A Parallel Repetition Theorem for Any Interactive Argument*

Iftach Haitner†

July 8, 2012

Abstract

A fundamental question in the study of protocols, is characterizing the effect *parallel repetition* has on the soundness error. While parallel repetition reduces the soundness error in interactive proofs and in special cases of interactive arguments (e.g., three-message protocols, Bellare, Impagliazzo, and Naor [FOCS '97], and public-coin protocols, Hästad, Pass, Pietrzak, and Wikström [TCC '10]), Bellare et al. gave an example of an interactive argument for which parallel repetition *does not* reduce the soundness error at all.

We show that by slightly modifying *any* interactive argument, in a way that preserves its completeness and only slightly deteriorates its soundness, we get a protocol for which parallel repetition *does* reduce the error (at a weakly exponential rate). In this modified version, the verifier flips at the beginning of each round an $(1-\frac{1}{2m},\frac{1}{2m})$ biased coin (i.e., 1 is tossed with probability $1/2m$), where m is the round complexity of the (original) protocol. If the outcome is one, the verifier halts the interaction and accepts. Otherwise, it sends the same message that the original verifier would. At the end of the protocol (if reached), the verifier accepts if and only if the original verifier would.

1 Introduction

In an interactive proof, a prover P is trying to convince the verifier V in the validity of a statement.[1] The basic properties such protocols should have are *completeness* and *soundness*. The completeness means that P convinces V to accept *valid* statements, where the

*A preliminary version appeared as [8].

†School of Computer Science, Tel Aviv University. Research supported by ISF grant 1076/11, and the Israeli Centers of Research Excellence (I-CORE) program, Center No. 4/11. Part of this work was done while at Microsoft Research, New England Campus. E-mail: iftachh@cs.tau.ac.il.

[1]Typically, P has some advantage over V, such as additional computational resources or some extra information, e.g., an NP witness that validates the claim.

soundness means that no cheating prover (of a certain class) can convince V to accept *invalid* statements. More generally, (P,V) has completeness β, if V accepts a valid statement x in $(\mathsf{P},\mathsf{V})(x)$, with probability at least β. Where V has soundness $1-\varepsilon$, with respect to a given class of algorithms, if no malicious P^* from this class can convince V to accept an invalid statement with probability greater than ε. The bound ε is typically called the *soundness error* of the protocol.

The basic distinction one may make regarding the above soundness definition, is whether it holds unconditionally (i.e., even an all-powerful prover cannot break the soundness), or only holds against computationally bounded provers. Protocols with unconditional soundness are called *interactive proofs*, whereas protocols with the weaker type of soundness are called *interactive arguments* (also known as, *computationally sound proofs*). This work focuses on computationally bounded provers, specifically, on polynomial-time ones.

A common paradigm for constructing protocols with low soundness error, is to start by constructing a protocol with noticeably smaller than one soundness error, and then manipulate the protocol to decrease its soundness error. The most natural such manipulation is repetition: repeat the protocol many times (with independent randomness), where the verifier accepts only if the verifiers (of the original protocol) accept in *all* executions. Such repetition can be done in two different ways, sequentially (known as *sequential repetition*), where the $(j+1)$ execution of the protocol is only started after the j'th execution is finished, or in parallel (known as *parallel repetition*), where all the executions are done simultaneously.

Sequential repetition is known to reduce the soundness error at an exponential rate in most computational models (cf., [5]), but has the undesired effect of increasing the round complexity of the protocol. Parallel repetition, on the other hand, does preserve the round complexity, and for the case of interactive proofs also reduces the soundness error at an exponential rate [7]. But, as shown by Bellare, Impagliazzo, and Naor [2], *might not* reduce the soundness error in interactive arguments.

Let us be more precise about the latter statement. Parallel repetition does reduce the soundness error in the case of three-message arguments [2], and in the case of public-coin verifiers [11] (see Section 1.3 for more details). On the negative side, [2] presented for any $k \in \mathbb{N}$, a 4-message protocol with soundness error $\frac{1}{2}$, whose k-parallel repetition soundness remains $\frac{1}{2}$. More recently, Pietrzak and Wikström [15] gave an example of a *single* 8-message protocol for which the above phenomena holds for all polynomial k simultaneously. Both results hold under common cryptographic assumptions.

1.1 Our Result

We present a simple method for transforming any m-round interactive argument whose soundness error is bounded away from one, into an m-round interactive argument with negligible soundness error. Given an m-round interactive algorithm V, define its *random-terminating* variant, denoted $\widetilde{\mathsf{V}}$, as follows: $\widetilde{\mathsf{V}}$ acts exactly as V would have, on the same input and seeing the same transcript, with a single twist: at the end of each round, $\widetilde{\mathsf{V}}$ tosses an $(1-\frac{1}{2m},\frac{1}{2m})$ biased coin (i.e., 1 is tossed with probability $1/2m$). If the outcome is one,

then $\widetilde{\mathsf{V}}$ accepts (i.e., outputs 1) and halts. Otherwise, it sends the same message that V would. At the end of the protocol (if reached), $\widetilde{\mathsf{V}}$ accepts if and only if V would.

Let (P,V) be an m-round interactive argument, and let $(\mathsf{P},\widetilde{\mathsf{V}})$ be its random-terminating variant (i.e., $\widetilde{\mathsf{V}}$ is the random-terminating variant of V). It is easy to verify that the completeness of $(\mathsf{P},\widetilde{\mathsf{V}})$ is at least as high as that of (P,V), where the soundness of $(\mathsf{P},\widetilde{\mathsf{V}})$, is at least $(1-\frac{1}{2m})^m \cdot \alpha \geq \alpha/2$, for α being the soundness of (P,V). Our main contribution is stated in the following theorem.

Theorem 1.1 (informal)**.** *Parallel repetition of the random-terminating variant of* any *interactive argument, reduces the soundness error at a weakly exponential rate.*[2]

The above theorem is applicable for any interactive protocol that can be cast as an interactive argument. For instance, our result yields a round-preserving binding amplification for computationally binding commitment schemes.[3] In addition, it extends to the "threshold case", where the prover in the k-fold repetition is only required to make $t < k$ of the verifiers accept. Finally, the choice $1/2m$ as the halting probability of $\widetilde{\mathsf{V}}$ is rather arbitrary; the proof of Theorem 1.1, given in Section 3, can be easily altered for other choices of interest.

1.2 Our Technique

Let $(\mathsf{P},\widetilde{\mathsf{V}})$ be the random-terminating variant an m-round interactive argument (P,V), and let $(\mathsf{P}^{(k)},\widetilde{\mathsf{V}}^{(k)})$ be its k'th parallel repetition. We show that any efficient strategy B that breaks the soundness of $(\mathsf{P}^{(k)},\widetilde{\mathsf{V}}^{(k)})$ with noticeable probability ε_k (e.g., larger than 0.001), implies an *efficient* algorithm A that breaks the soundness of $(\mathsf{P},\widetilde{\mathsf{V}})$ with high probability (e.g., 0.999). As a warm up, we start by presenting such strategy assuming that $(\mathsf{P}^{(k)},\widetilde{\mathsf{V}}^{(k)})$ is public coin (but not necessarily random terminating), and then explain how to adapt it to the (private-coin) random-terminating case.

Public-coin protocols: The following description loosely follows the approach presented in [10]. In order to interact with $\widetilde{\mathsf{V}}$, algorithm A emulates a random execution of $(\mathsf{B},\widetilde{\mathsf{V}}^{(k)})$, where the "real" $\widetilde{\mathsf{V}}$ plays the role of the i'th verifier in $\widetilde{\mathsf{V}}^{(k)}$, for a uniformly chosen $i \in [k]$, and A emulates the other $(k-1)$ verifiers and B by itself. Upon receiving the j'th message r from $\widetilde{\mathsf{V}}$, algorithm A starts by uniformly choosing the emulated verifiers's messages in the

[2] We are using a rather relaxed interpretation of weakly exponential rate, meaning that the soundness error is bounded by $\max\{\text{neg}, \exp(-k \cdot \text{poly}(\frac{1-\varepsilon}{m})\}$, where m is the round complexity of the protocol and ε is the soundness error of (a single instance of) the random-terminating protocol. See Theorem 3.3 for the exact statement.

