Extracting Computational Entropy and Learning Noisy Linear Functions

Chia-Jung Lee, Chi-Jen Lu, and Shi-Chun Tsai, *Member, IEEE*

***Abstract*—We study the task of deterministically extracting randomness from sources containing computational entropy. The sources we consider have the form of a conditional distribution $(f(\mathcal{X})|\mathcal{X})$, for some function f and some distribution $\mathcal{X}$, and we say that such a source has computational min-entropy k if any circuit of size 2^k can only predict $f(x)$ correctly with probability at most 2^{-k} given input x sampled from $\mathcal{X}$. We first show that it is impossible to have a seedless extractor to extract from one single source of this kind. Then we show that it becomes possible if we are allowed a seed which is weakly random (instead of perfectly random) but contains some statistical min-entropy, or even a seed which is not random at all but contains some computational min-entropy. This can be seen as a step toward extending the study of multisource extractors from the traditional, statistical setting to a computational setting. We reduce the task of constructing such extractors to a problem in computational learning theory: learning linear functions under arbitrary distribution with adversarial noise, and we provide a learning algorithm for this problem. In fact, this problem is a well-recognized one in computational learning theory and variants of this problem have been studied intensively before. Thus, in addition to its application to extractors, our learning algorithm also has independent interest of its own, and it can be considered as the main technical contribution of this paper.**

***Index Terms*—Computational min-entropy, randomness extractors, learning linear functions, computational complexity.**

I. Introduction

RANDOMNESS has become a useful tool in computer science, as the most efficient algorithms known for many important problems are randomized. However, when analyzing the performance of a randomized algorithm, we usually assume that the algorithm has access to a perfectly random source. In reality, the random sources we have access to are usually not perfect but may contain some amount of randomness. The amount of randomness in a source is usually measured by its min-entropy, where a source has min-entropy at least k if every element occurs with probability at most 2^{-k}. From a source with some min-entropy, we would like to have a procedure, called an *extractor* [30], [22], to extract almost perfect randomness, which can then be used for randomized algorithms.

Most works on extractors focused on *seeded* extractors, which can utilize an additional seed to aid the extraction. There has been a long and fruitful line of results on constructing seeded extractors (see [25] for a nice survey), which culminated in [21] and [13] with an optimal construction (up to constant factors). However, there is an issue with using seeded extractors. Namely, we need a seed which is perfectly random and independent of the source we extract from. How do we get such a seed? For some applications, this can be taken care of (e.g., by enumerating through all possible seed values), but for others, this seems to go back to the problem which we try to solve using extractors. Can we get rid of the need for a seed and have *seedless* extractors? For general sources, the answer has been known to be negative [7]. On the other hand, when the sources are restricted and have special structure, it becomes possible to have seedless extractors. Examples of such sources include samplable sources [28], bit-fixing sources [8], [18], [10], independent-symbol sources [17], [19], and multiple independent sources [7], [2], [3], [24], [6], [23].

In this paper, we would like to look for a more general class of sources from which seedless extraction is still possible. In particular, we will consider sources which may contain no randomness at all in a statistical sense, but *look* slightly random to computational-bounded observers, such as small circuits. That is, we will go from a traditional, statistical setting to a computational one. It is conceivable that in many situations when we consider a source random, it may in fact only appear so to us, while its actual statistical min-entropy may be much smaller (or even zero) especially if we take into account some correlated information which we can observe. Another application of this notion is in cryptography, and in fact the idea of extracting computational randomness has appeared implicitly long ago since [29], [11], [14], for the task of constructing pseudorandom generators from one-way functions. The idea is that given a one-way function g, it is hard to invert $g(y)$ to get y, and this means that given the (correlated) information $g(y)$, y still looks somewhat random, from which one can extract some bits that look almost random. However, while there is a natural and well-accepted definition for what it means that a distribution looks almost random [29], it seems less clear how to define that a distribution looks slightly random and how to measure the amount of randomness in it. In fact, there are several alternatives which all seem reasonable,

Manuscript received October 05, 2009; revised November 10, 2010; accepted December 08, 2010. Date of current version July 29, 2011. The work of C.-J. Lu was supported in part by the National Science Council of Taiwan under Contract NSC-97-2221-E-001-012-MY3. The work of S.-C. Tsai was supported in part by the National Science Council of Taiwan under Contracts NSC-97-2221-E-009-064-MY3 and NSC-98-2221-E-009-078-MY3. The material in this paper was presented in part at the 15th International Computing and Combinatorics Conference (COCOON), Niagara Falls, NY, 2009.

C.-J. Lee was with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan. She is now with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan (e-mail: leecj@iis.sinica.edu.tw).

C.-J. Lu is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan (e-mail: cjlu@iis.sinica.edu.tw).

S.-C. Tsai is with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: sctsai@csie.nctu.edu.tw).

Communicated by K. M. Martin, Associate Editor for Complexity and Cryptography.

Digital Object Identifier 10.1109/TIT.2011.2158897

but there are provable discrepancies among them [4], [15]. To extract randomness from a source with so-called HILL-entropy [4], the strongest among them, one can simply use any statistical extractor, but we would like to extract randomness from a broader class of sources. Here we consider a weaker (more general) notion of computational randomness, which appears in [15], and we call it *computational min-entropy*. A comparison with other notions of computational randomness can be found in [15].

A. Computational Min-Entropy

To model the more general situation that one may observe some correlated information about the source, we consider the setting with a pair of jointly distributed random variables $\mathcal{V}$ and $\mathcal{X}$, where $\mathcal{V}$ is the source from which we want to extract and $\mathcal{X}$ (could be empty) is some information which one can observe. To stress that we want to measure the randomness of $\mathcal{V}$ conditioned on $\mathcal{X}$ and to extract randomness from $\mathcal{V}$ given the information $\mathcal{X}$, we use the notation $(\mathcal{V}|\mathcal{X})$ to denote such a joint distribution. The correlation between $\mathcal{V}$ and $\mathcal{X}$ is modeled by $\mathcal{V} = f(\mathcal{X})$ for some function f. In the example of one-way permutation, f is the inverse function g^{-1}, which is hard to compute, and $\mathcal{X}$ is the distribution of $g(y)$ over a random y. Here in our definition, we allow f to be probabilistic and we even do not require it to have an efficient (or even computable) algorithm, and furthermore, we do not require $\mathcal{X}$ to be efficiently samplable either. We say that such a source $(f(\mathcal{X})|\mathcal{X})$ has computational min-entropy k if given input x sampled from $\mathcal{X}$, any circuit of size 2^k can only predict $f(x)$ correctly with probability at most 2^{-k}.[1] From the distribution $f(\mathcal{X})$, we would like to extract randomness which when given $\mathcal{X}$ still looks random to circuits of a certain size. Note that a source $\mathcal{Y}$ with statistical min-entropy k can be seen as such a source $(f(\mathcal{X})|\mathcal{X})$ with computational min-entropy k, where we can simply have no $\mathcal{X}$ or just have $\mathcal{X}$ taking a fixed value, and let f be a probabilistic function with $\mathcal{Y}$ as its output distribution. This means that extractors for sources with computational min-entropy can immediately work for sources with statistical min-entropy, and thus results in the computational setting can be seen as a generalization of those in the traditional, statistical setting. On the other hand, for a deterministic function f, $f(x)$ has no statistical min-entropy at all when given x. Still, according to our definition, as long as f is hard to compute, $(f(\mathcal{X})|\mathcal{X})$ in fact can have high computational min-entropy.

Extractors for such sources were implicitly proposed before [11], [14], and they are seeded ones. That is, they need an additional seed which must be perfectly random and independent of the source. In fact, it is known that any seeded statistical extractor with some additional *reconstruction* property (in the sense of [27]) gives a seeded extractor for such sources [4], [26], [15]. However, just as in the statistical setting, several natural questions arise in the computational setting too. To extract from such sources, do we really need a seed? Can we use a weaker seed which is only slightly random, instead of perfectly random, in a statistical sense, or an even weaker seed which only looks slightly random in a computational sense but may contain no randomness in a statistical sense? Seeing the seed as an additional independent source, a general question is: Can we have seedless extractors for multiple independent sources in which each source contains some computational min-entropy? We will try to answer these questions in this paper. One can see this as a step toward extending the study of multisource extractors from the traditional, statistical setting to a new, computational setting. One can also see this as providing a finer map for the landscape of statistical extractors, according to the degree of their reconstruction property.

[1] A more general definition is to have the circuit size as a separate parameter, but our extractor construction does not seem to work for this more general definition.

B. Our Results

First, we show that it is impossible to have seedless extractors for one single source, even if the source of length n can have a computational min-entropy as high as $n-2$ and even if we only want to extract one bit.

Next, we show that with the help of a weak seed, it becomes possible to extract randomness from such sources. We use a two-source extractor of Lee *et al.* [20], denoted as EXT, which takes two input strings $v, w \in \{0,1\}^n$, sees them as vectors from $\mathbb{F}^\ell$, where $\mathbb{F} = GF(2^m)$ for some m with $n = m\ell$, and outputs their inner product, denoted as $\langle v, w\rangle$, over $\mathbb{F}$. As shown in [20], it works for any two independent sources both containing some statistical min-entropy. Moreover, it is also known to work when one source contains some computational min-entropy and the other, the seed, is perfectly random (in a statistical sense) [12]. Our second result shows that it even works when the seed only contains some statistical min-entropy. More precisely, we show that given any source $(f(\mathcal{X})|\mathcal{X})$ with computational min-entropy $k_1 = n - k + O(k/\log k)$ and another independent source $\mathcal{W}$ with statistical min-entropy k, the output $\mathrm{EXT}(f(\mathcal{X}), \mathcal{W})$ given $\mathcal{X}$ cannot be distinguished from random with advantage $\varepsilon = 2^{-O(\sqrt{k/\log k})}$ by circuits of size $s = 2^{n-k+O(k/\log k)}$. That is, for any such Boolean circuit D, $|\Pr[D(\mathcal{X}, \mathrm{EXT}(f(\mathcal{X}))) = 1] - \Pr[D(\mathcal{X}, \mathcal{U}) = 1]| \le \varepsilon$, where $\mathcal{U}$ denotes the uniform distribution. Then we proceed to show that the extractor even works when the seed only contains computational min-entropy. More precisely, when we replace the source $\mathcal{W}$ by a source $(g(\mathcal{Y})|\mathcal{Y})$ with computational min-entropy k, $\mathrm{EXT}(f(\mathcal{X}), g(\mathcal{Y}))$ given $(\mathcal{X}, \mathcal{Y})$ still cannot be distinguished with advantage ε by circuits of size about s. This can be seen as a seedless extractor for two independent sources, both with computational min-entropy.

