Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 58 (2023)

Explicit Directional Affine Extractors and Improved Hardness for Linear Branching Programs

Xin Li * Yan Zhong †

July 3, 2024

Abstract

Affine extractors give some of the best-known lower bounds for various computational models, such as AC^0 circuits, parity decision trees, and general Boolean circuits. However, they are not known to give strong lower bounds for read-once branching programs (ROBPs). In a recent work, Gryaznov, Pudlák, and Talebanfard (CCC' 22) introduced a stronger version of affine extractors known as directional affine extractors, together with a generalization of ROBPs where each node can make linear queries, and showed that the former implies strong lower bound for a certain type of the latter known as strongly read-once linear branching programs (SROLBPs). Their main result gives explicit constructions of directional affine extractors for entropy $k > 2n/3$, which implies average-case complexity $2^{n/3-o(n)}$ against SROLBPs with exponentially small correlation. A follow-up work by Chattopadhyay and Liao (CCC' 23) improves the hardness to $2^{n-o(n)}$ at the price of increasing the correlation to polynomially large, via a new connection to sumset extractors introduced by Chattopadhyay and Li (STOC' 16) and explicit constructions of such extractors by Chattopadhyay and Liao (STOC' 22). Both works left open the questions of better constructions of directional affine extractors and improved average-case complexity against SROLBPs in the regime of small correlation.

This paper provides a much more in-depth study of directional affine extractors, SROLBPs, and ROBPs. Our main results include:

- An explicit construction of directional affine extractors with $k = o(n)$ and exponentially small error, which gives average-case complexity $2^{n-o(n)}$ against SROLBPs with exponentially small correlation, thus answering the two open questions raised in previous works.
- An explicit function in AC^0 that gives average-case complexity $2^{(1-\delta)n}$ against ROBPs with negligible correlation, for any constant $\delta > 0$. Previously, no such average-case hardness is known, and the best size lower bound for any function in AC^0 against ROBPs is $2^{\Omega(n)}$.

One of the key ingredients in our constructions is a new linear somewhere condenser for affine sources, which is based on dimension expanders. The condenser also leads to an unconditional improvement of the entropy requirement of explicit affine extractors with negligible error. We further show that the condenser also works for general weak random sources, under the Polynomial Freiman-Ruzsa Theorem in F_2^n, recently proved by Gowers, Green, Manners, and Tao (arXiv' 23).

*Department of Computer Science, Johns Hopkins University, lixints@cs.jhu.edu. Supported by NSF CAREER Award CCF-1845349 and NSF Award CCF-2127575.

†Department of Computer Science, Johns Hopkins University, yzhong36@jhu.edu. Supported by NSF CAREER Award CCF-1845349.

ISSN 1433-8092

Contents

1 Introduction

Randomness extractors are functions that extract almost uniform random bits from weak random sources that have poor quality. Although the original motivation of randomness extractors comes from bridging the gap between the quality of randomness required in typical applications and that available in practice, as pseudorandom objects, they turn out to have broad applications in computer science. For example, the kind of extractors known as *affine extractors* are shown to be closely connected to complexity theory. Indeed, they give strong size lower bounds for AC^0 circuits (constant depth circuits with NOT gates and unbounded fan-in AND, OR gates) by the standard switching lemma [Hås86], and are shown to give exponential size lower bounds for DNF circuits with a bottom layer of parity gates, together with strong average-case hardness for parity decision trees [CS16]. Via sophisticated gate elimination techniques, they also give the best-known size lower bounds for general Boolean circuits [DK11, FGHK16, LY22]. We define affine extractors below.

Definition 1 (Affine extractor)**.** *An (n, k) affine source is the uniform distribution over some affine subspace with dimension k, of the vector space F_2^n.*[1] *A function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ is an affine extractor for entropy k with error ε if for every (n, k) affine source X, we have*

$$\mathsf{Ext}(X) \approx_\varepsilon U_m,$$

where U_m stands for the uniform distribution over $\{0,1\}^m$, and $\approx_\varepsilon$ means ε close in statistical distance. We say Ext is explicit if it is computable by a polynomial-time algorithm.

However, affine extractors are not known to imply strong lower bounds for computational models that measure space complexity. For example, a natural model in this context is a branching program, which is a directed acyclic graph with one source and two sinks, and each non-sink node has out-degree 2. To define the computation of the branching program, one marks each non-sink node with the index of an input bit, and labels the two outgoing edges by 0 and 1, respectively. Furthermore, one sink is labeled by 1 and the other is labeled by 0. The program now computes any input by following the natural path from the source to one sink, while reading the corresponding input bits and going through the corresponding edges. The program accepts the input if and only if the path ends in the sink with label 1, and the size of the branching program is defined as the number of its nodes, which roughly corresponds to $2^{O(s)}$ where s is the space complexity of the computation.

Proving non-trivial lower bounds of an explicit function for general branching programs turns out to be a challenging problem. The best known bound is $\Omega(\frac{n^2}{\log^2 n})$ [Nec66] after decades of effort, which is not enough to separate P from $\mathsf{LOGSPACE}$. Thus, most research on lower bounds for branching programs has focused on restricted models, and the most well-studied is the model of *read-once branching program*, where on any computational path, any input bit is read at most once. Exponential lower bounds are known in this model [Weg88, Zák84, Dun85, Juk88, KMW91, SS92, Pon98, Gál97, BW98, ABCR99, Kab03], however, it is not clear if affine extractors imply strong lower bounds here. For example, the inner product is a good affine extractor for any entropy $k > n/2$, but it can be computed by a read-once branching program of size $O(n)$.

In a recent work [GPT22], Gryaznov, Pudlák, and Talebanfard introduced a generalization of affine extractors called *directional affine extractors* and a generalization of standard read-once branching programs called *read-once linear branching programs*, and show that explicit constructions of the former imply strong lower bounds for certain cases of the latter. We define the two generalizations below.

[1] More generally, affine sources and affine extractors can be defined over any finite field, but in this paper we focus on the binary field F_2.

Definition 2 (Directional affine extractor)**.** *A function* $\mathsf{DAExt} : \{0,1\}^n \to \{0,1\}^m$ *is a directional affine extractor for entropy* k *with error* ε *if for every* (n,k) *affine source* X *and every non-zero vector* $a \in \mathsf{F}_2^n$*, we have*

$$(\mathsf{DAExt}(X), \mathsf{DAExt}(X+a)) \approx_\varepsilon (U_m, \mathsf{DAExt}(X+a)).$$

We say the function is a (zero-error) directional affine disperser if there exists some $b \in \{0,1\}^m$ *such that*

$$\left|\mathsf{Supp}\left(\mathsf{DAExt}(X) \mid \mathsf{DAExt}(X+a) = b\right)\right| = 2^m$$

.

Remark 1. *Our definition is slightly more general than the definition in [GPT22], since we allow the extractor to output more than one bits. In the special case of* $m = 1$*, our definition implies that in [GPT22], the reverse is also true up to a small loss in parameters as shown in [CL23].*

Definition 3 (Linear branching program [GPT22])**.** *A linear branching program on* F_2^n *is a directed acyclic graph* P *with the following properties:*

- *There is only one source* s *in* P*.*
- *There are two sinks in* P*, labeled with* 0 *and* 1 *respectively.*
- *Every non-sink node* v *is labeled with a linear function* $\ell_v : \mathsf{F}_2^n \to \mathsf{F}_2$*. Moreover, there are exactly two outgoing edges from* v*, one is labeled with* 1 *and the other is labeled with* 0*.*

The size of P *is the number of non-sink nodes in* P*.* P *computes a Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *in the following way. For every input* $x \in \mathsf{F}_2^n$*,* P *follows the computation path by starting from* s*, and when on a non-sink node* v*, moves to the next node following the edge with label* $\ell_v(x) \in \{0,1\}$*. The computation ends when the path ends at a sink, and* $f(x)$ *is defined to be the label on this sink.*

[GPT22] defines two kinds of read-once linear branching programs (ROLBP for short). Specifically, given any linear branching program P and any node v in P, let Pre_v denote the span of all linear queries that appear on any path from the source to v, excluding the query ℓ_v. Let Post_v denote the span of all linear queries in the subprogram starting at v.

Definition 4 (Weakly read-once linear branching program)**.** *A linear branching program* P *is weakly read-once if for every inner node* v *of* P*, it holds that* $\ell_v \notin \mathsf{Pre}_v$*.*

Definition 5 (Strongly read-once linear branching program)**.** *A linear branching program* P *is strongly read-once if for every inner node* v *of* P*, it holds that* $\mathsf{Pre}_v \cap \mathsf{Post}_v = \{0\}$*.*

In this paper, we will focus on strongly read-once linear branching programs, and use SROLBP as a shorthand. As observed in [GPT22] and [CL23], even the more restricted SROLBPs generalize several important and well-studied computational models, for example, decision trees, parity decision trees, and standard read-once branching programs. These models have applications in diverse areas, such as learning theory, streaming algorithms, communication complexity and query complexity. Thus, just as the natural generalizations from AC^0 circuits to $\mathsf{AC}^0[\oplus]$ circuits (AC^0 with parity gates), and from decision trees to parity decision trees, studying the generalization from ROBPs to ROLBPs is also a natural direction. In addition, as observed in [GPT22], parity decision trees are the only case in $\mathsf{AC}^0[\oplus]$ for which we have strong average-case lower bounds, and they are closely related to tree-like resolution refutation proof systems. Thus studying ROLBPs as a generalization of parity decision trees is of particular interest (in fact, this is the original motivation in [GPT22]). We now define two complexity measures of SROLBPs below.

Definition 6. *For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, let $\mathsf{SROLBP}(f)$ denote the smallest possible size of a strongly read-once linear branching program that computes f, and $\mathsf{SROLBP}_\varepsilon(f)$ denote the smallest possible size of a strongly read-once linear branching program P such that*

$$\mathbf{Pr}_{x \leftarrow_U \mathsf{F}_2^n}[P(x) = f(X)] \geq \frac{1}{2} + \varepsilon.$$

The definition can be adapted to ROBP*s naturally.*

The main contribution of [GPT22] is to show that directional affine extractors give strong average-case hardness for SROLBPs. Specifically, they show that for any directional affine extractor DAExt for entropy k with error ε, we have $\mathsf{SROLBP}_{\sqrt{\varepsilon/2}}(\mathsf{DAExt}) \geq \varepsilon 2^{n-k-1}$. In addition, they give an explicit construction of directional affine extractor for $k \geq \frac{2n}{3} + c$ with $\varepsilon \leq 2^{-c}$, which also implies exponential average-case hardness for SROLBPs of size up to $2^{\frac{n}{3}-o(n)}$. Thus, directional affine extractors are indeed stronger than standard affine extractors and give strong lower bounds in more computational models. [GPT22] left open the question of explicit constructions of directional affine extractors for $k = o(n)$.

In a follow-up work, Chattopadhyay and Liao [CL23] showed that another kind of extractors, known as *sumset extractors*, also give strong average-case hardness for SROLBPs. These extractors were introduced by Chattopadhyay and Li [CL16b], which are extractors that work for the sum of two (or more) independent weak random sources. By using existing constructions of such extractors in [CL22], they give an explicit function Ext such that $\mathsf{SROLBP}_{n^{-\Omega(1)}}(\mathsf{Ext}) \geq 2^{n-\log^{O(1)} n}$, i.e., the branching program size lower bound becomes close to optimal, but the correlation increases from exponentially small to polynomially large. Similarly, [CL23] left open the question of obtaining improved average-case hardness against SROLBPs in the small correlation regime.

We remark that directional affine extractors are a special case of *affine non-malleable extractors*, which are defined by Chattopadhyay and Li [CL17]. Roughly, an affine non-malleable extractor is an affine extractor such that the output is still close to uniform, even conditioned on the output of the extractor where the input affine source is modified by any affine function with no fixed points. In this context, directional affine extractors just correspond to the case where the tampering function adds a non-zero affine shift to the source. Previously, the best affine non-malleable extractor due to Li [Li23] works for entropy $k \geq (1-\gamma)n$ for some small constant $\gamma < 1/3$ with error $2^{-\Omega(n)}$. Thus this does not give a better construction of directional affine extractors. However, [Li23] does give an improved sumset extractor, which yields an explicit function Ext such that $\mathsf{SROLBP}_\varepsilon(\mathsf{Ext}) \geq 2^{n-O(\log n)}$ for any constant $\varepsilon > 0$, i.e., the branching program size lower bound becomes optimal up to the constant in $O(.)$, but the correlation increases to any constant.

1.1 Our Results

In this paper, we present a much more in-depth study of directional affine extractors, affine non-malleable extractors, SROLBPs, and standard ROBPs. To begin with, we observe that it is not a priori clear that SROLBPs are more powerful than standard ROBPs. Indeed, it is easy to see that $\mathsf{AC}^0[\oplus]$ and parity decision trees are exponentially more powerful than AC^0 circuits and standard decision trees, respectively, since parity requires exponential size AC^0 circuits and decision trees. However, any parity function can be computed by an ROBP of size $O(n)$. Nevertheless, there are previous works [Oko93, Juk95, GI17] which showed that computing explicit characteristic functions of certain affine subspaces require ROBPs of size $2^{\Omega(n)}$ (e.g., the satisfiable Tseitin formulas in [GI17]). Since such functions are easily computable by an SROLBP of size $O(n)$, this provides a

separation between SROLBP and ROBP and shows that indeed SROLBPs are exponentially more powerful than ROBPs.

In turn, this further demonstrates that directional affine extractors have stronger properties than standard affine extractors, as they imply strong lower bounds for SROLBPs. Next, we give explicit constructions of directional affine extractors with much better parameters than that in [GPT22]. Our construction works for any linear entropy with exponentially small error.

Theorem 1. *For any constant $0 < \delta \leq 1$, there exists a family of explicit directional affine extractors $\mathsf{DAExt} : \{0,1\}^n \to \{0,1\}^m$ for entropy $k \geq \delta n$ with error $\varepsilon = 2^{-\Omega(n)}$ and output length $m = \Omega(n)$.*

In fact, our construction can work for slightly sub-linear entropy.

Theorem 2. *There exists a constant $c > 1$ and an explicit family of directional affine extractors $\mathsf{DAExt} : \{0,1\}^n \to \{0,1\}^m$ for entropy $k \geq cn(\log\log\log n)^2/\log\log n$ with error $\varepsilon = 2^{-n^{\Omega(1)}}$ and output length $m = n^{\Omega(1)}$, as well as an explicit family of directional affine dispersers for entropy $k \geq cn(\log\log n)^2/\log n$ with $m = n^{\Omega(1)}$.*

This theorem immediately gives much improved average-case hardness for SROLBPs.

Theorem 3. *There is an explicit function DAExt such that $\mathsf{SROLBP}_{2^{-n^{\Omega(1)}}}(\mathsf{DAExt}) \geq 2^{n-\tilde{O}(\frac{n}{\log\log n})}$, where $\tilde{O}(.)$ hides $(\log\log\log n)^2$ factors.*

In particular, we can achieve exponentially small correlation while obtaining a $2^{n-o(n)}$ size lower bound for SROLBPs, which is almost optimal. This significantly improves the $2^{n/3-o(n)}$ size lower bound in [GPT22] and the polynomially large correlation in [CL23]. Thus, Theorem 2 and 3 provide positive answers to the two open questions in [GPT22] and [CL23] mentioned before.

We remark that under our new definition, a directional affine extractor is strictly stronger than a standard affine extractor. Thus Theorem 2 also improves the entropy requirement of negligible error affine extractors, from the previously best-known result of $\frac{n}{\sqrt{\log\log n}}$ [Yeh11, Li11] to $\frac{cn(\log\log\log n)^2}{\log\log n}$.

We also revisit the hardness results for standard ROBPs. As mentioned before, exponential and even close to optimal size lower bounds are known for explicit functions in this model, where the current best result is an explicit function that requires ROBPs (in fact, SROLBPs) of size $2^{n-O(\log n)}$ [Li23]. However, there has also been a lot of interest in finding functions in lower complexity classes that give strong lower bounds for ROBPs. It is clear that the class NC^0 is not sufficient. Thus the next possible class is AC^0. Indeed there are previous works giving explicit AC^0 functions that require ROBPs of size $2^{\Omega(\sqrt{n})}$[Juk88, KMW91, Gál97, BW98] and even $2^{\Omega(n)}$ [GI17], yet there is no average-case hardness as far as we know. Here, we improve both the size lower bound and the average-case hardness by giving an explicit AC^0 function that has negligible correlation with ROBPs of size $2^{(1-\delta)n}$ for any constant $\delta > 0$.

Theorem 4. *For any constant $\delta > 0$ there is an explicit function AC^0-Ext in AC^0 such that $\mathsf{ROBP}_{2^{-\mathrm{poly}\log n}}(\mathsf{AC}^0\text{-Ext}) \geq 2^{(1-\delta)n}$.*

One of the key ingredients in our constructions is a new linear somewhere condenser for affine sources. Specifically, we have

Definition 7. *For any $0 < \delta < \gamma < 1$, a function $\mathsf{SCond} : \mathsf{F}_2^n \to (\mathsf{F}_2^m)^\ell$ is a (δ, γ) affine somewhere condenser, if it satisfies the following property: for any affine source X over F_2^n with entropy δn, let $(Y_1, \cdots, Y_\ell) = \mathsf{SCond}(X) \in (\mathsf{F}_2^m)^\ell$, then there exists at least one $i \in [\ell]$ such that Y_i is an affine source over F_2^m with entropy at least γm.*

Theorem 5. *There exists a constant $\beta > 0$ such that for any $0 < \delta \leq 1/2$, there is an explicit $(\delta, 1/2 + \beta)$ affine somewhere condenser $\mathsf{SCond} : \mathsf{F}_2^n \to (\mathsf{F}_2^m)^t$, where $t = \mathsf{poly}(1/\delta)$ and $m = n/\mathsf{poly}(1/\delta)$. Moreover, SCond is a linear function.*

We further show that (a slight modification of) this condenser works for general weak random sources, under the well-known Polynomial Freiman-Ruzsa Theorem in F_2^n, once one of the most important conjectures in additive combinatorics and very recently proved by Gowers, Green, Manners, and Tao [GGMT23]. See section 4 for details.

Previously, all condensers of this kind are based on sum-product theorems, and the function is a polynomial with degree $\mathsf{poly}(1/\delta)$ [BKS+05, Raz05, Zuc07]. In contrast, there exist constructions of linear *seeded* extractors, where if one lists the outputs of the extractor for all possible seeds, then we get a somewhere random source such that at least one output is close to uniform, and the function is a linear function. However, in many applications such as ours, one needs to use a somewhere condenser instead of simply listing all outputs of an extractor, since the former only gives a small number (e.g., a constant) of outputs as opposed to $\mathsf{poly}(n)$ outputs from the extractor. Hence, our linear somewhere condenser complements the existing sum-product theorem based somewhere condensers. Moreover, our construction of the condenser is based on *dimension expanders*, which are algebraic pseudorandom objects previously studied based on their own interests, with no clear applications in computer science as far as we know. Thus, our construction can be viewed as one of the first applications of dimension expanders in computer science.

Finally, we study the question of whether directional affine extractors can give strong lower bounds for the class of $\mathsf{AC}^0[\oplus]$ in a black box way. Cohen and Tal [CT15] showed via probablistic methods that standard affine extractors do not suffice since depth-3 $\mathsf{AC}^0[\oplus]$ circuits can compute optimal affine extractors. Using a slightly modified argument as that in [CT15], we show that even the stronger version of directional affine extractors does not suffice. Specifically, depth-3 $\mathsf{AC}^0[\oplus]$ circuits can also compute optimal directional affine extractors. This in turn provides a strong separation of $\mathsf{AC}^0[\oplus]$ from SROLBP.

Theorem 6. *There exists a function $f : \{0,1\}^n \to \{0,1\}$ which is a directional affine extractor for entropy k with error ε, where $k = \log \frac{n}{\varepsilon^2} + \log\log \frac{n}{\varepsilon^2} + O(1)$ such that the following properties hold.*

1. *f is a polynomial of degree $\log \frac{n}{\varepsilon^2} + \log\log \frac{n}{\varepsilon^2} + O(1)$.*
2. *f can be realized by a $\mathsf{XOR\text{-}AND\text{-}XOR}$ circuit of size $O((n/\varepsilon)^2 \cdot \log^3(n/\varepsilon))$.*
3. *f can be realized by a De Morgan formula of size $O((n^5/\varepsilon^2) \cdot \log^3(n/\varepsilon))$.*

1.2 Overview of the Techniques

Here we give a sketch of the main ideas used in this paper. For clarity, we shall be informal at places and ignore some technical details.

Directional affine extractors. Our starting point is the construction of affine extractors by Li [Li11], which works for sub-linear entropy with exponentially small error. We first briefly recall the construction there. Divide an affine source X of entropy rate δ into $O(1/\delta)$ blocks. By choosing the size of the blocks appropriately, one can show that there exists a "good" block X_g of entropy rate $\Omega(\delta)$, and the source X still has a lot of entropy conditioned on X_g (i.e., we get an affine *block source*). If we know the position of X_g, randomness extraction is easy: we apply a somewhere condenser (e.g., those in [BKS+05, Raz05, Zuc07]) to condense X_g into a matrix with a constant number of rows, such that at least one row has entropy rate $1 - \delta/2$. At this point, we can apply

a linear two-source extractor (e.g., the inner product function) to each row of the matrix and the source X to get an affine *somewhere random* source, conditioned on the fixing of X_g. This is another matrix with a constant number of rows, such that at least one row is uniform, and one can apply existing techniques to deterministically extract random bits from this source [Rao09].

However, when δ is small, we don't know which block X_g is good. Thus in [Li11], the construction tries all blocks, and then combines them together. To make this process work, the construction crucially maintains the following property: (*) for each block X_i, the output bits produced from this block are constant degree polynomials of the input bits, and the degrees decrease geometrically from the first block to the last block. With this property, the analysis goes by focusing on the first good block X_g. Notice that we can fix all the outputs produced from blocks before X_g, while all outputs produced from blocks after X_g have degrees less than those from X_g. Thus if we take the XOR of all these outputs, an XOR lemma of polynomials [VW08, BKS+10] guarantees the final output is still close to uniform. We note that the XOR lemma of polynomials only works for degree up to $\log n$. Hence it is important to keep the degree c of the outputs from each block to be as small as possible. Roughly, we will need $c^{O(1/\delta)} < \log n$.

Our strategy now is to adapt this construction to directional affine extractors. Towards this, we use techniques from constructions of non-malleable extractors since, as we remark before, directional affine extractors are a special case of affine non-malleable extractors. Recent constructions of non-malleable extractors usually consist of two steps: first, generate a small advice that is different from the tampered version with high probability, and then use the advice together with other tools (e.g., correlation breakers) to achieve non-malleability. Thus, our goal is to adapt these two steps to directional affine extractors while, at the same time, still maintaining property (*), which is crucial to achieving any linear entropy or slightly sub-linear entropy. We now explain both steps.

As before, for each block X_i we will get an output U_i, which is close to uniform if X_i is a good block. Divide U_i into two parts $U_i = U_{i1} \circ U_{i2}$. We will use U_{i1} to generate the advice and U_{i2} for the rest of the construction. Notice that from the tampered input $X' = X + a$ we also have a tampered version $U'_i = U'_{i1} \circ U'_{i2}$. In the following, we will always use letters with prime to denote the corresponding random variables produced from the tampered input. If $U_{i1} \neq U'_{i1}$ then we are done, otherwise we use $U_{i1} = U'_{i1}$ to sample some $\Omega(\delta^2 n)$ bits H_i from an encoding of X, using an asymptotically good binary linear code. Since $X' = X + a$, we have that $H_i + H'_i$ basically corresponds to the sampled bits from the encoding of a. Thus $H_i \neq H'_i$ with high probability by the distance of the linear code. However, we cannot just do sampling naively since we need to keep the degree to be a constant. Therefore, we also divide both U_{i1} and the encoding of X into $\Omega(\delta^2 n)$ blocks where each block contains a constant number of bits, and use each block of U_{i1} to sample one bit from the corresponding block of the encoding of X. By the distance property of the code, there are $\Omega(\delta^2 n)$ blocks of the encoding of X and X' that are different. Thus we still have $H_i \neq H'_i$ with high probability, and now each bit of H_i is a constant degree polynomial of the bits of U_{i1} and X. The advice string is now $U_{i1} \circ H_i$.

Once we have the advice, we can append it to another string extracted from X by using a linear seeded extractor and U_{i2} as the seed. Now notice that the string produced from X is different from the string produced from X' with high probability, and they are linearly correlated conditioned on the fixing of (U_i, U'_i). Thus we can apply, for example, a known affine non-malleable extractor (the state-of-the-art affine non-malleable extractor with negligible error only works for high entropy). However, the known construction of affine non-malleable extractor in [CL17] has super constant degree. Indeed, even one application of this extractor results in a polynomial of degree larger than $\log n$, which already defeats our purpose to get a directional affine extractor (we can still get a directional affine disperser, though).

To solve this problem, we develop new ideas that make use of the special structure of $X' = X + a$.

Recall that in our construction, for every block X_i we get a U_{i2}, which is close to uniform if X_i is good, and X still has enough entropy conditioned on X_i. Our idea now is to use a *seeded non-malleable extractor* snmExt instead, which is an extractor with a uniform random seed, such that if an adversary tampers with the seed but not the source, then the output of the extractor on the original inputs is close to uniform given the output on the tampered inputs. By appending the advice string to U_{i2} and getting $\tilde{U}_i = U_i \circ H_i$, we have $\tilde{U}_i \neq \tilde{U}'_i$ with high probability, and the seed $\tilde{U}_i$ has high entropy if H_i has small size, which suffices for the seeded non-malleable extractor as long as the extractor is strong. Now, if the seeded non-malleable extractor is also *linear* conditioned on any fixing of the seed, then we have $\mathsf{snmExt}(X', \tilde{U}'_i) = \mathsf{snmExt}(X, \tilde{U}'_i) + \mathsf{snmExt}(a, \tilde{U}'_i)$. Since $\mathsf{snmExt}(X, \tilde{U}_i)$ is close to uniform given $\mathsf{snmExt}(X, \tilde{U}'_i)$, and the extractor is strong (we can fix the seeds $(\tilde{U}_i, \tilde{U}'_i)$), this implies that $\mathsf{snmExt}(X, \tilde{U}_i)$ is close to uniform given $\mathsf{snmExt}(X', \tilde{U}'_i)$. [2]

Luckily, there are previous constructions of linear seeded non-malleable extractors due to Li [Li12], which are based on the inner product function. Moreover, this extractor also has the property that each output bit is a constant degree polynomial of the input bits. Thus everything seems to work out, except for one problem: the non-malleable extractor in [Li12] only works when the source has entropy rate $> 1/2$, but here our goal is to work for any linear (or slightly sub-linear) entropy. A natural idea would be to use the somewhere condenser (e.g., in [BKS+05, Raz05, Zuc07]) to boost the entropy rate of X. However, all known condensers of this kind are based on sum-product theorems, which are non-linear functions, and applying them changes the structure of $X' = X + a$, which is important for our construction. Another idea is to apply a linear seeded extractor to X and try all possible seeds. This indeed keeps the structure of $X' = X + a$, but will result in a $\mathsf{poly}(n)$ number of outputs, and combining them together will result in a polynomial of large, super constant degree.

This motivates another key ingredient in our construction, a new linear somewhere condenser for affine sources. In short, we construct a linear function which, given any affine source on n bits with entropy rate $0 < \delta \leq 1/2$, outputs $\mathsf{poly}(1/\delta)$ rows such that each row has $n/\mathsf{poly}(1/\delta)$ bits, and at least one row has entropy rate $1/2 + \beta$ for some absolute constant $\beta > 0$. This complements the sum-product based somewhere condensers, and can be viewed as a separate contribution of our work. We will explain the construction of this condenser later, but finish the description of our directional affine extractor here, assuming that we have the linear somewhere condenser.

The rest of the construction roughly goes as follows. We apply the linear somewhere condenser to the source X to get a constant number of rows, then apply snmExt to each row using $\tilde{U}_i$ as the seed. Thus we get a constant number of outputs such that at least one of them is close to uniform conditioned on the corresponding tampered output. Now we apply an *affine correlation breaker* such as those in [Li17, CGL22, CL22] to further break the correlations between different outputs, and combine these outputs together by taking the XOR. The correlation breaker guarantees that the final output is close to uniform conditioned on the tampered output. To keep the degree small, we need to replace all seeded extractors used in the correlation breaker with a constant degree linear seeded extractor in [Li11]. This keeps the output bits to be constant degree polynomials of the input bits, and the remaining construction is essentially the same as that in [Li11].

Linear somewhere condenser. We now describe our construction of the linear somewhere condenser. This is based on another pseudorandom object known as *dimension expander*. Informally, a dimension expander is a set of linear mappings from a vector space F^n to itself, such that for any linear subspace $V \subset F^n$ with small dimension $k \leq n/2$, the span of the union of all the images of V

[2] The actual analysis involves more details since here X is not independent of $(\tilde{U}_i, \tilde{U}'_i)$, but the property still holds due to the affine structure. We omit the details here.

under the set of linear mappings has dimension at least $(1+\alpha)k$ for some absolute constant $\alpha > 0$. Readers familiar with expander graphs can see that this is a linear algebraic analog of expander graphs. Thus, it is desirable to give explicit constructions of the set of linear mappings which has as few number of mappings as possible, where this number d is called the degree. Dimension expanders were first introduced by Barak, Impagliazzo, Shpilka, and Wigderson [BISW04], who also showed the existence of such objects. Later, Bourgain and Yehudayoff [Bou09, BY13] gave explicit constructions of dimension expanders with degree $d = O(1)$ over any field. Interestingly, as far as we know, there are no previous applications of dimension expanders in computer science, and they are mainly studied based on their own interests and connections to other algebraic pseudorandom objects. Thus our construction can be viewed as one of the first applications of dimension expanders in computer science.

