Electronic Colloquium on Computational Complexity, Report No. 75 (2021)

Affine Extractors for Almost Logarithmic Entropy

Eshan Chattopadhyay*
Cornell University
eshan@cs.cornell.edu

Jesse Goodman*
Cornell University
jpmgoodman@cs.cornell.edu

Jyun-Jie Liao*
Cornell University
jjliao@cs.cornell.edu

June 3, 2021

Abstract

We give an explicit construction of an affine extractor (over $\mathbb{F}_2$) that works for affine sources on n bits with min-entropy $k \geq \log n \cdot (\log \log n)^{1+o(1)}$. This improves prior work of Li (FOCS'16) that requires min-entropy at least $\text{poly}(\log n)$.

Our construction is based on the framework of using correlation breakers and resilient functions, a paradigm that was also used by Li. On a high level, the key sources of our improvement are based on the following new ingredients: (i) A new construction of an affine somewhere random extractor, that we use in a crucial step instead of a linear seeded extractor (for which optimal constructions are not known) that was used by Li. (ii) A near optimal construction of a correlation breaker for linearly correlated sources. The construction of our correlation breaker takes inspiration from an exciting line of recent work that constructs two-source extractors for near logarithmic min-entropy.

1 Introduction

The area of randomness extraction is concerned with producing truly random bits from defective sources of randomness. The motivation for this area stems from the fact that naturally occurring sources of randomness are typically defective, but applications in areas such as cryptography and distributed computing crucially require access to truly uniform bits.

A lot of research has gone into modeling weak sources of randomness, starting with early work of von Neumann [vN51] who considered the problem of extracting randomness from a stream of independent, biased bits. By now, the standard way of measuring the quality of a weak source $\mathbf{X}$ is using the notion of *min-entropy* defined as $H_\infty(\mathbf{X}) = \min_x \log(1/\Pr[\mathbf{X} = x])$. Note that for a distribution $\mathbf{X}$ on $\{0,1\}^n$, we have $0 \leq H_\infty(\mathbf{X}) \leq n$. We define an (n,k)-source to be a distribution on n bits with min-entropy at least k.

We are now ready to define the notion of an extractor for a class of sources. We measure the quality of the output of the extractor using the notion of statistical distance between distributions D_1 and D_2 (on some universe Ω) defined as $|D_1 - D_2| := \frac{1}{2}\sum_{x \in \Omega} |D_1(x) - D_2(x)|$.

Definition 1.1 (Deterministic extractors)**.** *An extractor* $\text{Ext} : \{0,1\}^n \to \{0,1\}^m$ *with error* ϵ *for a class of sources* $\mathcal{X}$ *satisfies the property that for any* $\mathbf{X} \in \mathcal{X}$*, we have* $|\text{Ext}(\mathbf{X}) - \mathbf{U}_m| \leq \epsilon$*.*

*Supported by NSF CAREER award 2045576

ISSN 1433-8092

A folklore result from the 80's rules out the possibility of an extractor (even with a single bit output) for the class of (n, k)-sources, for any $k \leq n-1$. Given this impossibility result, research on random extraction over the last four decades can be broadly classified into two directions: (i) Seeded extraction: the assumption in this setting is that the extractor has access to an independent seed that can be used to extract randomness from source [NZ96]. An impressive line of work have led to efficient constructions of seeded extractors with optimal parameters [LRVW03, GUV09, DKSS13]. (ii) Seedless (or Deterministic) extraction: here one makes additional assumption on the source $\mathbf{X}$ that enables the possibility of randomness extraction. Some examples include the class of bit-fixing sources[1] [CGH+85], sources sampled by a computationally bounded algorithms [TV00, KZ07], and sources that comprise of multiple independent sources [CG88, BIW06].

In this paper we focus on the setting of deterministic extraction[2], and in particular, we study the problem of randomness extraction from *affine sources* defined as follows.

Definition 1.2 (Affine sources)**.** *Fix a finite field $\mathbb{F}$ of size q, and parameters n, k. An $(n, k)_q$-affine source $\mathbf{X}$ is uniform over some (unknown) affine subspace of dimension k in $\mathbb{F}^n$. In other words, there exists linearly independent vectors $v_1, \ldots, v_k$ in $\mathbb{F}^n$ such that $\mathbf{X}$ is the distribution obtained by sampling $\lambda_1, \ldots, \lambda_k$ uniformly and independently from $\mathbb{F}$, and outputting $v_0 + \sum_{i=1}^{k} \lambda_i \cdot v_i$, for some $v_0 \in \mathbb{F}^n$.*

We note that extracting from affine sources falls into the line of investigation that studies extraction from sources sampled by computationally bounded algorithm (since each output bit is computationally restricted to be an affine function of the input randomness, namely the λ_i's).

Thus, an affine extractor $\mathrm{AffExt} : \mathbb{F}^n \to \{0,1\}^m$ for entropy k and error ε is such that for any $(n, k)_q$-affine source $\mathbf{X}$ (over $\mathbb{F}$, where $q = |\mathbb{F}|$), we have $|\mathrm{AffExt}(\mathbf{X}) - U_m| \leq \varepsilon$. For intuition consider the case of $m = 1$: in this case, an affine extractor is simply a 2-coloring of $\mathbb{F}^n$ such that every dimension k affine subspace of $\mathbb{F}^n$ is almost evenly colored.

A probabilistic argument shows the existence of excellent affine extractors: for example, setting the error ε to a constant and the output $m = k - O(1)$, a random function is an affine extractor for min-entropy $k > 2 \log n$. However, for applications it is desirable to find explicit constructions of such extractors (i.e., an extractor that has running time which is polynomial in the parameters n, q).

Prior work on explicitly constructing affine extractors can be classified into two settings depending on the size of the field:

- Large field setting ($q = \mathrm{poly}(n)$): In this setting, Gabizon and Raz [GR08] in fact constructed an affine extractor for lines (i.e., $k = 1$)[3]. More generally, they showed how to extract most of the entropy out of any $(n, k)_q$-affine source in this large field setting. A construction with improved error was given by Bourgain, Dvir, and Leeman [BDL16] assuming q is a prime, and further that $q - 1$ does not have too many prime factors.

- Small field setting ($q = O(1)$): The task of constructing affine extractors is generally more challenging as the field size gets smaller. In the setting of $q = O(1)$, Bourgain [Bou07], and subsequent works of Yehudayoff [Yeh11] and Li [Li11] gave explicit constructions for

[1] such sources have an unknown set of random coordinates

[2] see the excellent survey by Shaltiel [Sha04] for results on seeded extraction

[3] when n is large enough compared to q, the Hales-Jewett theorem rules out the possibility of an extractor for lines.

$k \geq n/\sqrt{\log \log n}$. DeVos and Gabizon [DG10] obtained a trade-off between the field size and entropy, and gave explicit affine extractors for $q = \Omega((n/k)^2)$ for fields with characteristic $\Omega(n/k)$, thus requiring linear entropy (i.e., $k = \Omega(n)$) for extraction from affine sources on fields with constant characteristic.

A vastly improved result was obtained by Li [Li16] who gave an explicit affine extractor that works for $q = 2$ (which is generally considered the hardest setting) and $k \geq C(\log n)^C$, for some large enough constant C.

1.1 Our Result

Our main result is a further improvement over the result of Li [Li16], and thus nearly matching the random construction, in terms of the min-entropy requirement.

Theorem 1. *For any constant $\varepsilon > 0$, there exists a constant $C > 0$ such that for all $n, k \in \mathbb{N}$ such that $k \leq n$, there exists an efficient construction of an extractor* $\mathrm{AffExt} : \mathbb{F}_2^n \to \{0,1\}$ *with error ε for $(n,k)_2$-affine sources with min-entropy $k \geq C \cdot \log n \cdot \log \log n \cdot (\log \log \log n)^6$.*

One drawback of our construction compared to [Li16] is the error of our extractor: while we can only achieve constant error, the construction in [Li16] achieves polynomially small error. We leave it as an interesting open question to reduce the error of our extractor construction.

1.2 Applications of Affine Extractors

The class of affine sources naturally generalize (oblivious) bit-fixing sources [CGH+85], where a bit-fixing source has some unknown set of random coordinates and the other coordinates are fixed to constants. Extractors for bit-fixing sources have applications in exposure resilient cryptography [CGH+85, KZ07]. In fact, this generalization of bit-fixing sources to affine sources played a key role in obtaining the best known extractors for bit-fixing sources [Rao09].

Affine extractors have found applications in deterministic extraction from many other models of weak sources. Viola [Vio14] proved that sources sampled by small circuits are close to a convex combination of affine sources (and thus an affine extractor can be used to extract from such *circuit sources*). In a recent work, Chattopadhyay and Goodman [CG20] proved a similar result for sources sampled by algorithms with limited memory [KRVZ06]. Such small-space sources are known to capture a wide variety of weak source models that have been considered for randomness extraction, including finite Markov chain sources [Blu86], symbol-fixing sources [KZ07], and (short) independent sources. Ben-Sasson and Zewi [BSRZ15] demonstrated another application by showing how to use any affine extractor to construct low-error two-source extractors in a black-box way under plausible conjectures in additive combinatorics. Cohen and Tal [CT15] used affine extractors to extract randomness from *variety sources*, which are distributions that are uniform on the set of common zeroes of a system of polynomial equations. Thus, affine extractors provide a unifying way to construct extractors for a wide range of well-studied models of weak sources.

Finally, another application of affine extractors over $\mathbb{F}_2$ is to circuit lower bounds–a breakthrough work of Find, Golovnev, Hirsch, and Kulikov [FGHK16] prove the best known Boolean circuit lower bound (roughly $3.01n$) for an affine extractor (with 1 bit output) that works for entropy $k = o(n)$.[4]

[4] in fact, they prove the lower bound for a weaker object known as an affine disperser which roughly satisfies the property that it is non-constant on any affine subspace of dimension k

2 Proof Overview

On a very high level, our construction follows the framework in [CZ19], which has been used to construct deterministic extractors in many recent works. The framework works as follows: given a source $\mathbf{X}$, we first convert it into a non-oblivious bit-fixing (NOBF) source, which is a source on $N = \text{poly}(n)$ bits such that $N - N^{\delta}$ of them are t-wise independent (see Definition 4.25). Then we apply an extractor for NOBF source to get the output.

A general strategy to construct an NOBF source from multiple independent sources was initiated in [Li13]. The strategy works by first taking a *strong seeded extractor*, which is a function Ext that takes d bits of extra randomness (i.e. a *seed*) $\mathbf{S}$ and convert $\mathbf{X}$ into a close-to-uniform string $\text{Ext}(\mathbf{X}, \mathbf{S})$, with high probability over the seed $\mathbf{S}$ (see Definition 4.5). Since in reality we do not have such a seed $\mathbf{S}$, we enumerate over all $D = 2^d$ possibility of the random seed and get a somewhere-random source (SR-source), which is a collection of D different strings such that most of them are close to uniform. However, note that the strings which are close to uniform are arbitrarily correlated with each other. The second step is to take another independent source to "break the correlation" between these uniform strings and make them t-wise independent. A function which can complete this task is called a correlation breaker [Coh16a]. Recent constructions of such object employ a technique known as alternating extraction [DP07], which uses strong seeded extractor as a building block.

The setting of affine extractor is trickier since there is only one source. At a first glance it doesn't seem like the above framework can be used to construct an affine extractor. However, Li [Li16] showed that this framework still can be used based on the crucial observation, originating in the work of Rao [Rao09]: if the strong seeded extractor Ext that we use is a linear function, for *every* fixing of the seed (such extractors are called *linear seeded extractor*), then there is still some "implicit independence" between the output $\mathbf{Y} = \text{Ext}(\mathbf{X})$ and the original source $\mathbf{X}$. Specifically, $\mathbf{X}$ can be written in the form $\mathbf{A} + \mathbf{B}$ such that $\mathbf{A}$ has some entropy and is independent of $(\mathbf{B}, \mathbf{Y})$. Then [Li16] showed that if we again use linear seeded extractors to construct the correlation breaker then it's possible to exploit this implicit independence.

However, the affine extractor in [Li16] requires $\text{polylog}(n)$ entropy because of the following reasons. First, to extract from a NOBF source, [Li16] used the derandomized Ajtai-Linial resilient function [CZ19, Mek17, Li16] in the last step, which requires the source to have poly-logarithimic entropy. Second, the correlation breaker in [Li16] also requires $\Omega(\log^2 n)$ entropy to work. In fact, even the state-of-the-art correlation breakers for independent sources [CGL16] required $\Omega(\log^2 n)$ entropy at the time, while in [Li16] the correlation breaker needs to work for two sources with linear correlation, which is even harder. Finally, many steps in this construction require a strong linear seeded extractor. However, the known constructions of strong linear seeded extractor usually require at least $\text{polylog}(n)$ entropy. Specifically, note that this framework heavily relies on strong linear seed extractor for two different purposes:

1. To convert $\mathbf{X}$ into a SR-source,

2. To construct the correlation breaker.

In both cases we need the error of extractor to be $1/\text{poly}(n)$, and in the first case we further need the seed to have length $d = O(\log(n))$ to make sure that there are only $2^d = \text{poly}(n)$ possibilities of seed to enumerate (since otherwise the running time of the affine extractor we construct will not be polynomial). The most commonly used strong linear seeded extractor is perhaps the Trevisan's

extractor [Tre01, RRV02], but it requires a seed of length at least $\log^2(n)$ in this setting. The extractor constructed in [Li16], while having $O(\log(n))$ seed length, requires the source to have $\log^c(n)$ entropy for some constant $c > 4$.

2.1 Bypassing the linear seeded extractor barrier

To solve the problem of not having a good enough linear seeded extractor, we take different approaches in the two cases. We first discuss the task of turning $\mathbf{X}$ into a SR-source, and explain the construction of our correlation breaker in Section 2.2.

In both cases, our starting point is a simple construction of strong linear seeded extractor which works as follows. To extract m uniform bits, our first step is to apply a strong lossless condenser on $\mathbf{X}$: this is a function that takes a seed and converts $\mathbf{X}$ into a shorter source $\mathbf{X}'$ of length $O(m)$ while still having roughly m bits of entropy. Using GUV condenser [GUV09], this step requires a seed of length $O(\log(n/\varepsilon))$ where ε is the error, and such condenser can also be made linear [Che10]. Our second step is to apply a linear universal hash function on $\mathbf{X}'$ and get a m-bit uniform string by Leftover Hash Lemma [HILL99]. This "condense-then-hash" extractor has optimal entropy requirement, but the seed length is $O(m + \log(n))$.

Now recall that in the first step we need the seed length to be $O(\log n)$, which means using this condense-then-hash extractor, we can only extract a string $\mathbf{Y}$ of length $O(\log n)$. However, this is not enough for a correlation breaker to work, even if we use the state-of-the-art correlation breaker for independent sources [Li19] (recall, we have to deal with the harder case of the linearly correlated sources).

To solve this problem, our observation is that while $|\mathbf{Y}| = O(\log n)$ is not enough for a correlation breaker to work, we require $\mathbf{Y}$ to be only slightly longer. In particular, we need $m = |\mathbf{Y}| = O(c(n) \cdot \log n)$ for some slow growing function $c = c(n)$. (We will see that we can take $c = \log\log(n)$ when discuss correlation breaker in Section 2.2.) Our idea is to use a recursive approach based on block-source extraction combined with a error reduction trick at the end, as follows:

- As before, we first use the GUV condenser to condense the source $\mathbf{X}$ into a source $\mathbf{X}'$ of length $n' = O(m)$ and entropy $0.9n'$. In other word, $\mathbf{X}'$ has *entropy rate* 0.9. This requires a seed of length $O(\log(n/\varepsilon))$.

- Next, we cut $\mathbf{X}'$ into two blocks, and a standard argument shows that each block still have entropy rate 0.8, even when conditioned on the other block. Then again we apply GUV condenser on each block to condense the entropy rate to 0.9, but this time we use one seed to condense both blocks. Intuitively this works because GUV condenser is strong (which can be considered as "success with high probability" and hence we can apply union bound on both blocks). Besides, note that this time the seed length is only $O(\log(m/\varepsilon))$.

- We again divide each block into two halves and get four blocks in total, and use one extra seed $O(\log(m/\varepsilon))$ to condense all four blocks. By repeat this step for $\log(c)$ times we eventually get c blocks, each having entropy roughly $O(\log(n))$.

- Finally we use another seed of length $O(\log(n/\varepsilon))$ to sample a linear hash function and extract from every block. The total seed length is $O(\log(n) + \log(c)\log(m/\varepsilon))$, which is

$O(\log(n)+\log(c)\log(1/\varepsilon))$ since m is short.[5] Now we get an extractor of seed length $O(\log(n))$, but with error $\varepsilon = n^{-O(1/\log(c))}$, which is slightly larger than what we need.

- To solve this problem, we apply the error reduction scheme in [BDT17], which reduces the error to $1/\text{poly}(n)$ but only increase the seed length by a constant factor. A drawback of this scheme is for every seed we get A different outputs such that only one of them is guaranteed to be uniform. In other word, we get a *somewhere random extractor* instead of an extractor.

We note that the weaker notion of a somewhere random extractor (instead of an extractor) suffices in our scheme of constructing affine extractors. Informally, we follow the approach of [BDT17], and apply correlation breakers on all outputs of the somewhere random extractor. Using idea in [Coh16a], we can simply merge these strings by taking the parity of these strings after we break their correlation.

Another possible concern is the following. In [BDT17] they start from error $1/\text{poly}(n)$, and reduce it to $1/n^C$ for any constant C, and thus get the parameter A (the number of different outputs) to be a constant. In our setting, we start from error slightly larger than $1/\text{poly}(n)$, and thus require $A = O(\log(c))$. So, in our construction, we need a correlation breaker which breaks the correlation between more strings. This implies that we need the output of our somewhere extractor, $\mathbf{Y}$ to be longer, and thus it increases the seed length of our extractor correspondingly. Nevertheless, we only need to increase the length of $\mathbf{Y}$ by a factor of A^2. Since A only has logarithmic dependence on the length of $\mathbf{Y}$, this will not be a problem.

2.2 Correlation breaker for linearly correlated source

In this section, we give a brief description of the main ideas that go into our correlation breaker construction, assuming some familiarity with techniques that are used in recent constructions of correlation breakers. In Section 3, we present a much more detailed account of our correlation breaker construction.

Many recent works successfully construct correlation breakers for independent sources with error $1/\text{poly}(n)$ which only require $\log^{1+o(1)}(n)$ entropy [CL16a, Coh16b, Li17, Coh17, Li19]. In fact, the state-of-the-art construction by Li [Li19] only requires $O(\log n \cdot \frac{\log\log n}{\log\log\log n})$ entropy. As we pointed out above, if we try to adapt these constructions to the setting of linearly correlated source using Trevisan's extractor or the linear seeded extractor in [Li16], the entropy requirement is at least $\log^2(n)$.

