Condensed Unpredictability

Maciej Skórski*[1], Alexander Golovnev†[2], and Krzysztof Pietrzak‡[3]

[1]University of Warsaw
[2]New York University
[3]IST Austria

Abstract

We consider the task of deriving a key with high HILL entropy (i.e., being computationally indistinguishable from a key with high min-entropy) from an unpredictable source.

Previous to this work, the only known way to transform unpredictability into a key that was ϵ indistinguishable from having min-entropy was via pseudorandomness, for example by Goldreich-Levin (GL) hardcore bits. This approach has the inherent limitation that from a source with k bits of unpredictability entropy one can derive a key of length (and thus HILL entropy) at most $k - 2\log(1/\epsilon)$ bits. In many settings, e.g. when dealing with biometric data, such a $2\log(1/\epsilon)$ bit entropy loss in not an option.

Our main technical contribution is a theorem that states that in the high entropy regime, unpredictability implies HILL entropy. Concretely, any variable K with $|K| - d$ bits of unpredictability entropy has the same amount of so called metric entropy (against real-valued, deterministic distinguishers), which is known to imply the same amount of HILL entropy. The loss in circuit size in this argument is exponential in the entropy gap d, and thus this result only applies for small d (i.e., where the size of distinguishers considered is exponential in d).

To overcome the above restriction, we investigate if it's possible to first "condense" unpredictability entropy and make the entropy gap small. We show that any source with k bits of unpredictability can be condensed into a source of length k with $k - 3$ bits of unpredictability entropy. Our condenser simply "abuses" the GL construction and derives a k bit key from a source with k bits of unpredicatibily. The original GL theorem implies nothing when extracting that many bits, but we show that in this regime, GL still behaves like a "condenser" for unpredictability. This result comes with two caveats (1) the loss in circuit size is exponential in k and (2) we require that the source we start with has *no* HILL entropy (equivalently, one can efficiently check if a guess is correct). We leave it as an intriguing open problem to overcome these restrictions or to prove they're inherent.

1 Introduction

Key-derivation considers the following fundamental problem: Given a joint distribution (X, Z) where $X|Z$ (which is short for "X conditioned on Z") is guaranteed to have some kind of entropy,

* maciej.skorski@gmail.com. Research supported by the WELCOME/2010-4/2 grant.

† alexgolovnev@gmail.com

‡ pietrzak@ist.ac.at. Research supported by ERC starting grant (259668-PSPC).

derive a “good" key $K = h(X, S)$ from X by means of some efficient key-derivation function h, possibly using public randomness S.

In practice, one often uses a cryptographic hash function like SHA3 as the key derivation function $h(.)$ [Kra10, DGH$^+$04], and then simply assumes that $h(.)$ behaves like a random oracle [BR93].

In this paper we continue the investigation of key-derivation with provable security guarantees, where we don't make any computational assumption about $h(.)$. This problem is fairly well understood for sources $X|Z$ that have high min-entropy (we'll formally define all the entropy notions used in 2 below), or are computationally indistinguishable from having so (in this case, we say $X|Z$ has high HILL entropy). In the case where $X|Z$ has k bits of min-entropy, we can either use a strong extractor to derive a $k - 2\log\epsilon^{-1}$ key that is ϵ-close to uniform, or a condenser to get a k bit key which is ϵ-close to a variable with $k - \log\log\epsilon^{-1}$ bits of min-entropy. Using extractors/condensers like this also works for HILL entropy, except that now we only get computational guarantees (pseudorandom/high HILL entropy) on the derived key.

Often one has to derive a key from a source $X|Z$ which has no HILL entropy at all. The weakest assumption we can make on $X|Z$ for any kind of key-derivation to be possible, is that X is hard to predict given Z. This has been formalized in [HLR07a] by saying that $X|Z$ has k bits of unpredictability entropy, denoted $H^{\mathsf{unp}}_{s}(X|Z) \geqslant k$, if no circuit of size s can predict X given Z with advantage $\geqslant 2^{-k}$ (to be more general, we allow an additional parameter $\delta \geqslant 0$, and $H^{\mathsf{unp}}_{\delta,s}(X|Z) \geqslant k$ holds if (X, Z) is δ-close to some distribution (Y, Z) with $H^{\mathsf{unp}}_{s}(Y|Z) \geqslant k$). We will also consider a more restricted notion, where we say that $X|Z$ has k bits of *list*-unpredictability entropy, denoted $H^{*\mathsf{unp}}_{s}(X|Z) \geqslant k$, if it has k bits of unpredictability entropy relative to an oracle Eq which can be used to verify the correct guess (Eq outputs 1 on input X, and 0 otherwise).[1] We'll discuss this notion in more detail below. For now, let us just mention that for the important special case where it's easy to verify if a guess for X is correct (say, because we condition on $Z = f(X)$ for some one-way function[2] f), the oracle Eq does not help, and thus unpredictability and list-unpredictability coincide. The results proven in this paper imply that from a source $X|Z$ with k bits of list-unpredictability entropy, it's possible to extract a k bit key with $k - 3$ bits of HILL entropy

Proposition 1. *Consider a joint distribution (X, Z) over $\{0,1\}^n \times \{0,1\}^m$ where*

$$H^{*\mathsf{unp}}_{s,\gamma}(X|Z) \geqslant k \tag{1}$$

Let $S \in \{0,1\}^{n\times k}$ be uniformly random and $K = X^T S \in \{0,1\}^k$, then the unpredictability entropy of K is

$$H^{\mathsf{unp}}_{s/2^{2k}poly(m,n),\gamma}(K|Z, S) \geqslant k - 3 \tag{2}$$

and the HILL entropy of K is

$$H^{\mathsf{HILL}}_{t,\epsilon+\gamma}(K|Z, S) \geqslant k - 3 \tag{3}$$

with[3] $t = s \cdot \frac{\epsilon^7}{2^{2k}poly(m,n)}$.

[1] We chose this name as having access to Eq is equivalent to being allowed to output a list of guesses. This is very similar to the well known concept of list-decoding.

[2] To be precise, this only holds for *injective* one-way functions. One can generalise list-unpredictability and let Eq output 1 on some set $\mathcal{X}$, and the adversary wins if she outputs any $X \in \mathcal{X}$. Our results (in particular Theorem 1) also hold for this more general notion, which captures general one-way functions by letting $\mathcal{X} = f^{-1}(f(X))$ be the set of all preimages of $Z = f(X)$.

[3] We denote with $poly(m, n)$ some fixed polynomial in (n, m), but it can denote different polynomial throughout the paper. In particular, the $poly$ here is not the same as in (2) as it hides several extra terms.

Proposition 1 follows from two results we prove in this paper.

First, in Section 4 we prove Theorem 1 which shows how to "abuse" Goldreich-Levin hardcore bits by generating a k bit key $K = X^T S$ from a source $X|Z$ with k bits of list-unpredictability. The Goldreich-Levin theorem [GL89] implies nothing about the pseudorandomness of $K|(Z,S)$ when extracting that many bits. Instead, we prove that GL is a good "condenser" for unpredictability entropy: if $X|Z$ has k bits of list-unpredictability entropy, then $K|(Z,S)$ has $k-3$ bits of unpredictability entropy (note that we start with list-unpredictability, but only end up with "normal" unpredictability entropy). This result is used in the first step in Proposition 1, showing that (1) implies (2).

Second, in Section 5 we prove our main result, Theorem 2 which states that any source $X|Z$ which has $|X|-d$ bits of unpredictability entropy, has the same amount of HILL entropy (technically, we show that it implies the same amount of metric entropy against deterministic real-valued distinguishers. This notion implies the same amount of HILL entropy as shown by Barak et al. [BSW03]). The security loss in this argument is exponential in the entropy gap d. Thus, if d is very large, this argument is useless, but if we first condense unpredictability as just explained, we have a gap of only $d = 3$. This result is used in the second step in Proposition 1, showing that (2) implies (3). In the two sections below we discuss two shortcomings of Theorem 1 which we hope can be overcome in future work.[4]

1.0.1 On the dependency on 2^k in Theorem 1.

As outlined above, our first result is Theorem 1, which shows how to condense a source with k bits of list-unpredictability into a k bit key having $k-3$ bits of unpredictability entropy. The loss in circuit size is $2^{2k}poly(m,n)$, and it's not clear if the dependency on 2^k is necessary here, or if one can replace the dependency on 2^k with a dependency on $poly(\epsilon^{-1})$ at the price of an extra ϵ term in the distinguishing advantage. In many settings $\log(\epsilon^{-1})$ is in the order of k, in which case the above difference is not too important. This is for example the case when considering a k bit key for a symmetric primitive like a block-cipher, where one typically assumes the hardness of the cipher to be exponential in the key-length (and thus, if we want ϵ to be in the same order, we have $\log(\epsilon^{-1}) = \Theta(k)$). In other settings, k can be superlinear in $\log(\epsilon^{-1})$, e.g., if the the high entropy string is used to generate an RSA key.

1.0.2 List vs. normal Unpredictability.

Our Theorem 1 shows how to condense a source where $X|Z$ has k bits of *list*-unpredictability entropy into a k bit string with $k-3$ bits unpredictability entropy. It's an open question to which extent it's necessary to assume *list*-unpredictability here, maybe "normal" unpredictability is already sufficient? Note that list-unpredictability is a lower bound for unpredictability as one always can ignore the Eq oracle, i.e., $H^{\mathsf{unp}}_{\epsilon,s}(X|Z) \geqslant H^{*\mathsf{unp}}_{\epsilon,s}(X|Z)$, and in general, list-unpredictability can be much smaller than unpredictability entropy.[5]

[4] After announcing this result at a workshop, we learned that Colin Jia Zheng proved a weaker version of this result. Theorem 4.18 in this PhD thesis, which is available via http://dash.harvard.edu/handle/1/11745716 also states that k bits of unpredictability imply k bits of HILL entropy. Like in our case, the loss in circuit size in his proof is polynomial in ϵ^{-1}, but it's also exponential in n (the length of X), whereas our loss is only exponential in the entropy gap $\Delta = n - k$.

[5] E.g., let X by uniform over $\{0,1\}^n$ and Z arbitrary, but independent of X, then for $s = \exp(n)$ we have $H^{\mathsf{unp}}_{s}(X|Z) = n$ but $H^{*\mathsf{unp}}_{s}(X|Z) = 0$ as we can simply invoke Eq on all $\{0,1\}^n$ until X is found.

Interestingly, we can derive a k bit key with almost k bits of HILL entropy from a source $X|Z$ which k bits unpredictability entropy $H^{\mathsf{unp}}_{\epsilon,s}(X|Z) \geqslant k$ in two extreme cases, namely, if either

1. if $X|Z$ has basically no HILL entropy (even against small circuits).

2. or when $X|Z$ has (almost) k bits of (high quality) HILL entropy.

In case 1. we observe that if $H^{\mathsf{HILL}}_{\epsilon,t}(X|Z) \approx 0$ for some $t \ll s$, or equivalently, given Z we can efficiently distinguish X from any $X' \neq X$, then the Eq oracle used in the definition of list-unpredictability can be efficiently emulated, which means it's redundant, and thus $X|Z$ has the same amount of list-unpredictability and unpredictability entropy, $H^{\mathsf{unp}}_{s,\epsilon}(X|Z) \approx H^{*\mathsf{unp}}_{s',\epsilon'}(X|Z)$ for $(\epsilon', s') \approx (\epsilon, s)$. Thus, we can use Theorem 1 to derive a k bit key with $k - O(1)$ bits of HILL entropy in this case. In case 2., we can simply use any condenser for min-entropy to get a key with HILL entropy $k - \log\log\epsilon^{-1}$ (cf. Figure 2). As condensing almost all the unpredictability entropy into HILL entropy is possible in the two extreme cases where $X|Z$ has either no or a lot of HILL entropy, it seems conceivable that it's also possible in all the in-between cases (i.e., without making any additional assumptions about $X|Z$ at all).

1.0.3 GL vs. Condensing.

Let us stress as this point that, because of the two issues discussed above, our result does not always allow to generate more bits with high HILL entropy than just using the Goldreich-Levin theorem. Assuming k bits of unpredictability we get $k-3$ of HILL, whereas GL will only give $k - 2\log(1/\epsilon)$. But as currently our reduction has a quantitatively larger loss in circuit size than the GL theorem, in order to get HILL entropy of the same quality (i.e., secure against (s,δ) adversaries for some fixed (s,δ)) we must consider the unpredictability entropy of the source $X|Z$ against more powerful adversaries than if we're about to use GL. And in general, the amount of unpredictability (or any other computational) entropy of $X|Z$ can decrease as we consider more powerful adversaries.

