Electronic Colloquium on Computational Complexity, Revision 3 of Report No. 7 (2011)

Pseudorandom Generators with Long Stretch and Low Locality from Random Local One-Way Functions

Benny Applebaum*

April 5, 2012

Abstract

We continue the study of *locally-computable* pseudorandom generators (PRG) $G : \{0,1\}^n \to \{0,1\}^m$ that each of their outputs depend on a small number of d input bits. While it is known that such generators are likely to exist for the case of small sub-linear stretch $m = n + n^{1-\delta}$, it is less clear whether achieving larger stretch such as $m = n + \Omega(n)$, or even $m = n^{1+\delta}$ is possible. The existence of such PRGs, which was posed as an open question in previous works (e.g., [Cryan and Miltersen, MFCS 2001], [Mossel, Shpilka and Trevisan, FOCS 2003], and [Applebaum, Ishai and Kushilevitz, FOCS 2004]), has recently gained an additional motivation due to several interesting applications.

We make progress towards resolving this question by obtaining several local constructions based on the one-wayness of "random" local functions – a variant of an assumption made by Goldreich (ECCC 2000). Specifically, we construct collections of PRGs with the following parameters:

- Linear stretch $m = n + \Omega(n)$ and constant locality $d = O(1)$.
- Polynomial stretch $m = n^{1+\delta}$ and *any* (arbitrarily slowly growing) super-constant locality $d = \omega(1)$, e.g., $\log^\star n$.
- Polynomial stretch $m = n^{1+\delta}$, constant locality $d = O(1)$, and inverse polynomial distinguishing advantage (as opposed to the standard case of $n^{-\omega(1)}$).

Our constructions match the parameters achieved by previous "ad-hoc" candidates, and are the first to do this under a one-wayness assumption. At the core of our results lies a new search-to-decision reduction for random local functions. This reduction also shows that some of the previous PRG candidates can be based on one-wayness assumptions. Altogether, our results fortify the existence of local PRGs of long stretch.

As an additional contribution, we show that our constructions give rise to strong inapproximability results for the densest-subgraph problem in d-uniform hypergraphs for constant d. This allows us to improve the previous bounds of Feige (STOC 2002) and Khot (FOCS 2004) from constant inapproximability factor to n^ε-inapproximability, at the expense of relying on stronger assumptions.

*School of Electrical Engineering, Tel Aviv University, bennyap@post.tau.ac.il. Work partially done while a postdoc at the Weizmann Institute of Science. Supported by Koshland and Alon Fellowship, ISF grant 1155/11, and by the Check Point Institute for Information Security.

ISSN 1433-8092

Contents

1 Introduction

Pseudorandomness is one of the most fundamental concepts in cryptography and complexity theory. Formally, a *pseudorandom generator* (PRG) $f : \{0,1\}^n \to \{0,1\}^m$ maps a random n-bit seed $x = (x_1, \ldots, x_n)$ into a longer pseudorandom string $y = (y_1, \ldots, y_m)$ such that no polynomial-time adversary can distinguish y from a truly random m-bit string with *distinguishing advantage* better than ε (by default, ε is negligible $n^{-\omega(1)}$). The notion of PRGs is highly non-trivial and, in some sense, very fragile: any noticeable regularity or correlation in the output y immediately violates pseudorandomness. This is, for example, fundamentally stronger than the notion of *one-way functions* which asserts that f is hard to invert on typical images. For this reason, a considerable research effort has been made to put PRGs on more solid ground at the form of one-wayness assumptions [40, 33, 23, 16, 38, 34, 27]. In this paper we aim to establish similar firm foundations for the case of locally-computable PRGs with large stretch.

Locally-computable PRGs offer an extreme level of parallelism. In such PRGs each output y_i depends only on a small number of d input bits. Our goal is to gain many pseudorandom bits, i.e., maximize the *stretch* $m - n$, while keeping the *locality* d as small as possible. Ultimately, d is a constant that does not grow with the total input length or the level of security; In this case, the computation can be carried in *constant*-parallel time, which is captured by the complexity class $\mathbf{NC^0}$.

This strong efficiency requirement seems hard to get as, at least intuitively, such form of locality may lead to algorithmic attacks. Still, it was shown in [6] that, for the regime of sub-linear stretch $m = n + n^{1-\delta}$, PRGs with constant locality exist under standard cryptographic assumptions (e.g., the hardness of factoring, discrete log, or lattice problems, or more generally, the existence of log-space computable one-way functions [26]). Unfortunately, it is unknown how to extend this result to linear stretch ($m > (1+\delta)n$) let alone polynomial stretch ($m > n^{1+\delta}$).[1] This raises the following question which was posed by several previous works (e.g., [14, 36, 6, 7, 29]):

> How long can be the stretch of a pseudorandom generator with locality d? How large should d be to achieve linear or polynomial stretch?

Local PRGs with linear and polynomial stretch are qualitatively different than ones with sub-linear stretch. For example, as shown in [29], such PRGs allow to improve the *sequential complexity* of cryptography: a linear-stretch PRG (hereafter abbreviated by LPRG) with constant locality would lead to implementations of primitives (e.g., public-key encryption, commitment schemes) with *constant* computational overhead, and a polynomial-stretch PRG (hereafter abbreviated by PPRG) with constant locality would lead to secure computation protocols with *constant* computational overhead – a fascinating possibility which is not known to hold under any other cryptographic assumption. From a practical point of view, large stretch PRGs with low locality give rise to highly efficient stream-ciphers that can be implemented by fast parallel hardware.

Furthermore, this difference between sub-linear and linear/polynomial stretch turns out to have applications also outside cryptography. As shown in [7], local PRGs with large stretch lead to strong (average-case) inapproximability results for constraint satisfaction problems such as Max3SAT under a natural distribution: LPRGs with constant locality rule out the existence of a PTAS, whereas

[1] In fact, for the special case of 4-local functions, there is a provable separation: such functions can compute sub-linear PRGs [6] but *cannot* compute polynomial-stretch PRGs [14, 36]. For larger locality $d \geq 5$, the best upper-bound on m is $n^{d/2}$ [36].

PPRGs yield *tight* bounds that match the upper-bounds achieved by the simple "random assignment" algorithm.

Although local LPRGs and PPRGs are extremely useful, their existence is not well established. Even when $d = O(\log n)$ it is unknown how to construct LPRGs, let alone PPRGs, whose security can be reduced to some other weaker cryptographic assumption, e.g., a one-wayness assumption.[2] This state of affairs have lead to a more direct approach. Rather than trying to obtain PPRGs or LPRGs based on one-wayness assumptions, several researchers suggested concrete candidates for LPRGs with constant locality and negligible distinguishing advantage [2, 7] and PPRGs with constant locality and inverse polynomial distinguishing advantage [36, 4]. (A detailed account of this line of works is given in Section 1.3.) All of these candidates are essentially based on what we call *random local functions*, i.e., each output bit is computed by applying some fixed d-local predicate Q to a randomly chosen d-size subset of the input bits. Formally, this can be viewed as selecting a random member from a collection $\mathcal{F}_{Q,n,m}$ of d-local functions where each member $f_{G,Q} : \{0,1\}^n \to \{0,1\}^m$ is specified by a d-uniform hypergraph G with n nodes and m hyperedges, and the i-th output of $f_{G,Q}$ is computed by applying the predicate Q to the d inputs that are indexed by the i-th hyperedge.

Remark 1.1 (Collection vs. Single function). *The above construction gives rise to a* collection *of local PRGs, where a poly-time preprocessing is used to publicly pick (once and for all) a random instance f from the collection. The adversary is given the description of the chosen function, and its distinguishing advantage is measured with respect to a random seed and a random member of the family. (See Section 3 and [22, Sec. 2.4.2], [28] for formal definitions). The use of collections is standard in the context of parallel cryptography (e.g., in number-theoretic constructions preprocessing is used to set-up the group or choose a random composite number [37, 15, 25]) and has no effect on the applications, hence we adopt it as our default setting. (See [6, Appendix A] for detailed discussion.) In our case the preprocessing typically has a parallel implementation in* $\mathbf{AC^0}$.

1.1 Our constructions

The gap between the rich applications of large-stretch PRGs in $\mathbf{NC^0}$ and the lack of provable constructions is highly unsatisfactory. Our goal in this paper is to remedy the situation by replacing the ad-hoc candidates with constructions whose security can be based on a more conservative assumption. Specifically, our results essentially show that in order to construct local PRGs with output length m it suffices to assume that random local functions are *one-way*; namely, that it is hard to *invert* a random member of the collection $\mathcal{F}_{Q,n,m'}$ for m' of the same magnitude as m. This one-wayness assumption, originally made by Goldreich [21] and further established in [39, 3, 13, 35, 10, 4, 11], is significantly weaker (i.e., more plausible) than the corresponding pseudorandomness assumption (see Section 1.3 for discussion). Let us now state our main results, starting with the case of linear stretch.

Theorem 1.2 (LPRG in $\mathbf{NC^0}$). *If the d-local collection $\mathcal{F}_{Q,n,m}$ is one-way for $m > \Omega_d(n)$, then there exists a collection of LPRGs with constant locality and negligible distinguishing advantage.*

[2] To the best of our knowledge, even the class $\mathbf{AC^0}$ (which is strictly stronger than $O(\log n)$-local functions) does not contain any (provably-secure) large-stretch PRG, and only in $\mathbf{TC^0}$, which is strictly more powerful than $\mathbf{AC^0}$, such constructions are known to exist [37].

Theorem 1.2 is applicable for every choice of predicate Q, and it provides the first construction of an LPRG in $\mathbf{NC^0}$ based on a one-wayness assumption. Moving to the case of polynomial stretch, we say that the predicate Q is *sensitive* if *some* of its coordinates i has full influence (i.e., flipping the value of the i-th variable always changes the output of Q).

Theorem 1.3 (weak-PPRG in $\mathbf{NC^0}$). *Let $\delta > 0$ be an arbitrary small constant, $m > n^{1+\delta}$ and Q be a d-local sensitive predicate. If the collection $\mathcal{F}_{Q,n,m}$ is one-way then:*

- *For every constant b, there exists a weak collection of PPRGs of output length n^b and distinguishing advantage at most $1/n^b$ with constant locality $d' = d'(d, b)$.*
- *Alternatively, it is possible to achieve polynomial stretch n^b and* negligible *distinguishing advantage $n^{-\omega(1)}$ at the expense of letting the locality $d' = \omega(1)$ be an arbitrarily slowly increasing function of n, e.g., $\log^*(n)$.*

The PPRGs constructions of Theorem 1.3 are the first to achieve constant locality and inverse-polynomial distinguishing advantage (resp., super-constant locality and negligible distinguishing advantage) under a one-wayness assumption. These parameters also match the ones achieved by known heuristic candidates for PPRGs.[3] We mention that there are sensitive predicates (e.g., $Q(x_1, x_2, x_3, x_4, x_5) = (x_1 \wedge x_2) \oplus x_3 \oplus x_4 \oplus x_5$) for which $\mathcal{F}_{Q,n,m=n^{1+\delta}}$ seems to be one-way [36, 13, 5].

Following the previous theorems, one may ask whether it is possible to show that $\mathcal{F}_{Q,n,m}$ *itself* is pseudorandom. We show that this is indeed the case:

Theorem 1.4 (Random $\mathbf{NC^0}$ function is PPRG). *The collection $\mathcal{F}_{Q,n,n^a}$ is a weak-PPRG with distinguishing advantage at most n^{-b}, assuming that the collection $\mathcal{F}_{Q,n,n^{3a+2b}}$ is one-way and that Q is sensitive.*[4]

As a corollary, we reduce the pseudorandomness of some of the previous ad-hoc constructions to a one-wayness assumption. We view Theorem 1.4 as one of the main conceptual contribution of this work. The ensemble $\mathcal{F}_{Q,n,m}$ is highly interesting for its own sake as it generalizes an important and well-studied family of random constraint satisfaction problems (e.g., random planted 3-SAT [20, 12, 1]). Indeed, the problem of inverting a random member of the ensemble $\mathcal{F}_{Q,n,m}$ boils down to solving a system of m random d-local (non-linear) equations of the form $y_i = Q(x_{i_1}, \ldots, x_{i_d})$ with a planted solution x. Theorem 1.4 yields an average-case search-to-decision reduction for this problem. Combined with the results of [7], it follows that any non-trivial approximation of the value of the system of equation allows to fully recover the planted solution.

1.2 New application: Hardness of the Densest-Subgraph Problem

We use Theorem 1.4 to derive new inapproximability results. This continues the line of research started by Feige [18] in which inapproximability follows from average-case hardness. For a d-uniform hypergraph G, we say that a set of nodes S *contains* an edge $e = (v_1, \ldots, v_d)$ if all the endpoints

[3] In the heuristic constructions the dependencies graph G should satisfy non-trivial expansion properties [7], but when $m = n^{1+\Omega(1)}$ and $d = O(1)$ it is unknown how to efficiently sample such a good expander with negligible failure probability.

[4] Some assumption on the predicate is needed as it seems likely that for some unbalanced predicate Q one-wayness may hold, whereas in this case the ensemble $\mathcal{F}_{Q,n,m}$ cannot be pseudorandom. Still, we can show that one-wayness with respect to a general predicate implies that the ensemble has large *pseudoentropy* in the sense of [26].

of e are in S, i.e., $v_1, \ldots, v_d \in S$. In the following think of d as a constant, $n < m < \text{poly}(n)$, and $p \in (0,1)$. In the p Densest-Sub-hypergraph Problem ($p - \mathsf{DSH}$) we are given a d-uniform hypergraph G with n nodes and m edges (hereafter referred to as an (n,m,d) graph) and should distinguish between:

- **No case ("Random").** Every set S of nodes of density p (i.e., size pn) in G contains at most $p^d(1+o(1))$ fraction of the edges.
- **Yes case ("Planted").** There exists a set S of nodes of density p in G that contains at least $p^{d-1}(1-o(1))$ fraction of the edges.

That is, the No instance corresponds to a random-looking graph, while the Yes instance contains a planted dense subgraph. The parameter p controls both the approximation ratio and the gap-location (i.e., size of the dense subgraph). This formulation of $p - \mathsf{DSH}$ was explicitly presented by Khot [32] (under the term "Quasi-random PCP"), and was implicit in the work of Feige [18]. These works showed that for some constant d, the problem is hard for $p = \frac{1}{2}$, assuming that **NP** cannot be solved in probabilistic sub-exponential time. The constant p can be improved by taking graph products, however, this increases the degree d. Hence, for a constant degree, the best known inapproximability ratio was constant. This is in sharp contrast with the best known algorithm [9] which achieves approximation ratio no better than $\Theta(n^{1/4})$ even in the simple case where $d = 2$ (which corresponds to the well known "Densest Subgraph problem" [19]). Our next theorem partially closes this gap:

Theorem 1.5. *Let d be a constant, Q be a d-ary predicate and $m \geq n^{1+\delta}$ where $\delta > 0$ is a constant. If $\mathcal{F}_{Q,n,m}$ is $\frac{1}{\log n}$-pseudorandom, then for every $n^{-\frac{\delta}{d+2}} \leq p \leq \frac{1}{2}$ the p-Densest-Subhypergraph problem is intractable with respect to d-uniform hypergraphs.*

By taking $p = \frac{1}{2}$, we obtain the same parameters as in [18, 32]. We can obtain much stronger inapproximability ratio of, say, $p = n^{-1/(d+2)}$ for a *fixed* locality d, assuming that $\mathcal{F}_{Q,n,n^2}$ is $\frac{1}{\log n}$-pseudorandom. As shown in Thm. 1.4, the latter assumption follows from the one-wayness of $\mathcal{F}_{n,m',Q}$ for sufficiently large polynomial $m'(n)$.

