Non-Malleable Extractors and Non-Malleable Codes: Partially Optimal Constructions

Xin Li
Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA
http://www.cs.jhu.edu/~lixints/
lixints@cs.jhu.edu

Abstract

The recent line of study on randomness extractors has been a great success, resulting in exciting new techniques, new connections, and breakthroughs to long standing open problems in several seemingly different topics. These include seeded non-malleable extractors, privacy amplification protocols with an active adversary, independent source extractors (and explicit Ramsey graphs), and non-malleable codes in the split state model. Previously, the best constructions are given in [54]: seeded non-malleable extractors with seed length and entropy requirement $O(\log n + \log(1/\epsilon)\log\log(1/\epsilon))$ for error ϵ; two-round privacy amplification protocols with optimal entropy loss for security parameter up to $\Omega(k/\log k)$, where k is the entropy of the shared weak source; two-source extractors for entropy $O(\log n \log\log n)$; and non-malleable codes in the 2-split state model with rate $\Omega(1/\log n)$. However, in all cases there is still a gap to optimum and the motivation to close this gap remains strong.

In this paper, we introduce a set of new techniques to further push the frontier in the above questions. Our techniques lead to improvements in all of the above questions, and in several cases partially optimal constructions. This is in contrast to all previous work, which only obtain close to optimal constructions. Specifically, we obtain:

1. A seeded non-malleable extractor with seed length $O(\log n) + \log^{1+o(1)}(1/\epsilon)$ and entropy requirement $O(\log\log n + \log(1/\epsilon))$, where the entropy requirement is asymptotically optimal by a recent result of Gur and Shinkar [40];
2. A two-round privacy amplification protocol with optimal entropy loss for security parameter up to $\Omega(k)$, which solves the privacy amplification problem completely;[1]
3. A two-source extractor for entropy $O(\frac{\log n \log\log n}{\log\log\log n})$, which also gives an explicit Ramsey graph on N vertices with no clique or independent set of size $(\log N)^{O(\frac{\log\log\log N}{\log\log\log\log N})}$; and
4. The first explicit non-malleable code in the 2-split state model with *constant* rate, which has been a major goal in the study of non-malleable codes for quite some time. One small caveat is that the error of this code is only (an arbitrarily small) constant, but we can also achieve negligible error with rate $\Omega(\log\log\log n/\log\log n)$, which already improves the rate in [54] exponentially.

We believe our new techniques can help to eventually obtain completely optimal constructions in the above questions, and may have applications in other settings.

2012 ACM Subject Classification Theory of computation → Expander graphs and randomness extractors

Keywords and phrases extractor, non-malleable, privacy, codes

Digital Object Identifier 10.4230/LIPIcs.CCC.2019.28

Funding *Xin Li*: Supported by NSF award CCF-1617713.

[1] Except for the communication complexity, which is of secondary concern to this problem.

© Xin Li;
licensed under Creative Commons License CC-BY
34th Computational Complexity Conference (CCC 2019).
Editor: Amir Shpilka; Article No. 28; pp. 28:1–28:49
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The study of randomness extractors has been a central line of research in the area of pseudorandomness, where the goal is to understand how to use randomness more efficiently in computation. As fundamental objects in this area, randomness extractors are functions that transform imperfect random sources into nearly uniform random bits. Their original motivation is to bridge the gap between the uniform random bits required in standard applications (such as in randomized algorithms, distributed computing, and cryptography), and practical random sources which are almost always biased (either because of natural noise or adversarial information leakage). However the study of these objects has led to applications far beyond this motivation, in several different fields of computer science and combinatorics (e.g., coding theory, graph theory, and complexity theory).

The inputs to a randomness extractor are usually imperfect randomness, modeled by the notion of general weak random sources with a certain amount of entropy.

▶ **Definition 1.** *The* min-entropy *of a random variable X is*

$$H_\infty(X) = \min_{x \in \mathsf{supp}(X)} \log_2(1/\Pr[X = x]).$$

For $X \in \{0,1\}^n$, we call X an $(n, H_\infty(X))$-source, and we say X has entropy rate $H_\infty(X)/n$.

An extensively studied model of randomness extractors is the so called *seeded extractors*, introduced by Nisan and Zuckerman [60]. The inputs to a seeded extractor are a general weak random source and a short independent uniform random seed. The random seed is necessary here since it is well known that no deterministic extractor with one general weak source as input can exist. Such extractors have wide applications in computer science.

▶ **Definition 2** (Seeded Extractor)**.** *A function* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k, ε)-extractor *if for every source X with min-entropy k and independent Y which is uniform on $\{0,1\}^d$,*

$$|\mathsf{Ext}(X, Y) - U_m| \leq \epsilon.$$

If in addition we have $|(\mathsf{Ext}(X, Y), Y) - (U_m, Y)| \leq \epsilon$ then we say it is a strong (k, ε)-extractor.

Through a long line of research, we now have explicit constructions of seeded extractors with almost optimal parameters (e.g., [55, 41, 33, 32]). In the last decade or so, the focus has shifted to several different but related models of randomness extractors, including seedless extractors and non-malleable extractors. The study of these topics has also been quite fruitful, leading to breakthroughs to several long standing open problems.

1.1 Seedless extractors

As the name suggests, a seedless extractor uses no uniform seed, and the only inputs are weak random sources. Here, again we have two different cases. In the first case, one puts additional restrictions on a single weak random source in order to allow possible extraction, thus obtaining deterministic extractors for special classes of (structured) sources. In the second case, the sources are still general weak random sources, but the extractor needs to use more than one sources. To make extraction possible, one typically assumes the input sources to the extractor are independent, and this kind of extractors are sometimes called independent source extractors.

Since the pioneering work of Chor and Goldreich [19], the study of independent source extractors has gained significant attention due to their close connections to explicit Ramsey graphs, and their applications in distributed computing and cryptography with general weak random sources [43, 42]. The goal here is to give explicit constructions that match the probabilistic bound: an extractor for just two independent (n, k) sources with $k \geq \log n + O(1)$ that outputs $\Omega(k)$ bits with exponentially small (in k) error. Note that an explicit two-source extractor for such entropy (even with one bit output and constant error) will give an (strongly) explicit Ramsey graph on N vertices with no clique or independent set of size $O(\log N)$, solving an open problem proposed by Erdős [37] in his seminal paper that inaugurated the probabilistic method.

While early progress on this problem has been quite slow, with the best known construction in almost 20 years only able to handle two independent (n, k) sources with $k > n/2$ [19], since 2004 there has been a long line of work [4, 5, 62, 10, 61, 6, 46, 48, 50, 49, 52, 20, 16, 53, 26, 13, 21, 8, 24, 54] introducing exciting new techniques to this problem. This line of work greatly improved the situation and led to a series of breakthroughs. Now we have three source extractors for entropy $k \geq \mathsf{polylog}(n)$ that output $\Omega(k)$ bits with exponentially small error [52, 7], two-source extractors for entropy $k \geq \mathsf{polylog}(n)$ that output $\Omega(k)$ bits with polynomially small error [16, 53, 57], and two-source extractors for entropy $k \geq O(\log n \log\log n)$ that output one bit with any constant error [54]. This also gives an explicit Ramsey graph on N vertices with no clique or independent set of size $(\log N)^{O(\log\log\log N)}$. Interestingly and somewhat surprisingly, the most recent progress which brought the entropy requirement close to optimal, has mainly benefited from the study of another kind of extractors, the so called *non-malleable extractors*, which we now describe below.

1.2 Non-malleable extractors

Non-malleable extractors are strengthening of standard extractors, where one requires that the output is close to uniform even given the output of the extractor on tampered inputs.

▶ **Definition 3** (Tampering Funtion). *For any function $f : S \to S$, We say f has no fixed points if $f(s) \neq s$ for all $s \in S$. For any $n > 0$, let $\mathcal{F}_n$ denote the set of all functions $f : \{0,1\}^n \to \{0,1\}^n$. Any subset of $\mathcal{F}_n$ is a family of tampering functions.*

Depending on what the tampering function acts on, we also have different models of non-malleable extractors. If the tampering acts on the seed of a seeded extractor, such extractors are called *seeded non-malleable extractors*, originally introduced by Dodis and Wichs [30].

▶ **Definition 4.** *A function* $\mathrm{snmExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a seeded non-malleable extractor for min-entropy k and error ϵ if the following holds: If X is an (n, k) source and $\mathcal{A} : \{0,1\}^d \to \{0,1\}^d$ is an arbitrary tampering function with no fixed points, then*

$$|\mathrm{snmExt}(X, U_d) \circ \mathrm{snmExt}(X, \mathcal{A}(U_d)) \circ U_d - U_m \circ \mathrm{snmExt}(X, \mathcal{A}(U_d)) \circ U_d| < \epsilon$$

where U_m is independent of U_d and X.

If the tampering acts on the sources of an independent source extractor, then we have *seedless non-malleable extractors*, originally introduced by Cheraghchi and Guruswami [18].

▶ **Definition 5.** *A function* $\mathsf{nmExt} : (\{0,1\}^n)^C \to \{0,1\}^m$ *is a* (k,ϵ)*-seedless non-malleable extractor for* C *independent sources, if it satisfies the following property: Let* $X_1, \cdots, X_C$ *be* C *independent* (n,k) *sources, and* $f_1, \cdots, f_C : \{0,1\}^n \to \{0,1\}^n$ *be* C *arbitrary tampering functions such that there exists an* f_i *with no fixed points,*[2] *then*

$$|\mathsf{nmExt}(X_1, \cdots, X_C) \circ \mathsf{nmExt}(f_1(X_1), \cdots, f_C(X_2)) - U_m \circ \mathsf{nmExt}(f_1(X_1), \cdots, f_C(X_2))| < \epsilon.$$

Seeded non-malleable extractors and privacy amplification

Seeded non-malleable extractors were introduced by Dodis and Wichs [30], to study the basic problem of *privacy amplification* [9]. Consider the situation where two parties with local (non-shared) uniform random bits try to convert a shared secret weak random source $\mathbf{X}$ into shared secret uniform random bits. They do this by communicating through a channel, which is watched by an adversary with unlimited computational power. Standard strong seeded extractors provide very efficient protocols for a passive adversary (i.e., can only see the messages but cannot change them), but fail for an active adversary (i.e., can arbitrarily change, delete and reorder messages). In the latter case, which is the focus of this paper, the main goal is to design a protocol that uses as few number of interactions as possible, and achieves a shared uniform random string $\mathbf{R}$ which has *entropy loss* (the difference between the length of the output and $H_\infty(\mathbf{X})$) as small as possible. Such a protocol is defined with a security parameter s, which means the probability that an active adversary can successfully make the two parties output two different strings without being detected is at most 2^{-s}. On the other hand, if the adversary remains passive, then the two parties should achieve a shared secret string that is 2^{-s}-close to uniform. We refer the reader to [29] for a formal definition.

A long line of work has been devoted to this problem [56, 27, 30, 63, 44, 11, 29, 25, 47, 48, 51, 12, 22, 23, 13, 21, 24, 54]. It is known that one round protocol can only exist when the entropy rate of $\mathbf{X}$ is bigger than 1/2, and the protocol has to incur a large entropy loss. When the entropy rate of $\mathbf{X}$ is smaller than 1/2, [30] showed that any protocol has to take at least two rounds with entropy loss at least $\Omega(s)$. Achieving a two-round protocol with entropy loss $O(s)$ for all possible security parameters s is thus the holy grail of this problem (note that s can be at most $\Omega(k)$ where $k = H_\infty(\mathbf{X})$).

While early works on this problem used various techniques, in [30], Dodis and Wichs introduced a major tool, the seeded non-malleable extractor defined above. They showed that two-round privacy amplification protocols with optimal entropy loss can be constructed using explicit seeded non-malleable extractors. Furthermore, non-malleable extractors exist when $k > 2m + 2\log(1/\varepsilon) + \log d + 6$ and $d > \log(n-k+1) + 2\log(1/\varepsilon) + 5$. Since then, the study of non-malleable extractors has seen significant progress starting from the first explicit construction in [29], with further connections to independent source extractors established in [48, 50, 16]. Previous to this work, the best known seeded non-malleable extractor is due to the author [54], which works for entropy $k \geq O(\log n + \log(1/\epsilon)\log\log(1/\epsilon))$ and has seed length $d = O(\log n + \log(1/\epsilon)\log\log(1/\epsilon))$. Although close to optimal, the extra $O(\log\log(1/\epsilon))$ factor in the entropy requirement implies that by using this extractor, one can only get two-round privacy amplification protocols with optimal entropy loss for security parameter up to $s = \Omega(k/\log k)$. This still falls short of achieving the holy grail, and may be problematic for some applications. For example, even if the shared weak source has slightly super-logarithmic entropy, the error of the protocol can still be sub-polynomially large; while

[2] The original definition of seedless non-malleable independent source extractors in [18] allows fixed points, but the two definitions are equivalent up to a small loss in parameters. See Section 7 for details.

ideally one can hope to get negligible error, which is important for other cryptographic applications based on this. The only previous protocol that can achieve security parameter up to $s = \Omega(k)$ is the work of [11], which has entropy loss $O(\log n + s)$ but also uses $O(\log n + s)$ rounds of interactions, much larger than 2. This also results in a total communication complexity of $O((\log n + s)^2)$ and requires the two parties' local random bits to be at least this long.

Seedless non-malleable extractors and non-malleable codes

Seedless non-malleable extractors were first introduced by Cheraghchi and Guruswami [18] to study non-malleable codes [36], a generalization of standard error correcting codes to handle a much larger class of attacks. Informally, a non-malleable code is defined w.r.t. a specific family of tampering functions $\mathcal{F}$. The code consists of a randomized encoding function E and a deterministic decoding function D, such that for any $f \in \mathcal{F}$, if a codeword $E(x)$ is modified into $f(E(x))$, then the decoded message $x' = D(f(E(x)))$ is either the original message x or a completely unrelated message. The formal definition is given in Section 7. In [36], Dziembowski et. al showed that such codes can be used generally in tamper-resilient cryptography to protect the memory of a device.

Even with such generalization, non-malleable codes still cannot exist if $\mathcal{F}$ is completely unrestricted. However, they do exist for many broad families of tampering functions. One of the most studied families of tampering functions is the so called *t-split-state* model. Here, a k-bit message x is encoded into a codeword with t parts $y_1, \cdots, y_t$, each of length n. An adversary can then arbitrarily tamper with each y_i independently. In this case, the rate of the code is defined as $k/(tn)$.

This model arises naturally in many applications, typically when different parts of memory are used to store different parts of $y_1, \cdots, y_t$. Such a code can also be viewed as a kind of "non-malleable secret sharing scheme". The case of $t = 2$ is the most useful and interesting setting, since $t = 1$ corresponds to the case where $\mathcal{F}$ is unrestricted. Again, there has been a lot of previous work on non-malleable codes in this model. In this paper we will focus on the information theoretic setting.

Dziembowski et. al [36] first proved the existence of non-malleable codes in the split-state model. Cheraghchi and Guruswami [17] showed that the optimal rate of such codes in the 2-split-state model is 1/2. Since then a major goal is to construct explicit non-malleable codes in the 2-split-state model with constant rate. The first construction appears in [34], with later improvements in [3, 2, 1], but all constructions only achieve rate $n^{-\Omega(1)}$.

Cheraghchi and Guruswami [18] found a way to construct non-malleable codes in the t-split state model using non-malleable t-source extractors. Chattopadhyay and Zuckerman [15] constructed the first seedless non-malleable extractor, which works for 10 independent sources with entropy $(1-\gamma)n$, and consequently they obtained a constant rate non-malleable code in the 10-split-state model. Subsequently, constructions of non-malleable two source extractors appeared in [12] and [54]. Both constructions work for min-entropy $k = (1-\gamma)n$, and the former gives a non-malleable code in the 2-split state model with rate $n^{-\Omega(1)}$ while the latter achieves rate $\Omega(\frac{1}{\log n})$. Very recently, a work by Kanukurthi et. al [45] achieved constant rate in the 4-split state model, and another one by Gupta et. al [39] achieved constant rate in the 3-split state model, but the best construction in the 2-split state model still only achieves rate $\Omega(\frac{1}{\log n})$ [54].

As can be seen from the above discussions, extensive past research has established strong connections among these different topics, and provided solutions close to optimal. However, there still remains a gap and the motivation to close this gap remains strong.

We also remark about the curious coincidences of some parameters here and the parameters in the literature of constructing unconditional pseudorandom generators for small space computation, another central line of research in the area of pseudorandomness. The holy grail in this case is to construct a logspace explicit pseudorandom generator (a function that stretches a short uniform random seed into a long string that looks random) which fools logspace computation with seed length $O(\log(n/\epsilon))$ and error ϵ. Such a construction would imply that randomness does not add any more power in logspace computation. Although for general logspace computation the best known pseudorandom generator remains Nisan's generator [59] which needs seed length $O(\log n \log(n/\epsilon))$, recently there have been several works achieving near optimal seed length for restricted cases. For example, Meka et al. [58] constructed a pseudorandom generator for width-3 branching programs with seed length $O(\log(n/\epsilon)\mathsf{poly}\log\log(n/\epsilon)) + O(\log n \log(1/\epsilon))$, and Doron et al. [31] constructed a pseudorandom generator for constant depth read once formulas with seed length $O(\log(n/\epsilon)\mathsf{poly}\log\log(n/\epsilon))$, improving a previous similar result for depth 2 read-once formulas by Gopalan et al. [38]. The extra $\mathsf{poly}\log\log(1/\epsilon)$ dependence on error ϵ is quite similar to the previously best known non-malleable extractors, and in the case of constant or polynomially small error the seed length becomes $\log n\mathsf{poly}\log\log n$, which is again quite similar to the entropy requirement of the previously best known two-source extractors.

The high level reason for these coincidences is that all constructions use some recursive steps (e.g., for $O(\log\log n)$ steps), where in each step one needs to use say $O(\log n)$ *independent* random bits, and thus the total entropy requirement becomes at least $\Omega(\log n \log\log n)$. Circumventing this barrier needs new techniques and can lead to improved or potentially optimal constructions, which is of great interests. In this sense, the results in this paper are the first kind to break this barrier, and we believe that the set of new techniques we introduce can lead to further improvements and potentially optimal constructions.

1.3 Our Results

In this paper we achieve improvements in all the questions discussed in the context of extractors, and in several cases partially optimal constructions. In contrast, all previous works only obtain close to optimal constructions. Our first theorem gives explicit seeded non-malleable extractors with optimal entropy requirement.

▶ **Theorem 6.** *There exists a constant $C > 1$ such that for any constant $a \in \mathcal{N}, \dashv \geq \in$, any $n, k \in \mathcal{N}$ and any $0 < \epsilon < 1$ with $k \geq C(\log\log n + a\log(1/\epsilon))$, there is an explicit construction of a strong seeded (k, ϵ) non-malleable extractor $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n) + \log(1/\epsilon)2^{O(a(\log\log(1/\epsilon))^{\frac{1}{a}})}$ and $m = \Omega(k)$.*

Note that this theorem provides a trade-off between the entropy requirement and the seed length. For example, if we take $a = 2$, then the entropy requirement is $O(\log\log n + \log(1/\epsilon))$ while the seed length is $O(\log n) + 2^{O(\sqrt{\log\log(1/\epsilon)})}\log(1/\epsilon) = O(\log n) + \log^{1+o(1)}(1/\epsilon)$. By a recent result of Gur and Shinkar [40], the entropy requirement in our construction is asymptotically optimal. We can also achieve smaller seed length while requiring slightly larger entropy.

▶ **Theorem 7.** *There exists a constant $C > 1$ such that for any $n, k \in \mathcal{N}$ and $0 < \epsilon < 1$ with $k \geq C(\log\log n + \log(1/\epsilon)\log\log\log(1/\epsilon))$, there is an explicit construction of a strong seeded (k, ϵ) non-malleable extractor $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n + \log(1/\epsilon)(\log\log(1/\epsilon))^2)$[3] and $m = \Omega(k)$.*

[3] The exponent 2 can be reduced to be arbitrarily close to $\log 3$.

Combined with the protocol in [30], we have the following theorem.

▶ **Theorem 8.** *There exists a constant $0 < \alpha < 1$ such that for any $n, k \in \mathcal{N}$, there is an explicit two-round privacy amplification protocol in the presence of an active adversary that achieves (1) any security parameter $s \leq \alpha k$, entropy loss $O(\log\log n + s)$ and communication complexity $O(\log n) + s2^{O(a(\log s)^{\frac{1}{a}})}$ for any constant integer $a \geq 2$, or (2) any security parameter $s \leq \alpha k/\log\log k$, entropy loss $O(\log\log n + s)$ and communication complexity $O(\log n + s\log^2 s)$.*

Our two-round protocol can achieve optimal entropy loss for security parameter up to $s = \Omega(k)$, thus achieving the holy grail of this problem. Compared to the $O(\log n + s)$-round protocol in [11], our protocol also has better dependence on n and significantly better communication complexity.

We remark that the $O(\log\log n)$ term is also the best possible (up to constant) if one wants to apply the two-round protocol in [30]. This is because the output of the non-malleable extractor is used as the key for a message authentication code (MAC) that authenticates the seed of a strong seeded extractor with security parameter s. Since the seed of the extractor uses at least $\Omega(\log n)$ bits, the MAC requires a key of length at least $\log\log n + s$. See [30] for more details.

▶ Remark 9. In Theorem 6 and Theorem 7, the dependence on error ϵ in the seed length and the entropy requirement can be switched. For example, in Theorem 6, we can also achieve $k \geq C\log\log n + \log(1/\epsilon)2^{C\cdot a(\log\log(1/\epsilon))^{\frac{1}{a}}}$ and $d = O(\log n + a\log(1/\epsilon))$, i.e., we can achieve asymptotically optimal parameters in either the seed length or the entropy requirement, but not in both.

We also have the following non-malleable two-source extractor and seeded non-malleable extractor.

▶ **Theorem 10.** *There exists a constant $0 < \gamma < 1$ and a non-malleable two-source extractor for $(n, (1-\gamma)n)$ sources with error $2^{-\Omega(n\log\log n/\log n)}$ and output length $\Omega(n)$.*

▶ **Theorem 11.** *There is a constant $C > 0$ such that for any $\epsilon > 0$ and $n, k \in \mathcal{N}$ with $k \geq C(\log\log n + \frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)})$, there is an explicit strong seeded non-malleable extractor for (n, k) sources with seed length $d = O(\log n + \frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)})$, error ϵ and output length $\Omega(k)$.*

Combined with the techniques in [8], we obtain the following theorems.

▶ **Theorem 12.** *For every constant $\epsilon > 0$, there exists a constant $C > 1$ and an explicit two source extractor $\mathsf{Ext} : (\{0,1\}^n)^2 \to \{0,1\}$ for entropy $k \geq C\frac{\log n\log\log n}{\log\log\log n}$ with error ϵ.*

▶ **Corollary 13.** *For every large enough integer N there exists a (strongly) explicit construction of a K-Ramsey graph on N vertices with $K = (\log N)^{O(\frac{\log\log\log N}{\log\log\log\log N})}$.*

Our result gives the first two-source extractor for entropy $o(\log n\log\log n)$ and the first explicit K-Ramsey graph on N vertices with $K = (\log N)^{o(\log\log\log N)}$. For non-malleable codes in the 2-split state model, we have the following theorem.

▶ **Theorem 14.** *There are constants $0 < \eta, \mu < 1$ such that for any $n \in \mathcal{N}$ and $2^{-\frac{\mu n}{\log n}} \leq \epsilon \leq \eta$ there exists an explicit non-malleable code in the 2-split-state model with block length $2n$, rate $\Omega(\frac{\log\log\log(1/\epsilon)}{\log\log(1/\epsilon)})$ and error ϵ.*

Note that if we choose $\epsilon = 2^{-c}$ for some constant $c > 1$, then we get a non-malleable code with rate $\Omega(\frac{\log\log c}{\log c})$ and error 2^{-c}. This gives the first construction of an explicit non-malleable code in the 2-split-state model with *constant* rate. Note that the error can be arbitrarily small, and the dependence of the rate on the error is pretty good. For example, even if one wants to achieve error $2^{-2^{100}}$, which is more than enough for any practical application, the rate is on the order of 1/16. On the other hand, if we choose $\epsilon = 2^{-\mathsf{polylog}(n)}$, then we get a non-malleable code with negligible error and rate $\Omega(\frac{\log\log\log n}{\log\log n})$, which already improves the rate in [54] exponentially.

We can also achieve close to exponentially small error with an improved rate.

▶ **Theorem 15.** *For any $n \in \mathcal{N}$ there exists a non-malleable code with efficient encoder-/decoder in the 2-split-state model with block length $2n$, rate $\Omega(\log\log n/\log n)$ and error $\epsilon = 2^{-\Omega(n\log\log n/\log n)}$.*

1.4 Overview of The Constructions and Techniques

We demonstrate our techniques here by an informal overview of our constructions. Throughout this section we will be mainly interested in the dependence of various parameters (e.g., seed length, entropy requirement) on the error ϵ, since this makes the presentation cleaner. The dependence on n comes from the alternating extraction between the seed and the source, thus the seed needs to have an $O(\log n)$ term while the source only needs an $O(\log\log n)$ term.