2 Entropy Notions

In this section we formally define the different entropy notions considered in this paper. We denote with $\mathcal{D}^{rand,\{0,1\}}_s$ the set of all *probabilistic* circuits of size s with *boolean* output, and $\mathcal{D}^{rand,[0,1]}_s$ denotes the set of all *probabilistic* circuits with *real-valued* output in the range $[0,1]$. The analogous *deterministic* circuits are denoted $\mathcal{D}^{det,\{0,1\}}_s$ and $\mathcal{D}^{det,[0,1]}_s$. We use $X \sim_{\epsilon,s} Y$ to denote computational indistinguishability of variables X and Y, formally[6]

$$X \sim_{\epsilon,s} Y \iff \forall \mathsf{C} \in \mathcal{D}^{rand,\{0,1\}}_s \;:\; |\Pr[\mathsf{C}(X)=1] - \Pr[\mathsf{C}(Y)=1]| \leqslant \epsilon \tag{4}$$

$X \sim_\epsilon Y$ denotes that X and Y have statistical distance ϵ, i.e., $X \sim_{\epsilon,\infty} Y$, and with $X \sim Y$ we denote that they're identically distributed. With U_n we denote the uniform distribution over $\{0,1\}^n$.

Definition 1. *The* **min-entropy** *of a random variable X with support $\mathcal{X}$ is*

$$H_\infty(X) = -\log_2 \max_{x\in\mathcal{X}} \Pr[X = x]$$

[6] Let us mention that the choice of the distinguisher class in (4) irrelevant (up to a small additive difference in circuit size), we can replace $\mathcal{D}^{rand,\{0,1\}}_s$ with any of the three other distinguisher classes.

For a pair (X,Z) of random variables, the **average min-entropy** *of X conditioned on Z is*

$$\widetilde{H}_\infty(X|Z) \;=\; -\log_2 \mathop{\mathbb{E}}_{z\leftarrow Z} \max_x \Pr[X=x|Z=z] \;=\; -\log_2 \mathop{\mathbb{E}}_{z\leftarrow Z} 2^{-H_\infty(X|Z=z)}$$

HILL entropy is a computational variant of min-entropy, where X (conditioned on Z) has k bits of HILL entropy, if it cannot be distinguished from some Y that (conditioned on Z) has k bits of min-entropy, formally

Definition 2 ([HILL99], [HLR07a])**.** *A random variable X has* **HILL entropy** *k, denoted by $H^{\mathsf{HILL}}_{\varepsilon,s}(X) \geq k$, if there exists a distribution Y satisfying $H_\infty(Y) \geq k$ and $X \sim_{\varepsilon,s} Y$.*

Let (X,Z) be a joint distribution of random variables. Then X has **conditional HILL entropy** *k conditioned on Z, denoted by $H^{\mathsf{HILL}}_{\varepsilon,s}(X|Z) \geq k$, if there exists a joint distribution (Y,Z) such that $\widetilde{H}_\infty(Y|Z) \geq k$ and $(X,Z) \sim_{\varepsilon,s} (Y,Z)$.*

Barak, Sahaltiel and Wigderson [BSW03] define the notion of metric entropy, which is defined like HILL, but the quantifiers are exchanged. That is, instead of asking for a single distribution (Y,Z) that fools all distinguishers, we only ask that for every distinguisher D, there exists such a distribution. For reasons discussed in Section 2.0.4, in the definition below we make the class of distinguishers considered explicit.

Definition 3 ([BSW03], [FR12])**.** *Let (X,Z) be a joint distribution of random variables. Then X has* **conditional metric entropy** *k conditioned on Z (against probabilistic boolean distinguishers), denoted by $H^{\mathsf{Metric},rand,\{0,1\}}_{\varepsilon,s}(X|Z) \geq k$, if for every $\mathsf{D} \in \mathcal{D}^{rand,\{0,1\}}_s$ there exists a joint distribution (Y,Z) such that $\widetilde{H}_\infty(Y|Z) \geq k$ and*

$$|\Pr[\mathsf{D}(X,Z)=1] - \Pr[\mathsf{D}(Y,Z)=1]| \leqslant \epsilon$$

More generally, for $class \in \{rand, det\}, range \in \{[0,1],\{0,1\}\}$, $H^{\mathsf{Metric},class,range}_{\varepsilon,s}(X|Z) \geq k$ if for every $\mathsf{D} \in \mathcal{D}^{class,range}_s$ such a (Y,Z) exists.

Like HILL entropy, also unpredictability entropy, which we'll define next, can be seen as a computational variant of min-entropy. Here we don't require indistinguishability as for HILL entropy, but only that the variable is hard to predict.

Definition 4 ([HLR07a])**.** *X has* **unpredictability entropy** *k conditioned on Z, denoted by $H^{\mathsf{unp}}_{\epsilon,s}(X|Z) \geq k$, if (X,Z) is (ϵ,s) indistinguishable from some (Y,Z), where no probabilistic circuit of size s can predict Y given Z with probability better than 2^{-k}, i.e.,*

$$H^{\mathsf{unp}}_{s,\epsilon}(X|Z) \geq k \iff \exists (Y,Z), (X,Z) \sim_{\varepsilon,s} (Y,Z)\ \forall \mathsf{C}, |\mathsf{C}| \leqslant s\ : \Pr_{(y,z)\leftarrow(Y,Z)}[\mathsf{C}(z)=y] \leqslant 2^{-k} \tag{5}$$

We also define a notion called "list-unpredictability", denoted $H^{*\mathsf{unp}}_{\epsilon,s}(X|Z) \geq k$, which holds if $H^{\mathsf{unp}}_{\epsilon,s}(X|Z) \geq k$ as in (5), but where C additionally gets oracle access to a function $\mathsf{Eq}(.)$ which outputs 1 on input y and 0 otherwise. So, C can efficiently test if some candidate guess for y is correct.[7]

[7] We name this notion "list-unpredictability" as we get the same notion when instead of giving C oracle access to $\mathsf{Eq}(.)$, we allow $\mathsf{C}(z)$ to output a list of guesses for y, not just one value, and require that $\Pr_{(y,z)\leftarrow(Y,Z)}[y \in \mathsf{C}(z)] \leqslant 2^{-k}$. This notion is inspired by the well known notion of list-decoding.

Remark 1 (The ϵ parameter)**.** *The ϵ parameter in the definition above is not really necessary, following [HLR07b], we added it so we can have a "smooth" notion, which is easier to compare to HILL or smooth min-entropy. If $\epsilon = 0$, we'll simply omit it, then the definition simplifies to*

$$H_s^{\mathsf{unp}}(X|Z) \geq k \iff \Pr_{(x,z)\leftarrow(X,Z)}[\mathsf{C}(z) = x] \leqslant 2^{-k}$$

Let us also mention that unpredictability entropy is only interesting if the conditional part Z is not empty as (already for s that is linear in the length of X) we have $H_s^{\mathsf{unp}}(X) = H_\infty(X)$ which can be seen by considering the circuit C (that gets no input as Z is empty) which simply outputs the constant x maximizing $\Pr[X = x]$.

2.0.4 Metric vs. HILL.

We will use a lemma which states that deterministic real-valued metric entropy implies the same amount of HILL entropy (albeit, with some loss in quality). This lemma has been proven by [BSW03] for the unconditional case, i.e., when Z in the lemma below is empty, it has been observed by [FR12, CKLR11] that the proof also holds in the conditional case as stated below

Lemma 1 ([BSW03, FR12, CKLR11])**.** *For any joint distribution $(X, Z) \in \{0,1\}^n \times \{0,1\}^m$ and any ϵ, δ, k, s*

$$H_{\epsilon,s}^{\mathsf{Metric},det,[0,1]}(X|Z) \geqslant k \quad \Rightarrow \quad H_{\epsilon+\delta,s\cdot\delta^2/(m+n)}^{\mathsf{HILL}}(X|Z) \geqslant k$$

Note that in Definition 2 of HILL entropy, we only consider security against probabilistic boolean distinguishers (as $\sim_{\epsilon,s}$ was defined this way), whereas in Definiton 3 of metric entropy we make the class of distinguishers explicit. The reason for this is that in the definition of HILL entropy the class of distinguishers considered is irrelevant (except for a small additive degradation in circuit size, cf. [FR12, Lemma 2.1]).[8] Unlike for HILL, for metric entropy the choice of the distinguisher class does matter. In particular, deterministic boolean metric entropy $H_{\epsilon,s}^{\mathsf{Metric},det,\{0,1\}}(X|Y) \geqslant k$ is only known to imply deterministic real-valued metric entropy $H_{\epsilon+\delta,s}^{\mathsf{Metric},det,[0,1]}(X|Y) \geqslant k - \log(\delta^{-1})$, i.e., we must allow for a $\delta > 0$ loss in distinguishing advantage, and this will at the same time result in a loss of $\log(\delta^{-1})$ in the amount of entropy. For this reason, it is crucial that in Theorem 2 we show that unpredictability entropy implies deterministic *real-valued* metric entropy, so we can then apply Lemma 1 to get the same amount of HILL entropy. Dealing with real-valued distinguishers is the main source of technical difficulty in the proof of the Theorem 2, proving the analogous statement for deterministic *boolean* distinguishers is much simpler.

3 Known Results on Provably Secure Key-Derivation

We say that a cryptographic scheme has security α, if no adversary (from some class of adversaries like all polynomial size circuits) can win some security game with advantage $\geqslant \alpha$ if the scheme is instantiated with a uniformly random string.[9] Below we will distinguish between *unpredictability*

[8] This easily follows from the fact that in the definition (4) of computational indistinguishability the choice of the distinguisher class is irrelevant.

[9] We'll call this string "key". Though in many settings (in particular when keys are not simply uniform random strings, like in public-key crypto) this string is not used as a key directly, but one rather should think of it as the randomness used to sample the actual keys.

applications, where the advantage bounds the probability of winning some security game (a typical example are digital signature schemes, where the game captures the existential unforgeability under chosen message attacks), and *indistinguishability* applications, where the advantage bounds the distinguishing advantage from some ideal object (a typical example is the security definition of pseudorandom generators or functions).

3.1 Key-Derivation from Min-Entropy

Strong Extractors. Let (X, Z) be a source where $\widetilde{H}_\infty(X|Z) \geqslant k$, or equivalently, no adversary can guess X given Z with probability better than 2^{-k} (cf. Def. 1). Consider the case where we want to derive a key $K = h(X, S)$ that is statistically close to uniform given (Z, S). For example, X could be some physical source (like statistics from keystrokes) from which we want to generate almost uniform randomness. Here Z models potential side-information the adversary might have on X. This setting is very well understood, and such a key can be derived using a strong extractor as defined below.

Definition 5 ([NZ93], [DORS08]). *A function* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$ *is an average-case* (k, ϵ)*-strong extractor if for every distribution* (X, Z) *over* $\{0,1\}^n \times \{0,1\}^m$ *with* $\widetilde{H}_\infty(X|Z) \geqslant k$ *and* $S \sim U_d$*, the distribution* $(\mathsf{Ext}(X, S), S, Z)$ *has statistical distance* ϵ *to* (U_ℓ, S, Z)*.*

Extractors Ext as above exist with $\ell = k - 2\log(1/\epsilon)$ [HILL99]. Thus, from any (X, Z) where $\widetilde{H}_\infty(X|Z) \geqslant k$ we can extract a key $K = \mathsf{Ext}(X, S)$ of length $k - 2\log(1/\epsilon)$ that is ϵ close to uniform [HILL99]. The entropy gap $2\log(1/\epsilon)$ is optimal by the so called "RT-bound" [RTS00], even if we assume the source is efficiently samplable [DPW14].

If instead of using a uniform ℓ bit key for an α secure scheme, we use a key that is ϵ close to uniform, the scheme will still be at least $\beta = \alpha + \epsilon$ secure. In order to get security β that is of the same order as α, we thus must set $\epsilon \approx \alpha$. When the available amount k of min-entropy is small, for example when dealing with biometric data [DORS08, BDK$^+$05], a loss of $2\log(1/\epsilon)$ bits (that's 160 bits for a typical security level $\epsilon = 2^{-80}$) is often unacceptable.

Condensers. The above bound is basically tight for many *indistinguishability* applications like pseudorandom generators or pseudorandom functions.[10] Fortunately, for many applications a close to uniform key is not necessary, and a key $|K|$ with min-entropy $|K| - \Delta$ for some small Δ is basically as good as a uniform one. This is the case for all *unpredictability* applications, which includes OWFs, digital-signatures and MACs.[11] It's not hard to show that if the scheme is α secure with a uniform key it remains at least $\beta = \alpha 2^\Delta$ secure (against the same class of attackers) if instantiated with any key K that has $|K| - \Delta$ bits of min-entropy.[12] Thus, for unpredictability applications we don't have

[10] For example, consider a pseudorandom function $\mathsf{F} : \{0,1\}^k \times \{0,1\}^a \to \{0,1\}$ and a key K that is uniform over all keys where $\mathsf{F}(K, 0) = 0$, this distribution is $\epsilon \approx 1/2$ close to uniform and has min-entropy $\approx |K| - 1$, but the security breaks completely as one can distinguish $\mathsf{F}(U_k, .)$ from $\mathsf{F}(K, .)$ with advantage $\beta \approx 1/2$ (by quering on input 0, and outputting 1 iff the output is 0).

