Simulating Auxiliary Inputs, Revisited *

Maciej Skorski **

maciej.skorski@mimuw.edu.pl
University of Warsaw

Abstract. For any pair (X, Z) of correlated random variables we can think of Z as a randomized function of X. If the domain of Z is small, one can make this function computationally efficient by allowing it to be only approximately correct. In folklore this problem is known as *simulating auxiliary inputs*. This idea of simulating auxiliary information turns out to be a very usefull tool, finding applications in complexity theory, cryptography, pseudorandomness and zero-knowledge. In this paper we revisit this problem, achieving the following results:

(a) We present a novel boosting algorithm for constructing the simulator. This boosting proof is of independent interest, as it shows how to handle "negative mass" issues when constructing probability measures by shifting distinguishers in descent algorithms. Our technique essentially fixes the flaw in the TCC'14 paper "How to Fake Auxiliary Inputs".

(b) The complexity of our simulator is better than in previous works, including results derived from the uniform min-max theorem due to Vadhan and Zheng. To achieve (s, ϵ)-indistinguishability we need the complexity $O\left(s \cdot 2^{5\ell}\epsilon^{-2}\right)$ in time/circuit size, which improve previous bounds by a factor of ϵ^{-2}. In particular, with we get meaningful provable security for the EUROCRYPT'09 leakage-resilient stream cipher instantiated with a standard 256-bit block cipher, like AES256.

Our boosting technique utilizes a two-step approach. In the first step we shift the current result (as in gradient or sub-gradient descent algorithms) and in the separate step we fix the biggest non-negative mass constraint violation (if applicable).

Keywords: simulating auxiliary inputs, boosting, leakage-resilient cryptography, stream ciphers, computational indistinguishability

1 Introduction

1.1 Simulating Correlated Information.

Informal Problem Statement Let $(X, Z) \in \mathcal{X} \times \mathcal{Z}$ be a pair of correlated random variables. We can think of Z as a *randomized* function of X. More

* The full (and updated) version of this paper is available at the Cryptology ePrint archive and the arXiv archive ({http://arxiv.org/abs/1503.00484}).

** Supported by the National Science Center, Poland (2015/17/N/ST6/03564).

precisely, consider the randomized function $h : \mathcal{X} \to \mathcal{Z}$, which for every x outputs z with probability $\Pr[Z = z|X = x]$. By definition it satisfies

$$(X, h(X)) \stackrel{d}{=} (X, Z) \tag{1}$$

however the function h is *inefficient* as we need to hardcode the conditional probability table of $Z|X$. It is natural to ask, if this limitation can be overcome

Q1: Can we represent Z as an *efficient* function of X?

Not surprisingly, it turns out that a positive answer may be given only in computational settings. Note that replacing the equality in Equation (1) by closeness in the total variation distance (allowing the function h to make some mistakes with small probability) is not enough [1]. This discussion leads to the following reformulated question

Q1': Can we *efficiently simulate* Z as a function of X?

Why it matters? Aside from being very foundational, this question is relevant to many areas of computer science. We will not discuss these applications in detail, as they are well explained in [JP14]. Below we only mention where such a generic simulator can be applied, to show that this problem is indeed well-motivated.

(a) Complexity Theory. From the simulator one can derive Dense Model Theorem [RTTV08], Impagliazzo's hardcore lemma [Imp95] and a version of Szemeredis Regularity Lemma [FK99].
(b) Cryptography. The simulator can be applied for settings where Z models short leakage from a secret state X. It provides tools for improving and simplifying proofs in leakage-resilient cryptography, in particular for leakage-resilient stream ciphers [JP14].
(c) Pseudorandomness. Using the simulator one can conclude results called chain rules [GW11], which quantify pseudorandomness in conditioned distributions. They can be also applied to leakage-resilient cryptography.
(d) Zero-knowledge. The simulator can be applied to represent the text exchanged in verifier-prover interactions Z from the common input X [CLP15].

Thus, the simulator may be used as a tool to unify, simplify and improve many results. Having briefly explained the motivation we now turn to answer the posed question, leaving a more detailed discussion of some applications to Section 1.6.

[1] Indeed, consider the simplest case $\mathcal{Z} = \{0, 1\}$, define X to be uniform over $\mathcal{X} = \{0, 1\}^n$, and take $Z = f(X)$ where f is a function which is 0.5-hard to predict by circuits exponential in n, Then $(X, h(X))$ and (X, Z) are at least $\frac{1}{4}$-away in total variation

1.2 Problem Statement

The problem of simulating auxiliary inputs in the computational setting can be defined precisely as follows

> Given a random variables $X \in \{0,1\}^n$ and correlated $Z \in \{0,1\}^\ell$, what is the minimal complexity s_h of a (randomized) function h such that the distributions of $h(X)$ and Z are (ϵ, s)-indistinguishable given X, that is
>
> $$|\,\mathbb{E}\,\mathrm{D}(X, h(X)) - \mathbb{E}\,\mathrm{D}(X, Z)| < \epsilon$$
>
> holds for all (deterministic) circuits D of size s?

The indistinguishability above is understood with respect to deterministic circuits. However it doesn't really matter for distinguishing two distributions, where randomized and deterministic distinguishers are equally powerful[2].

It turns out that it is relatively easy[3] to construct a simulator h with a polynomial blowup in complexity, that is when

$$s_h = \mathrm{poly}\left(s, \epsilon^{-1}, 2^\ell\right).$$

However, more challenging is to minimize the dependency on ϵ^{-1}. This problem is especially important for cryptography, where security definitions require the advantage ϵ to be possibly small. Indeed, for meaningful security $\epsilon = 2^{-80}$ or at least $\epsilon = 2^{-40}$ it makes a difference whether we lose ϵ^{-2} or ϵ^{-4}. We will see later how much inefficient bounds here may affect provable security of stream ciphers.

1.3 Related Works

Original work of Jetchev and Pietrzak (TCC'14) The authors showed that Z can be "approximately" computed from X by an "efficient" function h.

Theorem 1 ([JP14], corrected). *For every distribution (X, Z) on $\{0,1\}^n \times \{0,1\}^\ell$ and every ϵ, s, there exists a "simulator" $h : \{0,1\}^n \to \{0,1\}^\ell$ such that*

(a) $(X, h(X))$ and (X, Z) are (ϵ, s)-indistinguishable
(b) h is of complexity $s_\mathsf{h} = O\left(s \cdot 2^{4\ell}\epsilon^{-4}\right)$

The proof uses the standard min-max theorem. In the statement above we correct two flaws. One is a missing factor of 2^ℓ. The second (and more serious) one is the (corrected) factor ϵ^{-4}, claimed incorrectly to be ϵ^{-2}. The flaws are discussed in Appendix A.

[2] If two distributions can be distinguished by a randomized circuit, we can fix a specific choice of coins to achieve at least the same advantage

[3] We briefly sketch the idea of the proof: note first that it is easy to construct a simulator for every single distinguisher. Having realized that, we can use the min-max theorem to switch the quantifiers and get one simulator for all distinguishers.

Vadhan and Zheng (CRYPTO'13) The authors derived a version of Theorem 1 but with incomparable bounds

Theorem 2 ([VZ13]). *For every distribution X, Z on $\{0,1\}^n \times \{0,1\}^\ell$ and every ϵ, s, there exists a "simulator" $h : \{0,1\}^n \to \{0,1\}^\ell$ such that*

(a) $(X, h(X))$ and (X, Z) are (s, ϵ)-indistinguishable
(b) h is of complexity $s_{\mathsf{h}} = O\left(s \cdot 2^\ell \epsilon^{-2} + 2^\ell \epsilon^{-4}\right)$

The proof follows from a general regularity theorem which is based on their uniform min-max theorem. The additive loss of $O\left(2^\ell \epsilon^{-4}\right)$ appears as a consequence of a sophisticated weight-updating procedure. This error is quite large and may dominate the main term for many settings (whenever $s \ll \epsilon^{-2}$).

