How to Construct Random Unitaries

Fermi Ma

(joint w/ Hsin-Yuan Huang)

Haar measure: unitarily invariant measure on $\mathrm{SU}(2^n)$

i.e., for any unitary W, if $U \sim \mathrm{Haar}$, then $W \cdot U \sim \mathrm{Haar}$

Haar measure: unitarily invariant measure on $\mathrm{SU}(2^n)$

i.e., for any unitary W, if $U \sim$ Haar, then $W \cdot U \sim$ Haar

Fundamental to understanding quantum phenomena:

Haar measure: unitarily invariant measure on $\mathrm{SU}(2^n)$

i.e., for any unitary W, if $U \sim$ Haar, then $W \cdot U \sim$ Haar

Fundamental to understanding quantum phenomena:

- information scrambling
- entanglement
- quantum learning algorithms
- quantum crypto
- random quantum circuits
- unitary complexity
- quantum error correction

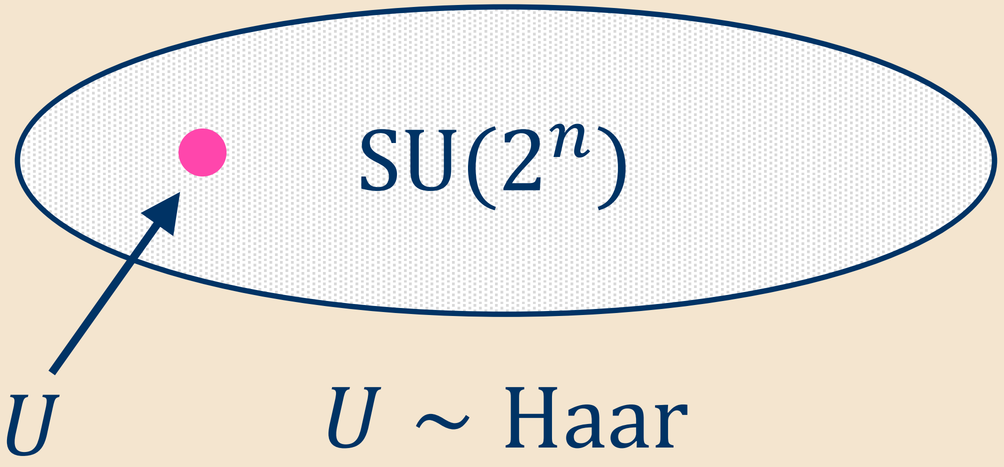
…

Challenge:

Haar-random unitaries are exponentially complex

Challenge:

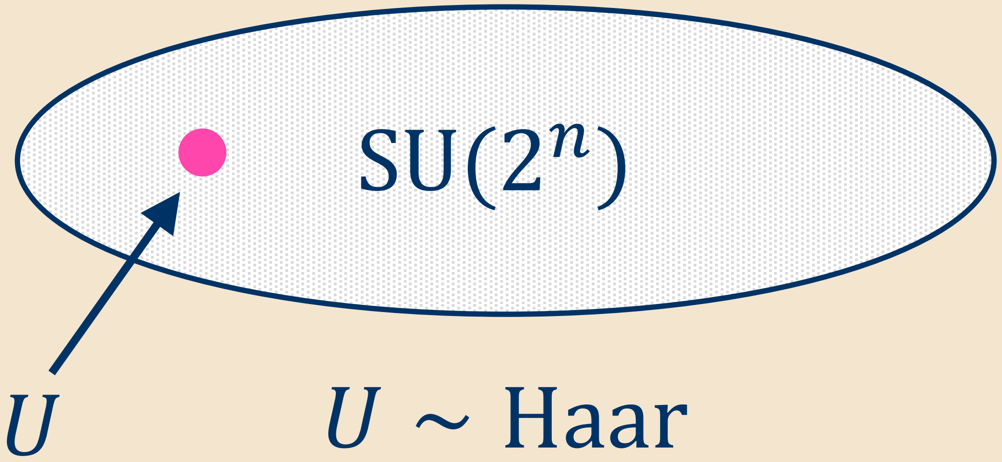
Haar-random unitaries are exponentially complex



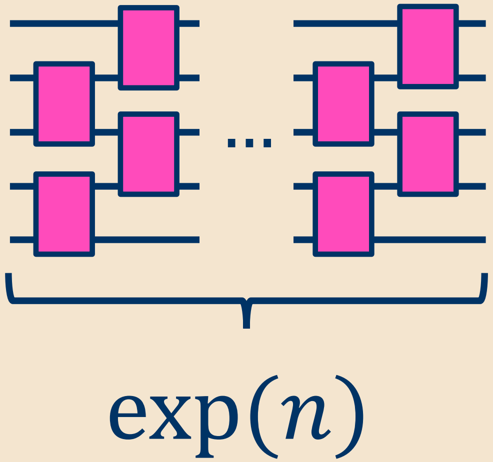
$U \sim \text{Haar}$

Challenge:

Haar-random unitaries are exponentially complex

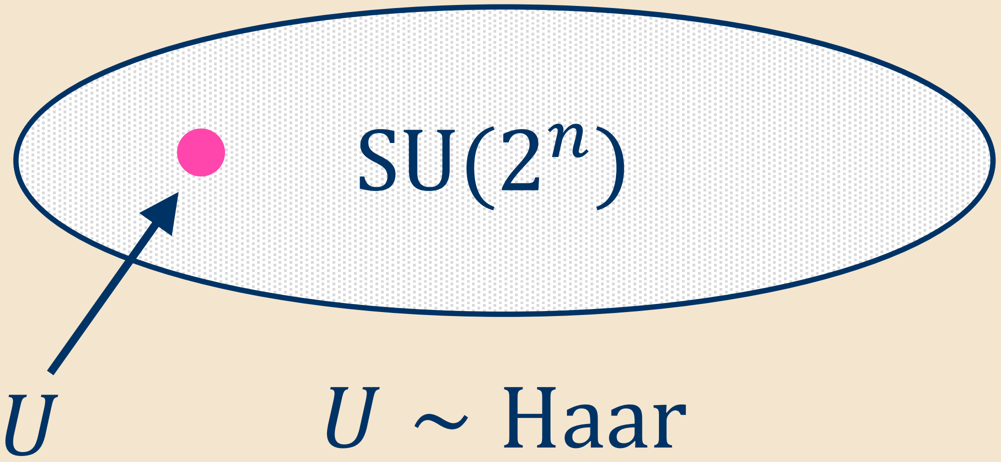


minimal circuit for U



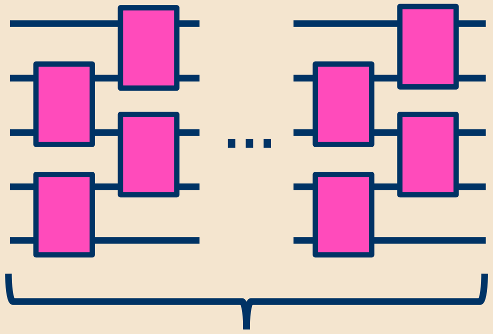
Challenge:

Haar-random unitaries are exponentially complex



$U \sim \text{Haar}$

minimal circuit for U



$\exp(n)$

This makes them impractical for most applications!

This motivates pseudorandom unitaries [JLS18].

This motivates pseudorandom unitaries [JLS18].

PRUs: efficiently-computable unitaries that are indistinguishable from Haar-random.

This motivates pseudorandom unitaries [JLS18].

PRUs: efficiently-computable unitaries that are indistinguishable from Haar-random.

PRU

$U \leftarrow \{U_k\}$

$\approx$

Haar random

$U \leftarrow \mathrm{SU}(2^n)$

computationally indistinguishable

This motivates pseudorandom unitaries [JLS18].

PRUs: efficiently-computable unitaries that are indistinguishable from Haar-random.

PRU		Haar random
$U \leftarrow \{U_k\}$	$\approx$	$U \leftarrow \mathrm{SU}(2^n)$

computationally indistinguishable

Classical analogue: pseudorandom functions (PRFs) or pseudorandom permutations (PRPs)

Open question: provably-secure PRUs

Open question: provably-secure PRUs

1) Many proposed constructions:

Open question: provably-secure PRUs

1) Many proposed constructions:

QFT $\times$ PRF $\times$ QFT $\times$ PRF

[JLS18]

Open question: provably-secure PRUs

1) Many proposed constructions:

QFT × PRF × QFT × PRF

[JLS18]

Clifford

PRP × PRF × C

[MPSY24]

Open question: provably-secure PRUs

1) Many proposed constructions:

QFT × PRF × QFT × PRF

[JLS18]

Clifford

PRP × PRF × C

[MPSY24]

2) Best-known security:

non-adaptive

[MPSY24,CBBDHX24]

Open question: provably-secure PRUs

1) Many proposed constructions:

QFT × PRF × QFT × PRF

[JLS18]

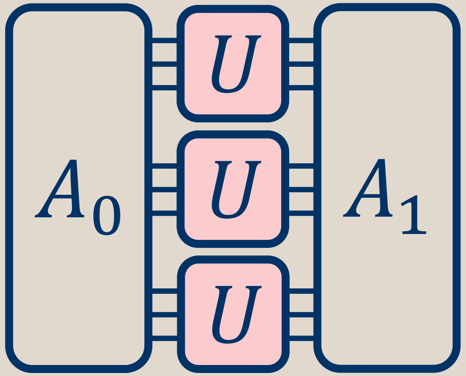
Clifford

PRP × PRF × C

[MPSY24]

2) Best-known security:
non-adaptive

[MPSY24,CBBDHX24]



non-adaptive distinguisher

This work: PRUs exist

(if one-way functions exist)

This work: PRUs exist

(if one-way functions exist)

Same construction as [MPSY24]:

PRP $\times$ PRF $\times$ C

This work: PRUs exist

(if one-way functions exist)

Same construction as [MPSY24]:

PRP × PRF × C

New technique: the path-recording oracle

This work: PRUs exist

(if one-way functions exist)

Same construction as [MPSY24]:

$\boxed{\text{PRP}} \times \boxed{\text{PRF}} \times \boxed{\text{C}}$

New technique: the path-recording oracle

- efficient simulation of Haar-random unitaries

This work: PRUs exist

(if one-way functions exist)

Same construction as [MPSY24]:

PRP × PRF × C

New technique: the path-recording oracle

- efficient simulation of Haar-random unitaries
- only uses basic quantum info (purification)

In fact, we go a step further.

In fact, we go a step further.

In the [JLS18] PRU definition, the distinguisher only queries U. **What if it queries $\boldsymbol{U}$ and $\boldsymbol{U^\dagger}$?**

In fact, we go a step further.

In the [JLS18] PRU definition, the distinguisher only queries U. **What if it queries $\boldsymbol{U}$ and $\boldsymbol{U^\dagger}$?**

We also prove that "strong" PRUs exist (assuming OWFs).

In fact, we go a step further.

In the [JLS18] PRU definition, the distinguisher only queries U. **What if it queries $\boldsymbol{U}$ and $\boldsymbol{U^\dagger}$?**

We also prove that "strong" PRUs exist (assuming OWFs).

Construction: $C_1 \times \mathrm{PRP} \times \mathrm{PRF} \times C_2$

In fact, we go a step further.

In the [JLS18] PRU definition, the distinguisher only queries U. **What if it queries U and $U^\dagger$?**

We also prove that "strong" PRUs exist (assuming OWFs).

Construction: $C_1 \times \text{PRP} \times \text{PRF} \times C_2$

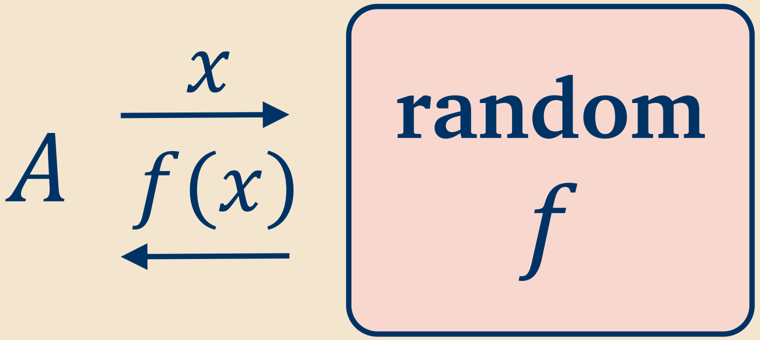
But for this talk, I'll focus on the weaker notion.