[11] [DY13] identify an interesting class of applications called "square-friendly", this class contains all unpredictability applications, and some indistinguishability applications like weak PRFs (which are PRFs that can only be queried on random inputs). This class of applications remains somewhat secure even for a small entropy gap Δ: For $\Delta = 1$ the security is $\beta \approx \sqrt{\alpha}$. This is worse that the $\beta = 2\alpha$ for unpredictability applications, but much better than the complete loss of security $\beta \approx 1/2$ required for some indistinguishability apps like (standard) PRFs.

[12] Assume some adversary breaks the scheme, say, forges a signature, with advantage β if the key comes from the distribution K. If we sample a uniform key instead, it will have the same distribution as K conditioned on an event that holds with probability $2^{-\Delta}$, and thus this adversary will still break the scheme with probability $\beta/2^\Delta$.

to extract an almost uniform key, but "condensing" X into a key with $|K| - \Delta$ bits of min-entropy for some small Δ is enough.

[DPW14] show that a $(\log\epsilon + 1)$-wise independent hash function $\mathsf{Cond} : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^\ell$ is a condenser with the following parameters. For any (X, Z) where $\widetilde{H}_\infty(X|Z) \geqslant \ell$, for a random seed S (used to sample a $(\log\epsilon + 1)$-wise independent hash function), the distribution $(\mathsf{Cond}(X, S), S)$ is ϵ close to a distribution (Y, S) where $\widetilde{H}_\infty(Y|Z) \geqslant \ell - \log\log(1/\epsilon)$. Using such an ℓ bit key (condensed from a source with ℓ bits min-entropy) for an unpredictability application that is α secure (when using a uniform ℓ bit key), we get security $\beta \leqslant \alpha 2^{\log\log(1/\epsilon)} + \epsilon$, which setting $\epsilon = \alpha$ gives $\beta \leqslant \alpha(1 + \log(1/\alpha))$ security, thus, security degrades only by a logarithmic factor.

3.2 Key-Derivation from Computational Entropy

The bounds discussed in this section are summarised in Figures 1 and 2 in Appendix A. The last row of Figure 2 is the new result proven in this paper.

HILL Entropy. As already discussed in the introduction, often we want to derive a key from a distribution (X, Z) where there's no "real" min-entropy at all $\widetilde{H}_\infty(X|Z) = 0$. This is for example the case when Z is the transcript (that can be observed by an adversary) of a key-exchange protocol like Diffie-Hellman, where the agreed value $X = g^{ab}$ is determined by the transcript $Z = (g^a, g^b)$ [Kra10,GKR04]. Another setting where this can be the case is in the context of side-channel attacks, where the leakage Z from a device can completely determine its internal state X.

If $X|Z$ has k bits of HILL entropy, i.e., is computationally indistinguishable from having min-entropy k (cf. Def. 2) we can derive keys exactly as described above assuming $X|Z$ had k bits of min-entropy. In particular, if $X|Z$ has $|K| + 2\log(1/\epsilon)$ bits of HILL entropy for some negligible ϵ, we can derive a key K that is pseudorandom, and if $X|Z$ has $|K| + \log\log(1/\epsilon)$ bits of HILL entropy, we can derive a key that is almost as good as a uniform one for any unpredictability application.

Unpredictability Entropy. Clearly, the minimal assumption we must make on a distribution $(X, Z) \in \{0,1\}^n \times \{0,1\}^m$ for any key derivation to be possible at all is that X is hard to compute given Z, that is, $X|Z$ must have some unpredictability entropy as in Definition 4. Goldreich and Levin [GL89] show how to generate pseudorandom bits from such a source. In particular, the Goldreich-Levin theorem implies that if $X|Z$ has at least $2\log\epsilon^{-1}$ bits of list-unpredictability, then the inner product $R^T X$ of X with a random vector R is ϵ indistinguishable from uniformly random (the loss in circuit size is $poly(n, m)/\epsilon^4$). Using the chain rule for unpredictability entropy,[13] we can generate an $\ell = k - 2\log\epsilon^{-1}$ bit long pseudorandom string that is $\ell\epsilon$ indistinguishable (the extra ℓ factor comes from taking the union bound over all bits) from uniform.

Thus, we can turn k bits of list-unpredictability into $k - 2\log\epsilon^{-1}$ bits of pseudorandom bits (and thus also that much HILL entropy) with quality roughly ϵ. The question whether it's possible to generate significantly more than $k - 2\log\epsilon^{-1}$ of HILL entropy from a source with k bits of (list-)unpredictability seems to have never been addressed in the literature before. The reason might be that one usually is interested in generating pseudorandom bits (not just HILL entropy), and for this, the $2\log\epsilon^{-1}$ entropy loss is inherent. The observation that for many applications high HILL

[13] Which states that if $X|Z$ has k bits of list-unpredictability, then for any (A, R) where R is independent of (X, Z), $X|(Z, A, R)$ has $k - |A|$ bits of list-unpredictability entropy. In particular, extracting ℓ inner product bits, decreases the list-unpredictability by at most ℓ.

entropy is basically as good as pseudorandomness is more recent, and recently gained attention by its usefulness in the context of leakage-resilient cryptography [DP08, DY13].

In this paper we prove that it's in fact possible to turn almost all list-unpredictability into HILL entropy.

4 Condensing Unpredictability

Below we state Theorem 1 whose proof is in Appendix B, but first, let us give some intuition. Let $X|Z$ have k bits of list-unpredictability, and assume we start extracting Goldreich-Levin hardcore bits $A_1, A_2, \ldots$ by taking inner products $A_i = R_i^T X$ for random R_i. The first extracted bits $A_1, A_2, \ldots$ will be pseudorandom (given the R_i and Z), but with every extracted bit, the list-unpredictability can also decrease by one bit. As the GL theorem requires at least $2\log\epsilon^{-1}$ bits of list-unpredictability to extract an ϵ secure pseudorandom bit, we must stop after $k - 2\log\epsilon^{-1}$ bits. In particular, the more we extract, the worse the pseudorandomness of the extracted string becomes. Unlike the original GL theorem, in our Theorem 1 we only argue about the unpredictability of the extracted string, and unpredictability entropy has the nice property that it can never decrease, i.e., predicting $A_1, \ldots, A_{i+1}$ is always at least as hard as predicting $A_1, \ldots, A_i$. Thus, despite the fact that once i approaches k it becomes easier and easier to predict A_i (given $A_1, \ldots, A_{i-1}, Z$ and the R_i's)[14] this hardness will still add up to $k - O(1)$ bits of unpredictability entropy.

The proof is by contradiction, we assume that $A_1, \ldots, A_k$ can be predicted with advantage 2^{-k+3} (i.e., does not have $k-3$ bits of unpredictability), and then use such a predictor to predict X with advantage $> 2^{-k}$, contradicting the k bit list-unpredictability of $X|Z$.

If $A_1, \ldots, A_k$ can be predicted as above, then there must be an index j s.t. A_j can be predicted with good probability conditioned on $A_1, \ldots, A_{j-1}$ being correctly predicted. We then can use the Goldreich-Levin theorem, which tells us how to find X given such a predictor. Unfortunately, j can be close to k, and to apply the GL theorem, we first need to find the right values for $A_1, \ldots, A_{j-1}$ on which we condition, and also can only use the predictor's guess for A_j if it was correct on the first $j-1$ bits. We have no better strategy for this than trying all possible values, and this is the reason why the loss in circuit size in Theorem 1 depends on 2^k.

In our proof, instead of using the Goldreich-Levin theorem, we will actually use a more fine-grained variant due to Hast which allows to distinguish between errors and erasures (i.e., cases where we know that we don't have any good guess. As outlined above, this will be the case whenever the predictor's guess for the first $j-1$ inner products was wrong, and thus we can't assume anything about the jth guess being correct). This will give a much better quantitative bound than what seems possible using GL.

Theorem 1 (Condensing Upredictability Entropy)**.** *Consider any distribution* (X, Z) *over* $\{0,1\}^n \times \{0,1\}^m$ *where*

$$H^{*\mathsf{unp}}_{\epsilon,s}(X|Z) \geqslant k$$

then for a random $R \leftarrow \{0,1\}^{k\times n}$

$$H^{\mathsf{unp}}_{\epsilon,t}(R.X|Z,R) \geqslant k - \Delta$$

[14] The only thing we know about the last extracted bit A_k is that it cannot be predicted with advantage $\geqslant 0.75$, more generally, A_{k-j} cannot be predicted with advantage $1/2 + 1/2^{j+2}$.

where[15]

$$t = \frac{s}{2^{2k}\,\mathrm{poly}(m,n)} \qquad , \qquad \Delta = 3$$

5 High Unpredictability implies Metric Entropy

In this section we state our main results, showing that k bits of unpredictability entropy imply the same amount of HILL entropy, with a loss exponential in the "entropy gap". The proof is in Appendix C.

Theorem 2 (Unpredictability Entropy Implies HILL Entropy)**.** *For any distribution* (X,Z) *over* $\{0,1\}^n \times \{0,1\}^m$*, if* $X|Z$ *has unpredictability entropy*

$$H^{\mathsf{unp}}_{\gamma,s}(X|Z) \geqslant k \tag{6}$$

then, with $\Delta = n-k$ *denoting the entropy gap,* $X|Z$ *has (real valued, deterministic) metric entropy*

$$H^{\mathsf{Metric},det,[0,1]}_{\epsilon+\gamma,t}(X|Z) \geqslant k \quad \textit{for} \quad t = \Omega\left(s \cdot \frac{\epsilon^5}{2^{5\Delta}\log^2\left(2^{\Delta}\epsilon^{-1}\right)}\right) \tag{7}$$

By Lemma 1 this further implies that $X|Z$ *has, for any* $\delta > 0$*, HILL entropy*

$$H^{\mathsf{HILL}}_{\epsilon+\delta+\gamma,\Omega(t\delta^2/(n+m))}(X|Z) \geqslant k$$

which for $\epsilon = \delta = \gamma$ *is*

$$H^{\mathsf{HILL}}_{3\epsilon,\Omega(s\cdot\epsilon^7/2^{5\Delta}(n+m)\log^2(2^{\Delta}\epsilon^{-1}))}(X|Z) \geqslant k$$

References

[BDK+05] Xavier Boyen, Yevgeniy Dodis, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Secure remote authentication using biometric data. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 147–163. Springer, May 2005.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[BSW03] B. Barak, R. Shaltiel, and A. Wigderson. Computational Analogues of Entropy. In S. Arora, K. Jansen, J. D. P. Rolim, and A. Sahai, editors, *RANDOM-APPROX 03*, volume 2764 of *LNCS*, pages 200–215. Springer, 2003.

[CKLR11] Kai-Min Chung, Yael Tauman Kalai, Feng-Hao Liu, and Ran Raz. Memory delegation. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 151–168. Springer, August 2011.

[DGH+04] Yevgeniy Dodis, Rosario Gennaro, Johan Håstad, Hugo Krawczyk, and Tal Rabin. Randomness extraction and key derivation using the CBC, cascade and HMAC modes. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 494–510. Springer, August 2004.

[15] We can set Δ to be any constant > 1 here, but choosing a smaller Δ would imply a smaller t.

[DORS08] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th FOCS*, pages 293–302. IEEE Computer Society Press, October 2008.

[DPW14] Yevgeniy Dodis, Krzysztof Pietrzak, and Daniel Wichs. Key derivation without entropy waste. In *EUROCRYPT 14*, LNCS. Springer, 2014.

[DY13] Yevgeniy Dodis and Yu Yu. Overcoming weak expectations. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 1–22. Springer, March 2013.

[FR12] Benjamin Fuller and Leonid Reyzin. Computational entropy and information leakage. Cryptology ePrint Archive, Report 2012/466, 2012. http://eprint.iacr.org/.

[GKR04] Rosario Gennaro, Hugo Krawczyk, and Tal Rabin. Secure Hashed Diffie-Hellman over non-DDH groups. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 361–381. Springer, May 2004.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989.

[Has03] Gustav Hast. Nearly one-sided tests and the Goldreich-Levin predicate. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 195–210. Springer, May 2003.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HLR07a] C.-Y. Hsiao, C.-J. Lu, and L. Reyzin. Conditional Computational Entropy, or Toward Separating Pseudoentropy from Compressibility. In M. Naor, editor, *EUROCRYPT 07*, volume 4515 of *LNCS*, pages 169–186. Springer, 2007.