A natural idea is to use the linear seeded extractor, based on the "condense-then-hash" approach discussed in Section 2.1, in the correlation breaker construction. It turns out that this is almost sufficient, except for one problem: the seed length of the condense-then-hash extractor is larger than the output length by a constant factor. This is a problem in all known approaches of constructing correlation breaker. Specifically, all the constructions of correlation breaker is based on alternating extraction which works as follows. First, take a small slice of $\mathbf{Q}_1$ from $\mathbf{Y}$, and compute $\mathbf{W}_1 = \text{Ext}(\mathbf{X}, \mathbf{Q}_1)$. Next, compute $\mathbf{Q}_2 = \text{Ext}(\mathbf{Y}, \mathbf{W}_1)$. Then compute $\mathbf{W}_2 = \text{Ext}(\mathbf{X}, \mathbf{Q}_2)$, and so on. Now note that if we take Ext to be the condense-then-hash extractor, then after $O(\log\log n)$ rounds the length actually decreases by a $O(\text{polylog}(n))$ factor.

[5] In fact, our final seed length is $O(\log(n) + \log^2(c)\log(1/\varepsilon))$, since we need to condense each block into entropy rate $1 - 1/\log(c)$ to make sure that we don't lose too much entropy when dividing blocks. We ignore this in the proof overview for simplicity.

To solve the above problem, we observe that it's actually not necessary to use a strong linear seeded extractor all the time. To see why this is the case, first we recap why a linear seeded extractor needs to be used when we consider linearly correlated source. Recall that $\mathbf{X}$ can be written as $\mathbf{A} + \mathbf{B}$, where $\mathbf{A}$ is independent of $\mathbf{B}, \mathbf{Y}$. Now let LExt denote a strong linear seeded extractor, and Ext be another strong seeded extractor. If we take a slice $\mathbf{Q}$ from $\mathbf{Y}$ and compute $\mathbf{W} = \mathrm{LExt}(\mathbf{X}, \mathbf{Q})$, then $\mathbf{W} = \mathbf{W}_A + \mathbf{W}_B$, where $\mathbf{W}_A = \mathrm{LExt}(\mathbf{A}, \mathbf{Q})$ and $\mathbf{W}_B = \mathrm{LExt}(\mathbf{B}, \mathbf{Q})$. Now note that conditioned on the fixing of $\mathbf{Q}$, $\mathbf{W}_A$ is uniform with high probability and is independent of $\mathbf{W}_B$. Therefore $\mathbf{W}$ is also uniform with high probability. When extracting from the $\mathbf{Y}$ side, we again use the fact that $\mathbf{W}$ can be written as $\mathbf{W}_A + \mathbf{W}_B$ where $\mathbf{W}_A$ is uniform and independent of $(\mathbf{W}_B, \mathbf{Y})$. Note that conditioned on $\mathbf{W}_B$, $\mathbf{W}$ should still be uniform and independent of $\mathbf{Y}$, and $\mathbf{Y}$ only loses a small amount of entropy (proportional to the length of $\mathbf{W}_B$). This ensures that $\mathrm{Ext}(\mathbf{Y}, \mathbf{W})$ is still uniform with high probability over $\mathbf{W}$ if $\mathbf{Y}$ has enough entropy. Now observe that this argument doesn't require Ext to be linear.

Based on this observation, we can do the alternating extraction in an "asymmetric" way: when we extract from $\mathbf{X}$, we use the condense-then-hash extractor, which takes a d_X-bit seed and output a uniform string with d_Y bits. Note that $d_X = c \cdot d_Y$ for some constant $c > 1$. Then when we extract from $\mathbf{Y}$, we use a optimal non-linear strong seeded extractor (e.g. the GUV extractor [GUV09]) which takes a d_Y-bit seed and output d_X bits. Now there is one remaining problem: the recent constructions of correlation breaker usually use more complicated sub-protocols as the building blocks. All these sub-protocols are also based on alternating extractions, and hence it is plausible that they can be instantiated with the condense-then-hash extractor using the idea above. However, it appears cumbersome to rework all the sub-protocols in this way, and in particular it is not clear how to combine them together without loss of parameters. Nevertheless, we observe that the idea above works for a more general setting of sub-protocols. That is:

- If a sub-protocol f is a function on $\mathbf{X}$ and takes a seed from $\mathbf{Y}$, f should be linear.
- If a sub-protocol g is a function on $\mathbf{Y}$ and takes a seed from $\mathbf{X}$, g doesn't need to be linear, but should work properly when $\mathbf{Y}$ is a weak source.

Usually the second case is easier to deal with: the same construction (or a slight change) of sub-protocols should still work most of the time. Therefore the construction will be simple if we minimize the amount of sub-protocols in the first case.

In fact, in our construction, all the functions we need in the first case are simply strong seeded extractors, and thus we can replace them with the condense-then-hash extractor. However, it is still not clear how to use previous results in a black-box fashion. Thus, we give a much more detailed explanation of the main ideas of our correlation breaker construction in Section 3. We formally present and prove our construction in Section 6.

2.3 Extracting from NOBF source

The final step is to extract from an NOBF source $\mathbf{Z}$ on N bits that we obtain from the affine source $\mathbf{X}$, first by converting it into an SR-source and then applying the correlation breaker. The derandomized Ajtai-Linial function [AL93, CZ19, Mek17] was shown to be an extractor for such sources with at least $N - N/\log^2 N$ good bits. However, this extractor needs min-entropy at least polylog(n) in the NOBF source, since good bits are required to be poly$(\log n)$-wise independent.

To circumvent this barrier, we recall a result of Viola [Vio14], who proved that majority can extract from an $O(1)$-wise independent NOBF source with constant error. Thus, this is better suited for our goal of constructing affine extractors for near logarithmic entropy. In fact, this resilient function is also used in recent constructions of two-source extractors that work for near logarithmic min-entropy based on the two-source framework of Ben-Aroya, Doron, and Ta-Shma [BDT17]. However, to use the majority function we require the NOBF source $\mathbf{Z}$ to have at least $N - N^{\delta}$ good bits, for $\delta < 1/2$. In fact, as pointed out in [BDT17], $\delta = 1/2$ is actually a barrier in the two-source setting if we create the SR-source using a seeded extractor, and their approach is based on the use of condensers for this task.

Interestingly, in our setting to produce such a SR-source from the affine source $\mathbf{X}$, since we just need a linear-seeded extractor that is required to work for affine sources, one can use the probabilistic method to show that a random construction can be used. However, we do not have explicit constructions of such optimal linear seeded extractors. Instead, we show that our somewhere random extractor from Section 2.1 can be used to construct the SR source with desired parameters.

2.4 Summary of construction

Finally we summarize our construction. Given an affine source $\mathbf{X}$, we run the following steps to extract a bit:

1. Take a strong linear seeded somewhere random extractor LSRExt with seed length $d = O(\log n)$, and for every $s \in \{0,1\}^d$ compute $(\mathbf{Y}_{s,1}, \ldots, \mathbf{Y}_{s,A}) := \text{LSRExt}(\mathbf{X}, s)$.
2. Take a correlation breaker ACB and compute $\mathbf{Z}_{s,j} := \text{ACB}(\mathbf{X}, \mathbf{Y}_{s,j}, (s,j))$ for every $s \in \{0,1\}^d, j \in [A]$. Here (s,j) serves as an "advice" to ACB (more details can be found in the next section).
3. For every $s \in \{0,1\}^d$, compute $\mathbf{P}_s := \bigoplus_{j=1}^{A} \mathbf{Z}_{s,j}$.
4. Compute the majority of $\mathbf{P}_1, \ldots, \mathbf{P}_{2^d}$.

Organization. In Section 3, we present a detailed explanation of the main ideas that are in our correlation breaker construction. Section 4 contains some notations and some standard results from previous works which we frequently use throughout the paper. Section 5 contains the construction of the strong linear seeded extractor with error slightly larger than $1/\text{poly}(n)$, along with the error reduction scheme. We present the formal construction and proof of our correaltion breaker in Section 6. Finally, in Section 7 we prove our main theorem, where we show how these building blocks are combined together to yield an affine extractor.

3 Correlation-Breaking Games

In this section, we present detailed explanation of the main ideas used in our correlation breaker. We explain these ideas using a few (related) two-party games that we introduce below. We hope this discussion provides more intuition to the readers to parse our more involved proofs in Section 6.

Recall that our goal is to use a source $\mathbf{X}$, that is independent or linearly correlated (defined in Section 3.9), to break the correlation between $\text{poly}(n)$ strings $\mathbf{Y}_1, \ldots, \mathbf{Y}_D$ and make them t-wise independent. The high level idea is that we compute the same function f, that is called as a

correlation breaker, on $\mathbf{X}$ and every $\mathbf{Y}_i$, to produce strings $\mathbf{Z}_1 = f(\mathbf{X}, \mathbf{Y}_1), \ldots, \mathbf{Z}_D = f(\mathbf{X}, \mathbf{Y}_D)$. The property we desire from f is the following: if there is a set $T \subset [D]$, such that for any $i \in T$, $\mathbf{Y}_i$ is uniform, then for any distinct $i_1, \ldots, i_t$ from T, the random variable $\mathbf{Z}_{i_1}$ looks uniform conditioned on $\{\mathbf{Z}_{i_j}\}_{j=2}^t$. For constructing the function f, it helps to think of it as a two-party game that we discuss in detail below.

Given the above discussion, it is enough to consider the following setting: let $\mathbf{Y}$ be a uniform, and let $\mathbf{Y}^1, \mathbf{Y}^2, \ldots, \mathbf{Y}^t$ be random variables that are arbitrarily correlated with $\mathbf{Y}$. $\mathbf{Y}^i$ is called the *i-th tampering* of $\mathbf{Y}$. As before, let $\mathbf{X}$ be a random variable that is independent (or linearly correlated) with $\mathbf{Y}, \{\mathbf{Y}^i\}_{i=1}^t$. We want to construct a correlation breaker f with the guarantee that the output $f(\mathbf{X}, \mathbf{Y})$ is uniform conditioned on the outputs computed using all of its t tampering $\{f(\mathbf{X}, \mathbf{Y}^i)\}_{i=1}^t$. Before discussing how to construct correlation breakers, we first introduce some convenient notation.

As alluded above, a useful perspective is to think of the computation of the correlation breaker as a two-party communication between $\mathbf{X}$ and $\mathbf{Y}$. This is because of the following reason: if $\mathbf{Z}$ is the transcript of a two-party communication between $\mathbf{X}$ and $\mathbf{Y}$, then $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ forms a Markov chain. This ensures that at any point of the computation we have two independence sources $\mathbf{X}, \mathbf{Y}$ to work with, conditioned on any fixing of $\mathbf{Z}$. Therefore each time one party can send a message as an independent randomness to help the other party complete some tasks. However, $\mathbf{Z}$ leaks some information about $\mathbf{X}$, and $\mathbf{Y}$; so ideally we want the length of $\mathbf{Z}$, i.e. the *communication complexity*, to be as small as possible.

We introduce some convenient notation.

Notations: Throughout this paper, we use $\mathbf{Y}^{[t]}$ to denote all the t tampering of $\mathbf{Y}$, and for any set $S \subseteq [t]$ such that $S = \{i_1, \ldots, i_k\}$, we also use $\mathbf{Y}^S$ to denote a collection $\mathbf{Y}^{i_1}, \ldots, \mathbf{Y}^{i_k}$. For any random variable $\mathbf{R}$ computed, we use $\mathbf{R}^i$ to denote the i-th tampered version of $\mathbf{R}$ which is computed using $\mathbf{Y}^i$.

In Sections 3.1 to 3.8, we discuss the two-party games and relevant techniques that are used in recent constructions of correlation breakers for independent sources, in increasing order of complexity. Along the way, we explain how we adapt some of these techniques to construct our correlation breaker in the linearly correlated setting. In Section 3.9, we define the correlation breaking game in the linearly correlated setting, and in Section 3.10 we summarize our construction.

We start with describing the correlation breaking game in the independent source setting, in the more general case that $\mathbf{X}$ also has its tampered versions $\mathbf{X}^1, \ldots, \mathbf{X}^t$.

3.1 Correlation-breaking game for independent sources

The setup is as follows: Quentin has a source $\mathbf{X}$ which is uniform and Wendy has a source $\mathbf{Y}$ which has some entropy. Further, Quentin and Wendy hold some tampered sources $(\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^t)$ and $(\mathbf{Y}^1, \mathbf{Y}^2, \ldots, \mathbf{Y}^t)$ respectively such that $\mathbf{X}^{[t]}$ can be arbitrarily correlated with $\mathbf{X}$ and $\mathbf{Y}^{[t]}$ can be arbitrarily correlated with $\mathbf{Y}$. The assumption is that $(\mathbf{X}, \mathbf{X}^{[t]})$ is independent of $(\mathbf{Y}, \mathbf{Y}^{[t]})$. Quentin and Wendy are going to run a two-party game as follows. The game starts with a public transcript $\mathbf{Z}$ and some "tampered transcripts" $(\mathbf{Z}^1, \ldots, \mathbf{Z}^t)$ which are all empty at the beginning. They need to choose a deterministic two-party communication protocol P, which is a sequence of deterministic function $(f_1, g_1, f_2, g_2, \ldots)$ so that in the first round Quentin sends a message $\mathbf{Q}_1 := f_1(\mathbf{X}, \mathbf{Z})$, and then $\mathbf{Q}_1$ is added to the transcript $\mathbf{Z}$. In the next round Wendy sends a message $\mathbf{W}_1 := g_1(\mathbf{Y}, \mathbf{Z})$, and then $\mathbf{W}_1$ is added to the transcript $\mathbf{Z}$. They keep sending messages computed with $f_2, g_2, \ldots$ until the protocol ends. However, there are also t "tampered communications" run in parallel.

When Quentin sends $\mathbf{Q}_1 := f_1(\mathbf{X}, \mathbf{Z})$, a tampered message $\mathbf{Q}_1^j := f_1(\mathbf{X}^j, \mathbf{Z}^j)$ is also sent and added to the tampered transcript $\mathbf{Z}^j$ for every $j \in [t]$. Similarly when Wendy sends a message there will also be t tampered messages sent simultaneously. At the end of the protocol, one of the party computes a output, which we denote as $\mathbf{R} = P(\mathbf{X}, \mathbf{Y})$, and $\mathbf{R}$ will not be added to the transcript $\mathbf{Z}$. Quentin and Wendy win the game if $\mathbf{R}$ is uniform conditioned on all the tampered output $\mathbf{R}^{[t]}$ where $\mathbf{R}^j = P(\mathbf{X}^j, \mathbf{Y}^j)$ and all the (tampered) transcripts $\mathbf{Z}, \mathbf{Z}^{[t]}$.

3.2 Alternating extraction

It is easy to see that Quentin and Wendy can never win the correlation-breaking game if $\mathbf{X}^1 = \mathbf{X}$ and $\mathbf{Y}^1 = \mathbf{Y}$, since this implies $\mathbf{R}^1 = \mathbf{R}$. However, it is possible to win a weaker game which we call a *look-ahead game*. In this game, Quentin and Wendy need to output multiple messages $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_\ell$. We say a message $\mathbf{R}$ has look-ahead property [DW09] if $\mathbf{R}$ is uniform conditioned on all the transcripts $\mathbf{Z}, \mathbf{Z}^{[t]}$ (but not necessarily on the tampered output $\mathbf{R}^{[t]}$). Quentin and Wendy win the look-ahead game if the output $\mathbf{R}$ has the look-ahead property. Winning the look-ahead game with one output is actually not very interesting since Quentin can just output a prefix of $\mathbf{X}$ while all the transcripts are empty. Now consider the ℓ-look-ahead game so that Quentin and Wendy need to *sequentially* compute and send $\mathbf{R}_1, \ldots, \mathbf{R}_\ell$ such that every $\mathbf{R}_i$ satisfies the look-ahead property at the moment it is computed. Note that the transcripts of $\mathbf{R}_i$ contain the previous outputs $\mathbf{R}_1, \ldots, \mathbf{R}_{i-1}$ and their tampered versions. This game is winnable with the alternating extraction [DP07] protocol, which works as follows. Observe that at any moment of the game, $(\mathbf{X}, \mathbf{X}^{[t]}) \leftrightarrow (\mathbf{Z}, \mathbf{Z}^{[t]}) \leftrightarrow (\mathbf{Y}, \mathbf{Y}^{[t]})$ is a Markov chain. Moreover, conditioned on $(\mathbf{Z}, \mathbf{Z}^{[t]})$, $\mathbf{Y}$ only loses roughly $(t+1)\ell$ bits of entropy where ℓ is the total length of messages from Wendy (and its tampered version). Therefore, if $\mathbf{Y}$ has high enough entropy at the beginning, it will still have some entropy remaining if the total length of messages (i.e. the *communication complexity*) from Wendy is not too long. If Quentin sends a string $\mathbf{Q}_1 = f(\mathbf{X}, \mathbf{Z})$ which is uniform condition on $(\mathbf{Z}, \mathbf{Z}^{[t]})$ (i.e. $\mathbf{Q}_1$ satisfies the look-ahead property), then Wendy can also get a uniform string $\mathbf{W}_1 = \mathrm{Ext}(\mathbf{Y}, \mathbf{Q}_1)$ conditioned on $(\mathbf{Z}, \mathbf{Z}^{[t]})$ by applying a seeded extractor. Moreover, if Ext is strong, $\mathbf{W}_1$ remains uniform even after $(\mathbf{Q}_1, \mathbf{Q}_1^{[t]})$ is added to $(\mathbf{Z}, \mathbf{Z}^{[t]})$, since the entropy of $\mathbf{W}_1$ purely comes from $\mathbf{Y}$. Therefore, $\mathbf{W}_1$ also satisfies the look-ahead property. This observation gives the alternating extraction protocol: Quentin sends $\mathbf{Q}_1$ which is a prefix of $\mathbf{X}$, Wendy sends $\mathbf{W}_1 := \mathrm{Ext}(\mathbf{Y}, \mathbf{Q}_1)$, Quentin sends $\mathbf{Q}_2 := \mathrm{Ext}(\mathbf{X}, \mathbf{W}_1)$, and so on. As long as $\mathbf{X}$ and $\mathbf{Y}$ still have enough entropy left, every message sent in this protocol satisfies the look-ahead property. The entropy requirement is roughly $O(\ell t \log n)$. (This $\log n$ is the seed length of each randomness extractor.)

