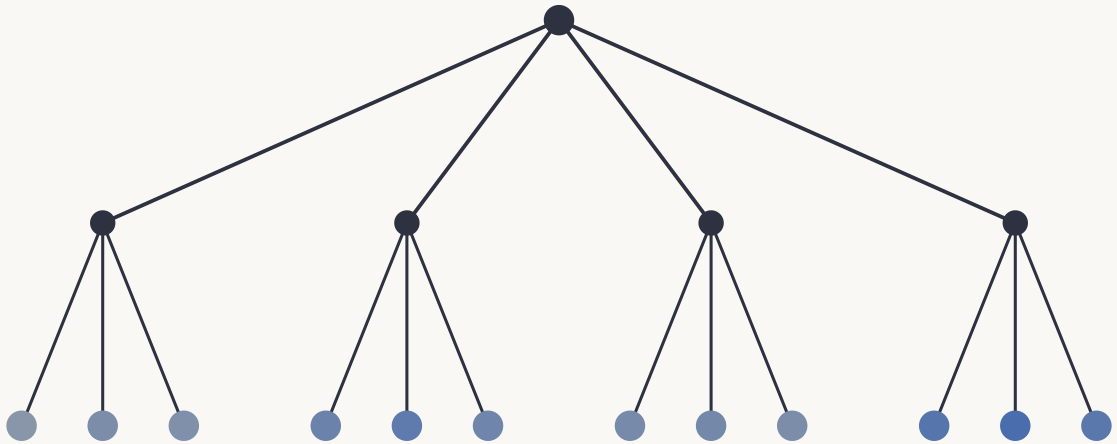
- bounded payoff
- coherent `A`
- inverse `A†`
- no measurement, no fresh randomness inside

ordinary sampling endpoint



- bounded payoff
- coherent `A`
- inverse `A†`
- measures, resamples

Go: compare against the best classical method



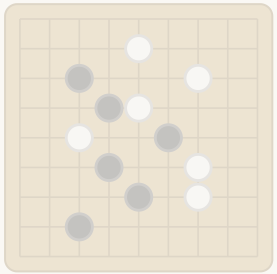
value network

positions are scored; nothing is rolled

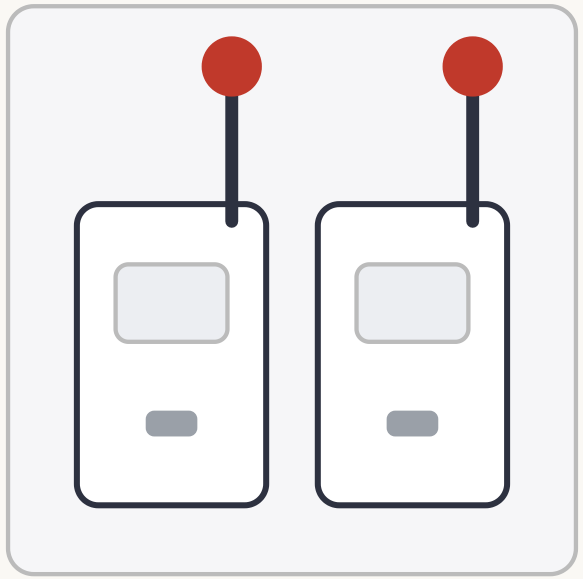
the tree never changes

Check the access and count the work

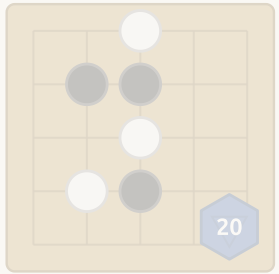
Two arms with nearly equal rewards



Go



two-arm bandit



close-calls game

The sampled bandit lacks an inverse

ACCESS CONTRACT

- bounded payoff
- coherent A
- inverse A†
- no measurement, no fresh randomness inside

MISSING A†

ordinary sampling endpoint

A†?

