2-Source Dispersers for $n^{o(1)}$ Entropy, and Ramsey Graphs Beating the Frankl-Wilson Construction

Boaz Barak* Anup Rao † Ronen Shaltiel‡ Avi Wigderson§

November 3, 2005

Abstract

The main result of this paper is an explicit disperser for two independent sources on n bits, each of entropy $k = n^{o(1)}$. Put differently, setting $N = 2^n$ and $K = 2^k$, we construct explicit $N \times N$ Boolean matrices for which no $K \times K$ submatrix is monochromatic. Viewed as adjacency matrices of bipartite graphs, this gives an explicit construction of K-Ramsey *bipartite* graphs of size N.

This greatly improves the previous the previous bound of $k = o(n)$ of Barak, Kindler, Shaltiel, Sudakov and Wigderson [BKS$^+$05]. It also significantly improves the 25-year record of $k = \tilde{O}(\sqrt{n})$ on the very special case of Ramsey graphs, due to Frankl and Wilson [FW81].

The construction uses (besides "classical" extractor ideas) almost all of the machinery developed in the last couple of years for extraction from independent sources, including:

- Bourgain's extractor for 2 independent sources of some entropy rate $< 1/2$ [Bou05]
- Raz' extractor for 2 independent sources, one of which has any entropy rate $> 1/2$ [Raz05]
- Rao's extractor for 2 independent block-sources of entropy $n^{\Omega(1)}$ [Rao05]
- The "Challenge-Response" mechanism for detecting "entropy concentration" of [BKS$^+$05].

The main novelty comes in a bootstrap procedure which allows the Challenge-Response mechanism of [BKS$^+$05] to be used with sources of less and less entropy, using recursive calls to itself. Subtleties arise since the success of this mechanism depends on restricting the given sources, and so recursion constantly changes the original sources. These are resolved via a new construct, in between a disperser and an extractor, which behaves like an extractor on sufficiently large subsources of the given ones.

Keywords: Extractors, Dispersers, Ramsey Graphs

*Department of Computer Science, Princeton University. boaz@cs.princeton.edu

†Department of Computer Science, University of Texas at Austin. arao@cs.utexas.edu. Most of this work was done while the author was visiting Princeton University and the Institute for Advanced Study. Supported in part by an MCD fellowship from UT Austin and NSF Grant CCR-0310960.

‡Ronen Shaltiel, University of Haifa, Mount Carmel, Haifa, Israel. ronen@cs.haifa.ac.il

§Institute for Advanced Study, Princeton, New Jersey. avi@math.ias.edu

Contents

1 Introduction

This paper deals with randomness extraction from weak random sources. Here a weak random source is a distribution which contains some entropy. The extraction task is to design efficient algorithms (called *extractors*) to convert this entropy into useful form, namely a sequence of independent unbiased bits. Beyond the obvious motivations (potential use of physical sources in pseudorandom generators and in derandomization), extractors have found applications in a variety of areas in theoretical computer science where randomness does not seem an issue, such as in efficient constructions of communication networks [WZ99, CRVW02], error correcting codes [TSZ04, Gur03], data structures [MNSW98] and more.

Most work in this subject over the last 20 years has focused on what is now called *seeded* extraction, in which the extractor is given as input not only the (sample from the) defective random source, but also a few truly random bits (called the *seed*). A comprehensive survey of much of this body of work is [Sha02].

Another direction, which has been mostly dormant till about two years ago, is (seedless, deterministic) extraction from few *independent* weak sources. This kind of extraction is important in several applications where it is unrealistic to have a short random seed or deterministically enumerate over its possible values. However, it is easily shown to be impossible when only one weak source is available. When at least 2 independent sources are available extraction becomes feasible. The 2-source case is the one we will focus on in this work.

The rest of the introduction is structured as follows. We'll start by describing our main result in the context of Ramsey graphs. We then move to the context of extractors and dispersers, describing the relevant background and stating our result in this language. Then we give an overview of the construction of our dispersers, leading to them in describing the main building blocks we construct first. This will be done in a high level form which will hopefully aid in reading the technical parts (which we have delayed to the appendix). As the construction is quite complex and its analysis quite subtle, we try to abstract away many of the technical difficulties so that the main ideas, structure and tools used are highlighted. For that reason we also often state definitions and theorems somewhat informally. However we give pointers to the places in the technical parts which elaborate and make everything precise and formal.

1.1 Ramsey Graphs

A graph is K-Ramsey if it contains no clique or independent set of size K. In 1947 Erdős published his paper inaugurating the *Probabilistic Method* with a few examples, including a proof that *most* graphs on $N = 2^n$ vertices are $2n$-Ramsey. The quest for constructing such graphs explicitly has existed ever since and lead to some beautiful mathematics.

The best record to date was obtained in 1981 by Frankl and Wilson [FW81], who used intersection theorems for set systems to construct N-vertex graphs which are $2^{\sqrt{n \log n}}$-Ramsey. This bound was matched by Alon [Alo98] using the *Polynomial Method*, by Grolmusz [Gro00] using low rank matrices over rings, and also by Barak, Impagliazzo and Wigderson [BIW04] boosting Abbot's method with almost k-wise independent random variables (a construction that was independently discovered by others as well). Remarkably all these different approaches got stuck at essentially the same bound.

The analogous question for bipartite graphs seemed much harder. Here a graph on two sets of N vertices is K-Ramsey if it has no $K \times K$ complete or empty bipartite subgraph. While Erdős' result on the abundance of $2n$-Ramsey graphs holds as is for bipartite graphs, the best explicit construction until recently was $2^{n/2}$-Ramsey, using the Hadamard matrix. This was improved last year, first to $o(2^{n/2})$ by Pudlak and Rődl [PR04] and then to $2^{o(n)}$ by Barak, Kindler, Shaltiel, Sudakov and Wigderson [BKS$^+$05].

It is convenient to view such graphs as functions $f : (\{0,1\}^n)2 \to \{0,1\}$, which allows for a formal definition of *explicitness*: we simply demand that the function f is computable in polynomial time. Most of the constructions mentioned above are explicit in this sense.[1]

[1]The Abbot's product based Ramsey-graph construction of [BIW04] and the bipartite Ramsey construction of [PR04] only

Our main result (stated informally) significantly improves the bounds in both the bipartite and non-bipartite settings:

Main Theorem. *For every N we construct explicit bipartite graphs which are $2^{n^{o(1)}}$-Ramsey. A standard transformation of these graphs also yields explicit ordinary Ramsey Graphs with the same parameters.*

1.2 Extractors and Dispersers from independent sources

This subsection will serve simultaneously as a brief review of past relevant work (with the goal of putting this paper in proper context), as well as for describing some of the tools we'll use. We start with the basic definitions of k-sources by Nisan and Zuckerman [NZ93] and of extractors and dispersers for independent sources by Santha and Vazirani [SV86].

Definition ([NZ93])**.** A distribution X on strings (their length will typically be clear from the notation) is a k-source if its entropy is at least k.[2]

Definition ([SV86])**.** A function $f : (\{0,1\}^n)^c \to \{0,1\}^m$ is a c-source *extractor* for entropy k if for every family of c independent k-sources $X_1, X_2, \cdots, X_c$, the output $f(X_1, \cdots, X_c)$ is a (nearly)[3] uniformly distributed string of m bits. f is a *disperser* for the same parameters if the output is simply required to have a full support.

In this language, Erdős' theorem says that most functions $f : (\{0,1\}^n)2 \to \{0,1\}$ are dispersers for entropy $1 + \log n$ (treating f as the characteristic function for the set of edges of the graph). The proof easily extends to show that indeed most such functions are in fact extractors. This naturally challenges us to find explicit functions f that are 2-source extractors.

Until one year ago, essentially the only known explicit construction was the Hadamard extractor Had defined by $\mathsf{Had}(x,y) = \langle x, y\rangle (\mod 2)$. It is an extractor for entropy $k > n/2$ as observed by Chor and Goldreich [CG88] and can be extended to give $m = \Omega(n)$ output bits as observed by Vazirani [Vaz85]. Over 20 years later, a recent breakthrough of Bourgain [Bou05] broke this "1/2 barrier" and can handle 2 sources of entropy $.4999n$, again with linear output length $m = \Omega(n)$. This seemingly minor improvement will be crucial for our work!

Theorem ([Bou05])**.** *There exists an explicit 2-source extractor $f : (\{0,1\}^n)2 \to \{0,1\}^m$ for entropy $.4999n$ and $m = \Omega(n)$.*

No better bounds are known for extractors. Now we turn our attention to dispersers. It turned out that progress for building good 2-source dispersers came via progress on extractors for more than 2 sources, all happening in fast pace in the last 2 years. The seminal paper of Bourgain, Katz and Tao [BKT04] proved the so-called "sum-product theorem" in prime fields, a result in arithmetic combinatorics, which has already found applications in diverse areas of mathematics, including analysis, number theory, group theory and ... extractor theory. Their work implicitly contained dispersers for $c = O(\log(n/k))$ independent sources of entropy k (with output $m = \Omega(k)$). The use of the "sum-product" theorem was then extended by Barak et. al. [BIW04] to give extractors with similar parameters. Note that for linear entropy $k = \Omega(n)$, the number of sources needed for extraction c is a constant!

Relaxing the independence assumptions via the idea of repeated condensing, allowed the reduction of the number of independent sources to $c = 3$, for extraction from sources of any linear entropy $k = \Omega(n)$, by Barak et. al. [BKS+05] and independently by Raz [Raz05].

satisfy a weaker notion of explicitness.

[2]For the purposes of this informal introduction it will not matter what definition we take for entropy. We will use min-entropy, but it is no loss of generality to imagine that X is uniformly distributed over some (unknown) set of size 2^k.

[3]This error, or distance to uniformity, is an extremely important parameter, but will be ignored in the introduction.

For 2 sources Barak et. al. [BKS+05] were able to construct dispersers for sources of entropy $o(n)$. To do this, they first showed that if the sources have extra structure (*block-source* structure, defined below), even extraction is possible from 2 sources. The notion of block-sources, capturing "semi independence" of parts of the source, was introduced by Chor and Goldreich [CG88]. It has been fundamental in the development of seeded extractors and as we shall see, is essential for us as well.

Definition ([CG88]). A distribution $X = X_1, \cdots X_c$ is a c-block-source of (block) entropy k if every block X_i has entropy k even conditioned on fixing the previous blocks $X_1, \cdots, X_{i-1}$ to arbitrary constants.

This definition allowed Barak et. al. [BKS+05] to show that their extractor for 4 independent sources, actually performs as well with only 2 independent sources, as long as both are 2-block-sources.

Theorem ([BKS+05]). *There exists an explicit extractor $f : (\{0,1\}^n)2 \to \{0,1\}$ for 2 independent 2-block-sources with entropy $o(n)$.*

There is no reason to assume that the given sources are block-sources, but it is natural to try and reduce to this case. This approach has been one of the most successful in the extractor literature. Namely try to partition a source X into two blocks $X = X_1, X_2$ such that X_1, X_2 form a 2-block-source. Barak et. al. introduced a new technique to do this reduction called the *Challenge-Response mechanism*, which is crucial for this paper. This method gives a way to "find" how entropy is distributed in a source X, to allow such a partition. This method succeeds only with small probability, dashing the hope for an extractor, but still yielding a disperser.

Theorem ([BKS+05]). *There exists an explicit 2-source disperser $f : (\{0,1\}^n)2 \to \{0,1\}^m$ for entropy $o(n)$.*

Reducing the entropy requirement of the above 2-source disperser, which is what we achieve in this paper, again needed progress on achieving a similar reduction for extractors with more independent sources. A few months ago Rao [Rao05] was able to significantly improve all the above results for $c \geq 3$ sources. Interestingly, his techniques do not use arithmetic combinatorics, which seemed essential to all the papers above. He improves the results of Barak et. al. [BIW04] to give $c = O((\log n)/(\log k))$-source extractors for entropy k. Note that now the number c of sources needed for extraction is constant, even when the entropy is as small as n^δ for any constant δ!

Again, when the input sources are block-sources with sufficiently many blocks, Rao proves that 2 independent sources suffice (though this result does rely on arithmetic combinatorics).

Theorem ([Rao05]). *There exists an explicit extractor $f : (\{0,1\}^n)2 \to \{0,1\}^m$ for for 2 independent c-block-sources with block entropy k and $m = \Omega(k)$, as long as $c = O((\log n)/(\log k))$.*

In this paper (see Theorem 2.4 below) we improve this result to hold even when only one of the 2 sources will be a c-block-source. The other source can be an arbitrary source with sufficient entropy. This will be a central building block in our construction. This extractor, like Rao's above, critically uses Bourgain's extractor mentioned above. In addition it uses a theorem of Raz [Raz05] allowing seeded extractors to have "weak" seeds, namely instead of being completely random they work as long as the seed has entropy rate $> 1/2$.

2 Main notions and results

The main result of this paper is an explicit disperser for 2 sources of entropy $n^{o(1)}$, significantly improving both the results of Barak et. al. [BKS+05] ($o(n)$ entropy). It also improves on Frankl and Wilson [FW81], who only built Ramsey Graphs, which in our terms is a disperser for two *identically distributed* sources with $\tilde{O}(\sqrt{n})$.

Theorem 2.1 (Main theorem, restated). *There exists an explicit* 2*-source disperser* $\mathsf{D} : (\{0,1\}^n)2 \to \{0,1\}$ *for entropy* $n^{o(1)}$*.*

Theorem 2.1 is proven in Appendix D. The construction of this disperser will use a construction of a an object which in some sense is stronger and and in another weaker than a disperser: a *subsource somewhere extractor*. We first define its "standard" cousin, the *somewhere extractor*, which is a function producing several outputs, one of which must be uniform. Again, here and later, we will ignore many technical issues such as error, min-entropy vs. entropy and more, in definitions and results, which are deferred to the technical sections.

Definition. A function $f : (\{0,1\}^n)2 \to (\{0,1\}^m)^\ell$ is a 2-source *somewhere extractor* with ℓ outputs, for entropy k, if for every 2 independent k-sources X, Y there exists an $i \in [\ell]$ such the ith output $f(X,Y)_i$ is a uniformly distributed string of m bits.

Here is a simple construction of such a somewhere extractor with ℓ as large as $\mathrm{poly}(n)$ (and the p in its name will stress the fact that indeed the number of outputs is that large). It will nevertheless be useful to us (though its description in the next sentence may be safely skipped). Define $\mathsf{pSE}(x,y)_i = \mathsf{V}(\mathsf{E}(x,i), \mathsf{E}(y,i))$ where E is a "strong" logarithmic seed extractor, and V is the Hadamard/Vazirani 2-source extractor. Using this construction, it is easy to see that:

Proposition 2.2. *For every* n, k *there is an explicit somewhere extractor* $\mathsf{pSE} : (\{0,1\}^n)2 \to (\{0,1\}^m)^\ell$ *with* $\ell = \mathrm{poly}(n)$ *outputs, for entropy* k*, and* $m = \Omega(k)$*.*

Before we define subsource somewhere extractor, we must first define a subsource.

Definition. For two distributions Z, Z' on the same universe U, we say that Z' is a *subsource* of Z if for some subset $U' \subseteq U$, Z' is the conditional distribution $Z|(Z \in U')$.

A *subsource* somewhere extractor only guarantees the "somewhere extractor" property on subsources X', Y' of the original input distributions X, Y (respectively). It will be extremely important for us to make these subsources as large as possible (i.e. we have to lose as little entropy as possible). Controlling these entropy *deficiencies* is a major technical complication we have to deal with. However we will be informal with it in this section, mentioning it only qualitatively when needed.

Definition. A function $f : (\{0,1\}^n)2 \to (\{0,1\}^m)^\ell$ is a 2-source *subsource somewhere extractor* with ℓ outputs for entropy k, if for every 2 independent k-sources X, Y there exists a subsource X' of X, a subsource Y' of Y and an $i \in [\ell]$ such the i^{th} output $f(X', Y')_i$ is a uniformly distributed string of m bits.

A central technical result for us is that with this "subsource" relaxation, we can have much fewer outputs – indeed we'll replace $\mathrm{poly}(n)$ outputs in our first construction above with $n^{o(1)}$ outputs.

Theorem 2.3 (Subsource somewhere extractor). *For every* $\delta > 0$ *there is an explicit subsource somewhere extractor* $\mathsf{SSE} : (\{0,1\}^n)2 \to (\{0,1\}^m)^\ell$ *with* $\ell = n^{o(1)}$ *outputs, for entropy* $k' = n^\delta$*, with output* $m = \sqrt{k'}$*.*

Theorem 2.3 is proven in Appendix C. We will describe the ideas used for constructing this important object and analyzing it in the next section, where we will also indicate how it is used in the construction of the final disperser. Here we state a central building block, mentioned in the previous section (as an improvement of the work of Rao [Rao05]). We construct an extractor for 2 independent sources one of which is a block-sources with sufficient number of blocks.

Theorem 2.4 (Block source and general source B.4). *There exists an explicit extractor* $\mathsf{B} : (\{0,1\}^n)2 \to \{0,1\}^m$ *for for* 2 *independent sources, one of which is a c-block-sources with block entropy* k *and the other a source of entropy* k*, with* $m = \Omega(k)$*, and* $c = O((\log n)/(\log k))$*.*

Theorem 2.4 is proven in Appendix B. A simple corollary of this block-source extractor B, is the following weaker (though useful) somewhere block-source extractor SB. A source $Z = Z_1, Z_2, \cdots, Z_t$ is a *somewhere c-block-source* of block entropy k if for some c indices $i_1 < i_2 < \cdots < i_c$ the source $Z_{i_1}, Z_{i_2}, \cdots, Z_{i_c}$ is a c-block-source. Collecting the outputs of B on every c-subset of blocks results in that somewhere extractor.

Corollary 2.5. *There exists an explicit somewhere extractor* $\mathsf{SB} : (\{0,1\}^n)2 \to (\{0,1\}^m)^\ell$ *for* 2 *independent sources, one of which is a somewhere c-block-sources with block entropy k and the other a source of entropy k, with* $m = \Omega(k)$*,* $c = O((\log n)/(\log k))$*, and* $\ell \leq t^c$*.*

In both the theorem and corollary above, the values of entropy k we will be interested in are $k = n^{\Omega(1)}$. It follows that a block-source with a constant $c = O(1)$ suffices.

3 The Challenge-Response mechanism

We now describe abstractly a mechanism which will be used by the disperser as well as the subsource somewhere extractor. Intuitively, this technique allows us to identify parts of a source which contain large amounts of entropy. One can hope that using such a method one can partition a given source into blocks in a way which make it a block-source, or alternatively focus on a part of the source which is unusually condensed with entropy - two cases which may simplify the extraction problem.

The reader may decide, now or in the middle of this section, to skip ahead to the next section which describes the construction of the subsource somewhere extractor SSE, which extensively uses this mechanism. Then this section may seem less abstract, as it will be clearer where this mechanism is used.

This mechanism was introduced by Barak et. al. [BKS+05], and was essential in their 2-source disperser. Its use in this paper is quite a bit more complicated (and in particular it calls itself recursively, a fact which creates many subtleties). However the basic idea behind it the mechanism is the same:

Let Z be a source and Z' a part of Z (Z projected on a subset of the coordinates). We know that Z has entropy k, and want to distinguish two possibilities: Z' has no entropy (it is fixed) or it has at least k' entropy. Z' will get a *Pass* or *Fail* grade, hopefully corresponding to the cases of high or no entropy in Z', respectively.

Anticipating the use of this mechanism, it is good idea to think of Z as a "parent" of Z', which wants to check if this "child" has sufficient entropy. Moreover, in the context of the initial 2 sources X, Y we will operate on, think of Z as a part of X, and thus that Y is independent of Z and Z'.

To execute this "test" we will compute two sets of strings (all of length m, say): the *Challenge* $C = \mathsf{C}(Z', Y)$ and the *Response* $R = \mathsf{R}(Z, Y)$. Z' fails if $C \subseteq R$ and passes otherwise.

The key to the usefulness of this mechanism is the following lemma, which states that what "should" happen, indeed happens after some restriction of the 2 sources Z and Y. We state it and then explain how the functions C and R are define to accommodate its proof.

Lemma 3.1. *Assume* Z, Y *are sources of entropy* k*.*

1. *If* Z' *has entropy* $k' + O(m)$*, then there is a subsources* $\hat{Z}$ *of* Z *and* $\hat{Y}$ *of* Y*, such that*

$$\Pr[\hat{Z}' \textit{ passes}] = \Pr[\mathsf{C}(\hat{Z}', \hat{Y}) \subseteq \mathsf{R}(Z, Y)] \geq 1 - n^{O(1)} 2^{-m}$$

2. *If* Z' *is fixed (namely, has zero entropy), then for some subsources* $\hat{Z}$ *of* Z *and* $\hat{Y}$ *of* Y*, we have*

$$\Pr[Z' \textit{ fails}] = \Pr[\mathsf{C}(\hat{Z}', \hat{Y}) \subseteq \mathsf{R}(\hat{Z}, \hat{Y})] = 1$$

Now let us explain the choice of the functions C and R. We will use some of the constructs above with parameters which don't quite fit. The reader should simply believe that we can make them fit, or peek in the technical sections for the precise definitions.

The response $\mathsf{R}(Z,Y) = \mathsf{pSE}(Z,Y)$ is chosen to be the somewhere extractor of Proposition 2.2. The challenge $\mathsf{C}(Z',Y) = \mathsf{SSE}(Z',Y)$ is chosen to be the subsource somewhere of Theorem 2.3.

Why does it work? We explain each of the two claims in the lemma in turn (and after each comment on the important parameters and how they differ from Barak et. al. [BKS+05]).

1. Z' has entropy. We need to show that Z' passes the test with high probability. We will point to the output string in $\mathsf{C}(\hat{Z}',\hat{Y}')$ which avoids $\mathsf{R}(\hat{Z},\hat{Y})$ with high probability as follows. It will be a union bound on all (poly(n) many) strings in $\mathsf{pSE}(\hat{Z},\hat{Y})$. We note that by the definition of the response function, to fix a particular element in the response set to a particular value can be done by fixing $\mathsf{E}(Z,i)$ and and $\mathsf{E}(Y,i)$ – keeping the restricted sources independent and losing only $O(m)$ entropy. But in the subsource of Z' guaranteed to exist by Theorem 2.3 we can afford to lose this entropy in Z', concluding that one of its outputs is uniform. The probability that this output will equal any fixed value is thus 2^{-m}, completing the argument. **Remark:** We note that we can handle the polynomial output size of pSE, since the uniform string has length $m = n^{\Omega(1)}$ (something which could not be done with the technology available to Barak et. al. [BKS+05]) is much longer and allows the above union bound.

2. Z' has no entropy. We now need to guarantee that in the chosen subsources (which we choose) $\hat{Z},\hat{Y}$, all strings in $C = \mathsf{C}(\hat{Z}',\hat{Y})$ are in $\mathsf{R}(\hat{Z},\hat{Y})$. First notice that as Z' is fixed, C is only a function of Y. We set $\tilde{Y}$ to be the subsource of Y that fixes all strings in $C = \mathsf{C}(Y)$ to their most popular values (losing only ℓm entropy from Y). We take care of including these fixed strings in $\mathsf{R}(Z,\tilde{Y})$ one at a time, by restricting to subsources assuring that. Let σ be any m-bit string we want to appear in $\mathsf{R}(Z,\tilde{Y})$. Recall that $\mathsf{R}(z,y) = \mathsf{V}(\mathsf{E}(z,i),\mathsf{E}(y,i))$. We pick a "good" seed i, and restrict Z, $\tilde{Y}$ to subsources with only $O(m)$ less entropy by fixing $\mathsf{E}(Z,i) = a$ and $\mathsf{E}(\tilde{Y},i) = b$ to values (a,b) for which $\mathsf{V}(a,b) = \sigma$. This is repeated successively ℓ times, and results in the final subsources $\hat{Z},\hat{Y}$ on which $\hat{Z}'$ fails with probability 1. **Remark:** Note that we keep reducing the entropy of our sources ℓ times, which necessitates that this ℓ be tiny (here we could not tolerate $poly(n)$, and indeed can guarantee $n^{o(1)}$, at least on a subsource – this is one aspect of how crucial the subsource somewhere extractor SSE is to the construction.

4 The subsource somewhere extractor SSE

We now explain some of the ideas behind the construction of the subsource somewhere extractor SSE of Theorem 2.3. The full details of the construction and proof are given in Appendix C. Consider the source X. We are seeking to find in it a somewhere c-block-source, so that we can use it (together with Y) in the block-source extractor of Theorem 2.5. Like in previous works in the extractor literature (e.g. [RSW00, LRVW03]) we use a "win-win" analysis which shows that either X is already a somewhere c-block-source, or it has a condensed part in which we proceed recursively on that part. Continuing this way we eventually reach a source so condensed for which the first case must happen.

Consider the tree of parts of the source X evolved by such recursion. Some node in the tree is "good", namely corresponds to a somewhere c-source extractor, but we don't know which. Since we only want a somewhere extractor, we can apply to each the somewhere block-source extractor of Corollary 2.5 – this will give us a random output in every "good" node of the tree. The usual idea is output all these values (and in seeded extractors, merge them using the externally given random seed). But we cannot afford to do that here, as there is no external seed, and the number of these outputs (the size of the tree) is far too large.