3.3 Breaking correlation with advice

Now we change the correlation-breaking game a little bit to get a actually winnable game. Suppose Quentin and Wendy further get some advice $(\alpha, \alpha^1, \ldots, \alpha^t) \in [2^a]$ such that $\alpha \neq \alpha^j$ for every $j \in [t]$. Then the actual communication is run with a protocol P_α chosen from a family of protocols $\{P_1, \ldots, P_{2^a}\}$. Moreover, for every $j \in [t]$, the j-th tampered communication is run with the protocol P_{α^j}. Since the actual protocol is different from all the tampered protocol, now it's possible that $\mathbf{R}$ is independent of $\mathbf{R}^{[t]}$ even if $\mathbf{X} = \mathbf{X}^j$ and $\mathbf{Y} = \mathbf{Y}^j$ for every $j \in [t]$. This is called *correlation breaker with advice* [CGL16, Coh16c]. In fact, consider the family of protocol such that P_i runs alternating extraction for i rounds and output the i-th message from Wendy. Then if $\alpha > \alpha^j$ for every $j \in [t]$, the output $\mathbf{R} := P_\alpha(\mathbf{X}, \mathbf{Y})$ is actually independent of $\mathbf{R}^{[t]} := P_{\alpha^{[t]}}(\mathbf{X}^{[t]}, \mathbf{Y}^{[t]})$ by the

look-ahead property. This idea first came in [Li13]. When the order of advice is unknown, there was a beautiful idea in [Coh16a] called "flip-flop" which resolves the issue. However, note that with only this idea $\mathbf{X}$ and $\mathbf{Y}$ need entropy roughly $O(2^a \cdot t \log n)$, and hence the protocol is only good enough when a is small (e.g. 1 bit).

3.4 Merging independence

To reduce the entropy requirement, a nice property of strong seeded extractor comes to the rescue. The property is called *independence preserving*. Suppose there is a source $\mathbf{Y}$ and a seed $\mathbf{Q}$, and each of them have a tampered version $\mathbf{Y}^1, \mathbf{Q}^1$. Now suppose $\mathbf{Q}$ is uniform conditioned on $\mathbf{Q}^1$. Then if one applies a seeded extractor and get $\mathrm{Ext}(\mathbf{Y}, \mathbf{Q})$, this string is also uniform conditioned on $\mathrm{Ext}(\mathbf{Y}^1, \mathbf{Q}^1)$. Too see why this is true, note that when conditioned on $\mathbf{Q}^1$ and $\mathrm{Ext}(\mathbf{Y}^1, \mathbf{Q}^1)$, $\mathbf{Q}$ is still uniform and independent of $\mathbf{Y}$, while $\mathbf{Y}$ only loses a small amount of entropy. Therefore $\mathrm{Ext}(\mathbf{Y}, \mathbf{Q})$ is still uniform. In other word, $\mathrm{Ext}(\mathbf{Y}, \mathbf{Q})$ preserves the independence of $\mathbf{Q}$ from its tampering $\mathbf{Q}^1$. This idea was used in [Li13] to get a better entropy requirement, which is only linear in a. We will see more details later. Besides, Ext can also preserve the independence on the other side: if $\mathbf{Y}$ has high entropy conditioned on $\mathbf{Y}^1$, then $\mathrm{Ext}(\mathbf{Y}, \mathbf{Q})$ is uniform conditioned on $\mathrm{Ext}(\mathbf{Y}^1, \mathbf{Q}^1)$. This can be proved using a similar argument. Based on this observation, [CS16] suggested a protocol which works as follows.[6] Suppose Quentin has two uniform strings $\mathbf{X}_1, \mathbf{X}_2$ such that either $\mathbf{X}_1$ is independent of $\mathbf{X}_1^1$ or $\mathbf{X}_2$ is independent of $\mathbf{X}_2^1$. Let $\mathbf{Q}_1$ be a prefix of $\mathbf{X}_1$. Now they can do two rounds of alternating extraction to compute $\mathbf{W}_1 = \mathrm{Ext}(\mathbf{Y}, \mathbf{Q}_1)$ and then $\mathbf{R} = \mathrm{Ext}(\mathbf{X}_2, \mathbf{W}_1)$. The output $\mathbf{R}$ should be independent of $\mathbf{R}^1$ by the independence-preserving property. In other word, they *merge* $\mathbf{X}_1, \mathbf{X}_2$ and *preserve the independence* of $\mathbf{X}_1$ or $\mathbf{X}_2$ from their tampered version. It's also possible to merge ℓ strings $\mathbf{X}_1, \ldots, \mathbf{X}_\ell$ by doing more rounds of alternating extraction [CL16a]. This protocol is called a non-malleable independence-preserving merger (NIPM) [CS16, CL16a].

In this paper we show that a stronger property holds, which we call *independence merging*. That is, suppose there exist $S, T \subseteq [t]$ such that $\mathbf{X}$ is uniform and independent of $\mathbf{X}^S$, and $\mathbf{Y}$ has high min-entropy conditioned on $\mathbf{Y}^T$. Then by taking a prefix $\mathbf{Q}$ of $\mathbf{X}$ and compute $\mathbf{W} = \mathrm{Ext}(\mathbf{Y}, \mathbf{Q})$, $\mathbf{W}$ is actually uniform conditioned on $\mathbf{W}^{S \cup T}$. In other word, the strong seeded extractor Ext *merges the independence* of $\mathbf{X}$ and $\mathbf{Y}$ from their tampered versions. To prove this, we can simply apply the argument in both cases of independence preserving together. Therefore, we can use the same alternating extraction protocol to merge the independence of $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\ell$ from their tampered versions, even if the independence is scattered on multiple different $\mathbf{X}_i$. This stronger property will help us deal with the t-tampering case directly.

3.5 Strongness of protocols

Suppose $\mathbf{R}$ is the output of some two-party communication protocol $P(\mathbf{X}, \mathbf{Y})$, $\mathbf{Z}$ is the transcript, and suppose $\mathbf{R}$ is computed by Wendy using some deterministic function $g(\mathbf{Y}, \mathbf{Z})$. Then $(\mathbf{X}, \mathbf{X}^{[t]}) \leftrightarrow (\mathbf{Z}, \mathbf{Z}^{[t]}) \leftrightarrow (\mathbf{R}, \mathbf{Y}, \mathbf{Y}^{[t]})$ forms a Markov chain. In all the two-party games we consider in this paper, $\mathbf{R}$ is uniform conditioned on the transcripts $(\mathbf{Z}, \mathbf{Z}^{[t]})$ and some of the tampered output. Therefore even if the whole source $\mathbf{X}$ (and $\mathbf{X}^{[t]}$) is sent by Quentin and added to the transcript, $\mathbf{R}$ is still uniform. In other word, the protocol P is *strong* in $\mathbf{X}$. Now observe that when a protocol P is

[6] The proof of this protocol actually first came in [CL16a]. [CS16] suggested this protocol but didn't prove it and used a different protocol instead.

strong in $\mathbf{X}$, we can actually re-design P in the following way: Quentin simply sends $\mathbf{X}$, and Wendy simulates the output of P using $\mathbf{X}, \mathbf{Y}$. When the protocol is re-designed in this way we say $\mathbf{X}$ is the *seed* of the protocol. Similarly we can let Wendy sends $\mathbf{Y}$ and Quentin simulates P if P is strong in $\mathbf{Y}$. It's also not hard to switch the strongness of a protocol using the idea of alternating extraction: if Wendy produces the output $\mathbf{R}$, we can let Wendy send $\mathbf{R}$ and let Quentin output $\mathrm{Ext}(\mathbf{X}, \mathbf{R})$ instead. Now the protocol becomes strong in $\mathbf{Y}$. The advantage of strongness is we can run many different protocols *in parallel*. That is, suppose Wendy holds many correlated source $\mathbf{Y}_1, \ldots, \mathbf{Y}_r$, Quentin holds $\mathbf{X}$ and another source $\mathbf{Q}$ correlated with $\mathbf{X}$, and they want to run many protocols $P_1(\mathbf{Q}, \mathbf{Y}_1), \ldots, P_\ell(\mathbf{Q}, \mathbf{Y}_\ell)$. Then Quentin can simply send $\mathbf{Q}$ (and $\mathbf{Q}^{[t]}$) and let Wendy simulate everything. This ensures that the total *communication complexity* is low, so that the source $\mathbf{X}$ in Quentin's hand only loses roughly $(t+1)|\mathbf{Q}|$ bits of entropy regardless of how many protocols there are. Besides, Wendy doesn't lose any entropy. This idea plays a crucial role in [CL16a] and the followup works.

Another advantage of strongness is, if we let Quentin send his whole source in a look-ahead game, Wendy doesn't need to send any of her output. Therefore all of Wendy's outputs $\mathbf{W}_1, \ldots, \mathbf{W}_\ell$ remain uniform but still have the look-ahead property (i.e. $\mathbf{W}_i$ is uniform conditioned on $\mathbf{W}_1, \ldots, \mathbf{W}_{i-1}$ and their tampering). Therefore these strings can be saved for later use. This is called a look-ahead extractor [DW09]. There's only one drawback: to run any protocol based on alternating extraction, usually the length of $|\mathbf{Q}|$ needs to be proportional to t. Therefore if we need $\mathbf{X}$ to still have some entropy left after sending $\mathbf{Q}$, the total entropy requirement for $\mathbf{X}$ becomes proportional to t^2. Nevertheless t is usually small compared to other parameters so this is not a big deal.

3.6 Correlation breaker based on somewhere independence

Now we are ready to introduce the general strategy for the correlation-breaking game. First Quentin and Wendy run a 2-look-ahead extractor and create two strings $\mathbf{W}_0, \mathbf{W}_1$ on Wendy's side. Now suppose the the advice is $\alpha \in \{0,1\}^a$, and we use α_j to denote the j-th bit of α. For every $j \in [a]$, define $\mathbf{V}_{2j-1} := \mathbf{W}_{\alpha_j}$ and $\mathbf{V}_{2j} := \mathbf{W}_{1-\alpha_j}$. Note that the pair $(\mathbf{V}_{2j-1}, \mathbf{V}_{2j})$ is defined in the "flip-flop" way [Coh16a] so that it will either be $(\mathbf{W}_0, \mathbf{W}_1)$ or $(\mathbf{W}_1, \mathbf{W}_0)$, depending on α_j. If $\alpha_j \neq \alpha_j^i$, in the position $\mathbf{V}_{2j-\alpha_j}$ where $\mathbf{W}_1$ is placed, the corresponding tampered version $\mathbf{V}^1_{2j-\alpha_j}$ should be $\mathbf{W}_0^i$. Therefore we get independence of $\mathbf{W}_1$ from $\mathbf{W}_0^i$ based on the look-ahead property. Now observe that $\mathbf{V}_1, \ldots, \mathbf{V}_{2a}$ is *somewhere-independent* [CS16] from their tampering. That is, for every $i \in [t]$ there exists some $j \in [2a]$ such that $\mathbf{V}_j$ is independent from $\mathbf{V}_j^i$. If they then use the independence merging protocol described above to merge these strings, they get $\mathbf{R}$ which is uniform conditioned on $\mathbf{R}^{[t]}$, and hence win the correlation-breaking game. The entropy requirement is proportional to $O(a)$ instead of 2^a. Moreover, the entropy requirement can be further improved by running the independence merging protocol in parallel. That is, if they merge every two strings in parallel and repeat for $\log(2a)$ rounds, eventually all the strings will be merged into one which collects all the independence. Intuitively the entropy requirement is proportional to $\log(a)$. This is the main idea in [CL16a]. However, as pointed out in [Li17], there were two obstacles in [CL16a] which prevented them from getting $O(\log a)$ dependence. We explain each of them in the next two paragraphs respectively.

3.7 Preparing seeds for sub-protocols

The strategy we described above runs many independence merging protocols in parallel for $\log(a)$ rounds. In the i-th round Quentin needs to prepare a seed $\mathbf{Q}_i$ with enough entropy conditioned on the transcripts. The strategy in [CL16a] is to take a prefix of $\mathbf{X}$ as $\mathbf{Q}_i$ in each round. However, suppose in the first round Quentin sends a prefix $\mathbf{Q}_1$. Then there are t-tampered messages $\mathbf{Q}_1^{[t]}$ sent at the same time. Therefore $\mathbf{X}$ loses about $(t+1)|\mathbf{Q}_1|$ entropy. In the next round, to ensure that the prefix $\mathbf{Q}_2$ of $\mathbf{X}$ still have some entropy, the length of $\mathbf{Q}_2$ needs to be $O(t|\mathbf{Q}_1|)$. Therefore the entropy requirement for $\mathbf{X}$ grows exponentially in the number of rounds. To solve this problem, [Li17] observed that they can first run a ℓ-look-ahead extractor which is strong in $\mathbf{Y}$ to help Quentin prepare $\ell = O(\log a)$ look-ahead strings $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_\ell$. Then in the i-th round Quentin simply sends $\mathbf{Q}_i$.

In our setting of linearly correlated sources (discussed below in Section 3.9), we find it slightly cumbersome to define a look-ahead protocol strong in $\mathbf{Y}$ because of the linear correlation between sources. Therefore, we take a slightly different strategy: we construct NIPM in the way that it can take a uniform seed $\mathbf{Q}_1$ and merge every two *weak* sources $(\mathbf{V}_1, \mathbf{V}_2), (\mathbf{V}_3, \mathbf{V}_4), \ldots, (\mathbf{V}_{2a-1}, \mathbf{V}_{2a})$ into uniform strings $\mathbf{W}_1, \ldots, \mathbf{W}_a$. Then before the start of next round, we take a prefix $\mathbf{P}$ of $\mathbf{W}_1$ and extract $\mathbf{Q}_2 = \mathrm{Ext}(\mathbf{X}, \mathbf{P})$. If $\mathbf{P}$ is short compared to each $\mathbf{W}_i$, then each $\mathbf{W}_i$ should still have enough entropy, and the merging process can continue. Essentially this is like running the look-ahead protocol "on the fly".

3.8 Entropy recycling

Second, when merging a block $(\mathbf{V}_1, \mathbf{V}_2)$ into a single source $\mathbf{W}$ using the alternating extraction protocol described above, the length of $\mathbf{W}$ is only $\beta|\mathbf{V}_1|$ for some constant $\beta < 1$. Therefore the entropy requirement for $\mathbf{Y}$ also grows exponentially in the number of rounds. To solve this problem, [Li19] observed that one can try to "recycle entropy" from $\mathbf{Y}$, which actually contains all the entropy Wendy currently has. Since $\mathbf{Y}$ can contain entropy much larger than $|\mathbf{V}_1|$, it is possible to recover the length of $\mathbf{W}$ to $|\mathbf{V}_1|$. [Li19] did this by using an extra seed from Quentin as a buffer to extract from $\mathbf{Y}$.

However, as we pointed out in the previous paragraph, preparing multiple seeds for Quentin using look-ahead extractor is cumbersome in the linearly correlated source setting. Therefore we choose to embed this approach into each NIPM sub-protocol.[7] That is, we change the protocol $\mathrm{NIPM}(\mathbf{Q}, (\mathbf{V}_1, \mathbf{V}_2))$ to $\mathrm{NIPM}_{rec}(\mathbf{Q}, (\mathbf{V}_1, \mathbf{V}_2, \mathbf{Y}))$ in the following way. First, run the original $\mathrm{NIPM}(\mathbf{Q}, (\mathbf{V}_1, \mathbf{V}_2))$ to get $\mathbf{P}$ which merges the independence of $\mathbf{V}_1$ and $\mathbf{V}_2$. Suppose $\mathbf{P}$ is computed on Wendy's side. Then Wendy sends $\mathbf{P}$ to Quentin, Quentin sends $\mathbf{S} = \mathrm{Ext}(\mathbf{Q}, \mathbf{P})$ to Wendy, and Wendy computes $\mathbf{W} = \mathrm{Ext}(\mathbf{Y}, \mathbf{S})$ as output. Note that this protocol is still strong in $\mathbf{X}$. Moreover, if $\mathbf{P}$ is uniform conditioned on $\mathbf{P}^T$ for some $T \subseteq [t]$, then based on the independence preserving property of strong seeded extractor, $\mathbf{W}$ should still be uniform conditioned on $\mathbf{W}^T$. Therefore if the entropy of $\mathbf{Q}$ can afford one more round of alternating extraction, NIPM_{rec} preserves every property we want for NIPM, but also has output length $|\mathbf{W}| = |\mathbf{V}_1|$.

[7] Another way used in [Li19] to do this is to let Quentin prepare a longer seed and only use a small part of it to run the NIPM. Then this longer seed will still have some leftover entropy after the NIPM is finished, and hence can be used as a buffer for entropy recycling. However this raises the entropy requirement by a $O(t)$ factor, which actually matters in our setting since $t = \omega(1)$.

3.9 Two-party games for linearly correlated sources

Finally we state how to modify the definition of correlation-breaking game to work for $(\mathbf{X} = \mathbf{A}+\mathbf{B}, \mathbf{Y})$ where $(\mathbf{A}, \mathbf{A}^{[t]})$ is independent of $(\mathbf{B}, \mathbf{B}^{[t]}, \mathbf{Y}, \mathbf{Y}^{[t]})$, $\mathbf{A}$ has some entropy and $\mathbf{Y}$ is uniform. In this modified game, Quentin holds $\mathbf{A}$, and Wendy holds $(\mathbf{B}, \mathbf{Y})$. Besides, there are two transcripts (and their t tampered versions): the normal transcript $\mathbf{Z}$, and a "write-only" transcript $\mathbf{Z}_B$. Then Quentin and Wendy run the game as if they are simulating a two-party communication between $\mathbf{X}$ and $\mathbf{Y}$. This works as follows. First, whenever Quentin wants to send or output $\mathbf{Q} = f(\mathbf{X}, \mathbf{Z})$, f must be a linear function for any fixed $\mathbf{Z}$, and Wendy must send $\mathbf{Q}_B = f(\mathbf{B}, \mathbf{Z})$ first. Besides, $\mathbf{Q}_B$ will be added to the write-only transcript $\mathbf{Z}_B$. After receiving $\mathbf{Q}_B$, Quentin sends or outputs $\mathbf{Q} = f(\mathbf{A}, \mathbf{Z}) + \mathbf{Q}_B$, and then $\mathbf{Q}$ is added to the transcript $\mathbf{Z}$. Second, Wendy's message should always be in the form $g(\mathbf{Y}, \mathbf{Z})$ for some deterministic function g, which means Wendy doesn't have access to $\mathbf{B}$ (except when helping Quentin compute $f(\mathbf{X}, \mathbf{Z})$). Then we want the output $\mathbf{R}$ to be uniform conditioned on $(\mathbf{R}^{[t]}, \mathbf{Z}, \mathbf{Z}^{[t]}, \mathbf{Z}_B, \mathbf{Z}_B^{[t]})$. Note that $(\mathbf{A}, \mathbf{A}^{[t]}) \leftrightarrow (\mathbf{Z}, \mathbf{Z}^{[t]}, \mathbf{Z}_B, \mathbf{Z}_B^{[t]}) \leftrightarrow (\mathbf{Y}, \mathbf{Y}^{[t]}, \mathbf{B}, \mathbf{B}^{[t]})$ is always a Markov chain.

