A one-query lower bound for unitary synthesis and breaking quantum cryptography

Fermi Ma

(Simons and Berkeley)

joint work with Alex Lombardi and John Wright

In complexity theory, problems have **classical inputs and outputs**.

In complexity theory, problems have **classical inputs and outputs**.

1) Given a 3-SAT formula ϕ, decide whether it is satisfiable.

In complexity theory, problems have **classical inputs and outputs**.

1) Given a 3-SAT formula ϕ, decide whether it is satisfiable.
2) Given a graph G, output a cycle that visits every vertex once.

In complexity theory, problems have **classical inputs and outputs**.

1) Given a 3-SAT formula ϕ, decide whether it is satisfiable.
2) Given a graph G, output a cycle that visits every vertex once.

This is even true for **quantum** complexity classes like BQP and QMA.

In complexity theory, problems have **classical inputs and outputs**.

1) Given a 3-SAT formula ϕ, decide whether it is satisfiable.
2) Given a graph G, output a cycle that visits every vertex once.

This is even true for **quantum** complexity classes like BQP and QMA.

3) Given a local Hamiltonian H, decide whether it has a low-energy ground state (QMA-complete).

In complexity theory, problems have **classical inputs and outputs**.

1) Given a 3-SAT formula ϕ, decide whether it is satisfiable.

2) Given a graph G, output a cycle that visits every vertex once.

This is even true for **quantum** complexity classes like BQP and QMA.

3) Given a local Hamiltonian H, decide whether it has a low-energy ground state (QMA-complete).

Key point: even though this problem is "about" quantum states, the input and output are classical.

But some problems have **quantum inputs/outputs**.

But some problems have **quantum inputs/outputs**.

- **(Search-QMA)** Given a local Hamiltonian, *output* its ground state.

But some problems have **quantum inputs/outputs**.

- **(Search-QMA)** Given a local Hamiltonian, *output* its ground state.
- **(Quantum State Tomography)** Given copies of an unknown quantum state $|\psi\rangle$, output a classical description of $|\psi\rangle$.

But some problems have **quantum inputs/outputs**.

- **(Search-QMA)** Given a local Hamiltonian, *output* its ground state.
- **(Quantum State Tomography)** Given copies of an unknown quantum state $|\psi\rangle$, output a classical description of $|\psi\rangle$.
- **(Distinguishing Mixed States)** Distinguish mixed states ρ_0, ρ_1, given one at random. (mixed state = distribution over pure states)

But some problems have **quantum inputs/outputs**.

- **(Search-QMA)** Given a local Hamiltonian, *output* its ground state.
- **(Quantum State Tomography)** Given copies of an unknown quantum state $|\psi\rangle$, output a classical description of $|\psi\rangle$.
- **(Distinguishing Mixed States)** Distinguish mixed states ρ_0, ρ_1, given one at random. (mixed state = distribution over pure states)

Many other examples: decoding black-hole radiation, AdS/CFT map.

But some problems have **quantum inputs/outputs**.

- **(Search-QMA)** Given a local Hamiltonian, *output* its ground state.
- **(Quantum State Tomography)** Given copies of an unknown quantum state $|\psi\rangle$, output a classical description of $|\psi\rangle$.
- **(Distinguishing Mixed States)** Distinguish mixed states ρ_0, ρ_1, given one at random. (mixed state = distribution over pure states)

Many other examples: decoding black-hole radiation, AdS/CFT map.

This talk:

How hard are “inherently quantum” problems?

Before we continue:

1-minute detour for quantum computing 101

Quantum Computing 101

Quantum Computing 101

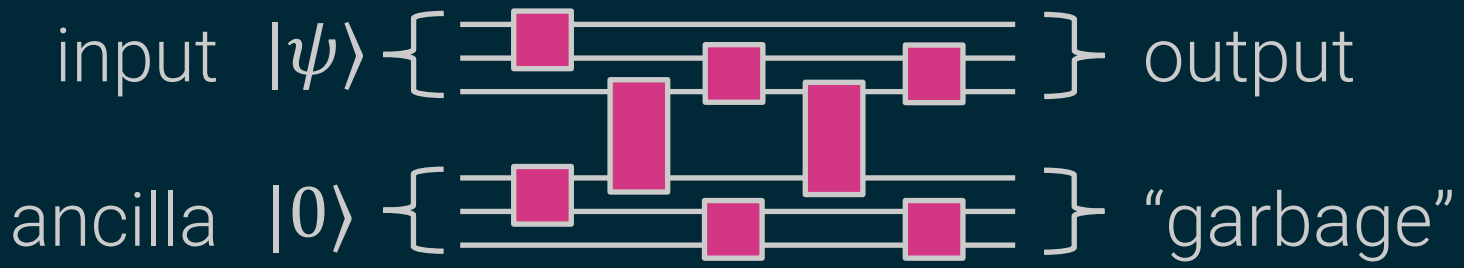
- n-qubit pure state $= 2^n$-dim unit vector $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.

Quantum Computing 101

- n-qubit pure state $= 2^n$-dim unit vector $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.
- n-qubit unitary $= 2^n \times 2^n$ complex rotation matrix.

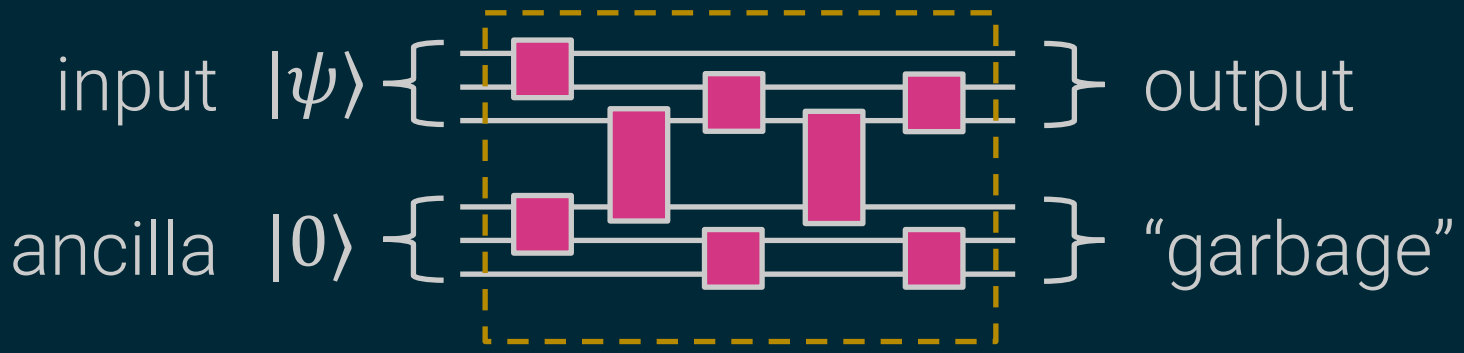
Quantum Computing 101

- n-qubit pure state $= 2^n$-dim unit vector $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.
- n-qubit unitary $= 2^n \times 2^n$ complex rotation matrix.
- quantum computers are modeled as quantum circuits:



Quantum Computing 101

- n-qubit pure state $= 2^n$-dim unit vector $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.
- n-qubit unitary $= 2^n \times 2^n$ complex rotation matrix.
- quantum computers are modeled as quantum circuits:



efficient unitary $= \mathbf{poly}(n)$-size circuit

Now back to:

How hard are inherently quantum problems?

How hard are inherently quantum problems?

Standard procedure: try to reduce your problem to a well-studied complexity class.

How hard are inherently quantum problems?

Standard procedure: try to reduce your problem to a well-studied complexity class.

Ex: is the problem easy given an $\mathbf{NP}$ oracle? a $\mathbf{PSPACE}$ oracle?

How hard are inherently quantum problems?

Standard procedure: try to reduce your problem to a well-studied complexity class.

Ex: is the problem easy given an $\mathbf{NP}$ oracle? a $\mathbf{PSPACE}$ oracle?

This works for some inherently quantum problems.

How hard are inherently quantum problems?

Standard procedure: try to reduce your problem to a well-studied complexity class.

Ex: is the problem easy given an $\mathbf{NP}$ oracle? a $\mathbf{PSPACE}$ oracle?

This works for some inherently quantum problems.

(Search-QMA) Given a local Hamiltonian, output its ground state.

How hard are inherently quantum problems?

Standard procedure: try to reduce your problem to a well-studied complexity class.

Ex: is the problem easy given an $\mathbf{NP}$ oracle? a $\mathbf{PSPACE}$ oracle?

This works for some inherently quantum problems.

(Search-QMA) Given a local Hamiltonian, output its ground state.

[INNRY22]: Search-QMA is easy given one query to a $\mathbf{PP}$ oracle.

How hard are inherently quantum problems?

Standard procedure: try to reduce your problem to a well-studied complexity class.

Ex: is the problem easy given an $\mathbf{NP}$ oracle? a $\mathbf{PSPACE}$ oracle?

But now, try this for the state distinguishing problem:

How hard are inherently quantum problems?

Standard procedure: try to reduce your problem to a well-studied complexity class.

Ex: is the problem easy given an $\mathbf{NP}$ oracle? a $\mathbf{PSPACE}$ oracle?

But now, try this for the state distinguishing problem:

(Distinguishing Mixed States) Distinguish mixed states ρ_0, ρ_1, given one of them at random.

How hard are inherently quantum problems?

Standard procedure: try to reduce your problem to a well-studied complexity class.

Ex: is the problem easy given an $\mathbf{NP}$ oracle? a $\mathbf{PSPACE}$ oracle?

But now, try this for the state distinguishing problem:

(Distinguishing Mixed States) Distinguish mixed states ρ_0, ρ_1, given one of them at random.

Not obvious: Is there ***any*** oracle that makes this easy?

How hard are inherently quantum problems?

Standard procedure: try to reduce your problem to a well-studied complexity class.

Ex: is the problem easy given an $\mathbf{NP}$ oracle? a $\mathbf{PSPACE}$ oracle?

But now, try this for the state distinguishing problem:

(Distinguishing Mixed States) Distinguish mixed states ρ_0, ρ_1, given one of them at random.

Not obvious: Is there ***any*** oracle that makes this easy? Even an oracle for the halting problem?

Given the ability to solve any **classical problem**,
is it easy to solve every **quantum problem**?

Given the ability to solve any **classical problem**,
is it easy to solve every **quantum problem**?

This is the Unitary Synthesis Problem [AK06].

