

Quantum Rewinding Tutorial Part 3:

Zero Knowledge Beyond Watrous

Alex Lombardi

(MIT → Simons & Berkeley)

Based on:

- “Post-Quantum Zero Knowledge, Revisited” by Alex Lombardi, Fermi Ma, and Nicholas Spooner (2022)

Today So Far

Post-Quantum Soundness:

- [Unruh12, Unruh16]: weak soundness for special protocols
- [CMSZ21]: strong soundness for more protocols
- Key components: **collapsing, special soundness**

Post-Quantum Zero-Knowledge:

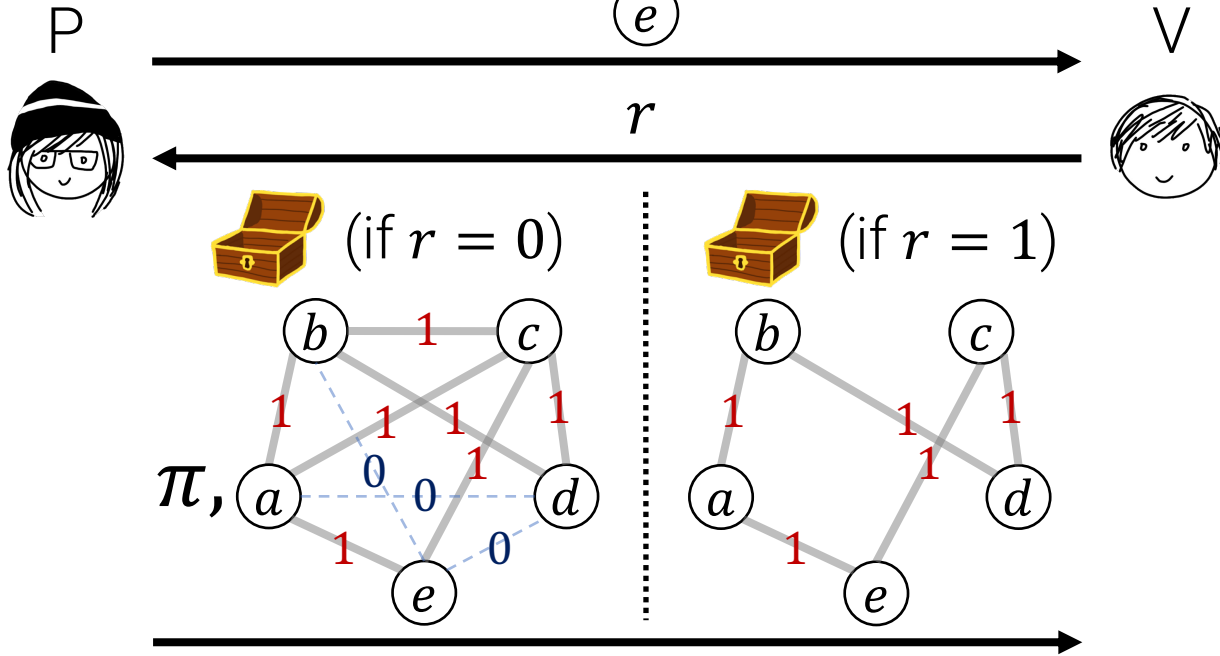
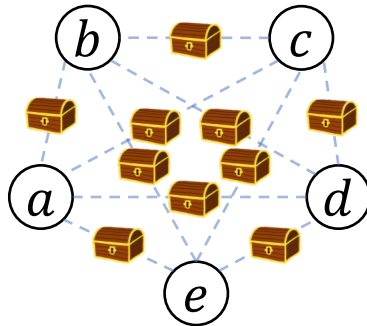
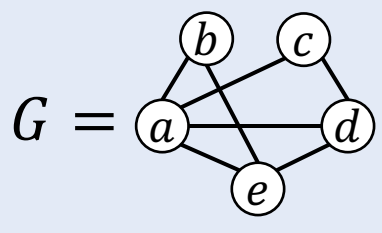
- [Watrous06] ZK for very special protocols
 - “Oblivious guessing simulator” -> full simulator
- ???



This talk: new toolkit for post-quantum ZK

What is [Watrous06] good for?

Blum's Protocol for Hamiltonian Cycle



Watrous simulates by guessing r .

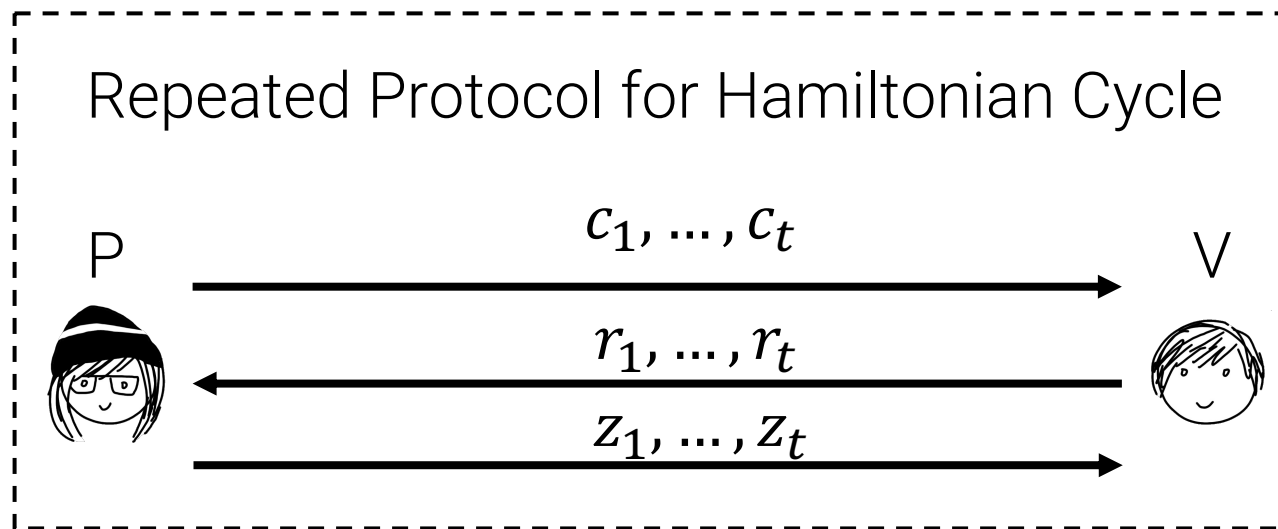
Works when r is one bit.

- Actual protocol is **sequential repetition** of Blum

Works when only poly many r , but soundness is still poor.

What about arbitrary length r ?

Simulation Beyond Guessing

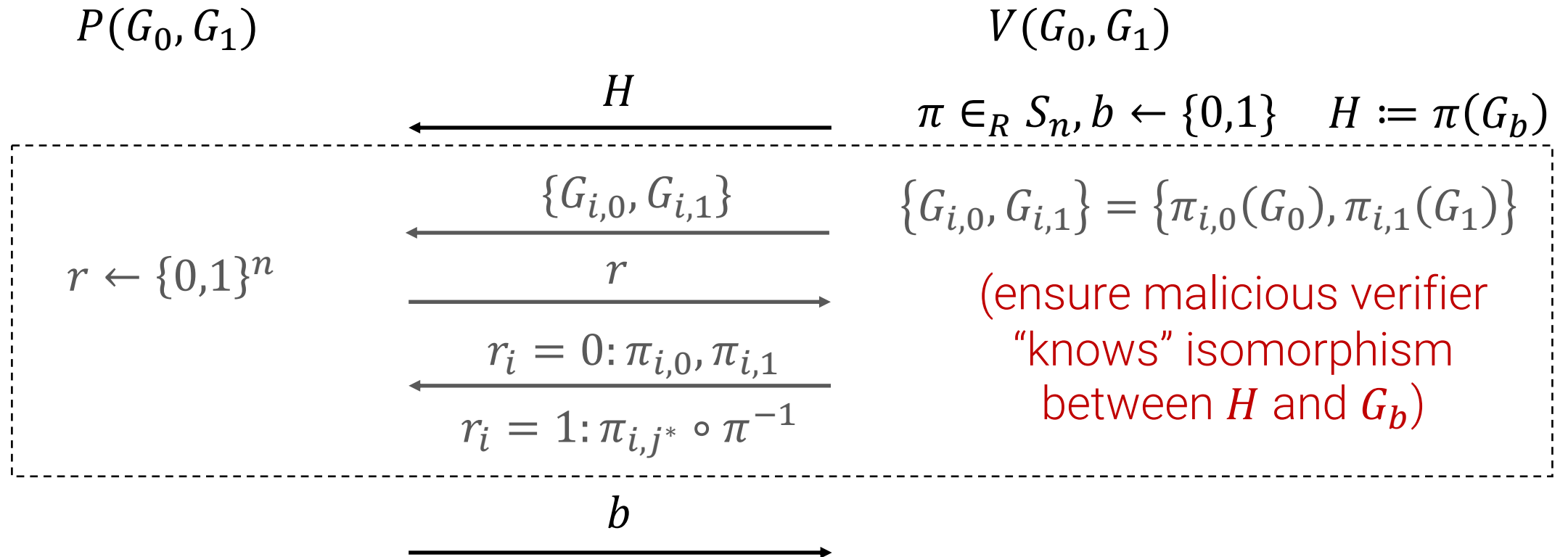


This is not even classically ZK!
(How does Sim get r_1, \dots, r_t ?)

- There are two classical “fixes” to this: paradigms due to Goldreich-Kahan [GK96] and Feige-Shamir [FS90]. **Have the verifier commit to its challenge in advance.**
- Simulation task is no longer about **guessing**, but about **extracting**
- New tools can show: both of these paradigms are post-quantum secure!

