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/outputs.

In complexity theory, problems have **classical** inputs/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/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/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/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).

Even though this problem is "about" quantum states, the input and output are classical.

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

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

- **State tomography:** output classical description of $|\psi\rangle$ given many copies of $|\psi\rangle$.

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

- **State tomography:** output classical description of $|\psi\rangle$ given many copies of $|\psi\rangle$.
- **Quantum error correction:** decode a noisy quantum error-correcting codeword $|c\rangle$.

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

- **State tomography:** output classical description of $|\psi\rangle$ given many copies of $|\psi\rangle$.
- **Quantum error correction:** decode a noisy quantum error-correcting codeword $|c\rangle$.
- **State distinguishing:** distinguish two mixtures of quantum states ρ_0, ρ_1, given one of them at random.

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

- **State tomography:** output classical description of $|\psi\rangle$ given many copies of $|\psi\rangle$.
- **Quantum error correction:** decode a noisy quantum error-correcting codeword $|c\rangle$.
- **State distinguishing:** distinguish two mixtures of quantum states ρ_0, ρ_1, given one of them at random.

Physics: “decoding” black-hole radiation, computing AdS/CFT map

What can complexity theory say about these inherently quantum problems?

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

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

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

Ex: is the problem easy given an oracle for NP? PSPACE?

But for some quantum problems, it's not clear if this can be done.

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

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

But for some quantum problems, it's not clear if this can be done.

State distinguishing: distinguish two mixtures of quantum states ρ_0, ρ_1, given one of them at random.

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

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

But for some quantum problems, it's not clear if this can be done.

State distinguishing: distinguish two mixtures of quantum states ρ_0, ρ_1, given one of them at random.

Not known how to solve this using ***any*** oracle

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

Ex: is the problem easy given an oracle for NP? PSPACE?

But for some quantum problems, it's not clear if this can be done.

State distinguishing: distinguish two mixtures of quantum states ρ_0, ρ_1, given one of them at random.

Not known how to solve this using ***any*** oracle, even an oracle for the halting problem!

Before we continue:

1-minute detour for quantum computing 101

Quantum Computing 101

Quantum Computing 101

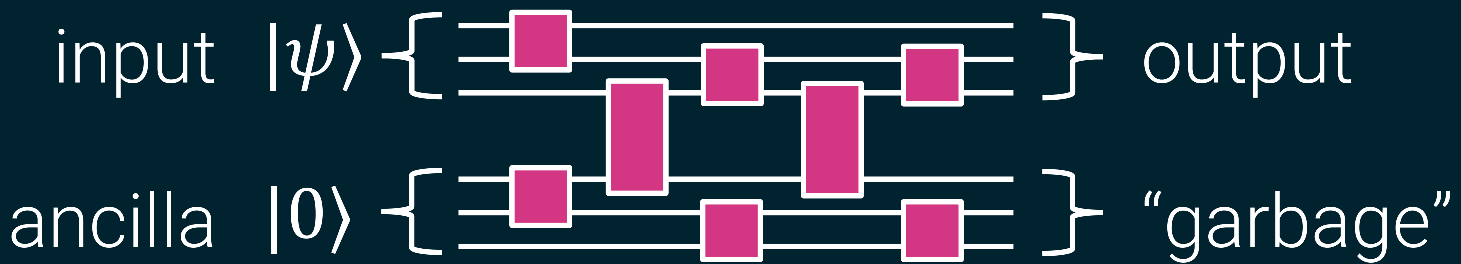
- n-qubit pure state = 2^n-dim unit vector $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.

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.

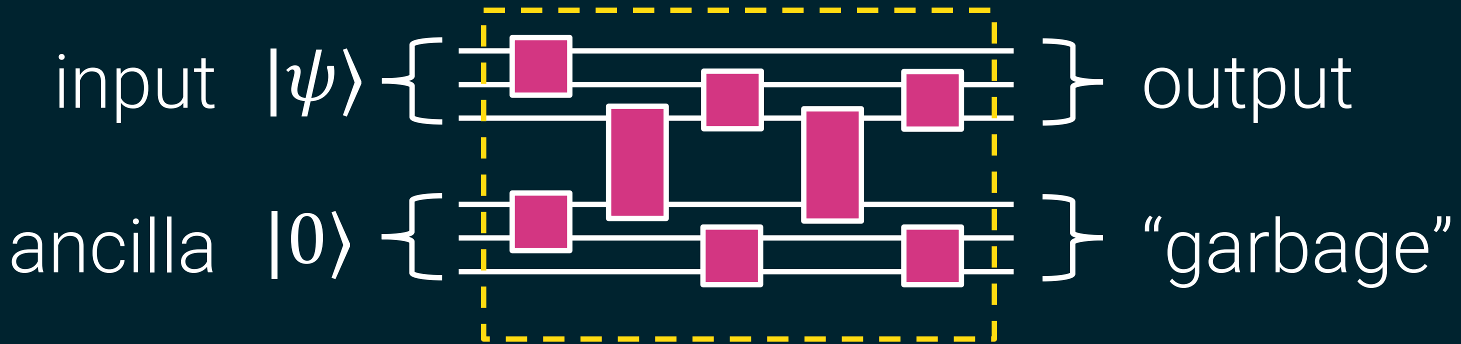
Quantum Computing 101

- 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:

Does complexity theory capture quantum problems?

Does complexity theory capture quantum problems?

- solving a quantum problem means implementing a **unitary**.

Does complexity theory capture quantum problems?

- solving a quantum problem means implementing a **unitary**.
- complexity theory is about implementing **functions**.

Does complexity theory capture quantum problems?

- solving a quantum problem means implementing a **unitary**.
- complexity theory is about implementing **functions**.

To apply complexity theory, we need to ***efficiently reduce*** the task of implementing a unitary U to implementing a function f.

Does complexity theory capture quantum problems?

- solving a quantum problem means implementing a **unitary**.
- complexity theory is about implementing **functions**.

To apply complexity theory, we need to ***efficiently reduce*** the task of implementing a unitary U to implementing a function f.

The Unitary Synthesis Problem [AK06]:

Is there a reduction that works for every U?

