Parallel Repetition for Post-Quantum Arguments

Andrew Huang
MIT CSAIL
ahuang@mit.edu

Yael Tauman Kalai
MIT CSAIL
tauman@mit.edu

June 2, 2025

Abstract

In this work, we show that parallel repetition of *public-coin* interactive arguments reduces the soundness error at an exponential rate even in the *post-quantum* setting. Moreover, we generalize this result to hold for threshold verifiers, where the parallel repeated verifier accepts if and only if at least t of the executions are accepted (for some threshold t). Prior to this work, these results were known only when the cheating prover was assumed to be classical.

We also prove a similar result for three-message *private-coin* arguments. Previously, Bostanci, Qian, Spooner, and Yuen (STOC 2024) proved such a parallel repetition result in the more general setting of quantum protocols, where the verifier and communication may be quantum. We consider only protocols where the verifier is classical, but obtain a simplified analysis, and for the more general setting of threshold verifiers.

Contents

1 Introduction **1**
1.1 Our Results 2

2 Technical Overview **3**
2.1 Public-Coin Protocols 4
2.2 Private-Coin Protocols 6

3 Preliminaries **8**
3.1 Interactive Protocols with Computational Soundness 9
3.2 Value Estimation 10
3.3 Information Theory 11

4 A Memoryless CMSZ **11**
4.1 Distributional Memoryless CMSZ 12

5 Parallel Repetition for Public-Coin Protocols **18**
5.1 Our Results 18

6 Parallel Repetition for Three-Message Protocols **28**
6.1 Our Results 28

7 Acknowledgments **38**

1 Introduction

One of the most important techniques in the study of interactive proofs, probabilistically checkable proofs, and multi-prover interactive proofs is that of *soundness amplification*, where the goal is to improve soundness (i.e., reduce the probability that a malicious prover makes the verifier accept a false statement). This has also been studied in the context of interactive arguments, which are interactive protocols that are guaranteed to be sound only against computationally bounded cheating provers. We focus on interactive arguments in the *post-quantum* setting, where the verifier (and thus the communication) remains classical but soundness is required to hold even against quantum adversaries of polynomial size.

One natural way to amplify soundness is via sequential repetition, namely, by repeating the protocol sequentially, where the verifier accepts only if all sequential executions are accepted. Indeed, this transformation is known to reduce the soundness error at an exponential rate, but it has the undesirable property of significantly increasing the round complexity of the original protocol. A preferable way to reduce the soundness error would be via *parallel repetition*, as it preserves the round complexity of the original protocol. However, quite surprisingly, in the context of interactive arguments, parallel repetition does not always reduce the soundness error [BIN97].

The classical setting. The question of parallel repetition has been studied quite extensively in the classical setting. It is a well-known fact that parallel repetition decreases the soundness error exponentially for proofs with statistical soundness [BM88, KW00]. However, for proofs with computational soundness, the situation is much more complicated.

The study of parallel repetition in the computationally sound setting dates back to the work of Yao [Yao82] on hardness amplification of one-way functions, which can be viewed as establishing that parallel repetition reduces the soundness error at an asymptotically optimal rate in every publicly-verifiable two-message argument. Later, Bellare, Impagliazzo, and Naor [BIN97] extended this result by proving that parallel repetition reduces the soundness error exponentially for any three-message interactive argument. Quite surprisingly, in that same paper, Bellare et al. also showed that this is not the case for four-message arguments, by constructing a family of four-message interactive arguments (assuming the existence of a non-malleable encryption scheme), where for each k, there exists a protocol whose k-fold parallel repetition has the same soundness error.[1] Loosely speaking, their interactive arguments are constructed to allow a cheating prover to coordinate among the different executions. More specifically, their cheating prover cheats by using the verifier's messages from other executions. Importantly, the verifier's messages are private-coin; otherwise they would be useless, as the cheating prover could have simulated them on her own.

Indeed, it was proven that for public-coin protocols,[2] and more generally for simulatable protocols,[3] parallel repetition reduces the soundness error exponentially [PV07, HPPW10, CL10, CP15]. All these works left open the following question: *for which multi-round private-coin protocols does parallel repetition reduce the soundness error, and at what rate?*

Haitner [Hai09] was the first to show a generic transformation that converts any interactive argument into one for which parallel repetition decreases the soundness error at an exponential rate.

[1] The communication complexity of a single-fold execution in their family of argument systems grows linearly with k. Later work by Pietrzak and Wikstrom [PW07] constructed an interactive argument (from standard computational assumptions) where the k-fold parallel repetition does not decrease the error probability below some constant for any polynomial k, and where the communication complexity of a single-fold execution does not depend on k.

[2] These are protocols where all the verifier's messages are computed by simply tossing random coins and sending the outcome.

[3] These are protocols where the verifier's messages are efficiently and computationally simulatable, but the verifier's verdict may still require private randomness.

Moreover, his transformation, known as *random-termination*, is extremely simple and preserves the round complexity of the original protocol. The transformation only slightly modifies the protocol by having the verifier terminate the protocol in each round with some small probability (in which case the verifier automatically accepts). Haitner proved that while this slightly increases the soundness error of the protocol, the soundness error decreases exponentially when repeated in parallel!

We emphasize that all the works mentioned above were in the classical setting and considered only classical cheating provers. Indeed, the proofs of these parallel repetition theorems proceeded by converting a cheating prover for the parallel-repeated protocol into a cheating prover for a single execution, by rewinding the parallel prover over and over again. We note that such (perfect) rewinding *cannot* be done in the quantum setting due to the no-cloning principle and the destructive nature of quantum measurements. Given the increasing progress towards building large-scale quantum computers, it is of utmost importance to ensure the security of our systems and soundness of our proofs against quantum cheating provers.

The quantum setting. A recent work by Bostanci, Qian, Spooner, and Yuen [BQSY24] considers the *quantum* setting where both the prover and verifier have quantum capabilities. They show that for three-message quantum argument systems, parallel repetition decreases soundness exponentially. They also demonstrate a round compression procedure which converts any quantum argument system with a polynomial number of rounds to a three-message, public-coin quantum argument system with polynomially-related soundness.

Importantly, this round compression procedure is unlikely to have a classical analogue, even for proof systems: while it is known that $\mathsf{QIP}(3) = \mathsf{QIP}$ [KW00], it is widely believed that $\mathsf{IP}(3) \neq \mathsf{IP}$, as otherwise $\mathsf{AM} = \mathsf{PSPACE}$, which implies the collapse of the polynomial hierarchy.

1.1 Our Results

In this work, we consider the *post-quantum* setting, where both the verifier and communication are restricted to being classical, but soundness is required to hold even against quantum cheating provers. We establish parallel repetition theorems for two classes of post-quantum interactive arguments.

First, we show that for public-coin protocols with an arbitrary polynomial number of rounds, parallel repetition decreases the soundness error at an exponential rate.

Theorem 1.1 (Informal)**.** *For any public-coin m-message protocol with soundness error s, its k-fold parallel repetition has soundness error $\leq f(s)^{O(k/m^2)} + \mathsf{negl}(\lambda)$, where $f(s) \approx 2^{-(1-s)^2}$.*

Our result is lossy in two aspects: first, the soundness error goes down exponentially in k/m^2 as opposed to k. Secondly, the base is $f(s)$ and opposed to s. We note that a similar soundness loss was given in the classical setting in [HPPW10] and [CP15]. We generalize Theorem 1.1 by considering the setting of threshold verifiers, who accept if the number of accepting transcripts among all executions exceeds some threshold.

Theorem 1.2 (Informal)**.** *For any public-coin m-message protocol with soundness error s, its k-fold parallel repetition with threshold $0 \leq t \leq k$ has soundness error $\leq f(s)^{O(k/m^2)} + \mathsf{negl}(\lambda)$, where $f(s) \approx 2^{-(t/k-s)^2}$.*

Second, we show a parallel repetition theorem for any three-message (possibly private-coin) protocol.

Theorem 1.3 (Informal)**.** *For any* 3*-message protocol with soundness error* s*, its* k*-fold parallel repetition has soundness error* $\leq f(s)^{O(k)} + \mathsf{negl}(\lambda)$*, where* $f(s) \approx 2^{-(1-o_k(1)-s)^2}$*.*[4]

We also generalize Theorem 1.3 to hold for threshold verifiers.

Theorem 1.4 (Informal)**.** *For any* 3*-message protocol with soundness error* s*, the* k*-fold parallel repetition with threshold* $0 \leq t \leq k$ *has soundness error* $\leq f(s)^{O(k)} + \mathsf{negl}(\lambda)$*, where* $f(s) \approx 2^{-(t/k-o_k(1)-s)^2}$*.*

Remark 1.5. Our proof is *mildly* non-black-box since we assume knowledge of the cheating prover's size. Our proof is also mildly non-constructive since we use multiple copies of the prover's original state. This latter property is known to be inherent for any reduction that needs to succeed with higher probability than the underlying prover (see for example [BQSY24]). As in [BQSY24], given a state which is already in a "good subspace", we only need one such copy for the reduction to run.

Open problems. We leave open the question of whether a fully black-box proof can be used to establish parallel repetition in the public-coin setting.[5] We also leave open the question of removing the round dependence from our soundness error in Theorems 1.1 and 1.2. This was done in the classical setting in [CP15] by analyzing relative entropy; we note that the direct quantum analogue of the classical chain rule does not hold for quantum relative entropy.[6]

Finally, and most interestingly, we leave open the question of obtaining a post-quantum parallel repetition theorem for private-coin protocols. We know that in order to obtain such a result, one would need to modify the underlying protocol, for example by using the random-termination transformation [Hai09]. Our techniques do not seem to extend to the multi-round private-coin setting (see Section 2 for a discussion).

2 Technical Overview

As mentioned in the introduction, the main challenge in the post-quantum (and quantum) setting lies in the fact that most classical reductions for proving parallel repetition inherently assume the ability to perfectly rewind a prover to its previous state. In the setting where the prover's state is quantum, this is impossible to implement efficiently due to the potentially destructive nature of measurement and the no-cloning principle.

Our first approach to bypass this problem is to use a technique introduced in [CMSZ22], which allows us to rewind the cheating prover to a state that is still "useful" in the sense that it can be used to cheat in the argument system. However, this state is not identical to the original state and the only guarantee we have is that the state is still a "good" state in the sense that it is accepting with sufficiently high probability. Unfortunately, this alone turns out not to be enough. As we elaborate below, the reason is that this "rewound" state may record information about past queries used by the reduction, and may deliberately fail on these queries. To make the analysis work, we use a technique called *flooding* [BBK22] which makes non-black box use of the cheating prover.

[4] It should be noted that [BQSY24] provides a tighter result in the three-message setting. We give a different proof of parallel repetition which extends to the setting of threshold verifiers when both the verifier and communication are classical.

[5] The proof in [BQSY24] for three-message protocols is black-box in this sense, but it is unclear how to extend it to the multi-round, post-quantum setting.

[6] In fact, there are explicit counterexamples to the classical chain rule when directly extended to the setting of quantum channels, and only a modified chain rule is known [FFRS20].

Throughout this overview, we focus on the problem of parallel repetition without a threshold (i.e., where the threshold is $t = k$) since this is where our conceptual ideas lie. We start by explaining our techniques for public-coin protocols.

2.1 Public-Coin Protocols

We show that for public-coin protocols, parallel repetition reduces the soundness at a (weakly) exponential rate even in the post-quantum setting; this is done by means of a reduction.

We assume that there is a QPT adversary $\mathcal{B}$ that succeeds in the k-fold repetition setting with probability $\zeta > 0$, and we construct a QPT adversary $\mathcal{A}$ which makes black-box calls to $\mathcal{B}$ and succeeds in a single protocol with probability roughly $1 - O\left(m\sqrt{\frac{-\log \zeta}{k}}\right)$, where m is the number of messages sent in the protocol (see Theorem 5.1). This implies that if the original protocol has constant post-quantum soundness then its k-fold parallel repetition has post-quantum soundness $\zeta = \exp\big(-O(k/m^2)\big)$.

The classical setting. Let us start by recalling the analysis in the classical setting (specifically from the works of [PV07, HPPW10]), and see where it breaks down in the post-quantum setting. In the classical analysis, $\mathcal{A}$ first picks a random execution $i \in [k]$. The basic idea now is for $\mathcal{A}$ to forward the verifier's queries to the k-fold adversary $\mathcal{B}$, by embedding in the ith execution the query it received from the external verifier, and simulating the queries for all other executions on its own. Namely, in round j, upon receiving a query q_j from the external verifier, $\mathcal{A}$ chooses a random vector $\bar{q}_j$ such that $\bar{q}_j^i = q_j$.[7] Then, when $\mathcal{A}$ receives answers $\bar{z}_j$ from $\mathcal{B}$, she checks whether a random continuation of the transcript is accepting. If this continuation is accepting, then she forwards the answer $\bar{z}_j^i$ to the verifier; otherwise she rewinds the prover $\mathcal{B}$ and tries again with fresh $\bar{q}_j$ such that $\bar{q}_j^i = q_j$. This method of picking the first successful random continuation can be thought of as a noisy measurement of the prover's residual success probability; namely, if the random continuation is successful, then the prover has likely found a "good" response.

The classical analysis then compares this reduction (which we label as Real) with an experiment (labeled Ideal) which simulates all verifiers on its own, conditioned on a random continuation being accepting. Raz's lemma (Lemma 3.7, [Raz95]) is used to argue that these two experiments are close: in particular, one can show that the distribution of $\bar{q}$ "hides" the location of the embedded execution. Specifically, Raz's lemma asserts that

$$\mathbb{E}_{i \leftarrow [k], q_j}[\mathsf{TD}((i, (\bar{q}_j \mid W)), (i, (\bar{q}_j \mid W, \bar{q}_j^i = q_j)))] \leq \sqrt{\frac{-\log(\Pr[W])}{k}},$$

where W is the event that the random continuation is accepting.

Thus, after m rounds, the expected statistical distance between the two experiments will grow to roughly $m\sqrt{\frac{-\log \zeta}{k}}$, which is why our adversary $\mathcal{A}$ for the single execution ultimately succeeds with probability $1 - O\left(m\sqrt{\frac{-\log \zeta}{k}}\right)$ (assuming the Ideal experiment succeeds with high probability). We note that the use of Raz's lemma is precisely why our soundness amplification is lossy.[8]

[7] From now on, we will use superscripts to refer to executions and subscripts to refer to rounds.

[8] Indeed, in the classical setting, Chung and Pass [CP15] provide an alternative analysis that avoids this loss by using KL-divergence instead of Raz's lemma. We leave it as an open problem to extend the analysis of [CP15] to the post-quantum setting.

First challenge in the quantum setting. The first issue that we are faced with in the post-quantum setting is that perfect rewinding is impossible due to the potentially destructive nature of measurement and the no-cloning principle. We overcome this barrier by utilizing a technique from [CMSZ22] that shows that if we start with a state that has acceptance probability ζ, and we measure only a few bits, say whether the prover's response $\bar{z}_j$ for a query $\bar{q}_j$ is "good", then while we cannot rewind back to the state we started with, we can go back to a different state that still has acceptance probability at least $\zeta - \varepsilon$ for any small ε. The runtime of this "repairing procedure" grows polynomially with $1/\varepsilon$.

Remark 2.1. The technique of [CMSZ22] was originally designed for the single-round public-coin setting, but can be easily generalized to the multi-round setting. Moreover, it can further be generalized to the private-coin setting, as long as the protocol is computationally simulatable and the verifier's verdict can be efficiently estimated. This latter extension will be useful to obtain parallel repetition for 3-round private-coin protocols. See Section 2.2 for details.

At this point, it is tempting to think that our ability to "rewind" the prover's state implies that we can use an analysis similar to that in the classical setting. However, there is still a major obstacle that differentiates the classical setting from the quantum/post-quantum settings.

Second challenge in the quantum setting. While in the classical setting, one can rewind the prover back to exactly the same state as before, this is not true in the quantum setting: although one can rewind to a state that has acceptance probability close to ζ, this state may be very different from the original state of the prover. In fact, it may (in theory) crucially remember information about $\bar{q}_j$ and i. If it does, then the prover may choose to perform poorly on any input $\bar{q}'_j$ such that $(\bar{q}'_j)^i = \bar{q}^i_j = q_j$ (where q_j is the query sent by the external verifier), while at the same time having an acceptance probability close to ζ on uniformly random queries.

Making the rewound state forgetful. Our strategy will be to force the rewound state to "forget" information about $\bar{q}_j$ (and in particular about the external verifier's query) by using a technique due to [BBK22] called *flooding*. The basic idea is to flood the cheating prover $\mathcal{B}$ with many dummy queries and then argue that $\mathcal{B}$ cannot remember the embedded query from the external verifier since it does not have enough memory to store all of the queries. The number of dummy queries we need to flood the state with depends on the size of the memory (i.e., number of qubits) that the prover has: if it has ℓ qubits, then the number of dummy queries will scale with ℓ. In particular, this technique is (mildly) non-black-box, and is the only non-black-box part of our reduction.

We formalize this by presenting a new "memoryless" version of the [CMSZ22] lemma, which provides a repairing technique that, while still not rewinding the state to the exact state it started with (which is inherent since performing a measurement can irreversibly destroy information), can rewind the state to one that is almost "independent" of the projection applied to the state prior to the rewinding. In other words, we ensure that the state "forgets" the projection. We believe that this lemma (formalized in Lemma 4.1) is of independent interest and will prove useful for other applications as well. We refer the reader to Section 4 for details.

The final reduction, outlined. Roughly speaking, our final reduction proceeds as follows:

1. Estimate the success probability of the k-fold cheating prover $\mathcal{B}$ given an initial state $|\psi\rangle$. With sufficiently (polynomially) many copies of $|\psi\rangle$, one can assume that at least one copy will have a sufficiently good outcome, giving us a residual state in the desired "good" subspace.

2. Choose a random execution $i \leftarrow [k]$ to embed the external verifier's queries in.

3. For each round, upon receiving a query from the external verifier, we will try the following sufficiently (polynomially) many times:

   (a) Generate a k-fold query $\bar{q}$ where we embed the external verifier's query in the ith coordinate and simulate the rest of the verifiers internally.

   (b) Test to see if $\bar{q}$ is a "good" query by computing $\mathcal{B}$'s response to $\bar{q}$ coherently and then running the projection which estimates whether or not the residual success probability of the prover is still high.

   (c) i. If it is, then measure the response, send the answer located in the ith coordinate to the external verifier, and proceed to the next round.

   ii. Otherwise, we use our memoryless repair procedure to repair the state back to one which is still good over random queries and has almost no memory of the previous query $\bar{q}$.

4. Conclude when a complete transcript has been formed.

As mentioned above, we instantiate these estimation and memoryless repair procedures using the algorithms given by [CMSZ22], and tweaking them using ideas from [BBK22] to create "memoryless" versions of these procedures. This allows us to test each embedded query many times, since to the prover each test looks nearly indistinguishable from a random query.

2.2 Private-Coin Protocols

Next, we move to consider the private-coin setting. The first problem that arises is that we can no longer estimate the success probability of a cheating prover since computing the verifier's verdict may require knowing the verifier's private coins. This problem also appears in the classical setting; indeed, it was shown [BIN97] that for private-coin protocols, parallel repetition does not always reduce the soundness error! The issue in the private-coin setting is that the prover may be able to use the verifier's messages from different executions to cheat, since these may actually contain useful information, as opposed to simply consisting of random coins.

Three-message protocols. Luckily, it turns out that such a counterexample does not exist for the special case of three-message protocols. Indeed, the works of [BIN97], [CHS05], and [HPPW10] demonstrated that in the classical setting, all three-message (including private-coin) protocols have exponentially decreasing soundness under parallel repetition. Recently, the work of [BQSY24] extended this result to the quantum (and post-quantum) settings.

In this work, we reprove this result in the post-quantum setting, but through a different framework that allows us to directly generalize to threshold-type verifiers. We take a similar approach to [BIN97] and [HPPW10], which implement probability estimation via the *soft-decision* approach, which we elaborate on below. We note that [BQSY24] more closely mirrors the *correlation reduction* approach taken by [CHS05], offering a tight (but more complicated) analysis. Our reduction in the three-message setting follows the same framework as the public-coin reduction with a simple substitution of the probability estimation procedure, again using flooding to make the quantum state "forgetful", allowing us to extend our results to the case of threshold-type verifiers.

Remark 2.2. In the three-message setting, one can achieve a straightforward black-box proof, avoiding the need to flood, but at the expense of the reduction being significantly more non-constructive. This is the case both in the post-quantum and quantum settings. Intuitively, one

can take a prover that has access to many copies of the original prover's residual state *after a good first response has already been measured*. The downside of this approach is that even if the original prover's starting state was efficiently preparable, its post-measurement residual state is unlikely to be efficiently preparable if the response space is super-polynomially large.