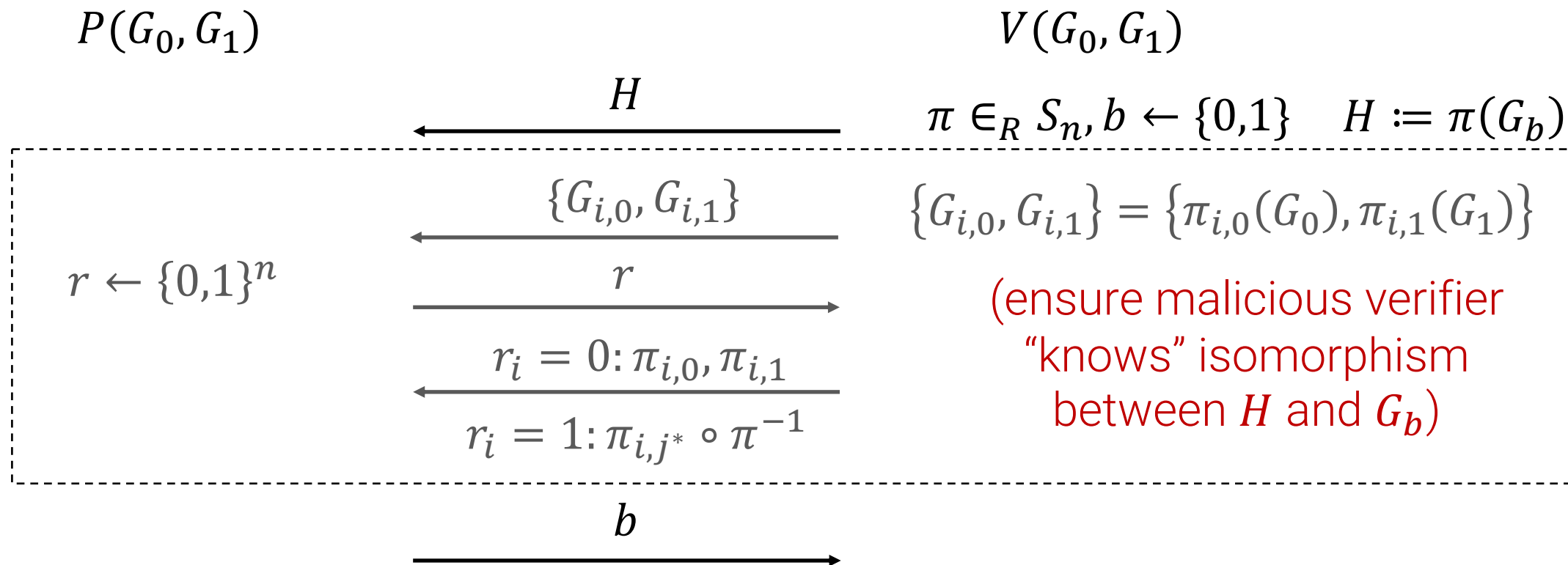
Example: Graph non-isomorphism

Graph non-isomorphism protocol [GMW86]



Aside: [GMW86] also gives GI protocol, which is ZK by Watrous

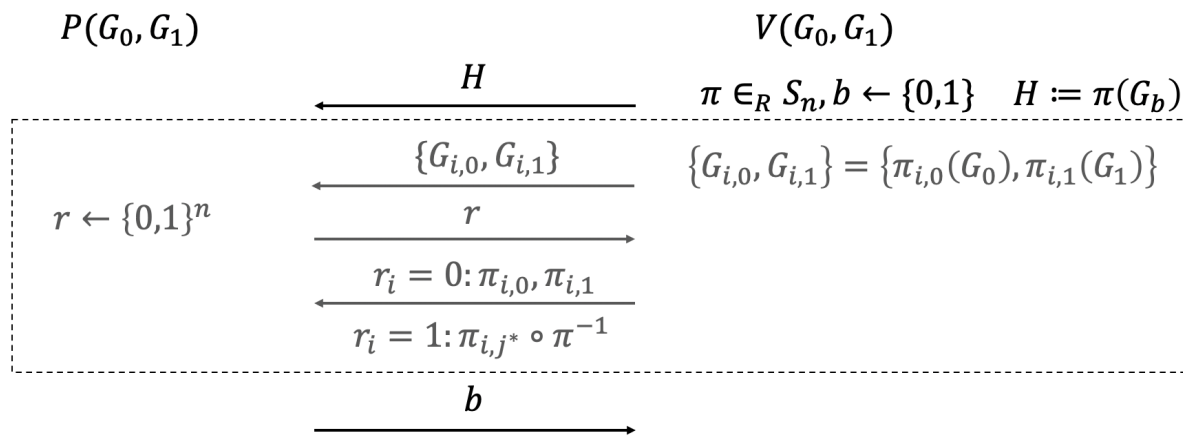
Graph non-isomorphism protocol [GMW86]



GNI protocol is classically (statistically) zero knowledge.

Is it post-quantum ZK?

Graph non-isomorphism protocol [GMW86]

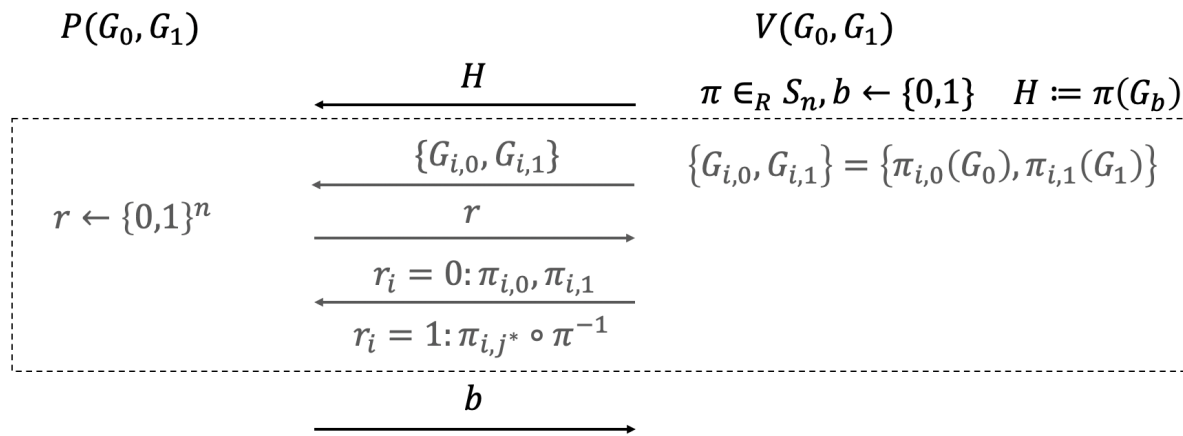


Intuition:

If G_0, G_1 are not isomorphic, any H defines “correct” b

Classical Sim: rewind V^* to extract b

Graph non-isomorphism protocol [GMW86]



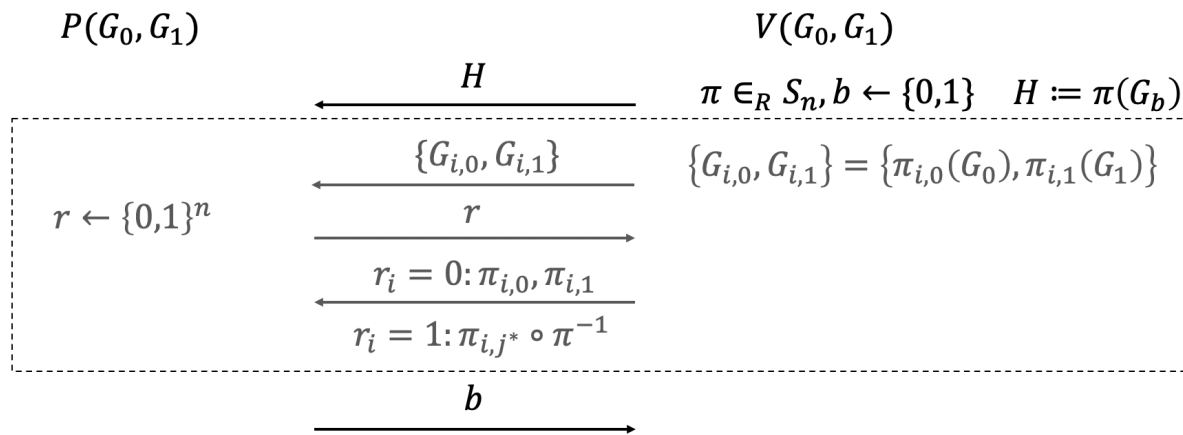
Sim:

- Run V^* once. If P does not reject, rewind V^* (on different r) *until it succeeds again*
- Sub-protocol is special sound \rightarrow extract and output b

Runtime: if V^* is ε -convincing,

- Sim does nothing with probability $1 - \varepsilon$
- Sim rewinds $1/\varepsilon$ times (in expectation) with probability ε
- Expected # of Sim rewinds is $\varepsilon \cdot 1/\varepsilon = 1$

Graph non-isomorphism protocol [GMW86]



Sim:

- Run V^* once. If P does not reject, rewind V^* (on different r) *until it succeeds again*
- Sub-protocol is special sound \rightarrow extract and output b

Takeaway:

ZK of this protocol essentially equivalent to a **soundness property** of the sub-protocol.

Want to be able to **find b** (in **expected poly time**) without changing **V 's state**.

Outline

- 1) Show how to extract with probability 1 in expected polynomial time.
- 2) How to extract in a way that *preserves* the adversary's state.
 - This more or less directly implies ZK simulation
- 3) Serious modelling issue: how to formalize expected polynomial time ZK simulation

Outline

- 1) Show how to extract with probability 1 in expected polynomial time.
- 2) How to extract in a way that *preserves* the adversary's state.
 - This more or less directly implies ZK simulation
- 3) Serious modelling issue: how to formalize expected polynomial time ZK simulation

Outline

1) Show how to extract with probability 1 in expected polynomial time.

- Defining “guaranteed extraction”
- Rewinding with abstract singular vector algorithms [GSLW19].
- New rewinding meta-algorithm

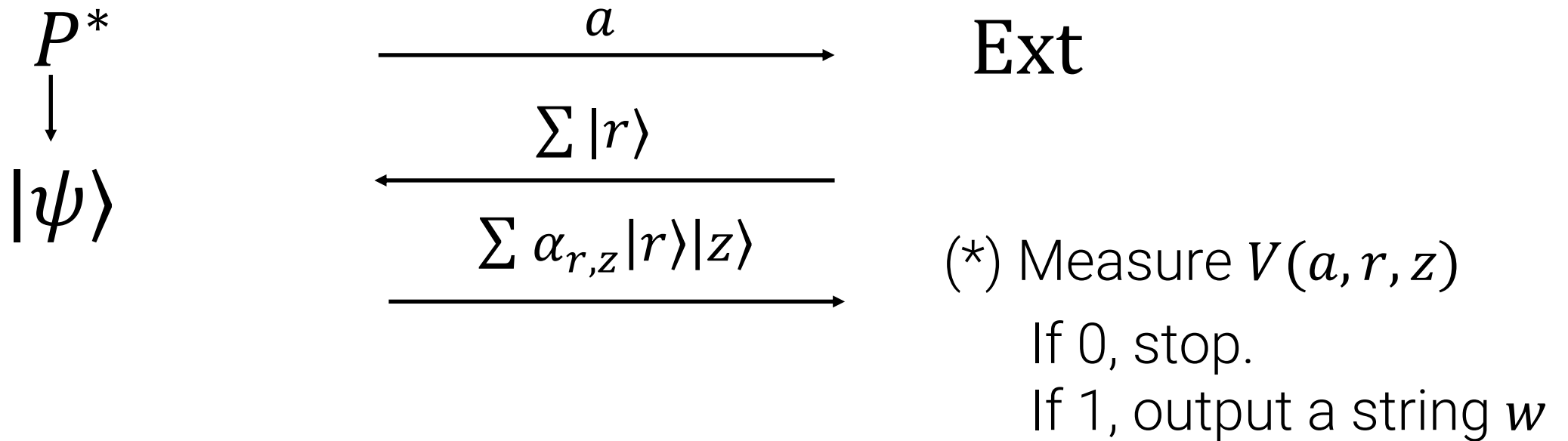
2) How to extract in a way that *preserves* the adversary’s state.