Interestingly, Theorem 1.5 yields average-case hardness with respect to a "planted distribution". Namely, we show that it is hard to distinguish a random graph (which is likely to be a "No" instance) from a random graph in which a dense random subgraph is planted. (Jumping ahead, the planted subgraph essentially encodes a preimage of the pseudorandom generator.) We also mention that such distributions are used by Arora et al. [8] to show that financial derivatives can be fraudulently mispriced without detection.

1.3 Discussion and Previous Works

1.3.1 Pseudorandomness of $\mathcal{F}_{Q,n,m}$

In [36] it was shown that, for proper choices of the predicate Q and the graph G, the function $f_{G,Q}$ fools linear-tests over $\mathbb{F}_2$. A similar result holds for a *random* member in $\mathcal{F}_{Q,n,m=n^{1+\varepsilon}}$ as long as Q is sufficiently "good" [4]. Recently, a full classification of good predicates (at the form of a dichotomy theorem) was established in [5].

Alechnovich [2] conjectured that the collection $\mathcal{F}_{Q=\oplus_p,n,m=\Theta(n)}$ is pseudorandom where $\oplus_p$ is a *randomized* predicate which computes $z_1 \oplus z_2 \oplus z_3$ and with some small probability $p < \frac{1}{2}$ flips the result. Although this construction does not lead directly to a local PRG (due to the use of noise), it was shown in [7] that it can be derandomized and transformed into an $\mathbf{NC^0}$ construction with linear stretch. (The restriction to linear stretch holds even if one strengthen Alekhnovich's assumption to $m = \text{poly}(n)$.)

Recently, it is was shown that, for the special case of the noisy-linear predicate $\oplus_p$, one-wayness implies weak pseudorandomness [4]. Specifically, if $\mathcal{F}_{\oplus_p,n,m=O(n\log n)}$ is one-way then $\mathcal{F}_{\oplus_p,n,m=O(n)}$ is pseudorandom with distinguishing advantage $\varepsilon = 1/n$.[5] Our work is highly inspired by this result. From a technical point of view, many of the ideas used in [4] heavily rely on the linear structure of $\oplus_p$, and so part of the challenge in establishing our reductions is to find analogues which work in the general case of arbitrary predicates. (See Section 2 for an overview of our proofs.) As a byproduct, our techniques provide a simpler proof for the case of $\oplus_p$ with slightly better parameters.

1.3.2 One-wayness of $\mathcal{F}_{Q,n,m}$

The ensemble $\mathcal{F}_{Q,n,m}$ was explicitly presented by Goldreich [21] who conjectured one-wayness for the case of $m = n$ and essentially every non-trivial predicate (e.g., non-linear and non-degenerate). In [21, 3, 13, 35, 17, 30] it is shown that a large class of algorithms (including ones that capture DPLL-based heuristics) fail to invert $\mathcal{F}_{Q,n,m}$ in polynomial-time. These results are further supported by the experimental study of [39, 13] which employs, among other attacks, SAT-solvers.

Recently, a strong self-amplification theorem was proved in [11] showing that for $m = \Omega_d(n)$ if $\mathcal{F}_{Q,n,m}$ is hard-to-invert over tiny (sub-exponential small) fraction of the inputs with respect to sub-exponential time algorithm, then the same ensemble is actually hard-to-invert over almost all inputs (with respect to sub-exponential time algorithms). In addition, the one-wayness of $\mathcal{F}_{Q,n,m}$ is actively challenged by the theoretical and practical algorithmic study of random constraint satisfaction problems (e.g., Random 3-SAT, see [20, 12, 1] for surveys). The fact that this research falls short of inverting $\mathcal{F}_{Q,n,m}$ provides a good evidence to its security.[6]

To summarize, when m is linear, i.e., $m = cn$ for arbitrary constant $c > 1$, it is unknown how to invert the function (with respect to a general predicate) in complexity smaller than $2^{\Omega(n)}$. It also seems reasonable to assume that for every constant $c > 1$ there exists a sufficiently large locality d and a predicate Q for which $\mathcal{F}_{Q,n,n^c}$ cannot be inverted in polynomial time.

1.3.3 One-wayness vs. pseudorandomness

The above works indicate that one-wayness is much more solid than pseudorandomness. We wish to emphasize that this is true even with respect to heuristic constructions. Indeed pseudorandomness is quite fragile, especially with low locality, as in this case even the task of avoiding simple regularities in the output is highly challenging.[7] In contrast, it seems much easier to find a "reasonable" candidate one-way functions (i.e., one that resists all basic/known attacks). Moreover, it is not hard to come up with examples for local functions whose one-wayness may be plausible but they

[5] We believe that this result can be combined with the techniques of [7] to yield a local *weak* LPRG with $\varepsilon = 1/\text{poly}(n)$ based on the one-wayness of $\mathcal{F}_{\oplus_p,n,m=O(n\log n)}$. However, it falls short of providing LPRG (i.e., with standard security) or weak PPRG.

[6] This research have lead to non-trivial algorithms which allow to invert $\mathcal{F}_{Q,n,m=\Omega_d(n)}$ when the predicate is correlated with one or two of its inputs [10], however, these attacks do not generalize to other predicates.

[7] This even led to the belief that weak non-cryptographic forms of pseudorandomness, e.g., ε-bias, cannot be achieved [14], which was refuted in a non-trivial way by [36].

fail to be pseudorandom (e.g., if the graph happen to have the same hyperedge twice, or if the predicate is unbalanced). The proof of our main theorems show that in this case, despite the existence of non-trivial regularities in the outputs, random local one-way functions achieve some form of pseudoentropy (i.e., weak unpredictability).

1.3.4 More on DSH

DSH is a natural generalization of the notoriously hard Densest k-Subgraph (DSG) problem (e.g., [19]) whose exact approximation ratio is an important open question. The best known algorithm achieves $O(n^{1/4})$-approximation [9], while known hardness results only rule out PTAS [32]. Naturally, DSH, which deals with hypergraphs, only seems harder. DSH has also a special role as a starting point for many other inapproximability results for problems like graph min-bisection, bipartite clique, and DSG itself [18, 32]. Recently, it was shown how to use the average-case hardness of DSH to plant a trapdoor in $\mathcal{F}_{Q,n,m}$, and obtain public-key encryption schemes [4]. This raises the exciting possibility that, for random local functions, it may be possible to go all the way from one-wayness to public-key cryptography: first assume that $\mathcal{F}_{Q,n,m}$ is one-way, then use Thm. 1.4 to argue that this collection is actually pseudorandom, then employ Thm. 1.5 to argue that DSH is hard over a planted distribution, and finally, use [4] to obtain a public-key cryptosystem. Unfortunately, the parameters given in Thm. 1.5 do not match the ones needed in [4]; still we consider the above approach as an interesting research direction.

2 Our Techniques

To illustrate some of our techniques, let us outline the proof of our main constructions starting with the case of polynomial stretch PRGs (Thms. 1.3 and 1.4).

2.1 Constructing Weak-PPRGs

Conceptually, we reduce pseudorandomness to one-wayness via the following approach: Suppose that we have an adversary which breaks the pseudorandomness properties of the function $f_{G,Q}(x)$ with respect to a random graph G. Then we can collect information about x, and eventually invert the function, by invoking the adversary multiple times with respect to many *different* graphs $G_1, \dots, G_t$ which are all close variants of the original G. Details follow.

2.1.1 The basic procedure

Due to the known reduction from pseudorandomness to unpredictability (aka Yao's theorem [40]), it suffices to reduce the task of inverting $\mathcal{F}_{Q,n,m}$ to the task of predicting the next bit in the output of $\mathcal{F}_{Q,n,k}$ with probability $\frac{1}{2}+\varepsilon$. Let us see how a prediction algorithm can be used to recover some information on the input.

Assume that the first input of Q has full influence, and that we are given an $(\frac{1}{2}+\varepsilon)$-predictor $\mathbf{P}$. This predictor is given a random (n,k,d) graph G, whose hyperedges are labeled by the string $y = f_{G,Q}(x)$, and it should predict the label $y_k = Q(x_S)$ of the last hyperedge $S = (i_1, \dots, i_d)$. (We can always reshuffle the edges of the graph to ensure last-bit predictability.) Given such a pair (G, y), let us replace the first entry i_1 of S with a random index $\ell \in [n]$ (hereafter referred to as "pivot"), and then invoke $\mathbf{P}$ on the modified pair. If the predictor succeeds and outputs $Q(x_{S'})$, then, by comparing this value to y_k, we get to learn whether the input bits x_ℓ and x_{i_1} are equal. Since the predictor may err, we can treat this piece of information as a single 2-LIN noisy equation of the form $x_\ell \oplus x_{i_1} = b$ where $b \in \{0,1\}$.

2.1.2 Collecting many 2-LIN equations

In order to recover x, we would like to collect many such equations and then solve a Max-2-LIN problem. To this end, we may partition the graph G and the output string y to many blocks $(G^{(i)}, y^{(i)})$ of size k each, and then apply the above procedure to each block separately. This gives us a highly-noisy system of 2-LIN equations with a very large noise rate of $\frac{1}{2}-\varepsilon$ where $\varepsilon < 1/k < 1/n$ corresponds to the quality of prediction. (This value of ε is dictated by Yao's theorem, which cannot be used with larger ε.)

One may try to "purify" the noise by collecting many (say n^2/ε^2) equations, and correcting the RHS via majority vote, however, this approach is doomed to fail as the noise is not random, and can be chosen *arbitrarily* by the adversary in a way that depends on the equations. To see this, consider the trivial predictor which predicts well only when the output depends on x_1, and otherwise outputs a random guess. This predictor satisfies our condition (i.e., its prediction advantage is $1/n$) but it seems to be totally useless since it works only for equations which involves x_1. As a result, repetition will not decrease the noise.

2.1.3 Partial rerandomization

We fix this problem by randomizing the blocks $(G^{(i)}, y^{(i)})$. Specifically, we will permute the nodes of each $G^{(i)}$ under a random permutation $\pi^{(i)} : [n] \to [n]$, and invoke our basic procedure on the pairs $(\pi^{(i)}(G^{(i)}), y^{(j)})$. This is essentially equivalent to shuffling the coordinates of x. Furthermore, this transformation does not affect the distribution of the graphs since edges were chosen uniformly at random any way. As a result, the noise (i.e., the event that $\mathbf{P}$ errs) becomes independent of the variables that participates in the equations, and the distribution of the prediction errors is "flattened" over all possible hyperedges. This transformation also yields a partial form of "random-self-reducibility": the input x is mapped to a random input of the same Hamming weight.

To show that the basic procedure succeeds well in each of the blocks, we would like to argue that the resulting pairs $(H^{(i)}, f_{H^{(i)}}(x^{(i)}))$ are uniformly and independently distributed, where $H^{(i)}$ is the permuted graph $\pi^{(i)}(G^{(i)})$ and $x^{(i)}$ is the permuted string $\pi^{(i)}(x)$. This is not true as all the strings $\pi^{(i)}(x)$ share the same weight. Still we can show that this dependency does not decrease the success probability too much. In fact, to reduce the overhead of the reduction, we introduce more dependencies. For example, we always apply the basic procedure with the same "pivot" ℓ. Again, the random permutation ensures that this does not affect the quality of the output too much. This optimization (and others) allow us to achieve a low overhead and take $k = m \cdot \varepsilon^2$. As a result, we derive Theorem 1.4 and obtain a PPRG with constant locality, some fixed polynomial stretch and polynomial distinguishing advantage. Standard amplification techniques now yield Theorem 1.3.

2.2 Constructing LPRGs

Let us move to the case of LPRGs (Thm. 1.2). We would like to use the "basic procedure" but our predicate is not necessarily sensitive. For concreteness, think of the majority predicate. In this case, when recovering a 2-LIN equation, we are facing two sources of noise: one due to the error of the prediction algorithm, and the other due to the possibility that the current assignment x_S is "stable" – flipping its i-location does not change the value of the predicate (e.g., in the case of majority, any assignment with less than $\lfloor d/2 \rfloor$ ones). Hence, this approach is useful only if the predictor's success probability is larger than the probability of getting a stable assignment. Otherwise, our predictor, which may act arbitrarily, may decide to predict well only when the assignments are stable, and make a random guess otherwise. Therefore, we can prove only $(1-\varepsilon)$-unpredictability for some constant $\varepsilon > 0$.[8] This seems problematic as the transformation from unpredictability to pseudorandomness (Yao's theorem) fail for this range of parameters.

The solution is to employ a different transformation. Recently, it was shown by [26] (HRV) how to convert unpredictablity into pseudorandomness via the use of randomness extractors. We further note that this transformation is local as long as the underlying extractor is locally computable. The only problem is that, in general, one can show that it is impossible to compute good randomness extractors with constant locality. Fortunately, it turns out that for the special case of constant unpredictability and linear stretch, the HRV techniques can be applied with low-quality extractors for which there are (non-trivial) local implementations [36, 7]. This allows us to transform any $n + \Omega(n)$-long sequence with constant $(1-\varepsilon)$-unpredictability into an LPRG, while preserving constant locality.

Let us return to the first step in which prediction is used for inversion. In the LPRG setting we would like to base our construction on one-wayness with respect to $O(n)$ output-length (rather than super-linear length). Hence, the overhead of the reduction should be small, and we cannot apply the basic procedure to independent parts of the output as we did in the PPRG case. Our solution is to iterate the basic procedure n times where the graph G, the m-bit string y, and the hyperedge S are all *fixed*, and in each iteration a different pivot $j \in [n]$ is being planted in S. We show that, whp, this allows to find a string x' which agrees with x on, say, $2/3$ of the coordinates. At this point we employ the algorithm of [10] which recovers x given such an approximation x' and $f_{G,Q}(x)$.

2.3 Hardness of DSH

We move on to Theorem 1.5 in which we show that the pseudorandomness of $f_{G,Q}(x)$ for a random G implies strong inapproximability for the densest subhypergraph problem. Recall that one can amplify the inapproximability gap at the expense of increasing the cardinality of the hyperedges by taking graph product. In a nutshell, we show that the strong nature of pseudorandomness allows to apply some form of product amplification for "free" without changing the graph.