Our aim then will be to significantly prune this number of candidates and in fact output only the candidates on *one* path to a *canonical* "good" node. Here first is a very informal description how this is done. To do so, before calling SSE recursively on a subpart of a current part of X, we'll use the "Challenge-Response" mechanism described above to check if "it has entropy".[4] We will recurse only with the *first* (in left-to-right

[4] We note that we ignore the additional complication that SSE will actually use recursion also to compute the challenge in the

order) part which passes the "entropy test". Thus note that we will follow a single path on this tree. The algorithm SSE will output only the sets of strings produced by applying the somewhere c-block-extractor SB on the parts visited along this path.

Proving that indeed there are subsources on which SSE will follow a path to a "good" (for these subsources!) node, is the heart of the analysis. It is especially complex due to the fact that the recursive call to SSE on a subparts of the current part is used to generate the Challenges for the Challenge-Response mechanism, and since SSE works only on a subsources we have to guarantee that restriction to these does not hamper the behavior of SSE in past and future calls to it!

Let us describe the recursive structure of SSE. We fix a parameter t (the constraints on this choice will be explained later) and recall that we are after a somewhere c-block-source (for some known constant c).

SSE gets as inputs a string z (sample from the current part Z of the original source X and a string y (sample from the source Y). Initially $Z = X$. It outputs a set of strings of length m.

If z is shorter than $\sqrt{k}$ (where k is the original entropy) SSE returns the empty set. Otherwise, it breaks z into t equal size strings $z = z_1, z_2, \cdots, z_t$, for each i computes the Challenge $\mathsf{C}(z_i, y)$ (using recursion) and Response $\mathsf{R}(z, y)$, and calls itself recursively with z_i, y for the smallest i that *passes* the challenge-response test. Finally, its output is the output of the recursive call, *union* with the t^c outputs of applying the somewhere c-block-source extractor $\mathsf{SB}(z, y)$.

Let us turn to the highlights of the analysis, for the proof of Theorem 2.3. Let k be the entropy of the source Z at some place in this recursion. Either one of its blocks Z_i has entropy k/c, in which case it is very condensed, since its size is n/t for $t \gg c$), or it must be that c of its blocks form a c-block source with block entropy k/t (which is sufficient for the extractor B used by SB). In the 2nd case the fact that $\mathsf{SB}(z, y)$ is part of the output of of our SSE guarantees that we are somewhere random. If the 2nd case doesn't hold, let Z_i be the leftmost condensed block. We want to ensure that (on appropriate subsources) SSE calls itself on that ith subpart. To do so, we fix all Z_j for $j < i$ to constants z_j. We are now in the position described in the Challenge-Response mechanism section, that (in each of the first i parts) there is either no entropy or lots of entropy. We further restrict to subsources as explained there which make all first $i - 1$ blocks *fail* the "entropy test", and the fact that Z_i still has lots of entropy after these restrictions (which we need to prove) ensures that indeed SSE will be recursively applied to it.

We note that while the procedure SSE can be described recursively, the formal analysis of fixing subsources is actually done globally, to ensure that indeed all entropy requirements are met along the various recursive calls.

Let us remark on the choice of the branching parameter t. On the one hand, we'd like to keep it small, as it dominates the number of outputs t^c of SB, and thus the total number of outputs (which is $t^c \log_t n$). For this purpose, any $t = n^{o(1)}$ will do. On the other hand, t should be large enough so that condensing is faster than losing entropy. Here note that if Z is of length n, its child has length n/t, while the entropy shrinks only from k to k/c. A simple calculation shows that if $k^{(\log t)/\log c} > n2$ then a c block-source must exist along such a path before the length shrinks to $\sqrt{k}$. Note that for $k = n^{\Omega(1)}$ a (large enough) constant t suffices (resulting in only logarithmic number of outputs of SSE). Optimizing t further to accommodate the requirements of its use by the final disperser D is explained in the technical section. This analysis is depicted pictorially in Figure 1.

5 The final disperser D

Following is a rough description of our disperser D proving Theorem 2.1. For full details and proof see Appendix D. The high level structure of D will resemble the structure of SSE - we will recursively split the source X and look for entropy in the parts. However now we must output a *single* value (rather than a set)

challenge-response mechanism.

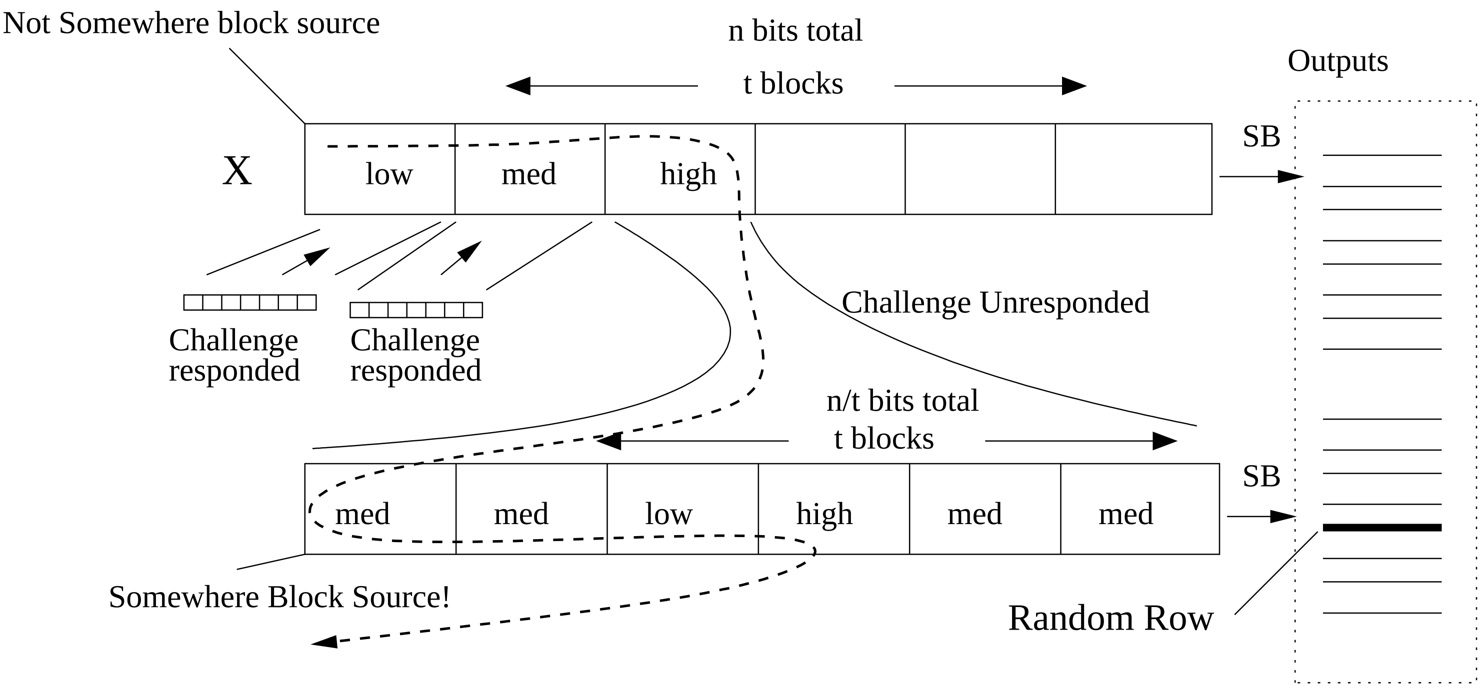

Figure 1: Analysis of the subsource somewhere extractor.

which can take both values 0 and 1. This was problematic in SSE, even knowing where the "good" part (containing a c-block-source) was! How can we do so now?

We now have at our disposal a much more powerful tool for generating challenges (and thus detecting entropy), namely the subsource somewhere disperser SSE. Note that in constructing SSE we only had essentially the somewhere c-block-source extractor SB to (recursively) generate the challenges, but it depended on a structural property of the block it was applied on. Now SSE does not assume any structure on its input sources (except sufficient entropy, but there is a catch – it only works on subsources of them! This will cause us a lot of head ache we will elaborate on later.

Let us now give a high level description of the disperser D. It too will be a recursive procedure. If when processing some part Z of X it "realizes" that a subpart Z_i of Z has entropy, but not *all* the entropy of Z (namely Z_i, Z is a 2-block-source) then we will halt and produce the output of D. Intuitively, thinking about the Challenge-Response mechanism TestBlock described above, the analysis implies that we can either *pass* or *fail* Z_i (on appropriate subsources). But this means that the outcome of this "entropy test" TestBlock **is** a 1-bit disperser!

To capitalize on this idea, we want to use SSE to identify such a block-source in the recursion tree. As before, we scan the blocks from left to right, and want to distinguish three possibilities.

1. **Small** Z_i has a small amount entropy. In this case we proceed to $i+1$.

2. **Medium** Z_i has "medium" entropy (Z_i, Z is a block-source). In which case we halt and produce an output (zero or one).

3. **Large** Z_i has essentially all entropy of Z. In this case we recurse on the condensed block Z_i.

As before, we use the Challenge-Response mechanism (with a twist). We will compute challenges $\mathsf{C}(Z_i, Y)$ and responses $\mathsf{R}(Z, Y)$, all strings of length m. The responses are computed exactly as before, using the somewhere extractor pSE. The Challenges are computed using our subsource somewhere extractor SSE.

We really have 4 possibilities to distinguish, since when we halt we also need to decide which output bit we give. We will do so by deriving three tests from the above challenges and responses: $(C_L, R_L), (C_M, R_M), (C_S, R_S)$

for Large, Medium and Small respectively, as follows. Let $m \geq m_L >> m_M >> M_S$ be appropriate integers: then in each of the tests above we restrict ourselves to prefixes of all strings of the appropriate lengths only (e.g. prefixes of length m_S in (C_S, R_S) etc).

First note the obvious monotonicity of the tests. If Z_i fails one of the tests it will certainly fail for shorter strings. Thus there are only four outcomes to the three tests (written in the order (L,M,S)): (pass, pass, pass), (pass, pass, fail), (pass, fail, fail) and (fail,fail,fail). We act as follows.

1. (fail, fail, fail): Assume Z_i has small entropy and proceed to block $i + 1$.
2. (pass, fail, fail): Assume Z_i is medium and output 0.
3. (pass, pass, fail): Assume Z_i is medium and output 1.
4. (pass, pass, pass): Assume Z_i is large and recurse on Z_i.

.

The analysis of this idea turns out to be more complex than it seems. There are two reasons for that. Now we briefly explain them and the way to overcome them in the construction and analysis.

The first reason is the fact mentioned above, that SSE which generates the challenges, works only on a subsources of the original sources. Restricting to these subsources at some level of the recursion (as required by the analysis of of the test) causes entropy loss which affects the both definitions (such as these entropy thresholds for decisions) and correctness of SSE in higher levels of recursion. Controlling this entropy loss is achieved by calling SSE recursively with smaller and smaller entropy requirements, which in turn limits the entropy which will be lost by these restrictions. In order not to lose all the entropy for this reason alone, we must work with special parameters of SSE, essentially requiring that at termination it has almost all the entropy it started with.

The second reason is the analysis of the test when we are in a medium block. In contrast the analysis of TestBlock above, we cannot consider the value of Z_i fixed when we need it to fail on the Medium and Small tests. We need to show that for these two tests (given a *pass* for Large), they come up both (pass, fail) and (fail, fail) each with positive probability.

Since the length of Medium challenges and responses is m_M, probability of failure is at least $\exp(-\Omega(m_M))$ (this follows relatively easily from the fact that the responses are somewhere random). If the Medium test fails so does the Small test, and thus (fail, fail) has a positive probability and our disperser D outputs 0 with positive probability.

To bound (pass, fail) we first observe (with a similar reasoning) that the Small test fails with probability at least $\exp(-\Omega(m_S))$. But we want the Medium test to pass at the same time. This probability is at least the probability that Small fails minus the probability that Medium fails. But we have a bound on the latter already in the analysis of TestBlock: it is at most poly$(n)exp(-\ell m_M)$. He comes our control of the different length into play - we can make the m_S sufficiently smaller than m_M to yield this difference positive. We conclude that our disperser D outputs 1 with positive probability as well.

We need only take care finally of termination, namely that the recurrence *always* arrives at a Medium subpart, but it is easy to chose entropy thresholds for Small, Medium and Large to ensure that this happens. Pictorially, this analysis is depicted in Figure 2.

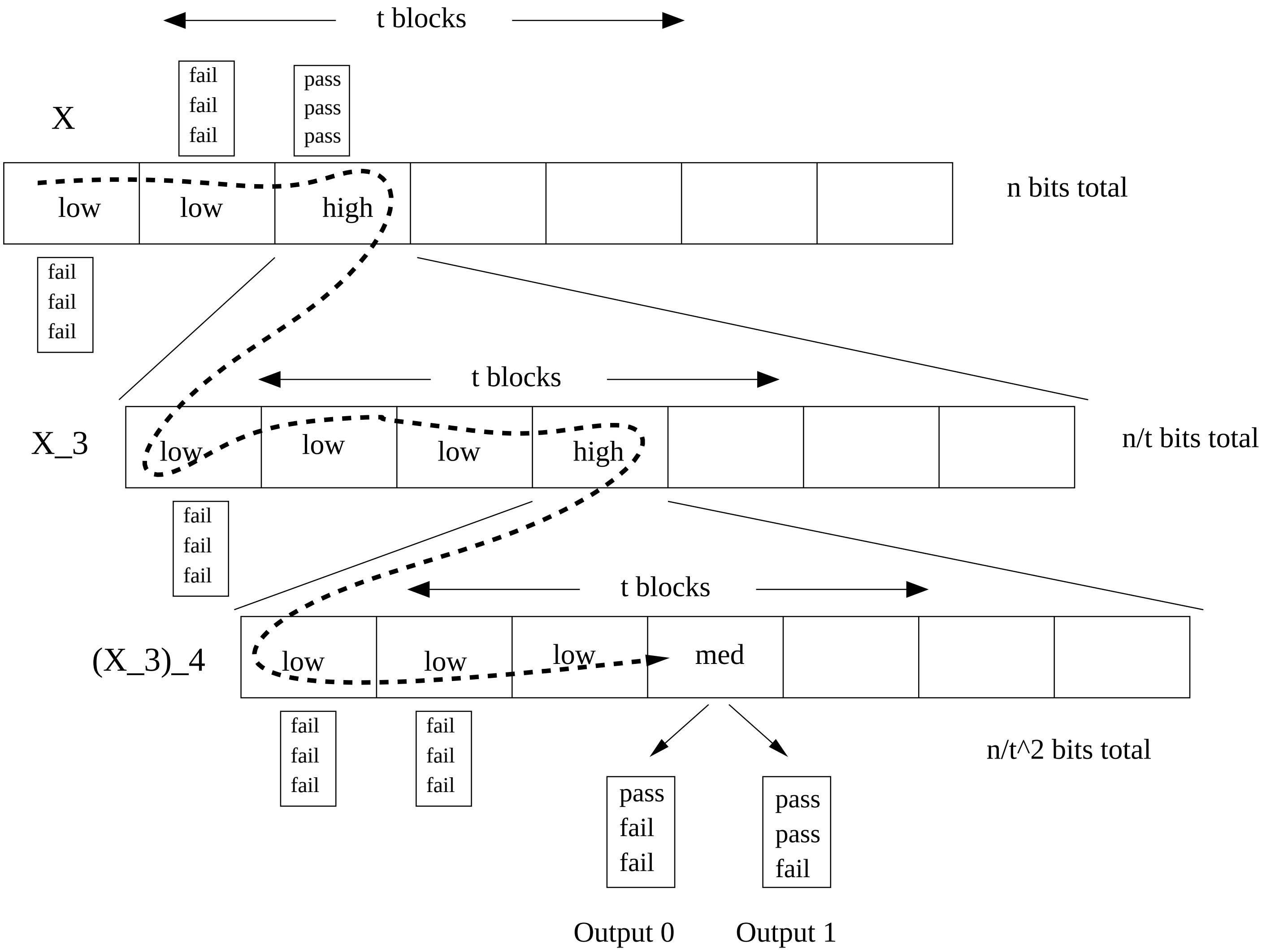

Figure 2: Analysis of the disperser.

References

[Alo98] N. Alon. The Shannon Capacity of a Union. *Combinatorica*, 18, 1998.

[BIW04] B. Barak, R. Impagliazzo, and A. Wigderson. Extracting Randomness Using Few Independent Sources. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 384–393, 2004.

[BKS+05] B. Barak, G. Kindler, R. Shaltiel, B. Sudakov, and A. Wigderson. Simulating Independence: New Constructions of Condensers, Ramsey Graphs, Dispersers, and Extractors. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 1–10, 2005.

[Bou05] J. Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 1:1–32, 2005.

[BKT04] J. Bourgain, N. Katz, and T. Tao. A Sum-Product Estimate in Finite Fields, and Applications. *Geometric and Functional Analysis*, 14:27–57, 2004.

[CRVW02] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness Conductors and Constant-Degree Lossless Expanders. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 659–668, 2002.

[CG88] B. Chor and O. Goldreich. Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[FW81] P. Frankl and R. M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1(4):357–368, 1981.

[Gro00] V. Grolmusz. Low Rank Co-Diagonal Matrices and Ramsey Graphs. *Electr. J. Comb*, 7, 2000.

[Gur03] V. Guruswami. Better Extractors for Better Codes? *Electronic Colloquium on Computational Complexity (ECCC)*, (080), 2003.

[LRVW03] C. J. Lu, O. Reingold, S. Vadhan, and A. Wigderson. Extractors: Optimal up to Constant Factors. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 602–611, 2003.

[MNSW98] P. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57:37–49, 1 1998.

[NZ93] N. Nisan and D. Zuckerman. More Deterministic Simulation in Logspace. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 235–244, 1993.

[PR04] P. Pudlak and V. Rodl. Pseudorandom sets and explicit constructions of Ramsey graphs. *Submitted for publication*, 2004.

[Rao05] A. Rao. Extractors for a Constant Number of Polynomial Min-entropy Independent Sources. Technical Report TR05-106, ECCC: Electronic Colloquium on Computational Complexity, 2005. Submitted to this conference.

[Raz05] R. Raz. Extractors with Weak Random Seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.

[RRV02] R. Raz, O. Reingold, and S. Vadhan. Extracting all the Randomness and Reducing the Error in Trevisan's Extractors. *J. Comput. Syst. Sci*, 65(1):97–128, 2002.

[RSW00] O. Reingold, R. Shaltiel, and A. Wigderson. Extracting Randomness via Repeated Condensing. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 22–31, 2000.

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

[Sha02] R. Shaltiel. Recent Developments in Explicit Constructions of Extractors. *Bulletin of the European Association for Theoretical Computer Science*, 77:67–95, 2002.

[TSZ04] A. Ta-Shma and D. Zuckerman. Extractor Codes. *IEEE Transactions on Information Theory*, 50, 2004.

[Tre01] L. Trevisan. Extractors and Pseudorandom Generators. *Journal of the ACM*, pages 860–879, 2001.

[Vaz85] U. Vazirani. Towards a Strong Communication Complexity Theory or Generating Quasi-Random Sequences from Two Communicating Slightly-random Sources (Extended Abstract). In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 366–378, 1985.

[WZ99] A. Wigderson and D. Zuckerman. Expanders that Beat the Eigenvalue Bound: Explicit Construction and Applications. *Combinatorica*, 19(1):125–138, 1999.

A Preliminaries

The following are some definitions and lemmas that are used throughout this paper. This section can be skimmed over in the first reading.

A.1 Basic notations and definitions.

Random variables, sources and min-entropy. We will usually deal with random variables which take values over $\{0,1\}^n$. We call such a random variable an *n-bit source*. The *min-entropy* of a random variable X, denoted by $H_\infty(X)$, is defined to equal $\min_x\{-\log_2(\Pr[X=x])\}$, or equivalently $\log_2(1/\max_x\{\Pr[X=x]\})$. If X is an n-bit source with $H_\infty(X) \geq k$ and n is understood from the context then we'll call X a *k-source*.

Statistical distance. If X and Y are random variables over some universe U, the *statistical distance* of X and Y, denoted by $|X-Y|$ is defined to be $\frac{1}{2}\sum_{u\in U}|\Pr[X=u]-\Pr[Y=u]|$.

We restate here the definition of a sub-source:

Definition A.1 (Subsources)**.** Given random variables X and X' on $\{0,1\}^n$ we say that X' is a *deficiency d subsource* of X and write $X' \subseteq X$ if there exists a set $A \subseteq \{0,1\}^n$ such that $(X|A) = X'$ and $\Pr[X \in A] \geq 2^{-d}$.

Definition A.2 (Block sources)**.** A distribution $X = X^1 \circ X^2 \circ \cdots \circ X^{\mathsf{C}}$ is called a $(k_1, k_2, \ldots, k_{\mathsf{C}})$-block source if for all $i = 1, \ldots, \mathsf{C}$, we have that for all $x_1 \in X^1, \ldots, x_{i-1} \in X^{i-1}$, $H_\infty(X^i|X^1 = x_1, \ldots, X^{i-1} = x_{i-1}) \geq k_i$, i.e., each block has high min-entropy even conditioned on the previous blocks. If $k_1 = k_2 = \cdots = k_{\mathsf{C}}$ then we say that X is a k-block source.

Definition A.3 (Somewhere $\mathcal{P}$ sources)**.** Let $\mathcal{P}$ be a property of sources. A source $X = (X_1, ..., X_t)$ is $(t \times r)$ *somewhere-$\mathcal{P}$* if each X_i takes values in $\{0,1\}^r$ and there is a random variable I taking values in $[t]$ s.t. for every $i \in [t]$, $X_I|I = i$ has property $\mathcal{P}$. If the property $\mathcal{P}$ is that the source is uniformly random, we will call the source somewhere random (SR-source for short).

Because we are mixing block sources and somewhere random sources, for clarity we will use the convention that the word *block* refers to a block of a block source. The word *row* will be used to refer to a row in a somewhere random source.

Definition A.4 (Weak somewhere random sources)**.** A source X is $(t\times r)$ *k-somewhere-random* (*k-SR-source* for short) if it is a somewhere $\mathcal{P}$ source where a source has property $\mathcal{P}$ if it has min-entropy k.

Often we will need to apply a function to every row of a somewhere source. We will adopt the following convention: if $f : \{0,1\}^r \times \{0,1\}^r \to \{0,1\}^m$ is a function and a, b are samples from $(t \times r)$ somewhere sources, $f(\vec{a}, \vec{b})$ refers to the $(t \times m)$ string whose first row is obtained by applying f to the first rows of a, b and so on.

Definition A.5 (Resilience of Properties)**.** Let $\mathcal{P}$ be a property of random variables. Let Z, X, Y be random variables on the same probability space s.t. Z has property $\mathcal{P}$. We will say that the property $\mathcal{P}$ of Z is *d-resilient with respect to X* if for every d deficiency subsource $X' \subseteq X$, $Z|(X \in \mathsf{Supp}(X'))$ has property $\mathcal{P}$. We will say that the property $\mathcal{P}$ of Z is *d-resilient with respect to X and d' resilient with respect to Y* if for every deficiency d subsource $X' \subseteq X$ and every deficiency d' subsource $Y' \subseteq Y$, $Z|(X \in \mathsf{Supp}(X'))$ has property $\mathcal{P}$.

A.1.1 Extractors, dispersers and their friends.

In this section we define some of the objects we will later use and construct. All of these objects will take two inputs and produce one output, such that under particular guarantees on the distribution of the input, we'll get some other guarantee on the distribution of the output. Various interpretation of this vague sentence lead to extractors, dispersers, somewhere extractors, block extractors etc..[5]

Definition A.6 (Two-source extractor)**.** Let $n_1, n_2, k_1, k_2, m, \epsilon$ be some numbers. A function $\mathsf{Ext} : \{0,1\}^{n_1 \times n_2} \to \{0,1\}^m$ is called an 2-*source extractor with* k_1, k_2 *min-entropy requirement*, n*-bit input*, m*-bit output and* ϵ*-statistical distance* if for every independent sources X and Y over $\{0,1\}^{n_1}$ and $\{0,1\}^{n_2}$ respectively satisfying

$$H_\infty(X) \geq k_1 \quad \text{and} \quad H_\infty(Y) \geq k_2 \tag{1}$$

it holds that

$$\left| \mathsf{Ext}(X, Y) - U_m \right| \leq \epsilon \tag{2}$$

In the common case of a *seeded extractor* we have $n_2 = k_2$ (and hence the second input distribution is required to be uniform). A non-trivial construction will satisfy of course $n_2 \ll m$ (and hence also $n_2 \ll k_1 < n$). Thus, two source extractors are strictly more powerful than seeded extractor. However, the reason seeded extractors are more popular is that they suffice for many applications, and that (even after this work) the explicit construction for seeded extractors have much better parameters than the explicit constructions for 2-source extractors with $k_1 \ll n_1$, $k_2 \ll n_2$. (Note that this is not the case for *non-explicit* construction, where 2-source extractors with similar parameters to the best possible seeded extractors can be shown to exist using the probabilistic method.)