As we show later, Theorem 2 and Theorem 1 give in fact comparable security bounds when applied to leakage-resilient stream ciphers (see Section 1.6)

1.4 Our Results

We reduce the dependency of the simulator complexity s_h on the advantage ϵ to only a factor of ϵ^{-2}, from the factor of ϵ^{-4}.

Theorem 3 (Our Simulator). *For every distribution X, Z on $\{0,1\}^n \times \{0,1\}^\ell$ and every ϵ, s, there exists a "simulator" $h : \{0,1\}^n \to \{0,1\}^\ell$ such that*

(a) $(X, h(X))$ and (X, Z) are (s, ϵ)-indistinguishable
(b) h is of complexity $s_{\mathsf{h}} = O\left(s \cdot 2^{5\ell} \log(1/\epsilon) \epsilon^{-2}\right)$

Below in Table 1 we compare our result to previous works.

Author	Technique	Advantage	Size	Cost of simulating
[JP14] (Theorem 1)	Min-Max	ϵ	s	$s_{\mathsf{h}} = O\left(s \cdot 2^{4\ell} \epsilon^{-4}\right)$
[VZ13] (Theorem 2)	Complicated Boosting			$s_{\mathsf{h}} = O\left(s \cdot 2^\ell / \epsilon^2 + 2^\ell \epsilon^{-4}\right)$
This paper (Theorem 3)	Simple Boosting			$s_{\mathsf{h}} = O\left(s \cdot 2^{5\ell} \epsilon^{-2}\right)$

Table 1. The complexity of simulating ℓ-bit auxiliary information given required indistinguishability strength, depending on the proof technique. For simplicity, terms polylog$(1/\epsilon)$ are omitted.

Our result is slightly worse in terms of dependency ℓ, but outperforms previous results in terms of dependency on ϵ^{-1}. However, the second dependency is more crucial for cryptographic applications. Note that the typical choice is sub-logarithmic leakage, that is $\ell = o\left(\log \epsilon^{-1}\right)$ is asymptotic settings[4] (see for example [CLP15]). Stated in non-asymptotic settings this assumption translates to $\ell < c \log \epsilon^{-1}$ where c is a small constant (for example $c = \frac{1}{12}$ see [Pie09]). In these settings, we outperform previous results.

[4] This is a direct consequence of the fact that we want ℓ to fit poly-preserving reductions

To illustrate this, suppose we want to achieve security $\epsilon = 2^{-60}$ simulating just one bit from a 256-bit input. As it follows from Table 1, previous bounds are useless as they give the complexity bigger than 2^{256} which is the worst complexity of all boolean functions over the chosen domain. In settings like this, only our bound can be applied to conclude meaningful results. For more concrete examples of settings where our bounds are even only meaningful, we refer to Table 2 in Section 1.6.

1.5 Our Techniques

Our approach utilizes a simple boosting technique: as long as the condition (a) in Theorem 3 fails, we can use the distinguisher to improve the simulator. This makes our algorithm constructive with respect to distinguishers obtained from an oracle[5], similarly to other boosting proofs [JP14,VZ13]. In short, if for a "candidate" solution h there exists D such that

$$\mathbb{E}\,\mathrm{D}(X, Z) - \mathbb{E}\,\mathrm{D}(X, h(X)) > \epsilon$$

then we construct a new solution h' using D and h, according to the equation[6]

$$\Pr[h'(x) = z] = \Pr[h(x) = z] + \gamma \cdot \mathsf{Shift}\,(\mathrm{D}(x, z)) + \mathsf{Corr}(x, z)$$

where

(a) The parameter γ is a*fixed step* chosen in advance (its optimal value depends on ϵ and ℓ and is calculated in the proof.)
(b) $\mathsf{Shift}\,(\mathrm{D}(x, z))$ is a *shifted* version of D, so that $\sum_z \mathsf{Shift}\,(\mathrm{D}(x, z)) = 0$. This restriction correspond to the fact that we want to preserve the constraint $\sum_z h(x, z) = 1$. More precisely, $\mathsf{Shift}\,(\mathrm{D}(x, z)) = \mathrm{D}(x, z) - \mathbb{E}_{z' \leftarrow U_\ell}\,\mathrm{D}(x, z)$
(c) $\mathsf{Corr}(x, z)$ is a *correction term* used to fix (some of) possibly negative weights.

The procedure is being repeated in a loop, over and over again. The main technical difficulty is to show that it eventually stops after not so many iterations.

Note that in every such a step the complexity cost of the shifting term is $O\left(2^\ell \cdot \mathrm{size}(\mathrm{D})\right)$[7]. The correction term, in our approach, does a search over z looking for the biggest negative mass, and redistributes it over the remaining points. Intuitively, it works because the total negative mass is getting smaller with every step. See Algorithm 1 for a pseudo-code description of the algorithm and the rest of Section 3 for a proof.

[5] The oracle evaluates the distance of the given candidate solution and the simulated distribution, answering with a distiguisher if the distance is smaller than required.
[6] As we already mentioned, we can assume that D is deterministic without loss of generality. Then all the terms in the equation are well-defined.
[7] By definition, it requires computing the average of $\mathrm{D}(x, \cdot)$ over 2^ℓ elements

1.6 Applications

Better security for the EUROCRYPT'09 stream cipher. The first construction of leakage-resilient stream cipher was proposed by Dziembowski and Pietrzak in [DP08]. On Figure 1 below we present a simplified version of this cipher [Pie09], based on a weak pseudorandom function (wPRF).

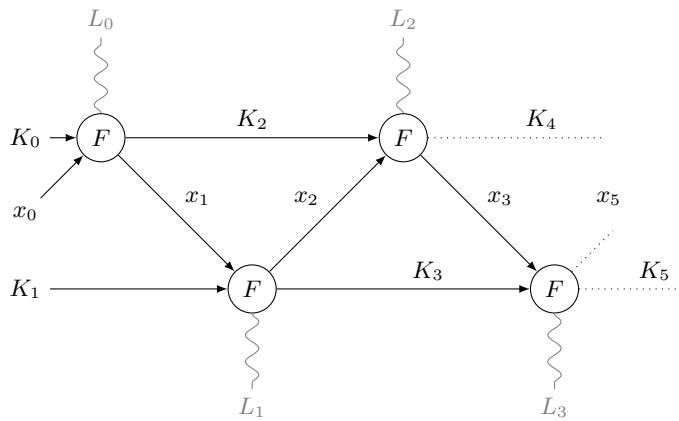

Fig. 1. The EUROCRYPT'09 stream cipher (adaptive leakage). F denotes a weak pseudorandom function. By K_i and x_i we denote, respectively, values of the secret state and keystream bits. Leakages are denotted in gray with L_i.

Jetchev and Pietrzak in [JP14] showed how to use the simulator theorem to simplify the security analysis of the EUROCRYPT'09 cipher. The cipher security depends on the complexity of the simulator as explained in Theorem 1 and Remark 2. We consider the following setting:

- number of rounds $q = 16$,
- F instantiated with AES256 (as in [JP14])
- cipher security we aim for $\epsilon' = 2^{-40}$
- $\lambda = 3$ bits of leakage per round

The concrete bounds for (q, ϵ', s')-security of the cipher (which roughly speaking means that q consecutive outputs is (s', ϵ')-pseudorandom, see Section 2 for a formal definition) are given in Table 2 below. We ommit calculations as they are merely putting parameters from Theorem 1, Theorem 2 and Theorem 3 into Remark 2 and assuming that AES as a weak PRF is (ϵ, s)-secure for any pairs $s/\epsilon \approx 2^k$ (following the similar example in [JP14]).

More generally, we can give the following comparison of security bounds for different wPRF-based stream ciphers, in terms of time-sccess ratio. The bounds in Table 3 follow from the simple lemma in Section 4, which shows how the time-success ratio changes under explicit reduction formulas.

Analysis/Authors	wPRF security	Leakage	Advantage ϵ'	Size s'
[JP14] (Theorem 1)	256	$\lambda = 3$	2^{-40}	0
[VZ13] (Theorem 2)				0
this paper (Theorem 3)				2^{66}

Table 2. The security of the EUROCRYPT'09 stream cipher, instantiated with AES256 as a weak PRF of roughly $k = 256$ bits of security. In this setting only our new bounds provide non-trivial bounds.

