A Tight Lower Bound for Entropy Flattening

Yi-Hsiu Chen*1, Mika Göös†1, Salil P. Vadhan‡2, and Jiapeng Zhang§3

1School of Engineering and Applied Sciences, Harvard University, USA
2Computer Science and Applied Mathematics, Harvard University, USA
3University of California San Diego, USA

July 19, 2018

Abstract

We study *entropy flattening*: Given a circuit $\mathcal{C}_X$ implicitly describing an n-bit source X (namely, X is the output of $\mathcal{C}_X$ on a uniform random input), construct another circuit $\mathcal{C}_Y$ describing a source Y such that (1) source Y is nearly *flat* (uniform on its support), and (2) the Shannon entropy of Y is monotonically related to that of X. The standard solution is to have $\mathcal{C}_Y$ evaluate $\mathcal{C}_X$ altogether $\Theta(n^2)$ times on independent inputs and concatenate the results (correctness follows from the asymptotic equipartition property). In this paper, we show that this is optimal among *black-box* constructions: Any circuit $\mathcal{C}_Y$ for entropy flattening that repeatedly queries $\mathcal{C}_X$ as an oracle requires $\Omega(n^2)$ queries.

Entropy flattening is a component used in the constructions of pseudorandom generators and other cryptographic primitives from one-way functions [HILL99, Rom90, Hol06, HHR06, HRVW09, HRV13, HHR+10, VZ12]. It is also used in reductions between problems complete for statistical zero-knowledge [Oka00, SV97, GSV99a, Vad99]. The $\Theta(n^2)$ query complexity is often the main efficiency bottleneck. Our lower bound can be viewed as a step towards proving that the current best construction of pseudorandom generator from arbitrary one-way functions by Vadhan and Zheng (STOC 2012) has optimal efficiency.

*yihsiuchen@g.harvard.edu. Supported by NSF grant CCF-1749750
†mika@seas.harvard.edu. Supported by Michael O. Rabin Postdoctoral Fellowship.
‡salil_vadhan@harvard.edu. Supported by NSF grant CCF-1749750.
§jpeng.zhang@gmail.com. Supported by NSF CCF-1614023

ISSN 1433-8092

1 Introduction

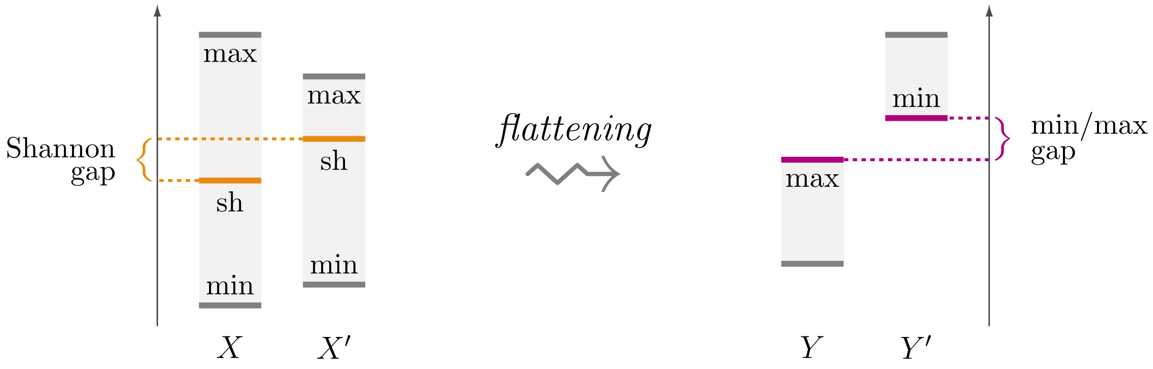
A *flat* source X is a random variable that is uniform on its support; equivalently, its Shannon entropy, min-entropy, and max-entropy are all equal:

$$\begin{aligned} \mathsf{H}_{\text{sh}}(X) &= \mathbb{E}_{x\sim X}\left[\log(1/\Pr[X=x])\right], \\ \mathsf{H}_{\min}(X) &= \min_x \log(1/\Pr[X=x]), \\ \mathsf{H}_{\max}(X) &= \log|\operatorname{Supp} X|. \end{aligned}$$

These can be far apart for *non-flat* sources, but we always have $\mathsf{H}_{\min}(X) \leq \mathsf{H}_{\text{sh}}(X) \leq \mathsf{H}_{\max}(X)$.

Entropy flattening is the following task: Given a circuit $\mathcal{C}_X$ implicitly describing an n-bit source X (namely, X is the output of $\mathcal{C}_X$ on a uniform random input), efficiently construct another circuit $\mathcal{C}_Y$ describing a "flattened" version Y of X. The goal is to have the output source Y (or a small statistical modification of it) be such that its min- and max-entropies are *monotonically* related to the Shannon entropy of X. Concretely, one interesting range of parameters is:

- if two *input* sources X and X' exhibit a 1-bit Shannon entropy gap, $\mathsf{H}_{\text{sh}}(X') \geq \mathsf{H}_{\text{sh}}(X) + 1$,
- then the two respective *output* sources Y and Y' must witness $\mathsf{H}_{\min}(Y') \geq \mathsf{H}_{\max}(Y) + 1$ (modulo a small modification to Y and Y').



Entropy flattening is used as an ingredient in constructions of pseudorandom generators and other cryptographic primitives from one-way functions [HILL99, Rom90, Hol06, HHR06, HRVW09, HRV13, HHR+10, VZ12] and in reductions between problems complete for (non-interactive) statistical zero-knowledge [Oka00, SV97, GSV99a, Vad99]. See Section 1.2 for a detailed discussion.

A solution: repeat X. The standard strategy for entropy flattening is to construct Y as the concatenation X^q of some q i.i.d. copies of the input source X. That is, in circuit language, $\mathcal{C}_Y(x_1,\ldots,x_q) = (\mathcal{C}_X(x_1),\ldots,\mathcal{C}_X(x_q))$. The well-known *asymptotic equipartition property* in information theory states that X^q is ε-close[1] to having min- and max-entropies closely approximated by $q\cdot\mathsf{H}_{\text{sh}}(X)$. (It is common to say that X^q has a certain ε-*smooth* min- and max-entropy [RW04].)

[1]Random variables Z_1 and Z_2 are ε-*close* if $d_{\text{TV}}(Z_1,Z_2) \leq \varepsilon$ where $d_{\text{TV}}(Z_1,Z_2)$ is the usual statistical (or total variation) distance, given by $d_{\text{TV}}(Z_1,Z_2) = \max_{T\subseteq\mathcal{Z}} |\Pr[Z_1\in T] - \Pr[Z_2\in T]|$.

Lemma 1.1 ([HILL99, HR11]). *Let X be an n-bit random variable. For any $q \in \mathbb{N}$ and $\varepsilon > 0$ there is an nq-bit random variable Y' that is ε-close to X^q such that*

$$\mathsf{H}_{\min}(Y'),\ \mathsf{H}_{\max}(Y') \ \in\ q \cdot \mathsf{H}_{\text{sh}}(X) \pm O\big(n\sqrt{q\log(1/\varepsilon)}\big).$$

In particular, it suffices to set $q = \tilde{\Theta}(n^2)$ in order to flatten entropy in the aforementioned interesting range of parameters (1-bit Shannon gap implies at least 1-bit min/max gap). The analysis here is also tight by a reduction to standard anti-concentration results: it is *necessary* to have $q = \Omega(n^2)$ in order for the construction $Y = X^q$ to flatten entropy.

1.1 Our Result

We show that any *black-box* construction for entropy flattening—that is, a circuit $\mathcal{C}_Y$ which treats $\mathcal{C}_X$ as a black-box oracle—requires $\Omega(n^2)$ oracle queries to $\mathcal{C}_X$. This is formalized in Theorem 1.1 below.

In particular, the simple "repeat-X" strategy is optimal among all black-box constructions. Besides querying $\mathcal{C}_X$ on independent inputs, a black-box algorithm has the freedom to perform adaptive queries, and it can produce outputs that are arbitrary functions of its query/answer execution log (rather than merely concatenating the answers). For example, this allows the use of hash functions and randomness extractors, which is indeed useful for variations of the flattening task (e.g., Lemma 2.1 below).

Query model. In our black-box model, the input source is now encoded as the output distribution of an *arbitrary* function $f : \{0,1\}^n \to \{0,1\}^m$ where $m = \Theta(n)$ (not necessarily computed by a small circuit); namely, the input source is $f(U_n)$ where U_n denotes the uniform distribution over n-bit strings. We consider oracle algorithms A^f that have query access to f. Given an n'-bit input w (thought of as a random seed) to A^f, the algorithm computes by repeatedly querying f (on query $x \in \{0,1\}^n$ it gets to learn $f(x)$), until it finally produces some m'-bit output string $A^f(w)$. We denote by $A^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ the function computed by A^f. Thus $A^f(U_{n'})$ is the output source.

Inputs/outputs. Our input sources come from the promise problem *Entropy Approximation* (EA); the circuit version of this problem is complete for the complexity class **NISZK** (non-interactive statistical zero-knowledge), as shown by Goldreich, Sahai, and Vadhan [GSV99a]. The EA promise problem is (here $\tau \in \mathbb{N}$ is a threshold parameter):

- YES input: (f, τ) such that $\mathsf{H}_{\text{sh}}(f(U_n)) \geq \tau + 1$.
- NO input: (f, τ) such that $\mathsf{H}_{\text{sh}}(f(U_n)) \leq \tau - 1$.

The goal of a flattening algorithm A^f (which also gets τ as input, but we supress this in our notation) is to produce an output distribution that is statistically close to having high min-entropy or low max-entropy depending on whether the input source f is a YES or a NO instance. We say that A^f is an (ε, Δ)-*flattening algorithm* if (here $\kappa = \kappa(\tau)$ is a parameter that A^f gets to choose):

- If (f, τ) is a YES input, then $A^f(U_{n'})$ is ε-close to a distribution Z_H with $\mathsf{H}_{\min}(Z_H) \geq \kappa + \Delta$.
- If (f, τ) is a NO input, then $A^f(U_{n'})$ is ε-close to a distribution Z_L with $\mathsf{H}_{\max}(Z_L) \leq \kappa - \Delta$.

The result. Our main result is the following.

Theorem 1.1. *There exist constants $\varepsilon, \Delta > 0$ such that every (ε, Δ)-flattening algorithm for n-bit oracles f requires $\Omega(n^2)$ oracle queries.*

In fact, our proof yields an even more fine-grained lower bound. Suppose we allow ε and Δ to vary subject to $n/25 \geq \Delta \geq \log(1/\varepsilon)$. Then our lower bound becomes $\Omega(n^2 \log(1/\epsilon))$, which is tight in both n and ε.

1.2 Relevance to Cryptographic Constructions

Pseudorandom generators from one-way functions. The use of flattening in complexity-based cryptography originates with the celebrated work of Håstad, Impagliazzo, Levin, and Luby (HILL) [HILL99], which showed how to construct a pseudorandom generator from any one-way function. The first step of their construction is to show how to obtain, from any one-way function, a *pseudoentropy generator*. That is, a polynomial-time computable function $f : \{0,1\}^n \to \{0,1\}^m$ such that $f(U_n)$ is computationally indistinguishable from a random variable Y such that $\mathsf{H}_{\text{sh}}(Y)$ is noticeably higher than $\mathsf{H}_{\text{sh}}(f(U_n))$. In other words, for some threshold τ_n and a nonnegligible gap parameter $\Delta_n \geq 1/\operatorname{poly}(n)$ it holds that:

1. $f(U_n)$ is computationally indistinguishable from a random variable Y with Shannon entropy at least $\tau_n + \Delta_n$, and
2. $f(U_n)$ has Shannon entropy at most $\tau_n - \Delta_n$.

Notice that if $\Delta_n = 1$, then Condition 2 says that the pair (f_n, τ_n) is a NO instance of EA (where f_n is the restriction of f to instances of length n). On the other hand, Condition 1 says that (f_n, τ_n) appears to be a YES instance of EA to computationally bounded algorithms (that get to observe an output of f_n on a uniformly random input). In the HILL construction, it turns out that $\Delta_n = \tilde{\Theta}(1/n)$ (rather than $\Delta_n = 1$), corresponding to an appropriate variant of EA.

Given the similarity with EA, it is natural that the next step of the HILL construction is flattening. Specifically evaluating f on many independent inputs yields a distribution that is close to having low max-entropy yet is computationally indistinguishable from having high min-entropy. Since Δ_n is not 1, but rather $\tilde{\Theta}(1/n)$, the number of copies needed for flattening becomes $q = \tilde{O}(n/\Delta_n)^2 = \tilde{\Theta}(n^4)$.

After flattening, universal hashing (or randomness extraction) is applied to obtain a pseudorandom generator $G^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$, where $G^f(U_{n'})$ is computationally indistinguishable from $U_{m'}$ (i.e. indistinguishable from min-entropy at least m') yet has max-entropy at most $n' \leq m' - 1$ (due to having a seed length of n'). (This step is a computational analogue of Lemma 2.1 below.)