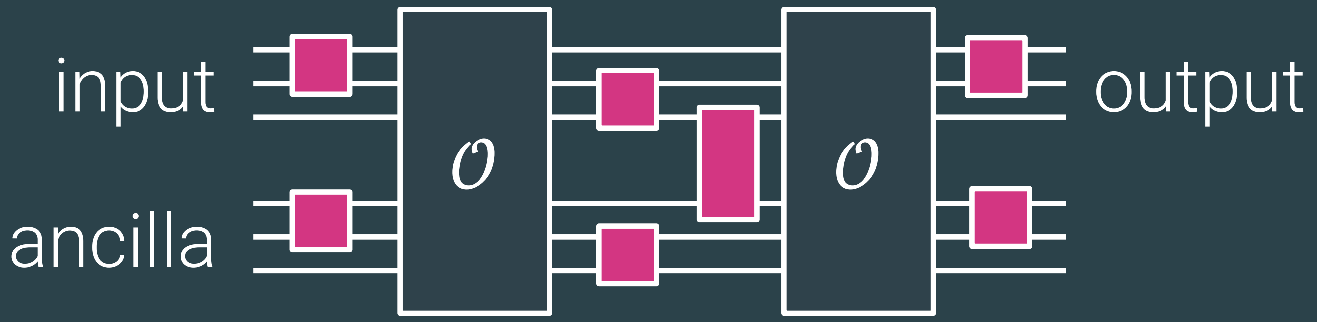
The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

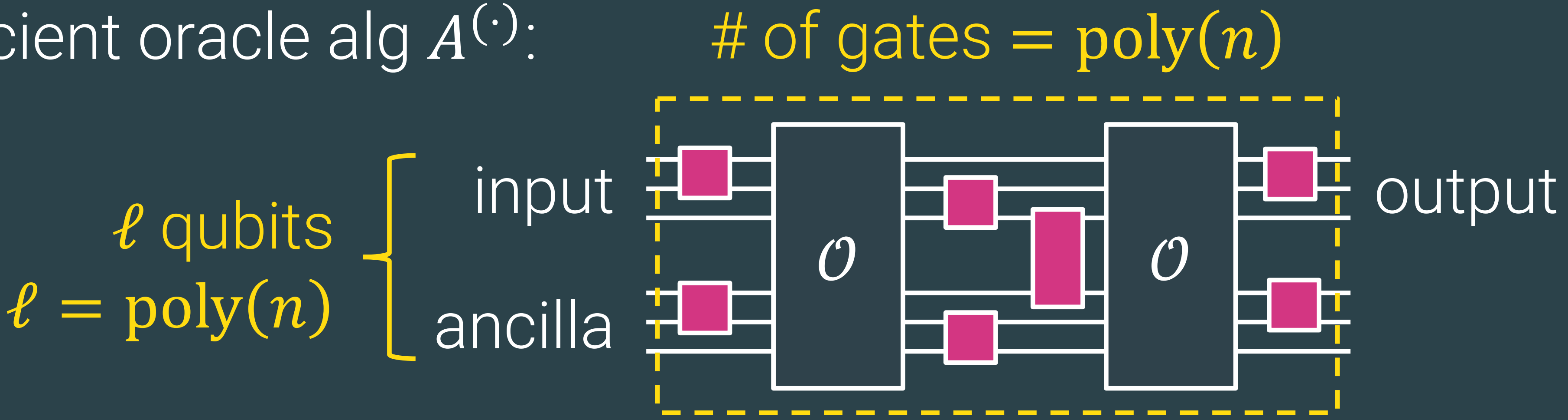
1) Efficient oracle alg $A^{(\cdot)}$:



The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

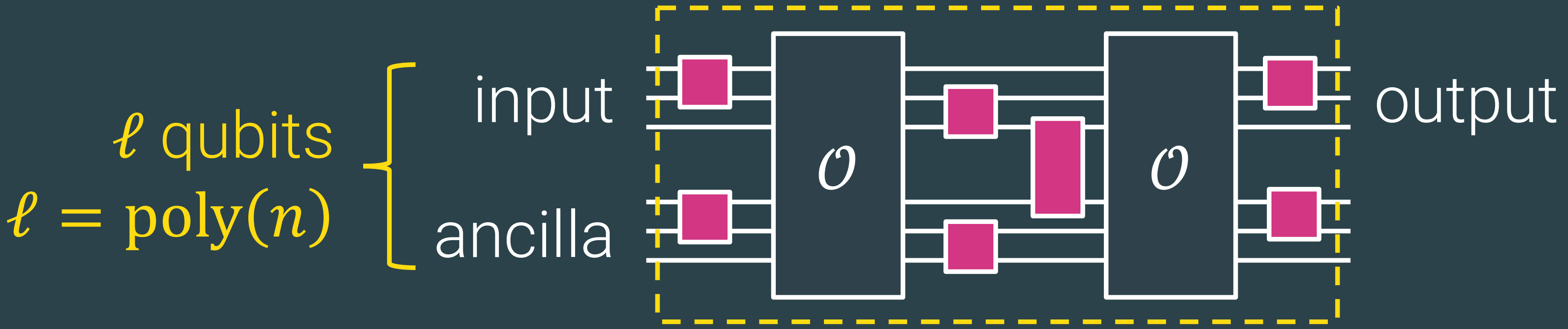
1) Efficient oracle alg $A^{(\cdot)}$:



The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

1) Efficient oracle alg $A^{(\cdot)}$: # of gates $= \mathbf{poly}(n)$

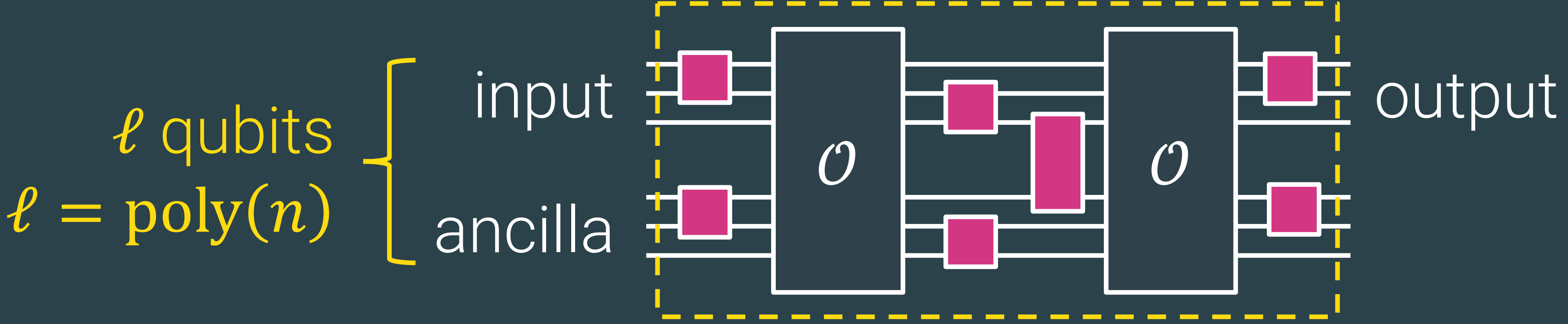


2) Given U, pick $f: \{0,1\}^{\ell} \to \{\pm 1\}$.

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

1) Efficient oracle alg $A^{(\cdot)}$: # of gates $= \mathbf{poly}(n)$

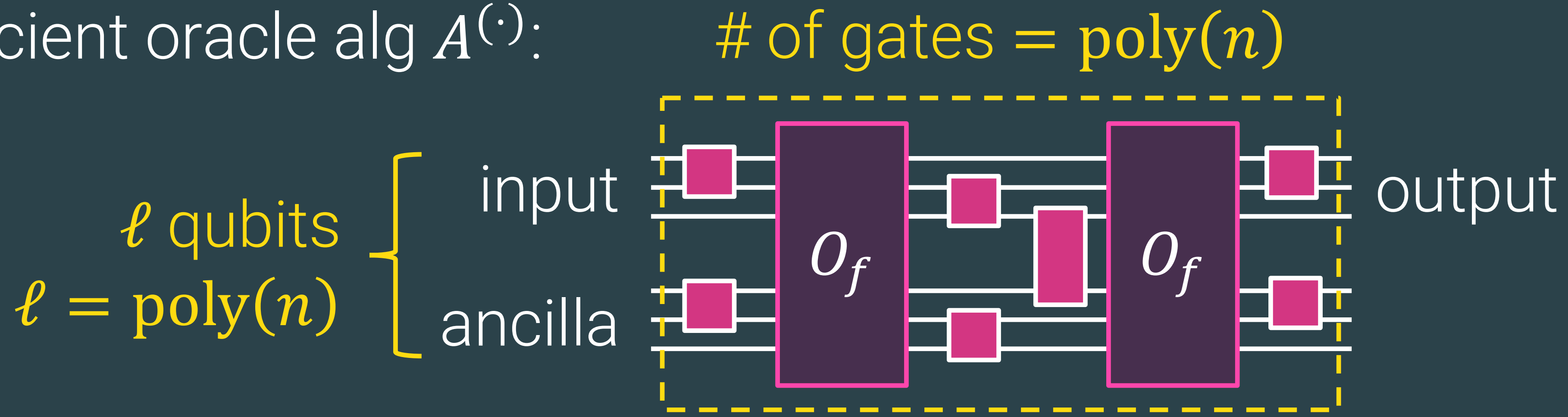


2) Given U, pick $f: \{0,1\}^\ell \to \{\pm 1\}$. Plug in $O_f: |z\rangle \to f(z) \cdot |z\rangle$.

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

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



2) Given U, pick $f: \{0,1\}^\ell \to \{\pm 1\}$. 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 $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

Prior best-known bounds

- Upper bound: $2^{n/2}$ queries [Ros22]

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

Prior best-known bounds

- Upper bound: $2^{n/2}$ queries [Ros22]
- Lower bound: none

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

Prior best-known bounds

- Upper bound: $2^{n/2}$ queries [Ros22]
- Lower bound: none

  Note: [AK06] prove a 1-query lower bound for a very special class of oracle 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 $f: \{0,1\}^{\ell} \to \{\pm 1\}$.

Useless for $\ell > 2n$.

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 $f\colon \{0,1\}^{\ell} \to \{\pm 1\}$.

Useless for $\ell > 2n$.

(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 $f: \{0,1\}^{\ell} \to \{\pm 1\}$.

Useless for $\ell > 2n$.

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

In fact, they can solve any **classical input**, **quantum output** problem.

[Aar16, INNRY22, Ros23]

This work

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 any algorithm that queries $f: \{0,1\}^{\ell} \to \{\pm 1\}$ on inputs of bounded length $\ell = o(2^n)$

This work

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

In fact, we rule out any algorithm that queries $f: \{0,1\}^{\ell} \to \{\pm 1\}$ on inputs of bounded length $\ell = o(2^n)$ even if they have:

- 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 any algorithm that queries $f: \{0,1\}^{\ell} \rightarrow \{\pm 1\}$ on inputs of bounded length $\ell = o(2^n)$ even if they have:

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

Note: when $\ell = 2^{2n}$, possible to learn description of U in one query.

Rest of this talk

Part 1:
Connect unitary synthesis to breaking quantum cryptography

Part 2:
A special case of our proof (if time)

Rest of this talk

Part 1:
Connect unitary synthesis to breaking quantum cryptography

Part 2:
A special case of our proof (if time)

We prove our result by studying **pseudorandom states (PRS)**.

We prove our result by studying **pseudorandom states (PRS)**.

PRS: efficiently-constructible family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

We prove our result by studying **pseudorandom states (PRS)**.

PRS: efficiently-constructible family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- Pseudorandom state $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$

We prove our result by studying **pseudorandom states (PRS)**.

PRS: efficiently-constructible family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- Pseudorandom state $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

We prove our result by studying **pseudorandom states (PRS)**.

PRS: efficiently-constructible family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- Pseudorandom state $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

Existence of secure PRS implies quantum bit commitments, secure computation, and many other important primitives.

We prove our result by studying **pseudorandom states (PRS)**.

PRS: efficiently-constructible family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- Pseudorandom state $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

Existence of secure PRS implies quantum bit commitments, secure computation, and many other important primitives.

Fundamental question: how hard is it to break a PRS?