Effectively this modified game is similar to a normal game between $\mathbf{A}$ and $\mathbf{Y}$, except for two things. First, f must be linear. In our construction we actually make sure that every function f Quentin uses is simply a strong linear seeded extractor. Second, whenever Quentin wants to send a message to help Wendy run some protocols, Wendy is forced to send $\mathbf{Q}_B$ which leaks some information about her source. This means every source Wendy holds in hand loses some entropy. Nevertheless, we will make sure that every sub-protocol Wendy simulates still works even if she only has weak sources. Finally, another slight difference on Wendy's side is she can only run some deterministic function using $\mathbf{Q} = \mathbf{Q}_A + \mathbf{Q}_B$ instead of $\mathbf{Q}_A$. But this is not a problem, since $\mathbf{Q}_B$ is independent of $\mathbf{Q}_A$ conditioned on the transcripts, which means the conditional entropy of $\mathbf{Q}$ is the same as $\mathbf{Q}_A$. Besides, $\mathbf{Q}$ is still independent from Wendy's side since $\mathbf{Q}_B$ is already in the transcript.

3.10 Summary of our correlation breaker construction

Finally we summarize our construction of a correlation breaker using the two-party game setting in Section 3.9. We use the following building blocks:

- A strong linear seeded extractor LExt, which is the "condense-then-hash" extractor (Lemma 4.9).
- A 2-look-ahead extractor laExt, which takes a weak source $\mathbf{Y}$ and an independent uniform seed $\mathbf{Q}$, then outputs two strings $(\mathbf{R}_0, \mathbf{R}_1)$ such that $\mathbf{R}_1$ is uniform conditioned on $(\mathbf{R}_0, \mathbf{R}_0^{[t]})$ (Lemma 6.3).
- A NIPM which takes two weak sources $\mathbf{V}_1, \mathbf{V}_2$, a entropy pool $\mathbf{Y}$ and an independent uniform seed $\mathbf{Q}$, then output a string $\mathbf{W}$ which merges the independence of $\mathbf{V}_1, \mathbf{V}_2$ from their tampering (Lemma 6.5).

Given a source $\mathbf{X} = \mathbf{A} + \mathbf{B}$, a uniform seed $\mathbf{Y}$, their tampering $\mathbf{A}^{[t]}, \mathbf{B}^{[t]}, \mathbf{Y}^{[t]}$ such that $(\mathbf{A}, \mathbf{A}^{[t]})$ is independent of $(\mathbf{B}, \mathbf{B}^{[t]}, \mathbf{Y}, \mathbf{Y}^{[t]})$, an advice $\alpha \in \{0,1\}^a$ and the tampered advice $\alpha^{[t]} \neq \alpha$, the construction works as follows:

1. Wendy sends $\mathbf{W}_0$ which is a prefix of $\mathbf{Y}$.

2. Wendy sends $\mathbf{Q}_{0B} = \mathrm{LExt}(\mathbf{B}, \mathbf{W}_0)$, then Quentin sends $\mathbf{Q}_0 = \mathrm{LExt}(\mathbf{X}, \mathbf{W}_0) = \mathrm{LExt}(\mathbf{A}, \mathbf{W}_0) + \mathbf{Q}_{0B}$.

3. Wendy computes $(\mathbf{R}_1, \mathbf{R}_0) = \mathrm{laExt}(\mathbf{Y}, \mathbf{Q}_0)$, and get a sequence of $2a$ somewhere-independent strings $\mathbf{V} = (\mathbf{V}_1, \ldots, \mathbf{V}_{2a})$ by assigning each string to be $\mathbf{R}_0$ or $\mathbf{R}_1$ based on α (see the discussion in Section 3.6 or a formal proof in Lemma 6.6).

4. Repeat the following steps for i from 1 to $\log(2a)$:

   I Wendy sends $\mathbf{W}_i$ which is a prefix of $\mathbf{V}_1$.

   II Wendy sends $\mathbf{Q}_{iB} = \mathrm{LExt}(\mathbf{B}, \mathbf{W}_i)$, then Quentin sends $\mathbf{Q}_i = \mathrm{LExt}(\mathbf{A}, \mathbf{W}_i) + \mathbf{Q}_{iB}$

   III Wendy merges each pair $(\mathbf{V}_{2j-1}, \mathbf{V}_{2j})$ into a single string $\mathbf{V}_j$ with the NIPM, using $\mathbf{Q}_i$ as the uniform seed and $\mathbf{Y}$ as the entropy pool. Note that the number of strings in $\mathbf{V}$ decreases by a factor of 2 after this step.

5. Now there is only one string in $\mathbf{V}$. Output $\mathbf{V}$, which is uniform conditioned on $\mathbf{V}^{[t]}$.

Note that the construction above has the following features:

- The only message from Quentin is the uniform seed $\mathbf{Q}_i$ in each round, which is computed by a strong linear seed extractor. The length of $\mathbf{Q}_i$ should be $O(t \log(n))$ for each sub-protocol to work.

- In each round Wendy sends a message $\mathbf{W}_i$ as the seed of Quentin's extraction, and also $\mathbf{Q}_{iB}$ to help Quentin compute $\mathbf{Q}_i$. Both of these messages has length $O(|\mathbf{Q}_i|)$ and cause the sources in Wendy's hand to lose $O(t\,|\mathbf{Q}_i|)$ bits of entropy. However, both the look-ahead extractor and the NIPM still work even if Wendy only has weak sources.

Finally observe that in each of the $O(\log(a))$ rounds, both parties need to send a message of length $O(t \log(n))$. Since in each round there are t tampered messages sent simultaneously, the entropy requirement of each side is $O(t^2 \log(a) \log(n))$. Moreover, the length of each $|\mathbf{V}_j|$ need to be $O(t\,|\mathbf{Q}_i|) = O(t^2 \log(n))$ to tolerate the entropy loss in each round, so there is an extra $O(t^3 \log(n))$ entropy requirement on $\mathbf{Y}$ in the look-ahead extractor. Nevertheless, this is dominated by $O(t^2 \log(a) \log(n))$ in our application.

4 Preliminaries

Notations and Conventions. The logarithm in this paper is always base 2. For every $n \in \mathbb{N}$, define $[n] = \{1, 2, \ldots, n\}$. We sometimes abuse notation and treat distributions and random variables as the same. We always write a random variable/distribution in boldface font. We use $\mathrm{Supp}(\mathbf{X})$ to denote the support of a distribution. We use $\mathbf{U}_n$ to denote the uniform distribution on $\{0,1\}^n$, and when $\mathbf{U}_n$ appears with other random variables in the same joint distribution, then $\mathbf{U}_n$ is considered to be independent of other random variables. Sometimes we simply write the uniform distribution as $\mathbf{U}$ if the length n is not relevant and is clear in the context. When there's a sequence of random variables $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_t$ in the context, we sometimes take a set $S \subseteq [t]$ and use $\mathbf{X}_S$ to denote the sequence of random variables using index in S as subscript, i.e. $\{\mathbf{X}_i\}_{i \in S}$. We also use similar notation for index on superscript. We use $\mathbf{X}_{<i}$ to denote $(\mathbf{X}_1, \ldots, \mathbf{X}_{i-1})$ and $\mathbf{X}_{\le i}$ to denote $(\mathbf{X}_1, \ldots, \mathbf{X}_i)$. If $\mathbf{X}_0$ is defined in the context, then $\mathbf{X}_0$ is also included in $\mathbf{X}_{<i}$ and $\mathbf{X}_{\le i}$.

4.1 Statistical Distance

Definition 4.1. *Let $\mathbf{D}_1, \mathbf{D}_2$ be two distributions on the same universe Ω. The* statistical distance *between $\mathbf{D}_1$ and $\mathbf{D}_2$ to be*

$$\Delta(\mathbf{D}_1; \mathbf{D}_2) := \max_{T \subseteq \Omega} (\mathbf{D}_1(T) - \mathbf{D}_2(T)) = \frac{1}{2} \sum_{s \in \Omega} |\mathbf{D}_1(s) - \mathbf{D}_2(s)|.$$

We say $\mathbf{D}_1$ is ε-close to $\mathbf{D}_2$ if $\Delta(\mathbf{D}_1; \mathbf{D}_2) \leq \varepsilon$, which is also denoted by $\mathbf{D}_1 \approx_\varepsilon \mathbf{D}_2$. Specifically, when there are two joint distributions $(\mathbf{X}, \mathbf{Z})$ and $(\mathbf{Y}, \mathbf{Z})$ such that $(\mathbf{X}, \mathbf{Z}) \approx_\varepsilon (\mathbf{Y}, \mathbf{Z})$, we sometimes write $(\mathbf{X} \approx_\varepsilon \mathbf{Y}) \mid \mathbf{Z}$ for short.

We frequently use the following standard properties.

Lemma 4.2. *For every distribution $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ on the same universe, the following properties hold:*

- *For any distribution $\mathbf{Z}$, $\Delta((\mathbf{D}_1, \mathbf{Z}); (\mathbf{D}_2, \mathbf{Z})) = \mathbb{E}_{z \sim \mathbf{Z}} [\Delta(\mathbf{D}_1|_{\mathbf{Z}=z}; \mathbf{D}_2|_{\mathbf{Z}=z})]$.*
- *For every function f, $\Delta(f(\mathbf{D}_1); f(\mathbf{D}_2)) \leq \Delta(\mathbf{D}_1; \mathbf{D}_2)$.*
- *$\Delta(\mathbf{D}_1; \mathbf{D}_3) \leq \Delta(\mathbf{D}_1; \mathbf{D}_2) + \Delta(\mathbf{D}_2; \mathbf{D}_3)$. (triangle inequality)*

4.2 Extractors and Condensers

Definition 4.3 (min-entropy)**.** *Let $\mathbf{X}$ be a distribution on some finite universe Ω. The* min-entropy *of $\mathbf{X}$ is*

$$\mathrm{H}_\infty(\mathbf{X}) := \min_{x \in \mathrm{Supp}(\mathbf{X})} \left(\log \left(\frac{1}{\Pr[\mathbf{X} = x]} \right) \right).$$

We say a distribution $\mathbf{X}$ over $\{0,1\}^n$ is a (n,k)-*source* if $\mathrm{H}_\infty(\mathbf{X}) \geq k$.

Definition 4.4 (conditional min-entropy)**.** *For joint distribution $(\mathbf{X}, \mathbf{Z})$, the* conditional min-entropy *of $\mathbf{X}$ given $\mathbf{Z}$ is*

$$\mathrm{H}_\infty(\mathbf{X} \mid \mathbf{Z}) := \min_{z \in \mathrm{Supp}(\mathbf{Z})} (\mathrm{H}_\infty(\mathbf{X}|_{\mathbf{Z}=z})).$$

Definition 4.5. *A function $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is called a (k, ε)-*seeded extractor *if for every independent $\mathbf{X}, \mathbf{Y}$ such that $\mathbf{X}$ is a (n,k)-source and $\mathbf{Y} = \mathbf{U}_d$,*

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

We say Ext *is a (k, ε)-strong seeded extractor if*

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

Definition 4.6. *A function $\mathrm{Con} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is called a (k, k', ε)-*strong condenser *if for every independent $\mathbf{X}, \mathbf{Y}$ such that $\mathbf{X}$ is a (n,k)-source and $\mathbf{Y} = \mathbf{U}_d$, there exists $\mathbf{Z}$ such that*

$$(\mathrm{Con}(\mathbf{X}, \mathbf{Y}) \approx_\varepsilon \mathbf{Z}) \mid \mathbf{Y}$$

and $\mathrm{H}_\infty(\mathbf{Z} \mid \mathbf{Y}) \geq k'$. If $k' = k$ then we say Con *is a (k, ε)-*strong lossless condenser.

We say Ext is linear if for every $s \in \{0,1\}^d$, $\mathrm{Ext}(\cdot, s)$ is a linear function over $\mathbb{F}_2$. Similarly we say Con is linear if for every $s \in \{0,1\}^d$, $\mathrm{Con}(\cdot, s)$ is a linear function over $\mathbb{F}_2$. In this paper, we use the strong lossless condenser in [GUV09], which can be made linear as observed in [Che10].

Lemma 4.7 ([GUV09],[Che10]). *For every $n \in \mathbb{N}$, $k \le n$, $\varepsilon > 0$ and $\alpha > 0$, there is an explicit (k,ε)-strong linear lossless condenser* $\mathrm{LCon} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *such that* $d = (1 + \frac{1}{\alpha}) \log(nk/\varepsilon) + O(1)$ *and* $m = (1+\alpha)k + d$.

We also use the strong seeded extractor based on Leftover Hash Lemma, which can also be made linear:

Lemma 4.8 ([HILL99]). *For every $n \in \mathbb{N}$, $k \le n$, $\varepsilon > 0$, there is an explicit (k,ε)-strong linear seeded extractor* $\mathrm{LHL} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *such that $d = n$ and $m = k - 2\log(1/\varepsilon)$.*

By the above lemmas we can construct a simple strong linear seeded extractor which has $O(\log(n/\varepsilon))$ seed length if the output length is also $O(\log(n/\varepsilon))$.

Lemma 4.9 (condense-then-hash extractor). *For every $n \in \mathbb{N}$, $k \le n$, $\varepsilon > 0$, there is an explicit (k,ε)-strong linear seeded extractor* $\mathrm{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *such that $k = m + 2\log(1/\varepsilon)$ and $d = 2m + 8\log(n/\varepsilon) + O(1)$.*

Proof. Define $\mathrm{LExt}(x,(s_1,s_2)) := \mathrm{LHL}(\mathrm{LCon}(x,s_1),s_2)$, where $\mathrm{LHL} : \{0,1\}^{n'} \times \{0,1\}^{d_2} \to \{0,1\}^m$ is the $(k,\varepsilon/2)$-strong linear seeded extractor from Lemma 4.8 and $\mathrm{LCon} : \{0,1\}^n \times \{0,1\}^{d_1} \to \{0,1\}^{n'}$ is the $(k,\varepsilon/2)$-strong linear lossless condenser from Lemma 4.7 by taking $\alpha = 1$. Observe that for every (n,k)-source $\mathbf{X}$ and independent seed $\mathbf{Y} = \mathbf{U}_{d_1}$, there exists $\mathbf{Z}$ such that $(\mathrm{LCon}(\mathbf{X},\mathbf{Y}),\mathbf{Y}) \approx_{\varepsilon/2} (\mathbf{Z},\mathbf{Y})$ and $\mathrm{H}_\infty(\mathbf{Z} \mid \mathbf{Y}) \ge k$. Therefore

$$(\mathrm{LExt}(\mathbf{X},(\mathbf{Y},\mathbf{U}_{d_2})),\mathbf{Y},\mathbf{U}_{d_2}) \approx_{\varepsilon/2} (\mathrm{LHL}(\mathbf{Z},\mathbf{U}_{d_2}),\mathbf{Y},\mathbf{U}_{d_2}) \approx_{\varepsilon/2} (\mathbf{U}_m,\mathbf{Y},\mathbf{U}_{d_2}).$$

Observe that LExt is linear. Therefore LExt is a (k,ε)-strong linear seeded extractor. Finally we calculate the seed length. Note that $d_1 = 2\log(nk/\varepsilon) + O(1)$ and $d_2 = n' = 2k + d_1$. Therefore

$$d = d_1 + d_2 = 2(k + d_1) \le 2(m + 2\log(1/\varepsilon) + 2\log(n^2/\varepsilon) + O(1)) \le 2m + 8\log(n/\varepsilon) + O(1).$$

□

Finally we also need the GUV extractor (which is non-linear):

Lemma 4.10 ([GUV09]). *There exists constants $\beta, c_{GUV} > 0$ such that for every $n \in \mathbb{N}$, $k \le n$, $\varepsilon > 2^{-\beta k}$, there is an explicit (k,ε)-strong seeded extractor* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *such that $d = c_{GUV}\log(n/\varepsilon)$ and $m = k/2$.*

4.3 Average Conditional Min-entropy

We also use the following relaxed notion of conditional min-entropy which is sometimes more convenient.

Definition 4.11 ([DORS08]). *For joint distribution $(\mathbf{X},\mathbf{Z})$, the* average conditional min-entropy *of $\mathbf{X}$ given $\mathbf{Z}$ is*

$$\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{Z}) := -\log\left(\mathop{\mathbb{E}}_{z \sim \mathbf{Z}}\left[\max_x(\Pr[\mathbf{X} = x \mid \mathbf{Z} = z])\right]\right).$$

Lemma 4.12 ([DORS08])**.** *Let* $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ *be random variables. Then*

$$\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid (\mathbf{Y}, \mathbf{Z})) \geq \widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{Z}) - \log(\mathrm{Supp}(\mathbf{Y})).$$

Lemma 4.13 ([DORS08])**.** *Let* $\mathbf{X}, \mathbf{Z}$ *be random variables. For every* $\varepsilon > 0$*,*

$$\Pr_{z \sim \mathbf{Z}}\left[\mathrm{H}_\infty(\mathbf{X}|_{\mathbf{Z}=z}) \geq \widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{Z}) - \log(1/\varepsilon)\right] \geq 1 - \varepsilon.$$

Lemma 4.14 ([DORS08])**.** *Let* $\varepsilon, \delta > 0$ *and* $\mathbf{X}, \mathbf{Z}$ *be a random variables such that* $\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{Z}) \geq k + \log(1/\delta)$*. Let* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a* (k, ε)*-seeded extractor. Then*

$$(\mathrm{Ext}(\mathbf{X}, \mathbf{U}_d) \approx_{\varepsilon+\delta} \mathbf{U}_m) \mid \mathbf{Z}.$$

4.4 Markov Chain

Definition 4.15. *Let* $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ *be random variables. We say* $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ *forms a Markov chain if* $\mathbf{X}$ *and* $\mathbf{Y}$ *are independent conditioned on any fixing of* $\mathbf{Z}$*.*

Note that $\mathbf{X}$ and $\mathbf{Y}$ are symmetric in the above definition. Therefore all the properties stated below should also hold for its symmetric variant. Besides, all the properties stated below hold for the degenerate case that $\mathbf{X}$ is empty.

In this paper we are not considering Markov chain as a sequence of events. Instead, we just borrow the definition to model the setting of *two-party communication*. That is, suppose there are two parties holding independent random variables $\mathbf{X}$ and $\mathbf{Y}$ respectively. Each time one of the party sends a new message based on their own random variable and all the messages sent before. If one consider $\mathbf{Z}$ as all the messages sent (i.e. the *transcript*), then $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ is actually a Markov chain. This can be formalized by the following lemma, for which we omit the proof since it's straightforward by definition.