[3] Given a weakly binding commitment (S,R), consider the protocol (P,V) where P and V play the role of S and R in a random commit stage of (S,R) respectively. Following the commit stage, P sends two strings to V, and V accepts, iff the strings are valid decommitments to two *different* values. The weakly binding property of (S,R), yields that the soundness error of (P,V) is noticeably far from one. Thus, Theorem 1.1 yields that the parallel repetition of the random-terminating variant of (P,V) has negligible soundness error. Namely, the parallel repetition of the random-terminating variant of (S,R) is (strongly) binding.

j'th round. Let $\overline{r} = (r_1, \ldots, r_k)$ be the messages of the real and emulated verifiers induced by this sampling (i.e., $r_i = r$, and the other r's are the emulated verifiers' messages chosen by A). Next, A estimates $\alpha_{\overline{r}}$ — the probability that B makes $\widetilde{\mathsf{V}}^{(k)}$ accept, conditioned on $\overline{r}$ and on the emulated interaction of the previous rounds. To do so, A samples many random continuations of the emulated execution $(\mathsf{B}, \widetilde{\mathsf{V}}^{(k)})$,[4] and measures the fraction of accepting ones (i.e., where all verifiers accept). If the estimated value of $\alpha_{\overline{r}}$ is not "significantly" smaller than ε_k, then A updates the state of the emulated verifiers according to $\overline{r}$, and sends $\overline{a}_i$ back to $\widetilde{\mathsf{V}}$, where $\overline{a}$ is the j'th answer of B to $\widetilde{\mathsf{V}}^{(k)}$ in the emulated execution. Otherwise ($\alpha_{\overline{r}}$ is too small), A repeats the above process, to a maximum of w/ε_k attempts (for some large enough w).[5] Note that whenever A does not abort in any of the m rounds, it is guaranteed to make (the real verifier) $\widetilde{\mathsf{V}}$ accept.

To prove that A breaks the soundness of $(\mathsf{A}, \widetilde{\mathsf{V}})$ with high probability, it suffices to show that the conditional success probability of the emulated B after getting the $j+1$ message from the real verifier, is not much smaller than $\alpha_{\overline{r}}$ (as estimated at the j'th round). While in the *worst case* these two value might be far apart, one can use a result by Raz [16] to show that they are close for a *uniformly chosen* i.

Random-termination protocols. When one tries to adopt the above strategy to non public-coin protocols, he should first decide what the values of $\overline{r}$ and $\alpha_{\overline{r}}$ defined stand for in this case. The first approach that comes to mind, is viewing $\overline{r}$ as the verifiers' *messages* (in the j'th round), and let $\alpha_{\overline{r}}$ be the probability that B makes $\widetilde{\mathsf{V}}^{(k)}$ accept conditioned on $\overline{r}$, and on the transcript produced in the previous rounds. The very same argument we used above, yields that A makes $\widetilde{\mathsf{V}}$ accepts with high probability. The problem is, however, that the resulting strategy is not necessarily efficient in the non public-coin case; in particular, estimating $\alpha_{\overline{r}}$ requires the ability of finding a random preimage of the function that maps the coins of $\widetilde{\mathsf{V}}^{(k)}$ to protocol transcripts, a task that seems infeasible even for random terminating $\widetilde{\mathsf{V}}$.

To circumvent the above obstacle, we adopt the public-coin strategy in a different way; we view $\overline{r}$ as the verifiers' *random coins* in the j'th round, and let $\alpha_{\overline{r}}$ be the probability that B makes $\widetilde{\mathsf{V}}^{(k)}$ accept conditioned on $\overline{r}$, and on the *coins* flipped in the previous rounds. As in the case of former approach, one can show that the resulting adversary makes $\widetilde{\mathsf{V}}$ accept with high probability.

At a first look, however, it seems that there is no hope to implement the above strategy; even an unbounded algorithm cannot evaluate $\alpha_{\overline{r}}$ in the general case.[6] Interestingly, we show

[4] In general, choosing random continuation means sampling a random execution of the protocol that is consistent with the current state (e.g., messages sent, or coins flipped). In the case of public-coin protocols, however, it simply means choosing at uniform the messages of all verifiers in the last $m-j$ rounds (for the i'th verifier, whose j'th message was already chosen by the real execution, only choose the messages for the last $m-j-1$ rounds).

[5] [11] use a different (and somewhat less intuitive) strategy for evaluating the quality of $\overline{r}$, which significantly simplifies the above analysis. The sampling method of the cheating prover for random-terminating verifiers, described in Section 3, is a variant of their approach.

[6] The random coins that the real verifier chooses in the j'th round, might only affect the transcript on a

that the following variant of the above strategy can be implemented, and in an *efficient* way, for any random-terminating verifier. Let $\widetilde{\mathsf{V}}$ be a random-terminating verifier, and assume without loss of generality that it chooses *all* but its decision bits (the bits uses for deciding whether or not to terminate the executions) before the interaction begins. In order to approximate the value of $\alpha_{\overline{r}}$, the cheating prover A samples the future random coins of all verifiers, *conditioned that the real verifier's decision bit in the end of the j'th round is one* (i.e., the i'th verifier decides to halt in the end of the j'th round). Sampling in this case is very easy; the real verifier sends no further messages, and the future random coins for the emulated verifiers are uniformly distributed over all possible strings that cause $\widetilde{\mathsf{V}}^{(k)}$ to accept.

The obvious problem with the above approach is that adding this additional conditioning might effect B's success probability (and hence, causing A to err in its estimation of $\alpha_{\overline{r}}$). Fortunately, it turns out that the latter undesired effect does not happen for most choices of i. For a fixed $i \in [k]$, let *real distribution* be that of the verifiers' (real and emulated) random coins induced by a random execution of $(\mathsf{A}, \widetilde{\mathsf{V}})$. We compare this distribution to its ideal version (hereafter, the *ideal distribution*), where the value of $\alpha_{\overline{r}}$ is estimated at each round without the additional conditioning (say, by giving A access to the random coins of the real verifier). Our main technical contribution is showing that the above distributions are statistically close for most values of $i \in [k]$. Since in the ideal case A makes $\widetilde{\mathsf{V}}$ accepts with high probability (as in the public-coin case), the above observation yields that the same holds also in a real interaction of A with $\widetilde{\mathsf{V}}$.

Bounding the distance between the ideal and real distributions. Let us briefly explain why these distributions should be close to each other.[7] We say that $i \in [k]$ *affects* $\overline{r}$ in the j'th round, if adding the condition that the i'th verifier halts at the end of the j'th round, significantly changes the value of $\alpha_{\overline{r}}$, i.e., of B's success probability, conditioned that the verifiers random coins are set to $\overline{r}$.

Fix a round j and a value for $\overline{r}$ sampled in the j'th round, let ℓ be the number of indices affecting $\overline{r}$, and assume for simplicity that there exists a set $\mathcal{S} \subseteq [k]$ of size at least $\ell/2$, such that the following holds for every $i \in \mathcal{S}$: conditioning that the i'th verifier halts at the end of the j'th round, significantly biases the value of $\alpha_{\overline{r}}$ towards *zero*. Assuming that $k \geq \ell m$, yields that with probability $1 - O(2^{-\ell})$ at least one of the verifiers indexed by $\mathcal{S}$ halts in *any* random continuation of the protocol, yielding that $\alpha_{\overline{r}} \in O(2^{-\ell})$.

Taking large enough ℓ (so that $2^{-\ell} << \varepsilon_k$), the above observation yields that for most values of $i \in [k]$, most values of $\overline{r}$ are selected with similar probability in the real and in the ideal case. Namely, that the real and ideal distributions are close.

later round. Therefore, the transcript of the protocol in the j'th round might not contain enough information for estimating $\alpha_{\overline{r}}$.

[7] We remark that the question in hand is similar in spirit to the issue mentioned in the last paragraph of the above discussion regarding public-coin protocols. Indeed, we formally answer the current question via (a more sophisticated use of) the very same lemma by Raz [16].

1.3 Related Work

Babai and Moran [1] showed that parallel repetition reduces the soundness error of Arthur-Merlin protocols, where Goldreich [7, Appendix C.1] showed the same for interactive proofs. Parallel repetition is also known to reduce the error in the important case of two-prover interactive proofs [16]. In all above examples, the soundness error reduces at exponential rate.

Moving to bounded provers, Bellare et al. [2] showed that the parallel repetition of three-message interactive arguments reduces the soundness error at a weakly exponential rate. Canetti et al. [3] gave a tighter result for two-message arguments, where Impagliazzo et al. [13] generalized [3] to the threshold case. For public-coin arguments, Pass and Venkitasubramaniam [14] gave a parallel repetition theorem for constant-round protocols. This result was recently extended to a polynomial number of rounds by Hästad et al. [11], and further improved by Chung and Liu [4] (achieving, exponential error reduction). The results of [11, 4] also extends to the threshold case. More importantly, [10, 4] ([10] is the preliminary version of [11]) extend to the so called "extendable and simulatable" verifiers — it is feasible to sample a random continuation of the verifier's actions, given any partial transcript of the protocol. Finally, [11] presents an extension to "δ-simulatable" verifiers, a class of verifiers that contains random-terminating verifiers as a special case.