The soft-decision approach. Let us recall the soft-decision approach of [BIN97] and [HPPW10]. In the private-coin setting, a major issue is the problem of estimating the success probability of the prover without the external verifier's randomness. A naive first attempt to determining whether a prover's response is good (in the sense that conditioned on this response, the prover's residual success probability is relatively high) might be to assume the external coordinate is good if all other coordinates are good as well. However, this intuitive approach does not quite work: consider the following example first given by Chung [Chu11] of a malicious prover $\mathcal{B}$ for the k-parallel-repeated protocol that will foil such a reduction. Suppose $\mathcal{B}$ interacts with the parallel verifier in such a way that it is accepted by all executions with probability δ^k, and for each $j \in [k]$, is accepted by all executions except the jth execution with probability $(1-\delta^k)/k$. Intuitively, conditioned on all but the ith coordinate accepting, $\mathcal{B}$ may actually only succeed on the ith coordinate with probability $\frac{\delta^k}{\delta^k+(1-\delta)^k/k} \ll \delta$. Such a prover can be considered to be one that has "bad correlations".

In order to deal with provers with this behavior, classical analyses have generally taken two approaches, the correlation reduction approach of [CHS05] (which is roughly the approach taken by [BQSY24]) and the soft-decision approach of [BIN97] and [HPPW10] (which is the approach we take in this work). Instead of taking only responses for which *all* executions, except the external execution, accept, we weigh our decision to take a response with probability exponentially proportional to the number of rejecting executions outside the external execution. That is, if z executions excluding the embedded one are rejecting, then we will accept the response with probability roughly 2^{-z}. Looking at the prover in our example, it is easy to see that this soft-decision approach will choose answers that are rejected on coordinate i only with probability $O(1/k)$. In general, this intuition explains how the soft-decision approach works to "smoothen" the distribution of errors made by the reduction. This approach also almost immediately generalizes to the setting of threshold verifiers, where we can set our acceptance probability to be roughly $\min\{1, 2^{k-t-z}\}$, where t is the threshold of the parallel verifier.[9]

Challenges in the quantum setting. The first challenge, which we encountered in the public-coin setting as well, is that perfect rewinding cannot be done in the quantum setting. As in the public-coin setting, we use [CMSZ22] to repair. As mentioned above, we cannot rewind to the exact same state, and we can only guarantee that we repair to a state that is still successful, i.e., succeeds on a random continuation with sufficiently high probability. As mentioned in Remark 2.1, the technique of [CMSZ22], which was originally designed for the single-round public-coin setting, can be easily extended to multiple rounds, as long as the protocol is computationally simulatable and the verifier's verdict can be efficiently estimated. Here, we run into an issue which is common to the classical setting as well – given the prefix of a transcript, how can we estimate if a continuation is accepting without access to the external verifier's randomness?

In the three-message setting, this is not a problem, since if the prefix consists of only the first prover message, then one can perfectly simulate a full transcript and compute the verifier's verdict, and if the prefix consists of a full transcript then one can implement the soft-decision approach to estimate the prover's success probability (see Remark 3.5 for more details).

[9] We note that in the classical setting, the soft-decision approach is known to be inherently lossy compared to the correlation reduction approach.

The fact that this rewinding is not perfect also brings with it challenges which we first encountered in the public-coin setting. Specifically, the state can depend on its projection history, and in particular can remember the query from the external verifier, and fail on that specific query while succeeding on random queries. In the public-coin setting we solved this issue by flooding. In fact, one can observe that flooding can always be applied to the first two messages of the protocol, which suffices for the three-message setting. The reason is that whenever we flood, we need to be able to sample the external verifier's query (and its private randomness) from the conditional distribution given the transcript prefix, but for the first two messages this prefix is empty, so in this case we can efficiently sample from the randomness distribution.

Multi-round protocols. Unfortunately, in the multi-round, private-coin setting (where the number of messages sent is more than three) we are unable to prove a parallel repetition theorem in the post-quantum setting. The first obstacle that arises is that after the first verifier's message is sent, it may be hard to simulate the verifier's subsequent messages without access to the private randomness of the verifier. This is a challenge that occurred even in the classical setting (where random termination enabled the overcoming of this problem).

In the post-quantum setting, even if the protocol is computationally simulatable, we still do not know how to prove a parallel repetition theorem. The problem is that every time we estimate the success probability of the prover we must do so without knowing the embedded execution's final verdict, and in the post-quantum setting, the state may remember that the embedded execution, denoted by i, is not being correctly estimated, and once it remembers i, it can behave in a way so that only the i'th execution (which has a fixed query given by the external verifier) will be rejected.

With both public-coin and three-message protocols, the issue of the state remembering i arose, but we were able to make the state "forgetful" by flooding. In the public-coin case, simulation of messages and verdict is trivial, and in the three-message case, simulation of the messages and verdict is possible since there is only one verifier message. However, in the multi-round setting, it is apparent that we cannot flood, since we cannot simulate the messages of the verifier and its verdict, after the first message has been fixed.

Indeed, our proofs in the public-coin and three-message settings rely substantially on the ability to flood, and this inability to flood in the general private-coin setting is one major obstacle to proving a parallel repetition theorem with the "estimate and repair" approach.

A note on flooding. We note that the key property required to apply the flooding technique is the ability to simulate the messages and the verdict of the external verifier given a prefix of the transcript. This allows us to show similar parallel repetition results for protocols which are public-coin except for (possibly) the *last* round, which includes both the public-coin and three-message settings. As many protocols for classical verification of quantum computation are public-coin except for the *first* round, determining if such protocols have exponentially decreasing soundness under parallel repetition seems to be one potential extension of interest.[10]

3 Preliminaries

We use PPT as a shorthand for probabilistic polynomial time and QPT as a shorthand for quantum polynomial time.

[10] We note that even a restriction to such protocols with only four messages would already have useful implications.

Notation 1. For any two quantum states $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$, we use the notations

$$\boldsymbol{\rho}_1 \stackrel{\varepsilon}{\equiv} \boldsymbol{\rho}_2 \quad \text{and} \quad \mathsf{TD}(\boldsymbol{\rho}_1, \boldsymbol{\rho}_2) \leq \varepsilon$$

to denote that the trace distance between $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$ is at most ε. In particular, this means that for any quantum adversary $\mathcal{A}$,

$$|\Pr[\mathcal{A}(\boldsymbol{\rho}_1) = 1] - \Pr[\mathcal{A}(\boldsymbol{\rho}_2) = 1]| \leq \varepsilon.$$

Definition 1 (Measurement)**.** A *real-valued measurement* M is a quantum circuit that takes as input a quantum state $\boldsymbol{\rho}$ and outputs a quantum state $\boldsymbol{\rho}^*$ and a real number p^*.

Definition 2 ((ε, δ)-Almost Projective Measurement, [CMSZ22])**.** A real-valued measurement M on a Hilbert space $\mathcal{H}$ is said to be (ε, δ)-*almost projective* if on input a quantum state $\boldsymbol{\rho} \in S(\mathcal{H})$, M satisfies

$$\Pr\left[|p^* - p^{**}| \geq \varepsilon \;\middle|\; \begin{array}{l} (\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{M}(\boldsymbol{\rho}) \\ (\boldsymbol{\rho^{**}}, p^{**}) \leftarrow \mathsf{M}(\boldsymbol{\rho^*}) \end{array}\right] \leq \delta.$$

We next define a family of measurements that are (ε, δ)-almost projective for arbitrary $\varepsilon, \delta > 0$.

Definition 3 (Almost Projective Measurement)**.** A real-valued measurement family $\mathsf{M} = \{\mathsf{M}_{\varepsilon,\delta}\}_{\varepsilon,\delta>0}$ on $\mathcal{H}$ is said to be *almost projective* if for all quantum states $\boldsymbol{\rho} \in S(\mathcal{H})$ and all parameters $\varepsilon, \varepsilon', \delta, \delta' \in (0, 1]$,

$$\Pr\left[|p^* - p^{**}| \geq \max\{\varepsilon, \varepsilon'\} \;\middle|\; \begin{array}{l} (\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\rho}) \\ (\boldsymbol{\rho^{**}}, p^{**}) \leftarrow \mathsf{M}_{\varepsilon',\delta'}(\boldsymbol{\rho^*}) \end{array}\right] \leq \max\{\delta, \delta'\}.$$

In particular, for all $\varepsilon, \delta > 0$, $\mathsf{M}_{\varepsilon,\delta}$ is an (ε, δ)-almost projective measurement.

We will use the well-known fact that any quantum measurement performed by a quantum circuit can be implemented projectively by deferring measurement. That is, any quantum circuit C can be equivalently implemented by applying a unitary dilation U (corresponding to replacing intermediate measurements with CNOT gates) followed by a measurement of the output register and an application of $U^\dagger$.

3.1 Interactive Protocols with Computational Soundness

In what follows, we focus on interactive protocols between two parties, where one party, denoted by V (for verifier), outputs a single bit, indicating acceptance or rejection. The other party, denoted by P (for prover), tries to convince the verifier to accept. We define the notion of computational soundness for such protocols. For the sake of simplicity, we focus on the case where the parties receive as input the security parameter 1^λ, and omit all additional inputs. We denote by

$$\langle (P, V)(1^\lambda) \rangle$$

the output bit of V after interacting with P.

Definition 4. An interactive protocol between parties P and V, where V outputs a single bit is said to have (computational) soundness ε for any polynomial-size cheating prover $\mathcal{A}$ there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$

$$\Pr\Big[\langle (\mathcal{A}, V)(1^\lambda) \rangle = 1\Big] \leq \varepsilon + \mu(\lambda).$$

Interactive protocols with computational soundness are also referred to as interactive arguments, and we will use these terms interchangeably. A public-coin interactive argument is a protocol where the verifier's messages to the prover consist entirely of random coins, and the verifier's verdict is deterministic and publicly computable given a complete transcript. We consider only protocols where the communication between the prover and verifier is classical and the verifier is a classical PPT algorithm.

Remark 3.1. We assume without loss of generality that the last message of any protocol is sent from the prover to the verifier. We call an interactive protocol an m-round protocol if it consists of m rounds of back and forth messages between the prover and verifier (as opposed to the alternative $2m$-round definition). For the sake of simplicity and without loss of generality, we also assume that the first message is sent by the verifier (unless we are in the three-message setting), as otherwise the verifier can always begin by sending coins that are ignored in the rest of the protocol.

Notation 2. The transcript of an m-round interactive protocol (P, V) consists of all messages sent between the prover and verifier, denoted by $\tau = (q_1, z_1, \ldots, q_m, z_m)$, where q_j is the query sent by V in the jth round, and z_j is the response sent by P in the jth round. In the case of three-message protocols, we instead use the notation $\tau = (z_1, q, z_2)$ to emphasize that only one query is sent by the verifier and that the first message is sent by the prover.

3.2 Value Estimation

We borrow the following lemma from [CMSZ22] (using the formalism from [LMS22]).

Lemma 3.2 (**Value Estimation and State Repair**, [CMSZ22, LMS22])**.** *Let $\mathcal{H}$ be a Hilbert space. There exist quantum algorithms:*

1. $(\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{ValEst}_{V,A}(\boldsymbol{\rho}, \varepsilon, \delta)$ *is given black-box access to a verifier circuit $V : \{0,1\}^d \times \{0,1\}^n \to \{0,1\}$ and a quantum circuit A, and on input a quantum state $\boldsymbol{\rho} \in S(\mathcal{H})$ and accuracy parameters $\varepsilon, \delta \in (0,1]$, outputs a quantum state $\boldsymbol{\rho}^* \in S(\mathcal{H})$ and value $p^* \in X \subseteq [-\frac{1}{2}, \frac{3}{2}]$ for some discrete set X where $N_{\varepsilon,\delta} := |X| = O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$.*[11]

2. $\boldsymbol{\sigma}^* \leftarrow \mathsf{Repair}_{\mathsf{M},\Pi}(\boldsymbol{\sigma}, y, p, \eta)$ *is given black-box access to an (ε, δ)-almost projective measurement M and a projective measurement $\Pi = (\Pi_y)_{y \in Y}$ on $\mathcal{H}$, and on input a quantum state $\boldsymbol{\sigma} \in S(\mathcal{H})$, an outcome $y \in Y$, a probability $p \in [0,1]$, and parameter $\eta \in (0,1]$, outputs a quantum state $\boldsymbol{\sigma}^* \in S(\mathcal{H})$.*

These algorithms satisfy the following guarantees:

1. ***Value Estimation:***

$$\mathbb{E}\left[p^* \mid (\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{ValEst}_{V,A}(\boldsymbol{\rho}, \varepsilon, \delta)\right] = \Pr\left[V(r,z) = 1 \;\middle|\; \begin{array}{l} r \leftarrow \{0,1\}^d \\ z \leftarrow A(\boldsymbol{\rho}, r) \end{array}\right].$$

2. $\mathsf{ValEst}^{V,A}_{\varepsilon,\delta} := \mathsf{ValEst}_{V,A}(\cdot, \varepsilon, \delta)$ ***is an Almost Projective Family:***

$$\Pr\left[|p^* - p^{**}| \geq \max\{\varepsilon, \varepsilon'\} \;\middle|\; \begin{array}{l} (\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{ValEst}_{V,A}(\boldsymbol{\rho}, \varepsilon, \delta) \\ (\boldsymbol{\rho}^{**}, p^{**}) \leftarrow \mathsf{ValEst}_{V,A}(\boldsymbol{\rho}^*, \varepsilon', \delta') \end{array}\right] \leq \max\{\delta, \delta'\}.$$

[11] The reader should think of p^* as a noisy estimate of the success probability, whose expectation is in $[0,1]$.

3. ***Repairing:*** *For any (ε, δ)-almost projective measurement M on $\mathcal{H}$,*

$$\Pr\left[|p^* - p^{**}| \geq 2\varepsilon \;\middle|\; \begin{array}{l} (\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{M}(\boldsymbol{\rho}) \\ (\boldsymbol{\sigma}, y) \leftarrow \Pi(\boldsymbol{\rho}^*) \\ \boldsymbol{\sigma}^* \leftarrow \mathsf{Repair}_{\mathsf{M},\Pi}(\boldsymbol{\sigma}, y, p^*, \eta) \\ (\boldsymbol{\rho}^{**}, p^{**}) \leftarrow \mathsf{M}(\boldsymbol{\sigma}^*) \end{array}\right] \leq N \cdot (\eta + \delta) + 4\sqrt{\delta},$$

where $N = |Y|$ is the number of outcomes the projective measurement Π can obtain.

ValEst *and* Repair *are oracle circuits with $O(|X|) = O\left(\frac{\log 1/\delta}{\varepsilon^2}\right)$ and $O(1/\eta)$ gates, respectively.*

Remark 3.3 (Computing ValEst coherently)**.** We note that by deferring measurements, we can coherently compute ValEst over some register using ancilla registers.

Remark 3.4 (Randomness in ValEst)**.** Note that even when the prover strategy has a fixed success probability of precisely 0 or 1, the ValEst procedure may alter the prover's state. Thus, in the case where the circuits A and V are trivial (i.e. when $d = n = 0$), we define ValEst to be the *deterministic* algorithm which simply returns the output of V and does not modify the quantum state. This extension/modification of ValEst clearly still satisfies properties 1 and 2.

Remark 3.5 (Computing queries in ValEst)**.** While the ValEst procedure was originally defined in the context of single-round, public-coin protocols, we can easily generalize it to the multi-round setting (as was done in [LMS22]) by having the randomness of V specify all of the verifier's randomness in the interactive protocol. With some slight modifications, one can also extend ValEst to work in the private-coin (three-message) setting by considering a restricted adversary which separately computes the verifier's query before applying a specified unitary corresponding to the prover's strategy (see Remark 6.3 for more details). This allows us to efficiently simulate random executions and estimate the prover's success probability.

3.3 Information Theory

Lemma 3.6 ([BBK22])**.** *Let $\vec{Y} = (Y_1, \ldots, Y_t)$ be a joint distribution over t classical random variables. Let $\vec{y}$ be distributed according to $\vec{Y}$. Let $\boldsymbol{s}$ be an ℓ-qubit random variable that has arbitrary dependence on $\vec{y}$. We let $\vec{y}_i$ denote the prefix $\vec{y}_i = (y_1, \ldots, y_i)$ for $1 \leq i \leq t$, and let $\vec{y}_0$ be the empty vector (and likewise for $\vec{Y}$). Let J be the uniform distribution over $[t]$ and let $j \leftarrow J$. Define $y' \leftarrow Y_j|(\vec{Y}_{j-1} = \vec{y}_{j-1})$. Then it holds that*

$$\mathsf{TD}((j, \vec{y}_{j-1}, y_j, \boldsymbol{s}), (j, \vec{y}_{j-1}, y', \boldsymbol{s})) \leq \sqrt{\ell/(2t)}.$$

Lemma 3.7 (Raz's Lemma, [Raz95, Hol07, Hai09])**.** *Let $X_1, \ldots, X_k$ be independent random variables over some probability space, and let $\overline{X} = (X_1, \ldots, X_k)$. Let W be a non-empty event on the same space. Then*

$$\mathbb{E}_{i \leftarrow [k]}[\mathsf{TD}((i, (\overline{X}|W)), (i, (\overline{X}|W, X_i)))] \leq \sqrt{-\log(\Pr[W])/k}.$$

4 A Memoryless CMSZ

The work of [CMSZ22] (Lemma 3.2) provides a technique for running a quantum adversary $\mathcal{A}$ interacting with a verifier V, measuring its response z, and then repairing its state to one that is

still accepted by V with roughly the same probability as the original state, where the runtime of the repair procedure grows polynomially with the number of possible prover responses. This is a powerful technique that allows us to rewind in the post-quantum setting. However, this rewinding is different from the rewinding in the classical setting in that the repaired state may be significantly different from the original state, and in particular may contain information about queries on which the adversary $\mathcal{A}$ was run. This makes it problematic to apply this technique to lift reductions to the post-quantum setting, and for parallel repetition in particular.

Specifically, in the analysis of parallel repetition, we reduce a cheating prover $\mathcal{B}$ for the k-fold repetition to a cheating prover $\mathcal{A}$ for a single execution. In a nutshell, $\mathcal{A}$ interacts with a verifier, it receives a query q, embeds this query in a random execution $i \leftarrow [k]$, and chooses the queries for the rest of the executions on its own. It sends these k queries to $\mathcal{B}$, computes the response of $\mathcal{B}$ (coherently) and estimates its success probability. If the success probability is not sufficiently high then it applies the Repair procedure and does this again with fresh queries in all executions except the embedded one. The issue is that the repaired state may remember information about the k queries on which it was run, and eventually remember the embedded query q, and be repaired to a state that generates accepted answers with high probability but always generates a rejected answer on the embedded query q. See Section 2.1 for a detailed discussion.

In this section we show how to make this Repair procedure *forgetful*, so that the repaired state does not have information about the embedded query. This requires the forgetful Repair procedure to have a bound on the size of the quantum state of $\mathcal{A}$, and thus is (mildly) non-black-box.

4.1 Distributional Memoryless CMSZ

The idea for making the Repair procedure from Lemma 3.2 forgetful is to run the adversary, followed by running the Repair procedure, sufficiently many times on dummy queries, so that the state will not be able to remember all of these queries due to lack of space. If the distribution of the dummy queries is the same as the distribution of the real query then the state should not be able to distinguish the real query from the dummy ones, and in this way, we control the information that can be remembered about the real query by the number of dummy queries we flood with.

We formalize this by defining two algorithms: $\mathsf{Prepare}$ and Repair' (to distinguish it from Repair from Lemma 3.2). The algorithm $\mathsf{Prepare}$ starts the flooding process. It is given black-box access to the family of projective measurements, denoted by $\mathcal{P}$, as well as access to an almost projective family $\mathsf{M} = \{\mathsf{M}_{\varepsilon,\delta}\}$. Intuitively, $\mathsf{Prepare}$ repeats the following "sufficiently many times," where the number of times is a random variable that depends on the size of the quantum state (denoted by ℓ).

1. Apply M to the state to obtain a value p'.
2. Choose at random a projection $\Pi \leftarrow \mathcal{P}$ and apply Π to the state.
3. Apply Repair to the state, which repairs the state back to a state for which M would give a value close to p' with high probability.

The Repair' algorithm first applies Repair to the state and then continues the flooding process, exactly as was done by $\mathsf{Prepare}$ in steps (1)-(3) above. The guarantees are formalized in the following lemma.