Lemma 4.16. *If* $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ *is a Markov chain, then for every deterministic function* f*, let* $\mathbf{W} = f(\mathbf{X}, \mathbf{Z})$*. Then*

- $(\mathbf{X}, \mathbf{W}) \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ *is a Markov chain.*
- $\mathbf{X} \leftrightarrow (\mathbf{W}, \mathbf{Z}) \leftrightarrow \mathbf{Y}$ *is a Markov chain.*

Sometimes we use "$\mathbf{W}$ *is a deterministic function of* $\mathbf{X}$ *(conditioned on* $\mathbf{Z}$*)" to refer to the first item, and "fix* $\mathbf{W}$*" to refer to the second item.*

The following two lemmas are straightforward and will usually be used implicitly.

Lemma 4.17. *If* $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ *is a Markov chain and* $(\mathbf{Y} \approx_\varepsilon \mathbf{U}) \mid \mathbf{Z}$*, then* $(\mathbf{Y} \approx_\varepsilon \mathbf{U}) \mid (\mathbf{Z}, \mathbf{X})$*.*

Lemma 4.18. *If* $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ *is a Markov chain, then* $\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{Z}) = \widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid (\mathbf{Z}, \mathbf{Y}))$*.*

The following lemma is a direct corollary of [Li15, Lemma 3.20].

Lemma 4.19. *Let* $(\mathbf{X}, \mathbf{W}) \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ *be a Markov chain. Suppose there exists a joint distribution* $(\widetilde{\mathbf{W}}, \mathbf{Z})$ *such that* $(\widetilde{\mathbf{W}} \approx_\varepsilon \mathbf{W}) \mid \mathbf{Z}$*. Then there exists a joint distribution* $(\mathbf{X}, \widetilde{\mathbf{W}}, \mathbf{Z}, \mathbf{Y})$ *such that*

$$(\widetilde{\mathbf{W}} \approx_\varepsilon \mathbf{W}) \mid (\mathbf{Z}, \mathbf{X}, \mathbf{Y})$$

and $(\mathbf{X}, \widetilde{\mathbf{W}}) \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ *forms a Markov chain.*

The following lemma is by embedding Lemma 4.17 into [CS16, Lemma 3.11].

Lemma 4.20. *Let* $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow (\mathbf{Y}, \mathbf{W})$ *be a Markov chain such that* $(\mathbf{Y} \approx_\delta \mathbf{U}_d) \mid \mathbf{Z}$ *and* $\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{Z}) \geq k + \log(1/\varepsilon)$*. Let* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a* (k, ε)*-strong seeded extractor. Then*

$$(\mathrm{Ext}(\mathbf{X}, \mathbf{Y}) \approx_{2\varepsilon+\delta} \mathbf{U}_m) \mid (\mathbf{Z}, \mathbf{Y}, \mathbf{W}).$$

4.5 Affine Sources

The following lemma from [Rao09, Li11] shows the implicit independence between an affine source $\mathbf{X}$ and any of its linear function $L(\mathbf{X})$.

Lemma 4.21 ([Rao09, Li11])**.** *Let* $\mathbf{X}$ *be any random variable sampled from a* (n, k)*-affine source, and let* $L : \{0,1\}^n \to \{0,1\}^m$ *be any linear function. Then there exist independent random variables* $\mathbf{A}, \mathbf{B}$ *such that*

- $\mathbf{X} = \mathbf{A} + \mathbf{B}$.
- *There is a bijection between* $\mathbf{B}$ *and* $\mathbf{L}(\mathbf{X})$
- $\mathrm{H}_\infty(\mathbf{A}) \geq k - m$

We also use the following lemma from [Rao09].

Lemma 4.22 ([Rao09])**.** *Let* $\mathrm{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a* (k, ε)*-strong linear condenser. Then for every* (n, k)*-affine source* $\mathbf{X}$*,*

$$\Pr_{s \sim \mathbf{U}_d}[\mathrm{LExt}(\mathbf{X}, s) \text{ is uniform}] \geq 1 - 2\varepsilon.$$

4.6 Dispersers

Definition 4.23. *We say a function* $\Gamma : [N] \times [D] \to [M]$ *is a* (K, ε)-disperser *if for every set* $X \subseteq [N]$ *with* $|X| \geq K$*, the set* $\Gamma(X) := \{\Gamma(x, y) \mid x \in X, y \in [D]\}$ *satisfies*

$$|\Gamma(X)| \geq \varepsilon M.$$

We use the following disperser from [Zuc07].

Lemma 4.24 ([Zuc07])**.** *For every constant* $\delta > 0$ *and* $\varepsilon = \varepsilon(n) > 0$*, there exists an efficient family of* $(K = N^\delta, \varepsilon)$*-disperser* $\Gamma : [N = 2^n] \times [D] \to [M]$ *such that* $D = O(\frac{n}{\log(1/\varepsilon)})$ *and* $M = \sqrt{K}$*.*

4.7 Non-oblivious bit-fixing sources

Definition 4.25. *A distribution* $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ *on* $\{0,1\}^n$ *is called* (t, γ)*-wise independent if for every subset* $S \subseteq [n]$ *of size* t*,*

$$\mathbf{X}_S \approx_\gamma \mathbf{U}_q.$$

If $\gamma = 0$ *we simply say* $\mathbf{X}$ *is* t*-wise independent.*

Lemma 4.26 ([AGM03])**.** *If* $\mathbf{X}$ *on* $\{0,1\}^n$ *is* (t, γ)*-wise independent, then* $\mathbf{X}$ *is* $n^t\gamma$*-close to a* t*-wise independent distribution.*

Definition 4.27. *A distribution* $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ *on* $\{0,1\}^n$ *is called a* (q, t, γ)*-non-oblivious bit-fixing (NOBF) source if there exists a set* Q *s.t.* $|Q| \leq t$ *and* $\mathbf{X}_{[n]\setminus Q}$ *is* (t, γ)*-wise independent.*

In [Vio14], Viola proved that the majority function is an extractor for $(q, t, 0)$-NOBF source. Combining the result with Lemma 4.26 gives the following lemma.

Lemma 4.28. *Let* $\mathrm{Maj} : \{0,1\}^n \to \{0,1\}$ *be the majority function such that* $\mathrm{Maj}(x) = 1$ *iff* $\sum_i x_i \geq \lceil n/2 \rceil$*. Then there exists a constant* $C_{4.28}$ *such that for every* (q, t, γ)*-NOBF source* $\mathbf{X} \in \{0,1\}^n$*,*

$$\Delta\left(\mathrm{Maj}(\mathbf{X}); \mathbf{U}_1\right) \leq C_{4.28}\left(\frac{\log t}{\sqrt{t}} + \frac{q}{\sqrt{n}}\right) + n^t\gamma.$$

5 Strong Linear Somewhere Random Extractor

In this section, we show how to construct a strong linear seeded somewhere random extractor with seed length $d = O(\log n)$ and error $\varepsilon = 1/\mathrm{poly}(n)$, with entropy requirement linear in the output length $m = c(n)\log n$. This object takes a seed in $[D = 2^d]$ and produce a "somewhere random" source. That is, given an affine source $\mathbf{X}$ and a seed $\mathbf{S}$, the object produce A different outputs $\mathrm{LSRExt}(\mathbf{X}, \mathbf{S}, 1), \ldots, \mathrm{LSRExt}(\mathbf{X}, \mathbf{S}, A)$ such that with high probability over $\mathbf{S}$ one of the output should be uniform. The formal definition is in the following theorem. Note that the definition below is equivalent to saying the error ε is $D^{-(1-\delta)}$. We choose this formulation because it is more well suited for for our application.

Theorem 5.1. *There exists a constant* $\beta_{5.1}$ *which satisfies the following. For every constant* $\delta_{5.1} > 0$*, there exists a constant* $C_{5.1}$ *such that every* $n \in \mathbb{N}$ *and every* $c = c(n) < 2^{\sqrt[3]{\log n}}$*, there an explicit function* $\mathrm{LSRExt} : \{0,1\}^n \times [D] \times [A] \to \{0,1\}^{c\log n}$ *such that*

- $D \leq n^{C_{5.1}}$
- $A \leq C_{5.1}\log^2(c(n))$
- *For every fixed* $s \in [D], z \in [A]$*, the function* $\mathrm{LSRExt}_{s,z}(x) := \mathrm{LSRExt}(x, s, z)$ *is linear.*
- *For every* $(n, \beta_{5.1}c\log(n))$*-affine source* X*, there exists a subset* $B \subseteq [D]$ *of size at most* $D^{\delta_{5.1}}$ *such that for every* $s \in [D]\setminus B$*,* $\exists z \in [A]$ *s.t.* $\mathrm{LSRExt}(X, s, z)$ *is uniform.*

To prove this theorem, we first construct a strong linear seeded extractor with $O(\log n)$ seed length and $O(m)$ entropy requirement, but the error will be slightly larger than $1/\mathrm{poly}(n)$. This extractor relies on "block-source condensing" using strong condenser, which is similar to the standard block-source extraction trick. We first prove a block-source condensing lemma in Section 5.1, and then show how to construct the extractor in Section 5.2. Finally we show how to adapt an error reduction technique from [BDT17] to our setting, and obtain the desired linear somewhere random extractor for affine source with error $1/\mathrm{poly}(n)$. This is presented in Section 5.3.

5.1 Block-source condensing

First we define a block source.

Definition 5.2 ([CG88])**.** $(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_t) \in (\{0,1\}^n)^t$ *is called a* (t, n, k)*-block source if for every* $i \in [t]$ *,* $\mathrm{H}_\infty(\mathbf{X}_i \mid \mathbf{X}_1, \ldots, \mathbf{X}_{i-1}) \geq k$*.*

It is well-known that a high-entropy source is close to a block source.

Lemma 5.3 ([GW97])**.** *Let* $\mathbf{X}$ *be a* $(2n, 2n-\Delta)$*-source. Divide* $\mathbf{X}$ *into two blocks* $\mathbf{X} = \mathbf{X}_1 \circ \mathbf{X}_2$*, each having* n *bits. Then* $(\mathbf{X}_1, \mathbf{X}_2)$ *is* ε*-close to a* $(2, n, n-\Delta-\log(1/\varepsilon))$*-block source.*

By induction we have the following corollary.

Corollary 5.4. *Let* $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$ *be a* $(t, 2n, 2n-\Delta)$*-source. Define* $\mathrm{Split}(\mathbf{X}) = (\mathbf{Y}_1, \ldots, \mathbf{Y}_{2t})$ *such that for every* $i \in [t]$*,* $\mathbf{X}_i = \mathbf{Y}_{2i-1} \circ \mathbf{Y}_{2i}$ *and* $\mathbf{Y}_{2i-1}, \mathbf{Y}_{2i}$ *have* n *bits each. Then* $\mathrm{Split}(\mathbf{X})$ *is* $t\varepsilon$*-close to a* $(2t, n, n-\Delta-\log(1/\varepsilon))$*-block source.*

Next we prove the block-source condensing lemma.

Lemma 5.5 (block-source condensing)**.** *Let* $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_t)$ *be a* (t, n, k)*-source and* $\mathrm{Con} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a* (k, k', ε)*-strong condenser. Let* $\mathbf{Y} = \mathbf{U}_d$*. Define*

$$\mathrm{BlockCon}(\mathbf{X}, \mathbf{Y}) := (\mathrm{Con}(\mathbf{X}_1, \mathbf{Y}), \mathrm{Con}(\mathbf{X}_2, \mathbf{Y}), \ldots, \mathrm{Con}(\mathbf{X}_t, \mathbf{Y})).$$

Then $(\mathbf{Y}, \mathrm{BlockCon}(\mathbf{X}, \mathbf{Y}))$ *is* $t\varepsilon$*-close to a distribution* $(\mathbf{Y}, \mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_t)$ *such that* $(\mathbf{Z}_1, \ldots, \mathbf{Z}_t)$ *is a* (t, m, k')*-block source conditioned on any fixing of* $\mathbf{Y}$*.*

Proof. Let $\overline{\mathbf{X}_i} = (\mathbf{X}_1, \ldots, \mathbf{X}_i)$. We prove by induction that for every $0 \le i \le t$, there exists $(\mathbf{Z}_{i+1}, \ldots, \mathbf{Z}_t)$ such that

$$(\overline{\mathbf{X}_i}, \mathbf{Y}, \mathrm{Con}(\mathbf{X}_{i+1}, \mathbf{U}_d), \ldots, \mathrm{Con}(\mathbf{X}_t, \mathbf{U}_d)) \approx_{(t-i)\varepsilon} (\overline{\mathbf{X}_i}, \mathbf{Y}, \mathbf{Z}_{i+1}, \ldots, \mathbf{Z}_t) \tag{1}$$

and $(\mathbf{Z}_{i+1}, \ldots, \mathbf{Z}_t)$ is a $(t-i, m, k')$-block source conditioned on any fixing of $\overline{\mathbf{X}_i}, \mathbf{Y}$. Note that the $i = 0$ case is what we want to prove in this lemma.

The base case $i = t$ is trivial. Now assume there exists $(\mathbf{Z}_{i+1}, \ldots, \mathbf{Z}_t)$ which satisfies the induction hypothesis. We show that there exists $\mathbf{Z}_i$ such that

$$(\overline{\mathbf{X}_{i-1}}, \mathbf{Y}, \mathrm{Con}(\mathbf{X}_i, \mathbf{Y}), \ldots, \mathrm{Con}(\mathbf{X}_t, \mathbf{Y})) \approx_{(t-i+1)\varepsilon} (\overline{\mathbf{X}_{i-1}}, \mathbf{Y}, \mathbf{Z}_i, \ldots, \mathbf{Z}_t) \tag{2}$$

and $(\mathbf{Z}_i, \mathbf{Z}_{i+1}, \ldots, \mathbf{Z}_t)$ is a $(t-i+1, m, k')$-block source conditioned on any fixing of $\overline{\mathbf{X}_{i-1}}, \mathbf{Y}$. First note that $\mathbf{X}_i$ has min-entropy at least k conditioned on any fixing of $\overline{\mathbf{X}_{i-1}}$. By the definition of strong condenser and by Lemma 4.19, there exists $\mathbf{Z}_i$ such that

$$(\overline{\mathbf{X}_{i-1}}, \mathbf{X}_i, \mathbf{Y}, \mathrm{Con}(\mathbf{X}_i, \mathbf{Y}), \mathbf{Z}_{i+1}, \ldots, \mathbf{Z}_t) \approx_\varepsilon (\overline{\mathbf{X}_{i-1}}, \mathbf{X}_i, \mathbf{Y}, \mathbf{Z}_i, \mathbf{Z}_{i+1}, \ldots, \mathbf{Z}_t), \tag{3}$$

$\mathrm{H}_\infty(\mathbf{Z}_i \mid \overline{\mathbf{X}_{i-1}}, \mathbf{Y}) \ge k'$ and $\mathbf{Z}_i \leftrightarrow (\overline{\mathbf{X}_i}, \mathbf{Y}) \leftrightarrow (\mathbf{Z}_{i+1}, \ldots, \mathbf{Z}_t)$ is a Markov chain. Since $(\mathbf{Z}_{i+1}, \ldots, \mathbf{Z}_t)$ is a $(t-i, m, k')$-block source conditioned on any fixing of $(\overline{\mathbf{X}_i}, \mathbf{Y})$, it's still a $(t-i, m, k')$-block source after further fixing $\mathbf{Z}_i$. Therefore $(\mathbf{Z}_i, \ldots, \mathbf{Z}_t)$ is a $(t-i+1, m, k')$-block source conditioned on any fixing of $(\overline{\mathbf{X}_{i-1}}, \mathbf{Y})$. By adding $\mathrm{Con}(\mathbf{X}_i, \mathbf{Y})$ to (1), applying triangle inequality with (3) and then removing $\mathbf{X}_i$, we get (2). □

5.2 Strong linear seeded extractor with slightly sub-optimal error

As described in Section 2, we first construct a strong linear seeded extractor with error ε and output length $c \log n$ using seed length $O(\log n + \log^2(c) \log(1/\varepsilon))$.

Theorem 5.6. *For every $c = c(n) < 2^{\sqrt[3]{\log n}}$ and every $\varepsilon = \varepsilon(n) > 0$, there exists a family of explicit (k, ε)-strong linear seeded extractor* $\mathrm{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{c \log n}$ *such that* $k = O(c \log(n) + c \log(c) \log(1/\varepsilon))$ *and* $d = O(\log(n) + \log^2(c) \log(1/\varepsilon))$.

Proof. Let $h = \log(c)$ and $\varepsilon_0 = \frac{\varepsilon}{4c}$. Let $k_0, k_1, \ldots, k_h, m_0, \ldots, m_h, d_0, \ldots, d_h, d_{ext} \in \mathbb{N}$ be some parameters to be defined later. Given a (n, k_0)-source $\mathbf{X}$, first we take some uniform seeds $(\mathbf{Y}_0, \mathbf{Y}_1, \ldots, \mathbf{Y}_h)$, each having length $d_0, d_1, \ldots, d_h$ respectively. Besides, we write $\overline{\mathbf{Y}_i} = (\mathbf{Y}_0, \mathbf{Y}_1, \ldots, \mathbf{Y}_i)$ for short. Our first step is to iteratively construct $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_h$ such that for every $0 \le i \le h$,

$$(\overline{\mathbf{Y}_i}, \mathbf{X}_i) \approx_{2^{i+1}\varepsilon_0} (\overline{\mathbf{Y}_i}, \mathbf{Z}_i) \tag{4}$$

where $\mathbf{Z}_i$ is a $(2^i, m_i, k_i)$-block source conditioned on any fixing of $\overline{\mathbf{Y}_i}$. To do this, first we define

$$\mathbf{X}_0 := \mathrm{LCon}_0(\mathbf{X}, \mathbf{Y}_0),$$

where $\mathrm{LCon}_0 : \{0,1\}^n \times \{0,1\}^{d_0} \to \{0,1\}^{m_0}$ is a (k_0, ε_0)-strong linear lossless condenser. Then the $i = 0$ case of (4) follows by definition of strong condenser. Next, for every $i \in [h]$, we define

$$\mathbf{X}_i := \mathrm{Split}(\mathrm{BlockCon}_i(\mathbf{X}_{i-1}, \mathbf{Y}_i)),$$

where Split is from Corollary 5.4 and $\mathrm{BlockCon}_i$ is the block-source condenser from Lemma 5.5 instantiated with a (k_i, ε_0)-strong linear lossless condenser $\mathrm{LCon}_i : \{0,1\}^{m_{i-1}} \times \{0,1\}^{d_i} \to \{0,1\}^{2m_i}$. Observe that conditioned on any fixing of $\overline{\mathbf{Y}_{i-1}}$, by Lemma 5.5 there exists $\mathbf{Z}'_i$ such that

$$(\mathbf{Z}'_i, \mathbf{Y}_i) \approx_{(2^{i-1})\varepsilon_0} (\mathrm{BlockCon}_i(\mathbf{Z}_{i-1}, \mathbf{Y}_i), \mathbf{Y}_i)$$

which is a $(2^{i-1}, 2m_i, k_{i-1})$-block source. By Corollary 5.4, if $k_{i-1} \ge m_i + k_i + \log(1/\varepsilon_0)$, then conditioned on any fixing of $\overline{\mathbf{Y}_i}$ there exists a $(2^i, m_i, k_i)$-block source $\mathbf{Z}_i$ such that

$$\begin{aligned}
(\overline{\mathbf{Y}_{i-1}}, \mathbf{Y}_i, \mathbf{X}_i) &\approx_{2^i\varepsilon_0} (\overline{\mathbf{Y}_{i-1}}, \mathbf{Y}_i, \mathrm{Split}(\mathrm{BlockCon}_i(\mathbf{Z}_{i-1}, \mathbf{Y}_i))) \\
&\approx_{2^{i-1}\varepsilon_0} (\overline{\mathbf{Y}_{i-1}}, \mathbf{Y}_i, \mathrm{Split}_i(\mathbf{Z}'_i)) \\
&\approx_{2^{i-1}\varepsilon_0} (\overline{\mathbf{Y}_{i-1}}, \mathbf{Y}_i, \mathbf{Z}_i).
\end{aligned}$$

Then (4) holds by triangle inequality. In the last step, we take one more uniform seed $\mathbf{Y}_{ext} \in \{0,1\}^{d_{ext}}$, and output

$$\mathrm{LExt}(\mathbf{X}, (\overline{\mathbf{Y}_h}, \mathbf{Y}_{ext})) := \mathrm{BlockExt}(\mathbf{X}_h, \mathbf{Y}_{ext})$$

where BlockExt is the block-source condenser from Lemma 5.5 instantiated with a (k_h, ε_0)-strong linear seeded extractor $\mathrm{LHL} : \{0,1\}^{m_h} \times \{0,1\}^{d_{ext}} \to \{0,1\}^{\log n}$ (note that an extractor is a special case of condenser). By (4) and Lemma 5.5 we can conclude that LExt is a (k, ε)-strong linear extractor with seed length $d = \sum_{i=0}^{h} d_i + d_{ext}$. Moreover, observe that LExt is linear since all the operations in this construction are linear if $\mathbf{Y}_0, \ldots, \mathbf{Y}_h, \mathbf{Y}_{ext}$ are fixed.

