Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 149 (2023)

Explicit Codes for Poly-Size Circuits and Functions that are Hard to Sample on Low Entropy Distributions

Ronen Shaltiel* Jad Silbak†

November 13, 2023

Abstract

Codes for poly-size circuits. Guruswami and Smith (J. ACM 2016) considered codes for channels that are poly-size circuits which modify at most a p-fraction of the bits of the codeword. This class of channels is significantly stronger than Shannon's binary symmetric channel (BSC), but weaker than Hamming's channels which are computationally unbounded.

The goal of this direction is to construct explicit codes (namely, codes with poly-time encoding and decoding algorithms) with rate $R(p) = 1 - H(p)$ (matching the capacity of the BSC, and *beating the capacity of codes for Hamming's channels*). This goal implies circuit lower bounds, and specifically that $\mathrm{E} = \mathrm{DTIME}(2^{O(n)})$ does not have poly-size circuits (and therefore explicit constructions need to be based on hardness assumptions).

We give the first explicit construction of such codes for poly-size channels. Specifically, for every $0 \leq p < \frac{1}{4}$, there are explicit codes with rate $R(p) = 1 - H(p)$, assuming E does not have size $2^{\Omega(n)}$ nondeterministic circuits. This hardness assumption was introduced in the context of hardness vs. randomness tradeoffs, and is by now standard in complexity theory.

Our result builds on, and improves the previous work of Guruswami and Smith, and Shaltiel and Silbak (FOCS 2022). (These works gave a randomized Monte-Carlo construction, rather than explicit codes).

Functions that are hard to sample on low entropy distributions. A key component in our codes (that may be of independent interest) is a new complexity theoretic notion of *hard to sample functions* (HTS): We say that a function f on n bits is an HTS for circuits of size n^c, if there exists a constant $c' > c$, such that for every randomized circuit A of size n^c that samples a distribution (X, Y) with $H_\infty(X) \geq c' \cdot \log n$, it holds that $\Pr[Y = f(X)] \leq \frac{1}{n^c}$.

This is inspired by a related notion introduced by Viola (SICOMP 2012), in which X is the uniform distribution. Here, we allow A to choose any distribution X (except for distributions X with very low min-entropy) and note that a circuit A of size n^c, may be hardwired with $\approx n^c$ outputs of f, and therefore, can easily produce pairs $(X, f(X))$ for a distribution X, with $H_\infty(X) \approx c \log n$.

Building on classical works on "hardness amplification" (and using many additional tools and ideas from pseudorandomness) we show that if E does not have size $2^{\Omega(n)}$ nondeterministic circuits, then for every constant c, there is an HTS that is computable in time $\mathrm{poly}(n^c)$.

Our codes are obtained by using our HTS (as well as additional tools and ideas) to achieve explicit constructions (under the hardness assumption) of several components in the code of Shaltiel and Silbak, replacing previously obtained randomized Monte-Carlo constructions of these components. We then need to revisit the codes of Shaltiel and Silbak, and significantly modify the construction and analysis, so that they works with the weaker components that we are able to explicitly construct.

*University of Haifa, Email: ronen@cs.haifa.ac.il.

†Northeastern University, Email: jadsilbak@gmail.com.

ISSN 1433-8092

Contents

1 Introduction

1.1 Codes and channels

Coding theory studies transmission of messages using noisy channels. In this paper we are interested in binary codes, and prefer to focus on decoding properties of a code, rather than combinatorial properties like minimal distance. More specifically, given a family $\mathcal{C}$ of (possibly randomized) functions $C : \{0,1\}^n \to \{0,1\}^n$ (which we call "channels") the goal is to design a code (namely, a pair $(\mathrm{Enc}, \mathrm{Dec})$ of an encoding map $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ and a decoding map $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k$) such that for every message $m \in \{0,1\}^k$ and every channel $C \in \mathcal{C}$, decoding is successful when the channel C chooses a "noise vector" $e = C(\mathrm{Enc}(m))$ as a function of $\mathrm{Enc}(m)$. More formally, decoding is successful if:

$$\mathrm{Dec}(\mathrm{Enc}(m) \oplus C(\mathrm{Enc}(m))) = m.$$

The rate of a code is $R = \frac{k}{n}$. For a family $\mathcal{C}$ of channels, we use $R(\mathcal{C})$ to denote the capacity of the family, which is the best possible rate of a code for this family.[1] For a family $\mathcal{C}$ of channels, there are two main goals:

1. Determine the capacity $R(\mathcal{C})$.
2. Construct explicit codes (namely codes with poly-time encoding and decoding algorithms).

Let us review some coding scenarios and channel families. In all examples below $0 \leq p < \frac{1}{2}$ is a parameter.

Binary symmetric channels. A binary symmetric channel (denoted by BSC_p) is the randomized function that ignores its input and produces a "noise vector" of n i.i.d. random bits, where each of them is one with probability p. This is a special case of an extensively studied class of randomized channels (often referred to as "Shannon channels"). A celebrated theorem of Shannon shows that $R(\mathrm{BSC}_p) = 1 - H(p)$.[2] Later work on code concatenation [For65] produced codes with explicit and even linear time algorithms [GI05].

Hamming channels. The class of Hamming channels (denoted by Ham_p) is the class of all functions such that for every input x, the relative Hamming weight of $C(x)$ is at most p.[3] (This corresponds to a channel that flips at most a p fraction of the bits). This class is probably the most studied class of channels, and yet, its capacity $R(\mathrm{Ham}_p)$ is not precisely understood. It is known that $R(\mathrm{Ham}_p) = 0$ for $p \geq \frac{1}{4}$, and that for $0 < p < \frac{1}{4}$, $R(\mathrm{Ham}_p) < 1 - H(p)$.[4] The Gilbert-Varshamov bound shows that $R(\mathrm{Ham}_p) \geq R^{\mathrm{GV}}(p) = 1 - H(2p)$, but explicit codes with this rate are unknown. Recently, there has been progress on explicit codes with rate that is close to the Gilbert-Varshamov bound for p approaching $\frac{1}{4}$ [TS17, JST21, BD22].

Poly-size channels. Lipton [Lip94] suggested to consider intermediate families of channels consisting of Hamming channels that are *computationally bounded*. Following a seminal paper of Guruswami and Smith [GS16], we will consider the class of Hamming channels that can be implemented by polynomial size circuits. More formally, let Ckt^s_p be the class of all $C \in \mathrm{Ham}_p$ such that C is a circuit of size s. We will focus on the case that $s = n^c$ for a constant c, and call this class poly-size channels.

[1] More formally, $R(\mathcal{C})$ is the largest number R such that for every $\epsilon > 0$, there exist infinitely many n, for which there exists a code for $\mathcal{C}$, with rate at least $R - \epsilon$. We mostly use the term "rate" for a specific (family of) codes, and "capacity" for a class of channels, but these terms are interchangeable in this paper.

[2] Here $H(p) = p \cdot \log(1/p) + (1-p) \cdot \log(1/(1-p))$ is Shannon's entropy function.

[3] The relative Hamming weight of a string $z \in \{0,1\}^n$ is $wt(z) = \frac{|\{i \in [n] : z_i \neq 0\}|}{n}$.

[4] This follows because by the Elias-Bassalygo bound, which states that $R(\mathrm{Ham}_p) < R_{\text{Elias-Bassalygo}}(p)$ where the latter is strictly smaller than $1 - H(p)$. We remark that the Elias-Bassalygo bound gives a stronger result, and that later work by McEliece, Rodemich, Rumsey and Welch [MRRW77] improves this bound in some ranges. We state the bound $R < 1 - H(p)$ to stress that $R(\mathrm{Ham}_p) < R(\mathrm{BSC}_p) = 1 - H(p)$.

It turns out that with the standard definition of codes, every code for circuits of *linear size* is also a code for Hamming channels.[5] This means that in order to take advantage of restricted families of channels, one needs to consider a different scenario. Several such scenarios were considered in the literature. In this paper, we follow the approach of Guruswami and Smith [GS16] and consider *stochastic codes*.

Stochastic codes. These are codes where the encoding algorithm is *randomized*, and decoding only needs to succeed with high probability. More precisely, an encoding map of a *stochastic code*, is a function $\mathrm{Enc} : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^n$, and a decoding map is a function $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k$. It is required that for every $m \in \{0,1\}^k$, and every channel C in the considered class:

$$\Pr_{S \leftarrow U_d}[\mathrm{Dec}(\mathrm{Enc}(m,S) \oplus C(\mathrm{Enc}(m,S)) = m] \geq 1 - \nu,$$

where ν is an error parameter. (A precise formal definition is given in Definition 2.10). Note that the decoding algorithm *does not* need to receive the randomness S chosen by the encoder, and so, these codes can be used in the standard coding communication scenario. The rate of a stochastic code is $R = \frac{k}{n}$.

Stochastic codes do not give an improvement in capacity in the case of Hamming channels (as it is easy to show that a stochastic code for Hamming channels yields a standard code with the same rate) but they do allow improved capacities for other classes. Such improvements were obtained for weaker sub-classes of Hamming channels such as "additive channels" [GS16] and "small space channels" [SS21b]. In both cases, the capacity was shown to be $1 - H(p)$, and this was achieved with polynomial time encoding and decoding algorithms. (See Section 1.4 for a brief survey).

Summing up, if one allows stochastic codes (instead of standard codes) and restricts the class of Hamming channels to be computationally bounded, then we may expect to obtain the optimal capacity of the BSC, beating the capacity of Hamming channels. Furthermore, for some weak classes of channels this was achieved with explicit stochastic codes.

1.1.1 An explicit construction of stochastic codes for poly-size circuits under hardness assumptions

In this paper we achieve a similar result for the *significantly stronger* class of poly-size channels. For this class, we cannot hope for an unconditional explicit code. This is because it is easy to see (see. e.g., [SS21a]) that an explicit stochastic code for a class $\mathcal{C}$, implies that the class $\mathrm{E} = \mathrm{DTIME}(2^{O(n)})$ is not contained in $\mathcal{C}$, and we do not know how to prove such lower bounds against poly-size circuits. Therefore, inspired by the "hardness vs. randomness tradeoffs" of Nisan and Wigderson [NW94] and Impagliazzo and Wigderson [IW97], Guruswami and Smith suggested to base explicit constructions of such stochastic codes on circuit lower bounds (which are often referred to as *hardness assumptions*).

Hardness assumptions. We say that "E is hard for exponential size circuits of some type", if there exist a problem $L \in \mathrm{E} = \mathrm{DTIME}(2^{O(n)})$ and a constant $\beta > 0$, such that for every sufficiently large n, circuits of size $2^{\beta \cdot n}$ (of the specified type) fail to compute the characteristic function of L on inputs of length n. (See Section 2.1 for a more formal definition).

The assumptions that E is hard for exponential size (deterministic) circuits was used by the celebrated paper of Impagliazzo and Wigderson [IW97] to imply that $\mathrm{BPP} = \mathrm{P}$. The assumption that E is hard for exponential size nondeterministic circuits (which originated in hardness versus randomness for AM) has also become standard in complexity theory, and was used in many results [FL97, KvM02, MV05, TV00, SU05, BOV07, GW02, GST03, SU06, SU09, Dru13, AASY15, BV17, AIKS16, HNY17, DMOZ22, BDL22, CT22]. It can be viewed as a scaled, nonuniform versions of the widely believed assumption that EXP $\neq$ NP.

[5] This follows as if there is a message $m \in \{0,1\}^k$ and a channel $C \in \mathrm{Ham}_p$ such that $\mathrm{Dec}(\mathrm{Enc}(m) \oplus C(\mathrm{Enc}(m))) \neq m$, then the channel $C'(x) = C(\mathrm{Enc}(m))$ is a channel on which decoding is not successful. Note that the channel C' computes a *constant function*, and therefore has low complexity in any nonuniform model of computation. We discuss the class of Hamming channels that are constant functions a.k.a. *additive channels* in Section 1.4.

Our result. Assuming a hardness assumption against nondeterministic circuits, we give the first explicit construction of codes for poly-size circuits, achieving the optimal capacity of $R = 1 - H(p)$ (matching the capacity of the BSC, and beating the capacity of Hamming channels). This is stated next (a more formal statement appears as Theorem 6.1).

Theorem 1.1 (Explicit stochastic codes for poly-size channels)**.** *If E is hard for exponential size nondeterministic circuits then for every constants $0 \leq p < \frac{1}{4}$, $c > 1$, and for every sufficiently small constant $\epsilon > 0$, for infinitely many n, there is stochastic code $(\mathrm{Enc}, \mathrm{Dec})$ for $\mathrm{Ckt}_p^{n^c}$ with rate $R \geq 1 - H(p) - \epsilon$, and success probability $1 - \frac{1}{n^c}$. Furthermore, the construction is explicit and $\mathrm{Enc}, \mathrm{Dec}$ are computable in time $\mathrm{poly}(n^c)$.*

Previous work on codes for poly-size circuits. In their seminal paper, Guruswami and Smith [GS16] introduced this problem, and showed that for $p > \frac{1}{4}$, the capacity of $\mathrm{Ckt}_p^{O(n)}$ is zero. For $0 \leq p < \frac{1}{4}$, they gave stochastic codes with rate $1 - H(p)$ for poly-size channels, which are similar to those in Theorem 1.1, except for two caveats:

1. The stochastic codes are not explicit, and instead, they were achieved by a "Monte-Carlo randomized construction". This means that there is a pre-processing stage in which a polynomially long string is chosen uniformly, and published (so that it is available to the encoding algorithm, decoding algorithm and the channel) and the correctness of the stochastic code is guaranteed with high-probability over this choice.[6] (See Definition 2.10 for a formal definition).
2. The code is *list-decodable* rather than *uniquely-decodable*, meaning that Dec is allowed to output a list of $L(\epsilon)$ candidates (where $L(\epsilon)$ is a constant that depends on ϵ) and it is required that w.h.p. the correct message appears in the list.

Later work by Shaltiel and Silbak [SS21a] showed how to make the construction of *list-decodable stochastic codes* explicit, under the hardness assumption that E is hard for exponential size circuits. Recently, Shaltiel and Silbak [SS22] gave a Monte-Carlo randomized construction of *uniquely-decodable* stochastic codes with rate $1 - H(p)$. They left the open problem of achieving an explicit construction of a uniquely decodable code, under hardness assumptions.

Comparison to previous work. In this paper, we obtain a stochastic code that doesn't suffer from any of the caveats! We achieve the best possible rate of $1 - H(p)$ (matching the capacity of the BSC, and *beating the rate of the best possible codes for Hamming channels*). This is achieved for every $0 \leq p < \frac{1}{4}$, with polynomial time encoding and decoding algorithms. (As in the case of all the aforementioned previous work, the polynomial in the running time of $\mathrm{Enc}, \mathrm{Dec}$ is larger than n^c, and it is open whether there exist codes where $\mathrm{Enc}, \mathrm{Dec}$ can be simulated by the channel. See Section 7 for a discussion).[7]

As noted earlier, the assumption that E is hard for exponential size nondeterministic circuits is a standard hardness assumption that is used in many contexts in complexity theory and cryptography. It is open whether the assumption can be relaxed to the necessary assumption that E is hard for poly-size circuits.

Explicit conditional constructions vs. randomized Monte-Carlo constructions. An explicit conditional construction under hardness assumptions (like the one in this paper) immediately implies an (unconditional) Monte-Carlo randomized construction. This is because by a standard counting argument, a random string R of length n^d (where d is slightly larger than c) can be interpreted as a truth table of a function $f : \{0,1\}^{d \log n} \to \{0,1\}$ that is hard for circuits of size n^c of *any* type (and indeed, n^c is almost exponential in $d \log n$).

[6] We remark that the random string that is chosen and made public, is of length larger than n^c, and so it is arguable in what sense the size n^c channel C can "read" it.

[7] Note that in contrast to codes for Hamming channels, for poly-size channels, the "list-decoding capacity" and "unique-decoding capacity" coincide for every $0 \leq p < \frac{1}{4}$. We remark that for $\frac{1}{4} < p < \frac{1}{2}$, the unique-decoding capacity is zero [GS16], whereas the list-decoding capacity is $1 - H(p)$, and this was achieved by the list-decodable stochastic codes of [GS16, SS21a]).

This means that a conditional explicit construction under hardness assumptions, can always be instantiated if one has access to an n^d bit long random string R (as in the Monte-Carlo setup).

Moreover, this example demonstrates why it is significantly more difficult to obtain explicit constructions under hardness assumptions, than Monte-Carlo randomized constructions: Unlike Monte-Carlo randomized constructions (which can make use of any pseudorandom object that can be shown to exist under the probabilistic method), explicit conditional constructions can only rely on pseudorandom objects that we can explicitly construct *unconditionally*, or under *hardness assumptions*.

Indeed, jumping ahead (see Section 1.3) some of the technical contribution of this paper is giving explicit conditional constructions of "functions that are hard to sample on low entropy distributions" and using these, to explicitly construct some of the pseudorandom objects that were used in the Monte-Carlo randomized construction of Shaltiel and Silbak [SS22].

1.2 Functions that are hard to sample on low entropy distributions (HTS)

We now present a new notion of "hard functions" that is inspired by, and builds on, notions of "hardness of sampling". Viola [Vio12] suggests to systematically study functions $f : \{0,1\}^n \to \{0,1\}^{n'}$ that are not only *hard to compute*, but also *hard to sample*, in the sense that no low-complexity sampling algorithm can sample a pair $(X, f(X))$ where X is uniform over $\{0,1\}^n$.

Previous work on this notion, focused mostly on weak classes (such as constant depth circuits [Vio12], or ROBPs [EGZ22]) whereas we will be interested in poly-size circuits (and cannot hope for unconditional results). More significantly, we will want functions such that it is hard to sample a pair $(X, f(X))$, not only for the uniform distribution, but also for every low min-entropy samplable X. In the definition below, we require an even stronger property: that every sampling circuit A is unlikely to produce a pair $(x, f(x))$ that wasn't initially "hardwired to A".

1.2.1 Definition of HTS using small sets

We say that a function $f : \{0,1\}^n \to \{0,1\}^{n'}$ is an ***h*-HTS** for circuits of size n^c, if for every size n^c circuit A that samples a distribution $Z = (X, Y)$ over $\{0,1\}^n \times \{0,1\}^{n'}$, there exists a set $H \subseteq \{0,1\}^n$ of size at most h, such that:

$$\Pr_{(X,Y)\leftarrow A}[X \notin H \text{ and } Y = f(X)] \leq \frac{1}{n^c}.$$

(A more formal definition appears in Definition 3.1).

In order to motivate this definition, note that a size n^c circuit A can be hardwired with $t \approx n^c$ pairs $(x_1, f(x_1)), \ldots, (x_t, f(x_t))$, which enables A to produce the distribution $(X, f(X))$, for a uniformly chosen $X \leftarrow \{x_1, \ldots, x_t\}$. This allows A to produce a distribution (X, Y) with $H_\infty(X) = \log t \approx c \log n$, on which $Y = f(X)$ with probability one.

Loosely speaking, in the case that $h = \mathrm{poly}(n^c)$, the definition above essentially says that this is the worst that a size n^c circuit A can do: For every circuit A, there is a set $H \subseteq \{0,1\}^n$ of size comparable to n^c of inputs, such that A is unlikely to sample a pair (X, Y) such that $X \notin H$ and $Y = f(X)$.[8]

For any value of h, the definition implies that f is *hard to sample* on every samplable distribution with min-entropy $k \approx \log h$. Specifically, it is immediate that if f is an h-HTS for circuits of size n^c, then f is:

Hard to sample on low min-entropy distributions: For every circuit A of size n^c that samples a distribution $Z = (X, Y)$ where $H_\infty(X) \geq \log h + c \log n$, $\Pr_{(X,Y)\leftarrow A}[Y = f(X)] \leq \frac{2}{n^c}$. [9]

[8] This "definitional methodology" of trying to argue that nonuniform adversaries are not able to produce a string that was not "hardwired to the circuit" has been used recently in several scenarios. Two examples that we are aware of, are definitions of multi-collision resistance of keyless hash functions [BKP18], and definitions of certain variants of non-malleable codes [FMVW16, SS22].

[9] While an HTS is hard to sample on low min-entropy distributions (see Section 3.1.2) there does not seem to be an implication

We would like to relate an HTS to the more standard notion of functions that are *hard to compute*. A function that is *hard to sample* on some samplable distribution X, is in particular *hard to compute* when the input is chosen from X. (This follows, because every circuit C which attempts to computes f on a samplable distribution X, induces a sampling algorithm A which samples X, and produces $Y = C(X)$). Specifically (see Section 3.1.2 for a more formal statement) if f is an h-HTS, then f is also:

Hard to compute on low min-entropy samplable distributions: For every X that is samplable by size n^c circuits, with $H_\infty(X) \geq \log h + c \log n$, and for every circuit C of size n^c, $\Pr[C(X) = f(X)] \leq \frac{2}{n^c}$.

In particular, If $h \leq 2^n - c \log n$ (that is, if h is only slightly smaller than the trivial 2^n) then f is:

Hard to compute on the uniform distribution: For $X \leftarrow U_n$, and for every circuit C of size n^c, $\Pr[C(X) = f(X)] \leq \frac{2}{n^c}$.

HTS vs. functions that are hard to compute on the uniform distribution. Functions that are hard to compute on the uniform distribution are widely studied in complexity theory, and have many applications. As we have observed above, an HTS (with say $h = \mathrm{poly}(n^c)$) is stronger than such functions in two respects:

1. The function f is hard to compute not just when the input X is chosen according to the uniform distribution, but also for every samplable distribution X with $H_\infty(X) \geq \log h \approx c \log n$.
2. This holds even if the circuit attempting to compute f doesn't receive X as input, and instead is allowed to sample X on its own (while potentially also preparing correlated side information).[10]

Our motivation for defining and constructing HTS is that we use them as a component in our construction of stochastic codes for poly-size channels. However, it seems to us that functions that are hard to compute/sample on low min-entropy samplable distributions are interesting regardless of this application.

1.2.2 An explicit construction of HTS under hardness assumptions

In this paper we give explicit constructions of hard to sample functions from standard hardness assumptions. This is similar in spirit to past work on "hardness amplification" [IW97, STV01] (and many subsequent works) which show how to construct functions that are hard to compute on the uniform distribution, assuming E is hard for exponential size circuits. We give two constructions of HTS based on the assumption that E is hard for exponential size nondeterministic circuits (this assumption has already appeared in constructions of hard functions [TV00, Dru13]). By the previous discussion this gives functions that are hard to compute on low min-entropy samplable distributions. To the best of our knowledge, prior to this work, there were no known constructions of functions that are hard to compute on low min-entropy samplable distributions.[11]

in the other direction. Loosely speaking, this is because the HTS guarantee applies to *every* circuit A of size n^c (including circuits that sample distributions X that do not have $H_\infty(X) \geq \log h$). To demonstrate this point, consider the case where A samples a distribution (X, Y) where X is only statistically close to a distribution X' that has large min-entropy. Such a distribution X may have very low min-entropy (and note that X' is not necessarily efficiently samplable). The notion of "hard to sample on low min-entropy samplable distributions" does not seem to imply that $\Pr_{(X,Y)\leftarrow A}[Y = f(X)]$ is small. In contrast, the definition of HTS that we chose (using the terminology of small sets, rather than the statement with min-entropy that appears in the abstract) applies to A, and *does* guarantee that $\Pr_{(X,Y)\leftarrow A}[Y = f(X)]$ is small (see discussion in Section 3.1.3). We highlight this technicality because this versatility of the definition of HTS will be crucial in our application to codes, and will also make an HTS more "composable". This "composability" will be very helpful when constructing an HTS (see Section 1.3.2).

[10] To demonstrate this point, consider the following example: Let $g : \{0,1\}^n \to \{0,1\}^n$ be a one-way permutation, and let $f = g^{-1}$. It is immediate that an efficient adversary A cannot compute $f(X)$ on $X \leftarrow U_n$. However, if we allow A to sample $X \leftarrow U_n$ on its own, then A can easily produce pairs of the form $(X, f(X))$ by sampling $Y \leftarrow U_n$ and producing $(g(Y), Y)$. This demonstrates that sampling may be easier than computing, and therefore, proving lower bounds on sampling is harder than proving lower bounds on computing.

[11] A *boolean function* that is "very hard" to compute on low min-entropy distributions, is in particular, an "extractor for samplable distributions" as defined by Trevisan and Vadhan [TV00]. Current extractors for samplable distributions [TV00, AASY15] (which are based on similar hardness assumptions) cannot handle distributions with min-entropy smaller than $n/2$. In contrast, we will be able to handle much lower min-entropy. Our results do not translate into extractors for samplable distributions, because our functions have large output length, but Theorem 1.3 can be viewed as a "condenser for low min-entropy samplable distributions".

A parameter that we have not yet discussed is the output length of an HTS. Naturally, shorter output length is preferable. The two constructions below represent two tradeoffs between the set size h, and the output length of f. (Jumping ahead, we will use HTS for several tasks in our construction of stochastic codes for poly-size circuit, and the different tasks will require different tradeoffs).

Theorem 1.2 (HTS with small h and large output length)**.** *If E is hard for exponential size nondeterministic circuits then for every constant $c > 1$, there exist constants $c', d > c$, such that for every sufficiently large n, there is a function $f : \{0,1\}^n \to \{0,1\}^{n^{c_0}}$ such that f is an $n^{c'}$-HTS for circuits of size n^c (here, $c_0 > 1$ is a universal constant). Furthermore, f is computable in time n^d.*

Theorem 1.2 is stated more formally in Theorem 3.4. This HTS has very low h that is optimal up to a polynomial. This will be crucial in some of our applications. However, the output length of f is larger than n. This turns out to be problematic in some other applications (as the ratio between $n + n'$ and n will sometimes come into the rate of the codes we construct). This motivates constructing a second HTS with $n' \ll n$. We are able to do this, for "medium sized h", namely $h = 2^{\delta n}$ for a small constant $\delta > 0$. This larger value of h is good enough for our intended application.

Theorem 1.3 (HTS with medium h and small output length)**.** *If E is hard for exponential size nondeterministic circuits then for every constants $c > 1$, $\delta > 0$ and $0 < \lambda < 1$, there exists a constant d such that for every sufficiently large n, there is a function $f : \{0,1\}^n \to \{0,1\}^{\lambda \cdot n}$ that is a $2^{\delta n}$-HTS for circuits of size n^c. Furthermore, f is computable in time n^d.*

Theorem 1.3 is stated more formally in Theorem 3.5. Our techniques can also give other tradeoffs (assuming stronger assumptions) see Section 7 for a discussion and open problems. Recently, Ball, Shaltiel and Silbak [BSS23] used our constructions of HTS (together with other components of this paper) to give an improved explicit construction of high rate non-malleable codes for poly-size circuits. We hope that these constructions of HTS (and functions that are hard to compute on low min-entropy samplable distributions) will turn out to be useful in other scenarios in complexity theory, cryptography, and pseudorandomness.

1.3 Overview of the technique

In this section we give an overview of the main ideas that we use. For this purpose we will allow ourselves to be informal. The later technical sections do not build on the content of this section, and the reader can skip to the technical section if they wish. In Sections 1.3.1, 1.3.2, 1.3.3 we explain our constructions of HTS. In Sections 1.3.4, 1.3.5, 1.3.6 we explain the construction of stochastic codes (building on our HTS).

1.3.1 Overview of a construction of a weak HTS (with small input length, and very large h)

We would like to construct an HTS based on hardness assumptions. Our starting point is the classical complexity theoretic results on "hardness amplification" due to Impagliazzo and Wigderson [IW97], and Sudan, Trevisan and Vadhan [STV01]. These show that for every constant $c > 1$, if E is hard for exponential size circuits, then the assumption can be "amplified" to give a function $f : \{0,1\}^{m=\Theta(\log n)} \to \{0,1\}^{m'=\Theta(\log n)}$ that is hard to compute on the uniform distribution: For every circuit C of size n^c, $\Pr_{X \leftarrow U_m}[C(X) = f(X)] \leq \frac{1}{n^c}$, and furthermore, f is computable in time $\mathrm{poly}(n^c)$. We would like to extend this result in two respects:

- We want this to hold even if X is sampled by the circuit (rather than given to it as input).
- We want this to hold even if X is a low min-entropy distribution (rather than the uniform distribution).

Handling the first item: sampling vs. computing. We solve the first issue at the cost of assuming a stronger assumption against nondeterministic circuits. We will now explain this approach.

When we want to prove that a function f is hard to sample, rather than hard to compute, our adversary is a sampling circuit A that produces a distribution $Z = (X, Y)$, and we want to bound the probability that $Y = f(X)$. We are worried that A may sample the input X to f, together with side information, that will help it to obtain $f(X)$ (as in the example with one-way permutations).

We use classical complexity theoretic results on "approximate counting and uniform sampling of NP witnesses" [Sto83, Sip83, JVV86, BGP00] (see Section 2.3.2). These results imply that an NP-circuit (this a circuit that in addition to the standard gates, is also equipped with gates that solve an NP-complete problem, see Definition 2.1) can "reverse" the sampling of a sampling circuit. More precisely, Let A be a circuit that samples some distribution $Z = (X, Y)$ (meaning that on a uniform R, $A(R) = (X, Y)$). These classical results imply that there is a randomized NP-circuit B of roughly the same size as A, that given an input x, produces a string R, such that R is uniformly distributed over "random coins" of A that produce output $X = x$. This means that given input x, the circuit B can "reverse" the sampling process of A, and obtain random coins that lead to x. When given x, we can apply B to obtain the random coins R, and then apply A on R. This computation is an NP-circuit C of roughly the same size as A that can produce any side information that A produces, and in particular, it can produce the output string Y.

This gives that there exists an NP-circuit C of roughly the same size of A, that computes $f(X)$ (given input X) with the same success probability that A achieves. In particular, for the case where X is the uniform distribution, we obtain that a function f that is *hard to compute* on the uniform distribution (for NP-circuits) is also *hard to sample* (for the case where X is the uniform distribution) for (standard) circuits.

It is standard (see, e.g. [KvM02]) that in the aforementioned hardness amplification result of [IW97, STV01], if one wants the function f to be hard to compute on the uniform distribution for NP-circuits (rather than for standard circuits) then it is sufficient that the initial assumption holds against NP-circuits (and in fact, using more care, and the "downwards collapse theorem of [SU06]) even against nondeterministic circuits.

Consequently, combining these results, under the assumption that E is hard for exponential size non deterministic circuits, we can get a function $f : \{0,1\}^{m=\Theta(\log n)} \to \{0,1\}^{m'=\Theta(\log n)}$ that is hard to compute on the uniform distribution (for NP circuits) and therefore, hard to sample for standard circuits (in the case where X is the uniform distribution).[12] Using a more delicate argument (that we will not explain in this overview) we argue that f is an h-HTS (according to the precise definition with small set) with $h = 2^{(1-\alpha)m}$ for some small constant $\alpha > 0$. The precise details are given in Section 3.3.

Handling the second item: from uniform to low min-entropy. In the next two subsections we present several techniques to take a "weak HTS" like the one we already constructed, and convert it to a strong HTS (like the ones that we promised to construct in Theorems 1.2 and 1.3). More precisely, we start with an h-HTS f on $m = O(\log n)$ bits, with $h = 2^{(1-\alpha)m}$ (note that here h is only slightly smaller than the trivial 2^m). We show how to convert f into an HTS f' (on n bits) that is claimed by Theorem 1.2 and Theorem 1.3.

1.3.2 Overview of the proof of Theorem 1.2: Reducing h using list-recoverable codes

By the previous discussion, under the assumption that E is hard for exponential size nondeterministic circuits, we have obtained a $f : \{0,1\}^{m=\Theta(\log n)} \to \{0,1\}^{m'=\Theta(\log n)}$ that is an h-HTS for size n^c circuits, with $h = 2^{(1-\alpha)m}$, where $\alpha > 0$ is some constant. We would like to transform f into the function guaranteed in Theorem 1.2, namely, a function $f' : \{0,1\}^n \to \{0,1\}^{n'=n^{O(1)}}$ that is an h'-HTS for $h' = \mathrm{poly}(n^c)$.

We will show how to construct f' from f using *list-recoverable codes*. List-recoverable codes are the following generalization of list-decodable codes: A a function $E : \{0,1\}^n \to (\{0,1\}^m)^D$ is an (h, L)-list

[12] Note that this already produces a function f that is somewhat hard to sample in a sense that resembles the one considered by Viola [Vio12].

recoverable code if for every collection $H_1, \ldots, H_D$ of subsets of $\{0,1\}^m$ of size h, the size of

$$H = \{x : \forall i \in [D], E(x)_i \in H_i\} \tag{1}$$

is at most L (see Definition 2.22 for a more formal definition). Let $E : \{0,1\}^n \to (\{0,1\}^m)^D$ be an (h, L)-list recoverable code. There are such explicit constructions (which we will elaborate on later) which for our setting of parameters (namely $m = O(\log n)$, and $h = 2^{(1-\alpha)\cdot m}$) achieve $L = \text{poly}(h) = \text{poly}(n^c)$, and $D = \text{poly}(n)$. We will now show how to use E in order to transform the h-HTS $f : \{0,1\}^m \to \{0,1\}^{m'}$ into a function $f' : \{0,1\}^n \to \{0,1\}^{n'=(m+m')D}$. We will then show that f' is an L-HTS (giving that f' has the parameters guaranteed by Theorem 1.2). For this purpose we define:

$$f'(x) = ((E(x)_1, f(E(x)_1)), \ldots, (E(x)_D), f(E(x)_D)),$$

In order to show that f' is an L-HTS, we will consider an adversary which is a sampling circuit A that samples $(X, Y) \in \{0,1\}^n \times \{0,1\}^{n'}$ and attempts to break f'. Note that by the definition of f', the string Y should be of the form $Y = ((X_1, Y_1), \ldots, (X_D, Y_D))$.

The key observation is that for every $i \in [D]$, A naturally induces a sampling circuit $\bar{A}_i$ that first simulates A, to sample (X, Y), but then discards X, and outputs only (X_i, Y_i). As $\bar{A}_i$ is a potential adversary for our initial f, the definition of HTS implies that for every $i \in [D]$, the circuit $\bar{A}_i$ has a small set $H_i \subseteq \{0,1\}^m$ of size h, such that $\Pr_{(X_i,Y_i)\leftarrow\bar{A}_i}[X_i \notin H_i \text{ and } Y_i = f(X_i)]$ is small. (Jumping ahead, we mention that here we crucially rely on the fact that we consider circuits A that try to *sample*, rather than *compute*).

We will use the collection $H_1, \ldots, H_D$, and the list-recoverability property to define a set H, using (1). We now argue that H is a suitable small set that can be used against A (as required by the definition in order to show that f' is an HTS). Note that list-recoverability immediately gives that $|H| \leq L$ as required.

List-recoverability also implies that $\{X \notin H\} \Rightarrow \{\exists i : E(X)_i \notin H_i\}$. Furthermore, by the definition of f' we have that $\{Y = f'(X)\} \Rightarrow \{\forall i : (X_i = E(X)_i \text{ and } Y_i = f(X_i))\}$. Together, this gives that:

$$\begin{aligned}\Pr_{(X,Y)\in A}[X \notin H \text{ and } Y = f'(X)] &\leq \Pr_{(X,Y)\leftarrow A}[\exists i : (X_i \notin H_i \text{ and } Y_i = f(X_i))] \\ &\leq \sum_i \Pr_{(X_i,Y_i)\leftarrow\bar{A}_i}[X_i \notin H_i \text{ and } Y_i = f(X_i)],\end{aligned}$$

which we can upper bound using the guarantee that f is an HTS (by a union bound). We conclude that f' is indeed an L-HTS. (The formal argument appears in Section 3).[13]

The use of list-recoverable codes allows us to take an existing HTS f on $m = O(\log n)$ bits, and where $h = 2^{(1-\alpha)m}$ is only a tiny bit smaller than the trivial 2^m, and convert it into a function f (on n bits) *without* increasing h, so that h is now polynomial in the input length n. The cost of this transformation is that the output length of f' is increased to at least $D \cdot m$ (which is at least n in any list-recoverable code).

List-recoverable codes from high error dispersers. We obtain our list-recoverable codes, by using explicit constructions of dispersers (see Section 2.4 for a formal definition) and using the relationship between list-recoverable codes and dispersers that was discovered by Ta-Shma and Zuckerman [TZ04]. An important aspect of our transformation from f to f' is that it does not require the error parameter of the disperser to be small, and we can use high error dispersers (where the error parameter ϵ approaches one, rather than zero).[14] While this is not crucial for proving Theorem 1.2, the ability to use high error dispersers is crucial in the proof of Theorem 1.3, as we will explain in the next section.

[13] The argument sketched above critically relies on the fact that for *any* adversary $\bar{A}_i$ for the initial HTS f, we can assign a small set H_i (regardless of whether the distribution X_i sampled by $\bar{A}_i$ has large min-entropy or not). This is crucial because we have no guarantee on the distributions that the various $\bar{A}_i$ sample. Here we critically rely on the difference between an HTS and a function that is hard to sample on low min-entropy distributions that was explained in Footnote 9.

[14] More precisely, the argument of Ta-Shma and Zuckerman [TZ04] can be used to show that a (k, ϵ)-disperser translates into a $(2^k, (1-\epsilon)\cdot 2^m)$-list-recoverable code. This allows taking $\epsilon = 1 - 2^{-\alpha m}$ (which approaches one) in the argument above.