The idea of randomly terminating the verifier, is inspired by the work of Haitner, Reingold, Vadhan, and Wee [9], showing a round-preserving binding amplification of a specific (weakly) computational binding commitment. The phenomena that modifying a verifier to send *less* information might help its parallel repetition to reduce the soundness error, is a reminisce of the work of Feige and Kilian [6] (given in the context of two-prover protocols).

1.4 Paper Organization

Notations and formal definitions appear in Section 2, where our main result is formally stated and proved in Section 3.

2 Preliminaries

We use calligraphic letters to denote sets, uppercase for random variables, and lowercase for values. For $n \in \mathbb{N}$, let $[n] = \{1, \cdots, n\}$. We let poly be the set of all polynomials and let PPTM stand for probabilistic algorithm (i.e., Turing machines) that runs in *strict* polynomial time.

Given an interactive protocol (A, B), we let (A, B) also denote a random execution of (A, B). In case the parities get private inputs i_{A} and i_{B} respectively, and a common input i_c, we denote a random execution of (A, B) with these inputs by $(\mathsf{A}(i_{\mathsf{A}}), \mathsf{B}(i_{\mathsf{B}}))(i_c)$. An interactive algorithm A has m rounds, if m bounds A's number of rounds in (A, B) for any algorithm B; in case A accepts inputs, m can be a function of its input length. Let $\mathsf{A}^{(k)}$, for $k \in \mathbb{N}$, be the interactive algorithm consists of k *independent* copies of A; a message sent to $\mathsf{A}^{(k)}$ is accepted

to be of the form $m_1, \ldots, m_k$ (otherwise, $\mathsf{A}^{(k)}$ aborts), one message for each copy of A. In each round, every copy of A tosses its random coins and process its message *independently*, then the k messages are bundled into a single message (separated by commas) and sent to the other party.

The statistical distance of two probability distributions P and Q over $\mathcal{U}$, denoted $\mathrm{SD}(P,Q)$, is defined as $\frac{1}{2} \cdot \sum_{x \in \mathcal{U}} |P(x) - Q(x)|$. Given a random variable X, let $x \leftarrow X$ indicate that x is selected according to X. Similarly, given a finite set $\mathcal{S}$, let $s \leftarrow \mathcal{S}$ denote that s is selected according to the uniform distribution on $\mathcal{S}$. Given a random variable X and an event W (over the same probability space), let $(X \mid W)$ denote the random variable induced by drawing at random from X, conditioned on W (if W is an empty event, then $(X \mid W)$ has all its mass on $\perp$). Given two random variables X and Y, let $(X \mid Y)$ be the distribution $(X \mid Y = y)_{y \leftarrow Y}$.

The following proposition plays an important role in our proofs.

Proposition 2.1. *Let Y and $X_1 \ldots, X_k$ be independent random variables over some probability space let W be a non-empty event in the same space and let $\alpha = \sqrt{-\log(\mathsf{Pr}[W])/k}$, then*

1. $\mathsf{Pr}_{i \leftarrow [k], x \leftarrow X_i}\left[\mathsf{Pr}[W \mid X_i = x] \notin (1 \pm \varepsilon) \cdot \mathsf{Pr}[W]\right] \leq \frac{2}{\varepsilon} \cdot \alpha$ *for any* $\varepsilon > 0$,
2. $\mathsf{E}_{i \leftarrow [k]}\left[\mathrm{SD}\left((i, (\overline{X} \mid W)), (i, (\overline{X} \mid W, X_i))\right)\right] \leq \alpha$ *where* $\overline{X} = (X_1 \ldots, X_k)$, *and*
3. $\mathsf{E}_{i \leftarrow [k]}\left[\mathrm{SD}\left((i, (Y, X_i \mid W)), (i, (Y \mid W), X_i)\right)\right] \leq \alpha$.

Proof. We use the following fact due to Holenstein [12] (simplifying lemma of Raz [16]).

Lemma 2.2. *([12, Corollary 4.3], simplified version) Let Y, $X_1 \ldots, X_k$, W and α be as in Proposition 2.1, then $\mathsf{E}_{i \leftarrow [k]}\left[\mathrm{SD}\left((Y, X_i \mid W), ((Y \mid W), X_i)\right)\right] \leq \alpha$.*

For $i \in [k]$, let $\mathcal{S}_i^- = \{x \in \mathrm{Supp}(X_i) \colon \mathsf{Pr}[W \mid X_i = x] < (1 - \varepsilon) \cdot \mathsf{Pr}[W]\}$ and let $\mathcal{S}_i^+ = \{x \in \mathrm{Supp}(X_i) \colon \mathsf{Pr}[W \mid X_i = x] > (1 + \varepsilon) \cdot \mathsf{Pr}[W]\}$. Since

$$\begin{aligned} \mathrm{SD}((X_i \mid W), X_i) &\geq \frac{1}{2} \cdot \left|\mathsf{Pr}_{X_i|W}[\mathcal{S}_i^-] - \mathsf{Pr}_{X_i}[\mathcal{S}_i^-]\right| + \frac{1}{2} \cdot \left|\mathsf{Pr}_{X_i|W}[\mathcal{S}_i^+] - \mathsf{Pr}_{X_i}[\mathcal{S}_i^+]\right| \\ &\geq \frac{\varepsilon}{2} \cdot \mathsf{Pr}_{X_i}[\mathcal{S}_i^- \cup \mathcal{S}_i^+], \end{aligned}$$

it follows that

$$\mathsf{Pr}_{i \leftarrow [k], x \leftarrow X_i}\left[\mathsf{Pr}[W \mid X_i = x] \notin (1 \pm \varepsilon) \cdot \mathsf{Pr}[W]\right] \leq \frac{2}{\varepsilon} \cdot \mathsf{E}_{i \leftarrow [k]}\left[\mathrm{SD}((X_i \mid W), X_i)\right], \qquad (1)$$

and the proof of the first item follows by Lemma 2.2.

To see that the second item also follows by Lemma 2.2, notice that sampling from $\left(i, (\overline{X} \mid W)\right)$ and $\left(i, (\overline{X} \mid W, X_i)\right)$, can be done by applying the same random function to $(Y, X_i \mid W)$ and $((Y \mid W), X_i)$ respectively, i.e., $f_i(x, y)$ returns a random sample from $(i, (\overline{X} \mid W, X_i = x))$ (ignoring y). The third item fellows for similar reasons, taking $f_i(y, x) := (i, y, x)$. □

2.1 Random-terminating Algorithms

Definition 2.3. *[random-terminating variant] Let* A *be a deterministic, no-input interactive algorithm, and let* $\delta \in [0,1]$*. We define the* δ-random-terminating variant of A*, denoted* $\widetilde{\mathsf{A}}$*, as follows: algorithm* $\widetilde{\mathsf{A}}$ *acts exactly as* A *does, but adds the following step at the end of each communication round: it tosses an* $(1-\delta, \delta)$ *biased coin (i.e.,* 1 *is tossed with probability* δ*), if the outcome is one, then* $\widetilde{\mathsf{A}}$ *accepts (i.e., outputs '1') and halts. Otherwise, it continues as* A *(seeing the same transcript) would.*

The above naturally extends to randomized algorithms that do accept inputs, where in this case we allow δ *to be a function of the input length.*

2.2 Smooth Sampling

Let $X^m = (X_1, \ldots, X_m)$ be a random variable over $\mathcal{U}^m$, and let $\mathcal{S} \subseteq \mathcal{U}^m$ be with $\mathsf{Pr}[X^m \in \mathcal{S}] = \varepsilon$. For $j \in [m]$ and $\overline{x} \in \mathcal{U}^j$, let $v(\overline{x}) = \mathsf{Pr}_{X^m | X^j = \overline{x}}[X^m \in \mathcal{S}]$, where $X^j = (X_1, \ldots, X_j)$. Consider the task of choosing $(x_1, \ldots, x_m) \in \mathcal{S}$ in rounds, where the value of x_j should be chosen in the j'th round.

The straightforward strategy for winning such a game is to try and maximize the value of $v(x_1, \ldots, x_j)$ in each round. Following [11] we use the following strategy, whose proof turned to be useful for the question studied in this paper (see Section 3).