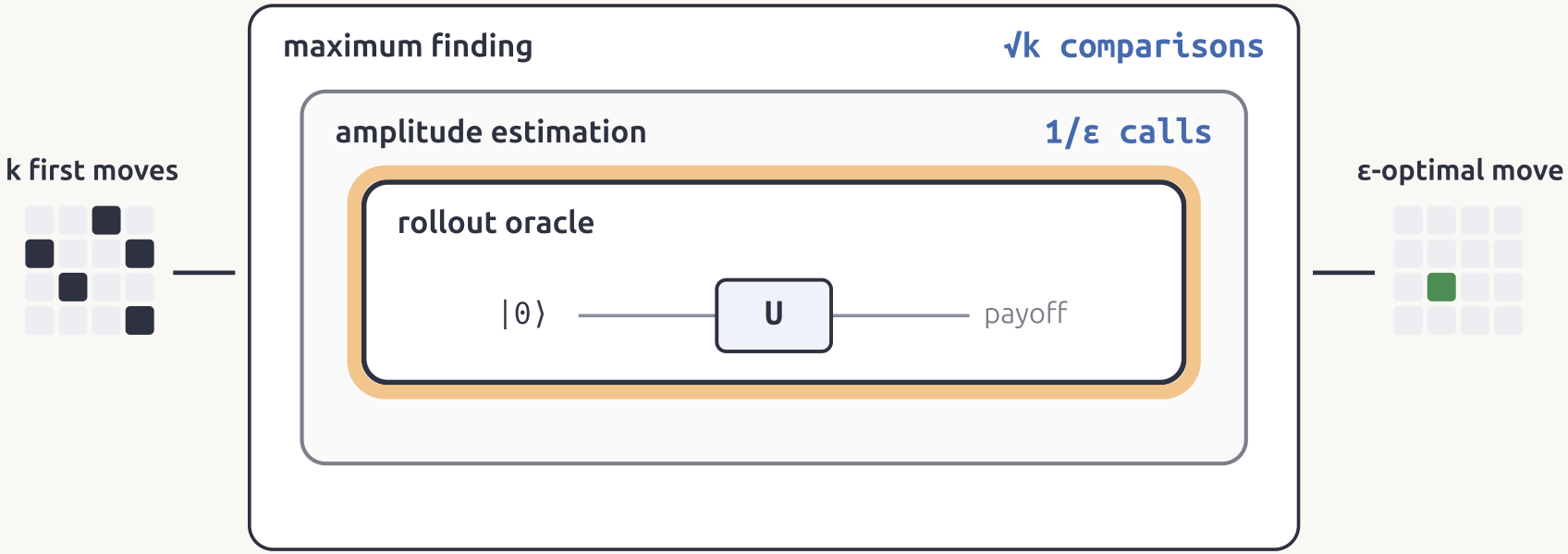
sample()

X′

a fresh random b

- bounded payoff
- coherent A
- inverse A†
- measures, resamples

Choosing among k actions



The quantum composition

maximum finding	amplitude estimation
~~k~~ → **$\sqrt{k}$**	~~$1/\varepsilon^2$~~ → **$1/\varepsilon$**

$\tilde{O}(\sqrt{k} \,/\, \varepsilon)$

$$\Omega\left(\frac{k}{\varepsilon^2}\right) \text{ rollout pulls} \qquad \text{versus} \qquad \widetilde{O}\left(\frac{\sqrt{k}}{\varepsilon}\right) \text{ calls to } A,\ A^\dagger$$

Check the access and count the work

Check the classical alternatives

Does sampling remain expensive after the classical shortcuts?

SMALL STATE SPACE

Enumerate outcomes or reuse repeated states. Check sampled answers against exact values.

LARGER STATE SPACE

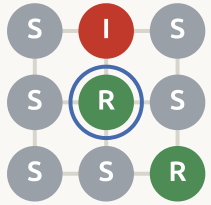
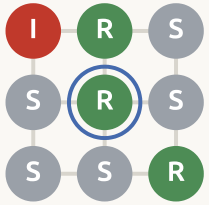
Try structure, approximations, and ways to reduce variance. Count their cost and error.

REMAINING UNCERTAINTY

Measure whether sampling still dominates at the precision the decision needs.

Randomness and a large tree alone don't establish a sampling lower bound.
The appendix gives a bound for a specific access model and hard family.

What carries over to the epidemic model?

Choose	Update	Score
		1 infected threshold = 2 payoff 1
Epidemic	Epidemic	Epidemic
susceptible-site mask	infection + recovery rules	compare count to threshold

Shared reversible circuit

randomness registers · reversible updates · cleared scratch

Back

4 / 4 · Swap the score

Complete

The structure carries over. State encoding and gate costs depend on the model.

