Quantum Rewinding Tutorial Part 2:

How to Run a Quantum Attacker Many Times

(or: The Unreasonable Effectiveness of Alternating Projectors)

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 (2021)

In the first talk, we saw how to use Unruh’s rewinding technique to prove post-quantum soundness of Blum’s protocol.

In the first talk, we saw how to use Unruh’s rewinding technique to prove post-quantum soundness of Blum’s protocol.

However, this technique has some major drawbacks:

In the first talk, we saw how to use Unruh's rewinding technique to prove post-quantum soundness of Blum's protocol.

However, this technique has some major drawbacks:

- We get bad soundness guarantees (can't rule out a quantum prover that breaks Blum with probability 0.7)

In the first talk, we saw how to use Unruh's rewinding technique to prove post-quantum soundness of Blum's protocol.

However, this technique has some major drawbacks:

- We get bad soundness guarantees (can't rule out a quantum prover that breaks Blum with probability 0.7)
- More serious issue: this technique only applies to a *very limited class* of protocols (e.g., Blum but not [GMW86] 3-coloring)

In the first talk, we saw how to use Unruh's rewinding technique to prove post-quantum soundness of Blum's protocol.

However, this technique has some major drawbacks:

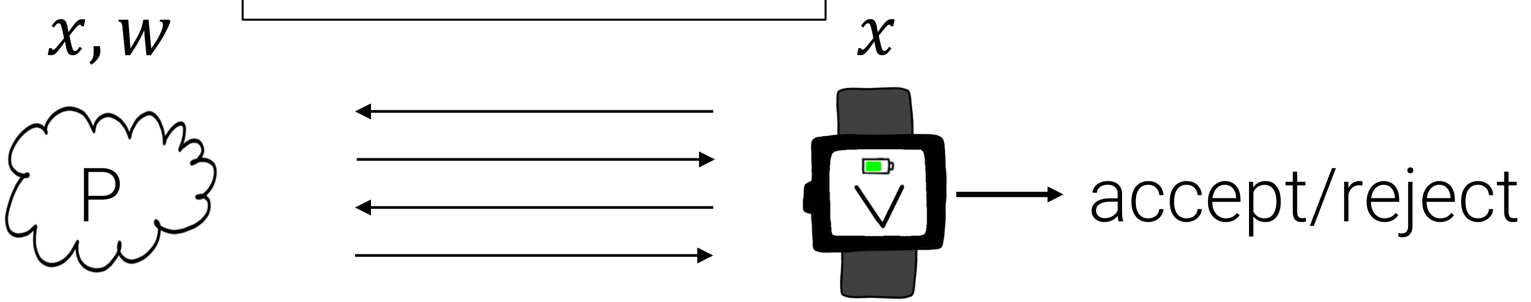
- We get bad soundness guarantees (can't rule out a quantum prover that breaks Blum with probability 0.7)
- More serious issue: this technique only applies to a *very limited class* of protocols (e.g., Blum but not [GMW86] 3-coloring)

Plan for this talk: we'll see a significantly more powerful rewinding technique due to [CMSZ21].

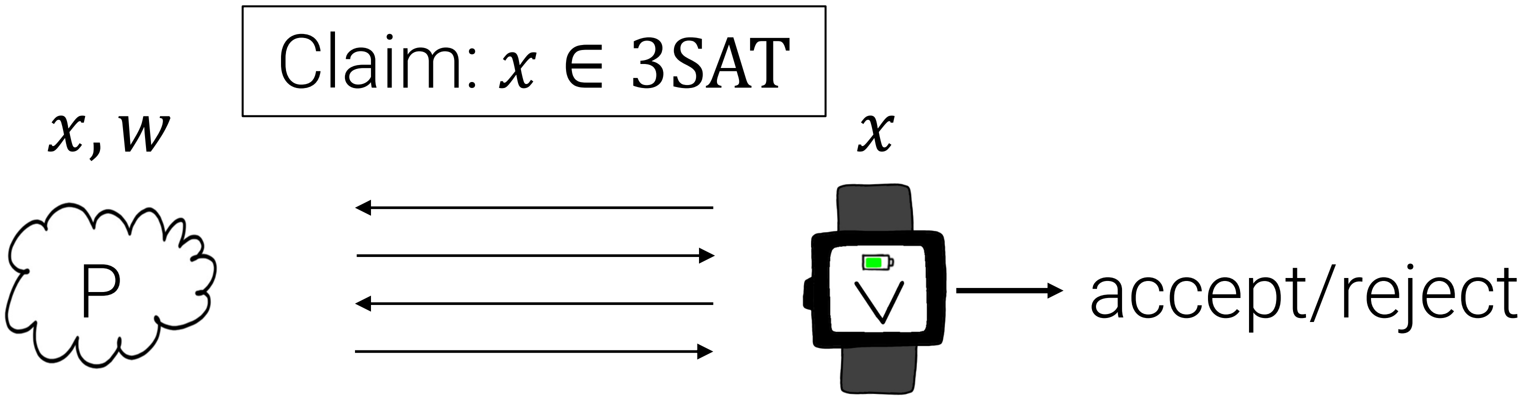
Motivating example:
Succinct Arguments for NP

Motivating example:
Succinct Arguments for NP

$\boxed{\text{Claim: } x \in \mathrm{3SAT}}$



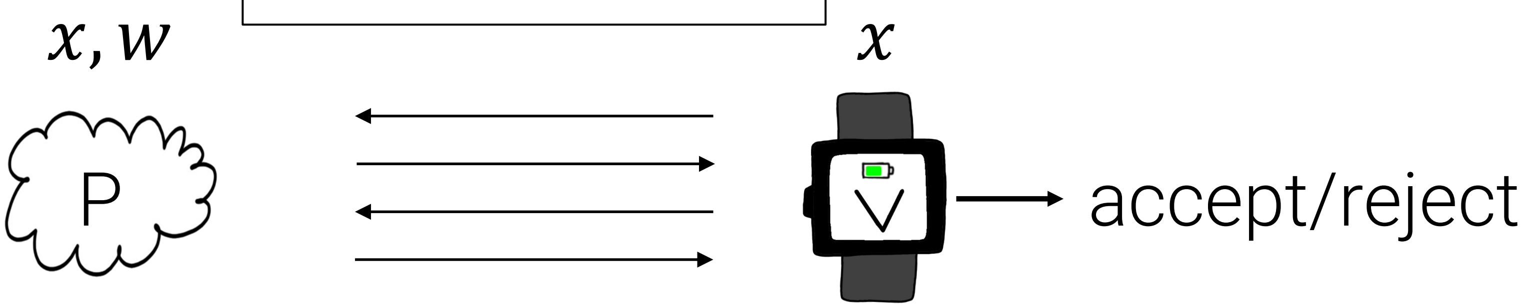
Motivating example:
Succinct Arguments for NP



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

Motivating example:
Succinct Arguments for NP

Claim: $x \in \mathrm{3SAT}$

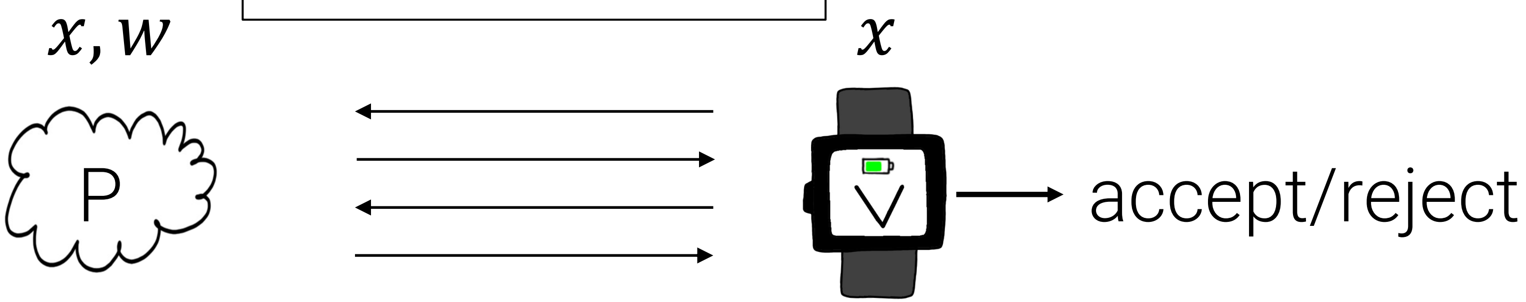


"Succinct" = communication + verifier efficiency is $\mathrm{poly}(\lambda, \log(|x| + |w|))$

"Argument" = sound against *efficient* cheating 

Motivating example:
Succinct Arguments for NP

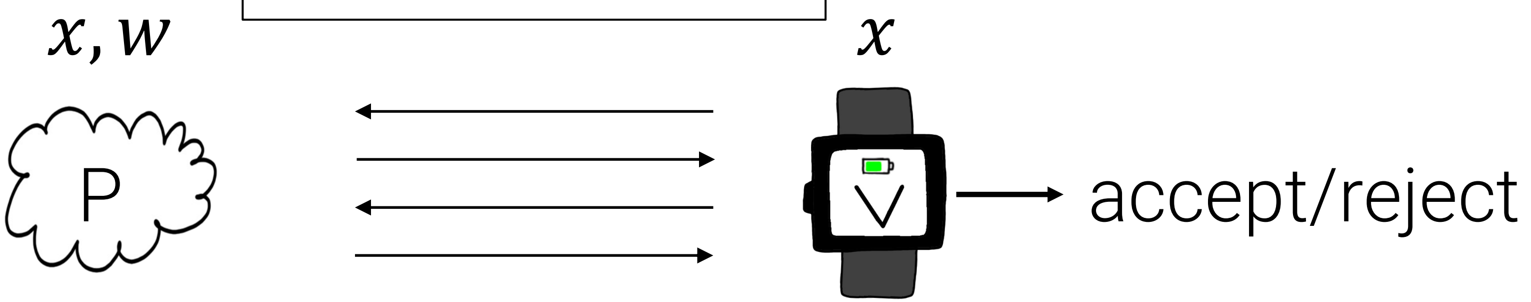
Claim: $x \in \mathrm{3SAT}$



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

Motivating example:
Succinct Arguments for NP

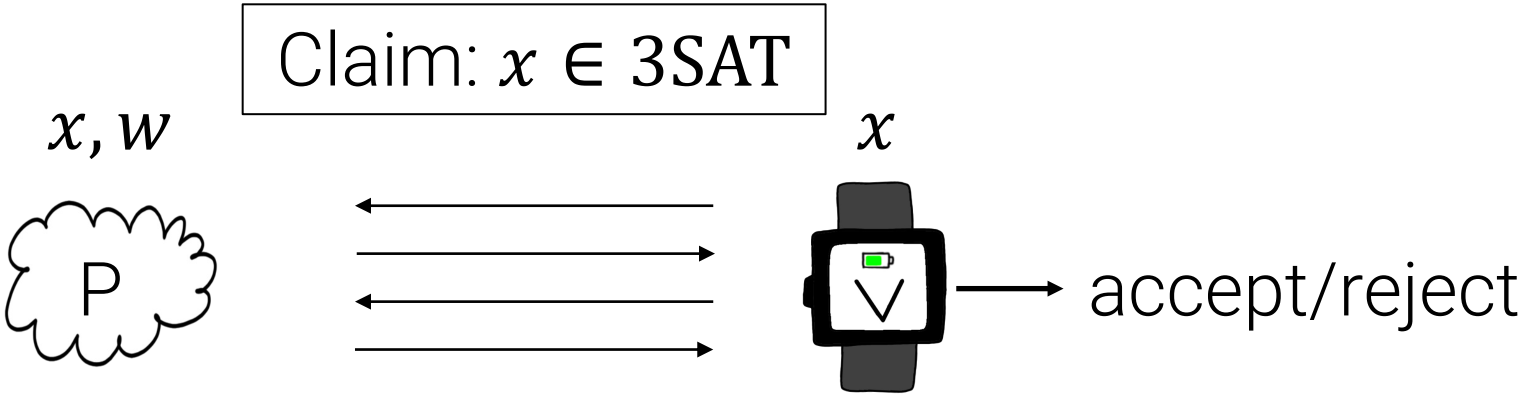
Claim: $x \in 3\mathrm{SAT}$



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

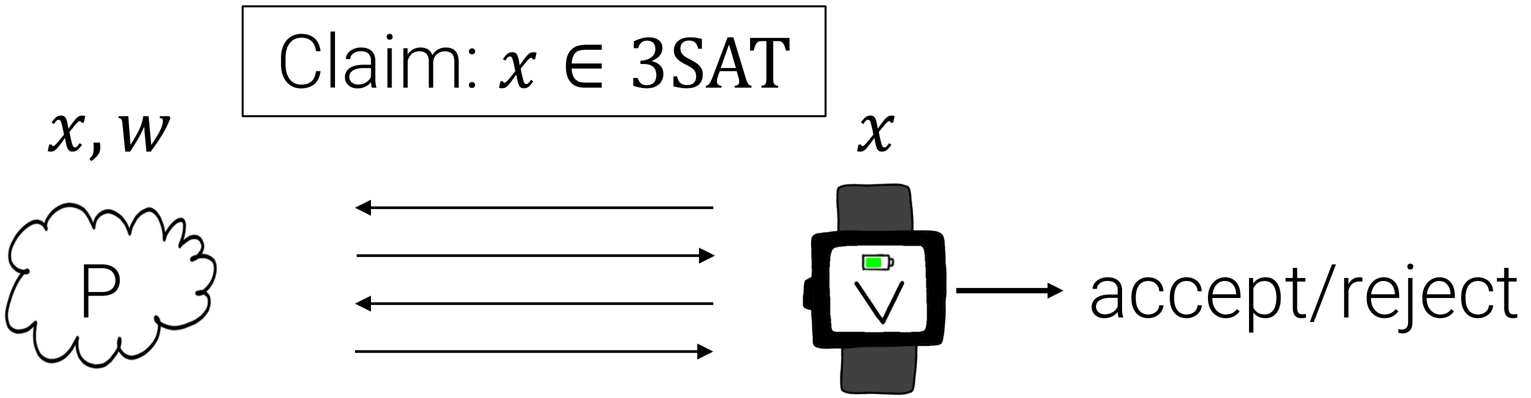
Many applications: universal arguments [BG01], zero knowledge [Barak01], SNARGs [Micali94, BCS16], …

Motivating example:
Succinct Arguments for NP



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

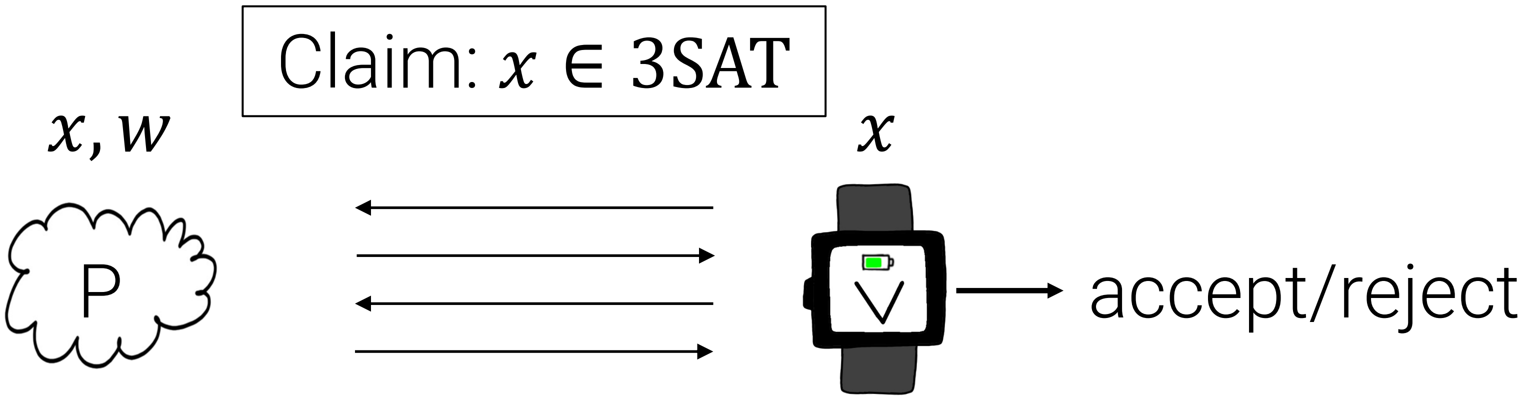
Motivating example:
Succinct Arguments for NP



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.

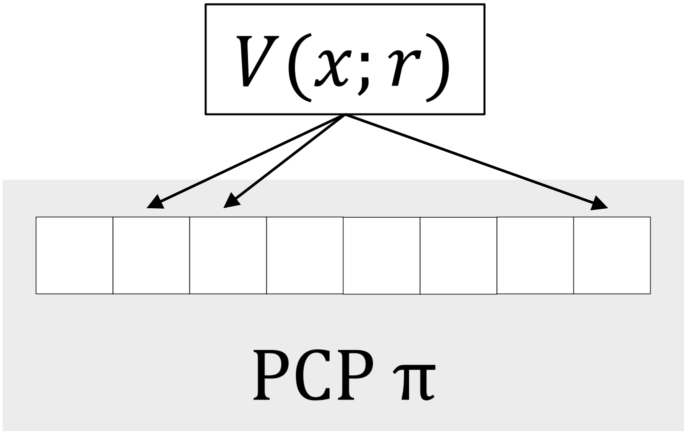
- Typically prove soundness using several transcripts to specify a witness.
- Succinct arguments inherently require many transcripts to specify a witness, so *lots* of rewinding is required.

Let’s see how Kilian’s protocol works

Kilian's protocol

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

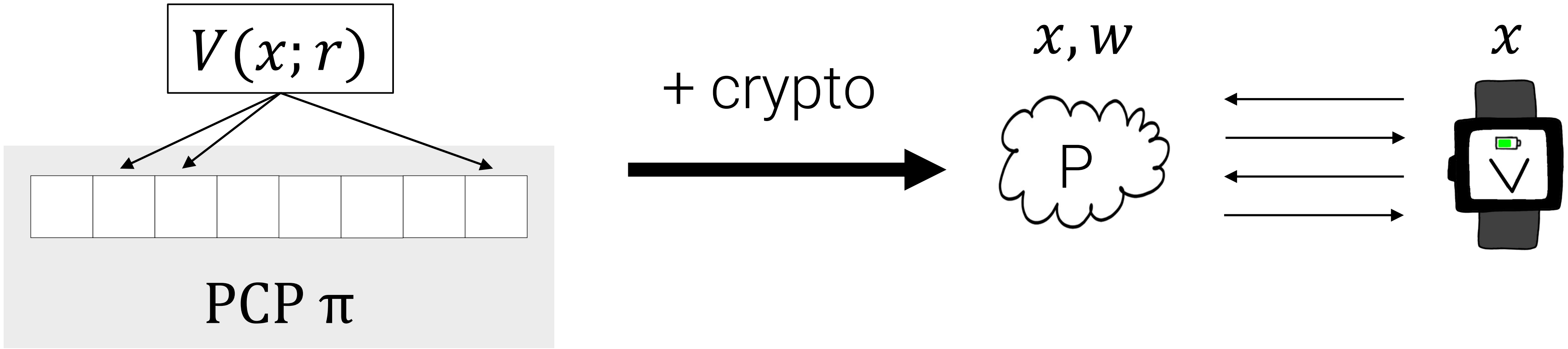
*[BFLS91,FGLSS91,AS92,ALMSS92]



Kilian's protocol

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

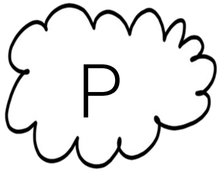
*[BFLS91,FGLSS91,AS92,ALMSS92]



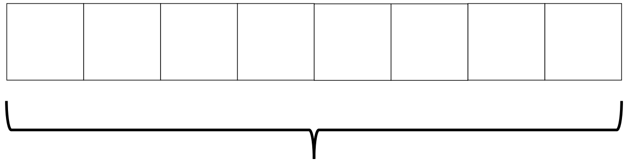
Kilian's protocol

x, w

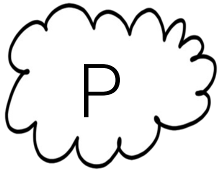
x



Encode w as PCP π



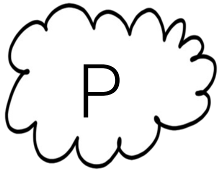
PCP π



sends short commitment to PCP π.

Kilian’s protocol

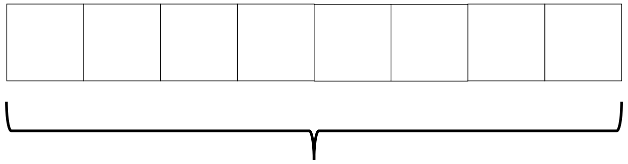
x, w



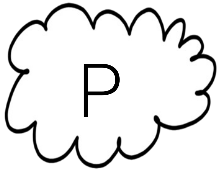
CRHF h

x

Encode w as PCP π



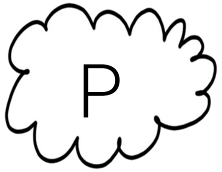
PCP π



sends short commitment to PCP π.

Kilian's protocol

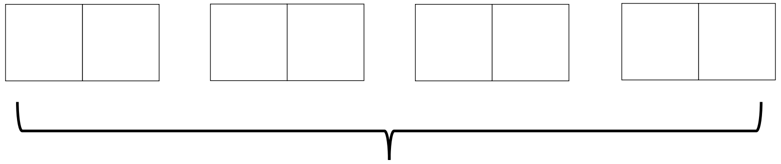
x, w



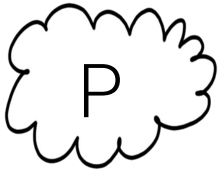
x

CRHF h

Encode w as PCP π



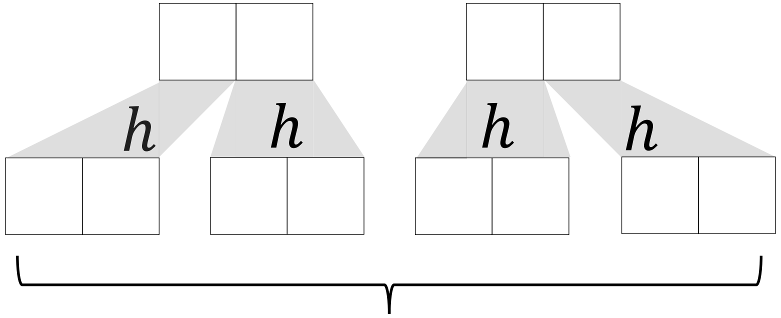
PCP π



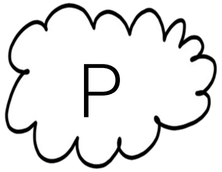
sends short commitment to PCP π.

Kilian’s protocol

x, w x

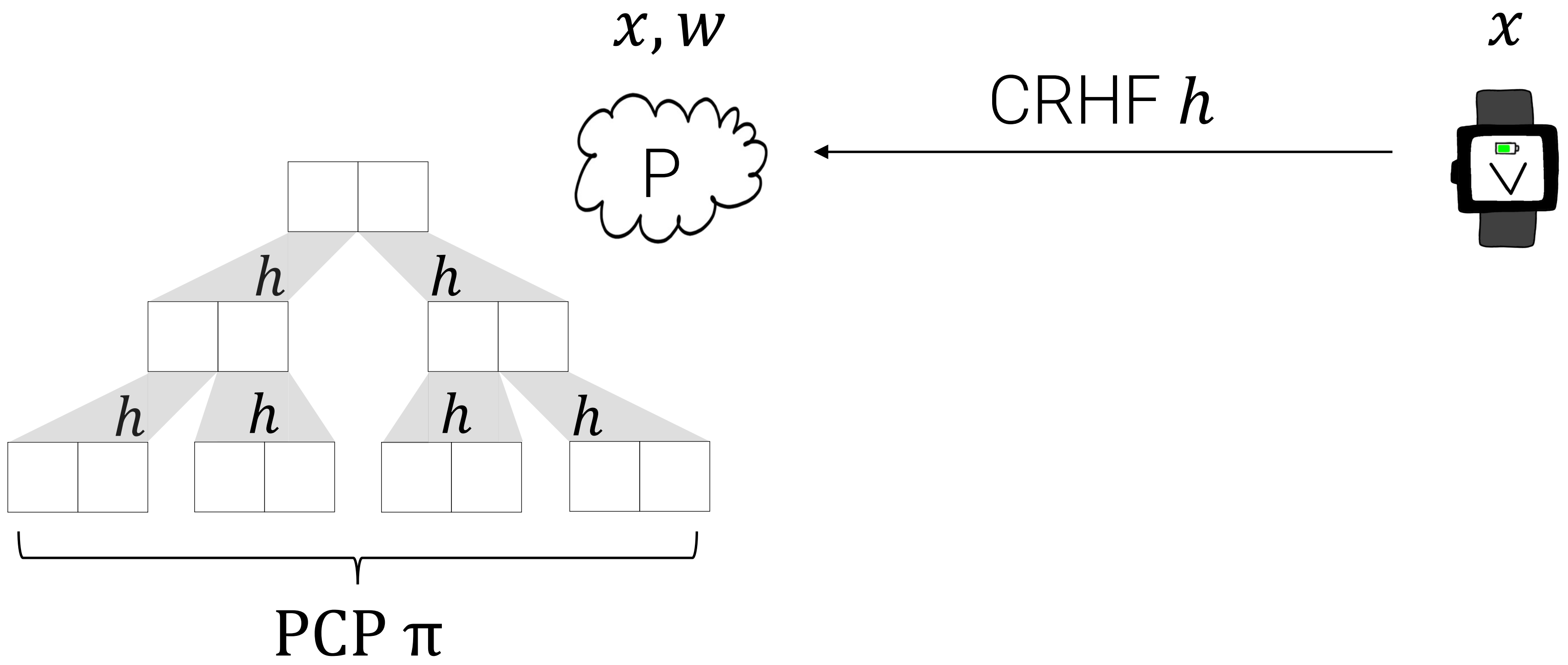


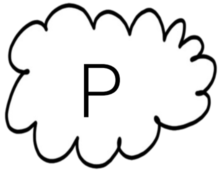
PCP π



sends short commitment to PCP π.

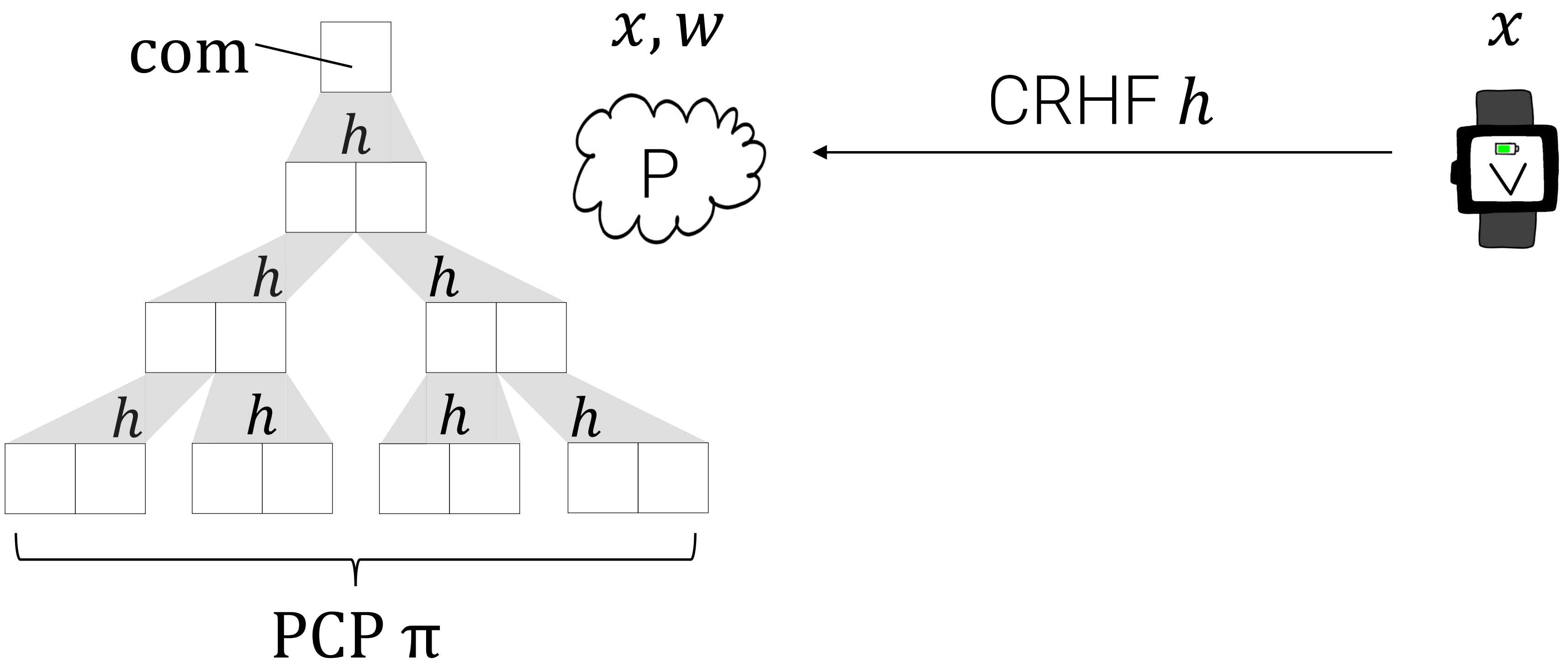
Kilian's protocol

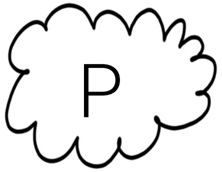




sends short commitment to PCP π.

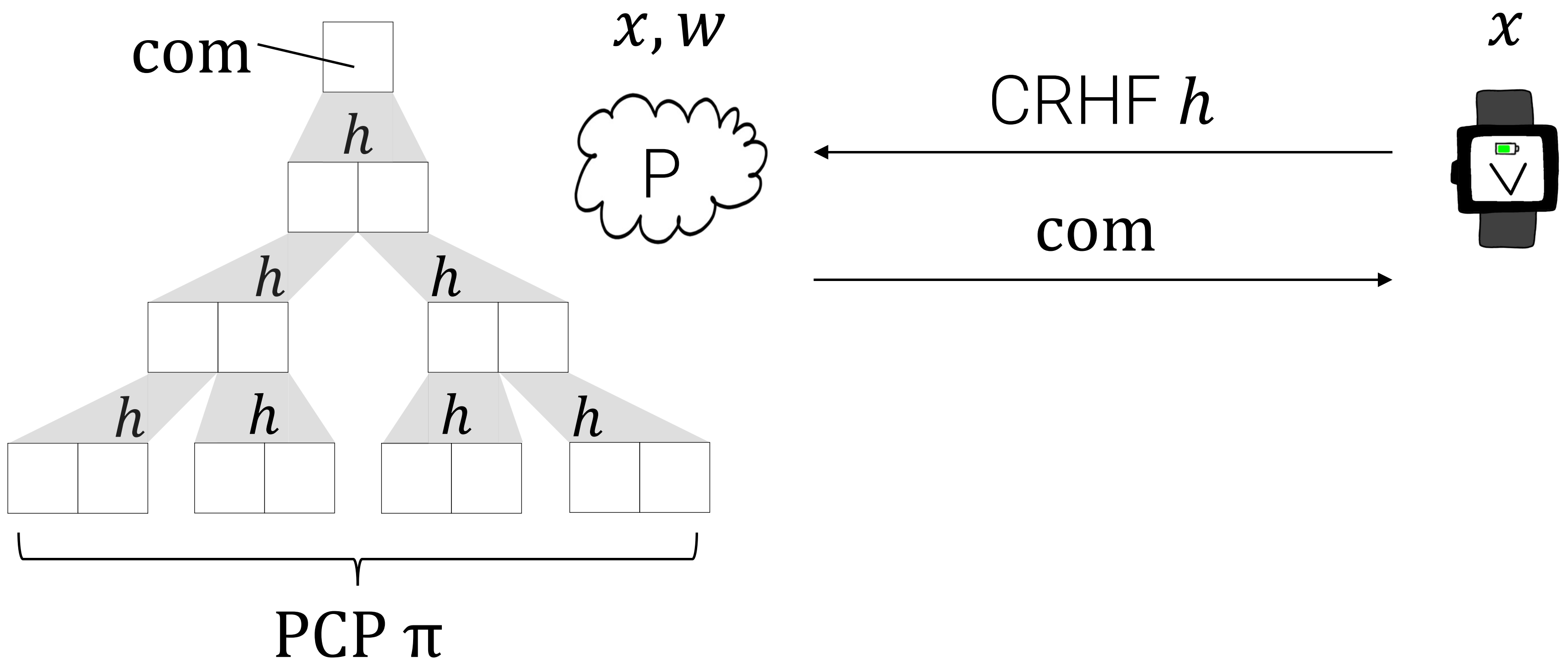
Kilian's protocol





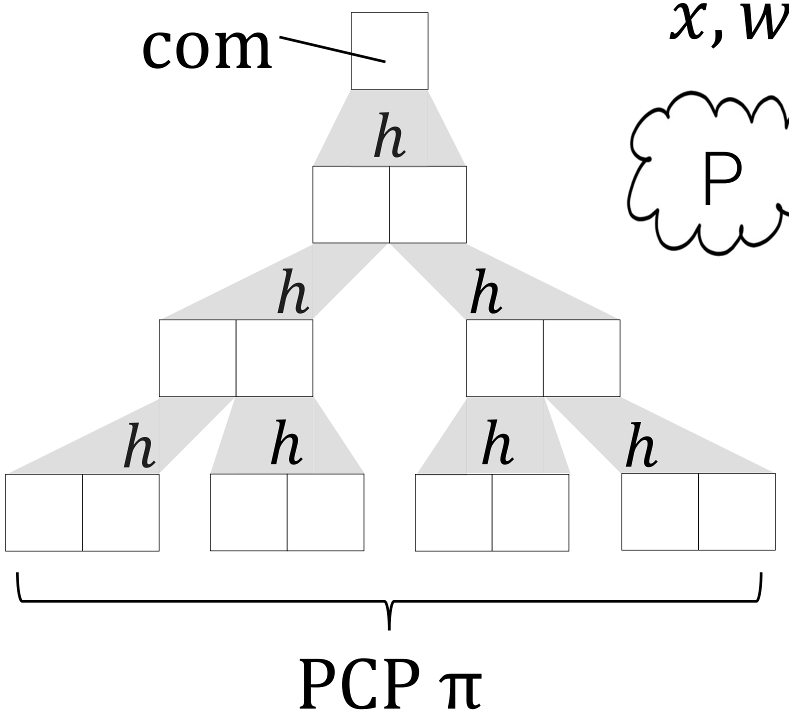
sends short commitment to PCP π.

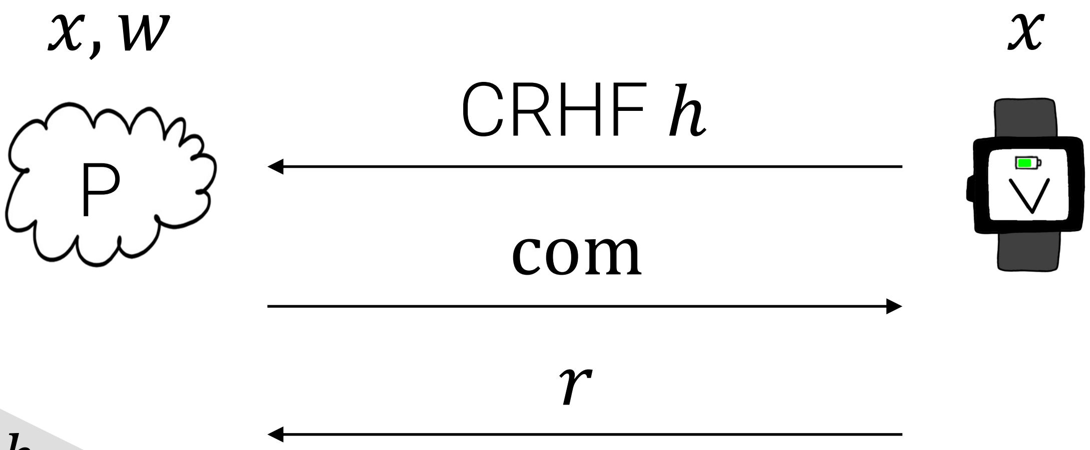
Kilian's protocol



P sends short commitment to PCP π.

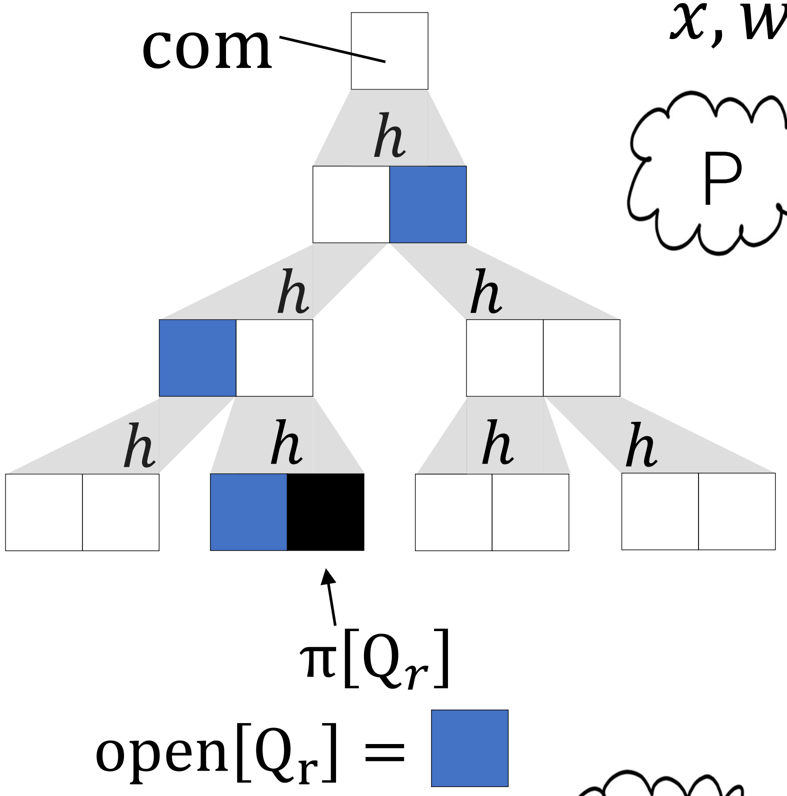
Kilian's protocol

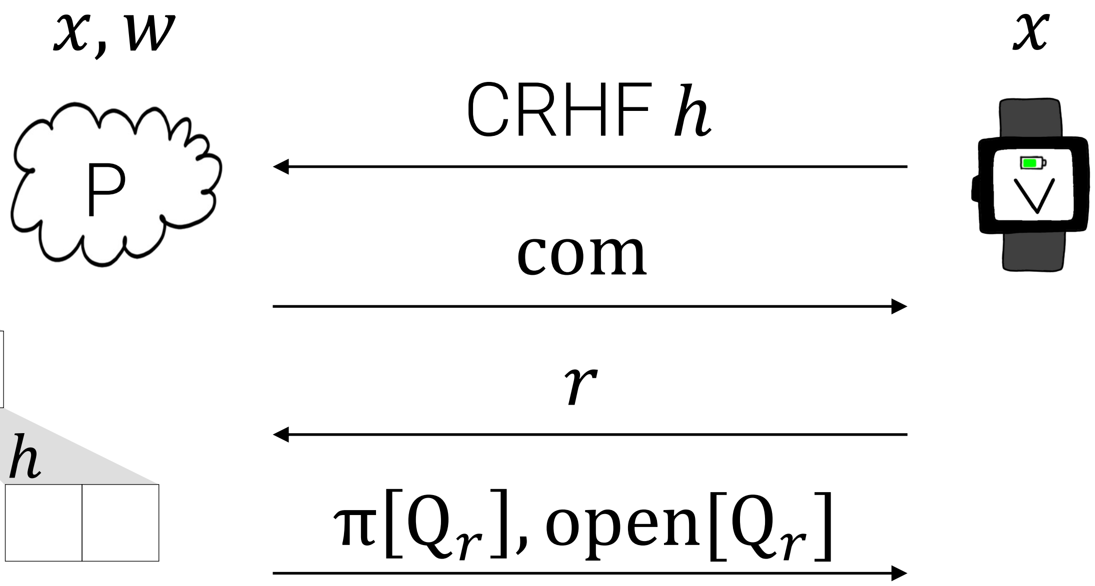


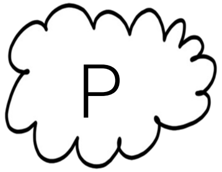


samples PCP verifier coins $r \leftarrow R$.