Given an explicit dimension expander $\{T_i\}_{i\in[d]}$ where each T_i is a linear mapping, and any affine source X with entropy rate $\delta \leq 1/2$, we first construct a basic somewhere condenser as follows. Divide X equally into $X = X_1 \circ X_2$, and our condenser produces $2d+2$ outputs: $(X_1, X_2, \{X_1 + T_i(X_2)\}_{i\in[d]}, \{T_i(X_1)+X_2\}_{i\in[d]})$. We show that at least one output has entropy rate $(1+\gamma)\delta$ for some constant $\gamma > 0$, and we give some intuition below. By the structure of affine sources, one can show that there exists another affine source X_3 independent of X_1 such that $X_2 = X_3 + L(X_1)$ for some linear function L. Let $H(X_1) = s$, $H(X_3) = r$ and $H(L(X_1)) = t$, then we have $s+r = \delta n$. If either s or r is small, e.g., $s \ll \delta n/2$, then we must have $r \gg \delta n/2$ and thus $H(X_2) = r+t \geq (1+\gamma)\delta n/2$. Therefore the entropy rate of X_2 is at least $(1+\gamma)\delta$. The case of $r \ll \delta n/2$ is similar. Hence, we only need to consider the case where $s \approx \delta n/2$ and $r \approx \delta n/2$, and notice that we must have either $s \leq \delta n/2$ or $r \leq \delta n/2$. Furthermore, in this case, t must be small, since otherwise, we would again have $H(X_2) = r + t \geq (1+\gamma)\delta n/2$.

For simplicity, assume that $s = r = \delta n/2$, and $t = 0$. Hence both X_1 and X_2 have entropy rate $\delta \leq 1/2$, and they are independent. Without loss of generality, assume the supports of both X_1 and X_2 are linear subspaces. By the property of the dimension expander, $\mathsf{Span}(\cup_{i\in[d]} T_i(X_1))$ has dimension at least $(1+\alpha)\delta n/2$. We now argue that there exists an $i \in [d]$ such that the support of $T_i(X_1)+X_2$ has dimension at least $(1+\alpha/d)\delta n/2$, which implies that $T_i(X_1)+X_2$ has entropy rate at least $(1+\alpha/d)\delta$. To see this, assume otherwise, then for any $i \in [d]$, any vector in the support of $T_i(X_1) + X_2$ can be expressed as a linear combination of the $r = \delta n/2$ basis vectors in the support of X_2 and $< (\alpha/d)\delta n/2$ other vectors. This implies that $\mathsf{Span}(\cup_{i\in[d]} T_i(X_1))$ has dimension $< \delta n/2 + d \cdot (\alpha/d)\delta n/2 = (1+\alpha)\delta n/2$, since any vector in $\mathsf{Span}(\cup_{i\in[d]} T_i(X_1))$ can be expressed as a linear combination of the $r = \delta n/2$ basis vectors in the support of X_2 and $< d \cdot (\alpha/d)\delta n/2$ other vectors. This contradicts the property of the dimension expander.

Thus, in all cases, we get the desired entropy rate boost. Our final somewhere condenser involves repeated uses of the basic condenser, as in previous works. It is easy to see that the entropy rate of at least one output will increase to $1/2 + \beta$ for some absolute constant $\beta > 0$ after $O(\log(1/\delta))$ uses of the basic condenser. The number of outputs is, therefore, $\mathsf{poly}(1/\delta)$ and each output has $n/\mathsf{poly}(1/\delta)$ bits. Finally, it is clear that the condenser is a linear function.

Once we have this linear condenser, we can even replace the somewhere condensers used in [Li11] by the new condenser. This further reduces the degree of the polynomials of the output bits (since previous somewhere condensers are polynomials instead of linear functions). Therefore we can push the entropy requirement of our directional affine extractor to be even better than that in [Li11], from $\frac{n}{\sqrt{\log\log n}}$ to $\frac{cn(\log\log\log n)^2}{\log\log n}$.

We show that a slight modification of our linear condenser also works for general weak random sources, under the Polynomial Freiman-Ruzsa Theorem. Roughly, the idea is to use a careful analysis of subsources and collision probability. Specifically, it is known that if the collision probability

of a distribution is small, then the distribution is close to having high min-entropy. On the other hand, if the collision probability is large, then (without loss of generality) assuming the distribution is the uniform distribution over some unknown subset, existing results in additive combinatorics imply that there is a large subset A in the support of the distribution such that the size of $A + A$ is not much larger than A. The Polynomial Freiman-Ruzsa Theorem then implies that there is another large subset $A' \subset A$ which is "close" to an affine subspace, which roughly reduces the analysis to the case of affine sources. See section 4 for the details.

AC^0 average-case hardness for ROBPs. To show AC^0 average-case hardness for ROBPs, we use a standard observation that if one conditions on an inner node, then the input bits prior to this node and the input bits after this node are still independent. We then construct an appropriate extractor in AC^0, which we call AC^0-Ext, for sources with such a structure. Specifically, given any ROBP of size s and any constant $\delta > 0$, we can find a cut or anti-chain (a maximal subset of vertices such that none of which is an ancestor of any other vertex) of size $O(s)$ at roughly depth δn above the sinks, so that conditioned on the fixing of any vertex in the cut, the input uniform random string X now becomes two independent weak sources A and B, where A corresponds to the first part of the program and B corresponds to the second part. Since we don't know the order of bits queried by the ROBP, the bits of the two sources are interleaved, and we view $X = A + B$. Using a standard averaging argument, one can show that with high probability, the following properties are satisfied: (1) A and B are supported on disjoint subsets of input bits; (2) A has min-entropy roughly $(1-\delta)n - \log s$ and B has min-entropy δn; and (3) B is an oblivious bit-fixing source, which is obtained by fixing some unknown bits in a uniform random string. If $s \leq 2^{(1-2\delta)n}$ then both A and B have entropy rate roughly δ. Now, our goal is to construct an extractor in AC^0 for sources with this structure, that is also *strong* in B. This means that even if we condition on the fixing of the vertex in the cut and B, the output of the extractor is still close to uniform. On the other hand, the output of the ROBP is completely determined by the vertex and B. Thus our extractor is average-case hard for ROBPs of size up to $2^{(1-2\delta)n}$.

As usual, the function AC^0-Ext will be compositions of different, more basic extractors as building blocks. Thus we need all these building blocks to be computable in AC^0. Here, we leverage the constructions from two previous works on extractors in AC^0: (1) the AC^0-computable extractors AC^0-BFExt for bit-fixing source by Cheng and Li [CL18], and (2) the AC^0-computable strong linear seeded extractors AC^0-LExt by Papakonstantinou, Woodruff, and Yang [PWY16].

Now we can describe our main idea of construction. Divide X into $t = O(1/\delta)$ blocks, and by an averaging argument, there exists a block B_g of B with entropy rate $\Omega(\delta)$. Now for the block $X_g = A_g + B_g$, we can fix A_g so that X_g is an oblivious bit-fixing source of entropy rate $\Omega(\delta)$ and is a deterministic function of B. We next fix the bits from B outside of the g-th block so that the source X outside of X_g is a deterministic function of A and thus independent of X_g. Moreover, A and X still have enough entropy left.

Applying the above-mentioned extractor AC^0-BFExt for bit-fixing sources to each block X_i, we convert X into a *somewhere random source* $Y = Y_1 \circ \cdots \circ Y_t$ where the row Y_g is a deterministic function of B_g and close to uniform, while all the other rows are deterministic functions of A. At this point, we can simply take the XOR of the Y_i's to obtain a close-to-uniform output. However, as mentioned before, we need the extractor to be strong in B and this simple approach is not sufficient. Instead, we fix all the outputs produced by AC^0-BFExt for X_i where $i \neq g$. Note that these are all deterministic functions of A. Thus conditioned on this fixing, Y becomes a deterministic function of B, which is independent of A. Moreover, as long as the output size of AC^0-BFExt is not too large, A still has enough entropy left. Since $X = A + B$, we can now apply a *strong* t-affine correlation

breaker as in [Li17, CL22] with each Y_i as the seed to extract from X a random string, and take the XOR of them. The property of the correlation breaker guarantees that the string produced from Y_g and X is close to uniform conditioned on all the other outputs and Y. Hence the XOR is also close to uniform conditioned on B. To ensure the correlation breaker is computable in AC^0, we replace all the strong (linear) seeded extractors in the known constructions of t-affine correlation breakers with the above-mentioned AC^0-LExt. Since $t = O(1/\delta)$ is a constant, the correlation breaker involves a constant number of compositions of AC^0-LExt, which is still in AC^0.

1.3 Organization of the Paper

The rest of the paper is organized as follows. In Section 2 we give some preliminary knowledge and some primitives from prior works. In Section 3 we describe our construction of linear somewhere random condenser for affine sources. Section 4 generalizes the construction to general weak sources under the Polynomial Freiman-Ruzsa Theorem. We give our construction of directional affine extractors in Section 5, and an AC^0 computable extractor against ROBP in Section 6. We present some open problems in Section 7. In the appendix we show that depth-3 $\mathsf{AC}^0[\oplus]$ circuits can compute optimal directional affine extractors, and give some omitted proofs.

2 Preliminaries

We often use capital letters for random variables and corresponding small letters for their instantiations. Let s, t be two integers, $\{V_1^1, V_1^2, \cdots, V_1^t, V_2^1, V_2^2, \cdots, V_2^t, \cdots, V_s^1, V_s^2, \cdots, V_s^t\}$ be a set of random variables. We use $V_i^{[t]}$ to denote the subset $\{V_i^1, \cdots, V_i^t\}$ and $V_{[s]}^j$ to denote the subset $\{V_1^j, \cdots, V_s^j\}$. We use $V_{[s]}^{[t]}$ as a shorthand for the whole set of random variables. We also use $i_{[t]}$ to denote the set of indices $\{i_1, i_2, \cdots, i_t\}$. 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. Let F_q denote the finite field of size q. All logarithms are to the base 2.

2.1 Probability Distributions and Entropy

Definition 8 (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 $\Delta(W, Z) \leq \varepsilon$. Let V also be a distribution on the set S. We sometimes use $W \approx_\varepsilon Z \mid V$ as a shorthand for $(W, V) \approx_\varepsilon (Z, V)$. We will use this two notations interchangeably throughout the paper. 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 7. *For any function α and two random variables A, B, we have $\Delta(\alpha(A), \alpha(B)) \leq \Delta(A, B)$.*

Definition 9 (Min-entropy)**.** *The* min-entropy *of a random variable X is defined as*

$$H_\infty(X) = \min_{x \in \mathsf{Supp}(X)}\{-\log \mathbf{Pr}[X = x]\}.$$

For a random variable $X \in \{0,1\}^n$, we say it is an (n, k)-source if $H_\infty(X) \geq k$. The entropy rate of X is defined as $H_\infty(X)/n$.

2.2 Somewhere Random Sources and Extractors

Definition 10 (Somewhere random sources)**.** *A source $X = (X_1, \cdots, X_t)$ is $(t \times r)$* somewhere-random *(SR-source for short) if each X_i takes values in $\{0,1\}^r$ and there is an i such that X_i is uniformly distributed.*

Definition 11. *An elementary somewhere-k-source is a vector of sources $(X_1, \cdots, X_t)$, such that some X_i is a k-source. A somewhere k-source is a convex combination of elementary somewhere-k-sources.*

Definition 12. *A function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(k \to \ell, \varepsilon)$-condenser if for every k-source X, $C(X, U_d)$ is ε-close to some ℓ-source. When convenient, we call C a rate-$(k/n \to \ell/m, \varepsilon)$-condenser.*

Definition 13. *A function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(k \to \ell, \varepsilon)$-somewhere-condenser if for every k-source X, the vector $(C(X,y)_{y \in \{0,1\}^d})$ is ε-close to a somewhere-ℓ-source. When convenient, we call C a rate-$(k/n \to \ell/m, \varepsilon)$-somewhere-condenser.*

Definition 14 (Seeded extractor)**.** *A function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a* strong (k, ε)-extractor *if for every source X with min-entropy k and independent Y which is uniform on $\{0,1\}^d$,*

$$(\mathsf{Ext}(X,Y), Y) \approx_\varepsilon (U_m, Y).$$

2.3 The Structure of Affine Sources

In this paper, affine sources encompass uniform distributions over linear subspaces and by affine functions we sometimes mean affine-linear functions.

Definition 15 (Affine source)**.** *Let F_q be the finite field with q elements. Denote by F_q^n the n-dimensional vector space over F_q. A distribution X over F_q^n is an $(n,k)_q$ affine source if there exist linearly independent vectors $a_1, \cdots, a_k \in \mathsf{F}_q^n$ and another vector $b \in \mathsf{F}_1^n$ s.t. X is sampled by choosing $x_1, \cdots, x_k \in \mathsf{F}$ uniformly and independently and computing*

$$X = \sum_{i=1}^{k} x_i a_i + b.$$

The min-entropy of affine source coincides with its standard Shannon entropy, we simply use $H(X)$ to stand for the entropy of an affine source X.

The following lemma is a slight generalization of its version in [Li11], where we show that L can be an affine function instead of just a linear function. We also prove that the entropy of X is constant conditioned on any fixing of $L(X)$. The readers are referred to Appendix B for a proof.

Lemma 8 (Affine conditioning [Li11])**.** *Let X be any affine source on $\{0,1\}^n$. Let $L : \{0,1\}^n \to \{0,1\}^m$ be any affine function. Then there exist independent affine sources A, B such that:*

- *$X = A + B$*
- *There exists $c \in \{0,1\}^m$, such that for every $b \in \mathsf{Supp}(B)$, it holds that $L(b) = c$.*
- *$H(A) = H(L(A))$ and there exists an affine function $L^{-1} : \{0,1\}^m \to \{0,1\}^n$ such that $A = L^{-1}(L(A))$.*
- *$H(X\,|_{L(X)=\ell}) = H(B)$ for all $\ell \in \mathsf{Supp}(L(X))$.*

The following definition is a specialization of conditional min-entropy for affine sources. It is well-defined by Lemma 8.

Definition 16 (Conditional min-entropy for affine sources)**.** *Let W and Z be two affine sources. Define*

$$H(W \mid Z) = H(W \mid_{Z=z}),\ \forall z \in \mathsf{Supp}(Z).$$

Lemma 9. *Let X, Y, Z be affine sources. Then $H(X \mid (Y,Z)) \geq H(X \mid Z) - \log(\mathsf{Supp}(Y))$.*

We will also need the following lemma from [Li11] when we do sequential conditioning on blocks of an affine source or argue about the total entropy of blocks of an affine source.

Lemma 10 (Affine entropy argument [Li11])**.** *Let X be any affine source on $\{0,1\}^n$. Divide X into t arbitrary blocks $X = X_1 \circ X_2 \circ \cdots \circ X_t$. Then there exists positive integers $k_1, \cdots, k_t$ such that,*

- *$\forall j, 1 \leq j \leq t$ and $\forall (x_1, \cdots, x_{j-1}) \in \mathsf{Supp}(X_1, \cdots, X_{j-1})$, $H(X_j \mid_{X_1 = x_1, \cdots, X_{j-1} = x_{j-1}}) = k_j$;*
- *$\sum_{i=1}^{t} k_i = H(X)$.*

2.4 Average Conditional Min-Entropy and Average-Case Seeded Extractors

Definition 17 (Average conditional min-entropy)**.** *The* average conditional min-entropy *is defined as*

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

Lemma 11 ([DORS08])**.** *For any $s > 0$, $\mathbf{Pr}_{w \leftarrow W}[H_\infty(X \mid W = w) \geq \widetilde{H}_\infty(X \mid W) - s] \geq 1 - 2^{-s}$.*

Lemma 12 ([DORS08])**.** *If a random variable B has at most 2^ℓ possible values, then $\widetilde{H}_\infty(A \mid B) \geq H_\infty(A) - \ell$.*

Lemma 13 ([DORS08])**.** *For any $\delta > 0$, if Ext is a (k, ε) extractor, then it is also a $(k + \log(1/\delta), \varepsilon + \delta)$ average case extractor.*

2.5 Alternating Extraction and Independence Merging

The following techniques underpin the construction of correlation breakers.

Definition 18 (L-alternating extraction)**.** *Let W be an (n_w, k_w)-source and $(Q_1, \cdots, Q_L)$ be L (n_q, k_q)-sources. Let $\mathsf{Ext}_q, \mathsf{Ext}_w$ be strong seeded extractors that extract s bits from sources with min-entropy k with error ε and seed length s. Let $S_1 = \mathsf{Slice}(Q_1, d)$ for some appropriate length d, $R_1 = \mathsf{Ext}_w(W, S_1), S_2 = \mathsf{Ext}_q(Q_2, R_1), \cdots, R_{L-1} = \mathsf{Ext}_w(W, S_{\ell-1}), S_L = \mathsf{Ext}_q(Q_L, R_{L-1})$, then L-alternating extraction$(Q_1, \cdots, Q_L, W) = S_L$.*

Lemma 14 (Look-ahead extractor [CGL16])**.** *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 bits from sources with min-entropy k with error*

ε and seed length s. Suppose (Y, Y') is independent of (W, W'), and $k_w, k_q \geq k + 2\ell s + 2\log(1/\varepsilon)$. Let laExt be the ℓ round look-ahead extractor using $\mathsf{Ext}_q, \mathsf{Ext}_w$, and $(R_1, \cdots, R_\ell) = \mathsf{laExt}_\ell(W, Y)$, $(R'_1, \cdots, R'_\ell) = \mathsf{laExt}_\ell(W', Y')$. Then for any $0 \leq j \leq \ell - 1$, we have

$$R_{j+1} \approx_{O(\ell\varepsilon)} U_s \mid (Y, Y', R_0, R'_0, \cdots, R_j, R'_j).$$

The following lemma captures an essential argument for the flip-flop and NIPM constructions, which are components of correlation breakers.

Lemma 15 (Independence-merging lemma [CGL22])**.** *Let $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be any (k, ε)-strong seeded extractor, $X, X^{[t]} \in \{0,1\}^n$, $Y, Y^{[t]} \in \{0,1\}^d$ such that $X, X^{[t]}$ are independent with $Y, Y^{[t]} \in \{0,1\}^d$, $W = \mathsf{Ext}(X, Y)$ and $W^j = \mathsf{Ext}(X^j, Y^j)$ for every $j \in [t]$. Suppose there exists $S, T \subseteq [t]$ such that*

- *$(Y, Y^S) \approx_\delta (U_d, Y^S)$;*
- *$\widetilde{H}_\infty(X \mid X^T, Z) \geq k + tm + \log(1/\varepsilon)$.*

Then

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

2.6 ε-Biased Space and XOR Lemmas

The tools in this subsection are utilized in [Li11] for their affine disperser and extractor constructions. We also adopt these techniques in our constructions of directional affine dispersers and extractors.

Definition 19 (ε-biased space)**.** *A random variable Z over $\{0,1\}$ is ε-biased if $|\mathbf{Pr}[Z = 0] - \mathbf{Pr}[Z = 1]| \leq \varepsilon$. A sequence of $0-1$ random variables $Z_1, \cdots, Z_m$ is ε-biased for linear tests if for any nonempty set $S \subset [m]$, the random variable $Z_S = \bigoplus_{i\in S} Z_i$ is ε-biased.*

Lemma 16 ([Vaz86])**.** *Let $Z_1, \cdots, Z_m$ be $0-1$ random variables that are ε-biased for linear tests. Then the distribution of $(Z_1, \cdots, Z_m)$ is $\varepsilon \cdot 2^{m/2}$-close to uniform.*

Definition 20. *For two functions $f, p : \{0,1\}^n \to \{0,1\}$, their correlation over the uniform distribution is defined as*

$$\mathsf{Cor}(f, p) = \Big|\mathbf{Pr}_x[f(x) = p(x)] - \mathbf{Pr}_x[f(x) \neq p(x)]\Big|,$$

where the probability is over the uniform distribution. For a class C of functions, we denote by $\mathsf{Cor}(f, C)$ the maximum of $\mathsf{Cor}(f, p)$ over all functions $p \in C$ whose domain is $\{0,1\}^n$.

Theorem 17 (XOR lemma for polynomials over F_2 [VW08, BKS+10])**.** *Let P_d stand for the class of all polynomials of degree at most d over F_2. Let $f : \{0,1\}^n \to \{0,1\}$ be a function such that $\mathsf{Cor}(f, P_d) \leq 1 - 2^{-d}$ and $f^{\oplus m}$ be the XOR of the value of f on m independent inputs. Then*

$$\mathsf{Cor}(f^{\oplus m}, P_d) \leq \mathsf{exp}(-\Omega(m/(4^d \cdot d))).$$

3 Linear Somewhere Condenser for Affine Sources

In this section we provide an explicit construction of a linear somewhere condenser for affine sources, or more conveniently, an affine somewhere condenser where each output is a linear function of the input. We begin with the definition.

Definition 21. *For any $0 < \delta < \gamma < 1$, a function $\mathsf{SCond} : \mathsf{F}_2^n \to (\mathsf{F}_2^m)^\ell$ is a (δ, γ) affine somewhere condenser, if it satisfies the following property: for any affine source X over F_2^n with entropy δn, let $(Y_1, \cdots, Y_\ell) = \mathsf{SCond}(X) \in (\mathsf{F}_2^m)^\ell$, then there exists at least one $i \in [\ell]$ such that Y_i is an affine source over F_2^m with entropy at least γm.*

We will prove the following theorem.

Theorem 18. *There exists a constant $\beta > 0$ such that for any $0 < \delta \leq 1/2$, there is an explicit $(\delta, 1/2 + \beta)$ affine somewhere condenser $\mathsf{SCond} : \mathsf{F}_2^n \to (\mathsf{F}_2^m)^t$, where $t = \mathsf{poly}(1/\delta)$ and $m = n/\mathsf{poly}(1/\delta)$. Moreover, SCond is a linear function.*

To prove the theorem we will use the following object known as a *dimension expander*.

Definition 22 (Dimension expander [BISW04, DS11])**.** *Let F be a field and let $T_1, \cdots, T_d : \mathsf{F}^n \to \mathsf{F}^n$ be linear mappings. The set $T = \{T_i\}_{i=1}^d$ is an α-dimension expander with degree d, if for every subspace $V \subset \mathsf{F}^n$ of dimension at most $n/2$ we have*

$$\dim\left(\sum_{i=1}^{d} T_i(V)\right) \geq (1+\alpha)\dim(V).$$

We say that T is explicit if there exists a $\mathsf{poly}(n)$-time algorithm that, on input n, outputs T.

Theorem 19 ([Bou09, BY13])**.** *There exist absolute constants $d \in \mathbb{N}$ and $0 < \alpha < 1$ such that over any field F, there exists an explicit family of α-dimension expanders with degree d.*

Given the above theorem we first provide a basic affine condenser:

Algorithm 1 $\mathsf{BasicCond}(x)$

Input: $x \in \mathsf{F}_2^n$ — an n bit string.
Output: $z \in (\mathsf{F}_2^m)^{2d+2}$ — an array of $2d+2$ bit strings with length m, where $m = n/2$ and d is the constant in Theorem 19.

Sub-Routines and Parameters:
Let $T = \{T_i\}_{i=1}^d$ be the α-dimension expander given by Theorem 19.

Divide x into 2 blocks $x = x_1 \circ x_2$ where each block has $n/2$ bits.
Let $z = z_1 \circ z_2 \circ \cdots \circ z_{2d+2}$, where $z_1 = x_1$, $z_2 = x_2$, and $z_{2i+1} = x_1 + T_i(x_2)$, $z_{2i+2} = x_2 + T_i(x_1)$, for any $i \in [d]$.

We will prove the following lemma.

Lemma 20. *For any $0 < \delta \leq 1/2$, $\mathsf{BasicCond}$ is a $(\delta, (1 + \frac{\alpha}{4d})\delta)$ affine somewhere condenser, where α, d are the constants in Theorem 19.*

Proof. Let X be any affine source over F_2^n with entropy $k = \delta n$. Without loss of generality, assume the support of X is a linear subspace V (if not, we can do the analysis for the corresponding linear subspace, and then add the affine shift, since we are always dealing with linear functions here). We start by giving a set of k base vectors for V. For this, consider the linear subspace $W \subseteq V$ s.t. the first $n/2$ bits of W are 0. Assume $\dim(W) = r$ and let $b_1, \cdots, b_r$ be a basis for W. Next, we extend these vectors to $b_1, \cdots, b_r, c_1, \cdots, c_s$ which form a complete basis for V, such that $s + r = k$.

Note that the vectors formed by the first $n/2$ bits of $\{c_i\}_{i=1}^s$ are also linearly independent, otherwise some linear combination of them will be in W. Let $\{\bar{c}_i\}_{i=1}^s$ be the first $n/2$ bits of $\{c_i\}_{i=1}^s$, and $\{\tilde{c}_i\}_{i=1}^s$ be the second $n/2$ bits of $\{c_i\}_{i=1}^s$. Similarly, let $\{\tilde{b}_i\}_{i=1}^r$ be the second $n/2$ bits of $\{b_i\}_{i=1}^r$ (recall the first $n/2$ bits are 0).

Now, let $\mathcal{Q} \subseteq [s]$ be such that $(\{\tilde{b}_i\}_{i=1}^r, \{\tilde{c}_i\}_{i\in\mathcal{Q}})$ form a basis of the supporting linear subspace of X_2. Let $C = \mathrm{span}(\{\tilde{c}_i\}_{i\in\mathcal{Q}})$. The source X is sampled by picking a uniform random vector $Y = (Y_1, \cdots, Y_k) \in \mathsf{F}_2^k$ and computing

$$\sum_{i=1}^{s} Y_i c_i + \sum_{j=1}^{r} Y_{s+j} b_j = \sum_{i=1}^{s} Y_i(\bar{c}_i, \tilde{c}_i) + \sum_{j=1}^{r} Y_{s+j}(0, \tilde{b}_j).$$

Thus the first $n/2$ bits are given by $\sum_{i=1}^s Y_i \bar{c}_i$, while the second $n/2$ bits are given by

$$\sum_{i=1}^{s} Y_i \tilde{c}_i + \sum_{j=1}^{r} Y_{s+j} \tilde{b}_j = \sum_{i\in\mathcal{Q}} Y_i \tilde{c}_i + \sum_{i\in[s]\setminus\mathcal{Q}} Y_i \tilde{c}_i + \sum_{j=1}^{r} Y_{s+j} \tilde{b}_j.$$

Note that for any $i \in [s] \setminus \mathcal{Q}$, $\tilde{c}_i$ can be expressed as a linear combination of $(\{\tilde{b}_i\}_{i=1}^r, \{\tilde{c}_i\}_{i\in\mathcal{Q}})$. Let $\overline{Y} = (\{Y_i\}_{i\in[s]\setminus\mathcal{Q}})$, then the above can be written as

$$\sum_{i\in\mathcal{Q}} (Y_i + L_i(\overline{Y}))\tilde{c}_i + \sum_{j=1}^{r} (Y_{s+j} + L_j(\overline{Y}))\tilde{b}_j,$$

where each L_i or L_j is a linear function from $\mathsf{F}_2^{s-|\mathcal{Q}|}$ to F_2.

It is easy to see that the k random bits $(\{Y_i\}_{i=1}^s, \{Y_{s+j} + L_j(\overline{Y})\}_{j=1}^r)$ are independent and uniform (in particular, any non-trivial parity of these bits is a uniform random bit). Similarly, the random bits $(\{Y_i + L_i(\overline{Y})\}_{i\in\mathcal{Q}}, \{Y_{s+j} + L_j(\overline{Y})\}_{j=1}^r)$ are also independent and uniform. Let $A = \mathrm{span}(\{\bar{c}_i\}_{i=1}^s)$, $B = \mathrm{span}(\{\tilde{b}_i\}_{i=1}^r)$, and $C = \mathrm{span}(\{\tilde{c}_i\}_{i\in\mathcal{Q}})$. So $\dim(A) = s$, $\dim(B) = r$, and let $\dim(C) = |\mathcal{Q}| = t$. By the above calculation, we have $H(X_1) = \dim(A) = s$, $H(X_2) = \dim(B) + \dim(C) = r + t$. Furthermore, let $X_3 = \sum_{j=1}^r (Y_{s+j} + L_j(\overline{Y}))\tilde{b}_j$ and $X_4 = \sum_{i\in\mathcal{Q}} (Y_i + L_i(\overline{Y}))\tilde{c}_i$, then X_3 is the uniform distribution over B and X_4 is the uniform distribution over C. Thus $H(X_3) = r$ and $H(X_4) = t$. We know $X_1 = \sum_{i=1}^s Y_i \bar{c}_i$. Thus X_1 and X_3 are independent, while X_4 is a deterministic function of X_1 (hence also independent of X_3). Note that $X_2 = X_3 + X_4$, and $X = (X_1, X_2) = (X_1, X_3 + X_4)$.

Note that $s + r = k = \delta n$. If $s \geq (\frac{1}{2} + \frac{\alpha}{8d})k$, then $H(X_1) = s \geq (\frac{1}{2} + \frac{\alpha}{8d})k = (1 + \frac{\alpha}{4d})\delta(n/2)$. Similarly, if $r \geq (\frac{1}{2} + \frac{\alpha}{8d})k$, then $H(X_2) = r + t \geq r \geq (1 + \frac{\alpha}{4d})\delta(n/2)$. In either case, we are done. Otherwise, we must have $s < (\frac{1}{2} + \frac{\alpha}{8d})k$ and $r < (\frac{1}{2} + \frac{\alpha}{8d})k$, which in turn implies that $s > (\frac{1}{2} - \frac{\alpha}{8d})k$ and $r > (\frac{1}{2} - \frac{\alpha}{8d})k$. Now if $t \geq \frac{\alpha}{4d}k$, then $H(X_2) = r + t > (\frac{1}{2} + \frac{\alpha}{8d})k = (1 + \frac{\alpha}{4d})\delta(n/2)$, and again we are done.

The only case left is when $(\frac{1}{2} - \frac{\alpha}{8d})k < s, r < (\frac{1}{2} + \frac{\alpha}{8d})k$ and $t < \frac{\alpha}{4d}k$. Since $s + r = k$, one of them must be at most $k/2 = \delta n/2$. We have two cases.

Case 1. $(\frac{1}{2} - \frac{\alpha}{8d})k < s \leq k/2$. In this case, $\dim(A) = s \leq \delta(n/2) \leq (1/2) \cdot (n/2)$. Consider the d linear mappings $\{T_i\}_{i=1}^d$ given by the dimension expander of Theorem 19. Note that $(1 + \alpha)s > (1 + \alpha)(\frac{1}{2} - \frac{\alpha}{8d})k > (\frac{1}{2} + \frac{\alpha}{8d})k > r$. We have the following claim.

Claim 21. *There exists an $i \in [d]$ such that* $\dim(T_i(A) + B) \geq r + \frac{(1+\alpha)s-r}{d}$.

