2016 IEEE 57th Annual Symposium on Foundations of Computer Science

Extractors for Near Logarithmic Min-Entropy

Gil Cohen, Leonard J. Schulman
Computing and Mathematical Sciences Department
California Institute of Technology
Pasadena, CA, USA
Email: coheng@caltech.edu

***Abstract*—The main contribution of this work is an explicit construction of extractors for near logarithmic min-entropy. For any $\delta > 0$ we construct an extractor for $O(1/\delta)$ n-bit sources with min-entropy $(\log n)^{1+\delta}$. This is most interesting when δ is set to a small constant, though the result also yields an extractor for $O(\log\log n)$ sources with logarithmic min-entropy.**

Prior to this work, the best explicit extractor in terms of supporting least-possible min-entropy, due to Li (FOCS'15), requires min-entropy $(\log n)^{2+\delta}$ from its $O(1/\delta)$ sources. Further, all current techniques for constructing multi-source extractors "break" below min-entropy $(\log n)^2$. In fact, existing techniques do not provide even a disperser for $o(\log n)$ sources each with min-entropy $(\log n)^{1.99}$.

Apart from being a natural problem, supporting logarithmic min-entropy has applications to combinatorics. A two-source disperser, let alone an extractor, for min-entropy $O(\log n)$ induces a $(\mathrm{polylog}\, n)$-Ramsey graph on n vertices. Thus, constructing such dispersers would be a significant step towards constructively matching Erdős' proof for the existence of $(2\log n)$-Ramsey graphs on n vertices.

Our construction does not rely on the sophisticated primitives that were key to the substantial recent progress on multi-source extractors, such as non-malleable extractors, correlation breakers, the lightest-bin condenser, or extractors for non-oblivious bit-fixing sources, although some of these primitives can be combined with our construction so to improve the output length and the error guarantee. Instead, at the heart of our construction is a new primitive called an *independence-preserving merger*. The construction of the latter builds on the alternating extraction technique.

I. Introduction

A *randomness extractor* is a function that produces truly random bits given a sample from a source that is "somewhat random". The standard measure for the amount of randomness in a source is its *min-entropy* which, up to a logarithmic scaling, is the probability to sample the most likely element to be sampled by the source.

Ideally, one would have liked to define a randomness extractor as a function $\mathsf{Ext}\colon \{0,1\}^n \to \{0,1\}^m$ such that for any n-bit random variable X with min-entropy k, $\mathsf{Ext}(X)$ is a close to uniform in statistical distance. Unfortunately, such a function does not exist even if one is satisfied with outputting a single bit, that has a constant bias, given a sample from a source with min-entropy as high as $n-1$.

One approach [CG88] to circumvent this negative result is to feed the extractor with more than one sample. A *multi-source extractor* is a function $\mathsf{Ext}\colon \left(\{0,1\}^n\right)^s \to \{0,1\}^m$ with the guarantee that if $X_1,\ldots,X_s$ are independent random variables, each having min-entropy k, then $\mathsf{Ext}(X_1,\ldots,X_s)$ is close to uniform. A simple probabilistic argument can be used to show that there exists a multi-source extractor already for $s=2$ sources. The min-entropy k that such a two-source extractor can support is $k=\log(n)+O(1)$. Further, one can output $m=k-O(1)$ bits, where in both instances, the $O(1)$ term depends solely on the statistical distance of the output to the uniform distribution.

Although this existential result is of interest, explicit constructions are far more desirable. As it turns out, the most challenging aspect of constructing multi-source extractors is to support low min-entropy. Indeed, even after an extensive research effort that spanned over 25 years, it was not until the recent work by Li [Li13a] that multi-source extractors with a constant number of sources could support poly-logarithmic min-entropy. This held even if one only wished to obtain a single output bit with a constant bias.

A. Applications to Ramsey theory

Apart from proving to be the most difficult aspect of constructing multi-source extractors, the problem of supporting low min-entropy, even when considering one output bit with constant bias, is of interest due to its applications to Ramsey theory. Recall that a graph on N vertices is called *K-Ramsey* if it contains no clique or independent set of size K. Ramsey [Ram28] proved that there does not exist a graph on N vertices that is $0.5\log N$-Ramsey. This result was later complemented by Erdős [Erd47], who proved that most graphs on N vertices are $(2+o(1))\log N$-Ramsey.

Unfortunately, Erdős' argument is non-constructive, and one does not obtain from Erdős' proof an example of a graph that is $(2+o(1))\log N$-Ramsey. A central problem in combinatorics is to match Erdős' proof, up to any multiplicative constant factor, with a constructive proof. That is, to come up with an explicit construction of an $O(\log N)$-Ramsey graph on N vertices.

Erdős' challenge gained significant attention in the literature, and the current best known constructions [Coh15c], [CZ15] achieve $K = 2^{\mathrm{poly}\log\log N}$, which is quasi-polynomially close to meeting Erdős' challenge. Both constructions, and also their predecessors [BKS$^+$05], [BRSW12], rely on the equivalence between *two-source dispersers* and *bipartite* Ramsey graph.

A two-source disperser for min-entropy k is a function $\mathsf{Disp}\colon \left(\{0,1\}^n\right)^2 \to \{0,1\}$ with the property that for any

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.27

two independent random variables X_1, X_2, each having min-entropy k, the output $\mathsf{Disp}(X_1, X_2)$ is non-constant. Note that a two-source extractor with a constant bias is in particular a two-source disperser. In fact, a two-source disperser can be thought of as a two-source extractor with any non-trivial guarantee on the bias. On the other hand, it is straightforward to show that a two-source disperser such as Disp above yields a $K = 2^k$-bipartite Ramsey graph on $N = 2^n$ vertices on each side, where a K-bipartite Ramsey graph is the natural analog of Ramsey graphs for bipartite graphs. Further, one can show that a bipartite Ramsey graph yields a Ramsey graph with the same parameters.

By this connection between Ramsey graphs and dispersers, it is evident that a two-source disperser (let alone a two-source extractor) for min-entropy $c \cdot \log n$ would immediately induce a $(\log N)^c$-Ramsey graph on N vertices, which is polynomially-close to optimal if c is constant. We remark that in order to resolve Erdős' challenge for bipartite graphs, one would have to construct a two-source disperser for min-entropy $\log(n) + O(1)$ which seems to be an extremely difficult task.

B. Where do existing techniques break?

Up until the recent work by Li [Li13a], all extractors for a constant number of sources could only support min-entropy $n^{\Omega(1)}$ [Rao09], [Li13b]. In [Li13a], and in a subsequent work [Li15b], Li significantly improved known results by constructing, for any constant $\delta > 0$, an extractor for $\lceil 14/\delta \rceil + 2$ sources with min-entropy $(\log n)^{2+\delta}$. That is, by using Li's extractor, one can support min-entropy that approaches arbitrarily close to $(\log n)^2$ – quadratically close to optimal, by consuming a large enough number of sources.

Based on ideas from [Li13a], [Li15b], subsequent works considered the problem of optimizing the number of sources while supporting min-entropy $(\log n)^c$, though possibly with a large exponent c. This includes constructions of three-source extractors [Li15b], two-source dispersers [Coh15c], and subsequently also two-source extractors [CZ15], [Li15a], [Mek15]. All of these constructions require exponents $c \gg 2$.

By inspection, all of the exciting techniques that were used for the construction of multi-source extractors seem to break below min-entropy $(\log n)^2$. When insisting on two sources, the situation is even worse in term of supported min-entropy as current constructions resort to structural results regarding the extent to which bounded independence fools certain types of circuits [Bra10], [Tal14], [KLW10] making these results costly in terms of min-entropy.

When considering two-source dispersers, current techniques require min-entropy at least $(\log n)^3$. Indeed, besides using a certain type of a three-source extractor [Li15b], for which we currently need high min-entropy, the construction by [Coh15c] is based on locating a nicely structured source with min-entropy $k/(\log n)^3$ inside each of the two min-entropy k sources on which the disperser operates. This approach only makes sense for $k > (\log n)^3$ even if one has access to an optimal three-source extractor.