- This more or less directly implies ZK simulation

3) Serious modelling issue: how to formalize expected polynomial time ZK simulation

Guaranteed Extraction

A 3-message protocol has **guaranteed extraction** for a relation $R(x, w)$ if it has an extraction procedure of the following form:



Where:

- If (*) does not abort, then $R(x, w)$ holds with $1 - \text{negl}(\lambda)$ probability
- **Ext** runs in expected poly-time

Achieving Guaranteed Extraction

Theorem [LMS22]:

Any 3-message protocol satisfying the following two conditions has guaranteed extraction for $R(x, w)$:

- k -Special Soundness: k accepting pairs (r_i, z_i) (for fixed a) \rightarrow valid w .
- Collapsing: (Partial) prover responses z are (computationally) unique

Conditions of this form hold for typical protocols (care required)

Achieving Guaranteed Extraction

Theorem [LMS22]:

Any 3-message protocol satisfying the following two conditions has guaranteed extraction for $R(x, w)$:

- k -Special Soundness: k accepting pairs (r_i, z_i) (for fixed a) \rightarrow valid w .
- Collapsing: (Partial) prover responses z are (computationally) unique

Q: Does CMSZ solve this problem?

A: If only life were so easy

Recap of [CMSZ21] Extraction

Start with prover state $|\psi\rangle$ that is γ -successful

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

$|\psi\rangle|+\rangle_R = \sum_j \alpha_j |v_j\rangle$ in the $(\Pi_{\text{Unif}}, \Pi_{\text{Acc}})$ Jordan basis

Step 1: MW alternating projectors outputs a value p

Invariant: $|\phi\rangle = \sum_j \alpha'_j |v_j\rangle$ where $p_j \geq p$ whp.

Step 2: Try to generate an accepting transcript on random r

Step 3: Repair state $|\psi'\rangle$ to (almost) maintain invariant

Recap of [CMSZ21] Extraction

Invariant: $|\phi\rangle = \sum_j \alpha'_j |v_j\rangle$ where $p_j \geq p$ almost surely.

Step 3: Repair state $|\psi'\rangle$ to (almost) maintain invariant

- Append a workspace register W for running MW
- $\Pi_p^* :=$ ``running MW unitary would result in a value $\geq p - \varepsilon$.
- $\Pi_{r,b}^* := \Pi_{r,b} \otimes |0\rangle\langle 0|_W$. $\Pi_{r,b}$ checks whether $P(r)$ makes V output b .

Alternate $(M_p^*, M_{r,b}^*)$ measurements until $M_p^* \rightarrow 1$

$|\psi'\rangle|0\rangle_W$ is the result of M_r^* applied to state satisfying invariant.

$O(1)$ alternations suffice in expectation to get back.

Recap of [CMSZ21] Extraction

First, let's make this procedure guarantee an output.

Recap of [CMSZ21] Extraction

Start with prover state $|\psi\rangle$ that is γ -successful

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

$|\psi\rangle|+\rangle_R = \sum_j \alpha_j |v_j\rangle$ in the $(\Pi_{\text{Unif}}, \Pi_{\text{Acc}})$ Jordan basis

Step 1: MW alternating projectors outputs a value p

How long? Given accuracy parameter ε , run in $1/\varepsilon^2$ time

Guaranteed extraction asks for a guarantee for *all* provers.

Naïve Guaranteed Extraction

Start with prover state $|\psi\rangle$ that is γ -successful

Step 0: Apply M_{Acc} to $|\psi\rangle|+\rangle_R$ (check if prover success).

If 1, continue.

$\Pi_{\text{Acc}}|\psi\rangle|+\rangle_R = \sum_j \sqrt{p_j} \alpha_j |w_j\rangle$ in the $(\Pi_{\text{Unif}}, \Pi_{\text{Acc}})$ Jordan basis

Step 1: MW alternating projectors outputs a value p

How long? Until your numerator is at least λ

$$\text{Expected runtime } \sum_j |\alpha_j|^2 p_j \cdot \frac{\lambda}{p_j} = \lambda$$

So, what is the overall running time?

Naïve Guaranteed Extraction

Step 3: Repair state $|\psi'\rangle$ to (almost) maintain invariant

Repair runtime: $O(1) \Pi_p^*$ measurement

Π_p^* corresponds to MW estimate to $\varepsilon = p^2$ accuracy.

Why? Need to repair k/p times, start at p .

Complexity of $\Pi_p^* = \frac{1}{\varepsilon^2} = \frac{1}{p^4}$.

Thus, total runtime of the extractor is $\frac{k}{p} \cdot \frac{1}{p^4} = \frac{1}{p^5}$ if you start at p .

Naïve Guaranteed Extraction

Thus, total runtime of the extractor is $\frac{k}{p} \cdot \frac{1}{p^4} = \frac{1}{p^5}$ if you start at p .

Q: What is p ?

$\Pi_{\text{Acc}}|\psi\rangle|+\rangle_R = \sum_j \sqrt{p_j} \alpha_j |w_j\rangle$ in the $(\Pi_{\text{Unif}}, \Pi_{\text{Acc}})$ Jordan basis

A: Arbitrary nonzero number???

Expected running time is $\sum_j \alpha_j^2 p_j \cdot \frac{1}{p_j^5} = \infty$

Achieving Guaranteed Extraction

Theorem [LMS22]:

Any 3-message protocol satisfying k -Special Soundness and collapsing has guaranteed extraction for $R(x, w)$:

Proof Idea:

Classical Rewinding Runtime

- With probability ε , rewind k/ε times
- Total work $\varepsilon \cdot k/\varepsilon \cdot 1 = k$

Quantum Rewinding Runtime

- With probability ε , rewind **and repair** k/ε times
- Total work $\varepsilon \cdot k/\varepsilon \cdot 1/\varepsilon^4 = \infty$

Achieving Guaranteed Extraction

Theorem [LMS22]:

Any 3-message protocol satisfying k -Special Soundness and collapsing has guaranteed extraction for $R(x, w)$:

Proof Idea:

Classical Rewinding Runtime

- With probability ε , rewind k/ε times
- Total work $\varepsilon \cdot k/\varepsilon \cdot 1 = k$

Our Rewinding Runtime

- With probability ε , rewind **and repair** k times
- Total work $\varepsilon \cdot k \cdot 1/\varepsilon = k$

Key Ideas

1. Alternating projectors is secretly three algorithms at the same time.

- Given $|\phi\rangle = \sum_j \alpha_j |v_j\rangle$ (in the Π_A, Π_B Jordan basis), approximately measure p_j (constant multiplicative accuracy).

(initial estimation)

- Given $|\phi\rangle = \sum_j \alpha_j |v_j\rangle$, accept if $p_j \geq p$ and reject if $p_j \leq p(1 - \epsilon)$

(implementation of Π_p^*)

- Given $|\phi\rangle = \sum_j \alpha_j |v_j\rangle$, find a state in $\text{im}(\Pi_B)$.

(repair procedure)

Key Ideas

1. Alternating projectors is secretly three algorithms at the same time.
 - We can describe a rewinding algorithm making black-box use of procedures for these three tasks.
 - All three can be done faster than alternating projectors!

Key Ideas

1. Alternating projectors is secretly three algorithms at the same time.
 - Given $|\phi\rangle = \sum_j \alpha_j |v_j\rangle$ (in the Π_A, Π_B Jordan basis), approximately measure p_j (constant multiplicative accuracy).

(Variable-Accuracy) Phase Estimation ($\frac{1}{\sqrt{p_j}}$ time using many QFTs)

- Given $|\phi\rangle = \sum_j \alpha_j |v_j\rangle$, accept if $p_j \geq p$ and reject if $p_j \leq p(1 - \epsilon)$

Gap Phase Estimation ($\frac{1}{\epsilon\sqrt{p}}$ time using QFT)

- Given $|\phi\rangle = \sum_j \alpha_j |v_j\rangle$, find a state in $\text{im}(\Pi_B)$.

Singular Vector Transform ($\frac{1}{\sqrt{p_j}}$ time using Est + Grover-type algorithm)

Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics

András Gilyén* Yuan Su† Guang Hao Low‡ Nathan Wiebe§

June 6, 2018

Abstract

Quantum computing is powerful because unitary operators describing the time-evolution of a quantum system have exponential size in terms of the number of qubits present in the system. We develop a new “Singular value transformation” algorithm capable of harnessing this exponential advantage, that can apply polynomial transformations to the singular values of a block of a unitary, generalizing the optimal Hamiltonian simulation results of Low and Chuang [LC17a]. The proposed quantum circuits have a very simple structure, often give rise to optimal algorithms and have appealing constant factors, while typically only use a constant number of ancilla qubits.

We show that singular value transformation leads to novel algorithms. We give an efficient solution to a “non-commutative” measurement problem used for efficient ground-state-preparation of certain local Hamiltonians, and propose a new method for singular value estimation. We also show how to exponentially improve the complexity of implementing fractional queries to unitaries with a gapped spectrum. Finally, as a quantum machine learning application we show how to efficiently implement principal component regression.

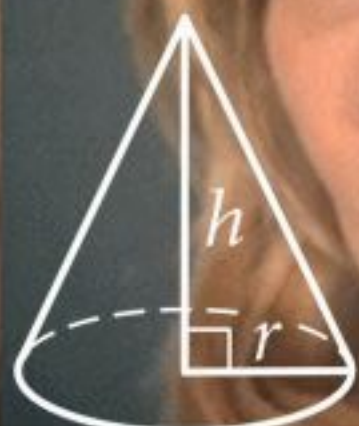

“Singular value transformation” is conceptually simple and efficient, and leads to a unified framework of quantum algorithms incorporating a variety of quantum speed-ups. We illustrate this by showing how it generalizes a number of prominent quantum algorithms, and quickly derive the following algorithms: optimal Hamiltonian simulation, implementing the Moore-Penrose pseudoinverse with exponential precision, fixed-point amplitude amplification, robust

+ variable-runtime modifications



$A = \pi r^2$
 $C = 2\pi r$

$V = \frac{1}{3} \pi r^2 h$





$V = \pi r^2 h$

	30°	45°	60°
sin	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$
cos	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$
tan	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$




$\int \sin x dx = -\cos x + C$
 $\int \frac{dx}{\cos^2 x} = \operatorname{tg} x + C$
 $\int \operatorname{tg} x dx = -\ln|\cos x| + C$
 $\int \frac{dx}{\sin x} = \ln\left|\operatorname{tg} \frac{x}{2}\right| + C$
 $\int \frac{dx}{a^2 + x^2} = \frac{1}{a} \operatorname{arctg} \frac{x}{a} + C$
 $\int \frac{dx}{x^2 - a^2} = \frac{1}{2a} \ln\left|\frac{x-a}{x+a}\right| + C$



$ax^2 + bx + c = 0$
 $a\left(x^2 + \frac{b}{a}x + \frac{c}{a}\right) = 0$
 $x^2 + 2\frac{b}{2a}x + \left(\frac{b}{2a}\right)^2 - \left(\frac{b}{2a}\right)^2 + \frac{4ac}{4a^2} = 0$
 $\left(x + \frac{b}{2a}\right)^2 - \frac{b^2 - 4ac}{4a^2} = 0$

Faster Guaranteed Extraction

Total runtime of the extractor is $\frac{k}{p} \cdot \frac{1}{p^2} = \frac{1}{p^3}$ if you start at p .