All recent constructions of non-malleable extractors essentially follow the same high level sketch: first obtain a small advice on $L = O(\log(1/\epsilon))$ bits such that with probability $1-\epsilon$, the advice is different from its tampered version. Then, a correlation breaker with advice (informally introduced in [12] and formally defined in [22]) is used to obtain the final output. A correlation breaker with advice is a function $\mathsf{AdvCB} : X \times Y \times \alpha \to V$ where X, Y are two independent sources (in the case of a seeded non-malleable extractor, Y can be viewed as the seed), such that if the advice α is not equal to its tampered version α', then the output $\mathsf{AdvCB}(X, Y, \alpha)$ is close to uniform conditioned on the tampered version $\mathsf{AdvCB}(X', Y', \alpha')$. There are several constructions of correlation breakers, and the goal is to minimize the entropy requirement of X and Y. Previously, the most efficient construction is based on a non-malleable independence preserving merger (NIPM for short, introduced in [26] and generalized in [13]) in [54], which achieves entropy requirement $O(\log L\log(1/\epsilon))$ using a recursive structure. As discussed before, improving this needs new techniques, and our main new idea is the following:

Idea 1: *Instead of using fresh randomness, we borrow techniques from pseudorandom generators for small space computation [59, 60] to* recycle *the randomness in each step of the independence preserving merger. In this sense, we construct* pseudorandom *independence preserving mergers.*

Using this idea, we can improve the entropy requirement of X and Y in two different settings. In the asymmetric case, one of them can be optimal while the other is larger, e.g., one can be $O(\log(1/\epsilon))$ and the other is $2^{O(\sqrt{\log L})}\log(1/\epsilon)$. This is good for seeded non-malleable extractors and privacy amplification protocols. In the symmetric case which is needed for two-source extractors, one can reduce the entropy requirement of both X and Y to $O(\frac{\log L}{\log\log L}\log(1/\epsilon))$.

Turning to achieving constant rate non-malleable codes, our main new idea is the following:

Idea 2: *In the connection between non-malleable codes and non-malleable two-source extractors found by Cheraghchi and Guruswami [18], the way of dealing with errors is* too coarse*. For most recent constructions of non-malleable two-source extractors, one can actually separate two different kinds of error, where one kind is relevant and the other is* irrelevant*.*

We now discuss both ideas in more details. We start with Idea 1 and first briefly recall the construction of the merger in [54]. The NIPM takes an $L \times m$ random matrix V with $m = O(\log(1/\epsilon))$ with the property that one row in the matrix is uniform given the corresponding row in its tampered version[4] (this can be obtained from the advice and inputs), and does the following. Suppose the matrix V is a deterministic function of the source X, then we first generate $\ell = \log L$ random variables $(Y_1, \cdots, Y_\ell)$ from Y, such that each Y_i is close to uniform given the previous random variables and their tampered versions (i.e., $(Y_1, Y_1', \cdots, Y_{i-1}, Y_{i-1}')$). We call this property the *look-ahead* property. Next, we run a simple merger for ℓ iterations, with each iteration using a new Y_i to merge every two consecutive rows in V, thus decreasing the number of rows by a factor of 2. We output the final matrix V which has one row.

Let's turn to the entropy requirement. In this construction each Y_i needs to have at least $\Omega(\log(1/\epsilon))$ bits in order to ensure the error is at most ϵ, thus it is clear that Y needs to have entropy at least $\Omega(\ell \log(1/\epsilon)) = \Omega(\log(1/\epsilon) \log\log(1/\epsilon))$. However, it turns out that X also needs to have such entropy, for the following two reasons. First, in each iteration after we apply the simple merger, the length of each row in the matrix decreases by a constant factor (due to the entropy loss of any seeded extractor). Thus we cannot afford to just repeat the process for ℓ times since that would require the original row in V (and hence X) to have entropy at least $\mathsf{polylog}(1/\epsilon)$. Instead, we again create ℓ random variables $(X_1, \cdots, X_\ell)$ from X with the look-ahead property, and in each iteration after merging we use each row of the matrix to extract from a new X_i (using a standard seeded extractor, and possibly after first extracting from another new Y_i), to restore the length of the rows in the matrix. We need the look-ahead property in $(X_1, \cdots, X_\ell)$ and $(Y_1, \cdots, Y_\ell)$ so that after each iteration we can fix the previously used random variables and maintain the independence of X and Y, as well as the fact that the matrix is a deterministic function of X. Each X_i again needs at least $\Omega(\log(1/\epsilon))$ bits so this puts a lower bound on the entropy of X.

Second, in order to prepare the random variables $(Y_1, \cdots, Y_\ell)$, we in fact run an alternating extraction protocol between (part of) X and Y. This protocol lasts 2ℓ rounds between X and Y, and in each round either X or Y needs to spend $\Omega(\log(1/\epsilon))$ random bits. This again puts a lower bound of $\Omega(\ell \log(1/\epsilon))$ on the entropy of X.

We remark that the above description is slightly different from the standard definition of an NIPM, where the only input besides the matrix V is Y. Indeed, in [54] it was presented as a correlation breaker. However, these two objects are actually similar, and for this paper it is more convenient to consider NIPMs with an additional input X, which is independent of Y but may be correlated with V. We will use this notion here and formally define it in Section 4.

Improved merger construction

To break the above barriers, our key observation is that we can *recycle* the entropy in X, similar in sprit to what has been done in previous constructions of pseudorandom generators for small space computation [59, 60]. Indeed, the random variables $(X_1, \cdots, X_\ell)$ can be

[4] Sometimes we also require the other rows to be uniform, in order to make the construction simpler. This is the case of this paper, but we ignore the issue here for simplicity and clarity.

replaced by the original source X, as long as we have slightly more (e.g., 2ℓ) Y_i's and they satisfy the look ahead property. To achieve this we crucially use the property that the NIPM only needs one row of V to be uniform given the corresponding row in its tampered version, and does not care about the dependence among the rows of V (they can have arbitrary dependence). Consider a particular iteration i in which we have just finished applying the simple merger. We can first fix all random variables $\{Y_j\}$ that have been used so far, and conditioned on this fixing we know that X and Y are still independent, and the matrix V is a deterministic function of X, which is independent of all random variables obtained from Y. To restore the length of each row in V, we use each row of V to first extract $O(\log(1/\epsilon))$ bits from Y_{j+1}, and then extract back from the original source X. Note that we only need to consider each row separately (since we don't care about the dependence among them). Assume row h in V has the property that V_h is uniform given V'_h (the tampered version). Since each random variable only has $O(\log(1/\epsilon))$ bits, as long as the entropy of X is $c\log(1/\epsilon)$ for a large enough constant $c > 1$, we can argue that conditioned on the fixing of (V_h, V'_h), X still has entropy at least some $O(\log(1/\epsilon))$. On the other hand since V_h is uniform given V'_h, their corresponding outputs after extracting from (Y_{j+1}, Y'_{j+1}) will also preserve this independence; and conditioned on the fixing of (V_h, V'_h), these outputs are deterministic functions of (Y, Y'), which are independent of (X, X'). Thus they can be used to extract back from (X, X') and preserve the independence. By standard properties of a strong seeded extractor, this holds even conditioned on the fixing of (Y_{j+1}, Y'_{j+1}). Note that conditioned on the further fixing of (Y_{j+1}, Y'_{j+1}), the new matrix is again a deterministic function of X, thus we can go into the next iteration. Therefore, by recycling the entropy in X, altogether we only need X to have entropy some $O(\log(1/\epsilon))$. In each iteration we use two new Y_i's so we need roughly 2ℓ such random variables.

However, we still need to address the second problem, where we need to generate the random variables $(Y_1, \cdots, Y_{2\ell})$. The old way to generate them by using an alternating extraction protocol requires entropy roughly $O(\ell\log(1/\epsilon))$ from X. To solve this problem, we develop a new approach that requires much less entropy from X. For simplicity assume that Y is uniform, we first take 2ℓ slices Y^i from Y, where Y^i has size $(2^i - 1)d$ for some $d = O(\log(1/\epsilon))$. This ensures that even conditioned on the fixing of $(Y^1, Y'^1, \cdots, Y^{i-1}, Y'^{i-1})$, the (average) conditional min-entropy of Y_i is at least $(2^i - 1)d - 2 \cdot (2^{i-1} - 1)d = d$. Then, we can take $O(\log(1/\epsilon))$ uniform bits obtained from X, and use the *same* bits to extract Y_i from Y^i for every i. As long as we use a strong seeded extractor here, we are guaranteed that $(Y_1, \cdots, Y_{2\ell})$ satisfy the look-ahead property; and moreover conditioned on the fixing of the $O(\log(1/\epsilon))$ bits from X, we have that $(Y_1, \cdots, Y_{2\ell})$ is a deterministic function of Y. Note here again we only require entropy $O(\log(1/\epsilon))$ from X, and together with the approach described above this gives us a non-malleable extractor where X can have entropy $O(\log(1/\epsilon))$. However Y will need to have entropy at least $2^{2\ell}O(\log(1/\epsilon)) = O(\log^3(1/\epsilon))$.

To improve the entropy requirement of Y, we note that in the above approach, we only used part of X once to help obtaining the $\{Y^i\}$. Thus we have to use larger and larger slices of Y which actually waste some entropy. Instead, we can use several parts of X, each with $O(\log(1/\epsilon))$ uniform bits. For example, suppose that we have obtained X^1 and X^2, where each is uniform on some $O(\log(1/\epsilon))$ bits and X^2 is uniform even conditioned on the fixing of (X^1, X'^1). We can now take some t slices $\{Y^i\}$ of Y, each of length $(2^i - 1) \cdot 2d$ for some parameters t, d. We first use X^1 to extract from each Y^i and obtain d uniform bits. Note that conditioned on the fixing of (X^1, X'^1), these t random variables already satisfy the look-ahead property. Now for each of these d bits obtained from Y^i, we can apply the same process, i.e., we take some t slices of these d bits, each of length $(2^i - 1) \cdot O(\log(1/\epsilon))$ and then use

X^2 to extract from each of them. This way we obtain t^2 random variables $\{Y_i\}$ that satisfy the look-ahead property. We can thus choose $t^2 = 2\ell$ which means $t = O(\sqrt{\ell})$. The entropy requirement of Y is roughly $(2^t - 1) \cdot (2^t - 1)O(\log(1/\epsilon)) = O(2^{2t} \log(1/\epsilon)) = 2^{O(\sqrt{\ell})} \log(1/\epsilon)$, while the entropy requirement for X is $2 \cdot O(\log(1/\epsilon)) + O(\log(1/\epsilon)) = O(\log(1/\epsilon))$. This significantly improves the entropy requirement of Y.

We can repeat the previous process and use some a parts $(X^1, \cdots, X^a)$ obtained from X. As long as a is a constant, X only needs entropy $O(a \log(1/\epsilon)) = O(\log(1/\epsilon))$, while the entropy requirement of Y is reduced to $2^{O(a\ell^{\frac{1}{a}})} \log(1/\epsilon) = 2^{O(a \log\log(1/\epsilon)^{\frac{1}{a}})} \log(1/\epsilon)$. To prepare the a parts of X, we perform an initial alternating extraction between X and Y, which only needs entropy $O(a \log(1/\epsilon))$ from either of them. This gives Theorem 6. In the extreme case, we can try to minimize the entropy requirement of Y by first creating $\log \ell + 1 = \log\log\log(1/\epsilon) + O(1)$ X^i's, and in each step using a new X^i to double the number of Y_i's. This can be done by using the same X^i to do an alternating extraction of two rounds with each Y_i in parallel. Thus after $\log \ell + 1$ steps we obtain $(Y_1, \cdots, Y_{2\ell})$. Now X needs to have entropy $O(\log(1/\epsilon) \log\log\log(1/\epsilon))$. Ideally, we would want to claim that Y needs entropy $O(\log(1/\epsilon) \log\log(1/\epsilon))$, but due to technical reasons we can only show that this works as long as Y has entropy $O(\log(1/\epsilon)(\log\log(1/\epsilon))^2)$.

The balanced case

In the above discussion, the entropy requirement for X and Y is unbalanced, in the sense that one of them can be quite small, while the other is relatively large. For applications to two-source extractors and non-malleable codes, we need a balanced entropy requirement. Upon first look it does not seem that our new techniques can achieve any improvement in this case, since we are still merging two rows of the matrix V in each step, and for this merging we need at least $\Omega(\log(1/\epsilon))$ fresh random bits. Note that we need $\ell = \log L = \log\log(1/\epsilon)$ steps to finish the merging, thus it seems the total entropy requirement is at least $\Omega(\log(1/\epsilon) \log\log(1/\epsilon))$.

Our key observation here is that we can again apply the idea of recycling entropy. Specifically, let us choose a parameter $t \in \mathcal{N}$ and we merge every t rows in the matrix V at each step, using some merger that we have developed above. For example, we can choose the merger which for merging t rows, requires X to have entropy $O(\log(1/\epsilon))$ and Y to have entropy $2^{O(\sqrt{\log t})} \log(1/\epsilon)$. This will take us $\frac{\log L}{\log t}$ steps to finish merging, and we will do it in the following way. First, we create $s = O(\frac{\log L}{\log t})$ random variables $X_1, \cdots, X_s$ that satisfy the look-ahead property. Then, in each step of the merging, we will use a new X_j. The X_j's can be prepared by taking a small slice of both X and Y and do an alternating extraction protocol with $O(s)$ rounds, which consumes entropy $O(s \log(1/\epsilon)) = O(\frac{\log L}{\log t} \log(1/\epsilon))$ from both X and Y. However, in each step of the merging, we will *not* use fresh entropy from Y, but will recycle the entropy in Y. Note that by doing this, we are recycling the entropy in both X and Y. The recycling in X is done within each step of applying the small merger, while the recycling in Y is done between these steps.

Now, consider a particular step i in the merging. Since we are using a new X_j in each step, we can fix all previous X_j's that have been used and their tampered versions. Conditioned on this fixing, the matrix V obtained so far (and the tampered version V') is a deterministic function of Y, therefore independent of X. We now want to claim that conditioned on the random variable (V, V'), Y still has high entropy. If this is true then we can take a new X_{j+1} and apply a strong seeded extractor to Y using X_{j+1} as the seed, and the extracted random bits (which are deterministic functions of Y conditioned on the fixing of X_{j+1}) can be used for merging in the next step. Also note that to apply the merger, we can take yet another

new X_{j+2} and use each row of V to extract from X_{j+2} and create a matrix W. Conditioned on the fixing of (V, V'), we have that (W, W') is a deterministic function of (X, X') and therefore independent of (Y, Y'). Moreover the independence between corresponding rows in (V, V') is preserved in (W, W') (i.e., there is also a row in W that is uniform given the corresponding row in W'). Thus now we can indeed apply the merger again to W and the extracted random bits from Y, possibly together with a new X_{j+3}. Again, this is similar in spirit to what has been done in previous constructions of pseudorandom generators for small space computation [59, 60].

The above idea indeed works, except for the following subtle point: in the first several steps of merging, the matrix V can have many rows and the size of V can be larger than the entropy of Y, unless Y has entropy $\Omega(\log^2(1/\epsilon))$. Thus conditioning on (V, V') may cause Y to lose all entropy. To get around this, we again use the fact that we only need one row in V to be independent of the corresponding row in V' (call this the good row), and does not care about the dependence between different rows. Thus in each step, we only need to condition on the fixing of the t rows that we are merging (and their tampered versions). This ensures that if originally there is a good row in these t rows, then after merging the output is also a good row in the new matrix. Thus, we only need the entropy of Y to be $O(t\log(1/\epsilon)) + 2^{O(\sqrt{\log t})}\log(1/\epsilon) + O(\frac{\log L}{\log t}\log(1/\epsilon)) = O(t\log(1/\epsilon) + \frac{\log L}{\log t}\log(1/\epsilon))$ since we will maintain the length of each row in V to be $O(\log(1/\epsilon))$. Now by choosing $t = \frac{\log L}{\log\log L}$, both X and Y only need entropy $O(\frac{\log L}{\log\log L}\log(1/\epsilon)) = O(\frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)})$. By the connections in [54, 8, 18], this dependence gives Theorem 10, 11, 12 and 15.

Non-malleable codes

To further improve the rate of non-malleable codes in the 2-split state model, we re-examine the connection between non-malleable codes and non-malleable two-source extractors found by Cheraghchi and Guruswami [18]. They showed that given a non-malleable two-source extractor with error ϵ and output length m, the uniform sampling of the pre-image of any given output gives an encoding of a non-malleable code in the 2-split state model with error roughly $2^m\epsilon$. This blow up of error comes from the conditioning on the event that the output of the extractor is a given string in $\{0,1\}^m$, which roughly has probability 2^{-m}. Therefore, one needs $m < \log(1/\epsilon)$, and thus the error of the extractor puts a limit on the rate of the code.

To break this barrier, we note that all recent constructions of non-malleable two-source extractors [12, 54] follow a very special framework. As mentioned before, these constructions first obtain an advice $\tilde{\alpha}$ such that with probability $1-\epsilon_1$ we have $\tilde{\alpha} \neq \tilde{\alpha}'$, where $\tilde{\alpha}'$ is the tampered version. Then, using a correlation breaker with advice one obtains the output. This part has error ϵ_2, and the final error of the extractor is $\epsilon_1 + \epsilon_2$.

In all previous work, this error is treated as a whole, but our key observation here is that these two errors ϵ_1 and ϵ_2 can actually be treated *separately*. More specifically, the error that matters most for the rate of the code is actually ϵ_2, not ϵ_1. Intuitively, this is because the event $\tilde{\alpha} \neq \tilde{\alpha}'$ is determined by a set of random variables that have small size compared to the length of X and Y. Thus even conditioned on the fixing of these random variables, X and Y still have plenty of entropy, which implies that the output of the extractor is still ϵ_2-close to uniform. Thus, as long as ϵ_2 is small, the output of the extractor is roughly independent of the event $\tilde{\alpha} \neq \tilde{\alpha}'$. Therefore, conditioned on any given output of the extractor, the event $\tilde{\alpha} \neq \tilde{\alpha}'$ still happens with probability roughly $1-\epsilon_1$ and we won't be paying a price of $2^m\epsilon_1$ here. Once this event happens, the correlation breaker ensures that the extractor is non-malleable with error ϵ_2, and we can use a similar argument as in [18] to get a non-malleable code with error roughly $2^m\epsilon_2$. Thus the total error of the non-malleable code is roughly $\epsilon_1 + 2^m\epsilon_2$. Now, we just need $m < \log(1/\epsilon_2)$.

We can now play with the two parameters ϵ_1, ϵ_2. The advice length L is $\Omega(\log(1/\epsilon_1))$ and we need to supply entropy $O(\frac{\log L}{\log\log L}\log(1/\epsilon_2))$ by using our improved correlation breaker. If we can achieve $L = \Theta(\log(1/\epsilon_1))$ then one can see that if we choose ϵ_1 to be any constant, then we can set $\epsilon_2 = 2^{-\Omega(n)}$ and also $m = \Omega(n)$, thus we get a constant rate non-malleable code. If we set $\epsilon_1 = 2^{-\mathsf{polylog}(n)}$ then we can set $\epsilon_2 = 2^{-\Omega(\frac{n\log\log\log n}{\log\log n})}$ and thus we get rate $\Omega(\frac{\log\log\log n}{\log\log n})$.

A technical issue here is how to achieve $L = \Theta(\log(1/\epsilon_1))$ for any ϵ_1. In [12, 54], the advice is obtained by using some random seed R to sample from an asymptotically good encoding of X, Y, and concatenating the sampled symbols with R. This puts a lower bound of $\log n$ on L, since we need at least this number of bits to sample from a string of length n. However this is not good enough to achieve constant rate. Our idea around this is to use repeated sampling. To illustrate the idea, suppose for example that we have obtained an advice V such that $V \neq V'$ with probability $1 - 1/\mathsf{poly}(n)$ and V has length $O(\log n)$. We now use another piece of independent random bits R_1 of length $O(\log\log n)$ to sample $O(\log\log n)$ bits from an asymptotically good encoding of V, and obtain a new advice V_1 by concatenating R_1 with the sample bits. This ensures that $V_1 \neq V_1'$ happens with probability $1 - 1/\mathsf{polylog}(n)$ conditioned on $V \neq V'$, and the length of V_1 is now $O(\log\log n)$. We repeat this process until we get the desired error ϵ_1 (e.g., a constant) and the advice length is now $L = \Theta(\log(1/\epsilon_1))$. Note that the total error is still $O(\epsilon_1)$, the total number of random bits needed is small, and the process terminates in roughly $\log^* n$ steps. To prepare the independent random bits used in repeated sampling, we first take a small slice of X and Y and do an alternating extraction with roughly $\log^* n$ steps, which guarantees the bits used for sampling in later steps are independent of the previous ones and their tampered versions. Finally, some extra work are needed here to take care of the issue of fixed points, which is more subtle than [18] since now we are treating the two errors ϵ_1 and ϵ_2 separately.

Organization. The rest of the paper is organized as follows. We give some preliminaries in Section 2, and define alternating extraction in Section 3. We present independence preserving mergers in Section 4, correlation breakers in Section 5, non-malleable extractors in Section 6, and non-malleable codes in Section 7. Finally we conclude with some open problems in Section 8.

2 Preliminaries

We often use capital letters for random variables and corresponding small letters for their instantiations. Let $|S|$ denote the cardinality of the set S. For ℓ a positive integer, U_ℓ denotes the uniform distribution on $\{0,1\}^\ell$. When used as a component in a vector, each U_ℓ is assumed independent of the other components. When we have adversarial tampering, we use letters with prime to denote the tampered version of random variables. All logarithms are to the base 2.

2.1 Probability Distributions

▶ **Definition 16** (statistical distance). *Let W and Z be two distributions on a set S. Their* statistical distance *(variation distance) is*

$$\Delta(W,Z) =: \max_{T\subseteq S}(|W(T) - Z(T)|) = \frac{1}{2}\sum_{s\in S}|W(s) - Z(s)|.$$

We say W is ε-close to Z, denoted $W \approx_\varepsilon Z$, if $\Delta(W, Z) \leq \varepsilon$. For a distribution D on a set S and a function $h : S \to T$, let $h(D)$ denote the distribution on T induced by choosing x according to D and outputting $h(x)$.

▶ **Lemma 17.** *For any function α and two random variables A, B, we have $\Delta(\alpha(A), \alpha(B)) \leq \Delta(A, B)$.*

2.2 Average Conditional Min Entropy

▶ **Definition 18.** *The* average conditional min-entropy *is defined as*

$$\begin{aligned}\widetilde{H}_\infty(X|W) &= -\log\left(\mathrm{E}_{w\leftarrow W}\left[\max_x \Pr[X = x|W = w]\right]\right)\\ &= -\log\left(\mathrm{E}_{w\leftarrow W}\left[2^{-H_\infty(X|W=w)}\right]\right).\end{aligned}$$

▶ **Lemma 19** ([28])**.** *For any $s > 0$, $\Pr_{w\leftarrow W}[H_\infty(X|W = w) \geq \widetilde{H}_\infty(X|W) - s] \geq 1 - 2^{-s}$.*

▶ **Lemma 20** ([28])**.** *If a random variable B has at most 2^ℓ possible values, then $\widetilde{H}_\infty(A|B) \geq H_\infty(A) - \ell$.*

2.3 Prerequisites from Previous Work

Sometimes it is convenient to talk about average case seeded extractors, where the source X has average conditional min-entropy $\widetilde{H}_\infty(X|Z) \geq k$ and the output of the extractor should be uniform given Z as well. The following lemma is proved in [28].

▶ **Lemma 21** ([28])**.** *For any $\delta > 0$, if Ext is a (k, ϵ) extractor then it is also a $(k+\log(1/\delta), \epsilon+\delta)$ average case extractor.*

For a strong seeded extractor with optimal parameters, we use the following extractor constructed in [41].

▶ **Theorem 22** ([41])**.** *For every constant $\alpha > 0$, there exists a constant $\beta > 0$ such that for all positive integers n, k and any $\epsilon > 2^{-\beta k}$, there is an explicit construction of a strong (k, ϵ)-extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n + \log(1/\epsilon))$ and $m \geq (1 - \alpha)k$. The same statement also holds for a strong average case extractor.*

▶ **Theorem 23** ([19])**.** *For every $0 < m < n$ there is an explicit two-source extractor $\mathsf{IP} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ based on the inner product function, such that if X, Y are two independent (n, k_1) and (n, k_2) sources respectively, then*

$$(\mathsf{IP}(X, Y), X) \approx_\epsilon (U_m, X) \text{ and } (\mathsf{IP}(X, Y), Y) \approx_\epsilon (U_m, Y),$$

where $\epsilon = 2^{-\frac{k_1+k_2-n-m-1}{2}}$.

The following standard lemma about conditional min-entropy is implicit in [60] and explicit in [56].

▶ **Lemma 24** ([56])**.** *Let X and Y be random variables and let $\mathcal{Y}$ denote the range of Y. Then for all $\epsilon > 0$, one has*

$$\Pr_Y\left[H_\infty(X|Y = y) \geq H_\infty(X) - \log|\mathcal{Y}| - \log\left(\frac{1}{\epsilon}\right)\right] \geq 1 - \epsilon.$$

We also need the following lemma.

▶ **Lemma 25.** *[49] Let (X, Y) be a joint distribution such that X has range $\mathcal{X}$ and Y has range $\mathcal{Y}$. Assume that there is another random variable X' with the same range as X such that $|X - X'| = \epsilon$. Then there exists a joint distribution (X', Y) such that $|(X, Y) - (X', Y)| = \epsilon$.*

3 Alternating Extraction

Our constructions use the following alternating extraction protocol as a key ingredient. Alternating extraction was first introduced in [35], and has now become an important tool in constructions related to extractors.