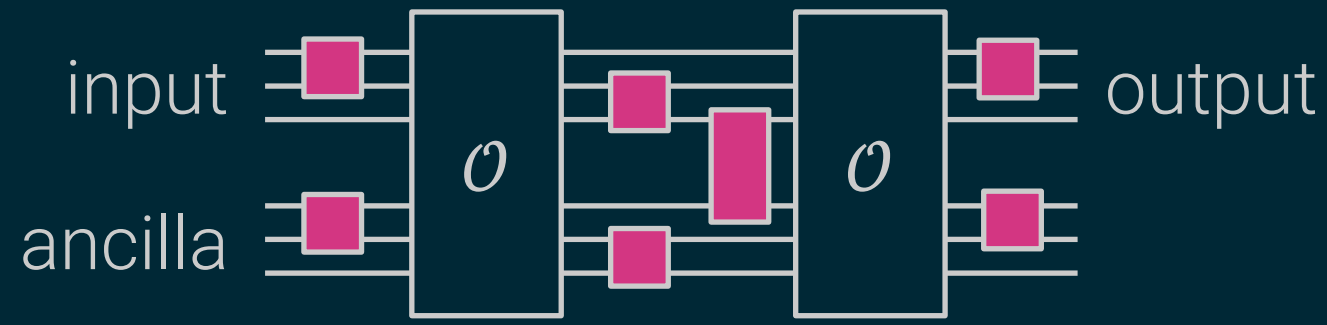
The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ such that for all n-qubit unitaries U, there exists a function f such that A^f implements U?

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ such that for all n-qubit unitaries U, there exists a function f such that A^f implements U?

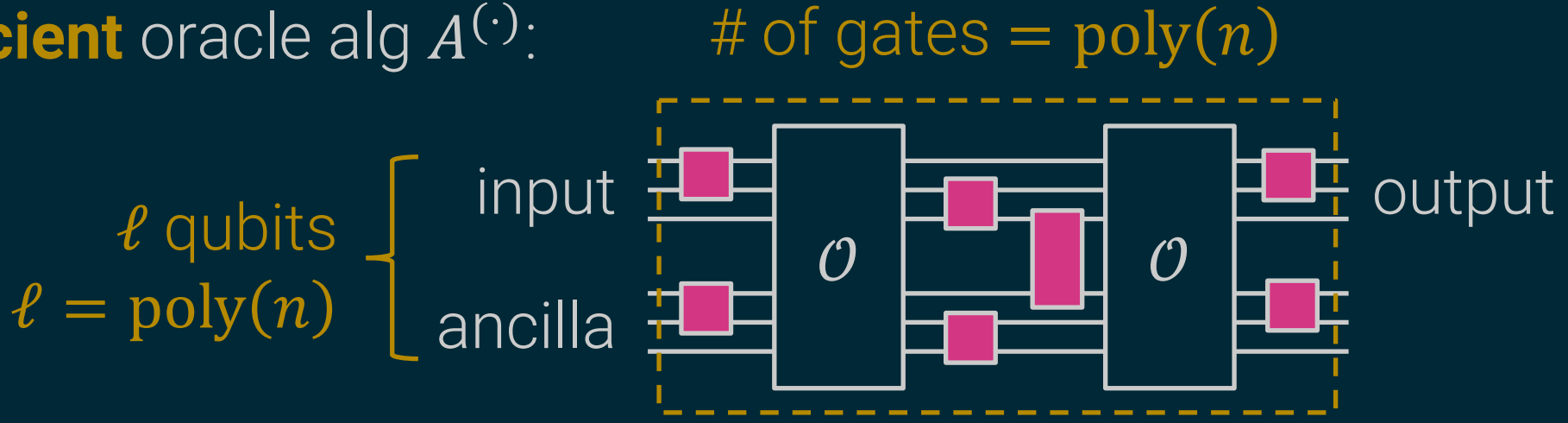
1) Fix efficient oracle alg $A^{(\cdot)}$:



The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ such that for all n-qubit unitaries U, there exists a function f such that A^f implements U?

1) Fix **efficient** oracle alg $A^{(\cdot)}$:

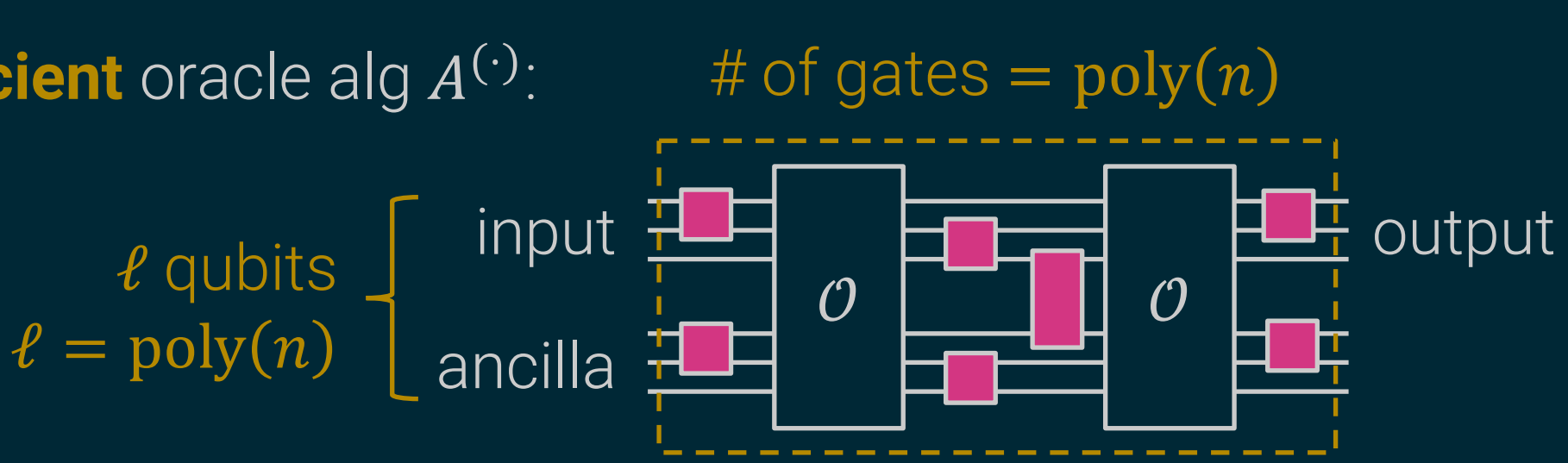


# of gates $= \mathrm{poly}(n)$

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ such that for all n-qubit unitaries U, there exists a function f such that A^f implements U?

1) Fix **efficient** oracle alg $A^{(\cdot)}$:

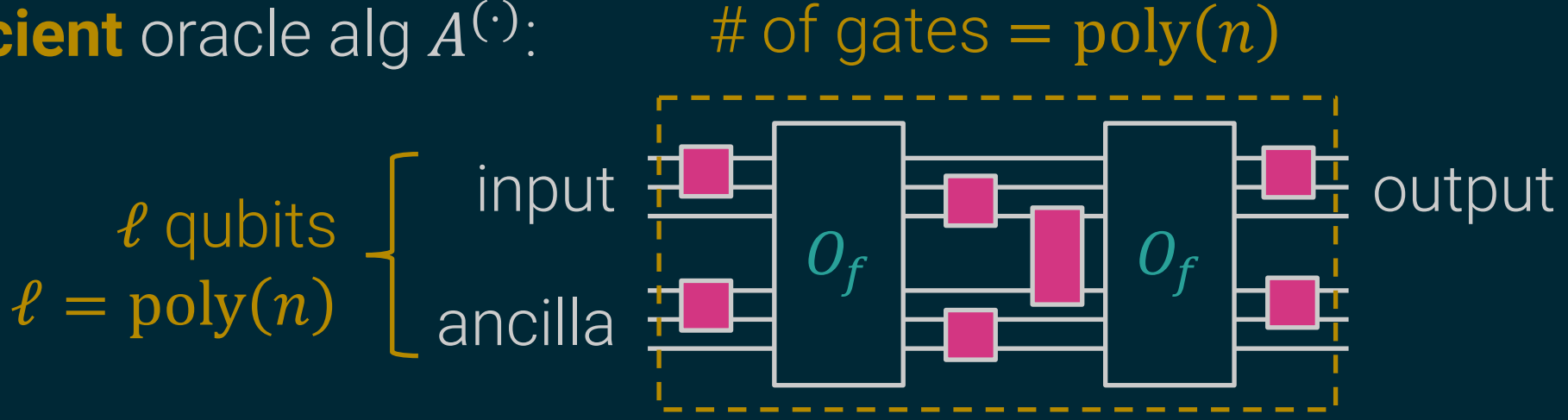


2) Given U, pick some $f: \{0,1\}^\ell \to \{\pm 1\}$.

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ such that for all n-qubit unitaries U, there exists a function f such that A^f implements U?

1) Fix **efficient** oracle alg $A^{(\cdot)}$:

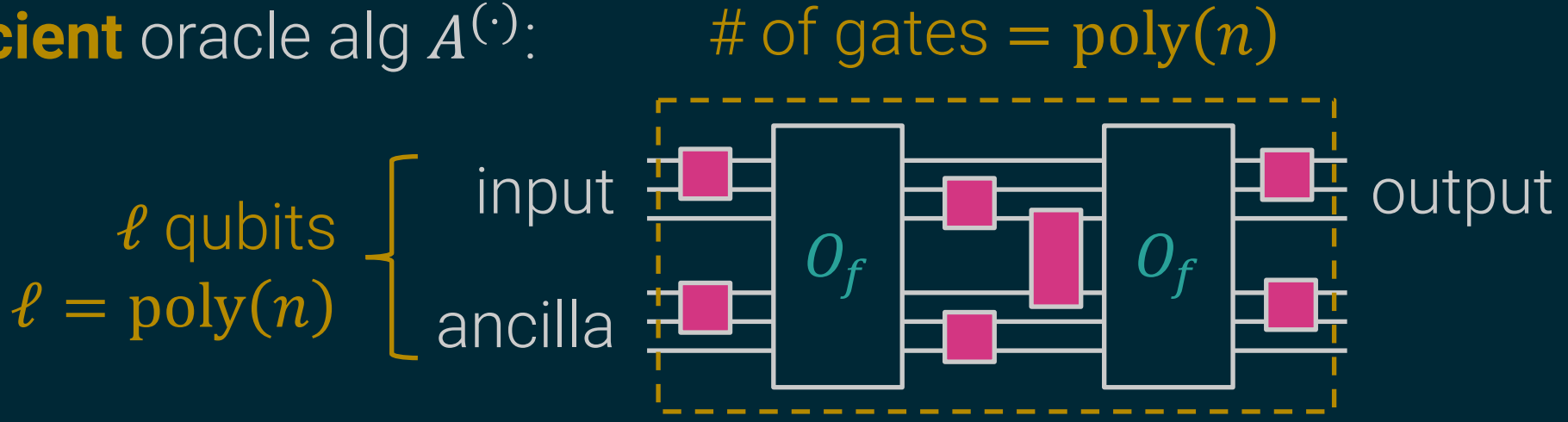


2) Given U, pick some $f: \{0,1\}^\ell \to \{\pm 1\}$. Plug in $O_f: |z\rangle \to f(z) \cdot |z\rangle$.

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ such that for all n-qubit unitaries U, there exists a function f such that A^f implements U?

1) Fix **efficient** oracle alg $A^{(\cdot)}$:



2) Given U, pick some $f: \{0,1\}^\ell \to \{\pm 1\}$. Plug in $O_f: |z\rangle \to f(z) \cdot |z\rangle$.

Rephrased: can quantum problems (unitaries) be **efficiently reduced** to classical problems (functions)?

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ such that for all n-qubit unitaries U, there exists a function f such that A^f implements U?

