Spoofing Linear Cross-Entropy Benchmarking in Shallow Quantum Circuits

Boaz Barak, Chi-Ning Chou, Xun Gao

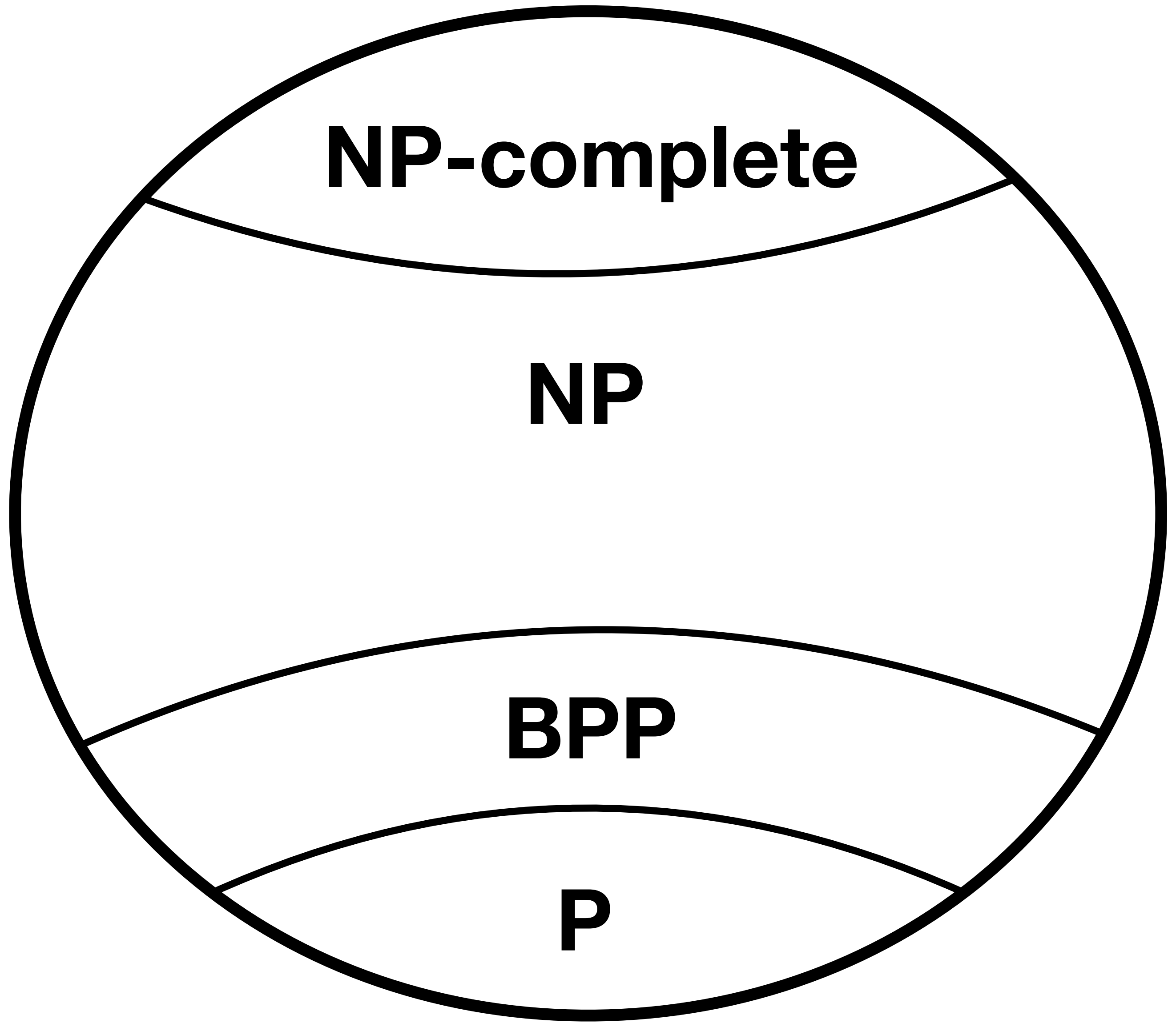
Harvard University

ITCS 2021

Extended Church-Turing Thesis (ECTT)

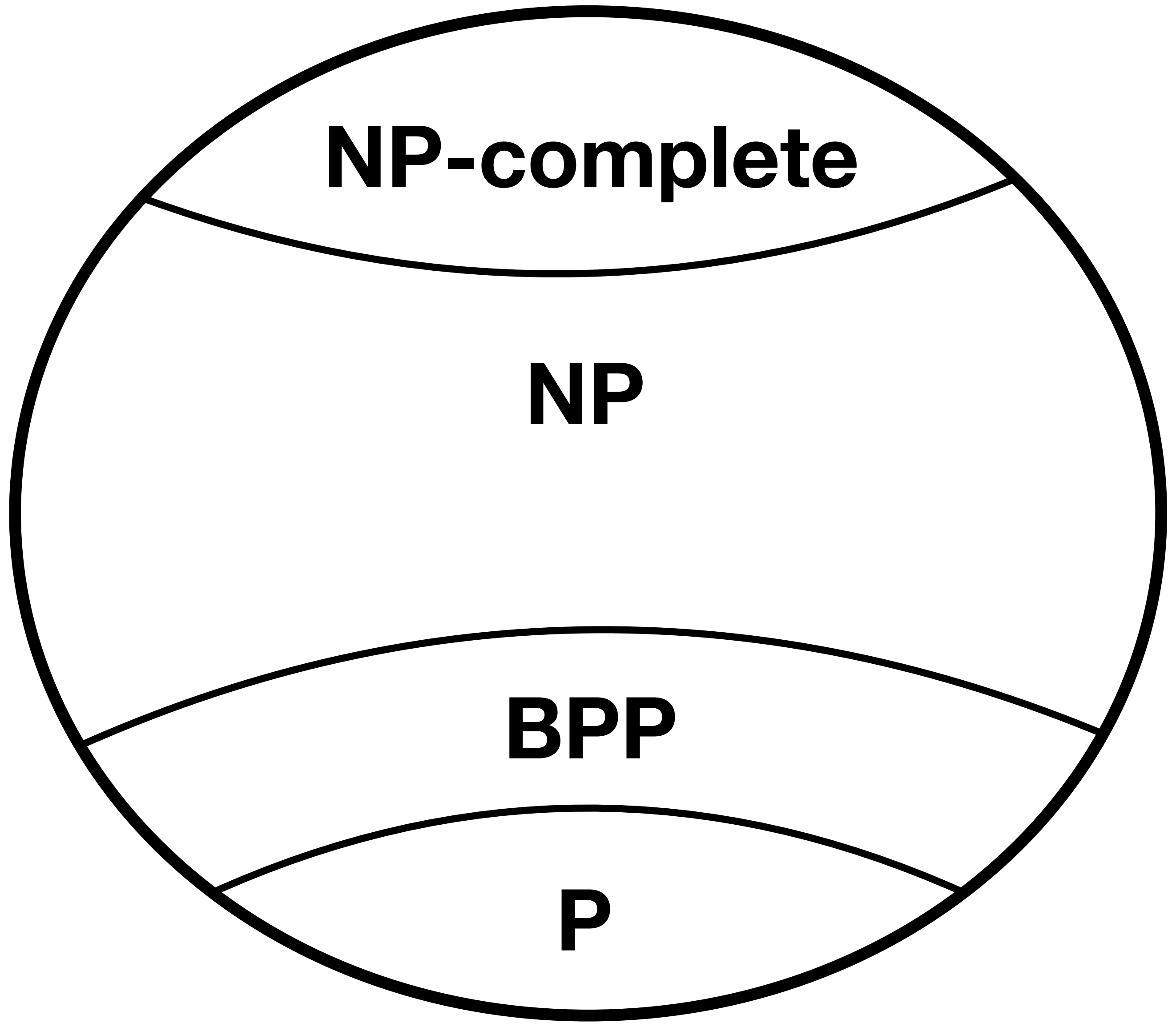
Extended Church-Turing Thesis (ECTT)

Feasible/Efficient computation in the physical world is in **BPP**.



Extended Church-Turing Thesis (ECTT)

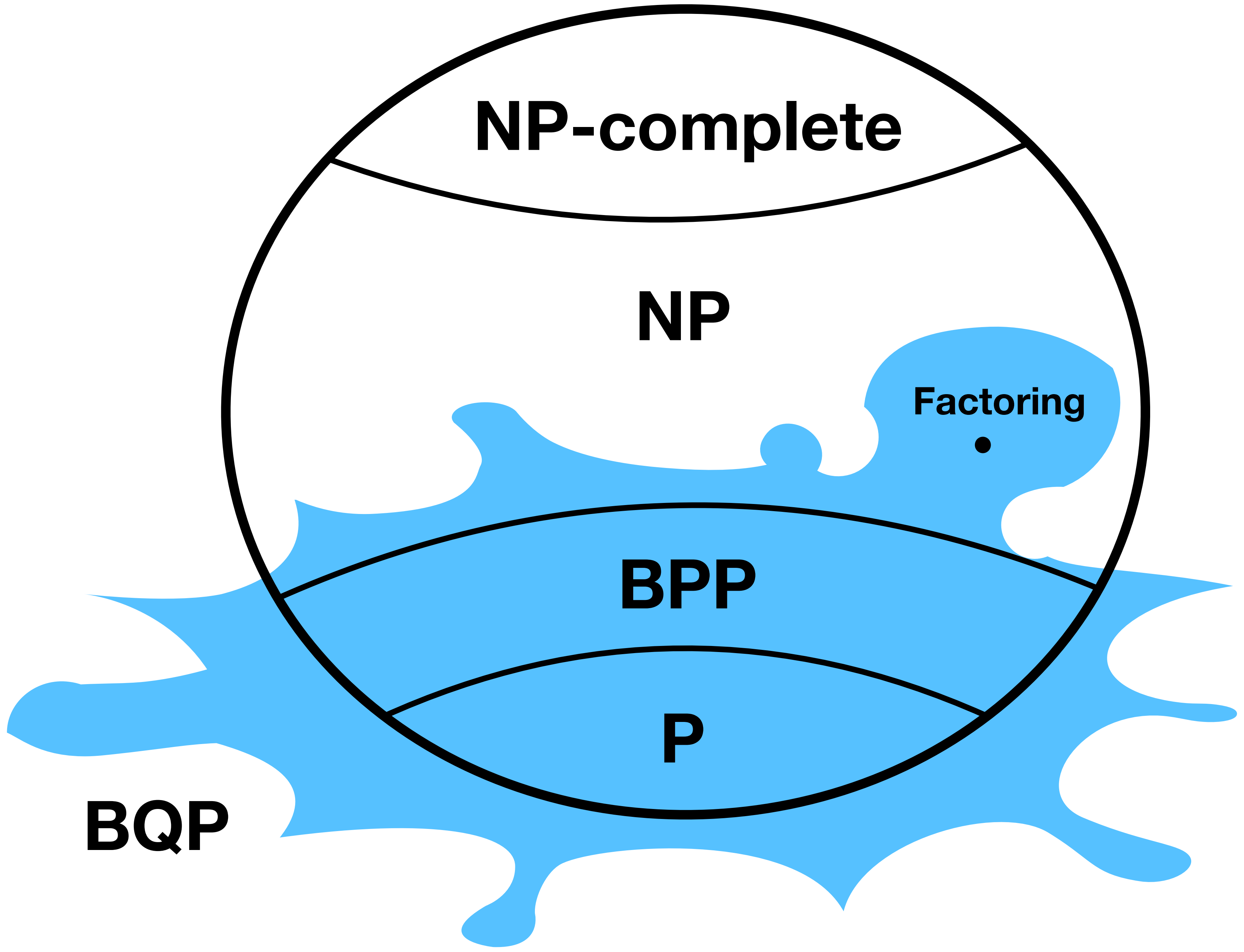
Feasible/Efficient computation in the physical world is in **BPP**.



Q: Can any *realistic* model solve a problem beyond BPP?

Extended Church-Turing Thesis (ECTT)

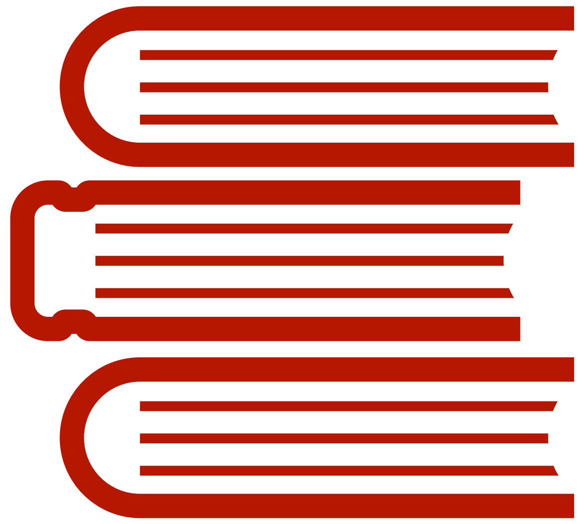
Feasible/Efficient computation in the physical world is in **BPP**.



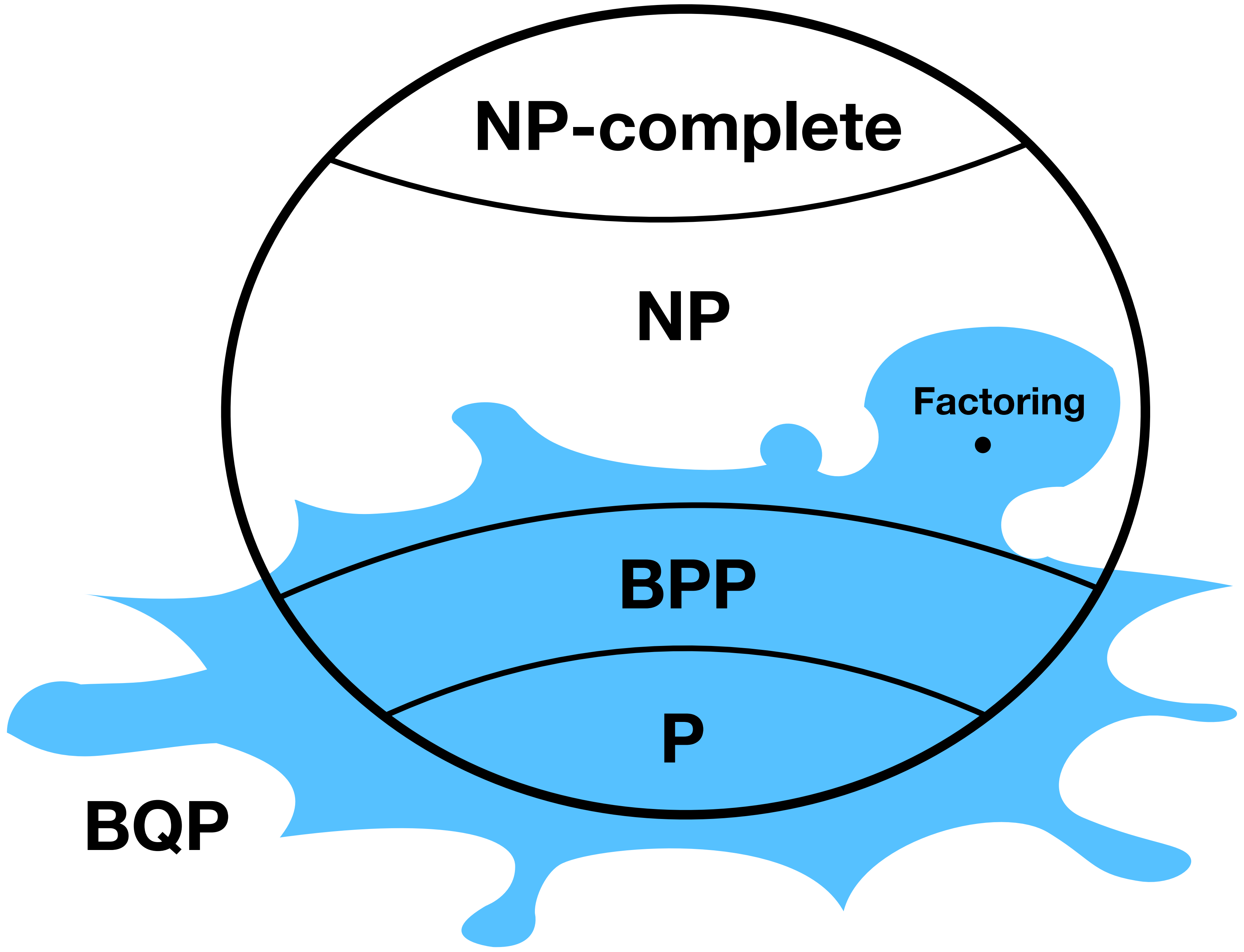
Q: Can any *realistic* model solve a problem beyond BPP?

A: Quantum computation!?

Extended Church-Turing Thesis (ECTT)



Feasible/Efficient computation in the physical world is in **BPP**.



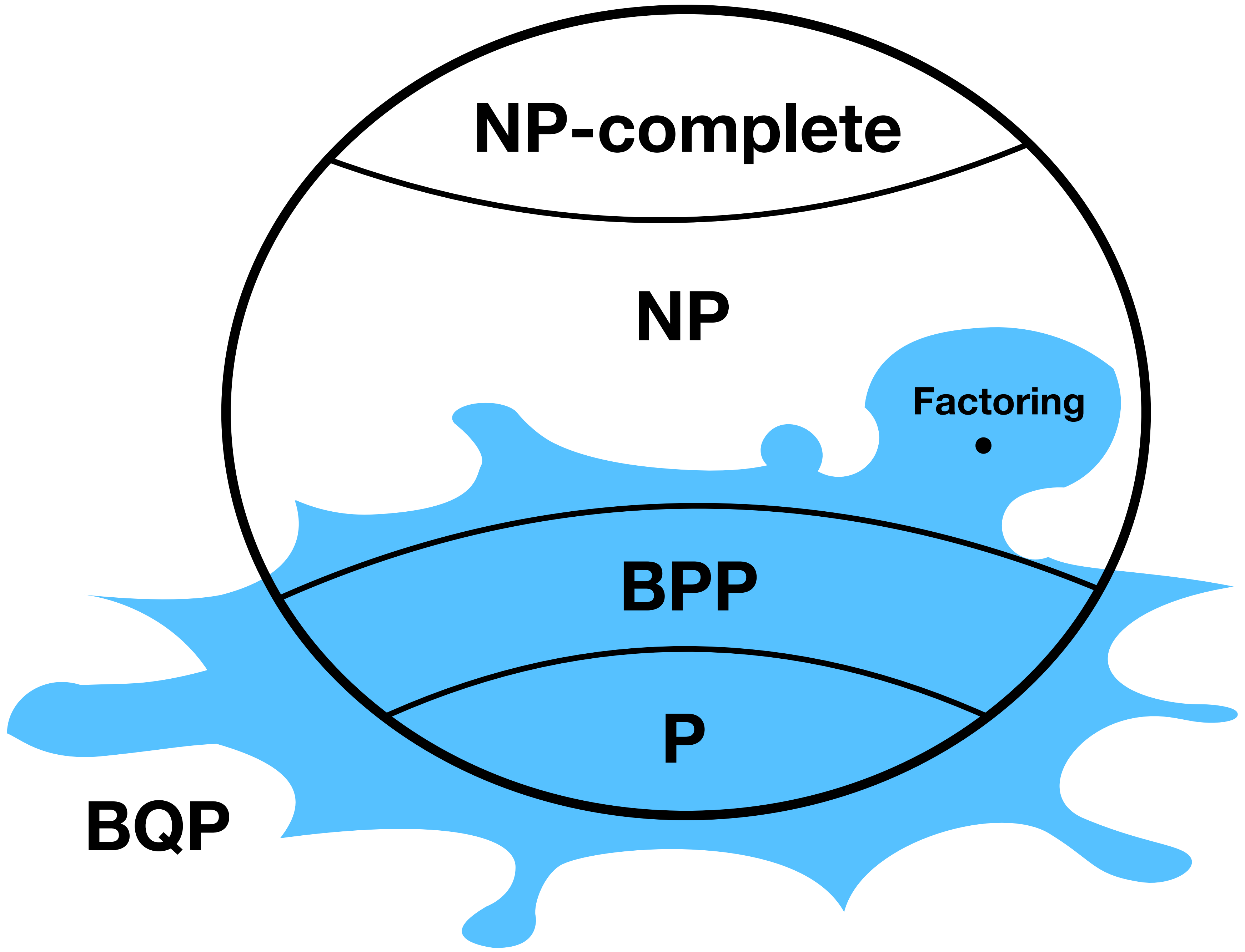
Q: Can any *realistic* model solve a problem beyond BPP?

A: Quantum computation!?

Q: Is BQP realizable? Does quantum computation break ECTT?

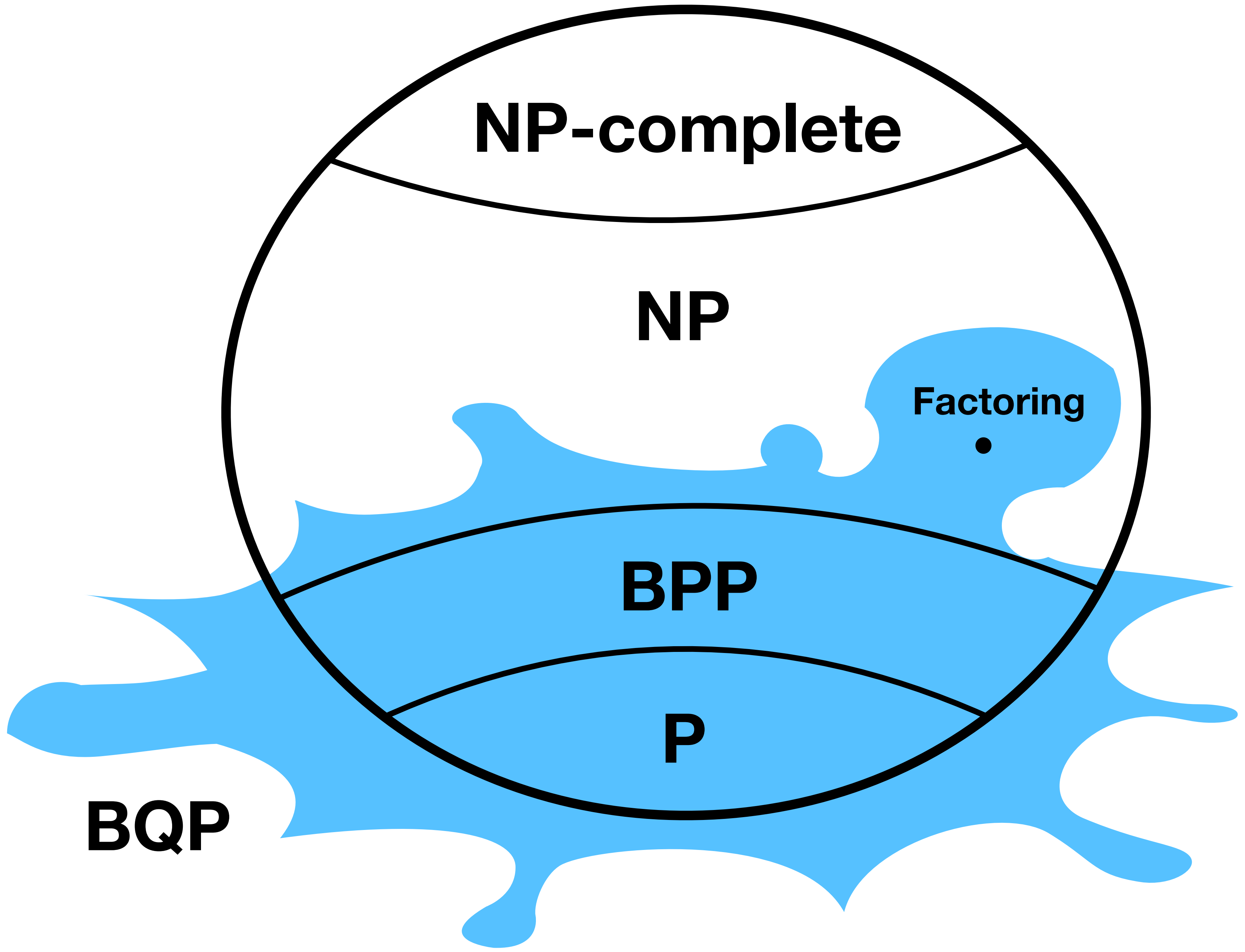
From Theory to the Real Physical World

[Shor 1994]: Quantum computer can solve **Factoring** in polynomial time!



From Theory to the Real Physical World

[Shor 1994]: Quantum computer can solve **Factoring** in polynomial time!

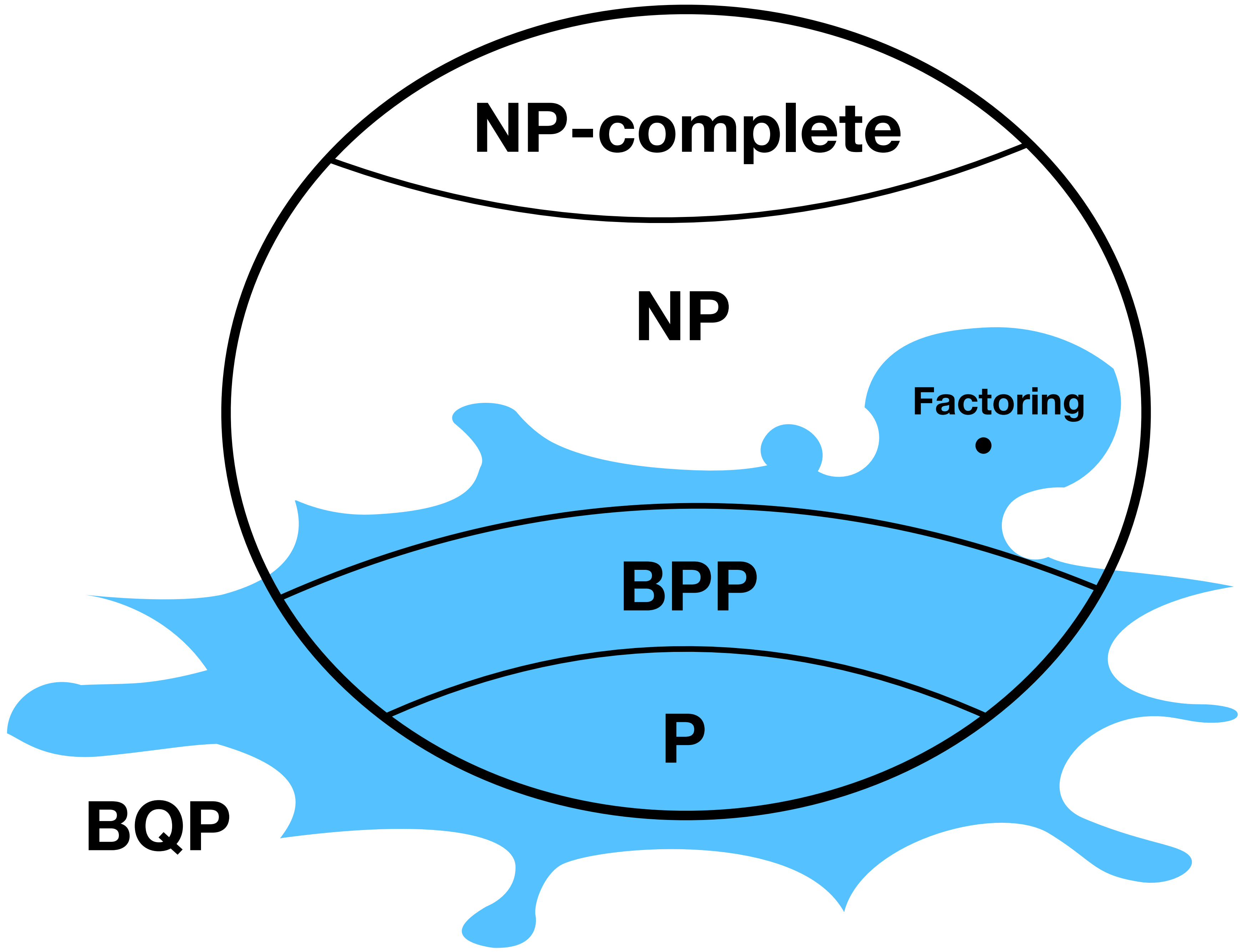


Q: Can we really build scalable quantum computer?

(It requires 2000+ qubits to demonstrate Shor's algorithm.)

From Theory to the Real Physical World

[Shor 1994]: Quantum computer can solve **Factoring** in polynomial time!

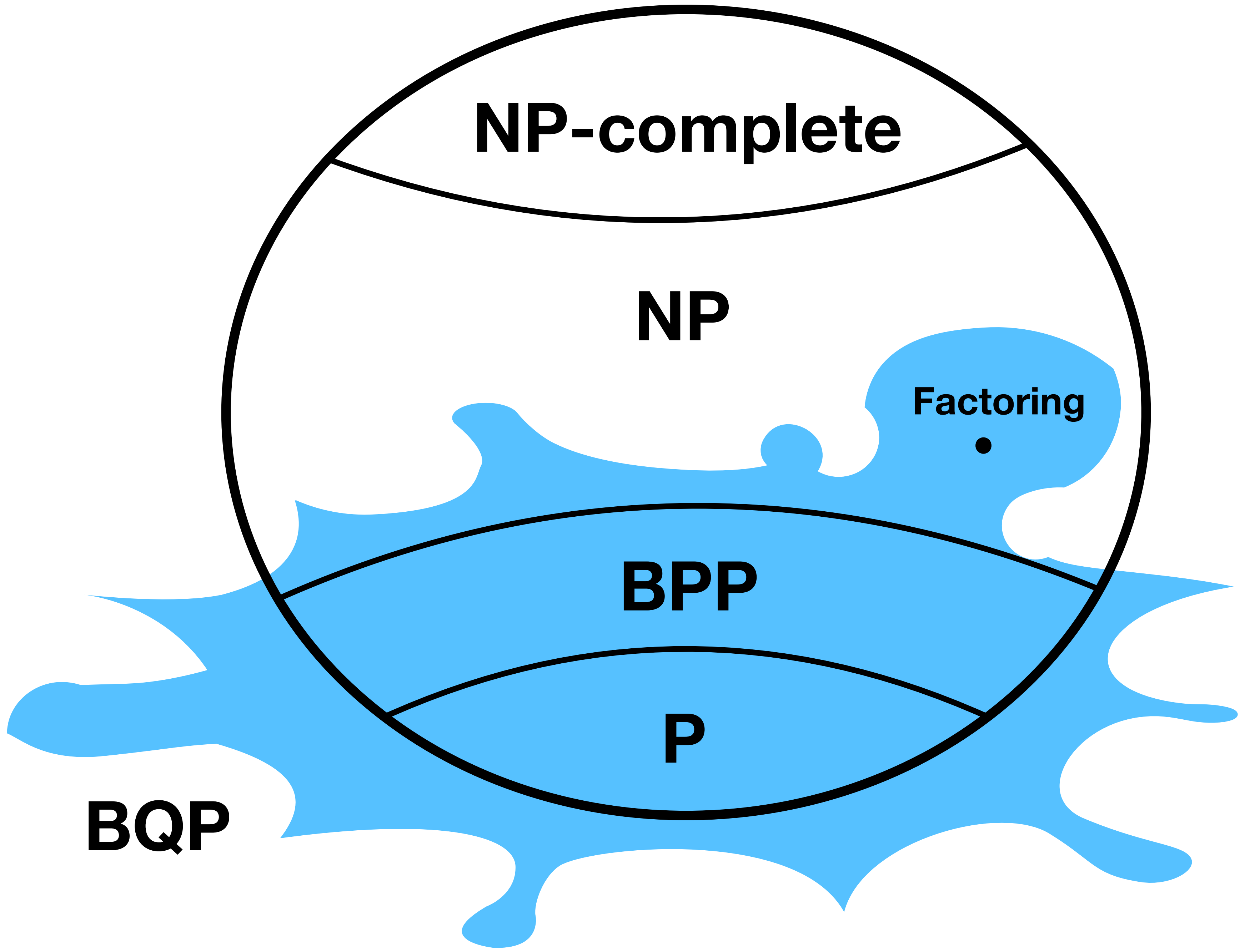


Q: Can we really build scalable quantum computer?
(It requires 2000+ qubits to demonstrate Shor's algorithm.)

Q: Maybe there's fundamental physical barriers?
(E.g., noise stability, error correction, etc.)

From Theory to the Real Physical World

[Shor 1994]: Quantum computer can solve **Factoring** in polynomial time!



Q: Can we really build scalable quantum computer?
(It requires 2000+ qubits to demonstrate Shor's algorithm.)

Q: Maybe there's fundamental physical barriers?
(E.g., noise stability, error correction, etc.)

Q: Can we refute ECTT with near-term technology?

Quantum Computational Supremacy

A compromise between theory and experiment

Quantum Computational Supremacy

A compromise between theory and experiment

Theory

Shor's algorithm

Grover's search

Simon's algorithm

Deutsch-Jozsa algorithm

Quantum Computational Supremacy

A compromise between theory and experiment

Theory

Shor's algorithm

Grover's search

Simon's algorithm

Deutsch-Jozsa algorithm

Experiment

2017 IBM **50** qubits

2018 Intel **49** qubits

2019 IBM **53** qubits

2019 Google **53** qubits

2020 IBM **65** qubits

2020 Pan's group **60** qubits

Quantum Computational Supremacy

A compromise between theory and experiment

Theory

Shor's algorithm

Grover's search

Simon's algorithm

Deutsch-Jozsa algorithm

Quantum Supremacy

Experiment

2017 IBM **50** qubits

2018 Intel **49** qubits

2019 IBM **53** qubits

2019 Google **53** qubits

2020 IBM **65** qubits

2020 Pan's group **60** qubits

Quantum Computational Supremacy

A compromise between theory and experiment

Theory

Shor's algorithm

Grover's search

Simon's algorithm

Deutsch-Jozsa algorithm

Quantum Supremacy

Refute ECTT in a **NISQ** (Noisy Intermediate-Scale Quantum) system!?

Experiment

2017 IBM **50** qubits

2018 Intel **49** qubits

2019 IBM **53** qubits

2019 Google **53** qubits

2020 IBM **65** qubits

2020 Pan's group **60** qubits

Quantum Computational Supremacy

A compromise between theory and experiment

Theory

Shor's algorithm

Grover's search

Simon's algorithm

Deutsch-Jozsa algorithm

Quantum Supremacy

Refute ECTT in a **NISQ** (Noisy Intermediate-Scale Quantum) system!?

Popular candidates:

- Random circuit sampling
- Boson sampling
- IQP

Experiment

2017 IBM **50** qubits

2018 Intel **49** qubits

2019 IBM **53** qubits

2019 Google **53** qubits

2020 IBM **65** qubits

2020 Pan's group **60** qubits

Quantum Computational Supremacy

A compromise between theory and experiment

Theory

Shor's algorithm

Grover's search

Simon's algorithm

Deutsch-Jozsa algorithm

Quantum Supremacy

Refute ECTT in a **NISQ** (Noisy Intermediate-Scale Quantum) system!?

Popular candidates:

- Random circuit sampling
- Boson sampling
- IQP

Experiment

2017 IBM **50** qubits

2018 Intel **49** qubits

2019 IBM **53** qubits

2019 Google **53** qubits

2020 IBM **65** qubits

2020 Pan's group **60** qubits

Quantum Circuits

- Quantum states $|\psi\rangle$.
- Quantum gates U.
- Quantum circuits C.
- Output distribution q_C.

Quantum Circuits

- Quantum states $|\psi\rangle$.

$$|\psi\rangle = \begin{bmatrix} \alpha_{0\cdots00} \\ \alpha_{0\cdots01} \\ \vdots \\ \alpha_{1\cdots11} \end{bmatrix}$$

n-qubit state — length 2^n unit complex vector

- Quantum gates U.

- Quantum circuits C.

- Output distribution q_C.

