New Imperfect Random Source with Applications to Coin-Flipping

Yevgeniy Dodis
MIT*

December 4, 2000

Abstract

We introduce a new *imperfect random source* that realistically generalizes the SV-source of Sántha and Vazirani [SV86] and the bit-fixing source of Lichtenstein, Linial and Saks [LLS89]. Our source is expected to generate a known sequence of (possibly dependent) random variables (for example, a stream of unbiased random bits). However, the realizations/observations of these variables could be imperfect in the following two ways: (1) inevitably, *each* of the observations could be *slightly* biased (due to noise, small measurements errors, imperfections of the source, etc.), which is characterized by the "statistical noise" parameter $\delta \in [0, \frac{1}{2}]$, and (2) *few* of the observations could be *completely* incorrect (due to very poor measurement, improper setup, unlikely but certain internal correlations, etc.), which is characterized by the "number of errors" parameter $b \geq 0$. While the SV-source considered only scenario (1), and the bit-fixing source — only scenario (2), we believe that our combined source is more realistic in modeling the problem of extracting quasi-random bits from physical sources. Unfortunately, we show that dealing with the *combination* of scenarios (1) and (2) is dramatically more difficult (at least from the point of randomness extraction) than dealing with each scenario individually. For example, if $b\delta = \omega(1)$, the adversary controlling our source can force the outcome of any bit extraction procedure to a constant with probability $1 - o(1)$, irrespective of the random variables, their correlation and the number of observations.

We also apply our source to the question of producing n-player collective coin-flipping protocols secure against *adaptive* adversaries. While the optimal non-adaptive adversarial threshold for such protocols is known to be $n/2$ [BN00], the optimal adaptive threshold is *conjectured* by Ben-Or and Linial [BL90] to be only $O(\sqrt{n})$. We give some evidence towards this conjecture by showing that there exists no *black-box transformation* from a *non-adaptively* secure coin-flipping protocol (with arbitrary conceivable parameters) resulting in an *adaptively* secure protocol tolerating $\omega(\sqrt{n})$ faulty players.

*Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139. Email: yevgen@theory.lcs.mit.edu.

1 Imperfect Random Sources

Imperfect Random Sources. Randomization has proved to be extremely useful and fundamental in may areas of computer science, such as approximation algorithms, counting problems, distributed computing, primality testing, cryptographic protocols and many others. The common abstraction used to introduce randomness into computation is that the underlying algorithm has access to a stream of completely unbiased and independent random bits. This abstraction allows one to use randomness in a clean way, separating out the issue of actually generating such "strong" random bits. Unfortunately, in reality we do not have sources that emit perfectly uniform and independent random bits. However, there are many sources (e.g., physical sources like Geiger counters or various computer statistics like disk access times) whose outputs (which need not be bits) are believed to be "somewhat random". Such sources are called *imperfect random sources*. A large amount of research (which we survey in a second) has been devoted to filling in the gap between such realistic imperfect sources and the ideal sources of randomness (that are actually used in designing various algorithms and protocols). Very roughly, we can separate two major and quite different questions addressed when studying imperfect random sources:

- *Simulation:* can we efficiently simulate a probabilistic (BPP) algorithm with our source?
- *Extraction:* can we extract almost (or slightly) perfect randomness from our source?

Simulation has been successfully done for more and more imperfect (so called "weak") random sources [VV85, V86, CG88, CW89, Z96, ACR$^+$99], culminating in using extremely weak sources [ACR$^+$99]. These works take advantage of the fact that even though it is impossible to generate almost random bits from the corresponding weak sources, it is possible to generate random strings that avoid falling into the negligibly small set of "bad" strings. Randomness extraction, which is also the objective of this paper, would provide a more direct and clean way to use an imperfect random source in place of ideal randomness for almost any application (including simulation and many others). Thus, extraction from the source is a very desirable property to have. Unfortunately, it is also much harder to achieve than simulation, even for relatively structured imperfect random sources.

Extraction from Imperfect Sources. Most initial works in imperfect random sources [vN51, B86, SV86, LLS89, CG88] considered what we call *streaming* sources. These sources output an *ordered stream* of bits,[1] but these bits could be somewhat biased and/or correlated (exact details depend on the streaming source considered). Since these sources are studied in this paper, we survey them in more detail a bit later.

A lot of work has also been done on sources that produce (at once) a string of n bits, some of which (say, b) are adversarially fixed, but the other $(n-b)$ are truly random. The goal of extraction for such sources is to design a function (called a *resilient* function) whose output is "close" to random no matter which b input bits are fixed. It turns out that there is a huge difference depending on whether the b "fixed" bits get set before or after the $(n-b)$ random bits are chosen. In the first scenario (studied by [V87, CFG$^+$85, BBR88, F92, KJS97, DSS00]) quite positive almost optimal results are known for extracting *many* bits (one bit is trivially extracted by the parity). In the second scenario (b fixed bits are set *after* the random bits)[2], even one bit is hard to extract: the optimal b for this task lies somewhere between $\Omega(n/\log^2 n)$ [AL93] and $O(n/\log n)$ [KKL89].

Originated by Chor and Goldreich [CG88], much subsequent research has been dedicated to various flavors of the so called *weak* random sources, where no string has a very high probability of occurring.[3] While such sources are very general (and, as we mentioned, can still be used to simulate BPP algorithms), they are also too broad for any kind of randomness extraction [CG88],[4] unless we make some relaxations. For example, Trevisan and Vadhan [TV00] consider the problem of extraction from *efficiently samplable* distributions with a given min-entropy. In another major development (introduced by Nisan and Zuckerman [NZ96]), the randomness extractor is allowed to use a small number of *truly random bits* in addition to the output of a given weak source. This line of work produced an immense amount of research and found many applications (see [NT99, T99, RSW00] and the references

[1] More generally, we can talk about a stream of random variables over larger domains. We deal with such generalized streaming sources in Section 5, and stick to bits for the purposes of exposition.

[2] Such resilient functions are equivalent to 1-round/1-bit *collective coin-flipping* protocols [BL90] discussed in Section 4.

[3] Specifically, a weak source is said to have *min-entropy* m if probability of every sample is at most 2^{-m}.

[4] For example, every deterministic boolean function from N bits can be fixed to a constant by a source of (huge) min-entropy $(N-1)$.

therein). Finally, we mention a series of other works [SV86, V87, CG88, TV00] which (quite successfully) try to extract randomness from several *independent* imperfect sources, which is quite a strong assumption.

Streaming Sources. We now come back to the streaming sources. Recall, our abstraction of randomness assumes the presence of a source that emits an ordered sequence ("stream") of unbiased and independent random bits. Similarly, streaming sources emit an ordered sequence of bits, but these bits could be somewhat biased and/or correlated. In other words, any streaming source has the same "syntax" as the ideal source, but the bias of each subsequent bit could depend in some way (how exactly depends on the streaming source considered) on the "state of the source" so far. Streaming sources model (quite realistically) any process that produces "imperfect randomness" incrementally over time (for example, most of the physical sources of randomness). While such sources are usually less general than the weak random sources, the perspective of successful extraction looks much brighter for many of such sources. The study of such sources is also useful in several other regards. Firstly, such sources are quite realistic for many situations, and yet correspond more closely to the ideal sources of randomness. Secondly, they relate to the study of "discrete control processes" [LLS89] (that examine how much "control" or "influence" over a given discrete process is needed in order to force some desired event). Thirdly, they allow us to distill and study the effects of several specific complications arising when dealing with physical sources. For example, (limited) bias of the coins, measurement errors, noise, or possible internal correlations between various samples produced. And finally, various streaming sources can arise in other scenarios, like collective coin-flipping (see [LLS89] and Section 4).

Prior Streaming Sources. Perhaps the first streaming source goes all the way back to von Newman [vN51] who showed how to extract perfect random bits from a sequence of *independent* coin tosses of the *same* biased coin (of unknown bias). Elias [E72] showed how to improve this result and extract perfect random bits at the optimal rate. Blum [B86] relaxed the independence requirement on the source by considering streaming sources generated by finite-state Markov chains (of *unknown* structure), and showed how to generalize von Newman's algorithm to still extract *perfect* bits from such a source (provided the Markov chain has enough entropy).

The next important development was made by Sántha and Vazirani [SV86] who considered a more general streaming source, called a *semi-random source* (or an *SV-source*). In this source each subsequent bit can be arbitrarily correlated with *all* the previous bits, as long as it has some uncertainty. More specifically, the source is specified by the "noise" parameter $0 \leq \delta \leq \frac{1}{2}$, and can produce an arbitrary sequence of bits $x_1, x_2, \ldots$ as long as $\Pr(x_i = 1 \mid x_1 \ldots x_{i-1}) \in [\frac{1}{2}-\delta, \frac{1}{2}+\delta]$. This source tries to model the fact that physical sources can never produce *completely* perfect bits (anyway, our observation of such sources is bound to introduce some noise). Alternatively, the stream of bits could be produced by a distributive coin-flipping protocol [BL90], where few malicious players can slightly bias each of the bits. Additionally, in both of the above examples the bits can be correlated in a very non-trivial way, so we are better off without making any assumptions about the nature of these correlations (except that no bit can be completely determined from the previous bits, which is also one of the main limitations of this source).

In a parallel development, Lichtinstein, Linial and Saks [LLS89] considered another streaming source, called the (adaptive) *bit-fixing* source. In this source (characterized by the "number of errors" parameter b) each next bit, depending on the previous bits, can be either perfectly random (which is one of the main limitations of this source) or completely fixed to 0 or 1. The only constraint is that at most b of the bits are fixed. This source tries to model the situation that some of the bits generated by a physical source could be *determined* from the previous bits, even though we assume that this does not happen very frequently (at most b times). Alternatively, it relates to the study of "discrete control processes" that we mentioned earlier, as well as to the problem of adaptive coin-flipping where each player sends at most one bit (see Section 4).

We will discuss what is known about the above two sources after we introduce our source and develop some appropriate notation.

Our Goal and Organization. In this paper we study a new streaming source that examines the implications of having *both* the problems of "constant small noise" and "rare total errors", naturally generalizing random sources of [SV86, LLS89]. The paper is organized as follows. In Section 2 we introduce the "bit version" of our source.

This initial restriction to bits is done for several good reasons: clarity of presentation, closer relationship with previous work, slightly tighter results than for the general case, and, finally, technically simpler (but conceptually representative) proofs. In particular, we completely characterize the possibility of bit extraction from our source. Unfortunately, for most interesting settings of parameters, no reasonable bit extraction turns out to be possible. Next, in Section 3 we take an alternative view of our source as a discrete control process. In other words, we examine if our source has enough "power" (or imperfection) to force the (ideally random) output stream to satisfy some desired property. In particular, we derive tight bounds on how "influential" our source is in this regard. In Section 4 we have our main application to collective coin-flipping: impossibility of *black-box transformations* from statically to good adaptively secure protocols. Finally, in Section 5 we discuss our general source (which can model an arbitrary "stochastic process" and not just a sequence of unbiased bits). We show that all the "bit-results" can be extended to the general source, and discuss some implications of that.

2 Bit Version of Our Source

Motivation. Recall that Sántha and Vazirani [SV86] tried to model the problem that each of the bits produced by a streaming source is unlikely to be perfectly random: *slight* errors (due to noise, measurement errors, imperfections of the source) are *inevitable*. However, the weakness of their approach is that no bit can be completely determined from the previous bits. On the other hand, Lichtenstein, Linial and Saks [LLS89] considered the problem that some (and hopefully *few*) of the bits could have non-trivial dependencies on the previous bits (due to internal correlations, poor measurement or improper setup), to the point of being *completely determined* by them. The weakness of their approach is the assumption that the other bits are *perfect*.