Expected running time is $\sum_j \alpha_j^2 p_j \cdot \frac{1}{p_j^3} = \infty$



Key Ideas

1. Alternating projectors is secretly three algorithms at the same time.
2. This rewinding template is doomed...
 - We need k/p transcripts if we start at eigenvalue p .
 - Repair... probably takes more than $O(1)$ total time
 - So... $\sum_j \alpha_j^2 p_j \cdot k/p_j \cdot t(p_j) = \infty$



Key Ideas

1. Alternating projectors is secretly three algorithms at the same time.
2. This rewinding template is doomed...
3. Our rewinding is too classical!
 - Instead of hoping that P^* answers r correctly, let's **force** P^* to answer correctly!
 - Amplify $\text{im}(\Pi_{\text{Unif}}) \rightarrow \text{im}(\Pi_{\text{Acc}})$

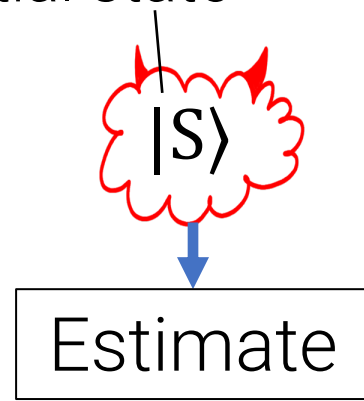
initial state



The [CMSZ21] Extraction Template

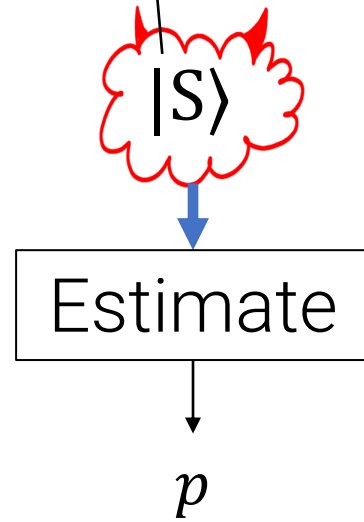
The [CMSZ21] Extraction Template

initial state



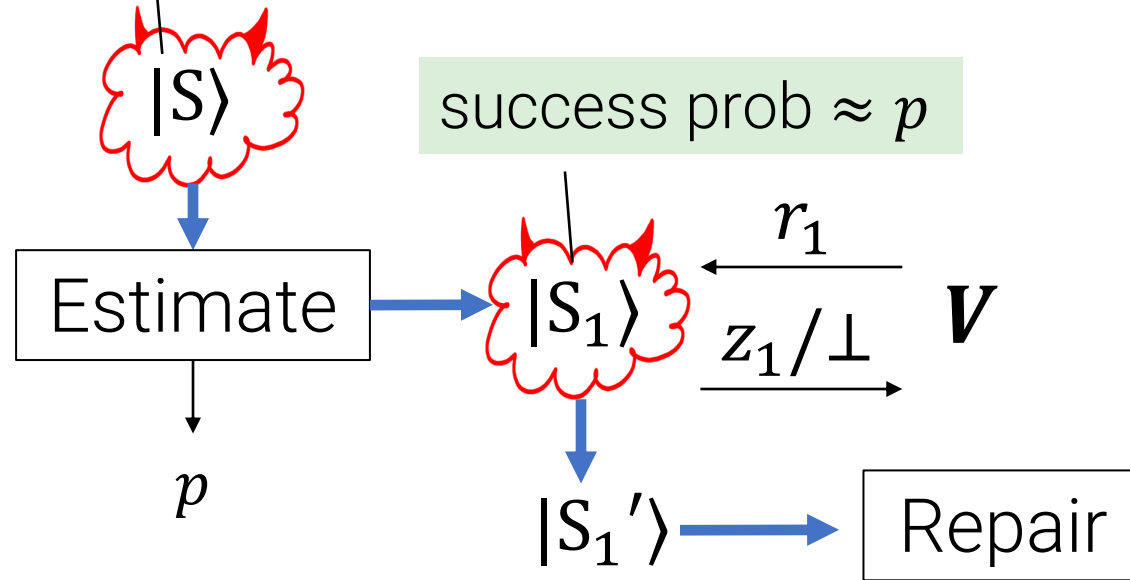
The [CMSZ21] Extraction Template

initial state



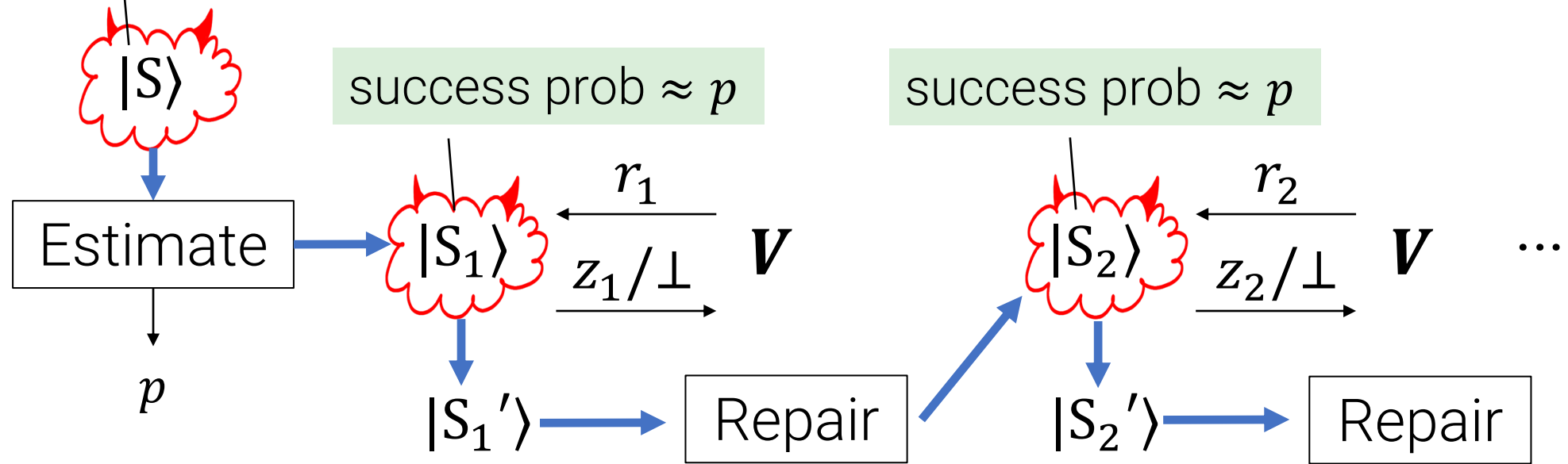
The [CMSZ21] Extraction Template

initial state

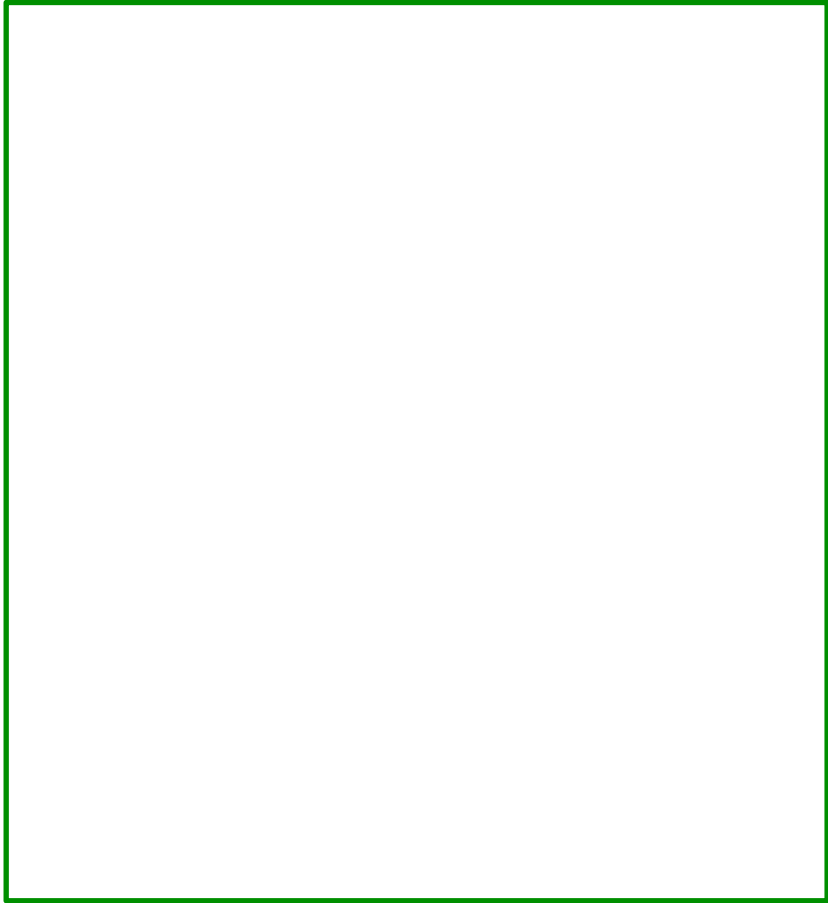


initial state

The [CMSZ21] Extraction Template




Our Extractor



Our Extractor

1) Use joint state of challenge-adversary.

superposition
of challenges

$$|+_R\rangle \otimes |S\rangle$$


Our Extractor

1) Use joint state of challenge-adversary.

superposition
of challenges

$$|+_R\rangle \otimes |S\rangle$$

Estimate $\rightarrow p$

Our Extractor

- 1) Use joint state of challenge-adversary.
- 2) QSVT ensures we only measure accepting (r, z) .

