Polynomial-time targeted attacks on coin tossing for any number of corruptions *

Omid Etesami† Ji Gao‡ Saeed Mahloujifar§ Mohammad Mahmoody¶

November 1, 2021

Abstract

Consider an n-message coin-tossing protocol between n parties $P_1, \ldots, P_n$, in which P_i broadcasts a single message w_i in round i (possibly based on the previously shared messages) and at the end they agree on bit b. A k-replacing adversary A_k can change up to k of the messages as follows. In every round i, the adversary who knows all the messages broadcast so far, as well as a message w_i that is prepared by P_i to be just sent, can can to replace the prepared message w_i with its own choice. A *targeted* adversary prefers the outcome $b' = 1$, and its bias is defined as $\mu' - \mu$, where $\mu' = \Pr[b' = 1]$ (resp. $\Pr[b = 1] = \mu$) refers to the probability of outputting 1 when the attack happens (resp. does not happen). In this work, we study k-replacing targeted attacks, their computational efficiency, and optimality, for all $k \in [n]$.

- **Large messages.** When the messages are allowed to be arbitrarily long, we show that polynomial-time k-replacing targeted attacks can achieve bias $\Omega(\mu k/\sqrt{n})$ for *any* k (and any protocol), which is optimal up to a constant factor for any $\mu = \Theta(1)$. Previously, it was known how to achieve such bias only for $k = \Omega(\sqrt{n})$ (Komargodski-Raz [DISC'18], Mahloujifar-Mahmoody [ALT'19], and Etesami-Mahloujifar-Mahmoody [SODA'20]). This proves a computational variant of the isoperimetric inequality for product spaces under $k = o(\sqrt{n})$ Hamming distance. As a corollary, we also obtain improved $\mathrm{poly}(n)$-time targeted poisoning attacks on deterministic learners, in which the adversary can increase the probability of any efficiently testable bad event over the produced model from $\mu = 1/\mathrm{poly}(n)$ to $\mu + \Omega(\mu k/\sqrt{n})$ by changing k out of n training examples.
- **Binary messages.** When the messages $w_1, \ldots, w_n$ are uniformly random bits, we show that if $\mu = \Pr[b = 1] = \Pr[\sum w_i \geq t] = \beta_n^{(t)}$ for $t \in [n]$ is the probability of falling into a Hamming ball, then polynomial-time k-replacing targeted attacks can achieve $\mu' = \Pr[b' = 1] = \beta_n^{(t-k)}$, which is optimal due to the simple majority protocol. Thus, as corollary we obtain an alternative proof of the Harper's celebrated vertex isoperimetric inequality in which the optimal adversary (that maps random points to a set of measure μ by changing at most k bits) is limited to be online and run in polynomial time. Previously, Lichtenstein, Linial, and Saks [Combinatorica'89] showed how to achieve $\mu' = \Pr[b' = 1] = \beta_{n-k}^{(t-k)}$ (using computationally unbounded attacks), which is optimal for adaptive adversaries who decide on corrupting parties before seeing their messages.

*This is the full version of a paper, with the same title, appearing in the Theory of Cryptography Conference (TCC) 2021.

†School of Mathematics, IPM, P.O. Box 19395-5746, Tehran, Iran.

‡University of Virginia, Charlottesville, VA, USA.

§Princeton University, Princeton, NJ, USA.

¶University of Virginia, Charlottesville, VA, USA.

Contents

1 Introduction

Collective coin tossing [Blu84] is a fundamental problem in cryptography in which a set of n parties aim to jointly produce a random bit b that remains (close to) random even if an adversary controls a subset of these parties. The simple majority protocol $\mathrm{maj}(b_1, \ldots, b_n)$, when n is odd and each bit b_i is broadcast by party P_i, is robust in the following strong sense: Any adversary who even gets to see *all* the messages and then replaces at most $k \in [n]$ of the them can only bias the output bit by at most by $O(k/\sqrt{n})$ [BOL90]. In a nutshell, in this work we ask *how optimal is the majority protocol against such attacks?* We study this question from various angels as explained below.

Problem setting. Suppose Π is an n-round coin-tossing protocol between n parties, where party P_i sends a single message w_i in round i that could depend on all the previous messages, and the final bit b is a deterministic function of all messages.[1] Now, suppose an adversary aims to increase the probability of $\Pr[b = 1]$. We call this a *targeted* attack, as adversary can choose the target direction of the bias.[2] We deal with k-*replacing* adversaries who can replace k of the messages as follows. Suppose messages $w_1, \ldots, w_{i-1}$ are already finalized and party P_i is about to send w_i in round i. The adversary will have a chance to replace w_i, based on the knowledge of w_i.[3] Equivalently, we will think of the protocol as a *random process* $(w_1, \ldots, w_n)$ with n steps, and a k-replacing adversary will be allowed to *immediately replace* the content

[1] This is also called a *single-turn* protocol.