While studying the above two imperfections *individually* has its advantages, we believe that their *combination* provides a more realistic view in modeling the extraction problem from physical sources. Additionally, we will see that our source relates to discrete control processes, and comes up naturally in the study of *black-box transformations* from non-adaptively to adaptively secure coin-flipping. Finally, it is very interesting to see if both imperfections (inevitable small noise and rare total errors) are sufficiently more difficult to deal with than any of them individually. Interestingly, we will show that the answer to this question is indeed positive.

Defining the Source. Our source is characterized by the "noise" parameter $\delta \in [0, \frac{1}{2}]$ and the "number of errors" parameter $b \geq 0$. It is also convenient to fix the number of bits, N, emitted by the source. Hence, our source generates N bits $x_1 \dots x_N$, where for $i = 1 \dots N$, the value of x_i can depend on $x_1 \dots x_{i-1}$ in one of the following two ways: (A) x_i could determined by $x_1 \dots x_{i-1}$ (but this can happen for at most b bits x_i), or (B) $\Pr(x_i = 1 \mid x_1 \dots x_{i-1}) \in [\frac{1}{2} - \delta, \frac{1}{2} + \delta]$. We call our source *Bias-Control Limited* (or simply BCL). Clearly, $b = 0$ corresponds to the SV-source, $\delta = 0$ yields the bit-fixing source, and $b = \delta = 0$ gives the perfect randomness.

In fact, since a good extraction function for (or any other usage of) our source should work for *any* (δ, b, N)-BCL source, it is more convenient to view our source as an "active entity" $\mathcal{A}$, usually seen as an *adversary*. Namely, in the *ideal* scenario the source would emit N truly random bits. However, this *adversary* $\mathcal{A}$ (which defines a particular (δ, b, N)-BCL source) can partially *influence* this ideal behavior. More specifically, given the outcomes of the first $(i-1)$ bits $x_1 \dots x_{i-1}$, $\mathcal{A}$ can influence the value of x_i using one of the following rules allowed:

(A) *Fix* x_i to a constant. This rule is called an *intervention*, and can be used at most b times.

(B) *Bias* x_i by any value $\leq \delta$. More specifically, set $x_i = 1$ with any probability inside $[\frac{1}{2} - \delta, \frac{1}{2} + \delta]$.[5]

Now we can quantitatively measure the "goodness" of our source for the problem of bit extraction.

Definition 1 *Let $\mathcal{A}$ be some (δ, b, N)-BCL source, and $f : \{0,1\}^N \to \{0,1\}$ be a function. Define*

- *$q(\delta, b, N, f, \mathcal{A})$ be the bias of the coin $f(x)$, where $x = x_1 \dots x_N$ was produced by $\mathcal{A}$.*
- *$q(\delta, b, N, f) = \max_{\mathcal{A}} q(\delta, b, N, f, \mathcal{A})$ (taken over all (δ, b, N)-BCL sources $\mathcal{A}$).*
- *$q(\delta, b, N) = \min_f q(\delta, b, N, f)$ (taken over all $f : \{0,1\}^N \to \{0,1\}$).*

[5] Recall, a *bias* of a bit c is defined to be $|\Pr(c = 1) - \frac{1}{2}|$.

Thus, $q(\delta, b, N)$ is the smallest bias of a coin that can be extracted from any (δ, b, N)-BCL source.

We remark that in applications δ, b and N will usually be functions of some other implicit parameter (clear from the context). For such sources we can use asymptotic notation (in this implicit parameter). In particular, we will say that one can extract an *almost* perfect bit from a (δ, b, N)-BCL source, if $q(\delta, b, N) = o(1)$, and a *slightly* random bit if $q(\delta, b, N) \leq \frac{1}{2} - \Omega(1)$. We will now survey the known results about the SV-source and the bit-fixing source, and then parallel them with our results.

Extraction from the Bit-Fixing Source. Recall, the bit-fixing source of [LLS89] corresponds to having b interventions and $\delta = 0$. Notice, that if we let f to be the majority function, we can tolerate $b = O(\sqrt{N})$ since any $c\sqrt{N}$ bits (for small enough constant c) do not influence the resulting (almost random) value of majority with probability $1 - o(1)$. Remarkably enough, Lichtinstein, Linial and Saks [LLS89] actually showed that this is the best bit extraction possible. In fact,

Theorem 1 ([LLS89]) *For any b, majority is the best bit extraction function for the bit-fixing source. In particular, $q(0, c_1\sqrt{N}, N) = o(1)$, while $q(0, c_2\sqrt{N}, N) = \frac{1}{2} - o(1)$ (for some $c_1 < c_2$).*

As a side note, a *random* function $f : \{0,1\}^N \to \{0,1\}$ is a terrible bit extraction function for the bit-fixing source even for $b = \omega(1)$. Indeed, with high probability the first $(N - b)$ bits do not fix f, so $\mathcal{A}$ can use the last b interventions to fix f. Another terrible function (even for $b = 1$) is any parity function: it can be fixed by fixing the last emitted bit. To summarize, we can tolerate $b = \Theta(\sqrt{N})$, and the majority is the best bit extraction function. However, a random function (and any parity function) is "bad" even if $b = \omega(1)$.

Extraction from the SV-source. Recall, the SV-source [SV86] corresponds to having $b = 0$, and where $\Pr(x_i = 1 \mid x_1 \dots x_{i-1}) \in [\frac{1}{2} - \delta, \frac{1}{2} + \delta]$. On a negative side, Sántha and Vazirani showed that one cannot extract a bit whose bias is less than δ. In other words, many samples (i.e., large N) from the source do not help: outputting x_1 is as good as we can get! Notationally,

Theorem 2 ([SV86]) *$q(\delta, 0, N) = \delta$. Thus, one can extract an almost perfect bit iff $\delta = o(1)$, and a slightly random bit iff $\delta = \frac{1}{2} - \Omega(1)$,*

Clearly, there are many (optimal) functions that extract a δ-biased coin from any SV-source: for example, any parity function will do. In fact, Boppanna and Narayanan [BN96] (extending the ideas of [AR89]) show that a vast majority of boolean functions from N bits extract a slighly random bit (provided, of course, $\delta = \frac{1}{2} - \Omega(1)$). Unfortunately, majority is not one of these functions (unless $\delta \ll 1/\sqrt{N}$, which will turn out to be important soon). Indeed, if our source always sets the 1-probability of the next bit to be $\frac{1}{2} + \delta$, the resulting bit will be 1 with probability $1 - o(1)$. In fact, Alon and Rabin [AR89] showed that *majority is the worst* bit extracting function. Namely, $q(\delta, 0, N, majority) \geq q(\delta, 0, N, f)$, for any f. To summarize, any parity function is an optimal bit extractor, a random function does quite well, while the majority is the worst.

Extraction from Our Source. Looking at the extreme cases of our source ($\delta = 0$ and $b = 0$), we notice that somewhat reasonable bit extraction (at least of slighltly random bits) is possible for both of them. However, the extraction functions are diametrically opposite. For the bit-fixing source the best function was the majority, and a random function (or any parity function) was terrible, while for the SV-source a random function was good (and any parity function is optimal), and the majority was the worst. Hence, the best bit extractor becomes the worst and vice versa! One may wonder if some extraction function can work reasonably well for both of these extreme cases, and hopefully provide a good extraction for our combined source as well. Unfortunately, we show that such a magic function does not exist for (any "interesting" setting of) our combined source. The following theorem is proved in Section 3:

Theorem 3 *If $b\delta = \omega(1)$, then it is impossible to extract a slightly random bit from a (δ, b, N)-BCL source, irrespective of the value of N! More precicely,*

$$q(\delta, b, N) \geq \frac{1}{2} - \frac{2}{(1+2\delta)^b} = \frac{1}{2} - \frac{1}{2^{\Omega(\delta b)-1}} \tag{1}$$

In particular, while for $\delta = 0$ we could tolerate $b = O(\sqrt{N})$ (and even extract an *almost* perfect coin), and for $b = 0$ could deal with $\delta < \frac{1}{2} - \Omega(1)$, now we cannot tolerate $b \to \infty$ for any (constant) $\delta > 0$, no matter how large N is. Also notice that the worst-case bias of any extracted coin *exponentially* approaches to $\frac{1}{2}$ as b grows.

Tightness. To see the tightness of our result, we would like to provide some intuition of why the expression $b\delta$ is important for our source. We look again at the majority function on N bits. Assume we are given b and δ (both functions of some other parameter). Under which conditions on N will the majority on N bits be a good bit extraction for (δ, b, N)-BCL source? A moment look at the binomial distribution reveals that if $N \ll b^2$, b interventions allow the adversary to almost control the coin. On the other hand, if $N \gg 1/\delta^2$, then the δ-bias at every step again allows the adversary to almost control the coin. Hence, if $b^2 \gg 1/\delta^2$, i.e. $b\delta \gg 1$, then no N will make the majority "good". This is not surprising in light of Theorem 3, but the converse statement is more interesting. It is easy to show that if $b^2 \ll 1/\delta^2$, i.e. $b\delta \ll 1$, *any* N such that $b^2 \ll N \ll 1/\delta^2$ will result in the majority being a good extractor (in fact, $N \approx b/\delta$ is the best). But what if $N > 1/\delta^2$? Of course, we know that the majority does not work then. However, we can can still trivially extract an almost random bit by simply ignoring some (say, the first or the last) $N - O(1/\delta^2)$ bits and taking the majority of the remaining $O(1/\delta^2)$ bits! Collecting these simple observations in a careful way, we get

Lemma 1 *If $b\delta = O(1)$, $b = O(\sqrt{N})$ (for small enough constants)[6] and $\delta = o(1)$, then one can extract an almost random bit from a (δ, b, N)-BCL source: $q(\delta, b, N) = o(1)$. In particular, such extraction can be done by applying the majority function to any $\min(N, O(1/\delta^2))$ bits of the source.*

The above Lemma shows the tightness of Theorem 3. Namely, for a large range of parameters we have: if $b\delta = O(1)$, then *almost* random bit can be extracted, while if $b\delta = \omega(1)$, not even a *slightly* random bit can be extracted (by Theorem 3). It is also curious to apply the above Lemma to the SV-source ($b = 0$), in particular because [AR89] showed that majority of *all* N bits is the *worst* extraction function. Well, if $\delta = O(1/\sqrt{N})$, the majority (while far from *optimal*) still extracts an almost random bit from the SV-source. If $\Omega(1/\sqrt{N}) \le \delta \le o(1)$, global majority is bad, but the majority of $O(1/\delta^2)$ bits still works. In fact, even if $\Omega(1) \le \delta \le \frac{1}{2} - \Omega(1)$, the above majority extracts a slightly random bit (notice, in this case an *almost* perfect bit is impossible by Theorem 2, since $\delta = \Omega(1)$).

Complete Picture. We also notice that Theorem 3 does not imply Theorems 1 and 2, which study the extreme cases of our source. However, by *combining* all three results with the previous discussion (in particular, Lemma 1), we get a complete characterization of the of bit extraction picture from any (δ, b, N)-BCL source (at least from the perspective of extracting almost and slightly random bits). Namely, the following list covers all the significant cases:

1. If $b = \Omega(\sqrt{N})$ or $\delta = \frac{1}{2} - o(1)$ or $b\delta = \omega(1)$, it is impossible to extract even a slightly random bit. These results follow from Theorem 1 (even for $\delta = 0$), Theorem 2 (even for $b = 0$) and Theorem 3 respectively.
2. If $\Omega(1) \le \delta \le \frac{1}{2} - \Omega(1)$ and $b = O(1)$, then one can extract a slightly random bit, but cannot extract an almost random bit (the lower bound follows from Theorem 2).
3. If $b = O(\sqrt{N})$ and $b\delta = O(1)$ and $\delta = o(1)$, then one can extract an almost random bit from our source. This is exactly Lemma 1.