Quentin: Q, S_1		Wendy: X
S_1	$\xrightarrow{S_1}$	
	$\xleftarrow{R_1}$	$R_1 = \mathsf{Ext}_w(X, S_1)$
$S_2 = \mathsf{Ext}_q(Q, R_1)$	$\xrightarrow{S_2}$	
	$\xleftarrow{R_2}$	$R_2 = \mathsf{Ext}_w(X, S_2)$
	$\cdots$	
$S_\ell = \mathsf{Ext}_q(Q, R_{\ell-1})$	$\xrightarrow{S_\ell}$	
		$R_\ell = \mathsf{Ext}_w(X, S_\ell)$

Figure 1 Alternating Extraction.

▶ **Definition 26** (Alternating Extraction). *Assume that we have two parties, Quentin and Wendy. Quentin has a source Q, Wendy has a source W. Also assume that Quentin has a uniform random seed S_1 (which may be correlated with Q). Suppose that (Q, S_1) is kept secret from Wendy and W is kept secret from Quentin. Let Ext_q, Ext_w be strong seeded extractors with optimal parameters, such as that in Theorem 22. Let r, s be two integer parameters for the protocol. For some integer parameter $\ell > 0$, the* alternating extraction protocol *is an interactive process between Quentin and Wendy that runs in ℓ steps.*

In the first step, Quentin sends S_1 to Wendy, Wendy computes $R_1 = \mathsf{Ext}_w(W, S_1)$. She sends R_1 to Quentin and Quentin computes $S_2 = \mathsf{Ext}_q(Q, R_1)$. In this step R_1, S_2 each outputs r and s bits respectively. In each subsequent step i, Quentin sends S_i to Wendy, Wendy computes $R_i = \mathsf{Ext}_w(W, S_i)$. She replies R_i to Quentin and Quentin computes $S_{i+1} = \mathsf{Ext}_q(Q, R_i)$. In step i, R_i, S_{i+1} each outputs r and s bits respectively. Therefore, this process produces the following sequence:

$$S_1, R_1 = \mathsf{Ext}_w(W, S_1), S_2 = \mathsf{Ext}_q(Q, R_1), \cdots,$$
$$S_\ell = \mathsf{Ext}_q(Q, R_{\ell-1}), R_\ell = \mathsf{Ext}_w(W, S_\ell).$$

The output of an alternating extraction protocol is often described as a *look-ahead extractor*, defined as follows. Let $Y = (Q, S_1)$ be a seed, the look-ahead extractor is defined as

$$\mathsf{laExt}(W, Y) = \mathsf{laExt}(W, (Q, S_1)) =: R_1, \cdots, R_\ell.$$

The following lemma is a special case of Lemma 6.5 in [12].

▶ **Lemma 27.** *Let W be an (n_w, k_w)-source and W' be a random variable on $\{0,1\}^{n_w}$ that is arbitrarily correlated with W. Let $Y = (Q, S_1)$ such that Q is a (n_q, k_q)-source, S_1 is a uniform string on s bits, and $Y' = (Q', S_1')$ be a random variable arbitrarily correlated with Y, where Q' and S_1' are random variables on n_q bits and s bits respectively. Let $\mathsf{Ext}_q, \mathsf{Ext}_w$ be strong seeded extractors that extract s and r bits from sources with min-entropy k with error ϵ and seed length $d \leq min\{r, s\}$. Suppose (Y, Y') is independent of (W, W'), $k_q \geqslant k + 2(\ell - 1)s + 2\log(\frac{1}{\epsilon})$,*

and $k_w \geqslant k + 2(\ell-1)r + 2\log(\frac{1}{\epsilon})$*. Let* laExt *be the look-ahead extractor defined above using* $\mathsf{Ext}_q, \mathsf{Ext}_w$*, and* $(R_1, \cdots, R_\ell) = \mathsf{laExt}(W, Y)$*,* $(R'_1, \cdots, R'_\ell) = \mathsf{laExt}(W', Y')$*. Then for any* $0 \leq j \leq \ell - 1$*, we have*

$$\begin{aligned}&(Y, Y', \{R_1, R'_1, \cdots, R_j, R'_j\}, R_{j+1})\\ \approx_{\epsilon_1} &(Y, Y', \{R_1, R'_1, \cdots, R_j, R'_j\}, U_r),\end{aligned}$$

where $\epsilon_1 = O(\ell\epsilon)$*.*

4 Non-Malleable Independence Preserving Merger

We now describe the notion of *non-malleable independence preserving merger*, introduced in [13] based on the notion of independence preserving merger introduced in [26].

▶ **Definition 28.** *A* (L, d', ε)-NIPM : $\{0,1\}^{Lm} \times \{0,1\}^d \to \{0,1\}^{m_1}$ *satisfies the following property. Suppose*

- $\mathbf{X}, \mathbf{X}'$ *are random variables, each supported on boolean* $L \times m$ *matrices s.t for any* $i \in [L]$*,* $\mathbf{X}_i = U_m$*,*
- $\{\mathbf{Y}, \mathbf{Y}'\}$ *is independent of* $\{\mathbf{X}, \mathbf{X}'\}$*, s.t* $\mathbf{Y}, \mathbf{Y}'$ *are each supported on* $\{0,1\}^d$ *and* $H_\infty(\mathbf{Y}) \geqslant d'$*,*
- *there exists an* $h \in [L]$ *such that* $(\mathbf{X}_h, \mathbf{X}'_h) = (U_m, \mathbf{X}'_h)$*,*

then

$$\begin{aligned}&|(L, d', \varepsilon)\text{-NIPM}(\mathbf{X}, \mathbf{Y}), (L, d', \varepsilon)\text{-NIPM}(\mathbf{X}', \mathbf{Y}')\\ &- U_{m_1}, (L, d', \varepsilon)\text{-NIPM}(\mathbf{X}', \mathbf{Y}')| \leqslant \epsilon.\end{aligned}$$

We have the following construction and theorem.

***L*-Alternating Extraction.** We extend the previous alternating extraction protocol by letting Quentin have access to L sources $Q_1, \ldots, Q_L$ (instead of just Q) which have the same length. Now in the i'th round of the protocol, he uses Q_i to produce the r.v $S_i = \mathsf{Ext}_q(Q_i, R_i)$. More formally, the following sequence of r.v's is generated: $S_1, R_1 = \mathsf{Ext}_w(W, S_1), S_2 = \mathsf{Ext}_q(Q_2, R_1), \ldots, R_{L-1} = \mathsf{Ext}_w(W, S_{\ell-1}), S_L = \mathsf{Ext}_q(Q_L, R_{L-1})$.

The NIPM is now constructed as follows. Let S_1 be a slice of $\mathbf{X}_1$ with length $O(\log(d/\varepsilon))$, then run the L-alternating extraction described above with $(Q_1, \ldots, Q_L) = (\mathbf{X}_1, \ldots, \mathbf{X}_L)$ and $W = \mathbf{Y}$. Finally output S_L.

▶ **Theorem 29** ([13])**.** *There exists a constant* $c > 0$ *such that for all integers* $m, d, d', L > 0$ *and any* $\epsilon > 0$*, with* $m \geqslant 4cL\log(d/\epsilon)$*,* $d' \geqslant 4cL\log(m/\epsilon)$*, the above construction* NIPM : $(\{0,1\}^m)^\ell \times \{0,1\}^d \to \{0,1\}^{m_1}$ *has output length* $m_1 \geq 0.2m$*, such that if the following conditions hold:*

- $\mathbf{X}, \mathbf{X}'$ *are random variables, each supported on boolean* $L \times m$ *matrices s.t for any* $i \in [L]$*,* $\mathbf{X}_i = U_m$*,*
- $\{\mathbf{Y}, \mathbf{Y}'\}$ *is independent of* $\{\mathbf{X}, \mathbf{X}'\}$*, s.t* $\mathbf{Y}, \mathbf{Y}'$ *are each supported on* $\{0,1\}^d$ *and* $H_\infty(\mathbf{Y}) \geqslant d'$*,*
- *there exists an* $h \in [L]$ *such that* $(\mathbf{X}_h, \mathbf{X}'_h) = (U_m, \mathbf{X}'_h)$*,*

then

$$|\text{NIPM}(\mathbf{X}, \mathbf{Y}), \text{NIPM}(\mathbf{X}', \mathbf{Y}'), \mathbf{Y}, \mathbf{Y}' - U_{m_1}, \text{NIPM}(\mathbf{X}', \mathbf{Y}'), \mathbf{Y}, \mathbf{Y}'| \leqslant L\epsilon.$$

It is sometimes more convenient to consider NIPMs which use an additional source X in the computation. We generalize the above definition as follows.

▶ **Definition 30.** *A* (L, d, d', ε)-NIPM : $\{0,1\}^{Lm} \times \{0,1\}^d \times \{0,1\}^{d'} \to \{0,1\}^{m_1}$ *satisfies the following property. Suppose*

- V, V' *are random variables, each supported on boolean* $L \times m$ *matrices s.t for any* $i \in [L]$, $V_i = U_m$,
- *there exists an* $h \in [L]$ *such that* $(V_h, V'_h) = (U_m, V'_h)$,
- $\mathbf{X}, \mathbf{X}'$ *are random variables, each supported on* d *bits, such that* $\mathbf{X}$ *is uniform conditioned on* (V, V'),
- $(\mathbf{Y}, \mathbf{Y}')$ *is independent of* $(V, V', \mathbf{X}, \mathbf{X}')$, *s.t* $\mathbf{Y}, \mathbf{Y}'$ *are each supported on* $\{0,1\}^{d'}$ *and* $\mathbf{Y}$ *is uniform,*

If the function is an NIPM *that is strong in* $\mathbf{Y}$ *then*

$$|(L, d, d', \varepsilon)\text{-NIPM}(V, \mathbf{X}, \mathbf{Y}), (L, d, d', \varepsilon)\text{-NIPM}(V', \mathbf{X}', \mathbf{Y}'), \mathbf{Y}, \mathbf{Y}' - U_{m_1}, (L, d, d', \varepsilon)\text{-NIPM}(V', \mathbf{X}', \mathbf{Y}'), \mathbf{Y}, \mathbf{Y}'| \leqslant \epsilon.$$

If the function is an NIPM *that is strong in* $\mathbf{X}$ *then*

$$|(L, d, d', \varepsilon)\text{-NIPM}(V, \mathbf{X}, \mathbf{Y}), (L, d, d', \varepsilon)\text{-NIPM}(V', \mathbf{X}', \mathbf{Y}'), \mathbf{X}, \mathbf{X}' - U_{m_1}, (L, d, d', \varepsilon)\text{-NIPM}(V', \mathbf{X}', \mathbf{Y}'), \mathbf{X}, \mathbf{X}'| \leqslant \epsilon.$$

We will now use the above construction to give another NIPM, which recycles the entropy. Specifically, we have the following construction.

▶ **Construction 31.** *Asymmetric* NIPM.

Inputs:

- $L, m, n, d \in \mathcal{N}$ *and an error parameter* $\epsilon > 0$ *such that* $m \geq c\log(d/\epsilon)$ *and* $d \geq c\log(n/\epsilon)$ *for some constant* $c > 1$.
- *A random variable* V *supported on a boolean* $L \times m$ *matrix.*
- *An* $(n, 6m)$ *source* $\mathbf{X}$.
- *Random variables* $\mathbf{Y}_1, \cdots, \mathbf{Y}_\ell$ *where* $\ell = \log L$ *and each* $\mathbf{Y}_i$ *is supported on* $\{0,1\}^d$.

Output: a random variable $\mathbf{W} \in \{0,1\}^m$.

Let $V^0 = V$. *For* $i = 1$ *to* $\log L$ *do the following.*

1. *Take a slice* $\mathbf{Y}_i^1$ *of* $\mathbf{Y}_i$ *with length* $d/3$. *Merge every two rows of* V^{i-1}, *using* $\mathbf{Y}_i^1$ *and the* NIPM *from Theorem 29. That is, for every* $j \leq t/2$ *where* t *is the current number of rows in* V^{i-1} *(initially* $t = L$*), compute* $\overline{V_j^{i-1}} = \text{NIPM}((V_{2j-1}^{i-1}, V_{2j}^{i-1}), \mathbf{Y}_i^1)$.
2. *For every* $j \leq t/2$, *compute* $\overline{\mathbf{Y}_{ij}} = \mathsf{Ext}_1(\mathbf{Y}_i, \overline{V_j^{i-1}})$, *where* Ext_1 *is the extractor in Theorem 22 and output* $d/4$ *bits.*
3. *For every* $i \leq t/2$, *compute* $\widetilde{V_j^{i-1}} = \mathsf{Ext}_2(\mathbf{X}, \overline{\mathbf{Y}_{ij}})$, *where* Ext_2 *is the extractor in Theorem 22 and output* m *bits.*
4. *Let* V^i *with the concatenation of* $\widetilde{V_j^{i-1}}, j = 1, \cdots, t/2$. *Note that the number of rows in* V^i *has decreased by a factor of* 2.

Finally output $\mathbf{W} = V^{\log L}$.

▶ **Lemma 32.** *There is a constant* $c > 1$ *such that suppose we have the following random variables:*

- V, V', *each supported on a boolean* $L \times m$ *matrix s.t for any* $i \in [L]$, $V_i = U_m$. *In addition, there exists an* $h \in [L]$ *such that* $(V_h, V'_h) = (U_m, V'_h)$.
- $\mathbf{X}, \mathbf{X}'$ *where* $\mathbf{X}$ *is an* $(n, 6m)$ *source.*

- *Random variables* $(\mathbf{Y}_1, \mathbf{Y}'_1), \cdots, (\mathbf{Y}_\ell, \mathbf{Y}'_\ell)$ *obtained from* $\mathbf{Y}, \mathbf{Y}'$ *deterministically, where* $\ell = \log L$*. These random variables satisfy the following look-ahead condition:* $\forall j < \ell$*, we have*

$$(\mathbf{Y}_j, \mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{j-1}, \mathbf{Y}'_{j-1}) = (U_d, \mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{j-1}, \mathbf{Y}'_{j-1}).$$

In addition, $(V, V', \mathbf{X}, \mathbf{X}')$ *is independent of* $(\mathbf{Y}, \mathbf{Y}')$*.*

Let $\mathbf{W}$ *be the output of the* NIPM *on* $(V, \mathbf{X}, \mathbf{Y}_1, \cdots, \mathbf{Y}_\ell)$ *and* $\mathbf{W}'$ *be the output of the* NIPM *on* $(V', \mathbf{X}', \mathbf{Y}'_1, \cdots, \mathbf{Y}'_\ell)$*. Then*

$$(\mathbf{W}, \mathbf{W}', \mathbf{Y}, \mathbf{Y}') \approx_{O(L\epsilon)} (U_m, \mathbf{W}', \mathbf{Y}, \mathbf{Y}').$$

Proof. We use induction to show the following claim.

▷ Claim 33. For every $0 \le i \le \ell = \log L$, the following holds after step i.

- V^i, V'^i are each supported on boolean $(t = L/2^i) \times m$ matrices s.t for any $j \in [t]$, $(V^i_j, \mathbf{Y}, \mathbf{Y}') \approx_{\epsilon_j} (U_m, \mathbf{Y}, \mathbf{Y}')$. In addition, there exists an $h \in [t]$ such that $(V^i_h, V'^i_h, \mathbf{Y}, \mathbf{Y}') \approx_{\epsilon_i} (U_m, V'^i_h, \mathbf{Y}, \mathbf{Y}')$. Here ϵ_i is the error after step i which satisfies that $\epsilon_0 = 0$ and $\epsilon_{i+1} \le 2\epsilon_i + 4\epsilon$.
- Conditioned on the fixing of $\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_j, \mathbf{Y}'_j$, each of V^i and V'^i is a deterministic function of $V, V', \mathbf{X}, \mathbf{X}'$.

For the base case of $i = 0$, the claim clearly holds. Now assume that the claim holds for i, we show that it holds for $i + 1$.

We first fix $\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_i, \mathbf{Y}'_i$. By the induction hypothesis, conditioned on the fixing of these random variables, each of V^i and V'^i is a deterministic function of $V, V', \mathbf{X}, \mathbf{X}'$, and thus independent of $(\mathbf{Y}_{i+1}, \mathbf{Y}'_{i+1})$. We only consider the row $h \in [t]$ such that $(V_h, V'_h) \approx_{4\cdot 2^i\epsilon} (U_m, V'_h)$, since the analysis for the rest of the rows are similar and simpler.

First we ignore the error ϵ_i. By Theorem 29, and note that we are merging every two rows at one step, we can choose a suitable constant $c > 1$ in the construction such that

$$(\overline{V^i_{h'}}, \overline{V'^i_{h'}}, \mathbf{Y}^1_{i+1}, \mathbf{Y}'^1_{i+1}) \approx_{2\epsilon} (U_{m_1}, \overline{V'^i_{h'}}, \mathbf{Y}^1_{i+1}, \mathbf{Y}'^1_{i+1}),$$

where $h' = \lceil \frac{h}{2} \rceil$ and $m_1 = 0.2m$. We now fix $(\mathbf{Y}^1_{i+1}, \mathbf{Y}'^1_{i+1})$. Note that conditioned on the fixing, $\mathbf{Y}_{i+1}$ still has average conditional min-entropy at least $d - d/3 = 2d/3$ and is independent of $(\overline{V^i_{h'}}, \overline{V'^i_{h'}})$. Now we can first fix $\overline{V'^i_{h'}}$ and then $\overline{\mathbf{Y}'_{ih'}}$. Note that conditioned on this fixing, $\overline{V^i_{h'}}$ is still (close to) uniform and the average conditional min-entropy of $\mathbf{Y}_{i+1}$ is at least $2d/3 - d/4 > d/3$. Thus as long as c is large enough, by Theorem 22 we have that

$$(\overline{\mathbf{Y}_{ih'}}, \overline{V^i_{h'}}) \approx_\epsilon (U_{d/4}, \overline{V^i_{h'}}).$$

We now further fix $\overline{V^i_{h'}}$. Note that conditioned on this fixing, $\overline{\mathbf{Y}_{ih'}}$ is still (close to) uniform. Moreover conditioned on all the random variables we have fixed, $\overline{\mathbf{Y}}_{ih'}$ is a deterministic function of $\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{i+1}, \mathbf{Y}'_{i+1}$ and thus independent of $\mathbf{X}, \mathbf{X}'$. Also conditioned on all the random variables we have fixed, the average conditional min-entropy of $\mathbf{X}$ is at least $6m - 2m_1 > 5m$.

We can now further fix $\widetilde{V'^i_{h'}}$, which is a deterministic function of $\mathbf{X}'$. Conditioned on this fixing the independence of random variables still holds, while the average conditional min-entropy of $\mathbf{X}$ is at least $5m - m = 4m$. Therefore by Theorem 22 we have that

$$(\widetilde{V^i_{h'}}, \overline{\mathbf{Y}_{ih'}}) \approx_\epsilon (U_m, \overline{\mathbf{Y}_{ih'}}).$$

Since we have already fixed $\overline{\mathbf{Y}'_{ih'}}$ and $\widetilde{V'^i_{h'}}$, and note that conditioned on this fixing, $(\mathbf{Y}, \mathbf{Y}')$ are independent of $\widetilde{V^i_{h'}}$ which is a deterministic function of $\mathbf{X}$, we also have that

$$(\widetilde{V^i_{h'}}, \widetilde{\mathbf{X}'_{h'}}, \mathbf{Y}, \mathbf{Y}') \approx_\epsilon (U_m, \widetilde{\mathbf{X}'_{h'}}, \mathbf{Y}, \mathbf{Y}').$$

Adding back all the errors we get that there exists an $h' \in [t]$ such that

$$(\widetilde{\mathbf{X}_{h'}}, \widetilde{V'^i_{h'}}, \mathbf{Y}, \mathbf{Y}') \approx_{\epsilon_{i+1}} (U_m, \widetilde{V'^i_{h'}}, \mathbf{Y}, \mathbf{Y}'),$$

where $\epsilon_{i+1} \leq 2\epsilon_i + 4\epsilon$. Furthermore, it is clear that conditioned on the fixing of $\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{i+1}, \mathbf{Y}'_{i+1}$, each of V^{i+1} and V'^{i+1} is a deterministic function of $V, V', \mathbf{X}, \mathbf{X}'$.

We can now estimate the final error to be $\epsilon_\ell \leq 4(\sum_{i=1}^{\ell} 2^i \epsilon) = O(L\epsilon)$. Finally, when the number or rows in V^i decreases to 1 after step ℓ, the output $\mathbf{W} = V^{\log L}$ satisfies the conclusion of the lemma. ◀

We will now construct another NIPM. First we need the following lemma.

▶ **Lemma 34.** *For any constant $a \in \mathcal{N}$, any $\ell, s \in \mathcal{N}$ and any $\epsilon > 0$ there exists an explicit function $\mathsf{Conv}_a : \{0,1\}^n \times \{0,1\}^{a \cdot d} \to \{0,1\}^{\ell \cdot s}$ with $d = O(\log(n/\epsilon))$ and $n = 2^{O(a \cdot \ell^{\frac{1}{a}})} \cdot s$ such that the following holds. Let (Y, Y') be two random variables each on n bits, and Y is uniform. Let $(X = (X_1, \cdots, X_a), X' = (X'_1, \cdots, X'_a))$ be random variables each on $a \cdot d$ bits, where each X_i and X'_i is on d bits. Further assume that (X, X') satisfies the following look-ahead property: $\forall i \in [a]$, we have*

$$(X_i, X_1, X'_1, \cdots, X_{i-1}, X'_{i-1}) = (U_d, X_1, X'_1, \cdots, X_{i-1}, X'_{i-1}).$$

Let $(W_1, \cdots, W_\ell) = \mathsf{Conv}_a(Y, X)$ and $(W'_1, \cdots, W'_\ell) = \mathsf{Conv}_a(Y', X')$. Then we have

$$(X, X', W_1, W'_1, \cdots, W_\ell, W'_\ell) \approx_{O(\ell\epsilon)} (X, X', U_s, W'_1, \cdots, U_s, W'_\ell),$$

where each U_s is independent of previous random variables but may depend on later random variables.

Proof. We will prove the lemma by induction on a. For the base case $a = 1$, consider the following construction. For $j = 1, \cdots, \ell$, let Y_j be a slice of Y with length $(2^j - 1) \cdot 2s$ (this is possible since the total entropy required is at most $2^\ell \cdot 2s$), and compute $W_j = \mathsf{Ext}(Y_j, X_1)$. Note that for any $j \in [\ell]$, conditioned on the fixing of $Y_1, Y'_1, \cdots, Y_{j-1}, Y'_{j-1}$, the average conditional min-entropy of Y_j is at least $(2^j - 1) \cdot 2s - 2(2^{j-1} - 1) \cdot 2s = 2s$. Thus by Theorem 22 we have that

$$(W_j, Y_1, Y'_1, \cdots, Y_{j-1}, Y'_{j-1}, X, X') \approx_\epsilon (U_s, Y_1, Y'_1, \cdots, Y_{j-1}, Y'_{j-1}, X, X').$$

Since $(W_1, W'_1, \cdots, W_{j-1}, W'_{j-1})$ is a deterministic function of $(Y_1, Y'_1, \cdots, Y_{j-1}, Y'_{j-1})$ and (X, X'), we also have that

$$(W_j, W_1, W'_1, \cdots, W_{j-1}, W'_{j-1}, X, X') \approx_\epsilon (U_s, W_1, W'_1, \cdots, W_{j-1}, W'_{j-1}, W, W').$$

By adding all the errors the statement of the lemma holds.

Now assume that the lemma holds for a, we will construct another function Conv_{a+1} for the case of $a + 1$. First choose a parameter $t \in \mathcal{N}$ to be decided later. For $j = 1, \cdots, \ell/t$, let Y_j be a slice of Y with length $(2^j - 1) \cdot 2m$, where m is the length of Y (i.e., n) for Conv_a when choosing $\ell = t$. Thus we have $m = 2^{O(a \cdot t^{\frac{1}{a}})} \cdot s$. Now, for every j we first

use X_1 to compute $\hat{W}_j = \mathsf{Ext}(Y_j, X_1)$ and output m bits, then compute $(\hat{W}_{1j}, \cdots, \hat{W}_{tj}) = \mathsf{Conv}_a(\hat{W}_j, X_2, \cdots, X_{a+1})$. The final outputs are obtained by combining all the $\{\hat{W}_{ij}\}$ in sequence.

Note that by the same argument as above, we have that

$$(X_1, X_1', \hat{W}_1, \hat{W'}_1, \cdots, \hat{W}_{\ell/t}, \hat{W'}_{\ell/t}) \approx_{O(\frac{\ell}{t}\epsilon)} (X_1, X_1', U_m, \hat{W'}_1, \cdots, U_m, \hat{W'}_{\ell/t}).$$

Now we can fix (X_1, X_1'). Note that conditioned on the fixing, $(\hat{W}_1, \hat{W'}_1, \cdots, \hat{W}_{\ell/t}, \hat{W'}_{\ell/t})$ is a deterministic function of (Y, Y'), thus independent of (X, X'). Now we can used the induction hypothesis to conclude that the statement holds for the case of $a+1$. Note that the total error is $O(\frac{\ell}{t}\epsilon) + \ell/t \cdot O(t\epsilon) = O(\ell\epsilon)$ since the part of $O(\frac{\ell}{t}\epsilon)$ decreases as a geometric sequence. Finally, the entropy requirement of Y is $(2^{\ell/t} - 1) \cdot 2m = (2^{\ell/t} - 1) \cdot 2 \cdot 2^{O(a \cdot t^{\frac{1}{a}})} \cdot s = 2^{l/t + O(a \cdot t^{\frac{1}{a}}) + 1} \cdot s$.

