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

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

Benny Applebaum*

January 17, 2011

Abstract

We continue the study of pseudorandom generators (PRG) $G : \{0,1\}^n \to \{0,1\}^m$ in $\mathbf{NC^0}$. While it is known that such generators are likely to exist for the case of small sub-linear stretch $m = n + n^{1-\varepsilon}$, it remains unclear whether achieving larger stretch such as $m = 2n$ or even $m = n + n^2$ 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 $\mathbf{NC^0}$ constructions of linear-stretch PRGs and polynomial-stretch weak-PRGs (where the distinguishing advantage is $1/\text{poly}(n)$ rather than negligible). These constructions are based on the one-wayness of "random" $\mathbf{NC^0}$ functions – a variant of an assumption made by Goldreich (ECCC 2000). Our techniques also show that some of the previous heuristic candidates can be based on one-way assumptions. We interpret these results as an evidence for the existence of $\mathbf{NC^0}$ PRGs of polynomially-long stretch.

We also 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, benny.applebaum@gmail.com. Work done while a postdoc at the Weizmann Institute of Science, supported by Koshland Fellowship.

ISSN 1433-8092

1 Introduction

The question of minimizing the parallel time complexity of cryptographic primitives has been the subject of an extensive body of research. (See, e.g., [2] and references therein.) At the extreme, one would aim for an ultimate level of efficiency at the form of *constant*-parallel time implementation. Namely, the goal is to have "local" cryptographic constructions in which each bit of the output depends only on a small constant number of input bits, and so, each output can be individually computed with complexity that does not grow with the total input length or the level of security; Such functions are 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. However, in the last few years, it was shown that, perhaps surprisingly, many cryptographic tasks can be implemented by $\mathbf{NC^0}$ functions under standard intractability assumptions [5, 4]. This includes basic primitives such as one-way functions, as well as highly complicated primitives such as public-key encryption schemes, digital signatures, and non-interactive zero-knowledge proofs.

A notable exception, for which such a result is unknown, is pseudorandom generators with *large* stretch. A pseudorandom generator (PRG) $G : \{0,1\}^n \to \{0,1\}^m$ is a deterministic function that expands a short random n-bit string (aka "seed") into a longer "pseudorandom" string of length $m > n$, such that the output cannot be distinguished from a truly random m-bit string with more than negligible probability. Our main focus with respect to PRGs is the difference between its output length m and its input length n, namely its *stretch* $m - n$. While it is known that PRGs with sub-linear stretch (i.e., $m = n + n^{1-\varepsilon}$) are likely to exist in $\mathbf{NC^0}$ [5], it is unclear whether better stretch is possible. Specifically, several previous works (e.g., [10, 21, 5, 6, 17]) posed the following question:

> How long can be the stretch of $\mathbf{NC^0}$ pseudorandom generators? Can it be linear, e.g., $m > (1+\varepsilon)n$, or even polynomial, e.g., $m > n^{1+\varepsilon}$?

This basic question is possibly the most important feasibility result left open in the field of parallel cryptography. It is also motivated by several concrete applications. For example, such PRGs would lead to highly efficient stream-ciphers that can be implemented by fast parallel hardware. They also lead to secure computation protocols with *constant* computational overhead [17] – a fascinating possibility which is not known to hold under any other cryptographic assumption. Finally, as shown in [6], such PRGs can be used to obtain strong (average-case) inapproximability results for constraint satisfaction problems such as Max3SAT, and thus provide simpler alternatives to the traditional PCP-based approach by relying on stronger assumptions.

Previous works. Despite previous efforts, there has been no convincing theoretical evidence supporting either a positive or a negative resolution of this question. On the negative side, it is well known [20] that *pseudorandom functions* (which can be viewed as PRGs with exponential stretch and direct access) cannot be implemented in $\mathbf{AC^0}$, which is strictly larger than $\mathbf{NC^0}$. However, this result does not extend to the case of polynomial stretch PRGs (PPRGs). Cryan and Miltersen [10] conjectured that PPRGs cannot be constructed in $\mathbf{NC^0}$, but were able to prove this only for the special case of 3-local functions (i.e., functions that each of their outputs depends on at most 3 inputs). This impossibility result was extended to 4-local functions by Mossel et al. [21]. For general locality d, the best upper-bound on the stretch is $n^{d/2}$ [21].

On the positive side, it is known that PRGs with *sub-linear* stretch are likely to exist in $\mathbf{NC^0}$ [5]; however, it is unknown how to expand the stretch of these PRGs without increasing their locality. Candidates for PPRGs were suggested in [21] and [3], and it was shown that these candidates resist some restricted non-trivial families of attacks (e.g., they generate ε-biased distribution [22]). However, none of them was proven to be secure via a reduction to a different, more established, cryptographic assumption. In fact, to the best of our knowledge, even the class $\mathbf{AC^0}$ does not contain any provably-secure PPRG, and only in $\mathbf{TC^0}$, which is strictly more powerful, such constructions are known to exist [23]. Finally, let us mention that even in the case of linear-stretch PRGs (LPRG), the only known $\mathbf{NC^0}$ construction [6] relies on an *indistinguishability* assumption (of [1]), rather than on a one-wayness assumption which is typically considered to be more conservative.

1.1 Our results

1.1.1 Constructions

We make progress towards resolving the question by obtaining local ($\mathbf{NC^0}$) constructions of LPRGs and weak-PPRGs, where in the latter the distinguishing advantage is upper-bounded by an arbitrary *fixed* inverse polynomial $1/n^{\delta}$ rather than negligible. Our constructions are based on the one-wayness of "random" local functions.

Random local one-way functions. For a length parameter $m = m(n)$ and a d-ary predicate $Q : \{0,1\}^d \to \{0,1\}$, we define the distribution $\mathcal{F}_{Q,m}$ over d-local functions $f : \{0,1\}^n \to \{0,1\}^m$ as follows: choose a random d-uniform hypergraph G with n nodes and m hyperedges by choosing each hyperedge uniformly and independently at random. (We refer to such graph as a random (m, n, d) graph). Then, define the d-local function $f = f_{G,Q} : \{0,1\}^n \to \{0,1\}^m$ to be the function whose i-th output is computed by applying the predicate Q to the d inputs that are indexed by the i-th hyperedge. We say that the predicate Q is *sensitive* if *some* of its coordinates i has full influence, that is flipping the value of the i-th variable always changes the output of Q.

