Electronic Colloquium on Computational Complexity, Revision 5 of Report No. 119 (2015)

Explicit Two-Source Extractors and Resilient Functions

Eshan Chattopadhyay*
Department of Computer Science,
Cornell University
eshan@cs.cornell.edu

David Zuckerman†
Department of Computer Science,
University of Texas at Austin
diz@cs.utexas.edu

January 7, 2019

Abstract

We explicitly construct an extractor for two independent sources on n bits, each with min-entropy at least $\log^C n$ for a large enough constant C. Our extractor outputs one bit and has error $n^{-\Omega(1)}$. The best previous extractor, by Bourgain, required each source to have min-entropy $.499n$.

A key ingredient in our construction is an explicit construction of a monotone, almost-balanced Boolean function on n bits that is resilient to coalitions of size $n^{1-\delta}$, for any $\delta > 0$. In fact, our construction is stronger in that it gives an explicit extractor for a generalization of non-oblivious bit-fixing sources on n bits, where some unknown $n - q$ bits are chosen almost polylog(n)-wise independently, and the remaining $q = n^{1-\delta}$ bits are chosen by an adversary as an arbitrary function of the $n - q$ bits. The best previous construction, by Viola, achieved $q = n^{1/2-\delta}$.

Our explicit two-source extractor directly implies an explicit construction of a $2^{(\log\log N)^{O(1)}}$-Ramsey graph over N vertices, improving bounds obtained by Barak et al. and matching an independent work by Cohen.

*This work was done when the author was a PhD student at the Univeristy of Texas at Austin partially supported by NSF Grant CCF-1218723 and CCF-1526952.

†Supported by NSF Grants CCF-1218723, CCF-1526952, and CCF-1705028, and a Simons Investigator Award (#409864).

ISSN 1433-8092

1 Introduction

We explicitly construct three related combinatorial objects: Ramsey graphs, bipartite Ramsey graphs, and two-source extractors. We do this by constructing an object that may seem unrelated, a resilient function. We begin by defining the first three objects and deferring the fourth to Section 1.2. We start with the combinatorial motivation, and discuss the computer science and randomness motivation in the next subsection.

In 1930, Ramsey [Ram30] showed that any graph on N nodes has a clique or independent set of size $(\log_2 N)/2$. In 1947, Erdös [Erd47] used the probabilistic method to show that there exist graphs on N nodes with no clique or independent set of size $2\log_2 N$. We call such a graph a Ramsey graph.

Definition 1.1 (Ramsey graph)**.** *An undirected graph on N vertices is called a K-*Ramsey graph *if it does not contain any independent set or clique of size K.*

It is natural to ask for an explicit construction, and indeed Erdös offered $100 for an explicit construction achieving $K = O(\log N)$. In what follows, we use the strongest computer science definition of explicit, namely, that there is a polynomial-time algorithm that determines whether there is an edge between two nodes. Since a node can be described using $\log N$ bits, this means that the algorithm's running time is polynomial in $\log N$.

Lindsey's lemma implies that a symmetric Hadamard matrix defines a $\sqrt{N}$-Ramsey graph on N vertices. There were constructions achieving smaller powers of N, until Frankl and Wilson [FW81] used intersection theorems to construct K-Ramsey graphs on N vertices, with $K = 2^{O(\sqrt{\log N \log\log N})}$. This remained the best known construction for a long time, with many other constructions [Alo98, Gro00, Bar06] achieving the same bound. Gopalan [Gop14] explained why approaches were stuck at this bound, showing that apart from [Bar06], all other constructions can be seen as derived from low-degree symmetric representations of the OR function. Finally, subsequent works by Barak et al. [BKS+10, BRSW12] obtained a significant improvement and gave explicit constructions of K-Ramsey graphs, with $K = 2^{2^{\log^{1-\alpha}(\log N)}}$, for some absolute constant α.

Next we define the related bipartite Ramsey graphs.

Definition 1.2 (Bipartite Ramsey graph)**.** *A bipartite graph with N left vertices and N right vertices is called a bipartite K-Ramsey graph if it does not contain any complete $K \times K$-bipartite subgraph or empty $K \times K$ subgraph.*

While non-explicit existence bounds for bipartite Ramsey graphs are similar to those for ordinary Ramsey graphs, constructing bipartite Ramsey graphs is harder. In particular, Barak et al. [BKS+10] showed how to use a given bipartite K-Ramsey graph on two sets of N vertices to construct a $2K$-Ramsey graph on N vertices. Moreover, the above constructions of ordinary Ramsey graphs with $K = 2^{O(\sqrt{\log N \log\log N})}$ do not work in the bipartite setting. In fact, until 2004, the best known construction was the Hadamard matrix, giving $K = \sqrt{N}$. This was slightly improved to $O(\sqrt{N}/2^{\sqrt{\log N}})$ by Pudlak and Rődl [PR04]. Barak et al. [BKS+10, BRSW12] did in fact construct bipartite K-Ramsey graphs, and hence achieved the bound mentioned above.

Finally, we define two-source extractors. These are like bipartite Ramsey graphs, except now we require the "right" number of edges between sets of size K, rather than at least one edge and one non-edge. For reasons that will become clearer in the next section when we discuss the computer science motivation, the parameter is defined as $k = \log_2 K$ rather than K.

Definition 1.3 (2-source extractor, graph formulation)**.** *A bipartite graph with N left vertices and N right vertices is called a (k, ε) two-source extractor if every $K \times K$ subgraph contains $(1/2 \pm \varepsilon)K^2$ edges, where $K = 2^k$.*

Observe that this is equivalent to an $N \times N$ matrix over $\{0, 1\}$ such that every $K \times K$ submatrix has $1/2 \pm \varepsilon$ fraction of 1's.

Clearly, a (k, ε)-two-source extractor is a bipartite 2^k-Ramsey graph if $\varepsilon < 1/2$. Two-source extractors have been even harder to construct than bipartite Ramsey graphs. A simple probabilistic argument shows the existence of (k, ε)-two-source extractors with $k \geq \log n + 2\log(1/\epsilon) + 1$. Chor and Goldreich [CG88] first defined these objects and used Lindsey's Lemma to show that a Hadamard matrix is a 2-source extractor for $k > n/2$. However, no further progress was made for around 20 years, when Bourgain [Bou05] broke the "half-barrier" and constructed a 2-source extractor for $k = (1/2 - \alpha)n$ for some small, unspecified $\alpha > 0$. This remained the best known result prior to our work.

Our main result is a two-source extractor for $k = \text{polylog}(n)$. We think of $\varepsilon = 1/\text{poly}(n)$ but state it more generally.

Theorem 1 (Main theorem)**.** *There exists a constant $C > 0$ such that for all $n \in \mathbb{N}$, there is a construction of a (k, ε)-two-source extractor on two sets of 2^n vertices with $k = \log^C(n/\varepsilon)$. It is explicit in that there is an algorithm running in time $\text{poly}(n/\varepsilon)$ that determines whether there is an edge between two nodes.*

Specifically, we can take $k = C_1(\log n)^{56}$, for a large enough constant C_1[1]. As corollaries, we get bipartite Ramsey graphs and ordinary Ramsey graphs.

Theorem 2. *There exists a constant $C > 0$ such that for all large enough $n \in \mathbb{N}$, there are explicit constructions of a bipartite K-Ramsey graph on $2N$ vertices and a Ramsey graph on N vertices, where $N = 2^n$ and $K = 2^{(\log\log N)^C}$.*

Independent work: In independent work, Cohen [Coh16b] used the challenge-response mechanism introduced in [BKS+10] with new advances in extractor constructions to obtain explicit constructions of bipartite Ramsey graphs with $K = 2^{(\log\log N)^{O(1)}}$. This is the same bound that we achieve. However, Cohen's construction is not a two-source extractor.

1.1 Randomness Extraction

Here we develop a computational perspective about extracting randomness, which makes the proof intuition clearer. The area of randomness extraction addresses the problem of obtaining nearly uniform bits from sources that are only weakly random. This is motivated by the ubiquitous use of randomness in many branches of computer science. Randomness is essential for cryptography and distributed computing. Many randomized algorithms, such as those to factor polynomials over large fields, are faster or simpler than their deterministic counterparts. Scientists and economists use randomness extensively in Monte Carlo simulations of complex systems like the climate or the economy.

Almost all of these uses of randomness require uniformly random, uncorrelated bits, but most easily-obtainable sources of randomness do not satisfy these conditions. In particular, programmers

[1] In the preliminary version of the work, we required $k = C_1(\log n)^{74}$ in Theorem 1. The improved bound in the current version is a result of recent improvements in constructions of *non-malleable extractors*, a key component in our construction that we describe later.

in practice try to accumulate entropy by using thermal noise or clock drift, but then this needs to be purified before using it to seed a pseudorandom generator; see e.g., [JK99, BH05].

As is standard, we model a weak source on n bits using min-entropy. A source $\mathbf{X}$ on n bits is said to have min-entropy at least k if for any x, $\mathbf{Pr}[\mathbf{X} = x] \leq 2^{-k}$.

Definition 1.4. *The min-entropy of a source* $\mathbf{X}$ *is* $H_\infty(\mathbf{X}) = \min_x(-\log(\Pr[\mathbf{X} = x]))$. *The min-entropy rate of a source* $\mathbf{X}$ *on* $\{0,1\}^n$ *is* $H_\infty(\mathbf{X})/n$. *A source* $\mathbf{X}$ *on* $\{0,1\}^n$ *with min-entropy at least* k *is called an* (n,k)*-source.*

An extractor $\mathrm{Ext} : \{0,1\}^n \to \{0,1\}^m$ is a deterministic function that takes input from an unknown weak source with sufficient min-entropy and produces nearly uniform bits. Unfortunately, a simple argument shows that it is impossible to design an extractor to extract even 1 bit for sources with min-entropy $n-1$. Specifically, one of $\mathrm{Ext}^{-1}(0)$ or $\mathrm{Ext}^{-1}(1)$ has size at least 2^{n-1}. If $\mathbf{X}$ is the uniform distribution on that set, then $\mathrm{Ext}(X)$ is always the same fixed value, but X has min-entropy at least $n-1$, contradicting the extractor requirement.

Broadly speaking, there are three approaches to circumvent this difficulty. First, one can add a small amount of high-quality randomness, called a seed, and extract out a much larger amount. Second, one can limit consideration to sources that have some structure, defined in algebraic or computational terms. We take a third approach: assume that there are two or more independent sources, each with sufficient min-entropy.[2] Santha and Vazirani [SV86] suggested this for a different but related model, and Chor and Goldreich [CG88] suggested it for our current model. A two-source extractor extracts randomness from two independent sources. An efficient two-source extractor could be quite useful in practice, if just two independent sources of entropy can be found.

We use the notion statistical distance, or variation distance, to measure the error of the extractor. The statistical distance between two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ over some universal set Ω is defined as $|\mathcal{D}_1 - \mathcal{D}_2| = \frac{1}{2}\sum_{d\in\Omega} |\mathbf{Pr}[\mathcal{D}_1 = d] - \mathbf{Pr}[\mathcal{D}_2 = d]|$. We say $\mathcal{D}_1$ is ϵ-close to $\mathcal{D}_2$ if $|\mathcal{D}_1 - \mathcal{D}_2| \leq \epsilon$ and denote it by $\mathcal{D}_1 \approx_\epsilon \mathcal{D}_2$.

Definition 1.5 (Two-source extractor)**.** *A function* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *is called a two-source extractor for min-entropy* k *and error* ϵ *if for any independent* (n,k)*-sources* $\mathbf{X}$ *and* $\mathbf{Y}$*, we have*

$$\mathrm{Ext}(\mathbf{X}, \mathbf{Y}) \approx_\varepsilon \mathbf{U}_m,$$

where $\mathbf{U}_m$ *is the uniform distribution on* m *bits. Further,* Ext *is said to be strong in* $\mathbf{Y}$ *if it also satisfies* $(\mathrm{Ext}(\mathbf{X}, \mathbf{Y}), \mathbf{Y}) \approx_\varepsilon (\mathbf{U}_m, \mathbf{Y})$*, where* $\mathbf{U}_m$ *is independent from* $\mathbf{Y}$*.*

It is straightforward to verify that this corresponds to the graph-theoretic formulation in Definition 1.3 when $m = 1$.

As mentioned above for the case $m = 1$, a simple probabilistic argument shows the existence of 2-source extractors for min-entropy $k \geq \log n + 2\log(1/\epsilon) + 1$. However, from a computer science perspective, it is important that the function Ext be efficiently computable, i.e., polynomial-time computable. This corresponds to the same notion of explicitness introduced in the graph-theoretic setting.

This question has drawn a lot of attention in the last three decades. Recapping the history above, Chor and Goldreich [CG88] used Lindsey's Lemma to show that the inner-product function is a 2-source extractor for min-entropy more than $n/2$. Using additive combinatorics, Bourgain [Bou05]

[2] The first and third approaches could also be viewed as special cases of the second approach, but we don't view this as helpful.

broke the "half-barrier" for min-entropy, and constructed a 2-source extractor for min-entropy $0.499n$. Raz [Raz05] obtained an improvement in terms of total min-entropy, and constructed 2-source extractors requiring one source with min-entropy more than $n/2$ and the other source with min-entropy $C \log n$.

The lack of progress on constructing two-source extractors motivated researchers to use more than two sources. Several researchers managed to construct excellent extractors using a constant number of sources [BIW06, Rao09a, RZ08, Li11, Li13a, Li13b] culminating in Li's construction of a 3-source extractor for polylogarithmic min-entropy [Li15]. Recently Cohen [Coh15] also constructed a 3-source extractor with one source having min-entropy δn, the second source having min-entropy $C \log n$ and the third source having min-entropy $C \log \log n$.

Summarizing, despite much attention and progress over the last 30 years, it remained open to explicitly construct two-source extractors for min-entropy rate significantly smaller than $1/2$. Our main result is an explicit two-source extractor for polylogarithmic min-entropy. We restate our main theorem in computer science terminology.

Theorem 1 [Main theorem, computer science formulation] There exists a constant $C > 0$ such that for all $n, k \in \mathbb{N}$ and any $\epsilon > 0$, satisfying $\log^C(n/\epsilon) \leq k \leq n$, there exists a 2-source extractor $2\text{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ computable in time $\text{poly}(n, 1/\epsilon)$ for min-entropy at least k and error ϵ.

As mentioned earlier, it is in fact enough to take $k = C_1(\log(n/\epsilon))^{56}$ in the above theorem for a large enough constant C_1.

By an argument of Barak [Rao09b], every 2-source extractor outputting 1 bit is also a strong 2-source extractor with similar parameters. Thus the extractor 2Ext in Theorem 1 is also a strong 2-source extractor.

Note that if our extractor is to run in polynomial time, then the error won't be negligible, meaning smaller than the reciprocal of any polynomial. Improving the error to negligible while outputting many bits would have applications in cryptography and distributed computing. For example, several researchers have studied whether cryptographic or distributed computing protocols can be implemented if the players' randomness is defective [DO03, GSV05, KLRZ08, KLR09]. Kalai et al. [KLRZ08] used C-source extractors to build network extractor protocols, which allow players to extract private randomness in a network with Byzantine faults. A better 2-source extractor with negligible error would improve some of those constructions. Kalai, Li, and Rao [KLR09] showed how to construct a 2-source extractor under computational assumptions, and used it to improve earlier network extractors in the computational setting; however, their protocols rely on computational assumptions beyond the 2-source extractor, so it would not be clear how to match their results without assumptions.

Subsequent Work: There have been many exciting developments after our work. Li [Li16] extended our construction to achieve an explicit strong 2-source extractor with output length $\Omega(k)$ bits. A sequence of works [Mek17, BDT17, CL16a, Coh16a, Coh17, Li17, Li18] built on our framework, improving the various components used, to lower the min-entropy requirement of the 2-source extractor (for constant error) to $C \log n (\log \log n)/\log \log \log n$.

Li used our construction to build an affine extractor for polylogarithmic min-entropy [Li16]. In another work, Chattopadhyay and Li [CL16b] used components from our construction to construct extractors for sumset sources, which allowed them to give improved extractors for sources that are generated by algorithms with access to limited memory. Ben-Aroya, Doron and Ta-Shma [BDT18] used components from our work to construct 2-source extractors, when the sources are of unequal length, and found interesting applications of these extractors to the theory of error-correcting codes.

Ben-Aroya et al. [BCDT18] extended our construction to give explicit constructions of 2-source condensers[3] with exponentially small error.

1.2 Resilient Functions

We assume basic familiarity with circuit complexity in this section. We refer the reader to Section 2.6 for a quick recap of relevant notions that are used in this section.

As part of our construction of two-source extractors, we construct new "resilient functions", which are interesting in their own right. Ben-Or and Linial [BL85] first studied resilient functions when they introduced the perfect information model of distributed computation. In the simplest version of this model, there are n computationally unbounded players that can each broadcast a bit once. At the end, some function is applied to the broadcast bits. In the collective coin-flipping problem, the output of this function should be a nearly-random bit. The catch is that some malicious coalition of players may wait to see what the honest players broadcast before broadcasting their own bits. Thus, a resilient function is one where the bit is unbiased even if the malicious coalition is relatively large (but not too large). We now introduce this notion more formally.

Definition 1.6 (Influence)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function on variables $x_1, \ldots, x_n$. The influence of a set $Q \subseteq \{x_1, \ldots, x_n\}$ on f, denoted by $\mathbf{I}_Q(f)$, is defined to be the probability that f is undetermined after fixing the variables outside Q uniformly at random. Further, for any integer q define $\mathbf{I}_q(f) = \max_{Q \subseteq \{x_1,\ldots,x_n\}, |Q|=q} \mathbf{I}_Q(f)$.*

Definition 1.7 (Resilient Function)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function on variables $x_1, \ldots, x_n$ and q any integer. We say f is (q, ϵ)-resilient if $\mathbf{I}_q(f) \leq \epsilon$.*

For example, the PARITY function (which outputs the sum modulo 2 of the input bits) is not even $(1, \varepsilon)$-resilient for any $\varepsilon < 1$. The constant function $f(x) = 0$ is $(n, 0)$-resilient, but uninteresting; we want a function that is almost balanced in the sense of having probability close to $1/2$ of being 1. The most natural choice here is MAJORITY (which outputs the majority of its input bits), which is $(c\sqrt{n}, .01)$ resilient. Somewhat surprisingly, there are almost-balanced functions significantly more resilient than MAJORITY. Ben-Or and Linial [BL85] showed that the iterated majority function is $(cn^{\log_3 2}, .01)$-resilient. Furthermore, Ajtai and Linial showed the existence of almost-balanced functions that are $(cn/\log^2 n, .01)$-resilient. Kahn, Kalai, and Linial [KKL88] showed that no function is $(\omega(n/\log n), .99)$-resilient.[4]

Since the Ajtai-Linial construction is not explicit, it is natural to ask whether we can come close to this bound explicitly. In unpublished work, Meka showed that for any $\delta > 0$, a suitable iteration of small Ajtai-Linial functions is $(n^{1-\delta}, .01)$-resilient [Mek09]. While this requires a brute-force search to find the small Ajtai-Linial function, it is explicit because this function is on few bits. We derandomize Ajtai-Linial without any small brute-force search. Moreover, our construction is monotone and is computable by a constant-depth circuit. Neither Ajtai-Linial nor Meka's constructions have these properties, and both of these properties are necessary for our use in the two-source extractor construction.