Lemma 4.1. *Let $\mathcal{H}$ be a Hilbert space. There exist quantum algorithms:*

1. $(\boldsymbol{\rho}', p') \leftarrow \mathsf{Prepare}_{\mathsf{M},\mathcal{P}}(\boldsymbol{\rho}, \varepsilon, \delta, \eta, \ell)$ *is given black-box access to an almost projective measurement family* M *and to a family of projective measurements* $\mathcal{P}$ *associated with a set of* N *outcomes in* $Y = \{y_1, \ldots, y_N\}$*. It takes as input a state* $\boldsymbol{\rho} \in S(\mathcal{H})$ *and parameters* $\varepsilon, \delta, \eta \in (0, 1]$ *and* $\ell \in \mathbb{N}$*, and outputs a quantum state* $\boldsymbol{\rho}'$ *and a value* p'*.*

2. $\boldsymbol{\sigma}^* \leftarrow \mathsf{Repair}'_{\mathsf{M},\mathcal{P},\Pi}(\boldsymbol{\sigma}, y, p, \varepsilon, \delta, \eta, \ell)$ *is given black-box access to* M *and* $\mathcal{P}$ *as above as well as black-box access to a projection* $\Pi \in \mathcal{P}$*. It takes as input a quantum state* $\boldsymbol{\sigma} \in S(\mathcal{H})$*, an outcome* $y \in Y$*, a value* p*, and parameters* $\varepsilon, \delta, \eta \in (0, 1]$ *and* $\ell \in \mathbb{N}$*, and outputs a quantum state* $\boldsymbol{\sigma}^* \in S(\mathcal{H})$*.*

These algorithms satisfy the following guarantees:

1. ***Size:*** $\mathsf{Prepare}$ *and* Repair' *are oracle circuits of size* $\mathrm{poly}(\ell, 1/\eta)$ *where all oracle calls to* M *are with parameters* ε' *and* δ' *that are polynomially related to* $\varepsilon, \delta, \eta, 1/\ell$*.*

2. ***Preparing Preserves Functionality:*** *For every almost-projective measurement family* M *and every family of* N*-outcome projections* $\mathcal{P}$*,*

$$\Pr\left[|p^* - p'| \geq 4\varepsilon \;\middle|\; \begin{array}{l} (\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\rho}) \\ (\boldsymbol{\rho}', p') \leftarrow \mathsf{Prepare}_{\mathsf{M},\mathcal{P}}(\boldsymbol{\rho}^*, \varepsilon, \delta, \eta, \ell) \end{array}\right] \leq N \cdot (\eta + 4\delta).$$

   In addition,

$$\Pr\left[|p' - p^{**}| \geq \varepsilon \;\middle|\; \begin{array}{l} (\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\rho}) \\ (\boldsymbol{\rho}', p') \leftarrow \mathsf{Prepare}_{\mathsf{M},\mathcal{P}}(\boldsymbol{\rho}^*, \varepsilon, \delta, \eta, \ell) \\ (\boldsymbol{\rho}^{**}, p^{**}) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\rho}') \end{array}\right] \leq \delta.$$

3. ***Repairing is Functional but Forgetful:*** *For every almost-projective measurement family* M*, every family of* N*-outcome projections* $\mathcal{P}$*, and every* $\Pi \in \mathcal{P}$*,*

$$\Pr\left[|p^* - p^{**}| \geq 4\varepsilon \;\middle|\; \begin{array}{l} (\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\rho}) \\ (\boldsymbol{\rho}', p') \leftarrow \mathsf{Prepare}_{\mathsf{M},\mathcal{P}}(\boldsymbol{\rho}^*, \varepsilon, \delta, \eta, \ell) \\ (\boldsymbol{\sigma}, y) \leftarrow \Pi(\boldsymbol{\rho}') \\ \boldsymbol{\sigma}^* \leftarrow \mathsf{Repair}'_{\mathsf{M},\mathcal{P},\Pi}(\boldsymbol{\sigma}, y, p', \varepsilon, \delta, \eta, \ell) \\ (\boldsymbol{\rho}^{**}, p^{**}) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\sigma}^*) \end{array}\right] \leq N \cdot (\eta + 4\delta).$$

   Moreover, for every $\mathsf{M}, \mathcal{P}$ *as above, and for every* ℓ*-qubit state* $\boldsymbol{\rho}$ *it holds that*

$$\mathsf{TD}((\Pi, \boldsymbol{\sigma}^*_\Pi), (\Pi', \boldsymbol{\sigma}^*_\Pi)) \leq N \cdot \eta$$

   where $\Pi, \Pi' \leftarrow \mathcal{P}$ *and where the state* $\boldsymbol{\sigma}^*_\Pi$ *is generated as follows:*

   (a) Compute $(\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\rho})$*.*
   (b) Compute $(\boldsymbol{\rho}', p') \leftarrow \mathsf{Prepare}_{\mathsf{M},\mathcal{P}}(\boldsymbol{\rho}^*, \varepsilon, \delta, \eta, \ell)$*.*
   (c) Compute $(\boldsymbol{\sigma}, y) \leftarrow \Pi(\boldsymbol{\rho}')$*.*
   (d) Compute $\boldsymbol{\sigma}^*_\Pi \leftarrow \mathsf{Repair}'_{\mathsf{M},\mathcal{P},\Pi}(\boldsymbol{\sigma}, y, p', \varepsilon, \delta, \eta, \ell)$*.*
   (e) Output $\boldsymbol{\sigma}^*_\Pi$*.*

Proof. Let us define $\mathsf{Prepare}$ and Repair', where Repair is the procedure defined in Lemma 3.2:

Definition 5 ($\mathsf{Prepare}_{\mathsf{M},\mathcal{P}}$)**.** On input $(\boldsymbol{\rho}^*, \varepsilon, \delta, \eta, \ell)$:

1. Set $\boldsymbol{\sigma}'' = \boldsymbol{\rho}^*$ and $T = 4\ell/\eta^3$. Sample $t \leftarrow [T]$ uniformly. Define $\mathsf{M}' := \mathsf{M}_{\varepsilon/2T, \delta^2/64T^2}$.
2. For $i = 1$ to $t - 1$:
   (a) Compute $(\boldsymbol{\rho}', p') \leftarrow \mathsf{M}'(\boldsymbol{\sigma}'')$.
   (b) Sample $\Pi_i \leftarrow \mathcal{P}$ uniformly and compute $(\boldsymbol{\sigma}', y) \leftarrow \Pi_i(\boldsymbol{\rho}')$.
   (c) Compute $\boldsymbol{\sigma}'' \leftarrow \mathsf{Repair}_{\mathsf{M}',\Pi_i}(\boldsymbol{\sigma}', y, p', \eta/2T)$.
3. Compute $(\boldsymbol{\rho}', p') \leftarrow \mathsf{M}'(\boldsymbol{\sigma}'')$.
4. Return $(\boldsymbol{\rho}', p')$.

Definition 6 ($\mathsf{Repair}'_{\mathsf{M},\mathcal{P},\Pi}$)**.** On input $(\boldsymbol{\sigma}, y, p', \varepsilon, \delta, \eta, \ell)$:

1. Set $T = 4\ell/\eta^3$. Define $\mathsf{M}' := \mathsf{M}_{\varepsilon/2T, \delta^2/64T^2}$.
2. Compute $\boldsymbol{\sigma}^* \leftarrow \mathsf{Repair}_{\mathsf{M}',\Pi}(\boldsymbol{\sigma}, y, p', \eta/2T)$.
3. For $i = 1$ to T:
   (a) Compute $(\boldsymbol{\rho}, p^{**}) \leftarrow \mathsf{M}'(\boldsymbol{\sigma}^*)$.
   (b) Sample $\Pi_i \leftarrow \mathcal{P}$ uniformly and compute $(\boldsymbol{\sigma}, y) \leftarrow \Pi_i(\boldsymbol{\rho})$.
   (c) Compute $\boldsymbol{\sigma}^* \leftarrow \mathsf{Repair}_{\mathsf{M}',\Pi_i}(\boldsymbol{\sigma}, y, p^{**}, \eta/2T)$.
4. Compute $(\boldsymbol{\sigma}^*, p^{**}) \leftarrow \mathsf{M}'(\boldsymbol{\sigma}^*)$.
5. Return $\boldsymbol{\sigma}^*$.

We begin by analyzing the size of $\mathsf{Prepare}$ and Repair'. First, note that both procedures have oracle access to M and $\mathcal{P}$, and Repair' has oracle access to Π, and all oracle calls to M are with parameters $\varepsilon/2T = \Theta(\varepsilon\eta^3/\ell)$ and $\delta^2/64T^2 = \Theta(\delta\eta^6/\ell^2)$, which are polynomially related to $\varepsilon, \delta, \eta, 1/\ell$, as desired.

In $\mathsf{Prepare}$, at most T calls are made to M' and to random $\Pi_i \leftarrow \mathcal{P}$, outside of Repair. Additionally, Repair makes $O(T/\eta)$ calls to M' and Π_i (for each $i \le t-1$) for a total of $O(T) + O(T^2/\eta)$ calls to measurements in M and $\mathcal{P}$. Thus, $\mathsf{Prepare}$ is an oracle circuit of size $O(T) + O(T^2/\eta) = O(\ell/\eta^3) + O(\ell^2/\eta^7) = \mathrm{poly}(\ell, 1/\eta)$.

Similarly, Repair' makes T calls to M' and random $\Pi_i \leftarrow \mathcal{P}$, outside of Repair, and its $O(T)$ calls to Repair themselves make $O(T/\eta)$ calls to M' and Π_i (for each $i \le t-1$). Thus, Repair' is also an oracle circuit of size $O(T) + O(T^2/\eta) = O(\ell/\eta^3) + O(\ell^2/\eta^7) = \mathrm{poly}(\ell, 1/\eta)$.

Next, we prove the first part of property 3. Let us fix any sampling of $t \in [T]$ in the first step of $\mathsf{Prepare}$ and prove our claim for this fixed t. Let us denote by p_0 the output of the first $\mathsf{M}_{\varepsilon,\delta}$, denote by p_i the output of the $\mathsf{M}' := \mathsf{M}_{\varepsilon/2T, \delta^2/64T^2}$ measurement in the ith loop of $\mathsf{Prepare}$, and denote by p_t the output of the final M' call in $\mathsf{Prepare}$. For each $j \in [T]$ let p_{t+j} denote the output of the jth call to M' in Repair', and p_{t+T+1} denote the output of the final M' call in Repair'. Let p_{t+T+2} denote the output of the second $\mathsf{M}_{\varepsilon,\delta}$ call outside of $\mathsf{Prepare}$ and Repair'.

Showing the first half of property 3 is equivalent to proving that

$$\Pr[|p_0 - p_{t+T+2}| \ge 4\varepsilon] \le N \cdot (\eta + 4\delta).$$

Observing that $\delta \leq 1$ and $T = 4\ell/\eta^3 \geq 1$, we know that $\varepsilon/T \leq \varepsilon$ and $\delta^2/64T^2 \leq \delta$. Thus, it follows that

$$\Pr[|p_0 - p_1| \geq \varepsilon], \Pr[|p_{t+T+1} - p_{t+T+2}| \geq \varepsilon] \leq \delta$$

by definition, and for all $1 \leq i \leq t+T$,

$$\Pr\left[|p_i - p_{i+1}| \geq \frac{\varepsilon}{T}\right] \leq N \cdot \left(\frac{\eta}{2T} + \frac{\delta^2}{64T^2}\right) + 4\sqrt{\frac{\delta^2}{64T^2}}$$

by Property 3 of Lemma 3.2.

Next, let

$$\mathsf{Good} := (|p_0 - p_1| \leq \varepsilon) \wedge (|p_{t+T+1} - p_{t+T+2}| \leq \varepsilon) \wedge \left(|p_i - p_{i+1}| \leq \frac{\varepsilon}{T}\right)_{i=1}^{t+T}$$

denote the event that all probabilities are close. Since $t+T \leq 2T$, it follows by a union bound that

$$\begin{aligned}
\Pr[\mathsf{Good}] &\geq 1 - \left[\delta + \delta + N \cdot \left(\frac{\eta}{2T} + \frac{\delta^2}{64T^2}\right) \cdot (t+T) + 4\sqrt{\frac{\delta^2}{64T^2}} \cdot (t+T)\right] \\
&\geq 1 - \left[\delta + \delta + N \cdot \left(\frac{\eta}{2T} + \frac{\delta^2}{64T^2}\right) \cdot 2T + 4\sqrt{\frac{\delta^2}{64T^2}} \cdot 2T\right] \\
&= 1 - [2\delta + N \cdot \eta + N \cdot \delta^2/32T + \delta] \\
&\geq 1 - [3\delta + N \cdot \eta + N \cdot \delta] \\
&\geq 1 - N \cdot (\eta + 4\delta).
\end{aligned}$$

Now note that if Good occurs, then

$$|p_0 - p_{t+T+2}| \leq \sum_{i=0}^{t+T+1} |p_i - p_{i+1}| \leq \varepsilon + \varepsilon + \frac{\varepsilon}{T} \cdot (t+T) \leq 4\varepsilon.$$

Thus,

$$\Pr[|p^* - p^{**}| \geq 4\varepsilon] \leq \Pr\left[\overline{\mathsf{Good}}\right] \leq N \cdot (\eta + 4\delta),$$

as desired.

We note that the first half of property 2 is equivalent to stating that

$$\Pr[|p_0 - p_t| \geq 4\varepsilon] \leq N \cdot (\eta + 4\delta),$$

which follows from the same argument as above. The second half of property 2 follows directly by observing that the last step of $\mathsf{Prepare}$ (which determines p') is a measurement M' and applying property 2 of Lemma 3.2.

Finally, to prove the second half of property 3, we first fix the state $\boldsymbol{\rho}$ and consider the current implementation:

Hybrid $\mathcal{H}_1$:

1. Sample $\Pi \leftarrow \mathcal{P}$.
2. Compute $(\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\rho})$.
3. Compute $(\boldsymbol{\rho}', p') \leftarrow \mathsf{Prepare}_{\mathsf{M},\mathcal{P}}(\boldsymbol{\rho}^*, \varepsilon, \delta, \eta, \ell)$.
4. Compute $(\boldsymbol{\sigma}, y) \leftarrow \Pi(\boldsymbol{\rho}')$.

5. Compute $\boldsymbol{\sigma}^*_{\Pi} \leftarrow \mathsf{Repair}'_{\mathsf{M},\mathcal{P},\Pi}(\boldsymbol{\sigma}, y, p', \varepsilon, \delta, \eta, \ell)$.
6. Output $(\Pi, \boldsymbol{\sigma}^*_{\Pi})$.

In what follows, we open the definition of $\mathsf{Prepare}$ and Repair', and replace steps 3-5 to produce the following identical distribution:

Hybrid $\mathcal{H}_2$:

1. Sample $\Pi \leftarrow \mathcal{P}$.
2. Compute $(\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\rho})$.
3. Set $\boldsymbol{\sigma}'' = \boldsymbol{\rho}^*$ and $T = 4\ell/\eta^3$. Sample $\Pi_1, \ldots, \Pi_T \leftarrow \mathcal{P}$ uniformly. Sample $t \leftarrow [T]$ uniformly and *replace* Π_t with Π. Define $\mathsf{M}' := \mathsf{M}_{\varepsilon/2T, \delta^2/64T^2}$.
4. For $i = 1$ to T:
   (a) Compute $(\boldsymbol{\rho}', p') \leftarrow \mathsf{M}'(\boldsymbol{\sigma}'')$.
   (b) Compute $(\boldsymbol{\sigma}', y_{\Pi_i}) \leftarrow \Pi_i(\boldsymbol{\rho}')$.
   (c) Compute $\boldsymbol{\sigma}'' \leftarrow \mathsf{Repair}_{\mathsf{M}',\Pi_i}(\boldsymbol{\sigma}', y_{\Pi_i}, p', \eta/2T)$.
5. For $i = T+1$ to $T+t$:
   (a) Compute $(\boldsymbol{\rho}', p') \leftarrow \mathsf{M}'(\boldsymbol{\sigma}'')$.
   (b) Sample $\Pi_i \leftarrow \mathcal{P}$ uniformly and compute $(\boldsymbol{\sigma}', y_{\Pi_i}) \leftarrow \Pi_i(\boldsymbol{\rho}')$.
   (c) Compute $\boldsymbol{\sigma}'' \leftarrow \mathsf{Repair}_{\mathsf{M}',\Pi_i}(\boldsymbol{\sigma}', y_{\Pi_i}, p', \eta/2T)$.
6. Compute $(\boldsymbol{\sigma}^*, p^{**}) \leftarrow \mathsf{M}'(\boldsymbol{\sigma}'')$.
7. Output $(\Pi, \boldsymbol{\sigma}^*)$.

Next, we note that since t, Π, and Π_t are sampled uniformly randomly, we can move the sampling of t to after step 4 and drop the planting of Π while maintaining an identical output distribution:

Hybrid $\mathcal{H}_3$:

1. Compute $(\boldsymbol{\rho}^*, p^*) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\rho})$.
2. Set $\boldsymbol{\sigma}'' = \boldsymbol{\rho}^*$ and $T = 4\ell/\eta^3$. Sample $\Pi_1, \ldots, \Pi_T \leftarrow \mathcal{P}$ uniformly. Define $\mathsf{M}' := \mathsf{M}_{\varepsilon/2T, \delta^2/64T^2}$.
3. For $i = 1$ to T:
   (a) Compute $(\boldsymbol{\rho}', p') \leftarrow \mathsf{M}'(\boldsymbol{\sigma}'')$.
   (b) Compute $(\boldsymbol{\sigma}', y_{\Pi_i}) \leftarrow \Pi_i(\boldsymbol{\rho}')$.
   (c) Compute $\boldsymbol{\sigma}'' \leftarrow \mathsf{Repair}_{\mathsf{M}',\Pi_i}(\boldsymbol{\sigma}', y_{\Pi_i}, p', \eta/2T)$.
4. Sample $t \leftarrow [T]$ uniformly and set $\Pi = \Pi_t$.
5. For $i = T+1$ to $T+t$:

(a) Compute $(\boldsymbol{\rho}', p') \leftarrow \mathsf{M}'(\boldsymbol{\sigma}'')$.

(b) Sample $\Pi_i \leftarrow \mathcal{P}$ uniformly and compute $(\boldsymbol{\sigma}', y_{\Pi_i}) \leftarrow \Pi_i(\boldsymbol{\rho}')$.

(c) Compute $\boldsymbol{\sigma}'' \leftarrow \mathsf{Repair}_{\mathsf{M}',\Pi_i}(\boldsymbol{\sigma}', y_{\Pi_i}, p', \eta/2T)$.

6. Compute $(\boldsymbol{\sigma}^*, p^{**}) \leftarrow \mathsf{M}'(\boldsymbol{\sigma}'')$.

7. Output $(\Pi_t, \boldsymbol{\sigma}^*)$.

Note that steps 5 and 6 of $\mathcal{H}_3$ have *no dependence* on $\Pi = \Pi_t$, so it suffices to show that the state $\boldsymbol{\sigma}$ at the end of step 4 satisfies

$$\mathsf{TD}((\Pi_t, \boldsymbol{\sigma}), (\Pi', \boldsymbol{\sigma})) \leq N \cdot \eta$$

for uniformly sampled $\Pi' \leftarrow \mathcal{P}$.

To this end, for every $j \in [N]$, define the random variable $C_j := |\{i \in [T] : y_{\Pi_i} = y_j\}|$ denoting the number of projections with outcome equal to y_j in steps 1-3, where $Y = \{y_1, \ldots, y_N\}$ is the set of N possible outcomes for all projections in $\mathcal{P}$. Note that for every $j \in [N]$,

$$\begin{aligned}\Pr[(y_{\Pi_t} = y_j) \wedge (C_j \leq \eta T/2)] &= \Pr[y_{\Pi_t} = y_j \mid C_j \leq \eta T/2] \cdot \Pr[C_j \leq \eta T/2] \\ &\leq \Pr[y_{\Pi_t} = y_j \mid C_j \leq \eta T/2] \leq \eta/2,\end{aligned}$$

where the probability is over the randomness in $\mathcal{H}_3$. Thus, by a union bound we have that

$$\Pr[\exists j \in [N] : (y_{\Pi_t} = y_j) \wedge (C_j \leq \eta T/2)] \leq N \cdot \eta/2.$$

Let B be the event that there exists $j \in [N]$ such that $y_{\Pi_t} = y_j$ and $C_j \leq \eta T/2$. By a simple coupling argument, we have that

$$\begin{aligned}\mathsf{TD}((\Pi_t, \boldsymbol{\sigma}), (\Pi', \boldsymbol{\sigma})) &= \Pr[B] \cdot \mathsf{TD}((\Pi_t, \boldsymbol{\sigma} \mid B), (\Pi', \boldsymbol{\sigma} \mid B)) + \Pr\left[\overline{B}\right] \cdot \mathsf{TD}\left(\left(\Pi_t, \boldsymbol{\sigma} \mid \overline{B}\right), \left(\Pi', \boldsymbol{\sigma} \mid \overline{B}\right)\right) \\ &\leq N \cdot \eta/2 + \mathsf{TD}\left(\left(\Pi_t, \boldsymbol{\sigma} \mid \overline{B}\right), \left(\Pi', \boldsymbol{\sigma} \mid \overline{B}\right)\right).\end{aligned}$$

But now note that the event $\overline{B}$ implies that for *every* possible outcome of y_{Π_t}, at least $\frac{\eta}{2}$ fraction of the T projections result in the same outcome as y_{Π_t}. Recall that all $\Pi \leftarrow \mathcal{P}$ are drawn from an identical distribution, so all projections which observed y_{Π_t} are drawn from the *same* distribution as Π_t. Thus, by Lemma 3.6,

$$\mathsf{TD}\left(\left(\Pi_t, \boldsymbol{\sigma} \mid \overline{B}\right), \left(\Pi', \boldsymbol{\sigma} \mid \overline{B}\right)\right) \leq \sqrt{\frac{\ell}{2(\frac{\eta}{2} \cdot T)}} = \eta/2,$$

and so we conclude that

$$\mathsf{TD}((\Pi, \boldsymbol{\sigma}), (\Pi', \boldsymbol{\sigma})) := \mathop{\mathsf{TD}}_{t \leftarrow [T]}((\Pi_t, \boldsymbol{\sigma}), (\Pi', \boldsymbol{\sigma})) \leq N \cdot \eta/2 + \eta/2 \leq N \cdot \eta,$$

as desired. □

Remark 4.2 (Efficiency)**.** We observe that when V and A are efficiently computable circuits (in some security parameter, say λ), then $\mathsf{ValEst}^{V,A}_{\cdot,\cdot}$ is an almost projective family such that for any ε, δ, $\mathsf{ValEst}^{V,A}_{\varepsilon,\delta}$ runs in time $\mathrm{poly}(\lambda, 1/\varepsilon, \log 1/\delta)$. If we then set $\mathsf{M} = \mathsf{ValEst}^{V,A}_{\cdot,\cdot}$ and assume $\mathcal{P}$ is an efficiently implementable projection family (i.e. it runs in $\mathrm{poly}(\lambda, 1/\varepsilon, \log 1/\delta)$ time) we see that both $\mathsf{Prepare}$ and Repair' can run in time $\mathrm{poly}(\lambda, 1/\varepsilon, \log 1/\delta, 1/\eta, \ell)$ with *no* oracle calls.

