Post-Quantum Proof Techniques, Part 2:

How to Run a Quantum Attacker Many Times

Fermi Ma

(Simons & Berkeley)

Based on:

- “Post-Quantum Succinct Arguments: Breaking the Quantum Rewinding Barrier” by Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry (FOCS 2021)

Last time, we used Unruh's lemma to show that Blum (w/ collapse-binding commitments) is a post-quantum proof of knowledge.

Last time, we used Unruh's lemma to show that Blum (w/ collapse-binding commitments) is a post-quantum proof of knowledge.

However, this technique has some drawbacks:

Last time, we used Unruh's lemma to show that Blum (w/ collapse-binding commitments) is a post-quantum proof of knowledge.

However, this technique has some drawbacks:

- We get weaker-than-expected security

Last time, we used Unruh's lemma to show that Blum (w/ collapse-binding commitments) is a post-quantum proof of knowledge.

However, this technique has some drawbacks:

- We get weaker-than-expected security (e.g., can only extract when quantum P^* convinces V with probability $1/\sqrt{2} + \varepsilon$)

Last time, we used Unruh's lemma to show that Blum (w/ collapse-binding commitments) is a post-quantum proof of knowledge.

However, this technique has some drawbacks:

- We get weaker-than-expected security (e.g., can only extract when quantum P^* convinces V with probability $1/\sqrt{2} + \varepsilon$)
- More serious issue: only works for a *very limited class* of protocols (e.g., Blum but not [GMW86] graph 3-coloring)

Last time, we used Unruh's lemma to show that Blum (w/ collapse-binding commitments) is a post-quantum proof of knowledge.

However, this technique has some drawbacks:

- We get weaker-than-expected security (e.g., can only extract when quantum P^* convinces V with probability $1/\sqrt{2} + \varepsilon$)
- More serious issue: only works for a *very limited class* of protocols (e.g., Blum but not [GMW86] graph 3-coloring)

This talk: we'll see a significantly more powerful rewinding technique of [CMSZ21].

What this talk will cover:

What this talk will cover:

1. Motivating example: Kilian’s succinct arguments for NP

What this talk will cover:

1. Motivating example: Kilian’s succinct arguments for NP
2. Why is post-quantum security of Kilian difficult?

What this talk will cover:

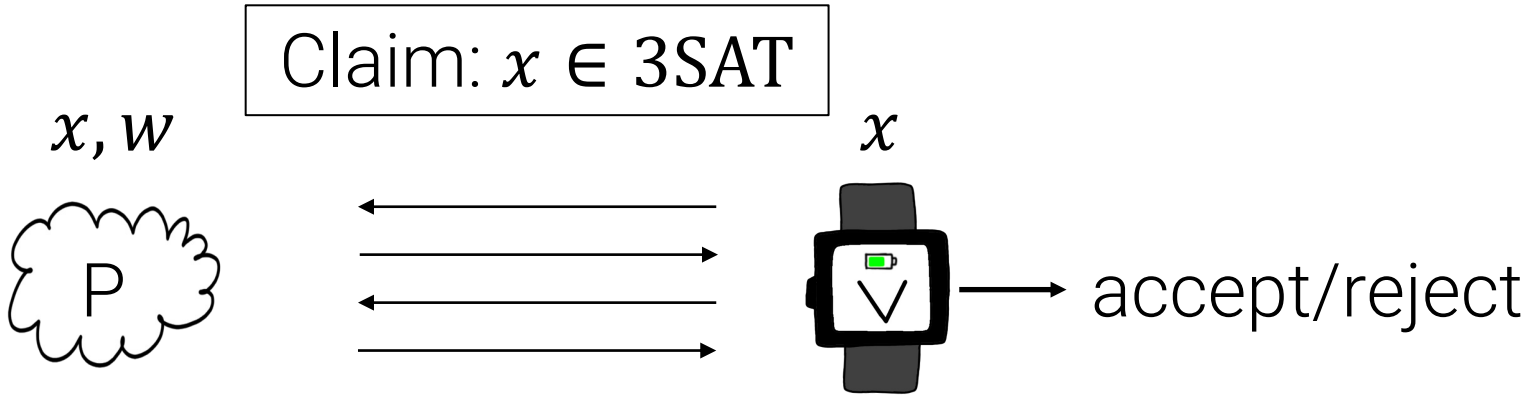
1. Motivating example: Kilian's succinct arguments for NP
2. Why is post-quantum security of Kilian difficult?
3. Rewinding a quantum attacker many times
   - New idea: "repair" the adversary after each query
   - Estimating success probability
   - The full rewinding procedure
   - Analysis

What this talk will cover:

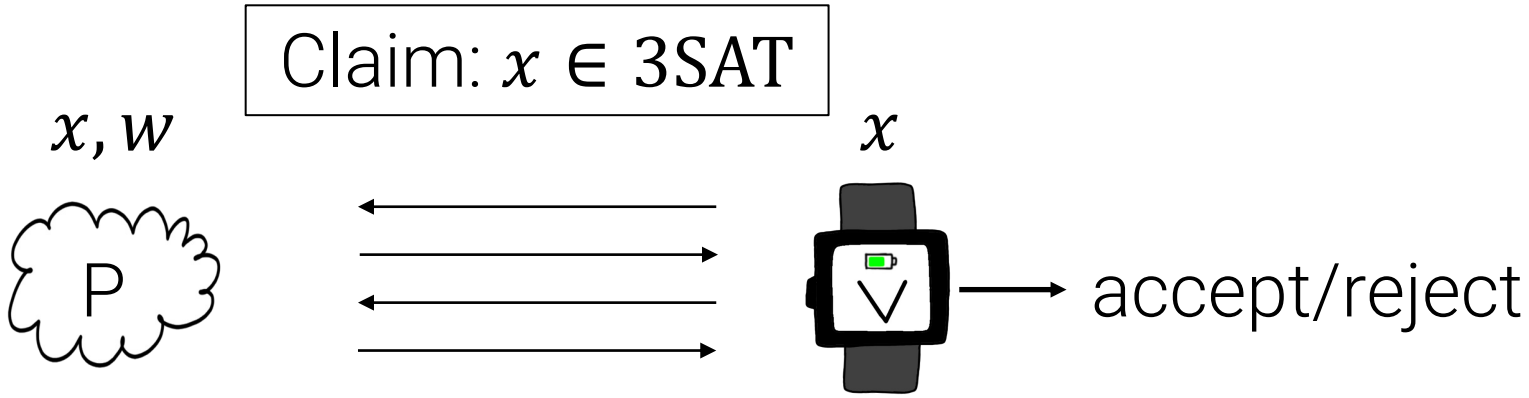
1. Motivating example: Kilian's succinct arguments for NP
2. Why is post-quantum security of Kilian difficult?
3. Rewinding a quantum attacker many times
   - New idea: "repair" the adversary after each query
   - Estimating success probability
   - The full rewinding procedure
   - Analysis

Motivating example:
Succinct Arguments for NP

Motivating example:
Succinct Arguments for NP

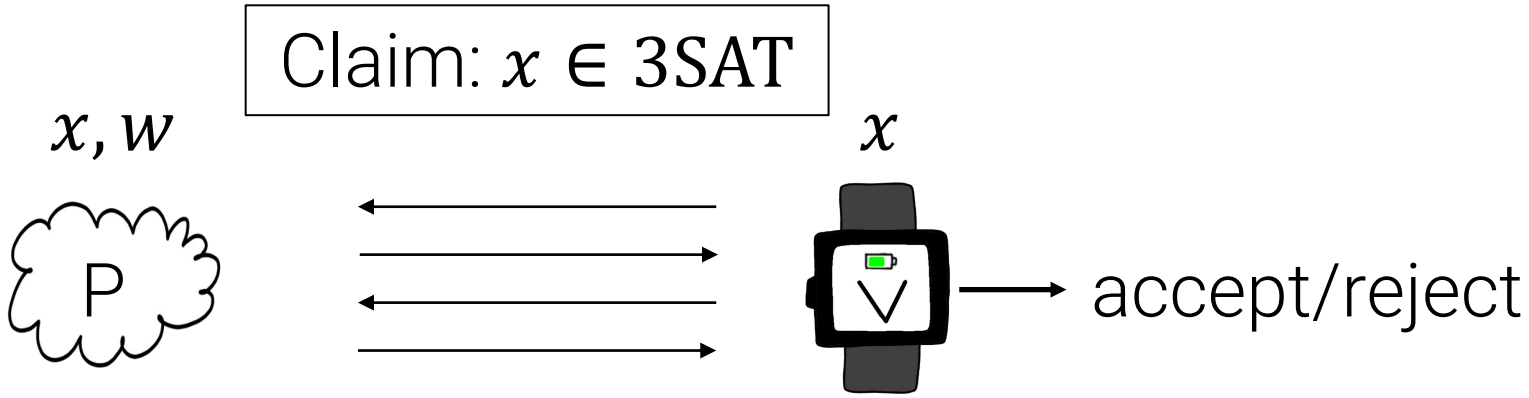


Motivating example:
Succinct Arguments for NP



"Succinct" = communication + verifier efficiency is $\mathrm{poly}(\lambda, \log(|x| + |w|))$

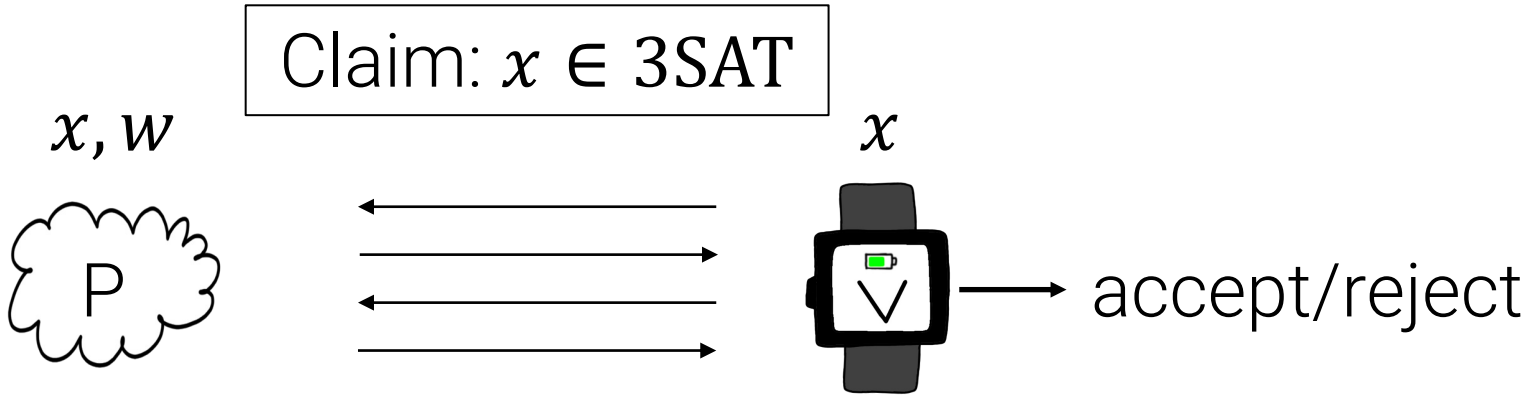
Motivating example:
Succinct Arguments for NP



"Succinct" = communication + verifier efficiency is $\mathrm{poly}(\lambda, \log(|x| + |w|))$

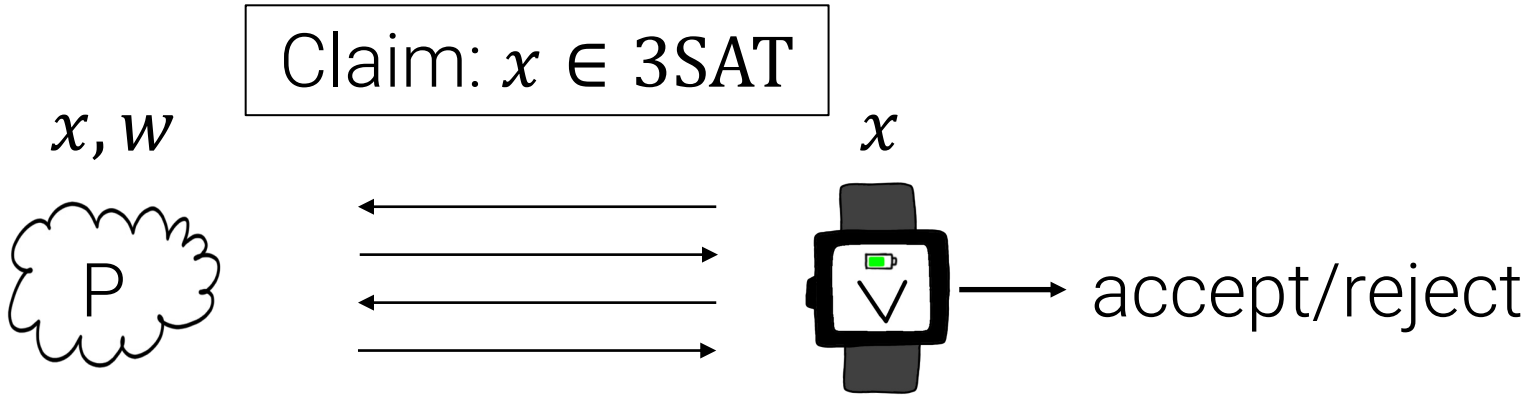
"Argument" = sound against ***efficient*** cheating 

Motivating example:
Succinct Arguments for NP



[Kilian92] constructs a 4-message succinct argument for NP from collision-resistant hash functions (CRHFs).

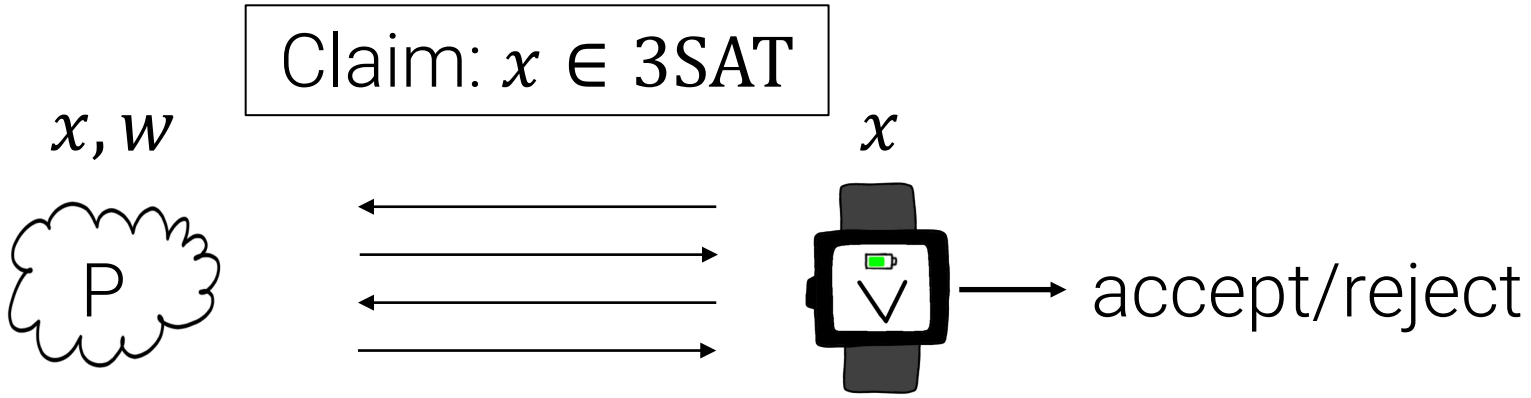
Motivating example:
Succinct Arguments for NP



[Kilian92] constructs a 4-message succinct argument for NP from collision-resistant hash functions (CRHFs).

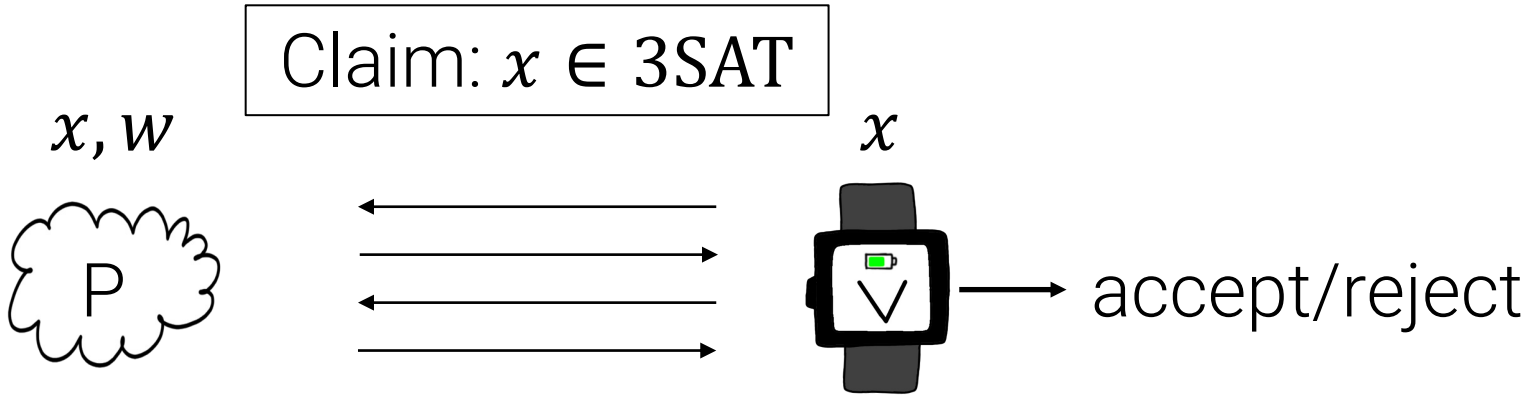
Many applications: universal arguments [BG01], zero knowledge [Barak01], SNARGs [Micali94, BCS16], …

Motivating example:
Succinct Arguments for NP



Extra motivation: studying quantum rewinding for succinct arguments will force us to develop general-purpose techniques.

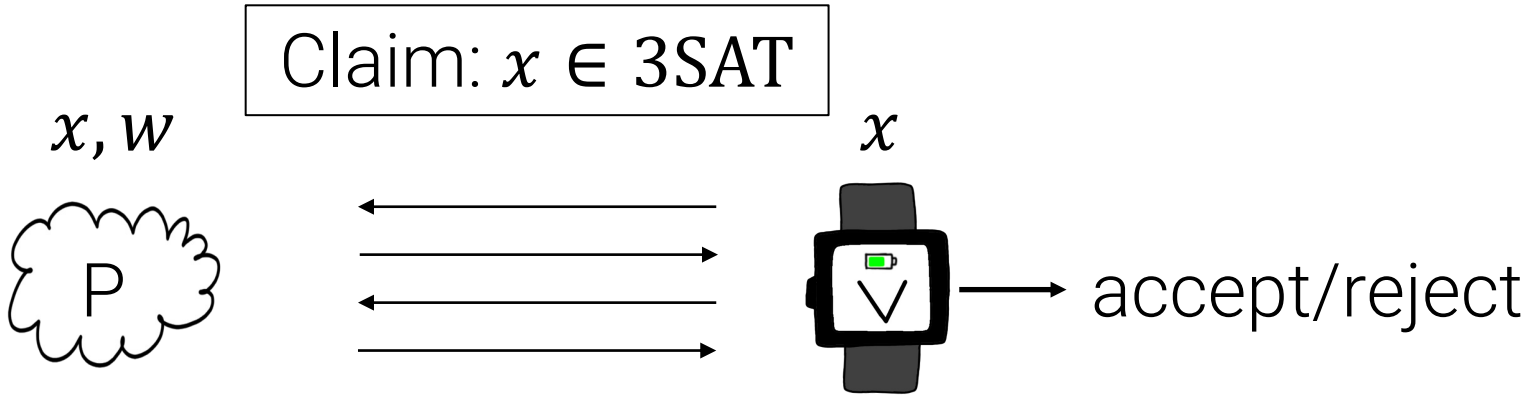
Motivating example:
Succinct Arguments for NP



Extra motivation: studying quantum rewinding for succinct arguments will force us to develop general-purpose techniques.

- Typically prove soundness using several transcripts to specify a witness.

Motivating example:
Succinct Arguments for NP



Extra motivation: studying quantum rewinding for succinct arguments will force us to develop general-purpose techniques.

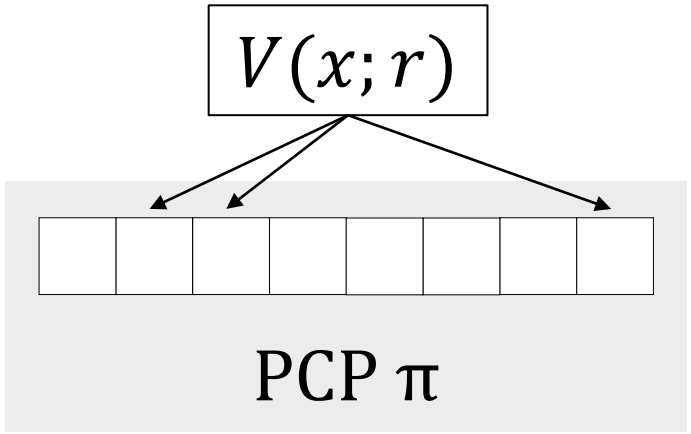
- Typically prove soundness using several transcripts to specify a witness.
- Succinct arguments inherently require many transcripts to specify a witness, so *lots* of rewinding is required.

Let’s see how Kilian’s protocol works

Kilian's protocol

Compile a ***probabilistically checkable proof**** (PCP) into an interactive argument system using cryptography.

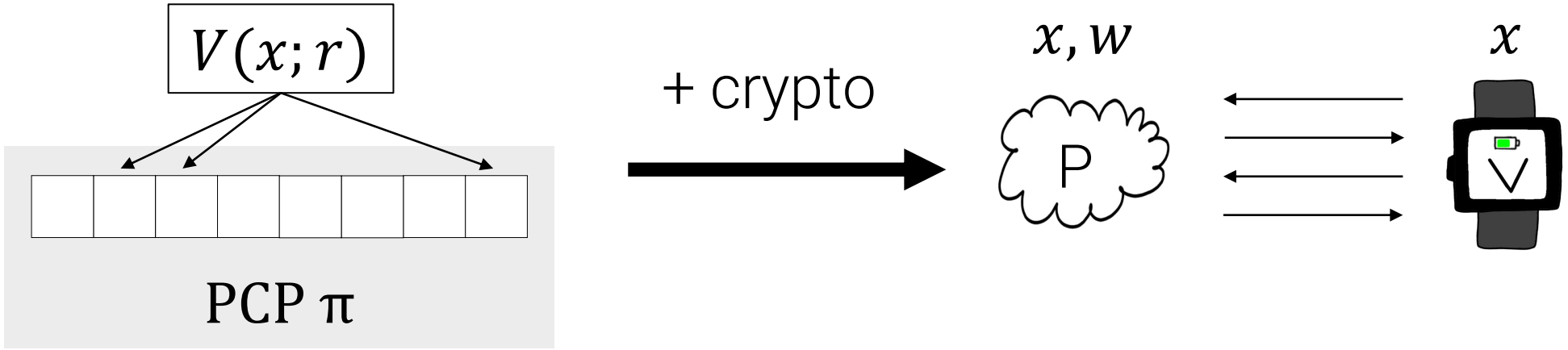
*[BFLS91,FGLSS91,AS92,ALMSS92]



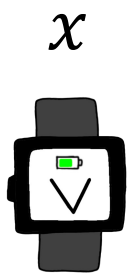
Kilian's protocol

Compile a ***probabilistically checkable proof**** (PCP) into an interactive argument system using cryptography.

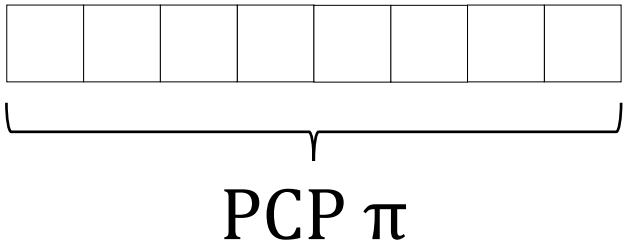
*[BFLS91,FGLSS91,AS92,ALMSS92]

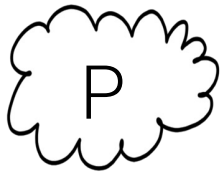


Kilian's protocol



Encode w as PCP π

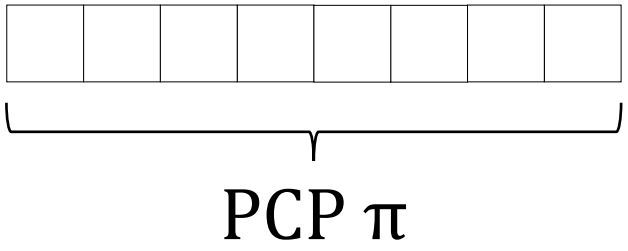


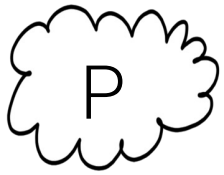


sends short commitment to PCP π.

Kilian's protocol

Encode w as PCP π

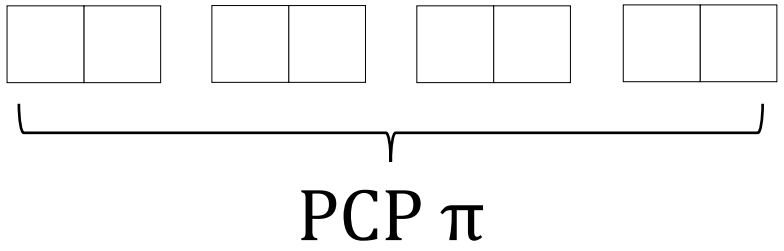


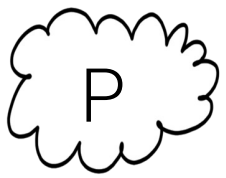


sends short commitment to PCP π.

Kilian's protocol

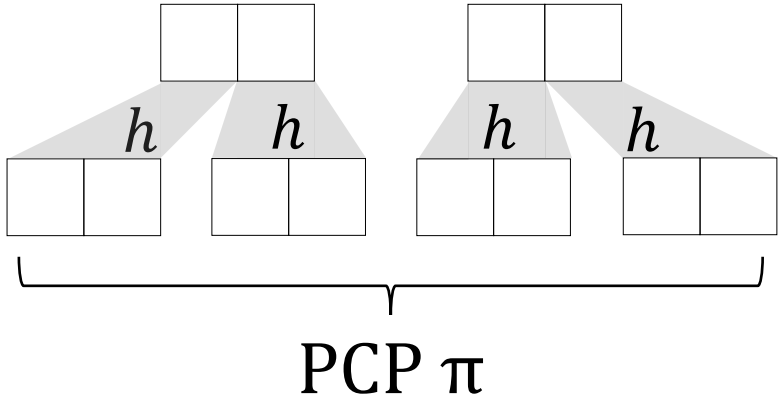
Encode w as PCP π

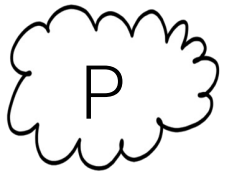




sends short commitment to PCP π.

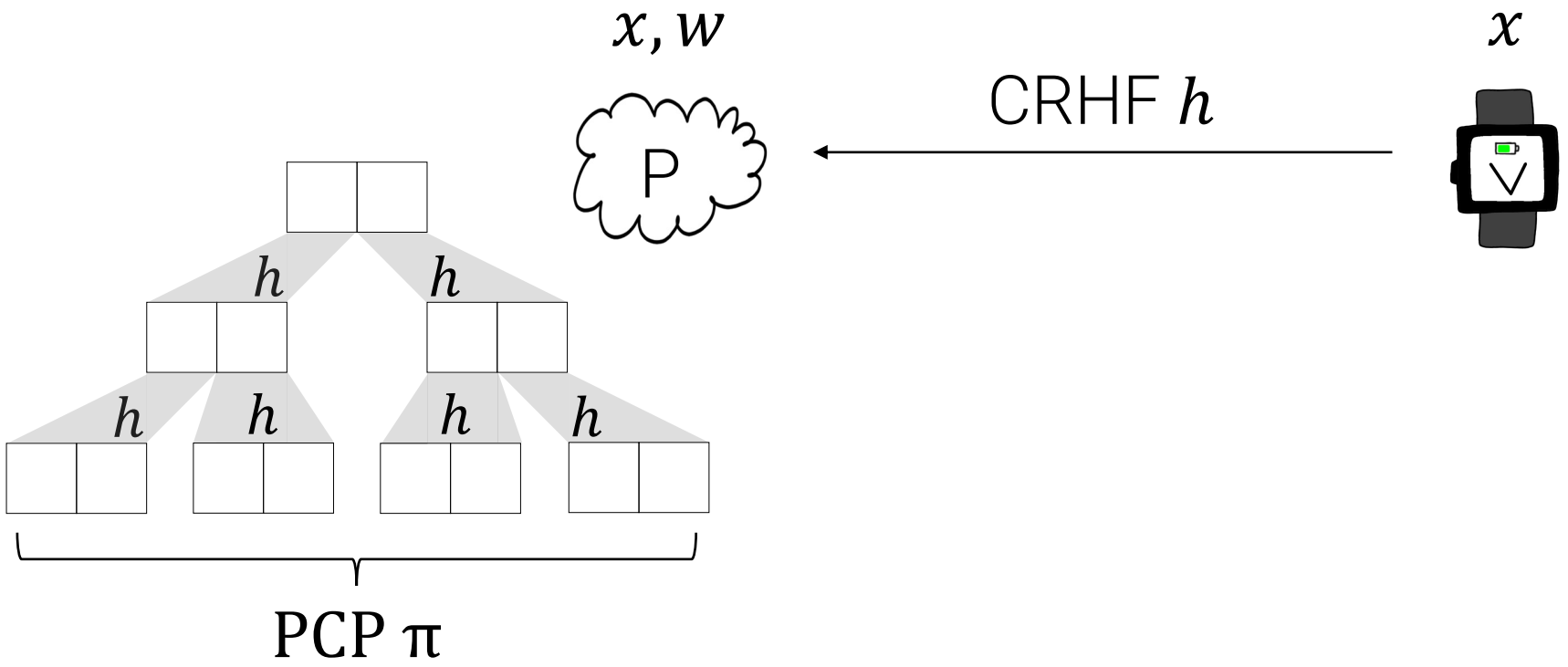
Kilian's protocol

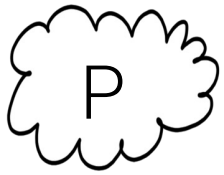




sends short commitment to PCP π.

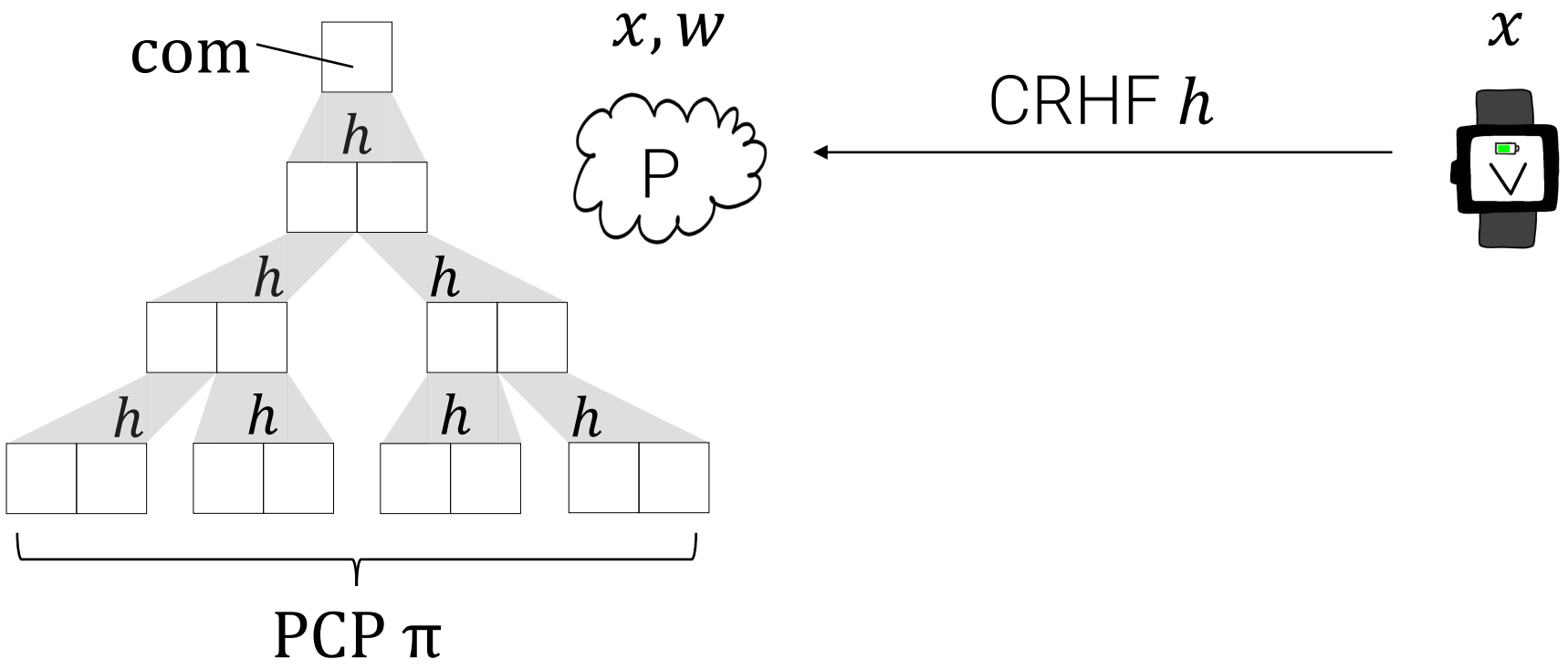
Kilian's protocol





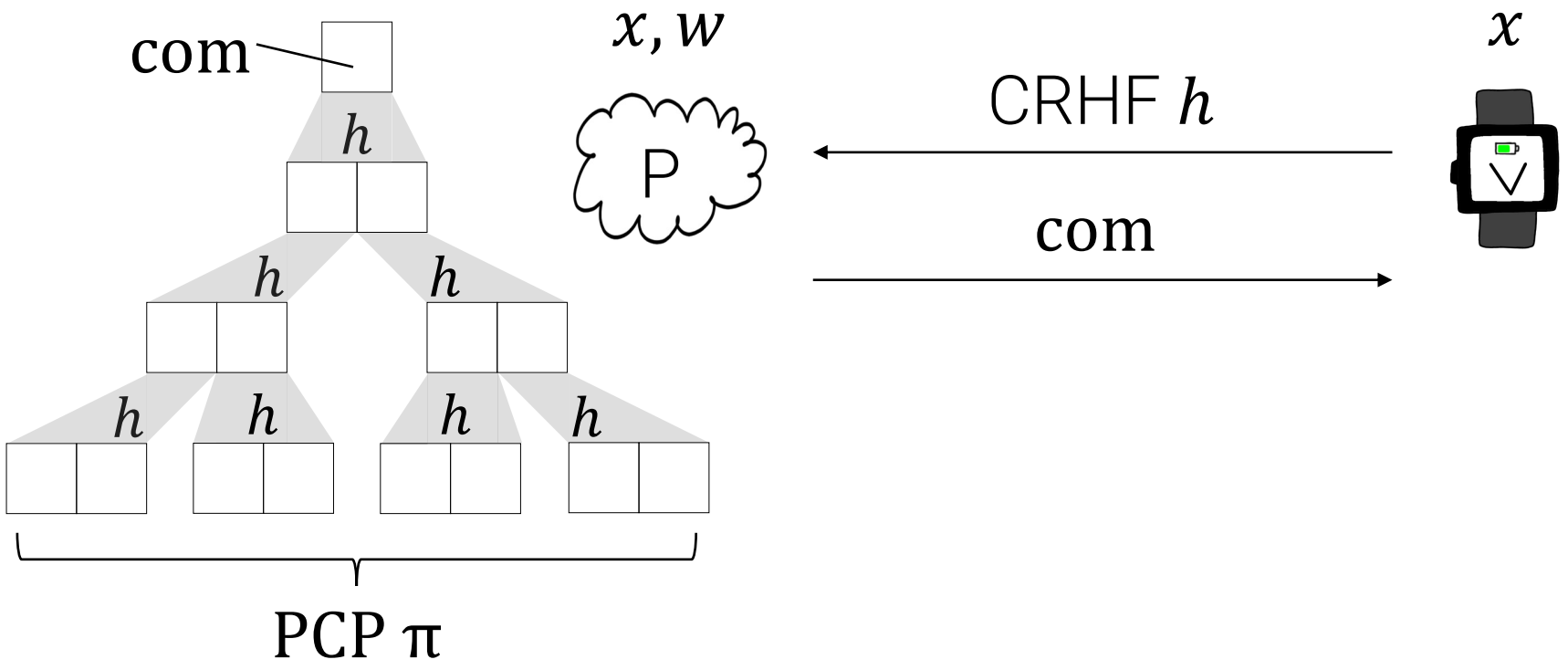
sends short commitment to PCP π.

Kilian's protocol



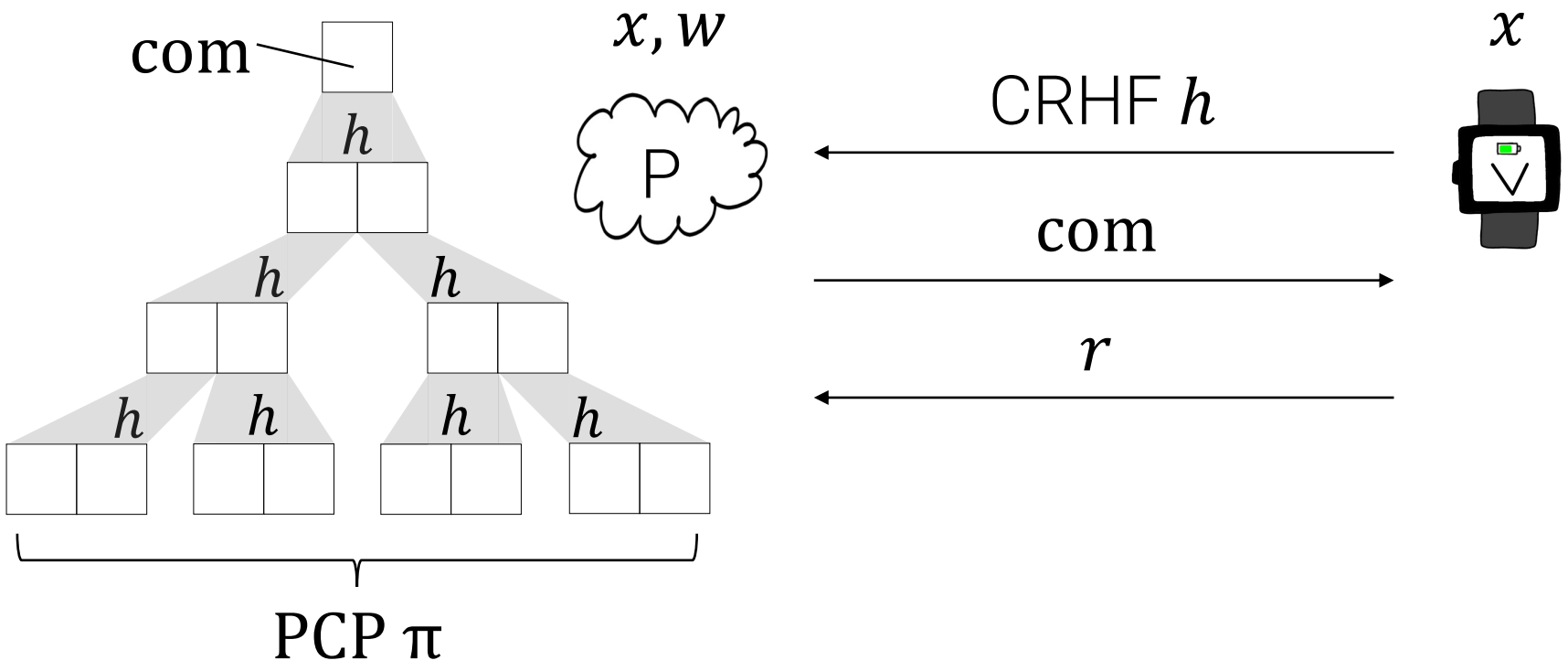
P sends short commitment to PCP π.

Kilian's protocol



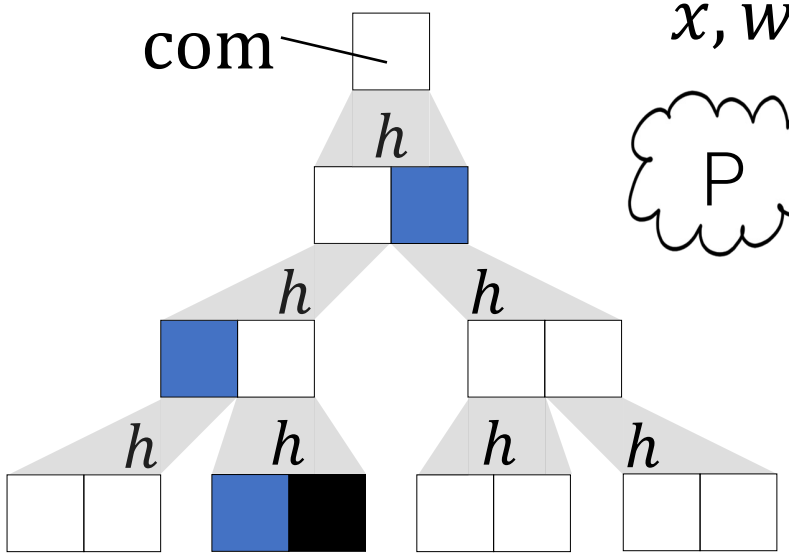
P sends short commitment to PCP π.

Kilian's protocol



samples PCP verifier coins $r \leftarrow R$.