We prove our result by studying **pseudorandom states (PRS)**.

PRS: efficiently-constructible family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- Pseudorandom state $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

Existence of secure PRS implies quantum bit commitments, secure computation, and many other important primitives.

Fundamental question: how hard is it to break a PRS?

Our answer: probably harder than computing any function.

We prove our result by studying **pseudorandom states (PRS)**.

PRS: efficiently-constructible family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- Pseudorandom state $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

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

We prove our result by studying **pseudorandom states (PRS)**.

PRS: efficiently-constructible family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- Pseudorandom state $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

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

Note: this result implies our unitary synthesis lower bound.

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the corresponding binary phase state $|\psi_h\rangle := \frac{1}{\sqrt{N}} \sum_{x \in [N]} h(x)|x\rangle$. (recall $N = 2^n$)

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the corresponding binary phase state $|\psi_h\rangle \coloneqq \frac{1}{\sqrt{N}} \sum_{x \in [N]} h(x) |x\rangle$. (recall $N = 2^n$)

PRS construction: given random oracle $R: [K] \times [N] \to \{\pm 1\}$, our PRS family is $\{|\psi_{R_k}\rangle\}_{k \in [K]}$ where $R_k(x) \coloneqq R(k, x)$.

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the corresponding binary phase state $|\psi_h\rangle \coloneqq \frac{1}{\sqrt{N}} \sum_{x \in [N]} h(x)|x\rangle$. (recall $N = 2^n$)

PRS construction: given random oracle $R: [K] \times [N] \to \{\pm 1\}$, our PRS family is $\{|\psi_{R_k}\rangle\}_{k \in [K]}$ where $R_k(x) \coloneqq R(k, x)$.

Adversary's task is to distinguish:

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the corresponding binary phase state $|\psi_h\rangle := \frac{1}{\sqrt{N}} \sum_{x \in [N]} h(x) |x\rangle$. (recall $N = 2^n$)

PRS construction: given random oracle $R: [K] \times [N] \to \{\pm 1\}$, our PRS family is $\{|\psi_{R_k}\rangle\}_{k \in [K]}$ where $R_k(x) := R(k, x)$.

Adversary's task is to distinguish:

- $|\psi_{R_k}\rangle$ for uniformly random $k \leftarrow [K]$

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the corresponding binary phase state $|\psi_h\rangle := \frac{1}{\sqrt{N}} \sum_{x \in [N]} h(x)|x\rangle$. (recall $N = 2^n$)

PRS construction: given random oracle $R: [K] \times [N] \to \{\pm 1\}$, our PRS family is $\{|\psi_{R_k}\rangle\}_{k \in [K]}$ where $R_k(x) := R(k, x)$.

Adversary's task is to distinguish:

- $|\psi_{R_k}\rangle$ for uniformly random $k \leftarrow [K]$
- $|\psi_h\rangle$ for uniformly random $h: [N] \to \{\pm 1\}$

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the corresponding binary phase state $|\psi_h\rangle \coloneqq \frac{1}{\sqrt{N}} \sum_{x \in [N]} h(x) |x\rangle$. (recall $N = 2^n$)

PRS construction: given random oracle $R: [K] \times [N] \to \{\pm 1\}$, our PRS family is $\{|\psi_{R_k}\rangle\}_{k \in [K]}$ where $R_k(x) \coloneqq R(k, x)$.

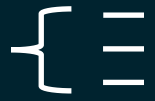
Adversary's task is to distinguish:

- $|\psi_{R_k}\rangle$ for uniformly random $k \leftarrow [K]$
- $|\psi_h\rangle$ for uniformly random $h: [N] \to \{\pm 1\}$

given 1 query to a function f, which can depend on R.

Next up: what does a one-query adversary look like?

