Electronic Colloquium on Computational Complexity, Report No. 14 (2016)

Extractors for Near Logarithmic Min-Entropy

Gil Cohen* Leonard J. Schulman†

February 2, 2016

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 $(\text{polylog}\, n)$-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*.

*Computing and Mathematical Sciences Department, Caltech. Email: coheng@caltech.edu.
†Computing and Mathematical Sciences Department, Caltech. Email: schulman@caltech.edu.

ISSN 1433-8092

Contents

1 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 (see Preliminaries for the formal definition).

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 $\mathsf{Ext}\colon (\{0,1\}^n)^s \to \{0,1\}^m$ 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 (see Appendix C for a summary of constructions from the literature).

1.1 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 $K=2^{\text{poly}\log\log N}$, 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 $\mathsf{Disp}\colon (\{0,1\}^n)^2 \to \{0,1\}$ with the property that for any 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.

1.2 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 $c \gg 2$ (see Appendix C).

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. We elaborate on this in Section 2.

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.

1.3 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 1.1. *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 1.1 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 1.1 can be an arbitrary function of n. In particular, by setting $\delta = (\log\log n)^{-1}$, Theorem 1.1 yields an explicit multi-source extractor for $2\log\log n + O(1)$ sources with min-entropy $O(\log n)$. More generally, Theorem 1.1 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 (see Appendix C).

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, Li12c, 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 make use of 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.

1.3.1 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. Further, in Section 1.3.2 we present an improved construction of extractors for zero-fixing sources that is based on multi-source extractors. In that context, the number of output bits is of central concern.

By using the primitives developed for the proof of Theorem 1.1, 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 1.2. *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.*

1.3.2 Zero-fixing extractors for near double-logarithmic entropy

As mentioned, multi-source extractors is one of the approaches taken in the literature to bypass the fact that one cannot extract randomness from a single source if the sole guarantee is a lower bound on its min-entropy. A second approach that was considered in the literature insists on extracting randomness from a single source, though with the guarantee that the source belongs to a restricted class of sources. Put in other words, assuming that the source has some structure.

A well-studied class of sources are *bit-fixing sources* [Vaz85, BBR85, CGH+85, KZ06, GRS06, Rao09, RV13, CS15]. An (n,k)-bit-fixing source is a random variable X over $\{0,1\}^n$, where some $n-k$ of the bits of X are fixed, and the joint distribution of the remaining k bits is uniform. Clearly, the class of (n,k)-bit-fixing sources is a very structured class of sources within all sources with min-entropy k, and so one might hope that we would have a complete understanding of this class. Unfortunately, however, this is not the case. There are basic aspects that are not well-understood even when restricting to a subclass of bit-fixing sources called *zero-fixing sources* [CS15, GVWZ15]. An (n,k)-zero-fixing source is an (n,k)-bit-fixing source in which all the fixed bits are set to 0.

One aspect which is unclear even when restricting ourselves to zero-fixing sources concerns the amount of "accessible entropy". Considering the bit-fixing case first, Kamp and Zuckerman [KZ06] observed that for *any* desired k, one can efficiently extract $0.5\log_2(k) - O(1)$ bits that are close to uniform from (n,k)-bit-fixing sources. That is, regardless of how low the entropy is, one can still "access" a logarithmic amount of it. Furthermore, there exists some constant $c > 1$ such that for any $k > (\log n)^c$, there are explicit constructions of extractors for (n,k)-bit-fixing sources, having $(1-o(1))k$ output bits [GRS06, Rao09]. These construc-

tions come close to the threshold of a random function, which extracts $k - O(1)$ bits for any $k > 2\log n$, and which fails to output even a single non-constant bit for $k < (1/2)\log n$.

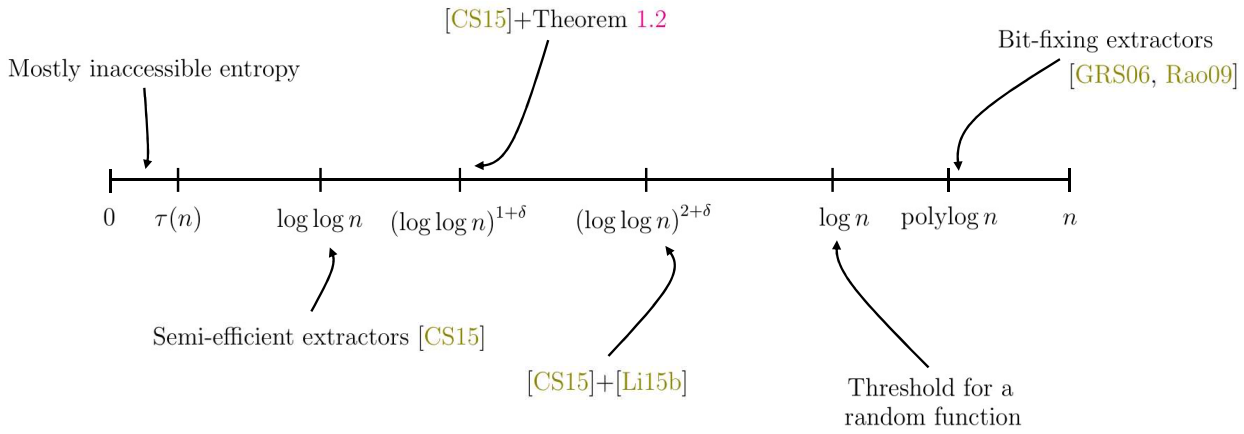


Figure 1: Summary of results on the accessible entropy of zero-fixing and bit-fixing sources. Theorem 1.3, which is obtained by the reduction of [CS15] together with Theorem 1.2, is the contribution of this work to our understanding of zero-fixing sources.

Reshef and Vadhan [RV13] considered the problem of how much of the entropy of an (n, k)-bit-fixing source is accessible in the low entropy regime $k = o(\log n)$. In [RV13] it is shown that any extractor that is computable by a space-bounded streaming algorithm – a computational model that captures the computational power of the extractor given by [KZ06], can output only $O(\log k)$ bits in this regime. Cohen and Shinkar [CS15] proved that regardless of computational aspects, most of the entropy is inaccessible for small values of k, even when restricting to zero-fixing sources. More precisely, there exists some slowly growing function $\tau(n)$ such that for $k < \tau(n)$, no function can extract $0.5\log_2(k) + O(1)$ bits that are close to uniform from (n, k)-zero-fixing sources. On the other hand, in [CS15] an extractor for (n, k)-bit-fixing sources with $k - O(1)$ output bits for any $k = \Omega(\log\log n)$ was shown to exists. Although this extractor is not explicit, its existence demonstrates that one can extract all the entropy of the source even in regime in which a random function fails.

Returning to zero-fixing sources, one can show that for $k = \Omega(\log\log n)$, a random function with $k - O(1)$ output bits is, with high probability, a zero-fixing extractor for (n, k)-sources. By establishing a reduction to multi-source extractors, and with Li's multi-source extractor [Li13a, Li15b] in hand, it was shown [CS15] that for any constant $\delta > 0$, there exists a $\text{poly}(n)$-time computable (n, k)-zero-fixing extractor for $k = (\log\log n)^{2+\delta}$, having $\Omega(k)$ output bits.

By plugging our extractor from Theorem 1.2 to the reduction from zero-fixing extractors to multi-source extractors [CS15], we obtain the following improved explicit zero-fixing extractor.

Theorem 1.3. *Let $\delta > 0$ be any constant, and let n be an integer. Set $k = (\log\log n)^{1+\delta}$.*

Then, there exists an efficiently-computable function

$$\mathsf{ZeroBFExt}\colon \{0,1\}^n \to \{0,1\}^{\Omega(k)},$$

with the following property. For any (n,k)-zero-fixing source X, it holds that $\mathsf{ZeroBFExt}(X)$ *is $2^{-k^{\Omega(1)}}$-close to uniform.*

1.4 Organization

In Section 2 we give a detailed, yet informal, overview of our extractor and its analysis. The reader may freely skip this section at any point as we make no use of the results that appear in it. Section 3 contains some standard notations and results that we frequently use throughout the paper. In Section 4 we design an efficient algorithm that transforms a constant number of sources to a sequence of what we call *somewhere-independent matrices*. Section 5 contains the construction of a single-source independence-preserving merger, though the min-entropy requirement of this merger is too high so to be used directly for our extractor. Based on this merger, in Section 6 we construct a multi-source independence-preserving merger that works for lower min-entropy. Finally, in Section 7 and Section 8 we prove Theorem 1.1 and Theorem 1.2, respectively. The proof of Theorem 1.3 is given in Section 9, though it can be read independently of previous sections.

2 An Informal Proof Overview

In this section we describe our extractors from Theorem 1.1 and Theorem 1.2. We also give a comprehensive and detailed overview of their analysis. Although informal, this overview presents most of the ideas that fit into the actual proof while overlooking mainly what we consider as distracting technicalities. Further, we allow ourselves to gradually develop the ideas so to motivate the final construction. This makes for a longer reading but hopefully for a more approachable presentation. At any rate, forward pointers to the formal proofs are given. We start by considering the extractor from Theorem 1.1 and only in Section 2.6 briefly discuss the further ideas involved in proving Theorem 1.2. We do not cover the proof of Theorem 1.3 in this section and the reader is referred to [CS15] for a proof overview that is relevant here as well.

2.1 The general strategy and context

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 is influenced by the recent breakthrough construction of two-source extractors by Chattopadhyay and Zuckerman [CZ15], and it is worth contrasting the two. 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 1.1, 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:

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 $r = \text{poly}(n)$.

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 $i_1, \ldots, i_t \in [r] \setminus \{g\}$, 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 $M_{i_1}, \ldots, M_{i_t}$. So, informally speaking, the matrix M_g is *somewhere independent* of $M_{i_1}, \ldots, M_{i_t}$. 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).

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, $M_{i_1}, \ldots, M_{i_t}$ 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 each of the following three sections we describe the ideas that go into each step, respectively.

2.2 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 (see Theorem 7.2). The goal of the first step of our construction is to achieve this bound on the number of bad variables without worrying at all 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 (see Definition 3.7). 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] (see Theorem 4.7). 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 $\beta > 0$ (see Theorem 4.5). 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.

For a formal treatment of the ideas that were covered in this section, we refer the reader to Lemma 4.3 and its proof that appears in Section 4.1.

2.3 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 2.3.3 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.

2.3.1 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. The unfamiliar reader is referred to Definition 3.6. 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.

2.3.2 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 (see Theorem 4.13). 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.

2.3.3 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 $\{\mathsf{AdvCB}(Y_i, W, \alpha_i)\}_{i=1}^{t}$ (see Definition 4.14 and Theorem 4.15).

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.

Working with algebraic-geometric codes is beneficial also for this implementation of Step 2. Indeed, the larger the advice length $a = \log_2 q$ is, the more min-entropy is required from W for the operation of AdvCB, and the larger must be the length of the X_i's, which in turn puts restrictions on the min-entropy of the sources used in Step 1. Thus, we would like to take a to be as small as possible. On the other hand, as before, the larger the number of rows m of the M_i's is, the larger the min-entropy that will be required from the sources used in Step 3. Thus, we want to minimize m as well. Recall that for the above analysis to go through, the relative distance was set to $\delta = 1 - 1/(10t)$. By using algebraic-geometric codes

we can get away with an advice of length $a = O(\log t)$ and $m = O(t \log r)$ rows. Working with Reed-Solomon codes or even with random codes will require more min-entropy from our sources. In particular, working with Reed-Solomon codes would prevent us from supporting min-entropy that is arbitrarily close to logarithmic.

2.4 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, and defer the presentation of the independence-preserving merger for two variables to Section 2.5.

Figure 2: An illustration of the two-variables independence-preserving merger. Each highlighted (darker) random variable represents a variable that is uniform conditioned on its respective random variable (which does not appear in the picture). On the left side, X is uniform even conditioned on X', whereas on the right side, Y is uniform even conditioned on Y'. Regardless of which random variable is uniform conditioned on its respective variable, the output of the merger is uniform conditioned on the output obtained by applying the merger to the other two random variables.

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 2.5 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.

2.4.1 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.

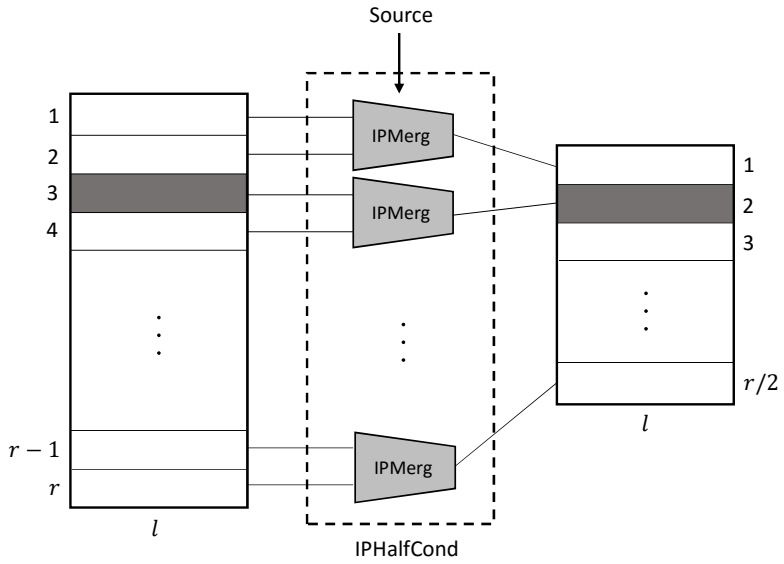


Figure 3: An illustration of the independence-preserving half condenser IPHalfCond that is based on the two-variables independence-preserving mergers IPMerg. The highlighted row of the input (output) matrix represents a row that is uniform conditioned on the respective row of the other input (output) matrix. The same sample from the source is fed to all instances of IPMerg used by the construction.

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.

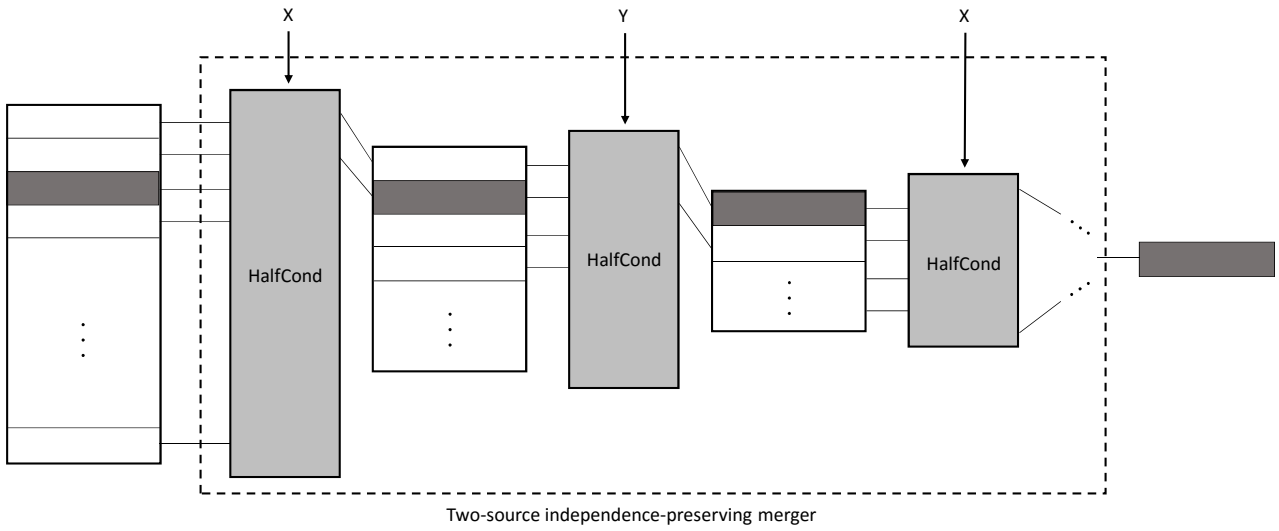


Figure 4: An illustration of the two-source independence-preserving merger that is based on IPHalfCond. Note that the sources X, Y are used in an alternating fashion. Each highlighted row represents a row that is uniform conditioned on the respective row in the other matrix.

Of course, one can invoke 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 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 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 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 $O(r \log n)$.

The formal construction and analysis of IPHalfCond and the merger induced by it appear in Section 5. In that section we actually show that one does not need two fresh sources for the merging process, and one of the two sources can correlate with the matrices to which we apply the merger. Therefore, in Section 5 we refer to the merger described above as a single-source independence-preserving merger. For simplicity of presentation, in this proof overview we do not take advantage of this fact, and consider this merger as a two-source independence-preserving merger. In the next section we use the latter merger to construct a multi-source independence-preserving merger that has a lower min-entropy requirement.

2.4.2 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 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 2.4.1. 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 (see Theorem 6.3), 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

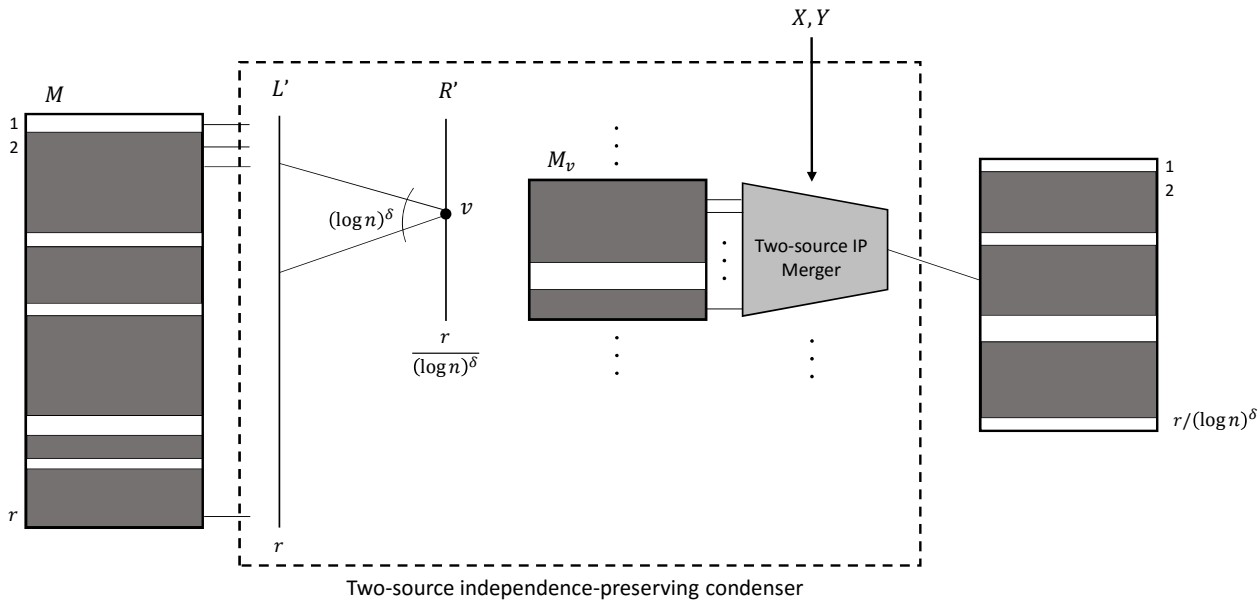


Figure 5: A two-source independence-preserving condenser. An expander graph is used to generate a sequence of $r/(\log n)^\delta$ matrices, each having $(\log n)^\delta$ rows (ignoring constant factors). The two-source independence-preserving merger from Section 2.4.1 is applied to each matrix in the sequence, using the same two sources X, Y. The outputs are then stacked as the rows of the output matrix. Each highlighted row represents a row that is uniform conditioned on the respective row in the other matrix.

independence-preserving merger described in Section 2.4.1 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.