To have yet another insight on these results, we can let $\sigma \stackrel{\text{def}}{=} \max(\delta, O(1/\sqrt{N}))$ to be the "effective noise" of our source. In other words, if $\delta \ll 1/\sqrt{N}$, increasing δ to $1/\sqrt{N}$ does not change the behavior of the source much. Then we can restate our main result as follows: when $b\sigma = \omega(1)$, no good extraction is possible, and if $b\sigma = O(1)$, good extraction becomes possible.

Expected Number of Interventions to Fix the Outcome. We also study another bit extraction measure of our source: the *expected* number of interventions to *always* fix the extracted coin (to 0 or 1). Due to space limitations, this is discussed in Appendix A, where we show that $O(1/\delta)$ expected interventions suffice irrespective of N.

[6] To avoid verbosity in the future discussion, the statement $\alpha = O(\beta)$ should be read as "$\alpha \le c\beta$ for a small enough (rather than any) constant c whose value is not important for the discussion", and similarly for $\alpha = \Omega(\beta)$.

3 Our Source as a Discrete Control Process

Alternative View of Our Source. Recall that we view our source as an adversary $\mathcal{A}$ who can *influence* the ideal behavior of the source by applying rules (A) and (B). So far we considered the task of $\mathcal{A}$ to be preventing good bit extraction. However, an equally (if not more) natural task for $\mathcal{A}$ would be to try to force some particular *event*, i.e. to force the string $x = x_1 \dots x_N$ to satisfy some particular property. For example, $\mathcal{A}$ may try to make the source emit more 1's than 0's (i.e., force the majority function to true). To formalize this, let $\mathcal{E}$ be an event (or property) on $\{0,1\}^N$. Equivalently, $\mathcal{E}$ can be viewed as a boolean function $e : \{0,1\}^N \to \{0,1\}$, or as a *language* $L = e^{-1}(1) \subseteq \{0,1\}^N$, via "$\mathcal{E}$ happened $\iff e(x) = 1 \iff x \in L$".

We can define the *natural probability* p of $\mathcal{E}$ to be the probability that $\mathcal{E}$ happened for the *ideal* source (in our case, emitting N perfect unbiased bits), i.e. $p = |L|/2^N$. We then say that $\mathcal{E}$ is *p-sparse*. Now we want to see if our adversary $\mathcal{A}$ has enough power to significantly influence the occurrence of $\mathcal{E}$ (i.e., to make $x \in L$). Such $\mathcal{A}$ can be viewed as a "controller": it takes "no effort" for $\mathcal{A}$ to *slightly* influence each x_i (i.e., apply rule (B)), and it takes "significant effort" to *control* x_i (i.e., apply rule (A)). Now, two dual questions naturally come up for a given δ, N and $\mathcal{E}$ (with natural probability p):

1. For a given number of interventions b, what is the largest probability of "success" that $\mathcal{A}$ can achieve? In particular, under what conditions can it be arbitrarily close to 1? Can the answer(s) depend on p but not on other specifics of $\mathcal{E}$?
2. Assume we want to *guarantee* success ($\mathcal{E}$ always happens), by allowing possibly unbounded number of interventions. What is the smallest *expected* number of interventions needed? Can the bound depend on p but not on other specifics of $\mathcal{E}$?

We define two natural measuress that allow us to study the quantities addressed in the above questions. For the first question, it is actually more convenient to study the complement notion of "smallest probability of failure" (i.e., to minimize $\Pr(e(x) = 0)$). Since δ is never going to change in our discussion, we omit it from all the notation below (even though all the bounds depend on δ).

Definition 2 *Define*

- $F(p, N, b) = \max_{\mathcal{E}} \min_{\mathcal{A}} \Pr(e(x) = 0)$*, taken over all p-sparse $\mathcal{E}$, and all (δ, b, N)-BCL $\mathcal{A}$.*
- $B(p, N) = \max_{\mathcal{E}} \min_{\mathcal{A}} \mathbf{E}[b]$*, taken over all p-sparse $\mathcal{E}$ and all N-bit sources $\mathcal{A}$ (with noise δ) necessarily producing x satisfying $\mathcal{E}$. Here $\mathbf{E}[b]$ stands for the expected number of interventions used by $\mathcal{A}$ (the expectation is over the usage of rule (B)).*

Thus, $F(p, N, b)$ is the worst (largest) probability of $\mathcal{A}$'s failure over all p-sparse events, and $B(p, N)$ is the smallest expected number of interventions $\mathcal{A}$ needs to always force any p-sparse $\mathcal{E}$. Notice, both quantities take the worst case w.r.t. p-sparse $\mathcal{E}$.

Bounding the Probability of Failure. We start with a tight bound on $F(p, N, b)$.

Theorem 4

$$F(p, N, b) \le \frac{1}{p \cdot (1 + 2\delta)^b} = 2^{\log(1/p) - \Theta(\delta b)} \tag{2}$$

In particular, if $\delta b = \omega(\log(1/p))$, $\mathcal{A}$ can force any p-sparse $\mathcal{E}$ with probability $1 - o(1)$.

We notice that N does not enter the equation. We also notice that Theorem 4 immediately implies Theorem 3. Indeed, for any bit extraction function f, the optimal way to bias the extracted coin is to try to force $f(x) = 0$ or $f(x) = 1$. Since one of these events has natural probability $p \ge 1/2$, the bound of Theorem 3 follows. Finally, the bound is almost tight, at least in several significant cases. For example, for $p = \frac{1}{2}$ we argued earlier that $\mathcal{A}$ cannot almost certainly force 1 on the majority of $\min(N, 1/\delta^2)$ bits when $\delta b = O(1)$. On the other hand, if e is the function that is 1 on the first $p2^N$ values of x (in the lexicographic order), $\mathcal{A}$ has to intervene at least $\Omega(\log(1/p))$ times in order to force $e(x) = 1$ with probability more than $\frac{1}{2} + \delta$.

Proof: The statement is true for $\delta = 0$ or $b = 0$, since $F(\cdot,\cdot,\cdot) \le 1 \le 1/p$, so assume $\delta > 0$ and $b \ge 1$. Define $g(p,b) = \frac{1}{p(1+2\delta)^b}$. We need to show that $F(p,N,b) \le g(p,b)$ for any $N \ge 1$, $1 \le b \le N$ and $0 \le p \le 1$. We prove this by induction on N. For $N = 1$, $F(0,1,b) = 1 < \infty = g(0,b)$, and $F(p,1,b) = 0 \le g(p,b)$ for $p > 0$ (here we used $b \ge 1$). Assume now the claim is true for $(N-1)$ and we want to show it for N.

Take any p-sparse $\mathcal{E}$ given by a function e. Let $e_0 : \{0,1\}^{N-1} \to \{0,1\}$ be the restriction of e when $x_1 = 0$. Similarly for e_1. This defines a p_0-sparse event $\mathcal{E}_0$ and a p_1-sparse event $\mathcal{E}_1$ satisfying $\frac{1}{2}(p_0 + p_1) = p$. Without loss of generality assume $p_0 \ge p \ge p_1$. Given such $\mathcal{E}$, our particular adversary $\mathcal{A}$ will consider two options and pick the best (using his unbounded computational resources): either he will use an intervention (he can do it since we assumed $b \ge 1$) and fix $x_1 = 0$, reducing the question to that of analyzing the p_0-sparse event $\mathcal{E}_0$ on $(N-1)$ variables and also reducing b by 1, or he will use rule (B) making the 0-probability of x_1 equal to $\frac{1}{2} + \delta$ and leaving the same b. By the definition of function $F(p,N,b)$, we know that in the first case the failure probability of $\mathcal{A}$ will be at most $F(p_0, N-1, b-1)$, and in the second case it will be at most $(\frac{1}{2} - \delta)F(p_1, N-1, b) + (\frac{1}{2} + \delta)F(p_0, N-1, b)$. Since the choice of $p_0 \ge p_1$ (i.e., how $\mathcal{E}$ splits into $\mathcal{E}_0$ and $\mathcal{E}_1$) such that $p_0 + p_1 = 2p$ is outside of our control, we will take the maximum over all such choices and obtain the following recurrence.

$$F(p,N,b) \le \max_{\substack{p_0 \ge p_1 \\ p_0+p_1=2p}} \min\left[F(p_0,N-1,b-1),\ \left(\frac{1}{2}-\delta\right)\cdot F(p_1,N-1,b) + \left(\frac{1}{2}+\delta\right)\cdot F(p_0,N-1,b)\right]$$

Let $p_0 = p(1+\beta)$ and $p_1 = p(1-\beta)$, where $0 \le \beta \le 1$ (since $p_0 \ge p \ge p_1$). Using our inductive assumption,

$$F(p,N,b) \le \max_{0\le\beta\le 1} \min\left(g(p(1+\beta),b-1),\ \left(\frac{1}{2}-\delta\right)\cdot g(p(1-\beta),b) + \left(\frac{1}{2}+\delta\right)\cdot g(p(1+\beta),b)\right) \stackrel{?}{\le} g(p,b)$$

Recalling the definition of g, it thus suffices to show that

$$\max_{0\le\beta\le 1} \min\left(\frac{1}{p(1+\beta)(1+2\delta)^{b-1}}\ ,\ \frac{\frac{1}{2}-\delta}{p(1-\beta)(1+2\delta)^b} + \frac{\frac{1}{2}+\delta}{p(1+\beta)(1+2\delta)^b}\right) \le \frac{1}{p(1+2\delta)^b}$$

$$\iff \max_{0\le\beta\le 1} \min\left(\frac{1+2\delta}{1+\beta}\ ,\ \frac{\frac{1}{2}-\delta}{1-\beta} + \frac{\frac{1}{2}+\delta}{1+\beta}\right) \le 1$$

We see that the expressions under the minimum are equal when $\beta = 2\delta$. We consider two cases.

- Case 1. Assume $\beta \ge 2\delta$. Then the minimum above is $\frac{1+2\delta}{1+\beta}$ and it suffices to show that $\frac{1+2\delta}{1+\beta} \le 1$, which is equivalent to our assumption on β.
- Case 2. Assume $\beta \le 2\delta$. Then the minimum above equals to $\frac{\frac{1}{2}-\delta}{1-\beta} + \frac{\frac{1}{2}+\delta}{1+\beta}$ and it suffices to show that $\frac{\frac{1}{2}-\delta}{1-\beta} + \frac{\frac{1}{2}+\delta}{1+\beta} \le 1$. But this is again equivalent to our assumption on β.

■

Bounding Expected Number of Interventions. We also show a tight bound on $B(p,N)$. Namely, $B(p,N) = O(\frac{1}{\delta}\log(1/p))$ (in particular, this bound is independent on N). Due to space limitations, the discussion and the proof appear in Appendix B.

4 Our Source and Collective Coin-Flipping

The Setting. *Collective Coin-Flipping* in the full-information model was introduced by Ben-Or and Linial [BL90]. In this model n computationally unbounded processors are trying to generate a random bit in a setting where only a *single broadcast channel* is available for communication. As usual, we assume that some of the players (at most b out of n, though) can be *faulty* or malicious, and in fact is controlled by a central adversary $\mathcal{A}$ (which is called b-bounded). In each round of the protocol every player can broadcast a message to the other players. A crucial complication is that the network is *asynchronous within a round*. For example, players cannot flip a coin by broadcasting a random bit and taking their exclusive OR: the last player to talk can completely control the output. Again taking the worst case scenario, we assume that in each round first $\mathcal{A}$ receives all the messages broadcast by the honest players, and only then decides which messages to send on behalf of the bad players. The output of the protocol is some pre-agreed deterministic function of the messages exchanged over the broadcast channel.