One-query adversaries

input $|\psi\rangle$ 

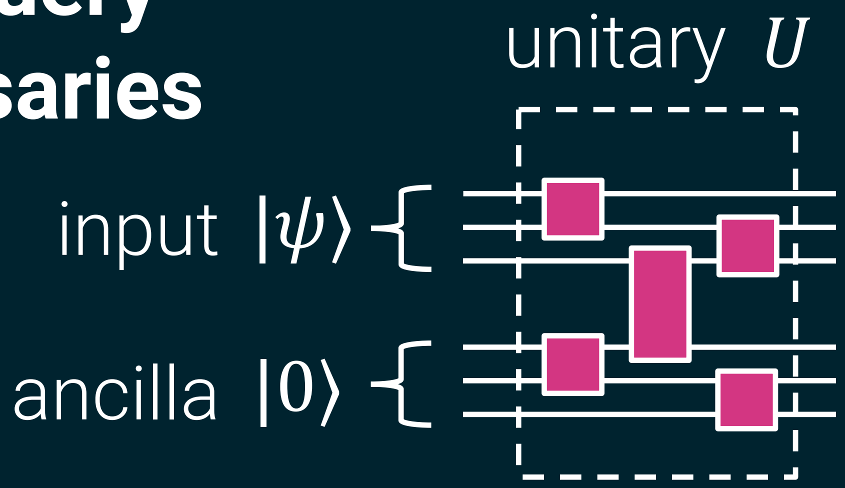
One-query adversaries

input $|\psi\rangle$

ancilla $|0\rangle$

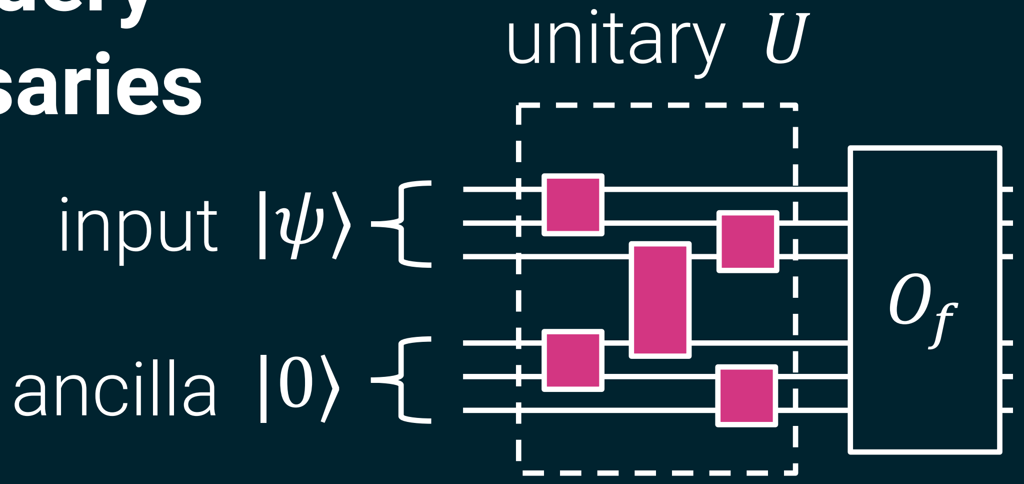
1) Initialize $\ell - n$ ancilla qubits

One-query adversaries



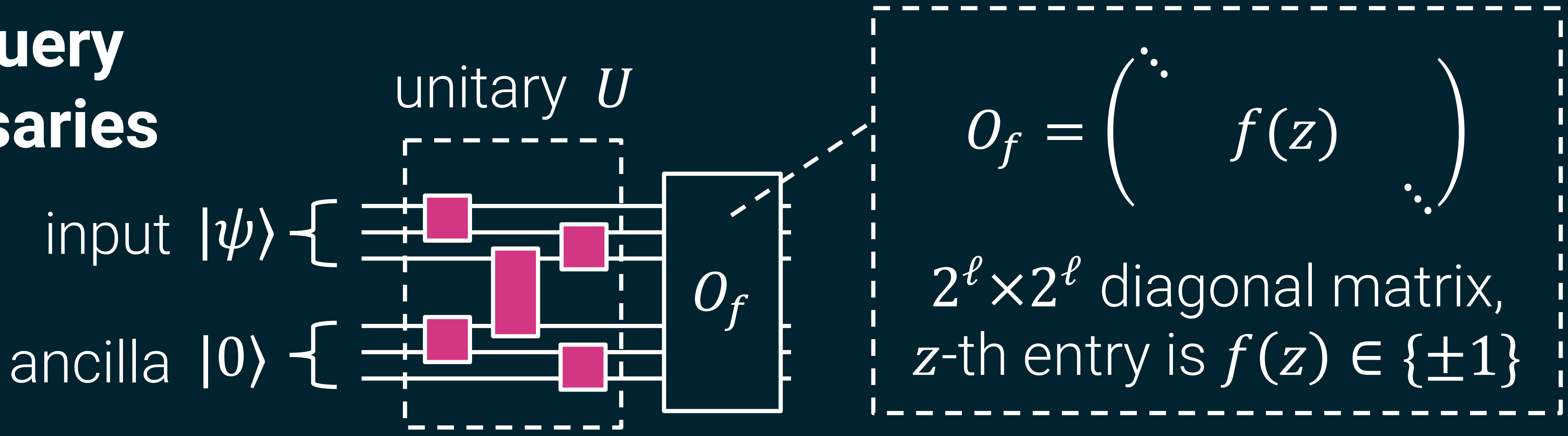
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.

One-query adversaries



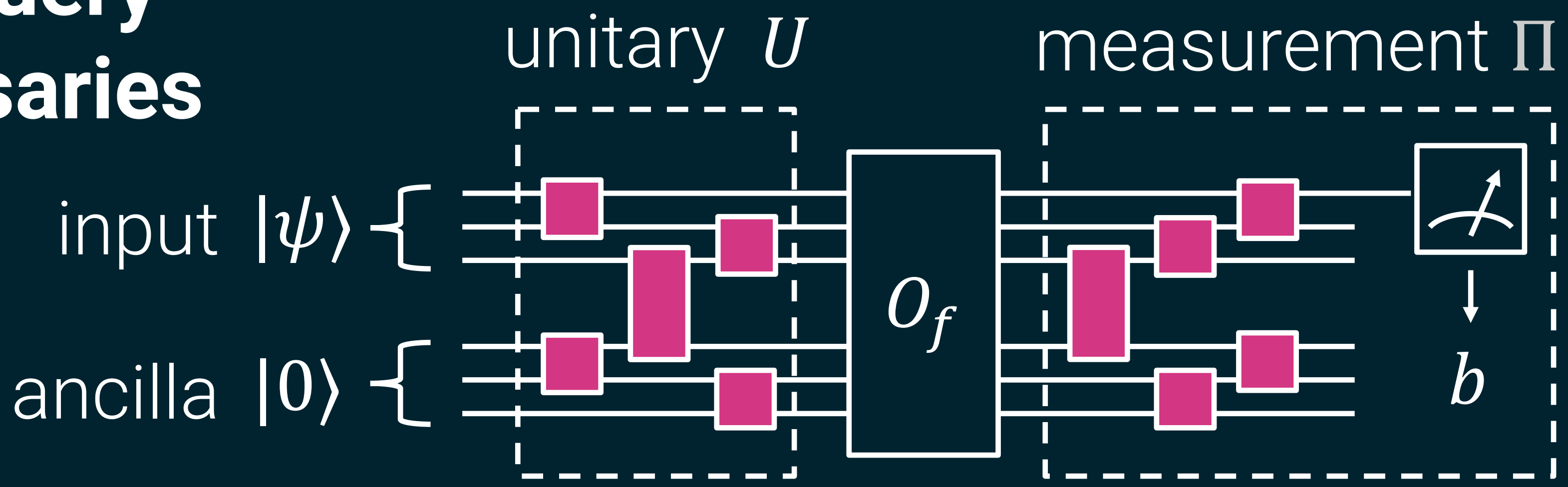
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.