Theorem 3 (Explicit resilient function)**.** *For any constant $\delta \in (0,1)$ and every large enough integer $n \in \mathbb{N}$, there exists a polynomial-time computable monotone Boolean function $f : \{0,1\}^n \to \{0,1\}$ satisfying:*

[3] the notion of a condensers is weaker than an extractor, and the output is only required to have high-min-entropy

[4] Recall that $f(n) = \omega(g(n))$ means that $\lim_{n\to\infty} f(n)/g(n) = \infty$.

- *f is a depth 4 circuit of size $n^{O(1)}$.*
- *$|\mathbf{E}[f(\mathbf{x})] - \frac{1}{2}| \leq \frac{1}{n^{\Omega(1)}}$.*
- *For any $q > 0$, we have $\mathbf{I}_q(f) \leq q/n^{1-\delta}$.*

1.2.1 Resilient Functions against t-wise Independence

In fact, our two-source extractor construction requires a stronger notion of resiliency, which is also interesting on its own. This is where we allow the $n - q$ good bits to be chosen from an arbitrary t-wise independent distribution.

Definition 1.8. *A distribution $\mathcal{D}$ on n bits is t-wise independent if the restriction of $\mathcal{D}$ to any t bits is uniform. More generally, $\mathcal{D}$ is (t, γ)-wise independent if the distribution obtained by restricting $\mathcal{D}$ to any t coordinates is γ-close to uniform.*

When $\mathcal{D}$ is a (t, γ)-wise distribution, we sometimes informally say that $\mathcal{D}$ is an *almost t-wise independent* distribution.

Definition 1.9 (Influence, general)**.** *Let $\mathbf{I}_{Q,\mathcal{D}}(f)$ denote the probability that f is undetermined when the variables outside Q are fixed by sampling from the distribution $\mathcal{D}$. Define $\mathbf{I}_{Q,t}(f) = \max_{\mathcal{D} \in \mathcal{D}_t} \mathbf{I}_{Q,\mathcal{D}}(f)$, where $\mathcal{D}_t$ is the set of all t-wise independent distributions. Finally, for any integer q, define $\mathbf{I}_{q,t}(f) = \max_{Q \subseteq \{x_1,\ldots,x_D\}, |Q|=q} \mathbf{I}_{Q,t}(f)$.*

Definition 1.10 (Resilient Function, general)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function on variables $x_1, \ldots, x_n$ and q any integer. We say f is t-independent (q, ϵ)-resilient if $\mathbf{I}_{q,t}(f) \leq \epsilon$.*

Viola [Vio14] first studied this model in the context of constructing extractors for circuit sources. He showed that the majority function is $O(1)$-independent $(q, .01)$-resilient for $q < D^{\frac{1}{2}-\tau}$, $\tau > 0$. No other t-independent resilient functions were known for $t < \sqrt{n}$. We show that any almost-balanced resilient function that is monotone and constant depth remains almost-balanced and resilient with respect to polylog-wise independence. Therefore, our explicit almost-balanced resilient function remains almost-balanced and resilient with respect to polylog-wise independence.

Theorem 4. *There exists a constant c such that for any constant $\delta \in (0,1)$ and every large enough integer $n \in \mathbb{N}$, there exists an efficiently computable monotone Boolean function $f : \{0,1\}^n \to \{0,1\}$ satisfying: For any $q > 0, t \geq c(\log n)^{18}$,*

- *f is a depth 4 circuit of size $n^{O(1)}$.*
- *For any t-wise independent distribution $\mathcal{D}$, $|\mathbf{E}_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x})] - \frac{1}{2}| \leq \frac{1}{n^{\Omega(1)}}$.*
- *$\mathbf{I}_{q,t}(f) \leq q/n^{1-\delta}$.*

Subsequent Work: Meka [Mek17] built on our ideas to give an explicit construction of a monotone almost-balanced resilient function that is polylog-independent $cn/\log^2 n$-resilient, matching the non-explicit resiliency obtained by Ajtai and Linial (except Ajtai and Linial achieved ordinary resiliency, not polylog-independent resiliency).

1.3 Seeded and Non-Malleable Extractors

Before describing our construction, we define two important ingredients, seeded extractors [NZ96] and non-malleable extractors [DW09]. A seeded extractor uses one (n, k)-source and a short seed to extract randomness. Seeded extractors have found numerous applications in seemingly unrelated areas; see e.g., Shaltiel's survey [Sha02].

Definition 1.11 ([NZ96])**.** *A function* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k, ε)*-seeded extractor if for any source* $\mathbf{X}$ *of min-entropy* k*, we have* $\text{Ext}(\mathbf{X}, \mathbf{U}_d) \approx_\varepsilon \mathbf{U}_m$*.* Ext *is strong if we have* $(\text{Ext}(\mathbf{X}, \mathbf{U}_d), \mathbf{U}_d) \approx_\varepsilon (\mathbf{U}_m, \mathbf{U}_d)$*, where* $\mathbf{U}_m$ *and* $\mathbf{U}_d$ *are independent.*

We use explicit constructions of seeded extractors with almost optimal parameters: $d = O(\log(n/\epsilon))$ and $m = \Omega(k)$ [LRVW03, GUV09, DKSS09].

Note that Ext being strong implies that the output of Ext is close to uniform even conditioned on the seed (with high probability). Specifically, we can view a strong extractor as consisting of $D = 2^d$ functions $f_s : \{0,1\}^n \to \{0,1\}^m$ where $f_s(x) = \text{Ext}(x, s)$. By an averaging argument, the strong extractor property ensures that for any source $\mathbf{X}$ of min-entropy k, for at least $1 - \sqrt{\varepsilon}$ of the seeds s, we have $f_s(\mathbf{X}) \approx_{\sqrt{\varepsilon}} \mathbf{U}_m$.

A non-malleable extractor is a strengthening of a strong seeded extractor where these outputs are not only uniform, but almost t-wise independent. Instead of giving the proper definition as given by Dodis and Wichs [DW09], here we state the property that we need. Let $\text{nmExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ be a (t, k, ϵ)-non-malleable extractor that outputs 1 bit with seed-legnth d, and set $D = 2^d$. We show in Lemma 2.17, that there exists a large subset of seeds $S \subseteq \{0,1\}^d$, $|S| \geq (1 - O(\sqrt{\epsilon}))D$, such that for any t distinct seeds $s_1, \ldots, s_t$ in S, we have

$$(\text{nmExt}(\mathbf{X}, s_1), \ldots, \text{nmExt}(\mathbf{X}, s_t)) \approx_\delta U_t,$$

where $\delta = O(t\sqrt{\epsilon})$. In other words, if we define $f_s(x) = \text{nmExt}(x, y)$, then not only are almost all $f_s(\mathbf{X})$ close to uniform, but almost all of them are almost t-wise independent.

Non-malleable extractors are much stronger than seeded extractors; for example, a non-malleable extractor can't ignore even one bit of its seed, or else the seeds could be grouped in pairs with the same functionality. The first constructions [DLWZ14, CRS14, Li12b] worked only for min-entropy at least $.49n$. The first construction to break this barrier was in the work of Chattopadhyay, Goyal, and Li [CGL16] who constructed non-malleable extractors requiring min-entropy $k = Ct \log^2(n/\epsilon)$ and seed-length $d = O(t^2 \log^2(n/\epsilon))$. Subsequently, there have been further improvements and we use the state-of-art construction from the work of Li [Li18] (see Theorem 2.19). We will end up needing t to be polylogarithmic, so the min-entropy and seed-length will both be polylogarithmic.

1.4 Construction Overview

Previous Techniques

As mentioned earlier, Bourgain's 2-source extractor for min-entropy $0.499n$ relied on new advances in additive combinatorics. Following this, Rao [Rao09a] introduced a novel elementary approach for extracting from multiple independent sources that relied on only explicit seeded extractors. His approach was to first convert the independent sources into matrices with many uniformly random rows, called somewhere-random sources, and then iteratively reduce the number of rows in one of the somewhere-random sources (while still maintaining a good fraction of uniform rows) using

the other somewhere-random sources. This allowed him to construct an explicit extractor for a constant number of sources with min-entropy n^γ for any constant $\gamma > 0$.

In a series of works [Li13b, Li13a, Li15], Li introduced a new way of iteratively reducing the number of rows in the somewhere-random sources. His idea was to use a few independent sources to construct a more structured somewhere-random source with the additional guarantee that the uniform rows are t-wise independent and then iteratively reduce the number of rows using leader election protocols from the work of Feige [Fei99]. Using this approach and clever compositions of extractors, Li [Li15] constructed an explicit extractor for 3 independent sources with polylogarithmic min-entropy.

We note that Li [Li15] had already shown how to use two independent sources to construct a single string where 2/3 of the bits are close to uniform, and all of these good bits are almost polylog-wise independent. By using a better seeded extractor, we could obtain a single string where at least a $(1 - n^{-\Omega(1)})$ fraction of the bits are almost polylog-wise independent.

Our Approach

There are three technical parts to our construction.

- First, we show how to use a non-malleable extractor to reduce 2 independent sources $\mathbf{X}$ and $\mathbf{Y}$, each on n bits, to a single string $\mathbf{Z}$ on poly(n) bits such that $(1 - n^{-\Omega(1)})$-fraction of the bits are almost polylog(n)-wise independent. This gives the same reduction as done by Li [Li15], but our construction is more modular. Li did not use non-malleable extractors, but instead used alternating extraction in clever ways. Our modularization led to the later improvements of two-source extractors.

  Further, we observe that a (q, t)-resilient function (see Definition 1.10), for appropriate parameters, is an extractor for the source $\mathbf{Z}$.

  We sketch our reduction using a non-malleable extractor and extraction using resilient functions in Subsection 1.4.1.

- Second, we show that a monotone resilient function in AC^0 is also resilient when the good bits are chosen polylog(n)-wise independently. (Recall that a function is in AC^0 if it is computable by a family of polynomial-sized circuits with constant depth, allowing unlimited fan-in AND and OR gates.) We discuss this in Subsection 1.4.2.

- Third, we show how to construct a monotone resilient function in AC^0. This is described in Subsection 1.4.3.

We note that Li did not use resilient functions, but instead iteratively used leader election protocols, which is why he obtained a 3-source extractor instead of a 2-source extractor.

1.4.1 A Non-Malleable Extractor and a Resilient Function Give a Two-source Extractor

To motivate our construction of 2-source extractors, let's first try to build a 1-source extractor (even though we know it is impossible). Let $\mathbf{X}$ be an (n, k)-source, where $k = \text{polylog}(n)$. Let Ext be a strong seeded extractor designed to extract 1 bit from min-entropy k with error ϵ, and view it as D functions $f_s(x) = \text{Ext}(x, s)$. From the earlier discussion about extractors, the concatenation of $f_s(X)$ over all seeds s gives a D-bit string $\mathbf{Z}$ where most individual bits are close to uniform.

Note that since the seed length of Ext is $O(\log n)$, $D = \text{poly}(n)$ (think of the error parameter $\epsilon = 1/n^{O(1)}$). At this point, we might hope to take the majority of these D bits of $\mathbf{Z}$ to obtain a bit that is close to uniform. However, the bits $f_s(X)$ for different seeds s may be correlated in arbitrary ways (even if individually the bits are close to uniform), so this approach doesn't work.

We can try to fix this approach by introducing some independence among the uniform bits. For example, if we obtain a source $\mathbf{Z}$ such that $D - D^{0.49}$ bits are uniform, and further these bits are (almost) constant-wise independent, then it is known that the majority function can extract an almost-uniform bit [Vio14]. In an attempt to obtain such a source, we use a non-malleable extractor. Let nmExt be a (t, k, ϵ)-non-malleable extractor that outputs 1 bit with seed-length d, and let $D = 2^d$. We proceed as in our first attempt, viewing the non-malleable extractor as D functions $f_s(x) = \text{Ext}(x, s)$. From the earlier discussion about non-malleable extractors, the concatenation of $f_s(X)$ over all seeds s gives a D-bit string $\mathbf{Z}$ where not only are most individual bits close to uniform, but almost all the bits are also almost t-wise independent. We could now try to set parameters so that the majority function extracts a bit from $\mathbf{Z}$. However, the majority function is resilient to at most $\sqrt{D}$ bad bits, but the number of bad bits in $\mathbf{Z}$ exceeds that (since $D > 1/\varepsilon^2$).

It is therefore natural to use a more resilient function. Specifically, we can use our new explicit resilient function that is resilient against $D^{1-\delta}$ bad bits, even if the good bits are only polylog-wise independent, for our choice of $\delta > 0$. We can indeed ensure that there are at most $D^{1-\delta}$ bad bits, and that the good bits are almost polylog-wise independent. The problem is that they are not exactly polylog-wise independent, but almost polylog-wise independent with too large an error. Specifically, we want to use a lemma of Alon, Goldreich, and Mansour [AGM03] saying that if every restriction of $\mathbf{Z}$ to t bits is γ-close to uniform, then the entire string $\mathbf{Z}$ is $D^t\gamma$-close to some t-wise independent distribution. The problem is that $D^t\gamma \geq 1$.

This is where the second source comes in. We use the second source to sample $D' \ll D$ pseudorandom indices $T \subseteq [D]$ in a way that the fraction of bad bits in the projected string $\mathbf{Z}_T$ remains almost the same as in $\mathbf{Z}$, with high probability. This can be done using an extractor-based sampler [Zuc97]. Now we apply Alon-Goldreich-Mansour to conclude that the good bits of $\mathbf{Z}_T$ are δ-close to a t-wise independent distribution, where $\delta = (D')^t\gamma \ll 1$. Thus, the output of our 2-source extractor is the new resilient function applied to $\mathbf{Z}_T$.

1.4.2 A Monotone Resilient Function in AC^0 Suffices

The only part missing from the description above is how to construct the resilient function. First, we show that a monotone resilient function in AC^0 is resilient even if the good bits are just polylog-wise independent. The key ingredient is Braverman's result that polylog-wise independence fools AC^0 [Bra10].

To elaborate, let $f : \{0,1\}^n \to \{0,1\}$ be a monotone resilient function in AC^0 that is almost unbiased. To explain the key observation, let $Q \subseteq [n]$ be any set of input variables, let x_Q and $x_{\overline{Q}}$ denote the projections of $x \in \{0,1\}^n$ to Q and $\overline{Q}$, and write $x = x_Q \circ x_{\overline{Q}}$. We observe that there is another AC^0 circuit $\mathcal{E}$ that decides whether an input $x_{\overline{Q}}$ leaves f undetermined, i.e., whether there exist x_Q and y_Q such that $f(x_Q \circ x_{\overline{Q}}) \neq f(y_Q \circ x_{\overline{Q}})$. Specifically, since f is monotone, $\mathcal{E}$ simply compares $f(0^Q \circ x_{\overline{Q}})$ with $f(1^Q \circ x_{\overline{Q}})$. Now Braverman's result implies that the bias of the circuit $\mathcal{E}$ is roughly the same when $X_{\overline{Q}}$ is drawn from a polylog-wise independent distribution and when it is drawn from the uniform distribution. This implies the resiliency of f is almost the same in these two scenarios.

1.4.3 Constructing the Resilient Function

Thus all that remains is to construct a monotone AC^0 circuit f, that is almost balanced under the uniform distribution, and $\mathbf{I}_q(f) = o(1)$ for $q < D^{1-\delta}$. The high level idea for this construction is to derandomize the probabilistic construction of Ajtai-Linial [AL93] using extractors. The tribes function introduced by Ben-Or and Linial [BL85] is a disjunction taken over AND's of equi-sized blocks of variables. The Ajtai-Linial function is essentially a conjunction of non-monotone tribes functions, with each tribes function using a different partition and the variables in each tribes function being randomly negated with probability $1/2$. Ajtai and Linial choose the partitions using the probabilistic method.

We sketch informally our ideas to derandomize and monotonize this construction. For each $i \in [R]$, let P^i be an equi-partition of $[n]$, $n = MB$, into blocks of size B. Let P^i_j denote the j'th block in P^i. Define f as the conjunction of the corresponding monotone tribes:

$$f(x) = \bigwedge_{1 \le i \le R} \bigvee_{1 \le j \le M} \bigwedge_{\ell \in P^i_j} x_\ell.$$

First, we abstract out properties that these partitions need to satisfy for f to be almost unbiased and also $(n^{1-\delta}, \epsilon)$-resilient. Informally, we show that

1. If for all i_1, i_2, j_1, j_2 with $(i_1, j_1) \neq (i_2, j_2)$, $|P^{i_1}_{j_1} \cap P^{i_2}_{j_2}| \le 0.9B$, then f is almost unbiased,
2. If for any set Q of size $q < n^{1-\delta}$, the number of partitions P^i containing a block P^i_j such that $|P^i_j \cap Q| > \delta B/2$ is $o(R)$, then f is $(n^{1-\delta}, \epsilon)$-resilient.

An ingredient in the proof of (1) is Janson's inequality (see Theorem 5.13).

It is important that unlike in Ajtai-Linial and earlier modifications [RZ01], we don't need to negate variables, and thus f is monotone.

The second property seems related to the sampler property of extractors captured in Theorem 2.12. However, a sampler or extractor would just give us one subset, whereas we want to partition the space into subsets. Our main idea here is to use shifts of the subset to create a partition. Specifically, we construct a family of equi-partitions of $[n] = [BM]$, with each block of a partition being of size B, from a seeded extractor $\mathrm{Ext} : \{0,1\}^r \times \{0,1\}^b \to \{0,1\}^m$ as follows. Each P^w corresponds to some $w \in \{0,1\}^r$. One block of P^w is $P^w_{\vec{0}} = \{(y, \mathrm{Ext}(x,y)) : y \in \{0,1\}^b\}$. The other block are shifts of this, i.e., for any $s \in \{0,1\}^m$, define $P^w_s = \{(y, \mathrm{Ext}(x,y) \oplus s) : y \in \{0,1\}^b\}$. This gives $R = 2^r$ partitions of $[n]$ with $n = 2^{m+b}$.