To see this, suppose for the sake of contradiction that for all $i \in [d]$, we have $\dim(T_i(A)+B) < r + \frac{(1+\alpha)s-r}{d}$. Then

$$\dim\left(\sum_{i=1}^{d} T_i(A)\right) < r + d \cdot \frac{(1+\alpha)s - r}{d} = (1+\alpha)s,$$

since any vector in $\sum_{i=1}^{d} T_i(A)$ can be expressed by a linear combination of the r basis vectors in B, and another $< d \cdot \frac{(1+\alpha)s-r}{d}$ vectors, where each $T_i(A)$ contributes $< \frac{(1+\alpha)s-r}{d}$ vectors.
Now for this particular $i \in [d]$, since X_1 and X_3 are independent, we must have

$$H(T_i(X_1) + X_3) \geq r + \frac{(1+\alpha)s - r}{d} = \frac{1+\alpha}{d}k + \frac{d-2-\alpha}{d}r \geq \left(\frac{1}{2} + \frac{\alpha}{2d}\right)k,$$

as long as $d \geq 3$.

Note that $T_i(X_1) + X_2 = T_i(X_1) + X_3 + X_4$, and X_4 is a deterministic function of X_1. Since $H(X_4) = t < \frac{\alpha}{4d}k$, we can fix X_4 and conditioned on any such fixing,

$$H(T_i(X_1) + X_2) \geq \left(\frac{1}{2} + \frac{\alpha}{2d}\right)k - t > \left(\frac{1}{2} + \frac{\alpha}{2d}\right)k - \frac{\alpha}{4d}k = \left(\frac{1}{2} + \frac{\alpha}{4d}\right)k.$$

Therefore, in the end we still have $H(T_i(X_1) + X_2) > (\frac{1}{2} + \frac{\alpha}{4d})k = (1 + \frac{\alpha}{2d})\delta(n/2)$.

Case 2. $(\frac{1}{2} - \frac{\alpha}{8d})k < r \leq k/2$. The proof of this case is similar, with a slight modification. Specifically, we have $\dim(B) = r \leq \delta(n/2) \leq (1/2) \cdot (n/2)$. Consider the d linear mappings $\{T_i\}_{i=1}^{d}$ given by the dimension expander of Theorem 19. By exactly the same argument as before, we have the following claim.

Claim 22. *There exists an $i \in [d]$ such that* $\dim(A + T_i(B)) \geq s + \frac{(1+\alpha)r-s}{d}$.

Now again, since X_1 and X_3 are independent, we must have

$$H(X_1 + T_i(X_3)) \geq s + \frac{(1+\alpha)r - s}{d} = \frac{1+\alpha}{d}k + \frac{d-2-\alpha}{d}s \geq \left(\frac{1}{2} + \frac{\alpha}{2d}\right)k,$$

as long as $d \geq 3$.

Note that $X_1 + T_i(X_2) = X_1 + T_i(X_3) + T_i(X_4)$, and X_4 is a deterministic function of X_1. Since $H(X_4) = t < \frac{\alpha}{4d}k$, we can fix X_4 and conditioned on any such fixing,

$$H(X_1 + T_i(X_2)) \geq \left(\frac{1}{2} + \frac{\alpha}{2d}\right)k - t > \left(\frac{1}{2} + \frac{\alpha}{2d}\right)k - \frac{\alpha}{4d}k = \left(\frac{1}{2} + \frac{\alpha}{4d}\right)k.$$

Therefore, in the end we still have $H(X_1 + T_i(X_2)) > (\frac{1}{2} + \frac{\alpha}{4d})k = (1 + \frac{\alpha}{2d})\delta(n/2)$.

□

We can now give our main condenser, which involves repeated use of the basic condenser.

Algorithm 2 $\mathsf{SCond}(x)$

Input: $x \in \mathsf{F}_2^n$ — an n bit string; $0 < \delta \leq 1/2$, a given parameter.
Output: $z \in (\mathsf{F}_2^m)^\ell$ — a matrix of ℓ bit strings with length m, where $m = n/\mathsf{poly}(1/\delta)$ and $\ell = \mathsf{poly}(1/\delta)$.

Sub-Routines and Parameters:
Let $\mathsf{BasicCond}$ be the basic condenser given by Algorithm 1.

Set $x^0 = x$ and let $i = 0$. Initially x^i has only $n_0 = 1$ row.

1. Repeat the following step for some $h = O(\log(1/\delta))$ steps: For each j and the j'th row x^i_j in x^i, apply $\mathsf{BasicCond}(x^i_j)$ to get $2d + 2$ rows. Concatenate them to get x^{i+1} with $n_{i+1} = n_i \cdot (2d + 2)$ rows. Set $i \leftarrow i + 1$.
2. Let $z = x^h$.

We can now prove our main theorem.

Proof of Theorem 18. We show that Algorithm 2 gives such an affine somewhere condenser. By Lemma 20, for any affine source X with $H(x) = \delta n$ for some $0 < \delta \leq 1/2$, after some $h' = O(\log(1/\delta))$ steps at least one of the rows $x^{h'}$ has entropy at least $n'/2$. Without loss of generality assume this row has entropy exactly $n'/2$ (otherwise we can first fix some basis vectors in the support linear subspace and thus reduce the row to a convex combination of affine sources with entropy exactly $n'/2$). Then after another step of applying $\mathsf{BasicCond}$, one of the output rows will have entropy rate at least $1/2(1 + \frac{\alpha}{4d}) = \frac{1}{2} + \frac{\alpha}{8d}$.

It's easy to see that SCond is a linear function, and thus each row in the final output is an affine source. Furthermore, since we divide each row into 2 equal blocks in every step and obtain $2d+2$ new rows from them, the final length of each row is $m = n/\mathsf{poly}(1/\delta)$ and we have altogether $\ell = \mathsf{poly}(1/\delta)$ rows. □

4 Linear Somewhere Condenser for General Weak Sources

We next show that our linear somewhere condenser also works for general weak random sources.

4.1 Some Useful Results

Definition 23. *The collision probability of a distribution $\mathcal{D}$ is defined as* $\mathsf{cp}(\mathcal{D}) = \mathbf{Pr}_{x,y \leftarrow_R \mathcal{D}}[x = y]$.

Definition 24. *We say a distribution $\mathcal{X}$ is a convex combination of distributions $\mathcal{X}_1, \cdots, \mathcal{X}_m$ if there exist numbers $p_1, \cdots, p_m \in [0, 1]$ such that $\sum_i p_i = 1$ and the random variable $\mathcal{X}$ is equal to $\sum_i p_i \mathcal{X}_i$.*

Lemma 23 ([BISW04])**.** *Let $\mathcal{X}$ be a distribution such that $\mathsf{cp}(\mathcal{X}) \leq \frac{1}{KL}$. Then $\mathcal{X}$ is of statistical distance $\frac{1}{\sqrt{L}}$ from having min-entropy at least $\log K$.*

We need the following results from additive combinatorics.

Lemma 24 (Plűnnecke-Ruzsa [TV06]). *Let A, B be finite subsets in an additive group G. Then*

$$|A + A| \leq \frac{|A + B|^4}{|A||B|^2}.$$

Lemma 25 (Balog-Szemeredi-Gowers [BS94, Gow98]). *Let A, B be finite subsets of an additive group G and let $|A|^{1-\rho_1} \leq |B| \leq |A|^{1+\rho_1}$. If $\mathsf{cp}(A + B) \geq |A|^{-(1+\rho_2-\rho_1)}$, then there exist subsets $A' \subseteq A, B' \subseteq B$ such that $|A'| \geq |A|^{1-10\rho_2}, |B'| \geq |B|^{1-10\rho_2}$, and $|A' + B'| \leq |A|^{1+\rho_1+10\rho_2}$.*

Theorem 26 (Polynomial Freiman-Ruzsa Theorem in F_2^n [GGMT23]). *Let $A \subset \mathsf{F}_2^n$ be a set such that $|A+A| \leq M|A|$. Then there exists a subset $A' \subset A$ of size $|A'| \geq M^{-c}|A|$ such that $|\mathsf{Span}(A')| \leq M^c|A|$, where $c \geq 0$ is an absolute constant.*

4.2 The Construction

We generalize our affine somewhere condenser as follows.

Algorithm 3 $\mathsf{BasicGCond}(x)$

Input: $x \in \{0,1\}^n$ — an n bit string.
Output: $z \in (\{0,1\}^m)^{2d+3}$ — an array of $2d+3$ bit strings with length m, where $m = n/2$ and d is the constant in Theorem 19.

Sub-Routines and Parameters:
Let $T = \{T_i\}_{i=1}^d$ be the α-dimension expander given by Theorem 19.

Divide x into 2 blocks $x = x_1 \circ x_2$ where each block has $n/2$ bits.
Let $z = z_1 \circ z_2 \circ \cdots \circ z_{2d+3}$, where $z_1 = x_1$, $z_2 = x_2$, and $z_{2i+1} = x_1 + T_i(x_2)$, $z_{2i+2} = x_2 + T_i(x_1)$, for any $i \in [d]$. Finally let $z_{2d+3} = x_1 + x_2$. Here all additions are viewing the inputs as elements in the field F_2^m.

We have the following lemma.

Lemma 27. *For any $0 < \delta \leq 1/2$, $\mathsf{BasicGCond}$ is a rate $(\delta \to (1 + \Omega(\frac{\alpha}{d}))\delta, 2^{-\Omega(\delta n)})$ somewhere condenser, where α, d are the constants in Theorem 19.*

To prove the lemma we first prove the following lemmas.

Lemma 28. *For any constant $c > 0$ there exists a constant $\varepsilon = \Omega(\frac{\alpha}{d})$ such that the following holds. Let A, B be finite subsets of F_2^n. For any $K \leq 2^{n/4}$, assume $K^{1-c\varepsilon} \leq |A|, |B| \leq K^{1+c\varepsilon}$. If $\mathsf{cp}(A + B) \geq K^{-(1+2\varepsilon)}$, then there exist subsets $\tilde{A} \subseteq A, \tilde{B} \subseteq B$ such that $|\tilde{A}| \geq K^{1-O(\varepsilon)}, |\tilde{B}| \geq K^{1-O(\varepsilon)}$, $|\mathsf{Span}(\tilde{A})| \leq K^{1+O(\varepsilon)}, |\mathsf{Span}(\tilde{B})| \leq K^{1+O(\varepsilon)}$, and at least one row in the output of $\mathsf{BasicCond}(\mathcal{A} \circ \mathcal{B})$ has min-entropy $(1 + \Omega(\varepsilon)) \log K$, where $\mathcal{A}, \mathcal{B}$ are the uniform and independent distributions over $\tilde{A}, \tilde{B}$ respectively.*

Proof. If $\mathsf{cp}(A + B) \geq K^{-(1+2\varepsilon)}$, by Lemma 25 there exist subsets $A' \subseteq A, B' \subseteq B$ such that $|A'| \geq |A|^{1-O(\varepsilon)}, |B'| \geq |B|^{1-O(\varepsilon)}$, and $|A' + B'| \leq |A|^{1+O(\varepsilon)} = K^{1+O(\varepsilon)}$. Then, by Lemma 24, we have $|A' + A'| \leq \frac{|A'+B'|^4}{|A'||B'|^2} \leq K^{1+O(\varepsilon)}$. Similarly we also have $|B' + B'| \leq K^{1+O(\varepsilon)}$.

Next, by Theorem 26, there exists a subset $\tilde{A} \subset A'$ of size $|\tilde{A}| \geq K^{-O(\varepsilon)}|A'| = K^{1-O(\varepsilon)}$ such that $|\mathsf{Span}(\tilde{A})| \leq K^{O(\varepsilon)}|A'| = K^{1+O(\varepsilon)}$. Similarly there also exists such a subset $\tilde{B} \subset B'$ with the

same property. Now let $\mathcal{A}', \mathcal{B}'$ be the uniform and independent distributions over $\mathsf{Span}(\tilde{A}), \mathsf{Span}(\tilde{B})$ respectively. Note that $\mathcal{A}', \mathcal{B}'$ are both affine sources and hence $\mathcal{A}' \circ \mathcal{B}'$ is also an affine source with entropy $\geq \log|\tilde{A}| + \log|\tilde{B}| = (1 - O(\varepsilon))2\log K$. Thus, without loss of generality we can view it as an affine source with entropy exactly $(1 - O(\varepsilon))2\log K < n/2$. Now by Lemma 20, at least one row in the output of $\mathsf{BasicCond}(\mathcal{A}' \circ \mathcal{B}')$ has entropy $(1 + \frac{\alpha}{4d})(1 - O(\varepsilon))\log K$. Note that $|\tilde{A}||\tilde{B}| \geq (|\mathsf{Span}(\tilde{A})||\mathsf{Span}(\tilde{B})|)^{(1-O(\varepsilon))}$. Thus the same row in the output of $\mathsf{BasicCond}(\mathcal{A} \circ \mathcal{B})$ has min-entropy at least $(1 + \frac{\alpha}{4d})(1 - O(\varepsilon))\log K - O(\varepsilon)\log K = (1 + \Omega(\varepsilon))\log K$, as long as $\varepsilon = \gamma\frac{\alpha}{4d}$ for a sufficiently small constant $\gamma > 0$. □

Lemma 29. *For any constant $c > 0$ there exists a constant $\varepsilon = \Omega(\frac{\alpha}{d})$ such that the following holds. Let A, B be finite subsets of F_2^n. For any $K \leq 2^{n/4}$, assume $K^{1-c\varepsilon} \leq |A|, |B| \leq K^{1+c\varepsilon}$ and $|\mathsf{Span}(B)| \leq K^{1+O(\varepsilon)}$. If $\mathsf{cp}(A + B) \geq K^{-(1+2\varepsilon)}$, then there exists a subset $\tilde{A} \subseteq A$ such that $|\tilde{A}| \geq K^{1-O(\varepsilon)}$, $|\mathsf{Span}(\tilde{A})| \leq K^{1+O(\varepsilon)}$, and at least one row in the output of $\mathsf{BasicCond}(\mathcal{A} \circ \mathcal{B})$ has min-entropy $(1 + \Omega(\varepsilon))\log K$, where $\mathcal{A}, \mathcal{B}$ are the uniform and independent distributions over $\tilde{A}, B$ respectively.*

Proof. The proof is exactly the same as the previous lemma, except in the second paragraph we can replace the set $\tilde{B}$ with B directly. □

We can now prove the following lemma.

Lemma 30. *There exists a constant $\varepsilon = \Omega(\frac{\alpha}{d})$ such that the following holds. Let A, B be finite subsets of F_2^n. For any $K \leq 2^{n/4}$, assume $K^{1-\varepsilon} \leq |A|, |B| \leq K^{1+\varepsilon}$. Let X, Y be the uniform and independent distributions over A, B respectively. Then $\mathsf{BasicGCond}(X \circ Y)$ is $K^{-\Omega(\varepsilon)}$-close to a somewhere-$(1 + \Omega(\varepsilon))\log K$ source. In particular, $X \circ Y$ can be divided into disjoint subsources, such that for each subsource, either (1) the probability mass is at most $2K^{-\varepsilon}$, or (2) the probability mass is at least $K^{-O(\varepsilon)}$, and the output of $\mathsf{BasicGCond}$ on the subsource is $K^{-\varepsilon}$-close to being an elementary somewhere $(1 + \Omega(\varepsilon))\log K$ source.*

Proof. We repeatedly apply Lemma 28 and Lemma 29, and dividing $A \times B$ into disjoint subsets as follows. First note that if $\mathsf{cp}(A + B) \leq K^{-(1+2\varepsilon)}$, then by Lemma 23, $X + Y$ is $K^{-\varepsilon/2}$-close to having min-entropy $(1 + \varepsilon)\log K$. Otherwise, by Lemma 28, there exist subsets $\tilde{A} \subseteq A, \tilde{B} \subseteq B$ such that $|\tilde{A}| \geq K^{1-O(\varepsilon)}, |\tilde{B}| \geq K^{1-O(\varepsilon)}$, $|\mathsf{Span}(\tilde{A})| \leq K^{1+O(\varepsilon)}, |\mathsf{Span}(\tilde{B})| \leq K^{1+O(\varepsilon)}$, and at least one row in the output of $\mathsf{BasicCond}(\mathcal{A} \circ \mathcal{B})$ has min-entropy $(1 + \Omega(\varepsilon))\log K$, where $\mathcal{A}, \mathcal{B}$ are the uniform and independent distributions over $\tilde{A}, \tilde{B}$ respectively.

Now consider the set $A^1 = A \setminus \tilde{A}$ and $B^1 = B \setminus \tilde{B}$. If $|A^1| \leq K^{1-2\varepsilon}$ and $|B^1| \leq K^{1-2\varepsilon}$, then the total probability mass in $X \circ Y$ corresponding to elements in $(A \times B) \setminus (\tilde{A} \times \tilde{B})$ is at most $\frac{|A^1||B| + |A||B^1|}{|A||B|} \leq 2K^{-\varepsilon}$, and we are done.

Otherwise, consider the following three sets: $\tilde{A} \times B^1$, $A^1 \times \tilde{B}$, and $A^1 \times B^1$. Note that these are disjoint subsets whose union equals $(A \times B) \setminus (\tilde{A} \times \tilde{B})$. We have several cases.

Case 1. Only one of $|A^1|$ and $|B^1|$ has size larger than $K^{1-2\varepsilon}$. Without loss of generality assume $|B^1| \leq K^{1-2\varepsilon}$. Note that in this case the total probability mass in $X \circ Y$ corresponding to elements in $(\tilde{A} \times B^1) \cup (A^1 \times B^1) = A \times B^1$ is at most $\frac{|A||B^1|}{|A||B|} \leq K^{-\varepsilon}$.

For $A \times \tilde{B}$, we repeatedly apply Lemma 29. Initially let $A^* = A$. As long as $|A^*| \geq K^{1-2\varepsilon}$, if $\mathsf{cp}(A^* + \tilde{B}) \leq K^{-(1+2\varepsilon)}$, then again by Lemma 23, the output of the sum of the random variables corresponding to the uniform and independent distributions over A^* and $\tilde{B}$ will be $K^{-\varepsilon/2}$-close to having min-entropy $(1 + \varepsilon)\log K$, and we stop here. Otherwise we use

Lemma 29 to find a subset $\tilde{A} \subseteq A^*$ such that $|\tilde{A}| \geq K^{1-O(\varepsilon)}$, $|\mathsf{Span}(\tilde{A})| \leq K^{1+O(\varepsilon)}$, and at least one row in the output of $\mathsf{BasicCond}(\mathcal{A} \circ \mathcal{B})$ has min-entropy $(1+\Omega(\varepsilon))\log K$, where $\mathcal{A}, \mathcal{B}$ are the uniform and independent distributions over $\tilde{A}, \tilde{B}$ respectively. We then remove $\tilde{A}$ from A^* and repeat. The process ends when $|A^*| < K^{1-2\varepsilon}$.

Thus, altogether, we have divided $A \times B$ into disjoint subsets, or equivalently, $X \circ Y$ into disjoint subsources, such that for each subsource, either the probability mass is at most $K^{-\varepsilon}$, or the output of $\mathsf{BasicGCond}$ on the subsource is $K^{-\varepsilon}$-close to being a somewhere $(1+\Omega(\varepsilon))\log K$ source.

Case 2. $|A^1| > K^{1-2\varepsilon}$ and $|B^1| > K^{1-2\varepsilon}$. We first apply the argument in Case 1 to $\tilde{A} \times B^1$ and $A^1 \times \tilde{B}$. Then we consider $A^1 \times B^1$. This is the same situation as when we start. Namely, if $\mathsf{cp}(A^1 + B^1) \leq K^{-(1+2\varepsilon)}$, then by Lemma 23 we are done. Otherwise by Lemma 28, there exist subsets $\tilde{A^1} \subseteq A^1, \tilde{B^1} \subseteq B^1$ such that $|\tilde{A^1}| \geq K^{1-O(\varepsilon)}, |\tilde{B^1}| \geq K^{1-O(\varepsilon)}$, $|\mathsf{Span}(\tilde{A^1})| \leq K^{1+O(\varepsilon)}, |\mathsf{Span}(\tilde{B})| \leq K^{1+O(\varepsilon)}$, and at least one row in the output of $\mathsf{BasicCond}(\mathcal{A}^1 \circ \mathcal{B}^1)$ has min-entropy $(1+\Omega(\varepsilon))\log K$, where $\mathcal{A}^1, \mathcal{B}^1$ are the uniform and independent distributions over $\tilde{A^1}, \tilde{B^1}$ respectively. We can therefore continue the analysis as before.

Combining the two cases, eventually we have divided $X \circ Y$ into disjoint subsources, such that for each subsource, either (1) the probability mass is at most $2K^{-\varepsilon}$, or (2) the output of $\mathsf{BasicGCond}$ on the subsource is $K^{-\varepsilon}$-close to being a somewhere $(1+\Omega(\varepsilon))\log K$ source.

Notice that when a subsource satisfies (2), its probability mass is always at least $K^{1-O(\varepsilon)} \cdot K^{1-O(\varepsilon)}/K^{2(1+\varepsilon)} = K^{-O(\varepsilon)}$. □

We can now prove Lemma 27.

Proof of Lemma 27. Given an $(n, \delta n)$ source X with $0 < \delta \leq 1/2$, and $X = X_1 \circ X_2$, without loss of generality we can assume that X is the uniform distribution over a set $S \subseteq \{0,1\}^n$ with $|S| = 2^{\delta n}$. We first pick a constant parameter $\lambda > 0$ to be chosen later. For $i \in [2]$ define $H_i = \{y \in \{0,1\}^m : \mathbf{Pr}[X_i = y] \geq 2^{-(1+\lambda)\delta m}\}$, which corresponds to the heavy elements in X_i. Notice that this implies for every i, $|H_i| \leq 2^{(1+\lambda)\delta m}$. Let $\tau = 2^{-\beta\delta m}$ for some constant $\beta > 0$ to be chosen later. We define the following sets.

1. $S' = \{x \in S : \exists i, x_i \notin H_i\}$.
2. For any $x \in S'$, define $I(x)$ to be the smallest i such that $x_i \notin H_i$, and $T_i = \{x \in S', I(x) = i\}$. Let $B = \{i \in [2] : |T_i| < 2^{(1-\beta)\delta m}\}$, and define $\tilde{S} = S' \setminus (\cup_{i \in B} T_i)$. Note that $|\cup_{i \in B} T_i| \leq 2\tau|S|$.
3. $S'' = \{x \in S : \forall i, x_i \in H_i\} = S \setminus S'$.

Note that for any $x \in \tilde{S}$, we have $I(x) \notin B$. Let $\tilde{X}$ be the uniform distribution over $\tilde{S}$. For any $i \in [2] \setminus B$, and any $y \in \{0,1\}^m$, conditioned on $I(\tilde{X}) = i$, we have $\mathbf{Pr}[\tilde{X}_i = y] \leq \frac{\mathbf{Pr}[X_i = y]}{2^{-\beta\delta m}} \leq 2^{-(1+\lambda-\beta)\delta m}$. Thus as long as $\beta \leq \lambda/2$, $\tilde{X}_i$ has min-entropy at least $(1+\lambda/2)\delta m$. Hence $\tilde{X}$ is an elementary somewhere-$(1+\lambda/2)\delta m$ source.

We now have two cases.

Case 1. $\mathbf{Pr}[X \in S'] \geq 1 - \tau$. In this case, notice that $\tilde{X}$ is $2\tau + \tau = 3\tau$-close to X, thus we are done.

Case 2. $\mathbf{Pr}[X \in S''] \geq \tau$. In this case, notice that $|S''| \geq \tau|S| = 2^{(2-\beta)\delta m}$. Also, S'' is a subset of $H_1 \times H_2$, so

$$|H_1 \times H_2| \geq |S''| \geq 2^{(2-\beta)\delta m}.$$

However, for each $i \in [2]$ we have $|H_i| \leq 2^{(1+\lambda)\delta m}$, and thus for each $i \in [2]$ we also have

$$|H_i| \geq 2^{(2-\beta)\delta m}/2^{(1+\lambda)\delta m} = 2^{(1-\beta-\lambda)\delta m}.$$

We now consider the source (Y_1, Y_2) where each Y_i is the independent uniform distribution over H_i. We will apply Lemma 30 by setting $K = 2^{\delta m} \leq 2^{n/4}$, and $\varepsilon \geq 2\lambda$. Notice that for any $i \in [2]$, we have $K^{1-\varepsilon} \leq |H_i| \leq K^{1+\varepsilon}$ since we have chosen $\beta \leq \lambda/2$.

Thus by Lemma 30, there exits a constant $c > 0$ such that $Y_1 \circ Y_2$ can be divided into disjoint subsources, such that for each subsource, either (1) the probability mass is at most $2K^{-\varepsilon}$, or (2) the probability mass is at least $K^{-c\varepsilon}$, and the output of $\mathsf{BasicGCond}$ on the subsource is $K^{-\varepsilon}$-close to being an elementary somewhere $(1 + \Omega(\varepsilon)) \log K$ source.

For each subsource Y^j in (2), we consider the intersection of its support with S''. If the intersection has probability mass at most $K^{-c\varepsilon-4\lambda}$, then we say it is a *bad* intersection, otherwise we say it is a *good* intersection. Notice that for a good intersection, the output of $\mathsf{BasicGCond}$ on the subsource defined as the uniform distribution over the intersection is $K^{-\varepsilon+4\lambda}$-close to being an elementary somewhere $(1 + \Omega(\varepsilon) - 4\lambda) \log K$ source. On the other hand, the total probability mass of the bad intersections is at most $K^{-4\lambda}$.

Notice that the probability mass of S'' in (Y_1, Y_2) is at least $2^{(2-\beta)\delta m}/(2^{(2+2\lambda)\delta m}) = 2^{-(\beta+2\lambda)\delta m}$. Hence if we define X'' as the uniform distribution over S'', then $\mathsf{BasicGCond}(X'')$ is $(2K^{-\varepsilon} + K^{-4\lambda})/(2^{-(\beta+2\lambda)\delta m}) + K^{-\varepsilon+4\lambda} \leq 2^{-\lambda\delta m}$-close to a somewhere $(1 + \Omega(\varepsilon) - 4\lambda) \log K = (1 + \lambda)\delta m$ source, as long as we take $\lambda = \gamma\varepsilon$ for a sufficiently small constant $\gamma > 0$.

Now define X' to be the uniform distribution over $S'' \cup \tilde{S}$. Then $\mathsf{BasicGCond}(X')$ is $2^{-\lambda\delta m}$-close to a somewhere $(1 + \lambda/2)\delta m$ source. Notice that X is 2τ-close to X'. Thus $\mathsf{BasicGCond}(X)$ is $2\tau + 2^{-\lambda\delta m}$-close to a somewhere $(1 + \lambda/2)\delta m$ source.

Setting $\beta = \lambda/2$, we have that in both cases, $\mathsf{BasicGCond}(X)$ is $2^{-\Omega(\delta n)}$-close to a somewhere $(1 + \Omega(\frac{\alpha}{d}))\delta m$ source. □

Our main condenser now involves repeated uses of the basic condenser.

Algorithm 4 $\mathsf{SGCond}(x)$

Input: $x \in \mathsf{F}_2^n$ — an n bit string; $0 < \delta \leq 1/2$, a given parameter.
Output: $z \in (\mathsf{F}_2^m)^\ell$ — a matrix of ℓ bit strings with length m, where $m = n/\mathsf{poly}(1/\delta)$ and $\ell = \mathsf{poly}(1/\delta)$.

Sub-Routines and Parameters:
Let $\mathsf{BasicGCond}$ be the basic condenser given by Algorithm 3.

Set $x^0 = x$ and let $i = 0$. Initially x^i has only $n_0 = 1$ row.

1. Repeat the following step for some $h = O(\log(1/\delta))$ steps: For each j and the j'th row x^i_j in x^i, apply $\mathsf{BasicGCond}(x^i_j)$ to get $2d + 3$ rows. Concatenate them to get x^{i+1} with $n_{i+1} = n_i \cdot (2d + 3)$ rows. Set $i \leftarrow i + 1$.
2. Let $z = x^h$.

By a similar argument as in the proof of Theorem 18, we can prove the following theorem.

Theorem 31. *There exists a constant $\beta > 0$ such that for any $0 < \delta \leq 1/2$, there is an explicit rate $(\delta \to 1/2 + \beta, 2^{-\Omega(m)})$ somewhere condenser $\mathsf{SGCond} : \{0,1\}^n \to (\{0,1\}^m)^t$, where $t = \mathsf{poly}(1/\delta)$ and $m = n/\mathsf{poly}(1/\delta)$. Moreover, SGCond is a linear function.*

5 Directional Affine Extractor

In this section, we describe our directional affine extractors for linear entropy with exponentially small error. The construction also works for sublinear entropy with a slight loss in the error and output length.

5.1 Low-Degree Affine Correlation Breaker

As we introduced in Section 1, keeping the outputs of the directional affine extractor *low-degree* is critical. However, our construction makes use of advice correlation breakers, and all existing correlation breakers have degrees forbiddenly high for our purpose. To handle this, we construct a family of low-degree correlation breakers. We assume that the input random variables to each of the following subroutines are affine. This assumption is valid since in the analysis of Algorithm 8 where we invoke Theorem 37 of IdACB, the input random variables are affine.

Substitutes for strong seeded extractors. We will base our construction on a similar framework to the advice correlation breaker in [CGL22]. To keep the degree low, we substitute the GUV extractors and the condense-then-hash extractors used throughout with the low-degree strong linear seeded extractor from Theorem 32.

Theorem 32 (Low-degree strong linear seeded extractors [Li11])**.** *There exists a constant $0 < \beta < 1$ such that for every $0 < \delta < 1$ and any $1/\sqrt{n} < \alpha < 1$ there exists a polynomial time computable function $\mathsf{LSExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ and a constant $0 < \beta < 1$ such that s.t.*

- *$d \leq \alpha n, m \geq \beta\delta\alpha n$.*
- *For any $(n, \delta n)$-affine source X, let R be the uniform distribution on $\{0,1\}^d$ independent of X. Then $(\mathsf{LSExt}(X,R), R)$ is $2^{-\Omega(\delta\alpha^2 n)}$-close to uniform.*
- *Each bit of the output is a degree 4 polynomial of the bits of the two inputs, and for any fixing of r the output is a linear function of x.*

Low-degree look-ahead extractor. The first step is to construct a low-degree look-ahead extractor which is a component of the low-degree advice correlation breaker. The following algorithm is such a construction instantiated with the low-degree strong linear seeded extractor in Theorem 32. Since the low-degree strong linear seeded extractor has shorter output length than the minimal seed length, we cannot directly apply existing lemmas about look-ahead extractors. Instead, we need to tailor a new set of parameters and a new theorem for the low-degree one.