As described, the pseudorandom generator G^f makes $q = \tilde{\Theta}(n^4)$ queries to the pseudoentropy generator f and hence to the one-way function. The actual HILL construction is more complex and inefficient, due in part to the fact that the entropy threshold τ_n is not known. To deal with the latter issue, they enumerate all $t = \Theta(n/\Delta_n) = \tilde{\Theta}(n^2)$ possibilities for the threshold τ_n to within precision Δ_n, construct a pseudorandom generator for each choice, and then combine the generators (which has a further cost in efficiency).

A series of subsequent works [Hol06, HHR06, HRV10, VZ12] improved the efficiency of the HILL construction. The state-of-the-art constructions [HRV10, VZ12] replace "pseudoentropy" with a more relaxed computational analogue of Shannon entropy ("next-bit pseudoentropy") and thereby obtain $\Delta_n = 1$ (or even $\Delta_n = \log n$), reducing the cost of flattening to $q = \tilde{\Theta}(n/\Delta_n)^2 = \tilde{\Theta}(n^2)$. In these constructions, the entropy threshold τ_n is also known (in fact $\tau_n = n$), but there still is an analogous cost of $\tilde{\Theta}(n)$ due to the fact that we don't know how the entropy is spread out among the bits of the output of the next-bit pseudoentropy generator f.

Overall, with the most efficient constructions to date, the pseudorandom generator makes $\tilde{\Theta}(n^3)$ queries to the one-way function, of which a $\tilde{\Theta}(n^2)$ factor is due to flattening. This complexity renders the constructions too inefficient for practice, and thus it is important to know whether a more efficient construction is possible.

Lower Bounds. The work of Gennaro, Gertner, Katz, and Trevisan [GGKT05] gave the first lower bound on constructing pseudorandom generators from one-way functions. Specifically they proved that any "black-box" construction of a pseudorandom generator $G^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ from a one-way function $f : \{0,1\}^n \to \{0,1\}^m$ requires $\Omega((m'-n')/\log n)$ queries to f. Thus, many queries are needed to construct a pseudorandom generator with large stretch. However, their lower bound says nothing about the number of queries needed to obtain a pseudorandom generator with small stretch (i.e., where $m' = n' + O(\log n)$), and indeed it applies even to one-way permutations f, where no flattening is needed and a pseudorandom generator with small stretch can be obtained with a single query to the one-way function [GL89].

For constructing pseudorandom generators with small stretch from one-way functions, Holenstein and Sinha [HS12] proved that any black-box construction requires $\tilde{\Omega}(n)$ queries. Their lower bound also does not tell us about flattening, as it applies even to *regular* one-way functions, which directly (with one query) give a separation between pseudo-*min*-entropy and max-entropy. Rather, their lower bound corresponds to the efficiency costs coming from not knowing the entropy thresholds τ_n mentioned above (or how the entropy is spread across the bits in the case of next-bit pseudoentropy).

Our lower bound for flattening (Theorem 1.1) can be viewed as a first-step towards proving that any black-box construction of pseudorandom generators from one-way functions requires $\tilde{\Omega}(n^2)$ queries. One might hope to also combine this with [HS12] and obtain a lower bound of $\tilde{\Omega}(n^3)$ queries, which would match the best-known construction of [VZ12].

Seed length. Another important and well-studied efficiency criterion for pseudorandom generator constructions is how the seed length n' of the pseudorandom generator $G^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ depends on the input length n of the one-way function $f : \{0,1\}^n \to \{0,1\}^m$. The standard method for flattening (Lemma 1.1) requires independent samples from the distribution being flattened, and thus the query complexity of flattening contributes a multiplicative factor to the seed length of the pseudorandom generator. For example, the construction of [VZ12] gives a pseudorandom generator with seed length $\tilde{\Theta}(n^2) \cdot n = \tilde{\Theta}(n^3)$, as $\tilde{\Theta}(n^2)$ independent evaluations of the one-way function (or corresponding pseudoentropy generator) are used for flattening. An interesting open problem is to show that independent evaluations are indeed necessary, and extend our lower bound on query complexity to a lower bound on the input length n' of the flattening algorithm $A^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$. This could be a first step towards proving a superlinear lower bound on the seed length of pseudorandom generators constructed (in a black-box way) from one-way functions, a long-standing open problem. We note that the existing lower bounds on query complexity of [GGKT05, HS12] cannot be turned into seed length lower bounds, as there are constructions of large-stretch pseudorandom generators from regular one-way functions with seed length $\tilde{O}(n)$ [HHR06]. That is, although those constructions make polynomially many queries to the one-way functions, the queries are highly correlated (and even adaptive).

Other Cryptographic Primitives. Flattening is also an efficiency bottleneck in the constructions of other cryptographic primitives from arbitrary one-way functions, such as universal one-way hash functions [Rom90, KK05, HHR+10] and statistically hiding commitment schemes [HNO+09, HRVW09]. In both cases, the state-of-the-art constructions begin by constructing a function f where there is a gap between its output entropy $H(f(U_n))$ and a computational analogue of Shannon entropy (namely, a form of "inaccessible entropy"). Then flattening is applied, after which some (possibly interactive) hashing techniques are used to obtain the final cryptographic primitive. Again, our lower bound on flattening can be viewed as a first step towards proving an efficiency lower bound on black-box constructions.

We note that there was a very fruitful interplay between this sequence of works on constructions of cryptographic primitives from one-way functions and general results about **SZK** and **NISZK**, with inspirations going in both directions (e.g., [NV06, HNO+09, OV08, HRVW09]). This reinforces the feeling that our lower bound for Flattening the **NISZK**-complete problem EA can help in understanding the aforementioned constructions.

2 Proof Overview

Our proof builds on the recent result of Lovett and Zhang [LZ17], who showed that there is no efficient black-box reduction (making polynomially many queries) from EA to its complement, thereby giving evidence that **NISZK** is not closed under complement and hence that **NISZK** $\neq$ **SZK**. The result of [LZ17] a qualitative one, whereas here we are concerned with a quantitative question: What is the exact query complexity of flattening? Nevertheless, we use a similar construction of hard instances as [LZ17] and make use of a variation of their key lemma.

2.1 Simplification: The SDU Problem

We find it convenient to work with a slightly simplified version of the flattening task, having one fewer parameter to worry about.

Definition 2.1 (Statistical distance from uniform (SDU))**.** *We say an algorithm $A^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ is a k-*SDU algorithm *if for all $f : \{0,1\}^n \to \{0,1\}^m$, we have*

- *If (f, τ) is a YES input to* EA*, then $A^f(U_{n'})$ is 2^{-k}-close to $U_{m'}$.*
- *If (f, τ) is a NO input to* EA*, then $\left|\mathrm{Supp}(A^f(U_{n'}))\right| \leq 2^{m'-k}$.*

Note that a k-SDU algorithm is a $(2^{-k}, k/2)$-flattening algorithm (with threshold $\kappa = m' - k/2$). Conversely, we can transform any flattening algorithm to a SDU algorithm using hashing similar to [GSV99a]:

Lemma 2.1. *If there exists a (ε, Δ)-flattening algorithm $A^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ for function $f : \{0,1\}^n \to \{0,1\}^m$ with query complexity q, then there exists a k-SDU algorithm $A^f : \{0,1\}^{n''} \to \{0,1\}^{n''-3k}$ where $n'' = O(n' + m')$ for function $f : \{0,1\}^n \to \{0,1\}^m$ with query complexity q and $k = \Omega(\min\{\Delta, \log(1/\varepsilon)\})$. In particular, there exists such a k-SDU algorithm with query complexity $O(k \cdot \min\{n, m\}^2)$.*

Remark 2.2. *Note that Lemma 2.2 guarantees not only that A is a k-SDU algorithm but also that its output length is only $3k$ bits shorter than its input length. This additional property will be useful in our proof.*

Here for our main result (Theorem 1.1), it suffices to prove an $\Omega(kn^2)$ query lower bound for any k-SDU algorithm $A^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ with $m' = n' - 3k$ and $k \leq n/25$.

Theorem 2.1. *Let $k \leq n$. Every k-SDU algorithm $A^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ for function $f : \{0,1\}^n \to \{0,1\}^m$ has query complexity $\Omega(kn^2)$.*

2.2 Hard Instances

We consider two input distributions $\mathcal{D}_H$ and $\mathcal{D}_L$ over functions $f : \{0,1\}^n \to \{0,1\}^{3n}$ such that the entropies of most functions in $\mathcal{D}_H$ and $\mathcal{D}_L$ are at least $\tau + 1$ and at most $\tau - 1$ (where $\tau = \Theta(n)$), respectively. To sample a function from $\mathcal{D}_H$, we randomly partition the domain of f into many blocks $B_1, B_2, \ldots, B_s$, each of size $2^n/s$ where $s = 2^{3n/4}$. For each block B_i,

- with probability $1/2+\Theta(1/n)$ we insert a high-entropy block: $f|_{B_i}$ will be a uniformly random mapping from B_i to $\{0,1\}^{3n}$; and
- with the remaining probability $1/2 - \Theta(1/n)$, we insert a low-entropy block: all elements of B_i are mapped to the same random element of $\{0,1\}^{3n}$.

The distribution $\mathcal{D}_L$ is the same, except we swap the two $1/2 \pm \Theta(1/n)$ probabilities.

Note that since the range $\{0,1\}^{3n}$ is so much larger than the domain $\{0,1\}^n$, with high probability f will be injective on the high-entropy blocks and will also have no collisions between different blocks. Under this condition, if we let $B(x)$ denote the block containing x (which is determined by $f(x)$) and p be the fraction of high entropy blocks, we have

$$\mathsf{H}_{\text{sh}}\left(f(U_n)\right) = \mathsf{H}_{\text{sh}}\left(B(U_n)\right) + \mathsf{H}_{\text{sh}}\left(f(U_n) \mid B(U_n)\right) \tag{2.1}$$

$$= \log_2 s + p \cdot \log_2\left(\frac{2^n}{s}\right) + (1-p)\cdot 0 = \frac{3n}{4} + p \cdot \frac{n}{4}. \tag{2.2}$$

Under $\mathcal{D}_H$ we have $p = \frac{1}{2} + \Theta(\frac{1}{n})$ whp, and under $\mathcal{D}_L$ we have $p = \frac{1}{2} - \Theta(\frac{1}{n})$ whp, which yields a constant gap in Shannon entropies, as desired.

2.3 Basic Intuition—and a Warning!

The first natural instinct—but too naive, we argue—is that since the bias between observing a high-entropy block versus a low-entropy block is only $\Theta(1/n)$, an anti-concentration bound should imply that distinguishing the two distributions takes $\Omega(n^2)$ queries.

This intuition indeed applies to simple bounded-error randomized decision trees (which output just a 1-bit answer). Concretely, suppose for simplicity that our input is just an n^2-bit string x (instead of an exponentially large oracle f): each bit x_i represents either a high-entropy block ($x_i = 1$) or a low-entropy block ($x_i = 0$). We are given the following *gap-majority* promise: the relative Hamming weight $|x|/n^2$ is either $1/2 + 1/n$ or $1/2 - 1/n$. It is a well-known fact that any bounded-error query algorithm needs $\Omega(n^2)$ queries to distinguish these two cases.

But surprisingly enough, there does exist[2] a flattening/SDU algorithm A^x that solves the *gap-majority* promise problem with only $O(n)$ queries! This suggests that any superlinear lower bound must somehow hide from the algorithm the type (high vs. low) of a queried block. Our choice of distributions $\mathcal{D}_H$ and $\mathcal{D}_L$ does indeed achieve this: since there are so many blocks, a single run of the algorithm is unlikely to query more than one point in a single block, and the marginal distribution of such a single query is the same in both $\mathcal{D}_H$ and $\mathcal{D}_L$. The more precise way in which we exploit the hidden type of a block is in invoking the main result of [LZ17]: when switching a high-entropy block in an f to a low-entropy block, the support of an SDU algorithm's output distribution, $\operatorname{Supp}\left(A^f(U_{n'})\right)$, cannot increase by much.

2.4 Technical Outline

Recall that $A^f(U_{n'})$ is almost-uniform when $f \sim \mathcal{D}_H$ has high entropy. For almost all $z \in \{0,1\}^{m'}$, most of the high-entropy functions f make the algorithm A^f output z (on some random seed):

$$\Pr_{f\sim\mathcal{D}_H}\left[\exists w \in \{0,1\}^{n'}, A^f(w) = z\right] \geq 1 - 2^{-\Omega(k)}. \tag{2.3}$$

On the other hand, since the support of $A^f(U_{n'})$ is small when f has low entropy, there should be many z such that when we sample f from $\mathcal{D}_L$, with high probability $A^f(w)$ does not output z:

$$\Pr_{f\sim\mathcal{D}_L}\left[\exists w \in \{0,1\}^{n'}, A^f(w) = z\right] \leq 2^{-\Omega(k)}. \tag{2.4}$$

To connect the high-entropy and low-entropy cases, we essentially prove that for many $z \in \{0,1\}^{m'}$ and every algorithm A making $o(kn^2)$ queries, we have

$$\Pr_{f\sim\mathcal{D}_H}\left[\exists w \in \{0,1\}^{n'}, A^f(w) = z\right] \leq 2^{o(k)} \cdot \Pr_{f\sim\mathcal{D}_L}\left[\exists w \in \{0,1\}^{n'}, A^f(w) = z\right] + O(2^{-k}). \tag{2.5}$$

As long as there exists z such that Equation (2.3), (2.4) and (2.5), the combination of those equations contradict inequality (2.5).