We now just need to choose a t to minimize this quantity. We can choose $t = \ell^{\frac{a}{a+1}}$ so that the entropy requirement of Y is $2^{O((a+1) \cdot \ell^{\frac{1}{a+1}})} \cdot s$. ◀

We now have the following construction.

▶ **Construction 35.** NIPM_x *(which is strong in Y) or* NIPM_y *(which is strong in X).*
Inputs:

- *An error parameter $\epsilon > 0$ and a constant $a \in \mathcal{N}$.*
- *A random variable V supported on a boolean $L \times m$ matrix.*
- *A uniform string X on d_1 bits.*
- *A uniform string Y on d_2 bits.*
- *Let $d = c\log(max\{d_1, d_2\}/\epsilon)$ for some constant $c > 1$.*

Output: NIPM_x *outputs a random variable $\mathbf{W}_x \in \{0,1\}^m$, and* NIPM_y *outputs $\mathbf{W}_y \in \{0,1\}^d$.*

1. *Let $\ell = \log L$.*[5] *Let X_0 be a slice of X with length $4a \cdot d$, and Y_0 be a slice of Y with length $4a \cdot d$. Use X_0 and Y_0 to run an alternating extraction protocol, and output $(R_0, \cdots, R_a) = \mathsf{IaExt}(X_0, Y_0)$ where each R_i has d bits.*
2. *Compute $Z = \mathsf{Ext}(Y, R_0)$ and output $d_2/2$ bits, where Ext is the strong seeded extractor from Theorem 22.*
3. *For every $i \in [L]$, compute $\overline{V_i} = \mathsf{Ext}(Y_0, V_i)$ and output d bits. Then, compute $\hat{V}_i = \mathsf{Ext}(X, \overline{V_i})$ and output m bits.*
4. *Compute $(Z_1, \cdots, Z_\ell) = \mathsf{Conv}_a(Z, R_1, \cdots, R_a)$ where each Z_i has d bits.*
5. NIPM_x *outputs $\mathbf{W}_x = \mathrm{NIPM}(\hat{V}, Z_1, \cdots, Z_\ell)$, where* NIPM *is the merger in Construction 31 and Lemma 32.* NIPM_y *outputs $\mathbf{W}_y = \mathsf{Ext}(Y, \mathbf{W}_x)$ with d bits.*

We now have the following lemma.

▶ **Lemma 36.** *There exist a constant $c > 1$ such that for any $\epsilon > 0$ and any $L, m, d_1, d_2, n \in \mathcal{N}$ such that $d \geq c(\log\max\{d_1, d_2\} + \log(1/\epsilon))$, $m \geq d$, $d_1 \geq 8a \cdot d + 6m$ and $d_2 \geq 8a \cdot d + c^{a \cdot \log^{\frac{1}{a}} L} \cdot d$, the above construction gives an $(L, d_1, d_2, O(L\epsilon))$-NIPM that is either strong in X or strong in Y.*

[5] Without loss of generality we assume that L is a power of 2. Otherwise add 0 to the string until the length is a power of 2.

Proof. Note that Y_0 has min-entropy $4ad \geq 4d$, thus by Theorem 22 we have that for every $i \in [L]$,

$$(\overline{V_i}, V_i) \approx_\epsilon (U_d, V_i),$$

and there exists an $h \in [L]$ such that

$$(\overline{V_h}, \overline{V'_h}, V_h, V'_h) \approx_\epsilon (U_d, \overline{V'_h}, V_h, V'_h).$$

Note that conditioned on the fixing of (V, V'), we have that (X, X') and (Y, Y') are still independent, and furthermore $(\overline{V}, \overline{V'})$ is a deterministic function of (Y, Y'). Note that conditioned on the fixing of (X_0, X'_0), the average conditional min-entropy of X is at least $8a \cdot d + 6m - 2 \cdot 4a \cdot d = 6m$. Thus again by Theorem 22 we have that for every $i \in [L]$,

$$(\hat{V}_i, \overline{V_i}) \approx_\epsilon (U_d, \overline{V_i}),$$

and there exists an $h \in [L]$ such that

$$(\hat{V}_h, \hat{V'}_h, \overline{V_h}, \overline{V'_h}) \approx_\epsilon (U_d, \hat{V'}_h, \overline{V_h}, \overline{V'_h}).$$

Note that now conditioned on the fixing of $(\overline{V_h}, \overline{V'_h})$, we have that (X, X') and (Y, Y') are still independent, and furthermore $(\hat{V}_h, \hat{V'}_h)$ is a deterministic function of (X, X'). Thus we basically have that conditioned on the fixing of (X_0, X'_0, Y_0, Y'_0), $(\hat{V}, \hat{V'})$ is a deterministic function of (X, X') and they satisfy the property needed by an NIPM.

Now, by Lemma 27, we have that

$$(Y_0, Y'_0, R_0, R'_0, \cdots, R_a, R'_a) \approx_{O(a^2\epsilon)} (Y_0, Y'_0, U_d, R'_0, \cdots, U_d, R'_a).$$

Note that conditioned on the fixing of (Y_0, Y'_0), we have that (X, X') and (Y, Y') are still independent, and furthermore $(R_0, R'_0, \cdots, R_a, R'_a)$ is a deterministic function of (X, X'). Also the average conditional min-entropy of Y is at least $d_2 - 2 \cdot 4a \cdot d = c^{a \cdot \log^{\frac{1}{a}} L} \cdot d > 3d_2/4$ for a large enough constant c. Thus by Theorem 22 we have that

$$(Z, R_0) \approx_\epsilon (U_{d_2/2}, R_0).$$

We can now fix (R_0, R_0). Note that now (Z_0, Z'_0) is a deterministic function of (Y, Y'), and $d_2/2 > \frac{1}{2} c^{a \cdot \log^{\frac{1}{a}} L} \cdot d$. Note that now $(R_1, R'_1, \cdots, R_a, R'_a)$ still satisfies the look-ahead property. Thus as long as c is large enough, by Lemma 34 we have that

$$(Z_1, Z'_1, \cdots, Z_\ell, Z'_\ell, X_0, X'_0) \approx_{O(\ell\epsilon)} (U_d, W'_1, \cdots, U_d, W'_\ell, X_0, X'_0).$$

We can now fix (X_0, X'_0), and note that conditioned on this fixing $(Z_1, Z'_1, \cdots, Z_\ell, Z'_\ell)$ is a deterministic function of (Y, Y'). In summary, conditioned on the fixing of (X_0, X'_0, Y_0, Y'_0), we have that $(\hat{V}, \hat{V'})$and $(Z_1, Z'_1, \cdots, Z_\ell, Z'_\ell)$ satisfy the conditions required by Lemma 32. Therefore we can now apply that lemma to finish the proof. The total error is at most $O(L\epsilon) + O(a^2\epsilon) + O(\epsilon) + O(\ell\epsilon) = O(L\epsilon)$. ◀

The extreme case of the above construction gives the following NIPM.

▶ **Construction 37.** NIPM_x *(which is strong in Y) or* NIPM_y *(which is strong in X).*
Inputs:

- *An error parameter $\epsilon > 0$.*
- *A random variable V supported on a boolean $L \times m$ matrix.*
- *A uniform string $\mathbf{X}$ on n bits.*
- *A uniform string $\mathbf{Y}$ on n' bits.*

Output: NIPM_x *outputs a random variable* $\mathbf{W}_x \in \{0,1\}^m$*, and* NIPM_y *outputs* $\mathbf{W}_y \in \{0,1\}^{O(\log(n/\epsilon))}$.

1. *Let* $d_1 = c\log(n'/\epsilon)$ *and* $d_2 = c\log(n/\epsilon)$*. Take a slice* $\mathbf{X}_0$ *of* $\mathbf{X}$ *with length* $10\log\log L \cdot d_1$*, and a slice* $\mathbf{Y}_0$ *of* $\mathbf{Y}$ *with length* $10\log\log L \cdot d_2$*.*
2. *Use* $\mathbf{X}_0$ *and* $\mathbf{Y}_0$ *to do an alternating extraction protocol, and output* $(R_0, R_1, \cdots, R_t) = \mathsf{IaExt}(\mathbf{X}_0, \mathbf{Y}_0)$ *where* $t = \log\log L$ *and each* R_i *has* $4d_1$ *bits, each* S_i *(used in the alternating extraction) has* d_2 *bits.*
3. *For each* $i \in [L]$*, compute* $\overline{\mathbf{Y}}_i = \mathsf{Ext}(\mathbf{Y}_0, V_i)$ *where each* $\overline{\mathbf{Y}}_i$ *outputs* d_2 *bits. Then compute* $\overline{V}_i = \mathsf{Ext}(\mathbf{X}, \overline{\mathbf{Y}}_i)$ *where each* $\overline{V}_i$ *outputs* m *bits. Here* Ext *is the strong seeded extractor from Theorem 22. Let* $\overline{V}$ *be the matrix whose* i*'th row is* $\overline{V}_i$*.*
4. *Let* $\mathbf{Y}_1^0 = \mathbf{Y}$*. For* $j = 0$ *to* $\log\log L$ *do the following. For* $h = 1$ *to* 2^j*, use* $\mathbf{Y}_h^j$ *and* R_j *to do an alternating extraction protocol, and output* $(S_{h1}^j, S_{h2}^j) = \mathsf{IaExt}(\mathbf{Y}_h^j, R_j)$*, where each* S_{hi}^j *has* $(\frac{\log^{\log a} L}{a^{j-1}} - 1)d_2$ *bits. Note that altogether we get* 2^{j+1} *outputs and relabel them as* $\mathbf{Y}_1^{j+1}, \cdots, \mathbf{Y}_{2^{j+1}}^{j+1}$*.*
5. *After the previous step, we get* $2\log L$ *outputs. Let them be* $\mathbf{Y}_1, \cdots, \mathbf{Y}_{2\log L}$*, and output* $\mathbf{W}_x = \mathrm{NIPM}(\overline{V}, \mathbf{X}, \mathbf{Y}_1, \cdots, \mathbf{Y}_{2\log L})$ *with* m *bits. Let* $\mathbf{W}_y = \mathsf{Ext}(\mathbf{Y}, \mathbf{W}_x)$ *with* d_2 *bits.*

We now have the following lemma.

▶ **Lemma 38.** *There is a constant* $c > 1$ *such that suppose we have the following random variables and conditions:*

- V, V'*, each supported on a boolean* $L \times m$ *matrix s.t for any* $i \in [L]$*,* $V_i = U_m$*. In addition, there exists an* $h \in [L]$ *such that* $(V_h, V_h') = (U_m, V_h')$*.*
- $\mathbf{Y}, \mathbf{Y}'$*, each supported on* n' *bits, where* $\mathbf{Y}$ *is uniform.*
- $\mathbf{X}, \mathbf{X}'$*, each supported on* n *bits, where* $\mathbf{X}$ *is uniform. In addition,* $\mathbf{X}$ *is independent of* (V, V')*, and* $(V, V', \mathbf{X}, \mathbf{X}')$ *is independent of* $(\mathbf{Y}, \mathbf{Y}')$*.*
- $m \geq c\log(n'/\epsilon)$*,* $n \geq 20c\log\log L\log(n'/\epsilon) + 6m$ *and* $n' \geq 20c\log^{\log a} L\log(n/\epsilon)$*.*

Let $(\mathbf{W}_x, \mathbf{W}_y)$ *be the outputs of* $(\mathrm{NIPM}_x, \mathrm{NIPM}_y)$ *on* $(V, \mathbf{X}, \mathbf{Y})$ *and* $(\mathbf{W}_x', \mathbf{W}_y')$ *be the outputs of the* $(\mathrm{NIPM}_x, \mathrm{NIPM}_y)$ *on* $(V', \mathbf{X}', \mathbf{Y}')$*. Then*

$$(\mathbf{W}_x, \mathbf{W}_x', \mathbf{Y}, \mathbf{Y}') \approx_{O(L\epsilon)} (U_m, \mathbf{W}_x', \mathbf{Y}, \mathbf{Y}')$$

and

$$(\mathbf{W}_y, \mathbf{W}_y', V, V', \mathbf{X}, \mathbf{X}') \approx_{O(L\epsilon)} (U_{O(\log(n/\epsilon))}, \mathbf{W}_y', V, V', \mathbf{X}, \mathbf{X}').$$

Proof. First, since $(V, V', \mathbf{X}, \mathbf{X}')$ is independent of $(\mathbf{Y}, \mathbf{Y}')$, as long as c is large enough, by Theorem 22 we know that for any $i \in [L]$,

$$(\overline{\mathbf{Y}}_i, V) \approx_\epsilon (U_d, V).$$

In addition, suppose for some $h \in [L]$ we have that $(V_h, V_h') = (U_m, V_h')$, then we can first fix V_h' and then $\overline{\mathbf{Y}}_h$. Conditioned on this fixing V_h is still uniform, the average conditional min-entropy of $\mathbf{Y}_0$ is at least $10\log\log L \cdot d - d > 3d$ and V_h and Y_0 are still independent, thus by Theorem 22 we have that

$$(\overline{\mathbf{Y}}_h, \overline{\mathbf{Y}}_h', V, V') \approx_\epsilon (U_d, \overline{\mathbf{Y}}_h', V, V').$$

In other words, the random variables $\{(\overline{\mathbf{Y}}_i, \overline{\mathbf{Y}}_i')\}$ inherit the properties of $\{(V_i, V_i')\}$. We now ignore the errors since this adds at most $L\epsilon$ to the final error. Now we fix (V, V'). Note that conditioned on this fixing, the random variables $(\overline{\mathbf{Y}}_i, \overline{\mathbf{Y}}_i')$ are deterministic functions of $(\mathbf{Y}_0, \mathbf{Y}_0')$, and are thus independent of $(\mathbf{X}, \mathbf{X}')$. Furthermore, we have that conditioned on this

fixing, $\mathbf{X}$ is still uniform. In addition, even conditioned on the fixing of $(\mathbf{X}_0, \mathbf{X}'_0)$, the average conditional min-entropy of $\mathbf{X}$ is at least $20c \log\log L \log(n'/\epsilon) + 6m - 2 \cdot 10 \log\log L \cdot d_1 = 6m$. Thus by the same argument before we have that for any $i \in [L]$,

$$(\overline{V}_i, \mathbf{Y}_0, \mathbf{X}_0, \mathbf{X}'_0) \approx_\epsilon (U_m, \mathbf{Y}_0, \mathbf{X}_0, \mathbf{X}'_0),$$

and that there exists an $h \in [L]$ such that

$$(\overline{V}_h, \overline{V}'_h, \mathbf{Y}_0, \mathbf{Y}'_0, \mathbf{X}_0, \mathbf{X}'_0) \approx_\epsilon (U_m, \overline{V}'_h, \mathbf{Y}_0, \mathbf{Y}'_0, \mathbf{X}_0, \mathbf{X}'_0).$$

We will again ignore the error for now since this adds at most $L\epsilon$ to the final error. Next, by Lemma 27 we have that for any $0 \le j \le t-1$,

$$(R_{j+1}, (R_1, R'_1, \cdots, R_j, R'_j), \mathbf{Y}_0, \mathbf{Y}'_0) \approx_{O(t\epsilon)} (U_{4d_1}, (R_1, R'_1, \cdots, R_j, R'_j), \mathbf{Y}_0, \mathbf{Y}'_0).$$

Thus by a hybrid argument and the triangle inequality, we have that

$$(\mathbf{Y}_0, \mathbf{Y}'_0, R_1, R'_1, \cdots, R_t, R'_t) \approx_{O(t^2\epsilon)} (\mathbf{Y}_0, \mathbf{Y}'_0, U_{4d_1}, R'_1, \cdots, U_{4d_1}, R'_t),$$

where each U_{4d_1} is independent of all the previous random variables (but may depend on later random variables). From now on, we will proceed as if each R_j is uniform given $(\mathbf{Y}_0, \mathbf{Y}'_0, \{R_1, R'_1, \cdots, R_{j-1}, R'_{j-1}\})$, since this only adds $O(t^2\epsilon)$ to the final error.

Now we can fix $(\mathbf{Y}_0, \mathbf{Y}'_0)$. Note that conditioned on this fixing, $(\overline{V}, \overline{V}', R_1, R'_1, \cdots, R_t, R'_t)$ are deterministic functions of $(V, V', \mathbf{X}, \mathbf{X}')$, and thus independent of $(\mathbf{Y}, \mathbf{Y}')$. Also note that conditioned on this fixing, the average conditional min-entropy of $\mathbf{Y}$ is at least $20 \log^{\log a} L \cdot d_2 - 2 \cdot 10 \log\log L \cdot d_2 > a^2 \log^{\log a} L \cdot d_2$. We now prove the following claim.

▷ Claim 39. Let $\overline{R_j} = (R_1, \cdots, R_j)$. Suppose that at the beginning of the j'th iteration, we have that conditioned on the fixing of $\overline{R_{j-1}}$, the following holds.

1. , $(\mathbf{X}, \mathbf{X}')$ is independent of $(\mathbf{Y}, \mathbf{Y}')$, and $(\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{2^j}, \mathbf{Y}'_{2^j})$ is a deterministic function of $(\mathbf{Y}, \mathbf{Y}')$.
2. For every $h \in [2^j]$, the average conditional min-entropy of $\mathbf{Y}_h$ given $(\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{h-1}, \mathbf{Y}'_{h-1})$ is at least $(\frac{\log^{\log a} L}{a^{j-2}} - 1)d_2$.

Then at the end of the j'th iteration, the following holds.

1. Conditioned on the fixing of $\overline{R_j}$, $(\mathbf{X}, \mathbf{X}')$ is independent of $(\mathbf{Y}, \mathbf{Y}')$, and $(\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{2^{j+1}}, \mathbf{Y}'_{2^{j+1}})$ is a deterministic function of $(\mathbf{Y}, \mathbf{Y}')$.
2. For every $h \in [2^{j+1}]$,

$$(\mathbf{Y}_h, (\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{h-1}, \mathbf{Y}'_{h-1}), \overline{R_j}) \approx_\epsilon (U_{(\frac{\log^{\log a} L}{a^{j-1}} - 1)d_2}, (\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{h-1}, \mathbf{Y}'_{h-1}), \overline{R_j}).$$

Proof of the claim. First, since the computation in the j'th iteration only involves (R_j, R'_j) and $(\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{2^j}, \mathbf{Y}'_{2^j})$, and (R_j, R'_j) is a deterministic function of $(\mathbf{X}, \mathbf{X}')$ conditioned on the fixing of the previous random variables, we know that at the end of the j'th iteration, conditioned on the fixing of $(R_1, \cdots, R_j)$ we have that $(\mathbf{X}, \mathbf{X}')$ is independent of $(\mathbf{Y}, \mathbf{Y}')$, and $(\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{2^{j+1}}, \mathbf{Y}'_{2^{j+1}})$ is a deterministic function of $(\mathbf{Y}, \mathbf{Y}')$.

Next, we use $(Z_1, Z'_1, \cdots, Z_{2^{j+1}}, Z'_{2^{j+1}})$ to represent the outputs computed from (R_j, R'_j) and $(\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{2^j}, \mathbf{Y}'_{2^j})$, and assume that $2\ell - 1 \le h \le 2\ell$ for some ℓ, then Z_h is obtained from $\mathbf{Y}_\ell$. We can now first fix $(\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{\ell-1}, \mathbf{Y}'_{\ell-1})$, and conditioned on this fixing

$\mathbf{Y}_\ell$ has average conditional min-entropy at least $(\frac{\log^{\log a} L}{a^{j-2}} - 1)d_2$. Now by Lemma 27 we have that

$$(S_1^\ell, R_j, R'_j) \approx_\epsilon (U_{(\frac{\log^{\log a} L}{a^{j-1}}-1)d_2}, R_j, R'_j)$$

and

$$(S_2^\ell, S_1^\ell, S_1'^\ell, R_j, R'_j) \approx_\epsilon (U_{(\frac{\log^{\log a} L}{a^{j-1}}-1)d_2}, S_1^\ell, S_1'^\ell, R_j, R'_j),$$

since $(\frac{\log^{\log a} L}{a^{j-2}} - 1)d_2 \geq 2 \cdot (\frac{\log^{\log a} L}{a^{j-1}} - 1)d_2 + (1+\alpha)(\frac{\log^{\log a} L}{a^{j-1}} - 1)d_2 + d_2$ and $4d_1 \geq 2d_1 + 1.1d_1 + 0.9d_1$. Thus as long as the constant c is large enough one can make sure that $min\{d_2, 0.9d_1\} \geq 2\log(1/\epsilon)$, and we can extract $(\frac{\log^{\log a} L}{a^{j-1}} - 1)d_2$ bits from entropy $(1+\alpha)(\frac{\log^{\log a} L}{a^{j-1}} - 1)d_2$ and d_1 bits from entropy $1.1d_1$. Note that $(Z_1, Z'_1, \cdots, Z_{2\ell-2}, Z'_{2\ell-2})$ are computed from $(\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{\ell-1}, \mathbf{Y}'_{\ell-1})$ and (R_j, R'_j), and $(\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{\ell-1}, \mathbf{Y}'_{\ell-1})$ are already fixed. Thus the second part of the claim also holds. ◁

Now note that at the beginning of the first iteration, the condition of the claim holds. Thus if we ignore the errors, then we can apply the claim repeatedly until the end of the iteration. At this time for each $h \in [\log L]$ we have that $\mathbf{Y}_h$ has at least $(\frac{\log^{\log a} L}{a^{\log\log L-1}} - 1)d_2 > d_2$ bits. Furthermore

$$(\mathbf{Y}_h, (\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{h-1}, \mathbf{Y}'_{h-1}), \overline{R_t}) \approx (U, (\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{h-1}, \mathbf{Y}'_{h-1}), \overline{R_t}).$$

The total error so far is $O(L\epsilon) + O(t^2\epsilon) + \sum_{j=0}^{\log\log L} 2^j \cdot 2\epsilon = O(L\epsilon)$. Note that now conditioned on all the fixed random variables $(\mathbf{X}_0, \mathbf{X}'_0, \mathbf{Y}_0, \mathbf{Y}'_0, \overline{R_t})$ (note that $\overline{R_t}$ is a deterministic function of $(\mathbf{X}_0, \mathbf{X}'_0, \mathbf{Y}_0, \mathbf{Y}'_0)$, we have that $(V, V', \mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{2\log L}, \mathbf{Y}'_{2\log L}, \mathbf{X}, \mathbf{X}')$ satisfies the conditions of the Lemma 32, since the average conditional min-entropy of X is at least $n - 20\log\log L \cdot d_1 \geq 6m$. Now we can apply Lemma 32 to show that

$$(\mathbf{W}_x, \mathbf{W}'_x, \mathbf{Y}, \mathbf{Y}') \approx (U_m, \mathbf{W}'_x, \mathbf{Y}, \mathbf{Y}'),$$

where the total error is $O(L\epsilon)+O(L\epsilon) = O(L\epsilon)$. Furthermore, note that conditioned on the fixing of $(\mathbf{Y}_1, \mathbf{Y}'_1, \cdots, \mathbf{Y}_{2\log L}, \mathbf{Y}'_{2\log L})$, we have that $(\mathbf{W}_x, \mathbf{W}'_x)$ is a deterministic function of $(V, V', \mathbf{X}, \mathbf{X}')$, and thus independent of $(\mathbf{Y}, \mathbf{Y}')$. Also note that $\mathbf{Y}$ has average conditional min-entropy at least $20c\log^{\log a} L\log(n/\epsilon) - 4\log L d_2 > 10d_2$. Thus by Theorem 22 we have that

$$(\mathbf{W}_y, \mathbf{W}'_y, \mathbf{W}_x, \mathbf{W}'_x) \approx (U_{d_2}, \mathbf{W}'_y, \mathbf{W}_x, \mathbf{W}'_x),$$

where the error is $O(L\epsilon)+O(\epsilon) = O(L\epsilon)$. Note that given $(\mathbf{W}_x, \mathbf{W}'_x)$, we have that $(\mathbf{W}_y, \mathbf{W}'_y)$ is a deterministic function of $(\mathbf{Y}, \mathbf{Y}')$. Thus we also have that

$$(\mathbf{W}_y, \mathbf{W}'_y, V, V', \mathbf{X}, \mathbf{X}') \approx_{O(L\epsilon)} (U_{d_2}, \mathbf{W}'_y, V, V', \mathbf{X}, \mathbf{X}').$$ ◀

5 Correlation Breaker with Advice

We now use our non-malleable independence preserving mergers to construct improved correlation breakers with advice. A correlation breaker uses independent randomness to break the correlations between several correlated random variables. The first correlation breaker appears implicitly in the author's work [49], and this object is strengthened and formally defined in [20]. A correlation breaker with advice additionally uses some string as an advice. This object was first introduced and used without its name in [12], and then explicitly defined in [22].

▶ **Definition 40** (Correlation breaker with advice). *A function*

$$\mathsf{AdvCB} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^L \to \{0,1\}^m$$

is called a (k, k', ε)-correlation breaker with advice if the following holds. Let Y, Y' be d-bit random variables such that $H_\infty(Y) \geq k'$. Let X, X' be n-bit random variables with $H_\infty(X) \geq k$, such that (X, X') is independent of (Y, Y'). Then, for any pair of distinct L-bit strings α, α',

$$(\mathsf{AdvCB}(X, Y, \alpha), \mathsf{AdvCB}(X', Y', \alpha')) \approx_\varepsilon (U, \mathsf{AdvCB}(X', Y', \alpha')).$$

In addition, we say that AdvCB *is strong if*

$$\begin{aligned}&(\mathsf{AdvCB}(X, Y, \alpha), \mathsf{AdvCB}(X', Y', \alpha'), Y, Y')\\ \approx_\varepsilon &(U, \mathsf{AdvCB}(X', Y', \alpha'), Y, Y').\end{aligned}$$

Our construction needs the following flip-flop extraction scheme, which was constructed by Cohen [20] using alternating extraction, based on a previous similar construction of the author [49]. The flip-flop function can be viewed as a basic correlation breaker, which (informally) uses an independent source $\mathbf{X}$ to break the correlation between two r.v's $\mathbf{Y}$ and $\mathbf{Y}'$, given an advice bit.