We do not know if the statistical extractors of [2], [3], [24], [6], and [23] for multiple independent sources can also work in the computational setting, since to work in this setting, we need them to have some reconstruction property. For the extractors from [11] and [12], this property can be translated to a task in learning theory, and the proofs there can be recast as providing an algorithm for learning linear functions under *uniform* distribution with adversarial noise. Our second result can be seen as a generalization of [11] and [12], but we are facing a more challenging learning problem: learning linear functions under *arbitrary* distribution with adversarial noise. Our third result provides an algorithm for this problem, which, in addition to being used to prove our second result, may have interest of its own.

In the learning problem, there is some unknown linear function $v : \mathbb{F}^\ell \to \mathbb{F}$, defined as $v(w) = \langle v, w\rangle$, which we want

to learn, and there is a distribution $\mathcal{W}$ over $\mathbb{F}^\ell = \{0,1\}^n$ from which we can sample w to obtain a training example $(w, q(w))$, for some function $q : \mathbb{F}^\ell \to \mathbb{F}$. The function q can be seen as a noisy version of v with some noise rate α, and there are two noise models. In the adversarial-noise model, q is a deterministic function such that $\Pr_{w\in\mathcal{W}}[q(w) \neq v(w)] \leq \alpha$. In the random-noise model, q is a probabilistic function such that independently for any w, $\Pr[q(w) \neq v(w)] \leq \alpha$. We consider the more difficult adversarial-noise model, and our algorithm works for an arbitrary distribution $\mathcal{W}$, while its complexity depends on the min-entropy k of $\mathcal{W}$. More precisely, our algorithm samples $2^{O(k/\log k)}$ training examples, runs in time $2^{n-k+O(k/\log k)}$, and with high probability outputs a list containing every linear function v satisfying $\Pr_{w\in\mathcal{W}}[q(w) \neq v(w)] \leq \alpha$, for $\alpha = (1 - 1/|\mathbb{F}|) - 2^{-O(\sqrt{k/\log k})}$. The factor 2^{n-k} in our running time is in fact unavoidable because one can easily find a distribution $\mathcal{W}$ (e.g., the first k bits perfectly random and the rest fixed) for which the number of such v's, and thus the running time, is in fact at least 2^{n-k}. Note that when $\mathcal{W}$ is the uniform distribution (with $k = n$), our algorithm runs in time $2^{O(n/\log n)}$ and takes $2^{O(n/\log n)}$ samples.

Previously, the algorithm of Blum, Kalai, and Wasserman [5] can learn under arbitrary distribution but in the random-noise model, while that of Feldman *et al.* [9] can learn in the adversarial-noise model but under the uniform distribution. Both algorithms learn the parity functions on n variables, tolerate a noise rate $\alpha \leq 1/2 - \Omega(1)$, run in time $2^{O(n/\log n)}$, and take $2^{O(n/\log n)}$ samples. Very recently, Kalai, Mansour, and Verbin [16] gave an algorithm which can learn the parity functions under arbitrary distribution in the adversarial-noise model, but the hypothesis they produce is not in the linear form, so it cannot be used for our extractors. Furthermore, their algorithm only produces one hypothesis instead of all the legitimate ones, and their technique does not seem to generalize from the parity functions to the linear functions over larger fields. Thus, to the best of our knowledge, the task our learning algorithm achieves has not been accomplished before. Finally, just as the result of [11] can yield a list-decoding algorithm for Hadamard codes, so can ours, while that of [16] cannot. In fact, our list-decoding algorithm can work even when all but 2^k symbols from the codeword are erased and an α fraction of the remaining symbols are corrupted. It can also be seen as list-decoding a *punctured* Hadamard code, where a punctured code is obtained from a code by deleting all but a small number of symbols from the codeword.

C. Our Techniques

For our impossibility result, we show that for any function $\mathrm{EXT} : \{0,1\}^n \to \{0,1\}$, there exists a function $f : \{0,1\}^{3n} \to \{0,1\}^n$ such that $(f(\mathcal{X})|\mathcal{X})$ has computational min-entropy $n-2$, but $\mathrm{EXT}(f(x))$ takes an identical value for all x. We show the existence of such a function f by a standard probabilistic argument: in fact, a random function from $\{0,1\}^{3n}$ to $\mathrm{EXT}^{-1}(b)$ is likely to work, for the $b \in \{0,1\}$ with the larger $\mathrm{EXT}^{-1}(b)$.

To show that our extractor works in the computational setting, we follow the approach of [11] and reduce it to the task of learning linear functions as we just discussed. More precisely, for the case when the source $(f(\mathcal{X})|\mathcal{X})$ has computational min-entropy and the seed $\mathcal{W}$ has statistical min-entropy, the reduction works as follows. Assume our extractor EXT does not work, and thus some efficient distinguisher can tell the distribution of $\mathrm{EXT}(f(x), \mathcal{W}) = \langle f(x), \mathcal{W}\rangle$ from random given x, for a large fraction of x from $\mathcal{X}$. For any such x, we can then predict the value $\langle f(x), \mathcal{W}\rangle$ with a good probability, given the ability to sample from $\mathcal{W}$, which can then be used by the learning algorithm to learn $f(x)$. This would give us an efficient algorithm for predicting $f(x)$ for those x's, if we could in fact sample $\mathcal{W}$ efficiently. However, this may not be the case in general as $\mathcal{W}$ could be any arbitrary distribution. Still, by an average argument, there must exist a small set of samples from $\mathcal{W}$ which preserve this predicting probability, so we can hard-wire them in to get a circuit which predicts f well. If the function f is hard, this is impossible, so the assumed distinguisher cannot exit, and EXT indeed works. For the case that the seed comes from a distribution $(g(\mathcal{Y})|\mathcal{Y})$ with computational min-entropy, observe that $g(\mathcal{Y})$ alone (without conditioning on $\mathcal{Y}$) must have some statistical min-entropy, because otherwise it becomes easy to predict. Then a very similar argument as above can be used.

Note that our results on extractors still depend on the existence of a good learning algorithm, and our main technical contribution can be seen as providing such an algorithm. Our algorithm can be seen as extending that of [5] from the random-noise model to the adversarial-noise model. Note that in the random-noise model, it is possible to predict the value of $v(w)$ with confidence for an input w by taking the majority vote on several independent predictions, while in the adversarial-noise model, this does not seem so and the learning task becomes much harder.

Our learning algorithm works as follows. We start by sampling some number K of training examples $(w, q(w))$ from $(\mathcal{W}, q(\mathcal{W}))$. Note that each example $(w, q(w))$ gives us a linear equation $\langle v, w\rangle = q(w)$ for the unknown v, so the K examples gives us a system of K linear equations, some of which may be wrong. We reduce the original problem of learning the unknown v to the problem of solving such a noisy system of linear equations. To solve the system, we proceed in two phases. In the forward phase, we start from the system, and use several iterations to produce smaller and smaller systems with fewer and fewer variables, until we have a small enough system which we can afford to solve using brute force. Then we enter the backward phase, and starting from the last system produced by the forward phase, we work backward on larger and larger systems produced in the forward phase to obtain solutions for more and more variables. Since the possible solutions may not be unique, we keep them all in a list in each iteration, and the list in the final iteration of the backward phase is our output, which we hope contains the correct v.

The forward phase is similar in spirit to an approach in [5]. The key is to guarantee that after each iteration, the solution v is still good for the new system in the sense that the new system still contains a good fraction of correct equations with respect to v, so that v will not be lost when solving this new system. Using an argument similar to that in [5], we can show that this does hold with a significant probability. On the other hand, it is not clear whether or not some iteration in the forward phase would turn many originally bad solutions into good ones for the new system (satisfying a good fraction of its equations). That is, not only is v a good so-

lution for the system, there are in fact too many good solutions for it. If this happens, then in the backward phase when we try to solve this system, we cannot afford to keep all such solutions, and we have the risk of losing the actual solution v. This tricky situation does not arise in the random-noise model considered in [5], so a much simpler algorithm works there. However, in the adversarial-noise model, this seems unavoidable. Fortunately, we can show that with high probability, the systems we produce indeed do not have too many good solutions. This turns out to rely on the fact that our extractor is also a good *statistical* extractor, together with the property, which we will show, that each system is likely to have a distribution which is close to some good distribution with high statistical min-entropy.

II. Preliminaries

For any $m \in \mathbb{N}$, let $\mathcal{U}_m$ denote the uniform distribution over $\{0,1\}^m$. Let $\mathsf{SIZE}(s)$ be the class of functions computable by Boolean circuits of size s. We say that a function $D : \{0,1\}^n \to \{0,1\}$ is an ε-distinguisher for two distributions $\mathcal{X}$ and $\mathcal{Y}$ over $\{0,1\}^n$ if

$$|\Pr[D(\mathcal{X}) = 1] - \Pr[D(\mathcal{Y}) = 1]| \geq \varepsilon.$$

All logarithms in this paper will have base two.

We consider two types of min-entropy: *statistical* min-entropy and *computational* min-entropy. The notion of statistical min-entropy is a standard one, usually just called min-entropy.

Definition 1: We say that a distribution $\mathcal{X}$ has *statistical min-entropy* at least k, denoted by $\mathrm{H}_\infty(\mathcal{X}) \geq k$, if for any x, $\Pr[\mathcal{X} = x] \leq 2^{-k}$.

Next, we define the notion of computational min-entropy. Here, we consider the more general setting of measuring the randomness of a distribution $\mathcal{V}$ given a correlated distribution $\mathcal{X}$, and we use $(\mathcal{V}|\mathcal{X})$ to denote such a joint distribution. The correlation between $\mathcal{V}$ and $\mathcal{X}$ is modeled by $\mathcal{V} = f(\mathcal{X})$ for some function f, which could be either probabilistic or deterministic.

Definition 2: We say that a distribution $(\mathcal{V}|\mathcal{X})$ has *computational min-entropy* k, denoted by $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) = k$, if for any $C \in \mathsf{SIZE}(2^k)$, $\Pr[C(\mathcal{X}) = \mathcal{V}] \leq 2^{-k}$.

We consider three kinds of extractors: *statistical* extractors, *hybrid* extractors and *computational* extractors. The notion of statistical extractors is a standard one for 2-source extractors, usually just called 2-source extractors, while we introduce the notions of hybrid extractors and computational extractors.