Kilian's protocol



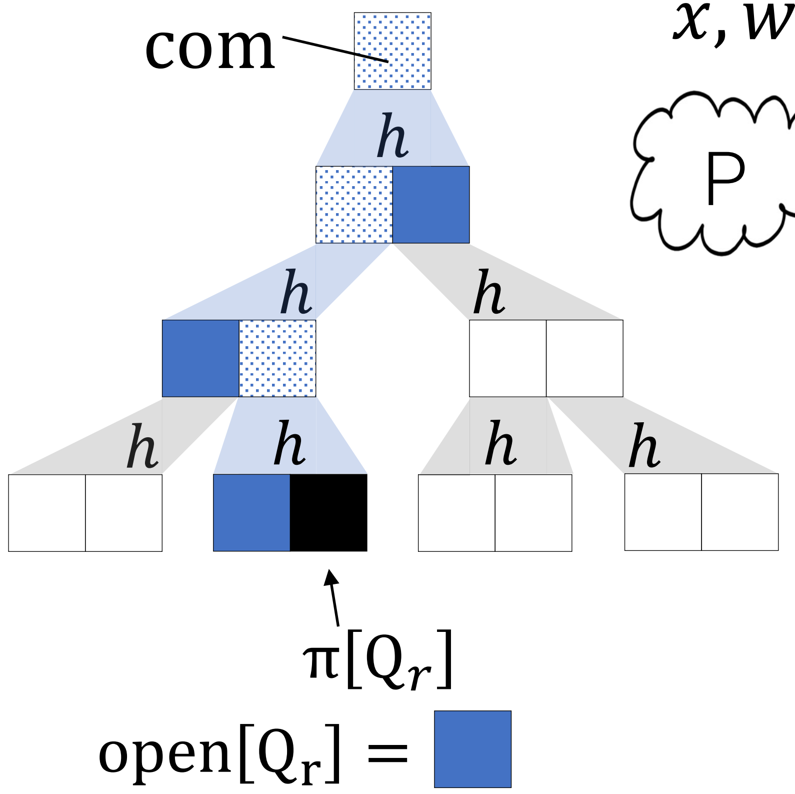


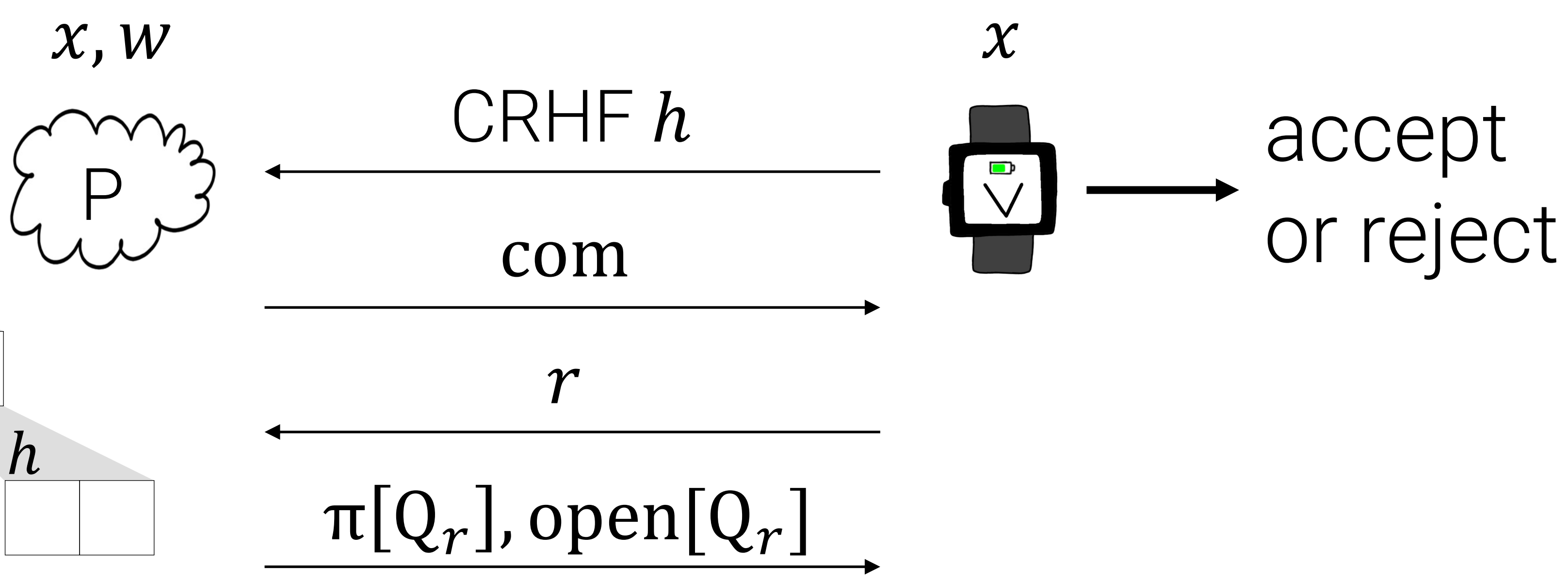


sends $\pi[\mathrm{Q}_r]$ + opening proofs

$\mathbf{Q}_r$ = indices PCP verifier checks on random coins r

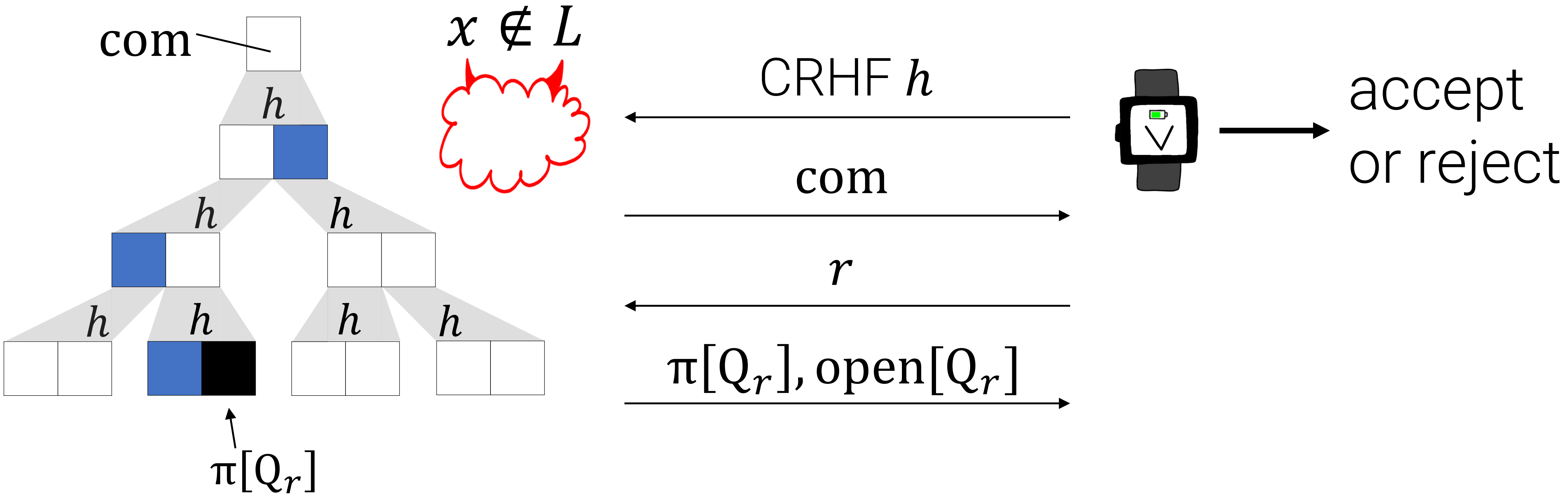
Kilian’s protocol





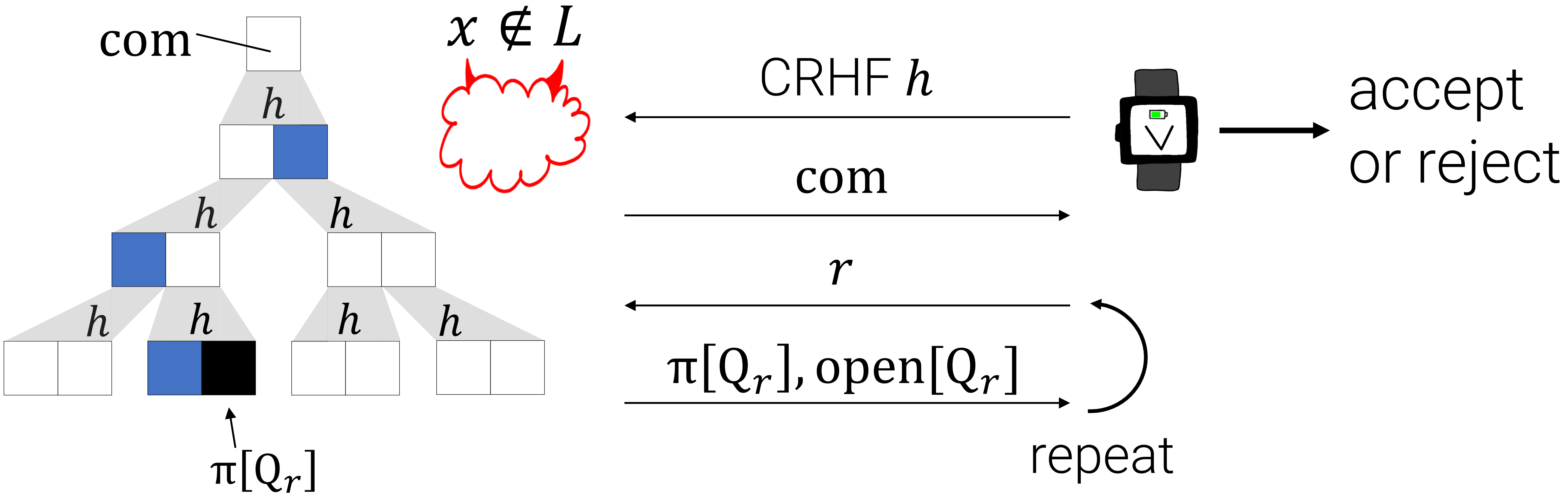
accepts if openings valid
+ PCP verifier accepts

Classical Security



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

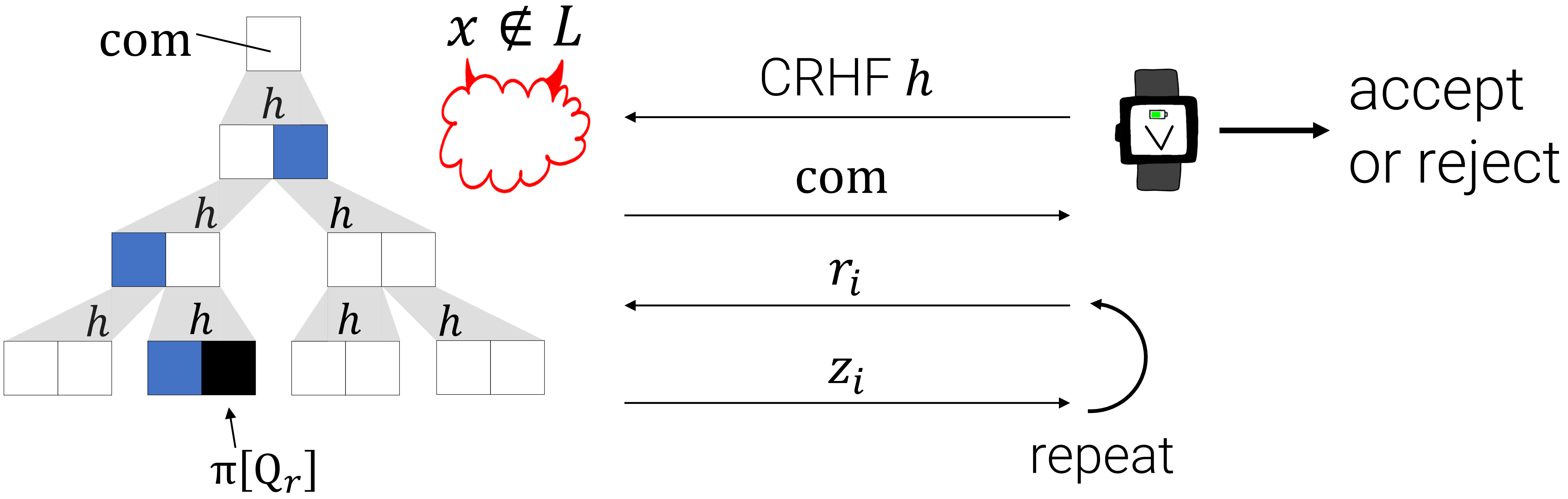
Classical Security



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

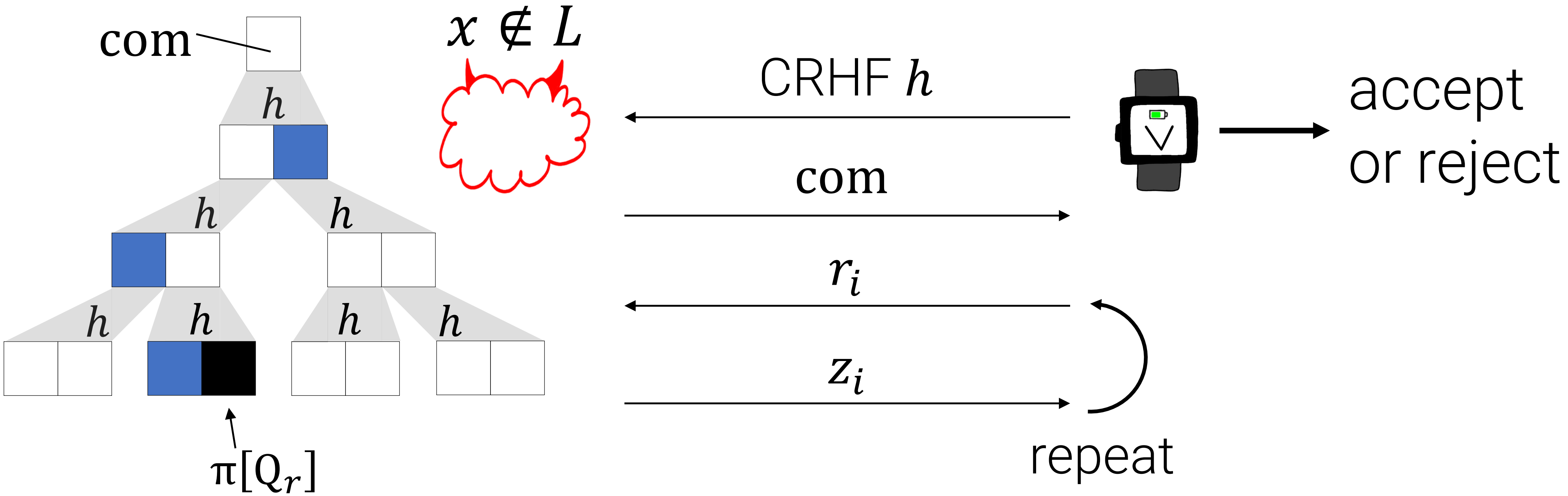
Formalize by *rewinding* last two messages many times.

Classical Security



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

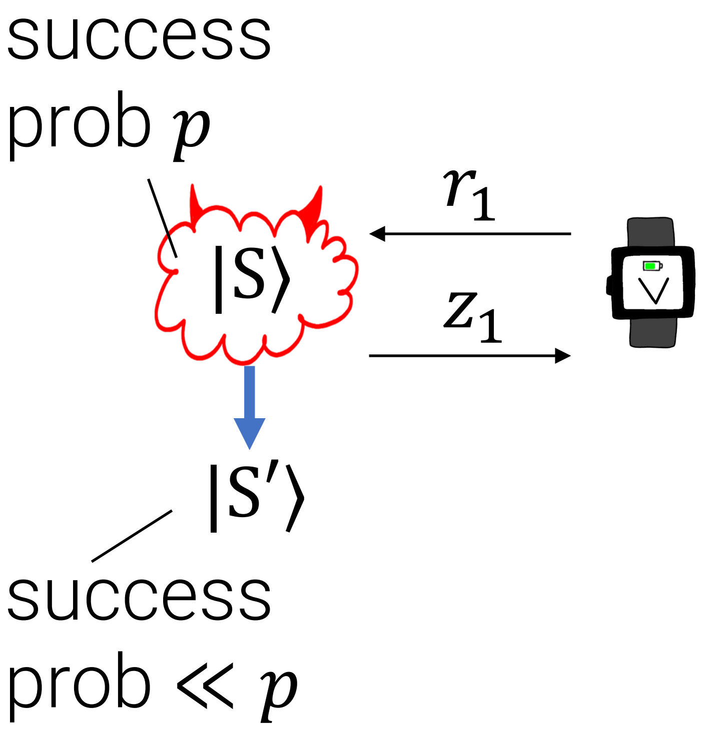
Classical Security



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

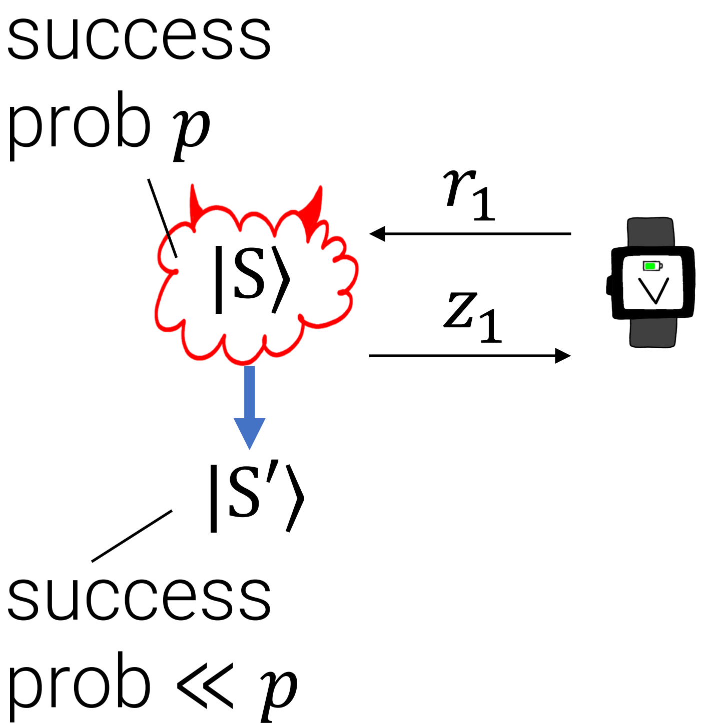
Eventually finds $\underbrace{\text{impossible } \pi}$ OR collision.

$\boxed{\text{Pr[PCP verifier accepts } \pi \text{] > PCP soundness error}}$



Define *success probability* as

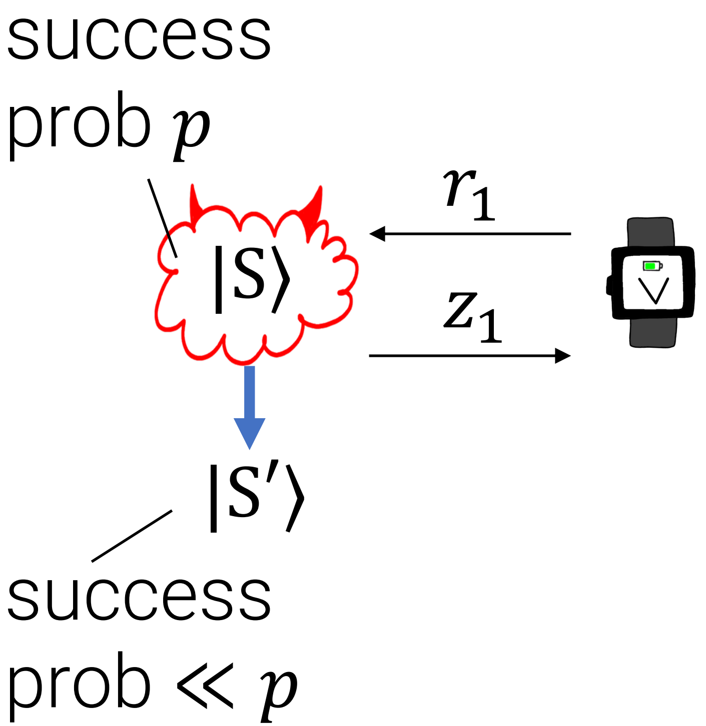
$p \coloneqq \Pr\limits_{r \leftarrow R}[$  wins$]$



Define *success probability* as

$$p := \Pr_{r \leftarrow R}[\,|S\rangle \text{ wins}]$$

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

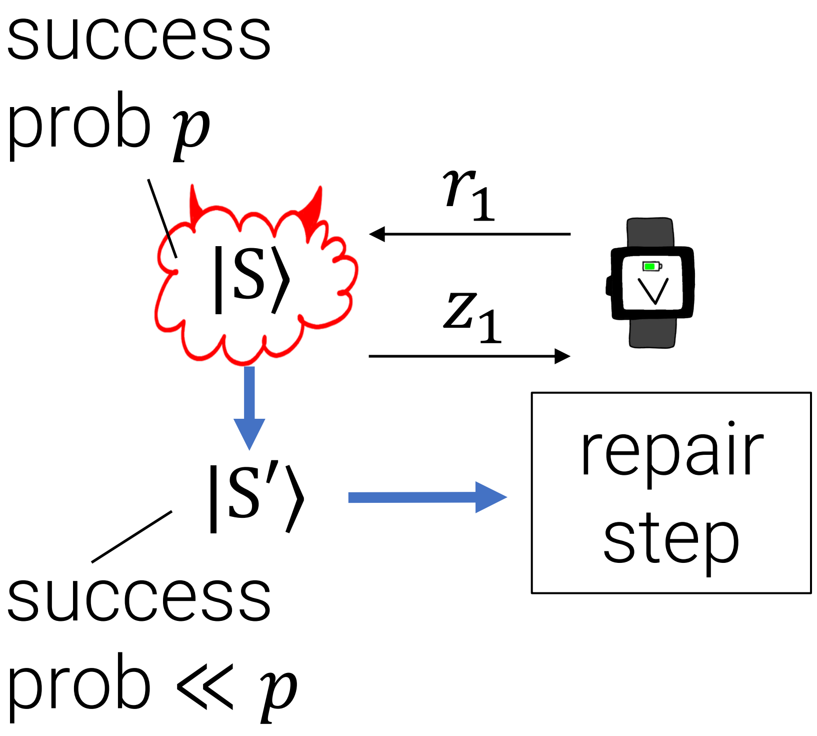


Define *success probability* as

$$p := \Pr_{r \leftarrow R}[\,|S\rangle \text{ wins}]$$

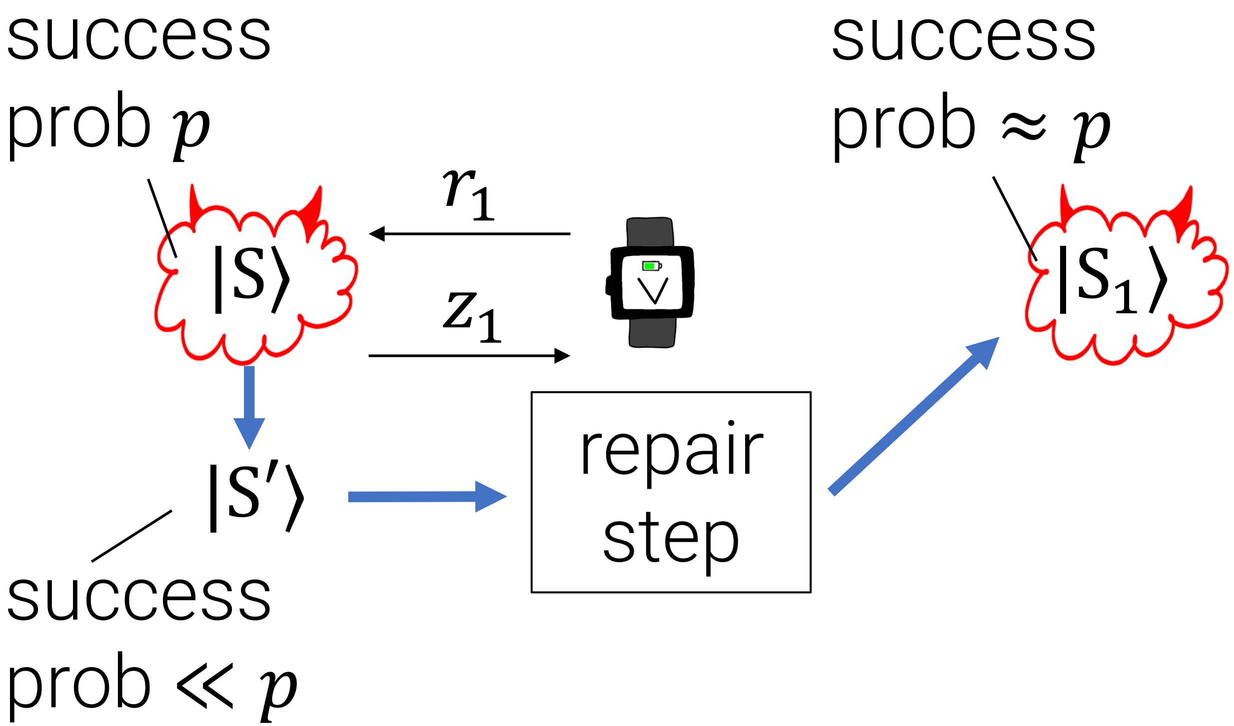
Problem: $|S'\rangle$ might not be a successful adversary!

This work: we devise a “repair” procedure to restore the original success probability.



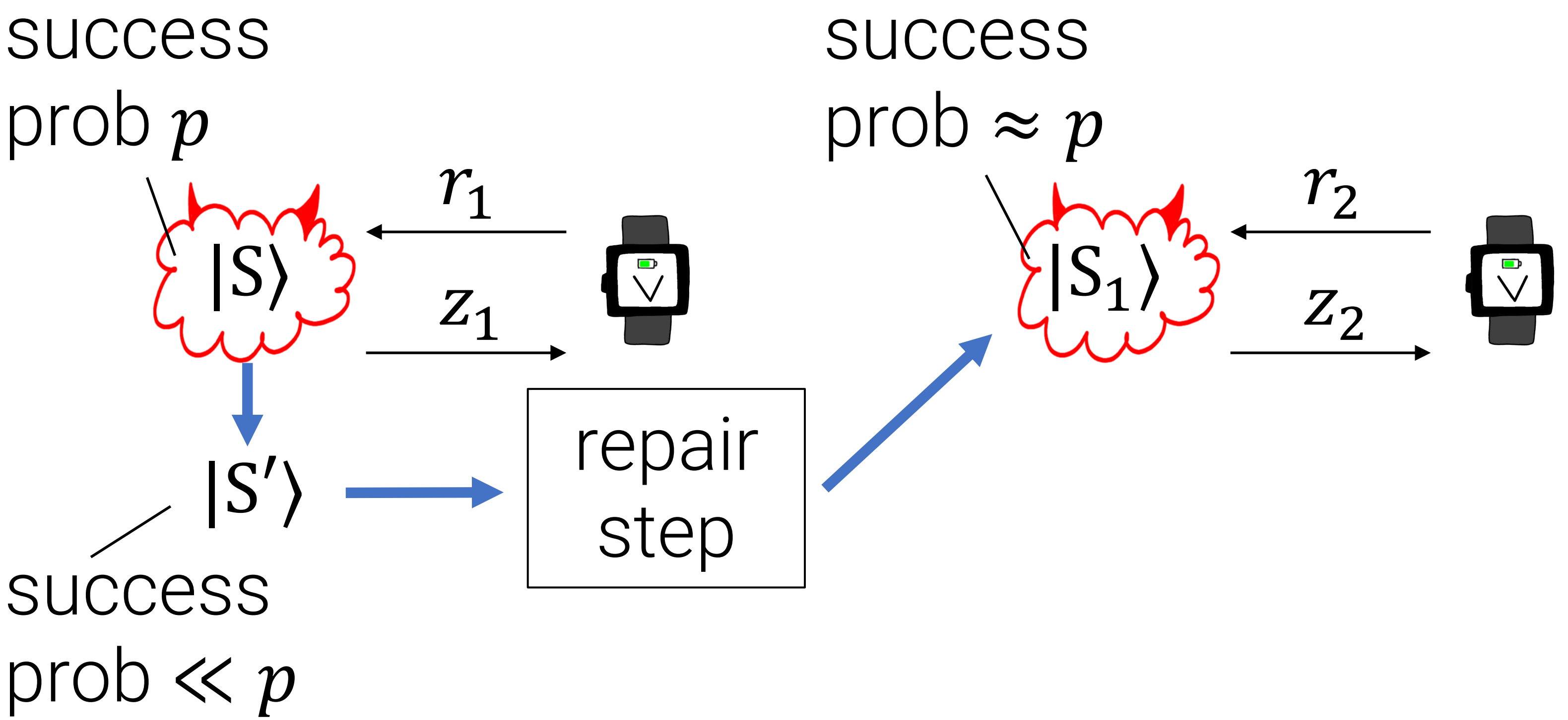
Problem: $|S'\rangle$ might not be a successful adversary!

This work: we devise a "repair" procedure to restore the original success probability.



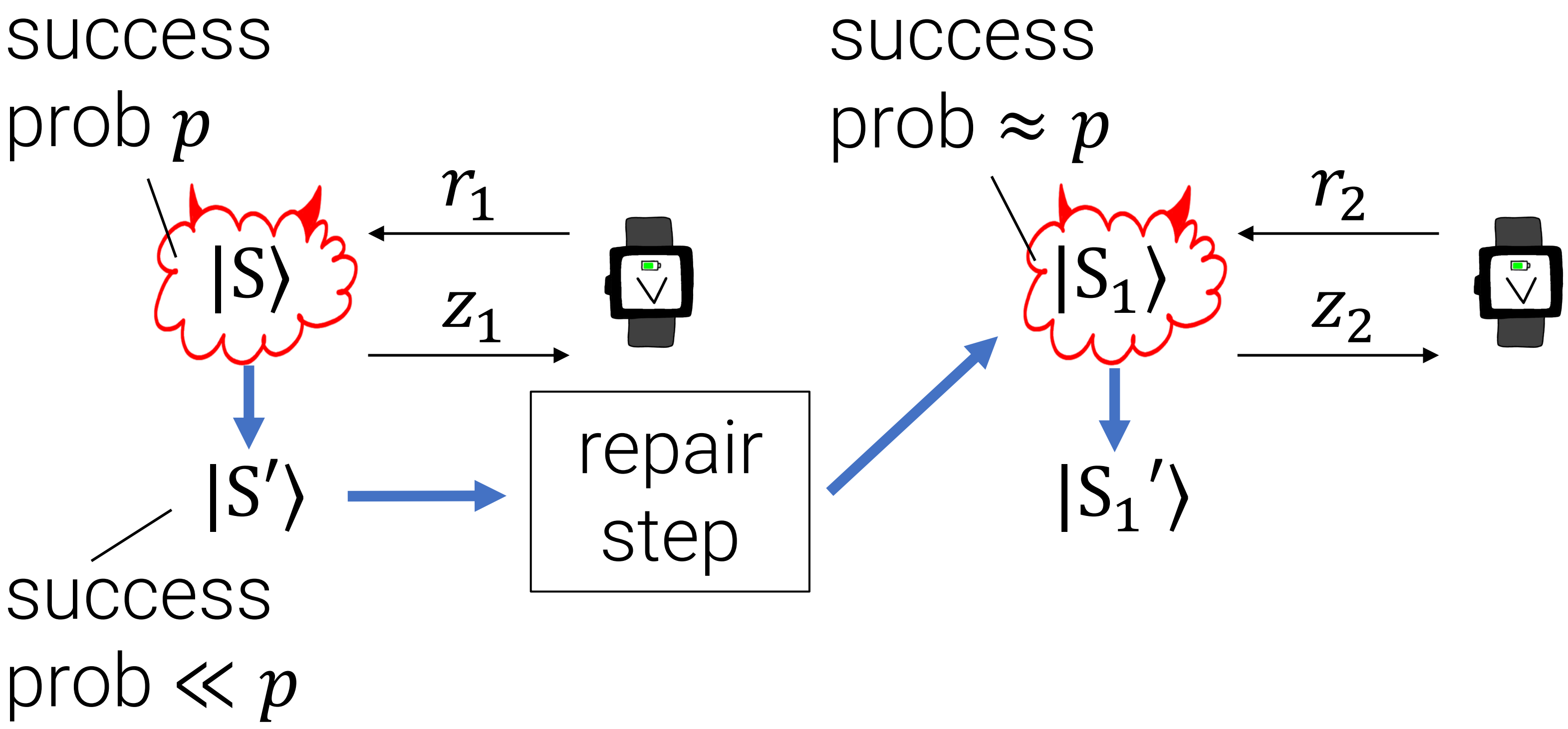
Problem: $|S'\rangle$ might not be a successful adversary!

This work: we devise a “repair” procedure to restore the original success probability.



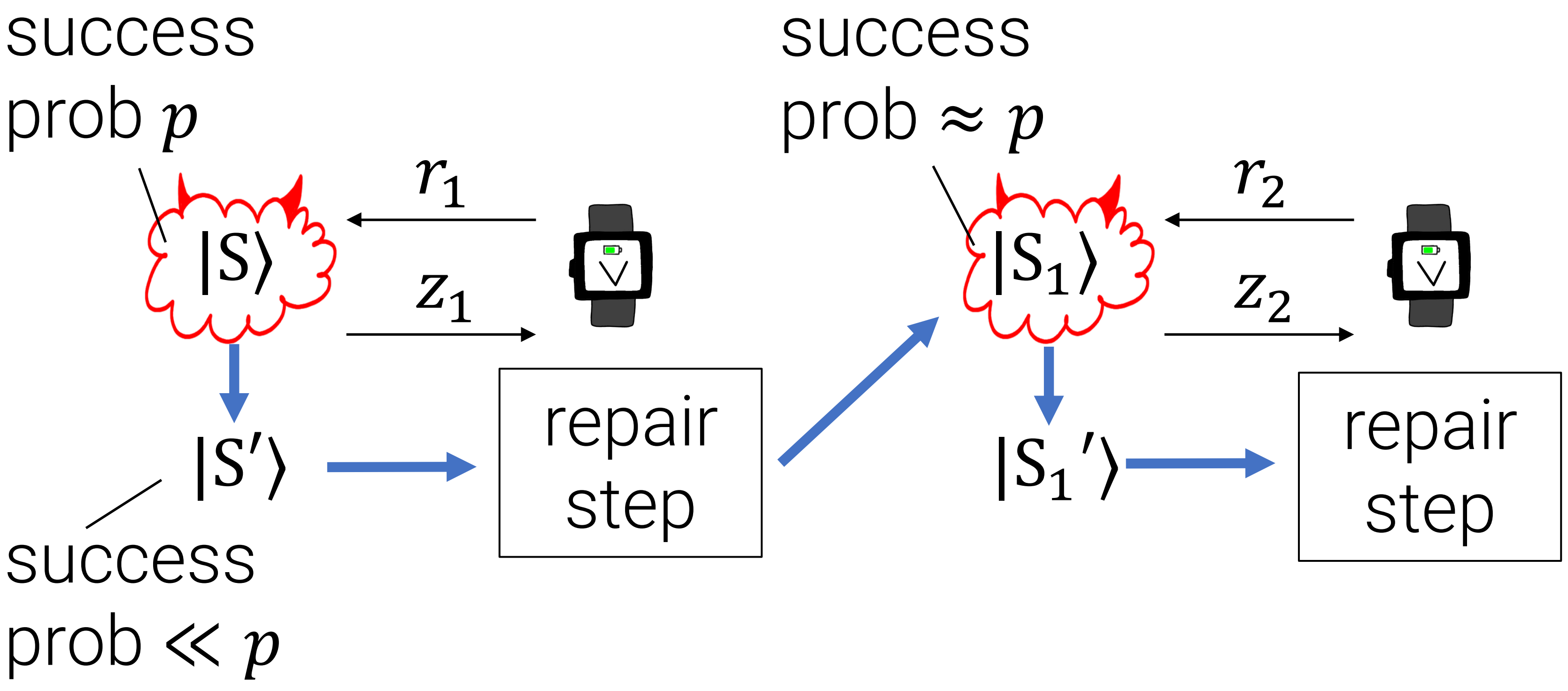
Problem: $|S'\rangle$ might not be a successful adversary!

This work: we devise a "repair" procedure to restore the original success probability.



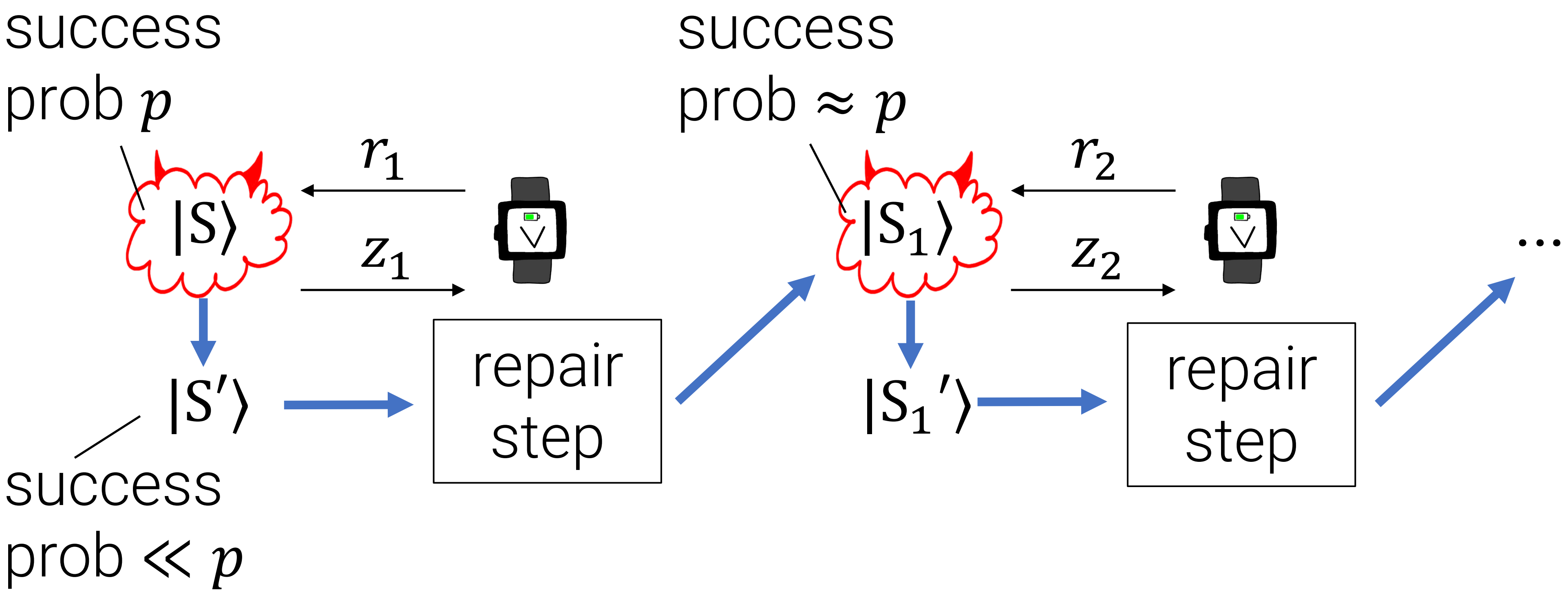
Problem: $|S'\rangle$ might not be a successful adversary!