5 Parallel Repetition for Public-Coin Protocols

In this section, we analyze the soundness of k-fold parallel repetition of public coin protocols, and prove that the soundness goes down exponentially in k. For any interactive protocol (P, V) we denote by $(P^{(k)}, V^{(t,k)})$ its t-threshold, k-fold parallel repetition. Namely, $V^{(t,k)}$ accepts at the end of the protocol if and only if at least t out of its k constituent verifiers accept.

5.1 Our Results

We prove the following parallel repetition theorem for public-coin protocols:

Theorem 5.1. *Let (P, V) be a m-round (where $m = m(\lambda)$ is polynomial) public-coin interactive argument. Suppose that there exists a quantum adversary $\mathcal{B}$, $\xi = \xi(\lambda) > 0$, and polynomials $1 \leq t = t(\lambda) \leq k = k(\lambda)$ such that for all $\lambda \in \mathbb{N}$,*

$$\Pr\Big[\langle(\mathcal{B}, V^{(t,k)})(1^\lambda)\rangle = 1\Big] \geq \xi. \tag{1}$$

Then there exists a quantum adversary $\mathcal{A}$ running in time $\mathrm{poly}(|\mathcal{B}|, \lambda, 1/\xi)$ and a negligible function negl such that for every $\lambda \in \mathbb{N}$,

$$\Pr\Big[\langle(\mathcal{A}, V)(1^\lambda)\rangle = 1\Big] \geq \frac{t}{k} - 2m\sqrt{\frac{-\log(\xi/3m^2)}{k}} - \mathsf{negl}(\lambda).$$

Note that Theorem 5.1 implies the following corollary:

Corollary 5.2. *Let (P, V) be a m-round (where $m = m(\lambda)$ is polynomial) public-coin interactive argument and let $\varepsilon = \varepsilon(\lambda)$ be any parameter such that for every QPT adversary $\mathcal{A}$ there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$,*

$$\Pr\Big[\langle(\mathcal{A}, V)(1^\lambda)\rangle = 1\Big] \leq \varepsilon(\lambda) + \mathsf{negl}(\lambda).$$

Fix any polynomials $1 \leq t = t(\lambda) \leq k = k(\lambda)$ such that $\varepsilon < t/k$.[12] Then for every QPT adversary $\mathcal{B}$ there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$,

$$\Pr\Big[\langle(\mathcal{B}, V^{(t,k)})(1^\lambda)\rangle = 1\Big] \leq \max\{6m^2 \cdot \exp\left(\frac{-k}{4m^2} \cdot \left(\frac{t}{k} - \varepsilon\right)^2\right), \mathsf{negl}(\lambda)\}.$$

Proof of Corollary 5.2. Suppose for the sake of contradiction that there exists a QPT adversary $\mathcal{B}$ that succeeds against $V^{(t,k)}$ with non-negligible probability $\xi > 6m^2 \cdot \exp\left(\frac{-k}{4m^2} \cdot \left(\frac{t}{k} - \varepsilon\right)^2\right) > 0$. Applying Theorem 5.1 gives a quantum adversary $\mathcal{A}$ running in time $\mathrm{poly}(\lambda, 1/\xi)$ such that

$$\begin{aligned}\Pr\Big[\langle(\mathcal{A}, V)(1^\lambda)\rangle = 1\Big] &\geq \frac{t}{k} - 2m\sqrt{\frac{-\log\left(\left[6m^2 \cdot \exp\left(\frac{-k}{4m^2} \cdot \left(\frac{t}{k} - \varepsilon\right)^2\right)\right]/3m^2\right)}{k}} - \mathsf{negl}(\lambda) \\ &= \frac{t}{k} - 2m\sqrt{\frac{-\log\left(2\exp\left(\frac{-k}{4m^2} \cdot \left(\frac{t}{k} - \varepsilon\right)^2\right)\right)}{k}} - \mathsf{negl}(\lambda)\end{aligned}$$

[12] The condition that $\varepsilon < t/k$ is necessary as otherwise any adversary $\mathcal{A}$ which succeeds with probability at least ε can be easily converted (by a direct product) into a k-fold adversary which already convinces $\geq \varepsilon k \geq t$ verifiers on expectation (and hence succeeds with probability at least 1/2).

$$
\begin{aligned}
&= \frac{t}{k} - 2m\sqrt{\frac{-\log\left(\exp\left(\frac{-k}{4m^2}\cdot\left(\frac{t}{k}-\varepsilon\right)^2\right)\right)-1}{k}} - \mathsf{negl}(\lambda)\\
&= \frac{t}{k} - 2m\sqrt{\frac{\frac{k}{4m^2}\cdot\left(\frac{t}{k}-\varepsilon\right)^2-1}{k}} - \mathsf{negl}(\lambda)\\
&= \frac{t}{k} - \sqrt{\left(\frac{t}{k}-\varepsilon\right)^2 - \frac{4m^2}{k}} - \mathsf{negl}(\lambda)\\
&\geq \frac{t}{k} - \left[\left(\frac{t}{k}-\varepsilon\right) - \frac{\frac{4m^2}{k}}{2\left(\frac{t}{k}-\varepsilon\right)}\right] - \mathsf{negl}(\lambda) \qquad (2)\\
&\geq \frac{t}{k} - \left(\frac{t}{k}-\varepsilon\right) + \frac{\frac{4m^2}{k}}{2\left(\frac{t}{k}-\varepsilon\right)} - \mathsf{negl}(\lambda)\\
&= \varepsilon + \frac{\frac{4m^2}{k}}{2\left(\frac{t}{k}-\varepsilon\right)} - \mathsf{negl}(\lambda)\\
&\geq \varepsilon + \frac{\frac{4m^2}{k}}{\frac{2t}{k}} - \mathsf{negl}(\lambda)\\
&= \varepsilon + \frac{2m^2}{t} - \mathsf{negl}(\lambda),
\end{aligned}
$$

where (2) follows from the fact that $\sqrt{x^2-y} \leq x - \frac{y}{2x}$ for $x, y > 0$ such that $x^2 - y > 0$[13]. Since m and t are polynomial, $\varepsilon + \frac{2m^2}{t} - \mathsf{negl}(\lambda) > \varepsilon + \mathsf{negl}(\lambda)$.

But since $\mathcal{B}$ has non-negligible advantage, then $1/\xi$ is polynomial in λ (infinitely often) and so $\mathcal{A}$ is QPT (infinitely often), in contradiction to the original assumption that no such QPT adversary exists. Thus, no such QPT adversary $\mathcal{B}$ exists either. □

Proof of Theorem 5.1. We assume that $m \geq 2$, since parallel repetition in the case of $m = 1$ can be easily shown using a test-and-check reduction (one can repeatedly test queries on fresh states until the first acceptance).

Denote by $|\psi\rangle$ the initial state of $\mathcal{B}$, and let $\Lambda = \mathrm{poly}(\lambda)$ be the number of qubits in $|\psi\rangle$. For the sake of simplicity, we assume without loss of generality that $|\psi\rangle$ contains designated registers $\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_m$, where $\mathcal{Z}_j$ is the register on which $\mathcal{B}$ writes its response to the j'th round queries. We denote all the other registers in $|\psi\rangle$ by $\mathcal{I}$. We write $Q_j^{(k)}$ to refer to the set of possible queries sent in round j in the k-fold repeated protocol $(P^{(k)}, V^{(t,k)})$ and $\bar{q}_j^i$ to refer to the ith coordinate of a query vector $\bar{q}_j \in Q_j^{(k)}$.

For every $j \in [m]$ we denote by U_j the collection of unitaries indexed by jth round queries $\bar{q}_j$ such that $U_j(\bar{q}_j)$ is the unitary that $\mathcal{B}$ applies to its quantum state upon receiving query $\bar{q}_j$ (which operates on $\mathcal{I}$ and $\mathcal{Z}_j$). We denote by

$$U^{(j)}(\bar{q}_j, \bar{q}_{j+1}, \ldots, \bar{q}_m) = U_m(\bar{q}_m)U_{m-1}(\bar{q}_{m-1})\ldots U_j(\bar{q}_j). \qquad (3)$$

The assumption that each $U_j(\bar{q}_j)$ is a unitary is without loss of generality, as any adversary not of this form can be "purified" into an adversary with unitary strategies, identical observable behavior, and constant factor blowup in size.

[13]The condition holds here since $c := 6m^2 \cdot \exp\left(\frac{-k}{4m^2}\cdot\left(\frac{t}{k}-\varepsilon\right)^2\right) < \xi \leq 1$, and so $2m\sqrt{\frac{-\log\left(c/3m^2\right)}{k}} = \sqrt{(\frac{t}{k}-\varepsilon)^2 - \frac{4m^2}{k}} > 0$.

For a complete transcript τ_m and index $i \in [k]$, define $\mathsf{Accept}_i(\tau_m) = 1$ iff the output of the ith verifier at the end of a protocol with transcript τ_m is '1', and $\mathsf{Accept}(\tau_m) = 1$ iff $\sum_i \mathsf{Accept}_i(\tau_m) \geq t$. For any partial transcript $\tau = (\bar{q}_1, \bar{z}_1, \ldots, \bar{q}_{j-1}, \bar{z}_{j-1})$, we define a classical function V_τ that takes as input a continuation of the transcript, namely $(\bar{q}_j, \bar{z}_j, \bar{q}_{j+1}, \bar{z}_{j+1}, \ldots, \bar{q}_m, \bar{z}_m)$, and outputs the (threshold) parallel verifier's decision predicate

$$V^{(t,k)}(\bar{q}_1, \bar{z}_1, \ldots, \bar{q}_m, \bar{z}_m) := \mathsf{Accept}(\bar{q}_1 || \bar{z}_1 || \ldots || \bar{q}_m || \bar{z}_m).$$

Additionally, for a partial transcript $\tau = (\bar{q}_1, \bar{z}_1, \ldots, \bar{q}_{\ell-1}, \bar{z}_{\ell-1})$, query $\bar{q}_\ell$, and parameters $\ell, \varepsilon, \delta$, denote by $U_{\mathsf{ValEst}}(\tau, \bar{q}_\ell, \ell, \varepsilon, \delta)$ the unitary that first coherently computes $\bar{z}_\ell = U_\ell(\bar{q}_\ell)$ and then coherently computes $\mathsf{ValEst}_{V_{\tau||\bar{q}_\ell||\bar{z}_\ell}, U^{(\ell+1)}}(\cdot, \varepsilon, \delta)$ (writing the outcome of the ValEst procedure in the ancilla register $\mathcal{E}$ using the ancilla register $\mathcal{F}$). We think of $U^{(\ell+1)}$ as a quantum circuit that takes as input $(\bar{q}_{\ell+1}, \ldots, \bar{q}_m)$ and outputs corresponding answers $(\bar{z}_{\ell+1}, \ldots, \bar{z}_m)$ (according to $U^{(\ell+1)}$), and we think of $V_{\tau||\bar{q}_\ell||\bar{z}_\ell}$ as the circuit that generates random queries $(\bar{q}_{\ell+1}, \ldots, \bar{q}_m)$[14] and upon receiving answers $(\bar{z}_{\ell+1}, \ldots, \bar{z}_m)$ outputs $\mathsf{Accept}(\tau_m)$ for $\tau_m \triangleq (\tau || \bar{q}_\ell || \bar{z}_\ell || \bar{q}_{\ell+1} || \bar{z}_{\ell+1} || \ldots || \bar{q}_m || \bar{z}_m)$.

Informal description of the reduction $\mathcal{A}$. Before we give the full reduction and analysis, we first informally describe our proof strategy. The adversary $\mathcal{A}$ uses as subroutines the algorithms ValEst, $\mathsf{Prepare}$, and Repair' as defined in Lemmas 3.2 and 4.1, and does the following:

1. Apply ValEst to the state $|\psi\rangle$ (which is the initial state of $\mathcal{B}$), to obtain a probability p_0. Given sufficiently many copies of the state $|\psi\rangle$, with high probability, for at least one of these states p_0 will be close to ξ, which is the success probability of $\mathcal{B}$. Specifically, we can assume that $p_0 \geq \xi - \varepsilon_0$ for some small ε_0 (which we set to be $\varepsilon_0 := \frac{\xi}{m^2}$).
2. Choose a random execution $i \leftarrow [k]$ to embed the queries in.
3. For each round $\ell \in [m]$, upon receiving a query q_ℓ from the external verifier V, do the following $\mathsf{iter} = \mathrm{poly}(\lambda, m, 1/\xi)$ times:
   (a) Generate a k-fold query $\bar{q}_\ell$ such that its i'th coordinate is equal to q_ℓ (i.e., embed the external verifier's query in the ith execution and simulate the remaining $k-1$ verifiers internally). Namely, generate $\bar{q}_\ell \leftarrow Q_\ell^{(k)}$ such that $\bar{q}_\ell^i = q_\ell$.
   (b) Let $\Pi = \mathsf{CheckCoins}_{\bar{q}_\ell}$ be the projection that estimates the acceptance probability of the answers given by $\mathcal{B}$ on queries $\bar{q}_\ell$. We would like to apply Π to the state so that if the resulting probability is not sufficiently high (close to ξ), then we can repair it in a *forgetful* way so that $\bar{q}_\ell$ (and in particular $\bar{q}_\ell^i$) is forgotten. This is implemented as instructed by Lemma 4.1, as follows:
   Let $V_{\tau_{\ell-1}}$ be the verifier circuit which samples queries $(\bar{q}_\ell, \ldots, \bar{q}_m)$, and upon receiving responses $(\bar{z}_\ell, \ldots, \bar{z}_m)$, outputs its verdict $\mathsf{Accept}(\tau_{\ell-1}, \bar{q}_\ell, \bar{z}_\ell, \ldots, \bar{q}_m, \bar{z}_m)$. In what follows, for ease of presentation, we slightly abuse notation and also denote by $U^{(\ell)}$ the adversary circuit which upon receiving queries $(\bar{q}_\ell, \ldots, \bar{q}_m)$ applies the unitary $U^{(\ell)}(\bar{q}_\ell, \ldots, \bar{q}_m)$ (which describes the answers of $\mathcal{B}$ from round ℓ onward, conditioned on the transcript in the first $\ell - 1$ rounds being $\tau_{\ell-1}$; see Equation (3)) to its state, thereby producing responses $\bar{z}_\ell, \ldots, \bar{z}_m$ in registers $\mathcal{Z}_\ell, \ldots, \mathcal{Z}_m$.[15] We can now define $M^\ell = \mathsf{ValEst}_{V_{\tau_{\ell-1}}, U^{(\ell)}}$, which is an almost-projective measurement family.

[14] Since the protocol is public-coin, $\bar{q}_{\ell+1}, \ldots, \bar{q}_m$ are independent/non-adaptively chosen random coins and can thus be sampled concurrently. Therefore, given access to $U^{(\ell+1)}$, a random continuation of the protocol can be thought of as a single-round quantum game. Hence, ValEst can be implemented essentially without modification (as in [LMS22]).

[15] Note that we also use $U^{(\ell)}$ to refer to the same unitary which is implemented by this circuit.

i. Apply M^ℓ to the state to obtain a probability estimate $p_{\ell-1}$.

ii. Run $\mathsf{Prepare}$ with respect to the family of projections $\mathcal{P} = \{\mathsf{CheckCoins}_{\bar{q}_\ell}\}_{\bar{q}_\ell \in Q_\ell^{(k)}}$ and the family of almost projections M^ℓ, on input the relevant parameters, including $p_{\ell-1}$.

iii. Apply Π to the state to obtain a probability estimate p.

iv. If p is not sufficiently high (i.e. $p < \xi - \ell\epsilon_0$) then repair the state by applying the (forgetful) Repair' procedure to it, and try again (go back to Item 3(b)i).

v. If p is sufficiently high (i.e. $p \geq \xi - \ell\epsilon_0$) then measure the answer register $\mathcal{Z}_\ell$ to obtain answers $\bar{z}_\ell$ corresponding to $\bar{q}_\ell$, send $\bar{z}_\ell^i$ to the verifier, and proceed to round $\ell + 1$.

We now formally describe algorithm $\mathcal{A}$ in the Figure 1 below.

Projection 1 : $\mathsf{CheckCoins}_{\ell,\tau_{\ell-1},\bar{q}}[\boldsymbol{\rho}]$

Let L be the number of ancilla qubits required to compute $U_{\mathsf{ValEst}}(\tau_{\ell-1}, \bar{q}, \ell, \varepsilon, \delta)$.

1. Apply $U_{\mathsf{ValEst}}(\tau_{\ell-1}, \bar{q}, \ell, \varepsilon, \delta)$ to $\boldsymbol{\rho} \otimes \left|0^L\right\rangle\left\langle 0^L\right|_{\mathcal{E},\mathcal{F}}$, resulting in state $\sum_{z_\ell, p_{z_\ell}} (\boldsymbol{\rho}'_{z_\ell})_{\mathcal{I},\mathcal{Z}_{-\ell}} \otimes |z_\ell\rangle\langle z_\ell|_{\mathcal{Z}_\ell} \otimes |p_{z_\ell}\rangle\langle p_{z_\ell}|_{\mathcal{E}} \otimes \left|\mathsf{junk}_{z_\ell,p_{z_\ell}}\right\rangle\left\langle \mathsf{junk}_{z_\ell,p_{z_\ell}}\right|_{\mathcal{F}}$.
2. Measure the bits in register $\mathcal{E}$, obtaining p and remaining state $\boldsymbol{\rho}''$.
3. Apply $U_{\mathsf{ValEst}}(\tau_{\ell-1}, \bar{q}, \ell, \varepsilon, \delta)^\dagger$ to get state $\boldsymbol{\sigma} = (\boldsymbol{\rho}''')_{\mathcal{I},\mathcal{Z}} \otimes \left|0^L\right\rangle\left\langle 0^L\right|_{\mathcal{E},\mathcal{F}}$. Discard registers $\mathcal{E}$ and $\mathcal{F}$ before returning $(p, \boldsymbol{\rho}''')$.

Algorithm 1 : The Algorithm $\mathcal{A}$

Let $\mathsf{iter} := \lambda m^2/\xi$ and $\Lambda := \mathsf{size}(|\psi\rangle)$ be the number of qubits of $|\psi\rangle$. $\mathcal{A}$ begins with state $|\psi\rangle^{\otimes \mathsf{iter}}$, and runs as follows:

1. Set $\varepsilon_0 := \frac{\xi}{m^2}, \varepsilon := \frac{\varepsilon_0}{16 \cdot \mathsf{iter}}, \delta := \min\{2^{-\lambda}, 2^{-k}\}, \eta := \frac{1}{2km \cdot \mathsf{iter}}/N_{\varepsilon,\delta}$, where $N_{\varepsilon,\delta}$ is the parameter from Lemma 3.2. Sample $i \leftarrow [k]$. Define $\tau_0 := \perp$.
2. For $s = 1$ to iter:
   (a) Set $\boldsymbol{\rho} = |\psi\rangle$ to be $\mathcal{A}$'s sth copy of $|\psi\rangle$. Run $(p_0, \boldsymbol{\rho}_0) \leftarrow \mathsf{ValEst}_{V_{\tau_0}, U^{(1)}}(\boldsymbol{\rho}, \varepsilon, \delta)$.
      i. If $p_0 \geq \xi - \varepsilon_0$, continue to step 3.
      ii. Else, if $p_0 < \xi - \varepsilon_0$ and $s = \mathsf{iter}$, return $(i, \perp)$.
3. For $\ell = 1$ to m:
   (a) Let $\mathcal{P} := \{\mathsf{CheckCoins}_{\ell,\tau_{\ell-1},\bar{q}}\}_{\bar{q} \in Q_\ell^{(k)}}$ be the projection family defined above. Define the almost projective measurement family $\mathsf{M}^\ell := \{\mathsf{ValEst}_{V_{\tau_{\ell-1}}, U^{(\ell)}}(\cdot, \varepsilon, \delta)\}_{\varepsilon,\delta>0}$.
   (b) Receive the ℓth-round message q_ℓ from V.
   (c) For $s = 1$ to iter:
      i. Generate a query $\bar{q}_\ell \leftarrow Q_\ell^{(k)}$ uniformly conditioned on $\bar{q}_\ell^i = q_\ell$. Let $\Pi := \mathsf{CheckCoins}_{\ell,\tau_{\ell-1},\bar{q}_\ell}$.
      ii. Compute $(\boldsymbol{\rho}'_{\ell-1}, p_{\ell-1}) \leftarrow \mathsf{M}^\ell_{\varepsilon,\delta}(\boldsymbol{\rho}_{\ell-1})$.
      iii. Compute $(\boldsymbol{\rho}''_{\ell-1}, p'_{\ell-1}) \leftarrow \mathsf{Prepare}_{\mathsf{M}^\ell,\mathcal{P}}(\boldsymbol{\rho}'_{\ell-1}, p_{\ell-1}, \varepsilon, \delta, \eta, \Lambda)$.
      If $p'_{\ell-1} < \xi - \ell\varepsilon_0 - (4s+1)\varepsilon$, return $(i, \perp)$.

iv. Compute $(\boldsymbol{\sigma}, p) \leftarrow \Pi[\boldsymbol{\rho}''_{\ell-1}]$.
If $p \geq \xi - (\ell+1)\varepsilon_0$, skip to step 3(d).
Else, if $p < \xi - (\ell+1)\varepsilon_0$ and $s = \mathsf{iter}$, return $(i, \bot)$.

v. Compute $\boldsymbol{\rho}_{\ell-1} = \mathsf{Repair}'_{\mathsf{M}^\ell, \mathcal{P}, \Pi}(\boldsymbol{\sigma}, p, p'_{\ell-1}, \varepsilon, \delta, \eta, \Lambda)$.