2.5 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:

Independence in X. The random variable X is uniform even conditioned on X'; or

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'.

2.5.1 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 2.1. *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 2.1 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 2.1, 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 2.1 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.

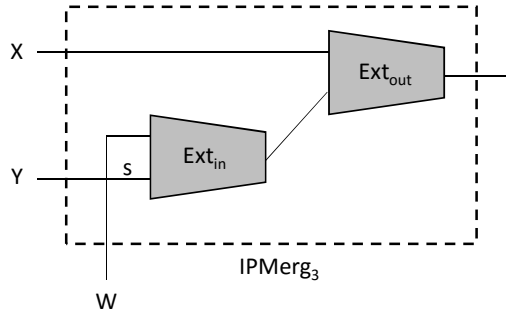
A revised idea would be to use the fresh source W as a "buffer" between X,Y, and define

$$\mathsf{IPMerg}_2(X,Y,W) = \mathsf{Ext}_{\mathsf{out}}(X, \mathsf{Ext}_{\mathsf{in}}(W,Y)).$$

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 2.2. *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 2.2 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. The attentive reader is referred to Section B for a proof sketch of Claim 2.2.

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 2.2, 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.

2.5.2 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

- 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 $\mathsf{a}(Y', Z, W)$.

- 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

$$\begin{aligned} X_{\mathsf{new}} &= \mathsf{b}(X, Z, W), \\ X'_{\mathsf{new}} &= \mathsf{b}(X', Z, W), \\ Y_{\mathsf{new}} &= \mathsf{a}(Y, Z, W), \\ Y'_{\mathsf{new}} &= \mathsf{a}(Y', Z, W), \end{aligned}$$

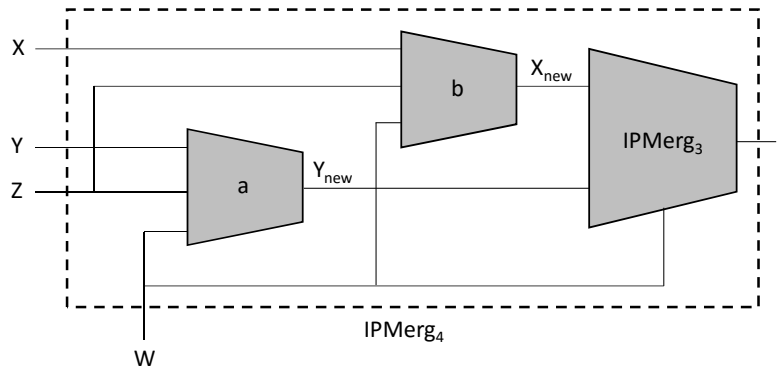
we see that if there is independence in X or in Y then one of the following holds:

1. X_{new} is close to uniform even conditioned on $Y_{\mathsf{new}}, X'_{\mathsf{new}}, Y'_{\mathsf{new}}$; or
2. Y_{new} is close to uniform even conditioned on Y'_{new}.

and furthermore, each of X_{new}, Y_{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. In particular, the proof sketch for Claim 2.2 that is given in Appendix B breaks.



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 simultaneously. So, in order to continue with the analysis we must present our construction for a, b.

2.5.3 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

$$\begin{aligned}\mathsf{a}&\colon \{0,1\}^{\ell} \times \{0,1\}^{n} \to \{0,1\}^{\ell},\\ \mathsf{b}&\colon \{0,1\}^{\ell} \times \{0,1\}^{n} \times \{0,1\}^{n} \to \{0,1\}^{\ell},\end{aligned}$$

by

$$\begin{aligned}\mathsf{a}(X,W) &= \mathsf{Ext}(W, X|_s),\\ \mathsf{b}(X,Z,W) &= \mathsf{Ext}(W, \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, X|_s))).\end{aligned}$$

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. The proof of (a formal restatement of) Claim 2.3 appears in Section 5.1.1.

Claim 2.3. *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 $\mathcal{Z}, \mathcal{A}, \mathcal{M}$.*

2.5.4 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 $\delta > 0$ (see Theorem 3.10). 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.

Analyzing the independence in X case. By Claim 2.3, 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.

Analyzing the independence in Y case. By Claim 2.3, $\mathsf{a}(Y, W)$ 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 2.3, $\mathsf{b}(X, Z, W)$ 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 $\mathsf{IPMerg}(X', Y', Z, W), \mathcal{H}''$.

2.6 Improving the output length and the error guarantee

In this section we give a brief account for the proof of Theorem 1.2. The construction starts by following the three steps that were described in the previous sections, though with two differences. First, instead of setting t to be a large constant that depends only on the desired error guarantee, we set $t = (\log n)^{\Omega(\delta)}$. Second, we do not require that the number of bad variables will be very small (that is, smaller than $\sqrt{r}$, with $r = \text{poly}(n)$ being the number of variables in the sequence). In fact, for what follows it suffices that 0.9 fraction of the variables are good. This allows for a simpler implementation of Step 1 that requires only 2 sources rather than a large constant that depends on the parameters of Bourgain's extractor (see Lemma 8.6).

At this point we obtain a sequence $X_1, \ldots, X_r$ such that all but 0.1 fraction of the variables are t-wise independent. Instead of applying the majority function, as was done in the proof of Theorem 1.1, we follow [Li13a] and perform a sequence of steps, where in each step we reduce the number of variables while maintaining the independence guarantee. To this end we make use of Li's condenser that is based on the lightest bin protocol of Feige [Fei99] (see Theorem 8.3).

By consuming a source with min-entropy $O(t \log n) = O((\log n)^{1+\delta})$, Li's condenser is able to transform the original sequence to a sequence of roughly $t \cdot r^{1/\sqrt{t}}$ variables that have the same guarantee as the input sequence. That is, 0.9 fraction of the variables in the output sequence are t-wise independent. By repeating this process for $O(1/\delta)$ steps, consuming one source per step, we obtain a sequence of $(\log n)^{O(\delta)}$ variables, which can then be merged, using a primitive called a merger with weak-seeds [Coh15a], by consuming one more source (see Theorem 8.5).

For a formal treatment of the ideas and tools that were presented in the section, we refer the reader to Section 8.

3 Preliminaries

In this section we set some notations that will be used throughout the paper and recall some of the results from the literature that we apply frequently.

Setting some standard notations. Unless stated otherwise, the logarithm in this paper is always taken base 2. For every natural number $n \geq 1$, define $[n] = \{1, 2, \ldots, n\}$. Throughout the paper, whenever possible, we avoid the use of floor and ceiling in order not to make the equations cumbersome. Whenever we say that a function is efficiently-computable we mean that the corresponding family of functions can be computed by a (uniform) algorithm that runs in polynomial-time in the input length.

In our proofs we consider sequences of matrices. We use the superscript to refer to a matrix in the sequence and the subscript to refer to a row of the matrix. That is, m^i_j denotes the j'th row of the i'th matrix in the sequence $\{m_k\}_k$. All the matrices consider in this paper are over $\{0, 1\}$.

Random variables and distributions. We sometimes abuse notation and syntactically treat random variables and their distribution as equal, specifically, we denote by U_m a random variable that is uniformly distributed over $\{0,1\}^m$. Furthermore, if U_m appears in a joint distribution (U_m, X) then U_m is independent of X. When m is clear from context, we omit it from the subscript and write U. The support of a random variable X is denoted by $\mathsf{supp}(X)$.

Let X, Y be two random variables. We say that Y is a *deterministic function of* X if the value of X determines the value of Y. Namely, there exists a function f such that $Y = f(X)$. Let $X, Y, Z_1, \ldots, Z_r$ be random variables.

Statistical distance. The *statistical distance* between two distributions X, Y on the same domain D is defined by $\mathsf{SD}(X,Y) = \max_{A \subseteq D} \{|\mathbf{Pr}[X \in A] - \mathbf{Pr}[Y \in A]|\}$. If $\mathsf{SD}(X,Y) \leq \varepsilon$ we write $X \approx_\varepsilon Y$ and say that X and Y are ε-close.

We make frequent use of the following lemmas.

Lemma 3.1. *Let X, X' be random variables on a common domain, and let f be a function on that domain. Then, $\mathsf{SD}(f(X), f(X')) \leq \mathsf{SD}(X, X')$.*

Lemma 3.2. *Let X, X' be two random variables on the same domain. Let Y, Z be a random variables such that for any $y \in \mathsf{supp}(Y)$, the random variables $X \mid Y = y$, $Z \mid Y = y$ are independent and the random variables $X' \mid Y = y$, $Z \mid Y = y$ are independent. Then,*

$$\mathsf{SD}((X,Y),(X',Y)) = \mathsf{SD}((X,Z,Y),(X',Z,Y)).$$

Min-entropy. The *min-entropy* of a random variable X, denoted by $H_\infty(X)$, is defined by $H_\infty(X) = \min_{x \in \mathsf{supp}(X)} \log_2(1/\mathbf{Pr}[X = x])$. If X is supported on $\{0,1\}^n$, we define the *min-entropy rate* of X by $H_\infty(X)/n$. In such case, if X has min-entropy k or more, we say that X is an (n,k)-source. When wish to refer to an (n,k)-source without specifying the quantitative parameters, we sometimes use the standard terms *source* or *weak-source*.

Average conditional min-entropy. Let X, W be two random variables. The *average conditional min-entropy* of X given W is defined as

$$\widetilde{H}_\infty(X \mid W) = -\log_2 \left(\mathop{\mathbf{E}}_{w \sim W} \left[2^{-H_\infty(X|W=w)} \right] \right).$$

Lemma 3.3 ([DORS08])**.** *Let X, Y, Z be random variables such that Y has support size at most 2^ℓ. Then,*

$$\widetilde{H}_\infty(X \mid (Y,Z)) \geq \widetilde{H}_\infty((X,Y) \mid Z) - \ell \geq \widetilde{H}_\infty(X \mid Z) - \ell.$$

In particular, $\widetilde{H}_\infty(X \mid Y) \geq H_\infty(X) - \ell$.

Lemma 3.4 ([DORS08])**.** *For any two random variables X, Y and any $\varepsilon > 0$, it holds that*

$$\mathop{\mathbf{Pr}}_{y \sim Y} \left[H_\infty(X \mid Y = y) < \widetilde{H}_\infty(X \mid Y) - \log(1/\varepsilon) \right] \leq \varepsilon.$$

Lemma 3.5. *Let X, Y, Z be random variables such that for any $y \in \mathsf{supp}(Y)$ it holds that $X \mid Y = y$ and $Z \mid Y = y$ are independent. Then, $\widetilde{H}_\infty(X \mid (Y,Z)) = \widetilde{H}_\infty(X \mid Y)$. In particular, if X and Z are independent then $\widetilde{H}_\infty(X \mid Z) = H_\infty(X)$.*

We also need the following standard definition of error correcting codes.

Definition 3.6. *Let Σ be some set. A mapping $\mathsf{ECC}\colon \Sigma^k \to \Sigma^n$ is called an error correcting code with relative-distance δ if for any $x, y \in \Sigma^k$, it holds that the Hamming distance between $\mathsf{ECC}(x)$ and $\mathsf{ECC}(y)$ is at least δn. The rate of the code, denoted by ρ, is defined by $\rho = k/n$. We say that the alphabet size of the code is $|\Sigma|$.*

Extractors. We provide standard definitions of extractors and state some of the results we use.

Definition 3.7 (Seeded extractors)**.** *A function $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is called a* seeded extractor *for min-entropy k with error guarantee ε if for any (n,k)-source X it holds that $\mathsf{Ext}(X,S) \approx_\varepsilon U_m$, where S is uniformly distributed over $\{0,1\}^d$ and is independent of X. We say that Ext is a* strong seeded-extractor *if $(\mathsf{Ext}(X,S),S) \approx_\varepsilon U_{m+d}$.*

Definition 3.8 (Multi-source extractors)**.** *A function $\mathsf{Ext}\colon (\{0,1\}^n)^b \to \{0,1\}^m$ is called an extractor for min-entropy k with error guarantee ε if for any independent (n,k)-sources $X_1, \ldots, X_b$, it holds that $\mathsf{Ext}(X_1, \ldots, X_b) \approx_\varepsilon U_m$. The extractor Ext is sometimes referred to as a b-source extractor.*

We sometimes say that an extractor Ext *supports* min-entropy k. By that we mean that Ext is an extractor for min-entropy k. Throughout the paper we make use of the following explicit strong seeded extractors.

Theorem 3.9 ([GUV09])**.** *There exists a universal constant $c > 0$ such that the following holds. For all positive integers n, k and $\varepsilon > 0$, there exists an efficiently-computable strong seeded-extractor $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ for min-entropy k, with error guarantee ε, seed length $d = c \cdot \log(n/\varepsilon)$, and $m = k/2$ output bits.*

The following theorem readily follows by Theorem 4 in [Raz05] and Theorem 3.9.

Theorem 3.10. *There exist universal constants c', c'' such that the following holds. Let n, k be integers and let $\varepsilon > 0$. Set $d = c' \cdot \log(n/\varepsilon)$. For all $k \geq c''d$, there exists an efficiently-computable function $\mathsf{Raz}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{k/2}$ with the following property. Let X be an (n,k)-source, and let Y be an independent $(d, 0.6d)$-source. Then, $(\mathsf{Raz}(X,Y),Y) \approx_\varepsilon (U,Y)$.*

A strong seeded extractor Ext has the guarantee that $(\mathsf{Ext}(W,S),S)$ is close to uniform whenever W has a sufficient amount of min-entropy and is independent of the uniform seed S. Throughout the paper, however, we will typically apply strong seeded extractors to a source and a seed that are functions of a common set of random variables. In particular, the source and the seed will not be independent. Nevertheless, conditioned on the "history"

of the computation led to these source and seed, independence does hold. Moreover, some further technical issues arise – the source does not have high min-entropy conditioned on any fixing of this history, and the seed is not uniform but rather close to uniform. The following lemma allows one to avoid facing these issues again and again, and so we find it very useful. A proof for Lemma 3.11 appears in Appendix A. We remark that a simple proof for Lemma 3.11 can be found if one is willing to replace the error in the lemma with $2\delta + 2\varepsilon$ (as apposed to $\delta + 2\varepsilon$). However, this loss can be avoided with some more work following [Rey11] and references therein.

Lemma 3.11. *Let* $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a strong seeded extractor for min-entropy* k *with error guarantee* ε*. Let* W, S *be random variables over* n*-bit strings and* d*-bit strings, respectively. Let* $\mathcal{H}$ *be some random variable such that*

$$\widetilde{H}_\infty(W \mid \mathcal{H}) \geq k + \log(1/\varepsilon),$$
$$(S, \mathcal{H}) \approx_\delta (U, \mathcal{H}).$$

Assume further that conditioned on $\mathcal{H}$*, the random variables* W, S *are independent. Then,*

$$(\mathsf{Ext}(W,S), S, \mathcal{H}) \approx_{\delta+2\varepsilon} (U, S, \mathcal{H}).$$

4 Generating a Sequence of Somewhere-Independent Matrices

In this section we show how to transform $O(1)$ sources with min-entropy $\widetilde{O}(t) \cdot \log n$ to a sequence of matrices such that all but few of the matrices are "somewhere independent". We start by formally defining this notion of independence.

Definition 4.1 (Somewhere-independent matrices)**.** *Let* $M, M^1, \ldots, M^t$ *be random variables in the form of* $r \times \ell$ *matrices. Let* $\mathcal{H}$ *be a random variable and let* $\delta > 0$*. We say that* M *is* $(\delta, \mathcal{H})$-somewhere independent of $M^1, \ldots, M^t$ *if the following holds:*

- *There exists* $g \in [r]$ *such that*

$$\left(M_g, \{M_g^i\}_{i=1}^t, \mathcal{H}\right) \approx_\delta \left(U, \{M_g^i\}_{i=1}^t, \mathcal{H}\right).$$

- *For any* $j \in [r]$, $(M_j, \mathcal{H}) \approx_\delta (U, \mathcal{H})$.

Furthermore, we set the following notations:

- *For any* $g \in [r]$ *as in the first item, we say that* M *is* $(\delta, \mathcal{H})$-independent of $M^1, \ldots, M^t$ at g.
- *For* $\alpha \in [0,1]$*, we say that* M *is* $(\alpha, \delta, \mathcal{H})$-independent of $M^1, \ldots, M^t$ *if there are* αr *elements* $g \in [r]$ *for which* M *is* $(\delta, \mathcal{H})$*-independent of* $M^1, \ldots, M^t$ *at* g.

- *If $\mathcal{H}$ is empty then we omit it from the notations set above.*

We remark that the somewhat technical presence of the random variable $\mathcal{H}$ in Definition 4.1 enables us to reuse the same source in our construction several times. The main result of this section is the following proposition.

Proposition 4.2. *There exists a universal constant $\alpha > 0$ such that the following holds. For all integers n, t, any $\varepsilon > 0$, and for any $\delta > 0$, there exists an efficiently-computable function*

$$\mathsf{GenSISeq}\colon (\{0,1\}^n)^b \to \left(\{0,1\}^{r' \times m}\right)^r,$$

with $b = 7/\alpha$, $r = n^{3/\alpha}$, $r' = O(t \log n)$, and $m = \Omega(\log(n/\varepsilon))$, such that the following holds. Let $X_1, \ldots, X_b$ be independent (n,k)-sources with

$$k = \Omega\left(t \log(t) \log\left(\frac{n \log t}{\varepsilon}\right)\right).$$

Then, there exists $B \subset [r]$, of size $|B| \leq r^{1/2-\alpha}$, and a sequence $M^1, \ldots, M^r$ of random variables in the form of $r' \times m$ matrices such that the following holds:

- *The sequence $\mathsf{GenSISeq}(X_1, \ldots, X_b)$ is r^{-1}-close to $(M^1, \ldots, M^r)$; and*
- *For any $g \in [r] \setminus B$ and any $i_1, \ldots, i_t \in [r] \setminus \{g\}$, it holds that M^g is $(0.9, \varepsilon)$-independent of $\{M^{i_j}\}_{j=1}^t$.*

The proposition readily follows by the following two lemmas.