This work: we devise a “repair” procedure to restore the original success probability.



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

This work: we devise a "repair" procedure to restore the original success probability.



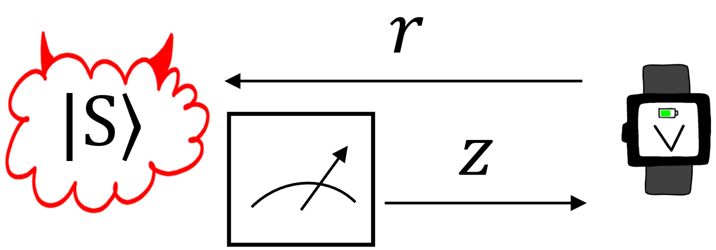
Problem: $|S'\rangle$ might not be a successful adversary!

This work: we devise a "repair" procedure to restore the original success probability.

First, recall a key idea from the first talk:

As long as the prover's response is "collapsing", measuring the prover's response amounts to measuring the bit indicating accept/reject.

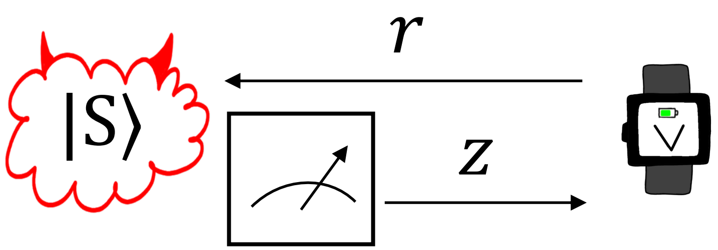
Recording the Verifier's Decision [Unruh12]



Naïve Measurement:

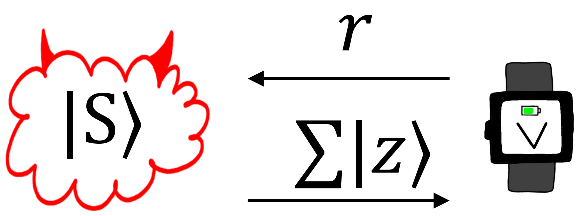
Measure $\sum |z\rangle$ right away.

Recording the Verifier's Decision [Unruh12]



Naïve Measurement:

Measure $\sum|z\rangle$ right away.

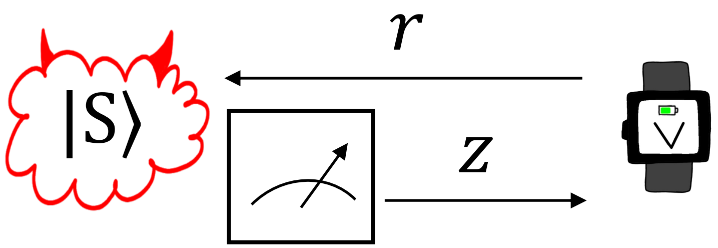


"Lazy" Measurement:

(1) Compute + measure $V(r, z)$.

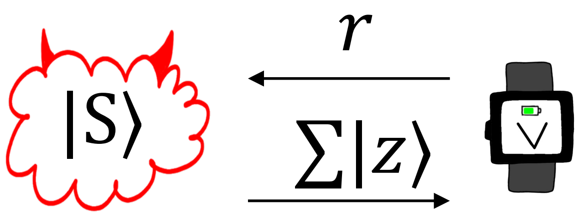
(2) Measure z if $V(r, z) = 1$.

Recording the Verifier's Decision [Unruh12]



Naïve Measurement:

Measure $\sum|z\rangle$ right away.



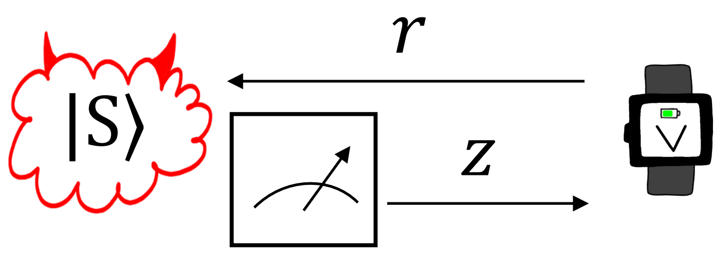
"Lazy" Measurement:

(1) Compute + measure $V(r,z)$.

(2) Measure z if $V(r,z) = 1$.

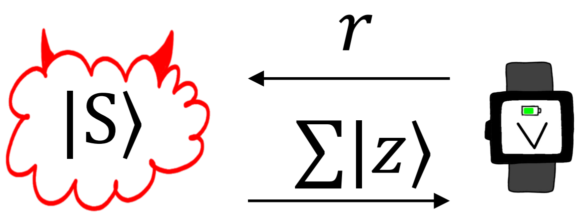
> [U16]: As long as z is "collapsing", measurement in step (2) causes *is undetectable to the prover*!

Recording the Verifier's Decision [Unruh12]



Naïve Measurement:

Measure $\sum|z\rangle$ right away.



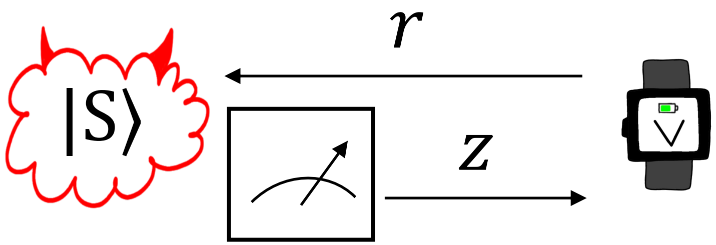
"Lazy" Measurement:

(1) Compute + measure $V(r,z)$.
(2) Measure z if $V(r,z) = 1$.

[U16]: As long as z is "collapsing", measurement in step (2) causes *is undetectable to the prover*!

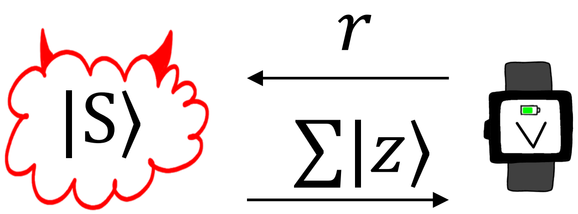
- Kilian's protocol satisfies this property if the CRHF is a "collapsing hash function".

Recording the Verifier's Decision [Unruh12]



Naïve Measurement:

Measure $\sum|z\rangle$ right away.



"Lazy" Measurement:

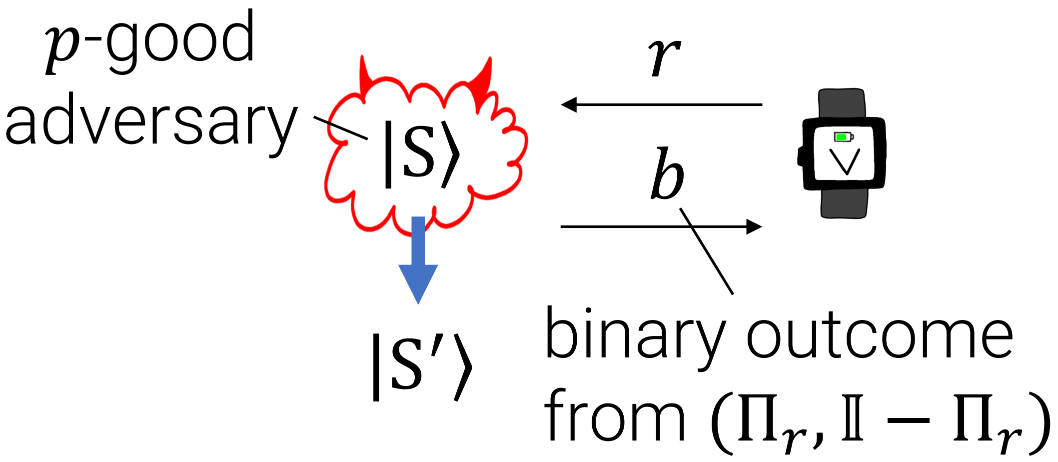
(1) Compute + measure $V(r,z)$.

(2) Measure z if $V(r,z) = 1$.

As in the first talk, collapsing allows us to treat the measurement of the prover's response on r as a **binary-outcome measurement** $(\Pi_r, \mathbb{I} - \Pi_r)$

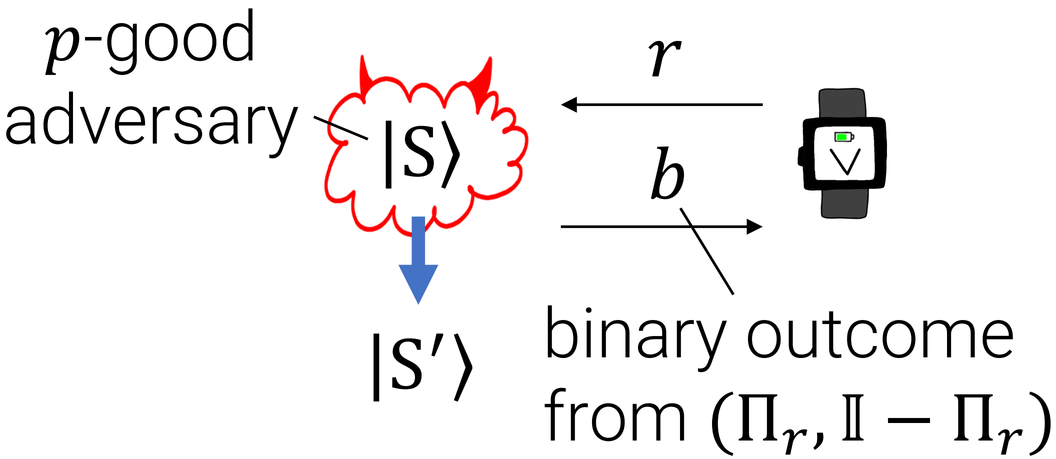
Rest of this talk: “repair” the prover’s state after a binary-outcome measurement.

Repairing the Prover After Measurement



State repair task:
Given $|S'\rangle$, efficiently produce a p-good adversary state.

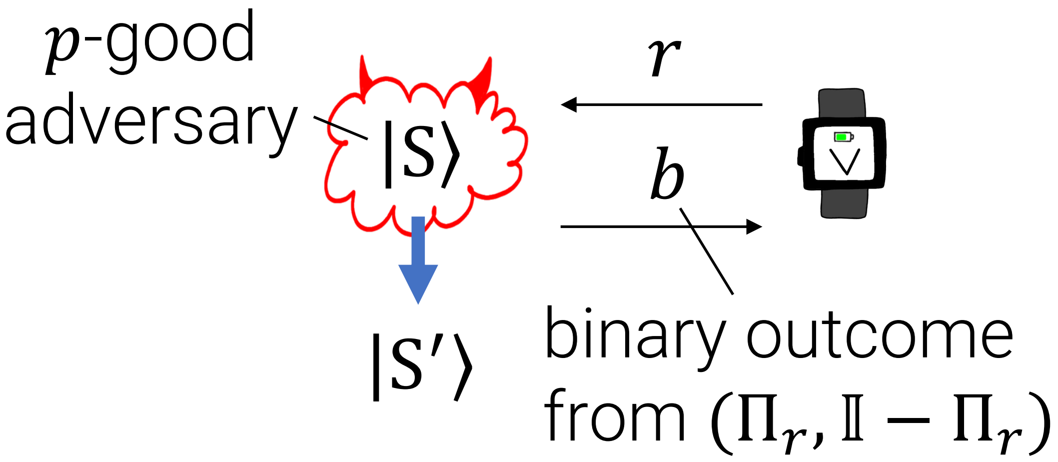
Repairing the Prover After Measurement



State repair task:
Given $|S'\rangle$, efficiently produce a p-good adversary state.

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

Repairing the Prover After Measurement

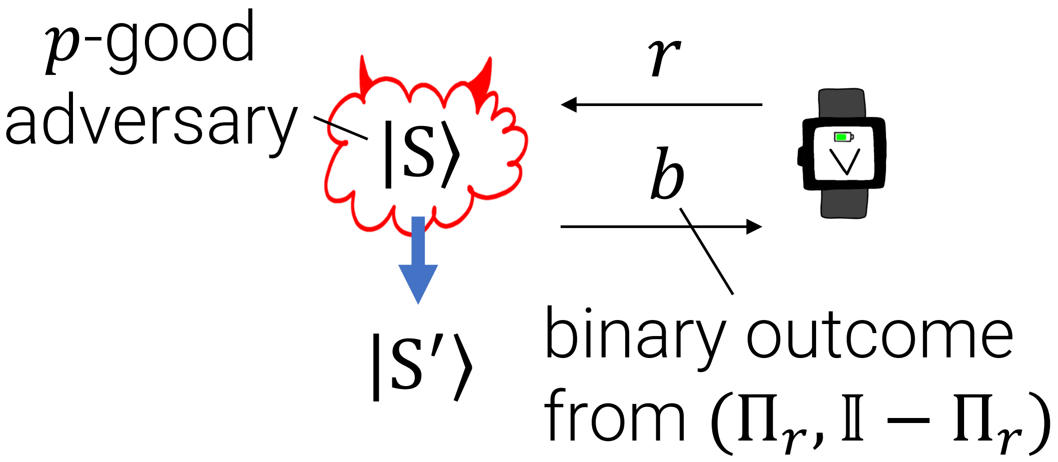


State repair task:
Given $|S'\rangle$, efficiently produce a p-good adversary state.

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

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

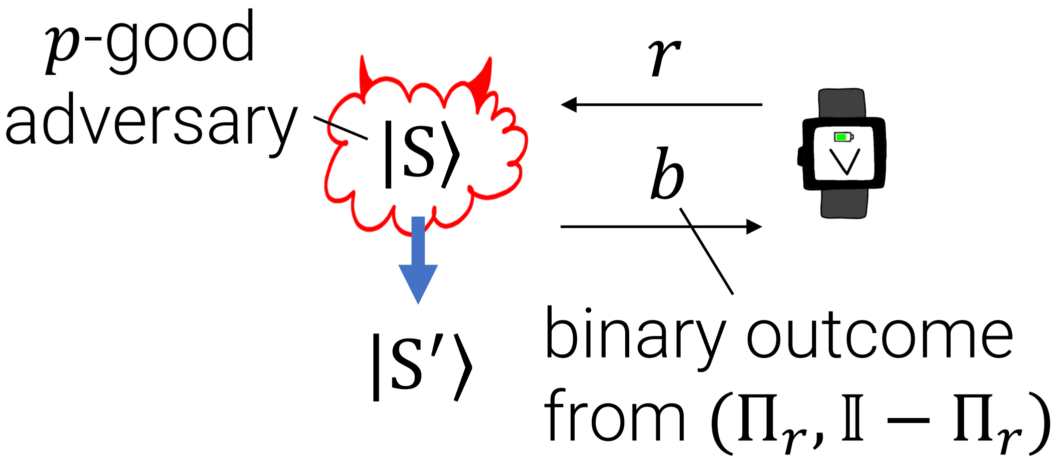
Repairing the Prover After Measurement



State repair task:
Given $|S'\rangle$, efficiently produce a p-good 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

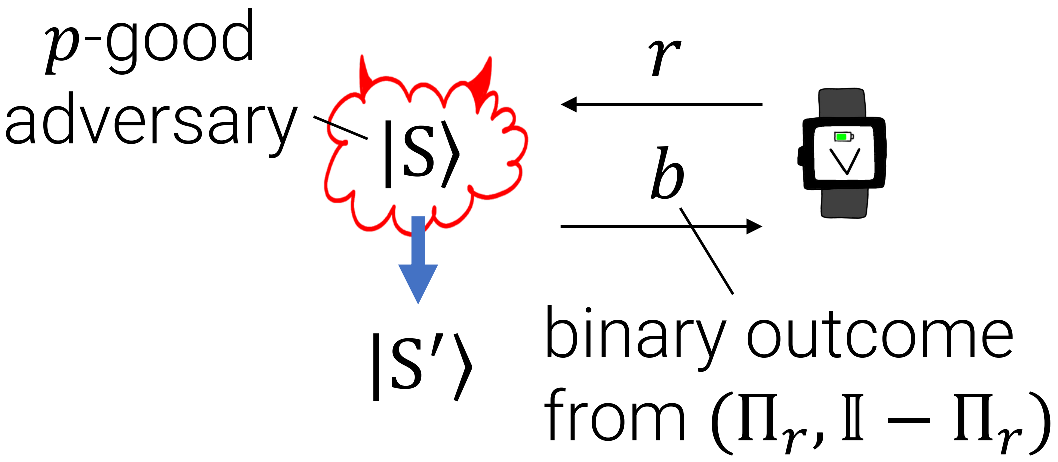


State repair task:
Given $|S'\rangle$, efficiently produce a p-good adversary state.

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



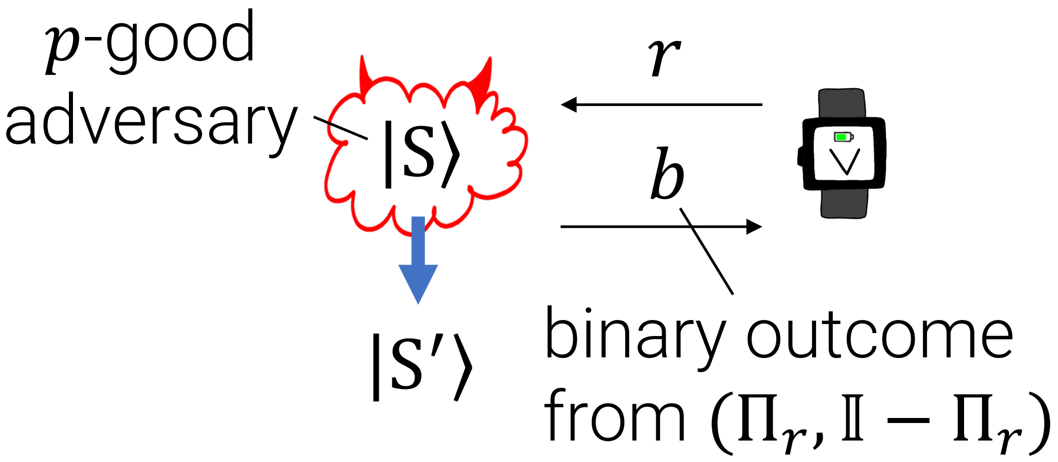
State repair task:
Given $|S'\rangle$, efficiently produce a p-good adversary state.

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 $|S'\rangle$, efficiently produce a p-good adversary state.

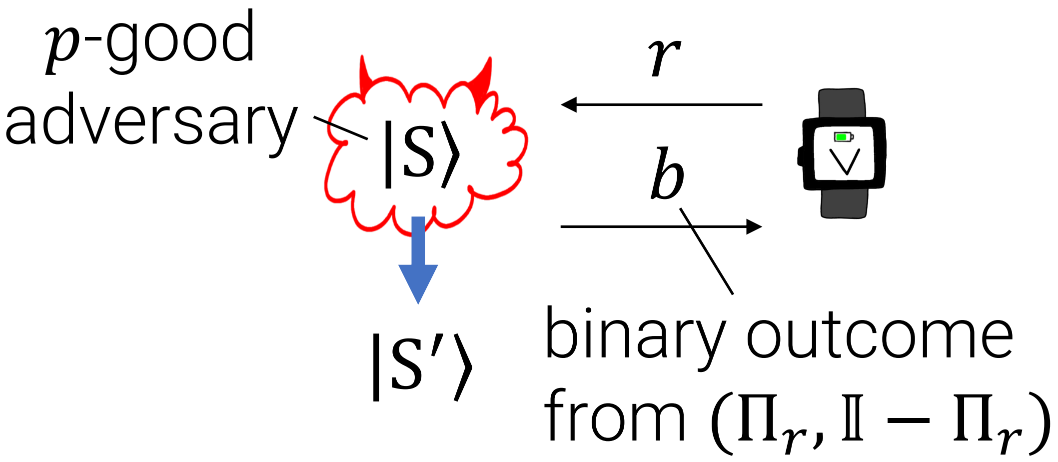
Watrous rewinding task: given verifier state $|\psi\rangle$ and projector Π_G indicating "successful simulation", output the state $\Pi_G|\psi\rangle_V|0\rangle_{Aux}$

Algorithm: alternate Π_0, Π_G measurements until Π_G accepts.

Why these projectors?

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

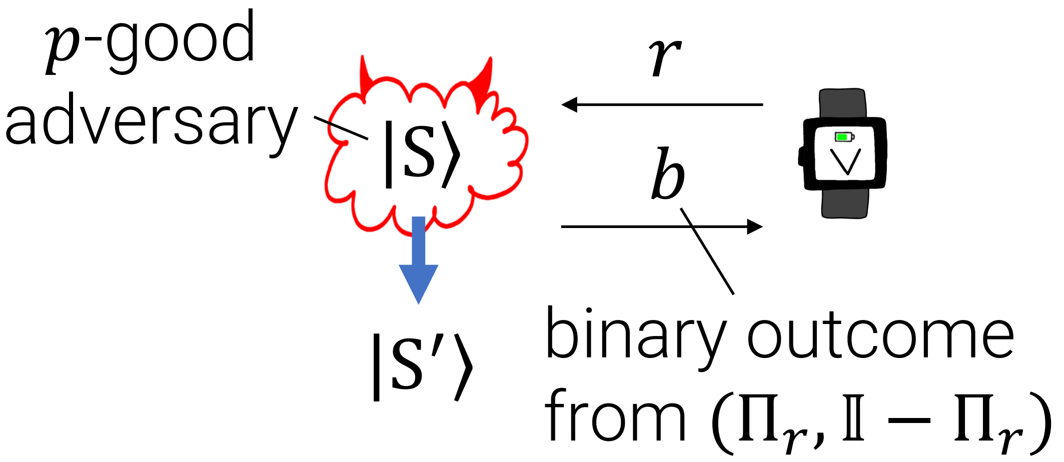
Repairing the Prover After Measurement



State repair task:
Given $|S'\rangle$, efficiently produce a p-good adversary state.

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

Repairing the Prover After Measurement

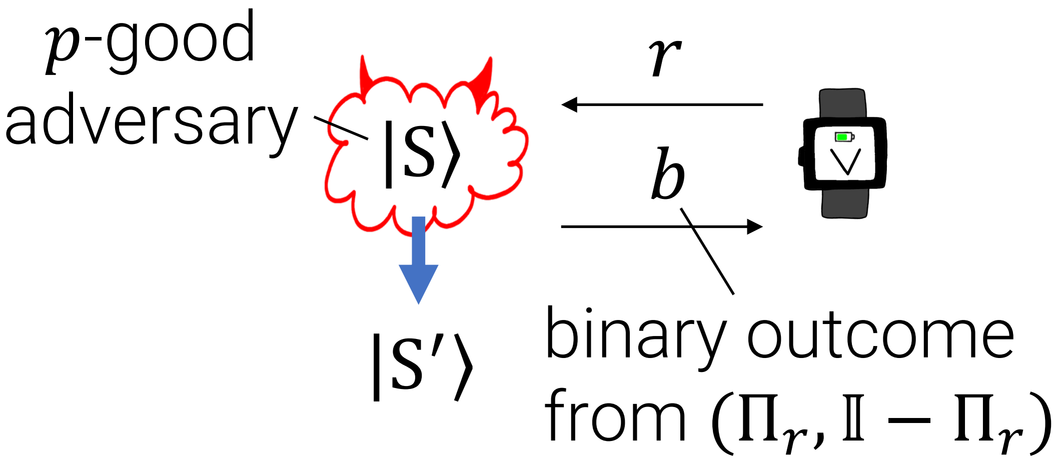


State repair task:
Given $|S'\rangle$, efficiently produce a p-good adversary state.

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

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

Repairing the Prover After Measurement



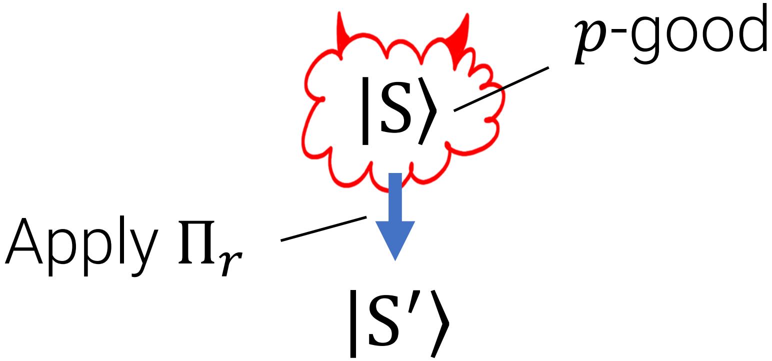
State repair task:
Given $|S'\rangle$, efficiently produce a p-good adversary state.