This $(\log n)^2$ "barrier" has held also when considering a super-constant number of sources. In fact, to the best of our knowledge, using existing methods, it was not known how to obtain a disperser for $o(\log n)$ sources with min-entropy $(\log n)^{1.99}$, let alone an extractor with a constant number of sources for such low min-entropy, which is what we are set to obtain.

C. Our contribution

The main contribution of this work is an explicit construction of multi-source extractors for near logarithmic min-entropy. More precisely, we prove the following.

Theorem I.1. *There exists a universal constant c such that the following holds. For any integer n and any* $\delta > 0$ *(that may depend on n), there exists an efficiently-computable extractor* $\mathsf{Ext}\colon (\{0,1\}^n)^b \to \{0,1\}$ *for* $b = 2/\delta + c$ *sources, each with min-entropy* $(\log n)^{1+\delta}$*, having output with bias* 0.01.

Theorem I.1 is most interesting when one takes δ to be a small constant as this keeps the number of sources constant while supporting close to optimal min-entropy. However, we stress that the parameter δ in Theorem I.1 can be an arbitrary function of n. In particular, by setting $\delta = (\log\log n)^{-1}$, Theorem I.1 yields an explicit multi-source extractor for $2\log\log n + O(1)$ sources with min-entropy $O(\log n)$. More generally, Theorem I.1 gives an explicit multi-source extractor for min-entropy $(\log n)^{1+o(1)}$ with a corresponding $\omega(1)$ number of sources.

It is also worth noting that besides supporting lower min-entropy, the number of sources required by our extractor is smaller than that required by [Li15b] in the most interesting range of parameters, namely, as δ approaches to zero.

Somewhat surprisingly, our extractor do not rely on any of the primitives that were developed and used by recent constructions of multi-source extractors, such as non-malleable extractors [DW09], [DLWZ14], [CRS14], [Li12a], [Li12b], [CGL15], [Coh15b], correlation breakers [Coh15a], [CGL15], the lightest-bin condenser [Li13a], [Li15b], or extractors for non-oblivious bit-fixing sources [AL93], [Vio14], [CZ15], [Mek15]. We only rely on the alternating extraction technique [DP07] and components that are by now considered standard, and which were available for close to a decade. These includes seeded extractors and condensers [GUV09], Raz's seeded extractor with weak-seeds [Raz05], Bourgain's two-source extractor [Bou05], error correcting codes, and expander graphs.

Although our construction does not yield improved Ramsey graphs, as it requires more than two sources, we believe it is a step towards such a construction. Our source of optimism is based on inspecting the research path that led to the construction of two-source extractors and dispersers for poly-logarithmic min-entropy. Indeed, it is evident that many of the ideas and objects that were used in such constructs were gradually developed in the context of multi-source extractors. More concretely, at the heart of our construction is a new primitive called an *independence-preserving merger*, which we hope will be of value in future constructions.

1) Improving the output length and the error guarantee: Given that one aims to optimize the supported min-entropy, and especially when having the application to Ramsey theory in mind, it is somewhat less pressing to output many bits or to guarantee a sub-constant error. Nevertheless, outputting many bits with a better error guarantee is a natural goal and, typically, extractors that have many output bits with low error guarantee allow for compositions with other primitives.

By using the primitives developed for the proof of Theorem I.1, together with several results from the literature, such as the condenser of Li [Li13a] that is based on the lightest-bin protocol [Fei99], and using mergers with weak-seeds [Coh15a], we can guarantee a low error and output many bits.

Theorem I.2. *For any integer* n *and any constant* $\delta > 0$*, there exists an efficiently-computable extractor* $\mathsf{Ext}\colon (\{0,1\}^n)^b \to \{0,1\}^m$ *for* $b = 16/\delta + O(1)$ *sources, each with min-entropy* $(\log n)^{1+\delta}$*, having error guarantee* $2^{-\Omega((\log n)^{\delta/4})}$ *and* $m = \Omega((\log n)^{1+\delta})$ *output bits.*

II. Proof Outline

Due to lack of space we do not give any formal proof in this version of the paper and refer the reader to its full version. Instead, in the sequel we present the outline of our proofs.

In this section we present the construction and outline the analysis of our extractor that is given by Theorem I.1. The main effort taken by our extractor is the efficient transformation of the sources it operates upon into a sequence of $\{0,1\}$ random variables $X_1,\ldots,X_r$, with $r = \mathrm{poly}(n)$, such that all but $r^{1/2-\alpha}$ of the random variables in the sequence are "good". By good, we mean that for some parameter t to be chosen later on, the joint distribution of every t-tuple of good variables is close to uniform. The parameter α is some small universal constant that is strictly larger than zero.

One can easily show that if all the good X_i's were jointly uniform then the majority function applied to the X_i's would have bias $O(r^{-\alpha})$. However, one cannot obtain such a strong independence, namely $t = \Omega(r)$, given few low min-entropy sources. Indeed, it is known [AGM03] that a t-wise independent distribution over r-bits has min-entropy $\Omega(t \log r)$, and the min-entropy has to come from the sources. Luckily, a result by Diakonikolas *et al.* [DGJ$^+$10] regarding the extent to which t-wise independence fools threshold functions, can be applied to show that as the good X_i's are t-wise independent, the bias of $\mathsf{Maj}(X_1,\ldots,X_r)$ is bounded by $\widetilde{O}(1/\sqrt{t}) + O(r^{-\alpha})$. Hence, by setting $t = \widetilde{O}(1/\varepsilon^2)$, one obtains an extractor with bias ε as the second summand is negligible when ε is a small constant, or slightly sub-constant in r.

This general scheme was suggest by Viola in the context of extractors for certain structured sources [Vio14], and a variant of the latter was adopted by Chattopadhyay and Zuckerman [CZ15] for their breakthrough construction of two-source extractors. We find it beneficial to contrast the [CZ15] strategy and that of Viola, which we adopt here. In [CZ15], the authors transformed two sources into a sequence of $\{0,1\}$ random variables $X_1,\ldots,X_r$, with $r = \mathrm{poly}(n)$, such that all but $r^{1-\beta}$ of the random variables are good, where $\beta > 0$ is some small constant. The notion of "good" is similar to ours, though with a much larger $t = (\log n)^c$. Here c is some large enough constant. In particular, by using an improvement of the original analysis of [CZ15] due to Meka [Mek15], one can set $c = 2$. At any rate, the value of t in [CZ15], [Mek15] is a function of n whereas our setting of t depends solely on the bias of the output which, for the proof of Theorem I.1, we think of as a small constant.

At this point, a so-called non-oblivious bit-fixing extractor is applied by [CZ15] to the X_i's. The reader does not need to worry about what that is exactly. By some further properties of this extractor, it is possible to show that the output has bias $1/\mathrm{poly}(n)$. Ingeniously, the t-wise independence enables the use of Braverman's result [Bra10], [Tal14] in a critical point of the analysis of [CZ15].

Unfortunately, as mentioned, generating a sequence of $\mathrm{poly}(n)$ bits that are $(\log n)^c$-wise independent requires min-entropy $(\log n)^{c+1}$ [AGM03]. Moreover, by inspection, the techniques that were used to generate the X_i's require min-entropy $(\log n)^2$ even for obtaining pairwise independence (namely, $t = 2$) across the good variables. Indeed, [CZ15] applies a non-malleable extractor by [CGL15] that has seed length $(\log n)^2$ and can only support min-entropy larger than $(\log n)^2$. The min-entropy requirement from the two sources on which the extractor of [CZ15] operates is induced directly by the seed length and min-entropy requirement of the non-malleable extractor. Even by using an improved construction of non-malleable extractors [Coh15b], which has seed length $O(\log n \cdot \log\log n)$ and supports min-entropy $O(\log n)$, one of the sources is required to have min-entropy $(\log n)^2$ due to the dependence of the seed in the error guarantee. Again, this already holds for $t = 2$, which is anyhow insufficient for [CZ15], [Li15a], [Mek15].