Lemma 4.3. *There exists a universal constant $\alpha > 0$ such that the following holds. For all integers n, ℓ, there exists an efficiently-computable function*

$$f\colon (\{0,1\}^n)^b \to \{0,1\}^{r \times \ell}$$

with $b = 7/\alpha$ and $r = n^{3/\alpha}$, such that the following holds. Let $X_1, \ldots, X_b$ be independent (n,k)-sources with $k = \Omega(\ell) + (3/\alpha^2) \log n$. Assume further that $k = n^{o(1)}$. Then, there exists $B \subset [r]$ of size $|B| \leq r^{1/2-\alpha}$ and a sequence $Z_1, \ldots, Z_r$ of ℓ-bit random variables such that the following holds:

- *$f(X_1, \ldots, X_b)$ is r^{-1}-close to $(Z_1, \ldots, Z_r)$.*
- *For any $j \in [r] \setminus B$, Z_j is uniformly distributed over $\{0,1\}^\ell$.*

Lemma 4.4. *For all integers r, ℓ, n, t, and for any $\varepsilon > 0$ such that*

$$\ell = \Omega\left(t \log(t) \cdot \log\left(\frac{n \log t}{\varepsilon}\right)\right),$$

there exists an efficiently-computable function

$$h\colon \{0,1\}^{r \times \ell} \times \{0,1\}^n \to \left(\{0,1\}^{r' \times m}\right)^r,$$

with $m = \Omega(\ell/(t \log t))$ *and* $r' = O(t \log r)$ *such that the following holds. Let* M *be a random variable in the form of an* $r \times \ell$ *matrix, and let* X *be an independent* (n, k)*-source with*

$$k = \Omega\left(t \log(t) \cdot \log\left(\frac{\ell \log t}{\varepsilon}\right)\right).$$

Then, for any $g \in [r]$ *such that* M_g *is uniform and for any* $i_1, \ldots, i_t \in [r] \setminus \{g\}$*, the matrix* $h(M, X)^g$ *is* $(0.9, \varepsilon)$*-independent of* $h(M, X)^{i_1}, \ldots, h(M, X)^{i_t}$*.*

The proofs for Lemma 4.3 and Lemma 4.4 use distinct set of tools and so in Section 4.1 and Section 4.2 we present the relevant tools for each of the lemmas, followed by the corresponding proof.

4.1 Proof of Lemma 4.3

For the proof of Lemma 4.3 we make use of Bourgain's two-source extractor [Bou05] and the lossless condenser of Guruswami *et al.* [GUV09].

Theorem 4.5 ([Bou05])**.** *There exists a universal constant* $\beta > 0$ *such that for any integer* n*, there exists an efficiently-computable two-source extractor*

$$\mathsf{Bour}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$$

for min-entropy $(1/2 - \beta)n$ *with error guarantee* $2^{-\Omega(n)}$ *and* $m = \Omega(n)$ *output bits.*

Definition 4.6 (Seeded condensers)**.** *A function* $\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is said to be a* $k \to_\varepsilon k'$ condenser *if for any* (n, k)*-source* X *and for any independent random variable* S *that is uniformly distributed over* d*-bit strings, it holds that* $\mathsf{Cond}(X, S)$ *is* ε*-close to a random variable with min-entropy* k'*. The function* Cond *is called a* lossless condenser *if* $k' = k + d$*.*

Theorem 4.7 ([GUV09])**.** *For any constant* $\tau > 0$ *(*τ *can be taken to be larger than* 1*), all integers* n, k *such that* $k \leq n$*, and for any* $\varepsilon > 0$*, there exists an efficiently-computable* $k \to_\varepsilon k + d$ *lossless condenser*

$$\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

having seed length $d = (1 + 1/\tau)\log(nk/\varepsilon) + O(1)$ *and* $m = 2d + (1 + \tau)k$ *output bits.*

For the proof of Lemma 4.3, we need the following lemma which, informally speaking, states that any seeded condenser is also strong. A similar lemma, with slightly weaker parameters, appears in [Li11].

Lemma 4.8. *Let* $\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a* $k \to_\varepsilon k'$ *condenser. Let* X *be an* (n, k)*-source and let* S *be an independent random variable that is uniformly distributed over* d*-bit strings. Then, for any* $\delta > 0$*, with probability* $1 - \delta$ *over* $s \sim S$ *it holds that* $\mathsf{Cond}(X, s)$ *is* $(2\varepsilon/\delta)$*-close to having min-entropy* $k' - d - \log(2/\delta)$*.*

Proof. Let $f(X,S)$ be a random variable with min-entropy k' that is ε-close to $\mathsf{Cond}(X,S)$. By Lemma 3.3,

$$\widetilde{H}_\infty\left(f(X,S) \mid S\right) \geq H_\infty\left(f(X,S)\right) - |S| \geq k' - d.$$

Thus, by Lemma 3.4, except with probability $\delta/2$ over $s \sim S$, it holds that $H_\infty(f(X,s)) \geq k' - d - \log(2/\delta)$. Now, by averaging, as $f(X,S) \approx_\varepsilon \mathsf{Cond}(X,S)$, we have that except with probability $\delta/2$ over $s \sim S$, $\mathsf{Cond}(X,s)$ is $(2\varepsilon/\delta)$-close to $f(X,s)$. Thus, by the union bound, except with probability δ over $s \sim S$, it holds that $\mathsf{Cond}(X,s)$ is $(2\varepsilon/\delta)$-close to a random variable with min-entropy $k' - d - \log(2/\delta)$, as stated. □

We also make use of the following lemmas.

Lemma 4.9 ([BIW06])**.** *Let $X_1, \ldots, X_c$ be independent random variables over $\{0,1\}^n$ such that $\mathsf{SD}(X_i, U_n) \leq \varepsilon$ for all $i \in [c]$. Then, $\mathsf{SD}(X_1 \oplus \cdots \oplus X_c, U_n) \leq \varepsilon^c$.*

Lemma 4.10 ([Li12b])**.** *Let X, X' be random variables with a common range such that $\mathsf{SD}(X, X') \leq \varepsilon$. Let (X,Y) be a joint distribution. Then, there exists a joint distribution (X',Y) such that $\mathsf{SD}((X,Y),(X',Y)) \leq \varepsilon$.*

Proof of Lemma 4.3. We start by describing the construction of f and then turn to the analysis. Let β be the universal constant from Theorem 4.5. Fix $i \in [b]$. Set

$$\begin{aligned}\tau &= 1 + 2\beta \\ \alpha &= \beta/16 \\ \varepsilon &= n^{-1/\alpha} \\ b &= 7/\alpha.\end{aligned}$$

Given inputs $x_1, \ldots, x_b \in \{0,1\}^n$ we define $f(x_1, \ldots, x_b)$ as follows. For $i \in [b]$, let m^i denote the $r \times m$ matrix obtained by applying the lossless condenser Cond from Theorem 4.7 with τ as defined above to each possible seed. By Theorem 4.7, we have that $r = O((nk/\varepsilon)^{1+1/\tau})$ and $m = 2d + (1+\tau)k$. For an odd $i \in [b]$, let $m^{i,i+1}$ be the $r \times m'$ matrix that is defined as follows. For $j \in [r]$, the j'th row of $m^{i,i+1}$ is given by $m^{i,i+1}_j = \mathsf{Bour}\left(m^i_j, m^{i+1}_j\right)$. By Theorem 4.5, $m' = \Omega(m)$. Lastly, we define the j'th row of $f(x_1, \ldots, x_b)$ by

$$f(x_1, \ldots, x_b)_j = \bigoplus_{\text{odd } i=1}^{b} m^{i,i+1}_j.$$

We now turn to the analysis. Fix $i \in [b]$. By Theorem 4.7, together with Lemma 4.8, except with probability $\delta = r^{-(1/2+\alpha)}$ over $s \sim U_d$, it holds that $\mathsf{Cond}(X_i, s)$ is ε'-close to an (m, k')-source, where $k' = k + d - \log(2/\delta) \geq k - d$, and $\varepsilon' = 2\varepsilon/\delta$.

Claim 4.11. $\varepsilon' \leq r^{-\alpha/3}$.

Proof. By Theorem 4.7 and as $\tau > 1$,

$$r = O\left(\left(\frac{nk}{\varepsilon}\right)^{1+1/\tau}\right) \leq O\left(k^2\right) \cdot \left(\frac{1}{\varepsilon}\right)^{\frac{(1+\alpha)(2+32\alpha)}{1+32\alpha}}. \tag{4.1}$$

Thus,

$$\frac{1}{\delta} = r^{1/2+\alpha} \leq O\left(k^2\right) \cdot \left(\frac{1}{\varepsilon}\right)^{\left(\frac{1}{2}+\alpha\right)\cdot\frac{(1+\alpha)(2+32\alpha)}{1+32\alpha}}.$$

Therefore,

$$\varepsilon' = \frac{2\varepsilon}{\delta} \leq O\left(k^2\right) \cdot \varepsilon^{1-\left(\frac{1}{2}+\alpha\right)\cdot\frac{(1+\alpha)(2+32\alpha)}{1+32\alpha}} \leq O\left(k^2\right) \cdot \varepsilon^{2\alpha} \leq \varepsilon^{\alpha},$$

where the penultimate inequality can be easily shown to hold for all $\alpha \leq 1/16$ (recall that $\alpha = \beta/16 \leq 1/16$), and the last inequality follows as $k = n^{o(1)} = (1/\varepsilon)^{o(1)}$ and since α is constant. Now, by Equation (4.1), together with $k = (1/\varepsilon)^{o(1)}$, it holds that $r \leq (1/\varepsilon)^3$, and so $\varepsilon' \leq r^{-\alpha/3}$. □

We proceed to verify that

Claim 4.12. $k - d \geq (1/2 - \beta)m$.

Proof. Recall that $m = 2d + (1+\tau)k$, and so a simple calculation can be used to show that the claim follows given that $k \geq d/\alpha$. As was shown in Claim 4.11, $r \leq (1/\varepsilon)^3$ and so, by our choice of ε, we get that $d \leq (3/\alpha)\log n$. Therefore, it suffices to show that $k \geq (3/\alpha^2)\log n$, which follows by the hypothesis of the lemma. □

By Claim 4.11 and Claim 4.12, there exists a set $B_i \subset [r]$ of size $|B_i| \leq \delta r = r^{1/2-\alpha}$ such that for any $j \notin B_i$ it holds that M_j^i is $r^{-\alpha/3}$-close to having min-entropy rate $1/2 - \beta$. Consider any odd $i \in [b]$. By Theorem 4.5, for any $j \notin B_i \cup B_{i+1}$, it holds that $M_j^{i,i+1}$ is ε''-close to uniform, where $\varepsilon'' = 2\varepsilon' + 2^{-\Omega(m)} = O(r^{-\alpha/3})$. Thus, by Lemma 4.9 and by our choice of b, for any $j \notin \cup_{i=1}^{b} B_i$, it holds that $f(X_1, \ldots, X_b)_j$ is r^{-2}-close to uniform.

Let $B = \cup_{i=1}^{b} B_i$ and note that $|B| \leq b \cdot r^{1/2-\alpha}$. We now prove the existence of a sequence of random variables $Z_1, \ldots, Z_r$ that is r^{-1}-close to $f(X_1, \ldots, X_b)$, with the property that for any $j \in [r] \setminus B$ it holds that Z_j is uniform. This is done by applying Lemma 4.10 for $r - |B|$ times. In each application, we replace one random variable that is r^{-2}-close to uniform by a truly uniform variable, while keeping the marginal distribution of the other variables intact. To conclude the proof, we note that the multiplicative constant factor of b in the size of B can be avoided by setting α to a slightly smaller value. □

4.2 Proof of Lemma 4.4

For the proof of Lemma 4.4 we make use of two main building blocks, first of which are Goppa codes (or algebraic-geometric codes). These are error correcting codes that come close to the Singleton bound using a surprisingly small alphabet size (see Theorem 4.13 below). Second, we make use of correlation breakers with advice, which were first implicitly constructed by [CGL15] (see Theorem 4.15) and explicitly defined in [Coh15b] (see Definition 4.14).

Theorem 4.13 ([GS95] (see also [Sti09])). *Let p be any prime number and let m be an even integer. Set $q = p^m$. For every $\rho \in [0,1]$ and for any large enough integer n, there exists an efficiently-computable rate ρ linear error correcting code $\mathsf{ECC}\colon \mathbb{F}_q^{\rho n} \to \mathbb{F}_q^n$ with relative distance δ such that*

$$\rho + \delta \geq 1 - \frac{1}{\sqrt{q}-1}.$$

Definition 4.14. *For an integer $t \geq 1$ a* t-correlation-breaker with advice *for min-entropy k and error guarantee ε is a function*

$$\mathsf{AdvCB}\colon \{0,1\}^\ell \times \{0,1\}^n \times \{0,1\}^a \to \{0,1\}^m$$

with the following property. Let $X, X^1, \ldots, X^t$ be ℓ-bit random variables such that X is uniform. Let W be an (n,k)-source that is independent of the joint distribution of $X, X^1, \ldots, X^t$. Then, for any $\alpha, \alpha^1, \ldots, \alpha^t \in \{0,1\}^a$ such that $\alpha \notin \{\alpha^i\}_{i=1}^t$, it holds that

$$\left(\mathsf{AdvCB}(X,W,\alpha), \{\mathsf{AdvCB}(X^i,W,\alpha^i)\}_{i=1}^t\right) \approx_\varepsilon \left(U, \{\mathsf{AdvCB}(X^i,W,\alpha^i)\}_{i=1}^t\right).$$

The third argument to the function AdvCB *is called the* advice.

Theorem 4.15 ([CGL15]). *For all integers ℓ, n, a, t and for any $\varepsilon > 0$ such that*

$$\ell = \Omega\left(at \cdot \log\left(\frac{an}{\varepsilon}\right)\right),$$

there exists an efficiently-computable t-correlation-breaker with advice $\mathsf{AdvCB}\colon \{0,1\}^\ell \times \{0,1\}^n \times \{0,1\}^a \to \{0,1\}^m$ for min-entropy

$$k = \Omega\left(at \cdot \log\left(\frac{a\ell}{\varepsilon}\right)\right),$$

with error guarantee ε, and $m = \Omega(\ell/(at))$ output bits.

With these results in hand, we are ready to prove Lemma 4.4.

Proof of Lemma 4.4. We first describe the construction of h and then turn to the analysis. We make use of the following building blocks:

- Set q to the least even power of two that is larger than $(20t+1)^2$. Identify $[r]$ with some arbitrary subset of $\mathbb{F}_q^{\log_2 r}$. Let $\mathsf{ECC}\colon \mathbb{F}_q^{\log_2 r} \to \mathbb{F}_q^{r'}$ be the error correcting code given by Theorem 4.13 set with relative distance $\delta = 1 - 1/(10t)$. By Theorem 4.13, such an explicit code exists with rate $\rho \geq 1/(20t)$, and so $r' \leq 20t \log_2 r$.
- Set $a = \log_2 q$ and note that $a = O(\log t)$. Let $\mathsf{AdvCB}\colon \{0,1\}^\ell \times \{0,1\}^n \times \{0,1\}^a \to \{0,1\}^m$ be the t-advice correlation breaker given by Theorem 4.15 for min-entropy k with error guarantee ε. Note that our assumption on k, ℓ and our choice of a suffice for the hypothesis of Theorem 4.15 to hold with $m = \Omega(\ell/(t \log t))$.

Let $m \in \{0,1\}^{r \times \ell}$ and $w \in \{0,1\}^n$. For $i \in [r]$ and $j \in [r']$, we define row j of the matrix $h(m,w)^i$ by

$$h(m,w)^i_j = \mathsf{AdvCB}\left(m^i, w, \mathsf{ECC}(i)_j\right).$$

With the construction in hand, we turn to the analysis. Let M be a random variable in the form of an $r \times \ell$ matrix, and let W be an independent (n,k)-source. Let $g \in [r]$ be such that M_g is uniform and consider any $i_1, \ldots, i_t \in [r] \setminus \{g\}$. Consider some fixed $v \in [t]$. By our choice of δ, the codeword $\mathsf{ECC}(g)$ agrees with $\mathsf{ECC}(i_v)$ on at most $1/(10t)$ fraction of the entries. Thus, there exists $B \subseteq [r']$ of size $|B| \leq r'/10$ such that for any $j \notin B$, it holds that $\mathsf{ECC}(g)_j \notin \{\mathsf{ECC}(i_v)_j\}_{v=1}^t$. As M_g is uniform, Theorem 4.15 implies that for any $j \notin B$,

$$\left(h(M,W)^g_j, \left\{h(M,W)^{i_v}_j\right\}_{v=1}^t\right) \approx_\varepsilon \left(U, \left\{h(M,W)^{i_v}_j\right\}_{v=1}^t\right).$$

As this holds for any $j \notin B$, and since $|B| \leq r'/10$, we have that $h(M,W)^g$ is $(0.9,\varepsilon)$-independent of $\{h(M,W)^{i_v}\}_{v=1}^t$, as stated. □

5 Single-Source Independence-Preserving Mergers for High Min-Entropy

In this section we construct an independence-preserving merger that uses a single auxiliary source of randomness. Unfortunately, the min-entropy required by the source will be too high so to prevent us from using this merger directly to the sequence of matrices obtained by the function $\mathsf{GenSISeq}$ given in Proposition 4.2. Nevertheless, the merger that we develop here will be used as a building block in the multi-source merger that is presented in Section 6.

Proposition 5.1. *For all integers n, r, and for any $\varepsilon > 0$, there exists an efficiently-computable function*

$$\mathsf{IPMerg}\colon \{0,1\}^{r \times s} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^s,$$

where $s = \Theta(\log(n/\varepsilon))$ such that the following holds. Let $\mathcal{M} = (M, M^1, \ldots, M^t)$ be a sequence of random variables in the form of $r \times s$ matrices. Assume that M is $(\delta, \mathcal{H})$-somewhere-independent of $M^1, \ldots, M^t$. Let Z, W be n-bit random variables such that conditioned on $\mathcal{H}$, the joint distribution of $\mathcal{M}, Z$ is independent of W. Assume further that

$$\widetilde{H}_\infty(W \mid \mathcal{H}) = \Omega\left(t^2 rs\right),$$
$$\widetilde{H}_\infty(Z \mid \mathcal{H}) = \Omega\left(t^2 rs\right).$$

Then,

$$\left(\mathsf{IPMerg}(M,Z,W), \left\{\mathsf{IPMerg}(M^i,Z,W)\right\}_{i=1}^t\right) \approx_{O(r(\delta+\varepsilon))} \left(U, \left\{\mathsf{IPMerg}(M^i,Z,W)\right\}_{i=1}^t\right).$$

We remark that although syntactically-wise IPMerg is given two n-bit strings as input, which we think of as sampled from the sources Z and W, we consider IPMerg as using only a single auxiliary source of randomness. This is as, according to Proposition 5.1, the source Z may correlate with $\mathcal{M}$, and only W is required to be a "fresh" source of randomness.