Our inequality (2.5) is similar to the key lemma of Lovett and Zhang [LZ17] except the inequality is reversed, we have an extra multiplicative factor of $2^{o(k)}$ and our lemma (necessarily) only applies to algorithms making $o(kn^2)$ queries (where the [LZ17] lemma applies even to exponentially many queries).

One key step toward the inequality (2.5) is to reverse the direction of the inequality by the

[2] Consider the following algorithm A^x on input a random seed w: query a sequence of random positions i (according to w) until a position with $x_i = 1$ is found. Output $A^x(w) = i$. It is easy to verify that this is an $(0, \Theta(1/n))$-flattening algorithm with expected query complexity $O(1)$. Repeating the algorithm some $\Theta(n)$ many times yields an $(0, \Omega(1))$-flattening algorithm with expected query complexity $O(n)$. Finally, we can make the algorithm abort if any run exceeds the expected query complexity by a large constant factor; this results in an $(\varepsilon, \Omega(1))$-flattening algorithm of worst-case query complexity $O(n)$.

following trick. We name elements of $\{0,1\}^{n'}$ as $w_1, \ldots, w_{2^{n'}}$ in some arbitrary fixed order. Then

$$\begin{aligned}
&\Pr_f\left[\exists w \in \{0,1\}^{n'}, A^f(w) = z\right] \\
&= \sum_{\ell=1}^{2^{n'}} \Pr_f\left[A^f(w_\ell) = z \text{ and } \nexists w \in \{w_1, \ldots, w_{\ell-1}\}, A^f(w) = z\right] \\
&= \sum_{\ell=1}^{2^{n'}} \left(1 - \Pr_f\left[\exists w \in \{w_1, \ldots, w_{\ell-1}\}, A^f(w) = z \mid A^f(w_\ell) = z\right]\right) \cdot \Pr_f\left[A^f(w_\ell) = z\right].
\end{aligned}$$

Having a negative sign, now we wish to relate the probability of

$$\Pr_f\left[\exists v \in \{w_1, \ldots, w_{\ell-1}\}, A^f(w) = z \mid A^f(w_\ell) = z\right]$$

over $\mathcal{D}_H$ and $\mathcal{D}_L$ in the same direction as [LZ17]. It is not a direct application of their lemma due to the fact that the block size is constant in their construction and our probability is conditioned on the event $A^f(w_\ell) = z$, but we prove a generalization (Lemma A.2) of their lemma that suffices. In fact, the proof we provide in Appendix B is simpler than the one in [LZ17] and yields better parameters.

Like in [LZ17], instead of considering the event $\exists w, A^f(w) = z$ in all the probabilities above, we further impose the restriction that $A^f(w)$ queries each block B_i of the domain at most once, since this event happens with high probability. Furthermore (unlike [LZ17]), we also restrict to the case that the number of high-entropy block queries is in the range $q \cdot \left(1/2 \pm (O(1/n) + O(1/\sqrt{q})\right)$ out of a total of q queries, which also occurs with high probability.

3 The Hard Distribution

Let $A^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ be a potential k-SDU algorithm for functions $f : \{0,1\}^n \to \{0,1\}^m$. Throughout, we will consider a fixed oracle algorithm A^f with query complexity q, and will omit the dependency of A in most notations. For a vector $\vec{X}$, we use $\vec{X}(j)$ to denote the j-th element of $\vec{X}$, and X means the unordered set $\{\vec{X}(j) : j \in [|\vec{X}|]\}$.

It is equivalent to interpret an element $\{0,1\}^n$ as an integer in $[N]$ where $N = 2^n$, since we do not make a use of any structure in $\{0,1\}^n$. Under this notation, we are considering a fixed oracle algorithm $A^f : [N'] \to [M']$ for functions $f : [N] \to [M]$ where $N' = 2^{n'}, M' = 2^{m'}, N = 2^n$ and $M = 2^m$. Actually, we will allow N, M, N' and M' to be arbitrary positive integers, not necessary power of 2.

Partition. Given parameters $s, t \in \mathbb{N}$ where $st = N$, and a function $f : [N] \to [M]$, we will partition the domain $[N]$ into s *blocks* $X_1, \ldots, X_s$ each of size t. We will also fix an order for the blocks and the elements in each block: $\vec{\mathbf{X}} = (\vec{X}_1, \ldots, \vec{X}_s)$. So $\vec{X}_i(j)$ denotes the j-th element of the i-th block. Given a vector $\vec{Y}_i \in [M]^t$, we use the shorthand $f(\vec{X}_i) = \vec{Y}_i$ to mean $f(\vec{X}_i(j)) = \vec{Y}_i(j)$, for all $j \in [t]$. Therefore, once vectors $\vec{Y}_1, \ldots, \vec{Y}_s \in [M]^t$ and a partition $\vec{\mathbf{X}}$ are determined, the function f is fully defined as $f(\vec{X}_i) = \vec{Y}_i$ for all $i \in [s]$.

Distributions.

- Let $\mathcal{X}_s$ be a uniform distribution over an ordered partitions $\mathbf{X} = (\vec{X_1}, \ldots, \vec{X_s})$ of $[N]$ where $|\vec{X_i}| = N/s = t$ for all $i \in [s]$.
- Let $\mathcal{Y}_0$ and $\mathcal{Y}_1$ be distributions on vectors $\vec{\mathbf{Y}} \in [M]^t$ defined as follows,

  For $\mathcal{Y}_0$, uniformly sample an element $y \overset{M}{\leftarrow}$, and output $\vec{Y}(1) = \cdots = \vec{Y}(t) = z$.

  For $\mathcal{Y}_1$, uniformly and independently sample $\vec{Y}(1), \ldots, \vec{Y}(t)$ from $[M]$.
- Given a vector $\vec{b} \in \{0,1\}^s$ and a partition $\mathbf{X} = (\vec{X_1}, \ldots, \vec{X_s})$ of $[N]$, we define the distribution $\mathcal{F}(\mathbf{X}, \vec{b})$ of function $f : [N] \to [M]$ such that $f(\vec{X_i}) = \vec{Y_i}$ where $\vec{Y_i} \leftarrow \mathcal{Y}_{\vec{b}(i)}$. Essentially, $\vec{b}$ indicates whether each block is "high entropy" or "low entropy".
- For $0 \le \alpha \le 1$, let $\mathcal{B}_\alpha$ be a distribution over vectors $\vec{b} \in \{0,1\}^s$, so that each entry of $\vec{b}$ is sampled from $\mathrm{Bern}(\alpha)$ independently.
- For $0 \le \alpha \le 1$, $\mathcal{D}_\alpha$ is a distribution on functions $f : [N] \to [M]$, a partition $\vec{\mathbf{X}}$, and an indicator vector $\vec{b}$, where $(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha$ means that $\vec{b} \sim \mathcal{B}_\alpha$, $\vec{\mathbf{X}} \sim \mathcal{X}_s$ and $f \sim \mathcal{F}(\vec{\mathbf{X}}, \vec{b})$.

Block-Compatibility. When an algorithm A runs with input $w \in [N']$ and oracle $f : [N] \to [M]$, let $\mathsf{Query}_f(w) \subseteq [N]$ be the set of the queries made by the algorithm $A^f(w)$ to f. We say w is *block-compatible* with $(f, \mathbf{X})$ if $|\mathsf{Query}_f(w) \cap X| \le 1$ for all blocks $X \in \mathbf{X}$. The set of block-compatible inputs with $(f, \mathbf{X})$ is denoted

$$\mathsf{BC}(f, \mathbf{X}) = \{w : w \text{ is block-compatible with } (f, \mathbf{X})\}$$

Construction. Set $m = 3n$, so $M = N^3$. Also, set $s = 2^{3n/4} = N^{3/4}$ and $t = 2^{n/4} = N^{1/4}$. Let the high entropy distribution be $\mathcal{D}_H \overset{\text{def}}{=} \mathcal{D}_{1/2+5/n}$ and the low entropy distribution be $\mathcal{D}_L \overset{\text{def}}{=} \mathcal{D}_{1/2+5/n}$. We claim that with high probability, a function f from $\mathcal{D}_H$ and $\mathcal{D}_L$ has entropy at least $\tau + 1$ and at most $\tau - 1$ for $\tau = 7n/8$.

Lemma 3.1. *Let the parameters be as above. Then we have*

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}}) \sim \mathcal{D}_H} [\mathsf{H}_{\text{sh}}(f) \ge \tau + 1] \ge 1 - 2^{-0.9n}$$
$$\Pr_{(f,\vec{b},\vec{\mathbf{X}}) \sim \mathcal{D}_L} [\mathsf{H}_{\text{sh}}(f) \le \tau - 1] \ge 1 - 2^{-0.9n}$$

Proof. For any pair of independent and random mappings to M, the collision probability is $1/M$. There are no more than N^2 pairs of inputs, so with probability at least $1 - N^2/M = 1 - 2^{-n}$, there is no collision when two images are sampled independently. Under that condition, as shown by Equation (2.1), let p be the fraction of high entropy blocks, namely p is the hamming weight of $\vec{b}$ divided by s, the entropy of the function f is

$$\mathsf{H}_{\text{sh}}(f(U_n)) = \frac{3n}{4} + p \cdot \frac{n}{4}.$$

Recall that when we sample $\vec{b}$ from $\mathcal{D}_H$, $\vec{b}(i) \sim \text{Bern}(1/2 + 5/n)$ for all $i \in [s]$. By the Chernoff bound,

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[p \geq \frac{1}{2} + \frac{4}{n}\right] \geq 1 - \frac{1}{4}2^{s\cdot(1/n)^2},$$

which implies

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\mathsf{H}_{\text{sh}}(f) \geq \frac{3n}{4} + \left(\frac{1}{2} + \frac{4}{n}\right)\cdot\frac{n}{4} = \frac{7n}{8} + 1\right] \geq 1 - 2^{-\frac{1}{4}\cdot s\cdot(1/n)^2} - 2^{-n} = 1 - 2^{-0.9n}.$$

Similarly, when sampling from $\mathcal{D}_L$,

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}\left[\mathsf{H}_{\text{sh}}(f) \leq \frac{3n}{4} + \left(\frac{1}{2} - \frac{4}{n}\right)\cdot\frac{n}{4} = \frac{7n}{8} - 1\right] \geq 1 - 2^{-\frac{1}{4}\cdot s\cdot(1/n)^2} - 2^{-n} = 1 - 2^{-0.9n}.$$

Taking $\tau = \frac{7n}{8}$ concludes the lemma. □

4 Query Lower Bound for SDU Algorithms

4.1 Proof Strategy

Let A^f be a k-SDU algorithm making q queries. We may assume wlog that the algorithm makes exact q oracle queries to f, and all the query positions are distinct. (It is useless to query the same positions, and if the number of queries is less than q, we simply make some dummy queries.) We derive the lower bound (Theorem 2.1) from the following two lemmas.

Lemma 4.1. *Let A^f be a k-SDU algorithm making q queries. For every $n > 25k$ and $z \in [M']$ that satisfies*

$$\mathop{\mathbb{E}}_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\left|\{w : A^f(w) = z\}\right|\right] \leq 2^{4k}, \tag{4.1}$$

we have

$$\begin{aligned}&\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right]\\ \leq\;& 2^{O\left(\frac{q}{n^2}+\sqrt{\frac{kq}{n^2}}\right)}\cdot\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] + O(2^{-k})\end{aligned} \tag{4.2}$$

Lemma 4.2. *There exists a universal constant $c > 0$ such that for every sufficiently large n and $k \leq n$, there is an output $z \in [M']$ that satisfies*

1. $\displaystyle\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \geq 1 - 2^{-ck} \geq \frac{1}{2}.$

2. $\displaystyle\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \leq 2^{-ck}.$

3. $\displaystyle\mathop{\mathbb{E}}_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\left|\{w : A^f(w) = z\}\right|\right] \leq 2^{4k}$

Theorem 2.1 follows by plugging z that satisfies the inequalities in Lemma 4.2 into Inequality (4.2). If $q = o(kn^2)$, then the exponent is $o(k)$, which yields a contradiction.

In the following section, we prove that most inputs are block-compatible and hence we can only consider the block-compatible inputs rather than the whole domain $[N']$. Then we prove Lemma 4.1 and 4.2 in Section 4.3 and 4.4, respectively.