The Goal. As we said, the objective of collective coin-flipping (parameterized by the number of players, n) is for the players to agree on a "random" bit, even in the presence of an adversary. Of course, the adversary $\mathcal{A}$ will introduce some bias into the coin. We let $\Delta_\Pi(b)$ be the largest bias achieved by a b-bounded adversary against protocol Π. Then, a coin-flipping protocol Π is said to be *(weakly)* $b(n)$*-resilient* if Π produces a *slightly* random coin: $\Delta_\Pi(b(n)) \leq \frac{1}{2} - \Omega(1)$, where the constant is *independent* of n. Similarly, Π is said to be *strongly* $b(n)$*-resilient* if Π produces an *almost* random coin: $\Delta_\Pi(b(n)) = o(1)$. Traditionally, the "standard" definition of resilience for coin-flipping is that of weak resilience, so this is the notion that we will use.[7]

Type of Adversary. So far we have been very vague about the type of adversary that we have. Perhaps most importantly, we have not talked about how and when the players becomes faulty. Most of the papers in the full-information model assume and *crucially use* the fact that the adversary $\mathcal{A}$ is *static* (or non-adaptive), i.e. it decides on which b parties to corrupt *before the protocol starts*. The honest players do not know which b players were selected by $\mathcal{A}$, but the resulting coin has to be slightly random for any *fixed* set of b players. A somewhat more realistic and much more powerful type of an adversary is an *adaptive* adversary. This adversary can listen to all the communication and corrupt up to b players anywhere *in the course of the execution*.

Coin-Flipping with Static Adversaries. The optimal resilient threshold for static adversaries in $n/2$: any $n/2$ players can always fix the coin [S89, BN00], while there exist $(\frac{1}{2} - \varepsilon)$-resilient protocols (even constructive and efficient ones) for any $\varepsilon > 0$ [BN00, ORV94, RZ98, F99]. We also point out a very simple dependence of the optimal bias $\Delta(b)$ (defined to be the smallest bias achieved by a coin-flipping protocol: $\min_\Pi \Delta_\Pi(b)$) on the number of players: $\Delta(b) = \Theta(b/n)$. The lower bound (which we will use in a second) was elegantly shown by Ben-Or and Linial [BL90], while the upper bound eventually followed from the series of works of [BL90, AL93, AN93] (for larger and larger b). Finally, we point out that *all the best statically secure coin-flipping protocols are not even* 1*-resilient against adaptive adversaries*. This is due to a historical feature that all such protocols first elect a single (hopefully, not faulty) representative player (called a *leader*), who then flips the final coin by itself. Corrupting such a leader at the end clearly controls the coin. More generally, the whole philosophy of most statically secure protocols seems to be not applicable in the adaptive world, as these protocols try to aggressively "eliminate" players (since a "patient" adaptive adversary can corrupt the few remaining players).

Coin-Flipping with Adaptive Adversaries. Adaptive adversaries were already considered in the original paper of Ben-Or and Linial [BL90]. In particular, it was observed there that the "majority" protocol (each player sends a random bit, and the final coin is their majority) achieves adaptive $\Theta(\sqrt{n})$-resilience. Surprisingly enough, this simple protocol is the *best known* adaptively secure coin-flipping protocol! In fact, Ben-Or and Linial [BL90] conjectured that this protocol to be optimal!

Conjecture 1 ([BL90]) *Majority is the optimal coin-flipping protocol against adaptive adversaries. In particular, the maximum threshold that can be tolerated is* $O(\sqrt{n})$.

This conjecture, if true, would imply that adaptive adversaries are much more powerful than static adversaries for the problem of collective coin-flipping (which can tolerate up to $n/2$ faulty players). Interestingly enough, the only result that in support of this conjecture comes from the bit-fixing source of [LLS89]. Namely, it is easy to see than when each player sends only 1 bit in the entire protocol, the optimal behavior of the adversary is exactly the same as in the bit-fixing source with b interventions! Since the majority was the best bit extraction function for the bit-fixing source, we get that Conjecture 1 if true if each player is restricted to send only 1 bit. This result is interesting since is already illustrates the power of adaptivity. Namely, in the static case one can achieve $\Omega(n/\log^2 n)$-resilience [AL93] when players send only 1 bit, even in one round. However, the above result supports Conjecture 1 much less than it seems to. Indeed, restricting each player to send at most 1 bit seems like a huge limitation. For example, we saw that it was very limiting *even for statically secure protocols* (recall, no function can be more than $O(n/\log n)$-resilient by the result of [KKL89], and there are general $n/2$-resilient statistically secure protocols [BN00, ORV94, RZ98, F99]). For adaptively secure protocols, the limitation seems to be even more severe (even though it would not be if Conjecture 1 was true).

[7] In fact, since our main result for coin-flipping is an impossibility result, it will become only stronger with strong resilience.

To summarize, adaptively secure coin-flipping is much less understood than its static counter-part, there seems to be some indication that adaptive adversaries are much more powerful than static adversaries, but there is little formal evidence supporting this claim.

Our Approach and Impossibility Result. Due to space limitations, we leave the formal treatment of the remainder of this section to Appendix C, and instead provide informal (but informative) intuition of our approach. We give another partial support to Conjecture 1 by looking at the problem of adaptivity from an entirely different angle: we examine the question of whether it is possible to obtain an adaptively secure coin-flipping protocols "for free"? More specifically, can we transform some good statically secure protocol Π so as to obtain a reasonable adaptively secure protocol, where the proof of adaptive security should only depend only on the knowledge that Π is statically secure (and not on any other specifics about Π)? While a positive answer to such kind of a question would seem astonishing (and, indeed, our answer will be mainly negative), we formulate the question in such a way that (at least from the first look) the negative answer is not obvious at all (in fact, we *will* be able to achieve $O(\sqrt{n})$-resilience, which is believed to be optimal, but show that our approach does not allow us to break this barrier).

We will try to sequentially run Π many (say, N) times against an adaptive adversary $\mathcal{A}$ who can corrupt up to b players. These runs produce coins $x_1 \dots x_N$. Of course, since we run a static protocol against an adaptive adversary, some of the x_i's might be very biased. However, $\mathcal{A}$ can corrupt at most b players! Thus, at least $(N-b)$ of the subprotocols were effectively run against a *static* adversary, and therefore produced somewhat random coins. Let us say that the bias of these coins is at least δ (which depends on b and the properties of Π). But then, even if the other b runs produced x_i which were completely fixed by $\mathcal{A}$,[8] we can view the resulting $x = x_1 \dots x_N$ as being produced by a (δ, b, N)-BCL source!

Notice also that b and δ depend on the properties of n, Π and our objective (how good of an adaptive protocol we want), and hence could be viewed as fixed. However, we have the power to make N arbitrarily huge, which seems to give us a considerable advantage. Unfortunately, the strong negative result of Theorem 3 shows that this advantage is, actually, an illusion. Namely, recall from our results that for known b and δ, the possibility of bit extraction from (δ, b, N)-BCL source depends on whether $b\delta = O(1)$ or $b\delta = \omega(1)$, i.e. a large number of repetitions N does *not* help. Nevertheless, when is $b\delta = O(1)$? Notice that the best δ we could hope for (without looking at the specifics of Π), while definitely no *more* than $\Delta(b)$, can not be much *less* than $\Delta(b) = \Theta(b/n)$ as well. For example, at the very beginning $\mathcal{A}$ could corrupt $b/2$ players that can achieve $\delta \geq \Delta(b/2) = \Theta(\Delta(b))$, and still have $b/2$ arbitrary corruptions left. Hence, our "black-box" approach can work (and actually *can be made* to work) only if $b \cdot \Theta(b/n) = O(1)$, i.e. $b = O(\sqrt{n})$. Since such b can be trivially achieved by the majority protocol, we cannot achieve adaptive security (beyond what is known) "for free".

Discussion. We are not claiming that black-box transformations are the most natural way to approach adaptive security. They are certainly not (in particular, Conjecture 1 reminds widely open). However, we feel that this approach is something that had to be tried, and, at least on the first look, our approach did seem quite promising (e.g., we could make N arbitrary large and hope to circumvent $b \ll N$ interventions). In fact, the proof that the approach fails is not trivial, and the "breaking point" is exactly (believed to be optimal) $b = \Theta(\sqrt{N})$. The latter "coincidence" does give some further evidence to Conjecture 1. Finally, the above connection to coin-flipping is a surprising application of our new source.

5 Our General Source

Modeling Any Stochastic Process. We finally introduce the general version of our source. So far we examined of question of how much various imperfections can affect an "ideal" stream of *unbiased random bits*. While this is an extremely natural ideal stream to consider, a lot of physical (and other) streaming sources of randomness do

[8] The "worst-case" assumption that corrupting even one player in Π will allow an adaptive $\mathcal{A}$ to control the coin might appear problematic. We point out three answers for that: (1) we are looking at *black-box* transformations; (2) all the best known static protocols are not adaptively 1-resilient, and (3) even some relaxed assumptions (e.g., corruption of c players can control the coin) will allow us to get weaker but non-trivial bounds.

not (even in the "ideal" scenario) produce a stream of bits, and what they produce need not be uncorrelated as well. In a much more general scenario, we can consider an arbitrary "ideal" *stochastic process* $\mathcal{P}$ that produces a sequence of random variables $X_1, X_2, \ldots$. The (known) ideal distribution (and even the domain!) of X_i can arbitrarily depend on the realizations of $X_1 \ldots X_{i-1} = x_1 \ldots x_{i-1}$. We denote this conditional distribution by $D_i = D_i(x_1 \ldots x_{i-1})$. Now, similarly to the "bit-case", we can study the effects of two imperfections on this ideal source: inevitable small statistical deviation of each X_i from D_i, and rare complete errors in the process. In fact, almost all the notions from the bit-case can be naturally extended, as long as we replace the notion of bias by a more general notion of a *statistical distance*.[9] In particular, we can view our general (δ, b, N)-BCL source (w.r.t. to a particular stochastic process $\mathcal{P}$ in mind) as an adversary $\mathcal{A}$ who, for $i = 1 \ldots N$ and given $x_1 \ldots x_{i-1}$, can influence the ideal sample of X_i using one of the following rules:

(A) Fix X_i to any constant in the support of D_i. This rule can be used at most b times, however.

(B) Sample X_i from any distribution D_i' (on the same support set) of statistical distance at most δ from D_i.

Power of Our Source. In particular, we can study our general (δ, b, N)-BCL source in relation to discrete control processes. For any event $\mathcal{E}$, we can define the *natural probability* p of $\mathcal{E}$ (w.r.t. $\mathcal{P}$), and in the same manner as before talk about quantities $F(p, N, b)$ and $B(p, N)$ studied in Section 3. In particular, $F(p, N, b)$ tells us the largest probability of $\mathcal{A}$'s failure to force some p-sparse event, and $B(p, N)$ studies the expected number of interventions needed to enforce any such $\mathcal{E}$. We obtain an (essentially) equally strong (but much more general) analog of Theorems 4 and 7.

Theorem 5

- $F(p, N, b) \leq \frac{(1-\delta)^b}{p} = 2^{\log(1/p) - \Omega(\delta b)}$.
  Thus, $\delta b = \omega(\log(\frac{1}{p})) \Rightarrow$ *$\mathcal{A}$ can force any p-sparse event with probability* $1 - o(1)$.
- $B(p, N) \leq \log_{1-\delta} p = O(\frac{1}{\delta} \cdot \log(1/p))$.

We notice the generality of this result: it holds for arbitrary stochastic process $\mathcal{P}$, arbitrary p-sparse events w.r.t. $\mathcal{P}$ (both of which are not chosen by $\mathcal{A}$), and the bounds do not explicitly depend on N. The proof of the above result is conceptually the same as what we had for the "bit-source". However, the generality of the statement makes the "algebra" and the details somewhat more challenging. The proof can be found in Appendix D.