Algorithm 5 (k,t,ε)-$\mathsf{laExt}(x,y)$

Input: Bit strings x, y of length n, d respectively. Initially, x has entropy k.
Output: Bit string (r_0, r_1) of length $2m$.
Subroutines and Parameters:
Let $s = d/(2+2t)$, where $C_0\sqrt{\frac{\log(1/\varepsilon)}{k}}n \geq s \geq C_1 n\sqrt{n}/k$ for some constants $C_0 > 0, C_1 > 1$.
Let $\mathsf{LSExt}^1_w : \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^{m_1}$ be the low-degree strong linear seeded extractor from Theorem 32 with $\delta_{32} = k/n$, $\alpha_{32} = d/((2t+2)n)$, error $\varepsilon_0 = 2^{-\Omega(kd^2/((t+1)^2n^2)}$ and output length $m_1 = \beta_{32}kd/((2t+2)n)$.
Let $\mathsf{LSExt}^1_q : \{0,1\}^d \times \{0,1\}^{m_1} \to \{0,1\}^{m_2}$ be the low-degree strong linear seeded extractor from Theorem 32 with $\delta_{32} = 1/2$, $\alpha_{32} = \beta_{32}k/((2+2t)n)$, error $\varepsilon_1 = 2^{-\Omega(dk^2/((t+1)^2n^2))} = \varepsilon_0^{\Omega(d/k)}$ and output length $m_2 = \beta_{32}^2 kd/(4(t+1)n)$.
Let $\mathsf{LSExt}^2_w : \{0,1\}^n \times \{0,1\}^{m_2} \to \{0,1\}^m$ be the low-degree strong linear seeded extractor from Theorem 32 with $\delta_{32} = k/(2n)$ and $\alpha_3 = \beta_{32}^2 kd/((4+4t)n^2)$, error $\varepsilon_2 = 2^{-\Omega(k^3d^2/((8+8t)^2n^4))} = \varepsilon_0^{\Omega(k^2/n^2)}$ and output length $m = \beta_{32}^3 k^2 d/((8+8t)n^2)$.

1. Let $s_0 = \mathsf{Slice}(y, s)$
2. Let $\tilde{r}_0 = \mathsf{LSExt}^1_w(x, s_0)$
3. Let $s_1 = \mathsf{LSExt}^1_q(y, \tilde{r}_0)$
4. Let $r_1 = \mathsf{LSExt}^2_w(x, s_1)$
5. Output $r_0 = \mathsf{Slice}(\tilde{r}_0, |r_1|), r_1$

Theorem 33 (2-look-ahead extractor)**.** *For every $t \leq \sqrt{n}$, $t \in \mathbb{N}$ and $\varepsilon > 0$, there exists an explicit function $\mathsf{laExt} : \{0,1\}^n \times \{0,1\}^d \to (\{0,1\}^m)^2$ which satisfies the following. Let $X, X^{[t]} \in \{0,1\}^n$ and $Y, Y^{[t]} \in \{0,1\}^d$ be random variables such that $\left(X, X^{[t]}\right)$ is independent of $\left(Y, Y^{[t]}\right)$, $Y = U_d$. There exists a large enough constant $C > 0$ such that if*

$$k = H(X) \geq C \max\left\{\left((t+1)^2 \log(1/\varepsilon) n^4/d^2\right)^{1/3}, (t+1)\sqrt{n}\right\};$$
$$n \geq d \geq C(t+1)\max\left\{n\sqrt{n}/k, \left(\log(1/\varepsilon)n^4/k^3\right)^{1/2}\right\},$$

then $(R_0, R_1) := \mathsf{laExt}(X, Y)$ and their tamperings $(R_0^{[t]}, R_1^{[t]})$ satisfy

$$(R_0 \approx_\varepsilon U_m) \mid (Y, Y^{[t]});$$
$$(R_1 \approx_\varepsilon U_m) \mid (Y, Y^{[t]}, R_0, R_0^{[t]}),$$

where $m = \Omega(k^2 d/((1+t)n^2))$.
Moreover, each bit of r_0' is a degree 4 polynomial of the input bits; each bit of r_1 is a degree 40 polynomial of the input bits.

Proof. We will show that Algorithm 5 is such a function. First we demonstrate that the choice of the parameters in Algorithm 5 are correct. The first constraint comes from the requirement of the minimal seed length of any LSExt, i.e., we need to guarantee that

$$d/((2t+2)n) > 1/\sqrt{n}\ ; \qquad \text{(seed length requirement of } \mathsf{LSExt}^1_w)$$
$$\beta_{32}k/((2+2t)n) > 1/\sqrt{n}\ ; \qquad \text{(seed length requirement of } \mathsf{LSExt}^1_q)$$
$$\beta_{32}^2 kd/((4+4t)n^2) > 1/\sqrt{n}\ , \qquad \text{(seed length requirement of } \mathsf{LSExt}^2_w)$$

this puts a lower bound for d:

$$d > \frac{4}{\beta_{32}^2} \cdot \frac{(t+1)n\sqrt{n}}{k}.$$

Since $n \geq d$, we have

$$k > \frac{4}{\beta_{32}^2} \cdot (t+1)n.$$

Let $\varepsilon \geq \varepsilon_0 + \varepsilon_1 + \varepsilon_2$, then there exists a constant λ such that $\varepsilon \geq 2^{-\lambda k^3 d^2/((t+1)^2 n^4)}$. Taking the logarithm on the error and isolating out k and d, we have

$$\frac{(t+1)^2 \log(1/\varepsilon) n^4}{\lambda} \leq k^3 d^2 \iff k \geq C_0 \left((t+1)^2 \log(1/\varepsilon) n^4/d^2\right)^{1/3};$$
$$\frac{(t+1)^2 \log(1/\varepsilon) n^4}{\lambda} \leq k^3 d^2 \iff d \geq C_0'(t+1) \left(\log(1/\varepsilon) n^4/k^3\right)^{1/2},$$

for some large enough constants C_0, C_0'. Therefore, if k and d satisfy the constraints in Theorem 33, they also works for Algorithm 5. Next, we prove the extraction properties of the look-ahead extractor.

1. Since $Y = U_d$, S_0 is uniform. Since Y is independent of X, X is independent of S_0. Since $H(X) \geq k$, by the property of strong seeded extractor of LSExt_w^1, $\tilde{R}_0 \approx_{\varepsilon_0} U_{m_1} \mid S_0$, which also implies that $R_0 \approx_{\varepsilon_0} U_m \mid (Y, Y^{[t]})$ given the independence between X and $(Y, Y^{[t]})$.

2. Since $S_0, S_0^{[t]}$ are linear functions of Y and $Y^{[t]}$, we have $H(Y \mid S_0, S_0^{[t]}) \geq d - (t+1) \cdot \frac{d}{2+2t} = d/2$. Since $\tilde{R}_0$ is ε_0 close to uniform, by the property of strong seeded extractor of LSExt_q^1, $S_1 \approx_{\varepsilon_0+\varepsilon_1} U_{m_2}$.

3. Conditioned on the fixings of $(S_0, S_0^{[t]})$, $R_0, R_0^{[t]}$ are linear functions of X and $X^{[t]}$. Therefore, we have $H(X \mid \tilde{R}_0, \tilde{R}_0^{[t]}) \geq k - (t+1)m_1 = k/2$. Since $S_1 \approx_{\varepsilon_0+\varepsilon_1} U_{m_2}$, by the property of strong seeded extractor of LSExt_w^2, $R_1 \approx_{\varepsilon_0+\varepsilon_1+\varepsilon_2} U_m \mid (R_0, R_0^{[t]}, Y, Y^{[t]})$.

Lastly, the degree of the output follows easily from the degree of the output of LSExt from Theorem 32. This completes the proof of Theorem 33. □

Low-degree non-malleable independence-preserving merger. The second step is to construct a low-degree non-malleable independence-preserving merger. Non-malleable independence-perserving merger was first defined in [CL16a] to merge a somewhere random source while preserving independence among itself and the tampered sources. We start with the definition.

Definition 25. *An (t, ℓ, ε)-$\mathsf{NIPM} : \{0,1\}^n \times (\{0,1\}^m)^\ell \to \{0,1\}^{m_1}$, or NIPM_ℓ for short, with error ε for $\ell \in \mathbb{N}$ is function which satisfies the following property. Suppose*

- *$V, V^{[t]}$ are random variables, each supported on boolean $\ell \times m$ matrices, s.t. for any $i \in [\ell]$, $V_i = U_m$;*
- *for every $j \in [t]$, there exists an $h_j \in [\ell]$ such that $(V_{h_j}, V_{h_j}^j) = (U_m, V_{h_j}^j)$;*
- *$X, X^{[t]}$ are random variables independent of $V, V^{[t]}$, each supported on d bits and X has enough entropy,*

then

$$\mathsf{NIPM}_\ell(X, V) \approx_\varepsilon U_{m_1} \mid (\mathsf{NIPM}_\ell(X^1, V^1), \cdots, \mathsf{NIPM}_\ell(X^t, V^t)).$$

Algorithm 6 $\mathsf{NIPM}_\ell(x, v)$

Input: x — an n bit string, v — an $\ell \times m$ bit matrix.
Output: z — an $m \cdot \prod_{i=1}^{\ell} \alpha_i$ bit string where each α_i is defined below for $i \in [\ell]$.

Sub-Routines and Parameters:
Let $\delta_w = k/2n$ be a lower bound on the assumed entropy rate of x in Definition 25, $\delta_q = 1/2$ a lower bound on the entropy rate of for each v_{h_i} where $i \in [\ell]$. Let $\alpha_1 = m/((3+3t)n)$.
For $i \in [\ell - 1]$:

- set δ_{32} and α_{32} from Theorem 32 to be δ_w and α_i respectively;
- set δ_{32} and α_{32} from Theorem 32 to be δ_q and $\delta_w \beta_{32} \alpha_i$ respectively;
- let $\alpha_{i+1} = \delta_q \delta_w \beta_{32}^2 \alpha_i$,

which gives

- $\alpha_i = (\delta_w \delta_q)^{i-1} \beta_{32}^{2i-2} \alpha_1$.

For $i \in [\ell - 1]$:

- $\mathsf{LSExt}_w^i : \{0,1\}^n \times \{0,1\}^{\alpha_i n} \to \{0,1\}^{\delta_w \beta_{32} \alpha_i n}$ be the extractor from Theorem 32 with error $\varepsilon_i^w = 2^{-\Omega(\delta_w^{2i-1} \delta_q^{2i-2} \alpha_1^2 n)}$.
- $\mathsf{LSExt}_q^i : \{0,1\}^m \times \{0,1\}^{\delta_w \beta_{32} \alpha_i n} \to \{0,1\}^{\alpha_{i+1} n}$ be the extractor from Theorem 32 with error $\varepsilon_i^q = 2^{-\Omega(\delta_q^{2i-1} \delta_w^{2i-1} \alpha_1^2 m)}$.

Let $s_1 = \mathsf{Slice}(v_1, \alpha_1 n)$.
For $i \in [\ell - 1]$:

1. $r_i = \mathsf{LSExt}_w^i(x, s_i)$
2. $s_{i+1} = \mathsf{LSExt}_q^i(v_{i+1}, r_i)$

Let $z = s_\ell$.

Theorem 34 (NIPM_ℓ)**.** *For every $\ell \in \mathbb{N}, \varepsilon > 0$, if there exists a large enough C such that*

- $k \geq C \max\left\{ \left((t+1)^2 \log(1/\varepsilon) n^{2\ell-1}/m^3\right)^{1/(2\ell-3)}, (t+1)\sqrt{n} \right\}$;
- $n \geq m \geq C \max\left\{ (t+1) n\sqrt{n}/k, ((t+1)\log(1/\varepsilon) n^{2\ell-1}/k^{2\ell-3})^{1/3} \right\}$,

then there exists an $\mathsf{NIPM}_\ell : \{0,1\}^n \times (\{0,1\}^m)^\ell \to \{0,1\}^{m_1}$ and constants $0 < \eta < 1$ and $\{\varepsilon_i^w, \varepsilon_i^q\}_{i=1}^{\ell-1}$ each larger than 0 such that

- $\varepsilon_i^w = \varepsilon^{\Omega((n^2/k^2)^{\ell-i})}$.
- $\varepsilon_i^q = \varepsilon^{\Omega((n^2/k^2)^{\ell-i})}$.
- $m_1 \geq \eta^\ell m/(t+1)$.

- *each output bit of* NIPM_ℓ *is a degree* $2^{\Theta(\ell)}$ *polynomial of the input bits.*

In the analysis of NIPM_ℓ (and later IdACB), we will be using the lemma below repeatedly. It is adjusted from Lemma 15 for affine sources. See Appendix B for a proof.

Lemma 35 (Independence-merging lemma for affine sources)**.** *Let* $\mathsf{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be any* (k,ε)*-strong linear seeded extractor,* $X_0 \in \{0,1\}^n$ *an affine source,* $X, X^{[t]} \in \{0,1\}^n$*,* $Y, Y^{[t]} \in \{0,1\}^d$ *all linear functions of* X_0*,* $W = \mathsf{LExt}(X,Y)$ *and* $W^j = \mathsf{LExt}(X^j, Y^j)$ *for every* $j \in [t]$*. Suppose there exists* $S, T \subseteq [t]$ *such that*

- $(Y, Y^S) \approx_\delta (U_d, Y^S)$*;*
- $H(X \mid X^T, Y, Y^{[t]}) \geq k + tm$*,*

then

$$W \approx_{\varepsilon+\delta} U_m \mid (W^{S\cup T}, Y, Y^{[t]}).$$

Proof of Theorem 34. We will show that Algorithm 6 is such an NIPM_ℓ. We first argue about the degree of each output bit. Let the degree of s_i be d_i for all $i \in [\ell]$, then they satisfy the following recursive formula

$$d_i = \begin{cases} 1 & \text{if } i = 1 \\ 3(3d_{i-1}+1)+1 = 9d_{i-1}+4 & \text{if } i > 1 \end{cases}$$

solving which gives us $d_\ell = \frac{9^{\ell-1}-1}{2}$.
We now use induction to show the following claim. We let $R^{[j]}_{h_{[j]}} := \{R^1_{h_1}, R^2_{h_2}, \cdots, R^j_{h_j}\}$.

Claim 36. *Without loss of generality, let* $1 \leq h_1 \leq \cdots \leq h_t \leq \ell$*. For every* $j \in [t]$*, the following holds after step* h_j

$$S_{h_j} \approx_{\sum_{i\in[h_j-1]}\varepsilon_i^w + \sum_{i\in[h_j-1]}\varepsilon_i^q} U_{\alpha_{h_j} n} \mid (S_{[i-1]}, S^{[t]}_{[i-1]}, R_{[i-1]}, R^{[t]}_{[i-1]}),$$
$$R_{h_j} \approx_{\sum_{i\in[h_j]}\varepsilon_i^w + \sum_{i\in[h_j-1]}\varepsilon_i^q} U_{\delta_w \beta^w_{h_j} \alpha_{h_j} n} \mid (R^{[j]}_{h_{[j]}}, S_{[h_j]}, S^{[t]}_{[h_j]}, R_{[h_j-1]}, R^{[t]}_{[h_j-1]}),$$

which implies that

$$S_\ell \approx_{\sum_{i\in[\ell-1]}(\varepsilon_i^q+\varepsilon_i^w)} U_{\alpha_\ell n} \mid (V^{[t]}_{h_{[t]}}, S^{[t]}_\ell, S_{[\ell-1]}, S^{[t]}_{[\ell-1]}, R_{[\ell-1]}, R^{[t]}_{[\ell-1]}).$$

Proof. We skip writing errors explicitly below whenever they can be easily seen to follow the claim.
Case $i \leq h_1 - 1$**.** We prove by induction that

$$\begin{aligned} S_i &\approx_{\sum_{j\in[i-1]}\varepsilon_j^w + \sum_{j\in[i-1]}\varepsilon_j^q} U_{\alpha_i n} \mid (S_{[i-1]}, S^{[t]}_{[i-1]}, R_{[i-1]}, R^{[t]}_{[i-1]}), \\ R_i &\approx_{\sum_{j\in[i]}\varepsilon_j^w + \sum_{j\in[i-1]}\varepsilon_j^q} U_{\delta_w \beta_i^w \alpha_i n} \mid (R_{[i-1]}, R^{[t]}_{[i-1]}, S_{[i]}, S^{[t]}_{[i]}). \end{aligned} \tag{1}$$

In round 1, since $V_1 = U_m$, $S_1 = U_{\alpha_1 n}$. Then by Lemma 35, $(R_1, S_1, S^{[t]}_1) \approx_\varepsilon (U_{\delta_w \beta_1^w \alpha_1 n}, S_1, S^{[t]}_1)$. Then, assume that Eqn. (1) holds $\forall i \in [h_1 - 2]$. Since

$H(V_{i+1} \mid R_{[i]}, R^{[t]}_{[i]}, S_{[i]}, S^{[t]}_{[i+1]}) = H(V_{i+1} \mid S_{[i]}, S^{[t]}_{[i+1]}) \geq m - (t+1)(\sum_{j=1}^{i+1}\alpha_j n) \geq m/2$, by the property of strong seeded extractor, the first part of Eqn. (1) holds. Since

$H(X \mid S_{[i]}, S^{[t]}_{[i]}, R_{[i]}, R^{[t]}_{[i+1]}) = H(X \mid R_{[i]}, R^{[t]}_{[i+1]}) \geq k - (t+1)\delta_w(\sum_{j=1}^{i+1} \beta_{32}\alpha_i n) \geq k/2$, by the property of strong seeded extractor, the second part of Eqn. (1) holds.

Case $i = h_j$. We prove by induction that

$$\begin{aligned} S_{h_j} &\approx U_{\delta_w \beta^w_{h_j} \alpha_{h_j} n} \mid (V^{[j]}_{h_{[j]}}, S^{[j]}_{h_{[j]}}, S_{[h_j-1]}, S^{[t]}_{[h_j-1]}, R_{[h_j-1]}, R^{[t]}_{[h_j-1]}), \\ R_{h_j} &\approx U_{\alpha_i n} \mid (R^{[j]}_{h_{[j]}}, S_{[h_j]}, S^{[t]}_{[h_j]}, R_{[h_j-1]}, R^{[t]}_{[h_j-1]}), \end{aligned} \tag{2}$$

then by Lemma 35, for $i \in [h_j + 1, h_{j+1} - 1]$, $j \in [t]$ (we defined $h_{t+1} - 1 := \ell$), it holds that

$$\begin{aligned} S_i &\approx U_{\delta_w \beta^w_i \alpha_i n} \mid (V^{[j]}_{h_{[j]}}, S^{[j]}_i, S_{[i-1]}, S^{[t]}_{[i-1]}, R_{[i-1]}, R^{[t]}_{[i-1]}), \\ R_i &\approx U_{\alpha_i n} \mid (R^{[j]}_i, S_{[i]}, S^{[t]}_{[i]}, R_{[i-1]}, R^{[t]}_{[i-1]}). \end{aligned} \tag{3}$$

In round $h_1 - 1$, $(V_{h_1}, V^1_{h_1}) \approx (U_m, V^1_{h_1})$. By Lemma 35,

$S_{h_1} \approx U_{\delta_w \beta^w_{h_1} \alpha_{h_1} n} \mid (V^1_{h_1}, S^1_{h_1}, S_{[h_1-1]}, S^{[t]}_{[h_j-1]}, R_{[h_1-1]}, R^{[t]}_{[h_1-1]})$. Then, again by Lemma 35, $R_{h_1} \approx U_{\alpha_1 n} \mid (R^1_{h_1}, S_{[h_1]}, S^{[t]}_{[h_1]}, R_{[h_1-1]}, R^{[t]}_{[h_1-1]})$. Assume that Eqn. (2) holds for $i \in h_{[j-1]}$, $j \in [t]$. Since $(V_{h_j}, V^j_{h_j}) \approx (U_m, V^j_{h_j})$ and that the second equation in Eqn. (3) for $i \in [h_j - 1]$ holds, then by Lemma 35, Eqn. (2) holds for $i = h_j$. □

Lastly, the setting of the parameters is similar to that of Theorem 33. This completes the proof of Theorem 34. □

Low-degree advice correlation breaker Now, we are ready to give the construction of the low-degree correlation breaker. We first give the definition of low-degree advice correlation breaker.

Definition 26 (IdACB)**.** *A function* $\mathsf{IdACB} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^a \to \{0,1\}^m$ *is an advice correlation breaker for linearly correlated sources if the following holds. Let*

- $A, A^{[t]}, B, B^{[t]}$ *be random variables on* $\{0,1\}^n$ *and* $Y, Y^{[t]}$ *be random variables on* $\{0,1\}^d$ *such that* $(A, A^{[t]})$ *is independent of* $(B, B^{[t]}, Y, Y^{[t]})$*. Moreover,* $H(A) \geq k$ *and* $Y = U_d$*;*
- $X = A + B, X^i = A^i + B^i$ *for every* $i \in [t]$*;*
- $\alpha, \alpha^1, \cdots, \alpha^t$ *be* a*-bit strings s.t.* $\alpha \neq \alpha^i$ *for every* $i \in [t]$*;*
- *each bit of the output is a constant degree polynomial of the inputs* X (X^i) *and* Y (Y^i)*,*

then

$$(\mathsf{IdACB}(X, Y, \alpha) \approx_\varepsilon U_m) \mid \left(\mathsf{IdACB}(X^1, Y^1, \alpha^1), \cdots, \mathsf{IdACB}(X^t, Y^t, \alpha^t)\right).$$

Moreover, if there are random variables X', A', B' *and* Y' *such that* $X' = A' + B'$ *and* $(Y = U_d) \mid Y'$*, then it also holds that*

$$(\mathsf{IdACB}(X, Y, \alpha) \approx_\varepsilon U_m) \mid \left(\mathsf{IdACB}(X', Y', \alpha), \mathsf{IdACB}(X^1, Y^1, \alpha^1), \cdots, \mathsf{IdACB}(X^t, Y^t, \alpha^t)\right).$$

We remark that Definition 26 differs from standard definitions in that it allows conditioning on an tampered output with the same advice, given that the seed is non-malleable to the tampered seed. We will be using this property in our proof for directional affine extractors.

In our construction, we also need the following function.

Definition 27 (FFAssign [CGL22]). *Let* $\mathsf{FFAssign} : (\{0,1\}^n)^2 \times \{0,1\}^a \to (\{0,1\}^n)^{2a}$ *be defined as follows. Let* $r_0, r_1 \in \{0,1\}^n$ *and* $\alpha \in \{0,1\}^a$*. Let* α_j *denote the* j*-th bit of* α*. Then* $\mathsf{FFAssign}(r_0, r_1, \alpha) := (r_{\alpha_1}, r_{1-\alpha_1}, \cdots, r_{\alpha_a}, r_{1-\alpha_a})$.

Algorithm 7 $\mathsf{IdACB}(x, y, id)$

Input: Bit strings $x = w + z, y, id$ of length n, d, a respectively, where $d < n$.
Output: Bit string y' of length n_2.
Subroutines and Parameters:
Let $\mathsf{LSExt} : \{0,1\}^n \times \{0,1\}^{m_1} \to \{0,1\}^{m_2}$ from Theorem 32 with $m_1 = d/(4+2t)$, output length $m_2 = \beta_{32} kd/((8+4t)n)$, entropy $k/2$ and error ε_1.
Let $\mathsf{IaExt} : \{0,1\}^d \times \{0,1\}^{m_2} \to (\{0,1\}^v)^2$ from Theorem 33 where $v = \Omega(m_1/(16+16t)) = \Omega(d/(32(1+t)^2))$ with entropy $d/3$ and error ε_2.
Let $\mathsf{FFAssign} : (\{0,1\}^v)^2 \times \{0,1\}^a \to (\{0,1\}^v)^{2a}$ from Definition 27.
Let $\mathsf{NIPM}_{2a} : \{0,1\}^n \times (\{0,1\}^v)^{2a} \to \{0,1\}^{n_2}$ from Theorem 34 with entropy $k/2$ and error ε_3.

1. Let $s = \mathsf{Slice}(y, m_1)$.
2. Let $q = \mathsf{LSExt}(x, s)$.
3. Let $(r_0, r_1) = \mathsf{IaExt}(y, q)$.
4. Let $(v_1, v_2, \cdots, v_{(2a-1)}, v_{2a}) = \mathsf{FFAssign}((r_0, r_1), \alpha)$.
5. Output $v^* = \mathsf{NIPM}_{2a}(x, v_1 \circ \cdots \circ v_{2a})$.

Theorem 37 (IdACB). *For every* $0 < \varepsilon < 1$ *and* $n \in \mathbb{N}$ *and every* k, d, t, a*, there exists a large enough* C *such that if*

- $k \geq C \max\left\{\left((t+1)^2 \log(1/\varepsilon) n^{4a-1}/m^3\right)^{1/(4a-3)}, (t+1)\sqrt{n}\right\}$;
- $d \geq C(t+1)^2 \max\left\{(t+1)n\sqrt{n}/k, ((t+1)\log(1/\varepsilon) n^{4a-1}/k^{4a-3})^{1/3}\right\}$,

then there exists a constant $1 > \eta > 0$ *and an* $\mathsf{IdACB} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^a \to \{0,1\}^m$ *which is a low-degree advice correlation breaker for linearly correlated sources s.t.*

- $m = \Omega(\eta^{2a} kd/((t+1)^3 n))$;
- *each output bit of* IdACB *is a degree* $2^{\Theta(2a)}$ *polynomial of the input.*

Proof. We will prove that Algorithm 7 gives such a function.
First we prove that IdACB satisfy Definition 26.

1. Let $Q_A := \mathsf{LSExt}(A, S)$, $Q'_A := \mathsf{LSExt}(A', S')$, $Q_B := \mathsf{LSExt}(B, S)$, $Q'_B := \mathsf{LSExt}(B', S')$. Also for all $i \in [t]$, let $Q^i_A := \mathsf{LSExt}(A^i, S^i)$, let $Q^i_B := \mathsf{LSExt}(B^i, S^i)$.
2. Since $Y = U_d \mid Y'$, then $S = U_{m_1} \mid Y'$. Since $H(X \mid Y', S, S^{[t]}) \geq H(A) \geq k \geq k/2 + (t+2)m_2$, by Lemma 35 $Q \approx_{\varepsilon_1} U_{m_2} \mid (Q', Y', S, S^{[t]})$.
3. First note that conditioned on S, since LSExt is a linear function, $Q = Q_A + Q_B$. Moreover, we have that Y is independent of Q_A further conditioned on Q_B. Since $H(Y \mid Y', S, S^{[t]}, Q_B, Q_B^{[t]}, Q') \geq d - (t+1)(m_1+m_2) \geq d/3$ and $Q_A \approx_{\varepsilon_1} U_{m_2} \mid (Q', Y', S, S^{[t]}, Q_B, Q_B^{[t]})$,

$$R_0 \approx_{\varepsilon_1+\varepsilon_2} U_v \mid (S, S^{[t]}, Q, Q^{[t]}, Q', Y', R'_0)$$

and

$$R_1 \approx_{\varepsilon_1+\varepsilon_2} U_v \mid (S, S^{[t]}, Q, Q^{[t]}, Q', Y', R_0, R'_0, R_0^{[t]}, R'_1).$$

4. By the Definition 27, $V_i \approx_{\varepsilon_1+\varepsilon_2} U_v \mid V', \forall i \in [2a]$. In addition, for every $i \in [t]$, there exists $h_i \in [2a]$ s.t. $V_{h_i} = R_1$ and $V^i_{h_i} = R^i_0$. Therefore for every $i \in [t]$, there exists $h_i \in [2a]$ s.t. $V_{h_i} \approx_{\varepsilon_1+\varepsilon_2} U_v \mid V'_{h_i}$.

5. Since $H(X \mid Q, Q', Q^{[t]}, Y', S, S^{[t]}, V'^*) \geq k - (2+t)m_2 - 2av \geq k/2$, by Theorem 34, $V^* \approx_{\varepsilon_1+\varepsilon_2+\varepsilon_3} U_{n_2} \mid (V^{[t]*}, V'^*)$.

Now since LSExt and IaExt cause a constant increase in the degree of the output bits, and NIPM_{2a} cause a $2^{\Theta(2a)}$ increase in the degree. each output bit of IdACB is a degree $2^{\Theta(\ell)}$ polynomial of the input.
Finally, the parameters constraints follows from those of Theorem 34. This completes the proof of Theorem 37. □

5.2 Directional Affine Extractor for Linear Entropy

Apart from the low-degree correlation breaker, we still need the following extractors as building blocks.

Theorem 38 ([CG88]). *For every constant $\delta > 0$, there exists a polynomial time algorithm $\mathsf{IP} : (\{0,1\}^n)^2 \to \{0,1\}^m$ such that if X is an (n, k_1) source, Y is an independent (n, k_2) source and $k_1 + k_2 \geq (1+\delta)n$, then*

$$\mathsf{IP}(X, Y) \approx_\varepsilon U_m \mid Y,$$

where $\varepsilon = 2^{-\frac{\delta n - m - 1}{2}}$.

Theorem 39 ([Li11]). *For every affine $t \times r$ somewhere random source X, there exists a function $\mathsf{AffineSRExt}$ such that $\mathsf{AffineSRExt}(X)$ outputs $m = r/t^{O(\log t)}$ bits that are $2^{-\Omega(r/t^{O(\log t)})}$-close to uniform. Moreover, each bit of the output is a degree $t^{O(1)}$ polynomial of the bits of the input.*

Theorem 40 (Seeded non-malleable extractor [Li12]). *For any constant $1 > \delta > 0$, let X be an (n, k)-source with $k = (1/2 + \delta)n$ and Y be the uniform distribution on $\{0,1\}^{n/2-1}$ independent of X. Let $b_1, \cdots, b_{n/2}$ be a basis of $\mathsf{F}_{2^{n/2}}$ regarded as a vector space over F_2. For each b_i, let $\overline{Y}_i = \left(b_i Y, b_i Y^3\right)$ where Y is regarded as an element in $\mathsf{F}^*_{2^{n/2}}$ and define one bit $Z_i = \mathsf{IP}(X, \overline{Y}_i)$ where IP is the inner product function over F_2^n. Choose $m = \Omega(n)$ bits from $\{Z_i\}$, let $\mathsf{snmExt}(X, Y) = (Z_{i_1}, \cdots, Z_{i_m})$. Let $\mathcal{A} : \{0,1\}^n \to \{0,1\}^n$ be any function without fixed point, then*

$$|\mathsf{snmExt}(X, Y), \mathsf{snmExt}(X, \mathcal{A}(Y)), Y - U_n, \mathsf{snmExt}(X, \mathcal{A}(Y)), Y| \leq 2^{-\Omega(n)}.$$

The following proposition about strong linear seeded extractors and affine sources is useful to us.

Proposition 41 ([Rao09]). *Let $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a linear strong seeded extractor for min-entropy k with error $\varepsilon < 1/2$. Let X be any affine source with entropy k. Then*

$$\mathbf{Pr}_{u \leftarrow_U U_d}[\mathsf{Ext}(X, u) = U_m] \geq 1 - \varepsilon.$$

We use the following lemma when arguing about strong seeded extractors with deficient seed.