Algorithm 2.4 (Sam)**.**

For $j = 1$ *to* m *do:*

1. *Do until a break occur:*
   (a) *Sample* $(x'_1, \ldots, x'_m) \leftarrow (X^m \mid X^{j-1} = (x_1, \ldots, x_{j-1}))$.
   (b) *Break the loop if* $(x'_1, \ldots, x'_m) \in \mathcal{S}$.
2. *Set* $x_j = x'_j$.

Output $(x_1, \ldots, x_m)$.

It is clear that Sam outputs $(x_1, \ldots, x_m) \in \mathcal{S}$ with probability one. We make the following observations (implicit in [11]):

Proposition 2.5.

1. $\mathsf{E}_{(x_1,\ldots,x_m) \leftarrow \mathrm{Sam}}[\frac{1}{v(x_1,\ldots,x_j)}] = 1/\varepsilon$ *for every* $j \in [m]$*, and*
2. $\mathsf{Pr}[\mathrm{Sam}$ *loops more than* $\frac{t}{\varepsilon}$ *times in the* j*'th round*$] \leq \frac{1}{t}$*, for every* $j \in [m]$ *and* $t > 0$.

Proof. Let $(Y_1, \ldots, Y_m)$ be the output of a random execution of Sam, and for $j \in [m]$, let $Y^j = (Y_1, \ldots, Y_j)$. Note that $\frac{1}{v(x_1,\ldots,x_j)}$ is the expected number of loops Sam does in the $j+1$

round, conditioned on $Y^j = (x_1, \ldots, x_j)$. Hence fact (2) follows by (1) and a Markov bound. By induction, the following holds for every $j \in [m]$ and $\overline{x} = (x_1, \ldots, x_j) \in \mathcal{U}^j$:

$$\begin{aligned}
\mathsf{Pr}_{Y^j}[\overline{x}] &= \mathsf{Pr}_{Y^{j-1}}[\overline{x}_{1\ldots,j-1}] \cdot \mathsf{Pr}_{Y_j|Y^{j-1}=\,\overline{x}_{1\ldots,j-1}}[x_j] \qquad (2)\\
&= \mathsf{Pr}_{X^{j-1}}[\overline{x}_{1\ldots,j-1}] \cdot \frac{v(\overline{x}_{1\ldots,j-1})}{\varepsilon} \cdot \mathsf{Pr}_{Y_j|Y^{j-1}=\,\overline{x}_{1\ldots,j-1}}[x_j]\\
&= \mathsf{Pr}_{X^{j-1}}[\overline{x}_{1\ldots,j-1}] \cdot \frac{v(\overline{x}_{1\ldots,j-1})}{\varepsilon} \cdot \mathsf{Pr}_{X_j|X^{j-1}=\,\overline{x}_{1\ldots,j-1}}[x_j] \cdot \frac{v(\overline{x})}{v(\overline{x}_{1\ldots,j-1})}\\
&= \mathsf{Pr}_{X^j}[\overline{x}] \cdot \frac{v(\overline{x})}{\varepsilon},
\end{aligned}$$

where the second equality holds by the induction hypothesis, and the third one since

$$\begin{aligned}
\mathsf{Pr}_{Y_j|Y^{j-1}=\,\overline{x}_{1\ldots,j-1}}[x_j] &= \sum_{\ell=1}^{\infty}(1 - v(\overline{x}_{1\ldots,j-1}))^{\ell-1} \cdot \mathsf{Pr}_{X^m|X^{j-1}=\,\overline{x}_{1\ldots,j-1}}[X^m \in \mathcal{S} \wedge X_j = x_j] \qquad (3)\\
&= \frac{1}{v(\overline{x}_{1\ldots,j-1})} \cdot \mathsf{Pr}_{X^m|X^{j-1}=\,\overline{x}_{1\ldots,j-1}}[X^m \in \mathcal{S} \wedge X_j = x_j]\\
&= \frac{1}{v(\overline{x}_{1\ldots,j-1})} \cdot \mathsf{Pr}_{X_j|X^{j-1}=\,\overline{x}_{1\ldots,j-1}}[x_j] \cdot \mathsf{Pr}_{X^m|X^j=\,\overline{x}}[X^m \in \mathcal{S}]\\
&= \frac{1}{v(\overline{x}_{1\ldots,j-1})} \cdot \mathsf{Pr}_{X_j|X^{j-1}=\,\overline{x}_{1\ldots,j-1}}[x_j] \cdot v(\overline{x}).
\end{aligned}$$

It follows that

$$\begin{aligned}
\mathsf{E}_{Y^j}\left[\frac{1}{v(Y^j)}\right] &= \sum_{\overline{x}\in\mathcal{U}^j} \mathsf{Pr}[Y^j = \overline{x}] \cdot \frac{1}{v(\overline{x})}\\
&= \sum_{\overline{x}\in\mathcal{U}^j} \frac{v(\overline{x})}{\varepsilon} \cdot \mathsf{Pr}[X^j = \overline{x}] \cdot \frac{1}{v(\overline{x})}\\
&= \frac{1}{\varepsilon} \cdot \sum_{\overline{x}\in\mathcal{U}^j} \mathsf{Pr}[X^j = \overline{x}] = \frac{1}{\varepsilon}.
\end{aligned}$$

□

3 The Parallel Repetition Theorem

In this section we formalize and prove Theorem 1.1. We start by presenting our main technical contribution.

Lemma 3.1. *There exists an oracle-aided interactive algorithm* A *such that the following holds: let* V *be a no-input, m-round interactive algorithm, let* $t, k \in \mathbb{N}$, $\delta \in [0,1]$ *and let* $\widetilde{\mathsf{V}}$ *be the* δ*-random-terminating variant of* V *(see Definition 2.3). Assume that*

$$\mathsf{Pr}[\textit{at least } t \textit{ of the } \widetilde{\mathsf{V}}\textit{'s accept in } (\mathsf{B}, \widetilde{\mathsf{V}}^{(k)})] = \varepsilon_k > 0 \qquad (4)$$

for some interactive algorithm B*, then*

$$\mathsf{Pr}[(\mathsf{A}^{\mathsf{V},\mathsf{B}}(k,t,m,\varepsilon_k,\delta),\widetilde{\mathsf{V}})=1] > \frac{t}{k} - O\left(\frac{m}{\delta}\cdot\sqrt{\frac{-\log\varepsilon_k}{k}}\right).$$

The running time of A *(excluding the oracle calls) is polynomial in* k*,* m*,* $1/\varepsilon_k$*,* $1/\delta$ *and the maximal message length in* $(\mathsf{B},\widetilde{\mathsf{V}}^{(k)})$*.*

The above yields the following useful corollary.

Corollary 3.2. *Let* A*,* V*,* B*,* t*,* k*,* δ *and* $\widetilde{\mathsf{V}}$ *be as in Lemma 3.1. The exists universal constant* $c>0$ *such that the following holds: assume*

$$\mathsf{Pr}[\textit{at least } t \textit{ of the } \widetilde{\mathsf{V}}\textit{'s accept in } (\mathsf{B},\widetilde{\mathsf{V}}^{(k)})] = \varepsilon_k \geq \exp\left(-c\cdot k\cdot\left(\frac{\frac{t}{k}-\varepsilon}{m/\delta}\right)^2\right)$$

for some $\varepsilon>0$*, then*

$$\mathsf{Pr}[(\mathsf{A}^{\mathsf{V},\mathsf{B}}(k,t,m,\varepsilon_k,\delta),\widetilde{\mathsf{V}})=1] > \varepsilon, \tag{5}$$

and

$$\mathsf{Pr}[(\mathsf{A}^{\mathsf{V},\mathsf{B}}(k,t,m,\varepsilon_k,\delta),\mathsf{V})=1] > 2\varepsilon-1. \tag{6}$$

Proof. Equation (5) holds by Lemma 3.1, taking small enough c. Where since the probability that $\widetilde{\mathsf{V}}$ doe not chose to early terminate in a random execution is $(1-\frac{1}{2m})^m \geq \frac{1}{2}$, Equation (5) yields that $\mathsf{Pr}[(\mathsf{A}^{\mathsf{V},\mathsf{B}}(k,t,m,\varepsilon_k,\delta),\mathsf{V})=1] \geq (\varepsilon-\frac{1}{2})/\frac{1}{2} = 2\varepsilon-1$, proving Equation (6). □

We prove Lemma 3.1 below, but first use it (via Corollary 3.2) for proving Theorem 1.1 restated below.[8]