[HLR07b] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 169–186. Springer, May 2007.

[Kra10] Hugo Krawczyk. Cryptographic extraction and key derivation: The HKDF scheme. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 631–648. Springer, August 2010.

[NZ93] Noam Nisan and David Zuckerman. More deterministic simulation in logspace. In *25th ACM STOC*, pages 235–244. ACM Press, May 1993.

[RTS00] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM J. Discrete Math.*, 13(1):2–24, 2000.

A Figures

Deriving a (pseudo)random key of length $\lvert K\rvert = k - 2\log\epsilon^{-1}$ from a source $(X, Z) \in \{0,1\}^n \times \{0,1\}^m$ where $X\lvert Z$ has k bits (min/HILL/list-unpredictability) entropy			
Entropy type	Entropy quantity and quality of source	Derive key K of length $k - 2\log\epsilon^{-1}$ as	Quality of derived key $H^{\mathsf{HILL}}_{\epsilon',s'}(K\lvert Z, S) = k - 2\log\epsilon^{-1} = \lvert K\rvert$ equivalently $(K, Z, S) \sim_{\epsilon',s'} (U_{\lvert K\rvert}, Z, S)$
min	$\widetilde{H}_\infty(X\lvert Z) = k$	$K = \mathsf{Ext}(X, S)$	$\epsilon' = \epsilon \qquad s' = \infty$
HILL	$H^{\mathsf{HILL}}_{\delta,s}(X\lvert Z) = k$	$K = \mathsf{Ext}(X, S)$	$\epsilon' = \epsilon + \delta \qquad s' \approx s$
Unpredict.	$H^{*\mathsf{unp}}_{\delta,s}(X\lvert Z) = k$	$K = \mathsf{GL}(X, S) = S^T X$	$\epsilon' = m\epsilon + \delta \qquad s' = s \cdot \epsilon^4/poly(m, n)$

Figure 1: Bounds on deriving a (pseudo)random key K of length $|K| = k - 2\log\epsilon^{-1}$ bit from a source $X|Z$ with k bits of min, HILL or list-unpredictability entropy. Ext is a strong extractor (e.g. leftover hashing), and GL denotes the Goldreich-Levin construction, which for $X \in \{0,1\}^n$ and $S \in \{0,1\}^{n\times|K|}$ is simply defined as $\mathsf{GL}(X, S) = S^T X$. Leftover hashing requires a seed of length $|S| = 2n$ (extractors with a much shorter seed $|S| = O(\log n + \log\epsilon^{-1})$ that extract $k - 2\log\epsilon^{-1} - O(1)$ bits also exist), whereas Goldreich-Levin requires a longer $|S| = |K|n$ bit seed. The above bound for HILL entropy even holds if $X|Z$ only has k bits of probabilistic boolean metric entropy (a notion implying the same amount of HILL entropy, albeit with a loss in circuit size), as shown in Theorem 2.5 of [FR12]

Deriving k bit key K with high HILL entropy from $X\lvert Z$ with k bits (min/HILL/list-unpredictability) entropy			
Entropy type	Entropy quantity and quality of soucre	Derive key of length $\lvert K\rvert = k$ as	Quantity and quality of HILL entropy of K $H^{\mathsf{HILL}}_{\epsilon',s'}(K\lvert Z, S) \geqslant k - \Delta$
min	$\widetilde{H}_\infty(X\lvert Z) = k$	$K = \mathsf{Cond}(X, S)$	$\epsilon' = \epsilon \qquad s' = \infty \qquad \Delta = \log\log\epsilon^{-1}$
HILL	$H^{\mathsf{HILL}}_{\delta,s}(X\lvert Z) = k$	$K = \mathsf{Cond}(X, S)$	$\epsilon' = \epsilon + \delta \qquad s' \approx s \qquad \Delta = \log\log\epsilon^{-1}$
Unpredict.	$H^{*\mathsf{unp}}_{\delta,s}(X\lvert Z) = k$	$K = \mathsf{GL}(X, S) = S^T X$	$\epsilon' = \epsilon + \delta \qquad s' = s \cdot \epsilon^7/2^{2k} poly(m, n) \qquad \Delta = 3$

Figure 2: Bounds on deriving a key of length k with min (or HILL) entropy $k - \Delta$ from a source $X|Z$ with k bits of min, HILL or unpredictability entropy. Cond denotes a $(\log\epsilon + 1)$ wise independent hash function, which is shown to be a good condenser (as stated in the table) for min-entropy in [DPW14]. The bounds for HILL entropy follow directly from the bound for min-entropy. The last row follows from the results in this paper as stated in Proposition 1.

B Proof of Theorem 1

We will use the following theorem due Hast [Has03] on decoding Hadamard code with errors and erasures.

Theorem 3 ([Has03]). *There is an algorithm* LD *that, on input* l *and* n *and with oracle access to a binary Hadamard code of* x *(where* $|x| = n$*) with an* e*-fraction of errors and an* s*-fraction of erasures, can output a list of* 2^l *elements in time* $O(nl2^l)$ *asking* $n2^l$ *oracle queries such that the probability that* x *is contained in the list is at least* 0.8 *if* $l \geqslant \log_2(20n(e + c)/(c - e)^2 + 1)$*, where* $c = 1 - s - e$ *(the fraction of the correct answers from the oracle).*

We'll often consider sequences $v_1, v_2, \ldots$ of values and will use the notation v_a^b to denote $(v_a, \ldots, v_b)$, with $v_a^b = \emptyset$ if $a > b$. v^b is short for $v_1^b = (v_1, \ldots, v_b)$.

of Theorem 1. It's sufficient to prove the theorem for $\epsilon = 0$, the general case $\epsilon \geqslant 0$ then follows directly by the definition of unpredictability entropy. To prove the theorem we'll prove its contraposition

$$H_t^{\mathsf{unp}}(R.X|Z, R) < k - \Delta \quad \Rightarrow \quad H_s^{*\mathsf{unp}}(X|Z) < k \tag{8}$$

The left-hand side of (8) means there exists a circuit A of size $|\mathsf{A}| \leqslant t$ such that

$$\Pr_{(x,z) \leftarrow (X,Z), r \leftarrow \{0,1\}^{k \times n}}[\mathsf{A}(z, r) = r.x] \geqslant 2^{-k+\Delta} \tag{9}$$

It will be convenient to assume that A initially flips a coin b, and if $b = 0$ outputs a uniformly random guess. This loses at most a factor 2 in A's advantage, i.e.,

$$\Pr_{(x,z) \leftarrow (X,Z), r \leftarrow \{0,1\}^{k \times n}}[\mathsf{A}(z, r) = r.x] \geqslant 2^{-k+\Delta-1} \tag{10}$$

but now we can assume that for any z, r and $w \in \{0,1\}^k$

$$\Pr[\mathsf{A}(z, r) = w] \geqslant 2^{-k-1} \tag{11}$$

Using Markov eq.(10) gives us

$$\Pr_{(x,z) \leftarrow (X,Z)}[\Pr_{r \leftarrow \{0,1\}^{k \times n}}[\mathsf{A}(z, r) = r.x] \geqslant 2^{-k+\Delta-2}] \geqslant 2^{-k+\Delta-2} \tag{12}$$

We call $(x, z) \in \text{supp}[(X, Z)]$ "good" if

$$(x, z) \text{ is good} \iff \Pr_{r \leftarrow \{0,1\}^{k \times n}}[\mathsf{A}(z, r) = r.x] \geqslant 2^{-k+\Delta-2} \tag{13}$$

Note that by eq.(12), $(z, x) \leftarrow (Z, X)$ is good with probability $\geqslant 2^{-k+\Delta-2}$.

We will use A to construct a new circuit B of size $s = O(t2^{2k} \operatorname{poly}(n))$ where

$$\Pr_{(x,z) \leftarrow (X,Z)}[\mathsf{B}(z) = x \ |(x, z) \text{ is good}] > 1/2 \tag{14}$$

Which with (14) and (12) further gives

$$\begin{aligned} \Pr_{(x,z) \leftarrow (X,Z)}[\mathsf{B}(z) = x] &= \Pr[\mathsf{B}(z) = x|(x, z) \text{ is good}] \cdot \Pr[(x, z) \text{ is good}] \\ &> 2^{-1} \cdot 2^{-k+\Delta-2} = 2^{-k+\Delta-3} \end{aligned} \tag{15}$$

contradicting the right-hand side of (8), and thus proving the theorem.

We'll now construct B satisfying (14), for this, consider any good (x, z). Let $R = R^k = (R_1, \ldots, R_k)$ be uniformly random and let $A = A^k = (A_1, \ldots, A_k)$ where $A_i = R_i.x$.

Let $\hat{A} \leftarrow \mathsf{A}(z, R)$ and define $\epsilon_i = \Pr_R[\hat{A}_i = A_i|\hat{A}^{i-1} = A^{i-1}]$. Using (13) in the last step

$$\prod_{i=1}^k \epsilon_i = \Pr_R[A = \hat{A}] = \Pr_R[\mathsf{A}(z, R) = R.x] \geqslant 2^{-k+\Delta-2}$$

Thus, here exists an i s.t., $\epsilon_i \geqslant 2^{\frac{-k+\Delta-2}{k}} = \frac{1}{2} + \delta$ with $\delta \approx \frac{\Delta-2}{k} \cdot \frac{\ln(2)}{2}$. We fix this i (we don't know which i is good, and later will simply try all of them). Then

$$\mathbb{E}_{R^{i-1}}[\Pr_{R_i, R_{i+1}^k}[\hat{A}_i = A_i \mid \hat{A}^{i-1} = A^{i-1}]] \geqslant 1/2 + \delta$$

Using Markov

$$\Pr_{R^{i-1}}[\Pr_{R_i, R_{i+1}^k}[\hat{A}_i = A_i \mid \hat{A}^{i-1} = A^{i-1}] \geqslant 1/2 + \delta/2] \geqslant \frac{\delta}{2} \tag{16}$$

We call r^{i-1} good if (note that by the previous equation a random r^{i-1} is good with probability $\geqslant \delta/2$).

$$r^{i-1} \text{ is good} \iff \Pr_{R_i, R_{i+1}^k}[\hat{A}_i = A_i \mid \hat{A}^{i-1} = A^{i-1}] \geqslant 1/2 + \delta/2 \tag{17}$$

From now on, we fix some good r^{i-1} and assume we know $a^{i-1} = r^{i-1}.x$ (later we'll simply try all possible choices for a^{i-1}).

We define a predictor $\mathsf{P}_i(r_i)$ that tries to predict $r_i.x$ given a random r_i (and also knows z, r^{i-1}, a^{i-1} as above) as follows

1. Sample random $r_{i+1}^k \leftarrow R_{i+1}^k$
2. Invoke $\hat{A}^k \leftarrow \mathsf{A}(z, r^{(i)}, x)$. Note that $r^{(i)} = (r^{i-1}, r_i, r_{i+1}^k)$ consists of the fixed r^{i-1}, the input r_i and the randomly sampled r_{i+1}^k.
3. if $\hat{A}^{i-1} = a^{i-1}$ output $\hat{A}_i$, otherwise output $\perp$.

Using (11), which implies $\Pr[\hat{A}^{i-1} = a^{i-1}] \geqslant 2^{-i}$, and (17) we can lower bound P_i's rate and advantage as

$$\begin{aligned} \Pr_{R_i}[\mathsf{P}_i(R_i) \neq \perp] &= \Pr[\hat{A}^{i-1} = a^{i-1}] \geqslant 2^{-i}, \\ \Pr_{R_i}[\mathsf{P}_i(R_i) = R_i.x] &\geqslant \Pr[\hat{A}^{i-1} = a^{i-1}](\frac{1}{2} + \delta/2). \end{aligned} \tag{18}$$

In terms of Theorem 3, we have a binary Hadamard code with $e + c = \Pr[\hat{A}^{i-1} = a^{i-1}]$, $c - e = \delta \cdot \Pr[\hat{A}^{i-1} = a^{i-1}]$, which implies that $(e+c)/(c-e)^2 \leqslant \frac{2^i}{\delta^2}$.

Now Theorem 3 implies that given such a predictor P we can output a list that contains x with probability > 0.8 in time $O(2^i \operatorname{poly}(m,n)) = O(2^k \operatorname{poly}(m,n))$, as we assume access to an oracle Eq with outputs 1 on input x and 0 otherwise, we can find x in this list with the same probability.