Prior work: best-known bounds for unitary synthesis

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ such that for all n-qubit unitaries U, there exists a function f such that A^f implements U?

Prior work: best-known bounds for unitary synthesis

Upper bounds

- [Ros22]: $2^{n/2}$-query algorithm based on Grover search

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ such that for all n-qubit unitaries U, there exists a function f such that A^f implements U?

Prior work: best-known bounds for unitary synthesis

Upper bounds

- [Ros22]: $2^{n/2}$-query algorithm based on Grover search
- [INNRY22]: 1-query algorithm if A^f queries f on 2^{2n}-length inputs. Idea: learn description of U with Bernstein-Vazirani algorithm.

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ such that for all n-qubit unitaries U, there exists a function f such that A^f implements U?

Prior work: best-known bounds for unitary synthesis

Upper bounds

- [Ros22]: $2^{n/2}$-query algorithm based on Grover search
- [INNRY22]: 1-query algorithm if A^f queries f on 2^{2n}-length inputs. Idea: learn description of U with Bernstein-Vazirani algorithm.

Lower bounds

- [AK06]: 1-query lower bound for very restricted class of algorithms.

Why has it been hard to prove lower bounds?

Why has it been hard to prove lower bounds?

(1) Counting arguments don’t work.

Why has it been hard to prove lower bounds?

(1) Counting arguments don't work.

- $2^{2^{2n}}$ different n-qubit unitaries (roughly).

Why has it been hard to prove lower bounds?

(1) Counting arguments don't work.

- $2^{2^{2n}}$ different n-qubit unitaries (roughly).
- $2^{2^{\ell}}$ different functions $f:\{0,1\}^{\ell} \to \{\pm 1\}$.

Why has it been hard to prove lower bounds?

(1) Counting arguments don't work.

- $2^{2^{2n}}$ different n-qubit unitaries (roughly).
- $2^{2^{\ell}}$ different functions $f: \{0,1\}^{\ell} \to \{\pm 1\}$.

Not useful when $\ell > 2n$.

Why has it been hard to prove lower bounds?

(1) Counting arguments don't work.

- $2^{2^{2n}}$ different n-qubit unitaries (roughly).
- $2^{2^{\ell}}$ different functions $f: \{0,1\}^{\ell} \to \{\pm 1\}$.

Not useful when $\ell > 2n$.

(2) Even ***one-query*** algorithms are very powerful!

Why has it been hard to prove lower bounds?

(1) Counting arguments don't work.

- $2^{2^{2n}}$ different n-qubit unitaries (roughly).
- $2^{2^{\ell}}$ different functions $f:\{0,1\}^{\ell} \to \{\pm 1\}$.

Not useful when $\ell > 2n$.

(2) Even ***one-query*** algorithms are very powerful!

In fact, they can solve any **classical input**, **quantum output** problem.
[Aar16, INNRY22, Ros23]

This work

This work

Main result: There is no efficient one-query oracle algorithm $A^{(\cdot)}$ for the Unitary Synthesis Problem.

This work

Main result: There is no efficient one-query oracle algorithm $A^{(\cdot)}$ for the Unitary Synthesis Problem.

In fact, we rule out oracle algorithms $A^{(\cdot)}$ with:

- unlimited space (number of qubits)

This work

Main result: There is no efficient one-query oracle algorithm $A^{(\cdot)}$ for the Unitary Synthesis Problem.

In fact, we rule out oracle algorithms $A^{(\cdot)}$ with:

- unlimited space (number of qubits)
- unlimited size (number of quantum gates)

This work

Main result: There is no efficient one-query oracle algorithm $A^{(\cdot)}$ for the Unitary Synthesis Problem.

In fact, we rule out oracle algorithms $A^{(\cdot)}$ with:

- unlimited space (number of qubits)
- unlimited size (number of quantum gates)
- one query to a function $f:\{0,1\}^{\ell} \to \{\pm 1\}$ where $\ell = o(2^n)$. (Bernstein-Vazirani algorithm requires $\ell = 2^{2n}$)

Plan for this talk

- Relating unitary synthesis to breaking quantum cryptography
- The Oracle State Distinguishing Game
- If time: our proof for a special class of adversaries

Plan for this talk

- **Relating unitary synthesis to breaking quantum cryptography**
- The Oracle State Distinguishing Game
- If time: our proof for a special class of adversaries

It turns out there is a deep connection between the Unitary Synthesis Problem and the following question:

How hard is it to break quantum cryptography?

It turns out there is a deep connection between the Unitary Synthesis Problem and the following question:

How hard is it to break quantum cryptography?

In fact, we obtain our unitary synthesis lower bound by proving a statement about quantum cryptography.

It turns out there is a deep connection between the Unitary Synthesis Problem and the following question:

How hard is it to break quantum cryptography?

In fact, we obtain our unitary synthesis lower bound by proving a statement about quantum cryptography.

Next: brief crypto background + explain this connection

Cryptographic pseudorandomness

Pseudorandom generator (PRG): Set of $K \ll 2^n$ efficiently constructible n-bit strings $\{x_k\}_{k\in[K]}$ s.t. the following are indistinguishable:

(1) Output random x_k.

(2) Output random n-bit string.

Cryptographic pseudorandomness

Pseudorandom generator (PRG): Set of $K \ll 2^n$ efficiently constructible n-bit strings $\{x_k\}_{k\in[K]}$ s.t. the following are indistinguishable:

(1) Output random x_k.

(2) Output random n-bit string.

Indistinguishability only possible against **efficient** adversaries.

Inefficient adversary can check if input is in $\{x_k\}_{k\in[K]}$.

Cryptographic pseudorandomness

Pseudorandom generator (PRG): Set of $K \ll 2^n$ efficiently constructible n-bit strings $\{x_k\}_{k\in[K]}$ s.t. the following are indistinguishable:

(1) Output random x_k.

(2) Output random n-bit string.

Indistinguishability only possible against **efficient** adversaries.

Inefficient adversary can check if input is in $\{x_k\}_{k\in[K]}$.

Existence of PRGs is equivalent to existence of many important crypto primitives (ex: private-key encryption).

Cryptographic pseudorandomness

Pseudorandom generator (PRG): Set of $K \ll 2^n$ efficiently constructible n-bit strings $\{x_k\}_{k\in[K]}$ s.t. the following are indistinguishable:

(1) Output random x_k.

(2) Output random n-bit string.

Breaking a PRG is a classical input, classical output task.

Cryptographic pseudorandomness

Pseudorandom generator (PRG): Set of $K \ll 2^n$ efficiently constructible n-bit strings $\{x_k\}_{k\in[K]}$ s.t. the following are indistinguishable:

(1) Output random x_k.

(2) Output random n-bit string.

Breaking a PRG is a classical input, classical output task.

If $\mathrm{P} = \mathrm{NP}$, all PRGs are broken:

- Given any $\mathbf{NP}$-complete $f\colon\{0,1\}^* \to \{\pm 1\}$, there is a poly-time oracle algorithm A^f that breaks any PRG (Turing reduction).

Cryptographic pseudorandomness

Pseudorandom states (PRS): Set of $K \ll 2^n$ efficiently constructible n-qubit states $\{|\psi_k\rangle\}_{k\in[K]}$ such that the following are indistinguishable:

(1) Output random $|\psi_k\rangle$.

(2) Output random n-qubit state.

Cryptographic pseudorandomness

Pseudorandom states (PRS): Set of $K \ll 2^n$ efficiently constructible n-qubit states $\{|\psi_k\rangle\}_{k\in[K]}$ such that the following are indistinguishable:

(1) Output random $|\psi_k\rangle$.

(2) Output random n-qubit state.

Indistinguishability only possible against **efficient** adversaries.

Cryptographic pseudorandomness

Pseudorandom states (PRS): Set of $K \ll 2^n$ efficiently constructible n-qubit states $\{|\psi_k\rangle\}_{k\in[K]}$ such that the following are indistinguishable:

(1) Output random $|\psi_k\rangle$.

(2) Output random n-qubit state.

Indistinguishability only possible against **efficient** adversaries.

Inefficient adversary can measure whether a given state is in the K-dimensional subspace $\mathrm{S} = \mathbf{span}\{|\psi_k\rangle\}_{k\in[K]}$.

Cryptographic pseudorandomness

Pseudorandom states (PRS): Set of $K \ll 2^n$ efficiently constructible n-qubit states $\{|\psi_k\rangle\}_{k\in[K]}$ such that the following are indistinguishable:

(1) Output random $|\psi_k\rangle$.

(2) Output random n-qubit state.

Indistinguishability only possible against **efficient** adversaries.

Inefficient adversary can measure whether a given state is in the K-dimensional subspace $\mathrm{S} = \mathbf{span}\{|\psi_k\rangle\}_{k\in[K]}$.

Existence of PRS implies many important quantum crypto primitives (ex: bit commitments, secure computation).

Cryptographic pseudorandomness

Pseudorandom states (PRS): Set of $K \ll 2^n$ efficiently constructible n-qubit states $\{|\psi_k\rangle\}_{k\in[K]}$ such that the following are indistinguishable:

(1) Output random $|\psi_k\rangle$.

(2) Output random n-qubit state.

Breaking a PRS is a quantum input, classical output task.

Cryptographic pseudorandomness

Pseudorandom states (PRS): Set of $K \ll 2^n$ efficiently constructible n-qubit states $\{|\psi_k\rangle\}_{k\in[K]}$ such that the following are indistinguishable:

(1) Output random $|\psi_k\rangle$.

(2) Output random n-qubit state.

Breaking a PRS is a quantum input, classical output task.

How hard is it to break a PRS?

Cryptographic pseudorandomness

Pseudorandom states (PRS): Set of $K \ll 2^n$ efficiently constructible n-qubit states $\{|\psi_k\rangle\}_{k\in[K]}$ such that the following are indistinguishable:

(1) Output random $|\psi_k\rangle$.

(2) Output random n-qubit state.

Breaking a PRS is a quantum input, classical output task.

How hard is it to break a PRS? Is it easy given:

- an oracle for NP?

Cryptographic pseudorandomness

Pseudorandom states (PRS): Set of $K \ll 2^n$ efficiently constructible n-qubit states $\{|\psi_k\rangle\}_{k\in[K]}$ such that the following are indistinguishable:

(1) Output random $|\psi_k\rangle$.

(2) Output random n-qubit state.

Breaking a PRS is a quantum input, classical output task.

How hard is it to break a PRS? Is it easy given:

- an oracle for $\mathbf{NP}$?

Unlikely!

[Kre21,KQST23] prove a black-box separation.

Cryptographic pseudorandomness

Pseudorandom states (PRS): Set of $K \ll 2^n$ efficiently constructible n-qubit states $\{|\psi_k\rangle\}_{k\in[K]}$ such that the following are indistinguishable:

(1) Output random $|\psi_k\rangle$.

(2) Output random n-qubit state.

Breaking a PRS is a quantum input, classical output task.

How hard is it to break a PRS? Is it easy given:

- an oracle for NP? Unlikely!
- an oracle for PSPACE? Not known!