Sampling General Distributions. We conclude by briefly pointing another implication of Theorem 5. First, if $\delta b = \Omega(1)$, no ideal process $\mathcal{P}$ yields a (δ, b, N)-BCL source where one could extract even a single slightly random bit. More generally, one can examine the question of sampling other distributions, and get the following impossibility result. Assume f is an extraction function that ideally extracts a random variable Y from our source. Assume the objective of $\mathcal{A}$ is to have our source generate Y' with the largest statistical distance from Y. For any event $\mathcal{E}$ on Y of natural probability p, we know that $\delta b = \omega(\log(1/p))$ implies that $\mathcal{A}$ can produce Y' satisfying $e(Y') = 1$ with probability $1 - o(1)$. Notice, in this case $\|Y - Y'\| \geq \|e(Y) - e(Y')\| = p - o(1)$. Hence, if we define the *fairness* $\gamma(Y)$ *of* Y[10] to be the largest $p \leq \frac{1}{2}$ such that some event $\mathcal{E}$ has natural probability p w.r.t. Y, Theorem 5 implies the following:

Corollary 1 *Assume an extraction procedure f ideally extracts a variable Y from a (δ, b, N)-BCL source, and let $\gamma = \gamma(Y)$ be the fairness of Y. Then*

- *If $\delta b = \omega(1)$, then $\mathcal{A}$ can produce Y' satisfying $\|Y - Y'\| \geq \gamma - o(1)$.*
- *If $\delta b = \omega(\log(1/\gamma))$, then $\mathcal{A}$ can produce Y' satisfying $\|Y - Y'\| \geq 1 - \gamma - o(1) \geq \frac{1}{2} - o(1)$.*

We notice that fairness of Y measures how good of a coin-flip we can deterministically extract from Y. We remark that any natural distribution has fairness $\Omega(1)$.[11] Thus, the above result says that for any such natural Y, $\delta b = \omega(1)$ implies that $\mathcal{A}$ can influence Y into Y' that is *statistically far* from Y.

[9] Recall, the *statistical distance* between random variables Z and W over a domain R is $\|Z - W\| = \frac{1}{2} \cdot \sum_{\alpha \in R} |\Pr(Z = \alpha) - \Pr(W = \alpha)|$. The same notation stands for the distributions generating Z and W.

[10] If Y is a bit, $\gamma(Y)$ is indeed $\min(\Pr(Y = 0), \Pr(Y = 1))$.

[11] For example, $\gamma(Y) \geq \frac{1}{2}(1 - \max_y \Pr(Y = y))$. Thus, $\gamma(Y) = o(1) \Rightarrow \exists y \ \Pr(Y = y) = 1 - o(1)$, making Y "almost constant".

References

[AL93] M. Ajtai, N. Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993.

[AN93] N. Alon, M. Naor. Coin-flipping games immune against linear-sized coalitions. *SIAM J. Comput.*, 22(2):403-417, 1993.

[AR89] N. Alon, M. Rabin. Biased Coins and Randomized Algorithms. *Advances in Computing Research*, 5:499-507, 1989.

[ACR+99] A. Andreev, A. Clementi, J. Rolim, L. Trevisan. Dispersers, deterministic amplification, and weak random sources. In *SIAM J. on Comput.*, 28(6):2103–2116, 1999.

[BBR88] C. Bennett, G. Brassard, and J. Robert. Privacy Amplification by public discussion. *SIAM J. on Computing*, 17(2):210–229, 1988.

[BL90] M. Ben-Or, N. Linial. Collective Coin-Flipping. In Slivio Micali, editor, *Randomness and Computation*, pp. 91-115, Academic Press, New York, 1990.

[B86] M. Blum. Independent unbiased coin-flipsfrom a correclated biased source — a finite state Markov chain. *Combinatorica*, 6(2):97–108, 1986.

[BN96] R. Boppana, B. Narayanan. The Biased Coin Problem. *SIAM J. Discrete Math.*, 9(1)29–36, 1996.

[BN00] R. Boppana, B. Narayanan. Perfect-information Leader Election with Optimal Resilience. *SIAM J. Comput.*, 29(4):1304-1320, 2000.

[CG88] B. Chor, O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Comput.*, 17(2):230–261, 1988.

[CFG+85] B. Chor, J. Friedman, O. Goldreich, J. Håstad, S. Rudich, R. Smolensky. The Bit Extraction Problem or t-resilient Functions. In *Proc. of* 26*th FOCS*, pp. 396–407, 1985.

[CW89] A. Cohen, A. Wigderson. Dispersers, deterministic amplification, and weak random sources. In *Proc. of* 30*th FOCS*, pp. 14–19, 1989.

[CL95] J. Cooper, N. Linial. Fast perfect-information leader-election protocols with linear immunity. *Combinatorica*, 15:319–332, 1995.

[DSS00] Y. Dodis, A. Sahai and A. Smith. All-or-Nothing Transforms, Resilient Functions and Adaptive Security. Preliminary version in submission, October 2000.

[E72] P. Elias. The Efficient Construction of an Unbiased Random Sequence. *Ann. Math. Stat.*, 43(3):865–870, 1972.

[F99] U. Feige. Noncryptographic Selection Protocols. In *Proc. of* 40*th FOCS*, pp. 142–152, 1999.

[F92] J. Friedman. On the Bit Extraction Problem In *Proc. of* 33*rd FOCS*, pp. 314–319, 1992.

[GGL98] O. Goldreich, S. Goldwasser, N. Linial. Fault-Tolerant Computation in the Full Information Model. *SIAM J. Comput.*, 27(2):506–544, 1998.

[KKL89] J. Kahn, G. Kalai, N. Linial. The Influence of Variables on Boolean Functions. In *Proc. of* 30*th FOCS*, pp. 68–80, 1989.

[KJS97] K. Kurosawa, T. Johansson and D. Stinson. Almost k-wise independent sample spaces and their cryptologic applications. In *Proc. of EuroCrypt*, pp. 409–421, 1997.

[LLS89] D. Lichtenstein, N. Linial, M. Saks. Some Extremal Problems Arising from Discrete Control Processes. *Combinatorica*, 9:269–287, 1989.

[NT99] N. Nisan, A. Ta-Shma. Extracting Randomness: a survey and new constructions. In *JCSS*, 58(1):148–173, 1999.

[NZ96] N. Nisan, D. Zuckerman. Randomness is Linear in Space. In *JCSS*, 52(1):43–52, 1996.

[ORV94] R. Ostrovsky, S. Rajagopalan, U. Vazirani. Simple and Efficient Leader Election in the Full Information Model. In *Proc. of* 26*th STOC*, pp. 234–242, 1994.

[RSW00] O. Reingold, R. Shaltiel, A. Wigderson. Extracting randomness via repeated condensing. To appear in *Proc. of* 41*st FOCS*, 2000.

[RSZ99] A. Russell, M. Saks, D. Zuckerman. Lower bounds for leader election and collective coin-flipping in the perfect information model. In *Proc. of* 31*st STOC*, pp. 339–347, 1999.

[RZ98] A. Russell, D. Zuckerman. Perfect information leader election in $\log^* n + O(1)$ rounds. In *Proc. of* 39*th FOCS*, pp. 576–583, 1998.

[S89] M. Saks. A robust noncryptographic protocol for collective coin flipping. *SIAM J. Discrete Math.*, 2(2):240–244, 1989.

[SV86] M. Sántha, U. Vazirani. Generating Quasi-Random Sequences from Semi-Random Sources. *Journal of Computer and System Sciences*, 33(1):75–87, 1986.

[T99] L. Trevisan. Construction of Extractors Using PseudoRandom Generators. In Proc. of 31st STOC, pp. 141–148, 1999.

[TV00] L. Trevisan, S. Vadhan. Extracting Randomness from Samplable Distributions. To appear in *Proc. of* 41*st FOCS*, 2000.

[V86] U. Vazirani. Randomness, Adversaries and Computation. *PhD Thesis*, University of California, Berkeley, 1986.

[V87] U. Vazirani. Strong Communication Complexity or Generating Quasi-Random Sequences from Two Communicating Semi-Random Sources. *Combinatorica*, 7(4):375–392, 1987.

[VV85] U. Vazirani, V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *Proc. of* 26*th FOCS*, pp. 417–428, 1985.

[vN51] J. von Newman. Various techniques used with connection with random digits. In *National Bureau of Standards, Applied Math. Series*, 12:36–38, 1951.

[Z96] D. Zuckerman. Simulating BPP Using a General Weak Random Source. *Algorithmica*, 16(4/5):367-391, 1996.

A Expected Number of Interventions to Fix the Coin

We also define another related measure for our source. Here we assume that our adversary, rather than trying to bias the coin with a limited number of interventions, is actually trying to *always* fix the coin (to 0 or 1), trying to minimize the *expected number of interventions* (hence, there is no absolute bound on the number of interventions). For example, if f is the majority on N bits, and the adversary $\mathcal{A}$ tries to make $f(x) = 1$, the optimal behavior of

$\mathcal{A}$ is the following. If setting $x_1 = 0$ will ensure that $f(x) = 0$ (i.e., there are $N/2 - 1$ zeros in $x_1 \ldots x_{i-1}$), use the intervention and set $x_i = 1$. Otherwise, use rule (B) making the bias of x_i toward 1 equal to δ.

We let $v(\delta, N)$ be the smallest expected number of interventions that suffice to fix the coin (to 0 or 1) for any bit extraction procedure applied to an N-bit source with noise parameter δ. We remark that this quantity does not make sense for the SV-source, but was studied for the bit-fixing source (corresponding to $\delta = 0$). In particular, Lichtenstein et al. [LLS89] showed that $v(0, N) = \Theta(\sqrt{N})$. While the optimal function is *not* the majority, majority is "close" and requires $\Omega(\sqrt{N})$ expected interventions as well. We show that

Theorem 6
$$v(\delta, N) \leq O\left(1/\delta\right) \tag{3}$$

In particular, if $\delta = \Omega(1)$, a constant expected number of interventions suffice irrespective of N!

We notice that the above result does not imply the result of [LLS89] that $v(0, N) = \Theta(\sqrt{N})$. However, by combining these two results we get the "complete picture". Indeed, it is easy to see that majority requires $\Omega(\sqrt{N})$ expected interventions even if $\delta = O(1/\sqrt{N})$. Thus, $v(\delta, N) = \Theta(\sqrt{N})$ for $\delta = O(1/\sqrt{N})$. On the other hand, the same argument shows that when $\delta = \Omega(1/\sqrt{N})$, majority of $O(1/\delta^2)$ (which is less than N) arbitrary bits of our source requires $\Omega(1/\delta)$ expected interventions, which is optimal (up to a constant factor) by Theorem 6. Recalling the definition of "effective noise" $\sigma = \max(\delta, O(1/\sqrt{N}))$, we get

Corollary 2 *For any δ and and N, $v(\delta, N) = \Theta(\min(1/\delta, \sqrt{N})) = \Theta(1/\sigma)$.*

The proof of Theorem 6 will follow from the discussion in Section 3 and Appendix B.

B Expected Number of Interventions to Force an Event

To state our bound on the number of interventions, it is more convenient to work with $\gamma \stackrel{\text{def}}{=} \frac{1}{2} - \delta$. This γ can be viewed as the minimal *fairness*[12] of the coin influenced by rule (B). We start from the following easily verified analytical Lemma whose proof we omit.

Lemma 2 *For any $0 < \gamma < \frac{1}{2}$ the equation*
$$z^{\frac{1}{\gamma}} + 1 = 2 \cdot z^{\frac{1}{\gamma} - 1} \tag{4}$$

has a unique solution $z_\gamma \in (1, 2)$. In addition, z_γ is a continuous decreasing function of γ such that $\lim_{\gamma \to 0} z_\gamma = 2$, $\lim_{\gamma \to \frac{1}{2}} z_\gamma = 1$, $\log_2 z_\gamma = \Theta(1 - 2\gamma)$, and for all $1 \leq w \leq z_\gamma$ we have $w^{1/\gamma} + 1 \leq 2 \cdot w^{1/\gamma - 1}$.