Suppose that for a random graph G, the pair (G, y) is indistinguishable from the pair $(G, f_{G,Q}(x))$, where y is a random m-bit string and x is a random n-bit string. Assume, without loss of generality, that $Q(1^d) = 1$. (Otherwise use its complement.) We define an operator ρ that given a graph G and an m-bit string z, deletes the i-th hyperedge if z_i is zero. It is not hard to see that ρ maps the "random" distribution to a random graph with $\sim m/2$ hyperedges which is likely to be a No-instance of $\frac{1}{2} - \mathsf{DSH}$. On the other hand, the pseudorandom distribution is mapped to a

[8] We show that the actual bound on ε depends on a new measure of "matching" sensitivity $\mu(Q)$ defined as follows: Look at the subgraph of the d-dimensional combinatorial hypercube whose nodes are the sensitive assignments of Q (i.e., the boundary and its neighbors), let M be a largest matching in the graph, and let $\mu(Q) = |M|/2^d$. For example, for majority with an odd arity d, it can be shown that all the assignments of Hamming weight $\lceil d/2 \rceil$ and $\lfloor d/2 \rfloor$ are in the matching and so the matching sensitivity is exactly $2\binom{d}{\lfloor d/2 \rfloor}/2^d$.

graph with a planted dense subgraph of density $\sim \frac{1}{2}$ (i.e., "Yes" instance of $\frac{1}{2} - \mathsf{DSH}$). Intuitively, this follows by noting that the set of nodes which are labeled by ones under x does not lose any hyperedge (as $Q(1^d) = 1$), while roughly half of the hyperedges are removed. (Otherwise, one can distinguish between the two distributions).

This leads to a basic hardness for $p = \frac{1}{2}$. Now, by a standard hybrid argument, one can show that the graph – which is a public index – can be reused, and so the tuple

$$(G, y^{(1)}, \dots, y^{(t)})$$

is indistinguishable from the tuple

$$(G, f_{G,P}(x^{(1)}), \dots, f_{G,Q}(x^{(t)}))$$

where the y's are random m-bit strings and the x's are random n-bit strings. Roughly speaking, each of these t copies allows us to re-apply the mapping ρ and further improve the parameter p by a factor of 2. (See full proof in Section 7.)

It is instructive to compare this to Feige's refutation assumption. The above distributions can be viewed as distributions over satisfiable and unsatisfiable CSPs where in both cases the graph G is randomly chosen. In contrast, Feige's refutation assumption is weaker as it essentially asks for distinguishers that work well with respect to arbitrary (worst-case) distribution over the satisfiable instances. Hence the graph cannot be reused and this form of amplification is prevented.

Organization Some preliminaries are given in Section 3 including background on Goldreich's function and cryptographic definitions. Sections 4– 6 are devoted to the proofs of Theorems 1.2–1.4, where Sections 4 and 5 describe the reductions from inversion to prediction (for the PPRG setting and for the LPRG setting), and Section 6 completes the proofs based on additional generic transformations. Finally, in Section 7, we prove Theorem 1.5.

3 Preliminaries

Basic notation We let $[n]$ denote the set $\{1, \dots, n\}$ and $[i..j]$ denote the set $\{i, i+1, \dots, j\}$ if $i \le j$, and the empty set otherwise. For a string $x \in \{0,1\}^n$ we let $x^{\oplus i}$ denote the string x with its i-th bit flipped. We let x_i denote the i-th bit of x. For a set $S \subseteq [n]$ we let x_S denote the restriction of x to the indices in S. If S is an ordered set $(i_1, \dots, i_d)$ then x_S is the *ordered* restriction of x, i.e., the string $x_{i_1} \dots x_{i_d}$. The Hamming weight of x is defined by $\mathrm{wt}(x) = |\{i \in [n] | x_i = 1\}|$. The uniform distribution over n-bit strings is denoted by $\mathcal{U}_n$.

Hypergraphs An (n, m, d) graph is a hypergraph over n vertices $[n]$ with m hyperedges each of cardinality d. We assume that each edge $S = (i_1, \dots, i_d)$ is ordered, and that all the d members of an edge are distinct. We also assume that the edges are ordered from 1 to m. Hence, we can represent G by an ordered list $(S_1, \dots, S_m)$ of d-sized (ordered) hyperedges. For indices $i \le j \in [m]$ we let $G_{[i..j]}$ denote the subgraph of G which contains the edges $(S_i, \dots, S_j)$. We let $\mathcal{G}_{n,m,d}$ denote the distribution over (n, m, d) graphs in which a graph is chosen by picking each edge uniformly and independently at random from all the possible $n^{(d)} \stackrel{\text{def}}{=} n \cdot (n-1) \cdot \ldots \cdot (n-d+1)$ ordered hyperedges.

Sensitivity and influence measures Every d-local predicate $Q : \{0,1\}^d \to \{0,1\}$ naturally partitions the d-dimensional Hamming cube into a zero-set V_0 and a one-set V_1 where $V_b = \{w \in \{0,1\}^d | Q(w) = b\}$. We let $H_Q = (V_0 \cup V_1, E)$ denote the bipartite graph induced by this partition, namely, $(u,v) \in V_0 \times V_1$ is an edge if the strings u and v differ in exactly one coordinate. We define the following measures of Q. We let $\partial(Q) = \Pr_{w \xleftarrow{R} \{0,1\}^d}[w \in V_1]$ denote the boundary of Q and let $\bar{\partial}(Q) = 1 - \partial(Q)$. A matching $M \subseteq V_0 \times V_1$ is a set of pair-wise *distinct* edges in H_Q, i.e., for every pair (u,v) and (u',v') in M we have $u \neq u'$ and $v \neq v'$. We will be interesting in the probability that a randomly selected node lands inside a maximal matching:

$$\begin{aligned}\mathrm{Match}(Q) &= \max_M \Pr_{w \xleftarrow{R} \{0,1\}^d}[\exists u \text{ s.t. } (w,u) \in M \text{ or } (u,w) \in M] \\ &= \max_M 2|M|/2^n,\end{aligned}$$

where the maximum is taken over all matchings in H_Q. The *matching density* $\mathrm{Match}(Q)$ will be used to measure the "sensitivity" of Q. We also rely on more traditional measures of sensitivity as follows. The influence of the i-th coordinate of Q is defined by $\mathrm{Inf}_i(Q) = \Pr_{w \xleftarrow{R} \{0,1\}^d}[Q(w) \neq Q(w^{\oplus i})]$. We let $\mathrm{Inf}_{\max}(Q)$ denote the maximal influence of a single coordinate $\max_{i \in [d]} \mathrm{Inf}_i(Q)$. The following simple proposition relates the different sensitivity measures.

Proposition 3.1. *For any d-local predicate Q we have:*

$$\begin{aligned}\mathrm{Inf}_{\max}(Q) &\leq \mathrm{Match}(Q) \\ &\leq 2\min(\partial(Q), \bar{\partial}(Q)) \leq \sum_i \mathrm{Inf}_i(Q) \leq 2d\partial(Q).\end{aligned}$$

Proof. Consider the graph H_Q and color each edge (u,v) by the color $i \in [d]$ for which $u = v^{\oplus i}$. The inequalities follow by counting edges while noting that $\mathrm{Inf}_{\max}(Q)$ measures the cardinality of the largest monochromatic matching (in nodes), $\sum_i \mathrm{Inf}_i(Q)$ measures the sum of degrees, and d is an upper bound on the maximal degree. $\square$

Also, recall that by [31], if Q is balanced then we also have $c \log d/d \leq \mathrm{Inf}_{\max}(Q)$ where c is a universal constant.

3.1 Cryptographic definitions

Collection of Functions Let $s = s(n)$ and $m = m(n)$ be integer-valued functions which are polynomially bounded. A collection of functions $F : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ takes two inputs a public collection index $k \in \{0,1\}^s$ and an input $x \in \{0,1\}^n$, the output $F(k,x)$ consists of the evaluation $F_k(x)$ of the point x under k-th function in the collection. We always assume that the collection is equipped with two efficient algorithms: an index-sampling algorithm K which given 1^n samples a index $k \in \{0,1\}^s$, and an evaluation algorithm which given $(1^n, k \in \{0,1\}^s, x \in \{0,1\}^n)$ outputs $F_k(x)$. We say that the collection is in $\mathbf{NC^0}$ if there exists a constant d (which does not grow with n) such that for every fixed k the function F_k has output locality of d. (In our case, k is typically the dependencies graph G.) All the cryptographic primitives in this paper are modeled as collection of functions. We will always assume that the adversary that tries to break the primitive gets the collection index as a public parameter. Moreover, our constructions are all in the "public-coin" setting, and so they remain secure even if the adversary gets the coins used to sample the index of the collection.

In the following definitions we let $F : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ be a collection of functions where K is the corresponding index-sampling algorithm. We also let $\varepsilon = \varepsilon(n) \in (0,1)$ be a parameter which measures the security of the primitive. All probabilities are taken over the explicit random variables and in addition over the internal coin tosses of the adversary algorithms.

One-way functions Informally, a function is one-way if given a random image y it is hard to find a preimage x. We will also use a stronger variant of approximate one-wayness in which even the easier task of finding a string which approximates the preimage is infeasible. Formally, for a proximity parameter $\delta = \delta(n) \in (0, \frac{1}{2})$ and security parameter $\varepsilon = \varepsilon(n) \in (0,1)$, we say that a collection of functions $F : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ is (ε, δ) *approximate one-way function* (AOWF) if for every efficient adversary $\mathcal{A}$ which outputs a poly(n) list of candidates, and sufficiently large n's the quantity

$$\Pr_{\substack{k \xleftarrow{R} K(1^n) \\ x \xleftarrow{R} \mathcal{U}_n \\ y = F_k(x)}} [\exists z \in \mathcal{A}(k,y), z' \in F_k^{-1}(y) \text{ s.t. } \Delta(z,z') \le \delta(n)],$$

is bounded by $\varepsilon(n)$ where $\Delta(\cdot,\cdot)$ denotes the relative Hamming distance. In the special case of $\delta = 0$, the collection F is referred to as ε *one-way*, or simply *one-way* if in addition ε is a negligible function. This is consistent with standard definitions of one-wayness (cf. [22]) as when $\delta = 0$, the algorithm can efficiently check which of the candidates (if any) is a preimage and output only a single candidate z rather than a list.

Indistinguishability Let $Y = \{Y_n\}$ and $Z = \{Z_n\}$ be a pair of distribution ensembles. We say that a pair of distribution ensembles $Y = \{Y_n\}$ and $Z = \{Z_n\}$ is ε-indistinguishable if for every efficient adversary $\mathcal{A}$, the *distinguishing advantage*

$$|\Pr[\mathcal{A}(1^n, Y) = 1] - \Pr[\mathcal{A}(1^n, Z) = 1]|$$

is at most $\varepsilon(n)$. We say that the ensembles are ε statistically-close (or statistically-indistinguishable) if the above holds for computationally unbounded adversaries.

Pseudorandom and unpredictability generators Let $m = m(n) > n$ be a length parameter. A collection of functions $F : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ is ε *pseudorandom generator* (PRG) if the ensemble $(K(1^n), F_{K(1^n)}(\mathcal{U}_n))$ is ε-indistinguishable from the ensemble $(K(1^n), \mathcal{U}_{m(n)})$. When ε is negligible, we refer to F as a *pseudorandom generator*. The collection F is ε *unpredictable generator* (UG) if for every efficient adversary $\mathcal{A}$ and every sequence of indices $\{i_n\}$, where $i_n \in [m]$, we have that

$$\Pr_{k \xleftarrow{R} K(1^n), x \xleftarrow{R} \mathcal{U}_n, y = F_k(x)} [A(k, y_{[1..i_n-1]}) = F_k(x)_{i_n}] < \varepsilon(n).$$

We say that F is ε *last-bit unpredictable generator* (LUG) if the above is true for the sequence of indices $i_n = m(n)$.

We refer to $m(n) - n$ as the *stretch* of the PRG (resp., UG), and classify it as *sublinear* if $m(n) - n = o(n)$, *linear* if $m(n) - n = \Omega(n)$ and *polynomial* if $m(n) - n > n^{1+\Omega(1)}$.

Remark 3.2 (Uniform unpredictability)**.** *One may consider a uniform version of the unpredictability definition where the sequence of indices $\{i_n\}$ should be generated in polynomial-time by an efficient algorithm which is given 1^n (and is allowed to err with negligible probability). We prefer the non-uniform version as it will be easier to work with. However, it is not hard to show that the two definitions are essentially equivalent. Formally, for any inverse polynomials ε, and δ the notion of ε-unpredictability (as per the above definition) implies* uniform *$(\varepsilon+\delta)$-unpredictability. To see this, consider an efficient adversary $\mathcal{A}$ that contradicts non-uniform unpredictability, and let us construct an efficient algorithm $\mathcal{B}$ that generates a "good" sequence of indices. The idea is to estimate the quantity p_i which is the success probability of $\mathcal{A}$ in predicting the i-th bit of the sequence $F_{K(1^n)}(\mathcal{U}_n)$ based on the $i-1$ prefix. By standard Chernoff bound, we can efficiently estimate each of the p_i's (for $i \in [m(n)]$) with an additive error of δ with all but exponentially small failure probability, and then choose the best index.*

3.2 Goldreich's Function

For a predicate $Q : \{0,1\}^d \to \{0,1\}$ and an (n, m, d) graph $G = ([n], (S_1, \dots, S_m))$ we define the function $f_{G,Q} : \{0,1\}^n \to \{0,1\}^m$ as follows: Given an n-bit input x, the i-th output bit y_i is computed by applying Q to the restriction of x to the i-th hyperedge S_i, i.e., $y_i = Q(x_{S_i})$. For $m = m(n)$, the function collection $\mathcal{F}_{Q,m}$ is defined via the mapping

$$(G, x) \mapsto f_{G,Q}(x),$$

where (for each length parameter n) G is an $(n, m(n), d)$ graph which serves as a public key, and x is an n-bit string which serves as an input. The key-generation selects the graph G uniformly at random from $\mathcal{G}_{n,m,d}$.

In the following we show that, for the ensemble $\mathcal{F}_{Q,m}$, last-bit unpredictability and standard unpredictability are equivalent, and so are approximate one-wayness and standard one-wayness. These properties will be useful later.