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

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

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

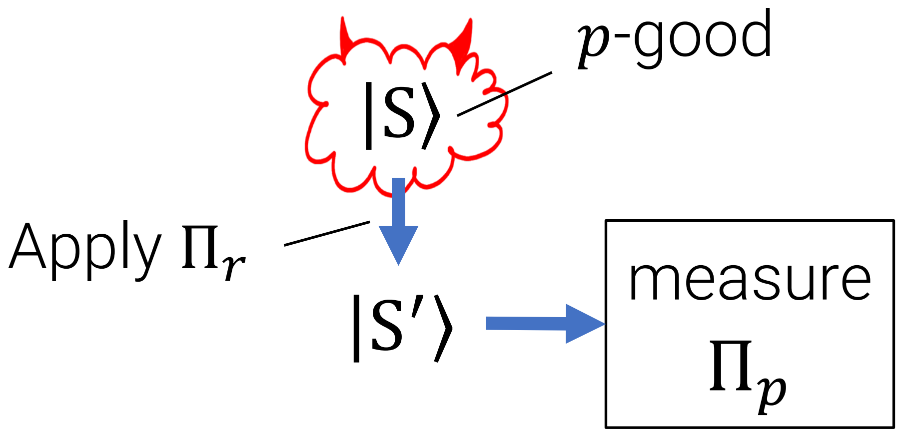
Repairing the Prover After Measurement



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

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

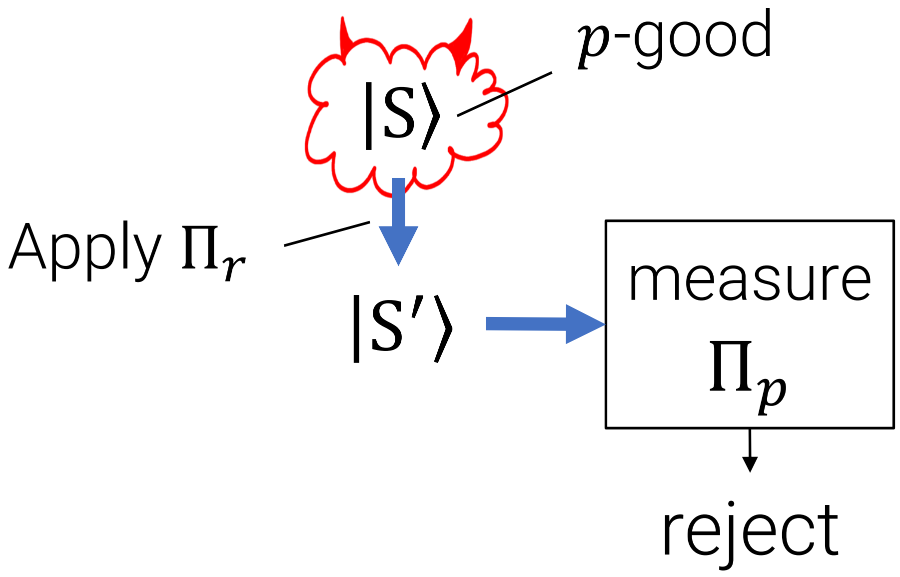
Repairing the Prover After Measurement



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

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

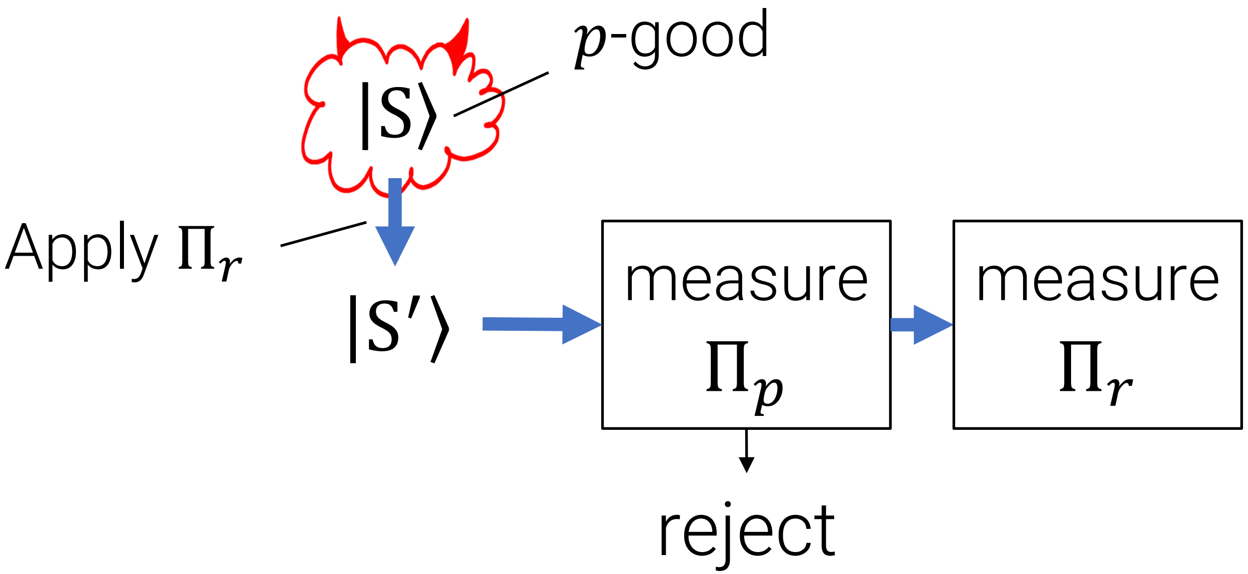
Repairing the Prover After Measurement



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

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

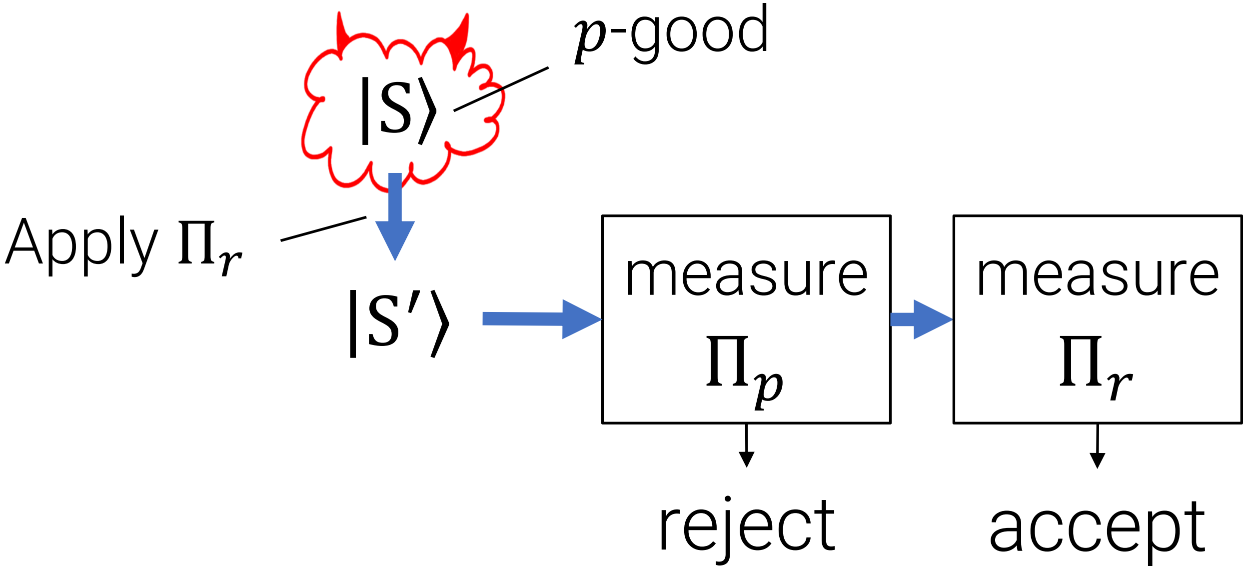
Repairing the Prover After Measurement



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

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

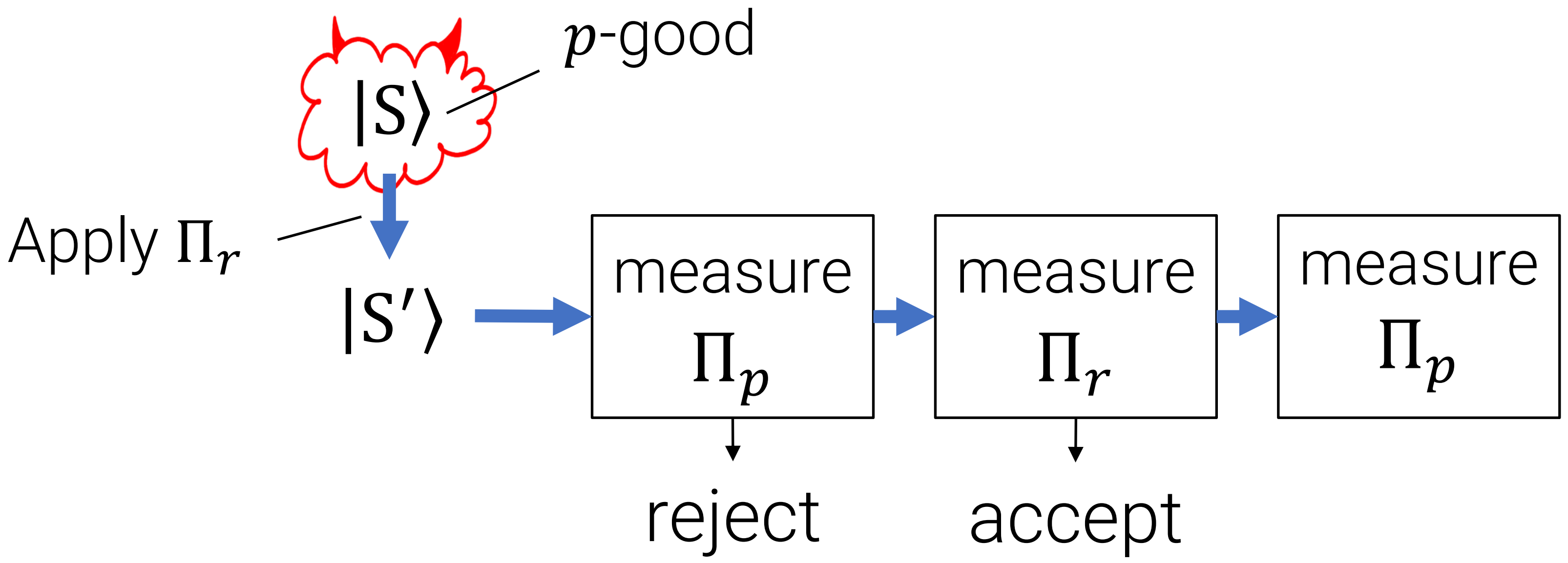
Repairing the Prover After Measurement



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

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

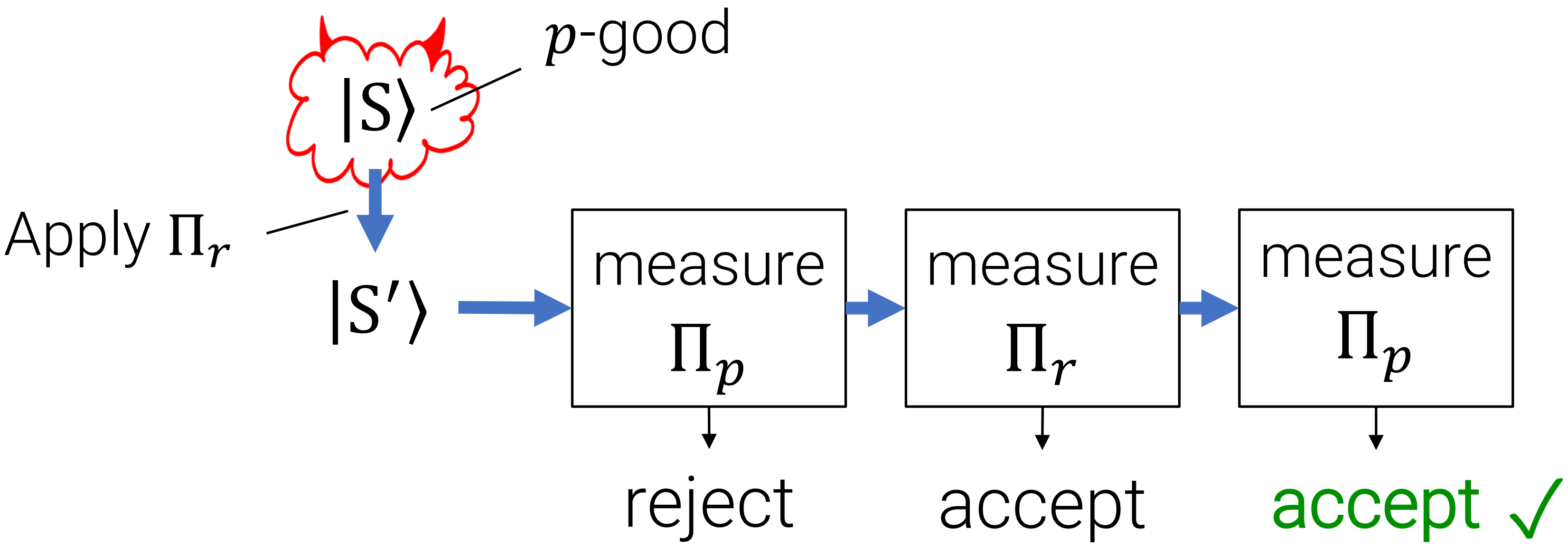
Repairing the Prover After Measurement



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

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

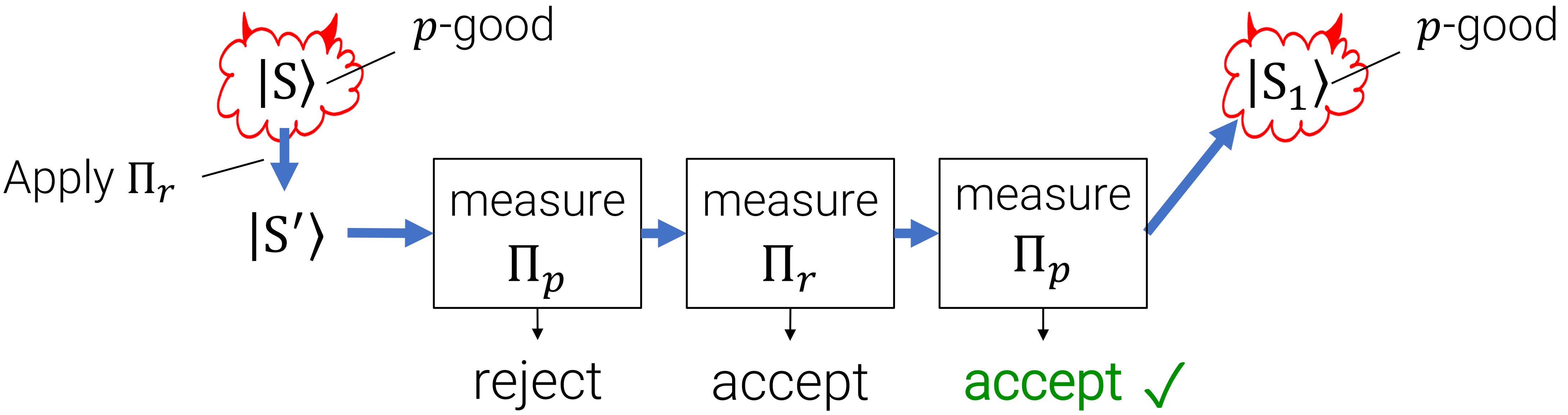
Repairing the Prover After Measurement



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

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

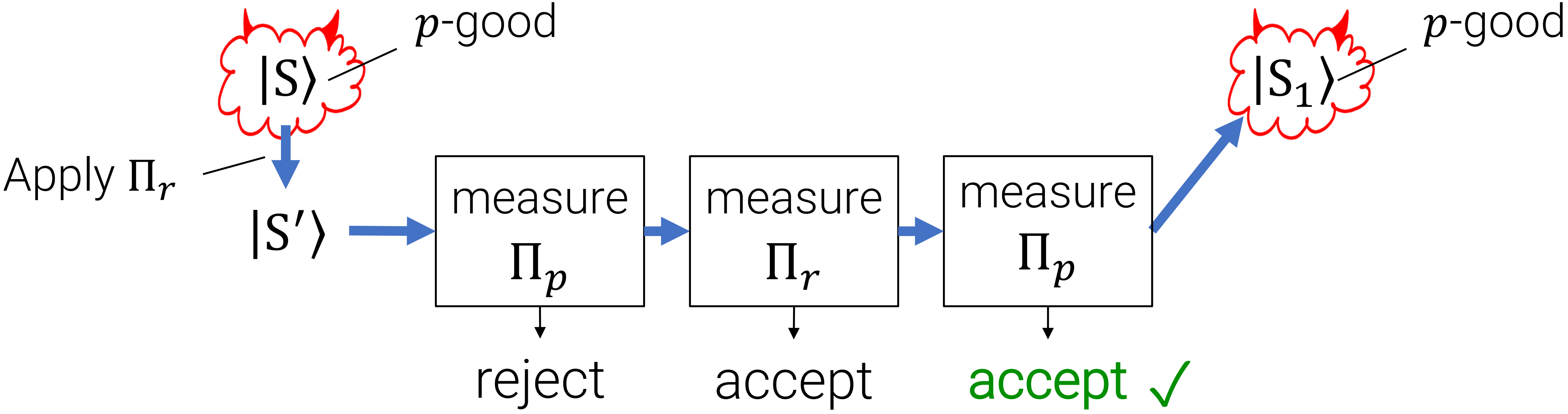
Repairing the Prover After Measurement



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

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

Repairing the Prover After Measurement

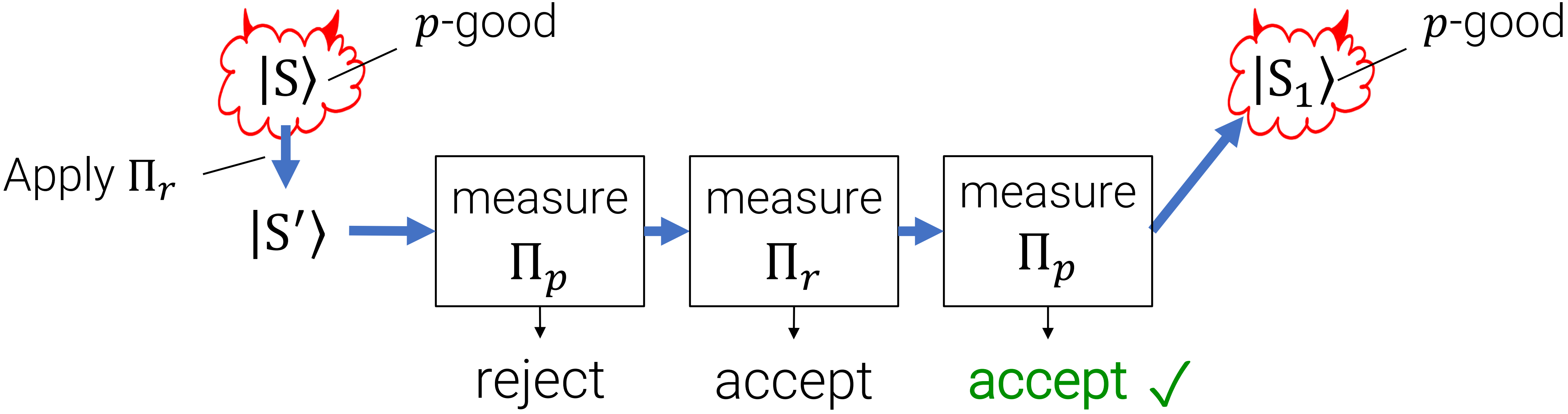


Missing Pieces

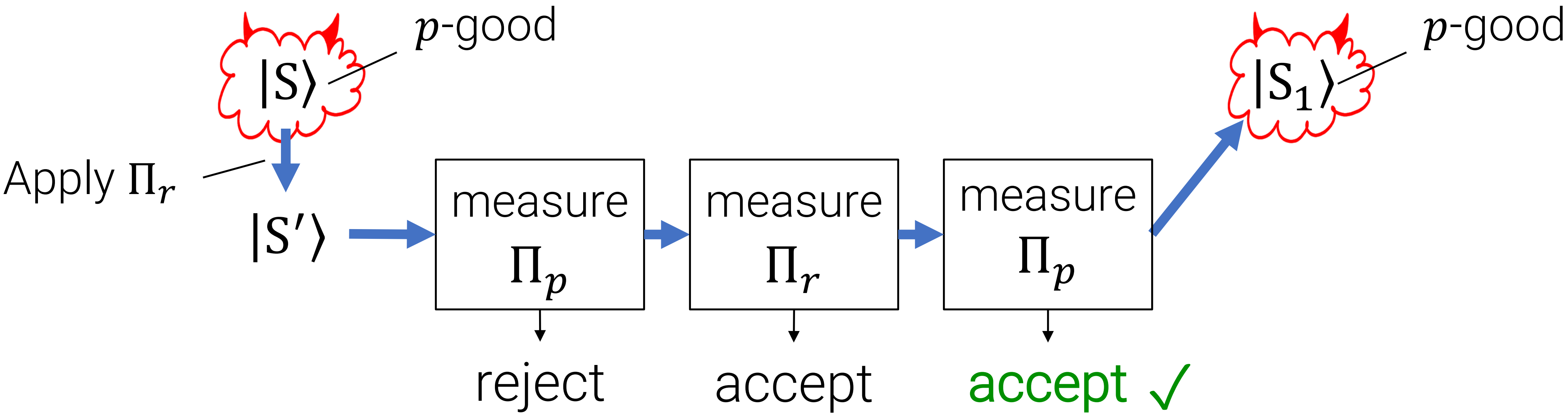
1) Why does this terminate?

2) How do we define/implement Π_p?

Why does this terminate?

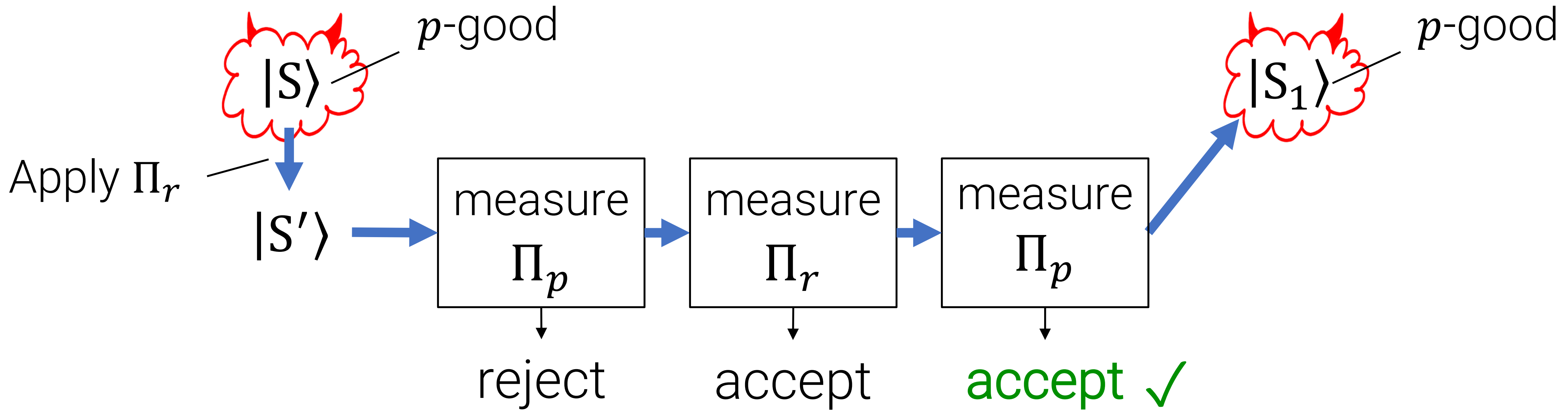


Why does this terminate?



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.

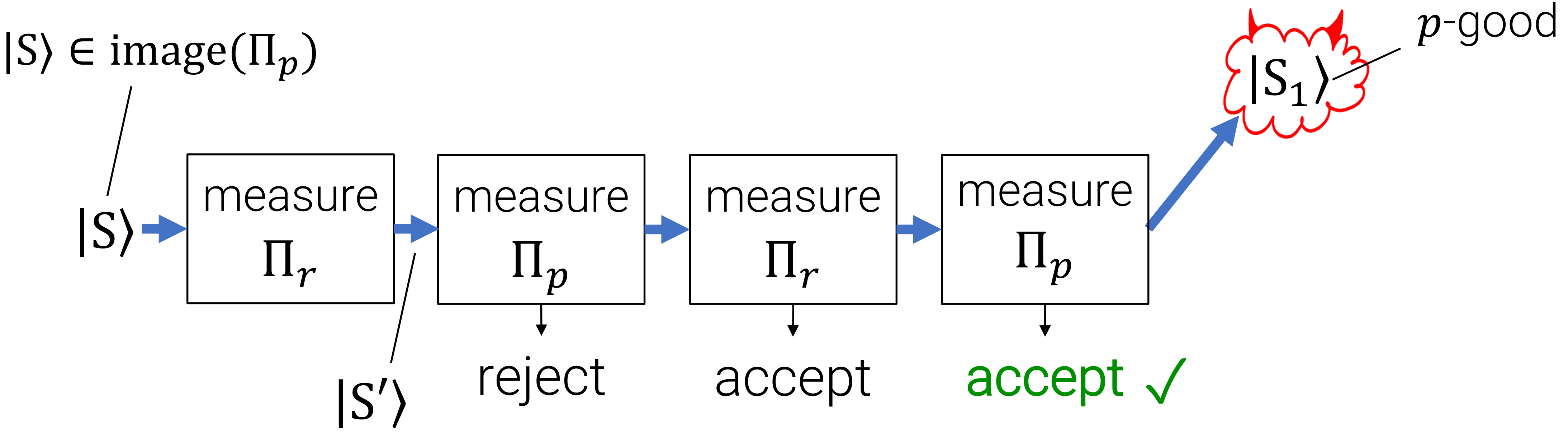
Why does this terminate?



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 $|S'\rangle$.

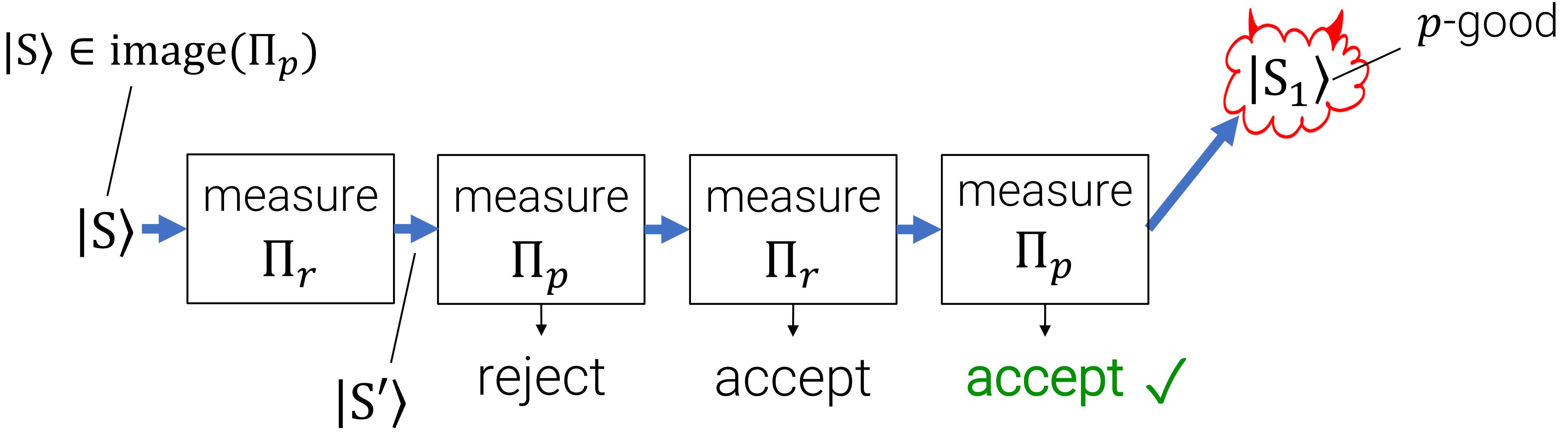
Why does this terminate?



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?

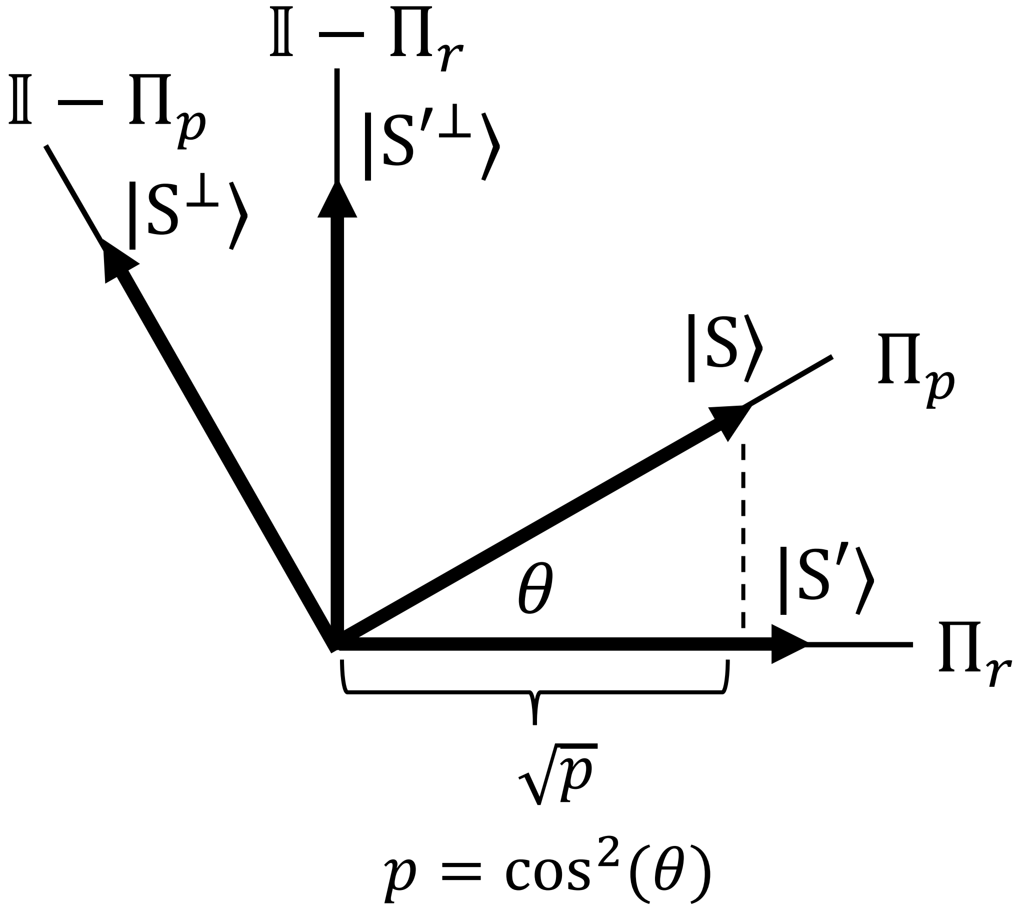


In the last talk, we used special properties of the two projectors to bound the runtime, but it's not clear what we can say about Π_r, Π_p.

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

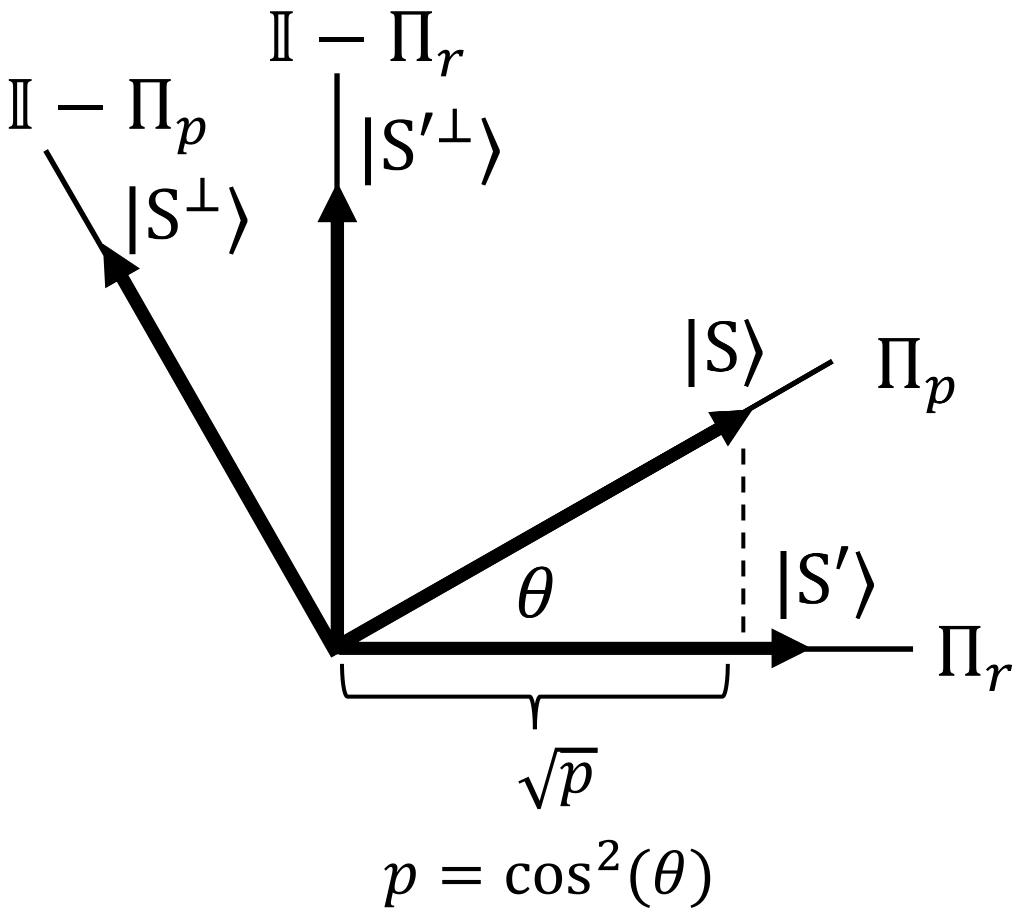
"Return to Subspace" Lemma: If we start at $|S\rangle \in \mathbf{image}(\Pi_p)$ and alternate Π_r, Π_p measurements, return to $\mathbf{image}(\Pi_p)$ in $O(1)$ expected steps.

"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.

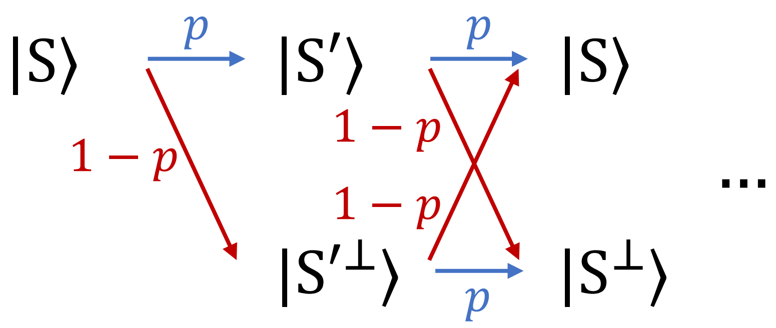


Consider the 2-D case.

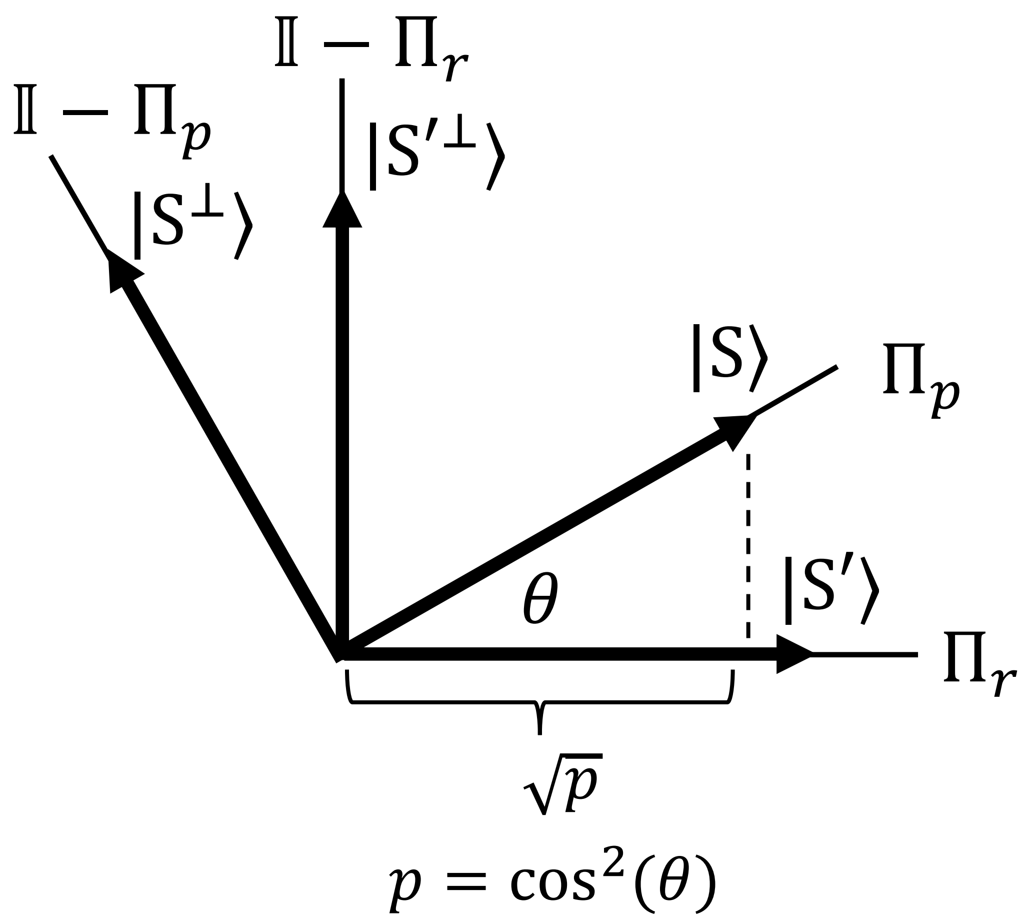
“Return to Subspace” Lemma: If we start at $|S\rangle \in \mathbf{image}(\Pi_p)$ and alternate Π_r, Π_p measurements, return to $\mathbf{image}(\Pi_p)$ in $O(1)$ expected steps.



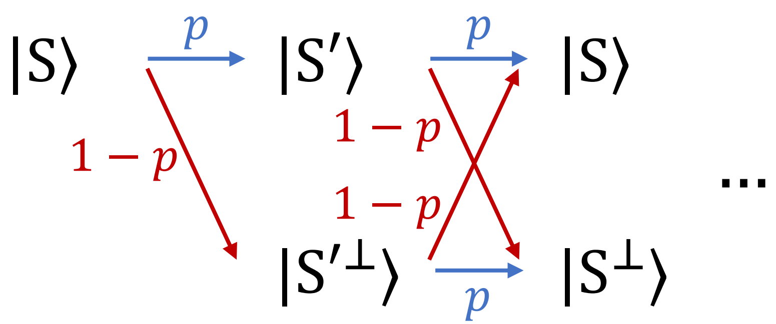
Consider the 2-D case.



"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.

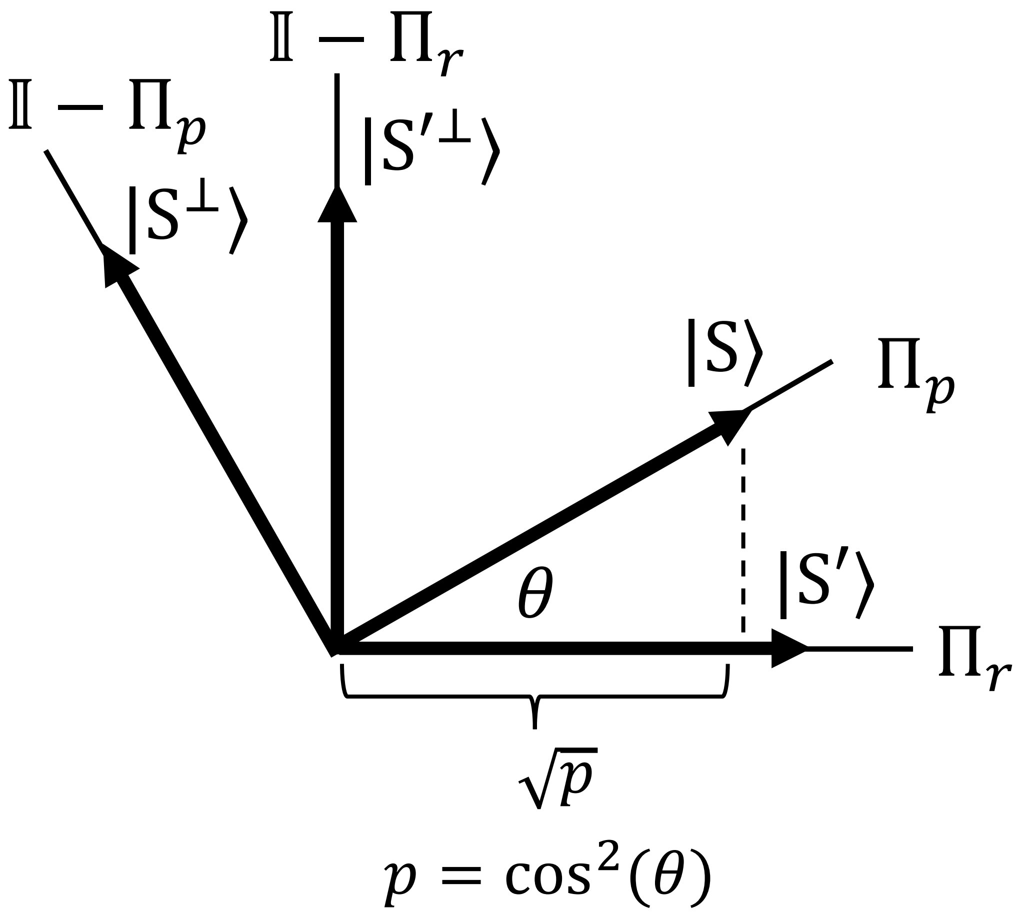


Consider the 2-D case.

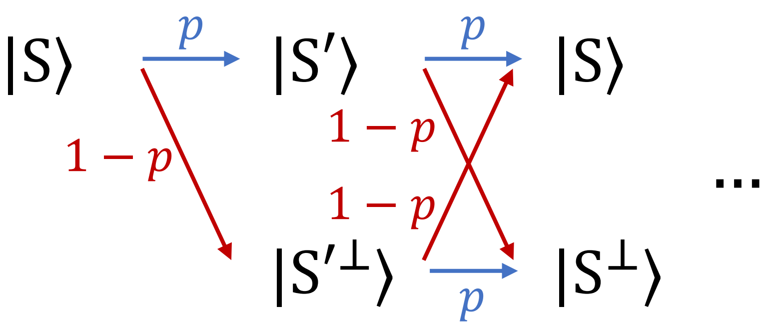


Simple calculation: time to return to $|S\rangle$ is *independent* of θ.

"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.



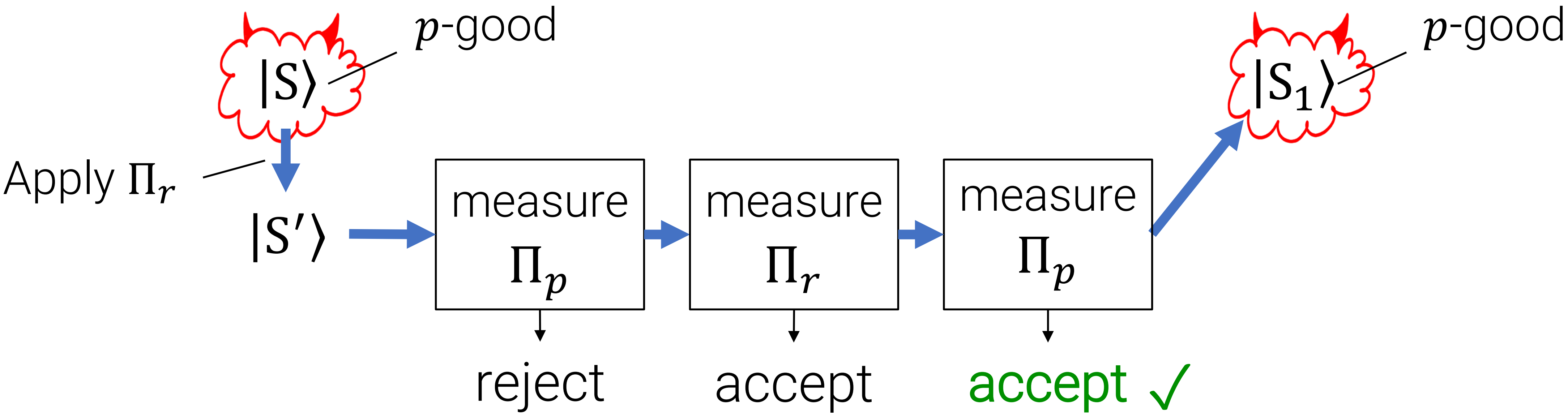
Consider the 2-D case.



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

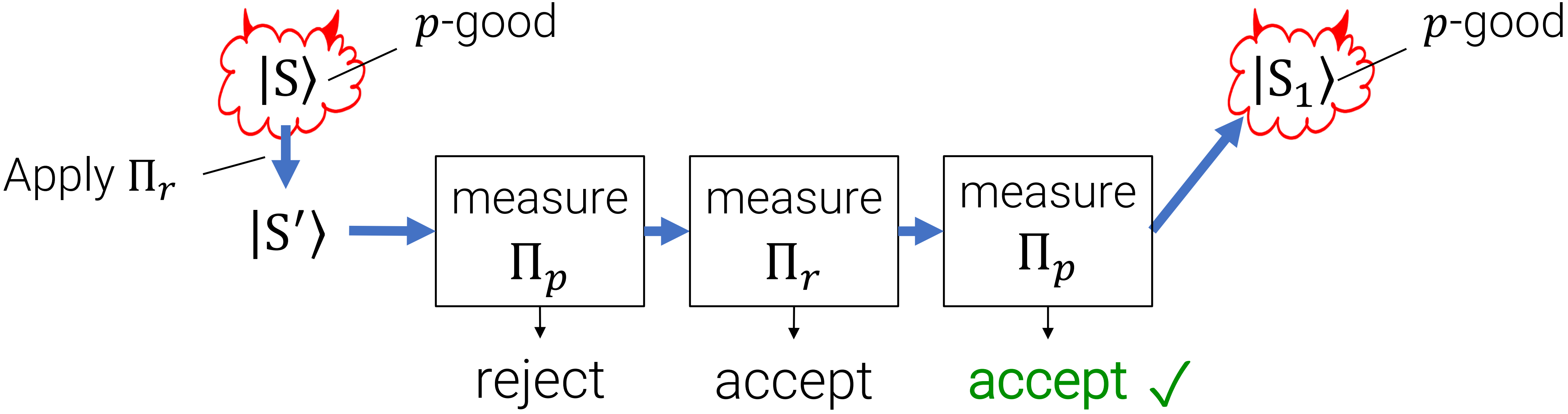


Missing Pieces

~~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*!

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)

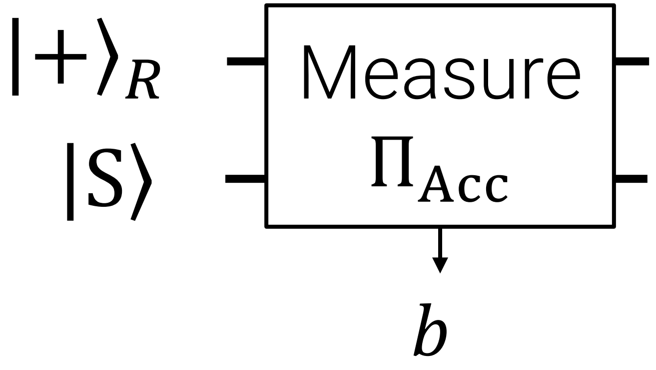
How to Estimate Success Probability [MW05,Z20]

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

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

How to Estimate Success Probability [MW05,Z20]

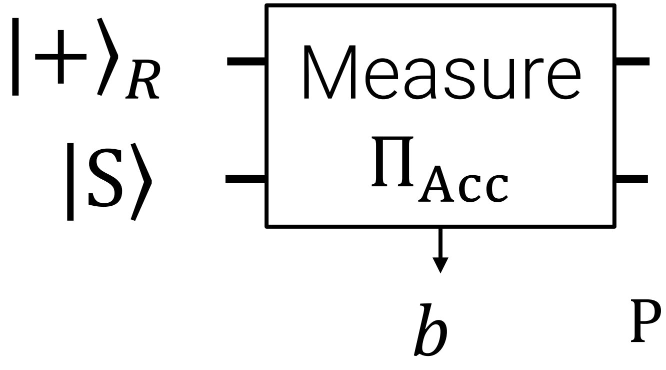
Instead of running the prover on a random challenge $r \leftarrow R$, we'll introduce a challenge register R and run the prover $|S\rangle$ in superposition on all challenges.



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

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.

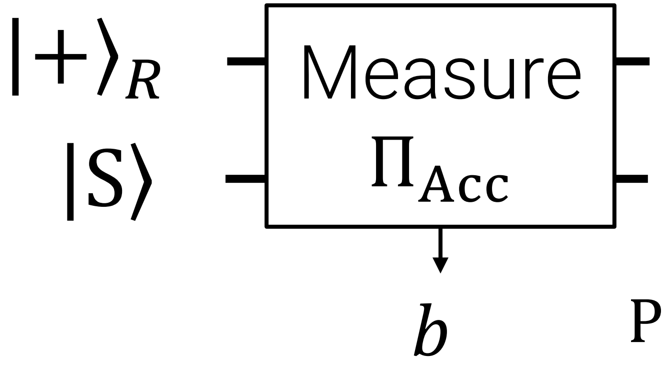


$\Pr[b = 1]$ is the success probability of $|S\rangle$

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

How to Estimate Success Probability [MW05,Z20]

[MW05, Z20]: learn success probability by alternating Π_{Acc} measurements with $\Pi_{\mathsf{Unif}} = |+\rangle\langle+|_R \otimes \mathbb{I}$ 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)
- $\Pi_{\mathsf{Acc}} \coloneqq \sum_r |r\rangle\langle r|_R \otimes \Pi_r$.
- $\Pi_{\mathsf{Unif}} \coloneqq |+\rangle\langle+|_R \otimes \mathbb{I}$.