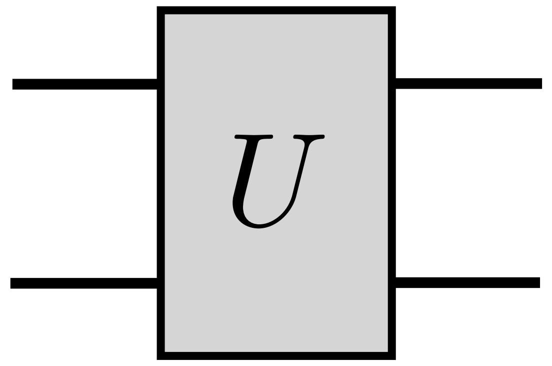
Quantum Circuits

- Quantum states $|\psi\rangle$.

$$|\psi\rangle = \begin{bmatrix} \alpha_{0\cdots 00} \\ \alpha_{0\cdots 01} \\ \vdots \\ \alpha_{1\cdots 11} \end{bmatrix}$$

n-qubit state — length 2^n unit complex vector

- Quantum gates U.



$$= U \otimes I_{2^{n-2}}$$

2-qubit gate — 2^n by 2^n unitary matrix

- Quantum circuits C.
- Output distribution q_C.

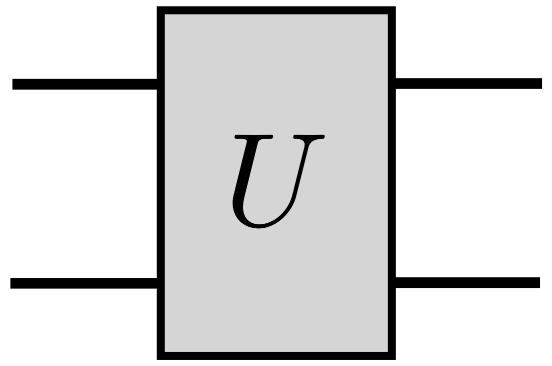
Quantum Circuits

- Quantum states $|\psi\rangle$.

$$|\psi\rangle = \begin{bmatrix} \alpha_{0\cdots00} \\ \alpha_{0\cdots01} \\ \vdots \\ \alpha_{1\cdots11} \end{bmatrix}$$

n-qubit state — length 2^n unit complex vector

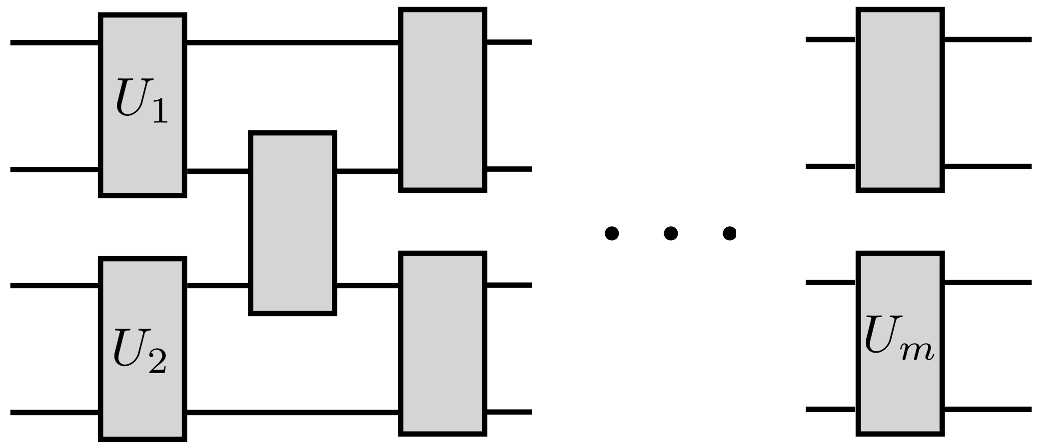
- Quantum gates U.



$$= U \otimes I_{2^{n-2}}$$

2-qubit gate — 2^n by 2^n unitary matrix

- Quantum circuits C.



$$= C$$

circuit with 1- and 2-qubit gates — 2^n by 2^n unitary matrix

- Output distribution q_C.

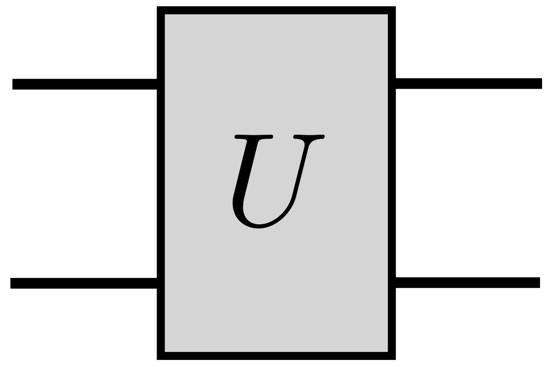
Quantum Circuits

- Quantum states $|\psi\rangle$.

$$|\psi\rangle = \begin{bmatrix} \alpha_{0\cdots00} \\ \alpha_{0\cdots01} \\ \vdots \\ \alpha_{1\cdots11} \end{bmatrix}$$

n-qubit state — length 2^n unit complex vector

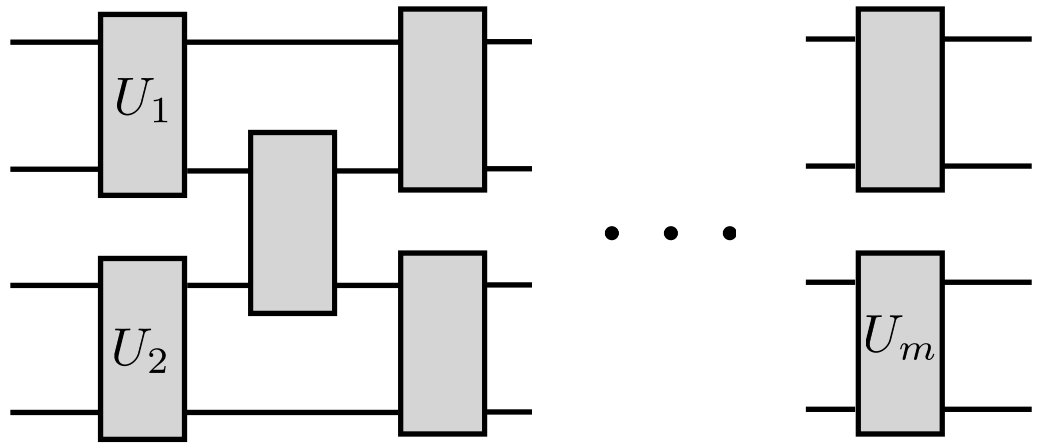
- Quantum gates U.



$$= U \otimes I_{2^{n-2}}$$

2-qubit gate — 2^n by 2^n unitary matrix

- Quantum circuits C.



$$= C$$

circuit with 1- and 2-qubit gates — 2^n by 2^n unitary matrix

- Output distribution q_C.

$$C|\psi\rangle = \begin{bmatrix} \alpha_{0\cdots00} \\ \alpha_{0\cdots01} \\ \vdots \\ \alpha_{1\cdots11} \end{bmatrix}$$

length 2^n unit complex vector

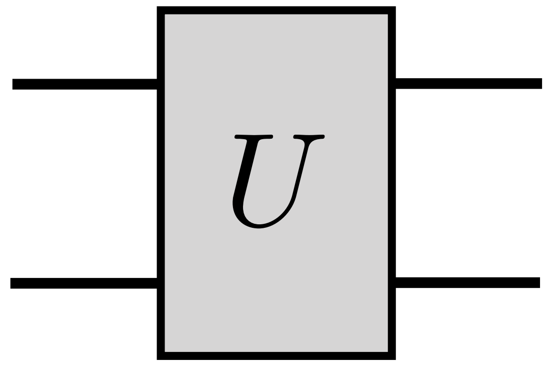
Quantum Circuits

- Quantum states $|\psi\rangle$.

$$|\psi\rangle = \begin{bmatrix} \alpha_{0\cdots00} \\ \alpha_{0\cdots01} \\ \vdots \\ \alpha_{1\cdots11} \end{bmatrix}$$

n-qubit state — length 2^n unit complex vector

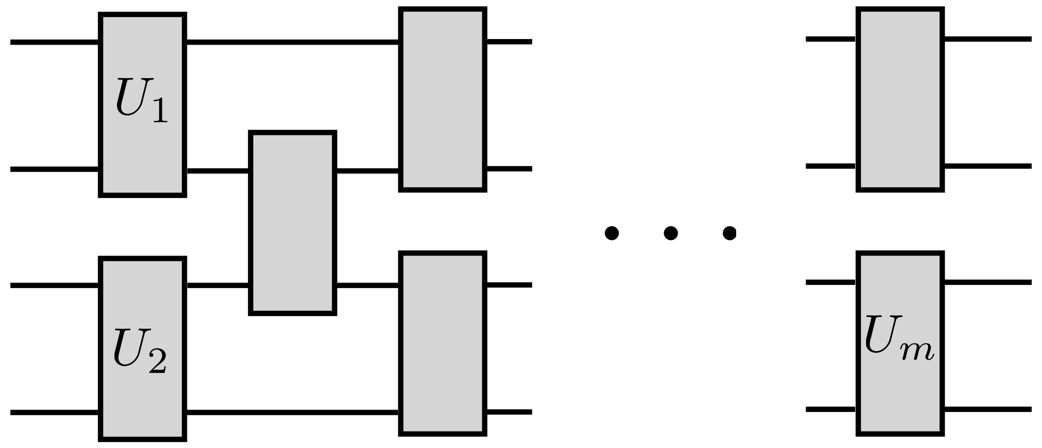
- Quantum gates U.



$$= \quad U \otimes I_{2^{n-2}}$$

2-qubit gate — 2^n by 2^n unitary matrix

- Quantum circuits C.



$$= \quad C$$

circuit with 1- and 2-qubit gates — 2^n by 2^n unitary matrix

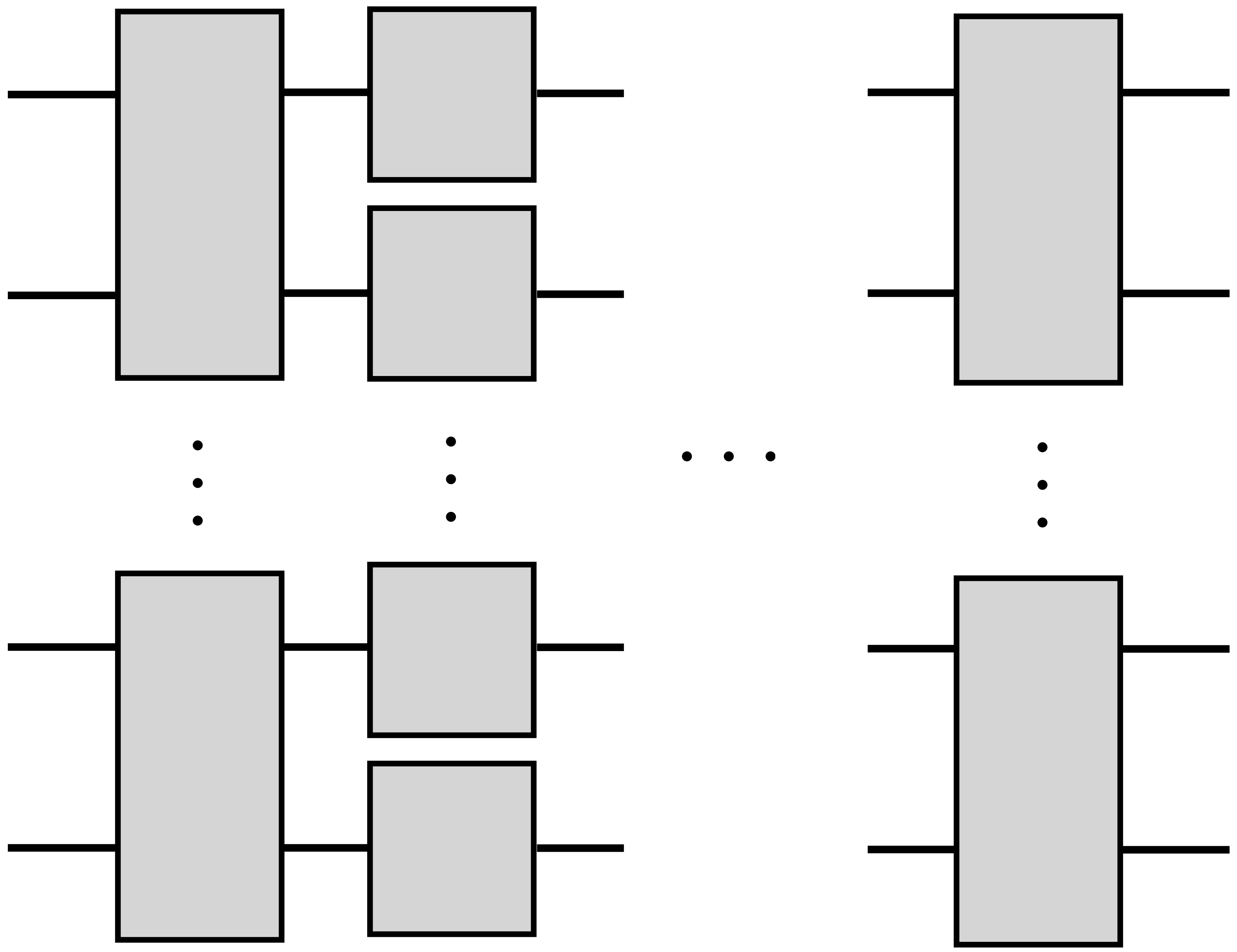
- Output distribution q_C.

$$C|\psi\rangle = \begin{bmatrix} \alpha_{0\cdots00} \\ \alpha_{0\cdots01} \\ \vdots \\ \alpha_{1\cdots11} \end{bmatrix} \qquad q_C(x) = |\alpha_x|^2$$

length 2^n unit complex vector

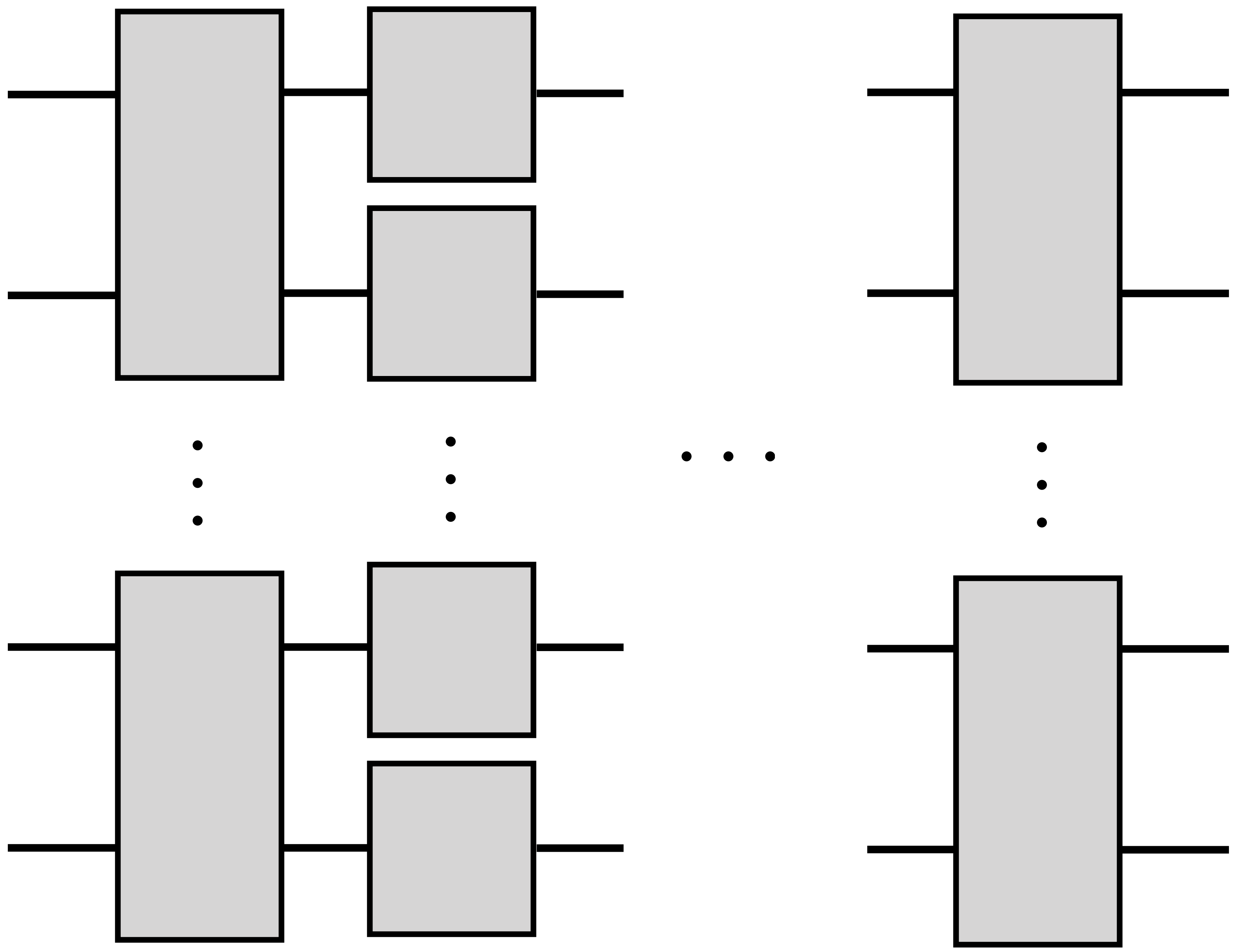
Random Circuit Sampling

The lead-candidate used by Google's Sycamore



Random Circuit Sampling

The lead-candidate used by Google's Sycamore

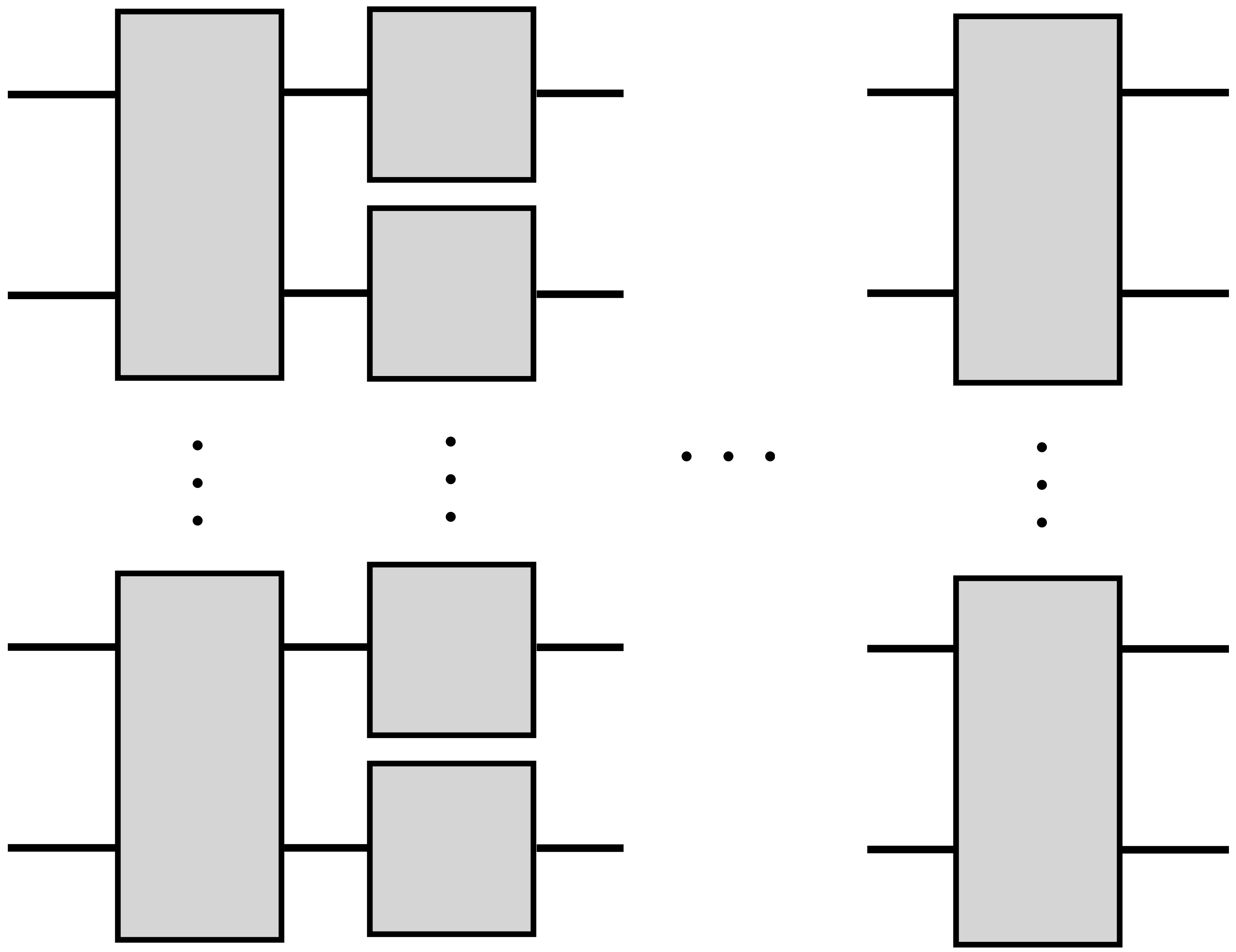


Challenger

(E.g., Quantum skeptics)

Random Circuit Sampling

The lead-candidate used by Google's Sycamore



Challenger

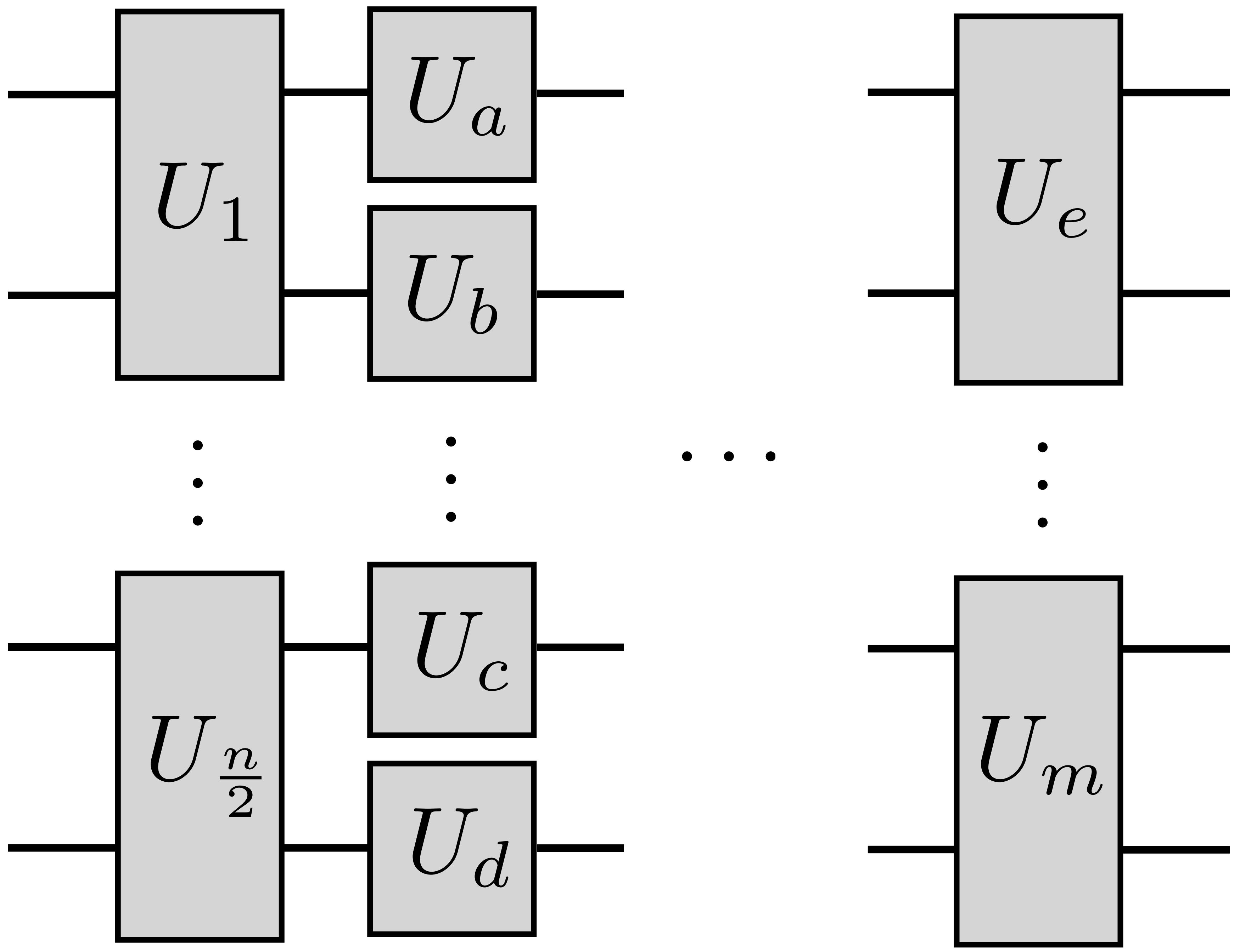
(E.g., Quantum skeptics)

Prover

(E.g., Google)

Random Circuit Sampling

The lead-candidate used by Google's Sycamore



Challenger

(E.g., Quantum skeptics)

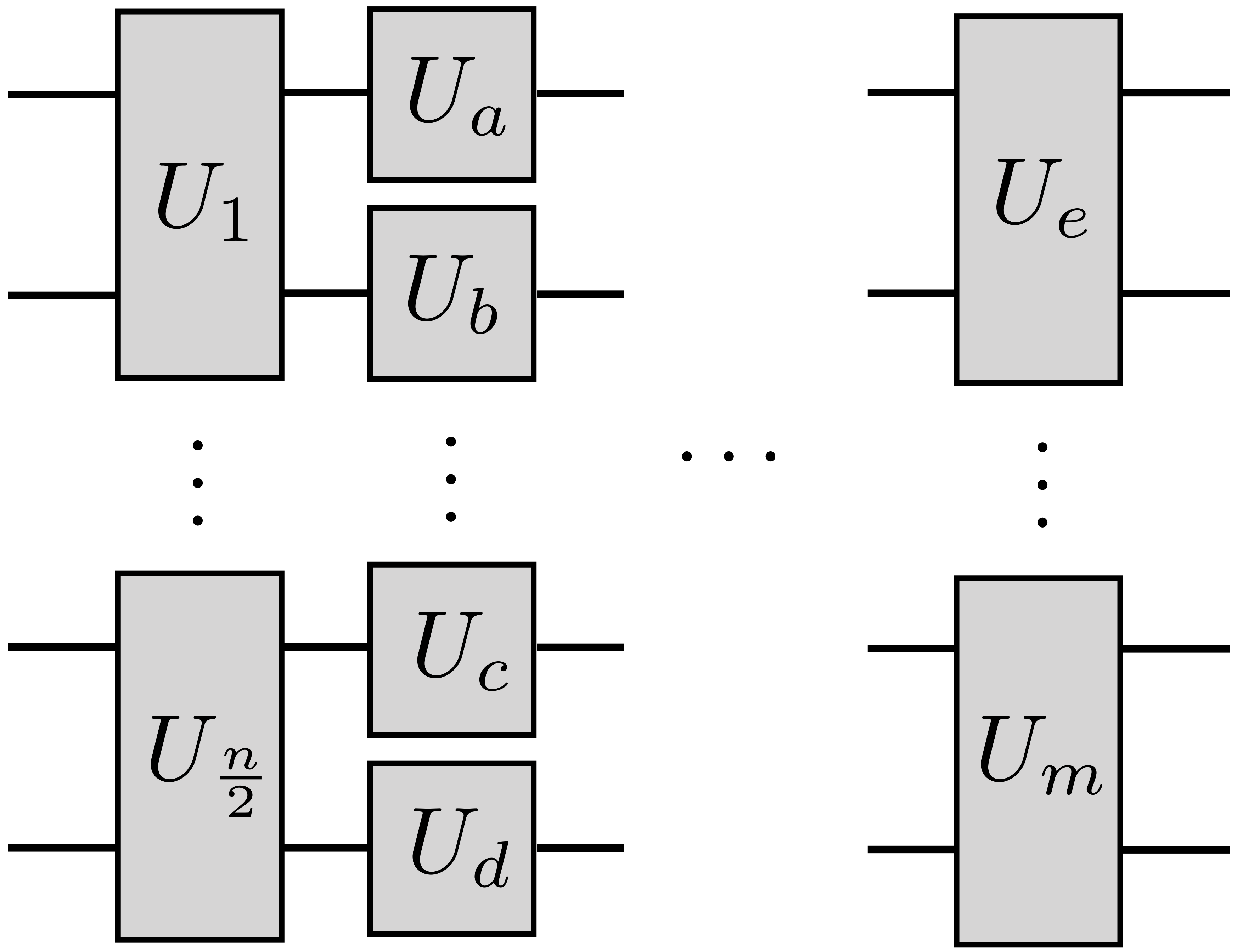
1. Randomly pick $U_1, \ldots, U_m$.

Prover

(E.g., Google)

Random Circuit Sampling

The lead-candidate used by Google's Sycamore



Challenger

(E.g., Quantum skeptics)

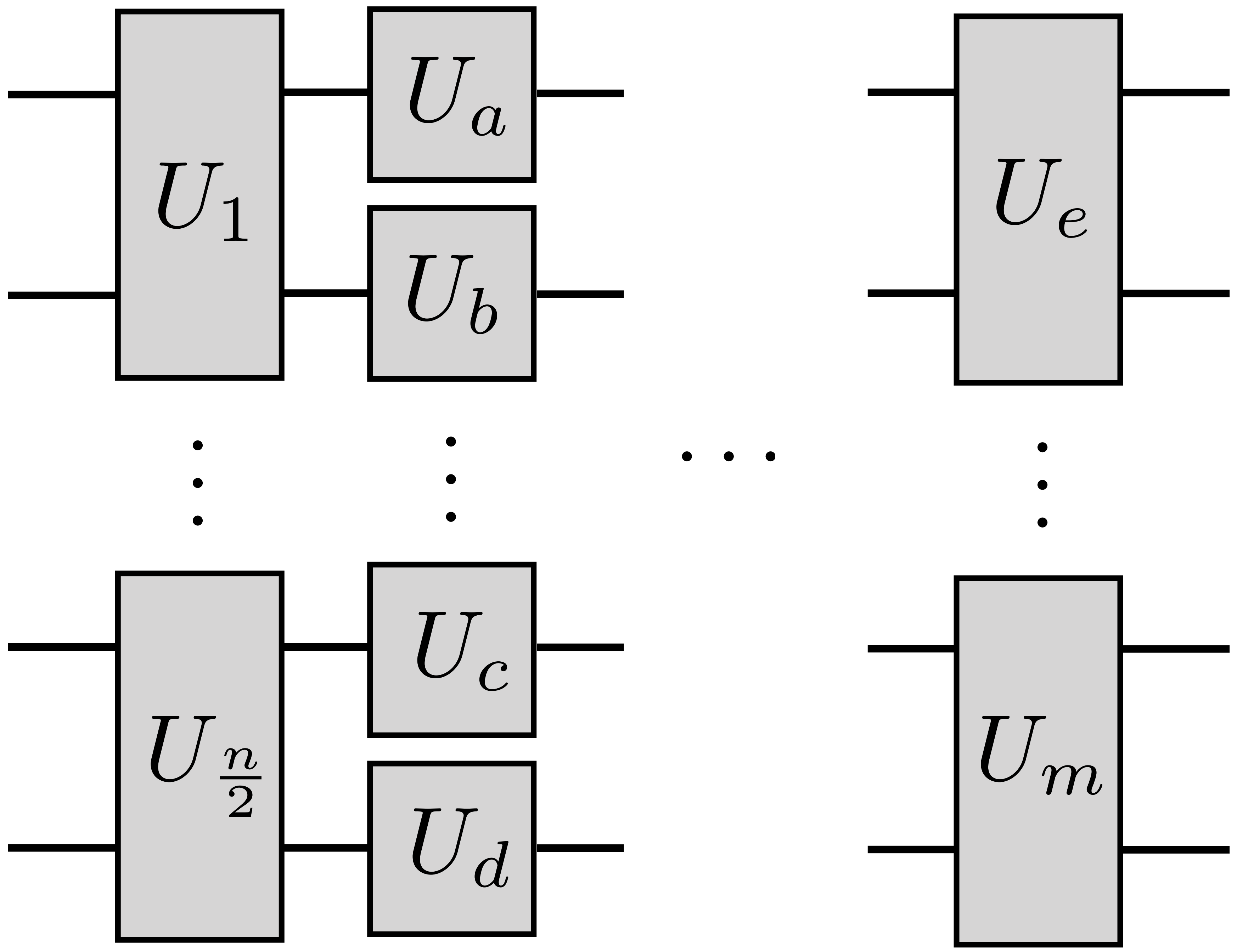
1. Randomly pick $U_1, \ldots, U_m$.
2. Pick input state, e.g., $|0^n\rangle$.

Prover

(E.g., Google)

Random Circuit Sampling

The lead-candidate used by Google's Sycamore



Challenger

(E.g., Quantum skeptics)

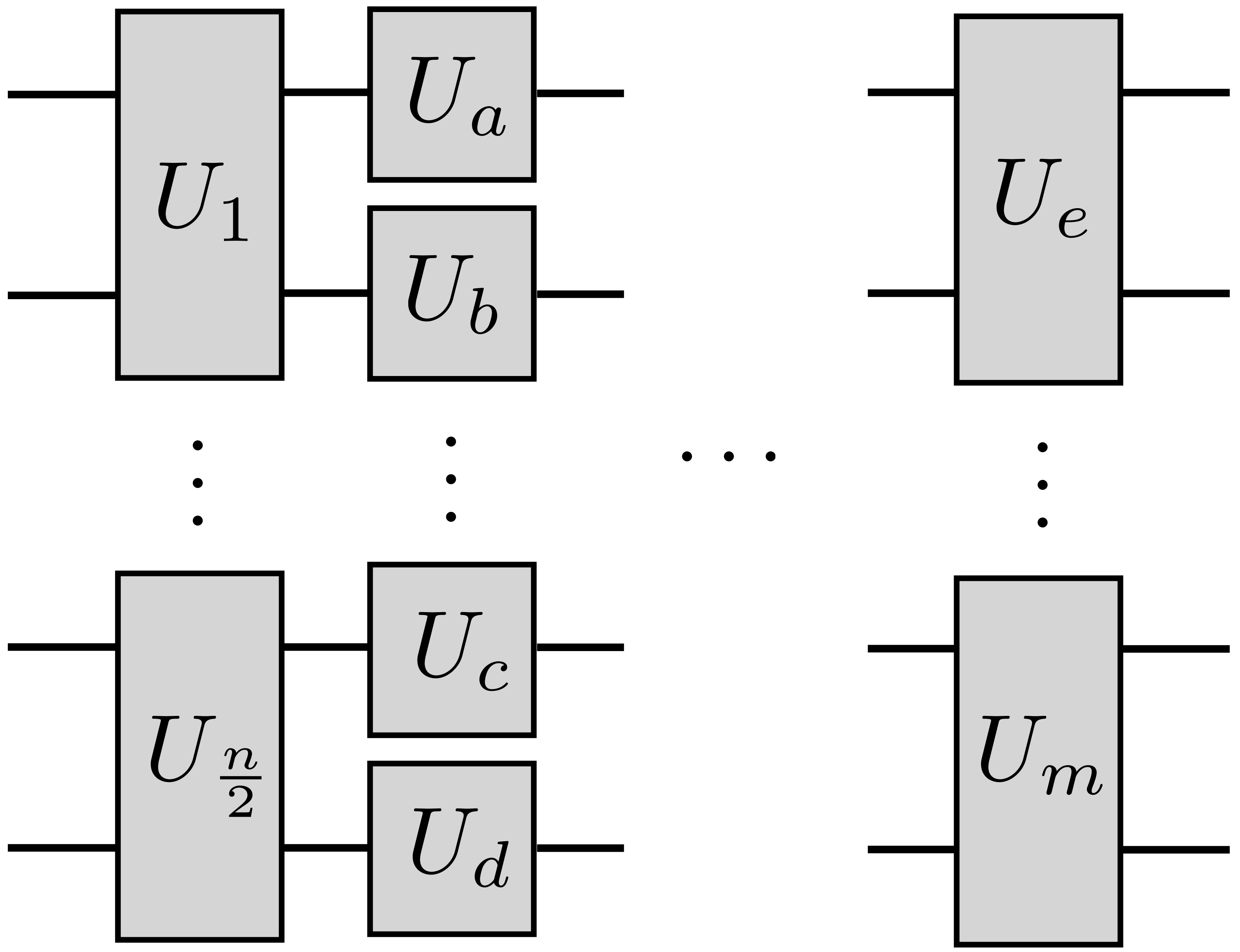
Prover

(E.g., Google)

1. Randomly pick $U_1, \ldots, U_m$.
2. Pick input state, e.g., $|0^n\rangle$.
3. Sample many strings from the output distribution q_C.