For any good enough extractor, we show that (2) is satisfied using a basic property of extractors and an averaging argument. To show that the partitions satisfy (1), we need an additional property of the extractor, which informally requires us to prove that the intersection of any two arbitrary shifts of neighbors of any two distinct nodes $w_1, w_2 \in \{0,1\}^r$ in G_{Ext} is bounded. This essentially is a strong variant of a design extractor of Li [Li12a]. We show that Trevisan's extractor has this property. This completes the informal sketch of our resilient function construction. We note that our actual construction is slightly more complicated and is a depth 4 circuit. The extra layer enables us to simulate each of the bits $x_1, \ldots, x_n$ having $\Pr[x_1 = 1]$ close to 1, which we need to make f almost unbiased.

1.5 Organization

We use Section 2 for preliminaries. In Section 3, we use non-malleable extractors to reduce the problem of constructing 2-source extractors to the problem of constructing a resilient function. In Section 4 we show that if f is computable by a polynomial sized constant depth monotone circuit, then in order to prove an upper bound for $\mathbf{I}_{q,t}(f)$, it is in fact enough to upper bound $\mathbf{I}_q(f)$. In Section 5 we explicitly construct such a function f with small $\mathbf{I}_q(f)$ that is computable by a polynomial sized constant depth monotone circuit. We prove Theorem 4 in Section 6. Finally, we prove Theorem 1 in Section 7.

2 Preliminaries

We reserve the letter e for the base of the natural logarithm. We use $\ln(x)$ for $\log_e(x)$, and $\log(x)$ for $\log_2(x)$.
We use $\mathbf{U}_m$ to denote the uniform distribution on $\{0,1\}^m$.
For any integer $t > 0$, $[t]$ denotes the set $\{1, \ldots, t\}$.
For a string y of length n, and any subset $S \subseteq [n]$, we use y_S to denote the projection of y onto the coordinates indexed by S.
For any binary strings $x, y \in \{0,1\}^n$, we use $\Delta(x,y)$ to denote the Hamming distance.

2.1 An Inequality

We frequently use the following inequality.

Claim 2.1. *For any $n > 1$ and $0 \leq x \leq n$, we have*

$$e^{-x}\left(1 - \frac{x^2}{n}\right) \leq \left(1 - \frac{x}{n}\right)^n \leq e^{-x}.$$

2.2 Some Probability Lemmas

Definition 2.2 ((n,τ)-Bernoulli distribution)**.** *A distribution on n bits is an (n,τ)-Bernoulli distribution, denoted by $\mathbf{Ber}(n,\tau)$, if each bit is independently set to 1 with probability τ and set to 0 with probability $1-\tau$.*

Lemma 2.3 ([GRS06])**.** *Let $\mathbf{X}$ be a random variable taking values in a set S, and let $\mathbf{Y}$ be a random variable on $\{0,1\}^t$. Assume that $|(\mathbf{X},\mathbf{Y}) - (\mathbf{X},\mathbf{U}_t)| \leq \epsilon$. Then for every $y \in \{0,1\}^t$,*

$$|(\mathbf{X}|Y = y) - \mathbf{X}| \leq 2^{t+1}\epsilon.$$

Lemma 2.4 ([Sha08])**.** *Let $\mathbf{X}_1, \mathbf{Y}_1$ be random variables taking values in a set S_1, and let $\mathbf{X}_2, \mathbf{Y}_2$ be random variables taking values in a set S_2. Suppose that*

1. *$|\mathbf{X}_2 - \mathbf{Y}_2| \leq \epsilon_2$.*
2. *For every $s_2 \in S_2$, $|(\mathbf{X}_1|\mathbf{X}_2 = s_2) - (\mathbf{Y}_1|Y_2 = s_2)| \leq \epsilon_1$.*

Then

$$|(\mathbf{X}_1, \mathbf{X}_2) - (\mathbf{Y}_1, \mathbf{Y}_2)| \leq \epsilon_1 + \epsilon_2.$$

Using the above results, we record a useful lemma.

Lemma 2.5. *Let $\mathbf{X}_1, \ldots, \mathbf{X}_t$ be random variables, such that each $\mathbf{X}_i$ takes values 0 and 1. Further suppose that for any subset $S = \{s_1, \ldots, s_r\} \subseteq [t]$,*

$$(\mathbf{X}_{s_1}, \mathbf{X}_{s_2} \ldots, \mathbf{X}_{s_r}) \approx_\epsilon (\mathbf{U}_1, \mathbf{X}_{s_2} \ldots, \mathbf{X}_{s_r}).$$

Then

$$(\mathbf{X}_1, \ldots, \mathbf{X}_t) \approx_{5t\epsilon} \mathbf{U}_t.$$

Proof. We prove this by induction on t. The base case when $t = 1$ is direct. Thus, suppose $t \geq 2$. It follows that

$$(\mathbf{X}_t, \mathbf{X}_1, \ldots, \mathbf{X}_{t-1}) \approx_\epsilon (\mathbf{U}_1, \mathbf{X}_1, \ldots, \mathbf{X}_{t-1}).$$

By an application of Lemma 2.3, for any value of the bit b,

$$|(\mathbf{X}_1, \ldots, \mathbf{X}_{t-1} | \mathbf{X}_t = b) - (\mathbf{X}_1, \ldots, \mathbf{X}_{t-1})| \leq 4\epsilon.$$

Further, by the induction hypothesis, we have

$$|(\mathbf{X}_1, \ldots, \mathbf{X}_{t-1}) - \mathbf{U}_{t-1}| \leq 5(t-1)\epsilon.$$

Thus, by the triangle inequality for statistical distance, it follows that for any value of the bit b,

$$|(\mathbf{X}_1, \ldots, \mathbf{X}_{t-1} | \mathbf{X}_t = b) - \mathbf{U}_{t-1}| \leq (5t-1)\epsilon.$$

Using Lemma 2.4 and the fact that $|\mathbf{X}_t - \mathbf{U}_1| \leq \epsilon$, it follows that

$$|(\mathbf{X}_1, \ldots, \mathbf{X}_t) - \mathbf{U}_t| \leq (5t-1)\epsilon + \epsilon = 5t\epsilon.$$

This completes the induction, and the lemma follows. □

We record a fact about almost t-wise independent distributions.

Theorem 2.6 ([AGM03])**.** *Let $\mathcal{D}$ be a (t, γ)-wise independent distribution on $\{0,1\}^n$. Then there exists a t-wise independent distribution that is $n^t\gamma$-close to $\mathcal{D}$.*

2.3 Extractors for NOBF Sources via Resilient Functions

Definition 2.7 (NOBF Sources)**.** *A source $\mathbf{Z}$ on $\{0,1\}^D$ is called a (q, t, γ)-non-oblivious bit-fixing source (NOBF source for short) if there exists a subset of coordinates $Q \subseteq [D]$ of size at most q such that the joint distribution of the bits indexed by $\overline{Q} = [D] \setminus Q$ is (t, γ)-wise independent. The bits in the coordinates indexed by Q are allowed to depend arbitrarily on the bits in the coordinates indexed by $\overline{Q}$. If $\gamma = 0$, we just say it is a (q, t)-NOBF source.*

The following is a simple corollary of Theorem 2.6 which states that it is enough to reason about (q, t)-NOBF sources instead of (q, t, γ)-NOBF sources (on n bits) by paying an additional error of γn^t.

Corollary 2.8. *Let $\mathbf{X}$ be a (q, t, γ)-NOBF source on n bits. Then, there exists a (q, t)-NOBF source $\mathbf{Y}$ on n bits such that $|\mathbf{X} - \mathbf{Y}| \leq \gamma n^t$.*

We recall a simple connection between the problem of constructing extractors for (q,t,γ)-NOBF sources and constructing (t,γ)-independent (q,ϵ_1)-resilient functions.

Lemma 2.9. *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function that is (t,γ)-independent (q,ϵ_1)-resilient. Further suppose that for any (t,γ)-wise independent distribution $\mathcal{D}$, $|\mathbf{E}_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x})]-\frac{1}{2}| \leq \epsilon_2$. Then f is an extractor for (q,t,γ)-NOBF sources with error $\epsilon_1+\epsilon_2$.*

Proof. Let $\mathbf{X}$ be a (q,t,γ)-non-oblivious bit-fixing source on n bits. Then $\mathbf{X}$ is sampled in the following way: For some fixed subset $Q \subset \{x_1,\ldots,x_n\}$ of q variables, the variables $\overline{Q} = [n] \setminus Q$ are drawn from some fixed (t,γ)-wise independent distribution $\mathcal{D}_1$ on $n-q$ bits, and the variables in Q are chosen arbitrarily depending on the values of the variables in $\overline{Q}$.

Let E be the following event: f is determined on fixing the variables in $\overline{Q}$ by sampling from $\mathcal{D}_1$ and leaving the remaining variables free. Since f is (t,γ)-independent (q,ϵ_1)-resilient, we have $\mathbf{Pr}[E] \geq 1-\epsilon_1$. Let $\mathcal{D}$ be any (t,γ)-wise independent distribution on n bits whose projection on to $\overline{Q}$ matches $\mathcal{D}_1$. It follows that

$$\left|\mathbf{Pr}_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x})=1]-\frac{1}{2}\right| \leq \epsilon_2.$$

We have,

$$\begin{aligned}
\mathbf{Pr}_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x})=1] &= \mathbf{Pr}_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x})=1|E]\mathbf{Pr}[E] + \mathbf{Pr}_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x})=1|\overline{E}]\mathbf{Pr}[\overline{E}] \\
&= \mathbf{Pr}_{\mathbf{x}\sim\mathbf{X}}[f(\mathbf{x})=1|E]\mathbf{Pr}[E] + \mathbf{Pr}_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x})=1|\overline{E}]\mathbf{Pr}[\overline{E}] \\
&= \mathbf{Pr}_{\mathbf{x}\sim\mathbf{X}}[f(\mathbf{x})=1] + \mathbf{Pr}[\overline{E}]\left(\mathbf{Pr}_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x})=1|\overline{E}] - \mathbf{Pr}_{\mathbf{x}\sim\mathbf{X}}[f(\mathbf{x})=1|\overline{E}]\right)
\end{aligned}$$

Hence,

$$|\mathbf{Pr}_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x})=1] - \mathbf{Pr}_{\mathbf{x}\sim\mathbf{X}}[f(\mathbf{X})=1]| \leq \mathbf{Pr}[\overline{E}] \leq \epsilon_1.$$

Thus,

$$\left|\mathbf{Pr}_{\mathbf{x}\sim\mathbf{X}}[f(\mathbf{x})=1]-\frac{1}{2}\right| \leq \epsilon_1+\epsilon_2.$$

□

2.4 Seeded Extractors and Samplers

We use the following strong seeded extractor constructed by Trevisan [Tre01], with subsequent improvements by Raz, Reingold and Vadhan [RRV02].

Theorem 2.10 ([Tre01, RRV02])**.** *There exists a constant $\lambda > 0$ such that for every $n,k,m \in \mathbb{N}$ and $\epsilon > 0$, with $m \leq k \leq n$, there exists an explicit strong-seeded extractor* $\mathrm{TExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *for min-entropy k and error ϵ, where $d = \lambda \cdot \left(\frac{\log^2(n/\epsilon)}{\log(k/m)}\right)$.*

We also use optimal constructions of strong-seeded extractors.

Theorem 2.11 ([GUV09])**.** *For any constant $\alpha > 0$, and all integers $n,k > 0$ and any $\epsilon > 0$, there exists a polynomial time computable strong-seeded extractor* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with $d = O(\log n + \log(1/\epsilon))$ and $m = (1-\alpha)k$.*

To ensure that for each $x \in \{0,1\}^n$, $\mathrm{Ext}(x,s_1) \neq \mathrm{Ext}(x,s_2)$ whenever $s_1 \neq s_2$, one can concatenate the seed to the output of Ext, though it is no longer strong.

2.4.1 Sampling Using Weak Sources

Sampling is a fundamental task in computer science, and a long line of work has been dedicated to constructing randomness efficient samplers. We require samplers that work with access to weak sources of randomness. Sipser [Sip88] introduced the notion of dispersers and showed applications to randomness efficient sampling for one-sided error. We use a technique of sampling for two-sided error using randomness extractor.

We first introduce a graph-theoretic view of extractors. Any seeded extractor $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ can also be viewed as an unbalanced bipartite graph G_{Ext} with 2^n left vertices (each of degree 2^d) and 2^m right vertices. We use $\mathcal{N}(x)$ to denote the set of neighbours of x in G_{Ext}. We call G_{Ext} the graph corresponding to Ext.

Theorem 2.12 ([Zuc97])**.** *Let* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a seeded extractor for min-entropy* k *and error* ϵ*. Let* $D = 2^d$*. Then for any set* $R \subseteq \{0,1\}^m$*,*

$$|\{x \in \{0,1\}^n : ||\mathcal{N}(x) \cap R| - \mu_R D| > \epsilon D\}| < 2^{k+1},$$

where $\mu_R = |R|/2^m$*.*

2.4.2 Shift-Design Extractors

We introduce the notion of a *shift-design extractor*. This generalizes the notion of design extractors introduced by Li [Li12a]. We first informally discuss the notion of design extractors and our generalization to shift-design extractors. Given a strong-seeded extractor $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, define the extractor Ext' that concatenates the seed to the output of Ext (i.e., such that $\text{Ext}'(x,y) = \text{Ext}(x,y) \circ y$). It is now useful to think in terms of the extractor graph $G_{Ext'}$ (see Section 2.4.1 where this view is introduced). Ext' is a design extractor if the collection of 2^n sets, each set corresponding to the set of neighbors $\mathcal{N}(x)$ of a vertex $x \in \{0,1\}^n$ on the left in G_{Ext}, form a design (i.e., the pairwise intersection of any two sets is bounded). We extend this to a more robust notion, and require that the design property holds even under arbitrary 'shifts' of the sets. We now formally define shift-design extractors.

Definition 2.13 (Shift-design extractor)**.** *Let* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a strong-seeded extractor. Let* $D = 2^d$*. If for any distinct* $y, y' \in \{0,1\}^n$*, and arbitrary* $h, h' \in \{0,1\}^m$

$$|\{(z, \text{Ext}(y,z) \oplus h) : z \in \{0,1\}^d\} \cap \{(z, \text{Ext}(y',z) \oplus h') : z \in \{0,1\}^d\}| \leq (1-\eta)D,$$

then Ext *is called an* η*-shift-design extractor.*

We prove that Trevisan's extractor [Tre01] (see Theorem 2.10) is a shift-design extractor. We first describe the construction of Trevisan's extractor. Our proof in fact requires us to reason about the way a single bit (say, the first bit) of the output is produced, and hence we present a simplified view of the construction. We refer the reader to Trevisan's paper [Tre01] for more details on the construction.

A one-bit version of Trevisan's extractor: Let r, b be integers, where we set the parameter b later. Let $B = 2^b$. On inputs $y \in \{0,1\}^r$ and $z \in \{0,1\}^b$, we describe the construction of the one-bit version of Trevisan's extractor $\text{TExt} : \{0,1\}^r \times \{0,1\}^b \to \{0,1\}$.

- Fix an asymptotically good binary linear error correcting code $\mathcal{C}$ with constant relative rate α, block length $\bar{r} = r/\alpha$ and relative minimum distance $\frac{1}{2} - \beta$ with $\beta < 1/10$ such that

$\mathcal{C}' = \text{span}\{\mathcal{C}, \vec{1}\}$ is also a code with distance $1/2 - \beta$ (where $\vec{1}$ denotes the all 1 string). [5] Let $\text{Enc} : \{0,1\}^r \to \{0,1\}^{\overline{r}}$ be the encoding function of $\mathcal{C}$.

- Let $b = \log(\overline{r})$, assuming without loss of generality that $\overline{r}$ is a power of 2.
- The output of TExt is the bit at the z'th coordinate (interpreting the string z as an integer in $[B]$ the natural way) of the string $c_y = \text{Enc}(y)$.

Remark 2.14. *We use the following fact about the multi-bit version of Trevisan's extractor (i.e.,* $\text{TExt} : \{0,1\}^r \times \{0,1\}^b \to \{0,1\}^m$, $m > 1$*): let* $\text{TExt}_1 : \{0,1\}^r \times \{0,1\}^b \to \{0,1\}$ *be the function that just outputs the first output bit of* TExt*. Then* TExt_1 *is exactly the same function as the one-bit Trevisan extractor described above.*

Lemma 2.15. *Let* $\text{TExt} : \{0,1\}^r \times \{0,1\}^b \to \{0,1\}^m$ *be Trevisan extractor from Theorem 2.10. Then,* TExt *is a* $\frac{1}{10}$*-shift-design extractor.*

Proof. We first prove the lemma for the case $m = 1$. It is then straightforward to extend this to $m > 1$. Let $y, y' \in \{0,1\}^r$, $y \neq y'$ and $h, h' \in \{0,1\}$. Let $c_y = \text{Enc}(y)$ and $c_{y'} = \text{Enc}(y')$.

Consider the case $h = h'$. From the above description of TExt (for the case $m = 1$), it follows that $|\{z \in \{0,1\}^b : (z, \text{TExt}(y,z)) \neq (z, \text{TExt}(y',z))\}| = \Delta(c_y, c_{y'}) \geq (\frac{1}{2} - \beta)B > B/10$, using the fact that c_y and $c_{y'}$ are distinct codewords in $\mathcal{C}$.

Now suppose $h \neq h'$. It follows that $\{z \in \{0,1\}^b : (z, \text{TExt}(y,z)) \neq (z, \text{TExt}(y',z))\} = \Delta(c_y, c_{y'} \oplus \vec{1}) \geq (\frac{1}{2} - \beta)B > B/10$, using the fact that c_y and $c_{y''} = c_{y'} \oplus \vec{1}$ are distinct codewords in $\mathcal{C}'$. The fact that $c_y \neq c_{y''}$ can be seen as follows: c_y and $c_{y'}$ are codewords in $\mathcal{C}$ and hence $c_y \oplus c_{y'}$ is a codeword in $\mathcal{C}$ (since it is a linear code). Since $\vec{1}$ is not a codeword in $\mathcal{C}$, it follows that $c_y \oplus c_{y'} \neq \vec{1}$. This completes the proof for the case $m = 1$.