[Kre21,KQST23] prove a black-box separation.

Cryptographic pseudorandomness

Pseudorandom states (PRS): Set of $K \ll 2^n$ efficiently constructible n-qubit states $\{|\psi_k\rangle\}_{k\in[K]}$ such that the following are indistinguishable:

(1) Output random $|\psi_k\rangle$.

(2) Output random n-qubit state.

Breaking a PRS is a quantum input, classical output task.

How hard is it to break a PRS? Is it easy given:

- an oracle for NP? Unlikely!
- an oracle for PSPACE? Not known!
- an oracle for ALL? Not known!

[Kre21,KQST23] prove a black-box separation.

Our second main result

Main result #2: Relative to a random oracle R, there exists a PRS secure against any efficient oracle adversary $A^{(\cdot)}$ that makes one query to an arbitrary function f_R, which can depend on R.

Our second main result

Main result #2: Relative to a random oracle R, there exists a PRS secure against any efficient oracle adversary $A^{(\cdot)}$ that makes one query to an arbitrary function f_R, which can depend on R.

Indicates that the existence of quantum crypto could be independent of traditional complexity theory.

Our second main result

Main result #2: Relative to a random oracle R, there exists a PRS secure against any efficient oracle adversary $A^{(\cdot)}$ that makes one query to an arbitrary function f_R, which can depend on R.

Immediate corollary

Main result #1: There is no efficient one-query oracle algorithm $A^{(\cdot)}$ for the Unitary Synthesis Problem.

Our second main result

Main result #2: Relative to a random oracle R, there exists a PRS secure against any efficient oracle adversary $A^{(\cdot)}$ that makes one query to an arbitrary function f_R, which can depend on R.

Immediate corollary

Why? For any PRS, there exists **some** unitary U that can break it.

Main result #1: There is no efficient one-query oracle algorithm $A^{(\cdot)}$ for the Unitary Synthesis Problem.

Our second main result

Main result #2: Relative to a random oracle R, there exists a PRS secure against any efficient oracle adversary $A^{(\cdot)}$ that makes one query to an arbitrary function f_R, which can depend on R.

Immediate corollary

Why? For any PRS, there exists **some** unitary U that can break it.

(e.g., if PRS states $\{|\psi_k\rangle\}_{k\in[K]}$ are orthogonal, the unitary that maps $|\psi_k\rangle \to |k\rangle$ would break the PRS.)

Main result #1: There is no efficient one-query oracle algorithm $A^{(\cdot)}$ for the Unitary Synthesis Problem.

Plan for this talk

- Relating unitary synthesis to breaking quantum cryptography
- **The Oracle State Distinguishing Game**
- If time: our proof for a special class of adversaries

Our approach: analyze the task of breaking an n-qubit PRS where the $K \ll N \coloneqq 2^n$ states are random binary phase states.

(for rest of this talk, $N \coloneqq 2^n$ and $[N] = \{0,1\}^n$)

Our approach: analyze the task of breaking an n-qubit PRS where the $K \ll N := 2^n$ states are random binary phase states.

(for rest of this talk, $N := 2^n$ and $[N] = \{0,1\}^n$)

Definitions

- For any function $h\colon [N] \to \{\pm 1\}$, its **binary phase state** is

$$|\psi_h\rangle := \frac{1}{\sqrt{N}} \sum_{x \in [N]} h(x) \cdot |x\rangle .$$

Our approach: analyze the task of breaking an n-qubit PRS where the $K \ll N := 2^n$ states are random binary phase states.

(for rest of this talk, $N := 2^n$ and $[N] = \{0,1\}^n$)

Definitions

- For any function $h\colon [N] \to \{\pm 1\}$, its **binary phase state** is

$$|\psi_h\rangle := \frac{1}{\sqrt{N}} \sum_{x \in [N]} h(x) \cdot |x\rangle .$$

- A **function family** $R\colon [K] \times [N] \to \{\pm 1\}$ specifies K binary phase states $\{|\psi_{R_k}\rangle\}_{k \in [K]}$ where $R_k(x) := R(k, x)$.

The Oracle State Distinguishing Game

The Oracle State Distinguishing Game

Setup: Sample random function family $R: [K] \times [N] \to \{\pm 1\}$

The Oracle State Distinguishing Game

Setup: Sample random function family $R: [K] \times [N] \to \{\pm 1\}$

1-query adversaries: Adversary makes one query to an arbitrary function $f_R: \{0,1\}^{\ell} \to \{\pm 1\}$ that depends on R.

The Oracle State Distinguishing Game

Setup: Sample random function family $R\colon [K]\times[N] \to \{\pm 1\}$

1-query adversaries: Adversary makes one query to an arbitrary function $f_R\colon \{0,1\}^{\ell} \to \{\pm 1\}$ that depends on R.

Our lower bound holds up to $\ell = o(2^n)$.

The Oracle State Distinguishing Game

Setup: Sample random function family $R\colon [K]\times[N] \to \{\pm 1\}$

1-query adversaries: Adversary makes one query to an arbitrary function $f_R\colon \{0,1\}^{\ell} \to \{\pm 1\}$ that depends on R.

Game: Sample $b \leftarrow \{0,1\}$.

The Oracle State Distinguishing Game

Setup: Sample random function family $R\colon [K]\times[N] \to \{\pm 1\}$

1-query adversaries: Adversary makes one query to an arbitrary function $f_R\colon \{0,1\}^{\ell} \to \{\pm 1\}$ that depends on R.

Game: Sample $b \leftarrow \{0,1\}$. Generate n-qubit state $|\psi\rangle$ as follows:

The Oracle State Distinguishing Game

Setup: Sample random function family $R: [K] \times [N] \to \{\pm 1\}$

1-query adversaries: Adversary makes one query to an arbitrary function $f_R: \{0,1\}^{\ell} \to \{\pm 1\}$ that depends on R.

Game: Sample $b \leftarrow \{0,1\}$. Generate n-qubit state $|\psi\rangle$ as follows:

- If $b = 0$, $|\psi\rangle = |\psi_{R_k}\rangle$ for random $k \leftarrow [K]$.

The Oracle State Distinguishing Game

Setup: Sample random function family $R\colon [K]\times[N] \to \{\pm 1\}$

1-query adversaries: Adversary makes one query to an arbitrary function $f_R\colon \{0,1\}^{\ell} \to \{\pm 1\}$ that depends on R.

Game: Sample $b \leftarrow \{0,1\}$. Generate n-qubit state $|\psi\rangle$ as follows:

- If $b = 0$, $|\psi\rangle = |\psi_{R_k}\rangle$ for random $k \leftarrow [K]$.
- If $b = 1$, $|\psi\rangle = |\psi_h\rangle$ for random $h\colon [N] \to \{\pm 1\}$.

The Oracle State Distinguishing Game

Setup: Sample random function family $R\colon [K]\times[N] \to \{\pm 1\}$

1-query adversaries: Adversary makes one query to an arbitrary function $f_R\colon \{0,1\}^\ell \to \{\pm 1\}$ that depends on R.

Game: Sample $b \leftarrow \{0,1\}$. Generate n-qubit state $|\psi\rangle$ as follows:

- If $b = 0$, $|\psi\rangle = |\psi_{R_k}\rangle$ for random $k \leftarrow [K]$.
- If $b = 1$, $|\psi\rangle = |\psi_h\rangle$ for random $h\colon [N] \to \{\pm 1\}$.

Send $|\psi\rangle$ to the adversary.

The Oracle State Distinguishing Game

Setup: Sample random function family $R\colon [K]\times[N] \to \{\pm 1\}$

1-query adversaries: Adversary makes one query to an arbitrary function $f_R\colon \{0,1\}^\ell \to \{\pm 1\}$ that depends on R.

Game: Sample $b \leftarrow \{0,1\}$. Generate n-qubit state $|\psi\rangle$ as follows:

- If $b = 0$, $|\psi\rangle = |\psi_{R_k}\rangle$ for random $k \leftarrow [K]$.
- If $b = 1$, $|\psi\rangle = |\psi_h\rangle$ for random $h\colon [N] \to \{\pm 1\}$.

Send $|\psi\rangle$ to the adversary. Adversary outputs a guess b'.

The Oracle State Distinguishing Game

Setup: Sample random function family $R\colon [K]\times[N] \to \{\pm 1\}$

1-query adversaries: Adversary makes one query to an arbitrary function $f_R\colon \{0,1\}^{\ell} \to \{\pm 1\}$ that depends on R.

Game: Sample $b \leftarrow \{0,1\}$. Generate n-qubit state $|\psi\rangle$ as follows:

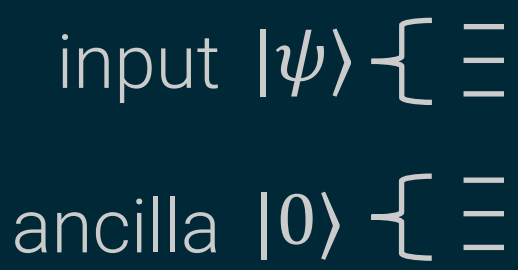
- If $b = 0$, $|\psi\rangle = |\psi_{R_k}\rangle$ for random $k \leftarrow [K]$.
- If $b = 1$, $|\psi\rangle = |\psi_h\rangle$ for random $h\colon [N] \to \{\pm 1\}$.

Send $|\psi\rangle$ to the adversary. Adversary outputs a guess b'.

Adversary wins $b = b'$.