For the proof of Proposition 5.1 we introduce and construct an object called an *independence-preserving half condenser*. This is the content of the following section. With the half condenser in hand, in Section 5.2 we can go ahead and prove Proposition 5.1.

5.1 Independence-preserving half-condensers

The main result of this section is the following lemma.

Lemma 5.2. *For all integers n, r, and for any $\varepsilon > 0$, there exists an efficiently-computable function*

$$\mathsf{IPHalfCond}\colon \{0,1\}^{r\times s} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{(r/2)\times s},$$

with $s = O(\log(n/\varepsilon))$, such that the following holds. Let $\mathcal{M} = (M, M^1, \ldots, M^t)$ be a sequence of random variables in the form of $r \times s$ matrices. Assume that M is $(\delta, \mathcal{H})$-somewhere-independent of $M^1, \ldots, M^t$. Let Z, W be n-bit random variables such that conditioned on $\mathcal{H}$, the joint distribution of $\mathcal{M}, Z$ is independent of W. Assume further that

$$\widetilde{H}_\infty(W \mid \mathcal{H}) = \Omega\left(t^2 rs\right),$$
$$\widetilde{H}_\infty(Z \mid \mathcal{H}) = \Omega\left(t^2 rs\right).$$

Let $M' = \mathsf{IPHalfCond}(M, Z, W)$ and $(M')^i = \mathsf{IPHalfCond}(M^i, Z, W)$. Then, there exists a random variable $\mathcal{H}'$ such that the following holds:

- *M' is $(2\delta + 12\varepsilon, \mathcal{H}')$-somewhere-independent of $\{(M')^i\}_{i=1}^t$.*
- *Conditioned on $\mathcal{H}'$, the random variables W, M', $\{(M')^i\}_{i=1}^t$ are jointly independent of Z.*
- *$\widetilde{H}_\infty(W \mid \mathcal{H}') \geq \widetilde{H}_\infty(W \mid \mathcal{H}) - O\left(t^2 rs\right)$.*
- *$\widetilde{H}_\infty(Z \mid \mathcal{H}') \geq \widetilde{H}_\infty(Z \mid \mathcal{H}) - O\left(t^2 rs\right)$.*

5.1.1 Establishing a hierarchy of independence

For the proof of Lemma 5.2, we need to establish a "hierarchy of independence" which is the content of this section. Then, in Section 5.1.2, we prove Lemma 5.2. The result that appears in this section relies on alternating extraction and appears, in slightly different forms, in previous works [DP07, DW09, Li13a, Coh15a].

Let n, b be some integers, and let $\varepsilon > 0$. Let $s = c \cdot \log(n/\varepsilon)$ be a length that suffices for a seed of the strong seeded extractor from Theorem 3.9 when given a sample from an n-bit source and with error guarantee ε. By Theorem 3.9, c is some universal constant. We further assume that $b \geq s$.

Let $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^b$ be the strong seeded extractor from Theorem 3.9 for min-entropy $2b$ set with error guarantee ε. We let $\mathsf{Ext}_s\colon \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^s$ denote the function obtained by applying Ext and taking only the length s prefix of the output. This is valid as we assume $b \geq s$. We define a pair of functions

$$\begin{aligned}\mathsf{a}&\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^b,\\ \mathsf{b}&\colon \{0,1\}^s \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^b,\end{aligned}$$

as follows. For $v \in \{0,1\}^s$, $z \in \{0,1\}^n$, and $w \in \{0,1\}^n$,

$$\begin{aligned}\mathsf{a}(v,w) &= \mathsf{Ext}(w,v),\\ \mathsf{b}(v,z,w) &= \mathsf{Ext}(w, \mathsf{Ext}_s(z, \mathsf{Ext}_s(w,v))).\end{aligned}$$

Lemma 5.3. *Let $\mathcal{M} = (M, M^1, \ldots, M^t)$ be a sequence of random variables in the form of $r \times s$ matrices such that M is $(\delta, \mathcal{H})$-independent of $M^1, \ldots, M^t$ at g. Let W, Z be n-bit random variables such that conditioned on $\mathcal{H}$, the joint distribution of $\mathcal{M}, Z$ is independent of W. Assume further that*

$$\begin{aligned}\widetilde{H}_\infty(W \mid \mathcal{H}) &\geq 3trb,\\ \widetilde{H}_\infty(Z \mid \mathcal{H}) &\geq 3trs.\end{aligned}$$

Write

$$\begin{aligned}\mathcal{A} &= \{\mathsf{a}(M_j, W) \mid j \in [r]\} \cup \left\{\mathsf{a}(M^i_j, W) \mid i \in [t], j \in [r]\right\},\\ \mathcal{Z} &= \{\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_j)) \mid j \in [r]\} \cup \left\{\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M^i_j)) \mid i \in [t], j \in [r]\right\}.\end{aligned}$$

Then, the following holds:

1. $\left(\mathsf{a}(M_g, W), \left\{\mathsf{a}(M^i_g, W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}\right) \approx_{\delta+2\varepsilon} \left(U, \left\{\mathsf{a}(M^i_g, W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}\right)$,
2. $\left(\mathsf{b}(M_g, Z, W), \left\{\mathsf{b}(M^i_g, Z, W)\right\}_{i=1}^t, \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}\right) \approx_{\delta+6\varepsilon} \left(U, \left\{\mathsf{b}(M^i_g, Z, W)\right\}_{i=1}^t, \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}\right)$.

Furthermore, for any $j \in [r]$,

3. $(\mathsf{a}(M_j, W), \mathcal{M}, \mathcal{H}) \approx_{\delta+2\varepsilon} (U, \mathcal{M}, \mathcal{H})$,
4. $(\mathsf{b}(M_j, Z, W), \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}) \approx_{\delta+6\varepsilon} (U, \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H})$.

Lastly,

5. $\widetilde{H}_\infty(Z \mid \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}) \geq \widetilde{H}_\infty(Z \mid \mathcal{H}) - 2(t+1)rs$.
6. $\widetilde{H}_\infty(W \mid \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}) \geq \widetilde{H}_\infty(W \mid \mathcal{H}) - (t+1)rb$.

Proof of Item 1. As M is $(\delta, \mathcal{H})$-independent of $M^1, \ldots, M^t$ at g, we have that

$$\left(M_g, \left\{M_g^i\right\}_{i=1}^t, \mathcal{H}\right) \approx_\delta \left(U, \left\{M_g^i\right\}_{i=1}^t, \mathcal{H}\right).$$

Conditioned on $\left\{M_g^i\right\}_{i=1}^t, \mathcal{H}$, the random variable M_g is independent of the joint distribution of $\left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t$. Hence, by Lemma 3.2,

$$\left(M_g, \left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \left\{M_g^i\right\}_{i=1}^t, \mathcal{H}\right) \approx_\delta \left(U, \left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \left\{M_g^i\right\}_{i=1}^t, \mathcal{H}\right).$$

By Lemma 3.3 and Lemma 3.5,

$$\widetilde{H}_\infty\left(W \mid \left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \left\{M_g^i\right\}_{i=1}^t, \mathcal{H}\right) \geq \widetilde{H}_\infty\left(W \mid \mathcal{H}\right) - tb \geq 2b + \log(1/\varepsilon).$$

The above equation, together with the fact that M_g is independent of W conditioned on $\left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t$, $\left\{M_g^i\right\}_{i=1}^t$, $\mathcal{H}$, enables us to apply Lemma 3.11 and conclude that

$$\left(\mathsf{a}(M_g, W), M_g, \left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \left\{M_g^i\right\}_{i=1}^t, \mathcal{H}\right) \approx_{\delta+2\varepsilon} \left(U, M_g, \left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \left\{M_g^i\right\}_{i=1}^t, \mathcal{H}\right).$$

Conditioned on M_g, $\left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t$, $\left\{M_g^i\right\}_{i=1}^t$, $\mathcal{H}$, the random variable $\mathsf{a}(M_g, W)$ is independent of $\mathcal{M}$. Hence, by Lemma 3.2,

$$\left(\mathsf{a}(M_g, W), \left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}\right) \approx_{\delta+2\varepsilon} \left(U, \left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}\right).$$

This concludes the proof of Item 1. □

Proof of Item 2. By Item 1,

$$\left(\mathsf{a}(M_g, W), \left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}\right) \approx_{\delta+2\varepsilon} \left(U, \left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}\right).$$

As $\mathsf{Ext}_s(W, M_g^i)$ is a prefix of $\mathsf{a}(M_g^i, W)$, conditioned on $\left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}$, the random variable $\mathsf{a}(M_g, W)$ is independent of the joint distribution of $\left\{\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_g^i))\right\}_{i=1}^t$. Thus, by Lemma 3.2,

$$\left(\mathsf{a}(M_g, W), \mathcal{H}_1\right) \approx_{\delta+2\varepsilon} \left(U, \mathcal{H}_1\right),$$

where $\mathcal{H}_1 = \left\{\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_g^i))\right\}_{i=1}^t, \left\{\mathsf{a}(M_g^i, W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}$. Note that conditioned on $\mathcal{H}_1$, the random variable $\mathsf{a}(M_g, W)$ is independent of Z. By Lemma 3.11, and since

$$\widetilde{H}_\infty\left(Z \mid \mathcal{H}_1\right) \geq \widetilde{H}_\infty\left(Z \mid \mathcal{H}\right) - ((t+1)rs + ts) \geq 2s + \log(1/\varepsilon),$$

we have that

$$\left(\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_g)), \mathsf{a}(M_g, W), \mathcal{H}_1\right) \approx_{\delta+4\varepsilon} \left(U, \mathsf{a}(M_g, W), \mathcal{H}_1\right).$$

Conditioned on $\mathsf{a}(M_g, W), \mathcal{H}_1$, the random variable $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_g))$ is independent of the joint distribution of $\mathcal{A}, \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t$. Thus, by Lemma 3.2,

$$\left(\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_g)), \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{H}_2\right) \approx_{\delta+4\varepsilon} \left(U, \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{H}_2\right),$$

where $\mathcal{H}_2 = \left\{\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_g^i))\right\}_{i=1}^t, \mathcal{A}, \mathcal{M}, \mathcal{H}$. Now, by Lemma 3.11, and since

$$\begin{aligned}\widetilde{H}_\infty\left(W \mid \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{H}_2\right) &\geq \widetilde{H}_\infty\left(W \mid \mathcal{H}\right) - ((t+1)rb + tb) \\ &\geq 2b + \log(1/\varepsilon),\end{aligned}$$

we have that

$$\begin{aligned}&\left(\mathsf{b}(M_g, Z, W), \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_g)), \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{H}_2\right) \approx_{\delta+6\varepsilon} \\ &\left(U, \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_g)), \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{H}_2\right).\end{aligned}$$

Note that $\mathsf{b}(M_g, Z, W)$ is independent of $\mathcal{Z}$ conditioned on $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_g))$, $\left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t$, $\mathcal{H}_2$. Hence, by Lemma 3.2,

$$\left(\mathsf{b}(M_g, Z, W), \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}\right) \approx_{\delta+6\varepsilon} \left(U, \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}\right),$$

which concludes the proof of Item 2. □

Proof of Item 3. Let $j \in [r]$. As M is $(\delta, \mathcal{H})$-somewhere-independent of $M^1, \ldots, M^t$, we have that $(M_j, \mathcal{H}) \approx_\delta (U, \mathcal{H})$. By Lemma 3.11, and since $\widetilde{H}_\infty(W \mid \mathcal{H}) \geq 2b + \log(1/\varepsilon)$,

$$(\mathsf{a}(M_j, W), M_j, \mathcal{H}) \approx_{\delta+2\varepsilon} (U, M_j, \mathcal{H}).$$

Note that conditioned on the fixing of $M_j, \mathcal{H}$, the random variable $\mathsf{a}(M_j, W)$ is independent of $\mathcal{M}$. Thus, by Lemma 3.2,

$$(\mathsf{a}(M_j, W), \mathcal{M}, \mathcal{H}) \approx_{\delta+2\varepsilon} (U, \mathcal{M}, \mathcal{H}).$$

□

Proof of Item 4. Let $j \in [r]$. By Item 3,

$$(\mathsf{a}(M_j, W), \mathcal{M}, \mathcal{H}) \approx_{\delta+2\varepsilon} (U, \mathcal{M}, \mathcal{H}).$$

By Lemma 3.11 and since

$$\widetilde{H}_\infty(Z \mid \mathcal{M}, \mathcal{H}) \geq \widetilde{H}_\infty(Z \mid \mathcal{H}) - (t+1)rs \geq 2s + \log(1/\varepsilon),$$

we have that

$$(\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_j)), \mathsf{a}(M_j, W), \mathcal{M}, \mathcal{H}) \approx_{\delta+4\varepsilon} (U, \mathsf{a}(M_j, W), \mathcal{M}, \mathcal{H}).$$

Note that conditioned on $\mathsf{a}(M_j, W), \mathcal{M}, \mathcal{H}$, the random variable $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_j))$ is independent of $\mathcal{A}$. Hence, by Lemma 3.2,

$$(\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_j)), \mathcal{A}, \mathcal{M}, \mathcal{H}) \approx_{\delta+4\varepsilon} (U, \mathcal{A}, \mathcal{M}, \mathcal{H}) .$$

By Lemma 3.11, and since

$$\widetilde{H}_\infty (W \mid \mathcal{A}, \mathcal{M}, \mathcal{H}) \geq \widetilde{H}_\infty (W \mid \mathcal{H}) - (t+1)rb \geq 2b + \log(1/\varepsilon),$$

we have that

$$(\mathsf{b}(M_j, Z, W), \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_j)), \mathcal{A}, \mathcal{M}, \mathcal{H}) \approx_{\delta+6\varepsilon} (U, \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_j)), \mathcal{A}, \mathcal{M}, \mathcal{H}) .$$

Conditioned on $\mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_j)), \mathcal{A}, \mathcal{M}, \mathcal{H}$, the random variable $\mathsf{b}(M_j, Z, W)$ is independent of $\mathcal{Z}$. Thus, by Lemma 3.2,

$$(\mathsf{b}(M_j, Z, W), \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}) \approx_{\delta+6\varepsilon} (U, \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}) .$$

□

Proof of Item 5 and Item 6. The proofs readily follow by applying Lemma 3.3 and Lemma 3.5 in a straightforward manner, and so we omit the details. □

5.1.2 Proof of Lemma 5.2

We first describe the function $\mathsf{IPHalfCond}$ and then turn to the analysis. We start with some preparations. First, as in Section 5.1.1, we set $s = c \cdot \log(n/\varepsilon)$ to be a length that suffices for a seed of the strong seeded extractor from Theorem 3.9 set with error guarantee ε, when given a sample from an n-bit source. By Theorem 3.9, c is some universal constant, and so indeed $s = \Theta(\log(n/\varepsilon))$ as stated. Let c''' be some large enough constant (to be chosen as a function of c and of the constants c', c'' from Theorem 3.10), and set

$$s' = c'''ts,$$
$$b = 2c''s'.$$

We make use of the following building blocks:

- Let $\mathsf{Ext}_{\mathsf{in}} \colon \{0,1\}^n \times \{0,1\}^b \to \{0,1\}^{s'}$ be the strong seeded extractor from Theorem 3.9 for min-entropy $2s'$, set with error guarantee ε. Note that, indeed, a seed of length b suffices.
- Let $\mathsf{Ext}_{\mathsf{out}} \colon \{0,1\}^b \times \{0,1\}^{s'} \to \{0,1\}^s$ be the strong seeded extractor from Theorem 3.10 set with error guarantee ε. By Theorem 3.10, $\mathsf{Ext}_{\mathsf{out}}$ is an extractor for min-entropy $\max(2s, c''s') = c''s'$, where c'' is the universal constant from Theorem 3.10.
- Let $\mathsf{a} \colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^b$ and $\mathsf{b} \colon \{0,1\}^s \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^b$ be the pair of functions defined in Section 5.1.1. Note that, as required, $b \geq s$.

With these building blocks in hand we are ready to define $\mathsf{IPHalfCond}$. Given $m \in \{0,1\}^{r\times s}$, $z \in \{0,1\}^n$, and $w \in \{0,1\}^n$, we define the j'th row of $\mathsf{IPHalfCond}(m,z,w)$ by

$$\mathsf{IPHalfCond}(m,z,w)_j = \mathsf{Ext}_{\mathsf{out}}\left(\mathsf{b}\left(m_{2j},z,w\right), \mathsf{Ext}_{\mathsf{in}}\left(z, \mathsf{a}\left(m_{2j-1},w\right)\right)\right).$$

Proof of Lemma 5.2. First, we define the random variable $\mathcal{H}'$ guaranteed by the lemma as

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

where $\mathcal{Z},\mathcal{A},\mathcal{M}$ are as defined in Lemma 5.3, and $\mathsf{Ext}_{\mathsf{in}}(Z,\mathcal{A})$ is a shorthand notation for $\{\mathsf{Ext}_{\mathsf{in}}(Z,A)\}_{A\in\mathcal{A}}$. The first, and main, item of Lemma 5.2 follows by the following claim.

Claim 5.4. *Let $g \in [r]$ be such that M is $(\delta,\mathcal{H})$-independent of $M^1,\ldots,M^t$ at g. Then $\mathsf{IPHalfCond}(M,Z,W)$ is $(2\delta+12\varepsilon,\mathcal{H}')$-independent of $\{\mathsf{IPHalfCond}(M^i,Z,W)\}_{i=1}^t$ at $\lceil g/2\rceil$.*

Claim 5.4 readily follows by the following three claims.

Claim 5.5. *For any $j \in [r/2]$ it holds that*

$$(\mathsf{IPHalfCond}(M,Z,W)_j,\mathcal{H}') \approx_{2\delta+12\varepsilon} (U,\mathcal{H}').$$

Claim 5.6. *Let $g \in [r]$ be an odd integer such that*

$$\left(M_g,\left\{M_g^i\right\}_{i=1}^t,\mathcal{H}\right) \approx_\delta \left(U,\left\{M_g^i\right\}_{i=1}^t,\mathcal{H}\right).$$

Then,

$$\left(\mathsf{IPHalfCond}(M,Z,W)_{g'},\left\{\mathsf{IPHalfCond}(M^i,Z,W)_{g'}\right\}_{i=1}^t,\mathcal{H}'\right) \approx_{2\delta+12\varepsilon}$$
$$\left(U,\left\{\mathsf{IPHalfCond}(M^i,Z,W)_{g'}\right\}_{i=1}^t,\mathcal{H}'\right),$$

where $g' = (g+1)/2$.