Proposition 3.3. *For every constant locality $d \in \mathbb{N}$, predicate $Q : \{0,1\}^d \to \{0,1\}$, $m = \mathrm{poly}(n)$ and $\varepsilon = 1/\mathrm{poly}(n)$ the following holds: If $\mathcal{F}_{Q,m}$ is ε last-bit unpredictable then $\mathcal{F}_{Q,m}$ is also $\varepsilon(1 + o(1))$-unpredictable.*

Proof. The proof follows from the invariance of the distribution $\mathcal{G}_{n,m,d}$ under permutations of the order of the edges. Formally, assume towards a contradiction, that there exists a next-bit predictor $\mathbf{P}$ and a sequence of indices $\{i_n\}$ such that

$$\varepsilon(n) = \Pr_{\substack{x \xleftarrow{R} \mathcal{U}_n, G \xleftarrow{R} \mathcal{G}_{n,m,d} \\ y = f_{G,Q}(x)}} [\mathbf{P}(G, y_{[1..i_n-1]}) = y_{i_n}],$$

for infinitely many n's. We construct a last-bit predictor $\mathbf{P}'$ with success probability of $\varepsilon - o(\varepsilon)$ as follows. First, use Remark 3.2 to efficiently find an index $j \in [m]$ such that, with probability $1 - \mathrm{neg}(n)$ over the coins of $\mathbf{P}'$, it holds that $\Pr[\mathbf{P}(G, y_{1..j}) = y_{j+1}] > \varepsilon(n) - \varepsilon(n)/n$ where the probability is taken over a random input and random coin tosses of $\mathbf{P}$. Now given an input

$(G, y_{[1..m-1]})$, construct the graph G' by swapping the j-th edge S_j of G with its last edge S_m. Then, $\mathbf{P}'$ invokes $\mathbf{P}$ on the input $(G', y_{[1..j-1]})$ and outputs the result. It is not hard to verify that this transformation maps the distribution $(G \xleftarrow{R} \mathcal{G}_{n,m,d}, f_{G,Q}(\mathcal{U}_n)_{[1..m-1]})$ to $(G \xleftarrow{R} \mathcal{G}_{n,m,d}, f_{G,Q}(\mathcal{U}_n)_{[1..j]})$, and so $\mathbf{P}'$ predicts the last bit with probability $\varepsilon(n) - \varepsilon(n)/n - \mathrm{neg}(n)$ as required. $\square$

Proposition 3.4. *For every constant locality $d \in \mathbb{N}$, predicate $Q : \{0,1\}^d \to \{0,1\}$, and fixed proximity parameter $\delta \in (0, \frac{1}{2})$ (which may depend on d), there exists a constant $c = c(d, \delta)$, such that for every inverse polynomial $\varepsilon = \varepsilon(n)$ the following hold.*

1. *For $m > cn$, if $\mathcal{F}_{Q,m}$ is ε one-way then $\mathcal{F}_{Q,m}$ is also $(\varepsilon' = \varepsilon + o(1), \delta)$ approximate one-way.*
2. *If $\mathcal{F}_{Q,m+cn}$ is ε one-way then $\mathcal{F}_{Q,m}$ is $(\varepsilon' = \varepsilon(1 + o(1)), \delta)$ approximate one-way.*

Proof. The proof is based on an algorithm of [10] which, given an approximation of x, inverts $f_{G,Q}(x)$. Formally, let $d \in \mathbb{N}$, $\delta \in (0, \frac{1}{2})$ be constants, and let Q be a d-local predicate. In [10, Thm. 2] it is shown that there exists a constant $c = c(d, \delta)$ and an efficient algorithm A that inverts $\mathcal{F}_{m,Q}$ given a δ-approximation of the preimage x, for every $m \geq cn$. More precisely, it is shown that for a fraction of $1 - o(1)$ of all (m, n, d) hypergraphs G, we have that

$$\Pr_{\substack{x \xleftarrow{R} \mathcal{U}_n \\ y = f_{G,Q}(x)}} [A(y, x') \in f^{-1}_{G,Q}(y) | \Delta(x', x) \leq \delta] > 1 - \mathrm{neg}(n), \tag{1}$$

where $\Delta(\cdot, \cdot)$ denotes the relative Hamming distance.

We can now prove the proposition. Suppose that $\mathcal{F}_{Q,m}$ is not (ε', δ) approximate one-way. That is, there exists an algorithm B which given $(G, y = f_{G,Q}(x))$ outputs a list of strings L such that with probability ε', over $G \xleftarrow{R} \mathcal{G}_{n,m,d}$ and $x \xleftarrow{R} \mathcal{U}_n$, the list L contains a string x' which δ-approximates x. To prove the first item (where $m > cn$), let B generate a list L, and for each member $x' \in L$ invoke the algorithm A with G, y and x'. At the end, check whether one of the outputs is a preimage of y and output this result. By a union bound, the overall success probability is $\varepsilon = \varepsilon' - o(1)$ as required.

We move to the second item, and construct an ε-inverter for $\mathcal{F}_{Q,m+cn}$. Given an input $(G, y = f_{G,Q}(x))$, partition G and y into two pairs (G_1, y_1) and (G_2, y_2) where G_1 (resp., y_1) consists of the first m hyperedges of G (resp., bits of y), and G_2 (resp., y_2) consists the last cn hyperedges of G (resp. bits of y). Now first apply B to (G_1, y_1) to obtain a list L of δ-close candidates, and then apply A to (G_2, y_2, x') for every $x' \in L$. Let us condition on the event that B succeeds, and the event that G_2 is a "good" graph for A, i.e., that G_2 satisfies Eq. 1. The two events are independent (as G_2 is independent of G_1) and so the probability that they both happen is $\varepsilon'(1 - o(1))$. Conditioned on this, the algorithm A succeeds with probability $1 - \mathrm{neg}(n)$, and so by a union bound we get that the overall success probability is $\varepsilon = \varepsilon'(1 - o(1)) - \mathrm{neg}(n) = \varepsilon'(1 - o(1))$, as needed. $\square$

4 Random Local Functions with $(\frac{1}{2} + 1/\mathrm{poly})$-Unpredictability

In this section we prove the following theorem:

Theorem 4.1 (one-way $\Rightarrow$ $(\frac{1}{2} + 1/\mathrm{poly})$-unpredictable)**.** *Let $d \in \mathbb{N}$ be a constant locality parameter and $Q : \{0,1\}^d \to \{0,1\}$ be a sensitive predicate. Then, for every $m \geq n$ and inverse polynomial ε, if $\mathcal{F}_{Q,m/\varepsilon^2}$ is ε-one-way then $\mathcal{F}_{Q,m}$ is $(\frac{1}{2} + c\varepsilon)$-UG, for some constant $c = c(d) > 0$.*

For simplicity, and, without loss of generality, we assume that the first variable of Q has maximal influence, i.e., $\mathrm{Inf}_1(Q) = 1$. We rely on the following notation. For a permutation $\pi : [n] \to [n]$ and an ordered set $S = \{i_1, \dots, i_d\} \subseteq [n]$ we let $\pi(S)$ denote the ordered set $\{\pi(i_1), \dots, \pi(i_d)\} \subseteq [n]$. For an (m, n, d) graph $G = (S_1, \dots, S_m)$ we let $\pi(G)$ denote the (m, n, d) graph $(\pi(S_1), \dots, \pi(S_m))$. For a string $x \in \{0,1\}^n$, the string $\pi(x)$ is the string whose coordinates are permuted under π.

To prove the theorem, assume towards a contradiction that we have a predictor $\mathbf{P}$ that predicts the last output with probability $\frac{1}{2} + \varepsilon$ for infinitely many n's where ε is an inverse polynomial. (A standard next-bit predictor can be transformed to such predictor by Prop. 3.3.) Syntactically, $\mathbf{P}$ takes as an input an $(m-1, n, d)$ graph G, an $(m-1)$-bit string y (supposedly $y = f_{G,Q}(x)$), and an hyperedge S, and outputs its guess for $Q(x_S)$.

In order to invert $\mathcal{F}_{Q,tm}$ we will make use of the following sub-routine Vote (Figure 1), which essentially corresponds to the "basic procedure" described in Section 2. The subroutine takes a "small" (n, m, d) graph G, a corresponding output string $y = f_{G,Q}(x)$, and uses the predictor $\mathbf{P}$ to approximate the value $x_i \oplus x_\ell$ where the indices i and ℓ are given as additional inputs. (The index ℓ is referred to as "global advice" as it will be reused among different iterations).

- **Input**: an (n, m, d) graph G, a string $y \in \{0,1\}^m$, an index $i \in [n]$.
- **Global advice**: index $\ell \in [n]$.

1. Choose a random hyperedge $S = (S_1, \dots, S_d)$ from G subject to the constraint $S_1 = i$ and $\ell \notin \{S_2, \dots, S_d\}$. Let s denote the index of S in G, i.e., $S = G_s$. If no such edge exist abort with a failure symbol.
2. Choose a random permutation $\pi : [n] \to [n]$. Let $G' = \pi(G_{-s})$ be the graph obtained by removing S and permuting the nodes under π. Similarly, let $y' = y_{-s}$ be the string y with its s-th bit removed. Finally, define the hyperedge
$$S' = \pi(S_{1 \leftarrow \ell}) = (\pi(\ell), \pi(S_2), \dots, \pi(S_d)).$$
3. Output $\mathbf{P}(G', y', S') \oplus y_s$.

Figure 1: Algorithm Vote.

The algorithm Invert (Figure 2) uses Vote to invert a random member of $\mathcal{F}_{Q,tm}$.

Analysis. From now on fix a sufficiently large input length n for which $\mathbf{P}$ is successful. Let us focus on the way our algorithm recovers one fixed variable $i \in [n]$. First we will show that in each call to the subroutine Vote, whenever the predictor predicts correctly, we get a "good" vote regarding whether x_i and x_ℓ agree. Hence, if our global guess b for x_ℓ is correct, and most of the predictions (in the i-th iteration of the outer loop) are good, we successfully recover x_i. In order to show that the predictor succeeds well, we should analyze the distribution on which it is invoked. In particular, we should make sure that the marginal distribution of each query to $\mathbf{P}$ is roughly uniform, and, that the dependencies between the queries (during the i-th iteration of the outer loop) are minor. This is a bit subtle, as there are some dependencies due to the common input x

- **Input**: an (n, tm, d) graph G and a string $y \in \{0,1\}^{tm}$, where t is a parmeter.

1. Partition the input (G, y) to t blocks of length m where $y^{(j)} = y_{[(j-1)m+1..jm]}$ and $G^{(j)} = G_{[(j-1)m+1..jm]}$.
2. Choose a random pivot $\ell \overset{R}{\leftarrow} [n]$, and a random bit b (our guess for x_ℓ).
3. For each $i \in [n]$ we recover the i-th bit of x as follows:
   (a) For each $j \in [t]$, invoke the subroutine Vote on the input $(G^{(j)}, y^{(j)}, i)$ with global advice ℓ, and record the output as $v_{i,j}$.
   (b) Set v_i to be the majority vote of all $v_{i,j}$'s.
4. If $b = 0$ output v; otherwise output the complement $\mathbf{1} - v$.

Figure 2: Algorithm Invert.

and common pivot ℓ. To cope with this, we will show (in Lemma 4.2) that these queries can be viewed as independent samples, alas taken from a "modified" distribution which is different from the uniform. Later (in Lemma 4.3) we will show that, whp, $\mathbf{P}$ predicts well over this modified distribution.

The modified distribution. Let X_k denote the set of all n-bit strings whose Hamming weight is exactly k. For $k \in [n]$ and a bit $\sigma \in \{0,1\}$ define the distribution $D_{k,\sigma}$ over tuples (G, r, y, T) as follows: the graph G is sampled from $\mathcal{G}_{n,m-1,d}$, the string r is uniformly chosen from X_k, the string y equals to $f_{Q,G}(r)$, and the hyperedge $T = \{T_1, \dots, T_d\}$ is chosen uniformly at random subject to $r_{T_1} = \sigma$. In Section 4.1, we prove the following lemma:

Lemma 4.2. *Let (G, y, ℓ, i) be the input to* Vote *where $G \overset{R}{\leftarrow} \mathcal{G}_{n,m,d}$, the indices $\ell \in [n]$ and $i \in [n]$ are arbitrarily* fixed *and $y = f_{Q,G}(x)$ for an arbitrary* fixed *$x \in \{0,1\}^n$. Consider the random process $\mathsf{Vote}(G, y, \ell, i)$ induced by the internal randomness and the distribution of G. Then, the following hold:*

1. *The process fails with probability at most $1/2$.*
2. *Conditioned on not failing, the random variable (G', x', y', S') is distributed according to $D_{\mathrm{wt}(x),x_\ell}$, where $x' = \pi(x)$ and $\mathrm{wt}(x)$ is the Hamming weight of x.*
3. *Conditioned on not failing, if the outcome of the predictor $\mathbf{P}(G', y', S')$ equals to $Q(x'_{S'})$ then the output of* Vote *is $x_i \oplus x_\ell$ (with probability 1).*

Our next goal is to show that with good probability over x and the pivot ℓ, the predictor $\mathbf{P}$ predicts well on the distribution $D_{\mathrm{wt}(x),x_\ell}$. In Section 4.2, we prove the following lemma:

Lemma 4.3. *With probability $\Omega(\varepsilon)$ over a random choice of the input $x \overset{R}{\leftarrow} \mathcal{U}_n$ and the pivot $\ell \overset{R}{\leftarrow} [n]$, we have that*

$$\Pr_{(G,r,y,T) \overset{R}{\leftarrow} D_{\mathrm{wt}(x),x_\ell}} [\mathbf{P}(G, y, T) = Q(r_T)] > \frac{1}{2} + \varepsilon/2.$$

We can now prove the theorem.

Proof of Thm. 4.1 given the lemmas. Let us condition on the event that x and ℓ satisfy the equation of Lemma 4.3, and that our guess b for x_ℓ was successful. By Lemma 4.3, this event happens with probability $\Omega(\varepsilon) \cdot \frac{1}{2} = \Omega(\varepsilon)$. From now on, we assume that x, ℓ and b are fixed. We claim that the probability that the output of Invert *disagrees* with x on the i-th bit for a fixed index $i \in [n]$ is at most $\exp(-\Omega(t\varepsilon^2))$. Indeed, Lemma 4.2 guarantees that each call to the subroutine Vote fails independently with probability at most $\frac{1}{2}$. Furthermore, conditioned on not failing, in each call to Vote the predictor $\mathbf{P}$ gets an *independent* sample from $D_{\mathrm{wt}(x),x_\ell}$. Hence, the subroutine returns a correct vote, i.e., $v_{i,j} = x_i \oplus x_\ell$, with probability $\frac{1}{2} + \Omega(\varepsilon)$. The rest of the analysis now follows from standard concentration bounds.

Formally, define a random variable α_j which takes the value 1 if the vote $v_{i,j}$ is good i.e., $v_{i,j} = x_i \oplus x_\ell$, takes the value -1 if the vote is incorrect, and takes the value 0 if the subroutine Vote fails. We recover x_i correctly if $\sum \alpha_j$ is positive (as our guess b for x_ℓ is assumed to be correct). As we already saw the α_j's are identically and independently distributed random variables which takes the value 0 with probability at most $1/2$, and conditioned on not being zero take the value 1 with probability at least $\frac{1}{2} + \Omega(\varepsilon)$. By a Chernoff bound, the probability of seeing at most $2t/3$ zeroes is at least $1 - \exp(-\Omega(t))$. Now, conditioned on this event, the $t' > t/3$ remaining non-zero α_i's are i.i.d random variables that take the value ± 1 w/p $\frac{1}{2} \pm \Omega(\varepsilon)$. Hence, by Hoeffding's inequality, the probability that their sum is negative is at most $\exp(-\Omega(t'\varepsilon^2)) = \exp(-\Omega(t\varepsilon^2))$. Overall, by a union bound, the probability that the i-th bit of x is badly recovered (i.e., $\sum \alpha_j \leq 0$) is at most $\exp(-\Omega(t\varepsilon^2)) + \exp(-\Omega(t)) < \exp(-\Omega(t\varepsilon^2))$.