1.3.3 Overview of the proof of Theorem 1.3: Obtaining an HTS with small output length

The function $f' : \{0,1\}^n \to \{0,1\}^{n'}$ that we achieved in the previous section has output length $n' \geq n$. In some of our applications of HTS (see Section 1.3.5) it will be crucial to have an HTS with $n' = \lambda \cdot n$ for a small constant $0 < \lambda < 1$, and we indeed promised to construct such an HTS in Theorem 1.3. The approach we used so far cannot give $n' < n$. This is because list-recoverable codes must have a codeword of length $Dm \geq n$ bits, just in order not to lose the information in the n bit message (let alone decode).

In order to obtain smaller output length, we will present a different interpretation of the aforementioned transformation from f to f' using extractors rather than list-recoverable codes. This interpretation is tailored to construct a function that is hard to sample on low min-entropy distributions (and doesn't quite translate to construct an HTS). Nevertheless, it will lead us to an approach that we can use to reduce the output length.

Rather than thinking of $E : \{0,1\}^n \to (\{0,1\}^m)^D$ as a list-recoverable code, let us define $\mathrm{Ext} : \{0,1\}^n \times [D] \to \{0,1\}^m$ by $\mathrm{Ext}(x,i) = E(x)_i$ and require that Ext is a strong extractor (see Definition 2.21 for a precise definition). Loosely speaking, this gives that if X has sufficiently large min-entropy, then there exists an $i \in [D]$ such that $\mathrm{Ext}(X,i)$ is statistically close to uniform. We can then hope to use the fact that in the i'th pair of outputs in the definition of f', we asked the adversary to produce a pair $(\mathrm{Ext}(X,i), f(\mathrm{Ext}(X,i)))$, and this task "resembles" the task of sampling a pair $(U_m, f(U_m))$ that intuitively breaks f.[15]

Continuing with this intuition, we observe that in our scenario, we are only interested in distributions X that are *samplable* by a size n^c circuit. This suggests that in the construction of f' described above, we can try to replace the extractor Ext (that is designed to work on *every* distribution X with $H_\infty(X) \geq k \approx \log h$) with an extractor $\mathrm{Ext}^{\mathrm{samp}}$ that is only guaranteed to work on *samplable* distributions X with $H_\infty(X) \geq k$. Such extractors for samplable distributions were introduced by Trevisan and Vadhan [TV00]. The advantage of these extractors is that they can be *seedless*, meaning that $D = 1$! Loosely speaking, this suggests that we can hope to get an f' with output length $n' = D(m + m') = m + m' < n$.[16]

A problem with this approach, is that even under very strong hardness assumptions, the best known extractors for samplable distributions do not work for low min-entropy distributions. More precisely, the best known constructions [TV00, AASY15] only achieve $k = (1-\alpha)n$ for some small constant $\alpha > 0$. This is a problem, as for our intended application in Section 1.3.5 (and for proving Theorem 1.3) we will need to be able to choose $h = 2^k$ for $k = \delta n$, where $\delta > 0$ is a small constant. This is a significant barrier, as using the current methodology for constructing extractors for samplable distributions, we cannot expect to get $k < n/2$.

Using seedless extractors and high-error seeded dispersers. We will prove Theorem 1.3 in two steps. First we will transform the weak HTS f (that we constructed from hardness amplification) into a function $f_2 : \{0,1\}^n \to \{0,1\}^{O(\log n)}$ (that has output length that is much shorter than input length). This will be done by the approach described above, using a suitable seedless extractor $\mathrm{Ext}^{\mathrm{Samp}}$ (that we construct from the hardness assumption by adapting an argument of Kinne, van-Melkebeek and Shaltiel [KvMS12]). As this extractor only works for very large min-entropy, we will obtain that f_2 is an h_2-HTS for $h_2 = 2^{n - O(\log n)}$ which is very large. We will then apply (a variant of) the transformation based on list-recoverable codes (explained earlier) to reduce h_2 at the cost of harming the output length. By using a careful argument, and a modified transformation, we are able to obtain an h_3-HTS f_3 where the output length is smaller than the input length, while obtaining $h_3 = 2^{\delta n}$, proving Theorem 1.3. This argument appears in Section 3.

[15] This argument doesn't quite work as sketched above, and in any case, in the analysis using list-recoverable codes, we did much better than we are hoping for here: We were able to get an HTS (and not just a function that is hard to sample on low min-entropy distributions). Furthermore, we were able to use dispersers (which are weaker than extractors) and moreover, we were able to use dispersers with high error ϵ approaching one, which makes no sense for the argument sketched above with extractors.

[16] We remark that due to the aforementioned difference between an HTS and a function that is hard to sample on low min-entropy distributions, in order to make this argument go through, we actually need a stronger object than an extractor for samplable distributions, called "relative error extractors for distributions that are weakly recognizable by NP-circuits". These were defined by Applebaum et al. [AASY15] (See Section 2.4 for a definition). We will ignore this technicality in this high level overview.

We stress that in order to make this argument work, it is crucial to use list-recoverable codes that are equivalent to high-error dispersers. More precisely, we use dispersers with error $\epsilon = 1 - o(1)$, and this is crucial as in contrast to dispersers with small ϵ, high-error dispersers can have $D \ll n$ for small values of k. Such dispersers were explicitly constructed by Zuckerman [Zuc07] based on "somewhere condensers" [BKS+10, Zuc07], and we use this construction in our HTS. (Details are given in Section 3).

1.3.4 Overview of the construction of explicit stochastic codes for poly-size circuits

The randomized Monte-Carlo construction of stochastic codes for poly-size circuits by Shaltiel and Silbak [SS22] builds on earlier work in the area [GS16, SS21a, SS21b, SS22] and uses many ideas and components. We will not give a detailed overview of this constructions here (a good overview can be found in [SS22]).

Many of the components used by the construction of [SS22] have explicit constructions (either conditionally or unconditionally). There are however two components for which [SS22] does not have an explicit construction, and instead settles for a Monte-Carlo randomized construction (that is, a nonexplicit construction using the probabilistic method). These components are called "evasive BSC codes" and "SS-non-malleable codes" and we will discuss them below.

In this paper we show how to explicitly construct these two components based on the HTS functions of Theorems 1.2 and Theorem 1.3. This will allow us to convert the randomized Monte-Carlo construction of stochastic codes for poly-size circuits of [SS22] into an explicit one (under hardness assumptions).[17]

The high level idea of [SS22]. Following earlier works [GS16, SS21a, SS21b], the stochastic code construction of Shaltiel and Silbak [SS22] is based on taking a code $(\mathrm{Enc}_{\mathrm{BSC}}, \mathrm{Dec}_{\mathrm{BSC}})$ for BSC_p (and recall that we have such explicit codes with the desired rate of $1 - H(p)$) and a PRG $G : \{0,1\}^{O(\log n)} \to \{0,1\}^n$ for circuits of size n^c (which we have by the classical results of Impagliazzo and Wigderson [IW97] under the assumption that E is hard for exponential size circuits).

The construction of stochastic code $\mathrm{Enc}(m, S)$ for poly-size circuits, starts by preparing $Z = \mathrm{Enc}_{\mathrm{BSC}}(m) \oplus G(S)$. Intuitively, such a string is pseudorandom from the point of view of a channel $C \in \mathrm{Ckt}_p^{n^c}$, and this intuitively means that C does not choose the noise pattern $C(Z)$ as a function of m. Using additional ideas that we will not explain, it is possible to arrange that C is "forced" to behave like BSC_p.

This means that when the decoding algorithm Dec receives the corrupted codeword $V = Z \oplus C(Z)$, it can correctly decode by computing $\mathrm{Dec}_{\mathrm{BSC}}(V \oplus G(S))$. However, for this, $\mathrm{Dec}_{\mathrm{BSC}}$ must know S (which is not a part of its input). The seminal work of Guruswami and Smith [GS16] developed techniques to somehow embed S into the codeword. These techniques were extended in subsequent work, and we will not try to explain them. What we will say, is that the final codeword is set up to be a function of S and $Z = \mathrm{Enc}_{\mathrm{BSC}}(m) \oplus G(S))$ (as well as some additional randomness).

Shaltiel and Silbak [SS21b, SS22] introduced two components that make the stochastic codes uniquely-decodable. We will now describe these two components, and show how to construct them based on an HTS.

1.3.5 Overview of an explicit construction of evasive BSC codes

The first component of [SS22] that we will discuss is an "evasive BSC code" (which emerged out of [SS21b] which handled "small space channels") and was tailored for poly-size circuits in [SS22].

Definition of evasive BSC codes. Let $(\mathrm{Enc}_{\mathrm{BSC}}, \mathrm{Dec}_{\mathrm{BSC}})$ be a code for BSC_p, we can assume w.l.o.g. that $\mathrm{Dec}_{\mathrm{BSC}}(v)$ outputs "$fail$" if $\delta(v, \mathrm{Enc}_{\mathrm{BSC}}(\mathrm{Dec}_{\mathrm{BSC}}(v))$ is slightly larger than p (as this is unlikely if

[17] We will not be able to exactly reproduce all the properties of the components used in [SS22] with an explicit conditional construction, and will have to settle for weaker components. Consequently, we will need to show that it is possible to carry out the approach of [SS22] with these weaker components. This requires significant work, as well as some new ideas, and the precise argument appears in Section 6. We will not elaborate on this part in this high level overview. We elaborate on some of these ideas in Section 6.2.1.

$v = \mathrm{Enc}_{\mathrm{BSC}}(m) \oplus \mathrm{BSC}_p$ and $\mathrm{Dec}_{\mathrm{BSC}}(v) = m$). The code is *evasive*, if for every channel $C \in \mathrm{Ckt}_p^{n^c}$,

$$\Pr_{Z \leftarrow U_n}[\mathrm{Dec}_{\mathrm{BSC}}(Z \oplus C(Z)) \neq fail] < \frac{1}{n^c}.$$

Note that in the experiment considered above, the channel C is applied on a uniform string $Z \leftarrow U_n$, rather than on a codeword. The evasiveness requirement says that C is unable to make Dec decode (rather than fail) in such a scenario. Loosely speaking, this notion is useful as in the construction described in the previous section, $Z = \mathrm{Enc}_{\mathrm{BSC}}(m) \oplus G(S)$ is pseudorandom (meaning that it is uniformly distributed from the point of view of C and $\mathrm{Dec}_{\mathrm{BSC}}$). This intuitively explains why it is natural to be interested in the case where $Z \leftarrow U_n$ (as in the evasiveness experiment).

Evasive BSC code from HTS with small output length. We will now show how to take a given explicit code $(\mathrm{Enc}_{\mathrm{BSC}}, \mathrm{Dec}_{\mathrm{BSC}})$ for BSC_p (for any $0 \leq p < \frac{1}{4}$) and convert it into one that is also evasive. We will require the $2^{\delta n}$-HTS $f : \{0,1\}^n \to \{0,1\}^{n' = \lambda \cdot n}$ of Theorem 1.3 (for an arbitrary constant $0 < \lambda < 1$).

We will define a new code $(\mathrm{Enc}', \mathrm{Dec}')$ as follows: $\mathrm{Enc}'(m) = \mathrm{Enc}_{\mathrm{BSC}}(m \circ f(m))$ (where "$\circ$" is string concatenation) and $\mathrm{Dec}'(v)$ will run $\mathrm{Dec}_{\mathrm{BSC}}(v)$. If $\mathrm{Dec}_{\mathrm{BSC}}(v)$ does not fail, then $\mathrm{Dec}_{\mathrm{BSC}}$ outputs a pair (x, y), and $\mathrm{Dec}'(v)$ will output x, if $y = f(x)$, and will fail otherwise.

Overall, both $\mathrm{Enc}', \mathrm{Dec}'$ run in time $\mathrm{poly}(n^c)$, and it is obvious that this is a good code for BSC_p. The rate of Enc' deteriorates (as we are using $\mathrm{Enc}_{\mathrm{BSC}}$ to encode a string of length $n + n'$ rather than of length n). This is why we insisted in Theorem 1.3 that $n' = \lambda \cdot n$ for an arbitrary constant $\lambda > 0$, meaning that by choosing $\lambda > 0$ to be sufficiently small, the rate is not harmed.

We now show that this code is evasive. Let $C \in \mathrm{Ckt}_p^{n^c}$ be some channel. We can consider the sampling circuit A that choose $Z \leftarrow U_n$, and outputs $(X, Y) = \mathrm{Dec}_{\mathrm{BSC}}(Z \oplus C(Z))$ (if $\mathrm{Dec}_{\mathrm{BSC}}$ does not fail).
As A cannot break f, we have that there exists a set $H \subseteq \{0,1\}^n$, of size $h = 2^{\delta n}$ such that

$$\Pr_{(X,Y) \leftarrow A}[X \notin H \text{ and } Y = f(X)] \leq \frac{1}{n^c}.$$

However, if $\mathrm{Dec}'(Z \oplus C(Z))$ does not fail, then it produces a message $X \circ Y$, such that $Y = f(X)$, and

$$\delta(\mathrm{Enc}_{\mathrm{BSC}}(X \circ f(X)), Z) \leq 2p + o(1).$$

(Here, $\delta(\cdot, \cdot)$ is the relative Hamming distance). This follows, because by the triangle inequality:

$$\delta(\mathrm{Enc}_{\mathrm{BSC}}(X \circ f(X)), Z) \leq \delta(\mathrm{Enc}_{\mathrm{BSC}}(X \circ f(X)), Z \oplus C(Z)) + \delta(Z \oplus C(Z), Z),$$

and both terms are bounded by about p. It follows that:

$$\Pr[\mathrm{Dec}'_{\mathrm{BSC}}(Z \oplus C(Z)) \neq fail] \leq \Pr[X \notin H \text{ and } Y = f(X) \text{ and } \delta(\mathrm{Enc}_{\mathrm{BSC}}(X \circ f(X)), Z) \leq 2p + o(1))].$$

By the HTS property, A is unlikely to output an $X \notin H$ such that $Y = f(X)$. Furthermore, for every $x \in H$,

$$\Pr[\delta(\mathrm{Enc}_{\mathrm{BSC}}(x \circ f(x)), Z) \leq 2p + o(1))] \text{ is exponentially small.}$$

This is because $p < \frac{1}{4}$, and a random string $Z \leftarrow U_n$ is unlikely to have relative distance significantly smaller than half from the fixed string $\mathrm{Enc}_{\mathrm{BSC}}(x \circ f(x))$. We have that $|H| \leq h = 2^{\delta n}$ and by choosing the constant $\delta > 0$ to be sufficiently small (which we are allowed in Theorem 1.3) we can do a union bound over all $x \in H$ and argue that it is unlikely that $\mathrm{Dec}'_{\mathrm{BSC}}(Z \oplus C(Z)) \neq fail$.

1.3.6 Overview of an explicit construction of SS-non-malleable codes

The second component of [SS22] that we will discuss is an "SS-non-malleable code" (introduced in [SS22]). This notion is related to the well studied notion of non-malleable codes (defined by Dziembowski, Pietrzak and Wichs [DPW18]) but has a twist that makes it suitable to the application of constructing stochastic codes.

Background on non-malleable codes. Loosely speaking, non-malleable codes are stochastic codes that are designed to work against channels $C : \{0,1\}^n \to \{0,1\}^n$, which when given a codeword z, produce a "corrupted word" $C(z)$. However, in contrast to earlier settings that we discussed, there will be no bound on the number of errors that C can induce. This means that C can erase the codeword and replace it with another string, and we cannot expect the decoding algorithm to produce the original message m. Instead, the goal is to show that C is unable to make the decoding algorithm produce a message $\bar{m} \neq m$ that is "related" to m.

SS-non-malleable codes. The construction of stochastic codes for poly-size circuits of [SS22] considers a case where the non-malleable stochastic code is used to encode a random seed $S \leftarrow \{0,1\}^d$ for a PRG $G : \{0,1\}^d \to \{0,1\}^n$ (as we have already discussed in Section 1.3.4).

Our goal is to design a stochastic code (that is algorithms $\mathrm{Enc}_{\mathrm{ssnm}}(S,R)$ and $\mathrm{Dec}_{\mathrm{ssnm}}(V)$) that is "SS-non-malleable" (here, "SS" stands for "small set"). This formally means that for every circuit C of size n^c, there exists a small set $H \subseteq \{0,1\}^d$ such that

$$\Pr_{S,R}[\mathrm{Dec}_{\mathrm{ssnm}}(C(G(S), \mathrm{Enc}_{\mathrm{ssnm}}(S,R))) \notin H \cup \{S\}] \leq \frac{1}{n^c}.$$

See Section 5 for a more precise definition. In words, it is required that a size n^c circuit C that sees $G(S)$ and $\mathrm{Enc}_{\mathrm{ssnm}}(S,R)$ (for uniformly chosen S,R) cannot prepare a string V that makes the decoding output a seed $\bar{S}$ that is not in $H \cup \{S\}$. Loosely speaking, this means that w.h.p. either $\bar{S} = S$ (and the correct seed is decoded) or the decoded seed $\bar{S}$ is in some small set H that was known in advance.

Indeed, the "definitional methodology" behind this definition is similar in spirit to our notion of HTS, and our definition of HTS is inspired by, and based on, this definition from [SS22].

Usefulness of SS-non-malleable codes. Recall that in Section 1.3.4 we explained that the construction of [SS22] makes use of a random seed S for the PRG, and $Z = \mathrm{Enc}_{\mathrm{BSC}}(m) \oplus G(S)$, when preparing a codeword for the final stochastic code. SS-non-malleable codes guarantee that an adversary that sees the codeword of the final stochastic code (which will be set up as a function of $\mathrm{Enc}_{\mathrm{ssnm}}(S,R)$ and $Z = \mathrm{Enc}_{\mathrm{BSC}}(m) \oplus G(S)$) cannot make the decoding algorithm $\mathrm{Dec}_{\mathrm{ssnm}}$ decode to a seed $\bar{S} \neq S$ that is "strongly correlated" with S.

On a more technical level, this notion can be used to guarantee that in this experiment, $G(S) \oplus G(\bar{S})$ is w.h.p. either the all zero's string (in case $\bar{S} = S$) or a pseudorandom string (which would follow if S and $\bar{S}$ are really uncorrelated in the information theoretic sense).[18] Shaltiel and Silbak [SS22] used such SS-non-malleable codes (in fact, for a somewhat stronger definition) in their construction of stochastic codes for poly-size circuits.

SS-non-malleable codes from non-malleable codes and an HTS. We start by noting that SS-non-malleable codes imply some sort of "hardness to sample" (and this observation led us to study HTS). More precisely, a possible attack on SS-non-malleable codes is to completely ignore the input $G(S), \mathrm{Enc}_{\mathrm{ssnm}}(S,R)$, and sample an encoding $\mathrm{Enc}_{\mathrm{ssnm}}(S',R')$ for uniformly chosen S',R' that are independent of S,R.

[18] We remark that to the best of our knowledge, other notions of non-malleable codes in the literature do not have this property. Loosely speaking, while one can use the pseudorandomness of G (in the case that G is seed extending, meaning that $S, G(S)$ is pseudorandom) together with the standard notion of non-malleability to argue that in the experiment above, S and $\bar{S}$ are computationally close to being independent. This does not give that they are statistically independent, and does not seem to imply that w.h.p. $G(S) \oplus G(\bar{S})$ is either the all zeros string, or pseudorandom.

This breaks the SS-non-malleability property, as it is unlikely that a uniform S' will land in $H \cup \{S\}$, for any small set H. This loosely means that $\mathrm{Enc}_{\mathrm{ssnm}}$ must be hard to sample on low min-entropy distributions.

We will now show how to construct an SS-non-malleable code using an HTS. We will require a (standard) non-malleable code for poly-size circuits (See Definition 5.2). Recently, Ball, Dachman-Soled and Loss [BDL22] gave a construction of non-malleable codes for poly-size circuits that is based on the assumption that E is hard for exponential size nondeterministic circuits. This means that we can get a non-malleable code $(\mathrm{Enc}_{\mathrm{nm}}, \mathrm{Dec}_{\mathrm{nm}})$ for size n^c circuits, under this hardness assumption.

We can also use this hardness assumption to instantiate Theorem 1.2 and obtain a $d^{O(1)}$-HTS $f : \{0,1\}^d \to \{0,1\}^{\mathrm{poly}(d)}$ for size n^c circuits. (This is not immediately clear from the way Theorem 1.2 is stated, and follows from a more general statement that appears in Theorem 3.4).

We will now use the same approach that we used in the previous construction of evasive BSC codes. Namely, we will define $\mathrm{Enc}_{\mathrm{ssnm}}(S, R) = \mathrm{Enc}_{\mathrm{nm}}(S \circ f(S), R)$. (Here, the rate is not that important as we are encoding a $d = O(\log n)$ bit string, and so, we can afford the HTS f of Theorem 1.2 that will have output length $\mathrm{poly}(d) > d$ but still much smaller than n). We will once again define $\mathrm{Dec}_{\mathrm{ssnm}}(V)$ by applying $\mathrm{Dec}_{\mathrm{nm}}(V)$ to obtain $X \circ Y$, and outputting X if and only if $Y = f(X)$.

The intuition is that the non-malleable code $\mathrm{Enc}_{\mathrm{nm}}$ is taking care of an adversary C that sees $\mathrm{Enc}_{\mathrm{nm}}(S \circ f(S), R)$, and prevents such an adversary from leading the decoding to a seed $\bar{S}$ that is correlated with S. Furthermore, the HTS is taking care of the attack described earlier (in which C ignores its input).

More formally, we can consider the "simulator" A^C of the adversary C (that is promised in the standard definition of non-malleability). When thinking of A^C as a sampling circuit, by the security of the HTS, we indeed know that this circuit A^C has a set H of size h, so that it is unlikely for A^C to produce a message $X \circ Y$ such that $X \notin H$ and $Y = f(X)$. This argument shows that we obtain SS-non-malleability, in case C does not see $G(S)$.

We still need to consider adversaries C that in addition to $\mathrm{Enc}_{\mathrm{ssnm}}(S, R)$ also see $G(S)$. We argue that if G is a "seed-extending" PRG, meaning that the distribution $(S, G(S))$ is computationally indistinguishable from (S, U_n), and if furthermore, G fools circuits that are sufficiently large to run $\mathrm{Enc}_{\mathrm{nm}}, \mathrm{Dec}_{nm}, C$ and f, then seeing the additional string $G(S)$ does not help the adversary C.

For this purpose we consider a circuit $D(S, Z)$ that computes $\mathrm{Enc}_{\mathrm{ssnm}}(S, R)$ and feeds $(Z, \mathrm{Enc}_{\mathrm{ssnm}}(S, R))$ to the adversary C. Note that if $Z = G(S)$, then this corresponds to the actual experiment that we are interested in. In the case that $Z = U_n$, we may as well pretend that C is randomized, and sampled $Z \leftarrow U_n$ on its own (and we have already argued SS-non-malleability against such adversaries, and so such an adversary C has a small set H as guaranteed by the SS-non-malleability definition).

The circuit D will continue to simulate the experiment. It will decode (using $\mathrm{Dec}_{\mathrm{ssnm}}$) to obtain a string $\bar{S}$. D will then check whether $\bar{S} \in H \cup \{S\}$. (For this we will hardwire the set H, which is a function of C, to the circuit D). As D cannot distinguish between $(S, G(S))$ and (S, U_n), we obtain that the set H (defined for the case that C does not see $G(S)$) is also suitable for the case that C does see $G(S)$, and we obtain full fledged SS-non-malleability.[19]

1.4 More related work on codes for computationally bounded channels

1.4.1 Stochastic codes for weaker classes of computationally bounded channels

Below is a brief survey the state of the art on stochastic codes for some other classes of computationally bounded channels.

[19] It is crucial to note that while D is a pretty large circuit, we crucially rely on the fact that D *does not* need to compute the PRG G. This is because a circuit D that can compute G can easily distinguish between $(S, G(S))$ and (S, U_n).

Additive channels. The seminal paper of Guruswami and Smith [GS16] considered (amongst other classes) the class of additive channels. This class (denoted by Add_p) contains all *constant* functions $C \in \mathrm{Ham}_p$. This means that an additive channel $C : \{0,1\}^n \to \{0,1\}^n$ has a predetermined noise vector $e \in \{0,1\}^n$ of Hamming weight at most p, and the channel C uses this noise vector *regardless* of its input. Guruswami and Smith [GS16] gave explicit constructions of stochastic codes with rate $1 - H(p)$ for Add_p. This result holds for every $0 \leq p < \frac{1}{2}$ (and in particular, this shows that the capacity is positive for $p > \frac{1}{4}$).

Online channels with small space. The class of space-bounded channels (denoted by Spc_p^s) is the class of all $C \in \mathrm{Ham}_p$, where C reads its input in one pass, using space s, and produces its i'th output bit before reading the $(i+1)$'th bit. Guruswami and Smith [GS16] showed that there do not exist stochastic codes with positive rate for $\mathrm{Spc}_p^{\log n}$ if $p > \frac{1}{4}$.

Shaltiel and Silbak [SS21b] (building on earlier work [GS16, SS21a, KSS19] that considered list-decoding) gave explicit stochastic codes for $\mathrm{Spc}_p^{n^{\Omega(1)}}$ with rate $1 - H(p)$ for every $0 \leq p < \frac{1}{4}$. For $\frac{1}{4} \leq p \leq \frac{1}{2}$, [KSS19] gave explicit stochastic codes with rate $1 - H(p)$ for $\mathrm{Spc}_p^{n^{\Omega(1)}}$ that are *list-decodable*. In fact, these stochastic codes work even against channels that are allowed to read the bits of the codeword in an order of their choice, and the encoding and decoding algorithms run in quasilinear time.

Causal channels. This is the class Spc_p^n in which there is no space restriction on the channel, but the channel has to decide whether to flip the i'th bit, based only on the first i bits of the codeword. The works of Dey, Jaggi, Langberg, and Sarwate [DJLS13] and Chen, Jaggi and Langberg [CJL15], determine the capacity of such channels. This capacity is $1 - H(p)$ for $p \leq p_0 \approx 0.0804$, and is strictly smaller than $1 - H(p)$ for $p_0 < p < \frac{1}{4}$. This is achieved by non-explicit constructions, and no explicit constructions are known.

1.4.2 Other scenarios of codes for computationally bounded channels

The notion of computationally bounded channels was also studied in other setups. We mention some of these works below.

Shared private randomness. We start with the notion of codes with "shared private randomness". While this setup was considered before the notion of stochastic codes, in this paper, it is natural to view it as a version of stochastic codes in which the decoding algorithm *does* receive the randomness S chosen by the encoding algorithm. This corresponds to a standard symmetric cryptography setup in which honest parties (the encoder and decoder) share a uniform private key S, and the bad party (the channel) does not get the key. Lipton [Lip94] and following work (see [Smi07] for more details) gave explicit constructions of uniquely decodable codes against computationally bounded channels, in this setup, with rate approaching $1 - H(p)$, under cryptographic assumptions.

Note that the setup of stochastic codes is lighter. The encoder and decoder do not need to share a private random key. Moreover, a fresh key can be chosen on the spot every time the encoder encodes a message.

A related notion of "private codes" was studied by Langberg [Lan04]. This is also in the setup of shared private randomness. Here channels are computationally unbounded, codes are existential rather than explicit, and have rate approaching $1 - H(p)$. The focus is on minimizing the length of the shared key. Langberg provides asymptotically matching upper and lower bounds of $\Theta(\log n + \log(1/\nu))$, on the amount of randomness that needs to be shared for unique decoding in this setup, where ν is the error parameter.

Public key setup. Micali et al. [MPSW10] considered computationally bounded channels, and a cryptographic "public key setup". Their focus is to use cryptographic tools to convert a given (standard) explicit list-decodable code into an explicit uniquely decodable code with the same rate (so that the constructed code works in this specific public key setup).

Organization of this paper

In Section 2 we give preliminaries, and formal definitions. We also state the many results that we use in our construction. In Section 3 we give a formal definition of HTS, and prove Theorems 1.2 and 1.3. In Section 4 we give a formal definition of evasive BSC codes (defined in [SS21b, SS22]) and give an explicit construction of such codes under hardness assumptions. In Section 5 we give a formal definition of SS-non-malleable codes (for a variant that is slightly weaker than the one defined in [SS22]), and give an explicit construction of such codes under hardness assumptions. In Section 6 we prove Theorem 1.1, giving a construction of stochastic codes for poly-size circuits under hardness assumptions. In Section 7 we present some open problems.

2 Preliminaries and ingredients

In this section we give formal definitions of the notions and ingredients used in the construction. We also cite previous results from coding theory and pseudorandomness that are used in the construction.

General notation. We use $[n]$ to denote $\{1,\ldots,n\}$. We sometimes use the notation $O_\lambda(\cdot)$ to emphasize that the constant hidden in the $O(\cdot)$ notation may depend on λ.

Probability distributions. We use U_n to define the uniform distribution on n bits. The ***statistical distance*** between two distributions P, Q over Ω is $\Delta(P,Q) = \max_{A\subseteq\Omega} |P(A) - Q(A)|$. Given a distribution P, we use $X \leftarrow P$ to denote the experiment in which the random variable X is chosen according to P. For a set A, we use $X \leftarrow A$ to denote the experiment in which X is chosen uniformly from A. We use $X_1,\ldots,X_n \leftarrow A$ to denote the experiment in which n variables are chosen uniformly from A with replacement.

Shannon's entropy function. We use $H(p)$ to denote the ***Shannon binary entropy*** function: $H(p) = p\cdot\log(1/p) + (1-p)\cdot\log(1/(1-p))$. It is standard that the derivative $H'(p)$ in the interval $(0,\frac{1}{2})$ is decreasing, and satisfies $H'(p) \le \log\frac{1}{p}$. This implies that for $0 < p < \frac{1}{2}$, if $p(1+\delta) < \frac{1}{2}$ then:

$$H(p(1+\delta)) \le H(p) + \delta\cdot p\cdot H'(p) \le H(p) + \delta. \tag{2}$$

We will also rely on the standard fact that $H(\frac{1}{2}-\epsilon) = 1 - O(\epsilon^2)$.

Hamming distance and weight. The ***Hamming weight*** of $x \in [q]^n$ is $\mathbf{WT}(x) = |\{i : x_i \neq 0\}|$. The ***relative Hamming weight*** of x is $wt(x) = \frac{\mathbf{WT}(x)}{n}$. The ***Hamming distance*** between $x, y \in [q]^n$ is $\Delta(x,y) = |\{i : x_i \neq y_i\}|$. The ***relative Hamming distance*** between $x, y \in [q]^n$ is $\delta(x,y) = \frac{\Delta(x,y)}{n}$.

Channels. Let BSC_p denote the distribution over n bits, where bits are i.i.d., and each bit has probability p to be one. We will sometimes abuse the notation and think of BSC_p as a probabilistic procedure that on input $z \in \{0,1\}^n$, produces the distribution BSC_p. Let Ham_p denote the class of functions $C : \{0,1\}^n \to \{0,1\}^n$ such that for every $z \in \{0,1\}^n$, $\mathsf{wt}(C(z)) \le p$. Let Ckt^s_p be the class of all functions in Ham_p that can be computed by size s circuits.

2.1 Circuits, hardness assumptions and pseudorandom generators

2.1.1 Various types of circuits

We formally define the circuit types that will be used in this paper.

Definition 2.1 (nondeterministic circuits, oracle circuits and Σ_i-circuits)**.** *A* randomized *circuit C has additional wires that are instantiated with uniform and independent bits. A* nondeterministic *circuit C has additional "nondeterministic input wires". We say that the circuit C evaluates to 1 on x iff there exist an assignment to the nondeterministic input wires that makes C output 1 on x. An oracle circuit $C^{(\cdot)}$ is a circuit which in addition to the standard gates uses an additional gate (which may have large fan in). When instantiated with a specific boolean function A, C^A is the circuit in which the additional gate is A. Given a boolean function $A(x)$, an A-circuit is a circuit that is allowed to use A gates (in addition to the standard gates). An $A_{||}$-circuit is a circuit that makes nonadaptive queries to its oracle A. (Namely, on every path from input to output, there is at most a single A gate). An NP-circuit is a SAT-circuit (where SAT is the satisfiability function) a Σ_i-circuit is an A-circuit where A is the canonical Σ_i^P-complete language. We use the notation NP$_{||}$-circuit and $\Sigma_{i||}$-circuit for the case where the queries are nonadaptive. The size of all circuits is the total number of wires and gates.*[20]

Remark 2.2 (Adding an oracle to relativizing statements)**.** *Most results in complexity theory relativize, and remain true if we add NP or Σ_i^P oracles. In such cases we will state the original result and say that the corollary (with oracles) follows by relativization.*

2.1.2 Hardness assumptions

We now define several types of hard functions and hardness assumptions.

Definition 2.3 (hard functions)**.** *We say that a function $f : \{0,1\}^m \to \{0,1\}^{m'}$ is ϵ-***hard** *for a class $\mathcal{C}$, if for every $C \in \mathcal{C}$, such that $C : \{0,1\}^m \to \{0,1\}^{m'}$,*

$$\Pr_{X \leftarrow U_m}[C(X) = f(X)] < \epsilon.$$

We say that f is **hard** *for $\mathcal{C}$ if f is 1-hard for $\mathcal{C}$.*

Assumption 2.4 (E is hard for exponential size circuits)**.** *We say that "E is hard for exponential size circuits of type X" if there exists a language L in $E = DTIME(2^{O(n)})$ and a constant $\beta > 0$, such that for every sufficiently large n, the characteristic function of L on inputs of length n is hard for circuits of size $2^{\beta n}$ of type X.*

Shaltiel and Umans [SU06] showed that certain hardness assumptions against nondeterministic circuits yield the same hardness against NP$_{||}$-circuits.

Theorem 2.5 (Downwards collapse theorem [SU06])**.** *If E is hard for exponential size nondeterministic circuits then E is hard for exponential size NP$_{||}$-circuits.*

2.1.3 Pseudorandom generators

We need the following standard definition of pseudorandom distributions and generators.

Definition 2.6 (Pseudorandom generators)**.** *A distribution X on n bits is ϵ-***pseudorandom** *for a class $\mathcal{C}$ of functions, if for every $C \in \mathcal{C}$, $|\Pr[C(X) = 1] - \Pr[C(U_n)] = 1]| \leq \epsilon$. A function $G : \{0,1\}^d \to \{0,1\}^n$ is an ϵ-***PRG** *for $\mathcal{C}$ if $G(U_d)$ is ϵ-pseudorandom for $\mathcal{C}$. G is* **seed-extending** *if the function $G'(x) = x \circ G(x)$ is an ϵ-PRG for $\mathcal{C}$.*

[20] An alternative approach is to define our model using the Karp-Lipton notation for Turing machines with advice. For $s \geq n$, a size $s^{\Theta(1)}$ deterministic circuit is equivalent to DTIME$(s^{\Theta(1)})/s^{\Theta(1)}$, a size $s^{\Theta(1)}$ nondeterministic circuit is equivalent to NTIME$(s^{\Theta(1)})/s^{\Theta(1)}$, a size $s^{\Theta(1)}$ NP-circuit is equivalent to DTIME$^{\text{NP}}(s^{\Theta(1)})/s^{\Theta(1)}$, a size $s^{\Theta(1)}$ nondeterministic NP-circuit is equivalent to NTIME$^{\text{NP}}(s^{\Theta(1)})/s^{\Theta(1)}$, and a size $s^{\Theta(1)}$ Σ_i-circuit is equivalent to DTIME$^{\Sigma_i^P}(s^{\Theta(1)})/s^{\Theta(1)}$.

The classical result of Impagliazzo and Wigderson [IW97] gives a PRG for poly-size circuits, under the assumption that E is hard for exponential size circuits.

Theorem 2.7 (PRGs from hardness assumptions [IW97])**.** *If E is hard for exponential size circuits then for every constant $c > 1$, there exists a constant $a > 1$ such that for every sufficiently large n, there is a $G : \{0,1\}^{a \cdot \log n} \to \{0,1\}^n$ that is a seed-extending $\frac{1}{n^c}$-PRG for circuits of size n^c. Furthermore, G is computable in time* $\mathrm{poly}(n^c)$.

We will also use PRGs for $\mathrm{NP}_{||}$-circuits. Klivans and van-Melkebeek observed that the proof of Impagliazzo and Wigderson relativizes. This means (by Remark 2.2) that one can get PRGs for NP-circuits by assuming that E is hard form exponential size NP-circuits. A more careful inspection of the argument shows that one can get PRGs for $\mathrm{NP}_{||}$-circuits by assuming that E is hard form exponential size $\mathrm{NP}_{||}$-circuits, and the latter assumption can be relaxed using Theorem 2.5. Altogether, this gives the following:

Theorem 2.8 (PRGs for $\mathrm{NP}_{||}$ from hardness assumptions [IW97, KvM02, SU06])**.** *If E is hard for exponential size nondeterministic circuits then for every constant $c > 1$, there exists a constant $a > 1$ such that for every sufficiently large n, there is a $G : \{0,1\}^{a \cdot \log n} \to \{0,1\}^n$ that is a seed-extending $\frac{1}{n^c}$-PRG for $\mathrm{NP}_{||}$-circuits of size n^c. Furthermore, G is computable in time* $\mathrm{poly}(n^c)$.