(d) Apply $U_{\mathsf{ValEst}}(\tau_{\ell-1}, \bar{q}_\ell, \ell, \varepsilon, \delta)$ to $\boldsymbol{\sigma} \otimes \left|0^L\right\rangle \left\langle 0^L\right|_{\mathcal{E},\mathcal{F}}$ before measuring the $\mathcal{Z}_\ell$ register to get response $\bar{z}_\ell$. Discard/measure the ancilla registers $\mathcal{E}$ and $\mathcal{F}$, resulting in collapsed state $\boldsymbol{\sigma}^*$. Set $\boldsymbol{\rho}_\ell = \boldsymbol{\sigma}^*$.

(e) Set $\tau_\ell = \tau_{\ell-1} || \bar{q}_\ell || \bar{z}_\ell$ and send $\bar{z}^i_\ell$ back to V.

4. Return (i, τ_m).

Runtime. The fact that $\mathcal{A}$ runs in time $\mathrm{poly}(|\mathcal{B}|, \lambda, 1/\xi)$ follows from the efficiency of ValEst', $\mathsf{Prepare}$, and Repair', and the fact that all parameters are polynomial in λ and $1/\xi$.

Soundness. Before we analyze the soundness of our reduction, we note that we can assume without loss of generality that $\xi \gg \delta$, as otherwise our guarantee is essentially trivially true.

We begin by considering the following set of hybrids $\mathcal{H}_j$ defined for each $j \in \{0, \ldots, m\}$:

- Hybrid $\mathcal{H}_j$: For rounds $1 \leq \ell \leq j$, replace the sampling of $\bar{q}_\ell$ in step 3(c)(i) with a completely uniform sample of $\bar{q}_\ell \leftarrow Q_\ell^{(k)}$ and derive partial transcript $\tau_j = (\bar{q}_1, \bar{z}_1, \ldots, \bar{q}_j, \bar{z}_j)$. Now, set $i \leftarrow [k]$ (instead of at step 1) and emulate a random execution of $(\mathcal{A}, V)$ conditioned on (i, τ_j) being the transcript of the protocol after the first j rounds to derive a full transcript τ_m and state $\boldsymbol{\rho}_m$. Return (i, τ_m).

Note that by definition, $\mathcal{H}_0$ produces the distribution of outputs over a random execution of $(\mathcal{A}, V)$.

Before we continue, we note a simple fact about delaying measurements:

Fact 5.2.1. *Fix any state $\boldsymbol{\rho}$, partial transcript $\tau = (\bar{q}_1, z_1, \ldots, \bar{q}_{\ell-1}, z_{\ell-1})$, and ℓ'th round query $\bar{q}_\ell$. Then the following two procedures return identical output distributions:*

- *Apply $U_{\mathsf{ValEst}}(\tau, \bar{q}_\ell, \ell, \varepsilon, \delta)$ to the state $\boldsymbol{\rho} \otimes \left|0^L\right\rangle \left\langle 0^L\right|_{\mathcal{E},\mathcal{F}}$, where $\mathcal{E}, \mathcal{F}$ are the ancilla registers that are initialized to 0^L (where L denotes the number of ancilla qubits used by U_{ValEst}). Then measure registers $\mathcal{E}$, followed by measuring registers $\mathcal{Z}_\ell$ and $\mathcal{F}$, to get p, z_ℓ, junk, and a collapsed state $\boldsymbol{\sigma}$. Return $(z_\ell, p, \boldsymbol{\sigma})$, where $\boldsymbol{\sigma}$ excludes the $\mathcal{E}$ and $\mathcal{F}$ registers that were measured and discarded.*

- *Apply $U_\ell(\bar{q}_\ell)$ to $\boldsymbol{\rho}$ and measure register $\mathcal{Z}_\ell$ to get z_ℓ and state $\boldsymbol{\rho}^*$. Apply*

$$(p, \boldsymbol{\sigma}) \leftarrow \mathsf{ValEst}_{V_{\tau || \bar{q}_\ell || z_\ell}, U^{(\ell+1)}}(\boldsymbol{\rho}'_{z_\ell}, \varepsilon, \delta)(\boldsymbol{\rho}^*)$$

and return $(z_\ell, p, \boldsymbol{\sigma})$.

The above fact follows from the principle of deferred measurement, which asserts that we can always delay the measurement of both p and z_ℓ by introducing ancilla registers $\mathcal{F}$ which are initialized to 0^L. Once all measurements are deferred, the order of measurement clearly does not matter.

In what follows we first prove that $\mathcal{H}_m$ returns an accepting transcript with high probability, and then use a hybrid argument to argue that this implies that $\mathcal{H}_0$ returns an accepting transcript with high probability. Proving that $\mathcal{H}_m$ returns an accepting transcript with high probability is done in two steps. First, we prove that the probability that $\mathcal{H}_m$ returns an accepting transcript is precisely the probability that it does not abort, and then we prove that the probability that it aborts is small.

Lemma 5.3. $\Pr_{(i,\tau_m)\leftarrow\mathcal{H}_m}[\mathsf{Accept}(\tau_m)=1] = 1 - \Pr_{(i,\tau_m)\leftarrow\mathcal{H}_m}[\tau_m=\bot]$.

Proof of Lemma 5.3. It suffices to prove that if $\mathcal{H}_m$ does *not* abort and returns a transcript, then this transcript must be accepting. If $\mathcal{H}_m$ does not abort, then this means that in round $\ell = m$, there is some final $\bar{q}^*$ such that $\mathsf{CheckCoins}_{m,\tau_{m-1},\bar{q}^*}$ on some state $\boldsymbol{\rho}_{m-1}$ returned $p \geq \xi - (m+1)\varepsilon_0$. Note that step 3 of $\mathsf{CheckCoins}$ is undone by step 3(d) of $\mathcal{H}_m$. This means that we applied U_{ValEst}, then measured some $p^* \geq \xi - (m+1)\varepsilon_0$, followed by measuring the response $\bar{z}_m$, and junk_{z_ℓ}. By Fact 5.2.1, this is equivalent to having measured $\bar{z}_m$ and computing ValEst on the resulting *full* transcript and receiving some probability $p^* \geq \xi - (m+1)\varepsilon_0 = \xi - \frac{m+1}{m^2}\xi > 0$. But when $\ell = m$ this ValEst is the *deterministic* function which computes precisely whether the full transcript is accepting and must either have value 0 or 1, and so $p^* = 1$. Thus, by our setting of parameters, the full transcript is accepting as long as $\mathcal{H}_m$ does not abort. □

We next argue that $\mathcal{H}_m$ aborts only with small probability.

Lemma 5.4. $\Pr_{(i,\tau_m)\leftarrow\mathcal{H}_m}[\tau_m=\bot] \leq 1/k + \mathsf{negl}(\lambda)$.

Proof of Lemma 5.4. We first bound the probability that $\mathcal{H}_m$ aborts in step 2:

Claim 5.4.1. *$\mathcal{H}_m$ aborts in step 2 with probability at most $\mathsf{negl}(\lambda)$.*

Proof of Claim 5.4.1. Since the starting state $|\psi\rangle$ has success probability which is at least ξ, by property 1 of Lemma 3.2, we have that

$$\mathbb{E}[p_0 | (p_0, \boldsymbol{\rho}_0) \leftarrow \mathsf{ValEst}_{V_{\tau_0},U^{(1)}}(|\psi\rangle, \varepsilon, \delta)] \geq \xi.$$

Recall that since ValEst takes on values in $[-\frac{1}{2}, \frac{3}{2}]$ (see Lemma 3.2), by a simple Markov argument, the probability that a given copy of $|\psi\rangle$ will succeed in step 2(a) is

$$\Pr\left[p_0 \geq \xi - \varepsilon_0 \;\middle|\; (p_0, \boldsymbol{\rho}_0) \leftarrow \mathsf{ValEst}_{V_{\tau_0},U^{(1)}}(|\psi\rangle, \varepsilon, \delta)\right] \geq 2\varepsilon_0/3.$$

Thus, by using a Chernoff bound we conclude that the probability of failing to succeed with any of the $\mathsf{iter} = \lambda/\varepsilon_0$ copies (and thus aborting in step 2) is at most $\mathsf{negl}(\lambda)$. □

To bound the probability of aborting in step 3, we will rely on the following basic fact from probability theory, that for every set of events $\{E_i\}_{i=1}^n$ it holds that

$$\Pr[\vee_i E_i] = \sum_i \Pr\left[E_i \mid (\wedge_{j<i} \overline{E_j})\right].$$

Thus, it suffices to bound the conditional probability of each "bad" event occurring assuming all previous "bad" events have not occurred.

Claim 5.4.2. *Conditioned on not aborting in step 2, $\mathcal{H}_m$ aborts in step 3 with probability at most $\frac{1}{k} + \mathsf{negl}(\lambda)$.*

Proof of Claim 5.4.2. We begin by analyzing step 3(c)(ii) when $s = 1$ and distinguish between the case where $\ell = 1$ and $\ell > 1$:

- $\ell = 1$: the ValEst measurement falls into the same projective family as $\mathcal{M}^1$, so by property 2 of Lemma 3.2, step 3(c)(ii) will return $p'_0 < \xi - \varepsilon_0 - \varepsilon$ with probability at most δ.

- $\ell > 1$: If $\mathcal{H}_m$ did not abort in the first $\ell - 1$ rounds, then the last invocation of $\mathsf{CheckCoins}$ in the $(\ell-1)$st round returned some $p > \xi - \ell\varepsilon_0$. Denote by $z_{\ell-1}$ the response in step 3(d) in the $(\ell-1)$st round. Fact 5.2.1 implies that the state at the beginning of the ℓth round looks identical to a state obtained after computing ValEst on some state and obtaining probability $p \geq \xi - \ell\varepsilon_0$. Since this ValEst measurement falls into the same projective family as $\mathcal{M}^\ell$, again by property 2 of Lemma 3.2, step 3(c)(ii) will return $p < \xi - \ell\varepsilon_0 - \varepsilon$ with probability at most δ.

Now, for iteration s, if step 3(c)(ii) returned some value $p \geq \xi - \ell\varepsilon_0 - (4(s-1)+1)\varepsilon$, then by property 2 of Lemma 4.1, the probability that step 3(c)(iii) aborts during iteration s is at most $N_{\varepsilon,\delta} \cdot (\eta + 4\delta)$, where $N_{\varepsilon,\delta} = O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$ is the number of possible outcomes of $\mathsf{CheckCoins}$ (see Lemma 3.2).

Next, we let x_s denote the probability that step 3(c)(iv) skips to step 3(d) in iteration s. By property 3 of Lemma 4.1 and a simple Markov argument,

$$\Pr[\text{step 3(c)(ii) in iter. } s+1 \text{ gives } p < \xi - \ell\varepsilon_0 - (4s+1)\varepsilon \mid \text{not aborting in iter. } s] \leq \frac{N_{\varepsilon,\delta}(\eta + 4\delta)}{1 - x_s}.$$

But since we do not skip in iteration s to step 3(d) with precisely probability $1 - x_s$, conditioned on step 3(c)(ii) being good in iteration s, the probability that we fail in step 3(c)(iv) *and* the next call to step 3(c)(ii) returns a bad probability is at most $N_{\varepsilon,\delta} \cdot (\eta + 4\delta)$.

Iterating this argument, we see that for a given round ℓ, the total probability that $\mathcal{H}_m$ aborts in step 3(c)(iii) and step 3(c)(ii) returns a value below what we expect (unless step 3(c)(iv) succeeds, in which case we are done with round ℓ) is at most $\delta + \mathsf{iter} \cdot 2N_{\varepsilon,\delta}(\eta + 4\delta) \leq 1/km + \mathsf{negl}(\lambda)$.

Otherwise, we can assume that in iteration s, no abort has occurred in step 3(c)(iii) up to iteration s in round ℓ. In iteration s, given that the last $\mathsf{Prepare}$ returned a $p \geq \xi - \ell\varepsilon_0 - (4s+1)\varepsilon$, we have by property 2 of Lemma 4.1 that any subsequent ValEst would return a $p \geq \xi - \ell\varepsilon_0 - (4s+2)\varepsilon$ with probability at least $1 - \delta$. But since such a ValEst is on expectation equal to the success probability of the prover's state at the start of step 3(c)(iv), this success probability is at least

$$(\xi - \ell\varepsilon_0 - (4s+2) \cdot \varepsilon) \cdot (1 - \delta) - \frac{1}{2} \cdot \delta \geq \xi - \ell\varepsilon_0 - (4 \cdot \mathsf{iter} + 2) \cdot \varepsilon - \frac{3\delta}{2} \geq \xi - (\ell + 1/2)\varepsilon_0 - \frac{3\delta}{2}.$$

If we denote the prover's state after step 3(c)(iii) by $\boldsymbol{\rho}$, it follows from property 1 of Lemma 3.2 that

$$\mathbb{E}_{\bar{q} \leftarrow Q_\ell^{(k)}} \left[p \;\middle|\; (p, \boldsymbol{\rho}') \leftarrow \mathsf{ValEst}_{V_{\tau_\ell}, U^{(\ell+1)}}(U_\ell(\bar{q}_\ell)\boldsymbol{\rho}, \varepsilon, \delta) \right] \geq \xi - (\ell + 1/2)\varepsilon_0 - \frac{3\delta}{2}.$$

Thus, by a simple Markov argument, it follows that step 3(c)(iv) will succeed with probability at least $\varepsilon_0/3 - \delta$ because $\bar{q}$ is sampled from the uniform distribution. Therefore, we can apply a Chernoff bound and bound the probability of aborting in step 3(c)(iv), which requires $\mathsf{iter} = \lambda/\varepsilon_0$ consecutive failures, by $\mathsf{negl}(\lambda)$.

Thus, by a union bound, the probability of aborting in step 3 is at most

$$m \cdot (1/km + \mathsf{negl}(\lambda)) + m \cdot \mathsf{negl}(\lambda) = 1/k + \mathsf{negl}(\lambda),$$

provided $m = \mathrm{poly}(\lambda)$. □

Combining the two claims with another a union bound gives an overall probability of at most

$$\mathsf{negl}(\lambda) + (1/k + \mathsf{negl}(\lambda)) = 1/k + \mathsf{negl}(\lambda),$$

as desired. □

Next, if we let $\mathcal{H}_j$ also denote the distribution of outputs that a random execution of $\mathcal{H}_j$ induces, then we claim the following is true:

Lemma 5.5. *For* $0 \leq j \leq m-1$, $\mathsf{TD}(\mathcal{H}_j, \mathcal{H}_{j+1}) \leq \sqrt{-\log(\xi/3m^2)/k} + 1/km + \mathsf{negl}(\lambda)$.

Before we prove Lemma 5.5, we first show how Lemmas 5.3, 5.4, and 5.5 can be used to establish soundness.

Observe that in hybrid $\mathcal{H}_m$, τ_m is *independent* of i. Therefore, we have that

$$\begin{aligned}\Pr_{(i,\tau_m)\leftarrow\mathcal{H}_m}[\mathsf{Accept}_i(\tau_m)=1] &\geq \Pr_{(i,\tau_m)\leftarrow\mathcal{H}_m}[\mathsf{Accept}_i(\tau_m)=1 \mid \mathsf{Accept}(\tau_m)=1]\cdot \Pr_{(i,\tau_m)\leftarrow\mathcal{H}_m}[\mathsf{Accept}(\tau_m)=1]\\ &= \frac{t}{k}\cdot \Pr_{(i,\tau_m)\leftarrow\mathcal{H}_m}[\mathsf{Accept}(\tau_m)=1].\end{aligned}$$

Consequently, it follows that

$$\begin{aligned}\Pr\Big[\langle(\mathcal{A},\widetilde{V})(\lambda,x)\rangle=1\Big] &= \Pr_{(i,\tau_m)\leftarrow\mathcal{H}_0}[\mathsf{Accept}_i(\tau_m)=1]\\ &\geq \Pr_{(i,\tau_m)\leftarrow\mathcal{H}_m}[\mathsf{Accept}_i(\tau_m)=1] - \sum_{j=0}^{m-1}\mathsf{TD}(\mathcal{H}_{j+1},\mathcal{H}_j)\\ &\geq \frac{t}{k}\cdot\Pr_{(i,\tau_m)\leftarrow\mathcal{H}_m}[\mathsf{Accept}(\tau_m)=1] - \sum_{j=0}^{m-1}\mathsf{TD}(\mathcal{H}_{j+1},\mathcal{H}_j)\\ &\geq \frac{t}{k}\left(1-\Pr_{(i,\tau_m)\leftarrow\mathcal{H}_m}[\tau_m=\perp]\right) - \sum_{j=0}^{m-1}\mathsf{TD}(\mathcal{H}_{j+1},\mathcal{H}_j)\\ &\geq \frac{t}{k}-\frac{1}{k}-\mathsf{negl}(\lambda) - m\cdot\left[\sqrt{\frac{-\log(\xi/3m^2)}{k}}+\frac{1}{km}+\mathsf{negl}(\lambda)\right]\\ &\geq \frac{t}{k}-\frac{1}{k}-\mathsf{negl}(\lambda) - m\cdot\left[\sqrt{\frac{-\log(\xi/3m^2)}{k}}+\frac{1}{km}+\mathsf{negl}(\lambda)\right]\\ &\geq \frac{t}{k}-\frac{2}{k}-m\sqrt{\frac{-\log(\xi/3m^2)}{k}}-\mathsf{negl}(\lambda),\\ &\geq \frac{t}{k}-2m\sqrt{\frac{-\log(\xi/3m^2)}{k}}-\mathsf{negl}(\lambda),\end{aligned}$$

as desired, which concludes the proof of Theorem 5.1. □

Thus, it remains to prove Lemma 5.5:

Proof of Lemma 5.5. It is easy to see that in the first $j-1$ rounds, $\mathcal{H}_j$ and $\mathcal{H}_{j+1}$ are identical. Note that the sampling of queries in rounds $j+1$ and onwards is also identical, so it suffices to show that at the end of the jth round (i.e. when step 3(d) is reached), the prover states in $\mathcal{H}_j$ and $\mathcal{H}_{j+1}$ are close in trace distance.

In the case of an abort in step 3(c)(iv) when $s = \mathsf{iter}$, we define the ending prover state to be any arbitrary quantum state (denoted by $\perp$). Without loss of generality, we can assume that both $\mathcal{H}_j$ and $\mathcal{H}_{j+1}$ start with the same fixed quantum state at the beginning of the jth iteration of step 3.

We now consider a new series of hybrids relating $\mathcal{H}_j$ to $\mathcal{H}_{j+1}$. For every $h \in [\mathsf{iter}]$, define $\mathcal{H}_{j,h}$ to be the algorithm which behaves identically to $\mathcal{H}_j$ except when $\ell = j$ and $s \in \{1,\ldots,h\}$, it replaces

the conditional sampling of $\bar{q}$ in step 3(c)(i) with a uniform sample. By definition, we have that $\mathcal{H}_{j,0} := \mathcal{H}_j$ and $\mathcal{H}_{j,\mathsf{iter}} := \mathcal{H}_{j+1}$. Now, for each hybrid $\mathcal{H}_{j,h}$, let $\mathcal{D}_h$ denote the distribution of prover states at the end of the jth round and $\boldsymbol{s}_h$ denote the distribution of stopping times, which is the iteration on which the $\mathsf{CheckCoins}$ procedure first succeeds in round j (or $\mathsf{iter}+1$ if $\mathsf{CheckCoins}$ never succeeds).