Definition 3: A function $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is called a

- (k_1, k_2, ε)-*statistical-extractor* if for any source $\mathcal{V}$ with $\mathrm{H}_\infty(\mathcal{V}) \geq k_1$ and any source $\mathcal{W}$, independent of $\mathcal{V}$, with $\mathrm{H}_\infty(\mathcal{W}) \geq k_2$, there is no ε-distinguisher (without any complexity bound) for the distributions $(\mathcal{W}, \mathrm{Ext}(\mathcal{V}, \mathcal{W}))$ and $(\mathcal{W}, \mathcal{U}_m)$.
- $(k_1, k_2, \varepsilon, s)$-*hybrid-extractor* if for any source $(\mathcal{V}|\mathcal{X})$ with $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$ and any source $\mathcal{W}$, independent of $(\mathcal{V}|\mathcal{X})$, with $\mathrm{H}_\infty(\mathcal{W}) \geq k_2$, there is no ε-distinguisher in $\mathsf{SIZE}(s)$ for the distributions $(\mathcal{X}, \mathcal{W}, \mathrm{Ext}(\mathcal{V}, \mathcal{W}))$ and $(\mathcal{X}, \mathcal{W}, \mathcal{U}_m)$.
- $(k_1, k_2, \varepsilon, s)$-*computational-extractor* if for any source $(\mathcal{V}|\mathcal{X})$ with $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$ and any source $(\mathcal{W}|\mathcal{Y})$, independent of $(\mathcal{V}|\mathcal{X})$, with $\mathrm{H}_c(\mathcal{W}|\mathcal{Y}) \geq k_2$, there is no ε-distinguisher in $\mathsf{SIZE}(s)$ for the distributions $(\mathcal{X}, \mathcal{Y}, \mathcal{W}, \mathrm{Ext}(\mathcal{V}, \mathcal{W}))$ and $(\mathcal{X}, \mathcal{Y}, \mathcal{W}, \mathcal{U}_m)$.

Remark 1: Note that the definition above corresponds to the notion of strong extractors in the setting of seeded statistical extractors, which guarantees that even given the seed (the second source), the output still looks random.

We will need the following statistical extractor from [20], which generalizes the construction from [7]. For any $m \in \mathbb{N}$ with $m|n$, let $\ell = n/m$, and see any $x \in \{0,1\}^n$ as an ℓ-dimensional vector $x = (x_1, x_2, \ldots, x_\ell)$ over $\mathbb{F} = GF(2^m)$. Then for any $x, y \in \mathbb{F}^\ell$, let $\langle x, y \rangle$ be their inner product over $\mathbb{F}$ defined as

$$\langle x, y \rangle = \sum_{i=1}^{\ell} x_i \cdot y_i.$$

Theorem 1: [20] The function $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ defined as $\mathrm{Ext}(u, v) = \langle u, v \rangle$ is a (k_1, k_2, ε)-statistical-extractor when $k_1 + k_2 \geq n + m + 2\log(1/\varepsilon) - 2$.

We will need the following fact about statistical extractors.

Lemma 1: Let $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ be any (k_1, k_2, ε)-statistical-extractor. Then for any source $\mathcal{W}$ over $\{0,1\}^n$ with $\mathrm{H}_\infty(\mathcal{W}) = k_2$ and any function $q : \{0,1\}^n \to \{0,1\}^m$, there are at most 2^{k_1} different v's satisfying

$$\Pr_{w \in \mathcal{W}}[q(w) = \mathrm{Ext}(v, w)] \geq 1/2^m + \varepsilon.$$

Proof: Let V be the set consisting of such v's and let $\mathcal{V}$ be the uniform distribution over V. Consider the distinguisher D defined as $D(w, u) = 1$ if $q(w) = u$ and $D(w, u) = 0$ otherwise. Then, the difference

$$\Pr_{v \in \mathcal{V}, w \in \mathcal{W}}[D(w, \mathrm{Ext}(v, w)) = 1] - \Pr_{w \in \mathcal{W}, u \in \mathcal{U}_m}[D(w, u) = 1]$$

is equal to

$$\Pr_{v \in \mathcal{V}, w \in \mathcal{W}}[q(w) = \mathrm{Ext}(v, w)] - \Pr_{w \in \mathcal{W}, u \in \mathcal{U}_m}[q(w) = u]$$

which is at least

$$1/2^m + \varepsilon - 1/2^m = \varepsilon.$$

This implies that $\log |V| = \mathrm{H}_\infty(\mathcal{V}) \leq k_1$, because otherwise it would contradict the fact that Ext is a good statistical extractor. ■

Finally, we will need the following lemma about obtaining predictors from distinguishers. The Boolean case ($m = 1$) is well known, and a proof for general m can be found in [12].

Lemma 2: For any source $\mathcal{Z}$ over $\{0,1\}^n$ and any function $b : \{0,1\}^n \to \{0,1\}^m$, if there is an ε-distinguisher D for the distributions $(\mathcal{Z}, b(\mathcal{Z}))$ and $(\mathcal{Z}, \mathcal{U}_m)$, then there is a predictor P with D as oracle which calls D once and runs in time $O(m)$ such that

$$\Pr_{z \in \mathcal{Z}}[P^D(z) = b(z)] \geq (1 + \varepsilon)/2^m.$$

III. An Impossibility Result

Just as in the statistical setting [7], we show that seedless extractors do not exist either in the computational setting. In fact, we show the impossibility result even for sources with a computational min-entropy as high as $n-2$.

Theorem 2: For any $n_1, n \in \mathbb{N}$ with $n_1 \geq 3n$ and for any function $\mathrm{Ext} : \{0,1\}^n \to \{0,1\}$, there exists a deterministic function $f : \{0,1\}^{n_1} \to \{0,1\}^n$ such that $\mathrm{H}_c(f(\mathcal{X})|\mathcal{X}) = n-2$ for $\mathcal{X} = \mathcal{U}_{n_1}$ but $\mathrm{Ext}(f(x))$ takes the same value for all x (so can be easily distinguished from random).

Proof: Consider any function $\mathrm{Ext} : \{0,1\}^n \to \{0,1\}$. Assume without loss of generality that $|\mathrm{Ext}^{-1}(1)| \geq 2^{n-1}$. Then we will show the existence of a function f such that $\mathrm{H}_c(f(\mathcal{X})|\mathcal{X}) = n-2$ but $\mathrm{Ext}(f(x)) = 1$ for all x. In fact, a standard argument can show that a random function is likely to work, as we will describe next.

Consider a random function $f : \{0,1\}^{n_1} \to \mathrm{Ext}^{-1}(1)$. Fix any $C : \{0,1\}^{n_1} \to \{0,1\}^n \in \mathsf{SIZE}(2^{n-2})$, and for each $x \in \{0,1\}^{n_1}$, define a binary random variable C_x such that $C_x = 1$ if and only if $C(x) = f(x)$. Observe that $\sum_x C_x$ is the number of x satisfying $C(x) = f(x)$. Note that

$$\begin{aligned}\mathop{\mathrm{E}}_f\left[\sum_x C_x\right] &= \sum_x \mathop{\mathrm{E}}_f[C_x] \\ &= \sum_x \Pr_f[C(x) = f(x)] \\ &\leq 2^{n_1-(n-1)},\end{aligned}$$

and let $\mu = 2^{n_1-(n-1)}$. Then by a Chernoff bound (see e.g., [1]), we have

$$\Pr_f\left[\sum_x C_x \geq 2\mu\right] \leq 2^{-\Omega(\mu)} = 2^{-\Omega(2^{n_1-n})}.$$

Since $|\mathsf{SIZE}(2^{n-2})| \leq 2^{O(n2^n)}$ and $n_1 \geq 3n$, a union bound gives

$$\begin{aligned}&\Pr_f\left[\exists C \in \mathsf{SIZE}(2^{n-2}) \text{ s.t. } \sum_x C_x \geq 2\mu\right] \\ &\leq 2^{O(n2^n)} \cdot 2^{-\Omega(2^{n_1-n})} \\ &< 1.\end{aligned}$$

Hence, there exists some f, such that $\Pr_x[C(x) = f(x)] < 2\mu \cdot 2^{-n_1} = 2^{-(n-2)}$ for any $C \in \mathsf{SIZE}(2^{n-2})$, but $\mathrm{Ext}(f(x)) = 1$ for any x. This completes the proof.

IV. Hybrid and Computational Extractors

In this section, we show that the function $\mathrm{Ext} : \mathbb{F}^\ell \times \mathbb{F}^\ell \to \mathbb{F}$ defined in Theorem 1 as

$$\mathrm{Ext}(v, w) = \langle v, w \rangle,$$

which is known to be a good statistical extractor, is also a good hybrid extractor and a good computational extractor.

Theorem 3: For any $k \geq \Omega(\log^2 n)$, any $m \leq O(\sqrt{k/\log k})$ dividing n, any $\varepsilon \geq 2^{-O(\sqrt{k/\log k})}$, any $s \leq 2^{n-k+O(k/\log k)}$, and for some $k_1 = n - k + O(k/\log k)$, the function $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ defined above is both a (k_1, k, ε, s)-hybrid-extractor and a (k_1, k, ε, s)-computational-extractor.

The proof for Theorem 3 relies on the following result, which gives an algorithm for the problem of learning linear functions under arbitrary distribution with adversarial noise.

Theorem 4: For any $k \geq \Omega(\log^2 n)$, any $m \leq O(k/\log k)$ dividing n, and any $\delta \geq 2^{-O(\sqrt{k/\log k})}$, there exists a learning algorithm A with the following property. Given any source $\mathcal{W}$ over $\{0,1\}^n = \mathbb{F}^\ell$ with $\mathrm{H}_\infty(\mathcal{W}) \geq k$ and any function $q : \mathbb{F}^\ell \to \mathbb{F}$, the algorithm A samples $2^{O(k/\log k)}$ training examples from the distribution $(\mathcal{W}, q(\mathcal{W}))$ and then runs in time $2^{n-k+O(k/\log k)}$ to output a list of size $2^{n-k+O(k/\log k)}$ which with probability $1 - o(1)$ contains every $v \in \mathbb{F}^\ell$ satisfying

$$\Pr_{w \in \mathcal{W}}[q(w) = \langle v, w \rangle] \geq 1/2^m + \delta.$$

Note that as in a standard learning-theoretical setting, we do not count the complexity of sampling the training examples (or just count each sampling as unit cost) in Theorem 4. We will prove the theorem in the next section, and now let us see how it is used to show Theorem 3.