Cipher	Analysis	Proof techniques	Security level	Comments
(1)	[Pie09]	Pseudoentropy chain rules	$k' \ll \frac{1}{8}k$	large number of blocks
(1)	[JP14]	Aux. Inputs Simulator (corr.)	$k' \approx \frac{k}{6} - \frac{5}{6}\lambda$	
(1)	[VZ13]	Aux. Inputs Simulator	$k' \approx \frac{k}{6} - \frac{1}{3}\lambda$	
(1)	**This work**	Aux. Inputs Simulator	$k' \approx \frac{k}{4} - \frac{4}{3}\lambda$	
(2)	[FPS12]	Pseudoentropy chain rules	$k' \approx \frac{k}{5} - \frac{3}{5}\lambda$	large public seed
(3)	[YS13]	Square-friendly apps.	$k' \approx \frac{k}{4} - \frac{3}{4}\lambda$	only in minicrypt

Table 3. Different bounds for wPRF-based leakage-resilient stream ciphers. k is the security level of the underlying wPRF. The value k' is the security level for the cipher, understood in terms of time-success ratio. the numbers denote: (1) The EUROCRYPT'09 cipher, (2) The CSS'10/CHESS'12 cipher, (3) The CT-RSA'13 cipher.

1.7 Organization

In Section 2 we discuss basic notions and definitions. The proof of Theorem 3 appears in Section 3.

2 Preliminaries

2.1 Notation

By $\mathbb{E}_{y \leftarrow Y} f(y)$ we denote an expectation of f under y sampled according to the distribution Y.

2.2 Basic Notions

Indistinguishability Let $\mathcal{V}$ be a finite set, and $\mathcal{D}$ be a class of deterministic $[0,1]$-valued functions on $\mathcal{V}$. For any two real functions f_1, f_2 on $\mathcal{V}$, we say that f_1, f_2 are $(\mathcal{D}, \epsilon)$-indistinguishable if

$$\forall \mathrm{D} \in \mathcal{D} : \quad \left| \sum_{x \in \mathcal{V}} \mathrm{D}(x) \cdot f_1(x) - \sum_{x \in \mathcal{V}} \mathrm{D}(x) \cdot f_2(x)) \right| \leqslant \epsilon$$

Note that the domain $\mathcal{V}$ depends on the context. If X_1, X_2 are two probability distributions, we say that they are (s, ϵ)-indistinguishable if their probability

mass functions are indistinguishable, that is when

$$\left|\sum_{x\in V} \mathrm{D}(x)\cdot \Pr[X_1 = x] - \sum_{x\in V} \mathrm{D}(x)\cdot \Pr[X_2 = x]\right| \leqslant \epsilon$$

for all $\mathrm{D} \in \mathcal{D}$. If $\mathcal{D}$ consists of all circuits of size s we say that f_1, f_2 are (s, ϵ)-indistinguishable.

Remark 1. This an extended notion of indistinguishability, borrowed from [TTV09], which captures not only probability measures but also real-valued functions. A good intuition is provided by the following observation [TTV09]: think of functions over $\mathcal{V}$ as $|\mathcal{V}|$-dimensional vectors then $\epsilon \geqslant |\sum_{x\in V} \mathrm{D}(x)\cdot f_1(x) - \sum_{x\in V} \mathrm{D}(x)\cdot f_2(x)| = |\langle f_1 - f_2, \mathrm{D}\rangle|$ means that f_1 and f_2 are *nearly orthogonal* for all test functions in $\mathcal{D}$.

Distinguishers In the definition above we consider deterministic distinguishers, as this is required by our algorithm. However, being randomized doesn't help in distinguishing, as any randomized-distinguisher achieving advantage ϵ when run on two fixed distributions can be converted into a deterministic distinguishers of the same size and advantage (by fixing one choice of coins). Moreover, any real-valued distinguisher can be converted, by a boolean threshold, into a boolean one with at least the same advantage [FR12].

Relative complexity We say that a function h has complexity at most T relative to the set of functions $\mathcal{D}$ if there are functions $\mathrm{D}_1, \ldots, \mathrm{D}_T$ such h can be computed by combining them using at most T of the following operations: (a) multiplication by a constant, (b) application of a boolean threshold function, (c) sum, (d) product.

2.3 Stream ciphers definitions

We start with the definition of weak pseudorandom functions, which are *computationally indistinguishable* from random functions, when queried on random inputs and fed with uniform secret key.

Definition 1 (Weak pseudorandom functions). *A function* $\mathrm{F} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^m$ *is an* (ϵ, s, q)*-secure weak PRF if its outputs on* q *random inputs are indistinguishable from random by any distinguisher of size* s*, that is*

$$|\Pr\left[\mathrm{D}\left((X_i)_{i=1}^q, \mathrm{F}((K, X_i)_{i=1}^q\right) = 1\right] - \Pr\left[\mathrm{D}\left((X_i)_{i=1}^q, (R_i)_{i=1}^q\right) = 1\right]| \leqslant \epsilon$$

where the probability is over the choice of the random $X_i \leftarrow \{0,1\}^n$*, the choice of a random key* $K \leftarrow \{0,1\}^k$ *and* $R_i \leftarrow \{0,1\}^m$ *conditioned on* $R_i = R_j$ *if* $X_i = X_j$ *for some* $j < i$*.*

Stream ciphers generate a keystream in a recursive manner. The security requires the output stream should be indistinguishable from uniform[8].

[8] We note that in a more standard notion the entire stream $X_1, \ldots, X_q$ is indistinguishable from random. This is implied by the notion above by a standard hybrid argument, with a loss of a multiplicative factor of q in the distinguishing advantage.

Definition 2 (Stream ciphers). *A* stream-cipher $\mathsf{SC} : \{0,1\}^k \to \{0,1\}^k \times \{0,1\}^n$ *is a function that, when initialized with a secret state* $S_0 \in \{0,1\}^k$*, produces a sequence of output blocks* $X_1, X_2, ...$ *computed as*

$$(S_i, X_i) := \mathsf{SC}(S_{i-1}).$$

A stream cipher SC *is* (ϵ, s, q)*-secure if for all* $1 \leqslant i \leqslant q$*, the random variable* X_i *is* (s, ϵ)*-pseudorandom given* $X_1, ..., X_{i-1}$ *(the probability is also over the choice of the initial random key* S_0*).*

Now we define leakage resilient stream ciphers, following the "only computation leaks" assumption.

Definition 3 (Leakage-resilient stream ciphers [JP14]). *A leakage-resilient stream-cipher is* $(\epsilon, s, q, \lambda)$*-secure if it is* (ϵ, s, q)*-secure as defined above, but where the distinguisher in the* j*-th round gets* λ *bits of arbitrary deceptively chosen leakage about the secret state accessed during this round. More precisely, before* $(S_j, X_j) := \mathsf{SC}(S_{j-1})$ *is computed, the distinguisher can choose any leakage function* f_j *with range* $\{0,1\}^\lambda$*, and then not only get* X_j*, but also* $\Lambda_j := f_j(\hat{S}_{j-1})$*, where* $\hat{S}_{j-1}$ *denotes the part of the secret state that was modified (i.e., read and/or overwritten) in the computation* $\mathsf{SC}(S_{j-1})$*.*

2.4 Security of leakage-resilient stream ciphers.

Best provable secure constructions of leakage-resilient stream ciphers are based on so called weak PRFs, primitives which look random when queried on random inputs ([Pie09,FPS12,JP14,DP10,YS13]). The most recent (TCC'14) analysis is based on a version of Theorem 1.

Theorem 4 (Proving Security of Stream Ciphers [JP14]). *If* F *is a* $(\epsilon_F, s_F, 2)$*-secure weak PRF then* SC^F *is a* $(\epsilon', s', q, \lambda)$*-secure leakage resilient stream cipher where*

$$\epsilon' = 4q\sqrt{\epsilon_F 2^\lambda}, \quad s' = \Theta(1) \cdot \frac{s_F \epsilon'^4}{2^{4\lambda}}.$$

Remark 2 (The exact complexity loss). An inspection of the proof in [JP14] shows that s_F equals the complexity of the simulator h in Theorem 1, with circuits of size s' as distingusihers and ϵ replaced by ϵ'.