Course paper · two implementations

Reuse the construction

Two models, checked

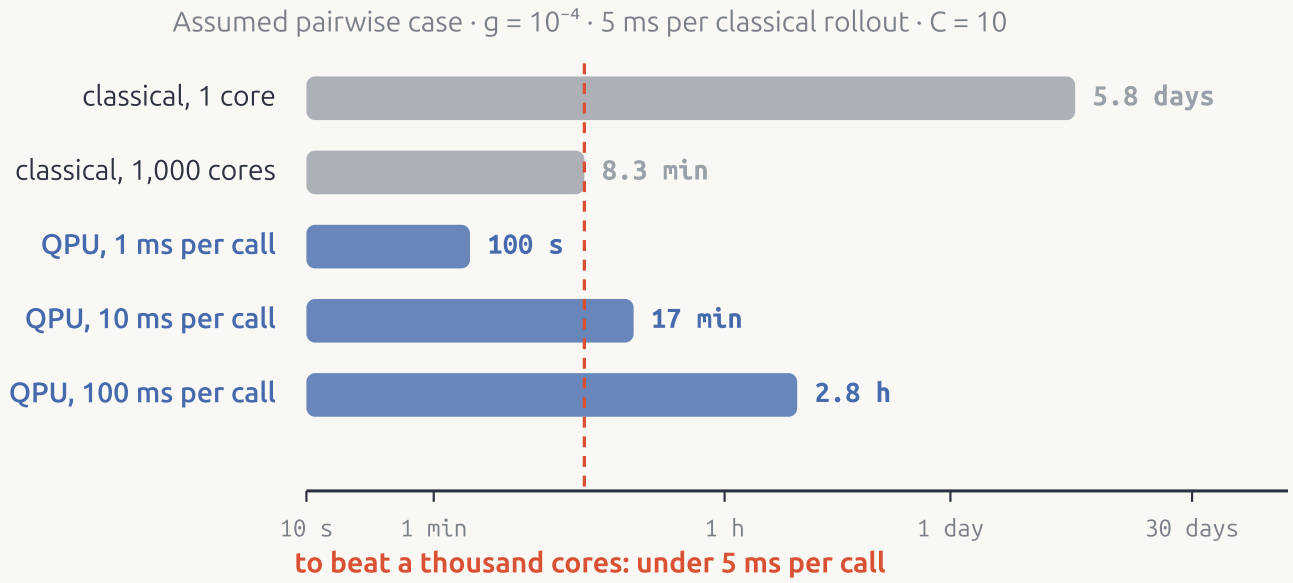
Two small oracle implementations have been checked.

3×3 · 2 rounds	Sway	Epidemic model
Qubits	169	146
Gates	9,768	6,472
Sampled payoff	0.281 ± 0.028	0.883 ± 0.020
Exact payoff	0.271	0.891

Tested branches agree with the classical simulations. Aggregate estimates agree with exact enumeration.

Course paper, Table I · published validation instances, separate from the live illustration · gate counts before decomposition and error-correction overhead

What would one close comparison cost?