Theorem 7
$$B(p, N) \leq \log_{z_\gamma}(1/p) = O\left(\frac{1}{1 - 2\gamma} \cdot \log(1/p)\right) = O\left(\frac{1}{\delta} \cdot \log(1/p)\right) \tag{5}$$

Again, notice that N does not enter the equation (the only dependence from N comes implicitly from p). We also notice that since each bit extraction function has either a majority of 1's or a majority of 0's, Theorem 7 immediately implies the bound given by Theorem 6. Finally, the bound is almost tight, at least in several significant cases. The examples are the same as for Theorem 4.

Proof: The proof is very analogous to that of Theorem 4. Let $z = z_\gamma$ and define $h(p) = \log_z(1/p)$. We need to show that $B(p, N) \leq h(p)$ for any $N \geq 1$ and $0 \leq p \leq 1$. We prove this by induction on N. For $N = 1$, $B(0, 1) = \infty = h(0)$, and $B(\frac{1}{2}, 1) = 1 \leq \log_z 2 = h(\frac{1}{2})$ (since $z \leq 2$) and $B(1, 1) = 0 = h(1)$. Assume now the claim is true for $(N - 1)$ and we want to show it for N.

Let p_0, p_1, $\mathcal{E}_0$, $\mathcal{E}_1$ have the same meaning they had in the proof of Theorem 4. In fact, our adversary $\mathcal{A}$ will be the same as well! In other words, he will consider spending one intervention to set $x_1 = 0$ against saving the intervention and making the 0-probability of x_1 equal to $\frac{1}{2} + \delta = 1 - \gamma$. The only difference with the setting of

[12] The *fairness* of a bit c is defined to be $\min(\Pr(c = 0), \Pr(c = 1)) = \frac{1}{2} - bias(c)$.

Theorem 4 is that there $\mathcal{A}$ could "run out" of his b interventions and also minimized a different quantity ($F(p, N, b)$, with different initial conditions), while in our case $\mathcal{A}$ will always use an extra intervention if this pays off. We get the following recurrence (recall, $\gamma = \frac{1}{2} - \delta$, $1 - \gamma = \frac{1}{2} + \delta$):

$$\begin{aligned} B(p,N) &\leq \max_{\substack{p_0 \geq p_1 \\ p_0+p_1=2p}} \min\left[B(p_0,N-1)+1\ ,\ \gamma \cdot B(p_1,N-1) + (1-\gamma)\cdot B(p_0,N-1)\right] && (6) \\ &= \max_{\substack{p_0 \geq p_1 \\ p_0+p_1=2p}} \left(B(p_0,N-1) + \min\left[\,1\ ,\ \gamma \cdot \{B(p_1,N-1) - B(p_0,N-1)\}\,\right]\ \right) && (7) \end{aligned}$$

Substituting as before $p_0 = p(1+\beta)$ and $p_1 = p(1-\beta)$, where $0 \leq \beta \leq \min(1, 1/p - 1) \leq 1$ and using our inductive assumption on $(N-1)$, we get

$$B(p,N) \leq \max_{0 \leq \beta \leq 1} \left(h(p(1+\beta)) + \min\left[\,1\ ,\ \gamma \cdot \{h(p(1-\beta)) - h(p(1+\beta))\}\,\right]\ \right) \stackrel{?}{\leq} h(p) \qquad (8)$$

Recalling the definition of h, it thus suffices to show that

$$\max_{0 \leq \beta \leq 1} \left(\log_z \frac{1}{p(1+\beta)} + \min\left[\,1\ ,\ \gamma \cdot \log_z \frac{1+\beta}{1-\beta}\,\right] \right) \leq \log_z \frac{1}{p}$$

It will now be convenient to make a change of variable and let $\beta = \frac{c-1}{c+1}$ for some $c \geq 1$ (this is always possible because $0 \leq \beta \leq 1$). Noticing that $\log_z(1/p)$ cancels, $1-\beta = 2/(c+1)$, $1+\beta = 2c/(c+1)$, $(1+\beta)/(1-\beta) = c$ and $1 = \log_z z$, we get that it suffices to show that

$$\begin{aligned} \max_{c \geq 1} \left(\log_z \frac{c+1}{2c} + \min\left[\ \log_z z\ ,\ \gamma \cdot \log_z c\ \right] \right) &\leq 0 \quad \Longleftrightarrow \\ \max_{c \geq 1} \left(\frac{c+1}{2c} \cdot \min\left[z, c^{\gamma}\right] \right) &\leq 1 \end{aligned}$$

We now make the final change of variable, letting $c = w^{1/\gamma}$. Then it suffices to show that

$$\max_{w \geq 1} \left(\frac{w^{1/\gamma}+1}{2w^{1/\gamma}} \cdot \min\left[z, w\right] \right) \leq 1 \qquad (9)$$

To show the last equation, we consider two cases.

- Case 1. Assume $w \leq z$. Then $\min[z, w] = w$ and it suffices to show $w^{1/\gamma} + 1 \leq 2w^{1/\gamma - 1}$, which follows from Lemma 2 since $1 \leq w \leq z$ by our assumption.
- Case 2. Assume $w \geq z$. Then $\min[z, w] = z$ and it suffices to show $(w^{1/\gamma}+1)z \leq 2w^{1/\gamma}$, which is the same as $w^{1/\gamma} \geq z/(2-z)$. But since $z = z_\gamma$ is the solution to Equation (4), it is easy to see that $z/(2-z) = z^{1/\gamma}$, so it suffices to show $w^{1/\gamma} \geq z^{1/\gamma}$, which is the same as our assumption $w \geq z$. ∎

C Impossibility of Black-Box Transformations

In this section we formally define "black-box transformations" from statically to adaptively secure coin-flipping protocols, relate them to our imperfect source, and show that this approach does not allow us to break the $b = O(\sqrt{n})$ barrier for adaptive protocols.

Black-Box Transformations. Assume we are given a protocol Π which is known be "very good" against *static* adversaries. We ask the question if it is possible to transform Π in a "black-box" way so as to obtain a "somewhat good" *adaptively secure* protocol Φ. To capture the intuition that we are really obtaining Φ from Π, we do not allow the player to send any messages outside those they send in Π, but allow them to run Π sequentially as many times as they wish. Of course, one might try to let the players run some sub-protocols in between running Π, but then it is very hard to say that we are really using Π and do not, say, run a brand new protocol in the middle and ignore everything that happens in Π. Thus, Φ can run Π any number of times times N, get some coins $x = x_1 \dots x_N$, and then can apply any deterministic function $f : \{0,1\}^N \to \{0,1\}$ to extract the final coin. This leads us to the following natural definition.

Definition 3 *Let N be any integer and $f : \{0,1\}^N \to \{0,1\}$ be any function. We let $\Phi(N, f, \Pi)$ (often we omit Π) be the protocol where players sequentially run N times the protocol Π, obtain coins $x_1 \dots x_N$, and output $f(x_1 \dots x_N)$ as the resulting coin. The class $\{\Phi(N, f, \Pi) \mid N \geq 1, f : \{0,1\}^N \to \{0,1\}\}$ is called the class of* black-box transformations *of* Π.

The (False) Hope. The intuitive reason why black-box transformations look very promising is the following. Assume that Π is $b(n)$-resilient and we wish to construct an adaptively $b(n)$-resilient $\Phi(N, f, \Pi)$. Ignoring the question of efficiency, we can make N arbitrarily large compared to $b(n)$ and n (e.g., 2^{2^n} if we so wish). Assume now $\mathcal{A}$ can adaptively corrupt up to $b(n)$ players. Let us take the worst case, and assume that whenever $\mathcal{A}$ corrupts even a single player in the middle of Π_i (the i-th run of Π), he controls x_i. *But this can happen at most $b(n) \ll N$ times.* And if $\mathcal{A}$ does not corrupt a player in the middle of Π, we know from the static security of Π that the coin is at least slightly random. Thus, at most $b \ll N$ of the x_i's are really biased, the remaining $(N - b)$ of the x_i's are at least slightly random (maybe even almost random). So it seems like there should not be a big problem to design a function f that would be able to "ignore" this "tiny" number b of "fixed" bits, and extract just a good random bit from the remaining $(N - b)$ "good" bits. Of course, we just asked the question if a good bit can be extracted from (δ, b, N)-BCL source is possible (where δ depends on the properties of Π)! Unfortunately, we showed in Theorem 3 some strong negative results concerning the bit extraction from our source. In particular, we will show that one cannot beat the simple majority protocol using the above approach.

Adaptive Adversary for a Black-Box Transformation. The definition of a black-box transformation views the protocol Π as "one piece" that is simply being run several times. Even though given a particular Π (and N and f), we will end up with a particular protocol $\Phi(N, f, \Pi)$ and can talk about it being adaptively $b(n)$-resilient, it is more natural to let the adaptive adversary $\mathcal{A}$ for Φ perform "meta-operations" on the entire run of each Π (consistent with the static security of Π). Namely, (1) $\mathcal{A}$ can decide not to corrupt any players during the run of Π, and then the bias of the resulting coin is what is achieved by the security of Π, or (2) $\mathcal{A}$ can decide to corrupt one or more player during the run of Π, and then we do not know anything about the resulting coin, and, therefore, have to assume the worst (i.e., $\mathcal{A}$ can fix the coin). We make this more formal.

Assume that given a fixed set B of faulty players, Π produces a at most a $\Delta_\Pi(B)$-biased coin for any static adversary who corrupts B at the beginning, and let $\Delta_\Pi(b) = \max_{|B|=b} \Delta_\Pi(B)$ be the best bias that a b-bounded static adversary can achieve. Let us denote by Π_i the i-th run of Π, and by x_i the resulting coin. As before, $\mathcal{A}$ is called b-bounded if he corrupts at most b players overall. However, now we assume that $\mathcal{A}$ (the adversary for $\Phi(N, f, \Pi)$) has the following capabilities:

(A) If $\mathcal{A}$ decides to corrupt at least one new player during the execution of Π_i, he can set the resulting coin x_i to any value.

(B) If at the beginning of Π_i the set of corrupted players is B and $\mathcal{A}$ decides not to corrupt new players during Π, the resulting coin x_i is at most $\Delta_\Pi(B)$-biased, but $\mathcal{A}$ can set the probability of $x_i = 0$ anywhere in the interval $[\frac{1}{2} - \Delta_\Pi(B), \frac{1}{2} + \Delta_\Pi(B)]$.

We justify assumptions (A) and (B) in two ways. First of all, we are talking about *black-box reductions*. In other words, we do not know and do not want to assume anything more about Π than what is given to us by the function $\Delta_\Pi(B)$. Thus, if $\mathcal{A}$ does not corrupt new players inside Π_i, we know that $\Pr(x_i = 0) \in [\frac{1}{2} - \Delta_\Pi(B), \frac{1}{2} + \Delta_\Pi(B)]$, but we cannot assume anything more, so we assume that $\mathcal{A}$ can set $\Pr(x_i = 0)$ anywhere in this interval. Similarly, once $\mathcal{A}$ corrupts a player inside Π_i, nothing can be said about the behavior of the resulting coin, so we again have to assume the worst case.

The other justification comes from the fact that all best non-adaptively secure coin-flipping protocols (e.g., [AN93, ORV94, RZ98, F99]) essentially satisfy both of these assumptions.[13] Assumption (A) because they always elect the leader, so corrupting the leader allows the adversary to control the coin. And assumption (B) because these protocols are actually symmetric in 0 and 1 and by making faulty players be "less and less faulty", they can indeed achieve essentially any probability inside the specified interval.