Claim 5.7. *Let $g \in [r]$ be an even integer such that*

$$\left(M_g,\left\{M_g^i\right\}_{i=1}^t,\mathcal{H}\right) \approx_\delta \left(U,\left\{M_g^i\right\}_{i=1}^t,\mathcal{H}\right).$$

Then,

$$\left(\mathsf{IPHalfCond}(M,Z,W)_{g'},\left\{\mathsf{IPHalfCond}(M^i,Z,W)_{g'}\right\}_{i=1}^t,\mathcal{H}'\right) \approx_{2\delta+12\varepsilon}$$
$$\left(U,\left\{\mathsf{IPHalfCond}(M^i,Z,W)_{g'}\right\}_{i=1}^t,\mathcal{H}'\right),$$

where $g' = g/2$.

Proof of Claim 5.5. Let $j \in [r/2]$. Recall that

$$\mathsf{IPHalfCond}(M, Z, W)_j = \mathsf{Ext}_{\mathsf{out}}\left(\mathsf{b}(M_{2j}, Z, W), \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W))\right).$$

By Item 2 of Lemma 5.3,

$$(\mathsf{b}(M_{2j}, Z, W), \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}) \approx_{\delta+6\varepsilon} (U, \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}).$$

Conditioned on $\mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}$, the random variable $\mathsf{b}(M_{2j}, Z, W)$ and $\mathsf{Ext}_{\mathsf{in}}(Z, \mathcal{A})$ are independent. Lemma 3.2 then implies that

$$(\mathsf{b}(M_{2j}, Z, W), \mathcal{H}') \approx_{\delta+6\varepsilon} (U, \mathcal{H}').$$

As $\mathcal{H}'$ contains $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W))$, Lemma 3.1 implies that

$$(\mathsf{IPHalfCond}(M, Z, W)_j, \mathcal{H}') \approx_{\delta+6\varepsilon} (\mathsf{Ext}_{\mathsf{out}}(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W))), \mathcal{H}').$$

We now turn to show that the right hand side

$$(\mathsf{Ext}_{\mathsf{out}}(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W))), \mathcal{H}') \approx_{\delta+6\varepsilon} (U, \mathcal{H}'). \tag{5.1}$$

Equation (5.1), together with the triangle inequality, will conclude the proof. To prove Equation (5.1), it suffices to show that

$$(\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W)), \mathcal{A}, \mathcal{M}, \mathcal{H}) \approx_{\delta+4\varepsilon} (U, \mathcal{A}, \mathcal{M}, \mathcal{H}). \tag{5.2}$$

Indeed, Equation (5.2) together with Lemma 3.11 imply that

$$\begin{aligned}&(\mathsf{Ext}_{\mathsf{out}}(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W))), \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W)), \mathcal{A}, \mathcal{M}, \mathcal{H}) \approx_{\delta+6\varepsilon} \\ &(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W)), \mathcal{A}, \mathcal{M}, \mathcal{H}).\end{aligned}$$

Now, conditioned on $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W)), \mathcal{A}, \mathcal{M}, \mathcal{H}$, we have that the random variable $\mathsf{Ext}_{\mathsf{out}}(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W)))$ is independent of $(\mathcal{Z}, \mathsf{Ext}_{\mathsf{in}}(Z, \mathcal{A}))$ and so Equation (5.1) follows by Lemma 3.2.

We turn to prove Equation (5.2). By Item 1 of Lemma 5.3,

$$(\mathsf{a}(M_{2j-1}, W), \mathcal{M}, \mathcal{H}) \approx_{\delta+2\varepsilon} (U, \mathcal{M}, \mathcal{H}).$$

As

$$\widetilde{H}_\infty(Z \mid \mathcal{M}, \mathcal{H}) \geq \widetilde{H}_\infty(Z \mid \mathcal{H}) - (t+1)rs \geq 2s' + \log(1/\varepsilon),$$

and since $\mathsf{a}(M_{2j-1}, W)$ is independent of Z conditioned on $\mathcal{M}, \mathcal{H}$, it holds that

$$(\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W)), \mathsf{a}(M_{2j-1}, W), \mathcal{M}, \mathcal{H}) \approx_{\delta+4\varepsilon} (U, \mathsf{a}(M_{2j-1}, W), \mathcal{M}, \mathcal{H}).$$

As $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{2j-1}, W))$ is independent of $\mathcal{A}$ conditioned on $\mathsf{a}(M_{2j-1}, W), \mathcal{M}, \mathcal{H}$, Equation (5.2) follows by Lemma 3.2. This concludes the proof. □

Proof of Claim 5.6. Recall that for an odd g,

$$\mathsf{IPHalfCond}(M,Z,W)_{(g+1)/2} = \mathsf{Ext}_{\mathsf{out}}\left(\mathsf{b}(M_{g+1},Z,W), \mathsf{Ext}_{\mathsf{in}}(Z,\mathsf{a}(M_g,W))\right),$$

and that

$$\mathsf{b}(M_{g+1},Z,W) = \mathsf{Ext}(W,\mathsf{Ext}_s(Z,\mathsf{Ext}_s(W,M_{g+1}))).$$

By Item 1 of Lemma 5.3,

$$\left(\mathsf{a}(M_g,W), \left\{\mathsf{a}(M_g^i,W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}\right) \approx_{\delta+2\varepsilon} \left(U, \left\{\mathsf{a}(M_g^i,W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}\right).$$

Note that conditioned on $\left\{\mathsf{a}(M_g^i,W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}$, the random variable $\mathsf{a}(M_g,W)$ is independent of the joint distribution of $\left\{\mathsf{Ext}_{\mathsf{in}}(Z,\mathsf{a}(M_g^i,W))\right\}_{i=1}^t$. Thus, by Lemma 3.2,

$$(\mathsf{a}(M_g,W),\mathcal{H}_1) \approx_{\delta+2\varepsilon} (U,\mathcal{H}_1),$$

where

$$\mathcal{H}_1 = \left\{\mathsf{Ext}_{\mathsf{in}}(Z,\mathsf{a}(M_g^i,W))\right\}_{i=1}^t, \left\{\mathsf{a}(M_g^i,W)\right\}_{i=1}^t, \mathcal{M}, \mathcal{H}.$$

By Lemma 3.11, together with the fact that

$$\widetilde{H}_\infty(Z \mid \mathcal{H}_1) \geq \widetilde{H}_\infty(Z \mid \mathcal{H}) - ((t+1)rs + ts') \geq 2s' + \log(1/\varepsilon),$$

we have that

$$(\mathsf{Ext}_{\mathsf{in}}(Z,\mathsf{a}(M_g,W)), \mathsf{a}(M_g,W), \mathcal{H}_1) \approx_{\delta+4\varepsilon} (U, \mathsf{a}(M_g,W), \mathcal{H}_1).$$

As conditioned on $\mathsf{a}(M_g,W), \mathcal{H}_1$, the random variables $\mathsf{Ext}_{\mathsf{in}}(Z,\mathsf{a}(M_g,W))$ and $\mathcal{A}$ are independent, Lemma 3.2 implies that

$$(\mathsf{Ext}_{\mathsf{in}}(Z,\mathsf{a}(M_g,W)), \mathcal{A}, \mathcal{H}_1) \approx_{\delta+4\varepsilon} (U, \mathcal{A}, \mathcal{H}_1).$$

Set

$$\mathcal{H}_2 = \{\mathsf{Ext}_s(Z,\mathsf{Ext}_s(W,M_{g+1}^i))\}_{i=1}^t, \mathsf{Ext}_s(Z,\mathsf{Ext}_s(W,M_{g+1})), \mathcal{A}, \mathcal{H}_1.$$

Conditioned on $\mathcal{H}_2$, we have that $\mathsf{Ext}_{\mathsf{in}}(Z,\mathsf{a}(M_g,W))$ is $(\delta+4\varepsilon)$-close to having min-entropy $s' - (t+1)s \geq 0.9s'$. As $\mathsf{Ext}_{\mathsf{in}}(Z,\mathsf{a}(M_g,W))$ is independent of the joint distribution of $\{\mathsf{IPHalfCond}(M^i,Z,W)_{(g+1)/2}\}_{i=1}^t$ conditioned on $\mathcal{H}_2$, we have that conditioned on $\mathcal{H}_3 = \{\mathsf{IPHalfCond}(M^i,Z,W)_{(g+1)/2}\}_{i=1}^t, \mathcal{H}_2$, the random variable $\mathsf{Ext}_{\mathsf{in}}(Z,\mathsf{a}(M_g,W))$ is $(\delta+4\varepsilon)$-close to having min-entropy $0.9s'$.

We now turn to show that conditioned on $\mathcal{H}_3$, the random variable $\mathsf{b}(M_{g+1},Z,W)$ is $(\delta+6\varepsilon)$-close to having min-entropy $c''s' + \log(1/\varepsilon)$, as required by $\mathsf{Ext}_{\mathsf{out}}$. To this end, we apply Item 4 of Lemma 5.3 to get

$$(\mathsf{b}(M_{g+1},Z,W), \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}) \approx_{\delta+6\varepsilon} (U, \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}).$$

Conditioned on $\mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}$, the random variable $\mathsf{b}(M_{g+1}, Z, W)$ is a deterministic function of W whereas $\{\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_g^i, W))\}_{i=1}^t$, which are the only random variables in $\mathcal{H}_2$ that do not appear in $\mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}$, are all deterministic functions of the independent random variable Z. Hence, by Lemma 3.2,

$$(\mathsf{b}(M_{g+1}, Z, W), \mathcal{H}_2) \approx_{\delta+6\varepsilon} (U, \mathcal{H}_2)\,.$$

Further, the joint distribution of $\{\mathsf{IPHalfCond}(M^i, Z, W)_{(g+1)/2}\}$ conditioned on $\mathcal{H}_2$ is a deterministic function of W that consists of ts bits. Therefore, conditioned on $\mathcal{H}_3$, the random variable $\mathsf{b}(M_{g+1}, Z, W)$ is $(\delta + 6\varepsilon)$-close to having min-entropy $b - ts \geq c''s' + \log(1/\varepsilon)$.

So far we proved that conditioned on $\mathcal{H}_3$, the source $\mathsf{b}(M_{g+1}, Z, W)$ to the extractor $\mathsf{Ext}_{\mathsf{out}}$ in the definition of $\mathsf{IPHalfCond}(M, Z, W)_{(g+1)/2}$ is $(\delta + 6\varepsilon)$-close to having sufficient min-entropy, and that the seed $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_g, W))$ to that extractor is $(\delta + 4\varepsilon)$-close to a $(s', 0.9s')$-source. Further, note that conditioned on $\mathcal{H}_3$, the source and the seed above are independent. Thus, by Lemma 3.11 (more precisely, by a straightforward adjustment of Lemma 3.11 to the case of extractors with weak seeds), we have that

$$\begin{aligned}&\left(\mathsf{IPHalfCond}(M, Z, W)_{(g+1)/2}, \left\{\mathsf{IPHalfCond}(M^i, Z, W)_{(g+1)/2}\right\}_{i=1}^t, \mathcal{H}_2\right) \approx_{2\delta+12\varepsilon} \\ &\left(U, \left\{\mathsf{IPHalfCond}(M^i, Z, W)_{(g+1)/2}\right\}_{i=1}^t, \mathcal{H}_2\right).\end{aligned}$$

We conclude the proof by noting that conditioned on $\mathcal{H}_2$, $\mathsf{IPHalfCond}(M, Z, W)_{(g+1)/2}$ is independent of $\mathsf{Ext}_{\mathsf{in}}(Z, \mathcal{A}), \mathcal{Z}$, and so we can apply Lemma 3.2 so to "complete" the history to $\mathcal{H}'$, namely,

$$\begin{aligned}&\left(\mathsf{IPHalfCond}(M, Z, W)_{(g+1)/2}, \left\{\mathsf{IPHalfCond}(M^i, Z, W)_{(g+1)/2}\right\}_{i=1}^t, \mathcal{H}'\right) \approx_{2\delta+12\varepsilon} \\ &\left(U, \left\{\mathsf{IPHalfCond}(M^i, Z, W)_{(g+1)/2}\right\}_{i=1}^t, \mathcal{H}'\right).\end{aligned}$$

□

Proof of Claim 5.7. Note that for an even integer g,

$$\mathsf{IPHalfCond}(M, Z, W)_{g/2} = \mathsf{Ext}_{\mathsf{out}}\left(\mathsf{b}(M_g, Z, W), \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{g-1}, W))\right),$$

where

$$\mathsf{b}(M_g, Z, W) = \mathsf{Ext}(W, \mathsf{Ext}_s(Z, \mathsf{Ext}_s(W, M_g))).$$

By Item 2 of Lemma 5.3,

$$\left(\mathsf{b}(M_g, Z, W), \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}\right) \approx_{\delta+6\varepsilon} \left(U, \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}\right).$$

Conditioned on $\mathcal{Z}, \mathcal{A}, \mathcal{M}, \mathcal{H}$, the random variable $\mathsf{b}(M_g, Z, W)$ is independent of $\mathsf{Ext}_{\mathsf{in}}(Z, \mathcal{A})$. Lemma 3.2 then implies that

$$\left(\mathsf{b}(M_g, Z, W), \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{H}'\right) \approx_{\delta+6\varepsilon} \left(U, \left\{\mathsf{b}(M_g^i, Z, W)\right\}_{i=1}^t, \mathcal{H}'\right).$$

As $\mathcal{H}'$ contains $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{g-1}, W))$, $\left\{\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M^i_{g-1}, W))\right\}_{i=1}^t$, Lemma 3.1 implies that

$$\begin{aligned}&\left(\mathsf{IPHalfCond}(M, Z, W)_{g/2}, \left\{\mathsf{IPHalfCond}(M^i, Z, W)_{g/2}\right\}_{i=1}^t, \mathcal{H}'\right) \approx_{\delta+6\varepsilon} \\ &\left(\mathsf{Ext}_{\mathsf{out}}(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{g-1}, W))), \left\{\mathsf{IPHalfCond}(M^i, Z, W)_{g/2}\right\}_{i=1}^t, \mathcal{H}'\right). \end{aligned} \tag{5.3}$$

We now turn to show that the right hand side

$$\begin{aligned}&\left(\mathsf{Ext}_{\mathsf{out}}(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{g-1}, W))), \left\{\mathsf{IPHalfCond}(M^i, Z, W)_{g/2}\right\}_{i=1}^t, \mathcal{H}'\right) \approx_{\delta+6\varepsilon} \\ &\left(U, \left\{\mathsf{IPHalfCond}(M^i, Z, W)_{g/2}\right\}_{i=1}^t, \mathcal{H}'\right). \end{aligned} \tag{5.4}$$

Equation (5.3) together with Equation (5.4) and the triangle inequality will then conclude the proof. In order to prove Equation (5.4), we deduce from with Equation 5.2 that

$$(\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{g-1}, W)), \mathcal{A}, \mathcal{M}, \mathcal{H}) \approx_{\delta+4\varepsilon} (U, \mathcal{A}, \mathcal{M}, \mathcal{H}).$$

Thus, by Lemma 3.11,

$$\begin{aligned}&(\mathsf{Ext}_{\mathsf{out}}(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{g-1}, W))), \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{g-1}, W)), \mathcal{A}, \mathcal{M}, \mathcal{H}) \approx_{\delta+6\varepsilon} \\ &(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{g-1}, W)), \mathcal{A}, \mathcal{M}, \mathcal{H}).\end{aligned}$$

Equation (5.4) then follows as conditioned on $\mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{g-1}, W)), \mathcal{A}, \mathcal{M}, \mathcal{H}$, the random variable $\mathsf{Ext}_{\mathsf{out}}(U, \mathsf{Ext}_{\mathsf{in}}(Z, \mathsf{a}(M_{g-1}, W)))$ is independent of the joint distribution of $\mathcal{Z}$, $\mathsf{Ext}_{\mathsf{in}}(Z, \mathcal{A})$, and $\left\{\mathsf{IPHalfCond}(M^i, Z, W)_{g/2}\right\}_{i=1}^t$. □

The second item of Lemma 5.2 follows by definition, and the third and fourth items follow by Lemma 5.3 together with a straightforward application of Lemma 3.3 and Lemma 3.5. □

5.2 Proof of Proposition 5.1

Proof of Proposition 5.1. First, we assume that r is a power of 2. If this is not the case, one can complement the number of rows of M to $2^{\lceil \log_2 r \rceil}$ by duplicating the last row. Clearly, M is still $(\delta, \mathcal{H})$-somewhere-independent of the matrices obtained by applying the same process to $M^1, \ldots, M^t$. Furthermore, the asymptotic statement of the lemma remains unchanged. Therefore, we may assume that $q = \log_2 r$ is an integer.

Let $s = \Theta(\log(n/\varepsilon))$ be the parameter that appears in the statement of Lemma 5.2. For $j = 0, 1, \ldots, q$ set $r_j = r \cdot 2^{-j}$, and note that $r_0 = r$ and $r_q = 1$. For $j = 0, 1, \ldots, q-1$, let

$$\mathsf{IPHalfCond}_j \colon \{0,1\}^{r_j \times s} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{r_{j+1} \times s}$$

be the function from Lemma 5.2 set with error guarantee ε.

Given inputs $m \in \{0,1\}^{r \times s}$, $w \in \{0,1\}^n$, and $z \in \{0,1\}^n$, we define a sequence of matrices $m_{(0)}, m_{(1)}, \ldots, m_{(q)}$, where $m_{(j)}$ is an $r_j \times s$ matrix. First, we set $m_{(0)} = m$. For $j = 0, \ldots, q-1$, we define

$$m_{(j+1)} = \begin{cases} \mathsf{IPHalfCond}_j\left(m_{(j)}, z, w\right), & j \text{ is even;} \\ \mathsf{IPHalfCond}_j\left(m_{(j)}, w, z\right), & j \text{ is odd.} \end{cases}$$

Finally, we define $\mathsf{IPMerg}(m, z, w) = m_{(q)}$.

Clearly, the fact that $\mathsf{IPHalfCond}$ is efficiently-computable, as guaranteed by Lemma 5.2, implies that IPMerg is also efficiently-computable. We now prove by induction that there is a sequence of random variables $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_{q-1}$ such that for any $j = 0, 1, \ldots, q-1$,

- $M_{(j)}$ is $(\delta_j, \mathcal{H}_j)$-somewhere independent of $\{M^i_{(j)}\}_{i=1}^t$, where $\delta_j = 2^j\delta + 12(2^j - 1)\varepsilon$.
- For an even integer j, conditioned on $\mathcal{H}_j$, the random variables $Z, M_{(j)}, \{M^i_{(j)}\}_{i=1}^t$ are jointly independent of W.
- For an odd integer j, conditioned on $\mathcal{H}_j$, the random variables $W, M_{(j)}, \{M^i_{(j)}\}_{i=1}^t$ are jointly independent of Z.