Rest of this talk

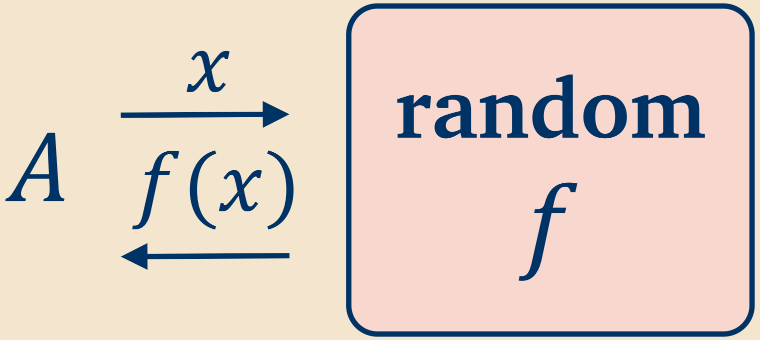
- Warmup: simulating a random function
- This work: simulating a random unitary
- Analyze PRUs

Warmup: simulating a random *function*

Warmup: simulating a random *function*

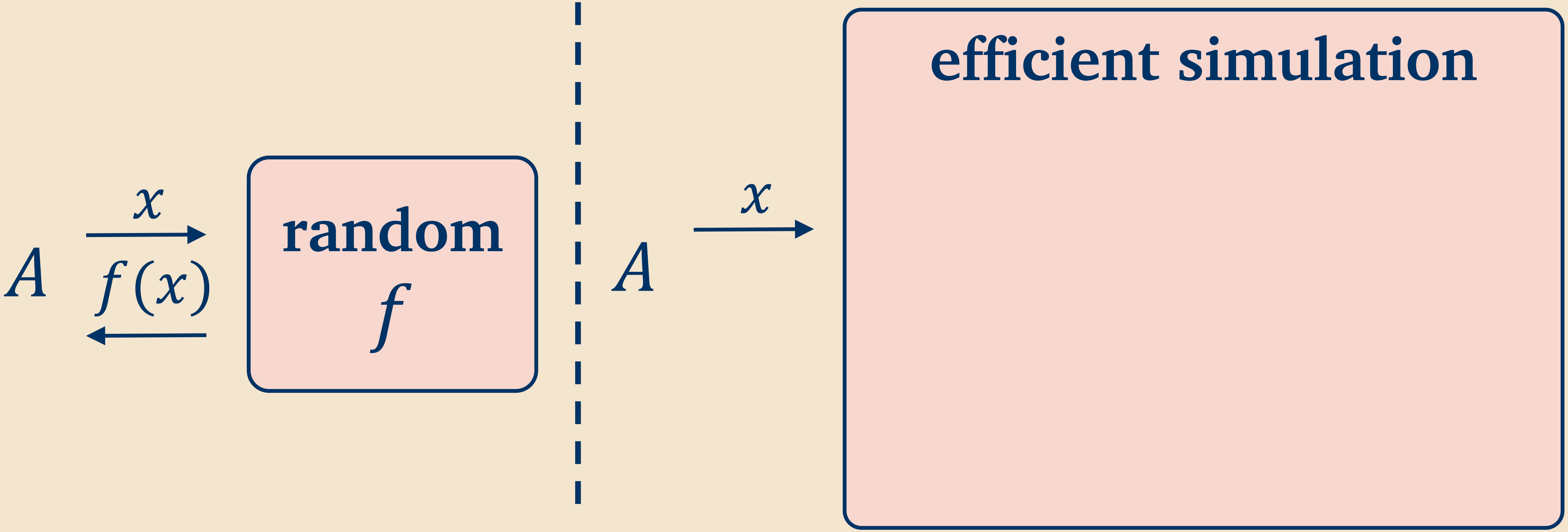


Warmup: simulating a random *function*

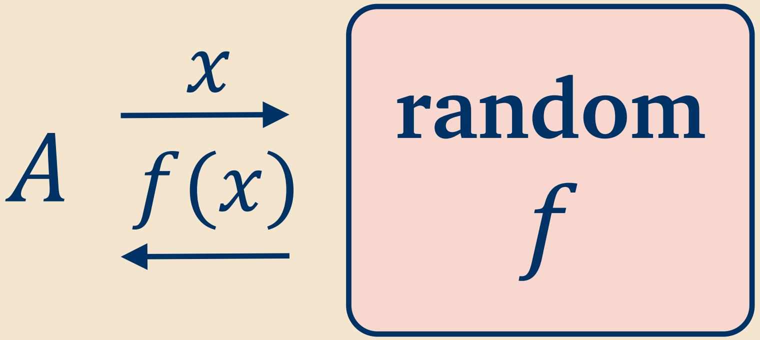


efficient simulation

Warmup: simulating a random *function*



Warmup: simulating a random *function*



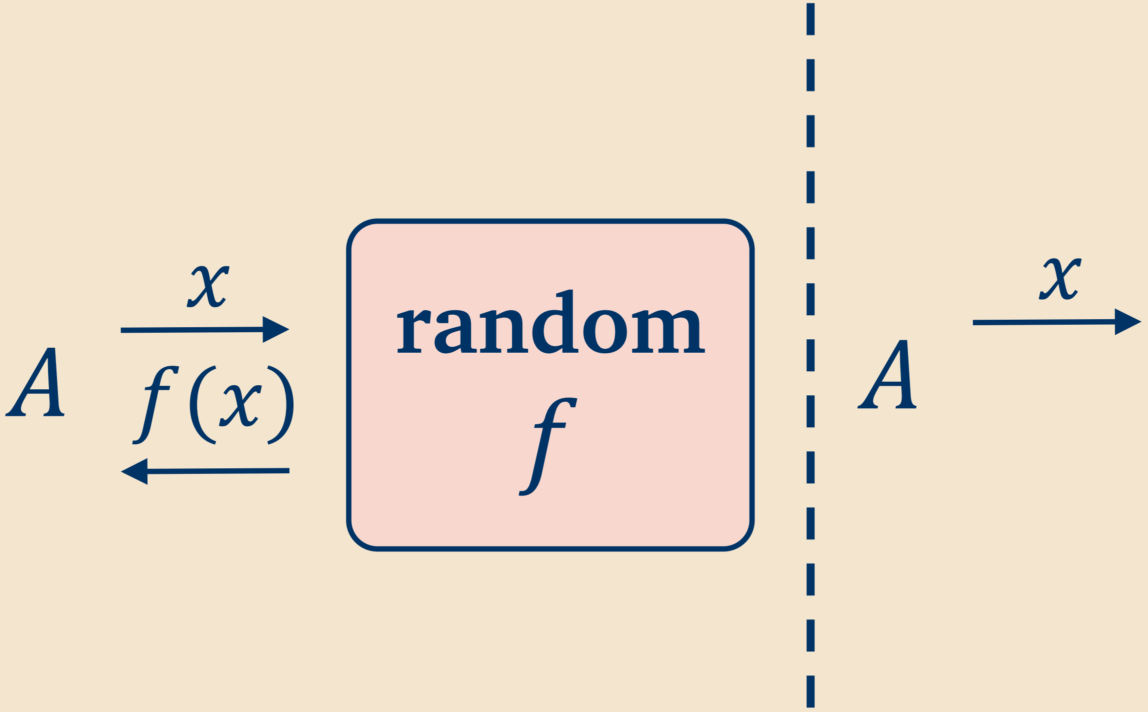
$A \xrightarrow{x}$

efficient simulation

If x is new:

- sample random y
- store (x, y)

Warmup: simulating a random *function*



efficient simulation

If x is new:

- sample random y
- store (x, y)

If x was seen before:

- return stored y

Warmup: simulating a random *function*

$A \underset{f(x)}{\overset{x}{\rightleftarrows}}$ **random** f

$A \underset{y}{\overset{x}{\rightleftarrows}}$

efficient simulation

If x is new:

- sample random y
- store (x, y)

If x was seen before:

- return stored y

Warmup: simulating a random *function*

$A \; \underset{\longleftarrow}{\overset{x}{\longrightarrow}} \; f(x)$

random

f

$A \; \underset{\longleftarrow}{\overset{x}{\longrightarrow}} \; y$

efficient simulation

If x is new:

- sample random y
- store (x, y)

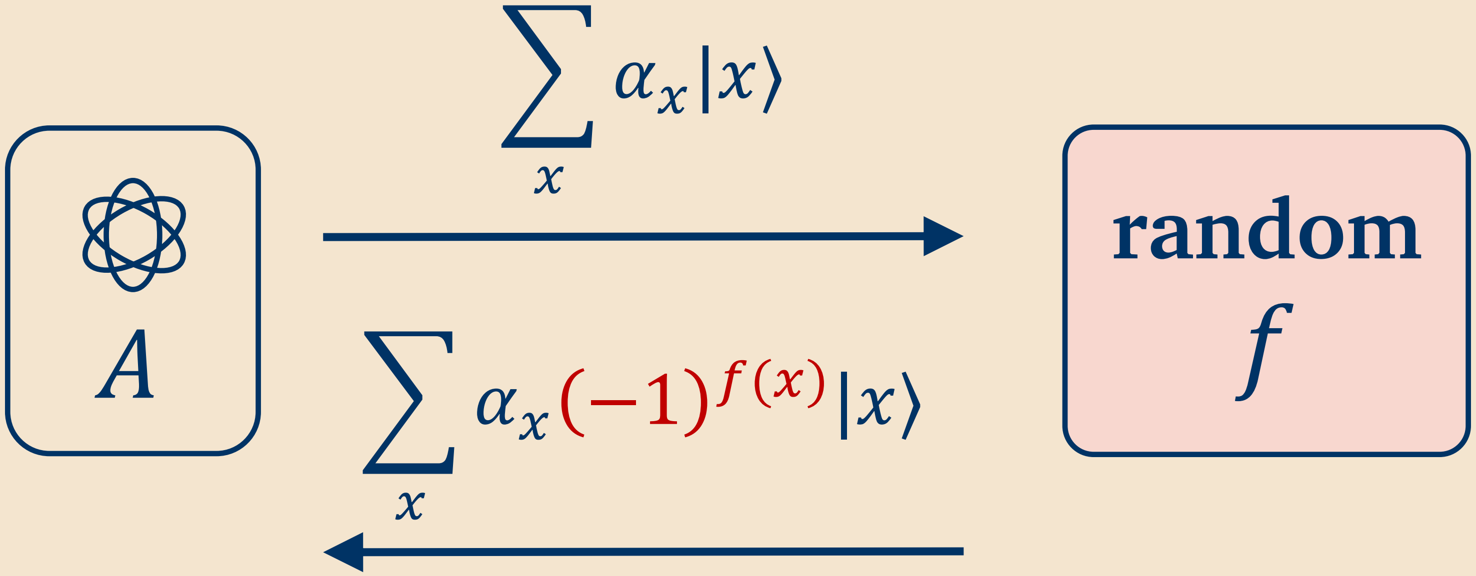
If x was seen before:

- return stored y

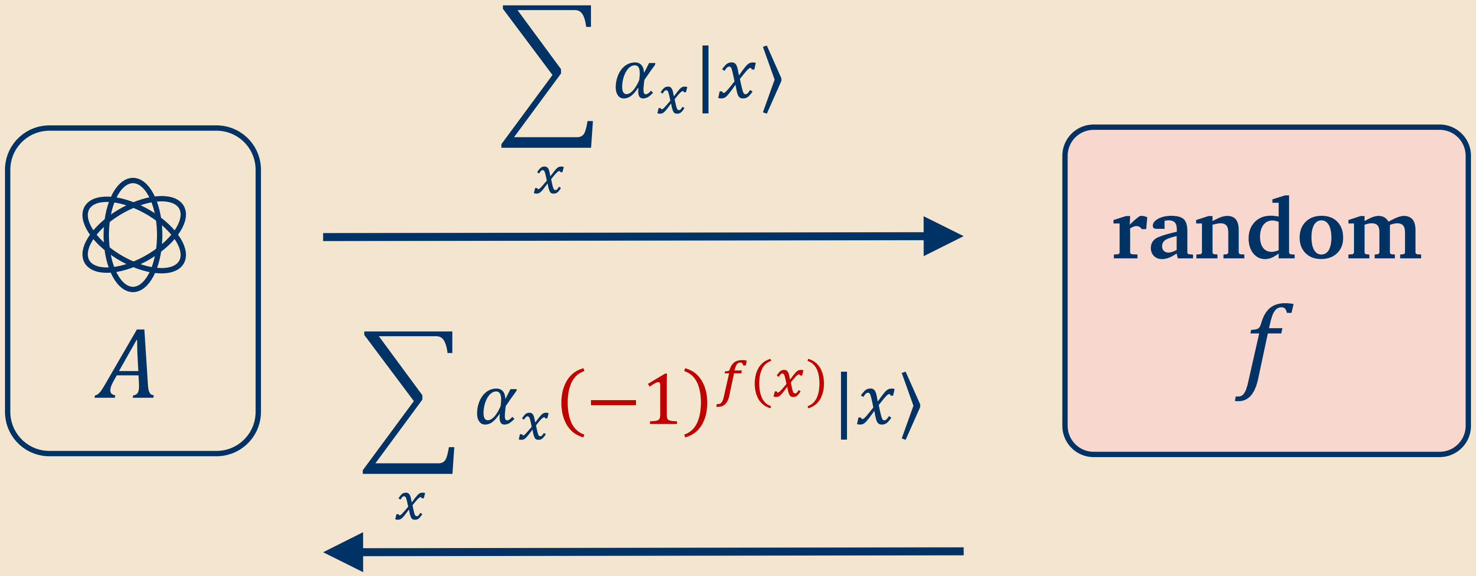
Clearly identical from $A's$ point of view!