Random Circuit Sampling

The lead-candidate used by Google's Sycamore



Challenger

(E.g., Quantum skeptics)

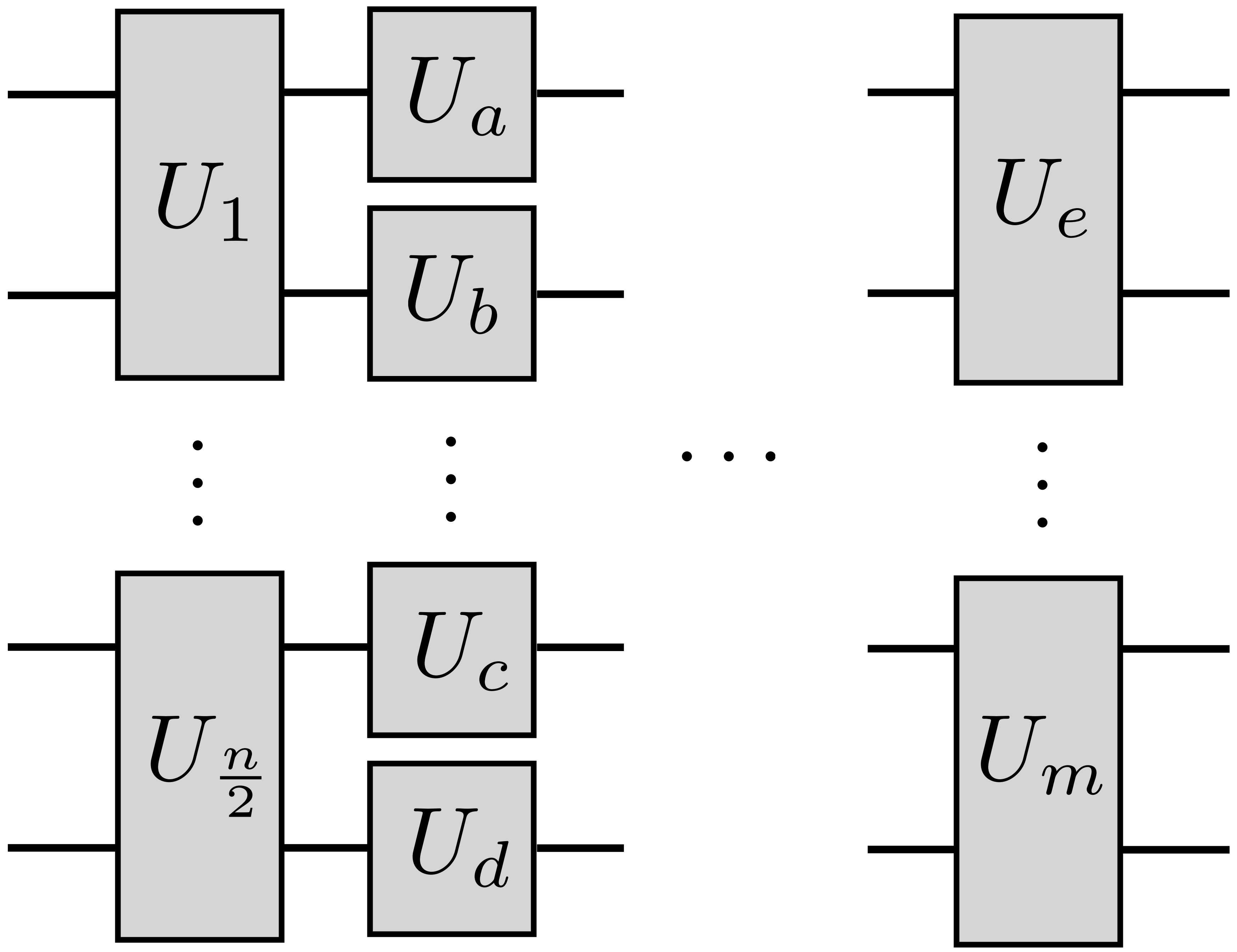
Prover

(E.g., Google)

1. Randomly pick $U_1, \ldots, U_m$.
2. Pick input state, e.g., $|0^n\rangle$.
3. Sample many strings from the output distribution q_C.
4. Verify samples follow q_C.

Random Circuit Sampling

The lead-candidate used by Google's Sycamore



Challenger

(E.g., Quantum skeptics)

Prover

(E.g., Google)

1. Randomly pick $U_1, \ldots, U_m$.
2. Pick input state, e.g., $|0^n\rangle$.
3. Sample many strings from the output distribution q_C.
4. Verify samples follow q_C.

Intuition: Without quantumness, the prover requires exponential time!?

How to Verify?

Challenger

(E.g., Quantum skeptics)

Prover

(E.g., Google)

1. Randomly pick $U_1, \ldots, U_m$.
2. Pick input state, e.g., $|0^n\rangle$.
3. Sample many strings from the output distribution q_C.
4. Verify samples follow q_C.

How to Verify?

Challenger
(E.g., Quantum skeptics)

Prover
(E.g., Google)

1. Randomly pick $U_1, \ldots, U_m$.
2. Pick input state, e.g., $|0^n\rangle$.
3. Sample many strings from the output distribution q_C.
4. Verify samples follow q_C.

- **Efficiency:** The verification should be scalable.

How to Verify?

Challenger

(E.g., Quantum skeptics)

Prover

(E.g., Google)

1. Randomly pick $U_1, \ldots, U_m$.
2. Pick input state, e.g., $|0^n\rangle$.
3. Sample many strings from the output distribution q_C.
4. Verify samples follow q_C.

- **Efficiency:** The verification should be scalable.
- **Completeness:** If the Prover's distribution is close to q_C, then accept w.h.p.

How to Verify?

Challenger

(E.g., Quantum skeptics)

Prover

(E.g., Google)

1. Randomly pick $U_1, \ldots, U_m$.
2. Pick input state, e.g., $|0^n\rangle$.
3. Sample many strings from the output distribution q_C.
4. Verify samples follow q_C.

- **Efficiency:** The verification should be scalable.
- **Completeness:** If the Prover's distribution is close to q_C, then accept w.h.p.
- **Soundness:** If the distribution came from a classical device, then reject w.h.p.

Linear Cross-Entropy Benchmarking

A statistic for verifying RCS-based quantum supremacy

Linear Cross-Entropy Benchmarking

A statistic for verifying RCS-based quantum supremacy

Definition (Linear XEB).

Linear Cross-Entropy Benchmarking

A statistic for verifying RCS-based quantum supremacy

Definition (Linear XEB).

Let C be an n-qubit quantum circuit and p be a distribution over n-bit strings. The *Linear XEB fidelity* of p w.r.t. C is defined as

Linear Cross-Entropy Benchmarking

A statistic for verifying RCS-based quantum supremacy

Definition (Linear XEB).

Let C be an n-qubit quantum circuit and p be a distribution over n-bit strings. The *Linear XEB fidelity* of p w.r.t. C is defined as

$$\mathcal{F}_C(p) = \mathop{\mathbb{E}}_{x \sim p}\left[2^n q_C(x) - 1\right].$$

Linear Cross-Entropy Benchmarking

A statistic for verifying RCS-based quantum supremacy

Definition (Linear XEB).

Let C be an n-qubit quantum circuit and p be a distribution over n-bit strings. The *Linear XEB fidelity* of p w.r.t. C is defined as

$$\mathcal{F}_C(p) = \mathop{\mathbb{E}}_{x \sim p}\left[2^n q_C(x) - 1\right].$$

- **Efficiency:** Need to compute the marginal probability $q_C(x)$.
  - Exponential time 😢

Linear Cross-Entropy Benchmarking

A statistic for verifying RCS-based quantum supremacy

Definition (Linear XEB).

Let C be an n-qubit quantum circuit and p be a distribution over n-bit strings. The *Linear XEB fidelity* of p w.r.t. C is defined as

$$\mathcal{F}_C(p) = \mathop{\mathbb{E}}_{x \sim p} \left[2^n q_C(x) - 1 \right].$$

- **Efficiency:** Need to compute the marginal probability $q_C(x)$.
  - Exponential time 😢
- **Completeness:** When C is random enough: $\mathcal{F}_C(q_C) = 1$.

Linear Cross-Entropy Benchmarking

A statistic for verifying RCS-based quantum supremacy

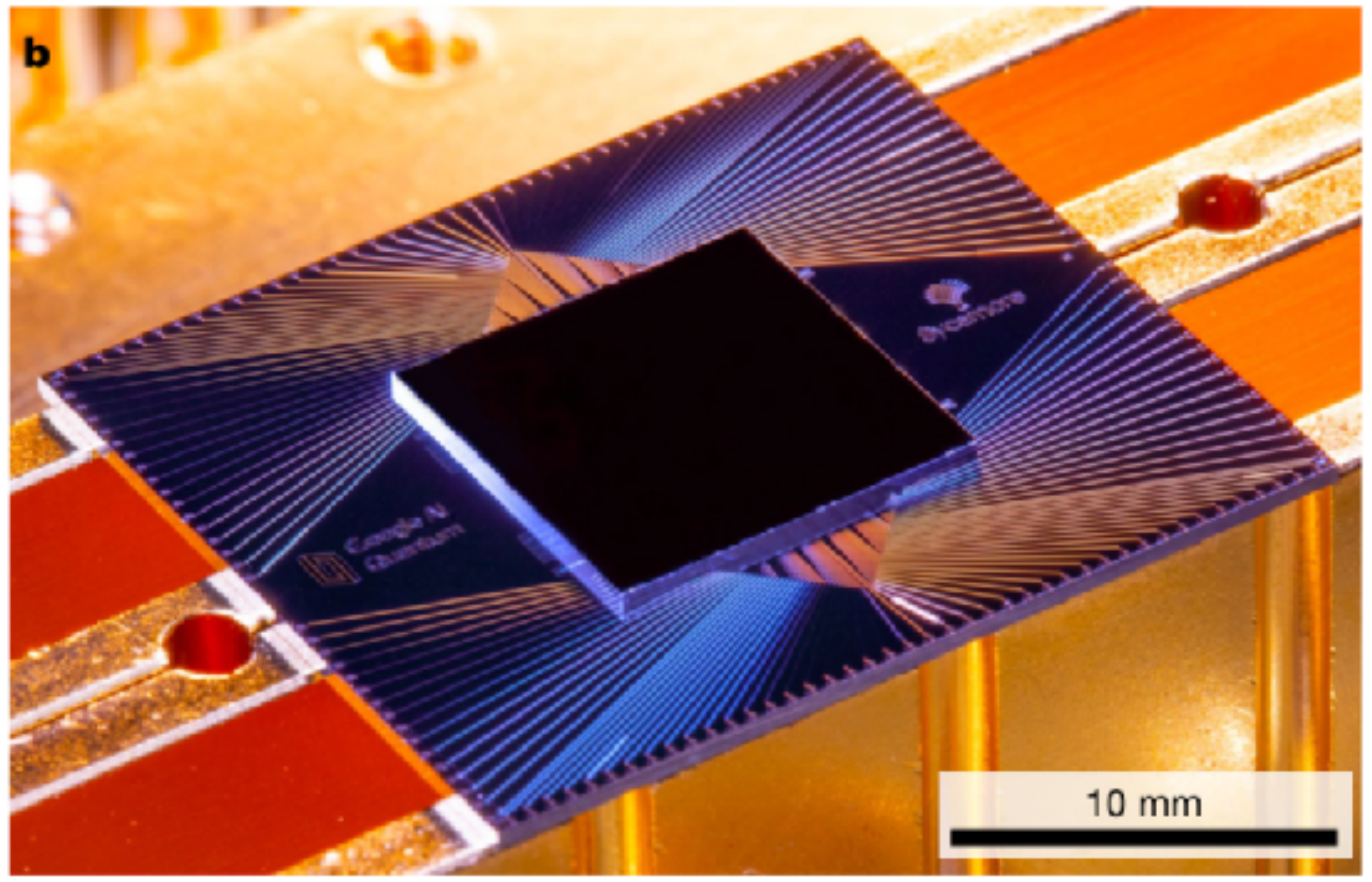
Definition (Linear XEB).

Let C be an n-qubit quantum circuit and p be a distribution over n-bit strings. The *Linear XEB fidelity* of p w.r.t. C is defined as

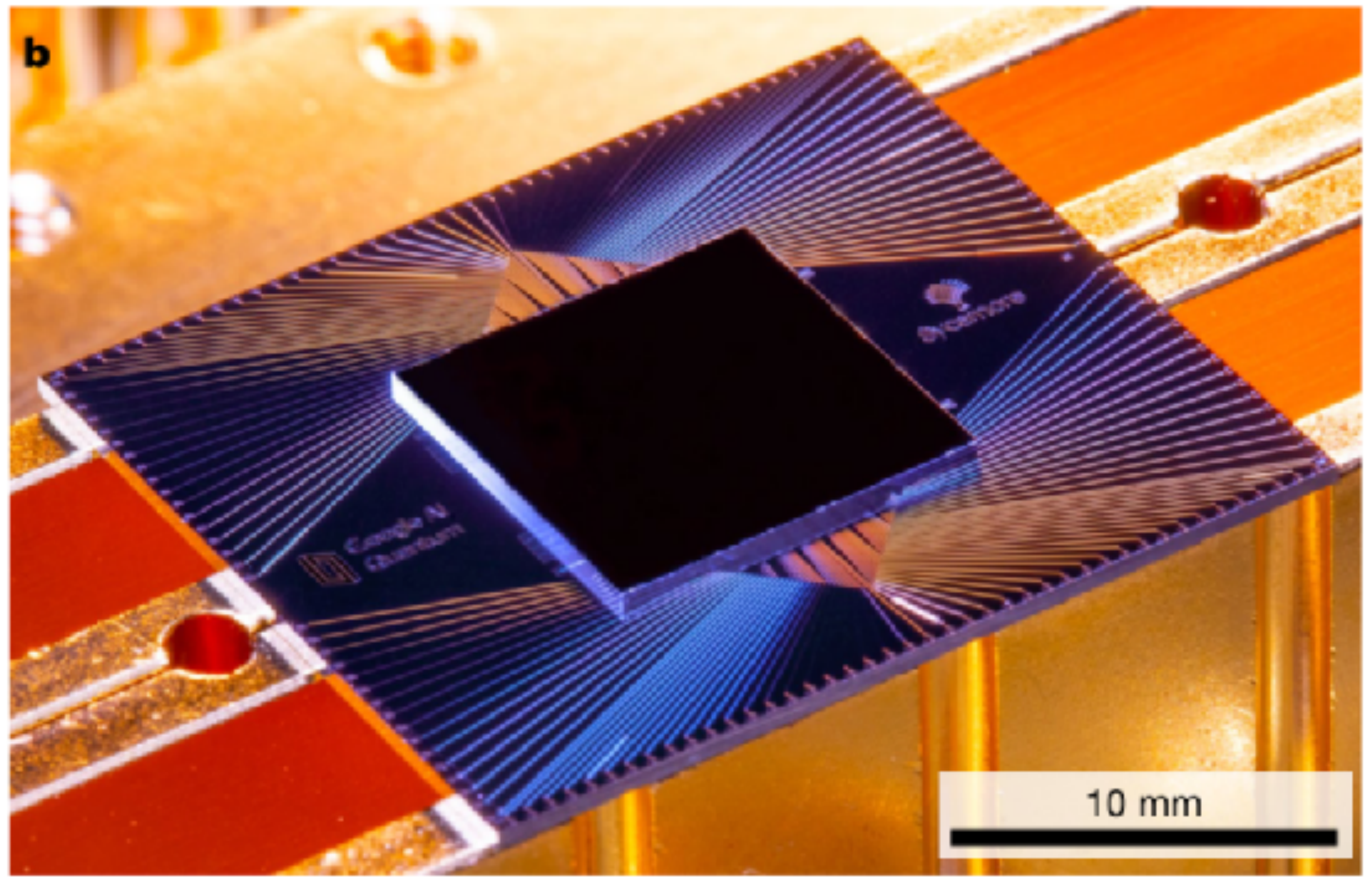
$$\mathcal{F}_C(p) = \mathop{\mathbb{E}}_{x \sim p}\left[2^n q_C(x) - 1\right].$$

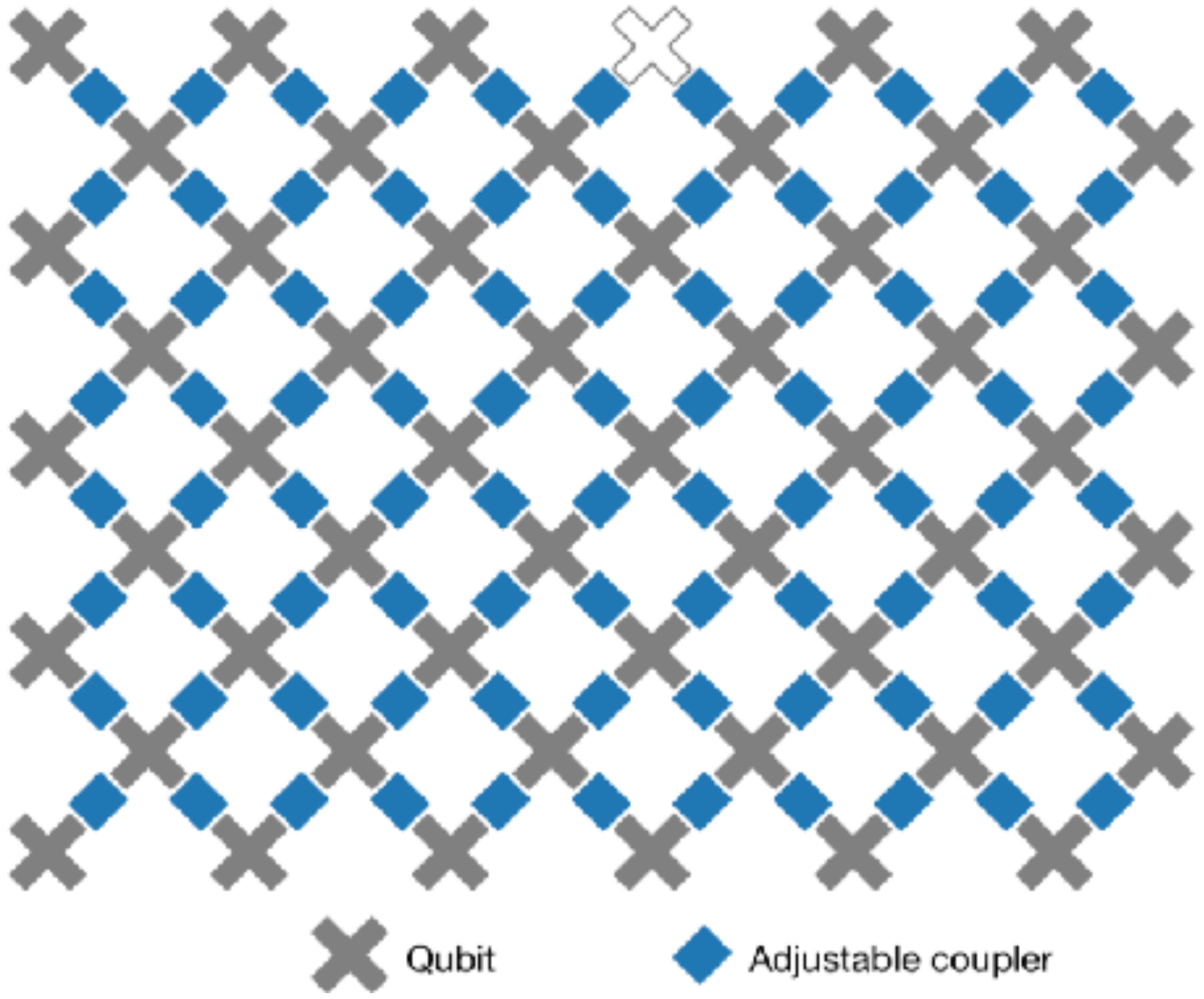
- **Efficiency:** Need to compute the marginal probability $q_C(x)$.
  - Exponential time 😢
- **Completeness:** When C is random enough: $\mathcal{F}_C(q_C) = 1$.
- **Soundness:** When p is the uniform distribution: $\mathcal{F}_C(p) \approx 0$.

Google's Quantum Supremacy

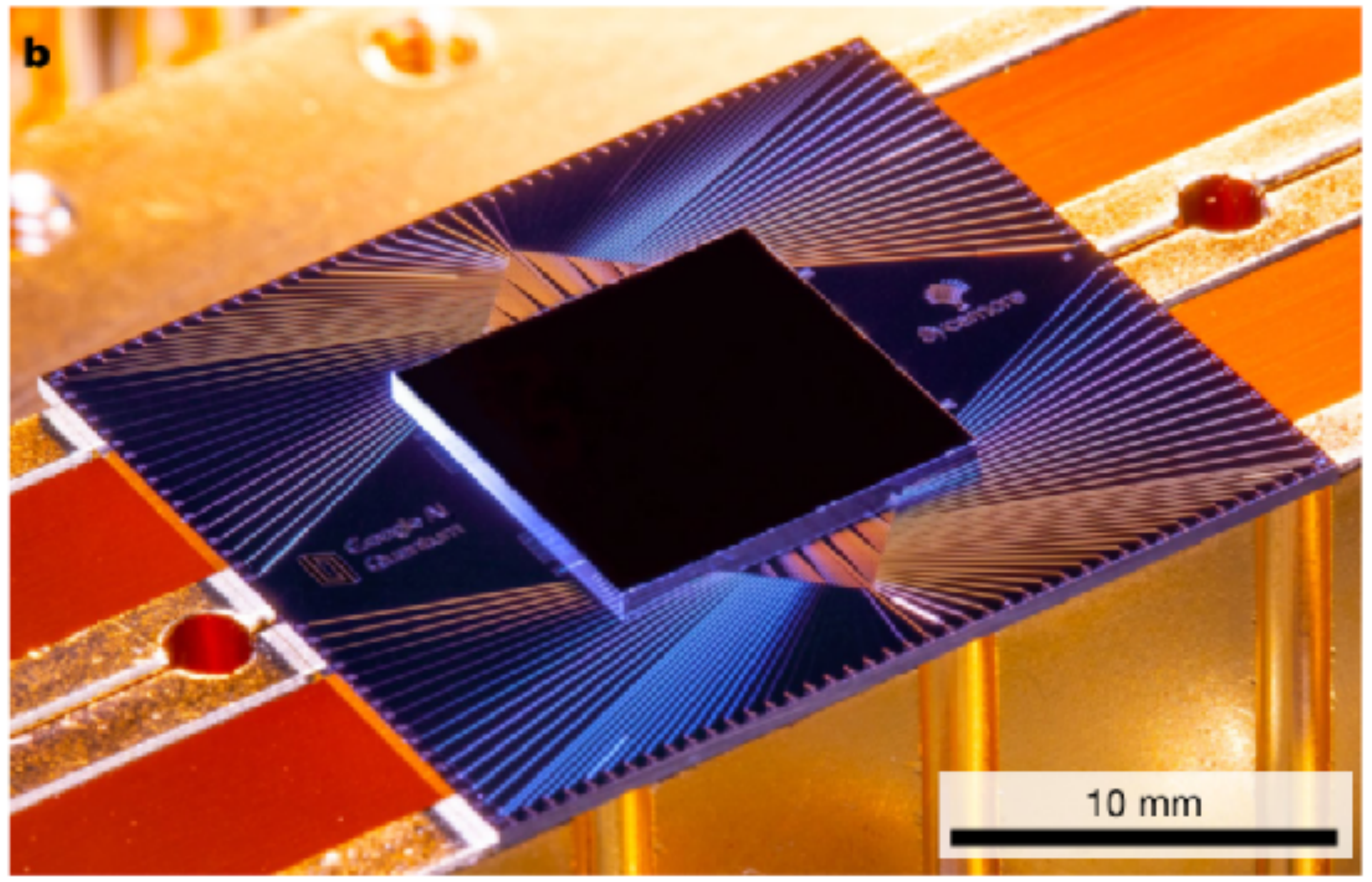


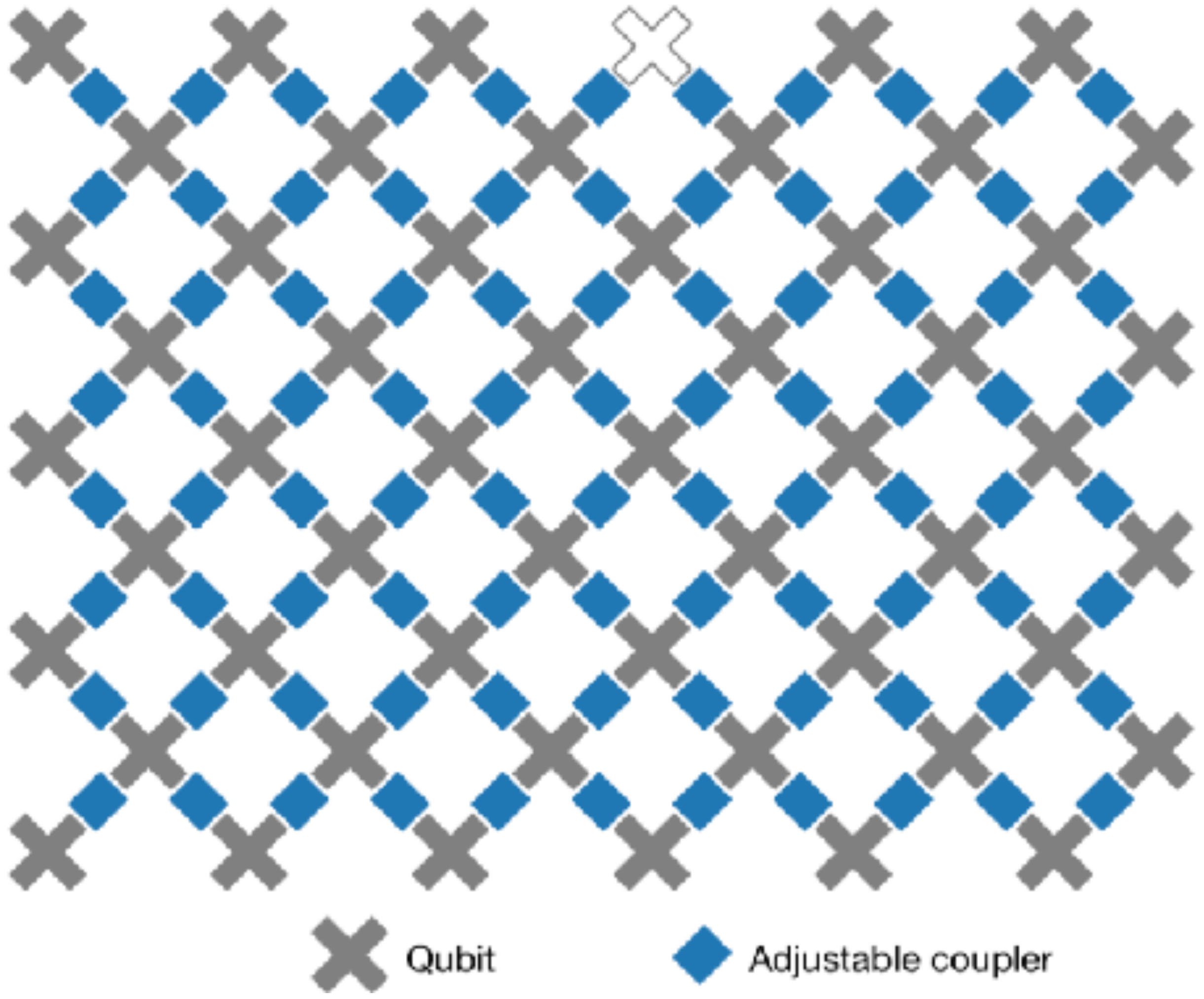
Google's Quantum Supremacy





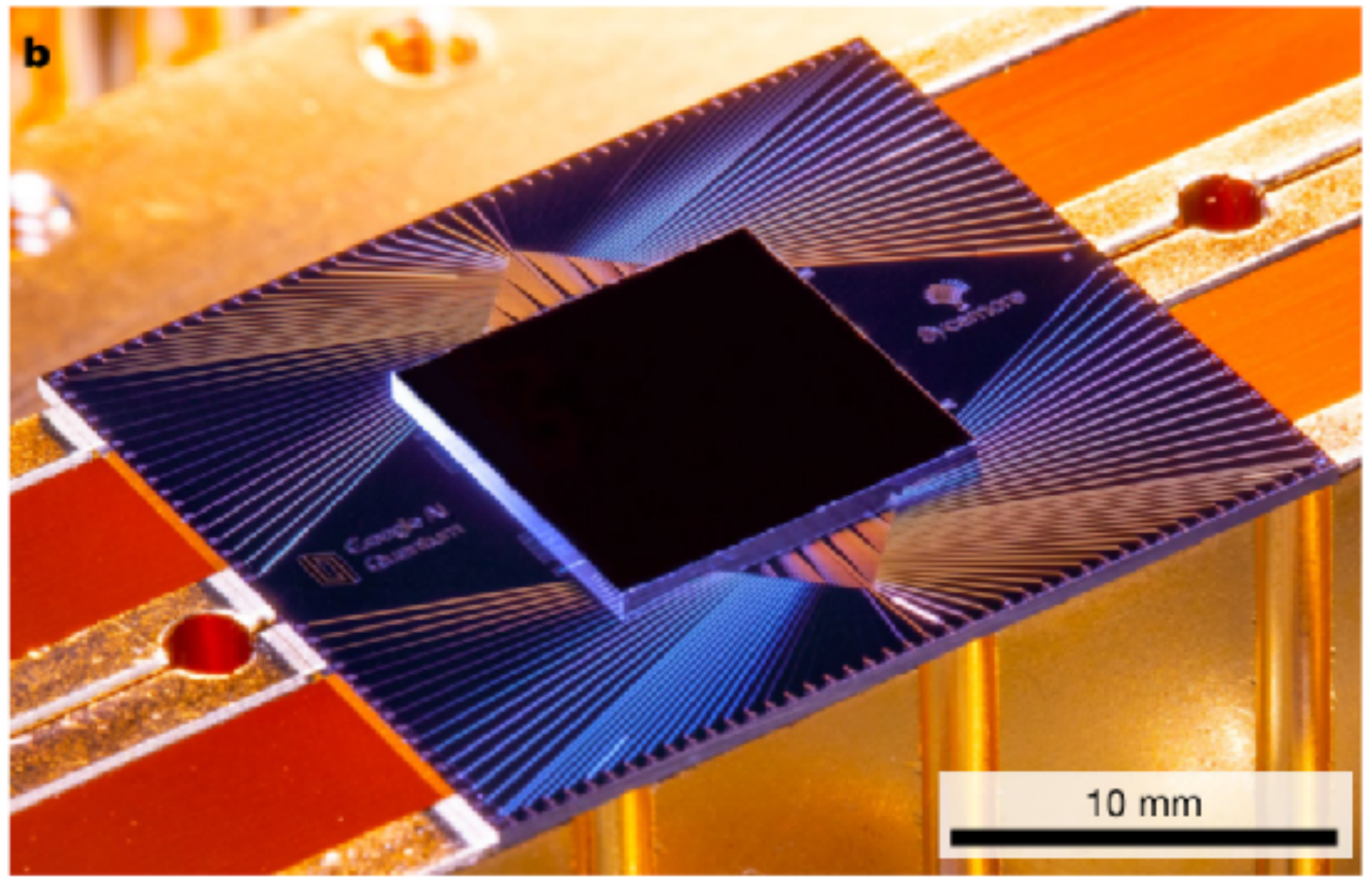
Google's Quantum Supremacy

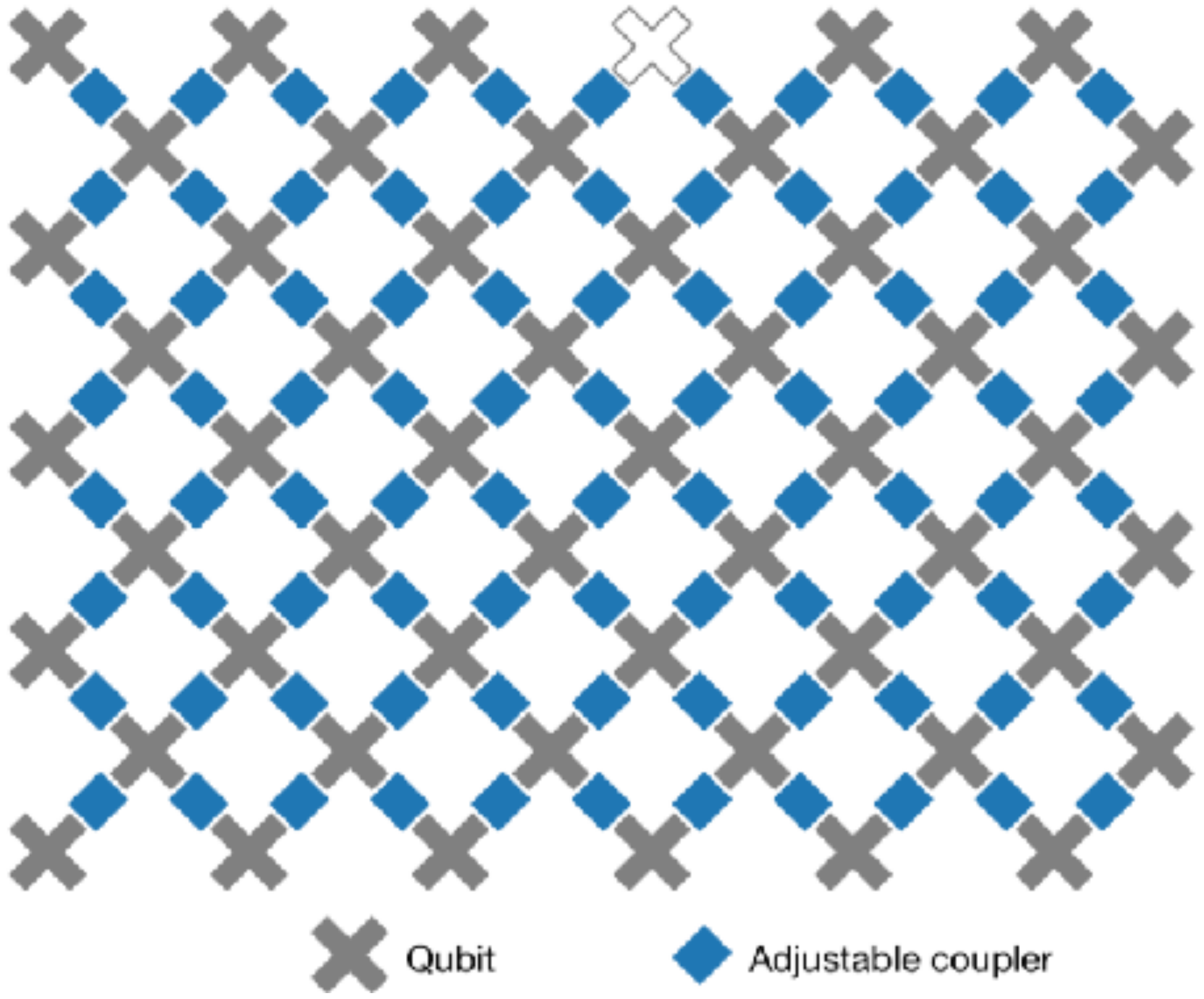




- 2D nearest neighbors circuit.