Finally we need to set the undefined parameters and show that there exist explicit constructions of $\mathrm{LCon}_0, \ldots, \mathrm{LCon}_h$ and LHL. First we set $k_h = \log(n) + 2\log(1/\varepsilon_0)$, so that LHL exists by Lemma 4.8. Next, for every $1 \le i \le h$, let LCon_i be the lossless condenser from Lemma 4.7 by setting $\alpha = \frac{1}{h+1}$. Then $2m_i = (1 + \frac{1}{h+1})k_{i-1} + d_i$ and $d_i = (h+2)\log(m_{i-1}k_{i-1}/\varepsilon_0) + O(1)$. Therefore we can rewrite the condition $k_{i-1} \ge m_i + k_i + \log(1/\varepsilon_0)$ as

$$k_{i-1} \ge 2\left(1 + \frac{1}{h}\right)\left(k_i + \frac{d_i}{2} + \log(1/\varepsilon_0)\right).$$

Set $k_{h-1}, \ldots, k_0$ based on this recurrence, and use the fact that $d_1 \geq \cdots \geq d_h$, we get that for every $0 \leq i < h$,

$$k_i \leq \left(2\left(1+\frac{1}{h}\right)\right)^{h-i}(\log(n) + d_1 + 4\log(1/\varepsilon_0)).$$

Since $(1+\frac{1}{h})^h < e$, the entropy requirement of LExt is $k_0 = O(c\log(n)+c\log(c)\log(1/\varepsilon))$, assuming that $O(h\log(m_0 \cdot k_0 \cdot c)) = O(\log n+\log(c)\log(1/\varepsilon))$, which is true based on our parameter restriction. Finally, for LCon_0, we take the condenser from Lemma 4.7 by setting $\alpha = 1$. Then we get $d_0 = O(\log(n/\varepsilon))$ and $m_0 = 2k_0 + d_0$. Therefore $d_1 = O(h\log(k_0/\varepsilon))$. The total seed length of this extractor is

$$d = \sum_{i=0}^{h} d_i + d_{ext} \leq d_0 + hd_1 + m_h = O(\log(n) + h^2\log(1/\varepsilon) + h^3) = O(\log(n) + \log^2(c)\log(1/\varepsilon)).$$

□

5.3 Error reduction

We now adapt the error reduction scheme in [BDT17] to our setting, and argue that the error reduction preserves linearity. In fact, there's a bonus in our setting: the general scheme in [BDT17] only gives a somewhere random condenser, but if we start with a strong linear seeded extractor and consider only affine source, we actually get a somewhere random extractor instead of a somewhere random condenser. The formal statement is as follows.

Lemma 5.7. *Suppose there exists an explicit (k, ε)-strong linear seeded extractor $\mathrm{LExt} : \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^m$ where $\varepsilon < 1/3$. Then for every constant $\delta > 0$ there exists an explicit function $\mathrm{LSRExt} : \{0,1\}^n \times [D] \times [A] \to \{0,1\}^m$ which satisfies the following:*

- $D = 2^{O(d')}$
- $A = O(\frac{d'}{\log(1/\varepsilon)})$
- *For every $s \in [D], z \in [A]$, the function $\mathrm{LSRExt}_{s,z}(x) := \mathrm{LSRExt}(x, s, z)$ is linear.*
- *For every (n,k)-affine source $\mathbf{X}$, there exists a set $B \subseteq [D]$ of size at most D^δ such that for every $s \in [D]\backslash B$, there exists $z \in [A]$ s.t. $\mathrm{LSRExt}(\mathbf{X}, s, z)$ is uniform.*

Proof. Let $\Gamma : [D] \times [A] \to [D' = 2^{d'}]$ be the $(D^\delta, 3\varepsilon)$-disperser from Lemma 4.24. Note that $D = 2^{2d'/\delta} = 2^{O(d')}$ and $A = O(\frac{\log D}{\log(1/\varepsilon)}) = O(\frac{d'}{\log(1/\varepsilon)})$. Define

$$\mathrm{LSRExt}(x, s, z) := \mathrm{LExt}(x, \Gamma(s, z)).$$

Observe that $\mathrm{LSRExt}_{s,z}$ is linear for every s, z since LExt is linear. It remains to prove the last property. Let B be the set which consists of every s s.t. $\forall z \in [A]$, $\mathrm{LExt}(\mathbf{X}, \Gamma(s, z))$ is not uniform. By Lemma 4.22, the number of seed $y \in [D']$ such that $\mathrm{LExt}(\mathbf{X}, \Gamma(s, z))$ is not uniform is at most $2\varepsilon D'$. Therefore $|\Gamma(B)| \leq 2\varepsilon D' < 3\varepsilon D'$, which implies that $|B| < D^\delta$. □

Now we are ready to prove Theorem 5.1.

Proof of Theorem 5.1. Let $\text{LExt} : \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{c \log n}$ be the explicit (k,ε)-strong linear seeded extractor from Theorem 5.6 such that $\varepsilon = n^{-\frac{1}{\log^2 c}}$. Then $k = O(c \log n)$ and $d' = O(\log n)$. Now we use Lemma 5.7 to construct the function $\text{LSRExt} : \{0,1\}^n \times [D] \times [A] \to \{0,1\}^{c \log n}$ based on LExt. Observe that $D = 2^{O(d')} = n^{O(1)}$ and $A = O(\frac{d'}{\log(1/\varepsilon)}) = O(\log^2 c)$. The last two properties follow from Lemma 5.7 directly. □

6 Correlation Breaker for Linearly Correlated Source

Informally speaking, a correlation breaker takes a source and an advice, where the advice is guaranteed to be different from its tampered version, and "break the correlation" between the source and its tampered versions. That is, it outputs a uniform random string which is independent of the tampered output. Typically the source consists of two independent sources, but in our setting we will take two linearly correlated sources with "implicit independence".

In this section, we construct a correlation breaker with advice for such linearly correlated source. This primitive first appeared in [Li16] and was explicitly defined in [CL16b]. The entropy requirement of the correlation breaker in these previous works was $\log^C(n)$, for some large enough constant C. In this section, we take inspiration from recent constructions of correlation breaker for independent sources, and improve the entropy requirement of correlation breakers in the linearly correlated setting to $O(\log^{1+o(1)}(n))$. We now state our result more formally.

Definition 6.1. *We say a function* $\text{ACB} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^a \to \{0,1\}^m$ *is a* (t,k,a,ε)*-advice correlation breaker for linearly correlated sources if the following holds. Let*

- $\mathbf{A}, \mathbf{A}^{[t]}, \mathbf{B}, \mathbf{B}^{[t]}$ *be random variables on* $\{0,1\}^n$ *and* $\mathbf{Y}, \mathbf{Y}^{[t]}$ *be random variables on* $\{0,1\}^d$ *such that* $(\mathbf{A}, \mathbf{A}^{[t]})$ *is independent of* $(\mathbf{B}, \mathbf{B}^{[t]}, \mathbf{Y}, \mathbf{Y}^{[t]})$*. Moreover,* $\mathrm{H}_\infty(\mathbf{A}) \geq k$ *and* $\mathbf{Y} = \mathbf{U}_d$*.*
- $\mathbf{X} = \mathbf{A} + \mathbf{B}, \mathbf{X}^i = \mathbf{A}^i + \mathbf{B}^i$ *for every* $i \in [t]$
- $\alpha, \alpha^1, \ldots, \alpha^t$ *be* a*-bit strings s.t.* $\alpha \neq \alpha^i$ *for every* $i \in [t]$*.*

then

$$(\text{ACB}(\mathbf{X}, \mathbf{Y}, \alpha) \approx_\varepsilon \mathbf{U}_m) \mid (\text{ACB}(\mathbf{X}^1, \mathbf{Y}^1, \alpha^1), \ldots, \text{ACB}(\mathbf{X}^t, \mathbf{Y}^t, \alpha^t)).$$

Theorem 6.2. *There exists another constant* $C_{6.2}$ *such that for every* $n, t, a, m \in \mathbb{N}$ *and* $\varepsilon > 0$*, and every* k, d *such that*

- $k \geq C_{6.2} t^2 \log(a) \log(\frac{na}{\varepsilon})$
- $d \geq C_{6.2} \left(t^2 \log(a) \log(\frac{na}{\varepsilon}) + t^3 \log(\frac{na}{\varepsilon}) + tm\right)$ *and* $d \leq n$*.*

there exists $\text{ACB} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^a \to \{0,1\}^m$ *which is a* (t,k,a,ε)*-advice correlation breaker for linearly correlated sources.*

Note that when $m = 1, t = O((\log\log\log(n))^2), a = O(\log(n)), \varepsilon = 1/\text{poly}(n)$, we need

$$k, d \geq O(\log(n) \log\log(n) (\log\log\log(n))^4).$$

In the rest of this section, for every random variable $\mathbf{R}$, we use $\mathbf{R}^i$ to represent their i-th tampered version. That is, if a random variable $\mathbf{R}$ is defined as $\mathbf{R} := f(\mathbf{X}_1, \ldots, \mathbf{X}_k)$ for some function f and random variables $\mathbf{X}_1, \ldots, \mathbf{X}_k$, then $\mathbf{R}^i := f(\mathbf{X}_1^i, \ldots, \mathbf{X}_k^i)$. If any of the $\mathbf{X}_j^i$ is not defined then we assume $\mathbf{X}_j^i = \mathbf{X}_j$.

6.1 Key Ingredients

The following lemma is a special case of [CGL16, Lemma 6.5], which is obtained by running alternating extraction [DP07, DW09] with the extractor in Lemma 4.10.

Lemma 6.3 (2-look-ahead extractor)**.** *For every $n, d, m \in \mathbb{N}$, $\varepsilon > 0$, there exists an explicit function $\mathrm{laExt}_2 : \{0,1\}^n \times \{0,1\}^d \to (\{0,1\}^m)^2$ which satisfies the following. Let $\mathbf{X}, \mathbf{X}^{[t]} \in \{0,1\}^n$ and $\mathbf{Y}, \mathbf{Y}^{[t]} \in \{0,1\}^d$ be random variables such that $(\mathbf{X}, \mathbf{X}^{[t]})$ is independent of $(\mathbf{Y}, \mathbf{Y}^{[t]})$, $\mathbf{Y} = \mathbf{U}_d$ and $\mathrm{H}_\infty(\mathbf{X}) = k$. Let $r = \max\{m, c_{GUV} \log(d/\varepsilon)\}$. If $d \geq (t+3)c_{GUV} \log(n/\varepsilon) + \log(1/\varepsilon)$ and $k \geq (t+3)r + \log(1/\varepsilon)$, then $(\mathbf{R}_0, \mathbf{R}_1) := \mathrm{laExt}_2(\mathbf{X}, \mathbf{Y})$ and their tampering $(\mathbf{R}_0^{[t]}, \mathbf{R}_1^{[t]})$ satisfy*

$$(\mathbf{R}_0 \approx_{O(\varepsilon)} \mathbf{U}_m) \mid (\mathbf{Y}, \mathbf{Y}^{[t]}),$$
$$(\mathbf{R}_1 \approx_{O(\varepsilon)} \mathbf{U}_m) \mid (\mathbf{Y}, \mathbf{Y}^{[t]}, \mathbf{R}_0, \mathbf{R}_0^{[t]}).$$

The following lemma generalizes the proof idea in [CL16a].

Lemma 6.4 (independence-merging lemma)**.** *Let $(\mathbf{X}, \mathbf{X}^{[t]}) \leftrightarrow \mathbf{Z} \leftrightarrow (\mathbf{Y}, \mathbf{Y}^{[t]})$ be a Markov chain, such that $\mathbf{X}, \mathbf{X}^{[t]} \in \{0,1\}^n$, $\mathbf{Y}, \mathbf{Y}^{[t]} \in \{0,1\}^d$. Moreover, suppose there exists $S, T \subseteq [t]$ such that*

- $(\mathbf{Y} \approx_\delta \mathbf{U}_d) \mid (\mathbf{Z}, \mathbf{Y}^S)$
- $\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid (\mathbf{X}^T, \mathbf{Z})) \geq k + tm + \log(1/\varepsilon)$

Let $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be any (k, ε)-strong seeded extractor, let $\mathbf{W} = \mathrm{Ext}(\mathbf{X}, \mathbf{Y})$ and $\mathbf{W}^j = \mathrm{Ext}(\mathbf{X}^j, \mathbf{Y}^j)$ for every $j \in [t]$. Then

$$(\mathbf{W} \approx_{2\varepsilon+\delta} \mathbf{U}_m) \mid (\mathbf{W}^{S \cup T}, \mathbf{Y}, \mathbf{Y}^{[t]}, \mathbf{Z}).$$

Proof. Without loss of generality, assume that S and T are disjoint. First we give a short sentence as a sketch of the proof: fix $\mathbf{X}^T, \mathbf{Y}^S$ and $\mathbf{W}^S$, observe that $\mathbf{X}$ still have enough entropy and $\mathbf{Y}$ is still uniform and independent of $\mathbf{X}$. Therefore $\mathbf{W} = \mathrm{Ext}(\mathbf{X}, \mathbf{Y})$ is still uniform after fixing $\mathbf{Y}$ and $\mathbf{Y}^{[t]}$. Now $\mathbf{W}^T$ is also fixed, so we can conclude that $\mathbf{W}$ is uniform conditioned on $\mathbf{Z}, \mathbf{Y}, \mathbf{Y}^{[t]}, \mathbf{W}^{S \cup T}$. The formal proof is as follows. First note that $\mathbf{Y}$ is δ-close to uniform conditioned on $\mathbf{Z}, \mathbf{Y}^S$, and will still be δ-close to uniform even when further conditioned on $\mathbf{X}^S, \mathbf{X}^T$. That is,

$$(\mathbf{Y} \approx_\delta \mathbf{U}_d) \mid (\mathbf{X}^T, \mathbf{Z}, \mathbf{X}^S, \mathbf{Y}^S).$$

Since $\mathbf{W}^S$ is a deterministic function of $\mathbf{X}^S, \mathbf{Y}^S$, this implies

$$(\mathbf{Y} \approx_\delta \mathbf{U}_d) \mid (\mathbf{X}^T, \mathbf{Z}, \mathbf{W}^S, \mathbf{Y}^S).$$

Besides, by Lemma 4.12,

$$\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid (\mathbf{X}^T, \mathbf{Z}, \mathbf{W}^S)) \geq k + (t - |s|)m + \log(1/\varepsilon) \geq k + \log(1/\varepsilon).$$

This implies

$$\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid (\mathbf{X}^T, \mathbf{Z}, \mathbf{W}^S, \mathbf{Y}^S)) \geq k + \log(1/\varepsilon)$$

since $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}^S$ is a Markov chain. Next observe that by fixing $\mathbf{X}^T$ and $\mathbf{Y}^S$ in the Markov chain $(\mathbf{X}, \mathbf{X}^{[t]}) \leftrightarrow \mathbf{Z} \leftrightarrow (\mathbf{Y}, \mathbf{Y}^{[t]})$, we get that $(\mathbf{X}, \mathbf{X}^S) \leftrightarrow (\mathbf{X}^T, \mathbf{Z}, \mathbf{Y}^S) \leftrightarrow (\mathbf{Y}, \mathbf{Y}^{[t]})$ is a Markov chain. Since $\mathbf{W}^S = \mathrm{Ext}(\mathbf{X}^S, \mathbf{Y}^S)$, by fixing $\mathbf{W}^S$ we get a Markov chain

$$\mathbf{X} \leftrightarrow (\mathbf{X}^T, \mathbf{Z}, \mathbf{W}^S, \mathbf{Y}^S) \leftrightarrow (\mathbf{Y}, \mathbf{Y}^{[t]}).$$

Therefore Lemma 4.20 implies

$$(\mathbf{W} \approx_{2\varepsilon+\delta} \mathbf{U}_m) \mid (\mathbf{X}^T, \mathbf{Z}, \mathbf{W}^S, \mathbf{Y}^S, \mathbf{Y}, \mathbf{Y}^{[t]}).$$

Now note that $\mathbf{W}^T$ is a deterministic function of $\mathbf{X}^T$ and $\mathbf{Y}^T$, so

$$(\mathbf{W} \approx_{2\varepsilon+\delta} \mathbf{U}_m) \mid (\mathbf{W}^T, \mathbf{Z}, \mathbf{W}^S, \mathbf{Y}^S, \mathbf{Y}, \mathbf{Y}^{[t]})$$

By removing duplicated $\mathbf{Y}^S$ we prove the claim. □

The following lemma gives a non-malleable independence-preserving merger (NIPM) for two strings with entropy recycling. Note that the definition is different from the t-tampering NIPM in [CL16a] in the following way. First, we allow $\mathbf{V}_1, \mathbf{V}_2$ to be weak sources with high conditional entropy conditioned on some tampering and require $\mathbf{S}$ to be uniform, while they require $\mathbf{V}_1, \mathbf{V}_2$ to be uniform but allow $\mathbf{S}$ to be a weak source in their basic merger. Second, in [CL16a] $\mathbf{V}_1, \mathbf{V}_2$ either is independent of all the tampering or has no independence guarantee at all, but in the lemma below we also consider partial independence so that $\mathbf{V}_1, \mathbf{V}_2$ might be independent of a subset of tampering. Besides, similar to [Li19], this NIPM takes an extra input $\mathbf{R}$ as entropy pool and assume that the output length is the same as $|\mathbf{V}_1|, |\mathbf{V}_2|$.