One-query adversaries



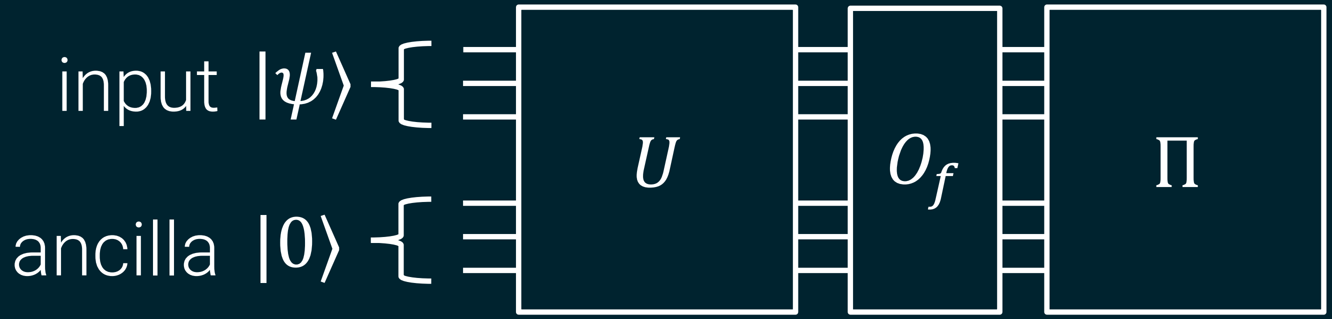
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \rightarrow f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.

One-query adversaries



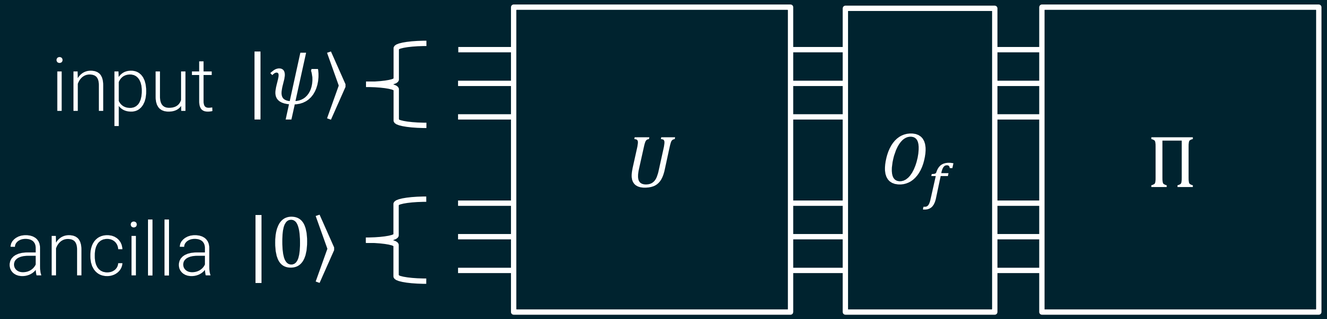
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.
4) Measure $\{\Pi, \mathrm{I} - \Pi\}$ and return 1 if outcome is Π.

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.

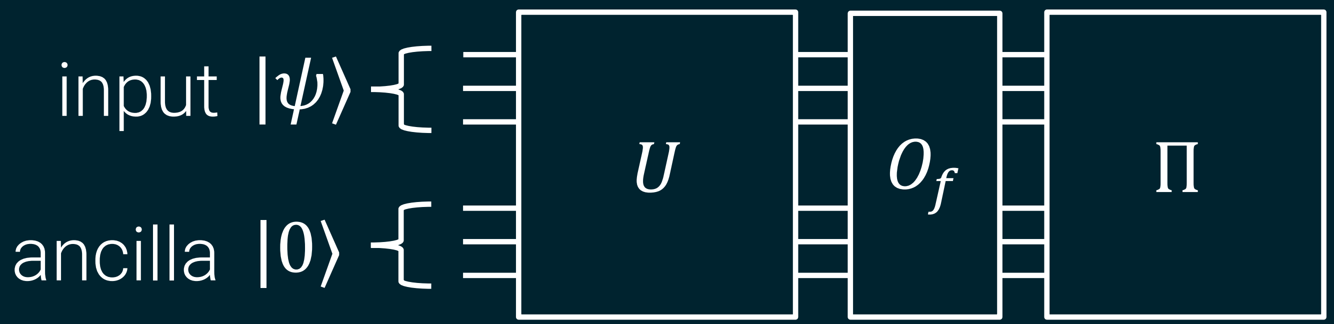
One-query adversaries



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

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

One-query adversaries

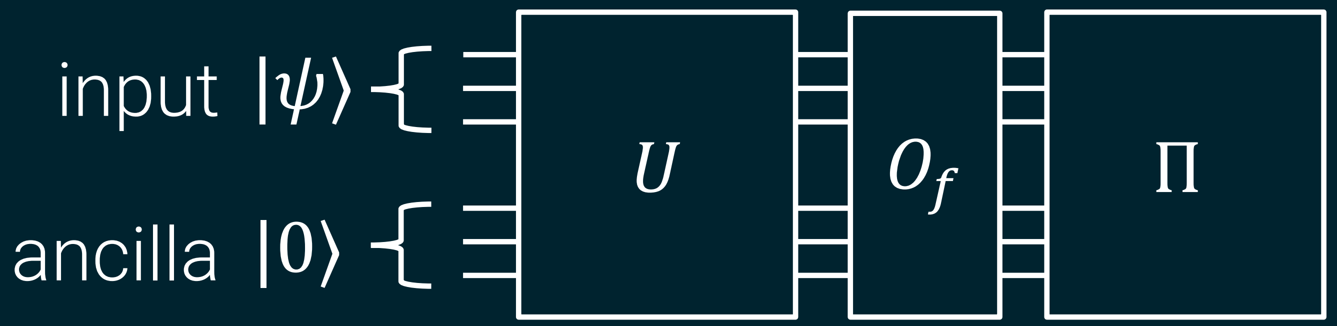


$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot U \cdot |\psi\rangle|0\rangle \right\|^2$$

Adversary's distinguishing advantage for fixed R 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



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

Adversary's distinguishing advantage for fixed R 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!

Adversary's distinguishing advantage for fixed R 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.

Adversary's distinguishing advantage for fixed R 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**.