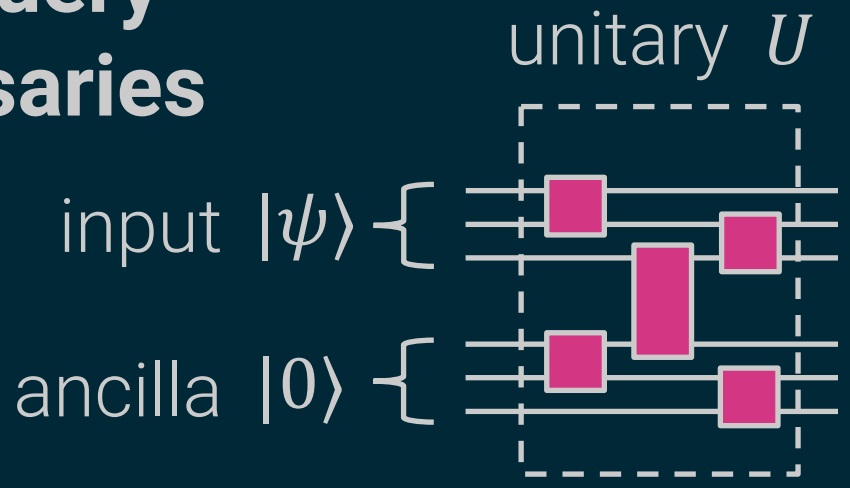
One-query adversaries

One-query adversaries



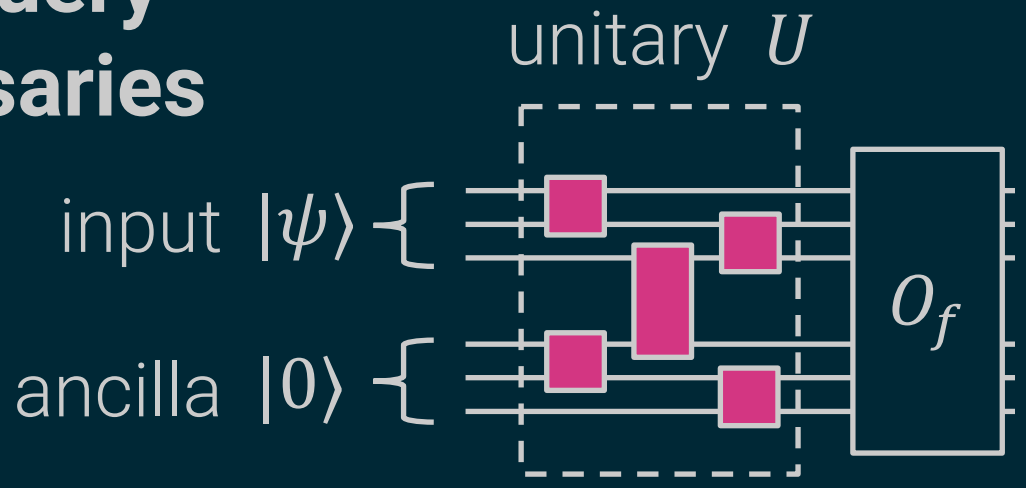
1) Initialize $\ell - n$ ancilla qubits

One-query adversaries



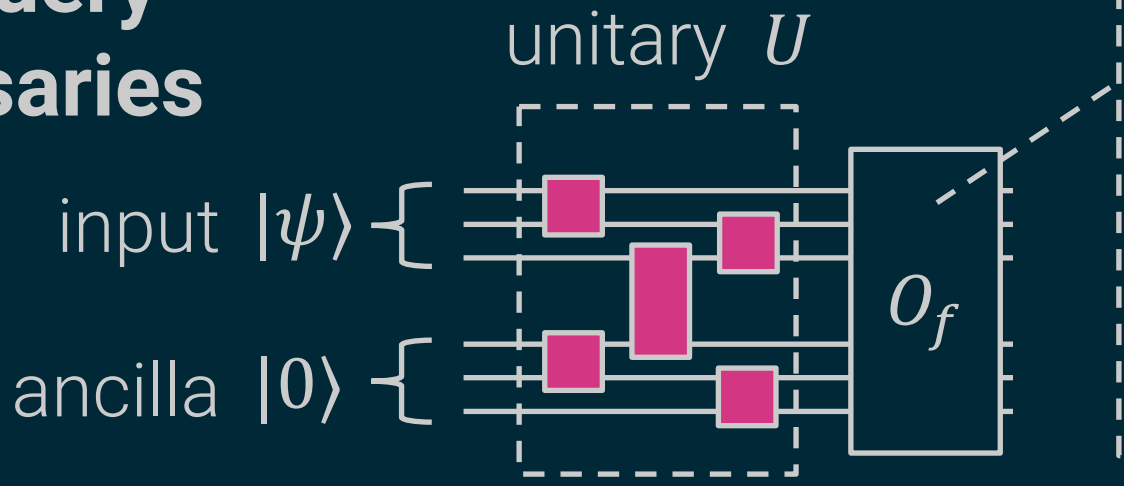
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.

One-query adversaries



1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.

One-query adversaries

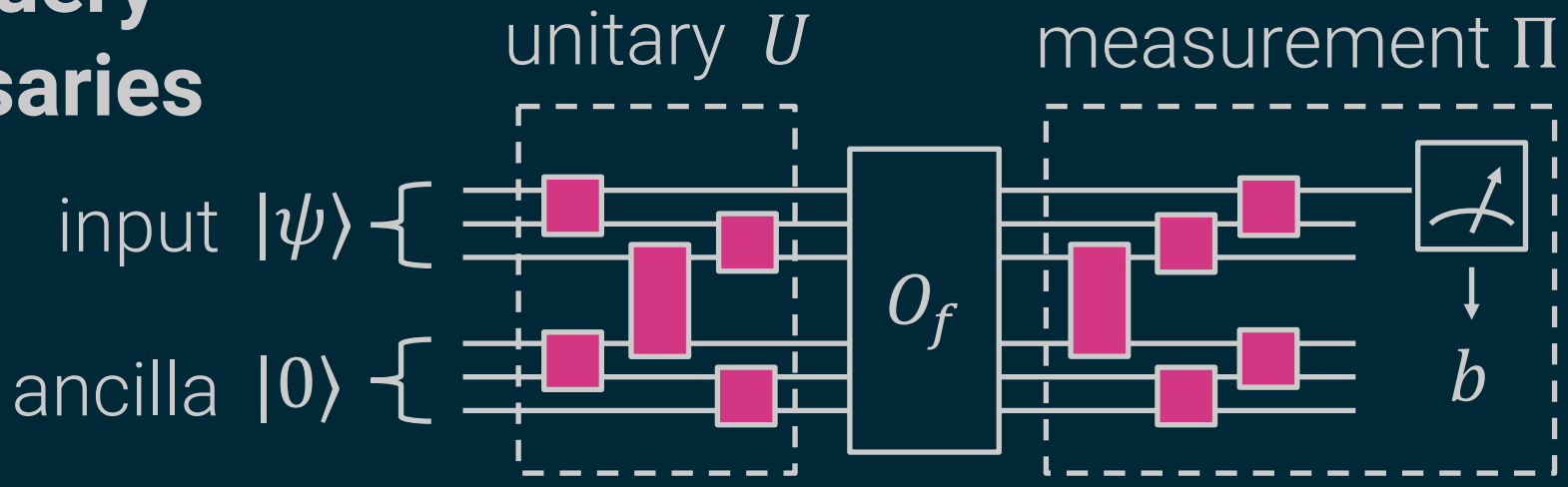


$$O_f = \begin{pmatrix} \ddots & & \\ & f(z) & \\ & & \ddots \end{pmatrix}$$

$2^\ell \times 2^\ell$ diagonal matrix, z-th entry is $f(z) \in \{\pm 1\}$

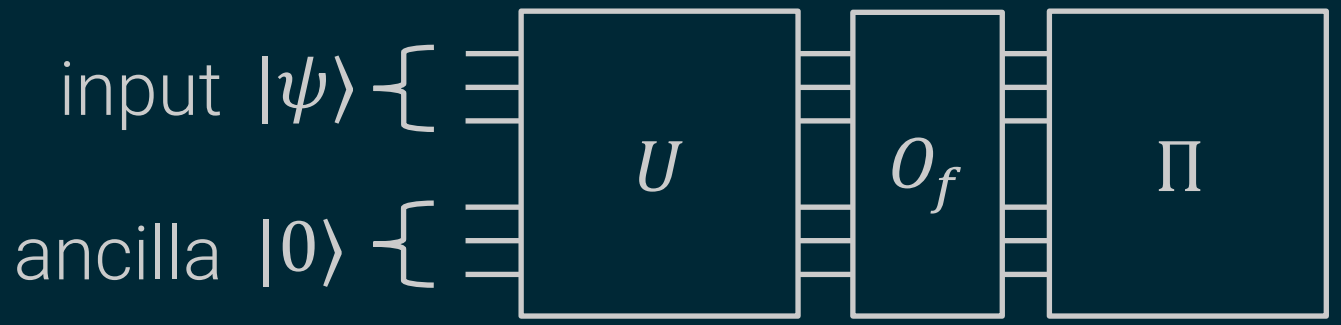
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.

One-query adversaries



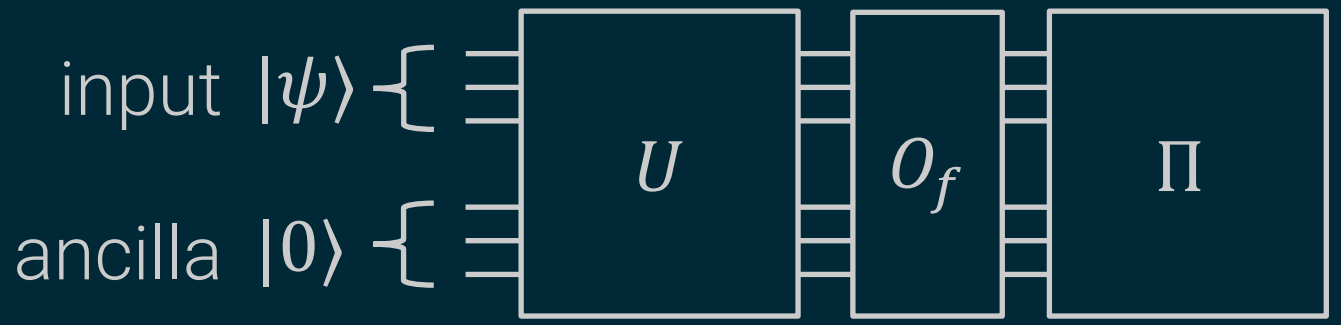
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.
4) Measure $\{\Pi, \mathrm{I} - \Pi\}$ and return 1 if outcome is Π.

One-query adversaries



1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.
4) Measure $\{\Pi, \mathrm{I} - \Pi\}$ and return 1 if outcome is Π.

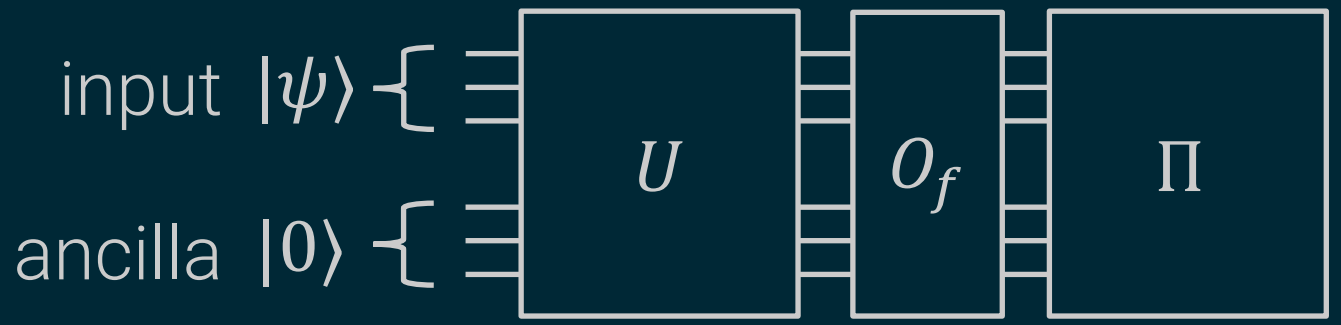
One-query adversaries



$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot U \cdot |\psi\rangle|0\rangle \right\|^2$$

1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.
4) Measure $\{\Pi, \mathrm{I} - \Pi\}$ and return 1 if outcome is Π.