Lemma 42 ([CGL16]). *Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be strong seeded extractor for min-entropy* k*, and error* ε*. Let* X *be a* (n,k)*-source and let* Y *be a source on* $\{0,1\}^d$ *with min-entropy* $d - \lambda$*. Then*

$$\mathsf{Ext}(X,Y) \approx_{2^\lambda \varepsilon} U_m \mid Y.$$

We now present our construction of directional affine extractor.

Algorithm 8 $\mathsf{DAExt}(x)$

Input: x — an n bit string.
Output: z — an m bit string with $\Omega(n)$.

Sub-Routines and Parameters:
Let $\ell_1 = \mathsf{poly}(2/\delta), \ell_1' = (n/(2(m'+k+1)))^{\log(2d_{19}+2)}$ where k is defined below, $\ell_2 = \mathsf{poly}(4/\delta) = \ell_3$. Let $\mathsf{BasicCond}$ be the basic condenser from Algorithm 1.
Let $\mathsf{SCond}_i : \{0,1\}^n \to \left(\{0,1\}^{n/\ell_i^{1/\log(2d_{19}+2)}}\right)^{\ell_i}$ for $i \in \{1,3\}$, $\mathsf{SCond}_2 : \{0,1\}^{n/t} \to \left(\{0,1\}^{n/\left(t\ell_2^{1/\log(2d_{19}+2)}\right)}\right)^{\ell_2}$, be linear affine condensers from Theorem 18.
Let $\mathsf{IP} : \left(\{0,1\}^{n/\left(t\ell_2^{1/\log(2d_{19}+2)}\right)}\right)^2 \to \{0,1\}^{\Omega(n)}$ be the two-source extractor from Theorem 38 with error $\varepsilon_1 = 2^{-\Omega(n)}$, set up to extract from two independent sources whose entropy rates sum up to more than $1 + 2\beta_{18}$.
Let $\mathsf{AffineSRExt}$ be the extractor for affine somewhere random sources from Theorem 39 with error $\varepsilon_2 = 2^{-\Omega(n)}$.
Let $\mathsf{LSExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{m'}$ be the strong linear seeded extractor from Theorem 32 set to extract from entropy $\delta n/2$ with error $\varepsilon_3 = 2^{-\Omega(n)}$.
Let $\mathsf{Enc} : \{0,1\}^n \to \{0,1\}^{\lambda n}$ be the encoding function of an asymptotically good linear binary code with constant relative rate $1/\lambda$ and constant relative distance β.
Let $\mathsf{snmExt} : \{0,1\}^{2(m'+k+1)} \times \{0,1\}^{m'+k} \to \{0,1\}^{n_1}$ be the seeded non-malleable extractor from Theorem 40 with error $\varepsilon_4 = 2^{-\Omega(n)}$. Choose m' and k such that $\log(n/(m'+k+1)) \in \mathbb{N}$ where

- $m' \leq \beta_{18}\delta^2 n/(300t\ell_2\ell_3')$, $k = \Omega(n) \leq \frac{n_1}{20\log(\lambda n/n_1)}$.

Let $\mathsf{IdACB} : \{0,1\}^n \times \{0,1\}^{n_1} \times \{0,1\}^{\log \ell_1'} \to \{0,1\}^{n_2}$ be the advice correlation breaker from Theorem 37 with output length $n_2 = O(\delta n_1/\mathsf{poly}(\ell_1'))$ and error $\varepsilon_5 = 2^{-\Omega(n)}$.

Let G be the generating matrix of an asymptotically good linear binary code with codeword length m_1 and constant relative distance γ. Thus G is an $\alpha m_1 \times m_1$ matrix for some constant $\alpha > 0$. Let G_i stand for the i'th row of the matrix.

Let $sc_1 \circ sc_2 \circ \cdots \circ sc_{\ell_1'} = \mathsf{BasicCond}^r \circ \mathsf{SCond}_1(x)$, where $r = \log(n/(m'+k+1)) - 1 - \ell_1^{1/\log(2d_{19}+2)}$. Divide x into t blocks $x = x_1 \circ \cdots \circ x_t$ where $t = 2^{\lceil \log(10/\delta) \rceil} \geq \delta/10$ and each block has n/t bits. For every i, $1 \leq i \leq t$ do the following.

1. Let $y_{i1} \circ \cdots \circ y_{i\ell_2} = \mathsf{SCond}_2(x_i)$, where y_{ij} is the j'th row of the matrix obtained by applying SCond_2 to x_i. Note that $\ell_2 = O(1)$ and each y_{ij} has $\Omega(n)$ bits.
2. Apply $\mathsf{BasicCond}^{\log t} \circ \mathsf{SCond}_3$ on x. That is, first apply SCond_3 on X, and then apply $\mathsf{BasicCond}$ $\log t$ times on the output so that we get ℓ_3' blocks $\mathsf{BasicCond}^{\log t} \circ \mathsf{SCond}_3(x) = x_1' \circ \cdots \circ x_{\ell_3'}'$, of equal size with each block having the same number of bits as y_{ij}. Note that $\ell_3' = O(1)$.
3. Apply IP to every pair of x_{j_1}' and y_{ij_2}, and output $\beta_{18}\delta^2 n/(300t\ell_2\ell_3')$ bits. Let sr_i be the matrix obtained by concatenating all the outputs $\mathsf{IP}(x_{j_1}', y_{ij_2})$, i.e., each row of sr_i is $\mathsf{IP}(x_{j_1}', y_{ij_2})$ for a pair (x_{j_1}', y_{ij_2}).
4. Let $r_i = \mathsf{AffineSRExt}(sr_i)$.
5. Let $u_i = \mathsf{LSExt}(x, r_i)$, set up to output m' bits.
6. Divide u_i into $u_{i1} \circ u_{i2}$ where u_{i1} has $k\log(\lambda n/k) \leq n_1/10$ bits and u_{i2} has $\geq m' - n_1/10$ bits.
7. Divide $\mathsf{Enc}(x)$ into k blocks of equal size such that $\mathsf{Enc}(x) = \tilde{x}^1 \circ \tilde{x}^2 \circ \cdots \circ \tilde{x}^k$ where each block has $O(1)$ bits. Divide u_{i1} into k equal blocks $u_{i1}^{(1)} \circ \cdots \circ u_{i1}^{(k)}$. Let $h_i = \tilde{x}^1_{|u_{i1}^{(1)}} \circ \tilde{x}^2_{|u_{i1}^{(2)}} \circ \cdots \circ \tilde{x}^k_{|u_{i1}^{(k)}}$ and $\tilde{u}_i = u_i \circ h_i$.
8. Let sn_{ij_3} be snmExt applied to each sc_{j_3} and $\tilde{u}_i$ and output $n_1 \leq m'/100$ bits. Let sn_i be the $\ell_1' \times n_1$ matrix obtained by concatenating sn_{ij_3} for $j_3 \in [\ell_1']$, i.e., the j-th row of sn_i is sn_{ij}.
9. Let $\tilde{y}_i = \bigoplus_{j=1}^{\ell_1'} \mathsf{IdACB}(x, sn_{ij}, j)$ and output $n_2 \leq m'/10000$ bits.
10. Let $w_i = \mathsf{LSExt}(x, \tilde{y}_i)$, set up to output $n_3 \leq m'/1000000$ bits.
11. Divide the bits of w_i into $s_i = \Omega(n)$ blocks of equal size, with each block having c_i number of bits for some constant c_i to be chosen later. For every $j = 1, \cdots, s_i$, compute one bit v_{ij} by taking the product of all the bits in the j'th block, i.e., $v_{ij} = \prod_{(j-1)c_i+1}^{jc_i} w_{i\ell}$.

Output $m_1 = \Omega(n)$ bits $\{z_j = \bigoplus_{i=1}^t v_{ij}\}$.

Disperser to Extractor.

For each codeword G_i, let $S_i = \{j \in [m_1] : \mathsf{G}_{ij} = 1\}$ be the set of indices s.t. the bit of the codeword G_i at those indices are 1. Define

$$o_i = \bigoplus z_{j:j\in S_i}$$

to be the bit associated with G_i, i.e., o_i is the XOR of the z_j's whenever the j'th index of the codeword G_i is 1.

Take a constant $0 < \beta' \leq \alpha$, where β' is chosen later. Output $o = (o_1, \cdots, o_{\beta' m_1})$.

Theorem 43. *For any constant $0 < \delta \leq 1$, there exists a family of functions $\mathsf{DAExt} : \{0,1\}^n \to \{0,1\}^m$ where $m = \Omega(n)$, such that for any affine source X of min-entropy at least δn, any nonzero $a \in \{0,1\}^n$, it holds that*

$$(\mathsf{DAExt}(X), \mathsf{DAExt}(X+a)) \approx_\varepsilon (U_m, \mathsf{DAExt}(X+a)),$$

where $\varepsilon = 2^{-\Omega(n)}$.

Proof. In the proof below, we have in mind two DAExt running in parallel, one with input X, the other with input $X + a =: X'$. We use $\{X_i\}_{i\in[t]}$, $\{SR_i\}_{i\in[t]}$, $\{R_i\}_{i\in[t]}$, $\{U_i\}_{i\in[t]}$, $\{\tilde{U}_i\}_{i\in[t]}$, $\{SN_i\}_{i\in[t]}$, $\{SC_i\}_{i\in[\ell_1]}$, $\{\tilde{Y}_i\}_{i\in[t]}$, $\{W_i\}_{i\in[t]}$ to denote the random variables generated in $\mathsf{DAExt}(X)$ and $\{X'_i\}_{i\in[t]}$, $\{SR'_i\}_{i\in[t]}$, $\{R'_i\}_{i\in[t]}$, $\{U'_i\}_{i\in[t]}$, $\{\tilde{U}'_i\}_{i\in[t]}$, $\{SN'_i\}_{i\in[t]}$, $\{SC'_i\}_{i\in[\ell_1]}$, $\{\tilde{Y}'_i\}_{i\in[t]}$, $\{W'_i\}_{i\in[t]}$ to denote the random variables generated in $\mathsf{DAExt}(X+a)$.

Throughout the proof, we maintain a random variable Z. We update Z each time a group of random variables has been fixed so that it represents all the random variables that have been fixed. We will make Z explicit each time it is revised. Initially, $Z = 0$.

We now show that Algorithm 8 is an efficient family of such functions. We first argue there exists an iteration g such that conditioned on all the random variables generated in the previous iterations, both X and X_g have $\Omega(\delta)$ entropy rate.

Lemma 44. *There exists $1 \leq g \leq t$ s.t. conditioned on any fixings of*

$$(X_i, X'_i, SR_i, SR'_i, R_i, R'_i, \tilde{U}_i, \tilde{U}'_i, SN_i, SN'_i, \tilde{Y}_i, \tilde{Y}'_i, W_i, W'_i)_{i\in[g-1]}$$

in order, X is an affine source with $H(X_g) \geq \delta n/(4t)$ and $H(X) \geq 3\delta n/5 + \delta n/(3t)$.

Proof. By Lemma 10, when dividing X into t blocks, there exist positive integers $k_1, \cdots, k_t$ which sum up to δn such that for any $i \in [t]$, conditioned on the fixing of $X_1, \cdots, X_{i-1}$, $H(X_i) = k_i$. Therefore, there must exists an i such that $k_i \geq \delta n/(3t)$. Let g be the minimal index such that $H(X_g) = k_g \geq \delta n/(3t)$.

1. Consider the affine source X and $X' = X + a$. Once we fix $(X_i = x_i)_{i\in[g-1]}$, $(X'_i := X_i + a_i = x_i + a_i)_{i\in[g-1]}$ are also fixed. Since X_i is an affine funciton of X, after this fixing, X and X' are still affine sources. By Lemma 10, after this fixing $H(X_g) = H(X'_g) \geq k_g \geq \delta n/(3t)$ and $H(X) = H(X') = \sum_{i=g}^{t} k_i \geq \delta n - (t-1)\cdot \delta n/(3t) \geq 2\delta n/3 + \delta n/(3t)$. Now set $Z = \{X_i, X'_i\}_{i\in[g-1]}$.

2. Note that conditioned on the fixing of $(X_i = x_i)_{i\in[g-1]}$ (thus $(X'_i = x'_i)_{i\in[g-1]}$), both $(SR_i)_{i\in[g-1]}$ and $(SR'_i)_{i\in[g-1]}$ are affine functions of X. In general, fixing $(SR_i = sr_i)_{i\in[g-1]}$ does not necessarily fix $(SR'_i = sr'_i)_{i\in[g-1]}$ and in the worst cases $SR'_i = sr'_i$ may be linearly independent with $SR_i = sr_i$. Let $\overline{SR} = SR_1 \circ \cdots \circ SR_{g-1}$ and $\overline{SR'} = SR'_1 \circ \cdots \circ SR'_{g-1}$. By Lemma 9, since $\overline{SR} \circ \overline{SR'}$ has at most $(\beta_{18}\delta^2 n/(300t)) \cdot t \cdot 2 = \beta_{18}\delta^2 n/150$ bits, $H(X \mid Z, \overline{SR} \circ \overline{SR'}) \geq 2\delta n/3 + \delta n/(3t) - H(\overline{SR} \circ \overline{SR'}(X)) \geq 2\delta n/3 + \delta n/(3t) - \beta_{18}\delta^2 n/150$.
Note that fixing $\overline{SR} \circ \overline{SR'}$ also fixes $\{R_i, R'_i\}_{i\in[g-1]}$. Now let $Z = Z \cup \{SR_i, SR'_i, R_i, R'_i\}_{i\in[g-1]}$.

3. Let $\overline{\tilde{U}} \circ \overline{\tilde{U}'} = \tilde{U}_1 \circ \cdots \circ \tilde{U}_{g-1} \circ \tilde{U}'_1 \cdots \circ \tilde{U}'_{g-1}$, then conditioned on any fixing of Z, $\overline{\tilde{U}}$ is an affine function of X and it has at most $(\beta_{18}\delta^2 n/(300t\ell_2\ell'_3)) \cdot t \cdot 2 = \beta_{18}\delta^2 n/(150\ell_2\ell'_3)$ bits. Therefore, by Lemma 9 $H(X \mid Z, \overline{\tilde{U}} \circ \overline{\tilde{U}'}) \geq 2\delta n/3 + \delta n/(3t) - \beta_{18}\delta^2 n/150 - \beta_{18}\delta^2 n/(150\ell_2\ell'_3)$. Now, $Z = Z \cup \{U_i, \tilde{U}_i\}_{i\in[g-1]}$.

4. Let $\overline{SN} \circ \overline{SN'} = SN_1 \circ \cdots \circ SN_{g-1} \circ SN'_1 \cdots \circ SN'_{g-1}$, then conditioned on any fixing of Z, $\overline{SN} \circ \overline{SN'}$ is an affine function of X and it has at most $\beta_{18}\delta^2 n/(15000\ell_2\ell'_3)$ bits. Therefore, by Lemma 9 $H(X \mid Z, \overline{SN} \circ \overline{SN'}) \geq 2\delta n/3 - \beta_{18}\delta^2 n/150 - 1.01 \cdot \beta_{18}\delta^2 n/(150\ell_2\ell'_3)$. Now, $Z = Z \cup \{\overline{SN}, \overline{SN'}\}$.

5. Let $\overline{\tilde{Y}} \circ \overline{\tilde{Y'}} = \tilde{Y}_1 \circ \cdots \circ \tilde{Y}_{g-1} \circ \tilde{Y}'_1 \cdots \circ \tilde{Y}'_{g-1}$, then conditioned on any fixing of Z, $\overline{\tilde{Y}} \circ \overline{\tilde{Y'}}$ is an affine function of X and it has at most $\beta_{18}\delta^2 n/(1500000\ell_2\ell'_3)$ bits. Therefore, by Lemma 9 $H(X \mid Z, \overline{\tilde{Y}\tilde{Y'}}) \geq 2\delta n/3 + \delta n/(3t) - \beta_{18}\delta^2 n/150 - 1.0101 \cdot \beta_{18}\delta^2 n/(150\ell_2\ell'_3)$. Now, $Z = Z \cup \{\overline{\tilde{Y}\tilde{Y'}}\}$.

6. Let $\overline{W} \circ \overline{W'} = W_1 \circ \cdots \circ W_{g-1} \circ W'_1 \cdots \circ W'_{g-1}$, then conditioned on any fixing of Z, $\overline{W} \circ \overline{W'}$ is an affine function of X and it has at most $\beta_{18}\delta^2 n/(150000000\ell_2\ell'_3)$ bits. Therefore, by Lemma 9 $H(X \mid Z, W, W') \geq 2\delta n/3 + \delta n/(3t) - \delta^2 n/150 - 1.010101 \cdot \beta_{18}\delta^2 n/(150\ell_2\ell'_3)$. Now, $Z = Z \cup \{W, W'\}$.

7. Therefore, $H(X_g \mid Z) \geq \delta n/(3t) - \beta_{18}(\delta^2 n/150 + \delta^2 n/(150\ell_2\ell'_3) + \delta^2 n/(15000\ell_2\ell'_3) + \delta^2 n/(1500000\ell_2\ell'_3) + \delta^2 n/(150000000\ell_2\ell'_3)) = \delta n/(3t) - \beta_{18}\delta^2 n/150 - 1.010101 \cdot \beta_{18} \cdot \delta^2 n/(150\ell_2\ell'_3) > \delta n/(4t)$ and $H(X \mid Z) \geq 2\delta n/3 + \delta n/(3t) - \beta_{18}\delta^2 n/150 - 1.010101 \cdot \beta_{18}\delta^2 n/(150\ell_2\ell'_3) > 3\delta n/5 + \delta n/(3t)$.

□

Lemma 45. *With probability $1 - 2^{-\Omega(n)}$ over the further fixings of X_g, R_g is $2^{-\Omega(n)}$-close to uniform.*

Proof. We examine the execution of DAExt on the good block X_g up to step 4.

1. By Lemma 44, $H(X_g \mid Z) \geq \delta n/(4t) \overset{\text{Theorem 18}}{\Longrightarrow} Y_{g1} \circ \cdots \circ Y_{g\ell_2} := \mathsf{SCond}(X_g) =$ somewhere-rate-$(1/2 + \beta_{18})$ source. WLOG, assume Y_{gi} has rate $1/2 + \beta_{18}$.

2. By Lemma 8, $\exists A_g, B_g$ s.t. $X = A_g + B_g, X_g(X) = X_g(A_g), H(X_g) = H(A_g)$, and X_g is independent with B_g.

3. After fixing X_g, B_g (thus X) has min-entropy at least $3\delta n/5 + \delta n/(3t) - \delta n/t \geq \delta n/4$. Now, let $Z = Z \cup \{X_g\}$.

4. Since the ℓ'_3 blocks $X = \tilde{X}_1 \circ \cdots \circ \tilde{X}_{\ell'_3}$ are obtained by applying $\mathsf{BasicCond}^{\log t} \circ \mathsf{SCond}_3$ on X, each $\tilde{X}_i$ is linear in X. Then $\tilde{X}_i(X) = \tilde{X}_i(A_g) + \tilde{X}_i(B_g)$ follows from Lemma 8. Let $\tilde{X}_i(A_g) = A_{gi}$ and $\tilde{X}_i(B_g) = B_{gi}$. By Theorem 18, there exists one block with entropy rate at least $1/2 + \beta_{18}$, let B_{gj} be such a block.

5. Note that $\mathsf{IP}(\tilde{X}_j, Y_{gi}) = \mathsf{IP}(A_{gj}, Y_{gi}) + \mathsf{IP}(B_{gj}, Y_{gi})$. Since Y_{gi} has entropy rate $\geq 1/2 + \beta_{18}$ and B_{gj} has entropy rate $\geq 1/2 + \beta_{18}$, by Lemma 38, with probability $(1 - \varepsilon_1)$ over the fixing of Y_{gi} (thus A_g and X_g), $\mathsf{IP}(B_{gj}, Y_{gi})$ is ε_1 close to uniform. Since A_g (thus A_{gj}) is fixed, the random variable $\mathsf{IP}(A_{gj}, Y_{gi})$ is fixed as well. Therefore, $\mathsf{IP}(B_{gj}, Y_{gi})$ is ε_1 close to uniform implies that $\mathsf{IP}(Y_{gi}, \tilde{X}_j)$ is ε_1 close to uniform. Therefore, with probability $(1 - \varepsilon_1 - \varepsilon_2)$ over the fixing of X_g, SR_g is $(\varepsilon_1 + \varepsilon_2)$ close to a somewhere random source.

6. $SR_g \approx_{\varepsilon_1+\varepsilon_2}$ somewhere random source $\overset{\text{Theorem 39}}{\Longrightarrow} R_g \approx_{\varepsilon_1+\varepsilon_2}$ uniform.

□

Lemma 46. *With probability $1 - 2^{-\Omega(n)}$ over further fixings of $\left(SR_g, SR'_g, R_g, R'_g\right)$, U_g is uniform.*

Proof. Since X_g is a linear function of X, conditioned on any fixing of it, it still holds that X is an affine source. Moreover, conditioned on the fixing of X_g (and thus X'_g as well), SR_g and SR'_g are linear functions of X. By Lemma 8, there exists independent affine sources $\tilde{A}_g$ and $\tilde{B}_g$ s.t. $X = \tilde{A}_g + \tilde{B}_g$, $SR_g \circ SR'_g(X) = SR_g \circ SR'_g(\tilde{A}_g)$ and $H(SR_g \circ SR'_g) = H(\tilde{A}_g)$. Thus $H(\tilde{B}_g) = H(X) - H(\tilde{A}_g) = H(X) - H(SR_g \circ SR'_g) \geq \delta n/2$.
Next note R_g is a deterministic function of SR_g thus independent of $\tilde{B}_g$. In addition, R_g is $2^{-\Omega(n)}$-close to uniform by Lemma 45. Now, by Theorem 32 and Proposition 41, with probability $(1-\varepsilon_3)$ over the fixings of R_g (and thus with probability $(1-\varepsilon_3)$ over the fixings of SR_g), $\mathsf{LSExt}(\tilde{B}_g, R_g)$ is uniform. Since LSExt is a linear function and $\mathsf{LSExt}(\tilde{A}_g, R_g)$ is fixed, with probability $(1-\varepsilon_3)$ over the fixings of R_g, $\mathsf{LSExt}(X, R_g)$ is uniform. □

At this point, set $Z = Z \cup \{SR_g, SR'_g, R_g, R'_g\}$. We have already shown that with high probability over the fixing of Z, U_g is uniform. Now, we want to establish that $\tilde{U}_g$ and $\tilde{U}'_g$, which are U_g and U'_g appended with advice are linearly correlated, i.e., there exists an affine function $\mathcal{A}$ without fixed points s.t. $\mathcal{A}(\tilde{U}_g) = \tilde{U}'_g$. We achieve this in the following two Lemmas.

Lemma 47. *Conditioned on Z, there exists $\tilde{A}_g, \tilde{B}_g$ s.t. $X = \tilde{A}_g + \tilde{B}_g$, $U_g(X) = U_g(\tilde{A}_g)$, and $U_g(\tilde{B}_g) = 0$. Moreover, U'_g is linearly correlated with U_g conditioned on any fixing of $U'_g(\tilde{B}_g)$.*

Proof. By Lemma 8, there exists $\tilde{A}_g, \tilde{B}_g$ s.t. $X = \tilde{A}_g + \tilde{B}_g$, $U_g(X) = \tilde{A}_g$, and $U_g(\tilde{B}_g) = 0$. Moreover, there exists an affine function L such that $\tilde{A}_g = L(U_g)$. Now conditioned on the fixing of $U'_g(\tilde{B}_g)$, U'_g is an affine function of $\tilde{A}_g$, and thus an affine function of U_g. □

As a reminder, at this stage, we have

$$Z = \{X_i, X'_i, SR_i, SR'_i, R_i, R'_i, \tilde{U}_i, \tilde{U}'_i, SN_i, SN'_i, \tilde{Y}_i, \tilde{Y}'_i, W_i, W_i\}_{i\in[g-1]} \cup \{X_g, X'_g, SR_g, SR'_g, U'_g(\tilde{B}_g)\}.$$

Lemma 48. *Conditioned on the event that U_g is uniform, with probability at least $1 - 2^{-\Omega(n)}$ over the fixing of $(Z, U_{g1}, U'_{g1}, H_g, H'_g)$, there exists an affine map $\mathcal{A} : \{0,1\}^{|\tilde{U}_g|} \to \{0,1\}^{|\tilde{U}_g|}$ without fixed point such that $\tilde{U}'_g = \mathcal{A}(\tilde{U}_g)$.*

Proof. We first show that $\tilde{U}_g \neq \tilde{U}'_g$ with high probability.

- If $U_{g1} \neq U'_{g1}$, then $\tilde{U}_g(X) \neq \tilde{U}'_g(X)$ always holds.
- If $U_{g1} = U'_{g1}$, we show that $H_g \neq H'_g$ with probability $1 - 2^{-\Omega(n)}$. Note that this is equivalent to $H_g \oplus H'_g \neq 0 \iff (\tilde{X}^1 \oplus \tilde{X}'^1)_{|U^{(1)}_{g1}} \circ (\tilde{X}^2 \oplus \tilde{X}'^2)_{|U^{(2)}_{g1}} \circ \cdots \circ (\tilde{X}^k \oplus \tilde{X}'_k)_{|U^{(k)}_{g1}} \iff \tilde{a}^1_{|U^{(1)}_{g1}} \circ \tilde{a}^2_{|U^{(2)}_{g1}} \circ \cdots \circ \tilde{a}^k_{|U^{(k)}_{g1}}$ with probability $1 - 2^{-\Omega(n)}$ where $\tilde{a}^j$ for $j \in [k]$ are obtained by divide the $\mathsf{Enc}(a)$ (i.e. the encoded shift between the source X and its tampering $X' = X + a$) into k equal blocks such that $\mathsf{Enc}(a) = \tilde{a}^1 \circ \tilde{a}^2 \circ \cdots \circ \tilde{a}^k$. Let $\ell_1, \cdots, \ell_k$ be the number of bits in each block that are non-zero. Since at least β fraction of bits in $\mathsf{Enc}(a)$ differs from the bits of the

codeword $0 = \mathsf{Enc}(0)$, $\sum_{i=1}^{k} \ell_i \geq \beta\lambda n$. Therefore we have

$$\begin{aligned}
\mathbf{Pr}\left[H_g \neq H'_g \mid U_{g1} = U'_{g1}\right] &= 1 - \prod_{i=1}^{k}\left(1 - \frac{\ell_i}{\lambda n/k}\right) \\
&\geq 1 - \left(\frac{\sum_{i=1}^{k}\left(1 - \frac{\ell_i}{\lambda n/k}\right)}{k}\right)^k \\
&\geq 1 - (1-\beta)^k \\
&\geq 1 - 2^{-\Omega(n)}. && (k = \Omega(n))
\end{aligned}$$

Therefore, in total, with probability $1 - 2^{-\Omega(n)}$, $\tilde{U}_g \neq \tilde{U}'_g$. By Lemma 47, U'_g is linearly correlated with U_g conditioned on Z. Now, note that $\tilde{U}_g$ is a composition of U_g with H_g with U_{g1}, H_g fixed. And the same holds for $\tilde{U}'_g$. Therefore there exists some affine map $\mathcal{A}$ such that $\mathcal{A}(\tilde{U}_g) = \tilde{U}'_g$. □

Lemma 49. *Conditioned on the further fixings of $(\tilde{U}_g, \tilde{U}'_g)$, there exists a constant $\beta > 0$ such that SC is a $(1/2 + \beta)$ affine somewhere random source.*

Proof. Let $Z = Z \cup \{\tilde{U}_g, \tilde{U}'_g\}$. It is easy to see that the bound $H(X \mid Z) \geq \delta n/2$ holds. Therefore, by Theorem 18, SC is a $(1/2 + \beta_{18})$ affine somewhere random source. □

Lemma 50. *Conditioned on U_g is uniform as well as $\tilde{U}_g$ and $\tilde{U}'_g$ are linearly correlated, $SN_g = \mathsf{snmExt}(SC_1, \tilde{U}_g) \circ \cdots \circ \mathsf{snmExt}(SC_{\ell'_1}, \tilde{U}_g)$ is $2^{-\Omega(n)}$ close to an affine somewhere random source. Moreover, there exists $h \in [\ell'_1]$ such that*

$$SN_{gh} \approx_{2^{-\Omega(n)}} U_{n_1}) \mid SN'_{gh}.$$

Proof. Since SC is a $(1/2 + \beta)$ affine somewhere random source, there exists an $h \in [\ell_3]$ such that $H(SC_h) \geq 1/2 + \beta$. For the seeds, the conditioning of (U_{g1}, U'_{g1}) cause a deficiency of at most $2^{0.22n_1}$ to $\tilde{U}_g$ from being uniform. Then by Theorem 40 and Lemma 42, conditioned on $(\tilde{U}_g, \tilde{U}'_g)$, we have

$$(\mathsf{snmExt}(SC_h, \tilde{U}_g) \approx_{2^{0.22n_1}\varepsilon_4} U_{n_1}) \mid \mathsf{snmExt}(SC'_h, \tilde{U}'_g).$$

Since $2^{0.22n_1}\varepsilon_4 = 2^{-\Omega(n)}$, snmExt is a linear function of SC conditioned on Z, it holds that

$$(SN_{gh} \approx_{2^{-\Omega(n)}} U_{n_1}) \mid SN'_{gh},$$

and SN_g is $2^{-\Omega(n)}$ close to an affine somewhere random source. □

Lemma 51. *With probability $1 - 2^{-\Omega(n)}$ over further fixings of $SN'_{gh} = \mathsf{snmExt}(SC'_h, \tilde{U}'_g)$, $\tilde{Y}_g \oplus \tilde{Y}'_g$ is uniform.*

Proof. First note that both X and SN_g (similarly X' and SN'_g) are affine sources. By Theorem 37, we have

$$\mathsf{IdACB}(X, SN_{gh}, h) \approx_{\varepsilon_5} U_{n_2} \mid \underbrace{\mathsf{IdACB}(X', SN'_{gh}, h)}_{\substack{\text{same advice but} \\ SN_{gh} \approx_{2^{-\Omega(n)}} U_{n_1} \mid SN'_{gh}}}, \underbrace{\left\{\mathsf{IdACB}(X, SN_{gj}, j), \mathsf{IdACB}(X', SN'_{gj}, j)\right\}}_{\substack{\text{the set contains the output } \forall j \in [\ell'_1]\setminus\{h\} \\ \text{different advice}}}.$$

Therefore, it holds that

$$\begin{aligned}\tilde{Y}_g \oplus \tilde{Y}'_g &= \bigoplus_{j\in[\ell'_1]} \mathsf{IdACB}(X, SN_{gj}, j) \oplus \bigoplus_{j\in[\ell'_1]} \mathsf{IdACB}(X', SN'_{gj}, j) \\ &= \mathsf{IdACB}(X, SN_{gh}, h) \oplus \left(\bigoplus_{j\in[\ell'_1]\setminus\{h\}} \mathsf{IdACB}(X, SN_{gj}, j) \oplus \bigoplus_{j\in[\ell'_1]} \mathsf{IdACB}(X', SN'_{gj}, j) \right) \\ &\approx_{2^{-\Omega(n)}} U_{n_2}.\end{aligned}$$

□

Let $Z = Z \cup \{SN'_{gh}\}$.