Adversary's distinguishing advantage for fixed R 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)

Rest of this talk

Part 1:
Connect unitary synthesis to breaking quantum cryptography

Part 2:
A special case of our proof (if time)

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

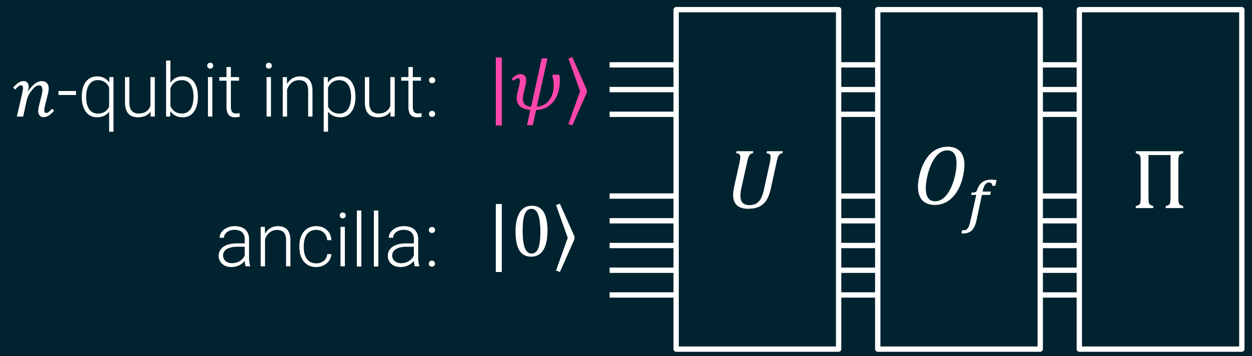
Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathbf{Id}$.

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 $U = \mathrm{Id}$.

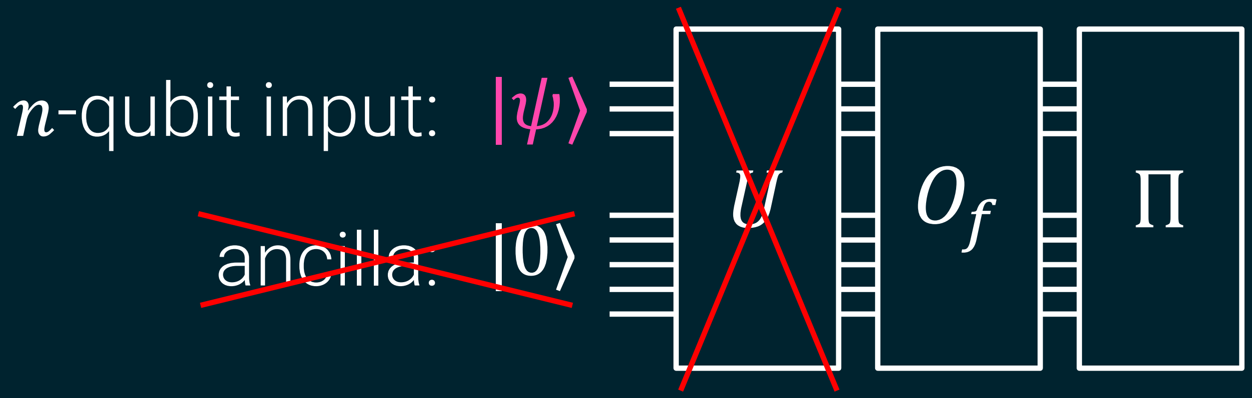
One-query adversaries:



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 $U = \mathrm{Id}$.

Special class: n-qubit input: $|\psi\rangle \equiv \boxed{O_f} \equiv \boxed{\Pi}$

A special class of one-query adversaries

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

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

projection

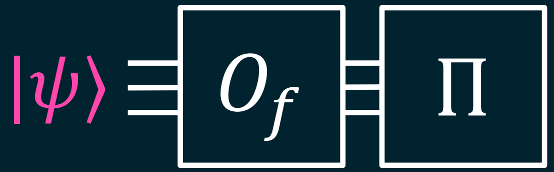
$$O_f = \begin{pmatrix} \ddots & & \\ & f(x) & \\ & & \ddots \end{pmatrix}$$

$N \times N$ diagonal matrix,
x-th entry is $f(x) \in \{\pm 1\}$

A special class of one-query adversaries

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

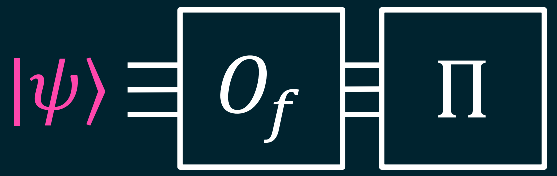
Special class: n-qubit input:



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

A special class of one-query adversaries

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

Special class: n-qubit input: 

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

Distinguishing advantage:

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

(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. $X_1, \ldots, X_K$

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

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

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

Adversary's advantage (for this special class):

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

Adversary's advantage (for this special class):

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

Matrix Chernoff:

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

Adversary's advantage (for this special class):

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

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

max over unit vectors

Adversary's advantage (for this special class):

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

random vectors

$\Downarrow$?

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):

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

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

$$= \frac{1}{K}\sum_k X_k - E[X]$$

f-dependent unit vector

Adversary's advantage (for this special class):

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

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

$$= \frac{1}{K}\sum_k X_k - E[X]$$

f-dependent unit vector

Then matrix Chernoff will bound the max over all unit vectors.

Adversary's advantage (for this special class):

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

Since all the terms look identical, it suffices to just look at one term.

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

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

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

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

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

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

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

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

(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 $\underbrace{\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle}_{}$

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

(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 $\langle v_f | \cdot \text{(random matrix)} \cdot | v_f \rangle$

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

(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 $\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle$

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

(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 $\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle$