Note: properly simulating a random function f is quite subtle if A can query f in superposition:

Note: properly simulating a random function f is quite subtle if A can query f in superposition:

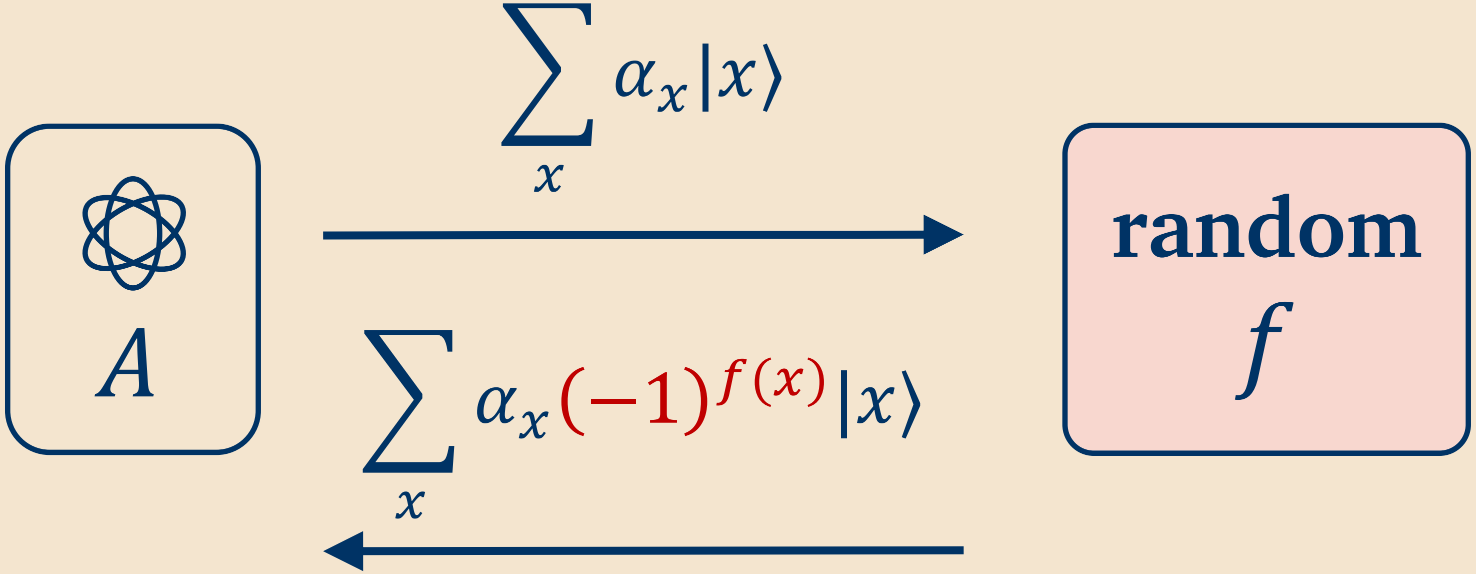


Note: properly simulating a random function f is quite subtle if A can query f in superposition:



[Z19] simulates this via the **compressed oracle.**

Note: properly simulating a random function f is quite subtle if A can query f in superposition:



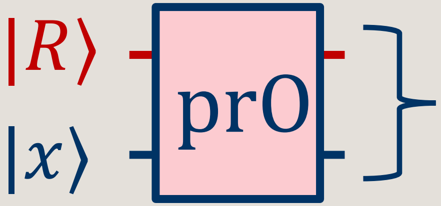
[Z19] simulates this via the **compressed oracle.**

Our **path-recording oracle** is a unitary analog of [Z19].

The path-recording oracle prO

$$|R\rangle\,|x\rangle \;\xrightarrow{\mathrm{prO}}\; \sum_{y \notin R} |y\rangle\,|R \cup \{(x, y)\}\rangle$$

The path-recording oracle prO



$$\sum_{y \notin R} |y\rangle |R \cup \{(x, y)\}\rangle$$

- $R = \{(x_1, y_1), \dots, (x_t, y_t)\}$
- sum over $y \notin \{y_1, \dots, y_t\}$

The path-recording oracle prO

$$|R\rangle, |x\rangle \xrightarrow{\mathrm{prO}} \sum_{y \notin R} |y\rangle |R \cup \{(x, y)\}\rangle$$

- $R = \{(x_1, y_1), \dots, (x_t, y_t)\}$
- sum over $y \notin \{y_1, \dots, y_t\}$

(actually, we should have a $\frac{1}{\sqrt{N - |R|}}$ in front)

The path-recording oracle prO

$$|R\rangle, |x\rangle \to \boxed{\text{prO}} \to \sum_{y \notin R} |y\rangle |R \cup \{(x,y)\}\rangle$$

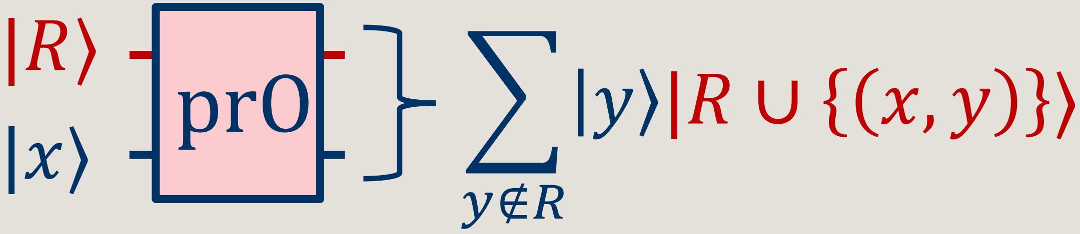
- $R = \{(x_1, y_1), \dots, (x_t, y_t)\}$
- sum over $y \notin \{y_1, \dots, y_t\}$

(actually, we should have a $\frac{1}{\sqrt{N-|R|}}$ in front)

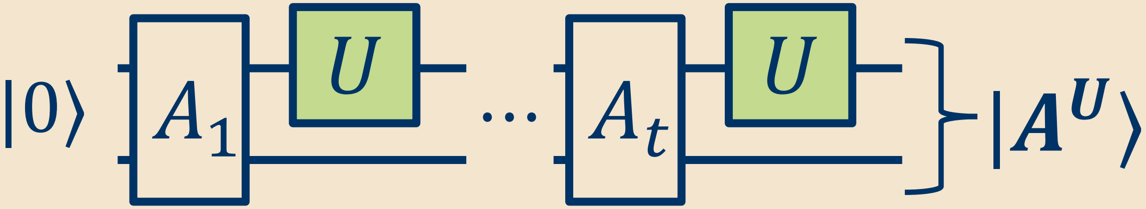
Note: prO is an isometry.

Intuition: $|y\rangle|R \cup \{(x,y)\}\rangle$ uniquely determines $|x\rangle|R\rangle$.

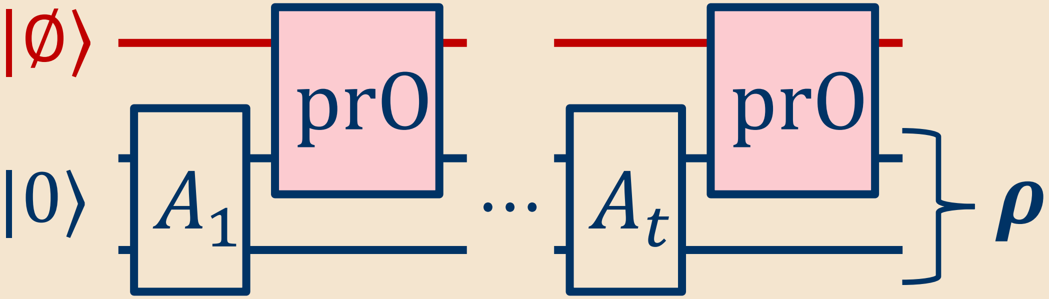
The path-recording oracle prO



- $R = \{(x_1, y_1), \ldots, (x_t, y_t)\}$
- sum over $y \notin \{y_1, \ldots, y_t\}$

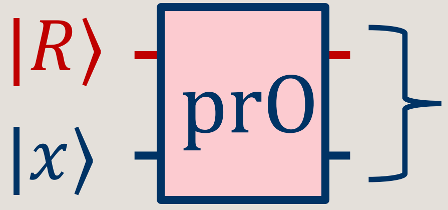


$U \leftarrow$ Haar



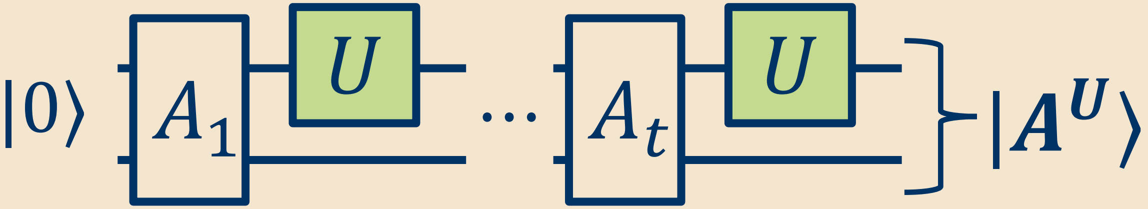
Efficient simulation

The path-recording oracle prO

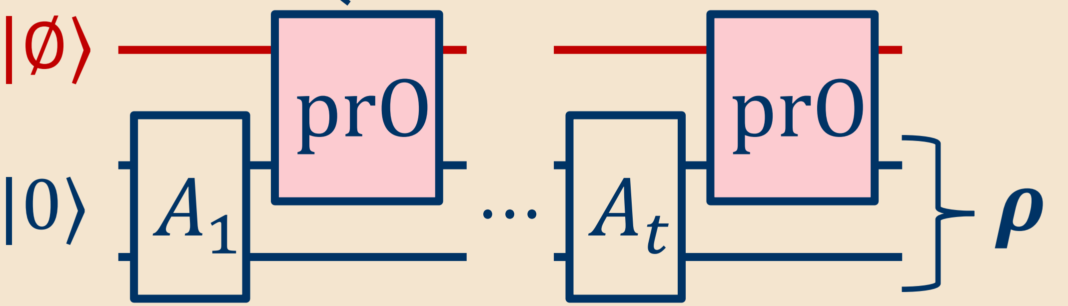


$$\sum_{y \notin R} |y\rangle |R \cup \{(x,y)\}\rangle$$

- $R = \{(x_1, y_1), \ldots, (x_t, y_t)\}$
- sum over $y \notin \{y_1, \ldots, y_t\}$



$U \leftarrow$ Haar



Efficient simulation

Claim: $\mathbb{E}_{U \leftarrow \mathbf{Haar}} |\boldsymbol{A}^U\rangle\langle \boldsymbol{A}^U|$ and $\boldsymbol{\rho}$ have trace distance $\leq t^2/2^n$.