This extends to the case $m > 1$ almost immediately in the following way. Let $y, y' \in \{0,1\}^r$, $y \neq y'$, and $h, h' \in \{0,1\}^m$. Let a and a' be the first bits of h and h' respectively. Further let TExt_1 be the function that outputs the first bit of TExt (i.e, $\text{TExt}_1(y,z) = \text{TExt}(y,z)_{[1]}$). It follows that

$$|\{(z, \text{TExt}(y,z) \oplus h) : z \in \{0,1\}^b\} \cap \{(z, \text{TExt}(y',z) \oplus h') : z \in \{0,1\}^b\}| \leq$$
$$|\{(z, \text{TExt}_1(y,z) \oplus a) : z \in \{0,1\}^b\} \cap \{(z, \text{TExt}_1(y',z) \oplus a') : z \in \{0,1\}^b\}| \leq 9B/10,$$

where the final inequality, by Remark 2.14, follows from the $m = 1$ case (for which we have proved the lemma). This completes the proof.

□

2.5 Non-Malleable Extractors

Non-malleable extractors were introduced by Dodis and Wichs [DW09] as a generalization of strong-seeded extractors. We define t-non-malleable extractors, which generalize the notion introduced in [DW09] (which corresponds to the case $t = 2$). The work of Cohen, Raz and Segev [CRS14] was the first to introduce the the notion of t-non-malleable extractors.

[5] In other words, one can start with any good linear code $\mathcal{C}'$ with block length $\overline{r}$ that has minimum distance $\frac{1}{2} - \beta$ and contains $\vec{1}$. Let $\{v_1, \ldots, v_{r+1}\}$ be a basis of $\mathcal{C}'$ with $v_{r+1} = \vec{1}$. Now $\mathcal{C}$ is defined to be the binary linear code generated by $\{v_1, \ldots, v_r\}$, i.e., $\mathcal{C} = \text{span}\{v_1, \ldots, v_r\}$.

Definition 2.16. *A function* $\text{nmExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (t,k,ϵ)*-non-malleable extractor if it satisfies the following property: If* $\mathbf{X}$ *is a* (n,k)*-source and* $\mathbf{Y}$ *is uniform on* $\{0,1\}^d$*, and* $f_1,\ldots,f_t$ *are arbitrary functions from* d *bits to* d *bits with no fixed points*[6]*, then*

$$\begin{aligned}&(\text{nmExt}(\mathbf{X},\mathbf{Y}),\text{nmExt}(\mathbf{X},f_1(\mathbf{Y})),\ldots,\text{nmExt}(\mathbf{X},f_t(\mathbf{Y})),\mathbf{Y})\\ &\approx_\epsilon (\mathbf{U}_m,\text{nmExt}(\mathbf{X},f_1(\mathbf{Y})),\ldots,\text{nmExt}(\mathbf{X},f_t(\mathbf{Y})),\mathbf{Y}).\end{aligned}$$

We prove a lemma that provides a useful alternate view of t-non-malleable extractors.

Lemma 2.17. *Let* $\text{nmExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ *be a* (t,k,ϵ)*-non-malleable extractor. Let* $\{0,1\}^d = \{s_1,\ldots,s_D\}$*,* $D = 2^d$*. Let* $\mathbf{X}$ *be any* (n,k)*-source. There exists a subset* $R \subseteq \{0,1\}^d$*,* $|R| \geq (1-\sqrt{\epsilon})D$ *such that for any distinct* $r_1,\ldots,r_t \in R$*,*

$$(\text{nmExt}(\mathbf{X},r_1),\ldots,\text{nmExt}(\mathbf{X},r_t)) \approx_{5t\sqrt{\epsilon}} \mathbf{U}_t.$$

Proof. Let

$$\begin{aligned}BAD = \{r \in \{0,1\}^d : \exists \text{ distinct } r_1,\ldots,r_t \in \{0,1\}^d,\\ \forall i \in [t]\ r_i \neq r,\ \text{s.t } |(\text{nmExt}(\mathbf{X},r),\text{nmExt}(\mathbf{X},r_1),\ldots,\text{nmExt}(\mathbf{X},r_t))-\\ (\mathbf{U}_1,\text{nmExt}(\mathbf{X},r_1),\ldots,\text{nmExt}(\mathbf{X},r_t))| > \sqrt{\epsilon}\}\end{aligned}$$

We define adversarial functions $f_1,\ldots,f_t$ as follows. For each $r \in BAD$, set $f_i(r) = r_i$, $i = 1,\ldots,t$ (the f_i's are defined arbitrarily for $r \notin BAD$, only ensuring that there are no fixed points). Let $\mathbf{Y}$ be uniform on $\{0,1\}^d$. It follows that

$$\begin{aligned}&|(\text{nmExt}(\mathbf{X},\mathbf{Y}),\text{nmExt}(\mathbf{X},f_1(\mathbf{Y})),\ldots,\text{nmExt}(\mathbf{X},f_t(\mathbf{Y})),\mathbf{Y})-\\ &(\mathbf{U}_1,\text{nmExt}(\mathbf{X},f_1(\mathbf{Y})),\ldots,\text{nmExt}(\mathbf{X},f_t(\mathbf{Y})),\mathbf{Y})| \geq \frac{\sqrt{\epsilon}}{2^d}|BAD|\end{aligned}$$

Thus $|BAD| \leq \sqrt{\epsilon}2^d$ using the property that nmExt is a (k,t,ϵ)-non-malleable extractor. Define $R = \{0,1\}^d \setminus BAD$. Using Lemma 2.5, it follows that R satisfies the required property. □

Remark 2.18. *In fact, the above proof gives us something stronger. It shows that for any seed* $s \in R$*, and any other* t *seeds* $s_1,\ldots,s_t$ *(not necessarily in* R*), we have*

$$|(\text{nmExt}(\mathbf{X},s),\text{nmExt}(\mathbf{X},s_1),\ldots,\text{nmExt}(\mathbf{X},s_t)) - (\mathbf{U}_1,\text{nmExt}(\mathbf{X},s_1),\ldots,\text{nmExt}(\mathbf{X},s_t))| \leq \sqrt{\epsilon}.$$

However, we do not use this stronger property in our analysis.

The first construction of explicit t-non-malleable extractor for polylogarithmic min-entropy (in fact, for any min-entropy significantly smaller than $n/2$) was given by Chattopadhyay, Goyal and Li [CGL16]. Subsequently, a long line of work improved on their methods and we use the state-of-the-art non-malleable extractor from the work of Li [Li18].

Theorem 2.19 ([Li18])**.** *There exists a constant* $c' > 0$ *such that for all* $n,t > 0$ *there exists an explicit* (t,k,ϵ)*-non-malleable extractor* $\text{nmExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$*, where* $k \geq c't\left(\log\log n + \frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)}\right)$ *and* $d = O\left(t^2 \cdot \left(\log\log n + \frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)}\right)\right)$.

[6] We say that x is a fixed point of a function f if $f(x) = x$.

2.6 Boolean Circuits, the Class AC^0 and Other Definitions

For the sake of being self-contained, we define Boolean circuits and related notions that we use in this paper. We refer the reader to the book by Arora and Barak [AB09] for a more comprehensive introduction to circuit complexity.

Definition 2.20. *For an integer $n > 0$, a Boolean circuit C with n inputs is a directed acyclic graph with n nodes having in-degree 0 called the input nodes, and one node, called the output node, having out-degree 0 and in-degree 1. The other nodes are called gates, and are labeled with one of $\wedge$ (logical AND), $\vee$ (logical OR), or $\neg$ (logical NOT).*

- *The fan-in of C is defined to be the maximum in-degree of a node in the graph corresponding to C.*
- *The size of C is defined to be the sum of the number of nodes and edges in the graph corresponding to C.*
- *The gate with an out-edge to the output node is called the root node.*
- *The depth of a gate is the length of the longest directed path from the gate to the root node.*
- *The length of the longest directed path from an input node to the root node is defined to be the depth of the circuit C.*
- *For the sake of convenience, given an input $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ to the circuit C, we allow $\{\neg x_1, \ldots, \neg x_n\}$ to be additional input variables available to C. (We still say that C is a circuit defined on n input bits.)*

Boolean functions computed by circuits A Boolean circuit C with n inputs naturally computes an output bit given an input $x \in \{0,1\}^n$. We use $C(x)$ to denote this output bit. We say that a function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a circuit of size s and depth d if there exists a Boolean circuit C on n inputs with size at most s and depth at most d such that $f(x) = C(x)$ for all $x \in \{0,1\}^n$.

Families of circuits A circuit family $\{C_n\}_{n\in\mathbb{N}}$ of size $s(n)$ and depth $d(n)$ is a sequence of circuits with C_n being a circuit with n inputs, size at most $s(n)$ and depth at most $d(n)$. We say that a language $L = \cup_{n\geq 0} L_i$, $L_n \subseteq \{0,1\}^n$ is recognized by a circuit family of size $s(n)$ and depth $c(n)$ if there exists a circuit family $\{C_n\}_{n\in\mathbb{N}}$ of size $s(n)$ and depth $d(n)$ such that for any integer $i > 0$ and $y \in \{0,1\}^i$, $y \in L_i$ iff $C_i(y) = 1$.

Definition 2.21. *The class AC^0 consists of all languages recognized by some circuit family $\{C_n\}_{n\in\mathbb{N}}$ with depth $d(n) = O(1)$, size $s(n) = \mathrm{poly}(n)$, and unbounded fan-in. By a slight abuse of notation, we say that the circuit family circuit $\{C_n\}_{n\in\mathbb{N}}$ is in AC^0.*

Definition 2.22. *A conjunctive normal form (abbrv. CNF) is a depth 2 circuit with an $\wedge$ gate at the root node and $\vee$ gates at depth 1. A disjunctive normal form (abbrv. DNF) is a depth 2 circuit with an $\vee$ gate at the root node and $\wedge$ gates at depth 1.*

Definition 2.23. *A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is called monotone if for any $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$ satisfying $\forall i \in [n]$, $x_i \leq y_i$, we have $f(x) \leq f(y)$.*

2.7 A Simple Lemma

We use the following lemma which shows that a small CNF can simulate a biased bit with high enough accuracy.

Lemma 2.24. *Suppose $\gamma < 9/10$. Then for any $\nu > 0$, there exists an explicit size h monotone* CNF $\mathcal{C}$ *on h bits, where $h = O\left(\frac{1}{\nu}\ln\left(\frac{1}{\nu}\right)\right)$, such that $\gamma - \nu \leq \mathbf{Pr}_{\mathbf{x}\sim\mathbf{U}_h}[\mathcal{C}(\mathbf{x}) = 0] < \gamma$.*

Proof. Let $h_2 = \lceil \log(2/\nu) \rceil$, and let h_1 be the largest integer such that $(1-2^{-h_2})^{h_1} \geq 1-\gamma$. Thus,

$$\begin{aligned}(1-\gamma) \leq (1-2^{-h_2})^{h_1} &\leq (1-\gamma)/(1-2^{-h_2}) \\ &< (1-\gamma)(1+2^{1-h_2}) \\ &\leq (1-\gamma)(1+\nu) \\ &< 1-\gamma+\nu\end{aligned}$$

and $h_1 = O(2^{h_2})$.

Define

$$\mathcal{C}(x) = \bigwedge_{g_1=1}^{h_1} \bigvee_{g_2=1}^{h_2} x_{g_1,g_2}.$$

and $h = h_1 h_2 = O(h_2 2^{h_2}) = O\left(\frac{1}{\nu}\log\left(\frac{1}{\nu}\right)\right)$.

Thus $\mathbf{Pr}_{\mathbf{x}\sim\mathbf{U}_h}[\mathcal{C}(\mathbf{x}) = 0] = 1 - (1-2^{-h_2})^{h_1}$, and hence

$$\gamma - \nu \leq \mathbf{Pr}_{\mathbf{x}\sim\mathbf{U}_h}[\mathcal{C}(\mathbf{x}) = 0] \leq \gamma.$$

□

3 Reduction to an NOBF Source

The main result in this section is a reduction from the problem of extracting from two independent (n,k)-sources to the task of extracting from a single (q,t)-NOBF source. We formally state the reduction in the following theorem.

Theorem 3.1. *Let* $\mathrm{nmExt} : \{0,1\}^n \times \{0,1\}^{d_1} \to \{0,1\}$ *be a (t,k,ϵ_1)-non-malleable extractor and let* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^{d_2} \to \{0,1\}^{d_1}$ *be a seeded extractor for min-entropy $k/2$ with error ϵ_2. Let $\{0,1\}^{d_2} = \{s_1, \ldots, s_{D_2}\}$, $D_2 = 2^{d_2}$. Suppose that* Ext *satisfies the property that for all $y \in \{0,1\}^n$, $\mathrm{Ext}(y,s) \neq \mathrm{Ext}(y,s')$ whenever $s \neq s'$. Define the function:*

$$\mathrm{reduce}(x,y) = \mathrm{nmExt}(x, \mathrm{Ext}(y,s_1)) \circ \ldots \circ \mathrm{nmExt}(x, \mathrm{Ext}(y,s_{D_2})).$$

If $\mathbf{X}$ *and* $\mathbf{Y}$ *are independent (n,k)-sources, then*

$$\mathbf{Pr}_{\mathbf{y}\sim\mathbf{Y}}[\mathrm{reduce}(\mathbf{X},\mathbf{y}) \text{ is } O(t\sqrt{\epsilon_1}D_2^t)\text{-close to a } (q,t)\text{-NOBF source}] \geq 1 - 2\cdot 2^{-k/2},$$

where $q = (\sqrt{\epsilon_1} + \epsilon_2)D_2$.

Proof. Let $R \subseteq \{0,1\}^{d_1}$ be such that for any distinct $r_1, \ldots, r_t \in R$,

$$(\mathrm{nmExt}(\mathbf{X}, r_1), \ldots, \mathrm{nmExt}(\mathbf{X}, r_t)) \approx_{5t\sqrt{\epsilon_1}} \mathbf{U}_t.$$

It follows by Lemma 2.17 that $|R| \geq (1-\sqrt{\epsilon_1})D_1$. Define $\mathrm{Samp}(y) = \{\mathrm{Ext}(y,s_1),\ldots,\mathrm{Ext}(y,s_{D_2})\} \subset \{0,1\}^{d_1}$. Using Theorem 2.12, we have

$$\Pr_{\mathbf{y}\sim\mathbf{Y}}[|\mathrm{Samp}(\mathbf{y})\cap R| \leq (1-\sqrt{\epsilon_1}-\epsilon_2)D_2] \leq 2\cdot 2^{-k/2}. \tag{1}$$

Consider any $\mathbf{y}$ such that $|\mathrm{Samp}(\mathbf{y})\cap R| \geq (1-\sqrt{\epsilon_1}-\epsilon_2)D_2$, and let $\mathbf{Z_y} = \mathrm{reduce}(\mathbf{X},\mathbf{y})$. Since the output bits of nmExt corresponding to seeds in $\mathrm{Samp}(\mathbf{y})\cap R$ are $(t, 5t\sqrt{\varepsilon_1})$-wise independent, we have that $\mathbf{Z_y}$ is a $((\sqrt{\epsilon_1}+\epsilon_2)D_2, t, 5t\sqrt{\varepsilon_1})$-NOBF source on D_2 bits.

Thus using (1), it follows that with probability at least $1-2\cdot 2^{-k/2}$ over $\mathbf{y}\sim\mathbf{Y}$, $\mathrm{reduce}(\mathbf{X},\mathbf{y})$ is a $((\sqrt{\epsilon_1}+\epsilon_2)D_2, t, 5t\sqrt{\epsilon_1})$-NOBF source on D_2 bits. The lemma now follows from Corollary 2.8. □

4 Monotone Constant-Depth Resilient Functions are t-Independent Resilient

Using the reduction from Section 3, we have now reduced the problem of extracting from two independent sources to extracting from a (q,t)-NOBF source. By Lemma 2.9, this translates to constructing a nearly balanced function f with small $\mathbf{I}_{q,t}(f)$.

We show that if f is computable by a polynomial sized constant depth monotone circuit, then in order to prove an upper bound for $\mathbf{I}_{q,t}(f)$, it is in fact enough to upper bound $\mathbf{I}_q(f)$, which is a simpler quantity to handle.

Theorem 4.1. *There exists a constant $b>0$ such that the following holds: Let $\mathcal{C}:\{0,1\}^n\to\{0,1\}$ be a monotone circuit in AC^0 of depth d and size m such that $|\mathbf{E}_{\mathbf{x}\sim\mathbf{U}_n}[\mathcal{C}(x)]-\frac{1}{2}| \leq \epsilon_1$. Suppose $q>0$ is such that $\mathbf{I}_q(\mathcal{C}) \leq \epsilon_2$. If $t \geq b(\log(m/\epsilon_3))^{3d+6}$, then $\mathbf{I}_{q,t}(\mathcal{C}) \leq \epsilon_2+\epsilon_3$ and $\mathbf{I}_{q,t,\gamma}(\mathcal{C}) \leq \epsilon_2+\epsilon_3+\gamma n^t$. Further, for any distribution $\mathcal{D}$ that is (t,γ)-wise independent, $|\mathbf{E}_{x\sim\mathcal{D}}[\mathcal{C}(x)]-\frac{1}{2}| \leq \epsilon_1+\epsilon_3+\gamma n^t$.*

We briefly sketched the main ideas of the proof of the above theorem in the introduction. We proceed to formally prove Theorem 4.1.

A crucial ingredient in our proof is the seminal result of Braverman [Bra10] that polylogarithmic independence 'fools' constant depth circuits. We state the result using refined bounds proved by Tal [Tal17].

Theorem 4.2 ([Bra10, Tal17])**.** *Let $\mathcal{D}$ be any $g(m,d,\epsilon)$-wise independent distribution on $\{0,1\}^n$. Then for any circuit $\mathcal{C}\in AC^0$ of depth d and size m,*

$$|\mathbf{E}_{\mathbf{x}\sim\mathbf{U}_n}[\mathcal{C}(\mathbf{x})] - \mathbf{E}_{\mathbf{x}\sim\mathcal{D}}[\mathcal{C}(\mathbf{x})]| \leq \epsilon$$

where $g(m,d,\epsilon) = O(\log(m/\epsilon))^{3d+3}$.

Proof of Theorem 4.1. The bound on $\mathbf{E}_{\mathbf{x}\sim\mathcal{D}}[\mathcal{C}(\mathbf{x})]$ is direct from Theorem 4.2 and Theorem 2.6. We now proceed to prove the influence property.