Proof: (of Theorem 3)

First, we prove that the function Ext is a good hybrid extractor. Consider any source $(\mathcal{V}|\mathcal{X})$ with $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$ and any source $\mathcal{W}$, which is independent of $(\mathcal{V}|\mathcal{X})$, with $\mathrm{H}_\infty(\mathcal{W}) \geq k$. Assume for the sake of contradiction that there exists an ε-distinguisher $D \in \mathsf{SIZE}(s)$ for the distributions $(\mathcal{X}, \mathcal{W}, \langle \mathcal{V}, \mathcal{W} \rangle)$ and $(\mathcal{X}, \mathcal{W}, \mathcal{U}_m)$. By Lemma 2, this implies the existence of a predictor $Q \in \mathsf{SIZE}(s + O(m))$ with

$$\Pr_{x \in \mathcal{X}, v \in \mathcal{V}, w \in \mathcal{W}}[Q(x, w) = \langle v, w \rangle] \geq (1 + \varepsilon)/2^m.$$

Let $\delta = \varepsilon/2^{m+1} \geq 2^{-O(\sqrt{k/\log k})}$, and call any (x, v) heavy if

$$\Pr_{w \in \mathcal{W}}[Q(x, w) = \langle v, w \rangle] \geq 1/2^m + \delta.$$

Then a Markov inequality shows that

$$\Pr_{x \in \mathcal{X}, v \in \mathcal{V}}[(x, v) \text{is heavy}] \geq \delta.$$

Given any heavy (x, v), we want to predict v from x with a good probability. This can be reduced to the task of learning the linear function $\langle v, \cdot \rangle$, through noisy training examples $(w, q(w))$, with $q(w) = Q(x, w)$, under the distribution $w \in \mathcal{W}$. Consider the algorithm C which on input x calls the algorithm A in Theorem 4 using the function $q(\cdot) = Q(x, \cdot)$, and outputs a random element in the list produced by A. It samples $2^{O(k/\log k)}$ independent elements, denoted as W, from $\mathcal{W}$, makes $2^{O(k/\log k)}$ calls to Q, and for any heavy (x, v) it outputs v with probability $(1 - o(1)) \cdot 2^{-(n-k+O(k/\log k))}$. Then $\Pr_{x,v,W}[C(x) = v]$ is at least

$$\Pr_{x,v}[(x, v) \text{ is heavy}] \cdot \Pr_{x,v,W}[C(x) = v \mid (x, v) \text{ is heavy}]$$

which is at least

$$\delta \cdot (1 - o(1)) \cdot 2^{-(n-k+O(k/\log k))} \geq 2^{-(n-k+O(k/\log k))}.$$

That is, we have

$$\Pr_{x,v,W}[C(x)=v] \geq 2^{-(n-k+O(k/\log k))}.$$

We are almost done except that we still cannot bound the complexity of the algorithm C because it needs a way to sample elements from the source $\mathcal{W}$ which may not have an efficient sampling algorithm, unlike in the learning setting where one does not count the complexity of sampling. Fortunately, by an average argument, the bound above still holds for some fixed W, and we can simply hard-wire it into C. Similarly, we can do this for other random choices of C, and it is not hard to show that one can have a resulting circuit of size

$$|W|^{O(1)} + 2^{O(k/\log k)} \cdot (s + O(m)) + 2^{n-k+O(k/\log k)}$$

which is at most

$$2^{n-k+O(k/\log k)}.$$

Thus, for some large enough $k_1 = n-k+O(k/\log k)$, we have a circuit of size smaller than 2^{k_1} which can predict v correctly with probability at least

$$2^{-(n-k+O(k/\log k))} > 2^{-k_1}.$$

This contradicts the assumption that $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$, which means that the distinguisher D assumed at the beginning cannot exist, so EXT is a good hybrid extractor as claimed.

Next, we prove that EXT is also a good computational extractor, and the proof is almost identical. Consider two independent sources $(\mathcal{V}|\mathcal{X})$ and $(\mathcal{W}|\mathcal{Y})$, with $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$ and $\mathrm{H}_c(\mathcal{W}|\mathcal{Y}) \geq k$. Observe that the distribution of $\mathcal{W}$ must have statistical min-entropy at least k, because otherwise the predictor which always outputs the value with the largest measure can predict $\mathcal{W}$ correctly with probability larger than 2^{-k}, a violation of the assumption that $\mathrm{H}_c(\mathcal{W}|\mathcal{Y}) \geq k$. Then we can follow the proof above: assuming the existence of a distinguisher for EXT, we can obtain a predictor of size smaller than 2^{k_1}, with some $2^{O(k/\log k)}$ elements from $(\mathcal{W},\mathcal{Y})$ hard-wired in it, which can predict $\mathcal{V}$ correctly with probability larger than 2^{-k_1}. This contradicts the fact that $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$, so EXT is a good computational extractor. ∎

V. Learning Noisy Linear Functions

In this section, we prove Theorem 4. Recall that given any source $\mathcal{W}$ over $\{0,1\}^n = \mathbb{F}^\ell$ with $\mathrm{H}_\infty(\mathcal{W}) \geq k$, any $\delta \geq 2^{-O(\sqrt{k/\log k})}$, and any function $q : \mathbb{F}^\ell \to \mathbb{F}$, we would like to learn some unknown $v \in \mathbb{F}^\ell$ such that

$$\Pr_{w\in\mathcal{W}}[q(w) = \langle v, w\rangle] \geq 1/2^m + \delta. \tag{1}$$

Since such v may not be unique, we will list them all. Let us first imagine one such fixed v.

We start by randomly choosing $K = 2^{c(k/\log k)}$ independent training examples (with replacement) from the distribution $(\mathcal{W}, q(\mathcal{W}))$, for some large enough constant c (depending on δ). Let $W^{(0)}$ denote the $K \times \ell$ matrix and $q^{(0)}$ the K-dimensional vector, both over $\mathbb{F}$, such that for each training example $(w, q(w))$, $W^{(0)}$ has $w \in \mathbb{F}^\ell$ as a row and $q^{(0)}$ has $q(w) \in \mathbb{F}$ as an entry. Note that each training example $(w, q(w))$, with $w = (w_1, w_2, \ldots, w_\ell)$, gives us a linear equation

$$w_1 v_1 + w_2 v_2 + \cdots + w_\ell v_\ell = q(w)$$

for $v = (v_1, v_2, \ldots, v_\ell) \in \mathbb{F}^\ell$. Thus from these K training examples, we obtain a system of K linear equations, denoted as $[W^{(0)}|q^{(0)}]$, and we would like to reduce the task of learning v to that of solving this system of linear equations. However, this system is highly noisy as about $1 - 1/2^m$ fraction of the equations are likely to be wrong, according to (1). We will roughly follow the approach of Gaussian elimination (which works for noiseless systems of linear equations), but will make substantial changes in order to deal with our noisy case.

Our algorithm consists of two phases: the forward phase, shown in Fig. 1, and the backward phase, shown in Fig. 2. The forward phase works as follows, which is similar to an approach of Blum *et al.* [5]. Starting from the system $[W^{(0)}|q^{(0)}]$ of linear equations, we use several iterations to produce smaller and smaller systems with fewer and fewer variables, until we have a small enough system which we can afford to solve using brute force. More precisely, we choose the parameters

$$T = \log\sqrt{k/\log k} \text{ and } d = k/(mT),$$

divide each row of $W^{(0)}$ into ℓ/d blocks, with each block containing d elements in $\mathbb{F}$, and proceed in T iterations, as shown in Fig. 1. Note that after iteration t, we have the system $[W^{(t)}|q^{(t)}]$ which has $\ell - dt$ variables and $K^{(t)}$ equations, with

$$\begin{aligned} K^{(t)} &\geq K - t2^{md} \\ &= 2^{c(k/\log k)} - t2^{k/T} \\ &\geq 2^{c(k/\log k)}/2 \\ &= K/2, \end{aligned}$$

1) For t from 1 to T do
   a) Partition the equations of $[W^{(t-1)}|q^{(t-1)}]$ into at most 2^{md} groups (recall that $|\mathbb{F}| = 2^m$) according to their first blocks in $W^{(t)}$ (same block value in the same group).
   b) Within each group, randomly select an equation which we call pivot.
   c) Within each group, subtract the pivot from each equation.
   d) Remove the pivots and delete the first block from each equation. Let $[W^{(t)}|q^{(t)}]$ be the resulting system of equations.

Fig. 1. FORWARD PHASE.

1) Set $V^{(T)} = \mathbb{F}^{(n-k)/m}$, and set $V^{(t)} = \emptyset$ for $0 \leq t \leq T-1$.
2) For t from $T-1$ down to 0 do
   (a) For any $z \in \mathbb{F}^d \times V^{(t+1)}$ which is δ_t-good for $[W^{(t)}|q^{(t)}]$:
   if $|V^{(t)}| \leq L$
   then include z into $V^{(t)}$,
   else report "error" and halt.
3) Output $V^{(0)}$.

Fig. 2. BACKWARD PHASE.

for a large enough constant c. The key is to guarantee that the system still contains a good fraction of correct equations. Let

$$\delta_0 = \delta/2 \text{ and } \delta_t = (\delta_{t-1}/2)^2 \text{ for } t \geq 1.$$

A simple induction shows that for $t < T$,

$$\delta_t = \delta^{2^t}/2^{3\cdot 2^t - 2} \geq (\delta/8)^{2^t} \geq 2^{-0.1c(k/\log k)} = K^{-0.1},$$

for a large enough constant c. We say that any $z \in \mathbb{F}^{\ell - dt}$ is δ_t-*good* for the system $[W^{(t)}|q^{(t)}]$ if it satisfies at least $1/2^m + \delta_t$ fraction of equations in the system. Let $v^{(t)} \in \mathbb{F}^{\ell - dt}$ denote v without its first t blocks, and we call the forward phase *good* if for every t, $v^{(t)}$ is δ_t-good for $[W^{(t)}|q^{(t)}]$. Lemma 3 below, which will be proved in Section V-A, guarantees that the forward phase is good with a significant probability.

Lemma 3: The forward phase is good with probability at least $2^{-O(k/\log k)}$.

For the backward phase, we start from the last system $[W^{(T)}|q^{(T)}]$ produced by the forward phase, and work backward on larger and larger systems produced in the forward phase to obtain solutions for more and more variables. More precisely, we go from $t = T-1$ down to $t = 0$, and while in iteration t, we try to find all possible solutions which extend solutions from iteration $t+1$ and are δ_t-good for $[W^{(t)}|q^{(t)}]$, as shown in Fig. 2. However, in order to bound the running time, we will stop including the solutions once their number grows beyond the threshold

$$L = 2^{n-k+m+T+2\log(1/\delta_T)} = 2^{n-k+O(k/\log k)}.$$

If this happens, we may fail to include the actual solution v in our final list. Call the backward phase *good* if for every t, the number of such δ_t-good solutions for $[W^{(t)}|q^{(t)}]$ is at most L, or equivalently, it never reports "error." Lemma 4 below, which will be proved in Section V-B, guarantees that the backward phase is indeed good with a high probability.