superposition
of challenges

$$|+\rangle_R \otimes |S\rangle$$

Estimate

p

QSVT

$|S_r\rangle$ guaranteed
to succeed on r

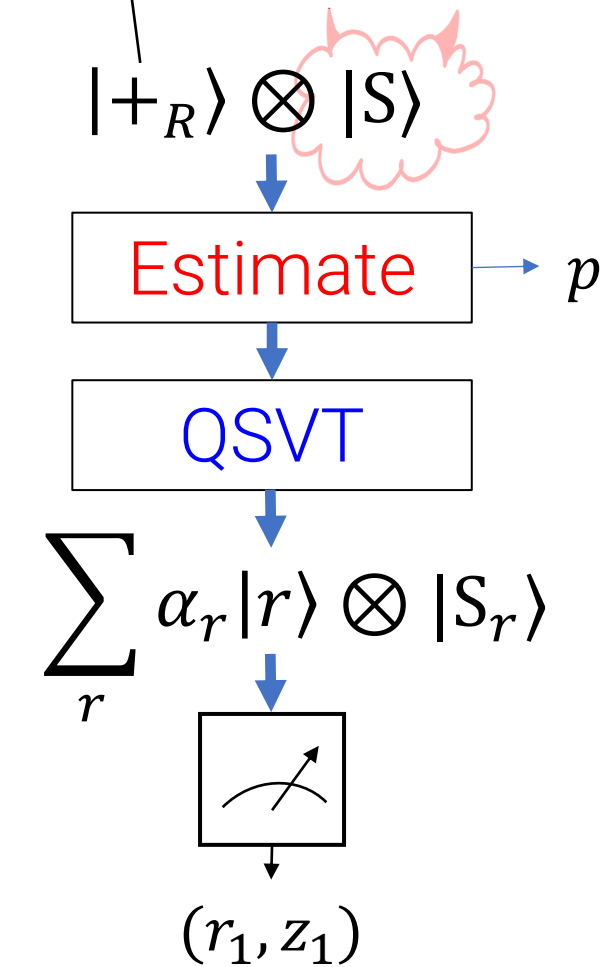
$$\sum_r \alpha_r |r\rangle \otimes |S_r\rangle$$

superposition of
accepting executions

Our Extractor

- 1) Use joint state of challenge-adversary.
- 2) QSVT ensures we only measure accepting (r, z) .

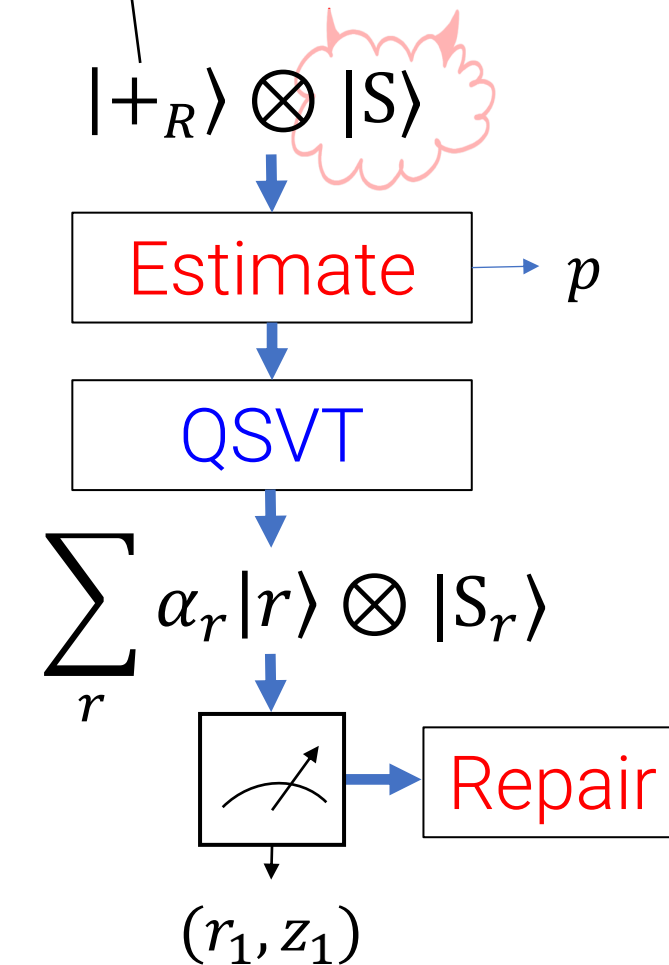
superposition
of challenges



Our Extractor

- 1) Use joint state of challenge-adversary.
- 2) QSVT ensures we only measure accepting (r, z) .

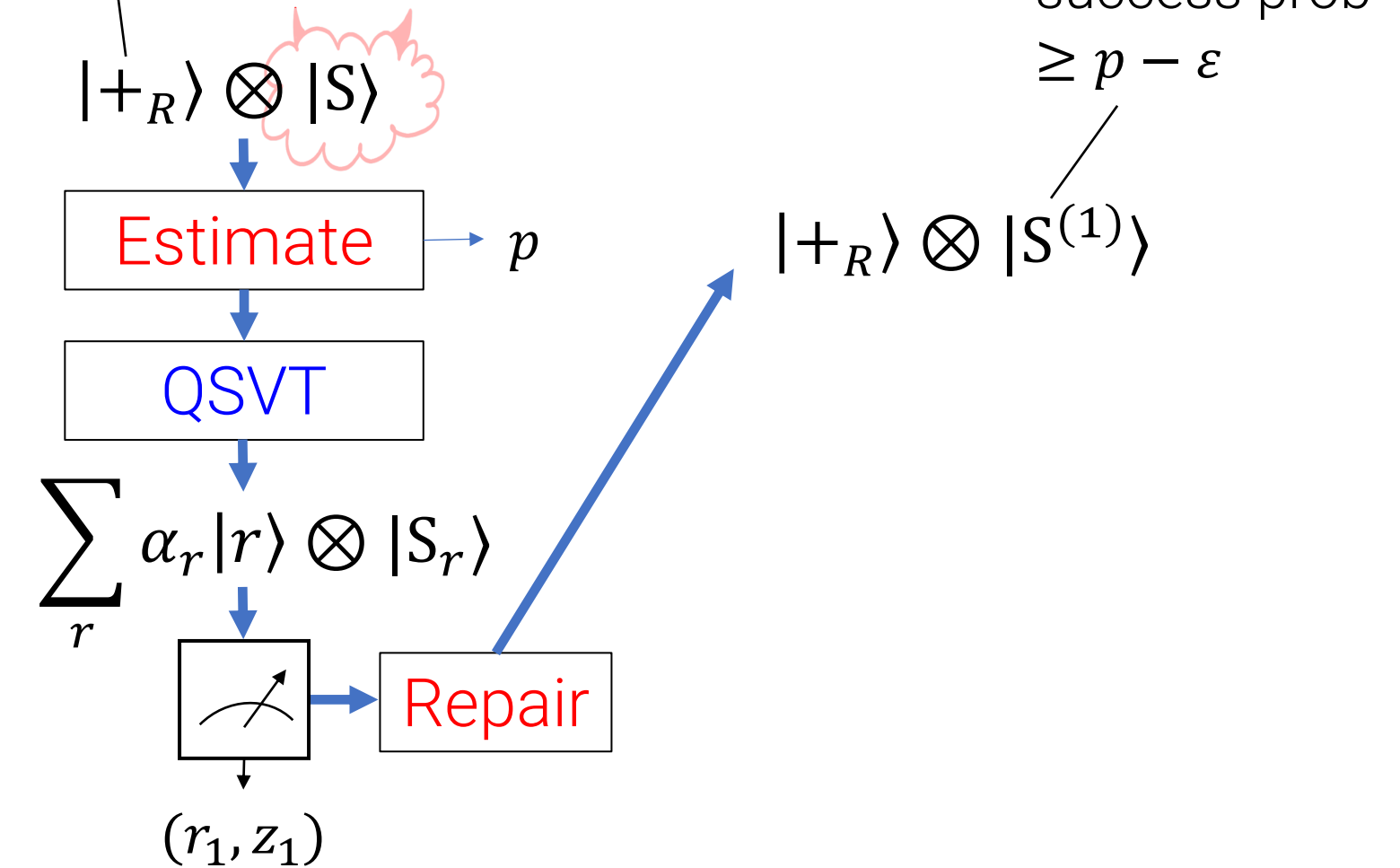
superposition
of challenges



Our Extractor

- 1) Use joint state of challenge-adversary.
- 2) QSVT ensures we only measure accepting (r, z) .

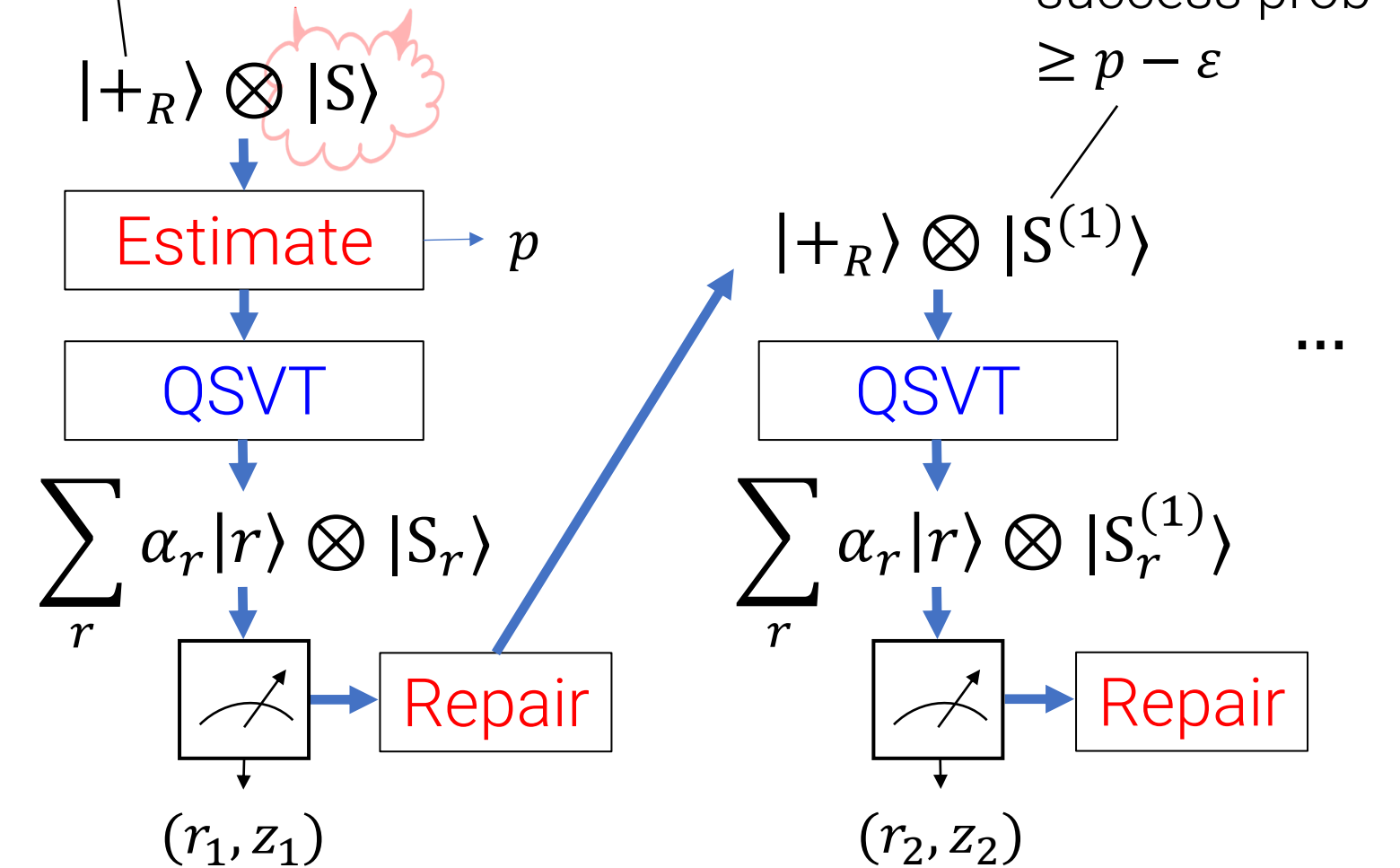
superposition
of challenges



Our Extractor

- 1) Use joint state of challenge-adversary.
- 2) QSVT ensures we only measure accepting (r, z) .