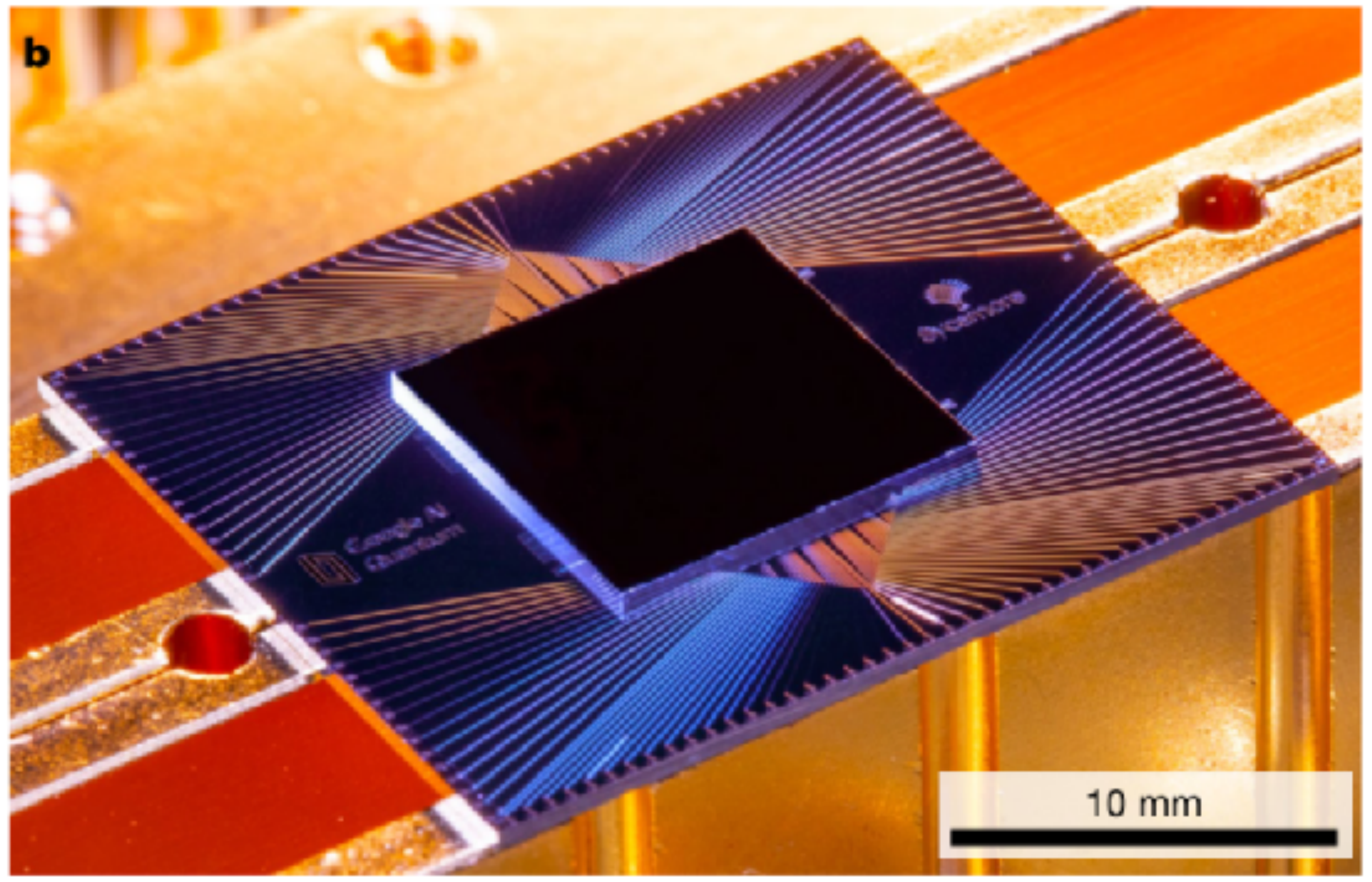
Google's Quantum Supremacy

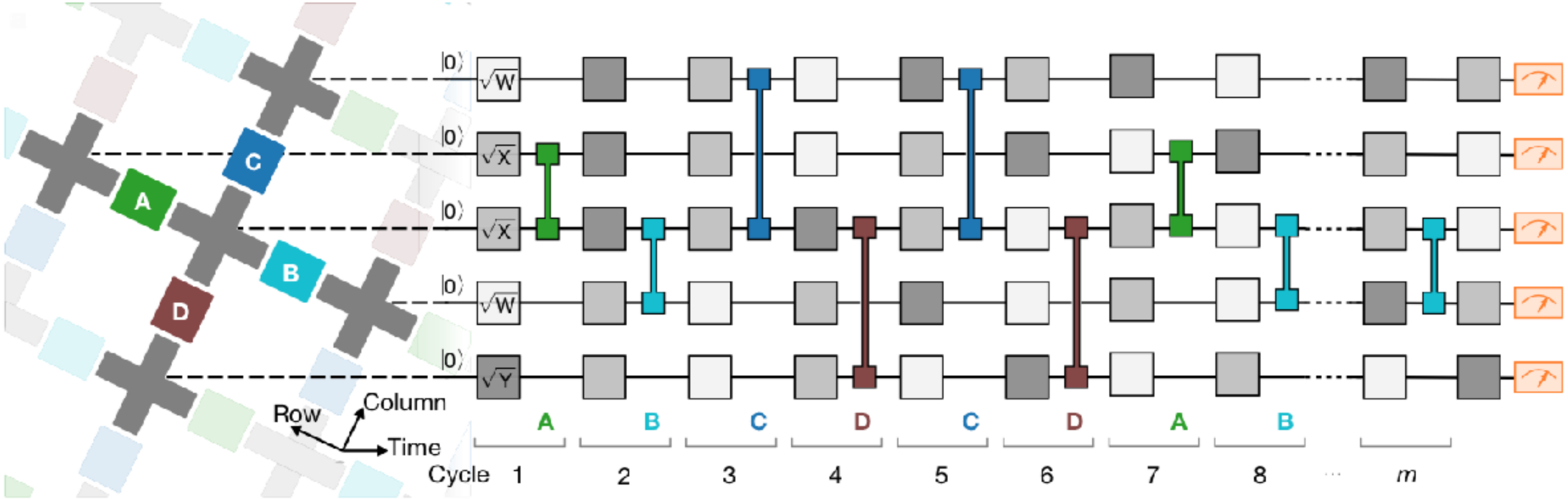


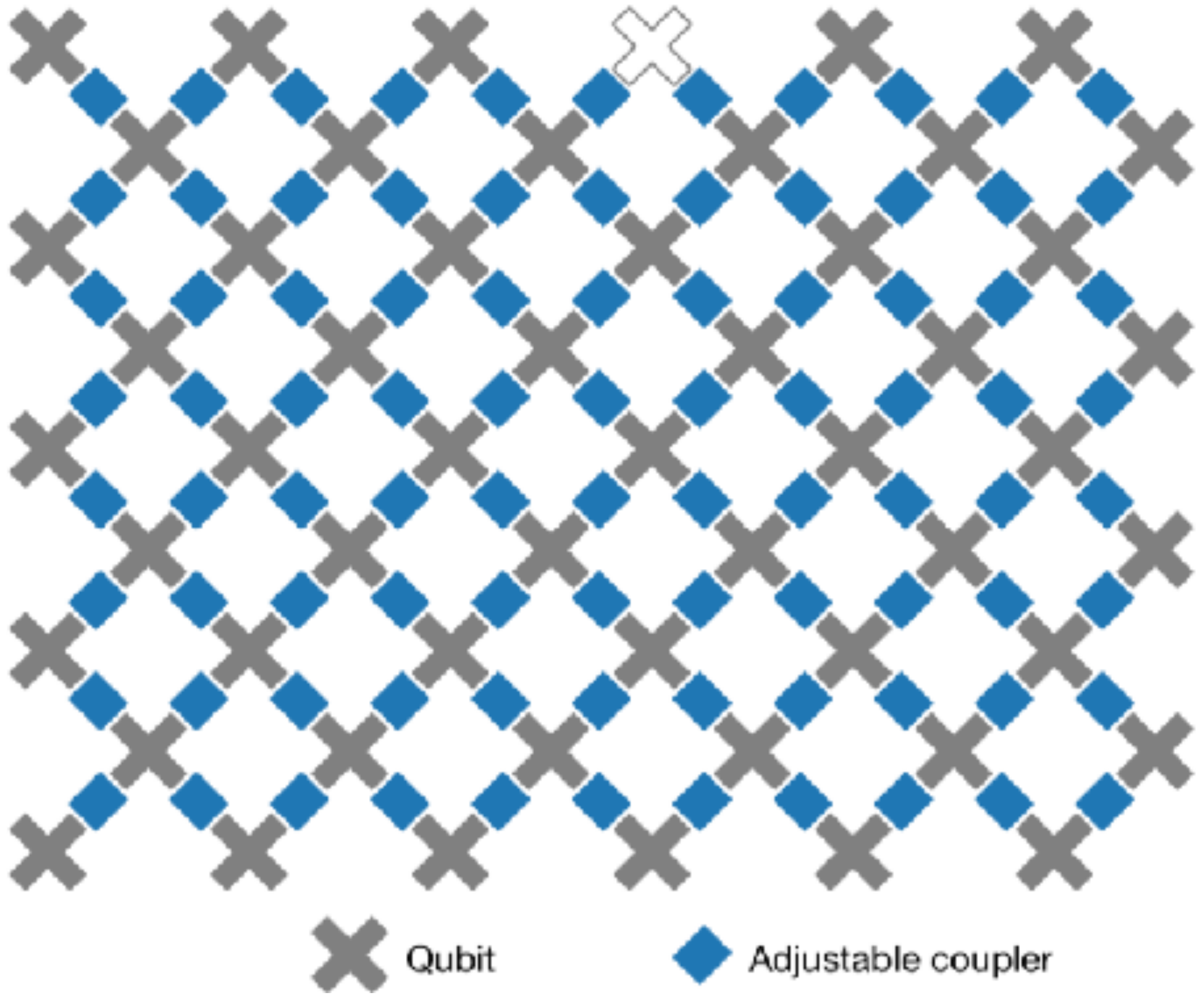


- 2D nearest neighbors circuit.
- #qubits: from 12 to 53.

Google's Quantum Supremacy

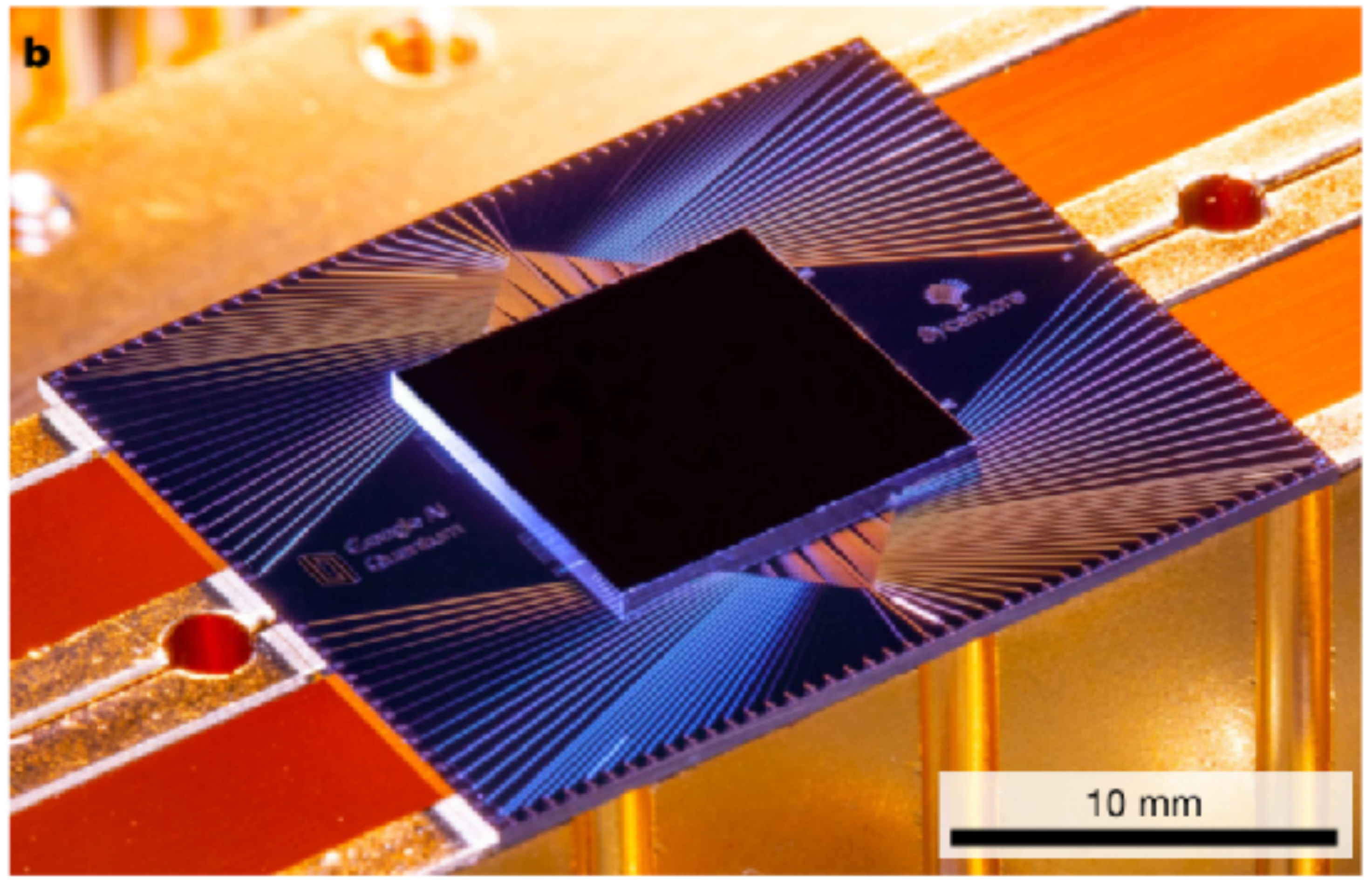


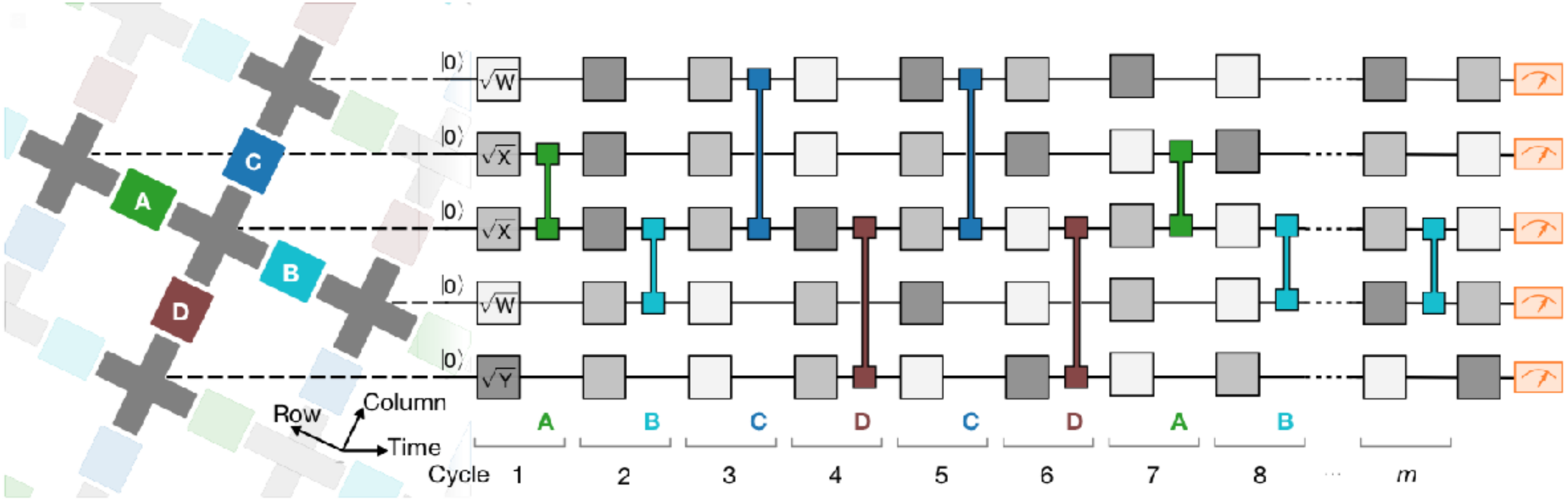


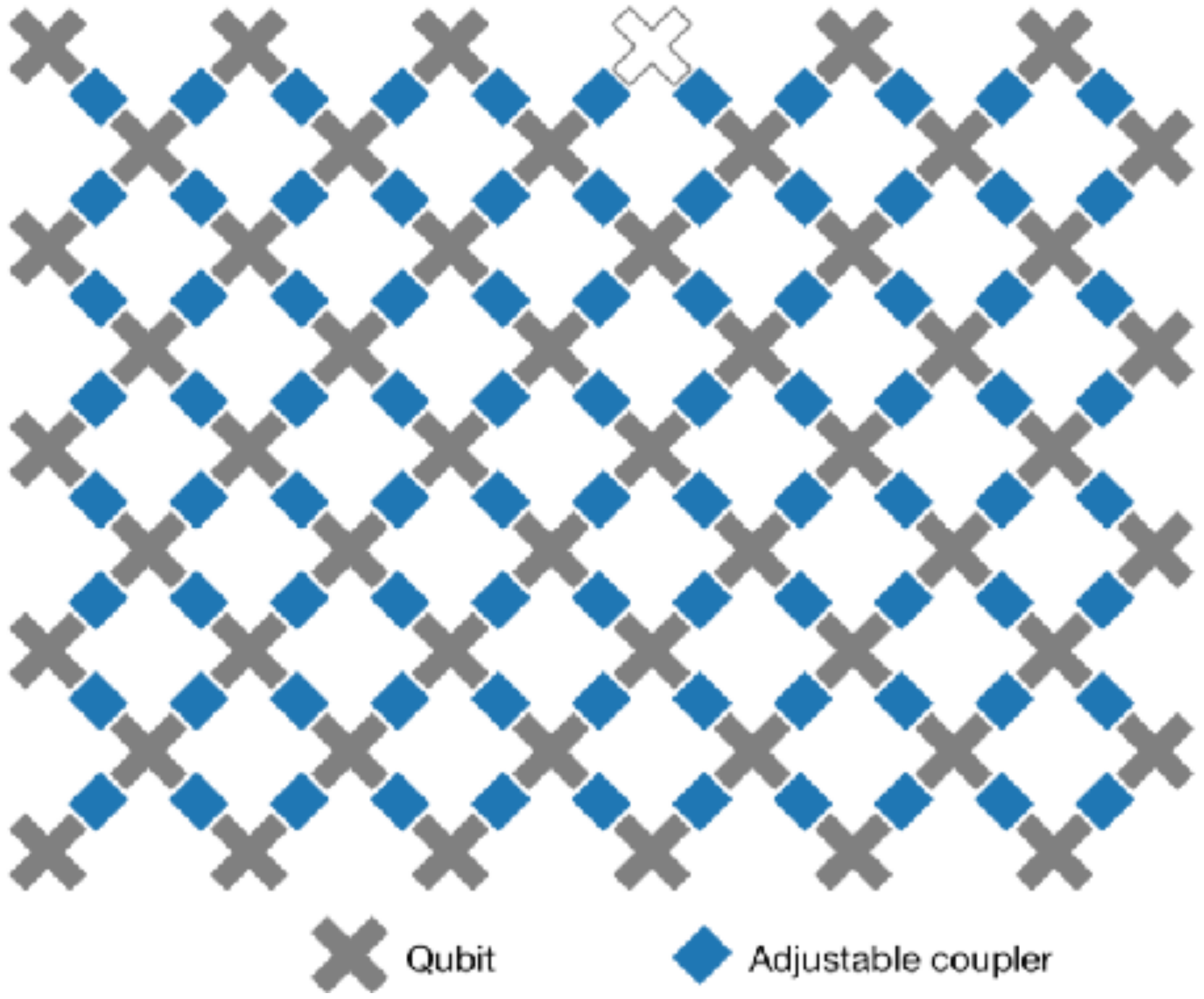


- 2D nearest neighbors circuit.
- #qubits: from 12 to 53.

Google's Quantum Supremacy





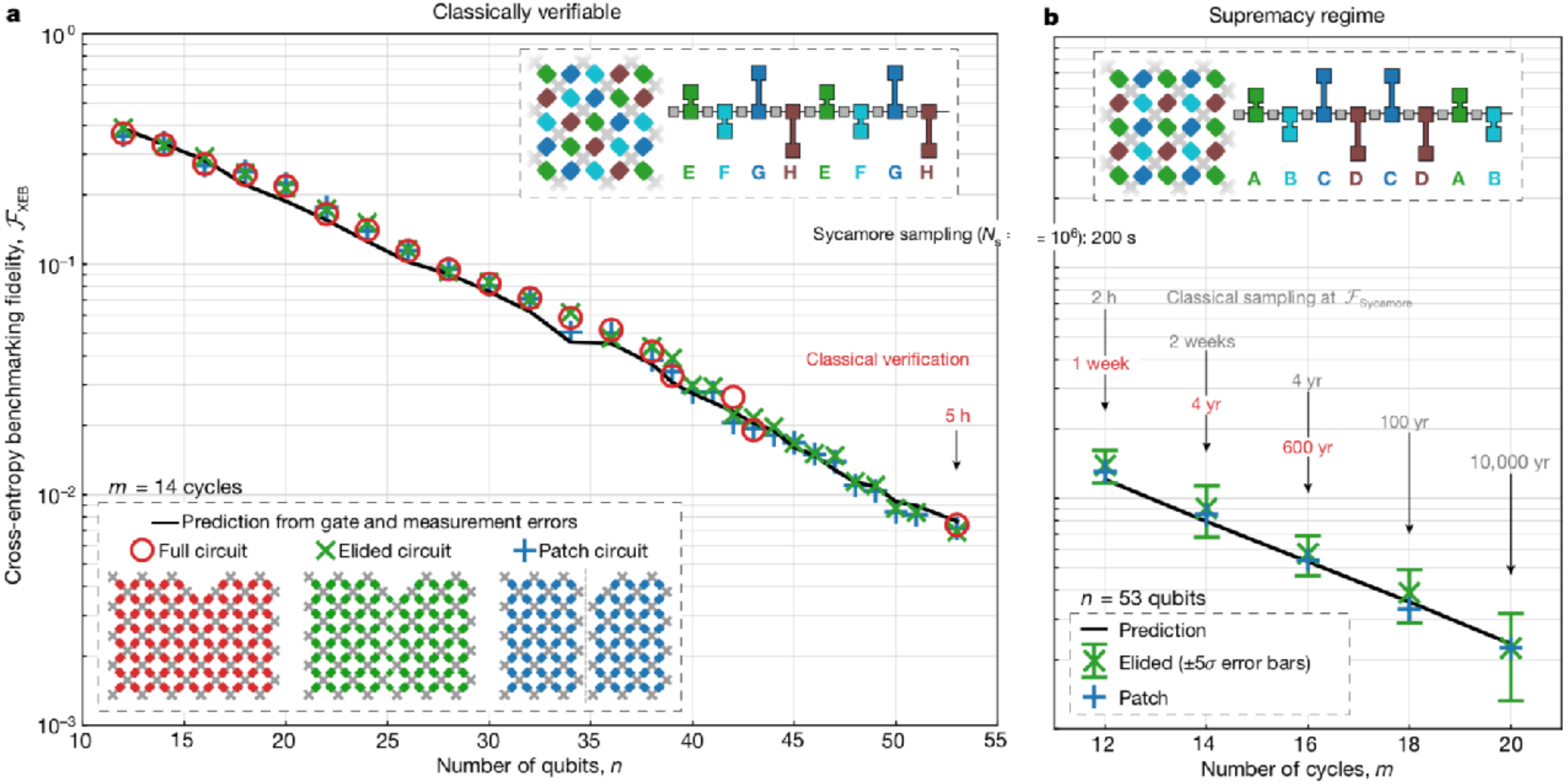


- 2D nearest neighbors circuit.
- #qubits: from 12 to 53.
- Depth: from 14 to 20.

* We use *m* for # gates but in Google's paper *m* denotes depth.

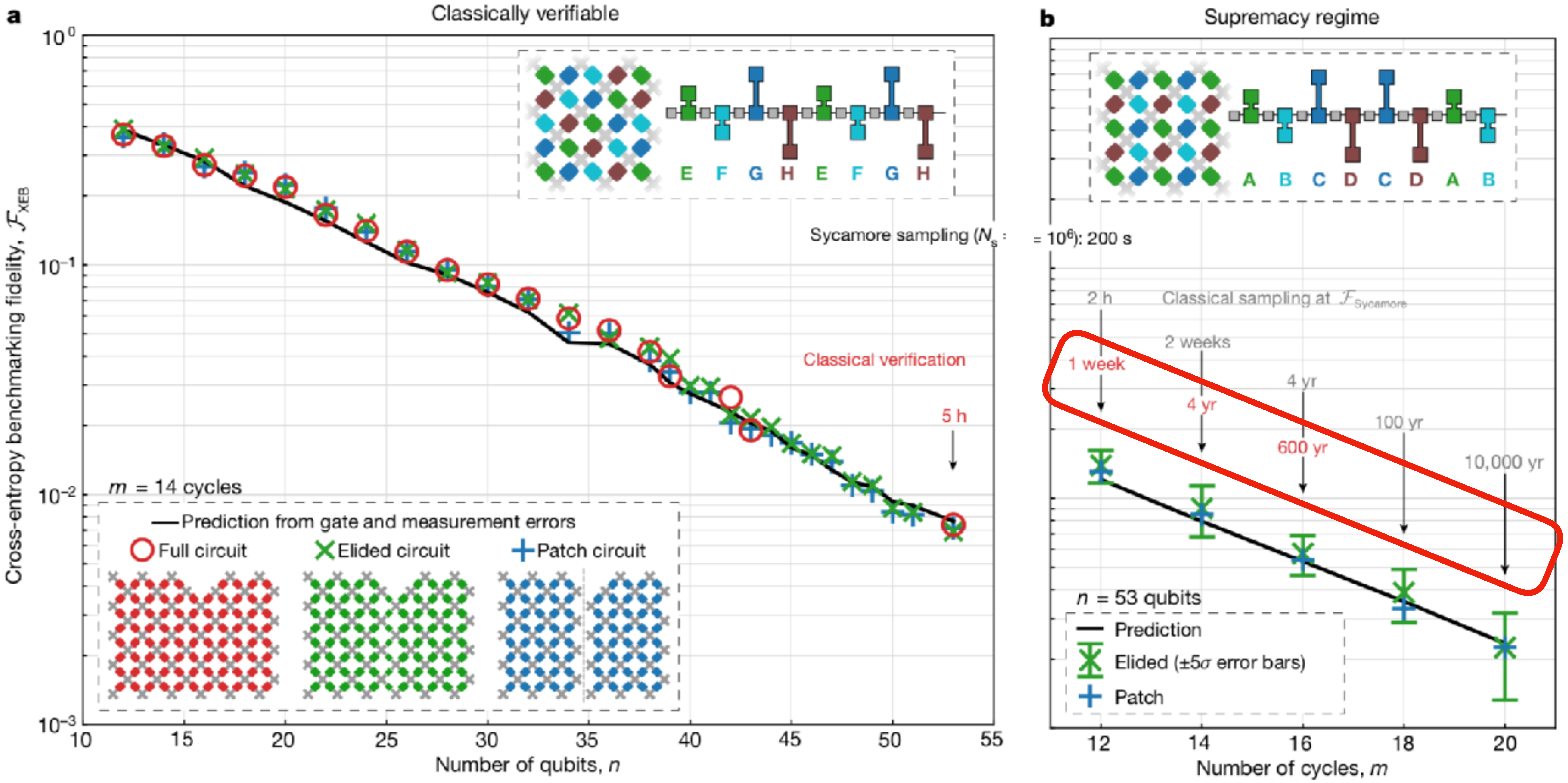
Google’s Quantum Supremacy

Google’s Quantum Supremacy



Classically verifiable Extrapolation

Google's Quantum Supremacy

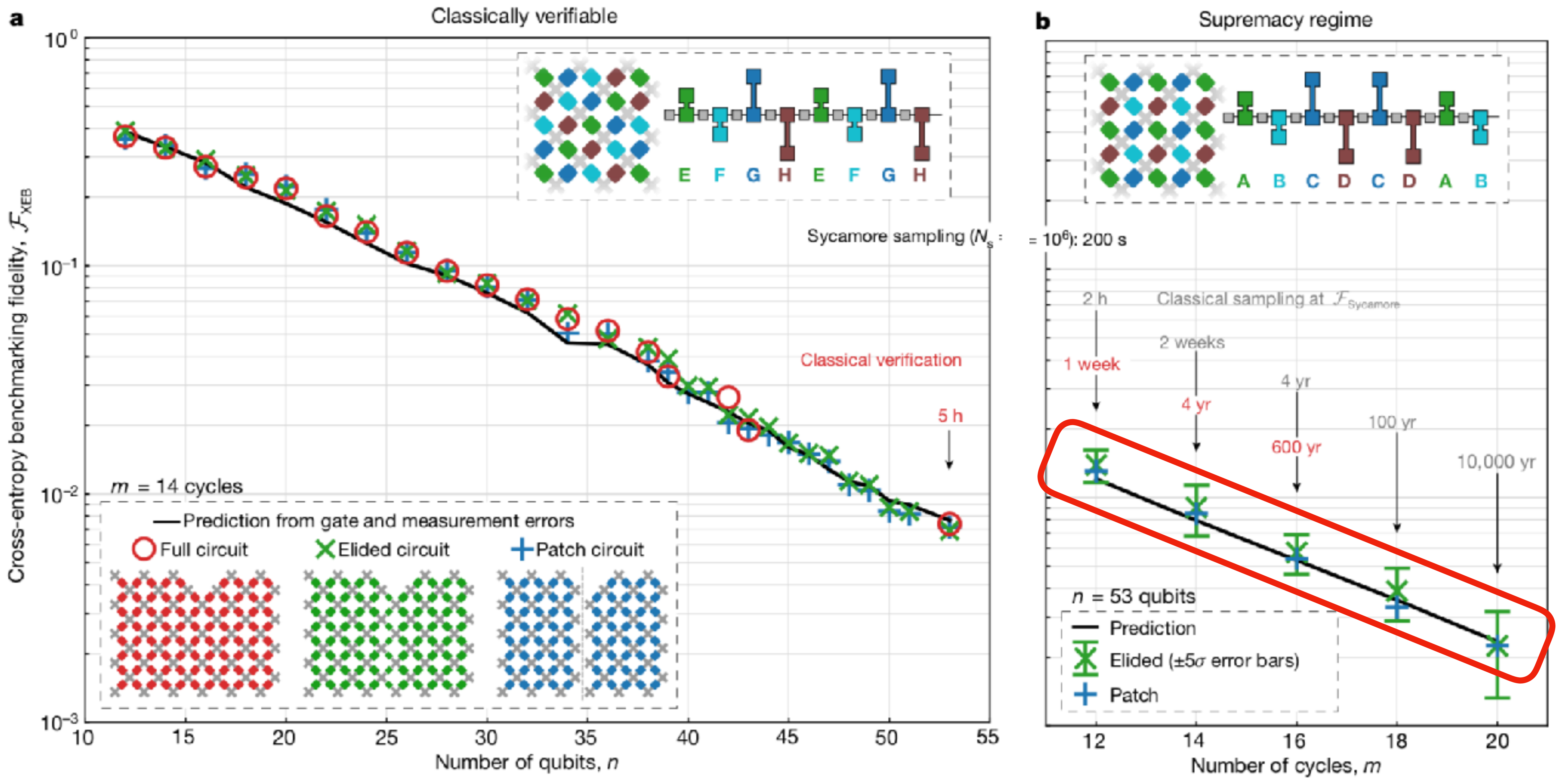


- They can only verify up to depth-14 so the rest were extrapolated.

Classically verifiable

Extrapolation

Google's Quantum Supremacy

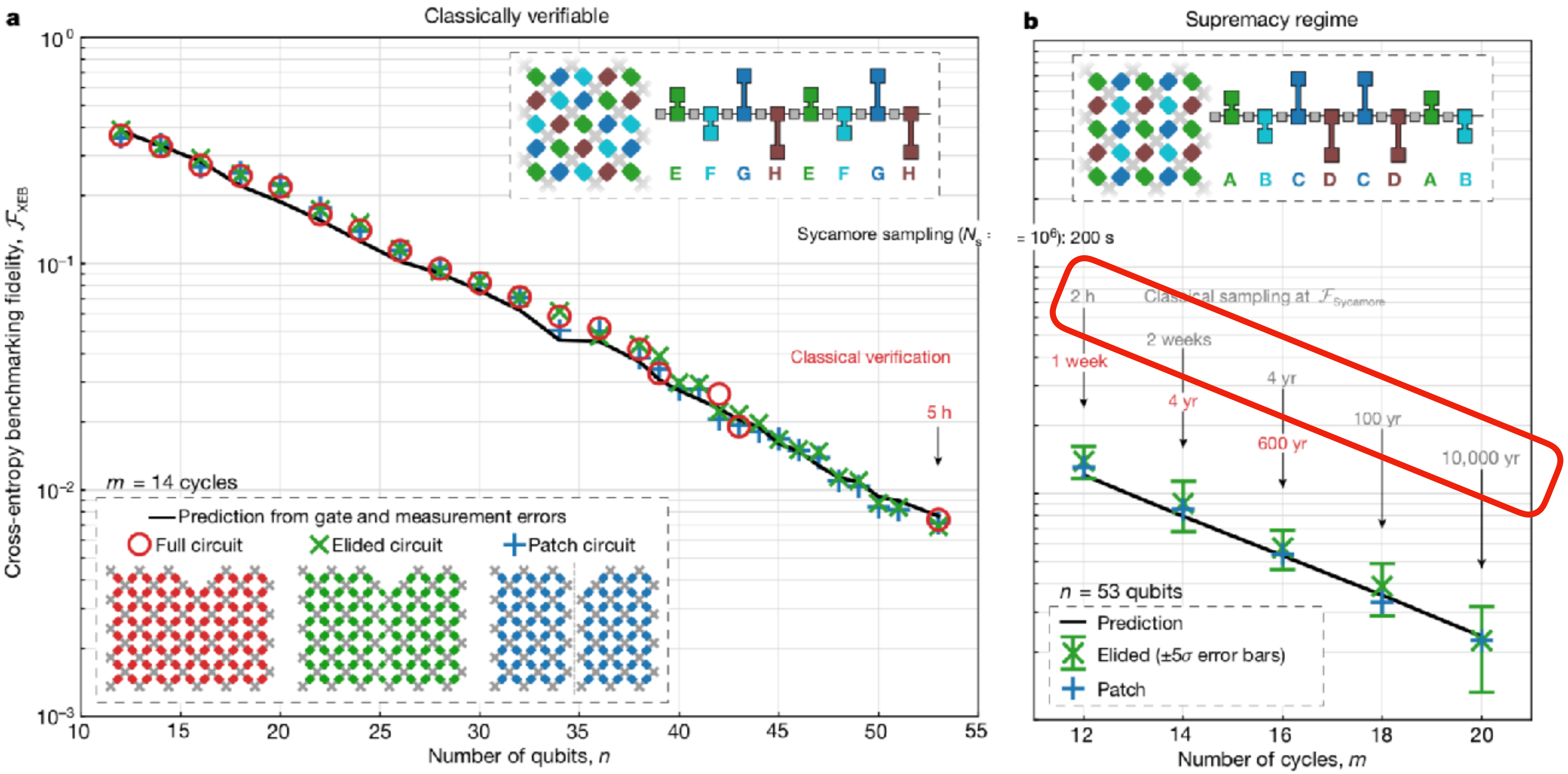


Classically verifiable

Extrapolation

- They can only verify up to depth-14 so the rest were extrapolated.
- Achieve $(2.24 \pm 0.21) \times 10^{-3}$ linear XEB when 53-qubit and depth-20 in few minutes.