Lemma 4: The backward phase is not good with probability at most $2^{-\Omega(k)}$.

From Lemma 3 and Lemma 4, the probability that both the forward and backward phases are good is at least

$$2^{-O(k/\log k)} - 2^{-\Omega(k)} = 2^{-O(k/\log k)}.$$

Assuming that both phases are good, a simple induction shows that $v^{(t)} \in V^{(t)}$ for any t and hence $v \in V^{(0)}$. Thus, we have shown that any fixed v satisfying the bound in (1) is contained in the list $V^{(0)}$ of size at most L with probability $2^{-O(k/\log k)}$. We can further reduce the probability of missing this v to $2^{-\omega(n)}$ by repeating the process $2^{O(k/\log k)}$ times, and take the union of the produced lists. Then a union bound shows that some v satisfying (1) is not included in the final output with probability only $o(1)$.

Finally, let us measure the complexity of our algorithm. First, $K \leq 2^{O(k/\log k)}$ training examples are sampled from the distribution $(\mathcal{W}, q(\mathcal{W}))$. Next, each iteration of the forward phase works on a system of at most K equations with at most n variables and runs in time $\text{poly}(K, n)$, and hence the whole forward phase runs in time

$$\begin{aligned} T \cdot \text{poly}(K,n) &= O(\log(k/\log k)) \cdot \text{poly}(K) \\ &\leq 2^{O(k/\log k)}, \end{aligned}$$

since $k \geq \Omega(\log^2 n)$. Then, each iteration of the backward phase runs in time

$$\begin{aligned} &O(2^{md} \cdot L \cdot K) \\ &\leq 2^{O(k/\log k)} \cdot 2^{n-k+O(k/\log k)} \cdot 2^{O(k/\log k)} \\ &\leq 2^{n-k+O(k/\log k)}, \end{aligned}$$

so the whole backward phase runs in time

$$O(\log(k/\log k)) \cdot 2^{n-k+O(k/\log k)} \leq 2^{n-k+O(k/\log k)}.$$

Finally, the process is repeated for $2^{O(k/\log k)}$ times, and thus the total running time is

$$\begin{aligned} &2^{O(k/\log k)} \cdot \left(2^{O(k/\log k)} + 2^{n-k+O(k/\log k)}\right) \\ \leq &2^{n-k+O(k/\log k)}. \end{aligned}$$

As a result, we have Theorem 4. To complete the proof, it remains to prove Lemma 3 and Lemma 4, which we do next.

A. Proof of Lemma 3

First, by a Chernoff bound, we know that $v = v^{(0)}$ satisfies less than $1/2^m + \delta_0$ fraction of equations in $[W^{(0)}|q^{(0)}]$ with probability at most $2^{-\Omega(\delta_0^2 K)} = o(1)$. That is, $v^{(0)}$ is δ_0-good for $[W^{(0)}|q^{(0)}]$ with probability $1 - o(1)$. Next, we need the following lemma.

Lemma 5: In the forward phase, if $v^{(t-1)}$ is δ_{t-1}-good for $[W^{(t-1)}|q^{(t-1)}]$, then $v^{(t)}$ is δ_t-good for $[W^{(t)}|q^{(t)}]$ with probability at least δ_t.

Proof: Let $\tau = \delta_{t-1}$. Assume that $v^{(t-1)}$ is τ-good, so it satisfies at least $\frac{1}{2^m} + \tau$ fraction of equations in the system $[W^{(t-1)}|q^{(t-1)}]$. Partition equations in the system $[W^{(t-1)}|q^{(t-1)}]$ into groups according to their first blocks, as in Step 1(a) of the forward phase. Suppose group i contains p_i fraction of equations in $[W^{(t-1)}|q^{(t-1)}]$ and $v^{(t-1)}$ satisfies $\frac{1}{2^m} + \tau_i$ fraction of equations in the group, for some $\tau_i \in [-\frac{1}{2^m}, 1 - \frac{1}{2^m}]$. Then we have

$$\sum_i p_i \cdot \left(\frac{1}{2^m} + \tau_i\right) \geq \frac{1}{2^m} + \tau. \tag{2}$$

We would like to count the expected fraction of new equations satisfied by $v^{(t-1)}$, where we count equations in their multiplicity. Before doing that, let us first count the fraction with respect to the system obtained before Step 1(d) (before removing pivots). Let us denote a generic equation of the system $[W^{(t-1)}|q^{(t-1)}]$ by $(w^{(t-1)}|q^{(t-1)})$. Consider any group i. For

$u \in \mathbb{F}$, let α_u denote the fraction of equations $(w^{(t-1)}|q^{(t-1)})$ in the group which are off by a value u in the sense that

$$q^{(t-1)} = \langle v^{(t-1)}, w^{(t-1)} \rangle + u.$$

Note that for $v^{(t-1)}$ to satisfy a new equation, which is the difference between two equations, these two involved equations must be off by the same value. Therefore, the expected fraction of new satisfied equations in this group is $\sum_u \alpha_u^2$, which under the constraint $\alpha_0 = \frac{1}{2^m} + \tau_i$ achieves its minimum when $\alpha_u = \frac{1}{2^m} - \frac{\tau_i}{2^m - 1}$ for all other $u \neq 0$. Hence, after one iteration, the expected fraction of new equations in group i (before removing pivots) satisfied by $v^{(t-1)}$ is at least

$$\begin{aligned}
&\left(\frac{1}{2^m} + \tau_i\right)^2 + (2^m - 1) \cdot \left(\frac{1}{2^m} - \frac{\tau_i}{2^m - 1}\right)^2 \\
&= 2^m \cdot \left(\frac{1}{2^m}\right)^2 + \frac{2-2}{2^m} \cdot \tau_i + \frac{2^m - 1 + 1}{2^m - 1} \cdot \tau_i^2 \\
&\geq \frac{1}{2^m} + \tau_i^2.
\end{aligned}$$

Combing all groups together, the expected fraction of satisfied equations overall (before removing the pivots) is at least

$$\begin{aligned}
\sum_i p_i \left(\frac{1}{2^m} + \tau_i^2\right) &= \frac{1}{2^m} + \sum_i p_i \tau_i^2 \\
&\geq \frac{1}{2^m} + \left(\sum_i p_i \tau_i\right)^2 \\
&\geq \frac{1}{2^m} + \tau^2,
\end{aligned}$$

where the first inequality is due to Jensen inequality, and the second inequality uses the bound $\sum_i p_i \tau_i \geq \tau$ implied by that in (2).

To get the expected fraction of satisfied equations in the final system $[W^{(t)}|q^{(t)}]$, after performing Step 1(d), observe that we only need to discard at most $2^{md} = 2^{O(k/\log k)}$ equations, each with measure $\frac{1}{K^{(t)}} \leq \frac{2}{K}$, so the total discarded measure, denoted as μ, is at most

$$2^{md} \cdot \frac{2}{K} \leq 2^{O(k/\log k)} \cdot 2 \cdot 2^{-c(k/\log k)} \leq \frac{\tau^2}{2},$$

for a large enough constant c. As a result, the expected fraction of equations in $[W^{(t)}|q^{(t)}]$ satisfied by $v^{(t)}$ is at least

$$\begin{aligned}
\frac{1}{1-\mu} \cdot \left(\frac{1}{2^m} + \tau^2 - \mu\right) &\geq \frac{1}{2^m} + \tau^2 - \mu \\
&\geq \frac{1}{2^m} + \frac{\tau^2}{2} \\
&= \frac{1}{2^m} + 2\delta_t,
\end{aligned}$$

by recalling that $\tau = \delta_{t-1}$ and $\delta_t = (\delta_{t-1}/2)^2$. Finally, by a Markov inequality, we have the lemma. ∎

Then by Lemma 5 and an induction, the forward phase is good with probability at least

$$\begin{aligned}
(1 - o(1)) \prod_{t=1}^{T} \delta_t &\geq (1 - o(1)) \prod_{t=1}^{T} (\delta/8)^{2^t} \\
&\geq (1 - o(1))(\delta/8)^{2^{T+1}} \\
&\geq 2^{-O(k/\log k)}.
\end{aligned}$$

This proves Lemma 3.

B. Proof of Lemma 4

Recall that a solution is δ_t-good for the system $[W^{(t)}|q^{(t)}]$ if it satisfies at least $1/2^m + \delta_t$ fraction of the equations. For any t such that $0 \leq t \leq T-1$, consider the following event:

- $B^{(t)}$: the number of δ_t-good solutions for $[W^{(t)}|q^{(t)}]$ exceeds L.

Thus, our goal is to show that

$$\Pr\left[\bigvee_{t=0}^{T-1} B^{(t)}\right] \leq 2^{-\Omega(k)}.$$

We will prove this by a union bound, so our goal is reduced to bounding each $\Pr[B^{(t)}]$ for $0 \leq t \leq T-1$.

To get a quick idea, let us first consider how to bound $\Pr[B^{(0)}]$. Note that since Ext is a good *statistical* extractor and $\mathcal{W}$ has a high min-entropy, Lemma 1 guarantees that the number of z satisfying the probability bound $\Pr_{w \in \mathcal{W}}[q(w) = \langle z, w \rangle] \geq 1/2^m + \delta_0/2$ is at most L. Any other z is very unlike to be δ_0-good for $[W^{(0)}|q^{(0)}]$ by a Chernoff bound because each row of $W^{(0)}$ is sampled independently from $\mathcal{W}$. Since $B^{(0)}$ happens only when any such z (not satisfying that probability bound) is δ_0-good, a union bound shows that $\Pr[B^{(0)}]$ is indeed small.

Now for $t \geq 1$, to follow this idea to bound $\Pr[B^{(t)}]$, we would also like the distribution of $W^{(t)}$, denoted as $\mathcal{S}^{(t)}$, to have the nice property that each of its rows comes independently from a high min-entropy source. Unfortunately, this is not true in general,[2] and a much more involved analysis is needed. Our approach is to consider the distribution $\mathcal{S}^{(t)}$ conditioned on the choice of pivots in the first t iterations. We call a particular choice of the pivots a *restriction* of the pivots, which includes fixing the indices and the values of some rows as pivots while leaving other rows free. We will show that the distribution $\mathcal{S}^{(t)}$ conditioned on most restrictions is close to a distribution with the nice property. For our purpose here, instead of using the standard definition of "closeness" (which would be measured according to the statistical distance), we consider the following one.

Definition 4: We say that two distributions are γ-close if the probabilities of any event according to the two distributions are within a multiplicative factor of γ from each other.