$$t_{\text{oracle}} < \frac{t_{\text{roll}}}{C \cdot P \cdot g}$$

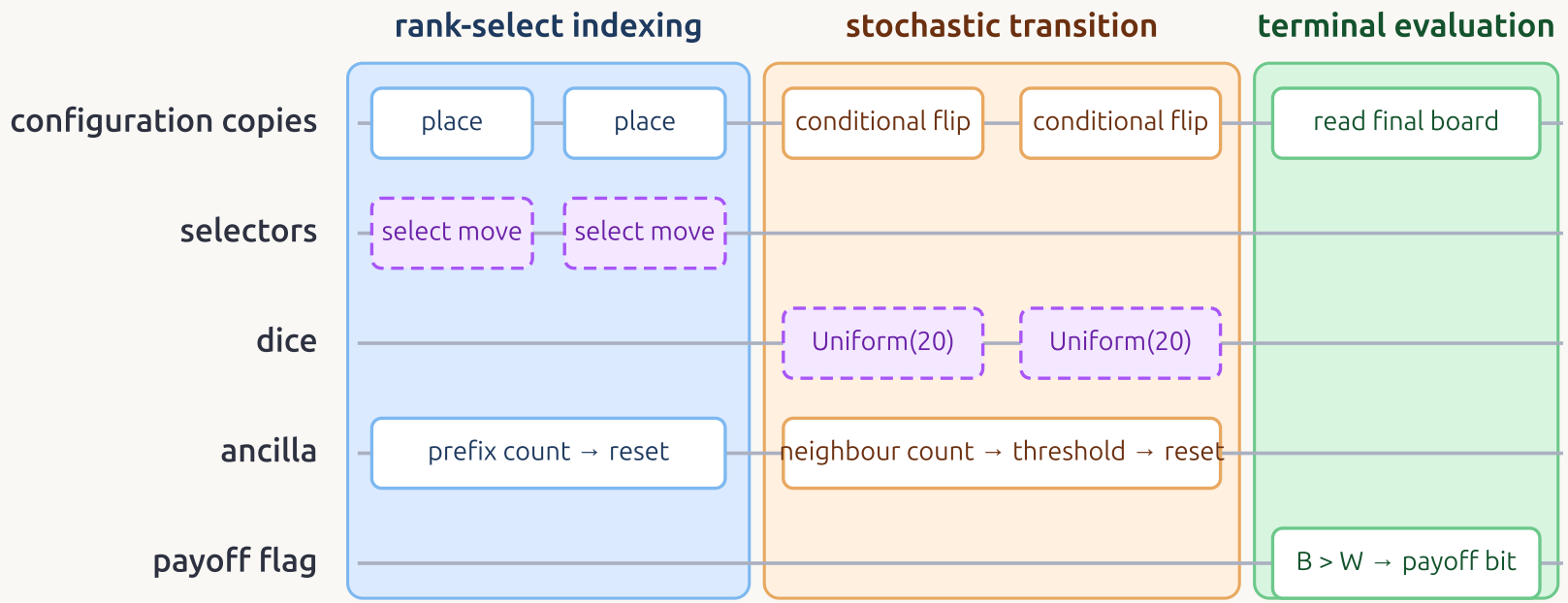
Reuse the construction

What we can reuse

Example	What we learned	Next step
Go	Rollouts are one classical approach.	Compare against stronger classical methods.
Sampled bandit	Close arms need many samples; the interface lacks A†.	Coherent access would need a different implementation.
Sway + epidemic	The shared oracle structure has two implementations.	Build the components and count their costs.

We have a reusable construction to study. A runtime advantage still depends on the instance and hardware costs.

Zoom into one query



Continue to Lesson 3

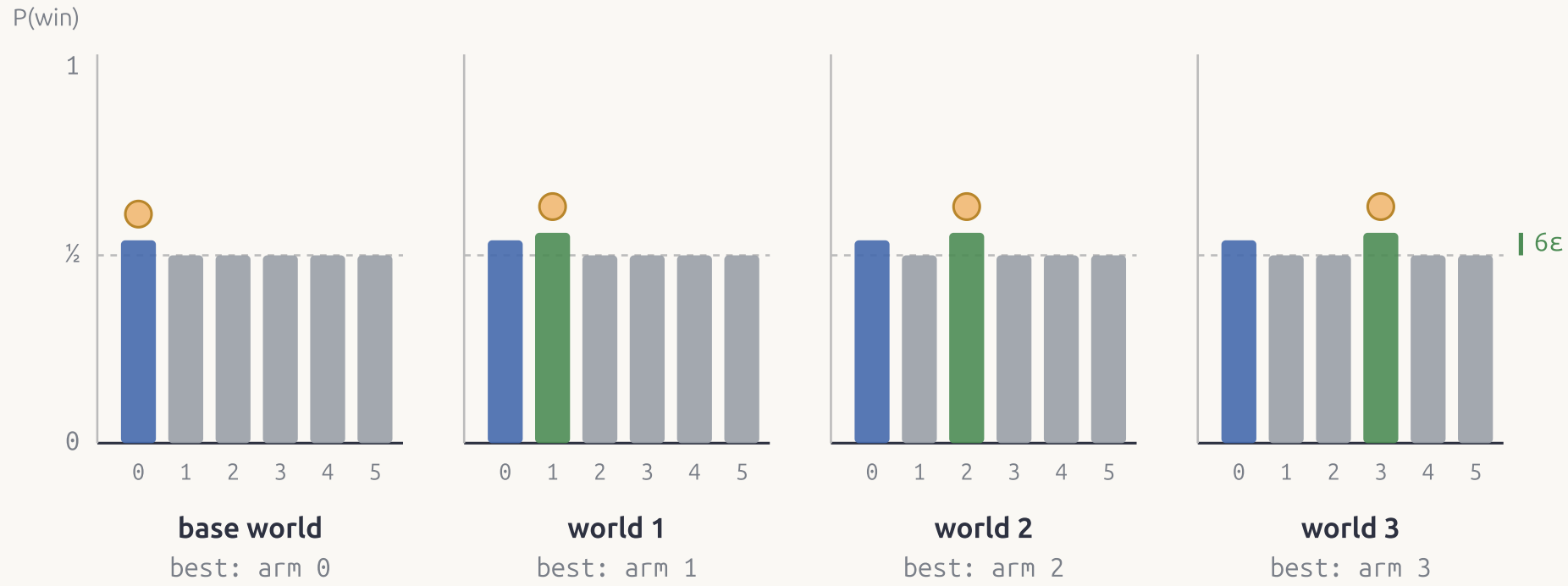
Next: build one rollout circuit.