Up next: a few examples

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

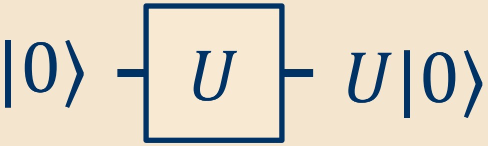
$$|0\rangle \; - \boxed{U} - \; U|0\rangle$$

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

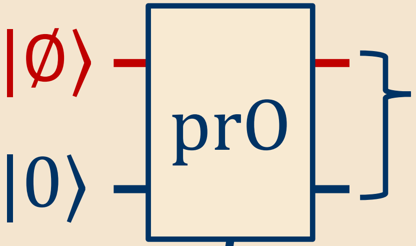
$$|0\rangle \;-\!\boxed{U}\!-\; U|0\rangle$$

For $U \leftarrow$ Haar, this is **maximally mixed**.

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



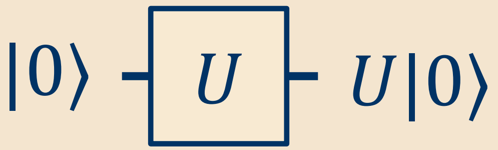
For $U \leftarrow$ Haar, this is **maximally mixed**.



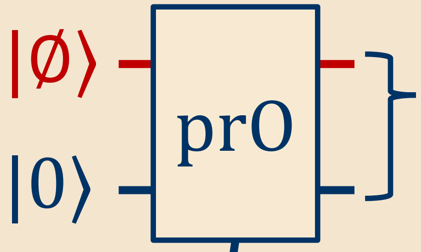
$\sum_y |y\rangle \otimes |\{(0,y)\}\rangle$

$$\mathrm{prO}\ |x\rangle|R\rangle = \sum_{y \notin R_Y} |y\rangle|R \cup \{(x,y)\}\rangle$$

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



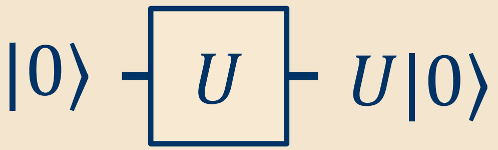
For $U \leftarrow$ Haar, this is **maximally mixed.**



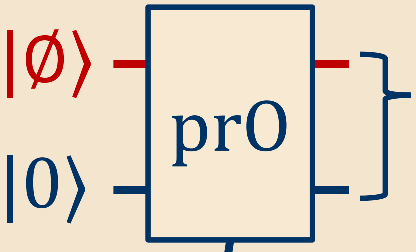
$\sum_y |y\rangle \otimes \underbrace{|\{(0,y)\}\rangle}_{\text{trace out/measure}}$

$\mathrm{prO}\,|x\rangle|R\rangle = \sum_{y \notin R_Y} |y\rangle|R \cup \{(x,y)\}\rangle$

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



For $U \leftarrow$ Haar, this is maximally mixed.

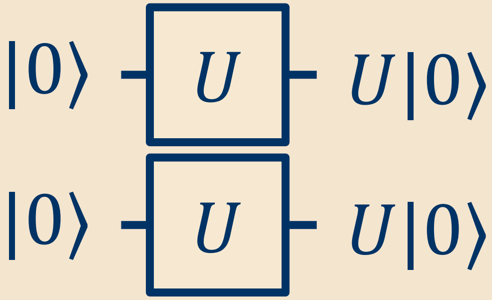


$$\sum_y |y\rangle \otimes \underbrace{|\{(0,y)\}\rangle}_{\text{trace out/measure}}$$

After tracing out: uniform mixture over $|y\rangle$, which is **maximally mixed**.

$$\boxed{\mathrm{prO}\,|x\rangle|R\rangle = \sum_{y \notin R_Y} |y\rangle|R \cup \{(x,y)\}\rangle}$$

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

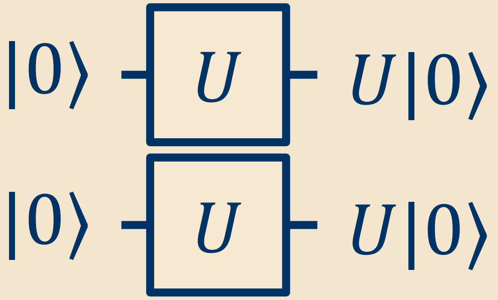


Example 2: two queries on $|0\rangle$

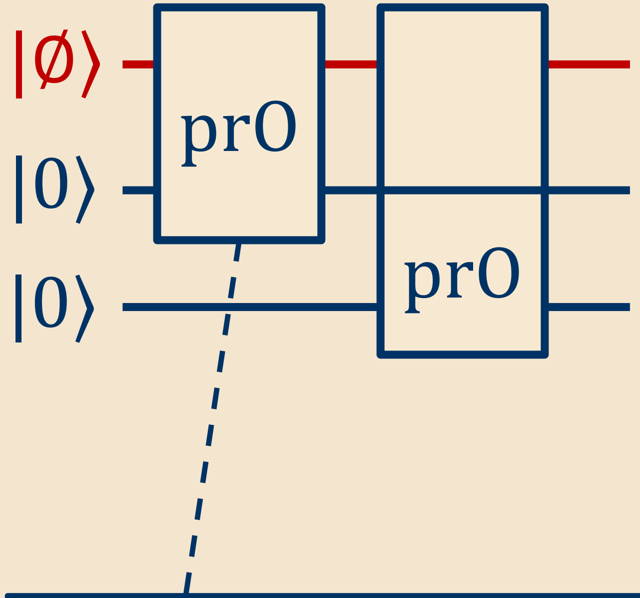
$$|0\rangle \, -\boxed{U}- \, U|0\rangle$$
$$|0\rangle \, -\boxed{U}- \, U|0\rangle$$

For $U \leftarrow$ Haar, this is **maximally mixed on the symmetric subspace (swap-invariant).**

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

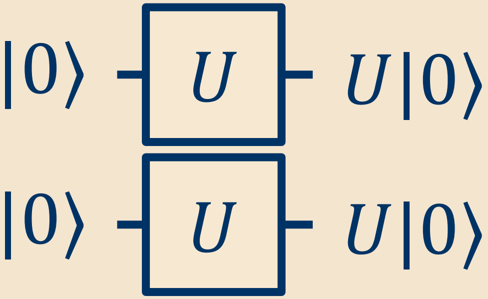


For $U \leftarrow$ Haar, this is **maximally mixed on the symmetric subspace (swap-invariant).**

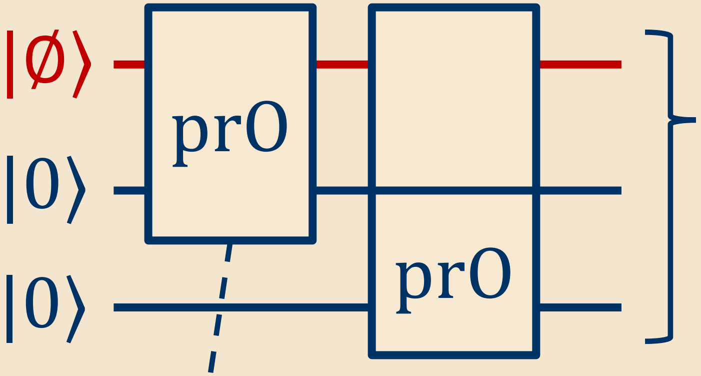


$$\mathrm{prO}\,|x\rangle|R\rangle = \sum_{y \notin R_Y} |y\rangle|R \cup \{(x,y)\}\rangle$$

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



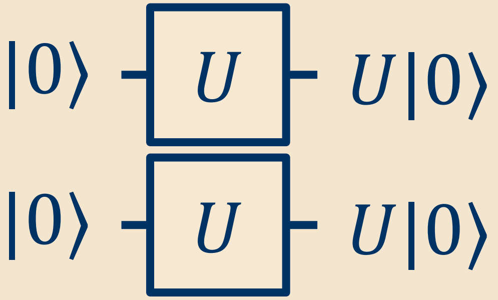
For $U \leftarrow$ Haar, this is **maximally mixed on the symmetric subspace (swap-invariant).**



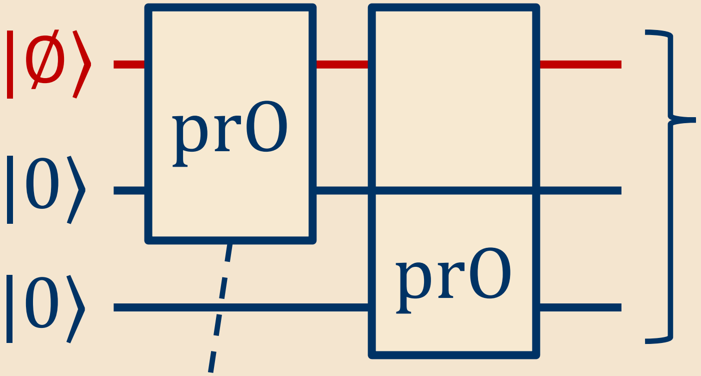
$\sum_{y_1 \neq y_2} |y_1, y_2\rangle \otimes |\{(0, y_1), (0, y_2)\}\rangle$

$\text{prO}\,|x\rangle|R\rangle = \sum_{y \notin R_Y} |y\rangle|R \cup \{(x, y)\}\rangle$

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



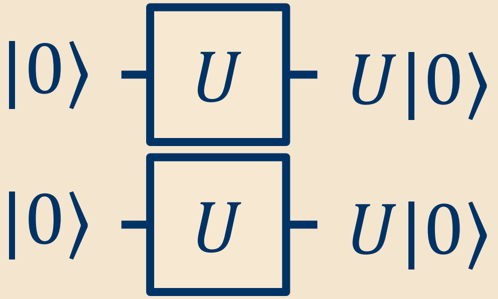
For $U \leftarrow$ Haar, this is **maximally mixed on the symmetric subspace (swap-invariant).**



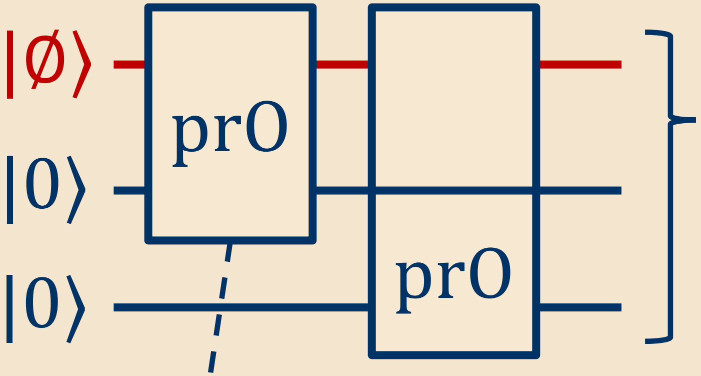
$$\sum_{y_1 \neq y_2} |y_1, y_2\rangle \otimes \underbrace{|\{(0, y_1), (0, y_2)\}\rangle}_{\text{trace out/measure}}$$

$$\mathrm{prO}\, |x\rangle|R\rangle = \sum_{y \notin R_Y} |y\rangle|R \cup \{(x, y)\}\rangle$$

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



For $U \leftarrow$ Haar, this is **maximally mixed on the symmetric subspace (swap-invariant).**



$$\sum_{y_1 \neq y_2} |y_1, y_2\rangle \otimes \underbrace{|\{(0, y_1), (0, y_2)\}\rangle}_{\text{trace out/measure}}$$

Remaining state: $|y_1, y_2\rangle + |y_2, y_1\rangle$ for random distinct y_1, y_2. This is **swap-invariant + almost maximally random**.