Google's Quantum Supremacy



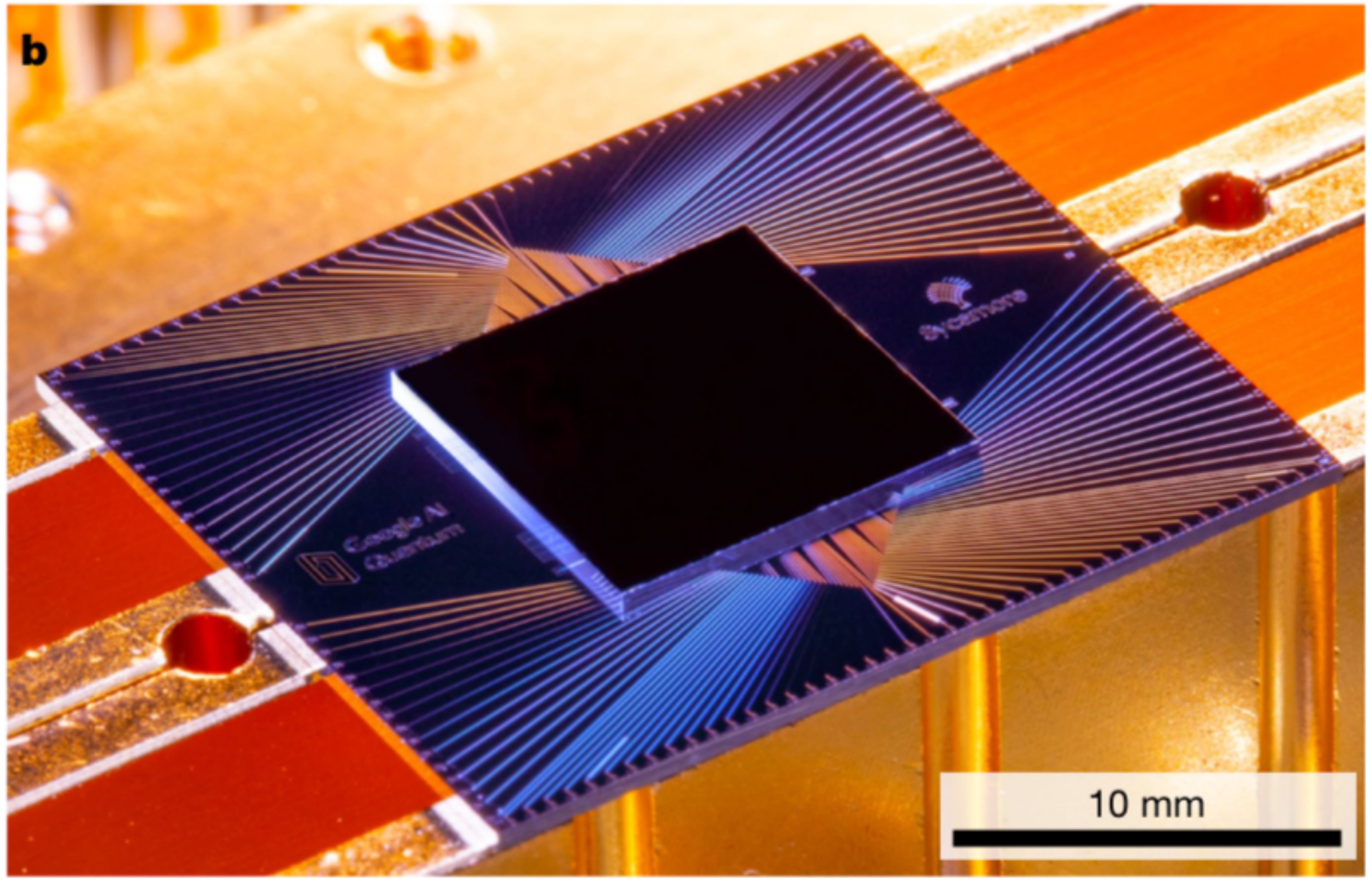
Classically verifiable

Extrapolation

- They can only verify up to depth-14 so the rest were extrapolated.
- Achieve $(2.24 \pm 0.21) \times 10^{-3}$ linear XEB when 53-qubit and depth-20 in few minutes.
- They conjectured classical sampling takes 10,000 years.

Does Google refute ECTT?

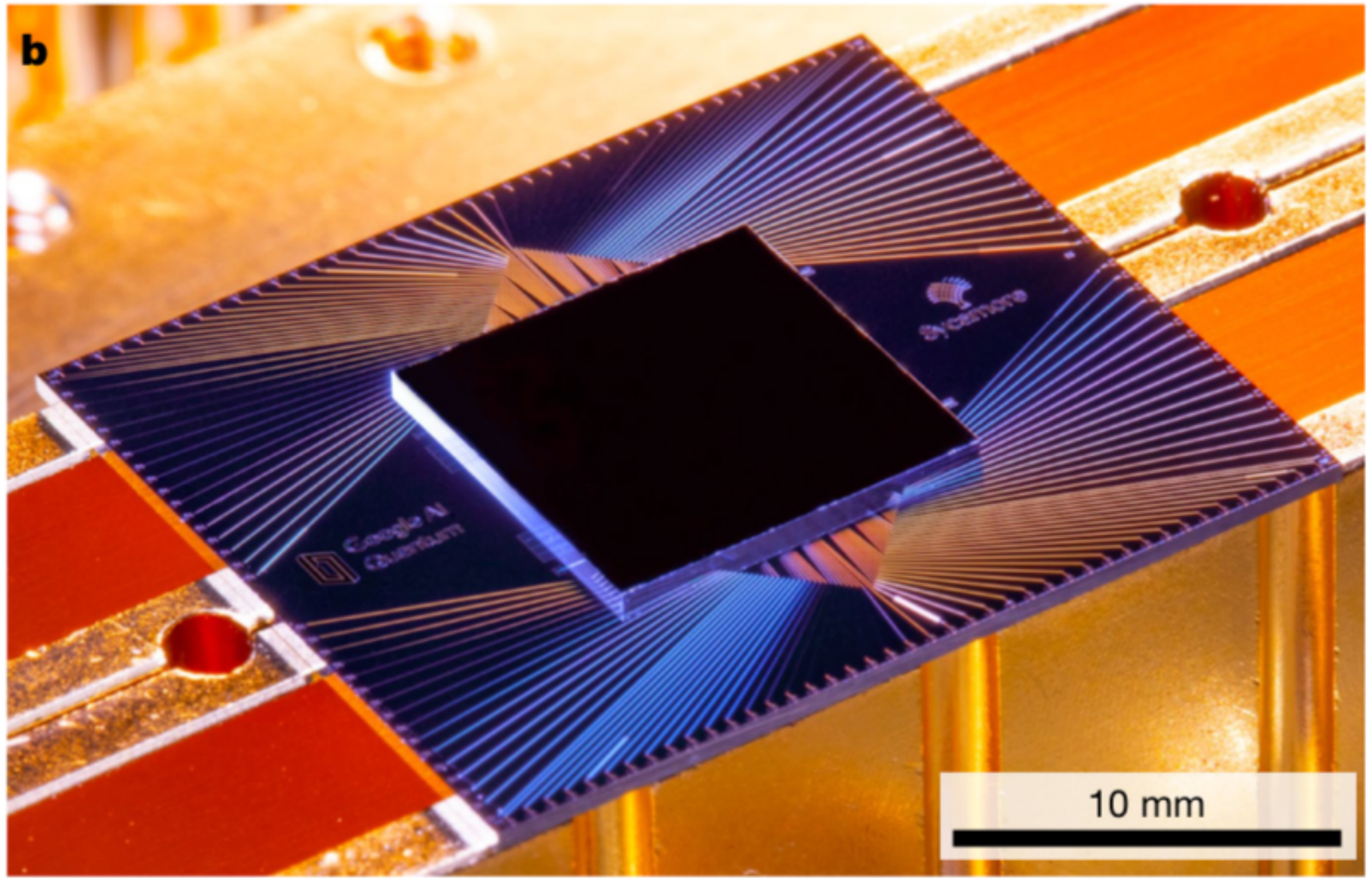
Google indeed had made significant progress, but…



Does Google refute ECTT?

Google indeed had made significant progress, but…

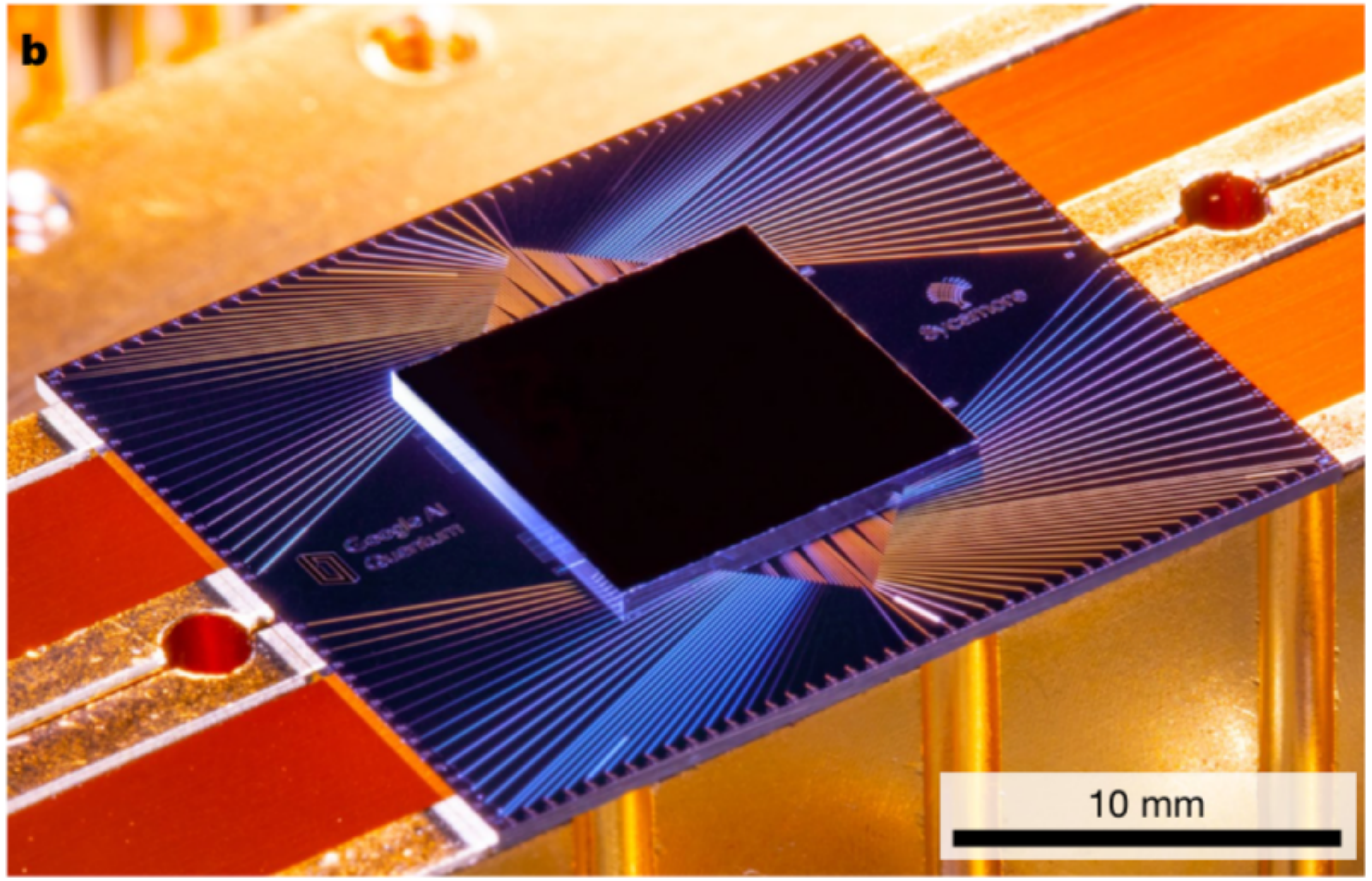
- The classical hardness is not solid.



Does Google refute ECTT?

Google indeed had made significant progress, but…

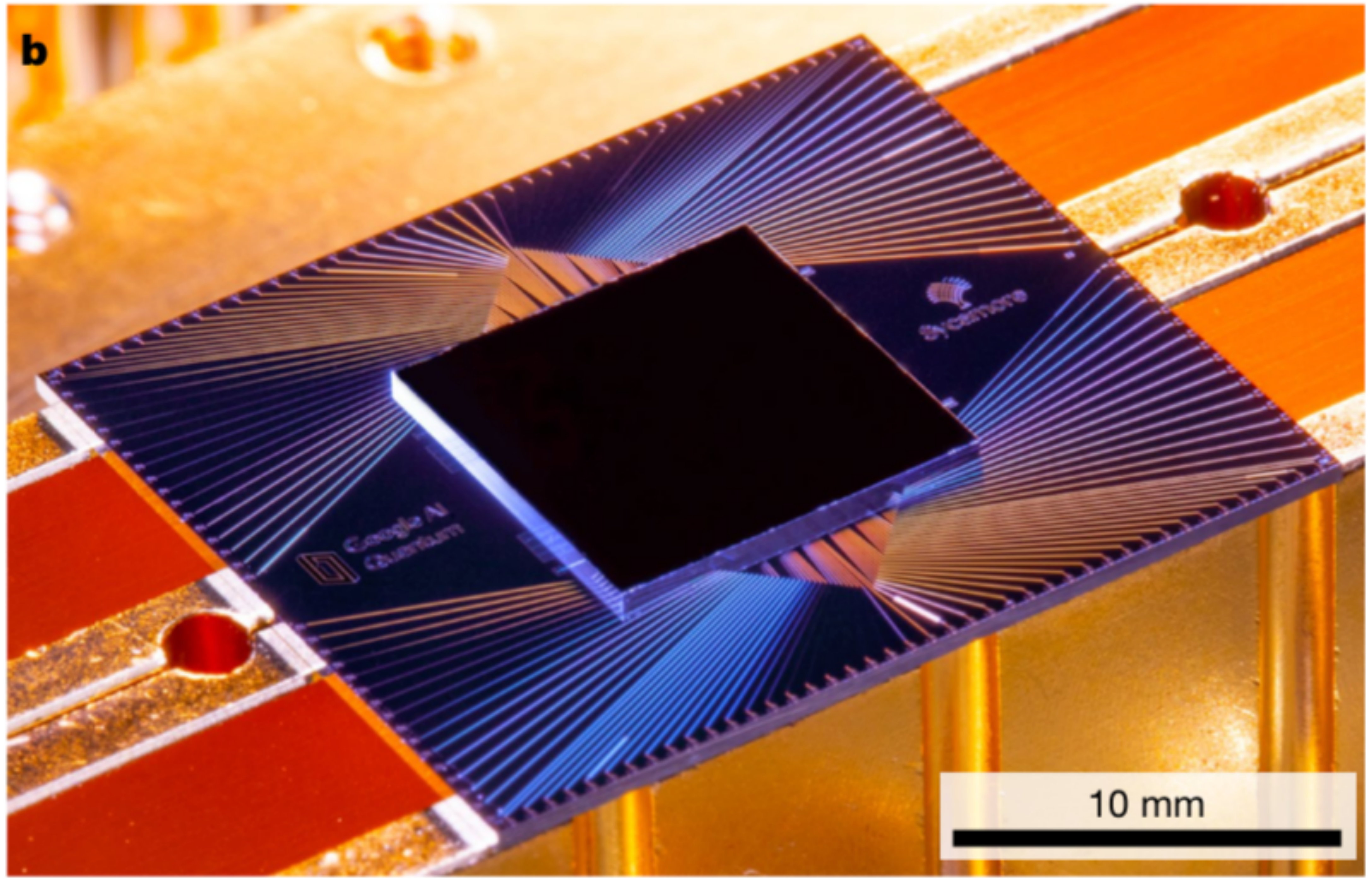
- The classical hardness is not solid.
  - The 10,000 years claim was for full simulation.



Does Google refute ECTT?

Google indeed had made significant progress, but…

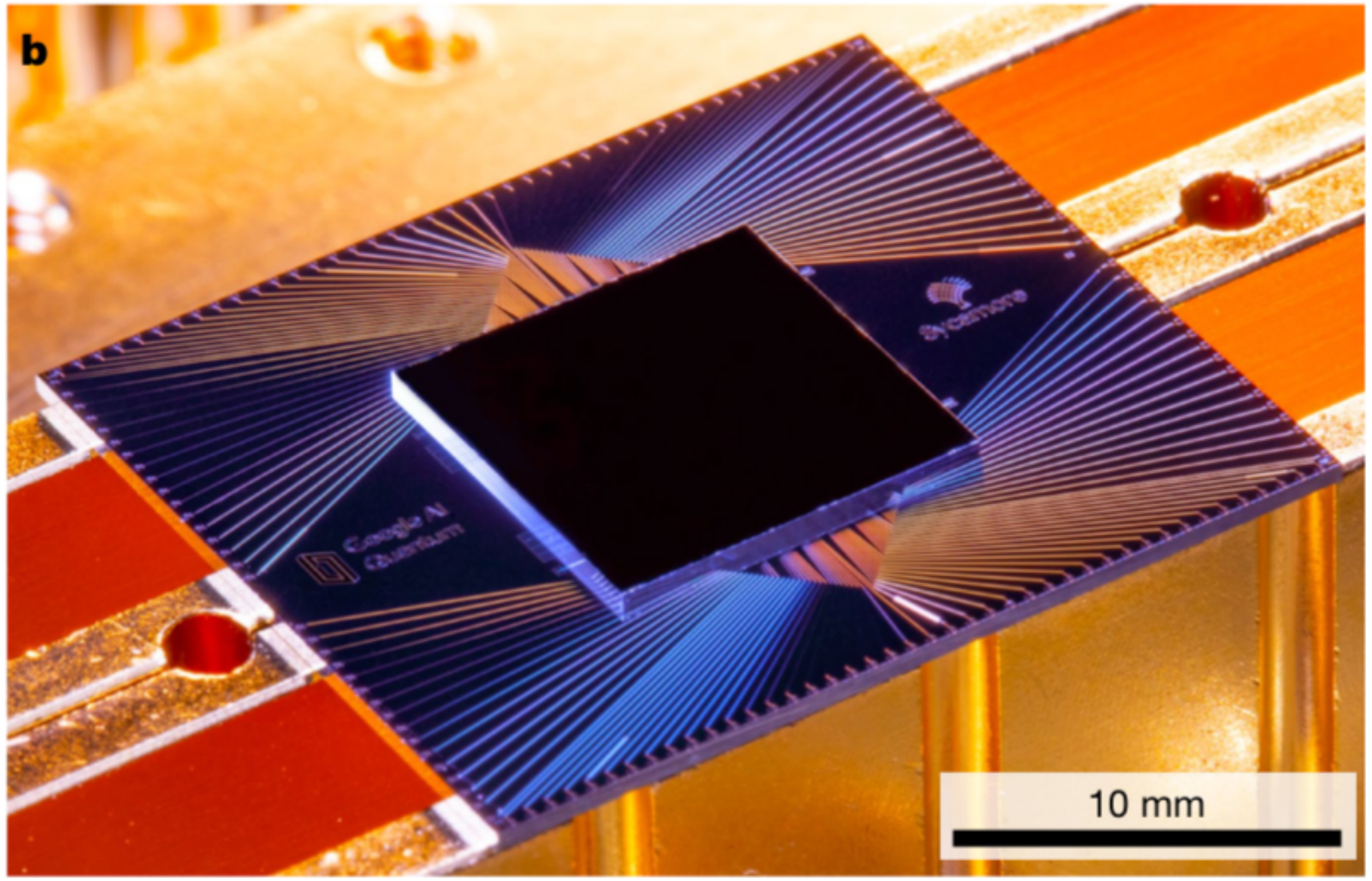
- The classical hardness is not solid.
  - The 10,000 years claim was for full simulation.
  - Other groups (e.g., IBM, Alibaba) claimed to have classical simulation running in few weeks/days.



Does Google refute ECTT?

Google indeed had made significant progress, but…

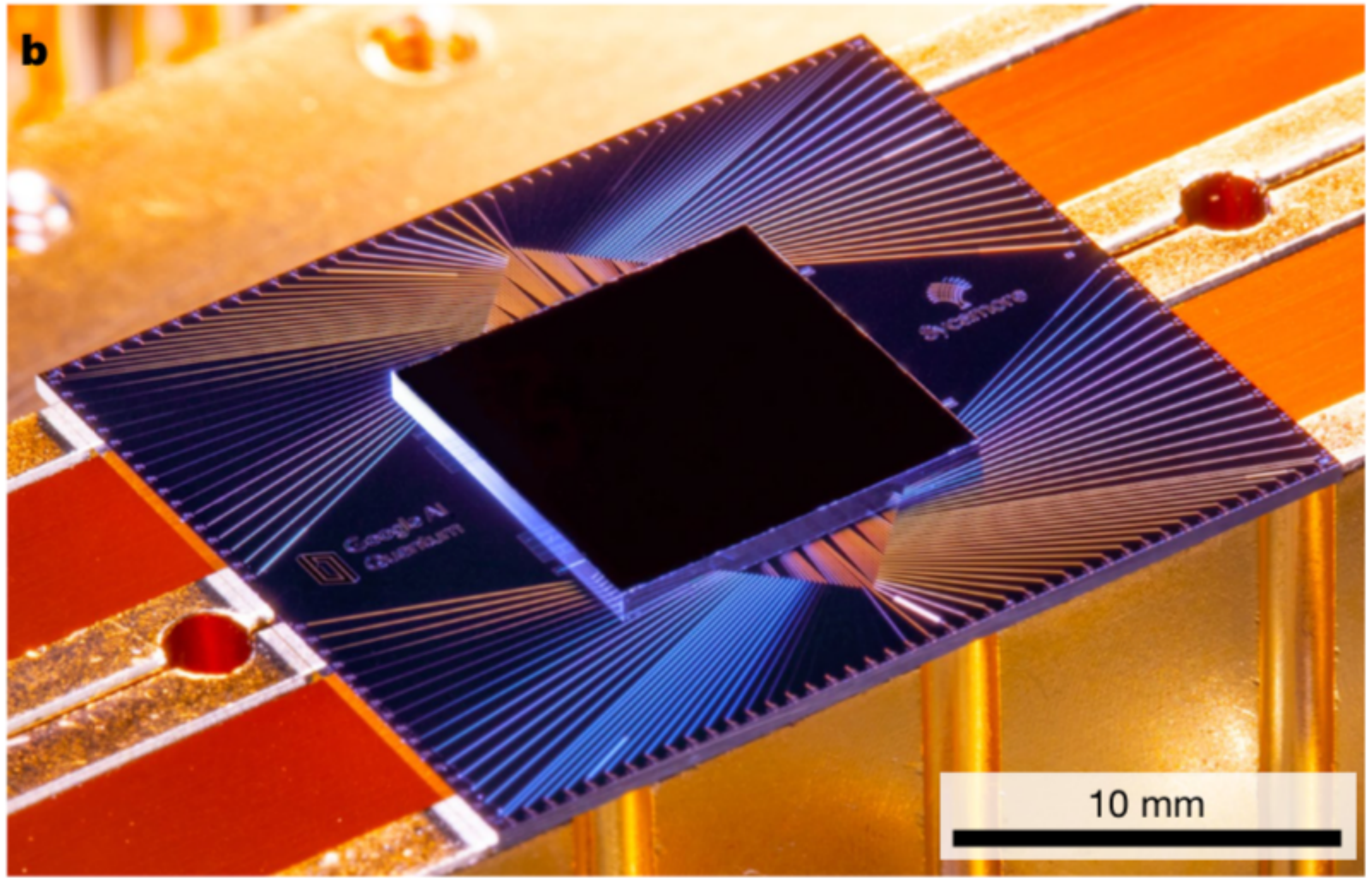
- The classical hardness is not solid.
  - The 10,000 years claim was for full simulation.
  - Other groups (e.g., IBM, Alibaba) claimed to have classical simulation running in few weeks/days.
  - Linear XEB is not very well-studied.



Does Google refute ECTT?

Google indeed had made significant progress, but…

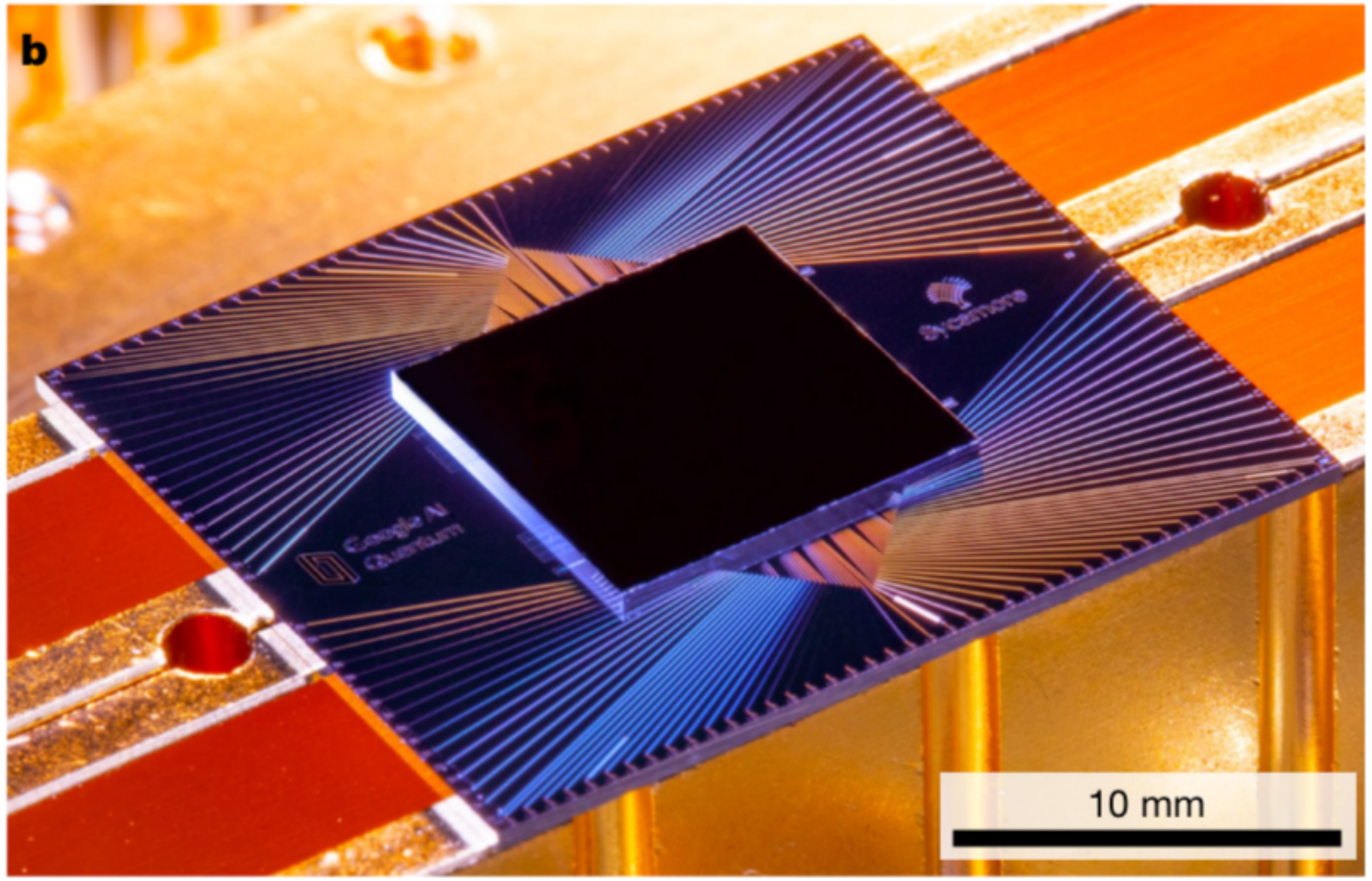
- The classical hardness is not solid.
  - The 10,000 years claim was for full simulation.
  - Other groups (e.g., IBM, Alibaba) claimed to have classical simulation running in few weeks/days.
  - Linear XEB is not very well-studied.
- Verification is too inefficient.



Does Google refute ECTT?

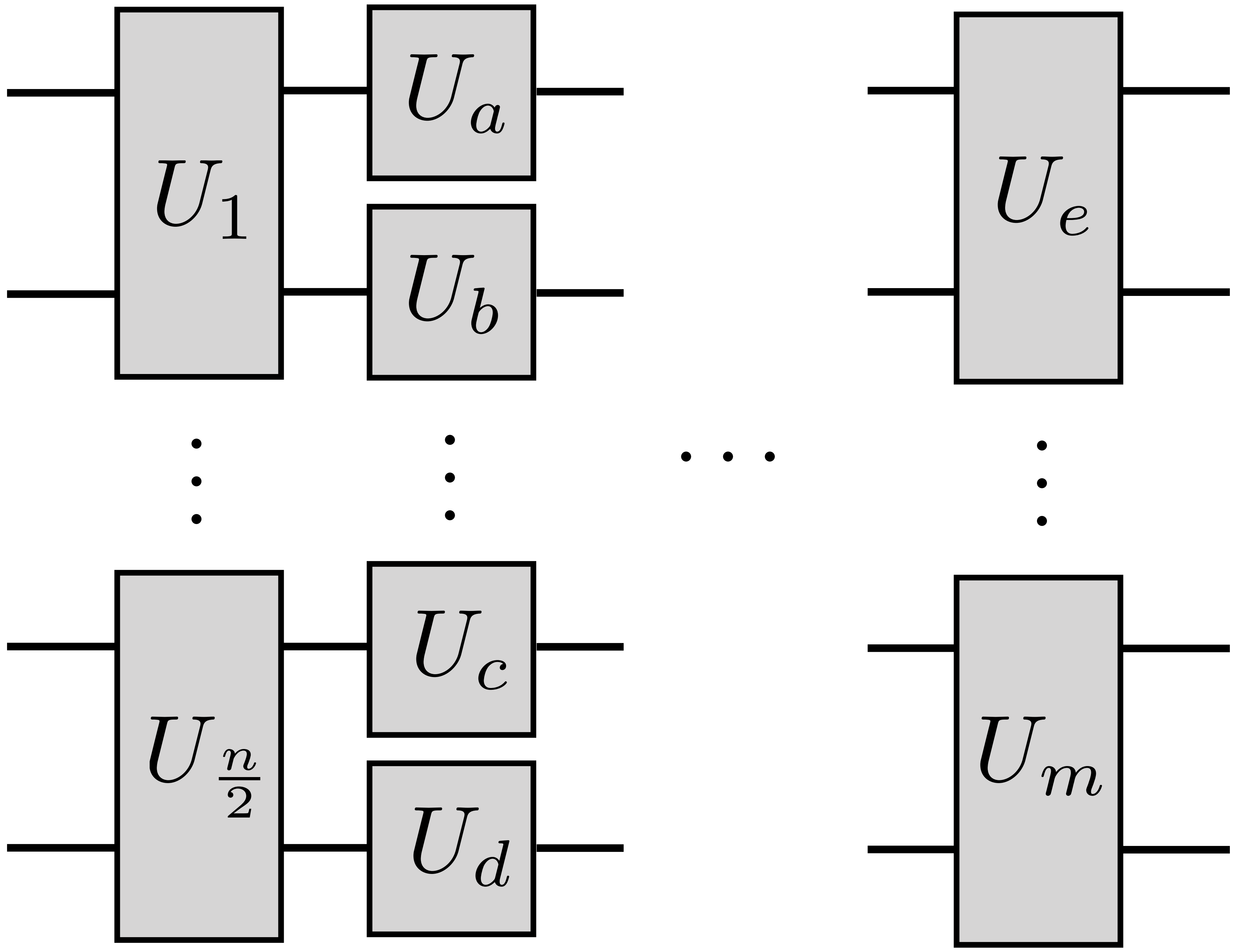
Google indeed had made significant progress, but…

- The classical hardness is not solid.
  - The 10,000 years claim was for full simulation.
  - Other groups (e.g., IBM, Alibaba) claimed to have classical simulation running in few weeks/days.
  - Linear XEB is not very well-studied.
- Verification is too inefficient.



Spoofing Linear XEB!?

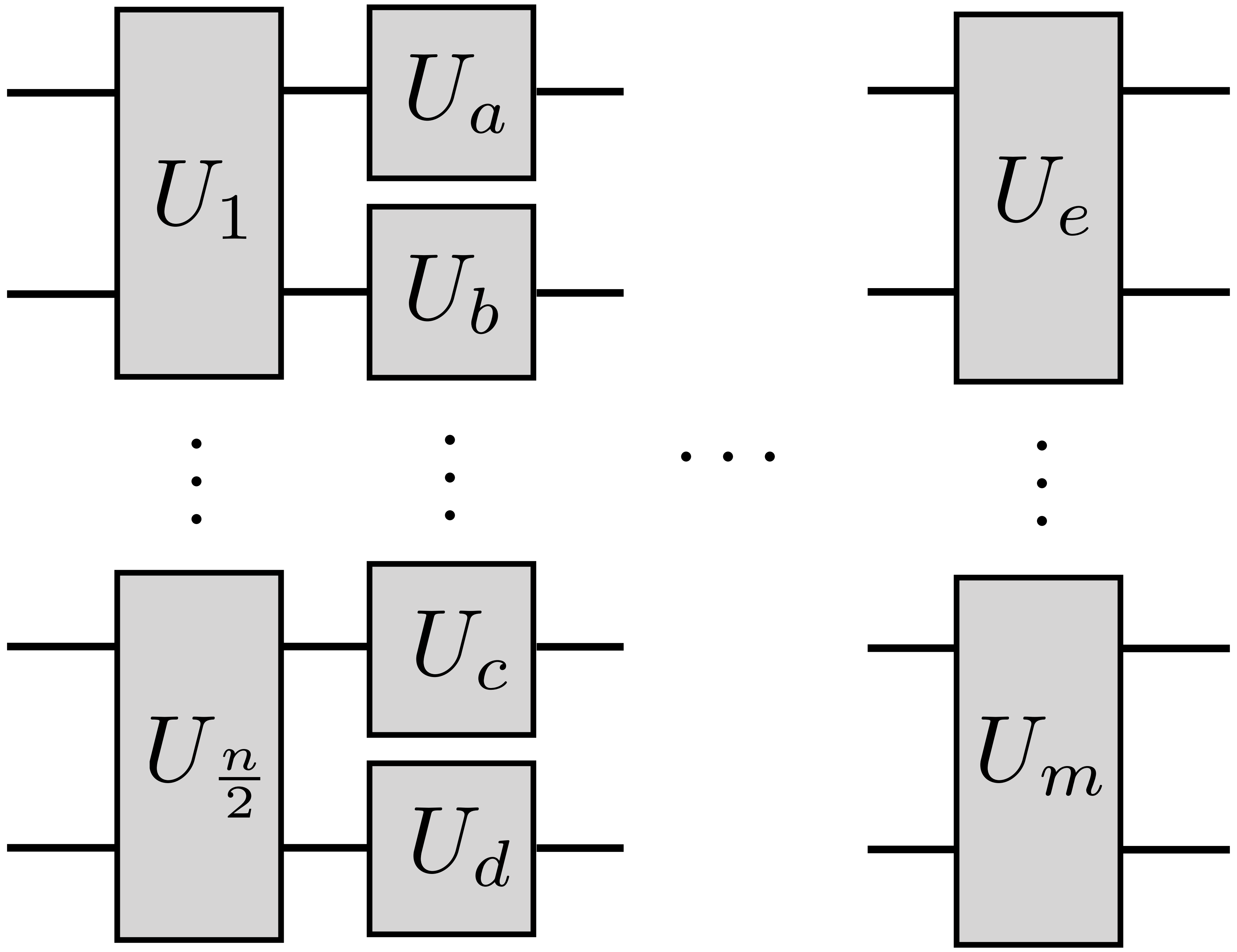
Are there classical algorithms specialized for spoofing Linear XEB?



$$\mathcal{F}_C(p) = \mathop{\mathbb{E}}_{x \sim p}\left[2^n q_C(x) - 1\right].$$

Spoofing Linear XEB!?

Are there classical algorithms specialized for spoofing Linear XEB?