Kilian's protocol



$\pi[Q_r]$

$\text{open}[Q_r] =$ 

x, w

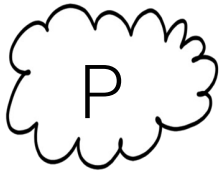
x

CRHF h

com

r

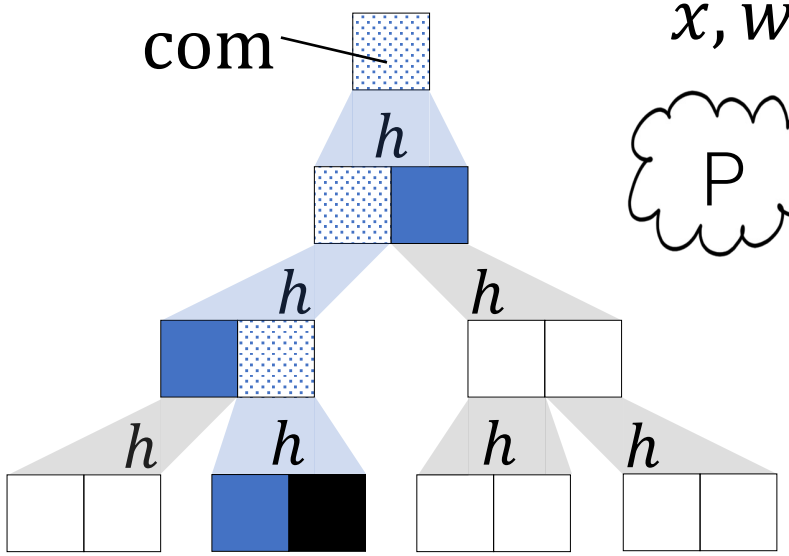
$\pi[Q_r], \text{open}[Q_r]$



sends $\pi[Q_r]$ + opening proofs

Q_r = indices PCP verifier checks on random coins r

Kilian's protocol



$\pi[Q_r]$

$\text{open}[Q_r] =$ 

x, w

x

CRHF h

com

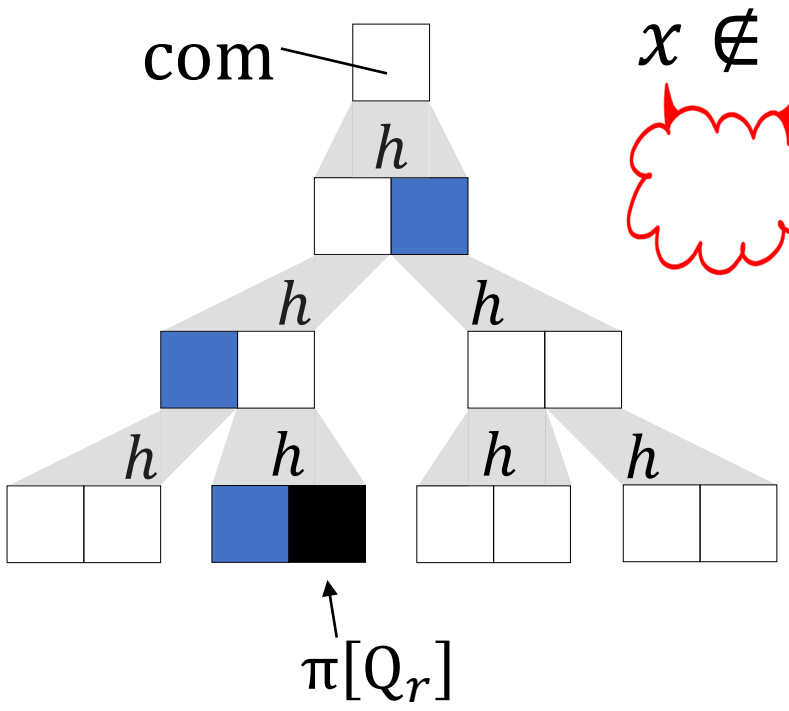
r

$\pi[Q_r], \text{open}[Q_r]$

accept
or reject

accepts if openings valid
+ PCP verifier accepts

Classical Security



CRHF h

com

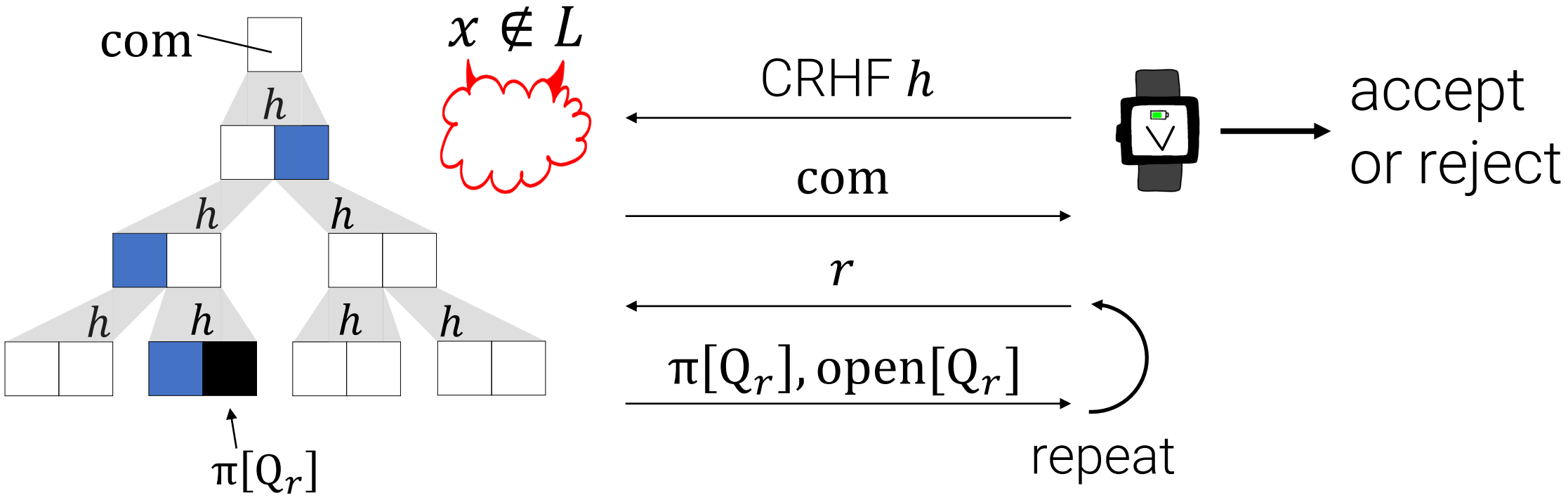
r

$\pi[Q_r], \text{open}[Q_r]$

accept
or reject

Intuition: want to show that the CRHF forces  to respond consistently with some PCP string π.

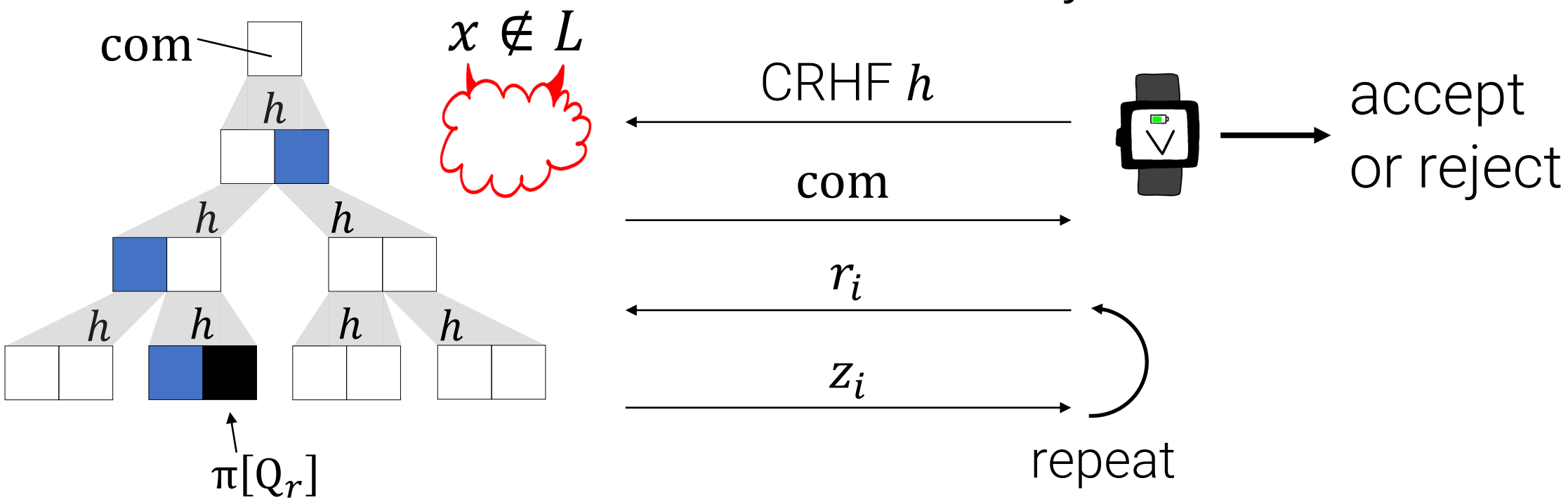
Classical Security



Intuition: want to show that the CRHF forces  to respond consistently with some PCP string π.

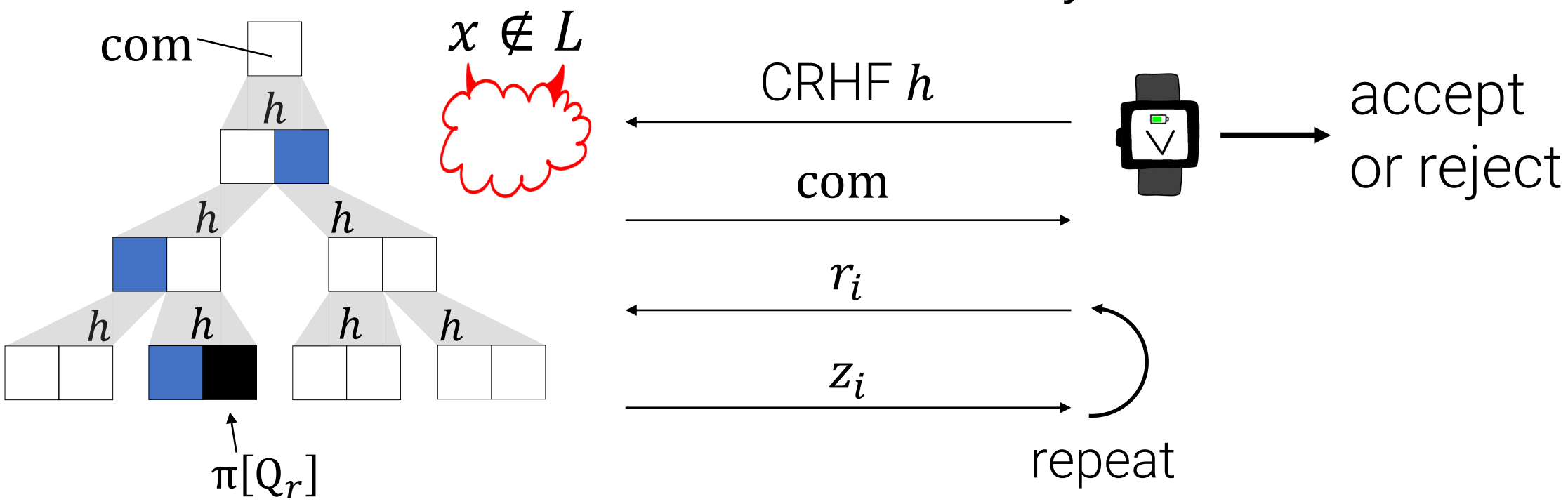
Formalize by ***rewinding*** last two messages many times.

Classical Security



Reduction's goal: record ***many*** accepting transcripts (r_i, z_i)

Classical Security



Reduction's goal: record ***many*** accepting transcripts (r_i, z_i)

Eventually finds impossible π OR collision.

Pr[PCP verifier accepts π] > PCP soundness error

What this talk will cover:

1. Motivating example: Kilian's succinct arguments for NP
2. Why is post-quantum security of Kilian difficult?
3. Rewinding a quantum attacker many times
   - New idea: "repair" the adversary after each query
   - Estimating success probability
   - The full rewinding procedure
   - Analysis

Recall the rewinding template from last talk:

Step 1: Collapse-binding commitments [U16]:

record adversary's response $\approx_c$ record 1-bit decision

+

Step 2: 1-bit-rewinding [U12,DFMS19]:

If we run a p-successful adversary on d random queries,

$$\Pr[\text{succeed } d \text{ times}] \geq p^{2d-1}$$

Recall the rewinding template from last talk:

Step 1: Collapse-binding commitments [U16]:

record adversary's response $\approx_c$ record 1-bit decision

+

Step 2: 1-bit-rewinding [U12,DFMS19]:

If we run a p-successful adversary on d random queries,

$$\Pr[\text{succeed } d \text{ times}] \geq p^{2d-1}$$

Step 1 works if the CRHF is a "collapsing" hash function [U16].

Recall the rewinding template from last talk:

Step 1: Collapse-binding commitments [U16]:

record adversary's response $\approx_c$ record 1-bit decision

+

Step 2: 1-bit-rewinding [U12,DFMS19]:

If we run a p-successful adversary on d random queries,

$$\Pr[\text{succeed } d \text{ times}] \geq p^{2d-1}$$

Step 1 works if the CRHF is a "collapsing" hash function [U16].

Aside: Do collapsing hash functions exist?

Yes! Can be built assuming learning with errors (LWE) [U16b] or many other assumptions [Z22].

Recall the rewinding template from last talk:

Step 1: Collapse-binding commitments [U16]:

record adversary's response $\approx_c$ record 1-bit decision

+

Step 2: 1-bit-rewinding [U12,DFMS19]:

If we run a p-successful adversary on d random queries,

$$\Pr[\text{succeed } d \text{ times}] \geq p^{2d-1}$$

✓

Recall the rewinding template from last talk:

Step 1: Collapse-binding commitments [U16]:

record adversary's response $\approx_c$ record 1-bit decision

+

Step 2: 1-bit-rewinding [U12,DFMS19]:

If we run a p-successful adversary on d random queries,

$$\Pr[\text{succeed } d \text{ times}] \geq p^{2d-1}$$

✓

???

Unfortunately, the 1-bit-rewinding lemma isn't enough.

Recall the rewinding template from last talk:

Step 1: Collapse-binding commitments [U16]:

record adversary's response $\approx_c$ record 1-bit decision

+

Step 2: 1-bit-rewinding [U12,DFMS19]:

If we run a p-successful adversary on d random queries,

$$\Pr[\text{succeed } d \text{ times}] \geq p^{2d-1}$$

✓

???

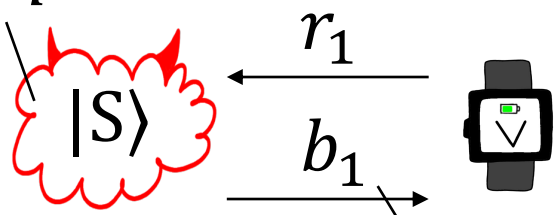
Unfortunately, the 1-bit-rewinding lemma isn't enough.

- classical reduction needs adversary to succeed $\mathbf{poly}(\lambda)$ times.
- But p^{2d-1} is only a noticeable probability when d is *constant*!

What this talk will cover:

1. Motivating example: Kilian's succinct arguments for NP
2. Why is post-quantum security of Kilian difficult?
3. Rewinding a quantum attacker many times
   - New idea: "repair" the adversary after each query
   - Estimating success probability
   - The full rewinding procedure
   - Analysis

success
prob p

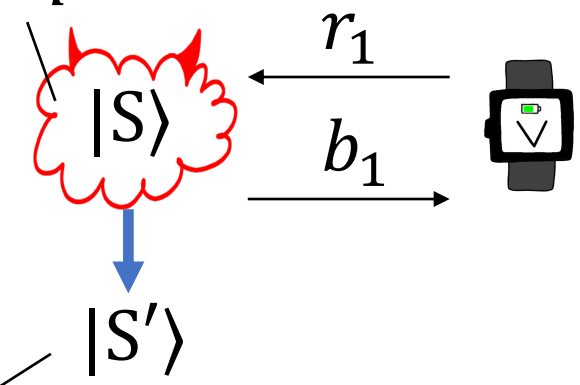


r_1

b_1

1-bit result of measuring
$(\Pi_{r_1}, \mathbb{I} - \Pi_{r_1})$

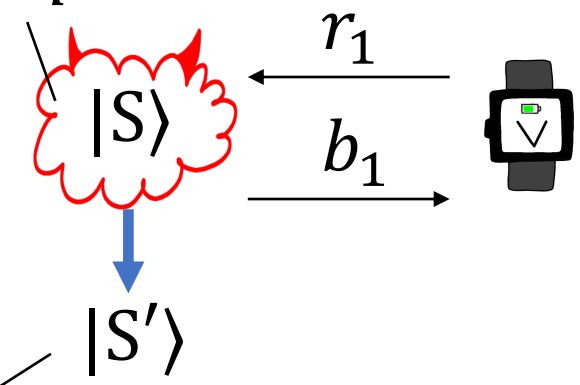
success
prob p



success
prob $\ll p$

Problem: $|S'\rangle$ might not be a successful adversary!

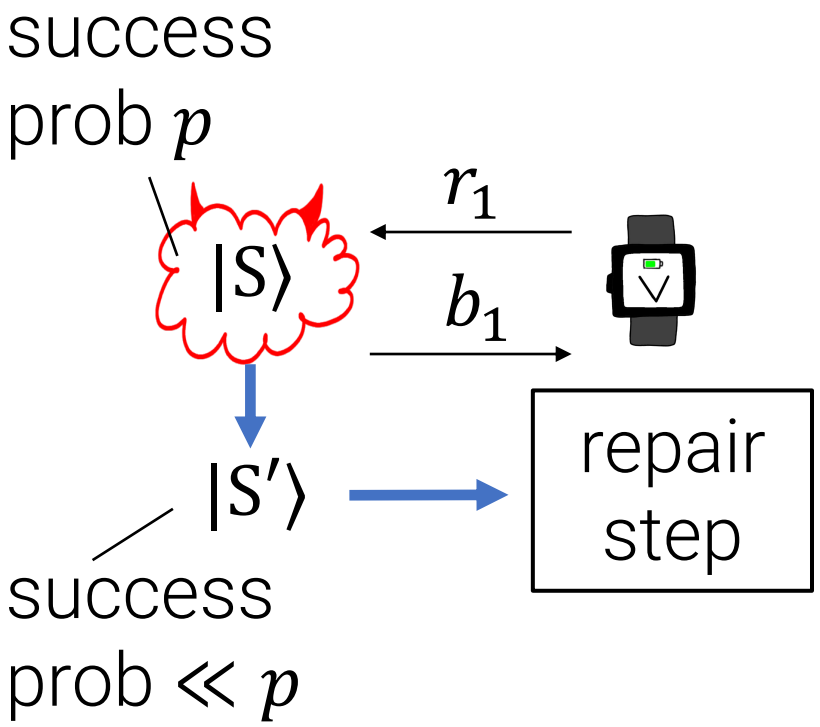
success
prob p



success
prob $\ll p$

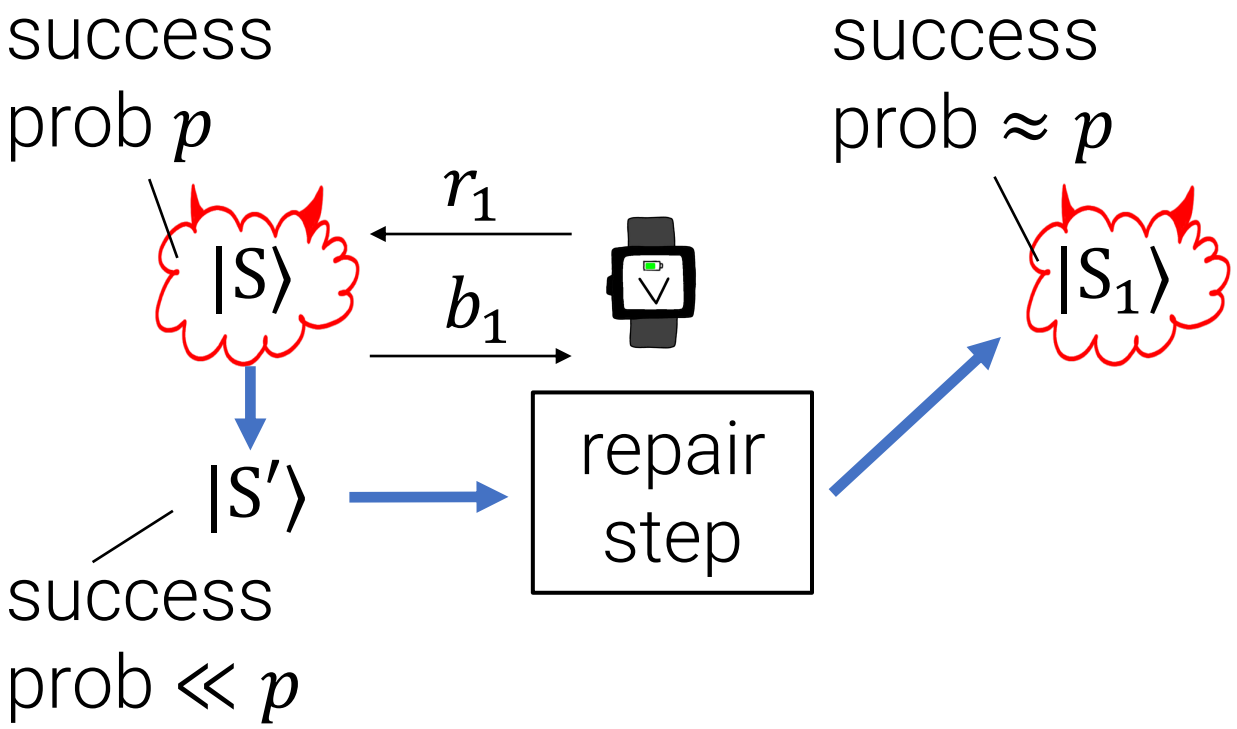
Problem: $|S'\rangle$ might not be a successful adversary!

[CMSZ21]: design a "repair" procedure to restore the original success probability.



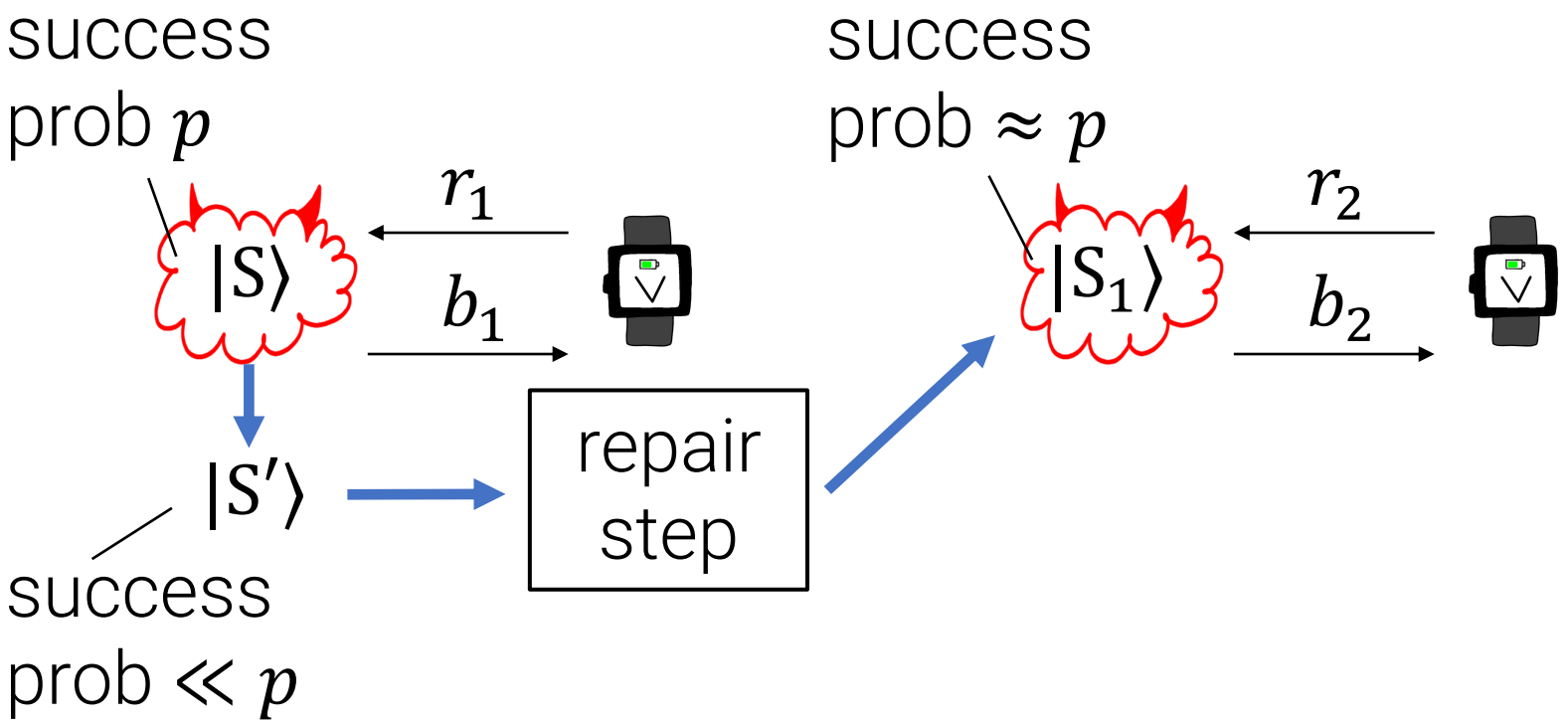
Problem: $|S'\rangle$ might not be a successful adversary!

[CMSZ21]: design a "repair" procedure to restore the original success probability.



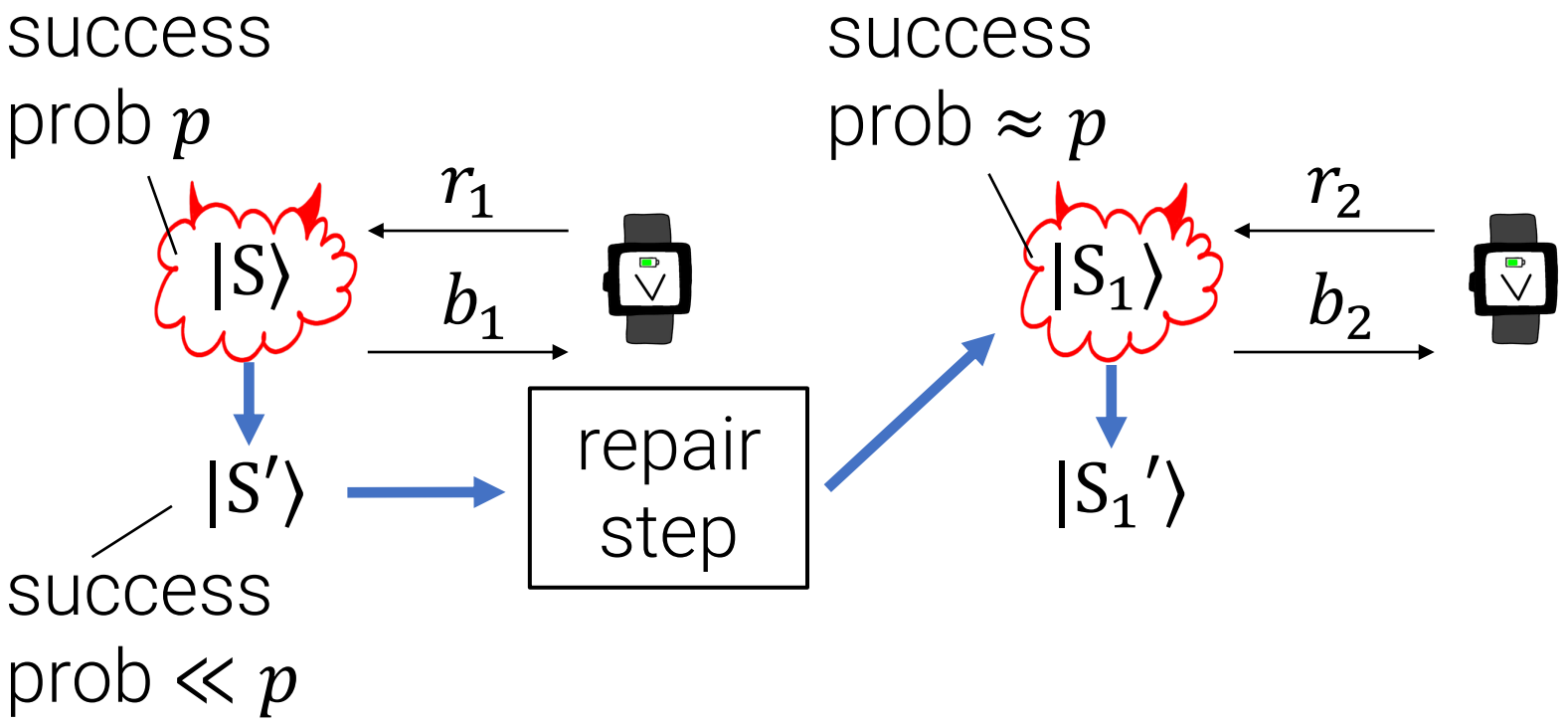
Problem: $|S'\rangle$ might not be a successful adversary!

[CMSZ21]: design a "repair" procedure to restore the original success probability.



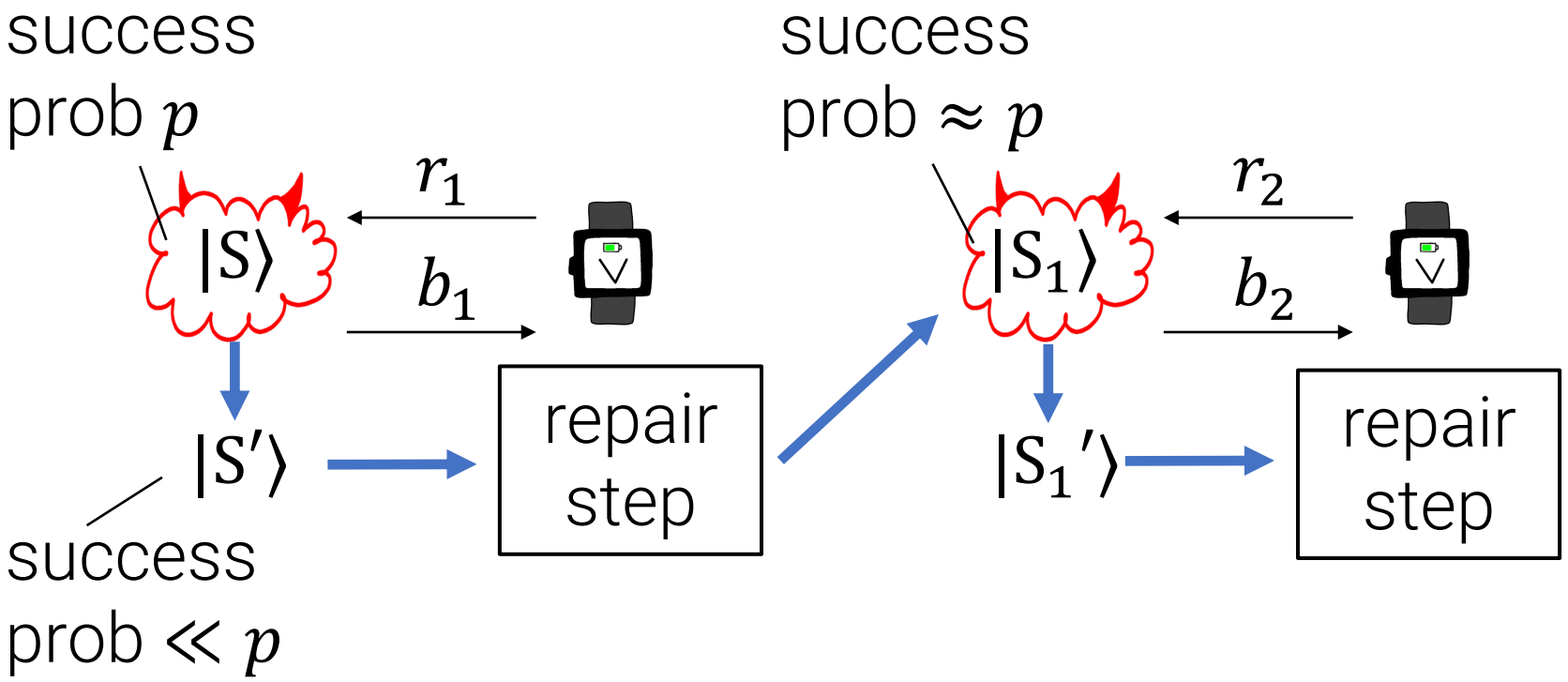
Problem: $|S'\rangle$ might not be a successful adversary!

[CMSZ21]: design a "repair" procedure to restore the original success probability.



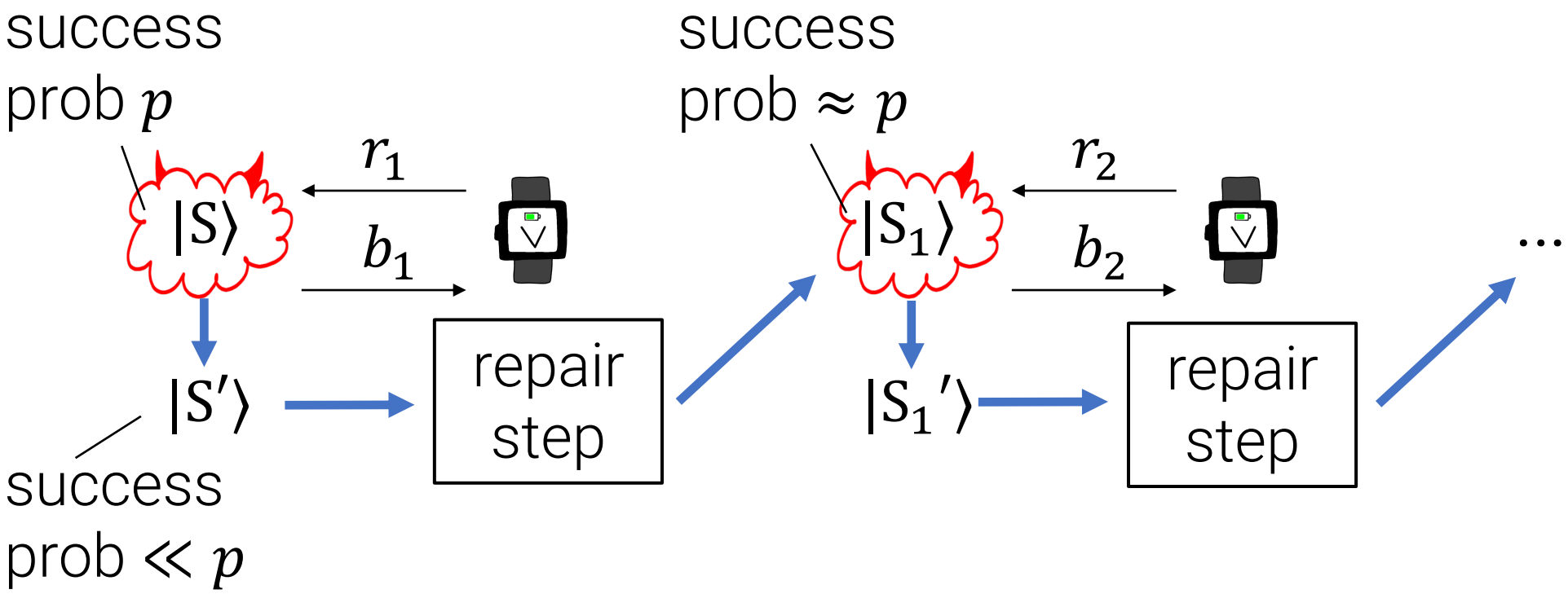
Problem: $|S'\rangle$ might not be a successful adversary!

[CMSZ21]: design a "repair" procedure to restore the original success probability.



Problem: $|S'\rangle$ might not be a successful adversary!

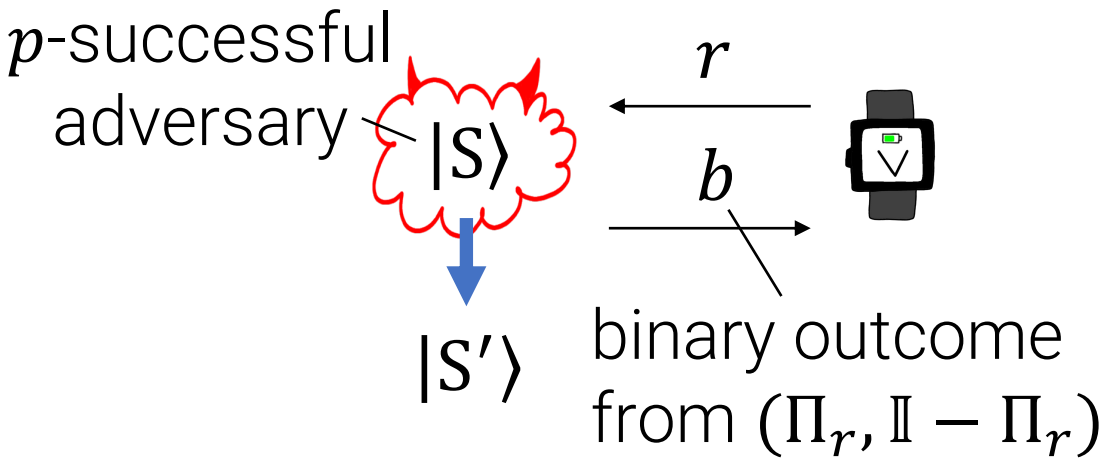
[CMSZ21]: design a "repair" procedure to restore the original success probability.



Problem: $|S'\rangle$ might not be a successful adversary!

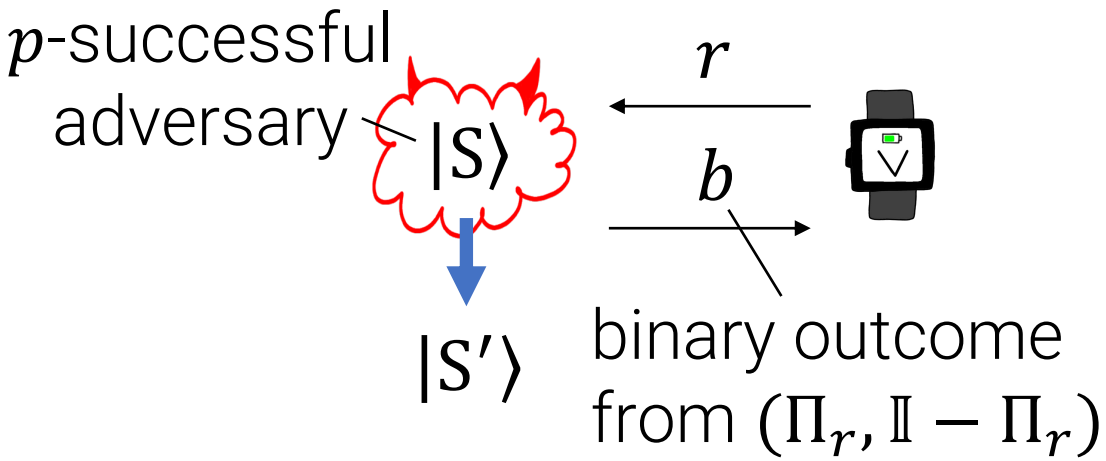
[CMSZ21]: design a "repair" procedure to restore the original success probability.

Repairing the Prover After Measurement



State repair task: Given $|S'\rangle$, generate a p-successful adversary state.

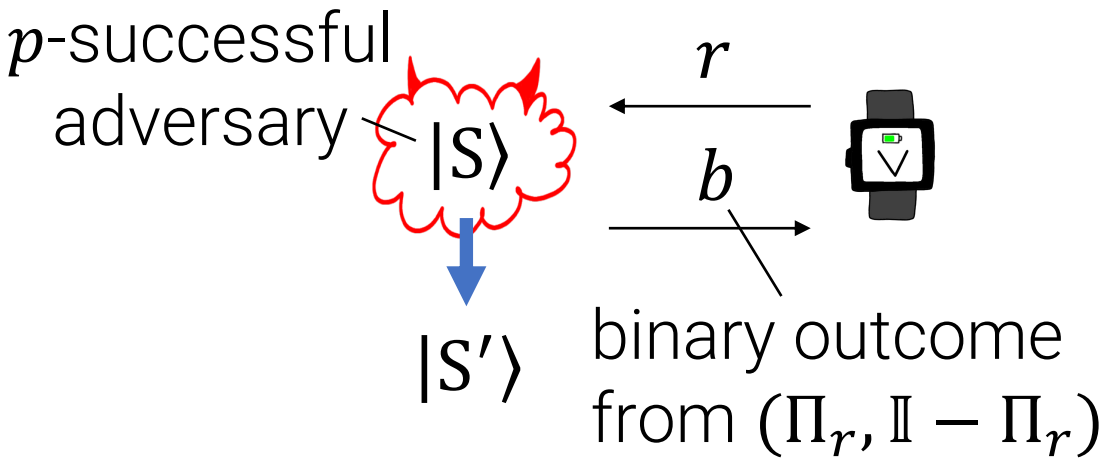
Repairing the Prover After Measurement



State repair task: Given $|S'\rangle$, generate a p-successful adversary state.

We'll use the [MW05] alternating projectors idea.