Lemma 52. *With probability $1 - 2^{-\Omega(n)}$ over the fixing of $(Z, \tilde{Y}_g, \tilde{Y}'_g)$, W_g is uniform conditioned on W'_g. Moreover, there exists a random variable $\hat{B}_g$ conditioned on which X is an affine function of W_g.*

Proof. By Lemma 35, $W_g \approx_{2^{-\Omega(n)}} U_{n_3} \mid W'_g$. Since conditioned on $(\tilde{Y}_g, \tilde{Y}'_g)$, W_g is a linear function of X, by Lemma 8, there exists $\hat{A}_g$ and $\hat{B}_g$ such that $W_g(X) = W_g(\hat{A}_g)$ and $W_g(\hat{B}_g) = 0$. Moreover, conditioned on the fixing of $\hat{B}_g$, X is an affine function of W_g. Now, let $Z = Z \cup \{\tilde{Y}_g, \tilde{Y}'_g, W'_g, \hat{B}_g\}$. □

Lemma 53. *For all $1 \le j \le m_1$, z_j is a constant degree polynomial of the bits of x.*

Proof.
We consider how the degree of the polynomial accumulates inside the for-loop of Algorithm 8.

1. According a similar argument to [Li11], each bit of u_i is a $O(1)$ degree polynomial of x.
2. In step 7, it suffices to consider each bit of h_i. For each $j \in [k]$, $\tilde{x}_j$ has $O(1)$ bits. Since to sample a bit from $\tilde{x}_j$, each $u_{i1}^{(j)}$ only needs to be $\log|\tilde{x}_j| = O(1)$ long to encode all the indices of $\tilde{x}_j$. Therefore, the j-th bit of h_i is a linear function of $\tilde{x}_j$ and a $\log|\tilde{x}_j| = O(1)$ degree polynomial of $u_{i1}^{(j)}$.
3. In step 8, each bit of sn_i is bilinear map on sc and $\tilde{u}_i$. Therefore, each bit of sn is a degree $O(1)$ polynomial of the bits of x.
4. In step 9, each bit of $\tilde{y}_i$ is a degree $O(1)$ polynomial of bits of x and sc according to Theorem 37.
5. In step 10, each bit of w_i is a constant degree polynomial of the inputs by Theorem 32.
6. In step 11, since each c_i is a constant, the degree of resulting monomials by taking products of c_i bits is constant. Therefore, each bit of v_{ij} for all $j \in [s_i]$ is a degree $O(1)$ polynomial of x.

Finally, it is direct that each bit of z_j is a constant degree polynomial of x for each $j \in [m_1]$. □

Lemma 54. *For any integer $s \in [m_1]$, let $\tilde{Z} = (\tilde{Z}_1, \cdots, \tilde{Z}_s)$ where $\tilde{Z}_j = \bigoplus_{i=g}^{t} V_{ij}$. Then conditioned on any fixing of W'_g and $\hat{B}_g = \hat{b}$, there exists some $b \in \{0,1\}^s$ such that*

$$\left|\mathsf{Supp}(\mathsf{DAExt}(X) \mid \mathsf{DAExt}(X') = b)\right| = 2^s.$$

Proof. First note that for all $i \in [t]$, each bit of W_i is a constant degree polynomial of the bit of X. Therefore, conditioned on the fixing of Z, for every $i \geq g+1$, each bit of $\tilde{W}_i$ and $\tilde{W}'_i$ is a degree $\leq c(\delta)$ polynomial of the bits of W_g. Thus, for every $i \geq g+1$, the degree of the bit in W_i and W'_i is $c(\delta)$ multiple of the degree of the bit in W_g. Therefore, if

$$c_i > c(\delta)c_{i+1}, \ \forall i$$

then the degree of the polynomials $\bigoplus_{i=g+1}^{t} V_{ij}$ and $\bigoplus_{i=g+1}^{t} V'_{ij}$ is less than the degree of V_{gj}. Therefore, there exists a fixing of $\{V_{ij}, V'_{ij}\}_{i\in[g+1-t],j\in[m]}$ such that $\tilde{Z}_j \oplus \tilde{Z}'_j$ can take both values in $\{0,1\}$. Since V'_{gj} is fixed, $\tilde{Z}'_j$ is fixed as well. This ensures there exists $\{\tilde{z}'_j\}_{j\in[m]}$ such that that $\tilde{Z}_j \mid (\tilde{Z}'_j = \tilde{z}'_j)$ can take both values in $\{0,1\}$.

Next we show that $\tilde{Z}_j$ take both values in $\{0,1\}$ conditioned on $\tilde{Z}_S$ where $S \subseteq [m] \setminus \{j\}$ where $\tilde{Z}_S$ denotes $(\tilde{Z}_i)_{i\in S}$. Assume that for some $(z_i)_{i\in S}$ such that when $(\tilde{Z}_i = z_i)_{i\in S}$, $\tilde{Z}_j$ is fixed to z_j, then it holds that

$$P_g = \prod_{i\in S} \left(\tilde{Z}_i + z_i + 1\right) \left(\tilde{Z}_j + z_j\right) \equiv 0.$$

However, this cannot be true since P_g has a monomial V_{gj} of bits from W_g that are different from the monomials of the same degree from $\tilde{Z}_i$ (if $z_j = 1$). Since W_g is uniform, V_{gj} is nonzero with any fixings of $\tilde{Z}_S$. Therefore P_g cannot always be 0. □

The techniques to bootstrap an extractor from a disperser follow essentially the same line as [Li11]. We restate them here for the completeness of the proof.

Lemma 55. *The random variables $O_1, \cdots, O_{\alpha m_1}$ form an ε-biased space.*

Proof. Let $\varnothing \neq T \subseteq [\alpha m_1]$, $S_i = \{j \in [m_1] : \mathsf{G}_{ij} = 1\}$, $S_T = \{j \in [m_1] : \oplus_{i\in T}\mathsf{G}_{ij} = 1\}$. Then

$$\bigoplus_{i\in T} O_i = \bigoplus Z_{j:j\in S_T}.$$

Since any non-zero linear combination of codewords is again a codeword. The set S_T has cardinality at least γm_1. Now note that conditioned on the fixing of Z, each O_i is a degree c_g polynomial of W_g. Moreover, $\forall i \in S_T$,

$$O_i = V_{gi} \oplus \bigoplus_{j=g+1}^{t} V_{ji}.$$

Since for all $i \in S_T$, V_{gi} is the product of some disjoint set (w.r.t. $V_{g\ell}$'s, $\forall \ell \in S_T \setminus \{i\}$) of the bits of W_g, it is easy to see that $\{V_{gi} : i \in S_T\}$ is a set of $|S_T| \geq \gamma m_1$ independent copies of the same function which we simply refer to as f. Since $P := \bigoplus_{i\in T} \left(\bigoplus_{j=g+1}^{t} V_{ji}\right)$ has degree less than $c_g - 1$ and f has degree c_g,

$$\mathsf{Cor}(f, P) \leq 1 - 2^{-c_g}.$$

Then by Theorem 17,

$$\mathsf{Cor}(f^{\oplus|S_T|}, P) \leq \exp(-\Omega(|S_T|/(4^{c_g-1} \cdot (c_g - 1)))) \leq 2^{-\Omega(\gamma m_1)}.$$

Since our choice of T is arbitrary, and there are at most $2^{\alpha m_1} - 1$ such choices, there exists an absolute constant c_0 s.t.

$$\mathsf{Cor}(f^{\oplus|S_T|}, P) \leq 2^{-c_0\gamma m_1)}$$

for any $\varnothing \neq T \subseteq [\alpha m_1]$.
Since $f^{\oplus|S_T|}$ is uniform, it holds that

$$\Delta(\bigoplus_{i\in T} O_i, U) \leq 2^{-c_0\gamma m_1}.$$

we conclude that $O_1, \cdots, O_{\alpha m_1}$ form an ε-biased space. □

Now by Lemma 16,

$$\Delta(O - U_{\beta' m_1}) \leq 2^{\beta' m_1/2} \cdot 2^{-c_0\gamma m_1}.$$

Choose $0 < \beta' \leq \alpha$ s.t. $\beta' \leq c_0\gamma$. Then

$$\Delta(O - U_m) \leq 2^{-c_0 m/2}.$$

Therefore, the output of Algorithm 8 are m bits that are $2^{-\Omega(m)}$-close to uniform. □

5.3 Directional Affine Disperser and Extractor for Sublinear Entropy Sources

In this subsection, we demonstrate how to push the entropy requirement of Algorithm 8 to sublinear. We first examine how we can do this for the disperser.

Theorem 56. *There exists a constant $c > 1$ and an efficient family of functions $\mathsf{DADisp} : \{0,1\}^n \to \{0,1\}^m$ such that $m = n^{\Omega(1)}$ and for every affine source X with entropy $cn(\log\log n)^2/\log n$, there exists some $b \in \{0,1\}^m$ such that $|\mathsf{Supp}(\mathsf{DAExt}(X) \mid \mathsf{DAExt}(X + a) = b)| = 2^m$.*

Proof Sketch. The disperser construction is Algorithm 8 up to the phase "Disperser to Extractor", except for now, we do not make assumptions about the entropy of the input. When pushing down the entropy requirement, we are mainly interested in 2 binding quantities — the bit length of W_g and the degree of each bit of $\{W_i : i \in [g+1, t]\}$.
We first examine the length of W_g in a step-by-step manner.
Step 1. In step 1 of Algorithm 8, by Theorem 18, we get that $\ell_2 = \mathsf{poly}(1/\delta)$, and each Y_{gj} has $n/\mathsf{poly}(1/\delta)$ bits.
Step 2. We get $\ell_3' = \mathsf{poly}(1/\delta)$.
Step 3. The total number of rows in the matrix SR_g is $\ell_2\ell_3' = \mathsf{poly}(1/\delta)$, with each row having $\delta^2 n/(300t\ell_2\ell_3') = n/(\mathsf{poly}(1/\delta))$ bits. By Theorem 38, the error is $2^{-n/\mathsf{poly}(1/\delta)}$.
Step 4. We apply $\mathsf{AffineSRExt}$. By Theorem 39, we get each R_g has $n/\mathsf{poly}(1/\delta)^{O(\log(1/\delta))}$ bits, with error $2^{-n/\mathsf{poly}(1/\delta)^{O(\log(1/\delta))}}$.
Step 5. By Theorem 32, after applying LSExt, U_g has $n/(1/\delta)^{O(\log(1/\delta))}$ bits with error $2^{-n/(1/\delta)^{O(\log(1/\delta))}}$.
Step 8. First note that by Theorem 18, SC_g has $n/(1/\delta)^{O(\log(1/\delta))}$ bits and SC has $\ell_1' = (1/\delta)^{O(\log(1/\delta))}$ rows. By Theorem 40, each row of SN_g has bits $n/(1/\delta)^{O(\log(1/\delta))}$ with error $2^{-n/(1/\delta)^{O(\log(1/\delta))}}$.
Step 9. By Theorem 37, $\tilde{Y}_g$ has $n/(1/\delta)^{O(\log(1/\delta))}$ bits with error $2^{-n/(1/\delta)^{O(\log(1/\delta))}}$.
Step 10. By Theorem 32, after applying LSExt, W_g has $n/(1/\delta)^{O(\log(1/\delta))}$ bits with error $2^{-n/(1/\delta)^{O(\log(1/\delta))}}$.
We now check if the degrees of the polynomials produced in **Step 11** satisfy the requirements as

in the analysis of Theorem 43, which adds constraints $c_i > c(\delta)c_{i+1}, \forall i$. First note that up to a sequence of fixings of r.v.s, X is an affine function of W_g. Now by Theorem 18, each bit of $\mathsf{SCond}_2(X_i)$ and $\mathsf{BasicCond}^{\log t} \circ \mathsf{SCond}_3(x)$ is a linear function of the input bits. The function IP is a degree 2 polynomial. Therefore each bit of SR_i is a degree 2 polynomial of the input bits. Since each bit of the output of $\mathsf{AffineSRExt}$ is a degree $\mathsf{poly}(1/\delta)$ polynomial of the input bits. Therefore each bit of R_i is a degree $\mathsf{poly}(1/\delta)$ of the bits of W_g. By Theorem 32, each bit of U_i is a constant degree polynomial of the input bits. Since Enc is linear and each bit of $\tilde{X}_i$ has $O(1)$ bits, each bit of h_i is a constant degree polynomial of the inputs. By Theorem 18 and Theorem 40, each bit of SN_{ij} is a constant degree polynomial of the input bits. By Theorem 37, each bit of $\tilde{Y}_i$ is a degree $2^{\log(\ell_1')} = 2^{\log((1/\delta)^{O(\log(1/\delta))})} = (1/\delta)^{O(\log(1/\delta))}$ degree polynomial of the input bits. By Theorem 32, each bit of W_i is a constant degree polynomial of the input bits. Thus, we conclude that for every $i \geq g+1$, each bit of W_i is a degree $(1/\delta)^{O(\log(1/\delta))}$ polynomial of the bits of W_g. Therefore, we have

$$c(\delta) = (1/\delta)^{O(\log(1/\delta))}.$$

Since we need $c_i > c(\delta)c_{i+1}$ for every $1 \leq i \leq 10/\delta$, we have the following upper bound for all the c_i's.

$$c(\delta)^{10/\delta} = ((1/\delta)^{O(\log(1/\delta))})^{O(1/\delta)} = (1/\delta)^{O((1/\delta)\log(1/\delta))}.$$

Since each W_i has $n/(1/\delta)^{O(\log(1/\delta))}$ bits, it suffices to have

$$n/(1/\delta)^{O(\log(1/\delta))} > (1/\delta)^{O((1/\delta)\log(1/\delta))}.$$

It suffices to take $\delta = c(\log\log n)^2/\log n$ for some constant c. □

We now discuss the case for the extractor.

Theorem 57. *There exists a constant $c > 1$ and an efficient family of functions $\mathsf{DAExt} : \{0,1\}^n \to \{0,1\}^m$ such that $m = n^{\Omega(1)}$ and for every affine source X with entropy $cn(\log\log\log n)^2/\log\log n$,*

$$(\mathsf{DAExt}(X), \mathsf{DAExt}(X+a)) \approx_\varepsilon (U_m, \mathsf{DAExt}(X+a)) \ ,$$

where $\varepsilon = 2^{-n^{\Omega(1)}}$.

Proof Sketch. We follow up on the discussion for sublinear entropy disperser. Assume that the entropy is set such that we indeed obtain a disperser. Note that the disperser outputs

$$n/\left((\log(1/\delta))^{O(1/\delta)} \cdot (1/\delta)^{O((1/\delta)\log(1/\delta))}\right) = n/(1/\delta)^{O((1/\delta)\log(1/\delta))}$$

bits. From this point, there is and only is one more constraint to consider which is on the degree of the polynomials. For the extractor, we need to guarantee that the XOR lemma from Theorem 17 yields subexponential error. In other words, we need to guarantee

$$\frac{n/(1/\delta)^{O((1/\delta)\log(1/\delta))}}{(1/\delta)^{O((1/\delta)\log(1/\delta))} \cdot 2^{(1/\delta)^{O((1/\delta)\log(1/\delta))}}} = n^{\Omega(1)}.$$

It suffices to take $\delta = c(\log\log\log n)^2/\log\log n$ for some constant c. □

6 Average-case AC^0 Hardness for Read-Once Branching Programs

In this section, we build an AC^0-computable extractor that are capable of extracting randomness from the preimage of any output of any read-once branching program of suitable size.

We use the following two constructions of extractors in AC^0 from previous works.

Theorem 58 ([CL18]). *For any constants $c \in \mathbb{N}$, $\delta \in (0,1]$, there exists an explicit deterministic $(k = \delta n, \varepsilon = 2^{-\log^c n})$-extractor $\mathsf{AC}^0\text{-BFExt} : \{0,1\}^n \to \{0,1\}^{\Omega(k)}$ that can be computed by AC^0 circuits of depth $O(c)$, for any (n,k)-bit-fixing source.*

Theorem 59 ([PWY16]). *For any constants $c \in \mathbb{N}$, $\delta \in (0,1]$, there exists an explicit strong linear seeded $(k = \delta n, \varepsilon = 2^{-\log^c n})$-extractor $\mathsf{AC}^0\text{-LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{\Omega(k)}$ that can be computed by AC^0 circuits of depth $O(c)$, with seed length $d = O(\log^{c+1} n)$.*

6.1 AC^0-Computable t-Affine Correlation Breaker

Our construction of AC^0-computable t-affine correlation breaker builds on the skeleton of the t-affine correlation breaker in [CL22], which in turn applies the standard correlation breaker in [Li17]. Towards this, we first give an AC^0-computable flip-flop, which is used as a subroutine in the standard correlation breaker. We then replace the strong seeded extractors in the standard correlation breaker and the t-affine correlation breaker with AC^0-LExt.

Algorithm 9 $\mathsf{AC}^0\text{-flip-flop}(x, y, b)$

Input: Uniform bit strings x, y of length n_1, n_1 respectively, a bit b and a circuit depth parameter $c \in \mathbb{N}$.
Output: Bit string $\hat{x}$ of length n_2.
Parameters and Subroutines: Let $n_2 = \Omega(n_1) \leq n_1/20$ and $d = \Omega(n_1) \leq n_2/10$. Let $\mathsf{AC}^0\text{-LExt}_1 : \{0,1\}^{n_1} \times \{0,1\}^d \to \{0,1\}^d$ be $(k_1 = n_1/10, \varepsilon_1 = 2^{-\log^c n})$-strong linear seeded extractor from Theorem 59, $\mathsf{AC}^0\text{-LExt}_2 : \{0,1\}^{n_2} \times \{0,1\}^d \to \{0,1\}^d$ be $(k_2 = n_2, \varepsilon_2 = 2^{-\log^c n})$-strong linear seeded extractors from Theorem 59. Let $\mathsf{AC}^0\text{-LExt}_3 : \{0,1\}^{n_1} \times \{0,1\}^d \to \{0,1\}^{n_2}$ be a $(k_3 = n_1/2, \varepsilon_3 = 2^{-\log^c n})$-strong linear seeded extractor from Theorem 59.
Let $\mathsf{laExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2+d} \to \{0,1\}^{2d}$ be a look-ahead extractor for an alternating extraction protocol run for 2 rounds using $\mathsf{AC}^0\text{-LExt}_1, \mathsf{AC}^0\text{-LExt}_2$ as the seeded extractors.

1. Let $\tilde{y} = \mathsf{Slice}(y, n_2)$, $s_0 = \mathsf{Slice}(\tilde{y}, d)$, $\mathsf{laExt}(x, (\tilde{y}, s_0)) = r_0, r_1$
2. Let $\overline{y} = \mathsf{AC}^0\text{-LExt}_3(y, r_b)$
3. Let $\overline{s_0} = \mathsf{Slice}(\overline{y}, d)$, $\mathsf{laExt}(x, (\overline{y}, \overline{s_0})) = \overline{r_0}, \overline{r_1}$
4. Let $\hat{y} = \mathsf{AC}^0\text{-LExt}_3(y, \overline{r_{1-b}})$
5. Let $y_0 = \mathsf{Slice}(\hat{y}, d)$
6. Output $\hat{x} = \mathsf{AC}^0\text{-LExt}_3(x, y_0)$

Theorem 60 (AC^0-flip-flop). *For any integer $c, n_1 > 0$ and any $\varepsilon > 0$, there exists an explicit function $\mathsf{AC}^0\text{-flip-flop} : \{0,1\}^{n_1} \times \{0,1\}^{n_1} \times \{0,1\} \to \{0,1\}^m$, satisfying the following: let X be an independent uniform source on n_1 bits, and X' be a random variable on n_1 bits arbitrarily correlated with X. Let Y be an independent uniform source on n_1 bits, and Y be a random variable on n_1 bits arbitrarily correlated with Y. Suppose (X, X') is independent of (Y, Y'). If $k = \Omega(n_1)$, then*

AC^0-flip-flop *can be computed by* AC^0 *circuits of depth* $O(c)$ *and for any bit* b*, it holds that*

$$\mathsf{AC}^0\text{-flip-flop}(X, Y, b) \approx_\varepsilon U_m \mid (Y, Y').$$

Furthermore, for any bits b, b' *with* $b \neq b'$*, we have*

$$\mathsf{AC}^0\text{-flip-flop}(X, Y, b) \approx_\varepsilon U_m \mid (\mathsf{AC}^0\text{-flip-flop}(X', Y', b'), Y, Y').$$

where $m \geq \Omega(k)$ *and* $\varepsilon = 6 \cdot 2^{-\log^c n_1}$.

Proof. We show that Algorithm 9 is a construction of such functions.

1. Since $S_0 = U_d$ and $\widetilde{H}_\infty(X) \geq k_1$, by the property of AC^0-LExt_1, conditioned on the fixings of S_0, $R_0 \approx_{\varepsilon_1} U_d$ is a linear function of X, thus of (Y, Y'). Since $\widetilde{H}_\infty(\tilde{Y} \mid S_0, S_0') \geq n_2 - 2d \geq k_2$ and $R_0 \approx_{\varepsilon_1} U_d$, by the property of AC^0-LExt_2, conditioned on the fixings of R_0, $S_1 \approx_{\varepsilon_1+\varepsilon_2} U_d$ is a linear function of Y', thus independent of X. Since $\widetilde{H}_\infty(X \mid R_0, R_0') \geq n_1 k - 2d \geq k_1$ and $S_1 \approx_{\varepsilon_1+\varepsilon_2} U_d$, by the property of AC^0-LExt_1, conditioned on S_1, $R_1 \approx_{\varepsilon_1+\varepsilon_2+\varepsilon_1} U_d$ is a linear function of X, thus independent of (Y, Y').

2. Since $R_b \approx_{\varepsilon_1+b(\varepsilon_1+\varepsilon_2)} U_d$ and R_b (R_b') is independent of Y (Y') conditioned on $\{S_0, S_1\}$ $(\{S_0', S_1'\})$. Fix (R_b, R_b') and $\overline{Y'}$, $\widetilde{H}_\infty(Y \mid S_0, S_0', S_1, S_1', \overline{Y'}) \geq n_1 - 4d - n_2 \geq k_3$, then by the property of AC^0-LExt_3, $\overline{Y} \approx_{\varepsilon_3+\varepsilon_1+b(\varepsilon_1+\varepsilon_2)} U_{n_2}$.

3. Now that $\overline{Y'}$ is fixed, we can fix $(\overline{R_0'}, \overline{S_1'}, \overline{R_1'})$. This only cause at most $2d$ entropy loss to X. Since $\overline{S_0}$ is a slice of $\overline{Y}$, then $\overline{S_0} \approx_{\varepsilon_3+\varepsilon_1+b(\varepsilon_1+\varepsilon_2)} U_d$. Since also $\widetilde{H}_\infty(X \mid R_0, R_0', R_1, R_1', \overline{R_0'}, \overline{R_1'}) \geq n_1 - 6d \geq k_1$, by the property of AC^0-LExt_1, conditioned on the fixings of $\overline{S_0}$, $\overline{R_0} \approx_{\varepsilon_3+2\varepsilon_1+b(\varepsilon_1+\varepsilon_2)}$ U_d is a linear function of X, thus independent of (Y, Y'). Since $\widetilde{H}_\infty(\overline{Y} \mid \overline{Y'}, S_0, S_0', S_1, S_1, \overline{S_0}, \overline{S_0'}) \geq n_2 - 6d \geq k_2$ and $\overline{R_0} \approx_{\varepsilon_3+2\varepsilon_1+b(\varepsilon_1+\varepsilon_2)} U_d$, by the property of AC^0-LExt_2, conditioned on the fixings of $\overline{R_0}$, $\overline{S_1} \approx_{\varepsilon_3+(2+b)\varepsilon_1+(b+1)\varepsilon_2} U_d$ is a linear function of $\overline{Y}$, thus independent of X. Since $\widetilde{H}_\infty(X \mid R_0, R_0', R_1, R_1', \overline{R_0}, \overline{R_0'}, \overline{R_1'}) \geq n_1 - 7d \geq k_1$ and $\overline{S_1} \approx_{\varepsilon_3+(2+b)\varepsilon_1+(b+1)\varepsilon_2} U_d$, by the property of AC^0-LExt_1, conditioned on the fixings of $\overline{S_1}$, $\overline{R_1} \approx_{\varepsilon_3+(3+b)\varepsilon_1+(b+1)\varepsilon_2} U_d$ is a linear function of X, thus independent of (Y, Y').

4. From the above analysis, for all $b' \in \{0,1\}$, $\overline{R_{1-b'}} \approx_{\varepsilon_3+3\varepsilon_1+\varepsilon_2} U_d \mid (Y, Y')$ and $\overline{R_{1-b}} \approx_{\varepsilon_3+3\varepsilon_1+\varepsilon_2}$ $U_d \mid (\overline{R_{1-b}'}, Y, Y')$. Therefore, conditioned on the fixing of $(\overline{R_{1-b}}, \overline{R_{1-b}'})$, $\hat{Y}$ $(\hat{Y}')$ is a linear function of Y (Y') and is therefore independent of X (X'). By Lemma 15, $\hat{Y} \approx_{2\varepsilon_3+3\varepsilon_1+\varepsilon_2}$ $U_{n_2} \mid (\hat{Y}', \overline{R_{1-b}}, \overline{R_{1-b}'})$.

5. Now it is easy to see that $Y_0 \approx_{2\varepsilon_3+3\varepsilon_1+\varepsilon_2} U_{n_2} \mid (Y_0, \overline{R_{1-b}}, \overline{R_{1-b}'})$, and Y_0 (Y_0') is independent of X (X'). We also have $\widetilde{H}_\infty(X \mid R_0, R_0', R_1, R_1', \overline{R_0}, \overline{R_1}, \overline{R_0'}, \overline{R_1'}) \geq n_1 - 8d \geq k_3$. By Lemma 15, it holds that $\hat{X} \approx_{3\varepsilon_3+3\varepsilon_1+\varepsilon_2} U_{n_1} \mid (\hat{X}', Y, Y')$.

This completes the proof of Theorem 60. □

AC^0-computable standard correlation breaker. The following algorithm is a modification of the correlation breaker in [Li17] so that it is computable by AC^0 circuits.

Definition 28 (AC^0-CB)**.** *A function* $\mathsf{AC}^0\text{-CB} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^a \to \{0,1\}^m$ *is a correlation breaker for entropy* k *with error* ε *that can be computed by* AC^0 *circuit of depth* c *(or a* (k, ε, c)*-affine correlation breaker for short) if for every* $X, X' \in \{0,1\}^n$*,* $Y, Y' \in \{0,1\}^d$*,* $\alpha, \alpha' \in \{0,1\}^a$ *s.t.*

- X *is an* (n,k) *source and* Y *is uniform*
- (X,X') *is independent of* (Y,Y')
- $\alpha \neq \alpha'$

AC^0-CB *can be computed by* AC^0 *circuits of depth* $O(c)$ *and*

$$\mathsf{AC}^0\text{-}\mathsf{CB}(X,Y,\alpha) \approx_\varepsilon U_m \mid \mathsf{AC}^0\text{-}\mathsf{AffCB}(X',Y',\alpha').$$

We say AC^0-CB *is strong if*

$$\mathsf{AC}^0\text{-}\mathsf{CB}(X,Y,\alpha) \approx_\varepsilon U_m \mid (\mathsf{AC}^0\text{-}\mathsf{AffCB}(X',Y',\alpha'),Y',Y).$$

Algorithm 10 AC^0-$\mathsf{CB}(x,y,id)$

Input: Bit strings x,y,id of length n,d,a respectively.
Output: Bit string $\hat{v}$ of length m.
Subroutines and Parameters:
Fix a constant c. Let $\ell = \log(a)$, $s = d/(1000(\ell+1))$, $r = s/a$, $m = \Omega(d)$.
Let $\mathsf{AC}^0\text{-}\mathsf{LExt}_0 : \{0,1\}^n \times \{0,1\}^{s_0} \to \{0,1\}^{d_0}$ be the AC^0-computable strong seeded extractor from 59 set to extract from a (n,d) source where $d = \Omega(n)$, seed $s_0 = O(\log^{c+1} n)$, output $d_0 = \Omega(d) \leq 0.3d$, $d_0 \geq 200\ell s$ and error $\varepsilon_n = 2^{-\log^c n}$.
Let $\mathsf{IP} : \{0,1\}^{d_0} \times \{0,1\}^{d_0} \to \{0,1\}^{d_0/6}$ be the two source extractor from Theorem 38 with error $\varepsilon_{\mathsf{IP}} = 2^{-\Omega(d)}$.
Let $\mathsf{AC}^0\text{-}\mathsf{laExt}_{2\ell+1} : \{0,1\}^d \times \{0,1\}^{d_0/6} \to \left(\{0,1\}^{3s}\right)^{2\ell+1}$ be the look-ahead extractor from Lemma 14 with the following extractors for Quentin and Wendy:

- $\mathsf{AC}^0\text{-}\mathsf{LExt}_q : \{0,1\}^{d_0/6} \times \{0,1\}^{3s} \to \{0,1\}^{3s}$ be the AC^0-computable strong seeded extractor from Theorem 59 set to extract for $(d_0/6, d_0/12)$ sources with error $\varepsilon_d = 2^{-\log^c d_0/6}$.
- $\mathsf{AC}^0\text{-}\mathsf{LExt}_w : \{0,1\}^d \times \{0,1\}^{3s} \to \{0,1\}^{3s}$ be the AC^0-computable strong seeded extractor from Theorem 59 set to extract from $(d, d/4)$ sources with error $\leq \varepsilon_d = 2^{-\log^c d_0/6}$.

Let $\mathsf{AC}^0\text{-}\mathsf{laExt}_{\ell+1} : \{0,1\}^n \times \{0,1\}^{d_0/6} \to \left(\{0,1\}^{3s}\right)^{\ell+1}$ be the look-ahead extractor from Lemma 14.