This already implies a weaker version of Thm. 4.1, as by taking $t = O(\log n/\varepsilon^2)$ we get that each bit of x is recovered with probability $1 - 1/n^2$ and so by, a union bound, we recover all the bits of x with overall probability of $\Omega(\varepsilon)(1 - o(1)) > \Omega(\varepsilon)$. This shows that $\mathcal{F}_{Q,O(m \log n/\varepsilon^2)}$ is $\Omega(\varepsilon)$-one-way. To obtain the stronger version (without the $\log n$ overhead), we employ Prop. 3.4. Namely, we let $t = O(1/\varepsilon^2)$, and so with probability $\Omega(\varepsilon)$ each bit of x is recovered with probability $3/4$. These predictions are not independent. However, by Markov (conditioned on the above) at least $2/3$ of the indices are recovered correctly with some constant probability, and overall we get an inverter that finds a $1/3$-close approximation of x with probability $\Omega(\varepsilon)$, which, by Prop. 3.4 (part 2), contradicts the $\Omega(\varepsilon)$-one-wayness of $\mathcal{F}_{Q,m'}$, where $m' = O(m/\varepsilon^2) + c_d n$ and c_d is a constant that depends only in the locality d. Overall, we showed that if $\mathcal{F}_{Q,m'}$ is ε'-one-way then $\mathcal{F}_{Q,m}$ is $\frac{1}{2} + \varepsilon$ hard to predict, for $\varepsilon' = \Omega(\varepsilon)$. By letting $\varepsilon' = c'\varepsilon$ for some constant $c' = c'(d)$, we can set $m' = m/\varepsilon'^2$ (as $m \geq n$), and derive the theorem. $\square$

In Section 4.3 we will show that the above theorem generalizes to variants of $\mathcal{F}_{Q,m}$ that capture some of the existing heuristic candidates.

4.1 Proof of Lemma 4.2

First item. We lower-bound the probability of failure. First, the probability that G has no hyperedge whose first entry equals to i is $(1 - 1/n)^m < (1 - 1/n)^n < 2/5$. Conditioned on having an hyperedge whose first entry is i, the probability of having ℓ as one of its other entries is at most $O(d/n)$. Hence, by a union bound, the failure probability is at most $2/5 + O(d/n) < 1/2$.

Second item. Fix x and let k be its Hamming weight. Let x_+ be the support of x, i.e., set of indices j such that $x_j = 1$. Consider the distribution of the pair (G, S) defined in Step 1 of Vote. This pair can be sampled independently as follows: first choose a random hyperedge S whose first entry is i and ℓ does not appear in its other entries, then construct G by choosing a random graph R from $\mathcal{G}_{n,m-1,d}$ and by planting S in a random location at R. From this view, it follows that the pair $(G_{-s}, S_{1\leftarrow\ell})$ (defined in Step 2) is independently distributed such that $G_{-s} \stackrel{R}{\leftarrow} \mathcal{G}_{n,m-1,d}$ and $S_{1\leftarrow\ell}$ is a random hyperedge whose first entry is ℓ. Since x is fixed and $y' = y_{-s} = f_{Q,G_{-s}}(x)$, we have now a full understanding of the distribution of the tuple $(G_{-s}, x, y', S_{1\leftarrow\ell})$.

We will now analyze the effect of the permutation π. Let $x' = \pi(x)$ and $G' = \pi(G_{-s})$. First, observe that for every fixed permutation π the tuple (G', x', y') satisfies $y' = f_{Q,G'}(x')$ since $y' = f_{Q,G_{-s}}(x)$. Moreover, since G_{-s} is taken from $\mathcal{G}_{n,m-1,d}$, so is $G' = \pi(G_{-s})$ even when π is fixed. Let us now pretend that the random permutation π is selected in two steps. First, choose a random set $A \subseteq [n]$ of size k and then, in the second step, choose a random permutation π_A subject to the constraint that $\pi(x_+) = A$.

Consider the distribution of x' which is induced by the random choice of A, i.e., before the *second* step was performed. Already in this phase we have that x' is uniformly and independently distributed according to X_k. Hence, $(G' \stackrel{R}{\leftarrow} \mathcal{G}_{n,m-1,d}, x' \stackrel{R}{\leftarrow} X_k, y' = f_{Q,G'}(x'))$.

Let us now fix both G' and A (and therefore also x') and so the only randomness left is due to the choice of π_A. We argue that the hyperedge $S' = \pi_A(S_{1\leftarrow\ell})$ is uniformly and independently distributed under the constraint that the first entry τ of S' satisfies $x'_\tau = x_\ell$. To see this, recall that the first entry of $S_{1\leftarrow\ell}$ is ℓ, and so the entry $S'_1 = \pi_A(\ell)$ is mapped to a random location in the set $\left\{j : x'_j = x_\ell\right\}$, also recall that the last $d-1$ entries of $S_{1\leftarrow\ell}$ are random indices (different than ℓ) and so for every fixing of π_A the $d-1$ last entries of S' are still random. This completes the proof as we showed that the tuple (G', x', y', S') is distributed properly.

Third item. Let us move to the third item. Suppose that $\mathbf{P}$ outputs the bit $Q(x'_{S'})$. Then, since $S' = \pi(S_{1\leftarrow\ell})$ and $x' = \pi(x)$, the result equals to $Q(x_{S_{1\leftarrow\ell}})$, which, by definition, can be written as $Q(x_S) \oplus (x_\ell \oplus x_i)$. Hence, when this bit is XOR-ed with $Q(x_S)$, we get $x_\ell \oplus x_i$, as required. □

4.2 Proof of Lemma 4.3

We define a set X of "good" inputs by taking all the strings of weight $k \in K$ for some set $K \subset [n]$. We will show that X captures $\Omega(\varepsilon)$ of the mass of all n-bit strings, and that for each good x the predictor $\mathbf{P}$ predicts well with respect to the cylinder $X_{\mathrm{wt}(x)}$. Specifically, let $p_k = \Pr[\mathcal{U}_n \in X_k]$ and let q_k be

$$\Pr_{x \stackrel{R}{\leftarrow} X_k, G \stackrel{R}{\leftarrow} \mathcal{G}_{n,m-1,d}, S \stackrel{R}{\leftarrow} \binom{[n]}{d}} [\mathbf{P}(G, f_{Q,G}(x), S) = Q(x_S)].$$

We let $X = \bigcup_{k \in K} X_k$ where K is defined via the following claim.

Claim 4.4. *There exists a set* $K \subseteq \left\{n/2 - n^{2/3}, \ldots, n/2 + n^{2/3}\right\}$ *for which:*

$$\sum_{k \in K} p_k > \varepsilon/3 \qquad (2)$$

$$\forall k \in K, q_k > \frac{1}{2} + \varepsilon/2 \qquad (3)$$

Proof. By definition, we have

$$\sum_{k=1}^{n} p_k \cdot q_k > \frac{1}{2} + \varepsilon.$$

By a Chernoff bound, for all $k \notin (n/2 \pm n^{2/3})$ we have $p_k < \mathrm{neg}(n)$, and therefore,

$$\sum_{k\in(n/2\pm n^{2/3})} p_k \cdot q_k > \frac{1}{2} + \varepsilon - \mathrm{neg}(n).$$

Let $K \subseteq (n/2 \pm n^{2/3})$ be the set of indices for which $q_k > \frac{1}{2} + \varepsilon/2$. By Markov's inequality, $\sum_{k\in K} p_k > \varepsilon/3$, as otherwise,

$$\frac{1}{2}+\varepsilon-\mathrm{neg}(n) < \sum_{k\in(n/2\pm n^{2/3})} p_k \cdot q_k = \sum_{k\in K} p_k \cdot q_k + \sum_{k\in(n/2\pm n^{2/3})\setminus K} p_k \cdot q_k < \varepsilon/3 + \left(\frac{1}{2}+\varepsilon/2\right) = \frac{1}{2}+5\varepsilon/6,$$

and, since ε is an inverse polynomial, we derive a contradiction for all sufficiently large n's. $\square$

For a bit $\sigma \in \{0,1\}$ let $q_{k,\sigma}$ be

$$\Pr_{x \overset{R}{\leftarrow} X_k, G \overset{R}{\leftarrow} \mathcal{G}_{n,m-1,d}, S \overset{R}{\leftarrow} \binom{[n]}{d}} [\mathbf{P}(G, f_{Q,G}(x), S) = Q(x_S) | x_{S_1} = \sigma],$$

where S_1 denotes the first entry of S. Observe that for every k there exists a $\sigma_k \in \{0,1\}$ for which $q_{k,\sigma_k} \geq q_k$. Hence, by the above claim, we have that with probability $\Omega(\varepsilon)$ over a random choice of the input $x \overset{R}{\leftarrow} \mathcal{U}_n$, we have that $x \in X$ and so

$$\Pr_{(G,r,y,T) \overset{R}{\leftarrow} D_{\mathrm{wt}(x),\sigma_{\mathrm{wt}(x)}}} [\mathbf{P}(G, y, T) = Q(r_T)] > \frac{1}{2} + \varepsilon/2.$$

To complete the proof of the lemma, observe that for every $x \in X$, since x is balanced, the probability that a random pivot $\ell \overset{R}{\leftarrow} [n]$ satisfies $x_\ell = \sigma_{\mathrm{wt}(x)}$ is at least $(n/2 - n^{2/3})/n = \frac{1}{2} - o(1)$. Hence, with probability $\Omega(\varepsilon)$ over the random choice of x and ℓ, we have that $q_{\mathrm{wt}(x),x_\ell} > \frac{1}{2} + \varepsilon/2$ as required. $\square$

4.3 Generalization to the noisy case

Let Q be a sensitive predicate. Consider the collection $\mathcal{F}'_{Q,m}$ which is indexed by a random (m,n,d) graph G, and given x it outputs $(G, f_{G,Q}(x) \oplus E)$, where E is a "noise" distribution over $\{0,1\}^m$ with the following properties: (1) it is independent of G and x; (2) it is invariant under permutations: for every $\pi : [m] \to [m]$ the random variable $\pi(E)$ is identically distributed as E; and (3) it can be partitioned to t blocks $E = (E_i)$ of length b each, such that each block is identically and independently distributed. We may also slightly generalize this and allow E to have an index k which is sampled and given as part of the index of the collection $\mathcal{F}'_{Q,m}$. One-wayness is defined with respect to x, that is, we say that $\mathcal{F}'_{Q,m}$ is ε-one-way if it is hard to recover x with probability ε. Theorem 4.1 can be generalized to this setting as follows.

Theorem 4.5 (Thm. 4.1: generalization). *Let $d \in \mathbb{N}$ be a constant locality parameter and $Q : \{0,1\}^d \to \{0,1\}$ be a sensitive predicate. Let $m \geq n$ be the block length of the noise E. Then, for every inverse polynomial ε, if $\mathcal{F}'_{Q,m \lg n/\varepsilon^2}$ is ε-one-way then $\mathcal{F}'_{Q,m}$ is $(\frac{1}{2} + \Omega(\varepsilon))$-unpredictable.*

The proof is the essentially the same as the proof of Thm. 4.1. Algorithm Invert is being used, and its analysis does not change due to the symmetry and independence of the noise. The only difference is that we do not know whether the reduction from approximate one-wayness to one-wayness holds and so we employ the algorithm invert with $t = \lg n/\varepsilon^2$ overhead to ensure inversion rather than approximate inversion.

This generalization can capture the case of noisy-local-parity construction ([2, 7, 4]) where Q is linear (i.e., "exclusive-or") and each bit of E is just an independently chosen noisy bit taken to be one with probability $p < \frac{1}{2}$ (e.g., 1/4). It also captures a variant of the MST construction [36], and so in both cases we prove weak pseudorandomness from one-wayness.

5 Random Local Functions are $(1 - \Omega(1))$-Unpredictable

We move on to the case of general predicate and prove the following theorem:

Theorem 5.1 (**one-way $\Rightarrow$ $(1 - \Omega(1))$-UG**). *For every constants $\varepsilon > 0$ and $d \in \mathbb{N}$, there exists a constant $c > 0$ such that the following holds. For every predicate $Q : \{0,1\}^d \to \{0,1\}$ and $m > cn$ if the collection $\mathcal{F}_{Q,m}$ is ε-one-way then it is also ε'-UG for some constant $\varepsilon' < 1$. In particular, $\varepsilon' = 1 - \mathrm{Match}(Q)/2 + \Theta(\varepsilon)$.*

By Propositions 3.3 and 3.4 (part 1), we can replace next-bit unpredictability with last-bit unpredictability and standard one-wayness with approximate one-wayness. Hence, it suffices to prove the following:

Theorem 5.2 (**approximate one-way $\Rightarrow$ LUG**). *For every polynomial $m = m(n)$, constant $d \in \mathbb{N}$, predicate $Q : \{0,1\}^d \to \{0,1\}$, and constant $0 < \varepsilon < \mu = \mathrm{Match}(Q)$, if the collection $\mathcal{F}_{Q,m}$ is $(\varepsilon/4, \frac{1}{2} + \varepsilon/6)$ approximate-one-way then it is $(1 - \mu/2 + \varepsilon)$-last-bit unpredictable generator.*

Recall, that $\mu > 2^{-d}$ for a non-fixed predicate and $\mu > \Omega(\log d/d)$ if the predicate is balanced. The proof of the theorem is given in Section 5.1.

5.1 Proof of Thm. 5.2

Assume, towards a contradiction, that there exists a last-bit predictor $\mathbf{P}$ that guesses the last bit of $(G, f_{G,Q}(x))$ with probability $(1 - \mu/2 + \varepsilon)$. As in the previous section, it will be convenient to view the input of the predictor $\mathbf{P}$ as composed of an $(m-1, n, d)$ graph G, an $(m-1)$-bit string y (supposedly $y = f_{G,Q}(x)$), and an hyperedge S. Under this convention, $\mathbf{P}$ outputs its guess for $Q(x_S)$. To prove the theorem we will construct an $(\varepsilon/4, \frac{1}{2} + \varepsilon/6)$ approximate-inversion algorithm A. This algorithm is based on the subroutine Approx depicted in Figure 3.

We analyze the Algorithm Approx. In order to succeed we intuitively need two conditions. (1) Sensitivity: flipping the ℓ-th entry of x_S should change the value of the predicate Q; and (2) correctness: The predictor should predict well over many of the i's. We will prove that conditions of this spirit indeed guarantee success, and then argue that the conditions hold with good enough probability (taken over a random input and the random coins of the algorithm).

