The cryptographic nature of quantum computation

Fermi Ma

UC Berkeley & Simons Institute

I study **quantum computers**.

I study **quantum computers**.

devices that exploit
quantum phenomena

I study **quantum computers**.

devices that exploit
quantum phenomena

I also study **cryptography**.

I study **quantum computers**.

devices that exploit
quantum phenomena

I also study **cryptography**.

build protocols that provably
resist adversarial behavior.

Quantum computers will revolutionize cryptography

Quantum computers will revolutionize cryptography

New threats

Shor's algorithm breaks RSA!

Quantum computers will revolutionize cryptography

New threats

Shor's algorithm breaks RSA!

How do we build classical cryptography that resists quantum attack?

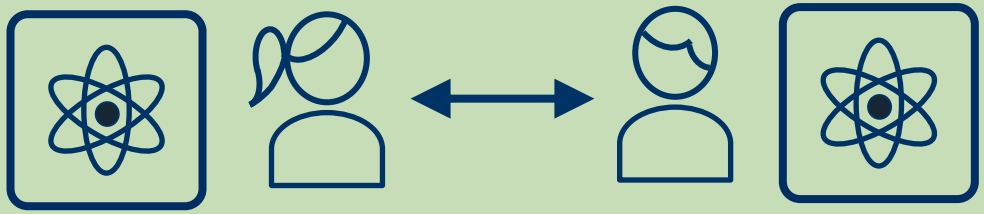
Quantum computers will revolutionize cryptography

New threats

Shor's algorithm breaks RSA!

How do we build classical cryptography that resists quantum attack?

New opportunities



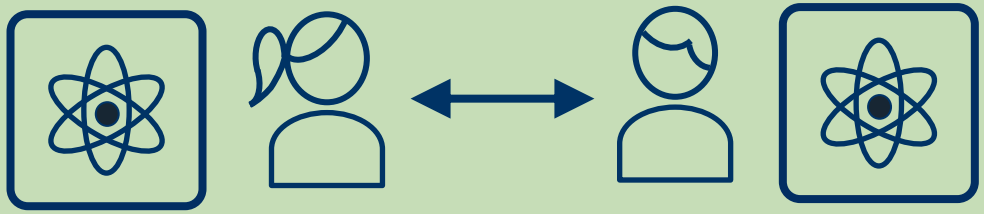
Quantum computers will revolutionize cryptography

New threats

Shor's algorithm breaks RSA!

How do we build classical cryptography that resists quantum attack?

New opportunities



Can we leverage quantum mechanics to build better cryptographic protocols?

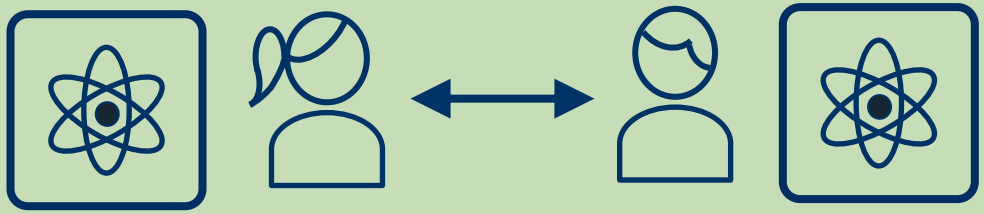
Quantum computers will revolutionize cryptography

New threats

Shor's algorithm breaks RSA!

How do we build classical cryptography that resists quantum attack?

New opportunities



Can we leverage quantum mechanics to build better cryptographic protocols?

quantum cryptography

But quantum cryptography is not just about cryptography…

Recently, quantum cryptography has been showing up in **fundamental physics**.

Recently, quantum cryptography has been showing up in **fundamental physics**.

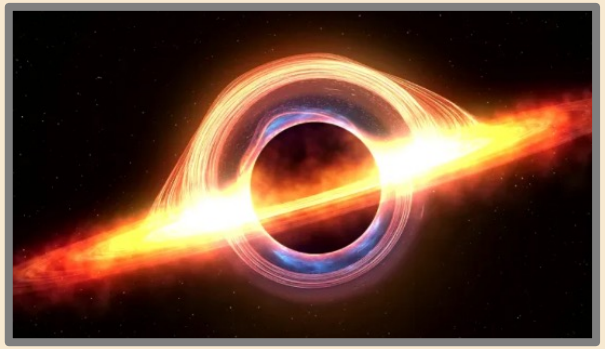
Ex: AMPS firewall paradox.

Recently, quantum cryptography has been showing up in **fundamental physics**.

Ex: AMPS firewall paradox. Involves a thought experiment in which an observer “decodes” black hole radiation.

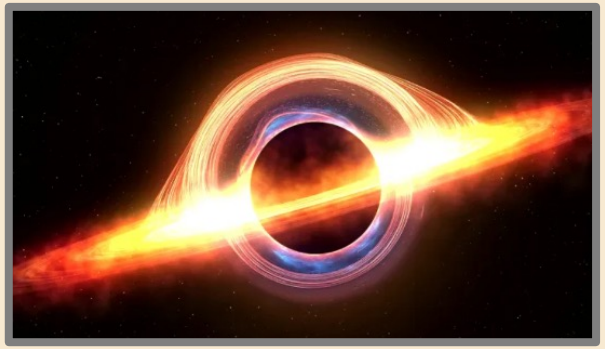
Recently, quantum cryptography has been showing up in **fundamental physics**.

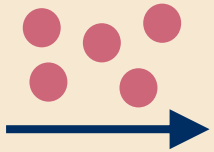
Ex: AMPS firewall paradox. Involves a thought experiment in which an observer "decodes" black hole radiation.



Recently, quantum cryptography has been showing up in **fundamental physics**.

Ex: AMPS firewall paradox. Involves a thought experiment in which an observer "decodes" black hole radiation.



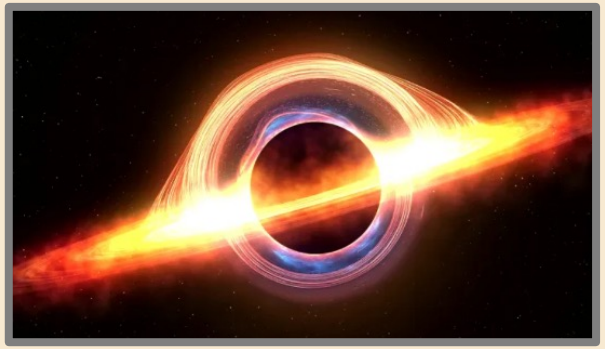


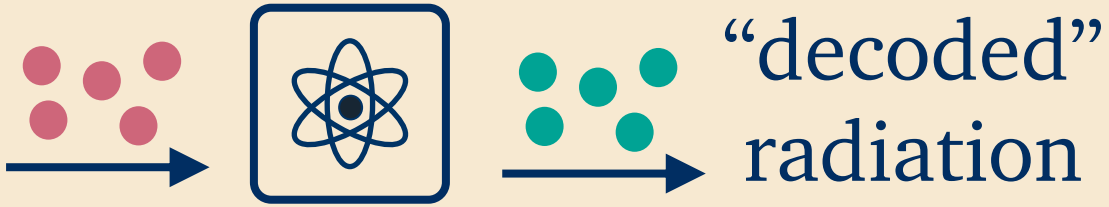
Recently, quantum cryptography has been showing up in **fundamental physics**.

Ex: AMPS firewall paradox. Involves a thought experiment in which an observer "decodes" black hole radiation.

Recently, quantum cryptography has been showing up in **fundamental physics**.

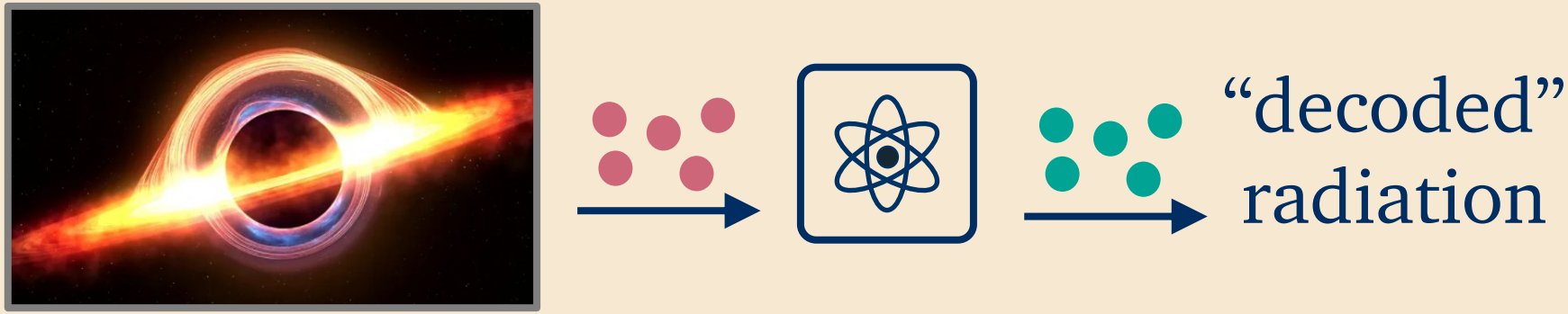
Ex: AMPS firewall paradox. Involves a thought experiment in which an observer "decodes" black hole radiation.





Recently, quantum cryptography has been showing up in **fundamental physics**.

Ex: AMPS firewall paradox. Involves a thought experiment in which an observer "decodes" black hole radiation.



[HH13] counterargument: this is **cryptographically** hard!

Recently, quantum cryptography has been showing up in **fundamental physics**.

Now common practice: model chaotic systems using cryptographic pseudorandomness

Recently, quantum cryptography has been showing up in **fundamental physics**.

Now common practice: model chaotic systems using cryptographic pseudorandomness

Kim-Preskill (2023)

"We shall assume that the unitary U that describes the formation and the evaporation of the black hole is **pseudorandom**"

Recently, quantum cryptography has been showing up in **fundamental physics**.

Now common practice: model chaotic systems using cryptographic pseudorandomness

Kim-Preskill (2023)

"We shall assume that the unitary U that describes the formation and the evaporation of the black hole is **pseudorandom**"

Yang-Engelhardt (2023)

"We use the common simplification of modeling black holes and more generally chaotic systems via **(pseudo)random dynamics**"

My research goals

My research goals

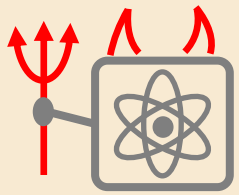
1) Develop theoretical foundations for quantum cryptography.

My research goals

1) Develop theoretical foundations for quantum cryptography.

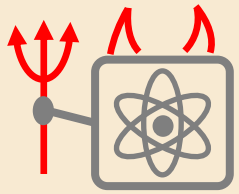
2) Apply these insights to the broader **theory of computation** and **fundamental physics**.

Highlights of my research



quantum
security

Highlights of my research

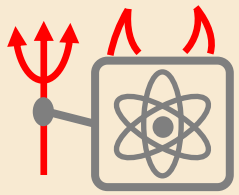


quantum security

- [CMSZ21, LMS22]: proved that textbook crypto protocols are quantum secure via “quantum rewinding”

  (FOCS 2021 special issue, FOCS 2022)

Highlights of my research



quantum security

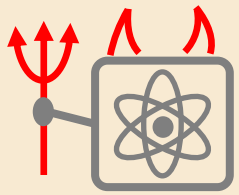
- [CMSZ21, LMS22]: proved that textbook crypto protocols are quantum secure via “quantum rewinding”

  (FOCS 2021 special issue, FOCS 2022)

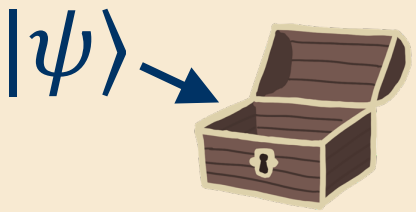
- These techniques have found widespread use

  [BBK22,BKLMMVVY23,BQSY24,CDGS24,MNZ24,…]

Highlights of my research

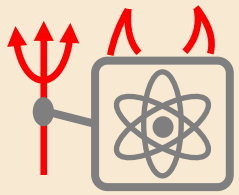


quantum security

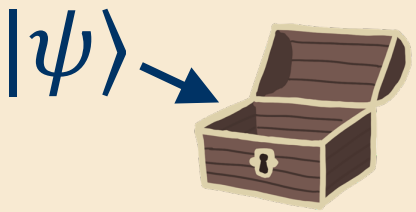


quantum primitives

Highlights of my research



quantum security

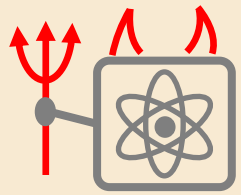


quantum primitives

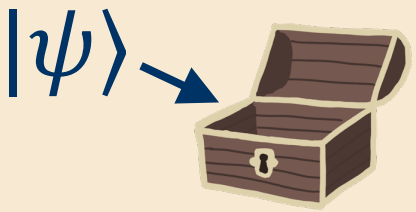
- [GJMZ23]: introduced commitments to quantum states

  (STOC 2023 special issue)

Highlights of my research



quantum security



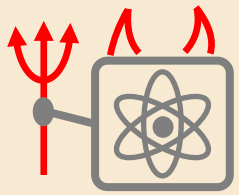
quantum primitives

- [GJMZ23]: introduced commitments to quantum states

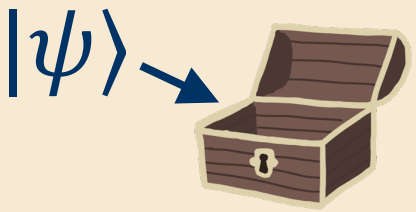
  (STOC 2023 special issue)

- [HH13,A16,B23]+[M23]: decoding black hole radiation is equivalent to breaking a “maximally entangled” commitment

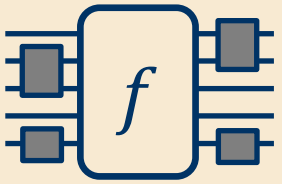
Highlights of my research



quantum security

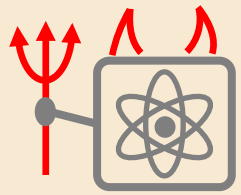


quantum primitives

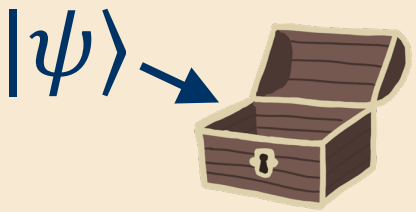


quantum hardness

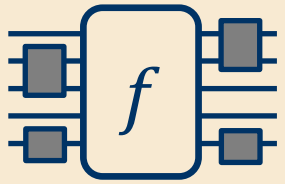
Highlights of my research



quantum security



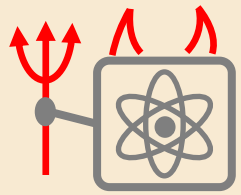
quantum primitives



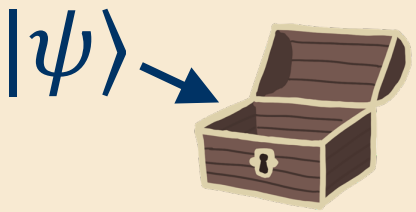
quantum hardness

How hard is it to break quantum crypto?

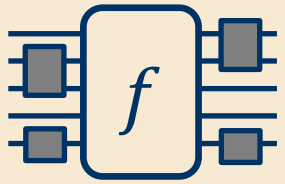
Highlights of my research



quantum security



quantum primitives

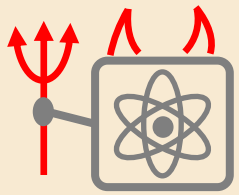


quantum hardness

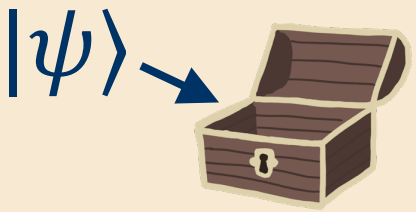
How hard is it to break quantum crypto?

- [K21,KQST23]: a magic device that solves NP-hard problems isn't enough!

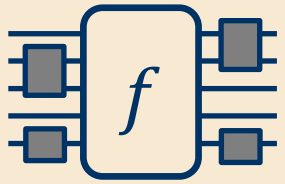
Highlights of my research



quantum security



quantum primitives



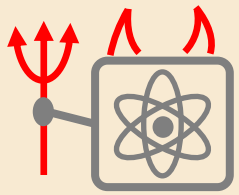
quantum hardness

How hard is it to break quantum crypto?

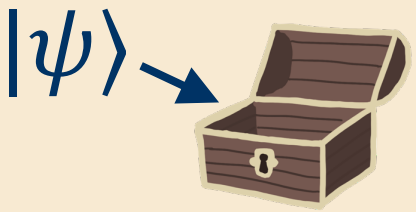
- [K21,KQST23]: a magic device that solves NP-hard problems isn't enough!
- [LMW24]: magic device that evaluates **any function (once)** isn't enough!

(STOC 2024)

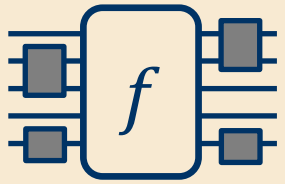
Highlights of my research



quantum security



quantum primitives



quantum hardness

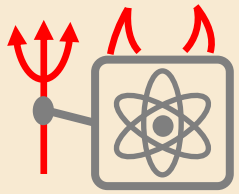
How hard is it to break quantum crypto?

- [K21,KQST23]: a magic device that solves NP-hard problems isn't enough!
- [LMW24]: magic device that evaluates **any function (once)** isn't enough! **(STOC 2024)**

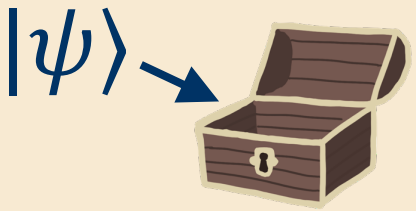
Featured in Quanta Magazine:

"Cryptographers Discover a New Foundation for Quantum Secrecy"

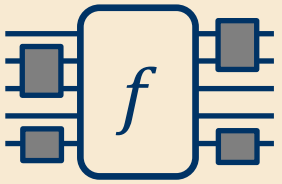
Highlights of my research



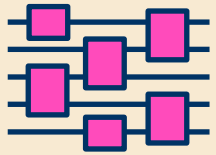
quantum security



quantum primitives

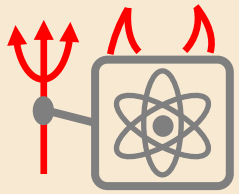


quantum hardness

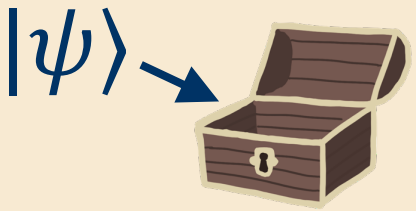


pseudorandom unitaries

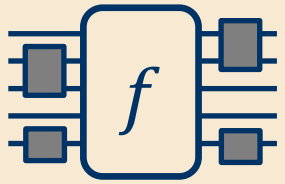
Highlights of my research



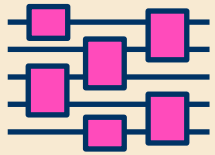
quantum security



quantum primitives



quantum hardness

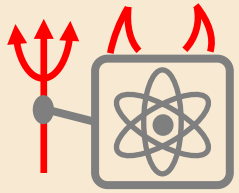


pseudorandom unitaries

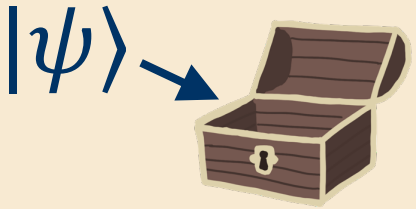
[MH24]: first proof that pseudorandom unitaries exist (under crypto assumptions).

(STOC 2025, QIP 2025 plenary talk)

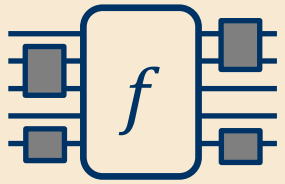
Highlights of my research



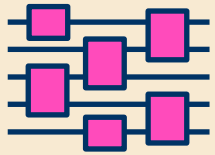
quantum security



quantum primitives



quantum hardness



pseudorandom unitaries

[MH24]: first proof that pseudorandom unitaries exist (under crypto assumptions).

(STOC 2025, QIP 2025 plenary talk)

Featured last week in Quanta Magazine:

"The High Cost of Quantum Randomness is Dropping"

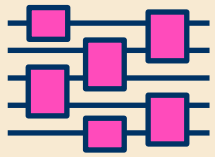
Highlights of my research

quantum security

quantum primitives

quantum hardness

Focus of today:



≈ Haar

pseudorandom unitaries

[MH24]: first proof that pseudorandom unitaries exist (under crypto assumptions).