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

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

(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 $D_{R\,k}$

So we can rewrite the distinguishing advantage as

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

So we can rewrite the distinguishing advantage as

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

So we can rewrite the distinguishing advantage as

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

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

So we can rewrite the distinguishing advantage as

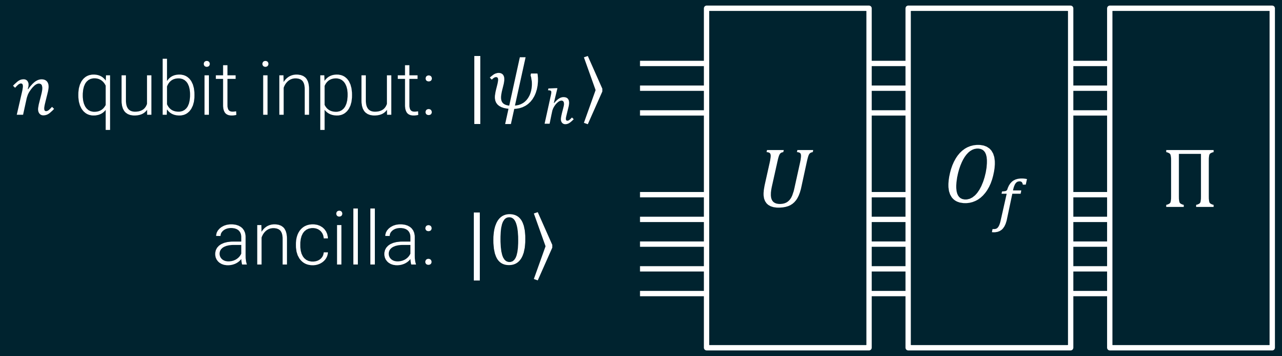
$$\langle +_N | O_f \left(\frac{1}{K} \sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h [D_h \cdot \Pi \cdot D_h] \right) \underbrace{O_f | +_N \rangle}_{\text{unit vector}}$$

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

by Matrix Chernoff on the i.i.d. bounded random matrices $D_{R_k} \cdot \Pi \cdot D_{R_k}$.

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

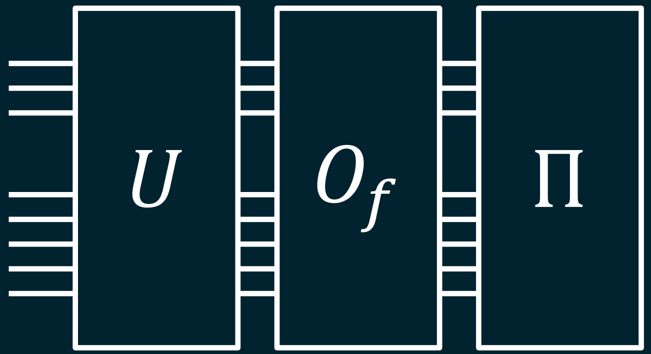
General one-query adversaries



General one-query adversaries

n qubit input: $|\psi_h\rangle$

ancilla: $|0\rangle$

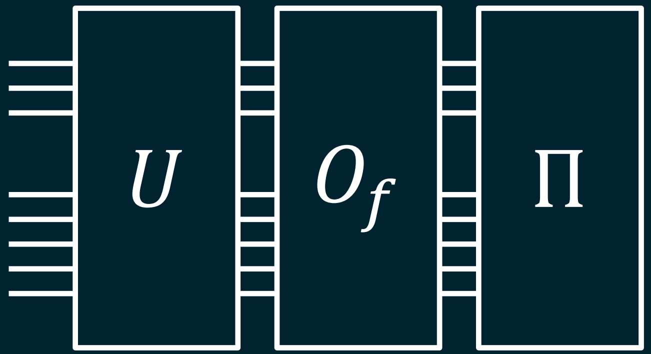


Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. “add ancillas + apply U”

General one-query adversaries

n qubit input: $|\psi_h\rangle$

ancilla: $|0\rangle$



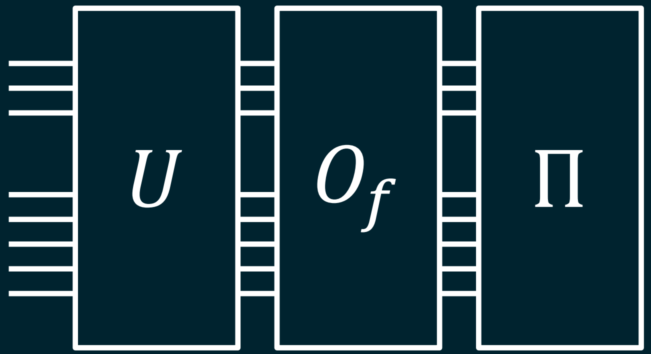
Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. "add ancillas + apply U"

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

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"

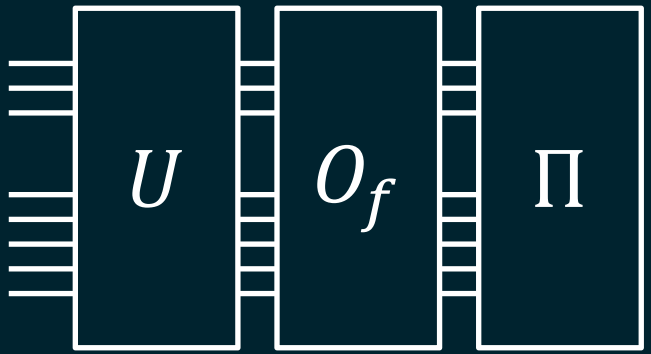
$$\Pr[A^f(|\psi_h\rangle) \text{ outputs } 1] = \langle +_N|\, \underbrace{D_h \cdot V^\dagger \cdot O_f} \cdot \Pi \cdot \underbrace{O_f \cdot V \cdot D_h}\, |+_N\rangle$$

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"

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

Challenge: unclear how to commute D_h and O_f!

Our solution: factor $V\,|\psi_h\rangle = \widetilde{D_h} \cdot |\mathrm{wt}_V\rangle$ w.r.t. a V-dependent unit vector $|\mathrm{wt}_V\rangle$ to obtain spectral relaxation.

Conclusions

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

Conclusions

- Implementing unitaries and breaking quantum crypto might be harder than any classical problem.
- Possibly no complexity-theoretic barriers to **unconditionally** proving hardness for many quantum tasks?

Conclusions

- Implementing unitaries and breaking quantum crypto might be harder than any classical problem.
- Possibly no complexity-theoretic barriers to **unconditionally** proving hardness for many quantum tasks?

Next steps:

Non-synthesis conjecture: our PRS 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 any classical problem.
- Possibly no complexity-theoretic barriers to **unconditionally** proving hardness for many quantum tasks?

Next steps:

Non-synthesis conjecture: our PRS 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 any classical problem.
- Possibly no complexity-theoretic barriers to **unconditionally** proving hardness for many quantum tasks?

Next steps:

Non-synthesis conjecture: our PRS 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!