Main Result. Our main result in coin-flipping is the following theorem, which states that using black-box reductions one cannot significantly beat the simple majority protocol, giving further support to Conjecture 1.

Theorem 8 *For any family of coin-flipping protocols Π, there is no black-box transformation resulting in an adaptively $\omega(\sqrt{n})$-resilient family of protocols $\Phi(N, f, \Pi)$.*

Reduction to Imperfect Random Sources. We reduce the proof of Theorem 8 to the analysis of our Bias-Control Limited source. Namely, assume $\Phi(N, f, \Pi)$ is adaptively $2b(n)$-resilient. We construct the following $2b(n)$-bounded adversary for Φ satisfying properties (A) and (B). Let $b = b(n)$, $\delta = \Delta_\Pi(b)$ and let B be the set of players of cardinality b achieving $\Delta_\Pi(B) = \Delta_\Pi(b) = \delta$. Before Π_1 starts, $\mathcal{A}$ corrupts all the players in B. Therefore, from now on in each of the N invocations of Π, $\mathcal{A}$ can set the 0-probability of x_i anywhere in at least the interval $[\frac{1}{2} - \delta, \frac{1}{2} + \delta]$. As $\mathcal{A}$ will later corrupt more players, this interval can only expand, but our particular $\mathcal{A}$ will not use it. If $\mathcal{A}$ decides to follow rule (A), he will corrupt a single player and set the corresponding bit x_i to the value he wants. Therefore, since Φ claims to be $2b$-resilient, $\mathcal{A}$ can use rule (A) exactly b times. Hence, now we *exactly reduced the possible behavior of $\mathcal{A}$ to an arbitrary (δ, b, N)-BCL source*.

Tracing back to the adaptive coin-flipping, once we decided to achieve adaptive $2b(n)$-resilience, there is fundamental limitation on how fair we can make the resulting coin, irrespective of how many times we run the black-box protocol Π. In other words, our informal intuition was wrong, when we claimed that we should be able to "overcome" any number b of completely biased bits when having an overwhelming majority of $(N - b)$ slightly random bits.

We can now apply Theorem 3 to establish the impossibility of black-box reductions given by Theorem 8. Recall that we concluded that it is impossible to obtain a weakly adaptively $2b$-resilient $\Phi(b, N, \Pi)$ if it is impossible to extract a slightly random bit from (δ, b, N)-BCL source, where $\delta = \Delta_\Pi(b)$. From the upper bound of Ben-Or and Linial [BL90] that we mentioned in Section 4, we know that $\Delta_\Pi(b) \geq \Omega(b/n)$. Thus $b\delta = \Omega(b^2/n)$. By Theorem 3, it is impossible to extract a slightly random bit whenever $b^2/n = \omega(1)$, i.e. $b = \omega(\sqrt{n})$, establishing Theorem 8.[14]

D Proof of Theorem 5

Both statements are proven by induction on N in a very similar manner. After establishing the base $N = 1$, we consider a recursive adversary who will either use an intervention on X_1 to force the most desirable value on the X_1 (but loose an intervention) and then behaves optimally, or will try to bias X_1 by δ towards the outcome that it prefers (without loosing an intervention) and then behaves optimally. By choosing the best of the above options and using the inductive assumption (since we reduced N in both cases), we will be able to complete the induction.

[13] In fact, it is easy to check that our main Theorem 8 holds on a "concrete level" if we replace Π with any of these protocols.

[14] If we want to extract *almost* random bit, it is impossible to do it if $b = \Omega(\sqrt{n})$.

We illustrate this first for $F(p, N, b)$. The statement is true for $\delta = 0$ or $b = 0$, since $F(\cdot, \cdot, \cdot) \leq 1 \leq 1/p$, so assume $\delta > 0$ and $b \geq 1$. Define $g(p, b) = (1 - \delta)^b/p$. We need to show that $F(p, N, b) \leq g(p, b)$ for any $N \geq 1$, $1 \leq b \leq N$ and $0 \leq p \leq 1$. We prove this by induction on N. For $N = 1$, $F(0, 1, b) = 1 < \infty = g(0, b)$, and $F(p, 1, b) = 0 \leq g(p, b)$ for $p > 0$ (here we used $b \geq 1$). Assume now the claim is true for $(N - 1)$ and we want to show it for N.

Let $D_1 = \{\alpha_1 \dots \alpha_t\}$ be the distribution on X_1 (thus, $\sum_i \alpha_i = 1$), which we can assume is the supported on the set $\{1 \dots t\}$. And let us take any p-sparse $\mathcal{E}$ given by a function e. For $1 \leq i \leq t$, let $e_i(X_2, \dots, X_n)$ be the restriction of e when $X_1 = i$. Each e_i defines a p_i-sparse event $\mathcal{E}_i$, which satisfy $\sum_i \alpha_i p_i = p$. Without loss of generality assume $p_1 \geq \dots \geq p_t$ (i.e., rename the α_i's to satisfy this). First, if $\alpha_1 > 1 - \delta$ (in particular, when $t = 1$), then we are done. Indeed, in this case our adversary can fix $X_1 = 1$ without using an intervention, reducing the analysis to that of a p_1-sparse event $\mathcal{E}_1$ with the same b. Since $p_1 \geq p$ and using the induction, we get $F(p, N, b) \leq F(p_1, N - 1, b) \leq (1 - \delta)^b/p_1 \leq (1 - \delta)^b/p$. Thus, assume $\alpha_1 \leq 1 - \delta$. Then there exists an index $k > 1$ such that $\alpha_k + \dots + \alpha_t \geq \delta$ but $\alpha_{k+1} + \dots + \alpha_t < \delta$. Then a particular distribution $D_1' = (\alpha_1' \dots \alpha_t')$ of statistical distance δ from D_1 is given by: $\alpha_1' = \alpha_1 + \delta$, $\alpha_i' = \alpha_i$ for $2 \leq i < k$, $\alpha_k' = \alpha_k + \dots + \alpha_t - \delta$ and $\alpha_i' = 0$ for $i > k$. In other words, we increase the "most desirable" (for the adversary) probability of $X_1 = 1$ by δ, an decreased the "least desirable" probabilities $\alpha_k \dots \alpha_t$ by δ (overall).

Now, our particular adversary $\mathcal{A}$ will consider two options and pick the best (using his unbounded computational resources): either he will use an intervention (he can do it since we assumed $b \geq 1$) and fix $X_1 = 1$, reducing the question to that of analyzing the p_1-sparse event $\mathcal{E}_1$ on $(N - 1)$ variables and also reducing b by 1, or he will use rule (B) with the probability distribution D_1' instead of D_1 (leaving the same b). Since $\mathcal{A}$ will pick the best of the two events, and using the inductive assumption, we get

$$F(p, N, b) \leq \min \left[F(p_1, N - 1, b - 1), \sum_{i=1}^{k} \alpha_i' F(p_i, N - 1, b) \right] \leq \min \left[g(p_1, b - 1), \sum_i \alpha_i' g(p_i, b) \right] \stackrel{?}{\leq} g(p, b)$$

Thus, it suffices to show the last inequality. Let $p_i = \beta_i p$, where $\beta_1 \geq \dots \geq \beta_t \geq 0$. Thus, $\sum_{i=1}^t \alpha_i \beta_i = 1$. Recalling now the definition of $g(p, b) = (1 - \delta)^b/p$, we need to show:

$$\min \left[\frac{(1 - \delta)^{b-1}}{\beta_1 p}, \sum_{i=1}^{k} \alpha_i' \cdot \frac{(1 - \delta)^b}{\beta_i p} \right] \stackrel{?}{\leq} \frac{(1 - \delta)^b}{p} \quad \Longleftrightarrow \quad \min \left[\frac{1}{(1 - \delta)\beta_1}, \sum_{i=1}^{k} \frac{\alpha_i'}{\beta_i} \right] \stackrel{?}{\leq} 1$$

Now, we would be done if $\beta_1 \geq 1/(1 - \delta)$, so let us assume that $\beta_1 < 1/(1 - \delta)$. With this in mind, we have to show that $\sum_{i=1}^k \alpha_i'/\beta_i \leq 1$. Unfortunately, we have very little control over α_i' and β_i. But we do know several things. First, α_i' form a distribution, and thus $\sum_{i=1}^k \alpha_i' = 1$. Also, since $\alpha_1' = \alpha_1 + \delta$, we have $\alpha_1' \geq \delta$. Second, recalling the definition of $\{\alpha_i'\}$ from $\{\alpha_i\}$, and using the fact that $\sum_{i=1}^t \alpha_i \beta_i = 1$, we get that $\sum_{i=1}^k \alpha_i' \beta_i = \sum_{i=1}^t \alpha_i \beta_i + \delta(\beta_1 - \beta_k) = 1 + \delta(\beta_1 - \beta_k)$. Finally, we know that $1/(1 - \delta) > \beta_1 \geq \dots \geq \beta_k \geq 0$. Thus, to complete the induction it suffices to show the following technical lemma, which we prove separately:

Lemma 3 *For any $k \geq 2$, any $1/(1 - \delta) > \beta_1 \geq \dots \geq \beta_k \geq 0$, any $\alpha_1' \geq \delta$, any $\alpha_2' \dots, \alpha_k' \geq 0$ satisfying: $\sum \alpha_i' = 1$ and $\sum \alpha_i' \beta_i = 1 + \delta(\beta_1 - \beta_k)$, we have*

$$\sum \frac{\alpha_i'}{\beta_i} \leq 1 \tag{10}$$

Proof: As a sanity check, we notice that we cannot have $\beta_k = 0$ (making the sum on the left equal to infinity). Indeed, $1 + \delta(\beta_1 - \beta_k) = \sum \alpha_i' \beta_i \leq \beta_1 \cdot \sum \alpha_i' = \beta_1 < 1 + \delta\beta_1$ (as $\beta_1 < 1/(1 - \delta)$), implying $\beta_k > 0$. Thus, all the β_i's are strictly positive. Now we show Equation (10) by induction on k.

The base case $k = 2$ is the most technical one to show. We know that $\alpha_1' + \alpha_2' = 1$ and $\alpha_1' \beta_1 + \alpha_2' \beta_2 = 1 + \delta(\beta_1 - \beta_2)$, where $\beta_1 \geq \beta_2$. This system of equations above has a unique solution for α_1' and α_2', unless we have $\beta_1 = \beta_2$. However, in the latter case the system is solvable only if $\beta_1 = \beta_2 = 1$, in which case $\alpha_1'/\beta_1 + \alpha_2'/\beta_2 = \alpha_1' + \alpha_2' = 1$, and we are done. Thus, assume $\beta_1 > \beta_2$. We then get a unique solution

$\alpha'_1 = \delta + (1-\beta_2)/(\beta_1 - \beta_2)$, and $\alpha'_2 = -\delta + (\beta_1 - 1)/(\beta_1 - \beta_2)$. We notice that since we know that $\alpha'_1 \geq \delta$, we get $\beta_2 \leq 1$. Using some simple algebra, we get

$$\frac{\alpha'_1}{\beta_1} + \frac{\alpha'_2}{\beta_2} = \frac{\beta_1 + \beta_2 - 1 - \delta(\beta_1 - \beta_2)}{\beta_1\beta_2} = 1 + \frac{(\beta_1 - 1)(1-\beta_2) - \delta(\beta_1 - \beta_2)}{\beta_1\beta_2} \stackrel{?}{\leq} 1 \tag{11}$$

Let $x = \beta_1 - 1$, $y = 1 - \beta_2$. As $1/(1-\delta) \geq \beta_1 \geq 1 \geq \beta_2 \geq 0$, we get that $x \in [0, \frac{\delta}{1-\delta}]$ and $y \in [0, 1]$. To show Equation (11), we need to show $(\beta_1 - 1)(1 - \beta_2) \leq \delta \cdot (\beta_1 - \beta_2)$. Noticing that $x + y = \beta_1 - \beta_2$, it remains to show that $xy \leq \delta(x + y)$, or $\delta/x + \delta/y \geq 1$ (in the latter part we used that $x, y \geq 0$). But using our upper bounds $x \leq \delta/(1-\delta)$ and $y \leq 1$, we indeed get $\delta/x + \delta/y \geq (1-\delta) + \delta = 1$.