2.5 Time-Success Ratio

The running time (circuit size) s and success probability ϵ of attacks (practical and theoretical) against a particular primitive or protocol may vary. For this reason Luby [LM94] introduced the time-success ratio $\frac{t}{\epsilon}$ as a universal measure of security. This model is widely used to analyze provable security, cf. [BL13] and related works.

Definition 4 (Security by Time-Success Ratio [LM94]). *A primitive P is said to be 2^k-secure if for* every adversary *with time resources (circuit size in the nonuniform model) s, the success probability in breaking P (advantage) is at most $\epsilon < s \cdot 2^{-k}$. We also say that the time-success ratio of P is 2^k, or that is has k bits of security.*

For example, AES with a 256-bit random key is believed to have 256 bits of security as a *weak* PRF[9].

3 Proof of Theorem 3

For technical convenience, we attempt to efficiently approximate the conditional probability function $g(x, z) = \Pr[Z = z|X = x]$ rather than building the sampler directly. Once we end with building an efficient approximation $h(x, z)$, we transform it into a sampler h_{sim} which outputs z with probability $h(x, z)$ (this transformation yields only a loss of $2^\ell \log(1/\epsilon)$). We are going to prove the following fact

> For every function g on $\mathcal{X} \times \mathcal{Z}$ which is a $\mathcal{X}$-conditional probability mass function over Z (that is $g(x, z) \geqslant 0$ for all x, z and $\sum_z g(x, z) = 1$ for every x), and for every class $\mathcal{D}$ closed under complements[10] there exists h such that
> (a) h is a $\mathcal{X}$-conditional probability mass function over Z
> (b) h is of complexity $s_h = O(2^{4\ell}\epsilon^{-2})$ with respect to $\mathcal{D}$
> (c) (X, Z) and $(X, h_{\mathsf{sim}}(X))$ are indistinguishable, which in terms of g and h means
>
> $$\left|\sum_z \mathop{\mathbb{E}}_{x\sim X} [\mathrm{D}(x, z) \cdot (g(x, z) - h(x, z))]\right| \leqslant \epsilon \qquad (2)$$

The sketch of the construction is shown in Algorithm 1. Here we would like to point out two things. First, we stress that we do not produce a strictly positive function; what our algorithm guarantees, is that the total negative mass is*small*. We will see later that this is enough. Second, our algorithm performs essentially same operations for every x, which is why its complexity depends only on $\mathcal{Z}$.

We denote for shortness $\overline{\mathrm{D}}(x, z) = \mathrm{D}(x, z) - \mathbb{E}_{z'\leftarrow U_{\mathcal{Z}}} \mathrm{D}(x, z')$ for any D (the "shift" transformation)

[9] We consider the security of AES256 as a weak PRF, and not a standard PRF, because of non-uniform attacks which show that no PRF with a k-bit key can have $s/\epsilon \approx 2^k$ security [DTT09], at least unless we additionally require $\epsilon \gg 2^{-k/2}$.

[10] This is a standard assumption in indistinguishability proofs. We can always extend the class by adding $-\mathrm{D}$ for every $\mathrm{D} \in \mathcal{D}$, which increases the complexity only by 1.

Algorithm 1: Construct the Auxiliary Inputs Simulator

input : Function $g : \{0,1\}^n \times \{0,1\}^\ell \to [0,1]$, accuracy paramter $\epsilon > 0$, class $\mathcal{D}$, step γ

output: Function h which is ϵ-indistinguishable from g under $\mathcal{D}$, add up to 1 for every x, and with total negative mass smaller $\gamma|\mathcal{Z}|^3$

1 $t \leftarrow 0$

2 $h^0(x,z) \leftarrow \frac{1}{|\mathcal{Z}|}$ for every x and z

3 **while** *exists* $\mathrm{D} \in \mathcal{D}$ *such that* $\mathbb{E}_{x\sim X}\left[\sum_z \overline{\mathrm{D}}(x,z)\cdot\left(g(x,z') - h^t(x,z')\right)\right] \geqslant \epsilon$ **do** `/* while the simulator is not good enough */`

4 $\quad \mathrm{D}^{t+1} \leftarrow \mathrm{D}$

5 $\quad$ **for** $z' \in \mathcal{Z}$ **do** `/* improve the simulator towards the distinguisher direction */`

6 $\quad\quad h^{t+1}(x,z') \leftarrow h^t(x,z') + \gamma \cdot \overline{\mathrm{D}^{t+1}}(x,z')$

7 $\quad t \leftarrow t+1$

8 $\quad m \leftarrow 0$

9 $\quad$ **for** $z' \in \mathcal{Z}$ **do** `/* locate the biggest negative point mass */`

10 $\quad\quad$ **if** $h^t(x,z') < m$ **then**

11 $\quad\quad\quad m \leftarrow h^t(x,z')$

12 $\quad\quad\quad z^- \leftarrow z'$

13 $\quad h^t(x,z^-) = 0$ `/* cut the biggest negative mass */` **for** $z' \in \mathcal{Z}$ **do**

14 $\quad\quad h^t(x,z') \leftarrow h^t(x,z') + \frac{m}{|\mathcal{Z}|-1}$ `/* redestribute the cut mass */`

15 **return** $h^t(x,z)$

Proof. Consider the functions h^t. Define $\tilde{h}^{t+1}(x,z) \stackrel{def}{=} h^t(x,z) + \gamma \cdot \overline{\mathrm{D}}^{t+1}(x,z)$. According to Algorithm 1, we have

$$h^{t+1}(x,z) = h^t(x,z) + \gamma \cdot \overline{\mathrm{D}}^{t+1}(x,z) + \theta^{t+1}(x,z) \tag{3}$$

with the correction term $\theta^t(x,z)$ that be computed recursively as (see Line 13 in Algorithm 1)

$$\theta^t(x,z) = 0$$
$$\theta^t(x,z) = \begin{cases} -\min\left(h^t(x,z) + \gamma \cdot \overline{\mathrm{D}}^{t+1}(x,z), 0\right), & \text{if } z = z^t_{\min}(x) \\ \frac{\min\left(h^t(x,z^t_{\min}(x))) + \gamma \cdot \overline{\mathrm{D}}^{t+1}(x,z^t_{\min}(x)), 0\right)}{\#\mathcal{Z}-1} & \text{if } z \neq z^t_{\min}(x) \end{cases} \quad t = 0,1,\ldots \tag{4}$$

where $z^t_{\min}(x)$ is one of the points z minimizing $h^t(x,z) + \gamma \cdot \overline{\mathrm{D}}^{t+1}(x,z)$ (chosen and fixed for every t) . In particular

$$h^t(x, z^t_{\min}(x))) + \gamma \cdot \overline{\mathrm{D}}^{t+1}(x, z^t_{\min}(x)) < 0 \iff \exists z: \; h^t(x,z) + \gamma \cdot \overline{\mathrm{D}}^{t+1}(x,z) < 0 \tag{5}$$

<u>Notation</u>: for notational convenience we indenify the functions $\mathrm{D}^t(x,z)$, $\overline{\mathrm{D}}^t(x,z)$, $\theta^t(x,z)$, $\tilde{h}^t(x,z)$ and $h^t(x,z)$ with matrices where x are columns and z are rows.

That is h_x^t denotes the $|\mathcal{Z}|$-dimensional vector with entries $h^t(x,z)$ for $z \in \mathcal{Z}$ and similarly for other functions $\mathrm{D}^t(x,z)$, $\overline{\mathrm{D}}^t(x,z)$, $\theta^t(x,z)$, $\tilde{h}^t(x,z)$.

Claim 1 (Complextity of Algorithm 1). T executions of the "while loop" can be realized with time $O\left(T \cdot |\mathcal{Z}| \cdot \mathrm{size}(\mathcal{D})\right)$ and memory $O(|\mathcal{Z}|)$. [11].