(STOC 2025, QIP 2025 plenary talk)

Featured last week in Quanta Magazine:

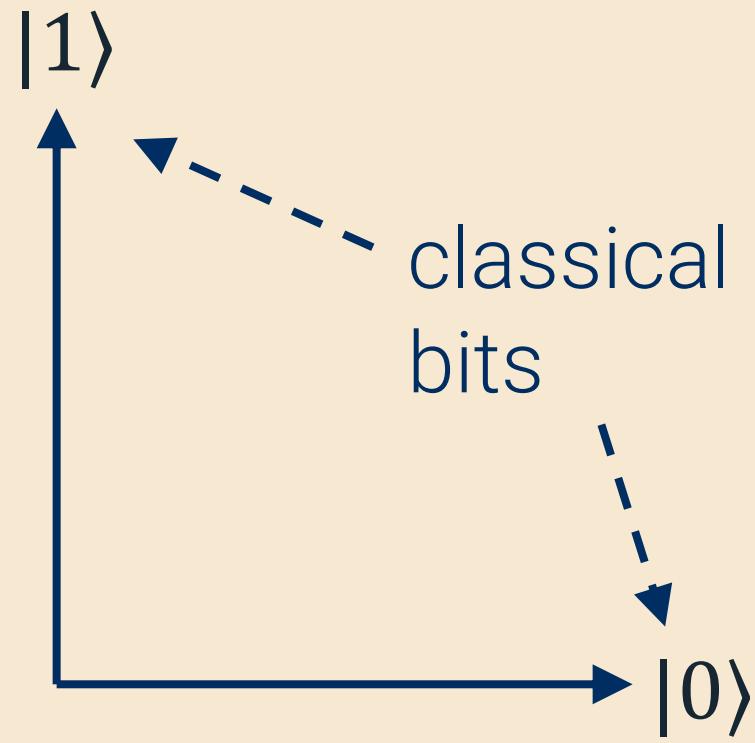
"The High Cost of Quantum Randomness is Dropping"

Pseudorandom unitaries

The quantum we'll need:

The quantum we'll need:

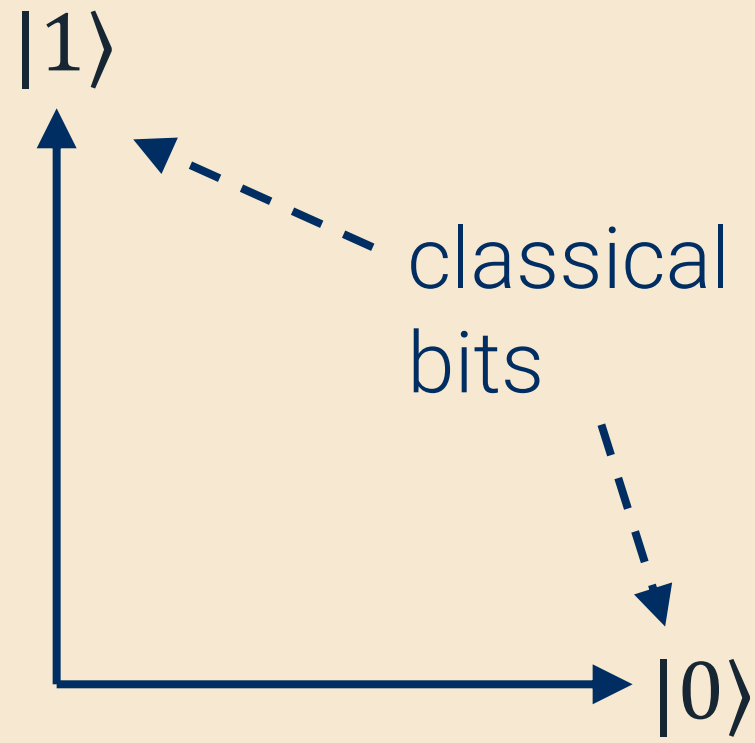
qubits: generalization of classical bits that allows **superposition**



The quantum we'll need:

qubits: generalization of classical bits that allows **superposition**

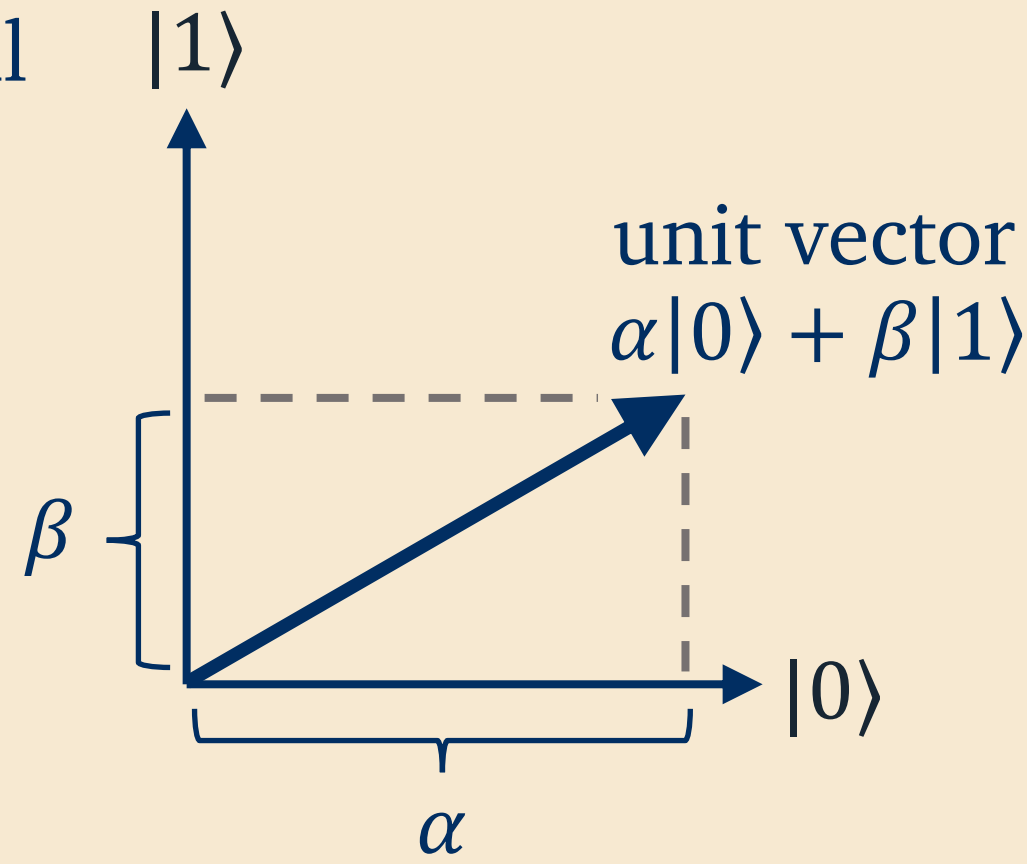
$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$$



The quantum we'll need:

qubits: generalization of classical bits that allows **superposition**

$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

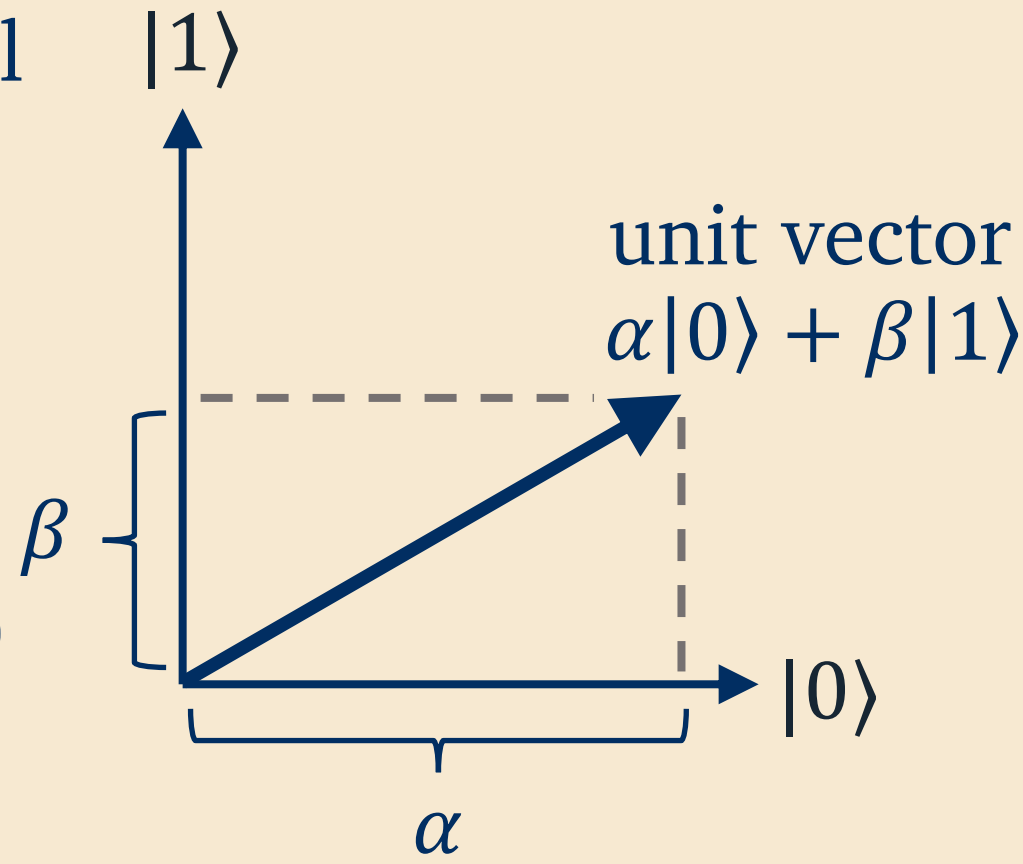


The quantum we'll need:

qubits: generalization of classical bits that allows **superposition**

$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$$

basic allowable operation: unitary transformation (rotation)

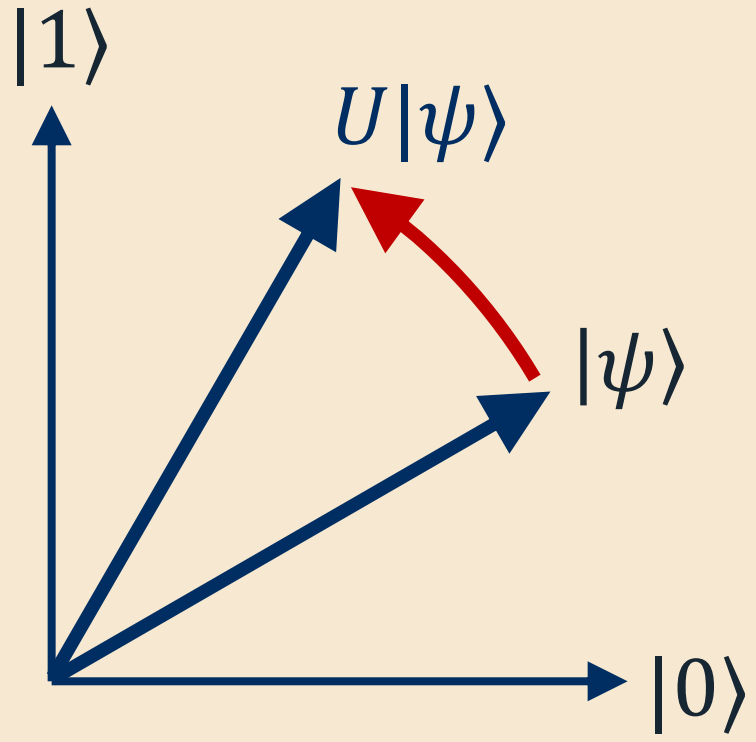


The quantum we'll need:

qubits: generalization of classical bits that allows **superposition**

$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$$

basic allowable operation: unitary transformation (rotation)

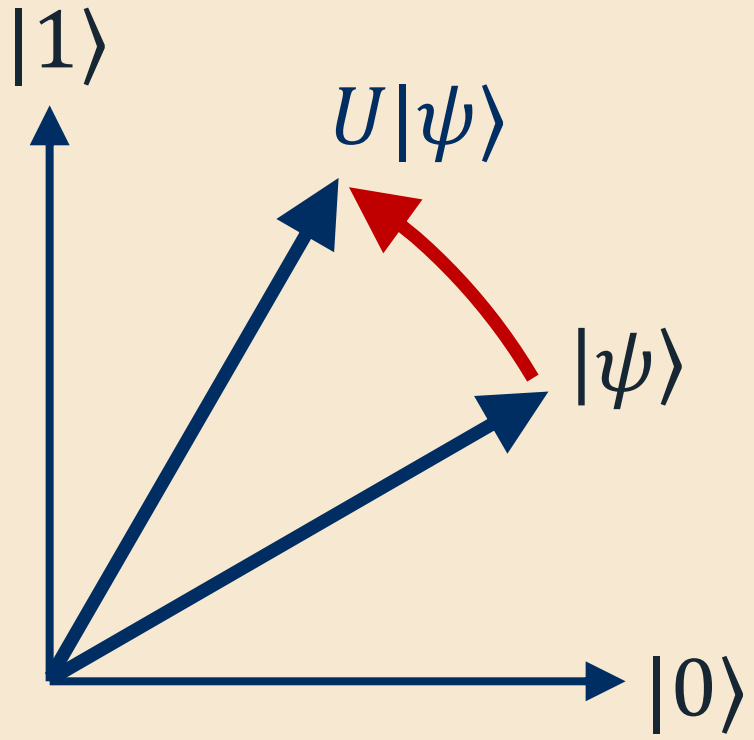


The quantum we'll need:

qubits: generalization of classical bits that allows **superposition**

$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$$

basic allowable operation: unitary transformation (rotation)



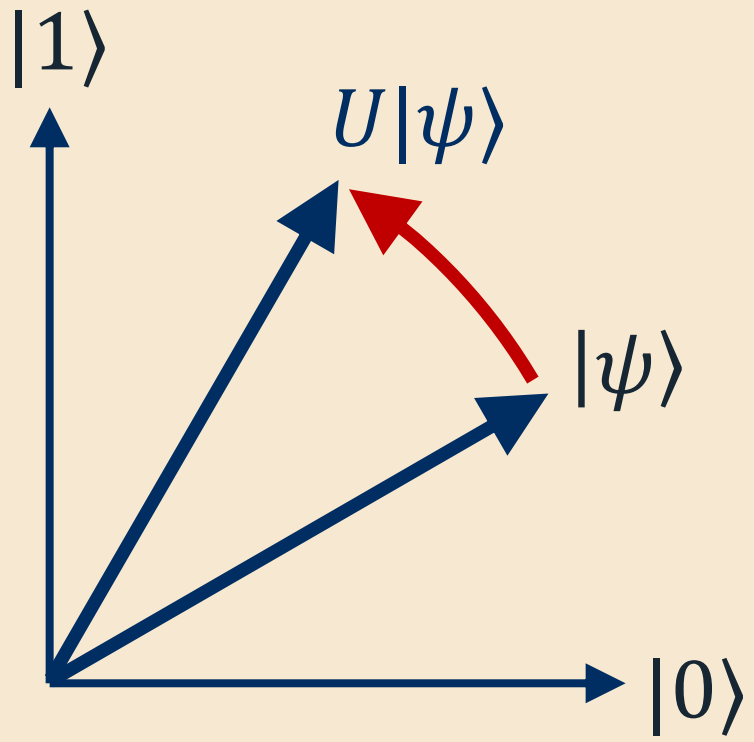
in general:

The quantum we'll need:

qubits: generalization of classical bits that allows **superposition**

$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$$

basic allowable operation:
unitary transformation (rotation)



in general:

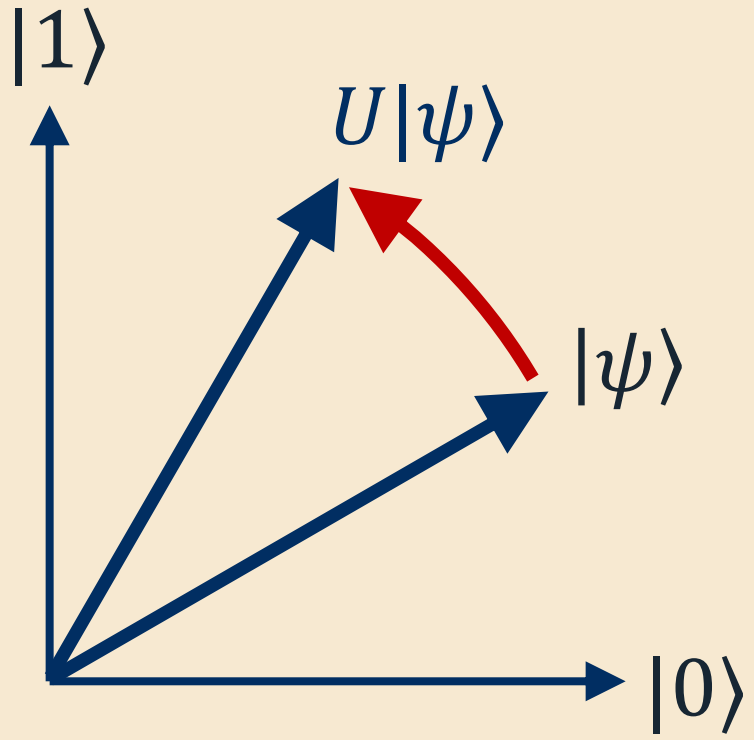
- n-qubit quantum state $\leftrightarrow$ 2^n-dimensional vector

The quantum we'll need:

qubits: generalization of classical bits that allows **superposition**

$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$$

basic allowable operation:
unitary transformation (rotation)



in general:

- n-qubit quantum state ↔ 2^n-dimensional vector
- n-qubit unitary ↔ $2^n \times 2^n$-dimensional unitary matrix

The quantum we'll need:

The quantum we'll need:

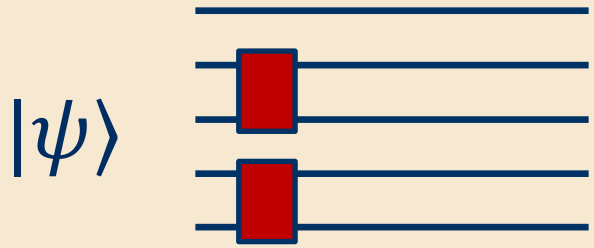
An n-qubit unitary is **efficiently computable** if it can be implemented as a poly(n)-size quantum circuit:

The quantum we'll need:

An n-qubit unitary is **efficiently computable** if it can be implemented as a poly(n)-size quantum circuit:

The quantum we'll need:

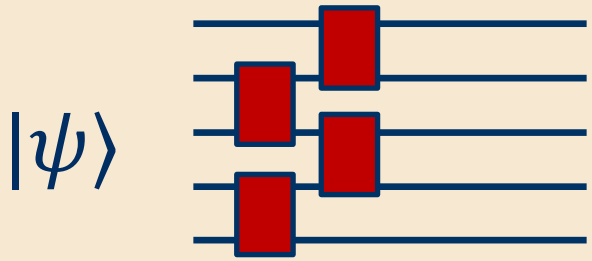
An n-qubit unitary is **efficiently computable** if it can be implemented as a poly(n)-size quantum circuit:



each gate ▮
is a 2-qubit unitary

The quantum we'll need:

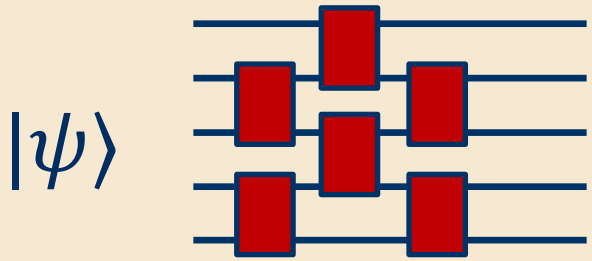
An n-qubit unitary is **efficiently computable** if it can be implemented as a poly(n)-size quantum circuit:



each gate ▮
is a 2-qubit unitary

The quantum we'll need:

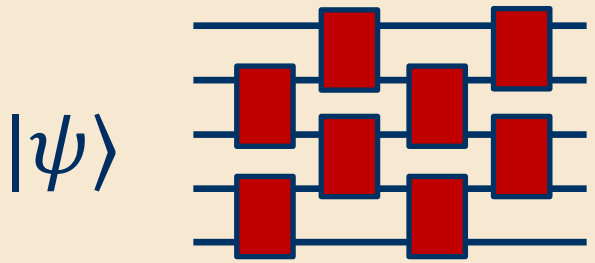
An n-qubit unitary is **efficiently computable** if it can be implemented as a poly(n)-size quantum circuit:



each gate ▮
is a 2-qubit unitary

The quantum we'll need:

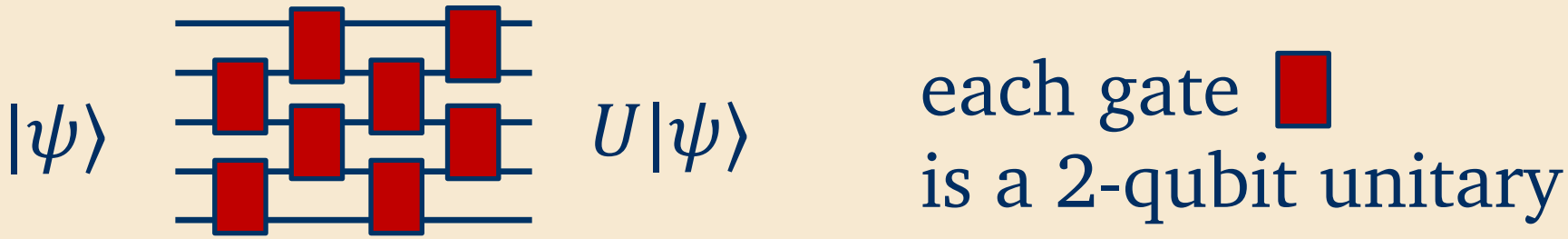
An n-qubit unitary is **efficiently computable** if it can be implemented as a poly(n)-size quantum circuit:



each gate ▮
is a 2-qubit unitary

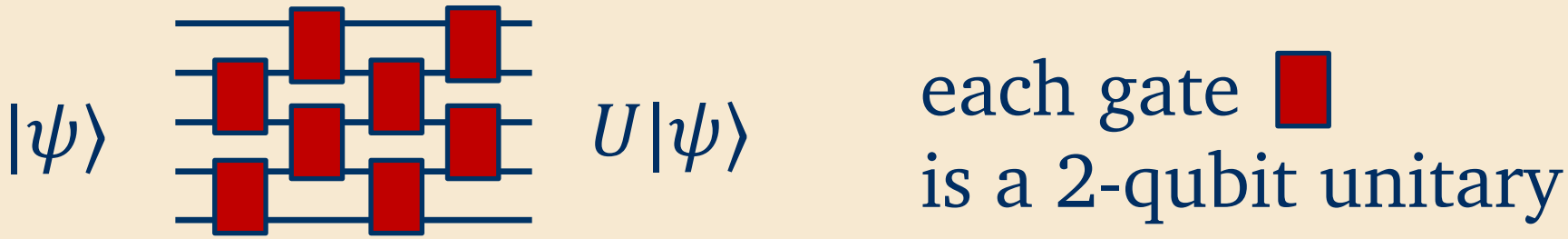
The quantum we'll need:

An n-qubit unitary is **efficiently computable** if it can be implemented as a poly(n)-size quantum circuit:



The quantum we'll need:

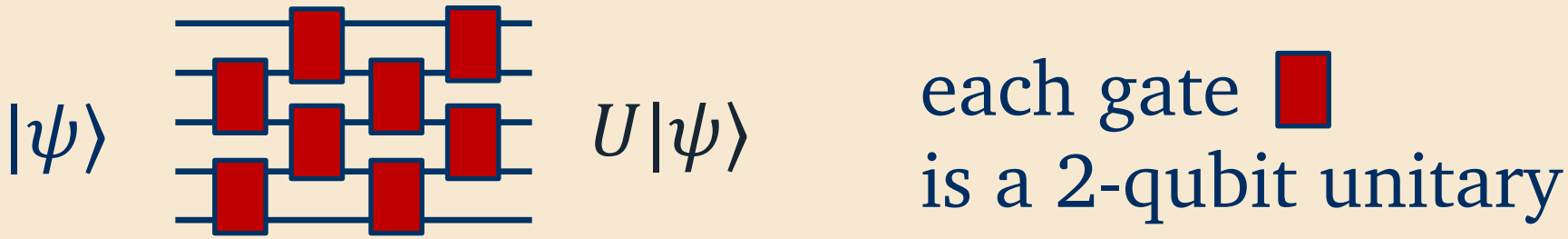
An n-qubit unitary is **efficiently computable** if it can be implemented as a poly(n)-size quantum circuit:



Not all unitaries can be implemented efficiently!

The quantum we'll need:

An n-qubit unitary is **efficiently computable** if it can be implemented as a poly(n)-size quantum circuit:



Not all unitaries can be implemented efficiently!

A random unitary is **unlikely** to be efficiently computable.

The quantum we'll need:

What exactly is a random unitary?

The quantum we'll need:

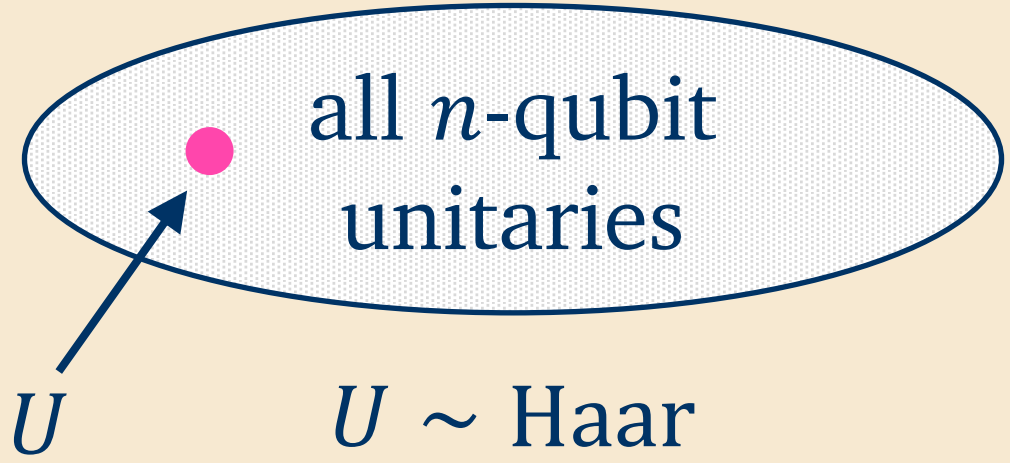
What exactly is a random unitary?

Haar measure: unique distribution on unitaries that is unchanged under **any** fixed unitary W

The quantum we'll need:

What exactly is a random unitary?

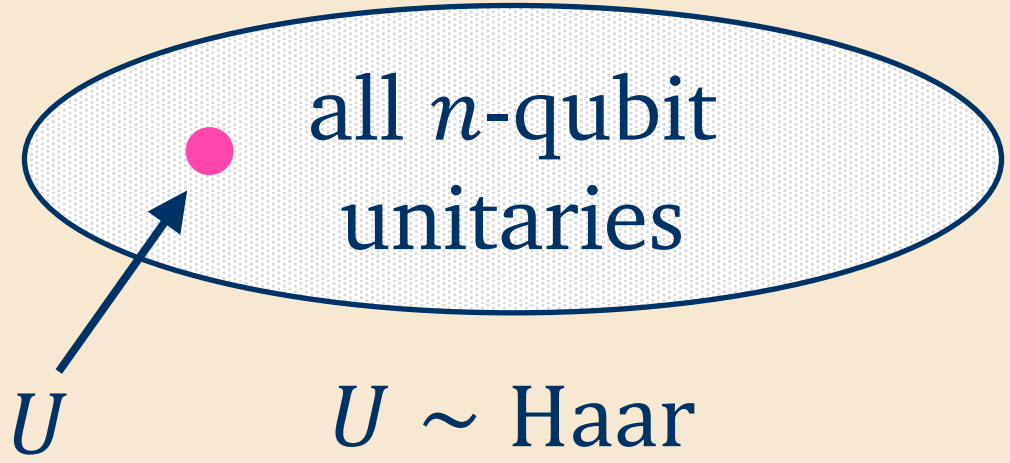
Haar measure: unique distribution on unitaries that is unchanged under **any** fixed unitary W

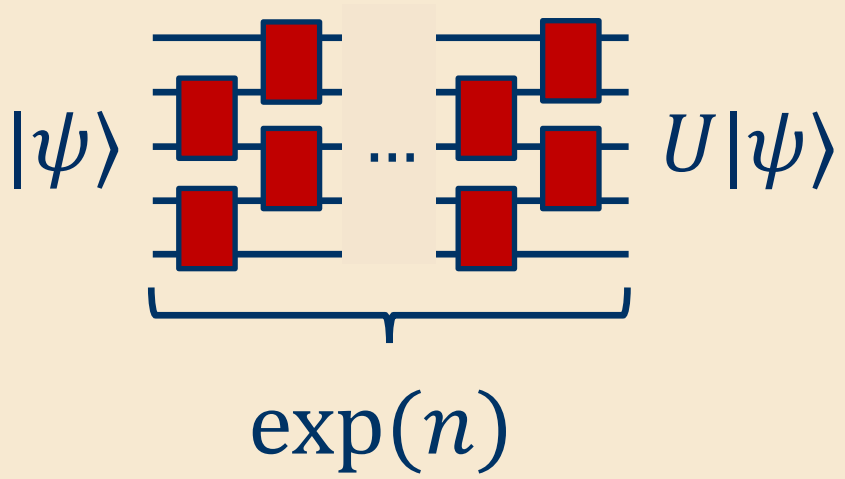


The quantum we'll need:

What exactly is a random unitary?

Haar measure: unique distribution on unitaries that is unchanged under **any** fixed unitary W





Haar-random unitaries come up everywhere in quantum:

Haar-random unitaries come up everywhere in quantum:

information scrambling

generate entanglement

quantum learning algorithms

...

quantum crypto

random quantum circuits

unitary complexity

quantum error correction

Haar-random unitaries come up everywhere in quantum:

information scrambling

generate entanglement

quantum learning algorithms

...

quantum crypto

random quantum circuits

unitary complexity

quantum error correction

But they're computationally infeasible to evaluate.

There's an analogous "problem" for functions: a random function on n bits is exponentially complex!

There's an analogous "problem" for functions: a random function on n bits is exponentially complex!

1	$f(1)$
2	$f(2)$
$\vdots$	$\vdots$
2^n	$f(2^n)$

random f

There's an analogous "problem" for functions: a random function on n bits is exponentially complex!

So in practice, we use **pseudorandom functions (PRFs).**

1	$f(1)$
2	$f(2)$
⋮	⋮
2^n	$f(2^n)$

random f

There's an analogous "problem" for functions: a random function on n bits is exponentially complex!

So in practice, we use **pseudorandom functions (PRFs).**

1	$f(1)$
2	$f(2)$
⋮	⋮
2^n	$f(2^n)$

random f

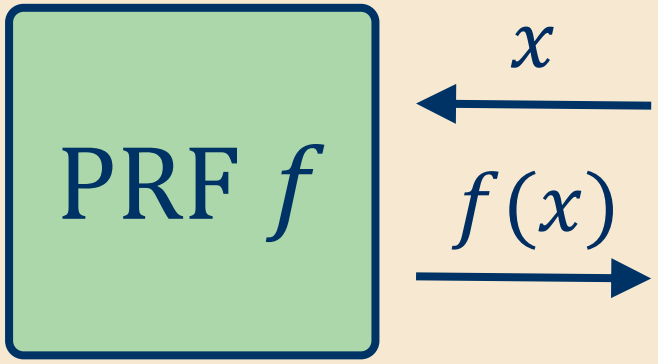
PRF f

There's an analogous "problem" for functions: a random function on n bits is exponentially complex!

So in practice, we use **pseudorandom functions (PRFs).**

1	$f(1)$
2	$f(2)$
$\vdots$	$\vdots$
2^n	$f(2^n)$

random f

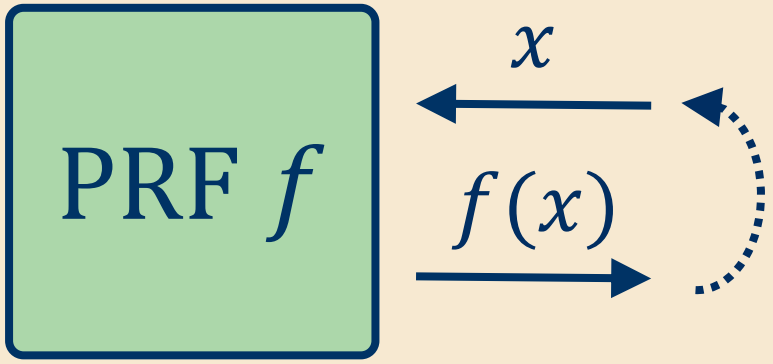


There's an analogous "problem" for functions: a random function on n bits is exponentially complex!

So in practice, we use **pseudorandom functions (PRFs).**

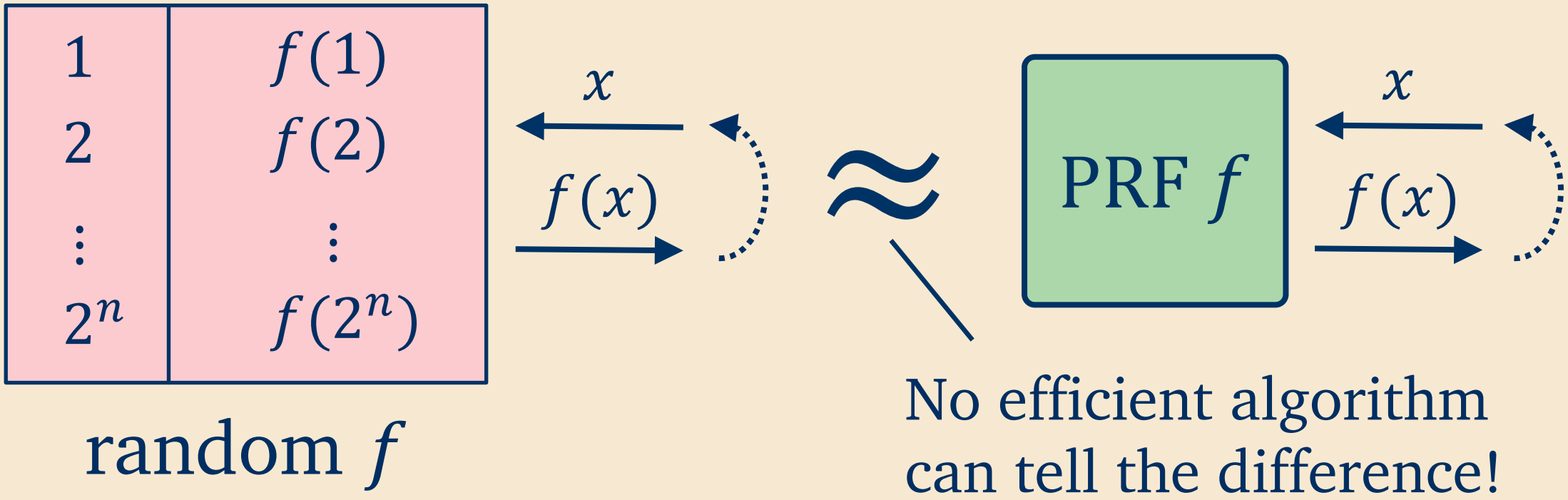
1	$f(1)$
2	$f(2)$
⋮	⋮
2^n	$f(2^n)$

random f



There's an analogous "problem" for functions: a random function on n bits is exponentially complex!

So in practice, we use **pseudorandom functions (PRFs).**

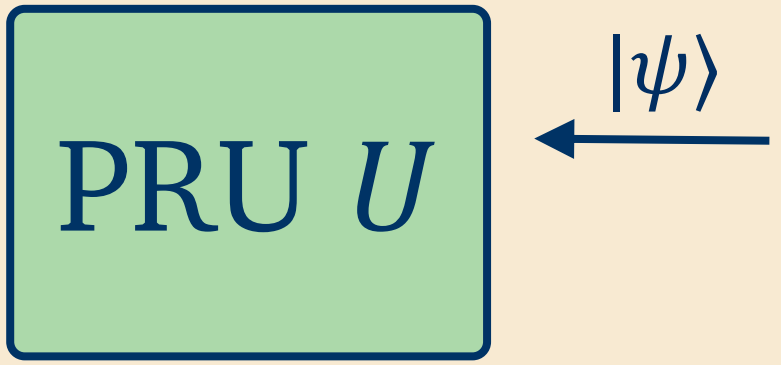


Pseudorandom unitaries (PRUs) [JLS18]

efficiently-computable unitaries that appear Haar-random

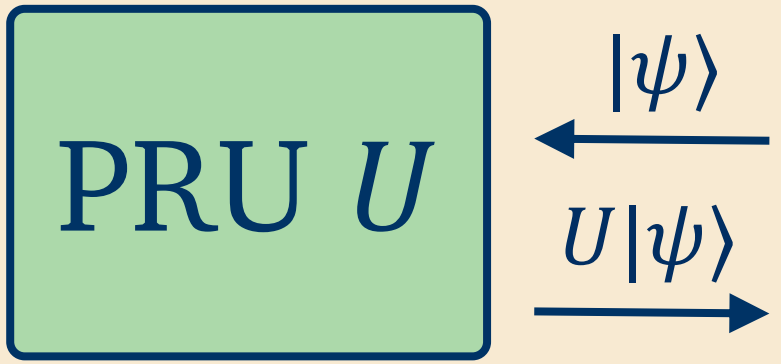
Pseudorandom unitaries (PRUs) [JLS18]

efficiently-computable unitaries that appear Haar-random



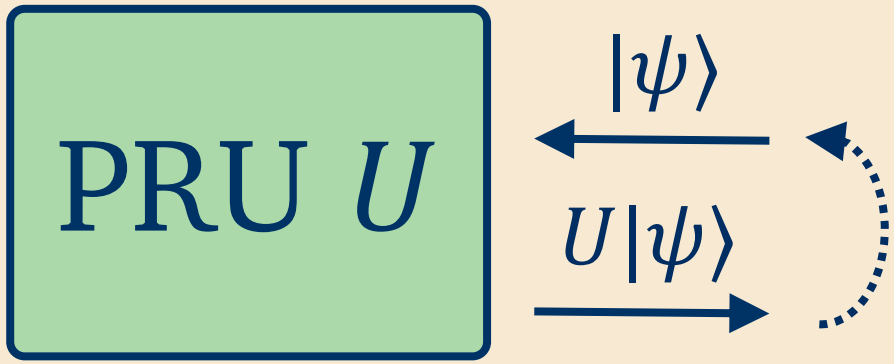
Pseudorandom unitaries (PRUs) [JLS18]

efficiently-computable unitaries that appear Haar-random



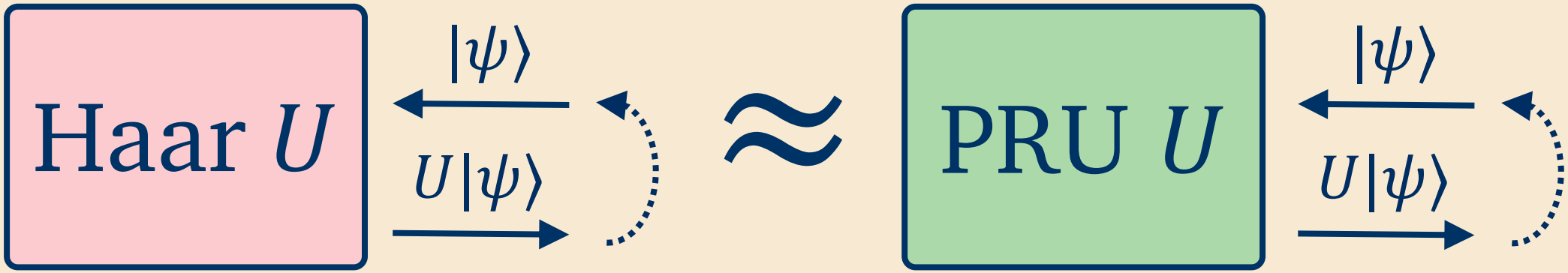
Pseudorandom unitaries (PRUs) [JLS18]

efficiently-computable unitaries that appear Haar-random



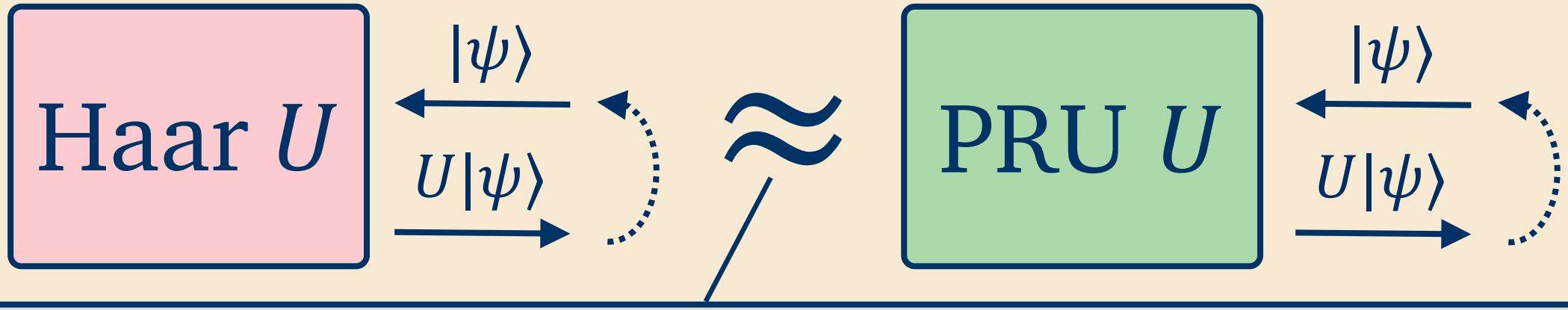
Pseudorandom unitaries (PRUs) [JLS18]

efficiently-computable unitaries that appear Haar-random



Pseudorandom unitaries (PRUs) [JLS18]