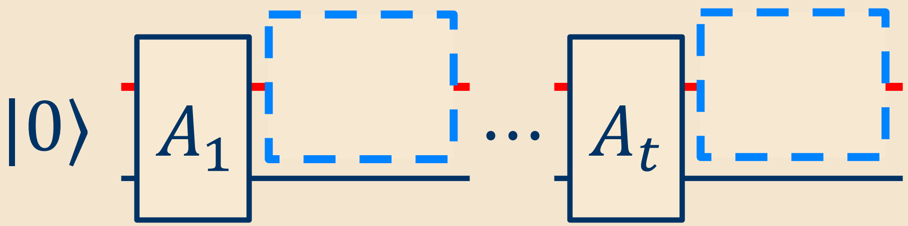
$$\text{prO}\,|x\rangle|R\rangle = \sum_{y \notin R_Y} |y\rangle|R \cup \{(x, y)\}\rangle$$

Next, we'll sketch:

path-recording oracle $\mathsf{prO} \approx$ Haar-random U

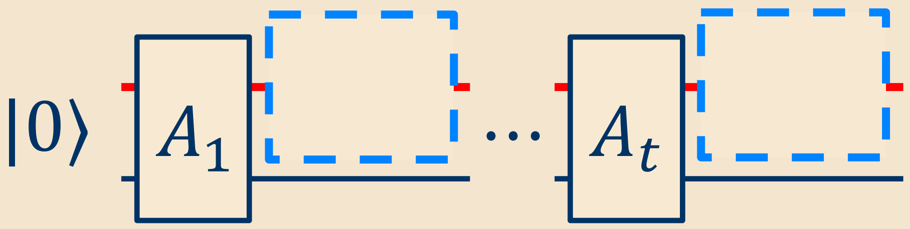
Next, we'll sketch:

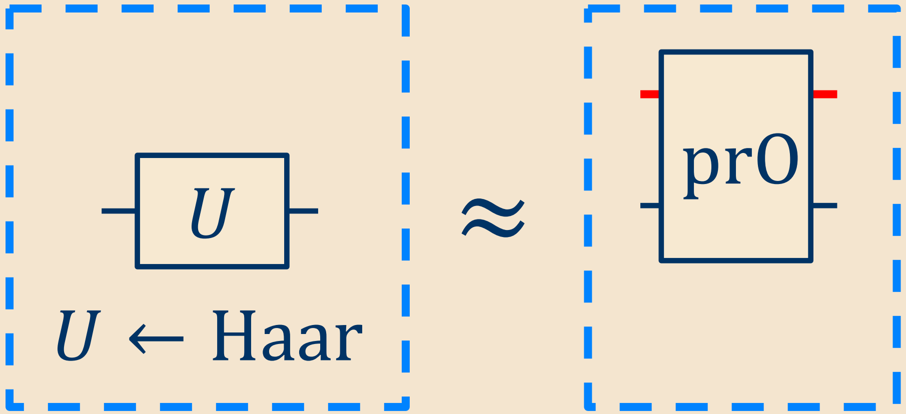
path-recording oracle $\mathrm{pr}\mathcal{O} \approx$ Haar-random U



Next, we'll sketch:

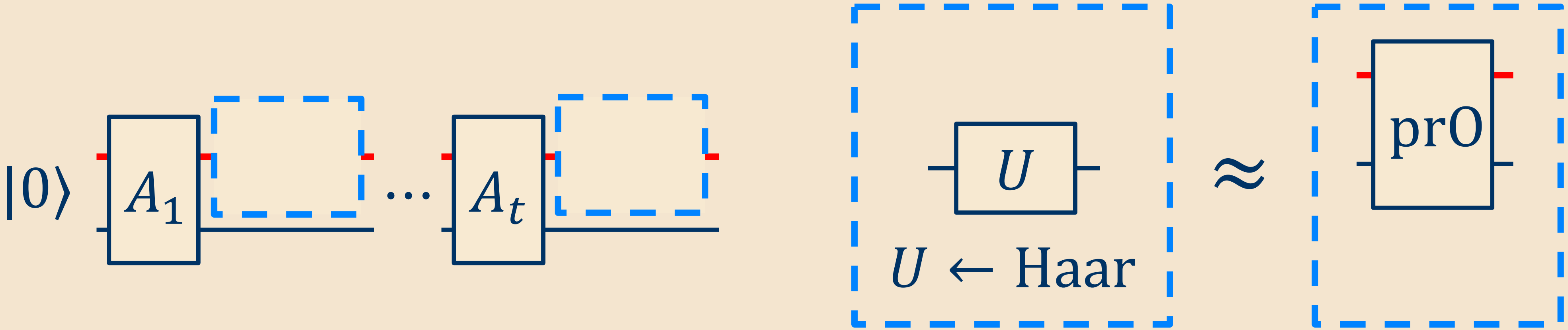
path-recording oracle $\mathrm{prO} \approx$ Haar-random U





Next, we'll sketch:

path-recording oracle prO $\approx$ Haar-random U



The plan: hybrid argument.

The same proof will also show existence of PRUs!

$U \leftarrow \text{Haar}$

Hybrid 0

$$-\boxed{U}-$$

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

$$-\boxed{PF}-\boxed{U}-$$

$$U \leftarrow \text{Haar}$$
$$P \leftarrow S_N$$
$$F \leftarrow \{\pm 1\}^N$$

$$\text{Hybrid } 0 \quad \equiv \quad \text{Hybrid } 1$$

$$-\boxed{U}-$$

$U \leftarrow \mathrm{Haar}$

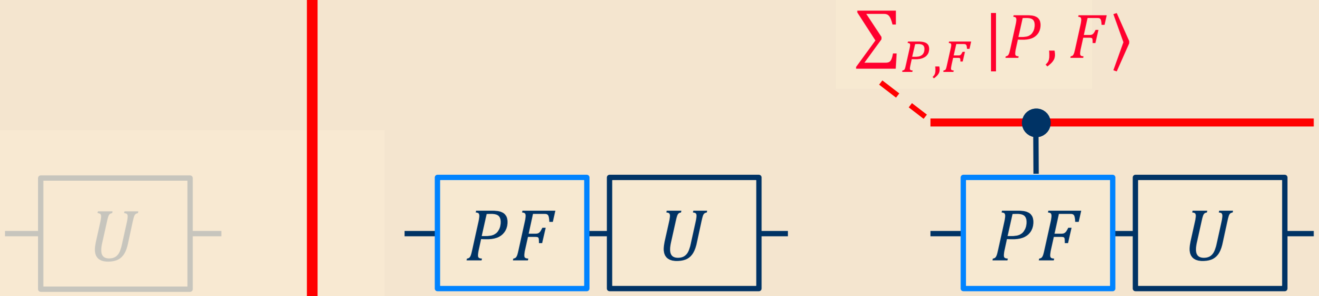
$$-\boxed{PF}-\boxed{U}-$$

$U \leftarrow \mathrm{Haar}$
$P \leftarrow S_N$
$F \leftarrow \{\pm 1\}^N$

Hybrid 0 $\equiv$ Hybrid 1

Step 1: insert random permutation P random ± 1 diagonal F.

$$P = \begin{pmatrix} & & 1 \\ 1 & & \\ & 1 & \end{pmatrix} \qquad F = \begin{pmatrix} +1 & & \\ & -1 & \\ & & -1 \end{pmatrix}$$

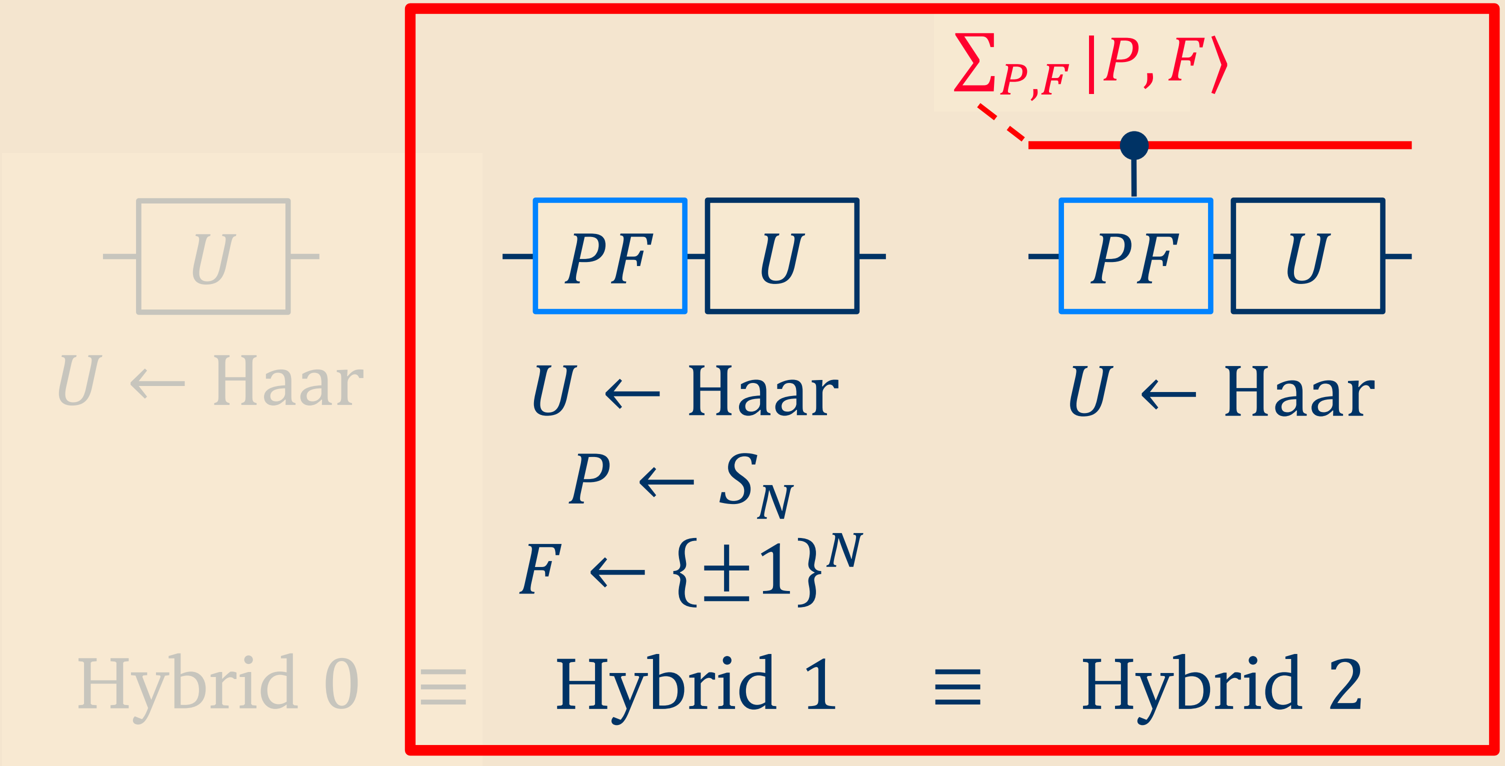


$U \leftarrow \text{Haar}$ | $U \leftarrow \text{Haar}$ | $U \leftarrow \text{Haar}$