Theorem 3.3 (restatement of Theorem 1.1)**.** *Let* V *be an* $m(n)$*-round interactive* PPTM *and assume that*

$$\mathsf{Pr}[(\mathsf{A},\mathsf{V})(x)=1] \leq \varepsilon(n)$$

for any PPTM A*,* $x\in\{0,1\}^n$ *and large enough* n*. Let* $\widetilde{\mathsf{V}}$ *be the* $\frac{1}{2m(n)}$*-random-terminating variant of* V *and let* $\widetilde{\varepsilon}(n)=(1+\varepsilon(n))/2$*.*

Then for any polynomial-time computable functions k*,* t *and* α*, where* k *and* t *are integer functions, and* $\alpha(n)\geq\exp\left(-c\cdot k(n)\cdot\left(\frac{\frac{t(n)}{k(n)}-\widetilde{\varepsilon}(n)}{m(n)^2}\right)^2\right)$*, for* $c>0$ *being a universal constant, it holds that*

$$\mathsf{Pr}[\textit{at least } t(n) \textit{ of the } \widetilde{\mathsf{V}}\textit{'s accept in } (\mathsf{B},\widetilde{\mathsf{V}}^{(k)})(x)] < \max\{\mathrm{neg}(n),\alpha(n)\},$$

for any PPTM B*,* $x\in\{0,1\}^n$ *and large enough* n*.*

[8] We state Theorem 1.1 in the uniform settings (i.e., against uniform provers), but the very same proof of Theorem 3.3, given below, yields an equivalent result for non-uniform provers.

Namely, for large enough k, an efficient B cannot cheat significantly more verifiers than the trivial bound (i.e., $\widetilde{\varepsilon}(n)$ fraction of them).

Proof. Assume towards a contradiction that there exist a PPTM B and $p \in \text{poly}$ with

$$\mathsf{Pr}[\text{at least } t(n) \text{ of the } \widetilde{\mathsf{V}}\text{'s accept in } (\mathsf{B}, \widetilde{\mathsf{V}}^{(k)})(x_n)] \geq \alpha(n) \geq 1/p(n).$$

for infinitely many n's, where $x_n \in \{0,1\}^n$ is some function of n. Corollary 3.2 yields that for the right choice of c (specifically, four time larger than the constant in the statement of corollary), it holds that

$$\mathsf{Pr}[(\mathsf{A}^{\mathsf{V}(x_n),\mathsf{B}(x_n)}(k(n), t(n), \alpha(n), 1/2m(n)), \mathsf{V}(x_n)) = 1] > 2\widetilde{\varepsilon}(n) - 1 = \varepsilon(n) \tag{7}$$

for infinitely many n's, where A is the oracle-aided algorithm guaranteed by Corollary 3.2. Since all the above algorithms and functions are polynomial-time computable, there exists a PPTM A' with

$$\mathsf{Pr}[(\mathsf{A}', \mathsf{V})(x_n)) = 1] > \varepsilon(n)$$

for infinitely many n's, in contradiction to the assumed soundness of V. □

Proof of Lemma 3.1. Fix V, B, t, k, ε_k and δ such that *Equation* (4) holds, and set $\varepsilon = \varepsilon_k$. We assume without loss of generality that $k > 1$, that V flips all its random coins before the interaction begins, and that B is deterministic.[9] For notational convenience, we view the "random-terminating coins" flipped by V at the end of the j'th round, as if they were flipped at the beginning of the $(j+1)$'th round. This yields that a full interaction of $(\mathsf{B}, \widetilde{\mathsf{V}}^{(k)})$ has $m+1$ rounds, where all that happens in the $(m+1)$'th "round", is the verifiers taking their acceptance decision (no message sent).

A partial view of $\widetilde{\mathsf{V}}^{(k)}$ in $(\mathsf{B}, \widetilde{\mathsf{V}}^{(k)})$ is of the form $\overline{\text{view}} = (\overline{r^1}, \overline{r^2}, \ldots, \overline{r^\ell})$, where $\overline{r^1} \in \{0,1\}^{k \cdot \text{len}}$ are the coins of the embedded V's, and $\overline{r^j} \in \{0,1\}^k$, for $j \in \{2, \ldots, \ell\}$, are the random-terminating coins of the $\widetilde{\mathsf{V}}$'s in the j'th round (arbitrary defined for those $\widetilde{\mathsf{V}}$'s that aborted before the j'th round). In particular, $\overline{r^{m+1}}_i$ determines whether the i'th $\widetilde{\mathsf{V}}$'s simply accepts, or only if the embedded V does. We included no messages in $\overline{\text{view}}$, since their values is determined by the coins (recall B is deterministic). We associate the following functions with such a view: let $\text{round}(\overline{\text{view}}) = \ell$, and for $j \in [\ell]$, let $\overline{r^j}(\overline{\text{view}}) = \overline{r^j}$ and $\overline{r^{1,\ldots,j}}(\overline{\text{view}}) = (\overline{r^1}, \ldots, \overline{r^j})$. Let $\text{Accept}_i(\overline{\text{view}}) = 1$ iff the output of i'th verifier in $\overline{\text{view}}$ is '1' (i.e., the i'th $\widetilde{\mathsf{V}}$ accepts), and let $\text{Accept}(\overline{\text{view}}) = 1$ iff $\sum_{j \in [k]} \text{Accept}_i(\overline{\text{view}}) \geq t$.

The definition of A given below follows rather closely the intuition given in Section 1.2. The main difference is that in order to choose the emulated verifiers' random coins, it uses a variant of the "smooth sampler" described in Section 2.2, rather than the threshold approach described in the introduction. For notational convenience, we assume below that $\widetilde{\mathsf{V}}$ sends in

[9] The only non trivial "without loss of generality" is B being deterministic. For that one can append in the reduction a random string to $\widetilde{\mathsf{V}}$'s first message, and instruct B to use the string attached to (the first message of) the first verifier in $\widetilde{\mathsf{V}}^{(k)}$ as its random coins.

each round the new set of *coins* it flipped in this round, and not the resulting message. This is merely done for presentation clarity, and we explain below how to do things without it.

In the following let $\mu = \frac{1}{\delta} \cdot \sqrt{\frac{-\log \varepsilon}{k}}$ and $w = \left\lceil \frac{2}{\varepsilon \mu} \right\rceil$.

Algorithm 3.4 (A)**.**

1. *Set* $i \leftarrow [k]$ *and* $\overline{\text{view}} = \perp$.
2. *For* $j = 1$ *to* $m + 1$ *do:*
   (a) *Get the next random coins* r *from* $\widetilde{\mathsf{V}}$.
   (b) *Set* $\overline{\text{view}} = (\overline{\text{view}}, \text{GetNextCoins}(\overline{\text{view}}, i, r))$.
   (c) *Send* $\overline{a^j}_i$ *back to* $\widetilde{\mathsf{V}}$*, where* $\overline{a^j}$ *is the message that* B *sends to* $\widetilde{\mathsf{V}}^{(k)}$ *in the* j*'th round of* $\overline{\text{view}}$.

Algorithm 3.5 (GetNextCoins)**.**

Input: a (partial) view of $\widetilde{\mathsf{V}}^{(k)}$ *—* $\overline{\text{view}}$*, an index* $i \in [k]$ *and a string* $r \in \{0,1\}^*$.

Operation:

Set $j = \text{round}(\overline{\text{view}}) + 1$ *(set* $j = 1$*, if* $\overline{\text{view}} = \perp$*), and do the following for* w *times:*

1. *Choose* $\overline{\text{view}}'$ *as* $\widetilde{\mathsf{V}}^{(k)}$*'s view in a random execution of* $(\mathsf{B}, \widetilde{\mathsf{V}}^{(k)})$*, condition on*
   (a) $\overline{r^{1,\ldots,j-1}}(\overline{\text{view}}') = \overline{\text{view}}$,
   (b) $\overline{r^j}(\overline{\text{view}}')_i = r$*, and*
   (c) $\overline{r^{j+1}}(\overline{\text{view}}')_i = 1$ *(only taken if* $j < m + 1$*).*
2. *If* $\text{Accept}(\overline{\text{view}}')$*, return* $\overline{r^j}(\overline{\text{view}}')$.

Abort the execution.