Using this, we can now construct an algorithm as claimed in (14) as follows: B will sample $i \in \{1, \ldots, k\}$ and then r^{i-1} at random. Then B calls P_i with all possible $a^{i-1} \in \{0,1\}^{i-1}$. We note that with probability $\delta/2k$ (we lose a factor k for the guess of i, and $\delta/2$ is the probability of sampling a good r^{i-1}) the predictor P_i will satisfy (18).

If x is not found, B repeats the above process, but stops if x is not found after $2k/\delta$ iterations. The success probability of B is $\approx (1 - 1/e)0.8 > 0.5$ as claimed, the overall running time we get is $O(2^{2k} \operatorname{poly}(m,n))$. □ □

C Proof of Theorem 2

It's sufficient to prove the theorem for $\gamma = 0$, the case $\gamma > 0$ then follows directly by definition of unpredictability entropy. Suppose for the sake of contradiction that (7) does not hold. That is, $H^{\mathsf{Metric},det,[0,1]}_{t,\epsilon}(X|Z) < k$, which means that there exists a distinguisher $\mathsf{D} : \{0,1\}^n \times \{0,1\}^m \to [0,1]$ of size t that satisfies

$$\mathbb{E}\mathsf{D}(X,Z) - \mathbb{E}\mathsf{D}(Y,Z) \geqslant \epsilon \quad \forall (Y,Z): \ \widetilde{H}_\infty(Y|Z) \geqslant k. \tag{19}$$

We will show how to construct an efficient algorithm that given Z uses D to predict X with probability at least 2^{-k}, contradicting (6). The core of the algorithm is the procedure Predictor described below.

```
Function PREDICTOR($z, \mathsf{D}', \ell$)
    Input  : $z \leftarrow Z$, $[0,2]$-valued distinguisher $\mathsf{D}'$
    Output: $x \in \{0,1\}^n$
    $b \leftarrow 1$, $i \leftarrow 1$
    while $b \neq 0$ and $i < \ell$ do
        $x \leftarrow \{0,1\}^n$
        $b \leftarrow$ BernoulliDistribution($\mathsf{D}'(x,z)/2$)          /* outputs 1 w.p.  $\mathsf{D}'(x,z)/2$ */
        if $b = 0$ then
            $i \leftarrow i + 1$
        else
            return $x$
        end
    end
    return $\perp$
```

Predictor(Z, D, ℓ) samples an element $x \in \{0,1\}^n$ according to some probability distribution. This distribution captures the following intuition: as the advantage $\mathbb{E}\mathsf{D}(X,Z) - \mathbb{E}\mathsf{D}(Y,Z)$ is positive (as assumed in (19)), we know that x being the correct guess for X is positively correlated with the value $\mathsf{D}(x,Z)$. The probability that Predictor(Z, D, ℓ) returns some particular value x as guess for X will be linear in $\mathsf{D}(x,Z)$.

Predictor(Z, D, ℓ) may also output $\perp$, which means it failed to sample an x according to this distribution. The probability of outputting $\perp$ goes exponentially fast to 0 as ℓ grows.

A toy example: predicting X when Z is empty and D is boolean. Suppose that $\mathbb{E}\mathsf{D}(X) - \mathbb{E}\mathsf{D}(Y) \geqslant \epsilon$ for all Y such that $H_\infty(Y) \geqslant k$. And assume that $\mathsf{D}(.)$ is boolean (not real valued as in our theorem). Then Predictor$(\emptyset, \mathsf{D}, \ell)$ will output a guess for X that (if it's not $\perp$) is a random value x satisfying $\mathsf{D}(x) = 1$. The probability that this guess for X is correct equals $\mathbb{E}\mathsf{D}(X)/|D|$ where $|D| = \sum_x \mathsf{D}(x)$. Consider now the distribution Y of min-entropy k that maximizes $\mathbb{E}\mathsf{D}(Y)$. We can assume that Y is flat and supported on those 2^k elements x for which the value $\mathsf{D}(x)$ is the biggest possible. Observe that since $\mathbb{E}\mathsf{D}(X) - \mathbb{E}\mathsf{D}(Y) > 0$, we have $\mathbb{E}\mathsf{D}(Y) < 1$ and since D is boolean, the support of Y contains all the elements x satisfying $\mathsf{D}(x) = 1$. Therefore we obtain

$\mathbb{E}\mathsf{D}(Y) = 2^{-k}|D|$. Now we can estimate the predicting probability from below as follows:

$$\Pr[X \text{ is predicted correctly}] = \frac{\mathbb{E}\mathsf{D}(X)}{|D|} \geqslant \frac{\mathbb{E}\mathsf{D}(Y)+\epsilon}{|D|} = 2^{-k} + \frac{\epsilon}{|D|}$$

The above probability holds for $\ell = \infty$, i.e., when predictor never outputs $\perp$. For efficiency reasons, we must use a finite, and not too big ℓ. The predictor will output $\perp$ with probability $(1-2^{-n}|D|)^{\ell}$ and thus

$$\Pr[\text{we predcit } X \text{ in time } \mathcal{O}(\ell \cdot \text{time}(D))] = \left(2^{-k} + \frac{\epsilon}{|D|}\right)\left(1 - \left(1-2^{-n}|D|\right)^{\ell}\right)$$

With a little bit of effort one can prove that setting $\ell = 1 + 2^{n-k}/\epsilon \approx 2^{\Delta}/\epsilon$ yields the success probability 2^{-k} independently of $|D|$.

Proof in general case - important issues Unfortunately, what we have proven above cannot be generalized easily to the case considered in the theorem, there are two obstacles. First, in the theorem we consider a conditional distribution $X|Z$ (i.e., the conditional part Z is not empty as above). Unfortunately we cannot simply make the above argument separately for all possible choices $Z = z$ of the conditional part, as we cannot guarantee that the conditional advantages $\epsilon(z) = \mathbb{E}\mathsf{D}(X|Z=z,z) - \mathbb{E}\mathsf{D}(Y|Z=z,z)$ are *all* positive; we only know that their average $\epsilon = \mathbb{E}_{z\leftarrow Z}\epsilon(z)$ is positive. Second, so far we assumed that D is boolean. This would only prove the theorem where the derived entropy in (7) is against deterministic *boolean* distinguishers, and this is not enough to conclude that we have the same amount of HILL entropy as discussed in Section 2.0.4.

Actual proof - preliminaries For real-valued distinguishers in the conditional case, just invoking $\textsc{Predictor}(Z, \mathsf{D}, \ell)$ on a D satisfying (19), will not give a predictor for X with advantage $> 2^{-k}$ in general. Instead, we first have to transform D into a new distingusiher D' that has the same distinguishing advantage, and for which we can prove that the predictor will work.

The way in which we modify D depends on the distribution $Y|Z$ that minimizes the left-hand side of (19). This distribution can be characterized as follows:

Lemma 2. *Given* $\mathsf{D} : \{0,1\}^n \times \{0,1\}^m \to [0,1]$ *consider the following optimization problem*

$$\begin{aligned} \max_{Y|Z} \quad & \mathbb{E}\mathsf{D}(Y,Z) \\ \text{s.t.} \quad & \widetilde{H}_{\infty}(Y|Z) \geqslant k \end{aligned} \tag{20}$$

The distribution $Y|Z = Y^*|Z$ *satisfying* $\widetilde{H}_{\infty}(Y^*|Z) = k$ *is optimal for (20) if and only if there exist real numbers* $t(z)$ *and a number* $\lambda \geqslant 0$ *such that for every* z

(a) $\sum_x \max(\mathsf{D}(x,z) - t(z), 0) = \lambda$

(b) If $0 < \mathbf{P}_{Y^*|Z=z}(x) < \max_{x'} \mathbf{P}_{Y^*|Z=z}(x')$ *then* $\mathsf{D}(x,z) = t(z)$.

(c) If $\mathbf{P}_{Y^*|Z=z}(x) = 0$ *then* $\mathsf{D}(x,z) \leqslant t(z)$

(d) If $\mathbf{P}_{Y^*|Z=z}(x) = \max_{x'} \mathbf{P}_{Y^*|Z=z}(x')$ *then* $\mathsf{D}(x,z) \geqslant t(z)$

Proof. The proof is a straightforward application of the Kuhn-Tucker conditions given in Appendix. □ □

Remark 2. *The characterization can be illustrated in an easy and elegant way. First, it says that the area under the graph of* $\mathsf{D}(x, z)$ *and above the threshold* $t(z)$ *is the same, no matter what* z *is (see Figure 3).*

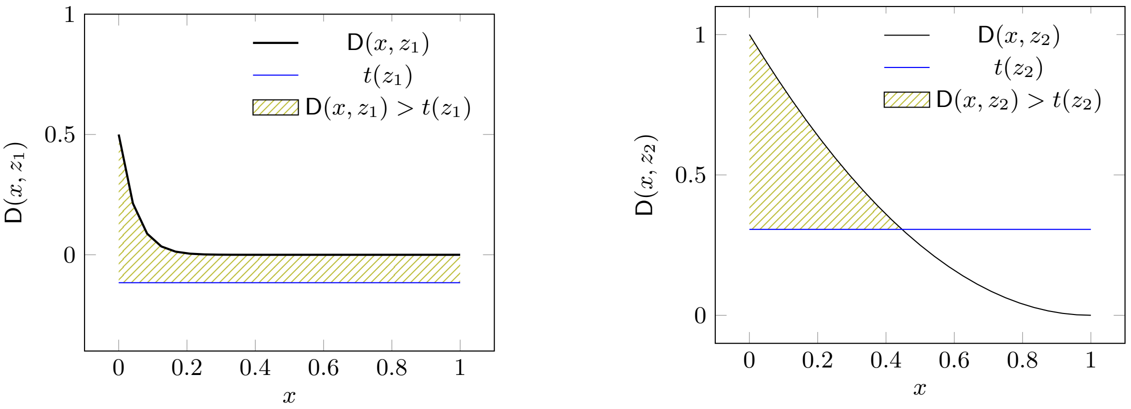

Figure 3: For every z, the (green) area under $\mathsf{D}(\cdot, z)$ and above $t(z)$ equals λ

Second, for every z *the distribution* $Y^*|Z = z$ *is flat over the set* $\{x : \mathsf{D}(x, z) > t(z)\}$ *and vanishes for* x *satisfying* $\mathsf{D}(x, z) < t(z)$*, see Fig. 4.*

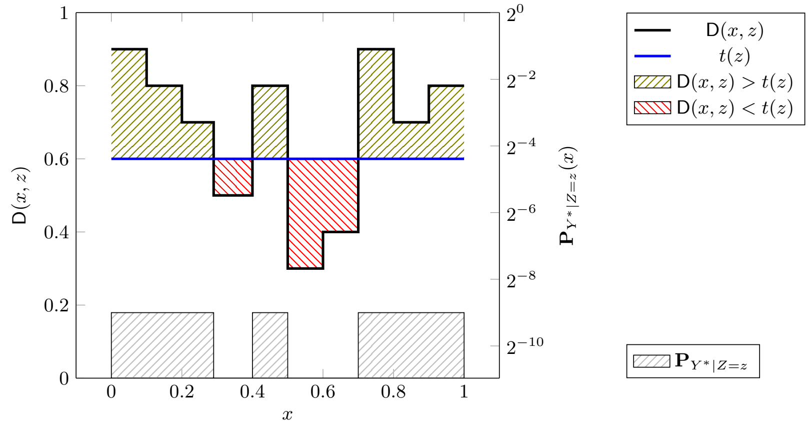

Figure 4: Relation between distinguisher $\mathsf{D}(x, z)$, threshold $t(z)$ and distribution $Y^*|Z = z$.

Note that because of “freedom" in defining the distribution on elements x satisfying $\mathsf{D}(x, z) = t(z)$ (2, point (b)), there could be many distributions $Y^*|Z$ corresponding to fixed numbers λ and $t(z)$ that satisfy the characterization above, and this way are optimal to (20) with $k = \widetilde{H}_\infty(Y^*|Z)$. For the sake of completeness we characterize bellow the all possible values of k that match to λ and $t(z)$. We note that this fact might be used to modify our nonuniform guessing algorithm into a uniform one.

Corollary 1. *Let $\mathsf{D} : \{0,1\}^n \times \{0,1\}^m \to [0,1]$ and $\lambda \in (0,1)$. Let $t(z) = t(\lambda, z)$ be the unique numbers that satisfy the condition (a) in Lemma 2. Define*

$$k(\lambda) = n - \log\left(\mathbb{E}_{z\leftarrow Z}\left[1/\mathbf{P}(\mathsf{D}(U,z) \geqslant t(z))\right]\right), \tag{21}$$

which is a non-decreasing right continuous function of λ. Let $k^-(\lambda) = \lim_{\lambda'\to\lambda^-} k(\lambda')$ and $k^+(\lambda) = \lim_{\lambda'\to\lambda^+} k(\lambda') = k(\lambda)$ be the one-sided limits. Then for every $Y^*|Z$ of min-entropy $k = \widetilde{H}_\infty(Y^*|Z)$ fulfilling (b),(c) and (d) we have $k^- \leqslant k \leqslant k^+$. Conversely, if k satisfies $k^- \leqslant k \leqslant k^+$ then there exists a distribution $Y^*|Z$ fulfilling (b),(c) and (d) such that $\widetilde{H}_\infty(Y^*|Z) = k$.