Our choice of the majority function, as opposed to the explicit non-oblivious bit-fixing extractors that were developed and used by [CZ15], [Mek15], is natural as we only need to produce a sequence of X_i's where the good variables in the sequence are t-wise independent, where t is decoupled from n, and is a function only of the desired bias of the output. We point out that, computational aspects aside, with some work one can show that for a constant bias it is possible to generate such a sequence using only two sources with logarithmic min-entropy. Unfortunately, as discussed above, all current techniques require min-entropy $(\log n)^2$ even for obtaining pairwise independence across the good X_i's.

Our strategy for generating a sequence of X_i's with the above mentioned property can be divided into three steps:

a) Step 1 – Ensuring that there are very few bad random variables.: By consuming a constant number of sources, we generate a sequence of ℓ-bit random variables $\{Y_i\}_{i=1}^r$ such that all but $r^{1/2-\alpha}$ of the variables are close to uniform, where $\alpha > 0$ is some small universal constant. Any $g \in [r]$ for which

Y_g is close to uniform is said to be *good*. Note that there is no guarantee on the correlations (or the lack there of) between the Y_i's. One should think of $\ell = O(\log n)$ and $r = \mathrm{poly}(n)$.

b) Step 2 – Obtaining somewhere-independent matrices.: Using a constant number of fresh sources, we transform each Y_i to a random variable in the form of a $(t \log n) \times \ell$ binary matrix M_i. The guarantee is that for any good $g \in [r]$ and for any $i_1, \dots, i_t \in [r] \setminus \{g\}$, there is some row in M_g that is close to uniform even conditioned on the joint distribution of the corresponding row of the matrices $M_{i_1}, \dots, M_{i_t}$. So, informally speaking, the matrix M_g is *somewhere independent* of $M_{i_1}, \dots, M_{i_t}$. In fact, this property holds for 0.9 fraction of the rows of every good matrix. Further, all rows of M_g, for a good g, are close to uniform (although possibly correlated amongst themselves and with rows of other matrices in the sequence).

c) Step 3 – Merging while preserving independence.: In the last step we consume $2/\delta + O(1)$ sources so to merge the rows of each M_i to a single bit, while preserving independence. That is, we construct what we call an *independence-preserving merger* which, given a matrix, outputs a bit with the property that when applied to somewhere-independent matrices, such as M_g, $M_{i_1}, \dots, M_{i_t}$ above, the merged bit of M_g is close to uniform even conditioned on the joint distribution of the other t merged bits. Of course, these merged bits will be our X_i's, to which we will eventually apply the majority function.

Most of the technical effort and novelty of this work is in implementing the third step, namely, in the construction of independence-preserving mergers. Though, a fair amount of work is also required for accomplishing the first two steps. In the next section we describe the ideas that go into each step.

III. A More Detailed Proof Outline

In the following sections we elaborate on each of the three steps of our construction of multi-source extractors.

A. Step 1 – Ensuring that there are very few bad random variables

In order to apply the majority function to r random variables in the presence of bad variables and obtain a low biased output bit, it is necessary that the number of bad variables is sufficiently smaller than $\sqrt{r}$. As mentioned, by the work of [DGJ+10], such a bound on the number of bad variables is sufficient for the same argument to hold even if the good variables are only guaranteed to be t-wise independent, where the larger one takes t, the smaller the bias of the output bit will be. The goal of the first step of our construction is to achieve this bound on the number of bad variables without worrying about independence across the good random variables.

Initiated in [Rao09], by now the standard method for transforming a weak-source into a sequence of random variables, most of which are uniform, is based on strong seeded extractors. Let $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$ be a strong seeded extractor with error guarantee ε. Let W be a weak-source with sufficient min-entropy as required by Ext. Set $r = 2^d$. We identify $\{0,1\}^d$ with $[r]$, and define for $i \in [r]$ the i'th random variable in the sequence by $X_i = \mathsf{Ext}(W, i)$. By the properties of strong seeded extractors, all but $\sqrt{\varepsilon}$ fraction of the X_i's are $\sqrt{\varepsilon}$-close to uniform. Lets round things up and assume that all but ε fraction of the variables are truly uniform. Namely, at most εr of the r variables are bad. Unfortunately for us, the seed length of a seeded extractor is provably always larger than $2\log(1/\varepsilon)$ [RTS00], and so the number of bad variables $\varepsilon r > \sqrt{r}$. That is, by using a strong seeded extractor this way, the number of bad variables is always larger than $\sqrt{r}$. In previous works this was never an issue. We, however, require that the number of bad variables would be very small in comparison to the size of the sequence.

Our solution to this problem is simple – we make use of seeded *condensers* rather than seeded extractors, as the former have a seed length with better dependence in the error guarantee. A seeded condenser is an efficiently-computable function $\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with the following property. For any (n,k)-source X and an independent random variable S that is uniform over $\{0,1\}^d$, it holds that $\mathsf{Cond}(X,S)$ is ε-close to having min-entropy at least k'. Note that an extractor is a special case of a condenser obtained by setting $k' = m$. If $k' = k + d$ we say that Cond is a *lossless condenser*.

Computational aspects aside, for seeded condensers, the dependence of the seed length in the desired error guarantee ε is only $\log(1/\varepsilon)$ as apposed to $2\log(1/\varepsilon)$. This makes all the difference. Luckily, explicit constructions come very close to the existential result in this respect. We use the lossless condenser by Guruswami *et al.* [GUV09]. Roughly speaking, for any $\tau > 0$ (which can also by taken to be larger than 1), Cond can be set to have a seed of length $d \approx (1 + 1/\tau)\log(n/\varepsilon)$ and $m \approx (1+\tau)k$ output bits.

One can show that for any $\delta > 0$, except with probability δ over $s \sim S$ it holds that $\mathsf{Cond}(X,s)$ is (ε/δ)-close to a $((1+\tau)k, k)$-source. A simple calculation then shows that if one aims for $\delta r < r^{1/2-\alpha}$ for some desired constant α, then one needs to take $\tau = 1 + O(\alpha)$, and by choosing ε, δ appropriately one get that all but $r^{1/2-\alpha}$ of the variables are $r^{-\Omega(\alpha)}$-close to having min-entropy rate $1/2 - O(\alpha)$.

We, however, want the good variables to be close to uniform and not just close to having min-entropy rate $1/2 - O(\alpha)$. To this end, we make use of Bourgain's two-source extractor [Bou05] that supports min-entropy rate $1/2 - \beta$ for some small universal constant $\beta > 0$. We set our α accordingly. Luckily, Bourgain's extractor outputs a constant fraction of the min-entropy with an exponentially low error guarantee, which is crucial for us.

With Bourgain's extractor in hand, we take a second source Y and compute $\mathsf{Cond}(Y,i)$ for all $i \in [r]$. We then define the random variable $Z_i = \mathsf{Bour}(\mathsf{Cond}(X,i), \mathsf{Cond}(Y,i))$. For any i that is a good seed for both X and Y, with respect to Cond, we have that Z_i is $r^{-\Omega(\alpha)}$-close to uniform. Thus, a good variable in the sequence of the Z_i's is not only close to having high min-entropy but is in fact close to uniform. Further, the number of bad variables increases by a factor of at most two, which is negligible.

At this point all but $O(r^{1/2-\alpha})$ of the random variables in the sequence are $r^{-\Omega(\alpha)}$-close to uniform. For technical reasons, we need the good variables to be even closer to uniform. For what comes next, r^{-2} will do. In order to reduce the statistical distance, we apply the procedure above to c/α pairs of independent sources $\{(X^j, Y^j)\}_{j=1}^{c/\alpha}$, for some large enough constant c, and take the bitwise XOR of the results. That is, the i'th random variable in the generated sequence $\{W_i\}_{i=1}^{r}$ is given by $W_i = \bigoplus_{j=1}^{c/\alpha} \mathsf{Bour}(\mathsf{Cond}(X^j, i), \mathsf{Cond}(Y^j, i))$. One can show that for any $i \in [r]$ that is a good seed for all sources in $\{(X^j, Y^j)\}_{j=1}^{c/\alpha}$, the random variable W_i is $(r^{-\Omega(\alpha)})^{c/\alpha}$-close to uniform. So, by setting c accordingly we can get the error guarantee below its desired bound.