Our main hardness assumption asserts that, for proper choice of predicate Q, a random member of the collection is hard to invert – technically, this means that $\mathcal{F}_{Q,m}$ is a collection of one-way functions [14, Sec. 2.4.2]. We will later discuss the plausibility of this assumption, but for now let us just mention that it was presented by Goldreich [13] for the case of $m = n$, and was further studied by several other works [24, 21, 9, 8, 3] for different ranges of parameters. We can now state our main theorem:

Theorem 1.1 (main theorem)**.** *Suppose that the d-local collection $\mathcal{F}_{Q,m}$ is one-way.*

1. ***(LPRG in $\mathbf{NC^0}$)** If $m > c \cdot n$ for some constant $c = c(d) > 1$, then there exists a collection of LPRGs in $\mathbf{NC^0}$.*
2. ***(weak-PPRG in $\mathbf{NC^0}$)** If $m > n^{1+\delta}$ for an arbitrary small constant $\delta > 0$ and Q is sensitive, then, for every constant b, there exists a weak collection of PPRGs of output length n^b and distinguishing gap at most $1/n^b$ with constant locality $d' = d'(d, b)$.*
3. ***(Random local functions are weak-PRGs)** If $m > n^{3a+2b}$ and Q is sensitive, then the collection $\mathcal{F}_{Q,n^a}$ is a weak-PPRG with distinguishing gap at most n^{-b}.*

The first item shows that LPRGs can be constructed based on the one-wayness of $\mathcal{F}_{Q,m}$ for an *arbitrary* predicate Q, and linear output length. The second item shows that the one-wayness of $\mathcal{F}_{Q,m}$ with super-linear output length and sensitive predicate, leads to a weak-PPRG in $\mathbf{NC^0}$ with an arbitrary fixed polynomial stretch and an arbitrary inverse fixed polynomial security. In fact, we will prove a more general trade-off between locality, stretch and security which allows to obtain a standard PPRG with an arbitrary polynomial stretch at the expense of letting the locality be an (arbitrarily slowly) increasing function of n, e.g., $d' = \log^*(n)$. Finally, observe that in the last item we show that $\mathcal{F}_{Q,m'}$ *itself* is weakly pseudorandom.

Let us elaborate on some of the aspects raised by this theorem.

Plausibility of the assumption. In general, $\mathcal{F}_{Q,m}$ becomes tractable when m is too large. In particular, it is not hard to see that the function can be efficiently inverted for $m = \Omega(n^d)$. On the other hand, 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 $2^{\Omega(n)}$.[1] It seems reasonable to assume that for every constant $c > 1$ there exists a sufficiently large locality d and a predicate Q for which $\mathcal{F}_{Q,n^c}$ cannot be inverted in polynomial time or even subexponential (i.e., $2^{n^{\varepsilon}}$) time. In fact, even for the case of, say, $m = n^{d/100}$, no polynomial-time inversion algorithm is known. We also mention that our theorems hold even if the level of one-wayness is quite weak (e.g., the collection cannot be inverted with probability better than $1/n$ or even $1/10$ in the first item). Finally, even if our assumptions may seem strong, the new results strictly improve the previous state, as all known heuristic candidates rely on stronger assumptions – i.e., on the *pseudorandomness* of random local functions. As a side note, we mention that our techniques also show that the security of some of these candidates (e.g., [21, 6]) can be based on one-way assumptions.

The gap between one-wayness and pseudorandomness. We would like to stress that there is a considerable gap between the hypothesis and the implication, since pseudorandomness is much more fragile than one-wayness. This is especially true in our local setting, as with low locality, even the task of avoiding simple regularities in the output is highly challenging.[2] In contrast, it seems much easier to find a "reasonable" candidate one-way functions (i.e., one that resists all basic/known attacks). The proof of the main theorem (and item 3 by itself), shows that in this case, despite the existence of non-trivial regularities in the outputs, one-way functions achieve some form of pseudoentropy (i.e., weak unpredictability).

Weak pseudorandomness. Our polynomial stretch PRGs are weak, i.e., their security is only inverse polynomial, rather than inverse super-polynomial as per the standard cryptographic definition. It is important to stress that this weakness refers only to the *distinguishing advantage* rather than to the *running-time* of the adversaries, which is super-polynomial or even sub-exponential, depending on the exact hardness of $\mathcal{F}_{Q,m}$ as a one-way function. (The overhead added by our reductions is minor.) Let us also mention that other heuristic candidates for $\mathbf{NC^0}$ PPRG suffer from a similar limitation. This can be partially explained by the fact that pseudorandomness requires the dependencies graph G to satisfy non-trivial expansion properties (see [6]), but when $m = n^{1+\Omega(1)}$

[1] In [8] it is shown that if the predicate satisfies some "simplicity" property, then it is possible to efficiently invert $\mathcal{F}_{Q,m}$ for $m > cn$ where $c = c(d)$ is some constant. However, nothing is known for general predicate.

[2] This even led to the belief that weak non-cryptographic forms of pseudorandomness, e.g., ε-bias, cannot be achieved [10], which was refuted in a non-trivial way by [21].

and $d = O(1)$ it is unknown how to sample such a good (m, n, d) expander with negligible failure probability. The lack of such explicit constructions currently forms a natural "barrier" towards realizing strong PPRGs with constant locality.[3]

The work of [3] (ABW). Our work is inspired by the recent results of [3] which showed a reduction from weak pseudorandomness to one-wayness for Goldreich's function instantiated with a specific "randomized" predicate. Specifically, this was shown for the case of the noisy-linear predicate $Q_{\oplus}(x_1, \ldots, x_d)$ which outputs the xor of the inputs and flips the result with some small probability. While this result has several interesting applications (see [3]), it falls short of providing polynomial-stretch PRGs with low locality due to the use of internal randomness. From a technical point of view, many of the ideas used in [3] heavily rely on the linear structure of $Q_{\oplus}$, and so part of the challenge in proving Thm. 1.1 is to find analogues which work in the general case of arbitrary predicates. (See Section 2 for an overview of our proofs.)