Generalization. We'll use many variants of extractors in this paper to various similar combinatorial objects. Most of the variants are obtained by giving different the conditions on the input (Equation 1) and the guarantee on the output (Equation 2). Some of the variants we will consider will be:

Dispersers. In dispersers, the output guarantee (Equation 2) is replaced with $|\mathsf{Supp}(\mathsf{Ext}(X, Y))| \geq (1-\epsilon)2^m$,

Somewhere extractors. In *somewhere extractors* the output guarantee (Equation 2) is replaced with the requirement that $|\mathsf{Ext}(X, Y) - Z| < \epsilon$ where Z is a *somewhere random source* of $t \times m$ rows for some parameter t.

Extractors for block sources. In *extractors for block sources* the input requirement (Equation 1) is replaced with requirement that X and Y are *block sources* of specific parameters. Similarly we will define extractors for other families of inputs (i.e. somewhere random sources) and extractors where each input should come from a different family.

Strong extractors. Many of these definition have also a *strong* variant, and typically constructions for extractors also achieve this strong variant. An extractor is *strong in the first input* if the output requirement (Equation 2) is replaced with $|(X, \mathsf{Ext}(X, Y)) - (X, U_m)| \leq \epsilon$. We define an extractor to be strong in the second input similarly. If the extractor is strong in both inputs, we simply say that it is *strong*.

Subsource extractors. This paper introduces a new definition of a *subsource extractor*. This definition, somewhat between an extractor and a disperser, is perhaps a bit convoluted but turns out to be very useful. We say that Ext is a subsource extractor if given sources X and Y it outputs a close-to-uniform distribution not necessarily on X and Y, but on some independent subsources $X^{\mathsf{good}} \subseteq X$ and $Y^{\mathsf{good}} \subseteq Y$ with some deficiency def. This smaller def is, the better the extractor (with $\mathsf{def} = 0$ making the definition equivalent to standard extractors, while it can be shown that $\mathsf{def} = n$ will yield a disperser). We also introduce a new parameter for subsource extractors called *resilience*. We say that a Ext has resiliency res if for *every*

[5] In fact this can be generalized even further to more than two sources (e.g., see [BIW04, BKS+05, Rao05]), but we do not need this generalization here.

independent $X' \subseteq X^{\mathsf{good}}$ and $Y' \subseteq Y^{\mathsf{good}}$ with deficiency at most res. This time, the larger res is, the better the extractor. We also generalize the notion of subsource extractors to variants such as somewhere extractors and extractors for block sources.

Remark A.7 (Input lengths)**.** Whenever we have a two source extractor $\mathsf{Ext} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ with inputs lengths n_1, n_2 and entropy requirement k_1, k_2 we can always invoke it on shorter sources with the same entropy, by simply padding it with zeros. In particular if we have an extractor with $n_1 = n_2$ we can still invoke it on inputs of *unequal* length by padding one of the inputs. The same observation holds for the other source types we'll use, namely *block* and *somewhere random* sources, if the padding is done in the appropriate way (i.e., pad each block for block sources, add all zero rows for somewhere random sources), and also holds for all the other extractor-like objects we consider (dispersers, somewhere extractors, and their subsource variant). In the following, whenever we invoke an extractor on inputs shorter than its "official" input length, this means that we use such a padding scheme.

A.2 Useful facts and lemmas.

A.2.1 Basic properties of sources and subsources.

Fact A.8. *If X is an (n, k)-source and X' is a deficiency d subsource of X then X' is an $(n, k - d)$-source.*

Fact A.9. *Let X be a random variable with $H_\infty(X) = k$. Let A be any event in the same probability space. Then*

$$H_\infty(X|A) < k' \Rightarrow \Pr[A] < 2^{k'-k}$$

Fact A.10 (Deficiency of sub-subsources.)**.** *Let X' be a subsource of X with deficiency d' and let X'' be a subsource of X' with deficiency d''. Then X'' is a subsource of X with deficiency $d' + d''$.*

Lemma A.11. *Let X be a k-source, Y be a (possibly dependent) random variable on $\{0,1\}^m$ and $\rho > 0$. Then, there exist a set A such that:*

- $\Pr[Y \in A] \leq \rho$.
- *For every $a \in A$, $(X|Y \in A)$ is a deficiency $m + \log(1/\rho)$ subsource of X.*

A.2.2 Fixing functions and projections.

Given a source X over $\{0,1\}^n$ and a function $F : \{0,1\}^n \to \{0,1\}^m$, we often will want to consider subsources of X where F is fixed to some value, and provide some bounds on the deficiency. Thus, the following lemma would be useful:

Lemma A.12 (Fixing a function.)**.** *Let X be a distribution over $\{0,1\}^n$, $F : \{0,1\}^n \to \{0,1\}^m$ be a function, and $\ell \geq 0$ some number. Then there exists a $a \in \{0,1\}^m$ and a deficiency m subsource X' of X such that $F(x) = a$ for every x in X'. Furthermore, for every $a \in \mathsf{Supp}(F(X))$ let X_a be the subsource of X defined by conditioning on $F(X) = a$. Then, if we choose a at random from the source $F(X)$ then with probability $\geq 1 - 2^{-\ell}$, the deficiency of X_a is at most $m + \ell$.*

Proof. Let $\ell > 0$ be some number and let A be the set of $a \in \{0,1\}^m$ such that $\Pr[F(x) = a] < 2^{-m-\ell}$. Since $|A| \leq 2^m$, we have that $\Pr[F(X) \in A] < 2^{-\ell}$. If we choose $a \leftarrow_{\mathrm{R}} F(X)$ and $a \notin A$, we get that $X|F(X) = a$ has deficiency $\leq m + \ell$. Choosing $\ell = 0$ we get the first part of the lemma, and choosing $\ell = m$ we get the second part. □

The following lemma will also be useful:

Lemma A.13. *Let X be an (n, k) source. Let $S \subseteq [n]$ with $|S| = n'$. Let $X_{|S}$ denote the projection of X to the bit locations in S. Then for every l, $X_{|S}$ is 2^{-l}-close to a $(n', k - (n - n') - l)$ source.*

Proof. Let $\overline{S}$ be the complement of S.

Then $X_{|S}$ is a convex combination over $X_{|\overline{S}}$. For each setting of $X_{|\overline{S}} = h$, we induce the distribution $X_{|S}|X_{|\overline{S}} = h$.

Define $H = \{h \in \{0,1\}^{n-n'} | H_\infty(X_{|S}|X_{|\overline{S}} = h) < n' - n + k - l\}$. Notice that $H_\infty(X_{|S}|X_{|\overline{S}} = h) = H_\infty(X|X_{|\overline{S}} = h)$. Then by Fact A.9, for every $h \in H$, $\Pr[X_{|\overline{S}} = h] < 2^{k-(n-n')-l-k} = 2^{-(-n'+n+l)}$. Since $|H| \leq 2^{n-n'}$, by the union bound we get that $\Pr[X_{|\overline{S}} \in H] \leq 2^{-l}$].

□

In some situations we will have a source that is statistically close to having high min-entropy, but not close enough. We can use the following lemma to lose something in the entropy to get 0 error.

Lemma A.14. *Let X be a random variable over $\{0,1\}^n$ s.t. X is $1/4$-close to an (n, k) source. Then there is a deficiency 2 subsource $X' \subseteq X$ s.t. X' is a $(n, k - 3)$ source.*

Proof. Let t be a parameter that we will pick later. Let $H \subseteq \mathsf{Supp}(X)$ be defined as $H = \{x \in \mathsf{Supp}(X) | \Pr[X = x] > 2^{-t}\}$. H is the set of heavy points of the distribution X. By the definition of H, $|H| \leq 2^t$.

Now we have that $\Pr[X \in H] - 2^{-k}|H| \leq \epsilon$, since X is ϵ-close to a source with min-entropy k. This implies that $\Pr[X \in H] \leq \epsilon + 2^{-k}|H| \leq \epsilon + 2^{t-k}$.

Now consider the subsource $X' \subseteq X$ defined to be $X|X \in (\mathsf{Supp}(X) \setminus H)$. For every $x \in \mathsf{Supp}(X')$, we get that

$\Pr[X' = x] = \Pr[X = x|X \notin H] \leq \frac{\Pr[X=x]}{\Pr[X \notin H]} \leq \frac{2^{-t}}{1-(\epsilon+2^{t-k})}$

Setting $t = k - 2$, we get that $\Pr[X' = x] \leq \frac{2^{-k+2}}{1-(\epsilon+2^{-2})} \leq 2^{-k+3}$.

□

A.2.3 Convex combinations.

Definition A.15 (Convex combination)**.** Let X be a random variable and let $\{Y_i\}_{i \in U}$ be a family of random variables indexed by an element in some universe U. We say that X is a convex combination of the family $\{Y_i\}$ if there exists a random variable I over U such that $X = Y_I$.

A key observation that is essential to our results is that random variables that are convex combinations of sources with some good property are usually good themselves. This is captured in the following easy propositions:

Proposition A.16. *Let X, Z be random variables s.t. X is a convex combination of sources which are ϵ-close to Z. Then X is ϵ-close to Z.*

Proposition A.17 (Preservation of properties under convex combination)**.** *Let A, B, Q be random variables over the same finite probability space such that*

$$\Pr_{q \leftarrow_R Q}[|(A|Q = q) - (B|Q = q)| \geq \epsilon_1] < \epsilon_2$$

then $|A - B| < \epsilon_1 + \epsilon_2$.

Proof.

$$
\begin{aligned}
&|A - B| \\
&= \sum_{x \in \mathsf{Supp}(A) \cup \mathsf{Supp}(B)} \Big| \Pr[A = x] - \Pr[B = x] \Big| \\
&\leq \sum_{q \in \mathsf{Supp}(Q)} \Pr[Q = q] \sum_{x \in \mathsf{Supp}(A) \cup \mathsf{Supp}(B)} \Big| \Pr[A = x | Q = q] - \Pr[B = x | Q = q] \Big| \\
&\leq \epsilon_2 + \sum_{\text{good } q \in \mathsf{Supp}(Q)} \Pr[Q = q] \epsilon_1 \\
&\leq \epsilon_1 + \epsilon_2
\end{aligned}
$$

□

This proposition will be used to simply the proofs that many of our constructions are *strong*. Usually it is easy to show that our output distributions are convex combinations of distributions that are close to having the strong property. Here we show that this implies that the output actually has the strong property. To make it easier to think about, think of X in the lemma below as one of the inputs, Q as some internal random variable, O as the output of our extractor/disperser and U as some independent random variable that is uniformly distributed. Then the following lemma is just a special case of the above proposition. In words it just says that if X is close to being independent of O for each subsource in the convex combination, it is fact close to being independent of O after the convex combination.

Lemma A.18 (Preservation of strongness under convex combination)**.** *Let X, O, U, Q be random variables over the same finite probability space, with U, O both random variables over $\{0,1\}^m$. Let $\epsilon_1, \epsilon_2 < 1$ be constants s.t. :*

$$\Pr_{q \leftarrow_R Q}[|(X|Q = q) \circ (O|Q = q) - (X|Q = q) \circ (U|Q = q)| \geq \epsilon_1] < \epsilon_2$$

,

i.e. conditioned on Q being fixed and good, *$X \circ O$ is statistically close to $X \circ U$.*

Then we get that $|X \circ O - X \circ U| < \epsilon_1 + \epsilon_2$.

Proposition A.19. *Let X, I be random variables s.t. X is a convex combination of random variables $\{X_i\}_{i \in I}$. Let f be some function s.t. for all $i \in I$, $f(X_i)$ is a convex combination of sources that have some property $\mathcal{P}$. Then $f(X)$ is a convex combination of sources that have property $\mathcal{P}$.*

A.2.4 Conditional entropy.

If $X = X_1 \circ \cdots \circ X_{\mathsf{t}}$ is a random variable (not necessarily a block source) over $\{0,1\}^n$ divided into t blocks in some way, and $x_1, \ldots, x_i$ are some strings with $0 \leq i < \mathsf{t}$, we use the notation $X|x_1, \ldots, x_i$ to denote the random variable X conditioned on $X_1 = x_1$,. . .,$X_i = x_1$. For $1 \leq i < j \leq \mathsf{t}$, we denote by $X_{i,ldots,j}$ the projection of X into the blocks $X_i, \ldots, X_j$. We have the following facts about such sources:

Lemma A.20 (Typical prefixes)**.** *Let $X = X_1 \circ \cdots \circ X_{\mathsf{t}}$ be a random variable divided into t blocks, let $X' = X|A$ be a deficiency d subsource of X, and let ℓ be some number. Then for every $1 \leq i \leq \mathsf{t}$, with probability at least $1 - 2^{-\ell}$, a random prefix $x_1, \ldots, x_i$ in X' satisfies $\Pr[X' \in A|x_1, \ldots, x_i] \geq 2^{-d-\ell}$.*

Proof. We denote by X^1 the first i blocks of X. Let B be the event over X^1 that $\Pr[X \in A|X^1] < 2^{-d-\ell}$. We need to prove that $\Pr[B|A] < 2^{-\ell}$ but this follows since $\Pr[B|A] = \frac{\Pr[A\cap B]}{\Pr[A]} \leq 2^d \Pr[A \cap B]$. However $\Pr[A \cap B] \leq \Pr[A|B] = \sum_{x \in B} \Pr[A|X^1 = x] Pr[X^1 = x|B] < 2^{-d-\ell}$. ☐

As a corollary we get the following

Corollary A.21 (Subsource of block sources). *Let $X = X_1 \circ \cdots \circ X_{\mathsf{C}}$ be a k-entropy C-block source (i.e., for every $x_1, \ldots, x_i \in \mathsf{Supp}(X_{1,\ldots,i})$, $H_\infty(X_{i+1}|X_{1,\ldots,i} = x_1, \ldots, x_i) > k$) and X' be a deficiency d subsource of X. Then X' is $\mathsf{C}2^{-d}$ statistically close to being a $k - 2d$ block source.*

Proof. Let $X' = X|A$ and define B to be following the event over X': $x = x_1, \ldots, x_{\mathsf{C}} \in B$ if for some $i \in [\mathsf{C}]$, $\Pr[X \in A|x_1, \ldots, x_i] < 2^{-2d}$. By Lemma A.20, $\Pr[X' \in B] < \mathsf{C}2^{-d}$. However, for every $x = x_1, \ldots, x_{\mathsf{C}} \in \bar{B} = A \setminus B$, we get that $Y' = X'_{i+1}|x_1, \ldots, x_{i-1}$ is a deficiency d subsource of $Y = X_{i+1}|x_1, \ldots, x_{i-1}$ and hence $H_\infty(Y') \geq H_\infty(Y) - 2d \geq k - 2d$. Hence $X'|\bar{B}$ is a $k - 2d$-block source of distance $\mathsf{C}2^{-d}$ from X'. ☐

If $X = X_1 \circ \cdots \circ X_{\mathsf{t}}$ is a source divided into t blocks then in general, the entropy of X_i conditioned on some prefix $x_1, \ldots, x_{i-1}$ can depend on the choice of prefix. However, the following lemma tells us that we can restrict to a low deficiency subsource on which this entropy is always roughly the same, regardless of the prefix. Thus we can talk about the conditional entropy of a block X_i without referring to a particular prefix of it.

Lemma A.22 (Fixing entropies). *Let $X = X_1 \circ X_2 \circ \cdots \circ X_{\mathsf{t}}$ be a t-block random variable over $\{0,1\}^n$, and let $0 = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_c = n$ be some numbers. Then, there is a deficiency $\mathsf{t}^2 \log c$ subsource X' of X and a sequence $\bar{e} = e_1, \ldots, e_{\mathsf{t}} \in [c]^{\mathsf{t}}$ such that for every $0 < i \leq \mathsf{t}$ and every sequence $x_1, \ldots, x_{i-1} \in \mathsf{Supp}(X'_{1,\ldots,i-1})$, we have that*

$$\tau_{e_i} \leq H_\infty(X'_i|x_1, \ldots, x_{i-1}) \leq \tau_{e_{i+1}} \tag{3}$$

Proof. We prove this by induction. Suppose this is true for up to $\mathsf{t} - 1$ block and we'll prove it for t blocks. For every $x_1 \in \mathsf{Supp}(X_1)$ define the source $Y(x_1)$ to be $X_{2,\ldots,i}|x_1$. By the induction hypothesis there exists a $(\mathsf{t}-1)^2 \log c$ deficiency subsource $Y'(x_1)$ of $Y(x_1)$ source and $\bar{e}(x_1) \in [c]^{\mathsf{t}-1}$ the sequence such that $Y'(x_1)$ satisfies Equation 3 with respect to $\bar{e}(x_1)$. Define the function $F : X_1 \to [c]^{\mathsf{t}-1}$ that maps x_1 to $\bar{e}(x_1)$ and pick a subsource X'_1 of X_1 of deficiency $(\mathsf{t}-1)\log c$ such that F is constant on X'_1. That is, there are some values $e_2, \ldots, e_{\mathsf{t}} \in [c]^{\mathsf{t}-1}$ such that $F(x_1) = e_2, \ldots, e_{\mathsf{t}}$ with probability 1. We let the source X' be X conditioned on the event that for $x_1, \ldots, x_{\mathsf{t}} \in X'$, $x_1 \in X'_1$ and $x_2, \ldots, x_{\mathsf{t}} \in Y(x_1)$.

The deficiency of X' is indeed at most $(\mathsf{t}-1)\log c + (\mathsf{t}-1)^2 \log c < \mathsf{t}^2 \log c$. ☐

A.3 Some results from previous works.

We'll use the following results from some previous works.

Theorem A.23 ([Tre01, RRV02]). *For every $n, k \in \mathbb{N}$, $\epsilon > 0$, there is an explicit (n, k, ϵ)-strong extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{k - O(\log^3(n/\epsilon))}$ with $d = O(\log^3(n/\epsilon))$.*

Theorem A.24 ([CG88, Vaz85]). *For all $n, \delta > 0$, there exists a polynomial time computable strong extractor $\mathsf{Vaz} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ with $m = \Omega(n)$ and error $\epsilon = 2^{-\Omega(n)}$.*

Ran Raz constructed an extractor that can extract when only one source is required to have min-entropy rate greater than half.

Theorem A.25. *[Raz05] For any n_1, n_2, k_1, k_2, m and any $0 < \delta < 1/2$ s.t. ,*

- $n_1 \geq 6 \log n_1 + 2 \log n_2$

- $k_1 \geq (0.5+\delta)n_1 + 3\log n_1 + \log n_2$
- $k_2 \geq 5\log(n_1 - k_1)$
- $m \leq \delta \min[n_1/8, b_2/40] - 1$

There is a polynomial time computable function $\mathsf{Raz} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ *s.t. if X is an (n_1, k_1) source and Y is an independent (n_2, k_2) source,*

$$|X \circ \mathsf{Raz}(X,Y) - X \circ U_m| < \epsilon$$

and

$$|Y \circ \mathsf{Raz}(X,Y) - Y \circ U_m| < \epsilon$$

where $\epsilon = 2^{-1.5m}$.

Recently Jean Bourgain constructed a strong 2-source extractor for min-entropy rate slightly less than half.

Theorem A.26 ([Bou05])**.** *There exists a polynomial time computable function* $\mathsf{Bou} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *and a universal constant α_0 s.t. if X, Y are independent $(n, (1/2 - \alpha_0)n)$ sources,*

$$|Y \circ \mathsf{Bou}(X,Y) - Y \circ U_m| \leq 2^{-\Omega(n)}$$

where $m = \Omega(n)$ and U_m is independent of Y.

Another construction we will use as a subroutine is the following extractor for the case of two independent aligned SR-sources by Rao.

Theorem A.27 ([Rao05])**.** *For every constant $\gamma < 1$ and n, n', t with $t = t(n, n')$ s.t. $t < n^\gamma$ and $t < n'^\gamma$ there exists a constant $\alpha < 1$ and a polynomial time computable function* $\mathsf{2SRExt} : \{0,1\}^{tn} \times \{0,1\}^{tn'} \to \{0,1\}^m$ *s.t. if X is a $(t \times n)$ SR-source and Y is an independent aligned $(t \times n')$ SR-source,*

$$|Y \circ \mathsf{2SRExt}(X,Y) - Y \circ U_m| < \epsilon$$

and

$$|X \circ \mathsf{2SRExt}(X,Y) - X \circ U_m| < \epsilon$$

where U_m is independent of X, Y, $m = \min[n, n'] - \min[n, n']^\alpha$ and $\epsilon = 2^{-\min[n,n']^{\Omega(1)}}$.

Some useful facts about extractors. We'll use the following simple facts about seeded and two-source extractors.

Proposition A.28. *Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a seeded (n, k, ϵ) strong extractor. Let X be any (n, k) source. Let $\{0,1\}^d = \{s_1, s_2, \ldots, s_{2^d}\}$. Then* $\mathsf{Ext}(X, s_1) \circ \mathsf{Ext}(X, s_2) \circ \cdots \circ \mathsf{Ext}(X, s_{2^d})$ *is ϵ-close to a $(2^d \times m)$ SR-source.*

Proof. Ommitted. □

Lemma A.29. *Let A and B be distributions on $\{0,1\}^n$ that are $1/100$-close to uniform. Let* $\mathsf{Vaz} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *be a 2-source extractor with entropy threshold $2n/3$ and error 2^{-2m} and let $c \in \{0,1\}^m$ be some fixed value. Then*

1. $\Pr[\mathsf{Vaz}(A,B)=c] \geq 2^{-2m}$.

2. There exist $a,b \in \{0,1\}^n$ *such that* $\mathsf{Vaz}(a,b)=c$, $\Pr[A=a] > 2^{-2n}$ *and* $\Pr[B=b] > 2^{-2n}$.

Proof. By Lemma A.14, there are deficiency 2 subsources $A' \subset A$, $B' \subset B$ with min-entropy $n-3$ each.
Then we get that $\Pr[\mathsf{Vaz}(A',B')=c] \geq 2^{-m} - 2^{-10m}$.
Thus to see the first statement, note that

$$\begin{aligned}
&\Pr[\mathsf{Vaz}(A,B)=c] \\
&\geq \Pr[\mathsf{Vaz}(A,B)=c | A \in \mathsf{Supp}(A'), B \in \mathsf{Supp}(B')] \Pr[A \in \mathsf{Supp}(A'), B \in \mathsf{Supp}(B')] \\
&\geq (1/2^{-4}) \Pr[\mathsf{Vaz}(A,B)=c | A \in \mathsf{Supp}(A'), B \in \mathsf{Supp}(B')] \\
&= (1/16) \Pr[\mathsf{Vaz}(A',B')=c] \\
&\geq 2^{-2m}
\end{aligned}$$

The second statement follows from the first. To see this, suppose the second claim is not true, then for every $a,b \in \{0,1\}^n$ with $\mathsf{Vaz}(a,b)=c$ we have that $\Pr[A=a] \leq 2^{-2n}$ or $\Pr[B=b] \leq 2^{-2n}$. This means that:

$$\begin{aligned}
\Pr[\mathsf{Vaz}(A,B)=c]& \\
&= \sum_{a \in A, b \in B \text{ s.t. } \mathsf{Vaz}(a,b)=c} \Pr[A=a, B=b] \\
&= \sum_{a \in A, b \in B \text{ s.t. } \mathsf{Vaz}(a,b)=c} \Pr[A=a] \Pr[B=b] \\
&\leq 2^{-2n}
\end{aligned}$$

This contradicts the first statement.

□

B Extractor for one block source and one general source

In this section we construct an extractor that works for two sources, given an assumption on one of the sources. The assumption is that the first source is a *block wise* source, which means that it is divided into C blocks such that each block has entropy above a certain threshold *even conditioned on all previous blocks*. As mentioned in the introduction, block sources turn out to be very useful in many settings in derandomization. We prove several variants of this extractor, the variant that will be useful for the construction of our two source disperser is the following:

Theorem B.1 (Block vs General Resilient Subsource Extractor)**.** *For every* n, n', k, *with* $k > \log^{10}(n+n')$ *there exists a polynomial time computable function* $\mathsf{BExt} : \{0,1\}^{\mathsf{C}n} \times \{0,1\}^{n'} \to \{0,1\}^m$ *with* $\mathsf{C} = O(\frac{\log n}{\log k})$ *s.t. , if* $X = X^1 \circ \cdots \circ X^{\mathsf{C}}$ *is a* k*-block source and* Y *is an independent* k*-source*

High Error on a Subsource *There is a deficiency* 2 *subsource* $Y' \subseteq Y$ *s.t.*

$$|\mathsf{BExt}(X,Y') - U_m| < 2^{-k^{\Omega(1)}}$$

Resilience *For all deficiency* $k/10$ *subsources* $\hat{X} \subseteq X$, $\hat{Y} \subseteq Y'$, *the above error bound still hold.*

With all constants in $O(\cdot)$ *and* $\Omega(\cdot)$ *notations being absolute constants independent of* n *and* k.