Consider any set Q of variables, $|Q|=q$. Let $\overline{Q} = [n]\setminus Q$. We construct a function $\mathcal{E}_Q:\{0,1\}^{n-q}\to\{0,1\}$ such that $\mathcal{E}_Q(y)=1$ if and only if $\mathcal{C}$ is undetermined when $x_{\overline{Q}}$ is set to y. Thus, it follows that

$$\mathbf{E}_{\mathbf{y}\sim\mathbf{U}_{n-q}}[\mathcal{E}_Q(\mathbf{y})] = \mathbf{Pr}_{\mathbf{y}\sim\mathbf{U}_{n-q}}[\mathcal{E}_Q(\mathbf{y})=1] = \mathbf{I}_Q(\mathcal{C}) \leq \epsilon_2.$$

Let $\mathcal{D}$ be any t-wise independent distribution. We have,

$$\mathbf{E}_{\mathbf{y}\sim\mathcal{D}}[\mathcal{E}_Q(\mathbf{y})] = \mathbf{Pr}_{\mathbf{y}\sim\mathcal{D}}[\mathcal{E}_Q(\mathbf{y})=1] = \mathbf{I}_{Q,\mathcal{D}}(\mathcal{C}).$$

Thus to prove that $\mathbf{I}_{Q,\mathcal{D}}(\mathcal{C}) \leq \epsilon_2 + \epsilon_3$, it is enough to prove that

$$|\mathbf{E}_{\mathbf{y}\sim\mathbf{U}_{n-q}}[\mathcal{E}_Q(\mathbf{y})] - \mathbf{E}_{\mathbf{y}\sim\mathcal{D}}[\mathcal{E}_Q(\mathbf{y})]| \leq \epsilon_3. \tag{2}$$

We construct $\mathcal{E}_Q$ as follows: Let $\mathcal{C}_0$ be the circuit obtained from $\mathcal{C}$ by setting all variables in Q to 0. Let $\mathcal{C}_1$ be the circuit obtained from $\mathcal{C}$ by setting all variables in Q to 1. Define $\mathcal{E}_Q := \neg(\mathcal{C}_0 = C_1)$. Since $\mathcal{C}$ is monotone, $\mathcal{E}_Q$ satisfies the required property. Further $\mathcal{E}_Q$ can be computed by a circuit in AC^0 of depth $d+2$ and size $4m+3$. It can be checked that the depth of $\mathcal{E}_Q$ can be reduced to $d+1$ by combining two layers. Thus (2) now directly follows by applying Theorem 4.2 on the depth-$(d+1)$ AC^0 circuit $\mathcal{E}_Q$ noting that $t \geq g(m, d+1, \epsilon_3)$, where g is the function defined in Theorem 4.2. □

5 A Monotone Resilient Function in AC^0

The main result in this section is an explicit construction of a function f which is resilient to coalitions, computable by a polynomial sized constant depth monotone circuit, and is almost balanced under the uniform distribution.

Theorem 3 (restated) For any constant $\delta \in (0,1)$, and every large enough integer n, there exists a polynomial time computable monotone Boolean function $f : \{0,1\}^n \to \{0,1\}$ satisfying:

- f is a depth 4 circuit in AC^0 of size $n^{O(1)}$.
- $\left|\mathbf{E}_{\mathbf{x}\sim\mathbf{U}_n}[f(\mathbf{x})] - \frac{1}{2}\right| \leq \frac{1}{n^{\Omega(1)}}$.
- For any $q > 0$, $\mathbf{I}_q(f) \leq q/n^{1-\delta}$.

We initially construct a depth 3 circuit which works, but then the inputs have to be chosen from independent Bernoulli distributions where the probability p of 1 is very different from $1/2$. By observing that we can approximate this Bernoulli distribution with a CNF on uniform bits, we obtain a depth 4 circuit which works for uniformly random inputs, and thus Theorem 3 follows.

We use Section 5.1 to describe the construction of the resilient function that works when the bits are biased. We use Section 5.2 to set up various parameters and state required relations that these parameters need to satisfy for our construction to hold. We state the main lemmas in Section 5.3. We use Sections 5.4, 5.5, 5.6 and 5.7 to prove these lemmas. In Section 5.8, we describe Construction 2 which satisfies Theorem 3. We use Appendix A to provide supplementary material for the proofs and arguments done in Section 5. In particular, Appendix A provides proofs of various bounds that are important for our argument to work but involve messy and routine calculations.

5.1 Our Construction

Our starting point is the work of Ajtai and Linial [AL93], who proved the existence of functions computable by linear sized depth 3 circuits in AC^0 that are $(\Omega(n/\log^2 n), \epsilon)$-resilient. However, this construction is probabilistic, and deterministically finding such functions requires time $n^{O(n^2)}$. Further these functions are not guaranteed to be monotone (or even unate[7]).

[7] A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is unate in a variable x_i if there exists $b \in \{0,1\}$ such that for all $x \in \{0,1\}^n$, $f(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n) \geq f(x_1, \ldots, x_{i-1}, 1-b, x_{i+1}, \ldots, x_n)$. f is unate if it is unate in each of the input variables.

We provide intuition of our construction in the introduction. We now present our construction. We carefully set parameters in Section 5.2. In this section, assume that r, b, m, k, ϵ are parameters that are fixed later. Let $R = 2^r, B = 2^b, M = 2^m$ and $s = MB$.

Construction 1: Let $\text{Ext} : \{0,1\}^r \times \{0,1\}^b \to \{0,1\}^m$ be a $\frac{1}{10}$-shift-design extractor set to extract from min-entropy k with error ϵ.
Let $\{0,1\}^r = \{v_1, \ldots, v_R\}$. We define a collection of R equi-partitions of $[s]$, $\mathcal{P} = \{P^{v_1}, \ldots, P^{v_R}\}$ as follows:

- Let G_{Ext} be the bipartite graph corresponding to Ext and let $\mathcal{N}(x)$, for any $x \in \{0,1\}^r$, denote the neighbours of x in G_{Ext}.
- For some $v \in \{0,1\}^r$, let $\mathcal{N}(v) = \{z_1, \ldots, z_B\}$. Define the partition P^v with blocks P^v_w for each $w \in \{0,1\}^m$ where

$$P^v_w = \{(j, z_j \oplus w) : j \in \{0,1\}^b\} \tag{3}$$

  ($\oplus$ denotes the bit-wise XOR of the two strings).

Claim 5.1 shows that P^v forms an equi-partition of $[s]$ with M blocks, each of size B.
Define the function $f_{\text{Ext}} : \{0,1\}^s \to \{0,1\}$ as:

$$f_{\text{Ext}}(y) = \bigwedge_{1 \le i \le R} \bigvee_{1 \le j \le M} \bigwedge_{\ell \in P^i_j} y_\ell.$$

Instantiation of Ext **in Construction 1:** We set $\text{Ext} : \{0,1\}^r \times \{0,1\}^b \to \{0,1\}^m$ to be the Trevisan extractor from Theorem 2.10 set to extract from min-entropy k with error ϵ.

We record a simple claim which shows that each P^v defined in Construction 1 is an equi-partition of $[s] = [BM]$ into blocks of size B.

Claim 5.1. *For any $v \in \{0,1\}^r$, P^v (described in Construction 1) is an equi-partition of $[s]$ into blocks of size B.*

Proof. We think of $[s]$ as the product set $[B] \times [M]$ and associate $[B]$ with $\{0,1\}^b$ and $[M]$ with $\{0,1\}^m$ in the natural way. We now prove that P^v is an equi-partition of $\{0,1\}^b \times \{0,1\}^m$. Recall that for each $w \in \{0,1\}^m$,

$$P^v_w = \{(j, z_j \oplus w) : j \in \{0,1\}^b\}$$

is a block of P^v (where $\mathcal{N}(v) = \{z_1, \ldots, z_B\}$ as defined in Construction 1). Clearly, for $i, j \in \{0,1\}^b$, $i \neq j$, and any $w, w' \in \{0,1\}^m$, $(i, z_i \oplus w) \neq (j, z_j \oplus w')$. This gives us that each P^v_w is of size exactly B.

Now suppose $|P^v_w \cap P^v_{w'}| > 0$ for distinct $w, w' \in \{0,1\}^b$. Then it must be the case that there is a $j \in \{0,1\}^b$ such that $(j, z_j \oplus w) = (j, z_j \oplus w')$ which is clearly a contradiction since $w \neq w'$. Thus, P^v_w and $P^v_{w'}$ are disjoint sets for distinct $w, w' \in \{0,1\}^m$. This completes the proof that P^v is an equi-partition of $[s]$. □

5.2 Various Parameters and their Relations

The construction in Section 5.1 involves many parameters that need to be set with care. We use this subsection to introduce parameters and present the way they are set up. Further, we list the key inequalities that they need to satisfy.

We begin with the simple observation that it is enough to prove Theorem 3 assuming $\delta \in (0, 1/10)$. This is straightforward since any f satisfying Theorem 3 for $\delta \in (0, 1/10)$ also satisfies the theorem for $\delta \in (0, 1)$. Thus, we assume $0 < \delta < 1/10$ for the rest of Section 5.

In the remaining parts of Sections 5, we assume that these parameters are set up in the way specified here. In Appendix A.0.1, we show that Construction 1 described in Section 5.1 can indeed be instantiated with parameters as specified here.

We now proceed to set up the various parameters:

- Let constants $\delta \in (0, 1/10)$ and $\epsilon_1 \in (0, 1/10)$, and any integer $r > 0$ be given as input parameters.
- Set $k = 2\delta r$, $m = k/2$.
- Let λ be the constant from Theorem 2.10.
- Let $\delta_2 > 0$ be a new parameter that we pick as follows. Define $\epsilon = 2^{-\delta_2\sqrt{r}}$ and $b = \frac{\lambda \log^2(r/\epsilon)}{\log(k/m)}$. Pick any δ_2 such that

$$\delta^2 r/40 \leq b = \lambda(\delta_2^2 r + \log^2 r + 2\delta_2\sqrt{r}\log r) \leq \delta^2 r/20 \tag{4}$$

- Define $\delta_1 = b/m$.
- Let $R = 2^r$, $B = 2^b$, $M = 2^m$ and $K = 2^k$.
- Let $s = BM$. Since $B = M^{\delta_1}$, thus $s = M^{1+\delta_1}$.
- Define $\gamma = \frac{\ln M - \ln\ln(R/\ln 2)}{B}$.
- Pick any p_1 that satisfies

$$(1 - B^{-\epsilon_1})\gamma \leq p_1 \leq \gamma \tag{5}$$

- Let $p_2 = (1 - p_1)^B$, $p_3 = (1 - p_2)^M$.
- For convenience, define $\alpha = p_3 R$.

The following are the inequalities that the above parameters need to satisfy for our construction to work:

$$\delta/40 \leq \delta_1 \leq \delta/20 \tag{6}$$

$$s^{1-\delta}/M < \epsilon < \delta/4 \tag{7}$$

5.3 Key Lemmas

We now state our key lemmas.

Lemma 5.2. *For any constants $0 < \delta, \epsilon_1 < 1/10$, and any integer $r > 0$, let $(1 - B^{-\epsilon_1})\gamma \leq p_1 \leq \gamma$. Then for any $q > 0$,*

$$\mathbf{I}_{q,\mathbf{Ber}(s,1-p_1)}(f_{\mathrm{Ext}}) \leq \frac{q}{s^{1-\delta}}.$$

Lemma 5.3. *For any constants $0 < \delta, \epsilon_1 < 1/10$, and any integer $r > 0$, let $(1 - B^{-\epsilon_1})\gamma \leq p_1 \leq \gamma$. Then, the following holds:*

$$\left|\mathbf{E}_{\mathbf{y}\sim\mathbf{Ber}(s,1-p_1)}[f_{\mathrm{Ext}}(\mathbf{y})] - \frac{1}{2}\right| \leq B^{-\Omega(1)}.$$

The proof of Lemma 5.2 is presented in Section 5.5 and the Lemma 5.3 is proved in Section 5.7.

5.4 Preparation for the Proof: Some Definitions and Easy Claims

In this short section, we record a few useful definitions and claims.

Define the following:

$$f^i_{\mathrm{Ext}}(y) = \bigvee_{1\leq j\leq M} \bigwedge_{\ell\in P^i_j} y_\ell,$$

where $i \in \{0,1\}^r$. Let y be sampled from $\mathbf{Ber}(s, 1-p_1)$. Define F_i be the event $f^i_{\mathrm{Ext}}(y) = 0$.

We record the following simple claims which are direct from the above definitions.

Claim 5.4. *For any $i \in \{0,1\}^r, j \in \{0,1\}^m$, $\mathbf{Pr}_{\mathbf{y}\sim\mathbf{Ber}(s,1-p_1)}[\bigwedge_{\ell\in P^i_j} \mathbf{y}_\ell = 1] = (1-p_1)^B = p_2$.*

Claim 5.5. *For any $i \in \{0,1\}^r$, $\mathbf{Pr}[F_i] = \mathbf{Pr}_{\mathbf{y}\sim\mathbf{Ber}(s,1-p_1)}[f^i_{\mathrm{Ext}}(\mathbf{y}) = 0] = (1-p_2)^M = p_3 = \frac{\alpha}{R}$.*

Claim 5.6. $\mathbf{E}_{\mathbf{y}\sim\mathbf{Ber}(s,1-p_1)}[f_{\mathrm{Ext}}(\mathbf{y})] = 1 - \mathbf{Pr}\left[\bigvee_{1\leq i\leq R} F_i\right]$.

5.5 Proof of Lemma 5.2 : Bound on Influence of Coalitions on f_{Ext}

Let Q be any set of variables of size $q < s^{1-\delta}$. We prove that $\mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f_{\mathrm{Ext}}) \leq q/s^{1-\delta}$. Recall that $f_{\mathrm{Ext}}(x) = \bigwedge_{i\in\{0,1\}^r} f^i_{\mathrm{Ext}}(x)$. The following is an easy observation: if for some fixing of the variables in $\overline{Q} = [n] \setminus Q$, f_{Ext} remains undetermined, it must be that at least some f^i_{Ext} is undetermined. Thus, by a union bound, we have

$$\mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f_{\mathrm{Ext}}) \leq \sum_{i\in\{0,1\}^r} \mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f^i_{\mathrm{Ext}}). \tag{8}$$

We first show that for any $i \in \{0,1\}^r$, $\mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f^i_{\mathrm{Ext}}) \leq \frac{1}{R}$. This follows from a direct calculation using the structure of f^i_{Ext}.

Claim 5.7. *For any $i \in \{0,1\}^r$, $\mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f^i_{\mathrm{Ext}}) \leq \frac{1}{R}$.*

Proof. The variables in Q can influence the outcome of f_{Ext} only if the AND of each block that does not contain a variable from Q evaluates to 0. There are at most q blocks of P^i which contain a variable from Q, and hence at least $M - q$ blocks with no variables from Q. For a y sampled

from $\mathbf{Ber}(s, 1-p_1)$, the probability that the AND of a block evaluates to 0 is exactly p_2. Thus, the probability that the AND of each block not containing a variable from Q evalues to 0 is at most $(1-p_2)^{M-q}$. Claim A.3 shows that $(1-p_2)^{M-q} \leq 1/R$. Thus the influence of Q is bounded by $\frac{1}{R}$. □

However, just using the bound from the above claim in (8) implies $\mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f_{\mathrm{Ext}}) \leq 1$ which is trivial. We show that the influence of the set Q is in fact much smaller than $\frac{1}{R}$ for most f^i_{Ext}. Informally, we prove the following. In Claim 5.10, we show that for most (i.e., $1-o(1)$ fraction of) $i \in \{0,1\}^r$, the set Q is 'well-spread' across the blocks of the partition P^i (Definition 5.8 below makes the notion of well-spread precise). We call such a P^i as 'good' (with respect to Q). In Claim 5.9 we show that the influence of Q on f^i_{Ext} (corresponding to a good P^i) is in fact smaller than $\frac{1}{2R}$ (see Claim 5.9 for the precise bound). The proof now follows using (8). The proof of Claim 5.10 uses the fact that seeded extractors are good samplers (see Section 2.4.1). Claim 5.9 follows from a direct calculation using the structure of f^i_{Ext}. We now present the details.

Definition 5.8. *For any $i \in \{0,1\}^r$ and $j \in \{0,1\}^m$, define a block P^i_j to be bad with respect to a subset of variables Q if $|P^i_j \cap Q| \geq 2\epsilon B$. Further call a partition P^i bad with respect to Q if it has a block which is bad. Otherwise, P^i is good.*

Claim 5.9. *Let P^i be a partition that is good with respect to a subset of variables Q, $|Q| = q$. If $q \leq s^{1-\delta}$, then $\mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f^i_{\mathrm{Ext}}) \leq \frac{q}{2Rs^{1-\delta}}$.*

Proof. We note that there are at least $M-q$ blocks in P^i that do not have any variables from Q. Each of the remaining blocks have at most $2\epsilon B$ variables from Q. An assignment of x leaves f^i_{Ext} undetermined only if: (a) there is no AND gate at depth 1 in f^i_{Ext} which outputs 1 and (b) There is at least one block with a variable from Q such that the non-Q variables are all set to 1. These two events are independent. Since there are at least $M-q$ blocks that do not have any variables from Q, the probability of (a) is bounded by $(1-p_2)^{M-q}$. From the calculation done in Claim 5.7, we have $(1-p_2)^{M-q} \leq 1/R$. We now bound the probability of (b). If there are h variables of Q in P^i_j, the probability that the non-Q variables are all 1's is exactly $(1-p_1)^{B-h}$. Thus the probability of event (b) is bounded by $q(1-p_1)^{B(1-2\epsilon)}$. By Claim A.4, we have $q(1-p_1)^{B(1-2\epsilon)} \leq \frac{q}{2s^{1-\delta}}$. This completes the proof. □

Claim 5.10. *Consider any subset of variables Q of size q. If $q \leq s^{1-\delta}$, then there are less than $2KM$ bad partitions with respect to Q.*

Proof. Suppose to the contrary that there are at least $2KM$ bad partitions with respect to Q. It follows by an averaging argument that there exists $j \in \{0,1\}^m$ such that the number of bad blocks among the $\{P^i_j : i \in \{0,1\}^r\}$ is at least $2K$. Recall that $P^i_j = \{(z, \mathrm{Ext}(i,z) \oplus j) : z \in \{0,1\}^b\}$. We now define the function $\mathrm{Ext}_j(x,y) = (y, \mathrm{Ext}(x,y) \oplus j)$. Since we have chosen Ext to be a strong-seeded extractor (recall that a shift-design extractor is also a strong-seeded extractor, see Definition 2.13), it follows that Ext_j is a seeded extractor for min-entropy k with error ϵ.