▶ **Theorem 41** ([20, 12]). *There exists a constant c_{41} such that for all $n > 0$ and any $\epsilon > 0$, there exists an explicit function* flip-flop $: \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, *$m = 0.4k$, satisfying the following: Let $\mathbf{X}$ be an (n, k)-source, and $\mathbf{X}'$ be a random variable on n bits arbitrarily correlated with $\mathbf{X}$. Let $\mathbf{Y}$ be an independent uniform seed on d bits, and $\mathbf{Y}'$ be a random variable on d bits arbitrarily correlated with $\mathbf{Y}$. Suppose $(\mathbf{X}, \mathbf{X}')$ is independent of $(\mathbf{Y}, \mathbf{Y}')$. If $k, d \geqslant C_{41} \log(n/\epsilon)$, then for any bit b,*

$$|\text{flip-flop}(\mathbf{X}, \mathbf{Y}, b), \mathbf{Y}, \mathbf{Y}' - U_m, \mathbf{Y}, \mathbf{Y}'| \leqslant \epsilon.$$

Furthermore, for any bits b, b' with $b \neq b'$,

$$\begin{aligned}&|\text{flip-flop}(\mathbf{X}, \mathbf{Y}, b), \text{flip-flop}(\mathbf{X}', \mathbf{Y}', b'), \mathbf{Y}, \mathbf{Y}'\\ &- U_m, \text{flip-flop}(\mathbf{X}', \mathbf{Y}', b'), \mathbf{Y}, \mathbf{Y}'| \leqslant \epsilon.\end{aligned}$$

5.1 Asymmetric correlation breaker

We will present correlation breakers that use general NIPMs. By plugging in various NIPMs this gives different correlation breakers.

▶ **Construction 42.** *Inputs:*

- *Let $\ell, m \in \mathcal{N}$ be two integers, $\epsilon > 0$ be an error parameter.*
- *X, Y, two independent sources on n bits and s bits respectively, with min-entropy at least $n - \ell$ and $s - \ell$.*
- *an advice string $\alpha \in \{0,1\}^L$.*
- *An $(L, d_1, d_2, O(L\epsilon))$-NIPM$_x$ that is strong in Y.*
- *Let* IP *be the two source extractor from Theorem 23.*

1. *Let $d' = O(\log(max\{n, s\}/\epsilon))$ be the seed length of the extractor from Theorem 22, and let $d = 8d'$. Let X^0 be a slice of X with length $d + 2\ell + 2\log(1/\epsilon)$, and Y^0 be a slice of Y with length $d + 2\ell + 2\log(1/\epsilon)$.*
2. *Compute $Z = \mathsf{IP}(X^0, Y^0)$ and output d bits.*

3. *Use X and Z to do an alternating extraction, and output two random variables $(X_0, X_1) = \mathsf{IaExt}(X, Z)$ where each X_i has $3m$ bits.*
4. *Use Y and Z to do an alternating extraction, and output two random variables $(Y_0, Y_1) = \mathsf{IaExt}(Y, Z)$ where each Y_i has $3d$ bits.*
5. *Use X_1, Y_1, α to obtain an $L \times m$ matrix V, where for any $i \in [L]$, $V_i = \text{flip-flop}(X_1, Y_1, \alpha_i)$ and outputs m bits.*
6. *Compute $\hat{X} = \mathsf{Ext}(X, Y_0)$ and output $n/2$ bits. Compute $\hat{Y} = \mathsf{Ext}(Y, X_0)$ and output $s/2$ bits. Here Ext is the strong seeded extractor from Theorem 22.*
7. *Output $\hat{V} = \text{NIPM}_x(V, \hat{X}, \hat{Y})$.*

We now have the following lemma.

▶ **Lemma 43.** *There exists a costant $c > 1$ such that the following holds. Suppose that there exists an $(L, d_1, d_2, O(L\epsilon))$-NIPM that is strong in Y which outputs m bits, then there exists an explicit $(n - \ell, s - \ell, O(L\epsilon))$ $\mathsf{AdvCB} : \{0,1\}^n \times \{0,1\}^s \times \{0,1\}^L \to \{0,1\}^m$ as long as $m \geq c \log(max\{n, s\}/\epsilon)$, $n \geq 20m + 2d_1 + 5\ell + 4\log(1/\epsilon)$ and $s \geq m + 2d_2 + 5\ell + 4\log(1/\epsilon)$.*

Proof. Throughout the proof we will use letters with prime to denote the corresponding random variables obtained from (X', Y', α'). First, notice that both X^0 and Y^0 have min-entropy at least $d + \ell + 2\log(1/\epsilon)$. Thus by Theorem 23 we have that

$$(Z, X^0) \approx_\epsilon (U_d, X^0)$$

and

$$(Z, Y^0) \approx_\epsilon (U_d, Y^0).$$

We now ignore the error ϵ. Note that conditioned on the fixing of (X^0, X'^0), (Z, Z') is a deterministic function of (Y^0, Y'^0), and thus independent of (X, X'). Moreover, the average conditional min-entropy of X given this fixing is at least $n - \ell - 2(d + 2\ell + 2\log(1/\epsilon)) \geq 10m$ as long as c is large enough. Thus by Lemma 27 (note that the extractor from Z side can use seed length d') we have that

$$(Y^0, Y'^0, X_0, X_0', X_1, X_1', Z, Z') \approx_{O(\epsilon)} (Y^0, Y'^0, U_{3m}, X_0', U_{d_1}, X_1', Z, Z'),$$

where each U_{3m} is uniform given the previous random variables, but may depend on later random variables. Similarly, note that conditioned on the fixing of (Y^0, Y'^0), (Z, Z') is a deterministic function of (X^0, X'^0), and thus independent of (Y, Y'). Moreover, the average conditional min-entropy of Y given this fixing is at least $s - \ell - 2(d + 2\ell + 2\log(1/\epsilon)) \geq 10d$. Thus by Lemma 27 we have that

$$(Y_0, Y_0', Y_1, Y_1', Z, Z', X^0, X'^0) \approx_{O(\epsilon)} (U_{3d}, Y_0', U_{d_2}, Y_1', Z, Z', X^0, X'^0),$$

where each U_{3d} is uniform given the previous random variables, but may depend on later random variables. We can now fix (X^0, X'^0, Y^0, Y'^0), and conditioned on this fixing, we have that (X, X') and (Y, Y') are still independent, (X_0, X_0', X_1, X_1') is a deterministic function of (X, X'), and (Y_0, Y_0', Y_1, Y_1') is a deterministic function of (Y, Y'). Further they satisfy the look-ahead properties in the previous two equations. We will ignore the error for now since this only adds at most $O(\epsilon)$ to the final error.

We now claim that conditioned on the fixing of $(X_0, X_0', Y_0, Y_0', Y_1, Y_1')$ (and ignoring the error), the random variables $(V, V', \hat{X}, \hat{X}')$ and $(\hat{Y}, \hat{Y}')$ satisfy the conditions required by Lemma 36. To see this, note that if we fix (Y_0, Y_0', Y_1, Y_1'), then the average conditional

min-entropy of Y is at least $s - \ell - 2(d + 2\ell + 2\log(1/\epsilon)) - 2 \cdot 3d > 2s/3$ as long as c is large enough. Thus by Theorem 22 we have that

$$(\hat{Y}, X_0, X_0') \approx_\epsilon (U_{s/2}, X_0, X_0').$$

Thus conditioned on the further fixing of (X_0, X_0'), we have that $(\hat{Y}, \hat{Y}')$ is a deterministic function of (Y, Y'), and $s/2 \geq d_2$. On the other hand, conditioned on the fixing of (X_0, X_0') and (Y_0, Y_0'), we have X_1 is still close to uniform. Thus by Theorem 41 we have that for any $i \in [L]$,

$$|V_i, Y_1, Y_1' - U_m, Y_1, Y_1'| \leqslant \epsilon$$

and there exists $i \in [L]$ such that

$$|V_i, V_i', Y_1, Y_1' - U_m, V_i', Y_1, Y_1'| \leqslant \epsilon.$$

We now further fix (Y_1, Y_1'). Note that conditioned on this fixing (X, X') and (Y, Y') are still independent. Furthermore (V, V') is now a deterministic function of (X_1, X_1'), and thus independent of (Y, Y'). Finally, note that conditioned on the fixing of (X_0, X_0', X_1, X_1'), the average conditional min-entropy of X is at least $n - \ell - 2(d + 2\ell + 2\log(1/\epsilon)) - 2 \cdot 3m > 2n/3$. Thus by Theorem 22 we have that

$$(\hat{X}, Y_0, Y_0') \approx_\epsilon (U_{n/2}, Y_0, Y_0').$$

Thus conditioned on the further fixing of (Y_0, Y_0'), we have that $(\hat{X}, \hat{X}')$ is a deterministic function of (X, X'), and $n/2 \geq d_1$. Thus, even if conditioned on the fixing of $(X_0, X_0', X_1, X_1', Y_0, Y_0', Y_1, Y_1')$, we have that $(\hat{X}$ is close to $U_{n/2}$. Since (V, V') is obtained from (X_1, X_1', Y_1, Y_1'), we know that $(\hat{X}$ is close to uniform even given $(X_0, X_0', Y_0, Y_0', Y_1, Y_1')$ and (V, V'). Thus by Lemma 36 we have that

$$(\hat{V}, \hat{V}', Y, Y') \approx (U_m, \hat{V}', Y, Y'),$$

where the error is $O(L\epsilon) + O(L\epsilon) + O(\epsilon) = O(L\epsilon)$. ◀

Next we give another correlation breaker, which recycles the randomness used.

▶ **Construction 44.** *Inputs:*

- *Let $\ell, m \in \mathcal{N}$ be two integers, $\epsilon > 0$ be an error parameter.*
- *X, Y, two independent sources on n bits with min-entropy at least $n - \ell$.*
- *an advice string $\alpha \in \{0,1\}^L$ and an integer $2 \leq t \leq L$.*
- *An $(L, d_1, d_2, O(L\epsilon))$-NIPM$_y$ that is strong in X.*
- *Let IP be the two source extractor from Theorem 23.*

1. *Let $d' = O(\log(n/\epsilon))$ be the seed length of the extractor from Theorem 22, and let $d = 8\frac{\log L}{\log t}d'$. Let X^0 be a slice of X with length $d + 2\ell + 2\log(1/\epsilon)$, and Y^0 be a slice of Y with length $d + 2\ell + 2\log(1/\epsilon)$.*
2. *Compute $Z = \mathsf{IP}(X^0, Y^0)$ and output d bits.*
3. *Use X and Z to do an alternating extraction, and output $3\frac{\log L}{\log t} + 1$ random variables $X_0, \cdots, X_{3\frac{\log L}{\log t}}$ where each X_i has d_1 bits.*
4. *Use Y and Z to do an alternating extraction, and output two random variables Y_0, Y_1 where each Y_i has d_2 bits.*
5. *Use X_0, Y_0, α to obtain an $L \times m$ matrix V^0, where for any $i \in [L]$, $V_i^0 = \text{flip-flop}(X_0, Y_0, \alpha_i)$ and outputs m bits.*

6. *For $i = 1$ to $\frac{\log L}{\log t}$ do the following. Merge every t rows of V^{i-1} using* NIPM_y *and (X_{3i-2}, Y_i), and output d' bits. Concatenate the outputs to become another matrix W^i. Note that W^i has L/t^i rows. Then for every row $j \in [L/t^i]$, compute $V^i_j = \mathsf{Ext}(X_{3i}, W^i_j)$ to obtain a new matrix V^i. Finally let $Y_{i+1} = \mathsf{Ext}(Y, X_{3i-1})$ and output d_2 bits.*
7. *Output $\hat{V} = V^{\frac{\log L}{\log t}}$.*

We now have the following lemma.

▶ **Lemma 45.** *There exists a costant $c > 1$ such that the following holds. Suppose that for any $t \in \mathcal{N}$ there exists an $(t, d_1, d_2, O(t\epsilon))$-$\mathrm{NIPM}_y$ that is strong in X which outputs $d' = O(\log(n/\epsilon))$ bits, then there exists an explicit $(n-\ell, n-\ell, O(L\epsilon))$ correlation breaker with advice $\mathsf{AdvCB} : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^L \to \{0,1\}^m$ as long as $d_1 \geq 4m$, $m \geq c\log(d_2/\epsilon)$, and $n \geq c\frac{\log L}{\log t}\log(n/\epsilon) + max\{8\frac{\log L}{\log t}d_1, 2t \cdot d' + 4d_2\} + 5\ell + 4\log(1/\epsilon)$.*

Proof. Throughout the proof we will use letters with prime to denote the corresponding random variables obtained from (X', Y', α'). First, notice that both X^0 and Y^0 have min-entropy at least $d + \ell + 2\log(1/\epsilon)$. Thus by Theorem 23 we have that

$$(Z, X^0) \approx_\epsilon (U_d, X^0)$$

and

$$(Z, Y^0) \approx_\epsilon (U_d, Y^0).$$

We now ignore the error ϵ. Note that conditioned on the fixing of (X^0, X'^0), (Z, Z') is a deterministic function of (Y^0, Y'^0), and thus independent of (X, X'). Moreover, the average conditional min-entropy of X given this fixing is at least $n-\ell-2(d+2\ell+2\log(1/\epsilon)) \geq 8\frac{\log L}{\log t}d_1$ as long as c is large enough. Thus by Lemma 27 (note that the extractor from Z side can use seed length d') we have that

$$\begin{aligned}&(Y^0, Y'^0, Z, Z', X_0, X'_0, \cdots, X_{3\frac{\log L}{\log t}}, X'_{3\frac{\log L}{\log t}})\\ \approx_{O((\frac{\log L}{\log t})^2\epsilon)}\ &(Y^0, Y'^0, Z, Z', U_{d_1}, X'_0, \cdots, U_{d_1}, X'_{3\frac{\log L}{\log t}}),\end{aligned}$$

where each U_{d_1} is uniform given the previous random variables, but may depend on later random variables. Similarly, note that conditioned on the fixing of (Y^0, Y'^0), (Z, Z') is a deterministic function of (X^0, X'^0), and thus independent of (Y, Y'). Moreover, the average conditional min-entropy of Y given this fixing is at least $n - \ell - 2(d + 2\ell + 2\log(1/\epsilon)) \geq 4d_2$. Thus by Lemma 27 we have that

$$(Z, Z', X^0, X'^0, Y_0, Y'_0, Y_1, Y'_1) \approx_{O(\epsilon)} (Z, Z', X^0, X'^0, U_{d_2}, Y'_0, U_{d_2}),$$

where each U_{d_2} is uniform given the previous random variables, but may depend on later random variables. We can now fix (X^0, X'^0, Y^0, Y'^0), and conditioned on this fixing, we have that (X, X') and (Y, Y') are still independent, $(X_0, X'_0, \cdots, X_{3\frac{\log L}{\log t}}, X'_{3\frac{\log L}{\log t}})$ is a deterministic function of (X, X'), and (Y_0, Y'_0, Y_1, Y'_1) is a deterministic function of (Y, Y'). Further they satisfy the look-ahead properties in the previous two equations. We will ignore the error for now since this only adds at most $O((\frac{\log L}{\log t})^2\epsilon)$ to the final error.

Now by Theorem 41 we have that for any $i \in [L]$,

$$|V^0_i, Y_0, Y'_0 - U_m, Y_0, Y'_0| \leqslant \epsilon$$

and there exists $i \in [L]$ such that

$$|V_i^0, V_i'^0, Y_0, Y_0' - U_m, V_i'^0, Y_0, Y_0'| \leqslant \epsilon.$$

We now further fix (Y_0, Y_0'). Note that conditioned on this fixing (X, X') and (Y, Y') are still independent. Furthermore (V^0, V'^0) is now a deterministic function of (X_0, X_0'), and thus independent of (Y, Y'). Thus by the property of NIPM$_y$ we have that for every row j in W^1,

$$(W_j^1, V^0, V'^0, X_1, X_1') \approx_{O(t\epsilon)} (U_{d'}, V^0, V'^0, X_1, X_1'),$$

and there exists a row j such that

$$(W_j^1, W_j'^1, V^0, V'^0, X_1, X_1') \approx_{O(t\epsilon)} (U_{d'}, W_j'^1, V^0, V'^0, X_1, X_1').$$

Note that we have fixed (X^0, X'^0, Y^0, Y'^0), and if we further condition on the fixing of $(X_0, X_0', Y_0, Y_0', X_1, X_1')$, then (W^1, W'^1) is a deterministic function of (Y, Y'). Furthermore (X, X') and (Y, Y') are still independent. We will now use induction to prove the following claim (note that we have already fixed (X^0, X'^0, Y^0, Y'^0)).

▷ Claim 46. Let $T_i = (Y_0, Y_0', X_0, X_0', \cdots, X_{3i-2}, X_{3i-2}')$. In the i' th iteration, the following holds.

1. Conditioned on the further fixing of T_i, we have that (X, X') and (Y, Y') are still independent, and furthermore (W^i, W'^i) is a deterministic function of (Y, Y').
2. For every row j in W^i,

   $$(W_j^i, T_i) \approx_{\epsilon_i} (U_{d'}, T_i),$$

   and there exists a row j such that

   $$(W_j^i, W_j'^i, T_i) \approx_{\epsilon_i} (U_{d'}, W_j'^i, T_i),$$

   where $\epsilon_i = O(\sum_{j=1}^{i} t^j \epsilon)$.

Proof of the claim. The base case of $i = 1$ is already proved above. Now suppose the claim holds for the i'th iteration, we show that it also holds for the $i+1$'th iteration.

To see this, note that conditioned on the fixing of T_i, (X, X') and (Y, Y') are still independent, and furthermore (W^i, W'^i) is a deterministic function of (Y, Y') and thus independent of (X, X'). Note that Y_{i+1} is computed from Y and X_{3i-1} while V^i is computed from X_{3i} and W^i. Thus if we further fix X_{3i-1}, X_{3i-1}' and (W^i, W'^i), then (X, X') and (Y, Y') are still independent, and furthermore Y_{i+1} is a deterministic function of Y and V^i is a deterministic function of X_{3i}. Now W^{i+1} is computed from V^i, X_{3i+1} and Y_{i+1}. Thus if we further fix (X_{3i}, X_{3i}') and (X_{3i+1}, X_{3i+1}') (i.e., we have fixed T_{i+1}) then (X, X') and (Y, Y') are still independent, and furthermore (W^{i+1}, W'^{i+1}) is a deterministic function of (Y, Y').

Next, let h be the row in W^i such that

$$(W_h^i, W_h'^i, T_i) \approx_{\epsilon_i} (U_{d'}, W_h'^i, T_i).$$

Note that V^i has the same number of rows as W^i, and consider the merging of some t rows in V^i that contain row h into W_j^{i+1} (the merging of the other rows is similar and simpler). Without loss of generality assume that these t rows are row $1, 2, \cdots, t$.

First, since for every row j in W^i,

$$(W_j^i, T_i) \approx_{\epsilon_i} (U_{d'}, T_i),$$

and rows h in W^i and W'^i satisfy the independence property, by Theorem 22 (and ignoring the error ϵ_i) we have that for every $j \in [t]$,

$$(V_j^i, T_i, X_{3i-1}, X'_{3i-1}, W_j^i, W_j'^i) \approx_\epsilon (U_m, T_i, X_{3i-1}, X'_{3i-1}, W_j^i, W_j'^i),$$

and

$$(V_h^i, V_h'^i, T_i, X_{3i-1}, X'_{3i-1}, W_j^i, W_j'^i) \approx_\epsilon (U_m, V_h'^i, T_i, X_{3i-1}, X'_{3i-1}, W_j^i, W_j'^i).$$

This is because X_{3i} has average conditional min-entropy at least d_1 even conditioned on the fixing of (X_{3i-1}, X'_{3i-1}). We now ignore the error ϵ. Note that conditioned on the fixing of $(W_j^i, W_j'^i)$, we have that $(V_j^i, V_j'^i)$ is a deterministic function of (X_{3i}, X'_{3i}), and thus independent of (Y, Y'). We now fix $\{(W_j^i, W_j'^i), j \in [t]\}$. Note that conditioned on this fixing $\{V_j^i, j \in [t]\}$ and $\{V_j'^i, j \in [t]\}$ each is a $t \times m$ matrix, and a deterministic function of (X_{3i}, X'_{3i}). Further note that they form two matrices that meet the condition to apply an NIPM. Since $\{(W_j^i, W_j'^i), j \in [t]\}$ is a deterministic function of (Y, Y'), conditioned on this fixing (X, X') and (Y, Y') are still independent. Furthermore the average conditional min-entropy of Y is at least $n - \ell - 2(d + 2\ell + 2\log(1/\epsilon)) - 2d_2 - 2td' \geq 2d_2$. Thus by Theorem 22 we have that

$$(Y_{i+1}, X_{3i-1}) \approx_\epsilon (U_{d_2}, X_{3i-1}).$$

Note that conditioned on the fixing of X_{3i-1}, we have that Y_{i+1} is a deterministic function of Y. Thus we can now further fix (X_{3i-1}, X'_{3i-1}), and conditioned on this fixing, Y_{i+1} is still close to uniform. To conclude, now conditioned on the fixing of $\{(W_j^i, W_j'^i), j \in [t]\}$ and (X_{3i-1}, X'_{3i-1}), we have that $\{V_j^i, j \in [t]\}$ and $\{V_j'^i, j \in [t]\}$ each is a $t \times m$ matrix, and a deterministic function of (X_{3i}, X'_{3i}); Y_{i+1} is still close to uniform and (Y_{i+1}, Y'_{i+1}) is a deterministic function of (Y, Y'). Furthermore X_{3i+1} is close to uniform. Now we can use the property of NIPM_y to show that after merging these t rows, the corresponding row j in W^{i+1} satisfies

$$\begin{aligned}&(W_j^{i+1}, W_j'^{i+1}, T_i, X_{3i-1}, X'_{3i-1}, X_{3i}, X'_{3i}, X_{3i+1}, X'_{3i+1})\\ \approx_{t\epsilon}&(U_{d'}, W_j'^{i+1}, T_i, X_{3i-1}, X'_{3i-1}, X_{3i}, X'_{3i}, X_{3i+1}, X'_{3i+1}).\end{aligned}$$

Adding back all the errors we get that

$$(W_j^{i+1}, W_j'^{i+1}, T_{i+1}) \approx_{\epsilon_{i+1}} (U_{d'}, W_j'^{i+1}, T_{i+1}),$$

where $\epsilon_{i+1} = t\epsilon_i + O(t\epsilon) = O(\sum_{j=1}^{i+1} t^j \epsilon)$. ◁

Now we are basically done. In the last iteration we know that $W^{\frac{\log L}{\log t}}$ has reduced to one row, and $W^{\frac{\log L}{\log t}}$ is close to uniform given $W'^{\frac{\log L}{\log t}}$. Also conditioned on the fixing of $T_{\frac{\log L}{\log t}}$ they are deterministic functions of (Y, Y'). Thus when we use $W^{\frac{\log L}{\log t}}$ to extract $V^{\frac{\log L}{\log t}}$ from $X_{3\frac{\log L}{\log t}}$, by Theorem 22 we have that

$$(\hat{V}, \hat{V}', Y, Y') \approx (U_m, \hat{V}', Y, Y'),$$

where the error is $O(\sum_{j=1}^{\frac{\log L}{\log t}} t^j \epsilon) + O((\frac{\log L}{\log t})^2 \epsilon) = O(L\epsilon)$. ◀

6 The Constructions of Non-Malleable Extractors

In this section we construct our improved seeded non-malleable extractors and seedless non-malleable extractors. Both the constructions follow the general approach developed in recent works [12, 13, 21, 54], i.e., first obtaining an advice and then applying an appropriate correlation breaker with advice. First we need the following advice generator from [12].

▶ **Theorem 47** ([12]). *There exist a constant $c > 0$ such that for all $n > 0$ and any $\epsilon > 0$, there exists an explicit function $\mathsf{AdvGen} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^L$ with $L = c\log(n/\epsilon)$ satisfying the following: Let X be an (n,k)-source, and Y be an independent uniform seed on d bits. Let Y' be a random variable on d bits s.t $Y' \neq Y$, and (Y,Y') is independent of X. Then with probability at least $1-\epsilon$, $\mathsf{AdvGen}(X,Y) \neq \mathsf{AdvGen}(X,Y')$. Moreover, there is a deterministic function g such that $\mathsf{AdvGen}(X,Y)$ is computed as follows. Let Y_1 be a small slice of Y with length $O(\log(n/\epsilon))$, compute $Z = \mathsf{Ext}(X,Y_1)$ where Ext is an optimal seeded extractor from Theorem 22 which outputs $O(\log(n/\epsilon))$ bits. Finally compute $Y_2 = g(Y,Z)$ which outputs $O(\log(1/\epsilon))$ bits and let $\mathsf{AdvGen}(X,Y) = (Y_1,Y_2)$.*

For two independent sources we also have the following slightly different advice generator.