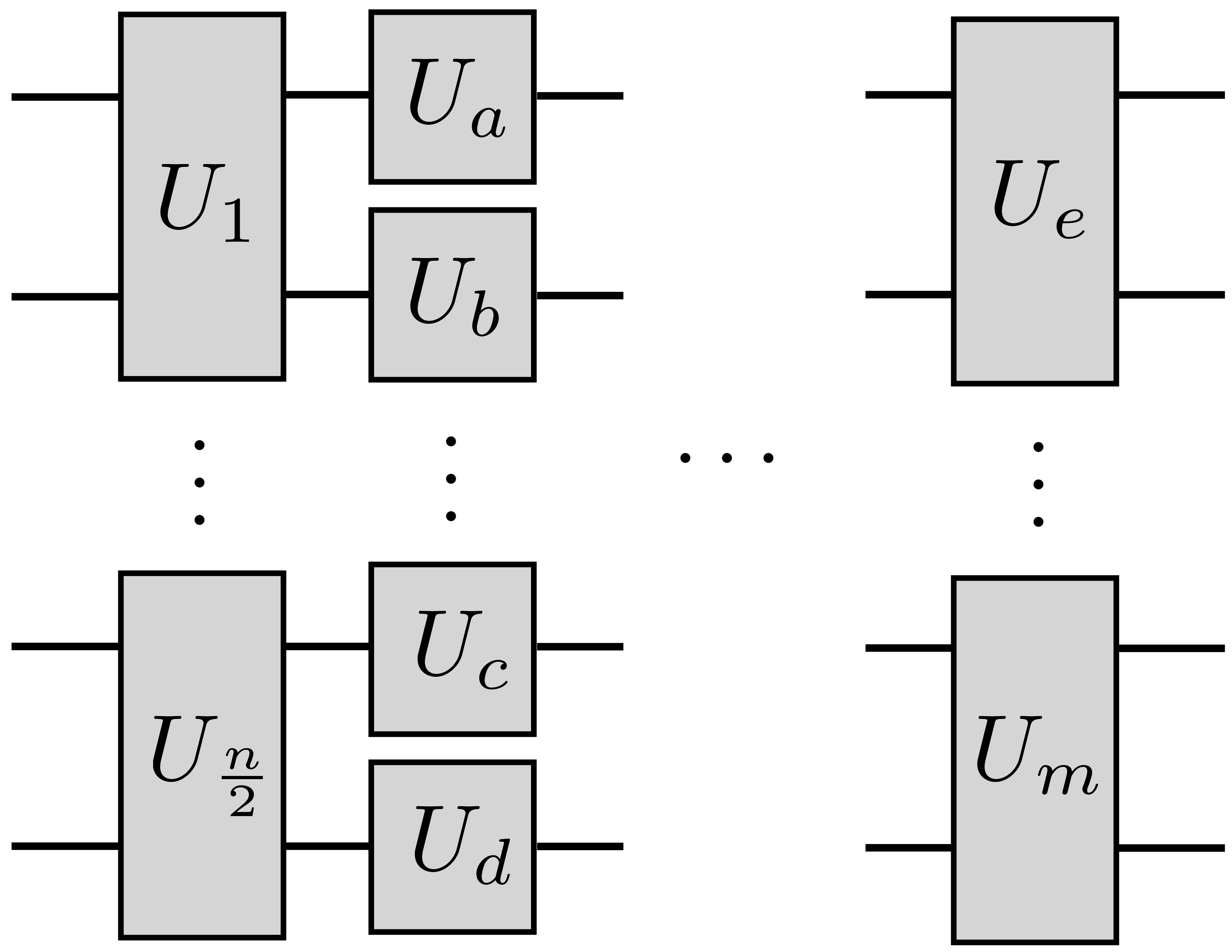


$$\mathcal{F}_C(p) = \mathop{\mathbb{E}}_{x\sim p}\left[2^n q_C(x) - 1\right].$$

- Is a full simulation for the circuit necessary for achieving high Linear XEB?

Spoofing Linear XEB!?

Are there classical algorithms specialized for spoofing Linear XEB?

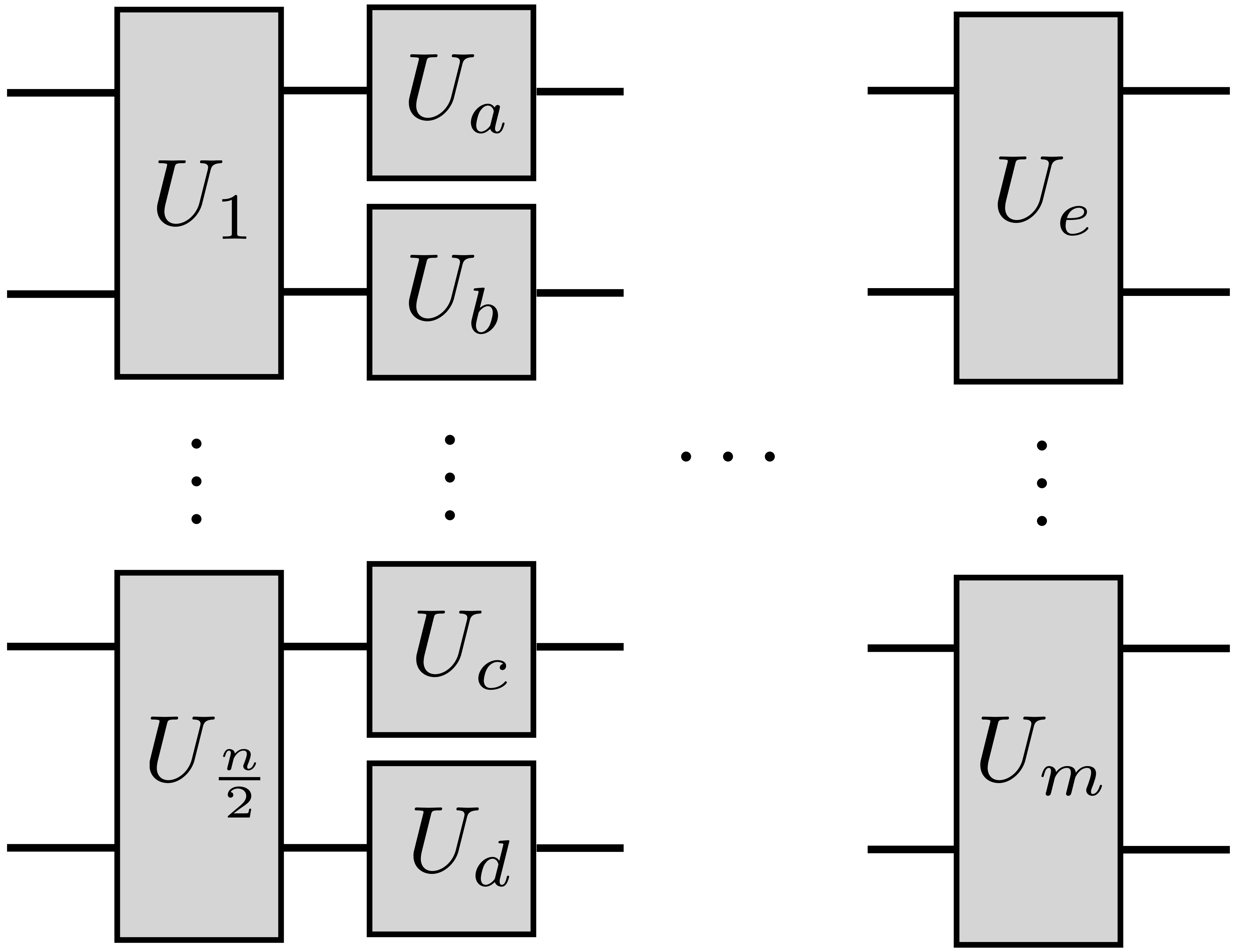


- Is a full simulation for the circuit necessary for achieving high Linear XEB?
- Can we leverage the structure of the circuits and the definition of Linear XEB!?

$$\mathcal{F}_C(p) = \mathop{\mathbb{E}}_{x \sim p} \left[2^n q_C(x) - 1 \right].$$

Spoofing Linear XEB!?

Are there classical algorithms specialized for spoofing Linear XEB?



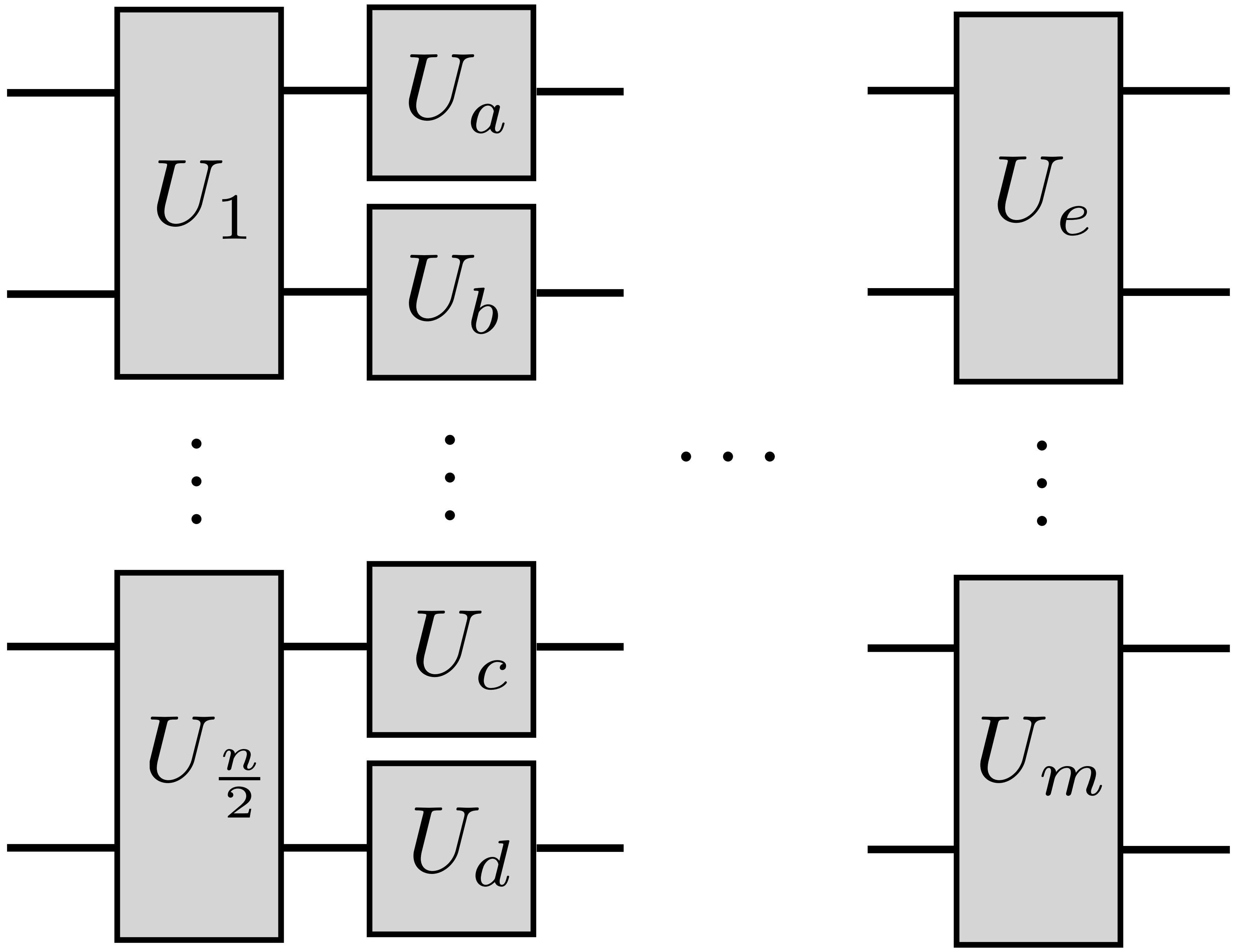
$$\mathcal{F}_C(p) = \mathop{\mathbb{E}}_{x \sim p}\left[2^n q_C(x) - 1\right].$$

- Is a full simulation for the circuit necessary for achieving high Linear XEB?
- Can we leverage the structure of the circuits and the definition of Linear XEB!?

Two difficulties to spoof Linear XEB:

Spoofing Linear XEB!?

Are there classical algorithms specialized for spoofing Linear XEB?



$$\mathcal{F}_C(p) = \mathop{\mathbb{E}}_{x\sim p}\left[2^n q_C(x) - 1\right].$$

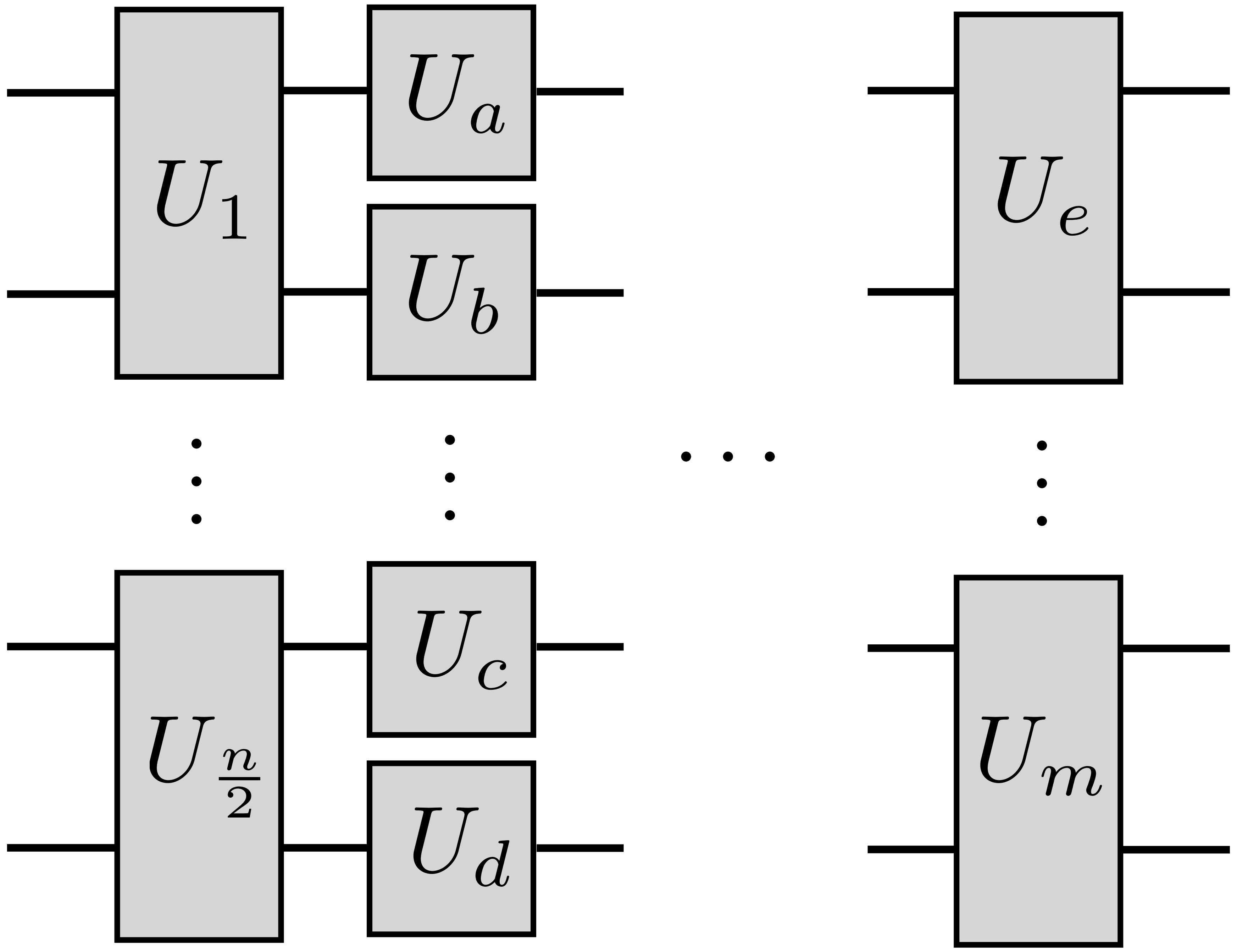
- Is a full simulation for the circuit necessary for achieving high Linear XEB?
- Can we leverage the structure of the circuits and the definition of Linear XEB!?

Two difficulties to spoof Linear XEB:

- Theoretically, challenging to analyze.

Spoofing Linear XEB!?

Are there classical algorithms specialized for spoofing Linear XEB?



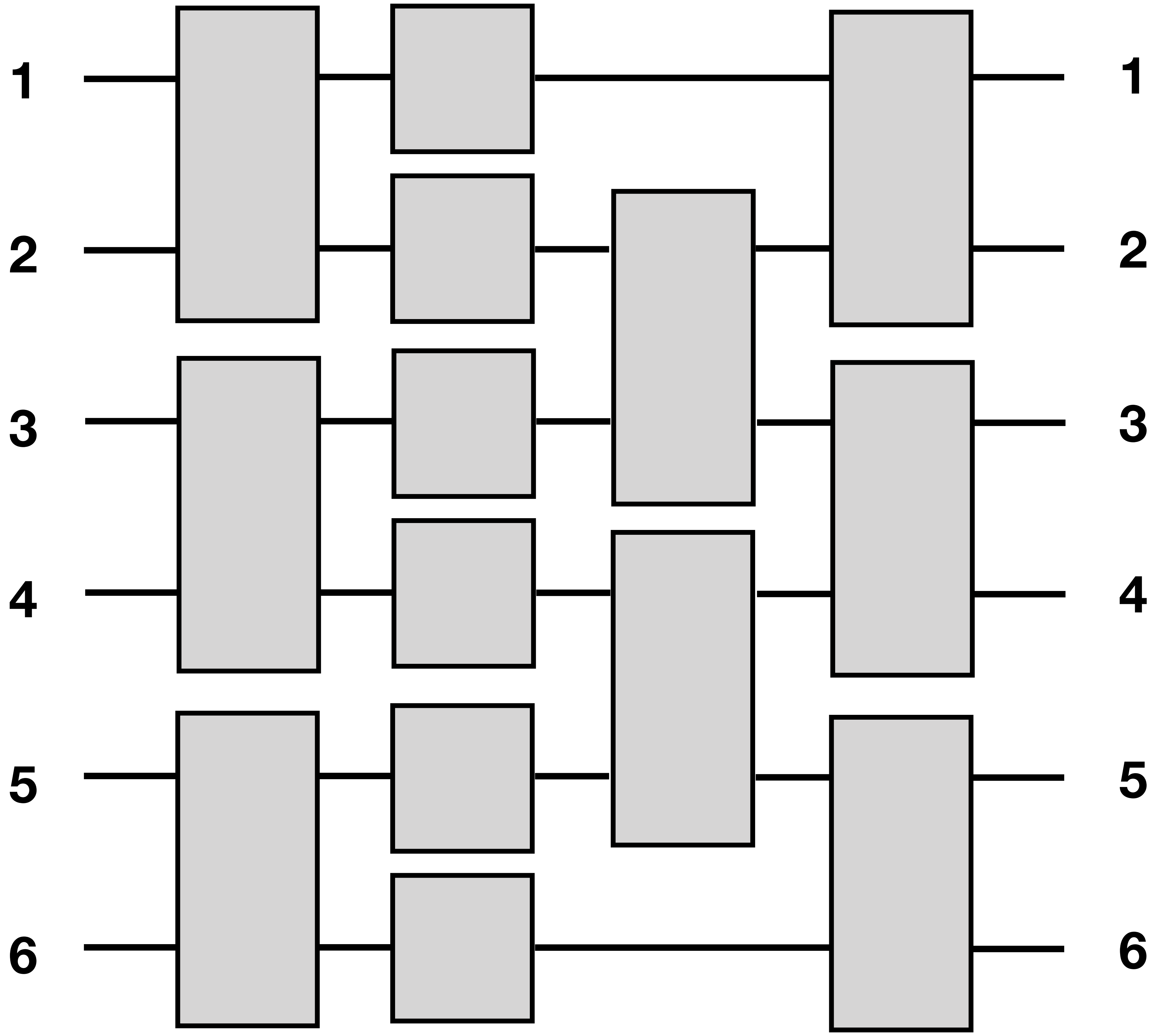
$$\mathcal{F}_C(p) = \mathop{\mathbb{E}}_{x \sim p} \left[2^n q_C(x) - 1 \right].$$

- Is a full simulation for the circuit necessary for achieving high Linear XEB?
- Can we leverage the structure of the circuits and the definition of Linear XEB!?

Two difficulties to spoof Linear XEB:

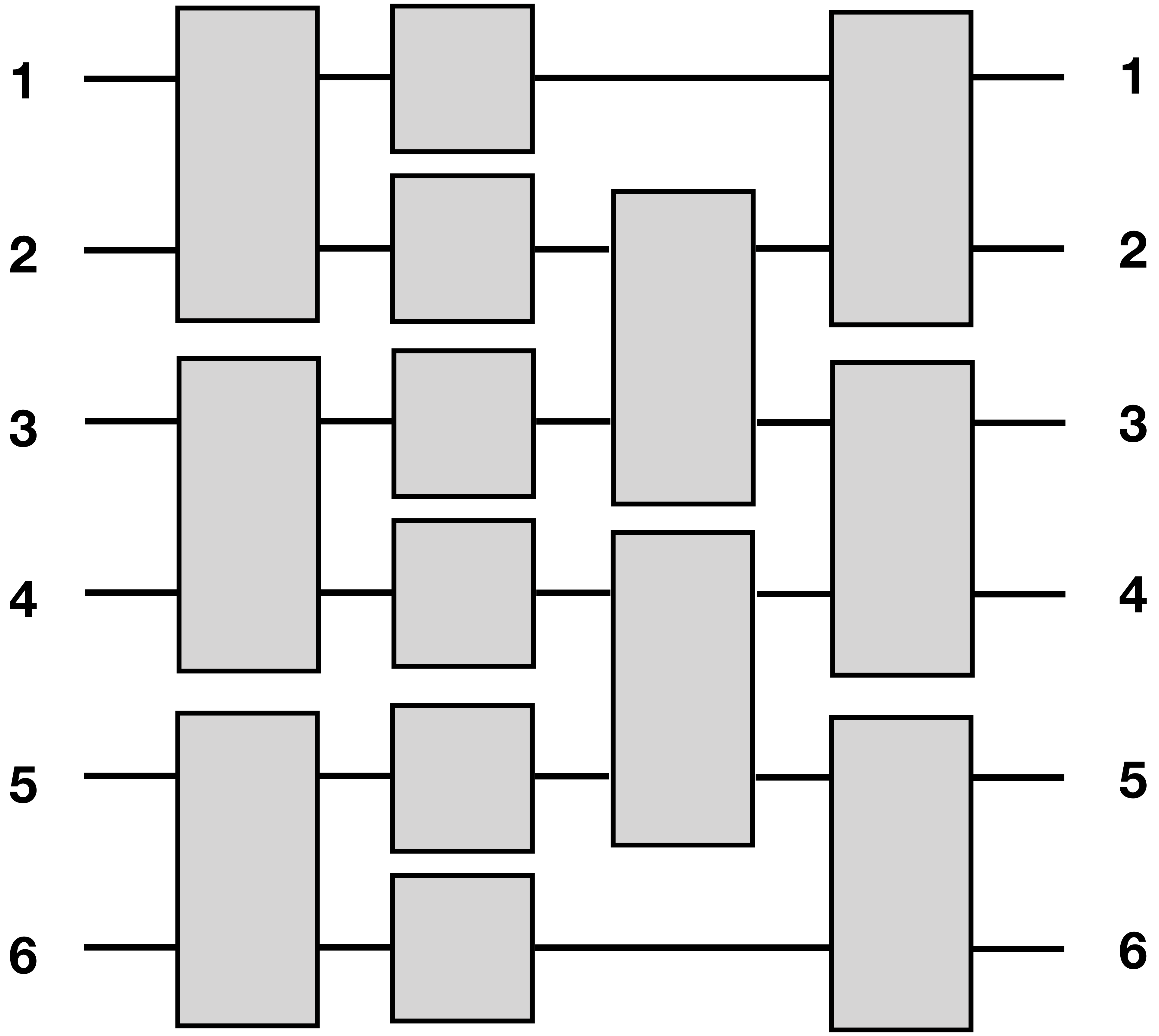
- Theoretically, challenging to analyze.
- Experimentally, the verification is inefficient.

Spoofing Linear XEB in Shallow Circuits



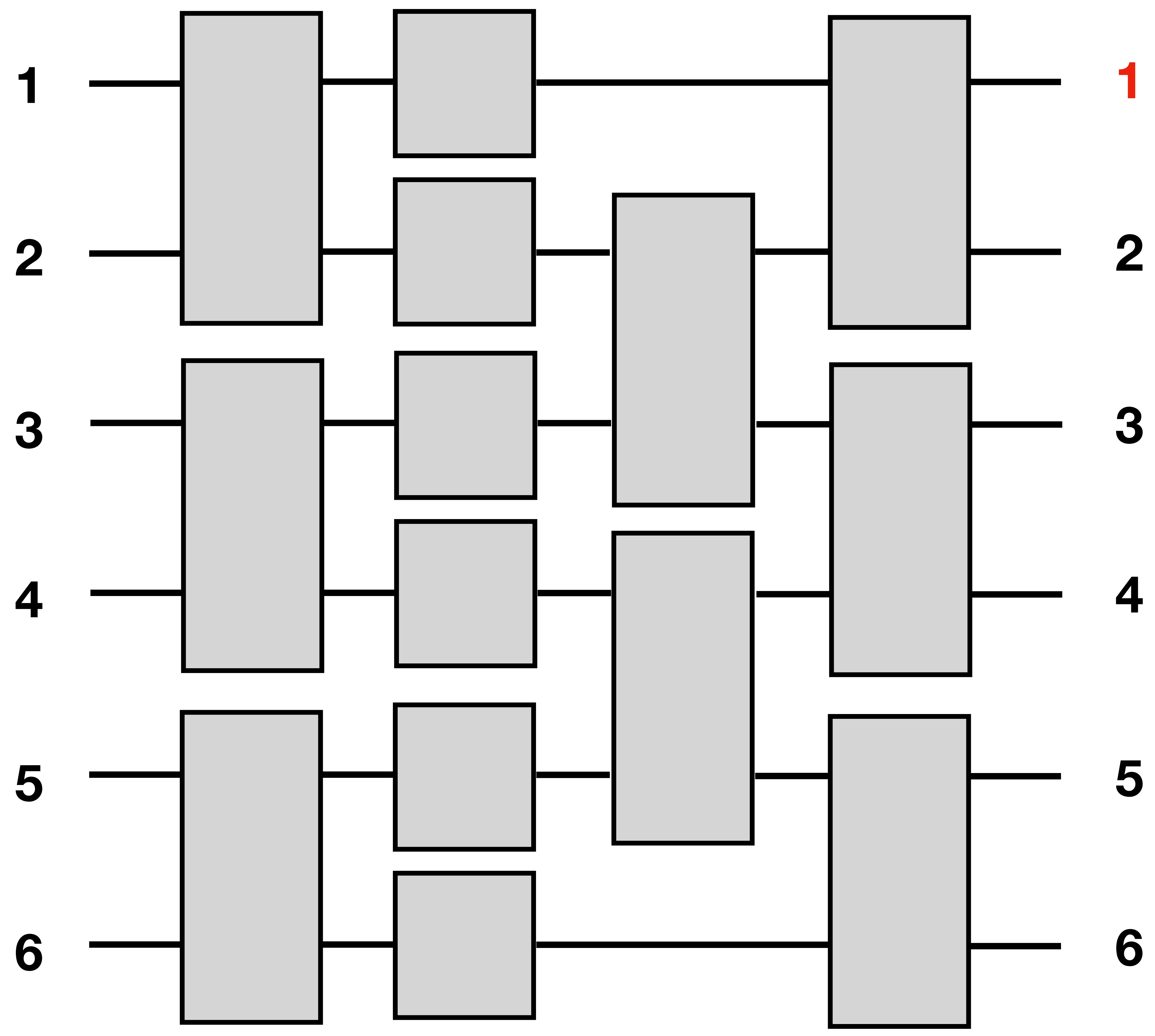
Spoofing Linear XEB in Shallow Circuits

Observation: The marginal of an output qubit only depends on its *lightcone*!



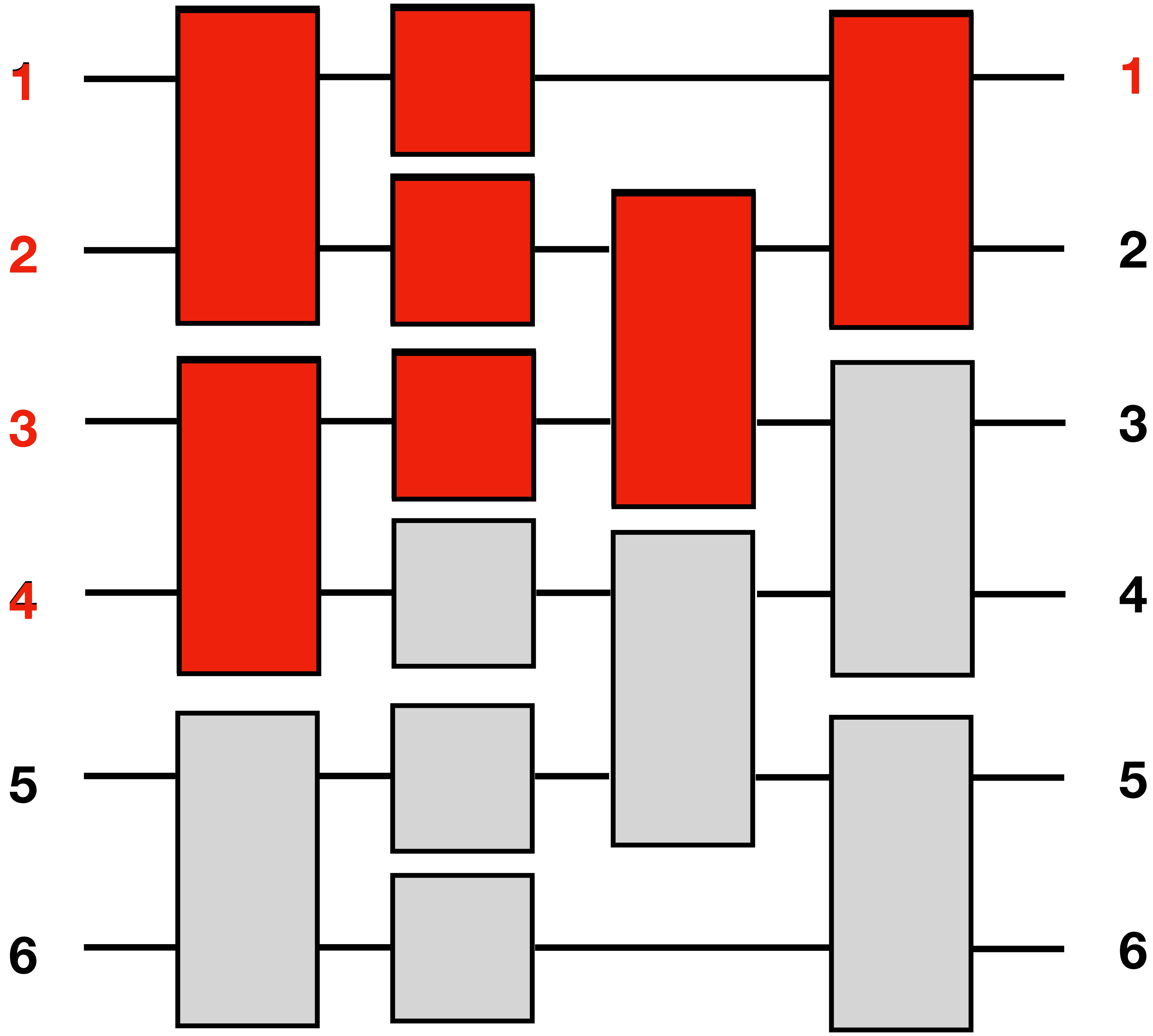
Spoofing Linear XEB in Shallow Circuits

Observation: The marginal of an output qubit only depends on its *lightcone*!



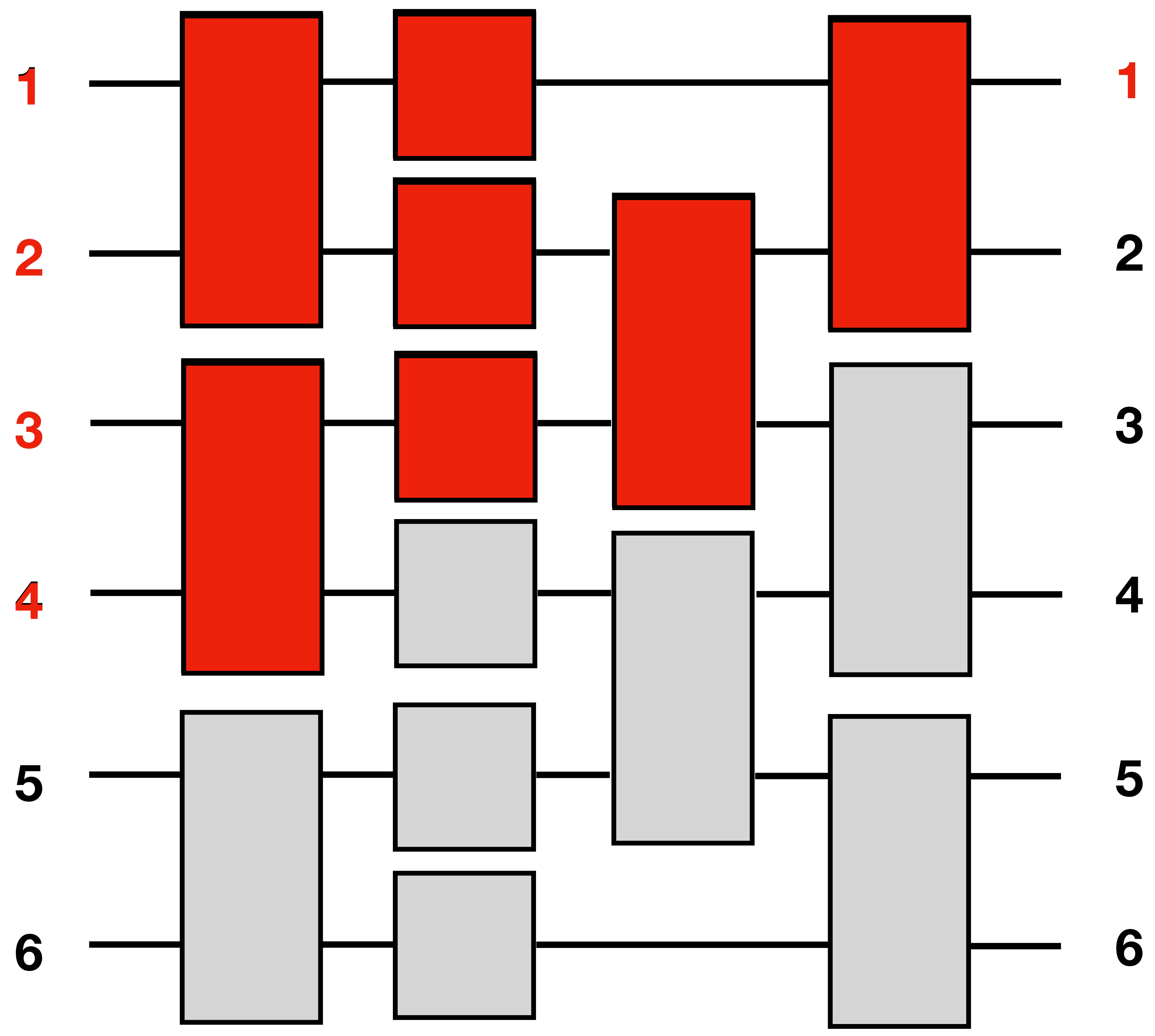
Spoofing Linear XEB in Shallow Circuits

Observation: The marginal of an output qubit only depends on its *lightcone*!