B.1 Other results.

Theorem B.1 is actually a corollary of the following more general theorem:

Theorem B.2 (Block vs General Resilient Subsource Extractor)**.** *There exist absolute constants $c_1, c_2, c_3 > 0$ such that for every n, n', k, k', with $k, k' > \log^{10}(n + n')$ there exists a polynomial time computable function* $\mathsf{BExt} : \{0,1\}^{\mathsf{C}n} \times \{0,1\}^{n'} \to \{0,1\}^m$ *with* $\mathsf{C} = c_1 \frac{\log n}{\log \hat{k}}$ *s.t. , if $X = X^1 \circ \cdots \circ X^{\mathsf{C}}$ is a k block source and Y is an independent (n', k')-source,*

High Error on a Subsource *There is a deficiency 2 subsource $Y' \subseteq Y$ s.t.*

$$|X \circ \mathsf{BExt}(X, Y') - X \circ U_m| < \epsilon$$

$$|Y \circ \mathsf{BExt}(X, Y') - Y' \circ U_m| < \epsilon$$

where U_m is independent of X and Y, and for $\hat{k} = \min\{k, k'\}$, $m = c_2 \hat{k}$ and $\epsilon = 2^{-\hat{k}^{c_3}}$.

Resilience *For all deficiency $\hat{k}/10$ subsources $\tilde{X} \subseteq X$, $\tilde{Y} \subseteq Y'$, the above error bounds still hold.*

To prove Theorem B.2, we prove several new extractors, some of which could be of independent interest. We state the results proven in this section below.

Theorem B.3 (Somewhere random vs Block Source Extractor)**.** *For every $t, n', k = k(t, n'), k' = k'(t, n)$, with $k, k' > \log^{10} t, k, k' > \log^{10} n$ there exists constants $\alpha, \beta < 1$ and a polynomial time computable function* $\mathsf{BExt} : \{0,1\}^{\mathsf{C}n} \times \{0,1\}^{n'} \to \{0,1\}^m$ *with* $\mathsf{C} = O(\frac{\log t}{\log k})$ *s.t. , if $X = X^1 \circ \cdots \circ X^{\mathsf{C}}$ is a $(k', \ldots, k')$ block source and Y is an independent $(t \times k)$ $(k - k^{\beta})$-SR-source,*

$$|X \circ \mathsf{SRvsBExt}(X, Y) - X \circ U_m| < \epsilon$$

$$|Y \circ \mathsf{SRvsBExt}(X, Y) - Y \circ U_m| < \epsilon$$

where U_m is independent of X and Y, $m = \hat{k} - \hat{k}^{\alpha}$, $\epsilon = 2^{-\hat{k}^{\Omega(1)}}$ and $\hat{k} = \min\{k, k'\}$.

Corollary B.4 (Block vs General Source Extractor)**.** *For every $n, n', k = k(n, n'), k' = k'(n, n')$ with $k, k' > \log^{10} n'$, $k, k' > \log^{10} n$, there exists a constant α and a polynomial time computable function* $\mathsf{BExt} : \{0,1\}^{\mathsf{C}n} \times \{0,1\}^{n'} \to \{0,1\}^m$ *with* $\mathsf{C} = O(\frac{\log n}{\log k})$ *s.t. , for constant $\gamma < 1$ if $X = X^1 \circ \cdots \circ X^{\mathsf{C}}$ is a $(k', \ldots, k')$ block source and Y is an independent (n, k) source,*

$$|X \circ \mathsf{BExt}(X, Y) - X \circ U_m| < \epsilon$$

$$|Y \circ \mathsf{BExt}(X, Y) - Y \circ U_m| < \epsilon$$

where U_m is independent of X and Y, $m = \hat{k} - \hat{k}^{\alpha}$,$\epsilon = 1/n^{\Omega(1)}$ and $\hat{k} = \min\{k, k'\}$.

Proof. This corollary follows from the theorem and Proposition A.28 (plugging in a seeded extractor which requires only $O(\log n)$ bits of seed and gives polynomially small error).

□

B.2 The block extractor

Here we will construct the extractor for Theorem B.3. We will obtain our extractor by iterated condensing of SR-sources. In each condensing step, we will consume one block of X to reduce the number of rows of the SR-source by a factor of $k^{\Omega(1)}$. Thus after $O(\log n/\log k)$ steps, we will have reduced the number of rows to just 1, at which point extraction becomes trivial. To build our condenser, we first build a strong extractor for one SR-source and one general source, when the SR-source has very few rows. Then we use this strong extractor as a black box to build our condenser. The basic extractor we will construct is captured by the following theorem:

Theorem B.5. *For every $n', k'(n), k(n)$ with $k', k > \log^{10} n, k' > \log^4 k, k > \log^4 k'$ and constants $0 < \gamma < 1/2$, there exists a constant $\beta < 1$ and a polynomial time computable function* $\mathsf{BasicExt} : \{0,1\}^{n'} \times \{0,1\}^{k^{\gamma+1}} \to \{0,1\}^m$ *s.t. if X is an (n', k') source and Y is a $(k^\gamma \times k)$ $(k - \hat{k}^\beta)$-SR-source,*

$$|Y \circ \mathsf{BasicExt}(X, Y) - Y \circ U_m| < \epsilon$$

and

$$|X \circ \mathsf{BasicExt}(X, Y) - X \circ U_m| < \epsilon$$

where U_m is independent of X, Y, $m = \hat{k} - \hat{k}^{\Omega(1)}$, $\epsilon = 2^{-\hat{k}^{\Omega(1)}}$ and $\hat{k} = \min\{k, k'\}$.

We will first discuss how to build our condenser assuming we had the above theorem. Then we will show how to give the extractor for the above theorem.

B.3 The condenser

Given $\mathsf{BasicExt}$ as promised by Theorem B.5, we will build a condenser for the input distribution. In each step we will use a block from X to condense the SR-source Y.

Construction: $\mathsf{Cond}(x, y)$

Set $\gamma \ll 1/2$ to some constant value. Let β be as in the hypothesis of Theorem B.3.

For these γ, β, let $\mathsf{BasicExt}$ be the function promised by Theorem B.5. Let m, ϵ be the output length and error of $\mathsf{BasicExt}$ respectively.

Input: $x = x^1 \circ x^2 \circ \cdots \circ x^{\mathsf{C}}$, a sample from a block source and y a sample from a $(t \times k)$ somewhere random source.

Output: $z = x^2 \circ x^3 \circ \cdots \circ x^{\mathsf{C}}$ and y' a $((t/k^\gamma) \times m)$ sample that we will claim comes from a somewhere random source.

1. Partition the t rows of y equally into t/k^γ parts, each containing k^γ rows. Let y_j denote the j'th such part.
2. For all $1 \le j \le t/k^\gamma$, let $y'_j = \mathsf{BasicExt}(x^1, y_j)$.
3. Let y' be the string with rows $y'_1, y'_2, \ldots, y'_{t/k^\gamma}$.

Given $X = X^1 \circ \cdots \circ X^{\mathsf{C}}$ and Y, the above algorithm uses X^1 to condense Y. Even though this introduces dependencies between X and Y, once we fix X^1, the two output distributions are once again independent. Formally we will argue that after applying the condenser, output random variables Z and Y' above are statistically close to a convex combination of independent sources, where Z is a block source with one less block than X, and Y' is an SR-source with much fewer rows than Y.

Lemma B.6. *Let X, Y be as above. Let ϵ be the error of* $\mathsf{BasicExt}$. *Then $Z = X^2 \circ \ldots \circ X^{\mathsf{C}}$ and Y' are $2\sqrt{\epsilon}$-close to a convex combination of sources where each source in the combination has*

1. *Z is a $(k', \ldots, k')$ block source*
2. *Y' is a $(t/k^\gamma, m)$ somewhere random source*
3. *Z is independent of Y*

Proof. Let $h \in [t/k^\gamma]$ be such that Y_h contains the random row. Consider the random variable X^1. We will call x^1 *good* if $|\mathsf{BasicExt}(Y_h, x^1) - U_m| < \sqrt{\epsilon}$, where m, ϵ are the output length and error of $\mathsf{BasicExt}$ respectively.

Then we make the following easy claims:

Claim B.7. *For good x^1,*

1. *$Z|X^1 = x^1$ is a $(k', \ldots, k')$ block source*
2. *$Y'|X^1 = x^1$ is a $\sqrt{\epsilon}$-close to being a $((t/k^\gamma) \times m)$ somewhere random source*
3. *$Z|X^1 = x^1$ is independent of $Y'|X^1 = x^1$*

Claim B.8. $\Pr[X^1$ *is not* good $] < \sqrt{\epsilon}$

These two claims clearly imply the lemma.

□

B.4 Proofs of the previous theorems

Armed with Lemma B.6, we can now prove the theorems.

Proof of Theorem B.3. We apply Cond repeatedly until the second source just contains one row. At this point we use the one row with Raz's extractor from Theorem A.25 with X to get the random bits.

To see that the bits obtained in this way are strong, first note that Raz's extractor is strong in both inputs. Let O be the random variable that denotes the output of our function $\mathsf{BExt}(X, Y)$. Let Q denote the concatenation of all the blocks of X that were consumed in the condensation process. Let U_m denote a random variable that is independent of both X, Y. Then we see that these variables satisfy the hypothesis of Lemma A.18, i.e. on fixing Q to a good value, Raz's extractor quarantees that the output is independent of both inputs, thus we must have that the output is close to being independent of both inputs. The dominant error term in BExt comes from the first step, when we convert Y to an SR-source.

□

Proof of Corollary B.4. Our first step will be to convert the general source Y into an SR-source. To do this we will use a strong seeded extractor and the Proposition A.28.

If we use a strong seeded extractor that requires only $O(\log n)$ bits of seed, the SR-source that we get will have only $\mathrm{poly}(n)$ rows. This adds a polynomial amount of error. Then we simply apply Theorem B.3. □

Proof of Theorem B.2. Set $\hat{k} = \min\{k, k'\}$.

Our construction for Corollary B.4 already satisfies the first conclusion of the theorem. The proof for this follows from LemmaA.14. After we convert Y to an SR-source with polynomially small error, we can go to a deficiency 2 subsource $Y' \subseteq Y$ which has high entropy in some row. This is good enough to use our extractor from Theorem B.3 and get the better error.

The resilience property is more tricky. We already showed that our construction is $\hat{k}^{\Omega(1)}$-resilient, but we need linear resilience. It is already highly resilient with respect to X, but resilience with respect to Y is not clear. The reason we don't get that high of a resilience for Y is that in our construction Theorem B.5 we will need to take a slice of the bits of $\vec{Y}$. If we had to guarantee high resilience, the slice we would need to take would have to be two large for the construction to succeed.

To get the high resilience we will have to modify our construction slightly. What we will do is modify our extractor from Corollary B.4.

As in that construction, we will first convert Y to an SR-source $\vec{Y}$ by using a strong seeded extractor and Proposition A.28. Next we will consume a single block from the block source to encode $\vec{Y}$ in a way that will guarantee the high resilience:

Construction: $\mathsf{Res}(x, \vec{y})$

Input: $x = x^1 \circ \cdots \circ x^{\mathsf{C}}$, a sample from an $(k', k', \ldots, k')$ blocksource and y' a sample from a $(t \times k)$ SR-source.

Output: $z = x^2 \circ \cdots \circ x^{\mathsf{C}}$ and q

Let Raz be as in Theorem A.25, setup to output $\Omega(\hat{k})$ bits when given an (n', k') source and an independent $(k, 0.6k)$ source.

1. Let $q = \mathsf{Raz}(\vec{y}, x^1)$, i.e. q is the sample from the SR-source obtained by running Raz with every row of $\vec{y}$ with x^1.
2. Output z, q.

Now that we have these two sources, our final output is obtained by applying the extractor from Theorem B.3 to Z, Q.

Claim B.9. *Let $X, \vec{Y}'$ be sources as above. Then let Z, Q be the results of applying our construction. Then we have that Z, Q are a convex combination of independent sources over the variable X^1, where for each source in the combination:*

- *Z is independent of Q*
- *Z is a $(k', \ldots, k')$ blocksource*
- *Q is $2^{-\hat{k}^{\Omega(1)}}$-close to a $(t \times \hat{k})$ SR-source*

Proof. Recall that $Y' \subseteq Y$ is a deficiency 2 subsource of Y such that $\vec{Y}'$ has extremely high entropy in one row. When we condition on X^1 the first two claims easily hold. The last one holds since Raz is a strong extractor with that kind of error. $\square$

So what did we gain by doing this? We have gained in the resilience.

Claim B.10. *We can setup our extractor so that for every $\hat{k}/10$ deficiency subsource $\hat{X} \subseteq X$, $\hat{Y} \subseteq Y'$, our extractor succeeds with error $2^{-\hat{k}^{\Omega(1)}}$.*

Proof. We simply setup the extractor assuming that we have slightly lower entropy $0.7k'$ in X. All the previous conclusions still hold with this new k'. However, now when the extractor is given $\hat{X}, \hat{Y}$ we have that $\hat{X}$ is still $\mathsf{C}2^{-\Omega(k')}$ close to a block source source with min-entropy $0.7k'$ by Lemma A.21).

$\vec{\hat{Y}}$ is still an SR-source with entropy rate at least 0.6 in its high entropy row (since it lost at most $0.1k$ entropy in the high row by Fact A.8). $\square$

These two claims conclude the proof of the theorem.

$\square$

B.5 Extractor for general source and a SR-source with few rows

Here we will show how to extract from an SR-source and an independent general source when the SR-source has very few rows. We will prove Theorem B.5.

We are trying to build an extractor that can extract from one $(k^\gamma \times k)$ k^β-SR-source Y and an independent (n', k') source X. We will reduce this to the case of two independent aligned SR-sources with few rows, for which we can use Theorem A.27. Throughout this section, we will use $\hat{k}$ to denote $\min\{k, k'\}$.

The plan is to use the structure in the SR-source Y to impose structure on the source X. Our goal will be to first use Y and X to get a list of candidate seeds, such that one seed in the list is close to uniformly random and independent of botch X and Y. Once we have this list, we can readily reduce the problem to that of extracting from independent aligned SR-sources with few rows.

In the following discussion, the term *slice* refers to a subset of the bits coming from an SR-source that takes a few bits of the SR-source from every row. To generate the list of seeds that we want, we will first take a small slice of the bits from Y, i.e. we take the first w bits of every row from Y, where w is a parameter that we will pick later (think of w as $\hat{k}^\mu$ for small μ). We will be able to guarantee that at least one of these rows has high entropy. Then we can use Raz's extractor Theorem A.25 with these bits to extract from X. This turns X into a $(k^\gamma \times w')$ SR-source X', where $w' = \hat{k}^{\theta(1)} \gg w$ is some parameter that we will pick later. The two sources that we have now (X' and Y) are not independent, but note that when we fix the slice of bits that we used, we get two independent sources. Y conditioned on the slice could potentially lose entropy in its high entropy row. Still we can expect this high entropy row to have about $k - k^\beta - wk^\gamma$ bits of entropy, since we fixed only wk^γ bits of Y. In the next step we take two slices, X' and another slice of Y and use these to get our target list of candidate seeds. We will guarantee that one seed in our list is independent of both X and Y! Once we have this list, we can use it with a traditional strong seeded extractor to extract from X and Y to get independent aligned SR-sources of the type that Theorem A.27 can handle.

Now we will describe the construction in more detail. Here is our extractor:

Construction: $\mathsf{Red}(x, y)$

Input: x, a sample from an (n', k') source and y a sample from a $(k^\gamma \times k)$ k^β-somewhere random source.

Output: z

Let w, w', l be parameters that we will pick later. We will need to ensure that $w' \gg w$.

Let Raz be the extractor from Theorem A.25 setup to extract w' bits from an (n', k') source, using a $(w, 0.9w)$ source as seed.

Let Vaz be the extractor from Theorem A.24, setup to extract from two $(w', w' - w - l)$ sources (we will choose $w' \gg w$).

Let $\mathsf{2SRExt}$ be the extractor from Theorem A.27.

Let Ext be the strong seeded extractor from Theorem A.23.

1. First take a slice of y of width w, i.e. take the first w bits of every row of y. Call the slice s.
2. Treating s as a list of k^γ seeds, use it to extract from x to get $s_1 = \mathsf{Raz}(x, \vec{s})$. We will set up Raz to output just w' bits in each invocation. The result is a string with k^γ rows, each of length w'.
3. Next take a slice of y of width w'. Call this slice s_2.
4. Let $\overline{s} = \mathsf{Vaz}(\vec{s_1}, \vec{s_2})$, i.e. we apply Vaz to the first seed from s_1 and the first seed from s_2 to get the first seed of $\overline{s}$ and so on. Thus $\overline{s}$ has k^γ rows.
5. Let $x' = \mathsf{Ext}(x, \vec{\overline{s}}), y' = \mathsf{Ext}(y, \vec{\overline{s}})$.
6. Use $\mathsf{2SRExt}$ to get $z = \mathsf{2SRExt}(x', y')$.

We will prove the following lemma:

Lemma B.11. *Given any (n, k) source X and a somewhere random source Y as above, we can pick w, w', l and a constant β s.t. $X' \circ Y'$ is $2^{-k^{\Omega(1)}}$-close to a convex combination of sources s.t. for any source in the combination:*

1. *X' is independent of Y'*
2. *X' is a $(k^{\gamma}/2 \times k' - k'^{\beta})$ SR-source*
3. *Y' is a $(k^{\gamma}/2 \times k - k^{\beta})$ SR-source*

Given the lemma, we have reduced the problem to one of extracting from aligned somewhere random sources. We know how to do this in a strong manner and so Theorem B.5 follows.

All that is left to do is to prove the lemma.

Proof of Lemma B.11. We assume that we have some fixed random variables X, Y that satisfy the hypotheses of the lemma. We will make several claims about the various random variables involved in the construction, setting w, w' along the way to ensure that our lemma is true. In the rest of this proof, a capital letter represents the random variable for the corresponding small letter in the construction above.

Recall that $\hat{k}^{\beta}$ (we are allowed to set $\beta < 1$ to anything we want) is the randomness deficiency of the random row in Y. Note that:

Claim B.12. *S is $2^{-\hat{k}^{\beta}}$ close to a $(k^{\gamma} \times w)$ $(w - 2\hat{k}^{\beta})$-SR-source*

Proof. This follows from an application of Lemma A.13. □

We will choose $w = \hat{k}^{\alpha_1}$ for some constant α_1 s.t. $\alpha_1 + \gamma < 1$ and $\alpha_1 \gg \beta$. Thus note that the random row of S is good enough to use as a seed for Raz's extractor. The result is S_1, a $(k^{\gamma} \times w')$ SR-source. We incur an error of $2^{-\Omega(w)} = 2^{-\hat{k}^{\alpha_1}}$ when we apply Raz. We will set $w' = \hat{k}^{\alpha_2}$, where $\alpha_2 > \alpha_1 + \gamma$ is another constant with $\alpha_2 + \gamma < 1$.

So w' is much larger than w, but much smaller than the total length of any of the SR-sources involved. We need the condition that $\alpha_2 + \gamma < 1$ to ensure that we can fix the bits in these seeds without losing all of the original entropy.

Now S_1 is correlated with both of our original sources. However, when we fix S, S_1 becomes independent of Y. Since Raz is a strong extractor, S_1 still contains a random row even when we fix S.

Claim B.13. *For* good *s, $(X \circ S_1)|S = s$ is independent of $Y|S = s$. Further $S_1|S = s$ is a $(k^{\gamma} \times w')$ SR-source.*

Also examining the error parameters of Raz shows that

Claim B.14. $\Pr[S \textit{ is good }] > 1 - 2^{-\Omega(w)}$

Thus with high probability we succeed and get a good list of seeds S_1 which is independent upto convex combinations from Y.

Now consider the random variable S_2.

Claim B.15. *S_2 is $2^{-\hat{k}^{\beta}}$ close to a $(k^{\gamma} \times w')$ $(w' - 2\hat{k}^{\beta})$-SR-source before we fixed S*

Proof. This follows from an application of Lemma A.13. □

Now we assume that S_2 is in fact a $w' - 2\hat{k}^{\beta}$-SR-source before we fixed S (by adding $2^{-\hat{k}\beta}$ to the final error).

After we fix S, S_2 can lose entropy in its random row, but not much. We can expect it to lose as many bits of entropy as there are in S, which is only $k^{\alpha_1 + \gamma}$. Since we picked $w' = k^{\alpha_2} \gg k^{\alpha_1 + \gamma}$, we get that S_2 still contains entropy.

Claim B.16. $\Pr_{s \leftarrow_R S}[S_2 | S = s$ *is a* $(k^\gamma \times w')$ $(w' - 2\hat{k}^\beta - l)$*-SR-source* $] > 1 - 2^{-l}$.

Proof. This is a straightforward application of Fact A.9. □

Setting $l = \hat{k}^\beta$, we get that this probability is at least $1 - 2^{-\hat{k}^\beta}$.

¿From the rest of the proof we assume that S is fixed to some good value s.

Note that S_2 is also aligned with S_1.

Thus our applcation of Vaz above succeeds. In the aligned good row, we give it two independent (after fixing S) sources which are statistically close to having extremely high entropy.

Now we are almost done. The result of Vaz is the random variable $\overline{S}$.

Claim B.17. $\overline{S}$ *is* $2^{-\Omega(w')}$ *close to a* $(k^\gamma, \Omega(w'))$ *SR-source.*

We also get that the random row of $\overline{S}$ is independent of both X and Y! Thus our final extraction step makes sense. Let $\overline{S}^1$ represent the random row of $\overline{S}$. Let α_3 be a small constant s.t. $2^{-\hat{k}^{100\alpha_3}}$ is the error of Ext. The final claims that we will is:

Claim B.18. *With high probability* $\overline{S}$ *contains a good seed to extract from each of the sources:*

$$\Pr_{\overline{s} \leftarrow_R \overline{S}}[|\mathsf{Ext}((Y|S = s, \overline{S} = \overline{s}), \overline{s}^1) - U_m| \geq 2^{-\hat{k}^{\alpha_3}}] < 2^{-\hat{k}^{\alpha_3}}$$

and

$$\Pr_{\overline{s} \leftarrow_R \overline{S}}[|\mathsf{Ext}((X|S = s, \overline{S} = \overline{s}), \overline{s}^1) - U_m| \geq 2^{-\hat{k}^{\alpha_3}}] < 2^{-\hat{k}^{\alpha_3}}$$

Sketch of proof. There are two ways in which the claim can fail. Either S, S_1, S_2 steal a lot of entropy from from X, Y, or they produce a bad seed in $\overline{S}$ to extract from $X|S_1 = s_1$ or $Y|S = s, S_2 = s_2$. Both events happen with small probability. Thus we can use the union bound to get our final estimate. □

This concludes the proof of the Lemma and hence the proof of Theorem B.5.

□

C A subsource somewhere extractor

The formal definition of a subsource somewhere extractor is the following:

Definition C.1 (subsource somewhere extractor)**.** A function $\mathsf{SSE} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to (\{0,1\}^m)^\ell$ is a *subsource somewhere extractor* with nrows output rows, *entropy threshold* (k_1, k_2), *deficiency* $(\mathsf{def}_1, \mathsf{def}_2)$, *resiliency* $(\mathsf{res}_1, \mathsf{res}_2)$ and *error* ϵ if for every (n_1, k_1)-source X and (n_2, k_2)-source Y there exist a deficiency def_1 subsource X^{good} of X and a deficiency def_2 subsource Y^{good} of Y such that for every deficiency res_1 subsource X' of X^{good} and deficiency res_2 subsource Y' of Y^{good}, the random variable $\mathsf{SSE}(X', Y')$ is ϵ-close to a $\ell \times m$ somewhere random distribution.

Whenever the subsource extractor is symmetric (meaning that $n_1 = n_2$, $k_1 = k_2$ and so on) we will only use one parameter for each pair and write that a extractor has entropy threshold k, deficiency def, resiliency res and error ϵ.

In this section, we prove the following theorem:

Theorem C.2 (A Subsource Somewhere Extractor)**.** *Let* $\delta > 0$ *be some constant. Then, for every* n *there exists a polynomial-time computable subsource somewhere extractor* $\mathsf{SSE} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{\mathsf{nrows} \cdot \mathsf{out}}$ *with* $k = n^\delta$ *entropy threshold,* $\mathsf{nrows} < \mathrm{polylog}(n)$ *output rows,* $\mathsf{out} > k^{1/10}$ *output length,* $\mathsf{def} = k/100$ *deficiency,* $\mathsf{res} = k^{1-1/\lg\lg k}$ *resiliency, and* $2^{k^{\Omega(1)}}$ *error.*

Name	Description	Restrictions	Notes
n	Input length		
k	Entropy	$k = n^\delta$	Assume δ is arbitrarily small constant.
def	Deficiency	$d = k/10$	Fixed to $k/10$ for simplicity. Use smaller k to get smaller deficiency.
res	Resiliency	$\mathsf{res} = \frac{k}{200n^\alpha \mathsf{t}}$	See def of α below. At least $k^{0.99}$. Is $k^{1-o(1)}$ for $\alpha = o(1)$.
out	Output length	$\mathsf{out} = \mathsf{res}^{1/5}$	See def of res below.
nrows	No. of output rows	$\mathsf{nrows} \leq (\log n)\mathsf{t}^\mathsf{C}$	See def of C, t below. At most $k^{0.1}$.
ϵ	Error	$\epsilon = 2^{k^{\Omega(1)}}$	Inherited from BExt.
C	No. blocks for BExt	$O(\log n/\log k)$	We always invoke BExt with entropy $\geq \sqrt{k}$
α	Resiliency parameter	$10\mathsf{C}\log(100\mathsf{C})/\log k < \alpha < \delta/100$	Smaller $\alpha \implies$ larger resiliency but more outputs. Typical setting: $\alpha = 1/\log\log n$.
t	Degree of partition tree	$\mathsf{t} = (100\mathsf{C})^{1/\alpha}$	

Table 1: Parameters used in construction of the subsource-somewhere extractor SSE.