This claim describes precisely resources required to compute the function h^T for every T. In order to bound T, we define the energy function as follows:

Claim 2 (Energy function). Define the auxiliary function

$$\Delta^t = \sum_{i=0}^{t-1} \mathop{\mathbb{E}}_{x \sim X} \left[\overline{\mathrm{D}}_x^{i+1} \cdot \left(g_x - h_x^i \right) \right]. \tag{6}$$

Then we have $\Delta^t = E_1 + E_2$ where

$$\begin{aligned} E_1 &= \tfrac{1}{\gamma} \mathbb{E}_{x \sim X} \left[\left(h_x^t - h_x^0 \right) \cdot g_x + \tfrac{1}{2} \textstyle\sum_{i=0}^{t-1} \left(h_x^{i+1} - h_x^i \right)^2 - \tfrac{1}{2} \left(\left(h_x^t \right)^2 - \left(h_x^0 \right)^2 \right) \right] \\ E_2 &= \tfrac{1}{\gamma} \mathbb{E}_{x \sim X} \left[- \textstyle\sum_{i=0}^{t-1} \theta_x^{i+1} \cdot \left(g_x - h_x^{i+1} \right) - \textstyle\sum_{i=0}^{t-1} \theta_x^{i+1} \cdot \left(h_x^{i+1} - h_x^i \right) \right] \end{aligned} \tag{7}$$

Note that all the symbols represent vectors and multiplications, including squares, should be understood as scalar products. The proof is based on simple algebraic manipulations and appears in Appendix B.

Remark 3 (Technical issues and intuitions). To upper-bound the formulas in Equation (7), we need the following important properties

(a) *Boundedness of correction terms*, that is ideally $|\theta^i(x.z)| = O(\mathrm{poly}(|\mathcal{Z}|) \cdot \gamma)$.
(b) *Acute angle between the correction and the error*, that is $\theta_x^i \cdot (g_x - h_x^i) \geqslant 0$.

Below we present an outline of the proof, discussing more technical parts in the appendix.

Proof outline. Indeed, with these assumptions we prove an upper bound on the energy function, namely

$$E_1 + E_2 \leqslant O\left(\mathrm{poly}(|\mathcal{Z}|) \cdot \left(t\gamma + \gamma^{-1} \right) \right), \tag{8}$$

which follows from the properties (a) and (b) above (they are proved in Claim 4 and Claim 3 below, and the inequality on $E_1 + E_2$ is derived in Claim 5). Note that, except a factor $\mathrm{poly}(|\mathcal{Z}|)$, our formula (not the proof, though) is identical to the bound used in [TTV09] (see Claim 3.4 in the eprint version). Indeed, our theorem is, to some extent, an extension to the main result in [TTV09] to cover the conditional case, where $|\mathcal{X}| > 1$. The main difference is that we show how to simulate a short leakage $|Z|$ given X, whereas [TTV09] shows how to simulate

[11] The RAM model

Z alone, under the assumption that the distribution of Z is dense in the uniform distribution (the min-entropy gap being small)[12].

Since the bound above is valid for any step t, and since on the other hand we have $t\epsilon \leqslant \Delta^t$ after t steps of the algorithm, we achieve a contradiction (to the number of steps) setting $\gamma = \epsilon/\mathrm{poly}(|\mathcal{Z})$. Indeed, suppose that $t\epsilon \leqslant A|\mathcal{Z}|^B(\gamma^{-1} + t\gamma)$ for some positive constants A, B. Since the step size γ can be chosen arbitrarily, we can set $\gamma = \frac{\epsilon}{2A|\mathcal{Z}|^B}$ which yields $\frac{t\epsilon}{2} \leqslant \frac{2A^2|\mathcal{Z}|^B}{\epsilon}$ or $t \leqslant 4A^2|\mathcal{Z}|^B\epsilon^{-2}$, which means that the algorithm terminates after at most $T = \mathrm{poly}(|\mathcal{Z}|)\epsilon^{-2}$ steps. Our proof goes exactly this way, except some extra optimization do obtain better exponent A.

We stress that it outputs only a *signed measure*, not a probability distribution yet. However, because of property (a) the negative mass is only of order $\mathrm{poly}(|\mathcal{Z}|)\epsilon$ and the function we end with can be simply rescaled (we replace negative masses by 0 and normalize the function dividing by a factor $1 - m$ where m is the total negative mass). With this transformation, we get the advantage $O(\mathrm{poly}(|\mathcal{Z}|)\epsilon)$ and lose an extra factor $O(|\mathcal{Z}|)$ in the complexity. Finally, we need to remember that we construct only a probability distribution function, not a sampler. Transforming it into a sampler yields an overhead of $O(\mathcal{Z})$. T his discussion shows that it is possible to build a sampler of complexity $\mathrm{poly}(|\mathcal{Z}|)\epsilon^{-2}$ with respect to $\mathcal{D}$, with advantage at most $O(\mathrm{poly}(|\mathcal{Z}|)\epsilon)$. Changing ϵ accordingly ($\epsilon \leftarrow \epsilon/\mathrm{poly}(|\mathcal{Z}|)$ we obtain the desired tradeoff.

A more careful inspection of the proof shows that we can actually achieve the claimed bound $|\mathcal{Z}|^5\epsilon^{-2}$ (see Remark 4 at the end of the proof).

Technical Discussion We note that condition (b) somehow means that mass cuts should go in the right direction, as it is much simpler to prove that Algorithm 1 terminates when there are no correction terms θ^t; thus we don't want to go in a wrong direction and ruin the energy gain. Concrete bounds on properties (a) and (b) are given in Claims 3 and 4.

In Algorithm 1 in every round we shift only one negative point mass (see Line 13). However, since this point mass is chosen to be as big as possible and since h^{t+1} and h^t differ only by a small term $\gamma \cdot \overline{\mathrm{D}}^{t+1}$ except the mass shift θ^{t+1}, one can expect that we have the negative mass under control. Indeed, this is stated precisely in Claim 3 below.

Claim 3 (The total negative mass is small). Let

$$\mathsf{NegativeMass}(h^t(x,\cdot)) = -\sum_z \min(h^t(x,z), 0)$$

be the total negative mass in $h^t(x,z)$ as the function of z. Then we have

$$\mathsf{NegativeMass}(h^t(x,\cdot) < |\mathcal{Z}|^3\gamma. \tag{9}$$

[12] It's not possible to extend the result from [TTV09] directly, the issue is that the constraints on the marginal distribution are not preserved. That's why [JP14] and this paper require much more extra work.

for every x and every t. In fact, for all x, z and t we have the following stronger bound

$$\max_z \left|\min(h^t(x,z),0)\right| < |\mathcal{Z}|\gamma.$$

The proof is based on a recurrence relation that links $\mathsf{NegativeMass}(h^{t+1}(x,\cdot)$ with $\mathsf{NegativeMass}(h^t(x,\cdot)$, and appears in Appendix C.

Claim 4 (The angle formed by the correction and the difference vector is acute). For every x, t we have $\mathsf{Angle}\left(\theta_x^{t+1}, g_x - h_x^{t+1}\right) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

The proof appears in Appendix D.

Having established Claims 3 and 4 we are now in position to prove a concrete bound in Equation (8). To this end, we give upper bounds on E_1 and E_2, defined in Equation (7), separately.

Claim 5 (Algorithm 1 terminates after a small number of steps.). The energy function in Claim 2 can be bounded as follows

$$E_1 \leqslant \gamma^{-1}\left(1 + 2|\mathcal{Z}|^2\gamma + |\mathcal{Z}|t\gamma^2 + |\mathcal{Z}|^3 t\gamma^2\right), \quad E_2 \leqslant 2|\mathcal{Z}|^2 t\gamma.$$

In particular, we conclude that with $\gamma = \frac{\epsilon}{8|\mathcal{Z}|^4}$ the algorithm terminates after at most $t = O(|\mathcal{Z}|^3)\epsilon^{-2}$ steps.