We'll implement the board, legal-move selection, random choices, and payoff in Lesson 3.

Continue to Lesson 3 | Optional proof and readout details

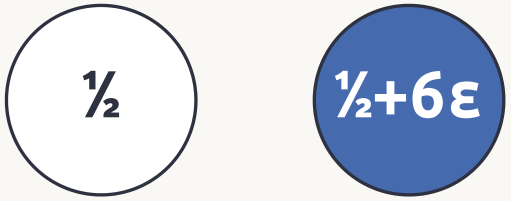
Course paper · Oracle implementation and proofs

A family of almost-identical worlds



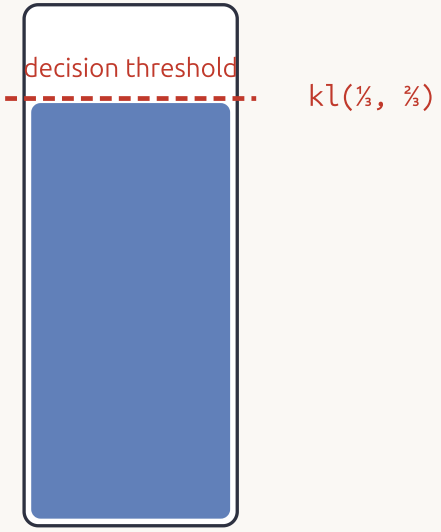
Optional: the sampling lower bound

One pull reveals only ε² of evidence



pulls: 24

≈ 1/ε² of them

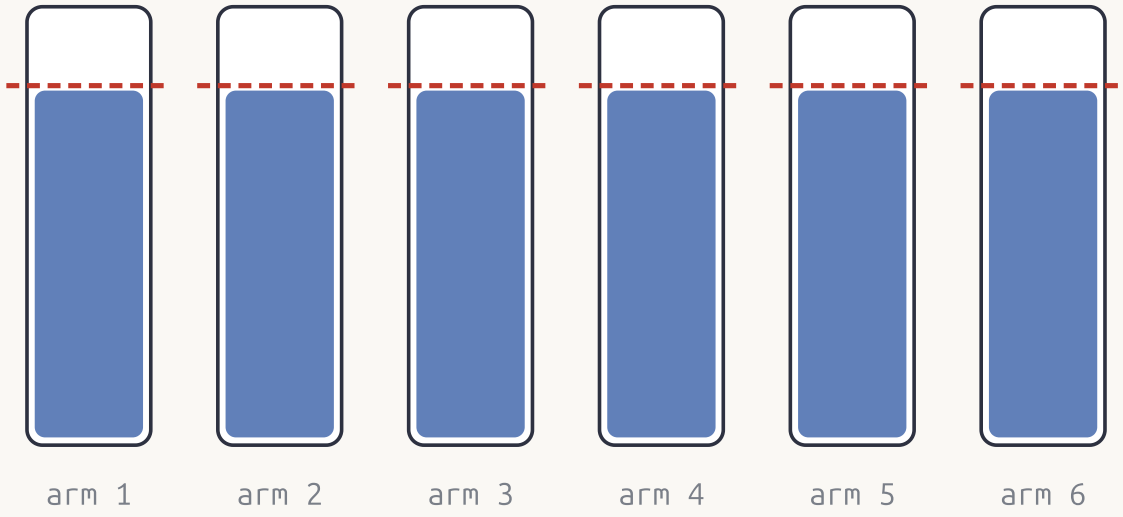