1.1.2 Hardness of the Densest-Subgraph Problem

Theorem 1.1 also leads to new inapproximability results. This continues the line of research started by Feige [11] 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 of e are in S, i.e., $v_1, \ldots, v_d \in S$. In the following think of d as a constant and $n < m < \text{poly}(n)$. For a parameter $p \in (0, 1)$, the p Densest-Sub-hypergraph Problem ($p - \mathsf{DSH}$) is the promise problem in which we are given an (m, n, d) graph G 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 ("Pseudorandom").** 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.

Observe that a random graph is likely to be a No-instance. In the above, p is a single parameter which 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 [19] (under the term "Quasi-random PCP"), and was implicit in the work of Feige [11]. These works showed that for some constant d, the problem is hard with $p = \frac{1}{2}$, 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. We prove the following result:

Theorem 1.2. *Let d be a constant, Q be a d-ary predicate and $m \geq n^{c+3}$ where $c > 0$ is a constant. If $\mathcal{F}_{m,Q}$ is $\frac{1}{n}$-pseudorandom, then for every $n^{-c/2d} \leq p \leq \frac{1}{2}$ the p-Densest-Subhypergraph problem is intractable with respect to d-uniform hypergraphs.*[4]

[3]A secondary limitation of our results is the fact that they yield $\mathbf{NC^0}$ *collections* rather than single *explicit* $\mathbf{NC^0}$ function. This is a minor technical issue with no actual effect on the applications, as the collection is indexed by a public-index (basically the graph) which can be sampled once and then fixed for all future usages. A similar restriction holds for heuristic constructions as well.

[4]We did not attempt to optimize the constraints and parameters and some of them can be improved.

By taking $p = \frac{1}{2}$, we obtain the same parameters as in [11, 19]. More interestingly, we can obtain much stronger inapproximability ratio of, say, $p = n^{-1/(2d)}$ for a *fixed* locality d, assuming that $\mathcal{F}_{n^4,Q}$ is $\frac{1}{n}$-pseudorandom (say when $m = n^4$). As shown in Item 3 of Thm. 1.1, the latter assumption follows from the one-wayness of $\mathcal{F}_{m',Q}$ for sufficiently large polynomial $m'(n)$ (e.g., n^{14}).

Another advantage of Theorem 1.2, is that it yields average-case hardness over samplable distributions. Namely, we construct a pair of distributions D_{yes} and D_{no} over hypergraphs which are indistinguishable and such that D_{yes} (resp., D_{no}) outputs whp a yes instance (resp., no instance). Specifically, D_{no} will be a distribution over random graphs (whose number of edges is chosen from the binomial distribution), and D_{yes} will be a distribution with a planted dense-subgraph which essentially encodes a preimage of the pseudorandom generator.

The source of improvement. In a nutshell, the source of improvement over previous results is due to the strong nature of pseudorandomness which allows us to apply some form of product amplification for "free" without increasing the degree. In more detail. Pseudorandomness means 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. We define a mapping ρ that given a graph G and an m-bit string z, outputs a new graph G' by throwing away all edges which are indexed by 0 under z. It can be shown that ρ maps the "random" distribution to No-instances of $\frac{1}{2} - \mathsf{DSH}$, and the pseudorandom distribution to Yes instances of $\frac{1}{2} - \mathsf{DSH}$. Intuitively, the latter follows by noting that, assuming that $Q(1^d) = 1$, the set of nodes which are indexed by ones under x, does not lose any hyperedge.

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 $(G, y^1, \ldots, y^{(t)})$ is indistinguishable from the tuple $(G, f_{G,P}(x^{(1)}), \ldots, f_{G,Q}(x^{(t)}))$ 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.

More on DSH. DSH is a natural generalization of the notoriously hard Densest k-Subgraph (DSG) problem (e.g., [12]) whose exact approximation ratio is an important open question. The best known algorithm achieves $O(n^{1/4})$-approximation [7], while known hardness results only rule out PTAS [19]. 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 [11, 19]. Recently, it was shown how to use the average-case hardness of DSH to plant a trapdoor in $\mathcal{F}_{Q,m}$, and obtain public-key encryption schemes [3]. This raises the exciting possibility that, for random local functions, there may be a "path" from one-wayness to public-key cryptography: first assume one-wayness of $\mathcal{F}_{Q,m}$, then use Thm. 1.1 to argue that this collection is actually pseudorandom, then employ Thm. 1.2 to argue that DSH is hard over a planted distribution, and finally, use [3] to obtain a public-key cryptosystem. Unfortunately, the parameters given in Thm. 1.2 do not match the ones needed in [3]; 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 Thm. 1.1.

2.1 Constructing Weak-PPRGs (Thm. 1.1– second item)

The basic procedure. Due to the known reduction from pseudorandomness to unpredictability (aka Yao's theorem [25]), it suffices to reduce the task of inverting $\mathcal{F}_{Q,m}$ to the task of predicting the next bit in the output of $\mathcal{F}_{Q,k}$ 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 ε-predictor $\mathbf{P}$. This predictor is given a random (k,n,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 $S = (i_1, \dots, i_d)$. 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 $b \in \{0,1\}$.

A problematic approach. 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 approach faces a serious difficulty due to the low quality of the prediction. Recall that we plan to employ Yao's theorem, and therefore our reduction should work with prediction advantage ε which is smaller than $1/k < 1/n$. Hence, the 2-LIN equations that we collect are highly noisy. 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 as it "gives only one bit of information".

Partial re-randomization. We fix the problem by "flattening" the distribution of the prediction errors over all possible hyperedges. This is done by re-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 as edges were chosen uniformly at random any way. This yield a partial form of "random-self-reducibility": any 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 are uniformly and independently distributed. This is not true as all the $x^{(j)}$ 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 output too much. This optimization (and others) allow us to achieve a low overhead and take $k = m \cdot \varepsilon^2$.