One-query adversaries

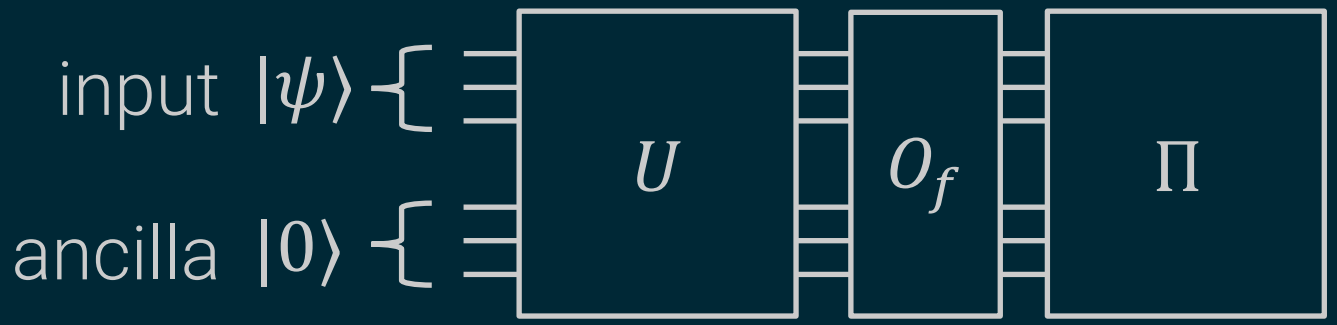


$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot U \cdot |\psi\rangle|0\rangle \right\|^2$$

For fixed R, the adversary's **distinguishing advantage** is:

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] - \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

One-query adversaries



$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot U \cdot |\psi\rangle|0\rangle \right\|^2$$

For fixed R, the adversary's **distinguishing advantage** is:

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] - \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

(adversary picks $f = f_R$ to maximize this)

This optimization problem is very subtle!

For fixed R, the adversary's **distinguishing advantage** is:

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] - \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

(adversary picks $f = f_R$ to maximize this)

This optimization problem is very subtle!

We show:

- Carefully-chosen **spectral relaxation** gives an upper bound in terms of the operator norm of a certain random matrix.

For fixed R, the adversary's **distinguishing advantage** is:

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] - \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

(adversary picks $f = f_R$ to maximize this)

This optimization problem is very subtle!

We show:

- Carefully-chosen **spectral relaxation** gives an upper bound in terms of the operator norm of a certain random matrix.
- We bound this norm by appealing to **matrix concentration**.

For fixed R, the adversary's **distinguishing advantage** is:

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] - \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

(adversary picks $f = f_R$ to maximize this)

Plan for this talk

- Relating unitary synthesis to breaking quantum cryptography
- The Oracle State Distinguishing Game
- **If time: our proof for a special class of adversaries**

A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

A special class of one-query adversaries

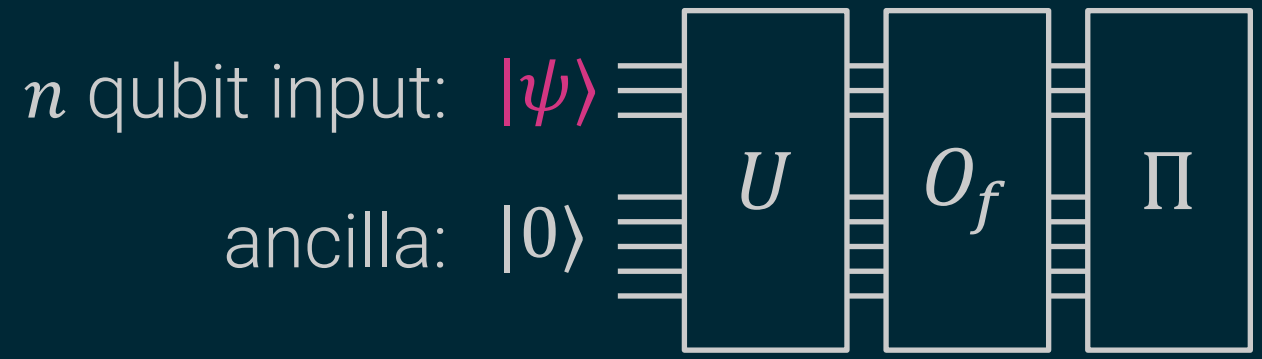
Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

Disclaimer: We can rule out these attacks with a counting argument, but today we'll see a different proof.

A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

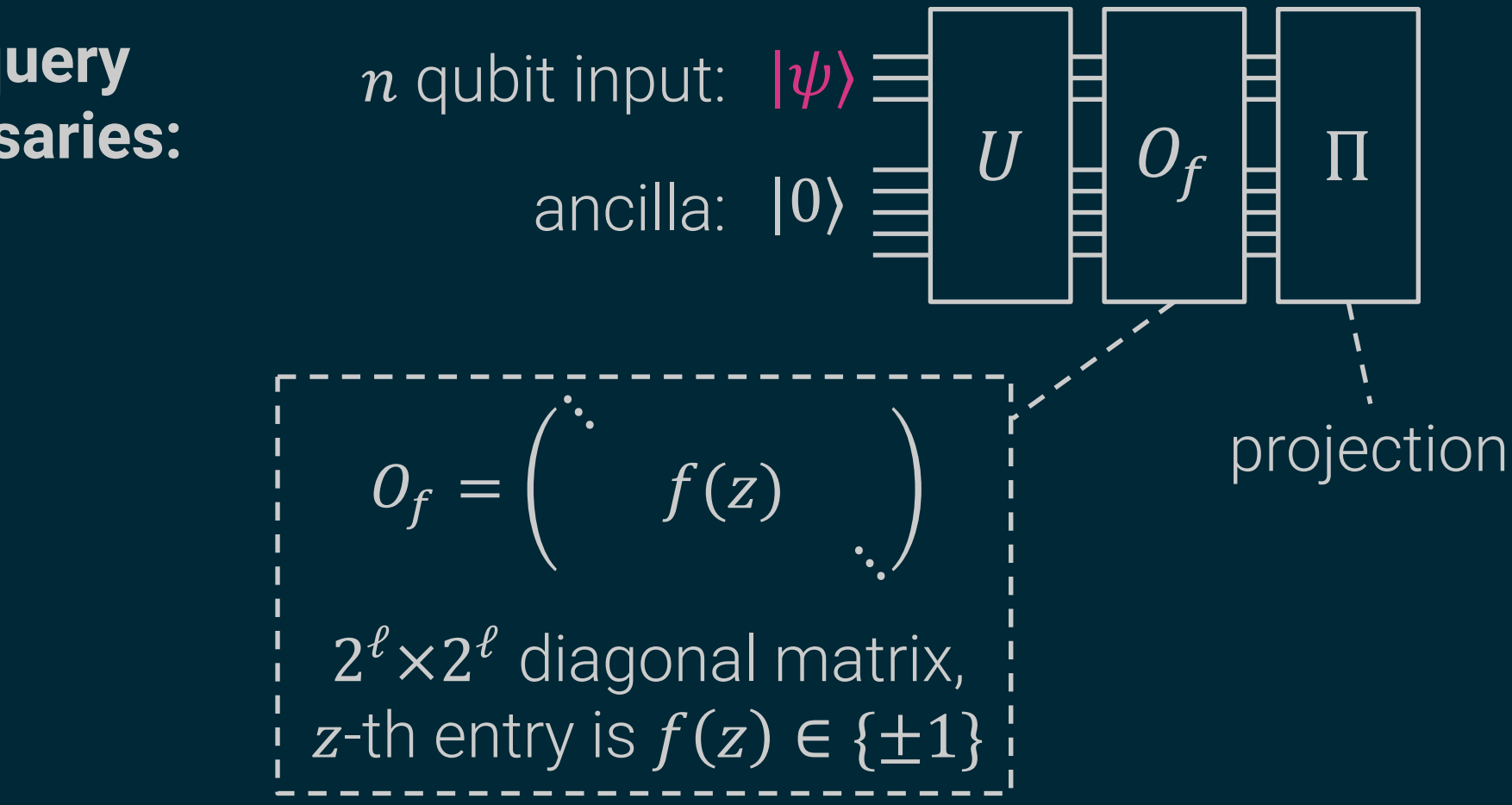
One-query adversaries:



A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

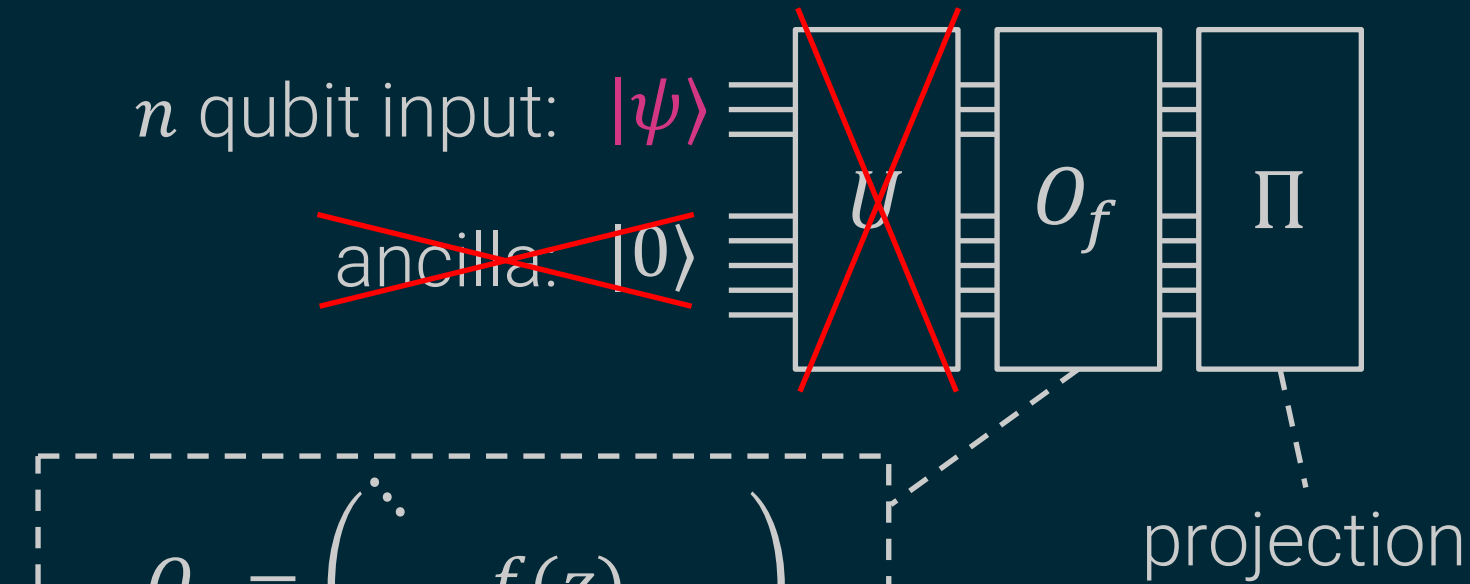
One-query adversaries:



A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

One-query adversaries:

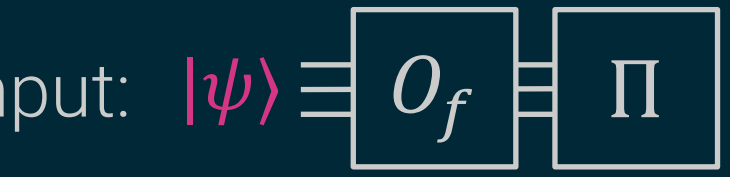


$$O_f = \begin{pmatrix} \ddots & & \\ & f(z) & \\ & & \ddots \end{pmatrix}$$

$2^\ell \times 2^\ell$ diagonal matrix,
z-th entry is $f(z) \in \{\pm 1\}$

A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

Special class: n qubit input: 

A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

Special class: n qubit input: $|\psi\rangle$ 

$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot |\psi\rangle \right\|^2$$

A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

Special class: n qubit input: $|\psi\rangle$ 

$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot |\psi\rangle \right\|^2$$

For these adversaries, the **distinguishing advantage** is:

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \langle \psi_{R_k} | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_{R_k}\rangle - \mathop{\mathbb{E}}_{h} \langle \psi_h | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle$$

(adversary picks $f = f_R$ to maximize this)

Technical tool: matrix concentration

Technical tool: matrix concentration

Scalar Chernoff bound: If X is a random **scalar** with bounded absolute value, then for i.i.d. $X_1, \dots X_K$

$$\left| \frac{1}{K} \sum_k X_k - \mathbb{E}[X] \right| \approx O\left(\frac{1}{\sqrt{K}}\right) \qquad \text{(w.h.p.)}$$

Technical tool: matrix concentration

Scalar Chernoff bound: If X is a random **scalar** with bounded absolute value, then for i.i.d. $X_1, \dots X_K$

$$\left|\frac{1}{K}\sum_k X_k - \mathbb{E}[X]\right| \approx O\left(\frac{1}{\sqrt{K}}\right) \qquad \text{(w.h.p.)}$$

Matrix Chernoff bound: If X is a random $L \times L$ **matrix** with bounded operator norm, then for i.i.d. $X_1, \dots, X_K$

$$\left\|\frac{1}{K}\sum_k X_k - \mathbb{E}[X]\right\|_{\text{op}} \approx O\left(\frac{\sqrt{\log(L)}}{\sqrt{K}}\right) \qquad \text{(w.h.p.)}$$

Adversary's advantage (for this special class):

$$\max_{f:[\mathrm{N}]\to\{\pm 1\}} \left| \frac{1}{K} \sum_{k} \langle \psi_{R_k} | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_{R_k}\rangle - \mathbb{E}_h \langle \psi_h | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle \right|$$

random vectors

Adversary's advantage (for this special class):

$$\max_{f:[\mathrm{N}]\to\{\pm 1\}} \left| \frac{1}{K} \sum_k \langle \psi_{R_k} | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_{R_k}\rangle - \mathbb{E}_h \langle \psi_h | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle \right|$$

random vectors

Matrix Chernoff:

$$\max_{|v\rangle} \left| \langle v | \cdot \left(\frac{1}{K} \sum_k X_k - \mathbb{E}[X] \right) \cdot |v\rangle \right|$$

random matrices

Adversary's advantage (for this special class):

$$\max_{f:[N]\to\{\pm 1\}} \left| \frac{1}{K} \sum_k \langle \psi_{R_k} | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_{R_k}\rangle - \mathbb{E}_h \langle \psi_h | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle \right|$$

max over matrices

random vectors

Matrix Chernoff:

$$\max_{|v\rangle} \left| \langle v | \cdot \left(\frac{1}{K} \sum_k X_k - \mathbb{E}[X] \right) \cdot |v\rangle \right|$$

random matrices

max over unit vectors

Adversary's advantage (for this special class):

$$\max_{f:[N]\to\{\pm 1\}} \left| \frac{1}{K} \sum_k \langle \psi_{R_k} | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_{R_k}\rangle - \mathbb{E}_h \langle \psi_h | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle \right|$$

max over matrices

random vectors

?

Matrix Chernoff:

$$\max_{|v\rangle} \left| \langle v | \cdot \left(\frac{1}{K} \sum_k X_k - \mathbb{E}[X] \right) \cdot |v\rangle \right|$$

random matrices

max over unit vectors

Adversary's advantage (for this special class):

$$\max_{f:[N]\to\{\pm 1\}} \left| \frac{1}{K} \sum_{k} \langle \psi_{R_k} | \cdot O_f \cdot \Pi \cdot O_f \cdot | \psi_{R_k} \rangle - \mathbb{E}_h \langle \psi_h | \cdot O_f \cdot \Pi \cdot O_f \cdot | \psi_h \rangle \right|$$

Key step: We'll refactor this as $\langle v_f | \cdot (\text{random matrix}) \cdot | v_f \rangle$

Adversary's advantage (for this special class):

$$\max_{f:[N]\to\{\pm 1\}} \left| \frac{1}{K} \sum_{k} \langle \psi_{R_k} | \cdot O_f \cdot \Pi \cdot O_f \cdot | \psi_{R_k} \rangle - \mathbb{E}_h \langle \psi_h | \cdot O_f \cdot \Pi \cdot O_f \cdot | \psi_h \rangle \right|$$

Key step: We'll refactor this as $\langle v_f | \cdot (\text{random matrix}) \cdot | v_f \rangle$

R_k-dependent random matrix with bounded norm

f-dependent unit vector

Adversary's advantage (for this special class):

$$\max_{f:[N]\to\{\pm 1\}} \left| \frac{1}{K} \sum_k \langle \psi_{R_k} | \cdot O_f \cdot \Pi \cdot O_f \cdot | \psi_{R_k} \rangle - \mathbb{E}_h \langle \psi_h | \cdot O_f \cdot \Pi \cdot O_f \cdot | \psi_h \rangle \right|$$

Key step: We'll refactor this as $\langle v_f | \cdot (\text{random matrix}) \cdot | v_f \rangle$

R_k-dependent random matrix with bounded norm

f-dependent unit vector

Let's see how!

We'll rewrite this as $\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle$

$\langle \psi_{R_k}| \; O_f \cdot \Pi \cdot O_f \; |\psi_{R_k}\rangle$

We'll rewrite this as $\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle$

$\langle \psi_{R_k}| \; O_f \cdot \Pi \cdot O_f \; |\psi_{R_k}\rangle$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix} \cdot \frac{1}{\sqrt{N}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

$N \times N$ diagonal matrix, x-th entry is $R_k(x)$

uniform superposition

We'll rewrite this as $\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle$

$$\langle \psi_{R_k}| \; O_f \cdot \Pi \cdot O_f \; |\psi_{R_k}\rangle$$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \underbrace{\begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix}}_{:= D_{R_k}} \cdot \underbrace{\frac{1}{\sqrt{N}}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{:= |+_N\rangle}$$

We'll rewrite this as $\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle$

$$\langle \psi_{R_k}|\ O_f \cdot \Pi \cdot O_f\ |\psi_{R_k}\rangle = \langle +_N|\ D_{R_k} \cdot O_f \cdot \Pi \cdot O_f \cdot D_{R_k}\ |+_N\rangle \qquad \textbf{(1)}$$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \underbrace{\begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix}}_{:= D_{R_k}} \cdot \underbrace{\frac{1}{\sqrt{N}}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{:= |+_N\rangle}$$

We'll rewrite this as $\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle$

$$\langle \psi_{R_k}|\ O_f \cdot \Pi \cdot O_f\ |\psi_{R_k}\rangle = \langle +_N|\ D_{R_k} \cdot O_f \cdot \Pi \cdot O_f \cdot D_{R_k}\ |+_N\rangle \qquad \textbf{(1)}$$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \underbrace{\begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix}}_{:= D_{R_k}} \cdot \underbrace{\frac{1}{\sqrt{N}}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{:= |+_N\rangle}$$

(2) O_f is a diagonal matrix, so it ***commutes*** with D_{R_k}

We'll rewrite this as $\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle$

$$\langle \psi_{R_k}|\ O_f \cdot \Pi \cdot O_f\ |\psi_{R_k}\rangle = \langle +_N|\ D_{R_k} \cdot O_f \cdot \Pi \cdot O_f \cdot D_{R_k}\ |+_N\rangle \quad \textbf{(1)}$$

$$= \langle +_N|\ O_f \cdot D_{R_k} \cdot \Pi \cdot D_{R_k} \cdot O_f |+_N\rangle \quad \textbf{(2)}$$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \underbrace{\begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix}}_{:= D_{R_k}} \cdot \underbrace{\frac{1}{\sqrt{N}}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{:= |+_N\rangle}$$

(2) O_f is a diagonal matrix, so it ***commutes*** with D_{R_k}

With this refactoring, we can rewrite the adversary's advantage:

$$\left| \frac{1}{K} \sum_k \langle \psi_{R_k} | \, O_f \cdot \Pi \cdot O_f \, | \psi_{R_k} \rangle - \mathbb{E}_h \langle \psi_h | \, O_f \cdot \Pi \cdot O_f \, | \psi_h \rangle \right|$$

With this refactoring, we can rewrite the adversary's advantage:

$$\left| \frac{1}{K} \sum_k \langle \psi_{R_k} | \, O_f \cdot \Pi \cdot O_f \, | \psi_{R_k} \rangle - \mathbb{E}_h \langle \psi_h | \, O_f \cdot \Pi \cdot O_f \, | \psi_h \rangle \right|$$

$$= \langle +_N | \, O_f \cdot \left(\frac{1}{K} \sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h [D_h \cdot \Pi \cdot D_h] \right) \cdot O_f \, | +_N \rangle$$

With this refactoring, we can rewrite the adversary's advantage:

$$\left| \frac{1}{K} \sum_k \langle \psi_{R_k} | \, O_f \cdot \Pi \cdot O_f \, | \psi_{R_k} \rangle - \mathbb{E}_h \langle \psi_h | \, O_f \cdot \Pi \cdot O_f \, | \psi_h \rangle \right|$$

$$= \langle +_N | \, O_f \cdot \left(\frac{1}{K} \sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h [D_h \cdot \Pi \cdot D_h] \right) \cdot \underbrace{O_f \, | +_N \rangle}_{\text{unit vector}}$$

With this refactoring, we can rewrite the adversary's advantage:

$$\left| \frac{1}{K} \sum_k \langle \psi_{R_k} | \, O_f \cdot \Pi \cdot O_f \, | \psi_{R_k} \rangle - \mathbb{E}_h \langle \psi_h | \, O_f \cdot \Pi \cdot O_f \, | \psi_h \rangle \right|$$

$$= \langle +_N | \, O_f \cdot \left(\frac{1}{K} \sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h [D_h \cdot \Pi \cdot D_h] \right) \cdot \underbrace{O_f \, | +_N \rangle}_{\text{unit vector}}$$

$$\leq \left\| \frac{1}{K} \sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h [D_h \cdot \Pi \cdot D_h] \right\|_{\text{op}}$$