4.2 Block-Compatible Inputs

As in [LZ17], we only consider block-compatible inputs, where each block is queried at most once. In that case, it is easier to compare the behavior of the SDU algorithms. Since there are exponentially many blocks but only polynomially many queries, intuitively, the probability of having block-compatible property is overwhelming if we randomly partition the domain of f. Formally,

Lemma 4.3. *For every $w \in [N']$ and $\alpha \in [0,1]$,*

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}[w \notin \mathsf{BC}(f,\mathbf{X})] \le \frac{q^2}{s} \le 2^{-0.6n}.$$

Proof. In order to handle adaptive algorithms, we consider the following procedure to sample $(f,\vec{b},\vec{\mathbf{X}})$, which is equivalent to sampling from $\mathcal{D}_\alpha$. Specifically, we sample the parts that are related to w first.

Procedure 4.1

1. Initially, $\vec{X}_i(j) = *$ and $\vec{b}(i) = *$ for all $i \in [s], j \in [t]$.
2. Simulate $A^f(w)$ handling the r-th oracle query x_r as follows. For $r = 1,\ldots,q$,
   (a) Based on previous queries and results as well as w, let the r-th query be x_r. Select (i,j) uniformly at random from $[s]\times[t]$ subject to $X_i(j) = *$ and assign $\vec{X}_i(j) = x_r$.
   (b) If $\vec{b}(i) = *$, then assign $\vec{b}(i) \sim \mathrm{Bern}(\alpha)$ and $\vec{Y}_i \sim \mathcal{Y}_{\vec{b}(i)}$.
   (c) Set $f(x_r) = Y_i(j)$ and return $f(x_r)$ as the answer to the query.
3. Assign the rest of the vectors $\vec{\mathbf{X}}$ and $\vec{b}$ by executing Step 2(a)–2(c) for all $x \in [N] \setminus \{x_1,\ldots,x_q\}$.

By the principle of deferred decisions, it can be verified that the joint distribution of $(f,\vec{\mathbf{X}},\vec{b})$ is identical to $\mathcal{D}_\alpha$.

Notice that $w \in \mathsf{BC}(f,\vec{\mathbf{X}},\vec{b})$ if and only if the sequence of q values of i selected in Step 2(a) are all distinct. The probability that the $(r+1)^{\text{st}}$ value of i is the same one comparing to the previous r values is at most $rt/(st-r) \le q/s$, since $r \le q-1$ and $qr \le st$. So the probability that there are any repetitions is at most q^2/s. $\square$

By Markov's inequality, almost all inputs are block-compatible.

Corollary 4.1. *For every $\alpha \in [0,1]$,*

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}\left[|\mathsf{BC}(f,\mathbf{X})| > N'\cdot(1-2^{-0.3n})\right] \ge 1-2^{-0.3n}$$

4.3 Proof of Lemma 4.1

Lemma 4.1. *Let A^f be a k-SDU algorithm making q queries. For every $n > 25k$ and $z \in [M']$ that satisfies*

$$\mathop{\mathbb{E}}_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\left|\{w : A^f(w) = z\}\right|\right] \le 2^{4k}, \tag{4.1}$$

we have

$$\begin{aligned}&\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right]\\ \le\quad & 2^{O\left(\frac{q}{n^2}+\sqrt{\frac{kq}{n^2}}\right)} \cdot \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] + O(2^{-k})\end{aligned} \tag{4.2}$$

Proof. Define the set

$$W_z(f,\mathbf{X}) = \{w : w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\}.$$

Let $w_1, \cdots, w_{N'}$ be all possible inputs sorted in arbitrary but fixed order. The first step is to break the event $\exists w \in W_z(f,\mathbf{X})$ to the events that w_ℓ is the "first" one in $W_z(f,\mathbf{X})$ for all $\ell \in [N']$.

$$\begin{aligned}&\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}\left[\exists w \in W_z(f,\mathbf{X})\right]\\ =\ & \sum_{\ell=1}^{N'} \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}\left[w_\ell \in W_z(f,\mathbf{X}) \wedge w_1,\dots,w_{\ell-1} \notin W_z(f,\mathbf{X})\right]\\ =\ & \sum_{\ell=1}^{N'} \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}\left[w_1,\dots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \mid w_\ell \in W_z(f,\mathbf{X})\right] \cdot \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}\left[w_\ell \in W_z(f,\mathbf{X})\right]\end{aligned}$$

Our goal is to switch the distribution from $\mathcal{D}_H$ to $\mathcal{D}_L$ and see how the probability changes. We do the switch using the following two claims.

Claim 4.1. *For every $w_\ell \in [N']$, $\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}\left[w_\ell \in W_z(f,\mathbf{X})\right]$ does not depend on $\alpha \in [0,1]$. In particular,*

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[w_\ell \in W_z(f,\mathbf{X})\right] = \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}\left[w_\ell \in W_z(f,\mathbf{X})\right].$$

Claim 4.2. *For every $w_\ell \in [N']$ and $z \in [M']$,*

$$\begin{aligned}&\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[w_1,\dots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \mid w_\ell \in W_z(f,\mathbf{X})\right]\\ \le\ & 2^{O\left(\frac{q}{n^2}+\sqrt{\frac{kq}{n^2}}\right)} \cdot \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}\left[w_1,\dots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \mid w_\ell \in W_z(f,\mathbf{X})\right] + O\left(\frac{q^2}{s}\right) + 2^{-5k}\end{aligned}$$

The intuition behind Claim 4.1 is that as long as w_ℓ is block-compatible, the query results are independently uniform over $[M]$ in both $\mathcal{D}_H$ or $\mathcal{D}_L$ case. Note that unlike Lemma 4.1, Claim 4.2 refers to *non*-membership in $W_z(f,\mathbf{X})$, which allows us to use the main lemma of Lovett and Zhang [LZ17], which provides an inequality in the opposite direction of Lemma 4.1. The formal proofs of these Claims are given in Section 4.3.1 and 4.3.2.

Once we have the above claims, we can prove the lemma:

$$\begin{aligned}
&\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}[\exists w \in W_z(f,\mathbf{X})] \\
&\leq 2^{O\left(\frac{q}{n^2}+\sqrt{\frac{kq}{n^2}}\right)} \cdot \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}[\exists w \in W_z(f,\mathbf{X})] + \left(O\left(\frac{q^2}{s}\right)+2^{-5k}\right)\cdot \sum_{\ell=1}^{2^{n'}} \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}[w_\ell \in W_z(f,\mathbf{X})] \\
&\leq 2^{O\left(\frac{q}{n^2}+\sqrt{\frac{kq}{n^2}}\right)} \cdot \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}[\exists w \in W_z(f,\mathbf{X})] + \left(O\left(2^{-n/5}\right)+2^{-5k}\right)\cdot \mathop{\mathbb{E}}_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\left|\{w : A^f(w)=z\}\right|\right] \\
&\leq 2^{O\left(\frac{q}{n^2}+\sqrt{\frac{kq}{n^2}}\right)} \cdot \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}[\exists w \in W_z(f,\mathbf{X})] + O(2^{-k}).
\end{aligned}$$

The second inequality is by the assumption of $n > 25k$, and the last inequality is by Inequality (4.1). □

4.3.1 Proof of Claim 4.1

Claim 4.1. *For every $w_\ell \in [N']$, $\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}[w_\ell \in W_z(f,\mathbf{X})]$ does not depend on $\alpha \in [0,1]$. In particular,*

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}[w_\ell \in W_z(f,\mathbf{X})] = \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}[w_\ell \in W_z(f,\mathbf{X})].$$

Proof. We factorize the probability into two parts and prove both of them are independent of α.

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}[w_\ell \in W_z(f,\mathbf{X})] = \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}\left[A^f(w_\ell)=z \mid w_\ell \in \mathsf{BC}(f,\mathbf{X})\right] \cdot \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}[w_\ell \in \mathsf{BC}(f,\mathbf{X})]$$

We use Procedure 4.1 to sample $(f,\vec{b},\vec{\mathbf{X}})$. We will prove the second factor is independent of α by induction over r. Conditioning on the first $(r-1)$ values of i selected in Step 2(a) being all distinct, that is, the block-compatible property has not been violated in the first r rounds, we have $\vec{b}(i) = *$ at the beginning of Step 2(b) in the r-th round. Thus no matter what α is and what $\vec{b}(i)$ is assigned, $Y_i(j)$ is uniform over $[M]$ in the r-th round. Therefore, under the assumed condition, the distribution of x_r and $f(x_r)$ are independent of α and the probability of maintaining the block-compatible property in the r-th round is independent of α. By induction, we know that the probability of maintaining the block-compatible property in all q rounds is independent of α.

For the first factor, as discussed above, conditioning on the block-compatible property, the distributions of x_r and $f(x_r)$ are independent of α, so the probability of getting z as the output of $A^f(w_\ell)$ is also independent of α. □

4.3.2 Proof of Claim 4.2

Claim 4.2. *For every $w_\ell \in [N']$ and $z \in [M']$,*

$$\begin{aligned}
&\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}[w_1,\ldots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \mid w_\ell \in W_z(f,\mathbf{X})] \\
&\leq 2^{O\left(\frac{q}{n^2}+\sqrt{\frac{kq}{n^2}}\right)} \cdot \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}[w_1,\ldots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \mid w_\ell \in W_z(f,\mathbf{X})] + O\left(\frac{q^2}{s}\right) + 2^{-5k}
\end{aligned}$$

Proof. We consider the following sampling procedure which is equivalent to sampling $(f, \vec{b}, \vec{\mathbf{X}})$ from $\mathcal{D}_\alpha$ conditioned on $w_\ell \in W_z(f, \mathbf{X})$ (Namely, $A^f(w_\ell) = z$ and $w_\ell \in \mathsf{BC}(f, \mathbf{X})$). We denote such a distribution as $(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha(w_\ell, z)$. It follows the same idea as in Procedure 4.1 — sampling the blocks that are queried by $A^f(w_\ell)$ first, and uses the rejection sampling to handle the condition $w_\ell \in W_z(f, \mathbf{X})$.

Procedure 4.2

1. Initially, $\vec{X}_i(j) = *$ and $\vec{b}(i) = *$ for all $i \in [s], j \in [t]$ and $f(x) = *$ for all $x \in [N]$.
2. Simulate $A^f(w_\ell)$ handling the r-th oracle query x_r as follows. For $r = 1 \dots, q$,
   (a) Based on previous queries and results as well as w, let the r-th query be x_r. Select (i, j) uniformly at random from $[s] \times [t]$ subject to $X_i(j) = *$ and assign $\vec{X}_i(j) = x_r$.
   (b) If $\vec{b}(i) = *$, then assign $\vec{b}(i) \sim \text{Bern}(\alpha)$ and $\vec{Y}_i \sim \mathcal{Y}_{\vec{b}(i)}$.
   (c) Set $f(x_r) = Y_i(j)$ and return $f(x_r)$ as the answer to the query.
3. If q values of i in Step 2(a) are not all distinct, or $A^f(w_\ell) \neq z$, **restart**.
4. For all (i, j) such that $\vec{b}(i) \neq *$ and $\vec{X}_i(j) = *$, randomly sample $x \in [N]$ that has not been assigned to any partition. Set $\vec{X}_i(j) = x$ and $f(x) = Y_i(j)$.
5. Denote the partially assigned (some of them are mapped to $*$) function and vectors sampled so far as $f^*, \vec{b}^*, (\vec{\mathbf{X}}^*) \sim \mathcal{D}^*_\alpha(w_\ell, z)$.
6. Assign the rest of the vectors $\vec{\mathbf{X}}$, $\vec{b}$ and the mapping f by executing Step 2(a)–(c) for all $x \in [N] \setminus \{x_1, \dots, x_q\}$ (instead of x_r).

Notice that until Step 5, information (including the partition $\vec{\mathbf{X}}^*$, function mapping f^* and the indicator $\vec{b}^*$) on exactly q blocks is decided.

The probability we consider then can be written as

$$\begin{aligned}
&\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}[w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid w_\ell \in W_z(f, \mathbf{X})] \\
=& \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha(w_\ell,z)}[w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X})] \\
=& \sum_{(f^*,\vec{b}^*,\vec{\mathbf{X}}^*)} \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha(w_\ell,z)}\left[w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*)\right] \times \Pr_{\mathcal{D}^*_\alpha(w_\ell,z)}\left[(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)\right]
\end{aligned}$$

Now we introduce a property of a partial indicator. We say a partial indicator is balanced if the number of zeros (low entropy block) and ones (high entropy block) are about the same.