- **Input**: (n, m, d) graph G and string $y \in \{0,1\}^m$.
- **Randomness**: Choose uniformly at random a set $S = (i_1, \ldots, i_d)$, and an index $\ell \in [d]$, as well as random coins r for $\mathbf{P}$.

1. For every $i \in [n]$: Let $\hat{x}_i = \mathbf{P}(G, y, S_{\ell \leftarrow i}; r)$, where $S_{\ell \leftarrow i}$ is the set obtained by replacing the ℓ-th entry in S with the index i, and $\mathbf{P}$ is always invoked with the same fixed sequence of coins r.
2. Output the candidate $\hat{x}$ and its complement.

Figure 3: Algorithm Approx.

We begin by formalizing these conditions. We say that the tuple (x, G, r, S, ℓ) is *good* if the following two conditions hold

$$Q(x_S) \quad \neq \quad Q(x_S^{\oplus \ell}) \tag{4}$$

where $z^{\oplus i}$ denotes the string z with its i-th bit flipped, and, in addition, for at least $(\frac{1}{2} + \varepsilon/6)$ fraction of the $i \in [n]$

$$\mathbf{P}(G, f_{G,Q}(x), S_{\ell \leftarrow i}; r) = Q(x_{S_{\ell \leftarrow i}}). \tag{5}$$

It is not hard to see that a good tuple leads to a good approximation of x.

Lemma 5.3. *If the tuple (x, G, r, S, ℓ) is good then either $\hat{x}$ or its complement agrees with x for a fraction of $(\frac{1}{2} + \varepsilon/6)$ of the indices.*

Proof. Let j_ℓ be the ℓ-th entry of S. Then, by Eq. 4, we can write

$$Q(x_{S_{\ell \leftarrow i}}) = Q(x_S) \oplus x_{j_\ell} \oplus x_i.$$

Hence, for every $i \in [n]$ for which Eq. 5 holds we have that

$$\hat{x}_i = \mathbf{P}(G, y, S_{\ell \leftarrow i}; r) = Q(x_{S_{\ell \leftarrow i}}) = Q(x_S) \oplus x_{j_\ell} \oplus x_i = b \oplus x_i,$$

where $b = Q(x_S) \oplus x_{j_\ell}$. Hence, if $b = 0$ the output $\hat{x}$ agrees with x on a fraction of $(\frac{1}{2} + \varepsilon/6)$ of its coordinates, and otherwise, the complement $1 - \hat{x}$ has such an agreement. □

In the next section, we will prove that for many of the triples (x, G, r), a randomly chosen (S, ℓ) forms a good tuple with probability $\Omega(\varepsilon\mu/d)$.

Lemma 5.4. *For at least $\varepsilon - \mathrm{neg}(n)$ fraction of the pairs (x, G), we have that*

$$\Pr_{S,\ell,r}[(x, G, r, S, \ell) \text{ is good}] > \Omega(\varepsilon\mu/d)). \tag{6}$$

We can now prove Thm. 5.2.

Proofof Thm. 5.2. Given an input G and a string $y = f_{G,Q}(x)$, invoke the algorithm Approx $t = O(\log(1/\varepsilon)d/(\varepsilon\mu))$ times each time with a randomly chosen coins, and output all the t candidates. We claim that with probability $\Omega(\varepsilon)$ the list contains a good candidate whose agreement with x is $(\frac{1}{2} + \varepsilon/6)n$. Indeed, let us condition on the event that the pair (G, x) satisfies Eq. 6, which, by Lemma 5.4, happens with probability at least $\varepsilon/2$. In this case, by Lemmas 5.3 and 5.4, in each iteration we will output with probability $\Omega(\varepsilon\mu/d)$ a good candidate whose agreement with x is $(\frac{1}{2} + \varepsilon/6)n$. Since the success probability of each iteration is independent of the others, at least one iteration succeeds with probability $1 - \varepsilon/4$, and so, by a union bound, the overall success probability is $\varepsilon/2 - \varepsilon/4 = \varepsilon/4$. $\square$

5.2 Proof of Lemma 5.4

Call x *balanced* if $\mathrm{wt}(x) \in (n/2 \pm n^{2/3})$. We call a triple (x, G, r) *good* if x is balanced and

$$\Pr_S[\mathbf{P}(G, f_{G,Q}(x), S; r) = Q(x_S)] > 1 - \mu/2 + \varepsilon/2. \tag{7}$$

Claim 5.5. *A random triple (x, G, r) is good with probability $\varepsilon - \mathrm{neg}(n)$.*

Proof. By our assumption on $\mathbf{P}$ we have that

$$\Pr_{G,S,x,r}[\mathbf{P}(G, f_{G,Q}(x), S; r) = Q(x_S)] > 1 - \mu/2 + \varepsilon.$$

Hence, by Markov's inequality and the fact that $\varepsilon < \mu$,

$$\Pr_{G,x,r}[(x, G) \text{ satisfy Eq. 7}] > \varepsilon/(\mu - \varepsilon) > \varepsilon.$$

Finally, by a Chernoff bound, a random x is balanced with probability $1 - \mathrm{neg}(n)$, and so can write

$$\Pr_{G,x \text{ is balanced},r}[(x, G) \text{ satisfy Eq. 7}] > \varepsilon - \mathrm{neg}(n),$$

and the claim follows. $\square$

Fix a good triple (x, G, r). We call S *predictable* if $\mathbf{P}(G, f_{G,Q}(x), S; r) = Q(x_S)$. The pair (S, ℓ) is *neighborhood predictable* (in short NP) if

$$\Pr_{i \in [n]}[S_{\ell \leftarrow i} \text{ is predictable}] > \frac{1}{2} + \varepsilon/6 \tag{8}$$

The pair (S, ℓ) is *sensitive* if $Q(x_S) \neq Q(x_S^{\oplus \ell})$. To prove Lemma 5.4 it suffices to show that

Lemma 5.6. *A fraction of at least $\frac{\varepsilon\mu}{3d} \cdot (1 - o(1))$ of the pairs (S, ℓ) are simultaneously sensitive and neighborhood predictable.*

Proof. We will need some definitions. For a set S let $x_S \in \{0, 1\}^d$ be the "label" of the set. Fix some maximal matching M of the predicate Q whose cardinality is $\mu 2^d$. We restrict our attention to sets S for which x_S appears in some edge in M. (Abusing notation, we write $x_S \in M$.) For such S, we define the *index* $\ell(S)$ to be the single integer $\ell \in [d]$ for which the pair $(x_S, x_S^{\oplus \ell})$ is an edge in

M. (Since M is a matching, S will have exactly one index.) Observe that, by definition, the pair $(S, \ell(S))$ is always sensitive. Hence, to prove the lemma, it suffices to show that the quantity

$$\Pr_{S,\ell}[x_S \in M \wedge \ell = \ell(S) \wedge (S, \ell(S)) \text{ is NP}] \geq \Pr_S[x_S \in M] \cdot \Pr_{\ell \xleftarrow{R} [d]}[\ell = \ell(S)] \cdot \Pr_{S \text{ s.t. } x_S \in M}[(S, \ell(S)) \text{ is NP}]$$

is lower bounded by $\frac{\varepsilon\mu}{3d} \cdot (1 - o(1))$. Since $\Pr_{\ell \xleftarrow{R} [d]}[\ell = \ell(S)] = 1/d$ it suffices to show that

$$\begin{aligned} \Pr_S[x_S \in M] &> \mu(1 - o(1)) && (9) \\ \Pr_{S \text{ s.t. } x_S \in M}[(S, \ell(S)) \text{ is NP}] &> \varepsilon/3. && (10) \end{aligned}$$

We begin with Eq. 9. Since x is balanced, the label x_S of a random set S is almost uniform over the set of d-bit strings, and therefore it hits M with probability close to its density μ. Formally, for a label $z \in \{0,1\}^d$, let p_z denote the probability that a random set S is labeled by z. Note that p_z depends only in the Hamming weight of z (and x). In particular, since x is balanced and d is small ($d < o(n^{1/3})$), we have

$$p_z \in \left[\left(\frac{n/2 - n^{2/3} - d}{n} \right)^d, \left(\frac{n/2 + n^{2/3}}{n - d} \right)^d \right] = [2^{-d} - o(1), 2^{-d} + o(1)],$$

and so Eq. 9 is derived via

$$\Pr_S[x_S \in M] = \sum_{z \in M} p_z = \left(\mu 2^d \cdot 2^{-d}(1 \pm o(1)) \right) = \mu(1 \pm o(1)).$$

From now on, we focus on proving Eq. 10. Define a directed graph H over d-size sets S for which $x_S \in M$, and for each node S and $i \in [n]$ add an edge $(S, S_{\ell(S) \leftarrow i})$. In these terms, Eq. 10 asserts that at least $\varepsilon/3$ of the nodes S in H, have many $(\frac{1}{2} + \varepsilon/6)$ predictable out-neighbors. By Markov's inequality, (10) follows from the following equation

$$\Pr_{S \text{ s.t. } x_S \in M, i \xleftarrow{R} [n], T = S_{\ell(S) \leftarrow i}}[T \text{ is predictable}] > \frac{1}{2} + \varepsilon/3. \tag{11}$$

We prove (11) in two steps: First (Claim 5.7), we argue that the graph H is symmetric and regular. Therefore, the set T, obtained by choosing a random starting point S and taking a single random step in H, is uniformly distributed over H; Second (Claim 5.8), we show that $\frac{1}{2} + \varepsilon/3$ of the nodes in H are predictable. By combining these facts together we derive Eq.11.

Claim 5.7. *The graph H is symmetric and each node has exactly n distinct neighbors including one self loop.*

Proof. We show that the graph is symmetric. Fix an edge $(S, T = S_{\ell \leftarrow i})$ where $x_S = z$ and ℓ be the index of S, i.e., $(z, z^{\oplus \ell})$ is an edge in M. We claim that ℓ is also the index of T. Indeed, by definition x_T is either z or $z^{\oplus \ell}$ and therefore T's index is ℓ. It follows that for every j the pair $(T, T_{\ell \leftarrow j})$ is also an edge in H and by taking j to be the ℓ-th entry of S we get that $(T, T_{\ell \leftarrow j} = S)$ is an edge. The rest of the claim follows directly from the definition of H. $\square$

In fact, it is not hard to verify that the edges form an equivalence relation and therefore the graph is composed of vertex-disjoint union of n-sized cliques. We now show that $\frac{1}{2}+\varepsilon/3$ of the nodes in H are predictable.

Claim 5.8. $\Pr_{S \text{ s.t. } x_S \in M}[S \textit{ is predictable}] > \frac{1}{2}+\varepsilon/3$.

Proof. By Bayes' theorem and the goodness of (x, G, r) we have

$$\begin{aligned}1-\mu/2+\varepsilon &< \Pr_S[S \text{ is predictable}]\\ &= \Pr_S[x_S \notin M]\cdot \Pr_{S \text{ s.t. } x_S\notin M}[S \text{ is predictable}] + \Pr_S[x_S \in M]\cdot \Pr_{S \text{ s.t. } x_S\in M}[S \text{ is predictable}],\end{aligned}$$

by rearranging the equation and by noting that $\Pr_{S \text{ s.t. } x_S\notin M}[S \text{ is predictable}]$ is at most 1, we get

$$\begin{aligned}\Pr_{S \text{ s.t. } x_S\in M}[S \text{ is predictable}] &> \left(\Pr_S[S \text{ is predictable}] - \Pr_S[x_S \notin M]\right)\cdot \frac{1}{\Pr_S[x_S\in M]}\\ &> \frac{1-\mu/2+\varepsilon-1+\Pr_S[x_S\in M]}{\Pr_S[x_S \in M]}.\end{aligned}$$

Recall that $\Pr_S[x_S \in M] = (\mu \pm o(1))$, hence, we conclude that

$$\begin{aligned}\Pr_{S \text{ s.t. } x_S\in M}[S \text{ is predictable}] &> \frac{1-\mu/2+\varepsilon-1+\mu-o(1)}{\mu+o(1)}\\ &= \frac{\mu/2+\varepsilon/2-o(1)}{\mu+o(1)}\\ &> \frac{1}{2}+\varepsilon/2-o(1),\end{aligned}$$

and the claim follows. □

Eq. 11 now follows from Claims 5.7 and 5.8. This completes the proof of Lemma 5.6. □

6 From Unpredictability to Pseudorandomness

We will prove Theorems 1.2, 1.3, and 1.4 by combining our "one-wayness to unpredictability" reductions (proved in Sections 5 and 4) with several generic transformations.

First we will need the well-known theorem of Yao [40] which shows that sufficiently strong unpredictability leads to pseudorandomness:

Fact 6.1 (Good UG $\Rightarrow$ PRG)**.** *A UG of output length $m(n)$ and unpredictability of $\frac{1}{2}+\varepsilon$, is a PRG with $m\cdot\varepsilon$ pseudorandomness.*

By combining this fact with Thm. 4.1 we obtain Thm. 1.4:

Corollary 6.2 (Thm. 1.4 restated)**.** *For every constant d, sensitive predicate $Q:\{0,1\}^d \to \{0,1\}$, length parameter $m(n)\geq n$, and an inverse polynomial $\delta(n)\in(0,1)$, if $\mathcal{F}_{Q,m^3/\delta^2}$ is one-way then $\mathcal{F}_{Q,m}$ is $c\delta$-pseudorandom, for some constant $c=c(d)>0$.*

Proof. By Thm. 4.1, if $\mathcal{F}_{Q,m^3/\delta^2}$ is one-way then $\mathcal{F}_{Q,m}$ is $(\frac{1}{2} + \Omega_d(\delta/m))$-unpredictable, and by Yao's theorem (Fact 6.1) the latter is $\Omega_d(\delta)$-pseudorandom. □

Recall that in Thm. 5.1 we showed that for constant ε and sufficiently large $m = \Omega(n)$ if $\mathcal{F}_{Q,m}$ is ε-one-way then it is also ε'-unpredictable for some related constant $\varepsilon' < 1$. We would like to use this theorem to obtain a linear stretch PRG. However, in this case Yao's theorem (Fact 6.1) is useless as we have only constant unpredictability. For this setting of parameters we give an alternative new $\mathbf{NC^0}$ transformation from UG to PRG which preserves linear stretch.

Theorem 6.3. *For every constant $0 < \varepsilon < \frac{1}{2}$, there exists a constant $c > 0$ such that any $\mathbf{NC^0}$ unpredictable generator $G : \{0,1\}^n \to \{0,1\}^{cn}$ which is $(\frac{1}{2} + \varepsilon)$-unpredictable, can be transformed into an $\mathbf{NC^0}$ pseudorandom generator with linear stretch (e.g., that maps n bits to $2n$ bits) and negligible distinguishing advantage.*

The theorem is proved by combining the techniques of [26] with non-trivial $\mathbf{NC^0}$ randomness extractors from [7]. The proof of this theorem is deferred to Section 6.1.