Lemma 6.5 (2-NIPM)**.** *For every $n, s, r \in \mathbb{N}$ and $\varepsilon > 0$, there exists an explicit function $\mathrm{NIPM}_2 : (\{0,1\}^n)^2 \times \{0,1\}^r \times \{0,1\}^s \to \{0,1\}^n$ which satisfies the following. Let $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_1^{[t]}, \mathbf{V}_2^{[t]} \in \{0,1\}^n, \mathbf{S}, \mathbf{S}^{[t]} \in \{0,1\}^s, \mathbf{R}, \mathbf{R}^{[t]} \in \{0,1\}^r$ and $\mathbf{Z}$ be any random variable. Let $T_1, T_2 \subseteq [t]$. Let $g = \max\{r, s, n\}$ and $d = c_{GUV} \log(g/\varepsilon)$. If the following conditions hold:*

- $(\mathbf{S}, \mathbf{S}^{[t]}) \leftrightarrow \mathbf{Z} \leftrightarrow (\mathbf{V}_1, \mathbf{V}_1^{[t]}, \mathbf{V}_2, \mathbf{V}_2^{[t]}, \mathbf{R}, \mathbf{R}^{[t]})$ *is a Markov chain.*
- $\widetilde{\mathrm{H}}_\infty(\mathbf{V}_1 \mid (\mathbf{V}_1^{T_1}, \mathbf{Z})), \widetilde{\mathrm{H}}_\infty(\mathbf{V}_2 \mid (\mathbf{V}_2^{T_2}, \mathbf{Z})) \geq (2t+3)d + \log(1/\varepsilon)$
- $(\mathbf{S} \approx_\delta \mathbf{U}_s) \mid \mathbf{Z}$*, and* $s \geq (3t+4)d + \log(1/\varepsilon)$
- $\widetilde{\mathrm{H}}_\infty(\mathbf{R} \mid \mathbf{Z}) \geq (t+2)n + (2t+2)d + \log(1/\varepsilon)$

Then $\mathbf{W} = \mathrm{NIPM}_2((\mathbf{V}_1, \mathbf{V}_2), \mathbf{R}, \mathbf{S})$ and its tampering $\mathbf{W}^{[t]}$ satisfy

$$(\mathbf{W} \approx_{\delta+O(\varepsilon)} \mathbf{U}_n) \mid (\mathbf{W}^{T_1 \cup T_2}, \mathbf{S}, \mathbf{S}^{[t]}, \mathbf{Z}).$$

Proof. This function requires the following building blocks.

- $\mathrm{Ext}_V : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^d$ is a $(2d, \varepsilon)$-strong seeded extractor from Lemma 4.10
- $\mathrm{Ext}_S : \{0,1\}^s \times \{0,1\}^d \to \{0,1\}^d$ is a $(2d, \varepsilon)$-strong seeded extractor from Lemma 4.10
- $\mathrm{Ext}_R : \{0,1\}^r \times \{0,1\}^d \to \{0,1\}^n$ is a $(2n, \varepsilon)$-strong seeded extractor from Lemma 4.10

The construction of NIPM_2 is as follows. Let $\mathbf{Q}_1$ be a length-d prefix of $\mathbf{S}$. Now apply the following steps:

$$\begin{aligned}
\mathbf{P}_1 &= \mathrm{Ext}_V(\mathbf{V}_1, \mathbf{Q}_1) \\
\mathbf{Q}_2 &= \mathrm{Ext}_S(\mathbf{S}, \mathbf{P}_1) \\
\mathbf{P}_2 &= \mathrm{Ext}_V(\mathbf{V}_2, \mathbf{Q}_1) \\
\mathbf{Q}_r &= \mathrm{Ext}_S(\mathbf{S}, \mathbf{P}_2) \\
\mathbf{W} &= \mathrm{Ext}_R(\mathbf{R}, \mathbf{Q}_r).
\end{aligned}$$

Then $\mathbf{W}$ will be the output of $\mathrm{NIPM}_2((\mathbf{V}_1, \mathbf{V}_2), \mathbf{R}, \mathbf{S})$. To prove the correctness, we start with the Markov chain $(\mathbf{S}, \mathbf{S}^{[t]}) \leftrightarrow \mathbf{Z} \leftrightarrow (\mathbf{V}_1, \mathbf{V}_1^{[t]}, \mathbf{V}_2, \mathbf{V}_2^{[t]}, \mathbf{R}, \mathbf{R}^{[t]})$, and then we fix $(\mathbf{Q}_1, \mathbf{Q}_1^{[t]})$, $(\mathbf{P}_1, \mathbf{P}_1^{[t]})$, $(\mathbf{Q}_2, \mathbf{Q}_2^{[t]})$ and $(\mathbf{P}_2, \mathbf{P}_2^{[t]})$ in order. Observe that as long as the extractor applied in each step satisfies the conditional min-entropy requirement in Lemma 6.4, by applying Lemma 6.4 in each step we get

$$\begin{aligned}
(\mathbf{P}_1 \approx_{\delta} \mathbf{U}_d) \mid (\mathbf{P}_1^{T_1} &, \mathbf{Z}, \mathbf{Q}_1, \mathbf{Q}_1^{[t]}) \\
(\mathbf{Q}_2 \approx_{\delta+2\varepsilon} \mathbf{U}_d) \mid (\mathbf{Q}_2^{T} &, \mathbf{Z}, \mathbf{Q}_1, \mathbf{Q}_1^{[t]}, \mathbf{P}_1, \mathbf{P}_1^{[t]}) \\
(\mathbf{P}_2 \approx_{\delta+4\varepsilon} \mathbf{U}_d) \mid (\mathbf{P}_2^{T_1 \cup T_2} &, \mathbf{Z}, \mathbf{Q}_1, \mathbf{Q}_1^{[t]}, \mathbf{P}_1, \mathbf{P}_1^{[t]}, \mathbf{Q}_2, \mathbf{Q}_2^{[t]}) \\
(\mathbf{Q}_r \approx_{\delta+6\varepsilon} \mathbf{U}_d) \mid (\mathbf{Q}_r^{T_1 \cup T_2} &, \mathbf{Z}, \mathbf{Q}_1, \mathbf{Q}_1^{[t]}, \mathbf{P}_1, \mathbf{P}_1^{[t]}, \mathbf{Q}_2, \mathbf{Q}_2^{[t]}, \mathbf{P}_2, \mathbf{P}_2^{[t]},) \\
(\mathbf{W} \approx_{\delta+8\varepsilon} \mathbf{U}_m) \mid (\mathbf{W}^{T_1 \cup T_2} &, \mathbf{Z}, \mathbf{Q}_1, \mathbf{Q}_1^{[t]}, \mathbf{P}_1, \mathbf{P}_1^{[t]}, \mathbf{Q}_2, \mathbf{Q}_2^{[t]}, \mathbf{P}_2, \mathbf{P}_2^{[t]}, \mathbf{Q}_r, \mathbf{Q}_r^{[t]}).
\end{aligned} \tag{5}$$

Besides, observe that by further fixing $(\mathbf{Q}_r, \mathbf{Q}_r^{[t]}, \mathbf{W}^{T_1 \cup T_2})$ in the last step we have a Markov chain

$$(\mathbf{S}, \mathbf{S}^{[t]}) \leftrightarrow (\mathbf{Z}, \mathbf{Q}_1, \mathbf{Q}_1^{[t]}, \mathbf{P}_1, \mathbf{P}_1^{[t]}, \mathbf{Q}_2, \mathbf{Q}_2^{[t]}, \mathbf{P}_2, \mathbf{P}_2^{[t]}, \mathbf{Q}_r, \mathbf{Q}_r^{[t]}, \mathbf{W}^{T_1 \cup T_2}) \leftrightarrow \mathbf{W}.$$

By applying Lemma 4.17 to (5) we prove the claim. Finally we need to verify that at each step the conditional entropy of $\mathbf{V}_1, \mathbf{V}_2, \mathbf{S}, \mathbf{R}$ actually satisfies the requirement in Lemma 6.4. Now we define $\mathbf{Z}' = (\mathbf{Q}_1^{[t]}, \mathbf{Q}_1), (\mathbf{P}_1^{[t]}, \mathbf{P}_1), (\mathbf{Q}_2^{[t]}, \mathbf{Q}_2)$. Observe that

$$\begin{aligned}
\widetilde{\mathrm{H}}_{\infty}(\mathbf{V}_1 \mid (\mathbf{Z}, \mathbf{Z}')) &\geq \widetilde{\mathrm{H}}_{\infty}(\mathbf{V}_1 \mid \mathbf{Z}) - (t+1)d \geq 2d + td + \log(1/\varepsilon) \\
\widetilde{\mathrm{H}}_{\infty}(\mathbf{V}_2 \mid (\mathbf{Z}, \mathbf{Z}')) &\geq \widetilde{\mathrm{H}}_{\infty}(\mathbf{V}_2 \mid \mathbf{Z}) - (t+1)d \geq 2d + td + \log(1/\varepsilon) \\
\widetilde{\mathrm{H}}_{\infty}(\mathbf{S} \mid (\mathbf{Z}, \mathbf{Z}')) &\geq \widetilde{\mathrm{H}}_{\infty}(\mathbf{S} \mid \mathbf{Z}) - 2(t+1)d \geq 2d + td + \log(1/\varepsilon) \\
\widetilde{\mathrm{H}}_{\infty}(\mathbf{R} \mid (\mathbf{Z}, \mathbf{Z}', \mathbf{P}_2^{[t]}, \mathbf{P}_2)) &\geq \widetilde{\mathrm{H}}_{\infty}(\mathbf{R} \mid \mathbf{Z}) - 2(t+1)d \geq 2n + tn + \log(1/\varepsilon).
\end{aligned}$$

Therefore in each application of Lemma 6.4 the entropy requirement is satisfied. □

The following lemma uses the flip-flop idea from [Coh16a].

Lemma 6.6. *Let* $\mathrm{FFAssign} : (\{0,1\}^n)^2 \times \{0,1\}^a \to (\{0,1\}^n)^{2a}$ *be the function defined as follows. Let* $r_0, r_1 \in \{0,1\}^n$ *and* $\alpha \in \{0,1\}^a$*. Let* α_j *denote the* j*-th bit of* α*. Then* $\mathrm{FFAssign}(r_0, r_1, \alpha) := (r_{\alpha_1}, r_{1-\alpha_1}, \ldots, r_{\alpha_a}, r_{1-\alpha_a})$*. Now suppose for some random variables* $\mathbf{R}_0, \mathbf{R}_0^{[t]}, \mathbf{R}_1, \mathbf{R}_1^{[t]} \in \{0,1\}^n$*,* $\mathbf{Z}$ *and some strings* $\alpha, \alpha^1, \ldots, \alpha^t \in \{0,1\}^a$*, the following conditions hold:*

- $(\mathbf{R}_0 \approx_{\varepsilon} \mathbf{U}_n) \mid \mathbf{Z}$
- $(\mathbf{R}_1 \approx_{\varepsilon} \mathbf{U}_n) \mid (\mathbf{Z}, \mathbf{R}_0^{[t]})$
- $\alpha^j \neq \alpha$ *for every* $j \in [t]$

Then there exist sets $S_1, \ldots, S_{2a} \subseteq [t]$ *which satisfy the following. First,* $\bigcup_{i=1}^{2a} S_i = [t]$*. Second, let* $(\mathbf{V}_1, \ldots, \mathbf{V}_{2a}) = \mathrm{FFAssign}(\mathbf{R}_0, \mathbf{R}_1, \alpha)$ *and* $(\mathbf{V}_1^j, \ldots, \mathbf{V}_{2a}^j) = \mathrm{FFAssign}(\mathbf{R}_0^j, \mathbf{R}_1^j, \alpha^j)$*. Then for every* $j \in [2a]$*,*

$$(\mathbf{V}_j \approx_{\varepsilon} \mathbf{U}_n) \mid (\mathbf{Z}, \mathbf{V}_j^{S_j}).$$

Proof. For every $j \in [a]$, let $S_{2j-\alpha_j} = \{i \in [t] : \alpha_j^i \neq \alpha_j\}$, $S_{2j-1+\alpha_j}$ be empty. Observe that for every $i \in [t]$, there exists p_i such that $\alpha_{p_i} \neq \alpha_{p_i}^i$, which implies $i \in S_{2p_i-\alpha_{p_i}}$. Therefore $\bigcup_{i=1}^{2a} S_i = [t]$. Besides, note that for every $j \in [2a]$, either $\mathbf{V}_j = \mathbf{R}_1$ and $\mathbf{V}_j^{S_j} = \mathbf{R}_0^{S_j}$ or $\mathbf{V}_j = \mathbf{R}_0$ and S_j is empty. By the given condition, $(\mathbf{V}_j \approx_\varepsilon \mathbf{U}_n) \mid (\mathbf{Z}, \mathbf{V}_j^{S_j})$ in both cases. □

6.2 Construction and proof

Now we are ready to prove Theorem 6.2.

Proof of Theorem 6.2. Let $\varepsilon_0 = (\varepsilon/Ca)$ for some large enough constant C. Let $s = c_{GUV} \log(n/\varepsilon_0)$, we take $s_y = (t+4)s$ and $s_x = (2t+9)s$. Let $v = \max\{(3t+5)(s_x+s_y), m\} = O(t^2 s + m)$. We need to guarantee that $d \geq (t+3)v + (2t+3)s_x + \log(2a)(t+1)(s_x+s_y) = O(t^2 \log(a)s + t^3 s + tm)$, and $\mathrm{H}_\infty(\mathbf{A}) \geq (\log(2a)(t+1)+1)s_y + 3\log(1/\varepsilon_0) = O(t^2 \log(a)s)$. To construct ACB, we need the following building blocks:

- $\mathrm{LExt} : \{0,1\}^n \times \{0,1\}^{s_x} \to \{0,1\}^{s_y}$, which is a $(s_y + 2\log(1/\varepsilon_0), \varepsilon_0)$-strong linear seeded extractor from Lemma 4.9.
- $\mathrm{laExt}_2 : \{0,1\}^d \times \{0,1\}^{s_y} \to (\{0,1\}^v)^2$ from Lemma 6.3 with error parameter ε_0.
- $\mathrm{FFAssign} : (\{0,1\}^v)^2 \times \{0,1\}^a \to (\{0,1\}^v)^{2a}$ from Lemma 6.6.
- $\mathrm{NIPM}_2 : (\{0,1\}^v)^2 \times \{0,1\}^d \times \{0,1\}^{s_y} \to (\{0,1\}^v)$ from Lemma 6.5 with error parameter ε_0.

Given $\mathbf{X} = \mathbf{A} + \mathbf{B}, \mathbf{Y}$, the construction works as follows.

1. Take $\mathbf{W}_0$ to be a prefix of $\mathbf{Y}$ of length s_x.
2. Compute $\mathbf{Q}_0 := \mathrm{LExt}(\mathbf{X}, \mathbf{W}_0)$.
3. Compute $(\mathbf{R}_0, \mathbf{R}_1) := \mathrm{laExt}_2(\mathbf{Y}, \mathbf{Q}_0)$.
4. Define $(\mathbf{V}_{0,1}, \mathbf{V}_{0,2}, \ldots, \mathbf{V}_{0,(2a-1)}, \mathbf{V}_{0,2a}) := \mathrm{FFAssign}((\mathbf{R}_0, \mathbf{R}_1), \alpha)$.
5. For i from 1 to $h = \log(2a)$, repeat the following steps:
   - I Take $\mathbf{W}_i$ to be a prefix of $\mathbf{V}_{i-1,1}$ of length s_x.
   - II Compute $\mathbf{Q}_i := \mathrm{LExt}(\mathbf{X}, \mathbf{W}_i)$.
   - III For every $j \in [2a/2^i]$, compute $\mathbf{V}_{i,j} := \mathrm{NIPM}_2((\mathbf{V}_{(i-1),(2j-1)}, \mathbf{V}_{(i-1),2j}), \mathbf{Y}, \mathbf{Q}_i)$.
6. Output $\mathrm{ACB}(\mathbf{X}, \mathbf{Y}, \alpha) := \mathbf{V}_{h,1}$.

To prove that this construction works, define $\mathbf{A}_i := \mathrm{LExt}(\mathbf{A}, \mathbf{W}_i), \mathbf{B}_i := \mathrm{LExt}(\mathbf{B}, \mathbf{W}_i)$. We also use $\mathbf{V}_i$ to denote $(\mathbf{V}_{i,1}, \ldots, \mathbf{V}_{i,(2a/2^i)})$. First note that $(\mathbf{A}_0 \approx_{\varepsilon_0} \mathbf{U}_{s_y}) \mid (\mathbf{W}_0, \mathbf{W}_0^{[t]}, \mathbf{B}_0, \mathbf{B}_0^{[t]})$. Since $\widetilde{\mathrm{H}}_\infty(\mathbf{Y} \mid (\mathbf{W}_0, \mathbf{W}_0^{[t]}, \mathbf{B}_0, \mathbf{B}_0^{[t]})) \geq (t+3)v + 2\log(1/\varepsilon_0)$, by Lemma 6.3 and Lemma 6.6, there exists sets $S_{0,1}, S_{0,2}, \ldots, S_{0,2a}$ such that $\bigcup_{j=1}^{2a} S_{0,j} = [t]$, and for every $j \in [2a]$ we have

$$(\mathbf{V}_{0,j} \approx_{O(\varepsilon_0)} \mathbf{U}_v) \mid (\mathbf{W}_0, \mathbf{W}_0^{[t]}, \mathbf{Q}_0, \mathbf{Q}_0^{[t]}, \mathbf{B}_0, \mathbf{B}_0^{[t]}, \mathbf{V}_{0,j}^{S_{0,j}}).$$

Now for every $i \in [h]$ and $j \in [2a/2^i]$, define $S_{i,j} = S_{i-1,2j-1} \cup S_{i-1,2j}$. Note that $S_{h,1} = [t]$. Besides, for i from 0 to h, define $\mathbf{Z}_i = (\mathbf{W}_0, \mathbf{W}_0^{[t]}, \mathbf{Q}_0, \mathbf{Q}_0^{[t]}, \mathbf{B}_0, \mathbf{B}_0^{[t]})$. We inductively prove the following claims for i from 1 to h.