[2] In contrast, *untargeted* adversaries can bias the output towards *either* of 0 or 1.

[3] This is also called the *strongly adaptive* corruption model [GKP15].

of k of the steps, in which case the rest of the random process will depend on the new values. The goal of the adversary is to increase the probability of $\Pr[b = 1]$ for a Boolean function $f(w_1, \ldots, w_n) = b \in \{0, 1\}$. Informally speaking, we would like to know what are the most robust random processes in this setting.

Targeted aspect. Studying targeted attacks is important due to several reasons. Firstly, targeted attacks allow modeling adversaries who have a particular output preferred in mind. For example, the coin tossing model's output might determine whether a contract would be signed or not. Then, a party who prefers signing the contract wants to increase the chance of outputting $b = 1$. Moreover, targeted attacks allow modeling attacks on specific "undesired" properties like $\mathcal{B}$ defined over random processes; namely, the adversary aims to increase the probability of B happening at the end. Below in the introduction and later in Section A we further discuss applications such as targeted poisoning attacks in adversarial machine learning and computational isoperimetry results.

Robustness of threshold functions. For a setting where w_i is a uniform random bit b_i, consider the threshold function f defined as $f(b_1, \ldots, b_n) = 1$ whenever $\sum b_i \geq t$ and let $\beta_n^{(t)} = \Pr[\sum b_i \geq t]$. Then we get a robust protocol in the following sense. Any k-replacing adversary will be limited to achieve $\Pr[b = 1] \leq \beta_n^{(t-k)}$, because all it can do is to replace k ones with zeros. In particular, it can be shown that for the majority function (for odd n) any k-replacing attack increase $\Pr[b = 1]$ by at most $O(k/\sqrt{n})$.

In this work, we study the optimality of the simple threshold/majority protocols and ask the following.

1. If $\Pr[b = 1] = 1/2$ holds originally, for a given fixed $k = k(n)$, can k-replacing adversaries increase the probability of $\Pr[b = 1]$ by $\Omega(k/\sqrt{n})$ in *every* n-step random process with arbitrarily long messages?
2. For simpler models such as those with uniformly random bits, can we obtain *optimal* attacks that prove the threshold protocols to be the best possible for all $\Pr[b = 1] = \beta_n^{(t)}$?

We answer both questions above affirmatively. Notably, we even obtain *polynomial-time* attacks. Before describing our results in details, we briefly discuss what was known before our work.

Previous work for uniform binary messages. Lichtenstein, Linial, and Saks [LLS89] showed that the threshold protocols are optimal when the messages are uniform random bits, but under a weaker attack model where the adversary is supposed to corrupt parties before seeing their message. In particular, they showed that if $\Pr[f(b_1, \ldots, b_n) = 1]$ without attack is the probability of the threshold function $\Pr[\sum b_i \geq t] = \beta_n^{(t)}$, then there is an *adaptive* attack with budget k that achieves $\Pr[f(b_1, \ldots, b_n) = 1] \geq \beta_{n-k}^{(t-k)}$. However, this attack was information theoretic and not polynomial time. It also remained open whether k-replacing attacks can improve upon the bound of [LLS89] and potentially match the robustness of threshold functions. In other words, prior to our work, it was not known whether threshold functions are optimal against k-replacing attacks.

Previous work on arbitrary length messages. Kalai, Komargodski, and Raz [KKR18] showed that in the "many-replacement" regime where $k = \Omega(\sqrt{n})$, a different attack in the binary setting of [LLS89] can be achieved in polynomial time.[4] Building upon [KKR18], Etesami, Mahloujifar and Mahmoody [MM19, EMM20] showed how to extend this result to arbitrary message length and obtain (again targeted) attacks

[4] Interestingly, the main result of [KKR18] focuses on *non-targeted* attacks and shows that the output of any single-turn protocol can be attacked (only information theoretically) by a (standard) adaptive *non-targeted* adversary replacing $k = \Omega(\sqrt{n})$ parties. The recent breakthrough of Haitner and Karidi-Heller [HKH20] generalized the main result of [KKR18] to any general, perhaps multi-turn, protocol. Our focus in this work, however, is on single-turn protocols.

in polynomial time, but again only when $k \geq \Omega(\sqrt{n})$. (See Section 1.2 for more discussions on why those proofs lead to many replacements.) Finally, Khorasgani, Maji, Mukherjee, and Wang [KMM19, KMW21] showed how to get *non-targeted* attacks for large messages when $k = 1$.