As a corollary of the above theorem and Thm. 5.1 we get:

Corollary 6.4 (Thm. 1.2 restated)**.** *Let $d \in \mathbb{N}$ be an arbitrary constant and $Q : \{0,1\}^d \to \{0,1\}$ be a predicate. Then there exists a constant $c = c_d$ such that if $\mathcal{F}_{Q,cn}$ is $\frac{1}{2}$-one-way then there exists a collection of PRGs which doubles its input in $\mathbf{NC^0}$.*

We mention that by standard techniques (see Fact 6.5 below), we can obtain any fixed linear stretch at the expense of increasing the locality to a different constant.

We will now show that for sensitive Q if $\mathcal{F}_{Q,n^{1+\delta}}$ is one-way then one get get arbitrary polynomial stretch and arbitrary (fixed) inverse polynomial security in $\mathbf{NC^0}$ (i.e., prove Thm. 1.3). For this purpose, we will need the following amplification procedures (together with Thm. 4.1):

Fact 6.5 (Amplifying unpredictability and stretch)**.** *For every polynomials $t = t(n)$ and $s = s(n)$:*

1. *A d-local UG $G : \{0,1\}^n \to \{0,1\}^{m(n)}$ with unpredictability of $\frac{1}{2} + \varepsilon(n)$, can be transformed into a (td)-local UG $G' : \{0,1\}^{n \cdot t} \to \{0,1\}^{m(n)}$ with unpredictability of $\varepsilon' = (\varepsilon(n))^{\Omega(t)} + \mathrm{neg}(n)$.*
2. *A d-local PRG $G : \{0,1\}^n \to \{0,1\}^{n^b}$ with pseudorandomness $\varepsilon(n)$, can be transformed into a (d^s)-local PRG $G' : \{0,1\}^n \to \{0,1\}^{n^{(b^s)}}$ with pseudorandomness $s\varepsilon(n)$.*

The above fact also holds with respect to collections. The first part is based on Yao's XOR-lemma, and may be considered to be a folklore, and the second part is based on standard composition. A proof is given in Section A for completeness.

We can prove Thm. 1.3.

Corollary 6.6 (Thm. 1.3 restated)**.** *For every constant d, sensitive predicate $Q : \{0,1\}^d \to \{0,1\}$, and constant $\delta > 0$. If $\mathcal{F}_{Q,n^{1+\delta}}$ is one-way then for every stretch parameter $1 < a < O(1)$ and security parameter $1 < b < o(n)$ there exists a collection of PRGs of output length n^a and pseudorandomness of $1/n^b + \mathrm{neg}(n)$ with locality $d' = (bd/\delta)^{O(\frac{\lg a}{\delta})}$.*

Note that for fixed $\delta > 0$, we can have PPRG with arbitrary fixed polynomial stretch and security with constant locality. Alternatively, by setting $b = b(n) = \omega(1)$ (e.g., $b = \log^*(n)$), we get a standard PPRG with slightly super constant locality.

Proof. Fix d, Q and δ, and assume that $\mathcal{F}_{Q,n^{1+\delta}}$ is one-way. With out loss of generality, $\delta \leq 1$. Then, by Thm. 4.1, $\mathcal{F}_{Q,n^{1+\delta/4}}$ is $(\frac{1}{2} + n^{-\delta/4})$-unpredictable. We can now amplify unpredictability via Fact 6.5, part 1.

Specifically, by taking $t = \Omega(b/\delta)$ we get a new generator G with input length $\ell = tn$, output length $n^{1+\delta/4} = \ell^{1+\delta/5}$, locality td and unpredictability of $n^{-(b+4)} = \ell^{-(b+3)}$. By Yao's theorem (Fact 6.1) the resulting collection is pseudorandom with security $\ell^{-(b+3)} \cdot \ell^{1+\delta/5} = \ell^{-(b+1)}$ (as $\delta \leq 1$).

Finally, increase the stretch of the PRG by applying s-composition (Fact 6.5, part 2), for $s = \lg(a)/\lg(1 + \delta/5)$. This leads to a PRG which stretches ℓ-bits to $\ell^{(1+\delta/5)^s} = \ell^a$ bits, with pseudorandomness of $s \cdot \ell^{-(b+1)} < \ell^{-b}$, and locality of $(td)^s = (bd/\delta)^{O(\frac{\lg a}{\delta})}$. $\square$

6.1 Proof of Thm. 6.3

We will prove the following weaker version of Thm. 6.3.

Theorem 6.7. *There exist constants $0 < \varepsilon_0 < \frac{1}{2}$ and $c_0 > 0$ such that if there exists an* $\mathbf{NC^0}$ *UG (resp., collection of UG) $G : \{0,1\}^n \to \{0,1\}^{c_0 n}$ which is $(\frac{1}{2} + \varepsilon_0)$-unpredictable, then there exists an* $\mathbf{NC^0}$ *PRG (resp., collection of PRG) with linear stretch.*

Note that this version implies Thm. 6.3, as for any fixed $\varepsilon > 0$ given $(\frac{1}{2} + \varepsilon)$-unpredictable generator $G : \{0,1\}^n \to \{0,1\}^{cn}$ with sufficiently large constant $c = c_\varepsilon$, we can amplify unpredictability (via Fact 6.5, part 2) and obtain a new UG in $\mathbf{NC^0}$ and unpredictability of $(\frac{1}{2} + \varepsilon_0)$ and stretch $c_0 n$.

To prove the theorem we will employ $\mathbf{NC^0}$ randomness extractors.

Extractors. The *min-entropy* of a random variable X is at least k if for every element x in the support of X we have that $\Pr[X = x] \leq 2^{-k}$. A mapping $\mathrm{Ext} : \{0,1\}^\ell \times \{0,1\}^n \to \{0,1\}^N$ is a (k, Δ) *randomness extractor* (or extractor in short), if for every random variable X over $\{0,1\}^n$ with min-entropy of k, we have that $\mathrm{Ext}(\mathcal{U}_\ell, X)$ is Δ statistically-close to the uniform distribution. We refer to Δ as the *extraction error*, and to the first argument of the extractor as the *seed*. We typically write $\mathrm{Ext}_r(x)$ to denote $\mathrm{Ext}(r, x)$. We will use the following fact which follows by combining Lemma 5.7 and Thm. 5.12 of [7]:

Fact 6.8 (Non-trivial extractors in $\mathbf{NC^0}$). *For some constants $\alpha, \beta < 1$ there exists an* $\mathbf{NC^0}$ *extractor* Ext *that extracts n bits from random sources of length n and min-entropy $\alpha \cdot n$, by using a seed of length βn. Furthermore, the error of this extractor is exponentially small in n.*

We mention that the extractor is relatively weak ($\alpha + \beta > 1$), however, it will be sufficient for our purposes.

Construction 6.9. *Let $G : \{0,1\}^n \to \{0,1\}^{cn}$ be a UG, and* $\mathrm{Ext} : \{0,1\}^{\beta n} \times \{0,1\}^n \to \{0,1\}^n$ *be the extractor of Fact 6.8. We define the following function $H : \{0,1\}^{n^2(1+c\beta)} \to \{0,1\}^{cn^2}$ as follows.*

- *Input: n independent seeds $x = (x^{(1)}, \ldots, x^{(n)}) \in (\{0,1\}^n)^n$ for the generator, and cn independent seeds for the extractor $z = (z^{(1)}, \ldots, z^{(cn)}) \in (\{0,1\}^{\beta n})^{cn}$.*
- *Output: Compute the $n \times cn$ matrix Y whose i-th row is $G(x^{(i)})$. Let Y_i denote the i-th* column *of Y, and output $(\mathrm{Ext}_{z^{(1)}}(Y_1), \ldots, \mathrm{Ext}_{z^{(cn)}}(Y_{cn}))$.*

Note that H has linear stretch if $c > 1/(1-\beta)$. Also, the locality of H is the product of the localities of G and Ext, and so it is constant. Let ε be a constant which is strictly smaller than $(1-\alpha)/2$.

Lemma 6.10. *If G is $(\frac{1}{2}+\varepsilon)$-unpredictable, then the mapping H is a pseudorandom generator.*

Proof. The proof follows (a special case of) the analysis of [26]. We sketch it here for completeness. First, by Proposition 4.8 of [26], we have that G being a $(\frac{1}{2}+\varepsilon)$-UG has next-bit pseudo-entropy in the following sense. For every sequence of efficiently computable index family $\{i_n\}$ and efficient distinguisher $\mathcal{A}$ there is a random binary variable W, jointly distributed with $G(\mathcal{U}_n)$, such that: (1) the Shannon entropy of W given the $i_n - 1$ prefix of $G(\mathcal{U}_n)$ is at least κ, where $\kappa = 1 - 2\varepsilon$; and (2) $\mathcal{A}$ cannot distinguish between $G(\mathcal{U}_n)_{[1..i_n]}$ and $(G(\mathcal{U}_n)_{[1..i_n-1]}, W)$ with more than negligible advantage, even when $\mathcal{A}$ is given an oracle which samples the joint distribution $(G(\mathcal{U}_n), W)$.

Then, we use Claim 5.3 of [26], to argue that the n-fold direct product $G^{(n)}$ which outputs the matrix Y (defined in Construction 6.9) has block pseudo-min-entropy of $n(\kappa - o(1))$ in the following sense. For every sequence of efficiently computable index family $\{i_n\}$ and efficient distinguisher $\mathcal{A}$ there is a random variable $W \in \{0,1\}^n$ jointly distributed with $Y = G^{(n)}(\mathcal{U}_{n\times n})$, such that: (1) the min-entropy of W given the first $i_n - 1$ columns of Y is at least $n(\kappa - o(1))$; and (2) $\mathcal{A}$ cannot distinguish between $Y_{[1..i_n]}$ and $(Y_{[1..i_n-1]}, W)$ with more than negligible advantage, even when $\mathcal{A}$ is given an oracle which samples the joint distribution (Y, W).

This means that for every family $\{i_n\}$ the distribution $(Y_{[1..i_n-1]}, \mathrm{Ext}_{\mathcal{U}_{\beta n}}(Y_{i_n}))$ is indistinguishable from $(Y_{[1..i_n-1]}, \mathcal{U}_n)$. Otherwise, an adversary $\mathcal{B}$ that contradicts this statement can be used to construct an adversary $\mathcal{A}$ which contradicts the previous claim. Specifically, $\mathcal{A}(M, v)$ chooses a random seed s for the extractor and invokes $\mathcal{B}$ on $(M, \mathrm{Ext}_s(v))$. If v is chosen from Y_{i_n} then $\mathcal{B}$ gets a sample from $(Y_{[1..i_n-1]}, \mathrm{Ext}_{\mathcal{U}_{\beta n}}(Y_{i_n}))$, and if v is chosen from W, $\mathcal{B}$ gets a sample from $(Y_{[1..i_n-1]}, \mathrm{Ext}_{\mathcal{U}_{\beta n}}(W))$ which is statistically close to $(Y_{[1..i_n-1]}, \mathcal{U}_n)$, as W has min-entropy of $n(\kappa - o(1)) > \alpha n$. Hence, $\mathcal{A}$ has the same distinguishing advantage as $\mathcal{B}$ (up to a negligible loss).

Finally, the above statement implies that for every family $\{i_n\}$ the distributions $H(\mathcal{U}_{n^2(1+c\beta)})_{[1..i_n]}$ is indistinguishable from $(H(\mathcal{U}_{n^2(1+c\beta)})_{[1..i_n-1]}, \mathcal{U}_1)$, and so H is $(\frac{1}{2}+\delta)$-unpredictable generator for negligible δ, and by Yao's theorem (Fact 6.1), it is pseudorandom. □

7 Inapproximability of the Densest-Subgraph Problem

We will prove the following theorem:

Theorem 7.1 (Thm. 1.5 restated)**.** *Assume that $\mathcal{F}_{m,Q}$ is $1/\log n$-pseudorandom where Q is a d-ary predicate and $m \geq n^{1+\delta}$ for some constant $\delta > 0$. Then for every $p \in (n^{-\frac{\delta}{d+2}}, \frac{1}{2})$ the p-Densest-Subhypergraph problem is intractable with respect to d-uniform hypergraphs.*

Note that the larger δ gets, the better inapproximaility ratio we obtain. Clearly, δ cannot be larger than $\delta(d)$ where $n^{1+\delta(d)}$ is the maximal stretch of d-local pseudorandom generators. For $d \geq 5$, the best upper-bound on $\delta(d)$ is roughly $d/2$ due to [36]. (For smaller d's, $m < O(n)$.)

From now on, we assume, without loss of generality, that $Q(1^d) = 1$, otherwise we can negate it, and use $1 - Q$ as our predicate. (It is not hard to see that pseudorandomness still holds.) Let p the parameter chosen in Theorem 7.1 and assume that it is a power of two, namely, $p = 2^{-t}$ for some integer $t \in (1, \frac{\delta \log n}{d+2}$. We define an operator ρ as follows. Given an (m, n, d) graph G, and a

$t \times m$ binary matrix $Y \in \{0,1\}^{t\times m}$, we view the i-th column of Y as a t-bit label for the i-th edge of G. Then, the operator $\rho(G,Y)$ outputs the (m',n,d) subgraph G' whose edges are those edges of G which are indexed under Y by the all-one string 1^t.

We construct a pair of distributions D_{yes} and D_{no} over hypergraphs which are indistinguishable, but D_{yes} (resp., D_{no}) outputs whp a yes instance (resp., no instance):

- **The distribution D_{no}.** Choose a random (m,n,d) graph G, and a random $t \times m$ binary matrix $Y \xleftarrow{R} \mathcal{U}_{t\times m}$. Output the subgraph $G' = \rho(G,Y)$.
- **The distribution D_{yes}.** Choose a random (m,n,d) graph G, and a random $t \times n$ binary matrix $X \xleftarrow{R} \mathcal{U}_{t\times n}$. Let $x^{(i)}$ be the i-th row of X, and define a $t \times m$ binary matrix Y whose i-th row is $f_{G,Q}(x^{(i)})$. Output the subgraph $G' = \rho(G,Y)$.

It is not hard to show that D_{no} and D_{yes} are weakly-indistinguishable.