We can now establish the inductive step. Assume $k \geq 3$ and the statement is true for $(k-1)$. Take any $1 < i < k$ (for example $i = k - 1$). We will now define a distribution $(\alpha''_1, \ldots, \alpha''_{k-1})$ on $(k-1)$ elements. This will be done by "removing" α'_i from our original distribution $\alpha'_1, \ldots, \alpha'_k$, and distributing its mass to the "neighbors" α'_{i-1} and α'_{i+1}. Since the are many way to distribute α'_i, we will choose a way that leaves the same β's (so that we do not violate any constraints on the β's and can use induction). More precisely, we let $\beta'_1 = \beta_1, \ldots, \beta'_{i-1} = \beta_{i-1},\ \beta'_i = \beta_{i+1}, \ldots, \beta'_{k-1} = \beta_k$ (i.e., "remove" β_i), and $\alpha''_1 = \alpha'_1, \ldots, \alpha''_{i-2} = \alpha'_{i-2},\ \alpha''_{i-1} = \alpha_{i-1} + x,\ \alpha''_i = \alpha_{i+1} + y,\ \alpha''_{i+1} = \alpha_{i+2}, \ldots, \alpha''_{k-1} = \alpha'_k$, where we will choose $x \geq 0$ and $y \geq 0$ satisfying the following two conditions:

$$\begin{cases} x + y = \alpha'_i \\ x \cdot \beta_{i-1} + y \cdot \beta_{i+1} = \alpha'_i \cdot \beta_i \end{cases}$$

We will elaborate on assigning such x and y in a second, but now we notice that the first condition above implies that $\sum_{j=1}^{k-1} \alpha''_j = \sum_{j \neq i} \alpha'_j + (x + y) = \sum_{j=1}^{k} \alpha'_j = 1$, while the second condition implies

$$\begin{aligned} \sum_{j=1}^{k-1} \alpha''_j \beta'_j &= \sum_{j=1}^{i-2} \alpha'_j\beta_j + (\alpha'_{i-1} + x)\beta_{i-1} + (\alpha'_{i+1} + y)\beta_{i+1} + \sum_{j=i+2}^{k} \alpha'_j\beta_j \\ &= \sum_{j \neq i} \alpha'_j\beta_j + x\beta_{i-1} + y\beta_{i+1} = \sum_{j=1}^{k} \alpha'_j\beta_j = 1 + \delta(\beta_1 - \beta_k) \\ &= 1 + \delta(\beta'_1 - \beta'_{k-1}) \end{aligned}$$

Also, since we will assign $x \geq 0$, we will get (even when $i = 2$) that $\alpha''_1 \geq \alpha'_1 \geq \delta$. Thus, $\alpha''_1, \ldots, \alpha''_{k-1}$ and $\beta'_1, \ldots, \beta'_{k-1}$ will satisfy all the preconditions of our statement for $(k-1)$, and hence our inductive assumption will tell us that $\sum_{j=1}^{k-1} \alpha''_i/\beta'_i \leq 1$. To complete the induction, we need to show two things: (a) how to assign the needed x and y, and (b) $\sum_{j=1}^{k} \alpha'_j/\beta_j \leq \sum_{j=1}^{k-1} \alpha''_j/\beta'_j$ (and we know that the latter is at most 1). From the definition of α''_j and β'_j, it is easy to see that the latter inequality is equivalent to

$$\frac{x}{\beta_{i-1}} + \frac{y}{\beta_{i+1}} \stackrel{?}{\geq} \frac{\alpha'_i}{\beta_i} \tag{12}$$

We now consider two cases. First, if $\beta_{i-1} = \beta_{i+1}$, then we in fact have $\beta_{i-1} = \beta_i = \beta_{i+1} = \beta$. In this case the above system for x and y has infinitely many solutions x and y (as long as $x + y = \alpha'_i$). Take any such solution where $x \geq 0$ and $y \geq 0$. To see that Equation (12) indeed holds, we see that $x/\beta_{i-1} + y/\beta_{i+1} = (x+y)/\beta = \alpha'_i/\beta_i$.

Now, we consider the interesting case $\beta_{i-1} > \beta_{i+1}$. Then the above system on x and y has a unique solution

$$\begin{cases} x = \alpha'_i \cdot (\beta_i - \beta_{i+1})/(\beta_{i-1} - \beta_{i+1}) \\ y = \alpha'_i \cdot (\beta_{i-1} - \beta_i)/(\beta_{i-1} - \beta_{i+1}) \end{cases}$$

Notice, we indeed have $x, y \geq 0$. Then the needed Equation (12) becomes

$$\begin{aligned} \frac{\alpha'_i}{\beta_{i-1} - \beta_{i+1}} \cdot \left[\frac{\beta_i - \beta_{i+1}}{\beta_{i-1}} + \frac{\beta_{i-1} - \beta_i}{\beta_{i+1}}\right] &\stackrel{?}{\geq} \frac{\alpha'_i}{\beta_i} \quad \Longleftrightarrow \\ \frac{\beta_{i+1}}{\beta_i} + \frac{\beta_i}{\beta_{i-1}} + \frac{\beta_{i-1}}{\beta_{i+1}} &\stackrel{?}{\geq} \frac{\beta_{i+1}}{\beta_{i-1}} + \frac{\beta_{i-1}}{\beta_i} + \frac{\beta_i}{\beta_{i+1}} \end{aligned}$$

However, the latter equation is easily seen to be true for any $\beta_{i-1} \geq \beta_i \geq \beta_{i+1} > 0$ (for example, multiplying both sides by $\beta_{i-1}\beta_i\beta_{i+1} > 0$ makes it equivalent to $(\beta_{i-1} - \beta_i)(\beta_i - \beta_{i+1})(\beta_{i-1} - \beta_{i+1}) \geq 0$, which is true). This established Equation (12) and completes the proof of the technical lemma. ■

We now establish the bound on $B(p, N)$ in a similar manner to that for $F(p, N, b)$ above. Luckily, a lot of technical machinery and notation has been developed already.

Let $z = 1/(1-\delta)$ and assume $\delta > 0$ (otherwise there is noting to prove), so that $z > 1$. Define $h(p) = \log_z(1/p)$. We need to show that $B(p, N) \leq h(p)$ for any $N \geq 1$ and $0 \leq p \leq 1$. We prove this by induction on N. For $N = 1$, we have: $B(0, 1) = \infty = h(0)$; if $0 < p \leq 1 - \delta$, then $B(p, 1) = 1 \leq \log_z(1/p) = h(p)$; and if $1 - \delta \leq p \leq 1$, then $B(p, 1) = 0 \leq h(p)$ (since then we can force $\mathcal{E}$ by applying rule (B)). Assume now the claim is true for $(N-1)$ and we want to show it for N.

We assume an *identical* notation with the previous proof for $F(p, N, b)$. Namely, denote by $D_1 = \{\alpha_1 \dots \alpha_t\}$ be the distribution on X_1, take any p-sparse $\mathcal{E}$, define p_i-sparse "projection" events $\mathcal{E}_i$, which satisfy $\sum_i \alpha_i p_i = p$, assume (without loss of generality) that $p_1 \geq \ldots \geq p_t$. As before, if $\alpha_1 > 1 - \delta$ (in particular, when $t = 1$), then we are done. Indeed, in this case our adversary can fix $X_1 = 1$ without using an intervention, reducing the analysis to that of a p_1-sparse event $\mathcal{E}_1$. Since $p_1 \geq p$ and using the induction, we get $B(p, N) \leq B(p_1, N-1) \leq \log_z(1/p_1) \leq \log_z(1/p)$. Thus, assume $\alpha_1 \leq 1 - \delta$. As before, take index $k > 1$ such that $\alpha_k + \ldots + \alpha_t \geq \delta$ but $\alpha_{k+1} + \ldots + \alpha_t < \delta$, and define the distribution $D_1' = (\alpha_1' \dots \alpha_t')$ of statistical distance δ from D_1 as before: $\alpha_1' = \alpha_1 + \delta$, $\alpha_i' = \alpha_i$ for $2 \leq i < k$, $\alpha_k' = \alpha_k + \ldots + \alpha_t - \delta$ and $\alpha_i' = 0$ for $i > k$. Again, our particular adversary $\mathcal{A}$ will consider two options and pick the best (using his unbounded computational resources): either he will use an intervention and fix $X_1 = 1$, reducing the question to that of analyzing the p_1-sparse event $\mathcal{E}_1$ on $(N-1)$, or he will use rule (B) (and save an intervention) with the probability distribution D_1' instead of D_1. Since $\mathcal{A}$ will pick the best of the two events, and using the inductive assumption, we get

$$B(p, N) \leq \min\left[1 + B(p_1, N-1),\ \sum_{i=1}^{k} \alpha_i' B(p_i, N-1)\right] \leq \min\left[1 + h(p_1),\ \sum_i \alpha_i' h(p_i)\right] \stackrel{?}{\leq} h(p)$$

Thus, it suffices to show the last inequality. Following again the same notation as before, let $p_i = \beta_i p$, where $\beta_1 \geq \ldots \geq \beta_t \geq 0$. Thus, $\sum_{i=1}^{t} \alpha_i \beta_i = 1$. Recalling now the definition of $h(p) = \log_z(1/p)$, we need to show:

$$\min\left[1 + \log_z\left(\frac{1}{\beta_1 p}\right), \sum_{i=1}^{k} \alpha_i' \cdot \log_z\left(\frac{1}{\beta_i p}\right)\right] \stackrel{?}{\leq} \log_z\left(\frac{1}{p}\right) \iff \min\left[\log_z\left(\frac{z}{\beta_1}\right), \sum_{i=1}^{k} \alpha_i' \log_z\left(\frac{1}{\beta_i}\right)\right] \stackrel{?}{\leq} 0$$

(in the last equivalence we used $\sum_{i=1}^{k} \alpha_i' = 1$). Now, if we convert the weighted sum of $\log$'s into a $\log$ of a product of exponents, write $0 = \log_z 1$, and then get rid of $\log_z$, we only need to show (recall, $z = 1/(1-\delta)$):

$$\min\left[\frac{1}{(1-\delta)\beta_1},\ \prod_{i=1}^{k}\left(\frac{1}{\beta_i}\right)^{\alpha_i'}\right] \stackrel{?}{\leq} 1$$

Now, we would be done if $\beta_1 \geq 1/(1-\delta)$, so let us assume that $\beta_1 < 1/(1-\delta)$. With this in mind, we have to show that $\prod_{i=1}^{k}(1/\beta_i)^{\alpha_i'} \leq 1$. However, by Cauchy-Schwartz inequality and since $\sum_{i=1}^{k} \alpha_i' = 1$, we have

$$\prod_{i=1}^{k}\left(\frac{1}{\beta_i}\right)^{\alpha_i'} \leq \left(\frac{\sum_{i=1}^{k} \alpha_i'/\beta_i}{\sum_{i=1}^{k} \alpha_i'}\right)^{\sum_{i=1}^{k} \alpha_i'} = \sum_{i=1}^{k} \frac{\alpha_i'}{\beta_i}$$

Hence, it suffices to show that $\sum_{i=1}^{k} \frac{\alpha_i'}{\beta_i} \leq 1$. But this is exactly the statement of Lemma 3! Indeed, we used the same notation for α_i' and β_i and have the same *identical* set on constraints on them. Hence, another application of Lemma 3 completes the induction and the overall proof of Theorem 5.