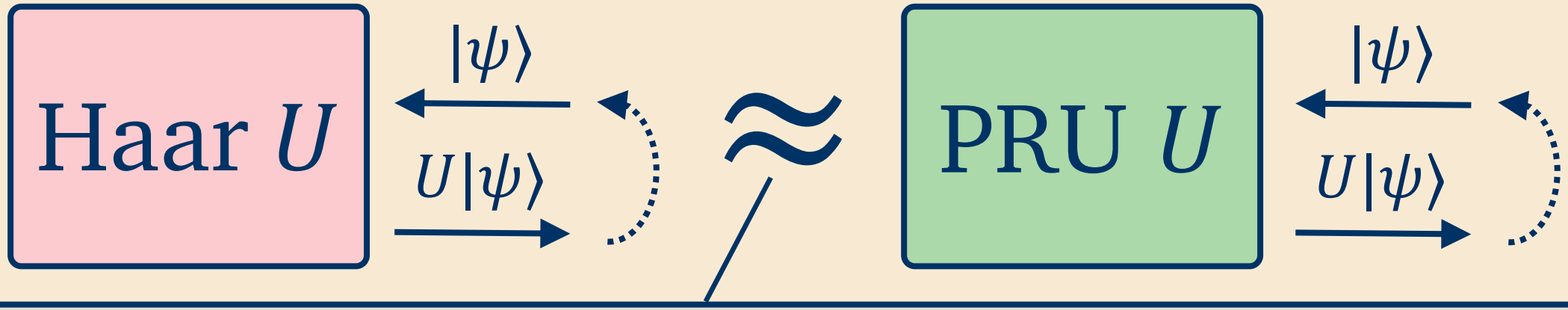
efficiently-computable unitaries that appear Haar-random



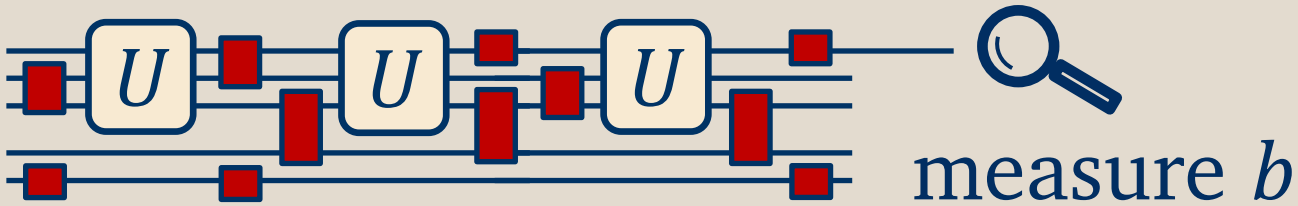
For any efficient algorithm A:

Pseudorandom unitaries (PRUs) [JLS18]

efficiently-computable unitaries that appear Haar-random

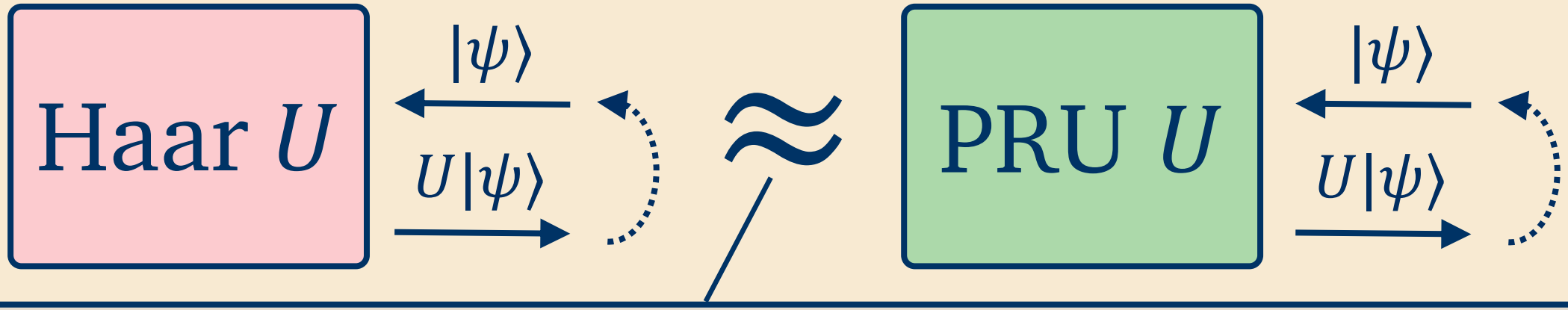


For any efficient algorithm A:

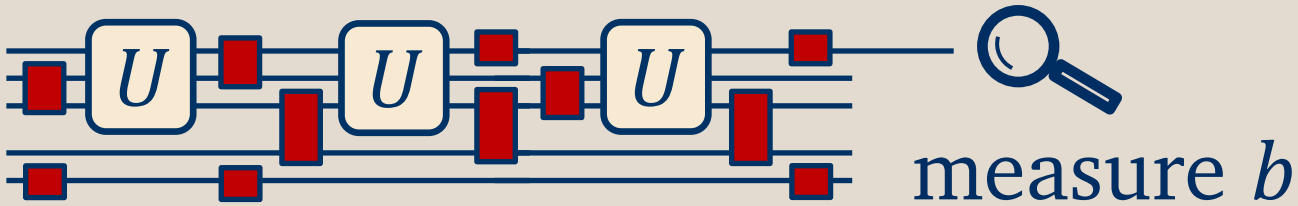


Pseudorandom unitaries (PRUs) [JLS18]

efficiently-computable unitaries that appear Haar-random



For any efficient algorithm A:



$$\Pr[b = 1 \mid U \leftarrow \text{Haar}] \approx \Pr[b = 1 \mid U \leftarrow \text{PRU}]$$

What was known about PRUs

What was known about PRUs

1) **Several candidate constructions** [JLS18,MPSY24,…]

What was known about PRUs

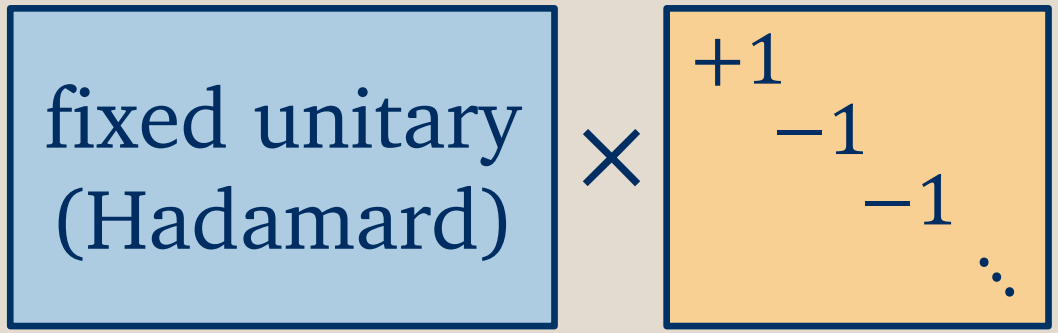
1) **Several candidate constructions** [JLS18,MPSY24,…]

[JLS18]
repeat many
times:

What was known about PRUs

1) **Several candidate constructions** [JLS18,MPSY24,…]

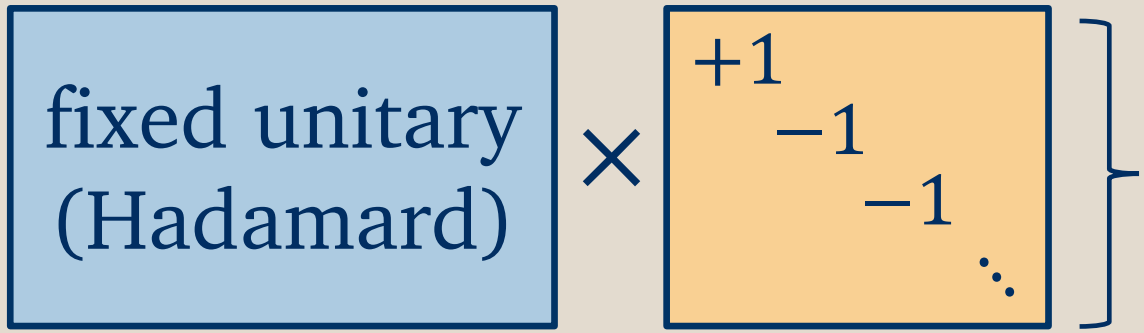
[JLS18]
repeat many
times:



What was known about PRUs

1) **Several candidate constructions** [JLS18,MPSY24,…]

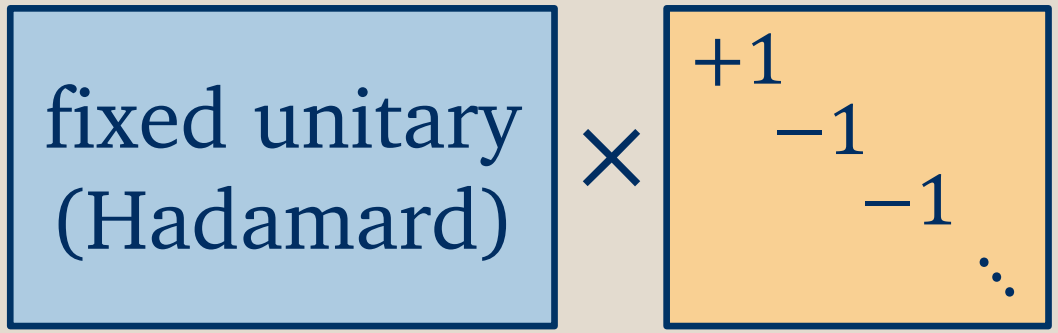
[JLS18] repeat many times:



What was known about PRUs

1) **Several candidate constructions** [JLS18,MPSY24,…]

[JLS18] repeat many times:

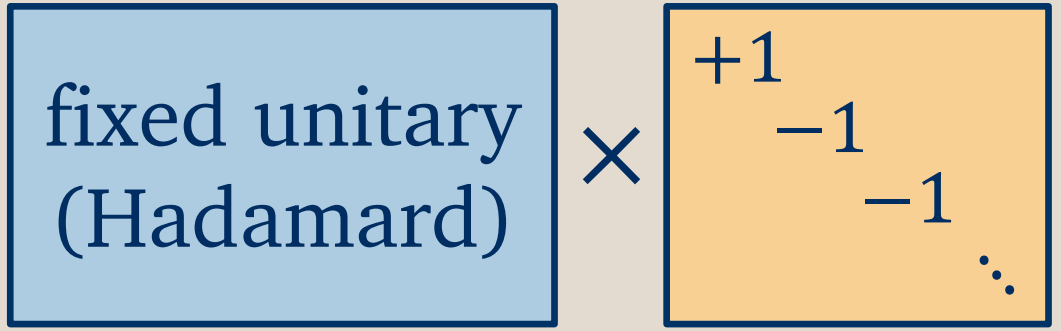


2) **Proofs of non-adaptive security** [MPSY24, CBBDHX24]

What was known about PRUs

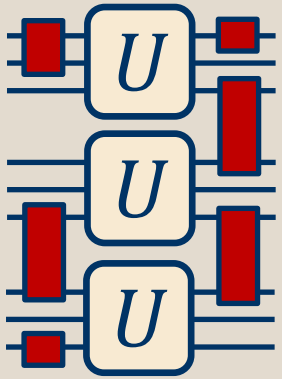
1) **Several candidate constructions** [JLS18,MPSY24,…]

[JLS18] repeat many times:



2) **Proofs of non-adaptive security** [MPSY24, CBBDHX24]

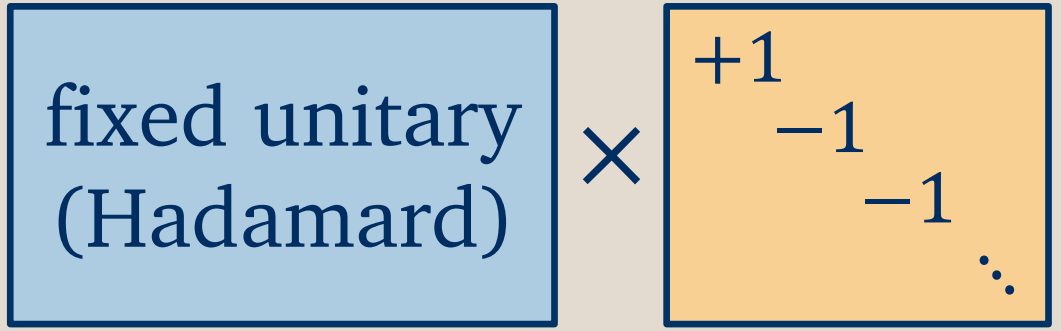
can analyze this:



What was known about PRUs

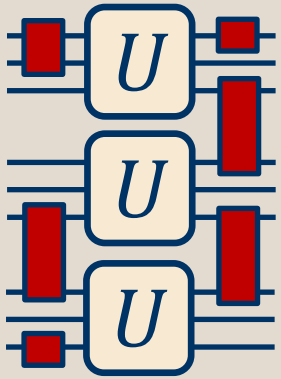
1) **Several candidate constructions** [JLS18,MPSY24,…]

[JLS18] repeat many times:

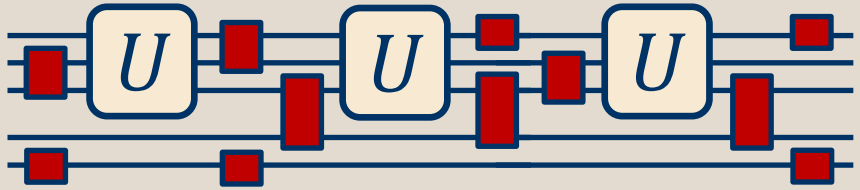


2) **Proofs of non-adaptive security** [MPSY24, CBBDHX24]

can analyze this:



but not this:



In [MH24], we resolve this question.

In [MH24], we resolve this question.

Theorem:
PRUs exist under cryptographic assumptions.

In [MH24], we resolve this question.

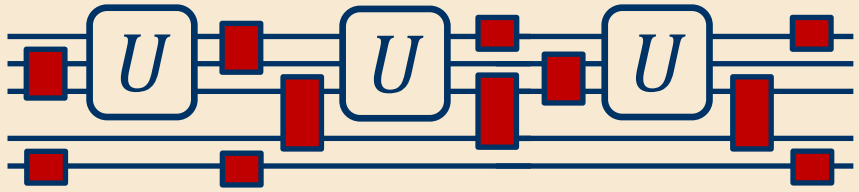
Theorem:
PRUs exist under cryptographic assumptions.

(the construction we analyze is by [MPSY24])

Why has it been hard to prove PRUs secure?

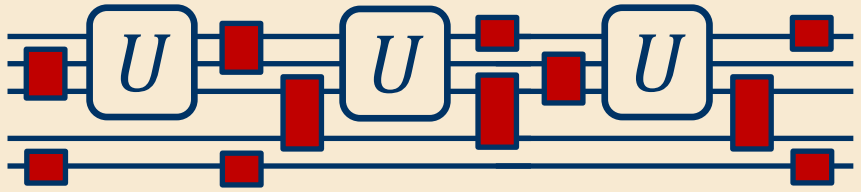
Why has it been hard to prove PRUs secure?

1) Hard to characterize behavior of an arbitrary algorithm:



Why has it been hard to prove PRUs secure?

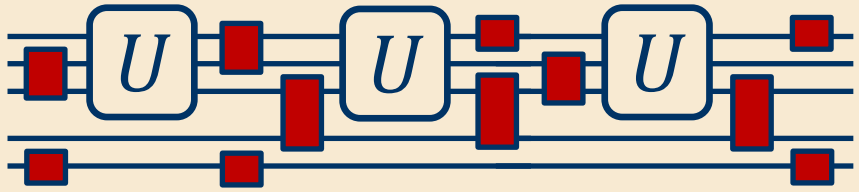
1) Hard to characterize behavior of an arbitrary algorithm:



2) Mathematics of random unitaries is complicated.

Why has it been hard to prove PRUs secure?

1) Hard to characterize behavior of an arbitrary algorithm:



2) Mathematics of random unitaries is complicated.

- Weingarten calculus
- representation theory
- free probability

Theorem 3.1. *Let* k *be a positive integer. For any permutation* $\sigma \in \mathcal{S}_k$ *and nonnegative integer* g*, we have*

$$(k-1)^g \#P(\sigma,|\sigma|) \leq \#P(\sigma,|\sigma|+2g) \leq (6k^{7/2})^g \#P(\sigma,|\sigma|).$$

Theorem 3.2. *For any* $\sigma \in \mathcal{S}_k$ *and* $d > \sqrt{6}k^{7/4}$*,*

$$\frac{1}{1-\frac{k-1}{d^2}} \leq \frac{(-1)^{|\sigma|} d^{k+|\sigma|}\, \mathrm{Wg}^U(\sigma,d)}{\#P(\sigma,|\sigma|)} \leq \frac{1}{1-\frac{6k^{7/2}}{d^2}}.$$

In addition, the l.h.s inequality is valid for any $d \geq k$.