2.2 Various kinds of error-correcting codes

In this section we give formal definitions of some of the various notions of error correcting codes used in this paper. A code is a pair $(\mathrm{Enc}, \mathrm{Dec})$ of encoding and decoding maps, and different notions are obtained by considering the requirements on the decoding algorithm.

2.2.1 Standard notions of error correcting codes

We start by giving definitions of error correcting codes that covers the standard cases of Hamming channels and binary symmetric channels.

Definition 2.9 (Codes for Shannon and Hamming channels)**.** *Let k, n be parameters and let* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ *and* $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k$ *be functions. We say that* $(\mathrm{Enc}, \mathrm{Dec})$*:*

- **decodes from t errors**, *if for every $m \in \{0,1\}^k$ and every $v \in \{0,1\}^n$ with $\Delta(\mathrm{Enc}(m), v) \leq t$, $\mathrm{Dec}(v) = m$.*
- **decodes from a distribution** *P, with success probability $1 - \nu$, if P is a distribution over $\{0,1\}^n$, $0 \leq \nu \leq 1$, and for every $m \in \{0,1\}^k$, $\Pr_{e \leftarrow P}[\mathrm{Dec}(\mathrm{Enc}(m) \oplus e) = m] \geq 1 - \nu$.*

The rate of the code is the ratio of the message length and output length of Enc*, where both lengths are measured in bits. That is the rate $R = \frac{k}{n}$.*

The code is **explicit** *if both encoding and decoding run in polynomial time. (Naturally, this makes sense only for a family of encoding and decoding functions with varying block length n, message length $k(n)$).*

The notion of "decoding from errors" corresponds to *Hamming channels*, where the decoding algorithm needs to decode (or list-decode) from a certain distance. We remark that it is standard that a code is decodable from t errors if and only if the Hamming distance between every two codewords is at least $2t + 1$.

The notion of decoding from P covers the case of *BSC channels*, where P is taken to be the distribution BSC_p of n i.i.d. bits where each bit has probability p to be one.

2.2.2 Stochastic codes for a class of channels

In this section we give a precise formal definition of the notion of stochastic codes for a class of channels (that was already explained in the introduction).

Definition 2.10 (Stochastic codes for channels)**.** *Let k, n, d be parameters and let* $\mathrm{Enc} : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^n$*, and* $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k$ *be functions. Let $\mathcal{C}$ be a class of functions from n bits to n bits. We say that* $(\mathrm{Enc}, \mathrm{Dec})$ *is a stochastic code for "channel class" $\mathcal{C}$, with success probability $1 - \nu$, if for every $m \in \{0,1\}^k$ and every $C \in \mathcal{C}$, setting $X = \mathrm{Enc}(m, U_d)$, we have that*

$$\Pr[\mathrm{Dec}(X \oplus C(X)) = m] \geq 1 - \nu.$$

The rate of the code is the ratio of the message length and output length of Enc*, where both lengths are measured in bits. That is the rate $R = \frac{k}{n}$.*

The code is **explicit** *if both encoding and decoding run in polynomial time. (Naturally, this makes sense only for a family of encoding and decoding functions with varying block length n, message length $k(n)$ and randomness $d(n)$).*

A **Monte-Carlo** *stochastic code with success $1 - \nu$ for a class $\mathcal{C}$, that uses q bits of Monte-Carlo randomness, with Monte-Carlo error $\eta > 0$, is a pair of functions* $\mathrm{Enc} : \{0,1\}^q \times \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^n$ *and* $\mathrm{Dec} : \{0,1\}^q \times \{0,1\}^n \to \{0,1\}^k$*, such that with probability $1 - \eta$ over choosing $y \leftarrow U_q$, the pair of functions* $\mathrm{Enc}_y(m, s) = \mathrm{Enc}(y, m, s)$ *and* $\mathrm{Dec}_y(v) = \mathrm{Dec}(y, v)$ *form a stochastic-code for $\mathcal{C}$ with success $1 - \nu$.*

A Monte-carlo stochastic code is **explicit** *if* Enc, Dec *run in time polynomial in n, and q is a polynomial in n. (Naturally, this makes sense only for a family of encoding and decoding functions with varying block length n, message length $k(n)$, seed length $d(n)$ and Monte-Carlo randomness length $q(n)$).*[21]

2.2.3 Stochastic codes with error-correcting and pseudorandomness properties

A key component in all previous constructions of stochastic codes for computationally bounded channels is a stochastic code that combines both error-correcting and pseudorandomness properties. This notion was introduced by Guruswami and Smith [GS16]. The high level approach is that constructing such codes with very small rate (in our case, the rate will be about $\log n/n$) is helpful in constructing the final code (loosely speaking, these codes play a role that resembles that of "inner codes" in the final construction). Shaltiel and Silbak [SS21a] relaxed the notion used by Guruswami and Smith to one that is easier to construct, and is still sufficient for the approach. For completeness, we include both notions (the original one is termed "strongly" and the relaxed one is termed "weakly").

Definition 2.11. *Let k, n, d be parameters and let* $\mathrm{Enc} : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^n$ *and* $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k$ *be functions. We say that* $(\mathrm{Enc}, \mathrm{Dec})$ *is a stochastic code that is:*

- *L-***weakly list-decodable** *with* **radius** *p if for every $y \in \{0,1\}^n$,* $\mathrm{Dec}(y)$ *produces a list of at most L messages, that contains all messages $m \in \{0,1\}^k$ for which there exists $r \in \{0,1\}^d$ such that $\delta(y, \mathrm{Enc}(m, r)) \leq p$.*
- *We replace "weakly" with "***strongly***" if* Dec *is required to produce a list of at most L pairs (m, r) that contains all pairs $(m, r) \in \{0,1\}^k \times \{0,1\}^d$ such that $\delta(y, \mathrm{Enc}(m, r)) \leq p$.*

[21] An alternative view of Monte-Carlo constructions (that is sometimes preferable) is that a Monte-Carlo construction is a randomized algorithm that tosses $q(n)$ coins, and produces circuits Enc, Dec, such that with probability $1 - \eta(n)$, the obtained circuits have the required property.

- ϵ-**pseudorandom** *for a class $\mathcal{C}'$ of functions from n bits to one bit, if for every message $m \in \{0,1\}^k$, $\mathrm{Enc}(m, U_d)$ is ϵ-pseudorandom for $\mathcal{C}'$.*

We will rely on the following Theorem by Shaltiel and Silbak [SS21a].

Theorem 2.12 (inner stochastic code for poly-size circuits [SS21a])**.** *If E is hard for exponential size circuits then for every constant $0 \leq p < \frac{1}{2}$, $c > 1$ and $a > 0$ there exist constants L, b, q such that for every sufficiently large n, there is a stochastic code $(\mathrm{Enc}, \mathrm{Dec})$ where $\mathrm{Enc} : \{0,1\}^{a \cdot \log n} \times \{0,1\}^{b \cdot \log n} \to \{0,1\}^{q \cdot \log n}$ is:*

- *L-weakly list decodable from radius p.*
- *$\frac{1}{n^c}$-pseudorandom for size n^c circuits.*

Furthermore, the code is explicit. Specifically, $\mathrm{Enc}, \mathrm{Dec}$ are computable in time $\mathrm{poly}(n^c)$, where the polynomial depends on p, a and the constant $\beta > 0$ hidden in the hardness assumption.

2.2.4 Reed-Solomon list-decoding

We will make use of the celebrated list-decoding algorithm of Sudan [Sud97] and Guruswami and Sudan [GS99].

Theorem 2.13 (Reed-Solomon list-decoding [Sud97, GS99])**.** *For every q that is a power of 2, and every $k \leq n \leq q$, let $S = \{x_1, \ldots, x_n\}$ be a subset of $\mathbb{F}_q$ of size n, and let $\mathrm{Enc}_{\mathrm{RS}} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ be the Reed-Solomon encoding map (namely $\mathrm{Enc}_{\mathrm{RS}}(a_0, \ldots, a_{k-1})_i = \sum_{0 \leq j < k} a_j x_i^j$). This map can be computed in time $\mathrm{poly}(n, \log q)$, and there is a polynomial time algorithm that if $a \geq \sqrt{k \cdot t}$, given t distinct pairs $(i_1, y_1), \ldots, (i_t, y_t) \in [n] \times \mathbb{F}_q$, the algorithm produces a list of size at most $\leq \frac{2t}{a}$ of all $m \in \mathbb{F}_q^k$ such that $|\{i : \mathrm{Enc}_{\mathrm{RS}}(m)_i = y_i\}| \geq a$. We can view $\mathrm{Enc}_{\mathrm{RS}}$ as a function $\mathrm{Enc}_{\mathrm{RS}} : \{0,1\}^{k \log q} \to (\{0,1\}^{\log q})^n$.*

2.3 Standard definitions and classical results from complexity theory

2.3.1 Samplable distributions

We will use the following definition and notation for samplable distributions.

Definition 2.14 (Sampling procedures and samplable distributions)**.** *For a function $A : \{0,1\}^r \to \{0,1\}^n$, we use $Z \leftarrow A$ to denote the experiment in which $W \leftarrow U_r$, and $Z = A(W)$, and say that Z is* **sampled** *by A. We say that the distribution Z is* **samplable** *by a class $\mathcal{C}$ of functions, if there exists $A \in \mathcal{C}$ that samples Z.*

2.3.2 Approximate counting and uniform sampling of NP witnesses

We use the classical result on approximate counting and uniform sampling of NP-witnesses [Sto83, Sip83, JVV86, BGP00], which we state below in a way that is convenient for our application.

Definition 2.15 (relative approximation)**.** *We say that a number p is an ϵ-relative approximation to q if $(1 - \epsilon) \cdot p \leq q \leq (1 + \epsilon) \cdot p$, and an ϵ-additive approximation to q if $|p - q| \leq \epsilon$.*

It is useful to note that if $0 \leq p \leq 1$ is an ϵ-relative approximation to q, then it is also an additive approximation to q. For $\epsilon \leq \frac{1}{2}$, we also have the following: If p is an ϵ-relative approximation to q, then q is an $O(\epsilon)$-relative approximation to p. If p is an ϵ-relative approximation to q and q is an ϵ-relative approximation to w, then p is an $O(\epsilon)$-relative approximation to w. If p' is an ϵ-relative approximation to p and q' is an ϵ-relative approximation to q, then a p'/q' is an $O(\epsilon)$-relative approximation to p/q. (The last property does not hold if we replace relative approximations with additive approximations).

Theorem 2.16 (approximate counting [Sto83, Sip83, JVV86]). *For every sufficiently large s, and every $\epsilon > 0$, there is a size $\mathrm{poly}(s/\epsilon)$ $NP_{||}$-circuit that given a size s circuit C, outputs an ϵ-relative approximation of $|\{x : C(x) = 1\}|$.*

Theorem 2.17 (uniform sampling [JVV86, BGP00]). *For every sufficiently large s, and every $\delta > 0$, there is a size $\mathrm{poly}(s, \log(1/\delta))$ randomized $NP_{||}$-circuit A that given a size s circuit $C : \{0,1\}^n \to \{0,1\}$, outputs a value in $\{0,1\}^n \cup \perp$ such that $\Pr[A(C) = \perp] \leq \delta$ and the distribution $(A(C)|A(C) \neq \perp)$ is uniform over $\{x : C(x) = 1\}$.*

Regarding the formulation of Theorems 2.16 and 2.17. The formulation in the two theorems only requires that the tasks be achieved by (nonuniform) circuits. The classical results in this area, are in fact stronger. Theorem 2.17 holds for A that is a randomized uniform algorithm with an NP oracle (which is stronger than the statement we give here). Theorem 2.16 holds for a counting procedure that is a randomized uniform algorithm with an NP oracle. Here, we state it for a circuit (which is nonuniform, and non-randomized). This immediately follows by Adleman's proof proof that extends to BPP is in P/poly) (which extends to show that $\mathrm{BPP}^{\mathrm{NP}}_{||}$ is in $\mathrm{P}^{\mathrm{NP}}_{||}$/poly.

Another difference in the our statements of Theorems 2.16 and 2.17 (compared to the way they are usually stated) is that we state them with $\mathrm{NP}_{||}$-circuits, rather than with NP-circuits. The fact that the queries to the NP oracle in these classical results can be made nonadaptive follows from a more careful implementation of the classical algorithms for approximate counting, and uniform sampling, given by Shaltiel and Umans [SU06].

2.4 Extractors, dispersers, samplers and list-recoverable codes

2.4.1 Averaging Samplers

The reader is referred to Goldreich's survey [Gol97] on averaging samplers.

Definition 2.18 (Averaging Samplers). *A function $\mathrm{Samp} : \{0,1\}^n \to (\{0,1\}^m)^t$ is an (ϵ, δ)-***Sampler** *if for every $f : \{0,1\}^m \to [0,1]$,*

$$\Pr_{(z_1,\ldots,z_t)\leftarrow \mathrm{Samp}(U_n)}[|\frac{1}{t}\sum_{i\in[t]} f(z_i) - \frac{1}{2^m}\sum_{x\in\{0,1\}^m} f(x)| > \epsilon] \leq \delta.$$

A sampler has **distinct samples** *if for every $x \in \{0,1\}^n$, the t elements in $\mathrm{Samp}(x)$ are distinct.*

Zuckerman [Zuc97] showed that extractors can be viewed as samplers. Moreover, "strong extractors" translate into samplers with distinct samples. Using this connection, and the competitive extractor constructions of Guruswami, Umans and Vadhan [GUV07], we obtain the following sampler. (In fact, the construction of [GUV07] translates into a sampler with much better parameters than the one cited here).

Theorem 2.19. *For every constant $c_1 \geq 1$ there exist constants $c_2 > 1$ and $c_3 > 0$, such that for every sufficiently large m, and $2^{0.1\cdot m} \leq t(m) \leq 2^m$, there is a $(2^{-c_3\cdot m}, 2^{-c_1\cdot m})$-sampler with distinct samples $\mathrm{Samp} : \{0,1\}^{c_2\cdot m} \to (\{0,1\}^m)^{t(m)}$. Furthermore, Samp is computable in time $t(m)\cdot\mathrm{poly}(c_2\cdot\log m)$.*

We remark that previous work in this area [GS16, SS21a, KSS19, SS21b] used "expander based samplers", rather than "extractor based ones". We need a sampler with shorter seed than what was used in previous work, and this is why we choose this sampler.

For technical reasons that we explain in Section 6.2.1, rather than applying samplers on a short seed S, we will first apply a PRG G against n^c circuits. This means that we do not have a shortage of random bits (as the output of the PRG can be large). Consequently, in this paper it is not crucial that the seed of the sampler is short.

2.4.2 Seeded extractors and dispersers

In this section we define seeded extractors and dispersers.

Definition 2.20 (min-entropy). *For a distribution X over $\{0,1\}^n$, $H_\infty(X) := \min_x \log \frac{1}{\Pr[X=x]}$, where the minimum is taken over all strings x in the support of X.*

Definition 2.21 (Strong extractors and dispersers). *A function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k,ϵ)-strong extractor if for every distribution X over $\{0,1\}^n$ with $H_\infty(X) \geq k$, the distribution $Z = (Y, E(X,Y))$ where $Y \leftarrow U_d$ is ϵ close to U_{m+d}. E is a (k,ϵ) strong disperser if the support size of Z is at least $(1-\epsilon)\cdot 2^{m+d}$.*

2.4.3 List-recoverable codes

We will use the following definition of list-recoverable codes.

Definition 2.22 (List-recoverable codes). *A function $E : \{0,1\}^n \to (\{0,1\}^m)^D$ is (ℓ, L)-**list-recoverable** if for every $S_1, \ldots, S_D \subseteq \{0,1\}^m$, such that for every $i \in [D]$, $|S_j| \leq \ell$, the set*

$$\mathrm{List}_E(S_1, \ldots, S_D) = \{x \in \{0,1\}^n : \forall i \in [D] : E(x)_i \in S_i\},$$

is of size at most L.

We remark that the definition above is for an "errorless version" of list-recoverable codes. In the more general setting, the definition of the set $\mathrm{List}_E(S_1, \ldots, S_D)$ has an additional "agreement parameter", measuring the fraction of i, for which $E(x)_i \in S_i$.

Ta-Shma and Zuckerman [TZ04] observed that there is a tight connection between extractors and (the more general version) of list-recoverable codes. In the proposition below, we state a version of this connection for errorless list-recoverable codes and strong dispersers.

Proposition 2.23 (Strong dispersers give list-recoverable codes [TZ04]). *For a function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, we define $D = 2^d$, and $E' : \{0,1\}^n \to (\{0,1\}^m)^D$ by $E'(x)_i = E(x,i)$. If E is a strong (k,ϵ)-disperser, then E' is $(\ell, 2^k)$-list-recoverable for every $\ell < (1-\epsilon)\cdot 2^m$.*

Proof. If E' is not $(\ell, 2^k)$-list-recoverable, then there exist $S_1, \ldots, S_D \subseteq \{0,1\}^m$, such that each of the sets is of size $\leq \ell$, and the set $\mathrm{List}_E(S_1, \ldots, S_D)$ is of size at larger than 2^k. Let $T \subseteq \{0,1\}^d \times \{0,1\}^m$ be defined by: $T = \{(i,z) : z \notin S_i\}$. In particular, if we set X to be the uniform distribution on $\mathrm{List}_E(S_1, \ldots, S_D)$, we have that:

$$\Pr_{Y \leftarrow U_d}[(Y, E(X,Y)) \in T] = 0.$$

By definition, T is of size larger than $2^d \cdot (2^m - \ell)$. This means that the support of $Z = (Y, E(X,Y))$ is of size that is upper bounded by the size of the complement of T. Overall, we obtain that the support of Z is of size at most $2^d \cdot \ell$ which is a contradiction to the definition of a strong disperser if $\ell < (1-\epsilon)\cdot 2^m$. □

We now cite several explicit constructions of extractors and dispersers.

Theorem 2.24 (strong extractors for small k [LRVW03, GUV07]). *There exists a constant $c_1 > 1$ such that for every sufficiently large n, and every $k > c_1 \log n$, there is a strong $(k, \frac{1}{10})$-extractor $E : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\Omega(k)}$. Furthermore, E can be computed in time $\mathrm{poly}(n)$.*

Theorem 2.25 (Strong high-error dispersers with constant seed length [Zuc07]). *For every constant $\delta > 0$, there exists a constant $b > 1$ such that for every sufficiently large n, and every $\epsilon = \epsilon(n) > 0$ there is a strong $(\delta \cdot n, 1-\epsilon)$-disperser $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ where $d = \log D$ for $D = \frac{bn}{\log \frac{1}{1-\epsilon}}$ and $m = \frac{\delta n}{2}$. Furthermore, E can be computed in time $\mathrm{poly}(n)$.*

We remark that the result of Zuckerman [Zuc07] is stronger than the one we state here.

2.4.4 Relative error extractors for weakly recognizable distributions

We use deterministic (seedless) extractors for various types of distributions. We will use constructions of such extractors that have *relative error*.

Definition 2.26 (statistical distance with relative error)**.** *We say that a distribution Z on $\{0,1\}^m$ is* **ϵ-close to uniform with relative error** *if for every event $A \subseteq \{0,1\}^m$, $|\Pr[Z \in A] - \mu(A)| \leq \epsilon \cdot \mu(A)$ where $\mu(A) = |A|/2^m$.*

The following proposition is immediate.

Proposition 2.27. *For every distribution Z over $\{0,1\}^m$, the following conditions are equivalent:*

- *Z is ϵ-close to uniform with relative error.*
- *For every event $A \subseteq \{0,1\}^m$, $\mu(A)$ is an ϵ-relative approximation to $\Pr[Z \in A]$.*
- *for every $z \in \{0,1\}^m$, $\frac{1}{2^m}$ is an ϵ-relative approximation of $\Pr[Z = z]$.*

Note that if Z is ϵ-close to uniform with relative error, then it is also ϵ-close to uniform. However, we now also get that for every event A, $\Pr[Z \in A] \leq (1+\epsilon) \cdot \mu(A)$ and this implies that even if ϵ is large, events that are negligible under the uniform distributions cannot become noticeable under Z.

We now introduce a revised definition of deterministic extractors by replacing the requirement that the output is ϵ-close to uniform by the requirement that the output is close to uniform with relative error.

Definition 2.28 (deterministic extractor with relative error)**.** *Let $\mathcal{C}$ be a class of distributions over $\{0,1\}^n$. A function $E : \{0,1\}^n \to \{0,1\}^m$ is a (k,ϵ)-relative-error extractor for $\mathcal{C}$ if for every distribution X in the class $\mathcal{C}$ such that $H_\infty(X) \geq k$, $E(X)$ is ϵ-close to uniform with relative error.*

We will use extractors for weakly recognizable distributions. Recognizable distributions were defined by Shaltiel [Sha09], and the notion used here of "weakly recognizable distributions" was defined by Applebaum et al. [AASY15].

Definition 2.29 (weakly recognizable distributions [AASY15])**.** *We say that a distribution X on n bits is* **weakly recognizable** *by a class $\mathcal{C}$ of functions $C : \{0,1\}^n \to \mathbb{R}$, if there exists a function C in $\mathcal{C}$ such that for every $x \in \{0,1\}^n$, $\Pr[X = x] = \frac{C(x)}{\sum_{x \in \{0,1\}^n} C(x)}$.*[22]

The notion of weakly recognizable distribution is helpful as by Theorem 2.16, for every distribution X that is samplable by circuits of size s, and every $\epsilon > 0$, there is a distribution X' that is weakly recognizable by $\mathrm{NP}_{||}$-circuits of size $\mathrm{poly}(s, \log(1/\epsilon))$ and is ϵ-close to X in relative statistical distance. (For our application, we will need the additional power of extracting from distributions that are weakly recognizable by $\mathrm{NP}_{||}$-circuits, rather than just for samplable distributions.

In Section 3.4 we make use of relative error extractors for distributions that are weakly recognizable by $\mathrm{NP}_{||}$-circuits. We construct such extractors (based on the assumption that E is hard for exponential size nondeterministic circuits) in Section 3.5.2. As we explain in that section, our approach is based on a related construction by Kinne, Shaltiel and van-Melkebeek [KvMS12]. We also mention that there are known constructions by Applebaum et al. [AASY15] that achieve better parameters, but are based on stronger assumptions.

[22] In the definition above we don't set an a-priori bound on length of integers used. In this paper we will always have that $\mathcal{C}$ will be size s circuits (of a certain type) and the size bound implies an upper bound of s on length of integers output by C. Moreover, for such a choice of $\mathcal{C}$, we may as well assume that C is hardwired with the constant $\sum_{x \in \{0,1\}^n} C(x)$. This means that by allowing C to be slightly larger, we can obtain a circuit C' such that $C'(x) = \Pr[X = x]$ (meaning that X is "computable" by C').

2.5 Pseudorandmly chosen permutations

We will use a permutation $\pi : [n] \to [n]$ to "reorder" the bits of a string $x \in \{0,1\}^n$: The i'th bit in the rearranged string will be the $\pi(i)$'th bit in x. This is captured in the definition below.

Definition 2.30 (Permuting strings)**.** *Given a string $x \in \{0,1\}^n$ and a permutation $\pi : [n] \to [n]$. Let $\pi(x)$ denote the string $x' \in \{0,1\}^n$ with $x'_i = x_{\pi(i)}$.*

As in previous works in this area, we will need to consider a channel that produces its error pattern by permuting a fixed error pattern e using a random permutation.

Let $\mathrm{UniPerm}_n$ denote the uniform distribution on the set of permutation $\pi : [n] \to [n]$. We will omit n when it is clear from the context. Fix some poly-time computable function function $F : \{0,1\}^{n^2} \to S_n$ such that $F(U_{n^2})$ is a 2^{-n}-close to $\mathrm{UniPerm}_n$. (Note that since $|S_n| = n!$ is not a power of 2, there has to be some statistical error, but using a seed sufficiently larger than $\log n!$, the error can be made small).

We will be interested in the distribution $F(G(U_d))$ where G is a function (that will later be chosen to be a PRG).

Definition 2.31 (Pseudorandomly-chosen permutation)**.** *Given a function $G : \{0,1\}^d \to \{0,1\}^{n^2}$ we define: $\pi^G : \{0,1\}^d \times [n] \to [n]$ by: $\pi^G(s,i) = (F(s))(i)$ (namely, applying the permutation $\pi = F(s)$ on i). We use π^G_s to denote the permutation π, defined by $\pi(i) = \pi^G(s,i)$. We omit G when it is clear from the context.*

We use the term "pseudorandonly-chosen permutation" to differentiate this notion from the cryptographic notion of "pseudorandom permutation" (which is different).

2.6 Decoding from errors induced by a random permutation

As is the case of earlier work on codes for bounded channels [GS16, SS21a, KSS19, SS21b] we will actually be interested in evasive codes that decode not just from BSC_p, but in a related (and more general) setup that we now explain. The definition below uses the notion of permuting strings from Definition 2.30.

Definition 2.32 (Noise induced by a distribution on permutations)**.** *Let Π be a distribution over permutations $\pi : [n] \to [n]$. For every $e \in \{0,1\}^n$, we can consider the "noise distribution" $\Pi(e)$, and let Perm_p^{Π} denote the class of all such distributions over all choices of $e \in \{0,1\}^n$ such that $\mathsf{wt}(e) \leq p$.*

We say that a pair $(\mathrm{Enc}, \mathrm{Dec})$ decodes from Perm_p^{Π} if it decodes from every distribution in Perm_p^{Π}.

Recall that we use $\mathrm{UniPerm}_n$ to denote the uniform distribution on permutations on $[n]$ and omit n when it is clear from the context. Note that for every $e \in \{0,1\}^n$ such that $\mathsf{wt}(e) = p$, the distribution $\mathrm{UniPerm}(e)$ is "somewhat similar" to BSC_p. More formally, individual bits of $\mathrm{UniPerm}(e)$ are distributed like individual bits of BSC_p, and while bits of $\mathrm{UniPerm}(e)$ are not independent, the correlation between "not too many" of them is "small". By the same rationale the distributions in $\mathrm{Perm}_p^{\mathrm{UniPerm}}$ are "somewhat similar" to the distributions in $\{\mathrm{BSC}_{p'} : p' \leq p\}$.

This similarity can be used to show that codes designed to decode from BSC_p, often also decode from $\mathrm{Perm}_p^{\mathrm{UniPerm}}$. We will be relying on such a code construction, which we now state. These constructions rely on good "standard codes" with high rate.

Theorem 2.33 ([Smi07, GS16, SS21a, KSS19])**.** *Let $R(p) = 1 - H(p)$. For every constant $0 \leq p < \frac{1}{4}$, and every sufficiently small constant $\epsilon > 0$, there exist infinitely many n, and functions $\mathrm{Enc} : \{0,1\}^{k(n)=(R(p)-\epsilon)\cdot n} \to \{0,1\}^n$, $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^{k(n)}$ such that:*

- *$(\mathrm{Enc}, \mathrm{Dec})$ decode from $\mathrm{Perm}_p^{\mathrm{UniPerm}}$ with success probability $1 - 2^{-\Omega(n^{0.1})}$.*
- *$(\mathrm{Enc}, \mathrm{Dec})$ are explicit.*

These constructions are based on concatenating an outer code with rate roughly $1 - \epsilon/2$ (that decodes from few errors) with a random inner code of rate $1 - H(p) - \epsilon/2$ that decodes from BSC_p. The monotonicity property (that is not stated explicitly in these results) is a byproduct of this concatenation approach. We also remark that the result is stated for infinitely many n, but this subset of integers is very dense, and depends on the density of block lengths n achieved by existing constructions of the outer codes.

Remark 2.34. *Theorem 2.33 is stated for infinitely many n, rather than all sufficiently large n. However, it works for a very dense set of n, as the construction is very simple and essentially concatenates a high-rate error correcting code that decode from few errors, with an inner code that is found with brute force.*

3 Functions that are hard to sample on low entropy distributions (HTS)

In this section we introduce a notion of functions that are hard to sample (HTS) on every low min-entropy distribution. As we have explained in Section 1.2, this definition is inspired by, and builds on thw work of Viola [Vio12] on hardness of sampling.

In Section 3.1 we define the new notion in a more general way than was done in Section 1.2, and discuss its meaning. In Section 3.2 we restate Theorems 1.2 and 1.3 (which are the main results of this section). In the remainder of the section we prove these theorems. In Section 3.3 we show how to use known hardness amplification results to obtain a weak HTS. In Section 3.4 we show how to convert a weak HTS to a stronger HTS. Finally in Section 3.5 we put things together and prove Theorems 1.2 and 1.3.

3.1 Definition of HTS

We would like to define a notion of hard function f, such that no sampling procedure A in some (possibly nonuniform) class $\mathcal{C}$ can sample a pair (X, Y) such that $Y = f(X)$ with large probability. An obvious difficulty is that (as explained in Section 1.2) A (which is nonuniform) may be hardwired with pairs of the form $(x_1, f(x_1)), \ldots, (x_t, f(x_t))$, and can therefore easily sample a distribution X, Y such that X is uniform over t elements, and $\Pr[Y = f(X)] = 1$.

The definition below handles this problem by requiring that for every A, there exists a small set H of inputs, such that A is considered "winning" only on $X \notin H$.

Definition 3.1 (Hard To Sample (HTS))**.** *A function $f : \{0,1\}^m \to \{0,1\}^{m'}$ is an (h, ρ)-**HTS** for a class $\mathcal{C}$ of functions, if for every $A \in \mathcal{C}$ that samples a distribution $Z = (X, Y)$ over $\{0,1\}^m \times \{0,1\}^{m'}$, there exists a set $H \subseteq \{0,1\}^m$ of size at most h, such that:*

$$\Pr_{(X,Y) \leftarrow A}[X \notin H \text{ and } Y = f(X)] \leq \rho.$$

*We say that f is an (h, ρ)-**natural-HTS** if the set $H = \left\{x : \Pr_{(X,Y)\leftarrow A}[X = x] \geq \frac{1}{h}\right\}$ satisfies the requirement above.*

In the next three subsections, we elaborate on the choices made in this definition, and its relation to other notions of hard functions.

3.1.1 Any HTS can be made natural at a small cost

Our definition of HTS does not enforce that f is natural (namely that H is the set of heaviest elements). Nevertheless, it is easy to observe that this holds w.l.o.g. if $1/\rho$ is small compared to h (and this will always be the case in this paper).

Proposition 3.2 (Any HTS is w.l.o.g. natural)**.** *If $f : \{0,1\}^m \to \{0,1\}^{m'}$ is an (h, ρ)-***HTS** *for a class $\mathcal{C}$, then f is an $(\frac{h}{\rho}, 2\rho)$-natural-HTS for $\mathcal{C}$.*

Proof. Let $A \in \mathcal{C}$ be some function, by Definition 3.1, there exists a set H of size at most h for A. Let $H' = \left\{x : \Pr_{(X,Y)\leftarrow A}[X = x] \geq \frac{\rho}{h}\right\}$. Note that every $x \notin H'$ is either in $H \setminus H'$ or in the complement of H. Therefore:

$$\begin{aligned}\Pr_{(X,Y)\leftarrow A}[X \notin H' \text{ and } Y = f(X)] &\leq \Pr_{(X,Y)\leftarrow A}[X \in H \setminus H'] + \Pr_{(X,Y)\leftarrow A}[X \notin H \text{ and } Y = f(X)] \\ &\leq h \cdot \frac{\rho}{h} + \rho \\ &\leq 2\rho,\end{aligned}$$

as required. □

The fact that we can essentially assume w.l.o.g. that any HTS is natural is helpful, as it gives that any $x \notin H$, does not have high weight according to X.

3.1.2 What kind of hardness is captured by an HTS?

We now observe that an HTS indeed implies several notions of hardness (as discussed in the introduction, and specifically in Section 1.2). The next proposition confirms he statement made in Section 1.2 that am HTS is "hard to sample on low min-entropy distributions" which implies "hard to compute on low min-entropy samplable distributions" which implies "hard to compute on the uniform distribution".

Proposition 3.3 (HTS implies hard functions)**.** *If $f : \{0,1\}^m \to \{0,1\}^{m'}$ is an (h, ρ)-HTS for size s circuits then:*

1. *For every circuit A of size s that samples a distribution (X, Y) such that $H_\infty(X) \geq \log h + \log(1/\rho)$,*

$$\Pr_{(X,Y)\leftarrow A}[Y = f(X)] \leq 2\rho.$$

2. *In particular, for every distribution X over $\{0,1\}^m$, such that $H_\infty(X) \geq \log h + \log(1/\rho)$, and X is samplable by size $s/2$ circuits, and for every circuit C of size $s/2$, we have that:*

$$\Pr[C(X) = f(X)] \leq 2\rho.$$

3. *In particular, if $h + \log(1/\rho) \leq m$ and if $s \geq 2m$, as $X = U_m$ is samplable by size $s/2$ circuits, we have that f is 2ρ-hard for circuits of size $s/2$.*

Proof. The first item follows because by the definition of HTS, there exists $H \subseteq \{0,1\}^m$ of size h such that:

$$\Pr_{(X,Y)\leftarrow A}[X \notin H \text{ and } Y = f(X)] \leq \rho.$$

As $H_\infty(X) \geq \log h + \log(1/\rho)$ we have that

$$\Pr_{(X,Y)\leftarrow A}[X \in H] \leq \frac{h}{2^{\log h + \log(1/\rho)}} \leq \rho.$$

Therefore,

$$\begin{aligned}\Pr_{(X,Y)\leftarrow A}[Y = f(X)] &= \Pr_{(X,Y)\leftarrow A}[X \notin H \text{ and } Y = f(X)] + \Pr_{(X,Y)\leftarrow A}[[X \in H \text{ and } Y = f(X)] \\ &\leq \rho + \Pr_{(X,Y)\leftarrow A}[X \in H] \\ &\leq 2\rho.\end{aligned}$$

For the second item, note that if there exists a size $s/2$ circuit D that samples a distribution X over $\{0,1\}^m$ with $H_\infty(X) \geq \log h + \log(1/\rho)$, and there also exists a circuit C of size $s/2$ such that $\Pr[C(X) = f(X)] > 2\rho$, then we can consider the size s circuit A that samples X using D and computes $Y = C(X)$, and use the previous item.

The third item follows by the min-entropy requirement on X. This gives a contradiction. □

3.1.3 HTS vs function that is hard to sample on low min-entropy distributions

We also comment that the first item in Proposition 3.3 (which we referred to as "hard to sample on low min-entropy distributions" in the introduction) is in some sense weaker than our definition of an HTS. This is because Definition 3.1 does not limit A to sample a high min-entropy distribution X, and the restriction holds for any A (even one that samples a low min-entropy distribution) guaranteeing that:

$$\begin{aligned}\Pr_{(X,Y)\leftarrow A}[Y = f(X)] &= \Pr_{(X,Y)\leftarrow A}[X \notin H \text{ and } Y = f(X)] + \Pr_{(X,Y)\leftarrow A}[X \in H \text{ and } Y = f(X)] \\ &\leq \rho + \Pr_{(X,Y)\leftarrow A}[X \in H],\end{aligned}$$

where H is a small the set of heaviest elements.

We cannot obtain such bounds from the hardness guaranteed in the first item of the proposition. This is because if A is a size s circuit that samples a pair (X,Y), then while we can expect that the distribution $X' = (X|X \notin H)$ has high min-entropy (as we have removed the heaviest elements) it is not necessarily the case that X' is samplable by circuits of small size. This means that we cannot control an A which samples a distribution X for which $H_\infty(X)$ is small, and yet $\Pr_{(X,Y)\leftarrow A}[X \notin H]$ is significant.

Moreover, in our applications, we will have auxiliary arguments showing that we are already winning in case $X \in H$ (regardless of whether $Y = f(X)$ or not, due to auxiliary reasons). However, it will be crucial to handle *all* efficient A. This will follow because by our definition every efficient A has a set H, and we are guaranteed that

$$\Pr_{(X,Y)\leftarrow A}[Y = f(X) \text{ and } X \notin H] \leq \rho.$$

3.2 Explicit constructions of HTS from hardness assumptions

We give two constructions of HTS. Both are based on the hardness assumption that E is hard for exponential size nondeterministic circuits. The next theorem (which is a formal restatement of Theorem 1.2) gives an HTS f for circuits of size n^c, with very small $h = \mathrm{poly}(n^c)$. A disadvantage of this theorem is that the output length of f is polynomially larger than the input length.

Theorem 3.4 is stated in a way that allows one to choose the input length m of f to be any value between $a \log n \leq m \leq n$ for a large constant a, and this statement is made, as we would like this flexibility in later applications.

Theorem 3.4 (HTS with small h)**.** *There exists a constant $c_0 > 1$ such that if E is hard for exponential size nondeterministic circuits then for every constant $c > 1$, there exist constants $a, d > 1$, such that for every*

sufficiently large n, and every m such that $a \log n \leq m \leq n$, there is a function $f : \{0,1\}^m \to \{0,1\}^{m^{c_0}}$ that is an (h, ρ)-HTS for circuits of size n^c, where $h = n^a$ and $\rho = n^{-c}$. Furthermore, f is computable in time n^d.

The next theorem (which is a formal restatement of Theorem 1.3 gives an HTS f for circuits of size n^c, with medium sized $h = 2^{\delta n}$. It has the advantage that the output length of f can be smaller than the input length.