Our goal is to show that $\mathcal{H}_{j,0}$ and $\mathcal{H}_{j,\mathsf{iter}}$ are close in trace distance:

$$\mathsf{TD}(\mathcal{H}_{j,0}, \mathcal{H}_{j,\mathsf{iter}}) \leq \sqrt{-\log(\xi/3m^2)/k} + 1/km + \mathsf{negl}(\lambda). \tag{4}$$

We prove Equation (4) by showing that for every $h \in [\mathsf{iter}]$, $\mathcal{H}_{j,h-1}$ and $\mathcal{H}_{j,h}$ are close in trace distance:

Claim 5.5.1. *For all $1 \leq h \leq \mathsf{iter}$,*

$$\mathsf{TD}(\mathcal{H}_{j,h-1}, \mathcal{H}_{j,h}) \leq \Pr[\boldsymbol{s}_{h-1} = h] \cdot \sqrt{\frac{-\log(\xi/3m^2)}{k}} + \frac{1}{km \cdot \mathsf{iter}} + \mathsf{negl}(\lambda).$$

Proof of Equation (4) *assuming Claim 5.5.1.* We first observe that for all $h < h'$, $\mathcal{H}_{j,h}$ and $\mathcal{H}_{j,h'}$ behave identically in the ℓth round up until the hth iteration (when $s = h$). This means that the trace distance between the states in $\mathcal{H}_{j,h}$ and $\mathcal{H}_{j,h'}$ up until step 3(c)(iv) in iteration h is zero. Additionally, since the conditionally sampled query in the $(h+1)$th iteration in $\mathcal{H}_{j,h}$ is drawn from an identical distribution as the uniformly sampled query given an arbitrary starting quantum state (since it has no dependence on i), for all $s' \leq h+1$, $\Pr[\boldsymbol{s}_h = s'] = \Pr[\boldsymbol{s}_{h'} = s']$.

Therefore, we have that

$$\begin{aligned}
\mathsf{TD}(\mathcal{H}_{j,0}, \mathcal{H}_{j,\mathsf{iter}}) &\leq \sum_{h=1}^{\mathsf{iter}} \mathsf{TD}(\mathcal{H}_{j,h-1}, \mathcal{H}_{j,h}) \\
&\leq \sum_{h=1}^{\mathsf{iter}} \left(\Pr[\boldsymbol{s}_{h-1} = h] \cdot \sqrt{\frac{-\log(\xi/3m^2)}{k}} + \frac{1}{km \cdot \mathsf{iter}} + \mathsf{negl}(\lambda) \right) \\
&= \sum_{h=1}^{\mathsf{iter}} \left(\Pr[\boldsymbol{s}_{\mathsf{iter}} = h] \cdot \sqrt{\frac{-\log(\xi/3m^2)}{k}} \right) + \frac{1}{km} + \mathsf{negl}(\lambda) \\
&\leq \sqrt{\frac{-\log(\xi/3m^2)}{k}} + \frac{1}{km} + \mathsf{negl}(\lambda),
\end{aligned}$$

as desired. □

We now prove Claim 5.5.1:

Proof of Claim 5.5.1. Fixing h, we observe (as noted earlier) that $\mathcal{H}_{j,h-1}$ and $\mathcal{H}_{j,h}$ behave identically in the first $h-1$ iterations, so $\mathsf{TD}(\mathcal{D}_{h-1}|_{\boldsymbol{s}_{h-1}<h}, \mathcal{D}_h|_{\boldsymbol{s}_h<h}) = 0$ and thus $\Pr[\boldsymbol{s}_{h-1} < h] = \Pr[\boldsymbol{s}_h < h]$. Additionally, since the first $h-1$ iterations have no dependence on i, we also have (as noted earlier) that $\Pr[\boldsymbol{s}_{h-1} = h] = \Pr[\boldsymbol{s}_h = h]$.

Without loss of generality, we can fix the state $\boldsymbol{\rho}$ in $\mathcal{H}_{j,h-1}$ before the start of the hth iteration (and assume this is the same starting state as in $\mathcal{H}_{j,h}$). A simple coupling argument now provides a bound on the desired trace distance:

$$\begin{aligned}
\mathsf{TD}(\mathcal{H}_{j,h-1}, \mathcal{H}_{j,h}) \leq \Pr[\boldsymbol{s}_{h-1} = h] \cdot \mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}=h}, \mathcal{H}_{j,h}|_{\boldsymbol{s}_h=h}) \\
+ \Pr[\boldsymbol{s}_{h-1} > h] \cdot \mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}>h}, \mathcal{H}_{j,h}|_{\boldsymbol{s}_h>h}).
\end{aligned}$$

We now bound this expression with the following two propositions:

Proposition 5.6. $\mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}=h}, \mathcal{H}_{j,h}|_{\boldsymbol{s}_h=h}) \leq \sqrt{-\log(\xi/3m^2)/k} + \mathsf{negl}(\lambda)$.

Proof of Proposition 5.6. First, it is easy to see that $\mathcal{H}_{j,s'-1}$ and $\mathcal{H}_{j,s'}$ behave identically before the s'th iteration. Thus, it suffices to bound the statistical distance between the joint distributions of accepting projections/queries with i, as the mixed states after continuing to the next round in both hybrids are entirely determined by the state before the beginning of the previous round and the queries being tested.[16]

Defining W to be the event which occurs when the $\mathsf{CheckCoins}$ returns some p such that $p \geq \xi - (\ell+1)\varepsilon_0$ given a fixed state $\boldsymbol{\rho}$ and projection $\Pi_{\bar{q}}$, we have that

$$\mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}=h}, \mathcal{H}_{j,h}|_{\boldsymbol{s}_h=h}) = \mathop{\mathbb{E}}_{i\leftarrow[k]}[\mathsf{TD}((i,(Q_\ell^{(k)}|W)),(i,(Q_\ell^{(k)}|W,(Q_\ell^{(k)})_i)))].$$

Since $\Pr[W] \geq \varepsilon_0/3 - \mathsf{negl}(\lambda)$ on average across all queries as previously argued by a simple application of a Markov bound, by Lemma 3.7 we have that

$$\begin{aligned}\mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}=h}, \mathcal{H}_{j,h}|_{\boldsymbol{s}_h=h}) &= \mathop{\mathbb{E}}_{i\leftarrow[k]}[\mathsf{TD}((i,(Q_\ell^{(k)}|W)),(i,(Q_\ell^{(k)}|W,(Q_\ell^{(k)})_i)))] \\ &\leq \sqrt{-\log(\Pr[W])/k} \leq \sqrt{\log(3/\varepsilon_0 + \mathsf{negl}(\lambda))/k} \\ &= \sqrt{\frac{-\log(\xi/3m^2)}{k}} + \mathsf{negl}(\lambda).\end{aligned}$$

□

Proposition 5.7. $\Pr[\boldsymbol{s}_{h-1} > h] \cdot \mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}>h}, \mathcal{H}_{j,h}|_{\boldsymbol{s}_h>h}) \leq \frac{1}{km\cdot\mathsf{iter}} + \mathsf{negl}(\lambda)$.

Proof of Proposition 5.7. When $h < \mathsf{iter}$, as noted earlier, it suffices to consider the state at the start of the hth iteration of step 3(c). Observing that this state has no dependence on i or q_j, this implies that the sampling of Π in both hybrids is drawn from the same uniform distribution over $\mathcal{P}$. Therefore, we can apply Lemma 4.1 and note that by a coupling argument, we have that

$$\mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}>h}, \mathcal{H}_{j,h}|_{\boldsymbol{s}_h>h}) \leq \mathsf{TD}(\mathcal{D}_{h-1}|_{\boldsymbol{s}_{h-1}>h}, \mathcal{D}_h|_{\boldsymbol{s}_h>h}) \leq \frac{\mathsf{TD}(\mathcal{D}_{h-1},\mathcal{D}_h)}{\Pr[\boldsymbol{s}_{h-1}>h]} \leq \frac{N_{\varepsilon,\delta}\cdot\eta}{\Pr[\boldsymbol{s}_{h-1}>h]},$$

and so

$$\Pr[\boldsymbol{s}_{h-1} > h] \cdot \mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}>h}, \mathcal{H}_{j,h}|_{\boldsymbol{s}_h>h}) \leq N_{\varepsilon,\delta}\cdot\eta \leq \frac{1}{km\cdot\mathsf{iter}}.$$

On the other hand, through an identical argument as in Lemma 5.3, we observe that

$$\Pr[\boldsymbol{s}_{\mathsf{iter}-1} > \mathsf{iter}] = \mathsf{negl}(\lambda),$$

and thus

$$\Pr[\boldsymbol{s}_{\mathsf{iter}-1} > \mathsf{iter}] \cdot \mathsf{TD}(\mathcal{H}_{j,\mathsf{iter}-1}|_{\boldsymbol{s}_{\mathsf{iter}-1}>\mathsf{iter}}, \mathcal{H}_{j,\mathsf{iter}}|_{\boldsymbol{s}_{\mathsf{iter}}>\mathsf{iter}}) \leq \Pr[\boldsymbol{s}_{\mathsf{iter}-1} > \mathsf{iter}]\cdot 1 = \mathsf{negl}(\lambda),$$

which finishes the proof. □

[16] Note that although round $j+1$ of our hybrids use $\bar{q}_j$, upon acceptance, $\bar{q}_j$ itself contains no information about i, so applications of M^{j+1} in step 3(c) leak no additional information about i.

Thus, we have that

$$\begin{aligned}\mathsf{TD}(\mathcal{H}_{j,h-1},\mathcal{H}_{j,h}) &\leq \Pr[\boldsymbol{s}_{h-1}=h]\cdot\mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}=h},\mathcal{H}_{j,h}|_{\boldsymbol{s}_h=h})\\ &\quad+\Pr[\boldsymbol{s}_{h-1}>h]\cdot\mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}>h},\mathcal{H}_{j,h}|_{\boldsymbol{s}_h>h})\\ &\leq \Pr[\boldsymbol{s}_{h-1}=h]\cdot\sqrt{\frac{-\log(\xi/3m^2)}{k}}+\frac{1}{km\cdot\mathsf{iter}}+\mathsf{negl}(\lambda),\end{aligned}$$

as claimed, concluding the proof of Claim 5.5.1. □

Since Equation (4) is equivalent to our claim, this concludes the proof of Proposition 5.5. □

6 Parallel Repetition for Three-Message Protocols

In this section, we analyze the soundness of parallel repetition of private-coin protocols that consist of three messages. Although exponentially-decreasing soundness from parallel repetition for three-message arguments was already demonstrated through other means in [BQSY24], we provide a simpler proof of parallel repetition (for protocols where the verifier is classical) and generalize from the direct-product setting to the setting of threshold verifiers.

In the private-coin setting, one cannot compute the predicate of the external verifier, and so a proxy for the external verifier's verdict must be used. Existing classical reductions take two general approaches: the correlation reduction strategy of [CHS05] and the *soft-decision* procedure of [BIN97] and [HPPW10]. [BQSY24] provides a quantum analogue of correlation reduction, while we consider an analogue of the soft-decision procedure (see Section 2 for more details).

6.1 Our Results

We prove the following parallel repetition theorem for three-message protocols:

Theorem 6.1. *Let (P,V) be a three-message interactive argument. Suppose that there exists a quantum adversary $\mathcal{B}$, $\xi=\xi(\lambda)>0$, and polynomials $1\leq t=t(\lambda)\leq k=k(\lambda)$ such that for all $\lambda\in\mathbb{N}$,*

$$\Pr\Big[\langle(\mathcal{B},V^{(t,k)})(1^\lambda)\rangle=1\Big]\geq\xi. \tag{5}$$

Then there exists a quantum adversary $\mathcal{A}$ running in time poly$(|\mathcal{B}|,\lambda,1/\xi)$ and a negligible function negl such that for every $\lambda\in\mathbb{N}$,

$$\Pr\Big[\langle(\mathcal{A},V)(1^\lambda)\rangle=1\Big]\geq\frac{t}{k}-\frac{2\log k}{\sqrt{k}}-3\sqrt{\frac{-\log\xi}{k}}-\mathsf{negl}(\lambda).^{17}$$

Note that Theorem 6.1 implies the following corollary:

Corollary 6.2. *Let (P,V) be a three-message interactive argument and let $\varepsilon=\varepsilon(\lambda)$ be any parameter such that for every QPT adversary $\mathcal{A}$ there exists a negligible function negl such that for every $\lambda\in\mathbb{N}$,*

$$\Pr\Big[\langle(\mathcal{A},V)(1^\lambda)\rangle=1\Big]\leq\varepsilon(\lambda)+\mathsf{negl}(\lambda).$$

[17] We remark that the precise constants here can be improved by tweaking the parameters of our reduction, but we will keep these for ease of presentation.

Fix any polynomials $1 \leq t = t(\lambda) \leq k = k(\lambda)$ such that $\varepsilon < \frac{t}{k} - \frac{2\log k}{\sqrt{k}}$.[18] *Then for every* QPT *adversary $\mathcal{B}$ there exists a negligible function* negl *such that for every $\lambda \in \mathbb{N}$,*

$$\Pr\Big[\langle(\mathcal{B}, V^{(t,k)})(1^\lambda)\rangle = 1\Big] \leq \max\{2\exp\left(\frac{-k}{9}\cdot\left(\frac{t-2\sqrt{k}\log k}{k} - \varepsilon\right)^2\right), \mathsf{negl}(\lambda)\}.$$

Proof of Corollary 6.2. Suppose for the sake of contradiction that there exists a QPT adversary $\mathcal{B}$ such that $\mathcal{B}$ succeeds against $V^{(t,k)}$ with non-negligible probability $\xi > 2\exp\left(\frac{-k}{9}\cdot\left(\frac{t-2\sqrt{k}\log k}{k} - \varepsilon\right)^2\right)$. Applying Theorem 6.1 then gives a quantum adversary $\mathcal{A}$ running in time $\mathrm{poly}(\lambda, 1/\xi)$ such that

$$\begin{aligned}
\Pr\Big[\langle(\mathcal{A}, V)(1^\lambda)\rangle = 1\Big] &\geq \frac{t}{k} - \frac{2\log k}{\sqrt{k}} - 3\sqrt{\frac{-\log\left(2\exp\left(-\frac{k}{9}\left(\frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \varepsilon\right)^2\right)\right)}{k}} - \mathsf{negl}(\lambda)\\
&= \frac{t}{k} - \frac{2\log k}{\sqrt{k}} - 3\sqrt{\frac{-\log\left(\exp\left(-\frac{k}{9}\left(\frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \varepsilon\right)^2\right)\right) - 1}{k}} - \mathsf{negl}(\lambda)\\
&= \frac{t}{k} - \frac{2\log k}{\sqrt{k}} - 3\sqrt{\frac{\frac{k}{9}\cdot\left(\frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \varepsilon\right)^2 - 1}{k}} - \mathsf{negl}(\lambda)\\
&= \frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \sqrt{\left(\frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \varepsilon\right)^2 - \frac{9}{k}} - \mathsf{negl}(\lambda).
\end{aligned}$$

Observe that since $\xi \leq 1$, this means that $c := 2\exp\left(\frac{-k}{9}\cdot\left(\frac{t-2\sqrt{k}\log k}{k} - \varepsilon\right)^2\right) < \xi \leq 1$ and so $3\sqrt{\frac{-\log c}{k}} = \sqrt{\left(\frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \varepsilon\right)^2 - \frac{9}{k}} > 0$. Since $\sqrt{x^2 - y} \leq x - \frac{y}{2x}$ and hence $-\sqrt{x^2-y} \geq -x + \frac{y}{2x}$ for $x, y > 0$ such that $x^2 - y > 0$, we have that

$$\begin{aligned}
\Pr\Big[\langle(\mathcal{A}, V)(1^\lambda)\rangle = 1\Big] &\geq \frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \sqrt{\left(\frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \varepsilon\right)^2 - \frac{9}{k}} - \mathsf{negl}(\lambda)\\
&\geq \frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \left(\frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \varepsilon\right) + \frac{9/k}{2\left(\frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \varepsilon\right)} - \mathsf{negl}(\lambda)\\
&= \varepsilon + \frac{9/k}{2\left(\frac{t}{k} - \frac{2\log k}{\sqrt{k}} - \varepsilon\right)} - \mathsf{negl}(\lambda)\\
&\geq \varepsilon + \frac{9/k}{2t/k} - \mathsf{negl}(\lambda) = \varepsilon + \frac{9}{2t} - \mathsf{negl}(\lambda).
\end{aligned}$$

Since t is polynomial, $\varepsilon + \frac{9}{2t} - \mathsf{negl}(\lambda) > \varepsilon + \mathsf{negl}(\lambda)$.

But if $\mathcal{B}$ has non-negligible advantage, then $1/\xi$ is polynomial in λ and so $\mathcal{A}$ is QPT, in contradiction to the original assumption that no such QPT adversary exists. Thus, no such QPT adversary $\mathcal{B}$ exists either. □

[18] The condition that $\varepsilon < \frac{t}{k} - \frac{2\log k}{\sqrt{k}}$ is similar to the condition in Corollary 5.2 but with some minor loss.

Proof of Theorem 6.1. Denote by $|\psi\rangle$ the initial state of $\mathcal{B}$, and let $\Lambda = \text{poly}(\lambda)$ be the number of qubits in $|\psi\rangle$. For the sake of simplicity, we assume without loss of generality that $|\psi\rangle$ contains designated registers $\mathcal{Z}_1$ and $\mathcal{Z}_2$, where $\mathcal{Z}_j$ is the register on which $\mathcal{B}$ writes its jth message. Since the prover's unitary in the first round is efficiently implementable and independent of the verifier's queries, we also assume without loss of generality that $\mathcal{B}$ determines its first round message by simply measuring the $\mathcal{Z}_1$ register of $|\psi\rangle$. We denote all the other registers in $|\psi\rangle$ by $\mathcal{I}$.

We denote by Q the distribution of queries of the single-fold verifier in the protocol (P, V); let $Q^{(k)}$ refer to the distribution of the queries sent in the k-fold repeated protocol $(P^{(k)}, V^{(t,k)})$ and $\bar{q}^i$ refer to the ith coordinate of the query vector $\bar{q}$.

We denote by $U_{\bar{q}}$ (which operates on registers $\mathcal{I}$ and $\mathcal{Z}_2$) the unitary that $\mathcal{B}$ applies to its quantum state upon receiving query $\bar{q}$ before measuring the $\mathcal{Z}_2$ register to compute its second round response. As in Section 5, the assumption that $U_{\bar{q}}$ is a unitary is without loss of generality, as any adversary not of this form can be "purified" into an adversary with unitary strategies, identical observable behavior, and constant factor blowup in size.

For a k-fold transcript $\tau = (\bar{z}_1, \bar{q}, \bar{z}_2)$, where $\bar{q}$ is generated with verifier internal randomness $\bar{r} = (\bar{r}^1, \ldots, \bar{r}^k)$, let $\tau^j = (\bar{z}_1^j, \bar{q}^j, \bar{z}_2^j)$ and $\bar{r}^j$ be the view and internal randomness of the j'th execution. Let $\mathsf{Accept}_j(\bar{r}, \tau) = 1$ if and only if the (single-execution) verifier accepts τ^j with respect to internal randomness $\bar{r}^j$. Similarly, given the verifier's internal randomness $\bar{r}^{(-i)}$ in all but the ith execution, one can define $\mathsf{Accept}_j(\bar{r}^{(-i)}, \tau) = 1$ for $j \neq i$ to occur exactly when the (single-execution) verifier accepts τ^j with respect to internal randomness $\bar{r}^j$. Again, for fixed threshold t we define $\mathsf{Accept}(\bar{r}, \tau) = 1$ iff $\sum_i \mathsf{Accept}_i(\bar{r}, \tau) \geq t$. We also define the classical function $\mathsf{SoftDecision}_{\nu,t}(i, \bar{r}^{(-i)}, \tau, \omega)$ parametrized by a threshold $0 \leq t \leq k$ and smoothness parameter $\nu > 0$ which takes as input an index $i \in [k]$, verifier internal randomness $\bar{r}^{(-i)}$, transcript τ, and random string $\omega \in \{0,1\}^{\text{poly}(\lambda)}$, and outputs a bit $b \in \{0,1\}$ such that if $\sum_{j \neq i} \mathsf{Accept}_j(\bar{r}^{(-i)}, \tau) = \ell$, then

$$\Pr_{\omega \leftarrow \{0,1\}^{\text{poly}(\lambda)}}[\mathsf{SoftDecision}_{\nu,t}(i, \bar{r}^{(-i)}, \tau, \omega) = 1] = \min\{1, 2^{\nu(\ell+1-t)}\}.$$

One can think of $\mathsf{SoftDecision}$ as using ω as its random coins to decide whether to accept with probability $\min\{1, 2^{\nu(\ell+1-t)}\}$.