Spoofing Linear XEB in Shallow Circuits

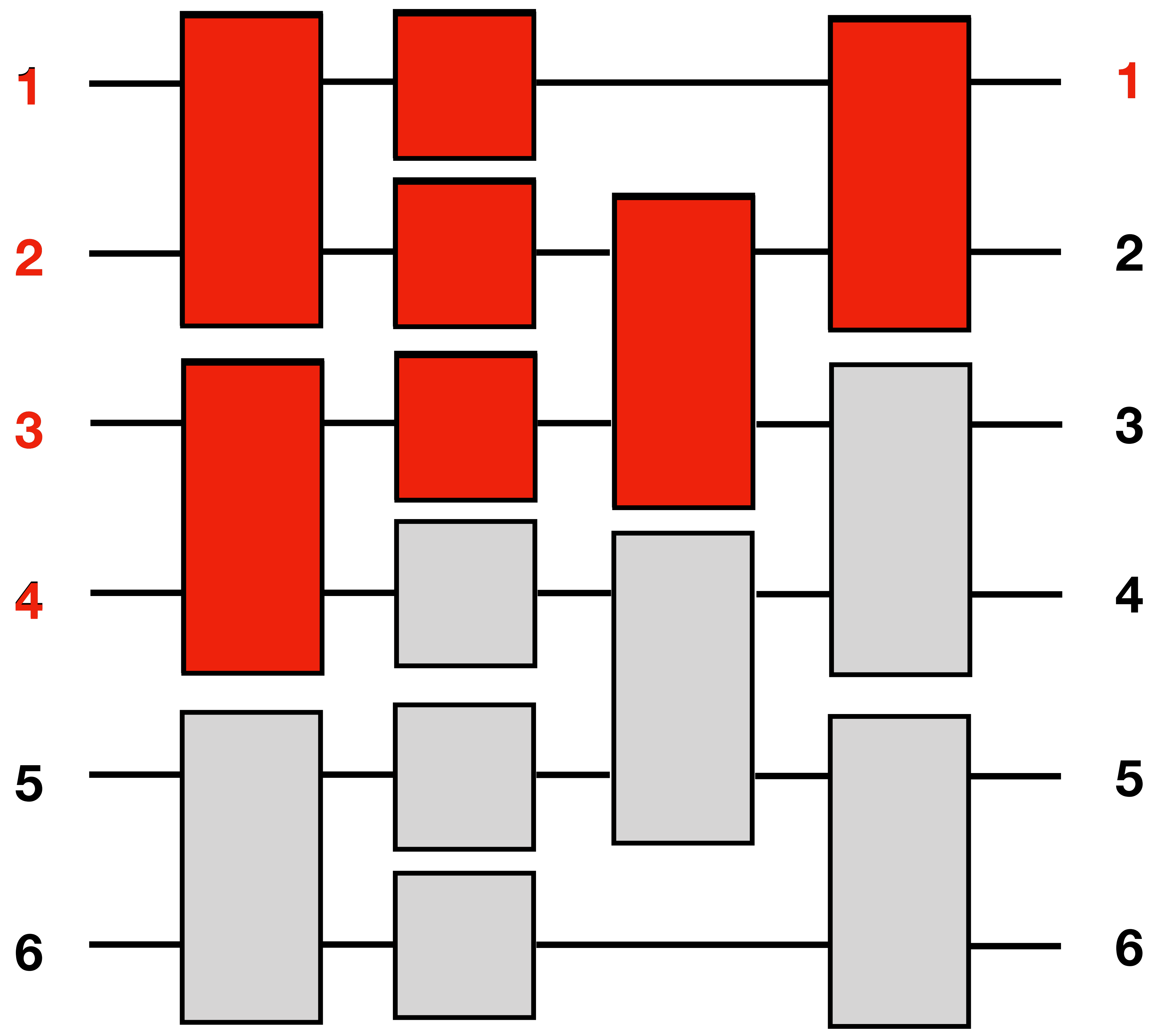
Observation: The marginal of an output qubit only depends on its *lightcone*!



- Calculating the marginal of output **1** only requires $O(2^4)$ time instead of $O(2^6)$ time.

Spoofing Linear XEB in Shallow Circuits

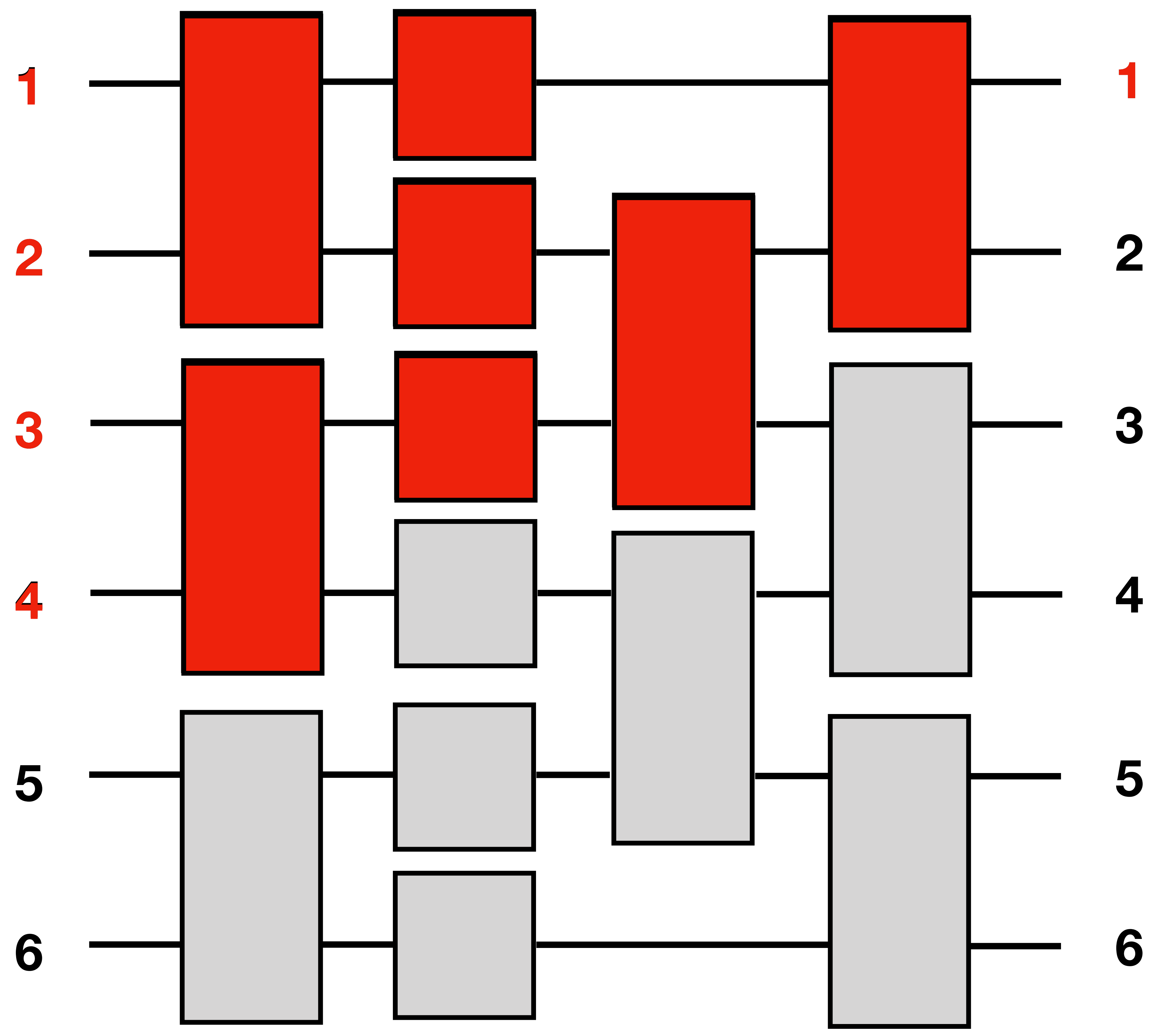
Observation: The marginal of an output qubit only depends on its *lightcone*!



- Calculating the marginal of output 1 only requires $O(2^4)$ time instead of $O(2^6)$ time.
- For a depth-d circuit, we can calculate the marginal of each output qubit in time $O(2^{2^d})$.

Spoofing Linear XEB in Shallow Circuits

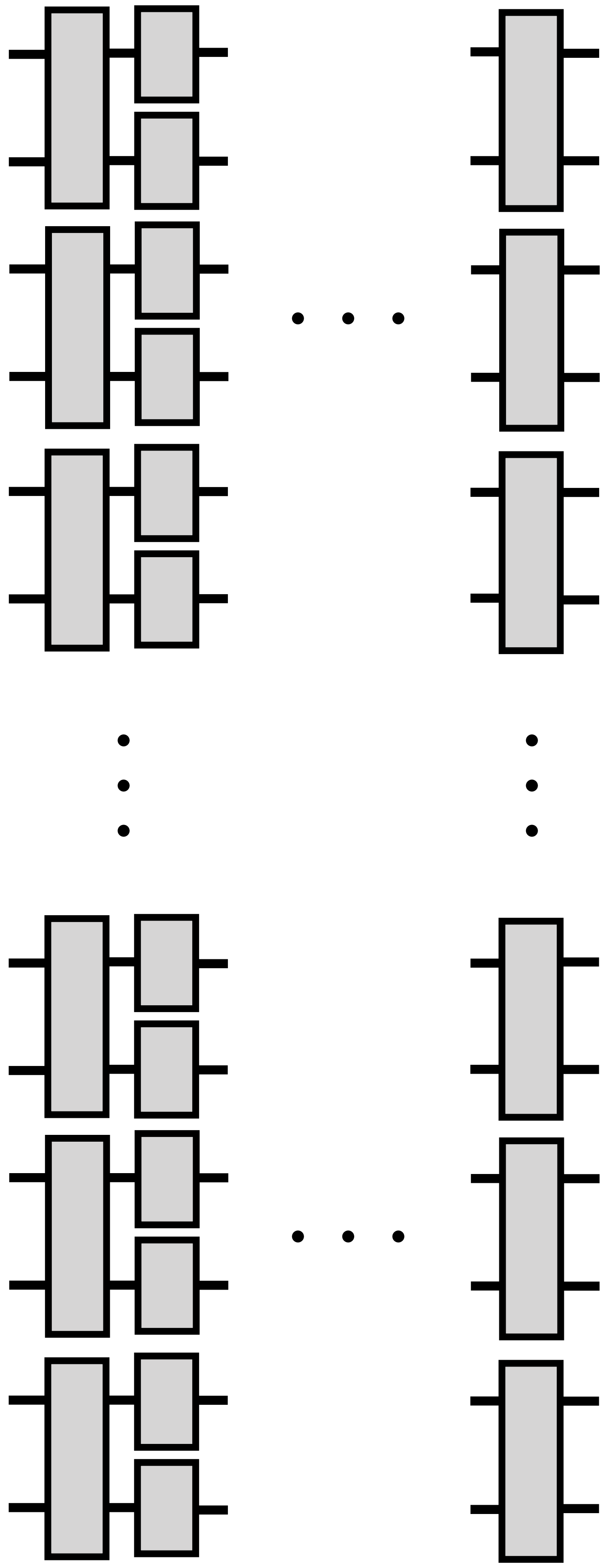
Observation: The marginal of an output qubit only depends on its *lightcone*!



- Calculating the marginal of output **1** only requires $O(2^4)$ time instead of $O(2^6)$ time.
- For a depth-d circuit, we can calculate the marginal of each output qubit in time $O(2^{2^d})$.

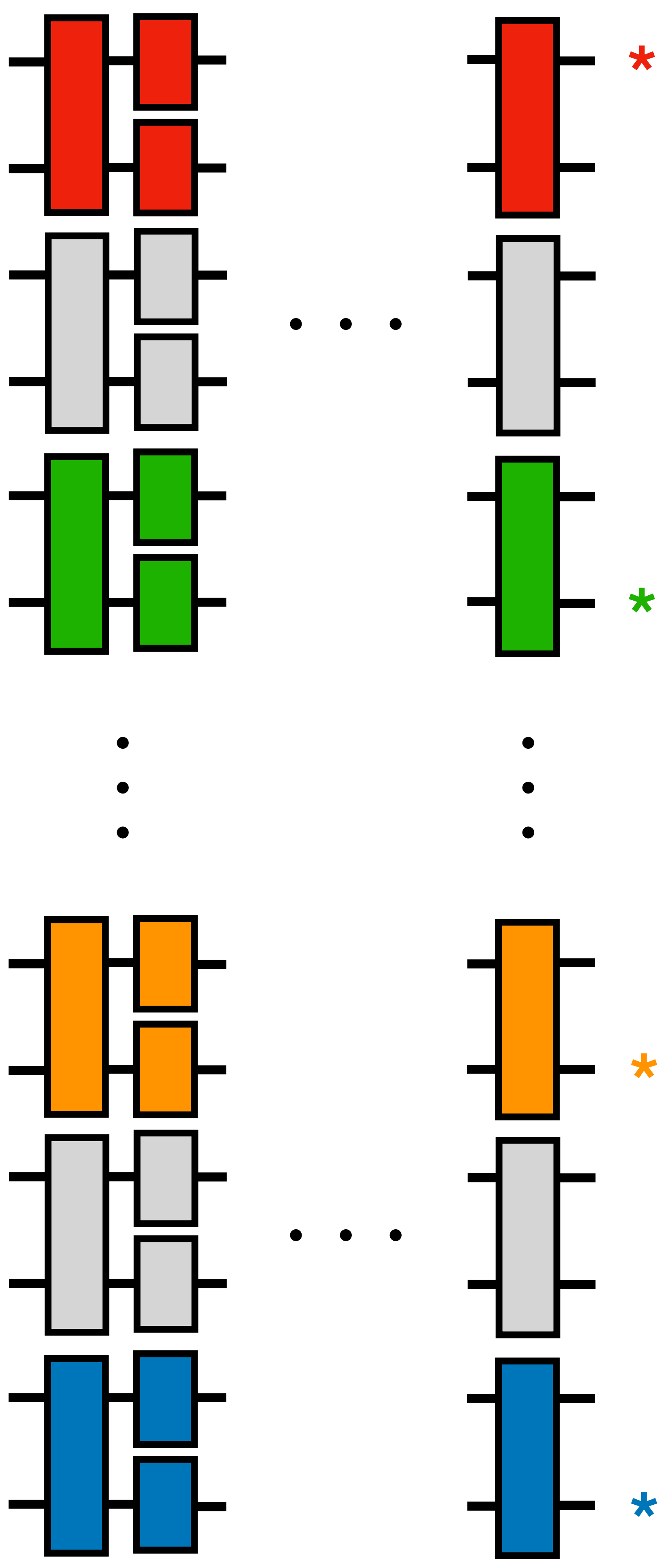
Idea: Use the marginal of each output qubit to perform biased sampling!

A Classical Algorithm Spoofing Linear XEB Using Lightcone



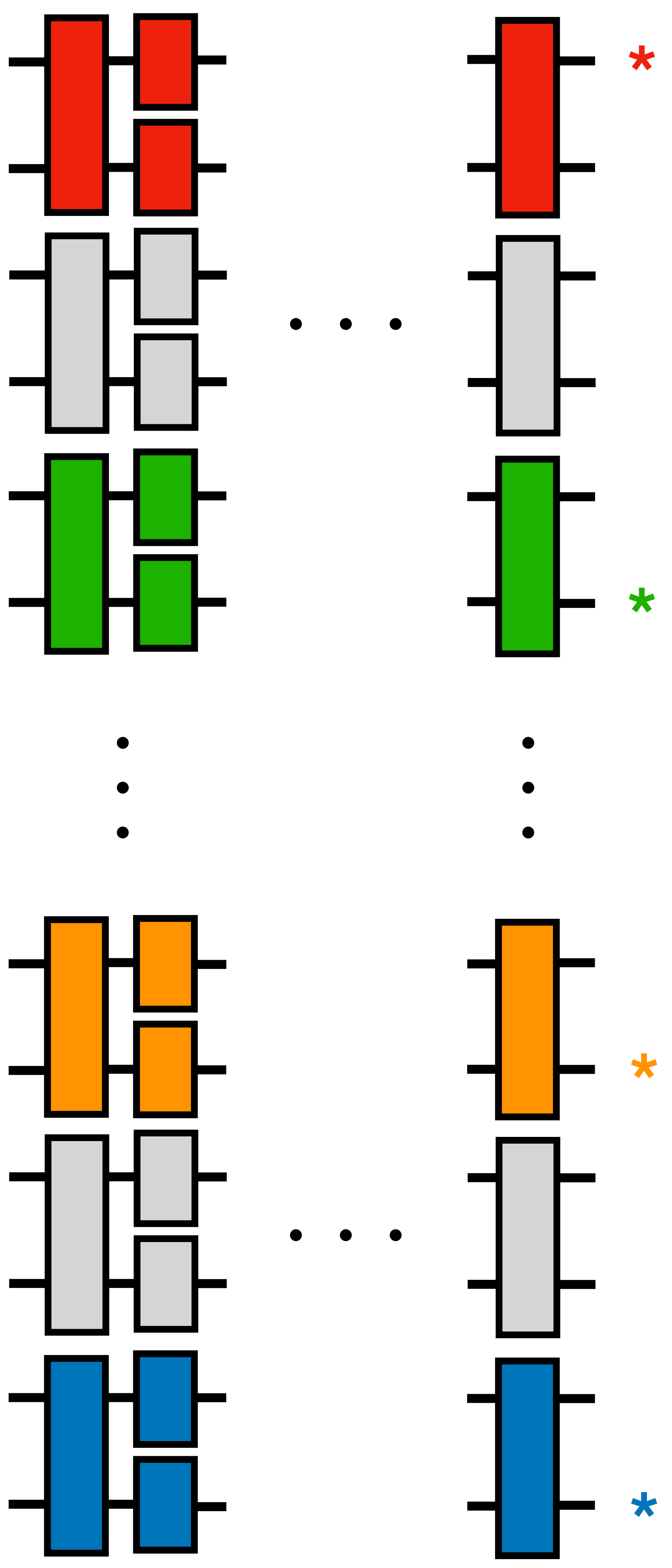
A Classical Algorithm Spoofing Linear XEB Using Lightcone

1. Find a set of output qubits whose lightcones are *disjoint* and of size L.



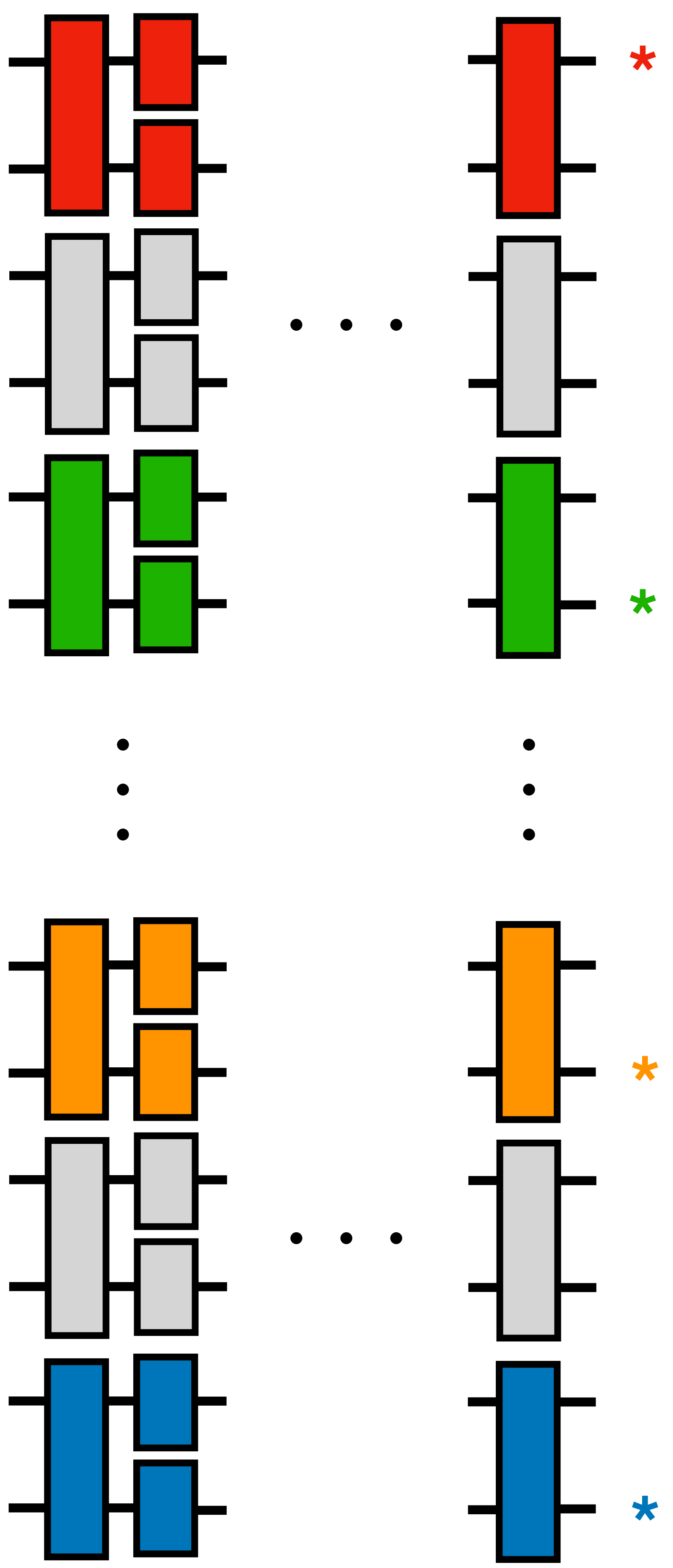
A Classical Algorithm Spoofing Linear XEB Using Lightcone

1. Find a set of output qubits whose lightcones are *disjoint* and of size L.
2. Calculate the marginal of each output qubits in time $O(2^L)$.



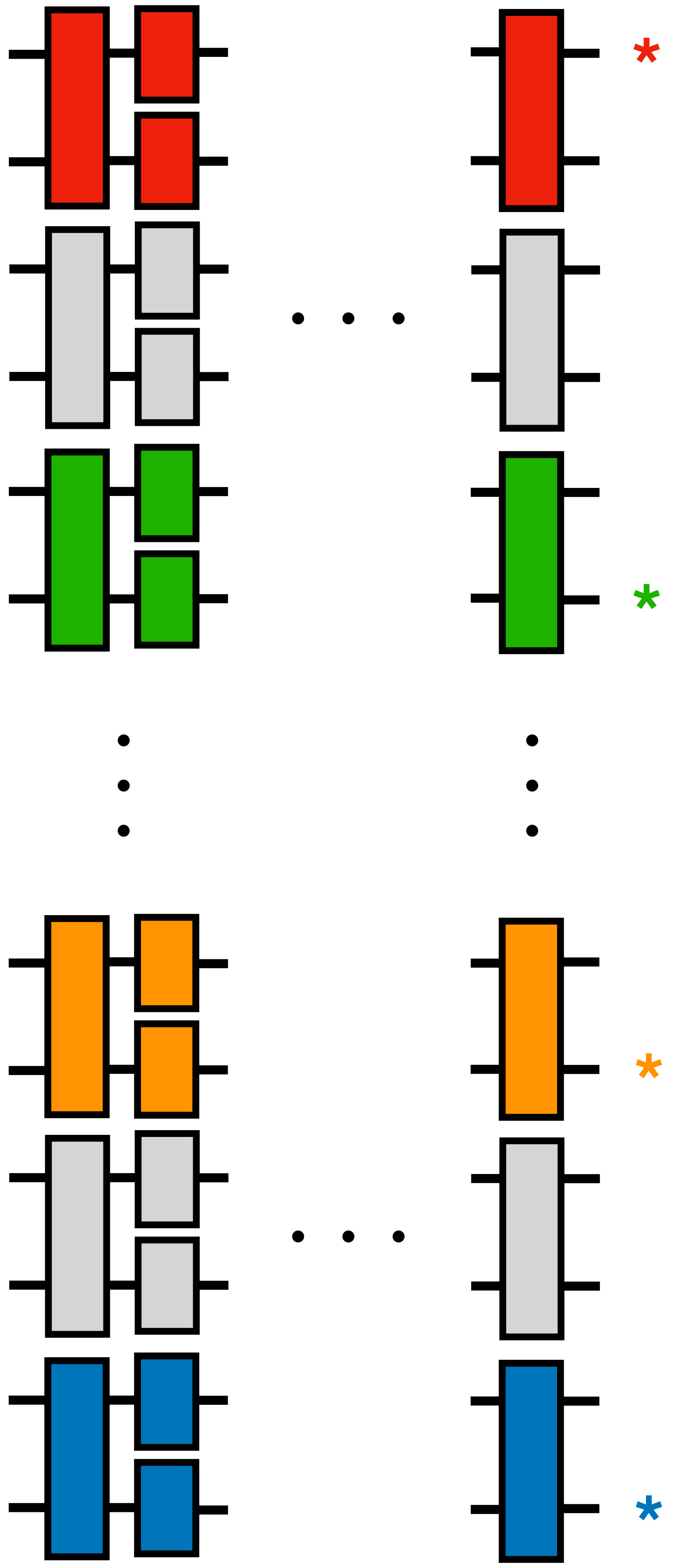
A Classical Algorithm Spoofing Linear XEB Using Lightcone

1. Find a set of output qubits whose lightcones are *disjoint* and of size L.
2. Calculate the marginal of each output qubits in time $O(2^L)$.
3. Sample the output qubits independently:



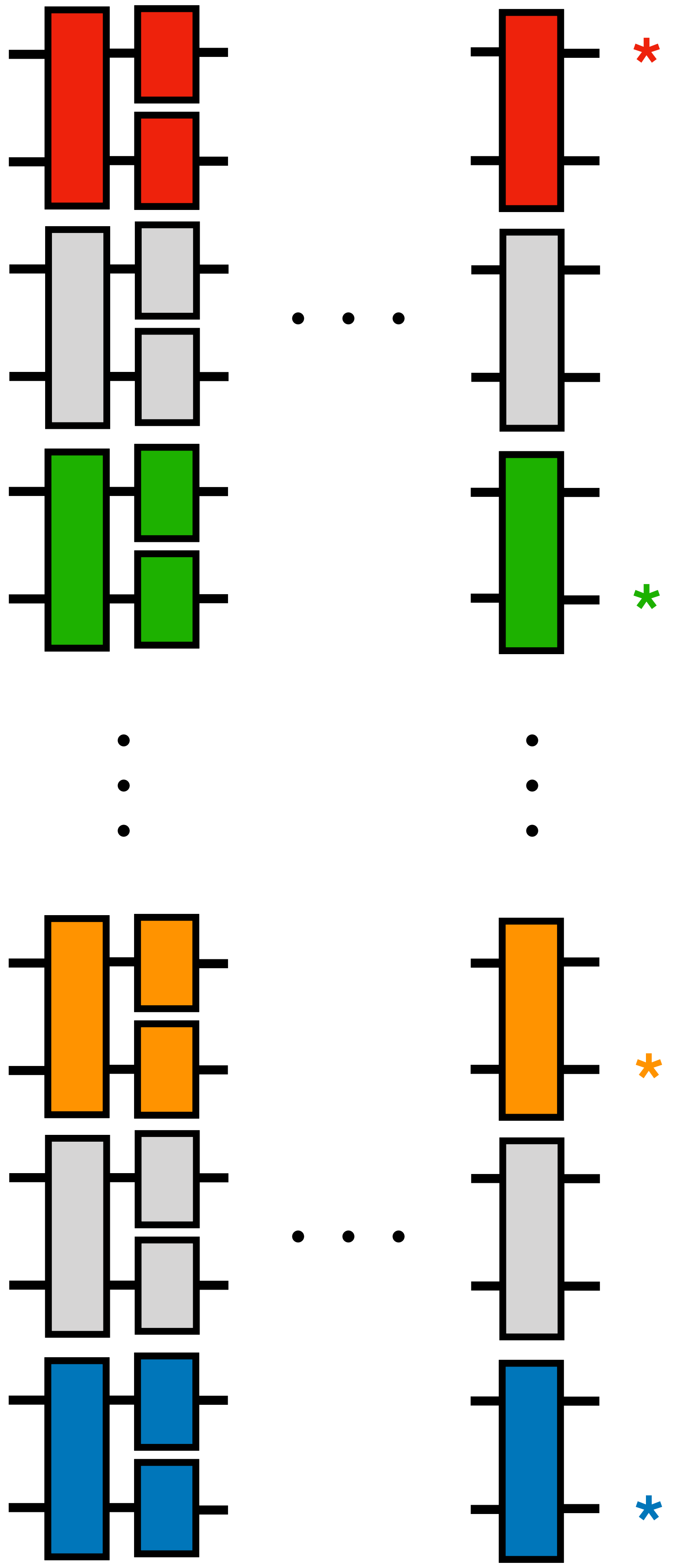
A Classical Algorithm Spoofing Linear XEB Using Lightcone

1. Find a set of output qubits whose lightcones are *disjoint* and of size L.
2. Calculate the marginal of each output qubits in time $O(2^L)$.
3. Sample the output qubits independently:
   - If the output qubits were chosen, then sample according to the marginal.



A Classical Algorithm Spoofing Linear XEB Using Lightcone

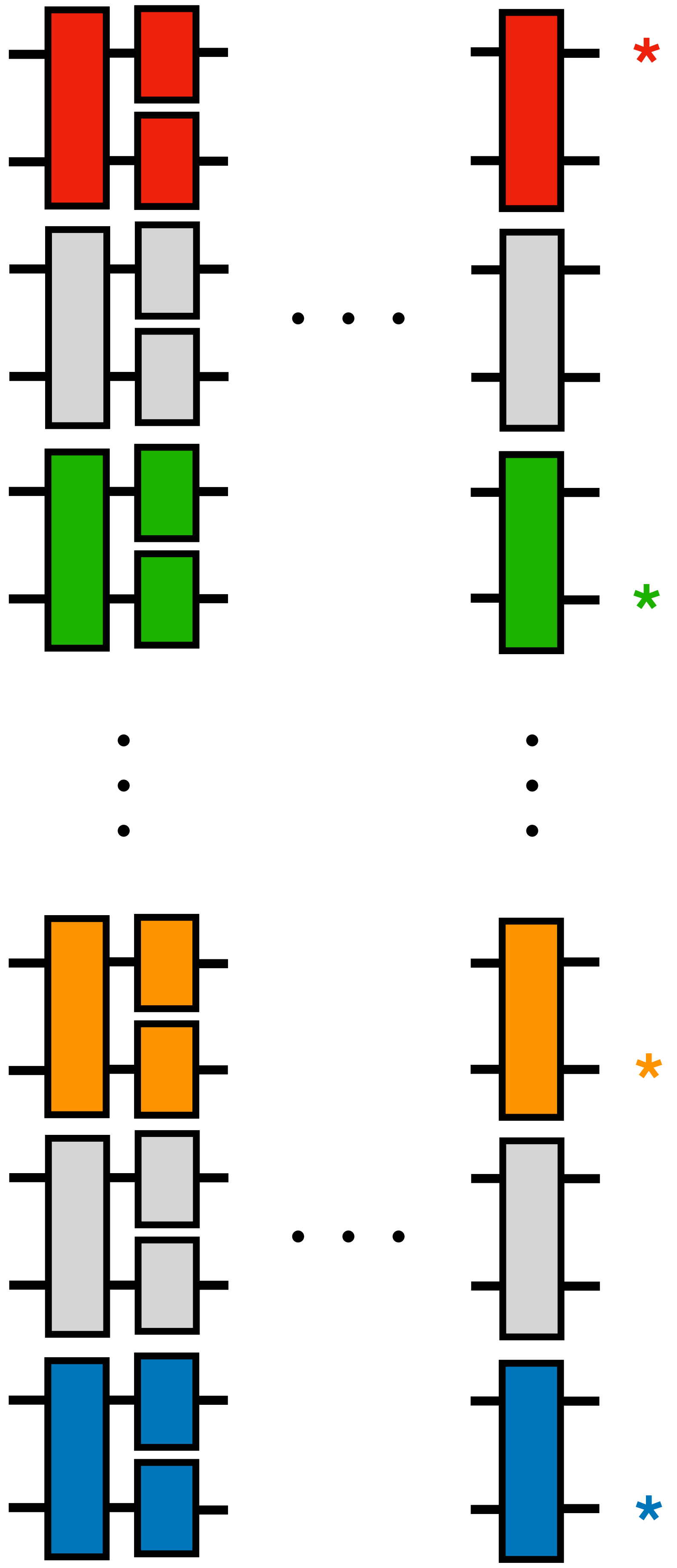
1. Find a set of output qubits whose lightcones are *disjoint* and of size L.
2. Calculate the marginal of each output qubits in time $O(2^L)$.
3. Sample the output qubits independently:
   - If the output qubits were chosen, then sample according to the marginal.
   - If not, then sample uniformly.



A Classical Algorithm Spoofing Linear XEB Using Lightcone

1. Find a set of output qubits whose lightcones are *disjoint* and of size L.
2. Calculate the marginal of each output qubits in time $O(2^L)$.
3. Sample the output qubits independently:
   - If the output qubits were chosen, then sample according to the marginal.
   - If not, then sample uniformly.

Theoretical analysis? Empirical performance? Other variants?



Theoretical Analysis

First non-trivial classical algorithm challenging Linear XEB

Theoretical Analysis

First non-trivial classical algorithm challenging Linear XEB

Theorem.

Theoretical Analysis

First non-trivial classical algorithm challenging Linear XEB

Theorem.

Let C be an n-qubit depth-d circuit with (i) lightcone size at most L and (ii) each 2-qubit gate is Haar random, then the algorithm outputs a distribution p in time and

Theoretical Analysis

First non-trivial classical algorithm challenging Linear XEB

Theorem.

Let C be an n-qubit depth-d circuit with (i) lightcone size at most L and (ii) each 2-qubit gate is Haar random, then the algorithm outputs a distribution p in time $\text{poly}(n, 2^L)$ and

Theoretical Analysis

First non-trivial classical algorithm challenging Linear XEB

Theorem.

Let C be an n-qubit depth-d circuit with (i) lightcone size at most L and (ii) each 2-qubit gate is Haar random, then the algorithm outputs a distribution p in time $\text{poly}(n, 2^L)$ and

$$\mathop{\mathbb{E}}_{C}[\mathcal{F}_C(p)] \geq (1 + 15^{-d})^{n/L} - 1\,.$$

Theoretical Analysis

First non-trivial classical algorithm challenging Linear XEB

Theorem.

Let C be an n-qubit depth-d circuit with (i) lightcone size at most L and (ii) each 2-qubit gate is Haar random, then the algorithm outputs a distribution p in time $\mathrm{poly}(n, 2^L)$ and

$$\mathop{\mathbb{E}}_{C}[\mathcal{F}_C(p)] \geq (1+15^{-d})^{n/L} - 1\,.$$

Corollary.

For Google's 2D nearest neighbors circuit (with Haar random gates), when d=O(log n), our algorithm outputs a distribution p in time and

Theoretical Analysis

First non-trivial classical algorithm challenging Linear XEB

Theorem.

Let C be an n-qubit depth-d circuit with (i) lightcone size at most L and (ii) each 2-qubit gate is Haar random, then the algorithm outputs a distribution p in time $\text{poly}(n, 2^L)$ and

$$\mathop{\mathbb{E}}_{C}[\mathcal{F}_C(p)] \geq (1 + 15^{-d})^{n/L} - 1\,.$$

Corollary.

For Google's 2D nearest neighbors circuit (with Haar random gates), when d=O(log n), our algorithm outputs a distribution p in time $2^{O(d^2)}$ and

Theoretical Analysis

First non-trivial classical algorithm challenging Linear XEB

Theorem.

Let C be an n-qubit depth-d circuit with (i) lightcone size at most L and (ii) each 2-qubit gate is Haar random, then the algorithm outputs a distribution p in time $\text{poly}(n, 2^L)$ and

$$\mathbb{E}_C[\mathcal{F}_C(p)] \geq (1 + 15^{-d})^{n/L} - 1 \,.$$

Corollary.