Let $\mathcal{N}_j(i)$ denote the set of neighbors of $i \in \{0,1\}^r$ in the graph corresponding to Ext_j. By construction of Ext_j, for any $i \in \{0,1\}^r$, $P^i_j = \mathcal{N}_j(i)$. It follows from the above discussion that

$$|\{i \in \{0,1\}^r : |\mathcal{N}_j(i) \cap Q| \geq 2\epsilon B|\}| \geq 2K.$$

Let $\mu_Q = q/M$. We have,

$$\mu_Q = q/M \leq s^{1-\delta}/M < \epsilon \qquad \text{(using (7))}$$

Thus, we have

$$|\{i \in \{0,1\}^r : |\mathcal{N}_j(i) \cap Q| \geq (\epsilon + \mu_Q)B\}| \geq |\{i \in \{0,1\}^r : |\mathcal{N}_j(i) \cap Q| \geq 2\epsilon B|\}| \geq 2K.$$

However this contradicts Theorem 2.12 used on Ext_j. Thus the number of bad blocks is bounded by $2KM$. □

Thus, we have

$$\begin{aligned}
\mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f_{\mathrm{Ext}}) &= \sum_{i \in \{0,1\}^r : P^i \text{ is bad}} \mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f_{\mathrm{Ext}}) \\
&\quad + \sum_{i \in \{0,1\}^r : P^i \text{ is good}} \mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f_{\mathrm{Ext}}) && \text{(using (8))} \\
&\leq (2KM) \cdot \frac{1}{R} + \sum_{i \in \{0,1\}^r : P^i \text{ is good}} \mathbf{I}_{Q,\mathbf{Ber}(s,1-p_1)}(f_{\mathrm{Ext}}) && \text{(using Claims 5.10, 5.7)} \\
&\leq \frac{2KM}{R} + R \cdot \frac{q}{2Rs^{1-\delta}} && \text{(using Claim 5.9)} \\
&= \frac{2}{R^{1-3\delta}} + \frac{q}{2s^{1-\delta}} && \text{(since } M = R^\delta, K = R^{2\delta}\text{)} \\
&< \frac{q}{s^{1-\delta}}
\end{aligned}$$

where the last inequality follows since $s = BM = M^{1+\delta_1} = R^{\delta(1+\delta_1)} < R^{2\delta}$ by (6), and hence $\frac{2}{R^{1-3\delta}} < \frac{1}{s} < \frac{q}{2s^{1-\delta}}$. This completes the proof of Lemma 5.2.

5.6 Towards Bounding Bias of f_{Ext}

In this section, we take an important step towards proving that f_{Ext} is almost balanced with respect to $\mathbf{Ber}(s, 1-p_1)$, i.e.,

$$\mathbf{E}_{\mathbf{y} \sim \mathbf{Ber}(s,1-p_1)}[f_{\mathrm{Ext}}(\mathbf{y})] \approx \frac{1}{2}.$$

To refresh the reader's memory, we first recall the construction of f_{Ext} and a few other definitions from Sections 5.1, 5.4. For each $v \in \{0,1\}^r$, define the partition P^v with blocks P^v_w, $w \in \{0,1\}^m$ where $P^v_w = \{(j, z_j \oplus w) : j \in \{0,1\}^b\}$. By Claim 5.1, P^v forms an equi-partition of $[s]$ with M blocks of size B. Then,

$$f^v_{\mathrm{Ext}}(y) = \bigvee_{1 \leq j \leq M} \bigwedge_{\ell \in P^v_j} y_\ell,$$

where $v \in \{0,1\}^r$. Finally, we have $f_{\mathrm{Ext}} : \{0,1\}^s \to \{0,1\}$ as:

$$f_{\mathrm{Ext}}(y) = \bigwedge_{1 \leq i \leq R} f^i_{\mathrm{Ext}}(y).$$

Recall that $p_2 = (1-p_1)^B, \quad p_3 = \frac{\alpha}{R} = (1-p_2)^M$. Further, F_i is the event $f^i_{\mathrm{Ext}}(y) = 0$, where y is sampled from $\mathbf{Ber}(s, 1-p_1)$.

By Claim 5.6, for any $v \in \{0,1\}^r$, we have $p_3 = \mathbf{Pr}_{\mathbf{y}\sim\mathbf{Ber}(s,1-p_1)}[f^v_{\mathrm{Ext}}(\mathbf{y}) = 0]$. Using Claim A.2, we have $p_3 \approx \frac{\ln 2}{R}$. Thus, if it was the case that the functions f^i_{Ext} were on disjoint sets of variables, then one could estimate $\mathbf{Pr}[f_{\mathrm{Ext}} = 1] \approx (1 - \frac{\ln 2}{R})^R \approx \frac{1}{2}$, and conclude that f_{Ext} is almost balanced with respect to $\mathbf{Ber}(s, 1-p_1)$.

However, the functions f^v_{Ext} are on the same set of variables, and hence the analysis described above (assuming independence) breaks down. Our key result in this section is that if the partitions are 'pairwise-good', then in fact the f^v_{Ext}'s behave as though they are independent in the following sense: for any c that is not too large and arbitrary $1 \le i_1 < \ldots < i_c \le R$, $\mathbf{Pr}\left[\bigwedge_{1\le g\le c} F_{i_g}\right] \approx \prod_{1\le g\le c} \mathbf{Pr}\left[F_{i_g}\right]$. We formally state this in Lemma 5.12.

We make the notion of 'pairwise-good' precise in Definition 5.11. Roughly, the notion of pairwise-good corresponds to the requirement that no two blocks from any two partitions have large intersection. The fact the partitions are pairwise-good follows from the fact that they are generated using the neighbor graph of a shift-design extractor. We prove this in the next section (see Lemma 5.15). We now make things more precise.

For ease of presentation, we slightly abuse notation and relabel the partitions in Construction 1 as $P^1, \ldots, P^R$, where for any $i \in [R]$, P^i corresponds to the partition P^{v_i} with v_i being the r bit string for the integer $i-1$. We use this notation in Section 5.7 as well.

Definition 5.11. *Let P^i, P^j be two equi-partitions of $[s]$ with blocks of size B. Then (P^i, P^j) is said to be pairwise-good if the size of the intersection of any block of P^i and any block of P^j is at most $0.9B$.*

A collection of equi-partitions $\mathcal{P} = \{P^1, \ldots, P^R\}$ is pairwise-good if for any distinct $i, j \in [R]$, (P^i, P^j) is pairwise-good.

The following is the main result of this section.

Lemma 5.12. *There exist constants $\beta_1, \beta_2 > 0$ such that for any $c \le s^{\beta_1}$, and arbitrary $1 \le i_1 < \ldots < i_c \le R$, the following holds:*

$$\left(\frac{\alpha}{R}\right)^c \le \mathbf{Pr}\left[\bigwedge_{1\le g\le c} F_{i_g}\right] \le \left(\frac{\alpha}{R}\right)^c \left(1 + \frac{1}{M^{\beta_2}}\right).$$

We also recall Janson's inequality [Jan90, BS89] which will play a crucial role in the proof. We follow the presentation in [AS92].

Theorem 5.13 (Janson's Inequality [Jan90, BS89, AS92])**.** *Let Ω be a finite universal set and let $\mathcal{O}$ be a random subset of Ω constructed by picking each $h \in \Omega$ independently with probability p_h. Let $Q_1, \ldots, Q_\ell$ be arbitrary subsets of Ω, and let $\mathcal{E}_i$ be the event $Q_i \subseteq \mathcal{O}$. Define*

$$\Delta = \sum_{i<j : Q_i \cap Q_j \neq \emptyset} \mathbf{Pr}\left[\mathcal{E}_i \wedge \mathcal{E}_j\right], \qquad D = \prod_{i=1}^{\ell} \mathbf{Pr}\left[\overline{\mathcal{E}_i}\right].$$

Assume that $\mathbf{Pr}[\mathcal{E}_i] \le \tau$ for all $i \in [\ell]$. Then

$$D \le \mathbf{Pr}\left[\bigwedge \overline{\mathcal{E}_i}\right] \le De^{\frac{\Delta}{1-\tau}}.$$

Proof of Lemma 5.12. Without loss of generality suppose $i_g = g$ for $g \in [c]$. We use Janson's inequality with $\Omega = [s]$, and $\mathcal{O}$ constructed by picking each $h \in [s]$ with probability $1 - p_1$. Further let $\mathcal{E}_{i,j}$ be the event that $P^i_j \subseteq \mathcal{O}$. Intuitively, $\mathcal{O}$ denotes the set of coordinates in y that are set to 1 for a sample y from $\mathbf{Ber}(s, 1-p_1)$. With this interpretation, the event $f^i_{\mathrm{Ext}}(y) = 0$ exactly corresponds to the event $\bigwedge_{1 \le j \le M} \overline{\mathcal{E}_{i,j}}$. Thus, we have

$$\mathbf{Pr}\left[\bigwedge_{1 \le g \le c} F_g\right] = \mathbf{Pr}\left[\bigwedge_{i \in [c], j \in \{0,1\}^m} \overline{\mathcal{E}_{i,j}}\right].$$

We now estimate D, Δ, γ to apply Janson's inequality. For any $i \in [c], j \in \{0,1\}^m$, we have $\mathbf{Pr}[\mathcal{E}_{i,j}] = \mathbf{Pr}[P^i_j \subseteq \mathcal{O}] = (1-p_1)^B = p_2$. Note that $\tau = p_2 < \frac{1}{2}$. Further

$$D = \prod_{i \in [c], j \in \{0,1\}^m} \mathbf{Pr}\left[\overline{\mathcal{E}_{i,j}}\right] = (1-p_2)^{Mc} = p_3^c = \left(\frac{\alpha}{R}\right)^c.$$

Finally, we have

$$\begin{aligned}
\Delta &= \sum_{i_1 < i_2 \in [c], j_1, j_2 \in \{0,1\}^m : P^{i_1}_{j_1} \cap P^{i_1}_{j_1} \neq \emptyset} \mathbf{Pr}[\mathcal{E}_{i_1,j_1} \wedge \mathcal{E}_{i_2,j_2}] \\
&\le \binom{c}{2} \max_{i_1 < i_2 \in [c]} \left\{ \sum_{j_1, j_2 \in \{0,1\}^m : P^{i_1}_{j_1} \cap P^{i_1}_{j_1} \neq \emptyset} \mathbf{Pr}[\mathcal{E}_{i_1,j_1} \wedge \mathcal{E}_{i_2,j_2}] \right\}
\end{aligned}$$

We observe that any $P^{i_1}_{j_1}$ can intersect at most B blocks of a partition P^{i_2}, where $i_1 \neq i_2$. Thus, the total number of pairs of blocks that intersect between two partitions P^{i_1} and P^{i_2}, $i_1 \neq i_2$, is bounded by $MB = s$. Thus, continuing with the estimate, we have

$$\Delta \le \binom{c}{2} \cdot s \cdot \max_{i_1 < i_2 \in [c], j_1, j_2 \in \{0,1\}^m : P^{i_1}_{j_1} \cap P^{i_2}_{j_2} \neq \emptyset} \{\mathbf{Pr}[\mathcal{E}_{i_1,j_1} \wedge \mathcal{E}_{i_2,j_2}]\}$$

Further, recall that $\mathcal{P}$ is pairwise-good. Thus it follows that for any distinct $i_1, i_2 \in [c]$, and $j_1, j_2 \in \{0,1\}^m$, $|P^{i_1}_{j_1} \cap P^{i_2}_{j_2}| \le 0.9B$. Thus, $|P^{i_1}_{j_1} \cup P^{i_2}_{j_2}| \ge 1.1B$ and hence for any $i_1 < i_2 \in [c], j_1, j_2 \in \{0,1\}^m$,

$$\mathbf{Pr}[\mathcal{E}_{i_1,j_1} \wedge \mathcal{E}_{i_2,j_2}] \le (1-p_1)^{\frac{11B}{10}} = p_2^{\frac{11}{10}}. \tag{9}$$

Thus,

$$\begin{aligned}
\Delta &\le \binom{c}{2} \cdot s \cdot p_2^{\frac{11}{10}} && \text{(using (9))} \\
&< \frac{1}{M^{\frac{1}{20} - 3\beta_1}} && \text{(by Claim A.7)}
\end{aligned}$$

We set $\beta_1 = 1/90$. It follows that

$$\Delta < M^{-\beta'},$$

where $\beta' = 1/70$.

Invoking Janson's inequality, we have

$$\left(\frac{\alpha}{R}\right)^c \leq \mathbf{Pr}\left[\bigwedge_{1\leq g\leq c} F_g\right] \leq \left(\frac{\alpha}{R}\right)^c e^{2M^{-\beta'}} \leq \left(1+\frac{4}{M^{\beta'}}\right)\left(\frac{\alpha}{R}\right)^c,$$

where the last inequality follows using the fact that for $0<\theta\leq 1$, $e^\theta \leq 1+2\theta$. This concludes the proof. □

5.7 Proof of Lemma 5.3: Bound on the Bias of f_{Ext}

The following two lemmas directly imply Lemma 5.3.

Lemma 5.14. *If $\mathcal{P}$ is pairwise-good, $|p-\frac{1}{2}| \leq B^{-\Omega(1)}$, where $p = \mathbf{Pr}_{\mathbf{y}\sim\mathbf{Ber}(s,1-p_1)}[f_{\text{Ext}}(\mathbf{y})=0]$.*

Lemma 5.15. *The set of partitions $\mathcal{P}=\{P^1,\ldots,P^R\}$ in Construction 1 is pairwise-good.*

Lemma 5.12 proved in Section 5.6 is a key component in proving Lemma 5.14. We begin by proving Lemma 5.15, which is easy to derive from the fact that our construction of P^i's uses shift-design extractors.

Proof of Lemma 5.15. Let $P^{i_1}_{j_1}$ and $P^{i_2}_{j_2}$ be any two blocks such that $i_1\neq i_2$. We need to prove that $|P^{i_1}_{j_1}\cap P^{i_2}_{j_2}|\leq 0.9B$. Recall that $P^{i_1}_{j_1}=\{(z,\text{Ext}(i_1,z)\oplus j_1): z\in\{0,1\}^b\}$, and similarly $P^{i_2}_{j_2}=\{(z,\text{Ext}(i_2,z)\oplus j_2): z\in\{0,1\}^b\}$. The bound on $|P^{i_1}_{j_1}\cap P^{i_2}_{j_2}|$ now directly follows from the fact that Ext is a $\frac{1}{10}$-shift-design extractor. □

We use the rest of the section to prove Lemma 5.14.

Proof of Lemma 5.14. Let $\mathcal{P}=\{P^1,\ldots,P^R\}$ be pairwise-good. We have,

$$p = \mathbf{Pr}_{\mathbf{y}\sim\mathbf{Ber}(s,1-p_1)}[f_{\text{Ext}}(\mathbf{y})=0] = \mathbf{Pr}\left[\bigvee_{1\leq i\leq R} F_i\right].$$

For $1\leq c\leq R$, let

$$S_c = \sum_{1\leq i_1<\ldots<i_c\leq R} \mathbf{Pr}\left[\bigwedge_{1\leq g\leq c} F_{i_g}\right].$$

Using the inclusion-exclusion principle, it follows that for any even $a\in[R]$,

$$\sum_{c=1}^{a}(-1)^{(c-1)}S_c \leq p \leq \sum_{c=1}^{a+1}(-1)^{(c-1)}S_c. \tag{10}$$

Fix $a=s^{\beta_3}$ (assume that a is even), $\beta_3=\min\{\beta_1/2,\beta_2/2\}$, where β_1,β_2 are the constants in Lemma 5.12. Now the idea is to use Lemma 5.12 to obtain tight estimates for S_c. Combining this with (10) proves the desired bound on p (recall that $p=\mathbf{Pr}_{\mathbf{y}\sim\mathbf{Ber}(s,1-p_1)}[f_{\text{Ext}}(\mathbf{y})=0]$).

Claim 5.16. $e^{-\alpha}-\frac{1}{M^{\beta_2/2}} \leq \sum_{c=1}^{a}(-1)^{c-1}S_c < \sum_{c=1}^{a+1}(-1)^{c-1}S_c \leq e^{-\alpha}+\frac{1}{M^{\beta_2/2}}$.

Proof. For any $c \le a+1$, using Lemma 5.12, we have

$$\binom{R}{c}\left(\frac{\alpha}{R}\right)^c \le S_c \le \binom{R}{c}\left(\frac{\alpha}{R}\right)^c\left(1+\frac{1}{M^{\beta_2}}\right).$$

By Claim A.8, it follows that for any $c \le a$,

$$\left|S_c - \frac{\alpha^c}{c!}\right| \le \frac{1}{M^{\beta_2}} \tag{11}$$

Also note that

$$S_{a+1} \le \frac{1}{a!} + \frac{1}{M^{\beta_2}} < \frac{2}{M^{\beta_2}}, \tag{12}$$

using $a = s^{\beta_3}$ and the inequality $a! \ge (a/e)^a$ (thus, using $s = MB$ and $B > e$, we have $a! > M^{\beta_3 M^{\beta_3}} > M^{\beta_2}$).