How to Estimate Success Probability [MW05,Z20]

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

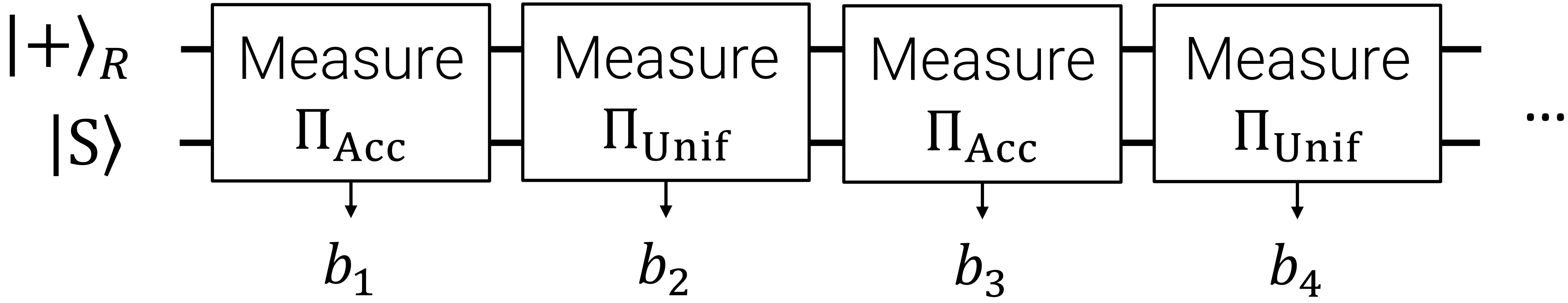
$|+\rangle_R$

$|S\rangle$

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

How to Estimate Success Probability [MW05,Z20]

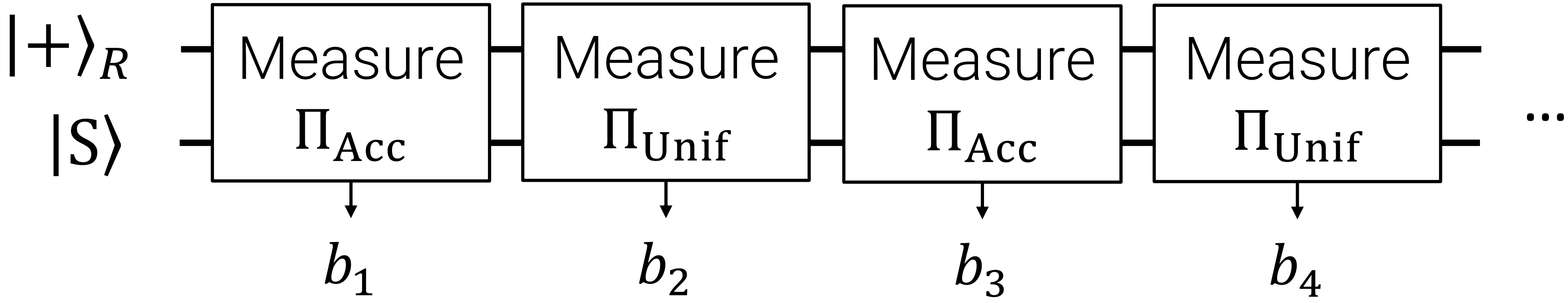
1) Initialize $|+_R\rangle|S\rangle$.
2) Alternate $M_{\mathrm{Acc}}, M_{\mathrm{Unif}}$ measurements, obtaining $(b_1, b_2, \ldots, b_T)$



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

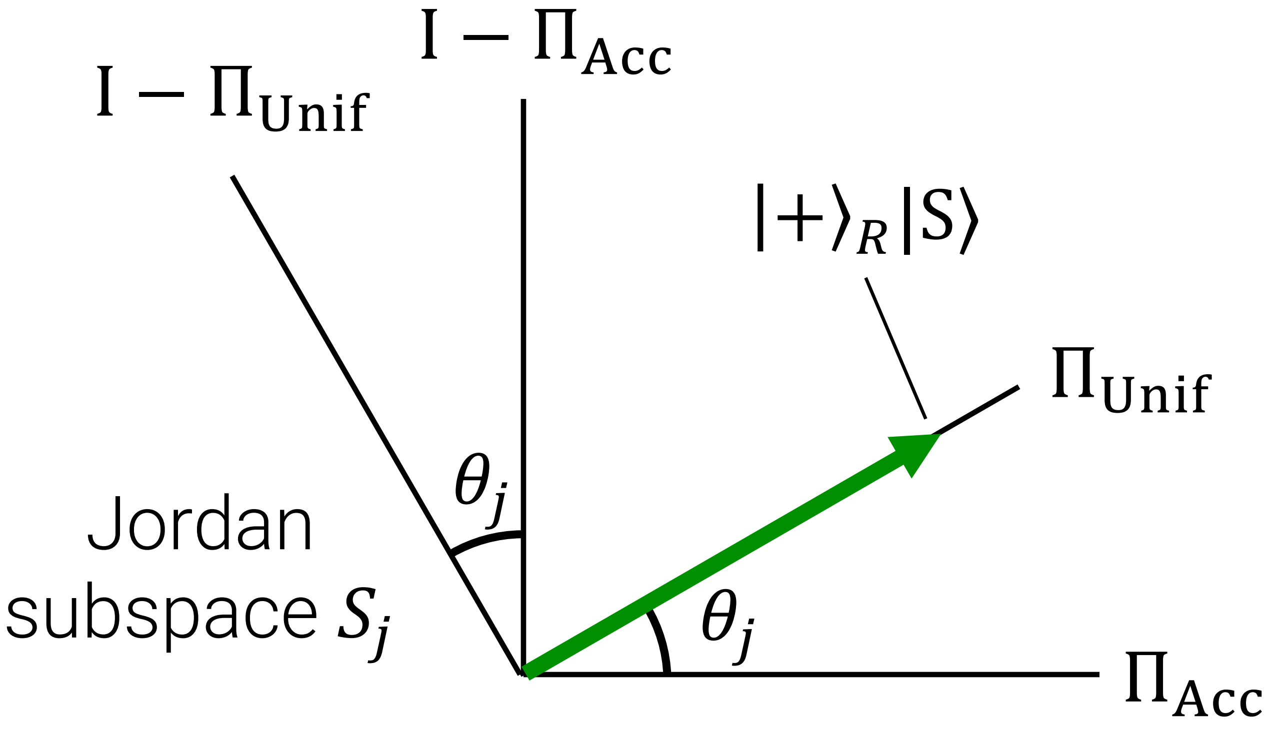
How to Estimate Success Probability [MW05,Z20]

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 $p = [\#\text{ of times } b_i = b_{i+1}]/(T-1)$



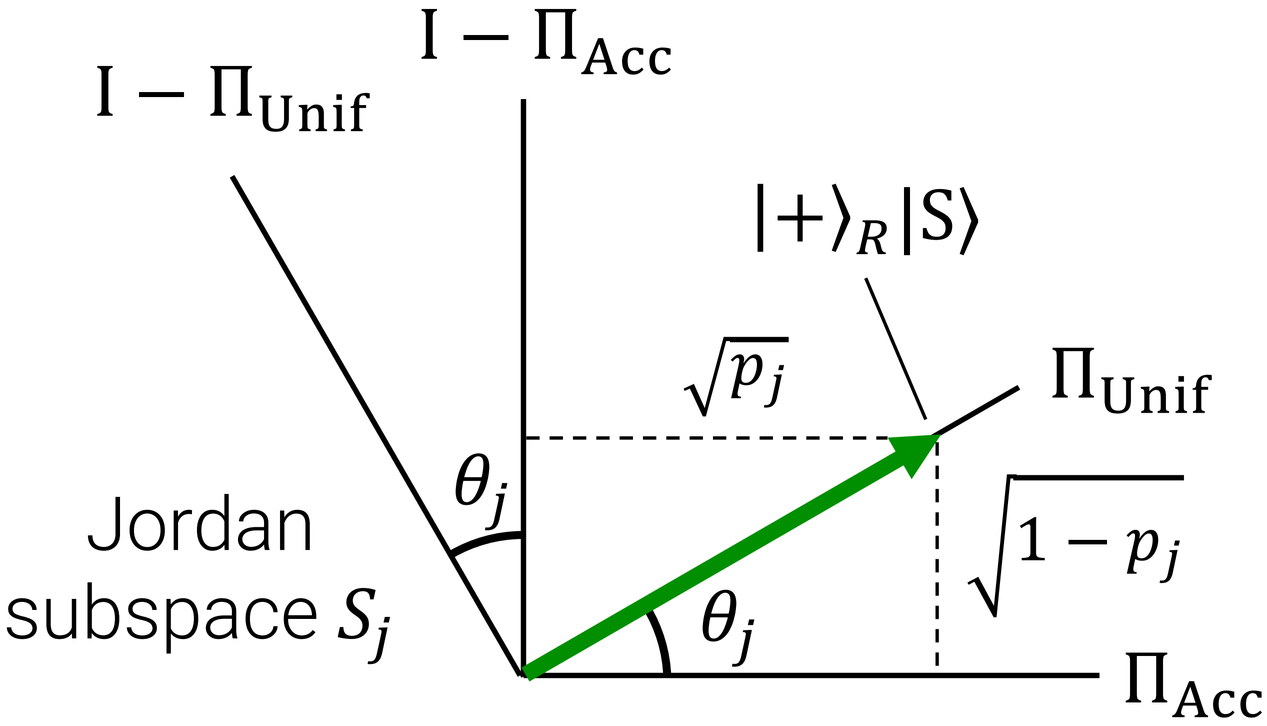
- $|+\rangle_R \coloneqq \frac{1}{\sqrt{R}} \sum_{r\in R} |r\rangle$ (uniform superposition of challenges)
- $\Pi_{\mathrm{Acc}} \coloneqq \sum_r |r\rangle\langle r|_R \otimes \Pi_r$.
- $\Pi_{\mathrm{Unif}} \coloneqq |+\rangle\langle +|_R \otimes \mathbb{I}$.

Why does this work?



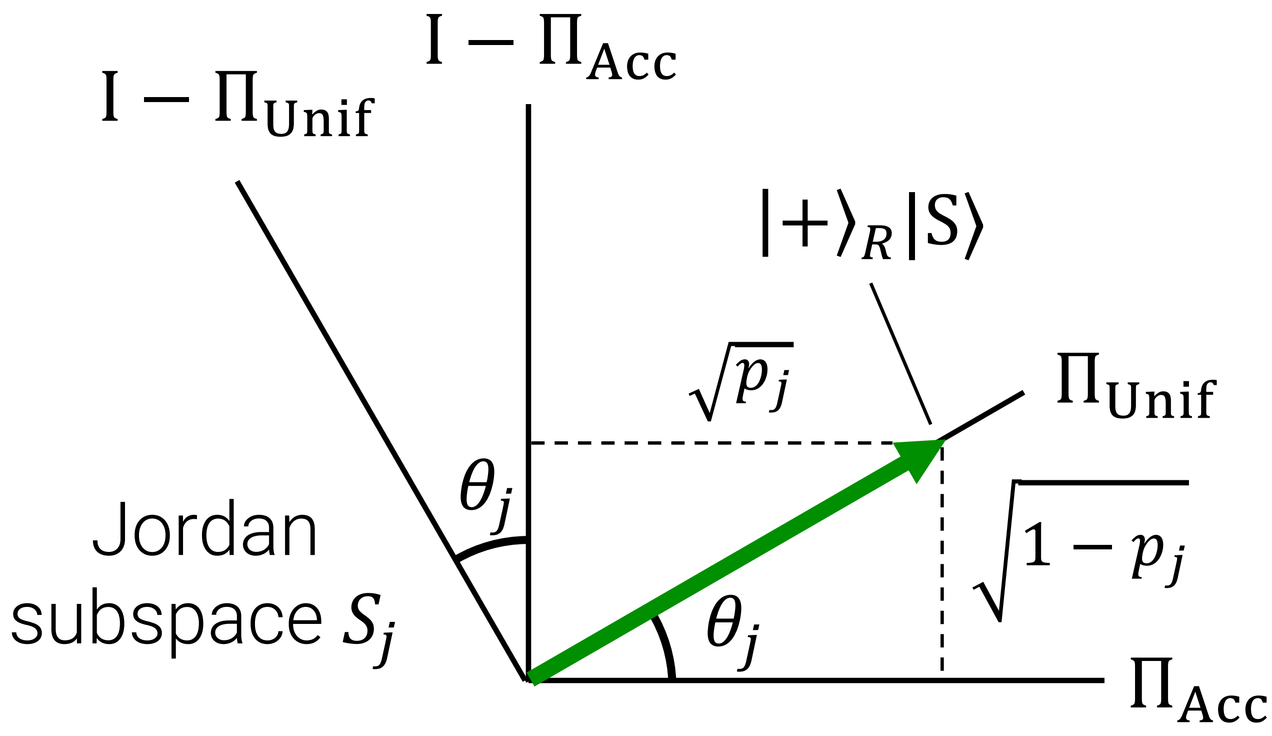
Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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



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

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

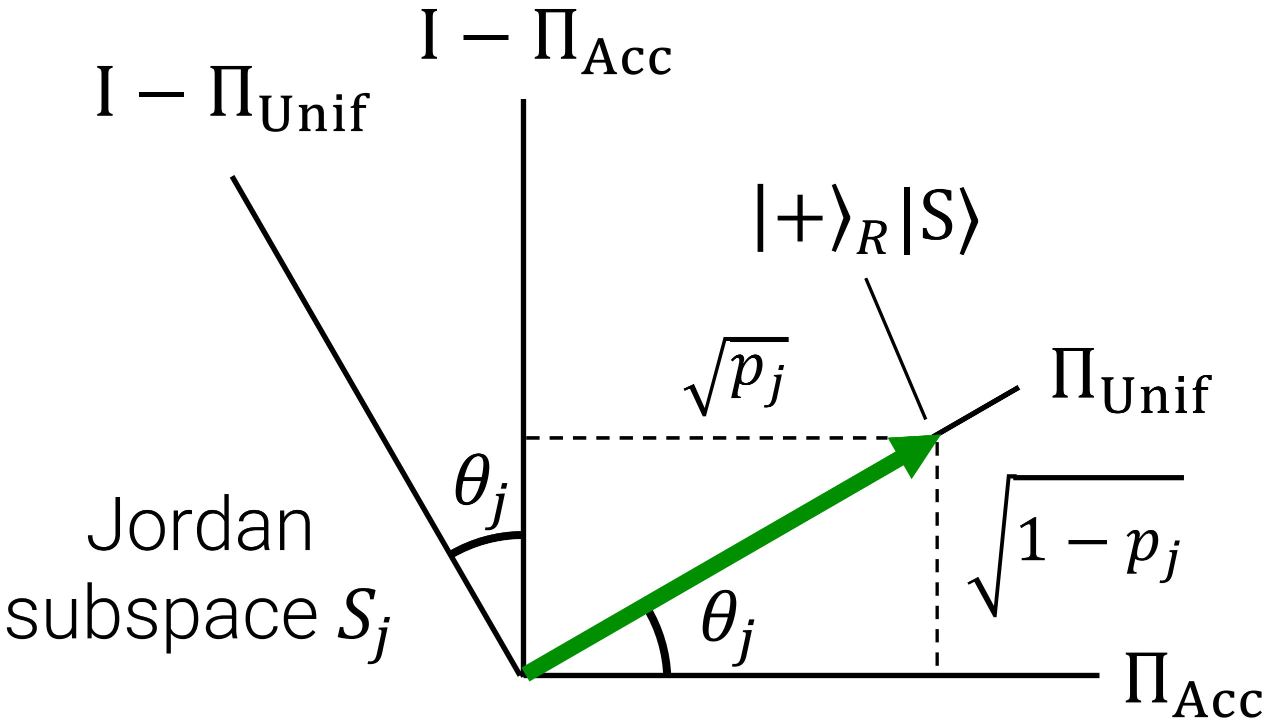


Eigenvalue $p_j = \cos^2(\theta_j) = \|\Pi_{\mathrm{Acc}}\,|+\rangle_R|\mathrm{S}\rangle\|^2$

(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 $|\mathrm{S}\rangle$.

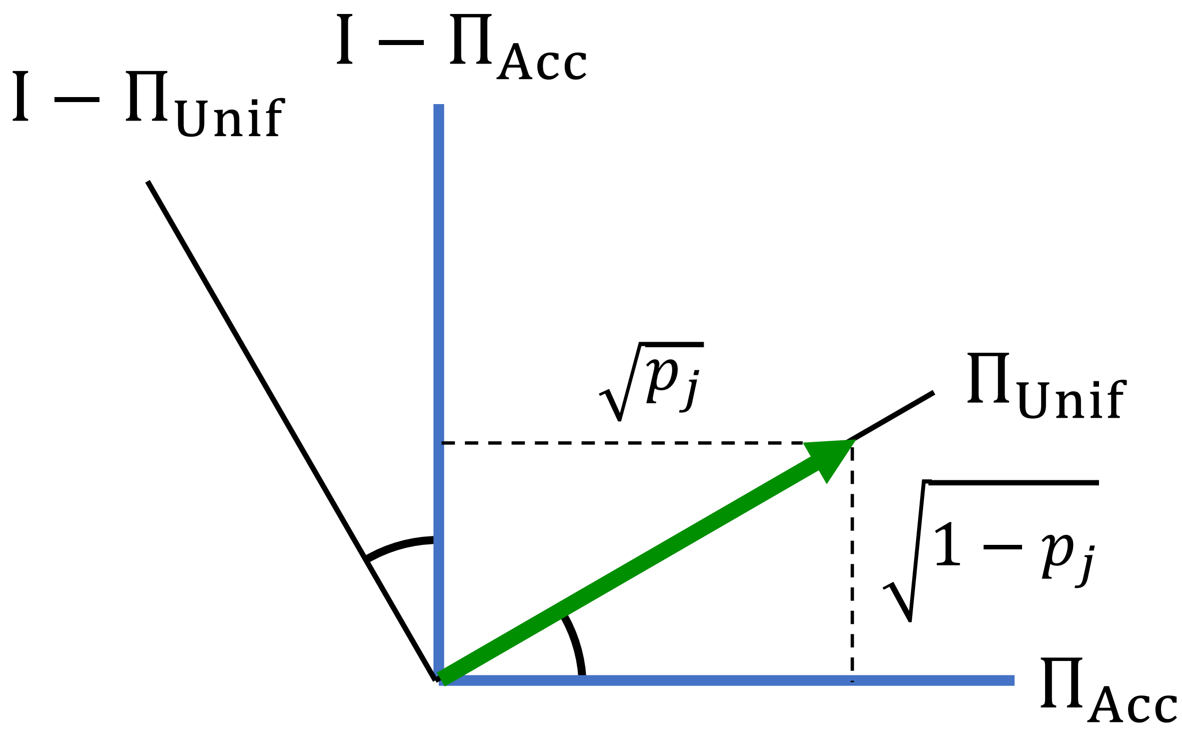


Suppose $|+\rangle_R|\mathrm{S}\rangle$ lies in a 2-dim Jordan subspace S_j.

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

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

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



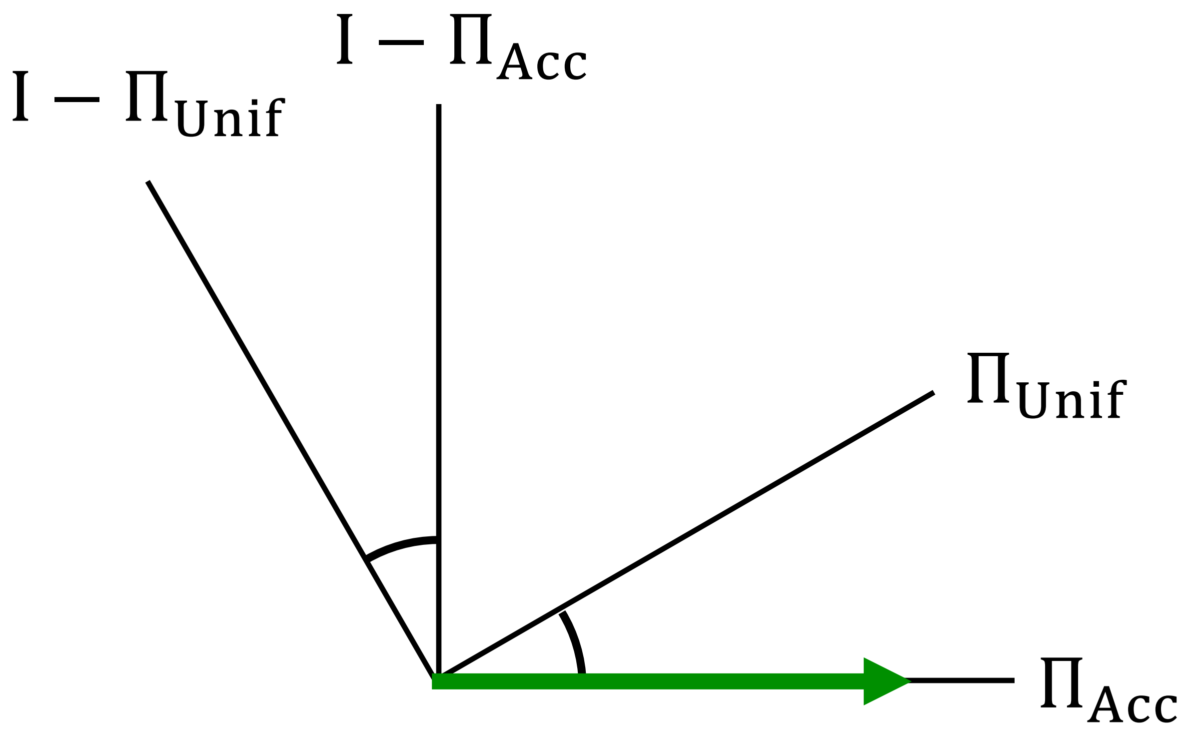
Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