$P \leftarrow S_N$

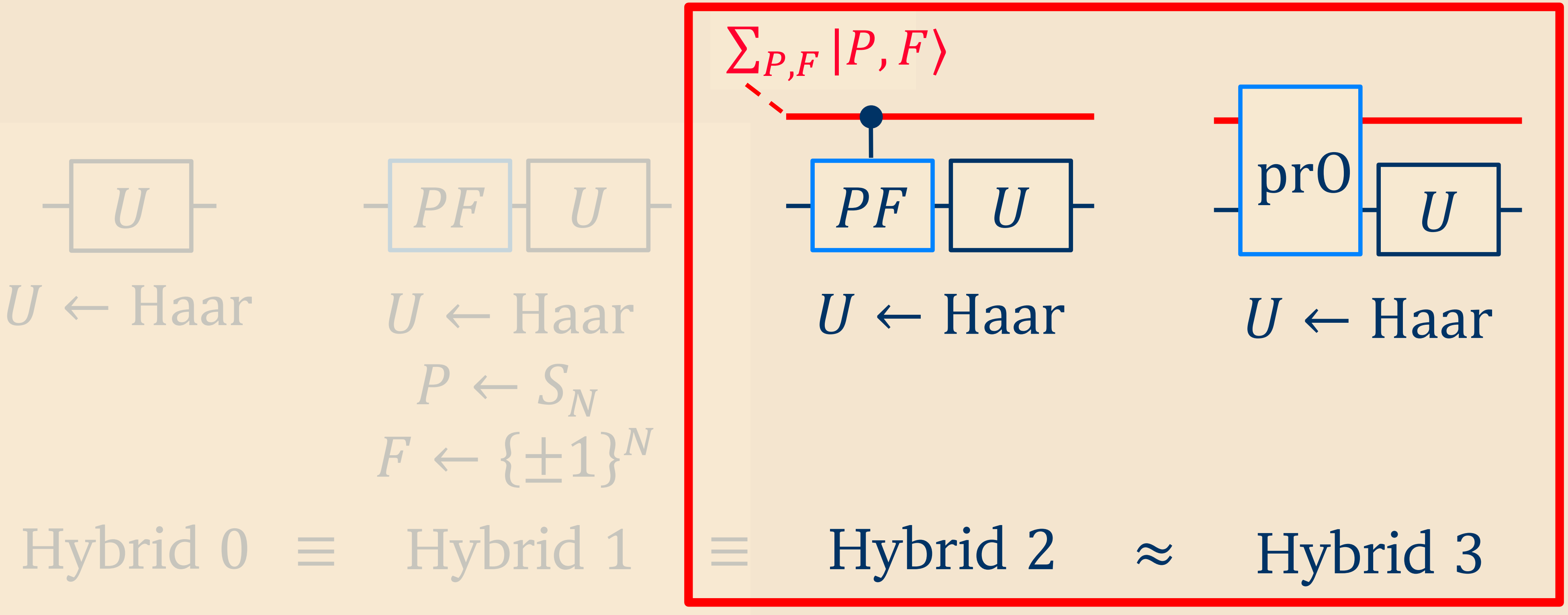
$F \leftarrow \{\pm 1\}^N$

Hybrid 0 $\equiv$ Hybrid 1 $\equiv$ Hybrid 2

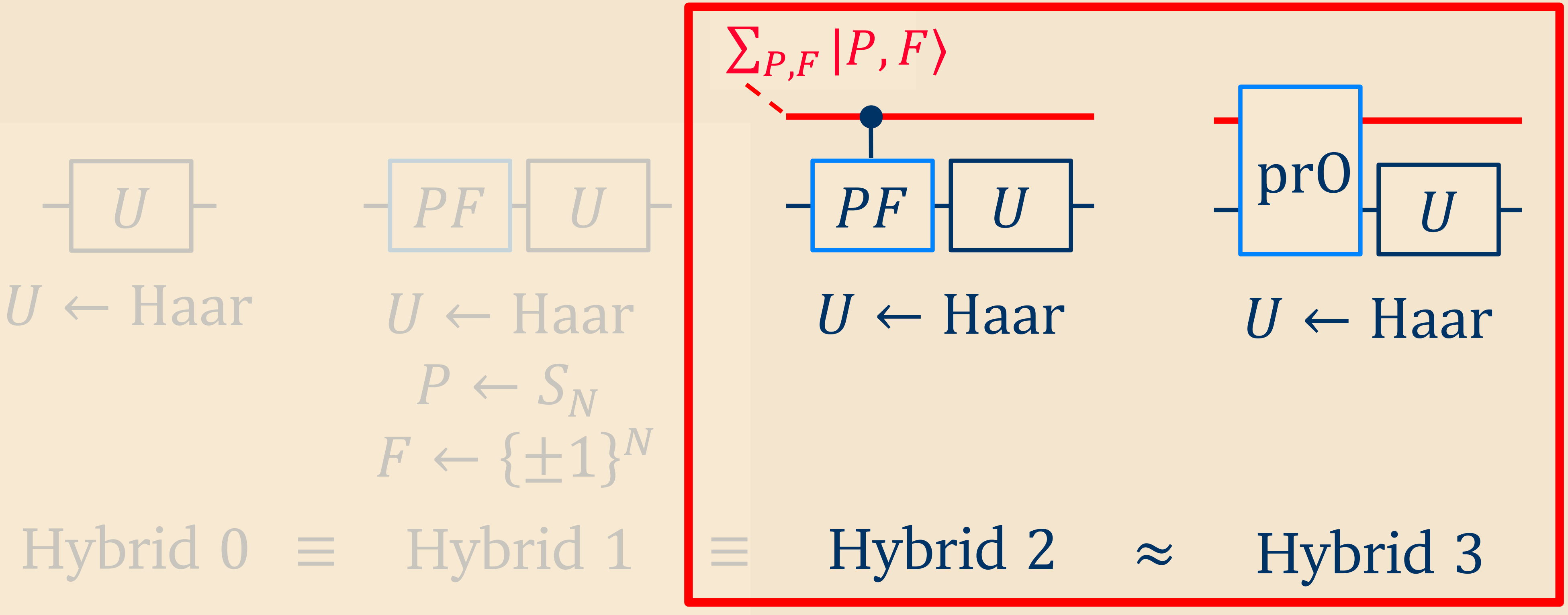


Step 2: replace random P, F with a purification.

- Initialize external/ancilla system to $\sum_{P,F} |P, F\rangle$
- On each query, apply $P \cdot F$ **controlled** on $|P, F\rangle$

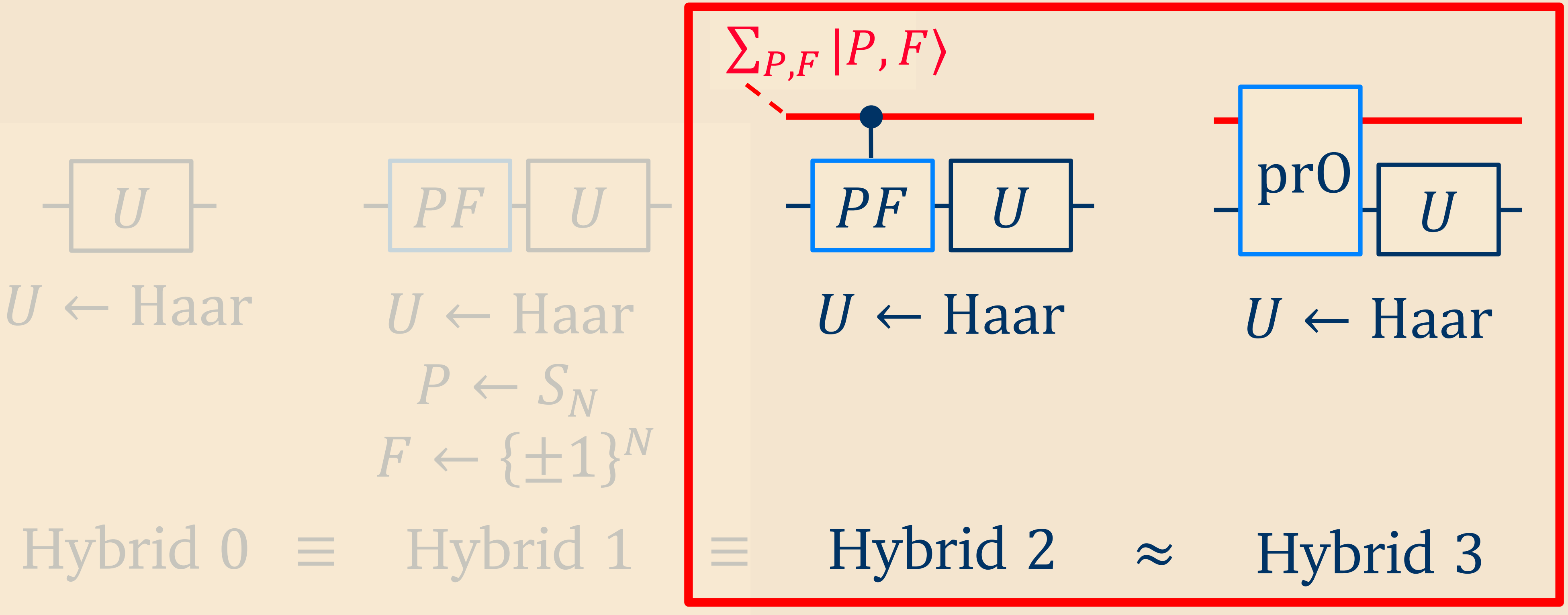


$\sum_{P,F} |P, F\rangle$
U
PF
U
PF
U
prO
U
$U \leftarrow \text{Haar}$
$U \leftarrow \text{Haar}$
$P \leftarrow S_N$
$F \leftarrow \{\pm 1\}^N$
$U \leftarrow \text{Haar}$
$U \leftarrow \text{Haar}$
Hybrid 0 $\equiv$ Hybrid 1 $\equiv$ Hybrid 2 $\approx$ Hybrid 3



Step 3: For any $R = \{(x_1, y_1), \ldots, (x_t, y_t)\}$ can define $|\Phi_R\rangle$ s.t.

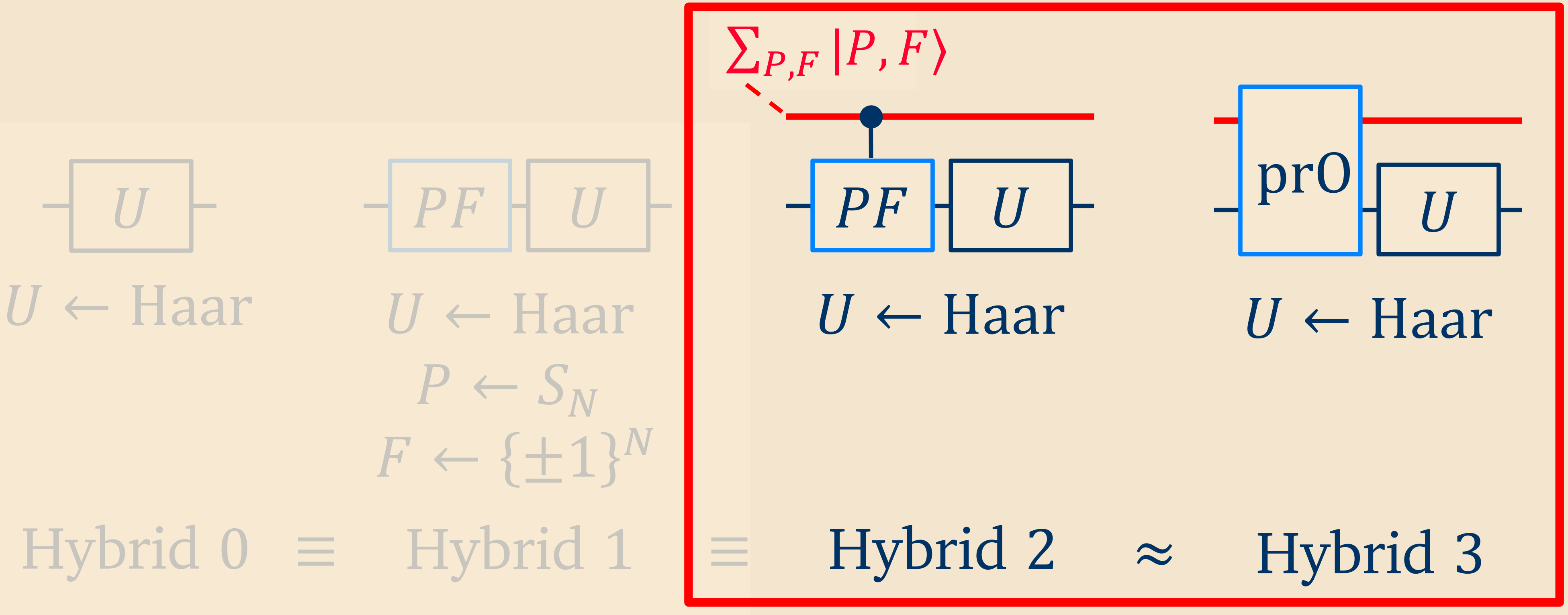
$$\text{ctl-PF} \cdot |x\rangle|\Phi_R\rangle = \sum_{y \notin R_Y} |y\rangle|\Phi_{R \cup \{(x,y)\}}\rangle$$



Step 3: For any $R = \{(x_1, y_1), \ldots, (x_t, y_t)\}$ can define $|\Phi_R\rangle$ s.t.

$$\text{ctl-PF} \cdot |x\rangle|\Phi_R\rangle = \sum_{y \notin R_Y} |y\rangle|\Phi_{R \cup \{(x,y)\}}\rangle$$