Theorem 3.5 (HTS with "medium" h and small output length)**.** *If E is hard for exponential size nondeterministic circuits then for every constants $c > 1$, $\delta > 0$ and $0 < \lambda < 1$, there exists a constant d such that for every sufficiently large n, there is a function $f : \{0,1\}^n \to \{0,1\}^{\lambda \cdot n}$ that is an (h, ρ)-HTS for circuits of size n^c, where $h = 2^{\delta n}$ and $\rho = n^{-c}$. Furthermore, f is computable in time n^d.*

In the remainder of this section we prove the two theorems. The proof will work in several stages. In Section 3.3 we show how to construct an HTS with weak parameters. In Section 3.4 we show how to transform a "weak HTS" into one with improved parameters. We give two such methods: The first is based on list-recoverable codes, and the second is based on extractors for weakly recognizable distributions.

Loosely speaking, the first approach can significantly improve the dependence of h on the input length (at the cost of harming the output length). The second approach can achieve short output length, but (as the parameters of known extractors for weakly recognizable distributions are not as good as we'd like) gives an HTS with rather large h. Our constructions of HTS are obtained in Section 3.5, by using specific constructions of list-recoverable codes and relative error extractors for weakly recognizable distributions. Along the way, we also give a construction of relative-error extractors for weakly-recognizable distributions, with parameters that are suitable for our application.

3.3 A weak HTS from hardness amplification

In this section we show how to construct an HTS that is weak (in the sense that it has h very close to 2^m) using the classical results on hardness amplification by [IW97, STV01]. We will prove the following:

Theorem 3.6 (HTS from hardness amplification)**.** *If E is hard for exponential size nondeterministic circuits then for every constant $c > 1$, there exist constants $a > 1$ and $0 < \gamma, \delta < 1$, such that for every sufficiently large n, there is a function $f : \{0,1\}^m \to \{0,1\}^{\delta m}$ that is an (h, ρ)-natural HTS for circuits of size n^c, where $m = a \log n$, $h = 2^{(1-\gamma)m}$, $\rho = n^{-c}$, and $h \leq \frac{\rho \cdot 2^m}{2}$. Furthermore, f is computable in time $\mathrm{poly}(n^c)$.*

We start by stating the classical hardness amplification results of [IW97, STV01]. The next theorem is stated for deterministic circuits in [IW97, STV01], but it was already observed before by Klivans and van-Melkebeek [KvM02] that the proof relativizes, and furthermore that the oracle queries are non-adaptive. Therefore, as explained in Remark 2.2 it immediately extends to $\mathrm{NP}_{||}$-circuits.

Theorem 3.7 (Hardness amplification for $\mathrm{NP}_{||}$-circuits [IW97, STV01, KvM02])**.** *If E is hard for exponential size $\mathrm{NP}_{||}$-circuits then for every constant $c > 1$, there exist constants $a > 1$ and $0 < \delta < 1$, such that for every sufficiently large n, there is a function $f : \{0,1\}^m \to \{0,1\}^{\delta m}$ that is n^{-c}-hard for $\mathrm{NP}_{||}$-circuits of size n^c, where $m = a \log n$. Furthermore, f is computable in time $\mathrm{poly}(n^c)$.*

Theorem 3.6 follows from Theorem 3.7, Theorem 2.5 (which weakens the hardness assumption in Theorem 3.7) and the next lemma, which states that a function that is ϵ-hard for size s $\mathrm{NP}_{||}$-circuits, gives a weak HTS with $h \approx \epsilon \cdot 2^m$ and $\rho \approx \epsilon$.

Lemma 3.8 (HTS from functions that are average-case hard for $\text{NP}_{||}$-circuits)**.** *If $f : \{0,1\}^m \to \{0,1\}^{m'}$ is ϵ-hard for size s $\text{NP}_{||}$-circuits, then f is an (h,ρ)-natural-HTS for circuits of size $s' = \frac{s^{\Omega(1)}}{\log(1/\rho)}$, where $h = \epsilon^{\frac{1}{2}} \cdot 2^m$, and $\rho = 32 \cdot \epsilon^{\frac{1}{4}}$.*

We conclude this subsection with the proof of Lemma 3.8.

3.3.1 Proof of Lemma 3.8

In the next definition we consider a "hybrid distribution" where given a sampling algorithm A that samples a pair (x,y), the first input x is chosen according to some other specified distribution X' and then y is chosen by the sampling algorithm conditioned on the first element being x

Definition 3.9. *For a function $A : \{0,1\}^r \to \{0,1\}^m \times \{0,1\}^{m'}$, we say that $x \in \{0,1\}^m$ is* **possible** *for A, if $\Pr_{(X,Y)\leftarrow A}[X = x] > 0$.*

For a string $x' \in \{0,1\}^m$ that is possible for A, we use $A^{x'}$ to denote the experiment which produces a pair (x,y) where $x = x'$ and y is chosen from in the experiment $(X,Y) \leftarrow A$ conditioned on the event $\{X = x'\}$.

For a distribution X' over strings that are possible for A, we use $A^{X'}$ to denote the experiment where after $x' \leftarrow X'$ is chosen and fixed, Y is chosen according to $A^{x'}$.

The next lemma shows that if f is not a natural-HTS, and is "broken" by some sampling algorithm A, then there exists a large subset T such that for $X' = U_T$ (the uniform distribution overt T), if we sample $(x,y) \leftarrow A^{X'}$, then the probability that $y = f(x)$ is large. This means that the function f is in some sense "easy" on the *uniform distribution* over some large set T.

Lemma 3.10. *Let $f : \{0,1\}^m \to \{0,1\}^{m'}$ be a function. If f is not an (h,ρ)-natural-HTS for A, then there exists a subset $T \subseteq \{0,1\}^m$ of size at least $\frac{\rho \cdot h}{4}$ such that every $x \in T$ is possible for A, and if we denote the uniform distribution over T by U_T, we have that:*

$$\Pr_{(X,Y)\leftarrow A^{U_T}}[Y = f(X)] \geq \frac{\rho}{4}.$$

Proof. We have that for $H = \left\{x : \Pr_{(X,Y)\leftarrow A}[X = x] \geq \frac{1}{h}\right\}$ it holds that:

$$\Pr_{(X,Y)\leftarrow A}[X \notin H \text{ and } Y = f(X)] > \rho.$$

For every $i \geq 0$, we define

$$W_i = \left\{x : \frac{1}{h \cdot 2^{i+1}} \leq \Pr_{(X,Y)\leftarrow A}[X = x] < \frac{1}{h \cdot 2^i}\right\}.$$

Note that the sets $\{W_i\}_{i=0}^{\infty}$ form a partition of $\{0,1\}^m \setminus H$. Let $t = \rho \cdot h/4$. We partition the set of integers into $B = \{i \geq 0 : |W_i| < t\}$ and $G = \{i \geq 0 : |W_i| \geq t\}$. Let $W_B = \cup_{i\in B} W_i$ and $W_G = \cup_{i\in G} W_i$. Our plan is to show that there exists $i \in G$, such that we can take $T = W_i$. Towards this goal we start by observing

that the probability of W_B is small:

$$\begin{aligned}\Pr_{(X,Y)\leftarrow A}[X \in W_B] &= \sum_{i\in B} \Pr_{(X,Y)\leftarrow A}[X \in W_i] \\ &< \sum_{i=0}^{\infty} \frac{t}{h\cdot 2^i} \\ &\leq 2\cdot\frac{t}{h} \\ &\leq \frac{\rho}{2}.\end{aligned}$$

As W_G, W_B and H form a partition of $\{0,1\}^m$, we have that:

$$\begin{aligned}\Pr_{(X,Y)\leftarrow A}[X \in W_G \text{ and } Y = f(X)] &\geq \Pr_{(X,Y)\leftarrow A}[X \notin H \text{ and } Y = f(X)] - \Pr_{(X,Y)\leftarrow A}[X \in W_B] \\ &\geq \rho - \frac{\rho}{2} \\ &\geq \frac{\rho}{2}.\end{aligned}$$

By averaging, this can be used to argue that there exists $i \in G$, such that

$$\Pr_{(X,Y)\leftarrow A}[X \in W_i \text{ and } Y = f(X) | X \in W_i] \geq \frac{\rho}{2}.$$

Note that the distribution of X under the condition above is "roughly uniform" over W_i, as all elements in W_i have roughly the same weight, and so for $T = W_i$, the distribution of X is "roughly" A^{U_T}.

We will now make this argument precise by first "rounding" the weights of all elements in each W_i, so that the weights will be exactly the same in a distribution X' (that is not very different from X) and then we apply the argument sketched above on X'. Details follow.

For every $x \in \{0,1\}^m$ we define:

$$p_x = \begin{cases} \Pr_{(X,Y)\leftarrow A}[X = x], & x \in H \\ \frac{1}{h\cdot 2^{i+1}}, & x \in W_i \end{cases}$$

Note that this is well defined, and for every $x \in \{0,1\}^m$,

$$p_x \leq \Pr_{(X,Y)\leftarrow A}[X = x] \leq 2p_x.$$

For every $x \in \{0,1\}^m$, we define $p'_x = \frac{p_x}{\sum_{x\in\{0,1\}^m} p_x}$, and let X' be the distribution such that $\Pr[X' = x] = p'_x$. We have that $\sum_{x\in\{0,1\}^m} p_x \geq \frac{1}{2}$, and this gives that for every x,

$$\frac{1}{2}\cdot \Pr_{(X,Y)\leftarrow A}[X = x] \leq \Pr[X' = x] \leq 2\cdot \Pr_{(X,Y)\leftarrow A}[X = x].$$

It follows that for every event E,

$$\frac{1}{2}\cdot \Pr_{(X,Y)\leftarrow A}[E] \leq \Pr_{(X,Y)\leftarrow A^{X'}}[E] \leq 2\cdot \Pr_{(X,Y)\leftarrow A}[E].$$

In particular, we can conclude that:

$$\Pr_{(X,Y)\leftarrow A^{X'}}[X \in W_G \text{ and } Y = f(X)] \geq \frac{\rho}{4}.$$

This gives that:

$$\begin{aligned}
\frac{\rho}{4} &\leq \Pr_{(X,Y)\leftarrow A^{X'}}[X \in W_G \text{ and } Y = f(X)] \\
&= \sum_{i=0}^{\infty} \Pr_{(X,Y)\leftarrow A^{X'}}[X \in W_G \text{ and } Y = f(X) | X' \in W_i] \cdot \Pr[X' \in W_i] \\
&= \sum_{i \in G} \Pr_{(X,Y)\leftarrow A^{X'}}[X \in W_G \text{ and } Y = f(X) | X' \in W_i] \cdot \Pr[X' \in W_i] \\
&\leq \max_{i \in G} \Pr_{(X,Y)\leftarrow A^{X'}}[X \in W_G \text{ and } Y = f(X) | X' \in W_i] \\
&= \max_{i \in G} \Pr_{(X,Y)\leftarrow A^{X'}}[Y = f(X) | X' \in W_i].
\end{aligned}$$

Let i^* be the index that maximizes the expression above. Let $T = W_{i^*}$ and note that the distribution $(X'|X' \in T)$ is the distribution U_T (that is the uniform distribution over T). This is because all elements in W_i were rounded to the same weight. We conclude that

$$\Pr_{(X,Y)\leftarrow A^{U_T}}[Y = f(X)] \geq \frac{\rho}{4},$$

and as $i \in G$, we have that $|T| = |W_i| \geq t = \frac{\rho \cdot h}{4}$. □

We are now ready to prove Lemma 3.8.

Proof. (of Lemma 3.8) By lemma 3.10 if f is not an (h, ρ)-natural-HTS for circuits of size s', then there exists a circuit A of size s' and a set $T \subseteq \{0,1\}^m$ of size $\frac{\rho \cdot h}{4}$ such that

$$\Pr_{(X,Y)\leftarrow A^{U_T}}[Y = f(X)] \geq \frac{\rho}{4}.$$

We now show that an $\mathrm{NP}_{||}$-circuit of size slightly larger than s' can (when given x) sample from A^x and therefore compute the function f too well on $X \leftarrow U_T$. Details follow.

For every $x \in \{0,1\}^m$ we define a circuit D_x which on input $w \in \{0,1\}^r$ (where r is the input length of A) performs the following: It computes $A(w)$ to produce strings $(x', y') \in \{0,1\}^m \times \{0,1\}^{m'}$ and outputs one iff $x' = x$.

Note that for every x that is possible for A, a uniform w such that $D_x(w) = 1$ is distributed exactly like A^x. By Theorem 2.17, there exists a randomized $\mathrm{NP}_{||}$-circuit T of size $\mathrm{poly}(s', \log(1/\rho))$ which when given x, outputs a value $W \in \{0,1\}^r \cup \{\perp\}$, such that:

- If there does not exist w, such that $D_x(w) = 1$, then $T(x)$ outputs $\perp$ with probability one.
- If there exists w such that $D_x(w) = 1$, then:
  - The probability that $T(x)$ outputs $\perp$ is at most $\rho/8$.
  - The distribution $T(x)$ conditioned on the event $\{T(x) \neq \perp\}$ is uniform over $\{w : D_x(w) = 1\}$.

This means that for x that is possible for w, conditioned on not answering $\perp$, $T(x)$ outputs a uniform W such that the first output of $A(W)$ is x, which means that $A(W)$ is distributed like A^x.

We consider the randomized $\mathrm{NP}_{||}$-circuit C which on input x:

- Computes $w = T(x)$, and outputs an arbitrary value if $w = \perp$.

- Computes $(x, y) = A(w)$ and outputs y.

It follows that for every x that is possible for A,

$$\Pr[C(x) = f(x)] = \Pr_{(X,Y)\leftarrow A^x}[Y = f(X)] - \frac{\rho}{8}.$$

Overall, we get that C is a circuit of size $\mathrm{poly}(s \cdot \log(1/\rho))$ such that

$$\begin{aligned}
\Pr_{X\leftarrow U_m}[C(X) = f(X)] &\geq \Pr_{X\leftarrow U_m}[C(X) = f(X) \text{ and } X \in T] \\
&\geq \Pr_{X\leftarrow U_T}[C(X) = f(X)] \cdot \Pr_{X\leftarrow U_m}[X \in T] \\
&\geq \left(\Pr_{(X,Y)\leftarrow A^{U_T}}[C(X) = f(X)] - \frac{\rho}{8}\right) \cdot \frac{\rho h}{4 \cdot 2^m} \\
&\geq \left(\frac{\rho}{4} - \frac{\rho}{8}\right) \cdot \frac{\rho h}{4 \cdot 2^m} \\
&= \frac{\rho^2 h}{32 \cdot 2^m} \\
&\geq \epsilon,
\end{aligned}$$

where the last inequality follows because we have chosen $h = \epsilon^{\frac{1}{2}} \cdot 2^m$, and $\rho = 32 \cdot \epsilon^{\frac{1}{4}}$. (Note that any choice of h and ρ satisfying the inequality will do).

We can fix the random coins of C to their "best value", removing the randomization and obtaining an $\mathrm{NP}_{||}$-circuit of size $s = \mathrm{poly}(s \cdot \log(1/\rho))$ with the same success probability. and this gives a contradiction. □

3.4 Strengthening a weak HTS

In this section we will show how to take an HTS on m bits with large h (that may be close to 2^m) and transform it into one over $n > m$ bits without increasing h by much (so that the ratio of h to input length is improved). We will use the following definition (which considers two transformations).

Definition 3.11 (HTS amplification)**.** *Given a function $f : \{0,1\}^m \to \{0,1\}^{m'}$ and a function $E : \{0,1\}^n \to (\{0,1\}^m)^D$, we define functions $T_{E,f} : \{0,1\}^n \to \{0,1\}^{D(m+m')}$ and $T'_{E,f} : \{0,1\}^n \to \{0,1\}^{Dm'}$ as follows: Given $x \in \{0,1\}^n$, we set $z_i = E(x)_i$, for every $i \in [D]$, and define:*

- $T_{E,f}(x) = ((z_1, f(z_1)), \ldots, (z_D, f(z_D)))$.
- $T'_{E,f}(x) = (f(z_1), \ldots, f(z_D))$.

3.4.1 Using list-recoverable codes

We show that if we take E to be a list-recoverable code than the transformations indeed provide an HTS where the ratio of h to input length can be improved.

Lemma 3.12 (Transformation using list-recoverable codes)**.** *If f is an (h, ρ)-HTS for size s circuits, and $E : \{0,1\}^n \to (\{0,1\}^m)^D$ is (h, L)-list-recoverable, then $T_{E,f}$ is an $(L, \rho \cdot D)$-HTS for size s circuits. If furthermore, E can be computed by size t circuits, then $T'_{E,f}$ is an $(L, \rho \cdot D)$-HTS for size $s - t$ circuits.*

Proof. Let A be a sampling circuit of size s that samples a pair $(x, y) \in \{0,1\}^n \times \{0,1\}^{D(m+m')}$. We think of y as a sequence $y = ((x_1, y_1), \ldots, (x_d, y_d))$ where for every $i \in [D]$, $(x_i, y_i) \in \{0,1\}^m \times \{0,1\}^{m'}$. For

every $i \in [D]$, we can define a sampling circuit A_i of the same size, which simulates A and outputs (x_i, y_i) (discarding the other outputs). As f is an (h, ρ)-HTS for size s circuits, it follows that for every $i \in [D]$, the sampling circuit A_i has a set $H_i \subseteq \{0,1\}^m$ of size at most h such that:

$$\Pr_{(X_i,Y_i)\leftarrow A_i}[X_i \notin H_i \text{ and } Y_i = f(X_i)] \leq \rho.$$

By the list-recoverability of E, we have that the set $H = \mathrm{List}_E(H_1, \ldots, H_D)$ is of size at most L. It follows that if $x \notin H$, then there exists $i \in [D]$ such that $z_i = E(x)_i \notin H_i$.

We will show that H satisfies the definition of an HTS for A. For this purpose, we will consider the probability space where

$$(X, Y) = (X, (X_1, Y_1), \ldots, (X_D, Y_D))) \leftarrow A,$$

and for every $i \in D$, we set $Z_i = E(X)_i$. We have that:

$$\begin{aligned}
\Pr[X \notin H \text{ and } Y = f(X)] &= \Pr[X \notin H \text{ and } \forall i \in [D] : (X_i = Z_i \text{ and } Y_i = f(X_i))] \\
&\leq \Pr[\exists i \in [D] : Z_i \notin H_i \text{ and } \forall i \in [D] : (X_i = Z_i \text{ and } Y_i = f(X_i))] \\
&\leq \Pr[\exists i \in [D] : (Z_i \notin H_i \text{ and } X_i = Z_i \text{ and } Y_i = f(X_i))] \\
&\leq \Pr[\exists i \in [D] : (X_i \notin H_i \text{ and } Y_i = f(X_i))] \\
&\leq \sum_{i \in [D]} \Pr[X_i \notin H_i \text{ and } Y_i = f(X_i)] \\
&\leq D \cdot \rho.
\end{aligned}$$

Here, the last inequality follows because the distribution of (X_i, Y_i) in our probability space is identical to $(X_i, Y_i) \leftarrow A_i$.

We now turn our attention to the "furthermore part". Note that if A of size s' samples $(x, (y_1, \ldots, y_D))$, then by applying E on x (producing $(z_1, \ldots, z_D)$) we can convert A to a sampling algorithm A' of size $s' + t$ that runs A, and outputs $(x, (z_1, y_1), \ldots, (x_D, y_D))$. The previous argument now works for A', proving the "furthermore part". □

3.4.2 Using extractors for weakly recognizable distributions

We can view a function $E : \{0,1\}^n \to \{0,1\}^m$ as a function $E : \{0,1\}^n \to (\{0,1\}^m)^D$ for $D = 1$, and apply the transformation of Definition 3.11 using such a function E. The next lemma shows that this can be used to strengthen an HTS using extractor for distributions that are weakly recognizable by $\mathrm{NP}_{||}$-circuits.

Lemma 3.13 (Transformation using extractors for recognizable distributions)**.** *There exists a constant $c_0 > 1$ such that if f is an (h, ρ)-HTS for size s circuits, $E : \{0,1\}^n \times \{0,1\}^m$ is a $(k, \frac{1}{10})$-relative-error extractor for distributions that are weakly-recognizable by size s^{c_0} $\mathrm{NP}_{||}$-circuits, and $h \leq \frac{\rho \cdot 2^m}{2}$, then $T_{E,f}$ is a $(\frac{2^{k+2}}{\rho}, 2\rho)$-HTS for size s circuits.*

Proof. Let A be a sampling circuit of size s that samples a pair $(x, y) \in \{0,1\}^n \times \{0,1\}^{m+m'}$. We think of y as a pair $y = (x_1, y_1) \in \{0,1\}^m \times \{0,1\}^{m'}$. For every $x \in \{0,1\}^n$, let $p_x = \Pr_{(X,Y)\leftarrow A}[X = x]$. Let $\epsilon > 0$ be small constant that will be chosen later. By Theorem 2.16 there is an $\mathrm{NP}_{||}$-circuit C of size $\mathrm{poly}(s)$ such that for every $x \in \{0,1\}^n$, $p'_x = C(x)$ is an ϵ-relative approximation of p_x. Let $t = \frac{2^{k+1}}{\rho}$ and define $H = \left\{x : p'_x \geq \frac{1}{t}\right\}$. We can use C to construct a size $\mathrm{poly}(s)$ $\mathrm{NP}_{||}$-circuit C' such that for every $x \in \{0,1\}^n$,

$$C'(x) = \begin{cases} C(x), & x \notin H \\ 0, & x \in H \end{cases}$$

By choosing $\epsilon > 0$ to be sufficiently small, it immediately follows that:

- For every $x \in H$, $\Pr[X = x] \geq \frac{1}{2t}$.
- For every $x \notin H$, $\Pr[X = x] \leq \frac{2}{t}$.

In particular, we have that $|H| \leq 2t = \frac{2^{k+2}}{\rho}$. We plan to show that H can serve as a good set for A. If $\Pr[X \notin H] \leq 2\rho$ then this trivially follows. Therefore, we will assume that $\Pr[X \notin H] > 2\rho$. We define $X' = (X|X \notin H)$, and note that it follows that for every $x \notin H$:

$$\Pr[X' = x] = \Pr[X = x|X \notin H] = \frac{\Pr[X = x]}{\Pr[X \notin H]} \leq \frac{2/t}{2\rho} = \frac{1}{t\rho}.$$

It follows that:

$$H_\infty(X') \geq \frac{1}{t\rho} = \log t - \log \frac{1}{\rho}.$$

Let X'' be the distribution that is weakly recognized by C'. It follows that for every $x \in \{0,1\}^n$, $\Pr[X'' = x]$ is an $O(\epsilon)$-relative approximation of $\Pr[X' = x]$, and in particular, for sufficiently small ϵ, we have that

$$H_\infty(X'') \geq H_\infty(X') - 1 \geq \log t - \log \frac{1}{\rho} - 1 \geq k.$$

Overall, we have that X'' is a distribution that is a high-min-entropy distribution that is recognizable by size $\mathrm{poly}(s)$ $\mathrm{NP}_{||}$-circuits, and by choosing the constant c_0 such that this size is bounded by s^{c_0}, it follows that $E(X'')$ is $1/10$-close in relative distance to the uniform distribution. We now consider the sampling circuit A' which acts exactly like A, but instead of outputting $(x, (x_1, y_1))$ it only outputs $(x_1, y_1) \in \{0,1\}^m \times \{0,1\}^{m'}$. By definition 3.1, this sampling circuit A' has a set $S \subseteq \{0,1\}^m$ of size h such that:

$$\Pr_{(X_1,Y_1) \leftarrow A'}[X_1 \notin S \text{ and } Y_1 = f(X_1)] \leq \rho.$$

By the properties of the extractor we have that:

$$\Pr[E(X'') \in S] \leq (1 + \frac{1}{10}) \cdot \Pr[U_m \in S] \leq (1 + \frac{1}{10}) \cdot \frac{h}{2^m}.$$

By the connection that we have already established between X' and X'' it follows that for every event F, $\Pr[X'' \in F]$ is an $O(\epsilon)$-relative approximation to $\Pr[X' \in F]$. In particular, we can conclude that:

$$\Pr[E(X') \in S] \leq (1 + O(\epsilon)) \cdot \Pr[E(X'') \in S] \leq \frac{2h}{2^m},$$

for sufficiently small $\epsilon > 0$. It also follows that:

$$\begin{aligned}
\Pr[X \notin H \text{ and } E(X) \in S] &= \Pr[X \notin H] \cdot \Pr[E(X) \in S|X \notin H] \\
&\leq \Pr[E(X') \in S] \\
&\leq \frac{2h}{2^m} \\
&\leq \rho,
\end{aligned}$$

where the last step follows from the requirements of the lemma. We are finally ready to show that H can serve as the required set for the sampling circuit A. In the calculation below, we consider the experiment

$(X, Y) \leftarrow A$, and the notation $Y = (X_1, Y_1)$.

$$\begin{aligned}
\Pr[X \notin H \text{ and } Y = T_{E,f}(X)] &= \Pr[X \notin H \text{ and } X_1 = E(X) \text{ and } Y_1 = f(X_1)] \\
&\leq \Pr[X \notin H \text{ and } X_1 = E(X) \text{ and } Y_1 = f(X_1) \text{ and } E(X) \in S] \\
&\quad + \Pr[X \notin H \text{ and } X_1 = E(X) \text{ and } Y_1 = f(X_1) \text{ and } E(X) \notin S] \\
&\leq \Pr[X \notin H \text{ and } E(X) \in S] + \Pr[Y_1 = f(X_1) \text{ and } X_1 \notin S] \\
&\leq \rho + \Pr_{(X_1,Y_1)\leftarrow A_1}[X_1 \notin S \text{ and } Y_1 = f(X_1)] \\
&\leq \rho + \rho \\
&= 2\rho.
\end{aligned}$$

Here, the fourth line follows because the distribution $(X_1, Y_1) \leftarrow A_1$ is identical to that of taking only (X_1, Y_1) when sampling $(X, (X_1, Y_1)) \leftarrow A$. □

3.5 Putting things together

We are finally ready to prove Theorems 3.4 and Theorem 3.5.

3.5.1 Proof of Theorem 3.4

We use the assumption to apply Theorem 3.6 and obtain an (h, ρ)-HTS $f : \{0,1\}^m \to \{0,1\}^{m'}$ for circuits of size n^c with the parameters specified in the theorem. We plan to use the transformation $T_{E,f}$ of Definition 3.11 to obtain a function $T_{E,f}$ with input length $\bar{m}$ that is between $a \log n$ and n (for a constant a that will be slightly larger than the constant a guaranteed in Theorem 3.6). We will use Proposition 2.23 to interpret the strong extractor of Theorem 2.24 as a list-recoverable code. More specifically, let $E : \{0,1\}^{\bar{m}} \times \{0,1\}^{O(\log \bar{m})} \to \{0,1\}^m$ be an $(O(m), \frac{1}{10})$-extractor (as is guaranteed by Theorem 2.24). By Proposition 2.23, this translates into a function $E : \{0,1\}^{\bar{m}} \to (\{0,1\}^m)^D$ that is $(2^{\bar{m}-1}, L)$-list-recoverable for $D = \bar{m}^{O(1)}$, and $L = 2^{O(m)} = n^{O(1)}$. We then apply the transformation $T_{E,f}$ of Definition 3.11, using Lemma 3.12. We indeed meet the condition of Lemma 3.12 and conclude that $T_{E,f} : \{0,1\}^{\bar{m}} \to \{0,1\}^{D(m+m')}$ is an (h', ρ')-HTS for circuits of size n^c. We have that $h' = L = n^{O(1)}$ and $\rho' = \rho \cdot D = n^{-(c-O(1))}$ (and we could have gotten $\rho' = n^{-c}$ by choosing the constant c with which we apply Theorem 3.6 to be sufficiently larger). Overall, we indeed obtain that there exists a constant $a > 1$ (that is possibly larger than the constant guaranteed in Theorem 3.6) such that for every $\bar{m}$ such that $a \log n \leq \bar{m} \leq n$, there is a function $T_{E,f} : \{0,1\}^{\bar{m}} \to \{0,1\}^{\bar{m}^{c_0}}$ (where c_0 is a universal constant determined by the constant hidden in the $O(\cdot)$ notation in the parameter d) that is an (h', ρ')-HTS as required. Note that as f can be computed in time n^c and the extractor can be computed in time $\bar{m}^{O(1)} = \mathrm{poly}(n^c)$ we indeed have that the function $T_{E,f}$ can be computed in time $\mathrm{poly}(n^c)$ where this polynomial does not depend on the input length of $T_{E,f}$.

3.5.2 A construction of extractors for weakly recognizable distributions with high min-entropy

In Section 3.4.2 we have shown how to improve the quality of an HTS using relative-error extractors for distributions which are weakly recognizable by $\mathrm{NP}_{||}$-circuits. In this subsection we give a construction of such extractors (for the $k = n - O(\log n)$) based on the assumption that E is hard for exponential size nondeterministic circuits.

Theorem 3.14 (Extractors for weakly recognizable large min-entropy distributions by $\mathrm{NP}_{||}$-circuits)**.** *If E is hard for exponential size nondeterministic circuits then for every constants a, c, c_1, and for every sufficiently large n, there is an $(n - a \cdot \log n, \frac{1}{n^c})$-relative-error extractor $E : \{0,1\}^n \to \{0,1\}^{c_1 \cdot \log n}$ for distributions weakly recognizable by size n^c $\mathrm{NP}_{||}$-circuits. Furthermore, E is computable in time $\mathrm{poly}(n^{a+c+c_1})$.*

We remark that using a construction of Applebaum et al. [AASY15], it is possible to obtain a relative error extractor for smaller k. More specifically, for $k = (1-\alpha) \cdot n$ where $\alpha > 0$ is some constant. However, this approach would need to assume the stronger assumption that E is hard for exponential size Σ_4-circuits. Here, we assume a weaker assumption and get a weaker conclusion.

The approach we use for proving Theorem 3.14 is similar to an approach of Kinne, Shaltiel and van Melkebeek [KvMS12] (see also [LZ19]). However, we need to analyze the construction more carefully, and also obtain that it gives "relative error" rather than just "additive error", and that it applies to "weakly recognizable distributions" and not just to "recognizable distributions".

The proof of Theorem 3.14 proceeds in two steps. In the first step (stated in Lemma 3.15) we argue that any seed extending PRG with small error, yields an extractor with the properties we require. In the second step (stated in Corollary 3.18) we use Theorem 2.8 to obtain a suitable PRG under the hardness assumption of Theorem 3.14.

Lemma 3.15 (PRG to extractor for recognizable distributions (adaptation of the argument of [KvMS12]))**.** *There exists a constant c_0, such that for every constant c, for every sufficiently large n, and every $m \leq n$, $\epsilon > 0$ and $\Delta > 0$*

- *If $E : \{0,1\}^n \to \{0,1\}^m$ satisfies that the function $G : \{0,1\}^n \to \{0,1\}^{n+m}$ defined by $G(x) = (x, E(x))$ is an $\frac{\epsilon}{2^{m+\Delta}}$-PRG for $NP_{||}$-circuits of size $n^{c \cdot c_0}$,*
- *Then E is an $(n-\Delta, \epsilon)$-relative error extractor for distributions that are weakly recognizable by size n^c $NP_{||}$-circuits.*

Remark 3.16. *This Lemma improves upon a similar Lemma from [KvMS12], in that it achieves a stronger conclusion under the same assumption. While we state the lemma for the class of $NP_{||}$-circuits, as in [KvMS12], it applies to many other circuit classes, including standard deterministic circuits, NP-circuit, and essentially any reasonable circuit class that is closed under composition.*

Proof. (of Lemma 3.15) Assume (for the purpose of contradiction) that X' is a distribution over $\{0,1\}^n$ such that $H_\infty(X') \geq n - \Delta$, where X' is weakly-recognizable by an $\text{NP}_{||}$-circuit C of size n^c, and that $E(X')$ is not ϵ-close to uniform with relative error. It follows that there exists $z \in \{0,1\}^m$ such that $\frac{1}{2^m}$ is not an ϵ-relative approximation of $\Pr[E(X') = z]$, meaning that:

$$|\frac{1}{2^m} - \Pr[E(X') = z]| > \epsilon \cdot \frac{1}{2^m}$$

We have that X' is weakly recognizable by an NP-$_{||}$-circuit C. This formally means that for every $x \in \{0,1\}^n$, $\Pr[X' = x] = \frac{C(x)}{\sum_{x \in \{0,1\}^n} C(x)}$. Furthermore, for every x, $C(x)$ is an integer, and its bit-length is obviously bounded by the size of C (which is n^c). This means that $N_0 = \sum_{x \in \{0,1\}^n} C(x)$ is an integer constant of bit-length at most n^{c+1}.

We plan to use C to construct a circuit T_z that distinguishes the PRG G from uniform. We will construct this circuit in stages.

Let A be an $\text{NP}_{||}$-circuit that is hardwired with N_0, and given $x \in \{0,1\}^n$, outputs $\frac{R(x)}{N_0} = \Pr[X' = x]$. Note that A is an $\text{NP}_{||}$-circuit of size $n^{c+O(1)}$.

Let $\tau = 2^{-(n-\Delta)}$, and let $B(x)$ be the randomized $\text{NP}_{||}$-circuit that on input x, computes $\alpha(x) = \frac{A(x)}{\tau}$, tosses a coin V according to the distribution $(\alpha(x), 1-\alpha(x))$, and outputs V. (Note that this is well defined as $0 \leq \alpha(x) \leq 1$, because $A(x) = \Pr[X' = x] \leq \tau$ for every $x \in \{0,1\}^n$).

We now define a randomized $\text{NP}_{||}$-circuit $T_z : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ as follows: Let $T_z(x,r)$ be the randomized $\text{NP}_{||}$-circuit, which answers one iff $B(x) = 1$ and $r = z$. Note that T_z is of size $n^{c+O(1)}$.

We plan to show that T_z distinguishes the output of G from the uniform distribution. More formally, let (X, R) be uniform over $\{0,1\}^n \times \{0,1\}^m$, we will get a contradiction if we show that

$$|\Pr[T_z(X,R)=1] - \Pr[T_z(X,G(X))] > \frac{\epsilon}{2^{m+\Delta}}.$$

We now proceed towards this goal. The definition of B was made so that:

Claim 3.17.

- $\Pr[B(X)=1] = 2^{-\Delta}$.
- $\Pr[E(X')=z] = \Pr[E(X)=z|B(X)=1]$.

Proof. (of claim 3.17) For the first item, we observe that:

$$\begin{aligned}
\Pr[B(X)=1] &= \sum_{x\in\{0,1\}^n} \Pr[B(X)=1 \text{ and } X=x] \\
&= \sum_{x\in\{0,1\}^n} \Pr[X=x]\cdot\Pr[B(X)=1|X=x] \\
&= \sum_{x\in\{0,1\}^n} \Pr[X=x]\cdot\Pr[B(x)=1] \\
&= \sum_{x\in\{0,1\}^n} 2^{-n}\cdot\frac{A(x)}{\tau} \\
&= \frac{2^{-n}}{\tau}\cdot\sum_{x\in\{0,1\}^n} A(x) \\
&= 2^{-\Delta}.
\end{aligned}$$

Here, the last equality follows because $\tau = 2^{-(n-\Delta)}$ and $\sum_{x\in\{0,1\}^n} A(x) = \sum_{x\in\{0,1\}^n} \Pr[X'=x] = 1$.

For the second item, we observe that it is sufficient to prove that for every $x \in \{0,1\}^n$:

$$\Pr[X=x|B(X)=1] = \Pr[X'=x].$$

This is because, this implies that X' is distributed like $(X|B(X)=1)$, and then it follows that

$$\Pr[E(X)=z|B(X)=1] = \Pr[E(X')=z].$$

We now observe that for every $x \in \{0,1\}^n$,

$$\begin{aligned}
\Pr[X=x|B(X)=1] &= \frac{\Pr[X=x \text{ and } B(X)=1]}{\Pr[B(X)=1]} \\
&= \frac{\Pr[X=x \text{ and } B(x)=1]}{2^{-\Delta}} \\
&= 2^{\Delta}\cdot\Pr[X=x]\cdot\Pr[B(x)=1] \\
&= 2^{\Delta}\cdot 2^{-n}\cdot\frac{A(x)}{\tau} \\
&= \Pr[X'=x].
\end{aligned}$$

Here, the last inequality follows because $\tau = 2^{-(n-\Delta)}$ and $A(x) = \Pr[X'=x]$. □

We proceed to compute the probability that T_z outputs one on (X, R) and on $X, E(X)$.

$$\begin{aligned}\Pr[T_z(X,R)=1] &= \Pr[B(X)=1 \text{ and } R=z] \\ &= \Pr[B(X)=1]\cdot\Pr[R=z] \\ &= 2^{-\Delta}\cdot 2^{-m}.\end{aligned}$$

Here, the last equality follows by the first item in Claim 3.17.

$$\begin{aligned}\Pr[T_z(X,E(X))=1] &= \Pr[B(X)=1 \text{ and } E(X)=z] \\ &= \Pr[B(X)=1]\cdot\Pr[E(X)=z|B(X)=1] \\ &= 2^{-\Delta}\cdot\Pr[E(X')=z].\end{aligned}$$

Here, the last equality follows by the first and second items in Claim 3.17. Altogether,

$$\begin{aligned}|\Pr[T_z(X,R)=1]-\Pr[T_z(X,E(X))=1]| &= |2^{-\Delta}\cdot 2^{-m}-2^{-\Delta}\cdot\Pr[E(X')=z]| \\ &= 2^{-\Delta}\cdot|2^{-m}-\Pr[E(X')=z]| \\ &> 2^{-\Delta}\cdot\epsilon\cdot\frac{1}{2^m} \\ &= \frac{\epsilon}{2^{m+\Delta}},\end{aligned}$$

By fixing the random coins of T_z to some fixed value, we can obtain a (non-randomized) NP $_{||}$-circuit of the same size that distinguishes the two distribution, and as this size is $n^{c+O(1)}$ we indeed get a contradiction. □

Lemma 3.15 requires a PRG with unusual parameters. Loosely speaking, such a PRG is "less competitive" than that of Theorem 2.8, and so it is easy to obtain such a PRG from Theorem 2.8. This is done in the next corollary.