Observe that one can generate the matrix $W^{(t)}$ in an alternative way by first choosing the pivots in t iterations and then generating the matrices $W^{(1)}, \ldots, W^{(t)}$ consistent with the pivots. Formally, the distribution $\mathcal{S}^{(t)}$ (the distribution of the matrix $W^{(t)}$) can be generated in two passes as follows. In the first pass, we select a restriction of pivots in the first t iterations, denoted as $R^{(1)}, \ldots, R^{(t)}$, by running the forward phase on the matrix $W^{(0)}$ sampled from $\mathcal{W}$ and collecting the pivots, which include the indices and the values of rows as pivots, in each iteration. In

[2]This is true in the simple case considered by [5] that one has $\mathcal{W} = \mathcal{U}_n$ to start with. In this case, for each t, one can easily show that each row of $W^{(t)}$ does come independently from the uniform distribution $\mathcal{U}_{n-tmd}$.

the second pass, we again sample a matrix $W^{(0)}$ from $\mathcal{W}$ and then run the forward phase accordingly for t iterations to derive the matrix $W^{(t)}$, under the condition, denoted as $I_{R^{(1)},\ldots,R^{(t)}}$, that the pivots selected in the t iterations match $R^{(1)},\ldots,R^{(t)}$. Let $\tilde{\mathcal{D}}^{(t)} = (\mathcal{S}^{(t)}|I_{R^{(1)},\ldots,R^{(t)}})$ denote such a conditional distribution of $W^{(t)}$ with respect to the restriction $R^{(1)},\ldots,R^{(t)}$. Now consider the following event about $\tilde{\mathcal{D}}^{(t)}$, over the distribution of $R^{(1)},\ldots,R^{(t)}$ selected in the first pass.

- $E^{(t)}$: the distribution $\tilde{\mathcal{D}}^{(t)}$ is γ_t-close to some distribution $\mathcal{D}^{(t)}$ which has $K^{(t)}$ rows, each coming independently from a distribution $\mathcal{W}^{(t)}$ with $\mathrm{H}_\infty(\mathcal{W}^{(t)}) \geq k - t(md+1)$, for some $\gamma_t \leq K^{2^{md}(2^t-1)} \leq 2^{\sqrt{K}}$.

The following lemma, which will be proved later, shows that when conditioned on $E^{(t)}$, the probability of $B^{(t)}$ is indeed small.

Lemma 6: For any t such that $0 \leq t \leq T-1$,

$$\Pr[B^{(t)} \mid E^{(t)}] \leq 2^{-\Omega(k)}.$$

Next, we would like to show that $E^{(t)}$ happens with high probability. Note that for $t = 0$, the event $E^{(0)}$ always happens because the initial distribution $\tilde{\mathcal{D}}^{(0)}$ has the nice property itself, so we have $\mathcal{D}^{(0)} = \tilde{\mathcal{D}}^{(0)}$ and $\gamma_0 = 1$. For $1 \leq t \leq T-1$, we use induction to show that

$$\begin{aligned}\Pr\left[\neg E^{(t)}\right] &\leq \Pr\left[\neg E^{(t)} \mid E^{(t-1)}\right] + \Pr\left[\neg E^{(t-1)}\right] \\ &\leq \sum_{\tau=1}^{t} \Pr\left[\neg E^{(\tau)} \mid E^{(\tau-1)}\right],\end{aligned}$$

and then we rely on the following lemma, which will be proved later.

Lemma 7: For any t such that $1 \leq t \leq T-1$,

$$\Pr\left[\neg E^{(t)} \mid E^{(t-1)}\right] \leq 2^{-\Omega(K)}.$$

From these two lemmas, we have that for any t such that $1 \leq t \leq T-1$,

$$\begin{aligned}&\Pr\left[B^{(t)}\right] \\ &\leq \Pr\left[B^{(t)} \mid E^{(t)}\right] + \Pr\left[\neg E^{(t)}\right] \\ &\leq \Pr\left[B^{(t)} \mid E^{(t)}\right] + \sum_{\tau=1}^{t} \Pr\left[\neg E^{(\tau)} \mid E^{(\tau-1)}\right] \\ &\leq 2^{-\Omega(k)}.\end{aligned}$$

For $t = 0$, we have

$$\Pr\left[B^{(0)}\right] = \Pr\left[B^{(0)} \mid E^{(0)}\right] \leq 2^{-\Omega(k)}.$$

As a result, a union bound gives us

$$\Pr\left[\bigvee_{t=0}^{T-1} B^{(t)}\right] \leq \sum_{t=0}^{T-1} \Pr\left[B^{(t)}\right] \leq T \cdot 2^{-\Omega(k)} = 2^{-\Omega(k)},$$

which proves Lemma 4. Thus, it remains to prove Lemma 6 and Lemma 7, which we do in the next two subsections.

C. Proof of Lemma 6

Let us first count the number of solutions z such that

$$\Pr_{w \in \mathcal{W}^{(t)}}\left[q^{(t)}(w) = \langle z, w\rangle\right] \geq 1/2^m + \delta_t/2.$$

Let Z denote the set of such z's. Note that $\mathcal{W}^{(t)}$ is a source over $\mathbb{F}^{\ell-td} = \{0,1\}^{(\ell-td)m}$ with $\mathrm{H}_\infty(\mathcal{W}^{(t)}) \geq k - t(md+1)$. Thus by Theorem 1 and Lemma 1, we have

$$\begin{aligned}|Z| &\leq 2^{(\ell-td)m+m+2\log(2/\delta_t)-2-(k-t(md+1))} \\ &= 2^{n-k+m+t+2\log(1/\delta_t)} \\ &\leq L.\end{aligned}$$

This means that for the event $B^{(t)}$ to happen, some $z \notin Z$ must be δ_t-good.

Consider any restriction $R^{(1)},\ldots,R^{(t)}$ such that the event $E^{(t)}$ happens. If we sample the matrix $W^{(t)}$ according to the distribution $\mathcal{D}^{(t)}$, which has each row coming independently from $\mathcal{W}^{(t)}$, then any fixed $z \notin Z$ is δ_t-good (satisfying at least $1/2^m + \delta_t$ fraction of equations in $[W^{(t)}|q^{(t)}]$) with probability at most $2^{-\Omega(\delta_t^2 K^{(t)})}$ by a Chernoff bound, and a union bound shows that

$$\begin{aligned}\Pr_{\mathcal{D}^{(t)}}\left[B^{(t)}\right] &\leq \Pr_{\mathcal{D}^{(t)}}\left[\exists z \notin Z : z \text{ is } \delta_t\text{-good}\right] \\ &\leq 2^n \cdot 2^{-\Omega(\delta_t^2 K^{(t)})} \\ &\leq 2^{-\Omega(K^{0.8})}.\end{aligned}$$

Now if we sample $W^{(t)}$ according to the distribution $\tilde{\mathcal{D}}^{(t)} = (\mathcal{S}^{(t)}|I_{R^{(1)},\ldots,R^{(t)}})$, which is γ_t-close to $\mathcal{D}^{(t)}$ (given that $E^{(t)}$ happens), the probability is only scaled up by a factor γ_t. Thus, we have

$$\begin{aligned}\Pr_{\tilde{\mathcal{D}}^{(t)}}\left[B^{(t)}\right] &\leq \gamma_t \cdot 2^{-\Omega(K^{0.8})} \\ &\leq 2^{\sqrt{K}} \cdot 2^{-\Omega(K^{0.8})} \\ &\leq 2^{-\Omega(k)}.\end{aligned}$$

Since the bound holds for any restriction $R^{(1)},\ldots,R^{(t)}$ such that the event $E^{(t)}$ happens, we have the lemma.

D. Proof of Lemma 7

Let us consider any restriction $R^{(1)},\ldots,R^{(t-1)}$ such that the event $E^{(t-1)}$ happens, and we will show that $E^{(t)}$ happens with high probability, over the selection of $R^{(t)}$. More precisely, the assumption that $E^{(t-1)}$ happens means that we start iteration t from the distribution $\tilde{\mathcal{D}}^{(t-1)}$ which is close to some nice distribution $\mathcal{D}^{(t-1)}$, and our task is to show that with high probability over the selection of $R^{(t)}$, the resulting conditional distribution $\tilde{\mathcal{D}}^{(t)}$ after iteration t is close to another nice distribution $\mathcal{D}^{(t)}$, so that $E^{(t)}$ happens. For this, we need to figure out which of these $R^{(t)}$'s make $E^{(t)}$ happen.

Note that for a restriction $R^{(t)}$, the corresponding distribution $\tilde{\mathcal{D}}^{(t)}$ is obtained by applying Steps 1(c) and 1(d) on the matrix $W^{(t-1)}$ sampled from $\tilde{\mathcal{D}}^{(t-1)}$ under the condition that it is consistent with $R^{(t)}$. The restriction $R^{(t)}$ fixes some $r \leq 2^{md}$ rows

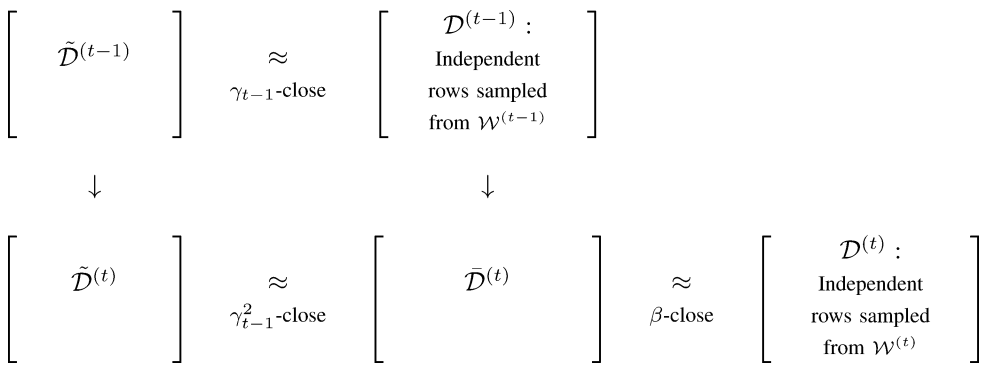

Fig. 3. If $\bar{\mathcal{D}}^{(t-1)}$ is close to $\mathcal{D}^{(t-1)}$, then $\bar{\mathcal{D}}^{(t)}$ is close to $\mathcal{D}^{(t)}$, conditioned on $I_{R^{(t)}}$.

of the matrix $W^{(t-1)}$ as pivots and it has the effect on the distribution $\tilde{\mathcal{D}}^{(t-1)}$ that all the rows of $W^{(t-1)}$ must belong to the r groups of those r rows. We would like the effect to be small, and we consider the following event, over the selection of $R^{(t)}$.