+ Θ(ε²) per pull

$$\mathrm{kl}\left(\tfrac{1}{2},\ \tfrac{1}{2}+6\varepsilon\right) \le 96\,\varepsilon^2 \qquad \mathrm{kl}\left(\tfrac{1}{3},\ \tfrac{2}{3}\right) = \tfrac{\ln 2}{3} \qquad \mathbb{E}[N_j] \ge \frac{\ln 2}{288\,\varepsilon^2}$$

Optional: the sampling lower bound

Every contender must be ruled out

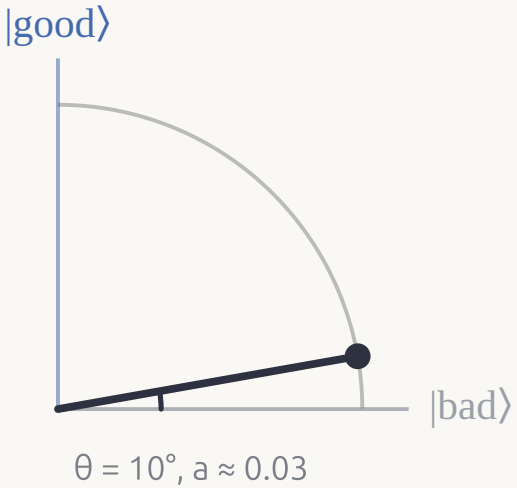


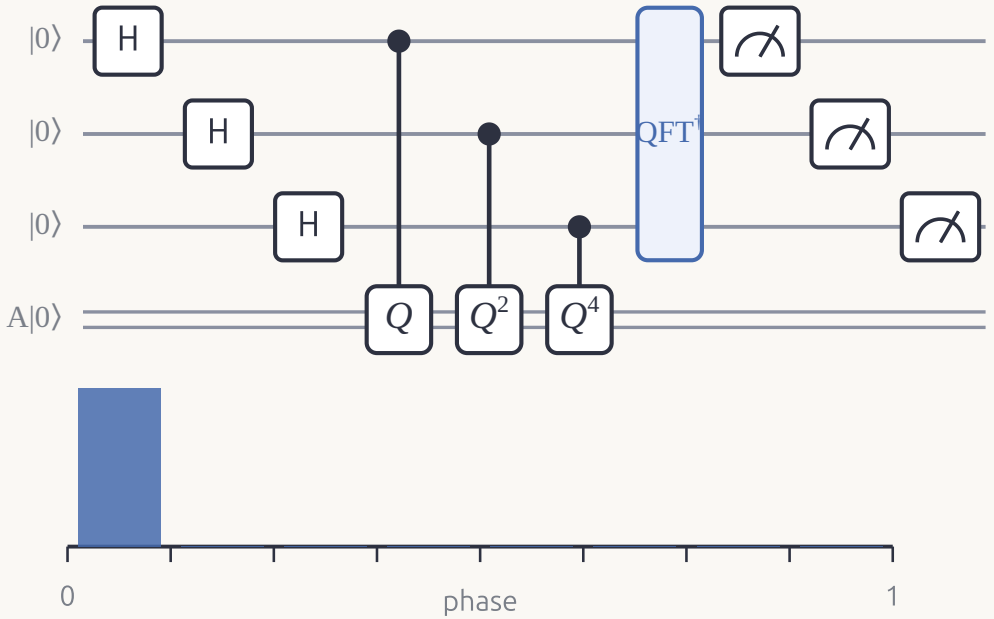
$$(k-1) \cdot \Omega(1/\varepsilon^2) = \Omega(k/\varepsilon^2)$$