Corollary 3.18. *If E is hard for exponential size nondeterministic circuits then for every constant $c_1, c_2 > 1$, and for every sufficiently large n, there is a $G : \{0,1\}^n \to \{0,1\}^{c_1\cdot\log n}$ that is a seed-extending $\frac{1}{n^{c_2}}$-PRG for NP$_{||}$-circuits of size n^{c_2}. Furthermore, G is computable in time* $\mathrm{poly}(n^{c_2})$.

Proof. We use the PRG that follows from Theorem 2.8, using c_2 as the constant c. The seed extending PRG that we obtain stretches $a\log n$ bits to n bits. We can chop the output length from n to the required $c_1\cdot\log n$ bits, and obtain a seed extending n^{-c_2}-PRG $G : \{0,1\}^{a\log n} \to \{0,1\}^{c_1\cdot\log n}$ for NP$_{||}$-circuits of size n^{c_2}. We can then increase the seed length from $a\log n$ to n (without harming the seed-extending property) by taking $G' : \{0,1\}^n \to \{0,1\}^{c_1\cdot\log n}$ to be $G'(x) = G(x|_{[a_1\cdot\log n]})$ (that is, by artificially increasing the seed length). □

Together, Lemma 3.15 and Corollary 3.18 give Theorem 3.14.

Proof. (of Theorem 3.14) Given constants a, c, c_1 as in Theorem 3.14, let c_0 be the universal constant from Lemma 3.15, and set $c_2 = \max(c+a+c_1, c_0\cdot c)$. We apply Corollary 3.18 to obtain an $E : \{0,1\}^n \to \{0,1\}^{c_1\cdot\log n}$ that is a seed-extending $\frac{1}{n^{c_2}}$-PRG for NP$_{||}$-circuits of size n^{c_2}. We have that $G(x)$ is seed extending, and so by Lemma 3.15 we obtain that taking $\Delta = a\cdot\log n$, G is an $(n - a\log n, \epsilon)$-relative error extractor for distributions that are weakly recognizable by size n^{c_2/c_0} NP$_{||}$-circuits, for

$$\epsilon = \frac{2^{a\log n + c_1\log n}}{n^{c_2}} = 2^{(a+c_1-c_2)\log n} \le n^{-c}.$$

By our choices we also have that $c_2/c_0 > c$ as required. □

3.5.3 Proof of Theorem 3.5

We will construct the required HTS in two steps, we first construct a shrinking HTS with very large h of $h = 2^{n-O(\log n)}$. This is done by taking the HTS from Theorem 3.6 and applying Lemma 3.13 using the suitable extractor for recognizable distributions from Theorem 3.14.

Lemma 3.19. *if E is hard for exponential size nondeterministic circuits then for every constant $c > 1$, there exists a constant $c_1 > 1$ such that for every constants $a > 1$ and $\eta > 0$, there exists a constant d such that for every sufficiently large n, there is a function $f : \{0,1\}^{\eta \cdot n} \to \{0,1\}^{2 \cdot c_1 \cdot \log n}$ that is an (h', ρ')-HTS for circuits of size n^c, where $h' = 2^{\eta \cdot n - a \log n + c \log n + O(1)}$ and $\rho' = 2 \cdot n^{-c}$. Furthermore, f is computable in time* $\mathrm{poly}(n^d)$.

Proof. We use the hardness assumption to apply Theorem 3.6 and obtain an (h, ρ)-HTS $f_1 : \{0,1\}^m \to \{0,1\}^{m'}$ for circuits of size n^c with the parameters specified in the theorem. In particular, we have that $m' \leq m = c_1 \cdot \log n$ for some constant $c_1 > 1$, and there exists a constant $\gamma > 0$ such that $h = 2^{(1-\gamma) \cdot m}$ and $\rho = n^{-c}$. Theorem 3.6 also guarantees that $h \leq \frac{\rho \cdot 2^m}{2}$.

Let $c' = c \cdot c_0$, where c_0 is the universal constant from Lemma 3.13. We will use the assumption to apply Theorem 3.14 and obtain a function $E : \{0,1\}^{\eta \cdot n} \to \{0,1\}^m$ that is a $(\eta \cdot n - a \cdot \log n, \frac{1}{n^{c'}})$-relative-error extractor for distributions recognizable by size $n^{c'}$ $\mathrm{NP}_{||}$-circuits. We will use the transformation T_{E,f_1} of Definition 3.11 (with $D = 1$) to obtain a function $f_2 = T_{E,f_1}$. We plan to apply Lemma 3.13, and have chosen the constant c' to be sufficiently large, so that we meet the conditions of the lemma, and obtain a function $f_2 : \{0,1\}^{\eta n} \to \{0,1\}^{m+m'}$ that is an (h', ρ')-HTS with $h' = \frac{2^{\eta n - a \log n + 2}}{n^{-c}}$ and $\rho' = 2n^{-c}$. Note that as $m' \leq m$ we have that the output length $m + m' \leq 2 \cdot c_1 \cdot \log n$. We also have that f_2 can be computed in polynomial time, where the polynomial n^d depends on the choices of the constants. □

We now prove that Theorem 3.5 follows from Lemma 3.19.

Proof. (of Theorem 3.5) Given $c > 1$, $\delta > 0$ and $0 < \lambda < 1$, we set $\eta = \delta/2$. At this point, the constant δ determines a constant b (that is guranteed in Theorem 2.25). We now choose a large constant a (that will be determined later) and apply Lemma 3.19 to obtain the function $f_2 : \{0,1\}^{\eta n} \to \{0,1\}^{2c_1 \log n}$ that is guaranteed by the lemma. This function is an (h', ρ')-HTS for circuits of size n^c, where $h' = 2^{\eta \cdot n - a \log n + c \log n + O(1)}$ and $\rho' = 2 \cdot n^{-c}$.

We will use the strong disperser of Theorem 2.25 with the constant $\delta > 0$ from the theorem statement, and $\epsilon(n) = 1 - 2^{-a \log n}$. We obtain a function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ that is a $(\delta \cdot n, \epsilon(n))$-strong disperser with $m = \eta n$, and $d = \log D$ for

$$D = \frac{bn}{\log \frac{1}{1-\epsilon}} = \frac{bn}{a \log n}.$$

The key observation is that as c_1, b and λ are already fixed, we are allowed to choose a to be sufficiently large so that:

$$D = \frac{bn}{a \log n} \leq \frac{\lambda \cdot n}{2 \cdot c_1 \log n}.$$

By Proposition 2.23 we can interpret E as a function $E : \{0,1\}^n \to (\{0,1\}^m)^D$ that is $(\ell, 2^{\delta n})$-list recoverable for every

$$\ell < (1 - \epsilon) \cdot 2^m = 2^{-a \log n} \cdot 2^{\eta n} = 2^{\eta n - a \log n} \leq h'.$$

In particular, we get that E is $(h', 2^{\delta n})$-list recoverable.

We now apply the transformation T'_{E,f_2} of Definition 3.11 on the function f_2 from Lemma 3.19, using Lemma 3.12. The running time of E (as a list-recoverable code) is D times its running time (as a disperser),

and as $D \leq n$, using Theorem 2.25, this is bounded by some fixed polynomial in n, and we can assume w.l.o.g. that n^c is larger than this polynomial. We use Lemma 3.12 to conclude that $f_3 = T'_{E,f_2} : \{0,1\}^n \to \{0,1\}^{D \cdot 2 \cdot c_1 \cdot \log n}$ is an (h, ρ)-HTS for circuits of size n^c for $h = 2^{\delta n}$ and $\rho = D \cdot \rho' \leq 2n^{-(c-1)}$ (and we could have gotten n^{-c} had we increased the c we intially used). The output length of f_3 is indeed

$$D \cdot 2 \cdot c_1 \cdot \log n \leq \lambda n.$$

Finally, by the efficiency of f and E, we conclude that $f_3 = T'_{E,f}$ can be computed in time polynomial in n. □

4 Evasive BSC codes

As explained in Section 1.3.5, evasive BSC codes were introduced in [SS21b] (for a different class of channels, namely space bounded channels) and in [SS22] for poly-size channels. In [SS22], the existence of such codes was proven using the probabilistic method (to yield a Monte-Carlo randomized construction). In this section we give an explicit construction of such codes under hardness assumptions. These codes will be used as a component in the construction of stochastic codes for poly-size circuits that is presented in Section 6.

4.1 Definition of evasive codes

We will be interested in codes that in additional to certain decoding properties, also have an evasiveness property that we now define.

Definition 4.1 (Evasive codes [SS22])**.** *Let* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ *and* $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{fail\}$. *We say that* $(\mathrm{Enc}, \mathrm{Dec})$ *are* ρ*-evasive for a function* $C : \{0,1\}^n \to \{0,1\}^n$ *if:*

$$\Pr_{Z \leftarrow U_n}[\mathrm{Dec}(Z \oplus C(Z)) \neq fail] \leq \rho.$$

We say that $(\mathrm{Enc}, \mathrm{Dec})$ *are* ρ*-evasive for a class* $\mathcal{C}$ *if* $(\mathrm{Enc}, \mathrm{Dec})$ *are* ρ*-evasive for every* C *in* $\mathcal{C}$.

As in [SS22] we will be interested in constructing evasive code for a variant of the BSC that emerges in this line of work. More specifically, we will be interested in codes for errors induced by a random permutation (defined in Section 2.6).

4.2 Explicit constructions of evasive codes for random permutations

The next theorem gives a conditional explicit construction of codes for random permutations that are evasive for poly-size channels. These codes are constructed under hardness assumptions, and have the optimal rate of $1 - H(p)$.

Theorem 4.2. *If E is hard for exponential size nondeterministic circuits then for every constant* $c > 1$*, every constant* $0 < p < \frac{1}{4}$*, and every sufficiently small constant* $\epsilon > 0$*, there exists a constant* $d > 1$ *such that for* $R = 1 - H(p) - \epsilon$*, and for infinitely many* n*, there are functions* $\mathrm{Enc} : \{0,1\}^{Rn} \to \{0,1\}^n$ *and* $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^{Rn} \cup \{fail\}$*, such that:*

- $(\mathrm{Enc}, \mathrm{Dec})$ *decode from* $\mathrm{Perm}_p^{\mathrm{UniPerm}}$ *with success probability* $1 - 2^{-\Omega(n^{0.1})}$.
- $(\mathrm{Enc}, \mathrm{Dec})$ *are* $\frac{1}{n^c}$*-evasive for* $\mathrm{Ckt}_p^{n^c}$.

- *There exists a universal constant a_0, such that for every $m \in \{0,1\}^k$, there exists a circuit A_m of size n^{a_0}, and for every $e \in \{0,1\}^n$,*

$$A_m(e) = \begin{cases} 1, & \mathrm{Dec}(\mathrm{Enc}(m) \oplus e) = m \\ 0, & \mathrm{Dec}(\mathrm{Enc}(m) \oplus e) \neq m \end{cases}$$

Furthermore, Enc *and* Dec *can be computed in time n^d.*

Theorem 4.2 is proven in Section 4.2.1. The proof of Theorem 4.2 works by applying Theorem 2.33 (which gives a code with the first property) and the second property is obtained by strengthening the code with a suitable HTS that is obtained under the hardness assumption.

Remark 4.3. *The reason that Theorem 4.2 is stated for "infinitely many n", rather than for "every sufficiently large n" is that we rely on Theorem 2.33 which is stated for "infinitely many n. As explained in Remark 2.34, it seems likely that a careful inspection of the components of Theorem 2.33 will reveal that the set of infinitely many n achieved there is very dense (and possibly even the set of "every sufficiently large n"). Such a statement would carry over to Theorem 4.2.*

We also remark that Somewhat surprisingly, if we are willing to settle for a larger rate $R^*(p)$ (such that $R^*(p) = R^{GV}(p) = 1 - H(2p)$ achieved by the Gilbert-Varshamov bound, then it is possible to give an unconditional *construction that is evasive for Hamming channels (that are computationally unbounded). This was shown in [SS22] for $R^*(p) = (1 - H(p)) \cdot \frac{1-4p}{1-2p}$ which is larger than $1 - H(2p)$ for every $0 < p < \frac{1}{4}$.*

In the remainder of this section we prove Theorem 4.2.

4.2.1 Proof of Theorem 4.2

We are given constants c, p and ϵ as in the theorem statement. We choose $\lambda = \frac{\epsilon}{2}$ and use the assumption that E is hard for exponential size nondeterministic circuits to apply Theorem 3.5 and obtain that for every sufficiently large k, there is a function $f : \{0,1\}^k \to \{0,1\}^{\lambda \cdot k}$ that is a $(h, \frac{1}{k^{c+1}})$-HTS for circuits of size k^{c+1}, with $h = 2^{\delta k}$ where we choose $\delta = \frac{1-H(2p)}{2}$ and note that $\delta > 0$ as $p < \frac{1}{4}$.

Let $k' = k + \lambda \cdot k$, and let $R = 1 - H(p) - \frac{\epsilon}{2}$. Let $n = k'/R$. We apply Theorem 2.33 to obtain that for infinitely many n, there are functions $\mathrm{Enc} : \{0,1\}^{k'} \to \{0,1\}^n$ and $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^{k'}$ such that $(\mathrm{Enc}, \mathrm{Dec})$ decode from $\mathrm{Perm}_p^{\mathrm{UniPerm}}$ with success probability $1 - 2^{-\Omega(n^{0.1})}$. We also have that these functions are explicit, and therefore, have circuits of size $\mathrm{poly}(n)$. We can assume w.l.o.g. that c is sufficiently large so that Enc, Dec have circuits of size n^c.

We will construct a pair of function $(\mathrm{Enc}', \mathrm{Dec}')$ where $\mathrm{Enc}' : \{0,1\}^k \to \{0,1\}^n$ and $\mathrm{Dec}' : \{0,1\}^n \to \{0,1\}^k \cup \{fail\}$ as follows:

- $\mathrm{Enc}'(x) = \mathrm{Enc}(x \circ f(x))$ and note that indeed $|x \circ f(x)| = k + \lambda \cdot k = k'$.
- $\mathrm{Dec}'(v)$ operates as follows:
  - Apply $\mathrm{Dec}(v)$ to obtain a string of length k' which we interpret as a pair $(x, y) \in \{0,1\}^k \times \{0,1\}^{\lambda \cdot k}$.
  - If $y \neq f(x)$ then $\mathrm{Dec}'(v)$ output "fail".
  - If $y = f(x)$, and $\delta(v, \mathrm{Enc}(x \circ f(x))) > p$ output "fail".
  - Finally, output x, if we reached this stage. (That is, if $y = f(x)$ and $\delta(v, \mathrm{Enc}(x \circ f(x))) \leq p$.

We will show that the pair $(\mathrm{Enc}', \mathrm{Dec}')$ meet the requirements of the theorem. First of all, we observe that the rate of $(\mathrm{Enc}, \mathrm{Dec})$ is:

$$\frac{k}{n} = \frac{k' - \lambda \cdot k}{n} = 1 - H(p) - \frac{\epsilon}{2} - \frac{\lambda \cdot k}{n} \geq 1 - H(p) - \frac{\epsilon}{2} - \frac{\epsilon}{2} = 1 - H(p) - \epsilon,$$

as required.

All error patterns e supported by $\mathrm{Perm}_p^{\mathrm{UniPerm}}$ have $\mathsf{wt}(e) \leq p$. This means that for every m,

$$\delta(\mathrm{Enc}(m), \mathrm{Enc}(m) \oplus e) \leq p,$$

and so when Dec' decodes a given string $v = \mathrm{Enc}(m) \oplus e$, it can safely fail if $\delta(\mathrm{Enc}(\mathrm{Dec}(v)), v) > p$, without harming the behavior against error patterns in the support of $\mathrm{Perm}_p^{\mathrm{UniPerm}}$. This means that we do not hurt the decoding properties of $(\mathrm{Enc}, \mathrm{Dec})$ against $\mathrm{Perm}_p^{\mathrm{UniPerm}}$, and can therefore conclude that $(\mathrm{Enc}', \mathrm{Dec}')$ decode from $\mathrm{Perm}_p^{\mathrm{UniPerm}}$ with success probability $1 - 2^{-\Omega(n^{0.1})}$.

It remains to prove that $(\mathrm{Enc}, \mathrm{Dec})$ are $\frac{1}{n^c}$-evasive for $\mathrm{Ckt}_p^{n^c}$. Let $C \in \mathrm{Ckt}_p^{n^c}$ be a channel. We will show that

$$\Pr_{Z \leftarrow U_n}[\mathrm{Dec}'(Z \oplus C(Z)) \neq fail] \leq \frac{1}{n^c}.$$

For this purpose we consider a sampling circuit A that works as follows:

- A samples $Z \leftarrow U_n$.
- It compute $Z \oplus C(Z)$.
- It applies Dec on $Z \oplus C(Z)$ to obtain a string of length k' which we interpret as $(X, Y) \in \{0,1\}^k \times \{0,1\}^{\lambda \cdot k}$.
- Finally, A outputs (X, Y).

Note that by definition, A has size $O(n^c) \leq k^{c+1}$. This means that f is secure against A, and there exists a set $H \subseteq \{0,1\}^k$ of size at most h, such that:

$$\Pr_{(X,Y) \leftarrow A}[X \notin H \text{ and } Y = f(X)] \leq \frac{1}{k^{c+1}}.$$

As the channel C produces an error pattern with relative weight at most p, and $\mathrm{Dec}'(v)$ fails if $\delta(\mathrm{Enc}(\mathrm{Dec}(v)), c) > p$, we have that in the experiment described above:

$$\left\{\mathrm{Dec}'(Z \oplus C(Z)) \neq fail\right\} \Rightarrow \left\{Y = f(X) \text{ and } \delta(\mathrm{Enc}(X \circ f(X)), Z) \leq 2p\right\}.$$

It follows that:

$$\begin{aligned}
\Pr_{Z\leftarrow U_n}[\mathrm{Dec}'(Z\oplus C(Z))\neq fail] &\leq \Pr[Y=f(X) \text{ and } \delta(\mathrm{Enc}(X\circ f(X)),Z)\leq 2p] \\
&= \Pr[X\in H \text{ and } Y=f(X) \text{ and } \delta(\mathrm{Enc}(X\circ f(X)),Z)\leq 2p] \\
&\quad + \Pr[X\notin H \text{ and } Y=f(X) \text{ and } \delta(\mathrm{Enc}(X\circ f(X)),Z)\leq 2p] \\
&\leq \Pr[X\in H \text{ and } \delta(\mathrm{Enc}(X\circ f(X)),Z)\leq 2p] + \Pr[X\notin H \text{ and } Y=f(X)] \\
&\leq \sum_{x\in H}\Pr[X=x \text{ and } \delta(\mathrm{Enc}(X\circ f(X)),Z)\leq 2p] + \frac{1}{k^{c+1}} \\
&\leq \sum_{x\in H}\Pr[\delta(\mathrm{Enc}(x\circ f(x)),Z)\leq 2p] + \frac{1}{k^{c+1}} \\
&\leq h\cdot 2^{-n\cdot(1-H(2p))} + \frac{1}{n^{c+1}} \\
&\leq 2^{\frac{(1-H(2p))k}{2}}\cdot 2^{-n\cdot(1-H(2p))} + \frac{1}{k^{c+1}} \\
&\leq 2^{\frac{-n\cdot(1-H(2p))}{2}} + \frac{1}{k^{c+1}} \\
&\leq \frac{1}{n^c}.
\end{aligned}$$

Where the last inequality follows because $p<\frac{1}{4}$, and $k=R\cdot n$.

Finally, for the third item we note that for every m, the circuit A_m can be hardwired with $m, f(m)$ and $z=\mathrm{Enc}(m)$. This means that $A_m(e)$ can compute $v=z\oplus e$ and $\mathrm{Dec}(v)$ without having to compute f. It can then apply Dec' on v to obtain x,y, and by comparing (x,y) to $(m,f(m))$, and checking whether $\delta(v,\mathrm{Enc}(m,f(m)))\leq p$, it can answer whether $\mathrm{Dec}(\mathrm{Enc}(m)\oplus e)=m$. The point is that while this requires A_m to compute Dec', A_m does not need to compute f, and therefore (by the explicitness of $(\mathrm{Enc}',\mathrm{Dec}')$ this can be done by a circuit of size n^{a_0}, where the a_0 is universal and does not depend on c.

5 SS-non-malleable codes

The notion of small set non-malleable codes was introduced in [SS22] as a component for the stochastic codes for poly-size channels. In [SS22], the existence of SS-non-malleable codes was proven using the probabilistic method (to yield a Monte-Carlo randomized construction). In this section we will define a notion of SS-non-malleability that is weaker than the one considered in [SS22], but still sufficient for the application of stochastic codes for poly-size channels. We will give an explicit construction of such codes under hardness assumptions. These codes will be used as a component in the construction of stochastic codes for poly-size circuits that is presented in Section 6.

5.1 Definitions of non-malleable and SS-non-malleable codes

Non-malleable codes are stochastic codes that provide some weak decoding guarantee, even against powerful adversaries that can completely modify the codeword. The next definition makes the baseline requirement that codes should decode correctly if no errors are introduced.

Definition 5.1. *A stochastic code (that is a pair of maps* $Enc:\{0,1\}^k\times\{0,1\}^d\to\{0,1\}^n$*, and* $\mathrm{Dec}:\{0,1\}^n\to\{0,1\}^k\cup\{fail\}$*) is* **valid** *if for every* $m\in\{0,1\}^k$ *and every* $s\in\{0,1\}^d$*,* $\mathrm{Dec}(\mathrm{Enc}(m,s))=m$*.*

In a seminal paper, Dziembowski, Pietrzak and Wichs [DPW18] introduced the notion of non-malleable codes. Loosely speaking, such codes are valid stochastic codes with an additional property that any adversary in the considered class cannot make the decoding algorithm decode to a message that is different and yet correlated with the original message.

Definition 5.2 (non-malleable codes [DPW18])**.** *A valid stochastic code (that is a pair of maps* $Enc : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^n$*, and* $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{fail\}$*) is* ϵ**-SS-non-malleable** *for a class* $\mathcal{C}$*, if for every* $C \in \mathcal{C}$*, there exists a distribution* D_C *that is supported on* $\{0,1\}^k \cup \{\mathrm{same}\}$*, and for every* $m \in \{0,1\}^k$*,*

$$\Delta(\mathrm{Dec}(C(\mathrm{Enc}(m, U_d))); \mathrm{Copy}(D_C, m)) \leq \epsilon,$$

where the function Copy *is defined by:*

$$\mathrm{Copy}(x, y) = \begin{cases} x, & x \neq \mathrm{same} \\ y, & x = \mathrm{same} \end{cases}$$

We say that the stochastic code is **simulatable** *by size* s*-circuits, if for every* $C \in \mathcal{C}$*, there exist a size* s *circuit* A_C *that samples* D_C*.*

Shaltiel and Silbak [SS22] defined a notion of SS-non-malleable codes that considers a scenario in which one is interested in encoding a random message $X \leftarrow U_k$, and the adversary C sees additional information $\psi(S)$ in addition to the encoding of S. Here, the focus is on functions ψ which are injective, and therefore (at least information theoretically) leak all the information about S. (In this paper, we will be interested in the case that ψ is a PRG that fools the adversary). The security guarantee is defined using a different approach than the one used in the seminal work of [DPW18]. The approach is similar in spirit to the way we defined an HTS in this paper (and in fact, our definition of HTS came out of the definition of [SS22]). It is required that for every adversary C, there exists a small set H of messages, such that it is unlikely that C can make the decoding decode to a message that is neither the original message, nore in H. (We remark that some works in non-malleable codes [FMVW16, JW15] considered concepts that are somewhat similar. See [SS22] for a discussion).

Definition 5.3 (SS-non-malleable codes (modified version of [SS22]))**.** *Let* ψ *be a function that on input* $s \in \{0,1\}^k$ *returns a string. A valid stochastic code (that is a pair of maps* $Enc : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^n$*, and* $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{fail\}$*) is* (ψ, h, ρ)**-SS-non-malleable** *for a class* $\mathcal{C}$*, if for every* $C \in \mathcal{C}$*, there exists a set* $H_C \subseteq \{0,1\}^k$*, with* $|H_C| \leq h$*, such that:*

$$\Pr_{X \leftarrow \{0,1\}^k, S \leftarrow \{0,1\}^d} [\mathrm{Dec}\left(C(\psi(X), \mathrm{Enc}(X, S))\right)) \notin H_C \cup \{X\} \cup \{fail\}] \leq \rho.$$

If we omit ψ*, then* ψ *is the constant function that outputs the empty string (meaning that* C *only receives* $\mathrm{Enc}(X, S)$*).*

This definition is somewhat different than the one used in [SS22], and the differences are detailed below.

Remark 5.4 (Comparison of this definition to [SS22])**.** *The definition of [SS22] has an additional integer parameter* v*, and it is required that security holds even if* C *observes* $\psi(S)$ *and* v *encodings of* X *(using* v *independent seeds* $S_1, \ldots, S_v$*). This added security was used in the construction of stochastic codes for poly-size channels of [SS22]. We work with a weaker definition (which corresponds to the case* $v = 1$ *in [SS22]) and will therefore need to modify the construction of stochastic codes for poly-size circuits that is given in Section 6.*

An additional difference is that [SS22] gives a probabilistic argument showing that there exist codes that are SS-non-malleable (for a function ψ *that gives more information about* S*, than a PRG). We will only be*

able to argue about the case that ψ is a seed-extending PRG (and this will require additional modification in the final construction of stochastic codes for poly-size circuits).

Finally, in [SS22] the probabilistic argument also gives that the stochastic code is pseudorandom, and decodes from small radius (in the sense defined in Definition 2.11). We will not be able to reproduce these properties for the SS-non-malleable code, and this is an additional complication that we will need to handle in the construction 6.

5.2 An explicit construction of SS-non-malleable codes under hardness assumptions

The main result of this section is an explicit construction of an SS-non-malleable code that is secure whenever the function ψ is a sufficiently strong seed-extending PRG.

Theorem 5.5. *There exists a constant $c_0 > 1$ such that if E is hard for exponential size nondeterministic circuits then for every constant $c > 1$, there exist constants $c' > d' \geq 1$, and constants $a' \geq 1$ such that for every constant $a \geq a'$ and for every sufficiently large n, setting $k = a \cdot \log n$, $k' = k^{c_0}$ and $d = n^{d'}$, there are functions $\mathrm{Enc} : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^{k'}$, and $\mathrm{Dec} : \{0,1\}^{k'} \to \{0,1\}^k \cup \{fail\}$) such that for every function $G : \{0,1\}^k \to \{0,1\}^n$ that is a seed-extending $\frac{1}{n^{c'}}$-PRG for circuits of size $n^{c'}$, the pair $(\mathrm{Enc}, \mathrm{Dec})$ is a $(G, n^{a'}, \frac{1}{n^c})$-SS-non-malleable codes for randomized circuits of size n^c. Furthermore, $\mathrm{Enc}, \mathrm{Dec}$ can be computed in time $n^{d'}$.*

The remainder of this section is devoted to proving Theorem 5.5. We build on recent results by Ball, Dachmman-Soled and Loss [BDL22] that give an explicit construction of non-malleable codes under hardness assumptions.

5.3 The non-malleable codes of Ball, Dachman-Soled and Loss

Recently, Ball, Dachman-Soled and Loss gave an explicit construction of non-malleable codes for poly-size circuits under hardness assumptions. Our construction of SS-non-malleable codes will use this result.

Theorem 5.6 (non-malleable under hardness assumptions [BDL22])**.** *If E is hard for nondeterministic circuits, then for every constant $c > 1$, there exist a constant $d' > 1$ such that for every sufficiently large n, and every $c \cdot \log n < k < n$, setting $k' = k^8$ and $d = n^{d'}$, there are functions $\mathrm{Enc} : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^{k'}$, and $\mathrm{Dec} : \{0,1\}^{k'} \to \{0,1\}^k \cup \{fail\}$) such that the pair $(\mathrm{Enc}, \mathrm{Dec})$ is a $\frac{1}{n^c}$-non-malleable code for circuits of size n^c. Furthermore, $\mathrm{Enc}, \mathrm{Dec}$ can be computed in time $n^{d'}$.*

We remark that the codes of [BDL22] can also encode short strings, and the requirement that $k > c \cdot \log n$ is made in order to simplify the statement.

The guarantee obtained by [BDL22] is statistical, and they do not claim in their statement that their code is efficiently simulatable. Nevertheless, it is standard to show that such a code is efficiently simulatable.

Proposition 5.7. *For every constant $c > 1$, there exists a constant $e > 1$ such that the non-malleable code of Theorem 5.6 is simulatable by circuits of size n^e. Furthermore, the statement holds not only for deterministic circuits of size n^c, but also for randomized circuits of size n^c.*

For completeness, we provide a proof of this proposition.

Proof. For every circuit C of size n^c, we will construct a circuit A_C that samples the distribution D_C. For this purpose, we define the function Uncopy as follows:

$$\mathrm{Uncopy}(x, y) = \begin{cases} same, & x = y \\ y, & x \neq y \end{cases}$$

Let A_C be the circuit that chooses $M \leftarrow U_k$, and computes $\bar{M} = \mathrm{Dec}(C(\mathrm{Enc}(M, U_d)))$, and outputs $\hat{M} = \mathrm{Uncopy}(M, \bar{M})$. Note that this circuit is indeed of size n^e for some constant e that depends on the constant c and the d' from Theorem 5.6 (which in turn depends on c).

The intuition is that as $\bar{M}$ is statistically close to $\mathrm{Copy}(D_C, M)$, we can hope to obtain a distribution that is close to D_C when applying Uncopy on both sides. However, we should be careful as Uncopy does not always reverse the operation of Copy. What is true (and easy to check) is that for every $x \in \{0,1\}^k \cup \{\mathrm{same}\}$ and $y \in \{0,1\}^k$, we have that

$$\mathrm{Uncopy}(y, \mathrm{Copy}(x, y)) \neq x \Leftrightarrow x = y.$$

We can therefore conclude that:

$$\mathrm{Uncopy}(M, \mathrm{Copy}(D_C, M)) \neq D_C \Leftrightarrow D_C = M.$$

However, as M is uniform over $\{0,1\}^k$, and is independent of D_C, the latter event occurs with probability 2^{-k}.

By the definition of D_C and the fact that it is independent of M, we have that for every $m \in \{0,1\}^k$

$$\Delta((\bar{M}|M = m); \mathrm{Copy}(D_C, m)) \leq \frac{1}{n^c}.$$

Which implies that:

$$\Delta((M, \bar{M}); (M, \mathrm{Copy}(D_C, M))) \leq \frac{1}{n^c}.$$

We can now apply the function Uncopy to both sides and conclude that:

$$\Delta(\mathrm{Uncopy}(M, \bar{M}); \mathrm{Uncopy}(M, \mathrm{Copy}(D_C, M))) \leq \frac{1}{n^c}.$$

Which gives that:

$$\Delta(\hat{M}; \mathrm{Uncopy}(M, \mathrm{Copy}(D_C, M))) \leq \frac{1}{n^c}.$$

However, we have already seen that the distribution on the right is identical to D_C except for with probability 2^{-k}. Overall, we conclude that:

$$\Delta(\hat{M}; D_C) \leq \frac{1}{n^c} + \frac{1}{2^k}.$$

This means that the distribution sampled by A_C shows the non-malleability of the code with error

$$\epsilon = \frac{1}{n^c} + \left(\frac{1}{n^c} + \frac{1}{2^k}\right) \leq \frac{3}{n^c},$$

where the latter inequality follows by our requirements on k. This holds also if C is a randomized circuit, as the simulating circuit A can toss the randomness for C, and for every choice of random coins r for C, the simulating circuit A (with fixed r) produces a distribution $D_{C,r}$ that is good for C (with fixed r). Finally, the difference between $\frac{1}{n^c}$ and $\frac{3}{n^c}$ is immaterial in the statemenmt of Theorem 5.6. □

5.4 Using HTS to convert non-malleable codes into SS-non-malleable codes

We will show how to convert the non-malleable codes into SS-non-malleable codes in two steps. In the first step we will use an HTS to convert an efficiently simulatable non-malleable code into an SS-non-malleable code for the constant function ψ that outputs an empty string. We will then show how to convert such an SS-non-malleable code into one that is secure when G is a sufficiently strong seed-extending PRG.

The next definition gives the construction for the first step.

Definition 5.8. *Let k, d, n, m' be parameters. Given:*

- *A function $f : \{0,1\}^k \to \{0,1\}^{m'}$.*
- *Functions $\mathrm{Enc} : \{0,1\}^{k+m'} \times \{0,1\}^d \to \{0,1\}^n$, and $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^{k+m'} \cup \{fail\}$)*

We define a pair of functions $\mathrm{Enc}_f : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^n$ and $\mathrm{Dec}_f : \{0,1\}^n \to \{0,1\}^k$ as follows:

- $\mathrm{Enc}_f(x, s) = \mathrm{Enc}(x \circ f(x), s)$.
- $\mathrm{Dec}_f(z)$ *works by applying* $\mathrm{Dec}(z)$. *If* $\mathrm{Dec}(z)$ *does not fail, then it outputs $x \circ y \in \{0,1\}^{k+m'}$ and* Dec_f *outputs x if $y = f(x)$, and fails otherwise.*

We now claim that using an HTS, the following construction indeed produces an SS-non-malleable code (where ψ is the function that outputs the empty string).

Lemma 5.9. *Let k, d, n, m' be parameters, and let* $(\mathrm{Enc}, \mathrm{Dec})$ *and f be as in Definition 5.8. If*

- $(\mathrm{Enc}, \mathrm{Dec})$ *is an ϵ-non-malleable code for randomized circuits of size s, and is simulatable by circuits of size s', and*
- *f is an (h, ρ)-HTS for circuits of size s'.*

Then $(\mathrm{Enc}_f, \mathrm{Dec}_f)$ *is an $(h, \epsilon + \rho)$-SS-non-malleable for randomized circuits of size s.*

The proof of Lemma 5.9 is given in Section 5.4.1.

The next lemma implements the second step in our plan, and shows that an SS-non-malleable code where ψ is the function that outputs an empty string, is secure against $\psi = G$ if G is a sufficiently strong seed-extending PRG.

Lemma 5.10. *There exists a polynomial p such that the following holds: Let* $\mathrm{Enc} : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^n$ *and* $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{fail\}$ *be size t circuits, such that* (Enc, Dec) *form an (h, ρ)-non-malleable code for randomized circuits of size s. If $G : \{0,1\}^k \to \{0,1\}^n$ is a seed-extending ϵ-PRG for circuits of size $p(t, s, h, k)$, then* $(\mathrm{Enc}, \mathrm{Dec})$ *is a $(G, h, \rho + \epsilon)$-SS-non-malleable for randomized circuits of size s.*

The proof of Lemma 5.10 is given in Section 5.4.2.

We are finally ready to prove Theorem 5.5.

Proof. (of Theorem 5.5) Using the assumption that E is hard for exponential size nondeterministic circuits, we can apply Theorem 5.6 with the additional properties guaranteed in Proposition 5.7. We will choose $k = (\log n)^{c_1}$ for a universal constant c_1 that will be specified later.

Using the assumption that E is hard for exponential size nondeterministic circuits, we can also apply Theorem 3.4 to obtain an (h, n^{-c})-HTS $f : \{0,1\}^{a \cdot \log n} \to \{0,1\}^{(\log n)^{O(1)}}$ for $h = n^{a'}$, where we cannot choose a', but can choose a to be much larger than a'. We will also make sure to set Theorem 3.4 so that this HTS will be against circuits of size $\mathrm{poly}(n^e)$ (where n^e is the size of the simulating circuit guaranteed in Proposition 5.7) for a sufficiently large polynomial that will be chosen later. This is done, as we plan to apply Lemma 5.9 which requires that the HTS will be secure against the simulating circuit A (rather than the original circuit C). The parameters are set up so that we can apply Lemma 5.9 and obtain $\mathrm{Enc}_f : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^{k'}$ and $\mathrm{Dec}_f : \{0,1\}^{k'} \to \{0,1\}^k \cup \{fail\}$ for $k = a \log n$, and $k' = \mathrm{poly}(\log n)$ for some universal polynomial. (Here we needed to set up $\mathrm{Enc}, \mathrm{Dec}$ to encode strings of length $(\log n)^{c_1}$ because the input length of Enc is the sum of the input length of Enc_f and the output length of f. We were shooting to obtain input length $k = a \log n$, and can indeed choose the constant c_1 so that this can be obtained. A consequence is

that k' is a polynomial in k which indeed gives that $k' = (\log n)^{c_0}$ for some universal constant c_0. Applying Lemma 5.9 we indeed get that $(\mathrm{Enc}_f, \mathrm{Dec}_f)$ are $(n^{a'}, n^{-c})$-SS-non-malleable for randomized circuits of size n^c.