Note that no message is sent by A at the $(m+1)$'th round, and this round is only included for notational convince. We also note that since GetNextCoins conditions that $\widetilde{\mathsf{V}}_i^{(k)}$ (i.e., the real verifier) halts at the end of the j'th round and since B only sees the verifiers' messages, it is easy to modify A to only use the *messages* sent by $\widetilde{\mathsf{V}}_i^{(k)}$ (and not its random coins), while maintaining the same success probability.[10] Finally, it is easy to verify that given oracle

[10] In this variant, GetNextCoins aims to find a good value for $\overline{\text{view}}'_{-i}$ — the emulated verifiers's view. Note that when conditioning on $r_{j+1}(\overline{\text{view}}'_i) = 1$ (where $\overline{\text{view}}'_i$ is the real verifier's view), the value of $\overline{\text{view}}'_{-i}$ along with $\widetilde{\mathsf{V}}_i^{(k)}$'s messages in the first j rounds, suffice to compute $\text{Accept}(\overline{\text{view}}' = (\overline{\text{view}}'_i, \overline{\text{view}}'_{-i}))$ and to compute B's first j messages in $\overline{\text{view}}'$.

access to B and V, and the values of k, t, ε and δ, the running time of A (and of its realistic variant) fulfills the requirement of the lemma.

In the following we analyze the success probability of A. We assume that A outputs the value of $(i, \overline{\text{view}})$ at the end of the interaction; outputs $\perp$, in case one of the calls to GetNextCoins aborts. Let $\widetilde{\text{GetNextCoins}}$ be the "unbounded variant" of GetNextCoins — the loop in $\widetilde{\text{GetNextCoins}}$ runs until a good value for $\overline{\text{view}}'$ is found (i.e., $\text{Accept}(\overline{\text{view}}') = 1$); $\widetilde{\text{GetNextCoins}}$ aborts, in case no such good value exists. We define that if $\widetilde{\text{GetNextCoins}}$ is called with the parameter i set to $\perp$, it does the sampling of Step 1 without conditions (b) and (c).

For $\ell \in \{0, \ldots, m+1\}$, the experiment Exp^ℓ is defined as follows:

Experiment 3.6 (Exp^ℓ)**.**

1. *Set* $\overline{\text{view}} = \perp$.
2. *For* $j = 1$ *to* ℓ *do:*
   Set $\overline{\text{view}} = (\overline{\text{view}}, \widetilde{\text{GetNextCoins}}(\overline{\text{view}}, \perp, \perp))$.
3. *Set* $i \leftarrow [k]$*; emulate a random execution of* $(\mathsf{A}, \widetilde{\mathsf{V}})$*, condition that* $(i, \overline{\text{view}})$ *is* A*'s view after the* ℓ*'th call to* GetNextCoins*, and output the same output that* A *does.*

Note that Exp^0 describes a random execution of $(\mathsf{A}, \widetilde{\mathsf{V}})$. It is also clear that Exp^{m+1} always outputs an accepting $\overline{\text{view}}$, which is independent of the index i. In the following we abuse notation and view Exp^ℓ also as the random variable the output of a random execution of Exp^ℓ induces, and prove the lemma via the following claim.

Claim 3.7. $\text{SD}(\text{Exp}^\ell, \text{Exp}^{\ell+1}) \in O(\mu)$*, for every* $\ell \in \{0, \ldots, m\}$.

Assuming Claim 3.7, it holds that

$$\begin{aligned}
\mathsf{Pr}[(\mathsf{A}^{\mathsf{V},\mathsf{B}}(k,t,\varepsilon,\delta), \widetilde{\mathsf{V}}) = 1] &= \mathsf{Pr}_{(i,\overline{\text{view}})\leftarrow\text{Exp}^0}[\text{Accept}_i(\overline{\text{view}})] \\
&\geq \mathsf{Pr}_{(i,\overline{\text{view}})\leftarrow\text{Exp}^{m+1}}[\text{Accept}_i(\overline{\text{view}})] - \text{SD}(\text{Exp}^0, \text{Exp}^{m+1}) \\
&\geq \frac{t}{k} - \text{SD}(\text{Exp}^0, \text{Exp}^{m+1}) \\
&\geq \frac{t}{k} - \sum_{\ell=0}^{m} \text{SD}(\text{Exp}^\ell, \text{Exp}^{\ell+1}) \\
&\geq \frac{t}{k} - O(m\mu),
\end{aligned}$$

concluding the proof of Lemma 3.1. The second inequality holds since in Exp^{m+1}, the value of i is independent of $\overline{\text{view}}$ and at least t sub-verifiers accept in $\overline{\text{view}}$, and the last one by Claim 3.7. □

3.1 Proving Claim 3.7

We prove the claim for $\ell \in \{1, \ldots, m-1\}$, where the simpler case $\ell = m$ follows by similar lines.

The only difference between Exp^{ℓ} and $\text{Exp}^{\ell+1}$ is in their $(\ell+1)$ "round". In this round $\text{Exp}^{\ell+1}$ calls $\text{Get}\widetilde{\text{NextCoins}}(\overline{\text{view}}, \perp, \perp)$, and Exp^{ℓ} (implicitly via the emulation of $(\mathsf{A}, \widetilde{\mathsf{V}})$) calls $\text{GetNextCoins}(\overline{\text{view}}, I, R^1)$, where I is uniformly distributed over $[k]$ and R^1 is distributed according to the coins flipped by $\widetilde{\mathsf{V}}$ at the $(\ell+1)$'th round. Let Prefix be the distribution induced by the value of $\overline{\text{view}}$ after the ℓ'th round of Exp^{ℓ} (or, equivalently, of $\text{Exp}^{\ell+1}$). For $\overline{\text{view}} \in \text{Supp}(\text{Prefix})$, define the distributions $\mathsf{D}^{0,\overline{\text{view}}}$ and $\mathsf{D}^{1,\overline{\text{view}}}$ as follows:

- $\mathsf{D}^{0,\overline{\text{view}}} := \left(I, \text{GetNextCoins}(\overline{\text{view}}, I, R^1)\right)$, and
- $\mathsf{D}^{1,\overline{\text{view}}} := \left(I, \text{Get}\widetilde{\text{NextCoins}}(\overline{\text{view}}, \perp, \perp)\right)$.

It is easy to verify that

$$\text{SD}(\text{Exp}^{\ell}, \text{Exp}^{\ell+1}) = \mathsf{E}_{\overline{\text{view}} \leftarrow \text{Prefix}}\left[\text{SD}(\mathsf{D}^{0,\overline{\text{view}}}, \mathsf{D}^{1,\overline{\text{view}}})\right] \tag{8}$$

The following claim is where the heart of the proof lies.

Claim 3.8. *For every* $\overline{\text{view}} \in \text{Supp}(\text{Prefix})$*, it holds that*

$$\text{SD}(\mathsf{D}^{0,\overline{\text{view}}}, \mathsf{D}^{1,\overline{\text{view}}}) \in O\left(\frac{1}{\delta} \cdot \sqrt{\frac{-\log(\alpha(\overline{\text{view}}))}{k}} + \left(1 - \frac{\alpha(\overline{\text{view}})}{2}\right)^{w}\right),$$

where $\alpha(\overline{\text{view}}) = \mathsf{Pr}[\text{Accept}(\overline{\text{View}})]$*, for* $\overline{\text{View}}$ *being* $\widetilde{\mathsf{V}}^{(k)}$*'s view in a random execution of* $(\mathsf{B}, \widetilde{\mathsf{V}}^{(k)})$ *conditioned on* $\overline{r^{1,\ldots,\ell}}(\overline{\text{View}}) = \overline{\text{view}}$*.*

Before proving Claim 3.8, we first use it to prove Claim 3.7.

Proof of Claim 3.7. Let $\overline{\text{View}}$ be $\widetilde{\mathsf{V}}^{(k)}$'s view in a random execution of $(\mathsf{B}, \widetilde{\mathsf{V}}^{(k)})$. Observe that choosing $\overline{\text{view}}$ by applying the first ℓ rounds of Exp^{ℓ}, is equivalent to the way 2.4 chooses the value of $(x_1, \ldots, x_\ell)$, with respect to $X^{m+1} = (\overline{r^1}(\overline{\text{View}}), \ldots, \overline{r^{m+1}}(\overline{\text{View}}))$ and $\mathcal{S} = \{\overline{\text{view}} \in \text{Supp}(\overline{\text{View}}) \colon \text{Accept}(\overline{\text{View}}) = 1\}$. Hence, Proposition 2.5 yields that

$$\mathsf{E}_{\overline{\text{view}} \leftarrow \text{Prefix}}\left[1/\alpha(\overline{\text{view}})\right] = 1/\varepsilon \tag{9}$$