Remark 6.3. Recall that in Lemma 3.2, ValEst is defined with respect to a verdict function V and adversary circuit A, where A takes as input randomness u and returns a response z, and V is a deterministic function of u and z. For our setting, we take $A := A_{\bar{z}_1}$ to be the algorithm that has the first message $\bar{z}_1$ of $\mathcal{B}$ hardwired and on input $\bar{r}$, which represents the verifier's random coins, computes the verifier's message $\bar{q}$ corresponding to the randomness $\bar{r}$, and then applies $U_{\bar{q}}$ to obtain $\bar{z}_2$, and outputs $(\bar{q}, \bar{z}_2)$. Now, we define $V := V_{\bar{z}_1}$ to be the verdict function which takes as input $\bar{r}, (\bar{q}, \bar{z}_2)$ and returns $\mathsf{Accept}(\bar{r}, \bar{z}_1 || \bar{q} || \bar{z}_2)$.[19] Note that the estimated success probability in this ValEst is indeed the same success probability of the original prover $\mathcal{B}$ in the three-message protocol.

The general framework of our reduction is the same as in the public-coin setting, so we refer the reader to the informal explanation given in Section 5 for some intuition behind the reduction in the three-message setting as well.

Taking ValEst, $\mathsf{Prepare}$, and Repair' to be the algorithms defined in Lemmas 3.2 and 4.1, our algorithm $\mathcal{A}$ is described in the figure below (Figure 2).

[19] Note that $\bar{q}$ is technically redundant since it can be computed from $\bar{r}$ and $\bar{z}_1$. We include it only for notational convenience.

Projection 2 : $\underline{\mathsf{SoftDecisionProj}_{\bar{z}_1,j,\bar{r}^{(-j)},q,\omega}[\boldsymbol{\rho}]}$

1. Use $\bar{r}^{(-j)}$ to compute query $\bar{q}^{(-j)}$ and complete $\bar{q}^{(-j)}$ with $\bar{q}^j = q$ to get query $\bar{q}$.
2. Apply $U_{\bar{q}}$ to $\boldsymbol{\rho}$, resulting in state $\boldsymbol{\rho}' = \sum_{\bar{z}_2} (\boldsymbol{\rho}'_{\bar{z}_2})_{\mathcal{I},\mathcal{Z}_1} \otimes |\bar{z}_2\rangle \langle \bar{z}_2|_{\mathcal{Z}_2}$.
3. Apply the classical function $\mathsf{SoftDecision}_{\nu,t}(j, \bar{r}^{(-j)}, \bar{z}_1||\bar{q}||\cdot, \omega)$ to $\boldsymbol{\rho}' \otimes |0\rangle\langle 0|_{\mathcal{E}}$ coherently over responses $\bar{z}_2$ and write the outcome in register $\mathcal{E}$, resulting in state $\sum_{\bar{z}_2} (\boldsymbol{\rho}'_{\bar{z}_2})_{\mathcal{I},\mathcal{Z}_1} \otimes |\bar{z}_2\rangle\langle \bar{z}_2|_{\mathcal{Z}_2} \otimes |b_{\bar{z}_2}\rangle\langle b_{\bar{z}_2}|_{\mathcal{E}}$.
4. Measure and discard register $\mathcal{E}$, obtaining b and remaining state $\boldsymbol{\rho}''$.
5. Apply $U_{\bar{q}}^{\dagger}$ to get state $\boldsymbol{\sigma} = \boldsymbol{\rho}'''_{\mathcal{I},\mathcal{Z}_1,\mathcal{Z}_2}$ before returning $(\boldsymbol{\rho}''', b)$.

Algorithm 2 : The Algorithm $\mathcal{A}$

Let $\mathsf{iter} := 4\lambda/\xi$ and $\Lambda := \mathsf{size}(|\psi\rangle)$ be the number of qubits of $|\psi\rangle$. $\mathcal{A}$ begins with state $|\psi\rangle^{\otimes \mathsf{iter}}$, and runs as follows:

1. Set $\varepsilon_0 := \xi/4, \varepsilon := \frac{\varepsilon_0}{16\cdot\mathsf{iter}}, \delta := \min\{2^{-\lambda}, 2^{-k}\}, \eta := \frac{1}{4k\cdot\mathsf{iter}}/N_{\varepsilon,\delta}$, $\nu := \sqrt{-\log \xi/k}$, where $N_{\varepsilon,\delta}$ is the parameter from Lemma 3.2. Sample $i \leftarrow [k]$.
2. For $s = 1$ to iter:
   (a) Set $\boldsymbol{\rho} = |\psi\rangle$ to be $\mathcal{A}$'s sth copy of $|\psi\rangle$. Measure the $\mathcal{Z}_1$ register to get response $\bar{z}_1$ and leftover state $\boldsymbol{\rho}_0$. Run $(p_0, \boldsymbol{\rho}_1) \leftarrow \mathsf{ValEst}_{V_{\bar{z}_1}, A_{\bar{z}_1}}(\boldsymbol{\rho}_0, \varepsilon, \delta)$.
      i. If $p_0 \geq \xi - \varepsilon_0$, fix $\bar{z}_1$ and send $\bar{z}_1^i$ to V before continuing to step 3.
      ii. Else, if $p_0 < \xi - \varepsilon_0$ and $s = \mathsf{iter}$, return $(i, \perp)$.
3. Let $\mathcal{P} := \{\mathsf{SoftDecisionProj}_{\bar{z}_1,j,\bar{r}^{(-j)},q,\omega}\}_{j\in[k],\bar{r}^{(-j)}\in\{0,1\}^{\mathrm{poly}(\lambda)},q\in Q,\omega\in\{0,1\}^{\mathrm{poly}(\lambda)}}$ be the projection family defined above. Define the almost projective measurement family $\mathsf{M} := \mathsf{ValEst}_{V_{\bar{z}_1}, A_{\bar{z}_1}}$.
4. Receive the message q from V.
5. For $s = 1$ to iter:
   (a) Generate verifier randomness $\bar{r}^{(-i)} \leftarrow \{0,1\}^{\mathrm{poly}(\lambda)}$ for executions $[k] \setminus \{i\}$ and generate a uniformly random string $\omega \leftarrow \{0,1\}^{\mathrm{poly}(\lambda)}$. Let $\Pi := \mathsf{SoftDecisionProj}_{\bar{z}_1,i,\bar{r}^{(-i)},q,\omega}$.
   (b) Compute $(\boldsymbol{\rho}'_1, p_1) \leftarrow \mathsf{M}_{\varepsilon,\delta}(\boldsymbol{\rho}_1)$.
   (c) Compute $(\boldsymbol{\rho}''_1, p'_1) \leftarrow \mathsf{Prepare}_{\mathsf{M},\mathcal{P}}(\boldsymbol{\rho}'_1, p_1, \varepsilon, \delta, \eta, \Lambda)$.
   If $p'_1 < \xi - \varepsilon_0 - (4s+1)\varepsilon$, return $(i, \perp)$.
   (d) Compute $(\boldsymbol{\sigma}, b) \leftarrow \Pi[\boldsymbol{\rho}''_1]$.
   If $b = 1$, skip to step 6.
   Else, if $b = 0$ and $s = \mathsf{iter}$, return $(i, \perp)$.
   (e) Compute $\boldsymbol{\rho}_1 = \mathsf{Repair}'_{\mathsf{M},\mathcal{P},\Pi}(\boldsymbol{\sigma}, b, p'_1, \varepsilon, \delta, \eta, \Lambda)$.
6. Use $\bar{r}^{(-i)}$ to compute query $\bar{q}^{(-i)}$ and complete $\bar{q}^{(-i)}$ with $\bar{q}^i = q$ to get query $\bar{q}$. Apply $U_{\bar{q}}$ to $\boldsymbol{\sigma}$ before measuring the $\mathcal{Z}_2$ register to get response $\bar{z}_2$.
7. Set $\tau = \bar{z}_1||\bar{q}^*||\bar{z}_2$ and send $\bar{z}_2^i$ back to V.
8. Return (i, τ).

Runtime. The fact that $\mathcal{A}$ runs in time $\mathrm{poly}(|\mathcal{B}|, \lambda, 1/\xi)$ follows from the efficiency of ValEst',

$\mathsf{Prepare}$, and Repair', and the fact that all parameters are polynomial in λ and $1/\xi$.

Soundness. Before we analyze the soundness of our reduction, we note that we can assume without loss of generality that $\xi \gg \delta$, as otherwise our guarantee is essentially trivially true.

We begin by considering the following hybrids $\mathcal{H}_0$ and $\mathcal{H}_1$:

- Hybrid $\mathcal{H}_0$: Run a random execution of $\mathcal{A}$ and return (i, τ).
- Hybrid $\mathcal{H}_1$: In step 5(a), replace i with a uniformly random index $j \leftarrow [k]$, sample a fresh query $q \leftarrow Q$, and define $\Pi := \mathsf{SoftDecisionProj}_{\bar{z}_1, j, \bar{r}^{(-j)}, q, \omega}$ instead. Alternatively, one can think of replacing the conditionally sampled projection Π in step 5(a) with an unconditionally sampled projection $\Pi \leftarrow \mathcal{P}$. Having done this for all iterations, derive a full transcript τ. Now, sample $i \leftarrow [k]$ and return (i, τ).

Note that by definition, $\mathcal{H}_0$ outputs a random transcript as produced by $(\mathcal{A}, V)$, while $\mathcal{H}_1$ outputs a transcript that has *no dependence* on i.

It is easy to see that step 6 of $\mathcal{A}$ undoes the effect of step 5 in $\mathsf{SoftDecisionProj}$, and since step 6 is only reached when an outcome of $b = 1$ is measured, this means that the state after measuring $\bar{z}_2$ has the property that $\mathsf{SoftDecision}_{\nu,t}(j, \bar{r}^{(-j)}, \bar{z}_1 || \bar{q} || \bar{z}_2, \omega) = 1$.

We will mirror the argument from the proof of Theorem 5.1, although with a slightly different ordering.

Lemma 6.4. $\Pr_{(i,\tau) \leftarrow \mathcal{H}_1}[\tau = \bot] \leq 1/2k + \mathsf{negl}(\lambda)$.

Proof of Lemma 6.4. We first bound the probability that $\mathcal{H}_1$ aborts in step 2:

Claim 6.4.1. *$\mathcal{H}_1$ aborts in step 2 with probability at most $\mathsf{negl}(\lambda)$.*

Proof of Claim 6.4.1. First, note that because the starting state $|\psi\rangle$ has success probability which is at least ξ, by property 1 of Lemma 3.2 and Fact 5.2.1, we have that

$$\mathbb{E}[p_0 | (p_0, \boldsymbol{\rho}') \leftarrow \mathsf{ValEst}_{V_{\bar{z}_1}, A_{\bar{z}_1}}(\boldsymbol{\rho}_0, \varepsilon, \delta)] \geq \xi.$$

Recall that since ValEst takes on values in $[-\frac{1}{2}, \frac{3}{2}]$ (see Lemma 3.2), by a simple Markov argument, the probability that any particular iteration of step 2(a) will succeed is

$$\Pr\left[p_0 \geq \xi - \varepsilon_0 \;\middle|\; (p_0, \boldsymbol{\rho}') \leftarrow \mathsf{ValEst}_{V_{\bar{z}_1}, A_{\bar{z}_1}}(\boldsymbol{\rho}_0, \varepsilon, \delta)\right] \geq 2\varepsilon_0/3.$$

Thus, by using a Chernoff bound we conclude that the probability of aborting in step 2 is at most $\mathsf{negl}(\lambda)$. □

We now bound the probability of $\mathcal{H}_1$ aborting in step 5(c) or 5(d):

Claim 6.4.2. *Conditioned on not aborting in step 2, $\mathcal{H}_1$ aborts in step 5 with probability at most $\frac{1}{2k} + \mathsf{negl}(\lambda)$.*

Proof of Claim 6.4.2. We begin by analyzing step 5(b) when $s = 1$: since the last ValEst measurement in step 2(a) before $\mathcal{A}$ moved on to step 3 falls into the same projective family as $\mathcal{M}$, by property 2 of Lemma 3.2, step 5(b) will return $p_1 < \xi - \varepsilon_0 - \varepsilon$ with probability at most δ.

Now, for all iterations $s \geq 1$, if step 5(b) returned some value $p_1 \geq \xi - \varepsilon_0 - (4(s-1)+1)\varepsilon$, then by property 2 of Lemma 4.1, the probability that step 5(c) aborts during iteration s is at most $N_{\varepsilon,\delta} \cdot (\eta + 4\delta)$, where $N_{\varepsilon,\delta} = O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$ is the number of possible outcomes of $\mathsf{CheckCoins}$ (see Lemma 3.2).

Next, we let x_s denote the probability that step 5(d) skips to step 6 in iteration s. By property 3 of Lemma 4.1 and a simple Markov argument,

$$\Pr[\text{step 5(b) in iter. } s+1 \text{ gives } p < \xi - \varepsilon_0 - (4s+1)\varepsilon \mid \text{not aborting in iter. } s] \leq \frac{N_{\varepsilon,\delta}(\eta + 4\delta)}{1 - x_s}.$$

But since we do not skip in iteration s to step 6 with precisely probability $1 - x_s$, conditioned on step 5(b) being good in iteration s, the probability that we fail in step 5(d) *and* step 5(b) is bad in iteration $s+1$ is at most $N_{\varepsilon,\delta} \cdot (\eta + 4\delta)$.

Iterating this argument, we see that the total probability that $\mathcal{H}_1$ aborts in step 5(c) and step 5(b) returns a value below what we expect in any iteration (unless step 5(d) succeeds, in which case we are done) is at most $\delta + \mathsf{iter} \cdot 2N_{\varepsilon,\delta}(\eta + 4\delta) \leq 1/2k + \mathsf{negl}(\lambda)$.

Otherwise, we can assume that in iteration s, no abort has occurred in step 5(c) up to iteration s in round ℓ. In iteration s, given that the last $\mathsf{Prepare}$ returned a $p \geq \xi - \varepsilon_0 - (4s+1)\varepsilon$, we have by property 2 of Lemma 4.1 that any subsequent ValEst would return a $p \geq \xi - \varepsilon_0 - (4s+2)\varepsilon$ with probability at least $1 - \delta$. But since such a ValEst is on expectation equal to the success probability of the prover's state at the start of step 5(d), this success probability is at least

$$(\xi - \varepsilon_0 - (4s+2) \cdot \varepsilon) \cdot (1 - \delta) - \frac{1}{2} \cdot \delta \geq \xi - \varepsilon_0 - (4 \cdot \mathsf{iter} + 2) \cdot \varepsilon - \frac{3\delta}{2} \geq \xi - \frac{3\varepsilon_0}{2} - \frac{3\delta}{2}.$$

We now note that the event that $\sum_{j \neq i} \mathsf{Accept}_j(\bar{r}^{(-j)}, \tau) \geq t-1$ immediately implies that $\mathsf{SoftDecision}$ returns 1 (and in particular this also means that if $\sum_j \mathsf{Accept}_j(\bar{r}, \tau) \geq t$ then $\mathsf{SoftDecision}$ will return 1), so the probability of success at step 5(d) is at least the true success probability of the state at the start of step 5(d). Thus, the probability of succeeding at step 5(d) is always at least

$$\xi - \frac{3\varepsilon_0}{2} - \frac{3\delta}{2} \geq 2\varepsilon_0,$$

for sufficiently large λ. Therefore, we can apply a Chernoff bound and bound the probability of aborting in step 5(d), which requires $\mathsf{iter} = \lambda/\varepsilon_0$ consecutive failures, by $\mathsf{negl}(\lambda)$.

The claim then follows immediately from a union bound. □

The two claims, combined with a second union bound, bound the probability of $\mathcal{H}_1$ aborting in any step by at most

$$\mathsf{negl}(\lambda) + (1/2k + \mathsf{negl}(\lambda)) = 1/2k + \mathsf{negl}(\lambda),$$

as desired. □

Next, if we let $\mathcal{H}_j$ also denote the distribution of outputs that a random execution of $\mathcal{H}_j$ induces, then we claim the following is true:

Lemma 6.5. $\mathsf{TD}(\mathcal{H}_0, \mathcal{H}_1) \leq \sqrt{-\log \xi / k} + 1/4k + \mathsf{negl}(\lambda)$.

Proof of Lemma 6.5. It is easy to see that step 1 of both $\mathcal{H}_0$ and $\mathcal{H}_1$ are identical, so we can assume without loss of generality that both $\mathcal{H}_0$ and $\mathcal{H}_1$ start with the same fixed quantum state at the beginning of step 5.

In the case of an abort in step 5(d) when $s = \mathsf{iter}$, we define the ending prover state to be any arbitrary quantum state (denoted by $\perp$). It suffices to show that if and when step 6 is reached, the prover states in $\mathcal{H}_0$ and $\mathcal{H}_1$ are close in trace distance.

We now consider a new series of hybrids relating $\mathcal{H}_0$ to $\mathcal{H}_1$. Define $\mathcal{H}_{0,h}$ to be the algorithm which behaves identically to $\mathcal{H}_0$ except when $1 \leq s \leq h$, it samples fresh queries $q \leftarrow Q$ in step 5(a)

when determining Π instead of using the verifier's query. By definition, we have that $\mathcal{H}_{0,0} := \mathcal{H}_0$ and $\mathcal{H}_{0,\mathsf{iter}} := \mathcal{H}_1$. Now, for each hybrid $\mathcal{H}_{0,h}$, let $\mathcal{D}_h$ denote the distribution of prover states at the end of step 6 (or steps 5(c)/5(d) if an abort has occurred) and $\boldsymbol{s}_h$ denote the distribution of stopping times, which is the iteration on which the $\mathsf{SoftDecisionProj}$ procedure first succeeds (or $\mathsf{iter}+1$ if $\mathsf{SoftDecisionProj}$ never succeeds).

Therefore, our goal is to show that $\mathcal{H}_{0,0}$ and $\mathcal{H}_{0,\mathsf{iter}}$ are close in trace distance:

$$\mathsf{TD}(\mathcal{H}_{0,0}, \mathcal{H}_{0,\mathsf{iter}}) \leq \sqrt{-\log \xi/k} + 1/4k + \mathsf{negl}(\lambda). \tag{6}$$

We prove Equation (6) by proving that for every $h \in [\mathsf{iter}]$, $\mathcal{H}_{0,h-1}$ and $\mathcal{H}_{0,h}$ are close in trace distance:

Claim 6.5.1. *For all $1 \leq h \leq \mathsf{iter}$,*

$$\mathsf{TD}(\mathcal{H}_{0,h-1}, \mathcal{H}_{0,h}) \leq \Pr[\boldsymbol{s}_{h-1} = h] \cdot \sqrt{-\log \xi/k} + \frac{1}{4k \cdot \mathsf{iter}} + \mathsf{negl}(\lambda).$$

Proof of Equation 6 assuming Claim 6.5.1. We first observe that for all $h < h'$, $\mathcal{H}_{0,h}$ and $\mathcal{H}_{0,h'}$ behave identically up until the hth iteration (when $s = h$). This means that the trace distance between the states in $\mathcal{H}_{0,h}$ and $\mathcal{H}_{0,h'}$ up until step 5(d) in iteration h is zero. Additionally, since the projection Π in the $(h+1)$th iteration in $\mathcal{H}_{0,h}$ is drawn from the same distribution as the in the $(h+1)$th iteration of $\mathcal{H}_{0,h'}$ given an arbitrary starting quantum state which has no dependence on i, for all $s' \leq h+1$, $\Pr[\boldsymbol{s}_h = s'] = \Pr[\boldsymbol{s}_{h'} = s']$.

Therefore, we have that

$$\begin{aligned}
\mathsf{TD}(\mathcal{H}_{0,0}, \mathcal{H}_{0,\mathsf{iter}}) &\leq \sum_{h=1}^{\mathsf{iter}} \mathsf{TD}(\mathcal{H}_{0,h-1}, \mathcal{H}_{0,h}) \\
&\leq \sum_{h=1}^{\mathsf{iter}} \left(\Pr[\boldsymbol{s}_{h-1} = h] \cdot \sqrt{-\log \xi/k} + \frac{1}{4k \cdot \mathsf{iter}} + \mathsf{negl}(\lambda) \right) \\
&= \sum_{h=1}^{\mathsf{iter}} \left(\Pr[\boldsymbol{s}_{\mathsf{iter}} = h] \cdot \sqrt{-\log \xi/k} \right) + \frac{1}{4k} + \mathsf{negl}(\lambda) \\
&\leq \sqrt{-\log \xi/k} + \frac{1}{4k} + \mathsf{negl}(\lambda),
\end{aligned}$$

as desired. □

We now prove Claim 6.5.1:

Proof of Claim 6.5.1. Fixing h, we observe (as noted earlier) that $\mathcal{H}_{0,h-1}$ and $\mathcal{H}_{0,h}$ behave identically in the first $h-1$ iterations, so $\mathsf{TD}(\mathcal{D}_{h-1}|_{\boldsymbol{s}_{h-1}<h}, \mathcal{D}_h|_{\boldsymbol{s}_h<h}) = 0$ and thus $\Pr[\boldsymbol{s}_{h-1} < h] = \Pr[\boldsymbol{s}_h < h]$. Additionally, since the first $h-1$ iterations have no dependence on i, we also have (as noted earlier) that $\Pr[\boldsymbol{s}_{h-1} = h] = \Pr[\boldsymbol{s}_h = h]$.