▶ **Theorem 48** ([12]). *There exist constants $0 < \gamma < \beta < 1$ such that for all $n > 0$ and any $\epsilon \geq \epsilon'$ for some $\epsilon' = 2^{-\Omega(n)}$, there exists an explicit function $\mathsf{AdvGen} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^L$ with $L = 2\beta n + O(\log(1/\epsilon))$ satisfying the following: Let X,Y be two independent $(n,(1-\gamma)n)$-sources, and (X',Y') be some tampered versions of (X,Y), such that (X,X') is independent of (Y,Y'). Furthermore either $X \neq X'$ or $Y \neq Y'$. Then with probability at least $1-\epsilon$, $\mathsf{AdvGen}(X,Y) \neq \mathsf{AdvGen}(X',Y')$. Moreover, there is a deterministic function g such that $\mathsf{AdvGen}(X,Y)$ is computed as follows. Let X_1,Y_1 be two small slice of X,Y respectively, with length βn, compute $Z = \mathsf{IP}(X,Y_1)$ where IP is the inner product two source extractor from Theorem 23 which outputs $\Omega(n)$ bits. Finally compute $X_2 = g(X,Z), Y_2 = g(Y,Z)$ which both output $O(\log(1/\epsilon))$ bits and let $\mathsf{AdvGen}(X,Y) = (X_1,X_2,Y_1,Y_2)$.*

By using these advice generators, the general approach of constructing seeded non-malleable extractors and seedless non-malleable extractors can be summarized in the following two theorems.

▶ **Theorem 49.** *[12, 13, 21, 54] There is a constant $c > 1$ such that for any $n,k,d \in \mathcal{N}$ and $\epsilon_1,\epsilon_2 > 0$, if there is a $(k - c\log(n/\epsilon_1), d - c\log(n/\epsilon_1), \epsilon_2)$ advice correlation breaker $\mathsf{AdvCB} : \{0,1\}^k \times \{0,1\}^d \times \{0,1\}^{c\log(n/\epsilon_1)} \to \{0,1\}^m$, then there exists an $(O(k), \epsilon_1+\epsilon_2)$ seeded non-malleable extractor $\mathsf{nmExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$. Furthermore if $m \geq c\log(d/\epsilon_1)$ then there exists an $(O(k), \epsilon_1+\epsilon_2)$ seeded non-malleable extractor $\mathsf{nmExt} : \{0,1\}^n \times \{0,1\}^{O(d)} \to \{0,1\}^{\Omega(k)}$.*

Sketch. The seeded non-malleable extractor is constructed as follows. First use the seed and the source to obtain an advice as in Theorem 47 with error $\epsilon_1/3$, however when we compute $Z = \mathsf{Ext}(X,Y_1)$ we in fact output $Z_1 = \mathsf{Ext}(X,Y_1)$ with k bits and choose Z to be a small slice of Z_1 with length $O(\log(n/\epsilon))$. Then we can fix the random variables (Y_1,Y_1',Z,Z',Y_2,Y_2'). Note that conditioned on this fixing (X,X') is still independent of (Y,Y'), and (Z_1,Z_1') is a deterministic function of (X,X') thus is independent of (Y,Y'). Furthermore with probability $1-\epsilon_1/3$, Z_1 has min-entropy at least $k - O(\log(n/\epsilon_1))$ and Y has min-entropy at least $d - O(\log(n/\epsilon_1))$. We can now apply the correlation breaker to (Z_1,Y) and the advice to get the desired output, where the total error is at most $\epsilon_1/3 + \epsilon_1/3 + \epsilon_1/3 + \epsilon_2 = \epsilon_1 + \epsilon_2$. If the output m is large enough (i.e., $m \geq c\log(d/\epsilon_1)$), then we can use it to extract from Y and then extract again from Z_1 to increase the output length to $\Omega(k)$. ◀

▶ **Theorem 50.** *[12, 13, 21, 54] There are constants $c > 1$, $0 < \gamma < \beta < 1/100$ such that for any $n \in \mathcal{N}$ and $\epsilon_1, \epsilon_2 > 0$, if there is a $((1-2\beta)n - c\log(n/\epsilon_1), (1-2\beta)n - c\log(n/\epsilon_1), \epsilon_2)$ advice correlation breaker* $\mathsf{AdvCB} : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^{2\beta n + c\log(1/\epsilon_1)} \to \{0,1\}^m$*, then there exists an $((1-\gamma)n, (1-\gamma)n, \epsilon_1 + \epsilon_2)$ non-malleable two source extractor* $\mathsf{nmExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$*. Furthermore if $m \geq c\log(n/\epsilon_1)$ then there exists an $((1-\gamma)n, (1-\gamma)n, \epsilon_1 + \epsilon_2)$ non-malleable two source extractor* $\mathsf{nmExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{\Omega(n)}$*.*

Sketch. The non-malleable two-source extractor is constructed as follows. First use the two independent sources (X, Y) to obtain an advice as in Theorem 48 with error $\epsilon_1/3$, then we can fix the random variables $(X_1, X_1', Y_1, Y_1', X_2, X_2', Y_2, Y_2')$. Note that conditioned on this fixing (X, X') is still independent of (Y, Y'), furthermore with probability $1 - \epsilon_1/3$, both X and Y have min-entropy at least $(1-\gamma)n - \beta n - c\log(1/\epsilon_1) \geq (1-2\beta)n - c\log(1/\epsilon_1)$. We can now apply the correlation breaker to (X, Y) and the advice to get the desired output, where the total error is at most $\epsilon_1/3 + \epsilon_1/3 + \epsilon_1/3 + \epsilon_2 = \epsilon_1 + \epsilon_2$. If the output m is large enough (i.e., $m \geq c\log(d/\epsilon_1)$), then we can use it to extract from Y and then extract again from X to increase the output length to $\Omega(n)$. ◀

Combined with our new correlation breakers with advice, we have the following new constructions of non-malleable extractors.

▶ **Theorem 51.** *There exists a constant $C > 1$ such that for any constant $a \in \mathcal{N}, \dashv \geq \in$, any $n, k \in \mathcal{N}$ and any $0 < \epsilon < 1$ with $k \geq C(\log n + a\log(1/\epsilon))$, there is an explicit construction of a strong seeded (k, ϵ) non-malleable extractor $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n) + \log(1/\epsilon)2^{O(a(\log\log(1/\epsilon))^{\frac{1}{a}})}$ and $m = \Omega(k)$. Alternatively, we can also achieve entropy $k \geq C\log n + \log(1/\epsilon)2^{C\cdot a(\log\log(1/\epsilon))^{\frac{1}{a}}}$ and $d = O(\log n + a\log(1/\epsilon))$.*

Proof. The theorem is obtained by combining Theorem 49, Lemma 43 and Lemma 36. We choose an error ϵ' to be the error in Theorem 49, Lemma 43 and Lemma 36. Thus the total error is $O(L\epsilon')$ where $L = O(\log(n/\epsilon'))$. To ensure $O(L\epsilon') = \epsilon$ it suffices to take $\epsilon' = \frac{\epsilon}{c\log(n/\epsilon)}$ for some constant $c > 1$. We know $\ell = O(\log(n/\epsilon'))$. Therefore to apply Lemma 43 and Lemma 36, we need to find m, d', d_1, d_2 such that

$$d' \geq c(\log\max\{d_1, d_2\} + \log(1/\epsilon')), m \geq d', d_1 \geq 8a\cdot d' + 6m \text{ and } d_2 \geq 8a\cdot d' + c^{a\cdot\log^{\frac{1}{a}} L}\cdot d'.$$

Then we can take

$$k = O(d_1 + m + \ell + \log(1/\epsilon')) \text{ and } d = O(d_2 + m + \ell + \log(1/\epsilon')).$$

It can be seen that we can take $m = O(\log(n/\epsilon'))$, $d' = O(\log\log n + \log(1/\epsilon'))$, $d_1 = 8a\cdot d' + 6m = O(\log n + a\log(1/\epsilon'))$ and $d_2 = 2^{O(a(\log\log(n/\epsilon'))^{\frac{1}{a}})}\cdot d'$. We now consider two cases. First, $\log(1/\epsilon') > \frac{\log n}{c'^{a(\log\log n)^{\frac{1}{a}}}}$ for some large constant c'. In this case we have that

$$\log(1/\epsilon') > \frac{\log n}{c'^{a(\log\log n)^{\frac{1}{a}}}} > \sqrt{\log n}$$

for any $a \geq 2$. Thus

$$\log\log(n/\epsilon')) = \log(\log n + \log(1/\epsilon')) < \log(\log^2(1/\epsilon') + \log(1/\epsilon')) < 2\log\log(1/\epsilon') + 1.$$

Also note that $d' = O(\log\log n + \log(1/\epsilon')) = O(\log(1/\epsilon'))$. Thus in this case we have $d_2 \leq O(\log(1/\epsilon'))2^{O(a(\log\log(1/\epsilon'))^{\frac{1}{a}})} = \log(1/\epsilon')2^{O(a(\log\log(1/\epsilon'))^{\frac{1}{a}})}$. Next, consider the case

where $\log(1/\epsilon') \leq \frac{\log n}{c'^{a(\log\log n)^{\frac{1}{a}}}}$. In this case note that we have $\log(1/\epsilon') < \log n$ and thus $2^{O(a(\log\log(n/\epsilon'))^{\frac{1}{a}})} < 2^{O(a(\log\log(n))^{\frac{1}{a}})}$. Therefore when c' is large enough and $a \geq 2$ we have that

$$d_2 \leq 2^{O(a(\log\log(n))^{\frac{1}{a}})}(\log\log n + \log(1/\epsilon')) \leq \log n.$$

Therefore altogether we have that $d_2 \leq (\log n + \log(1/\epsilon')2^{O(a(\log\log(1/\epsilon'))^{\frac{1}{a}})})$ and $d = O(d_2 + m + \ell + \log(1/\epsilon')) = O(\log n) + \log(1/\epsilon')2^{O(a(\log\log(1/\epsilon'))^{\frac{1}{a}})}$. Note that $\log(1/\epsilon') = \log(1/\epsilon) + \log(\log n + \log(1/\epsilon)) + O(1)$, a careful analysis similar as above shows that we also have that

$$d = O(\log n) + \log(1/\epsilon)2^{O(a(\log\log(1/\epsilon))^{\frac{1}{a}})}.$$

Note that the correlation breaker is completely symmetric to both sources, and the only difference is in generating the advice. Thus after advice generation which costs both sources $O(\log(n/\epsilon))$ entropy, we can switch the role of the seed and the source. Therefore we can also get the other setting of parameters where $k \geq C\log n + \log(1/\epsilon)2^{C\cdot a(\log\log(1/\epsilon))^{\frac{1}{a}}}$ and $d = O(\log n + a\log(1/\epsilon))$. ◀

By using this theorem, we can actually improve the entropy requirement of the non-malleable extractor. Specifically, we have the following theorem.

▶ **Theorem 52.** *There exists a constant $C > 1$ such that for any constant $a \in \mathcal{N}, \dashv \geq \in$, any $n, k \in \mathcal{N}$ and any $0 < \epsilon < 1$ with $k \geq C(\log\log n + a\log(1/\epsilon))$, there is an explicit construction of a strong seeded (k, ϵ) non-malleable extractor $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n) + \log(1/\epsilon)2^{O(a(\log\log(1/\epsilon))^{\frac{1}{a}})}$ and $m = \Omega(k)$. Alternatively, we can also achieve entropy $k \geq C\log\log n + \log(1/\epsilon)2^{C\cdot a(\log\log(1/\epsilon))^{\frac{1}{a}}}$ and $d = O(\log n + a\log(1/\epsilon))$.*

Proof. We start by taking a slice of the seed Y_1 with length $O(\log(n/\epsilon))$ to extract from the source, and output some $k' = 0.9k$ uniform bits with error $\epsilon/2$. Note that conditioned on the fixing of (Y_1, Y_1') where Y_1' is the tampered version, the two sources are still independent, and the seed now has average conditional entropy at least $d - O(\log(n/\epsilon))$. We now switch the role of the seed and the source, and use the output of the extractor from the source as the seed of a non-malleable extractor and apply Theorem 51 with error $\epsilon/2$, so that the final error is ϵ.

Note that now we know the original seed is different from its tampered version, so we only need to obtain advice from the original seed and thus the advice size is $O(\log(d/\epsilon))$. Now we only need

$$k \geq C(\log d + a\log(1/\epsilon))$$

and

$$d - O(\log(n/\epsilon)) \geq C\log k + \log(1/\epsilon)2^{C\cdot a(\log\log(1/\epsilon))^{\frac{1}{a}}}.$$

Thus we can choose

$$k \geq C'(\log\log n + a\log(1/\epsilon))$$

for some slightly larger constant $C' > 1$, while the requirement of the seed is still

$$d = O(\log n) + \log(1/\epsilon)2^{O(a(\log\log(1/\epsilon))^{\frac{1}{a}})}.$$

Similarly, we can switch the role of the seed and the source to get the other setting of parameters. ◀

The next theorem improves the seed length, at the price of using a slightly larger entropy.

▶ **Theorem 53.** *There exists a constant $C > 1$ such that for any $n, k \in \mathcal{N}$ and $0 < \epsilon < 1$ with $k \geq C(\log n + \log(1/\epsilon) \log\log\log(1/\epsilon))$, there is an explicit construction of a strong seeded (k, ϵ) non-malleable extractor $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n + \log(1/\epsilon)(\log\log(1/\epsilon))^2)$ and $m = \Omega(k)$.*

Proof. The theorem is obtained by combining Theorem 49, Lemma 43 and Lemma 38. Again, We choose an error ϵ' to be the error in Theorem 49, Lemma 43 and Lemma 38. Thus the total error is $O(L\epsilon')$ where $L = O(\log(n/\epsilon'))$. To ensure $O(L\epsilon') = \epsilon$ it suffices to take $\epsilon' = \frac{\epsilon}{c\log(n/\epsilon)}$ for some constant $c > 1$. We also know $\ell = O(\log(n/\epsilon'))$ in Lemma 43. Thus to apply Lemma 38, we need to find m, d_1, d_2 such that (for simplicity, we choose $a = 4$ in Lemma 38),

$$m \geq c\log(d_2/\epsilon'), d_1 \geq 20c\log\log L\log(d_2/\epsilon') + 6m \text{ and } d_2 \geq 20c\log^2 L\log(d_1/\epsilon').$$

Then we can take

$$k = O(d_1 + m + \ell + \log(1/\epsilon')) \text{ and } d = O(d_2 + m + \ell + \log(1/\epsilon')).$$

A careful but tedious calculation shows that we can choose $k \geq C(\log n + \log(1/\epsilon')\log\log\log(1/\epsilon'))$ for some large enough constant $C > 1$, and $d = O(\log n + \log(1/\epsilon')(\log\log(1/\epsilon'))^2)$. Note that we can choose $m = O(\log(n/\epsilon'))$ for a large enough constant in $O(.)$, thus by Theorem 49 we can get an output length of $\Omega(k)$. Finally, note that $\log(n/\epsilon') = O(\log(n/\epsilon))$, thus the theorem follows. ◀

Similar to what we have done above, we can also use this to get improved parameters. Specifically, we have

▶ **Theorem 54.** *There exists a constant $C > 1$ such that for any $n, k \in \mathcal{N}$ and $0 < \epsilon < 1$ with $k \geq C(\log\log n + \log(1/\epsilon)\log\log\log(1/\epsilon))$, there is an explicit construction of a strong seeded (k, ϵ) non-malleable extractor $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n + \log(1/\epsilon)(\log\log(1/\epsilon))^2)$ and $m = \Omega(k)$. Alternatively, we can also achieve entropy $k \geq C(\log\log n + \log(1/\epsilon)(\log\log(1/\epsilon))^2)$ and seed length $d = O(\log n + \log(1/\epsilon)\log\log\log(1/\epsilon))$.*

For non-malleable two-source extractors we have the following theorem.

▶ **Theorem 55.** *There exists a constant $0 < \gamma < 1$ and a non-malleable two-source extractor for $(n, (1-\gamma)n)$ sources with error $2^{-\Omega(n\log\log n/\log n)}$ and output length $\Omega(n)$.*

Proof. The theorem is obtained by combining Theorem 50, Lemma 45 and Lemma 36. Again, we choose an error ϵ' to be the error in Theorem 49, Lemma 43 and Lemma 38. Thus the total error is $O(L\epsilon')$ where $L = O(n)$. To ensure $O(L\epsilon') = \epsilon$ it suffices to take $\epsilon' = \frac{\epsilon}{cn}$ for some constant c. We also know $\ell = 2\beta n + o(n)$ for some constant $\beta < 1/100$ in Lemma 45. We choose $a = 2$ in Lemma 36 and thus we obtain a correlation breaker with $m = O(\log(n/\epsilon'))$, $d_1 = O(\log(n/\epsilon'))$ and $d_2 = \log(n/\epsilon')2^{O(\sqrt{\log t})}$ where t is the parameter in Construction 44 with $t \leq L$. Note that this also satisfies that $d_1 \geq 4m$ and $m \geq c\log(d_2/\epsilon)$ as required by Lemma 45.

Now we need to ensure that

$$(1-\beta)n \geq c\frac{\log L}{\log t}\log(n/\epsilon') + max\{8\frac{\log L}{\log t}d_1, 2t \cdot d' + 4d_2\} + 5\ell + 4\log(1/\epsilon'),$$

where $d' = O(\log(n/\epsilon'))$. We choose $t = \frac{\log L}{\log\log L}$ and this gives us

$$(1-12\beta)n \geq C\frac{\log L}{\log\log L}\log(n/\epsilon'),$$

for some constant $C > 1$. Note that $\log(n/\epsilon') = O(\log(n/\epsilon))$ thus we can set $\epsilon = 2^{-\Omega(n\log\log n/\log n)}$ and satisfy the above inequality. ◀

For applications in two-source extractors, we first need the following generalization of non-malleable extractors, which allows multiple tampering.

▶ **Definition 56** (Seeded t-Non-malleable extractor). *A function* $\mathrm{snmExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a seeded t-non-malleable extractor for min-entropy k and error ϵ if the following holds : If X is a source on $\{0,1\}^n$ with min-entropy k and $\mathcal{A}_1, \cdots, \mathcal{A}_t : \{0,1\}^d \to \{0,1\}^d$ are t arbitrary tampering functions with no fixed points, then*

$$\big|\mathrm{snmExt}(X, U_d) \circ \{\mathrm{snmExt}(X, \mathcal{A}_i(U_d)), i \in [t]\} \circ U_d \\ - U_m \circ \{\mathrm{snmExt}(X, \mathcal{A}_i(U_d)), i \in [t]\} \circ U_d\big| < \epsilon$$

where U_m is independent of U_d and X.

The following theorem is a special case of Theorem 8.6 proved in [54].

▶ **Theorem 57.** *Suppose there is a function f, a constant $\gamma > 0$ and an explicit non-malleable two-source extractor for $(f(\epsilon), (1-\gamma)f(\epsilon))$ sources with error ϵ and output length $\Omega(f(\epsilon))$. Then there is a constant $C > 0$ such that for any $0 < \epsilon < 1$ with $k \geq Ct^2(\log n + f(\epsilon))$, there is an explicit strong seeded t-non-malleable extractor for (n,k) sources with seed length $d = Ct^2(\log n + f(\epsilon))$, error $O(t\epsilon)$ and output length $\Omega(f(\epsilon))$.*

Combined with Theorem 55, this immediately gives the following theorem.

▶ **Theorem 58.** *There is a constant $C > 0$ such that for any $0 < \epsilon < 1$ and $n, k \in \mathcal{N}$ with $k \geq Ct^2(\log n + \frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)})$, there is an explicit strong seeded t-non-malleable extractor for (n,k) sources with seed length $d = Ct^2(\log n + \frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)})$, error $O(t\epsilon)$ and output length $\Omega(k/t^2)$. As a special case, there exists a seeded non-malleable extractor for entropy $k \geq C(\log n + \frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)})$ and seed length $d = C(\log n + \frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)})$.*

Similar techniques as above can reduce the $\log n$ term in the entropy requirement to $\log\log n$, so we get

▶ **Theorem 59.** *There is a constant $C > 0$ such that for any $0 < \epsilon < 1$ and $n, k \in \mathcal{N}$ with $k \geq C(\log\log n + \frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)})$, there is an explicit strong seeded non-malleable extractor for (n,k) sources with seed length and seed length $d = C(\log n + \frac{\log(1/\epsilon)\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)})$.*

Ben-Aroya et. al [8] proved the following theorem.

▶ **Theorem 60** ([8]). *Suppose there is a function f and an explicit strong seeded t-non-malleable extractor (n,k') sources with seed length and entropy requirement $d = k' = f(t,\epsilon)$, then for every constant $\epsilon > 0$ there exist constants $t = t(\epsilon), c = c(\epsilon)$ and an explicit extractor* $\mathsf{Ext} : (\{0,1\}^n)^2 \to \{0,1\}$ *for two independent (n,k) sources with $k \geq f(t, 1/n^c)$ and error ϵ.*

Combined with Theorem 53, this immediately gives the following theorem.

▶ **Theorem 61.** *For every constant $\epsilon > 0$, there exists a constant $C > 1$ and an explicit two source extractor* $\mathsf{Ext} : (\{0,1\}^n)^2 \to \{0,1\}$ *for entropy $k \geq C\frac{\log n \log\log n}{\log\log\log n}$ with error ϵ.*

7 Non-Malleable Two-Source Extractor and Non-Malleable Code

Formally, non-malleable codes are defined as follows.

▶ **Definition 62** ([1]). *Let* NM_k *denote the set of trivial manipulation functions on* k*-bit strings, which consists of the identity function* $I(x) = x$ *and all constant functions* $f_c(x) = c$*, where* $c \in \{0,1\}^k$*. Let* $E : \{0,1\}^k \to \{0,1\}^m$ *be an efficient randomized* encoding *function, and* $D : \{0,1\}^m \to \{0,1\}^k$ *be an efficient deterministic* decoding *function. Let* $\mathcal{F} : \{0,1\}^m \to \{0,1\}^m$ *be some class of functions. We say that the pair* (E, D) *defines an* $(\mathcal{F}, k, \epsilon)$-non-malleable code*, if for all* $f \in \mathcal{F}$ *there exists a probability distribution* G *over* NM_k*, such that for all* $x \in \{0,1\}^k$*, we have*

$$|D(f(E(x))) - G(x)| \leq \epsilon.$$

▶ Remark 63. The above definition is slightly different form the original definition in [36]. However, [1] shows that the two definitions are equivalent.

We will mainly be focusing on the following family of tampering functions in this paper.

▶ **Definition 64.** *Given any* $t > 1$*, let* $\mathcal{S}_n^t$ *denote the tampering family in the* t-split-state-model*, where the adversary applies* t *arbitrarily correlated functions* $h_1, \cdots, h_t$ *to* t *separate,* n*-bit parts of string. Each* h_i *can only be applied to the* i*-th part individually.*

We remark that even though the functions $h_1, \cdots, h_t$ can be correlated, their correlation is independent of the original codewords. Thus, they are actually a convex combination of independent functions, applied to each part of the codeword. Therefore, without loss of generality we can assume that each h_i is a deterministic function, which acts on the i-th part of the codeword individually.We will mainly consider the case of $t = 2$, i.e., the two-split-state model. We recall the original definition of non-malleable two-source extractors by Cheraghchi and Gursuswami [18]. First we define the following function.

$$\mathrm{copy}(x, y) = \begin{cases} x & \text{if } x \neq same^\star \\ y & \text{if } x = same^\star \end{cases}$$

▶ **Definition 65** (Seedless Non-Malleable 2-Source Extractor). *A function* $\mathsf{nmExt} : (\{0,1\}^n)^2 \to \{0,1\}^m$ *is a* (k, ϵ)*-seedless non-malleable extractor for two independent sources, if it satisfies the following property: Let* X, Y *be two independent* (n, k) *sources, and* $f_1, f_2 : \{0,1\}^n \to \{0,1\}^n$ *be two arbitrary tampering functions, then*

1. $|\mathsf{nmExt}(X, Y) - U_m| \leq \epsilon$.
2. *There is a distribution* $\mathcal{D}$ *over* $\{0,1\}^m \cup \{same^\star\}$ *such that for an independent* Z *sampled from* D*, we have*

$$(\mathsf{nmExt}(X, Y), \mathsf{nmExt}(f_1(X), f_2(Y))) \approx_\epsilon (\mathsf{nmExt}(X, Y), \mathrm{copy}(Z, \mathsf{nmExt}(X, Y))).$$

Cheraghchi and Gursuswami [18] showed that the relaxed definition 5 implies the above general definition with a small loss in parameters. Specifically, we have

▶ **Lemma 66** ([18]). *Let* nmExt *be a* $(k - \log(1/\epsilon), \epsilon)$*-non-malleable two-source extractor according to Definition 5. Then* nmExt *is a* $(k, 4\epsilon)$*-non-malleable two-source extractor according to Definition 65.*

The following theorem was proved by Cheraghchi and Gursuswami [18], which establishes a connection between seedless non-malleable extractors and non-malleable codes.

▶ **Theorem 67.** *Let* $\mathsf{nmExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *be a polynomial time computable seedless* 2*-non-malleable extractor at min-entropy* n *with error* ϵ*. Then there exists an explicit non-malleable code with an efficient decoder in the* 2*-split-state model with block length* $= 2n$*, rate* $= \frac{m}{2n}$ *and error* $= 2^{m+1}\epsilon$*.*

One can construct a non-malleable code in the 2-split-state model from a non-malleable two-source extractor as follows: Given any message $s \in \{0,1\}^m$, the encoding $\mathsf{Enc}(s)$ is done by outputting a uniformly random string from the set $\mathsf{nmExt}^{-1}(s) \subset \{0,1\}^{2n}$. Given any codeword $c \in \{0,1\}^{2n}$, the decoding $\mathsf{Dec}(c)$ is done by outputting $\mathsf{nmExt}(c)$. Thus, to get an efficient encoder we need a way to efficiently uniformly sample from the pre-image of any output of the extractor.