Consider a variant of $\text{Exp}^{\ell+1}$ where in the $(\ell+1)$'th call to $\text{Get}\widetilde{\text{NextCoins}}$, a successful loop (i.e., $\text{Accept}(\overline{\text{view}}') = 1$) causes a return only with probability half, and let L denote the number of loops done in the $(\ell+1)$'th call of this variant. Since $\mathsf{E}[L] = \mathsf{E}_{\overline{\text{view}} \leftarrow \text{Prefix}}\left[1/2\alpha(\overline{\text{view}})\right] = 2/\varepsilon$, a Markov bound yields that $\mathsf{Pr}[L \geq w] \leq \mu$ (recall $w = \left\lceil \frac{2}{\varepsilon\mu} \right\rceil$). Noting that $(1 - \frac{\alpha(\overline{\text{view}})}{2})^w$

is the probability of $L \geq w$ conditioned on $\overline{\text{view}}$ being the ℓ-round view of the above variant, it follows that

$$\mathsf{E}_{\overline{\text{view}} \leftarrow \text{Prefix}}[(1 - \frac{\alpha(\overline{\text{view}})}{2})^w] \leq \mu \tag{10}$$

Finally, applying Jensen's inequality with respect to the function $\sqrt{\log(\cdot)/k}$ and the set $\{\frac{1}{\alpha(\overline{\text{view}})}\}_{\overline{\text{view}} \in \text{Prefix}}$, yields that

$$\mathsf{E}_{\overline{\text{view}} \leftarrow \text{Prefix}} \left[\sqrt{\log(1/\alpha(\overline{\text{view}}))/k} \right] \leq \sqrt{\log(\mathsf{E}_{\overline{\text{view}} \leftarrow \text{Prefix}}[1/\alpha(\overline{\text{view}})])/k} = \sqrt{\frac{-\log \varepsilon}{k}} \tag{11}$$

We conclude that

$$\text{SD}(\text{Exp}^\ell, \text{Exp}^{\ell+1}) = \mathsf{E}_{\overline{\text{view}} \leftarrow \text{Prefix}} \left[\mathsf{D}^{0,\overline{\text{view}}}, \mathsf{D}^{1,\overline{\text{view}}} \right] \in O\left(\frac{1}{\delta} \cdot \sqrt{\frac{-\log \varepsilon}{k}} + \mu \right) \in O(\mu).$$

□

Proof of Claim 3.8. Fix $\overline{\text{view}} \in \text{Supp}(\text{Prefix})$, let $\alpha = \alpha(\overline{\text{view}})$, $\mathsf{D}^0 = \mathsf{D}^{0,\overline{\text{view}}}$ and $\mathsf{D}^1 = \mathsf{D}^{1,\overline{\text{view}}}$. Let $\overline{\text{View}}$ be $\widetilde{\mathsf{V}}^{(k)}$'s view in a random execution of $(\mathsf{B}, \widetilde{\mathsf{V}}^{(k)})$, conditioned on $\overline{r^{1,\ldots,\ell}}(\overline{\text{View}}) = \overline{\text{view}}$. Let $\overline{R^1} = \overline{r^{\ell+1}}(\overline{\text{View}})$, $\overline{R^2} = \overline{r^{\ell+2}}(\overline{\text{View}})$ and let $W = 1$ iff $\text{Accept}(\overline{\text{View}})$ (i.e., $\mathsf{Pr}[W] = \alpha$). Note that $\{\overline{R^1}_1, \ldots, \overline{R^1}_k, \overline{R^2}_1, \ldots, \overline{R^2}_k\}$ are independent random variables, that each of the Boolean $\overline{R^2}_i$'s is one with probability δ, and that the $\overline{R^1}_i$'s are also identical random variables.[11] Recall that I is uniformly distributed over $[k]$, that R^1 is distributed according to the distribution of $\overline{R^1}_1$ (which is that of $\overline{R^1}_i$, for any $i \in [k]$). For $r^1 \in \text{Supp}(R^1)$ and $r^2 \in \{0,1\}$, let $\mathsf{T}(r^1, r^2)$ be the distribution of $\left(I, (\overline{R^1} \mid W, \overline{R^1}_I = r^1, \overline{R^2}_I = r^2)\right)$; we alow both r^1 and r^2 to take the value $\perp$, meaning that the relevant conditioning is omitted (e.g., $\mathsf{T}(\perp, 1) = (I, (\overline{R^1} \mid W, \overline{R^2}_I = 1))$).

It is easy to verify that D^1 is equivalent to $\mathsf{T}(\perp, \perp)$, where we also show that D^0 is close to $\mathsf{T}(R^1, 1)$, letting $\mathsf{T}(R^1, 1) := (\mathsf{T}(r^1, 1))_{r^1 \leftarrow R^1}$ (both facts are proved in Claim 3.11). Hence, all is left to do is proving that $\mathsf{T}(\perp, \perp)$ is close to $\mathsf{T}(R^1, 1)$. We do so by relating their distance to the following well studied question: given a tuple $\overline{X} = (X_1, \ldots, X_k)$ of independent random variables and an event W, what is the distance between $(I, (\overline{X} \mid W))$ and $(I, (\overline{X} \mid W, X_I))$?[12] Roughly (see intuition in the last part of Section 1.2), the two distributions are close, as long as $\mathsf{Pr}[W]$ is not "too small". Raz [16] proves a concrete bound for this distance (a result which we use here via Proposition 2.1).

We formally prove Claim 3.8 via the following steps: we prove (Claim 3.9) that $\mathsf{T}(\perp, \perp)$ is close to $\mathsf{T}(\perp, 1)$ and then prove that (Claim 3.10) that $\mathsf{T}(\perp, 1)$ is close to $\mathsf{T}(R^1, 1)$, yielding that $\mathsf{T}(\perp, \perp)$ is close to $\mathsf{T}(R^1, 1)$. We conclude the proof by Claim 3.11, showing that D^1 is the distribution of $\mathsf{T}(\perp, \perp)$, and D^0 is close to $\mathsf{T}(R^1, 1)$.

[11] The distribution of each of the $\overline{R^1}_i$'s is either uniform over $\{0,1\}^{\text{len}}$ (for $\ell = 0$), or the same as of the $\overline{R^2}_i$'s.

[12] Recall that $(\overline{X} \mid W)$ is the distribution induced by drawing a sample from $\overline{X}$, conditioned on W, and that $(\overline{X} \mid W, X_i)$ is the distribution induced by the above process when also conditioning on $X_i = x_i$, for $x_i \leftarrow X_i$.

Claim 3.9. $\mathrm{SD}\left(\mathsf{T}(\perp,\perp),\mathsf{T}(\perp,1)\right) \leq \frac{2}{\delta}\cdot\sqrt{\frac{-\log\alpha}{k}}$.

Proof. Applying Proposition 2.1(3) to $Y = \overline{R^1}$, $\overline{X} = \overline{R^2}$ and W, yields that

$$\mathrm{SD}\left(\left(I,(\overline{R^1},\overline{R^2}_I \mid W)\right),\left(I,(\overline{R^1}\mid W),\overline{R^2}_I)\right)\right) \leq \sqrt{\frac{-\log\alpha}{k}} \tag{12}$$

Since $\mathsf{Pr}\left[\overline{R^2}_I = 1\right] = \delta$, it follows that

$$\mathrm{SD}\left(\left(I,(\overline{R^1}\mid W,\overline{R^2}_I = 1)\right),\left(I,(\overline{R^1}\mid W)\right)\right) \leq \frac{2}{\delta}\cdot\sqrt{\frac{-\log\alpha}{k}},$$

concluding the claim's proof. □

We next prove that $\mathsf{T}(\perp,1)$ is close to $\mathsf{T}(R^1,1)$, and conclude that $\mathsf{T}(\perp,\perp)$ is close to $\mathsf{T}(R^1,1)$.

Claim 3.10. $\mathrm{SD}\left(\mathsf{T}(\perp,1),\mathsf{T}(R^1,1)\right) \leq \frac{2}{\delta}\cdot\sqrt{\frac{-\log\alpha}{k}}$.