B. Step 2 – Obtaining somewhere-independent matrices

At this point we are given the sequence of r random variables computed in Step 1, where all but $r^{1/2-\alpha}$ of the variables are good. Our goal now is to produce a sequence of matrices $M_1, \ldots, M_r$ such that for every good g, the matrix M_g is somewhere-independent of any t other matrices $M_{i_1}, \ldots, M_{i_t}$ in the sequence. As mentioned, we in fact need to guarantee that 0.9 fraction of the rows of M_g are close to uniform even conditioned on the corresponding row of the matrices $M_{i_1}, \ldots, M_{i_t}$, and that all rows of M_g are close to uniform.

In the following section we describe a fairly simple algorithm for solving this task. The downside of this solution is that it requires a number of sources that depends on ε – the bound on the bias of the output bit. This suffices if one is interested in some constant guarantee on the bias and is not too bothered with the number of sources consumed, as long as it is a constant independent of n. Nevertheless, one can do better. In Section III-B3 we give a solution that consumes only a single source. This solution, however, relies on *correlation breakers with advice* – a primitive that was introduced in the context of non-malleable extractors [CGL15], [Coh15b]. Unfortunately, current constructions of correlation breakers with advice are fairly involved, and so it is beneficial to also have the simpler, though source-wise more expensive solution, which we now present.

1) A simple yet source-wise expensive solution: For both the simple and the more involved implementations of Step 2, we make use of error correcting codes. For parameters q, m to be chosen later on, let $\mathsf{ECC}\colon \mathbb{F}_q^k \to \mathbb{F}_q^m$ be an error correcting code, where we identify $\mathbb{F}_q^k$ with $[r]$. Here $\mathbb{F}_q$ stands for the finite field with q elements. Note that $m \approx \log(r)/\rho$ with ρ being the rate of the code. We set the relative distance of the code to $\delta = 1 - 1/(10t)$.

We apply Step 1 not once but q times, each time with a fresh set of (a constant number of) sources, so to obtain q independent sequences which we denote by $\{X_i^1\}_{i=1}^{r}, \ldots, \{X_i^q\}_{i=1}^{r}$. For $i \in [r]$ and $j \in [m]$, we define the j'th row of the matrix M_i as $(M_i)_j = X_i^{\mathsf{ECC}(i)_j}$. In words, we use the j'th entry of the codeword that corresponds to the message i so to decide from which sequence to take the j'th row of M_i.

The analysis is straightforward and proceeds as follows. Fix $g \in [r]$ that is good for all q sequences, and consider any $i_1, \ldots, i_t \in [r] \setminus \{g\}$. By our choice of δ, for any fixed $c \in [t]$, the codewords $\mathsf{ECC}(g)$ and $\mathsf{ECC}(i_c)$ agree on at most $1/(10t)$ fraction of their entries. Thus, $\mathsf{ECC}(g)_j \notin \{\mathsf{ECC}(i_c)_j\}_{c=1}^{t}$ for at least 0.9 fraction of $j \in [m]$. For any such j, by the independence across the sequences generated in Step 1, $(M_g)_j$ is uniform and independent of the joint distribution of $\{(M_{i_c})_j\}_{c=1}^{t}$, as desired. Note that the number of bad variables increased by a multiplicative factor of q, though this loss is negligible.

2) What code should we use?: We briefly discuss the choice of the parameters m, q – the block-length and field size of ECC. Note that the number of sources consumed by the solution described in the previous section, grows linearly with q. Thus, it is important to work with a code that has a small alphabet size. In particular, we cannot use, say, Reed-Solomon codes as this would require us to consume $\Omega(\log n)$ sources. Moreover, we also want a code with high rate as the latter affects m – the number of rows of the generated matrices, which in turn puts restrictions on the min-entropy required from the sources used in Step 3.

As it turns out, the family of algebraic-geometric codes (also known as Goppa codes) is a suitable choice in our setting. These are codes that approach the Singleton bound using a strikingly small alphabet size. More precisely, with such codes one can obtain $\rho + \delta \geq 1 - \frac{1}{\sqrt{q}-1}$. Thus, with alphabet of size $q = O(t^2)$, the code can have the required relative distance $\delta = 1 - 1/(10t)$ and rate $\rho = \Omega(1/t)$. As we set $t = \widetilde{O}(\varepsilon^{-2})$, this translates to a solution that consumes $\widetilde{O}(\varepsilon^{-4})$ sources. The number of rows of the generated matrices is then $\widetilde{O}(\varepsilon^{-2}) \cdot \log n$.

We stress that algebraic-geometric codes have an extremely good dependence on the field size, from which we benefit. Indeed, even random codes, as used in the proof of the Gilbert-Varshamov bound, require alphabet size $q = 2^{\Omega(t)}$ for our choice of δ. This in turn would require us to consume $2^{\widetilde{O}(\varepsilon^{-2})}$ sources.

3) A solution that consumes a single source: In this section we describe a second implementation for Step 2 that has the advantage of consuming only a single source. To this end, we make use of a *t-correlation breaker with advice*. Roughly speaking, this is a function that breaks the correlations between random variables given an "advice" and using a fresh weak-source of randomness. More formally, a t-correlation breaker with advice is a function $\mathsf{AdvCB}\colon \{0,1\}^\ell \times \{0,1\}^n \times \{0,1\}^a \to \{0,1\}^\ell$ with the following property. For any arbitrarily correlated ℓ-bit random variables $Y, Y_1, \ldots, Y_t$, with Y uniform, any a-bit strings $\alpha, \alpha_1, \ldots, \alpha_t$, and for any weak-source W that is independent of the joint distribution of $Y, Y_1, \ldots, Y_t$, it holds that whenever $\alpha \notin \{\alpha_i\}_{i=1}^{t}$, the random variable $\mathsf{AdvCB}(Y, W, \alpha)$ is close to uniform even conditioned on the joint distribution of $\{\mathsf{AdvCB}(Y_i, W, \alpha_i)\}_{i=1}^{t}$.

With correlation breakers in hand, we are ready to define the sequence of M_i's. Let $\{X_i\}_{i=1}^{r}$ be the sequence generated in Step 1. Note that, unlike the first implementation, we only generate a single sequence. Let W be a weak-source that is independent of this sequence. For $i \in [r]$ and $j \in [m]$, we define the j'th row of M_i by $(M_i)_j = \mathsf{AdvCB}\left(X_i, W, \mathsf{ECC}(i)_j\right)$. That is, we use the j'th entry of the codeword corresponding to message i as the advice for the correlation breaker.

The analysis proceeds as follows. Let X_g be a good variable and let $i_1, \ldots, i_t \in [r] \setminus \{g\}$. As before, by our choice of δ, for any fixed $c \in [t]$, the codewords $\mathsf{ECC}(g)$ and $\mathsf{ECC}(i_c)$ agree on at most $1/(10t)$ fraction of their entries. Thus, $\mathsf{ECC}(g)_j \notin \{\mathsf{ECC}(i_c)_j\}_{c=1}^{t}$ for at least 0.9 fraction of $j \in [m]$. Therefore, and using the fact that X_g is uniform, the property of AdvCB implies that $(M_g)_j$ is close to uniform even conditioned on the joint distribution of $\{(M_{i_c})_j\}_{i=1}^{c}$ for 0.9 fraction of $j \in [m]$, as desired.

To summarize, while in this solution we consumed a single source so to break the undesired correlations, in the first solution we used more sources so not to introduce undesired correlations to begin with.

C. Step 3 – Merging while preserving independence

As mentioned, most of the technical effort of this work is invested in the third step, in which we merge the rows of each matrix M_i, computed in the previous step, while preserving the independence across the sequence. In this section we show how to use $2/\delta + O(1)$ sources, each with min-entropy $(\log n)^{1+\delta}$, so to accomplish this task.