First, note that by Claim 4 we have $-\sum_{i=0}^{t-1}\theta_x^{i+1}\cdot\left(g_x - h_x^{i+1}\right) \leqslant 0$. Second, by definition of the sequence $(h^i)_i$ we have $-\sum_{i=0}^{t-1}\theta_x^{i+1}\cdot\left(h_x^{i+1} - h_x^i\right) = -\sum_{i=0}^{t-1}\theta_x^{i+1}\cdot\theta_x^{i+1} - \sum_{i=0}^{t-1}\gamma\theta_x^{i+1}\cdot\overline{\mathrm{D}}_x^{i+1}$ which is at most $2|\mathcal{Z}|^3 t\gamma^2$, because of Equation (9) (the sum of absolute correction terms $\sum_z |\theta^{i+1}(x,z)|$ is, by definition, twice the total negative mass, and $|\overline{\mathrm{D}}^{i+1}(x,z)| \leqslant 1$). This proves that

$$E_2 \leqslant \frac{1}{\gamma}\cdot 2|\mathcal{Z}|^3 t\gamma^2 = 2|\mathcal{Z}|^3 t\gamma.$$

To bound E_1, note that we have to bounds two non-negative terms, namely $\frac{1}{2}\sum_i\left(h_x^{i+1} - h_x^i\right)^2$ and $\left(h_x^t - h_x^0\right)\cdot g_x$. As for the first one, we have

$$\left(h_x^{i+1} - h_x^i\right)^2 = \left(\gamma\overline{\mathrm{D}}_x^{i+1} + \theta_x^{i+1}\right)^2 \leqslant 2(\gamma\overline{\mathrm{D}}_x^{i+1})^2 + 2\left(\theta_x^{i+1}\right)^2,$$

where the inequality follows by the Cauchy-Schwarz inequality[13]. We trivially have $\left(\overline{\mathrm{D}}_x^{i+1}\right)^2 \leqslant |\mathcal{Z}|$ (because of $|\overline{\mathrm{D}}(x,z)| \leqslant 1$). By the definition of correction terms in Equation (4) we have $\left(\theta_x^{i+1}\right)^2 = \sum_z(\theta^{i+1}(x,z))^2 < 2(\theta^{i+1}(x,z_0))^2$, where $\theta^{i+1}(x,z_0)$ is the smallest negative mass , which is at most $(2|\mathcal{Z}|^3\gamma)^2$ by Equation (9) . Thus, we have $\left(h_x^{i+1} - h_x^i\right)^2 \leqslant 2|\mathcal{Z}|\gamma^2 + 8|\mathcal{Z}|^6\gamma^2$. To bund $\left(h_x^t - h_x^0\right)\cdot g_x$ note that $-h_x^0\cdot g_x \leqslant 0$ and that $h_x^t\cdot g_x \leqslant \max_z |h^t(x,z)|$ (because

[13] Or cam be concluded from the parallelogram identity $(x+y)^2 + (x-y)^2 = x^2 + y^2$

$g(x,z) \geqslant 0$ and $\sum_x g(x,z) = 1$) which means $h^t_x \cdot g_x \leqslant 1 + 2\mathsf{NegativeMass}(h^t_x)$ (as $\sum_z \max(h^t(x,z),0) = 1 - \sum_z \min(h^t(x,z),0) = 1 + \mathsf{NegativeMass}(h^t_x)$ and $-\sum_z \min(h^t(x,z),0) = \mathsf{NegativeMass}(h^t_x)$ by $\sum_z \max(h^t(x,z) = 1$ and the definition of the total negative mass). This allows us to estimate E_1 as follows

$$E_1 \leqslant \gamma^{-1}\left(1 + 2|\mathcal{Z}|^3\gamma + |\mathcal{Z}|t\gamma^2 + 4|\mathcal{Z}|^6 t\gamma^2\right)$$

After t steps, the energy is at least $t\epsilon$. On the other hand, it at most $E_1 + E_2$. Since $|\mathcal{Z}|, |\mathcal{Z}|^3 \leqslant |\mathcal{Z}|^6$, we obtain

$$t\epsilon < \gamma^{-1} + 2|\mathcal{Z}|^3 + 7|\mathcal{Z}|^6 t\gamma$$

Since this is true for any positive γ, we choose $\gamma = \frac{\epsilon}{14|\mathcal{Z}|^6}$, which gives us (slightly weaker than claimed)

$$t < 32|\mathcal{Z}|^6\epsilon^{-2}.$$

Remark 4 (Optimized bounds). By the second part of Claim 3 we have $|\theta^i(x,z)| < |\mathcal{Z}|\gamma$ for every x, z and i. An inspection of the discussion above shows that this allows us to improve the bounds on E_1, E_2

$$E_1 \leqslant \gamma^{-1}\left(1 + 2|\mathcal{Z}|^2\gamma + |\mathcal{Z}|t\gamma^2 + |\mathcal{Z}|^2 t\gamma^2\right), \quad E_2 \leqslant 2|\mathcal{Z}|^2 t\gamma$$

Setting $\gamma = \frac{\epsilon}{8|\mathcal{Z}|^2}$ we get $E_1 + E_2 \leqslant 20|\mathcal{Z}|^2\epsilon^{-1}$ and $t \leqslant 20|\mathcal{Z}|^2\epsilon^{-2}$.

This finishes the proof of the claim.

From Claim 5 we conclude that after $t = O\left(|\mathcal{Z}|^2\epsilon^{-2}\right)$ steps we end up with a function $h = h^t$ that is (s,ϵ)-indistinguishable from g, because the algorithm terminated (and, clearly, has the complexity at most $O\left(|\mathcal{Z}|^3\epsilon^{-2}\right)$ relative to circuits of size s (including an overhead of $O(|\mathcal{Z}|)$ to compute $\overline{\mathrm{D}}$ from D). To finish the proof, we need to solve two issues

Claim 6 (From the signed measure to the probability measure). Let h^t be the output of the algorithm. Define the probability distribution

$$h(x,z) = \frac{\max(h^t(x,z),0)}{\sum_{z'} \max(h^t(x,z'),0)}$$

for every x, z. Then $h^t(x,\cdot)$ and $h(x,\cdot)$ are $O(\epsilon)$-statistically close for every x.

To prove the claim, we note that $\sum_{z'} \max(h^t(x,z'),0)$ equals $1+\beta$ where $\beta = \mathsf{NegativeMass}(h^t(x,\cdot)$. Thus we have $|h(x,z) - h^t(x,z)| \leqslant |h^t(x,z)| \cdot \frac{\beta}{1+\beta}$. Since $\sum_{z'} |h^t(x,z')| = \sum_{z'} \max(h^t(x,z'),0) - \sum_{z'} \min(h^t(x,z'),0) = 1 + 2\beta$, we get $\sum |h(x,z) - h^t(x,z)| = O(\beta)$ which is $O(\epsilon)$ by Claim 3 for γ defined as in Claim 5.

Recall that we have constructed an approximating probability measure h for the probability mass function g, which is not a sampler yet. However, we can fix it by rejection sampling, as shown below.

Claim 7 (From the pmf to the sampler). There exists a (probabilistic) function $h_{\mathsf{sim}} : \mathcal{X} \to \mathcal{Z}$ which calls $h(x, z)$ (defined as above) at most $O(|\mathcal{Z}| \log(1/\epsilon))$ times and for every x the distribution of its output is ϵ-close to $h(x, \cdot)$ for every x.

The proof goes by a simple rejection sampling argument: we sample a point $z \leftarrow \mathcal{Z}$ at random and reject with probability $h(x, z)$. The rejection probability in one turn is $\frac{1}{|\mathcal{Z}|}$. If we repeat the experiment $|\mathcal{Z}| \log(1/\epsilon)|$ then the probability of rejection in every round is only ϵ. On the other hand, conditioned on the opposite event, we get the distribution identical to $h(x, \cdot)$. So the distance is at most ϵ as claimed. note that

The last two claims prove that the distribution of $h_{sim}(x)$ is $(s, O(\epsilon))$-close to $h^t_x = h^t(x, \cdot)$, for every x. Since h^t, as a function of x, z is (s, ϵ)-close to g, and g is the conditional distribution of $Z|X$, we obtain

$$X, h_{sim}(X) \approx^{s,O(\epsilon)} X, Z$$

and the complexity of the final sampler $h_{sim}(X)$ is $O(|\mathcal{Z}|^5 \epsilon^{-2})$

4 Time-success ratio under algebraic transformations

In Lemma 1 below we provide a quantitative analysis of how the time-success ratio changes under concrete formulas in security reductions.