Unif

$b_0 = 1$

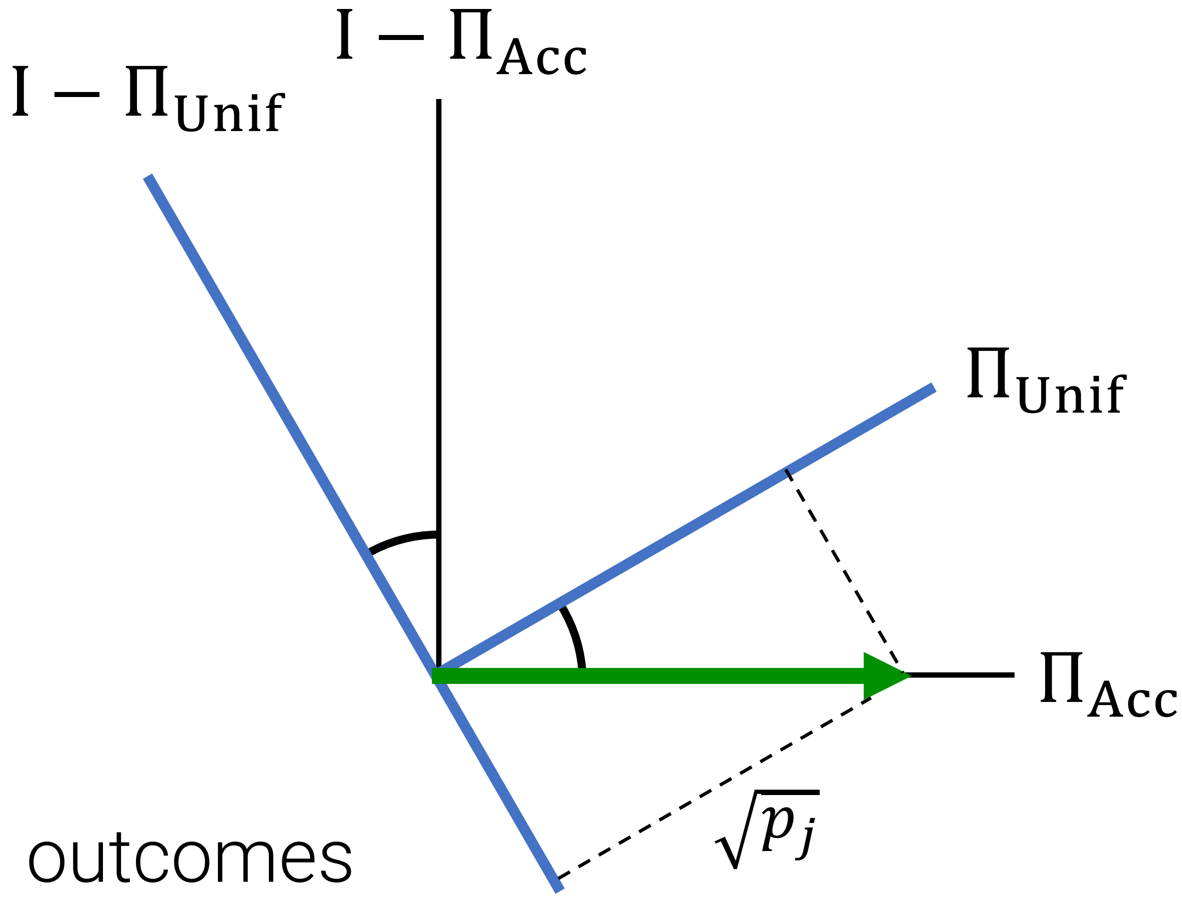


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

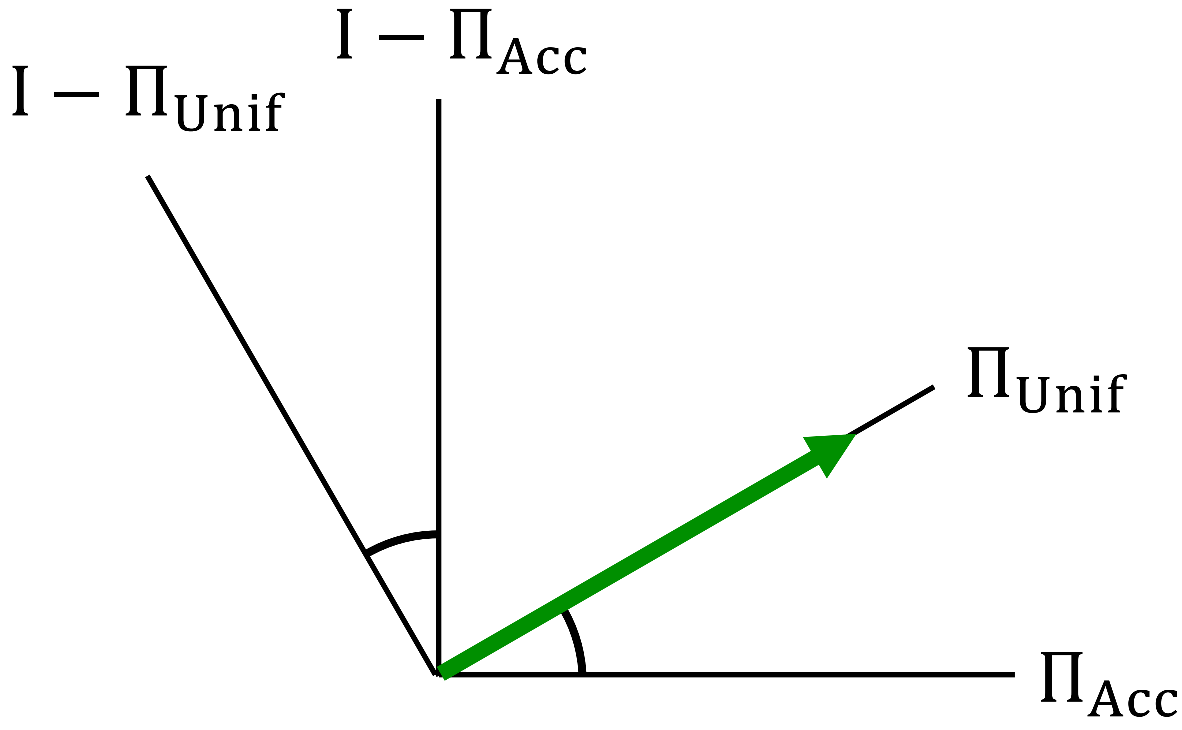


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

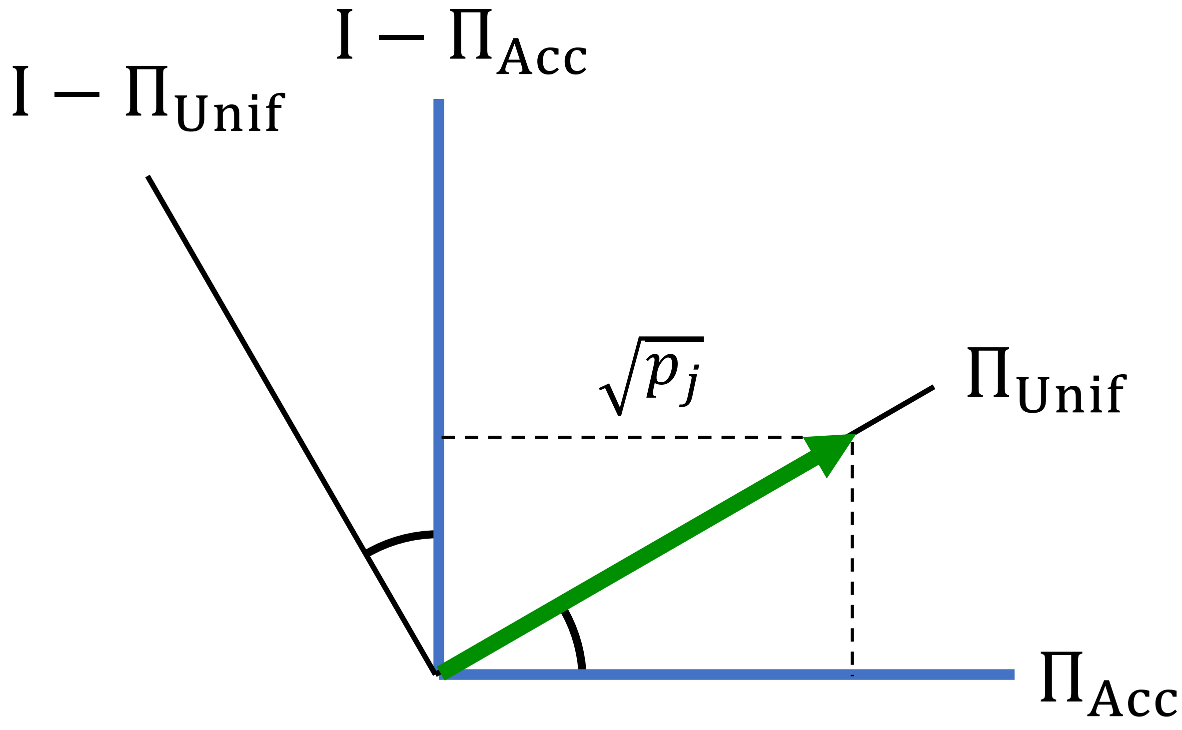


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

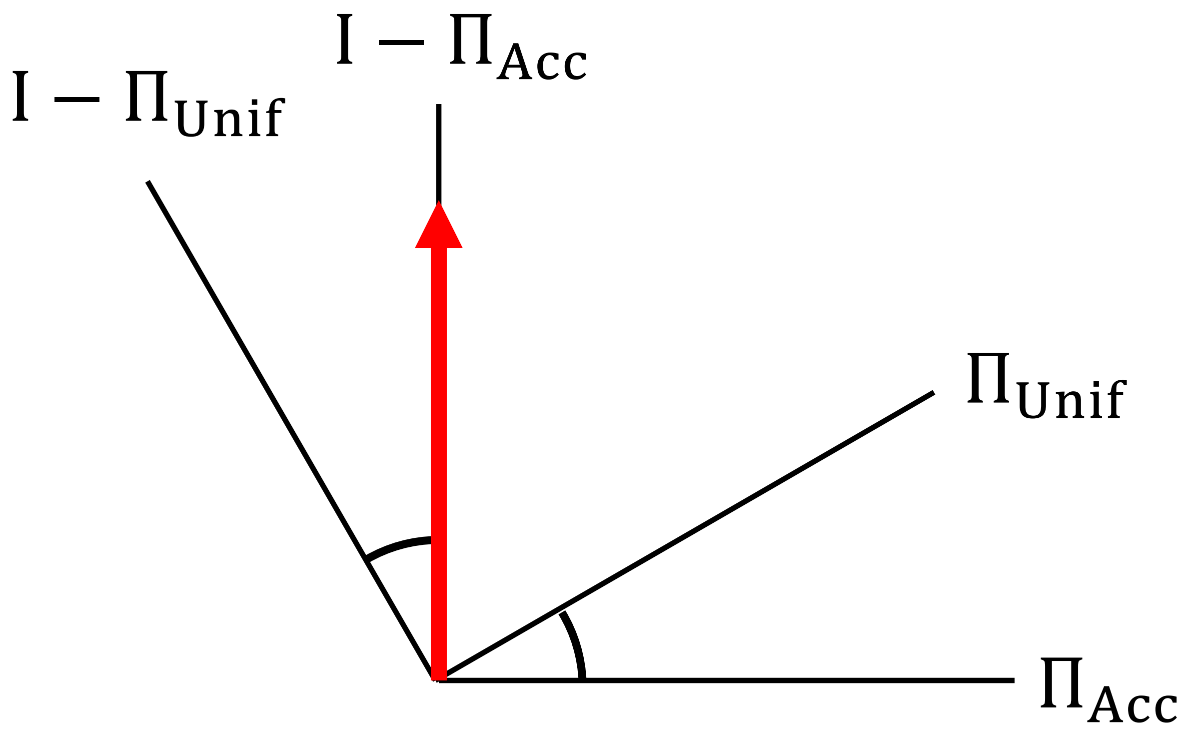


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

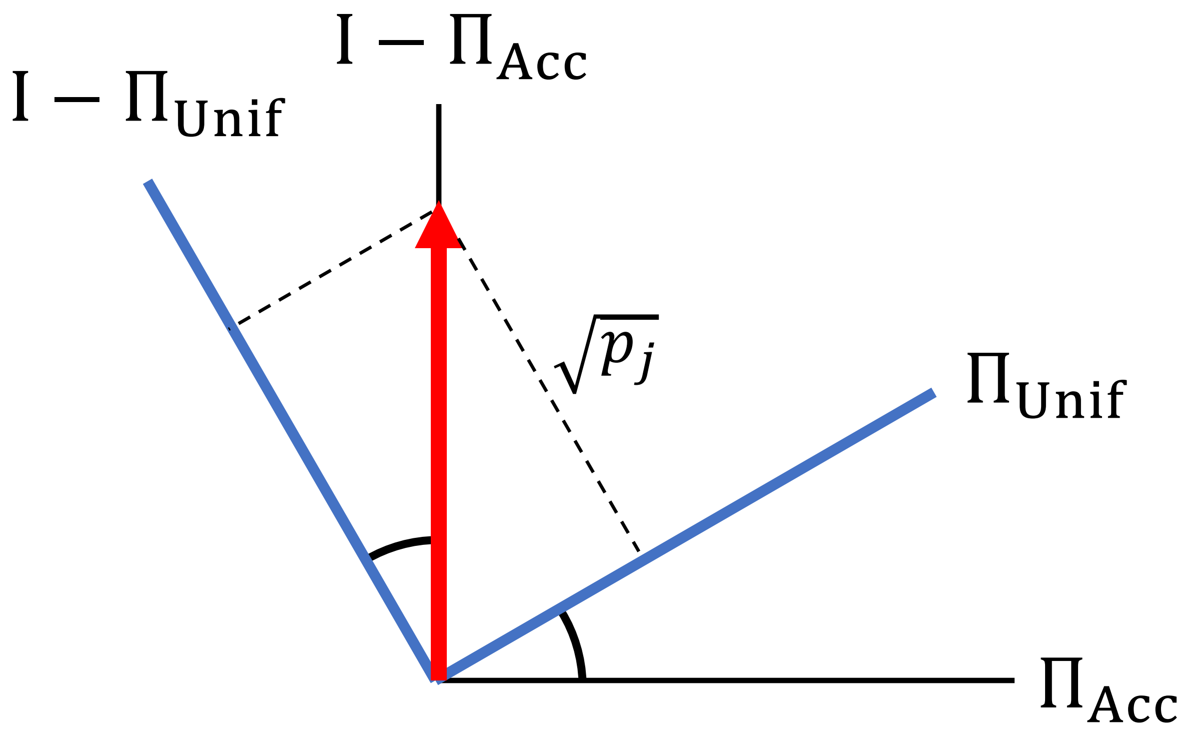


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

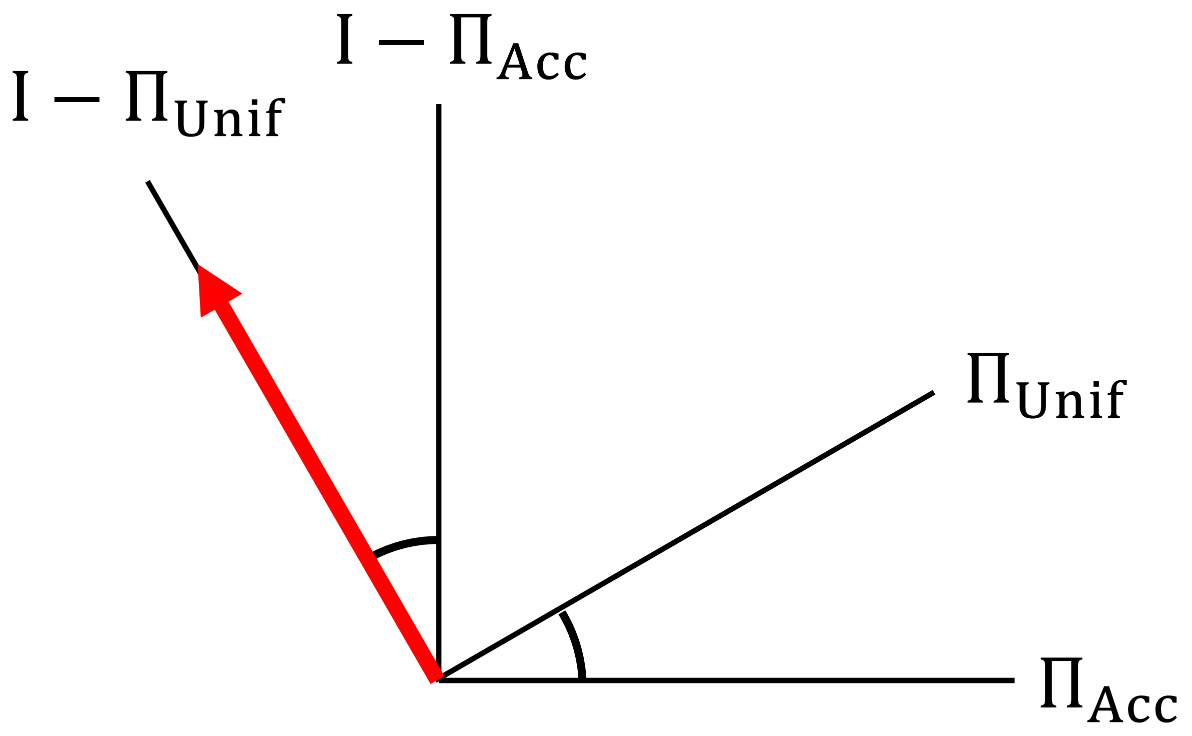


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

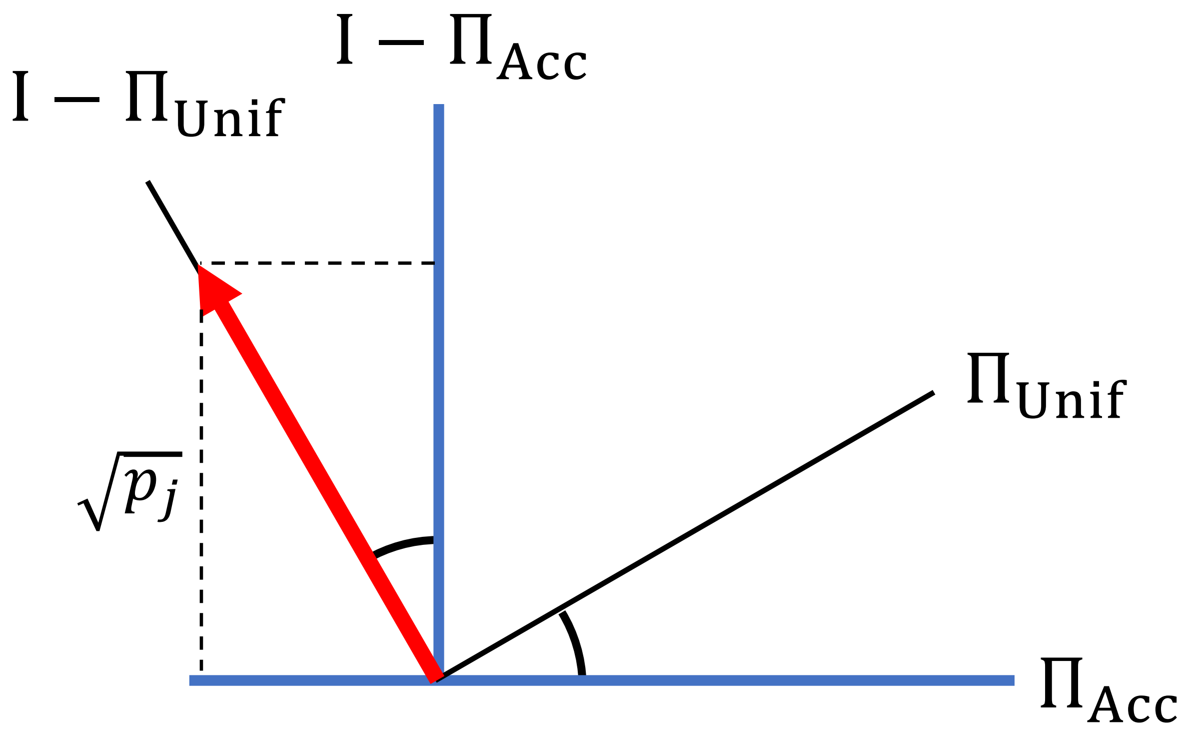


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

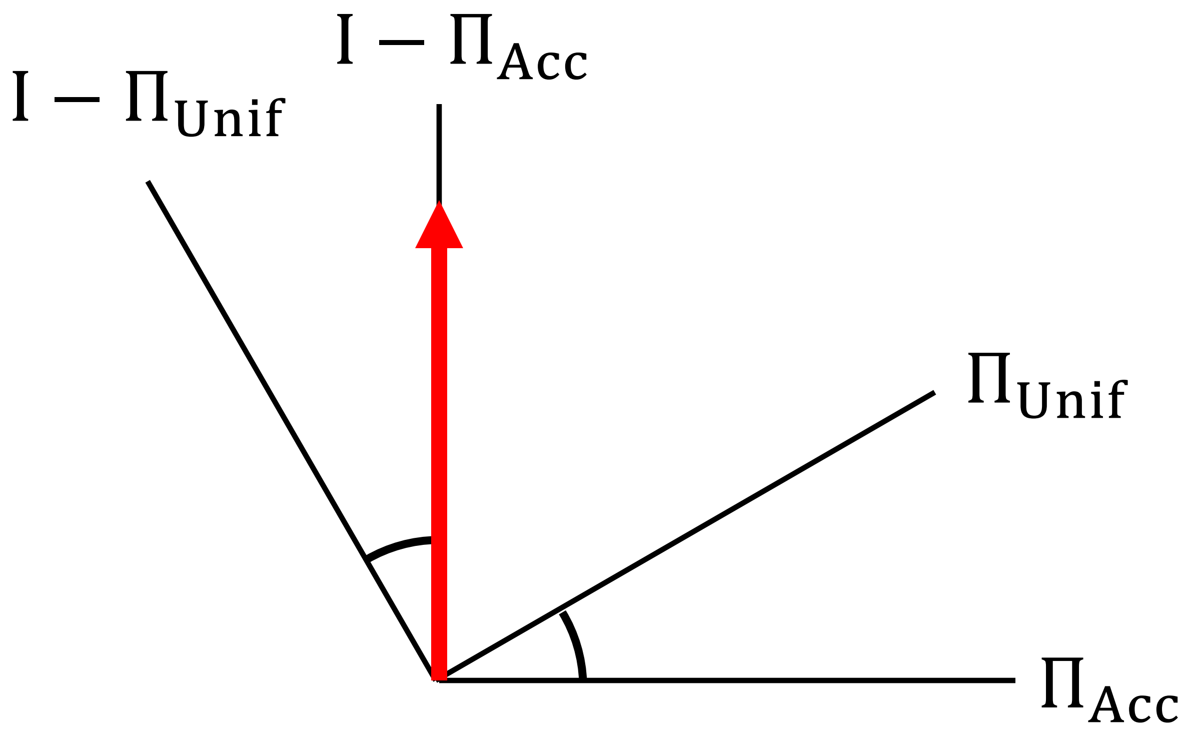


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

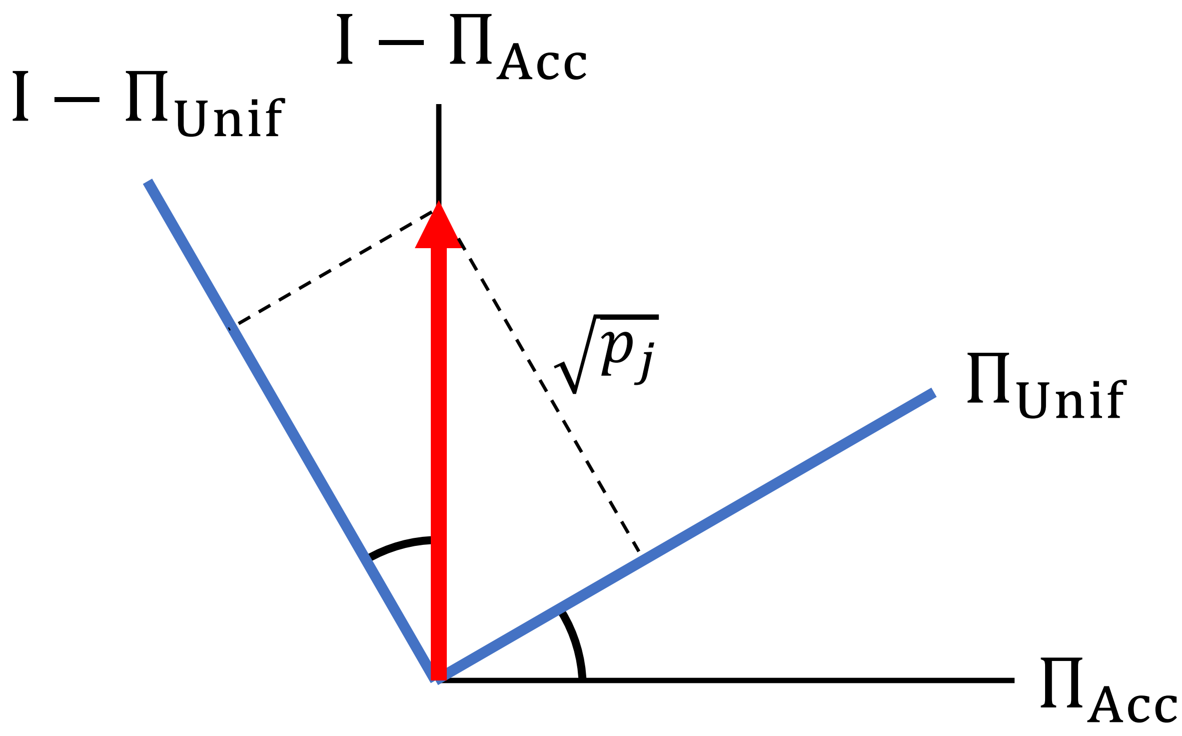


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

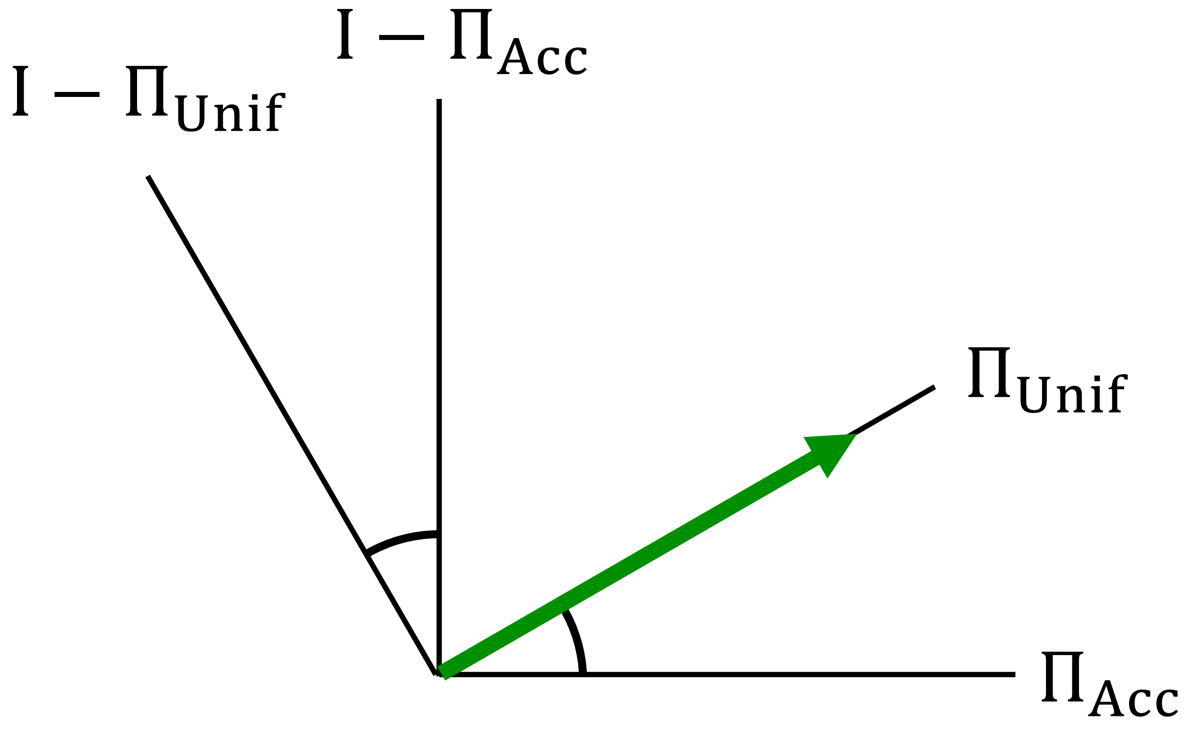


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

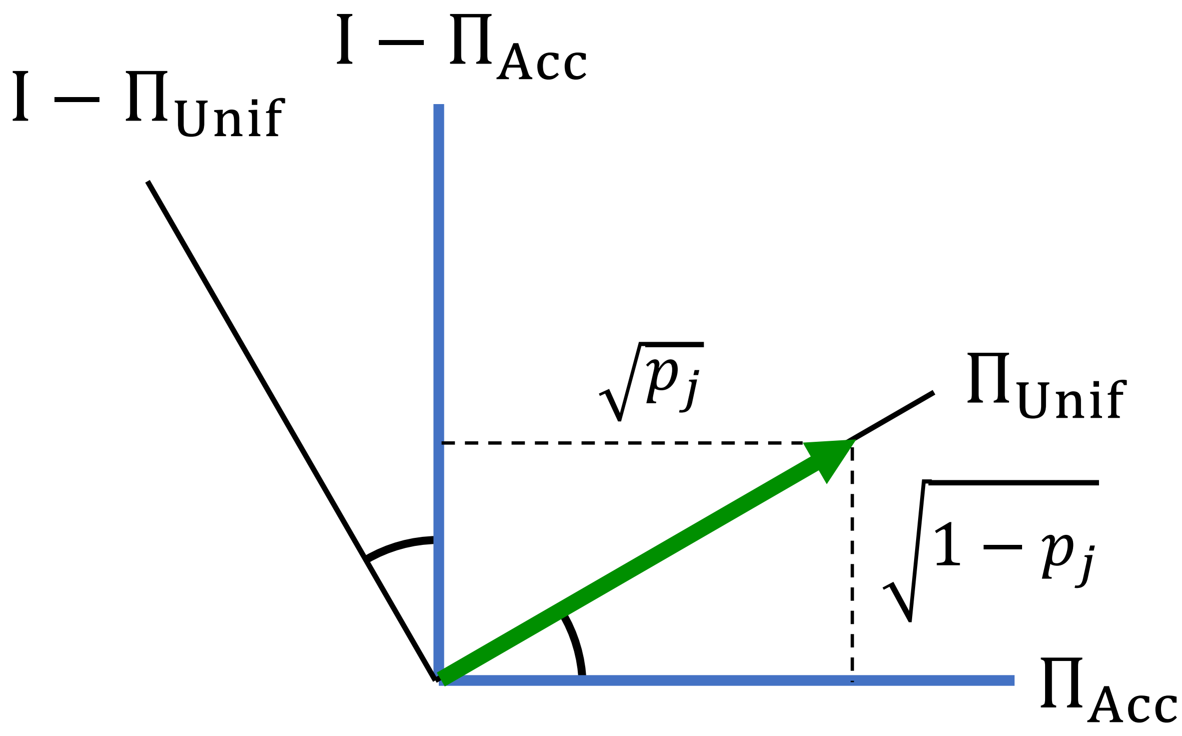


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

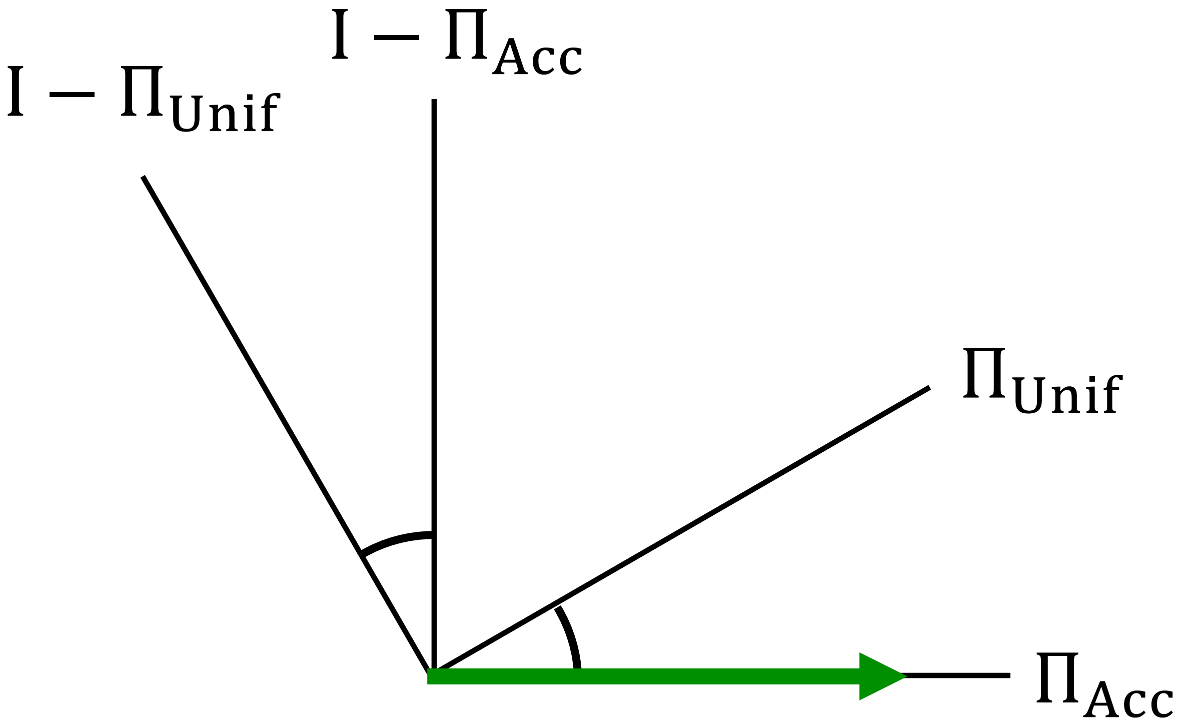


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

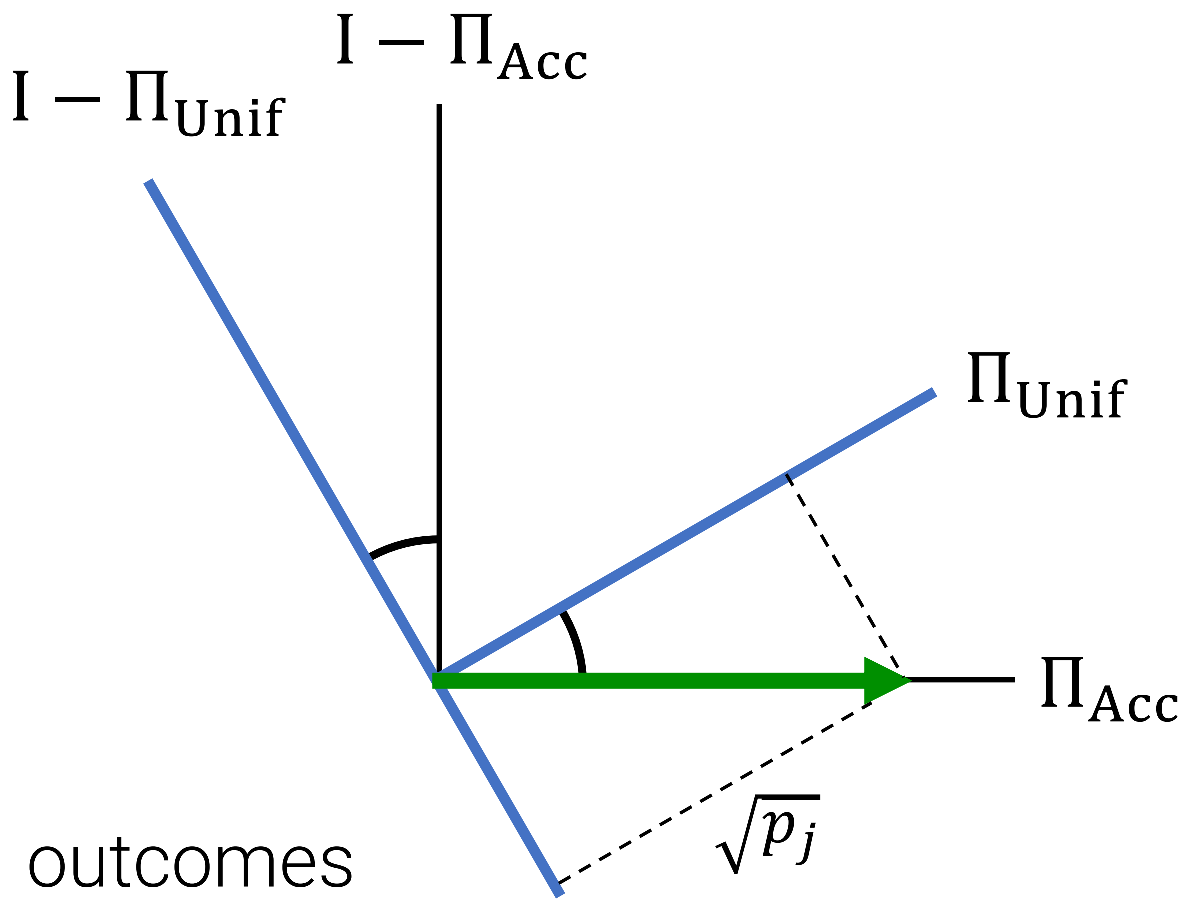


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

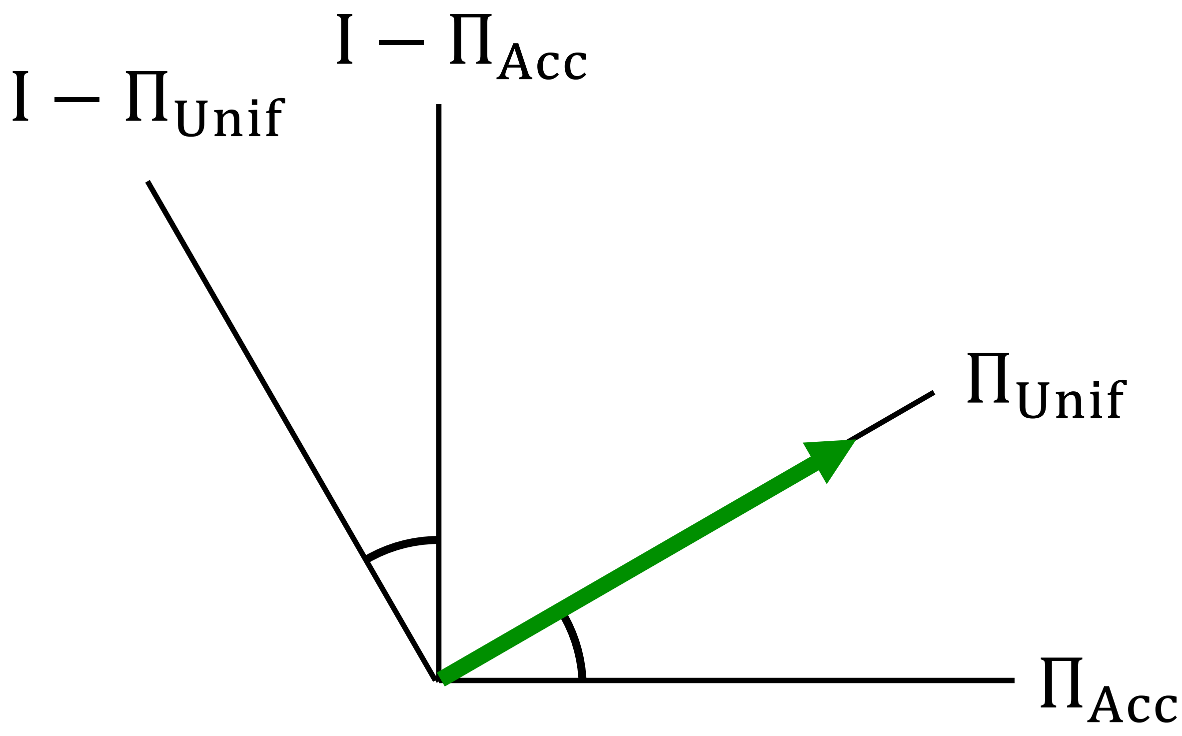


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

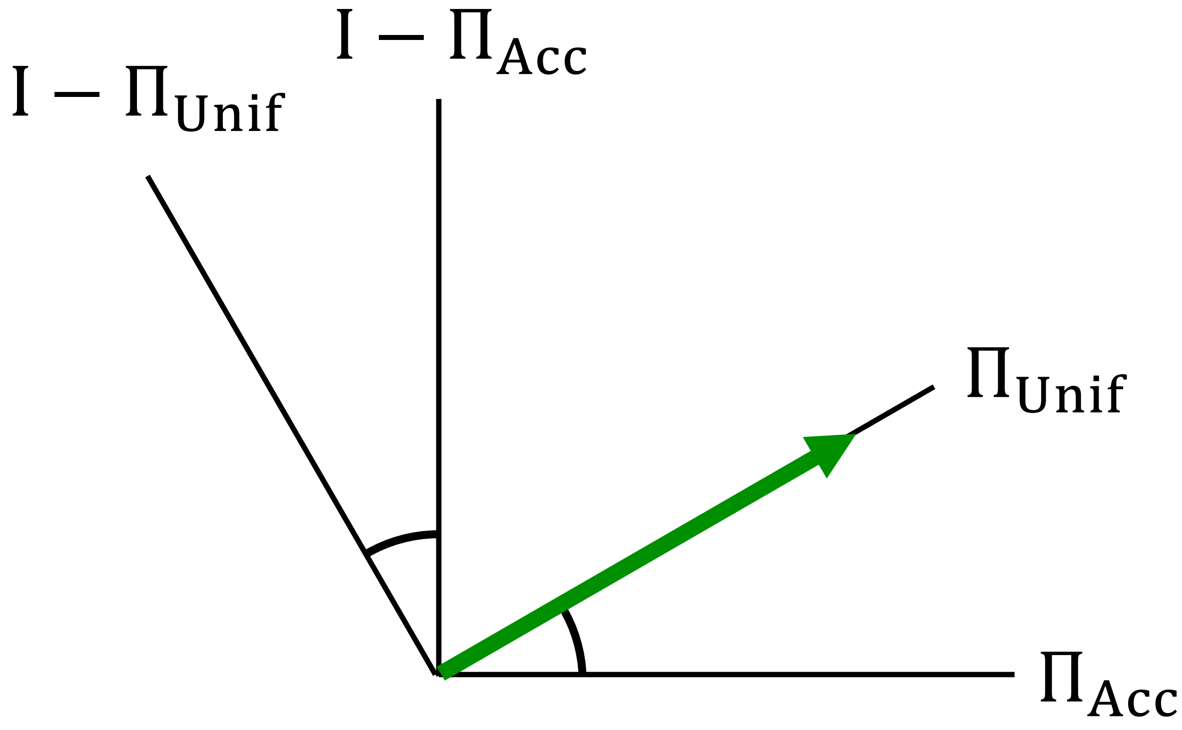


Suppose $|+\rangle_R|S\rangle$ lies in a 2-dim Jordan subspace S_j.

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

outcomes

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

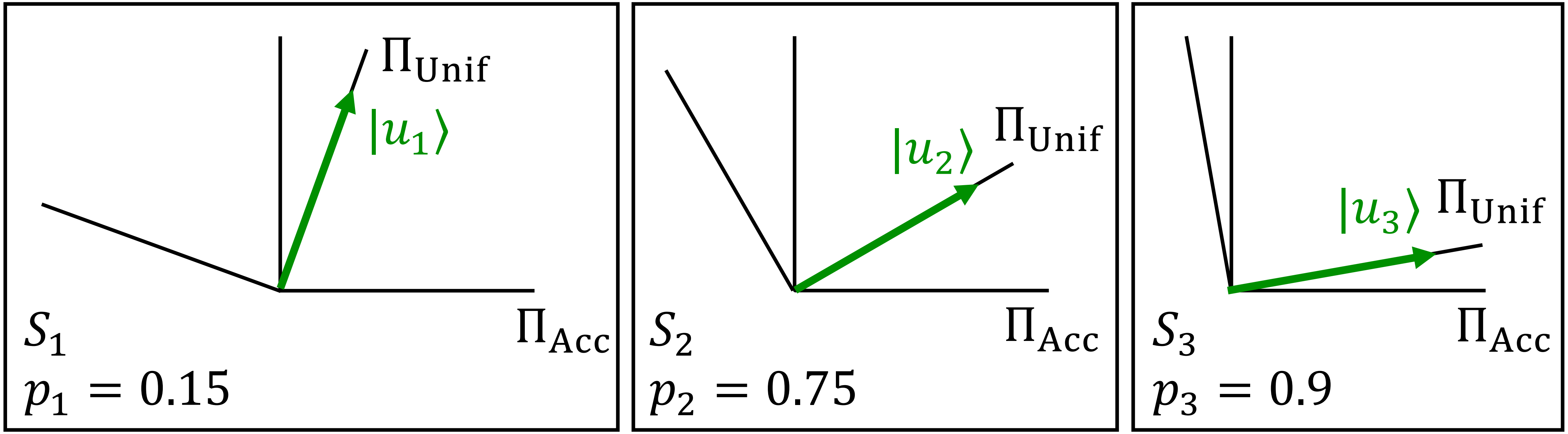


In this example, $b_i = b_{i+1}$ occurs 6 times out of 8, so we estimate $\approx 6/8$.

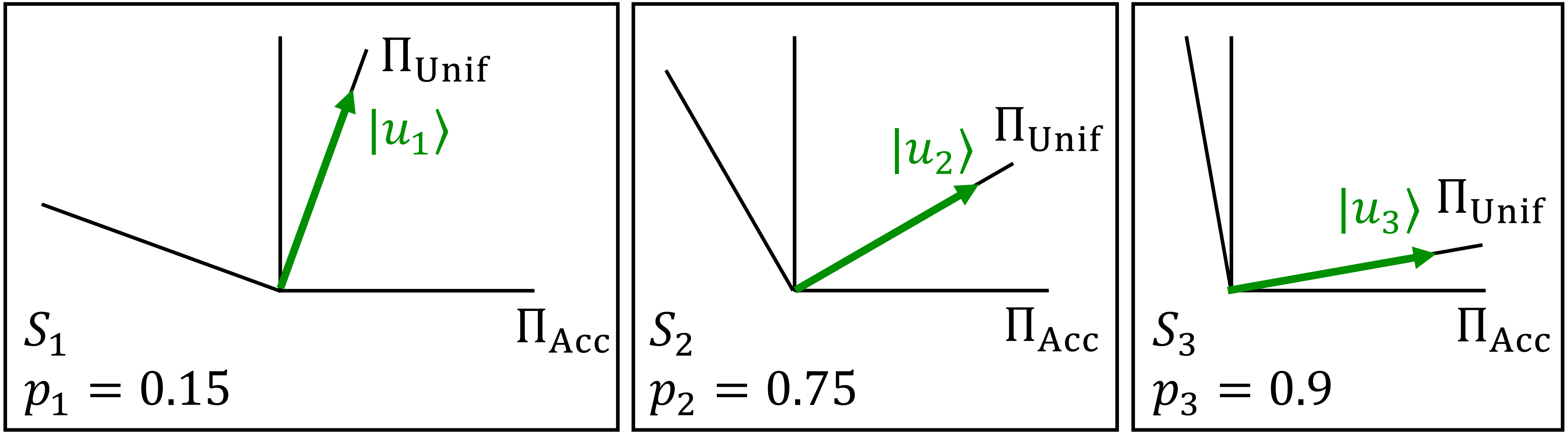
outcomes

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

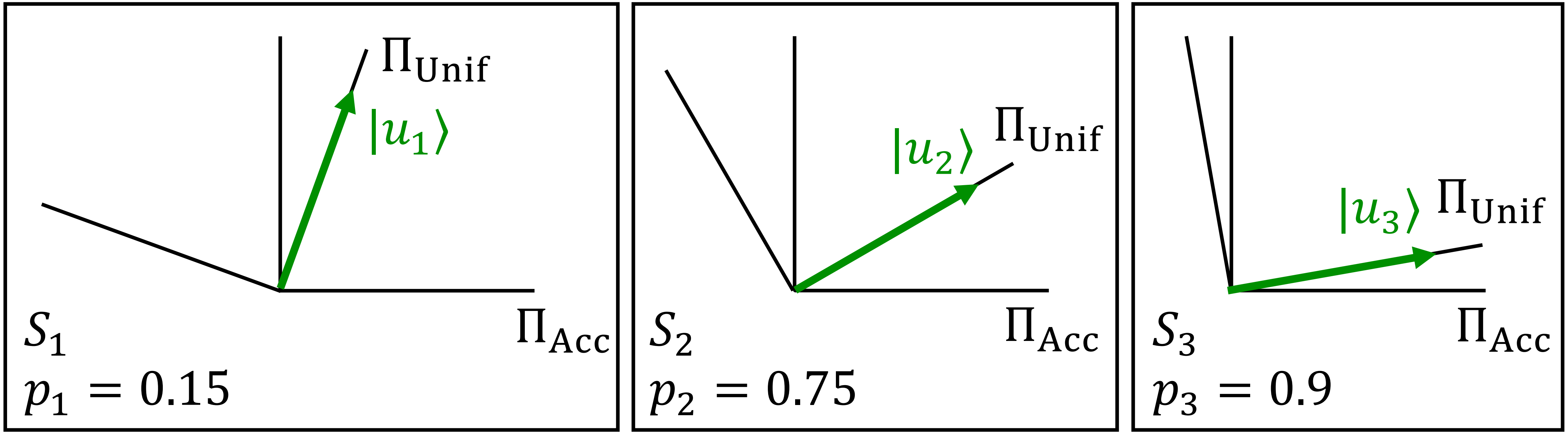


Suppose $|+\rangle_R \otimes |S\rangle = \alpha_1 |u_1\rangle + \alpha_2 |u_2\rangle + \alpha_3 |u_3\rangle$.

success prob p_1

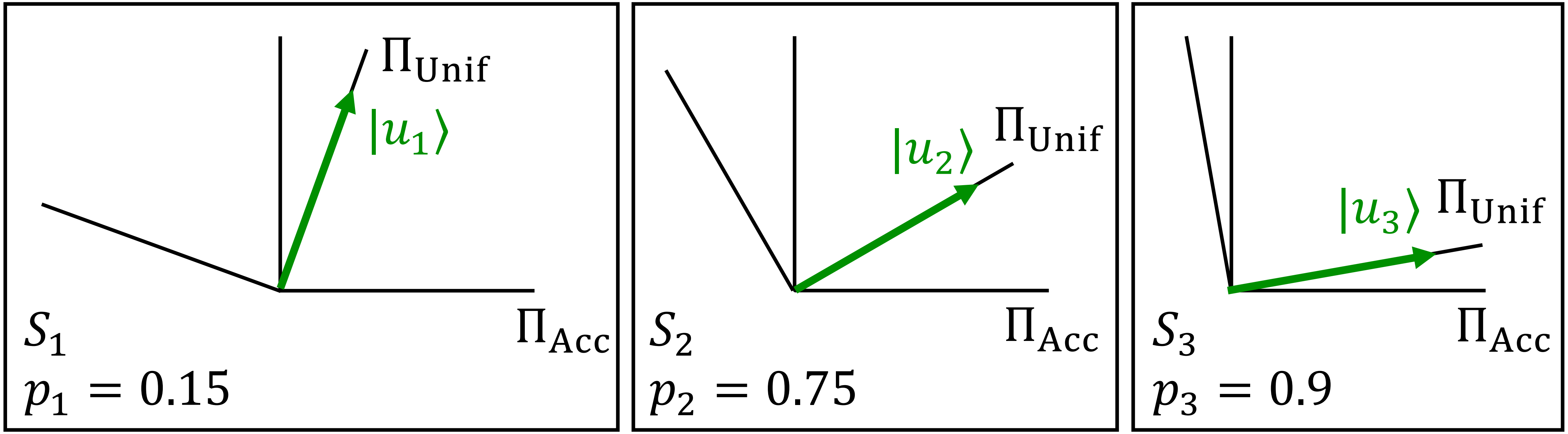
success prob p_2

success prob p_3



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

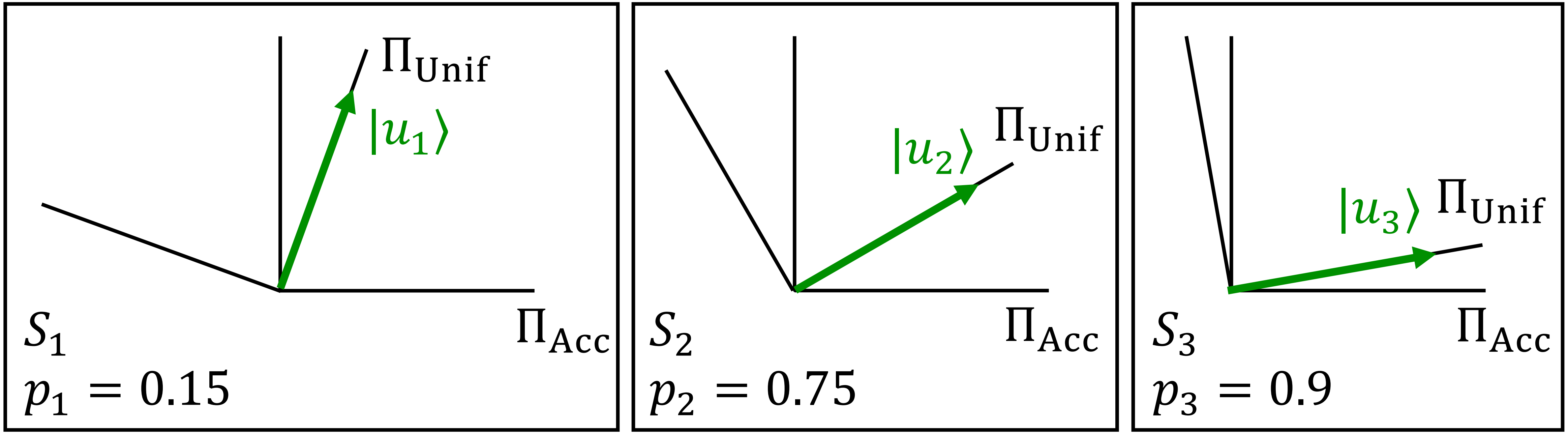
Key fact: Alternating measurement outcomes distributed as though $|+\rangle_R \otimes |S\rangle$ were contained in S_j with prob $|\alpha_j|^2$.



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.



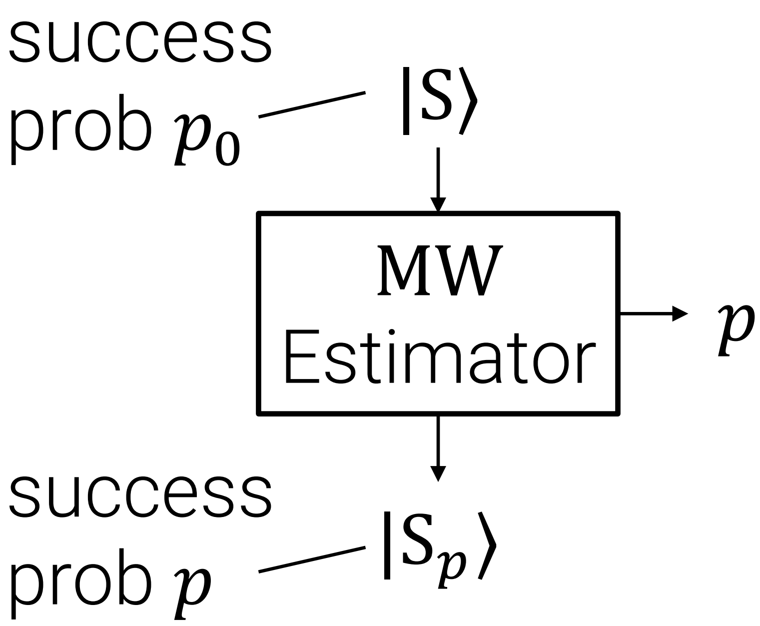
Suppose $|+\rangle_R \otimes |S\rangle = \alpha_1 |u_1\rangle + \alpha_2 |u_2\rangle + \alpha_3 |u_3\rangle$.

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

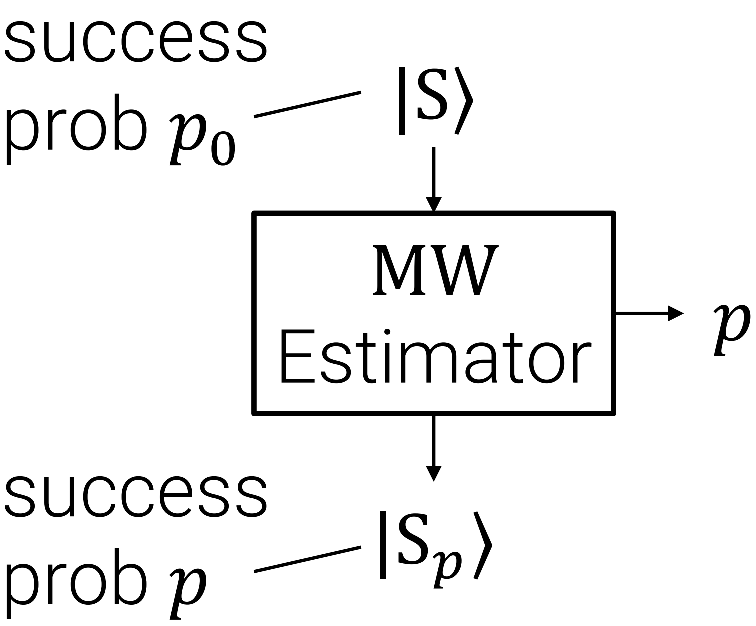
- w/ prob $\approx |\alpha_j|^2$ obtain estimate $\approx p_j$ and leftover state $\approx |u_j\rangle$

success
prob p_0 $|S\rangle$

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



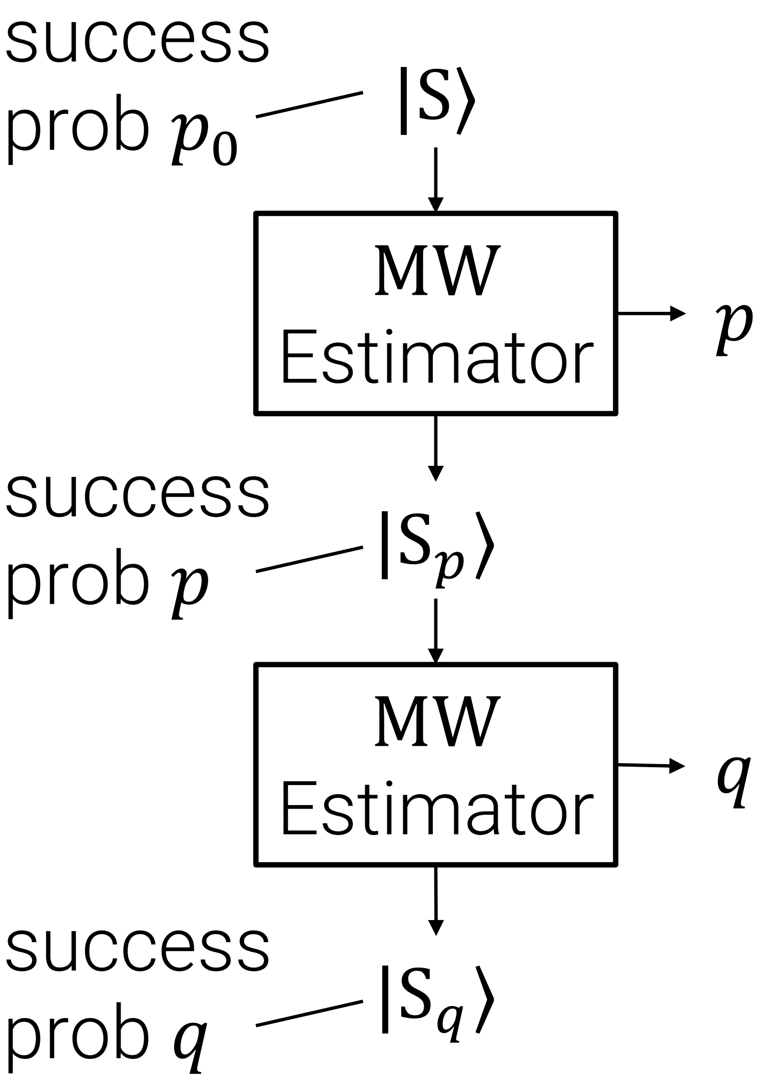
We'll need two key properties about the MW estimator.