The strategy that we employ is to reduce the problem of merging any number of variables (namely, the rows of a matrix) while preserving independence to the simplest case of merging only two variables while preserving independence. As this atomic primitive is the most delicate to analyze, in this section we only describe the reduction to the two variables case. Let us start by giving a precise formulation for the problem of merging two random variables while preserving independence. For simplicity, we consider only the case $t = 1$, though what to be presented next can be easily generalized to any t. Indeed, only Step 2 required a non-trivial idea to support arbitrary large t without increasing the number of sources.

We are given a pair of ℓ-bit random variables X, Y, both of which are uniform, though they may correlate arbitrarily. Let X', Y' be a second pair of ℓ-bit random variables. We are not guaranteed that these random variables are uniform. More perilously, X', Y' may arbitrarily correlate amongst themselves and with X, Y. Assume, however, that we are guaranteed that at least one of the following holds:

1) X is uniform even conditioned on X'; or
2) Y is uniform even conditioned on Y'.

Our goal is to merge X, Y while preserving this independence. More precisely, we would like to design an efficiently-computable function

$$\mathsf{IPMerg}\colon \{0,1\}^{\ell} \times \{0,1\}^{\ell} \to \{0,1\}^{\ell}$$

such that $\mathsf{IPMerg}(X, Y)$ is close to uniform even conditioned on $\mathsf{IPMerg}(X', Y')$. In Section III-D we show how to accomplish this task. As a matter of fact, such a primitive does not exist per se, and some fresh randomness, in the form of an independent weak-source, is required for the purpose of independence-preserving merging.

1) Independence-preserving half condensers and mergers: In this section we show how to obtain an independence-preserving merger for an arbitrary number of variables given the two-variables independence-preserving merger IPMerg, discussed in the previous section, as a black box. Due to its high min-entropy requirement, this new merger will not be the actual merger that will be used to merge the rows of the matrices obtained by Step 2. Nevertheless, the merger will be used as a building block for the construction of the final merger that will be used by our extractor.

With IPMerg in hand, one can easily implement what we call an *independence-preserving half-condenser*. This is a function $\mathsf{IPHalfCond}\colon \{0,1\}^{r \times \ell} \to \{0,1\}^{(r/2) \times \ell}$ such that if M, M' are random variables in the form of $r \times \ell$ binary matrices with M being somewhere-independent of M', and where each row of M is uniform, then $\mathsf{IPHalfCond}(M)$ is somewhere-independent of $\mathsf{IPHalfCond}(M')$, and each row of $\mathsf{IPHalfCond}(M)$ is uniform. Indeed, for any $j \in [r/2]$, one can simply set $\mathsf{IPHalfCond}(M)_j = \mathsf{IPMerg}(M_{2j-1}, M_{2j})$. It is easy to see that the independence is preserved. Indeed, if M_g is close to uniform even conditioned on M'_g for some $g \in [r]$ then by the guarantee of IPMerg, and by construction, $\mathsf{IPHalfCond}(M)_{\lceil g/2 \rceil}$ is close to uniform even conditioned on $\mathsf{IPHalfCond}(M')_{\lceil g/2 \rceil}$. Further, one can show that all rows of $\mathsf{IPHalfCond}(M)$ are close to uniform.

Of course, one can invoke $\mathsf{IPHalfCond}$ again, this time applied not to M, M', but rather to $\mathsf{IPHalfCond}(M)$ and $\mathsf{IPHalfCond}(M')$, so to reduce the number of rows by a factor of 4 while preserving independence, and so forth. By a careful analysis, one can show that the use of a fresh weak-source per application of $\mathsf{IPHalfCond}$ is not required. Instead, one can "juggle" between two sources – for even iterations use one source and for odd iterations use the other. So, using only two sources, one can apply $\mathsf{IPHalfCond}$ for $\log r$ iterations and merge M to a random variable in the form of a string which is independent of the string obtained by merging M'.

Unfortunately, for the first iteration alone, the min-entropy required by the source for $\mathsf{IPHalfCond}$ is $\Theta(r \log n)$. As the M_i's obtained by Step 2 have $r = \Omega(\log n)$ rows, this requires $\Omega(\log^2 n)$ min-entropy from the sources, which is more than what we can afford. On the other hand, we used only 2 sources for the entire merging process, and as the number of rows decreases exponentially with the number of iterations, the min-entropy requirement for the first iteration dominates the total min-entropy that is needed for the entire merging process. That is, the merger described above works when given two independent sources with min-entropy $O(r \log n)$.

2) Multi-source independence-preserving condensers and mergers: As mentioned, the merger that was constructed in the previous section requires more min-entropy than we can afford from its two auxiliary sources. We start this section by presenting an independence-preserving condenser that is guaranteed to work even with much lower min-entropy sources. This condenser, however, will require two auxiliary sources for its operation as apposed to $\mathsf{IPHalfCond}$ that used a single source. The construction of this two-source independence-preserving condenser relies on the two-source independence-preserving merger that was constructed in Section III-C1. We then turn to construct the multi-source independence-preserving merger that will be used by our extractor.

For the construction of our two-source independence-preserving condenser we make use of expander graphs. More precisely, by using an appropriate explicit expander graph, for any integer r and for any $\varepsilon > 0$, one can obtain a bipartite graph $G = (L, R, E)$, with $|L| = |R| = r$ and right-degree $d = O(1/\varepsilon)$, that has the following property. For any set $B \subset L$ of size $|B| \leq 0.1r$, all but ε fraction of the vertices in R have a neighbor outside of B. Therefore, by throwing away all but an arbitrary subset of $10\varepsilon r$ vertices from R, one obtains a bipartite graph $G' = (L', R', E')$ with $L' = L$, $|R'| = 10\varepsilon r$, and right-degree $d = O(1/\varepsilon)$, such that for any set B as above, all but 0.1 fraction of the vertices in R' have a neighbor outside of B. The important point here is that the size of R' can be made much smaller than the size of L' at the expense of increasing the right-degree.

Set $\varepsilon = (\log n)^{-\delta}$ and let $G' = (L', R', E)$ be the graph described above with $|L'| = r$, $|R'| = r' = O(r/(\log n)^{\delta})$, and right-degree $d = O(1/\varepsilon) = O((\log n)^{\delta})$. We identify L' with $[r]$, and for each $v \in R'$ consider the $d \times \ell$ matrix M_v that is obtained by taking the rows of M which correspond to the neighbors of v in G'. To summarize, we associate with M a sequence of r' matrices of order $d \times \ell$.

Our two-source independence-preserving condenser is defined as follows. We apply the independence-preserving merger described in Section III-C1 to each of the matrices M_v, with the same pair of sources for all v. This yields an $r' \times \ell$ matrix. We now show that 0.9 fraction of the rows of this matrix are close to uniform even conditioned on the corresponding row of the condensed M'.

The analysis is straightforward – as M is independent of M' in 0.9 fraction of its rows, the property of G' guarantees that for 0.9 fraction of $v \in R'$ it holds that M_v is somewhere-independent of M'_v. Thus, for any such v, we have that the merged value of M_v is close to uniform even conditioned on the merged value of M'_v.

By consuming two sources with min-entropy $d \log n = (\log n)^{1+\delta}$ we condense the $r \times \ell$ matrix M to a matrix with $r/(\log n)^{\delta}$ rows while preserving the independence guarantee for 0.9 fraction of the rows. As $r = O(\log n)$, one can repeat this condensing process for $1/\delta$ iterations, each time using two fresh sources, so to obtain an independence-preserving merger that consumes $2/\delta$ sources each with min-entropy $(\log n)^{1+\delta}$. This will be our final merger.

D. A two-variables independence-preserving merger

In previous sections we saw how to reduce the construction of multi-source extractors to that of merging two random variables while preserving independence. Let us recall the setting. We are given a pair of ℓ-bit random variables X, Y, both of which are uniform, though they may correlate arbitrarily. Let X', Y' be a second pair of ℓ-bit random variables that are arbitrarily correlated amongst themselves and with X, Y. We are guaranteed that at least one of the following holds: (1) Independence in X – the random variable X is uniform even conditioned on X'; or (2) Independence in Y – the random variable Y is uniform even conditioned on Y'.