2.2 Constructing LPRGs

Let us move to the case of LPRGs (the first item of Thm. 1.1). 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 ε-unpredictability for some constant $\varepsilon < 1$.[5] 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. Specifically, it turns out that the recent transformation of [16] (HRV), which is based on randomness extractors, works well in this range of parameters. The only problem is that, in general, one can show that $\mathbf{NC^0}$ cannot compute good randomness extractors. Fortunately, it turns out that for the special case of constant unpredictability and linear stretch, the HRV construction can be instantiated with low-quality extractors for which there are (non-trivial) $\mathbf{NC^0}$ implementations [21, 6]. This allows us to transform any $\Omega(n)$-long sequence with constant ε-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 with the *same* graph G, hyperedge S, and m-bit string y, where 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 more than $\frac{1}{2}$ of the coordinates. At this point we employ the recent algorithm of [8] which recovers x given such an approximation x' and $f_{G,Q}(x)$.

Organization. Some preliminaries are given in Section 3 including background on Goldreich's function and cryptographic definitions. Sections 4– 6 are devoted to the proof of Thm. 1.1, where Sections 4 and 5 describe the reductions from inversion to prediction (for the LPRG setting and for the PPRG setting), and Section 6 completes the proof of the main theorem based on additional generic transformations. Finally, in Section 7, we prove Thm. 1.2.

[5] 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 $\lfloor d/w \rfloor$ are in the matching and so the matching sensitivity is exactly $2\binom{d}{\lfloor d/2 \rfloor}/2^d$.

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.

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 $m = m(n)$, d, and a predicate $Q : \{0,1\}^d \to \{0,1\}$, we let $\mathcal{F}_{Q,m} : \{0,1\}^* \to \{0,1\}^*$ be the mapping that for each length parameter n takes as an input a pair of an (n,m,d) graph G and an n-bit string x, and outputs the pair $(G, f_{G,Q_n}(x))$.

Sensitivity and influence measures. Let $Q : \{0,1\}^d \to \{0,1\}$ be a predicate. We associate with Q a bipartite graph $G_Q = (V_0 \cup V_1, E)$ where $V_b = \left\{w \in \{0,1\}^d | Q(w) = b\right\}$ and $(u,v) \in V_0 \times V_1$ is an edge if there exists an $i \in [d]$ for which $u = v^{\oplus i}$. 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 G_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:

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

where the maximum is taken over all matchings in G_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 predicate $Q : \{0,1\}^d \to \{0,1\}$ we have:*

$$\mathrm{Inf}_{\max}(Q) \le \mathrm{Match}(Q) \le 2\min(\partial(Q), \bar{\partial}(Q)) \le \sum_i \mathrm{Inf}_i(Q) \le 2d\partial(Q).$$

Proof. Consider the graph G_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 [18], if Q is balanced then we also have $c \log d/d \le \mathrm{Inf}_{\max}(Q)$ 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: a 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. 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 $\mathrm{poly}(n)$ list of candidates, and sufficiently large n's we have that

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

where dist 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.[6]

[6]Note that in the case, of $\delta = 0$, we can assume that the list contains a single candidate, as the algorithm can efficiently check which of the candidates (if any) is a preimage. Hence, the notion of $(0,\varepsilon)$-approximate one-wayness is indeed equivalent to the standard notion of ε one-wayness.

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 $\mathcal{A}$, $|\Pr[\mathcal{A}(1^n, Y) = 1] - \Pr[\mathcal{A}(1^n, Z) = 1]| < \varepsilon(n)$. 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* 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 $i \in [n]$) with an additive error of δ with all but exponentially small failure probability, and then choose the best index.*

3.2 Properties of Goldreich's function

The following propositions shows 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.

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

Proof. The proof follows easily from the symmetric structure of $\mathcal{F}$. Assume towards a contradiction that $\mathcal{F}_{Q,m}$ can be predicted with success probability ε. Suppose that there exists a next-bit predictor $\mathbf{P}$ and a sequence of indices $\{i_n\}$ such that

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

We construct a last-bit predictor $\mathbf{P}'$ with success probability of $\alpha - o(\alpha)$ as follows. First, use Remark 3.2 to efficiently find an index $j \in [m]$ such that, with probability $1 - \text{neg}(n)$ over the coins of $\mathbf{P}'$, it holds that $\Pr[\mathbf{P}(G, y_{1..j}) = y_{j+1}] > \alpha(n) - \alpha(n)^2$ 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 the claim follows. □

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. Assume, without loss of generality, that Q is a non-constant d local predicate (otherwise, the theorem is trivially true), and let $0 < \delta < \frac{1}{2}$ be a fixed proximity parameter (that may depend on d). In Thm. 2 of [8] 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 fixed proximity parameter $\delta \in (0, \frac{1}{2})$. More precisely, it is shown that for a fraction of $1 - o(1)$ of all (m, n, d) hypergraphs G, we have that

$$\Pr_{x \xleftarrow{R} \mathcal{U}_n, y = f_{G,Q}(x)} [\forall x' \text{ s.t. } \mathsf{dist}(x, x') \leq \delta, A(y, x') \in f_{G,Q}^{-1}(y)] > 1 - \text{neg}(n). \tag{1}$$

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 $(G, y = f_{G,Q}(x))$, where G is a random (m, n, d) graph and $x \xleftarrow{R} \mathcal{U}_n$, finds a string x' which δ-approximates x with probability ε' (for infinitely many n's). To prove the first item (where $m > cn$) invoke B, obtain an approximation x' w.p. ε', feed the algorithm A with G, y and x' and output its 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 (resp., bits) of G (resp. of y). Now first apply B to (G_1, y_1) to obtain an approximation x' and then apply A to (G_2, y_2, x'). 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 and so the probability that they both happen is $\varepsilon'(1 - o(1))$. Conditioned on this, the algorithm A succeeds with probability $1 - \text{neg}(n)$, and so by a union bound we get that the overall success probability is $\varepsilon = \varepsilon'(1 - o(1)) - \text{neg}(n) = \varepsilon'(1 - o(1))$, as needed. □

4 Random Local Functions with Constant Unpredictability

We will prove the following theorem:

Theorem 4.1 (one-way $\Rightarrow$ somewhat-unpredictable)**.** *For every constants ε 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'-unpredictable for some constant $\varepsilon' < 1$. (In particular, $\varepsilon' = 1 - \text{Match}(Q)/2 + \Theta(\varepsilon)$.)*

By Propositions 3.3 and 3.4 (part 1), we can replace next-bit prediction with last-bit predictor and exact inversion with approximate inversion. Hence, it suffices to prove the following:

Theorem 4.2 (approximate one-way $\Rightarrow$ last-bit unpredictability)**.** *For every polynomial $m = m(n)$, constant $d \in \mathbb{N}$, predicate $Q : \{0,1\}^d \to \{0,1\}$, and constant $0 < \varepsilon < \mu = \text{Match}(Q)$, 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 4.1.