Repairing the Prover After Measurement

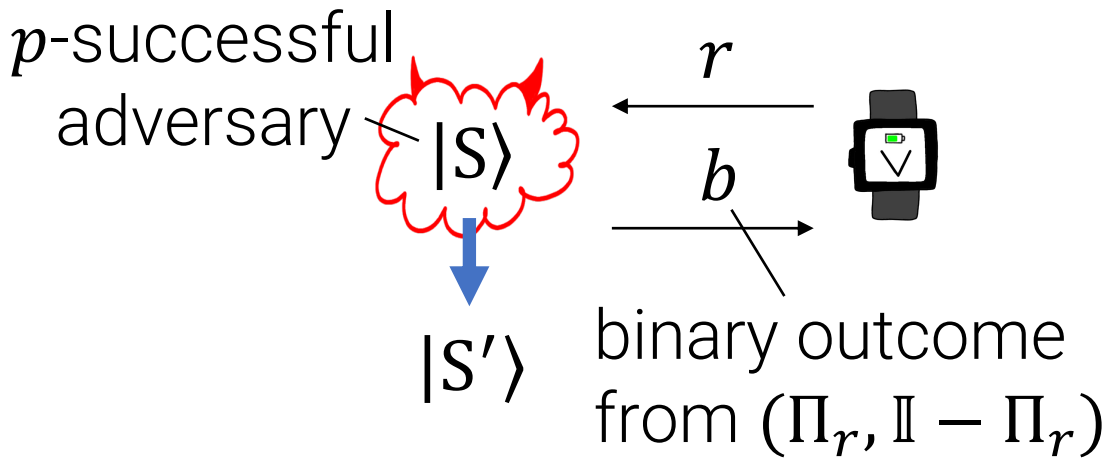


State repair task: Given $|S'\rangle$, generate a p-successful adversary state.

We'll use the [MW05] alternating projectors idea.

But which projectors do we use? Recall what we did last time.

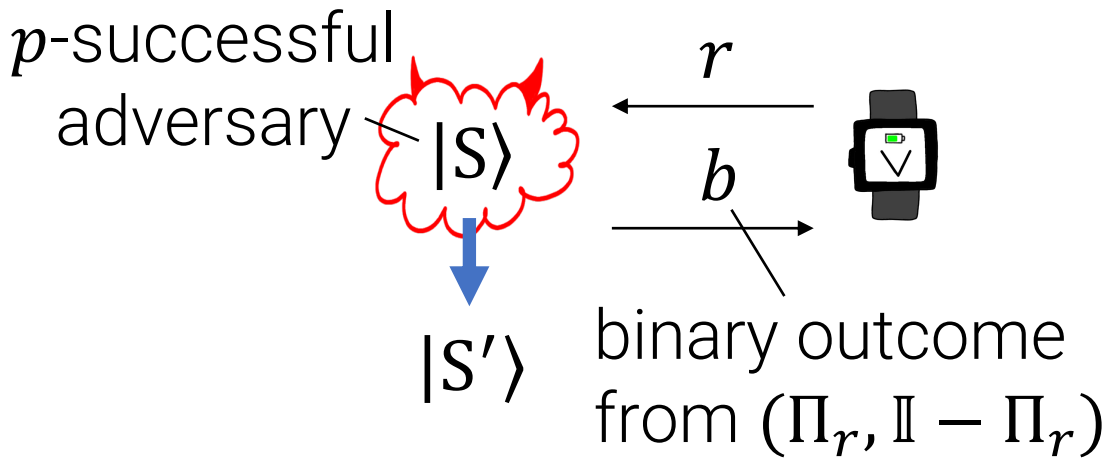
Repairing the Prover After Measurement



State repair task: Given $|S'\rangle$, generate a p-successful adversary state.

Watrous rewinding task: given verifier state $|\psi\rangle$ and projector Π_G indicating "successful simulation", output the state $\Pi_G|\psi\rangle_V|0\rangle_{Aux}$

Repairing the Prover After Measurement

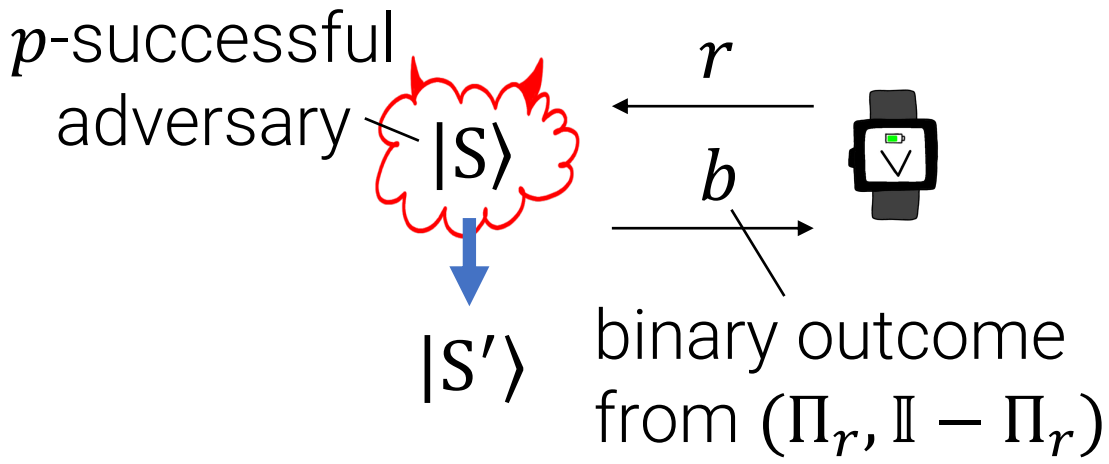


State repair task: Given $|S'\rangle$, generate a p-successful adversary state.

Watrous rewinding task: given verifier state $|\psi\rangle$ and projector Π_G indicating "successful simulation", output the state $\Pi_G|\psi\rangle_V|0\rangle_{Aux}$

Algorithm: alternate Π_0, Π_G measurements until Π_G accepts.

Repairing the Prover After Measurement



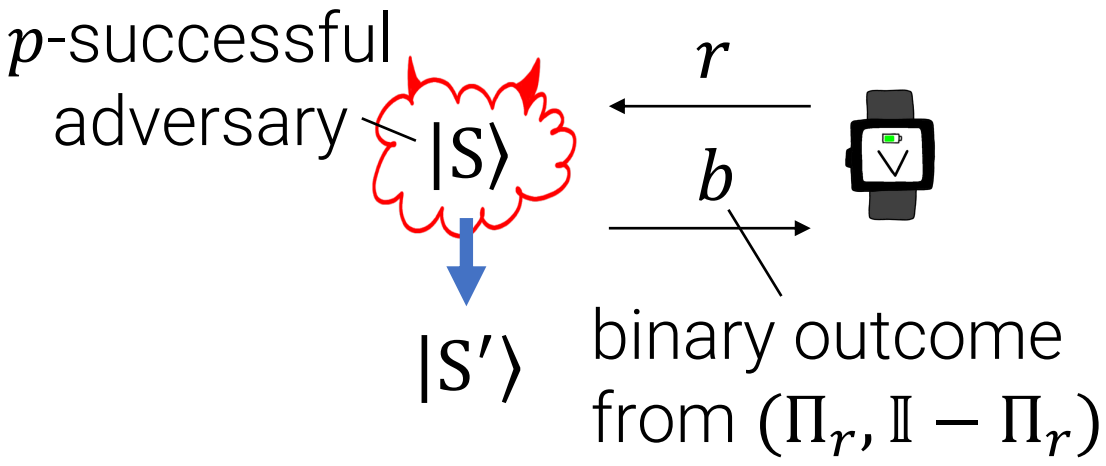
State repair task: Given $|S'\rangle$, generate a p-successful adversary state.

Watrous rewinding task: given verifier state $|\psi\rangle$ and projector Π_G indicating "successful simulation", output the state $\Pi_G|\psi\rangle_V|0\rangle_{Aux}$

Algorithm: alternate Π_0, Π_G measurements until Π_G accepts.

Why these projectors?

Repairing the Prover After Measurement



State repair task: Given $|S'\rangle$, generate a p-successful adversary state.

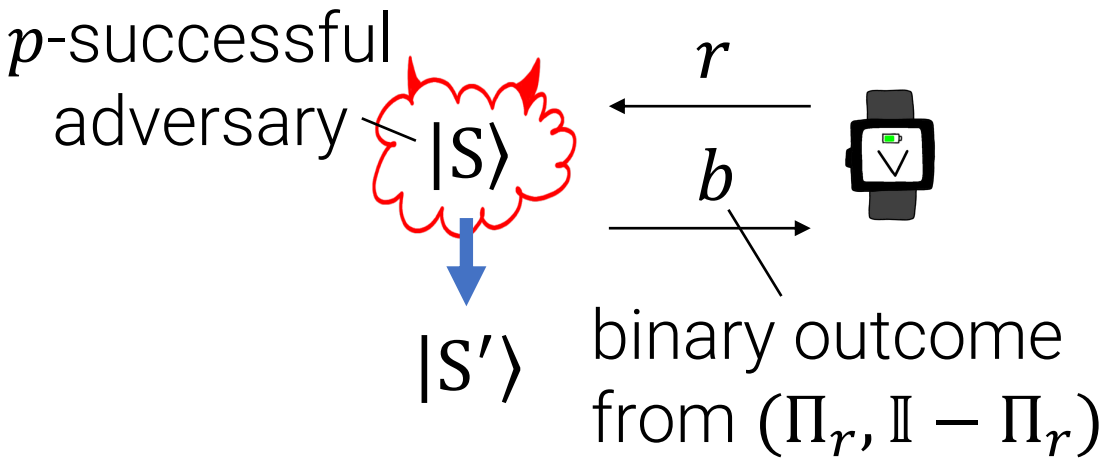
Watrous rewinding task: given verifier state $|\psi\rangle$ and projector Π_G indicating "successful simulation", output the state $\Pi_G|\psi\rangle_V|0\rangle_{Aux}$

Algorithm: alternate Π_0, Π_G measurements until Π_G accepts.

Why these projectors?

- $\mathbf{image}(\Pi_0)$ contains the *initial state* $|\psi\rangle|0\rangle$
- $\mathbf{image}(\Pi_G)$ contains the *target state* $\Pi_G|\psi\rangle_V|0\rangle_{Aux}$

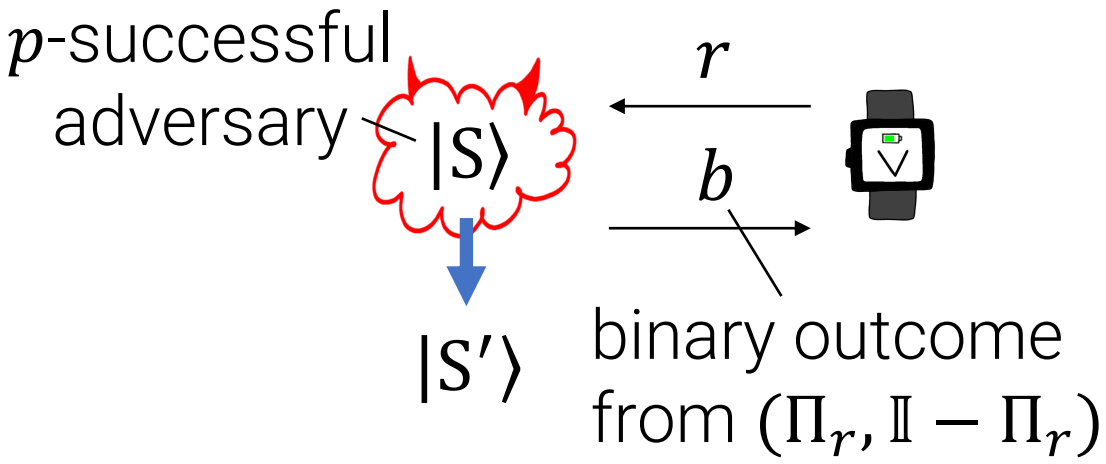
Repairing the Prover After Measurement



State repair task: Given $|S'\rangle$, generate a p-successful adversary state.

How do we apply the [MW05,W05] approach to our setting?

Repairing the Prover After Measurement

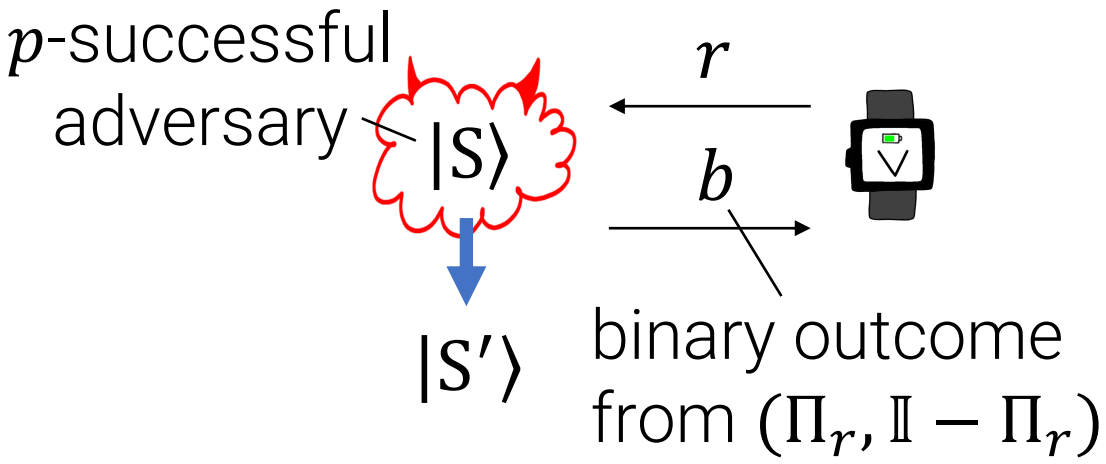


State repair task: Given $|S'\rangle$, generate a p-successful adversary state.

How do we apply the [MW05,W05] approach to our setting?

Oversimplification: suppose we can efficiently implement $(\Pi_p, \mathbb{I} - \Pi_p)$ where $\mathbf{image}(\Pi_p)$ corresponds to p-successful adversary states.

Repairing the Prover After Measurement



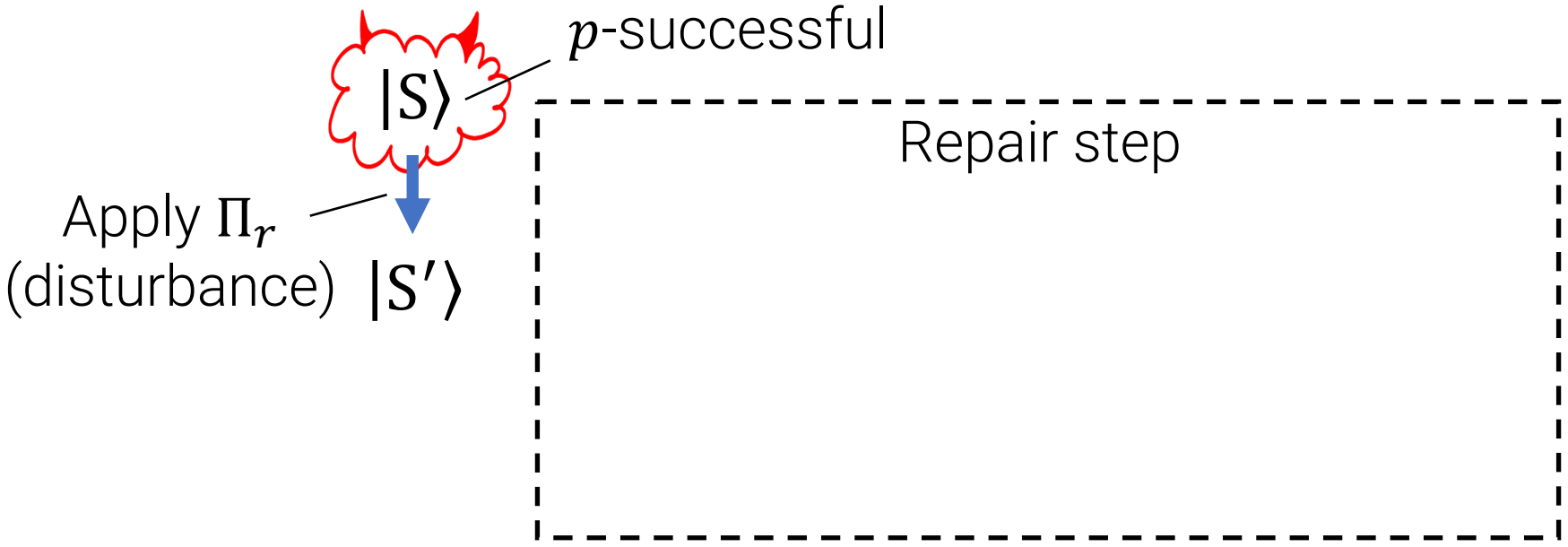
State repair task: Given $|S'\rangle$, generate a p-successful adversary state.

How do we apply the [MW05,W05] approach to our setting?

Oversimplification: suppose we can efficiently implement $(\Pi_p, \mathbb{I} - \Pi_p)$ where $\mathbf{image}(\Pi_p)$ corresponds to p-successful adversary states.

Proposal: just alternate Π_r, Π_p measurements until Π_p accepts!

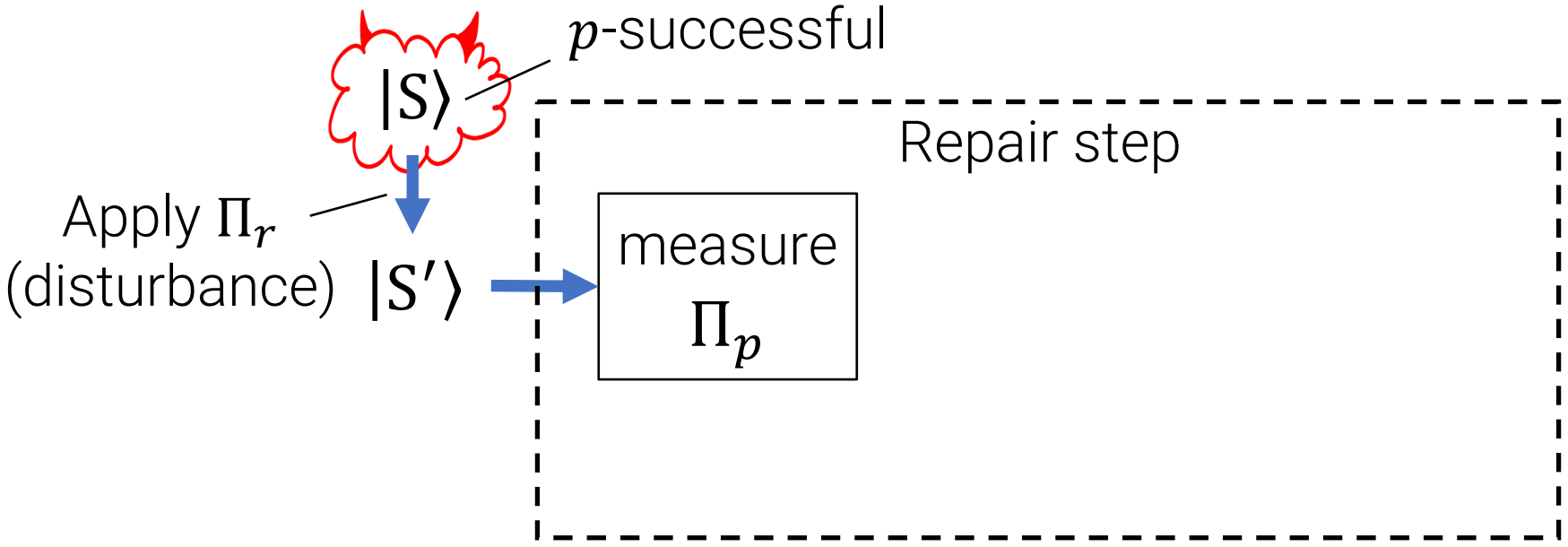
Repairing the Prover After Measurement



Oversimplification: suppose we can efficiently implement $(\Pi_p, \mathbb{I} - \Pi_p)$ where $\mathbf{image}(\Pi_p)$ corresponds to p-successful adversary states.

Proposal: just alternate Π_r, Π_p measurements until Π_p accepts!

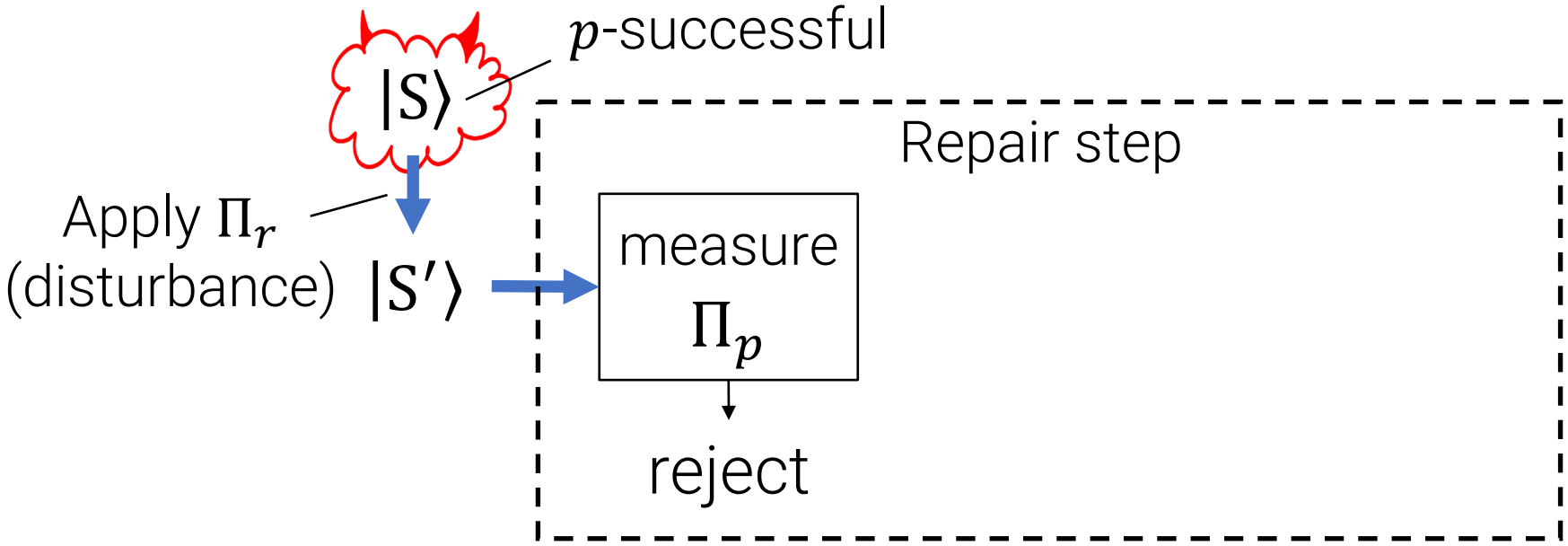
Repairing the Prover After Measurement



Oversimplification: suppose we can efficiently implement $(\Pi_p, \mathbb{I} - \Pi_p)$ where $\mathbf{image}(\Pi_p)$ corresponds to p-successful adversary states.

Proposal: just alternate Π_r, Π_p measurements until Π_p accepts!

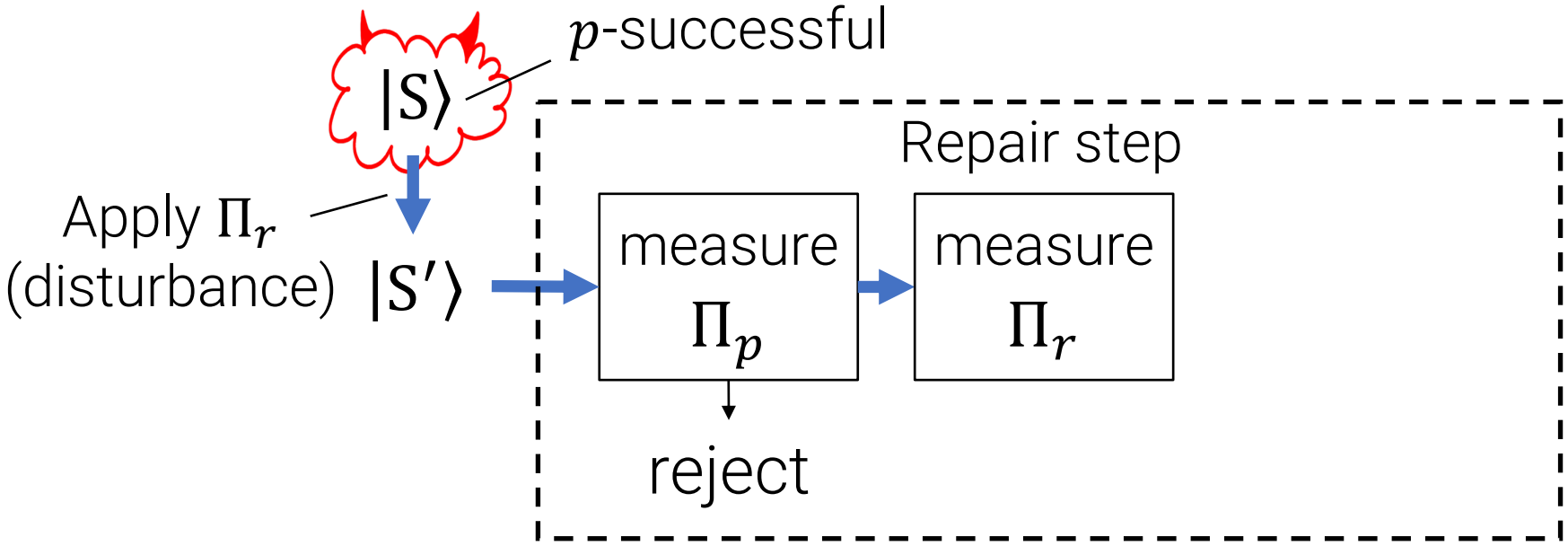
Repairing the Prover After Measurement



Oversimplification: suppose we can efficiently implement $(\Pi_p, \mathbb{I} - \Pi_p)$ where $\mathbf{image}(\Pi_p)$ corresponds to p-successful adversary states.

Proposal: just alternate Π_r, Π_p measurements until Π_p accepts!

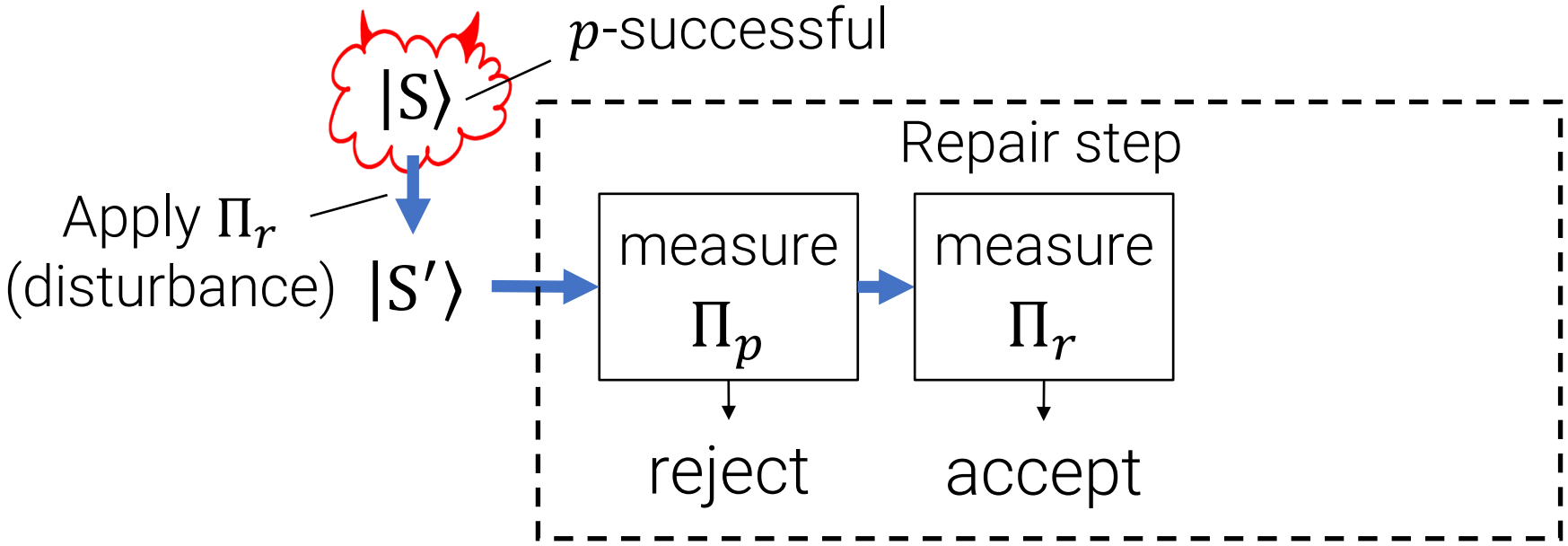
Repairing the Prover After Measurement



Oversimplification: suppose we can efficiently implement $(\Pi_p, \mathbb{I} - \Pi_p)$ where $\mathbf{image}(\Pi_p)$ corresponds to p-successful adversary states.

Proposal: just alternate Π_r, Π_p measurements until Π_p accepts!

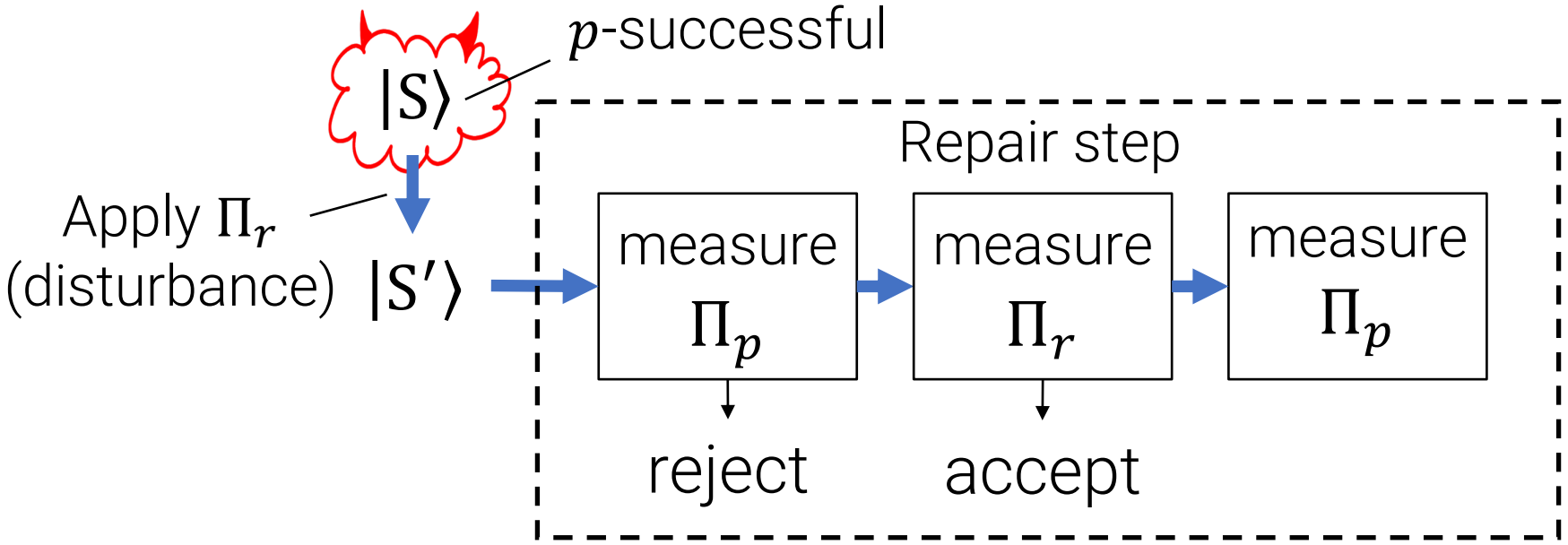
Repairing the Prover After Measurement



Oversimplification: suppose we can efficiently implement $(\Pi_p, \mathbb{I} - \Pi_p)$ where $\mathbf{image}(\Pi_p)$ corresponds to p-successful adversary states.

Proposal: just alternate Π_r, Π_p measurements until Π_p accepts!

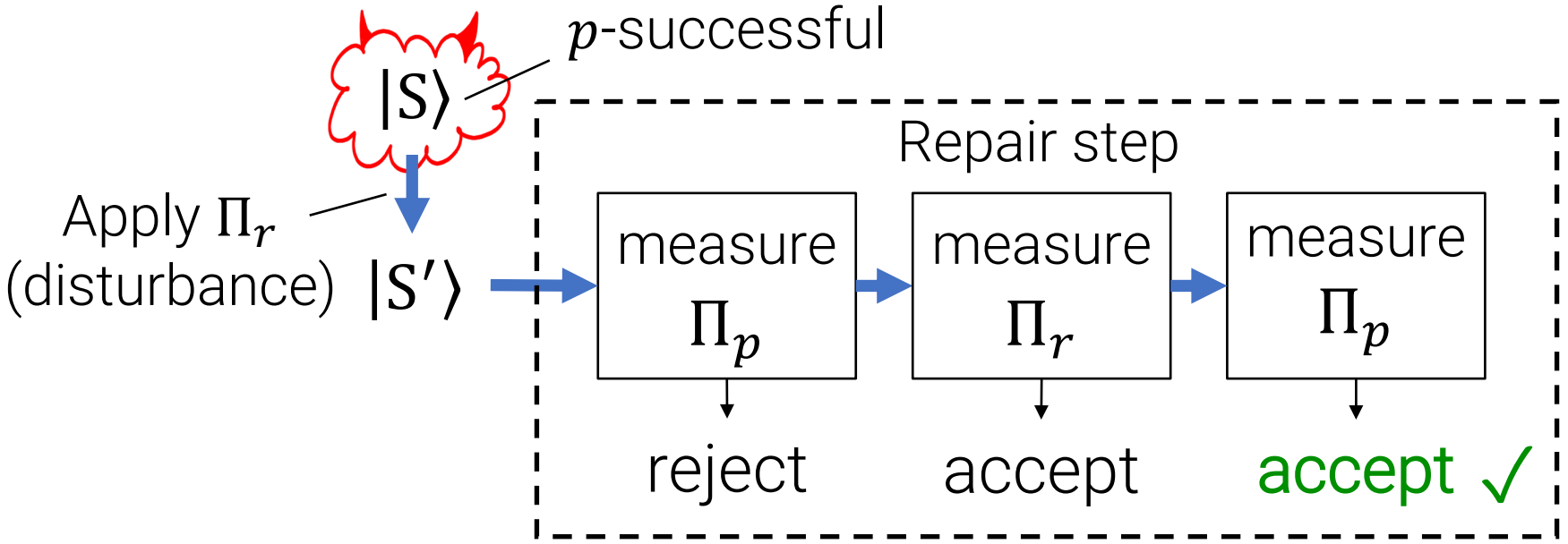
Repairing the Prover After Measurement



Oversimplification: suppose we can efficiently implement $(\Pi_p, \mathbb{I} - \Pi_p)$ where $\mathbf{image}(\Pi_p)$ corresponds to p-successful adversary states.

Proposal: just alternate Π_r, Π_p measurements until Π_p accepts!

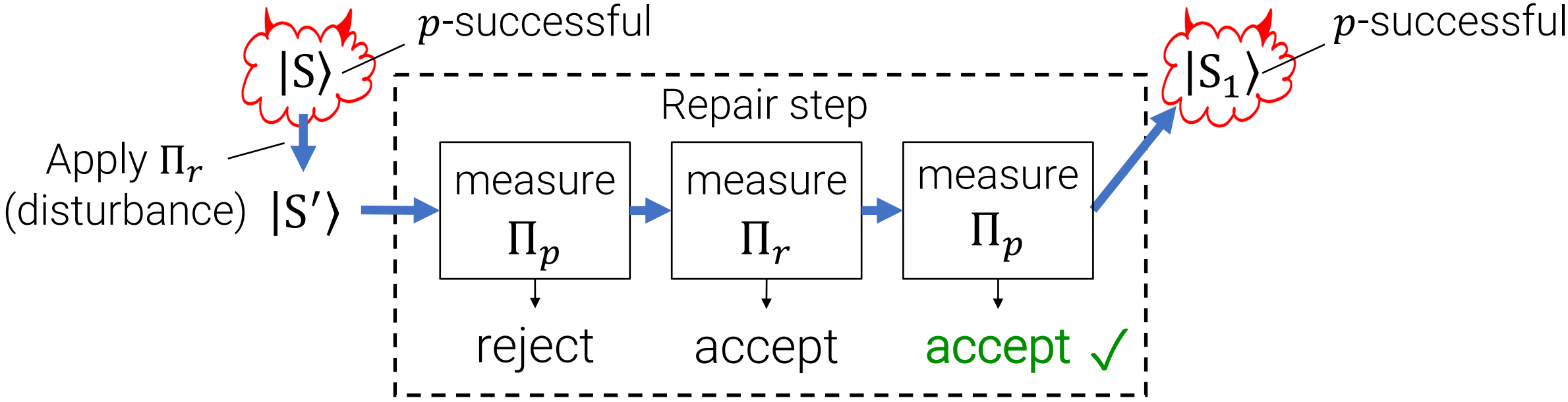
Repairing the Prover After Measurement



Oversimplification: suppose we can efficiently implement $(\Pi_p, \mathbb{I} - \Pi_p)$ where $\mathbf{image}(\Pi_p)$ corresponds to p-successful adversary states.

Proposal: just alternate Π_r, Π_p measurements until Π_p accepts!

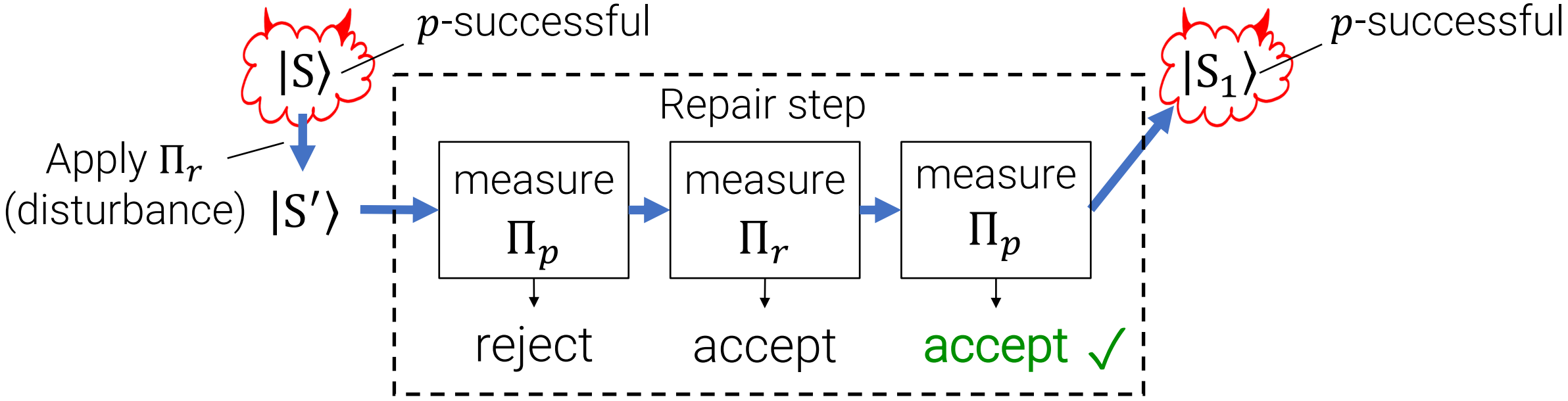
Repairing the Prover After Measurement



Oversimplification: suppose we can efficiently implement $(\Pi_p, \mathbb{I} - \Pi_p)$ where $\mathbf{image}(\Pi_p)$ corresponds to p-successful adversary states.

Proposal: just alternate Π_r, Π_p measurements until Π_p accepts!

Repairing the Prover After Measurement

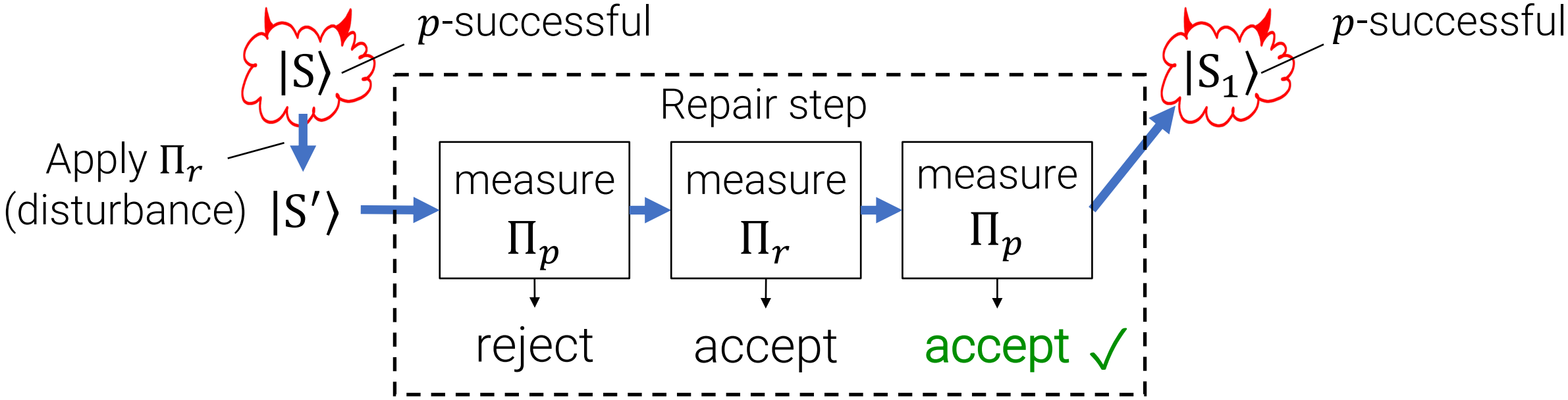


Missing Pieces

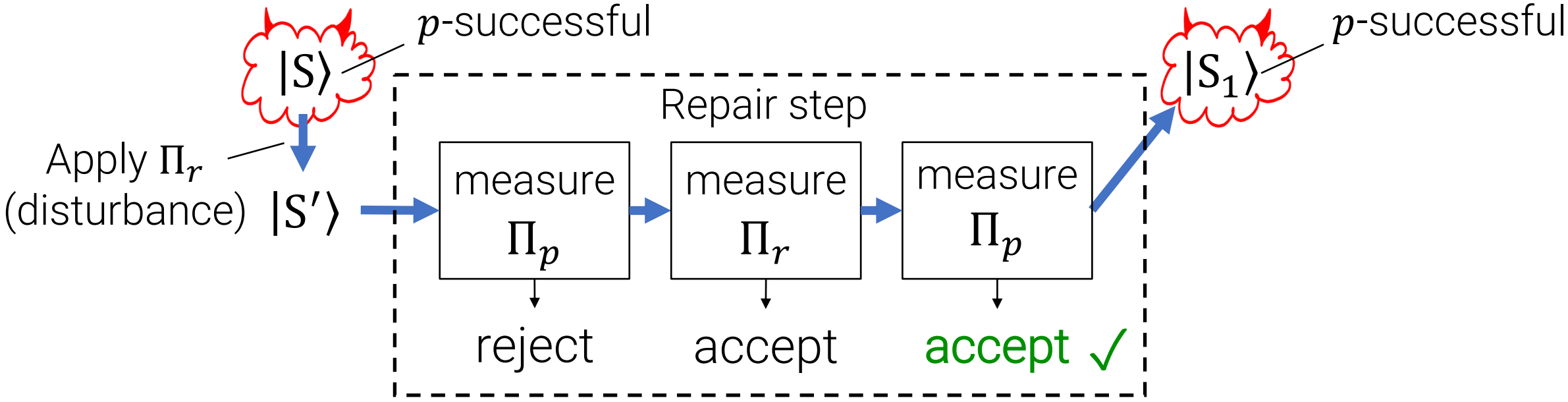
1) Why does this terminate?