We finally apply Lemma 5.10 to argue that $(\mathrm{Enc}_f, \mathrm{Dec}_f)$ remain secure even if the adversary C, also sees $G(X)$. For this purpose, we note that $\mathrm{Enc}_f, \mathrm{Dec}_f$ can be computed in time $t(n)$ for some polynomial t. This is because computing $\mathrm{Enc}_f, \mathrm{Dec}_f$ reduces to computing $\mathrm{Enc}, \mathrm{Dec}$ and f (which all can be computed in time polynomial in n). Using Lemma 5.10, this gives that if G is a seed-extending PRG for circuits of size $p(t(n), n^c, h, k)$ (which is yet a larger polynomial in n) then $(\mathrm{Enc}_f, \mathrm{Dec}_f)$ form a $(G, n^{a'}, n^{-c})$-SS-non-malleable code for randomized circuits of size n^c. □

5.4.1 Proof of Lemma 5.9

We have that $(\mathrm{Enc}, \mathrm{Dec})$ is ϵ-non-malleable for randomized circuits of size s and is simulatable by circuits of size s'. Let C be a randomized circuit of size s. By the non-malleability of $(\mathrm{Enc}, \mathrm{Dec})$, C has a simulating circuit A of size s'. We will consider the following probability space:

- $X \leftarrow U_k$.
- $Y = f(X)$.
- $S \leftarrow U_d$.
- $(X_R, Y_R) = \mathrm{Dec}_f(C(\mathrm{Enc}_f(X \circ f(X), S)))$.
- $(X_M, Y_M) \leftarrow A$.
- $(X_I, Y_I) = \mathrm{Copy}((X_M, Y_M), (X, Y))$.

By the non-malleability of $(\mathrm{Enc}, \mathrm{Dec})$ we have that for every x

$$\Delta((X_R, Y_R)|X = x); ((X_I, Y_I|X = x)) \leq \epsilon.$$

Which implies that:

Claim 5.11. $\Delta((X, X_R, Y_R); (X, X_I, Y_I) \leq \epsilon$.

We use only this claim later on, and note that this property seems weaker than full-fledged non-malleability, as it only protects the encoding of the specific message distribution $m = (X, Y)$, rather than protecting the encoding of every individual message.

By the HTS property of f against the circuit A, we have that there exists a set $H \subseteq \{0,1\}^k$ of size at most h, such that:

$$\Pr[X_M \notin H \text{ and } Y_M = f(X_M)] \leq \rho.$$

In order to prove the SS-non-malleability of $(\mathrm{Enc}_f, \mathrm{Dec}_f)$ we plan to use the set H against C, and our goal is to prove that:

Claim 5.12. $\Pr[\mathrm{Dec}_f(C(\mathrm{Enc}_f(X, S))) \notin H \cup X \cup \{fail\}] \leq \epsilon + \rho$

Proof. We first observe that by the definition of $(\mathrm{Enc}_f, \mathrm{Dec}_f)$ we have that:

$$\{\mathrm{Dec}_f(C(\mathrm{Enc}_f(X, S))) \notin H \cup X \cup \{fail\}\} \Rightarrow \{X_R \notin H \text{ and } Y_R = f(X_R) \text{ and } X_R \neq X\}.$$

We therefore can conclude that:

$$\begin{aligned}\Pr[\mathrm{Dec}_f(C(\mathrm{Enc}_f(X, S))) \notin H \cup X \cup \{fail\}] &\leq \Pr[X_R \notin H \text{ and } Y_R = f(X_R) \text{ and } X_R \neq X] \\ &\leq \Pr[X_I \notin H \text{ and } Y_I = f(X_I) \text{ and } X_I \neq X] + \epsilon,\end{aligned}$$

where the second inequality follows by Claim 5.11. The pair (X_I, Y_I) were obtained by

$$(X_I, Y_I) = \text{Copy}((X_M, Y_M), (X, Y)).$$

This means that whenever $X_I \neq X$ then $(X_M, Y_M) = (X_I, Y_I)$ and we can conclude that:

$$\begin{aligned}\Pr[X_I \notin H \text{ and } Y_I = f(X_I) \text{ and } X_I \neq X] &\leq \Pr[X_M \notin H \text{ and } Y_M = f(X_M) \text{ and } X_M \neq X] \\ &\leq \Pr[X_M \notin H \text{ and } Y_M = f(X_M)] \\ &\leq \rho.\end{aligned}$$

Overall, we conclude that

$$\Pr[\text{Dec}_f(C(\text{Enc}_f(X, S))) \notin H \cup X \cup \{fail\}] \leq \epsilon + \rho,$$

as required. □

5.4.2 Proof of Lemma 5.10

Let $C : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a randomized circuit of size s, our goal is to show that there exists a set $H \subseteq \{0,1\}^k$ of size h such that

$$\Pr_{X \leftarrow \{0,1\}^k, S \leftarrow \{0,1\}^d} [\text{Dec}\left(C(G(X), \text{Enc}(X, S))\right) \notin H \cup \{X\} \cup \{fail\}] \leq \epsilon + \rho.$$

For this purpose, we will consider a randomized circuit $C' : \{0,1\}^n \to \{0,1\}^n$ defined by $C'(z) = C(U_n, z)$. Intuitively, the randomized circuit C' replaces the input $G(X)$ with U_n. By the SS-non-malleability of the (Enc, Dec) we have that there exists a set H of size at most h (for C') which satisfies:

$$\Pr_{X \leftarrow \{0,1\}^k, S \leftarrow \{0,1\}^d} \left[\text{Dec}\left(C'(\text{Enc}(X, S))\right) \notin H \cup \{X\} \cup \{fail\}\right] \leq \rho.$$

We plan to use this H for C, and use the pseudorandomness of G to argue that C (that sees $G(X)$) cannot benefit from this additional input.

For this purpose, we will consider the following randomized circuit B which receives input $(x, z) \in \{0,1\}^k \times \{0,1\}^n$.

- B simulates the encoding/decoding experiment using C and z as follows: Choose $S \leftarrow U_d$, and computes $\bar{x} = \text{Dec}(C(z, \text{Enc}(x, S))$.
- If $\bar{x} \in H \cup \{x\} \cup \{fail\}$ answer one, and otherwise answer zero.

This definition is made so that the following holds:

Claim 5.13.

- *For $X \leftarrow U_k$ and $Z = G(X)$, $\{B(X, Z) = 1\} \Leftrightarrow \{\text{Dec}\left(C(G(X), \text{Enc}(X, S))\right) \notin H \cup \{X\} \cup \{fail\}\}$.*
- *For $X \leftarrow U_k$ and $Z \leftarrow U_n$, $\{B(X, Z) = 1\} \Leftrightarrow \{\text{Dec}\left(C'(\text{Enc}(X, S))\right) \notin H \cup \{X\} \cup \{fail\}\}$.*
- *B can be computed by a randomized circuit of size $p(t, s, h, k)$ for some universal polynomial p.*

G is a seed-extending ϵ-PRG for circuits of size $p(t,s,h,k)$, meaning that $(X, G(X))$ is indistinguishable from (X, U_n) for circuits of that size. We conclude that for

$$p = \Pr_{X \leftarrow \{0,1\}^k, S \leftarrow \{0,1\}^d} \left[\mathrm{Dec}\left(C(G(X), \mathrm{Enc}(X,S))\right) \notin H \cup \{X\} \cup \{fail\} \right],$$

we have that:

$$\begin{aligned} p &= \Pr_{X \leftarrow U_k} [B(X, G(X)) = 1] \\ &\leq \Pr_{X \leftarrow U_k, Z \leftarrow U_n} [B(X,Z) = 1] + \epsilon \\ &= \Pr_{X \leftarrow \{0,1\}^k, S \leftarrow \{0,1\}^d} \left[\mathrm{Dec}\left(C'(\mathrm{Enc}(X,S))\right) \notin H \cup \{X\} \cup \{fail\} \right] + \epsilon \\ &\leq \rho + \epsilon. \end{aligned}$$

6 A construction of stochastic codes for poly-size circuits based on hardness assumptions

In this section we prove our main result, providing an explicit construction of stochastic codes for poly-size circuits based on hardness assumption (proving Theorem 1.1). We start by restating Theorem 1.1 in a more precise way:

Theorem 6.1 (Explicit stochastic codes for poly-size channels)**.** *If E is hard for exponential size nondeterministic circuits then for every constants $0 \leq p < \frac{1}{4}$, $c > 1$, and for every sufficiently small constant $\epsilon > 0$, there exists a constant d such that for infinitely many N, there is a stochastic code $(\mathrm{Enc}, \mathrm{Dec})$ for $\mathrm{Ckt}_p^{N^c}$ with:*

- *Rate $R \geq 1 - H(p) - \epsilon$.*
- *Success probability $1 - \frac{1}{N^c}$.*

Furthermore, the construction is explicit and $\mathrm{Enc}, \mathrm{Dec}$ are computable in time N^d.

Remark 6.2. *Theorem 6.1 is stated for "infinitely many n", rather than for "every sufficiently large n". However, it works for a very dense set of n, and the only component in the construction that is stated for infinitely many n is Theorem 2.33 which works for a dense set of n. See Remarks 4.3, and Remark 2.34.*

In the remainder of the section we prove Theorem 6.1. In Section 6.1 we present our construction. In Section 6.2.1 we compare our construction to those used in previous work. The proof is given in Section 6.2.

6.1 The construction

In this section we present our construction of stochastic codes for channels that are implementable by poly-size circuits.

The construction and analysis in this chapter build heavily on previous work starting with the seminal paper of Guruswami and Smith [GS16], and using ideas from later works [GS16, SS21a, SS21b, SS22]. We would like to imitate the construction of Shaltiel and Silbak [SS22] (that relied on Monte-Carlo constructions of evasive BSC codes, and SS-non-malleable codes, now that we have given explicit constructions of these objects. Unfortunately, we cannot do this directly because the objects that we explicitly construct are not as strong as the ones that were guaranteed in the Monte-Carlo constructions of [SS22]. More specifically, the

explicit SS-non-malleable codes that we construct do not have some of the properties that the Monte-Carlo ones from [SS22] have. (See Remark 5.4 for a detailed comparison).

This forces us to modify the construction in several respects, in order to get by with this weaker component. The changes in constructions are natural (and some of them build on earlier ideas) and yet, they lead to some subtle changes which make the analysis somewhat more cumbersome. Nevertheless, we stress that on an "intuitive high-level" both construction and analysis are essentially similar to that of [SS22]. We will not give an intuitive explanation of the overall construction and analysis, as such an explanation appears in [SS22]. Instead, we will try to explain some of the differences in Section 6.2.1.

The construction is detailed in four figures: Figure 1 lists parameters, Figure 2 lists ingredients that we use, Figure 3 describes the encoding algorithm, and Figure 4 describes the decoding algorithm. We start with some notation and definitions.

Partitioning codewords into control blocks and data blocks. The construction will think of codewords $c \in \{0,1\}^N$ as being composed of $n = n_{\text{ctrl}} + n_{\text{data}}$ blocks of length $b = N/n$. We specify the precise choices of $n, b, n_{\text{ctrl}}, n_{\text{data}}$ in Figure 1.

Figure 1: Parameters for stochastic code

In this figure we make some of the parameter choices for the construction of Theorem 6.1.

Hardness assumption: We are assuming that E is hard for exponential size nondeterministic circuits.

We are given constants:

- $0 < p < \frac{1}{4}$ - The fraction of errors we need to recover from.
- $\epsilon > 0$ - We will construct a stochastic code Enc with output length N, and rate $R \geq R'(p) - \epsilon$. We assume that $\epsilon > 0$, is sufficiently small in terms of p.
- c - We are aiming to construct a stochastic code for circuits of size N^c with success $1 - \nu$ for $\nu = \frac{1}{N^c}$.

Other parameters that we choose:

- N - The length (in bits) of the codeword. Throughout, we assume that N is sufficiently large, and that other parameters are either constants, or chosen as a function of N.
- Let $b = c_b \cdot \log N$, where c_b is a constant that we choose later in Figure 2.
- Let $n = N/b$. We split the N output bits of the codeword to n blocks of length b.
- Let $n_{\text{ctrl}} = n^{0.9}$ be the number of "control blocks", and $n_{\text{data}} = n - n_{\text{ctrl}}$ be the number of "data blocks".
- Let $N_{\text{ctrl}} = b \cdot n_{\text{ctrl}}$ and $N_{\text{data}} = b \cdot n_{\text{data}}$. (Note that: $n = n_{\text{ctrl}} + n_{\text{data}}$, $N = N_{\text{ctrl}} + N_{\text{data}}$).
- Let $c_0 > 1$ be a sufficiently large universal constant that we will choose in the proof of Theorem 6.1.

We use these choices to choose ingredients (a sampler, a PRG, a pseudorandomly chosen permutation, a pseudorandom code, an SS-non-malleable code, an instantiation of the Reed-Solomon code and an evasive BSC code) that will be used in the construction. These choices are made in Figure 2.

We now set up some notation. Given a subset $I \subseteq [n]$ of n_{ctrl} distinct indices, we can decompose c into its data part $c_{\text{data}} \in \{0,1\}^{N_{\text{data}} = n_{\text{data}} \cdot b}$ and its control part $c_{\text{ctrl}} \in \{0,1\}^{N_{\text{ctrl}} = n_{\text{ctrl}} \cdot b}$. Similarly, given strings c_{data} and c_{ctrl} we can prepare the codeword c (which we denote by $(c_{\text{data}}, c_{\text{ctrl}})^I$ by the reverse operation. This is stated formally in the definition below.

Definition 6.3 (Data and control portion of a codeword)**.** *We view strings $c \in \{0,1\}^N$ as composed of n blocks of length $b = N/n$, so that $c \in (\{0,1\}^b)^n$, and c_i denotes the b bit long i'th block of c.*

Let $I = \{i_1, \ldots, i_{n_{\text{ctrl}}}\} \subseteq [n]$ be a subset of indices of size n_{ctrl}.

Figure 2: Ingredients for stochastic code

We are assuming that E is hard for exponential size nondeterministic circuits. In Figure 1 we specified parameters that are used by the construction. More specifically, we when given constants p, ϵ, c, and a specified codeword length N, we have chosen parameters $n, b, n_{\text{ctrl}}, n_{\text{data}}, N_{\text{ctrl}}, N_{\text{data}}$ that were chosen as a function of previous choices, and of a constant $c_b \geq 1$, that was not yet specified. In this figure, we will specify the ingredients that will be used in our construction, and choose c_b. The constant c_0 will be chosen in the proof to be sufficiently large.

Averaging Sampler: We use Theorem 2.19 (choosing $c_{\text{samp}} = c + c_0$) to obtain a $(\frac{1}{\log n}, \frac{1}{n^{c_{\text{samp}}}})$-sampler with distinct samples $\text{Samp} : \{0,1\}^{d_{\text{samp}}} \to [n]^{n_{\text{ctrl}}}$. We indeed meet the condition that the number of samples $n_{\text{ctrl}} = n^{0.9} \geq 2^{0.1 \log n}$. By Theorem 2.19 we have an explicit construction with seed length $d_{\text{samp}} = O_{c,c_0}(\log n) = O_{c,c_0}(\log N)$.

RS code: Let $\text{Enc}_{\text{RS}} : \{0,1\}^{N^{0.1}} \to (\{0,1\}^{\log N})^{n_{\text{ctrl}}}$ be the Reed-Solomon code from Theorem 2.13

Pseudorandom code: Let $c_{\text{pr}} = c + c_0$. Using the hardness assumption, we apply Theorem 2.12 (choosing $p = 1/3$, $c = c_{\text{pr}}$ and $a = 2$) to obtain constants L_{pr}, c_d and the constant c_b (which we have promised to choose earlier). We also obtain a stochastic code $(\text{Enc}_{\text{pr}}, \text{Dec}_{\text{pr}})$ where $\text{Enc}_{\text{pr}} : \{0,1\}^{2 \cdot \log N} \times \{0,1\}^{d = c_d \cdot \log N} \to \{0,1\}^{b = c_b \cdot \log N}$ and $\text{Dec}_{\text{pr}} : \{0,1\}^{b = c_b \cdot \log N} \to \{0,1\}^{2 \cdot \log N}$. We are guaranteed that the code is L_{pr}-wealy-list decodable from radius $1/3$, and $\frac{1}{N^{c_{\text{pr}}}}$-pseudorandom for size $N^{c_{\text{pr}}}$ circuits. Furthermore, there exists a constant $d_{\text{pr}} > c_{\text{pr}}$ such that (Enc, Dec) run in time $N^{d_{\text{pr}}}$, and have circuits of size $N^{d_{\text{pr}}}$.

SS-non-malleable code: Let $c_{\text{ssnm}} = d_{\text{pr}} + c_0$. Using the hardness assumption, we apply Theorem 5.5 (choosing $c = c_{\text{ssnm}}$) to obtain constants $d_{\text{ssnm}} > c_{\text{ssnm}}, c'_{\text{ssnm}} > d_{\text{ssnm}}$ and a constant $a'_{\text{ssnm}} < c'_{\text{ssnm}}$, as well as a stochastic code $(\text{Enc}_{\text{ssnm}}, \text{Dec}_{\text{ssnm}})$. We are allowed to choose any constant $c_k \geq a'_{\text{ssnm}}$, and setting $\ell = c_k \cdot \log N$, we have that $\text{Enc}_{\text{ssnm}} : \{0,1\}^{\ell} \times \{0,1\}^{N^{d_{\text{ssnm}}}} \to \{0,1\}^{(\log N)^{O(1)}}$ (where the constant hidden in the $O(1)$ is universal). We are also guaranteed that for every $\hat{G} : \{0,1\}^{\ell} \to \{0,1\}^{N}$ that is a seed-extending $\frac{1}{N^{c'_{\text{ssnm}}}}$-PRG for circuits of size $N^{c'_{\text{ssnm}}}$, the stochastic code $(\text{Enc}_{\text{ssnm}}, \text{Dec}_{\text{ssnm}})$ is $(\hat{G}, N^{a'_{\text{ssnm}}}, \frac{1}{N^{c_{\text{ssnm}}}})$-SS-non-malleable for randomized circuits of size $N^{c_{\text{ssnm}}}$. Furthermore $\text{Enc}_{\text{ssnm}}, \text{Dec}_{\text{ssnm}}$ run in time $N^{d_{\text{ssnm}}}$, and have circuits of size $N^{d_{\text{ssnm}}}$.

BSC code that is evasive: Let $c_{\text{BSC}} = d_{\text{ssnm}} + c_0$, $p_{\text{BSC}} = p \cdot (1 + \frac{\epsilon}{10})$, and $R_{\text{BSC}} = 1 - H(p) - \epsilon/3$. Using the hardness assumption, we apply Theorem 4.2 (using $c = c_{\text{BSC}}$, $p = p_{\text{BSC}}, \epsilon = \epsilon/3$). We obtain a constant d_{BSC} and a code $(\text{Enc}_{\text{BSC}}, \text{Dec}_{\text{BSC}})$ where $\text{Enc}_{\text{BSC}} : \{0,1\}^{R_{\text{BSC}} \cdot N_{\text{data}}} \to \{0,1\}^{N_{\text{data}}}$ and $\text{Dec}_{\text{BSC}} : \{0,1\}^{N_{\text{data}}} \to \{0,1\}^{R_{\text{BSC}} N_{\text{data}}} \cup \{fail\}$ that run in time $N^{d_{\text{BSC}}}$, and have circuits of size $N^{d_{\text{BSC}}}$. (Theorem 4.2 only gives a guarantee for infinitely many choices of N_{data}, and using the fact that N is uniquely derived from N_{data}, this translates to the guarantee made in Theorem 6.1 that the theorem holds for infinitely many N. See Remark 6.2 for a discussion). We are also guaranteed that the code is $\frac{1}{N^{c_{\text{BSC}}}}$-evasive for $\text{Ckt}^{N^{c_{\text{BSC}}}}_{p_{\text{BSC}}}$, and decodes from $\text{Perm}^{\text{UniPerm}}_{p_{\text{BSC}}}$.

PRG against circuits: Let c_{PRG} be a constant that is larger than $c_0 + c + d_{\text{ssnm}} + c'_{\text{ssnm}} + a'_{\text{ssnm}} + d_{\text{BSC}} + d_{\text{pr}}$. Using the hardness assumption, we apply Theorem 2.7 to obtain that there exists a constant $c_k > 1$ (and this is where we choose the constant c_k specified earlier) and a seed extending $\frac{1}{3 \cdot N^{c_{\text{PRG}}}}$-PRG, $G : \{0,1\}^{d_{\text{PRG}} = \frac{c_k}{3} \cdot \log N} \to \{0,1\}^{N^2}$ for circuits of size $N^{c_{\text{PRG}}}$. Furthermore, G runs in time $\text{poly}(N^{c_{\text{PRG}}})$. We will sometimes view G as a function that outputs only N_{data} bits by truncating the output to length N_{data}.

Pseudorandomly chosen permutation: Let $\pi^G : \{0,1\}^{d_{\text{PRG}}} \times [N_{\text{data}}] \to [N_{\text{data}}]$ be the function defined in Definition 2.31, so that for a string $s_\pi \in \{0,1\}^{d_{\text{PRG}}}$, $\pi_{s_\pi} = \pi^G_{s_\pi}$ is a permutation over $[N_{\text{data}}]$.

Control strings, and a combined PRG: As $d_{\text{samp}} = O_{c,c_0}(\log N)$ we have that $d_{\text{PRG}} = \frac{c_k}{3} \cdot \log N$ is at least d_{samp}. We will assume w.l.o.g. that $d_{\text{samp}} = d_{\text{PRG}}$. As we have already chosen $\ell = c_k \cdot \log N$, this means that $\ell' = \frac{\ell}{3} = \frac{c_k}{3} \cdot \log N$ is the length of seeds for the sampler, PRG, and pseudorandomly chosen permutation. Given $s \in \{0,1\}^{\ell}$ (which we call a "control string") we think of it as consisting of three such seeds $s = (s_{\text{samp}}, s_\pi, s_{\text{PRG}})$ where each is of length ℓ'. We also define a "combined PRG" $\hat{G} : \{0,1\}^{\ell} \to \{0,1\}^{3N^2}$ by $\hat{G}(s) = G(s_{\text{samp}}), G(s_\pi), G(s_{\text{PRG}})$, and it follows by a hybrid argument that $\hat{G}$ is a seed-extending $\frac{1}{N^{c'_{\text{ssnm}}}}$-PRG for circuits of size $N^{c'_{\text{ssnm}}}$ (meaning that $\hat{G}$ meets the requirement made by the SS-non-malleable code).

- *Given strings* $c_{\text{data}} \in \{0,1\}^{N_{\text{data}}}$ *and* $c_{\text{ctrl}} \in \{0,1\}^{N_{\text{ctrl}}}$ *we define an* N *bit string* c *denoted by* $(c_{\text{data}}, c_{\text{ctrl}})^I$ *as follows: We think of* $c_{\text{data}}, c_{\text{ctrl}}, c$ *as being composed of blocks of length* b *(that is* $c_{\text{data}} \in (\{0,1\}^b)^{n_{\text{data}}}$, $c_{\text{ctrl}} \in (\{0,1\}^b)^{n_{\text{ctrl}}}$ *and* $c \in (\{0,1\}^b)^n$*). We enumerate the indices in* $[n] \setminus I$ *by* $j_1, \ldots, j_{n_{\text{data}}}$ *and set* $c_\ell = \begin{cases} (c_{\text{ctrl}})_k & \text{if } \ell = i_k \text{ for some } k; \\ (c_{\text{data}})_k & \text{if } \ell = j_k \text{ for some } k \end{cases}$
- *Given a string* $c \in \{0,1\}^N$ *(which we think of as* $c \in (\{0,1\}^b)^n$*) we define strings* $c^I_{\text{data}}, c^I_{\text{ctrl}}$ *by* $c^I_{\text{ctrl}} = c|_I$ *and* $c^I_{\text{data}} = c|_{[n]\setminus I}$*, (namely the strings restricted to the indices in* I*,* $[n] \setminus I$*, respectively).*

We omit the superscript I *when it is clear from the context.*

6.2 Proof of Theorem 6.1

This section is devoted to proving Theorem 6.1, and show the correctness of the main construction.

The setup: Throughout the remainder of the section, we fix the setup of Theorem 6.1. Specifically, let $0 \leq p < \frac{1}{4}$, $c \geq 1$ be constants, and let $\epsilon > 0$ be a sufficiently small constant. We use these choices to set up the parameters and ingredients as explained in Figure 1 and Figure 2. The choices in Figure 1 allow us to choose a universal constant $c_0 > 1$ that we will choose later. Let N be sufficiently large, we will argue the correctness for infinitely many N. More specifically, as chosen in Figure 2, we will assume that $N_{\text{data}} = N - N_{\text{ctrl}}$ is one of the lengths for which the code $(\text{Enc}_{\text{BSC}}, \text{Dec}_{\text{BSC}})$ was guaranteed to be correct in Figure 2, and there are indeed infinitely many choices of such N. We set $\nu = \frac{1}{N^c}$ to be the required error parameter.

Let $\text{Enc} : \{0,1\}^{RN} \times \{0,1\}^{\ell + n_{\text{ctrl}} \cdot d + N^{d_{\text{ssnm}}}} \to \{0,1\}^N$ and $\text{Dec} : \{0,1\}^N \to \{0,1\}^{RN} \cup \{fail\}$ be the functions specified in Figures 3 and Figure 4 using the ingredients and parameter choices in Figure 1 and Figure 2. Overall, by construction, the algorithms (Enc, Dec) run in time polynomial in n.

Bounding The Rate. By Figure 3, the rate R of Enc is given by:

$$\begin{aligned} R &= \frac{R_{\text{BSC}} \cdot N_{\text{data}}}{N} \\ &= (1 - H(p_{\text{BSC}}) - \frac{\epsilon}{3}) \cdot (1 - \frac{n_{\text{ctrl}}}{n}) \\ &\geq (1 - H(p) - \frac{\epsilon}{10} - \frac{\epsilon}{3}) \cdot (1 - \frac{\epsilon}{10}) \\ &\geq 1 - H(p) - \epsilon, \end{aligned}$$

where the third line follows because $n_{\text{ctrl}} = n^{0.9}$, and using Equation (2) from Section 2, we have that:

$$H(p_{\text{BSC}}) = H(p \cdot (1 + \frac{\epsilon}{10})) \leq H(p) + \frac{\epsilon}{10}.$$

Road map for arguing the correctness of decoding. The main part in proving Theorem 6.1 is showing that the decoding algorithm is correct. The remainder of this section is devoted to this proof, and in this section we give a roadmap of this proof.

The setup:

- Let $m \in \{0,1\}^{RN}$ be a message.
- Let $C : \{0,1\}^N \to \{0,1\}^N$ be a channel in $\text{Ckt}_p^{N^c}$

Figure 3: Encoding algorithm for stochastic code

Preperations: In this figure we use the parameters and ingredients of Figures 1 and Figure 2 to define the stochastic encoding map. The encoding map will encode messages of length $R_{\mathrm{BSC}} \cdot N_{\mathrm{data}}$, and will output words of length N. This gives that it will have rate $R = \frac{R_{\mathrm{BSC}} \cdot N_{\mathrm{data}}}{N}$. The stochastic encoding map and will use a seed of length $\ell + n_{\mathrm{ctrl}} \cdot d + N^{d_{\mathrm{ssnm}}}$.

Definition: We define $\mathrm{Enc} : \{0,1\}^{R \cdot N} \times \{0,1\}^{\ell + n_{\mathrm{ctrl}} \cdot d + N^{d_{\mathrm{ssnm}}}} \to \{0,1\}^N$ as follows:

Input:

- A message $m \in \{0,1\}^{R_{\mathrm{BSC}} \cdot N_{\mathrm{data}}}$.
- A "random coin" for the stochastic encoding that consists of:
  - $s = (s_{\mathrm{samp}}, s_\pi, s_{\mathrm{PRG}})$. Recall that $s_{\mathrm{samp}}, s_\pi, s_{\mathrm{PRG}} \in \{0,1\}^{\ell'}$, and $\ell' = \ell/3$, so that $s \in \{0,1\}^{\ell}$.
  - $r_0 \in \{0,1\}^{N^{d_{\mathrm{ssnm}}}}$.
  - $r_1, \ldots, r_{n_{\mathrm{ctrl}}} \in \{0,1\}^d$.

Output: A codeword $c = \mathrm{Enc}(m; (s, r_0, r_1, \ldots, r_{n_{\mathrm{ctrl}}}))$ of length N.

Operation:

Determine control blocks: Apply $\mathrm{Samp}(G(s_{\mathrm{samp}}))$ to generate *control* $= \{i_1, \ldots, i_{n_{\mathrm{ctrl}}}\} \subseteq [n]$. These blocks will be called "control blocks", and the remaining n_{data} blocks will be called "data blocks".

Prepare data part: We prepare a string c_{data} of length N_{data} as follows:

- Encode m by $x = \mathrm{Enc}_{\mathrm{BSC}}(m)$.
- Generate an N_{data} bit string y by reordering the N_{data} bits of the encoding using the (inverse of) the permutation $\pi_{s_\pi}(\cdot) = \pi^G(s_\pi, \cdot)$. More precisely, $y = \pi_{s_\pi}^{-1}(x) = \pi_{s_\pi}^{-1}(\mathrm{Enc}_{\mathrm{BSC}}(m))$.
- Mask y using PRG. That is, $c_{\mathrm{data}} = y \oplus G(s_{\mathrm{PRG}}) = \pi_{s_\pi}^{-1}(\mathrm{Enc}_{\mathrm{BSC}}(m)) \oplus G(s_{\mathrm{PRG}})$. (Here we truncate the output of G to length N_{data}).

Prepare control part: We prepare a string c_{ctrl} of length N_{ctrl} (which we view as n_{ctrl} blocks of length b) as follows:

- Compute $o = \mathrm{Enc}_{\mathrm{ssnm}}(s, r_0)$.
- Compute $w = \mathrm{Enc}_{\mathrm{RS}}(o)$. Note that $\mathrm{Enc}_{\mathrm{RS}}$ was set to encode inputs of length $N^{0.1}$, and so, can be used on o, which is an output of $\mathrm{Enc}_{\mathrm{ssnm}}$ (which is of length $\log^{O(1)} N$). We obtain $w \in (\{0,1\}^{\log N})^{n_{\mathrm{ctrl}}}$.
- For every $j \in [n_{\mathrm{ctrl}}]$, compute $(c_{\mathrm{ctrl}})_j = \mathrm{Enc}_{\mathrm{pr}}((i_j, w_j), r_j)$. (Here, we encode the pair (i_j, w_j) which we can view as a string of length $\log n + \log N \leq 2 \log N$ (and so we can apply $\mathrm{Enc}_{\mathrm{pr}}$ on such a string).

Merge data and control parts: We prepare the final output codeword $c \in \{0,1\}^N$ by merging c_{data} and c_{ctrl}. That is, $c = (c_{\mathrm{data}}, c_{\mathrm{ctrl}})^{\textit{control}}$.

We will keep these choices of m, C fixed throughout this section.

We need to show that w.h.p. the message m is decoded correctly when applying encoding, channel and decoding. We will refer to this experiment as the encoding/decoding experiment, and will denote it by $\mathrm{expr}^{\mathrm{ed}}(m, C)$. In this experiment, $S \in \{0,1\}^\ell$, $R_0 \in \{0,1\}^{N^{d_{\mathrm{ssnm}}}}$ and $R_1, \ldots, R_{n_{\mathrm{ctrl}}} \in \{0,1\}^d$ are chosen uniformly at random. We set $R = (R_0, R_1, \ldots, R_{n_{\mathrm{ctrl}}})$. The string $Z = \mathrm{Enc}(m; (S, R))$ is the codeword, $E = C(Z)$ is the error pattern chosen by the channel, $\bar{V} = Z \oplus E$ is the received word given to the decoding, and $\bar{M} = \mathrm{Dec}(\bar{V})$ is the message returned by the decoding. We use the convention that capital letters denote the random variables associated with small letters used in the construction, and a complete specification of

Figure 4: Decoding algorithm for stochastic code

Definition: We define $\mathrm{Dec} : \{0,1\}^N \to \{0,1\}^{R \cdot N} \cup \{fail\}$ as follows:

Input: A "received word" $\bar{v} \in \{0,1\}^N$.

Output: A message $\bar{m} \in \{0,1\}^{R \cdot N}$ or $fail$.

Operation of decoding algorithm: On input $\bar{v} \in \{0,1\}^N$:

Compute control candidates:

- For $i \in [n]$, let $\mathrm{List}_i = \mathrm{Dec}_{\mathrm{pr}}(\bar{v}_i)$. (Here $\bar{v}_i$ is the i'th block of $\bar{v}$). Recall that for every i, List_i is of size L_{pr}.
- Let List denote the union over all the elements in the n lists.
- Every element in List can be thought of as a pair $(i, w) \in \{0,1\}^{\log n} \times \{0,1\}^{\log N}$ (in a similar way to what was done in the encoding algorithm).
- In List there are at most $t = L_{\mathrm{pr}} \cdot n$ such pairs. We now run the list-decoding algorithm of Theorem 2.13 assuming that the "agreement parameter" $a = \frac{n_{\mathrm{ctrl}}}{100}$. We obtain a list of at most $2t/a = 200 \cdot L_{\mathrm{pr}} \cdot \frac{N}{n_{\mathrm{ctrl}}} \leq 200 \cdot L_{\mathrm{pr}} \cdot N^{0.2} \leq n$ strings $\bar{o}_1, \ldots, \bar{o}_n$ where each is an element of $\{0,1\}^{N^{0.1}}$, and by only looking at a prefix, we view it as a string of length that is the output length of $\mathrm{Enc}_{\mathrm{ssnm}}$.
- For every $i \in [n]$, compute $\bar{s}_i = \mathrm{Dec}_{\mathrm{ssnm}}(\bar{o}_i)$. We define *candidates* $= \{\bar{s}_i : i \in [n]\}$.

Compute valid candidates: We say that $\bar{s}$ is ***successful***, if when computing the procedure *DecodeUsingCandidate*$(\bar{s})$ (that is defined below) we obtain $\bar{m}(\bar{s}) \neq fail$.

Output message: If there exists a single $s^* \in$ *candidates* that is successful, output $\bar{m}(s^*)$. Otherwise, output $fail$.

Internal procedure *DecodeUsingCandidate*: On input $\bar{s} \in \{0,1\}^{\ell}$ (which we think of as a candidate for the control string) this procedure works as follows:

Determine control blocks: Apply $\mathrm{Samp}(G(\bar{s}_{\mathrm{samp}}))$ to generate $\overline{control} = \{\bar{i}_1, \ldots, \bar{i}_{n_{\mathrm{ctrl}}}\}$. Compute $\bar{v}_{\mathrm{data}} = \bar{v}^{\overline{control}}_{\mathrm{data}}$.

Unmask PRG: Compute $\bar{y} = \bar{v}_{\mathrm{data}} \oplus G(\bar{s}_{\mathrm{PRG}})$. (Here we truncate the output of G to length N_{data}).

Reverse permutation: Let $\bar{x}$ be the N_{data} bit string obtained by "undoing" the permutation. More precisely, let $\pi_{\bar{s}_\pi}(\cdot) = \pi^G(\bar{s}_\pi, \cdot)$, and let $\bar{x} = \pi_{\bar{s}_\pi}(\bar{y}) = \pi_{\bar{s}_\pi}(\bar{v}_{\mathrm{data}} \oplus G(\bar{s}_{\mathrm{PRG}}))$.

Decode data: Compute $\bar{m} = \mathrm{Dec}_{\mathrm{BSC}}(\bar{x})$.

output: We use $\bar{m}(\bar{s})$ to denote the answer $\bar{m}$ when *DecodeUsingCandidate* is applied on $\bar{s}$.

experiment $\mathrm{expr}^{\mathrm{ed}}(m, C)$ is given in Figure 5.

In order to complete the proof of Theorem 6.1 we need to show that the probability that the decoded message $\bar{M}$ is equal to m is large. More precisely, that:

$$\Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[\bar{M} = m] \geq 1 - \nu. \tag{3}$$

Recall that in the experiment, every candidate control string $\bar{s} \in$ *CANDIDATES* is used to produce a candidate message $\bar{M}(\bar{s})$.

The correct control string is one of the candidates. We first claim that w.h.p. the correct control string S is in *CANDIDATES* and that when decoding using this candidate we obtain the correct message m. (Loosely speaking, the earlier work of [GS16, SS21a] that obtained list-decoding, stopped here, and outputted the list of messages $\left\{\bar{M}(\bar{s}) : \bar{s} \in \textit{CANDIDATES}\right\}$). The next lemma is stating that this list indeed contains the original

Figure 5: The encoding/decoding experiment $\mathrm{expr}^{\mathrm{ed}}(m, C)$.

Parameters: A message $m \in \{0,1\}^{RN}$ and a channel $C \in \mathrm{Ckt}_p^{n^c}$.