Lemma 1 (Time-success ratio for algebraic transformations). *Let a, b, c and A, B, C be positive constants. Suppose that P' is secure against adversaries (s', ϵ'), whenever P is secure against adversaries (s, ϵ), where*

$$\begin{aligned} s' &= s \cdot c\epsilon^C - b\epsilon^{-B} \\ \epsilon' &= a\epsilon^A. \end{aligned} \tag{10}$$

In addition, suppose that the following condition is satisfied

$$A \leqslant C + 1. \tag{11}$$

Then the following is true: if P is 2^k-secure, then P' is $2^{k'}$-secure (in the sense of Definition 4) where

$$k' = \begin{cases} \frac{A}{B+C+1} k + \frac{A}{B+C+1} (\log c - \log b) - \log a, & b \geqslant 1 \\ \frac{A}{C+1} k + \frac{A}{C+1} \log c - \log a, & b = 0 \end{cases} \tag{12}$$

The proof is elementary though not immediate. It can be found in [Skó15].

Remark 5 (On the technical condition (11)). This condition is satisfied in almost all applications, at in the reduction proof typically ϵ' cannot be better (meaning higher exponent) than ϵ. Thus, quite often we have $A \leqslant 1$.

References

A More on the flaw in [JP14]

In the original setting we have $\mathcal{Z} = \{0,1\}^\lambda$. In the proof of the claimed better bound $O\left(s \cdot 2^{3\lambda}\epsilon^{-2}\right)$ there is a mistake on page 18 (eprint version), when the authors enforce a signed measure to be a probability measure by a mass shifting argument. The number M defined there is in fact a function of x and is hard to compute, whereas the original proof amuses that this is a constant independent of x. During iterations of the boosting loop, this number is used to modify distinguishers class step by step, which drastically blows up the complexity (exponentially in the number of steps, which is already polynomial in ϵ). In the min-max based proof giving the bound $O\left(s \cdot 2^{3\lambda}\epsilon^{-4}\right)$ a fixable flaw is a missing factor of 2^λ in the complexity (page 16 in the eprint version), which is because what is constructed in the proof is only a probability mass function, not yet a sampler [Pie15].

B Proof of Claim 2

We can rewrite Equation (6) as

$$\begin{aligned}\Delta^t &= \frac{1}{\gamma} \mathop{\mathbb{E}}_{x\sim X}\left[\sum_{i=0}^{t-1}\left(\left(h_x^{i+1} - h_x^i\right) - \theta_x^{i+1}\right)\cdot\left(g_x - h_x^i\right)\right] \\ &= \frac{1}{\gamma} \mathop{\mathbb{E}}_{x\sim X}\left[\sum_{i=0}^{t-1}\left(h_x^{i+1} - h_x^i\right)\cdot\left(g_x - h_x^i\right) - \sum_{i=0}^{t-1}\theta_x^{i+1}\cdot\left(g_x - h_x^i\right)\right] \end{aligned} \tag{13}$$

First, note that

$$\begin{aligned}\sum_{i=0}^{t-1}\left(h_x^{i+1} - h_x^i\right)\cdot\left(g_x - h_x^i\right) = \\ &= \left(h_x^t - h_x^0\right)\cdot g_x - \sum_{i=0}^{t-1} h_x^i \cdot \left(h_x^{i+1} - h_x^i\right) \\ &= \left(h_x^t - h_x^0\right)\cdot g_x + \frac{1}{2}\sum_{i=0}^{t-1}\left(h_x^{i+1} - h_x^i\right)\cdot\left(h_x^{i+1} - h_x^i\right) + \\ &\quad - \frac{1}{2}\sum_{i=0}^{t-1}\left(h_x^{i+1} + h_x^i\right)\cdot\left(h_x^{i+1} - h_x^i\right) \\ &= \left(h_x^t - h_x^0\right)\cdot g_x + \frac{1}{2}\sum_{i=0}^{t-1}\left(h_x^{i+1} - h_x^i\right)^2 - \frac{1}{2}\left(\left(h_x^t\right)^2 - \left(h_x^0\right)^2\right)\end{aligned} \tag{14}$$

As to the second term in Equation (13), we observe that

$$-\sum_{i=0}^{t-1}\theta_x^{i+1}\cdot\left(g_x-h_x^i\right)=-\sum_{i=0}^{t-1}\theta_x^{i+1}\cdot\left(g_x-h_x^{i+1}\right)-\sum_{i=0}^{t-1}\theta_x^{i+1}\cdot\left(h_x^{i+1}-h_x^i\right) \quad (15)$$

C Proof of Claim 3

Proof (Proof of Claim 3). We start by comparing the total negative mass in the functions $h^{t+1}=h^t+\overline{\mathrm{D}}^{t+1}+\theta^{t+1}$ and h^t. Suppose first that $\tilde{h}^t(x,z_0)<0$ where $z_0=z^t_{\min}(x)$. Since $\sum_{z\neq z_0}\tilde{h}^{t+1}=1-\tilde{h}^{t+1}(x,z_0)$, there exists z_1 such that $\tilde{h}^{t+1}(x,z_1)\geqslant\frac{1-\tilde{h}^{t+1}(x,z_0)}{|\mathcal{Z}|-1}>0$. Combining this with Equation (4) we obtain

$$h^{t+1}(x,z_1)=\tilde{h}^{t+1}(x,z_1)+\frac{\tilde{h}^{t+1}(x,z_0)}{|\mathcal{Z}|-1}\geqslant\frac{1}{|\mathcal{Z}|-1} \quad (16)$$

These observations together with Equation (3) give us

$$\begin{aligned}\sum_{z\in\mathcal{Z}}\min\left(h^{t+1}(x,z),0\right)&=\sum_{z\in\mathcal{Z}}\min\left(\tilde{h}^{t+1}(x,z)+\theta^{t+1}(x,z),0\right)\\&=\sum_{z\in\mathcal{Z}\setminus\{z_0,z_1\}}\min\left(\tilde{h}^{t+1}(x,z)+\frac{\tilde{h}^{t+1}(x,z_0)}{|\mathcal{Z}|-1},0\right)\\&\geqslant\sum_{z\in\mathcal{Z}\setminus\{z_0,z_1\}}\min\left(\tilde{h}^{t+1}(x,z),0\right)+(|\mathcal{Z}|-2)\cdot\frac{\tilde{h}^{t+1}(x,z_0)}{|\mathcal{Z}|-1}\\&=\sum_{z\in\mathcal{Z}}\min\left(\tilde{h}^{t+1}(x,z),0\right)+(|\mathcal{Z}|-2)\cdot\frac{\tilde{h}^{t+1}(x,z_0)}{|\mathcal{Z}|-1}-\tilde{h}^{t+1}(x,z_1)\\&=\sum_{z\in\mathcal{Z}}\min\left(\tilde{h}^{t+1}(x,z),0\right)+\min\left(\frac{\tilde{h}^{t+1}(x,z_0)}{|\mathcal{Z}|-1},0\right)\end{aligned} \quad (17)$$

where the inequality line follows from $\tilde{h}^{t+1}(x,z_0)<0$ and Equation (16). But by the definition of $z_0=z^t_{\min}(x)$ we have $\tilde{h}^{t+1}(x,z_0)=\min_z\tilde{h}^{t+1}(x,z)$. Since this value is negative, we get

$$\tilde{h}^{t+1}(x,z_0)\leqslant\frac{1}{|\mathcal{Z}|-1}\cdot\sum_{z\in\mathcal{Z}}\min\left(\tilde{h}^{t+1}(x,z),0\right) \quad (18)$$