2) How do we define/implement Π_p?

Why does this terminate?

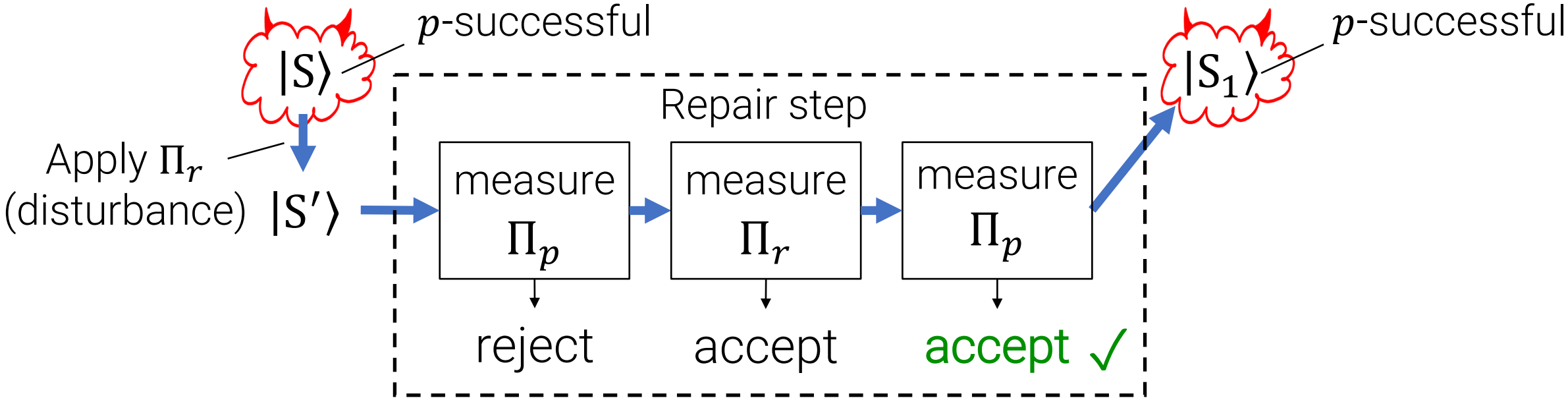


Why does this terminate?



In the last talk, we used special properties of the two projectors to bound the runtime, but it's not clear what we can say about Π_r, Π_p.

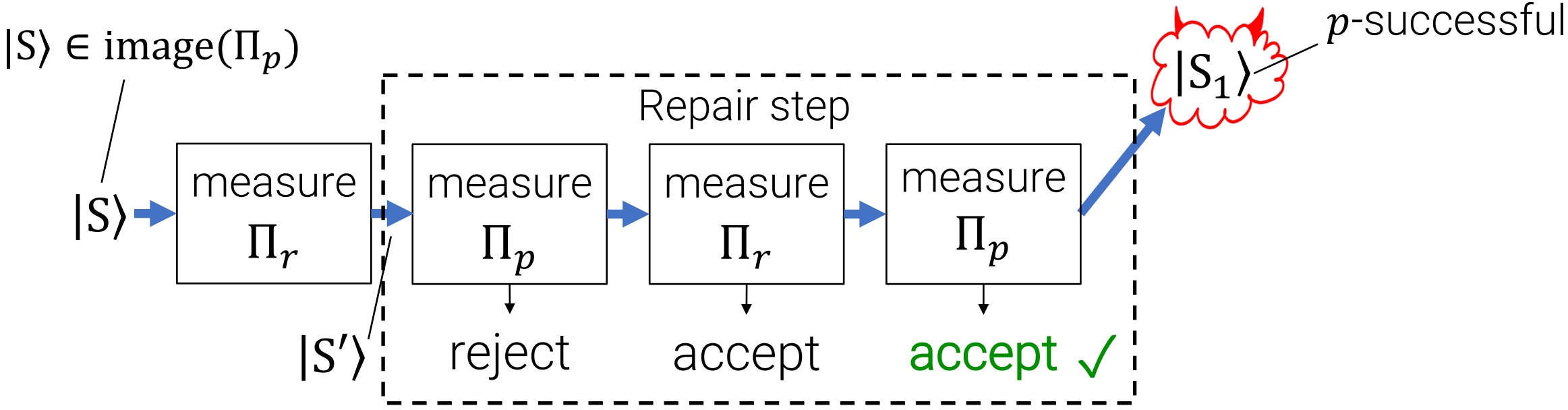
Why does this terminate?



In the last talk, we used special properties of the two projectors to bound the runtime, but it's not clear what we can say about Π_r, Π_p.

Insight: analyze runtime starting from $|S\rangle$, not $|S'\rangle$.

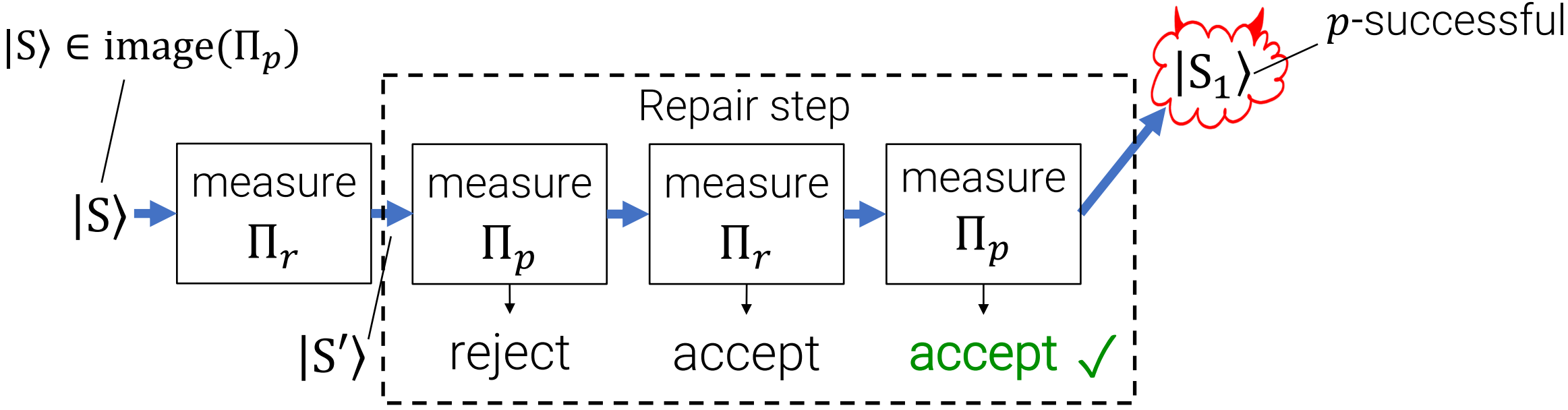
Why does this terminate?



In the last talk, we used special properties of the two projectors to bound the runtime, but it's not clear what we can say about Π_r, Π_p.

Insight: analyze runtime starting from $|S\rangle$, not $|S'\rangle$.

Why does this terminate?

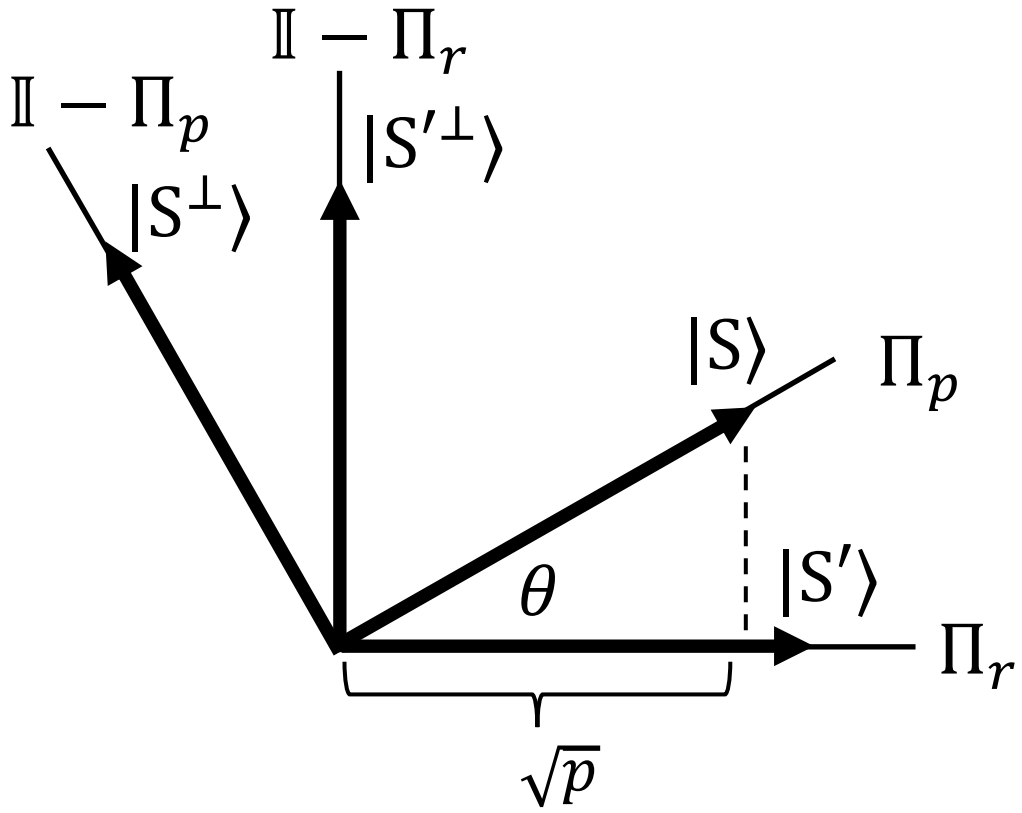


In the last talk, we used special properties of the two projectors to bound the runtime, but it's not clear what we can say about Π_r, Π_p.

Insight: analyze runtime starting from $|S\rangle$, not $|S'\rangle$. Why does this help?

"Return to Subspace" Lemma: If we start at $|S\rangle \in \mathbf{image}(\Pi_p)$ and alternate Π_r, Π_p measurements, return to $\mathbf{image}(\Pi_p)$ in $O(1)$ expected steps.

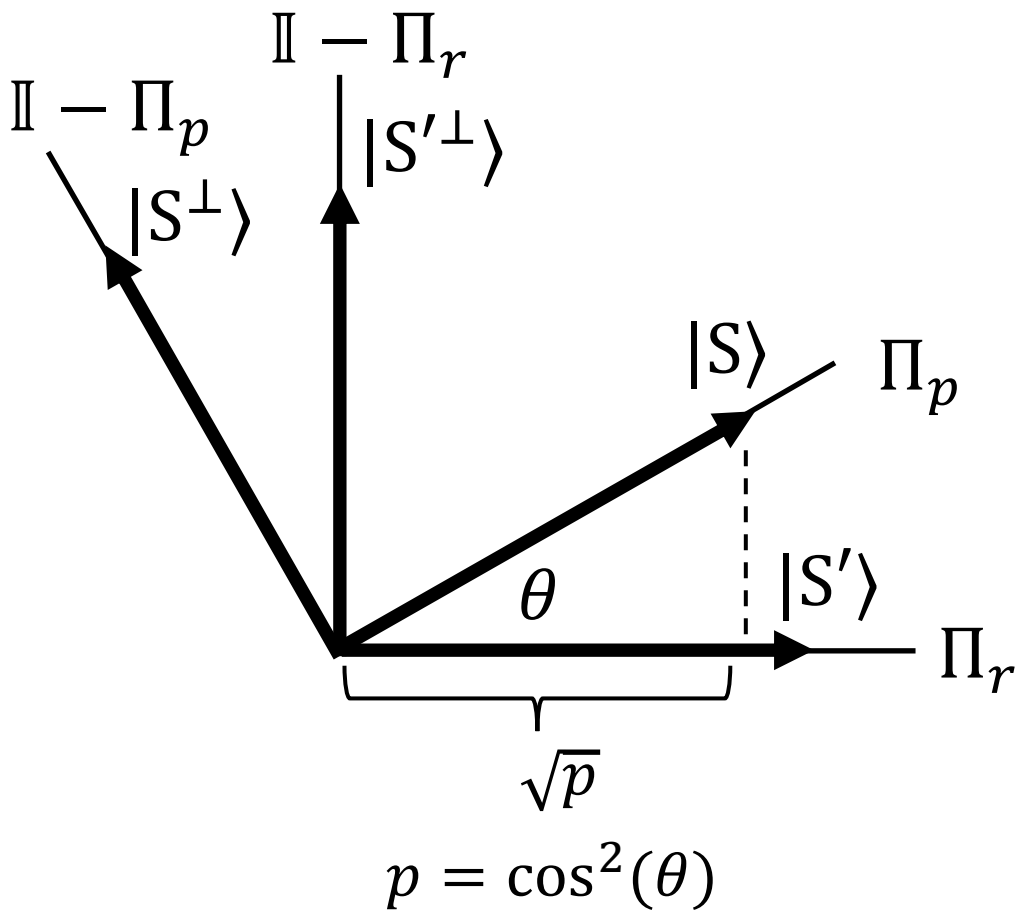
"Return to Subspace" Lemma: If we start at $|S\rangle \in \mathbf{image}(\Pi_p)$ and alternate Π_r, Π_p measurements, return to $\mathbf{image}(\Pi_p)$ in $O(1)$ expected steps.



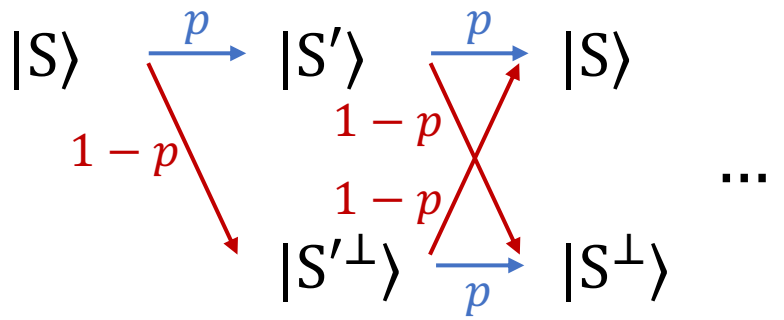
$p = \cos^2(\theta)$

Consider the 2-D case.

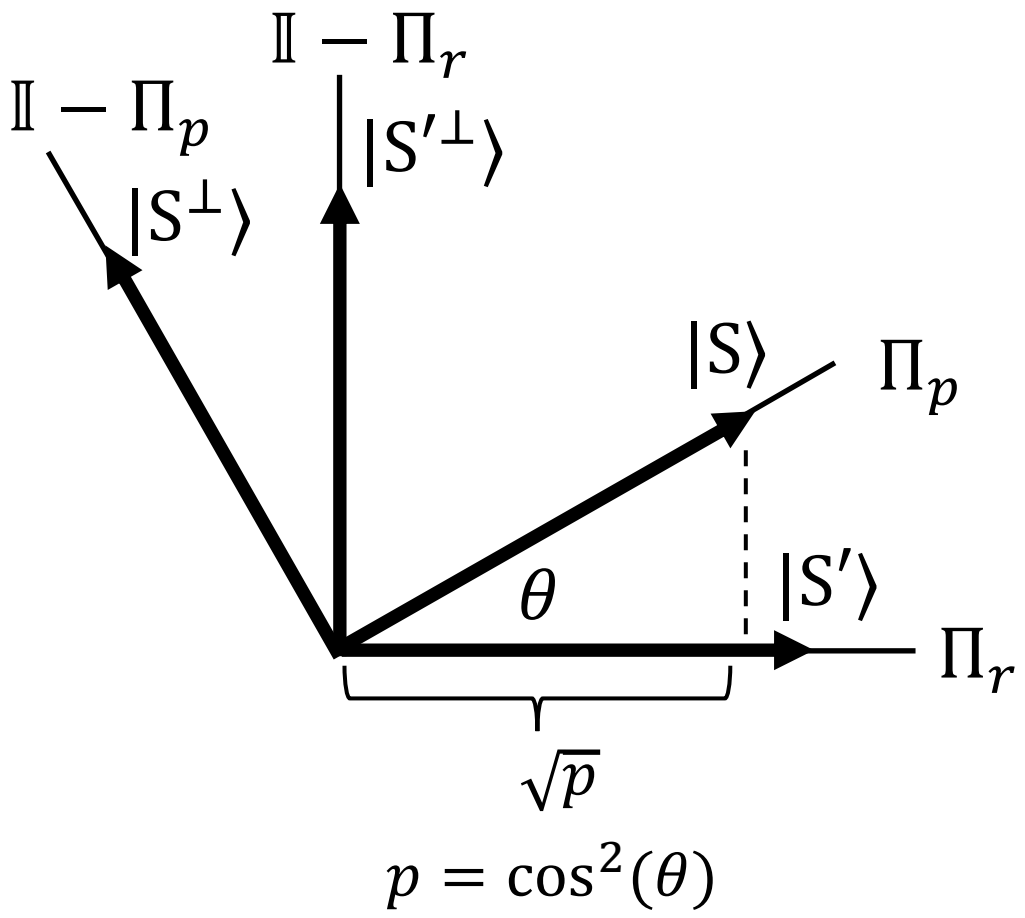
"Return to Subspace" Lemma: If we start at $|S\rangle \in \mathbf{image}(\Pi_p)$ and alternate Π_r, Π_p measurements, return to $\mathbf{image}(\Pi_p)$ in $O(1)$ expected steps.



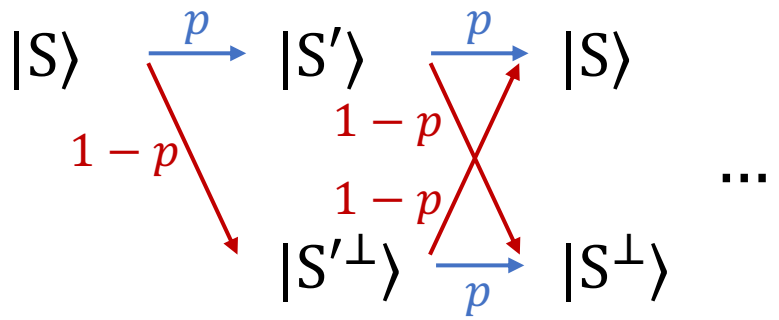
Consider the 2-D case.



"Return to Subspace" Lemma: If we start at $|S\rangle \in \mathbf{image}(\Pi_p)$ and alternate Π_r, Π_p measurements, return to $\mathbf{image}(\Pi_p)$ in $O(1)$ expected steps.

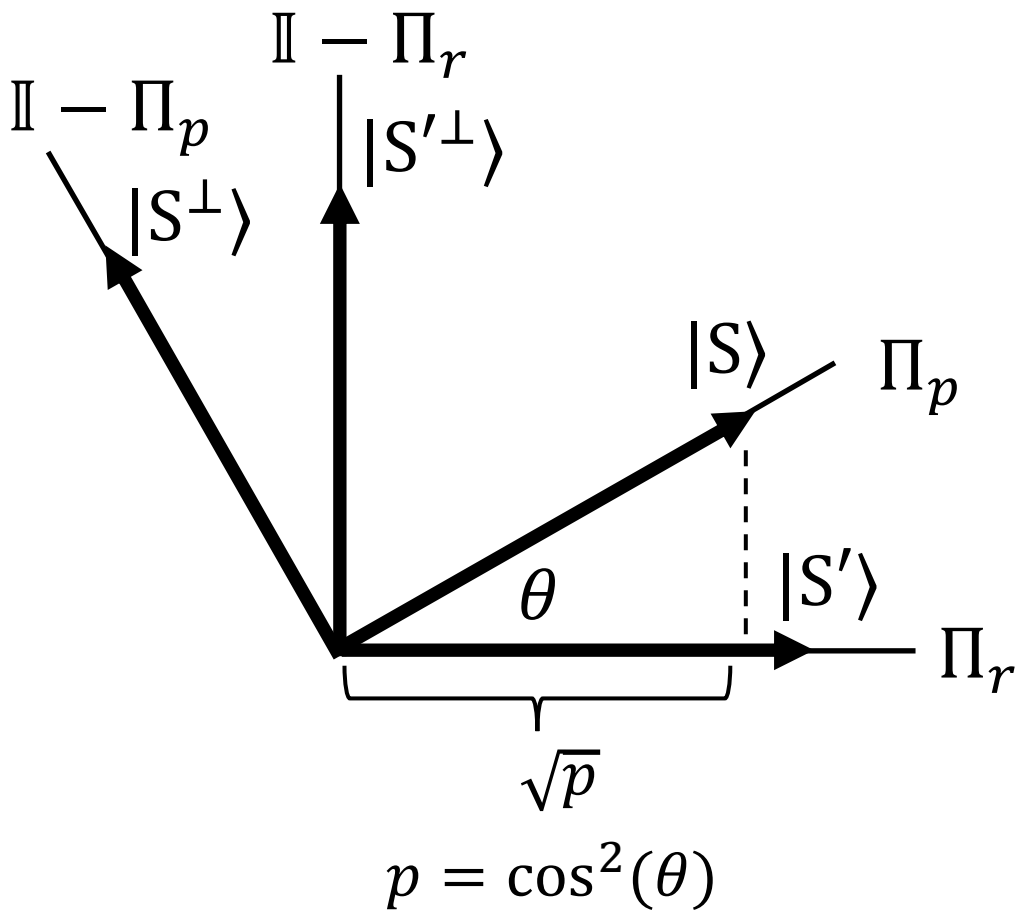


Consider the 2-D case.

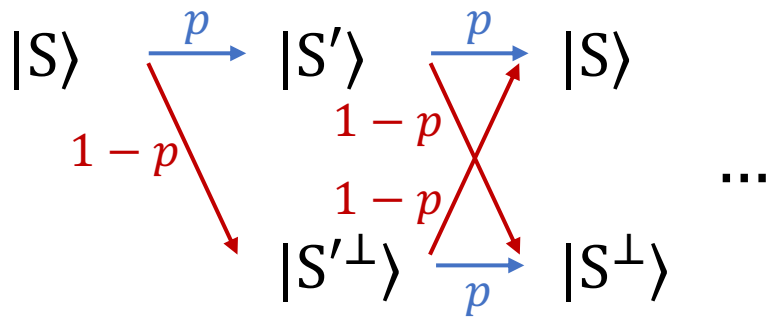


Simple calculation: time to return to $|S\rangle$ is *independent* of θ.

"Return to Subspace" Lemma: If we start at $|S\rangle \in \mathbf{image}(\Pi_p)$ and alternate Π_r, Π_p measurements, return to $\mathbf{image}(\Pi_p)$ in $O(1)$ expected steps.



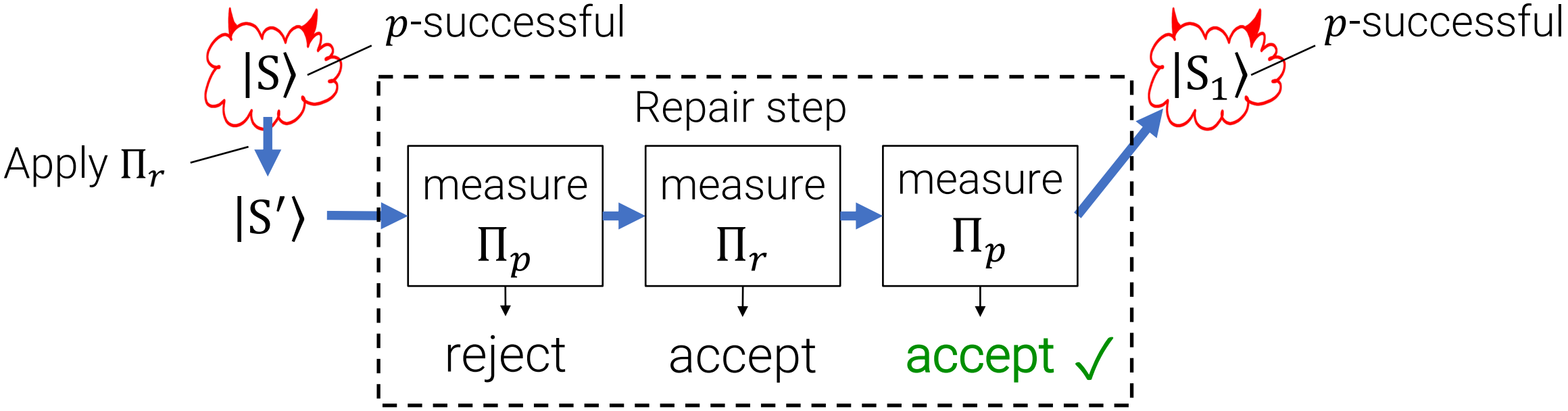
Consider the 2-D case.



Simple calculation: time to return to $|S\rangle$ is *independent* of θ.

This extends to the general case by Jordan's lemma.

Repairing the Prover After Measurement

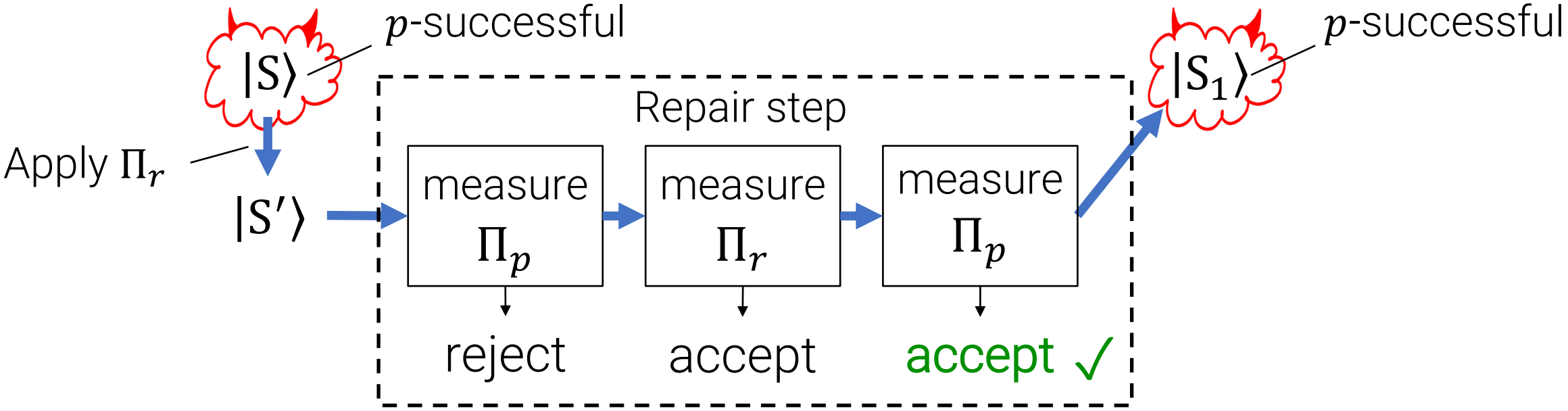


Missing Pieces

~~1) Why does this terminate?~~

2) How do we define/implement Π_p?

Repairing the Prover After Measurement



Missing Pieces

~~1) Why does this terminate?~~

2) How do we define/implement Π_p?

How do we define/implement Π_p?

Rephrased: how do we measure the prover's success probability?

How do we define/implement Π_p?

Rephrased: how do we measure the prover's success probability?

Bad news: we can't do this efficiently.

How do we define/implement Π_p?

Rephrased: how do we measure the prover's success probability?

Bad news: we can't do this efficiently.

Good news: we can ***approximately measure*** the success probability…

How do we define/implement Π_p?

Rephrased: how do we measure the prover's success probability?

Bad news: we can't do this efficiently.

Good news: we can ***approximately measure*** the success probability…

How?

How do we define/implement Π_p?

Rephrased: how do we measure the prover's success probability?

Bad news: we can't do this efficiently.

Good news: we can ***approximately measure*** the success probability…

How?

Alternating projectors ***again***!

What this talk will cover:

1. Motivating example: Kilian's succinct arguments for NP
2. Why is post-quantum security of Kilian difficult?
3. Rewinding a quantum attacker many times
   - New idea: "repair" the adversary after each query
   - Estimating success probability
   - The full rewinding procedure
   - Analysis

How to Estimate Success Probability [MW05,Z20]

Instead of running the prover on a random challenge $r \leftarrow R$, we'll introduce a challenge register R and run the prover $|S\rangle$ in superposition on all challenges.

How to Estimate Success Probability [MW05,Z20]

Instead of running the prover on a random challenge $r \leftarrow R$, we'll introduce a challenge register R and run the prover $|S\rangle$ in superposition on all challenges.

- $|+\rangle_R := \frac{1}{\sqrt{R}} \sum_{r \in R} |r\rangle$ (uniform superposition of challenges)

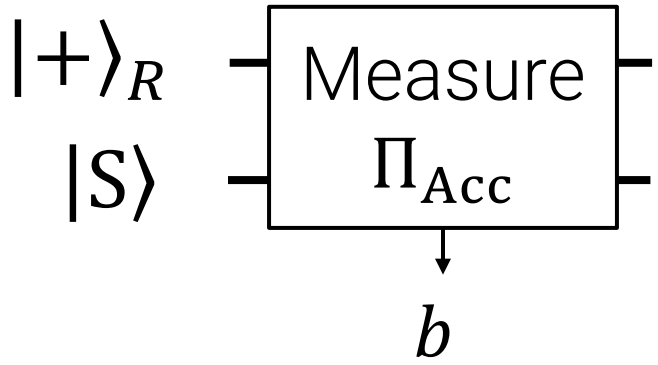
How to Estimate Success Probability [MW05,Z20]

Instead of running the prover on a random challenge $r \leftarrow R$, we'll introduce a challenge register R and run the prover $|\mathrm{S}\rangle$ in superposition on all challenges.

- $|+\rangle_R := \frac{1}{\sqrt{R}} \sum_{r \in R} |r\rangle$ (uniform superposition of challenges)
- $\Pi_{\mathrm{Acc}} := \sum_r |r\rangle\langle r|_R \otimes \Pi_r$.

How to Estimate Success Probability [MW05,Z20]

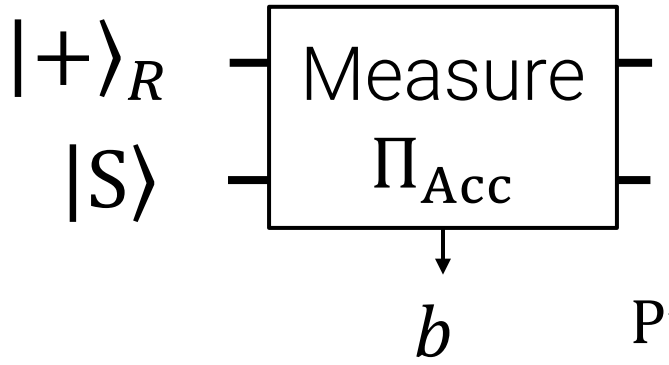
Instead of running the prover on a random challenge $r \leftarrow R$, we'll introduce a challenge register R and run the prover $|S\rangle$ in superposition on all challenges.



- $|+\rangle_R := \frac{1}{\sqrt{R}} \sum_{r \in R} |r\rangle$ (uniform superposition of challenges)
- $\Pi_{\mathrm{Acc}} := \sum_r |r\rangle\langle r|_R \otimes \Pi_r$.

How to Estimate Success Probability [MW05,Z20]

Instead of running the prover on a random challenge $r \leftarrow R$, we'll introduce a challenge register R and run the prover $|S\rangle$ in superposition on all challenges.

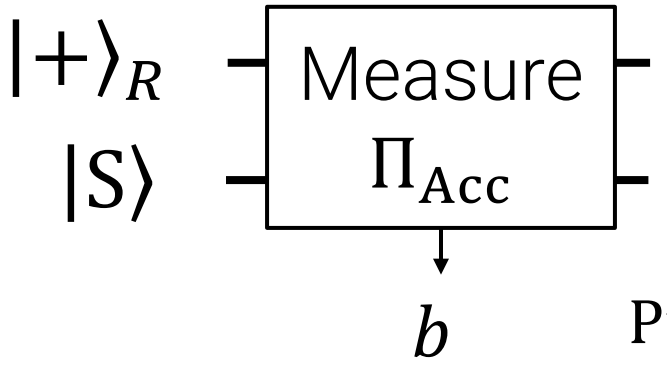


$\Pr[b = 1]$ is the success probability of $|S\rangle$

- $|+\rangle_R := \frac{1}{\sqrt{R}} \sum_{r \in R} |r\rangle$ (uniform superposition of challenges)
- $\Pi_{\mathrm{Acc}} := \sum_r |r\rangle\langle r|_R \otimes \Pi_r$.

How to Estimate Success Probability [MW05,Z20]

[MW05, Z20]: learn success probability
by alternating $\Pi_{\mathbf{Acc}}$ measurements with
$\Pi_{\mathbf{Unif}} = |+\rangle\langle+|_R \otimes \mathbb{I}$ measurements



$\Pr[b = 1]$ is the success probability of $|S\rangle$

- $|+\rangle_R \coloneqq \frac{1}{\sqrt{R}} \sum_{r \in R} |r\rangle$ (uniform superposition of challenges)
- $\Pi_{\mathbf{Acc}} \coloneqq \sum_r |r\rangle\langle r|_R \otimes \Pi_r$.
- $\Pi_{\mathbf{Unif}} \coloneqq |+\rangle\langle+|_R \otimes \mathbb{I}$.

How to Estimate Success Probability [MW05,Z20]

1) Initialize $|+_R\rangle|S\rangle$.

$|+\rangle_R$

$|S\rangle$

- $|+\rangle_R := \frac{1}{\sqrt{R}}\sum_{r\in R}|r\rangle$ (uniform superposition of challenges)
- $\Pi_{\mathrm{Acc}} := \sum_r |r\rangle\langle r|_R \otimes \Pi_r$.
- $\Pi_{\mathrm{Unif}} := |+\rangle\langle+|_R \otimes \mathbb{I}$.

How to Estimate Success Probability [MW05,Z20]

1) Initialize $|+_R\rangle|\mathrm{S}\rangle$.
2) Alternate $M_{\mathrm{Acc}}, M_{\mathrm{Unif}}$ measurements, obtaining $(b_1, b_2, \ldots, b_T)$

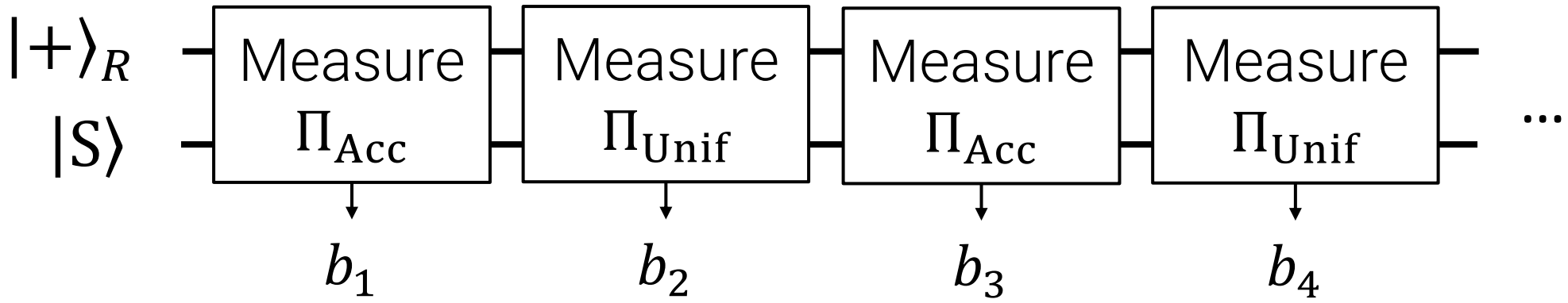
$|+\rangle_R$

$|\mathrm{S}\rangle$

- $|+\rangle_R := \frac{1}{\sqrt{R}}\sum_{r\in R}|r\rangle$ (uniform superposition of challenges)
- $\Pi_{\mathrm{Acc}} := \sum_r |r\rangle\langle r|_R \otimes \Pi_r$.
- $\Pi_{\mathrm{Unif}} := |+\rangle\langle+|_R \otimes \mathbb{I}$.

How to Estimate Success Probability [MW05,Z20]

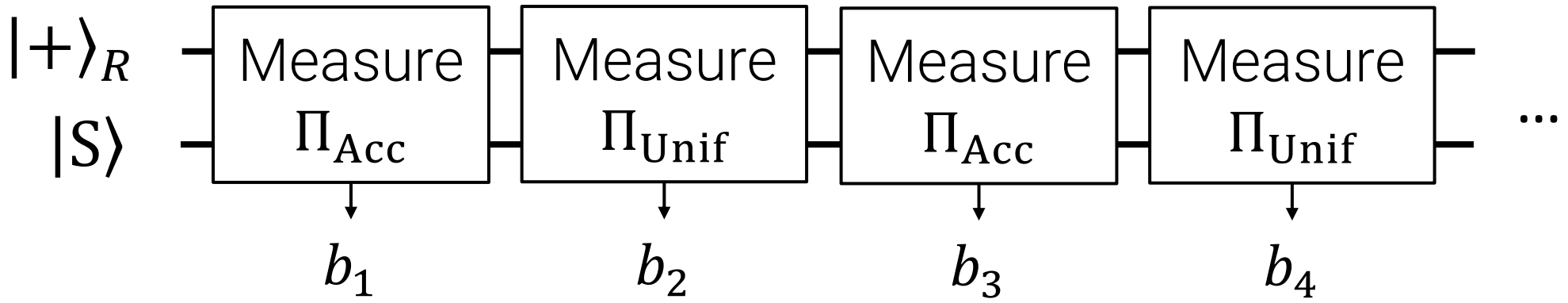
1) Initialize $|+_R\rangle|S\rangle$.
2) Alternate $M_{\mathrm{Acc}}, M_{\mathrm{Unif}}$ measurements, obtaining $(b_1, b_2, \ldots, b_T)$



- $|+\rangle_R := \frac{1}{\sqrt{R}} \sum_{r \in R} |r\rangle$ (uniform superposition of challenges)
- $\Pi_{\mathrm{Acc}} := \sum_r |r\rangle\langle r|_R \otimes \Pi_r$.
- $\Pi_{\mathrm{Unif}} := |+\rangle\langle +|_R \otimes \mathbb{I}$.