Since our new non-malleable two-source extractor follows the same structure as in [54], we can use the same sampling procedure there to efficiently uniformly sample from the pre-image of any output of the extractor. We briefly recall the construction and sampling procedure in [54].

The extractor construction and sampling

The high level structure of the non-malleable two-source extractor in [54] is as follows. First take two small slices (X_1, Y_1) of both sources and apply the inner product based two-source extractor, as in Theorem 23. Then, use the output to sample $O(\log(1/\epsilon))$ bits from the encodings of both sources, using a randomness efficient sampler and an asymptotically good linear encoding of the sources. We need an asymptotically good encoding since then we only need to sample $O(\log(1/\epsilon))$ bits to ensure that the sampling of two different codewords are different with probability at least $1 - \epsilon$. The advice is then obtained by combining the slices and the sample bits. Now, take two larger slices (X_2, Y_2) of both sources and apply the correlation breaker. Finally, take another larger slice of either source (say X_3 from X) and apply a strong linear seeded extractor, which is easy to invert and has the same pre-image size for any output. By limiting the size of each slice to be small, the construction ensures that there are at least $n/2$ bits of each source that are only used in the encoding of the sources but never used in the subsequent extraction.

Now to sample uniformly from the pre-image of any output, we first uniformly independently generate the slices (X_1, Y_1, X_2, Y_2) and the sampled bits Z. From these we can compute the coordinates of the sampled bits and the output of the correlation breaker. Now we can invert the linear seeded extractor and uniformly sample X_3 given the output of the extractor and the output of the correlation breaker (which is used as the seed of the linear seeded extractor). Now, to sample the rest of the bits, we need to condition on the event that the sample bits from the encoding of the sources are indeed Z. Note that Z has size at most αn for some small constant $\alpha < 1/2$ since we can restrict the error to be at least some $2^{-\Omega(n)}$. Also note that for each source we have already sampled some bits but there are still at least $n/2$ un-sampled free bits, thus we insist on that no matter which αn columns of the generating matrix of the encoding we look at, the sub matrix corresponding to these columns and the last $n/2$ rows have full column rank. If this is true then no matter which coordinates we use and what Z is, the pre-image always have the same size and we can uniformly sample from the pre-image by solving a system of linear equations.

In [54], we use the Reed-Solomn encoding for each source with field $\mathbf{F}_q$ for $q \approx n$. This is asymptotically good and also satisfies the property that any sub matrix with less columns than rows has full column rank since it is a Vandermonde matrix. However in this case each symbol has roughly $\log n$ bits so we can sample at most $n/\log n$ symbols (otherwise fixing them may already cost us all the entropy), thus the best error we can get using this encoding is $2^{-n/\log n}$.

We now give a new construction of non-malleable two-source extractors for two $(n,(1-\gamma)n)$ sources, where $0<\gamma<1$ is some constant. First, we need the following ingredients.

▶ **Theorem 68** ([54]). *There exists a constant $0<\alpha<1$ such that for any $n\in\mathcal{N}$ and $2^{-\alpha n}<\epsilon<1$ there exists a linear seeded strong extractor $\mathsf{IExt}:\{0,1\}^n\times\{0,1\}^d\to\{0,1\}^{0.3d}$ with $d=O(\log(n/\epsilon))$ and the following property. If X is a $(n,0.9n)$ source and R is an independent uniform seed on $\{0,1\}^d$, then*

$$|(\mathsf{IExt}(X,R),R)-(U_{0.3d},R)|\le\epsilon.$$

Furthermore for any $s\in\{0,1\}^{0.3d}$ and any $r\in\{0,1\}^d$, $|\mathsf{IExt}(\cdot,r)^{-1}(s)|=2^{n-0.3d}$.

▶ **Definition 69** (Averaging sampler [64]). *A function $\mathsf{Samp}:\{0,1\}^r\to[n]^t$ is a (μ,θ,γ) averaging sampler if for every function $f:[n]\to[0,1]$ with average value $\frac{1}{n}\sum_i f(i)\geqslant\mu$, it holds that*

$$\Pr_{i_1,\ldots,i_t\leftarrow\mathsf{Samp}(U_R)}\left[\frac{1}{t}\sum_i f(i)<\mu-\theta\right]\le\gamma.$$

Samp *has distinct samples if for every $x\in\{0,1\}^r$, the samples produced by $\mathsf{Samp}(x)$ are all distinct.*

▶ **Theorem 70** ([64]). *Let $1\ge\delta\ge3\tau>0$. Suppose that $\mathsf{Samp}:\{0,1\}^r\to[n]^t$ is an (μ,θ,γ) averaging sampler with distinct samples for $\mu=(\delta-2\tau)/\log(1/\tau)$ and $\theta=\tau/\log(1/\tau)$. Then for every δn-source X on $\{0,1\}^n$, the random variable $(U_r,X_{Samp(U_r)})$ is $(\gamma+2^{-\Omega(\tau n)})$-close to (U_r,W) where for every $a\in\{0,1\}^r$, the random variable $W|_{U_r=a}$ is $(\delta-3\tau)t$-source.*

▶ **Theorem 71** ([64]). *For every $0<\theta<\mu<1$, $\gamma>0$, and $n\in\mathcal{N}$, there is an explicit (μ,θ,γ) averaging sampler $\mathsf{Samp}:\{0,1\}^r\to[n]^t$ that uses*

- *t distinct samples for any $t\in[t_0,n]$, where $t_0=O(\frac{1}{\theta^2}\log(1/\gamma))$, and*
- *$r=\log(n/t)+\log(1/\gamma)\mathsf{poly}(1/\theta)$ random bits.*

7.1 A new advice generator

Here we show that we can give a new advice generator with optimal advice length. We have the following construction. Let (X,Y) be two independent $(n,(1-\tau)n)$ sources. Let IP be the inner product two-source extractor from Theorem 23, and $\mathsf{Samp}:$ be the sampler from Theorem 70. Let $L>0$ be a parameter, and $c>0$ be a constant to be chosen later. We have the following algorithm.

1. Let $n_1=3\tau n$. Divide X into $X=(X_1,X_2)$ such that X_1 has n_1 bits and X_2 has $n_2=(1-3\tau)n$ bits. Similarly divide Y into $Y=(Y_1,Y_2)$ such that Y_1 has n_1 bits and Y_2 has $n_2=(1-3\tau)n$ bits.
2. Compute $Z=\mathsf{IP}(X_1,Y_1)$ which outputs $r=\Omega(n)\le\tau n$ bits.
   item Let $\mathbf{F}$ be the finite field $\mathbf{F}_{2^{\log n}}$. Let $n_0=\frac{n_2}{\log n}$. Let $\mathsf{RS}:\mathbf{F}^{n_0}\to\mathbf{F}^n$ be the Reed-Solomon code encoding n_0 symbols of $\mathbf{F}$ to n symbols in $\mathbf{F}$ (we slightly abuse the use of RS to denote both the code and the encoder). Thus RS is a $[n,n_0,n-n_0+1]_n$ error correcting code. Let $\hat{X}_2$ be X_2 written backwards, and similarly $\hat{Y}_2$ be Y_2 written backwards. Let $\overline{X}_2=\mathsf{RS}(\hat{X}_2)$ and $\overline{Y}_2=\mathsf{RS}(\hat{Y}_2)$.
3. Use Z to sample $r/\log n$ distinct symbols from $\overline{X}_2$ (i.e., use each $\log n$ bits to sample a symbol), and write the symbols as a binary string $\tilde{X}_2$. Note that $\tilde{X}_2$ has r bits. Similarly, use Z to sample $r/\log n$ distinct symbols from $\overline{Y}_2$ and obtain a binary string $\tilde{Y}_2$ with r bits.

4. Let $V_1 = X_1 \circ Y_1 \circ \tilde{X}_2 \circ \tilde{Y}_2$.
5. Take a slice of X_2 with length $15\tau n$, and let it be X_3. Similarly, take a slice of Y_2 with length $10\tau n$, and let it be Y_3. Compute $W = \mathsf{IP}(X_3, Y_3)$ which outputs $r = \Omega(n) \leq \tau n$ bits.
6. Take a slice of X_2 with length $40\tau n$, and let it be X_4. Use W and X_4 to do an alternating extraction protocol for $L = \log^* n$[6] rounds, and output $(R_1, \cdots, R_L) = \mathsf{IaExt}(X_4, W)$, where each S_i, R_i used in the alternating extraction has $\tau n/\log n$ bits.
7. Set $i = 1$ and let n_1 be the length of V_1, which is at most $8\tau n$. While $L < c \log n_i$ do the following: encode V_i to $\tilde{V}_i$ using an asymptotically good binary error correcting code. Cut R_i into $O(\log n_i)$ bits. Use the sampler from Theorem 71 and R_i to sample $\log n_i$ bits of $\tilde{V}_i$, let the sampled string be $\overline{V_i}$. Set $V_{i+1} = R_i \circ \overline{V_i}$ and let $i = i + 1$.
8. Finally, cut R_i into $O(\log n_i)$ bits. Use the sampler from Theorem 71 and R_i to sample $L - |R_i|$ bits of $\tilde{V}_i$, let the sampled string be $\overline{V_i}$. Set $\tilde{\alpha} = R_i \circ \overline{V_i}$ which has length L.

We have the following lemma.

▶ **Lemma 72.** *There are constants $0 < \tau, \mu < 1$ and $C > 1$ such that the following holds. Let (X, Y) be two independent $(n, (1-\tau)n)$ sources, and (X', Y') be their tampered versions. Assume that either the tampering function f on X or the tampering function g on Y has no fixed point. For any L such that $C \leq L \leq \frac{\mu n}{\log n}$, with probability $1 - 2^{-\Omega(L)}$ over the fixing of $(X_1, Y_1, \tilde{X}_2, \tilde{Y}_2, X_3, Y_3, X_4)$ and the tampered versions $(X_1', Y_1', \tilde{X}_2', \tilde{Y}_2', X_3', Y_3', X_4')$, we have that $\tilde{\alpha} \neq \tilde{\alpha}'$. Moreover, conditioned on these fixings, X and Y are independent, and the average conditional min-entropy of both X and Y is $(1 - O(\tau))n$.*

Proof. As usual we use letters with primes to denote the tampered versions of random variables. First note that both X_1 and Y_1 have min-entropy at least $2\tau n$, thus by Theorem 23, we have that

$$(Z, X_1) \approx_{2^{-\Omega(n)}} (U, X_1),$$

and

$$(Z, Y_1) \approx_{2^{-\Omega(n)}} (U, Y_1).$$

If $X_1 \neq X_1'$ or $Y_1 \neq Y_1'$ then we have $V_1 \neq V_1'$. Now consider the case where $X_1 \neq X_1'$ and $Y_1 \neq Y_1'$. In this case we have $Z = Z'$ and either $X_2 \neq X_2'$ or $Y_2 \neq Y_2'$. Without loss of generality assume that $X_2 \neq X_2'$. We can now first fix (X_1, X_1'). Note that conditioned on this fixing, $Z = Z'$ is a deterministic function of Y, and thus independent of (X_2, X_2'). The Reed-Solomon encoding of $\hat{X}_2$ and $\hat{X}_2'$ ensures that $\overline{X}_2$ and $\overline{X}_2'$ differ in at least $n - n_0 + 1 > 0.9n$ symbols. Thus, with probability $1 - 2^{-\Omega(n)} - 2^{-\Omega(r/\log n)} = 1 - 2^{-\Omega(n/\log n)}$ over Z, we have that $\tilde{X}_2 \neq \tilde{X}_2'$. Therefore, altogether with probability $1 - 2^{-\Omega(n/\log n)}$ over the fixing of $(X_1, Y_1, \tilde{X}_2, \tilde{Y}_2)$ and $(X_1', Y_1', \tilde{X}_2', \tilde{Y}_2')$ we have that $V_1 \neq V_1'$.

We now fix $(X_1, Y_1, \tilde{X}_2, \tilde{Y}_2)$ and $(X_1', Y_1', \tilde{X}_2', \tilde{Y}_2')$. Note that conditioned on this fixing, X and Y are independent. Moreover, the average conditional min-entropy of both X_3 and Y_3 is at least $15\tau n - \tau n - 2\tau n - 3\tau n = 9\tau n$. Thus by Theorem 23, we have that

$$(W, X_3) \approx_{2^{-\Omega(n)}} (U, X_3).$$

[6] Here by $\log^* n$ we mean the number of steps it takes to get down to c' by computing $n \to c \log n$ for some constants c, c'.

We ignore the error for now since this only adds $2^{-\Omega(n)}$ to the final error. We now fix (X_3, X_3'). Note that conditioned on this fixing, (W, W') is a deterministic function of (Y, Y'), and thus independent of (X, X'). Further, the average conditional min-entropy of X_4 is at least $40\tau n - \tau n - 2(15\tau n + \tau n) - 3\tau n = 4\tau n$. Thus by Lemma 27 we have that for any $0 \le j \le L-1$,

$$(W, W', \{R_1, R_1', \cdots, R_j, R_j'\}, R_{j+1}) \approx_{\epsilon'} (W, W', \{R_1, R_1', \cdots, R_j, R_j'\}, U),$$

where $\epsilon' = O(L2^{-\Omega(n/\log n)}) = 2^{-\Omega(n/\log n)}$. Since conditioned on the fixing of (W, W'), the random variables $\{R_i, R_i'\}$ are deterministic functions of (X, X') and independent of (Y, Y'), we also have that

$$(Y_3, Y_3', \{R_1, R_1', \cdots, R_j, R_j'\}, R_{j+1}) \approx_{\epsilon'} (Y_3, Y_3', \{R_1, R_1', \cdots, R_j, R_j'\}, U).$$

We now further fix (Y_3, Y_3'). Note that now we have fixed $(X_1, Y_1, \tilde{X}_2, \tilde{Y}_2, X_3, Y_3)$ and $(X_1', Y_1', \tilde{X}_2', \tilde{Y}_2', X_3', Y_3')$. Ignoring the error for now let's assume that $V_1 \neq V_1'$ (note that (V_1, V_1') are now fixed) and for any $0 \le j \le L-1$,

$$(\{R_1, R_1', \cdots, R_j, R_j'\}, R_{j+1}) = (\{R_1, R_1', \cdots, R_j, R_j'\}, U).$$

Let j be the index when the protocol executes step 8. We know that $j \le L$ since in each step the length of the string V_i goes from n_i to $O(\log n_i)$. We have the following observation. For any $1 \le i \le j$, we have that V_i is a deterministic function of $(R_1, \cdots, R_{i-1})$; similarly, V_i' is a deterministic function of $(R_1', \cdots, R_{i-1}')$. Next, we have the following claim.

▷ Claim 73. For any $1 \le i < j$, suppose that conditioned on the fixing of $(R_1, \cdots, R_{i-1}), (R_1', \cdots, R_{i-1}')$ we have $V_i \neq V_i'$, then with probability $1 - 2^{-\Omega(\log n_i)}$ over the further fixing of (R_i, R_i'), we have $V_{i+1} \neq V_{i+1}'$. Suppose that conditioned on the fixing of $(R_1, \cdots, R_{j-1}), (R_1', \cdots, R_{j-1}')$ we have $V_j \neq V_j'$, then with probability $1 - 2^{-\Omega(L)}$ over the further fixing of (R_j, R_j'), we have $\tilde{\alpha} \neq \tilde{\alpha}'$.

Proof of the claim. Suppose that conditioned on the fixing of $(R_1, \cdots, R_{i-1}), (R_1', \cdots, R_{i-1}')$ we have $V_i \neq V_i'$. Note that now (V_i, V_i') is also fixed. We know that R_i is still uniform. Again, we have two cases. First, if $R_i \neq R_i'$, then we definitely have $V_{i+1} \neq V_{i+1}'$. Otherwise, we have $R_i = R_i'$. The encoding of V_i and V_i' ensures that at least a constant fraction of bits in $\tilde{V}_i$ and $\tilde{V}_i'$ are different. Thus by Theorem 71 with probability $1 - 2^{-\Omega(\log n_i)}$ over the further fixing of (R_i, R_i'), we have that $\overline{V_i} \neq \overline{V_i}'$ and thus $V_{i+1} \neq V_{i+1}'$.

For the case of $i = j$, the argument is the same, except now we are sampling $L - O(\log n_j)$ bits, and the probability that $\overline{V_i} \neq \overline{V_i}'$ is $2^{-\Omega(L - O(\log n_j))} = 2^{-\Omega(L)}$ since $L \ge c \log n_j$. ◁

Now we are basically done. Since we start with $V_1 \neq V_1'$, at the end the probability that $\tilde{\alpha} \neq \tilde{\alpha}'$ is at least

$$\Pi_{i=1}^{j-1}(1 - 2^{-\Omega(\log n_i)}) \cdot (1 - 2^{-\Omega(L)}).$$

Note that for any $1 \le i < j$ we have $n_{i+1} = O(\log n_i)$, so $2^{-\Omega(\log n_i)} \le 2^{-\Omega(\log n_i)}/2$. Thus the terms $2^{-\Omega(\log n_i)}$ form at least a geometric expression and hence this probability is at least $1 - O(2^{-\Omega(L)}) = 1 - 2^{-\Omega(L)}$. Adding back all the errors, and noticing that $C \le L \le \frac{\mu n}{\log n}$ for some properly chosen constants C and μ, the final error is still $1 - 2^{-\Omega(L)}$. Moreover, since the size of each random variable in $(X_1, Y_1, \tilde{X}_2, \tilde{Y}_2, X_3, Y_3, X_4)$ is at most $O(\tau n)$, conditioned on the fixing of $(X_1, Y_1, \tilde{X}_2, \tilde{Y}_2, X_3, Y_3, X_4)$ and the tampered versions $(X_1', Y_1', \tilde{X}_2', \tilde{Y}_2', X_3', Y_3', X_4')$, the average conditional min-entropy of both X and Y is $(1 - O(\tau))n$. ◀

We now use the above advice generator to give a new construction of non-malleable two-source extractors. Let (X, Y) be two independent $(n, (1-\gamma)n)$ sources with $\gamma \leq \tau$ where τ is the constant in Lemma 72.

- Let AdvGen be the advice generator from Lemma 72 for some error ϵ_1.
- Let AdvCB be the correlation breaker with advice from Lemma 45 with error some ϵ_2, using the merger from Lemma 36.
- Let IExt be the invertible linear seeded extractor form Theorem 68.

1. Compute $\tilde{\alpha} = \mathsf{AdvGen}(X, Y)$.
2. Consider the unused part of X. Divide it into (X_5, X_6, X_7) where X_5, X_6 has length $\alpha n, \beta n$ for some constants $\beta > \alpha > \gamma$, and X_7 is the rest of X with length at least $n/2$. Similarly, divide the unused part of Y into (Y_5, Y_6, Y_7) where Y_5, Y_6 has length $\alpha n, \beta n$ and Y_7 is the rest of Y with length at least $n/2$ (this can be ensured by choosing α, β, γ to be small enough).
3. Compute $V = \mathsf{AdvCB}(X_5, Y_5, \tilde{\alpha})$ which outputs $d = O(\log(n/\epsilon_2))$ bits.
4. Finally compute $W = \mathsf{IExt}(Y_6, V)$ which outputs $\Omega(n)$ bits.

We need the following proposition.

▶ **Proposition 74** ([18]). *Let D and D' be distributions over the same finite space Ω, and suppose they are ϵ-close to each other. Let $E \subseteq \Omega$ be any event such that $D(E) = p$. Then, the conditional distributions $D|E$ and $D'|E$ are (ϵ/p)-close.*

We now have the following theorem.

▶ **Theorem 75.** *Assume that either the tampering function f on X or the tampering function g on Y has no fixed point. There exist a constant $C > 1$ such that as long as $n \geq C\frac{\log\log(1/\epsilon_1)}{\log\log\log(1/\epsilon_1)}\log(n/\epsilon_2)$, the above non-malleable two-source extractor gives a non-malleable code with error $\epsilon_1 + O(\log(1/\epsilon_1)\sqrt{\epsilon_2})$ and rate $\Omega(\log(1/\epsilon_2)/n)$.*

Proof. First note that by Lemma 72, conditioned on the fixing of $H = (X_1, Y_1, \tilde{X}_2, \tilde{Y}_2, X_3, Y_3, X_4)$ and the tampered versions $H' = (X_1', Y_1', \tilde{X}_2', \tilde{Y}_2', X_3', Y_3', X_4')$, X and Y are independent, and the average conditional min-entropy of both X and Y is $(1 - O(\gamma))n$. If in addition we have that $\tilde{\alpha} \neq \tilde{\alpha}'$, then we will apply Lemma 45 and Lemma 36. Note that in order to set the error of the advice generator to be ϵ_1, we need to set the advice length to be $L = O(\log(1/\epsilon_1))$ by Lemma 72. Thus in Lemma 45 we need to merge $L = O(\log(1/\epsilon_1))$ rows.

Again, as in Theorem 55, we know that when we apply the correlation breaker to X_5 and Y_5, the entropy loss of both of them is $O(\gamma n)$. By choosing α, β, γ appropriately we can ensure that X_5 and Y_5 have sufficient entropy in them. We choose $a = 2$ in Lemma 36 and thus we obtain a correlation breaker with $m = O(\log(n/\epsilon_2))$, $d_1 = O(\log(n/\epsilon_2))$ and $d_2 = \log(n/\epsilon_2)2^{O(\sqrt{\log t})}$ where t is the parameter in Construction 44 with $t \leq L$. Note that this also satisfies that $d_1 \geq 4m$ and $m \geq c\log(d_2/\epsilon)$ as required by Lemma 45.

Now we need to ensure that

$$(\alpha - O(\gamma))n \geq c\frac{\log L}{\log t}\log(n/\epsilon_2) + max\{8\frac{\log L}{\log t}d_1, 2t \cdot d' + 4d_2\} + 5\ell + 4\log(1/\epsilon_2),$$

where $d' = O(\log(n/\epsilon_2))$. We choose $t = \frac{\log L}{\log\log L}$ and this gives us

$$n \geq C\frac{\log L}{\log\log L}\log(n/\epsilon_2),$$

for some constant $C > 1$. That is, we need

$$n \geq C \frac{\log\log(1/\epsilon_1)}{\log\log\log(1/\epsilon_1)} \log(n/\epsilon_2),$$

for some constants $C > 1$. As long as this condition is satisfied, conditioned on the event that $\tilde{\alpha} \neq \tilde{\alpha}'$, we have that

$$(V, V', H, H', X, X') \approx_{O(L\epsilon_2)} (U, V', H, H', X, X').$$

By choosing $\beta > \alpha$ appropriately, we can ensure that conditioned on the fixing of the previous random variables in the computation, Y_6 has entropy $\Omega(n)$ and (V, V') is a deterministic function of (X, X') and thus independent of (Y, Y'). Thus eventually we get

$$(W, W', H, H', X, X') \approx_{O(L\epsilon_2)} (U, W', H, H', X, X').$$

However, note that our construction is a two-source extractor itself. Thus, regardless of whether $\tilde{\alpha} \neq \tilde{\alpha}'$, we have that

$$(W, H, H', X, X') \approx_{O(L\epsilon_2)} (U, H, H', X, X').$$

We can cut the output length of the extractor to be $m = \Theta(\log(1/\epsilon_2))$ such that for any s in the support, we have $\Pr[U = s] = 2^{-m} = \sqrt{\epsilon_2}$. Thus we have for any s,

$$(H, H', X, X'|W = s) \approx_{O(L\sqrt{\epsilon_2})} (H, H', X, X'|U = s).$$

This means for any s,

$$(H, H', X, X'|W = s) \approx_{O(L\sqrt{\epsilon_2})} (H, H', X, X').$$

Let A be the event that $\tilde{\alpha} \neq \tilde{\alpha}'$. Note that $\Pr[A] \geq 1 - \epsilon_1$. Since A is determined by (H, H'), we have that for any s, $|\Pr[A|W = s] - \Pr[A]| \leq O(L\sqrt{\epsilon_2})$.

We now consider the probability distribution $(W'|W = s, A)$. This time we first condition on A. Note that conditioned on this event, we have

$$(W, W', H, H', X, X') \approx_{O(L\epsilon_2)} (U, W', H, H', X, X').$$

Thus again here we have that for any s,

$$(W', H, H', X, X'|W = s) \approx_{O(L\sqrt{\epsilon_2})} (W', H, H', X, X').$$

Therefore, we have for any s,

$$(W'|W = s, A) \approx_{O(L\sqrt{\epsilon_2})} (W'|A).$$

We can now bound the statistical distance between $(W'|W = s)$ and W'. We have

$$\begin{aligned}
&\left|(W'|W = s) - W'\right| \\
=&\left|(\Pr[A|W = s](W'|W = s, A) + \Pr[\bar{A}|W = s](W'|W = s, \bar{A})) - (\Pr[A](W'|A) + \Pr[\bar{A}](W'|\bar{A}))\right| \\
\leq&\left|\Pr[A]((W'|W = s, A) - W'|A)\right| + \left|(\Pr[A|W = s] - \Pr[A])(W'|W = s, A)\right| \\
&+ \left|\Pr[\bar{A}]((W'|W = s, \bar{A}) - (W'|\bar{A}))\right| + \left|(\Pr[\bar{A}|W = s] - \Pr[\bar{A}])(W'|W = s, \bar{A})\right| \\
\leq&\left|\Pr[A]((W'|W = s, A) - W'|A)\right| + \left|\Pr[\bar{A}]((W'|W = s, \bar{A}) - (W'|\bar{A}))\right| + O(L\sqrt{\epsilon_2}) \\
\leq&\left|((W'|W = s, A) - W'|A)\right| + \Pr[\bar{A}] \\
\leq&\epsilon_1 + O(L\sqrt{\epsilon_2}).
\end{aligned}$$

Note that the distribution of W' is a fixed probability distribution which is independent of s. Thus the construction gives a non-malleable code with error $\epsilon_1 + O(L\sqrt{\epsilon_2}) = \epsilon_1 + O(\log(1/\epsilon_1)\sqrt{\epsilon_2})$, and the rate of the code is $\Omega(\log(1/\epsilon_2)/n)$. ◀

We need to use the following simple inequality:

▶ **Fact 76.** *For any $0 < x \leq 1/3$, we have $1 - 3x \leq \frac{1-x}{1+x} < \frac{1+x}{1-x} \leq 1 + 3x$.*

We now have the following lemma, which gives a construction of non-malleable codes in the general case.