Definition 4.2 (Balance). *Let $\vec{b}^* \in \{0, 1, *\}^s$ be a "partial" indicator vector where there are q non-star entries. We say it is* balanced *if the number of 1s is in $[q \cdot (1/2 - 5/n - \sqrt{25k/q}), q \cdot (1/2 + 5/n + \sqrt{25k/q})]$.*

According to Procedure 4.2, each non-star entry of $\vec{b}^*$ is sampled uniformly and independently from Bern(α). When $\alpha \in [1/2 - 5/n, 1/2 + 5/n]$, by Chernoff bound, we have

$$\Pr_{f^*,\vec{b}^*,(\vec{\mathbf{X}}^*)\sim\mathcal{D}_\alpha^*(w_\ell,z)}\left[\vec{b}^* \text{ is balanced}\right] \geq 1 - 2^{-5k}.$$

And thus we can sum over only balanced $\vec{b}^*$ by paying an additive term.

$$\begin{aligned}
&\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha}[w_1,\ldots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \mid w_\ell \in W_z(f,\mathbf{X})] \\
\leq\ & 2^{-5k} + \sum_{\substack{(f^*,\vec{b}^*,\vec{\mathbf{X}}^*) \\ \text{where } \vec{b}^* \text{ is balanced}}} \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha(w_\ell,z)}\left[w_1,\ldots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \mid (f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] \\
&\times \Pr_{\mathcal{D}_\alpha(w_\ell,z)^*}\left[(f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] \qquad (4.3)
\end{aligned}$$

Now we use the following two claims (proved in the later paragraphs) to connect the high entropy case ($\mathcal{D}_H$) and the low entropy case ($\mathcal{D}_L$) on those two factors.

Claim 4.3. *For every $w_\ell \in [N']$, $z \in [M']$ and every possible $(f^*,\vec{b}^*,\vec{\mathbf{X}}^*)$ from $\mathcal{D}_H^*(w_\ell,z)$, we have*

$$\begin{aligned}
&\Pr_{\mathcal{D}_H(w_\ell,z)}\left[w_1,\ldots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \,\middle|\, (f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] \\
&\quad\leq \Pr_{\mathcal{D}_L(w_\ell,z)}\left[w_1,\ldots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \,\middle|\, (f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] + O\left(\frac{q^2}{s}\right)
\end{aligned} \qquad (4.4)$$

Claim 4.4. *For every $w_\ell \in [N']$, $z \in [M']$ and every $(f^*,\vec{b}^*,\vec{\mathbf{X}}^*)$ where $\vec{b}^*$ is balanced,*

$$\Pr_{\mathcal{D}_H^*(w_\ell,z)}\left[(f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] \leq 2^{O\left(\frac{q}{n^2}+\sqrt{\frac{kq}{n^2}}\right)} \cdot \Pr_{\mathcal{D}_L^*(w_\ell,z)}\left[(f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] \qquad (4.5)$$

Inserting Inequalities (4.4) and (4.5) to Equation (4.3) with $\alpha = 1/2 + 5/n$, we conclude the claim.

□

Proof of Claim 4.3

Claim 4.3. *For every $w_\ell \in [N']$, $z \in [M']$ and every possible $(f^*,\vec{b}^*,\vec{\mathbf{X}}^*)$ from $\mathcal{D}_H^*(w_\ell,z)$, we have*

$$\begin{aligned}
&\Pr_{\mathcal{D}_H(w_\ell,z)}\left[w_1,\ldots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \,\middle|\, (f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] \\
&\quad\leq \Pr_{\mathcal{D}_L(w_\ell,z)}\left[w_1,\ldots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \,\middle|\, (f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] + O\left(\frac{q^2}{s}\right)
\end{aligned} \qquad (4.4)$$

Proof. We will use a variation of the main lemma (Lemma 3) in [LZ17].

Lemma 4.4. *Let $\hat{A}^{\hat{f}} : [\hat{N}'] \to [\hat{M}']$ be an algorithm making at most q oracle queries to $\hat{f} : [\hat{N}] \to [\hat{M}]$. Let $\hat{\mathcal{D}}_H = \hat{\mathcal{D}}_{1/2+5/n}$ and $\hat{\mathcal{D}}_L = \hat{\mathcal{D}}_{1/2-5/n}$ be the distribution over a function $\hat{f} : [\hat{N}] \to [\hat{M}]$, a partition $\vec{\hat{\mathbf{X}}} \in ([\hat{N}]^{\hat{t}})^{\hat{s}}$, and an indication vector $\vec{\hat{b}} \in \{0,1\}^{\hat{s}}$ defined in Section 3. Then for all $z \in [\hat{N}']$,*

$$\Pr_{(\hat{f},\vec{\hat{b}},\vec{\hat{\mathbf{X}}})\sim\hat{\mathcal{D}}_L}\left[\exists w \in \mathsf{BC}(\hat{f},\hat{\mathbf{X}}), \hat{A}^{\hat{f}}(w) = z\right] - \Pr_{(\hat{f},\vec{\hat{b}},\vec{\hat{\mathbf{X}}})\sim\hat{\mathcal{D}}_H}\left[\exists w \in \mathsf{BC}(\hat{f},\hat{\mathbf{X}}), \hat{A}^{\hat{f}}(w) = z\right] \leq \frac{O(q^2)}{\hat{s}}$$

Proof. See Appendix A.1. □

For a fixed $(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)$, apply the above lemma in the following way:

- Let $\hat{s} = s - q$, $\hat{t} = t$, and so $\hat{N} = \hat{s} \cdot \hat{t} = N - qt$.
- Let $S = \{x \mid f^*(x) = *\} \subseteq [N]$, $I = \{i \mid \vec{b}^*(i) = *\} \subseteq [s]$ and $\pi_X : S \to [\hat{N}]$, $\pi_I : I \to [\hat{s}]$ be arbitrary bijection mappings. Then we define $\hat{f}, \vec{\hat{X}}$ and $\vec{\hat{b}}$ as follows.

$$\begin{cases} \forall\, \hat{x} \in [\hat{N}] & , \quad \hat{f}(\hat{x}) \overset{\text{def}}{=} f(\pi_X^{-1}(\hat{x})) \\ \forall\, (\hat{i},\hat{j}) \in [\hat{s}] \times [\hat{t}] & , \quad \vec{\hat{X}}_{\hat{i}}(\hat{j}) \overset{\text{def}}{=} \pi_X(\vec{X}_{\pi_I^{-1}(\hat{i})}(\hat{j})) \\ \forall\, \hat{i} \in [\hat{s}] & , \quad \vec{\hat{b}}(\hat{i}) \overset{\text{def}}{=} \vec{b}(\pi_I^{-1}(\hat{i})) \end{cases}.$$

- For $\hat{w} \in [\hat{N}]$, define $\hat{A}^{\hat{f}}(\hat{w})$ to simulate $A^f(w)$ and $w \in \{w_1, \ldots, w_{\ell-1}\}$ in the following way. It first check that if $w \notin \{w_1, \ldots, w_{\ell-1}\}$, output something not equal to z. Otherwise simulate $A^f(w)$ and when A makes a query $x \in \mathbf{X}^*$, $\hat{A}$ hardwire the result $f(x)$ as the answer. When $x \in S$, return $\hat{f}(\pi_X(x))$ as the answer.

By the above mapping, we have

$$\begin{aligned}\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_\alpha(w_\ell,z)}&\left[\exists w \in \mathsf{BC}(f,\mathbf{X}) \cap \{w_1,\ldots,w_{\ell-1}\}, A^f(w) = z \,\middle|\, (f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] \\ &= \Pr_{(\hat{f},\vec{\hat{b}},\vec{\hat{\mathbf{X}}})\sim\hat{\mathcal{D}}_\alpha}\left[\exists w \in \mathsf{BC}(\hat{f},\hat{\mathbf{X}}), \hat{A}^{\hat{f}}(w) = z\right].\end{aligned}$$

By Lemma 4.4,

$$\begin{aligned}&\Pr_{\mathcal{D}_H(w_\ell,z)}\left[w_1,\ldots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \,\middle|\, (f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] \\ &= 1 - \Pr_{\mathcal{D}_H(w_\ell,z)}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}) \cap \{w_1,\ldots,w_{\ell-1}\}, A^f(w) = z \,\middle|\, (f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] \\ &= 1 - \Pr_{(\hat{f},\vec{\hat{b}},\vec{\hat{\mathbf{X}}})\sim\hat{\mathcal{D}}_H}\left[\exists w \in \mathsf{BC}(\hat{f},\hat{\mathbf{X}}), \hat{A}^{\hat{f}}(w) = z\right] \\ &\leq 1 - \Pr_{(\hat{f},\vec{\hat{b}},\vec{\hat{\mathbf{X}}})\sim\hat{\mathcal{D}}_L}\left[\exists w \in \mathsf{BC}(\hat{f},\hat{\mathbf{X}}), \hat{A}^{\hat{f}}(w) = z\right] + O\left(\frac{q^2}{\hat{s}}\right) \\ &= 1 - \Pr_{\mathcal{D}_L(w_\ell,z)}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}) \cap \{w_1,\ldots,w_{\ell-1}\}, A^f(w) = z \,\middle|\, (f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] + O\left(\frac{q^2}{s}\right) \\ &= \Pr_{\mathcal{D}_L(w_\ell,z)}\left[w_1,\ldots,w_{\ell-1} \notin W_z(f,\mathbf{X}) \,\middle|\, (f^*,\vec{b}^*,\vec{\mathbf{X}}^*)\right] + O\left(\frac{q^2}{s}\right).\end{aligned}$$

□

Proof of Claim 4.4

Claim 4.4. *For every $w_\ell \in [N']$, $z \in [M']$ and every $(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)$ where $\vec{b}^*$ is balanced,*

$$\Pr_{\mathcal{D}_H^*(w_\ell,z)}\left[(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)\right] \leq 2^{O\left(\frac{q}{n^2}+\sqrt{\frac{kq}{n^2}}\right)} \cdot \Pr_{\mathcal{D}_L^*(w_\ell,z)}\left[(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)\right] \tag{4.5}$$

Proof. The only difference between $\mathcal{D}_L(w_\ell, z)$ and $\mathcal{D}_H(w_\ell, z)$ is when sampling $\vec{b}^*$. Recall that a balanced partial indicator means the hamming weight is within the range $q \cdot \left(1/2 \pm \left(1/n + \sqrt{25k/q}\right)\right)$. Since we only consider the cases where $\vec{b}^*$ is balanced, the ratio can be bounded as follows.

$$\begin{aligned}\frac{\Pr_{\mathcal{D}_H^*(w_\ell,z)}\left[(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)\right]}{\Pr_{\mathcal{D}_L^*(w_\ell,z)}\left[(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)\right]} &\leq \left(\frac{\frac{1}{2}+\frac{5}{n}}{\frac{1}{2}-\frac{5}{n}}\right)^{q\left(\frac{1}{2}+\left(\frac{1}{n}+\sqrt{\frac{25k}{q}}\right)\right)} \left(\frac{\frac{1}{2}-\frac{5}{n}}{\frac{1}{2}+\frac{5}{n}}\right)^{q\left(\frac{1}{2}-\left(\frac{1}{n}+\sqrt{\frac{25k}{q}}\right)\right)} \\ &\leq \left(1+\frac{10}{n}\right)^{2q\left(\frac{1}{n}+\sqrt{\frac{25k}{q}}\right)} \left(1-\frac{10}{n}\right)^{-2q\left(\frac{1}{n}+\sqrt{\frac{25k}{q}}\right)} \\ &\leq 2^{O\left(\frac{q}{n^2}+\sqrt{\frac{kq}{n^2}}\right)}\end{aligned} \tag{4.6}$$

□

4.4 Proof of Lemma 4.2

Lemma 4.2. *There exists a universal constant $c > 0$ such that for every sufficiently large n and $k \leq n$, there is an output $z \in [M']$ that satisfies*

1. $\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\exists w \in \mathsf{BC}(f, \mathbf{X}), A^f(w) = z\right] \geq 1 - 2^{-ck} \geq \frac{1}{2}$.
2. $\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}\left[\exists w \in \mathsf{BC}(f, \mathbf{X}), A^f(w) = z\right] \leq 2^{-ck}$.
3. $\mathbb{E}_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\left|\{w : A^f(w) = z\}\right|\right] \leq 2^{4k}$

Proof. In this proof, we abuse notation by denoting $\mathsf{BC}(f, \mathbf{X})$ also to be the uniform distribution over the set $\mathsf{BC}(f, \mathbf{X})$. We will show that that for a random z sampled from $[M']$, it satisfies each property with probability at least $1 - 2^{-\Omega(k)}$, and hence by the union bound, it satisfies all three properties with probability at least $1 - 2^{-\Omega(k)}$. In particular, there exists $z \in [M']$ satisfying all three conditions simultaneously.