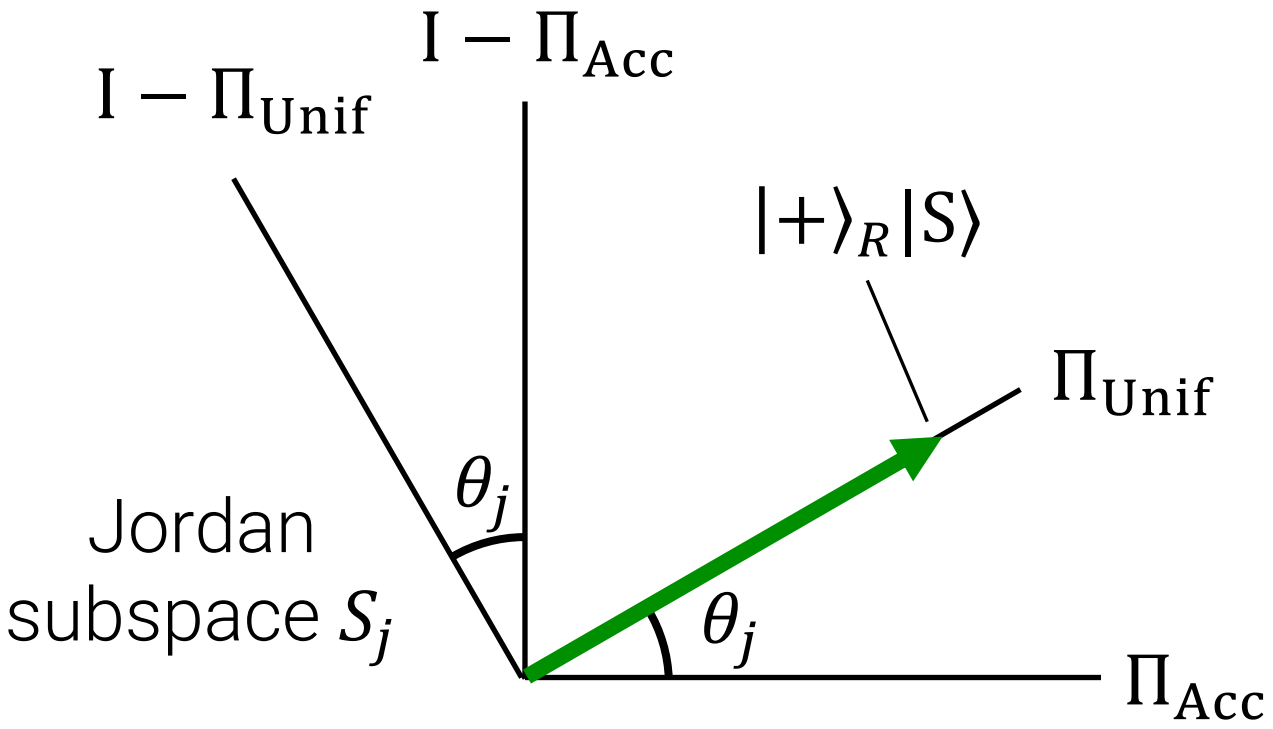
How to Estimate Success Probability [MW05,Z20]

1) Initialize $|+_R\rangle|S\rangle$.
2) Alternate $M_{\mathrm{Acc}}, M_{\mathrm{Unif}}$ measurements, obtaining $(b_1, b_2, \ldots, b_T)$
3) Output $p = [\#\ \text{of times}\ b_i = b_{i+1}]/(T-1)$

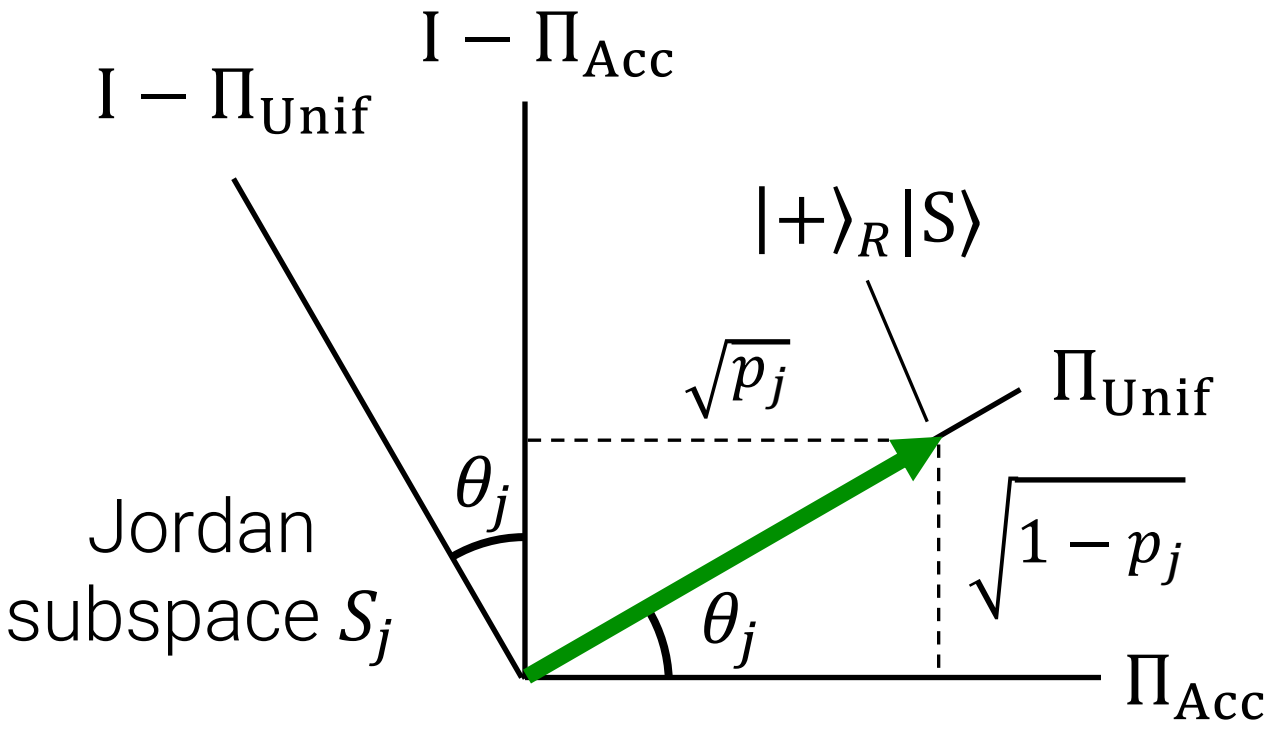


- $|+\rangle_R \coloneqq \frac{1}{\sqrt{R}} \sum_{r \in R} |r\rangle$ (uniform superposition of challenges)
- $\Pi_{\mathrm{Acc}} \coloneqq \sum_r |r\rangle\langle r|_R \otimes \Pi_r$.
- $\Pi_{\mathrm{Unif}} \coloneqq |+\rangle\langle +|_R \otimes \mathbb{I}$.

Why does this work?



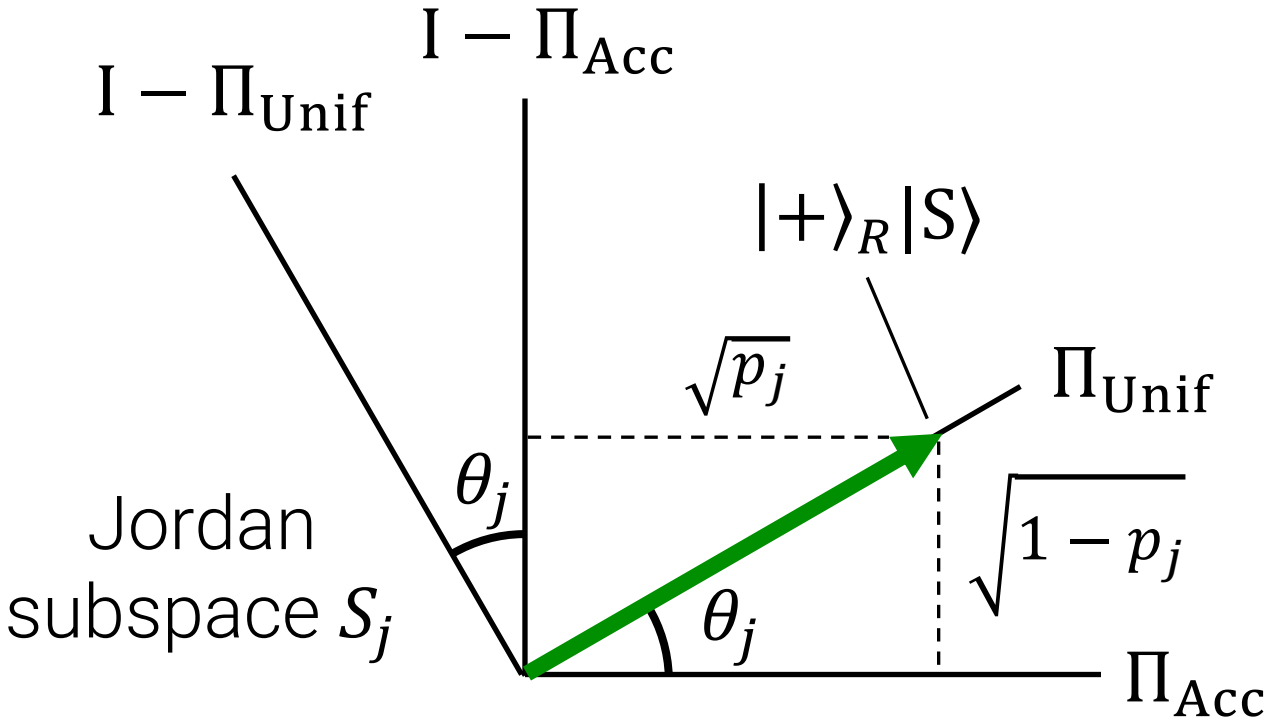
Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.



Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

Eigenvalue $p_j = \cos^2(\theta_j) = \|\Pi_{\mathrm{Acc}}|+\rangle_R|\mathrm{S}\rangle\|^2$

(p_j is an eigenvalue of $\Pi_{\mathrm{Unif}}\Pi_{\mathrm{Acc}}\Pi_{\mathrm{Unif}}$)

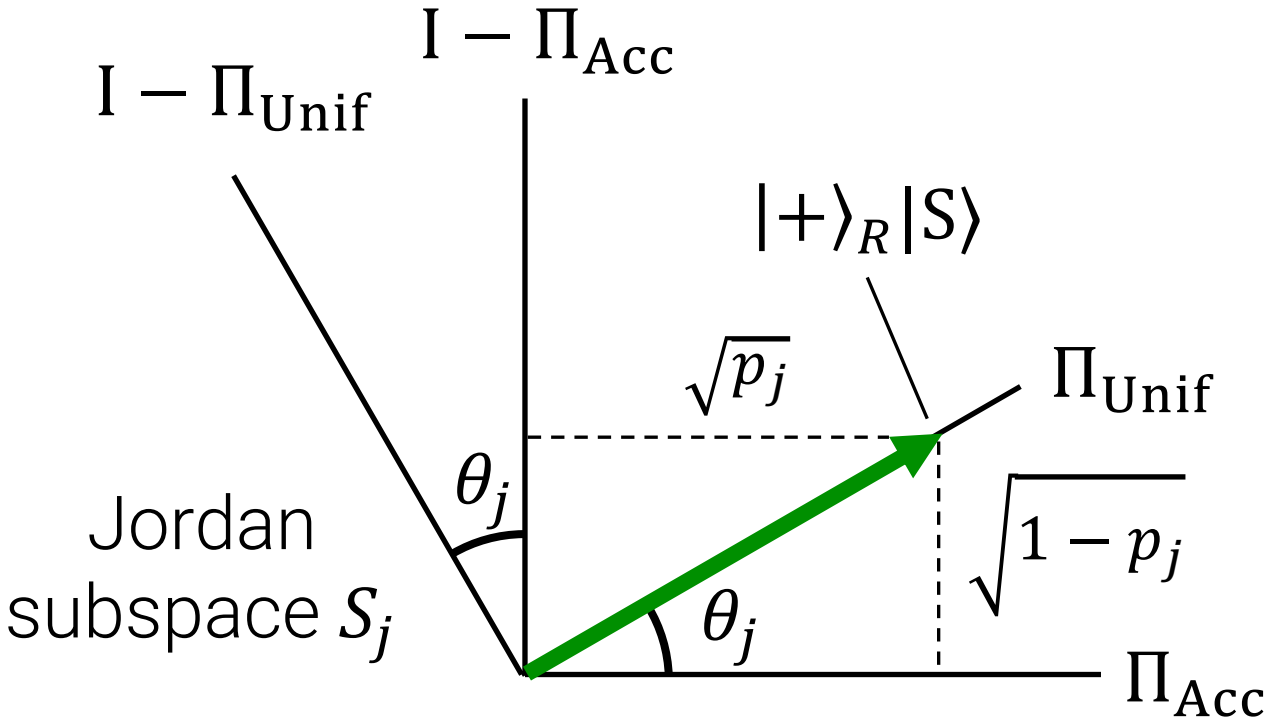


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.

Eigenvalue $p_j = \cos^2(\theta_j) = \|\Pi_{\mathrm{Acc}}|+\rangle_R|\mathrm{S}\rangle\|^2$

(p_j is an eigenvalue of $\Pi_{\mathrm{Unif}}\Pi_{\mathrm{Acc}}\Pi_{\mathrm{Unif}}$)

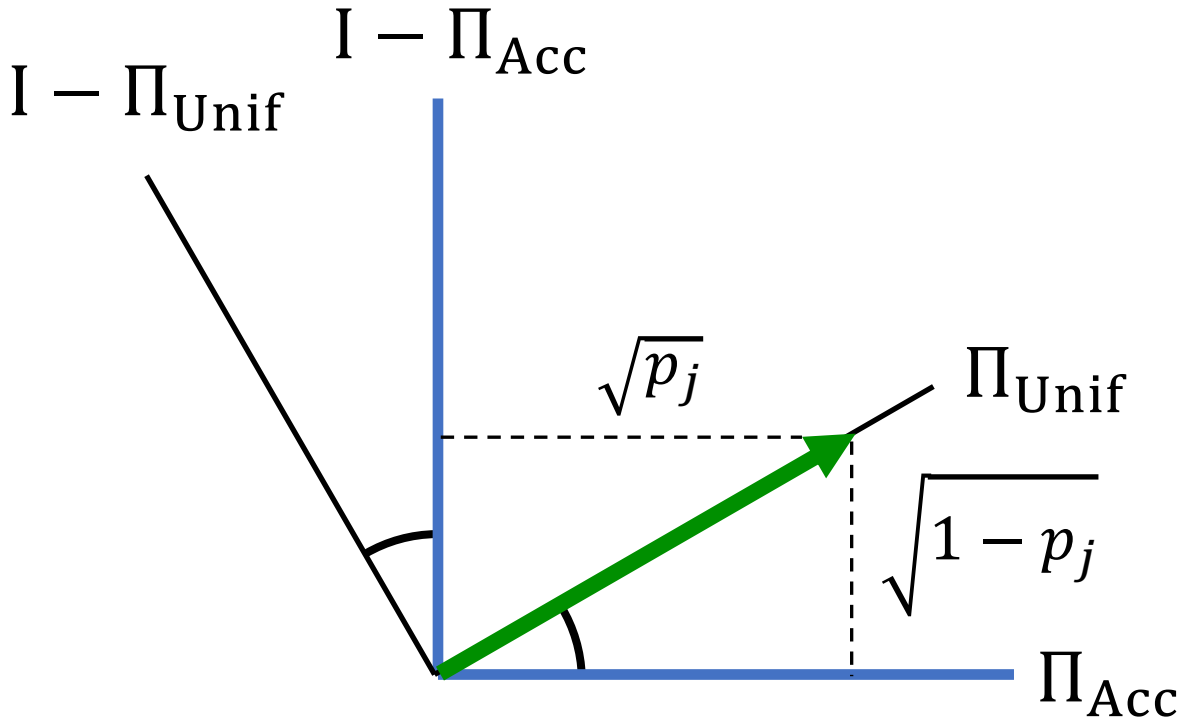


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathbf{Acc}}, \Pi_{\mathbf{Unif}}$ measurements gives p_j

Eigenvalue $p_j = \cos^2(\theta_j) = \|\Pi_{\mathbf{Acc}}|+\rangle_R|\mathrm{S}\rangle\|^2$

(p_j is an eigenvalue of $\Pi_{\mathbf{Unif}}\Pi_{\mathbf{Acc}}\Pi_{\mathbf{Unif}}$)



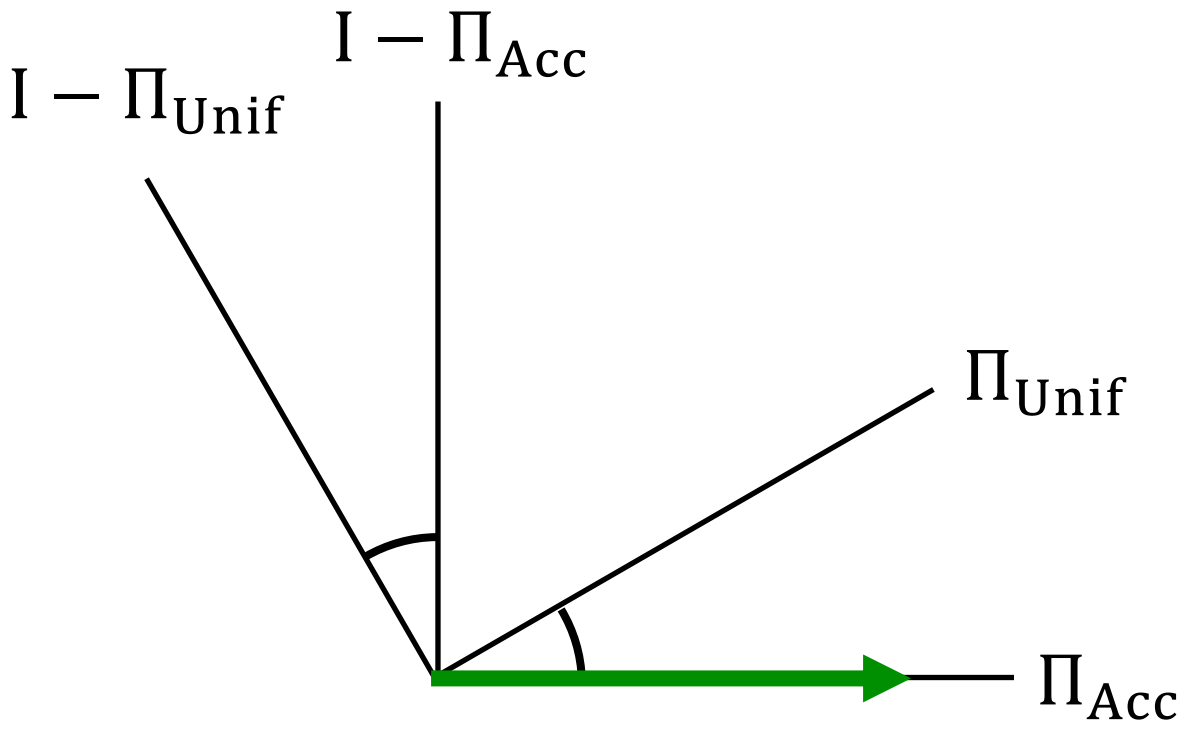
Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif

$b_0 = 1$

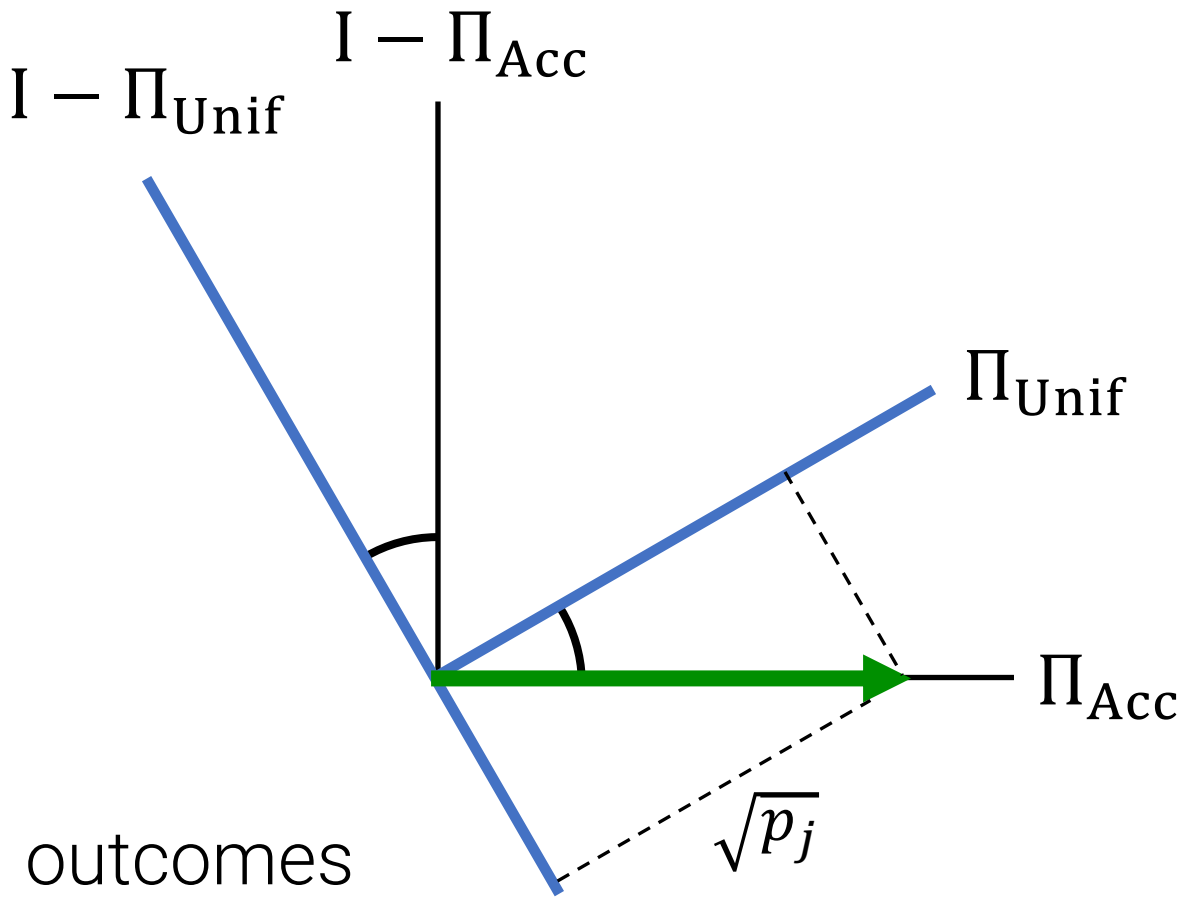


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

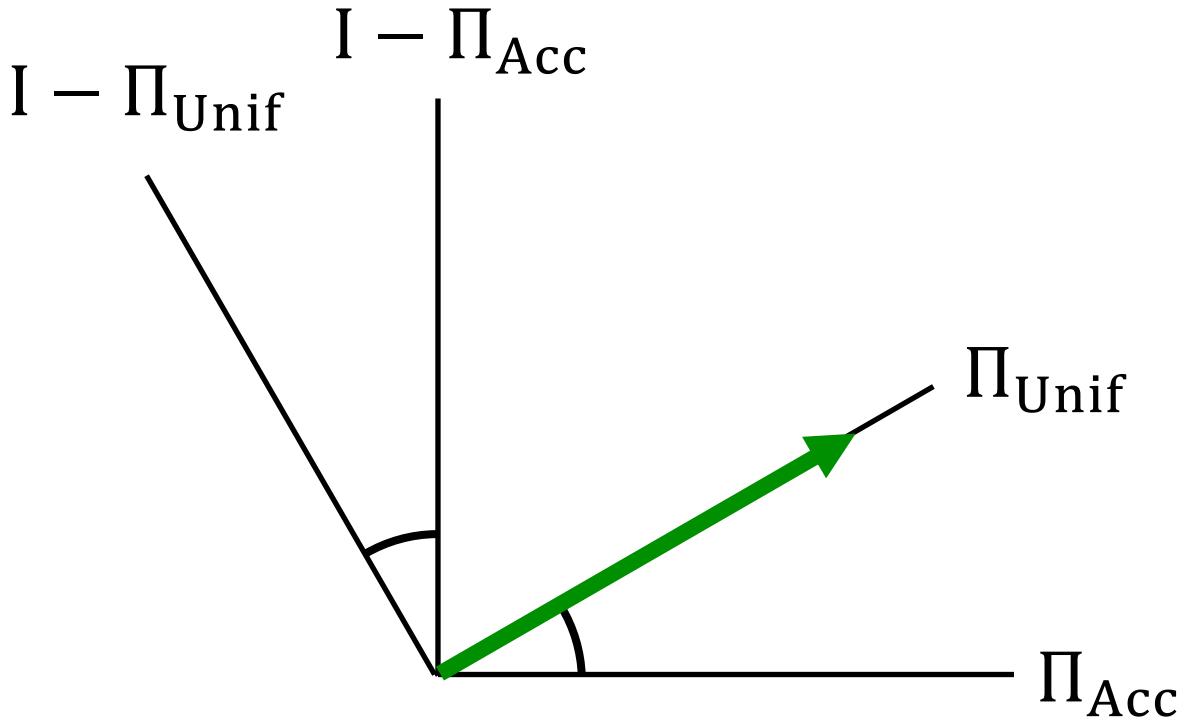
outcomes

Unif	Acc
$b_0 = 1$	$b_1 = 1$



Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

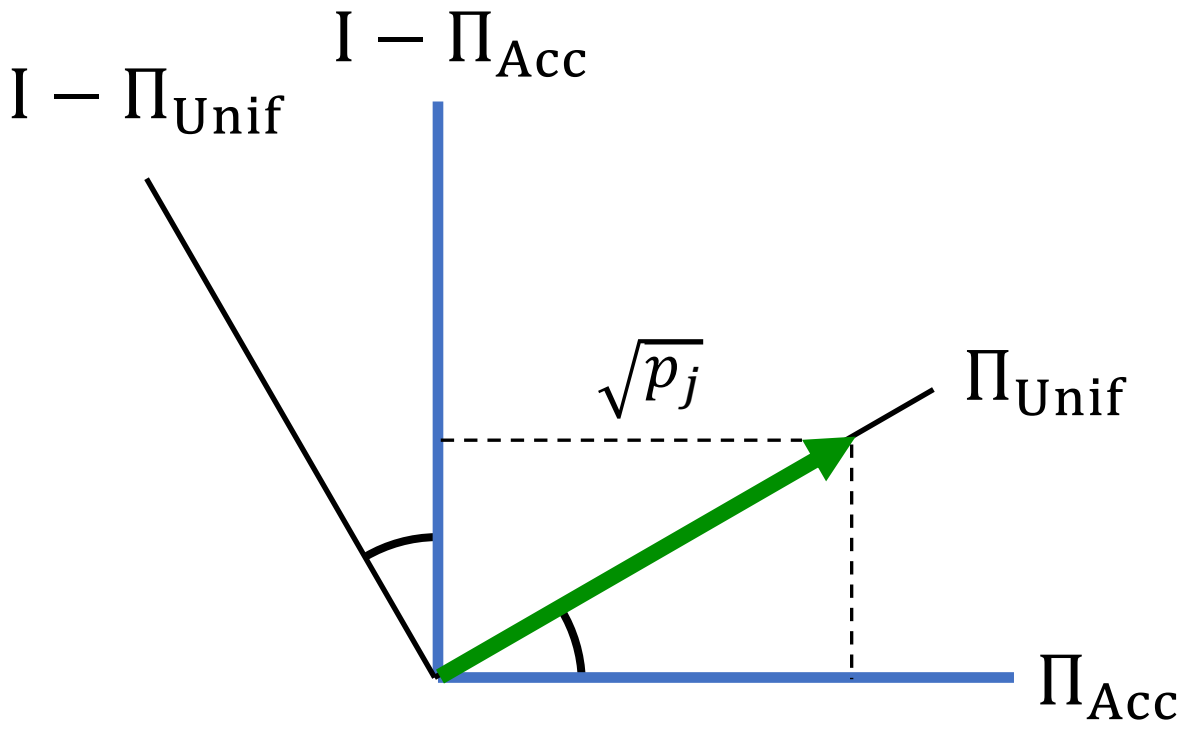


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$

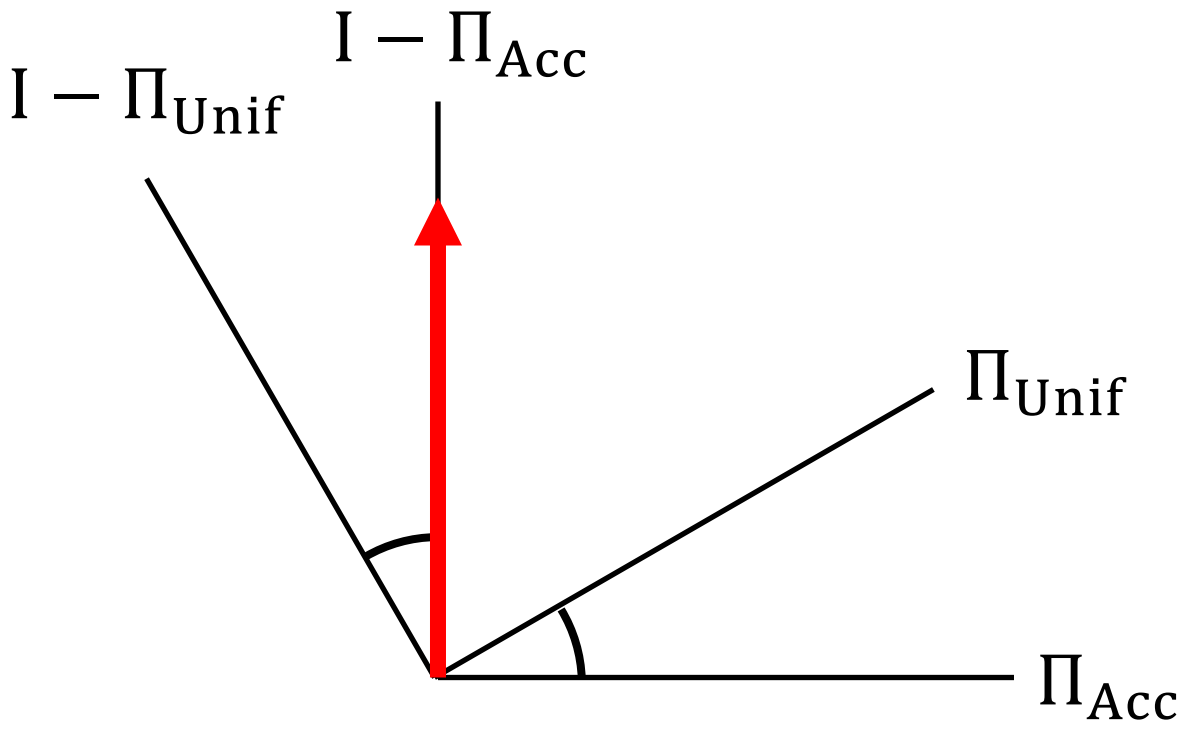


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$

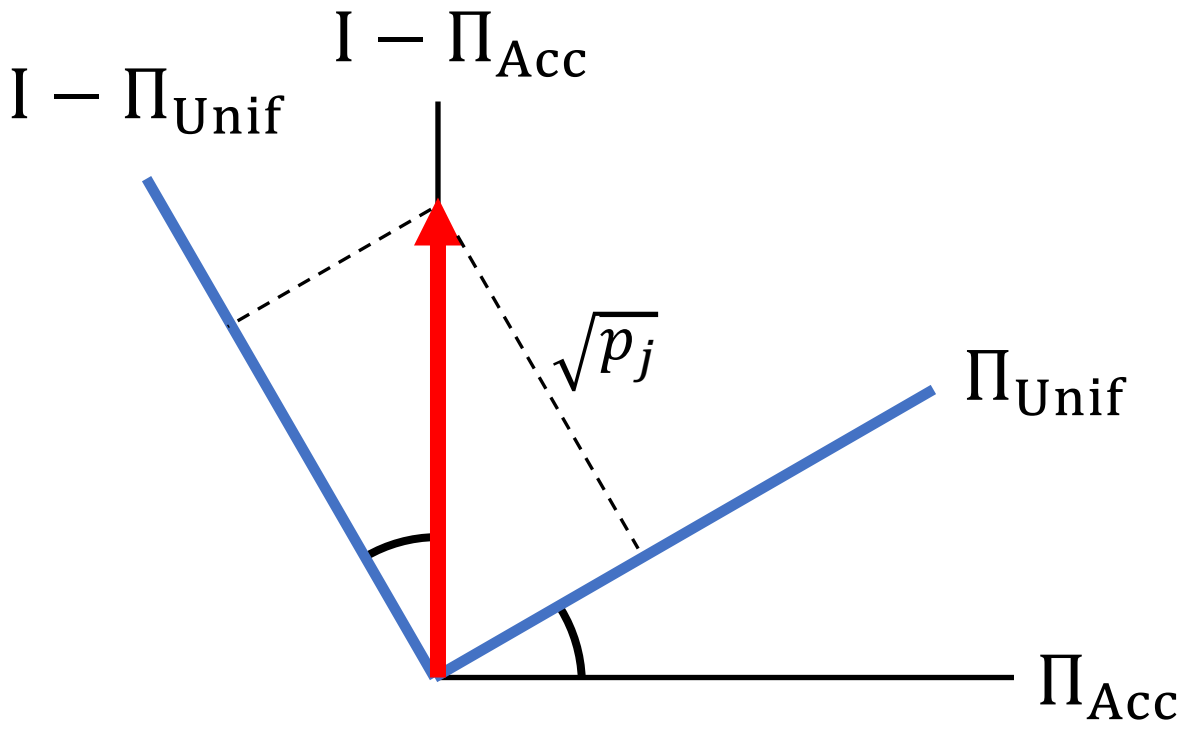


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif	Acc
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$

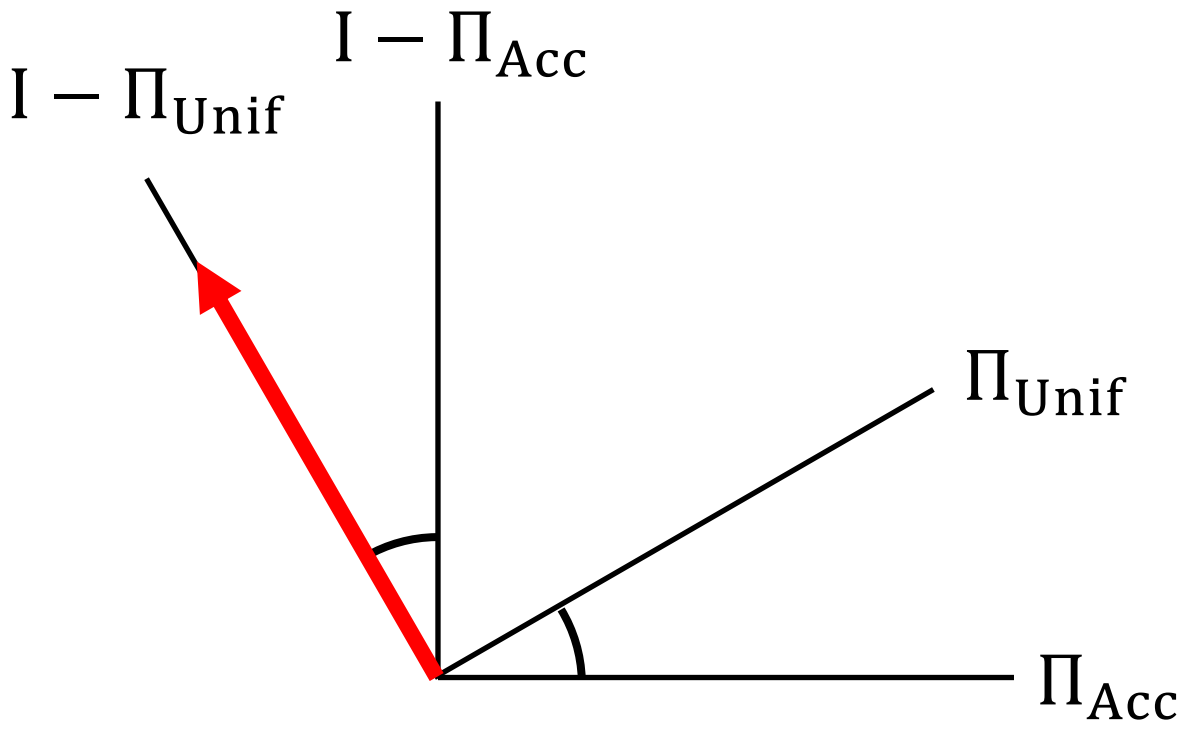


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif	Acc
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$

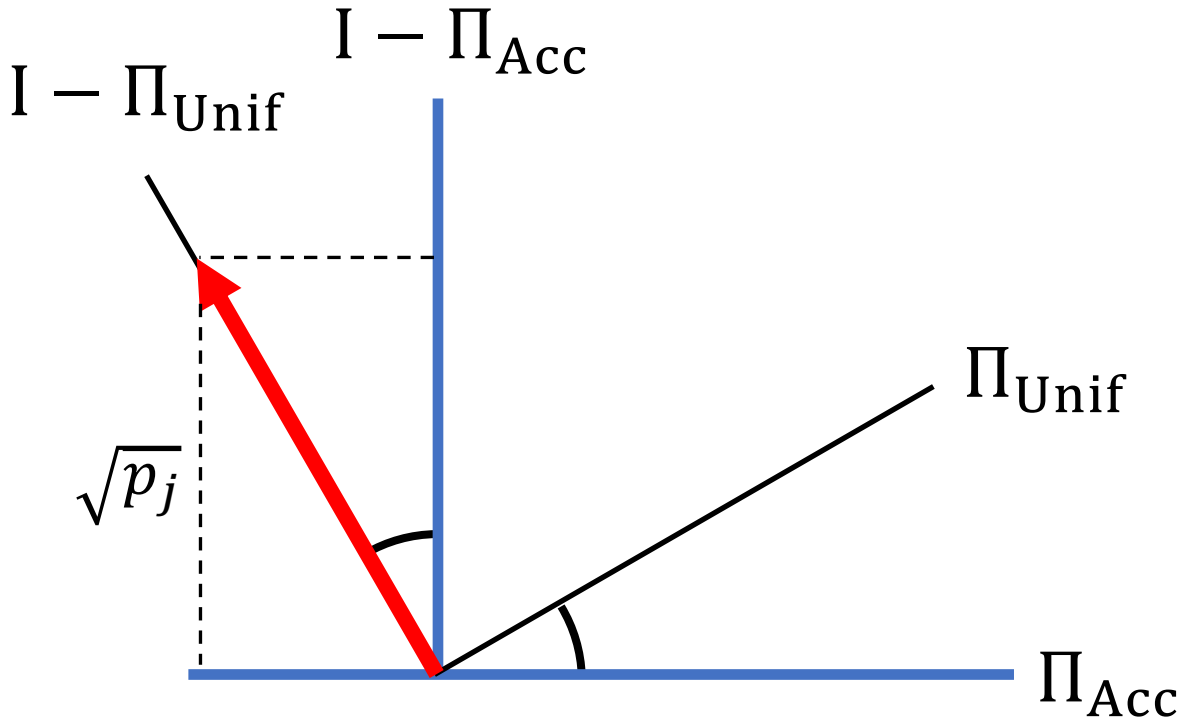


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif	Acc	Unif
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$	$b_4 = 0$

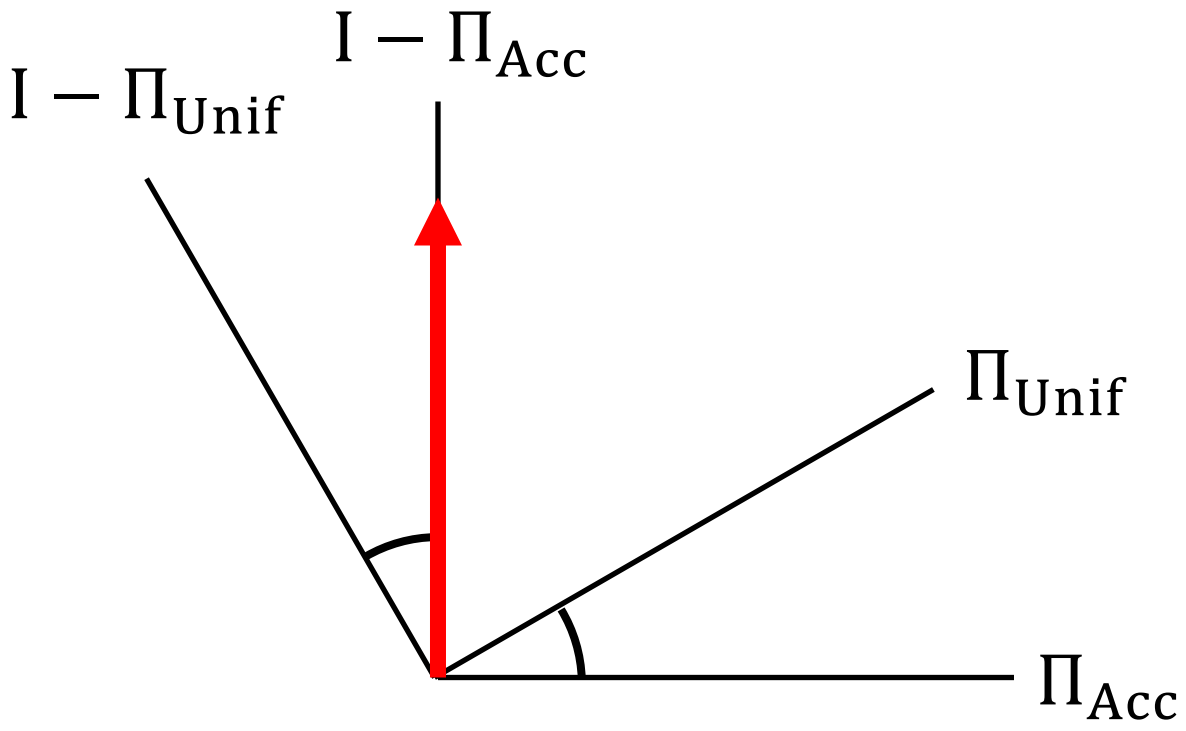


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif	Acc	Unif
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$	$b_4 = 0$