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

Key Properties

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

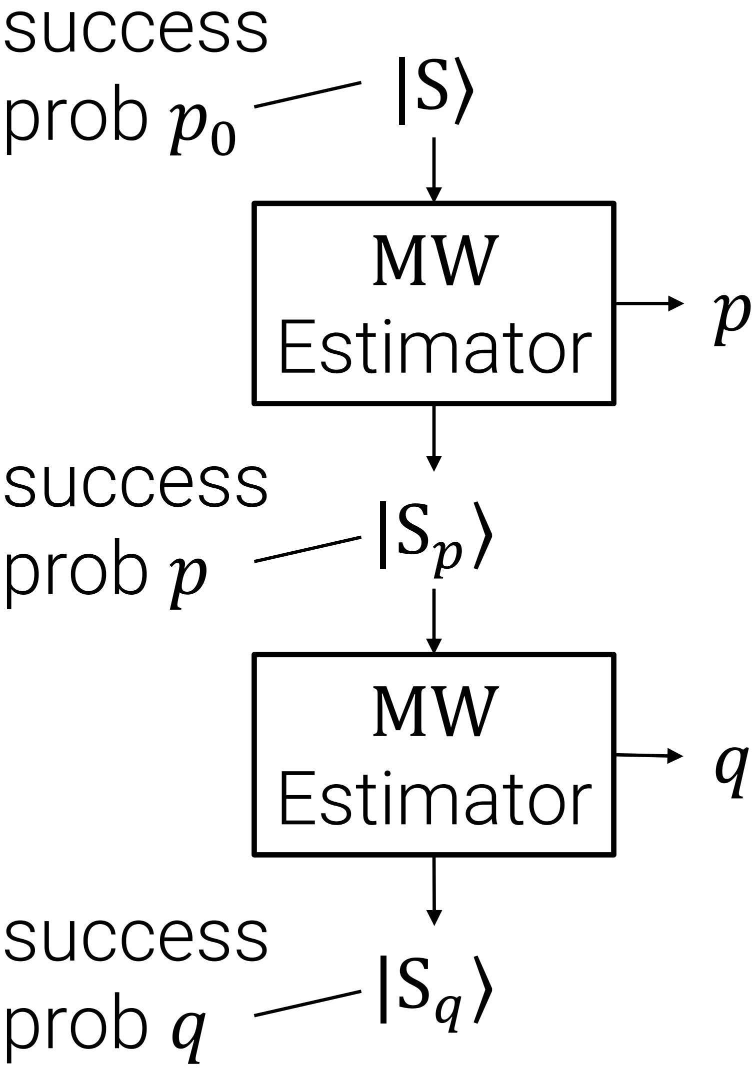


We'll need two key properties about the $\mathbf{MW}$ estimator.

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.



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. Formally, 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

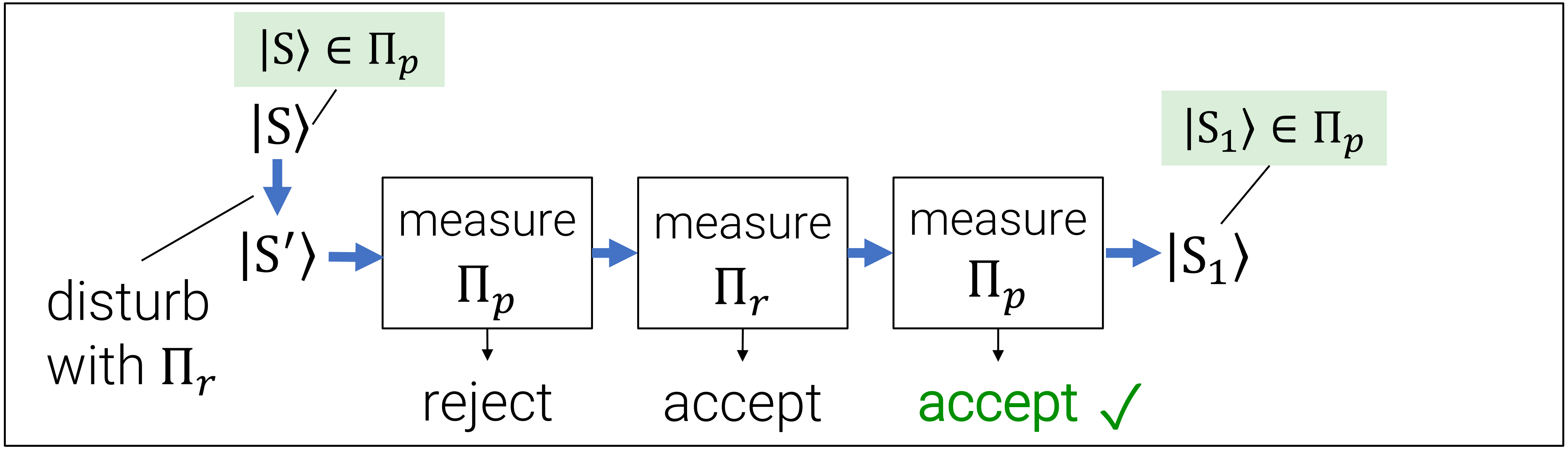
1) $\mathbb{E}[p] = p_0$

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

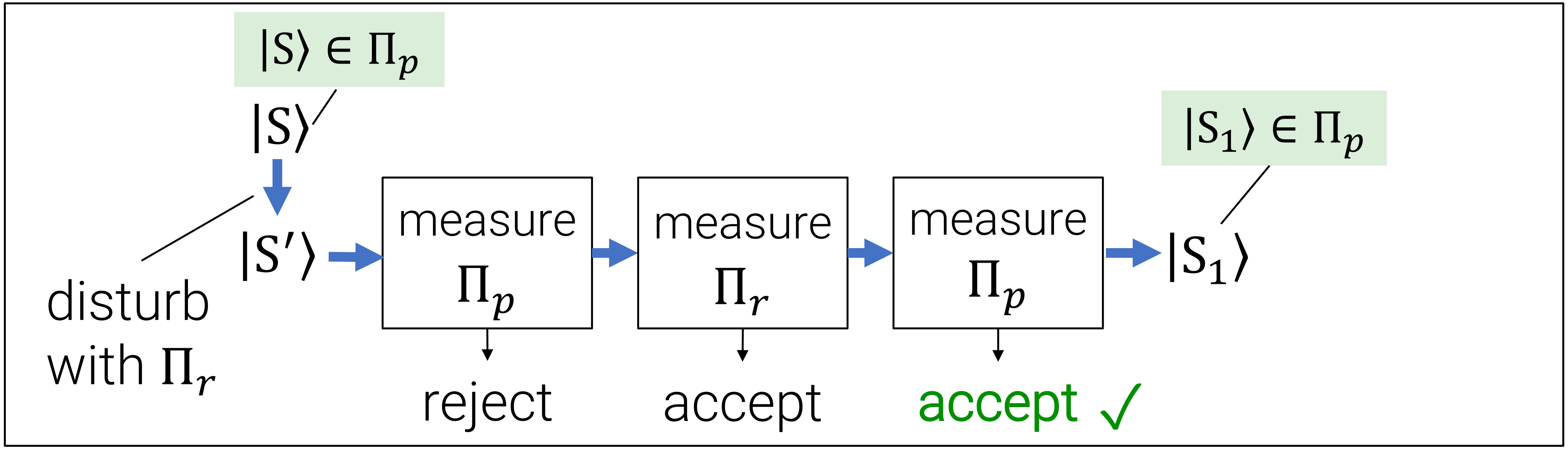
$$\mathbf{Pr}[|p - q| \leq \varepsilon] \geq 1 - \delta$$

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

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



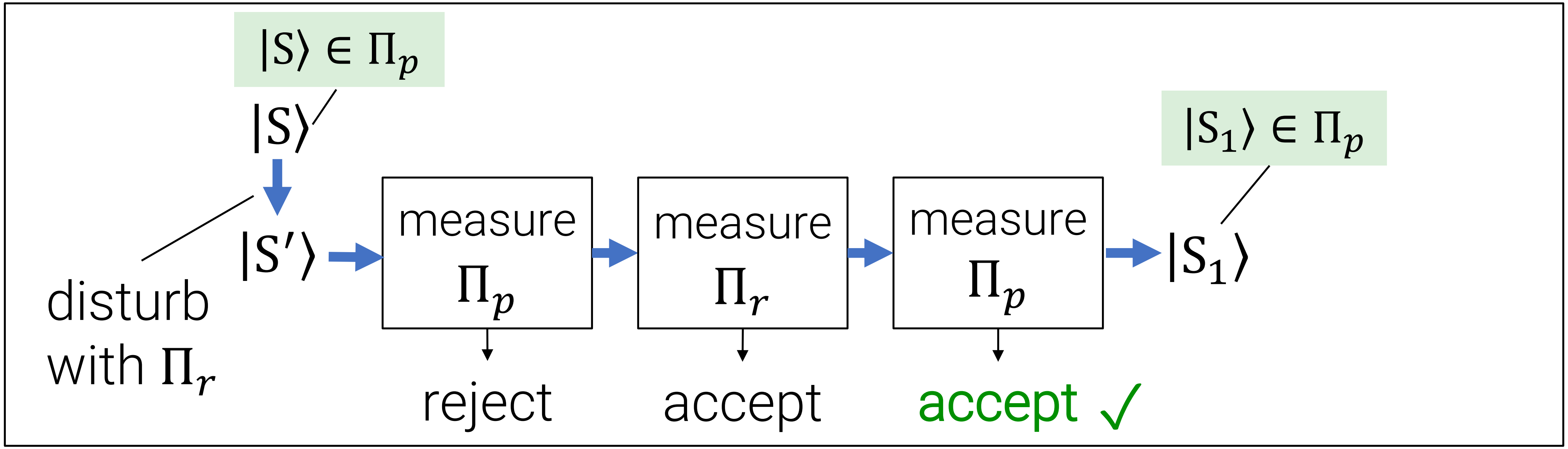
Recall: in our high-level sketch, we assumed we could *exactly* measure Π_p, i.e., whether success prob $\geq p$.



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:

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

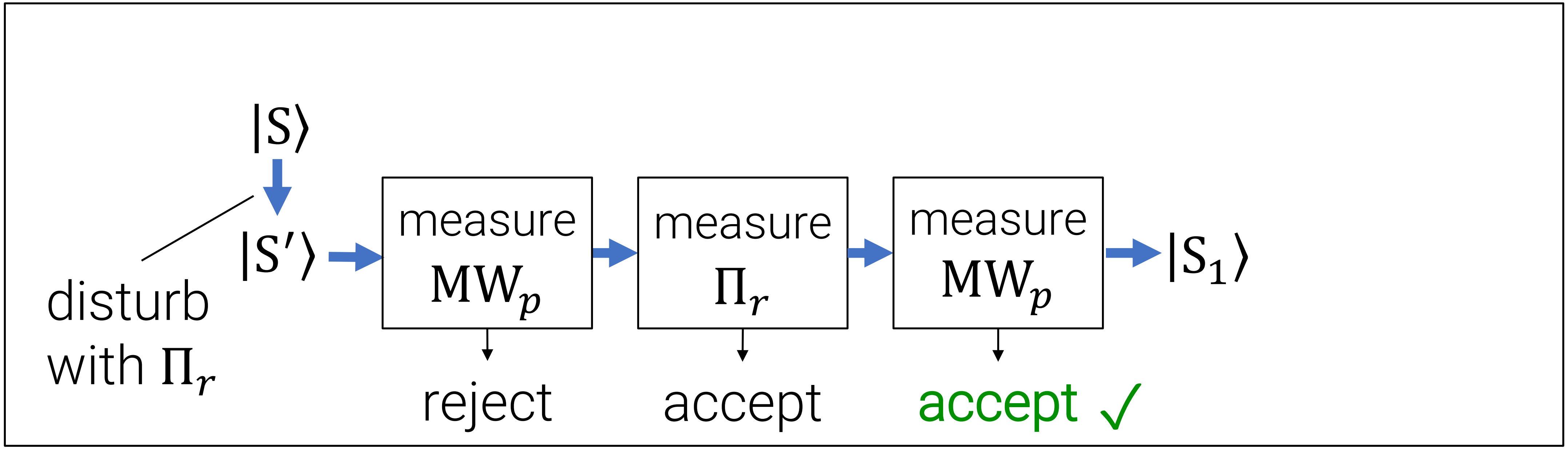


Recall: in our high-level sketch, we assumed we could *exactly* measure Π_p, i.e., whether success prob $\geq p$.

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

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

Idea: run Marriott-Watrous on Marriott-Watrous!

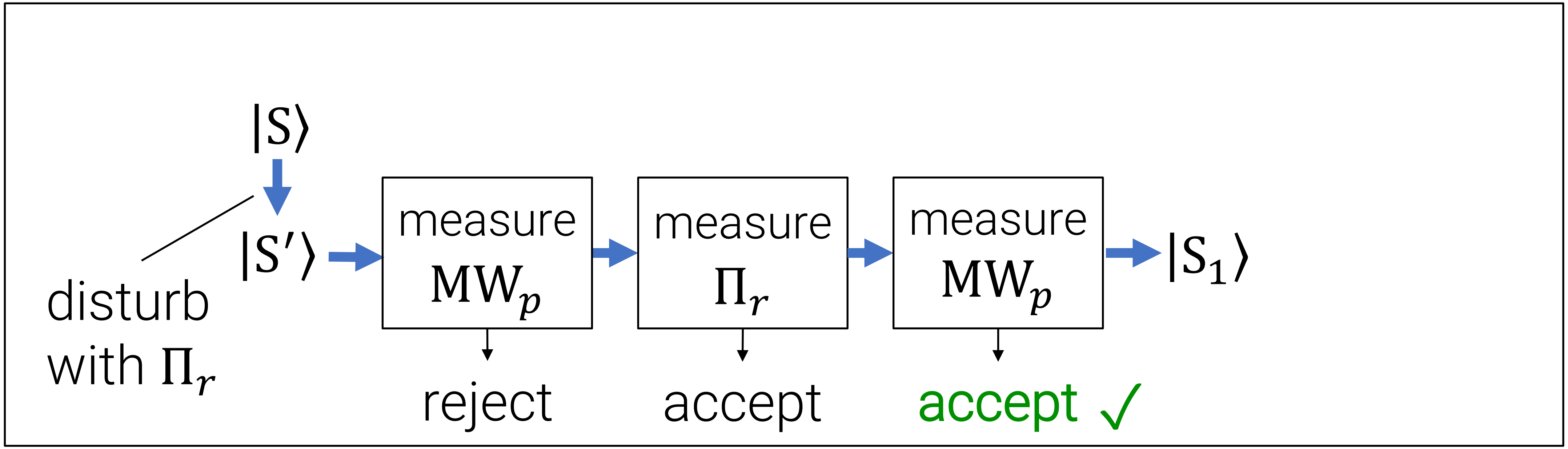


Recall: in our high-level sketch, we assumed we could *exactly* measure Π_p, i.e., whether success prob $\geq p$.

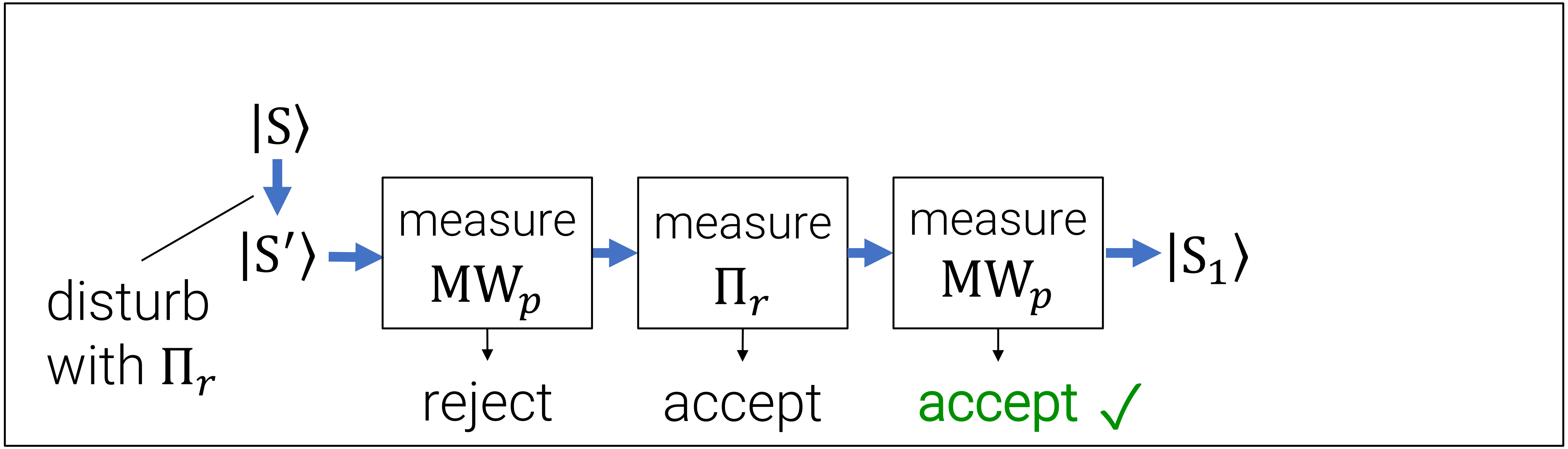
We don't know how to measure Π_p, but we can approximate it:

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

Idea: run Marriott-Watrous on Marriott-Watrous!

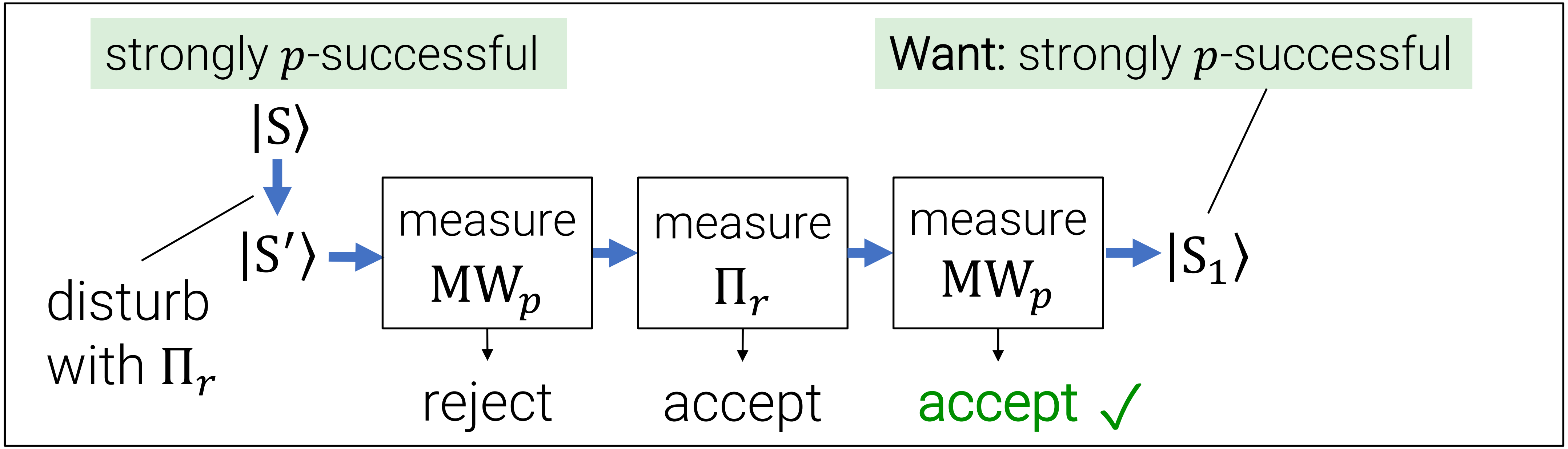


Subtle point: Just restoring "success probability" is not enough!



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

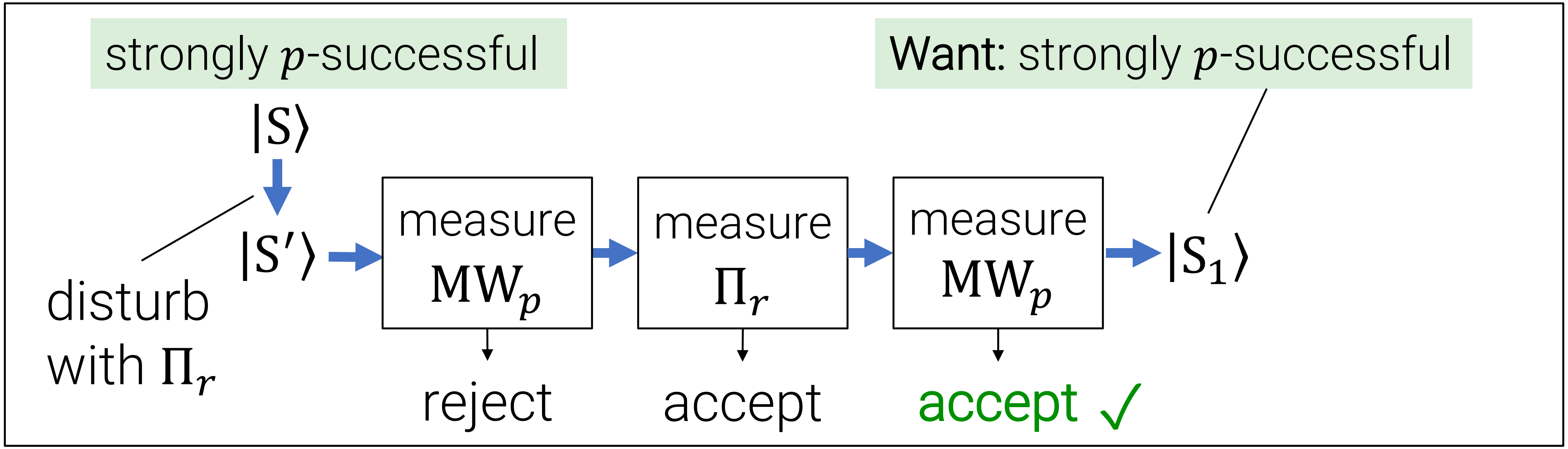
Definition: $|S\rangle$ is *strongly $\boldsymbol{p}$-successful* if it is concentrated on $(\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}})$-Jordan subspaces with eigenvalue $\geq p$



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

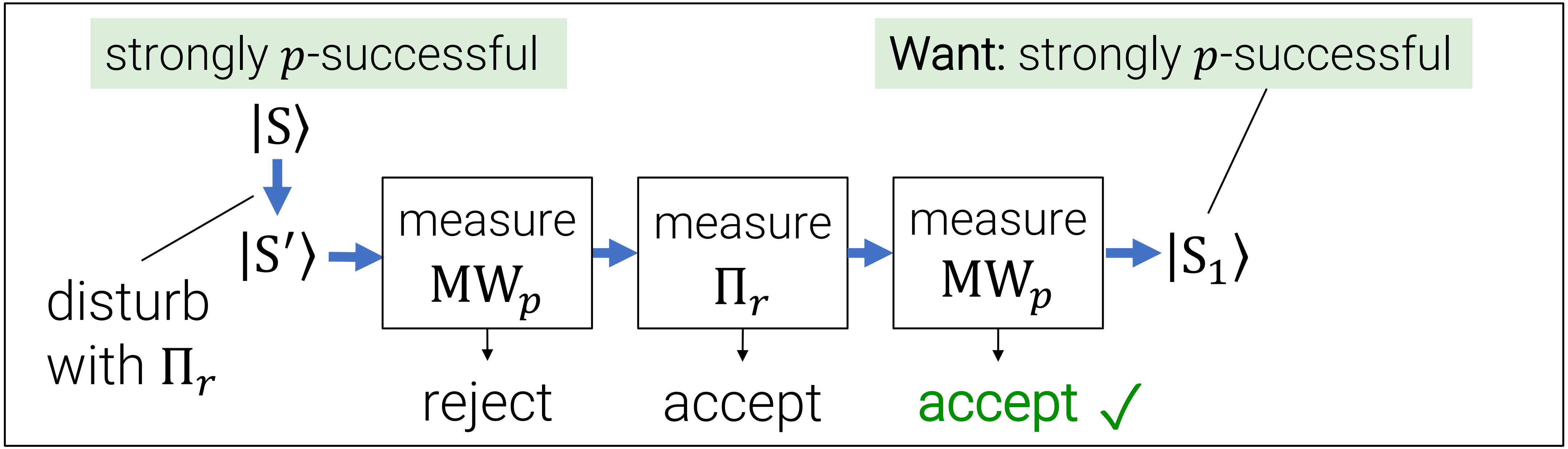
Definition: $|S\rangle$ is *strongly p-successful* if it is concentrated on $(\Pi_{\mathrm{Acc}}, \Pi_{\mathrm{Unif}})$-Jordan subspaces with eigenvalue $\geq p$

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:

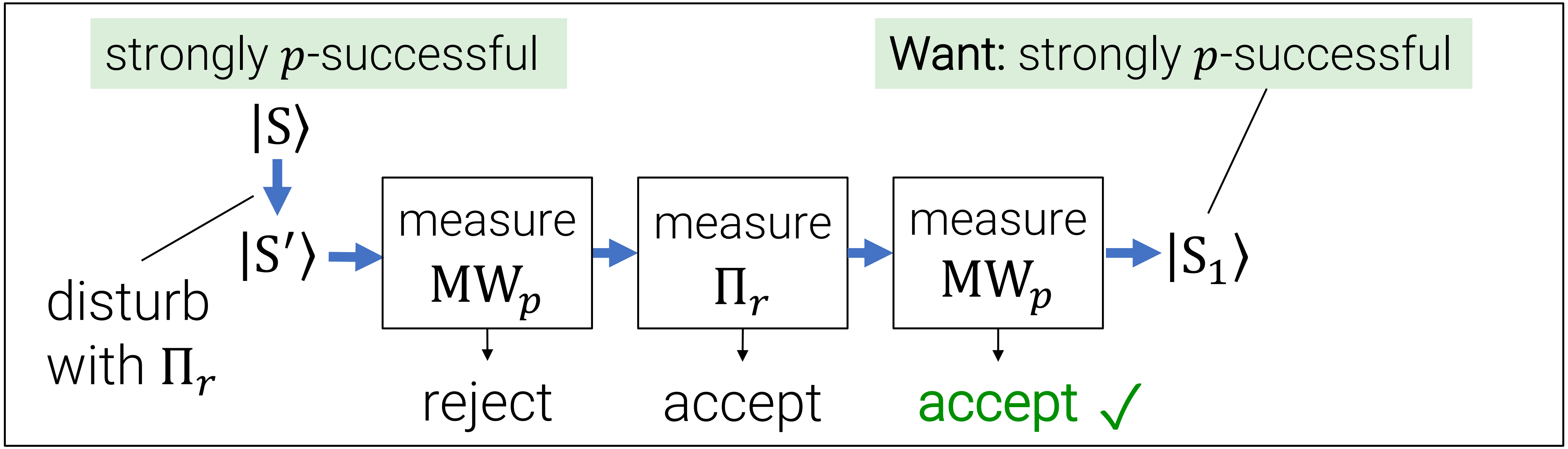
Our proof that this procedure terminates requires the measurements to be projective, but MW_p is not!



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!

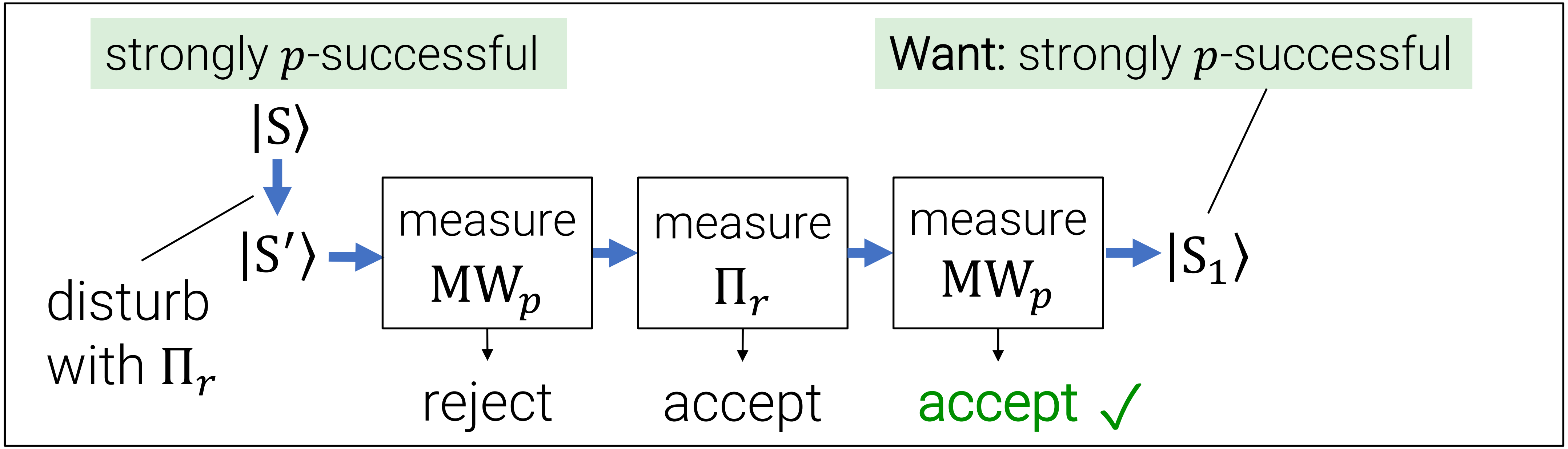
(running it twice may give different outcomes)



This seems promising, but we have a problem:

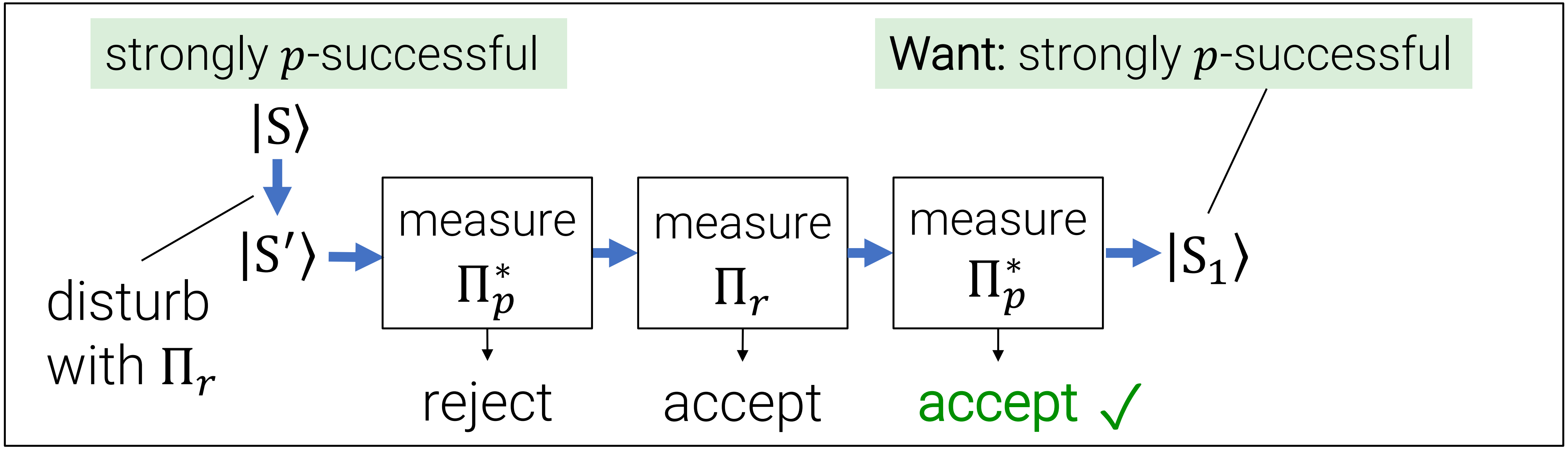
Our proof that this procedure terminates requires the measurements to be projective, but $\mathbf{MW}_p$ is not!

Easy(?) fix: Make $\mathbf{MW}_p$ projective by expanding the Hilbert space.



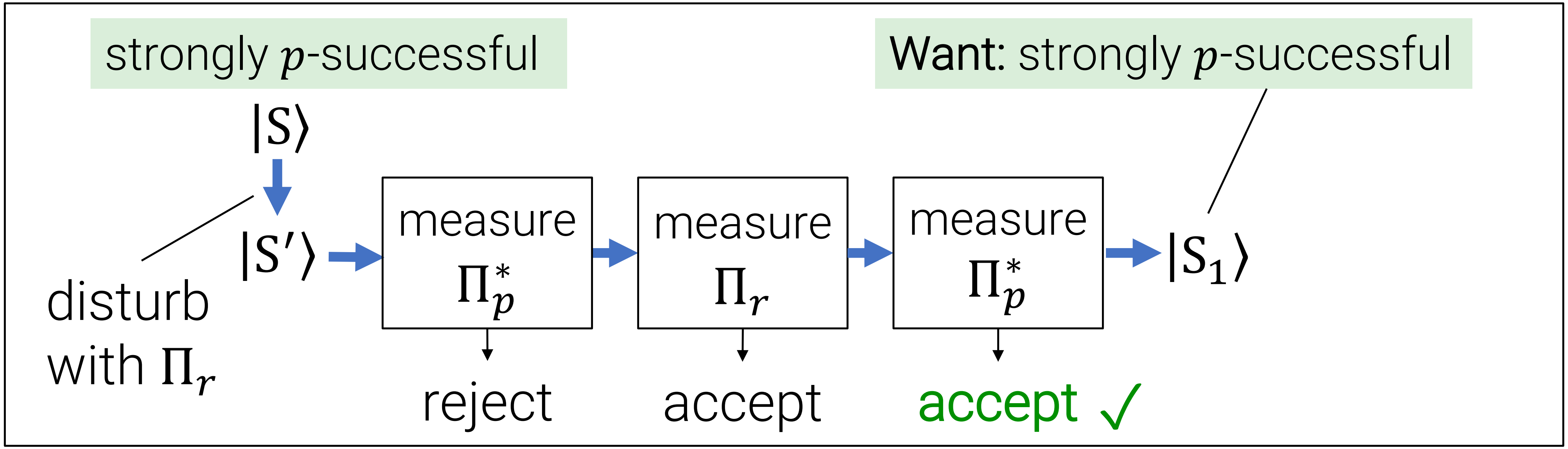
Measuring $|S'\rangle$ with MW_p can be implemented as a projective measurement of some Π_p^* on $|S'\rangle_A|0\rangle_W \in A \otimes W$.

adversary state register workspace/ancilla



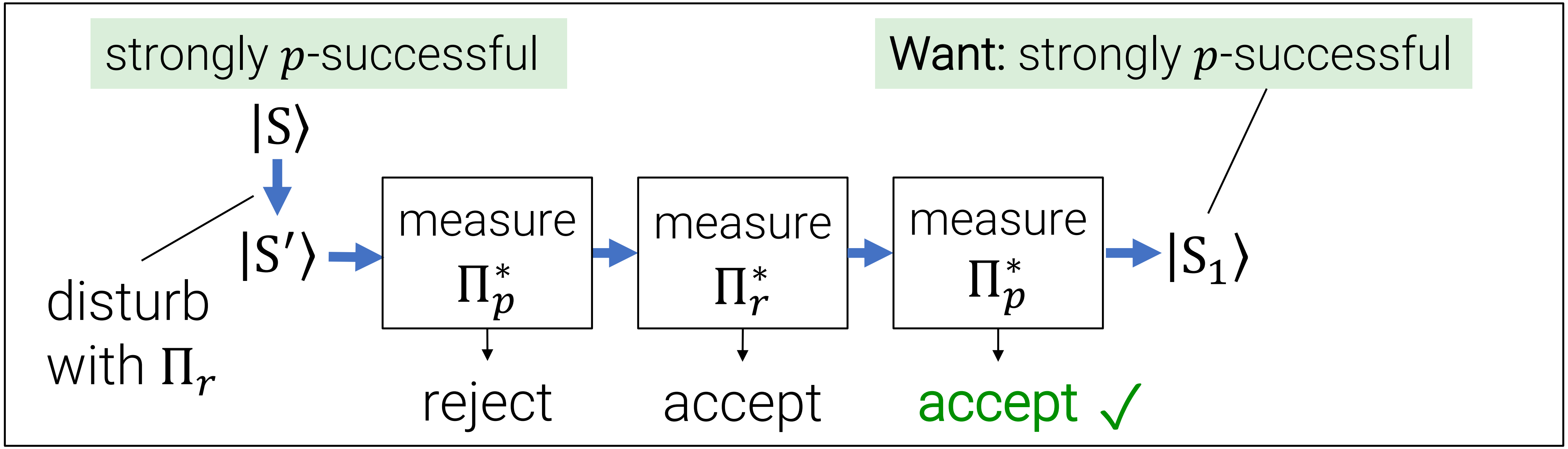
Measuring $|S'\rangle$ with MW_p can be implemented as a projective measurement of some Π_p^* on $|S'\rangle_A|0\rangle_W \in A \otimes W$.

adversary state register workspace/ancilla

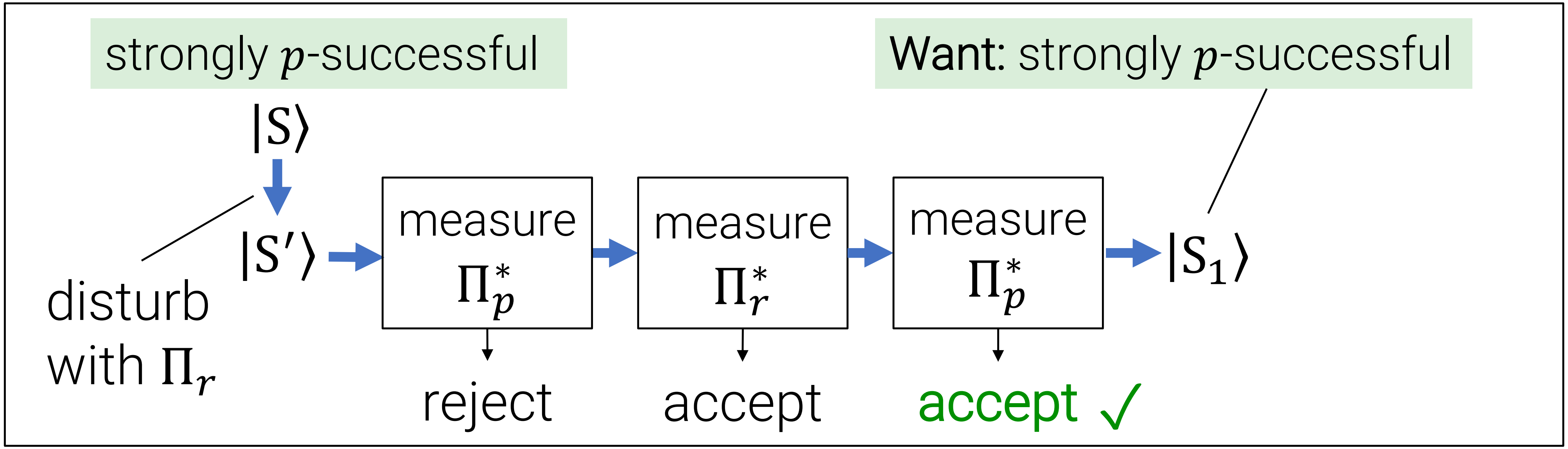


Measuring $|S'\rangle$ with MW_p can be implemented as a projective measurement of some Π_p^* on $|S'\rangle_A|0\rangle_W \in A \otimes W$.