Finally, by the classical Taylor's theorem and the inequalities above, we have

$$\left|e^{-\alpha} - \sum_{c=1}^{a}(-1)^{c-1}\frac{\alpha^c}{c!}\right| < \frac{\alpha^{a+1}}{(a+1)!} < (\alpha e/(a+1))^{a+1} < \frac{1}{M^{\beta_2}}. \tag{13}$$

,

We are now ready to prove Claim 5.16. We have,

$$\begin{aligned}
\left|\sum_{c=1}^{a}(-1)^{c-1}S_c - e^{-\alpha}\right| &\le \left|\sum_{c=1}^{a}(-1)^{c-1}S_c - \sum_{c=1}^{a}(-1)^{c-1}\frac{\alpha^c}{c!}\right| + \frac{1}{M^{\beta_2}} && \text{(using (13))}\\
&\le \left(\sum_{c=1}^{a}\left|S_c - \frac{\alpha^c}{c!}\right| + \frac{1}{M^{\beta_2}}\right) + \frac{1}{M^{\beta_2}}\\
&\le \frac{a+1}{M^{\beta_2}} && \text{(using (11))}
\end{aligned}$$

Using (12), we also have

$$\left|\sum_{c=1}^{a+1}(-1)^{c-1}S_c - e^{-\alpha}\right| \le \frac{a+3}{M^{\beta_2}}.$$

Finally, using the fact that $(a+3)M^{-\beta_2} \le M^{-\beta_2/2}$, it follows that

$$e^{-\alpha} - \frac{1}{M^{\beta_2/2}} \le \sum_{c=1}^{a}(-1)^{c-1}S_c < \sum_{c=1}^{a+1}(-1)^{c-1}S_c \le e^{-\alpha} + \frac{1}{M^{\beta_2/2}}.$$

□

We are now very close to proving Lemma 5.14. Using (10) and Claim 5.16, we have

$$|p - e^{-\alpha}| \le \frac{1}{M^{\beta_2/2}}.$$

Using Claim A.6, we have

$$\left|e^{-\alpha} - \frac{1}{2}\right| \le \frac{1}{2B^{\epsilon_3}}.$$

Hence, we have $\left|p - \frac{1}{2}\right| < \frac{1}{2B^{\epsilon_3}} + \frac{1}{M^{\beta_2/2}} = B^{-\Omega(1)}$ since $M > B$. This concludes the proof. □

5.8 Proof of Theorem 3

We obtain our resilient function with respect to the uniform distribution by a simple modification of Construction 1 using the fact that we can simulate biased bits quite accurately via small CNFs (see Lemma 2.24).

Construction 2: A constant $\delta \in (0, 1/10)$ and an integer $n > 0$ are input parameters. We set $\epsilon_1 = \delta/4$. We think of r as an unfixed variable and set up the remaining variables, as in Construction 1 following the description in Section 5.2. We do not fix the parameter p_1 as yet.
Next, we fix the parameter r as follows. Let the parameter ν in Lemma 2.24 be set to γ/B^{ϵ_1} and let $\mathcal{C}$ be the size h monotone CNF circuit guaranteed by Lemma 2.24, where $h < B^{1+2\epsilon_1}$. Thus, $(1 - B^{-\epsilon_1})\gamma \leq \mathbf{Pr}_{\mathbf{x}\sim\mathbf{U}_h}[\mathcal{C}(\mathbf{x}) = 0] < \gamma$. Choose the largest integer r such that we have $n' = sh = BMh < n$. It follows that for this choice of r, $n' = \Omega(n)$. Set $p_1 = \mathbf{Pr}_{\mathbf{x}\sim\mathbf{U}_h}[\mathcal{C}(\mathbf{x}) = 0]$. It is immediate that p_1 satisfies (5).
Let $f_{\mathrm{Ext}} : \{0,1\}^s \to \{0,1\}$ be the function from Construction 1, with Ext instantiated as in Construction 1 (using the Trevisan extractor). Define f be the function derived from f_{Ext} by replacing each variable y_i by a copy of the monotone CNF $\mathcal{C}$ set up above. Thus f is defined on n' bits.

We observe that:

- the size of the coalition (denoted by the parameter q) is at most $n^{1-\delta} = (n')^{1-\delta'}$, where $\delta' = \delta - o(1)$. Thus, we may assume $n = n' = BMh$ and $\delta = \delta'$.
- since TExt is a polynomial time function, f_{Ext} can be constructed in polynomial time. Thus f is computable by a polynomial time algorithm. Further, f is an $O(RMBh) = n^{O(1)}$ sized monotone circuit in AC^0 of depth 4.

The required bounds on the bias of f and on $\mathbf{I}_q(f)$ are now straightforward using Lemma 5.3 and Lemma 5.2 respectively, and we omit the details. □

6 Proof of Theorem 4

In this short section we prove Theorem 4, which gives an explicit nearly balanced function f with small $\mathbf{I}_{q,t}(f)$. The proof is almost direct from the results of Sections 4 and 5. Informally, the result in Section 4 is that if f is a constant depth monotone circuit, then in order to prove an upper bound for $\mathbf{I}_{q,t}(f)$, it is in fact enough to upper bound $\mathbf{I}_q(f)$. In Section 5, we exactly construct such a constant depth monotone circuit that has small $\mathbf{I}_q(f)$. We now present the details.

Fix $\delta = \nu/2$. Let $f : \{0,1\}^n \to \{0,1\}$ be the function from Theorem 3 such that for any $q > 0$, $\mathbf{I}_q(f) \leq q/n^{1-\delta}$. Also we have that f is monotone and can be computed by a depth 4 AC^0 circuit $\mathcal{C}$ of size $m = \mathrm{poly}(n)$.

Fix $\epsilon_3 = 1/n$. Thus by Theorem 4.1, it follows that there exists a constant b such that for any $t \geq b(\log(m/\epsilon_3))^{18}$, $q > 0$,

$$\mathbf{I}_{q,t}(f) \leq \epsilon_3 + \frac{q}{n^{1-\delta}} < \frac{2q}{n^{1-\delta}} < \frac{q}{n^{1-\nu}},$$

where the last inequality uses the fact that $\delta = \nu/2$ and hence, assuming n is large enough, $n^{\nu/2} > 2$.

We conclude by noting that f is unbiased under any t-wise independent distribution. Using Theorem 4.1 and the fact that f is computable by a constant depth polyomial sized circuit $\mathcal{C}$ that for any t-wise independent distribution $\mathcal{D}$, we have

$$\left|\mathbf{E}_{\mathbf{x}\sim\mathcal{D}}[f(x)] - \frac{1}{2}\right| \leq \frac{1}{n} + \frac{1}{n^{\Omega(1)}}.$$

7 Proof of Theorem 1

We informally recall Theorem 1. We show that for all n and ϵ, there exists a 2-source extractor $2\text{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ computable in time $\text{poly}(n, 1/\epsilon)$ for min-entropy at least $C_1 \log^C(n/\epsilon)$ and error ϵ.

We set up the required ingredients and parameters as follows:

- Let t, k, ϵ_1 be parameters that we fix later. Let $\text{nmExt} : \{0,1\}^n \times \{0,1\}^{d_1} \to \{0,1\}$ be a (t, k, ϵ_1)-non-malleable extractor from Theorem 2.19. Thus $d_1 = O(t^2 \log\log n) + O(t^2 \log(1/\epsilon_1)) \cdot o(\log\log(1/\epsilon_1))$, for some constant c_1. For such an extractor to exists, we require $k \geq c' t \log\log n + c' t \log(1/\epsilon_1)) \cdot o(\log\log(1/\epsilon_1))$.
- Let $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{d_1}$ be the seeded extractor from Theorem 2.11, with the modification that the seed is concatenated with the output, set to extract from min-entropy $k/2$ with error ϵ_2. Thus, $d = c_2 \log(n/\epsilon_2)$, for some constant c_2. Let $D = 2^d = (n/\epsilon_2)^{c_2}$. Such an extractor exists for $k \geq 3d_1$.
- Let $t = b(\log(D/\epsilon))^{18}$, for a large enough constant b.
- Choose $\delta > 0$, such that $\delta' = 2\delta c_2 < 9/10$.
- Let $f : \{0,1\}^D \to \{0,1\}$ be the function from Theorem 4 such that $\mathbf{I}_{q,t}(f) \leq q/D^{1-\delta}$ and for any t-wise independent distribution $\mathcal{D}$, $\left|\mathbf{E}_{v\sim\mathcal{D}}[f(v)] - \frac{1}{2}\right| \leq D^{-\beta}$ for some small constant β.
- Let $\epsilon_3 = C_3 t \sqrt{\epsilon_1} D^t$, for a large enough constant C_3.
- Pick ϵ_1, ϵ_2 such that the following inequalities are satisfied:
  - $D = (n/\epsilon_2)^{c_2} \geq \max\{(8/\epsilon)^{1/\beta}, (8/\epsilon)^{2/\delta}\}$,
  - $\epsilon_2 \leq D^{-2\delta}/2 = (\epsilon_2/n)^{\delta'}$,
  - $\sqrt{\epsilon_1} \leq \frac{1}{4C_3 t D^{t+1}}$.

  Thus, we can pick $\epsilon_2 = \min\{n\epsilon^{\frac{1}{c_2\beta}}, n\epsilon^{\frac{2}{c_2\delta}}, 1/n^{\delta'/(1-\delta')}\}$ and $\epsilon_1 = 1/(4C_3 t D^{t+1})^2$.
- With this setting of parameters, it can be checked that we require $k \geq C_1(\log(n/\epsilon))^{56}$, for a large enough constant C_1.

Let $\{0,1\}^{d_2} = \{r_1, \ldots, r_{D_2}\}$. Define

$$\text{reduce}(x, y) = \text{nmExt}(x, \text{Ext}(y, r_1)) \circ \ldots \circ \text{nmExt}(x, \text{Ext}(y, r_{D_2}))$$

and

$$2\text{Ext}(x, y) = f(\text{reduce}(x, y)).$$

Let $\mathbf{X}$ and $\mathbf{Y}$ be any two independent (n,k)-sources, where $k \geq C_1 \log^C(n/\epsilon)$. We prove that

$$|(2\text{Ext}(\mathbf{X},\mathbf{Y}),\mathbf{Y}) - (\mathbf{U}_1,\mathbf{Y})| \leq \epsilon.$$

Let $\mathbf{Z} = \text{reduce}(\mathbf{X},\mathbf{Y})$. Theorem 3.1 implies that with probability at least $1 - 2 \cdot 2^{-k/2} > 1 - \frac{\epsilon}{2}$ over $\mathbf{y} \sim \mathbf{Y}$, the conditional distribution $\mathbf{Z}|\mathbf{Y} = \mathbf{y}$ is ϵ_3-close to a (q,t)-non-oblivious bit-fixing source on M bits, where by our choice of parameters,

- $q = (\sqrt{\epsilon_1} + \epsilon_2)D < D^{1-2\delta}$,
- $\epsilon_3 = C_3 t \sqrt{\epsilon_1} D^t < \epsilon/4$.

Thus, for each such $\mathbf{y}$,

$$\begin{aligned} |f(\text{reduce}(\mathbf{X},\mathbf{y})) - \mathbf{U}_1| &\leq \epsilon_3 + \frac{q}{D^{1-\delta}} + D^{-\beta} \\ &\leq \frac{\epsilon}{4} + D^{-\delta} + \frac{\epsilon}{8} \\ &\leq \frac{\epsilon}{4} + \frac{\epsilon}{8} + \frac{\epsilon}{8} = \frac{\epsilon}{2}. \end{aligned}$$

Thus, we have

$$|(2\text{Ext}(\mathbf{X},\mathbf{Y}),\mathbf{Y}) - (\mathbf{U}_1,\mathbf{Y})| \leq \epsilon.$$

We finally note that it is direct from the description of the construction that the extractor runs in time $\text{poly}(n, 1/\epsilon)$. This completes the proof.

Acknowledgments

We thank anonymous referees for helpful comments that led to improvements in the presentation of the paper. In particular, a referee for the Annals of Mathematics gave detailed valuable comments and suggestions. We are grateful to Xin Li for an observation which led to our theorem working with general ϵ. We also thank Ran Raz for reminding us that every 2-source extractor is strong.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[AGM03] Noga Alon, Oded Goldreich, and Yishay Mansour. Almost k-wise independence versus k-wise independence. *Inf. Process. Lett.*, 88(3):107–110, 2003.

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

[Alo98] Noga Alon. The Shannon capacity of a union. *Combinatorica*, 18(3):301–310, 1998.

[AS92] Noga Alon and Joel Spencer. *The Probabilistic Method*. John Wiley, 1992.

[Bar06] Boaz Barak. A Simple Explicit Construction of an $n^{\tilde{O}(\log n)}$-Ramsey Graph. Technical report, Citeseer, 2006.

[BCDT18] Avraham Ben-Aroya, Gil Cohen, Dean Doron, and Amnon Ta-Shma. Two-source condensers with low error and small entropy gap via entropy-resilient functions. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:66, 2018.

[BDT17] Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. An efficient reduction from two-source to non-malleable extractors: achieving near-logarithmic min-entropy. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1185–1194, 2017.

[BDT18] Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. Near-optimal strong dispersers, erasure list-decodable codes and friends. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:65, 2018.

[BH05] Boaz Barak and Shai Halevi. A model and architecture for pseudo-random generation with applications to/dev/random. In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 203–212, 2005.

[BIW06] Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. *SIAM J. Comput.*, 36(4):1095–1118, December 2006.

[BKS+10] Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors. *J. ACM*, 57(4), 2010.

[BL85] Michael Ben-Or and Nathan Linial. Collective coin flipping, robust voting schemes and minima of Banzhaf values. In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 408–416, 1985.

[Bou05] J. Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 01(01):1–32, 2005.

[Bra10] Mark Braverman. Polylogarithmic independence fools AC^0 circuits. *J. ACM*, 57(5), 2010.

[BRSW12] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2-source dispersers for $n^{o(1)}$ entropy, and Ramsey graphs beating the Frankl-Wilson construction. *Annals of Mathematics*, 176(3):1483–1543, 2012. Preliminary version in STOC '06.

[BS89] Ravi Boppona and Joel Spencer. A useful elementary correlation inequality. *Journal of Combinatorial Theory, Series A*, 50(2):305 – 307, 1989.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In *STOC*, 2016.

[CL16a] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors, and almost optimal privacy amplification protocols. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 158–167, 2016.

[CL16b] Eshan Chattopadhyay and Xin Li. Extractors for sumset sources. In *STOC*, 2016.

[Coh15] Gil Cohen. Local correlation breakers and applications to three-source extractors and mergers. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.

[Coh16a] Gil Cohen. Making the most of advice: New correlation breakers and their applications. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 188–196, 2016.

[Coh16b] Gil Cohen. Two-source dispersers for polylogarithmic entropy and improved Ramsey graphs. In *STOC*, 2016.

[Coh17] Gil Cohen. Towards optimal two-source extractors and ramsey graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1157–1170, 2017.

[CRS14] Gil Cohen, Ran Raz, and Gil Segev. Nonmalleable extractors with short seeds and applications to privacy amplification. *SIAM Journal on Computing*, 43(2):450–476, 2014.

[DKSS09] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the method of multiplicities, with applications to Kakeya sets and mergers. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 181–190, 2009.

[DLWZ14] Yevgeniy Dodis, Xin Li, Trevor D. Wooley, and David Zuckerman. Privacy amplification and non-malleable extractors via character sums. *SIAM Journal on Computing*, 43(2):800–830, 2014.

[DO03] Y. Dodis and R. Oliveira. On extracting private randomness over a public channel. In *RANDOM 2003, 7th International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 252–263, 2003.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *STOC*, pages 601–610, 2009.

[Erd47] P. Erdős. Some remarks on the theory of graphs. *Bull. Amer. Math. Soc.*, 53(4):292–294, 04 1947.

[Fei99] Uriel Feige. Noncryptographic selection protocols. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 142–153, 1999.

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

[Gop14] Parikshit Gopalan. Constructing Ramsey graphs from Boolean function representations. *Combinatorica*, 34(2):173–206, 2014.

[Gro00] Vince Grolmusz. Low rank co-diagonal matrices and Ramsey graphs. *Electr. J. Comb.*, 7, 2000.

[GRS06] Ariel Gabizon, Ran Raz, and Ronen Shaltiel. Deterministic extractors for bit-fixing sources by obtaining an independent seed. *SIAM J. Comput.*, 36(4):1072–1094, 2006.

[GSV05] S. Goldwasser, M. Sudan, and V. Vaikuntanathan. Distributed computing with imperfect randomness. In P. Fraigniaud, editor, *Proceedings of the 19th International Symposium on Distributed Computing DISC 2005*, volume 3724 of *Lecture Notes in Computer Science*, pages 288–302. Springer, 2005.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *J. ACM*, 56(4), 2009.

[Jan90] Svante Janson. Poisson approximation for large deviations. *Random Structures & Algorithms*, 1(2):221–229, 1990.

[JK99] Benjamin Jun and Paul Kocher. The Intel random number generator. *Cryptography Research Inc. white paper*, 1999.

[KKL88] Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on Boolean functions (extended abstract). In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 68–80, 1988.

[KLR09] Y. Kalai, X. Li, and A. Rao. 2-source extractors under computational assumptions and cryptography with defective randomness. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 617–626, 2009.

[KLRZ08] Y. Kalai, X. Li, A. Rao, and D. Zuckerman. Network extractor protocols. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 654–663, 2008.

[Li11] Xin Li. Improved constructions of three source extractors. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, June 8-10, 2011*, pages 126–136, 2011.

[Li12a] Xin Li. Design extractors, non-malleable condensers and privacy amplification. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 837–854, 2012.

[Li12b] Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, pages 688–697, 2012.

[Li13a] Xin Li. Extractors for a constant number of independent sources with polylogarithmic min-entropy. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 100–109, 2013.

[Li13b] Xin Li. New independent source extractors with exponential improvement. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 783–792, 2013.

[Li15] Xin Li. Three-source extractors for polylogarithmic min-entropy. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.

[Li16] Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 168–177, 2016.

[Li17] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1144–1156. ACM, 2017.

[Li18] Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. *CoRR*, abs/1804.04005, 2018.

[LRVW03] Chi-Jen Lu, Omer Reingold, Salil P. Vadhan, and Avi Wigderson. Extractors: optimal up to constant factors. In *STOC*, pages 602–611, 2003.

[Mek09] Raghu Meka. Explicit coin flipping protocols. Unpublished manuscript, 2009.

[Mek17] Raghu Meka. Explicit resilient functions matching ajtai-linial. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1132–1148, 2017.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.

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

[Ram30] Frank P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society, Series 2*, 30:264–286, 1930.

[Rao09a] Anup Rao. Extractors for a constant number of polynomially small min-entropy independent sources. *SIAM J. Comput.*, 39(1):168–194, 2009.

[Rao09b] Anup Rao. Extractors for low-weight affine sources. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, 2009.

[Raz05] Ran Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.

[RRV02] Ran Raz, Omer Reingold, and Salil Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. *JCSS*, 65(1):97–128, 2002.

[RZ01] A. Russell and D. Zuckerman. Perfect-information leader election in $\log^* n + O(1)$ rounds. *JCSS*, 63:612–626, 2001.

[RZ08] Anup Rao and David Zuckerman. Extractors for three uneven-length sources. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, 11th International Workshop, APPROX 2008, and 12th International Workshop, RANDOM 2008, Boston, MA, USA, August 25-27, 2008. Proceedings*, pages 557–570, 2008.

[Sha02] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

[Sha08] Ronen Shaltiel. How to get more mileage from randomness extractors. *Random Struct. Algorithms*, 33(2):157–186, 2008.