Further, there exists a universal constant c such that for any $j \geq 1$,

- $\widetilde{H}_\infty(W \mid \mathcal{H}_j) \geq \widetilde{H}_\infty(W \mid \mathcal{H}) - ct^2 s \cdot \sum_{i=0}^{j-1} r_i$.
- $\widetilde{H}_\infty(Z \mid \mathcal{H}_j) \geq \widetilde{H}_\infty(Z \mid \mathcal{H}) - ct^2 s \cdot \sum_{i=0}^{j-1} r_i$.

The basis of the induction follows by the hypothesis of the lemma with $\mathcal{H}_0 = \mathcal{H}$. Consider $j > 0$. By the induction hypothesis, there exists a random variable $\mathcal{H}_{j-1}$ such that:

- $M_{(j-1)}$ is $(\delta_{j-1}, \mathcal{H}_{j-1})$-somewhere independent of $\{M^i_{(j-1)}\}_{i=1}^t$.
- If j is even then conditioned on $\mathcal{H}_{j-1}$, the random variables $W, M_{(j-1)}, \{M^i_{(j-1)}\}_{i=1}^t$ are jointly independent of Z.
- If j is odd then conditioned on $\mathcal{H}_{j-1}$, the random variables $Z, M_{(j-1)}, \{M^i_{(j-1)}\}_{i=1}^t$ are jointly independent of W.
- $\widetilde{H}_\infty(W \mid \mathcal{H}_{j-2}) \geq \widetilde{H}_\infty(W \mid \mathcal{H}) - ct^2 s \cdot \sum_{i=0}^{j-1} r_i$.
- $\widetilde{H}_\infty(Z \mid \mathcal{H}_{j-2}) \geq \widetilde{H}_\infty(Z \mid \mathcal{H}) - ct^2 s \cdot \sum_{i=0}^{j-1} r_i$.

Therefore, by Lemma 5.2, there exists a random variable $\mathcal{H}_j$ such that $M_{(j)}$ is $(\delta_j, \mathcal{H}_j)$-somewhere-independent of $\{M^i_{(j)}\}_{i=1}^t$. It is easy to verify that the remaining items holds.

By the above, we have that $M_{(q)}$ is $(\delta_q, \mathcal{H}_q)$-somewhere independent of $\{M^i_{(q)}\}_{i=1}^t$, where $\delta_q = O(r(\delta + \varepsilon))$. This concludes the proof. □

6 Multi-Source Independence-Preserving Mergers for Low Min-Entropy

The main result of this section is the following proposition which, based on results from the previous section, gives an independence-preserving merger that consumes several low min-entropy sources for the merging process.

Proposition 6.1. *For all integers n, r, t, for any $\sigma > 0$ such that $r = \sigma^{-b}$ for some integer b, and for any $\varepsilon > 0$, there exists an efficiently-computable function*

$$\mathsf{IPMultiMerg}\colon \{0,1\}^{r\times s} \times (\{0,1\}^n)^{2b+1} \to \{0,1\}^s,$$

where $s = \Theta(\log(n/\varepsilon))$ such that the following holds. Let $\mathcal{M} = (M, M^1, \ldots, M^t)$ be a sequence of random variables in the form of $r \times s$ matrices. Assume that M is $(0.9, \delta, \mathcal{H})$-somewhere-independent of $M^1, \ldots, M^t$. Let $V_0, \ldots, V_b$, $W_0, \ldots, W_{b-1}$ be independent n-bit random variables, all of which but for V_0 are independent of $\mathcal{M}$ conditioned on $\mathcal{H}$. Assume that $\widetilde{H}_\infty(V_i \mid \mathcal{H}) \geq k$ for $i = 0, \ldots, b$ and that $\widetilde{H}_\infty(W_i \mid \mathcal{H}) \geq k$ for $i = 0, \ldots, b-1$, where $k = \Omega\left(t^2\sigma^{-1}s\right)$. Set

$$Z = \mathsf{IPMultiMerg}(M, W_0, \ldots, W_{b-1}, V_0, \ldots, V_b),$$

and for $i \in [t]$ set

$$Z^i = \mathsf{IPMultiMerg}(M^i, W_0, \ldots, W_{b-1}, V_0, \ldots, V_b).$$

Then,

$$\left(Z, \left\{Z^i\right\}_{i=1}^t\right) \approx_{O(r(\delta+\varepsilon))} \left(U, \left\{Z^i\right\}_{i=1}^t\right).$$

The construction of $\mathsf{IPMultiMerg}$ is based on a two-source independence-preserving condenser IPCond which we construct in the following section. Then, in Section 6.2, we prove Proposition 6.1.

6.1 Two-source independence-preserving condensers

In this section we construct a two-source independence-preserving condenser, which is denoted by IPCond. The construction of IPCond is based on the merger IPMerg (which in turn is based on the condenser $\mathsf{IPHalfCond}$) from Lemma 5.2.

Lemma 6.2. *For all integers n, m, r, for any $\varepsilon > 0$, and for any $\sigma > 0$, there exists an efficiently-computable function*

$$\mathsf{IPCond}\colon \{0,1\}^{r\times s} \times (\{0,1\}^n)^3 \to \{0,1\}^{(\sigma r)\times s},$$

where $s = \Theta(\log(n/\varepsilon))$ such that the following holds. Let $\mathcal{M} = (M, M^1, \ldots, M^t)$ be a sequence of random variables in the form of $r \times s$ matrices. Assume that M is $(0.9, \delta, \mathcal{H})$-independent of $M^1, \ldots, M^t$. Let Z, W be n-bit random variables such that conditioned on $\mathcal{H}$, the joint distribution of $\mathcal{M}, Z$ is independent of W. Let V be an n-bit random variable that, conditioned on $\mathcal{H}$, is independent of the joint distribution of $Z, W, \mathcal{M}$. Assume further that

$$\begin{aligned}
\widetilde{H}_\infty(W \mid \mathcal{H}) &= \Omega\left(t^2\sigma^{-1}s\right), \\
\widetilde{H}_\infty(Z \mid \mathcal{H}) &= \Omega\left(t^2\sigma^{-1}s\right), \\
\widetilde{H}_\infty(V \mid \mathcal{H}) &= \Omega\left(ts\right).
\end{aligned}$$

Then, $\mathsf{IPCond}(M, Z, W, V)$ is $(0.9, \delta', \mathcal{H}')$-independent of $\{\mathsf{IPCond}(M^i, Z, W, V)\}_{i=1}^t$, where $\mathcal{H}' = W, Z, \mathcal{M}$ and $\delta' = O(\sigma^{-1}(\delta + \varepsilon))$.

For the proof of Lemma 6.2 we make use of the following theorem.

Theorem 6.3 ([KPS85, Gol11])**.** *For any integer r, and for all $\varepsilon > 0$, $\delta > 0$, there exists an explicit bipartite graph $G = (L, R, E)$ with $|L| = |R| = r$, and right degree $d = O(1/(\varepsilon^2\delta))$, such that the following holds. For any $G \subseteq L$, of size $|G| \geq \varepsilon r$, it holds that $1 - \delta$ fraction of the vertices in R have a neighbour in G.*

The following is an easy corollary of Theorem 6.3.

Corollary 6.4. *For any constant $\varepsilon > 0$, any integer r, and any $\sigma = \sigma(r) > 0$, there exists an explicit bipartite graph $G = (L, R, E)$ with $|L| = r$, $|R| = \sigma r$, and right degree $d = O(\sigma^{-1})$, such that the following holds. For any $B \subset L$ with size $|B| \leq \varepsilon r$, 0.9 fraction of the vertices in R have a neighbour outside of B.*

The corollary follows simply by applying Theorem 6.3 with $\delta = \sigma/10$, and disregard all but (arbitrarily chosen) σr vertices in R.

Proof of Lemma 6.2. We first describe the construction of IPCond and then turn to the analysis. To this end, we make use of the following building blocks:

- Let $G = (L, R, E)$ be the explicit bipartite graph from Corollary 6.4 with $|L| = r$, $|R| = \sigma r$, and $\varepsilon = 0.1$. By Corollary 6.4, G has right degree $d = O(\sigma^{-1})$.
- Let $\mathsf{IPMerg}\colon \{0,1\}^{d\times s} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^s$ be the function given by Proposition 5.1 set with error guarantee ε.
- Let $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^s$ be the strong seeded extractor from Theorem 3.9 for min-entropy $2s$, set with error guarantee ε.

Given a matrix $m \in \{0,1\}^{r\times s}$ and n-bit strings z, w, v, we define $\mathsf{IPCond}(m, z, w, v)$ as follows. Using the bipartite graph G, we associate with the $r \times s$ binary matrix m a sequence of σr binary matrices $m_{(1)}, \ldots, m_{(\sigma r)}$, each of order $d \times s$ as follows. We identify L with $[r]$ and R with $[\sigma r]$. For $j = 1, \ldots, \sigma r$, let $m_{(j)}$ denote the matrix obtained by taking the rows of m that correspond to the d neighbors of j in G. We define

$$\mathsf{IPCond}(m, z, w, v)_j = \mathsf{Ext}(v, \mathsf{IPMerg}(m_{(j)}, z, w)).$$

We now turn to the analysis. Let $B \subseteq [r]$ be the set of indices such that for any $g \notin B$, M is $(\delta, \mathcal{H})$-independent of $M^1, \ldots, M^t$ at g. By the hypothesis of the lemma, $|B| \leq 0.1r$. By the property of G, and by construction, there exists a subset $B' \subseteq [\sigma r]$ with $|B'| \leq 0.1\sigma r$ such that for any $g' \notin B'$ it holds that the matrix $M_{(g')}$ is $(\delta, \mathcal{H})$-somewhere-independent of $\{M^i_{(g')}\}_{i=1}^t$. Therefore, by Proposition 5.1, for any such g' there exists a random variable $\mathcal{H}_{g'}$ such that

$$\left(\mathsf{IPMerg}(M_{(g')}, Z, W), \left\{\mathsf{IPMerg}(M^i_{(g')}, Z, W)\right\}_{i=1}^t, \mathcal{H}_{g'}\right) \approx_{O(\sigma^{-1}(\delta+\varepsilon))}$$
$$\left(U, \left\{\mathsf{IPMerg}(M^i_{(g')}, Z, W)\right\}_{i=1}^t, \mathcal{H}_{g'}\right).$$

As conditioned on $\left\{\mathsf{IPMerg}(M^i_{(g')}, Z, W)\right\}_{i=1}^t$, $\mathcal{H}_{g'}$, the random variable $\mathsf{IPMerg}(M_{(g')}, Z, W)$ is independent of V, Lemma 3.2 implies that

$$\left(\mathsf{IPMerg}(M_{(g')}, Z, W), V, \left\{\mathsf{IPMerg}(M^i_{(g')}, Z, W)\right\}_{i=1}^t, \mathcal{H}_{g'}\right) \approx_{O(\sigma^{-1}(\delta+\varepsilon))}$$
$$\left(U, V, \left\{\mathsf{IPMerg}(M^i_{(g')}, Z, W)\right\}_{i=1}^t, \mathcal{H}_{g'}\right).$$

By Lemma 3.1,

$$\left(\mathsf{IPMerg}(M_{(g')}, Z, W), \left\{\mathsf{IPCond}(M^i, Z, W, V)_{g'}\right\}_{i=1}^t, \mathcal{H}_{g'}\right) \approx_{O(\sigma^{-1}(\delta+\varepsilon))}$$
$$\left(U, \left\{\mathsf{IPCond}(M^i, Z, W, V)_{g'}\right\}_{i=1}^t, \mathcal{H}_{g'}\right).$$

As $\widetilde{H}_\infty\left(V \mid \left\{\mathsf{IPCond}(M^i, Z, W, V)_{g'}\right\}_{i=1}^t, \mathcal{H}_{g'}\right) \geq \widetilde{H}_\infty(V \mid \mathcal{H}) - ts \geq 2s + \log(1/\varepsilon)$, Lemma 3.11 implies that

$$\left(\mathsf{IPCond}(M, Z, W, V)_{g'}, \mathsf{IPMerg}(M_{(g')}, Z, W), \left\{\mathsf{IPCond}(M^i, Z, W, V)_{g'}\right\}_{i=1}^t, \mathcal{H}_{g'}\right) \approx_{O(\sigma^{-1}(\delta+\varepsilon))}$$
$$\left(U, \mathsf{IPMerg}(M_{(g')}, Z, W), \left\{\mathsf{IPCond}(M^i, Z, W, V)_{g'}\right\}_{i=1}^t, \mathcal{H}_{g'}\right).$$

Now, conditioned on $\mathsf{IPMerg}(M_{(g')}, Z, W)$, $\{\mathsf{IPCond}(M^i, Z, W, V)_{g'}\}_{i=1}^t$, $\mathcal{H}_{g'}$, the random variable $\mathsf{IPCond}(M, Z, W, V)_{g'}$ is a deterministic function of V, and so it is independent of the joint distribution of $W, Z, \mathcal{M}$. Thus, by Lemma 3.2,

$$\left(\mathsf{IPCond}(M, Z, W, V)_{g'}, \left\{\mathsf{IPCond}(M^i, Z, W, V)_{g'}\right\}_{i=1}^t, W, Z, \mathcal{M}, \mathcal{H}\right) \approx_{O(\sigma^{-1}(\delta+\varepsilon))}$$
$$\left(U, \left\{\mathsf{IPCond}(M^i, Z, W, V)_{g'}\right\}_{i=1}^t, W, Z, \mathcal{M}, \mathcal{H}\right).$$

This concludes the proof. □

6.2 Proof of Proposition 6.1

Proof of Proposition 6.1. We start by describing the construction of $\mathsf{IPMultiMerg}$ and then turn to the analysis. To this end, we make use of the two-source independence-preserving condenser IPCond given by Lemma 6.2. For $j = 0, \ldots, b$, let $r_j = r\sigma^j$. Note that $r_0 = r$ and $r_b = 1$. For $j = 0, 1, \ldots, b-1$, let

$$\mathsf{IPCond}_j \colon \{0,1\}^{r_j \times s} \times (\{0,1\}^n)^3 \to \{0,1\}^{r_{j+1} \times s}$$

be the independence-preserving condenser from Lemma 6.2 set with error guarantee ε. Given a matrix $m \in \{0,1\}^{r \times s}$ and n-bit strings $v_0, \ldots, v_b$, $w_0, \ldots, w_{b-1}$, we define a sequence of matrices $m_{(0)}, \ldots, m_{(b)}$ where, for $j = 0, \ldots, b$, the matrix $m_{(j)}$ is of order $r_j \times s$. First, we let $m_{(0)} = m$. For $j = 0, \ldots, b-1$, we define

$$m_{(j+1)} = \mathsf{IPCond}_j\left(m_{(j)}, v_j, w_j, v_{j+1}\right).$$

Finally, we define $\mathsf{IPMultiMerg}(m, w_0, \ldots, w_{b-1}, v_0, \ldots, v_b) = m_{(b)}$.

With the construction in hand, we turn to the analysis. For $j = 0, \ldots, b$, let $\mathcal{M}_{(j)} = \{M^i_{(j)}\}_{i=1}^t \cup \{M_{(j)}\}$. Set $\mathcal{H}_0 = \mathcal{H}$, and for $j > 0$ define $\mathcal{H}_j = V_{j-1}, W_{j-1}, \mathcal{M}_{(j-1)}, \mathcal{H}_{j-1}$. Further, set $\delta_0 = \delta$ and for $j = 1, \ldots, b$ define $\delta_j = \sigma^{-1}(\delta_{j-1} + \varepsilon)$.

We prove by induction on $j = 0, \ldots, b$ that $M_{(j)}$ is $(0.9, \delta_j, \mathcal{H}_j)$-independent of $M^1_{(j)}, \ldots, M^t_{(j)}$. The base case $j = 0$ follows by the hypothesis of the proposition. Now, by the induction hypothesis, the joint distribution of $\mathcal{M}_{(j)}, V_j$ is independent of W_j conditioned on $\mathcal{H}_j$, and V_{j+1} is independent of the joint distribution of $\mathcal{M}_{(j)}, V_j, W_j$ conditioned on $\mathcal{H}_j$. Further, the average conditional min-entropy of V_j, W_j, and V_{j+1} with respect to $\mathcal{H}_j$ satisfy the hypothesis of Lemma 6.2. Thus, by Lemma 6.2, $M_{(j+1)}$ is $(0.9, \delta_{j+1}, \mathcal{H}_{j+1})$-independent of $M^1_{(j+1)}, \ldots, M^t_{(j+1)}$. The proof then follows as $M_{(b)}$ is $(0.9, \delta_b, \mathcal{H}_b)$-independent of $M^1_{(b)}, \ldots, M^t_{(b)}$, and since $\delta_b = O(r(\varepsilon + \delta))$. □

7 Proof of Theorem 1.1

In this section we prove Theorem 1.1. We start by giving a more formal restatement that also refers to the error guarantee of the extractor.

Theorem 7.1. *There exists a universal constant c such that the following holds. For any integer n, any $\delta > 0$, and any $\varepsilon > 0$, there exists an efficiently-computable extractor $\mathsf{Ext}\colon (\{0,1\}^n)^b \to \{0,1\}$ for $b = 2/\delta + c$ sources with min-entropy $(\log n)^{1+\delta} \cdot \varepsilon^{-7}$, having error guarantee ε.*

For the proof of Theorem 7.1 we make use of the following theorem that, roughly speaking, states that bounded independence fools threshold functions.

Theorem 7.2 ([DGJ+10])**.** *Let D be a t-wise independent distribution on $\{\pm 1\}^n$, and let $h\colon \{\pm 1\}^n \to \{\pm 1\}$ be a halfspace. Then,*

$$\left| \mathop{\mathbf{E}}_{x \sim D}[h(x)] - \mathop{\mathbf{E}}_{x \sim U}[h(x)] \right| \leq \frac{c \cdot \log t}{\sqrt{t}},$$

for some universal constant c.

We also need the following standard fact.

Fact 7.3. *For any odd integer n and for any $t < n/2$,*

$$\sum_{k=0}^{\lfloor n/2 \rfloor - t} \binom{n}{k} \geq 2^{n-1} - O\left(\frac{t \cdot 2^n}{\sqrt{n}}\right).$$

Proof. The proof follows as $\sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{k} = 2^{n-1}$ and since

$$\sum_{k=\lfloor n/2 \rfloor - t + 1}^{\lfloor n/2 \rfloor} \binom{n}{k} \leq t \cdot \binom{n}{\lfloor n/2 \rfloor} = O\left(\frac{t \cdot 2^n}{\sqrt{n}}\right).$$

□

The following is an easy corollary of Theorem 7.2.