Encoding phase: Choose uniformly at random $S \in \{0,1\}^{\ell}$, $R_0 \in \{0,1\}^{N^{d_{ssnm}}}$, and $R_1, \ldots, R_{n_{\mathrm{ctrl}}} \in \{0,1\}^d$, and let $Z = \mathrm{Enc}(m; (S, R))$. More specifically, divide S into three parts of length $\ell' = \ell/3$, so that $S = (S_{\mathrm{samp}}, S_{\mathrm{PRG}}, S_\pi)$ and perform the following:

- $\mathit{CONTROL} = \{I_1 < \ldots < I_{n_{\mathrm{ctrl}}}\} = \mathrm{Samp}(G(S_{\mathrm{samp}}))$.
- $x = \mathrm{Enc}_{\mathrm{BSC}}(m)$
- $Y = \pi_{S_\pi}^{-1}(x)$.
- $Z \in \{0,1\}^N$ is defined as follows:
  - $Z_{\mathrm{data}}^{\mathit{CONTROL}} = Y \oplus G(S_{\mathrm{PRG}})$.
  - $Z_{\mathrm{ctrl}}^{\mathit{CONTROL}}$ is defined as follows:
    - $O = \mathrm{Enc}_{\mathrm{ssnm}}(S, R_0)$.
    - $W = \mathrm{Enc}_{\mathrm{RS}}(O)$.
    - For every $j \in [n_{\mathrm{ctrl}}]$, $Z_{I_j} = \mathrm{Enc}_{\mathrm{pr}}((I_j, W_j), R_j)$.

Channel phase: Let $E = C(Z)$ and $\bar{V} = Z \oplus E$.

Decoding phase: Let $\bar{M} = \mathrm{Dec}(\bar{V})$. More specifically:

Compute candidates:

- For $i \in [n]$, let $\mathrm{LIST}_i = \mathrm{Dec}_{\mathrm{pr}}(\bar{V}_i)$.
- Let LIST denote the union over all the elements in the n lists.
- Run the list-decoding algorithm of Theorem 2.13 on the t pairs in LIST (as in the decoding algorithm, that is assuming that the "agreement parameter" $a = \frac{n_{\mathrm{ctrl}}}{100}$) to obtain a list $\bar{O}_1, \ldots, \bar{O}_n$.
- For every $i \in [n]$, let $\bar{S}_i = \mathrm{Dec}_{\mathrm{ssnm}}(\bar{O}_i)$.
- Let $\mathit{CANDIDATES} = \{\bar{S}_i : i \in [n]\}$.

Decode using candidates: For every $\bar{s} \in \mathit{CANDIDATES}$, compute $\mathit{DecodeUsingCandidate}(\bar{s})$, more specifically:

- Let $\overline{\mathit{CONTROL}}(\bar{s}) = \mathrm{Samp}(G(\bar{s}_{\mathrm{samp}}))$ and compute $\bar{V}_{\mathrm{data}}(\bar{s}) = \bar{V}(\bar{s})_{\mathrm{data}}^{\overline{\mathit{CONTROL}}}$.
- Let $\bar{Y}(\bar{s}) = \bar{V}_{\mathrm{data}}(\bar{s}) \oplus G(\bar{s}_{\mathrm{PRG}})$.
- Let $\bar{X}(\bar{s}) = \pi_{\bar{s}_\pi}(\bar{Y}(\bar{s}))$.
- Let $\bar{M}(\bar{s}) = \mathrm{Dec}_{\mathrm{BSC}}(\bar{X}(\bar{s}))$.

Compute valid candidates: For every $\bar{s} \in \mathit{CANDIDATES}$, determine whether $\bar{s}$ is successful, that is, if $\bar{M}(\bar{s}) \neq fail$.

Output message: If there exists a single $\bar{s} \in \mathit{CANDIDATES}$ that is successful, we denote it by S^* and the final output is $\bar{M} = \bar{M}(S^*)$, otherwise we set $S^* = fail$ and $\bar{M} = fail$.

message m.

Lemma 6.4 (The correct control string is one of the candidates)**.**

$$\Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[S \in \mathit{CANDIDATES} \text{ and } \bar{M}(S) = m] \geq 1 - \nu/2.$$

Loosely speaking, this is supposed to follow by the correctness of the list-decoding algorithm of [GS16] (with modifications in [SS21a]) which guarantees that the correct candidate control string appears in the list

CANDIDATES, and that when decoding with this candidate, the original message m is obtained. We explain the technique of previous work [GS16, SS21a, KSS19, SS21b] and prove Lemma 6.4 in Section 6.2.4. The argument is very similar to that used in the aforementioned previous work, but one has to be more careful, because of the reasons explained in Section 6.2.1.

All incorrect candidates are unsuccessful. This part of the analysis was the main contribution of [SS21b, SS22], showing that one can "get rid" of incorrect candidates, and achieve *unique decoding*. That is, we will show that w.h.p. only the candidate S is successful. Meaning that our decoding algorithm has that w.h.p. $S^* = S$ (and we identify the correct candidate). This is formally stated in the next lemma.

Lemma 6.5 (Only the correct candidate survives)**.**

$$\Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[S^* = S] \geq 1 - \nu/2.$$

Together, Lemmata 6.4 and 6.5 imply that with probability at least $1 - \nu$, we have that $S^* = S$ and $\bar{M} = \bar{M}(S^*) = \bar{M}(S) = m$. This means that (3) holds, and the correct message is decoded with probability $1 - \nu$, concluding the proof of Theorem 6.1.

The argument for proving Lemma 6.5. For this part, we will use the same overall approach of [SS22]. We will first use the SS-non-malleability of $\mathrm{Enc}_{\mathrm{ssnm}}$ to prove the following lemma:

Lemma 6.6 (Using SS-non-malleability)**.** *There exists a set $H \subseteq \{0,1\}^{c_k \cdot \log N}$ of size at most $N^{a'_{\mathrm{ssnm}}+1}$ such that:*

$$\Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[\exists i \in [n] : \bar{S}_i \notin H \cup \{S\} \cup \{fail\}] \leq N^{-(c_{\mathrm{ssnm}}-1)}.$$

The proof of Lemma 6.6 appears in Section 6.2.2, and implements the intuition used in proving a similar lemma in [SS22]. However, there are some technical differences that are caused by the modified construction.

Next we will use the evasiveness property of Dec to show that for every fixed $\bar{s} \in \{0,1\}^{c_k \cdot \log N}$, the probability that the procedure *DecodeUsingCandidate*$(\bar{s})$ will not fail is small, where the probability is in $\mathrm{expr}^{\mathrm{ed}}(m, C)$.

Lemma 6.7 (Using evasiveness)**.** *For every fixed $\bar{s} \in \{0,1\}^{c_k \cdot \log N}$,*

$$\Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[\textit{DecodeUsingCandidate}(\bar{s}) \neq fail] \leq \frac{1}{N^{c_{\mathrm{PRG}}}} + \frac{1}{N^{c_{\mathrm{BSC}}}}.$$

The proof of Lemma 6.7 appears in Section 6.2.3 and implements the same overall approch that was used for proving a similar lemma in [SS22]. This argument is now much more delicate due to the fact that our SS-non-malleable code, does not share some of the properties that were used in [SS22].

Putting the two lemmas together, we conclude that it is unlikely that there exists $i \in [n]$ such that $\bar{S}_i$ is successful, and $\bar{S}_i \neq S$. This is done formally in the next claim.

Claim 6.8.

$$\Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[\exists i \in [n] : \textit{DecodeUsingCandidate}(\bar{S}_i) \neq fail, \textit{ and } \bar{S}_i \neq S] < \nu/2.$$

Proof. Let:

$$P = \Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[\exists i \in [n] : \textit{DecodeUsingCandidate}(\bar{S}_i) \neq fail \text{ and } \bar{S}_i \neq S].$$

We have that:

$$\begin{aligned}
P &\leq \Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[\exists i \in [n] : \bar{S}_i \notin H \cup \{S\} \cup \{fail\}] + \Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[\exists \bar{s} \in H : \textit{DecodeUsingCandidate}(\bar{s}) \neq fail] \\
&\leq N^{-(c_{\mathrm{ssnm}}-1)} + \sum_{\bar{s} \in H} \Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[\textit{DecodeUsingCandidate}(\bar{s}) \neq fail] \\
&\leq N^{-(c_{\mathrm{ssnm}}-1)} + N^{a'_{\mathrm{ssnm}}+1} \cdot (\frac{1}{N^{c_{\mathrm{PRG}}}} + \frac{1}{N^{c_{\mathrm{BSC}}}}) \\
&\leq \frac{\nu}{2},
\end{aligned}$$

where the second inequality is by Lemma 6.6, the third inequality is by Lemma 6.7, and the final inequality follows because $\nu = N^{-c}$, and we have that $c_{\mathrm{ssnm}} > c + c_0$, and both c_{BSC} and c_{PRG} are larger than $c + c_{a'_{\mathrm{ssnm}}} + c_0$. We can choose c_0 to be sufficiently large so that the inequality holds. □

We are now ready to prove Lemma 6.5.

Proof. (of Lemma 6.5) By Lemma 6.4 we have that:

$$\Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[\exists i : \bar{S}_i = S \text{ and } \textit{DecodeUsingCandidate}(\bar{S}_i) = m \neq fail] \geq 1 - \nu/2.$$

Combining this with Claim 6.8 we have that except with probability ν, we have that in $\mathrm{expr}^{\mathrm{ed}}(m, C)$ the two events below occur:

- $\{\exists i \in [n] : \bar{S}_i = S$ and $\bar{S}_i$ is successful$\}$.
- $\{\forall i \in [n] :$ either $\bar{S}_i = S$ or $\bar{S}_i$ is not suceessful$\}$.

When these two events occur, we have that there is a unique $\bar{s} \in$ *CANDIDATES* that is successful, and $S^* = \bar{s} = S$. □

This concludes the proof of Theorem 6.1. It remains to prove the lemmas that appeared inside the proof, and this is done in the remainder of the section. However, before doing that, we compare the construction and proof to those in [SS22].

6.2.1 Comparison of the construction and analysis to earlier work

The reason that we cannot use the construction and proof of [SS22] is that the SS-non-malleable codes that we have do not have some of the additional properties that were achieved by the probabilistic argument in [SS22], see Remark 5.4. This creates several complications:

- Rather than having one "control code" $\mathrm{Enc}_{\mathrm{ctrl}}$ that is SS-non-malleable, pseudorandom, and decodes from a about a $1/4$ fraction of errors (as is the case in [SS22]). We now have several codes where each one has one of these properties. Fortunately, this isn't a problem, as we can combine these components together (in fact, this was already done in some of the previous papers [GS16, SS21a]): When preparing the n_{ctrl} control blocks $c_1, \ldots, c_{n_{\mathrm{ctrl}}}$, the previous construction took $c_j = \mathrm{Enc}_{\mathrm{ctrl}}(s, r_j)$ for independent random strings $r_1, \ldots, r_{n_{\mathrm{ctrl}}}$. We can achieve the same effect by first using a random string r_0, to compute $o = \mathrm{Enc}_{\mathrm{ssnm}}(s, r_0)$. We then encode o using a Reed-Solomon code $\mathrm{Enc}_{\mathrm{RS}}$, and each output symbol is encoded with the pseudorandom code $\mathrm{Enc}_{\mathrm{pr}}$.

- A more serious problem is that in [SS22] the SS-non-malleable code was secure when choosing the function $\psi(s)$ which in addition to $G(s_{\mathrm{PRG}})$ also produced a description of $\mathrm{Samp}(s_{\mathrm{samp}})$ and π_{s_π}. The second two components do not look pseudorandom. We are forced to use

$$\psi(s) = \hat{G}(s) = G(s_{\mathrm{samp}}), G(s_\pi), G(s_{\mathrm{PRG}}),$$

  as we must use a function ψ that is a seed extending PRG. This creates complications as in order to use SS-non-malleability, in the proof of Lemma 6.6, we need to argue in that a small circuit that sees $\hat{G}(S), \mathrm{Enc}_{\mathrm{ssnm}}(S)$ can prepare the corresponding codeword $\mathrm{Enc}(m; (S, R))$. This is a problem, as such an adversary has no information about S. However, by carefully modifying the construction (so that *control* and s_π are functions of $G(s_\pi)$ and $G(s_{\mathrm{samp}})$ we can arrange it so that the codeword can indeed be prepared by a small circuit that sees $\hat{G}(S), \mathrm{Enc}_{\mathrm{ssnm}}(S)$.
- The previous modification causes other problems. It is now computationally difficult to compute the *control* and π_{s_π} as a function of s, as this requires computing G (which cannot be done by a small circuit). This creates complications in the proof of Lemma 6.7. Fortunately, in that proof, it is possible to do the analysis when S_{PRG} is kept uniform, and S_{samp}, S_π are fixed. In such a setup, *control* and π_{s_π} are fixed and can be hardwired to the circuit.
- The fact that the "control code" is composed of several components, dictates a complexity leveraging argument where some components fool other components. In particular, we are now forced to set the evasiveness property of $(\mathrm{Enc}_{\mathrm{BSC}}, \mathrm{Dec}_{\mathrm{BSC}})$ to work against circuits that can compute $\mathrm{Enc}_{\mathrm{pr}}, \mathrm{Dec}_{\mathrm{pr}}$. This in particular means that the pseudorandomness properties of $\mathrm{Enc}_{\mathrm{pr}}$ is quite weak, and we cannot use its pseudorandomness against some of the other components. In particular, when arguing that the codeword is pseudorandom, the control part (that was prepared using $\mathrm{Enc}_{\mathrm{pr}}$) is pseudorandom with error ϵ_{pr} which is not sufficiently small to do a union bound over all elements in the set H obtained by SS-non-malleability. Because of this, we cannot rely on the pseudorandomness of the control part, when arguing that evasiveness on random strings implies evasiveness on pseudorandom strings. Instead, in the proof of Lemma 6.7 we use a delicate argument that pretends that the control block is composed of "errors" introduced by the channel.
- Another consequence of this "leveraging" is that we cannot expect $\mathrm{Enc}_{\mathrm{pr}}$ to be pseudorandom for circuits that can compute $\mathrm{Dec}_{\mathrm{BSC}}$. This means that while the codeword looks pseudorandom to the channel, it does not look pseudorandom when the channel is composed with $\mathrm{Dec}_{\mathrm{BSC}}$. Fortunately, this is easy to handle, as in the proof of Lemma 6.4 it is sufficient that $\mathrm{Enc}_{\mathrm{pr}}$ is pseudorandom for a circuit that needs to check whether $\mathrm{Dec}_{\mathrm{BSC}}$ returns the specific message m, and this is the reason we require the final condition in Theorem 4.2, so that a small circuit can perform this task, without running $\mathrm{Dec}_{\mathrm{BSC}}$.

6.2.2 Using SS-non-malleability: Proof of Lemma 6.6

We need to define a set $H \subseteq \{0,1\}^{c_k \cdot \log N}$ of size $N^{c_{a'_{\mathrm{ssnm}}+1}}$. We will use the channel circuit C to define n adversaries $C^1, \ldots, C^n$ for the SS-non-malleability property of $\mathrm{Enc}_{\mathrm{ssnm}}$. Recall that any adversary C^i to the SS-non-malleability is expecting to receive input of the form:

$$\hat{G}(S), \mathrm{Enc}_{\mathrm{ssnm}}(S, R_0),$$

for a uniformly chosen $S \leftarrow \{0,1\}^{c_k \cdot \log N}$, and a uniformly chosen $R_0 \leftarrow \{0,1\}^{N^{d_{\mathrm{ssnm}}}}$. In our setting, $\hat{G}(S) = G(S_{\mathrm{samp}}), G(S_\pi), G(S_{\mathrm{PRG}})$ as defined in Figure 2. Intuitively, such an adversary is trying to produce a string that will be decoded by $\mathrm{Dec}_{\mathrm{ssnm}}$ to a string $\bar{S} \neq S$ that is "correlated" with S.

Definition 6.9. *For every $i \in [n]$, we define a randomized circuit C^i as follows:*

Input: *The adversary C^i receives a string g, (which is supposed to be $\hat{G}(S)$) and a string e which is supposed to be $\mathrm{Enc}_{\mathrm{ssnm}}(S, R_0)$.*

Operation: *The adversary C^i is hardwired with the fixed message m, the fixed string $\mathrm{Enc}_{\mathrm{BSC}}(m)$, and the circuit C. It will act as follows:*

Simulate the encoding: *We will now notice that if indeed g and e are as intended, then C^i can prepare a string that is distributed like $Z = \mathrm{Enc}(m; (S, R))$ in $\mathrm{expr}^{\mathrm{ed}}(m, C)$. More specifically.*

- *The control indices* **CONTROL** $= \{I_1 < \ldots < I_{n_{\mathrm{ctrl}}}\}$ *are obtained by* **CONTROL** $= \mathrm{Samp}(G(S_{\mathrm{samp}}))$, *and $G(S_{\mathrm{samp}})$ is a substring of $\hat{G}(S)$, and so, C^i can prepare* **CONTROL** *by applying* Samp.
- *The permutation π_{S_π} is obtained (see Definition 2.31) by a straightforward computation as a function of $G(S_\pi)$. Therefore, a full description of this permutation is available to C^i who has $G(S_\pi)$ as a substring of $\hat{G}(S)$.*
- *The string $G(S_{\mathrm{PRG}})$ is a substring of $\hat{G}(S)$ and is available to C^i.*
- *C^i (that is a randomized circuit) can choose $R_1, \ldots, R_{n_{\mathrm{ctrl}}} \leftarrow \{0,1\}^{n^{d_{\mathrm{ssnm}}}}$ (just like the encoding does).*
- *Assuming the inputs are as intended, from here, C^i (that is hardwired with $\mathrm{Enc}_{\mathrm{BSC}}(m)$ and receives $\mathrm{Enc}_{\mathrm{ssnm}}(S, R_0)$ as input) can prepare $Z = \mathrm{Enc}(m; (S, R_0, R_1, \ldots, R_{n_{\mathrm{ctrl}}}))$ by following the same procedure that* Enc *uses (as in $\mathrm{expr}^{\mathrm{ed}}(m, C)$). (A key observation is that C^i does not need to compute $\mathrm{Enc}_{\mathrm{ssnm}}$, π_G, G or $\mathrm{Enc}_{\mathrm{BSC}}$. The only nontrivial operation it needs to perform is computing $\mathrm{Enc}_{\mathrm{pr}}$, and therefore, simulating the encoding can be computed by a circuit of size $N^{d_{\mathrm{pr}}+c_1}$ for a universal constant c_1).*

Simulate the channel: *The adversary will compute $E = C(Z)$ and $\bar{V} = Z \oplus E$. (This is what the channel C does on the encoding, and if the inputs to C^i are as intended, then this is a simulation of $\bar{V}$ from $\mathrm{expr}^{\mathrm{ed}}(m, C)$).*

Simulate the beginning of the decoding: *C^i implements the "compute candidates" phase of $\mathrm{expr}^{\mathrm{ed}}(m, C)$ exactly in the same manner as in $\mathrm{expr}^{\mathrm{ed}}(m, C)$ to produce $\bar{O}_1, \ldots, \bar{O}_n$. The only nontrivial operation in this phase is running $\mathrm{Dec}_{\mathrm{pr}}$ which can be done in size $N^{d_{\mathrm{pr}}}$.*

Output a single candidate: *Finally, C^i outputs $\bar{O}_i$.*

We record the following obvious properties of the adversary C^i:

Claim 6.10. *There exists a universal constant c_1 such that for every $i \in [n]$, C^i is a randomized circuit of size $N^c + O(N^{c_1+d_{\mathrm{pr}}})$, and furthermore:*

- *Let S, R_0 be chosen as in $\mathrm{expr}^{\mathrm{ed}}(m, C)$. For every $i \in [n]$, if C^i receives the input:*

$$\hat{G}(S), \mathrm{Enc}_{\mathrm{ssnm}}(S, R_0)$$

then it outputs a variable that is distributed like $\bar{O}_i$ from $\mathrm{expr}^{\mathrm{ed}}(m, C)$.

- *In particular, it follows that in the decoding phase in $\mathrm{expr}^{\mathrm{ed}}(m, C)$ we have that $\bar{S}_i = \mathrm{Dec}_{\mathrm{ssnm}}(\bar{S}_i)$ (in $\mathrm{expr}^{\mathrm{ed}}(m, C)$) is distributed like:*

$$\mathrm{Dec}_{\mathrm{ssnm}}(C^i(\hat{G}(S), \mathrm{Enc}_{\mathrm{ssnm}}(S, R_0))).$$

Recall that we have chosen $c_{\mathrm{ssnm}} = d_{\mathrm{pr}} + c_0$, where $d_{\mathrm{pr}} > c$. We will choose the constant c_0 to be sufficiently large so that the size of C^i is bounded by $N^{c_{\mathrm{ssnm}}}$. Recall also, that in Figure 2 we have established

that $\hat{G}$ is a seed-extending $\frac{1}{N^{c'_{\text{ssnm}}}}$-PRG for circuits of size $N^{c'_{\text{ssnm}}}$, and have chosen Enc_{ssnm} so that it is a $(\hat{G}, N^{a'_{\text{ssnm}}}, \frac{1}{N^{c_{\text{ssnm}}}})$-SS-non-malleable for circuits of size $N^{c_{\text{ssnm}}}$. This means that for every $i \in [n]$, C^i is an adversary that is handled by the SS-non-malleability $(\text{Enc}_{\text{ssnm}}, \text{Dec}_{\text{ssnm}})$. We can conclude that for every $i \in [n]$, there exists a set $H_{C^i} \subseteq \{0,1\}^{c_k \cdot \log N}$ with $|H_{C^i}| \leq N^{a'_{\text{ssnm}}}$, such that

$$\Pr_{\substack{S \leftarrow \{0,1\}^{c_k \cdot \log N}, R_0 \leftarrow \{0,1\}^{N^{d_{\text{ssnm}}}} \\ R_1, \ldots, R_{n_{\text{ctrl}}} \leftarrow \{0,1\}^{c_d \cdot \log N}}} [\text{Dec}(C^i(\hat{G}(S), \text{Enc}_{\text{ssnm}}(S, R_0))) \notin H_{C^i} \cup \{S\} \cup \{fail\}] \leq \frac{1}{N^{c_{\text{ssnm}}}}.$$

By Claim 6.10, we get that for every $i \in [n]$:

$$\Pr_{\text{expr}^{\text{ed}}(m,C)} [\bar{S}_i \notin H_{C^i} \cup \{S\} \cup \{fail\}] < \frac{1}{N^{c_{\text{ssnm}}}}.$$

We define:

$$H = \bigcup_{i \in [n]} H_{C^i}.$$

It follows that $|H| \leq n \cdot N^{a'_{\text{ssnm}}} \leq N^{a'_{\text{ssnm}}+1}$, and by a union bound over the $n \leq N$ choices for $i \in [n]$, we get that:

$$\begin{aligned} \Pr_{\text{expr}^{\text{ed}}(m,C)} [\exists i \in [n] : \bar{S}_i \notin H \cup \{S\} \cup \{fail\}] &\leq \sum_{i \in [n]} \Pr_{\text{expr}^{\text{ed}}(m,C)} [\bar{S}_i \notin H_{C^i} \cup \{S\} \cup \{fail\}] \\ &< n \cdot N^{-c_{\text{ssnm}}} \\ &\leq N^{-(c_{\text{ssnm}}-1)}, \end{aligned}$$

and Lemma 6.6 follows.

6.2.3 Using evasiveness: Proof of Lemma 6.7

Fix some $\bar{s} \in \{0,1\}^{c_k \cdot \log N}$. We are interested in the operation of *DecodeUsingCandidate*$(\bar{s})$ in the experiment $\text{expr}^{\text{ed}}(m, C)$. In this proof, it will be convenient to consider a version of $\text{expr}^{\text{ed}}(m, C)$ where S_{samp}, S_π and $R_0, \ldots, R_{n_{\text{ctrl}}}$ are fixed (which means that only S_{PRG} is chosen at random).

Definition 6.11. *For every $s_{\text{samp}}, s_\pi \in \{0,1\}^{\ell'}$, every $r_0 \in \{0,1\}^{n^{d_{\text{ssnm}}}}$ and every $r_1, \ldots, r_{n_{\text{ctrl}}} \in \{0,1\}^{c_d \cdot \log N}$ we define the experiment $\text{expr}^{\text{ed}}_{s_{\text{samp}}, s_\pi, r_0, r_1, \ldots, r_{n_{\text{ctrl}}}}$ to be the experiment $\text{expr}^{\text{ed}}(m, C)$ conditioned on the event:*

$$\{S_{\text{samp}} = s_{\text{samp}}, S_\pi = s_\pi, R_0 = r_0, R_1 = r_1, \ldots, R_{n_{\text{ctrl}}} = r_{n_{\text{ctrl}}}\}.$$

Namely, the experiment in which only S_{PRG} is chosen at random.

It is sufficient to prove Lemma 6.7 for every fixing of S_{samp}, S_π and $R_0, \ldots, R_{n_{\text{ctrl}}}$. For this purpose, we consider some values of s_{samp}, s_π and $r_0, \ldots, r_{n_{\text{ctrl}}}$, and let expr denote the experiment $\text{expr}^{\text{ed}}_{s_{\text{samp}}, s_\pi, r_0, r_1, \ldots, r_{n_{\text{ctrl}}}}$. We need to prove that:

$$\Pr_{\text{expr}} [\textit{DecodeUsingCandidate}(\bar{s}) \neq fail] \leq \frac{1}{N^{c_{\text{PRG}}}} + \frac{1}{N^{c_{\text{BSC}}}}.$$

We plan to use the pseudorandomness of the seed-extending PRG G, and the evasiveness of $(\text{Enc}_{\text{BSC}}, \text{Dec}_{\text{BSC}})$. Our plan is to define two circuits: a "distinguisher" to the PRG, and a "channel" for the evasiveness experiment. Loosely speaking, the distinguisher circuit, will check whether the channel makes Dec_{BSC} output a

value different from fail, and we will arrange it so that on the pseudorandom distribution $(S_{\mathrm{PRG}}, G(S_{\mathrm{PRG}}))$ this experiment will correspond with expr while on the uniform distribution (S_{PRG}, B) where $B \leftarrow U_{N_{\mathrm{data}}}$ is independent of S_{PRG}, this will correspond to the evasiveness experiment considered in Definition 4.1. We will construct the "distinguisher" and "channel" in stages, starting with the following sub-procedures P_{data} and P_{ctrl}:

Definition 6.12. *We define a circuit $P_{\mathrm{data}} : \{0,1\}^{N_{\mathrm{data}}} \to \{0,1\}^{N_{\mathrm{data}}}$ as follows: $P_{\mathrm{data}}(b) = \pi_{s_\pi}^{-1}(\mathrm{Enc}_{\mathrm{BSC}}(m)) \oplus b$. Note that s_π and $\mathrm{Enc}_{\mathrm{BSC}}(m)$ are fixed, and can be hardwired to P_{data}.*

This definition was made so that:

Claim 6.13.

- *P_{data} is a circuit of size $\mathrm{poly}(N)$.*
- *$P_{\mathrm{data}}(G(S_{\mathrm{PRG}}))$ is distributed like $Z_{\mathrm{data}}^{\mathit{control}}$ in expr.*
- *$P_{\mathrm{data}}(U_{N_{\mathrm{data}}})$ is uniform over $\{0,1\}^{N_{\mathrm{data}}}$.*

Definition 6.14. *We define a circuit $P_{\mathrm{ctrl}} : \{0,1\}^{\ell'} \to \{0,1\}^{N_{\mathrm{ctrl}}}$ as follows:*

- *On input $s_{\mathrm{PRG}} \in \{0,1\}^{\ell'}$, compute $o = \mathrm{Enc}_{\mathrm{ssnm}}((s_{\mathrm{samp}}, s_\pi, s_{\mathrm{PRG}}))$. (Here, s_{samp} and s_π are fixed, and s_{PRG} is the input to P_{ctrl}).*
- *Compute $w = \mathrm{Enc}_{\mathrm{RS}}(o, r_0)$.*
- *Note that control $= \mathrm{Samp}(G(s_{\mathrm{samp}}) = \{i_1 < \ldots < i_{n_{\mathrm{ctrl}}}\}$ is fixed, and can be hardwired to P_{ctrl}.*
- *For every $j \in [n_{\mathrm{ctrl}}]$ compute $z_j = \mathrm{Enc}_{\mathrm{pr}}((i_j, w_j), r_j)$.*
- *Output $z_1, \ldots, z_{n_{\mathrm{ctrl}}}$.*

This definition was made so that:

Claim 6.15.

- *P_{ctrl} is a circuit of size $N^{d_{\mathrm{pr}}+O(1)} + N^{d_{\mathrm{ssnm}}} + N^{O(1)}$.*
- *$P_{\mathrm{ctrl}}(S_{\mathrm{PRG}})$ is distributed like $Z_{\mathrm{ctrl}}^{\mathit{control}}$ in expr.*

Together, this implies that:

Claim 6.16.

- *For $S_{\mathrm{PRG}} \leftarrow \{0,1\}^{\ell'}$ and $B = G(S_{\mathrm{PRG}})$, $X = P_{\mathrm{data}}(B)$, and $Y = P_{\mathrm{ctrl}}(S_{\mathrm{PRG}})$, we have that $(X, Y)^{\mathit{control}}$ is distributed like the codeword Z from expr.*
- *For $S_{\mathrm{PRG}} \leftarrow \{0,1\}^{\ell'}$ and $B \leftarrow \{0,1\}^{N_{\mathrm{data}}}$. $X = P_{\mathrm{data}}(B)$, and $Y = P_{\mathrm{ctrl}}(S_{\mathrm{PRG}})$, we have that X, Y are independent, and X is uniform over $\{0,1\}^{N_{\mathrm{data}}}$.*

Our plan is to prove the lemma by relating the two experiments above to expr, and the evasiveness experiment. We then plan to use the pseudorandomness of G. For this purpose, we will define a potential distinguisher for G.

Definition 6.17. *We define a circuit $D_{\bar{s}} : \{0,1\}^{\ell'} \times \{0,1\}^{N_{\mathrm{data}}} \to \{0,1\}$ as follows: Given input $s_{\mathrm{PRG}} \in \{0,1\}^{\ell'}$ and $b \in \{0,1\}^{N_{\mathrm{data}}}$ do the following:*

Simulate encoding: *Prepare a string z as follows:*

- *Compute* $x = P_{\mathrm{data}}(b)$ *and* $y = P_{\mathrm{ctrl}}(s_{\mathrm{PRG}})$.
- *Compute* $z = (x, y)^{\textit{control}}$. *(That is, we prepare a "codeword").*

Simulate channel: *Prepare a string* $\bar{v}$ *as follows:*

- *Compute* $e = C(z)$. *(That is, prepare an error pattern).*
- *Compute* $\bar{v} = z \oplus e$. *(That is compute the received word).*

Simulate *DecodeUsingCandidate*$(\bar{s})$: *Apply DecodeUsingCandidate as follows:*

- *Compute* $\bar{v}_{\mathrm{data}} = \overline{v_{\mathrm{data}}^{\textit{control}}}$.
- *Compute* $\bar{y} = \bar{v}_{\mathrm{data}} \oplus G(\bar{s}_{\mathrm{PRG}})$ *noting that* $G(\bar{s}_{\mathrm{PRG}})$ *is fixed and can be hardwired to the circuit.*
- *Compute* $\bar{x} = \pi_{\bar{s}_\pi}(\bar{y})$ *noting that* $\pi_{\bar{s}_\pi}$ *is fixed and can be hardwired to the circuit.*
- *Compute* $\bar{m} = \mathrm{Dec}_{\mathrm{BSC}}(\bar{x})$.

Produce output: *Output zero if* $\bar{m} = fail$, *and one otherwise.*

This definition was made so that:

Claim 6.18.

- $D_{\bar{s}}$ *is a circuit of size* $N^{d_{\mathrm{pr}}+d_{\mathrm{ssnm}}+d_{\mathrm{BSC}}+c+O(1)}$. *(Here, the main point is that* $D_{\bar{s}}$ *applied* G *on fixed strings, and therefore does not need to run the PRG* G *which would have required more resources).*
- *For* $S_{\mathrm{PRG}} \leftarrow \{0,1\}^{\ell'}$ *and* $B = G(S_{\mathrm{PRG}})$,

$$\Pr[D_{\bar{s}}(S_{\mathrm{PRG}}, B) = 1] = \Pr_{\mathrm{expr}}[\textit{DecodeUsingCandidate}(\bar{s}) \neq fail].$$

(This immediately follows as $D_{\bar{s}}$ *simulates all the steps correctly).*

This connects the probability that $D_{\bar{s}}$ answers one on $(S_{\mathrm{PRG}}, G(S_{\mathrm{PRG}}))$ to the probability of the event that we want to bound.

We have chosen c_{PRG} to be larger than $d_{\mathrm{pr}} + d_{\mathrm{ssnm}} + d_{\mathrm{BSC}} + c + c_0$. By choosing c_0 to be sufficiently large, we have that $D_{\bar{s}}$ is a circuit of size $N^{c_{\mathrm{PRG}}}$. As G is a seed-extending $\frac{1}{N^{c_{\mathrm{PRG}}}}$-PRG for circuits of size $N^{c_{\mathrm{PRG}}}$, we conclude that:

Claim 6.19.

$$\Pr_{S_{\mathrm{PRG}} \leftarrow \{0,1\}^{\ell'}}[D_{\bar{s}}(S_{\mathrm{PRG}}, G(S_{\mathrm{PRG}})) = 1] \leq \Pr_{S_{\mathrm{PRG}} \leftarrow \{0,1\}^{\ell'}, B \leftarrow \{0,1\}^{N_{\mathrm{data}}}}[D_{\bar{s}}(S_{\mathrm{PRG}}, B) = 1] + \frac{1}{N^{c_{\mathrm{PRG}}}}.$$

In order to prove the lemma, it remains to show that:

$$\Pr_{S_{\mathrm{PRG}} \leftarrow \{0,1\}^{\ell'}, B \leftarrow \{0,1\}^{N_{\mathrm{data}}}}[D_{\bar{s}}(S_{\mathrm{PRG}}, B) = 1] \leq \frac{1}{N^{c_{\mathrm{BSC}}}}.$$

In fact, we will show that for every fixing s_{PRG} of S_{PRG},

$$\Pr_{B \leftarrow \{0,1\}^{N_{\mathrm{data}}}}[D_{\bar{s}}(s_{\mathrm{PRG}}, B) = 1] \leq \frac{1}{N^{c_{\mathrm{BSC}}}}.$$

For this purpose, we fix some value of s_{PRG}. We would like to relate the experiment above to the evasiveness experiment of Definition 4.1. For this purpose, we will consider the operation of $D_{\bar{s}}(s_{\mathrm{PRG}}, B)$ on $B \leftarrow \{0,1\}^{N_{\mathrm{data}}}$. Note that in this experiment:

- $y = P_{\text{ctrl}}(s_{\text{PRG}})$ is fixed.
- $X = P_{\text{data}}(B)$ is uniformly distributed.
- Therefore, the constructed word $Z = (X, y)^{\overline{\textit{control}}}$ consists of a fixed control part, and a uniform data part (where these parts were chosen according to s_{samp}).
- The channel C is applied on Z to produce the received word $\bar{V} = Z \oplus C(Z)$.
- We would like to understand this operation (of going from Z to $\bar{V}$) as applying a channel $C' : \{0,1\}^{N_{\text{data}}} \to \{0,1\}^{N_{\text{data}}}$ on $Z_{\text{data}}^{\overline{\textit{control}}}$ to produce $\bar{V}_{\text{data}}^{\overline{\textit{control}}}$. The motivation is that the "simulate *DecodeUsingCandidate*$(\bar{s})$" is only interested in $\bar{V}_{\text{data}}^{\overline{\textit{control}}}$. (This gives rise to a channel C' that receives $Z_{\text{data}}^{\overline{\textit{control}}}$ and outputs a noise vector e' of length N_{data} that leads to the received word $\bar{V}_{\text{data}}^{\overline{\textit{control}}} = Z_{\text{data}}^{\overline{\textit{control}}} \oplus e'$.
- C' relies on C (which has access to all of Z, and not just to $Z_{\text{data}}^{\overline{\textit{control}}}$. Therefore, the circuit size of C' will be the sum of the circuit size of C, and the circuit complexity of generating Z from $Z_{\text{data}}^{\overline{\textit{control}}}$.
- Fortunately, at this point Z_i for $i \in \overline{\textit{control}}$, is either fixed (if it belongs to y) or uniform and independent of $Z_{\text{data}}^{\overline{\textit{control}}}$ (if the block Z_i is a part of X).
- This means that the circuit complexity of the channel C' is comparable to that of C, and C' has circuits of size $N^{c+O(1)}$.
- While the string $Z_{\text{data}}^{\overline{\textit{control}}}$ is not uniformly distributed, every block of $Z_{\text{data}}^{\overline{\textit{control}}}$ is either a block of X (which is uniformly distributed) or a block of y (which is fixed). The length of y is N_{ctrl}. This means we can imagine that C' receive a uniform $Z' \leftarrow \{0,1\}^{N_{\text{data}}}$ (as in the evasiveness experiment) and its first step is to replace at most N_{ctrl} of its input bits by fixed values from y and then proceeds as before. (This may potentially add N_{ctrl} errors to the pN errors introduced by C, and so C' may introduce $pN + N_{\text{ctrl}}$ errors on its N_{data} bit long input).