Predicting given the thresholds $t(z)$. We use the numbers $t(z)$ to modify D and then we call the procedurePredictor on the modified distinguisher. Lemma 3 below shows that we could efficiently predict X from Z, assuming we knew the numbers $t(z)$ for all z in the support of Z (later, we'll show how to efficiently approximate them)

Lemma 3. *Let $Y^*|Z$ be the distribution satisfying $\widetilde{H}_\infty(Y^*|Z) = k$ and maximizing $\mathbb{E}\mathsf{D}(Y,Z)$ over $\widetilde{H}_\infty(Y|Z) \geqslant k$, where $k < n$ and D satisfies (19). Let $t(z)$ be as in Lemma 2. Define*

$$\mathsf{D}'(x,z) = \max(\mathsf{D}(x,z) - t(z), 0) \tag{22}$$

and set $\ell = 2\cdot 2^{n-k}\epsilon^{-1}$ in the algorithm PREDICTOR*. Then we have*

$$\Pr\left(\textsc{Predictor}(Z,\mathsf{D}',\ell) = X\right) \geqslant 2^{-k}\left(1 + 2^{k-n}\epsilon\right) \tag{23}$$

Proof. We start by calculating the probability on the left-hand side of(23)

Claim 1. *For any[16] D', the algorithm* PREDICTOR *outputs X given $Z = z$ with probability*

$$\Pr_{X,Z}\left(\textsc{Predictor}(Z,\mathsf{D}',\ell) = X \,\middle|\, Z = z\right) = 2^{-n-1} g\left(\frac{\mathbb{E}\mathsf{D}'(U,z)}{2}\right)\cdot \mathbb{E}\mathsf{D}'(X|Z=z,z) \tag{24}$$

where U is uniform over $\{0,1\}^n$ and g is defined by $g(d) = \frac{1-(1-d)^\ell}{d}$ (so $g(d) \approx 1/d$ for large ℓ)

of Claim. It is easy to observe that

$$\Pr[\textsc{Predictor}(z,\mathsf{D}',\ell) = x \,|\, \textsc{Predictor}(z,\mathsf{D}',\ell) \neq \perp] = \frac{\mathsf{D}'(x,z)}{\sum\limits_x \mathsf{D}'(x,z)} \tag{25}$$

In turn, for every round $i = 1,\ldots,\ell$ of the execution, the probability that PREDICTOR stops and outputs x' is equal to $\Pr[U = x']\mathsf{D}'(x',z)/2 = 2^{-n-1}\mathsf{D}'(x',z)$, the probability that it outputs anything (and thus leaves the while loop) is thus $\sum_{x'}\Pr[U = x']\cdot\left(1 - \frac{\mathsf{D}'(x',z)}{2}\right) = 1 - \frac{\mathbb{E}\mathsf{D}'(U,z)}{2}$. So the probability of not leaving the while loop for ℓ rounds (in this case the output is $\perp$) is

$$\Pr[\textsc{Predictor}(z,\mathsf{D}',\ell) = \perp] = 1 - \left(1 - \frac{\mathbb{E}\mathsf{D}'(U,z)}{2}\right)^\ell \tag{26}$$

[16] We will only use the claim for the distinguisher D' as constructed above, but the claim holds in general.

Combining the last two formulas we obtain

$$\Pr[\textsc{Predictor}(z,\mathsf{D}') = x] = 2^{-n-1} g(\mathbb{E}\mathsf{D}'(U,z)/2)\cdot \mathsf{D}'(x,z) \tag{27}$$

Hence

$$\begin{aligned}\Pr[\textsc{Predictor}(z,\mathsf{D}') = X|Z=z] &= \sum_x \Pr[\textsc{Predictor}(z,\mathsf{D}') = x, X=x|Z=z] \\ &= \sum_x \Pr[\textsc{Predictor}(z,\mathsf{D}') = x]\Pr[X=x|Z=z] \\ &= 2^{-n-1} g(\mathbb{E}\mathsf{D}'(U,z)/2)\sum_x \mathsf{D}'(x,z)\Pr[X=x|Z=z] \\ &= 2^{-n-1} g(\mathbb{E}\mathsf{D}'(U,z)/2)\mathbb{E}\mathsf{D}'(X|Z=z,z) \qquad (28)\end{aligned}$$

and the claim follows. □ □

Now we can see why we cannot apply the algorithm Predictor using the distinguisher D satisfying only (19) directly. According to the last formula, the success probability would be an averaged sum of products $g(\mathbb{E}\mathsf{D}(U,z))\cdot\mathbb{E}\mathsf{D}(X|Z=z,z)$ over z. We know the average of the second factors of these products, but in general cannot compare the values of $\mathbb{E}\mathsf{D}(U,z)$ for different z's. The crucial observation is that the distinguisher D' we defined satisfies the same inequality (19) as D (though, D' has the range $[0,2]$ not $[0,1]$ as D). Moreover D' has a special form which allows us to simplify expression (23). The details are given in the next two claims

Claim 2. *We have* $\mathbb{E}\mathsf{D}'(X,Z)-\mathbb{E}\mathsf{D}'(Y,Z)\geqslant\epsilon$ *for all* $Y|Z:\ \widetilde{H}_\infty(Y|Z)\geqslant k$

of Claim. We argue that (a): $\mathbb{E}\mathsf{D}'(X,Z)-\mathbb{E}\mathsf{D}'(Y^*,Z)\geqslant\mathbb{E}\mathsf{D}(X,Z)-\mathbb{E}\mathsf{D}(Y^*,Z)$ and (b): $Y^*|Z$ maximizes $\mathsf{D}'(Y,Z)$ over $\widetilde{H}_\infty(Y|Z)\geqslant k$. For the proof of (a), observe that by (22) we have $\mathsf{D}'(x,z)\geqslant \mathsf{D}(x,z)-t(z)$ for every x and z. Hence $\mathbb{E}\mathsf{D}'(X,Z)\geqslant\mathbb{E}\mathsf{D}(X,Z)-t(z)$. Moreover, if $\mathsf{D}(x,z)-t(z)<0$ then Lemma 2 implies $\mathbf{P}_{Y^*|Z=z}(x)=0$ and thus $\mathbb{E}\mathsf{D}'(Y^*|Z=z,z)=\mathbb{E}\mathsf{D}(Y^*|Z=z)-t(z)$. Hence, for all z we have

$$\mathbb{E}\mathsf{D}'(X|Z=z)-\mathbb{E}\mathsf{D}'(Y^*|Z=z,z)\geqslant\mathbb{E}\mathsf{D}(X|Z=z,z)-\mathbb{E}\mathsf{D}(Y^*|Z=z,z)$$

The proof of (a) follows now by taking the average over z. The proof of (b) follows by observing that D' satisfies the characterization in (2) with $t(z)=0$ for all z. □ □

Claim 3. *The exists a number* $\lambda'\in(0,1)$ *such that* $\mathbb{E}\mathsf{D}'(U,z)=\lambda'$ *for every* z.

Proof. Lemma 2 implies $\sum_x \mathsf{D}'(x,z)=\lambda$ for every z. We can define $\lambda'=2^{-n}\lambda$ and then it remains to show $\lambda<2^n$ and $\lambda>0$. Observe that the case $t(z)<0$ in Lemma 2 is possible if and only if $\mathbf{P}_{Y^*|Z=z}(x)=\max_{x'}\mathbf{P}_{Y^*|Z=z}(x')$ for all x, which means $H_\infty(Y^*|Z=z)=n$. Since $k<n$, we have $t(z)\geqslant 0$ for at least one z and then $\lambda=\sum_x\max(\mathsf{D}(x,z)-t(z),0)\leqslant\sum_x\mathsf{D}(x,z)$ which essentially means $\lambda\leqslant 2^n$. Lemma 2 guarantees that $\lambda\geqslant 0$, therefore we need to show that $\lambda\notin\{0,2^n\}$. Observe that if $\lambda=0$ then the condition $\sum_x\mathsf{D}'(x,z)=\lambda$ implies $\mathsf{D}'(x,z)=0$ for all x and z, contradicting to Claim 2 because $\epsilon>0$. In turn, if $\lambda=2^n$ then from Lemma 2 we get $\mathsf{D}(\cdot,z)\equiv 1$ and $t(z)=0$ for all z such that $t(z)\geqslant 0$. This is possible only if $\mathbf{P}_{Y^*|Z=z}(x)=\max_{x'}\mathbf{P}_{Y^*|Z=z}(x')$ for all x which means $H_\infty(Y^*|Z=z)=n$ if $t(z)\geqslant 0$. But then $H_\infty(Y^*|Z=z)=n$ for all z which contradicts $k<n$. □ □

To calculate the success probability we need one more observation. The following claim shows that support of D' is contained in the support of Y^*.

Claim 4. *For every z we have*

$$\mathbb{E}\mathsf{D}'(Y^*|Z=z,z) = \mathbb{E}\mathsf{D}'(U,z)\cdot 2^n \max_{x'} \mathbf{P}_{Y^*|Z=z}(x'). \tag{29}$$

of Claim. By Lemma 2, $\mathsf{D}(x,z) > t(z)$ only if $\mathbf{P}_{Y^*|Z=z}(x) = \max_{x'} \mathbf{P}_{Y^*|Z=z}(x')$ therefore

$$\begin{aligned}\mathbb{E}\mathsf{D}'(Y^*|Z=z,z) &= \sum_x \max(\mathsf{D}(x,z)-t(z),0)\mathbf{P}_{Y^*|Z=z}(x)\\ &= \sum_x \max(\mathsf{D}(x,z)-t(z),0)\max_{x'}\mathbf{P}_{Y^*|Z=z}(x'),\end{aligned}$$

and the claim follows by the definition of D'. □ □

Now we are ready to prove the main result. From Claim 1 and Claim 3 we obtain

$$\begin{aligned}\Pr\left(\textsc{Predictor}(Z,\mathsf{D}',\ell)=X\right) &= 2^{-n-1}\mathbb{E}_{z\leftarrow Z}\left[g(\lambda'/2)\cdot\mathsf{D}'(X|Z=z,z)\right]\\ &= 2^{-n-1}g(\lambda'/2)\cdot\mathbb{E}\mathsf{D}'(X,Z)\end{aligned} \tag{30}$$

Claim 2 applied to $Y=Y^*$ yields now the following estimate

$$\Pr\left(\textsc{Predictor}(Z,\mathsf{D}',\ell)=X\right) \geqslant 2^{-n-1}g(\lambda'/2)\cdot\left(\mathbb{E}\mathsf{D}'(Y^*,Z)+\epsilon\right). \tag{31}$$

Observe that Claim 4, Claim 3, and $\widetilde{H}_\infty(Y^*|Z)=k$ imply

$$\begin{aligned}\mathbb{E}\mathsf{D}'(Y^*,Z) = \mathbb{E}_{z\leftarrow Z}\left[\mathsf{D}'(Y^*|Z=z,z)\right] &= \mathbb{E}_{z\leftarrow Z}\left[\mathbb{E}\mathsf{D}'(U,z)\cdot 2^n\max_{x'}\mathbf{P}_{Y^*|Z=z}(x')\right]\\ &= 2^n\lambda'\cdot\mathbb{E}_{z\leftarrow Z}\left[\max_{x'}\mathbf{P}_{Y^*|Z=z}(x')\right] = 2^{n-k}\lambda'\end{aligned} \tag{32}$$

Plugging this into (31) we get the following bound

$$\begin{aligned}\Pr\left(\textsc{Predictor}(Z,\mathsf{D}',\ell)=X\right) &\geqslant 2^{-n-1}g(\lambda'/2)\cdot\left(2^{n-k}\lambda'+\epsilon\right)\\ &= 2^{-k}\left(1-(1-\lambda'/2)^\ell\right)\left(1+\frac{2^{k-n-1}\epsilon}{\lambda'/2}\right)\end{aligned} \tag{33}$$

To give a lower bound on the success probability it remains to minimize the last expression over $\lambda'\in(0,1)$. This is answered below

Claim 5. *Let $h(s)=(1-(1-s)^\ell)(1+as^{-1})$, where $a>0$ and $\ell\geqslant 1+a^{-1}$. Then $h(s)\geqslant h(1)=1+a$ for all $s\in[0,1]$.*

of Claim. The proof uses standard calculus and is given in the appendix. □ □

Computing $t(z)$ from λ So far, we have shown how to construct the predicting algorithm provided that we are given the numbers $t(z)$. Now we will prove that one can compute them *approximately* and use *successfully* in place of the original ones. We start with a few useful facts about the auxiliary function g already introduced in Claim 1 in the proof of Lemma 3. Below we summarize its fundamental properties.