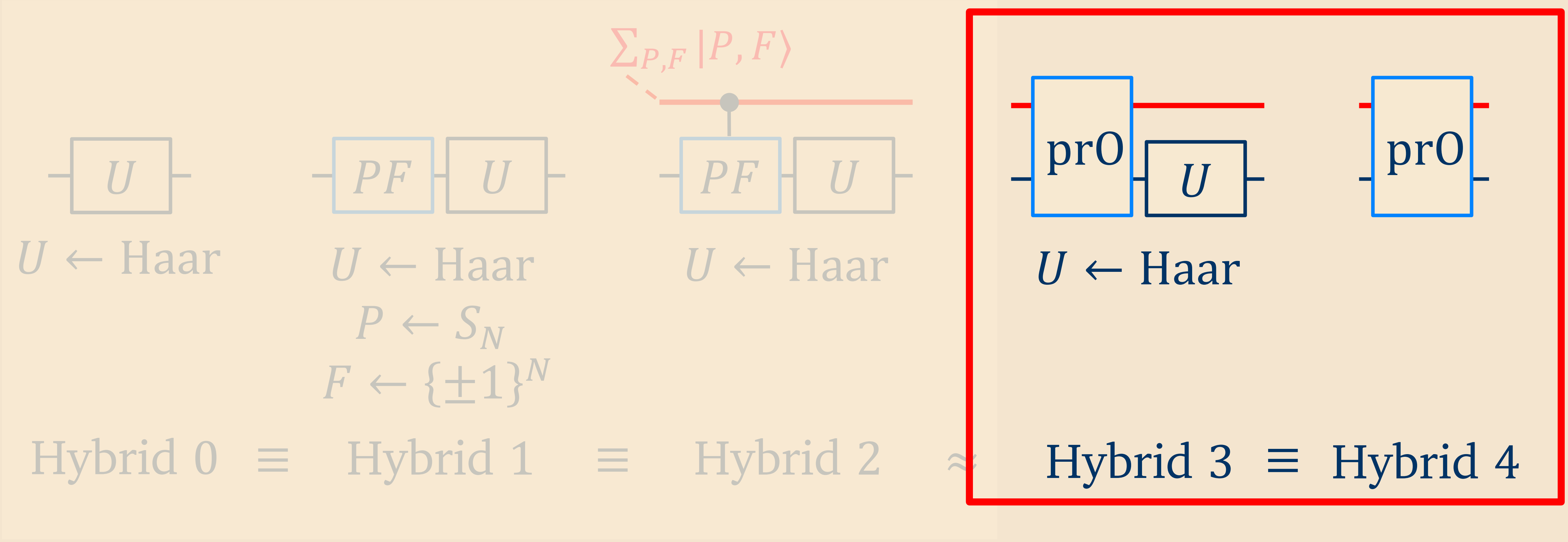
- **Intuition:** ctl-PF behaves like prO, up to relabeling $|\Phi_R\rangle \mapsto |R\rangle$



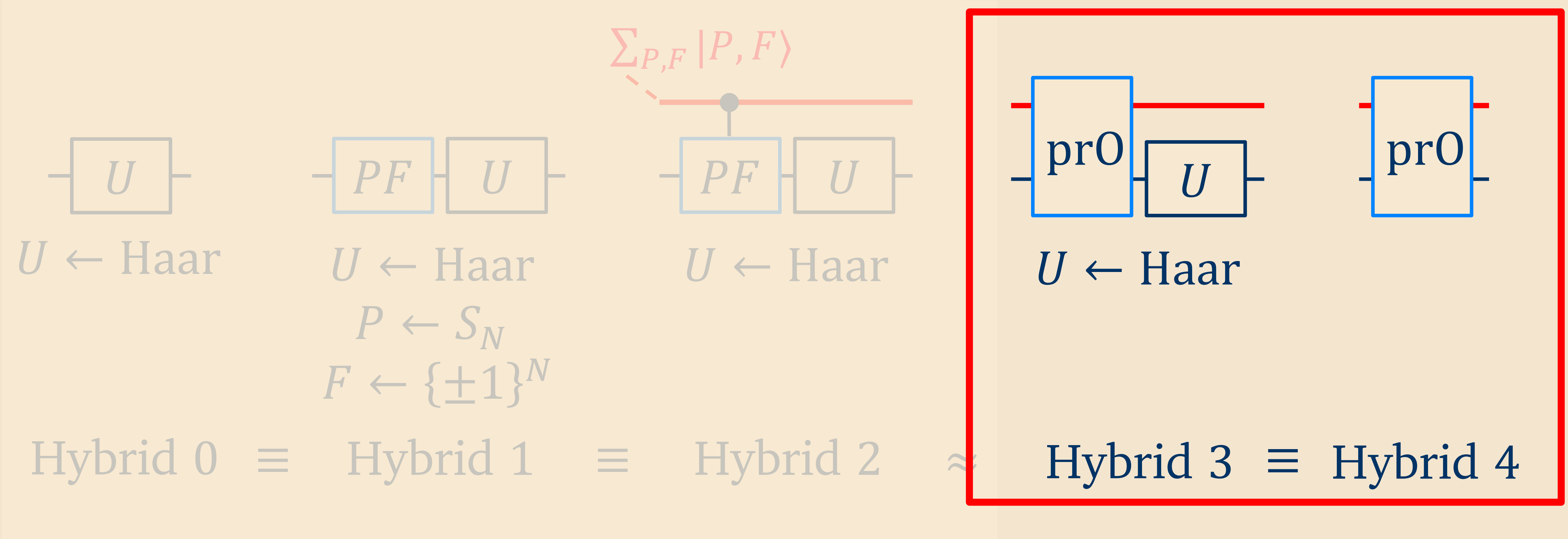
Step 3: For any $R = \{(x_1, y_1), \ldots, (x_t, y_t)\}$ can define $|\Phi_R\rangle$ s.t.

$$\text{ctl-PF} \cdot |x\rangle|\Phi_R\rangle = \sum_{y \notin R_Y} |y\rangle|\Phi_{R \cup \{(x,y)\}}\rangle$$

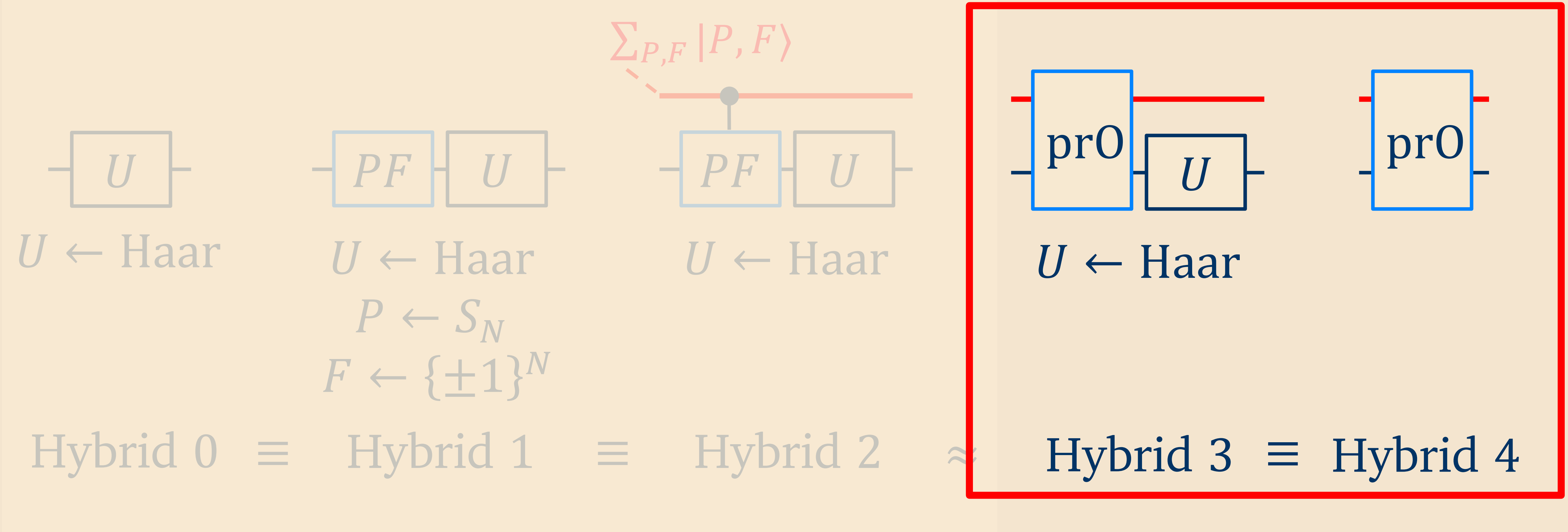
- **Intuition:** ctl-PF behaves like prO, up to relabeling $|\Phi_R\rangle \mapsto |R\rangle$
- Actually, $\{|\Phi_R\rangle\}_R$ aren't fully orthogonal. But composing with $U \leftarrow$ (2-design) makes the "non-orthogonal" ones hard to find.



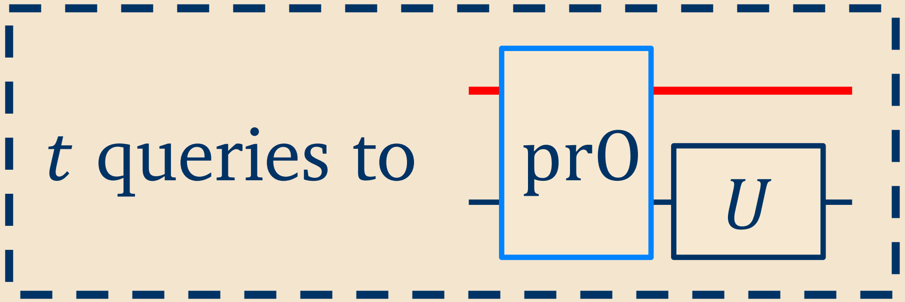
$\sum_{P,F} |P,F\rangle$
U
PF
U
PF
U
prO
U
prO
$U \leftarrow \text{Haar}$
$U \leftarrow \text{Haar}$
$P \leftarrow S_N$
$F \leftarrow \{\pm 1\}^N$
$U \leftarrow \text{Haar}$
$U \leftarrow \text{Haar}$
Hybrid 0 $\equiv$ Hybrid 1 $\equiv$ Hybrid 2 $\approx$ Hybrid 3 $\equiv$ Hybrid 4

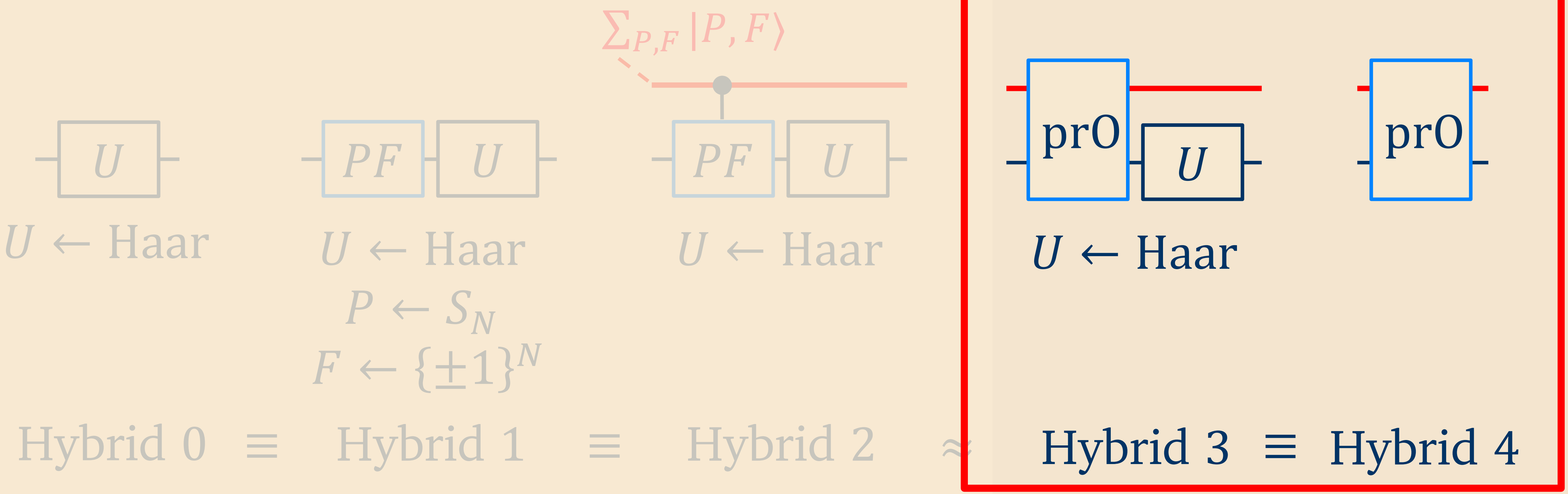


Step 4: Turns out prO has the following magical property:

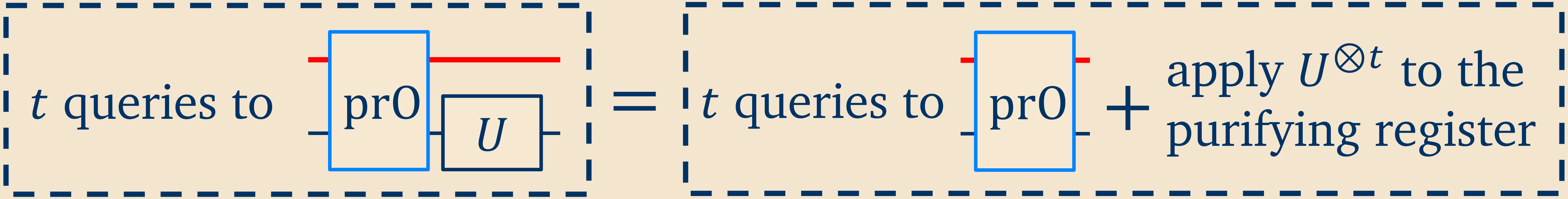


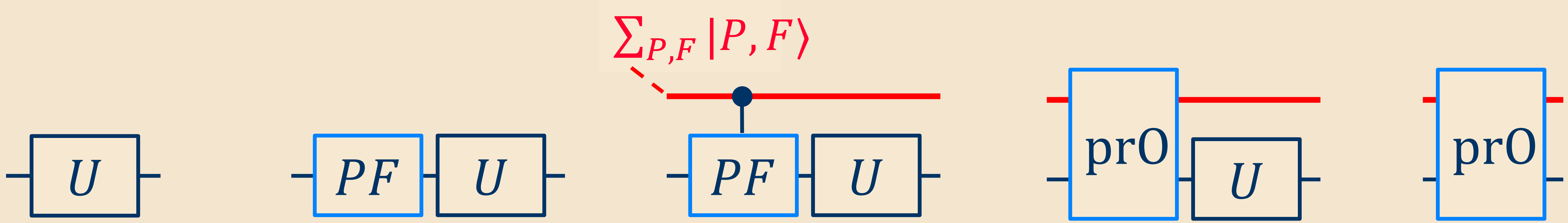
Step 4: Turns out prO has the following magical property:





Step 4: Turns out prO has the following magical property:





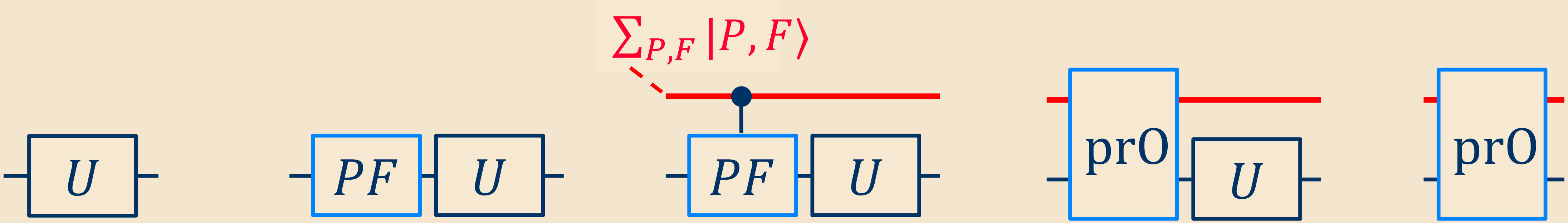
$U \leftarrow \text{Haar}$

$U \leftarrow \text{Haar}$
$P \leftarrow S_N$
$F \leftarrow \{\pm 1\}^N$

$U \leftarrow \text{Haar}$

$U \leftarrow \text{Haar}$

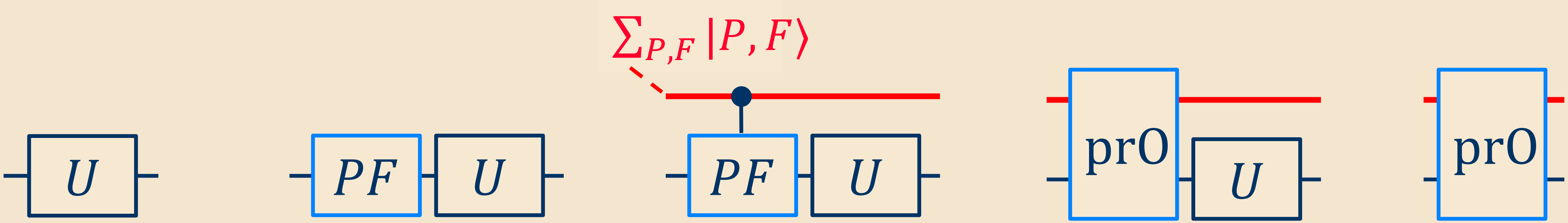
Hybrid 0 $\equiv$ Hybrid 1 $\equiv$ Hybrid 2 $\approx$ Hybrid 3 $\equiv$ Hybrid 4



$U \leftarrow \text{Haar}$ | $U \leftarrow \text{Haar}$, $P \leftarrow S_N$, $F \leftarrow \{\pm 1\}^N$ | $U \leftarrow \text{Haar}$ | $U \leftarrow \text{Haar}$

Hybrid 0 $\equiv$ Hybrid 1 $\equiv$ Hybrid 2 $\approx$ Hybrid 3 $\equiv$ Hybrid 4

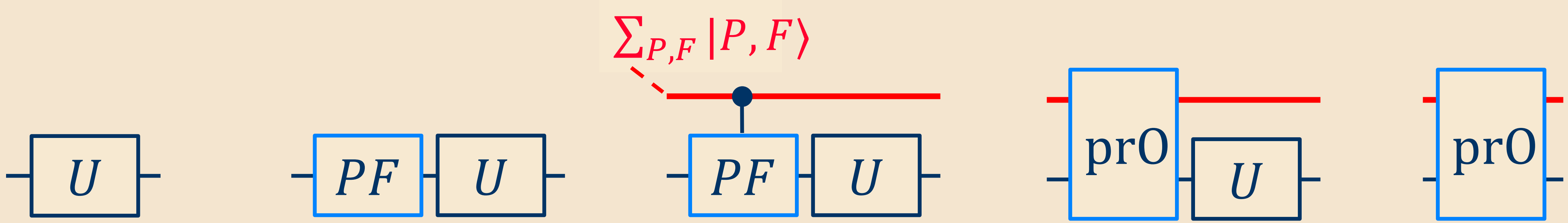
How we get PRUs:



$U \leftarrow \text{Haar}$ | $U \leftarrow \text{Haar}$, $P \leftarrow S_N$, $F \leftarrow \{\pm 1\}^N$ | $U \leftarrow \text{Haar}$ | $U \leftarrow \text{Haar}$

Hybrid 0 $\equiv$ Hybrid 1 $\equiv$ Hybrid 2 $\approx$ Hybrid 3 $\equiv$ Hybrid 4

How we get PRUs: Hybrid 2 $\approx$ Hybrid 4 works for any **2-design**.

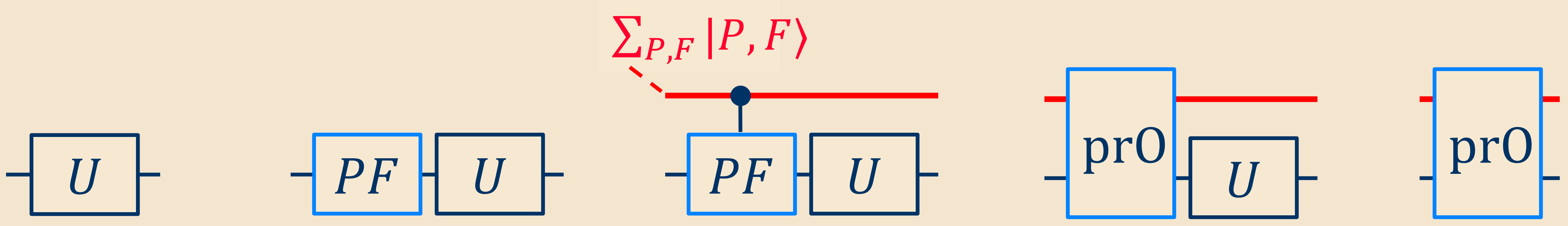


$U \leftarrow \text{Haar}$ | $U \leftarrow \text{Haar}$, $P \leftarrow S_N$, $F \leftarrow \{\pm 1\}^N$ | $U \leftarrow \text{Haar}$ | $U \leftarrow \text{Haar}$

Hybrid 0 $\equiv$ Hybrid 1 $\equiv$ Hybrid 2 $\approx$ Hybrid 3 $\equiv$ Hybrid 4

How we get PRUs: Hybrid 2 $\approx$ Hybrid 4 works for any **2-design**.
So by a triangle inequality:

$$PF \cdot (\text{Clifford C}) \approx PF \cdot (\text{Haar } U)$$



$U \leftarrow$ Haar

$U \leftarrow$ Haar
$P \leftarrow S_N$
$F \leftarrow \{\pm 1\}^N$

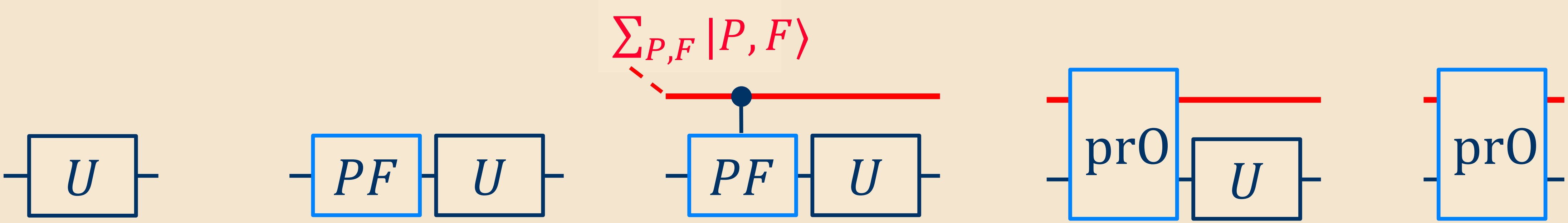
$U \leftarrow$ Haar

$U \leftarrow$ Haar

Hybrid 0 $\equiv$ Hybrid 1 $\equiv$ Hybrid 2 $\approx$ Hybrid 3 $\equiv$ Hybrid 4

How we get PRUs: Hybrid 2 $\approx$ Hybrid 4 works for any **2-design**.
So by a triangle inequality:

$$PF \cdot (\text{Clifford C}) \approx PF \cdot (\text{Haar } U) \equiv \text{Haar } U$$



$U \leftarrow$ Haar

$U \leftarrow$ Haar
$P \leftarrow S_N$
$F \leftarrow \{\pm 1\}^N$

$U \leftarrow$ Haar

$U \leftarrow$ Haar

Hybrid 0 $\equiv$ Hybrid 1 $\equiv$ Hybrid 2 $\approx$ Hybrid 3 $\equiv$ Hybrid 4

How we get PRUs: Hybrid 2 $\approx$ Hybrid 4 works for any **2-design**.
So by a triangle inequality:

$$PF \cdot (\text{Clifford C}) \approx PF \cdot (\text{Haar } U) \equiv \text{Haar } U$$

Finally, replace P and F with pseudorandom.

The path-recording oracle gives a new way to analyze Haar-random unitaries.

The path-recording oracle gives a new way to analyze Haar-random unitaries.

Meta-approach: to prove a statement about algorithms that use a Haar-random U, just prove it with prO!

The path-recording oracle gives a new way to analyze Haar-random unitaries.

Meta-approach: to prove a statement about algorithms that use a Haar-random U, just prove it with prO!

Can be significantly easier than bounding moments of U.

The path-recording oracle gives a new way to analyze Haar-random unitaries.

Meta-approach: to prove a statement about algorithms that use a Haar-random U, just prove it with prO!

Can be significantly easier than bounding moments of U.

Already several applications:

- [MH24]: elementary proof of [SHH24] gluing lemma
- [ABGL24]: compress PRU key length + other results

PRU future directions

PRU future directions

- **Complexity:** unitary natural proofs barrier?

PRU future directions

- **Complexity:** unitary natural proofs barrier?
- **Math:** implications for random matrix theory?

PRU future directions

- **Complexity:** unitary natural proofs barrier?
- **Math:** implications for random matrix theory?
- **Physics:** are random circuits PRUs?

PRU future directions

- **Complexity:** unitary natural proofs barrier?
- **Math:** implications for random matrix theory?
- **Physics:** are random circuits PRUs?
- **Cryptography:**
  - relationship to PRPs?
  - applications to uncloneable crypto?
  - PRUs without one-way functions?

PRU future directions

- **Complexity:** unitary natural proofs barrier?
- **Math:** implications for random matrix theory?
- **Physics:** are random circuits PRUs?
- **Cryptography:**
  - relationship to PRPs?
  - applications to uncloneable crypto?
  - PRUs without one-way functions?

Thanks!