Summing up this discussion, when $D_{\bar{S}}$ receives the fixed s_{PRG}, and a uniform $B \leftarrow \{0,1\}^{N_{\text{data}}}$, there is a channel C' of size $N^{c+O(1)}$ that induces $pN + N_{\text{ctrl}}$ errors, so that if we define $h : \{0,1\}^{N_{\text{data}}} \to \{0,1\}^{N_{\text{data}}}$ according to the final two operations of *DecodeUsingCandidate*, namely:

$$h(z') = \pi_{\bar{s}_\pi}(z' \oplus G(\bar{s}_{\text{PRG}})),$$

then, we have that:

Claim 6.20.

$$\Pr_{B \leftarrow \{0,1\}^{N_{\text{data}}}}[D_{\bar{s}}(s_{\text{PRG}}, B) = 1] = \Pr_{Z' \leftarrow \{0,1\}^{N_{\text{data}}}}[\text{Dec}_{\text{BSC}}(h(Z' \oplus C'(Z'))) \neq fail].$$

The right hand side, is almost exactly the probability considered in the evasiveness experiment. It would be exactly the evasiveness experiment if h is the identity function. Note that in the definition of h, $G(\bar{s}_{\text{PRG}}) = g$ for some fixed string g, and $\pi_{\bar{s}_\pi}$ is some fixed permutation σ. This means that $h(z') = \sigma(z' \oplus g)$. We will soon observe that the evasiveness guarantee also holds with such a function h. Before doing that, let us first compute the fraction of errors introduced by C'.

The fraction of errors introduced by C' is

$$\frac{pN + N_{\text{ctrl}}}{N_{\text{data}}} = p \cdot \frac{N}{N_{\text{data}}} + \frac{N_{\text{ctrl}}}{N_{\text{data}}} = p(1 + o(1)) + \frac{n_{\text{ctrl}}}{n_{\text{data}}} = p(1 + o(1)) \leq p_{\text{BSC}},$$

where here we use that $n_{\text{ctrl}} = n^{0.9}$ and $p_{\text{BSC}} = p \cdot (1 + \frac{\epsilon}{10})$.

The circuit C' is a circuit of size $N^{c+O(1)}$ and we have chosen $c_{\text{BSC}} = d_{\text{ssnm}} + c_0$ which is larger than $c + O(1)$ for a sufficiently large c_0. We will now argue that the evasiveness of $(\text{Enc}_{\text{BSC}}, \text{Dec}_{\text{BSC}})$ also holds with the function h.

Claim 6.21.

$$\Pr_{Z' \leftarrow \{0,1\}^{N_{\text{data}}}}[\text{Dec}_{\text{BSC}}(h(Z' \oplus C'(Z'))) = fail] \leq \frac{1}{N^{c_{\text{BSC}}}}.$$

Proof. We have that $h(z') = \sigma(z' \oplus g)$ for a fixed string $g \in \{0,1\}^{N_{\text{data}}}$ and a fixed permutation σ over $[N_{\text{data}}]$. We will reduce the experiment in the claim to the evasiveness experiment. Consider the experiment in the claim, namely $Z' \leftarrow \{0,1\}^{N_{\text{data}}}$. Let us define the random variable

$$W = \sigma(Z' \oplus g).$$

Note that just like Z', W is also uniformly distributed over $\{0,1\}^{N_{\text{data}}}$. We now define a new "channel" $T(w)$ as follows:

$$T(w) = \sigma(C'(\sigma^{-1}(w) \oplus g)).$$

We have already established that C' is a circuit of size $N^{C_{\text{BSC}}}$ that induces at most a p_{BSC} fraction of errors. Note that by definition, T inherits these two properties of C'.

The definition of W and T is made so that:

$$\begin{aligned} W \oplus T(W) &= \sigma(Z' \oplus g) \oplus \sigma(C'(\sigma^{-1}(\sigma(Z' \oplus g)) \oplus g)) \\ &= \sigma(Z' \oplus g) \oplus \sigma(C'(Z' \oplus g \oplus g)) \\ &= \sigma(Z' \oplus g) \oplus \sigma(C'(Z')) \\ &= \sigma(Z' \oplus C'(Z') \oplus g) \\ &= h(Z' \oplus C(Z')). \end{aligned}$$

In other words,

$$\Pr_{Z' \leftarrow \{0,1\}^{N_{\text{data}}}}[\text{Dec}_{\text{BSC}}(h(Z' \oplus C'(Z'))) \neq fail] = \Pr_{W \leftarrow \{0,1\}^{N_{\text{data}}}}[\text{Dec}_{\text{BSC}}(W \oplus T(W)) \neq fail].$$

The right hand side is exactly the scenario considered in the evasiveness experiment. Therefore, by the evasiveness of $(\text{Enc}_{\text{BSC}}, \text{Dec}_{\text{BSC}})$ this probability is upper bounded by $\frac{1}{N^{c_{\text{BSC}}}}$. □

Altogether, we get that:

$$\begin{aligned} \Pr_{\text{expr}}[\textit{DecodeUsingCandidate}(\bar{s}) \neq fail] &= \Pr_{S_{\text{PRG}} \leftarrow \{0,1\}^{\ell'}}[D_{\bar{s}}(S_{\text{PRG}}, G(S_{\text{PRG}})) = 1] \\ &\leq \Pr_{S_{\text{PRG}} \leftarrow \{0,1\}^{\ell'}, B \leftarrow \{0,1\}^{N_{\text{data}}}}[D_{\bar{s}}(S_{\text{PRG}}, B) = 1] + \frac{1}{N^{c_{\text{PRG}}}} \\ &= \Pr_{Z' \leftarrow \{0,1\}^{N_{\text{data}}}}[\text{Dec}_{\text{BSC}}(h(Z' \oplus C'(Z'))) \neq fail] + \frac{1}{N^{c_{\text{PRG}}}} \\ &= \Pr_{W \leftarrow \{0,1\}^{N_{\text{data}}}}[\text{Dec}_{\text{BSC}}(W \oplus T(W)) \neq fail] + \frac{1}{N^{c_{\text{PRG}}}} \\ &\leq \frac{1}{N^{c_{\text{BSC}}}} + \frac{1}{N^{c_{\text{PRG}}}}, \end{aligned}$$

as required.

Remark 6.22. *In the proof above, we treated Y as if it has no pseudorandomness properties. This seems strange as the code Enc_{pr} has pseudorandomness properties, and we could have hoped to use these. The reason that this was not done is that the error parameter in the pseudorandomness of Enc_{pr} is $\frac{1}{N^{c_{\text{pr}}}}$ which is larger than the error of $\frac{1}{N^{c_{\text{PRG}}}} + \frac{1}{N^{c_{\text{BSC}}}}$ that we obtained. This difference is critical, as we are planning to use this error to do a union bound over the $N^{C_{a'_{\text{ssnm}}}}$ elements in H, and the way we chose the parameters dictates that $N^{c_{\text{pr}}} < N^{C_{a'_{\text{ssnm}}}}$ (as we have chosen the parameters of the SS-non-malleable code Enc_{ssnm} as a function of Enc_{pr} and leveraged the assumption, so that Enc_{ssnm} fools adversaries that are able to run $\text{Enc}_{\text{pr}}, \text{Dec}_{\text{pr}}$.*

6.2.4 The correct control string is one of the candidates: Proof of Lemma 6.4

The proof of Lemma 6.4 is very similar to the corresponding proofs in earlier works [GS16, SS21a, SS22] that achieved list-decoding, rather than unique decoding. We cannot directly cite these proofs, as our construction has some differences compared to the aforementioned previous work (and these differences are necessary to make argument go through). Nevertheless, the argument that we present here is essentially identical to previous work, modulu these modifications.

We need the following notion of a milestone function. This notion emerged from previous work [GS16, SS21a].

Definition 6.23 (Milestone function)**.** *We define a function* $A : \{0,1\}^{\ell'} \times \{0,1\}^{\ell'} \times \{0,1\}^N \to \{0,1\}$ *as follows: On inputs* $s_{\mathrm{samp}}, s_\pi \in \{0,1\}^{\ell'}$ *and* $e \in \{0,1\}^n$*,* $A(s_{\mathrm{samp}}, s_\pi, e)$ *outputs one iff there exist at least* $a = \frac{n_{\mathrm{ctrl}}}{100}$ *elements* $i \in I = \mathrm{Samp}(G(s_{\mathrm{samp}}))$ *such that* $\mathsf{wt}(e_i) \leq \frac{1}{3}$*, and*

$$\mathrm{Dec}_{\mathrm{BSC}}(\mathrm{Enc}_{\mathrm{BSC}}(m) \oplus \pi_{s_\pi}(e^I_{\mathrm{data}})) = m.$$

We define $A_{s_{\mathrm{samp}},s_\pi}(e) = A(s_{\mathrm{samp}}, s_\pi, e)$.

Recall that at this point, we have fixed m and C, and A is designed so that in the experiment $\mathrm{expr}^{\mathrm{ed}}(m, C)$:

- $A(S_{\mathrm{samp}}, S_\pi, C(Z))$ checks whether there exist a control blocks $i \in I = \mathrm{Samp}(G(S_{\mathrm{samp}}))$ on which the block $E_i = C(Z)_i$ (of the error vector $E = C(Z)$) has low weight. If this happens then for these i, when computing $\mathrm{List}_i = \mathrm{Dec}_{\mathrm{pr}}(\bar{V}_i)$, we have that the error E_i added to Z_i is of relative Hamming weight at most $1/3$. This is exactly the scenario for which the decoding algorithm $\mathrm{Dec}_{\mathrm{pr}}$ was designed. This is because the code $(\mathrm{Enc}_{\mathrm{pr}}, \mathrm{Dec}_{\mathrm{pr}})$ was set up to be L_{pr}-weakly-list-decodable from radius $1/3$. It follows that for these a choices of i, List_i contains the correct string. This implies that in $\mathrm{List} = \cup_{i\in[n]}\mathrm{List}_i$, there are at least a pairs that are the original pairs. The Reed-Solomon code $\mathrm{Enc}_{\mathrm{RS}}$ was set up to list-decode if at least a out of the given pairs are correct. It follows that there exists $i^* \in [n]$ such that $\bar{O}_{i^*}$ is the original string O. This implies that $\bar{S}_{i^*} = \mathrm{Dec}_{\mathrm{ssnm}}(\bar{O}_{i^*})$ is the original control string S, and so the original control string S will appears in the list *CANDIDATES*.
- Furthermore, when continuing the decoding in $\mathrm{expr}^{\mathrm{ed}}(m, C)$ using the candidate $\bar{S}_i = S$, the original message m will be decoded. (Note that the noise pattern that is applied to $\mathrm{Enc}_{\mathrm{BSC}}$ for fixed e, during the decoding by $\mathrm{Dec}_{\mathrm{BSC}}$ is indeed $\pi_{s_\pi}(e)$).

By this discussion we have that:

$$\Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[S \in \textit{CANDIDATES} \text{ and } \bar{M}(S) = m] \geq \Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[A(S_{\mathrm{samp}}, S_\pi, C(Z)) = 1]. \tag{4}$$

On the other hand, the construction of Enc and Dec was set up so that, decoding is successful against additive channels. This is stated next.

Lemma 6.24. *For every* $e \in \{0,1\}^N$ *with* $\mathsf{wt}(e) \leq p$*,*

$$\Pr_{S_{\mathrm{samp}},S_\pi \leftarrow \{0,1\}^{\ell'}}[A(S_{\mathrm{samp}}, S_\pi, e) = 1] \geq 1 - N^{-c_{\mathrm{samp}}} - N^{-(c_{\mathrm{PRG}}-1)}.$$

Proof. We will now analyze what happens in this scenario, where $e \in \{0,1\}^N$ is fixed, and S_{samp}, S_π are uniform.

To prove the lemma, we make the following definition. For every $i \in [n]$, let $f(i) = \mathsf{wt}(e_i)$. By the properties of the Sampler, we have that when choosing $S_{\text{samp}} \leftarrow \{0,1\}^{\ell'}$, and taking $\{i_1, \ldots, i_{n_{\text{ctrl}}}\} = \text{Samp}(G(S_{\text{samp}}))$, the probability that

$$\frac{1}{n_{\text{ctrl}}} \cdot \sum_{j \in [n_{\text{ctrl}}]} f(i_j) > p + \frac{1}{\log n}$$

is at most $\frac{1}{N^{c_{\text{samp}}}} + \frac{1}{N^{c_{\text{PRG}}}}$. (This follows by the choice of parameters of the Sampler Samp in Figure 1 using Theorem 2.19, and we need to pay an additional additive error for the error of G, as we applied Samp on $G(S_{\text{samp}})$ and recall that the error of G is $\frac{1}{N^{c_{\text{PRG}}}}$).

Recall that $p < \frac{1}{4}$. Therefore, when this event occurs, by Markov's inequality the number of $j \in [n_{\text{ctrl}}]$ such that $f(i_j) > 2/3$ is at most $\frac{n_{\text{ctrl}}/4}{1/3} = n_{\text{ctrl}} \cdot \frac{3}{4}$. It indeed follows that, for at least $a = \frac{n_{\text{ctrl}}}{100}$ choices of j, $f(i_j) = \mathsf{wt}(e_{i_j}) \leq 1/3$, and the first condition in the definition of A is met.

We now show that the second condition is met w.h.p. over the choice of $S_\pi \leftarrow \{0,1\}^{\ell'}$. This is essentially the scenario for which Dec_{BSC} was designed (see Figure 1) of decoding from a pseudorandomly chosen permutation. More precisely, the weight of e is pN, and even if all the pN ones in e end up in e^I_{data} then the fraction of ones in the data part of e is at most:

$$\frac{pN}{N_{\text{data}}} = p \cdot \frac{1}{1 - \frac{n_{\text{ctrl}}}{n}} \leq p \cdot (1 + \frac{\epsilon}{10}) = p_{\text{BSC}},$$

where the last inequality holds for large enough N, by our choice that $n_{\text{ctrl}} = n^{0.9}$. In Figure 2, we have set up the code $(\text{Enc}_{\text{BSC}}, \text{Dec}_{\text{BSC}})$ to decode from $\text{Perm}^{\text{UniPerm}_{N_{\text{data}}}}_{p_{\text{BSC}}}$, and because π_{S_π} is pseudorandomly chosen (with error $N^{-c_{\text{PRG}}}$), rather than uniformly chosen, we get decoding error

$$2^{-\Omega(N^{0.09})} + N^{-c'_{\text{PRG}}} \leq N^{-(c_{\text{PRG}}-1)},$$

for sufficiently large N. Here, we make use of the fact that G fools circuits of size $N^{c_{\text{PRG}}}$, and we have chosen $c_{\text{PRG}} \geq d_{\text{BSC}} + c_0$, This gives that for every fixing s_{samp} of S_{samp}, Dec_{BSC} (which is of size $N^{d_{\text{BSC}}}$) does not distinguish between uniform and pseudorandomly chosen permutations.

Overall, by a union bound, we have that:

$$\Pr_{Z_{\text{samp}}, Z_\pi \leftarrow \{0,1\}^{\ell'}}[A(S_{\text{samp}}, S_\pi, e) = 1] \geq 1 - N^{-c_{\text{samp}}} - N^{-(c_{\text{PRG}}-1)},$$

as required. □

We would like to argue that Lemma 6.24 implies a bound in the scenario considered in (4) (where $e = C(Z)$, rather than e being a fixed vector that is independent of Z) . For this purpose, we observe that for every fixed values $s_{\text{samp}}, s_\pi \in \{0,1\}^{\ell'}$, A can be computed by a small circuit.

Claim 6.25. *The function $T(e) = A_{s_{\text{samp}}, s_\pi}(C(e))$ can be computed by a circuit of size N^{c+c_1} for a universal constant c_1.*

Proof. For fixed s_{samp}, s_π, such a circuit can be hardwired with $I = \text{Samp}(G(s_{\text{samp}}))$ and π_{s_π}. Therefore, after computing $C(e)$, it is easy to compute $e' = \pi_{s_\pi}(e^I_{\text{data}})$ in fixed polynomial time. This takes care of the first condition in the definition of A. For the second condition, we need to check whether $\text{Dec}_{\text{BSC}}(\text{Enc}_{\text{BSC}}(m) \oplus e') = m$.

Recall that in Figure 2 we took Dec_{BSC} to be the code from Theorem 4.2. Applying Dec_{BSC} directly would take time $N^{d_{\text{BSC}}}$ (which we cannot afford). Fortunately, in Theorem 4.2 (3rd item) we have set things

up so that once we compute e', checking whether $\mathrm{Dec}_{\mathrm{BSC}}(\mathrm{Enc}_{\mathrm{BSC}}(m) \oplus e') = m$ can be done by a circuit of fixed polynomial size. Overall, we indeed obtain that $T(e)$ can be computed by a circuit of size N^{c+c_1} for some universal constant c_1. □

We will now prove that the encoding of a message m is pseudorandom, even when fixing s_π and s_{samp}.

Lemma 6.26 (Pseudorandomness of encoding)**.** *For every message $m \in \{0,1\}^{RN}$, $s_{\mathrm{samp}} \in \{0,1\}^{\ell'}$, $s_\pi \in \{0,1\}^{\ell'}$, and $r_0 \in \{0,1\}^{N^{d_{\mathrm{ssnm}}}}$, let $V = \mathrm{Enc}(m; (s_\pi, s_{samp}, S_{PRG}, r_0, R_1, \cdots, R_{n_{ctrl}}))$ be a random variable (defined over the probability space where $S_{PRG}, R_1, \cdots, R_{n_{ctrl}}$ are chosen uniformly and independently). V is $N^{-(c_{\mathrm{pr}}-3)}$-pseudorandom for circuits of size $N^{c_{\mathrm{pr}}-3}$.*

Proof. We assume for contradiction that there exists a circuit D of size $N^{c_{\mathrm{pr}}-3}$

$$|\Pr[D(V)=1] - \Pr[D(U_N)=1]| > N^{-(c_{\mathrm{pr}}-3)}.$$

We will prove that one of the following holds:

- There exists a size $N^{c_{\mathrm{pr}}}$ circuit $C : \{0,1\}^{N_{\mathrm{data}}} \to \{0,1\}$ such that:
$$|\Pr[C(G(S_{\mathrm{PRG}}))=1] - \Pr[C(U_{N_{\mathrm{data}}})=1]| > N^{-c_{\mathrm{pr}}}.$$
- There exists $z' \in \{0,1\}^\ell$ and a size $N^{c_{\mathrm{pr}}}$ circuit $C : \{0,1\}^b \to \{0,1\}$, such that:
$$|\Pr[C(\mathrm{Enc}_{\mathrm{pr}}(z', U_d))=1] - \Pr[C(U_b)=1]| > N^{-c_{\mathrm{pr}}}.$$

This suffices, as the lemma follows by the pseudorandomness properties of G and $\mathrm{Enc}_{\mathrm{pr}}$.

We now prove that one of the two items hold. We partition V into $V = (V_{\mathrm{data}}, V_{\mathrm{ctrl}})^{\mathrm{Samp}(G(s_{\mathrm{samp}}))}$. We have that D distinguishes $V = (V_{\mathrm{data}}, V_{\mathrm{ctrl}})$ from $U_N = (U_{\mathrm{data}}, U_{\mathrm{ctrl}})$ with probability greater than $N^{-(c_{\mathrm{pr}}-3)}$, we do a hybrid argument and consider the hybrid distribution $H = (V_{\mathrm{data}}, U_{\mathrm{ctrl}})$. It follows that:

- Either D distinguishes H from U_N with probability $N^{-(c_{\mathrm{pr}}-3)}/2$,
- or, D distinguishes H from V with probability $N^{-(c_{\mathrm{pr}}-3)}/2$.

In the first case, we have that V_{data} and U_{ctrl} are independent, and an averaging argument gives that there exists a fixed value v'_{ctrl}, such that D distinguishes $(U_{\mathrm{data}}, v'_{\mathrm{ctrl}})$ from $(V_{\mathrm{data}}, v'_{\mathrm{ctrl}})$ with probability $N^{-(c_{\mathrm{pr}}-3)}/2$. This gives that there exists a size $N^{c_{\mathrm{pr}}-3}$ circuit $C : \{0,1\}^{N_{\mathrm{data}}} \to \{0,1\}$ such that the first item holds.

In the second case, we have that m and s_π are fixed and therefore the string $y = \pi_{s_\pi}^{-1}(\mathrm{Enc}_{\mathrm{BSC}}(m))$ used in the encoding algorithm is also fixed. The encoding algorithm computes the data part by xoring y with $G(S_{\mathrm{PRG}})$ and therefore $V_{\mathrm{data}} = G(S_{\mathrm{PRG}}) \oplus y$. By an averaging argument, there exists a fixing s'_{PRG} such that D distinguishes $((G(s'_{\mathrm{PRG}}) \oplus y), U_{\mathrm{ctrl}})$ from $(((G(s'_{\mathrm{PRG}}) \oplus y), V_{\mathrm{ctrl}}) | S_{\mathrm{PRG}} = s'_{\mathrm{PRG}})$ with probability $N^{-(c_{\mathrm{pr}}-3)}/2$.

We get that there exists a size $N^{c_{\mathrm{pr}}-3}$ circuit $C' : \{0,1\}^{n_{\mathrm{ctrl}} \cdot d} \to \{0,1\}$ such that D' distinguishes U_{ctrl} from $V'_{\mathrm{ctrl}} = (V_{\mathrm{ctrl}} | S_{PRG} = s'_{\mathrm{PRG}})$.

In the latter distribution, the three components of the control string $S = (S_{\mathrm{samp}}, S_\pi, S_{\mathrm{PRG}})$ are fixed. This means that when preparing the control part, $o = \mathrm{Enc}_{\mathrm{ssnm}}(s, r_0)$ and $w = \mathrm{Enc}_{\mathrm{RS}}(o)$ are fixed, we now recall that the encoding procedure prepares the j'th block of the control part c_{ctrl}, by $\mathrm{Enc}_{\mathrm{pr}}((i_j, w_j), r_j)$.

Having fixed $S_{\mathrm{PRG}} = s'_{\mathrm{PRG}}$ the only random variables that remain unfixed in V'_{ctrl} are $R_1, \ldots, R_{n_{\mathrm{ctrl}}}$. This means that there exist fixed strings $x_1, \ldots, x_{n_{\mathrm{ctrl}}} \in \{0,1\}^{2\log N}$ such that $(V'_{\mathrm{ctrl}})_j = \mathrm{Enc}_{\mathrm{ctrl}}(x_j, R_j)$ and in particular, the n_{ctrl} blocks are independent. We have that D' distinguishes V'_{ctrl} from U_{ctrl} with probability $N^{-(c_{\mathrm{pr}}-3)}/2$, and by a standard hybrid argument, there exists a circuit C of size $N^{c_{\mathrm{pr}}-3}$ such that C distinguishes $(V'_{\mathrm{ctrl}})_j = \mathrm{Enc}_{\mathrm{ctrl}}(x_j, R_j)$ from uniform with probability $n_{\mathrm{ctrl}} \cdot \frac{N^{-(c_{\mathrm{pr}}-3)}}{2} \geq N^{-c_{\mathrm{pr}}}$ and the second item follows. □

By the lemma, the distribution $Z \leftarrow \mathrm{expr}^{\mathrm{ed}}(m, C)$ conditioned on the event $\{S_{\mathrm{samp}} = s_{\mathrm{samp}}, S_\pi = s_\pi\}$ is $N^{-(c_{\mathrm{pr}}-3)}$-pseudorandom against circuits of size $N^{c_{\mathrm{pr}}-3}$. We have chosen $c_{\mathrm{pr}} = c + c_0$, and by taking c_0 to be sufficiently large, we can guarantee that the circuit that computes $T(e) = A_{s_{\mathrm{samp}}, s_\pi}(C(e))$ is of size $N^{c_{\mathrm{pr}}-3}$. It follows that for every fixed values $s_{\mathrm{samp}}, s_\pi \in \{0,1\}^{\ell'}$:

$$\left| \Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[A_{s_{\mathrm{samp}}, s_\pi}(C(Z)) = 1 \mid S_{\mathrm{samp}} = s_{\mathrm{samp}}, S_\pi = s_\pi] - \Pr[A_{s_{\mathrm{samp}}, s_\pi}(C(U_N)) = 1] \right| \leq N^{-(c_{\mathrm{pr}}-3)}. \tag{5}$$

We are now ready to prove Lemma 6.4. Let:

$$P = \Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[S \in \textit{CANDIDATES} \text{ and } \bar{M}(S) = m].$$

$$\begin{aligned}
P &\geq \Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[A_{S_{\mathrm{samp}}, S_\pi}(C(Z)) = 1] \\
&= \mathbb{E}_{s_{\mathrm{samp}}, s_\pi \leftarrow \{0,1\}^{\ell'}} \left[\Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[A_{S_{\mathrm{samp}}, S_\pi}(C(Z)) = 1 \mid S_{\mathrm{samp}} = s_{\mathrm{samp}}, S_\pi = s_\pi] \right] \\
&\geq \mathbb{E}_{s_{\mathrm{samp}}, s_\pi \leftarrow \{0,1\}^{\ell'}} \left[\Pr_{\mathrm{expr}^{\mathrm{ed}}(m,C)}[A_{S_{\mathrm{samp}}, S_\pi}(C(U_n)) = 1 \mid S_{\mathrm{samp}} = s_{\mathrm{samp}}, S_\pi = s_\pi] - N^{-(c_{\mathrm{pr}}-3)} \right] \\
&= \mathbb{E}_{s_{\mathrm{samp}}, s_\pi \leftarrow \{0,1\}^{\ell'}} \left[\Pr[A_{s_{\mathrm{samp}}, s_\pi}(C(U_n)) = 1] \right] - N^{-(c_{\mathrm{pr}}-3)} \\
&= \Pr_{S_{\mathrm{samp}}, S_\pi \leftarrow \{0,1\}^{\ell'}}[A(S_{\mathrm{samp}}, S_\pi, C(U_n)) = 1] - N^{-(c_{\mathrm{pr}}-3)} \\
&\geq 1 - N^{-c_{\mathrm{samp}}} - N^{c_{\mathrm{PRG}}-1} - N^{-(c_{\mathrm{pr}}-1)} - N^{-(c_{\mathrm{pr}}-3)} \\
&\geq 1 - \nu/2,
\end{aligned}$$

where the first line is by (4), the third line is by (5), the fifth line is by Lemma 6.24, and the final inequality is because $\nu = N^{-c}$ and $c_{\mathrm{samp}}, c_{\mathrm{PRG}}$ and c_{pr} were chosen to be at least $c + c_0$, and we can choose the constant c_0 to be sufficiently large.

7 Open problems

7.1 Open problems on stochastic codes for poly-size circuits.

In this paper we give the first explicit construction of stochastic codes for $\mathrm{Ckt}_p^{n^c}$, and achieve the optimal rate $R = 1 - H(p)$. This is achieved under the assumption that E is hard for exponential size nondeterministic circuits.

Relaxing the hardness assumption. Shaltiel and Silbak [SS21a] showed that the assumption that E is hard for poly-size circuits is necessary. Can we relax the assumption above? Can we construct similar codes using the assumption that E is hard for exponential size circuits?

Stochastic codes against channels that can simulate encoding and decoding. In our results, encoding and decoding algorithms run in time $\mathrm{poly}(n^c)$. That is, the channel cannot perform decoding. This is the case in all past work in this area. Is this necessary? Can we obtain stochastic codes against channels that can run the decoding algorithm, say under cryptographic assumptions?

7.2 Open problems on HTS

We believe that the notion of HTS introduced here is of independent interest. It will be interesting to explore additional applications.

Obtaining both small h and small output simultaneously. Each of the two Theorems 3.4 and Theorem 3.5 optimized one parameter at the cost of the other. Can we explicitly construct a (h, ρ) HTS $f : \{0,1\}^n \to \{0,1\}^{o(n)}$ for circuits of size n^c with $h = \mathrm{poly}(n^c)$ and $\rho = n^{-c}$? We remark that an easy application of the probabilistic method shows that such an HTS exists, and can have output length $c \log n + 1$. One avenue to achieve this is to improve the construction of high-error dispersers (Theorem 2.25) that we use as a component so that it works for lower min-entropy, while still having $D \ll n$.

We also remark that using our techniques (and a different choice of extractors for weakly recognizable distribution) it is possible to obtain a version of Theorem 3.5 (namely, an HTS with $h = 2^{\delta n}$ for any constant $\delta > 0$) in which the output length is linear in $c \log n$, under the assumption that E is hard for exponential size Σ_2-circuits.

Relaxing the hardness assumption. Is a hardness assumption against nondeterministic circuits necessary in order to obtain the HTS of Theorems 3.4 and 3.5?

References

[AASY15] B. Applebaum, S. Artemenko, R. Shaltiel, and G. Yang. Incompressible functions, relative-error extractors, and the power of nondeterministic reductions. In *30th Conference on Computational Complexity*, pages 582–600, 2015.

[AIKS16] S. Artemenko, R. Impagliazzo, V. Kabanets, and R. Shaltiel. Pseudorandomness when the odds are against you. In *31st Conference on Computational Complexity, CCC*, volume 50, pages 9:1–9:35, 2016.

[BD22] G. Blanc and D. Doron. New near-linear time decodable codes closer to the GV bound. *Electron. Colloquium Comput. Complex.*, TR22-027, 2022.

[BDL22] M. Ball, D. Dachman-Soled, and J. Loss. (nondeterministic) hardness vs. non-malleability. In *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference*, volume 13507, pages 148–177, 2022.

[BGP00] M. Bellare, O. Goldreich, and E. Petrank. Uniform generation of np-witnesses using an np-oracle. *Inf. Comput.*, 163(2):510–526, 2000.

[BKP18] N. Bitansky, Y. Tauman Kalai, and O. Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 671–684, 2018.

[BKS+10] B. Barak, G. Kindler, R. Shaltiel, B. Sudakov, and A. Wigderson. Simulating independence: New constructions of condensers, ramsey graphs, dispersers, and extractors. *J. ACM*, 57(4), 2010.

[BOV07] B. Barak, S. J. Ong, and S. P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007.

[BSS23] M. Ball, R. Shaltiel, and J. Silbak. Non-malleable codes with optimal rate for poly-size circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, (167), 2023.

[BV17] N. Bitansky and V. Vaikuntanathan. A note on perfect correctness by derandomization. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 10211, pages 592–606, 2017.

[CJL15] Z. Chen, S. Jaggi, and M. Langberg. A characterization of the capacity of online (causal) binary channels. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 287–296, 2015.

[CT22] L. Chen and R. Tell. When arthur has neither random coins nor time to spare: Superfast derandomization of proof systems. *Electron. Colloquium Comput. Complex.*, TR22-057, 2022.

[DJLS13] B. K. Dey, S. Jaggi, M. Langberg, and A. D. Sarwate. Upper bounds on the capacity of binary channels with causal adversaries. *IEEE Transactions on Information Theory*, 59(6):3753–3763, 2013.

[DMOZ22] D. Doron, D. Moshkovitz, J. Oh, and D. Zuckerman. Nearly optimal pseudorandomness from hardness. *J. ACM*, 69(6):43:1–43:55, 2022.

[DPW18] S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes. *J. ACM*, 65(4):20:1–20:32, 2018.

[Dru13] Andrew Drucker. Nondeterministic direct product reductions and the success probability of SAT solvers. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 736–745, 2013.

[EGZ22] Chattopadhyay E, J. Goodman, and D. Zuckerman. The space complexity of sampling. In *13th Innovations in Theoretical Computer Science Conference, ITCS*, volume 215, pages 40:1–40:23, 2022.

[FL97] U. Feige and C. Lund. On the hardness of computing the permanent of random matrices. *Computational Complexity*, 6(2):101–132, 1997.

[FMVW16] S. Faust, P. Mukherjee, D. Venturi, and D. Wichs. Efficient non-malleable codes and key derivation for poly-size tampering circuits. *IEEE Trans. Inf. Theory*, 62(12):7179–7194, 2016.

[For65] G. D. Forney. *Concatenated codes.* PhD thesis, Massachusetts Institute of Technology, 1965.

[GI05] V. Guruswami and P. Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, 2005.

[Gol97] O. Goldreich. A sample of samplers - a computational perspective on sampling (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, 4(20), 1997.

[GS99] V. Guruswami and M. Sudan. Improved decoding of reed-solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.

[GS16] V. Guruswami and A. Smith. Optimal rate code constructions for computationally simple channels. *Journal of the ACM (JACM)*, 63(4):35, 2016.

[GST03] Dan Gutfreund, Ronen Shaltiel, and Amnon Ta-Shma. Uniform hardness versus randomness tradeoffs for arthur-merlin games. *Computational Complexity*, 12(3-4):85–130, 2003.

[GUV07] V. Guruswami, C. Umans, and S. P. Vadhan. Unbalanced expanders and randomness extractors from parvaresh-vardy codes. In *CCC*, pages 96–108, 2007.

[GW02] O. Goldreich and A. Wigderson. Derandomization that is rarely wrong from short advice that is typically good. In *APPROX-RANDOM*, pages 209–223, 2002.

[HNY17] P. Hubácek, M. Naor, and E. Yogev. The journey from NP to TFNP hardness. In *8th Innovations in Theoretical Computer Science Conference, ITCS*, volume 67, pages 60:1–60:21, 2017.

[IW97] R. Impagliazzo and A. Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *STOC*, pages 220–229, 1997.

[JST21] F. G. Jeronimo, S. Srivastava, and M. Tulsiani. Near-linear time decoding of ta-shma's codes via splittable regularity. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1527–1536, 2021.

[JVV86] M. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.

[JW15] Z. Jafargholi and D. Wichs. Tamper detection and continuous non-malleable codes. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC*, volume 9014, pages 451–480, 2015.

[KSS19] S. Kopparty, R. Shaltiel, and J. Silbak. Quasilinear time list-decodable codes for space bounded channels. *To appear in the 60th Annual Symposium on Foundations of Computer Science (FOCS)*, 2019.

[KvM02] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.

[KvMS12] J. Kinne, D. van Melkebeek, and R. Shaltiel. Pseudorandom generators, typically-correct derandomization, and circuit lower bounds. *Computational Complexity*, 21(1):3–61, 2012.

[Lan04] M. Langberg. Private codes or succinct random codes that are (almost) perfect. In *45th Symposium on Foundations of Computer Science (FOCS 2004)*, pages 325–334, 2004.

[Lip94] R. J. Lipton. A new approach to information theory. In *11th Annual Symposium on Theoretical Aspects of Computer Science*, pages 699–708, 1994.

[LRVW03] C.J. Lu, O. Reingold, S. P. Vadhan, and A. Wigderson. Extractors: optimal up to constant factors. In *STOC*, pages 602–611. ACM, 2003.

[LZ19] F. Li and D. Zuckerman. Improved extractors for recognizable and algebraic sources. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, volume 145 of *LIPIcs*, pages 72:1–72:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[MPSW10] Silvio Micali, Chris Peikert, Madhu Sudan, and David A. Wilson. Optimal error correction for computationally bounded noise. *IEEE Trans. Information Theory*, 56(11):5673–5680, 2010.

[MRRW77] R. McEliece, E. Rodemich, H. Rumsey, and L. Welch. New upper bounds on the rate of a code via the delsarte-macwilliams inequalities. *IEEE Transactions on Information Theory*, 23(2):157–166, 1977.

[MV05] P. Bro Miltersen and N. V. Vinodchandran. Derandomizing arthur-merlin games using hitting sets. *Computational Complexity*, 14(3):256–279, 2005.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *JCSS: Journal of Computer and System Sciences*, 49, 1994.

[Sha09] R. Shaltiel. Weak derandomization of weak algorithms: explicit versions of yao's lemma. In *CCC*, 2009.

[Sip83] M. Sipser. A complexity theoretic approach to randomness. In *STOC*, pages 330–335, 1983.

[Smi07] A. D. Smith. Scrambling adversarial errors using few random bits, optimal information reconciliation, and better private codes. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 395–404, 2007.

[SS21a] R. Shaltiel and J. Silbak. Explicit list-decodable codes with optimal rate for computationally bounded channels. *Comput. Complex.*, 30(1):3, 2021.

[SS21b] R. Shaltiel and J. Silbak. Explicit uniquely decodable codes for space bounded channels that achieve list-decoding capacity. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1516–1526, 2021.

[SS22] R. Shaltiel and J. Silbak. Error correcting codes that achieve BSC capacity against channels that are poly-size circuits. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 13–23, 2022.

[Sto83] L. J. Stockmeyer. The complexity of approximate counting. In *STOC*, pages 118–126, 1983.

[STV01] M. Sudan, L. Trevisan, and S. P. Vadhan. Pseudorandom generators without the xor lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[SU05] R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *J. ACM*, 52(2):172–216, 2005.

[SU06] R. Shaltiel and C. Umans. Pseudorandomness for approximate counting and sampling. *Computational Complexity*, 15(4):298–341, 2006.

[SU09] R. Shaltiel and C. Umans. Low-end uniform hardness versus randomness tradeoffs for am. *SIAM J. Comput.*, 39(3):1006–1037, 2009.

[Sud97] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13, 1997.

[TS17] A. Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 238–251, 2017.

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

[TZ04] A. Ta-Shma and D. Zuckerman. Extractor codes. *IEEE Trans. Inf. Theory*, 50(12):3015–3025, 2004.

[Vio12] E. Viola. The complexity of distributions. *SIAM J. Comput.*, 41(1):191–218, 2012.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Struct. Algorithms*, 11(4):345–367, 1997.

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

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il