- $\mathsf{AC}^0\text{-}\mathsf{LExt}_q : \{0,1\}^{d_0/6} \times \{0,1\}^{3s} \to \{0,1\}^{3s}$ be the AC^0-computable strong seeded extractor from Theorem 59 set to extract from $(d_0/6, d_0/12)$ sources with error $\varepsilon_d = 2^{-\log^c d_0/6}$.
- $\mathsf{AC}^0\text{-}\mathsf{LExt}'_w : \{0,1\}^n \times \{0,1\}^{3s} \to \{0,1\}^{3s}$ be the AC^0-computable strong seeded extractor from Theorem 59 set to extract from $(n, d/4)$ sources with error $\varepsilon_n = 2^{-\log^c n}$.

Let $\mathsf{AC}^0\text{-}\mathsf{flip\text{-}flop} : \{0,1\}^{3s} \times \{0,1\}^{3s} \times \{0,1\}^a \to \{0,1\}^r$ be the AC^0-computable flip-flop from Theorem 60 with error $6 \cdot \varepsilon_n$.
Let $\mathsf{AC}^0\text{-}\mathsf{LExt} : \{0,1\}^{3s} \times \{0,1\}^r \to \{0,1\}^r$ be the AC^0-computable strong seeded extractor from Theorem 59 set to extract from uniform sources with error $\varepsilon_s = 2^{-\log^c(3s)}$.
Let $\mathsf{AC}^0\text{-}\mathsf{LExt}' : \{0,1\}^{3s} \times \{0,1\}^r \to \{0,1\}^r$ be the AC^0-computable strong seeded extractor from Theorem 59 set to extract form uniform sources with error ε_s.
Let $\mathsf{AC}^0\text{-}\mathsf{LExt}'' : \{0,1\}^d \times \{0,1\}^r \to \{0,1\}^s$ be the AC^0-computable strong seeded extractor from Theorem 59 set to extract from a $(d, d/4)$ source with error ε_d.
Let $\mathsf{AC}^0\text{-}\mathsf{LExt}''' : \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^m$ be the AC^0-computable strong seeded extractor from Theorem 59 set to extract from a $(n, d/4)$ source with error ε_n.

Let AC^0-NIPM_2 construction be 2-alternating extraction $\{0,1\}^r \times \{0,1\}^r \times \{0,1\}^{3s} \to \{0,1\}^r$ from Definition 18 with the following extractors for Quentin and Wendy:

- AC^0-$\mathsf{LExt}'_q : \{0,1\}^r \times \{0,1\}^r \to \{0,1\}^{r/2}$ be the AC^0-computable strong seeded extractor from Theorem 59 set to extract from (r,r) sources with error $\varepsilon_r = 2^{-\log^c r}$.
- AC^0-$\mathsf{LExt}''_w : \{0,1\}^{3s} \times \{0,1\}^{r/2} \to \{0,1\}^r$ be the AC^0-computable strong seeded extractor from Theorem 59 set to extract from $(3s,s)$ sources with error ε_s.

---

1. Let $y_0 \circ y_1 = \mathsf{Slice}(y, s_0 + 0.3d)$ where y_0 has length s_0 and $x_1 = \mathsf{AC}^0\text{-}\mathsf{LExt}_0(x, y_0)$.
2. Compute $z = \mathsf{IP}(x_1, y_1)$.
3. Let $r_0, r_1, \cdots, r_{2\ell} = \mathsf{AC}^0\text{-}\mathsf{laExt}_{2\ell+1}(y, z)$.
4. Let $s_0, s_1, \cdots, s_\ell = \mathsf{AC}^0\text{-}\mathsf{laExt}_{\ell+1}(x, z)$.
5. Let V^0 be an $a \times r$ matrix whose i'th row is $V^0_i = \mathsf{AC}^0\text{-}\mathsf{flip\text{-}flop}(s_0, r_0, \alpha_i)$ and has r bits.
6. For $j = 1, \cdots, \ell$ do the following. Merge the matrix v^{j-1} two rows by two rows: Note that v^{j-1} has $a/2^{j-1}$ rows, for $i = 1, \cdots, a/2^j$, compute $\overline{v^{j-1}_i} = \mathsf{AC}^0\text{-}\mathsf{NIPM}(v^{j-1}_{2i-1}, v^{j-1}_{2i}, r_{2j-1})$ which outputs r bits, and $\tilde{v}^{j-1}_i = \mathsf{AC}^0\text{-}\mathsf{LExt}(r_{2j}, \overline{v^{j-1}_i})$ which has r bits. Finally compute $v^j_i = \mathsf{AC}^0\text{-}\mathsf{LExt}'(s_j, \tilde{v}^{j-1}_i)$ which has r bits.
7. Compute $\hat{v} = \mathsf{AC}^0\text{-}\mathsf{LExt}'''(x, \mathsf{AC}^0\text{-}\mathsf{LExt}''(y, v^\ell))$.

Theorem 61 (AC^0-CB)**.** *For every constant c, there exists an explicit strong correlation breaker $\{0,1\}^n \times \{0,1\}^d \times \{0,1\}^a \to \{0,1\}^{\Omega(d)}$ for entropy d with error $\varepsilon = O(a \cdot 2^{-\log^c n})$, where $d = \Omega(n)$ and $a = O(\frac{d}{\log^c d})$. Moreover, the correlation breaker is computable by AC^0 circuits of depth $O(c)$.*

Proof. We show that Algorithm 10 gives such a correlation breaker. We shall analyze the algorithm step by step.

Step 1. Fix (Y_0, Y'_0), conditioned on this fixing, X_1 is a linear function of X and is independent of (Y_1, Y'_1) and (Y, Y').

Step 2. Since $X_1 \approx_{\varepsilon_n} U_{0.3d}$ and $Y_1 = U_{0.3d}$, by the definition of IP, $Z \approx_{\varepsilon_{\mathsf{IP}}} U_{d_0/6}$.

Step 3.

- Further fix (Y_1, Y'_1), conditioned on this fixing, it holds (Z, Z') is a deterministic function of (X_1, X'_1), and thus (Z, Z') is independent of (Y, Y').
- $Z \approx_{\varepsilon_{\mathsf{IP}}} U_{d_0/6}$ and $\widetilde{H}_\infty(Y \mid Y_0, Y'_0, Y_1, Y'_1) \geq d - 2s_0 - 2 \cdot d_0 \geq 0.3d$.
- By Lemma 14, since $\widetilde{H}_\infty(Y \mid Y_1, Y'_1) \geq 0.3d \geq d/4 + 2(2\ell+1)(3s) + 2\log(1/\varepsilon_d)$ and $d_0/6 \geq d_0/12 + 2(2\ell+1)(3s) + 2\log(1/\varepsilon_d)$, we have that for any $0 \leq j \leq 2\ell - 1$, it holds that

$$R_{j+1} \approx_{O(\ell\varepsilon_d)} U_{3s} \mid (Z, Z', R_0, R'_0, \cdots, R_j, R'_j).$$

  By a hybrid argument and the triangle inequality, we have that

$$(Z, Z', R_0, R'_0, \cdots, R_{2\ell}, R'_{2\ell}) \approx_{O(\ell^2\varepsilon_d)} (Z, Z', U_{3s}, R'_0, \cdots, U_{3s}, R'_{2\ell}). \tag{4}$$

  where each U_{3s} is independent of all the previous random variables (but may depend on later random variables).

- Conditioned on the fixing of (Z, Z'), we have $\{(R_i, R'_i)\}_{i\in[0,2\ell]}$ is a deterministic function of (Y_1, Y'_1), thus independent of (X, X').

Step 4.

- Fix (X_1, X'_1), conditioned on this fixing, it holds (Z, Z') is a deterministic function of (Y_1, Y'_1), and thus (Z, Z') is independent of (X, X').
- $Z \approx_{\varepsilon_{\mathsf{IP}}} U_{d_0/6}$ and $\widetilde{H}_\infty(X \mid X_1, X'_1) \geq d - 2 \cdot d_0 \geq 0.4d$.
- By Lemma 14, since $\widetilde{H}_\infty(X \mid X_1, X'_1) \geq 0.4d \geq d/4 + 2(\ell+1)(3s) + 2\log(1/\varepsilon_n)$ and $d_0/6 \geq d_0/12 + 2(\ell+1)(3s) + 2\log(1/\varepsilon_d)$, we have that for any $0 \leq j \leq \ell - 1$, it holds that

$$S_{j+1} \approx_{O(\ell(\varepsilon_n+\varepsilon_d)/2)} U_{3s} \mid (Z, Z', \{S_0, S'_0, \cdots, S_j, S'_j\}).$$

  By a hybrid argument and the triangle inequality, we have that

$$(Z, Z', S_0, S'_0, \cdots, S_\ell, S'_\ell) \approx_{O(\ell^2(\varepsilon_n+\varepsilon_d)/2)} (Z, Z', U_{3s}, S'_0, \cdots, U_{3s}, S'_\ell). \tag{5}$$

  where each U_{3s} is independent of all the previous random variables (but may depend on later random variables).

- Conditioned on the fixing of (Z, Z'), we have $\{(S_i, S'_i)\}_{i\in[0,\ell]}$ is a deterministic function of (X_1, X'_1), thus independent of (Y, Y').

Therefore, we conclude that conditioned on the fixing of $(X_1, X'_1, Y_1, Y'_1, Z, Z')$, we have $\{(R_i, R'_i)\}_{i\in[0,2\ell]}$ is a deterministic function of (Y, Y'), and $\{(S_i, S'_i)\}_{i\in[0,\ell]}$ is a deterministic function of (X, X'), thus they are independent. Moreover each R_i and S_i is close to uniform given the previous random variables. From now on, we will assume that each R_i and S_i are uniform (*) and add back an error of $O(\ell^2(\varepsilon_n + \varepsilon_d))$ in the end. Since in the algorithm and the analysis below, each R_i and S_i are used at most twice either as source of seed, this is sufficient.

Step 5. By Theorem 60, for all $i \in [a]$, $V^0_i \approx_{O(\varepsilon_n)} U_s$. Moreover, since $\alpha \neq \alpha'$, there exists an $i \in [a]$ such that $V^0_i \approx_{O(\varepsilon_n)} U_s \mid (V'^0_i, R_0, R'_0)$. Now that conditioned on the fixing of (R_0, R'_0), (V^0, V'^0) is a deterministic function of (S_0, S'_0), and thus independent of $\{(R_i, R'_i)\}_{i\in[2\ell]}$.

Step 6. First note that the followings:

1. conditioned on the fixing of (R_0, R'_0), (V^0, V'^0) is a linear function of (S_0, S'_0).
2. Each row of V^0 is close to uniform and there exists a row in V^0 that is close to uniform even conditioned on the corresponding row in V'^0.

Along the analysis below, we prove by induction that for any $j \in [0, \ell]$,

(a) each row of V^j is close to uniform, and there exists a row in V^j that is close to uniform even conditioned on the corresponding row in V'_j.

For any $j \in [\ell]$, it holds that

(b) conditioned on the fixing of $(R_0, R'_0, \cdots, R_{2j-2}, R'_{2j-2})$, (V^{j-1}, V'^{j-1}) is a linear functions of $(S_0, S'_0, \cdots, S_{j-1}, S'_{j-1})$.

(c) each row of $\overline{V^{j-1}}$ $(\tilde{V}^{j-1})$ is close to uniform, and there exists a row in $\overline{V^{j-1}}$ $(\tilde{V}^{j-1})$ that is close to uniform even conditioned on the corresponding row in $\overline{V'^{j-1}}$ $(\tilde{V}'^{j-1})$.

For each iteration $j \in [\ell]$, Step 6 generates 3 new somewhere random matrices: $\overline{V^{j-1}}$, $\tilde{V}^{j-1}$, and V^j of size $(a/2^j) \times r$,$(a/2^j) \times r$, and $(a/2^j) \times r$ respectively. Each one of them has some properties:
Matrix $\overline{V^{j-1}}$. Conditioned on the fixings of $(R_0, R'_0, \cdots, R_{2j-2}, R'_{2j-2})$, by our assumption (*), $R_{2j-1} = U_{3s}$. Now, condition on $(R_0, R'_0, \cdots, R_{2j-1}, R'_{2j-1})$, by Lemma 14, each row of $\overline{V^{j-1}}$ is $O(2^{j-1}(\varepsilon_s + \varepsilon_n))$ close to uniform. Since there exists one row in V^{j-1} that is close to uniform even given the corresponding row in V'^{j-1}, by Lemma 15, there is one row in $\overline{V^{j-1}}$ that is close to uniform even conditioned on the same row in $\overline{V'^{j-1}}$. Moreover, conditioned on the fixing of $(R_0, R'_0, \cdots, R_{2j-1}, R'_{2j-1})$, $(\overline{V^{j-1}}, \overline{V'^{j-1}})$ is a linear function of (V^{j-1}, V'^{j-1}), which, by induction hypothesis, is a linear function $(S_0, S'_0, \cdots, S_{j-1}, S'_{j-1})$, and thus independent of R_{2j}.
Matrix $\tilde{V}^{j-1}$. First note that the i-th row of the matrix $\tilde{V}^{j-1}$ is obtained by using the i-th row of matrix $\overline{V^{j-1}}$ to extract from S_j, for each $i \in [a/2^j]$. In addition, conditioned on $\overline{V^{j-1}}$, $\tilde{V}^{j-1}$ is a deterministic function of R_{2j}. Since $R_{2j} = U_{3s} \mid (R_0, R'_0, \cdots, R_{2j-1}, R'_{2j-1})$ and $\widetilde{H}_\infty(R_{2j} \mid \overline{V^{j-1}}, \tilde{V}^{j-1}_{[u]}) \geq 3s - ur \geq 3s - ar/2^{j-1} \geq 3s - ar \geq s + \log(1/\varepsilon_s)$ where $u \in [a/2^j - 1]$, each row of $\tilde{V}^{j-1}$ is uniform by the definition of AC^0-LExt. Since there is one row in $\overline{V^{j-1}}$ that is $O(2^{j-1}(\varepsilon_s + \varepsilon_n))$ close to uniform conditioned on the corresponding row in $\overline{V'^{j-1}}$, by Lemma 15, there is also one row in $\tilde{V}^{j-1}$ that is close to uniform even conditioned on the corresponding row in $\tilde{V}'^{j-1}$.
Matrix V^j. First note that the i-th row of the matrix V^j is obtained by using the i-th row of matrix $\tilde{V}^{j-1}$ to extract from S_j, for each $i \in [a/2^j]$. In addition, conditioned on $\tilde{V}^{j-1}$, V^j is a deterministic function of S_j. Since $S_j = U_{3s} \mid (S_0, S'_0, \cdots, S_{j-1}, S'_{j-1})$ and $\widetilde{H}_\infty(S_j \mid \tilde{V}^{j-1}, V^j_{[u]}) \geq 3s - ur \geq 3s - ar/2^{j-1} \geq 3s - ar \geq s + \log(1/\varepsilon_s)$ where $u \in [a/2^j - 1]$, each row of V^j is $O(2^j(\varepsilon_s + \varepsilon_n))$ close to uniform by the definition of AC^0-LExt'. Since there is one row in $\tilde{V}^{j-1}$ that is close to uniform conditioned on the corresponding row in $\tilde{V}'^{j-1}$, by Lemma 15, there is also one row in V^j that is close to uniform even conditioned on the corresponding row in V'^j.
Setting $j = \ell$, we get $\mathrm{ins}V^\ell \approx_{O(2^\ell(\varepsilon_s+\varepsilon_n))} U_r \mid V'^\ell$.
Step 7. Note that $H(Y \mid \{R_i, R'_i\}_{i\in[0,2\ell]}) \geq d/4 + 2\log(1/\varepsilon_d)$ and $H(X \mid \{S_i, S'_i\}_{i\in[0,\ell]}) \geq d + 2\log(1/\varepsilon_n)$, since $V^\ell \approx_{O(a(\varepsilon_s+\varepsilon_n))} U_r \mid V'^\ell$, by 2 iterative use of Lemma 15, it follows that $\hat{V} \approx_{O(\varepsilon_n+\ell^2(\varepsilon_n+\varepsilon_d)+a(\varepsilon_s+\varepsilon_n)+\varepsilon_d+\varepsilon_n)} U_m \mid \hat{V}' \iff \hat{V} \approx_{O(a\varepsilon_n)} U_m \mid \hat{V}'$. Since $\hat{V}$ is a deterministic function of X conditioned on $(Y, Y', \{S_i, S'_i\}_{i\in[0,\ell]})$ and AC^0-LExt''' is strong, it also holds that $\hat{V} \approx_{O(a\varepsilon_n)} U_m \mid (\hat{V}', Y, Y')$. This completes the proof of Theorem 61. □

AC^0-computable t-affine correlation breaker. The following definition is a modification of t-affine correlation breaker [CL22] into the AC^0-computable setting.

Definition 29 (AC^0-AffCB). *A function $\mathsf{AC}^0\text{-}\mathsf{AffCB} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^a \to \{0,1\}^m$ is a t-affine correlation breaker for entropy k with error ε that can be computed by AC^0 circuit of depth c (or a (t, k, ε, c)-affine correlation breaker for short) if for every $X, A, B \in \{0,1\}^n$, $Y, Y^{[t]} \in \{0,1\}^d$, Z and string $\alpha, \alpha^{[t]} \in \{0,1\}^a$ s.t.*

- $X = A + B$
- $\widetilde{H}_\infty(A \mid Z) \geq k$
- $(Y, Z) = (U_d, Z)$
- *A is independent of $(B, Y, Y^{[t]})$ given Z*
- *$\alpha, \alpha^1, \cdots, \alpha^t$ be a-bit strings s.t. $\alpha \neq \alpha^i$ for every $i \in [t]$*

AC^0-AffCB *can be computed by* AC^0 *circuits of depth* $O(c)$ *and*

$$\mathsf{AC}^0\text{-}\mathsf{AffCB}(X, Y, \alpha) \approx_\varepsilon U_m \mid \left\{\mathsf{AC}^0\text{-}\mathsf{AffCB}(X^i, Y^i, \alpha^i)\right\}_{i\in[t]}.$$

We say AC^0-AffCB *is strong if*

$$\mathsf{AC}^0\text{-}\mathsf{AffCB}(X, Y, \alpha) \approx_\varepsilon U_m \mid \left(\left\{\mathsf{AC}^0\text{-}\mathsf{AffCB}(X^i, Y^i, \alpha^i)\right\}_{i\in[t]}, Y^{[t]}, Y\right).$$

Algorithm 11 below is a construction of strong (t, k, ε, c)-affine correlation breaker.

Algorithm 11 AC^0-$\mathsf{AffCB}(x, y, id)$

Input: Bit strings $x = w + z, y, id$ of length $n, d = \Omega(n), a$ respectively.
Output: Bit string $q_{\lceil \log t\rceil}$ of length r.
Subroutines and Parameters:
Fix a constant c. Let $d_0' = O(\log^{c+1} n)$, $d_0 \le \min\{k, d\}/(10t + 10)$, $d_x \le d_0/(2\log t)$, $r = k/(10 + 10t)$, $d_y = \frac{r}{4t\log t}$.
Let $\mathsf{AC}^0\text{-}\mathsf{LExt} : \{0,1\}^n \times \{0,1\}^{d_0'} \to \{0,1\}^{d_0}$ be the AC^0-computable strong seeded extractor from Theorem 59 with error $\varepsilon_n = 2^{-\log^c n}$.
Let $\mathsf{AC}^0\text{-}\mathsf{CB} : \{0,1\}^d \times \{0,1\}^{d_0} \times \{0,1\}^a \to \{0,1\}^{d_x}$ be the AC^0-computable correlation breaker from Theorem 61 with error $\varepsilon' = O(a \cdot 2^{-\log^c d})$.
Let $\mathsf{AC}^0\text{-}\mathsf{LExt}' : \{0,1\}^n \times \{0,1\}^{d_x} \to \{0,1\}^r$ be the AC^0-computable strong seeded extractor from Theorem 59 with error $\varepsilon_n = 2^{-\log^c n}$.
Let $\mathsf{AC}^0\text{-}\mathsf{LExt}_w : \{0,1\}^d \times \{0,1\}^{d_y} \to \{0,1\}^{d_x}$ be the AC^0-computable strong seeded extractor from Theorem 59 with error $\varepsilon_d = 2^{-\log^c d}$.
Let $\mathsf{AC}^0\text{-}\mathsf{LExt}_q : \{0,1\}^r \times \{0,1\}^{d_x} \to \{0,1\}^{d_y}$ be the AC^0-computable strong seeded extractor from Theorem 59 with error $\varepsilon_r = 2^{-\log^c r}$.

Let $y_0 = \mathsf{Slice}(y, d_0')$
Let $x_0 = \mathsf{AC}^0\text{-}\mathsf{LExt}(x, y_0)$
Let $y_1 = \mathsf{AC}^0\text{-}\mathsf{CB}(y, x_0, \alpha)$
Let $q_0 = \mathsf{AC}^0\text{-}\mathsf{LExt}'(x, y_1)$
For every i, $1 \le i \le \lceil \log t\rceil$ do the following

1. Let $s_{i-1} = \mathsf{Slice}(q_{i-1}, d_y)$
2. Let $r_{i-1} = \mathsf{AC}^0\text{-}\mathsf{LExt}_w(y, s_{i-1})$
3. Let $\overline{s_i} = \mathsf{AC}^0\text{-}\mathsf{LExt}_q(q_{i-1}, r_{i-1})$
4. Let $\overline{r_i} = \mathsf{AC}^0\text{-}\mathsf{LExt}_w(y, \overline{s_i})$
5. Let $q_i = \mathsf{AC}^0\text{-}\mathsf{LExt}'(x, \overline{r_i})$

Theorem 62 (AC^0-AffCB)**.** *For every* $c \in \mathbb{N}$*, constant* $0 < \delta < 1$ *and* $n \in \mathbb{N}$ *and every* k, d, t, a*, there exists a constant* C *such that if*

- $k \ge \delta n$
- $d = \Omega(n)$ *and* $d \le n$
- $t = O(1)$

- $a \leq C\frac{n}{\log^c(n)}$

then there exists a strong AC^0-$\mathsf{AffCB} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^a \to \{0,1\}^m$ *which is computable by depth* $O(c)$ AC^0 *circuits*

- $m = \Omega(k)$
- $\varepsilon = O(2^{-\log^{c-1} k})$

Proof. We will prove that Algorithm 11 gives such a function.
First we prove that AC^0-AffCB satisfy Definition 29.

1. For all $i \in [t]$, let $X^i_{0,A} := \mathsf{AC}^0\text{-}\mathsf{LExt}(A, Y^i_0)$, $X^i_{0,B} := \mathsf{AC}^0\text{-}\mathsf{LExt}(B, Y^i_0)$, $Q^i_{0,A} := \mathsf{AC}^0\text{-}\mathsf{LExt}'(A, Y^i_1)$, $Q^i_{0,B} := \mathsf{AC}^0\text{-}\mathsf{LExt}'(B, Y^i_1)$. Let Z be Z_{29} from Definition 29.

2. By definition of AC^0-LExt,
$$X_{0,A} \approx_{\varepsilon_n} U_{d_0} \mid (Z, Y_0, Y_0^{[t]}, X_{0,B}, X_{0,B}^{[t]}).$$

3. Since $\widetilde{H}_\infty(Y \mid Z, Y_0, Y_0^{[t]}, X_{0,B}, X_{0,B}^{[t]}) \geq d - (t+1)d'_0 \geq 9d/10$, $R_{1,A}, R_{1,A}^{[t]}$ are independent of $Y, Y^{[t]}$ given $Z, Y_0, Y_0^{[t]}, X_{0,B}, X_{0,B}^{[t]}$, and AC^0-CB is a strong correlation breaker, it holds $\forall i \in [t]$ that
$$Y_1 \approx_{\varepsilon_n+\varepsilon'} U_{d_x} \mid (Y_1^i, Z, Y_0, Y_0^{[t]}, X_{0,B}, X_{0,B}^{[t]}, X_0, X_0^i).$$

4. Since conditioned on the fixing of $X_0, X_{0,B}^{[t]}$, Y_1 is a deterministic function of Y and is independent of $X_0^{[t]}$,
$$Y_1 \approx_{\varepsilon_n+\varepsilon'} U_{d_x} \mid (Y_1^i, Z, Y_0, Y_0^{[t]}, X_{0,B}, X_{0,B}^{[t]}, X_0, X_0^{[t]}).$$

5. By Lemma 15, it holds $\forall i \in [t]$ that
$$Q_{0,A} \approx_{2\varepsilon_n+\varepsilon'} U_r \mid (Q^i_{0,A}, Z, Y_0, Y_0^{[t]}, X_{0,B}, X_{0,B}^{[t]}, X_0, X_0^{[t]}, Y_1, Y_1^{[t]}).$$
Since $(Q_{0,B}, Q_{0,B}^{[t]})$ is independent of $Q_{0,A}$, let
$$Z_0 := (Z, Y_0, Y_0^{[t]}, X_{0,B}, X_{0,B}^{[t]}, X_0, X_0^{[t]}, Y_1, Y_1^{[t]}, Q_{0,B}, Q_{0,B}^{[t]}),$$
it also holds that
$$Q_{0,A} \approx_{2\varepsilon_n+\varepsilon'} U_r \mid (Q^i_{0,A}, Z_0),$$
which is equivalent to
$$Q_0 \approx_{2\varepsilon_n+\varepsilon'} U_r \mid (Q^i_0, Z_0).$$

Claim 63. *Each one of* $\overline{S_i}$, $\overline{R_i}$, Q_i, R_i *is close to uniform and independent of every* $\min\{2^i, t\}$ *tampered r.v.'s.*

Proof. For each $i \in [\lceil \log t \rceil]$, let
$$Z_{i,1,B} := (Z_{i-1}, S_{i-1,B}, S_{i-1,B}^{[t]});\ Z_{i,2} := (Z_{i,1,B}, S_{i-1}, S_{i-1}^{[t]});\ Z_{i,3} := (Z_{i,2}, R_{i-1}, R_{i-1}^{[t]})$$
$$Z_{i,3,B} := (Z_{i,3}, \overline{S_{i,B}}, \overline{S_{i,B}}^{[t]});\ Z_{i,4} := (Z_{i,3}, \overline{S_i}, \overline{S_i}^{[t]});\ Z_i := (Z_{i,4}, \overline{R_i}, \overline{R_i}^{[t]}),$$

let T_i be any subset of $[t]$ of size 2^i if $2^i \le t$, otherwise, let it be $[t]$. Now we define an ordering for the claims $\mathcal{C}$ according to which we prove by induction. The first claim is Sub-step 5 with $i = 0$. Then the claims follow the order of round i, Sub-step 1; round i, Sub-step 2, ..., round i, Sub-step 5, round $i+1$, Sub-step 1; round $i+1$, Sub-step 2, ..., round $\lceil \log t \rceil$, Sub-step 5. First note that by the above arguments, the claim in Sub-step 5 below holds for $i = 0$. It is clear that claims in $\mathcal{C}$ of order $\le k$ implies that of order $k+1$.

Sub-step 1: $S_{i-1} \approx_{2\varepsilon_n+\varepsilon'+(i-1)(2\varepsilon_d+\varepsilon_n+\varepsilon_r)} U_{d_y} \mid (S_{i-1}^{T_{i-1}}, Z_{i-1})$; $S_{i-1,A} \approx_{2\varepsilon_n+\varepsilon'} U_{d_y} \mid (S_{i-1,A}^{T_{i-1}}, Z_{i,1,B})$ as long as the statement in Sub-step 5 holds for $i-1$.

Sub-step 2: It holds by Lemma 15 that $R_{i-1} \approx_{2\varepsilon_n+\varepsilon'+\varepsilon_d+(i-1)(2\varepsilon_d+\varepsilon_n+\varepsilon_r)} U_{d_x} \mid (R_{i-1}^{T_{i-1}}, Z_{i,2})$ as long as the statement in Sub-step 1 holds and $\widetilde{H}_\infty(Y \mid Z_{i,2}) \ge 9d/10 - 2(i-1)(t+1)d_x \ge d/2$.

Sub-step 3: It holds by Lemma 15 that $\overline{S_i} \approx_{2\varepsilon_n+\varepsilon'+\varepsilon_d+\varepsilon_r+(i-1)(2\varepsilon_d+\varepsilon_n+\varepsilon_r)} U_{d_x} \mid (\overline{S_i}^{T_i}, Z_{i,3})$; $\overline{S_{i,A}} \approx_{2\varepsilon_n+\varepsilon'+\varepsilon_d+\varepsilon_r+(i-1)(2\varepsilon_d+\varepsilon_n+\varepsilon_r)} U_{d_x} \mid (\overline{S_{i,A}}^{T_i}, Z_{i,3,B})$ as long as the statement in Sub-step 2 holds and $\widetilde{H}_\infty(Q_{i-1} \mid Z_{i,3}) \ge r - (2i-1)(t+1)d_y \ge r/2$.

Sub-step 4: It holds by Lemma 15 that $\overline{R_i} \approx_{2\varepsilon_n+\varepsilon'+2\varepsilon_d+\varepsilon_r+(i-1)(2\varepsilon_d+\varepsilon_n+\varepsilon_r)} U_{d_y} \mid (\overline{R_i}^{T_i}, Z_{i,4})$ as long as the statement in Sub-step 3 holds and $\widetilde{H}_\infty(Y \mid Z_{i,4}) \ge 9d/10 - (2i-1)(t+1)d_x \ge d/2$.

Sub-step 5: It holds by Lemma 15 that $Q_i \approx_{2\varepsilon_n+\varepsilon'+i(2\varepsilon_d+\varepsilon_n+\varepsilon_r)} U_r \mid (Q_i^{T_i}, Z_i)$ as long as the statement in Sub-step 4 holds and $\widetilde{H}_\infty(X \mid Z_i) \ge k - (d_0+r)(t+1) - 2(i-1)(t+1)d_x \ge k/2$.

□

Now, note that conditioned on $Z_{\lceil \log t \rceil}$, which contains $(\overline{R_{\lceil \log t \rceil}}, \overline{R_{\lceil \log t \rceil}}^{[t]})$, $Q_{\lceil \log t \rceil} \approx_{O(\varepsilon'+(\log t)\varepsilon_n)} U_r \mid Q^{[t]}_{\lceil \log t \rceil}$. Moreover, $Q_{\lceil \log t \rceil}, Q^{[t]}_{\lceil \log t \rceil}$ are deterministic functions of X and are independent of $Y, Y^{[t]}$. Therefore, we have

$$Q_{\lceil \log t \rceil} \approx_{O((a+\log t)\cdot 2^{-\log^c n})} U_r \mid (Q^{[t]}_{\lceil \log t \rceil}, Y, Y^{[t]}).$$

This completes the proof of Theorem 62. □

6.2 AC^0-Computable Extractor for Read-Once Branching Program Sources

Algorithm 12 $\mathsf{AC}^0\text{-Ext}(x)$

Input: x — an n bit string.
Output: z — an m bit string with $m = \Omega(n)$.