C.1 Construction.

C.1.1 Parameters and precise statement.

We actually prove a more general theorem that implies Theorem C.2. We use the following parameters (see Table 1 for a summary of all parameters, including some mentioned later on):

- We let $k = n^\delta$ be the entropy of the input sources.
- Let $\mathsf{C} = O\left(\frac{\log n}{\log k}\right)$ be the number of blocks that the extractor BExt of Theorem B.3 requires to extract from $\sqrt{k}$ entropy. (See Corollary C.4 below for the precise parameters we use BExt for.)
- Let $10\mathsf{C}\log(100\mathsf{C})/\log k < \alpha < \delta/100$. One natural setting is $\alpha = 1/\log\log n$.
- Let $\mathsf{t} = (100\mathsf{C})^{1/\alpha}$. Note that our condition on α implies that $\mathsf{t} < k^{1/(10\mathsf{C})}$.

Lemma C.3 (Subsource Somewhere Extractor)**.** *For every n and $k = n^\delta$, let the parameters $\mathsf{t}, \mathsf{C}, \alpha$ be as above. There exists a somewhere subsource extractor $\mathsf{SSE} : \{0,1\}^{2n} \to \{0,1\}^{\mathsf{out}}$ with the following parameters:*

- *Number of output rows $\mathsf{nrows} \leq (\log n) \cdot \mathsf{t}^\mathsf{C}$. (Note that $\mathsf{nrows} < k^{0.1}$)*
- *Deficiency $\mathsf{def} = k/10$.*
- *Resiliency equal to res for $\mathsf{res} = \frac{k}{200n^\alpha \mathsf{t}}$. (Note that $\mathsf{res} > k^{0.99}$)*
- *Output length equal to $\mathsf{out} = \mathsf{res}^{1/5}$.*
- *Statistical error $2^{-k^{\Omega(1)}}$ where the constant exponent is at least $1/2$ times the exponent in the error of the extractor BExt of Lemma A.14.*

Proving Theorem C.2 from Lemma C.3. To obtain an extractor with the parameters of Theorem C.2, given n and $k = n^\delta$, simply invoke Lemma C.3 with parameters $k' = k/50$ and $\alpha = \frac{1}{2}\delta\log\log k$. We get deficiency $k/100$ and resiliency $\omega(k^{1-1/\log\log k})$ and output $k^{1/4}$. Since C is constant we get that the number of outputs is $\mathrm{polylog}(k) = \mathrm{polylog}(n)$.

C.1.2 Components

Block extractor. We'll use the following corollary of Lemma A.21.

Corollary C.4 (Block extractor.). *Let the parameters* $\mathsf{C}, n, k, \mathsf{res}, \mathsf{out}$ *be as above (with* $\mathsf{res} > k^{0.9}$ *and* $\mathsf{out} = \mathsf{res}^{1/5}$. *Then, there is a polynomial-time computable function* $\mathsf{BExt} : \{0,1\}^{\mathsf{C}n} \times \{0,1\}^n \to \{0,1\}^{\mathsf{out}}$ *satisfying:*

For every every independent sources $X \in \{0,1\}^{\mathsf{C}n}$ *and* $Y \in \{0,1\}^n$ *with* $H_\infty(Y) \geq 10\mathsf{res}$ *and* $X = X_1 \circ \cdots \circ X_{\mathsf{C}}$ *a* $10\mathsf{res}$ *block source,*[6] *there exist deficiency* $\sqrt{\mathsf{res}}$ *subsources* $X^{\mathsf{good}} \subseteq X$ *and* $Y^{\mathsf{good}} \subseteq Y$ *such that for every* $X' \subseteq X^{\mathsf{good}}$ *and* $Y' \subseteq Y^{\mathsf{good}}$ *with deficiency at most* res*,*

$$\left|\mathsf{BExt}(X', Y') - U_{\mathsf{out}}\right| < 2^{-k^{\Omega(1)}}$$

Seeded extractor. Let Ext be a seeded extractor that on inputs of length between at most n and seed of length $5\log n$ outputs a string of size $10\mathsf{out}$ such that for every source X with $H_\infty(X) \geq \mathsf{res}$, with probability at least 0.9 over $s \leftarrow_{\mathrm{R}} \{0,1\}^{5\log n}$,

$$\left|\mathsf{Ext}(X, s) - U_{10\mathsf{out}}\right| < 1/10$$

Two-source extractor for high min-entropy. We let $\mathsf{Vaz} : \{0,1\}^{10\mathsf{out}} \times \{0,1\}^{10\mathsf{out}} \to \{0,1\}^{\mathsf{out}}$ denote a two source extractor such that for every independent X, Y over $\{0,1\}^{10\mathsf{out}}$ with $H_\infty(X), H_\infty(Y) \geq 8\mathsf{out}$, we have that with probability at least $1 - 2^{-2\mathsf{out}}$ over $x \leftarrow_{\mathrm{R}} X$,

$$\left|\mathsf{Vaz}(x, Y) - U_{\mathsf{out}}\right| < 2^{-2\mathsf{out}}$$

Definition of response function. For every string x of length at most n, string $y \in \{0,1\}^n$, and $s_1, s_2 \in \{0,1\}^{5\log n}$ we define $\mathsf{Response}_{s_1,s_2}(x,y)$ to be $\mathsf{Vaz}(\mathsf{Ext}(x, s_1), \mathsf{Ext}(y, s_2))$. We define $\mathsf{Response}(x,y)$ to be the set $\{\mathsf{Response}_{s_1,s_2}(x,y)\}$ for all $s_1, s_2 \in \{0,1\}^{5\log n}$.

C.1.3 Operation of the algorithm.

On inputs $x, y \in \{0,1\}^n$, the subsource extractor SSE will do as follows:

(We assume without loss of generality that n is an integer power of t as this can be obtained by padding.)

1. We define a degree-t with depth $\log n/\log \mathsf{t} < \log n$ tree $\mathcal{T}_{n,\mathsf{t}}$ that we call the n,t *partition tree.* The nodes of $\mathcal{T}_{n,\mathsf{t}}$ are subintervals of $[1 \ldots n]$ defined in the following way:

   (a) The root of the tree is the interval $[1, n]$.

   (b) If a node v is identified with the interval $[a, b]$ then we let $v_1, \ldots, v_{\mathsf{t}}$ denote the t consecutive disjoint length-$|v|/\mathsf{t}$ subintervals of v. That is, $v_i = [a + (t-1)i, a + ti - 1]$. We let the i^{th} child of v be v_i.

   (c) The leaves of the tree are the length 1 intervals.

   For a string $x \in \{0,1\}^n$ and a set $S \subseteq [1, n]$ we'll denote by x_S the projection of x onto the coordinates of S. If v is a node in $\mathcal{T}_{n,\mathsf{t}}$ then x_v denotes the projection of x into the interval v.

2. On inputs $x, y \in \{0,1\}^n$, the algorithm SSE, working from the leaves upwards, will define for each node v in the tree $\mathcal{T}_{n,\mathsf{t}}$ an *associated set of values* $\mathsf{vals}(v)$ in the following way:

   (a) Each element in $\mathsf{vals}(v)$ is a pair of the form $\langle S, z\rangle$ where S is a subset of $[1, n]$ and $z \in \{0,1\}^{\mathsf{out}}$. It will always be the case that if $\langle S, z\rangle \in \mathsf{vals}(v)$ then S is the disjoint union of C equal length subintervals of v and $z = \mathsf{BExt}(x_S, y)$.

[6] That is, for every $i < \mathsf{C}$ and $x_1, \ldots, x_i \in \mathsf{Supp}(X_{1,\ldots,i})$, $H_\infty(X_{i+1}|x_1, \ldots, x_i) > 10\mathsf{res}$.

(b) If v is a leaf then $\mathsf{vals}(v)$ is the empty set.

(c) If v is not a leaf then $\mathsf{vals}(v)$ contains the following elements:

i. For each C-tuple $1 \leq i_1 < i_2 < \cdots < i_{\mathsf{C}} \leq \mathsf{t}$ let $S = v_{i_1} \cup v_{i_2} \cup \cdots \cup v_{i_{\mathsf{C}}}$ and add the pair $\langle S, \mathsf{BExt}(x_S, y)\rangle$ to $\mathsf{vals}(v)$.

ii. If v is not a leaf, let $v_1, \ldots, v_{\mathsf{t}}$ be its t children. Let R be the set $\mathsf{Response}(x_v, y)$. We say that v_i *passes* a test, denoted by $\mathsf{Test}_{x,y}(v_i) = \mathsf{pass}$, if there exists a pair $\langle S, z\rangle \in \mathsf{vals}(v_i)$ such that $z \notin R$. Otherwise, we say that $\mathsf{Test}_{x,y}(v_i) = \mathsf{fail}$. If there exists a child v_i of v that passes the test, then we let i_0 be the smallest such i and add to $\mathsf{vals}(v)$ all the elements of $\mathsf{vals}(v_{i_0})$.

3. The output of SSE is computed as follows: For every element $\langle S, z\rangle \in \mathsf{vals}([1, n])$ (i.e., in the list of values associated with the root of $\mathcal{T}_{n,\mathsf{t}}$) output z.

Sanity check: number of outputs. It's easy to verify on inputs $x, y \in \{0,1\}^n$ SSE does indeed output a list of at most $\mathsf{t}^{\mathsf{C}} \log n$ values. Indeed, the depth of the tree is less than $\log n$ and the number of outputs at a node is the number of outputs of one of its children plus t^{C} .

C.2 Analysis: Proof of Lemma C.3.

Recall that to show that SSE is a subsource somewhere extractor we need to come up with subsources $X^{\mathsf{good}} \subseteq X$, $Y^{\mathsf{good}} \subseteq Y$ satisfying the conditions of Definition C.1 with the appropriate parameters. The analysis proceeds in several steps. In each step we make a restriction on one or both of the input sources. When we're done, we'll get the desired subsources X^{good}, Y^{good}.

C.2.1 Step 1: Preprocess X.

The first step involves only the first source X. We'll restrict X to a subsource $\hat{X}$ that will have some attractive properties for us. We first prove a general lemma — Lemma C.5 — and then use it to prove in Lemma C.6 to show that we obtain the desired subsource $\hat{X}$.

Lemma C.5 (Two-types lemma.)**.** *Let X be a general k source over $\{0,1\}^n$ divided into t blocks $X = X_1 \circ \cdots \circ X_{\mathsf{t}}$. Let C be some positive integer and let $k' < k$ be some number. Then, there exists a subsource $X' \subseteq X$ of deficiency at most $d = \mathsf{C}k' + 2\mathsf{t}^2$ that satisfies one of the following properties:*
Either

Somewhere large source — one "large" block *There exists $i \in [\mathsf{t}]$ such that the first $i-1$ blocks of X' (namely $X'_1, \ldots, X'_{i-1}$) are* constant*, and $H_\infty(X_i) \geq k'$. We call such a source a k'-*somewhere large source.

or

Somewhere block-source — C "medium" blocks *There exist $0 < i_1 < i_2 < \cdots < i_{\mathsf{C}} \leq \mathsf{t}$ such that the first $i_1 - 1$ blocks of X' are constant for every $j \in [\mathsf{C}]$, and every $x_1, x_2, \ldots, x_{i_j - 1} \in \mathsf{Supp}(X'_{1\ldots i_j - 1})$,*

$$H_\infty\Big(X'_{i_j} | X'_{i_1} = x_1, \ldots, X'_{i_{j-1}} = x_{j-1}\Big) \geq k'/\mathsf{t}$$

We call such a source a $(\mathsf{C}, k'/\mathsf{t})$-somewhere block-source*.*

Proof. We let $\tau_0 = 0$, $\tau_1 = k'/\mathsf{t}$, $\tau_2 = k'$ and $\tau_3 = n$ and use Lemma A.22 to reduce X to a deficiency $4\mathsf{t}^2$ source X'' such that for every $i \in [\mathsf{t}]$ and every $x_1, \ldots, x_{i-1} \in \mathsf{Supp}(X''_{1,\ldots,i-1})$, the conditional entropy $H_\infty(X''_i | x_1, \ldots, x_{i-1})$ always falls into the same interval of $[0, k'/\mathsf{t}]$, $[k'/\mathsf{t}, k']$ and $[k', 2^n]$ regardless of the choice $x_1, \ldots, x_i$.

We call blocks where this conditional entropy falls into the interval $[0, k'/\mathsf{t})$ "small", blocks where this entropy falls into the interval $[k'/\mathsf{t}, k')$ "medium" and blocks where it is at least than k' "large". We divide to two cases: **Case 1:** if there are at most $\mathsf{C}-1$ medium blocks before the first large block, we let i be the position of the first large block and fix the first $i-1$ blocks to their most typical values. The conditional entropy X_1 given this prefix is still at least k'. Furthermore, since we fixed at most t small blocks and at most C medium blocks the overall deficiency is at most $(\mathsf{C}-1)k' + \mathsf{t}k'/\mathsf{t} = \mathsf{C}k'$. **Case 2:** If there are at least C medium blocks in the source, we let i be the position of the first medium block and fix the first $i-1$ blocks to their most typical value. All medium blocks remain medium conditioned on this prefix and the entropy we lose is at most $\mathsf{t}k'/\mathsf{t} \le k'$. $\square$

We'll now use Lemma C.5 to show that we can restrict the input source X to a subsource X^{sb} (for "somewhere block") satisfying some attractive properties:

Lemma C.6. *Let X be a source over $\{0,1\}^n$ with min-entropy k. Let $\mathsf{C}, \mathsf{t}, \alpha$ be values satisfying $\mathsf{t} \ge (10\mathsf{C})^{1/\alpha}$. Then, there exists a deficiency $k/10 + 4\mathsf{t}^2 \log n$ subsource X^{sb} of X and a vertex $v^{\mathsf{b}} = [a,b]$ of $\mathcal{T}_{n,\mathsf{t}}$ with the following properties:*

- *The source $X^{\mathsf{sb}}_{v^{\mathsf{b}}}$ is a $(\mathsf{C}, \frac{k}{10\mathsf{tC}n^{\alpha}})$-somewhere block source.*
- *There is a value $x \in \{0,1\}^{a-1}$ such that $X^{\mathsf{sb}}_{[1,a-1]} = x$ with probability 1.*

Proof. We'll prove the lemma by induction on $\lceil \log(n/k) \rceil = \lceil \log n - \log k \rceil$. If $n = k$ then this is the uniform distribution and everything is trivial. We invoke Lemma C.5 with parameter $k' = k/(10\mathsf{C})$ to obtain a deficiency $k/10 + 4\mathsf{t}^2$ subsource X' that is either k'-somewhere large or $(\mathsf{C}, k'/\mathsf{t})$-somewhere block source.

If X' is $(\mathsf{C}, k'/\mathsf{t})$-somewhere block source then since $k'/\mathsf{t} = k/(10\mathsf{tC})$ we have that $X'_{[1,n]}$ satisfies the first property. The second property holds trivially for $v^{\mathsf{b}} = [1,n]$.

The second option is that X' is a k'-somewhere large source. We let i be the index of the large block of entropy k', and let v_i be the interval $[(i-1)t+1, it]$. Note that $X'_{1,(i-1)t}$ attains some fixed value with probability 1. Let $n' = |v_i| = n/\mathsf{t}$. Since $\frac{n'}{k'} = \frac{n}{k}\frac{10\mathsf{C}}{\mathsf{t}} < \frac{n}{4k}$ we have that $\log(n'/k') < \log(n/k) - 2$ and so can assume by the induction hypothesis that the statement holds for the source $Z = X'_{v_i}$. This means that we have a subsource $Z' = Z|A$ of deficiency $k'/10 + 4\mathsf{t}^2 \log n'$ of Z and a node v'^{b} in the tree $\mathcal{T}_{n',\mathsf{t}}$ such that $Z'_{v'^{\mathsf{b}}}$ is a $(\mathsf{C}, \frac{k'}{10\mathsf{tC}n'^{\alpha}})$-somewhere block source. Substituting $k/(10\mathsf{C})$ for k' and n/t for n' we get that the large block has at least $\frac{k}{(10\mathsf{tC})n^{\alpha}} \cdot \frac{\mathsf{t}^{\alpha}}{10\mathsf{C}}$ entropy. However, since $\mathsf{t}^{\alpha} \ge 10\mathsf{C}$ this is sufficient to meet our requirements.

We define $X^{\mathsf{sb}} = X'|A$ (that is, the natural extension of the subsource Z' to a subsource of X'). Since $\Pr[A] \ge 2^{-k/10 + 4\mathsf{t}^2 \log n'}$ and since $\log n' \le \log n - 1$, this source has $k/4 + 4\mathsf{t}^2(\log n - 1)$ deficiency as a subsource of X' and hence at most $k/4 + 4\mathsf{t}^2 \log n$ deficiency as a subsource of X. If $v'^{\mathsf{b}} = [a', b']$ we let $v^{\mathsf{b}} = [(i-1)t + a', (i-1)t + b']$. It is clear the X^{sb} and v^{b} satisfy both our properties. $\square$

Result of Step 1. We apply Lemma C.6 to the input source X with our parameters k, t, α as chosen in Section C.1.1. We obtain a deficiency $k/9$ subsource (since $4\mathsf{t}^2 = o(k)$) X^{sb} of X, a node $v^{\mathsf{b}} = [a,b]$ in the tree $\mathcal{T}_{n,\mathsf{t}}$, and a subset $S^{\mathsf{b}} \subseteq v^{\mathsf{b}}$ satisfying: **(i)** X^{sb}_{sbw} is a C block source with each block having conditional entropy at least $20\mathsf{res}$ and **(ii)** for every $1 \le i < a$, X^{sb}_i is fixed.

C.2.2 Step 2: Getting a random value at the node v^{b}.

In the second state of the analysis, we'll restrict X^{sb} and Y to deficiency $\sqrt{\mathsf{res}}$ sources $X^{\mathsf{ag}}, Y^{\mathsf{ag}}$ that are "almost good" in the sense that in these sources the value $\mathsf{BExt}(X^{\mathsf{ag}}_{S^{\mathsf{b}}}, Y^{\mathsf{ag}})$ that is contained in $\mathsf{vals}(v^{\mathsf{b}})$ will be indeed distributed almost uniformly. (The reason this is not enough is that we need to ensure this value also propagates to the root of the tree.)

Lemma C.7. *Let X^{sb}, Y be as above. Let X^{ag} and Y^{ag} be the $\sqrt{\mathsf{res}}$ deficiency subsources of X^{sb} and Y obtained by restricting $X^{\mathsf{sb}}_{S^{\mathsf{b}}}$ and Y to the good subsources for the extractor* BExt *as in Corollary C.4. Then, for every subsources X'', Y'' of X and Y respectively of deficiency at most* $2\mathsf{res}$, *it holds that*

$$\left|\mathsf{BExt}(X^{\mathsf{ag}}_{S^{\mathsf{b}}}, Y^{\mathsf{ag}}) - U_{\mathsf{out}}\right| < 2^{-k^{\Omega(1)}}$$

Proof. Follows immediately from Corollary C.4. ◻

Result of step 2. At this point we have the sources X^{ag}, Y^{ag} with deficiency at most $k/9 + \sqrt{\mathsf{res}}$, a node v^{b} and set S^{b} satisfying that $\mathsf{BExt}(X^{\mathsf{ag}}_{S^{\mathsf{b}}}, Y^{\mathsf{ag}})$ is close to uniform, even when restricting to independent deficiency $2\mathsf{res}$ subsources.

C.2.3 Step 3: Fixing the challenges.

Our desired good subsources X^{good} and Y^{good} will be deficiency $\sqrt{\mathsf{res}}$ subsources of X^{ag} and Y^{ag}. The restriction will be the case that for every left uncle w of v^{b}, $\mathsf{Test}_{X^{\mathsf{good}}, Y^{\mathsf{good}}}(w) = \mathsf{fail}$ with probability one. We'll start with the following claim about the $\mathsf{Response}$ function:

Claim C.8. *Let X and Y be any two independent sources of lengths at most n with $H_\infty(X), H_\infty(Y) > \mathsf{res}$ and let $z \in \{0,1\}^{\mathsf{out}}$ be a string. Then, there exist independent subsources $X' \subseteq X$ and $Y' \subseteq Y$ with deficiency at most* $11\mathsf{out}$ *such that*

$$\Pr[a \in \mathsf{Response}(X', Y')] = 1$$

Proof. Recall that

$$\mathsf{Response}(x, y) = \{\mathsf{Vaz}(\mathsf{Ext}_{s_1}(x), \mathsf{Ext}_{s_2}(y)) \mid s_1, s_2 \in \{0,1\}^{5\log n}\}$$

Where Ext is a strong extractor with output length $10\mathsf{out}$. This means that if we choose s_1, s_2 at random then with high (and in particular with positive) probability we'll have that both distributions $X_1 \stackrel{def}{=} \mathsf{Ext}_{s_1}(X)$ and $Y_1 \stackrel{def}{=} \mathsf{Ext}_{s_2}(Y)$ satisfy $|X_1 - U_{10\mathsf{out}}|, |Y_1 - U_{10\mathsf{out}}| < 0.1$. By Lemma A.14 there are deficiency 2 sources $X'' \subseteq X$ and $Y'' \subseteq Y$ such that $X''_1 \stackrel{def}{=} \mathsf{Ext}_{s_1}(X'')$, and $Y''_1 \stackrel{def}{=} \mathsf{Ext}_{s_2}(Y'')$ both have min entropy at least $10\mathsf{out} - 3$. By the fact that $\mathsf{Vaz}(\cdot,\cdot)$ is a strong two source extractor, we get that with probability at least $1 - 2^{-\mathsf{out}} > 0.9$ over a random $x \in X''_1$,

$$\left|\mathsf{Vaz}(x, Y''_2) - U_{\mathsf{out}}\right|_1 < 2^{-2\mathsf{out}} \tag{4}$$

Choose at random $x_1 \leftarrow_{\mathrm{R}} X''_1$, then with probability at least 0.9, $\Pr[X''_1 = x_1] > 2^{-10\mathsf{out}-5}$ and hence by the union bound with positive probability x_1 satisfies both this condition and Equation 4. The first condition implies that if we set X' to be X conditioned on $\mathsf{Ext}_{s_1}(X) = x_1$ then X' is a subsource of X with deficiency at most $10\mathsf{out} + 5$. For this value x_1, we know that $\Pr[\mathsf{Vaz}(x_1, Y''_1) = z] \geq 2^{-\mathsf{out}} - 2^{-2\mathsf{out}} > 2^{-\mathsf{out}-1}$. Thus, if we look at the function $y_1 \mapsto \mathsf{Vaz}(x, y'_1)$ then z is a value of this function attained with probability at least $2^{-\mathsf{out}-1}$ over Y''_1, thus if we let Y' to be Y conditioned on $\mathsf{Vaz}(x_1, \mathsf{Ext}_{s_2}(Y)) = z$ then Y' is a subsource of Y with deficiency at most $\mathsf{out} + 1 + 2$. Clearly, $\Pr[\mathsf{Vaz}(\mathsf{Ext}_{s_1}(Y'), \mathsf{Ext}_{s_2}(X')) = z] = 1$. ◻

Left family of v^{b}. Let v^{b} be the vertex obtained at Step 1 (Lemma C.6). We let $\mathcal{P}_{v^{\mathsf{b}}}$ denote the path from v^{b} to the root in the tree $\mathcal{T}_{n,\mathsf{t}}$. That is, the set of nodes other than v^{b} on the path from v^{b} to the root. We let $\mathcal{L}_{v^{\mathsf{b}}}$ denote the *left family* if v^{b}, by which we mean the following: for every node $w \in \mathcal{P}_{v^{\mathsf{b}}}$ with $w \neq v^{\mathsf{b}}$, w has one child w_i in the path $\mathcal{P}_{v^{\mathsf{b}}}$, we add all the children w_j for $j < i$ to the set $\mathcal{L}_{v^{\mathsf{b}}}$. It's easy to see that property **(ii)** from the Step 1 implies that for every $w \in \mathcal{L}_{v^{\mathsf{b}}}$ is fixed to some constant with probability one. We'll now show that by restricting to a subsource, we can ensure that every such $w \in \mathcal{L}_{v^{\mathsf{b}}}$ fails the test with probability one.