4.1 Proof of Thm. 4.2

To prove the theorem we consider the following algorithm (see Figure 1) which makes calls to a last-bit predictor $\mathbf{P}$. 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 $Q(x_S)$.

- Input: an (n, m, d) graph G and a 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 1: Basic Algorithm.

We analyze the algorithm. 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).

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

$$Q(x_S) \quad \neq \quad Q(x_S^{\oplus \ell}) \tag{2}$$

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 $i \in [n]$

$$\mathbf{P}(G, f_{G,Q}(x), S_{\ell \leftarrow i}; r) = Q(x_{S_{\ell \leftarrow i}}). \tag{3}$$

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

Lemma 4.3. *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. 2, 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. 3 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 4.4. *For at least $\varepsilon - \mathrm{neg}(n)$ fraction of the pairs (x, G), we have that*

$$\Pr_{S,\ell,r}[(x, G, r, S, \ell) \text{ is good}] > \Omega(\varepsilon\mu/d)). \tag{4}$$

We can now prove Thm. 4.2.

Proof of Thm. 4.2. Given an input G and a string $y = f_{G,Q}(x)$, invoke the basic algorithm $O(d/(\varepsilon\mu))$ times each time with a randomly chosen coins, and output all the $O(d/(\varepsilon\mu))$ candidates. Let us condition on the event that the pair (G, x) satisfies Eq. 4, which, by Lemma 4.4, happens with probability at least $\varepsilon/2$. In this case, by Lemmas 4.3 and 4.4, 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, we can make sure that at least one iteration succeeds with probability $\varepsilon/4$, and so, by a union bound, the overall success probability is $\varepsilon/2 - \varepsilon/4 = \varepsilon/4$. □

4.2 Proof of Lemma 4.4

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

$$\Pr_S[\mathbf{P}(G, f_{G,Q}(x), S; r) = Q(x_S)] > 1 - \mu/2 + \varepsilon/2. \tag{5}$$

Claim 4.5. *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 $\varepsilon < \mu$,

$$\Pr_{G,x,r}[(x, G) \text{ satisfy Eq. 5}] > \varepsilon/(\mu - \varepsilon) > \varepsilon.$$

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

$$\Pr_{G,x \text{ is balanced},r}[(x, G) \text{ satisfy Eq. 5}] > \varepsilon - \mathrm{neg}(n),$$

and the claim follows. □

Fix a good triple (x, G, r). Let us define for every set S the event $A(S)$ which happens if $\mathbf{P}(G, f_{G,Q}(x), S; r) = Q(x_S)$. To prove Lemma 4.4 it suffices to show that

Lemma 4.6. *For a fraction of at least $\frac{\varepsilon\mu}{3d} \cdot (1 - o(1))$ of the pairs (S, ℓ), the following hold:*

$$Q(x_S) \neq Q(x_S^{\oplus \ell}) \tag{6}$$

$$\Pr_{i \in [n]}[A(S_{\ell \leftarrow i})] > \frac{1}{2} + \varepsilon/6 \tag{7}$$

Proof. First, we will need some definitions. For a set S let $x_S \in \{0,1\}^d$ be the "label" of the set. Let M be a maximal matching of the predicate Q whose cardinality is $\mu 2^d$. We restrict our attention to sets S for which $x_S \in M$. For such S, we define the *index* $\ell(S)$ to be the single integer $\ell \in [n]$ 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, we have that $Q(x_S) \neq Q(x_S^{\oplus \ell})$, where ℓ is the index of S. Hence, to prove the lemma, it suffices to show that the following probabilistic event E:

$$x_S \in M \bigwedge \ell = \ell(S) \bigwedge \Pr_{i \in [n]}[A(S_{\ell \leftarrow i})] > \frac{1}{2} + \varepsilon/6,$$

happens with probability at least $\frac{\varepsilon\mu}{3d} \cdot (1 - o(1))$ over a random choice of S and ℓ. $\Pr_{S,\ell}[E]$ is lower-bounded by

$$\Pr_S[x_S \in M] \cdot \Pr_{\ell \xleftarrow{R} [d]}[\ell = \ell(S)] \cdot \Pr_{S \text{ s.t. } x_S \in M}\left[\Pr_{i \xleftarrow{R} [n]}[A(S_{\ell(S) \leftarrow i})] > \frac{1}{2} + \varepsilon/6\right].$$

Clearly, we have that $\Pr_{\ell \xleftarrow{R} [d]}[\ell = \ell(S)] = 1/d$ and so it suffices to show that

$$\Pr_S[x_S \in M] > \mu - o(1) \tag{8}$$

$$\Pr_{S \text{ s.t. } x_S \in M}\left[\Pr_{i \xleftarrow{R} [n]}[A(S_{\ell(S) \leftarrow i})] > \frac{1}{2} + \varepsilon/6\right] > \varepsilon/3. \tag{9}$$

Before we prove Eq. 8 and 9, we need the following simple observation. Note that the labels of S (which are d-bit strings) induce a partition over all the $n^{(d)}$ sets to 2^d classes. 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, we have

Claim 4.7. *For every $z \in \{0,1\}^d$, $p_z \in 2^{-d} \pm o(1)$.*

Proof. Since x is balanced $(\frac{n/2 - n^{2/3} - d}{n})^d < p_z < (\frac{n/2 + n^{2/3}}{n-d})^d$, and the claim follows as $d < o(n^{1/3})$. $\square$

Hence, Eq. 8 follows as

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

From now on, we focus on proving Eq. 9. We begin by showing that $\mathbf{P}$ succeeds well with respect to a set S chosen uniformly over all S's for which x_S is a node in M.

Claim 4.8. $\Pr_{S \text{ s.t. } x_S \in M}[A(S)] > \frac{1}{2} + \varepsilon/3$.