- $(\mathbf{A}, \mathbf{A}^{[t]}) \leftrightarrow \mathbf{Z}_{<i} \leftrightarrow (\mathbf{B}, \mathbf{B}^{[t]}, \mathbf{Y}, \mathbf{Y}^{[t]}, \mathbf{V}_{(i-1)}, \mathbf{V}_{(i-1)}^{[t]}, \mathbf{W}_i, \mathbf{W}_i^{[t]}, \mathbf{B}_i, \mathbf{B}_i^{[t]})$ forms a Markov chain. This is by induction hypothesis and by the fact that $\mathbf{W}_i, \mathbf{B}_i$ is a deterministic function of $\mathbf{B}, \mathbf{V}_{(i-1),1}$.
- $(\mathbf{W}_i \approx_{O(2^i\varepsilon_0)} \mathbf{U}_{s_x}) \mid \mathbf{Z}_{<i}$. (by (6))
- $\widetilde{\mathrm{H}}_\infty(\mathbf{A} \mid \mathbf{Z}_{<i}) \geq \mathrm{H}_\infty(\mathbf{A}) - i(t+1)s_y \geq s_y + 3\log(1/\varepsilon_0)$ (by induction hypothesis and chain rule)
- $(\mathbf{A}_i \approx_{O(2^i\varepsilon_0)} \mathbf{U}_{s_y}) \mid (\mathbf{Z}_{<i}, \mathbf{W}_i, \mathbf{W}_i^{[t]}, \mathbf{B}_i, \mathbf{B}_i^{[t]})$ (by LExt being an extractor)
- $(\mathbf{A}, \mathbf{A}^{[t]}, \mathbf{A}_i, \mathbf{A}_i^{[t]}) \leftrightarrow (\mathbf{Z}_{<i}, \mathbf{W}_i, \mathbf{W}_i^{[t]}, \mathbf{B}_i, \mathbf{B}_i^{[t]}) \leftrightarrow (\mathbf{B}, \mathbf{B}^{[t]}, \mathbf{Y}, \mathbf{Y}^{[t]}, \mathbf{V}_{(i-1)}, \mathbf{V}_{(i-1)}^{[t]})$ forms a Markov chain
- $(\mathbf{Q}_i \approx_{O(2^i\varepsilon_0)} \mathbf{U}_{s_y}) \mid (\mathbf{Z}_{<i}, \mathbf{W}_i, \mathbf{W}_i^{[t]}, \mathbf{B}_i, \mathbf{B}_i^{[t]})$ (sum of $\mathbf{U}$ and any fixed string is still $\mathbf{U}$)
- $(\mathbf{A}, \mathbf{A}^{[t]}, \mathbf{Q}_i, \mathbf{Q}_i^{[t]}) \leftrightarrow (\mathbf{Z}_{<i}, \mathbf{W}_i, \mathbf{W}_i^{[t]}, \mathbf{B}_i, \mathbf{B}_i^{[t]}) \leftrightarrow (\mathbf{B}, \mathbf{B}^{[t]}, \mathbf{Y}, \mathbf{Y}^{[t]}, \mathbf{V}_{(i-1)}, \mathbf{V}_{(i-1)}^{[t]})$ forms a Markov chain. (This is because $\mathbf{Q}_i = \mathbf{A}_i + \mathbf{B}_i$.)
- For every $j \in [2a/2^{i-1}]$, there exists $\widetilde{\mathbf{V}}_{(i-1),j}$ s.t.

  $$\widetilde{\mathrm{H}}_\infty\left(\widetilde{\mathbf{V}}_{(i-1),j} \mid \left(\mathbf{V}_{(i-1),j}^{S_{(i-1),j}}, \mathbf{Z}_{<i}, \mathbf{W}_i, \mathbf{W}_i^{[t]}, \mathbf{B}_i, \mathbf{B}_i^{[t]}\right)\right) \geq v-(t+1)(s_x+s_y) \geq (2t+3)s_x+\log(1/\varepsilon_0)$$

  and

  $$\left(\mathbf{V}_{(i-1),j} \approx_{O(2^i\varepsilon_0)} \widetilde{\mathbf{V}}_{(i-1),j}\right) \mid \left(\mathbf{V}_{(i-1),j}^{S_{(i-1),j}}, \mathbf{Z}_{<i}, \mathbf{W}_i, \mathbf{W}_i^{[t]}, \mathbf{B}_i, \mathbf{B}_i^{[t]}\right).$$

  This $\widetilde{\mathbf{V}}_{(i-1),j}$ comes from the $\mathbf{U}_v$ in (6), which has $\widetilde{\mathrm{H}}_\infty\left(\widetilde{\mathbf{V}}_{(i-1),j} \mid \left(\mathbf{V}_{(i-1),j}^{S_{(i-1),j}}, \mathbf{Z}_{<i}\right)\right) = v$. The conditional entropy bound is by chain rule (Lemma 4.12).
- $\widetilde{\mathrm{H}}_\infty\left(\mathbf{Y} \mid \left(\mathbf{Z}_{<i}, \mathbf{W}_i, \mathbf{W}_i^{[t]}, \mathbf{B}_i, \mathbf{B}_i^{[t]}\right)\right) \geq d - i(t+1)(s_x+s_y) \geq (t+2)v + (2t+2)s_x + \log(1/\varepsilon_0)$.
- For every $j \in [2a/2^i]$,

  $$\left(\mathbf{V}_{i,j} \approx_{O(2^{i+1}\varepsilon_0)} \mathbf{U}_v\right) \mid \left(\mathbf{V}_{i,j}^{S_{i,j}}, \mathbf{Z}_{<i}, \mathbf{W}_i, \mathbf{W}_i^{[t]}, \mathbf{B}_i, \mathbf{B}_i^{[t]}, \mathbf{Q}_i, \mathbf{Q}_i^{[t]}\right),$$

  which can be rewritten as

  $$\left(\mathbf{V}_{i,j} \approx_{O(2^{i+1}\varepsilon_0)} \mathbf{U}_v\right) \mid \left(\mathbf{V}_{i,j}^{S_{i,j}}, \mathbf{Z}_{\leq i}\right). \tag{6}$$

  This is by applying NIPM_2 (Lemma 6.5) on $(\widetilde{\mathbf{V}}_{(i-1),(2j-1)}, \widetilde{\mathbf{V}}_{(i-1),2j})$. The error blows up by a factor 2 and will dominate the additive $O(\varepsilon_0)$ error in previous steps.
- $(\mathbf{A}, \mathbf{A}^{[t]}) \leftrightarrow \mathbf{Z}_{\leq i} \leftrightarrow (\mathbf{B}, \mathbf{B}^{[t]}, \mathbf{Y}, \mathbf{Y}^{[t]}, \mathbf{V}_i, \mathbf{V}_i^{[t]})$ forms a Markov chain. (by fixing $\mathbf{Q}_i, \mathbf{Q}_i^{[t]}$)

Finally observe that (6) for $i = h, j = 1$ is exactly what we want. The error is $O(2^h\varepsilon_0) = O(a\varepsilon_0)$. □

7 Main Theorem

In this section we prove our main theorem.

Theorem 7.1 (Theorem 1, restated)**.** *For every constant $\varepsilon > 0$, there exists a constant C such that for every large enough n, there exists an explicit extractor* $\mathrm{AffExt} : \{0,1\}^n \to \{0,1\}$ *for (n,k)-affine source with error ε for any $k \geq C \log(n) \log\log(n) \log\log\log^6(n)$.*

Proof. Let $\mathbf{X}$ be a (n,k)-affine source. Let $t = O(\frac{\log^2(1/\varepsilon)}{\varepsilon^2})$ be large enough so that $C_{4.28}\frac{\log(t)}{\sqrt{t}} \leq \frac{\varepsilon}{3}$. Let $\mathrm{LSRExt} : \{0,1\}^n \times [D] \times [A] \to \{0,1\}^{c\log n}$ be a strong linear seeded somewhere random extractor from Theorem 5.1 where $\delta_{5.1} = 0.4$, and assume that c and k is large enough so that

- k satisfies the entropy requirement in Theorem 5.1, i.e. $k \geq \beta_{5.1} c \log(n)$.
- An explicit (tA, k_0, a, γ)-advice correlation breaker $\mathrm{ACB} : \{0,1\}^n \times \{0,1\}^{c\log(n)} \times \{0,1\}^a \to \{0,1\}$ exists based on Theorem 6.2, where $k_0 = k - tA(c\log(n))$, $a = \log(AD)$ and $\gamma = \varepsilon/(3tD^t)$.

We set the parameters later. Now for every $s \in \{0,1\}^D, z \in \{0,1\}^A$, compute

$$\begin{aligned}\mathbf{Y}_{s,z} &:= \mathrm{LSRExt}(\mathbf{X}, s, z)\\ \mathbf{R}_{s,z} &:= \mathrm{ACB}(\mathbf{X}, \mathbf{Y}_{s,z}, (s,z))\\ \mathbf{P}_s &:= \oplus_{j=1}^{z} \mathbf{R}_{s,j}.\end{aligned}$$

Then the output is

$$\mathrm{AffExt}(\mathbf{X}) := \mathrm{Maj}(\mathbf{P}_1, \ldots, \mathbf{P}_D).$$

Observe that the running time is polynomial, since all the functions are explicit and $D, A \leq n^{O(1)}$. Next we prove the correctness of this construction. First note that by Theorem 5.1 there exists a set $B \in [D]$ of size at most $D^{0.4}$ such that for $\forall s \in [D]\backslash B, \exists z \in [A]$ s.t. $\mathbf{Y}_{s,z}$ is uniform. Now consider any $T \subseteq [D]\backslash B$ such that $|T| = t$. Let $T = \{s_1, s_2, \ldots, s_t\}$. For every $i \in [t]$, let $z_i \in [A]$ be the index such that $\mathbf{Y}_{s_i,z_i}$ is uniform. Observe that for every $i \in [t], j \in [A]$, the function $\mathrm{LSRExt}(\cdot, s_i, j)$ is linear, and so is their concatenation. Since their total length is $tA(c\log(n))$, by Lemma 4.21, there exists $\mathbf{A}, \mathbf{B}$ such that $\mathbf{A} + \mathbf{B} = \mathbf{X}$, $\mathbf{A}$ is independent of $(\mathbf{B}, \{\mathbf{Y}_{s_i,j}\}_{i\in[t],j\in[A]})$, and $\mathrm{H}_\infty(\mathbf{A}) \geq k - tA(c\log(n) = k_0$. Then for every $i \in [t]$, by the definition of ACB we get that for every $i \in [t]$,

$$(\mathbf{R}_{s_i,z_i} \approx_\gamma \mathbf{U}_1) \mid \left(\{\mathbf{R}_{s_i,j}\}_{j\in[A]\backslash\{z_i\}}, \{\mathbf{R}_{s',j}\}_{s'\in T\backslash\{s_i\}, j\in[A]}\right).$$

This implies that

$$(\mathbf{P}_{s_i} \approx_\gamma \mathbf{U}_1) \mid \left(\{\mathbf{P}_{s'}\}_{s'\in T\backslash\{s_i\}}\right),$$

and then

$$(\mathbf{P}_{s_1}, \ldots, \mathbf{P}_{s_{i-1}}, \mathbf{P}_{s_i}, \mathbf{U}_{t-i}) \approx_\gamma (\mathbf{P}_{s_1}, \ldots, \mathbf{P}_{s_{i-1}}, \mathbf{U}_1, \mathbf{U}_{t-i}).$$

By triangle inequality we eventually get

$$(\mathbf{P}_{s_1}, \mathbf{P}_{s_2}, \ldots, \mathbf{P}_{s_t}) \approx_{t\gamma} \mathbf{U}_t.$$

In other word, for every $T \subseteq [D]\backslash B$ s.t. $|T| = t$, we have $\mathbf{P}_T$ is $t\gamma$-close to $\mathbf{U}_t$, which means $(\mathbf{P}_1, \ldots, \mathbf{P}_D)$ is a $(D^{0.4}, t, t\gamma)$-NOBF source. Therefore by Lemma 4.28 we can conclude that

$$\Delta\left(\mathrm{AffExt}(\mathbf{X}); \mathbf{U}_1\right) \leq C_{4.28}\frac{\log(t)}{\sqrt{t}} + C_{4.28}D^{-0.1} + tD^t\gamma,$$

which is at most ε as long as n is large enough so that $C_{4.28}D^{-0.1} \leq \varepsilon/3$.

Finally we consider the parameter restriction of c, k. First note that

$$\begin{aligned} c\log(n) &\geq C_{6.2}\left((tA)^3 + (tA)^2\log\log(DA)\right)\log\left(\frac{n\log(DA)}{\varepsilon/3tD^t}\right) \\ &= O\left(\left(\log^6(c) + \log^4(c)\log\log(n)\right)\cdot\left(\log(n) + \log\log(c)\right)\right). \end{aligned}$$

Observe that there exists some large enough constant C_0 such that $c = C_0\log\log(n)(\log\log\log(n))^4$ satisfies the above restriction. This implies $A = O(\log^2(c)) = O((\log\log\log(n))^2)$. Next, we need

$$k - (tAc\log(n)) \geq C_{6.2}\left((tA)^2\log\log(DA)\right)\log\left(\frac{n\log(DA)}{\varepsilon/3tD^t}\right) = O(c\log(n)).$$

Therefore the entropy requirement is $k = O(Ac\log(n)) = O(\log(n)\log\log(n)(\log\log\log(n))^6)$. Note that the entropy requirement for *Theorem* 5.1, $k \geq \beta_{5.1}c\log(n)$, is also satisfied. □

Acknowledgement

We thank David Zuckerman for discussions about affine extractors, which led to this work.

References

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

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

[BDL16] Jean Bourgain, Zeev Dvir, and Ethan Leeman. Affine extractors over large fields with exponential error. *computational complexity*, 25(4):921–931, 2016.

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

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

[Blu86] Manuel Blum. Independent unbiased coin flips from a correlated biased source—a finite state markov chain. *Combinatorica*, 6(2):97–108, 1986.

[Bou07] Jean Bourgain. On the construction of affine extractors. *GAFA Geometric And Functional Analysis*, 17(1):33–57, 2007.

[BSRZ15] Eli Ben-Sasson and Noga Ron-Zewi. From affine to two-source extractors via approximate duality. *SIAM Journal on Computing*, 44(6):1670–1697, 2015.

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

[CG20] Eshan Chattopadhyay and Jesse Goodman. Explicit extremal designs and applications to extractors. *Electron. Colloquium Comput. Complex.*, 27:106, 2020.

[CGH+85] Benny Chor, Oded Goldreich, Johan Hasted, Joel Freidmann, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 396–407. IEEE, 1985.

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 285–298. ACM, 2016.

[Che10] Mahdi Cheraghchi. *Applications of Derandomization Theory in Coding*. PhD thesis, EPFL, Lausanne, Switzerland, 2010.

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

[CL16b] Eshan Chattopadhyay and Xin Li. Extractors for sumset sources. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 299–311. ACM, 2016.

[Coh16a] Gil Cohen. Local correlation breakers and applications to three-source extractors and mergers. *SIAM J. Comput.*, 45(4):1297–1338, 2016.

[Coh16b] Gil Cohen. Making the most of advice: New correlation breakers and their applications. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 188–196. IEEE Computer Society, 2016.

[Coh16c] Gil Cohen. Non-malleable extractors - new tools and improved constructions. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 8:1–8:29. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

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

[CS16] Gil Cohen and Leonard J. Schulman. Extractors for near logarithmic min-entropy. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 178–187. IEEE Computer Society, 2016.

[CT15] Gil Cohen and Avishay Tal. Two structural results for low degree polynomials and applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, pages 680–709, 2015.

[CZ19] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. *Annals of Mathematics*, 189(3):653–705, 2019.

[DG10] Matt DeVos and Ariel Gabizon. Simple affine extractors using dimension expansion. In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 50–57. IEEE, 2010.

[DKSS13] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the method of multiplicities, with applications to kakeya sets and mergers. *SIAM Journal on Computing*, 42(6):2305–2328, 2013.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 227–237. IEEE Computer Society, 2007.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 601–610. ACM, 2009.

[FGHK16] Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-3n lower bound for the circuit complexity of an explicit function. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 89–98, 2016.

[GR08] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. *Combinatorica*, 28(4):415–440, 2008.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from parvaresh-vardy codes. *J. ACM*, 56(4):20:1–20:34, 2009.

[GW97] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Struct. Algorithms*, 11(4):315–343, 1997.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[KRVZ06] Jesse Kamp, Anup Rao, Salil Vadhan, and David Zuckerman. Deterministic extractors for small-space sources. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 691–700, 2006.

[KZ07] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2007.

[Li11] Xin Li. A new approach to affine extractors and dispersers. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, USA, June 8-10, 2011*, pages 137–147. IEEE Computer Society, 2011.

[Li13] Xin Li. Extractors for a constant number of independent sources with polylogarithmic min-entropy. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 100–109. IEEE Computer Society, 2013.

[Li15] Xin Li. Non-malleable condensers for arbitrary min-entropy, and almost optimal protocols for privacy amplification. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 502–531. Springer, 2015.

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

[Li17] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1144–1156. ACM, 2017.

[Li19] Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 28:1–28:49. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[LRVW03] Chi-Jen Lu, Omer Reingold, Salil Vadhan, and Avi Wigderson. Extractors: Optimal up to constant factors. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 602–611, 2003.

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

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[Rao09] Anup Rao. Extractors for low-weight affine sources. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 95–101. IEEE Computer Society, 2009.

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

[Sha04] Ronen Shaltiel. Recent developments in explicit constructions of extractors. In *Current Trends in Theoretical Computer Science: The Challenge of the New Century Vol 1: Algorithms and Complexity Vol 2: Formal Models and Semantics*, pages 189–228. World Scientific, 2004.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *J. ACM*, 48(4):860–879, 2001.

[TV00] Luca Trevisan and Salil Vadhan. Extracting randomness from samplable distributions. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 32–42. IEEE, 2000.

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

[vN51] J. von Neumann. Various techniques used in connection with random digits. *Applied Math Series*, 12:36–38, 1951. Notes by G.E. Forsythe, National Bureau of Standards. Reprinted in *Von Neumann's Collected Works*, 5:768-770, 1963.

[Yeh11] Amir Yehudayoff. Affine extractors over prime fields. *Combinatorica*, 31(2):245–256, 2011.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory Comput.*, 3(1):103–128, 2007.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il