Lemma C.9. *Let X^{ag} be the subsource obtained from Step 2 (Lemma C.7) with v^{b} the vertex obtained from Step 1 (Lemma C.6), and let Y^{ag} and independent source with at least* res *entropy. Then there exist two independent subsources X^{good} and Y^{good} of X^{ag} and Y^{ag} respectively with deficiency at most $\sqrt{\mathsf{res}}$ such that for every $w \in \mathcal{L}_{v^{\mathsf{b}}}$,*

$$\Pr[\mathsf{Test}_{X^{\mathsf{good}},Y^{\mathsf{good}}}(w) = \mathsf{fail}] = 1$$

Proof. We construct the subsources iteratively, making a restriction for every $w \in \mathcal{L}_{v^{\mathsf{b}}}$ and every element $\langle S, z\rangle \in \mathsf{vals}(w)$. Note that for every node w, the set $\mathsf{vals}(w)$ depends only on x_w and y. Note that $\mathsf{vals}(w)$ has at most nrows elements, where each element is a pair consisting of a set and a string in out, and with the set being a union of C intervals. Thus $\mathsf{vals}(w)$ can be described using at most $\mathsf{nrows}(\mathsf{C}2\log n + \mathsf{out}) < 2 \cdot \mathsf{nrows} \cdot \mathsf{out} < \mathsf{res}^{0.3}$.

For every $w \in \mathcal{L}_{v^{\mathsf{b}}}$, we know that X^{ag}_w is fixed, and hence $\mathsf{vals}(w)$ is a random variable depending only on Y^{ag}. Thus, we can restrict Y^{ag} to some $\hat{Y}^{\mathsf{ag}}$ with deficiency at most $\mathsf{res}^{0.3}$ such that $\mathsf{vals}(w)$ is fixed on X^{ag} and $\hat{Y}^{\mathsf{ag}}$. Let v be the parent of w. We know that v is an ancestor of v^{b} and hence $S^{\mathsf{b}} \subseteq v^{\mathsf{b}} \subseteq v$, it follows that $H_\infty(X^{\mathsf{ag}}_v) \geq (20\mathsf{C}-1)\mathsf{res}$ and in particular larger than $2\mathsf{res}$.

Now, using Claim C.8, as long as $X^{\mathsf{ag}}_{v^{\mathsf{b}}}$ and Y^{ag} have at least res min entropy, for every element $\langle S, z\rangle$ in $\mathsf{vals}w$, z is fixed and so we can restrict $X^{\mathsf{ag}}_{v^{\mathsf{b}}}$ and $\hat{Y}^{\mathsf{ag}}$ to subsources on which $z \in \mathsf{Response}(x_{v^{\mathsf{b}}}, y)$ with probability one, losing at most $11\mathsf{out}$ in the deficiency. We continue in this for all elements in $\mathcal{V}$, losing overall at most $\mathsf{nrows} \cdot 11\mathsf{out} < \mathsf{res}^{0.3}$. When we're done we obtain subsources $\hat{X^{\mathsf{ag}}}$ and $\hat{Y^{\mathsf{ag}}}$ of X^{ag} and Y^{ag} respectively with deficiency at most $2\mathsf{res}^{0.3}$ such that $\Pr[\mathsf{Test}_{\hat{X^{\mathsf{ag}}},\hat{Y^{\mathsf{ag}}}}(w) = \mathsf{fail}] = 1$. We repeat this process for every $w \in \mathcal{L}_{v^{\mathsf{b}}}$ to obtain the subsources X^{good}, Y^{good}. Since there are at most $\mathsf{t}\log n \ll \mathsf{res}^{0.1}$ of them, the overall deficiency is less than $\sqrt{\mathsf{res}}$. $\square$

Result of Step 3: At this point we know that **(i)** X^{good} and Y^{good} are deficiency $\sqrt{\mathsf{res}}$ subsources of the sources X^{ag} and Y^{ag} satisfying the properties assured by Lemma C.7 **(ii)** In particular (i) implies that X^{good} and Y^{good} are subsources with deficiency less than $k/9 + 2\sqrt{\mathsf{res}}$ of the original sources X and Y **(iii)** Since X^{good} is a subsource of X^{sb} with deficiency at most $2\sqrt{\mathsf{res}}$, we have that $H_\infty(X^{\mathsf{good}}_{S^{\mathsf{b}}}) \geq H_\infty(X^{\mathsf{sb}}_{S^{\mathsf{b}}}) - 2\sqrt{\mathsf{res}} \geq (\mathsf{C}-1)20\mathsf{res}$ and **(iv)** For every $w \in \mathcal{L}_{v^{\mathsf{b}}}$, $\mathsf{Test}(w) = \mathsf{fail}$ with probability one over X^{good} and Y^{good}.

C.2.4 Step 4: Finishing up.

Let X' and Y' be any independent subsources of X^{good} and Y^{good} with deficiency res. To finish the proof of Lemma C.3, we need to prove that $\mathsf{SSE}(X', Y')$ is close to a somewhere random distribution. Because X^{good} and Y^{good} are deficiency $\sqrt{\mathsf{res}} < \mathsf{res}$ of X^{ag} and Y^{ag}, we already know from Lemma C.7 that $z = \mathsf{BExt}(X'_{S^{\mathsf{b}}}, Y')$ will be almost uniformly distributed. Thus, all that we need to show is that with high probability the pair $\langle S^{\mathsf{b}}, z\rangle$ will be contained in $\mathsf{vals}([1, n])$ (i.e., in the set of values of the root of $\mathcal{T}_{n,\mathsf{t}}$. Because in X^{good} and Y^{good} (and hence in any of their subsources) all of the "left family" $w \in \mathcal{L}_{v^{\mathsf{b}}}$ fail the test, to prove that z does indeed propagate to the root it is sufficient to prove that for every set of responses R, z will be contained in R with $2^{-k^{\Omega(1)}}$ probability (we can then do a union bound over the nodes on the path from v^{b} to the root). This is what we prove below.

Lemma C.10. *Let X' and Y' be any independent* res *deficiency subsources of the sources X^{good} and Y^{good} above. Let v be any node in the tree $\mathcal{T}_{n,\mathsf{t}}$, then*

$$\Pr[\mathsf{BExt}(X'_{S^{\mathsf{b}}}, Y') \in \mathsf{Response}(X'_v, Y)] < 2^{-k^{\Omega(1)}}$$

where $S^{\mathsf{b}} \subseteq v^{\mathsf{b}}$ is as above.

Proof. We'll prove by showing that for every $s_1, s_2 \in \{0,1\}^{5\log n}$,

$$p \stackrel{def}{=} \Pr[\mathsf{BExt}(X'_{S^{\mathsf{b}}}, Y') = \mathsf{Response}_{s_1,s_2}(X'_v, Y)] < 2^{-k^{\Omega(1)}}$$

the result will then follow by a union bound.

Note that for the node v, $\mathsf{Response}_{s_1,s_2}(x_v, y) = \mathsf{Vaz}(\mathsf{Ext}_{s_1}(x_v), \mathsf{Ext}_{s_2}(y))$ where Ext has output length $10\mathsf{out}$. It follows that p is equal to

$$\sum_{z_1,z_2\in\{0,1\}^{10\mathsf{out}}} \Pr[\mathsf{BExt}(X'_{S^{\mathsf{b}}}, Y') = \mathsf{Vaz}(z_1, z_2) | \mathsf{Ext}_{s_1}(X'_v) = z_1, \mathsf{Ext}_{s_2}(Y') = z_2] \Pr[\mathsf{Ext}_{s_1}(X'_v) = z_1] \Pr[\mathsf{Ext}_{s_2}(Y') = z_2]$$

Let G_1 be the set of z_1's such that $\Pr[\mathsf{Ext}_{s_1}(X'_v) = z_1] \geq 2^{-30\mathsf{out}}$ and similarly G_2 is the set of z_2's such that $\Pr[\mathsf{Ext}_{s_2}(Y') = z_2] \geq 2^{-30\mathsf{out}}$, then p is at most

$$\sum_{z_1\in G_1, z_2\in G_2} \Pr[\mathsf{BExt}(X'_{S^{\mathsf{b}}}, Y') = \mathsf{Vaz}(z_1, z_2) | \mathsf{Ext}_{s_1}(X'_v) = z_1, \mathsf{Ext}_{s_2}(Y') = z_2] \Pr[\mathsf{Ext}_{s_1}(X'_v) = z_1] \Pr[\mathsf{Ext}_{s_2}(Y') = z_2]$$
$$+ 2^{20\mathsf{out}} 2^{-30\mathsf{out}}$$

However, for every $z_1 \in G_1$ and $z_2 \in G_2$ define the subsources $X'' \stackrel{def}{=} X' | \mathsf{Ext}_{s_1}(X'_v) = z_1$ and $Y'' \stackrel{def}{=} Y' | \mathsf{Ext}_{s_2}(Y') = z_2$. Note that on X'' and Y'' we have that $\mathsf{Vaz}(X'', Y'')$ is fixed to the value $r = \mathsf{Vaz}(z_1, z_2)$ with probability one. Because of the definitions of G_1 and G_2 we have that X'' and Y'' have deficiency less than $30\mathsf{out} < \sqrt{\mathsf{res}}$ as subsources of X' and Y' respectively and hence have deficiency less than $2\mathsf{res}$ as subsources of X^{ag} and Y^{ag} respectively. Yet this implies that $\mathsf{BExt}(X'', Y'')$ is 2^{-k^c} close to the uniform distribution for some constant $c > 0$ which implies that it will only hit the value r with at most this probability. It follows that

$$p \leq \sum_{z_1\in G_1, z_2\in G_2} 2^{-k^c} \Pr[\mathsf{Ext}_{s_1}(X'_v) = z_1] \Pr[\mathsf{Ext}_{s_2}(Y') = z_2] + 2^{20\mathsf{out}} 2^{-30\mathsf{out}}$$

However, since out is at least $k^{1/10}$ and since $\sum_{z_1,z_2\in\{0,1\}^{10\mathsf{out}}} \Pr[\mathsf{Ext}_{s_1}(X'_v) = z_1] \Pr[\mathsf{Ext}_{s_2}(Y') = z_2] = 1$ we're done. $\square$

D A (Single Output) 2-Source Disperser.

In this section we prove the main theorem of the paper. That is, we prove:

Theorem D.1 (2-Source Disperser). *For every $\delta > 0$ there exists a polynomial time computable function $Disp : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ that is a 2-source disperser with entropy threshold $k = n^{\delta}$.*

Before proving the theorem we will need some preliminaries. In particular, we will need to extend the challenge/response mechanism already used in the previous section (this time using the construction of the subsource somewhere extractor as a black box). This is what we do first.

D.1 High level description of the algorithm.

We now give some high level description (but with more details than the introduction) of our construction for a two source disperser. This algorithm uses some elaborations and extensions on the subsource somewhere extractor of Appendix C and hence it is highly recommended that the one reads that section first before the current one.

Consider the challenge response mechanism used in Appendix C. There given a source X and a block (i.e., projection to some set of coordinates S) X_S of X, we produced a *somewhere random list of values* $\mathsf{vals}(X_S)$ from X_S, and tested whether the list $\mathsf{vals}(X_S)$ is contained in a response set R (where both the list and the response set were produced with the help of the second source Y). If this happened then we concluded that probably X_S has very small (or zero) entropy, and otherwise we concluded that it probably has non-negligible amount of entropy.

A crucial point in this section is to produce a *finer resolution* challenge/response mechanism. This mechanism will not produce just a true or false value but values from a larger range. Initially, think of a mechanism that outputs S if the block X_S is small entropy, M if it entropy at least some threshold k_{min} but there's also some entropy in X other than in X_S (i.e., X has conditional entropy given X'), and L if X_S has essentially all (or a very large fraction) of the entropy of X. It seems that if we have such a test we can also add a value M^+ that should be output if X has entropy at least, say $2k_{min}$ but not everything. This will mean that if we have X_S is a source of $2k_{min}$ entropy then intuitively on one hand the testing procedure should output M^+ with very high probability, but on the other hand, we can restrict to a subsource X'_S of X_S that has only k_{min} entropy, and on this subsource the test will output 1. This means that if X_S is a block with $2k_{min}$ entropy but not everything, the test will output both M and M^+ with positive probability. (We'll actually directly prove that this value M^+ is outputted with positive probability on intermediate entropy sources.)

This naturally suggests the following algorithm: Given X break X into two pieces X_1 and X_2: X_1 will be relatively very small (e.g., of size n/t for say $t = n^{0.01}$) and X_2 will be the rest (e.g., of size $n - n/t$). Now, test X_1:

- If we get S then throw away X_1 and continue recursively on X_2. Note that we did not lose a lot of entropy by throwing away X_1.
- If we get L then then throw away X_2 and continue recursively on X_1. Note that in this case we have significantly condensed the entropy we're working with.
- If we get either M or M^+ then halt and output a value, say 0 if we get M and 1 if we get M^+.

Roughly speaking this is the algorithm we use. However, due to many subtleties, we cannot analyze it in such a simple way, and the analysis is best done non-recursively.

Getting a finer grained test. Our general idea will be to use a similar test as was done in the construction of the subsource somewhere extractor SSE in Appendix C. This time, we will use SSE as a black box to generate the challenges. When testing a block X_S of X, we will use SSE on X_S and the second source Y to generate a "challenge" list $C(X_S)$ of nrows values each of length out. Note that out can be as high as a fixed power of k_{min} (e.g. $k_{min}^{1/5}$) while $\mathsf{nrows} = k^{o(1)}$ (in fact can be polylogarithmic in k). We look again at the response set R (which is generated in the same way as in Appendix C using the function $\mathsf{Response}$ on X and the second input Y). Now let $\ell \leq \mathsf{out}$ be some number. We say that the challenge *ℓ-fails* the test if all for every string $z \in C(X_S)$, the truncation of z to the first ℓ bits is a substring of some string in the set R. Note that if $C(X_S)$ is indeed somewhere random then it will pass this test with probability about $1 - n^{O(1)}2^{-\ell}$. Also note that if the challenge ℓ-fails the test then it also ℓ'-fails it for any $\ell' \leq \ell$.

Our idea will be to have three numbers $\ell_1 \gg \ell_2 \gg \ell_3$ and:

f-f-f Output S if $C(X_S)$ ℓ_1-failed (and hence also ℓ_2-failed and ℓ_3-failed)

p-p-p Output L if $C(X_S)$ passes all tests (i.e., does not ℓ_3-fail)

p-f-f/p-p-f If $C(X_S)$ passes ℓ_1 but ℓ_3-fails, then output M if it ℓ_2-fails and M^+ otherwise.

Why does this work? This is not exactly compatible with our intuition above, so let's try to understand how could this work. Similar to the previous section, the analysis will involve restricting to subsources. The idea is the following:

First, if a block has zero entropy, then we can ensure it ℓ_i-fails by losing at most $\mathsf{nrows} \cdot \ell_i$ entropy. Thus, on small entropy sources, we can restrict to subsources with small deficiency to ensure that the test ℓ_1-fails (and hence ℓ_i-fails for all i) with probability one. Thus indeed, for a small block we our test will return (on a subsource) the output S with probability one.

Secondly, if a block has high entropy, then by similar analysis as before we can show that it will pass ℓ_3 (and hence also ℓ_1 and ℓ_2) with very high probability (e.g., something like $1 - n^{O(1)}2^{-\ell_3}$. (There is an important subtlety, which we ignore for a moment, that this will not happen on the original sources but rather on some subsources of them.)

The case of a medium entropy block is more interesting. Here we show the following: as long as a block does not have *all* of the source's entropy, then the probability that it ℓ_i-fails is bounded from below by roughly $2^{-\mathsf{nrows} \cdot \ell_i}$. This can be shown by considering any possibly fixed value for X_S. For every such fixing, X_S is a zero entropy source, and hence, as we observed before there exists a subsource $Y \subseteq Y$ taking up $2^{-\mathsf{nrows} \cdot \ell_i}$ fraction of Y on which the test fails. Even though $2^{-\mathsf{nrows} \cdot \ell_i}$ is a very small number this lower bound will still be useful for us, since we know that with probability $\geq 1 - n^{O(1)}2^{-\ell_1}$ the block ℓ_1-passes, and so if $\ell_1 \gg \mathsf{nrows} \cdot \ell_2$ with some positive probability it will happen that $C(X_S)$ both ℓ_1-passes and ℓ_2-fails. Similarly, with some positive probability it will happen that $C(X_S)$ both ℓ_2-passes and ℓ_3-fails.

All of this ignored the complication that to make a block with some entropy pass the test, we need to restrict to some subsource (since SSE is not a somewhere extractor but only a *subsource* somewhere extractor), and once we continue and make other restrictions we might ruin this property. We can handle this for one block (since we only make this restriction once) but this will be problematic as we continue to subblocks of this block by recursion. Here of course we'll heavily rely on the *resilience* of SSE. Unfortunately, for SSE it holds that the resilience (the amount of deficiency it can tolerate, denoted res there) is actually smaller than the deficiency it requires (denoted there def), by a factor of about nrows (which as we said can be as small as polylog(n)). This means that we will need to use versions of SSE with successively smaller parameters as we proceed in the algorithm, such that the resiliency of an invocation of SSE is larger than the sum of deficiencies of all future invocations that may affect it. Fortunately, if we set the parameter t right (recall that t is the ratio between the source length and block length, and can be a small power of n) then the recursion depth is constant and we only need to make a constant number of such adjustments.

Guide to rest of the section. The construction of the entropy testing mechanism is described in Appendix D.2. We construct there first a "binary" tester (returning fail or pass) and then use the results of three such testers (in the manner described above) to decide on the course of action. The reader may want to look ahead to the construction of the disperser itself, described in Appendix D.4.

D.2 The construction of TestBlock

In this section we construct our implementation for an "entropy testing" mechanism. This is a powerful building block that when given a source X (and an additional independent source Y) allows us to find pieces in X which contain "a lot of entropy". More formally, when given an (n, k)-source X, a parameter t (measuring the number of blocks) and a threshold parameter k_{min} we design a procedure $\mathsf{TestBlock}_t(x, y, i)$ that outputs either pass or fail.

While the next paragraph is not formally true and is an oversimplification of what we are able to prove about TestBlock in Section D.3 (see Theorems D.3,D.5,D.6) it is helpful to imagine that TestBlock has the following properties:

- If the i'th block has no entropy (or very low entropy) then TestBlock fails.
- If the i'th block has all the entropy then TestBlock passes (with high probability).
- If the i'th block has at least "medium entropy" but not all of the entropy then TestBlock passes with high probability, but nevertheless we also know that it fails with noticeable probability.

How to construct a Disperser when given TestBlock: Indeed, the third item above is close in spirit to a disperser: It says that TestBlock outputs two different values. This means that once we identify blocks that have some entropy but not all of the initial entropy then we can construct a disperser.

We use TestBlock itself to find such blocks. we use TestBlock to identify and "get rid" of blocks that contain zero (or very low) entropy and concentrate on pieces of the source that have high entropy. When getting rid of low entropy blocks. We increase the rate of the source. Applying this procedure sufficiently many times we can guarantee that the entropy in our source is higher than the length of a block and then the third item above trivially holds.

One point that is not covered in this outline is that when TestBlock passes we are not sure whether the block it passed on is a medium entropy block (and then we should output pass) or because the block contains all the entropy of the source (and then we should concentrate on it). Indeed, we will apply several instantiations of TestBlock and do a case analysis of their outputs to distinguish between these two cases.

An important ingredient in our construction will be the subsource somewhere extractor of Theorem C.2 (Appendix C).

Construction D.2 ($\mathsf{TestBlock}_t(x, y, i)$: Check whether a block has high entropy)**.**

Inputs:

- strings x, y of length $n_1 \leq n_2$.
- an integer $i \in [t]$.

Basic parameters:

- $n_1 \leq n_2$ - the length of inputs x and y.
- t - The number of blocks in x.
- $b = n_1/t$ the length of blocks in x.

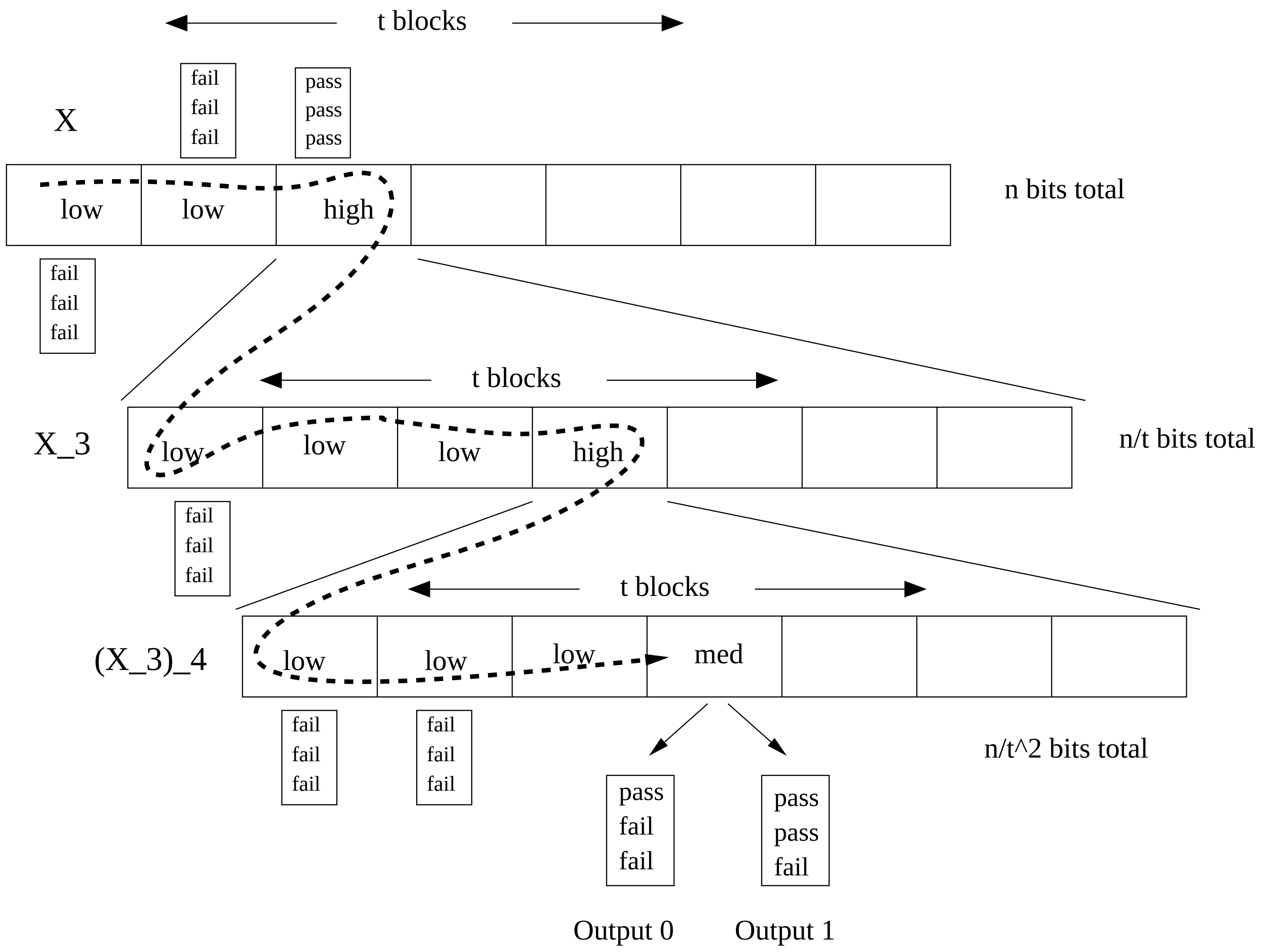

Figure 3: Finding the medium block

- i - the index of the block in x that is being checked for high entropy.
- k_{min} - the minimal amount of entropy in the block x_i that should make the test pass.

Additional internal parameters:

- ℓ (to be defined later on as $\ell = \mathsf{nrows} \cdot \mathsf{len}$ - the length of "challenge strings". Governs the deficiency of subsources on which we can show that the procedure works.
- $w \geq \ell$ - the length of "response generating blocks".

Ingredients:

- A resilient 2-source subsource somewhere extractor $\mathsf{SSE} : \{0,1\}^b \times \{0,1\}^{n_2} \to (\{0,1\}^{\mathsf{len}})^{\mathsf{nrows}}$ with nrows output rows of length len, entropy threshold (k_{min}, k_2), deficiency def, resiliency res and error $\bar{\epsilon}$.
- Two seeded strong extractors $\mathsf{Ext}_1, \mathsf{Ext}_2$ both with seed length $10 \log n_2 > 10 \log n_1$, output length w and error $1/100$. (See Appendix A.3.
  - $\mathsf{Ext}_1 : \{0,1\}^{n_1} \times \{0,1\}^{10 \log n_2} \to \{0,1\}^w$ is a strong $(w, 1/100)$-extractor.

- $\mathsf{Ext}_2 : \{0,1\}^{n_2} \times \{0,1\}^{10\log n_2} \to \{0,1\}^w$ is a strong $(w, 1/100)$-extractor.