Without loss of generality, we can fix the state $\boldsymbol{\rho}$ in $\mathcal{H}_{0,h-1}$ before the start of the hth iteration (and assume this is the same starting state as in $\mathcal{H}_{0,h}$). A simple coupling argument now provides a bound on the desired trace distance:

$$\begin{aligned}
\mathsf{TD}(\mathcal{H}_{0,h-1}, \mathcal{H}_{0,h}) \leq \Pr[\boldsymbol{s}_{h-1} = h] \cdot \mathsf{TD}(\mathcal{H}_{0,h-1}|_{\boldsymbol{s}_{h-1}=h}, \mathcal{H}_{0,h}|_{\boldsymbol{s}_h=h}) \\
+ \Pr[\boldsymbol{s}_{h-1} > h] \cdot \mathsf{TD}(\mathcal{H}_{0,h-1}|_{\boldsymbol{s}_{h-1}>h}, \mathcal{H}_{0,h}|_{\boldsymbol{s}_h>h}).
\end{aligned}$$

We can now bound this expression with the following two propositions:

Proposition 6.6. $\Pr[\boldsymbol{s}_{h-1} > h] \cdot \mathsf{TD}(\mathcal{H}_{0,h-1}|_{\boldsymbol{s}_{h-1}>h}, \mathcal{H}_{0,h}|_{\boldsymbol{s}_h>h}) \leq \frac{1}{4k\cdot\mathsf{iter}} + \mathsf{negl}(\lambda)$.

Proof of Proposition 6.6. When $h < \mathsf{iter}$, as noted earlier, it suffices to consider the state at the start of the hth iteration of step 5. Observing that this state has no dependence on i or q (the external verifier's query), this implies that the sampling of Π in both hybrids is drawn from the same uniform distribution over $\mathcal{P}$. Therefore, we can apply Lemma 4.1 and note that by a coupling argument, we have that

$$\mathsf{TD}(\mathcal{H}_{0,h-1}|_{\boldsymbol{s}_{h-1}>h}, \mathcal{H}_{0,h}|_{\boldsymbol{s}_h>h}) \leq \mathsf{TD}(\mathcal{D}_{h-1}|_{\boldsymbol{s}_{h-1}>h}, \mathcal{D}_h|_{\boldsymbol{s}_h>h}) \leq \frac{\mathsf{TD}(\mathcal{D}_{h-1}, \mathcal{D}_h)}{\Pr[\boldsymbol{s}_{h-1} > h]} \leq \frac{N_{\varepsilon,\delta}\cdot\eta}{\Pr[\boldsymbol{s}_{h-1} > h]},$$

and so

$$\Pr[\boldsymbol{s}_{h-1} > h] \cdot \mathsf{TD}(\mathcal{H}_{0,h-1}|_{\boldsymbol{s}_{h-1}>h}, \mathcal{H}_{0,h}|_{\boldsymbol{s}_h>h}) \leq N_{\varepsilon,\delta}\cdot\eta \leq \frac{1}{4k\cdot\mathsf{iter}}.$$

On the other hand, through an identical argument as in Lemma 6.8, we observe that

$$\Pr[\boldsymbol{s}_{\mathsf{iter}-1} > \mathsf{iter}] = \mathsf{negl}(\lambda),$$

and thus

$$\Pr[\boldsymbol{s}_{\mathsf{iter}-1} > \mathsf{iter}] \cdot \mathsf{TD}(\mathcal{H}_{0,\mathsf{iter}-1}|_{\boldsymbol{s}_{\mathsf{iter}-1}>\mathsf{iter}}, \mathcal{H}_{0,\mathsf{iter}}|_{\boldsymbol{s}_{\mathsf{iter}}>\mathsf{iter}}) \leq \Pr[\boldsymbol{s}_{\mathsf{iter}-1} > \mathsf{iter}] \cdot 1 = \mathsf{negl}(\lambda),$$

which finishes the proof. □

Proposition 6.7. $\mathsf{TD}(\mathcal{H}_{j,h-1}|_{\boldsymbol{s}_{h-1}=h}, \mathcal{H}_{j,h}|_{\boldsymbol{s}_h=h}) \leq \sqrt{\frac{-\log\xi}{k}}$.

Proof of Proposition 6.7. First, it is easy to see that $\mathcal{H}_{0,s'-1}$ and $\mathcal{H}_{0,s'}$ behave identically before the s'th iteration. Thus, by a coupling argument, we can fix the state (which does not yet have any dependence on i) at the start of the s'th iteration. Note that since the transcript is fully measured upon success and both hybrids use a query $\bar{q}$ drawn from the same distribution (given a quantum state which has no dependence on i) and first round message $\bar{z}_1$, it suffices to bound the trace distance of (i,τ) and (j,τ) conditioned on success in the s'th iteration, where τ is the transcript outputted by hybrid $\mathcal{H}_{0,s'}$.[20]

To this end, using another coupling argument, it suffices to consider for two fixed indices $i, j \leftarrow [k]$, fixed transcript τ, and fixed internal randomness $\bar{r}$, the potential difference in verdicts of $\mathsf{SoftDecision}_{\nu,t}(i, \bar{r}^{(-i)}, \tau, \omega)$ and $\mathsf{SoftDecision}_{\nu,t}(j, \bar{r}^{(-j)}, \tau, \omega)$. In the case where $i = j$, they behave identically, so we assume that $i \neq j$. Then, note that if $\sum_{n\neq i,j} \mathsf{Accept}_n(\bar{r},\tau) = \ell$, we have that

$$\ell + 1 \leq 1 + \sum_{n\neq i} \mathsf{Accept}_n(\bar{r},\tau), 1 + \sum_{n\neq j} \mathsf{Accept}_n(\bar{r},\tau) \leq \ell + 2,$$

and so for random $i, j \leftarrow [k]$

$$\left|\Pr_{\omega}[\mathsf{SoftDecision}_{\nu,t}(i, \bar{r}^{(-i)}, \tau, \omega) = 1] - \Pr_{\omega}[\mathsf{SoftDecision}_{\nu,t}(j, \bar{r}^{(-j)}, \tau, \omega) = 1]\right| \leq 1 - 2^{-\nu} \leq \nu.$$

Since the probability of overall acceptance (i.e. the $\mathsf{SoftDecision}$ projection returning 1) is the same for random i as it is for random j, we conclude that the trace distance of the outputs of the two hybrids conditioned on success is at most $\nu = \sqrt{\frac{-\log\xi}{k}}$. □

[20] We could equally have considered the transcript outputted by hybrid $\mathcal{H}_{0,s'-1}$, but note that they would behave identically in the s'th iteration.

Thus, we have that

$$\begin{aligned}\mathsf{TD}(\mathcal{H}_{0,h-1},\mathcal{H}_{0,h}) &\le \Pr[\boldsymbol{s}_{h-1}=h]\cdot\mathsf{TD}(\mathcal{H}_{0,h-1}|_{\boldsymbol{s}_{h-1}=h},\mathcal{H}_{0,h}|_{\boldsymbol{s}_h=h})\\ &\quad+\Pr[\boldsymbol{s}_{h-1}>h]\cdot\mathsf{TD}(\mathcal{H}_{0,h-1}|_{\boldsymbol{s}_{h-1}>h},\mathcal{H}_{0,h}|_{\boldsymbol{s}_h>h})\\ &\le \Pr[\boldsymbol{s}_{h-1}=h]\cdot\sqrt{\frac{-\log\xi}{k}}+\frac{1}{4k\cdot\mathsf{iter}}+\mathsf{negl}(\lambda),\end{aligned}$$

as claimed, concluding the proof of Proposition 6.5.1. □

This concludes the proof of Lemma 6.5. □

For complete transcripts τ, we define $\bar{r}_\tau$ to be the internal randomness of the k-fold verifier which is consistent with τ. Similarly, $\bar{r}_\tau^{(-j)}$ refers to the randomness of all but the jth coordinate of the k-fold verifier which is consistent with τ.

Lemma 6.8. $\Pr_{(i,\tau)\leftarrow\mathcal{H}_1,\omega}[\exists j:\mathsf{SoftDecision}_{\nu,t}(j,\bar{r}_\tau^{(-j)},\tau,\omega)=1]\ge 1-\Pr_{(i,\tau)\leftarrow\mathcal{H}_1}[\tau=\perp]$.

Proof of Lemma 6.8. Like in the proof of Lemma 5.3, it suffices to prove that if $\mathcal{H}_1$ does *not* abort and returns a transcript τ, then for the j and $\bar{r}$ corresponding to the last projection computed in step 5(d) (that is, $\bar{r}$ is the completion of $\bar{r}^{(-j)}$ with the randomness used to generate the associated query q), $\mathsf{SoftDecision}_{\nu,t}(j,\bar{r}^{(-j)},\tau,\omega)=1$ and $\bar{r}$ is by definition the verifier randomness which is consistent with τ. This follows directly from the observation that step 6 of $\mathcal{H}_1$ undoes step 5 of the last (successful) $\mathsf{SoftDecisionProj}$ projection and so we can apply Fact 5.2.1. □

Finally, define $\mathsf{Good}_{\nu,t,j}:=\{(\tau,\omega)\mid\mathsf{SoftDecision}_{\nu,t}(j,\bar{r}_\tau^{(-j)},\tau,\omega)=1\}$ to be the set of transcripts τ and randomness ω such that the $\mathsf{SoftDecision}$ function returns 1 on τ given randomness ω. Then we have the following:

Lemma 6.9. *For all $\nu>0$, $t=t(\lambda)$, and j,*

$$\Pr_{(i,\tau)\leftarrow\mathcal{H}_1,\omega}\left[\mathsf{Accept}_i(\bar{r}_\tau,\tau)=1\mid(\tau,\omega)\in\mathsf{Good}_{\nu,t,j}\right]\ge\frac{t}{k}-\frac{\log k}{\sqrt{k}}-2\sqrt{\frac{-\log\xi}{k}}-\frac{2}{\sqrt{k}}-\mathsf{negl}(\lambda).$$

Proof of Lemma 6.9. Note that we can assume that $k\ge 16$ as otherwise the inequality is trivially satisfied. Consider the modified function $\mathsf{SoftDecision}'_{\nu,t}$ which on input $\bar{r}_\tau$, τ, and ω, returns 1 with probability $\min\{1,2^{-\nu(\ell-t)}\}$ where $\ell=\sum_i\mathsf{Accept}_i(\bar{r},\tau)$. We first consider the difference in the probability that $\mathsf{SoftDecision}'_{\nu,t}$ returns 1 and the probability that $\mathsf{SoftDecision}_{\nu,t}(j,\cdot,\cdot,\cdot)$ returns 1 on input τ. It is clear from a coupling argument that for any transcript, the largest possible difference between these two probabilities is $1-2^{-\nu}\le\nu$, so we have that

$$\Pr_{(i,\tau)\leftarrow\mathcal{H}_1,\omega}[\mathsf{SoftDecision}'_{\nu,t}(\bar{r}_\tau,\tau,\omega)=1\mid(\tau,\omega)\in\mathsf{Good}_{\nu,t,j}]\ge 1-\nu.$$

Thus, it suffices to bound

$$\Pr_{(i,\tau)\leftarrow\mathcal{H}_1,\omega}\left[\mathsf{Accept}_i(\bar{r}_\tau,\tau)=1\mid\mathsf{SoftDecision}'_{\nu,t}(\bar{r}_\tau,\tau,\omega)=1\right],$$

since $\Pr_{(i,\tau)\leftarrow\mathcal{H}_1,\omega}\left[\mathsf{Accept}_i(\bar{r}_\tau,\tau)=1\mid(\tau,\omega)\in\mathsf{Good}_{\nu,t,j}\right]$ is at least the product of these two lower bounds.

Here, we use the following lemma from [HPPW10]:

Lemma 6.10 ([HPPW10])**.** *Let $D_1, \ldots, D_k$ be binary random variables where $\Pr\left[\sum_{i=1}^k D_i \geq t\right] \geq \varepsilon$, let $L = \sum_{i=1}^k D_i$, let $t \leq k$, $\nu > 0$, and let W be a binary random variable such that $\Pr[W = 1 \mid L = \ell] = \min(1, 2^{\nu(\ell - t)})$. Then*

$$\frac{1}{k}\sum_{i=1}^{k} \Pr[D_i = 0 \mid W = 1] \leq 1 - \frac{t}{k} + \frac{1}{k\nu}(\log k - \log \varepsilon) + \frac{4}{\nu^2 k^2}.$$

Setting $\nu = \sqrt{-\log \xi / k}$ like in our reduction and letting $D_i := \mathsf{Accept}_i$, we first recall that in every iteration of step 5 of $\mathcal{H}_1$, the $\mathsf{SoftDecisionProj}$ projection in step 5(d) always has a minimum success probability of $2\varepsilon_0 = \xi/2$ for sufficiently large λ, we find that

$$\frac{1}{k}\sum_{i=1}^{k} \Pr_{(i,\tau)\leftarrow\mathcal{H}_1,\omega}[\mathsf{Accept}_i(\bar{r}_\tau, \tau) = 0 \mid \mathsf{SoftDecision}'_{\nu,t}(\bar{r}_\tau, \tau, \omega) = 1] \leq 1 - \frac{t}{k} + \frac{\log k}{\sqrt{k}} + \sqrt{\frac{-\log(\xi/2)}{k}} + \frac{4}{k} + \mathsf{negl}(\lambda)$$

Since τ is independent of i in $\mathcal{H}_1$, this means that

$$\begin{aligned}\Pr_{(i,\tau)\leftarrow\mathcal{H}_1,\omega}[\mathsf{Accept}_i(\bar{r}_\tau, \tau) = 1 \mid \mathsf{SoftDecision}'_{\nu,t}(\bar{r}_\tau, \tau, \omega) = 1] &\geq \frac{t}{k} - \frac{\log k}{\sqrt{k}} - \sqrt{\frac{-\log \xi}{k}} - \frac{1}{\sqrt{k}} - \frac{4}{k} - \mathsf{negl}(\lambda) \\ &\geq \frac{t}{k} - \frac{\log k}{\sqrt{k}} - \sqrt{\frac{-\log \xi}{k}} - \frac{2}{\sqrt{k}} - \mathsf{negl}(\lambda).\end{aligned}$$

Multiplying these two lower bounds, we conclude that

$$\begin{aligned}\Pr_{(i,\tau)\leftarrow\mathcal{H}_1,\omega}\left[\mathsf{Accept}_i(\bar{r}_\tau, \tau) = 1 \mid (\tau, \omega) \in \mathsf{Good}_{\nu,t,j}\right] &\geq (1 - \nu)\left(\frac{t}{k} - \frac{\log k}{\sqrt{k}} - \sqrt{\frac{-\log \xi}{k}} - \frac{2}{\sqrt{k}} - \mathsf{negl}(\lambda)\right) \\ &\geq \frac{t}{k} - \frac{\log k}{\sqrt{k}} - \sqrt{\frac{-\log \xi}{k}} - \frac{2}{\sqrt{k}} - \nu - \mathsf{negl}(\lambda) \\ &= \frac{t}{k} - \frac{\log k}{\sqrt{k}} - 2\sqrt{\frac{-\log \xi}{k}} - \frac{2}{\sqrt{k}} - \mathsf{negl}(\lambda),\end{aligned}$$

as desired. □

Consequently, from Lemmas 6.4, 6.5, 6.8, and 6.9, it follows that

$$\begin{aligned}\Pr\left[\langle(\mathcal{A}, \widetilde{V})(\lambda, x)\rangle = 1\right] &= \Pr_{(i,\tau)\leftarrow\mathcal{H}_0}[\mathsf{Accept}_i(\bar{r}_\tau, \tau) = 1] \\ &\geq \Pr_{(i,\tau)\leftarrow\mathcal{H}_1}[\mathsf{Accept}_i(\bar{r}_\tau, \tau) = 1] - \mathsf{TD}(\mathcal{H}_1, \mathcal{H}_0) \\ &\geq \left(\frac{t}{k} - \frac{\log k}{\sqrt{k}} - 2\sqrt{\frac{-\log \xi}{k}} - \frac{2}{\sqrt{k}} - \mathsf{negl}(\lambda)\right)\left(1 - \Pr_{(i,\tau)\leftarrow\mathcal{H}_1}[\tau = \perp]\right) - \mathsf{TD}(\mathcal{H}_1, \mathcal{H}_0) \\ &\geq \left(\frac{t}{k} - \frac{\log k}{\sqrt{k}} - 2\sqrt{\frac{-\log \xi}{k}} - \frac{2}{\sqrt{k}} - \Pr_{(i,\tau)\leftarrow\mathcal{H}_1}[\tau = \perp] - \mathsf{negl}(\lambda)\right) - \mathsf{TD}(\mathcal{H}_1, \mathcal{H}_0) \\ &\geq \left(\frac{t}{k} - \frac{\log k}{\sqrt{k}} - 2\sqrt{\frac{-\log \xi}{k}} - \frac{2}{\sqrt{k}} - \frac{1}{2k} - \mathsf{negl}(\lambda)\right) - \frac{1}{4k} - \sqrt{\frac{-\log \xi}{k}} - \mathsf{negl}(\lambda) \\ &\geq \frac{t}{k} - \frac{\log k}{\sqrt{k}} - \frac{3}{\sqrt{k}} - 3\sqrt{\frac{-\log \xi}{k}} - \mathsf{negl}(\lambda).\end{aligned}$$

Since we can assume without loss of generality that $k > 8$ (as $\mathcal{A}$ always suceeds with non-negative probability), we have that

$$\begin{aligned}\Pr\Big[\langle(\mathcal{A},\widetilde{V})(\lambda,x)\rangle = 1\Big] &\geq \frac{t}{k} - \frac{\log k}{\sqrt{k}} - \frac{3}{\sqrt{k}} - 3\sqrt{\frac{-\log\xi}{k}} - \mathsf{negl}(\lambda) \\ &\geq \frac{t}{k} - \frac{2\log k}{\sqrt{k}} - 3\sqrt{\frac{-\log\xi}{k}} - \mathsf{negl}(\lambda),\end{aligned}$$

concluding the proof of Theorem 6.1. □

7 Acknowledgments

We would like to deeply thank Rachel Zhang and Fermi Ma for their many valuable discussions and perspectives, as well as useful comments and suggestions which simplified the presentation of our results. We would also like to thank Alex Lombardi for his insights.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-25-C-0300. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

References

[BBK22] Nir Bitansky, Zvika Brakerski, and Yael Tauman Kalai. Constructive post-quantum reductions. In *Annual International Cryptology Conference*, pages 654–683. Springer, 2022.

[BIN97] Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does parallel repetition lower the error in computationally sound protocols? In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 374–383. IEEE, 1997.

[BM88] László Babai and Shlomo Moran. Arthur-merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988.

[BQSY24] John Bostanci, Luowen Qian, Nicholas Spooner, and Henry Yuen. An efficient quantum parallel repetition theorem and applications. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1478–1487, 2024.

[CHS05] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In *Theory of Cryptography Conference*, pages 17–33. Springer, 2005.

[Chu11] Kai-Min Chung. *Efficient parallel repetition theorems with applications to security amplification*. Harvard University, 2011.

[CL10] Kai-Min Chung and Feng-Hao Liu. Parallel repetition theorems for interactive arguments. In *Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings 7*, pages 19–36. Springer, 2010.

[CMSZ22] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. Post-quantum succinct arguments: breaking the quantum rewinding barrier. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 49–58. IEEE, 2022.

[CP15] Kai-Min Chung and Rafael Pass. Tight parallel repetition theorems for public-coin arguments using kl-divergence. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 229–246. Springer, 2015.

[FFRS20] Kun Fang, Omar Fawzi, Renato Renner, and David Sutter. Chain rule for the quantum relative entropy. *Physical review letters*, 124(10):100501, 2020.

[Hai09] Iftach Haitner. A parallel repetition theorem for any interactive argument. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 241–250. IEEE, 2009.

[Hol07] Thomas Holenstein. Parallel repetition: simplifications and the no-signaling case. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 411–419, 2007.

[HPPW10] Johan Håstad, Rafael Pass, Krzysztof Pietrzak, and Douglas Wikström. An efficient parallel repetition theorem. In *Theory of Cryptography: 7th Theory of Cryptography*

Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings 7, pages 1–18. Springer, 2010.

[KW00] Alexei Y. Kitaev and John Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 608–617. ACM, 2000.

[LMS22] Russell WF Lai, Giulio Malavolta, and Nicholas Spooner. Quantum rewinding for many-round protocols. In *Theory of Cryptography Conference*, pages 80–109. Springer, 2022.

[PV07] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. An efficient parallel repetition theorem for arthur-merlin games. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 420–429. ACM, 2007.

[PW07] Krzysztof Pietrzak and Douglas Wikström. Parallel repetition of computationally sound protocols revisited. In Salil P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 86–102. Springer, 2007.

[Raz95] Ran Raz. A parallel repetition theorem. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 447–456, 1995.

[Yao82] Andrew C Yao. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pages 80–91. IEEE, 1982.