Lemma 4. *For $\ell > 1$ the function $g(d) = \frac{1-(1-d)^{\ell}}{d}$ on $[0,1]$ satisfies:*

(a) g is continuous at 0 and decreasing

(b) g is convex

(c) for any $d_2 > d_1$ we have $g(d_2) > g(d_1)\left(1 - \frac{\ell}{2}\cdot|d_2 - d_1|\right)$

of Lemma. The proof uses elementary calculus and is referred to the appendix □ □

The entire solution is based on the next two lemmas. The first lemma is based on the intuition that replacing D by a distinguisher which approximates it close enough should not affect the success probability of $\textsc{Predictor}(Z, \mathsf{D}, \ell)$ very much. For technical reasons we present this statement assuming *one-sided $\mathcal{L}^1$-approximation.* The second lemma describes an efficient algorithm which obtains λ as a *hint* on its input and computes approximations for $t(z)$ from below, for every z.

Lemma 5. *Let $\mathsf{D}_1, \mathsf{D}_2 : \{0,1\}^n \times \{0,1\}^m \to [0,1]$ be any two functions satisfying*

(a) $\mathsf{D}_2(x,z) \geqslant \mathsf{D}_1(x,z)$ for all x, z

(b) $\mathbb{E}\mathsf{D}_2(U,z) - \mathbb{E}\mathsf{D}_1(U,z) \leqslant \delta$ for all z

Then we have

$$\Pr\left(\textsc{Predictor}(Z, \mathsf{D}_2, \ell) = X\right) \geqslant (1 - \ell\delta/2)\Pr\left(\textsc{Predictor}(Z, \mathsf{D}_1, \ell) = X\right) \tag{34}$$

of Lemma. We have

$$\begin{aligned}\Pr\left(\textsc{Predictor}(z, \mathsf{D}_2, \ell) = X | Z = z\right) &= g(\mathbb{E}\mathsf{D}_2(U,z))\mathbb{E}\mathsf{D}_2(X|Z=z,z) \\ &\geqslant g(\mathbb{E}\mathsf{D}_2(U,z))\mathbb{E}\mathsf{D}_1(X|Z=z,z),\end{aligned} \tag{35}$$

where the inequality follows from $\mathsf{D}_2 \geqslant \mathsf{D}_1 \geqslant 0$. The assumptions (a) and (b) imply $|\mathbb{E}\mathsf{D}_1(U,z) - \mathbb{E}\mathsf{D}_2(U,z)| \leqslant \delta$ for every z. From property (c) in Lemma 4 it follows that

$$g(\mathbb{E}\mathsf{D}_2(U,z)) \geqslant g(\mathbb{E}\mathsf{D}_1(U,z))(1 - \ell\delta/2)$$

for every z. Combining the last two estimates we get

$$\begin{aligned}\Pr\left(\textsc{Predictor}(z, \mathsf{D}_2, \ell) = X | Z = z\right) &\geqslant (1 - \ell\delta/2)\cdot g(\mathbb{E}\mathsf{D}_1(U,z))\mathbb{E}\mathsf{D}_1(X|Z=z,z) \\ &= (1 - \ell\delta/2)\cdot \Pr\left(\textsc{Predictor}(z, \mathsf{D}_1) = X|Z=z\right)\end{aligned} \tag{36}$$

Taking the average over $z \leftarrow Z$ completes the proof. □ □

Lemma 6. *Let* $\mathsf{D} : \{0,1\}^n \to [0,1]$ *be any function computable in time* s*, let* $\lambda \in (0,1)$ *and* $t \in [0,1]$ *be a number such that* $\mathbb{E}\max(\mathsf{D}(U) - t, 0) = \lambda$*. There exists a probabilistic algorithm* $\textsc{FindThreshold}(\mathsf{D}, \lambda, \delta, N)$ *that runs in time* $\mathcal{O}(\log(1/\delta)N \cdot \mathrm{time}(\mathsf{D}))$ *and with probability at least* $1 - 2\log(12/\delta)\mathrm{e}^{-N\delta^2/3}$ *outputs a number* t' *such that* $\mathbb{E}\max(\mathsf{D}(U) - t', 0) \in [\lambda, \lambda + \delta]$*. In particular,* $t' \leqslant t$*.*

of Lemma. The idea is pretty simple: given t' we approximate values $\mathbb{E}\max(\mathsf{D}(U) - t', 0)$ by sampling and by comparing the result with λ, we can find the right value of t' using binary search. This corresponds to finding a blue line on Fig. 4 such that the green area above is sufficiently close to λ.

```
Function FindThreshold($\mathsf{D}, \lambda, \delta, N$)
  Input  : $\mathsf{D} : \{0,1\}^n \to [0,1]$, $\lambda \in (0,1)$, parameters $\delta, N$
  Output: $t'$ such that $\mathbb{E}\max(\mathsf{D}(U) - t', 0) \in [\lambda, \lambda + \delta]$
  $t^- \leftarrow -1, t^+ \leftarrow 1$
  repeat
      $t' \leftarrow (t^- + t^+)/2$
      $x_1, \ldots, x_N \leftarrow U$                              /* fresh values every time */
      $\lambda' \leftarrow N^{-1}\sum_{j=1}^{N} \max(\mathsf{D}(x_j) - t', 0)$     /* $\lambda' \approx \mathbb{E}\max(\mathsf{D}(U) - t_i, 0)$ */
      if $\lambda' > \lambda + \frac{2\delta}{3}$ then
          $t^- \leftarrow t'$
      else if $\lambda' < \lambda + \frac{\delta}{3}$ then
          $t^+ \leftarrow t'$
      else
          return $t'$
      end
  until $t^+ - t^- \leqslant \frac{\delta}{12}$
  if $t' < -1 + \frac{\delta}{12}$ then
      $t' \leftarrow -1$
  return $t'$
```

The function $h(t') = \mathbb{E}\max(\mathsf{D}(U) - t', 0)$ is clearly non-increasing with respect to t' and changes from $1 + \mathbb{E}\mathsf{D}(U)$ at $t' = -1$ to 0 for $t = 1$. Moreover, it is strictly decreasing in a small neighborhood of $t' = t$ and for all $t' < t$. Indeed, since $\lambda > 0$ there is at least one x such that $\mathsf{D}(x) > t$. Taking $t' < t'' \leqslant \min_{x:\mathsf{D}(x) > t} \mathsf{D}(x)$ we see that $h(t') - h(t'') \geqslant 2^{-n}(t'' - t') > 0$. Hence, $t' > t$ implies $\mathbb{E}\max(\mathsf{D}(U) - t', 0) < \mathbb{E}\max(\mathsf{D}(U) - t, 0) = \lambda$. This proves the second part of the statement. Denote by $\lambda'_i, t'_i, t^-_i, t^+_i$ the values assigned in round i to λ', t', t^-, t^+ respectively. Observe that by the Chernoff Bound[17] and the union bound over at most $\log(12/\delta)$ rounds of the execution, with probability $p = 1 - 2\log(12/\delta)\exp(-N\delta^2/3)$ we have $|\lambda'_i - h(t_i)| < \frac{\delta}{12}$ for every round i. Note that with the same probability the algorithm satisfies the invariant property: if there is $t_0 \in [t^-_i, t^+_i]$ such that $h(t_0) \in \left[\lambda + \frac{5\delta}{12}, \lambda + \frac{7\delta}{12}\right]$ and the algorithm jumps to round $i+1$ then $t_0 \in \left[t^-_{i+1}, t^+_{i+1}\right]$. Suppose that $h(t_0) \in \left[\lambda + \frac{5\delta}{12}, \lambda + \frac{7\delta}{12}\right]$ for some $t_0 \in [-1, 1]$. Now we have two possibilities: either

[17] We use the following version: let $X_1, \ldots, X_N$ be $[0,1]$-valued independent random variables, let $X = \sum_{i=1}^{N} X_i$ and $\mu = \mathbb{E}X$. Then $\Pr(|X - \mu| > \delta\mu) < 2\exp(-\mu\delta^2/3)$

we terminate with t_i such that $\lambda_i \in \left[\lambda + \frac{\delta}{3}, \lambda + \frac{2\delta}{3}\right]$ which means $h(t_i) \in \left[\lambda + \frac{3\delta}{12}, \lambda + \frac{7\delta}{12}\right]$ and we are done, or we will eventually find such t_0 up to an error $\frac{\delta}{12}$. Since $|h(t_2) - h(t_1)| \leqslant |t_2 - t_1|$ for any t_1, t_2, the returned number t' satisfies $h(t_0) - \frac{\delta}{12} \leqslant h(t') \leqslant h(t_0) + \frac{\delta}{12}$, in particular it satisfies the desired inequality. It remains to consider the case when either $h(t) < \lambda + \frac{5\delta}{12}$ for all t or $h(t) > \lambda + \frac{7\delta}{12}$. Since $h(1) = 0$ the second is clearly impossible. In the first case we have $h(t) \leqslant h(-1) < \lambda + \frac{5\delta}{12}$, which means that in every round i we have $t_i^- = -1$ and either we terminate with t_i such that $\lambda_i' \in \left[\lambda + \frac{\delta}{3}, \lambda + \frac{2\delta}{3}\right]$ which means $h(t_i) \in \left[\lambda + \frac{3\delta}{12}, \lambda + \frac{7\delta}{12}\right]$ and we are done, or in every round i we do the assignment $t_{i+1}^+ = t_i$ which yields $t_i = -1 + 2^{-i+1}$ and the main loop halts with $t_i < -1 + \frac{\delta}{12}$. The algorithm outputs then -1 which satisfies the desired inequality, because of the assumption $h(-1) < \lambda + \frac{5\delta}{12}$ and the trivial inequality $h(-1) \geqslant 1 \geqslant \lambda$. □ □

Let D' be as in Lemma 3. Let $t'(z) = \textsc{FindThreshold}(\mathsf{D}, \lambda, \delta, N)$, define $D''(x,z) = \max(\mathsf{D}(U,z) - t'(z), 0)$. Denote by $\Pr[bad]$ the probability that $\mathbb{E}D''(U,z) \notin [\lambda, \lambda + \delta]$ (i.e. probability of failure of the algorithm FindThreshold). If the event bad doesn't occur then $\mathsf{D}'' \geqslant \mathsf{D}'$ and $\mathbb{E}\mathsf{D}''(U,z) \leqslant \mathbb{E}\mathsf{D}'(U,z) + \delta$. Applying the last two claims we obtain

$$\Pr\left[\textsc{Predictor}(z, \mathsf{D}'', \ell)\right] \geqslant 2^{-k}\left(1 + 2^{k-n}\epsilon\right) \cdot \left(1 - \frac{\ell\delta}{2}\right)\Pr[\neg bad] \tag{37}$$

By the elementary inequality $(1+s)(1-s/4)^2 \geqslant 1$ valid for $s \in [0,1]$, for this probability to be bigger than 2^{-k} it is enough to require

$$\ell\delta/2 \leqslant 2^{k-n}\epsilon/4 \tag{38}$$

$$2\log(12/\delta)\exp(-N\delta^2)/3) \leqslant 2^{k-n}\epsilon/4 \tag{39}$$

The solution for the first inequality is $\delta = \mathcal{O}(2^{2(k-n)}\epsilon^2)$ which implies $\delta \ll \epsilon$. The second one gives us $N = \Omega\left((1/\delta)^2(\log\log(1/\delta) + n - k + \log(1/\epsilon)\right)$ which can be simplified to $N = \Omega\left((1/\delta)^2(\log(1/\delta)\right)$. The total running time is (up to a constant factor) the time needed for invoking $\mathcal{O}\left(\ell \cdot N\log(1/\delta)\right) = \mathcal{O}\left((2^{\Delta}/\epsilon)^5 \log^2\left(2^{\Delta}/\epsilon\right)\right)$ times of the distinguisher D .