With this refactoring, we can rewrite the adversary's advantage:

$$\left| \frac{1}{K} \sum_k \langle \psi_{R_k} | \, O_f \cdot \Pi \cdot O_f \, | \psi_{R_k} \rangle - \mathbb{E}_h \langle \psi_h | \, O_f \cdot \Pi \cdot O_f \, | \psi_h \rangle \right|$$

$$= \langle +_N | \, O_f \cdot \left(\frac{1}{K} \sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h [D_h \cdot \Pi \cdot D_h] \right) \cdot \underbrace{O_f \, | +_N \rangle}_{\text{unit vector}}$$

$$\leq \left\| \frac{1}{K} \sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h [D_h \cdot \Pi \cdot D_h] \right\|_{\text{op}}$$

Matrix Chernoff: For i.i.d. bounded random $L \times L$ matrices X_k:

$$\left\| \frac{1}{K} \sum_k X_k - \mathbb{E}[X] \right\|_{\text{op}} \approx O\left(\frac{\sqrt{\log(L)}}{\sqrt{K}} \right) \quad \text{(w.h.p.).}$$

With this refactoring, we can rewrite the adversary's advantage:

$$\left|\frac{1}{K}\sum_k \langle\psi_{R_k}|\, O_f \cdot \Pi \cdot O_f\, |\psi_{R_k}\rangle - \mathbb{E}_h \langle\psi_h|\, O_f \cdot \Pi \cdot O_f\, |\psi_h\rangle\right|$$

$$= \langle +_N|\, O_f \cdot \left(\frac{1}{K}\sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h[D_h \cdot \Pi \cdot D_h]\right) \cdot \underbrace{O_f\, |+_N\rangle}_{\text{unit vector}}$$

$$\leq \left\|\frac{1}{K}\sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h[D_h \cdot \Pi \cdot D_h]\right\|_{\text{op}}$$

Matrix Chernoff: For i.i.d. bounded random $L \times L$ matrices X_k:

$$\left\|\frac{1}{K}\sum_k X_k - \mathbb{E}[X]\right\|_{\text{op}} \approx O\left(\frac{\sqrt{\log(L)}}{\sqrt{K}}\right) \quad \text{(w.h.p.).}$$

We'll set $X_k = D_{R_k} \cdot \Pi \cdot D_{R_k}$. This matrix has norm 1 since Π is a projector and D_{R_k} is a unitary.

With this refactoring, we can rewrite the adversary's advantage:

$$\left|\frac{1}{K}\sum_k \langle\psi_{R_k}|\, O_f \cdot \Pi \cdot O_f\, |\psi_{R_k}\rangle - \mathbb{E}_h \langle\psi_h|\, O_f \cdot \Pi \cdot O_f\, |\psi_h\rangle\right|$$

$$= \langle +_N|\, O_f \cdot \left(\frac{1}{K}\sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h[D_h \cdot \Pi \cdot D_h]\right) \cdot O_f\, |+_N\rangle$$

$$\leq \left\|\frac{1}{K}\sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h[D_h \cdot \Pi \cdot D_h]\right\|_{\text{op}} \approx O\left(\sqrt{\frac{n}{K}}\right)$$ **Final bound**

Matrix Chernoff: For i.i.d. bounded random $L \times L$ matrices X_k:

$$\left\|\frac{1}{K}\sum_k X_k - \mathbb{E}[X]\right\|_{\text{op}} \approx O\left(\frac{\sqrt{\log(L)}}{\sqrt{K}}\right) \quad \text{(w.h.p.).}$$

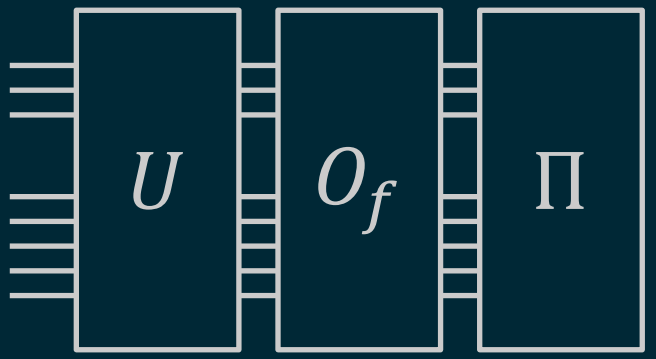
We'll set $X_k = D_{R_k} \cdot \Pi \cdot D_{R_k}$. This matrix has norm 1 since Π is a projector and D_{R_k} is a unitary.

Extending this proof to general one-query adversaries requires more care.

General one-query adversaries

n qubit input: $|\psi_h\rangle$

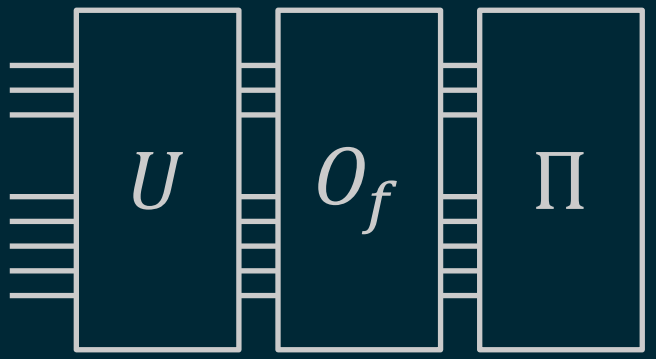
ancilla: $|0\rangle$



General one-query adversaries

n qubit input: $|\psi_h\rangle$

ancilla: $|0\rangle$

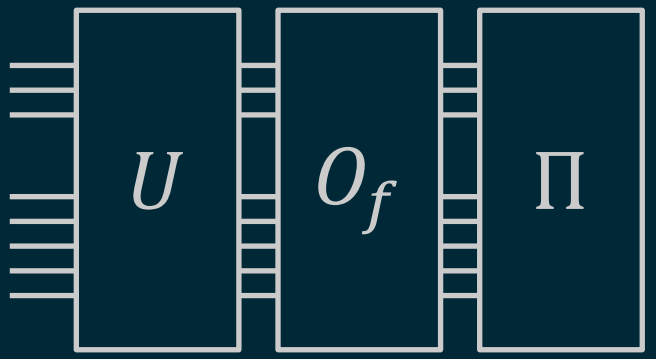


Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. “add ancillas + apply U”

General one-query adversaries

n qubit input: $|\psi_h\rangle$

ancilla: $|0\rangle$



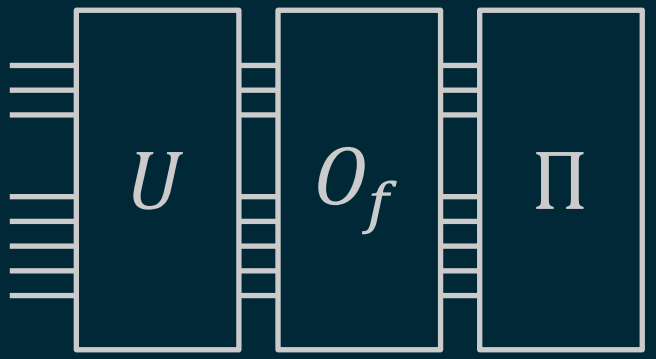
Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. "add ancillas + apply U"

$$\Pr[A^f(|\psi_h\rangle) \text{ outputs } 1] = \langle +_N| D_h \cdot V^\dagger \cdot O_f \cdot \Pi \cdot O_f \cdot V \cdot \mathrm{D}_h |+_N\rangle$$

General one-query adversaries

n qubit input: $|\psi_h\rangle$

ancilla: $|0\rangle$



Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. "add ancillas + apply U"

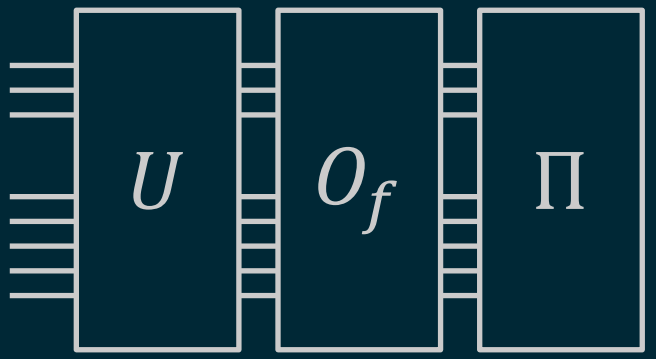
$$\Pr[A^f(|\psi_h\rangle) \text{ outputs } 1] = \langle +_N | \underbrace{D_h \cdot V^\dagger \cdot O_f}_{} \cdot \Pi \cdot \underbrace{O_f \cdot V \cdot \mathrm{D}_h}_{} | +_N \rangle$$

Challenge: unclear how to commute D_h and O_f!

General one-query adversaries

n qubit input: $|\psi_h\rangle$

ancilla: $|0\rangle$



Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. "add ancillas + apply U"

$$\Pr[A^f(|\psi_h\rangle) \text{ outputs } 1] = \langle +_N | \underbrace{D_h \cdot V^\dagger \cdot O_f}_{} \cdot \Pi \cdot \underbrace{O_f \cdot V \cdot \mathrm{D}_h}_{} |+_N\rangle$$

Challenge: unclear how to commute D_h and O_f!

Idea: obtain spectral relaxation by factoring $V|\psi_h\rangle = \widetilde{D_h} \cdot |\mathrm{wt}_V\rangle$ w.r.t. a V-dependent unit vector $|\mathrm{wt}_V\rangle$.

Conclusions

- Implementing unitaries and breaking quantum crypto might be harder than solving any classical problem.

Conclusions

- Implementing unitaries and breaking quantum crypto might be harder than solving any classical problem.
- Hardness comes from **quantum input** (breaking PRS is a **classical output** task).

Conclusions

- Implementing unitaries and breaking quantum crypto might be harder than solving any classical problem.
- Hardness comes from **quantum input** (breaking PRS is a **classical output** task).
- Next steps: extend to $\mathbf{poly}(n)$ queries?

Conclusions

- Implementing unitaries and breaking quantum crypto might be harder than solving any classical problem.
- Hardness comes from **quantum input** (breaking PRS is a **classical output** task).
- Next steps: extend to $\mathbf{poly}(n)$ queries?

Strong non-synthesis conjecture:
The Oracle State Distinguishing Game is hard for any efficient oracle adversary A^f that makes $\mathbf{poly}(n)$ queries to f.

Conclusions

- Implementing unitaries and breaking quantum crypto might be harder than solving any classical problem.
- Hardness comes from **quantum input** (breaking PRS is a **classical output** task).
- Next steps: extend to $\mathbf{poly}(n)$ queries?

Strong non-synthesis conjecture:
The Oracle State Distinguishing Game is hard for any efficient oracle adversary A^f that makes $\mathbf{poly}(n)$ queries to f.

Challenge: hard to find the right spectral relaxation past one query.

Conclusions

- Implementing unitaries and breaking quantum crypto might be harder than solving any classical problem.
- Hardness comes from **quantum input** (breaking PRS is a **classical output** task).
- Next steps: extend to $\mathbf{poly}(n)$ queries?

Strong non-synthesis conjecture:
The Oracle State Distinguishing Game is hard for any efficient oracle adversary A^f that makes $\mathbf{poly}(n)$ queries to f.

Challenge: hard to find the right spectral relaxation past one query.

Thanks for listening!