1.

$$\begin{aligned}\Pr_{z\sim\{0,1\}^{m'}}\left[z \notin A^f(\mathsf{BC}(f, \mathbf{X})\right] &= 1 - \frac{\left|\mathrm{Supp}(A^f(\mathsf{BC}(f, \mathbf{X})))\right|}{[M']} \\ &\leq d_{\mathrm{TV}}\left(A^f(\mathsf{BC}(f, \mathbf{X})), U_{m'}\right) \\ &\leq d_{\mathrm{TV}}\left(A^f(U_{n'}), U_{m'}\right) + d_{\mathrm{TV}}\left(\mathsf{BC}(f, \mathbf{X}), U_{m'}\right) \\ &= d_{\mathrm{TV}}\left(A^f(U_{n'}), U_{m'}\right) + 1 - \frac{|\mathsf{BC}(f, X)|}{[N']}\end{aligned} \tag{4.7}$$

Take the expectation over $(f, \vec{b}, \vec{\mathbf{X}})$ from $\mathcal{D}_H$ for Equation (4.7). By Lemma 3.1, Definition 2.1 and Corollary 4.1 we have

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H, z\sim[M']}\left[z \notin A^f(\mathsf{BC}(f,\mathbf{X}))\right] \le \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\mathsf{H}_{\text{sh}}(f) < \tau+1\right] + 2^{-k} + 2^{-0.3n} \tag{4.8}$$

$$\le 2^{-0.9n} + 2^{-k} + 2^{-0.3n} \le 2^{-0.2k} \tag{4.9}$$

By the Markov inequality,

$$\Pr_{z\in[M']}\left[\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \ge 1 - 2^{-0.1k}\right] \ge 1 - 2^{-0.1k}.$$

2. By Lemma 3.1 and Definition 2.1, we have

$$\begin{aligned} & \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L, z\sim[M']}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \\ \le\ & \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L, z\sim[M']}\left[\exists w \in [N'], A^f(w) = z\right] \\ \le\ & \Pr_{z\sim[M']}\left[\exists w \in [N'], A^f(w) = z \mid \mathsf{H}_{\text{sh}}(f) \le \tau - 1\right] + \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}\left[\mathsf{H}_{\text{sh}}(f) > \tau - 1\right] \\ \le\ & 2^{-k} + 2^{-0.9n} \le 2^{-0.8k}. \end{aligned}$$

By the Markov inequality,

$$\Pr_{z\in[M']}\left[\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \le 2^{-0.1k}\right] \ge 1 - 2^{-0.7k}.$$

3. Since $m' = n' + 3k$,

$$\mathop{\mathbb{E}}_{z\in[M']}\left[\left|\{w : A^f(w) = z\}\right|\right] = \Pr_{z\in[M']}\left[\sum_{w\in[N']} I(A^f(w) = z)\right] = 2^{n'} \cdot 2^{-m'} = 2^{3k}.$$

In particular,

$$\mathop{\mathbb{E}}_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H, z\in[M']}\left[\left|\{w : A^f(w) = z\}\right|\right] = 2^{3k}.$$

By the Markov inequality,

$$\Pr_{z\sim[M']}\left[\mathop{\mathbb{E}}_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\left|\{w : A^f(w) = z\}\right|\right] \le 2^{4k}\right] \ge 1 - 2^{-k}.$$

□

5 Acknowledgement

S.V. thanks Iftach Haitner, Or Meir, and Colin Zheng for many illuminating discussions about this problem in the past.

References

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[GGKT05] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 25–32, 1989.

[GSV99a] Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Can statistical zero knowledge be made non-interactive? or on the relationship of SZK and NISZK. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 467–484, 1999.

[GSV99b] Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Can statistical zero knowledge be made non-interactive? or on the relationship of SZK and NISZK. *Electronic Colloquium on Computational Complexity (ECCC)*, 6(13), 1999.

[HHR06] Iftach Haitner, Danny Harnik, and Omer Reingold. Efficient pseudorandom generators from exponentially hard one-way functions. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 228–239, 2006.

[HHR$^+$10] Iftach Haitner, Thomas Holenstein, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 616–637, 2010.

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

[HNO$^+$09] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil P. Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM J. Comput.*, 39(3):1153–1218, 2009.

[Hol06] Thomas Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 443–461, 2006.

[HR11] Thomas Holenstein and Renato Renner. On the randomness of independent experiments. *IEEE Trans. Information Theory*, 57(4):1865–1871, 2011.

[HRV10] Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 437–446, 2010.

[HRV13] Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. *SIAM J. Comput.*, 42(3):1405–1430, 2013.

[HRVW09] Iftach Haitner, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Inaccessible entropy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 611–620, 2009.

[HS12] Thomas Holenstein and Makrand Sinha. Constructing a pseudorandom generator requires an almost linear number of calls. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 698–707. IEEE, 2012.

[KK05] Jonathan Katz and Chiu-Yuen Koo. On constructing universal one-way hash functions from arbitrary one-way functions. *IACR Cryptology ePrint Archive*, 2005:328, 2005.

[LZ17] Shachar Lovett and Jiapeng Zhang. On the impossibility of entropy reversal, and its application to zero-knowledge proofs. In *Theory of Cryptography Conference*, pages 31–55. Springer, 2017.

[NV06] Minh-Huyen Nguyen and Salil P. Vadhan. Zero knowledge with efficient provers. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 287–295, 2006.

[Oka00] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *J. Comput. Syst. Sci.*, 60(1):47–108, 2000.

[OV08] Shien Jin Ong and Salil P. Vadhan. An equivalence between zero knowledge and commitments. In *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008.*, pages 482–500, 2008.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 387–394, 1990.

[RW04] Renato Renner and Stefan Wolf. Smooth rényi entropy and applications. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, page 233. IEEE, 2004.

[SV97] Amit Sahai and Salil P. Vadhan. A complete promise problem for statistical zero-knowledge. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 448–457, 1997.

[Vad99] Salil Pravin Vadhan. *A study of statistical zero-knowledge proofs.* PhD thesis, Citeseer, 1999.

[VZ12] Salil P. Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 817–836, 2012.

A Missing Proofs

A.1 Proof of Lemma 4.4

We restate the lemma as follows. Note that it is unnecessarily that $N = 2^n$ or being a power of two (and similarly for M, N' and M').

Lemma A.1. *Let $A^f : [N'] \to [M']$ be an algorithm making at most q oracle queries to $f : [N] \to [M]$. Let $\mathcal{D}_H = \mathcal{D}_{1/2+5/n}$ and $\mathcal{D}_L = \mathcal{D}_{1/2-5/n}$ be the distribution over a function $f : [N] \to [M]$, a partition $\vec{\mathbf{X}} \in ([N]^t)^s$, and the indication vector $\vec{b} \in \{0,1\}^s$ as defined in Section 3. Then for all $z \in [N]$,*

$$\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_L}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] - \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_H}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \le \frac{O(q^2)}{s}.$$

Besides the parameters difference, a key difference between Lemma A.1 and the key lemma in [LZ17] is that in our construction, the indicator vectors b consist of s independent Bernoulli random variables, while in their case, the number of ones, namely the Hamming weight is fixed. Formally, they consider the following distribution.

Definition A.1. *For $i \in [s]$, $\tilde{\mathcal{D}}_i$ is the distribution over functions $f : [N] \to [M]$ and partitions $\vec{\mathbf{X}}$ defined as follows. Let $\vec{b_i} = (\underbrace{1,\ldots,1}_{i},\underbrace{0,\ldots,0}_{s-i})$. Then $(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_i$ denotes that $\vec{\mathbf{X}} \sim \mathcal{X}_s$ and $f \sim \mathcal{F}(\vec{\mathbf{X}}, \vec{b_i})$.*

A more direct analogue of the key lemma in [LZ17] (with improved parameters and a simplified proof) can be stated using our notation:

Lemma A.2. *Let $A^f : [N'] \to [M']$ be an algorithm, which makes at most q queries to its oracle $f : [N] \to [M]$. If $2q^2 < i$, then for all $z \in \{0,1\}^{m'}$,*

$$\Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_{i-1}}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] - \Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_i}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \le \frac{O(q^2)}{i^2}.$$

We provide a simpler proof of Lemma A.2 below. First we prove Lemma A.1 using Lemma A.2.

Proof of Lemma A.1. By telescoping over i in Lemma A.2, we get that for $\frac{1}{4} \le \alpha < \beta \le 1$ where αs and βs are integers, we have

$$\Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_{\alpha s}}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] - \Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_{\beta s}}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \le \frac{O(q^2(\beta-\alpha))}{s}.$$

Conditioning on the Hamming weight of $\vec{b}$ being αs when we sample $\mathcal{D}_{1/2-4/n}$ or $\mathcal{D}_{1/2+4/n}$, the probability of the event $\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z$ is same as sampling from $\tilde{\mathcal{D}}_{\alpha s}$, because this event is invariant to permuting the indices of the s blocks, so the vector $\vec{b} = (\underbrace{1, \ldots, 1}_{\alpha s}, \underbrace{0, \ldots, 0}_{s-\alpha s})$ is equivalent to any other vector of the same Hamming weight. Hence, we have

$$\begin{aligned}
&\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_{1/2\pm 4/n}}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \\
&\qquad = \sum_{h=0}^{s} \Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_h}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \cdot \Pr\left[\mathrm{Bin}(s, 1/2 \pm 4/n) = h\right],
\end{aligned}$$

where Bin is the binomial distribution. By the Chernoff bound,

$$\begin{aligned}
&\Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_{1/2-4/n}}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] - \Pr_{(f,\vec{b},\vec{\mathbf{X}})\sim\mathcal{D}_{1/2+4/n}}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \\
&\le 2^{-\Omega(s)} + \sum_{s/4<h<3s/4} \Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_h}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \cdot \Pr\left[\mathrm{Bin}(s, 1/2 + 4/n) = h\right] \\
&\qquad - \sum_{s/4<h<3s/4} \Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_h}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \cdot \Pr\left[\mathrm{Bin}(s, 1/2 - 4/n) = h\right]
\end{aligned}$$

Then by symmetry ($\Pr\left[\mathrm{Bin}(s,p) = h\right] = \Pr\left[\mathrm{Bin}(s,1-p) = s-h\right]$) and the bound we got at the beginning by telescoping, the difference is bounded by

$$\begin{aligned}
&\sum_{s/4<h<3s/4} \left(\Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_h}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] - \Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_{s-h}}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \right) \\
&\qquad\qquad \times \Pr\left[\mathrm{Bin}(s, 1/2 - 4/n) = h\right] + 2^{-\Omega(s)} \\
&\le \sum_{s/4<h<s/2} \frac{O\left(q^2(s-2h)/s\right)}{s} \cdot \Pr\left[\mathrm{Bin}(s, 1/2 - 4/n) = h\right] + 2^{-\Omega(s)} \\
&\le \frac{O(q^2)}{s}.
\end{aligned}$$

. $\square$

B Proof of Lemma A.2

Lemma A.2. *Let $A^f : [N'] \to [M']$ be an algorithm, which makes at most q queries to its oracle $f : [N] \to [M]$. If $2q^2 < i$, then for all $z \in \{0,1\}^{m'}$,*

$$\Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_{i-1}}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] - \Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_i}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right] \le \frac{O(q^2)}{i^2}.$$

Proof. Distributions $\tilde{\mathcal{D}}_{i-1}$ and $\tilde{\mathcal{D}}_i$ differ only on the block $\vec{X_i}$. So an equivalent way to sample both distributions is that we can first sample the partition $\vec{\mathbf{X}}$, and the mapping except on the set X_i. In particular, we sample $\vec{Y_1}, \ldots, \vec{Y_{i-1}} \sim \mathcal{Y}_0$ and $\vec{Y_{i+1}}, \ldots, \vec{Y_s} \sim \mathcal{Y}_1$. After that, for fixed $\vec{\mathbf{X}}$ and $\vec{Y_1}, \ldots, \vec{Y_{i-1}}, \vec{Y_{i+1}}, \ldots, Y_s$, we sample $\vec{Y_i}$ from $\mathcal{Y}_1$ or $\mathcal{Y}_0$ for distribution $\tilde{\mathcal{D}}_i$ or $\tilde{\mathcal{D}}_{i-1}$, respectively.

For notational convenience, we define

$$\begin{aligned}
\vec{\mathbf{X}}_{-i} &\stackrel{\text{def}}{=} (\vec{X}_1, \ldots, \vec{X}_{i-1}, \vec{X}_{i+1}, \ldots, \vec{X}_s) \\
\vec{\mathbf{X}}_{\leq i} &\stackrel{\text{def}}{=} (\vec{X}_1, \ldots, \vec{X}_i) \\
\vec{\mathbf{X}}_{>i} &\stackrel{\text{def}}{=} (\vec{X_{i+1}}, \ldots, \vec{X}_n)
\end{aligned}$$

Now the difference of the probabilities can be written as

$$\Delta_i = \mathop{\mathbb{E}}_{\vec{\mathbf{Y}}_{-i},\vec{\mathbf{X}}}\left[\Pr_{\vec{Y_i}\sim\mathcal{Y}_0}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right]\right] - \mathop{\mathbb{E}}_{\vec{\mathbf{Y}}_{-i},\vec{\mathbf{X}}}\left[\Pr_{\vec{Y_i}\sim\mathcal{Y}_1}\left[\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z\right]\right]. \tag{B.1}$$

If the block $\mathbf{X}_i$ is not queried, then $A^f(w)$ behaves identically under the two distributions. Thus, to compare two probabilities better, we refine the event $\exists w \in \mathsf{BC}(f,\mathbf{X}), A^f(w) = z$ based on the block $\mathbf{X}_i$. For given $f, \vec{\mathbf{X}}$ and z, we define the following events.

$$\begin{aligned}
\forall j \in [t]\ ,\ E_{f,\vec{\mathbf{X}},z}(j) &\stackrel{\text{def}}{=} \left[\exists w \in \mathsf{BC}(f,\mathbf{X}) \text{ s.t. } A^f(w) = z \wedge \vec{X}_i(j) \in \mathsf{Query}_f(w)\right] \\
E_{f,\vec{\mathbf{X}},z}(\perp) &\stackrel{\text{def}}{=} \left[\exists w \in \mathsf{BC}(f,\mathbf{X}) \text{ s.t. } A^f(w) = z \wedge \mathsf{Query}_f(w) \cap X_i = \emptyset\right],
\end{aligned}$$

where $\mathsf{Query}_f(w)$ is the set of the queries made by the algorithm $A^f(w)$ to the f with input w.