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

$$1 - \mu/2 + \varepsilon < \Pr_S[A(S)] = \Pr_S[x_S \notin M] \cdot \Pr_{S \text{ s.t. } x_S \notin M}[A(S)] + \Pr_S[x_S \in M] \cdot \Pr_{S \text{ s.t. } x_S \in M}[A(S)],$$

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

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

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

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

and the claim follows. □

Note that in Eq. 9, we are actually interested in prediction over a random "neighbor" $S_{\ell(S) \leftarrow i}$ of S. To analyze this, we need one final observation. We use the graph M to define a larger graph H over all the sets S for which $x_S \in M$. The edges of H are defined as follows: each S is connected to n nodes where the i-th node is $S_{\ell \leftarrow i}$ where ℓ is the index of S. We put forward some basic facts about the graph H:

Claim 4.9. *The graph H is undirected and each node has exactly n distinct neighbors including one self loop.*

Proof. We show that the graph is undirected. 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. □

In fact, it is not hard to verify that the edges form an equivalence relation and therefore the graph is composed of n-sized cliques. We can now prove Eq. 9. Namely, that **P** predicts well over a set S' which is a random neighbor of a random set S (for which $x_S \in M$):

Claim 4.10. *For at least $\varepsilon/3$ fraction of all sets S for which $x_S \in M$ we have*

$$\Pr_{i \xleftarrow{R} [n]}[A(S_{\ell(S) \leftarrow i})] > \frac{1}{2} + \varepsilon/6.$$

Proof. First note that

$$\Pr_{S \text{ s.t. } x_S \in M, i \xleftarrow{R} [n], T=S_{\ell(S)\leftarrow i}}[A(T)] = \Pr_{S \text{ s.t. } x_S \in M}[A(S)]. \tag{10}$$

Indeed, the set T is chosen by first choosing a random node S in the *regular* graph H and then letting T be a random neighbor of S in H. Hence, since H is a regular graph, T is just a random node (uniformly distributed over all S for which $x_S \in M$). Now by Claim 4.8, the rhs of Eq. 10 is at least $\frac{1}{2} + \varepsilon/3$, and so the current claim (4.10) follows from Markov's inequality. □

This completes the proof of Lemma 4.6. □

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

In this section we prove the following theorem:

Theorem 5.1 (one-way $\Rightarrow$ $(\frac{1}{2}+1/\text{poly})$-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 $(\frac{1}{2} + c\varepsilon)$-unpredictable, 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., $\text{Inf}_1(Q) = 1$. We rely on the following notation. For a permutation $\pi : [n] \to [n]$ and an ordered set $S = \{i_1, \ldots, i_d\} \subseteq [n]$ we let $\pi(S)$ denote the ordered set $\{\pi(i_1), \ldots, \pi(i_d)\} \subseteq [n]$. For an (m, n, d) graph $G = (S_1, \ldots, S_m)$ we let $\pi(G)$ denote the (m, n, d) graph $(\pi(S_1), \ldots, \pi(S_m))$. For a string $x \in \{0,1\}^n$, the string $\pi(x)$ is the string whose coordinates are permuted according to 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 $Q(x_S)$. Consider the algorithm Invert (see Figure 2), which is parameterized with t, and makes calls to the subroutine Vote (Figure 3).

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 and common pivot ℓ. To cope with this, we will show (in Lemma 5.2) that these queries can be viewed as independent samples, alas taken from a "modified" distribution which is different from the uniform. Later (in Lemma 5.3) we will show that, whp, $\mathbf{P}$ predicts well over this distribution as well.

- 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 $\ell \xleftarrow{R} [n]$, 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.

- 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 $S = (S_1, \ldots, S_d)$ from G subject to the constraint $S_1 = i$ and $\ell \notin \{S_2, \ldots, S_d\}$. 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. Let G' be the same as G except that the hyperedge S is removed. Similarly, let y' be the string y with its s-th bit removed. Define the hyperedge $S' = (\ell, S_2, \ldots, S_d)$.
3. Choose a random permutation $\pi : [n] \to [n]$, and let $(H = \pi(G'), y', T = \pi(S'))$.
4. Output $\mathbf{P}(H, z, T) \oplus y_s$.

Figure 3: Algorithm Vote.

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 $T = \{T_1, \ldots, T_d\}$ is chosen uniformly at random subject to $r_{T_1} = \sigma$. In Section 5.1, we prove the following lemma:

Lemma 5.2. *Let (G, y, ℓ, i) be the input to* Vote *where $G \stackrel{R}{\leftarrow} \mathcal{G}_{n,m,d}$, 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 $(H, \pi(x), y', T)$ is distributed according to D_{k,x_ℓ}, where k is the Hamming weight of x.*
3. *Conditioned on not failing, if the outcome of the predictor $\mathbf{P}(H, y', T)$ equals to $Q(\pi(x)_T)$ 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 5.2, we prove the following lemma:

Lemma 5.3. *With probability $\Omega(\varepsilon)$ over a random choice of the input $x \stackrel{R}{\leftarrow} \mathcal{U}_n$ and the pivot $\ell \stackrel{R}{\leftarrow} [n]$, we have that*

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

We can now prove the theorem.

Proof of Thm. 5.1 given the lemmas. Let us condition on the event that x and ℓ satisfy the equation of Lemma 5.3, and that our guess b for x_ℓ was successful. By Lemma 5.3, 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. Let us now upper-bound the probability that the output of Invert *disagrees* with x on the i-th bit for a fixed index $i \in [n]$. 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. Observe that we recover x_i correctly if $\sum \alpha_j$ is positive (as our guess b for x_ℓ is assumed to be correct). By Lemmas 5.2 and 5.3, 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 $\frac{1}{2} + \Omega(\varepsilon)$. We claim that the probability of $\sum \alpha_j \leq 0$ is at most $\exp(-\Omega(t\varepsilon^2))$. Indeed, first observe that, by a Chernoff bound, the probability of seeing at most $2t/3$ zeroes is at least $1 - \exp(-\Omega(t))$. Now, conditioned 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 Hoefding'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. 5.1, as by taking $t = O(\lg n/\varepsilon^2)$ 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 $\mathcal{F}_{Q,O(m \lg n/\varepsilon^2)}$ is $\Omega(\varepsilon)$-one-way. To obtain the stronger version (without the $\lg n$ 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 $1/3$-approximation of x with probability $\Omega(\varepsilon)$, which, by Prop. 3.4 (part 2), contradicts the $\Omega(\varepsilon)$-one-wayness of $\mathcal{F}_{Q,O(m/\varepsilon^2)+c_d n}$, where 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 $m' = O(m/\varepsilon^2) + c_d n$ and $\varepsilon' = \Omega(\varepsilon)$. 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 5.3 we will show that the above theorem generalizes to variants of $\mathcal{F}_{Q,m}$ that capture some of the existing heuristic candidates.