For Google's 2D nearest neighbors circuit (with Haar random gates), when d=O(log n), our algorithm outputs a distribution p in time $2^{O(d^2)}$ and

$$\mathbb{E}_C[\mathcal{F}_C(p)] \geq \frac{1}{\text{poly}(n)} \,.$$

Intuition

Intuition

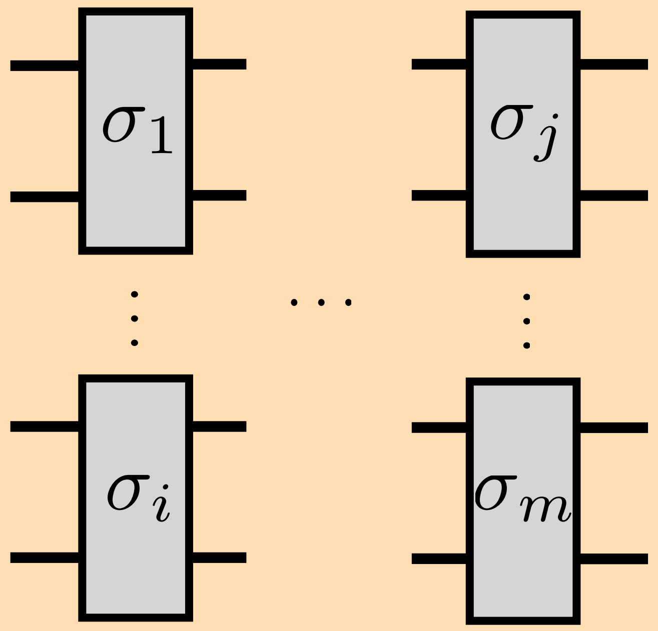
Linear XEB
as a Random Walk

Intuition

Linear XEB as a Random Walk

$$\mathop{\mathbb{E}}_{C}[\mathcal{F}_C(q_C)] =$$

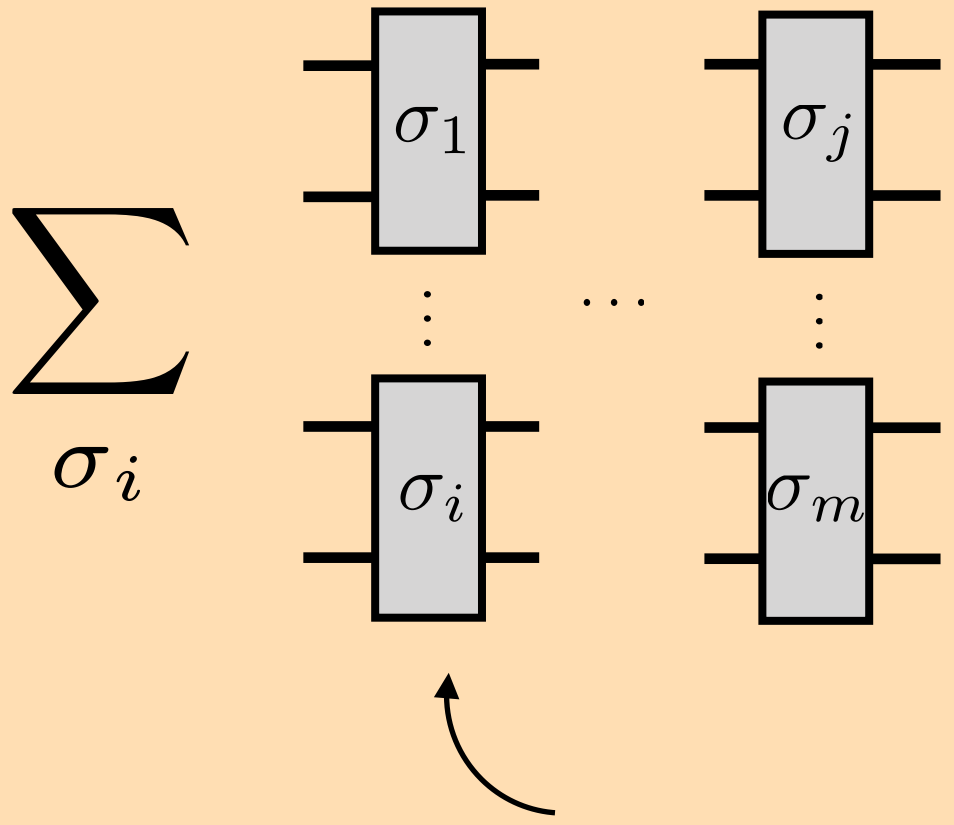
$$\sum_{\sigma_i}$$



Intuition

Linear XEB as a Random Walk

$$\mathop{\mathbb{E}}_{C}[\mathcal{F}_C(q_C)] =$$

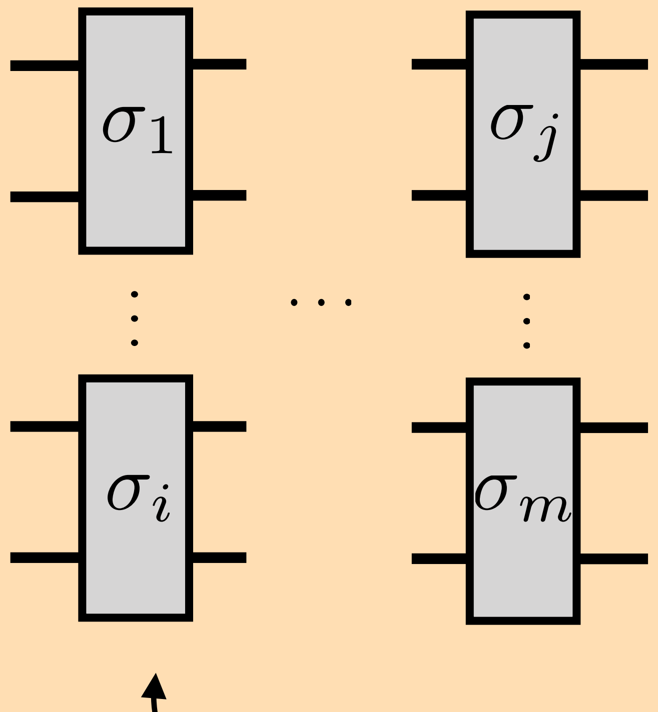


Certain non-negative tensor network

Intuition

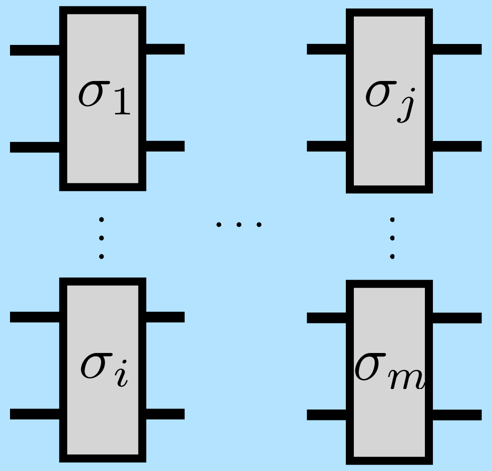
Linear XEB as a Random Walk

$$\mathop{\mathbb{E}}_{C}[\mathcal{F}_C(q_C)] = \sum_{\sigma_i}$$

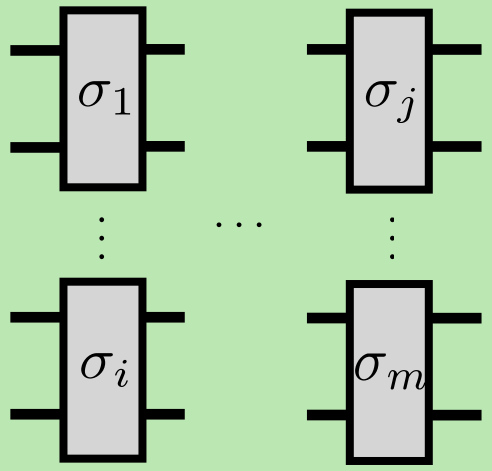


Certain non-negative tensor network

Google's Noisy Simulation



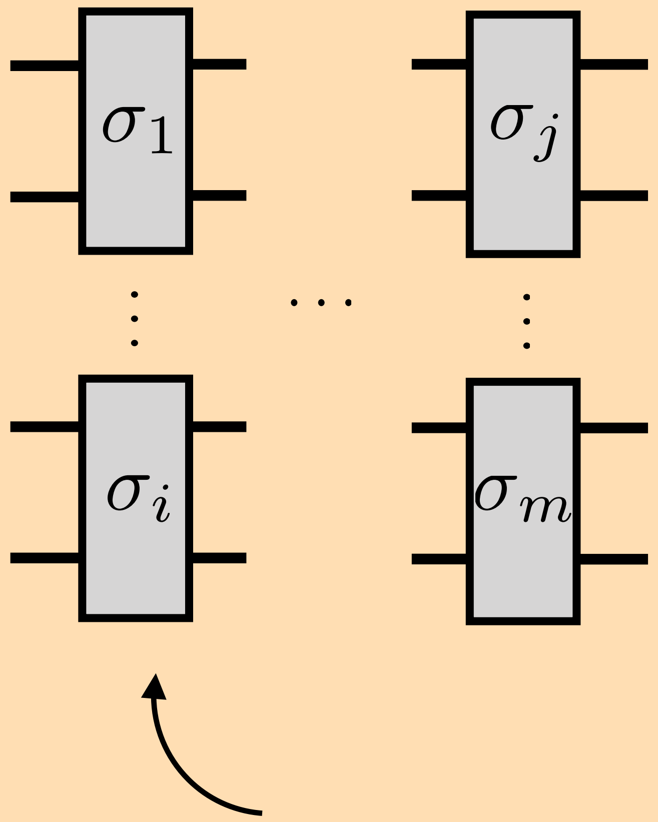
Our Lightcone Algorithm



Intuition

Linear XEB as a Random Walk

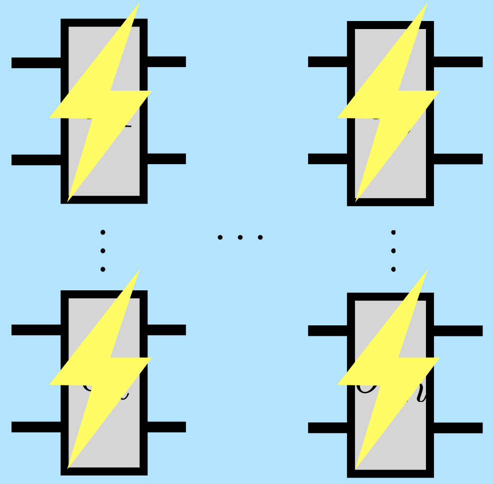
$$\mathop{\mathbb{E}}_{C}[\mathcal{F}_C(q_C)] = \sum_{\sigma_i}$$



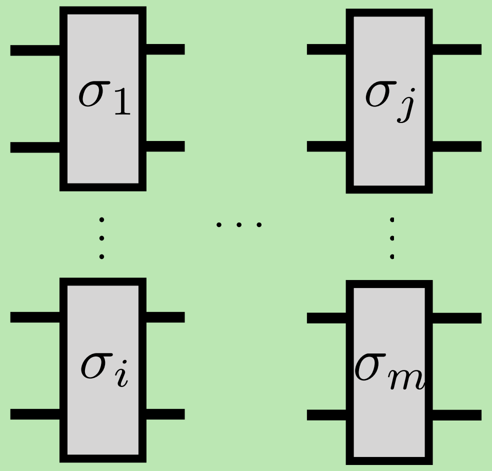
Certain non-negative tensor network

Google's Noisy Simulation

Noises are inherent in the real-world quantum devices.



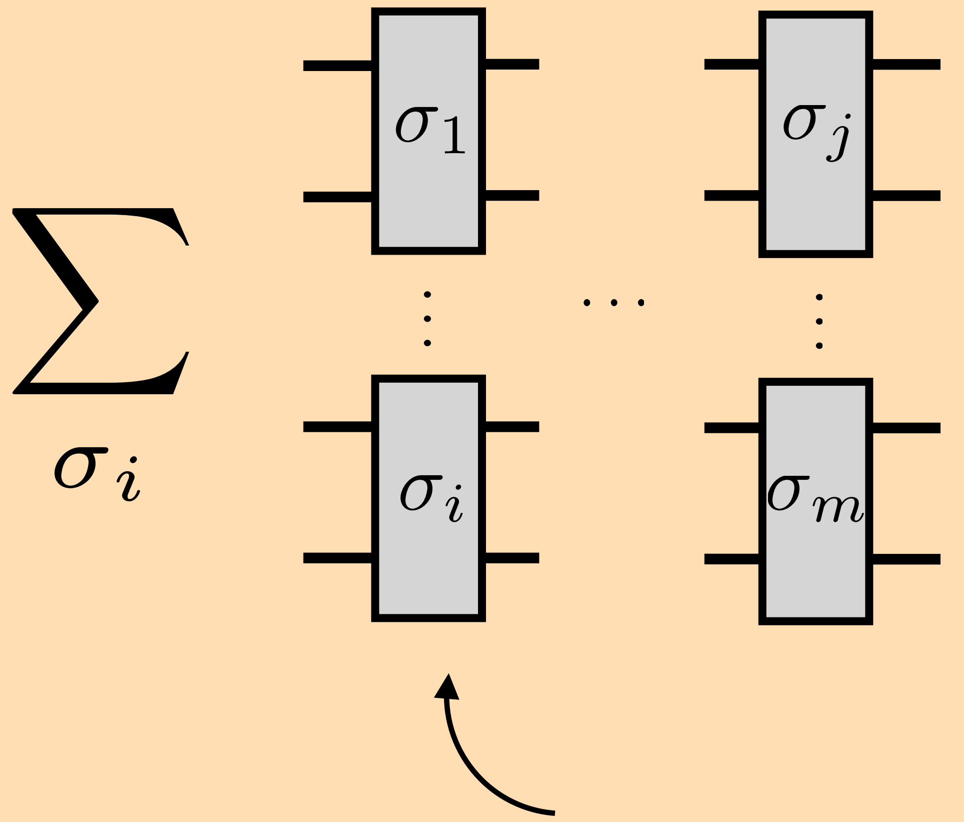
Our Lightcone Algorithm



Intuition

Linear XEB as a Random Walk

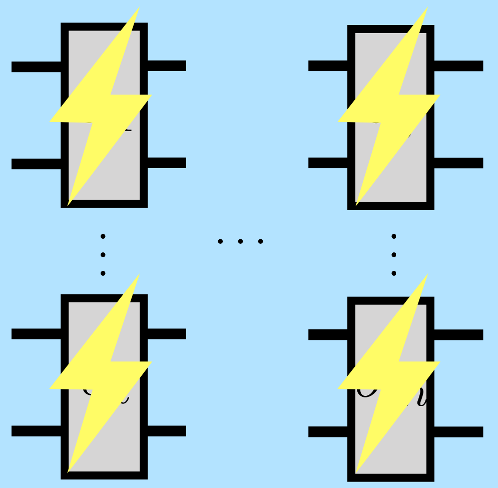
$$\mathbb{E}_{C}[\mathcal{F}_C(q_C)] =$$



Certain non-negative tensor network

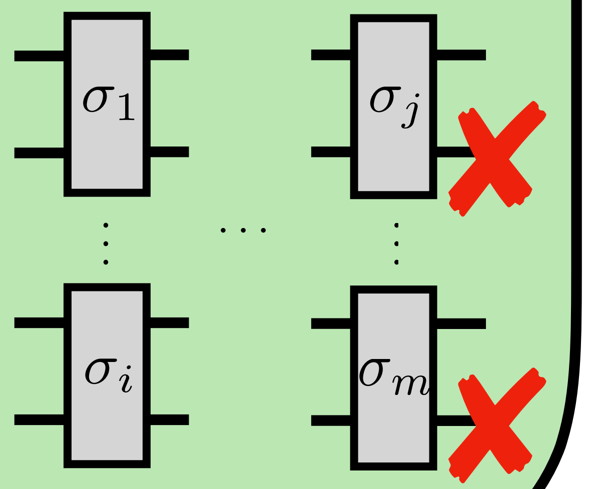
Google's Noisy Simulation

Noises are inherent in the real-world quantum devices.



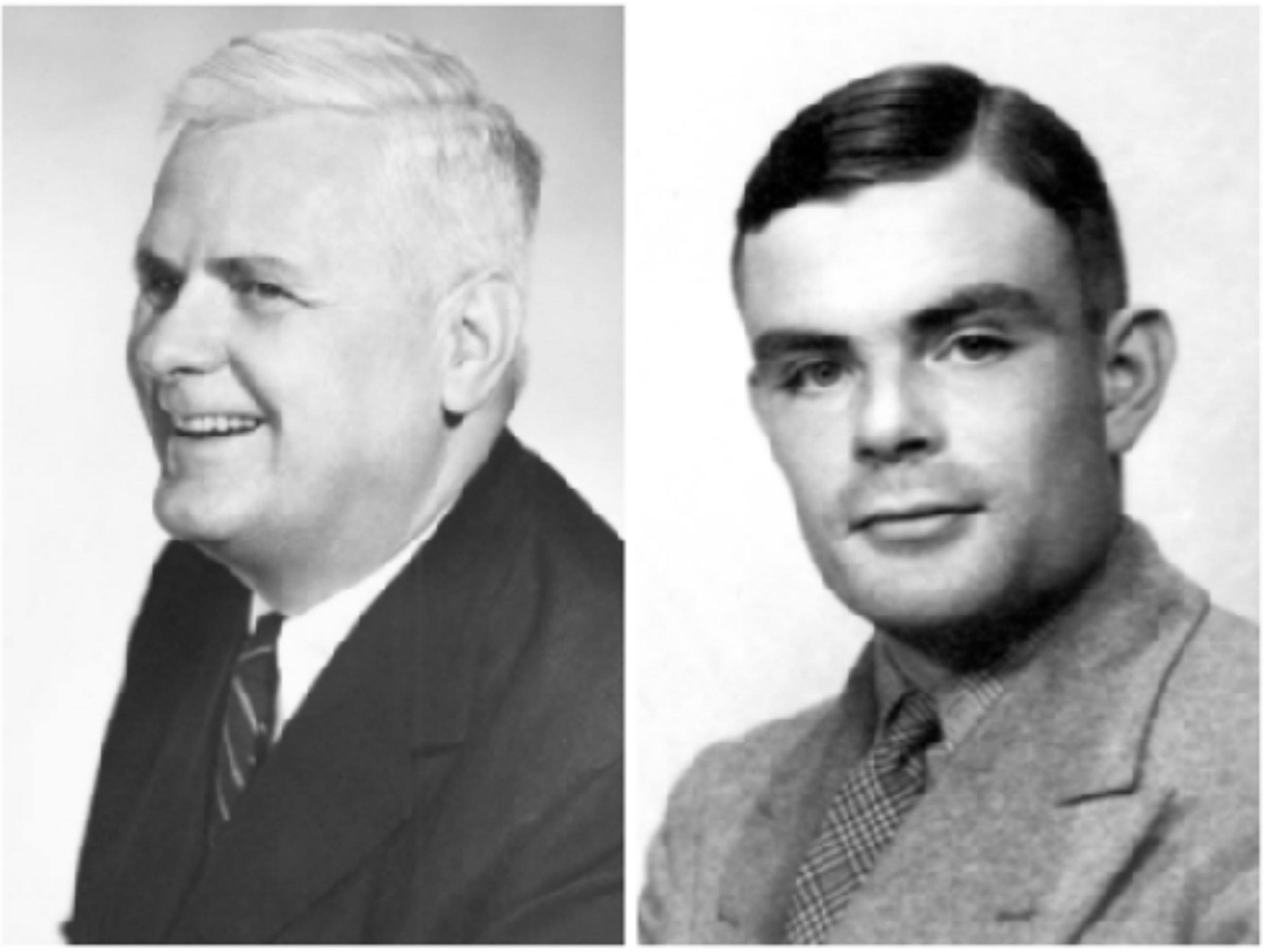
Our Lightcone Algorithm

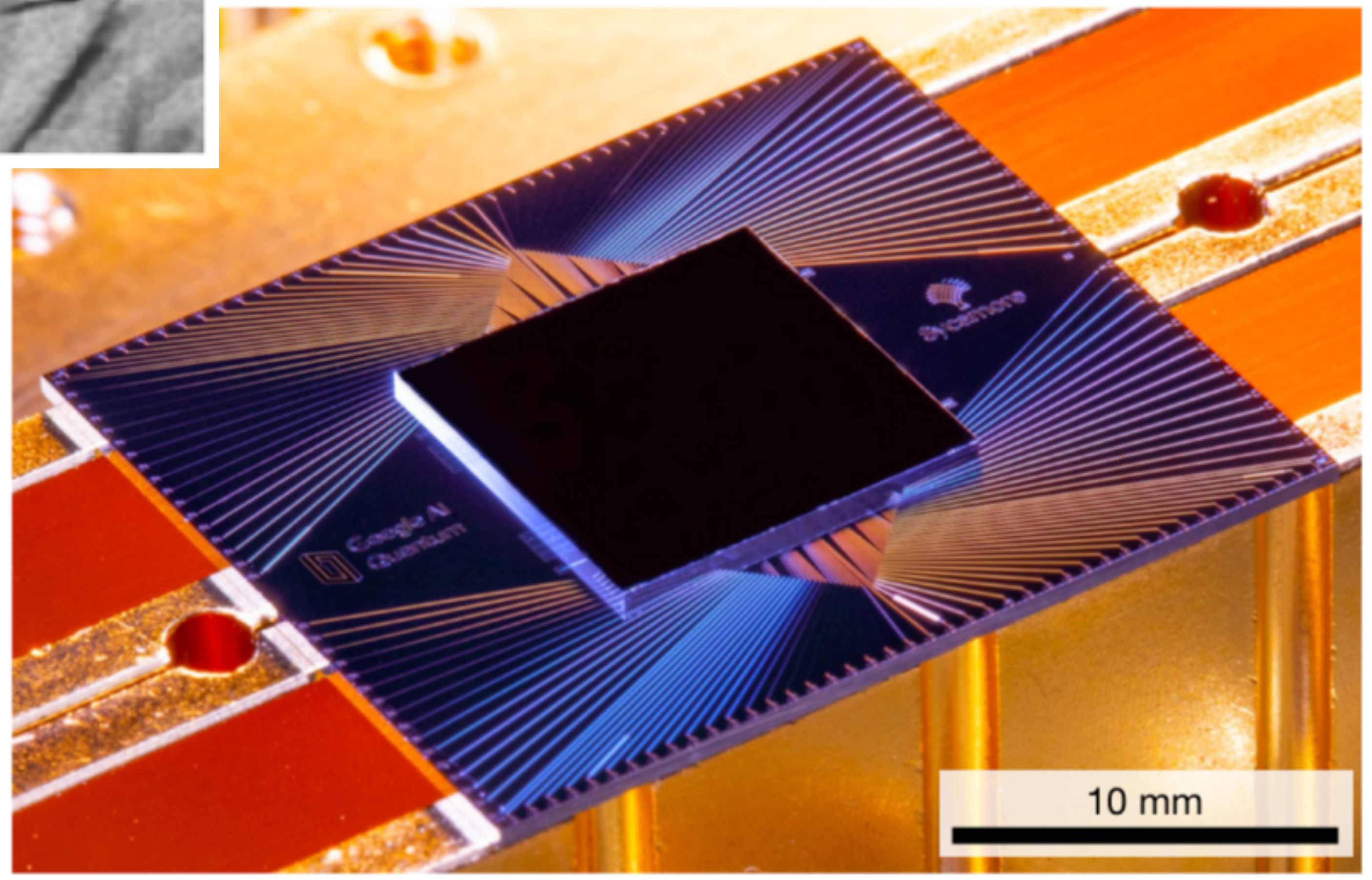
Picking disjoint lightcones = adding noise to non-chosen output qubits.



Perspectives

We are still few steps away from the quantum supremacy regime?

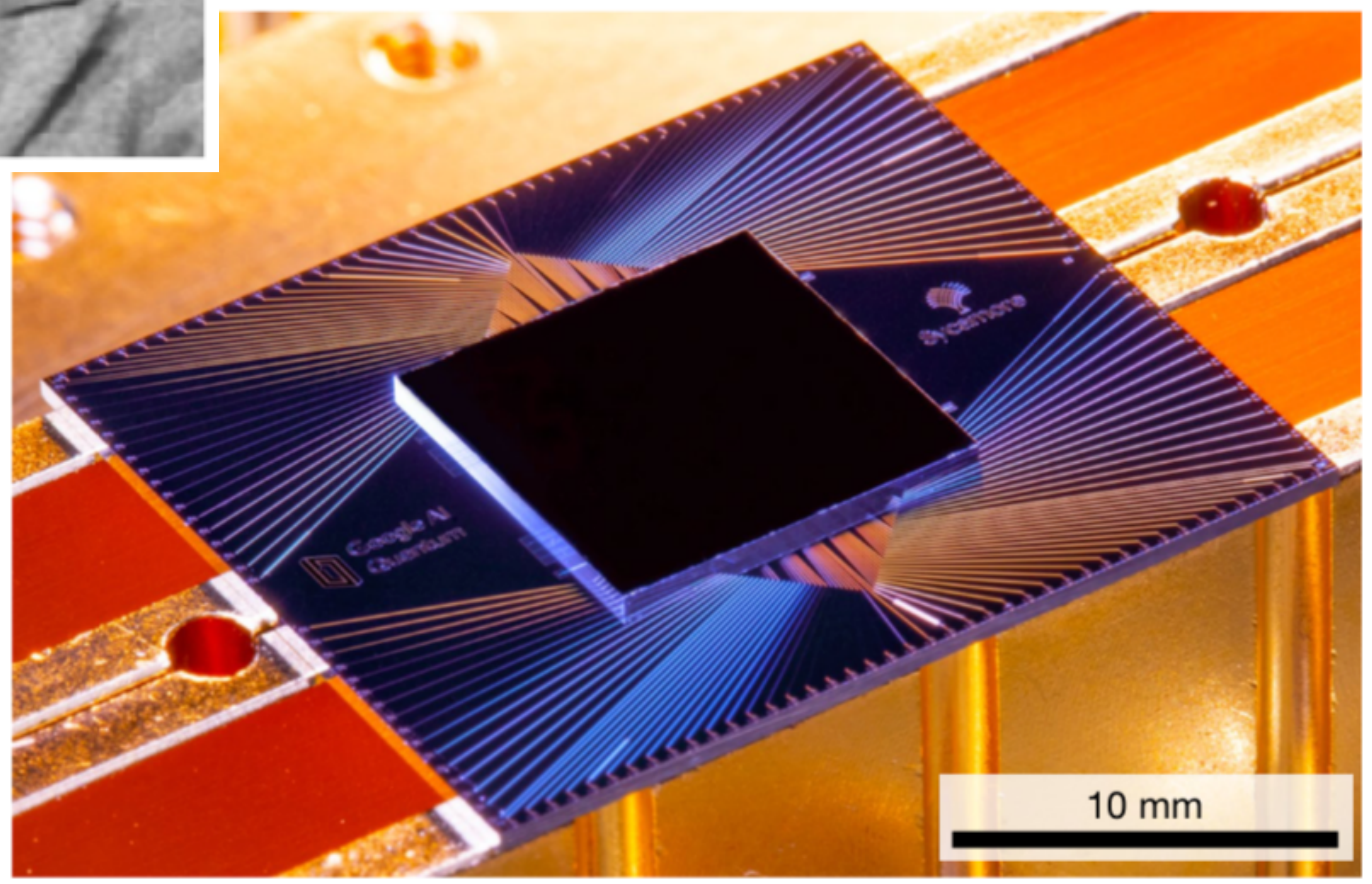




Perspectives

We are still few steps away from the quantum supremacy regime?

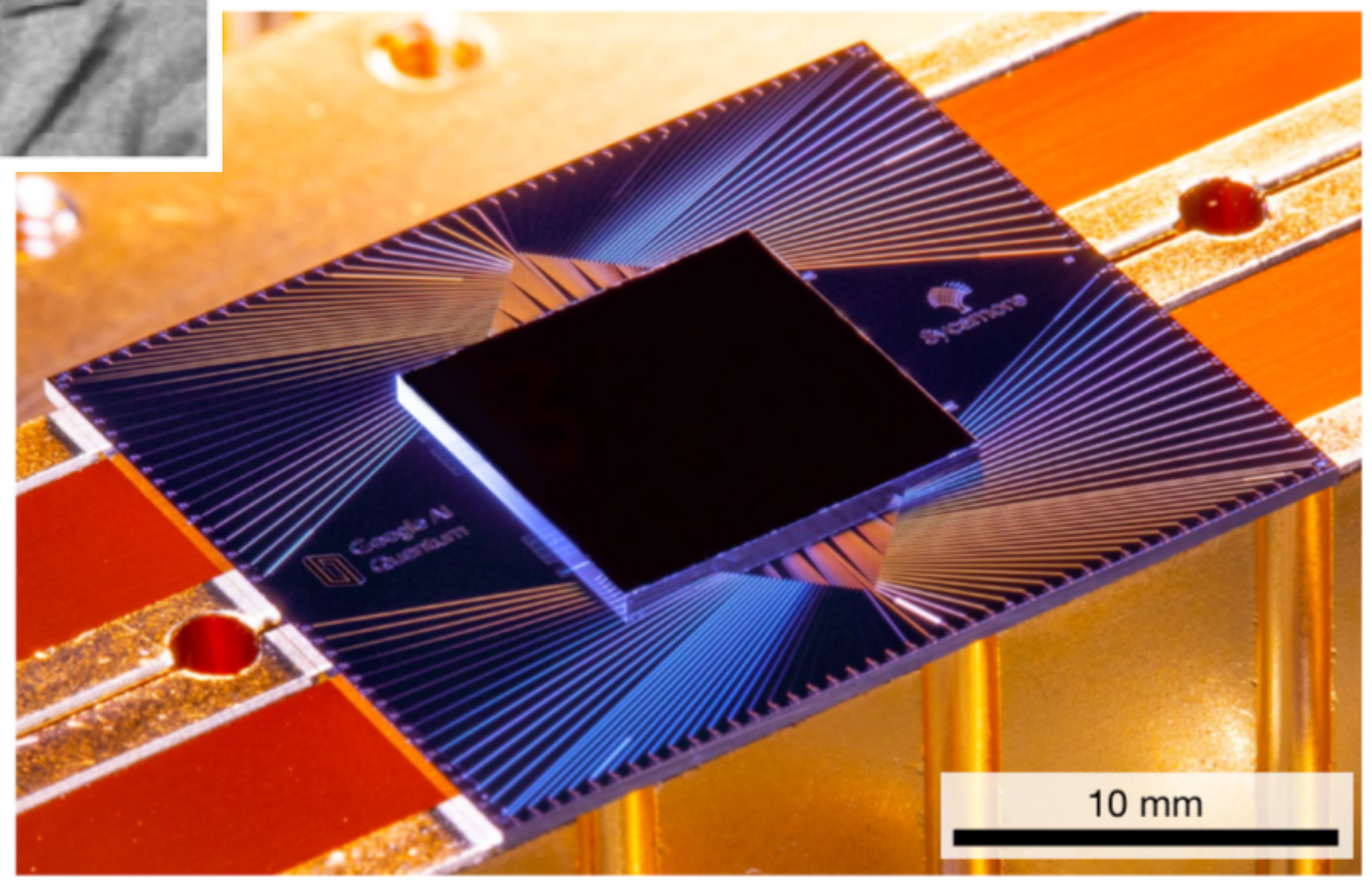
- There is a classical algorithm spoofing Linear XEB (in shallow circuits)!



Perspectives

We are still few steps away from the quantum supremacy regime?

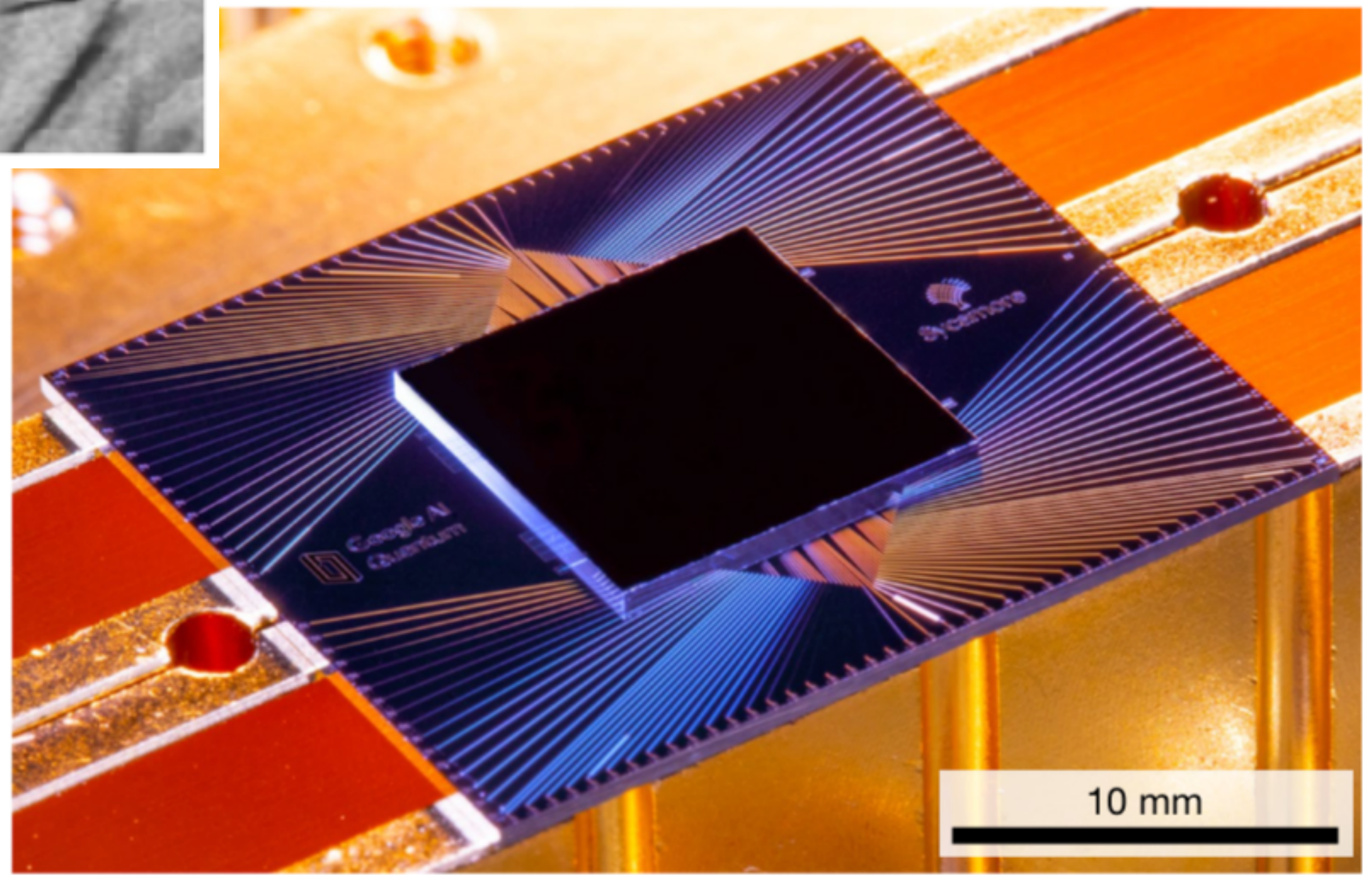
- There is a classical algorithm spoofing Linear XEB (in shallow circuits)!
- Noise in real-world quantum devices is essential and subtle.



Perspectives

We are still few steps away from the quantum supremacy regime?

- There is a classical algorithm spoofing Linear XEB (in shallow circuits)!
- Noise in real-world quantum devices is essential and subtle.
- A random walk picture for Linear XEB.



Future Directions

Quantum supremacy requires both theory and experiment!

Future Directions

Quantum supremacy requires both theory and experiment!

Theory

Experiment

New Proposal?

Future Directions

Quantum supremacy requires both theory and experiment!

Theory

- Deeper understanding in Linear XEB.

Experiment

New Proposal?

Future Directions

Quantum supremacy requires both theory and experiment!

Theory

- Deeper understanding in Linear XEB.
- Classical hardness
  [Aaronson-Chen 17] [Bouland et al. 19][Aaronson-Guu 20].

Experiment

New Proposal?

Future Directions

Quantum supremacy requires both theory and experiment!

Theory

- Deeper understanding in Linear XEB.
- Classical hardness
  [Aaronson-Chen 17] [Bouland et al. 19][Aaronson-Guu 20].

Experiment

- Increase #qubits, better noise-tolerance, etc.

New Proposal?

Future Directions

Quantum supremacy requires both theory and experiment!

Theory

- Deeper understanding in Linear XEB.
- Classical hardness
  [Aaronson-Chen 17] [Bouland et al. 19][Aaronson-Guu 20].

Experiment

- Increase #qubits, better noise-tolerance, etc.
- Faster verification.

New Proposal?

Future Directions

Quantum supremacy requires both theory and experiment!

Theory

- Deeper understanding in Linear XEB.
- Classical hardness
  [Aaronson-Chen 17] [Bouland et al. 19][Aaronson-Guu 20].

Experiment

- Increase #qubits, better noise-tolerance, etc.
- Faster verification.
- Classical simulation.

New Proposal?

Future Directions

Quantum supremacy requires both theory and experiment!

Theory

- Deeper understanding in Linear XEB.
- Classical hardness
  [Aaronson-Chen 17] [Bouland et al. 19][Aaronson-Guu 20].

Experiment

- Increase #qubits, better noise-tolerance, etc.
- Faster verification.
- Classical simulation.

New Proposal?

- More scalable quantum supremacy protocol?

Future Directions

Quantum supremacy requires both theory and experiment!

Theory

- Deeper understanding in Linear XEB.
- Classical hardness
  [Aaronson-Chen 17] [Bouland et al. 19][Aaronson-Guu 20].

Experiment

- Increase #qubits, better noise-tolerance, etc.
- Faster verification.
- Classical simulation.

New Proposal?

- More scalable quantum supremacy protocol?
- More solid complexity-theoretic foundation on the classical hardness?

Future Directions

Quantum supremacy requires both theory and experiment!

Theory

- Deeper understanding in Linear XEB.
- Classical hardness
  [Aaronson-Chen 17] [Bouland et al. 19][Aaronson-Guu 20].

Experiment

- Increase #qubits, better noise-tolerance, etc.
- Faster verification.
- Classical simulation.

New Proposal?

- More scalable quantum supremacy protocol?
- More solid complexity-theoretic foundation on the classical hardness?

Thanks for your attention!