Sub-Routines and Parameters:
Let $\mathsf{AC}^0\text{-BFExt} : \{0,1\}^n \to \{0,1\}^{n_1}$ be a linear seeded extractor from Theorem 58 set to extract from min-entropy $k_1 = \delta n$ with error $\varepsilon_1 = 2^{-\log^c(n/t)}$.
Let $\mathsf{AC}^0\text{-AffCB} : \{0,1\}^n \times \{0,1\}^{n_1} \times \{0,1\}^a \to \{0,1\}^m$, $a = \log(t)$, be the t-affine correlation breaker from Theorem 62 with error $\varepsilon_2 = O(2^{-\log^{c-1} n})$.

1. Divide x into $t = 2/\delta$ blocks such that $x = x_1 \circ \cdots \circ x_t$.
2. Let $y_1 \circ \cdots \circ y_t = \mathsf{AC}^0\text{-BFExt}(x_1) \circ \cdots \circ \mathsf{AC}^0\text{-BFExt}(x_t)$ such that each y_i is of length $n_1 < \delta^2/100n$ bits.
3. Let s be a $t \times m$ matrix whose i'th row s_i, is $\mathsf{AC}^0\text{-AffCB}(x, y_i, i)$.
4. Output $z = \bigoplus_{j=1}^t s_i$.

Theorem 64. *For any constant $0 < \delta \leq 1$, there exists a family of functions $\mathsf{AC}^0\text{-}\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ computable in AC^0, such that for any sources $X = A + B$ where A and B are independent and have disjoint spans, A has entropy δn and B is an almost bit-fixing source of entropy δn, $\mathsf{AC}^0\text{-}\mathsf{Ext}(X) \approx_\varepsilon U_m \mid B$ for $m = \Omega(n)$ and $\varepsilon = O(2^{-\log^{c-1} n})$.*

Proof.

Lemma 65. *There exists $g \in [t]$ such that conditioned on the fixing of $\{B_1, \cdots, B_{g-1}, A_g, B_{g+1}, \cdots, B_t\}$, the followings are true.*

- *X_g is an almost bit-fixing source of entropy rate δ and $Y_g \approx_{2^{-\log^c(n/t)}} U_{n_1}$.*
- *Y_g is a deterministic function of B.*
- *$\{Y_1, \cdots, Y_{g-1}, Y_{g+1}, \cdots, Y_t\}$ are deterministic functions of A.*

Proof. Since $X = A + B$, then $X_i = A_i + B_i$ for all $i \in [t]$. Since $H_\infty(B) \geq \delta n$, and each B_i is of block length n/t, there exists $g \in [t]$ such that $H_\infty(B_g) \geq \delta n/t = \delta^2 n/2$. Now by the extraction property of $\mathsf{AC}^0\text{-}\mathsf{BFExt}$, we have $Y_g \approx_{\varepsilon_1} U_{n_1}$. □

Lemma 66. *Conditioned on the additional fixing of $Y = \{Y_i\}_{i \in [t] \setminus \{g\}}$, $H_\infty(A) \geq \delta n/4$.*

Proof. Since X_g has entropy at most $\delta n/2$, $H_\infty(B_g) \geq \delta^2 n/2$. Now as A_g is independent of B_g, $H_\infty(A) \leq (1-\delta)\delta n/2$. Since $|Y| \leq t n_1 \leq \delta n/50$, we have $\widetilde{H}_\infty(A \mid A_g, Y) \geq \delta n - (1-\delta)\delta n/2 - \delta n/50 \geq \delta n/4$. □

Lemma 67. *With probability $1 - \varepsilon_2$ over the fixings of $(A_g, \{S_i, Y_i\}_{i \in [t]}, B)$, $Z \approx_{2\varepsilon_1 + \varepsilon_2} U_m$.*

Proof. Let $Z_{29} = \{Y_i, B_i\}_{i \in [t] \setminus \{g\}} \cup \{A_g\}$. By Lemma 65, with probability $1 - 2^{-\log^c(n/t)}$, Y is a somewhere random source. Moreover, since A and B are independent, we have $Y_g = U_{n_1} \mid Z$. By Lemma 66, $\widetilde{H}_\infty(A \mid Z_{29}) \geq 4\delta/n$. By Theorem 62, $S_g \approx_{\varepsilon_1 + \varepsilon_2} U_m \mid (\{S_i\}_{i \in [t] \setminus \{g\}}, Y^{[t]})$. Since Y_g is a deterministic function of B_g, and conditioned on Y_g and Z_{29}, S_g is a deterministic function of A, it holds that $S_g \approx_{\varepsilon_1 + \varepsilon_2 + \varepsilon_1} U_m \mid (\{S_i, Y_i\}_{i \in [t] \setminus \{g\}}, B)$, which implies $Z \approx_{2\varepsilon_1 + \varepsilon_2} U_m \mid B$. □

□

Theorem 68. *For any constant $\delta > 0$, let $\mathsf{AC}^0\text{-}\mathsf{Ext}$ be a function from Theorem 64 for $\delta_{64} = \delta/3$ with error $\varepsilon = 2^{-\Omega(\log^{c-1} n)}$, then*

$$\mathsf{ROBP}_{2\varepsilon}(\mathsf{AC}^0\text{-}\mathsf{Ext}) > 2^{(1-\delta)n}.$$

We prove the above theorem in two steps. First, we recall a lemma in [CL23] and show that there exists a sum of two sources $X = A + B$ with the following 3 properties, (1) A and B are supported on disjoint subsets of input bits; (2) A has min-entropy $(1-\delta)n - \log s$ and B has min-entropy at least δn; and (3) B is an oblivious bit-fixing source. Then we show that the output of our extractor is close to uniform conditioned on the output of ROBP.

Lemma 69 (A special case of Lemma 3.1 from [CL23])**.** *Let X be a uniform random variable over F_2^n. For every read-once BP $f : \mathsf{F}_2^n \to \{0,1\}$ of size s and every $d \in [n]$, there exists a random variable E, and random variables $A, B \in \mathsf{F}_2^n$ s.t.*

- *E has support size at most $2s$.*

- $X = A + B$.
- *For every* $e \in \mathsf{Supp}(E)$, *define* $A_e = A\,|_{E=e}$, $B_e = B\,|_{E=e}$, *Then we have*
  - A^e *and* B^e *are independent.*
  - B^e *is uniform over a subset of coordinates* V_e^B *of dimension* d.
  - *There exists a complemented subspace* V_e^A *of* V_e^B *such that* $A_e \in V_e^A$.
- *There exists a deterministic function* g *s.t.* $g(E, B) = f(X)$.

Then we prove the claim below, which implies the average-case lower bound of ROBP.

Claim 70. *For any constant* $\delta > 0$, *let* AC^0-Ext *be a function from Theorem 64 with* $\delta_{64} = \delta/3$ *outputting* 1 *bit with error* ε, *and* $f : \{0,1\}^n \to \{0,1\}$ *be any* ROBP *of size* $s = 2^{(1-\delta)n}$. *Let* X *be a uniform random variable over* F_2^n. *Then*

$$(\mathsf{AC}^0\text{-}\mathsf{Ext}(X), B, E, f(X)) \approx_\varepsilon (U, B, E, f(X)).$$

Proof. Note that AC^0-Ext is a strong $(\delta n/3, \varepsilon)$ extractor, then by Lemma 13, it is a $(\delta n/3 + \mathsf{poly}\log n, 2\varepsilon)$ average case extractor. Since $\widetilde{H}_\infty(A \mid E) = 2^{n-\delta n/3-\log(2s)} = 2\delta/3 - 1 \geq \delta n/3 + \mathsf{poly}\log n$, we have

$$(\mathsf{AC}^0\text{-}\mathsf{Ext}(X), B, E) \approx_{2\varepsilon} (U, B, E).$$

Since $f(X) = g(E, B)$ is a deterministic function of E and B, we can conclude that

$$(\mathsf{AC}^0\text{-}\mathsf{Ext}(X), B, E, f(X)) \approx_{2\varepsilon} (U, B, E, f(X)).$$

□

7 Open Problems

Our work leaves several natural open problems. The most obvious is to further improve the constructions of directional affine extractors and the average-case hardness for SROLBPs. It would also be quite interesting to show any hardness of explicit functions for WROLBPs, which appears to require new ideas. Finally, it is an interesting question to see if there exist functions in AC^0 that achieve optimal hardness for ROBPs, or strong hardness for SROLBPs.

Acknowledgement

We thank anonymous reviewers for their helpful comments and a reviewer for pointing us to [GI17].

References

[ABCR99] Alexander E. Andreev, Juri L. Baskakov, Andrea E. F. Clementi, and José D. P. Rolim. Small pseudo-random sets yield hard functions: New tight explict lower bounds for branching programs. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *Lecture Notes in Computer Science*, pages 179–189. Springer, 1999.

[BISW04] Boaz Barak, Russel Impagliazzo, Amir Shpilka, and Avi Wigderson. Definition and existence of dimension expanders. Discussion (no written record), 2004.

[BKS+05] 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.

[BKS+10] Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 488–497. IEEE Computer Society, 2010.

[Bou09] Jean Bourgain. Expanders and dimensional expansion. *Comptes Rendus Mathematique*, 347(7):357–362, 2009.

[BS94] Antal Balog and Endre Szemerédi. A statistical theorem of set addition. *Combinatorica*, 14:263–268, 1994.

[BW98] Beate Bollig and Ingo Wegener. A very simple function that requires exponential size read-once branching programs. *Inf. Process. Lett.*, 66(2):53–57, 1998.

[BY13] Jean Bourgain and Amir Yehudayoff. Expansion in $SL_2(\mathbb{R})$ and monotone expanders. *Geometric and Functional Analysis*, 23(1):1–41, 2013.

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

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 285–298, New York, NY, USA, 2016. Association for Computing Machinery.

[CGL22] Eshan Chattopadhyay, Jesse Goodman, and Jyun-Jie Liao. Affine extractors for almost logarithmic entropy. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 622–633, 2022.

[CL16a] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors, and almost optimal privacy amplification protocols. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 158–167, 2016.

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

[CL17] Eshan Chattopadhyay and Xin Li. Non-malleable codes and extractors for small-depth circuits, and affine functions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1171–1184. ACM, 2017.

[CL18] Kuan Cheng and Xin Li. Randomness extraction in AC0 and with small locality. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 37:1–37:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[CL22] Eshan Chattopadhyay and Jyun-Jie Liao. Extractors for sum of two sources. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1584–1597. ACM, 2022.

[CL23] Eshan Chattopadhyay and Jyun-Jie Liao. Hardness against linear branching programs and more. In *Proceedings of the Conference on Proceedings of the 38th Computational Complexity Conference*, CCC '23, Dagstuhl, DEU, 2023. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[CS16] Gil Cohen and Igor Shinkar. The complexity of DNF of parities. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 47–58. ACM, 2016.

[CT15] Gil Cohen and Avishay Tal. Two structural results for low degree polynomials and applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015*, 2015.

[DK11] Evgeny Demenkov and Alexander Kulikov. An elementary proof of 3n-o(n) lower bound on the circuit complexity of affine dispersers. In *Proceedings of the 36th international conference on Mathematical foundations of computer science*, pages 256–265, 2011.

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

[DS11] Zeev Dvir and Amir Shpilka. Towards dimension expanders over finite fields. *Combinatorica*, 31(3):305, 2011.

[Dun85] Paul E. Dunne. Lower bounds on the complexity of 1-time only branching programs. In Lothar Budach, editor, *Fundamentals of Computation Theory, FCT '85, Cottbus, GDR, September 9-13, 1985*, volume 199 of *Lecture Notes in Computer Science*, pages 90–99. Springer, 1985.

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

[Gál97] Anna Gál. A simple function that requires exponential size read-once branching programs. *Inf. Process. Lett.*, 62(1):13–16, 1997.

[GGMT23] W. T. Gowers, Ben Green, Freddie Manners, and Terence Tao. On a conjecture of marton, 2023.

[GI17] Ludmila Glinskih and Dmitry Itsykson. Satisfiable Tseitin Formulas Are Hard for Nondeterministic Read-Once Branching Programs. In Kim G. Larsen, Hans L. Bodlaender, and Jean-Francois Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017)*, volume 83 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:12, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[Gow98] William T. Gowers. A new proof of szemerédi's theorem for arithmetic progressions of length four. *Geometric & Functional Analysis GAFA*, 8:529–551, 1998.

[GPT22] Svyatoslav Gryaznov, Pavel Pudlák, and Navid Talebanfard. Linear Branching Programs and Directional Affine Extractors. In *37th Computational Complexity Conference (CCC 2022)*, volume 234, pages 4:1–4:16, 2022.

[Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 6–20. ACM, 1986.

[Juk88] Stasys Jukna. Entropy of contact circuits and lower bounds on their complexity. *Theor. Comput. Sci.*, 57:113–129, 1988.

[Juk95] Stasys Jukna. A note on read-k times branching programs. *RAIRO - Theoretical Informatics and Applications*, 28:75–83, 01 1995.

[Kab03] Valentine Kabanets. Almost k-wise independence and hard boolean functions. *Theor. Comput. Sci.*, 297(1-3):281–295, 2003.

[KMW91] Matthias Krause, Christoph Meinel, and Stephan Waack. Separating the eraser turing machine classes l_e, nl_e, co-nl_e and p_e. *Theor. Comput. Sci.*, 86(2):267–275, 1991.

[Li11] Xin Li. A new approach to affine extractors and dispersers. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC*, 2011.

[Li12] Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, 2012.

[Li17] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, STOC 2017, page 1144–1156, New York, NY, USA, 2017. Association for Computing Machinery.

[Li23] Xin Li. Two source extractors for asymptotically optimal entropy, and (many) more. Technical report, Arxiv, 2023. `https://arxiv.org/abs/2303.06802`.

[LY22] Jiatu Li and Tianqi Yang. 3.1n - o(n) circuit lower bounds for explicit functions. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 1180–1193, New York, NY, USA, 2022. Association for Computing Machinery.

[Nec66] E. I. Nechiporuk. On a boolean function. *Doklady of the Academy of Sciences of the USSR*, 164(4):765–766, 1966.

[Oko93] EA Okolnishnikova. On lower bounds for branching programs. *Siberian Advances in Mathematics*, 3:152–156, 1 1993.

[Pon98] Stephen Ponzio. A lower bound for integer multiplication with read-once branching programs. *SIAM Journal on Computing*, 28(3):798–815, 1998.

[PWY16] Periklis A Papakonstantinou, David P Woodruff, and Guang Yang. True randomness from big data. *Scientific reports*, 6:33740, 2016.

[Rao09] Anup Rao. Extractors for low-weight affine sources. In *Proceedings of the 2009 24th Annual IEEE Conference on Computational Complexity*, CCC '09, page 95–101. IEEE Computer Society, 2009.

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

[SS92] Janos Simon and Mario Szegedy. A new lower bound theorem for read-only-once branching programs and its applications. In *Advances In Computational Complexity Theory*, 1992.

[TV06] Terence Tao and Van H. Vu. *Additive Combinatorics*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2006.

[Vaz86] Umesh Virkumar Vazirani. *Randomness, Adversaries and Computation (Random Polynomial Time)*. PhD thesis, University of California, Berkeley, 1986. AAI8718194.

[VW08] Emanuele Viola and Avi Wigderson. Norms, xor lemmas, and lower bounds for polynomials and protocols. *Theory of Computing*, 4(7):137–168, 2008.

[Weg88] Ingo Wegener. On the complexity of branching programs and decision trees for clique functions. *J. ACM*, 35(2):461–471, 1988.

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

[Zák84] Stanislav Zák. An exponential lower bound for one-time-only branching programs. In Michal Chytil and Václav Koubek, editors, *Mathematical Foundations of Computer Science 1984, Praha, Czechoslovakia, September 3-7, 1984, Proceedings*, volume 176 of *Lecture Notes in Computer Science*, pages 562–566. Springer, 1984.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Theory of Computing*, 2007.

A Depth 3 $\mathsf{AC}^0[\oplus]$ Circuits Can Compute Optimal Directional Affine Extractors

In this section, we extend the results in [CT15] and prove depth 3 $\mathsf{AC}^0[\oplus]$ circuits can compute optimal directional affine extractors given by the probabilistic method.

Existence of Directional Affine Extractors. We first display the optimal directional affine extractor.

Claim 71. *There exist universal constants n_0, c such that the following holds. For every $\varepsilon > 0$ and $n > n_0$ there exists a directional affine extractor for dimension k with bias ε, $F : \mathsf{F}_2^n \to \mathsf{F}_2$, where $k = \log \frac{n}{\varepsilon^2} + \log\log \frac{n}{\varepsilon^2} + c$.*

Proof. For the purpose of this proof, it is more convenient to work with the definition of DAExt in [GPT22].

Definition 30. *A boolean function $f : \mathsf{F}_2^n \to \mathsf{F}_2$ is a directional affine extractor for dimension d with bias ε if for every affine subspace X, every non-zero a, it holds that*

$$\mathsf{DAExt}(X) + \mathsf{DAExt}(X + a) \approx_\varepsilon U_1.$$

This definition is equivalent to Definition 2 up to a quadratic blow-up in the error. Check Appendix B in [CL23] for a proof. Let $F : \mathsf{F}_2^n \to \mathsf{F}_2$ be a random function, namely, $\{F(x), x \in \mathsf{F}_2^n\}$ are fresh random bits. Fix an affine subspace $U \subseteq \mathsf{F}_2^n$ of dimension k, a non-zero $a \in \mathsf{F}_2^n$. Depending on whether $U + a$ coincides with U, there are two cases to consider.

Case 1. $U + a \neq U$. For any $x_1, x_0 \in U$, $x_1 \neq x_0$, since $x_1 + x_0 \in U$, it holds that $x_1 + a \notin \{x_0, x_0 + a\}$. Therefore, $\{F(x) + F(x + a), x \in U\}$ are independent random bits and it holds that

$$\mathbf{Pr}\left[\frac{1}{2^k}\left|\sum_{x\in U}(-1)^{F(x)+F(x+a)}\right| \geq \varepsilon\right] \leq 2 \cdot e^{-\frac{2^k\varepsilon^2}{2}}. \qquad \text{(Hoeffding Inequality)}$$

Case 2. $U + a = U$. For any $x_1, x_0 \in U$ $x_1 \neq x_0$, $x_1 \in \{x_0, x_0 + a\} \iff x_1 = x_0 + a$. If this is the case, then $F(x)+F(x+a) = F(x+a)+F((x)$. Therefore, $\{\left((-1)^{F(x)+F(x+a)} + (-1)^{F(x+a)+F(x)}\right)/2, x \in \mathsf{F}_2^n\}$ are independent random variables supported on $\{-1, 1\}$ and it holds that

$$\mathbf{Pr}\left[\frac{1}{2^k}\left|\sum_{x\in U}(-1)^{F(x)+F(x+a)}\right| \geq \varepsilon\right] \leq 2 \cdot e^{-\frac{2^{k-1}\varepsilon^2}{2}}. \qquad \text{(Hoeffding Inequality)}$$

The number of pairs of affine subspaces of the same underlining linear subspace is bounded by $\binom{2^n}{2}\binom{2^n}{k} \leq 2^{(k+2)n}$. Hence by Union Bound over all pairs of affine subspaces of the same underlining linear subspace, if $2^{(k+2)n} \cdot 2 \cdot e^{-\frac{2^k\varepsilon^2}{2}} = 2^{(k+2)n+1-\frac{2^k\varepsilon^2}{\ln 4}} < 1$ and $2^{(k+2)n} \cdot 2 \cdot e^{-\frac{2^{k-1}\varepsilon^2}{2}} = 2^{(k+2)n+1-\frac{2^{k-1}\varepsilon^2}{\ln 4}} < 1$ then there exists a directional affine extractor of dimension k with error ε. It is verified that the same choice of $k = \log \frac{n}{\varepsilon^2} + \log\log \frac{n}{\varepsilon^2} + c$ for some fixed constant c as in [CT15] suffices for the above inequalities to hold.

Existence of Sumset Linear Injectors. The following definition of sumset linear injectors slightly generalize the notion of injector in [CT15]. They will be applied in the construction of a more "structured" random function which is a DAExt.

Definition 31. *An (n, k_1, k_2, d) sumset linear injector with size m is a family of $d \times n$ matrices $\{A_1, \cdots, A_m\}$ over F_2 with the following property: for every pairs of subspaces $U, V \subseteq \mathsf{F}_2^n$ of dimension k_1, k_2 respectively where $\dim(U \cap V) \leq 1$, there exists an $i \in [m]$ such that $\ker(A_i) \cap (U + V) = \{0\}$.*

Lemma 72. *For every n, k_1, k_2 such that $2 \leq k_1, k_2 \leq n$, there exists an $(n, k_1, k_2, k_1 + k_2 + 1)$ linear injector with size $m = n(k_1 + k_2)$.*

Proof. Fix a pair of subspaces $U, V \subseteq \mathsf{F}_2^n$ of dimension k_1, k_2 respectively where $U \cap V = \{0\}$. Let A be a $d \times n$ matrix such that every entry of A is sampled from F_2 uniformly and independently at random. For every $u + v \in (U + V) \setminus \{0\}$ it holds that $\mathbf{Pr}[A(u + v) = 0] = 2^{-d}$. By taking the union bound over all pairs of elements in $U \setminus \{0\}$ and $V \setminus \{0\}$, we get that

$$\mathbf{Pr}\left[\ker(A) \cap (U + V) \neq \{0\}\right] \leq 2^{k_1+k_2-d}.$$

Let $A_1, \cdots, A_m$ be $d \times n$ matrices such that the entry of each of the matrices is sampled from F_2 uniformly and independently at random. By the above equation, it holds that

$$\mathbf{Pr}\left[\forall i \in [m]\ \ker(A_i) \cap (U + V) \neq \{0\}\right] \leq 2^{m(k_1+k_2-d)}.$$

The number of sum of two linear subspaces of dimension k_1 and k_2 is bounded by $\binom{2^n}{k_1}\binom{2^n}{k_2}$, which is bounded above by $2^{n(k_1+k_2)-2}$ for $k \geq 2$. Thus if $2^{n(k_1+k_2)-2} \cdot 2^{m(k_1+k_2-d)} < 1$ there exists an (n, k_1, k_2, d) linear injector with size m. The latter equation holds for $d = k_1 + k_2 + 1$ and $m = n(k_1 + k_2)$. □

More Structured Random Functions. Now we apply the sumset injector to reduce the randomness used in Claim 71.

Lemma 73. *Let n_0, c be the constants from Claim 71. Let $n > n_0$ and let k, ε be such that $k = \log \frac{n}{\varepsilon^2} + \log\log \frac{n}{\varepsilon^2} + c$. Let $\{A_1, \cdots, A_m\}$ be an $(n, k, 2, d)$ linear injector with size m. Then, there exists functions $f_1, \cdots, f_m : \mathsf{F}_2^d \to \mathsf{F}_2$ such that the function $f : \mathsf{F}_2^n \to \mathsf{F}_2$ defined by*

$$f(x) = \bigoplus_{i=1}^{m} f_i(A_i x) \tag{6}$$

is a directional affine extractor for dimension k with bias ε.

Proof. The proof idea is that "$(U + V)$-wise" independence, where U is any affine subspace and $V = \{0, a\} = \mathrm{span}\{0, a\}$ for any $a \neq 0 \in \mathsf{F}_2$ suffices for the proof of Claim 71. In other words, we only need $\{f(x)\}_{x \in U \cup (U+a)}$ to be independent random bits, instead of full independence over the truthtable of f. We now construct such a random function, and by replacing the random function in the proof of Claim 71 with this newly constructed function, we find optimal directional affine extractors in a restricted class of random functions. This will enable us to argue about its complexity. Let $F_1, \cdots, F_m : \mathsf{F}_2^d \to \mathsf{F}_2$ be independent random functions, that is, the random bits $\left\{F_i(x) : i \in [m], x \in \mathsf{F}_2^d\right\}$ are independent. Define the random function $F : \mathsf{F}_2^n \to \mathsf{F}_2$ as follows

$$F(x) = \bigoplus_{i=1}^{m} F_i(A_i x).$$

Let $(U, U + a)$ be any pair of affine subspaces of the same underlining linear subspace U', let $V = \mathrm{span}\{0, a\}$. By Definition 31, there exists an $i \in [m]$ such that $\ker(A_i) \cap (U' + V) = \{0\}$. This implies that for every two distinct elements $u, v \in U$ it holds that $A_i(u), A_i(v), A_i(u + a), A_i(v + a)$ are pairwise distinct. Otherwise we would reach the contradiction that $A_i(u + v) = 0$ or $A_i(a) = 0$ or $A_i(u + v + a) = 0$ and thus $u + v$ or a or $u + v + a$, a non-zero vector in $U' + V$, lies in $\ker(A_i)$. Since F_i is a random function, and A_i is injective on $U \cup (U + a)$, the random bits $\{F_i(u)\}_{u \in U \cup (U+a)}$ are independent. Since for all $x \in (U \cup (U + a))$, the fresh random coin $F_i(A_i x)$ is used and only used to generate $F(x)$, it holds that $F(x)$ is independent and random in $U \cup (U + a)$. □

Theorem 74. *Let f be the function from Eqn. (6), where $\{A_1, \cdots, A_{n(k+2)}\}$ is the $(n, k, 2, k+3)$ sumset linear injector from Lemma 72. Then, f is a directional affine extractor of dimension k and error ε, where $k = \log \frac{n}{\varepsilon^2} + \log\log \frac{n}{\varepsilon^2} + O(1)$. Moreover,*

1. $\deg(f) = \log \frac{n}{\varepsilon^2} + \log\log \frac{n}{\varepsilon^2} + O(1)$.
2. *f can be realized by a* XOR-AND-XOR *circuit of size $O((n/\varepsilon)^2 \cdot \log^3(n/\varepsilon))$.*
3. *f can be realized by a De Morgan formula of size $O((n^5/\varepsilon^2) \cdot \log^3(n/\varepsilon))$.*

Proof. Exactly the same as [CT15]. □

□

B Missing Proofs

B.1 Proof of Lemma 8

We recall Lemma 8.

Lemma 8 (Affine conditioning [Li11])**.** *Let X be any affine source on $\{0,1\}^n$. Let $L : \{0,1\}^n \to \{0,1\}^m$ be any affine function. Then there exist independent affine sources A, B such that:*

- $X = A + B$
- *There exists $c \in \{0,1\}^m$, such that for every $b \in \mathsf{Supp}(B)$, it holds that $L(b) = c$.*
- *$H(A) = H(L(A))$ and there exists an affine function $L^{-1} : \{0,1\}^m \to \{0,1\}^n$ such that $A = L^{-1}(L(A))$.*
- *$H(X\,|_{L(X)=\ell}) = H(B)$ for all $\ell \in \mathsf{Supp}(L(X))$.*

Proof. We prove the second and fourth bullet points.
2nd bullet point. Let $L = \overline{L} + c'$ where $\overline{L} : \{0,1\}^n \to \{0,1\}^m$ is a linear function. Consider the set $\mathsf{Supp}(X) \cap \mathsf{Ker}(\overline{L})$ which is a linear subspace, let B be this linear subspace with an arbitrary affine shift c'', then it holds that $L(B) = L(c'') = \overline{L}(c'') + c' := c$. Let $A = X - B$. Then $L(A) = L(X) - L(B) = \overline{L}(X) + c' - c = \overline{L}(X - (B - c'')) - \overline{L}(c'') = \mathsf{Supp}(X) \cap \mathsf{Span}(\overline{L}) - \overline{L}(c'')$.
4th bullet point. For any $\ell \in \mathsf{Supp}(L(X))$, conditioned on the fixing of $L(X) = \ell$, by the second bullet it holds that $L(X) = L(A) + L(B) = L(A) + c = \ell$. By the third bullet, this implies $A = L^{-1}(L(A)) = L^{-1}(\ell - c)$. Therefore, $H(X\,|_{L(X)=\ell}) = H(L^{-1}(\ell - c) + B) = H(B)$, thus independent of ℓ. □

B.2 Proof of Lemma 35

We recall Lemma 35.

Lemma 35 (Independence-merging lemma for affine sources)**.** *Let $\mathsf{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be any (k, ε)-strong linear seeded extractor, $X_0 \in \{0,1\}^n$ an affine source, $X, X^{[t]} \in \{0,1\}^n$, $Y, Y^{[t]} \in \{0,1\}^d$ all linear functions of X_0, $W = \mathsf{LExt}(X, Y)$ and $W^j = \mathsf{LExt}(X^j, Y^j)$ for every $j \in [t]$. Suppose there exists $S, T \subseteq [t]$ such that*

- *$(Y, Y^S) \approx_\delta (U_d, Y^S)$;*

- $H(X \mid X^T, Y, Y^{[t]}) \geq k + tm$,

then

$$W \approx_{\varepsilon+\delta} U_m \mid (W^{S\cup T}, Y, Y^{[t]}).$$

Proof. First note that since $X, X^T, Y, Y^{[t]}$ are linear functions of X_0, by Lemma 8, the entropy of X given $X^T, Y, Y^{[t]}$ is constant. Therefore, it suffices to use shannon entropy H instead of average case min-entropy $\widetilde{H}_\infty$.
Conditioned on the fixings of Y^S, it holds that W^S are linear functions of X^S and therefore linear functions of X_0. By Lemma 8, there exists affine sources $A = W^S(X)$ and B such that $X = A + B$. By Lemma 9, $H(B) = H(X \mid Y^S, W^S) \geq H(X) - |S| \cdot m$. Now further condition on $(X^T, Y, Y^{[t]\setminus S})$, we have that $H(B \mid X^T, Y, Y^{[t]\setminus S}) \geq H(X \mid X^T, Y, Y^T) - |S| \cdot m \geq k$. By Proposition 41, it follows that with probability $1 - \varepsilon$, $\mathsf{LExt}(X, Y) = \mathsf{LExt}(B, Y) + \mathsf{LExt}(A, Y) = \mathsf{LExt}(B, Y) + \text{const} = U_m$. Since W^T is a deterministic function of X^T and Y^T, what we have shown implies

$$W \approx_{\varepsilon+\delta} U_m \mid (W^{S\cup T}, Y, Y^{[t]}).$$

□

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