Our goal is to design an efficiently-computable function $\mathsf{IPMerg}\colon \{0,1\}^{\ell} \times \{0,1\}^{\ell} \to \{0,1\}^{\ell}$ such that $\mathsf{IPMerg}(X, Y)$ is close to uniform even conditioned on $\mathsf{IPMerg}(X', Y')$. Of course, simply outputting, say, the first input, won't do as although we will output a uniform string, it could be the case that this string is the one correlated with the corresponding string in the other pair. In fact, as mentioned in the previous section, there is no function with such a guarantee. Therefore, relaxing the problem a bit, we would like to design an efficiently-computable function $\mathsf{IPMerg}\colon \{0,1\}^{\ell} \times \{0,1\}^{\ell} \times \{0,1\}^{n} \to \{0,1\}^{\ell}$ such that $\mathsf{IPMerg}(X, Y, W)$ is uniform even conditioned on $\mathsf{IPMerg}(X', Y', W)$, where W is an (n, k)-source that is independent of the joint distribution of X, Y, X', Y'.

1) Some preliminary suggestions: A good starting point for motivating our construction is the following useful property of strong seeded extractors. Roughly speaking, it can be shown that as long as one uses a fresh seed, previous outputs of an extractor do not reveal information about the future output, even if the sources being used are arbitrarily correlated. More precisely, we have the following fact.

Fact III.1. *Let* $\mathsf{Ext}\colon \{0,1\}^{n} \times \{0,1\}^{d} \to \{0,1\}^{m}$ *be a strong seeded extractor for min-entropy k. Let S be a random variable that is uniformly distributed over* $\{0,1\}^{d}$*. Let W, W', S' be arbitrarily correlated random variables that are jointly independent of S. Assume further that $H_{\infty}(W) \gg k$. Then,* $\mathsf{Ext}(W, S)$ *is close to uniform even conditioned on* $\mathsf{Ext}(W', S')$.

It is not hard to prove Fact III.1 though we omit its proof. We will not use this fact anyhow and recalled it here for motivating the actual construction. At any rate, given Fact III.1, a first attempt would be to completely ignore W, and define $\mathsf{IPMerg}_1(X, Y, W) = \mathsf{Ext}(X, Y)$. The reasoning behind this suggestion is the following: If there is independence in Y then one might make the hasty conclusion that regardless of the correlations between the sources X and X', Fact III.1 tells us that $\mathsf{Ext}(X, Y)$ is close to uniform even conditioned on $\mathsf{Ext}(X', Y')$. This, of course, is flawed – the source X and the seed Y to Ext are correlated and so, regardless of X', Y', the output $\mathsf{Ext}(X, Y)$ is not necessarily close to uniform. Of course, we knew all along that one must use the extra randomness of W, so this idea was bound to fail.

A revised idea would be to use the fresh source W as a "buffer" between X, Y, and define $\mathsf{IPMerg}_2(X, Y, W) = \mathsf{Ext}_{\mathsf{out}}(X, \mathsf{Ext}_{\mathsf{in}}(W, Y))$. This idea in fact almost works when there is independence in Y. We revise the construction a bit further and define $\mathsf{IPMerg}_3(X, Y, W) = \mathsf{Ext}_{\mathsf{out}}(X, \mathsf{Ext}_{\mathsf{in}}(W, Y|_s))$, where $Y|_s$ stands for the length s

prefix of Y. In particular, we set $s = \ell/10$. We further set the output length of the inner extractor $\mathsf{Ext}_{\mathsf{in}}$ (which is the seed length for the outer extractor $\mathsf{Ext}_{\mathsf{out}}$) to s. The outer extractor $\mathsf{Ext}_{\mathsf{out}}$ is also set to output s bits given a source X with min-entropy 0.7ℓ. By this choice of parameters, IPMerg_3 has output length $s = \ell/10$ rather than ℓ, though the reader should not worry about this issue as the number of output bits can be easily increased back to ℓ using standard techniques, and in any case, our final construction does not suffer this shrinkage in the output length. One can now prove the following claim.

Claim III.2. *Assume that $k \gg \ell$. If there is independence in Y then $\mathsf{IPMerg}_3(X, Y, W)$ is close to uniform even conditioned on $\mathsf{IPMerg}_3(X', Y', W)$.*

We will not make use of Claim III.2 since, as we discuss next, we do not know how to prove a similar statement for the independence in X case. Nevertheless, IPMerg_3 will be used in our final construction.

What can go wrong by using IPMerg_3 assuming that there is independence in X rather than in Y? At first look, IPMerg_3 seems promising. Indeed, in that case X is uniform even conditioned on the source X', the seed Y is uniform, and thanks to the buffer source W, $\mathsf{Ext}_{\mathsf{in}}(W, Y|_s)$ yields a seed for the outer extractor $\mathsf{Ext}_{\mathsf{out}}$ that is independent of X. What harm can the correlation between Y and Y' cause? Well, recall that each of the pairs (X, Y), (Y, Y'), (Y', X') might be correlated. Therefore, by conditioning on Y, Y' we may introduce correlations between X and X'. Thus, our proof strategy employed in Claim III.2, which involves conditioning on the values of (prefixes of) Y, Y' is problematic.

Although we do not know how to show that IPMerg_3 works when there is independence in X, it does work when there is independence in Y. Moreover, the problem we have seems to arise only due to the correlations between X and the other random variables. In fact, this can be made formal as one can show that IPMerg_3 works perfectly assuming both X, Y are uniform and assuming that one of the following holds:

1) X is uniform even conditioned on the joint distribution of Y, X', Y'; or
2) Y is uniform even conditioned on Y'.

Note that Case 2 is the original independence in Y case, so the analysis above holds for that case. Case 1 is stronger than the independence in X case in that it assumes that X is uniform not only conditioned on X' but rather conditioned on all other random variables in the picture.

In the next section we show how to guarantee that one of these stronger properties holds given only the original assumption. This reduction, however, will cause some further complications that will require our attention.

2) Relying on a hierarchy of independence: The discussion above leads us to consider the following problem. Let X, X', Y, Y' be random variables for which the original assumption holds, namely, both X and Y are uniform though correlated, and we have independence either in X or in Y. As before, let W be a fresh (n, k)-source. This source was used in the suggestion above as a "buffer" between X and Y. We will make further use of W, and so this auxiliary source has several conceptual roles in the final construction.

For what comes next, we also need a second (n, k)-source Z, though we do not consider this source as a new source of randomness with respect to X, X', Y, Y', as we do not require that Z is independent of (X, X', Y, Y'). We only need Z to have some min-entropy left even conditioned on (X, X', Y, Y'). Having the big picture in mind, the variables X, X', Y, Y' are not given as inputs to our extractor but are computed by the first two steps. The source Z is one of the sources used by these steps, and we make sure that even conditioned on (X, X', Y, Y'), the source Z has some min-entropy left.