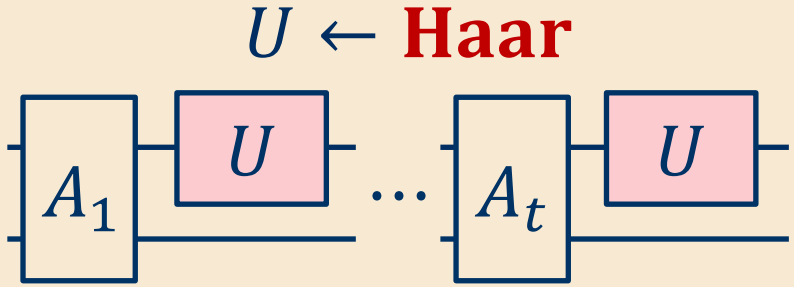
Cartoon overview of our proof

Cartoon overview of our proof

Want to show:
For all efficient adversaries A,

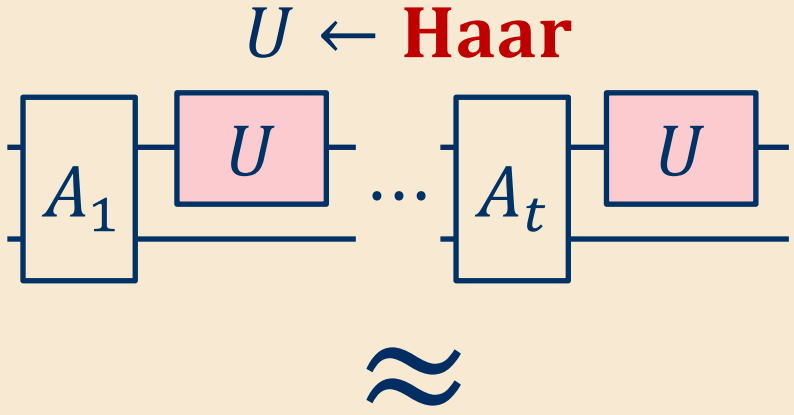
Cartoon overview of our proof

Want to show:
For all efficient adversaries A,



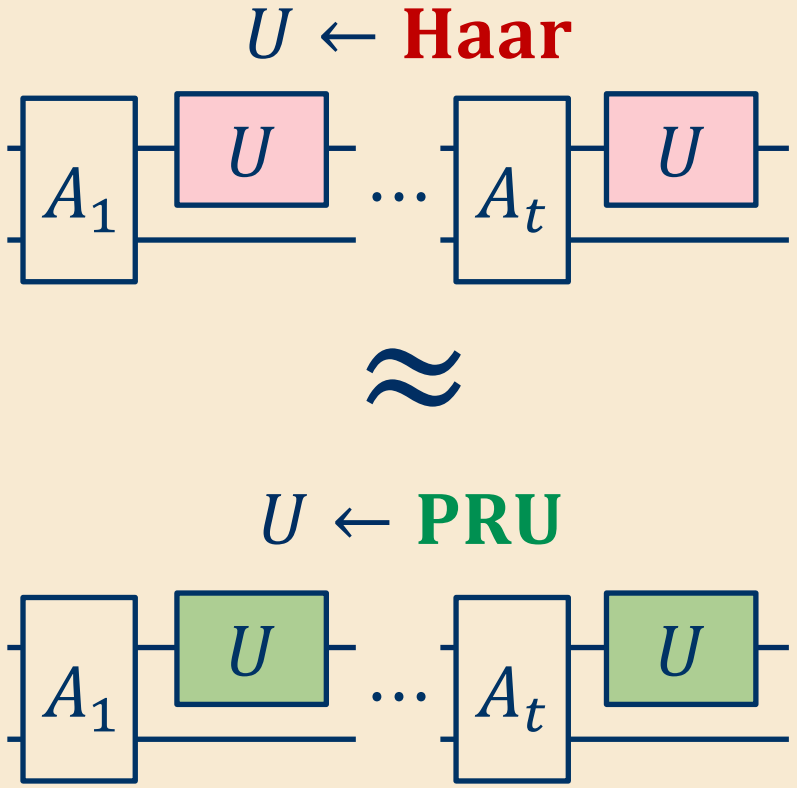
Cartoon overview of our proof

Want to show:
For all efficient adversaries A,



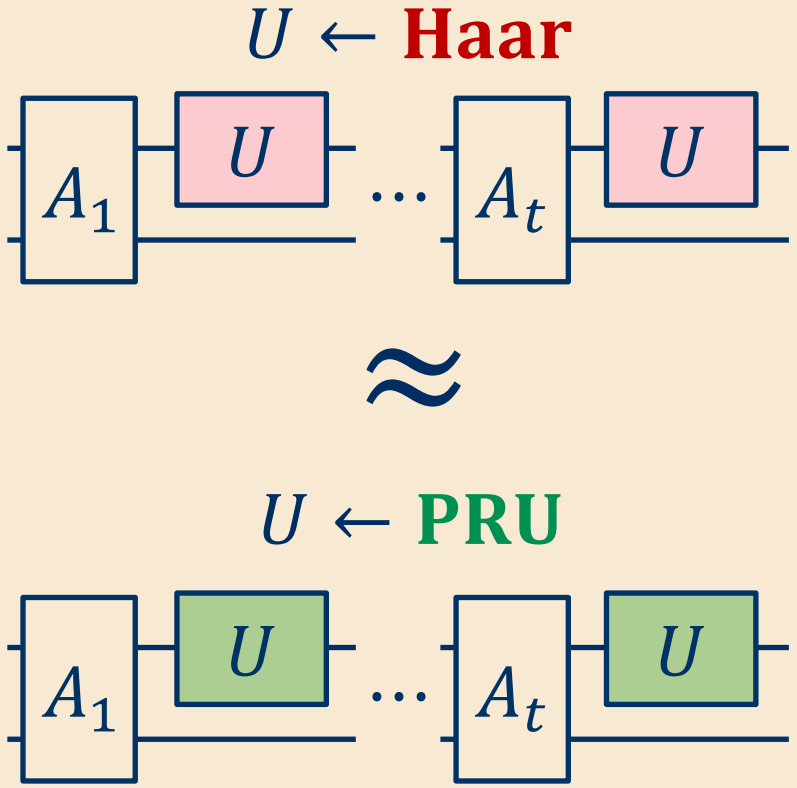
Cartoon overview of our proof

Want to show:
For all efficient adversaries A,



Cartoon overview of our proof

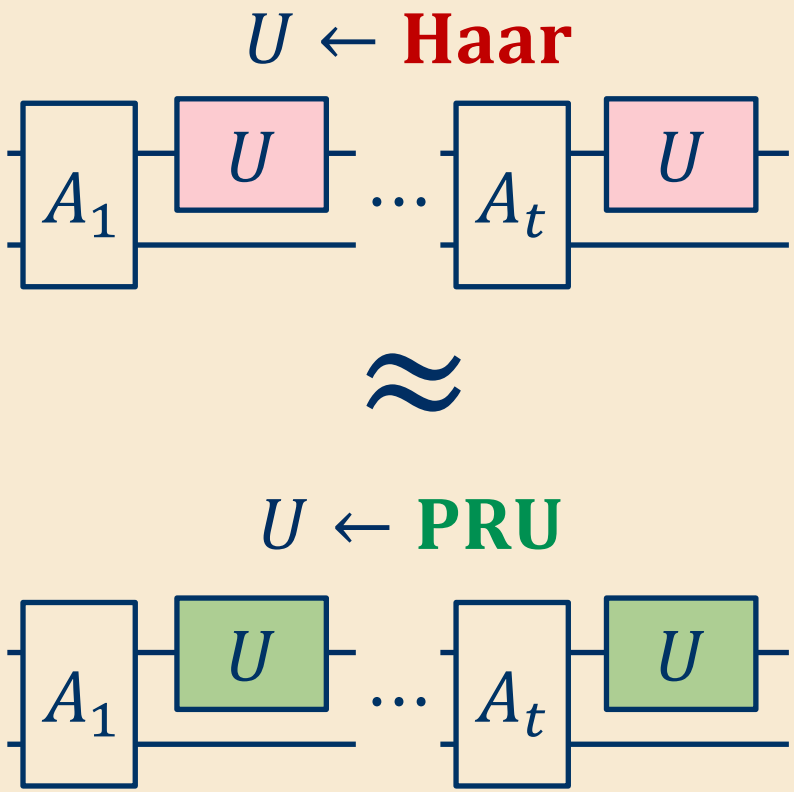
Want to show:
For all efficient adversaries A,



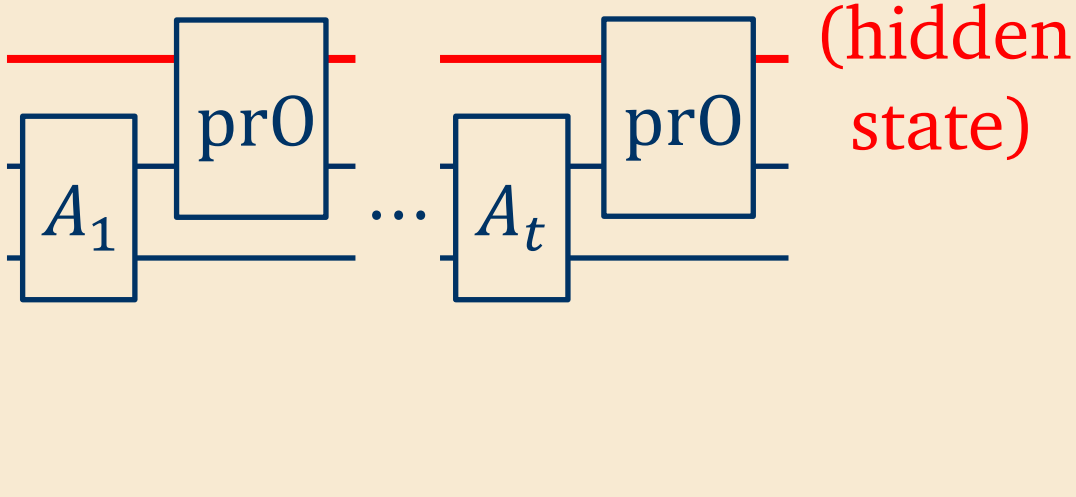
Proof strategy: show that both are indistinguishable from

Cartoon overview of our proof

Want to show:
For all efficient adversaries A,

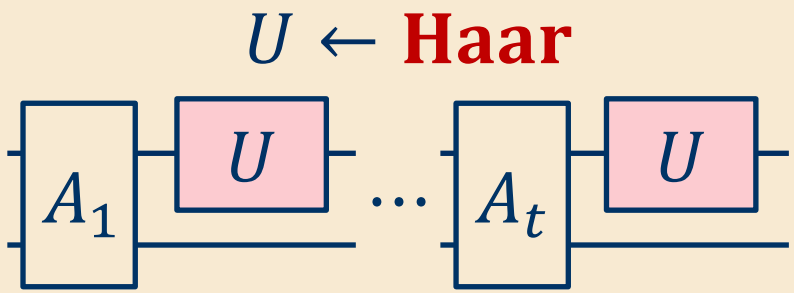


Proof strategy: show that both are indistinguishable from

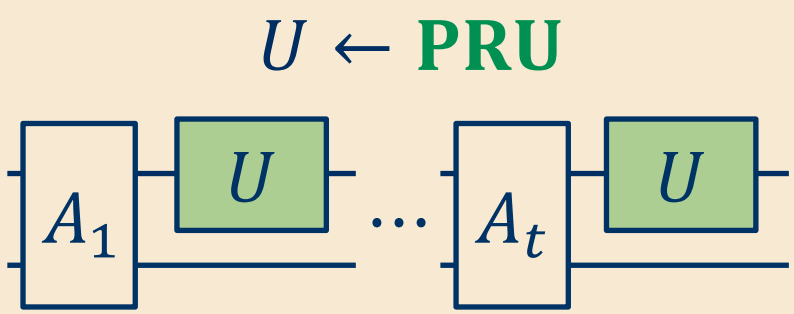


Cartoon overview of our proof

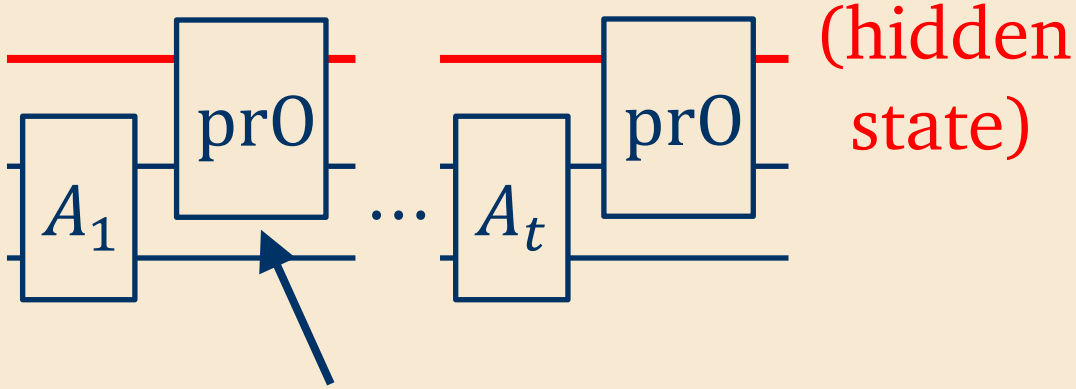
Want to show:
For all efficient adversaries A,



$\approx$



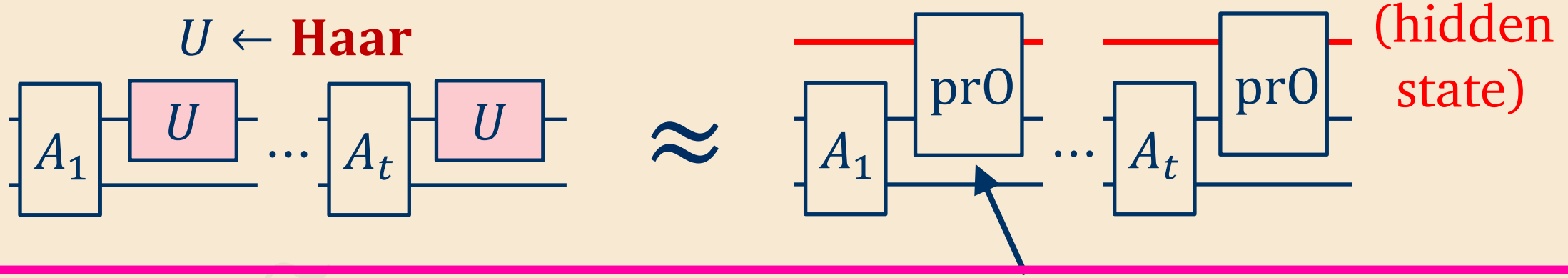
Proof strategy: show that both are indistinguishable from



$\approx$

(path-recording oracle)
a data structure that performs "lazy sampling" of a Haar-random unitary

most of the proof



(path-recording oracle)
a data structure that performs "lazy sampling" of a Haar-random unitary

Lazy sampling of a random **function**

Lazy sampling of a random **function**

Goal: efficiently implement an algorithm that queries a random **function** f.

Lazy sampling of a random **function**

Goal: efficiently implement an algorithm that queries a random **function** f.

Solution:
- only sample $f(x)$ when needed, "on the fly"

Lazy sampling of a random **function**

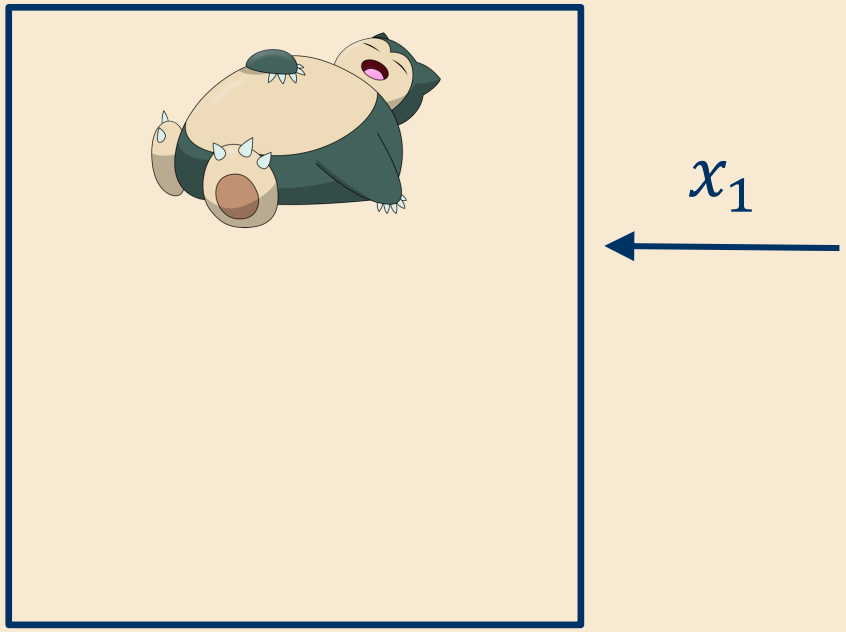
Goal: efficiently implement an algorithm that queries a random **function** f.

Solution:
- only sample $f(x)$ when needed, "on the fly"
- remember what you sampled (for consistency)

Lazy sampling of a random **function**

Goal: efficiently implement an algorithm that queries a random **function** f.

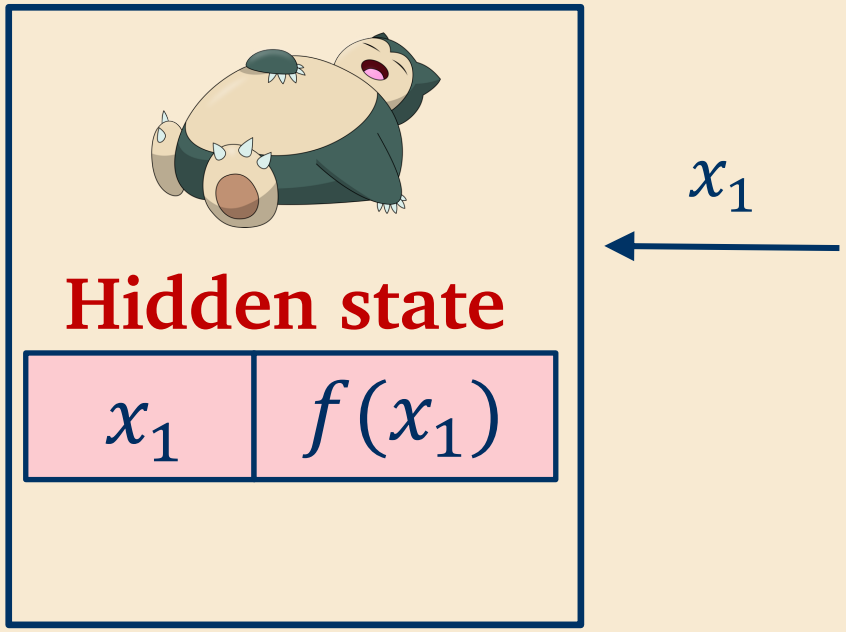
Solution:
- only sample $f(x)$ when needed, "on the fly"
- remember what you sampled (for consistency)



Lazy sampling of a random **function**

Goal: efficiently implement an algorithm that queries a random **function** f.

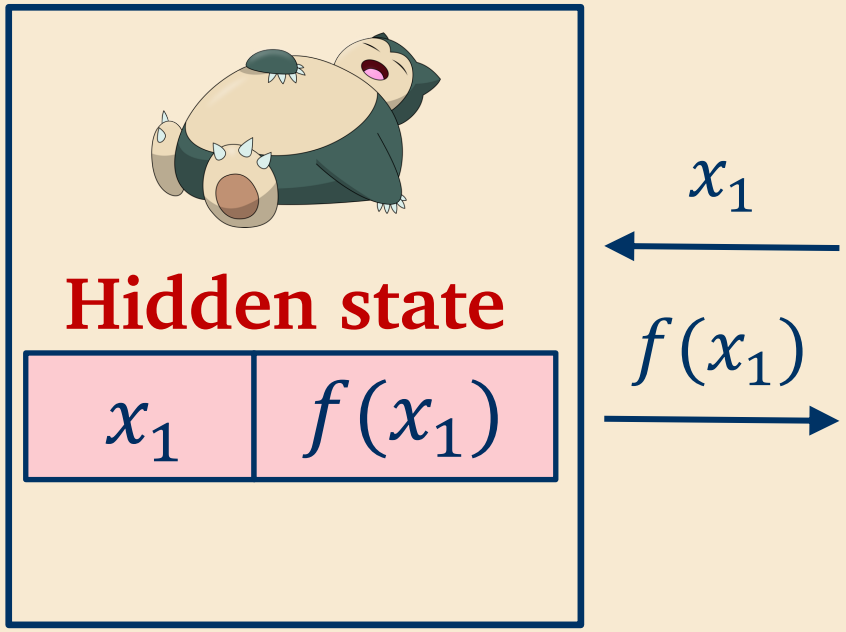
Solution:
- only sample $f(x)$ when needed, "on the fly"
- remember what you sampled (for consistency)



Lazy sampling of a random **function**

Goal: efficiently implement an algorithm that queries a random **function** f.

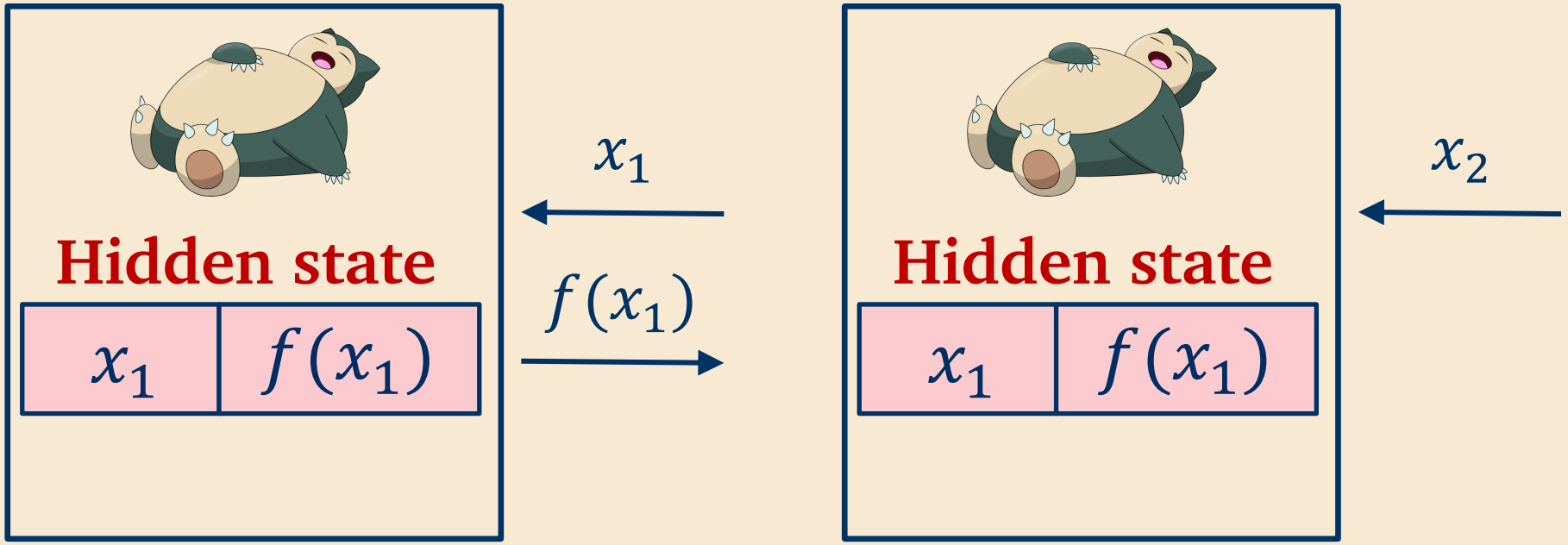
Solution:
- only sample $f(x)$ when needed, "on the fly"
- remember what you sampled (for consistency)



Lazy sampling of a random **function**

Goal: efficiently implement an algorithm that queries a random **function** f.

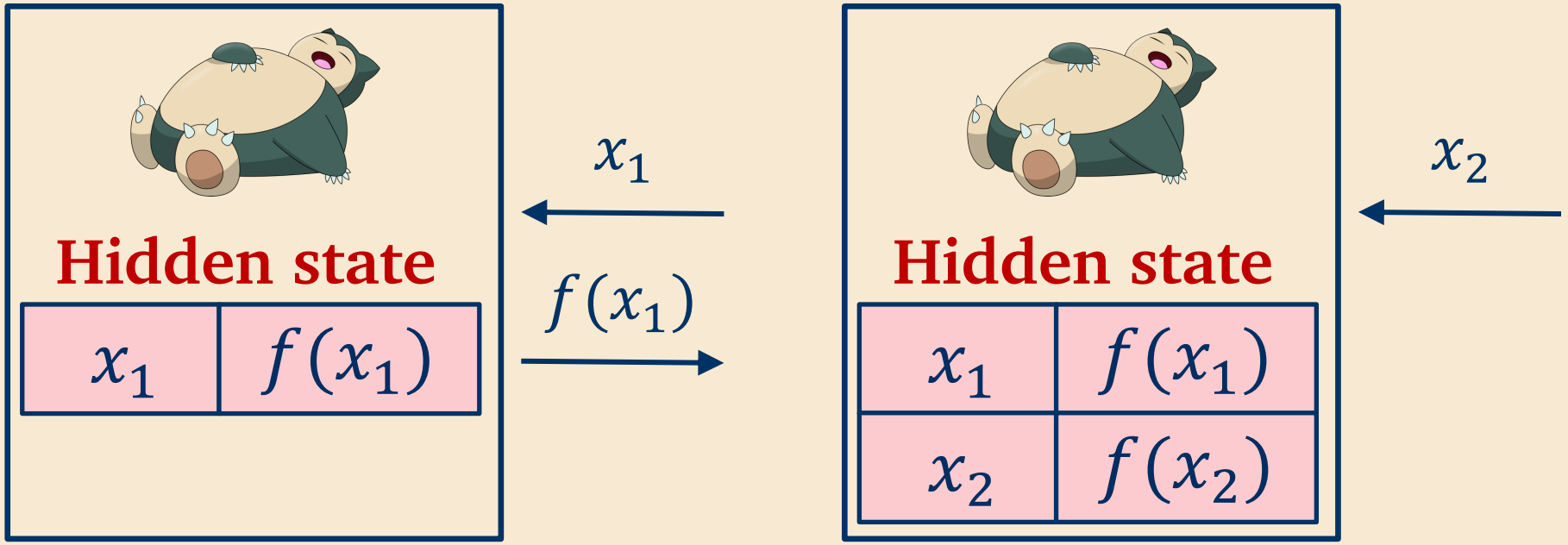
Solution:
- only sample $f(x)$ when needed, "on the fly"
- remember what you sampled (for consistency)



Lazy sampling of a random **function**

Goal: efficiently implement an algorithm that queries a random **function** f.