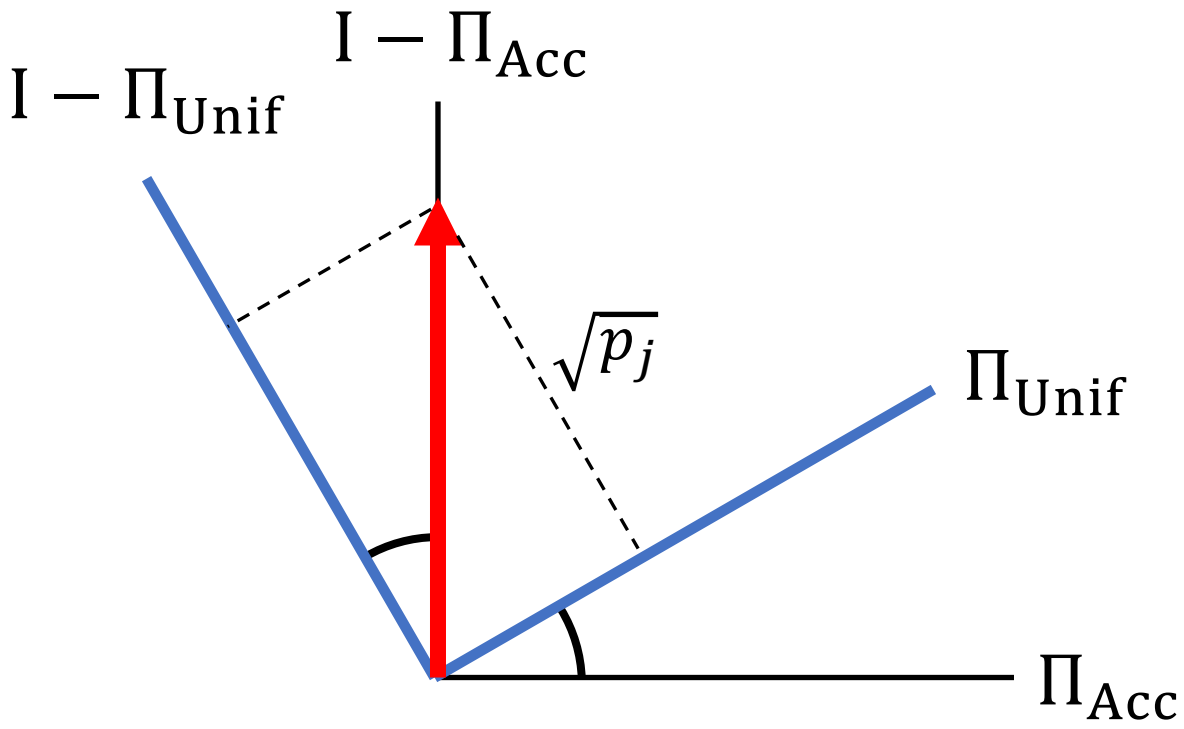


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif	Acc	Unif	Acc
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$	$b_4 = 0$	$b_5 = 0$

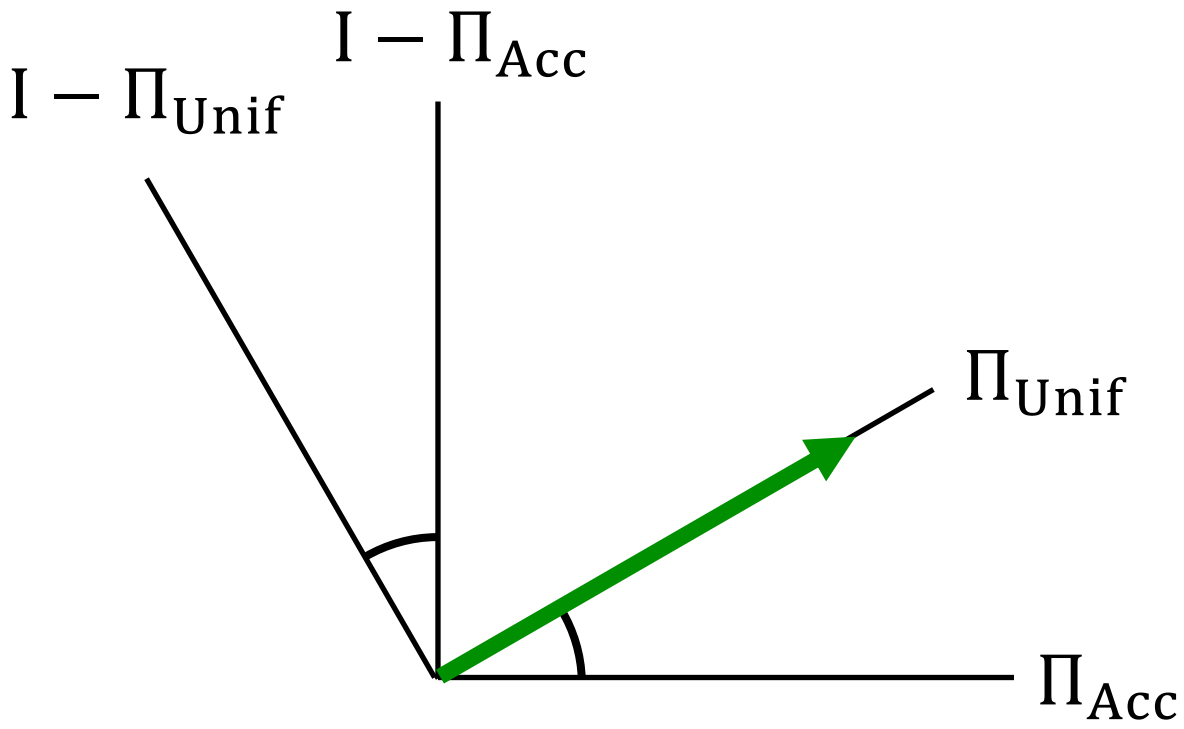


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif	Acc	Unif	Acc
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$	$b_4 = 0$	$b_5 = 0$

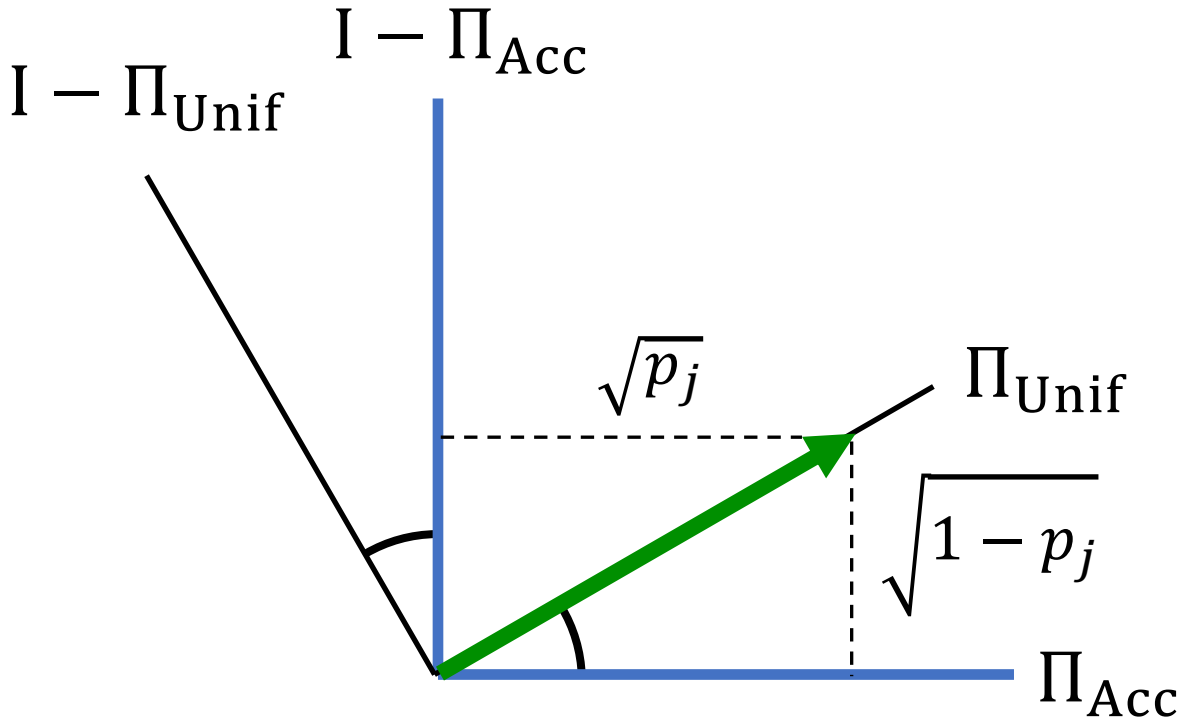


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif	Acc	Unif	Acc	Unif
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$	$b_4 = 0$	$b_5 = 0$	$b_6 = 1$

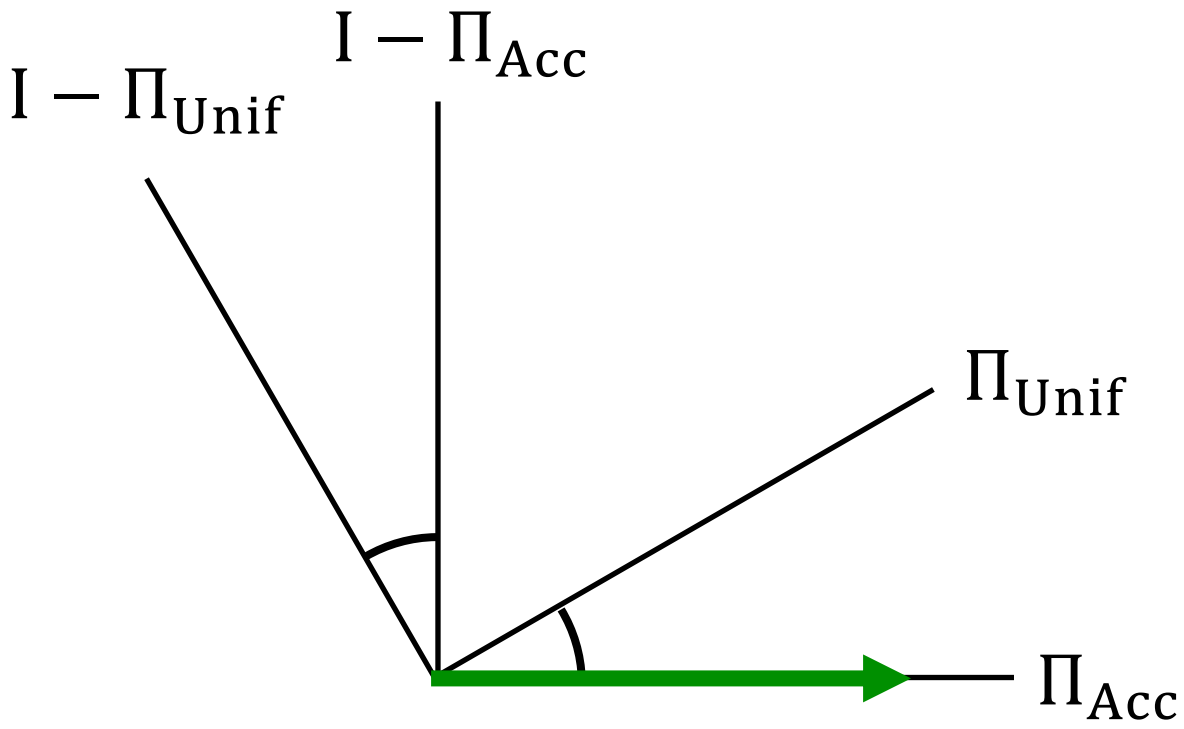


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif	Acc	Unif	Acc	Unif
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$	$b_4 = 0$	$b_5 = 0$	$b_6 = 1$

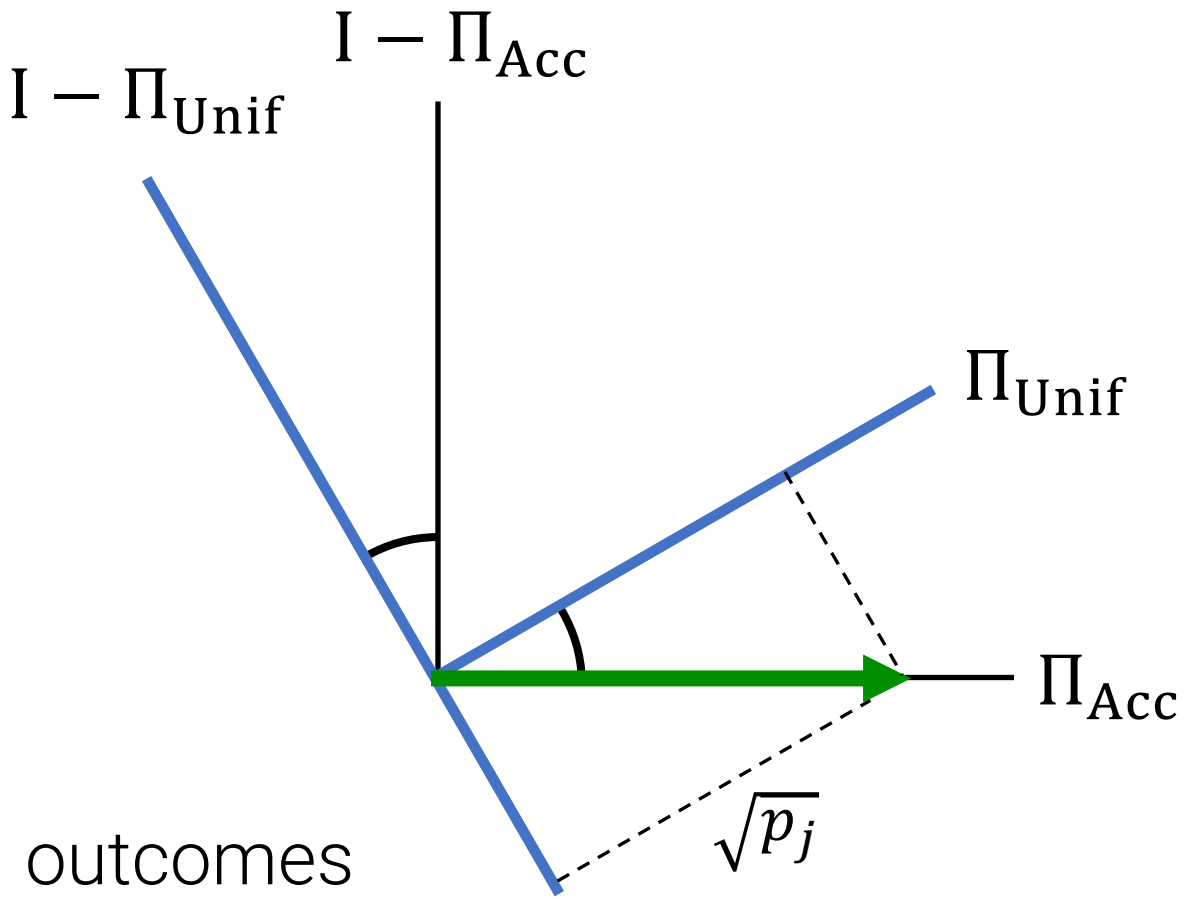


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

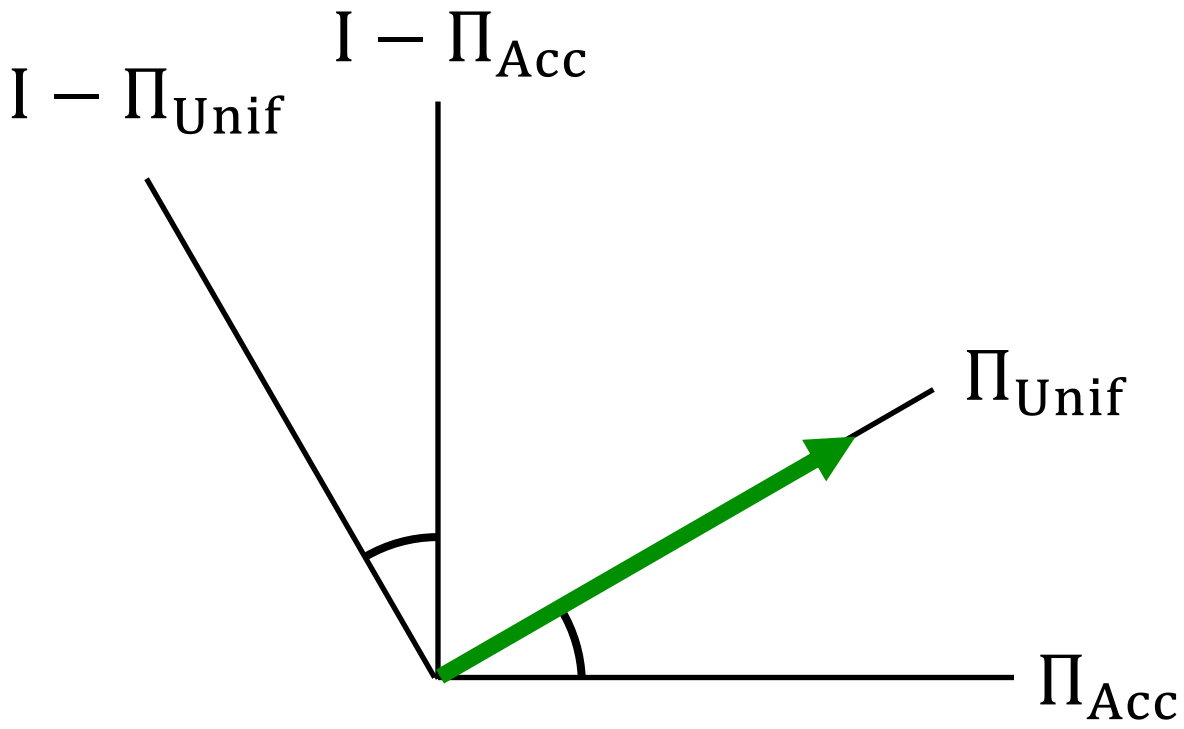
Unif	Acc	Unif	Acc	Unif	Acc	Unif	Acc
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$	$b_4 = 0$	$b_5 = 0$	$b_6 = 1$	$b_7 = 1$



Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

Unif	Acc	Unif	Acc	Unif	Acc	Unif	Acc
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$	$b_4 = 0$	$b_5 = 0$	$b_6 = 1$	$b_7 = 1$

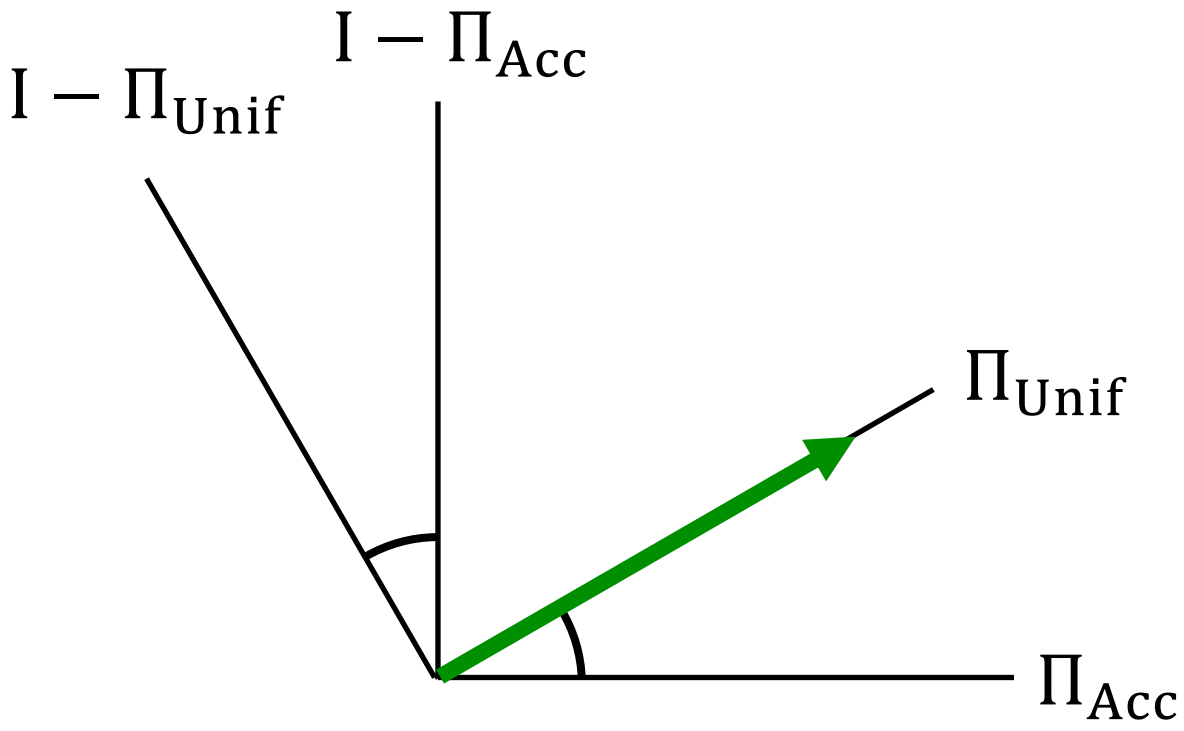


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

- p_j = success prob of $|\mathrm{S}\rangle$.
- alternating $\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}}$ measurements gives p_j

outcomes

Unif	Acc	Unif	Acc	Unif	Acc	Unif	Acc	Unif
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$	$b_4 = 0$	$b_5 = 0$	$b_6 = 1$	$b_7 = 1$	$b_8 = 1$

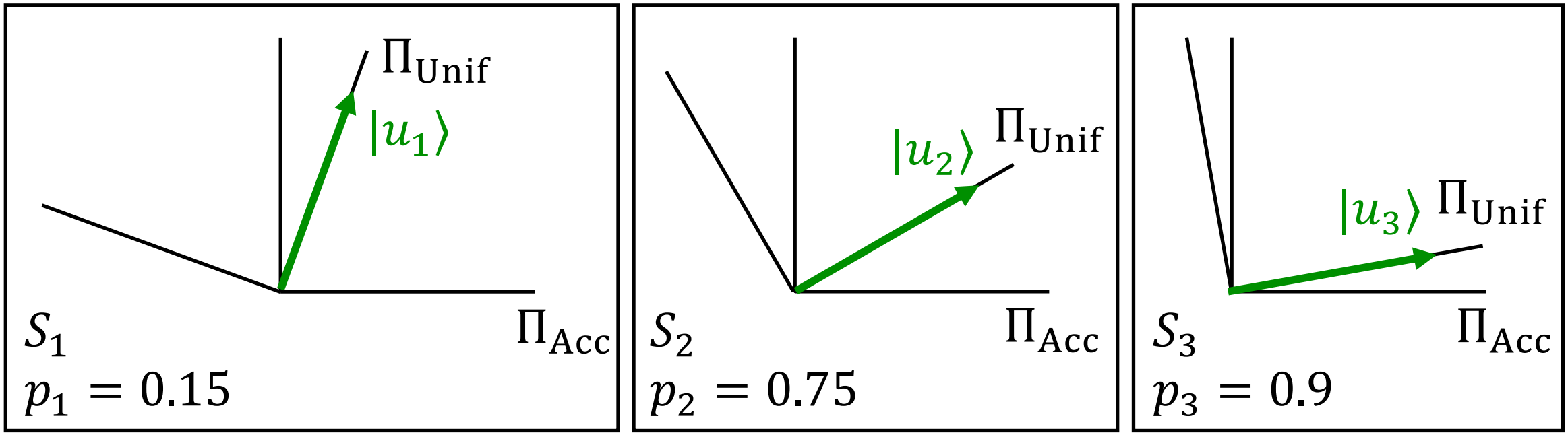


In this example, $b_i = b_{i+1}$ occurs 6 times out of 8, so we estimate $\approx 6/8$.

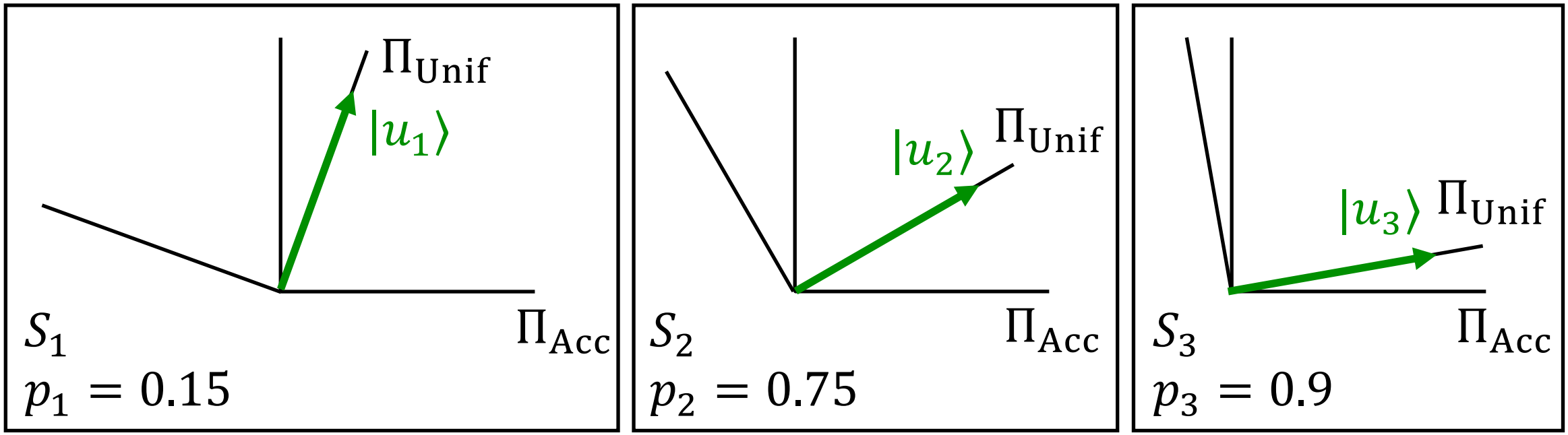
outcomes

Unif	Acc	Unif	Acc	Unif	Acc	Unif	Acc	Unif
$b_0 = 1$	$b_1 = 1$	$b_2 = 1$	$b_3 = 0$	$b_4 = 0$	$b_5 = 0$	$b_6 = 1$	$b_7 = 1$	$b_8 = 1$

In general, $|+\rangle_R \otimes |S\rangle$ can have components in more than one Jordan subspace S_j.



Suppose $|+\rangle_R \otimes |S\rangle = \alpha_1|u_1\rangle + \alpha_2|u_2\rangle + \alpha_3|u_3\rangle$.

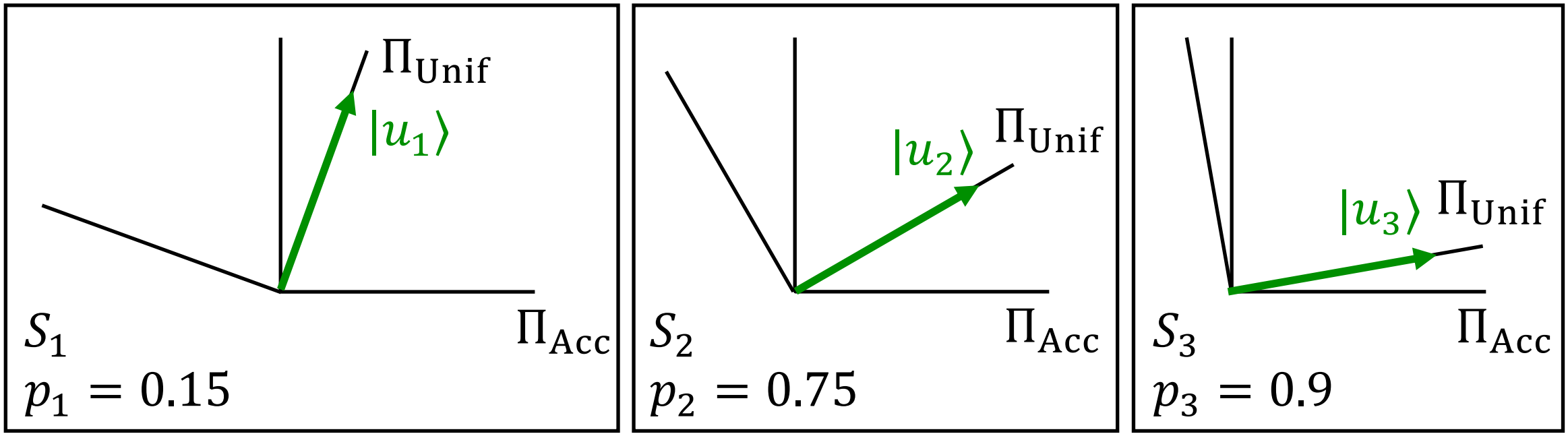


Suppose $|+\rangle_R \otimes |S\rangle = \alpha_1|u_1\rangle + \alpha_2|u_2\rangle + \alpha_3|u_3\rangle$.

success prob p_1

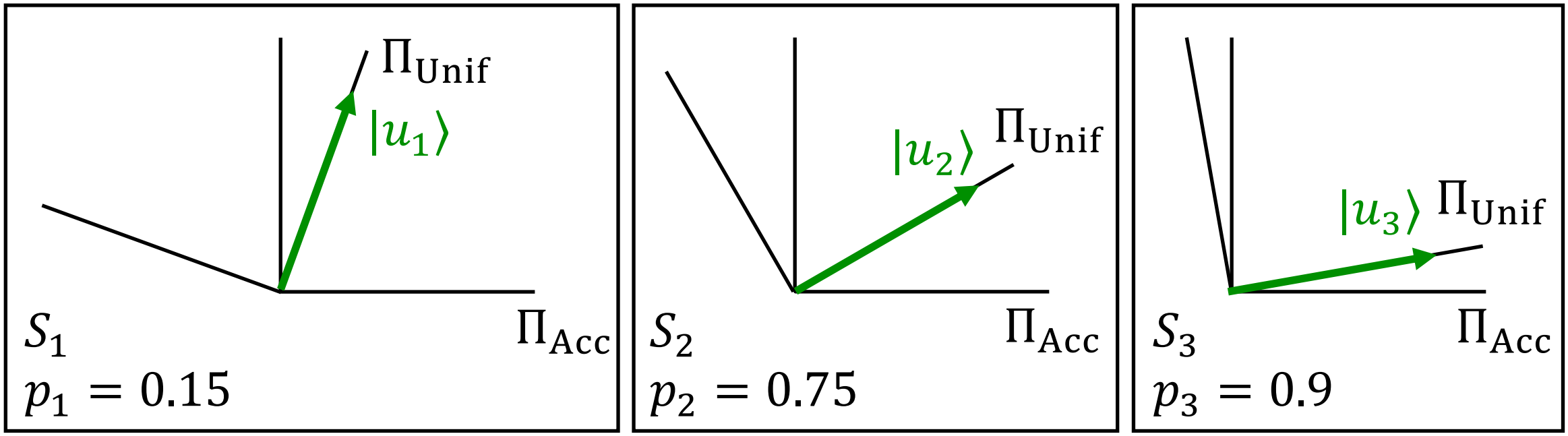
success prob p_2

success prob p_3



Suppose $|+\rangle_R \otimes |\mathrm{S}\rangle = \alpha_1|u_1\rangle + \alpha_2|u_2\rangle + \alpha_3|u_3\rangle$.

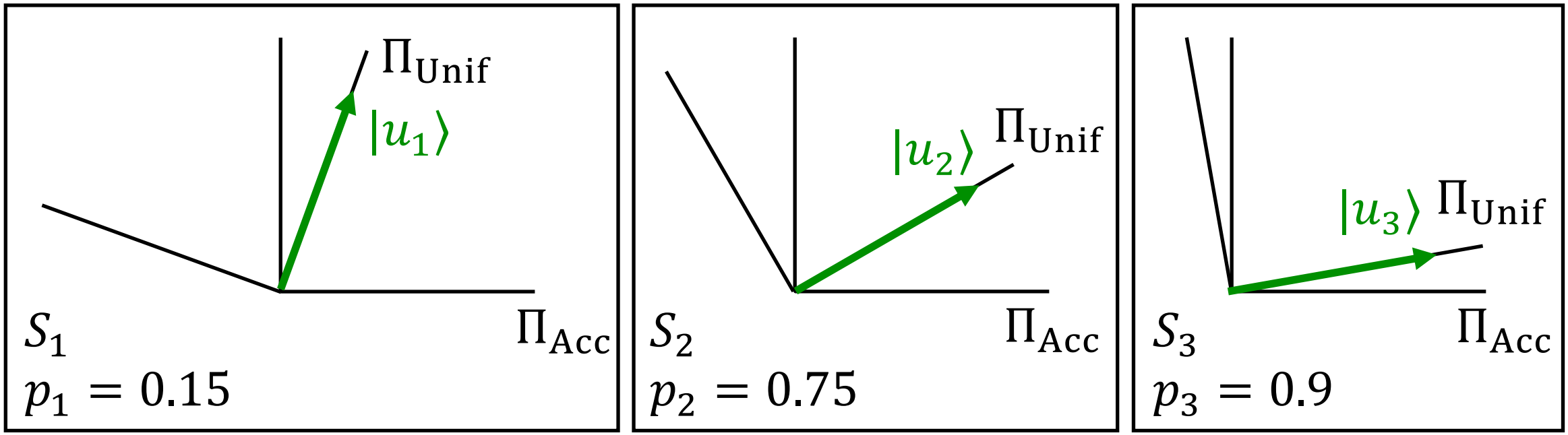
Key fact: Alternating measurement outcomes distributed as though $|+\rangle_R \otimes |\mathrm{S}\rangle$ were contained in S_j with prob $|\alpha_j|^2$.



Suppose $|+\rangle_R \otimes |S\rangle = \alpha_1|u_1\rangle + \alpha_2|u_2\rangle + \alpha_3|u_3\rangle$.

Key fact: Alternating measurement outcomes distributed as though $|+\rangle_R \otimes |S\rangle$ were contained in S_j with prob $|\alpha_j|^2$.

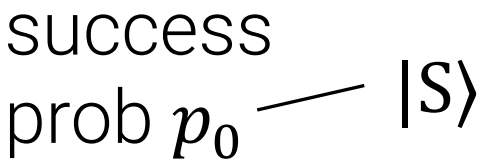
Leftover state concentrated on S_j's most consistent w/ outcomes.



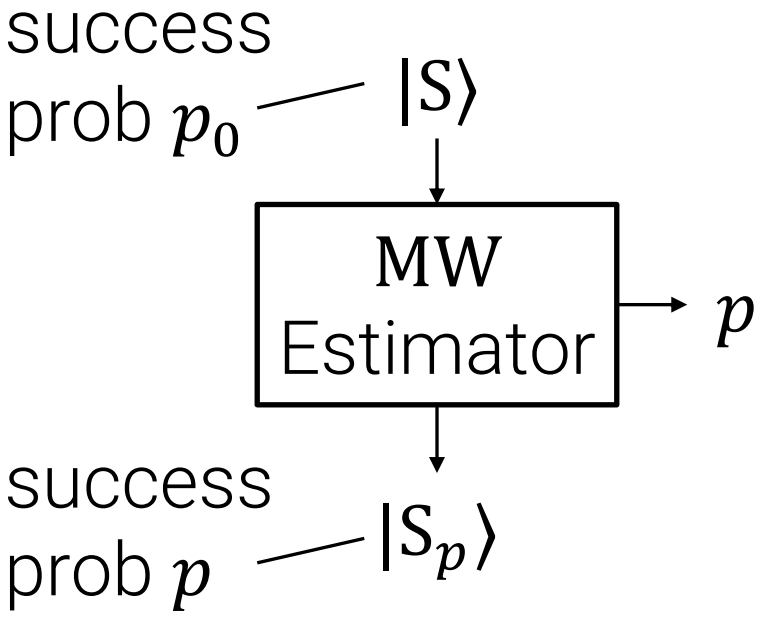
Suppose $|+\rangle_R \otimes |S\rangle = \alpha_1|u_1\rangle + \alpha_2|u_2\rangle + \alpha_3|u_3\rangle$.

[MW05] Estimation "approximately" projects onto $\{S_j\}$

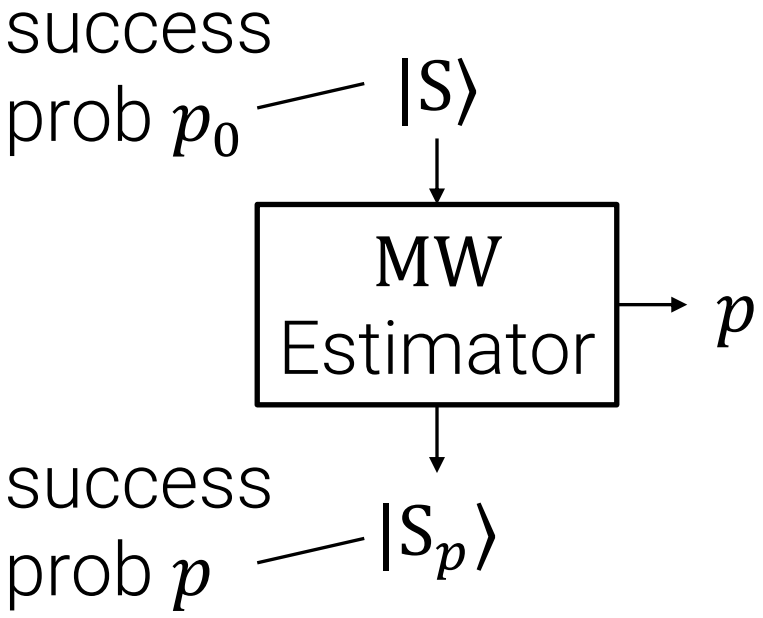
- w/ prob $\approx |\alpha_j|^2$ obtain estimate $\approx p_j$ and leftover state $\approx |u_j\rangle$



We'll need two key properties about the $\mathbf{MW}$ estimator.



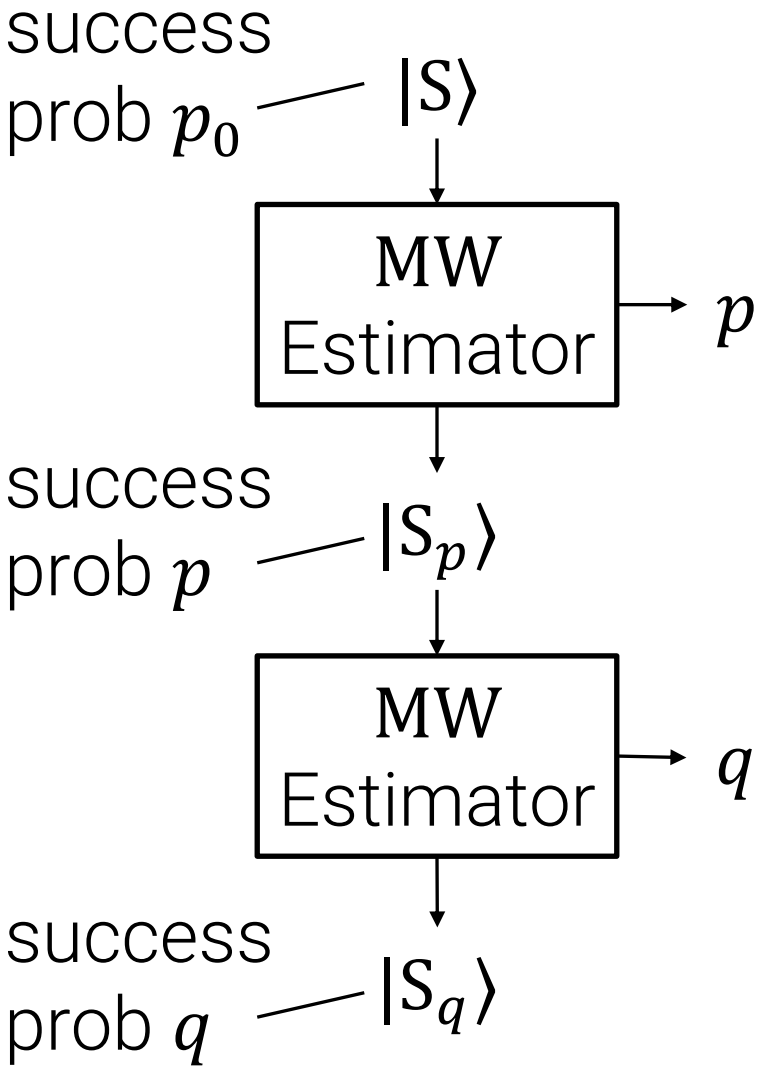
We'll need two key properties about the MW estimator.



We'll need two key properties about the MW estimator.

Key Properties

1) $\mathbb{E}[p] = p_0$

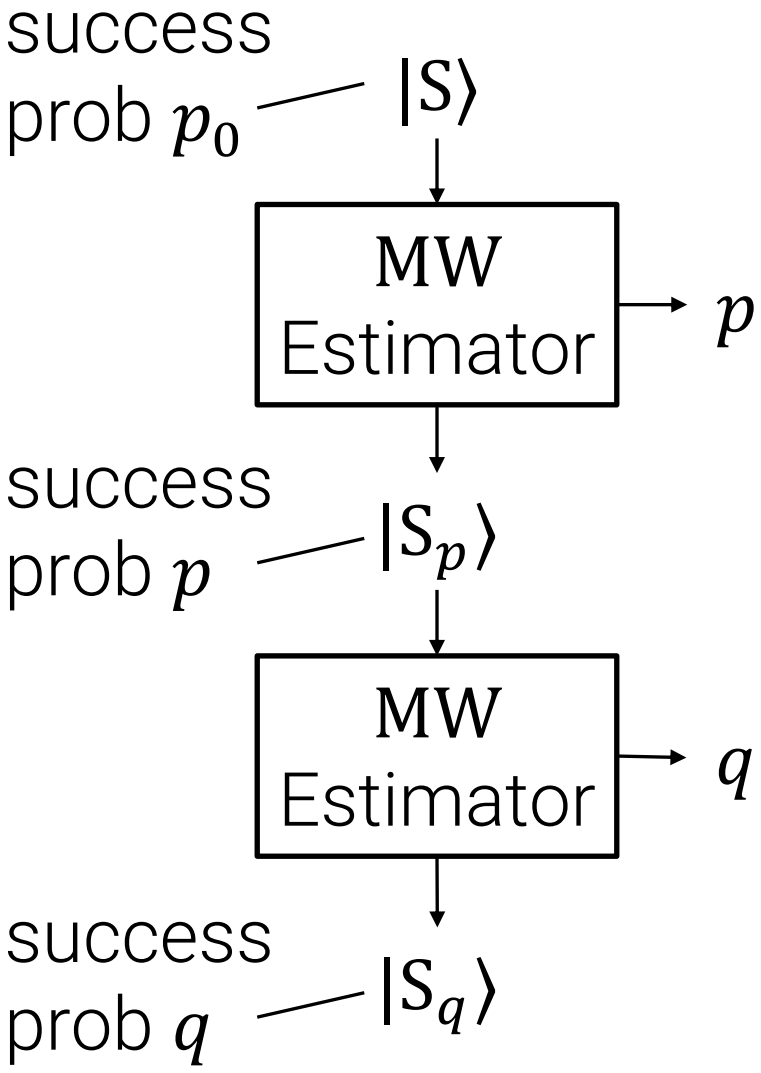


We'll need two key properties about the MW estimator.