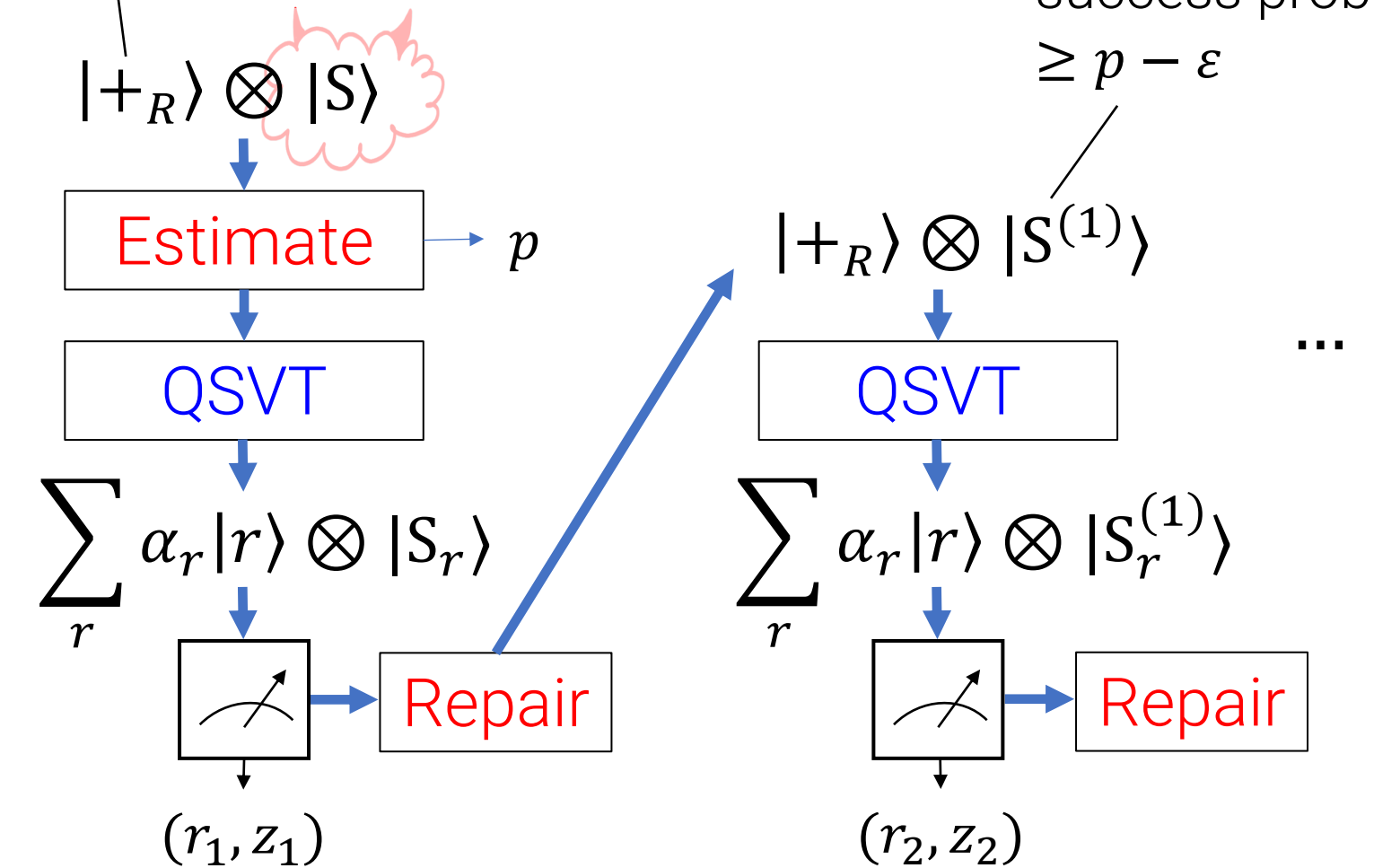
superposition
of challenges



Our Extractor

- 1) Use joint state of challenge-adversary.
- 2) QSVT ensures we only measure accepting (r, z) .
- 3) **Fast algorithms + careful analysis + miracle**: repair takes $1/p$ time.

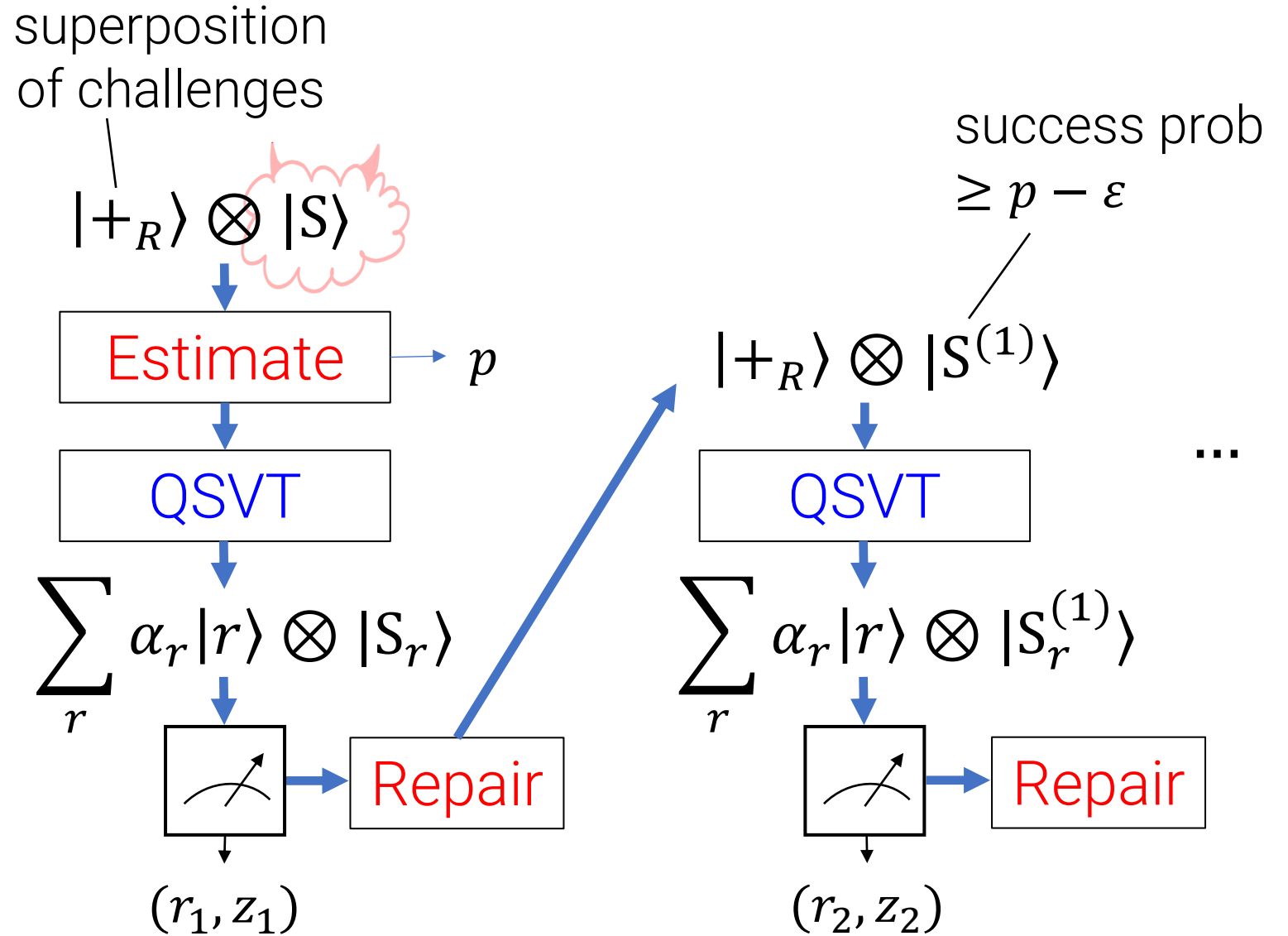
superposition
of challenges



Our Extractor

Why does Repair work?

- In [CMSZ], “Return to subspace principle”
- If $|S^{(p)}\rangle$ was disturbed only a small amount, then Repair will return $|S'_r\rangle$ to the p -subspace quickly
- But measurement of r is a huge disturbance...