- $G^{(t)}$: those elements in the support of $\mathcal{W}^{(t-1)}$ which would belong to those r groups of $R^{(t)}$ when selected as rows of $W^{(t-1)}$ (i.e., those with their first blocks matching one of the first blocks of the r rows in $R^{(t)}$) have a combined measure of $\rho \geq 1/2$ in the distribution $\mathcal{W}^{(t-1)}$.

We will show that if $G^{(t)}$ happens then $E^{(t)}$ happens. For this, let us consider any fixed restriction $R^{(t)}$ such that $G^{(t)}$ happens, and let us use $I_{R^{(t)}}$ to denote the event that the pivots chosen in iteration t match those in $R^{(t)}$. Our approach is illustrated in Fig. 3.

First, let us consider the case of starting iteration t from the nice distribution $\mathcal{D}^{(t-1)}$, instead of $\tilde{\mathcal{D}}^{(t-1)}$, conditioned on $I_{R^{(t)}}$, and let $\bar{\mathcal{D}}^{(t)}$ be the resulting distribution after iteration t. The following claim shows that $\bar{\mathcal{D}}^{(t)}$ is in fact close to a nice distribution.

Claim 1: For some $\beta \leq K^{2^{md}}$, the distribution $\bar{\mathcal{D}}^{(t)}$ is β-close to some nice distribution $\mathcal{D}^{(t)}$ described in the event $E^{(t)}$ (i.e., $\bar{\mathcal{D}}^{(t)}$ has $K^{(t)}$ rows, each coming independently from a distribution $\mathcal{W}^{(t)}$ with $\mathrm{H}_\infty(\mathcal{W}^{(t)}) \geq k - t(md+1)$).

Next, let us go back to the actual situation of starting iteration t from the distribution $\tilde{\mathcal{D}}^{(t-1)}$, instead of $\mathcal{D}^{(t-1)}$ as we did in the above claim. Using the assumption that $\tilde{\mathcal{D}}^{(t-1)}$ is close to $\mathcal{D}^{(t-1)}$, our next claim shows that when we start iteration t from the distribution $\tilde{\mathcal{D}}^{(t-1)}$ conditioned on $I_{R^{(t)}}$, the resulting distribution $\tilde{\mathcal{D}}^{(t)}$ is close to the distribution $\bar{\mathcal{D}}^{(t)}$.

Claim 2: The distribution $\tilde{\mathcal{D}}^{(t)}$ is γ_{t-1}^2-close to the distribution $\bar{\mathcal{D}}^{(t)}$.

From these two claims, we can conclude that $\tilde{\mathcal{D}}^{(t)}$ is γ_t-close to $\mathcal{D}^{(t)}$, for $\gamma_t = \gamma_{t-1}^2 \beta \leq \gamma_{t-1}^2 K^{2^{md}}$, which by induction is at most

$$K^{2^{md}(2^t-2)} K^{2^{md}} \leq K^{2^{md}(2^t-1)} \leq 2^{\sqrt{K}}.$$

This implies that for any restriction $R^{(t)}$ such that the event $G^{(t)}$ happens, the event $E^{(t)}$ must happen as well. Therefore, the probability that $E^{(t)}$ does not happen is at most the probability that $G^{(t)}$ does not happen, which we bound by the following claim.

Claim 3: The probability over the selection of $R^{(t)}$ that $G^{(t)}$ does not happen is at most $2^{-\Omega(K)}$.

We have shown that for any restriction $R^{(1)}, \ldots, R^{(t-1)}$ such that the event $E^{(t-1)}$ happens, the probability, over the selection of $R^{(t)}$, that the event $E^{(t)}$ does not happen is at most $2^{-\Omega(K)}$. This implies that $\Pr[\neg E^{(t)} \mid E^{(t-1)}] \leq 2^{-\Omega(K)}$, which proves Lemma 7. Thus, it remains to prove the three claims above, which we do next.

Proof: (of Claim 1)

Recall that we have fixed a restriction $R^{(t)}$ which fixes some r rows as pivots such that the event $G^{(t)}$ happens, and we use $I_{R^{(t)}}$ to denote the event that the pivots selected during iteration t match those in the restriction $R^{(t)}$. In this claim, we consider the situation of starting iteration t from the nice distribution $\mathcal{D}^{(t-1)}$ conditioned on the event $I_{R^{(t)}}$.

First, let us see how the distribution $\mathcal{D}^{(t-1)}$ is affected by the conditioning on $I_{R^{(t)}}$. Consider any fixed matrix M of $K^{(t)} = K^{(t-1)} - r$ rows, insert the rows of $R^{(t)}$ at the proper places to get a fixed matrix $W^{(t-1)}$ of $K^{(t-1)}$ rows, and let us use $I_{W^{(t-1)}}$ to denote the event that a randomly sampled matrix from $\mathcal{D}^{(t-1)}$ equals this matrix $W^{(t-1)}$. If the matrix has a row not in the r groups of $R^{(t)}$, then $\Pr_{\mathcal{D}^{(t-1)}}[I_{W^{(t-1)}} \mid I_{R^{(t)}}] = 0$. Otherwise, $\Pr_{\mathcal{D}^{(t-1)}}[I_{W^{(t-1)}} \mid I_{R^{(t)}}]$ is

$$\frac{\left(\prod_{j=1}^{K^{(t)}} \mathcal{W}^{(t-1)}(M_j)\right) \cdot \left(\prod_{i=1}^{r} \frac{1}{\ell_i'+1}\right)}{\sum_{\ell_1+\cdots+\ell_r=K^{(t)};\ell_i\geq 0} \binom{K^{(t)}}{\ell_1,\cdots,\ell_r} \cdot \left(\prod_{i=1}^{r} \rho_i^{\ell_i}\right) \cdot \left(\prod_{i=1}^{r} \frac{1}{\ell_i+1}\right)},$$

where $\mathcal{W}^{(t-1)}(M_j)$ is the measure of the j'th row of M in $\mathcal{W}^{(t-1)}$, ℓ_i' is the number of rows of M in group i, and ρ_i is the measure of group i in $\mathcal{W}^{(t-1)}$. Note that for some $\alpha_1, \alpha_2 \in [K^{-r}, 1]$, the numerator equals

$$\left(\prod_{j=1}^{K^{(t)}} \mathcal{W}^{(t-1)}(M_j)\right) \cdot \alpha_1,$$

while the denominator equals

$$\begin{aligned}
&\sum_{\ell_1+\cdots+\ell_r=K^{(t)};\ell_i\geq 0} \binom{K^{(t)}}{\ell_1,\cdots,\ell_r} \cdot \left(\prod_{i=1}^{r} \rho_i^{\ell_i}\right) \cdot \alpha_2 \\
&= \left(\sum_{i=1}^{r} \rho_i\right)^{K^{(t)}} \cdot \alpha_2 \\
&= \rho^{K^{(t)}} \cdot \alpha_2,
\end{aligned}$$

where $\sum_{i=1}^{r} \rho_i = \rho \geq 1/2$ as we assume that the event $G^{(t)}$ happens. As a result, for $\beta = \frac{\alpha_1}{\alpha_2} \in [K^{-r}, K^r]$, we have

$$\Pr_{\mathcal{D}^{(t-1)}}[I_{W^{(t-1)}} \mid I_{R^{(t)}}] = \left(\prod_{j=1}^{K^{(t)}} \frac{\mathcal{W}^{(t-1)}(M_j)}{\rho} \right) \cdot \beta.$$

Note that the first factor above can be seen as the probability when we sample each row of the matrix independently according a new distribution $\tilde{\mathcal{W}}^{(t-1)}$, which is the distribution $\mathcal{W}^{(t-1)}$ restricted to those r groups of $R^{(t)}$ and normalized by their measure ρ. Thus, although the conditioning on the event $I_{R^{(t)}}$ may destroy the independence so that we can no longer see each row as coming independently from $\mathcal{W}^{(t-1)}$, we can somehow have the independence restored by considering another distribution $\tilde{\mathcal{W}}^{(t-1)}$ with some distortion factor β. More precisely, we have shown that the distribution $\mathcal{D}^{(t-1)}$ conditioned on the event $I_{R^{(t)}}$ is β-close to a nice distribution, denoted as $\hat{D}^{(t-1)}$, which has each of its remaining row (not fixed by $R^{(t)}$) coming independently from $\tilde{\mathcal{W}}^{(t-1)}$, with

$$\begin{aligned} \mathrm{H}_\infty(\tilde{\mathcal{W}}^{(t-1)}) &\geq \mathrm{H}_\infty(\mathcal{W}^{(t-1)}) - \log(1/\rho) \\ &\geq k - (t-1)(md+1) - 1. \end{aligned}$$

Next, let us see what the resulting distribution $\bar{\mathcal{D}}^{(t)}$ will be when Steps 1(c) and 1(d) are performed on the distribution $\mathcal{D}^{(t-1)}$ conditioned on $I_{R^{(t)}}$. Again, we first consider the case of applying the two steps on the nice distribution $\hat{\mathcal{D}}^{(t-1)}$ instead. When we perform Step 1(c) to subtract from each row its corresponding pivot, which is a fixed value, each resulting row still remains independent from others. However, the distribution of each resulting row is now changed to another distribution which may have a smaller min-entropy than that of $\tilde{\mathcal{W}}^{(t-1)}$, because different initial rows after subtracting their corresponding pivots may result in the same value. Still, the number of such initial rows can be at most 2^{md} since no two such rows can come from the same group, which implies that the min-entropy only decreases by at most md. Then after performing Step 1(d) to remove the pivots and delete the first blocks, the resulting matrix has each row coming independently from some distribution $\mathcal{W}^{(t)}$ with min-entropy at least

$$\mathrm{H}_\infty(\tilde{\mathcal{W}}^{(t-1)}) - md \geq k - t(md+1).$$

That is, after performing Steps 1(c) and 1(d) on the distribution $\hat{\mathcal{D}}^{(t-1)}$, the resulting distribution, denoted as $\mathcal{D}^{(t)}$, satisfies the condition in event $E^{(t)}$. Finally, let us get back to the actual case of starting with the distribution $\mathcal{D}^{(t-1)}$ conditioned on $I_{R^{(t)}}$. Since it is β-close to $\hat{\mathcal{D}}^{(t-1)}$, the resulting distribution $\bar{\mathcal{D}}^{(t)}$ after applying the two steps is β-close to the corresponding resulting distribution $\mathcal{D}^{(t)}$, which proves the claim. ∎

Proof: (of Claim 2)