What the theorem does, and does not, say

PROVEN	$\tilde{O}(\sqrt{k}/\varepsilon)$ upper bound · $\Omega(k/\varepsilon^2)$ lower bound on the hard family
PROVEN, ROBUSTNESS	the hard instance recurs across an exponential family (bounded influence)
ASSUMED	coherent access: A, $A^\dagger$, no measurement inside
NOT INCLUDED	unrestricted classical structure · FT overhead · wall clock

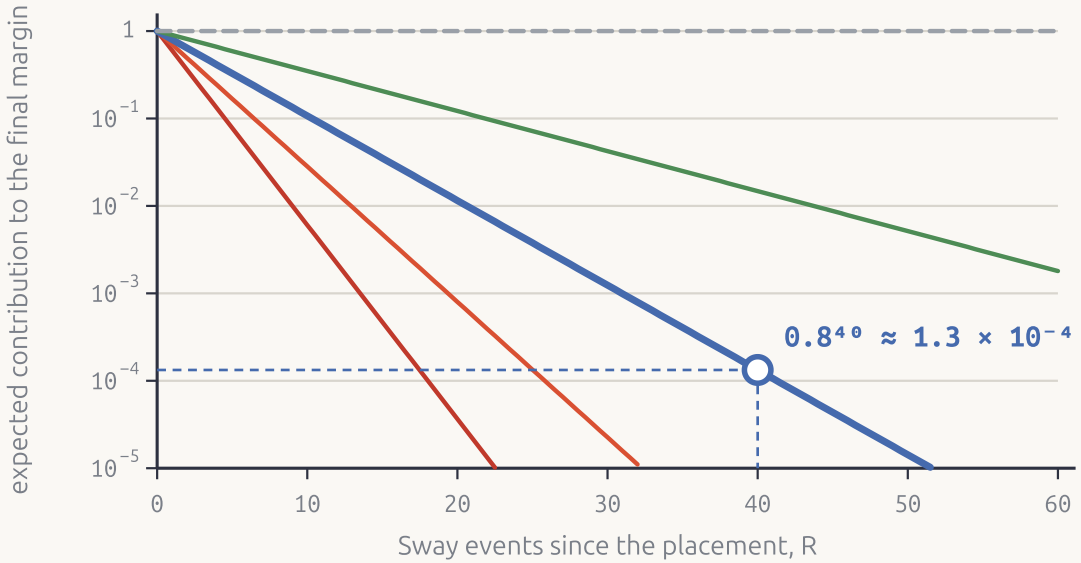
The Fourier-based readout





read θ, then a = sin²θ

A simple model of fading influence



friends	1 – 2p	after 40
c = 0	0.60	1.3×10^{-9}
c = 1	0.70	6.4×10^{-7}
c = 2	**0.80**	$\mathbf{1.3\times10^{-4}}$
c = 3	0.90	0.015
c = 4	1.00	1

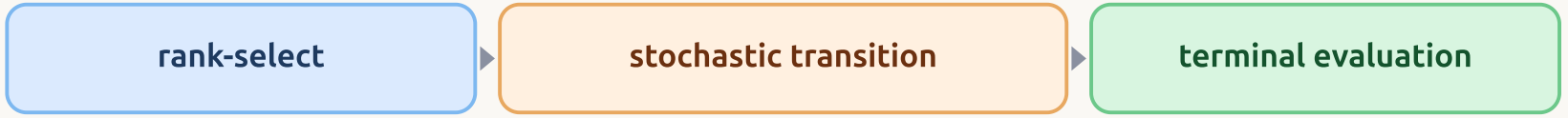
c = 4 never flips while its support holds

Fixed-support model: expected color margin after 40 events. Win-rate gaps need separate measurement.

$$\mathbb{E}[\text{contribution after } R \text{ events}] = (1-2p)^R$$

Optional: readout and resource details

Larger compiled circuits



	Sway		epidemic (SIR)	
instance	qubits	gates	qubits	gates
validated				
3×3, H=2	169	9,768	146	6,472
5×5, H=3	667	55,597	–	–
compiled only				
5×5, H=5	916	76,720	767	56,602
10×10, H=5	3,363	481,201	2,893	280,230
10×10, H=10	6,503	793,901	5,463	558,995
20×20, H=10	25,189	6,072,641	21,409	2,592,183

pre-decomposition, before FT overhead

Back to Lesson 3 and course links