The main events that we care about is the union of the above events we defined, so for $\mathcal{Y} \in \{\mathcal{Y}_0, \mathcal{Y}_1\}$

$$\begin{aligned}
\Pr_{\vec{Y_i}\sim\mathcal{Y}}\left[\exists w \in \mathsf{BC}(f,\mathbf{X})\right] &= \Pr_{\vec{Y_i}\sim\mathcal{Y}}\left[E_{f,\vec{\mathbf{X}},z}(\perp) \vee \left(\bigvee_{j=1}^{t} E_{f,\vec{\mathbf{X}},z}(j)\right)\right] \\
&= \Pr_{\vec{Y_i}\sim\mathcal{Y}}\left[E_{f,\vec{\mathbf{X}},z}(\perp)\right] + \Pr_{\vec{Y_i}\sim\mathcal{Y}}\left[\neg E_{f,\vec{\mathbf{X}},z}(\perp) \wedge \left(\bigvee_{j=1}^{t} E_{f,\vec{\mathbf{X}},z}(j)\right)\right].
\end{aligned}$$

An important observation is that the event $E_{f,\vec{\mathbf{X}},z}(\perp)$ does not depend on $f(X_i)$, so sampling $\vec{Y_i}$ from $\mathcal{Y}_0$ or $\mathcal{Y}_1$ does not affect the probability of the event. Hence, Equation (B.1) can be written as

$$\begin{aligned}
\Delta_i = &\mathop{\mathbb{E}}_{\vec{\mathbf{Y}}_{-i},\vec{\mathbf{X}}}\left[\Pr_{\vec{Y_i}\sim\mathcal{Y}_0}\left[\neg E_{f,\vec{\mathbf{X}},z}(\perp) \wedge \left(\bigvee_{j=1}^{t} E_{f,\vec{\mathbf{X}},z}(j)\right)\right]\right] \\
&- \mathop{\mathbb{E}}_{\vec{\mathbf{Y}}_{-i},\vec{\mathbf{X}}}\left[\Pr_{\vec{Y_i}\sim\mathcal{Y}_1}\left[\neg E_{f,\vec{\mathbf{X}},z}(\perp) \wedge \left(\bigvee_{j=1}^{t} E_{f,\vec{\mathbf{X}},z}(j)\right)\right]\right].
\end{aligned}$$

Now, for the probability over $\mathcal{Y}_0$ part, we apply the union bound.

$$\mathop{\mathbb{E}}_{\vec{\mathbf{Y}}_{-i},\vec{\mathbf{X}}}\left[\Pr_{\vec{Y_i}\sim\mathcal{Y}_0}\left[\neg E_{f,\vec{\mathbf{X}},z}(\perp) \wedge \left(\bigvee_{j=1}^{t} E_{f,\vec{\mathbf{X}},z}(j)\right)\right]\right] \leq \mathop{\mathbb{E}}_{\vec{\mathbf{Y}}_{-i},\vec{\mathbf{X}}}\left[\sum_{j=1}^{t}\Pr_{\vec{Y_i}\sim\mathcal{Y}_0}\left[\neg E_{f,\vec{\mathbf{X}},z}(\perp) \wedge E_{f,\vec{\mathbf{X}},z}(j)\right]\right]$$

For the $\mathcal{Y}_1$ part, we bound the probability via the inclusion-exclusion principle.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\vec{\mathbf{Y}}_{-i},\vec{\mathbf{X}}}\left[\Pr_{\vec{Y_i}\sim\mathcal{Y}_1}\left[\neg E_{f,\vec{\mathbf{X}},z}(\perp)\wedge\left(\bigvee_{j=1}^{t}E_{f,\vec{\mathbf{X}},z}(j)\right)\right]\right]\\
\geq&\mathop{\mathbb{E}}_{\vec{\mathbf{Y}}_{-i},\vec{\mathbf{X}}}\left[\sum_{j=1}^{t}\Pr_{\vec{Y_i}\sim\mathcal{Y}_1}\left[\neg E_{f,\vec{\mathbf{X}},z}(\perp)\wedge E_{f,\vec{\mathbf{X}},z}(j)\right]-\Pr_{\vec{Y_i}\sim\mathcal{Y}_1}\left[\exists j\neq j',\neg E_{f,\vec{\mathbf{X}},z}(\perp)\wedge E_{f,\vec{\mathbf{X}},z}(j)\wedge E_{f,\vec{\mathbf{X}},z}(j')\right]\right]
\end{aligned}$$

Observe that $A^f(w)$ only queries X_i at most once for all $w\in\mathsf{BC}(f,\mathbf{X})$, and the marginal distributions of the mapping on $\vec{X_i}(j)$ for every $j\in[t]$ are the same in both $\mathcal{Y}_1$ and $\mathcal{Y}_0$ cases, so for every $j\in[t]$

$$\Pr_{\vec{Y_i}\sim\mathcal{Y}_0}\left[\neg E_{f,\vec{\mathbf{X}},z}(\perp)\wedge E_{f,\vec{\mathbf{X}},z}(j)\right]=\Pr_{\vec{Y_i}\sim\mathcal{Y}_1}\left[\neg E_{f,\vec{\mathbf{X}},z}(\perp)\wedge E_{f,\vec{\mathbf{X}},z}(j)\right]$$

Therefore, the difference between two cases is bounded as

$$\begin{aligned}
\Delta_i&\leq\mathop{\mathbb{E}}_{\vec{\mathbf{Y}}_{-i},\vec{\mathbf{X}}}\left[\Pr_{\vec{Y_i}\sim\mathcal{Y}_1}\left[\exists j\neq j',\neg E_{f,\vec{\mathbf{X}},z}(\perp)\wedge E_{f,\vec{\mathbf{X}},z}(j)\wedge E_{f,\vec{\mathbf{X}},z}(j')\right]\right]\\
&=\Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_i}\left[\exists j\neq j',\neg E_{f,\vec{\mathbf{X}},z}(\perp)\wedge E_{f,\vec{\mathbf{X}},z}(j)\wedge E_{f,\vec{\mathbf{X}},z}(j')\right]. \qquad \text{(B.2)}
\end{aligned}$$

To bound the term, we consider another way to sample $(f,\vec{\mathbf{X}})$ from $\tilde{\mathcal{D}}_i$. Given $(f,\vec{\mathbf{X}})$, we define $B:\mathbf{X}_{\leq i}\to[i]$ by

$$B(x)=\text{the block that } x \text{ is in} = \text{the unique } i'\leq i \text{ s.t. } \exists j,\vec{X}'_i(j)=x.$$

We will re-sample the "B part" after getting $(f,\vec{\mathbf{X}})$. Namely, we will sample B given fixed f and $\vec{\mathbf{X}}_{>i}$) using principle of deferred decisions. Note that conditioned on f and $\vec{\mathbf{X}}_{>i}$, B is a uniformly random regular mapping from $\vec{X}_{\leq i}$ to $[i]$ where regular means that all preimage sets $B^{-1}(i')$ are of size t.

Along the way of sampling B, $B:\vec{\mathbf{X}}_{\leq i}\to[i]\cup\{*\}$ where "$*$" represent values not yet determined as before. Initially, $B(x)=*$ for all $x\in\vec{\mathbf{X}}_{\leq i}$. For an input $w\in[N']$, we say w is *partially block compatible*, written as $w\in\mathsf{PBC}(f,\vec{\mathbf{X}}_{>i},B)$ if $A^f(w)$ queries each block (defined by $\vec{\mathbf{X}}_{>i}$ or B) at most once (among the queries whose block is determined).

The procedure for sampling B given fixed f and $\vec{\mathbf{X}}_{>i}$ is as follows.

Procedure B.1

```
1. Set $B(x)=*$ for all $x\in[N]\setminus\mathbf{X}_{>i}$.
2. While $\exists w$ s.t. $w\in\mathsf{PBC}(f,\vec{\mathbf{X}}_{>i},B)$ and $A^f(w)=z$,
    (a) Randomly assign $B$ on undetermined element of $\mathsf{Query}_f(w)$ conditional on assignment to $B$ so far. That is, for each $x\in\mathsf{Query}_f(w)$ s.t. $B(x)=*$ set $B(x)=i'$ with probability $\frac{t-|\{x':B(x')=i'\}|}{it-|\{x':B(x')\neq*\}|}$.
3. Randomly assign $B$ on all undetermined elements conditioned on assignment to $B$ so far.
```

By considering the above sampling procedure, let w_ℓ be the value of w chosen in the ℓ-th iteration of the while loop (Step 2). Then we define the following events for $\ell \in \mathbb{N}$.

$$\begin{aligned}
E_\ell^{(0)} =& \big[\text{None of the new assignments to } B \text{ in } \ell\text{-th iteration equal } i \\
&\quad \wedge \text{ after } \ell\text{-th iteration, } w_\ell \neq \mathsf{PBC}(f, \vec{\mathbf{X}}_{>i}, B).\big] \\
E_\ell^{(1)} =& [\text{Exactly one of the new assignments to } B \text{ in } \ell\text{-th iteration equals } i.] \\
E_\ell^{(\geq 2)} =& [\text{At least two of the new assignments to } B \text{ in } \ell\text{-th iteration equals } i.]
\end{aligned}$$

Denote the assignments of B after the first $\ell - 1$ rounds in the while loop as $B_{\ell-1}$. Suppose $q\ell \leq it/2$ and $q^2 \leq i/2$, then we have

$$\begin{aligned}
\Pr\left[E_\ell^{(0)} \mid B_{\ell-1}\right] &\leq \binom{q}{2} \cdot \frac{t}{it - q\cdot(\ell-1)} \leq \frac{1}{2} \\
\Pr\left[E_\ell^{(1)} \mid B_{\ell-1}\right] &\leq q \cdot \frac{t}{it - q\cdot(\ell-1)} \leq \frac{2q}{i} \\
\Pr\left[E_\ell^{(\geq 2)} \mid B_{\ell-1}\right] &\leq \sum_{j=2}^{q} \binom{q}{j} \cdot \left(\frac{t}{it - q\cdot(\ell-1)}\right)^j \leq \frac{2q}{i^3} \\
&\leq \sum_{j=2}^{q} \left(\frac{q}{2}\right)^j \left(\frac{2}{i}\right)^j \leq \sum_{j=2}^{q} \left(\frac{q}{i}\right)^j \leq \frac{2q^2}{i^2}
\end{aligned} \tag{B.3}$$

Let L be a parameter to be chosen later. The event in Equation (B.2) happens only if the event $E^{(1)}$ happens at least twice in the while loop and for the rest of the while loops, $E^{(0)}$ or $E^{(\geq 2)}$ happens. We focus on the sampling procedure for the first L rounds. Then Equation (B.2) can be bounded as

$$\Pr_{(f,\vec{\mathbf{X}})\sim\tilde{\mathcal{D}}_i}\left[\exists j \neq j', \neg E_{f,\vec{\mathbf{X}},z}(\perp) \wedge E_{f,\vec{\mathbf{X}},z}(j) \wedge E_{f,\vec{\mathbf{X}},z}(j')\right]$$

$$\leq \sum_{0<\ell_1<\ell_2\leq L} \Pr\left[\left(E_1^{(0)} \wedge \cdots \wedge E_{\ell_1-1}^{(0)}\right) \wedge E_{\ell_1}^{(1)} \wedge \left(E_{\ell_1+1}^{(0)} \wedge \cdots \wedge E_{\ell_2-1}^{(0)}\right) \wedge E_{\ell_2}^{(1)}\right] \tag{B.4}$$

$$+ \sum_{0<\ell\leq L} \Pr\left[\left(F_1 \wedge \cdots \wedge E_{\ell-1}^{(0)}\right) \wedge E_\ell^{(1)} \wedge \left(E_{\ell+1}^{(0)} \wedge \cdots \wedge E_L^{(0)}\right)\right] \tag{B.5}$$

$$+ \sum_{0<\ell\leq L} \Pr\left[\left(E_1^{(0)} \wedge \cdots \wedge E_{\ell-1}^{(0)}\right) \wedge E_\ell^{(\geq 2)}\right] \tag{B.6}$$

$$+ \Pr\left[E_1^{(0)} \wedge \cdots, \wedge E_L^{(0)}\right] \tag{B.7}$$

As long as $qL \le it/2$, we can bound Equation (B.4), (B.5), (B.6) and (B.7) using Equation B.3:

$$\begin{aligned}
&\text{Equation (B.4)} \le \sum_{0<\ell_1<\ell_2\le L} \frac{1}{2^{\ell_2-2}} \cdot \frac{4q^2}{i^2} \le 16 \sum_{0<\ell_2\le L} \frac{\ell_2}{2^{\ell_2}} \cdot \frac{q^2}{i^2} = O\left(\frac{q^2}{i^2}\right)\\
&\text{Equation (B.5)} \le \sum_{0<\ell\le L} \frac{1}{2^{L-1}} \cdot \frac{2q}{i} = O\left(2^{-L}\right)\\
&\text{Equation (B.6)} \le \sum_{0<\ell\le L} \frac{1}{2^{\ell-1}} \cdot \frac{2q^2}{i^2} = O\left(\frac{q^2}{i^2}\right)\\
&\text{Equation (B.5)} \le O\left(2^{-L}\right)
\end{aligned}$$