▶ **Lemma 77.** *Assume $\mathsf{2Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ satisfies the following conditions:*

- *It is a two-source extractor for entropy $n - \log(1/\epsilon')$ with error $\epsilon' \leq 2^{-(m+2)}$.*
- *It is a non-malleable two-source extractor for entropy $n - \log(1/\epsilon')$, which gives a non-malleable code in the two-split state model with error ϵ when either the tampering function f or the tampering function g has no fixed point.*

Then $\mathsf{2Ext}$ gives non-malleable code in the two-split state model with error $\epsilon + 2^{m+4}\epsilon'$.

Proof. Consider the tampering function $f : \{0,1\}^n \to \{0,1\}^n$ and $g : \{0,1\}^n \to \{0,1\}^n$. Let X and Y be two independent uniform distributions on $\{0,1\}^n$, let $p_0 = \Pr[f(X) = X]$, $q_0 = \Pr[g(Y) = Y]$, $p_1 = \Pr[f(X) \neq X] = 1 - p_0$ and $q_1 = \Pr[g(Y) \neq Y] = 1 - q_0$. Let the subsource of X conditioned on $f(X) = X$ be X_0, and the subsource of X conditioned on $f(X) \neq X$ be X_1. Thus $X = p_0X_0 + p_1X_1$. Similarly, we can define the subsources Y_0, Y_1 of Y such that $Y = q_0Y_0 + q_1Y_1$.

Consider the pairs of subsources (X_0, Y_0), (X_0, Y_1), (X_1, Y_0), and (X_1, Y_1), which have probability mass p_0q_0, p_0q_1, p_1q_0 and p_1q_1 respectively. Note that we have

$$(X, Y) = p_0q_0(X_0, Y_0) + p_0q_1(X_0, Y_1) + p_1q_0(X_1, Y_0) + p_1q_1(X_1, Y_1).$$

Let $W = \mathsf{2Ext}(X, Y)$. Consider any $s \in \{0,1\}^m$ and the uniform distribution on the pre-image of $W = s$ in (X, Y), call it Z_s. For any $i, j \in \{0,1\}$, let the subsource Z_{ij} stand for the uniform distribution on the pre-image of $W = s$ in (X_i, Y_j). Further let $r_{ij} = \Pr[\mathsf{2Ext}(X_i, Y_j) = s]$. Then we have

$$Z_s = \frac{\sum_{i,j} p_iq_jr_{ij}Z_{ij}}{\sum_{i,j} p_iq_jr_{ij}} = \sum_{i,j} \alpha_{ij}Z_{ij},$$

where $\alpha_{ij} = \frac{p_iq_jr_{ij}}{\sum_{i,j} p_iq_jr_{ij}}$.

We now have the following claim.

▷ Claim 78. For any $i, j \in \{0,1\}$, we have

- If either $p_i < \epsilon'$ or $q_j < \epsilon'$, then $\alpha_{ij} \leq 2^{m+1}\epsilon'$.
- Otherwise, $|\alpha_{ij}/(p_iq_j) - 1| \leq 2^{m+2}\epsilon'$

Proof of the claim. Note that $\sum_{i,j} p_iq_jr_{ij} = \Pr[W = s]$, and we have $\Pr[W = s] \geq 2^{-m} - \epsilon' > 2^{-(m+1)}$. Thus if either $p_i < \epsilon'$ or $q_j < \epsilon'$, we have

$$\alpha_{ij} = \frac{p_iq_jr_{ij}}{\sum_{i,j} p_iq_jr_{ij}} < 2^{m+1}\epsilon'.$$

Otherwise, both $p_i \geq \epsilon'$ and $q_j \geq \epsilon'$. This means that both X_i and Y_j have min-entropy at least $n - \log(1/\epsilon')$. Therefore we have $|r_{ij} - 2^{-m}| \leq \epsilon'$. Note that $\alpha_{ij}/(p_iq_j) = r_{ij}/\Pr[W = s]$ and we also have $.|\Pr[W = s] - 2^{-m}| \leq \epsilon'$. Since $\epsilon' \leq 2^{-(m+2)}$ by Fact 76 we have that

$$|\alpha_{ij}/(p_iq_j) - 1| \leq 2^{m+2}\epsilon'.$$ ◀

We now consider the distribution $\mathsf{2Ext}(T(Z_s))$, where for any distribution Z on $\{0,1\}^n \times \{0,1\}^n$, $T(Z)$ stands for the distribution $(f(x), g(y))$ where (x, y) is sampled from Z. Note that $\mathsf{2Ext}(Z_s)$ is fixed to s and and $\mathsf{2Ext}(T(Z_s))$ is the distribution of the decoded message after tampering. We have that $T(Z_s) = \sum_{i,j} \alpha_{ij} T(Z_{ij})$ and $\mathsf{2Ext}(T(Z_s)) = \sum_{i,j} \alpha_{ij} \mathsf{2Ext}(T(Z_{ij}))$. We will show that this distribution is close to the following distribution. For any $i, j \in \{0,1\}$ that are not both 0, if either $p_i < \epsilon'$ or $q_j < \epsilon'$, we define the distribution D_{ij} on $\{0,1\}^m$ to be a fixed constant (e.g., $\Pr[D_{ij} = 0^m] = 1$); otherwise since both X_i and Y_j have min-entropy at least $n - \log(1/\epsilon')$, $\mathsf{2Ext}$ gives a non-malleable code and thus $\mathsf{2Ext}(T(Z_{ij})$ is ϵ-close to a distribution D_{ij} independent of s. We let D_{00} be the distribution obtained by the identity function, i.e., for any s, D_{00} is fixed to be $I(s) = s$. We now claim that $\mathsf{2Ext}(T(Z_s))$ is close to the distribution $\sum_{i,j} p_i q_j D_{ij}$. We have

$$\left| \mathsf{2Ext}(T(Z_s)) - \sum_{i,j} p_i q_j D_{ij} \right| = \left| \sum_{i,j} \alpha_{ij} \mathsf{2Ext}(T(Z_{ij})) - \sum_{i,j} p_i q_j D_{ij} \right| \leq \sum_{i,j} \left| \alpha_{ij} \mathsf{2Ext}(T(Z_{ij})) - p_i q_j D_{ij} \right| .$$

For any $i, j \in \{0,1\}$, if either $p_i < \epsilon'$ or $q_j < \epsilon'$, we have the following bound.

$$|\alpha_{ij} \mathsf{2Ext}(T(Z_{ij})) - p_i q_j D_{ij}| \leq |\alpha_{ij} \mathsf{2Ext}(T(Z_{ij}))| + |p_i q_j D_{ij}| \leq 2^{m+1}\epsilon' + \epsilon' < 2^{m+2}\epsilon'.$$

Otherwise if i, j are not both 0 we have the following bound.

$$\begin{aligned} |\alpha_{ij} \mathsf{2Ext}(T(Z_{ij})) - p_i q_j D_{ij}| &\leq p_i q_j |\mathsf{2Ext}(T(Z_{ij})) - D_{ij}| + |(\alpha_{ij} - p_i q_j) \mathsf{2Ext}(T(Z_{ij}))| \\ &\leq p_i q_j \epsilon + 2^{m+2}\epsilon'. \end{aligned}$$

For the case of $i = j = 0$, we have that for any $(x, y) \in \mathsf{Supp}(Z_{00})$, $f(x) = x$ and $g(y) = y$. Thus $\mathsf{2Ext}(T(Z_{ij})) = s = D_{00}$ and we have

$$\begin{aligned} |\alpha_{ij} \mathsf{2Ext}(T(Z_{ij})) - p_i q_j D_{ij}| &\leq p_i q_j |\mathsf{2Ext}(T(Z_{ij})) - D_{ij}| + |(\alpha_{ij} - p_i q_j) \mathsf{2Ext}(T(Z_{ij}))| \\ &\leq 2^{m+2}\epsilon'. \end{aligned}$$

Therefore altogether we have

$$\left| \mathsf{2Ext}(T(Z_s)) - \sum_{i,j} p_i q_j D_{ij} \right| \leq \sum_{i,j} (p_i q_j \epsilon + 2^{m+2}\epsilon') = \epsilon + 2^{m+4}\epsilon'.$$

Since $\sum_{i,j} p_i q_j D_{ij}$ is obtained by $G(s)$ where G is a fixed probability distribution on the identity function and constant functions (the distribution of G only depends on f and g, but not on s), this implies that we have a non-malleable code in the 2 split-state model with error $\epsilon + 2^{m+4}\epsilon'$. ◀

We now have the following theorem.

▶ **Theorem 79.** *There are constants $0 < \eta, \mu < 1$ such that for any $n \in \mathcal{N}$ and $2^{-\frac{\mu n}{\log n}} \leq \epsilon \leq \eta$ there exists an explicit non-malleable code in the 2-split-state model with block length $2n$, rate $\Omega(\frac{\log\log\log(1/\epsilon)}{\log\log(1/\epsilon)})$ and error ϵ.*

Proof. We combine Theorem 75 and Lemma 77. Note that in Theorem 75, the construction is itself a two-source extractor for entropy $(1-\gamma)n$ with error $O(\log(1/\epsilon_1)\epsilon_2)$. To apply Theorem 75, we just need to ensure that

$$n \geq C \frac{\log\log(1/\epsilon_1)}{\log\log\log(1/\epsilon_1)} \log(n/\epsilon_2)$$

for some constant $C > 1$. We set $\epsilon_1 = \epsilon/2$ and $\epsilon_2 = 2^{-\Omega(\frac{\log\log\log(1/\epsilon)n}{\log\log(1/\epsilon)})}$. Note that

$$C\frac{\log\log(1/\epsilon_1)}{\log\log\log(1/\epsilon_1)}\log(n/\epsilon_2) = O(\frac{\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)}\log n) + O(\frac{\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)}\log(1/\epsilon_2)).$$

Since $2^{-\frac{\mu n}{\log n}} \leq \epsilon$ we have $\frac{\log\log(1/\epsilon)}{\log\log\log(1/\epsilon)}\log n = O(\frac{\log^2 n}{\log\log n})$. Thus we can set $\epsilon_2 = 2^{-\Omega(\frac{\log\log\log(1/\epsilon)n}{\log\log(1/\epsilon)})}$ to satisfy the inequality. Now we apply Lemma 77. We can set $\epsilon' = O(\log(1/\epsilon_1)\epsilon_2)$ since by Theorem 75 the construction is both a two-source extractor and a non-malleable two-source extractor for entropy $(1-\gamma)n$, and as long as $\epsilon \leq \eta$ for some appropriately chosen $\eta < 1$ we have $\log(1/\epsilon') \leq \gamma n$. Since in Theorem 75 we set the output of the extractor to be $m = \Theta(\log(1/\epsilon_2))$ such that $2^{-m} = \sqrt{\epsilon_2}$, we have that $\epsilon' \leq 2^{-(m+2)}$ and $2^{m+4}\epsilon' = O(\log(1/\epsilon_1)\sqrt{\epsilon_2})$. Thus by Lemma 77 the final error of the non-malleable code is

$$\epsilon_1 + O(\log(1/\epsilon_1)\sqrt{\epsilon_2}) + 2^{m+4}\epsilon' = \epsilon/2 + O(\log(1/\epsilon)\sqrt{\epsilon_2}).$$

Finally, notice that

$$\sqrt{\epsilon_2} = 2^{-\Omega(\frac{\log\log\log(1/\epsilon)n}{\log\log(1/\epsilon)})} \leq \alpha\frac{\epsilon}{\log(1/\epsilon)}$$

for any arbitrary constant $\alpha > 0$, since the latter is at least $\frac{1}{n}2^{-\frac{\mu n}{\log n}}$ and ϵ_2 is $2^{-\Omega(\frac{n\log\log n}{\log n})}$. Thus the final error of the non-malleable code is at most $\epsilon/2 + \epsilon/2 = \epsilon$, while the rate of the code, by Theorem 75, is $\Omega(\log(1/\epsilon_2)/n) = \Omega(\frac{\log\log\log(1/\epsilon)}{\log\log(1/\epsilon)})$. ◀

Next, we show how to achieve better error in the non-malleable two-source extractor and non-malleable codes. Recall that a bottleneck for error is the use of Reed-Solomon code in the construction. In order to get better error, we instead use a binary linear error correcting code and its generating matrix. It is easy to show using standard probabilistic argument that there exists a binary generating matrix that satisfies our requirements.

▶ **Theorem 80.** *There exists constants $0 < \alpha, \beta < 1$ such that for any $n \in \mathcal{N}$ there exists an $n \times m$ matrix over $\mathbf{F}_2$ with $n = \beta m$ which is the generating matrix of an asymptotically good code. Furthermore, Any sub-matrix formed by taking αn columns and the last $n/2$ rows has full column rank. In addition, for some $\epsilon = 2^{-O(n)}$, an ϵ-biased sample space over nm bits generates such a matrix with probability $1 - 2^{-\Omega(n)}$.*

Proof. We take an ϵ-biased sample space over nm bits for some $\epsilon = 2^{-O(n)}$. First, consider the sum of the rows over any non-empty subset of the rows. The sum is an m-bit string such that any non-empty parity is ϵ-close to uniform. Thus by the XOR lemma it is $2^{m/2}\epsilon$-close to uniform. We know a uniform m-bit string has weight $d = m/4$ with probability at least $1 - 2^{-\Omega(m)}$. Thus for this string the probability is at least $1 - 2^{-\Omega(m)} - 2^{m/2}\epsilon$. By a union bound the total failure probability is at most $2^n(2^{-\Omega(m)} + 2^{m/2}\epsilon) = 2^{-\Omega(n)}$ by an appropriate choice of β and $\epsilon = 2^{-O(n)}$.

Next, consider any sub-matrix formed by taking βm columns and the last $n/2$ rows, if it's truly uniform, then the probability that it has full column rank is at least $1 - \alpha n 2^{\alpha n - n/2} \geq 1 - 2^{-n/4}$ for $\alpha < 1/5$. Now by a union bound the total failure probability is at most

$$\binom{m}{\alpha n}(2^{-n/4} + \epsilon) \leq \left(\frac{em}{\alpha n}\right)^{\alpha n} 2^{-n/4+1} = \left(\frac{e}{\beta\alpha}\right)^{\alpha n} 2^{-n/4+1},$$

if we choose $\epsilon < 2^{-n/4}$. Note that for a fixed β, the quantity $(\frac{e}{\beta\alpha})^{\alpha}$ goes to 1 as α goes to 0. Thus we can choose α small enough such that this failure probability is also $2^{-\Omega(n)}$. Therefore altogether the failure probability is $2^{-\Omega(n)}$. ◀

Note that an ϵ-biased sample space over nm bits can be generated using $O(\log(nm/\epsilon)) = O(n)$ bits if $\epsilon = 2^{-O(n)}$. Now for any length $n \in \mathcal{N}$, we can compute the generating matrix (either using an ϵ-biased sample space or compute it deterministically in $2^{O(n)}$ time) once in the pre-processing step, and when we do encoding and decoding of the non-malleable code, all computation can be done in polynomial time.

Combining Theorem 67 and Theorem 55, we immediately obtain the following theorem.

▶ **Theorem 81.** *For any $n \in \mathcal{N}$ there exists a non-malleable code with efficient encoder-/decoder in the 2-split-state model with block length $2n$, rate $\Omega(\log\log n/\log n)$ and error $= 2^{-\Omega(n \log\log n/\log n)}$.*

8 Discussion and Open Problems

Several natural open problems remain here. The most intriguing one is how far we can push our new techniques. As mentioned above, one bottleneck here is that the computation of the merger is not a small space computation. If one can find a more succinct way to represent the computation, then it will certainly lead to further improvements (e.g., decrease the entropy requirement in two-source extractors to $O(\log n\sqrt{\log\log n})$). If in addition we can find a way to apply the recursive construction as in Nisan's generator [59], then it is potentially possible to decrease the entropy requirement in two-source extractors to $O(\log n \log\log\log n)$. We also believe our approach has the potential to eventually achieve truly optimal (up to constants) constructions. In addition, our techniques of treating the errors separately in non-malleable two-source extractors, may be useful in helping improve the rate of non-malleable codes for other classes of tampering functions (e.g., the affine tampering function and small depth circuits studied in [14]).

References

1. D. Aggarwal, Y. Dodis, T. Kazana, and M. Obremski. Non-malleable Reductions and Applications. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, 2015.
2. Divesh Aggarwal. Affine-evasive sets modulo a prime. Technical Report 2014/328, Cryptology ePrint Archive, 2014.
3. Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable Codes from Additive Combinatorics. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, 2014.
4. Boaz Barak, R. Impagliazzo, and Avi Wigderson. Extracting Randomness Using Few Independent Sources. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 384–393, 2004.
5. Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating Independence: New Constructions of Condensers, Ramsey Graphs, Dispersers, and Extractors. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 1–10, 2005.
6. Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2 Source Dispersers for $n^{o(1)}$ Entropy and Ramsey Graphs beating the Frankl-Wilson Construction. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006.
7. Avraham Ben-Aroya, Gil Cohen, Dean Doron, and Amnon Ta-Shma. Two-Source Condensers with Low Error and Small Entropy Gap via Entropy-Resilient Functions. Technical Report TR18-066, ECCC, 2018.

8 Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. Explicit two-source extractors for near-logarithmic min-entropy. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, 2017.

9 Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy Amplification by Public Discussion. *SIAM Journal on Computing*, 17(2):210–229, April 1988.

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

11 N. Chandran, B. Kanukurthi, R. Ostrovsky, and L. Reyzin. Privacy amplification with asymptotically optimal entropy loss. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, pages 785–794, 2010.

12 Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-Malleable Extractors and Codes, with their Many Tampered Extensions. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, 2016.

13 Eshan Chattopadhyay and Xin Li. Explicit Non-Malleable Extractors, Multi-Source Extractors and Almost Optimal Privacy Amplification Protocols. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, 2016.

14 Eshan Chattopadhyay and Xin Li. Non-Malleable Codes and Extractors for Small-Depth Circuits, and Affine Functions. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, 2017.

15 Eshan Chattopadhyay and David Zuckerman. Non-malleable Codes against Constant Split-State Tampering. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 306–315, 2014.

16 Eshan Chattopadhyay and David Zuckerman. Explicit Two-Source Extractors and Resilient Functions. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, 2016.

17 Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. In *ITCS*, pages 155–168, 2014.

18 Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable Coding against Bit-Wise and Split-State Tampering. In *TCC*, pages 440–464, 2014.

19 Benny Chor and Oded Goldreich. Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

20 Gil Cohen. Local Correlation Breakers and Applications to Three-Source Extractors and Mergers. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.

21 Gil Cohen. Making the Most of Advice: New Correlation Breakers and Their Applications. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, 2016.

22 Gil Cohen. Non-Malleable Extractors - New Tools and Improved Constructions. In *Proceedings of the 31st Annual IEEE Conference on Computational Complexity*, 2016.

23 Gil Cohen. Non-Malleable Extractors with Logarithmic Seeds. Technical Report TR16-030, ECCC, 2016.

24 Gil Cohen. Two-Source Extractors for Quasi-Logarithmic Min-Entropy and Improved Privacy Amplification Protocols. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, 2017.

25 Gil Cohen, Ran Raz, and Gil Segev. Non-Malleable Extractors with Short Seeds and Applications to Privacy Amplification. *SIAM Journal on Computing*, 43(2):450–476, 2014.

26 Gil Cohen and Leonard Schulman. Extractors for Near Logarithmic Min-Entropy. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, 2016.

27 Y. Dodis, J. Katz, L. Reyzin, and A. Smith. Robust Fuzzy Extractors and Authenticated Key Agreement from Close Secrets. In *Advances in Cryptology — CRYPTO '06, 26th Annual International Cryptology Conference, Proceedings*, pages 232–250, 2006.

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

29 Yevgeniy Dodis, Xin Li, Trevor D. Wooley, and David Zuckerman. Privacy Amplification and Non-Malleable Extractors Via Character Sums. *SIAM Journal on Computing*, 43(2):800–830, 2014.

30 Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 601–610, 2009.

31 Dean Doron, Pooya Hatami, and William Hoza. Near-Optimal Pseudorandom Generators for Constant-Depth Read-Once Formulas. Technical Report TR18-183, ECCC, 2018.

32 Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the Method of Multiplicities, with applications to Kakeya Sets and Mergers. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, 2009.

33 Zeev Dvir and Avi Wigderson. Kakeya sets, new mergers and old extractors. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008.

34 Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable Codes from Two-Source Extractors. In *CRYPTO (2)*, pages 239–257, 2013.

35 Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-Resilient Secret Sharing. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 227–237, Washington, DC, USA, 2007. IEEE Computer Society. doi:10.1109/FOCS.2007.35.

36 Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-Malleable Codes. In *ICS*, pages 434–452, 2010.

37 P. Erdős. Some remarks on the theory of graphs. *Bulletin of the American Mathematics Society*, 53:292–294, 1947.

38 Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, 2012.

39 Divya Gupta, Hemanta K. Maji, and Mingyuan Wang. Constant-rate Non-malleable Codes in the Split-state Model. Technical Report Report 2017/1048, Cryptology ePrint Archive, 2018.

40 Tom Gur and Igor Shinkar. An Entropy Lower Bound for Non-Malleable Extractors. Technical Report TR18-008, ECCC, 2018.

41 Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced Expanders and Randomness Extractors from Parvaresh-Vardy Codes. *Journal of the ACM*, 56(4), 2009.

42 Yael Kalai, Xin Li, and Anup Rao. 2-Source Extractors Under Computational Assumptions and Cryptography with Defective Randomness. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 617–628, 2009.

43 Yael Tauman Kalai, Xin Li, Anup Rao, and David Zuckerman. Network Extractor Protocols. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 654–663, 2008.

44 B. Kanukurthi and L. Reyzin. Key agreement from close secrets over unsecured channels. In *EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2009.

45 Bhavana Kanukurthi, Lakshmibhavana Obbattu, and Sruthi Sekar. Four-state Non-malleable Codes with Explicit Constant Rate. In *Fifteenth IACR Theory of Cryptography Conference*, 2017.

46 Xin Li. Improved Constructions of Three Source Extractors. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, pages 126–136, 2011.

47 Xin Li. Design Extractors, Non-Malleable Condensers and Privacy Amplification. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 837–854, 2012.

48 Xin Li. Non-Malleable Extractors, Two-Source Extractors and Privacy Amplification. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, pages 688–697, 2012.

49 Xin Li. Extractors for a Constant Number of Independent Sources with Polylogarithmic Min-Entropy. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 100–109, 2013.

50 Xin Li. New Independent Source Extractors with Exponential Improvement. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 783–792, 2013.

51 Xin Li. Non-Malleable Condensers for Arbitrary Min-Entropy, and Almost Optimal Protocols for Privacy Amplification. In *12th IACR Theory of Cryptography Conference*, pages 502–531. Springer-Verlag, 2015. LNCS 9014.

52 Xin Li. Three Source Extractors for Polylogarithmic Min-Entropy. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.

53 Xin Li. Improved Two-Source Extractors, and Affine Extractors for Polylogarithmic Entropy. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, 2016.

54 Xin Li. Improved Non-Malleable Extractors, Non-Malleable Codes and Independent Source Extractors. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, 2017.

55 C. J. Lu, Omer Reingold, Salil Vadhan, and Avi Wigderson. Extractors: Optimal up to Constant Factors. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 602–611, 2003.

56 Ueli M. Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In *Advances in Cryptology — CRYPTO '97, 17th Annual International Cryptology Conference, Proceedings*, 1997.

57 Raghu Meka. Explicit resilient functions matching Ajtai-Linial. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2015.

58 Raghu Meka, Omer Reingold, and Avishay Tal. Pseudorandom Generators for Width-3 Branching Programs. Technical Report TR18-112, ECCC, 2018.

59 Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12:449–461, 1992.

60 Noam Nisan and David Zuckerman. Randomness is Linear in Space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

61 Anup Rao. Extractors for a Constant Number of Polynomially Small Min-entropy Independent Sources. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006.

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

63 Renato Renner and Stefan Wolf. Unconditional authenticity and privacy from an arbitrarily weak secret. In *Advances in Cryptology — CRYPTO '03, 23rd Annual International Cryptology Conference, Proceedings*, pages 78–95, 2003.

64 Salil P. Vadhan. Constructing Locally Computable Extractors and Cryptosystems in the Bounded-Storage Model. *J. Cryptology*, 17(1):43–77, 2004. doi:10.1007/s00145-003-0237-x.