1.1 Our results

Previous works left open our two main questions. In this work, we resolve both of these questions and show that (1) when $\Pr[b = 1] = \Theta(1)$, then majority is optimal up to a constant factor against k-replacing adversaries for all adversary budget k (including the "few corruption regime"), and (2) when messages are uniformly random bits, for any initial probability of Hamming balls $\Pr[b = 1] = \Pr[\sum b_i \geq t]$, the corresponding threshold function is optimal, *even up to exact constants*.

Long messages. We start by describing our main results about attacking protocols with long messages.

Theorem 1 (Main result 1 – arbitrary messages)**.** *Let Π be any single-turn polynomial-time coin-tossing protocol between n parties to obtain an output bit b in which, originally (before any attack) it holds that $\Pr[b = 1] = \mu$. For any $k \in [n]$, there is a k-replacing polynomial time attack that increases the probability of outputting $b = 1$ by a probability that can get arbitrarily close to:*

$$\left(1 - \left(1 - \frac{\mu}{\sqrt{n}}\right)^k\right) \cdot \left(1 - e^{-2} - \mu\right).$$

When $k = \omega(\sqrt{n/\mu})$, one can use a different attack from [EMM20] which shows how to get almost full bias (i.e., probability of outputting 1 close to one). The novelty of Theorem 1 is for the case of few corruptions $k = O(\sqrt{n/\mu})$. In particular, for every $k = O(\sqrt{n/\mu})$ Theorem 1 guarantees bias of $\Omega(k \cdot \mu/\sqrt{n})$. Therefore, Theorem 1 resolves our first main question above; i.e., the majority protocol of [BOL89] is optimal, up to a constant factor, for targeted attacks on any single-turn protocol when $\mu = \Theta(1)$.

To prove Theorem 1, we use ideas from the attack of [MM19] (see Section 1.2). See Theorems 15 (resp. Theorem 18) for a formalization of the information-theoretic (resp. computational) variant.

Reduction to gap finders. We further observe (in Appendix C) a connection between the gap finding attacks of [CI93, KMM19] for martingales and targeted attacks on coin tossing. In [CI93], it was shown how to obtain *non-targeted* attacks from such gap finders. We note, however, that this observation is merely for sake of completeness, and the results obtained this way are subsumed by our main result of Theorem 1.

Uniform binary messages. Our next result solves the problem completely for protocols with uniform random bits, as long as the probability of outputting 1 is that of a threshold function.

Theorem 2 (Main result 2 – uniformly random bits)**.** *Let Π be any single-turn polynomial-time coin-tossing protocol between n parties to obtain an output bit b in which the parties share uniformly random bits $b_1, \ldots, b_n$. Suppose originally (before any attack) it holds that $\Pr[b = 1] \geq \Pr[\sum b_i \geq t] = \beta_n^{(t)}$ for $t \in [n]$. Then, for any $k \in [t]$, there is a k-replacing attack that increases the probability of outputting $b = 1$ to at least $\beta_n^{(t-k)}$. Moreover, if it further holds that $\Pr[b = 1] \geq 1/\operatorname{poly}(n)$ is non-negligible, then there will be* polynomial-time *k-replacing attacks that can get arbitrarily close to the same bound of $\beta_n^{(t-k)}$.*

To prove Theorem 2, we also use ideas from the recent work of [KMW21]. See Theorem 26 (resp. Theorem 39) for a formal version of the information theoretic (resp. computational) variant of Theorem 2.

Note that Theorem 2 shows something perhaps surprising about the power of *online* attacks against coin tossing protocols. It shows that online attacks are *as powerful* as offline attacks, when we consider

the most robust functions with $\Pr[b = 1] = \beta_n^{(t)}$ being that of a Hamming ball. In fact, we present such attacks that run in polynomial time, and this implies a new tight computational variant for the celebrated vertex isoperimetry inequality of Harper [Har66] for sets of probability $\beta_n^{(t)}$. Indeed, the vertex isoperimetric inequality in the Boolean hypercube states that for any set $\mathcal{S} \subseteq \{0,1\}^n$ of probability $\Pr[(b_1 \ldots, b_n) \in \mathcal{S}] = \beta_n^{(t)}$, the probability of the set of points (inside or outside $\mathcal{S}$) with a neighbor in $\mathcal{S}$ of distance at most k is at least $\beta_n^{(t-k)}$. Our Theorem 2 matches this bound exactly, and even shows how to *find* such close neighbors (in $\mathcal{S}$) in *polynomial time* and even in an *online manner* for at least $\beta_n^{(t-k)}$ fraction of $\{0,1\}^n$.