If we choose $L = 2\log(i/q)$ (which satisfies $qL \le it/2q$), then all Equation (B.4), (B.5), (B.6) and (B.7) is at most $O\left(q^2/i^2\right)$, and so is Δ_i. □

B.1 Proof of Lemma 2.1

Lemma 2.1. *If there exists a (ε, Δ)-flattening algorithm $A^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ for function $f : \{0,1\}^n \to \{0,1\}^m$ with query complexity q, then there exists a k-SDU algorithm $A^f : \{0,1\}^{n''} \to \{0,1\}^{n''-3k}$ where $n'' = O(n' + m')$ for function $f : \{0,1\}^n \to \{0,1\}^m$ with query complexity q and $k = \Omega(\min\{\Delta, \log(1/\varepsilon)\})$. In particular, there exists such a k-SDU algorithm with query complexity $O(k \cdot \min\{n,m\}^2)$.*

Claim B.1. *If there exists a (ε, Δ)-flattening algorithm $A^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ for function $f : \{0,1\}^n \to \{0,1\}^m$ with query complexity q, then there exists an k-SDU algorithm $B^f : \{0,1\}^{n''} \to \{0,1\}^{m''}$ where $n'' = O(n'+m')$ and $m'' = O(n'+m')$ for function $f : \{0,1\}^n \to \{0,1\}^m$ with query complexity q and $k = \Omega(\min\{\Delta, \log(1/\varepsilon)\})$.*

Claim B.2. *If there exists a k-SDU algorithm $A^f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ for function $f : \{0,1\}^n \to \{0,1\}^m$ with query complexity q, then there exists an $(k-1)$-SDU algorithm $B^f : \{0,1\}^{n''} \to \{0,1\}^{m''}$ where $n'' = O(n')$ and $m'' = n'' - 3k$ for function $f : \{0,1\}^n \to \{0,1\}^m$ with query complexity q.*

proof of Claim B.1. This proof mostly follows the idea in [GSV99b]. It suffices to prove the existence of $\Omega(k)$-SDU algorithm for $k = \min\{\Delta, \log(1/\varepsilon)\}$. Let $\mathcal{H}_{a,b}$ be a family of 2-universal hash function from a bits to b bits. We sample hash functions h_1 and h_2 from $\mathcal{H}_{m',\kappa}$ and $\sim \mathcal{H}_{n',n'-\kappa-k/3}$, respectively, where κ is the parameter chosen by the flattening algorithm A^f. We will show that

$$B^f(w, h_1, h_2) = \left(h_1, h_1(A^f(w)), h_2, h_2(w)\right)$$

is a $\Omega(k)$-SDU algorithm. We denote the output of $B^f(w, h_1, h_2)$ as a jointly distributed random variables (H_1, Z_1, H_2, Z_2) when $w \sim U_{n'}, h_1 \sim \mathcal{H}_{m',\kappa}$ and $h_2 \sim \mathcal{H}_{n',n'-\kappa-k/3}$.

1. When $(f, \tau) \in \mathrm{EA}_Y$, there exists a distribution Z_H with $\mathsf{H}_{\min}(Z_H) \ge \kappa + \Delta$ such that $d_{\mathrm{TV}}\left(A^f(U_{n'}), Z_H\right) \le \varepsilon$. First, we show that (H_1, Z_1) is close to uniform. By the Leftover Hash Lemma, $d_{\mathrm{TV}}\left((H_1, H_1(Z_H), (H_1, U_\kappa)\right) \le 2^{-\Delta/3}$, and so

$$\begin{aligned}
d_{\mathrm{TV}}\big((H_1, Z_1), (H_1, U_\kappa)\big) \le& d_{\mathrm{TV}}\left(A^f(U_{n'}), Z_H\right) + d_{\mathrm{TV}}\big((H_1, H_1(Z_H), (H_1, U_\kappa)\big)\\
\le& 2^{-\Delta/3} + \varepsilon \le 2^{-\Omega(k)}.
\end{aligned}$$

For the (H_2, Z_2) of part, we will show that with high probability over sampling (h_1, z_1) from (H_1, Z_1), the distribution (H_2, Z_2) conditioned on (h_1, z_1) is close to uniform. Since (H_1, Z_1) is $2^{-\Omega(k)}$-close to uniform, by the Markov inequality, with probability at least $1 - 2^{-\Omega(k)}$ over choosing (h_1, z_1) from (H_1, Z_1), we have

$$\Pr\left[h_1(A^f(U_{n'})) = z_1\right] = \Pr\left[Z_1 = z_1 \mid H_1 = h_1\right] \geq \frac{1}{2} \cdot 2^{-\kappa}.$$

Thus, except for $2^{-\Omega(k)}$ probability over (h_1, z_1), the number of w such that $h_1(A^f(w)) = z_1$ is at least $2^{n'-\kappa-1}$. Again, by the Leftover Hash Lemma, (H_2, Z_2) is $2^{-\Omega(k)}$-close to uniform conditioned on any such (h_1, z_1). We then can conclude that (H_1, Z_1, H_2, Z_2) is $2^{-\Omega(k)}$-close to uniform.

2. When $(f, \tau) \in \mathrm{EA}_N$, there exists a distribution Z_L with $\mathsf{H}_{\max}(Z_L) \leq \kappa - \Delta$ such that $d_{\mathrm{TV}}\left(A^f(U_{n'}), Z_L\right) \leq \varepsilon$. For every fixed h_1 and h_2, we will bound the support size of (Z_1, H_2, Z_2) conditioned on $H_1 = h_1$ and $H_2 = h_2$. We divide $\mathrm{Supp}(Z_1, Z_2)$ into three subset according to $z_1 \in \mathrm{Supp}(Z_1)$.

$$\begin{cases} S_1 = \{(z_1, z_2) : z_1 \in \mathrm{Supp}(Z_L)\} \\ S_2 = \{(z_1, z_2) : \Pr\left[Z_1 = z_1\right] \geq 2^{-\kappa-2k/3} \text{ and } z_1 \notin \mathrm{Supp}(Z_L)\} \\ S_3 = \{(z_1, z_2) : \Pr\left[Z_1 = z_1\right] < 2^{-\kappa-2k/3} \text{ and } z_1 \notin \mathrm{Supp}(Z_L)\} \end{cases}$$

Since, $\mathrm{Supp}(Z_1, Z_2) = S_1 \cup S_2 \cup S_3$, it suffices to show that $|S_i| \leq 2^{-\Omega(k)} \cdot \left|\{0,1\}^\kappa \times \{0,1\}^{n'-\kappa-k/3}\right|$ for all $i = 1, 2, 3$.

   (a) For S_1, by definition, $\mathsf{H}_{\max}(Z_L) \leq \kappa - \Delta$ implies that $|\mathrm{Supp}(Z_L)| / |\{0,1\}^\kappa| \leq 2^{-\Delta}$, and so

$$|S_1| \leq 2^{-\Delta} \cdot \left|\{0,1\}^\kappa \times \{0,1\}^{n'-\kappa-k/3}\right| \leq 2^{-\Omega(k)} \cdot \left|\{0,1\}^\kappa \times \{0,1\}^{n'-\kappa-k/3}\right|.$$

   (b) For S_2, since $d_{\mathrm{TV}}\left(A^f(U_{n'}), Z_L\right) \leq \varepsilon$, $\sum_{z_1 \notin \mathrm{Supp}(Z_L)} \Pr\left[Z_1 = z_1\right] \leq \varepsilon$. Each z_1 such that $\Pr\left[Z_1 = z_1\right] \geq 2^{-\kappa-2k/3}$ contributes at least $2^{-\kappa-2k/3}$ towards ε, so

$$\left|\{z_1 : \Pr\left[Z_1 = z_1\right] \geq 2^{-\kappa-2k/3} \text{ and } z_1 \notin \mathrm{Supp}(Z_L)\}\right| \leq \varepsilon \cdot 2^{\kappa+2k/3}.$$

   Then we have $|S_2| \leq 2^{-\Omega(k)} \left|\{0,1\}^\kappa \times \{0,1\}^{n'-\kappa-k/3}\right|$, since $k \leq \log(1/\varepsilon)$.

   (c) For S_3, if $\Pr\left[Z_1 = z_1\right] < 2^{-\kappa-2k/3}$, then the number of $w \in \{0,1\}^{n'}$ such that $h_1(A^f(w)) = z_1$ is at most $2^{n'-\kappa-2k/3}$, which is at most a $2^{-k/3}$ fraction of $\{0,1\}^{n'-\kappa-k/3}$. Therefore, $|S_3| \leq 2^{-\Omega(k)} \cdot \left|\{0,1\}^\kappa \times \{0,1\}^{n'-\kappa-k/3}\right|$.

Thus, we conclude that B^f is a $\Omega(k)$-SDU algorithm. □

proof of Claim B.2.

Definition B.1 (average min-entropy [DORS08])**.** *Let (X, Y) be jointed distributed random variables. The average min-entropy of X conditioned on Z is*

$$\mathsf{H}_{\min}(X|Y) \overset{\text{def}}{=} \log\left(\frac{1}{\mathbb{E}_{y\leftarrow Y}\left[\max_x \Pr\left[X = x|Y = y\right]\right]}\right)$$

Lemma B.1 (Generalized Leftover Hash Lemma [DORS08])**.** *Let (X, Y) be a jointed distributed random variables such that $\mathsf{H}_{\min}(X|Y) \geq k$. Let $\mathcal{H}_{n,m} = \{h : \{0, 1\}^n \to \{0, 1\}^m\}$ be a family of universal hash function where h can be described in $(n + m)$ bits and $m = k - 2\log(1/\varepsilon) + 2$. Then*

$$d_{\mathrm{TV}}\left((h(X), Y, h), (U_m, Y, h)\right) \leq \varepsilon$$

where U_m is a uniform m bits string.

Let $\mathcal{H}_{n',n'-m'-3k} = \{h : \{0, 1\}^n \to \{0, 1\}^m\}$ be a family of universal hash function where h can be described in $d = 2n' - m' - 3k$ bits. Based on the given k-SDU algorithm $A^f : \{0, 1\}^{n'} \to \{0, 1\}^{m'}$, we define the algorithm $B^f : \{0, 1\}^{n'+d} \to \{0, 1\}^{n'+d-3k}$ as

$$B^f(w, h) \overset{\text{def}}{=} (A^f(w), h(w), h).$$

By the chain rule of average min-entropy ([DORS08, Lemma 2.2b])

$$\mathsf{H}_{\min}(w|A(w)) \geq \mathsf{H}_{\min}(w) - |A(w)| = n' - m',$$

and hence

$$d_{\mathrm{TV}}((A(w), \mathrm{Ext}(w, v)), (A(w), U_{n'-m'+d-2k-O(1)})) \leq 2^{-k}.$$

Therefore, when $\mathsf{H}_{\mathrm{sh}}(f) \geq \tau + 1$

$$\begin{aligned}
&d_{\mathrm{TV}}\left(B^f(U_{n'+d}), U_{n'+d-3k}\right)\\
&= d_{\mathrm{TV}}\left((A^f(w), h(w), h), (U_{m'}, U_{n'-m'+d-3k})\right)\\
&= d_{\mathrm{TV}}\left(A^f(w), U_{m'}\right) + d_{\mathrm{TV}}\left((A^f(w), h(w), h), (A^f(w), U_{n'-m'+d-3k})\right)\\
&\leq 2^{-k} + 2^{-k} = 2^{-(k-1)}.
\end{aligned}$$

The last inequality is by the property of k-SDU algorithm and Lemma B.1.

On the other hand, if $\mathsf{H}_{\mathrm{sh}}(f) \leq \tau - 1$,

$$\left|\mathrm{Supp}(B^f(U_{n'+d}))\right| \leq 2^{m'-k} \cdot 2^{n'-m'+d-3k} \leq 2^{(n'+d-3k)-k}.$$

Therefore, B^f is an $(k - 1)$-SDU algorithm with desired parameter. □