Key Properties

1) $\mathbb{E}[p] = p_0$

2) If we apply MW twice, the two outcomes p, q are close with high probability.



We'll need two key properties about the $\mathbf{MW}$ estimator.

Key Properties

1) $\mathbb{E}[p] = p_0$

2) If we apply $\mathbf{MW}$ twice, the two outcomes p, q are close with high probability. Formally, $\mathbf{MW}$ achieves

$$\Pr[|p - q| \leq \varepsilon] \geq 1 - \delta$$

with $poly(\frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right))$ runtime.

For this talk, we'll need to know two things about the $\mathbf{MW}$ estimator.

As in [Zha20], we call this "(ε, δ)-almost-projective."

Key Properties

1) $\mathbb{E}[p] = p_0$

2) If we apply $\mathbf{MW}$ twice, the two outcomes p, q are close with high probability. Formally, $\mathbf{MW}$ achieves

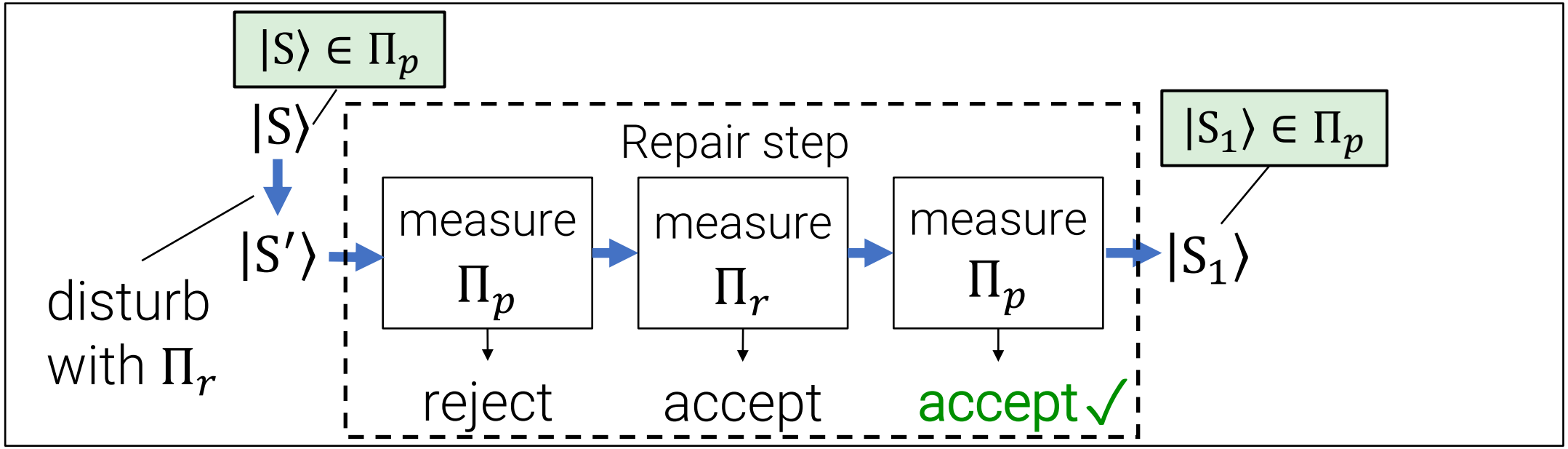
$$\Pr[|p - q| \leq \varepsilon] \geq 1 - \delta$$

with $poly(\frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right))$ runtime.

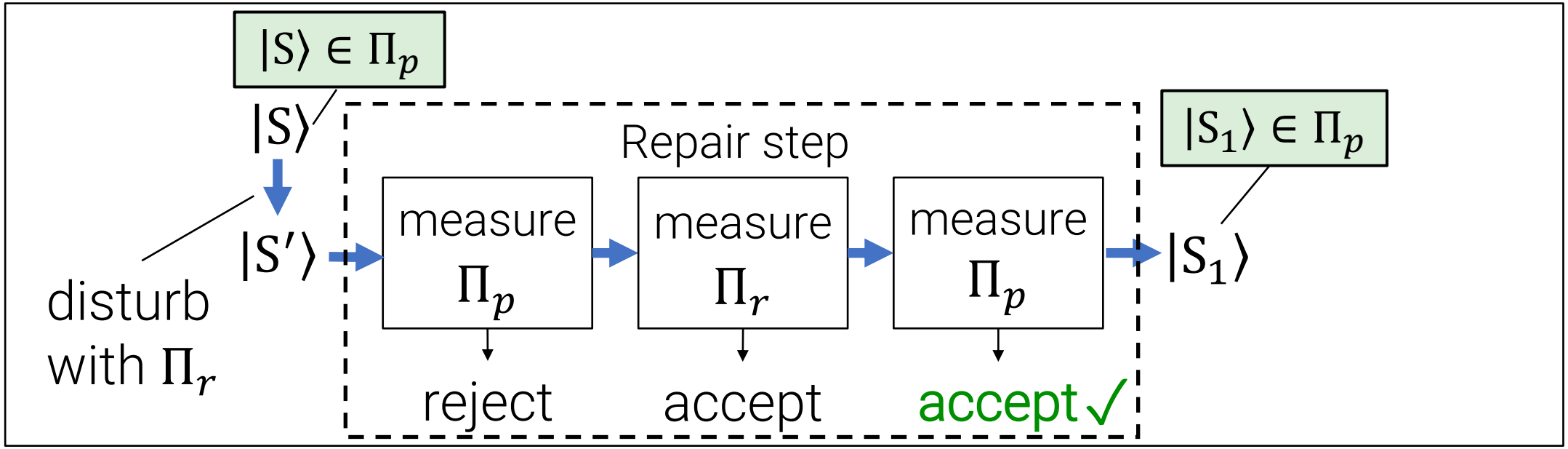
What this talk will cover:

1. Motivating example: Kilian's succinct arguments for NP
2. Why is post-quantum security of Kilian difficult?
3. Rewinding a quantum attacker many times
   - New idea: "repair" the adversary after each query
   - Estimating success probability
   - The full rewinding procedure
   - Analysis

Let’s see how [MW05] fits into our approach.



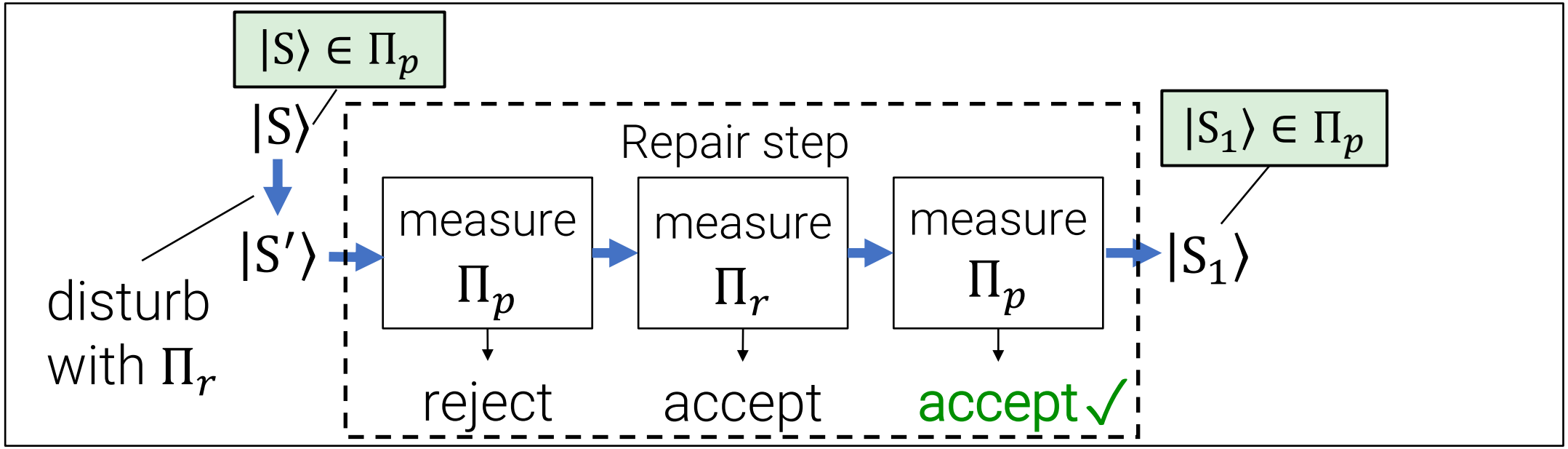
Recall: in our high-level sketch, we assumed we could ***exactly*** measure Π_p, i.e., whether success prob $\geq p$.



Recall: in our high-level sketch, we assumed we could ***exactly*** measure Π_p, i.e., whether success prob $\geq p$.

We don't know how to measure Π_p, but we can approximate it:

$\mathbf{MW}_p$**:** run the $\mathbf{MW}$ estimator and accept if the output is $\geq p$.

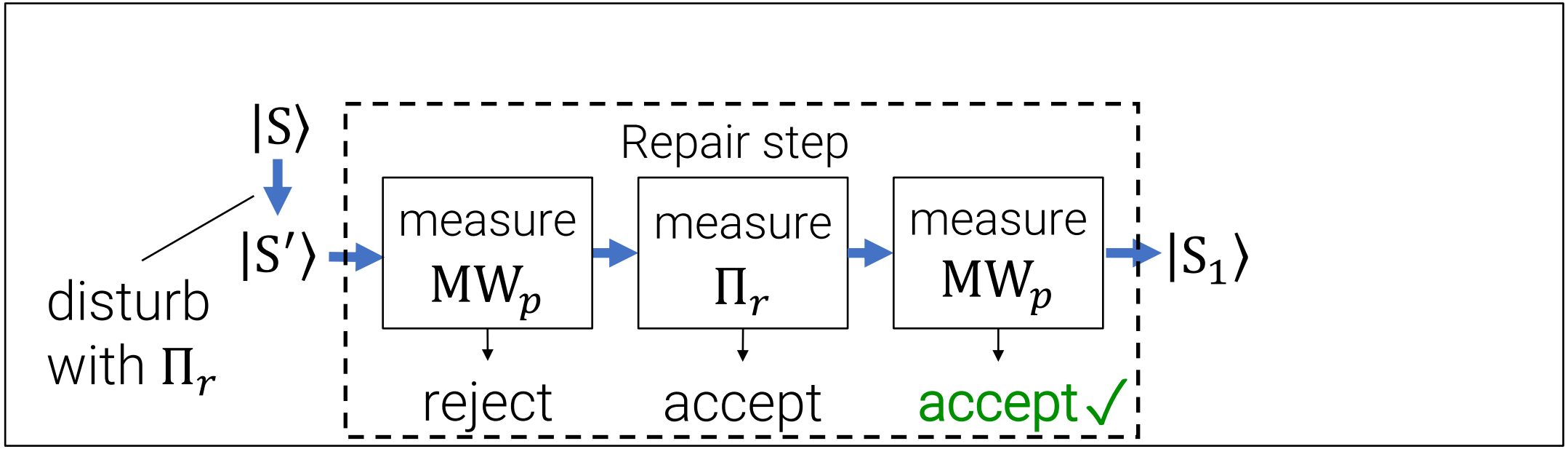


Recall: in our high-level sketch, we assumed we could ***exactly*** measure Π_p, i.e., whether success prob $\geq p$.

We don't know how to measure Π_p, but we can approximate it:

$\mathbf{MW}_p$**:** run the $\mathbf{MW}$ estimator and accept if the output is $\geq p$.

Idea: run Marriott-Watrous on Marriott-Watrous!

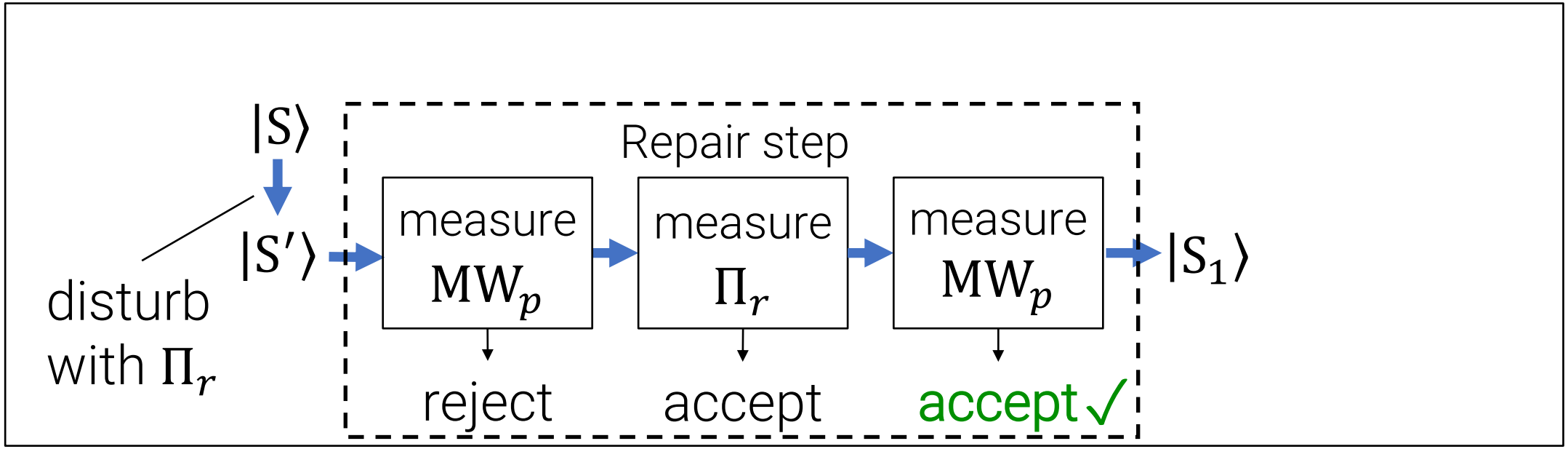


Recall: in our high-level sketch, we assumed we could ***exactly*** measure Π_p, i.e., whether success prob $\geq p$.

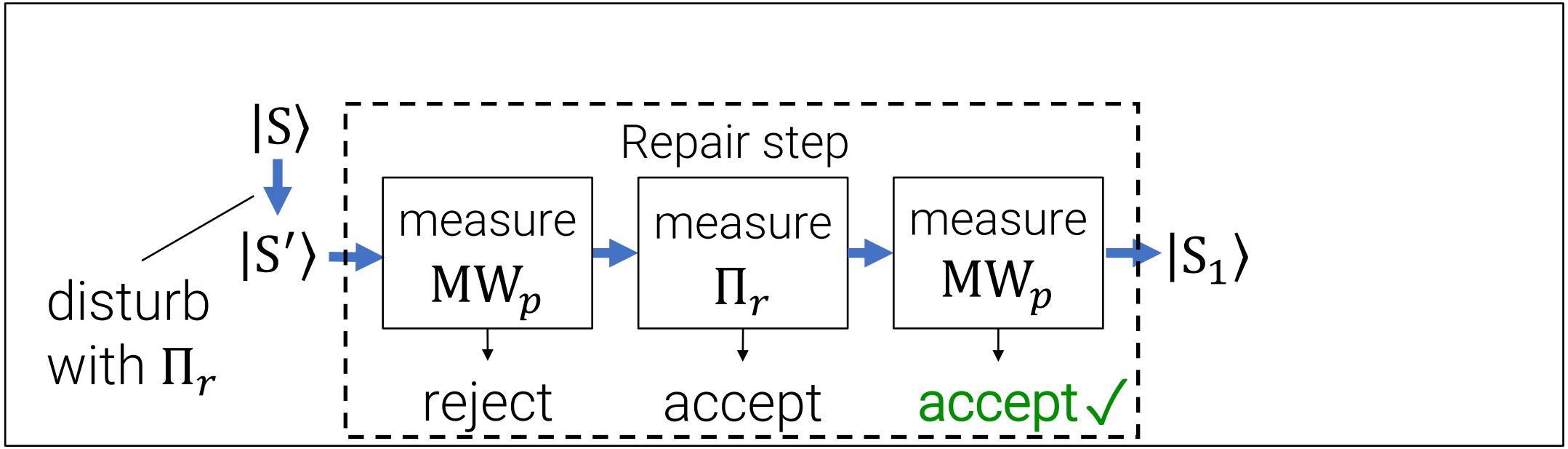
We don't know how to measure Π_p, but we can approximate it:

MW_p**:** run the MW estimator and accept if the output is $\geq p$.

Idea: run Marriott-Watrous on Marriott-Watrous!

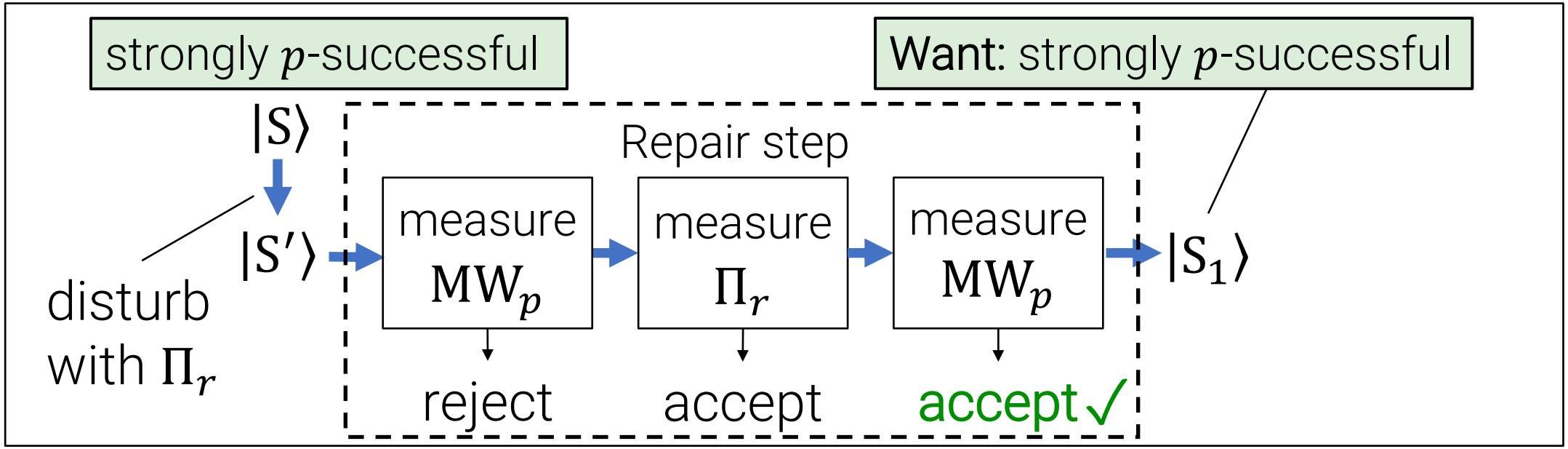


Subtle point: Just restoring "success probability" is not enough!



Subtle point: Just restoring "success probability" is not enough!

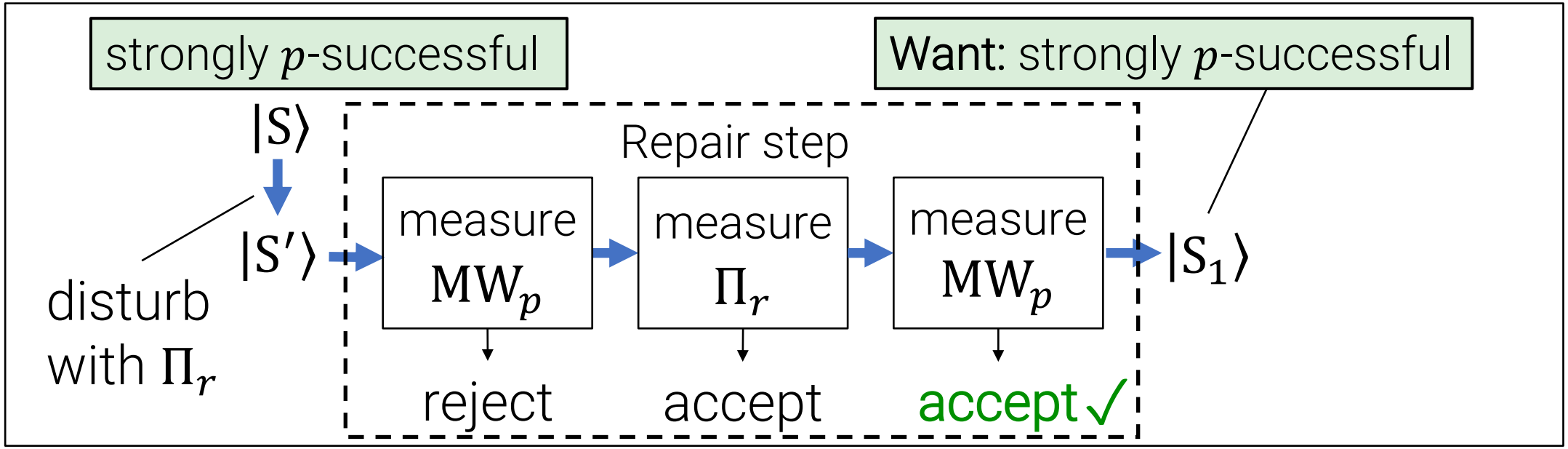
Definition: $|S\rangle$ is *strongly p-successful* if it is concentrated on $(\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}})$-Jordan subspaces with eigenvalue $\geq p$



Subtle point: Just restoring "success probability" is not enough!

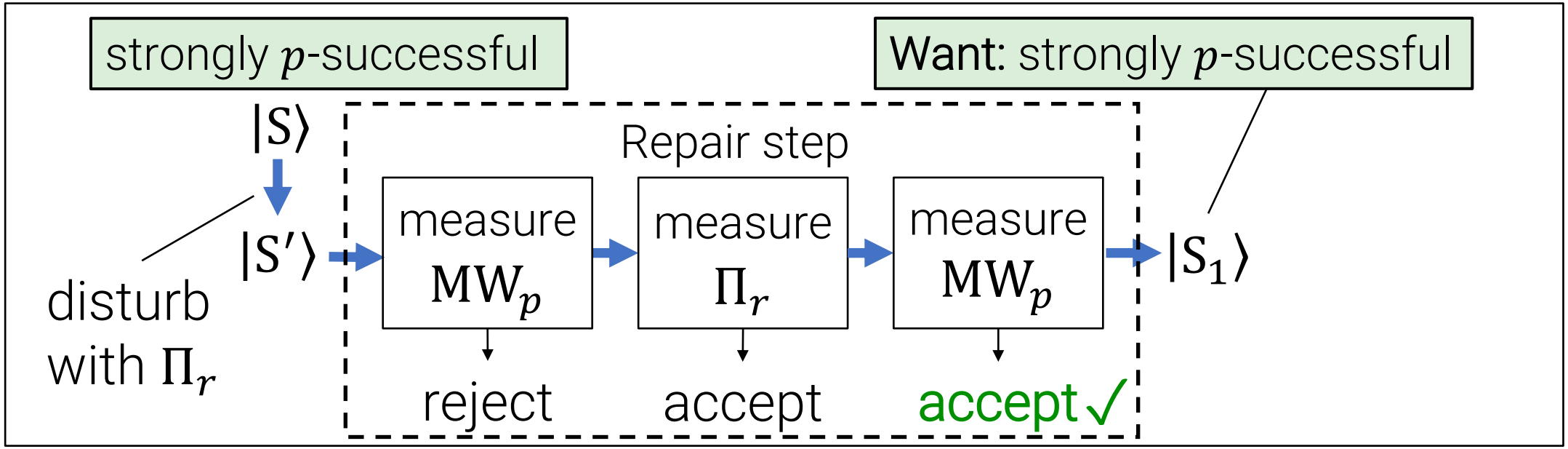
Definition: $|S\rangle$ is *strongly p-successful* if it is concentrated on $(\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}})$-Jordan subspaces with eigenvalue $\geq p$

We want: If $|S\rangle$ is strongly p-successful, then $|S_1\rangle$ is strongly p-successful



This seems promising, but we have a problem:

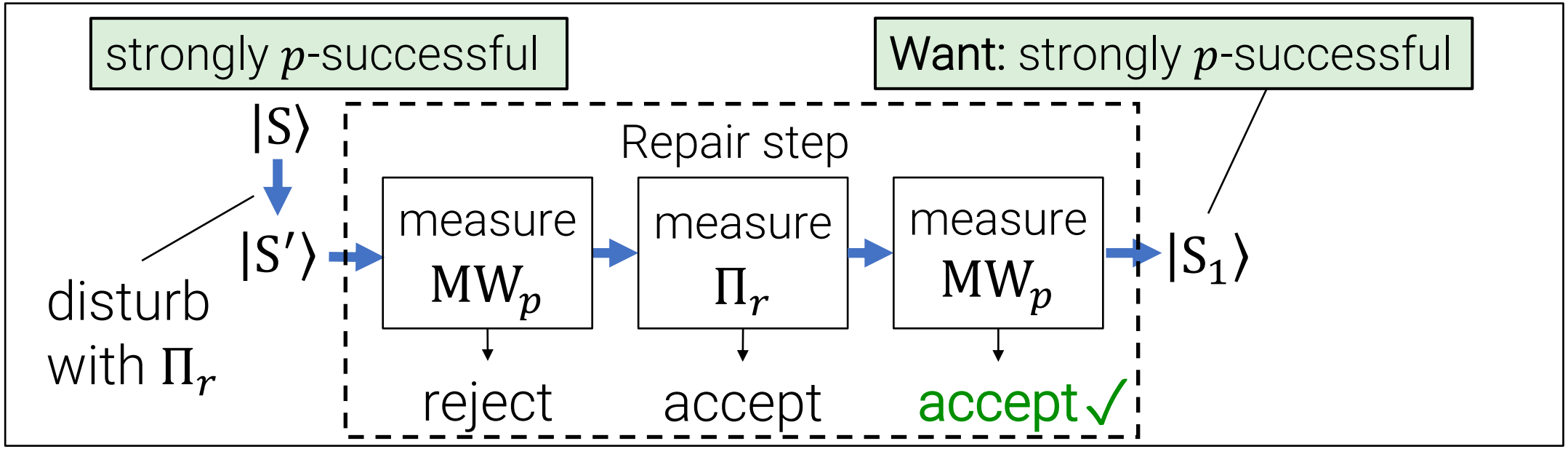
Our proof that this procedure terminates requires the measurements to be projective, but MW_p is not!



This seems promising, but we have a problem:

Our proof that this procedure terminates requires the measurements to be projective, but MW_p is not!

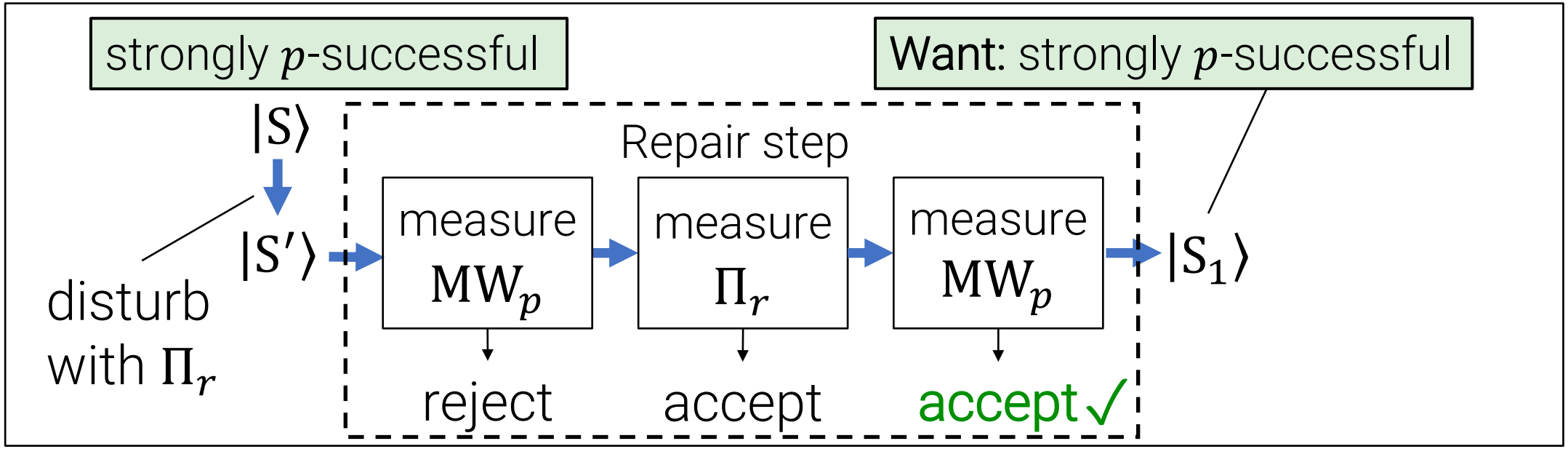
(running it twice may give different outcomes)



This seems promising, but we have a problem:

Our proof that this procedure terminates requires the measurements to be projective, but MW_p is not!

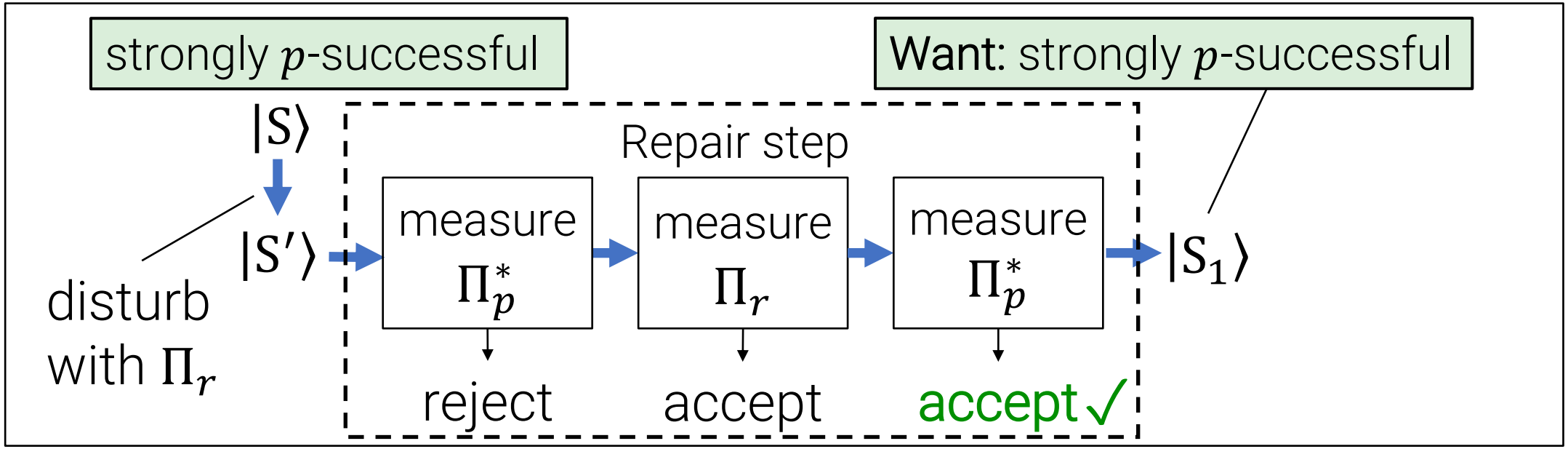
Easy(?) fix: Make MW_p projective by expanding the Hilbert space.



Measuring $|S'\rangle$ with MW_p can be implemented as a projective measurement of some Π_p^* on $|S'\rangle_A|0\rangle_W \in A \otimes W$.

adversary state register

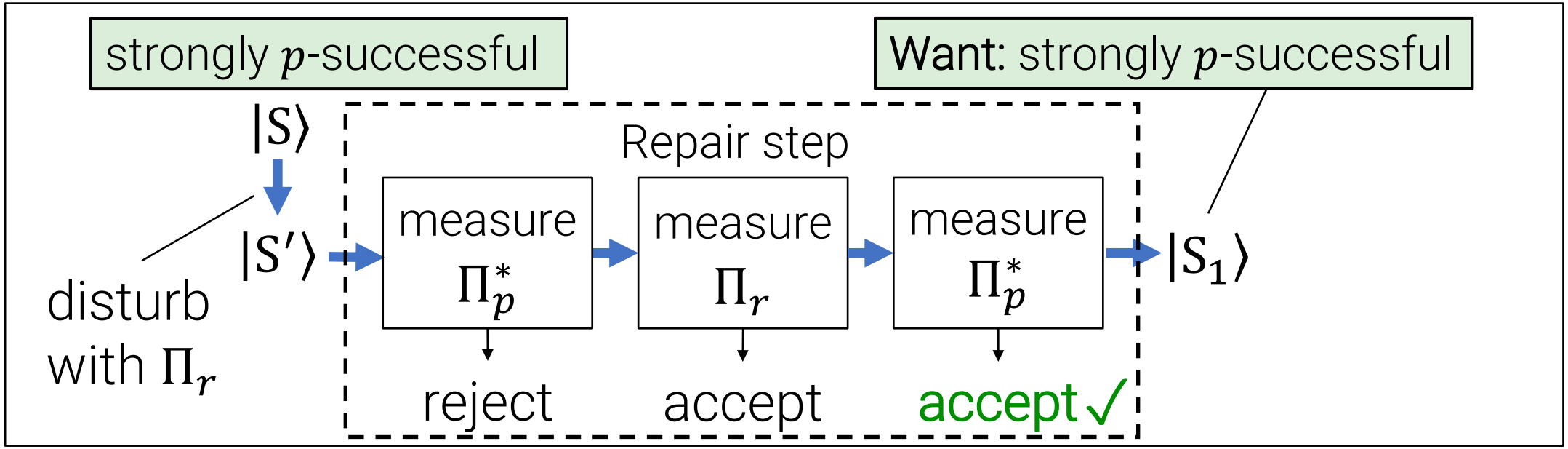
workspace/ancilla



Measuring $|S'\rangle$ with MW_p can be implemented as a projective measurement of some Π_p^* on $|S'\rangle_A|0\rangle_W \in A \otimes W$.

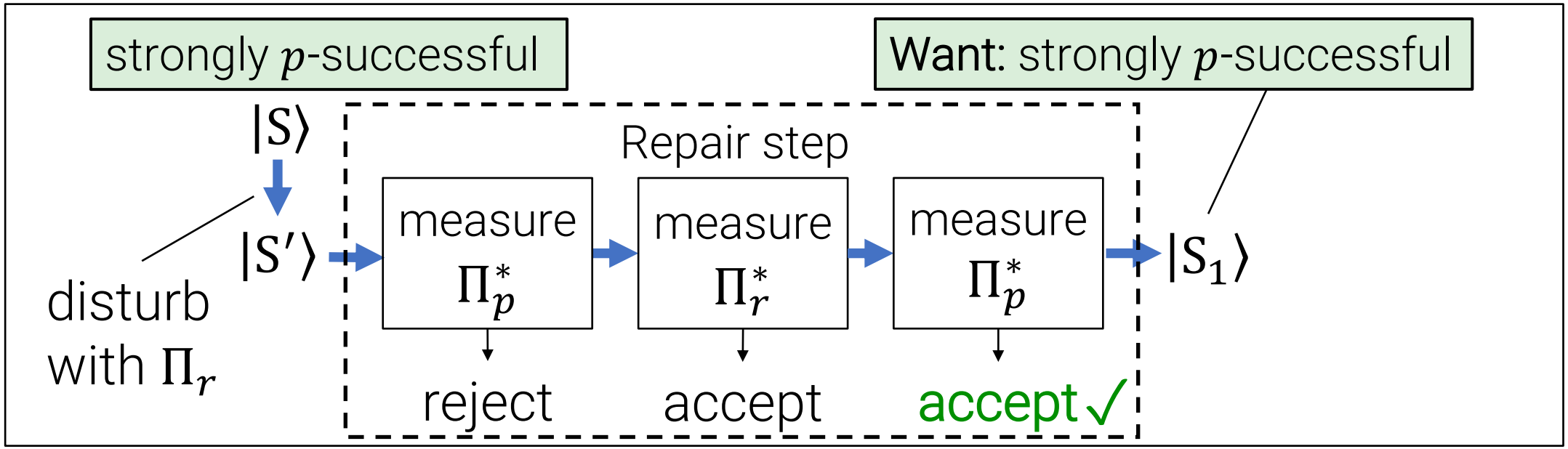
adversary state register

workspace/ancilla

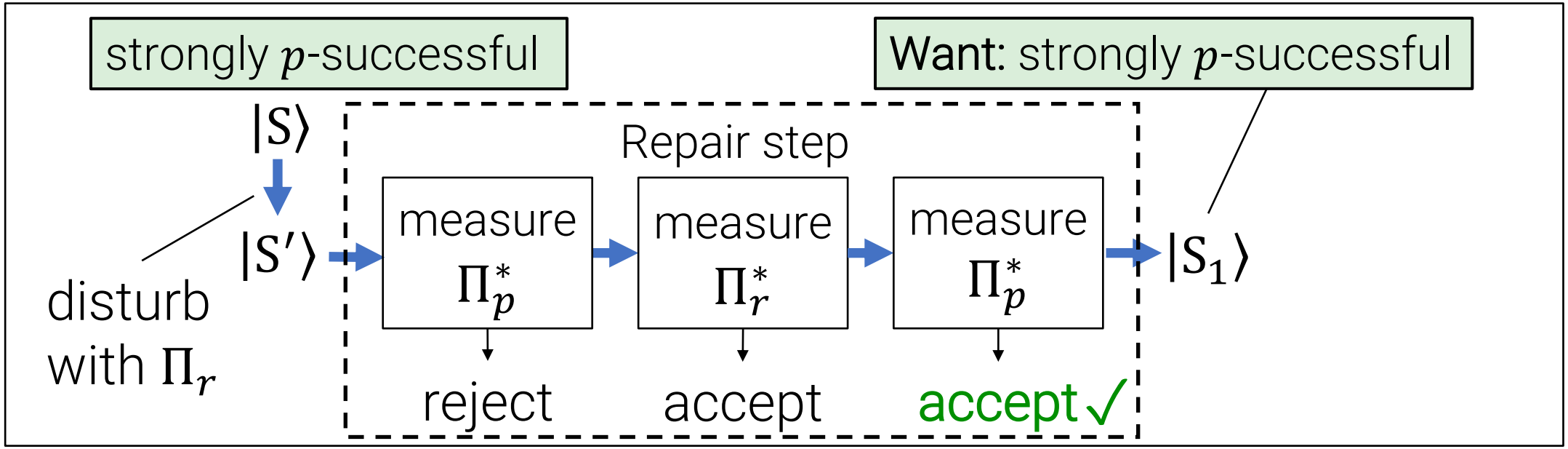


Measuring $|S'\rangle$ with $\mathbf{MW}_p$ can be implemented as a projective measurement of some Π_p^* on $|S'\rangle_A|0\rangle_W \in A \otimes W$.

But we need to be careful: Simply being in $\mathbf{image}(\Pi_p^*)$ doesn't tell us anything! If the ancilla is not $|0\rangle$, then measuring Π_p^* does not correspond to running $\mathbf{MW}_p$.

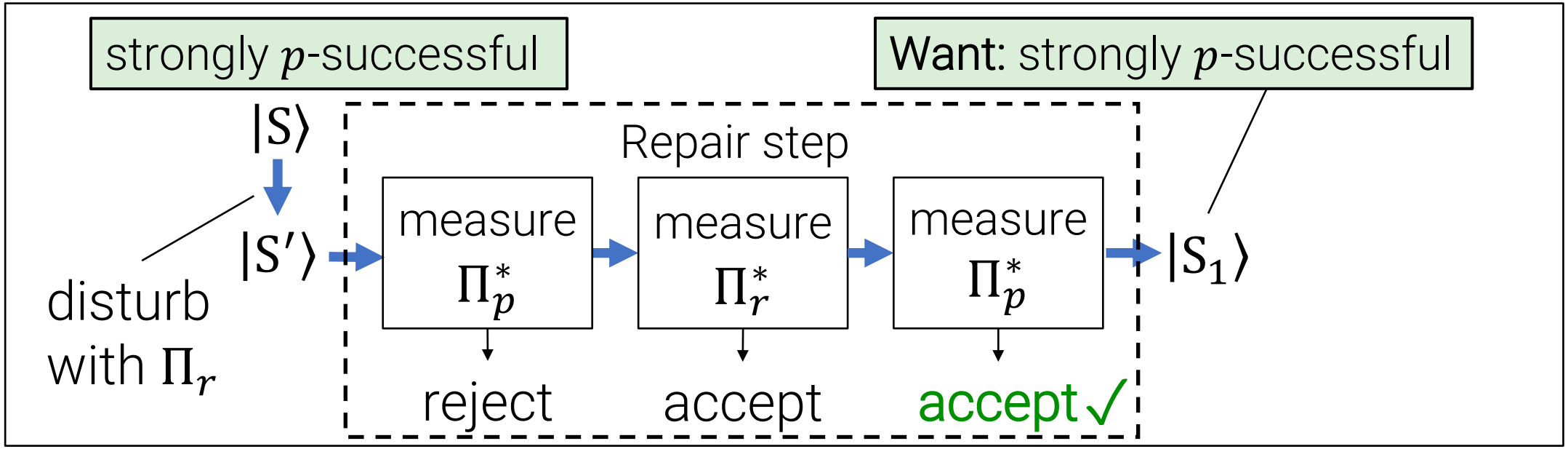


Our solution is re-define Π_r to $\Pi_r^* := \Pi_r \otimes |0\rangle\langle 0|_W$, so that each measurement of Π_r^* attempts to "reset" the W to $|0\rangle_W$.



This is essentially the full repair procedure!

Our solution is re-define Π_r to $\Pi_r^* := \Pi_r \otimes |0\rangle\langle 0|_W$, so that each measurement of Π_r^* attempts to "reset" the W to $|0\rangle_W$.



This is essentially the full repair procedure!

Our solution is re-define Π_r to $\Pi_r^* := \Pi_r \otimes |0\rangle\langle 0|_W$, so that each measurement of Π_r^* attempts to "reset" the W to $|0\rangle_W$.