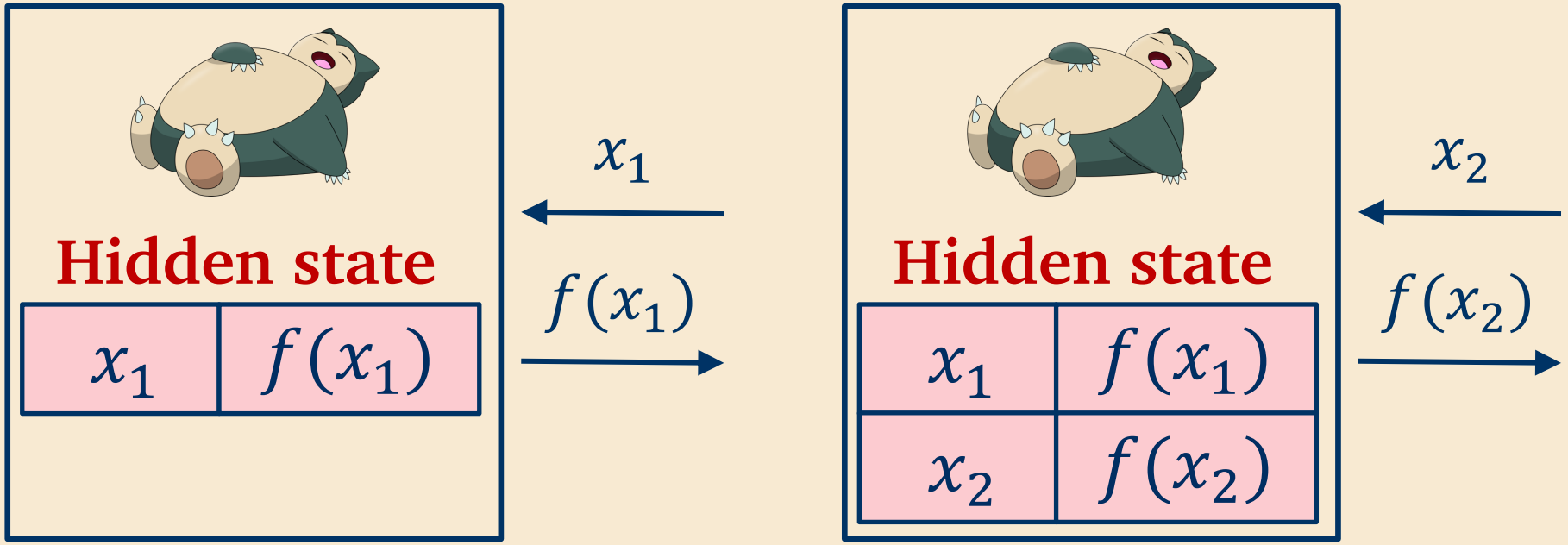
Solution:
- only sample $f(x)$ when needed, "on the fly"
- remember what you sampled (for consistency)



Lazy sampling of a random **function**

Goal: efficiently implement an algorithm that queries a random **function** f.

Solution:
- only sample $f(x)$ when needed, "on the fly"
- remember what you sampled (for consistency)



Lazy sampling of a random **unitary**

Lazy sampling of a random **unitary**

Goal: efficiently implement a quantum algorithm that queries a Haar-random unitary U.

Lazy sampling of a random **unitary**

Goal: efficiently implement a quantum algorithm that queries a Haar-random unitary U.

A priori, not clear how to do this!

Lazy sampling of a random **unitary**

Goal: efficiently implement a quantum algorithm that queries a Haar-random unitary U.

A priori, not clear how to do this!

Our solution: the path-recording oracle

Lazy sampling of a random **unitary**

Goal: efficiently implement a quantum algorithm that queries a Haar-random unitary U.

A priori, not clear how to do this!

Our solution: the path-recording oracle

We use **entanglement** with a **hidden data structure** that succinctly "remembers" enough information to spoof a Haar-random U.

Lazy sampling of a random **unitary**

Goal: efficiently implement a quantum algorithm that queries a Haar-random unitary U.

A priori, not clear how to do this!

Our solution: the path-recording oracle

We use **entanglement** with a **hidden data structure** that succinctly "remembers" enough information to spoof a Haar-random U.

Classical case: the data structure is the list of $(x, f(x))$ pairs.

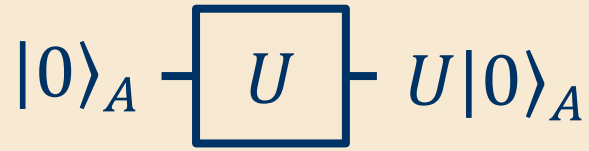
Up next:

"Derive" the path-recording oracle through simple examples

Example 1: one query on $|0\rangle$

Example 1: one query on $|0\rangle$

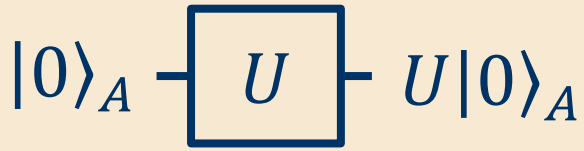
The algorithm:



$(U \leftarrow \text{Haar})$

Example 1: one query on $|0\rangle$

The algorithm:

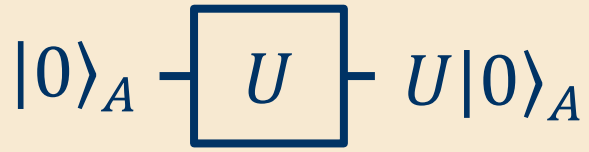


Fact: this is the "maximally mixed" state

$(U \leftarrow \text{Haar})$

Example 1: one query on $|0\rangle$

The algorithm:



Fact: this is the "maximally mixed" state

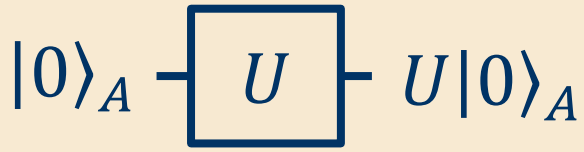
$(U \leftarrow \text{Haar})$

How to "spoof" it:

$|0\rangle_A$ $\qquad$ $\sum_y |y\rangle_A |y\rangle_S$ $\qquad$ **(S register is hidden)**

Example 1: one query on $|0\rangle$

The algorithm:



Fact: this is the "maximally mixed" state

$(U \leftarrow \text{Haar})$

How to "spoof" it:

$|0\rangle_A$

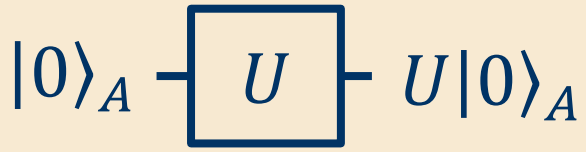
Fact: this is ***also*** the maximally mixed state.

$$\sum_y |y\rangle_A |y\rangle_S$$

(S register is hidden)

Example 1: one query on $|0\rangle$

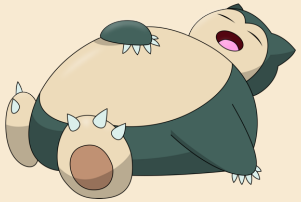
The algorithm:



Fact: this is the "maximally mixed" state

$(U \leftarrow \text{Haar})$

How to "spoof" it:



Fact: this is ***also*** the maximally mixed state.

$$\sum_y |y\rangle_A |y\rangle_S$$

(S register is hidden)

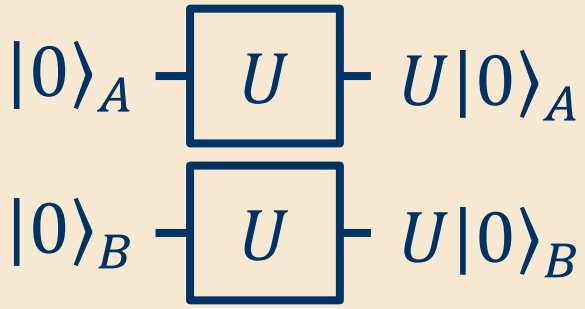
Idea 1: entanglement with a hidden register S can simulate one query to U.

Example 2: two queries on $|0\rangle$

Example 2: two queries on $|0\rangle$

The algorithm:

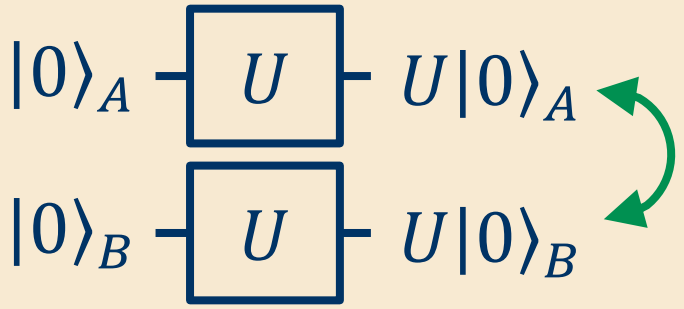
$(U \leftarrow \text{Haar})$



Example 2: two queries on $|0\rangle$

The algorithm:

$(U \leftarrow \text{Haar})$

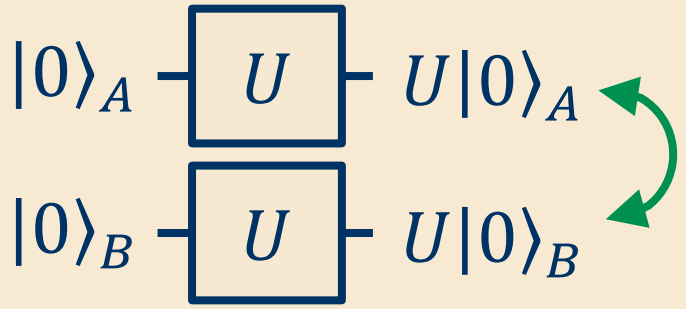


Fact: this is the maximally mixed "symmetric" state

Example 2: two queries on $|0\rangle$

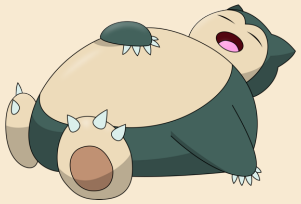
The algorithm:

$(U \leftarrow \text{Haar})$



Fact: this is the maximally mixed "symmetric" state

How to "spoof" it:



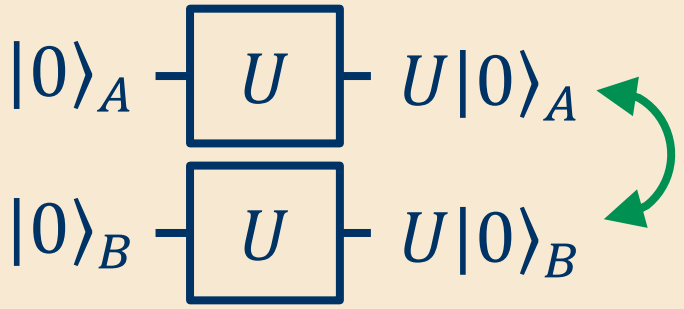
$$\sum_{y_1,y_2} |y_1\rangle_A |y_2\rangle_B |\{y_1, y_2\}\rangle_S$$

(S register is hidden)

Example 2: two queries on $|0\rangle$

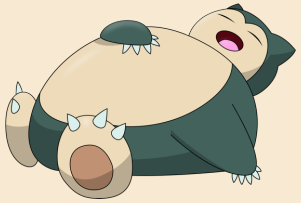
The algorithm:

$(U \leftarrow \text{Haar})$



Fact: this is the maximally mixed "symmetric" state

How to "spoof" it:



also symmetric!

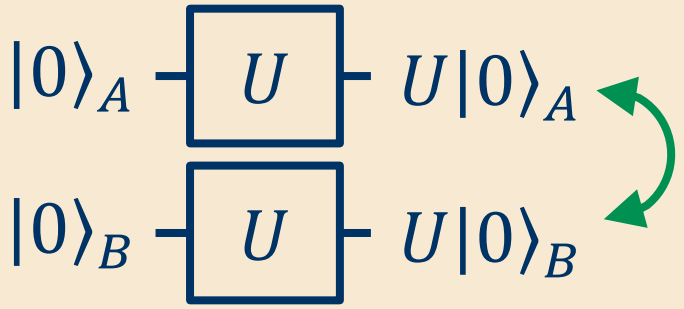
$$\sum_{y_1,y_2} |y_1\rangle_A |y_2\rangle_B |\{y_1, y_2\}\rangle_S$$

(S register is hidden)

Example 2: two queries on $|0\rangle$

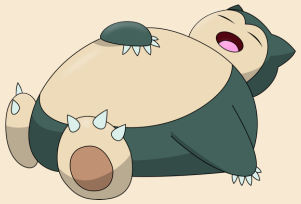
The algorithm:

$(U \leftarrow \text{Haar})$



Fact: this is the maximally mixed "symmetric" state

How to "spoof" it:



also symmetric!

$$\sum_{y_1,y_2} |y_1\rangle_A |y_2\rangle_B |\{y_1, y_2\}\rangle_S$$

(S register is hidden)

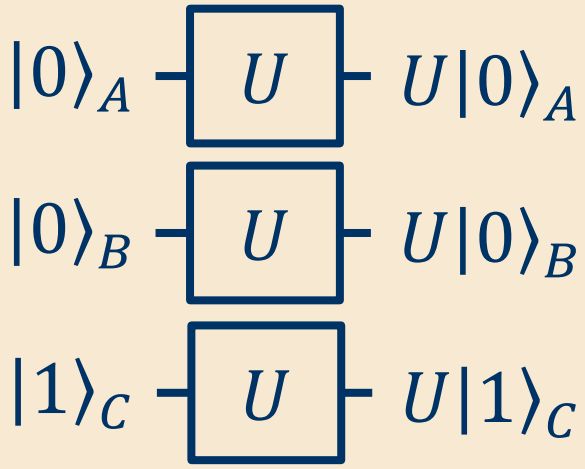
Idea 2: use an **unordered set** to spoof "swap-symmetry".

Example 3: mixed queries

Example 3: mixed queries

The algorithm:

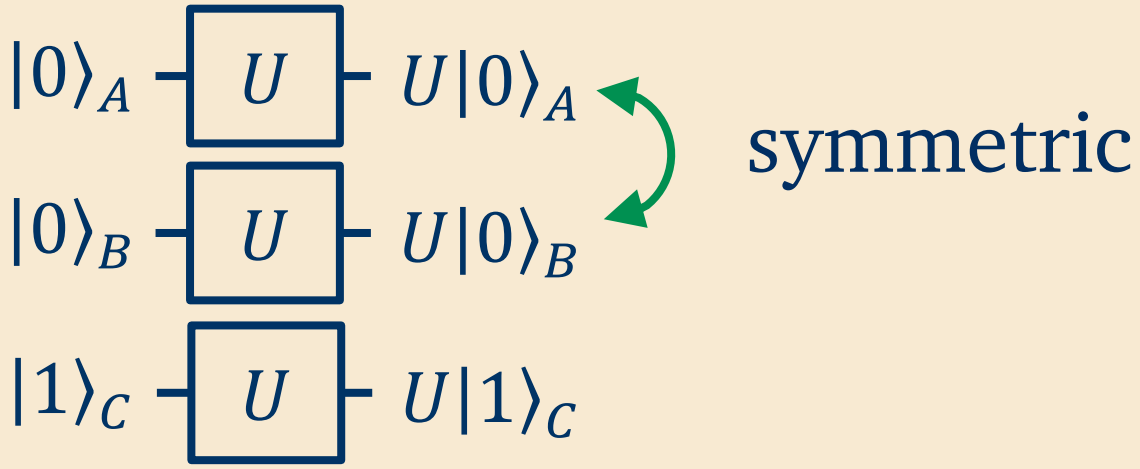
$(U \leftarrow \text{Haar})$



Example 3: mixed queries

The algorithm:

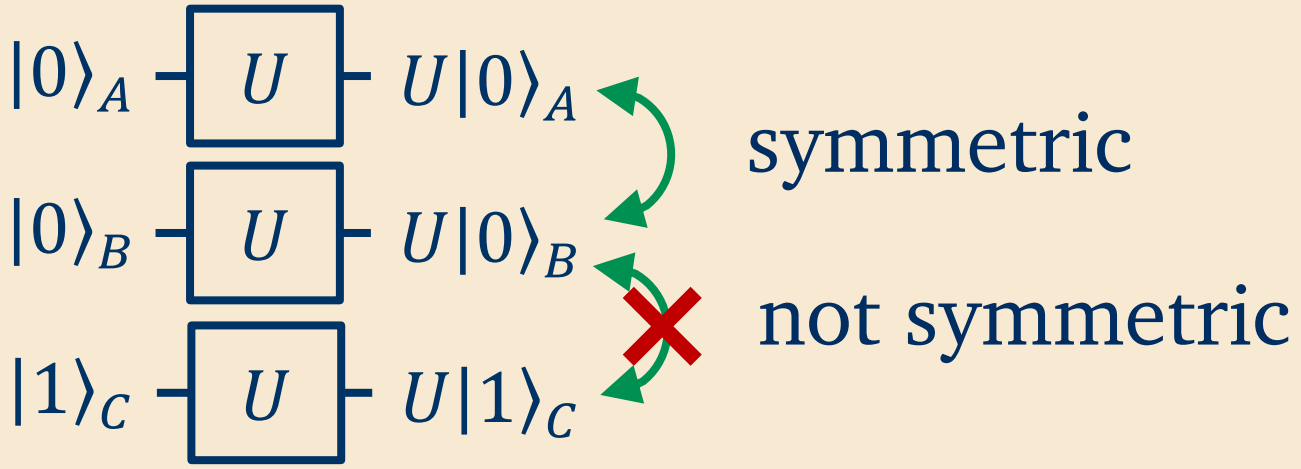
$(U \leftarrow \text{Haar})$



Example 3: mixed queries

The algorithm:

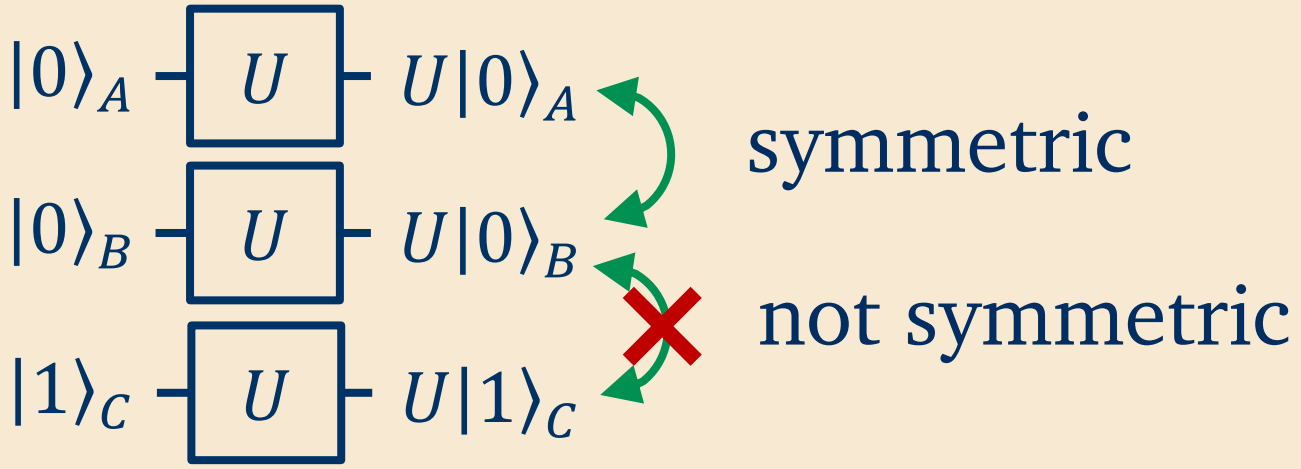
$(U \leftarrow \text{Haar})$



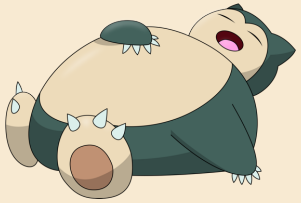
Example 3: mixed queries

The algorithm:

$(U \leftarrow \text{Haar})$



How to "spoof" it:

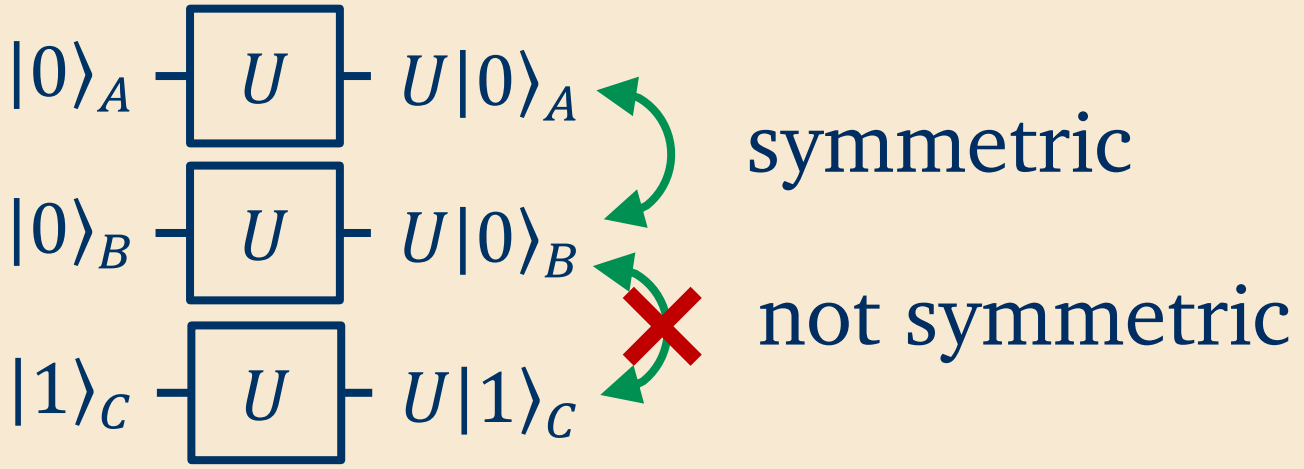


$$\sum_{y_1,y_2,y_3} |y_1\rangle_A |y_2\rangle_B |y_3\rangle_C |\{(0,y_1),(0,y_2),(1,y_3)\}\rangle_S$$

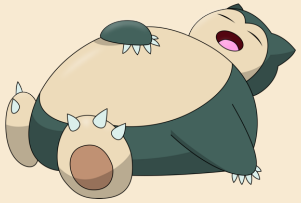
Example 3: mixed queries

The algorithm:

$(U \leftarrow \text{Haar})$



How to "spoof" it:

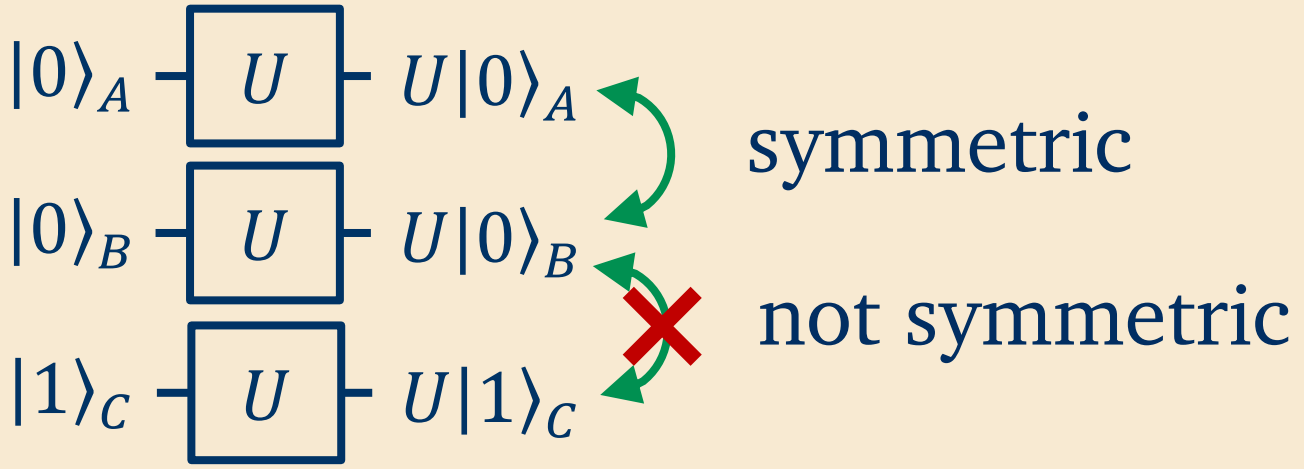


$$\sum_{y_1,y_2,y_3} |y_1\rangle_A |y_2\rangle_B |y_3\rangle_C |\{(0,y_1),(0,y_2),(1,y_3)\}\rangle_S$$

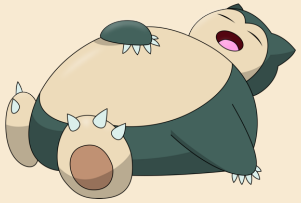
Example 3: mixed queries

The algorithm:

$(U \leftarrow \text{Haar})$



How to "spoof" it:

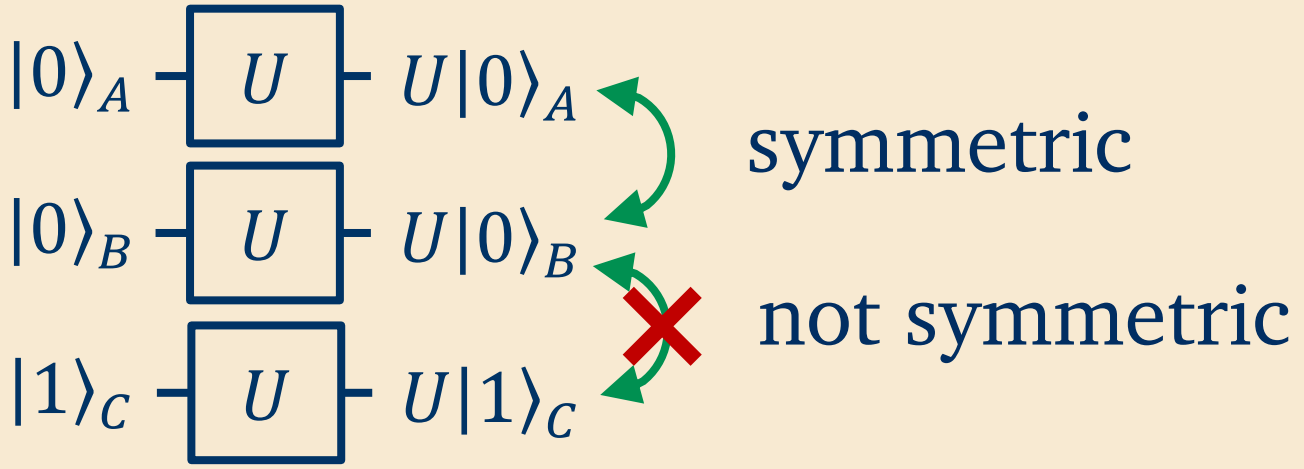


$$\sum_{y_1,y_2,y_3} |y_1\rangle_A |y_2\rangle_B |y_3\rangle_C |\{(0,y_1),(0,y_2),(1,y_3)\}\rangle_S$$

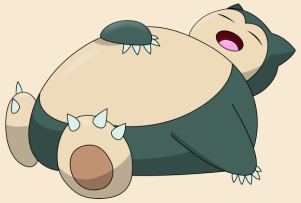
Example 3: mixed queries

The algorithm:

$(U \leftarrow \text{Haar})$



How to "spoof" it:

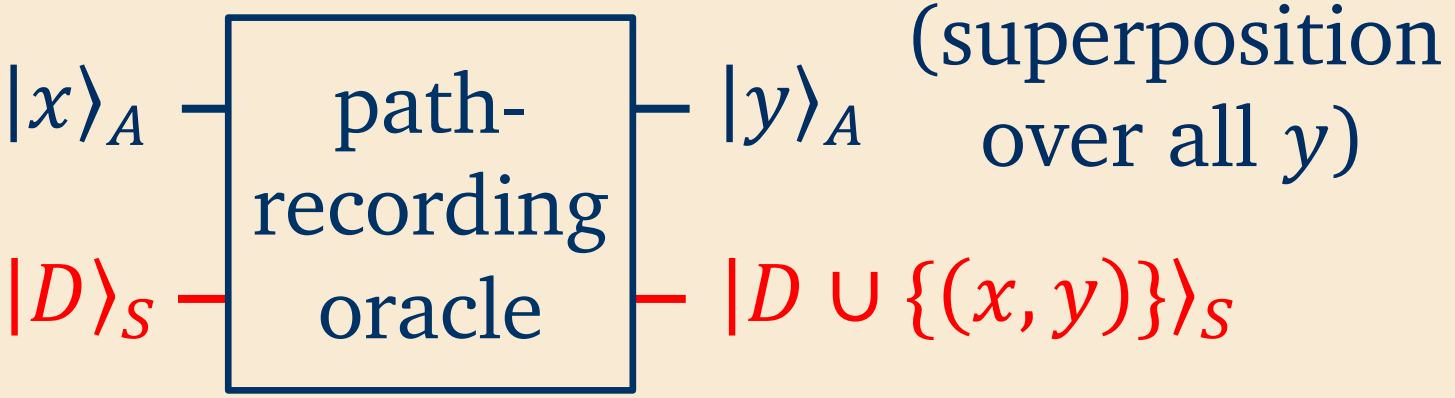


$$\sum_{y_1,y_2,y_3} |y_1\rangle_A |y_2\rangle_B |y_3\rangle_C |\{(0,y_1),(0,y_2),(1,y_3)\}\rangle_S$$

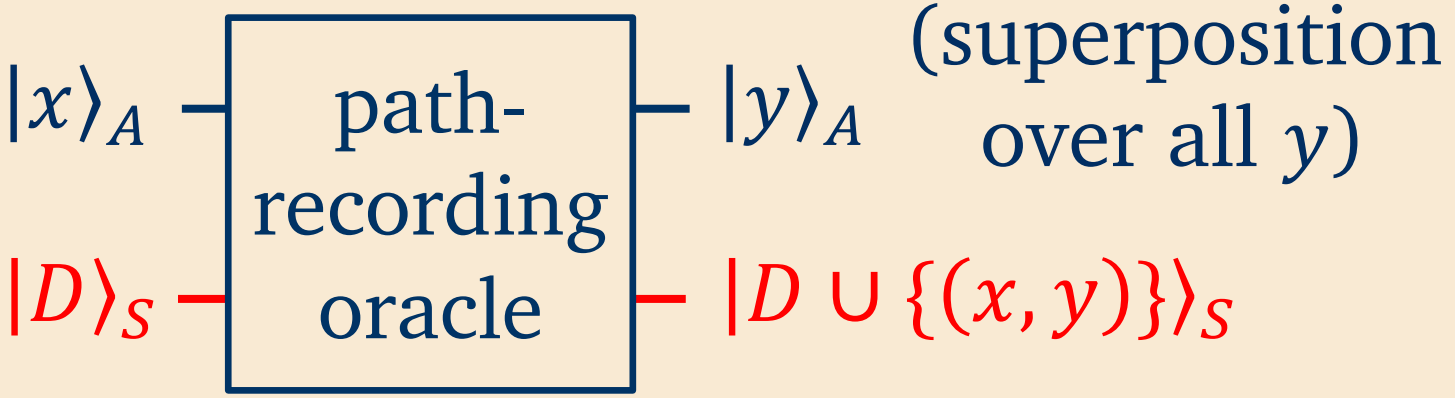
Idea 3: use **ordered pairs** to simulate symmetry "structure"

We can generate all of these examples by simply replacing each query to U with this:

We can generate all of these examples by simply replacing each query to U with this:

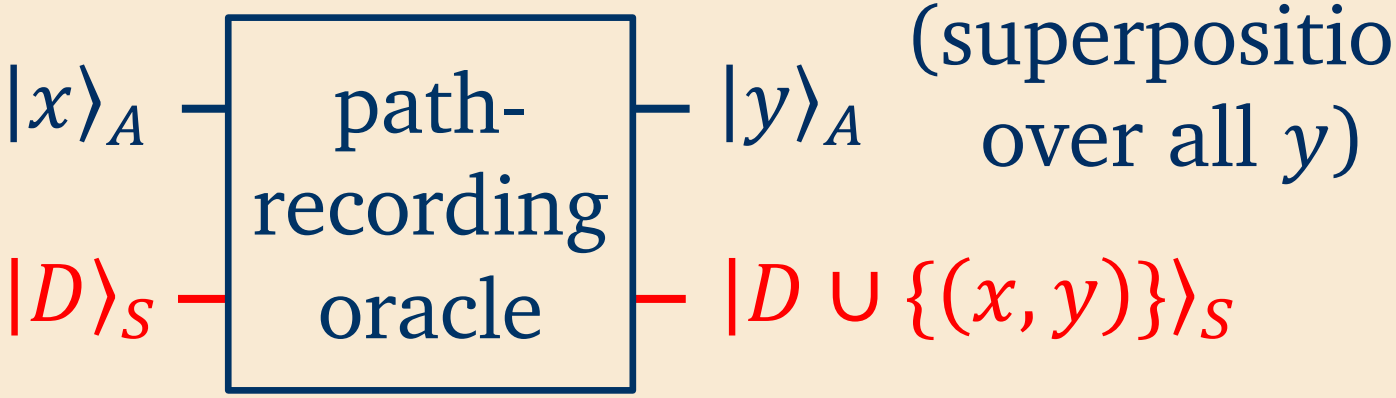


We can generate all of these examples by simply replacing each query to U with this:



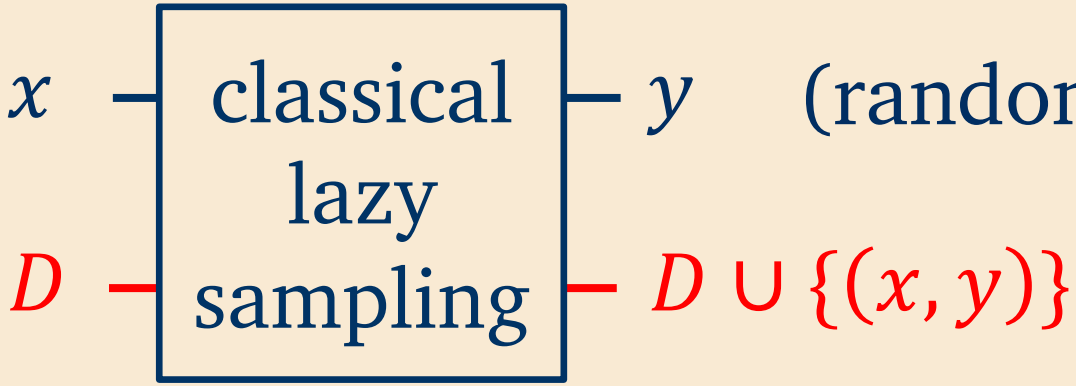
Note the similarity to classical lazy sampling:

We can generate all of these examples by simply replacing each query to U with this:



(superposition over all y)

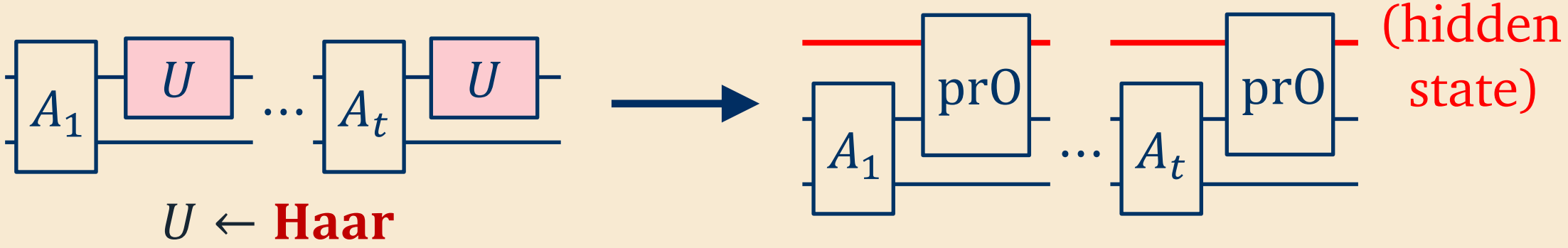
Note the similarity to classical lazy sampling:



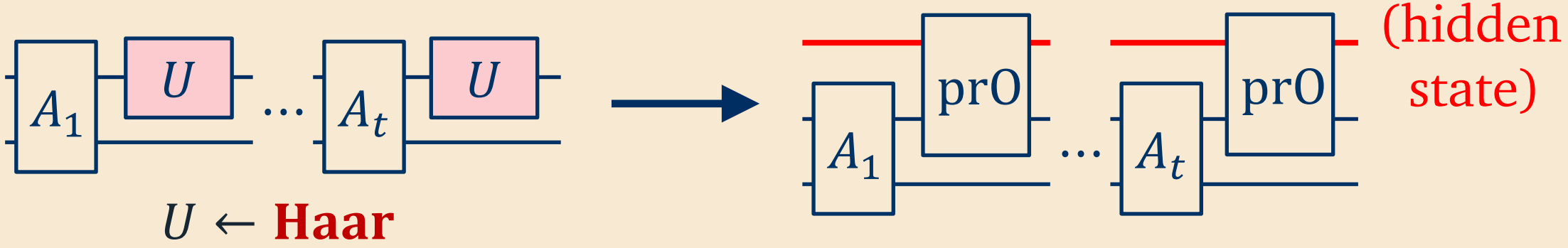
(random y)

The path-recording oracle is actually a general-purpose tool for analyzing Haar-random unitaries.

The path-recording oracle is actually a general-purpose tool for analyzing Haar-random unitaries.

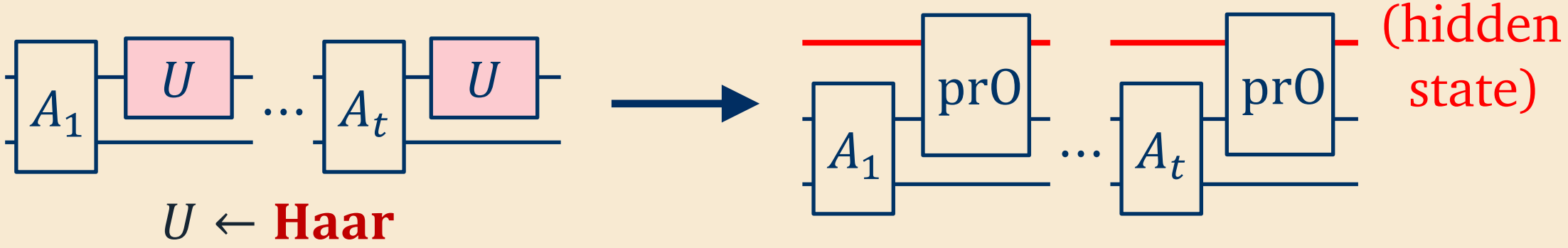


The path-recording oracle is actually a general-purpose tool for analyzing Haar-random unitaries.



Many statements about Haar-random U can be reduced to simple claims about this data structure

The path-recording oracle is actually a general-purpose tool for analyzing Haar-random unitaries.



Many statements about Haar-random U can be reduced to simple claims about this data structure

- [**M**H24]: elementary proof of [SHH24] gluing lemma
- [ABGL24]: compress PRU key length + other results

In my view: our quantum understanding is on par with where our classical understanding was 40 years ago.

In my view: our quantum understanding is on par with where our classical understanding was 40 years ago.

Pseudorandom functions [GGM84] ⟷ Pseudorandom unitaries [MH24]

In my view: our quantum understanding is on par with where our classical understanding was 40 years ago.

Pseudorandom functions [GGM84]	⟷	Pseudorandom unitaries [**M**H24]
[GGM84] proof: Lazy sampling of a random function	⟷	[**M**H24] proof: Lazy sampling of a random unitary

Future directions

Question 1a: Can we **unconditionally prove** that no efficient algorithm can break **quantum** cryptography?

Question 1a: Can we **unconditionally prove** that no efficient algorithm can break **quantum** cryptography?

Analogous classical question: can you prove that no efficient algorithm breaks **classical** cryptography?

Question 1a: Can we **unconditionally prove** that no efficient algorithm can break **quantum** cryptography?

Analogous classical question: can you prove that no efficient algorithm breaks **classical** cryptography?

Complexity-theoretic barrier: this implies P ≠ NP.

Question 1a: Can we **unconditionally prove** that no efficient algorithm can break **quantum** cryptography?

Analogous classical question: can you prove that no efficient algorithm breaks **classical** cryptography?

Complexity-theoretic barrier: this implies P ≠ NP.