[Sip88] Michael Sipser. Expanders, randomness, or time versus space. *Journal of Computer and System Sciences*, 36(3):379–383, 1988.

[SV86] Miklos Santha and Umesh V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33:75–87, 1986.

[Tal17] Avishay Tal. Tight bounds on the Fourier spectrum of AC0. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 15:1–15:31, 2017.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, pages 860–879, 2001.

[Vio14] Emanuele Viola. Extractors for circuit sources. *SIAM J. Comput.*, 43(2):655–672, 2014.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures and Algorithms*, 11:345–367, 1997.

A Auxiliary Material for Section 5

This section contains supplementary material to the proofs and arguments done in Section 5.

A.0.1 Validity of Instantiation of Ext in Construction 1

Claim A.1. *The instantiation of* Ext *in Construction* 1 *(see Section* 5.2*) is valid and the parameters satisfy the inequalities (*6*) and (*7*).*

Proof. Recall that $\mathrm{Ext} : \{0,1\}^r \times \{0,1\}^b \to \{0,1\}^m$ is set to be the Trevisan extractor. It is set to extract from min-entropy k with error $\epsilon = 2^{-\delta_2\sqrt{r}}$ with output length $m = k/2$. By Theorem 2.10, it follows that the seed length of Ext is

$$b = \frac{\lambda \log^2(r/\epsilon)}{\log(k/m)} = \frac{\lambda \log^2(r/2^{-\delta_2\sqrt{r}})}{\log 2} = \lambda(\delta_2^2 r + \log^2 r + 2\delta_2\sqrt{r}\log r),$$

where λ is the constant from Theorem 2.10.

By our choice of δ_2, we have

$$\delta^2 r/40 \leq b = \lambda(\delta_2^2 r + \log^2 r + 2\delta_2\sqrt{r}\log r) \leq \delta^2 r/20.$$

Further, $\delta_1 = b/m$. This implies that $\delta_1 = b/m = b/(\delta r)$ satisfies $\delta/40 \leq \delta_1 \leq \delta/20$ as required in (6).

Next we claim that $M^{-\delta_1} < \epsilon < \delta/4$. Observe that this immediately implies (7) since

$$s^{1-\delta}/M = (MB)^{1-\delta}/M < B/M^{\delta} = M^{\delta_1-\delta} \leq 1/M^{19\delta/20} < 1/M^{\delta_1}.$$

Since $\epsilon = 2^{-\Omega(\sqrt{r})}$, it follows that $\epsilon < \delta/4$. Further, since $m = \Omega(r)$, it follows that $\epsilon > 2^{-\delta_1 m} = M^{-\delta_1}$.

We conclude by observing that by Lemma 2.15, we have Ext is a $\frac{1}{10}$-shift-design extractor. □

A.0.2 Useful Bounds for Construction 1

In this section, we assume that the parameters are picked in the way described in Section 5.2 and prove various bounds that complements arguments and proofs done in Section 5.

We first list some useful inequalities which are direct or almost directly implied by our choice of parameters and inequalities imposed on them in Section 5.2. We also list a few general inequalities that we frequently use in calculations. The claims in this section are routine calculations given the following list of inequalities.

$$0 < \delta, \epsilon_1 < 1/10 \tag{14}$$
$$\delta/40 \leq \delta_1 \leq \delta/20 \tag{15}$$
$$s^{1-\delta}/M < \epsilon < \delta/4 \tag{16}$$
$$m = \delta r,\ r = M^{o(1)} \tag{17}$$
$$b = \delta_1 m \tag{18}$$
$$(1 - B^{-\epsilon_1})\gamma \leq p_1 \leq \gamma \tag{19}$$
$$b = \Omega(r). \text{ Thus, for any constant } \nu > 0, B^\nu > r \tag{20}$$
$$\gamma = \frac{\ln M - \ln\ln(R/\ln 2)}{B},\ e^{-\gamma B} = \frac{\ln R - \ln\ln 2}{M} \tag{21}$$
$$\gamma < \frac{\ln M}{B} < r/B \tag{22}$$
$$\text{For any positive } \theta \leq 1, 2^\theta \leq 1 + \theta \tag{23}$$
$$\text{For any positive } \theta \leq 1, e^\theta \leq 1 + 2\theta \tag{24}$$
$$\text{For any } n > 1 \text{ and } 0 \leq x \leq n, \text{ we have } e^{-x}\left(1 - \frac{x^2}{n}\right) \leq \left(1 - \frac{x}{n}\right)^n \leq e^{-x} \tag{25}$$

Claim A.2. *The following inequalities hold: Let $\epsilon_2 = \epsilon_1/2$. Then,*

1. $$p_2 \leq \frac{r}{M}$$

2. $$\frac{1}{2R} \leq \left(\frac{\ln 2}{R}\right)\left(1 - \frac{2r}{B^{\epsilon_2}}\right) \leq p_3 \leq \left(\frac{\ln 2}{R}\right)\left(1 + \frac{r}{B^{\epsilon_2}}\right) \leq \frac{0.9}{R}$$

Proof. We in fact prove the following:

1. $$\frac{\ln R - \ln\ln 2}{M}\left(1 - \frac{1}{B^{\epsilon_2}}\right) \leq p_2 \leq \frac{\ln R - \ln\ln 2}{M}\left(1 + \frac{1}{B^{\epsilon_2}}\right) \leq \frac{r}{M}$$

2. $$\frac{1}{2R} \leq \left(\frac{\ln 2}{R}\right)\left(1 - \frac{2r}{B^{\epsilon_2}}\right) \leq p_3 \leq \left(\frac{\ln 2}{R}\right)\left(1 + \frac{r}{B^{\epsilon_2}}\right) \leq \frac{0.9}{R}$$

We start out by proving bounds on p_2. We have,

$$\begin{aligned}
p_2 &= (1-p_1)^B \\
&\geq (1-\gamma)^B && \text{(using (19))} \\
&\geq e^{-\gamma B}(1-\gamma^2 B) && \text{(by (25))} \\
&= \frac{\ln R - \ln\ln 2}{M}(1-\gamma^2 B) && \text{(using (21))} \\
&\geq \frac{\ln R - \ln\ln 2}{M}\left(1-\frac{r^2}{B}\right) && \text{(using (22))}
\end{aligned}$$

We now upper bound p_2. We have,

$$\begin{aligned}
p_2 &= (1-p_1)^B \\
&\leq (1-\gamma(1-B^{-\epsilon_1}))^B && \text{(using (19))} \\
&\leq e^{-\gamma B(1-B^{-\epsilon_1})} && \text{(by (25))} \\
&= \left(\frac{\ln R - \ln\ln 2}{M}\right)^{1-B^{-\epsilon_1}} && \text{(using (21))} \\
&< \left(\frac{\ln R - \ln\ln 2}{M}\right) M^{B^{-\epsilon_1}} \\
&< \left(\frac{\ln R - \ln\ln 2}{M}\right) e^{\delta r B^{-\epsilon_1}} && \text{(using (17))} \\
&\leq \frac{\ln R - \ln\ln 2}{M}\left(1+\frac{r}{B^{\epsilon_1}}\right) && \text{(using (24) and } 2\delta < 1)
\end{aligned}$$

Thus,

$$\frac{\ln R - \ln\ln 2}{M}\left(1-\frac{1}{B^{\epsilon_2}}\right) \leq p_2 \leq \frac{\ln R - \ln\ln 2}{M}\left(1+\frac{1}{B^{\epsilon_2}}\right),$$

using $\epsilon_2 = \epsilon_1/2$ and (20).

Further, since $\ln R = r \cdot \ln 2 < 0.9r$ and $\left(1+\frac{1}{B^{\epsilon_2}}\right) < 1.01$, it follows that

$$\frac{\ln R - \ln\ln 2}{M}\left(1+\frac{1}{B^{\epsilon_2}}\right) < r/M.$$

We now proceed to establish bounds on p_3. We have,

$$
\begin{aligned}
p_3 &= (1-p_2)^M \\
&\geq \left(1-\left(\frac{\ln R-\ln\ln 2}{M}\right)\left(1+\frac{1}{B^{\epsilon_2}}\right)\right)^M && \text{(using the upper bound established on } p_2\text{)} \\
&\geq \left(1-\frac{(\ln R-\ln\ln 2)^2}{M}\left(1+\frac{1}{B^{\epsilon_2}}\right)^2\right)\left(\frac{\ln 2}{R}\right)e^{\frac{-(\ln R-\ln\ln 2)}{B^{\epsilon_2}}} && \text{(by (25))} \\
&> \left(1-\frac{(\ln R-\ln\ln 2)^2}{M}\left(1+\frac{1}{B^{\epsilon_2}}\right)^2\right)\left(\frac{\ln 2}{R}\right)e^{-r/B^{\epsilon_2}} && \text{(since } e^r > R\text{)} \\
&> \left(1-\frac{2r^2}{M}\right)\left(\frac{\ln 2}{R}\right)e^{-r/B^{\epsilon_2}} \\
&> \left(1-\frac{2r^2}{M}\right)\left(\frac{\ln 2}{R}\right)\left(1-\frac{r}{B^{\epsilon_2}}\right) && \text{(since for any } \theta\in\mathbb{R},\ e^\theta \geq 1+\theta\text{)} \\
&\geq \left(1-\frac{r}{B}\right)\left(\frac{\ln 2}{R}\right)\left(1-\frac{r}{B^{\epsilon_2}}\right) && \text{(since } M > 2rB \text{ using (17) and (18))} \\
&\geq \left(\frac{\ln 2}{R}\right)\left(1-\frac{2r}{B^{\epsilon_2}}\right) && \text{(using (20))}
\end{aligned}
$$

We now prove the upper bound on p_3.

$$
\begin{aligned}
p_3 &\leq \left(1-\left(\frac{\ln R-\ln\ln 2}{M}\right)\left(1-\frac{1}{B^{\epsilon_2}}\right)\right)^M && \text{(using the lower bound established on } p_2\text{)} \\
&\leq \left(\frac{\ln 2}{R}\right)^{1-B^{-\epsilon_2}} && \text{(by (25))} \\
&< \left(\frac{\ln 2}{R}\right)R^{B^{-\epsilon_2}} \\
&= \left(\frac{\ln 2}{R}\right)2^{r\cdot B^{-\epsilon_2}} \\
&\leq \left(\frac{\ln 2}{R}\right)\left(1+\frac{r}{B^{\epsilon_2}}\right) && \text{(using (23), since by (20), } \frac{r}{B^{\epsilon_2}} < 1\text{).}
\end{aligned}
$$

Thus,

$$\left(\frac{\ln 2}{R}\right)\left(1-\frac{2r}{B^{\epsilon_2}}\right) \leq p_3 \leq \left(\frac{\ln 2}{R}\right)^{1-\frac{r}{B}} \leq \left(\frac{\ln 2}{R}\right)\left(1+\frac{r}{B^{\epsilon_2}}\right).$$

Using (20), it follows that

$$\frac{1}{2R} \leq \left(\frac{\ln 2}{R}\right)\left(1-\frac{2r}{B^{\epsilon_2}}\right) \leq p_3 \leq \left(\frac{\ln 2}{R}\right)\left(1+\frac{r}{B^{\epsilon_2}}\right) \leq \frac{0.9}{R}.$$

□

Claim A.3. $(1-p_2)^{M-q} \leq \frac{1}{R}$.

Proof. We have,

$$\begin{aligned}
(1-p_2)^{M-q} &\le p_3^{1-\frac{s^{1-\delta}}{M}} \\
&\le p_3(2R)^{\frac{s^{1-\delta}}{M}} && \text{(since } p_3 > \frac{1}{2R} \text{ by Claim A.2)} \\
&\le p_3 e^{r/M^{\delta/2}} && \text{(since } s = M^{1+\delta_1} < M^{1+\frac{\delta}{2}}/2 \text{ using (15))} \\
&\le p_3(1+\frac{2r}{M^{\delta/2}}) && \text{(using } e^{\theta} \le 1+2\theta, \text{ if } \theta \in [0,1]) \\
&< \frac{0.9}{R}(1+\frac{2r}{M^{\delta/2}}) && \text{(by Claim A.2)} \\
&< 1/R && \text{(using (17))}
\end{aligned}$$

□

Claim A.4. $q(1-p_1)^{B(1-2\epsilon)} < \frac{q}{2s^{1-\delta}}$.

Proof. We have,

$$\begin{aligned}
q(1-p_1)^{B(1-2\epsilon)} &= qp_2^{1-2\epsilon} \\
&\le \frac{qr}{M^{1-2\epsilon}} && \text{(since } p_2 < r/M \text{ by Claim A.2)} \\
&= \frac{qr}{M^{1-\frac{\delta}{2}}} && \text{(since } \epsilon < \delta/4 \text{ using (16))} \\
&< \frac{q}{M^{1-\frac{2\delta}{3}}} && \text{(using (17))} \\
&< \frac{q}{2s^{1-\delta}} && \text{(since } s = M^{1+\delta_1} < M^{1+\frac{\delta}{4}} \text{ using (15)).}
\end{aligned}$$

□

We record a simple claim that is direct from Claim A.2. Recall that $\alpha = p_3 R$ (see Section 5).

Claim A.5. *There exists a small constant* $\epsilon_3 > 0$ *such that* $1 - \frac{1}{B^{\epsilon_3}} \le \frac{\alpha}{\ln 2} \le 1 + \frac{1}{B^{\epsilon_3}}$.

Claim A.6. $\left|e^{-\alpha} - \frac{1}{2}\right| \le \frac{1}{2B^{\epsilon_3}}$.

Proof. Recall that from Claim A.5, we have

$$\ln 2\left(1 - \frac{1}{B^{\epsilon_3}}\right) \le \alpha \le \ln 2\left(1 + \frac{1}{B^{\epsilon_3}}\right).$$

Using this, we have

$$\begin{aligned}
\left|e^{-\alpha} - \frac{1}{2}\right| &\le \max\{2^{\theta-1} - \frac{1}{2}, \frac{1}{2} - 2^{-(\theta+1)}\} \\
&= \max\{(2^{\theta}-1)/2, (1-2^{-\theta})/2\} \\
&= (2^{\theta}-1)/2 \le \theta/2.
\end{aligned}$$

where the final inequality uses the fact that $2^{\eta} \le 1+\eta$ for any positive $\eta \le 1$.

Thus,

$$\left|e^{-\alpha} - \frac{1}{2}\right| \le \frac{1}{2B^{\epsilon_3}}.$$

□

Claim A.7. $\frac{s^{1+2\beta_1}r^2}{M^{\frac{11}{10}}} < \frac{1}{M^{\frac{1}{20}-3\beta_1}}$, *where* β_1 *is the constant from Lemma 5.12.*

Proof. We have,

$$\begin{aligned}
\frac{s^{1+2\beta_1}r^2}{M^{\frac{11}{10}}} &= \frac{(MB)^{1+2\beta_1}r^2}{M^{\frac{11}{10}}} && \text{(using } s = MB\text{)} \\
&= \frac{B^{1+2\beta_1}r^2}{M^{\frac{1}{10}-2\beta_1}} \\
&= \frac{M^{\delta_1(1+2\beta_1)}r^2}{M^{\frac{1}{10}-2\beta_1}} && \text{(using } B = M^{\delta_1} \text{ by (18))} \\
&< \frac{1}{M^{\frac{1}{20}-3\beta_1}} && \text{(using that by (15), } \delta_1 < 1/20\text{; and (17))}
\end{aligned}$$

□

Claim A.8. *Let* $a = s^{\beta_3}$ *be the parameter used in the proof of Lemma 5.14. For any* $c \le a$, $\left|S_c - \frac{\alpha^c}{c!}\right| \le \frac{1}{M^{\beta_2}}$.

Proof. Recall from the proof of Lemma 5.14 that $\beta_3 = \min\{\beta_1/2, \beta_2/2\}$, where β_1, β_2 are constants defined in Lemma 5.12. We have,

$$\begin{aligned}
\binom{R}{c}\left(\frac{\alpha}{R}\right)^c &\le \frac{R^c}{c!}\frac{\alpha^c}{R^c} \\
&= \frac{\alpha^c}{c!}
\end{aligned}$$

and

$$\begin{aligned}
\binom{R}{c}\left(\frac{\alpha}{R}\right)^c &= \frac{R(R-1)\dots(R-c+1)}{R^c}\frac{\alpha^c}{c!} \\
&\ge \left(1 - \frac{(c-1)^2}{R}\right)\frac{\alpha^c}{c!} && \text{(using } (1-\alpha)(1-\beta) \ge 1-\alpha-\beta\text{, if } \alpha\beta \ge 0\text{)} \\
&\ge \left(1 - \frac{a^2}{R}\right)\frac{\alpha^c}{c!} && \text{(since } c \le a+1\text{)} \\
&\ge \left(1 - \frac{1}{R^{1-\beta_2}}\right)\frac{\alpha^c}{c!}
\end{aligned}$$

by our choice of a.

Thus, for any $c \le a$, we have

$$\begin{aligned}
\left|S_c - \frac{\alpha^c}{c!}\right| &\le \frac{\alpha^c}{c!}\cdot\left(\frac{1}{M^{\beta_2}} + \frac{1}{R^{1-\beta_2}}\right) \\
&\le \frac{1}{2}\left(\frac{1}{M^{\beta_2}} + \frac{1}{R^{1-\beta_2}}\right) && \text{(since } \alpha < 2\text{)} \\
&\le \frac{1}{M^{\beta_2}},
\end{aligned}$$

where the last inequality uses the fact that $R = M^{1/\delta}$ (using (17)), and hence $R^{1-\beta_2} > M^{\beta_2}$ using the inequality $(1-\beta_2)/\beta_2 > 1 > \delta$. □

A.0.3 A Bound for Construction 2

We provide the proof of a bound used in arguing the correctness of Construction 2. The inequalities listed in the previous section continue to hold, and we use them in proving the following lemma.

Claim A.9. *We assume the setup of parameters as described in Construction 2. Then,* $s^{1-\frac{\delta}{2}} \geq n^{1-\delta}$.

Proof. We have

$$\begin{aligned}
s^{1-\frac{\delta}{2}} &= (MB)^{1-\frac{\delta}{2}} \\
&> (MB)^{\left(1+\frac{\delta}{2}\right)(1-\delta)} \\
&> (MB^3)^{1-\delta} && \text{(since } M^{\delta/2} > M^{2\delta_1} = B^2 \text{ by (15), (18))} \\
&\geq (MBh)^{1-\delta} = n^{1-\delta} && \text{(using } h \leq B^{1+\frac{\delta}{2}} \text{ and } n = MBh)
\end{aligned}$$

□

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il