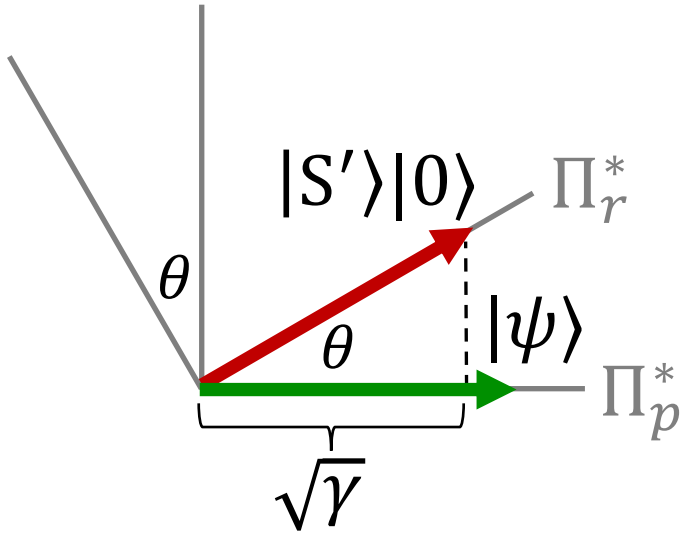
Not obvious: ***why*** does this choice of Π_r^* make repair work?

What this talk will cover:

1. Motivating example: Kilian's succinct arguments for NP
2. Why is post-quantum security of Kilian difficult?
3. Rewinding a quantum attacker many times
   - New idea: "repair" the adversary after each query
   - Estimating success probability
   - The full rewinding procedure
   - Analysis

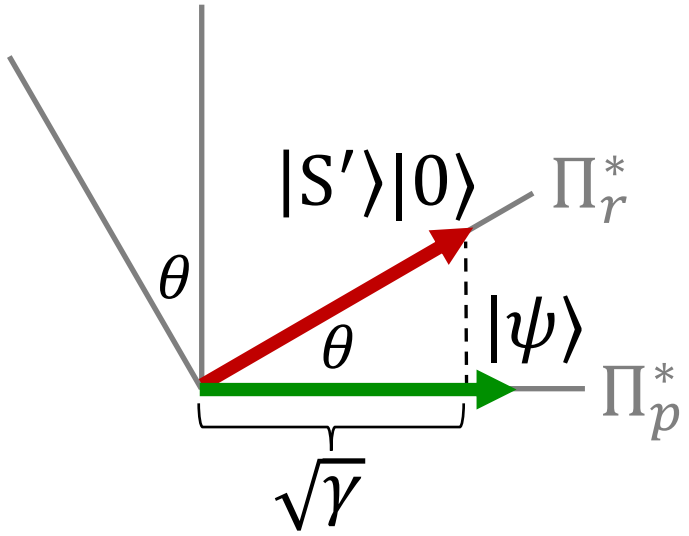
In a nutshell: $\Pi_r^* \coloneqq \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the $\mathbf{MW}$ procedure.

In a nutshell: $\Pi_r^* \coloneqq \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the $\mathbf{MW}$ procedure.



In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

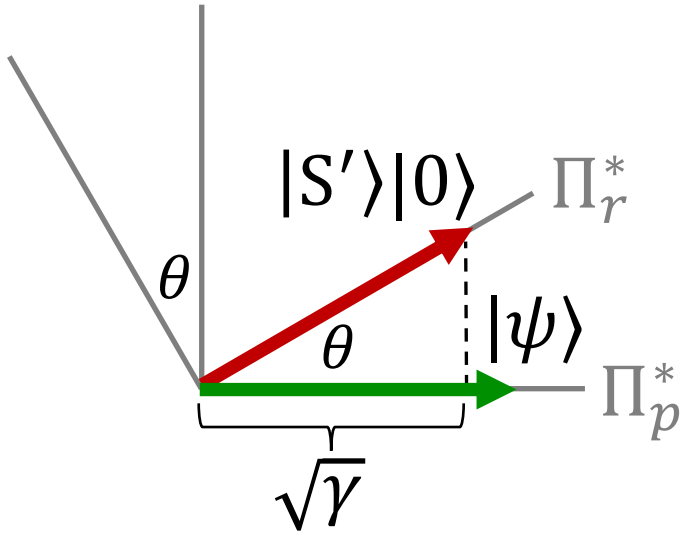
In a nutshell: $\Pi_r^* \coloneqq \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the MW procedure.



In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

Claim: ψ_A is a strongly $(p - \varepsilon)$-successful adversary state.

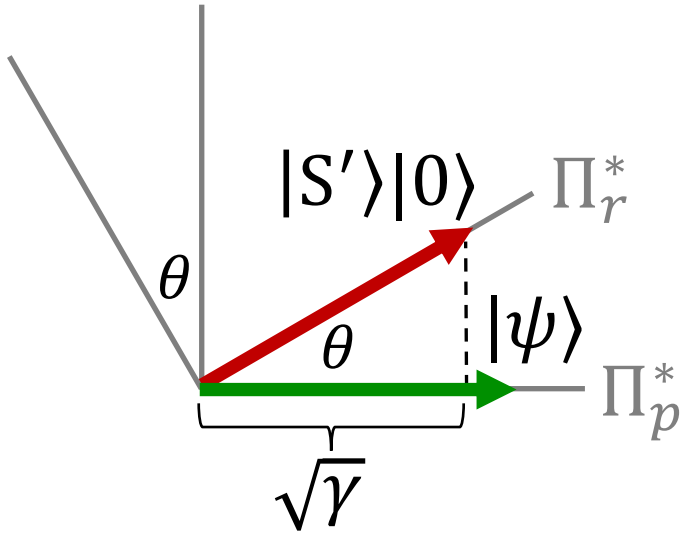
In a nutshell: $\Pi_r^* \coloneqq \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the $\mathbf{MW}$ procedure.



In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

Claim: ψ_A is a strongly $(p - \varepsilon)$-successful adversary state.

In a nutshell: $\Pi_r^* \coloneqq \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the $\mathbf{MW}$ procedure.

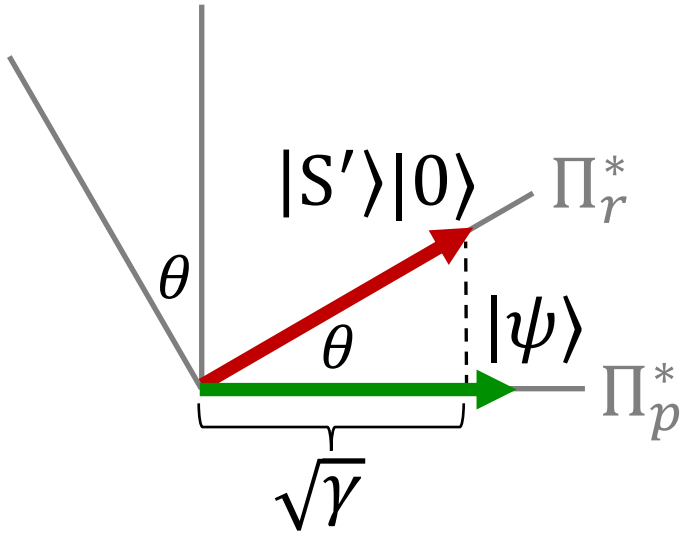


In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

Claim: ψ_A is a strongly $(p - \varepsilon)$-successful adversary state.

Proof Sketch

In a nutshell: $\Pi_r^* := \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the MW procedure.



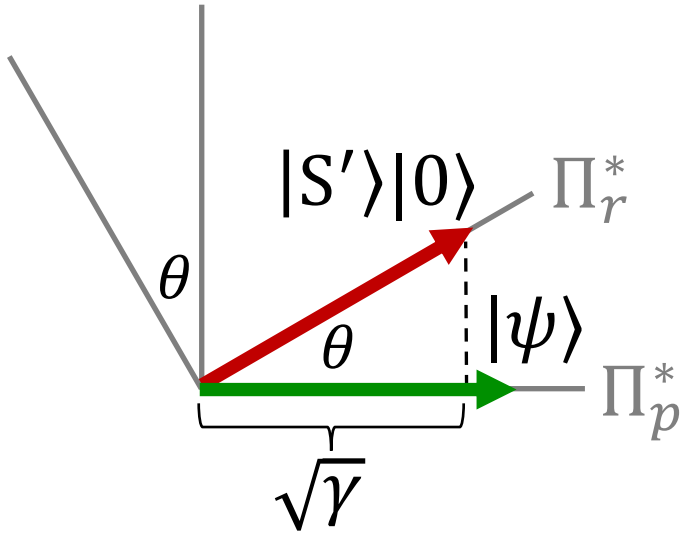
In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

Claim: ψ_A is a strongly $(p - \varepsilon)$-successful adversary state.

Proof Sketch

- If we run MW *twice* on $|S'\rangle$, the bad event (two estimates are $\geq \varepsilon$ apart) occurs with probability $\leq \delta$.

In a nutshell: $\Pi_r^* := \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the MW procedure.



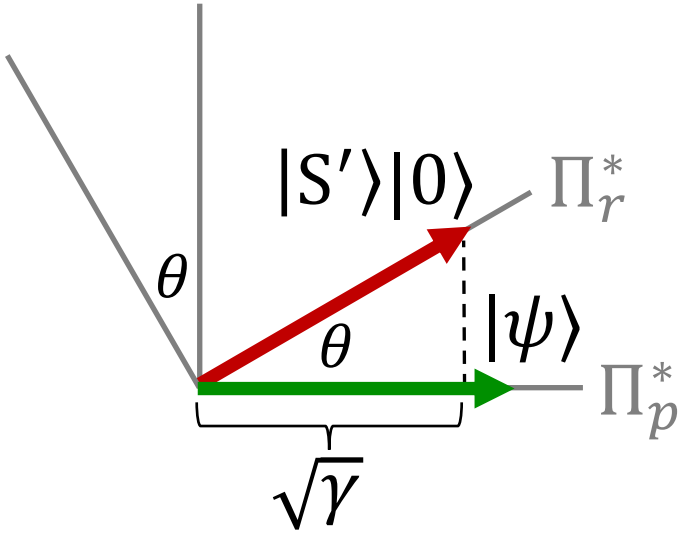
In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

Claim: ψ_A is a strongly $(p - \varepsilon)$-successful adversary state.

Proof Sketch

- If we run MW *twice* on $|S'\rangle$, the bad event (two estimates are $\geq \varepsilon$ apart) occurs with probability $\leq \delta$.
- $|\psi\rangle \propto \Pi_p^*|S'\rangle|0\rangle$ is the state after running MW on $|S'\rangle$ *once* and conditioning on output $\geq p$

In a nutshell: $\Pi_r^* := \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the MW procedure.



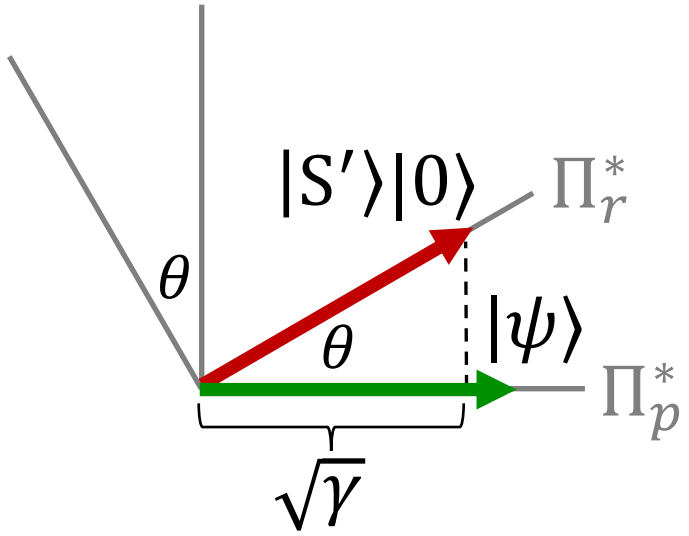
In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

Claim: ψ_A is a strongly $(p-\varepsilon)$-successful adversary state.

Proof Sketch

- If we run MW *twice* on $|S'\rangle$, the bad event (two estimates are $\geq \varepsilon$ apart) occurs with probability $\leq \delta$.
- $|\psi\rangle \propto \Pi_p^*|S'\rangle|0\rangle$ is the state after running MW on $|S'\rangle$ *once* and conditioning on output $\geq p$, which occurs with probability γ.

In a nutshell: $\Pi_r^* := \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the MW procedure.



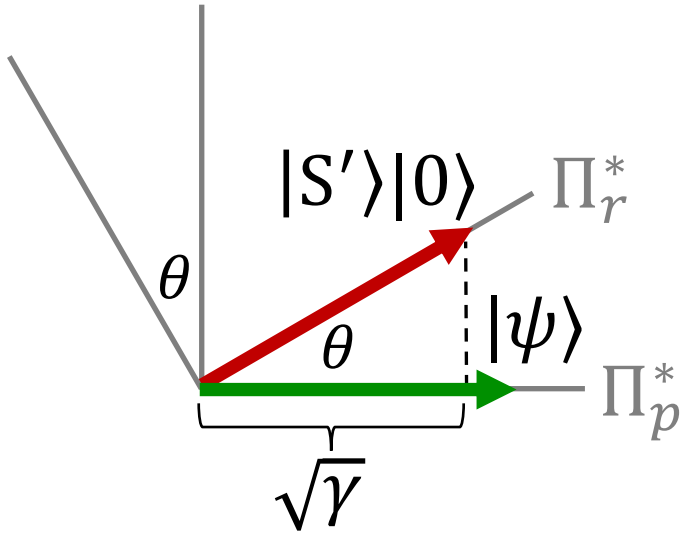
In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

Claim: ψ_A is a strongly $(p-\varepsilon)$-successful adversary state.

Proof Sketch

- If we run MW *twice* on $|S'\rangle$, the bad event (two estimates are $\geq \varepsilon$ apart) occurs with probability $\leq \delta$.
- $|\psi\rangle \propto \Pi_p^*|S'\rangle|0\rangle$ is the state after running MW on $|S'\rangle$ *once* and conditioning on output $\geq p$, which occurs with probability γ.
- After conditioning, the bad event happens with probability $\leq \delta/\gamma$.

In a nutshell: $\Pi_r^* := \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the MW procedure.



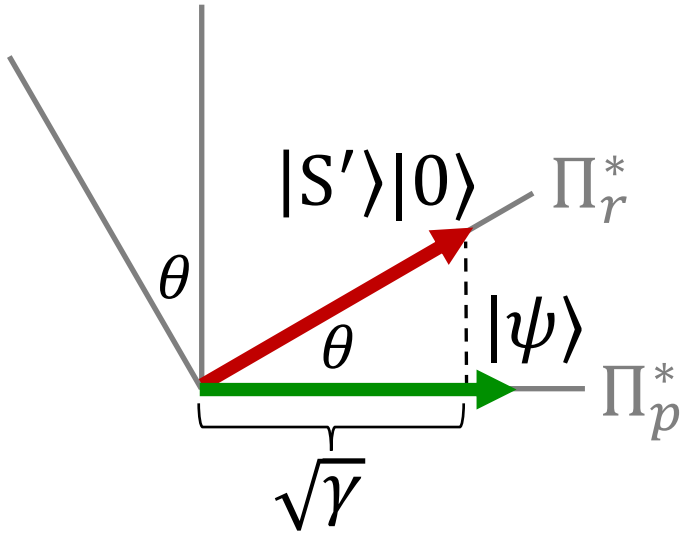
In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

Claim: ψ_A is a strongly $(p-\varepsilon)$-successful adversary state.

Proof Sketch

- If we run MW *twice* on $|S'\rangle$, the bad event (two estimates are $\geq \varepsilon$ apart) occurs with probability $\leq \delta$.
- $|\psi\rangle \propto \Pi_p^*|S'\rangle|0\rangle$ is the state after running MW on $|S'\rangle$ *once* and conditioning on output $\geq p$, which occurs with probability γ.
- After conditioning, the bad event happens with probability $\leq \delta/\gamma$. This implies $\Pr_{\mathrm{MW}(\psi_A)\to q}[q \geq p-\varepsilon] \geq 1-\delta/\gamma$.

In a nutshell: $\Pi_r^* := \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can interpret the Jordan subspaces for Π_r^*, Π_p^* in terms of the **MW** procedure.



In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

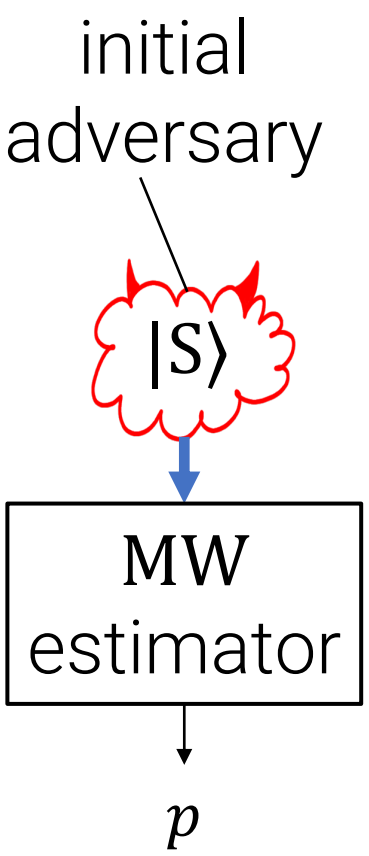
Claim: ψ_A is a strongly $(p - \varepsilon)$-successful adversary state.

Proof Sketch

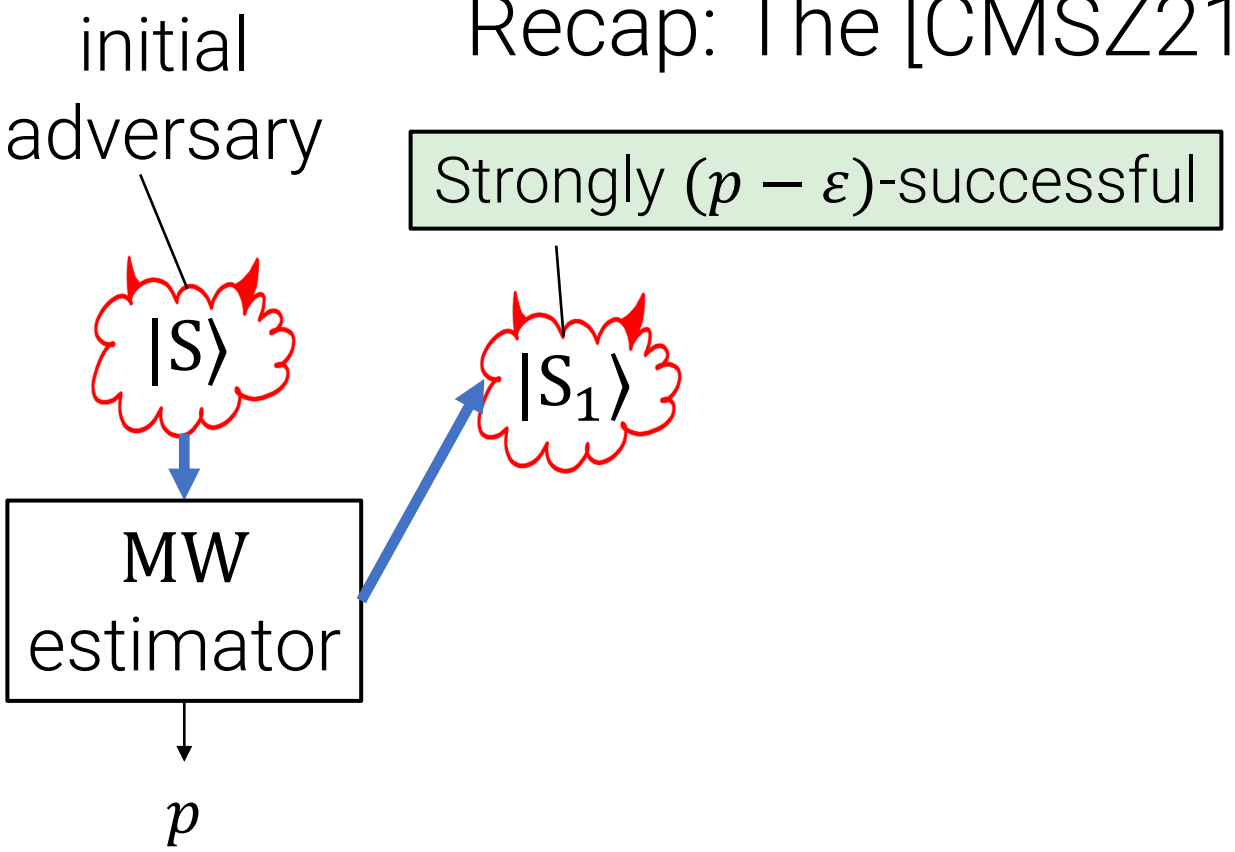
For the general case, need to show that most of the state is on subspaces where γ_j is not too small.

Recap: The [CMSZ21] Rewinding Procedure

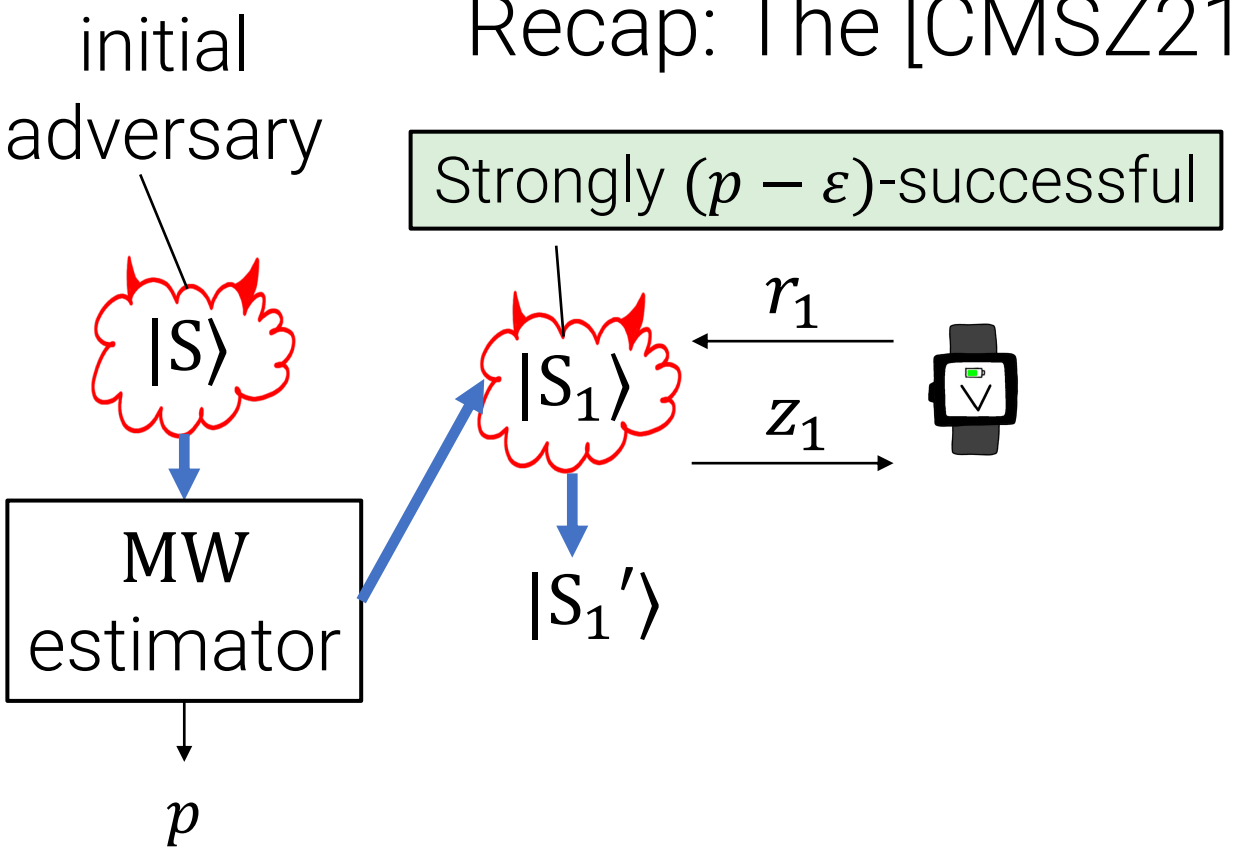
Recap: The [CMSZ21] Rewinding Procedure



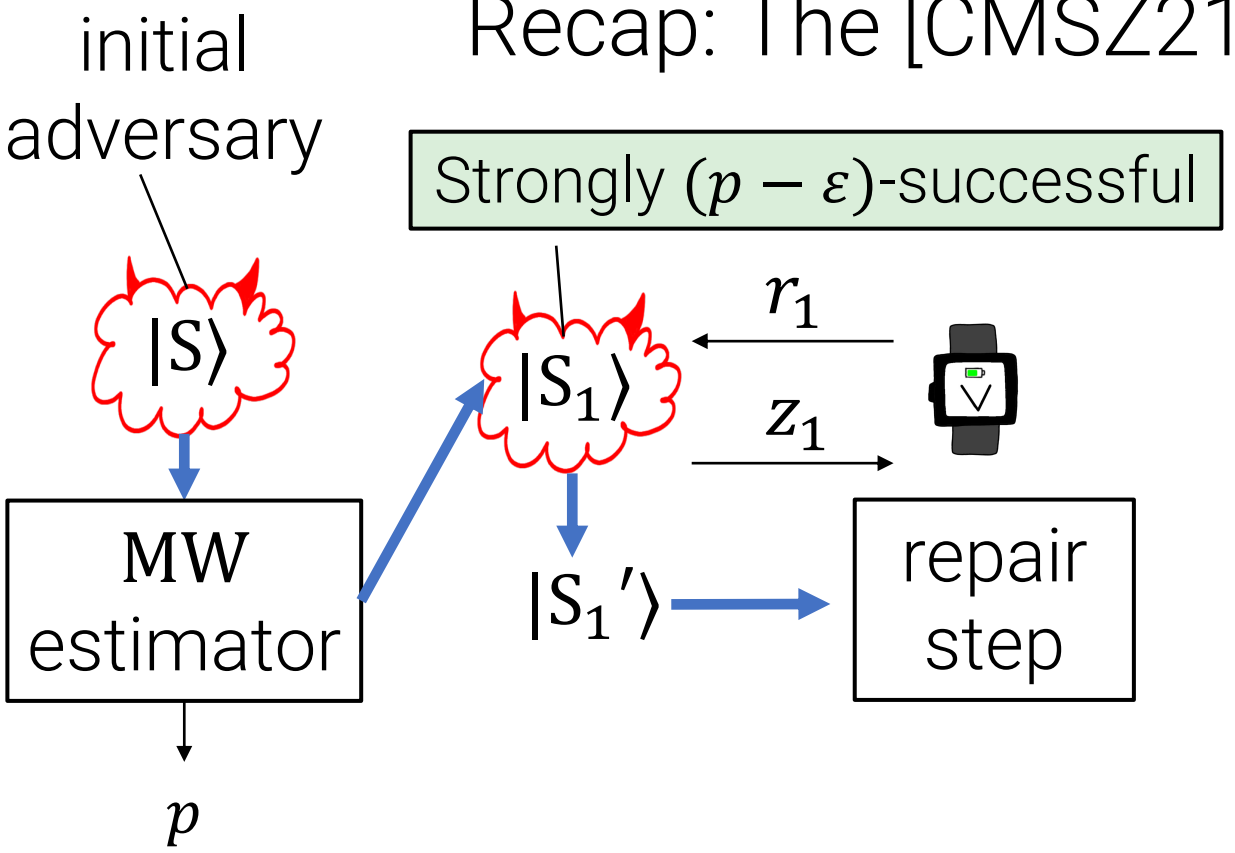
Recap: The [CMSZ21] Rewinding Procedure



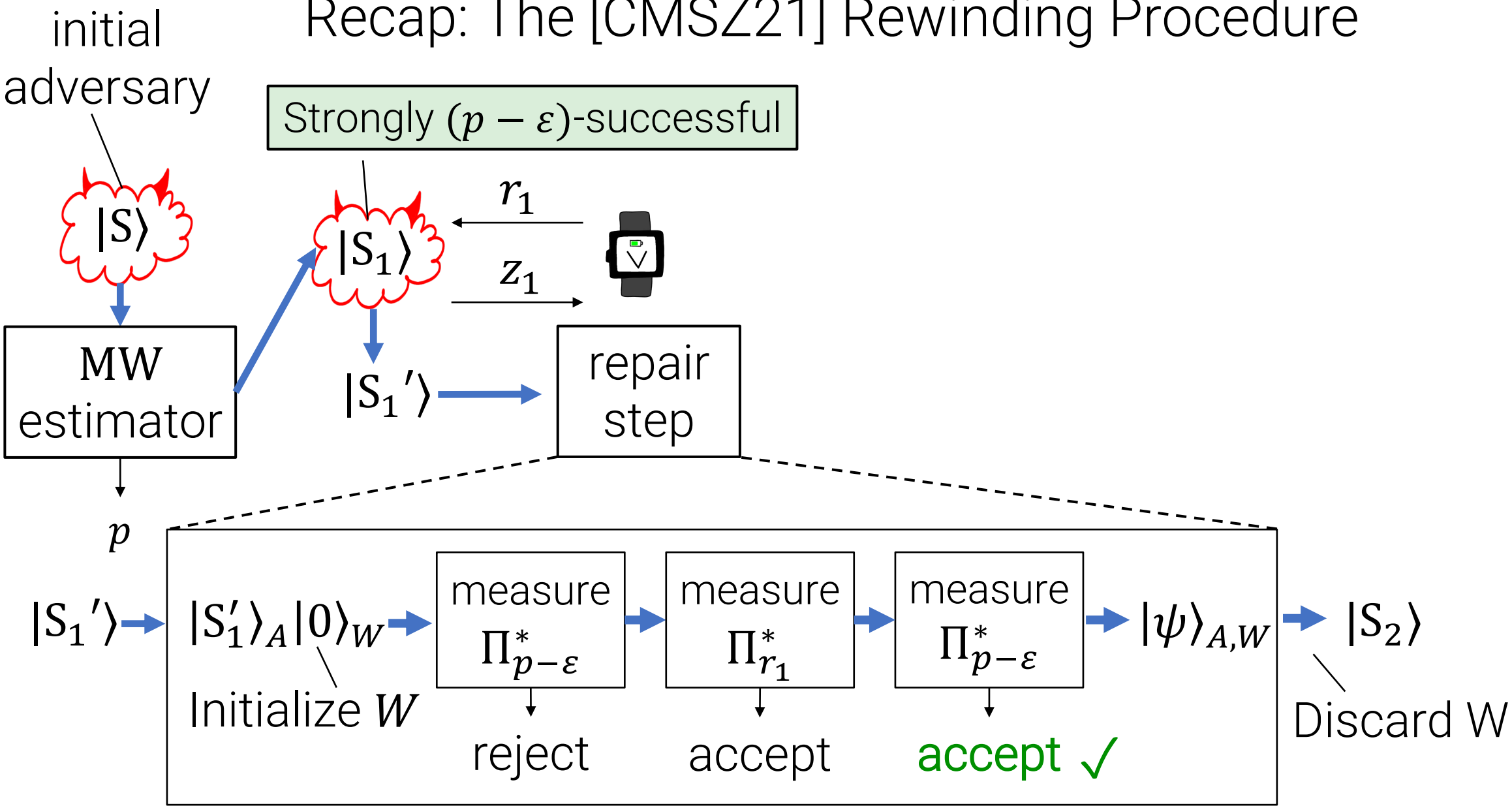
Recap: The [CMSZ21] Rewinding Procedure



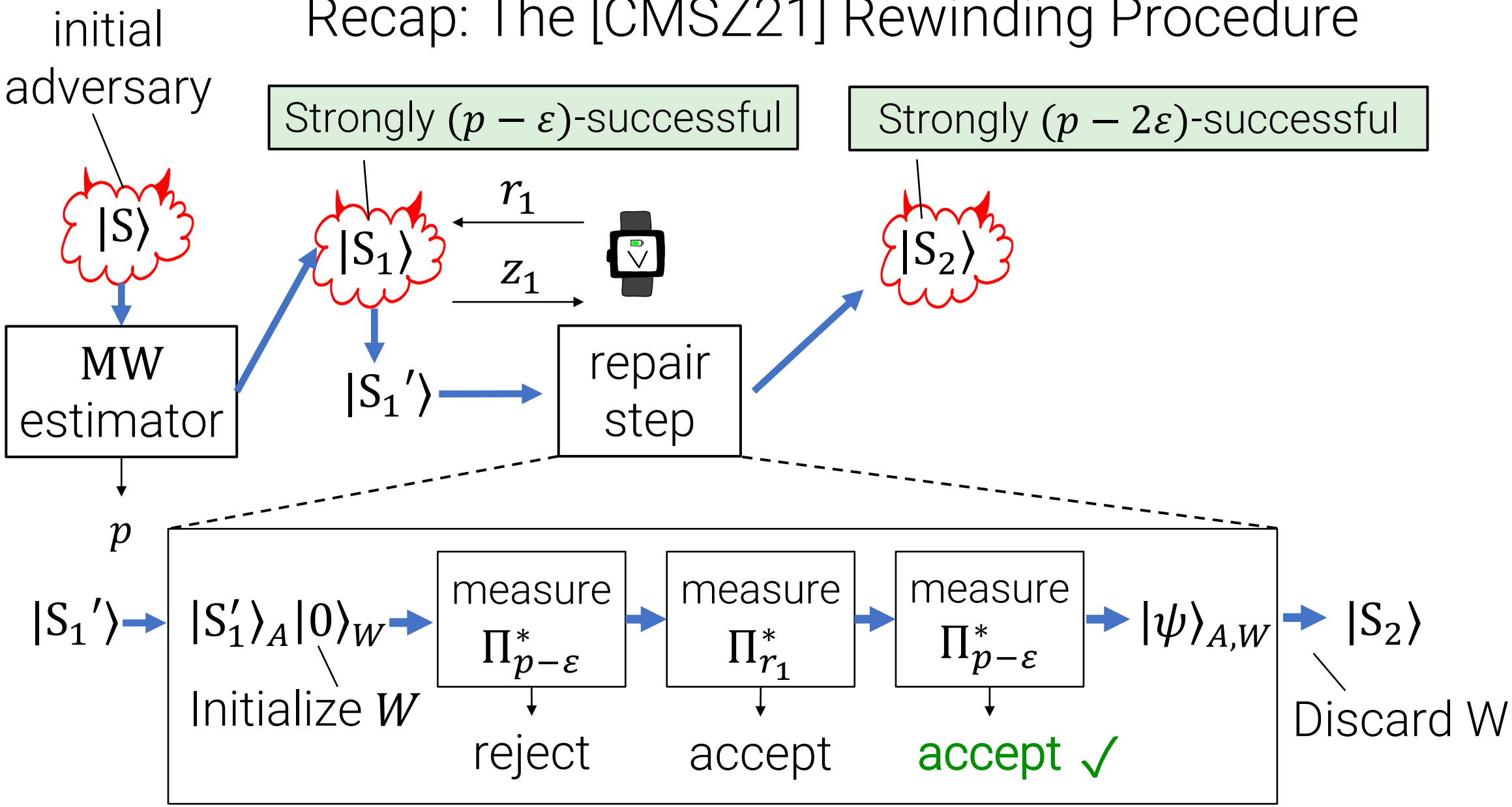
Recap: The [CMSZ21] Rewinding Procedure



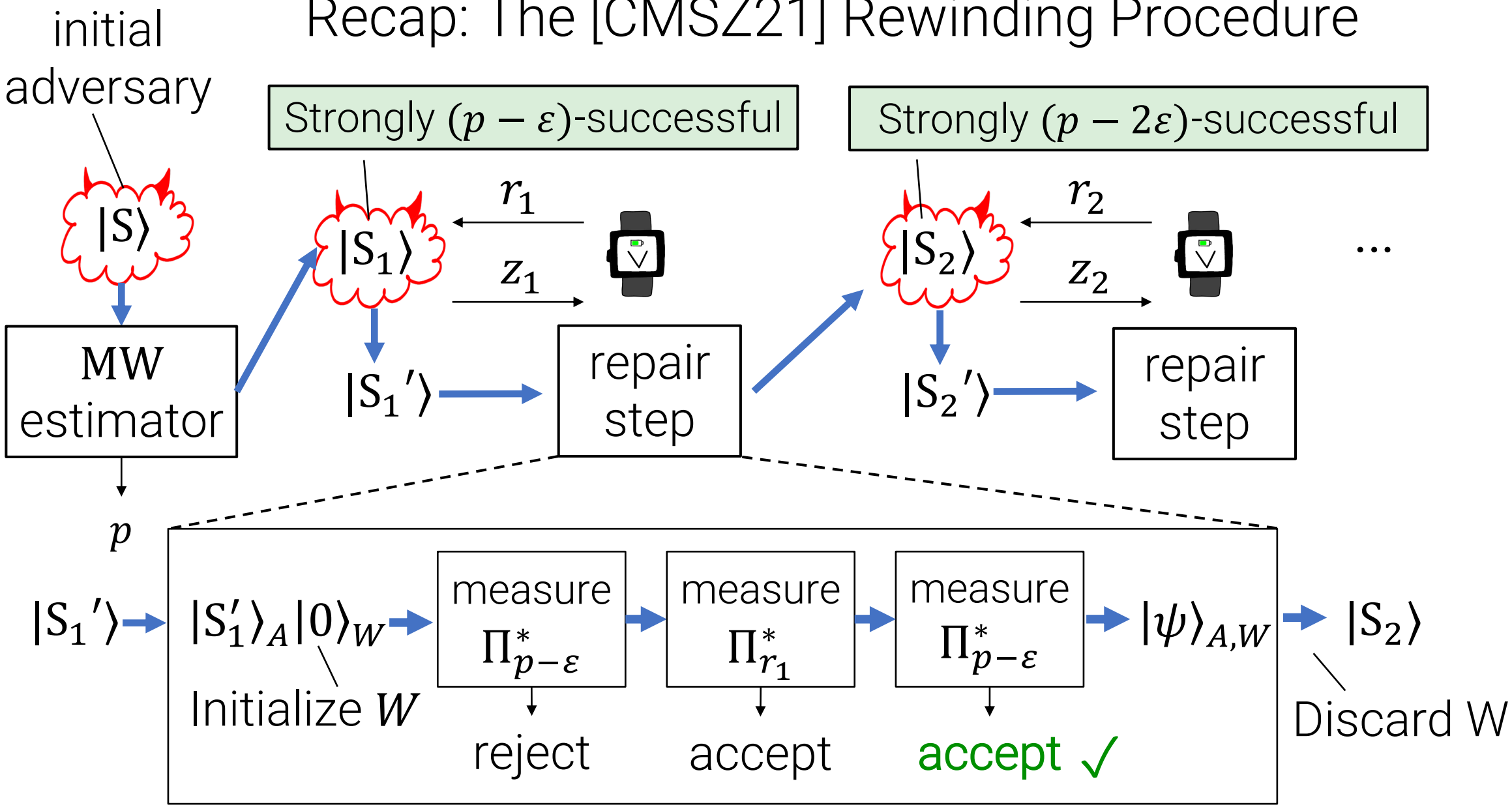
Recap: The [CMSZ21] Rewinding Procedure



Recap: The [CMSZ21] Rewinding Procedure



Recap: The [CMSZ21] Rewinding Procedure



Where does this leave us?

Where does this leave us?

- We showed how to rewind and obtain ***arbitrarily many*** accepting protocol transcripts.

Where does this leave us?

- We showed how to rewind and obtain ***arbitrarily many*** accepting protocol transcripts.
- This gives post-quantum soundness of Kilian's protocol and ***optimal*** soundness error for many other protocols (e.g., Blum).

Where does this leave us?

- We showed how to rewind and obtain ***arbitrarily many*** accepting protocol transcripts.
- This gives post-quantum soundness of Kilian's protocol and ***optimal*** soundness error for many other protocols (e.g., Blum).
- This technique has found many other applications:

Where does this leave us?

- We showed how to rewind and obtain ***arbitrarily many*** accepting protocol transcripts.
- This gives post-quantum soundness of Kilian's protocol and ***optimal*** soundness error for many other protocols (e.g., Blum).
- This technique has found many other applications:
  - [Bitansky-Brakerski-Kalai'22]: "advice preserving" non-interactive quantum reductions

Where does this leave us?

- We showed how to rewind and obtain ***arbitrarily many*** accepting protocol transcripts.
- This gives post-quantum soundness of Kilian's protocol and ***optimal*** soundness error for many other protocols (e.g., Blum).
- This technique has found many other applications:
  - [Bitansky-Brakerski-Kalai'22]: "advice preserving" non-interactive quantum reductions
  - [Lombardi-M-Spooner'22]: post-quantum zero knowledge

Where does this leave us?

- We showed how to rewind and obtain ***arbitrarily many*** accepting protocol transcripts.
- This gives post-quantum soundness of Kilian's protocol and ***optimal*** soundness error for many other protocols (e.g., Blum).
- This technique has found many other applications:
  - [Bitansky-Brakerski-Kalai'22]: "advice preserving" non-interactive quantum reductions
  - [Lombardi-M-Spooner'22]: post-quantum zero knowledge
  - [Lai-Malavolta-Spooner'22]: quantum rewinding for many-round protocols

Where does this leave us?

- We showed how to rewind and obtain ***arbitrarily many*** accepting protocol transcripts.
- This gives post-quantum soundness of Kilian's protocol and ***optimal*** soundness error for many other protocols (e.g., Blum).
- This technique has found many other applications:
  - [Bitansky-Brakerski-Kalai'22]: "advice preserving" non-interactive quantum reductions
  - [Lombardi-M-Spooner'22]: post-quantum zero knowledge
  - [Lai-Malavolta-Spooner'22]: quantum rewinding for many-round protocols
  - [Gunn-Ju-M-Zhandry'22]: quantum succinct arguments (from assumptions weaker than one-way functions)

Thank You!

Questions?