[LMW24]: in the quantum setting, there might be no barriers from traditional complexity theory!

Question 1a: Can we **unconditionally prove** that no efficient algorithm can break **quantum** cryptography?

- But if PRUs exist, there might be a different barrier.

Question 1a: Can we **unconditionally prove** that no efficient algorithm can break **quantum** cryptography?

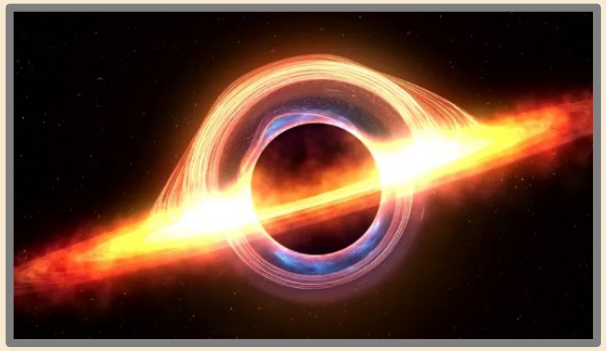
- But if PRUs exist, there might be a different barrier.
- Classically, there's the **natural proofs barrier** [RR]: PRFs "explain" why we haven't proved P ≠ NP.

Question 1a: Can we **unconditionally prove** that no efficient algorithm can break **quantum** cryptography?

- But if PRUs exist, there might be a different barrier.
- Classically, there's the **natural proofs barrier** [RR]: PRFs "explain" why we haven't proved P ≠ NP.

Question 1b: can PRUs "explain" why we can't unconditionally prove that quantum cryptography exists?

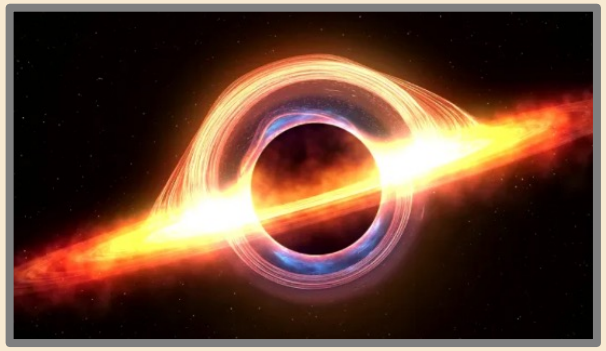
Question 2: Physicists model chaotic systems as pseudorandom unitaries. Is this actually justifiable?



$\overset{?}{\approx}$

PRU

Question 2: Physicists model chaotic systems as pseudorandom unitaries. Is this actually justifiable?

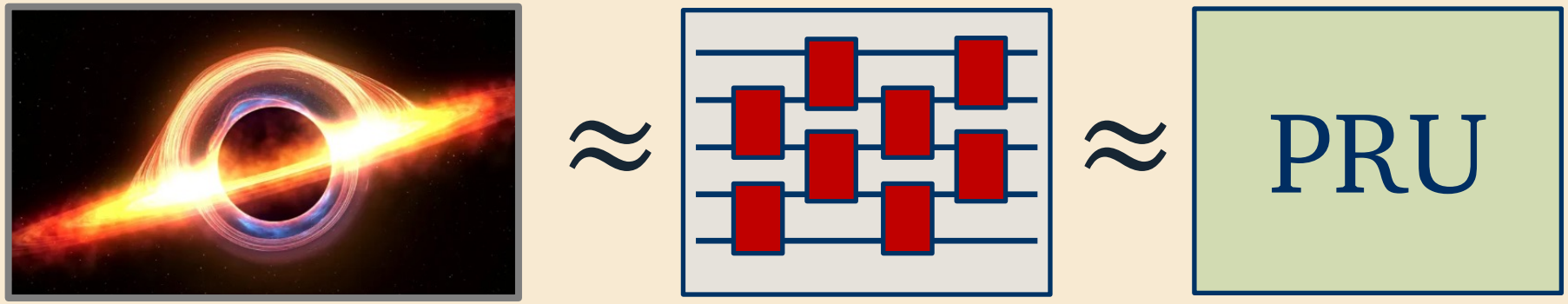


$\overset{?}{\approx}$

PRU

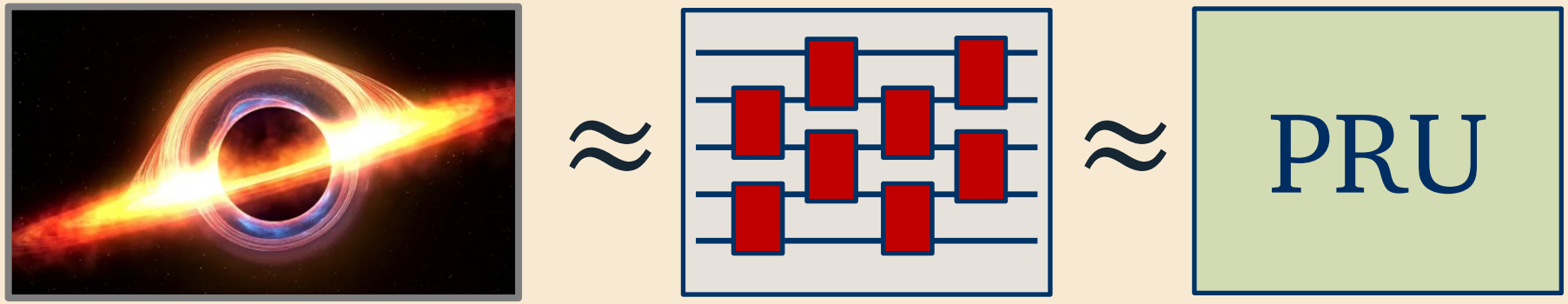
Black hole scrambling is usually modeled as a random poly(n)–size quantum circuit.

Question 2: Physicists model chaotic systems as pseudorandom unitaries. Is this actually justifiable?



Black hole scrambling is usually modeled as a random poly(n)–size quantum circuit.

Question 2: Physicists model chaotic systems as pseudorandom unitaries. Is this actually justifiable?



Black hole scrambling is usually modeled as a random poly(n)–size quantum circuit.

Question 2': Is a random quantum circuit a PRU?

Upcoming work with Alex Lombardi:

Upcoming work with Alex Lombardi:

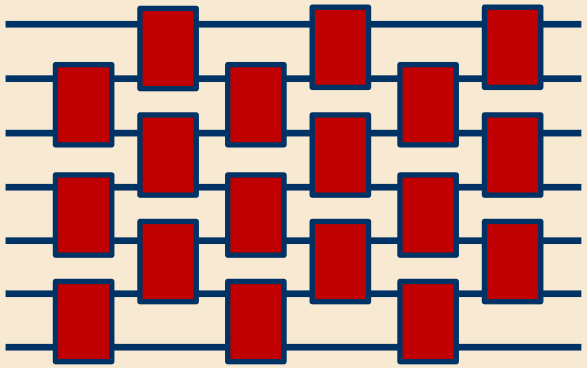
We prove that random quantum circuits are PRUs under a “structure-hiding” assumption.

What is a random quantum circuit?

What is a random quantum circuit? Many possible defs!

What is a random quantum circuit? Many possible defs!

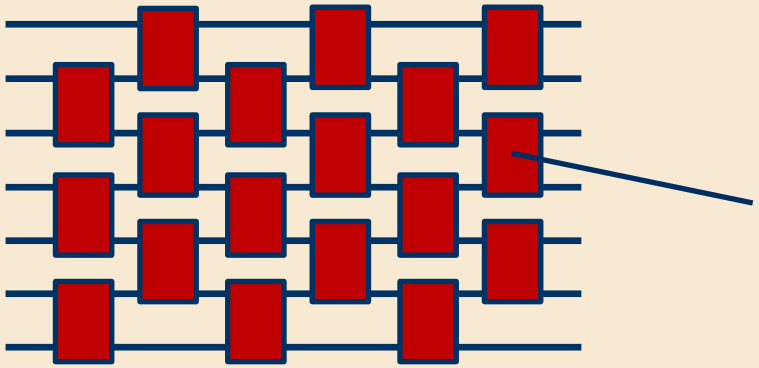
Ex: brickwork architecture



depth d

What is a random quantum circuit? Many possible defs!

Ex: brickwork architecture

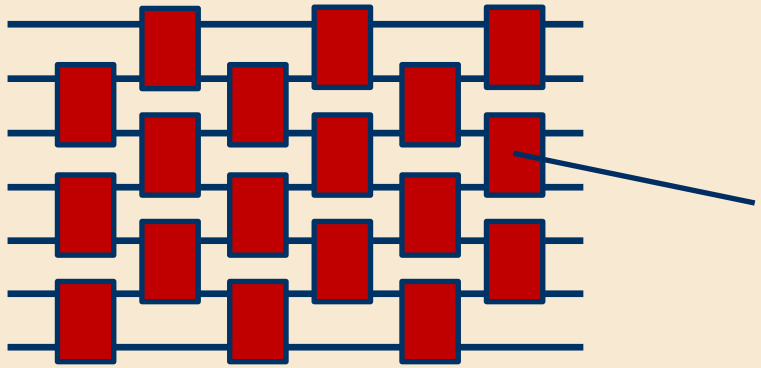


depth d

each gate is an independent 2-qubit Haar-random unitary

What is a random quantum circuit? Many possible defs!

Ex: brickwork architecture



depth d

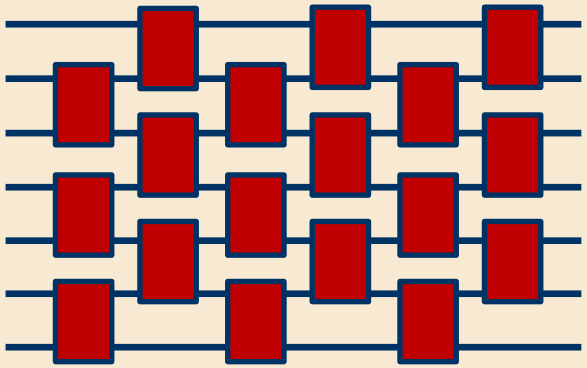
each gate is an independent 2-qubit Haar-random unitary

Actually, the distribution is still not fully specified… there are two possible “parities”.

What is a random quantum circuit? Many possible defs!

Ex: brickwork architecture

“Even”



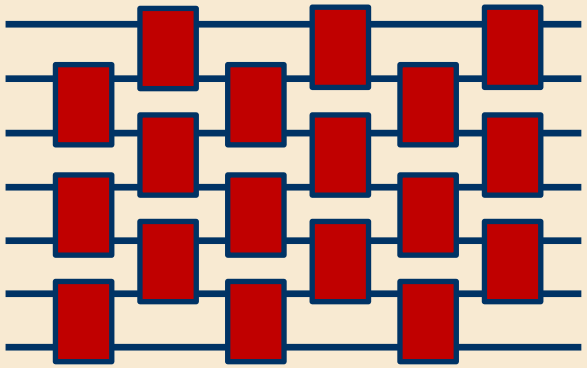
depth d

Actually, the distribution is still not fully specified…
there are two possible “parities”.

What is a random quantum circuit? Many possible defs!

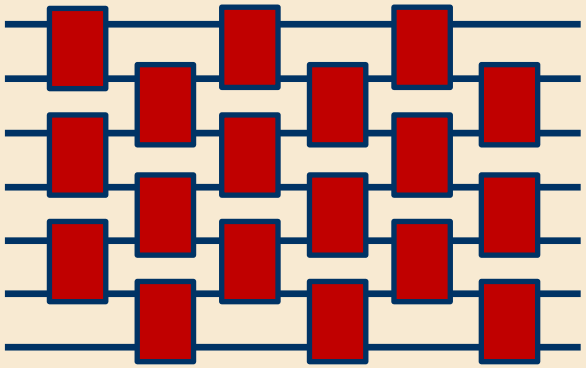
Ex: brickwork architecture

“Even”



depth d

“Odd”

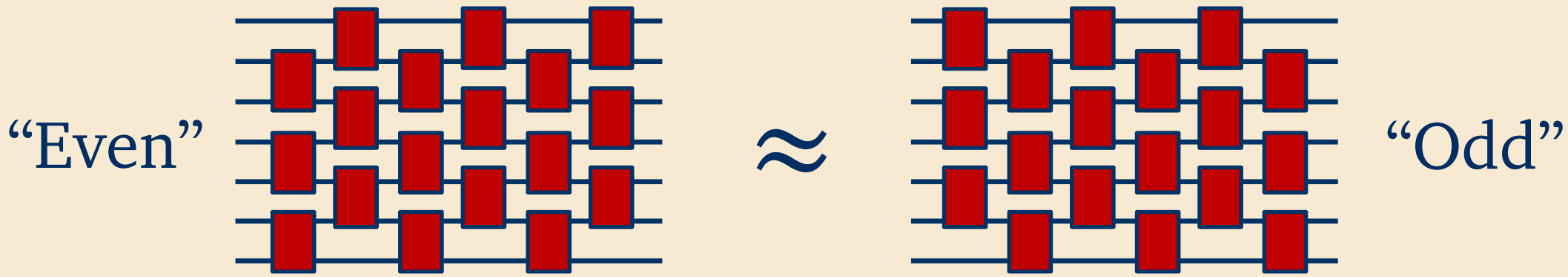


depth d

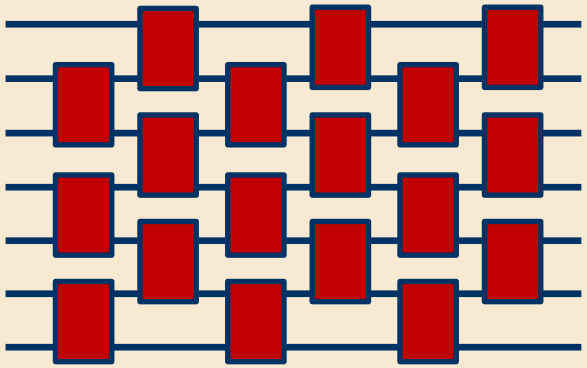
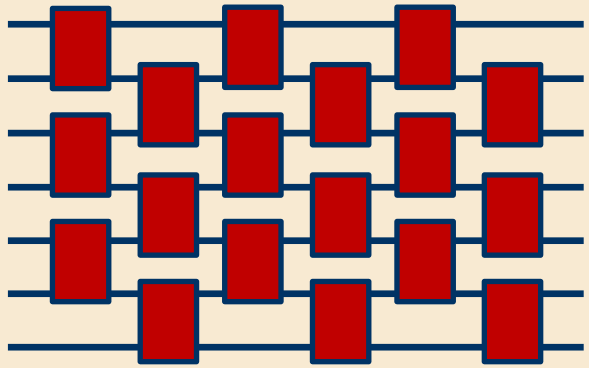
Actually, the distribution is still not fully specified…
there are two possible “parities”.

Our result: If even and odd parity random circuits are computationally indistinguishable

Our result: If even and odd parity random circuits are computationally indistinguishable



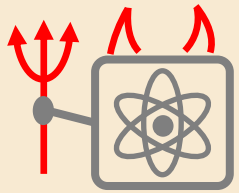
Our result: If even and odd parity random circuits are computationally indistinguishable

“Even”  ≈  “Odd”

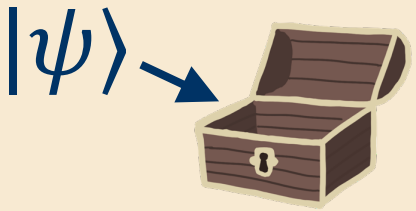
depth d depth d

…this implies a random quantum circuit is a PRU!

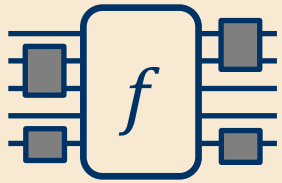
Other future directions



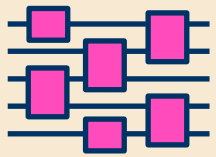
quantum security



quantum primitives



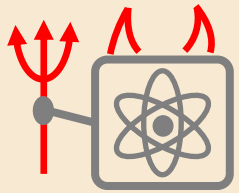
quantum hardness



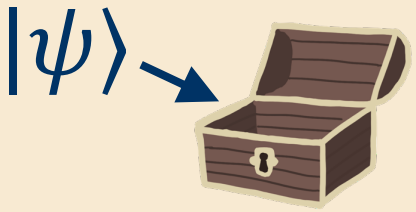
pseudo-randomness

Other future directions

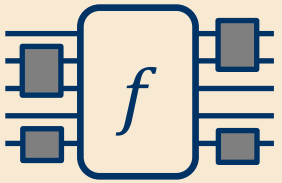
1) general framework for proving quantum security



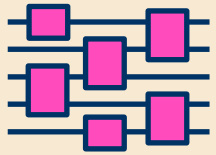
quantum security



quantum primitives



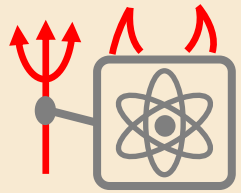
quantum hardness



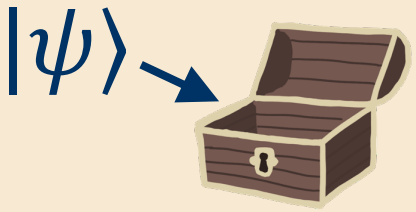
pseudo-randomness

Other future directions

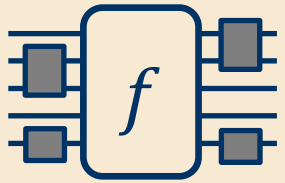
1) general framework for proving quantum security



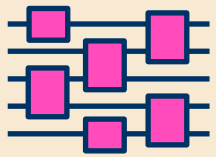
quantum security



quantum primitives



quantum hardness



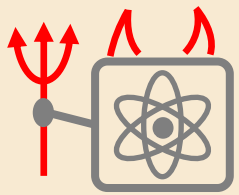
pseudo-randomness

2) “minimal” assumption for quantum crypto?

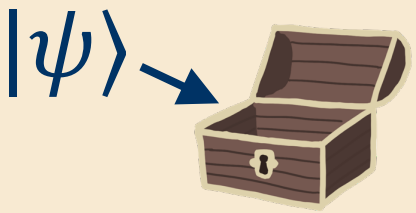
Other future directions

1) general framework for proving quantum security

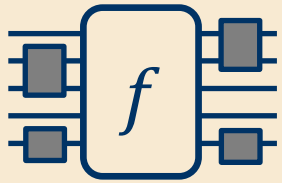
3) build out a theory of physically relevant hardness



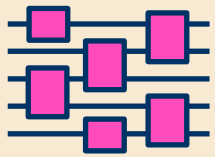
quantum security



quantum primitives



quantum hardness



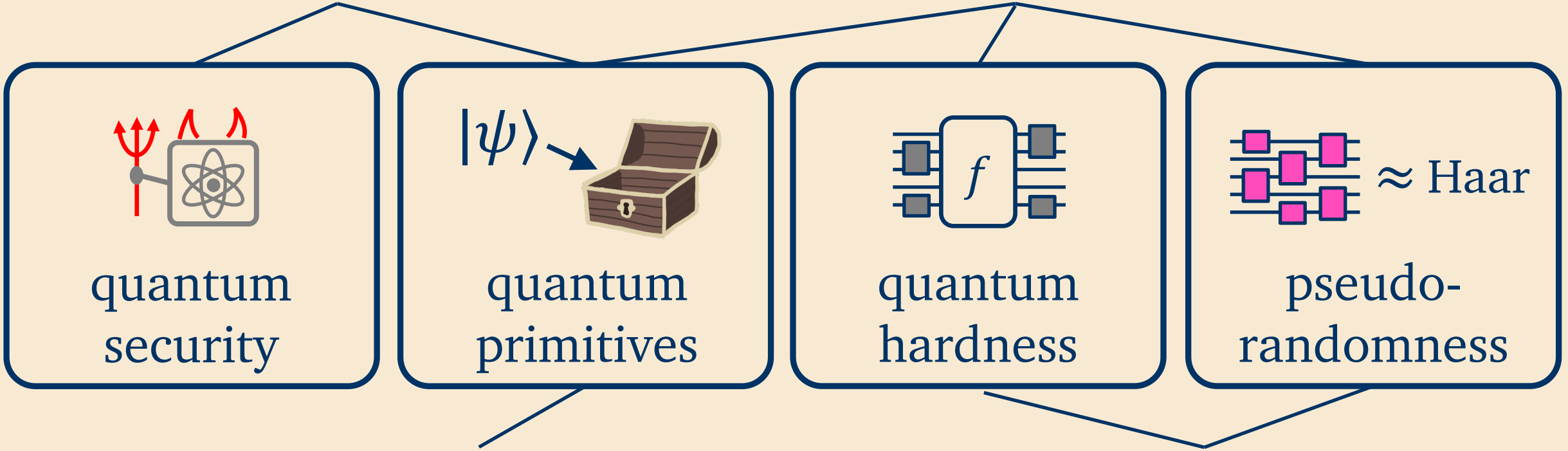
pseudo-randomness

2) “minimal” assumption for quantum crypto?

Other future directions

1) general framework for proving quantum security

3) build out a theory of physically relevant hardness



2) "minimal" assumption for quantum crypto?

4) computational complexity of unitaries

Thanks!