- A 2-source extractor $\mathsf{Vaz} : (\{0,1\}^w)^2 \to \{0,1\}^\ell$ with entropy threshold 9ℓ and error $2^{-2\ell}$.

Informal statement of Goal: (See Theorems D.3,D.5) If X, Y are sources with sufficient entropy then if X_i is fixed then TestBlock fails (on a large subsource) and if X_i has entropy larger than k_{min} then TestBlock passes (with high probability over some large subsource). Moreover if $(X|X_i)$ has even few bits of entropy then TestBlock fails with noticeable probability.

Operation:

Computing the challenge: Let $Challenge(x, y) = \mathsf{SSE}(x_i, y)$ (this is a string of length ℓ).

Computing the responses: For every $s_1, s_2 \in \{0,1\}^{10\log n_2}$, compute $\mathsf{Ext}_1(x, s_1)$ and $\mathsf{Ext}_2(x, s_2)$. Let $\mathsf{Response}_{s_1,s_2}(x, y) = \mathsf{Vaz}(\mathsf{Ext}_1(x, s_1), \mathsf{Ext}_2(x, s_2))$ (note that this is a string of length ℓ).

Deciding whether to pass or fail: We say that the test passes if for every $s_1, s_2 \in \{0,1\}^{10\log n_2}$, $\mathsf{Response}_{s_1,s_2}(x, y) \neq Challenge(x, y)$.

D.3 Properties of TestBlock

We now prove that TestBlock has the desired properties.

Theorem D.3 (TestBlock fails on fixed blocks)**.** *Let X be an (n_1, w) source and Y be an $(n_2, \ell + w)$ source. Let $i \in [t]$ be an index such that X_i is fixed. Then there exist deficiency $3w$ subsources X' of X and Y' of Y such that* $\Pr[\mathsf{TestBlock}_t(X', Y', i)$ *fails*$] = 1$.

Proof. We have that X_i equals some fixed string x_i. It follows that $Challenge(X, Y) = \mathsf{SSE}(X_i, Y) = \mathsf{SSE}(x_i, Y)$ does not depend on X and is a function of Y. By Lemma A.12 there exists a subsource $\tilde{Y}$ of Y of deficiency ℓ such that $Challenge(X, \tilde{Y})$ is fixed to some value c. We have that X is a w-source and by Fact A.8 $\tilde{Y}$ is a $(w + \ell - \ell)$-source. As these thresholds are high enough for the seeded extractors $\mathsf{Ext}_1, \mathsf{Ext}_2$ it follows that there exist s_1 and s_2 such that $\mathsf{Ext}_1(X, s_1), \mathsf{Ext}_2(\tilde{Y}, s_2)$ are both 1/100-close to uniform. By Lemma A.29 (2nd item) applied to $A = \mathsf{Ext}_1(X, s_1)$ and $B = \mathsf{Ext}_2(\tilde{Y}, s_2)$ there exist strings $\hat{x}, \hat{y} \in \{0,1\}^w$ such that $\mathsf{Vaz}(\hat{x}, \hat{y}) = c$, $\Pr[E(X, s_1) = \hat{x}] > 2^{-2w}$ and $\Pr[E(\tilde{Y}, s_2) = \hat{y}] > 2^{-2w}$. Let $X' = (X|E(X, s_1) = \hat{x}$ and $Y' = (\tilde{Y}|E(\tilde{Y}, s_2) = \hat{y})$. We have that X' is a deficiency $2w$ subsource of X and Y' is a deficiency $2w$ subsource of $\tilde{Y}$ and therefore a deficiency $3w$ subsource of Y. Note that $\mathsf{Response}_{s_1,s_2}(X', Y') = \mathsf{Vaz}(\hat{x}, \hat{y}) = c = Challenge(X', Y')$. Thus, $\mathsf{TestBlock}_t(X', Y', i)$ fails on X', Y' with probability one. □

The next Theorem shows that TestBlock passes on high entropy blocks. We first need the following definition.

Definition D.4 (TestBlock passes resiliently)**.** Let X' and Y' be independent distributions. We say that TestBlock passes on X' and Y' with probability p and resiliency r if for every deficiency r subsources X'' of X' and Y'' of Y', $\Pr[\mathsf{TestBlock}_t(X'', Y'', i)$ passes$] \geq p$.

Theorem D.5 (TestBlock passes on blocks with sufficient entropy)**.** *Let $i \in [t]$ be an index such that X_i is a (b, k_{min})-source, Let Y be a (n_2, k_{min})-source. Then:*

1. *(Simple version) If* $\mathsf{res} \geq 20\ell$ *then there exist deficiency* def *subsources X' of X and Y' of Y such that* $\Pr[\mathsf{TestBlock}_t(X', Y', i)$ *passes*$] \geq 1 - poly(n_2)(2^{-\mathsf{len}} + \bar{\epsilon})$.

2. *(Resilient version) Let* $r = \mathsf{res} - 20\ell$ *then there exist deficiency $\bar{d_1}$ subsources X' of X and Y' of Y such that* TestBlock *passes on X', Y' with probability $1 - poly(n_2)(2^{-\mathsf{len}} + \bar{\epsilon})$ and resiliency r.*

Note that the first item follows from the second item.

Proof. It suffices to prove the second item. Let X', Y' be the subsources of deficiency def guaranteed by the properties of the subsource somewhere extractor SSE. It suffices to show that for every every subsources X'' of X' and Y'' of Y' of deficiency r, and for every $s_1, s_2 \in \{0,1\}^{10\log n_2}$, $\Pr[\mathsf{Response}_{s_1,s_2}(X'', Y'') = \mathit{Challenge}(X'', Y'')] \leq 2 \cdot 2^{-m} + \bar{\epsilon}$ This is because in any X'', Y'' we can do a union bound over the $\mathit{poly}(n_2)$ choices for s_1, s_2 and ensure that no response string hits the challenge.

We now concentrate on some choice of X'', Y'' and s_1, s_2 as above. By Lemma A.11 (with X'' playing the part of X, $E(X'', s_1)$ playing the part of y and $\rho = 2^{-\ell}$ there is a set A_1 such that $\Pr[\mathsf{Ext}_1(X'', s_1) \notin A_1] \leq 2^{-\ell}$ and for every $a_1 \in A_1$, $X''' = (X''|\mathsf{Ext}_1(X'', s_1) = a_1)$ is a deficiency 20ℓ subsource of X''. We repeat this argument for Y'' and conclude that there is a set A_2 such that $\Pr[\mathsf{Ext}_2(Y'', s_2) \notin A_2] \leq 2^{-\ell}$ and for every $a_2 \in A_2$, $Y''' = (Y''|\mathsf{Ext}_2(Y'', s_2) = a_2)$ is a deficiency 20ℓ subsource of Y''.

$$\Pr[\mathsf{Response}_{s_1,s_2}(X'', Y'') = \mathit{Challenge}(X'', Y'')] \tag{5}$$
$$\leq \Pr[\mathsf{Response}_{s_1,s_2}(X'', Y'') = \mathit{Challenge}(X'', Y'')|\mathsf{Ext}_1(X'', s_1) \in A_1 \cap \mathsf{Ext}_2(Y'', s_2) \in A_2] + 2 \cdot 2^{-\ell}$$

Thus, it is sufficient to bound

$$\begin{aligned}\Pr[\mathsf{Response}_{s_1,s_2}(X'', Y'') &= \mathit{Challenge}(X'', Y'')|\mathsf{Ext}_1(X'', s_1) \in A_1 \cap \mathsf{Ext}_2(Y'', s_2) \in A_2] \\ &= \Pr[\mathsf{Response}_{s_1,s_2}(X''', Y''') = \mathit{Challenge}(X''', Y''')\end{aligned}$$

Note that $\mathsf{Ext}_1(X''', s_1)$ and $\mathsf{Ext}_2(Y''', s_2)$ are fixed and therefore $\mathsf{Response}_{s_1,s_2}(X''', Y''')$ is fixed. We have that X''' is a deficiency 20ℓ of X'' which is in turn a deficiency r of X'. It follows that X''' is a deficiency $r + 20\ell \leq \mathsf{def}_1$ of X'. The same reasoning gives that Y''' is a deficiency $r + 20\ell \leq \mathsf{def}_2$ of Y'. We have that SSE is a subsource somewhere extractor with entropy threshold k_{min} and resiliency def. It follows that it works well on X_i''' and Y''' and therefore $\mathit{Challenge}(X''', Y''') = \mathsf{SSE}(X_i''', Y''')$ is $\bar{\epsilon}$-close to a somewhere random distribution. Note that the somewhere random distribution has min-entropy len and therefore the probability that the Challenge string will hit the response string in this setting is at most $2^{-\mathsf{len}} + \bar{\epsilon}$. We conclude that overall equation 5 can be bounded by $2^{-\mathsf{len}} + \bar{\epsilon} + 2 \cdot 2^{-\ell} \leq 2 \cdot 2^{-\mathsf{len}} + \bar{\epsilon}$ as required. ☐

We also need the next Theorem which shows that $\mathsf{TestBlock}$ fails with noticeable probability even on medium entropy blocks. A weakness of this next Theorem is that we are not able to guarantee that the failure happens on a subsource.

Theorem D.6 ($\mathsf{TestBlock}$ fails with noticeable probability on blocks that don't steal all the entropy)**.** *Let Y be an $(n_2, \ell+w)$ source. Let $i \in [t]$ be an index such that there exists a set $G \in \{0,1\}^b$ with $\Pr[X_i \in G] \geq 1/2$ and for every $x_i \in G$, The distribution $(X|X_i = x_i)$ is an (n_1, w)-source. Then $\Pr[\mathsf{TestBlock}_t(X, Y, i)$ fails$] \geq 2^{-4\ell}$.*

Proof. Choose some $x_1 \in \{0,1\}^b$ and consider the source $\tilde{X} = (X|X_i = x_i)$. We have that $\tilde{X}_i$ equals some fixed string x_i. From here we can continue in a similar fashion to the proof of Theorem D.3 as the i'th block is fixed: It follows that $\mathit{Challenge}(\tilde{X}_i, Y) = \mathsf{SSE}(\tilde{X}_i, Y) = \mathsf{SSE}(x_i, Y)$ does not depend on $\tilde{X}$ and is a function of Y. By Lemma A.12 there exists a subsource $\tilde{Y}$ of Y of deficiency ℓ such that $\mathit{Challenge}(\tilde{X}, \tilde{Y})$ is fixed to some value c. We have that $\tilde{X}$ is an (n_1, w)-source and by Fact A.8 $\tilde{Y}$ is a $(w + \ell - \ell)$-source. As these thresholds are high enough for the seeded extractors $\mathsf{Ext}_1, \mathsf{Ext}_2$ it follows that there exist s_1 and s_2 such that $E(X, s_1), E(\tilde{Y}, s_2)$ are both $1/100$-close to uniform. We now use Lemma A.29 (but unlike the proof of Theorem D.3 we use the 1st item of the Lemma as we don't care about subsources and want the probability to depend on ℓ rather than on w.) By Lemma A.29 (1st item) applied to $A = E(\tilde{X}, s_1)$ and $B = E(\tilde{Y}, s_2)$ we get that $\Pr[\mathsf{Vaz}(\mathsf{Ext}_1(\tilde{X}, s_1), \mathsf{Ext}_2(\tilde{Y}, s_2)) = c] \geq 2^{-2\ell}$ and $\Pr[\mathsf{TestBlock}_t(\tilde{X}, \tilde{Y}, i]$ fails$] \geq 2^{-2\ell}$. Let $A \subseteq \{0,1\}^n$ be a set such that $(Y|A) = \tilde{Y}$. As Y is a we know that there exists such an A with $\Pr[Y \in A] \geq 2^{-\ell}$. It follows that

$$\Pr[\mathsf{TestBlock}_t(\tilde{X}, Y, i) \text{ fails}] \geq \Pr[\mathsf{TestBlock}_t(\tilde{X}, \tilde{Y}, i) \text{ fails}|Y \in A] \Pr[Y \in A] \geq 2^{-2\ell} \cdot 2^{-\ell} = 2^{-3\ell}$$

To conclude the argument we note that:

$$\Pr[\mathsf{TestBlock}_t(X,Y,i) \text{ fails}] \geq \sum_{x_1 \in G} \Pr[\mathsf{TestBlock}_t(X,Y,i) \text{ fails} | X_i = x_i] \Pr[X_i = x_i] \geq 2^{-4\ell}$$

where the inequality follows because we already have the inequality for $\tilde{X} = (X|X_i = x_i)$.

□

D.4 The construction of a disperser with one output

We are now ready to prove Theorem D.1. We start with the construction:

Construction D.7 ($\mathsf{Disp}(x,y)$ 1-output subsource extractor)**.**

Inputs:

- strings x, y of length $n_1 \leq n_2$.
- $k = n^\delta$ - the entropy threshold guarantee on the two sources.

Basic parameters:

- $n_1 \leq n_2$ the length of inputs x and y
- $b = n_1/t$ the length of blocks in x.
- $\nu > 0$ a parameter needed to define t in the next item. We will later pick ν to be a very small constant.
- $t = n_1^\nu$ a parameter that measures the number of blocks.

Global internal parameters: By that we mean parameters that are not modified during the run.

- $k_{min} = \sqrt{k}/10$. This is the threshold of what we consider at least medium entropy. We will always apply $\mathsf{TestBlock}$ with this threshold.
- $h = 1/\nu$ - a bound on the number of levels in the recursion.
- $w = k^{1/4}$ - This is the parameter w from $\mathsf{TestBlock}$. We always call $\mathsf{TestBlock}$ with this parameter.
- $\ell = k^{1/5}$ - This is also a parameter from $\mathsf{TestBlock}$. We will call $\mathsf{TestBlock}$ with various values of ℓ throughout the construction. However, we promise that all of them will be smaller than this parameter.

The parameters of SSE: We use a subsource somewhere extractor on a varying sequence of entropy thresholds k_{p} (to be defined later on). All these thresholds will be larger than n^α for some fixed $\alpha > 0$. Thus, we can have Theorem C.2 supply us with such a subsource somewhere extractor supply us with a function SSE which meets these requirements. We now explain what we require of the other parameters of SSE (which are expressed as a function of the entropy threshold parameter). It is important to note that Theorem C.2 can meet this requirements (and is in fact a lot stronger). We use the following parameters:

- $\beta > 0$ a constant that governs the error. More precisely, on entropy threshold k we assume that SSE has error smaller than 2^{-k^β}.
- $\mu > 0$ a parameter that governs the number of output rows. More precisely, on entropy threshold k we assume that SSE has at most k^μ rows. Note that in Theorem C.2 we can choose $\mu = o(1)$ and therefore we assume that $\mu = 1/1000h$. This will later be used to generate a series of h numbers that are all between $k^{A_1} \leq k^{A_2}$ (for two constants A_1,A_2) such that every number is greater than the next one by a factor of k^μ.

- $\gamma = 1/1000h$: This is a parameter so that governs the resiliency. More precisely, on entropy threshold k we assume that SSE has resiliency at least $k^{1-\gamma}$. Note that in Theorem C.2 we can choose $\gamma = o(1)$ and therefore we assume that $\gamma \leq 1/1000h$.
- Whenever we apply SSE we will make sure that its error parameter $\log(1/\epsilon)$ is at least a constant factor times len. Thus, the error is always exponentially small in the output length.

A tree of intervals: We repeat here the definition of a partition tree (see Section C.1.3).

1. Assume without loss of generality that n is an integer power of t (can be done by padding).
2. We define a degree-t with depth $\log n/\log \mathsf{t} < \log n$ tree $\mathcal{T}_{n,\mathsf{t}}$ that we call the n, t *partition tree*. The nodes of $\mathcal{T}_{n,\mathsf{t}}$ are subintervals of $[1 \dots n]$ defined in the following way:
   (a) The root of the tree is the interval $[1, n_1]$.
   (b) If a node v is identified with the interval $[a, b]$ then we let $v_1, \dots, v_{\mathsf{t}}$ denote the t consecutive disjoint length-$|v|/\mathsf{t}$ subintervals of v. That is, $v_i = [a + (t-1)i, a + ti - 1]$. We let the i^{th} child of v be v_i.
   (c) The leaves of the tree are the length 1 intervals.
3. For a string $x \in \{0,1\}^n$ and a set $S \subseteq [1, n]$ we'll denote by x_S the projection of x into the coordinates of n. In particular if v is a node in $\mathcal{T}_{n,\mathsf{t}}$ then x_v denotes the projection of x into the interval v.
4. We define the *left family* of a node v in the tree, as in Appendix C. We let $\mathcal{P}_v$ denote the path from v to the root in the tree $\mathcal{T}_{n,\mathsf{t}}$. That is, the set of nodes other than v^{b} on the path from v to the root. We let $\mathcal{L}_v$ denote the *left family* if v, by which we mean the following: for every node $w \in \mathcal{P}_v$ with $w \neq v$, w has one child w_i in the path $\mathcal{P}_v$, we add all the children w_j for $j < i$ to the set $\mathcal{L}_v$.

Operation of $\mathsf{Disp}(x, y)$:

High level outline:

- We start at the root of $\mathcal{T}_{n,\mathsf{t}}$ and define an iterative process where in each step we select a child of the current node v by evaluating a function of x_v and y which we describe later on.
- If our internal node satisfies a certain property (that is tested as a function of x_v and y) we obtain our output bit (that is also computed as a function of x_v and y).
- We later show that leaves of the tree satisfy the property above and therefore we must stop and output some bit eventually.

Describing the process on one level:

Parameters: For each level p of the tree (we make the convention that the root is at level 0) we will have the following parameters:

1. n_{p} - The length of intervals at a level. By the definition of the tree $n_{\mathsf{p}} = n_1/t^{\mathsf{p}}$.
2. b_{p} - the length of intervals at the next level by the definition of the tree $n_{\mathsf{p}} = n_{\mathsf{p}}/t = n_1/t^{\mathsf{p}+1}$.
3. d_{p} - A deficiency parameter. (Loosely speaking thus measures how much deficiency we are allowed to use at the subtree with root v). We set $d_0 = \sqrt{k}/100$ and $d_{\mathsf{p}+1}$ will be chosen later on as a function of the parameters in level p.

Ingredients per level:

- A resilient 2-source subsource somewhere extractor $\mathsf{SSE} : \{0,1\}^{b_\mathsf{p}} \times \{0,1\}^{n_2} \to (\{0,1\}^{\mathsf{len}_\mathsf{p}})^{\mathsf{nrows}_\mathsf{p}}$ with $\mathsf{nrows}_\mathsf{p} = k^\mu$ output rows of length len_p, entropy threshold $d_\mathsf{p}/10$. deficiency $\mathsf{def} = d_\mathsf{p}/100$, resiliency $\mathsf{res}_\mathsf{p} = (\mathsf{def}_\mathsf{p})^{1-\gamma}$ and error $\bar{\epsilon}_\mathsf{p} = 2^{-\mathsf{def}_\mathsf{p}^\beta}$ (note that we'll always have $\mathsf{def} > k^{1/10}$ and so regardless of p the error will be at most $2^{-k^{\beta/10}}$ which is negligible). Note that these parameters exactly match our assumption on SSE.
- Two seeded strong extractors $\mathsf{Ext}_1, \mathsf{Ext}_2$ both with seed length $10 \log n_2 > 10 \log n_\mathsf{p}$ (as $n_2 \geq n_1 \geq n_\mathsf{p}$, output length w and error $1/100$. More precisely,
  - $\mathsf{Ext}_1 : \{0,1\}^{n_\mathsf{p}} \times \{0,1\}^{10 \log n_2} \to \{0,1\}^w$ is a strong $(w, 1/100)$-extractor.
  - $\mathsf{Ext}_2 : \{0,1\}^{n_2} \times \{0,1\}^{10 \log n_2} \to \{0,1\}^w$ is a strong $(w, 1/100)$-extractor.
- A 2-source extractor $\mathsf{Vaz} : (\{0,1\}^w)^2 \to \{0,1\}^\ell$ with entropy threshold 9ℓ and error $2^{-2\ell}$.

Component: Three instantiations of $\mathsf{TestBlock}$: At every level p we define three instantiations of $\mathsf{TestBlock}$ denoted $\mathsf{TestBlock}_\mathsf{p}^{(e)}$ for $e \in [3]$. (Note that we omit the index p whenever it is clear from the context.).

Parameters: For $e \in [3]$ we'll have parameters $\mathsf{len}_\mathsf{p}^{(e)}$ and $\ell_\mathsf{p}^{(e)}$ such that:

- for every p and e: $k^{\beta/30} \leq \mathsf{len}_\mathsf{p}^{(e)} \leq k^{\beta/20}$ (That is these numbers are always between two polynomials in k).
- for every p and e: $\ell_\mathsf{p}^{(e)} = \mathsf{nrows}_\mathsf{p} \cdot \mathsf{len}_\mathsf{p}^{(e)} = k^\mu \cdot \mathsf{len}_\mathsf{p}^{(e)}$.
- for every p: $\ell_\mathsf{p}^{(1)} \cdot k^{0 \cdot \mu} = \mathsf{len}_\mathsf{p}^{(1)} \cdot k^{1 \cdot \mu} = \ell_\mathsf{p}^{(2)} \cdot k^{2 \cdot \mu} = \mathsf{len}_\mathsf{p}^{(2)} \cdot k^{3 \cdot \mu} = \ell_\mathsf{p}^{(3)} \cdot k^{4 \cdot \mu} = \mathsf{len}_\mathsf{p}^{(3)} \cdot k^{5 \cdot \mu} = \ell_{\mathsf{p}+1}^{(1)} \cdot k^{6 \cdot \mu}$.
- That is in the same level every len is greater than the next ℓ by a k^μ factor and the smallest number of level p is larger than the largest one of level $\mathsf{p}+1$ by a factor of k^μ.

Operation of $\mathsf{TestBlock}^{(e)}$ at level p:

- For $e \in [3]$, recall that Vaz is a 2-source extractor with output length ℓ, let $\mathsf{Vaz}^{(e)}(x,y) = \mathsf{Vaz}(x,y)_{1,\ldots,\ell_\mathsf{p}^{(e)}}$ (that is truncating the output of Vaz to length $\ell^{(e)}$. Note that $\mathsf{Vaz}^{(e)}$ is an extractor with the same parameters as Vaz except for the output length).
- For $e \in [3]$, recall that SSE is a subsource somewhere extractor with output length len_p, let $\mathsf{SSE}^{(e)}(x,y) = \mathsf{SSE}(x,y)_{1,\ldots,\mathsf{len}^{(e)}}$ (that is truncating the output of SSE to length $\mathsf{len}_\mathsf{p}^{(e)}$. Note that $\mathsf{SSE}^{(e)}$ is a subsource somewhere extractor with the same parameters as SSE except for the output length).
- Define $\mathsf{TestBlock}^{(e)}(x,y,i)$ to be the function $\mathsf{TestBlock}_t(x,y,i)$ defined using the ingredients $\mathsf{Vaz}^{(e)}$ and $\mathsf{SSE}^{(e)}$ and the parameters of the current level p.
- It follows from the way $\mathsf{TestBlock}$ is constructed that for every $e \in \{1,2\}$, and every x, y and i if $\mathsf{TestBlock}^{(e)}(x,y,i)$ fails then $\mathsf{TestBlock}^{(e+1)}(x,y,i)$ fails. This is proven in Lemma D.8.

The operation performed in level p:

Beginning:

- We have a node v at level p:
- We also have two strings $x \in \{0,1\}^{n_\mathsf{p}}$ and $y \in \{0,1\}^{n_2}$.(We will maintain the guarantee that our current x is in fact the original string x restricted to the interval of v).
- Note that this indeed satisfied when we start the process at level 0.

Finding a "high entropy" block:

1. For $i \in [t]$ run $\mathsf{TestBlock}^{(1)}(x,y,i)$. If none of the blocks passed then output an arbitrary value.

2. Let i be the first block such that $\mathsf{TestBlock}^{(1)}(x,y,i)$ passes. For $e \in \{2,3\}$ let $val^{(e)}$ denote the output of $\mathsf{TestBlock}^{(e)}(x,y,i)$. Recall that it is impossible that $val^{(2)} = \mathsf{fail}$ and $val^{(3)} = \mathsf{pass}$ and therefore there are three possibilities:
3. $val^{(2)} = \mathsf{fail}$ and $val^{(3)} = \mathsf{fail}$: Output '*zero*'.
4. $val^{(2)} = \mathsf{pass}$ and $val^{(3)} = \mathsf{fail}$: Output '*one*'.
5. $val^{(2)} = \mathsf{pass}$ and $val^{(3)} = \mathsf{pass}$: Then
6. Let v_i denote the i'th child of v. If v_i is a leaf output an arbitrary value.
7. Continue the process using the node v_i and strings x_{v_i} and setting $d_{\mathsf{p}+1} = \mathsf{res}_{\mathsf{p}}/100 = (d_{\mathsf{p}}/100)^{1-\gamma}/100$. (this means that $d_{\mathsf{p}+1}$ is essentially $d_{\mathsf{p}}^{1-\gamma}$.)