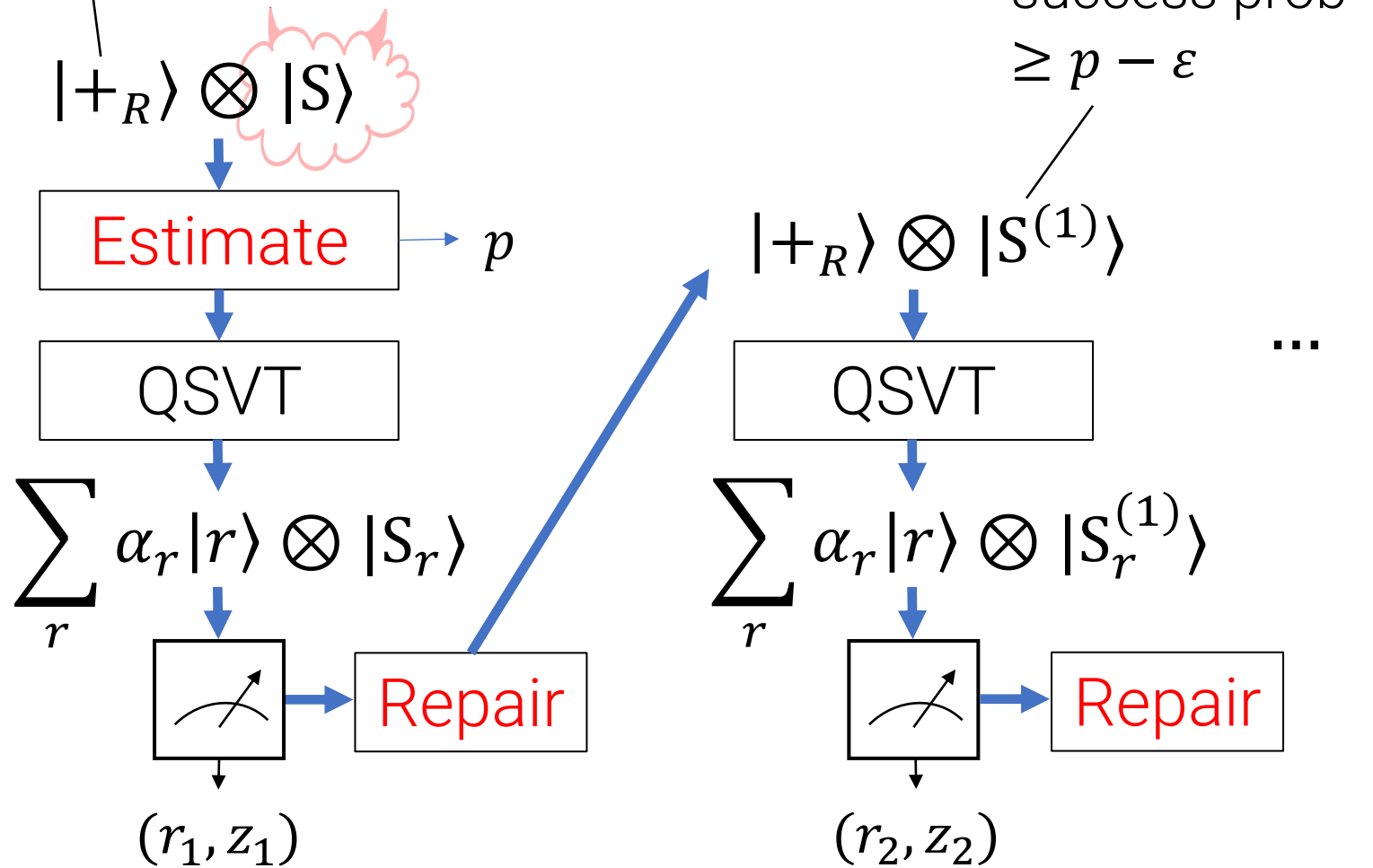


Our Extractor

Our Analysis:

- 1) Measuring (r_i, z_i) looks disturbing, but:
- 2) There *exists* a good $|\hat{S}\rangle$ such that $|S'_1\rangle$ *could have* been obtained from a $\log 1/p$ disturbance to $|\hat{S}\rangle$.
- 3) Fast repair succeeds after $1/\sqrt{p}$ calls to Estimate.

superposition
of challenges



Not shown: very insidious problem with the runtime

Solution: calculate how long repair should take before disturbing the state.

Outline

- 1) Show how to extract with probability 1 in expected polynomial time.
- 2) How to extract in a way that *preserves the adversary's state*.
 - This more or less directly implies ZK simulation
- 3) Serious modelling issue: how to formalize expected polynomial time ZK simulation

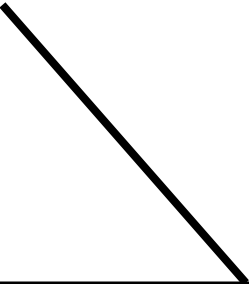
State-Preserving Extraction

A protocol has **state-preserving extraction** if it has an extraction procedure $\text{SPExt}^{P^*} \rightarrow (\tau, w, |\psi\rangle)$ such that:

- The probability that τ is accepting but w is invalid is negligible.
- SPExt runs in expected polynomial time.
- $(\tau, |\psi\rangle)$ is computationally indistinguishable from real P^* view

Dual notion to post-quantum ZK.

Lemma: Guaranteed extraction \rightarrow state-preserving extraction for nice protocols.



(Informally) Whenever the extractor outputs a (computationally) unique string

Example to have in mind: commit-and-prove protocols

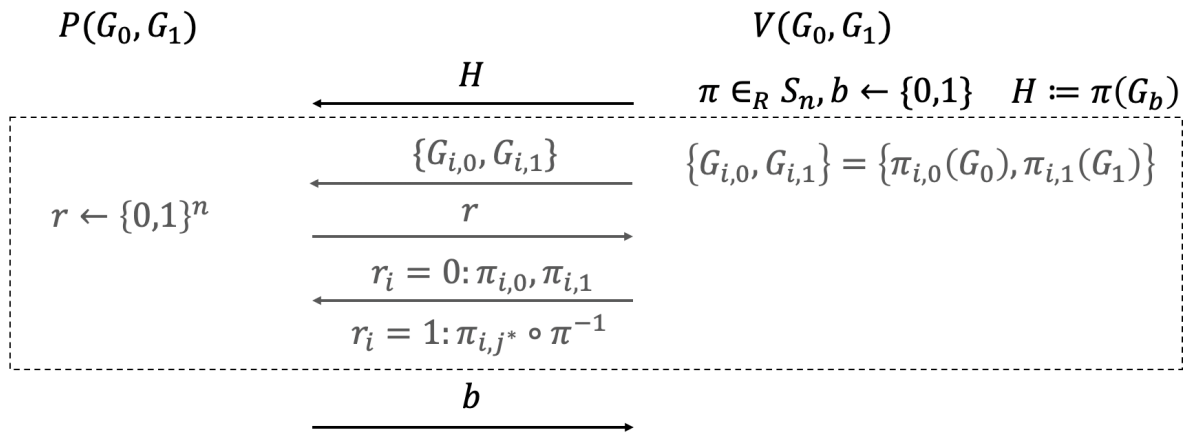
Reduction to Guaranteed Extraction

SPExt Sketch:

1. Run the guaranteed extractor coherently
2. Measure (a, b) and (if $b = 1$) measure w
3. Uncompute (1)
4. Run P^* forwards to get (r, z)

w measurement in (2) is undetectable, so (1) and (3) “cancel out.”

Putting everything together



- 1) GNI 3-message subprotocol has guaranteed extraction
- 2) So it has state-preserving extraction
- 3) So the GNI protocol is post-quantum ZK!

Outline

- 1) Show how to extract with probability 1 in expected polynomial time.
- 2) How to extract in a way that *preserves* the adversary's state.
 - This more or less directly implies ZK simulation
- 3) Serious modelling issue: how to formalize expected polynomial time ZK simulation

What is our simulator doing?

SPExt Sketch:

1. Run the guaranteed extractor coherently
2. Measure (a, b) and (if $b = 1$) measure w
3. Uncompute (1)
4. Run P^* forwards to get (r, z)

(1) and (3) think of the guaranteed extractor as a unitary U + measurement, and apply U (step 1) and U^\dagger (step 3).

Is this expected polynomial time? What is expected polynomial time??

What is Expected Quantum Polynomial Time?

We would like to think of our QTMs as finishing their computation when they reach the final state q_f . However, it is unclear how we should regard a machine which reaches a superposition in which some configurations are in state q_f but others are not. We try to avoid such difficulties by saying that a QTM halts on a particular input if it reaches a superposition consisting entirely of configurations in state q_f .

Quantum complexity theory*

Ethan Bernstein[†] Umesh Vazirani[‡]

September 8, 1997

Abstract

In this paper we study quantum computation from a complexity theoretic viewpoint. Our first result is the existence of an efficient universal quantum Turing Machine in Deutsch's model of a quantum Turing Machine [20]. This construction is substantially more complicated than the corresponding construction for classical Turing Machines - in fact, even simple primitives such as looping, branching and composition are not straightforward in the context of quantum Turing Machines. We establish how these familiar primitives can be implemented, and also introduce some new, purely quantum mechanical primitives, such as changing the computational basis, and carrying out an arbitrary unitary transformation of polynomially bounded dimension.

We also consider the precision to which the transition amplitudes of a quantum Turing Machine need to be specified. We prove that $O(\log T)$ bits of precision suffice to support a T step computation. This justifies the claim that the quantum Turing Machine model should be regarded as a discrete model of computation and not an analog one.

We give the first formal evidence that quantum Turing Machines violate the modern (complexity theoretic) formulation of the Church-Turing thesis. We show the existence of a problem, relative to an oracle, that can be solved in polynomial time on a quantum Turing Machine, but requires super-polynomial time on a bounded-error probabilistic Turing Machine; and thus not in the class **BPP**. The class **BQP**, of languages that are efficiently decidable (with small error-probability) on a quantum Turing Machine, satisfies: $\mathbf{BPP} \subset \mathbf{BQP} \subset \mathbf{P}^{\#\mathbf{P}}$. Therefore there is no possibility of giving a math-

Variable-Runtime Quantum Turing Machines

[Deutsch85, Ozawa98]

Register	Function	Initialization
Q	State Machine	$ q_0\rangle$
W	Workspace	$ 0\rangle$
A	Input/Output	$ \psi\rangle$

The “transition function” of a QTM is defined so that if Q contains the “halting state” q_f , no additional computation is done.