Combining Equation (17) and Equation (18) we obtain

$$-\sum_{z\in\mathcal{Z}}\min\left(h^{t+1}(x,z),0\right)\leqslant-\left(1-\frac{1}{(|\mathcal{Z}|-1)^2}\right)\sum_{z\in\mathcal{Z}}\min\left(\tilde{h}^{t+1}(x,z),0\right). \quad (19)$$

Since $|h^{t+1}(x,z) - \tilde{h}^t(x,z)| \leqslant \gamma$ by Equation (3), we get the following recursion

$$-\sum_{z\in\mathcal{Z}} \min\left(h^{t+1}(x,z),0\right) \leqslant -\left(1-\frac{1}{(|\mathcal{Z}|-1)^2}\right)\sum_{z\in\mathcal{Z}} \min\left(h^t(x,z),0\right) + |\mathcal{Z}|\gamma \tag{20}$$

which can be rewritten as

$$\mathsf{NegativeMass}\left(h^{t+1}(x,\cdot)\right) < \left(1-\frac{1}{|\mathcal{Z}|^2}\right)\mathsf{NegativeMass}\left(h^t(x,\cdot)\right) + |\mathcal{Z}|\gamma. \tag{21}$$

which is in addition trivially true if $\tilde{h}^{t+1}(x,z) \geqslant 0$ for all z. Since we have $\mathsf{NegativeMass}\left(h^0(x,\cdot)\right) = 0$, expanding this recursion till $t=0$ gives an upper bound $|\mathcal{Z}|\gamma \cdot \sum_{j\leqslant t+1}\left(1-|\mathcal{Z}|^{-2}\right)^j$ which is smaller than by $|\mathcal{Z}|^3\gamma$ by the convergence of the geometric series. This finishes the proof of the first part.

To prove the second part, recall that by the definition of z_0 we have $\tilde{h}^{t+1}(x,z_0) = \min_z \tilde{h}^{t+1}(x,z)$. Suppose that $\tilde{h}^{t+1}(x,z_0) < 0$ (that is, there is a negative mass in $\widetilde{h}^{t+1}(x,\cdot)$). Now, by the definition of h^{t+1}, we get

$$\begin{aligned}\max_z \left|\min(h^{t+1}(x,z),0)\right| &= \max_{z\neq z_0}\left|\min(h^{t+1}(x,z),0)\right| \\ &= \max_{z\neq z_0}\left|\min\left(\widetilde{h}^{t+1}(x,z) + \frac{|\widetilde{h}^{t+1}(x,z_0)|}{|\mathcal{Z}|-1},0\right)\right|.\end{aligned}$$

Suppose that $\widetilde{h}^{t+1}(x,z) + \frac{|\widetilde{h}^{t+1}(x,z_0)|}{|\mathcal{Z}|-1} \leqslant 0$ for some z. Then, by the definition of z_0, we also have

$$\begin{aligned}0 &\geqslant \widetilde{h}^{t+1}(x,z) + \frac{|\widetilde{h}^{t+1}(x,z_0)|}{|\mathcal{Z}|-1} \\ &\geqslant \widetilde{h}^{t+1}(x,z_0) + \frac{|\widetilde{h}^{t+1}(x,z_0)|}{|\mathcal{Z}|-1} \\ &= -\left(1-\frac{1}{|\mathcal{Z}|-1}\right)\left|\widetilde{h}^{t+1}(x,z_0)\right|.\end{aligned}$$

From this we conclude that for *any* z we have

$$\min\left(\widetilde{h}^{t+1}(x,z) + \frac{|\widetilde{h}^{t+1}(x,z_0)|}{|\mathcal{Z}|-1},0\right) \geqslant -\left(1-\frac{1}{|\mathcal{Z}|-1}\right)\left|\widetilde{h}^{t+1}(x,z_0)\right|.$$

and thus

$$\max_{z\neq z_0}\left|\min\left(\widetilde{h}^{t+1}(x,z) + \frac{|\widetilde{h}^{t+1}(x,z_0)|}{|\mathcal{Z}|-1},0\right)\right| \leqslant \left(1-\frac{1}{|\mathcal{Z}|-1}\right)\left|\widetilde{h}^{t+1}(x,z_0)\right|$$

which means that (still assuming that $\widetilde{h}^{t+1}(x,z_0) < 0$)

$$\max_z\left|\min(h^{t+1}(x,z),0)\right| \leqslant \left(1-\frac{1}{|\mathcal{Z}|-1}\right)\max_z\left|\min\left(\widetilde{h}^{t+1}(x,z),0\right)\right|.$$

Note that $0 \geqslant \min\left(\widetilde{h}^{t+1}(x,z),0\right) \geqslant \min\left(h^t(x,z),0\right) - \gamma$ by the definition of h^{t+1} and $\widetilde{h}^{t+1}$. Then

$$\max_z \left|\min(h^{t+1}(x,z),0)\right| \leqslant \left(1 - \frac{1}{|\mathcal{Z}|-1}\right) \max_z \left|\min(h^t(x,z),0)\right| + \gamma.$$

Note that this inequality is true even if $\widetilde{h}^{t+1}(x,z_0) = 0$, that is $\widetilde{h}^{t+1}(x,z) \geqslant 0$ for all z as then $h^{t+1}(x,z) \geqslant 0$ for all z. By expanding this recursion, and noticing that $\min(h^0(x,z),0) = 0$ for all x,z by definition, we get

$$\max_z \left|\min(h^{t+1}(x,z),0)\right| \leqslant \gamma \sum_{j=0}^{t} \left(1 - \frac{1}{|\mathcal{Z}|-1}\right)^j < |\mathcal{Z}|\gamma.$$

D Proof of Claim 4

Proof. If $\theta^{t+1}(x,z) = 0$ then there is nothing to prove. Suppose that $\theta^{t+1}(x,z) < 0$. Let $z_0 = z^t_{\min}(x)$. According to Equation (4) we have $\theta^{t+1}(x,z_0) = -\tilde{h}^{t+1}(x,z_0)$ and $\theta^{t+1}(x,z) = \frac{\tilde{h}^{t+1}(x,z_0)}{\#\mathcal{Z}-1}$ for $z \neq z_0$. Therefore

$$\begin{aligned}
\theta^{t+1}_x \cdot \left(g_x - \tilde{h}^{t+1}_x\right) &= -\tilde{h}^{t+1}(x,z_0)\left(g(x,z_0) - \tilde{h}^{t+1}(x,z_0)\right) + \\
&\quad + \sum_{z \neq z_0} \frac{\tilde{h}^{t+1}(x,z_0)}{|\mathcal{Z}|-1} \cdot \left(g(x,z) - \tilde{h}^{t+1}(x,z)\right) \\
&= -\tilde{h}^{t+1}(x,z_0)\left(g(x,z_0) - \tilde{h}^{t+1}(x,z_0)\right) \\
&\quad - \frac{\tilde{h}^{t+1}(x,z_0)}{|\mathcal{Z}|-1}\left(g(x,z_0) - \tilde{h}^{t+1}(x,z_0)\right) \qquad (22)
\end{aligned}$$

and

$$-\theta^{t+1}_x \cdot \theta^{t+1}_x = -\tilde{h}^{t+1}(x,z_0) \cdot \tilde{h}^{t+1}(x,z_0)\left(1 + \frac{1}{|\mathcal{Z}-1|}\right). \qquad (23)$$

Putting Equations (22) and (23) together we obtain

$$\begin{aligned}
\theta^{t+1}_x \cdot \left(g_x - h^{t+1}_x\right) &= \theta^{t+1}_x \cdot \left(g_x - \tilde{h}^{t+1}_x\right) - \theta^{t+1}_x \cdot \theta^{t+1}_x \\
&= -\left(1 + \frac{1}{|\mathcal{Z}|-1}\right) \tilde{h}^{t+1}(x,z_0) \cdot g(x,z_0)
\end{aligned}$$

which is positive because $\tilde{h}^{t,r}(x,z_0) < 0$ and $g(x,z_0) \geqslant 0$. This proves Claim 4.