We would like to design a pair of functions $\mathsf{a}\colon \{0,1\}^\ell \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^\ell$, $\mathsf{b}\colon \{0,1\}^\ell \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^\ell$, such that: (1) If there is independence in X then $\mathsf{b}(X, Z, W)$ is close to uniform even conditioned on the joint distribution of $\mathsf{b}(X', Z, W)$, $\mathsf{a}(Y, Z, W)$, and $\mathsf{a}(Y', Z, W)$; and, (2) If there is independence in Y then $\mathsf{a}(Y, Z, W)$ is close to uniform even conditioned on $\mathsf{a}(Y', Z, W)$. We further require that each of $\mathsf{b}(X, Z, W)$, $\mathsf{a}(Y, Z, W)$ is uniform. By setting $X_{\mathsf{new}} = \mathsf{b}(X, Z, W)$, $X'_{\mathsf{new}} = \mathsf{b}(X', Z, W)$, $Y_{\mathsf{new}} = \mathsf{a}(Y, Z, W)$, and $Y'_{\mathsf{new}} = \mathsf{a}(Y', Z, W)$, we see that if there is independence in X or in Y then one of the following holds: (1) X_{new} is close to uniform even conditioned on $Y_{\mathsf{new}}, X'_{\mathsf{new}}, Y'_{\mathsf{new}}$; or Y_{new} is close to uniform even conditioned on Y'_{new}. Furthermore, each of $X_{\mathsf{new}}, Y_{\mathsf{new}}$ is close to uniform. This is exactly the stronger guarantee that we set off to obtain. It is therefore tempting to just go ahead and use this reduction in a black-box manner, and define $\mathsf{IPMerg}_4(X, Y, Z, W) = \mathsf{IPMerg}_3(X_{\mathsf{new}}, Y_{\mathsf{new}}, W)$. Indeed, using the functions a, b, we "transformed" the original guarantee on X, Y, X', Y' to the stronger guarantee on $X_{\mathsf{new}}, Y_{\mathsf{new}}, X'_{\mathsf{new}}, Y'_{\mathsf{new}}$, under which one might hope that IPMerg_3 can be shown to work. However, by a more careful inspection one can see that a new problem arises – the variables X_{new}, X'_{new}, Y_{new}, and Y'_{new} are no longer independent of W.

So, unfortunately, the idea of breaking the correlations between X and (Y, X', Y') so to handle the independence in X case in a black-box manner while keeping intact the analysis of the independence in Y case fails. Fortunately, however, the *specific* way in which we implement a, b does allow us to make use of the ideas developed so far. It turns out that by a suitable modification to IPMerg_4, and by using specific instantiation for a, b, we can handle both cases. So, in order to continue with the analysis we must present our construction for a, b.

3) A specific implementation for establishing a hierarchy of independence: In this section we present a specific implementation for the functions a, b that were presented in the previous section. The construction is based on the technique of alternating extraction. As it turns out, we can also define the function a with one argument less, and so we define $\mathsf{a}\colon \{0,1\}^\ell \times \{0,1\}^n \to \{0,1\}^\ell$, $\mathsf{b}\colon \{0,1\}^\ell \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^\ell$, by $\mathsf{a}(X, W) = \mathsf{Ext}(W, X|_s)$, $\mathsf{b}(X, Z, W) = \mathsf{Ext}(W, \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X|_s)))$. Here Ext is a strong seeded extractor with ℓ output bits, and Ext_s is the extractor obtained by truncating the output of Ext after s bits. We argue that this implementation meets our needs. To be more precise, we "cluster" the random variables that are obtained in different stages of the computation of a, b, and set

$$\begin{aligned}\mathcal{M} &= X, Y, X', Y',\\ \mathcal{A} &= \mathsf{a}(X, W), \mathsf{a}(X', W), \mathsf{a}(Y, W), \mathsf{a}(Y', W),\\ \mathcal{Z} &= \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X|_s)), \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X'|_s)).\end{aligned}$$

So, $\mathcal{M}$ denotes the two pairs of random variables that are fed as inputs to the mergers. The next stage of computation is captured by $\mathcal{A}$. Note, in particular that this also includes $\mathsf{Ext}_s(W, X|_s)$ and $\mathsf{Ext}_s(W, X'|_s)$. Lastly, $\mathcal{Z}$ denotes the seeds fed to the outer extractor in the computation of b.

We do not analyze a, b here and are satisfied with stating the following claim. In the next section we show how one can use this specific implementation of a, b so to obtain our final two-variables independence-preserving merger.

Claim III.3. *With the notation set so far, the following holds.*

- *If there is independence in X then $\mathsf{b}(X, Z, W)$ is close to uniform even conditioned on $\mathsf{b}(X', Z, W), \mathcal{Z}, \mathcal{A}, \mathcal{M}$.*
- *If there is independence in Y then $\mathsf{a}(Y, W)$ is close to uniform even conditioned on $\mathsf{a}(Y', W), \mathcal{M}$.*
- *Regardless of whether there is independence in X or in Y, it holds that $\mathsf{a}(Y, W)$ is close to uniform conditioned on $\mathcal{M}$. Further, $\mathsf{b}(X, Z, W)$ is close to uniform conditioned on $\mathcal{Z}, \mathcal{A}, \mathcal{M}$.*

4) Our final two-variables independence-preserving merger: Before presenting our two-variables independence-preserving merger, we need to acquire one more object – an extractor with weak-seeds due to Raz [Raz05]. This is a strong seeded extractor that works even if the seed is not uniform, but rather has min-entropy rate $1/2 + \delta$ for an arbitrarily small constant $\delta > 0$. With Raz's extractor and with a, b that were defined in the previous section, we can finally define our two-variables independence-preserving merger by $\mathsf{IPMerg}(X, Y, Z, W) = \mathsf{Raz}\left(\mathsf{b}(X, Z, W), \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))\right)$. We set the output length of the inner extractor $\mathsf{Ext}_{\mathsf{in}}$ to s', and the output length of Raz to m. Recall that ℓ is the output length of a, b. We set things up such that $s' \gg s$ and $\ell \gg m$. This is our way of making sure that some random variables will have enough min-entropy even conditioned on some other, shorter, random variables. Our goal is to show that $\mathsf{IPMerg}(X, Y, Z, W)$ is close to uniform even conditioned on $\mathsf{IPMerg}(X', Y', Z, W) = \mathsf{Raz}\left(\mathsf{b}(X', Z, W), \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y', W))\right)$. For the analysis we consider two cases, corresponding to whether there is independence in X or independence in Y.

a) Analyzing the independence in X case.: By Claim III.3, the source $\mathsf{b}(X, Z, W)$ to Raz is close to uniform even conditioned on $\mathsf{b}(X', Z, W), \mathcal{A}, \mathcal{Z}, \mathcal{M}$. Note also that conditioned on these random variables, $\mathsf{b}(X, Z, W)$ is a deterministic function of W, and so $\mathsf{b}(X, Z, W)$ is close to uniform conditioned on the other source $\mathsf{b}(X', Z, W)$ (which we already know) and on the seeds $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$, $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y', W))$ to Raz (as they are deterministic functions of Z conditioned on $\mathcal{A}$).

Intuitively, this allows us to "replace" the source $\mathsf{b}(X, Z, W)$ in Raz by the uniform distribution. That is, it is enough to show that $\mathsf{Raz}(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W)))$ is close to uniform conditioned on $\mathsf{IPMerg}(X', Y', Z, W)$, where U is uniform and independent of all the other random variables in the picture.

This is a much easier task! Indeed, the uniform distribution is some valid source for Raz (granted, with lots of min-entropy). So, as long as the seed $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$ fed to Raz is close to uniform, we have that with high probability over $s \sim \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$, the output $\mathsf{Raz}(U, s)$ is close to uniform. Now, this random variable is completely independent of all other random variables, and in particular it is close to uniform even conditioned on $\mathsf{IPMerg}(X', Y', Z, W)$. To show that $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$ is close to uniform is not hard and we skip the proof. We now move to the more delicate case.

b) Analyzing the independence in Y case.: By Claim III.3, $\mathsf{a}(Y, W)$ is close to uniform even conditioned on $\mathsf{a}(Y', W), \mathcal{M}$. We note that $\mathsf{a}(Y, W)$ is independent of Z conditioned on $\mathcal{M}$, and so $\mathsf{a}(Y, W)$ is close to uniform even conditioned on $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y', W))$, $\mathsf{a}(Y', W)$, and $\mathcal{M}$. Therefore, the seed $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$ to Raz is close to uniform even conditioned on

$$\mathcal{H} = \mathsf{a}(Y, W), \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y', W)), \mathsf{a}(Y', W), \mathcal{M}.$$

In fact, as $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$ is independent of $\mathcal{A}$ conditioned on $\mathcal{H}$, we have that $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$ is close to uniform even conditioned on

$$\mathcal{H}' = \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y', W)), \mathcal{A}, \mathcal{M}.$$

Recall that

$$\mathsf{b}(X', Z, W) = \mathsf{Ext}(W, \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X'|_s))).$$

Note that the variable $\mathsf{Ext}_s(W, X'|_s)$ is contained in $\mathcal{A}$, which in turn is contained in $\mathcal{H}'$. Thus, $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X'|_s))$ is a deterministic function of Z conditioned on $\mathcal{H}'$. Now, although the seed $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$ is also a deterministic function of Z conditioned on $\mathcal{H}'$, and in particular it may correlate with $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X'|_s))$, we can still condition on $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X'|_s))$ and, with high probability, get that the seed $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$ has min-entropy $s' - s$. In fact, as we would like to remove the correlations between the source and the seed fed to Raz, we also condition on the "part" of the source $\mathsf{b}(X, Z, W)$ that correlates with the seed $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$, that is, we would like to condition on both $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X|_s))$ and $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X'|_s))$. By interpreting $s' \gg s$ as $s' > 20s$ we have that conditioned on

$$\mathcal{H}'' = \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X|_s)), \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X'|_s)), \mathcal{H}',$$

the seed $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$ has min-entropy $s' - 2s > 0.9s'$.