Corollary 7.4. *Let $X_1, \ldots, X_r$ be a sequence of $\{0,1\}$ random variables. Let $\alpha > 0$ be some constant. Assume that there exists $B \subseteq [r]$, with $|B| \leq r^{1/2-\alpha}$, such that the random variables $\{X_i \mid i \in [r] \setminus B\}$ are t-wise independent and uniform. Then,*

$$\operatorname{bias}\left(\mathsf{Maj}(X_1, \ldots, X_r)\right) = O\left(\frac{\log t}{\sqrt{t}} + r^{-\alpha}\right).$$

Proof. As the majority function is symmetric, we may assume for the analysis that B consists of the largest $r^{1/2-\alpha}$ integers in $[r]$. That is, the distribution $X_1, \ldots, X_g$ is t-wise independent and each X_i is uniform, where $g = r - r^{1/2-\alpha}$. Let $U_1, \ldots, U_g$ be independent and uniform $\{0,1\}$ random variables. Note that if $\mathsf{Maj}(X_1, \ldots, X_r) = 1$ then

$$\sum_{i=1}^{g} X_i \geq \frac{r}{2} - r^{1/2-\alpha} = \frac{g}{2} - \frac{r^{1/2-\alpha}}{2}.$$

By Theorem 7.2, the probability that the latter event holds is bounded above by

$$\mathbf{Pr}\left[\sum_{i=1}^{g} U_i \geq \frac{g}{2} - \frac{r^{1/2-\alpha}}{2}\right] + O\left(\frac{\log t}{\sqrt{t}}\right) \leq \frac{1}{2} + O\left(r^{-\alpha} + \frac{\log t}{\sqrt{t}}\right),$$

where the inequality follows by Fact 7.3. Thus,

$$\mathbf{Pr}\left[\mathsf{Maj}(X_1, \ldots, X_r) = 1\right] \leq \frac{1}{2} + O\left(r^{-\alpha} + \frac{\log t}{\sqrt{t}}\right).$$

By a symmetric argument to the one used above, it holds that

$$\mathbf{Pr}\left[\mathsf{Maj}(X_1, \ldots, X_r) = 0\right] \leq \frac{1}{2} + O\left(r^{-\alpha} + \frac{\log t}{\sqrt{t}}\right).$$

This concludes the proof. □

For the proof of Theorem 7.1, we also require the following lemma.

Lemma 7.5 ([AGM03])**.** *Let $X_1, \ldots, X_n$ be $\{0,1\}$ random variables. Assume that for any $\emptyset \neq I \subseteq [n]$, with size $|I| \leq t$, the joint distribution of $\{X_i\}_{i \in I}$ is ε-close to uniform. Then, $X_1, \ldots, X_n$ is $(n^t \cdot \varepsilon)$-close to a t-wise independent distribution.*

We are now ready to prove Theorem 7.1.

Proof of Theorem 7.1. Let α be the universal constant from Proposition 4.2. By Proposition 4.2, applied with error guarantee $\varepsilon = n^{-6t/\alpha}$, given $7/\alpha$ sources with min-entropy $\widetilde{O}(t^2) \cdot \log n$, one can invoke the function $\mathsf{GenSISeq}$ so to efficiently generate a sequence of $r = n^{3/\alpha}$ random variables in the form of $r' \times s$ matrices $M^1, \ldots, M^r$, with $r' = O(t \log n)$, and $s = \Omega(\log(n/\varepsilon))$, having the following guarantee. There exists $B \subset [r]$, of size $|B| \leq r^{1/2-\alpha}$,

such that the sequence $M^1, \ldots, M^r$ is r^{-1}-close to a sequence $(M')^1, \ldots, (M')^r$ for which the following holds. For any $g \in [r] \setminus B$, and for any $i_1, \ldots, i_t \in [r] \setminus \{g\}$, it holds that $(M')^g$ is $(0.9, r^{-2t})$-independent of $\{(M')^{i_j}\}_{j=1}^t$.

We can now apply the independence-preserving merger IPMultiMerg given by Proposition 6.1 with $\sigma = (r')^{-\delta}$ and with error guarantee r^{-2t} to the sequence $\{M^i\}_{i=1}^r$. Note that the row of length of the matrices $\{M^i\}_i$ is sufficiently large. By Proposition 6.1, one can use $2/\delta + O(1)$ sources with min-entropy $\widetilde{O}(t^{3+\delta}) \cdot (\log n)^{1+\delta}$ so to obtain a sequence of $\{0,1\}$ random variables $Z_1, \ldots, Z_r$ with the following property. There exists a sequence of $\{0,1\}$ random variables $Z'_1, \ldots, Z'_r$ that is r^{-1}-close to the computed sequence $Z_1, \ldots, Z_r$, for which the following holds. For any $g \in [r] \setminus B$, and any $i_1, \ldots, i_t \in [r] \setminus \{g\}$,

$$\left((Z')_g, \{(Z')_{i_j}\}_{j=1}^t\right) \approx_\varepsilon \left(U, \{(Z')_{i_j}\}_{j=1}^t\right),$$

where $\varepsilon = O(r' \cdot r^{-2t}) \leq r^{-3t/2}$.

By Lemma 7.5, the joint distribution of the random variables $\{(Z')_i\}_{i=1}^r$ is $\varepsilon r^t \leq r^{-t/2}$-close to a sequence $\{Z''_i\}_{i=1}^r$ such that for any $I \subseteq [r] \setminus B$, of size $|I| \leq t$, the joint distribution of $\{Z''_i\}_{i \in I}$ is uniform. By Corollary 7.4, the random variable $\mathsf{Maj}(Z''_1, \ldots, Z''_r)$ has bias $O(r^{-\alpha} + \log(t)/\sqrt{t})$. As this bound asymptotically dominates the distance between the sequence $\{Z_i\}_i$ and the sequence $\{Z''_i\}_i$, the bias of the computed value $\mathsf{Maj}(Z_1, \ldots, Z_r)$ is too bounded above by $O(r^{-\alpha} + \log(t)/\sqrt{t})$. This concludes the proof. □

8 Proof of Theorem 1.2

In this section we prove Theorem 1.2. The two main building blocks we need from the literature are the condenser of Li [Li13a], that is based on the lightest bin protocol by Feige [Fei99], as well as mergers with weak-seeds [Coh15a]. We state these results and some definitions we use.

Definition 8.1. *Let M be a random variable in the form of an $r \times \ell$ matrix. We say that M is a* somewhere-random source *if there exists $g \in [r]$ such that M_g is uniformly distributed.*

Definition 8.2. *Let M be a random variable in the form of an $r \times \ell$ matrix. We say that M is* (t, ε, δ)-independent *if there exists $G \subseteq [r]$ of size $|G| \geq \delta r$ such that for any $I \subseteq G$, $|I| \leq t$, the joint distribution of $\{M_i\}_{i \in I}$ is ε-close to uniform.*

Theorem 8.3 ([Li13a])**.** *For any constant $0 < \gamma < 1$ there exists $c = c(\gamma)$ such that the following holds. Let r, ℓ, n, k, t be integers such that $t \geq c$ is even, and let $\varepsilon > 0$. Assume that*

$$\begin{aligned} t &\leq \sqrt{r}, \\ \varepsilon &\leq r^{-6t}, \\ k &= \Omega(t \cdot \log(n/\varepsilon)), \\ \ell &= \Omega(\log(n/\varepsilon)). \end{aligned}$$

Then, there exists an efficiently-computable function

$$\mathsf{LightestBin}\colon \{0,1\}^{r\times\ell} \times \{0,1\}^n \to \{0,1\}^{r'\times\ell'},$$

with $r' = O(t \cdot r^{2/\sqrt{t}})$ *and* $\ell' = k/(2t)$*, such that the following holds. Let* M *be a random variable in the form of an* $r \times \ell$ *matrix that is* (t,ε,δ)*-independent for some constant* $\delta > 1/2$*. Let* W *be an* (n,k)*-source that is independent of* M*. Then,* $\mathsf{LightestBin}(M,W)$ *is* $r^{-\sqrt{t}/2}$*-close to a random variable that is* $(t,\varepsilon',(1-\gamma)\delta)$*-independent, where* $\varepsilon' \leq (r')^{-6t}$*.*

Definition 8.4 (Mergers with weak-seeds)**.** *A function*

$$\mathsf{Merg}\colon \{0,1\}^{r\times\ell} \times \{0,1\}^n \to \{0,1\}^m$$

is called a merger with weak-seeds *for entropy* k*, with error guarantee* ε*, if the following holds. For any* $r \times \ell$ *somewhere-random source* X *and an independent* (n,k)*-weak-source* Y*, it holds that* $\mathsf{Merg}(X,Y) \approx_\varepsilon U$*.*

Theorem 8.5 ([Coh15a])**.** *For all integers* n, r *and for any* $\varepsilon > 0$*, there exists a* $\mathrm{poly}(n, r, \log(1/\varepsilon))$*-time computable merger with weak-seeds for entropy* k*, with error guarantee* ε*,*

$$\mathsf{Merg}\colon \{0,1\}^{r\times\ell} \times \{0,1\}^n \to \{0,1\}^m,$$

with

$$\begin{aligned}
\ell &= \Theta\left(r^2 \cdot \log(r) \cdot \log\left(\frac{nr}{\varepsilon}\right)\right),\\
k &= \Omega\left(r \cdot \log(r) \cdot \log\left(\frac{r \cdot \log n}{\varepsilon}\right)\right),\\
m &= \ell/(2r).
\end{aligned}$$

We also make use of the following lemma.

Lemma 8.6 ([Li15b])**.** *There exists a universal constant* c *such that the following holds. For all integers* n, ℓ*, there exists an efficiently-computable function*

$$f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{r\times\ell},$$

where $r = n^c$*, such that the following holds. Let* X, Y *be independent* (n,k)*-sources with* $k = \Omega(\ell + \log(n/\varepsilon))$*. Then,* $f(X,Y)$ *is* ε*-close to a random variable* M *in the form of an* $r \times \ell$ *matrix such that* 0.9 *fraction of the rows of* M *are uniform.*

With these building blocks, we are ready to prove Theorem 1.2.

Proof of Theorem 1.2. Unlike most proofs in this paper, we find it more convenient to describe the construction of the extractor along with its analysis.

Set $t = (\log n)^{\delta/4}$ and $\varepsilon = n^{-\Omega(t)}$ for some large enough hidden constant in the exponent. Lemma 8.6 implies that by consuming two sources with min-entropy $\Omega((\log n)^{1+\delta})$, one can

obtain a random variable that is ε-close to an $r \times \ell$ matrix M, with $r = n^c$ and $\ell = \Omega((\log n)^{1+\delta})$ such that the following holds. There exists $G \subseteq [r]$, $|G| \geq 0.9r$, such that for any $g \in G$, M_g is uniform. Here, c is the universal constant from Lemma 8.6.

By consuming one more source, Lemma 4.4 implies that we can efficiently transform the $r \times \ell$ matrix obtained above to a sequence of matrices $M_1, \ldots, M_r$, each of order $r' \times m$, where $r' = O(t \log n)$ and $m = \Omega(\log(n/\varepsilon))$, having the following property. For any $g \in G$ and for any $i_1, \ldots, i_t \in [r] \setminus \{g\}$, the random variable M_g is $(0.9, \varepsilon)$-independent of $M_{i_1}, \ldots, M_{i_t}$.

Next, we set $\sigma = (\log n)^{-\delta/4}$ and apply Proposition 6.1 to the sequence of matrices $M_1, \ldots, M_r$. By Proposition 6.1, using $8/\delta + O(1)$ fresh sources, each having min-entropy $\Omega((\log n)^{1+\delta})$, we obtain an $r \times \Omega(\log(n/\varepsilon))$ matrix that is $(t, \varepsilon', 0.9)$-independent, with $\varepsilon' = O(\varepsilon r')$. By consuming one more source, we can increase the row length from $\Omega(\log(n/\varepsilon))$ to $\Omega(t \cdot \log(n/\varepsilon))$, which is required for the next step.

We now apply Theorem 8.3 so to obtain an $r_1 \times \ell_1$ matrix that is $(t, \varepsilon_1, 0.9(1-\gamma))$-independent, where

$$\begin{aligned} r_1 &= O\left((\log n)^{\delta/4} \cdot \left(2^{2c}\right)^{(\log n)^{1-\delta/8}}\right), \\ \ell_1 &= \Omega\left((\log n)^{1+\delta}/t\right) = \Omega\left((\log n)^{1+3\delta/4}\right), \\ \varepsilon_1 &= r_1^{-6t}. \end{aligned}$$

Here, again, c is the universal constant from Lemma 8.6. We set the value of γ later on.

We can repeat this process, namely, apply Theorem 8.3 using a fresh source so to obtain an $r_2 \times \ell_2$ matrix that is $(t, \varepsilon_2, 0.9(1-\gamma)^2)$-independent, where

$$\begin{aligned} r_2 &= O\left((\log n)^{\delta/4} \cdot \left(2^{4c}\right)^{(\log n)^{1-\delta/4}}\right), \\ \ell_2 &= \ell_1, \\ \varepsilon_2 &= r_2^{-6t}, \end{aligned}$$

After $j = \lceil 8/\delta \rceil + 1$ iterations, we obtain an $r_j \times \ell_j$ matrix, where $r_j = O((\log n)^{\delta/4})$ and $\ell_j = \Omega((\log n)^{1+3\delta/4})$ that is ε_j-close to a somewhere-random source, with $\varepsilon_j = 2^{-\Omega((\log n)^{\delta/4})}$. Note that for this we need to choose γ such that $(1-\gamma)^{\lceil 8/\delta \rceil + 1} > 1/2$. As δ is constant, one can set the value of the constant γ accordingly.

At this point we can apply Theorem 8.5 so to obtain a string S, which is $O(\varepsilon_j)$-close to uniform on $(\log n)^{1+\delta/4}$-bit strings. By consuming one more source with min-entropy $(\log n)^{1+\delta}$, we can use S as a seed to the seeded extractor from Theorem 3.9 so to extract $\Omega((\log n)^{1+\delta})$ bits that are $O(\varepsilon_j)$-close to uniform. This concludes the proof. □

9 Proof of Theorem 1.3

In this section we prove Theorem 1.3. Our proof follows [CS15] with the sole difference of using the extractor from Theorem 1.2 rather than the extractor by [Li15b]. For completeness, we give the necessary details. We start by recalling the definition of the following class of sources from [CS15] and its relation to zero-fixing sources given in the subsequent claim.

Definition 9.1 (Fixed-weight sources)**.** *Let $n \geq k \geq w$ be integers. A random variable $X \subseteq \{0,1\}^n$ is called an* (n,k,w)-fixed-weight source *if there exists $S \subseteq [n]$, with size $|S| = k$, such that a sample from $x \sim X$ is obtained as follows. First, one samples a string $x' \in \{0,1\}^k$ of weight w, uniformly at random from all $\binom{k}{w}$ such strings. Then one sets $x|_S = x'$, and $x_i = 0$ for all $i \notin S$.*

Claim 9.2 ([CS15])**.** *Let X be an (n,k)-zero-fixing source. Then, X is $2^{-\Omega(k)}$-close to a convex combination of (n,k,w)-fixed-weight sources, with $k/3 \leq w \leq 2k/3$.*

For the proof we make use of the following result from [CS15] which, informally speaking, shows that is is possible to efficiently transform a single fixed-weight source to several independent weak-sources.

Theorem 9.3 ([CS15])**.** *For any integers n, c, where c is a power of 2, there exists an $O(cn)$-time computable function*

$$\mathsf{Splitter}\colon \{0,1\}^n \to (\{0,1\}^n)^c\,,$$

with the following property. Let X be an (n,k,w)-fixed-weight source, with $k/3 \leq w \leq 2k/3$. Let $(Y_1,\ldots,Y_c) = \mathsf{Splitter}(X)$, with $Y_i \in \{0,1\}^n$ for all $i \in [c]$. Then, there exist random variables $M_1,\ldots,M_{c-1}$, and deterministic functions $k_1,\ldots,k_{c-1}$ of them, such that conditioned on any fixing $(M_1,\ldots,M_{c-1}) = (m_1,\ldots,m_{c-1})$, the following holds:

- *The random variables $Y_1,\ldots,Y_c$ are independent.*
- *For every $i \in [c]$, the random variable Y_i is an (n,k_i,w_i)-fixed-weight source, with $w_i \in [w/c - 1, w/c]$, and $k_1 + \cdots + k_c = k - c + 1$.*

Furthermore, except with probability $c \cdot 2^{-\Omega(k/(c\cdot\log^2 c))}$ over the fixings of $(M_1,\ldots,M_{c-1})$, it holds that for all $i \in [c]$, $k_i \geq 0.9k/c$.

We also make use of the following (seedless) lossless condenser for bit-fixing sources due to Rao [Rao09].

Theorem 9.4 ([Rao09])**.** *For all integers n, k, there exists an efficiently-computable linear transformation $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^{k\log n}$, such that for any (n,k)-bit-fixing source X it holds that Cond restricted to X is one-to-one.*

With these results in hand, we are ready to prove Theorem 1.3.

Proof of Theorem 1.3. We first describe the construction of $\mathsf{ZeroBFExt}$ and then turn to the analysis. We make use of the following building blocks:

- Let $\mathsf{Ext}\colon \left(\{0,1\}^{k\log n}\right)^c \to \{0,1\}^\ell$ be the multi-source extractor from Theorem 1.2, set to extract $\ell = \Omega(k)$ bits from c independent $(k\log n, k)$-weak-sources, with $k \geq O(\log^{1+\delta}(k\log n))$. By Theorem 1.2, it suffices to take $c = O(1/\delta)$. Note further that $k = O((\log\log n)^{1+\delta})$.

- Let $\mathsf{Splitter}\colon \{0,1\}^n \to (\{0,1\}^n)^c$ be the function given by Theorem 9.3.
- Let $\mathsf{Cond}\colon \{0,1\}^n \to \{0,1\}^{k \log n}$ be the lossless-condenser given by Theorem 9.4.

Let $x \in \{0,1\}^n$. We define $\mathsf{ZeroBFExt}(x)$ as follows. First, we compute $(y_1, \ldots, y_c) = \mathsf{Splitter}(x)$. Secondly, for each $i \in [c]$ we compute $z_i = \mathsf{Cond}(y_i)$. Finally, we define $\mathsf{ZeroBFExt}(x) = \mathsf{Ext}(z_1, \ldots, z_c)$.