D.5 Analysis

Let $k = n^{\delta}$ and let X be an (n_1,k)-source and Y be an (n_2,k)-source such that X and Y are independent. We set:

- $C = 2$.
- $\alpha = \log(200)/\nu \log n$.

By choosing $\nu > 0$ to be a sufficiently small constant we can make all the requirements below hold:

$$n^{2\nu} \leq \frac{k}{40/\log n} \tag{6}$$

$$k/n^{2\nu} \geq 2\sqrt{k} \tag{7}$$

$$3k^4 \cdot n^{\nu}/\nu \leq \sqrt{k}/100 \tag{8}$$

$$\frac{\sqrt{k}\nu}{300n^{\nu}} \geq k^{1/4} \tag{9}$$

D.5.1 Preparations:

We need the following property of the three instantiations of $\mathsf{TestBlock}$.

Lemma D.8. *For every* p *and* $e \in \{1,2\}$ *and for every* $x \in \{0,1\}^{n_{\mathsf{p}}}$, $y \in \{0,1\}^{n_2}$ *and* $i \in t$, *if* $\mathsf{TestBlock}^{(e)}(x,y,i)$ *passes then* $\mathsf{TestBlock}^{(e+1)}(x,y,i)$ *passes.*

Proof. Immediate from the construction. □

We also need the following Lemma on the behavior of the parameters during the operation.

Lemma D.9. *The sum of all deficiencies* $\sum_{1\leq \mathsf{p}\leq h} d_{\mathsf{p}}$ *is at most* $\sqrt{k}/10 = k_{min}$ *and for every* $\mathsf{p} \leq h$*:*

- *The resliency at level* p *is larger than the sum of deficiencies in the levels below it. That is,* $\mathsf{res}_{\mathsf{p}} - \sum_{\mathsf{p}<\mathsf{p}'\leq h} d_{\mathsf{p}'} > k^{6/16}$.
- $\bar{\epsilon}_{\mathsf{p}} < 2^{-k^{-\beta/8}}$.

Proof. Follows from the way we defined the parameters. □

D.5.2 Fixing prefixes

We apply Lemma C.6 on X. Note that $t = (100C)^{1/\alpha}$. By using equations (6),(7) we get the following corollary:

Corollary D.10. *There exists a deficiency $k/10 + 4t^2 \log n \leq k/5$ subsource X^{sb} (for "somewhere blockwise") of X and a vertex $v^{\mathsf{b}} = [a,b]$ of $\mathcal{T}_{n,\mathsf{t}}$ with the following properties:*

- *The source $X^{\mathsf{sb}}_{v^{\mathsf{b}}}$ is a $2\sqrt{k}$-somewhere blockwise source.*
- *There is a value $x \in \{0,1\}^{a-1}$ such that $X^{\mathsf{sb}}_{[1,a-1]} = x$ with probability 1.*

Let $\mathsf{p}(v^{\mathsf{b}})$ denote the level of v^{b}. By the guarantee on v^{b} we have that $X^{\mathsf{sb}}_{v^{\mathsf{b}}}$ has an index i^* such that the blocks $1, \dots, i^*-1$ of $X^{\mathsf{sb}}_{v^{\mathsf{b}}}$ are fixed, the i'th block has entropy at least $2\sqrt{k}$ and for every value x_{i^*}, $\mathsf{Vaz}(X^{\mathsf{sb}}_{v^{\mathsf{b}}} | X^{\mathsf{sb}}_{v^{\mathsf{b}},i^*} = x_{i^*}) \geq 2\sqrt{k}$.

D.5.3 Making sure that TestBlock fails on "wrong nodes"

Lemma D.11. *There exists a subsource X' of X^{sb} of deficiency $\sqrt{k}/100$ and a subsource Y' of Y of deficiency $\sqrt{k}/100$ such that for every $\mathsf{p} \leq \mathsf{p}(v^{\mathsf{b}})$ every node $v \in \mathcal{L}_{v^{\mathsf{b}}}$ on level p then for any i such that the interval of v_i (recall that v_i is the i'th child of v) is contained in $[1, a-1]$*

$$\Pr[\mathsf{TestBlock}^1_{\mathsf{p}}(X', Y', i) \textit{ fails}] = 1$$

Proof. This follows because we can go over all the nodes in the left family of v^{b} and apply Theorem D.3 to get a subsource of deficiency $3w$ in both X and Y on which $\mathsf{TestBlock}^{(3)}$ fails with probability one. Since $|\mathcal{L}_{v^{\mathsf{b}}}| \leq t \cdot h \leq n^{\nu}/\nu$. Therefore the total deficiency encountered in this process is $3w \cdot n^{\nu}/\nu \leq \sqrt{k}/100$. (The last inequality holds because of equation 8). Note that X^{sb} and Y have entropy at least $2\sqrt{k}$. This means that in particular when applying Theorem D.3 even after we encountered all that deficiency the sources X and Y still have entropy at least $\sqrt{k} > d_0/10$. Note that for every p, $d_{\mathsf{p}+1} \leq d_{\mathsf{p}}$ and thus for all p, $\sqrt{k} > d_{\mathsf{p}}/10$. We have applied the TestBlock function using SSE with entropy threshold $d_{\mathsf{p}}/10$ and thus, each time we applied Theorem D.3 there was enough entropy in the corresponding sources to apply the Theorem.

□

D.5.4 Going down the path to v^{b}

We currently have subsources X' and Y' with the following properties:

- X' and Y' have min-entropy at least $\sqrt{k}$.
- For every $\mathsf{p} \leq \mathsf{p}(v^{\mathsf{b}})$ every node $v \in \mathcal{L}_{v^{\mathsf{b}}}$ on level p then for any i such that the interval of v_i (recall that v_i is the i'th child of v) is contained in $[1, a-1]$.

We need the following definition.

Definition D.12 (Arriving to v^{b})**.** We say that $x \in \{0,1\}^{n_1}$ and $y \in \{0,1\}^{n_2}$ arrive to v^{b} if on input x, y the procedure of Disp defined above visits node v^{b}.

Lemma D.13. *There exist subsources $\tilde{X}$ and $\tilde{Y}$ of X' and Y' of deficiency $\sqrt{k}/10$ such that for any node v on level $\mathsf{p} < \mathsf{p}(v^{\mathsf{b}})$ that is on the path from the root to v^{b} and any child v_i of v that is also on the path to v^{b}: $\mathsf{TestBlock}^{(3)}_{\mathsf{p}}(\tilde{X}_v, \tilde{Y}, i)$ passes with probability $1 - 2^{-\ell^{(1)}_{\mathsf{p}(v^{\mathsf{b}})} \cdot k^{\mu/100}}$ and resiliency $\mathsf{res}_{\mathsf{p}} - \sum_{\mathsf{p} < \mathsf{p}' \leq h} d_{\mathsf{p}'}$.*

Proof. We construct $\tilde{X}$ and $\tilde{Y}$ iteratively as follows: We start at the root. Let $\mathsf{p} < \mathsf{p}(v^{\mathsf{b}})$ be a level. (Initially, $\mathsf{p} = 0$) and suppose we already found subsources $X^{\mathsf{p}'}, Y^{\mathsf{p}'}$ of deficiency at most $\sum_{1 \leq j \leq \mathsf{p}'-1} d_j$ for every $\mathsf{p}' < \mathsf{p}$ (initially these are the original sources) and furthermore we assume that $\mathsf{TestBlock}^{(3)}_{\mathsf{p}'}$ passes with probability $1 - poly(n_2)(2^{-\mathsf{len}^{(3)}_{\mathsf{p}'}} + \bar{\epsilon}_{\mathsf{p}'})$ on $X^{\mathsf{p}'}, Y^{\mathsf{p}'}$ with resiliency $\mathsf{res}_{\mathsf{p}'}$ (initially this holds in an empty sense). Our goal is to construct the next subsources $X^{\mathsf{p}}, Y^{\mathsf{p}}$. For this purpose we will use Theorem D.5 (2nd item) on $\mathsf{TestBlock}^{(3)}_{\mathsf{p}}$. We first verify that we meet the requirements of the Theorem. We note that by Lemma D.9, whenever we run SSE the entropy threshold we chose (which is $\Omega(d_{\mathsf{p}})$) is larger then k_{min}. (We have to check this so we can use Theorem D.5). We also need to check that even in the subsources $X^{\mathsf{p}'}, Y^{\mathsf{p}'}$ (which have total deficiency at most $\sum_{\mathsf{p}} d_{\mathsf{p}} \leq \sqrt{k}/10$) the entropy of $X_{v^{\mathsf{b}}}, Y$ is larger than k_{min}. This holds because initially the entropy of $X_{v^{\mathsf{b}}}$ is at least $\sqrt{k}$ and $k_{min} = \sqrt{k}/10$. Thus, we can apply Theorem D.5 (2nd item) and obtain subsources $X^{\mathsf{p}}, Y^{\mathsf{p}}$ of deficiency $\mathsf{def}_{\mathsf{p}} \leq d_{\mathsf{p}}$. We have that $\mathsf{TestBlock}^{(3)}_{\mathsf{p}}$ succeeds in these subsources with probability $1 - poly(n_2)(2^{-\mathsf{len}^{(3)}_{\mathsf{p}}} + \bar{\epsilon}_{\mathsf{p}})$ and resiliency $\mathsf{res}_{\mathsf{p}}$. Thus, we can indeed continue these process until we construct $X^{\mathsf{p}(v^{\mathsf{b}})}, Y^{\mathsf{p}(v^{\mathsf{b}})}$.

For each $\mathsf{p} < \mathsf{p}(v^{\mathsf{b}})$ we have that $\mathsf{TestBlock}^{(3)}_{\mathsf{p}}$ passes with high probability on $X^{\mathsf{p}}, Y^{\mathsf{p}}$ with resiliency $\mathsf{res}_{\mathsf{p}}$. This means that it passes with the same probability in $X^{\mathsf{p}(v^{\mathsf{b}})}, Y^{\mathsf{p}(v^{\mathsf{b}})}$ with resiliency at least the original resiliency (in $X^{\mathsf{p}}, Y^{\mathsf{p}}$) minus the deficiency of $X^{\mathsf{p}(v^{\mathsf{b}})}, Y^{\mathsf{p}(v^{\mathsf{b}})}$ in $X^{\mathsf{p}}, Y^{\mathsf{p}}$. Thus, we conclude that $\mathsf{TestBlock}^{(3)}_{\mathsf{p}}$ passes with probability $1 - poly(n_2)(2^{-\mathsf{len}^{(3)}_{\mathsf{p}}} + \bar{\epsilon}_{\mathsf{p}})$ in $X^{\mathsf{p}(v^{\mathsf{b}})}, Y^{\mathsf{p}(v^{\mathsf{b}})}$ with resiliency $\mathsf{res}_{\mathsf{p}} - \sum_{\mathsf{p} < \mathsf{p}' \leq h} d_{\mathsf{p}'}$.

We now observe that for $\mathsf{p} < \mathsf{p}(v^{\mathsf{b}})$:

$$1 - poly(n_2)(2^{-\mathsf{len}^{(3)}_{\mathsf{p}}} + \bar{\epsilon}_{\mathsf{p}}) \geq 1 - 2^{-\ell^{(1)}_{\mathsf{p}(v^{\mathsf{b}})}}$$

This is because the term of level $\mathsf{p}(v^{\mathsf{b}}) - 1$ is the largest one, and so it is sufficient to prove this for it. For this purpose we note that $2^{-\mathsf{len}}$ is significantly larger the $\bar{\epsilon}$ as the len's were chosen to be smaller than $k^{-\beta/20}$ while even the largest $\bar{\epsilon}$ is smaller than $k^{-\beta/8}$. We also used the fact that $\ell^{(1)}_{\mathsf{p}(v^{\mathsf{b}})} k^{\mu} \leq \mathsf{len}^{(3)}_{\mathsf{p}}$ so that we can replace $\mathsf{len}^{(3)}_{\mathsf{p}}$ with $\ell^{(1)}{}_{\mathsf{p}(v^{\mathsf{b}})}$ and get rid of the $poly(n_2)$ term. ☐

Summary of current situation: We have found two sources $\tilde{X}, \tilde{Y}$ such that: at any level $\mathsf{p} < \mathsf{p}(v^{\mathsf{b}})$ and node v on level p that is on the path from the root to v^{b} $\mathsf{TestBlock}^{(3)}$ (and therefore $\mathsf{TestBlock}^{(1)}$ and $\mathsf{TestBlock}^{(2)}$) pass (with high probability) when checking i such that v_i is the child of v that is on the path. Furthermore, for every $i' < i$, the node $v_{i'}$ is in $\mathcal{L}_{v^{\mathsf{b}}}$ and hence fails $\mathsf{TestBlock}^{(1)}$ (and therefore $\mathsf{TestBlock}^{(2)}$ and $\mathsf{TestBlock}^{(3)}$ with probability one. This means that with high probability the child v_i is selected. Thus, it follows (by a union bound over errors on the path) that with high probability we arrive to v^{b} on $\tilde{X}, \tilde{Y}$.

D.5.5 Arguing that Disp outputs two different values

Recall that $X^{\mathsf{sb}}_{v^{\mathsf{b}}}$ has an index i^* such that the blocks $1, \ldots, i^* - 1$ of $X^{\mathsf{sb}}_{v^{\mathsf{b}}}$ are fixed, the i'th block has entropy at least $2\sqrt{k}$ and for every value x_{i^*}, $\mathsf{Vaz}(X^{\mathsf{sb}}_{v^{\mathsf{b}}} | X^{\mathsf{sb}}_{v^{\mathsf{b}}, i^*} = x_{i^*}) \geq 2\sqrt{k}$.

We have restricted our attention to a subsource of X^{sb} of total deficiency at most $\sqrt{k}/3$. Informally, it follows that the source we obtain is still a block-wise source with about the same parameters. This will allow us to prove the following Lemma.

Lemma D.14. *For $e \in \{1, 2\}$ there exist subsources $\tilde{X}'_e$ and $\tilde{Y}'_e$ such that:*

- $\Pr[\mathsf{TestBlock}^{(e)}_{\mathsf{p}(v^{\mathsf{b}})}(\tilde{X'}_{v^{\mathsf{b}}}, \tilde{Y'}, i^*)$ *fails*$] \leq 2^{-\mathsf{len}^{(e)}_{\mathsf{p}(v^{\mathsf{b}})} \cdot k^{\mu/100}}$.
- *For $e' > e$,* $\Pr[\mathsf{TestBlock}^{(e')}_{\mathsf{p}(v^{\mathsf{b}})}(\tilde{X'}_{v^{\mathsf{b}}}, \tilde{Y'}, i^*)$ *fails*$] \geq 2^{-\ell^{(e')}_{\mathsf{p}(v^{\mathsf{b}})} \cdot k^{\mu/100}}$.

- *With probability* $1-2^{-\ell^{(1)}_{\mathsf{p}(v^{\mathsf{b}})}\cdot k^{\mu/10}}$, $\tilde{X}', \tilde{Y}'$ *arrive to* v^{b}.

We first note that from this Lemma it follows that $\mathsf{Disp}(\tilde{X}', \tilde{Y}') = \{0,1\}$ which concludes the proof of the Theorem. When reading the quantities above the reader should remember that $k^{\mu/100}$ is so small that it can be safely ignored.

Lemma D.15. *For every* $b \in \{0,1\}$, $\Pr[\mathsf{Disp}(\tilde{X}, \tilde{Y}) = b] > 0$.

Proof. (of Lemma D.15) We start with $b = 0$. To show that Disp outputs 0 it is sufficient to show that with positive probability both events below happen.

- $\tilde{X}, \tilde{Y}$ arrive to v^{b}.
- $\mathsf{TestBlock}^{(1)}_{\mathsf{p}(v^{\mathsf{b}})}(\tilde{X}_{v^{\mathsf{b}}}, \tilde{Y}, i^*)$ passes.
- $\mathsf{TestBlock}^{(2)}_{\mathsf{p}(v^{\mathsf{b}})}(\tilde{X}_{v^{\mathsf{b}}}, \tilde{Y}, i^*)$ fails.

This is because by Lemma D.8 $\mathsf{TestBlock}^{(3)}$ fails whenever $\mathsf{TestBlock}^{(2)}$ fails. We also need that x, y that reached v^{b} if $\mathsf{TestBlock}^{(1)}$ passes we indeed choose i^*. Indeed, it follows from Lemma D.14 that in the subsources $\tilde{X}'_1, \tilde{Y}'_1$ the events above happen positive probability as

$$2^{-\ell^{(1)}_{\mathsf{p}(v^{\mathsf{b}})}\cdot k^{\mu/10}} < 2^{-\mathsf{len}^{(1)}_{\mathsf{p}(v^{\mathsf{b}})}\cdot k^{\mu/100}} < 2^{-\ell^{(2)}_{\mathsf{p}(v^{\mathsf{b}})}\cdot k^{\mu/100}}$$

We now consider $b = 1$. To show that Disp outputs 1 it is sufficient to show that with positive probability the three events below happen.

- $\tilde{X}, \tilde{Y}$ arrive to v^{b}.
- $\mathsf{TestBlock}^{(2)}_{\mathsf{p}(v^{\mathsf{b}})}(\tilde{X}_{v^{\mathsf{b}}}, \tilde{Y}, i^*)$ passes.
- $\mathsf{TestBlock}^{(3)}_{\mathsf{p}(v^{\mathsf{b}})}(\tilde{X}_{v^{\mathsf{b}}}, \tilde{Y}, i^*)$ fails.

This is becasue x, y that reached v^{b} if $\mathsf{TestBlock}^{(2)}$ then by Lemma D.8 $\mathsf{TestBlock}^{(1)}$ passes and we indeed choose i^*. Indeed, it follows from Lemma D.14 that in the subsources $\tilde{X}'_1, \tilde{Y}'_1$ the events above happen positive probability as

$$2^{-\ell^{(1)}_{\mathsf{p}(v^{\mathsf{b}})}\cdot k^{\mu/10}} < 2^{-\mathsf{len}^{(2)}_{\mathsf{p}(v^{\mathsf{b}})}\cdot k^{\mu/100}} < 2^{-\ell^{(3)}_{\mathsf{p}(v^{\mathsf{b}})}\cdot k^{\mu/100}}$$

□

Thus, to conclude the proof of Theorem D.1 we are left with the task of proving Lemma D.14.

Proof. (of Lemma D.14) As we have quite a bit of indices in this Lemma we use the convention that if we want to add a subscript b to X_a then we write $X_{a,b}$. Thus, for example if we want to write X_u for $u = v_i$ we write $X_{v,i}$.

We have that $\tilde{X}$ is a deficiency $\sqrt{k}/10$ subsource of X^{sb}. Recall that X^{sb}_{v,i^*} is a $2\sqrt{k}$-source. Thus, $\tilde{X}_{v,i^*}$ is a $\sqrt{k}$-source. We now consider the application of $\mathsf{TestBlock}^{(e)}$ at level $\mathsf{p}(v^{\mathsf{b}})$ when applied on $\tilde{X}'_{v^{\mathsf{b}}}$ with index i^*. We have that $H_\infty(\tilde{X}_{v^{\mathsf{b}},i}) \geq k_{min}$ and therefore we can apply Theorem D.5 (1st item) and obtain an even smaller pair of subsources $\tilde{X}'_e$ and $\tilde{Y}'_e$ of $\tilde{X}$ and $\tilde{Y}$ with deficiency $\mathsf{def}_{\mathsf{p}(v^{\mathsf{b}})}$ such that:

$$\Pr[\mathsf{TestBlock}^{(e)}(\tilde{X}'_{e,v^{\mathsf{b}}}, \tilde{Y}'_e, i^*) \text{ passes}] \geq 1 - poly(n_2)(2^{-\mathsf{len}^{(e)}_{\mathsf{p}(v^{\mathsf{b}})}} + \bar{\epsilon}_{\mathsf{p}(v^{\mathsf{b}})}) \geq 1 - 2^{-\mathsf{len}^{(e)}_{\mathsf{p}(v^{\mathsf{b}})}\cdot k^{\mu/100}} \quad (10)$$

This proves the first item. For the 2nd item we use Theorem D.6 on $\mathsf{TestBlock}^{(e')}_{\mathsf{p}(v^{\mathsf{b}})}(\tilde{X}'_{e,v^{\mathsf{b}}}, \tilde{Y}'_e, i^*)$. We need to verify that there exists a set $G \subseteq \{0,1\}^{n_{\mathsf{p}(v^{\mathsf{b}})}/t}$ such that:

- $\Pr[\tilde{X'}_{e,v^{\mathsf{b}},i^*} \in G] \geq 1/2$.

- For every $x_{i^*} \in G$, $H_\infty(\tilde{X'}_{e,v^{\mathsf{b}}} | \tilde{X'}_{e,v^{\mathsf{b}},i^*} = x_{i^*}) \geq \sqrt{k} \geq w$.

To prove this, note that $\tilde{X'}_e$ is a deficiency at most $\sqrt{k}/10$ of X^{sb}. By Corollary D.10 we have that X^{sb} is a block-wise source with entropy $2\sqrt{k}$ in the i^*'th block and conditional entropy of $2\sqrt{k}$ in one of the following blocks. It follows that: $H_\infty(X^{\mathsf{sb}}_{v^{\mathsf{b}},i^*}) \geq 2\sqrt{k}$ and for every value $x_{i^*} \in \{0,1\}^{n_{\mathsf{p}(v^{\mathsf{b}})}/t}$, $(X^{\mathsf{sb}} | X^{\mathsf{sb}}_{i^*} = x_{i^*})$ is a $2\sqrt{k}$-source. Thus, having lost deficiency at most $\sqrt{k}/10$ the two items above follow from the following claim:

Claim D.16. *If X has the property that $H_\infty(X_i) \geq k$ and for every x_i, $(X|X_i = x_i)$ is a k-source and let X' be a deficiency d subsource of X. Then there exists a set G such that:*

- $\Pr[X' \in G] \geq 1/2$.

- *For every $x_i \in G$, $H_\infty(X'|X'_i = x_i) \geq k - 1$.*

Proof. (of claim) Let A be the event such that $X' = (X|A)$. Let $G = \{x_i : \Pr[A|X'_i = x_i] \geq 1/2\}$. The first item immediately follows. For the second one note that for $x \in A$:

$$\Pr[X' = x|X'_i = x_i] = \Pr[X = x|X_i = x_i, X \in A] = \frac{\Pr[X = x \cap X \in A|X_i = x_i]}{\Pr[X \in A|X_i = x_i]}$$

$$= \frac{\Pr[X = x|X_i = x_i]}{\Pr[X \in A|X_i = x_i]} \leq 2^{-(k-1)}$$

$\square$

Applying Theorem D.6 we conclude that:

$$\Pr[\mathsf{TestBlock}^{(e')}_{\mathsf{p}(v^{\mathsf{b}})}(\tilde{X'}_{e,v^{\mathsf{b}}}, \tilde{Y}'_e, i^*) \text{ fails}] \geq 2^{-4\ell^{(e')}_{\mathsf{p}(v^{\mathsf{b}})}} \geq 2^{-\ell^{(e')}_{\mathsf{p}(v^{\mathsf{b}})} \cdot k^{\mu/100}}$$

and this proves the 2nd item. We now move to the third item. By Lemma D.13 we have that for any node v on level $\mathsf{p} < \mathsf{p}(v^{\mathsf{b}})$ that is on the path from the root to v^{b} and any child v_i of v that is also on the path to v^{b}: $\mathsf{TestBlock}^{(3)}_{\mathsf{p}}(\tilde{X}_v, \tilde{Y}, i)$ passes with probability $1 - 2^{-\ell^{(1)}_{\mathsf{p}(v^{\mathsf{b}})} \cdot k^{\mu/100}}$ and resiliency $\mathsf{res}_{\mathsf{p}} - \sum_{\mathsf{p} < \mathsf{p}' \leq h} d_{\mathsf{p}'}$. We have restricted $\tilde{X}$ and $\tilde{Y}$ to subsources $\tilde{X}'_e, \tilde{Y}'_e$ of deficiency $\mathsf{def}_{\mathsf{p}(v^{\mathsf{b}})}$. Thus, as for every p, $\mathsf{res}_{\mathsf{p}} - \sum_{\mathsf{p} < \mathsf{p}' \leq h} d_{\mathsf{p}'} \geq \mathsf{res}_{\mathsf{p}}/2 \geq d_{\mathsf{p}+1} \geq \mathsf{def}_{\mathsf{p}+1}$ we have enough resiliency to conclude that in each node v above $\mathsf{TestBlock}^{(3)}_{\mathsf{p}}(\tilde{X}'_{ev}, \tilde{Y}'_e, i)$ passes with probability $1 - 2^{-\ell^{(1)}_{\mathsf{p}(v^{\mathsf{b}})} \cdot k^{\mu/100}}$.

Note that when all these events occur we reach v^{b}. Thus, by a union bound over the small errors above we get that with probability $1 - 2^{-\ell^{(1)}_{\mathsf{p}(v^{\mathsf{b}})} \cdot k^{\mu/10}}$, v^{b} is reached. This completes the proof of the third item.

$\square$