At this point, the output $\mathsf{IPMerg}(X', Y', Z, W)$ is a deterministic function of $\mathsf{b}(X', Z, W)$ which, in turn, is a deterministic function of W. Thus, we can fix the output $\mathsf{IPMerg}(X', Y', Z, W)$ without affecting the seed $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$. That is, we have that the latter seed is close to having min-entropy rate 0.9 even conditioned on $\mathsf{IPMerg}(X', Y', Z, W), \mathcal{H}''$. This is good enough for a seed passed to Raz.

To conclude the proof, it suffices to show that the source $\mathsf{b}(X, Z, W)$ fed to Raz has sufficient amount of min-entropy even conditioned on the same set of variables $\mathsf{IPMerg}(X', Y', Z, W), \mathcal{H}''$. Indeed, note that conditioned on these variables, the source $\mathsf{b}(X, Z, W)$ and seed $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$ fed to Raz are independent. This is why we also bothered to condition on $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X|_s))$ – the part of the source $\mathsf{b}(X, Z, W)$ that correlates with the seed $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y, W))$.

We now turn to show that the source $\mathsf{b}(X, Z, W)$ has high min-entropy even conditioned on $\mathsf{IPMerg}(X', Y', Z, W), \mathcal{H}''$. By Claim III.3, $\mathsf{b}(X, Z, W)$ is close to uniform conditioned on $\mathcal{Z}, \mathcal{A}, \mathcal{M}$. Note that conditioned on $\mathcal{Z}, \mathcal{A}, \mathcal{M}$, the source $\mathsf{b}(X, Z, W)$ is independent of $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y', W))$, and so the source $\mathsf{b}(X, Z, W)$ is close to uniform even conditioned on $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(Y', W))$,$\mathcal{Z}, \mathcal{A}, \mathcal{M}$. At this point, $\mathsf{IPMerg}(X', Y', Z, W)$ is a deterministic function of $\mathsf{b}(X', Z, W)$ which, as we condition on $\mathcal{Z}$, is in turn a deterministic function of W. As $\ell \gg m$, we are guaranteed that even conditioned on $\mathsf{IPMerg}(X', Y', Z, W)$, the source $\mathsf{b}(X, Z, W)$, which was close to uniform prior to the conditioning, has not lost much of its min-entropy. As $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X|_s))$ and $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X'|_s))$ are contained in $\mathcal{Z}$, the latter conditioning on $\mathsf{IPMerg}(X', Y', Z, W)$ completes the list of random variables on which we condition to the desired set $\mathsf{IPMerg}(X', Y', Z, W), \mathcal{H}''$.

References

[AGM03] N. Alon, O. Goldreich, and Y. Mansour. Almost k-wise independence versus k-wise independence. *Information Processing Letters*, 88(3):107–110, 2003.

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

[BKS+05] B. Barak, G. Kindler, R. Shaltiel, B. Sudakov, and A. Wigderson. Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors. In *Proceedings of the thirty-seventh annual ACM Symposium on Theory of Computing*, pages 1–10. ACM, 2005.

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

[Bra10] M. Braverman. Polylogarithmic independence fools AC0 circuits. *Journal of the ACM (JACM)*, 57(5):28, 2010.

[BRSW12] B. Barak, A. Rao, R. Shaltiel, and A. Wigderson. 2-source dispersers for $n^{o(1)}$ entropy, and Ramsey graphs beating the Frankl-Wilson construction. *Annals of Mathematics*, 176(3):1483–1544, 2012.

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

[CGL15] E. Chattopadhyay, V. Goyal, and X. Li. Non-malleable extractors and codes, with their many tampered extensions. *arXiv preprint arXiv:1505.00107*, 2015.

[Coh15a] G. Cohen. Local correlation breakers and applications to three-source extractors and mergers. In *IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 845–862. IEEE, 2015.

[Coh15b] G. Cohen. Non-malleable extractors – new tools and improved constructions. In *Electronic Colloquium on Computational Complexity (ECCC)*, page 183, 2015.

[Coh15c] G. Cohen. Two-source dispersers for polylogarithmic entropy and improved Ramsey graphs. *arXiv preprint arXiv:1506.04428*, 2015.

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

[CZ15] E. Chattopadhyay and D. Zuckerman. Explicit two-source extractors and resilient functions. *Electronic Colloquium on Computational Complexity (ECCC)*, 2015.

[DGJ+10] I. Diakonikolas, P. Gopalan, R. Jaiswal, R. Servedio, and E. Viola. Bounded independence fools halfspaces. *SIAM Journal on Computing*, 39(8):3441–3462, 2010.

[DLWZ14] Y. Dodis, X. Li, T. D. Wooley, and D. Zuckerman. Privacy amplification and nonmalleable extractors via character sums. *SIAM Journal on Computing*, 43(2):800–830, 2014.

[DP07] S. Dziembowski and K. Pietrzak. Intrusion-resilient secret sharing. In *48th Annual IEEE Symposium on Foundations of Computer Science*, pages 227–237, 2007.

[DW09] Y. Dodis and D. Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the forty-first annual ACM Symposium on Theory of Computing*, pages 601–610. ACM, 2009.

[Erd47] P. Erdős. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53(4):292–294, 1947.

[Fei99] U. Feige. Noncryptographic selection protocols. In *40th Annual Symposium on Foundations of Computer Science*, pages 142–152. IEEE, 1999.

[GUV09] V. Guruswami, C. Umans, and S. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM*, 56(4):20, 2009.

[KLW10] A. Klivans, H. Lee, and A. Wan. Mansour's conjecture is true for random DNF formulas. In *23st Annual Conference on Learning Theory - COLT 2010*, pages 368–380, 2010.

[Li12a] X. Li. Design extractors, non-malleable condensers and privacy amplification. In *Proceedings of the forty-fourth annual ACM Symposium on Theory of Computing*, pages 837–854, 2012.

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

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

[Li13b] X. Li. New independent source extractors with exponential improvement. In *Proceedings of the forty-fifth annual ACM Symposium on Theory of Computing*, pages 783–792. ACM, 2013.

[Li15a] X. Li. Improved constructions of two-source extractors. In *Electronic Colloquium on Computational Complexity (ECCC)*, page 125, 2015.

[Li15b] X. Li. Three-source extractors for polylogarithmic min-entropy. *Electronic Colloquium on Computational Complexity (ECCC)*, 2015.

[Mek15] R. Meka. Explicit resilient functions matching Ajtai-Linial. *arXiv preprint arXiv:1509.00092*, 2015.

[Ram28] F. P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, 30(4):338–384, 1928.

[Rao09] A. Rao. Extractors for a constant number of polynomially small min-entropy independent sources. *SIAM Journal on Computing*, 39(1):168–194, 2009.

[Raz05] R. Raz. Extractors with weak random seeds. In *Proceedings of the thirty-seventh annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.

[RTS00] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[Tal14] A. Tal. Tight bounds on the fourier spectrum of AC0. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 21, page 174, 2014.

[Vio14] E. Viola. Extractors for circuit sources. *SIAM Journal on Computing*, 43(2):655–672, 2014.