5.1 Proof of Lemma 5.2

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. 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') (defined in Step 2) is independently distributed such that $G' \overset{R}{\leftarrow} \mathcal{G}_{n,m-1,d}$ and S' is a random hyperedge whose first entry is ℓ. Since x is fixed and $y' = f_{Q,G'}(x)$, we have now a full understanding of the distribution of the tuple (G', x, y', S').

We will now analyze the effect of the permutation π. Let $x' = \pi(x)$ and $H = \pi(G')$. First, observe that for every fixed permutation π the tuple (H, x', y') satisfies $y' = f_{Q,H}(x')$ since $y' = f_{Q,G'}(x)$. Moreover, since G' is taken from $\mathcal{G}_{n,m-1,d}$, so is $H = \pi(G)$ 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, $(H \overset{R}{\leftarrow} \mathcal{G}_{n,m-1,d}, x' \overset{R}{\leftarrow} X_k, y' = f_{Q,H}(x'))$.

Let us now fix both H 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 $T = \pi_A(S')$ is uniformly and independently distributed under the constraint that the first entry τ of T satisfies $x'_\tau = x_\ell$. To see this, recall that $S'_1 = \ell$, and so the entry $T_1 = \pi_A(\ell)$ 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' are random indices (different than ℓ) and so for every fixing of π_A the $d-1$ last entries of T are still random. This completes the proof as we showed that the tuple (H, x', y', T) is distributed properly.

Third item. Let us move to the third item. Suppose that $\mathbf{P}$ outputs the bit $Q(\pi(x)_T)$. Then, since $T = \pi(S')$, the result equals to $Q(x_{S'})$, 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. $\square$

5.2 Proof of Lemma 5.3

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

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

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

Claim 5.4. *There exists a set $K \subseteq \{n/2 - n^{2/3}, \ldots, n/2 + n^{2/3}\}$ for which:*

$$\sum_{k \in K} p_k > \varepsilon/3 \tag{11}$$

$$\forall k \in K, q_k > \frac{1}{2} + \varepsilon/2 \tag{12}$$

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

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

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 $x \xleftarrow{R} \mathcal{U}_n$, we have that $x \in X$ and so

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

To complete the proof of the lemma, observe that for every $x \in X$, since x is balanced, the probability that a random pivot $\ell \xleftarrow{R} [n]$ 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. □

5.3 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 5.1 can be generalized to this setting as follows.

Theorem 5.5 (Thm. 5.1: 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. 5.1. 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 ([1, 6, 3]) 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 [21], and so in both cases we prove weak pseudorandomness from one-wayness.

6 From Unpredictability to Pseudorandomness

We will prove the main theorem by combining our "one-wayness to unpredictability" reductions (proved in Sections 4 and 5) with several generic transformations.

First we will need the well-known theorem of Yao [25] 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. 5.1 we obtain item 3 of Thm. 1.1:

Corollary 6.2 (Thm. 1.1, item 3 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. 5.1, if $\mathcal{F}_{Q,m^3/\delta^2}$ is one-way then $\mathcal{F}_{Q,m}$ is $(\frac{1}{2}+\Omega(\delta/m))$-unpredictable, and by Yao's theorem (Fact 6.1) the latter is $\Omega(\delta)$-pseudorandom. □

Recall that in Thm. 4.1 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).*

The theorem is proved by combining the techniques of [16] with non-trivial $\mathbf{NC^0}$ randomness extractors from [6]. The proof of this theorem is deferred to Section 6.1.

As a corollary of the above theorem and Thm. 4.1 we get:

Corollary 6.4 (Thm. 1.1, item 1 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. 1.1, item 2). For this purpose, we will need the following amplification procedures (together with Thm. 5.1):

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. 1.1, item 2.

Corollary 6.6 (Thm. 1.1, item 2 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. 5.1, $\mathcal{F}_{Q,n^{1+\delta/4}}$ 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, μ) *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 μ statistically-close to the uniform distribution. We refer to μ 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 [6]:

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.*

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 [16]. We sketch it here for completeness. First, by Proposition 4.8 of [16], 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 μ, where $\mu = 1-2\varepsilon$; 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 [16], 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 $n(\mu-o(1))$ 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 $G(\mathcal{U}_n)$, such that: (1) the min-entropy of W given the first i_n-1 columns of Y is at least $n(\mu-o(1))$; 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 $n(\mu-o(1)) > \alpha n$. 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. 1.2 restated)**.** *Let $d \in \mathbb{N}$ be a constant, Q be a d-ary predicate, and $m \geq n^c$, where $c > 3$ is a constant. If $\mathcal{F}_{m,Q}$ is $\varepsilon = o(1/(\sqrt{n}\cdot\log n))$-pseudorandom, then for every $1/n^{\frac{c-3}{2d}} \leq p \leq \frac{1}{2}$ the p-Densest-Subhypergraph problem is intractable with respect to d-uniform hypergraphs.*[7]

Note that the larger c gets, the better inapproximaility ratio we obtain. Clearly, c cannot be larger than $c(d)$ where $n^{c(d)}$ is the maximal stretch of d-local pseudorandom generators. Currently, the best upper-bound on $c(d)$ is roughly $d/2$ due to [21].

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 there exists an integer t for which $2^{-t} = p$, i.e., $1 \leq t \leq O(\log n)$. 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.

[7] We did not attempt to optimize the parameters and constraints, and some of them (e.g., $c > 3$) can be slightly improved.

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 $G' = \rho(G, Y)$.

First, we show that D_{no} and D_{yes} are weakly-indistinguishable.