Turning to the analysis, by Claim 9.2, X is $2^{-\Omega(k)}$-close to a convex combination of (n,k,w)-weight-fixing sources $\{X_w\}_{w=k/3}^{2k/3}$. Thus, $\mathsf{Splitter}(X)$ is $2^{-\Omega(k)}$-close to a convex combination of the random variables $\{\mathsf{Splitter}(X_w)\}_{w=k/3}^{2k/3}$. We denote $((Y_w)_1, \ldots, (Y_w)_c) = \mathsf{Splitter}(X_w)$. Fix $w \in [k/3, 2k/3]$. By Theorem 9.3, conditioned on a suitable set of random variables, $(Y_w)_1, \ldots, (Y_w)_c$ are independent. Moreover, except with probability $2^{-\Omega(k)}$ with respect to the conditioning, it holds that for all $i \in [c]$, $(Y_w)_i$ is an $(n, k', w/c)$-weight-fixing source, with $k' \geq 0.9k/c$. Since

$$\binom{k'}{w/c} \geq \left(\frac{k'}{w/c}\right)^{w/c} \geq \left(\frac{0.9k}{w}\right)^{w/c} \geq \left(\frac{0.9}{2/3}\right)^{w/c} = 2^{\Omega(k)},$$

we have that $H_\infty((Y_w)_i) = \Omega(k)$ for all $i \in [c]$, except with probability $2^{-\Omega(k)}$.

Recall that $Z_i = \mathsf{Cond}(Y_i)$. With the notation above, we have that Z_i is $2^{-\Omega(k)}$-close to a convex combination of $(Z_w)_i = \mathsf{Cond}((Y_w)_i)$, where $k/3 \leq w \leq 2k/3$. Since $(Y_w)_i$ is contained in some (n,k)-bit-fixing source, Theorem 9.4 guarantees that Cond restricted to the support of $(Y_w)_i$ is one-to-one, and so $H_\infty((Z_w)_i) = H_\infty((Y_w)_i) = \Omega(k)$. Thus, except with probability $2^{-\Omega(k)}$, we have that for all $i \in [c]$, $(Z_w)_i$ is a $(k \log n, \Omega(k))$-weak source. This implies that $\mathsf{ZeroBFExt}(X_w) = \mathsf{Ext}((Z_w)_1, \ldots, (Z_w)_c)$ is $2^{-(\log\log n)^{\Omega(1)}}$-close to uniform, which completes the proof of the theorem as $\mathsf{ZeroBFExt}(X)$ is $2^{-\Omega(k)}$-close to a convex combination of $\{\mathsf{ZeroBFExt}(X_w)\}_{w=k/3}^{2k/3}$. □

10 Summary and Open Problems

In this paper we gave an explicit construction of an extractor for $2/\delta + O(1)$ sources, each with min-entropy $(\log n)^{1+\delta}$, for any $\delta > 0$. The end goal would be to construct a two-source extractor, or even a two-source disperser, for min-entropy $O(\log n)$. This would yield significantly improved constructions of Ramsey graphs. The next natural step towards this goal is to devise an extractor or a disperser for a constant number of sources with logarithmic or even quasi-logarithmic min-entropy.

Another interesting problem is to come up with a construction of multi-source extractors or dispersers that has a simple and succinct description, though possibly with an involved analysis. Bourgain's two-source extractor [Bou05], which has a clean and simple description, and whose analysis relies on results from point-line incidence bounds over finite fields, certainly falls under this category. Unfortunately, however, Bourgain's extractor only supports min-entropy rate slightly below 1/2. The multi-source extractor by Barak *et al.*

[BIW06] is another example of an extractor that is very simple to describe, as its construction only involves a simple sequence of additions and multiplications. The analysis of this extractor is based on a statistical analog of the deep sum-product theorem from additive combinatorics. However, for the latter extractor to support min-entropy rate δ, it requires poly$(1/\delta)$-sources. Constructing dispersers is based on the challenge-response mechanism [BKS+05, BRSW12, Coh15c] and, unfortunately, these are also highly involved constructs.

It is interesting to point out that a little over a decade ago, an analog situation occurred for seeded extractors – after a long line of research that was accumulated to the fairly involved seeded extractor by Lu *et al.* [LRVW03], a significantly simpler extractor (also with better parameters) was obtained by Guruswami *et al.* [GUV09], and in subsequent works [DW11, DKSS09, TSU12].

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.

[BBR85] C. H. Bennett, G. Brassard, and J. M. Robert. How to reduce your enemy's information. In *Advances in Cryptology (CRYPTO)*, volume 218, pages 468–476. Springer, 1985.

[BIW06] B. Barak, R. Impagliazzo, and A. Wigderson. Extracting randomness using few independent sources. *SIAM Journal on Computing*, 36(4):1095–1118, 2006.

[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.

[BSZ11] E. Ben-Sasson and N. Zewi. From affine to two-source extractors via approximate duality. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, pages 177–186. ACM, 2011.

[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.

[CGH+85] B. Chor, O. Goldreich, J. Håstad, J. Freidmann, S. Rudich, and R. Smolensky. The bit extraction problem or t-resilient functions. In *in Proceedings of the 26th Annual Symposium on Foundations of Computer Science, 1985.*, pages 396–407. IEEE, 1985.

[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.

[CS15] G. Cohen and I. Shinkar. Zero-fixing extractors for sub-logarithmic entropy. In *Automata, Languages, and Programming*, pages 343–354. Springer, 2015.

[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.

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

[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.

[DORS08] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[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.

[DW11] Z. Dvir and A. Wigderson. Kakeya sets, new mergers, and old extractors. *SIAM Journal on Computing*, 40(3):778–792, 2011.

[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.

[Gol11] Oded Goldreich. A sample of samplers: A computational perspective on sampling. In Oded Goldreich, editor, *Studies in Complexity and Cryptography*, volume 6650 of *Lecture Notes in Computer Science*, pages 302–332. Springer, 2011.

[GRS06] A. Gabizon, R. Raz, and R. Shaltiel. Deterministic extractors for bit-fixing sources by obtaining an independent seed. *SIAM Journal on Computing*, 36(4):1072–1094, 2006.

[GS95] A. Garcia and H. Stichtenoth. A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vladut bound. *Inventiones Mathematicae*, 121(1):211–222, 1995.

[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.

[GVWZ15] O. Goldreich, E. Viola, A. Wigderson, and D. Zuckerman. On randomness extraction in ac0. In *30th Conference on Computational Complexity (CCC 2015)*, volume 33, pages 601–668. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015.

[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.

[KPS85] R. M. Karp, N. Pippinger, and M. Sipser. A time-randomness tradeoff. In *AMS Conference on Probabilistic Computational Complexity*, 1985.

[KZ06] J. Kamp and D. Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2006.

[Li11] X. Li. Improved constructions of three source extractors. In *IEEE 26th Annual Conference on Computational Complexity*, pages 126–136, 2011.

[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 condensers for arbitrary min-entropy, and almost optimal protocols for privacy amplification. *arXiv preprint arXiv:1211.0651*, 2012.

[Li12c] 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.

[LRVW03] C.J. Lu, O. Reingold, S. Vadhan, and A. Wigderson. Extractors: Optimal up to constant factors. In *Proceedings of the thirty-fifth annual ACM Symposium on Theory of Computing*, pages 602–611. ACM, 2003.

[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.

[Rey11] L. Reyzin, 2011. http://www.cs.bu.edu/~reyzin/teaching/s11cs937/notes-leo-2.pdf.

[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.

[RV13] Y. Reshef and S. Vadhan. On extractors and exposure-resilient functions for sublogarithmic entropy. *Random Structures & Algorithms*, 42(3):386–401, 2013.

[Sti09] H. Stichtenoth. *Algebraic function fields and codes*, volume 254. Springer Science & Business Media, 2009.

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

[TSU12] A. Ta-Shma and C. Umans. Better condensers and new extractors from Parvaresh-Vardy codes. In *IEEE 27th Annual Conference on Computational Complexity (CCC)*, pages 309–315. IEEE, 2012.

[Vaz85] V. U. Vazirani. Towards a strong communication complexity theory or generating quasi-random sequences from two communicating slightly-random sources. In *Proceedings of the seventeenth annual ACM Symposium on Theory of Computing*, pages 366–378, 1985.

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

A Proof of Lemma 3.11

In this section we prove Lemma 3.11. We start by proving the following lemma by [Rey11] (see further references therein).

Lemma A.1 ([Rey11])**.** *Let A, B, C, D, and F be random variables such that C, D have the same domain. Suppose that for any $a \in \mathsf{supp}(A)$,*

- *$(C \mid A = a)$ and $(F \mid A = a)$ are independent*

- $(C \mid A = a)$ *and* $(B \mid A = a)$ *are independent*
- $(D \mid A = a)$ *and* $(F \mid A = a)$ *are independent*
- $(D \mid A = a)$ *and* $(B \mid A = a)$ *are independent*

Let f *be a function such that*

$$(f(B, C), C, A) \approx_{\varepsilon} (F, C, A)\,.$$

Assume further that $(D, A) \approx_{\delta} (C, A)$*. Then,*

$$(f(B, D), D, A) \approx_{\delta+\varepsilon} (F, D, A)\,.$$

For the proof of Lemma A.1 we make use of the following simple and well-known fact.

Fact A.2. *Let* X, Y *be two random variables with a common domain* D*. Let* $A = \{z \in D \mid \mathbf{Pr}[X = z] > \mathbf{Pr}[Y = z]\}$*. Then,*

$$\mathsf{SD}(X, Y) = \sum_{z \in A} \mathbf{Pr}[X = z] - \mathbf{Pr}[Y = z].$$

Proof of Lemma A.1. For ease of reading, for any $a \in \mathsf{supp}(A)$ and any random variable X, we denote $(X \mid A = a)$ by X_a. Now,

$$\mathsf{SD}\left((f(B, D), D, A), (F, D, A)\right) = \mathop{\mathbf{E}}_{a \sim A}\left[\mathsf{SD}\left((f(B_a, D_a), D_a), (F_a, D_a)\right)\right]$$

Fix $a \in \mathsf{supp}(A)$. By the hypothesis, B_a, D_a are independent and also F_a, D_a are independent. Hence,

$$\mathsf{SD}\left((f(B_a, D_a), D_a), (F_a, D_a)\right) = \sum_{x} \mathbf{Pr}[D_a = x] \cdot \mathsf{SD}(f(B_a, x), F_a).$$

The expression on the right hand side can be written as $\alpha(a) + \beta(a)$ where

$$\alpha(a) = \sum_{x} \left(\mathbf{Pr}[D_a = x] - \mathbf{Pr}[C_a = x]\right) \cdot \mathsf{SD}(f(B_a, x), F_a),$$

and

$$\beta(a) = \sum_{x} \mathbf{Pr}[C_a = x] \cdot \mathsf{SD}(f(B_a, x), F_a).$$

As C_a, B_a are independent and C_a, F_a are independent,

$$\beta(a) = \mathsf{SD}\left((f(B_a, C_a), C_a), (F_a, C_a)\right),$$

and so

$$\mathop{\mathbf{E}}_{a \sim A}\left[\beta(a)\right] = \mathsf{SD}\left((f(B, C), C, A), (F, C, A)\right) \leq \varepsilon.$$

We now turn to bound $\alpha(a)$. Let T be the set of all elements x such that $\mathbf{Pr}[D_a = x] > \mathbf{Pr}[C_a = x]$. By Fact A.2 and since $\mathsf{SD}(\cdot) \in [0,1]$,

$$\begin{aligned}\alpha(a) &= \sum_x (\mathbf{Pr}[D_a = x] - \mathbf{Pr}[C_a = x]) \cdot \mathsf{SD}(f(B_a, x), F_a) \\ &\leq \sum_{x \in T} (\mathbf{Pr}[D_a = x] - \mathbf{Pr}[C_a = x]) \cdot \mathsf{SD}(f(B_a, x), F_a) \\ &\leq \sum_{x \in T} \mathbf{Pr}[D_a = x] - \mathbf{Pr}[C_a = x] \\ &= \mathsf{SD}(D_a, C_a).\end{aligned}$$

Hence,

$$\mathop{\mathbf{E}}_{a \sim A}[\alpha(a)] = \mathsf{SD}\left((C, A), (D, A)\right) \leq \delta.$$

This concludes the proof. □

We are now ready to prove Lemma 3.11.

Proof of Lemma 3.11. By Lemma A.1 applied with $A = \mathcal{H}$, $B = W$, $C = U_1$, $D = S$, $F = U_2$, and $f = \mathsf{Ext}$, and since W is independent of S conditioned on $\mathcal{H}$, it suffices to show that

$$(\mathsf{Ext}(W, U_1), U_1, \mathcal{H}) \approx_{2\varepsilon} (U_2, U_1, \mathcal{H}).$$

By Lemma 3.4, except with probability ε over $h \sim \mathcal{H}$ it holds that $H_\infty(W \mid (\mathcal{H} = h)) \geq k$. Conditioned on this event, the source fed to Ext has min-entropy k which is sufficient for Ext, and so the output of the extractor is ε-close to uniform. By taking into account the fact that with probability at most ε, W has insufficient min-entropy as required by Ext, the above equation follows. □

B Proof Sketch for Claim 2.2

In this section we sketch the proof for Claim 2.2.

Proof sketch for Claim 2.2. The general proof strategy is to show that there exists an event which occurs with probability close to 1, conditioned on which, $\mathsf{IPMerg}_3(X', Y', W)$ is fixed to a constant, whereas $\mathsf{IPMerg}_3(X, Y, W)$ is close to uniform. Clearly, this suffices to conclude the proof. This event is found by performing a sequence of steps, where in each step we fix some carefully chosen random variable.

We start by conditioning on the fixing $Y'|_s = y'|_s$ for $y'|_s \sim Y'|_s$. As we assume that Y is independent of Y', we have that Y remains intact even conditioned on this fixing. Moreover, since $s = \ell/10$, one can show that with high probability over $y'|_s \sim Y'|_s$, the random variable $X \mid (Y'|_s = y'|_s)$ has min-entropy rate roughly 0.9. Keeping some min-entropy in X conditioned on the fixing of $Y'|_s$ is the reason why we switched from IPMerg_2 to IPMerg_3.

Next, we condition on the fixing of the seed $w' \sim \mathsf{Ext}_{\mathsf{in}}(W, y'|_s)$ to the outer extractor $\mathsf{Ext}_{\mathsf{out}}$ in the definition of $\mathsf{IPMerg}_3(X', Y', W)$. Note that this seed is a deterministic function of W, as we already conditioned on the fixing $Y'|_s = y'|_s$. Thus, conditioning on the fixing $\mathsf{Ext}_{\mathsf{in}}(W, y'|_s) = w'$ does not introduce any correlations between W and the joint distribution of the remaining random variables X, Y, X', Y'. Furthermore, as the output length of $\mathsf{Ext}_{\mathsf{in}}$ is set to s, with high probability over $w' \sim \mathsf{Ext}_{\mathsf{in}}(W, y'|_s)$, the random variable W has min-entropy at least $k - s$ conditioned on the fixing of $\mathsf{Ext}_{\mathsf{in}}(W, y'|_s)$. Thus, by our assumption $k \gg \ell$, only a negligible fraction of the min-entropy of W was lost conditioned on the pair of fixings done so far.

We now fix $y|_s \sim Y|_s$. As $Y|_s$ is uniform and independent of W conditioned on the fixings done so far, and since W has sufficient min-entropy, we have that with high probability over $y|_s \sim Y|_s$, the seed $\mathsf{Ext}_{\mathsf{in}}(W, y|_s)$ to the outer extractor $\mathsf{Ext}_{\mathsf{out}}$ in the definition of $\mathsf{IPMerg}_3(X, Y, W)$ is close to uniform. Informally speaking, this fixing further reduces the min-entropy of X by s, and so X is still left with min-entropy $k - 2s$.

Conditioned on the fixings done so far, $\mathsf{IPMerg}_3(X', Y', W) = \mathsf{Ext}_{\mathsf{out}}(X', w')$ is a deterministic function of X' that consists of s bits. By conditioning on the fixing of $z' \sim \mathsf{IPMerg}_3(X', Y', W)$, we have that X has min-entropy at least $\ell - 3s = 0.7\ell$.

To summarize, with high probability over the fixings done so far, $\mathsf{IPMerg}_3(X', Y', W)$ is fixed to a constant whereas the outer extractor in the definition of $\mathsf{IPMerg}_3(X, Y, W)$ gets X, which has min-entropy 0.7ℓ, as a source, and the independent seed $\mathsf{Ext}_{\mathsf{in}}(W, y|_s)$ which is close to uniform. Thus, with probability close to 1 over the fixings done so far, we have that $\mathsf{IPMerg}_3(X, Y, W)$ is close to uniform whereas $\mathsf{IPMerg}_3(X', Y', W)$ is fixed. By the above discussion, this proves the claim. □

C Multi-Source Extractors and Dispersers From the Literature

In the following table we give a summary of explicit multi-source extractors and dispersers from the literature as well as our contribution. For the sake of readability, whenever possible, the supported min-entropy was written accurately only up to multiplicative constant factors. Further, information concerning the error guarantee and number of output bits is omitted. Any appearance of δ should be considered with a universal quantifier. Unless otherwise stated, δ must be taken as a constant. Any appearance of β is meant under an existential quantifier. Unless specified otherwise, the construction is an extractor (as apposed to a disperser).

Construction	Min-entropy	Number of sources	Comments
[CG88]	$(1/2+\delta)n$	2	
[CG88]	$o(n)$	2	conditional
[BIW06]	δn	$\mathrm{poly}(1/\delta)$	
[BKS+05]	$o(n)$	3	
[BKS+05]	$o(n)$	2	disperser
[Bou05]	$(1/2-\beta)n$	2	
[Raz05]	$(1/2+\delta)n$, $\log n$	2	
[Raz05, Rao09]	δn, $\log n$, $\log n$	3	
[BRSW12]	$2^{(\log n)^{1-\beta}}$	2	disperser
[Rao09, BRSW12]	$\max(k, (\log n)^{10})$	$O(\log n/\log k)$	
[BSZ11]	$0.4n$	2	conditional
[Li11]	$n^{1/2+\delta}$	3	
[Li13b]	$\max(k, (\log n)^4)$	$O(\log(\log n/\log k))$	
[Li13a]	$(\log n)^{2+\delta}$	$O(1/\delta)+O(1)$	
[Li15b]	$(\log n)^{2+\delta}$	$\lceil 14/\delta \rceil + 2$	
[Li15b]	$(\log n)^{12}$	3	
[Coh15a]	$(\log n)^7$	3	
[Coh15a]	δn, $\log n$, $\log\log n$	3	
[Coh15c]	$\mathrm{polylog}\, n$	2	disperser
[CZ15]	$(\log n)^{74}$	2	
[Li15a]	$\mathrm{polylog}\, n$	2	
[Mek15]	$(\log n)^{10}$	2	
This work	$(\log n)^{1+\delta}$	$2/\delta+O(1)$	δ may depend on n
Optimal	$\log n$	2	

ECCC ISSN 1433-8092
http://eccc.hpi-web.de