Proof. Applying Proposition 2.1(2) to $\overline{X} = ((\overline{R^1}_1,\overline{R^2}_1),\ldots,(\overline{R^1}_k,\overline{R^2}_k))$ and W, yields that $\mathrm{SD}\left(\left(I,(\overline{X}\mid W)\right),\left(I,(\overline{X}\mid W,\overline{X}_I)\right)\right) \leq \sqrt{\frac{-\log\alpha}{k}}$. Since $\mathsf{Pr}[\overline{X}_I = (\cdot,1)] = \delta$, it follows that $\mathrm{SD}\left(\left(I,(\overline{X}\mid W,(\overline{X}_I)_2 = 1)\right),\left(I,(\overline{X}\mid W,(\overline{X}_I)_1,(\overline{X}_I)_2 = 1)\right)\right) \leq \frac{2}{\delta}\cdot\sqrt{\frac{-\log\alpha}{k}}$. We conclude that

$$\begin{aligned}\mathrm{SD}\left(\mathsf{T}(\perp,1),\mathsf{T}(R^1,1)\right) &= \mathrm{SD}\left(\left(I,(\overline{R^1}\mid W,\overline{R^2}_I = 1)\right),\left(I,(\overline{R^1}\mid W,\overline{R^1}_I,\overline{R^2}_I = 1)\right)\right)\\ &\leq \mathrm{SD}\left(\left(I,(\overline{X}\mid W,(\overline{X}_I)_2 = 1)\right),\left(I,(\overline{X}\mid W,(\overline{X}_I)_1,(\overline{X}_I)_2 = 1)\right)\right)\\ &\leq \frac{2}{\delta}\cdot\sqrt{\frac{-\log\alpha}{k}}.\end{aligned}$$

□

Combining Claims 3.9 and 3.10 yields that

$$\mathrm{SD}\left(\mathsf{T}(\perp,\perp),\mathsf{T}(R^1,1)\right) \leq \frac{4}{\delta}\cdot\sqrt{\frac{-\log\alpha}{k}} \tag{13}$$

and we conclude the proof of Claim 3.8 using the following claim.

Claim 3.11. D^1 *is equivalent to* $\mathsf{T}(\perp,\perp)$ *and* $\mathrm{SD}(\mathsf{D}^0,\mathsf{T}(R^1,1)) \leq \frac{4}{\delta}\cdot\sqrt{\frac{-\log\alpha}{k}} + (1-\frac{\alpha}{2})^w$.

Proof. For $\overline{r^1} \in \mathrm{Supp}(\overline{R^1})$ it holds that

$$\begin{aligned}\mathsf{D}^1(\cdot,\overline{r^1}) &= \mathsf{Pr}[\widetilde{\mathrm{GetNextCoins}}(\overline{\mathrm{view}},\perp,\perp)=\overline{r^1}] \\ &= \mathsf{Pr}[\overline{R^1}=\overline{r^1}\wedge W]\cdot\sum_{z=0}^{\infty}(1-\mathsf{Pr}[W])^{z-1} \\ &= \frac{\mathsf{Pr}[\overline{R^1}=\overline{r^1}\wedge W]}{\mathsf{Pr}[W]} = \mathsf{Pr}_{\mathsf{T}(\perp,\perp)}[\overline{r^1}],\end{aligned} \tag{14}$$

yielding that D^1 is the distribution of $\mathsf{T}(\perp,\perp)$.

A similar arguments yields that the distribution of $\mathsf{T}(R^1,1)$ is $\widetilde{\mathsf{D}}^0$, where $\widetilde{\mathsf{D}}^0$ be the variant of D^0 obtained by using $\widetilde{\mathrm{GetNextCoins}}$ instead of GetNextCoins, where a simple coupling argument yields that $\mathrm{SD}(\mathsf{D}^0,\widetilde{\mathsf{D}}^0)=\mathsf{Pr}_{\mathsf{D}^0}[(\cdot,\perp)]-\mathsf{Pr}_{\widetilde{\mathsf{D}}^0}[(\cdot,\perp)]$. Since $\mathsf{Pr}_{\widetilde{\mathsf{D}}^0}[(\cdot,\perp)]\leq\mathsf{Pr}_{\mathsf{D}^0}[(\cdot,\perp)]$, we conclude the proof by for showing that $\mathsf{Pr}_{\mathsf{D}^0}[(\cdot,\perp)]$ is small. We write

$$\mathsf{Pr}_{\mathsf{D}^0}[(\cdot,\perp)] = \mathsf{E}_{i\leftarrow[k],r^1\leftarrow R^1}\left[\beta(i,r^1)\cdot\sum_{z=w}^{\infty}(1-\beta(i,r^1))^z\right] = \mathsf{E}_{i\leftarrow[k],r^1\leftarrow R^1}\left[(1-\beta(i,r^1))^w\right] \tag{15}$$

letting $0\cdot\infty=1$, where $\beta(i,r^1):=\mathsf{Pr}[W\mid(\overline{R^1}_i,\overline{R^2}_i)=(r^1,1)]$. Applying Proposition 2.1(1) with respect to $X_1=(\overline{R^1}_1,\overline{R^2}_1),\ldots,X_k=(\overline{R^1}_k,\overline{R^2}_k)$, W and $\varepsilon=\frac{1}{2}$, yields that

$$\mathsf{Pr}_{i\leftarrow[k],r^1\leftarrow R^1}\left[\beta(i,r^1)<\alpha/2\right]\leq\frac{4}{\delta}\cdot\sqrt{\frac{-\log\alpha}{k}} \tag{16}$$

and we conclude that

$$\begin{aligned}\mathrm{SD}(\mathsf{D}^0,\mathsf{T}(R^1,1)) &= \mathrm{SD}(\mathsf{D}^0,\widetilde{\mathsf{D}}^0) \\ &\leq \mathsf{Pr}_{\mathsf{D}^0}[(\cdot,\perp)] \\ &\leq \frac{4}{\delta}\cdot\sqrt{\frac{-\log\alpha}{k}}+(1-\frac{\alpha}{2})^w.\end{aligned}$$

□

Combining Equation (13) and Claim 3.11, yields that $\mathrm{SD}(\mathsf{D}^0,\mathsf{D}^1)\in O\left(\frac{1}{\delta}\cdot\sqrt{\frac{-\log\alpha}{k}}+(1-\frac{\alpha}{2})^w\right)$, concluding the claim's proof. □

Acknowledgment

I am very thankful to Oded Goldreich, Thomas Holenstein, Tal Moran, Rafael Pass, Omer Reingold, Alex Samorodnitsky and Salil Vadhan for useful discussions. Alex deserves special thanks, for advising me from the very first step of this project. I also thank the anonymous referees for their very useful comments.

References

[1] L. Babai and S. Moran. Arthur-Merlin games: A randomized proof system and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

[2] M. Bellare, R. Impagliazzo, and M. Naor. Does parallel repetition lower the error in computationally sound protocols? In *the 37th Annual Symposium on Foundations of Computer Science (FOCS)*, 1997.

[3] R. Canetti, S. Halevi, and M. Steiner. Hardness amplification of weakly verifiable puzzles. In *Theory of Cryptography, Second Theory of Cryptography Conference (TCC)*, 2005.

[4] K.-M. Chung and F.-H. Liu. Parallel repetition theorems for interactive arguments. In *Theory of Cryptography, Seventh Theory of Cryptography Conference (TCC)*, 2010.

[5] I. B. Damgärd and B. Pfitzmann. Sequential iteration arguments and an efficient zero-knowledge argument for NP. In *ICALP: Annual International Colloquium on Automata, Languages and Programming*, 1998.

[6] U. Feige and J. Kilian. Two prover protocols: low error at affordable rates. In *the 26th Annual ACM Symposium on Theory of Computing (STOC)*, 1994.

[7] O. Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness.* Springer, 1999.

[8] I. Haitner. A parallel repetition theorem for any interactive argument. In *the 50th Annual Symposium on Foundations of Computer Science (FOCS)*, 2009.

[9] I. Haitner, O. Reingold, S. Vadhan, and H. Wee. Inaccessible entropy. In *the 41st Annual ACM Symposium on Theory of Computing (STOC)*, 2009.

[10] J. Hästad, R. Pass, K. Pietrzak, and D. Wikström. An efficient parallel repetition theorem. Unpublished manuscript, 2008.

[11] J. Hästad, R. Pass, K. Pietrzak, and D. Wikström. An efficient parallel repetition theorem. In *Theory of Cryptography, Seventh Theory of Cryptography Conference (TCC)*, 2010.

[12] T. Holenstein. Parallel repetition: simplifications and the no-signaling case. *Theory of Computing*, 5:141–172, 2009. Preliminary version in *STOC'07*.

[13] R. Impagliazzo, R. Jaiswal, and R. Kabanets. Approximately list-decoding direct product codes and uniform hardness amplification. In *the 46th Annual Symposium on Foundations of Computer Science (FOCS)*, 2006.

[14] R. Pass and M. Venkitasubramaniam. An efficient parallel repetition theorem for arthur-merlin games. In *the 39th Annual ACM Symposium on Theory of Computing (STOC)*, 2007.

[15] K. Pietrzak and D. Wikström. Parallel repetition of computationally sound protocols revisited. In *Theory of Cryptography, Fourth Theory of Cryptography Conference (TCC)*, 2007.

[16] R. Raz. A parallel repetition theorem. *Journal of the ACM*, 27(3):763–803, 1998. Preliminary version in *STOC'95*.