D Proof of Lemma 2

Proof. Consider the following linear optimization program

$$\begin{array}{ll}
\underset{P_{x,z}, a_z}{\text{maximize}} & \displaystyle\sum_{x,z} \mathsf{D}(x,z)P(x,z) \\
\text{subject to} & -P_{x,z} \leqslant 0,\ (x,z) \in \{0,1\}^n \times \{0,1\}^m \\
& \displaystyle\sum_x P_{x,z} - \mathbf{P}_Z(z) = 0,\ z \in \{0,1\}^m \\
& P_{x,z} - a_z \leqslant 0,\ z \in \{0,1\}^m \\
& \displaystyle\sum_z a_z - 2^{-k} \leqslant 0
\end{array} \tag{40}$$

This problem is equivalent to (20) if we define $\mathbf{P}_{Y,Z}(x,z) = P(x,z)$ and replace the condition $\sum_z \max_x \mathbf{P}_{Y,Z}(x,z) \leqslant 2^{-k}$, which is equivalent to $\widetilde{H}_\infty(Y|Z) \geqslant k$, by the existence of numbers $a_z \geqslant \max_x \mathbf{P}_{Y,Z}(x,z)$ such that $\sum_z a_z \leqslant 2^{-k}$. The solutions of (40) can be characterized as follows:

Claim 6. *The numbers* $(P_{x,z})_{x,z}, (a_z)_z$ *are optimal for* (40) *if and only if there exist numbers* $\lambda^1(x,z) \geqslant 0$, $\lambda^2(z) \in \mathbb{R}$, $\lambda^3(x,z) \geqslant 0$, $\lambda^4 \geqslant 0$ *such that*

(a) $\mathsf{D}(x,z) = -\lambda^1(x,z) + \lambda^2(z) + \lambda^3(x,z)$ *and* $0 = -\sum_x \lambda^3(x,z) + \lambda^4$

(b) We have $\lambda^1(x,z) = 0$ *if* $P_{x,z} > 0$, $\lambda^3(x,z) = 0$ *if* $P_{x,z} < a_z$, $\lambda^4 = 0$ *if* $\sum_z a_z < 2^{-k}$.

of Claim. This is a straightforward application of KKT conditions. □ □

It remains to apply and simplify the last characterization. Let $(P^*_{x,z})_{x,z}, (a^*_z)_z$ be optimal for (40), where $P^*(x,z) = \mathbf{P}_{Y^*,Z}(x,z)$, and $\lambda^1(x,z), \lambda^2(z), \lambda^3(x,z), \lambda^4(x)$ be corresponding multipliers given by the last claim. Define $t(z) = \lambda^2(z)$ and $\lambda = \lambda^4$. Observe that for every z we have $a^*_z \geqslant \max_x \mathbf{P}(x,z) \geqslant 2^{-n}\mathbf{P}_Z(z) > 0$ and thus for every (x,z) we have

$$\lambda^1(x,z) \cdot \lambda^3(x,z) = 0 \tag{41}$$

If $P^*(x,z) = 0$ then $P^*(x,z) < a^*(z)$ and $\lambda^3(x,z) = 0$, hence $\mathsf{D}(x,z) \leqslant t(z)$ which proves (c). If $P^*(x,z) = \max_{x'} P^*(x,z)$ then $P^*(x,z) < 0$ and $\lambda^1(x,z) = 0$ which proves (d). Finally observe that (41) implies

$$\max(\mathsf{D}(x,z) - t(z), 0) = \max(-\lambda^1(x,z) + \lambda^3(x,z), 0) = \lambda^3(x,z)$$

Hence, the assumption $\sum_x \lambda^3(x,z) = \lambda^4 = \lambda$ proves (a).

Suppose now that the characterization given in the Lemma is satisfied. Define $P^*(x,z) = \mathbf{P}_{Y,Z}(x,z)$ and $a_z = \max_z \mathbf{P}_{Y^*,Z}(x,z)$, let $\lambda^3(x,z) = \max(\mathsf{D}(x,z) - t(z), 0)$, $\lambda^1(x,z) = \max(t(z) - \mathsf{D}(x,z), 0)$ and $\lambda^4 = \lambda$. We will show that these numbers satisfy the conditions described in the last claim. By definition we have $-\lambda^1(x,z) + \lambda^2(z) + \lambda^3(x,z) = \mathsf{D}(x,z)$, by the assumptions we get $\sum_x \lambda^3(x,z) = \lambda = \lambda^4$. This proves part (a). Now we verify the conditions in (b). Note that $\mathsf{D}(x,z) < t(z)$ is possible only if $\mathbf{P}_{Y^*|Z=z}(x) = 0$ and $\mathsf{D}(x,z) > t(z)$ is possible only if $\mathbf{P}_{Y^*|Z=z}(x) = \max_{x'} \mathbf{P}_{Y^*|Z=z}(x')$. Therefore, if $\mathbf{P}_{Y,Z}(x,z) > 0$ then we must have $\mathsf{D}(x,z) \geqslant t(z)$ which means that $\lambda^1(x,z) = 0$. Similarly if $\mathbf{P}_{Y,Z}(x,z) < \max_z \mathbf{P}_{Y^*,Z}(x,z)$ then $\mathsf{D}(x,z) \leqslant t(z)$ and $\lambda^3(x,z) = 0$. Finally, since we assume $\widetilde{H}_\infty(Y^*|Z) = k$ we have $\sum_z a_z = 2^{-k}$ and thus there is no additional restrictions on λ^4. □ □

E Proof of Corollary1

of Corollary. Let $y_{\max}(z) = \max_{x'} \mathbf{P}_{Y|Z=z}(x')$. Consider the function

$$f^\delta_z(x) = \begin{cases} y_{\max}(z) + \delta, & \mathsf{D}'(x,z) > t(z) \\ \frac{1 - \#\{x\colon\ \mathsf{D}'(x.z) > t(z)\} \cdot (y_{\max} + \delta)}{\#\{x\colon\ \mathsf{D}'(x,z) = t(z)\}}, & \mathsf{D}'(x,z) = t(z) \\ 0, & \mathsf{D}'(x,z) < t(z) \end{cases} \tag{42}$$

This function defines a distribution that satisfies

$$f^\delta_z(x) \leqslant \max_{x'} f^\delta_z(x') \quad \forall x : \mathsf{D}'(x,z) \leqslant t(z) \tag{43}$$

if and only if δ satisfies

$$\frac{1}{\#\left\{x:\ \mathsf{D}'(x.z) \geqslant t(z)\right\}} \leqslant y_{\max}(z)+\delta \leqslant \frac{1}{\#\left\{x:\ \mathsf{D}'(x.z) > t(z)\right\}} \tag{44}$$

In particular these conditions are satisfied for $\delta=0$. Suppose now that there are z_i and x_i for $i=1,2$ such that $0<\mathbf{P}_{Y^*|Z=z_i}(x_i)<\max_{x'}\mathbf{P}_{Y^*|Z=z}(x')$. Define δ by

$$\delta=\min\left(y_{\max}(z_1)-\frac{1}{\#\left\{x:\mathsf{D}'(x,z_1)\geqslant t(z_1)\right\}},\ \frac{1}{\#\left\{x:\mathsf{D}'(x,z_2)>t(z_2)\right\}}-y_{\max}(z_2)\right)$$

By Lemma 2 we immediately obtain that $\delta \geqslant 0$. It follows easily from the definition of δ that the number $-\delta$ satisfies (44) with $z=z_1$ and that δ satisfies (44) for $z=z_2$. We can see now that if we replace the distribution $Y^*|Z=z_1$ by $f_{z_1}^{-\delta}$ and the distribution $Y^*|Z=z_2$ by $f_{z_2}^{\delta}$ then we obtain the distribution $Y'|Z$ satisfying conditions in Lemma 2 and $\widetilde{H}_{\infty}(Y'|Z)=k$. Finally, observe that $\delta=\frac{1}{\#\{x:\mathsf{D}'(x,z_2)>t(z_2)\}}-y_{\max}(z_2)$ means that the distribution $Y'|Z=z_2$ is uniform on $\{x:\mathsf{D}'(x.z_2)>t(z_2)\}$. In turn, if $\delta=y_{\max}(z_1)-\frac{1}{\#\{x:\mathsf{D}'(x,z_1)\geqslant t(z_1)\}}$ then the distribution $Y'|Z=z_1$ is uniform on $\{x:\mathsf{D}'(x,z_1)\geqslant t(z_1)\}$. □ □

F Proof of Claim 5, Lemma 3

Proof. We check that $\lim_{s\to 0}h(s)=a\ell$ and thus the function h is continuous on the interval $[0,1]$. This means that h attains its minimum at some point $s=s_0$. There is nothing to prove if $s_0\in\{0,1\}$. Suppose that $s_0\in(0,1)$. Then we must have $\frac{\partial h}{\partial s}\big|_{s=s_0}=0$. The first derivative of the function h is given by the following formula

$$\begin{aligned}\frac{\partial h}{\partial s} &= \frac{s\ell(a+s)(1-s)^{\ell-1}+a\left((1-s)^{\ell}-1\right)}{s^2}\\ &= \frac{-a+(1-s)^{\ell-1}\left(a(1-s)+(a+s)\ell s\right)}{s^2}\end{aligned} \tag{45}$$

Therefore for $s=s_0$ we obtain $(1-s_0)^{\ell-1}=\frac{a}{a(1-s_0)+(a+s_0)\ell s_0}$ and hence

$$\begin{aligned}h(s_0) &= \left(1-(1-s_0)\cdot(1-s_0)^{\ell-1}\right)\left(1+as_0^{-1}\right)\\ &= \frac{(a+s_0)^2\ell}{a(1-s_0)+(a+s_0)\ell s_0}\end{aligned} \tag{46}$$

Note that the last expression is increasing with respect to ℓ and that from the assumption we have $\ell>\frac{1+a}{a+s_0}$. Using this we obtain

$$h(s_0)\geqslant\frac{(a+s_0)(1+a)}{a(1-s_0)+(1+a)s_0}=1+a \tag{47}$$

which completes the proof. □ □

The lemma follows now immediately by combining (33) and the last claim. □ □

G Proof of Lemma 4

of Lemma. It is easy to see that $\lim_{d\to 0^+} g(d) = \ell$. We have

$$\frac{\partial g(d)}{\partial d} = \frac{(1-d)^{\ell-1}(d(\ell-1)+1)-1}{d^2} \tag{48}$$

Using the inequality $1-d \leqslant \mathrm{e}^{-d}$ we obtain

$$\frac{\partial g(d)}{\partial d} \leqslant \frac{\mathrm{e}^{-d(\ell-1)}\,(d(\ell-1)+1)-1}{d^2} \leqslant 0$$

Where the second inequality follows from the inequality $\mathrm{e}^s \geqslant 1+s$ applied for $s = d(\ell-1)$. This proves (a). The second derivative is given by

$$\frac{\partial^2 g(d)}{\partial d^2} = -\frac{(1-d)^{\ell-2}\left(2+2d(\ell-2)+d^2((\ell-2)^2+\ell-2)\right)-2}{d^3} \tag{49}$$

Using $1-d \leqslant \mathrm{e}^{-d}$ and applying the inequality $\mathrm{e}^s \geqslant 1+s+\frac{1}{2}s^2$, which holds for $s \geqslant 0$, for $s = d(\ell-1)$ we obtain

$$\begin{aligned}\frac{\partial^2 g(d)}{\partial d^2} &= -\frac{(1-d)^{\ell-2}\left(2+2d(\ell-2)+d^2((\ell-2)^2+\ell-2)\right)-2}{d^3} \\ &\geqslant -\frac{(1-d)^{\ell-1}\left(2+2d(\ell-1)+d^2(\ell-1)^2\right)-2}{d^3} \\ &\geqslant -\frac{\mathrm{e}^{-d(\ell-1)}\left(2+2d(\ell-1)+d^2(\ell-1)^2\right)-2}{d^3} \\ &\geqslant -\frac{2-2}{d^3} = 0,\end{aligned} \tag{50}$$

which proves (b). Finally, note that by convexity we have

$$g(d_2)-g(d_1) \geqslant (d_2-d_1)\cdot \left.\frac{\partial g(d)}{\partial d}\right|_{d=d_1}. \tag{51}$$

Since $g(d) > 0$ and $\frac{\partial \ln g(d)}{\partial d} = \frac{\partial g(d)}{\partial d}/g(d)$ we can rewrite this as

$$\frac{g(d_2)-g(d_1)}{g(d_1)} \geqslant (d_2-d_1)\cdot \left.\frac{\partial \ln g(d)}{\partial d}\right|_{d=d_1}. \tag{52}$$

Note that the function $d \to \ln g(d)$ is convex, as the composition of the convex function $g(\cdot)$ and the convex increasing function $\ln(\cdot)$. Therefore,

$$\frac{\partial \ln g(d)}{\partial d} \geqslant \left.\frac{\partial \ln g(d)}{\partial d}\right|_{d=0} = -\frac{\ell-1}{2} \tag{53}$$

Combining the last two inequalities yields

$$\frac{g(d_2)-g(d_1)}{g(d_1)} > -\frac{\ell}{2}\cdot(d_2-d_1), \quad d_2-d_1 > 0. \tag{54}$$

which completes the proof of (c). □ □