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

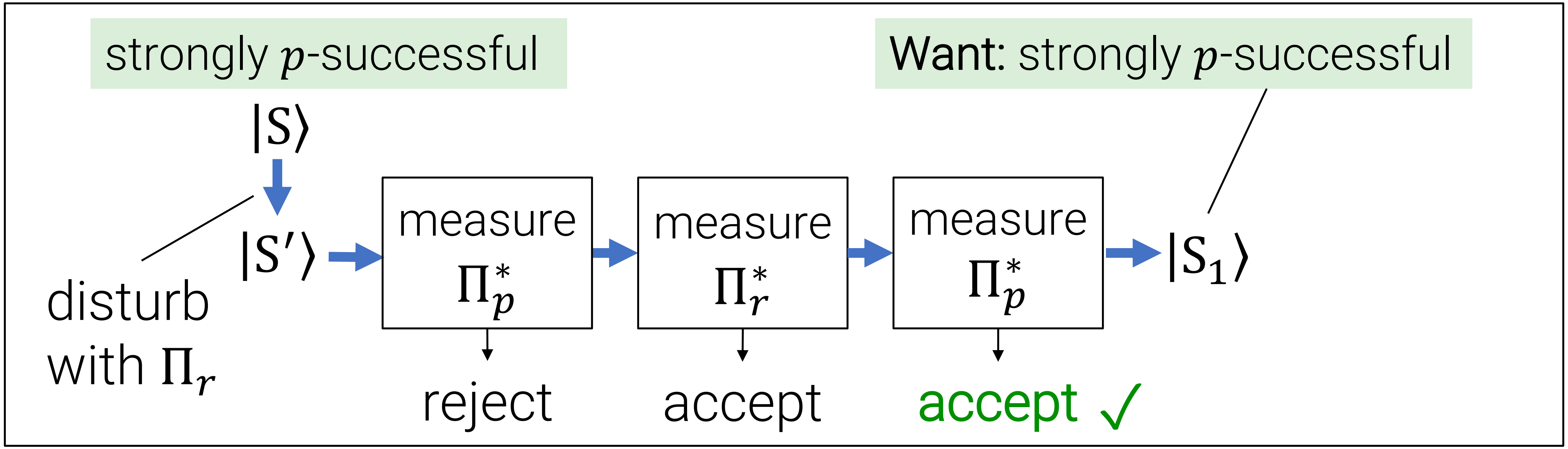


Our solution is re-define Π_r to $\Pi_r^* \coloneqq \Pi_r \otimes |0\rangle\langle 0|_W$, so that each measurement of Π_r^* attempts to "reset" the W to $|0\rangle_W$.



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 $|0\rangle_W$.



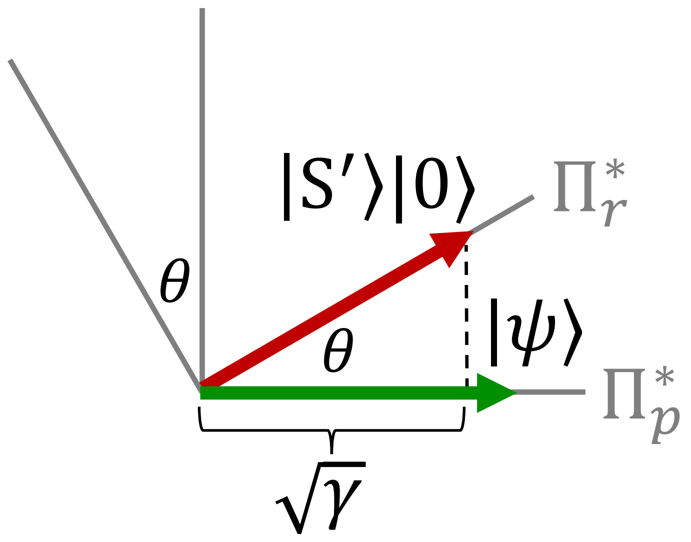
This is essentially the full repair procedure!

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 $|0\rangle_W$.

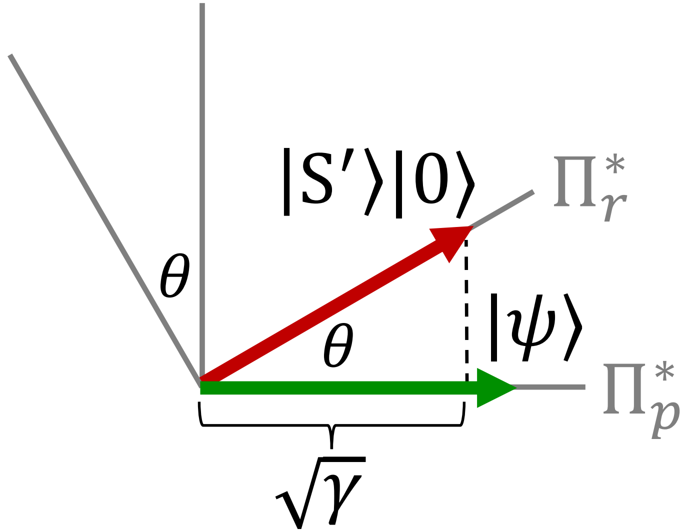
Not obvious: ***why*** does this choice of Π_r^* make repair work?

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

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

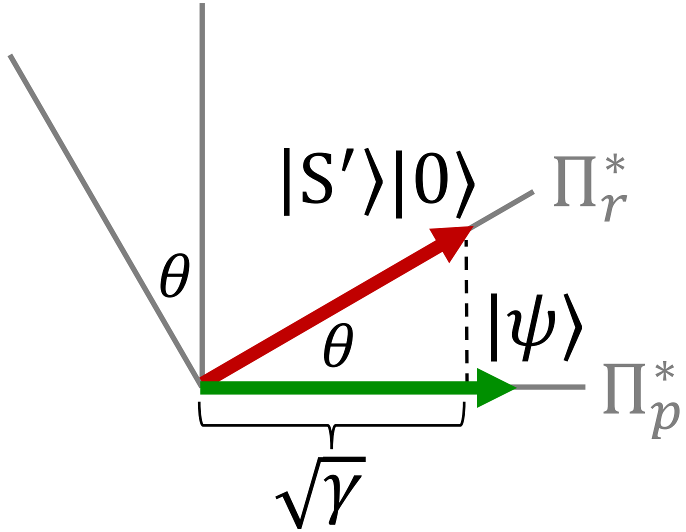


In a nutshell: $\Pi_r^* \coloneqq \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can analyze the Jordan subspaces for Π_r^*, Π_p^* in terms of the $\mathbf{MW}$ procedure.



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

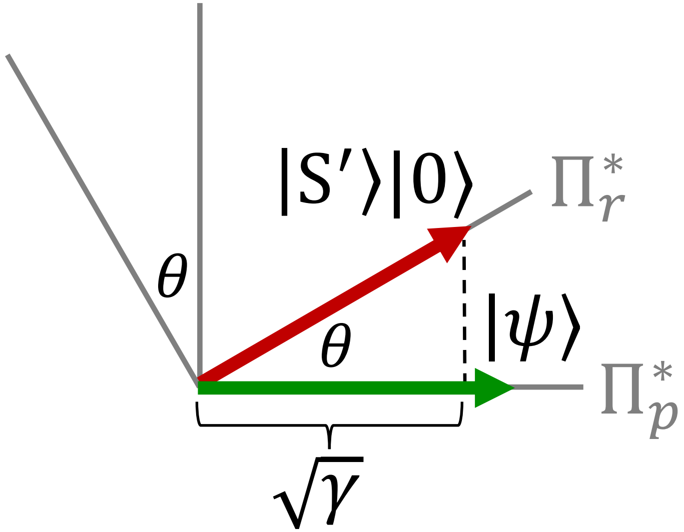
In a nutshell: $\Pi_r^* \coloneqq \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can analyze the Jordan subspaces for Π_r^*, Π_p^* in terms of the $\mathbf{MW}$ procedure.



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 corresponds to a strongly $(p - \varepsilon)$-successful adversary.

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



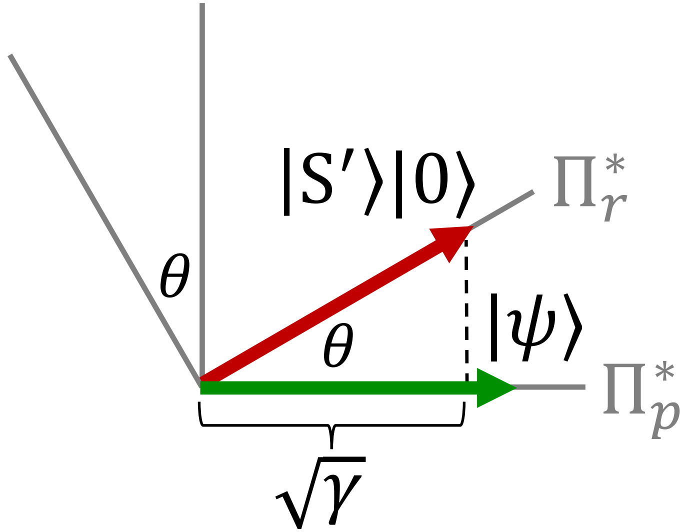
In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

Claim: ψ_A corresponds to a strongly $(p - \varepsilon)$-successful adversary.

Proof Sketch

1) If we run $\mathbf{MW}$ twice, two estimates are ε-close with prob $1 - \delta$.

In a nutshell: $\Pi_r^* := \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can analyze 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.

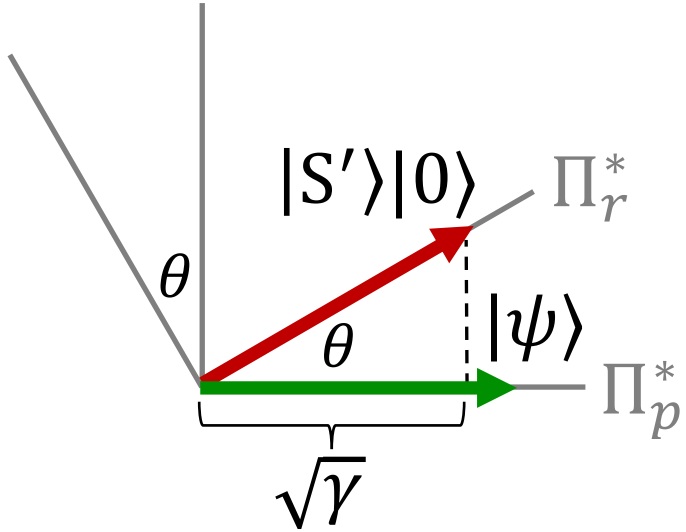
Claim: ψ_A corresponds to a strongly $(p - \varepsilon)$-successful adversary.

Proof Sketch

1) If we run MW twice, two estimates are ε-close with prob $1 - \delta$.

2) $|\psi\rangle = \Pi_p^*|S'\rangle|0\rangle$ corresponds to running $\mathsf{MW}(|S'\rangle) \to q$ and *conditioning* on $q \geq p$.

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



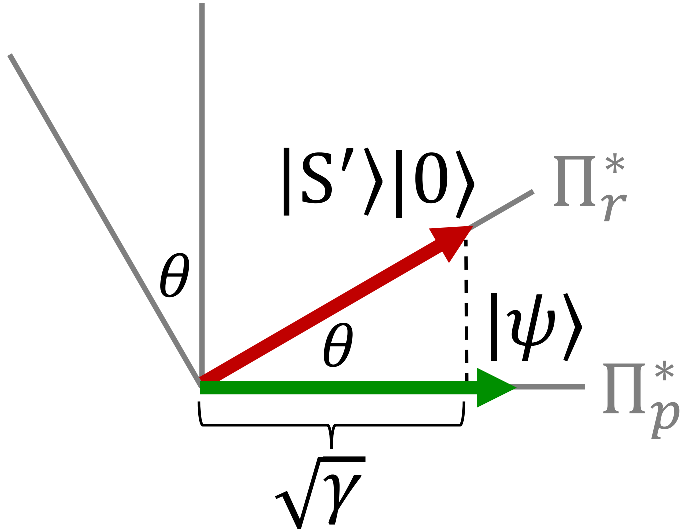
In any 2-D Jordan subspace: if we start at $|S'\rangle|0\rangle$ we end up at $|\psi\rangle$ after Π_p^* accepts.

Claim: ψ_A corresponds to a strongly $(p - \varepsilon)$-successful adversary.

Proof Sketch

1) If we run $\mathbf{MW}$ twice, two estimates are ε-close with prob $1 - \delta$.
2) $|\psi\rangle = \Pi_p^* |S'\rangle|0\rangle$ corresponds to running $\mathbf{MW}(|S'\rangle) \to q$ and *conditioning* on $q \geq p$.
3) Markov: if we run $\mathbf{MW}$ on ψ_A, get $\geq p - \varepsilon$ with prob $1 - \delta/\gamma$.

In a nutshell: $\Pi_r^* \coloneqq \Pi_r \otimes |0\rangle\langle 0|_W$ works because we can analyze 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 corresponds to a strongly $(p - \varepsilon)$-successful adversary.

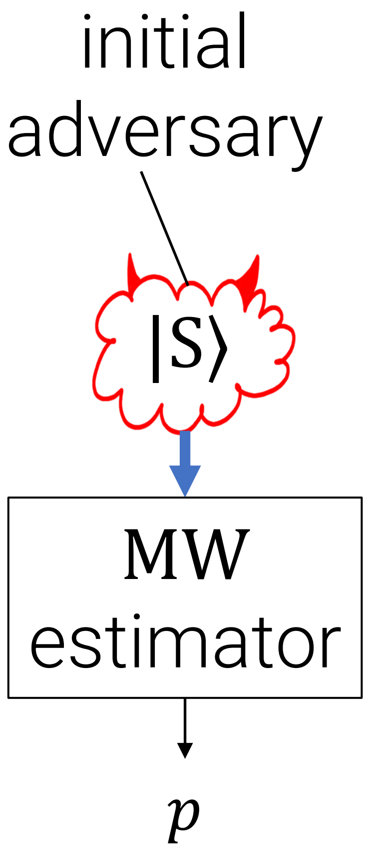
Proof Sketch

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

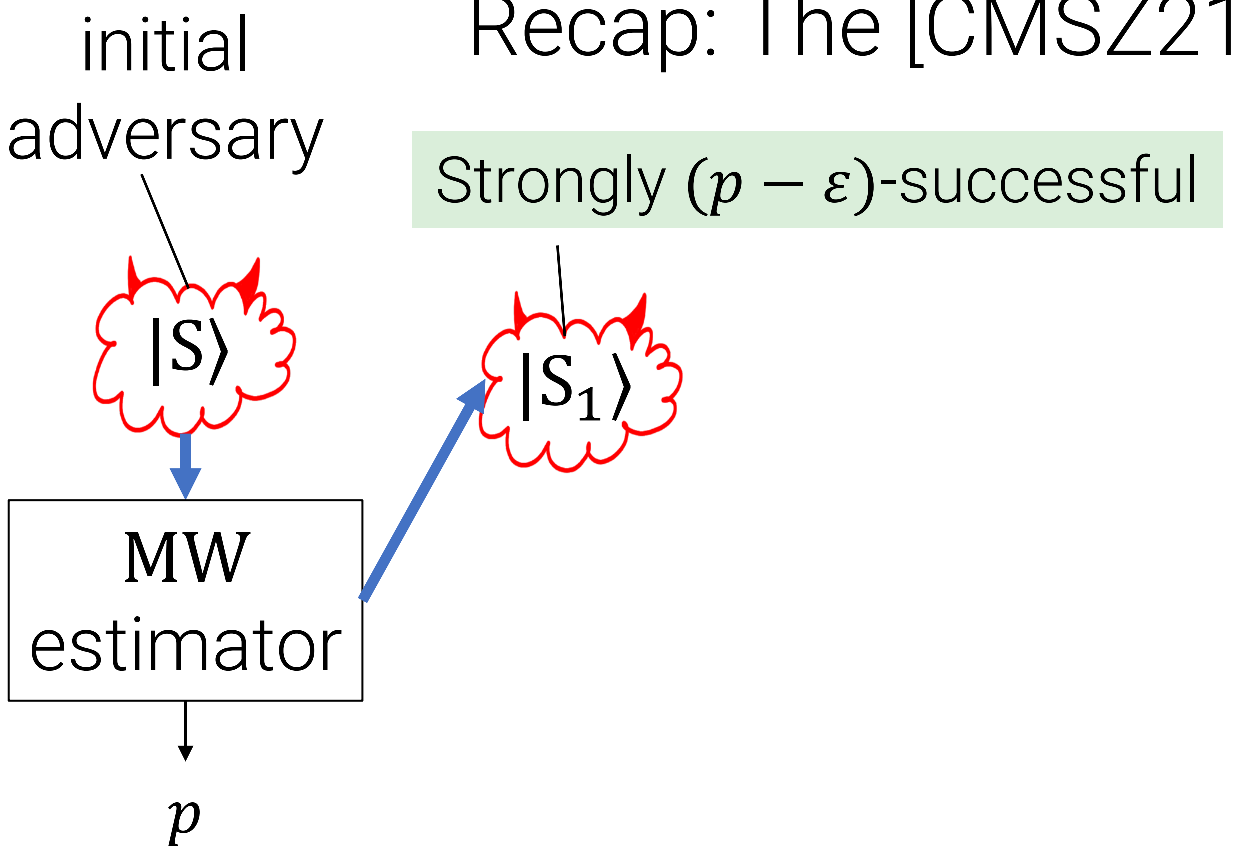
Recap: The [CMSZ21] Rewinding Procedure

initial
adversary

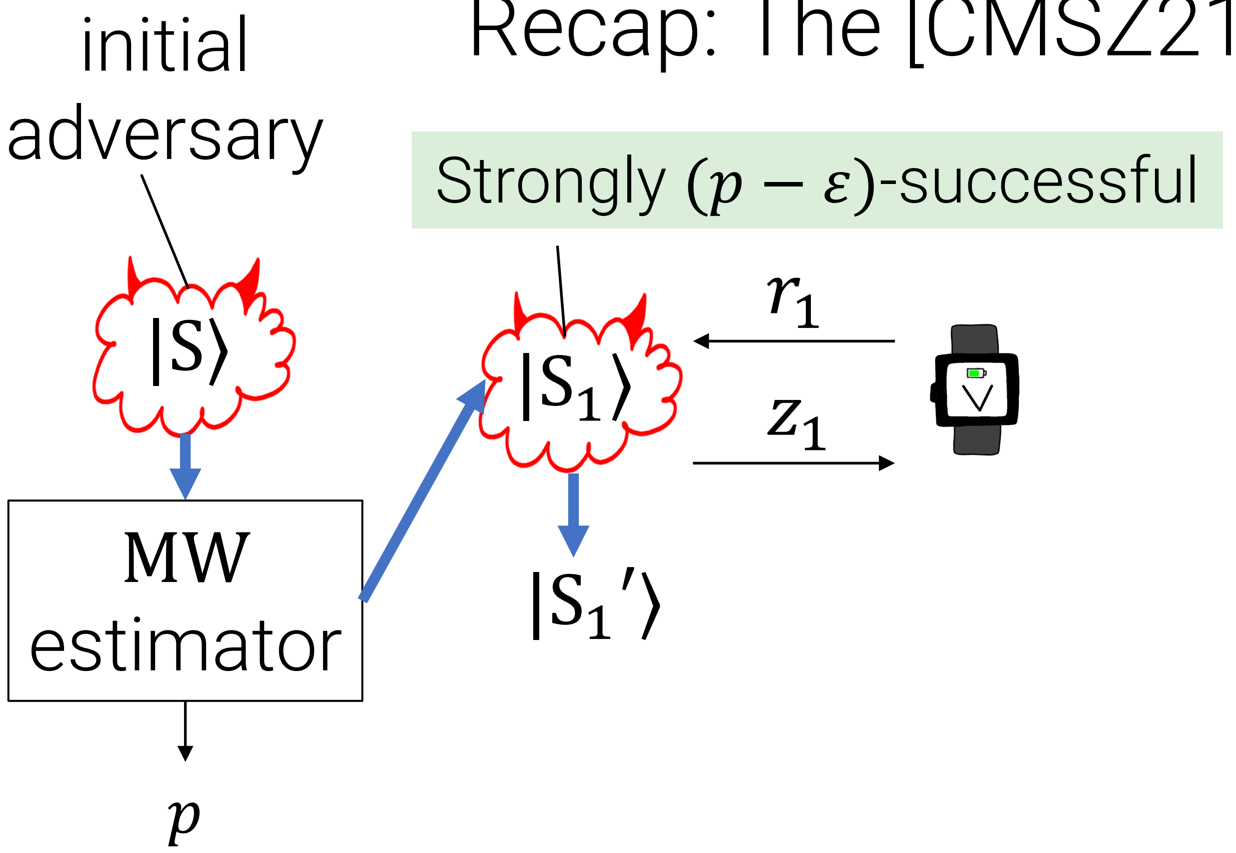
Recap: The [CMSZ21] Rewinding Procedure



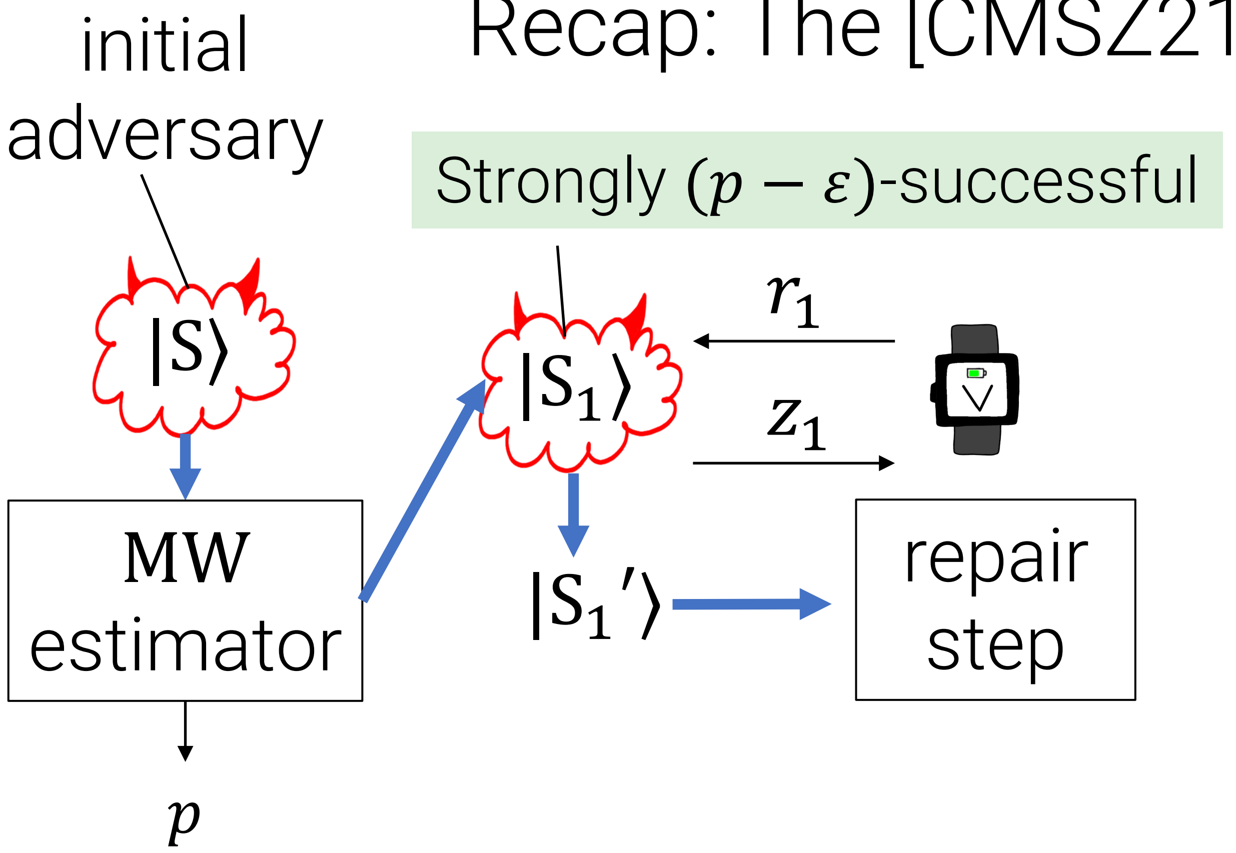
Recap: The [CMSZ21] Rewinding Procedure



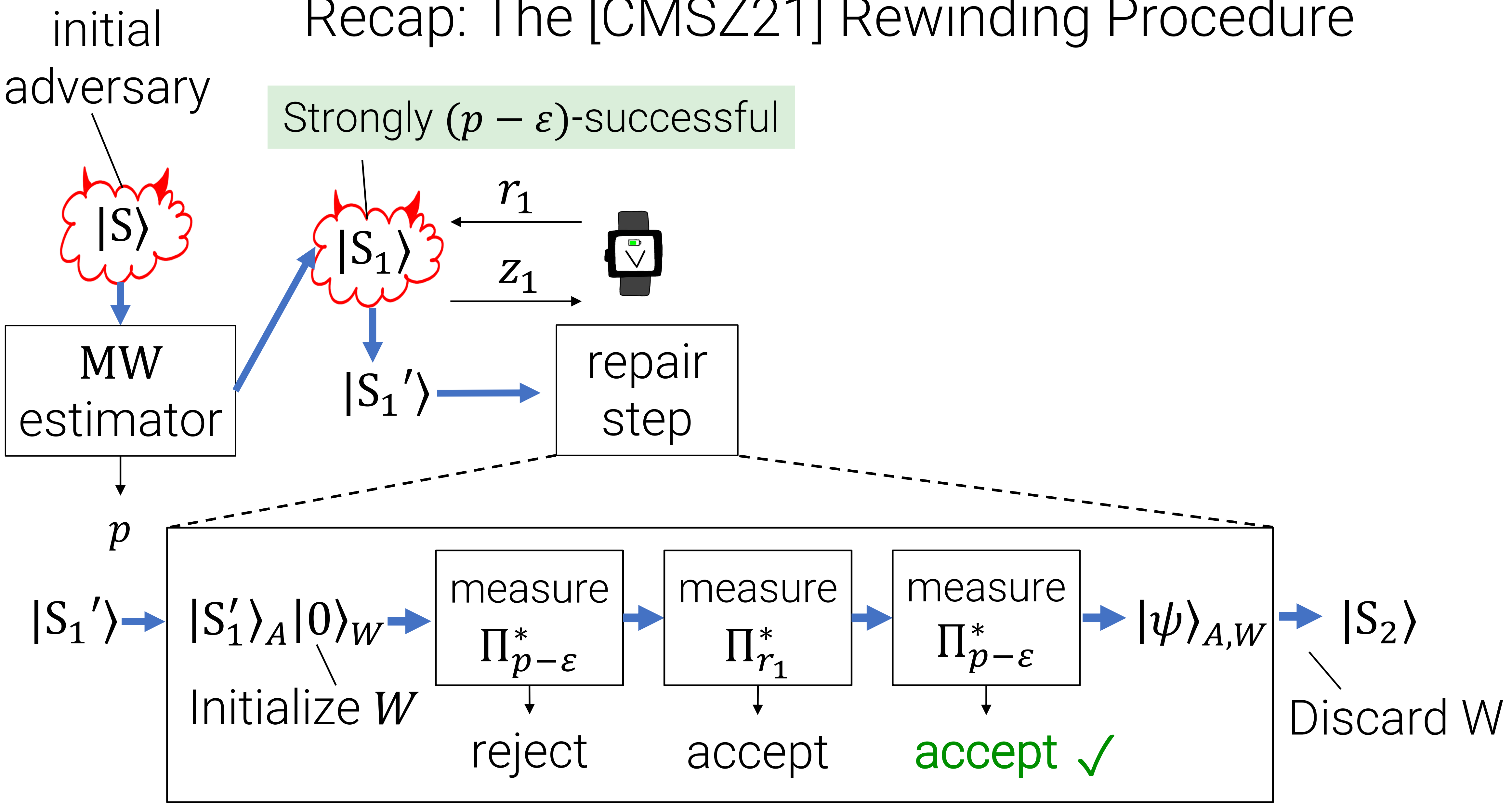
Recap: The [CMSZ21] Rewinding Procedure



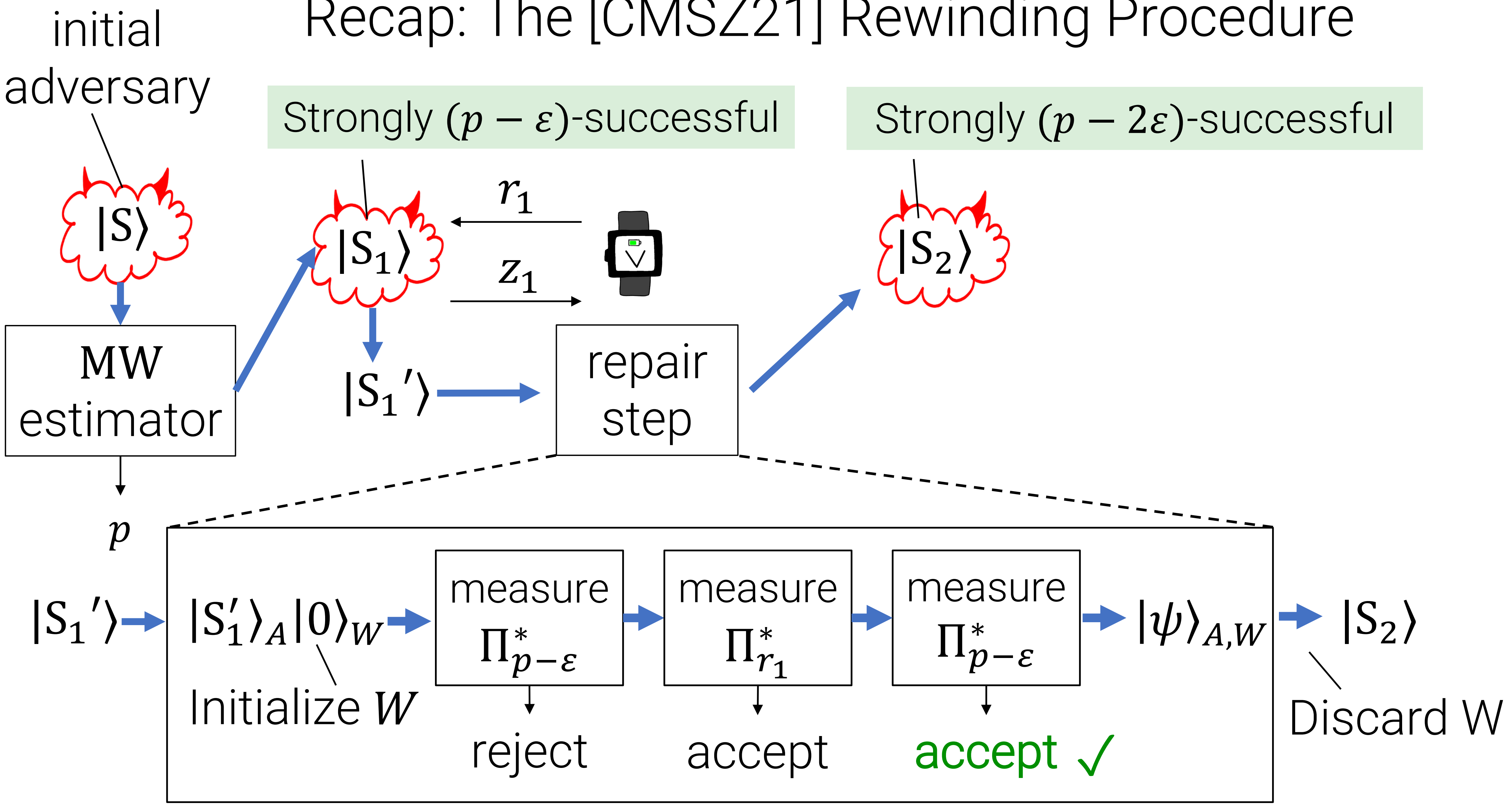
Recap: The [CMSZ21] Rewinding Procedure



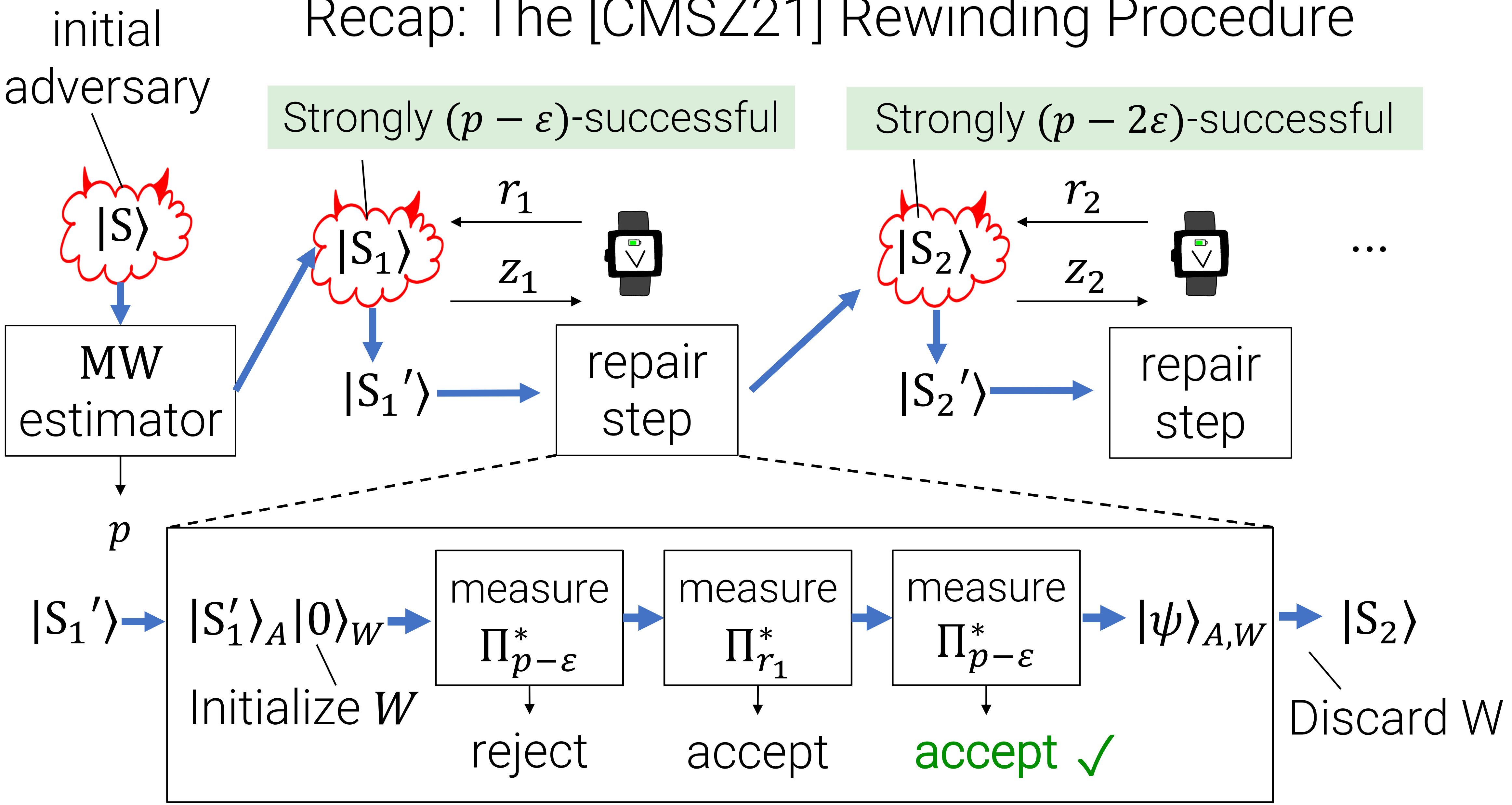
Recap: The [CMSZ21] Rewinding Procedure



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 as well as *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 as well as *optimal* soundness error for many other protocols (e.g., Blum).
- This technique has found many other applications:
  - [Bitansky-Brakerski-Kalai22]: "advice preserving" non-interactive quantum reductions
  - [Lai-Malavolta-Spooner22]: quantum rewinding for many-round protocols
  - [Gunn-Ju-Ma-Zhandry22]: quantum-communication succinct arguments

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 as well as *optimal* soundness error for many other protocols (e.g., Blum).
- This technique has found many other applications:
  - [Bitansky-Brakerski-Kalai22]: "advice preserving" non-interactive quantum reductions
  - [Lai-Malavolta-Spooner22]: quantum rewinding for many-round protocols
  - [Gunn-Ju-Ma-Zhandry22]: quantum-communication succinct arguments

So have we resolved quantum rewinding?

Where does this leave us?

The [CMSZ21] technique is *still* not as powerful as classical rewinding:

Where does this leave us?

The [CMSZ21] technique is ***still*** not as powerful as classical rewinding:

- Needs advice: an explicit lower bound on the prover's initial success probability is needed to guarantee extraction. In general, this lower bound may not be physically accessible.

Where does this leave us?

The [CMSZ21] technique is *still* not as powerful as classical rewinding:

- Needs advice: an explicit lower bound on the prover's initial success probability is needed to guarantee extraction. In general, this lower bound may not be physically accessible.

- Much slower than classical rewinding: if prover is ε-successful, it takes $1/\varepsilon^5$ steps to extract!

Where does this leave us?

The [CMSZ21] technique is *still* not as powerful as classical rewinding:

- Needs advice: an explicit lower bound on the prover's initial success probability is needed to guarantee extraction. In general, this lower bound may not be physically accessible.

- Much slower than classical rewinding: if prover is ε-successful, it takes $1/\varepsilon^5$ steps to extract!

- Doesn't preserve the prover's state: prover state after extraction may be completely different than the adversary's real (post-execution) state. This is *by design*, since repair only restores success probability.

This concludes: the unreasonable effectiveness of alternating projectors in quantum rewinding.

Thank You!

Questions?