Variable-Runtime Quantum Turing Machines

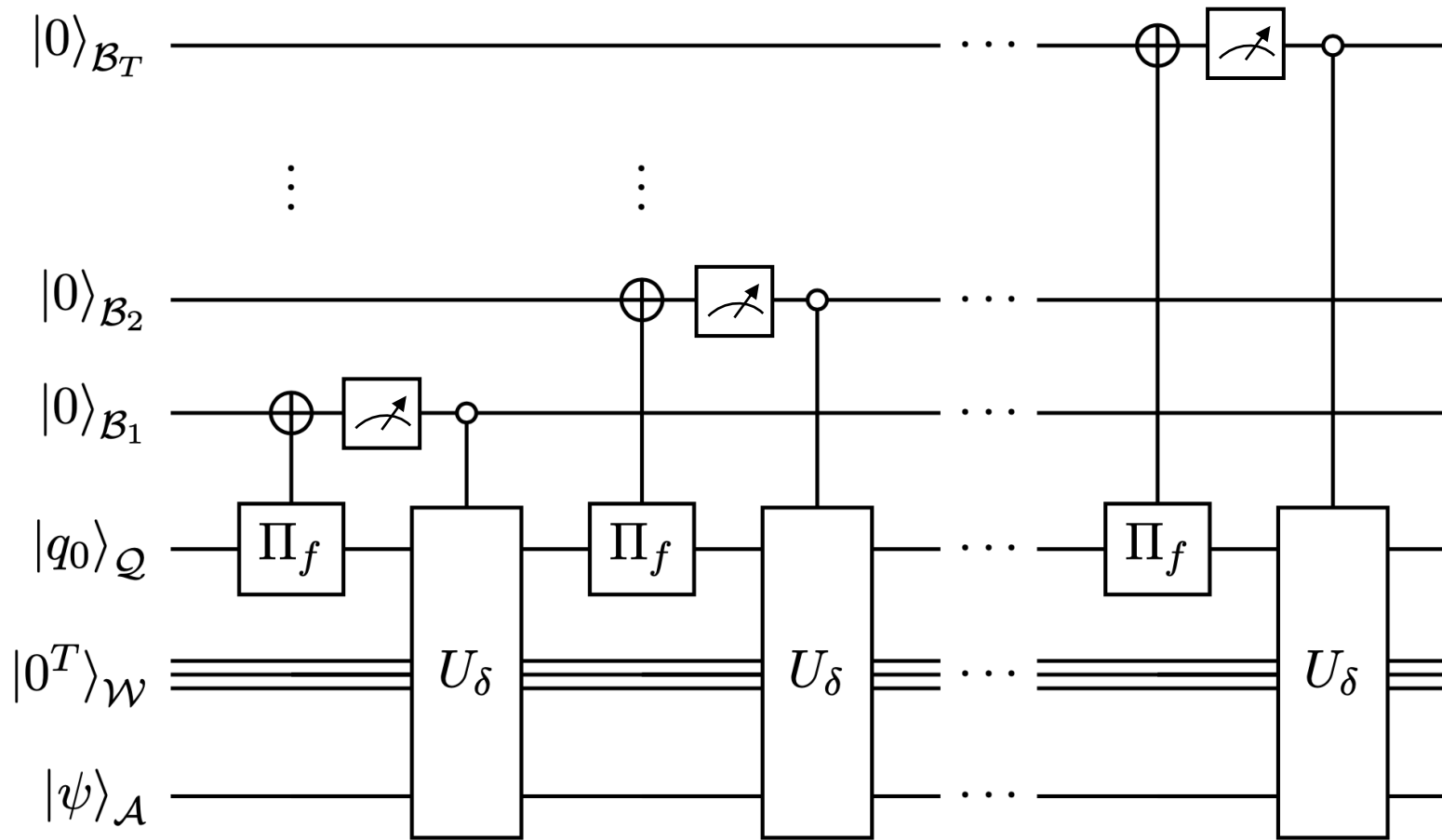
[Deutsch85, Ozawa98]

Register	Function	Initialization
Q	State Machine	$ q_0\rangle$
W	Workspace	$ 0\rangle$
A	Input/Output	$ \psi\rangle$

Observation [M97, O98a, LP98, O98b]:

- 1) The runtime of a QTM is always **effectively measured**
- 2) Computation paths with different running times do not interfere

Measured-Runtime Expected Poly Time (EQPT_m)



\mathcal{B} records the runtime; we require polynomial expected value.

EQPT_m is a bad model for ZK simulation

ZK: the view of every (malicious) verifier can be simulated in EPT

- However, different verifiers will have different simulator *runtime distributions*
- Computation paths with different running times do not interfere
- This means that an EQPT_m simulator will break a superposition of two verifiers with different runtime distributions.

Theorem [Chia-Chung-Liu-Yamakawa, FOCS '21]

Quantum EQPT_m black-box simulation for constant-round protocols is impossible

What is the definition of classical ZK?

Theorem [Barak-Lindell '02]: Constant-round black-box ZK is **impossible** with fixed polynomial-time simulation.

Solution [GMR85,GMW86,GK96,FS90]: **expected** polynomial-time simulation

$$\text{Sim Output} = \sum_t a_t M^{(t)}(x, st)$$

$$\sum_t a_t \cdot t = \text{poly}(\lambda)$$

In this "branch", $M(x, st)$ runs in time t .

Coherent-Runtime Expected Poly Time (EQPT_c)

$$\text{Output} = \sum_t \alpha_t (U^{(t)})^\dagger C(U^{(t)}) \cdot \Pi^{(t)} |\psi\rangle$$

On this state, running U for t steps will reach end.

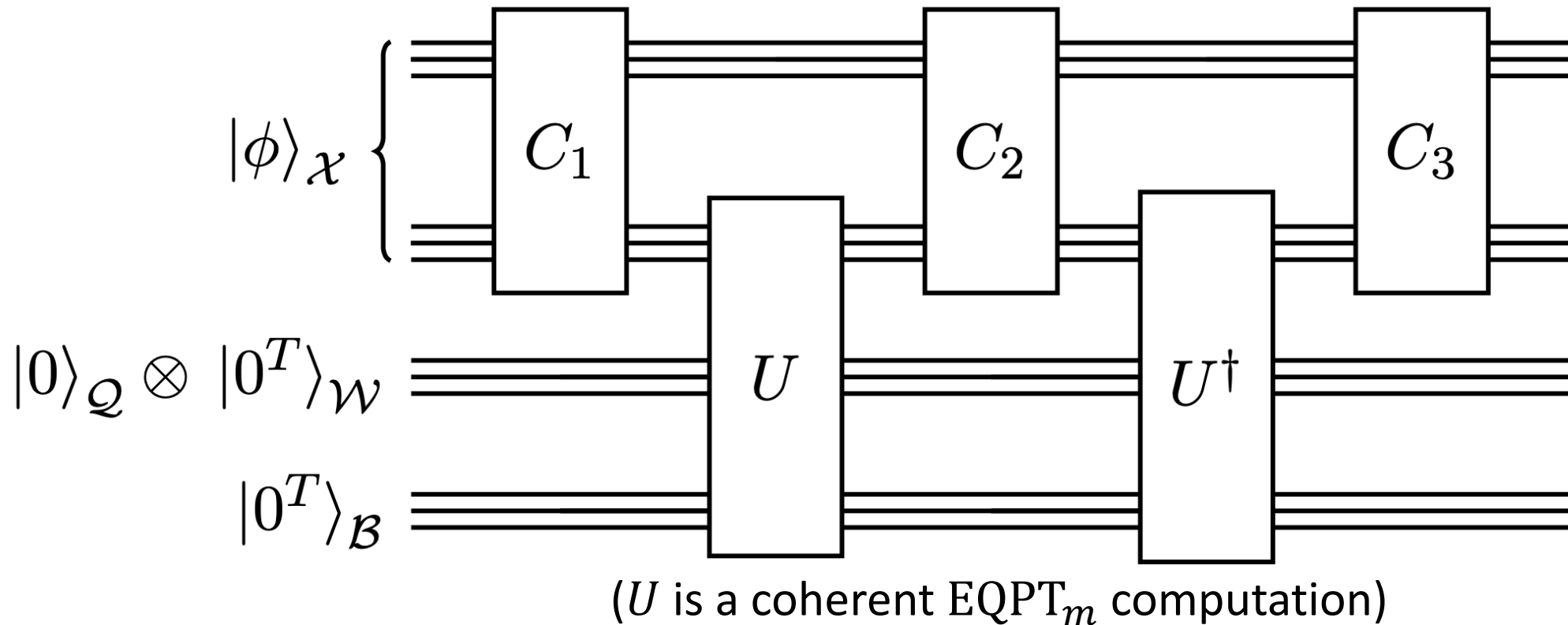
$$\sum_t |\alpha_t|^2 \cdot t = \text{poly}(\lambda)$$

In this "branch," both U and U^\dagger run for t steps.

Allows implementing projective measurements from EQPT_m procedures.

Coherent-Runtime Expected Poly Time (EQPT_c)

Make use of U^\dagger to uncompute the runtime.



Coherent-Runtime Expected Poly Time (\mathbf{EQPT}_c)

For the purposes of ZK simulation, \mathbf{EQPT}_c is as good as classical EPT:

- **Usefulness:** Captures all of our ZK simulators
- **Approximability:** Can be truncated to time $\text{poly}(\lambda, 1/\varepsilon)$ with ε error.
- **Doesn't solve hard problems:** If a computational assumption (e.g. LWE) is broken by \mathbf{EQPT}_c then it is broken by **BQP**.

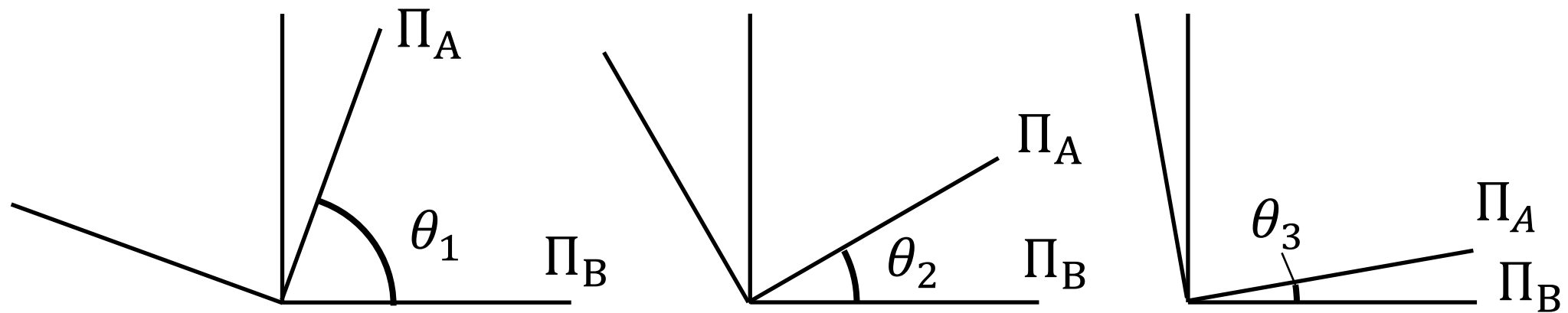
We propose \mathbf{EQPT}_c to be the “right” quantum analogue of classical EPT simulation.

Recap

- ZK Simulation by State-Preserving Extraction
 - **Guaranteed extraction** + collapsing \rightarrow state-preserving extraction
- Extraction with (fast) singular vector algorithms
 - Jordan singular value estimation + singular vector transform
 - Amplify **prover state onto accepting transcript**, measure transcript, and then repair. Appeal to pseudoinverse state to bound runtime.
- **Coherent-runtime** expected quantum polynomial time simulation

Final Thoughts

- Very open: conditions under which a cryptographic security property is **generically post-quantum**.
- **Singular vector algorithms** are good tools for crypto security proofs
 - Crypto also formulates interesting quantum algorithmic problems!
- **Collapsing** is an extremely important security notion
 - When are quantum phenomena collapsing? Current tools are limited
- What quantum computational models accurately characterize cryptographic security properties?



Thank you!