Claim 7.2. *If $\mathcal{F}_{m,Q}$ is ε-pseudorandom then the ensembles D_{no} and D_{yes} (indexed by n) are $t\cdot\varepsilon = o(1/\sqrt{n})$-indistinguishable.*

Proof. A $t\varepsilon$-distinguisher immediately leads to a $t\varepsilon$-distinguisher between the distributions $(G, y^{(1)}, \dots, y^{(t)})$ and $(G, f_{G,Q}(x^{(1)}), \dots, f_{G,Q}(x^{(t)}))$ 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 $\mathcal{F}_{m,Q}$. □

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

Proof. The first item follows from an additive Chernoff bound: define m Bernoulli r.v.'s, where the i-th variable is 1 if the i-th hyperedge is chosen. Since the number of r.v.'s is m, the probability of having $m' = (p \pm 1/n)m$ is $1 - \mathrm{neg}(m) = 1 - \mathrm{neg}(n)$.

To prove the second item, let us condition on the event $m' > n^{c-1}$, which by the previous argument happens w/p $1 - \mathrm{neg}(n)$. (Recall that $c < d$ and so $1/n < p$). 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 an additive Chernoff bound, the probability that S contains a set of edges of density $p^d + 1/n^{(c/2)-1}$ is bounded by $\exp(-2m'/n^{c-2}) = \exp(-2n)$. Therefore, by a union bound, the probability that this happens for some set S is at most $\exp(-2n + n) = \mathrm{neg}(n)$. Finally, observe that our choice of p gurentess that $1/n^{(c/2)-1} = o(p^d)$. □

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

Lemma 7.4. *With probability at least $1/\sqrt{n}$, a graph G chosen from D_{yes} has a sub-graph of density p^d that contains a fraction of at least $p^{d-1}(1 - o(1))$ edges.*

We mention that a tighter analysis can be used to improve the quantity $1/\sqrt{n}$.

Proof. Label the i-th *node* of G by t-bit column of the matrix X, and let S be the set of nodes which are labeled by the all-one string. Consider the following event E in which: (1) S is of density exactly p; (2) At least $p^d - 1/n^{(c/2)-1}$ fraction of the edges of the original graph G fall in S; (3) The number of remaining edges m' in G' is at most $(p+1/n)m$.

First observe 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 p-dense set S contains a set of edges of density at least $(p^d - 1/n^{(c/2)-1})m/m' > \frac{p^d-1/n^{(c/2)-1}}{p+1/n}$. Observe that the restriction of p to $1/n^{\frac{c-3}{2d}} \le p \le \frac{1}{2}$, implies that the "error" terms $1/n^{(c/2)-1}$ and $1/n$ are $o(p^d)$ and $o(p)$, respectively. Hence, the density of the set of edges that fall into S can be written as $p^{d-1} \cdot \frac{1-o(1)}{1+o(1)} > p^{d-1}(1-o(1))$.

Now, let us bound the probability of the event E. First, since each node falls in S independently with probability p, we have (by standard properties of the binomial distribution) that the sub-event (1) holds with probability at least $\Omega(1/\sqrt{n})$. Conditioned on (1), the sub-event (2) happens with all but negligible probability due to additive Chernoff bound. Hence, (1) and (2) happen simultaneously w/p $\Omega(1/\sqrt{n})$.

Finally, we argue that the probability β that (3) holds is at least $1-\text{neg}(n)-t\cdot\varepsilon = 1-o(1/\sqrt{n})$. Indeed, consider the algorithm which attempts to distinguish D_{no} from D_{yes} by looking at m' and accepting iff it $m' \le (p+1/n)m$. By Lemma 7.3 this leads to a distinguisher with advantage $1-\text{neg}(n)-\beta$, which, by Claim 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 $\Omega(1/\sqrt{n})$. $\square$

Let us now prove Theorem 7.1.

Proof of Thm. 7.1. Lemma 7.3 guarantees that a graph sampled from D_{no} is almost always a NO instance, whereas, by Lemma 7.4, a graph sampled from D_{yes} is a YES instance with probability at least $\Omega(1/\sqrt{n})$. Hence, an algorithm that solves $p-\mathsf{DSH}$ for d-uniform hypergraphs can distinguish between the two distributions with advantage at least $\Omega(1/\sqrt{n})$, in contradiction to Claim 7.2. $\square$

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 would also like to thank Uri Feige, Yuval Ishai and Alex Samorodnitsky for many valuable conversations.

References

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

[2] B. Applebaum. *Cryptography in Constant Parallel Time.* Phd thesis, Technion, Israel Institute of Technology, Augost 2007.

[3] B. Applebaum, B. Barak, and A. Wigderson. Public-key cryptography from different assumptions. In *42nd ACM Symposium on Theory of Computing, (STOC 2010)*, pages 171–180, 2010.

[4] B. Applebaum, Y. Ishai, and E. Kushilevitz. Computationally private randomizing polynomials and their applications. *Journal of Computational Complexity*, 15(2):115–162, 2006.

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

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

[7] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an $(^{1/4})$ approximation for densest -subgraph. In *STOC*, pages 201–210, 2010.

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

[9] J. Cook, O. Etesami, R. Miller, and L. Trevisan. Goldreich's one-way function candidate and myopic backtracking algorithms. In O. Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 521–538. Springer, 2009.

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

[11] U. Feige. Relations between average case complexity and approximation complexity. In *STOC*, pages 534–543, New York, May 19–21 2002. ACM Press.

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

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

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

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

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

[17] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography with constant computational overhead. In R. E. Ladner and C. Dwork, editors, *STOC*, pages 433–442. ACM, 2008.

[18] Kahn, Kalai, and Linial. The influence of variables on boolean functions. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 1988.

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

[20] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993. Preliminary version in Proc. 30th FOCS, 1989.

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

[22] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993. Preliminary version in Proc. 22th STOC, 1990.

[23] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004. Preliminary version in Proc. 38th FOCS, 1997.

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

[25] 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 $k_1, \dots, k_t \in \{0,1\}^s$ and $x_1, \dots, x_t \in \{0,1\}^n$ let $F^{t\oplus}_{k_1,\dots,k_t}(x_1, \dots, x_t) = F_{k_1}(x_1) \oplus \dots \oplus F_{k_t}(x_t)$.

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 $\mathcal{A}'$

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

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

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

where each $y^{(j)}$ is sampled uniformly and independently from $Y_{[1..i_n-1]}$. By Yao's XOR lemma (cf. [15]), 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 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 [14, Chp. 3, Ex. 19].)

ECCC ISSN 1433-8092

http://eccc.hpi-web.de