In this claim, we go back to the actual situation of starting iteration t from the distribution $\tilde{\mathcal{D}}^{(t-1)}$, instead of $\mathcal{D}^{(t-1)}$ as we just did. We would like to show that the resulting distribution $\tilde{\mathcal{D}}^{(t)}$ when starting from $\tilde{\mathcal{D}}^{(t-1)}$ is γ_{t-1}^2-close to the distribution $\bar{\mathcal{D}}^{(t)}$ when starting from $\mathcal{D}^{(t-1)}$. For this, it suffices to show that for any event A, the probabilities of $\Pr_{\mathcal{D}^{(t-1)}}[A \mid I_{R^{(t)}}]$ and $\Pr_{\tilde{\mathcal{D}}^{(t-1)}}[A \mid I_{R^{(t)}}]$ are within a multiplicative factor of γ_{t-1}^2. This is true because from the fact that $\mathcal{D}^{(t-1)}$ and $\tilde{\mathcal{D}}^{(t-1)}$ are γ_{t-1}-close, we know that $\Pr_{\mathcal{D}^{(t-1)}}[I_{R^{(t)}}]$ and $\Pr_{\tilde{\mathcal{D}}^{(t-1)}}[I_{R^{(t)}}]$ are within a multiplicative factor of γ_{t-1}, and so are $\Pr_{\mathcal{D}^{(t-1)}}[A \wedge I_{R^{(t)}}]$ and $\Pr_{\tilde{\mathcal{D}}^{(t-1)}}[A \wedge I_{R^{(t)}}]$. ∎

Proof: (of Claim 3)

Note that the restriction $R^{(t)}$ can be selected by sampling a matrix $W^{(t-1)}$ according to the distribution $\tilde{\mathcal{D}}^{(t-1)}$ and then applying Steps 1(a) and 1(b) to select the pivots. Thus, the probability that $G^{(t)}$ does not happen is at most the probability that all the $K^{(t-1)}$ rows of $W^{(t-1)}$ lie in some r groups with a combined measure of $\rho \leq 1/2$ in the distribution $\mathcal{W}^{(t-1)}$.

Again, let us first consider the case of sampling $W^{(t-1)}$ according to the distribution $\mathcal{D}^{(t-1)}$, instead of $\tilde{\mathcal{D}}^{(t-1)}$. Note that there are at most $2^{2^{md}}$ ways of choosing the r groups with a combined measure of $\rho \leq 1/2$ in $\mathcal{W}^{(t-1)}$, and the probability that all the $K^{(t-1)} \geq K/2$ independent rows lie in any particular choice of such r groups is at most $(1/2)^{K/2}$. Then a union bound shows that the probability of having $\rho \leq 1/2$ is at most

$$2^{2^{md}} \cdot (1/2)^{K/2} \leq 2^{-\Omega(K)}.$$

Next, let us go back to actual case of sampling $W^{(t-1)}$ according to the distribution $\tilde{\mathcal{D}}^{(t-1)}$. Note that the probability of having $\rho \leq 1/2$ according to $\tilde{\mathcal{D}}^{(t-1)}$ can only be larger than that according to $\mathcal{D}^{(t-1)}$ by at most a factor of γ_{t-1}, and hence it is still at most

$$\gamma_{t-1} \cdot 2^{-\Omega(K)} \leq 2^{\sqrt{K}} \cdot 2^{-\Omega(K)} \leq 2^{-\Omega(K)}.$$

∎

REFERENCES

[1] N. Alon and J. Spencer, *The Probabilistic Method*. : John Wiley, 1992.

[2] B. Barak, R. Impagliazzo, and A. Wigderson, "Extracting randomness using few independent sources," *SIAM J. Comput.*, vol. 36, no. 4, pp. 1095–1118, 2006.

[3] B. Barak, G. Kindler, R. Shaltiel, B. Sudakov, and A. Wigderson, "Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors," in *Proc. 37th Annu. ACM Symp. on Theory of Computing (STOC'05)*, 2005, pp. 1–10.

[4] B. Barak, R. Shaltiel, and A. Wigderson, "Computational analogues of entropy," in *Proc. 7th Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM'03)*, 2003, pp. 200–215.

[5] A. Blum, A. Kalai, and H. Wasserman, "Noise-tolerant learning, the parity problem, and the statistical query model," *J. ACM*, vol. 50, no. 4, pp. 506–519, 2003.

[6] J. Bourgain, "More on the sum-product phenomenon in prime fields and its applications," *Int. J. Numb. Theory*, vol. 1, no. 1, pp. 1–32, 2005.

[7] B. Chor and O. Goldreich, "Unbiased bits from sources of weak randomness and probabilistic communication complexity," *SIAM J. Comput.*, vol. 17, no. 2, pp. 230–261, Apr. 1988.

[8] B. Chor, O. Goldreich, J. Håstad, J. Friedman, S. Rudich, and R. Smolensky, "The bit extraction problem of t-resilient functions," in *Proc. 26th Annu. IEEE Symp. Found. Comput. Sci. (FOCS'85)*, pp. 396–407.

[9] V. Feldman, P. Gopalan, S. Khot, and A. Ponnuswami, "On agnostic learning of parities, monomials, and halfspaces," *SIAM J. Comput.*, vol. 39, no. 2, pp. 606–645, 2009.

[10] A. Gabizon, R. Raz, and R. Shaltiel, "Deterministic extractors for bit-fixing sources by obtaining an independent seed," *SIAM J. Comput.*, vol. 36, no. 4, pp. 1072–1094, 2006.

[11] O. Goldreich and L. A. Levin, "A hard-core predicate for all one-way functions," in *Proc. 21st Annu. ACM Symp. Theory Comput. (STOC'89)*, pp. 25–32.

[12] O. Goldreich, R. Rubinfeld, and M. Sudan, "Learning polynomials with queries: The highly noisy case," *SIAM J. Discrete Math.*, vol. 13, no. 4, pp. 535–570, 2000.
[13] V. Guruswami, C. Umans, and S. Vadhan, "Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes," *J. ACM*, vol. 56, no. 4, 2009, Art. 20.
[14] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A pseudorandom generator from any one-way function," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1364–1396, 1999.
[15] C.-Y. Hsiao, C.-J. Lu, and L. Reyzin, "Conditional computational entropy, or toward separating pseudoentropy from compressibility," in *Proc. Adv. Cryptol.—EUROCRYPT*, 2007, pp. 169–186.
[16] A. Kalai, Y. Mansour, and E. Verbin, "On agnostic boosting and parity learning," in *Proc. 40th Annu. ACM Symp. Theory Comput. (STOC'08)*, pp. 629–638.
[17] J. Kamp, A. Rao, S. Vadhan, and D. Zuckerman, "Deterministic extractors for small-space sources," in *Proc. 38th Annu. ACM Symp. Theory Comput. (STOC'06)*, pp. 691–700.
[18] J. Kamp and D. Zuckerman, "Deterministic extractors for bit-fixing sources and exposure-resilient cryptography," *SIAM J. Comput.*, vol. 36, no. 5, pp. 1231–1247, 2007.
[19] C.-J. Lee, C.-J. Lu, and S.-C. Tsai, "Deterministic extractors for independent-symbol sources," in *Proc. 33rd Int. Colloq. Automata, Lang., Program. (ICALP 2006)*, pp. 84–95.
[20] C.-J. Lee, C.-J. Lu, S.-C. Tsai, and W.-G. Tzeng, "Extracting randomness from multiple independent sources," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 2224–2227, Jun. 2005.
[21] C.-J. Lu, O. Reingold, S. Vadhan, and A. Wigderson, "Extractors: Optimal up to constant factors," in *Proc. 35th Annu. ACM Symp. Theory Comput. (STOC'03)*, pp. 602–611.
[22] N. Nisan and D. Zuckerman, "Randomness is linear in space," *J. Comput. Syst. Sci.*, vol. 52, no. 1, pp. 43–52, 1996.
[23] A. Rao, "Extractors for a constant number of polynomially small min-entropy independent sources," *SIAM J. Comput.*, vol. 39, no. 1, pp. 168–194, 2009.
[24] R. Raz, "Extractors with weak random seeds," in *Proc. 37th Annu. ACM Symp. Theory Comput. (STOC'05)*, pp. 11–20.
[25] R. Shaltiel, "Recent developments in explicit constructions of extractors," *Bull. Eur. Assoc. Theor. Comput. Sci.*, vol. 77, pp. 67–95, 2002.
[26] A. Ta-Shma and D. Zuckerman, "Extractor codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3015–3025, Dec. 2004.
[27] L. Trevisan, "Extractors and pseudorandom generators," *J. ACM*, vol. 48, no. 4, pp. 860–879, 2001.
[28] L. Trevisan and S. Vadhan, "Extracting randomness from samplable distributions," in *Proc. 41st Annu. IEEE Symp. Found. Comput. Sci. (FOCS'00)*, pp. 32–42.
[29] A. C. Yao, "Theory and applications of trapdoor functions," in *Proc. 23rd Annu. IEEE Symp. Found. Comput. Sci. (FOCS'82)*, pp. 80–91.
[30] D. Zuckerman, "General weak random sources," in *Proc. 31st Annu. IEEE Symp. Found. Comput. Sci. (FOCS'90)*, pp. 534–543.

Chia-Jung Lee received the B.S. degree from the National Taiwan Normal University, Taipei, Taiwan, in 2000, and the Ph.D. degree in computer science from the National Chiao-Tung University, Hsinchu, Taiwan, in 2010. She is now doing postdoctoral research at the Institute of Information Science, Academia Sinica, Taipei, Taiwan. Her research interests are randomness in computation, cryptography, and theoretical computer science.

Chi-Jen Lu received his B.S. and M.S. degrees from National Taiwan University, Taiwan, in 1988 and 1990 respectively, and his Ph.D. degree from University of Massachusetts at Amherst, USA, in 1999, all in computer science. He is currently a research fellow in the Institute of Information Science, Academia Sinica, Taiwan. His research interests include randomness in computation, computational complexity, cryptography, game theory, and machine learning.

Shi-Chun Tsai (M'06) received his B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taiwan, in 1984 and 1988, respectively, and the Ph.D. degree in computer science from the University of Chicago, USA, in 1996. During 1993–1996, he served as a Lecturer in the Computer Science Department, University of Chicago. During 1996–2001, he was Associate Professor of Information Management Department, and Computer Science and Information Engineering Department, National Chi Nan University, Taiwan. He has been with the Department of Computer Science, National Chiao Tung University, Taiwan since 2001, and was promoted to full Professor in 2007. He is currently serving as the Director of the Information Technology Service Center of National Chiao Tung University. His research interests include computational complexity, algorithms, coding theory, and combinatorics.