Lemma 7.2. *If $\mathcal{F}_{m,Q}$ is ε-pseudorandom then the ensembles D_{no} and D_{yes} (indexed by n) cannot be distinguished with advantage better than $t\varepsilon$, which, under our choice of parameters, is at most $\frac{1}{2} - 1/(d+2)$.*

Proof. A $t\varepsilon$-distinguisher immediately leads to a $t\varepsilon$-distinguisher between the distributions

$$(G, y^{(1)}, \ldots, y^{(t)})$$

and

$$(G, f_{G,Q}(x^{(1)}), \ldots, f_{G,Q}(x^{(t)}))$$

where G is a random (m,n,d) graph, the y's are random m-bit strings and the x's are random n-bit strings. By a standard hybrid argument this leads to an ε distinguisher for $\mathcal{F}_{m,Q}$. Finally, observe that since $p < 1/n$ (as δ is upper-bounded by $d/2$) we have that $t\varepsilon < \frac{d \log n}{2(d+2)\log n} = \frac{1}{2} - 1/(d+2)$. ☐

Let us analyze D_{no}. Since Y and G are independent, we can redefine D_{no} as follows: (1) choose Y uniformly at random, (2) determine which of the columns of Y equal to the all one string, and (3) then choose the corresponding hyperedge uniformly at random. Hence, G' is just a random $\mathcal{G}_{m',n,d}$ graph where m' is sampled from the binomial distribution $\mathrm{Bin}(p,m)$, where $p = 2^{-t}$. Therefore, standard calculations show that

Lemma 7.3. *With all but negligible probability, the graph G' chosen from D_{no} satisfies the following: (1) It has $m' = mp(1 \pm 1/\log n)$ edges; and (2) Every set S of nodes of density p contains at most $p^d(1+o(1))$ fraction of the edges.*

Proof. The first item follows from a multiplicative Chernoff bound: define m independent Bernoulli random variables, where the i-th variable is 1 if the i-th hyperedge is chosen. Since each random variable succeeds with probability p, the probability of having $m' = (1 \pm 1/\log n)pm$ successes is at least $1 - \exp(-\Omega(mp/\log^2 n)) > 1 - \exp(-\Omega(n))$.

To prove the second item, let us condition on the event $m' > pm/2$, which by the previous argument happens w/p $1 - \mathrm{neg}(n)$. Fix such an m', and let $G' \xleftarrow{R} \mathcal{G}_{m',n,d}$. Consider a fixed set of nodes S of size pn in G'. Every edge of G' falls in S with probability p^d. Hence, by a multiplicative Chernoff bound, the probability that S contains a set of edges of density $p^d(1+1/\log n)$ is bounded by $\exp(-\Omega(p^d m'/\log^2 n)) < \exp(-\Omega(n^{1+\frac{\delta}{d+2}}/\log^2 n)) < \exp(-2n)$. Therefore, by a union bound, the probability that this happens for some set S is at most $\exp(-2n+n) = \mathrm{neg}(n)$. ☐

On the other hand, we prove that D_{yes} has a planted *dense* sub-graph.

Lemma 7.4. *With probability at least* $\frac{1}{2} + 1/(d+2) - \text{neg}(n)$*, a graph* G' *chosen from* D_{yes} *has a set of nodes* S *of density* p *that contains a fraction of at least* $p^{d-1}(1-o(1))$ *edges.*

Proof. Label each *node* of G by the corresponding t-bit column of the matrix X, and let T be the set of nodes which are labeled by the all-one string. Let $S \subseteq T$ be the lexicographically first pn nodes of T; if $|T| \leq pn$ then $S = T$.

Consider the following event E in which: (1) S is of density at least $q = p(1 - 1/\log n)$; (2) At least $q^d(1 - 1/\log n)$ fraction of the edges of the original graph G fall in S; (3) The number of remaining edges m' in G' is at most $pm(1 + 1/\log n)$.

The main observation is that edges which fall into S are labeled by the all-one strings as $Q(1^d) = 1$, and so they also appear in G'. Hence, if E happens, then in G' the q-dense set of nodes S contains a set of edges of density at least

$$\frac{q^d(1 - 1/\log n)m}{m'} \geq \frac{p^d m(1 - 1/\log n)^{d+1}}{pm(1 + 1/\log n)} > p^{d-1}(1 - o(1)).$$

Since, $q \leq p$ we can always pad S with additional nodes and obtain a p dense set S' which contains $p^{d-1}(1 - o(1))m'$ edges as required.

It remains to show that the event E happens with probability $\frac{1}{2} + 1/(d+2) - \text{neg}(n)$. First, since each node falls in T independently with probability p, it follows from a multiplicative Chernoff bound, that T contains at least $p(1 - 1/\log n)n$ nodes with all but negligible probability $\exp(-\Omega(pn/\log^2 n)) = \exp(-n^{\Omega(1)})$. (Recall that $\delta < d/2$ and therefore $pn = n^{\Omega(1)}$.) Hence, the sub-event (1) holds with all but negligible probability. Conditioned on (1), the sub-event (2) corresponds to having at least $q^d(1 - 1/\log n)$ fraction of the hyper-edges fall into a set S of density q. Since each edge falls in S independently with probability q^d, (2) happens with all but $\exp(-\Omega(q^d m/\log^2 n)) < \text{neg}(n)$ probability (due to a multiplicative Chernoff bound). Hence, (1) and (2) happen simultaneously w/p $1 - \text{neg}(n)$.

Finally, we argue that the probability β that (3) holds is at least $1 - \text{neg}(n) - t \cdot \varepsilon = \frac{1}{2} + 1/(d+2) - \text{neg}(n)$. Indeed, consider the algorithm which attempts to distinguish D_{no} from D_{yes} by looking at m' and accepting if and only if $m' \leq (p + 1/\log n)m$. By Lemma 7.3 this leads to a distinguisher with advantage $1 - \text{neg}(n) - \beta$, which, by Lemma 7.2, can be at most $t \cdot \varepsilon$.

To complete the proof, observe, that, by a union bound, we have that (3) holds together with (1) and (2) with probability $\frac{1}{2} + 1/(d+2) - \text{neg}(n)$. □

By Lemmas 7.3 and 7.4 an algorithm that solves $p - \mathsf{DSH}$ for d-uniform hypergraphs can distinguish between the two distributions with advantage at least $(1 - \text{neg}(n)) - (\frac{1}{2} - 1/(d+2) + \text{neg}(n)) > \frac{1}{2}$, contradicting Lemma 7.2. Hence, Thm. 7.1 is derived. □

Acknowledgement. The author is grateful to Oded Goldreich for closely accompanying this research, and for countless insightful comments and conversations that significantly affected the results of this paper. We also thank Uri Feige, Yuval Ishai and Alex Samorodnitsky for many valuable discussions.

References

[1] D. Achlioptas. *Handbook of Satisfiability*, chapter Random Satisfiability, pages 243–268. IOS Press, 2009.

[2] M. Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307. IEEE Computer Society, 2003.

[3] M. Alekhnovich, E. A. Hirsch, and D. Itsykson. Exponential lower bounds for the running time of DPLL algorithms on satisfiable formulas. *J. Autom. Reasoning*, 35(1-3):51–72, 2005.

[4] B. Applebaum, B. Barak, and A. Wigderson. Public-key cryptography from different assumptions. In *Proc. of 42nd STOC*, pages 171–180, 2010.

[5] B. Applebaum, A. Bogdanov, and A. Rosen. A dichotomy for local small-bias generators. In *Proc. of 9th TCC*, pages 1–18, 2012.

[6] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM Journal on Computing*, 36(4):845–888, 2006.

[7] B. Applebaum, Y. Ishai, and E. Kushilevitz. On pseudorandom generators with linear stretch in NC^0. *J. of Computational Complexity*, 17(1):38–69, 2008.

[8] S. Arora, B. Barak, M. Brunnermeier, and R. Ge. Computational complexity and information asymmetry in financial products. *Commun. ACM*, 54(5):101–107, 2011.

[9] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an $\mathrm{O}(n^{1/4})$ approximation for densest k-subgraph. In *Proc. of 42nd STOC*, pages 201–210, 2010.

[10] A. Bogdanov and Y. Qiao. On the security of goldreich's one-way function. In *Proc. of 13th RANDOM*, pages 392–405, 2009.

[11] A. Bogdanov and A. Rosen. Input locality and hardness amplification. In *Proc. of 8th TCC*, pages 1–18, 2011.

[12] A. Coja-Oghlan. Random constraint satisfaction problems. In *Proc. 5th DCM*, 2009.

[13] J. Cook, O. Etesami, R. Miller, and L. Trevisan. Goldreich's one-way function candidate and myopic backtracking algorithms. In *Proc. of 6th TCC*, pages 521–538, 2009.

[14] M. Cryan and P. B. Miltersen. On pseudorandom generators in NC^0. In *Proc. 26th MFCS*, 2001.

[15] N. Dedic, L. Reyzin, and S. P. Vadhan. An improved pseudorandom generator based on hardness of factoring. In *Proc. 3rd SCN*, 2002.

[16] B. den Boer. Diffie-hillman is as strong as discrete log for certain primes. In *Proc. of 8th CRYPTO*, pages 530–539, 1988.

[17] S. O. Etesami. Pseudorandomness against depth-2 circuits and analysis of goldreich's candidate one-way function. Technical Report EECS-2010-180, UC Berkeley, 2010.

[18] U. Feige. Relations between average case complexity and approximation complexity. In *Proc. of 34th STOC*, pages 534–543, 2002.

[19] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[20] A. Flaxman. Random planted 3-SAT. In M.-Y. Kao, editor, *Encyclopedia of Algorithms*. Springer, 2008.

[21] O. Goldreich. Candidate one-way functions based on expander graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(090), 2000.

[22] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[23] O. Goldreich, H. Krawczyk, and M. Luby. On the existence of pseudorandom generators. *SIAM J. Comput.*, 22(6):1163–1175, 1993. Preliminary version in Proc. 29th FOCS, 1988.

[24] O. Goldreich, N. Nisan, and A. Wigderson. On yao's XOR-lemma. *Electronic Colloquium on Computational Complexity (ECCC)*, 2(50), 1995.

[25] O. Goldreich and V. Rosen. On the security of modular exponentiation with application to the construction of pseudorandom generators. *J. Cryptology*, 16(2):71–93, 2003.

[26] I. Haitner, O. Reingold, and S. P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In *Proc. of 42nd STOC*, pages 437–446, 2010.

[27] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[28] Herzberg and Luby. Public randomness in cryptography. In *Proc. of 12th CRYPTO*, 1992.

[29] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography with constant computational overhead. In *Proc. of 40th STOC*, pages 433–442, 2008.

[30] D. Itsykson. Lower bound on average-case complexity of inversion of goldreich's function by drunken backtracking algorithms. In *Computer Science - Theory and Applications, 5th International Computer Science Symposium in Russia*, pages 204–215, 2010.

[31] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *Proc. of 29th FOCS*, pages 68–80, 1988.

[32] S. Khot. Ruling out PTAS for graph min-bisection, densest subgraph and bipartite clique. In *Proc. of 45th FOCS*, pages 136–145, 2004.

[33] L. A. Levin. One-way functions and pseudorandom generators. In *Proc. of 17th STOC*, pages 363–365, 1985.

[34] U. M. Maurer and S. Wolf. The relationship between breaking the diffie-hellman protocol and computing discrete logarithms. *SIAM J. Comput.*, 28(5):1689–1721, 1999.

[35] R. Miller. Goldreich's one-way function candidate and drunken backtracking algorithms. Distinguished major thesis, University of Virginia, 2009.

[36] E. Mossel, A. Shpilka, and L. Trevisan. On ϵ-biased generators in NC^0. In *Proc. 44th FOCS*, pages 136–145, 2003.

[37] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[38] N. Nisan and A. Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[39] S. K. Panjwani. An experimental evaluation of goldreich's one-way function. Technical report, IIT, Bombay, 2001.

[40] A. C. Yao. Theory and application of trapdoor functions. In *Proc. 23rd FOCS*, pages 80–91, 1982.

A Omitted proofs

A.1 Amplifying unpredictability and stretch

We will prove Fact 6.5.

Part 1: unpredictability amplification. Define the UG collection $F^{t\oplus} : \{0,1\}^{st} \times \{0,1\}^{nt} \to \{0,1\}^m$ to be the bit-wise xor of t independent copies of F, i.e., for $k_1, \ldots, k_t \in \{0,1\}^s$ and $x_1, \ldots, x_t \in \{0,1\}^n$ let $F^{t\oplus}_{k_1,\ldots,k_t}(x_1, \ldots, x_t) = F_{k_1}(x_1) \oplus \ldots \oplus F_{k_t}(x_t)$.

Fix some $t = t(n)$, and assume, towards a contradiction, that there exists an algorithm $\mathcal{A}$ and a sequence of indices $\{i_n\}$ such that

$$\Pr[\mathcal{A}(Y^{t(\oplus)}_{[1..i_n-1]}) = Y^{t(\oplus)}_{i_n}] > \frac{1}{2} + \delta,$$

for infinitely many m's and $\delta = \varepsilon^{\Omega(t)} + \mathrm{neg}(n)$. Then, there exists another adversary $\mathcal{A}'$

$$\Pr[\mathcal{A}'(Y^{(1)}_{[1..i_n-1]}, \ldots, Y^{(t))}_{[1..i_n-1]}) = Y^{t(\oplus)}_{i_n}] > \frac{1}{2} + \delta,$$

for the same input lengths. Define a randomized predicate P_n which given an $i_n - 1$ bit string y samples a bit b from the conditional distribution $Y_m | Y_{1..i_n-1} = y$. Then, the last equation can be rewritten as

$$\Pr[\mathcal{A}'(y^{(1)}, \ldots, y^{(t)}) = \bigoplus_{j\in[t]} P_n(y^{(j)})] > \frac{1}{2} + \delta,$$

where each $y^{(j)}$ is sampled uniformly and independently from $Y_{[1..i_n-1]}$. By Yao's XOR lemma (cf. [24]), such an efficient adversary $\mathcal{A}'$ implies an adversary $\mathcal{A}''$ for which

$$\Pr[\mathcal{A}''(Y_{[1..i_n-1]}) = P_n(Y_{[1..i_n-1]}) = Y_{i_n}] > \frac{1}{2} + \varepsilon,$$

for the same input lengths, in contradiction to the unpredictability of Y.

Uniformity. In order to apply the above argument in a fully uniform setting we should make sure that pairs $Y_{[1..i_n-1]}, Y_{i_n}$ are efficiently samplable. Since Y is efficiently samplable it suffices to show that the sequence $\{i_n\}$ is uniform, i.e., can be generated in time $\text{poly}(n)$. In fact, to get our bound, it suffices to have a uniform sequence $\{i'_n\}$ for which $\mathcal{A}$ achieves prediction probability of $\frac{1}{2} + \delta - \sqrt{\delta}$. Hence, we can use Remark 3.2.

Part 2: stretch amplification. Let G be the original collection of PRGs with key sampling algorithm K. We define the s-wise composition of G as follows. The collection $G^{(s)}_{\vec{k}}(x)$ is indexed by s-tuple of "original" indices $\vec{k} = (k_0, \ldots, k_s)$ where the i-th entry is sampled uniformly and independently by invoking the original index sampling generator K on $(1^{n^{(b^i)}})$. We define $G^{(0)}_{\vec{k}}(x)$ to be $G_{k_0}(x)$, and for every $i > 0$ we let $G^{(i)}_{\vec{k}}(x) = G_{k_i}(G^{(i-1)}_{\vec{k}}(x))$. Clearly, the resulting collection has output length of $n^{(b^s)}$ and locality d^s. A standard hybrid argument shows that the security is $s\varepsilon(n)$. (See [22, Chp. 3, Ex. 19].)

ECCC ISSN 1433-8092
http://eccc.hpi-web.de