	Targeted	Poly-time	Corruption model	Budget k	Messages	Rounds
[LLS89]	✓	-	Adaptive	Any	Uniform bits	Any
[KKR18]	✓	✓	Adaptive	$\Omega(\sqrt{n})$	Uniform bits	Any
This work	✓	✓	Replacing	Any	Uniform bits	Any
[MM19, EMM20]	✓	✓	Replacing	$\Omega(\sqrt{n})$	Arbitrary	Any
This work	✓	✓	Replacing	Any	Arbitrary	Any
[CI93]	-	-	Replacing	1	Arbitrary	Any
[GKP15]	-	-	Replacing	$\Omega(\sqrt{n})$	Arbitrary	1
[McD89, Tal95]	-	-	Replacing	Any	Arbitrary	1
[KKR18, HKH20]	-	-	Adaptive	$\Omega(\sqrt{n})$	Arbitrary	Any
[KMM19]	-	-	Replacing	1	Arbitrary	Any
[KMW21]	-	-	Adaptive	1	Arbitrary	Any

Table 1: Summary of related attacks on single-turn coin tossing protocols.

Applications. We can directly apply the attacks of Theorems 1 and 2 to obtain the applications below.

- **Targeted data poisoning on learners.** Theorem 1 can model any random process $(w_1, \ldots, w_m)$ that generates an object h that might or might not belong to an (undesirable) set $\mathcal{B}$ with some probability μ. In that case, we can define the output of the process to be $b = 1$ if $h \in \mathcal{B}$, and then an adversary can *increase* the probability of falling into $\mathcal{S}$ through a k-replacing attack. Now, suppose w_i is a batch of data provided by the i^{th} party, and let h be a model that is deterministically trained on the data set $w_1 \cup \cdots \cup w_n$. Suppose there is an specific (efficiently testable) property $\mathcal{B}$ defined over h that an adversary wants to increase its probability (e.g., h makes a specific decision on a particular test instance). Theorem 15 shows that the adversary can always increase the probability of $\mathcal{B}$ from μ to $\mu + \Omega(k/\sqrt{n})$ by changing only k of the training batches. (See Theorem 43 for a formalization of our result.) Previously, Etesami, Mahloujifar, and Mahmoody [MM19, EMM20] proved such results only for when $k \geq \Omega(\sqrt{n})$ and Diochnos, Mahloujifar, and Mahmoody [MM17, MDM18, MMM19] proved a weaker bound of $\mu + \Omega(k/n)$.
- **Computational isoperimetry in product spaces.** Let $\mathbf{w}_{\leq n} \equiv (\mathbf{w}_1 \times \cdots \times \mathbf{w}_n)$ be a product distribution of dimension n, and let HD be the Hamming distance $\text{HD}(w_{\leq n}, w'_{\leq n}) = |\{i \mid w_i \neq w'_i\}|$. Then, a basic question in functional analysis is how quickly noticeable events expand under Hamming distance. It is known, e.g., by results implicit in [AM80, MS86] and explicit in [McD89, Tal95][5] that if a set $\mathcal{S}$ has measure μ, the k-expansion of it (i.e., the set of points with a neighbor in $\mathcal{S}$ of distance at most k) will have have measure at least $\mu + \Omega(k \cdot \mu/\sqrt{n})$ for sufficiently large $k = \Omega(\sqrt{n/\mu})$. The works of [MM19, EMM20] introduced an *algorithmic* variant of the measure concentration phe-

[5] A weaker version for uniform bits is known as the blowing-up lemma [Mar74].

nomenon and obtained polynomial time algorithms that given a random point $w_{\leq n} \in \mathbf{w}$, it can *find* a neighbor of distance at most k in $\mathcal{S}$ with probability $\mu + \Omega(k \cdot \mu/\sqrt{n})$.

Their result above only apply to the setting where $k \geq \Omega(\sqrt{n/\mu})$, and it remained open to obtain such computational concentration for any small $k = o(\sqrt{n/\mu})$. For such small k, the problem is perhaps more suitable to be called an *isoperimetric* problem. By applying our Theorem 1 we directly get computational isoperimetry results for any $k = O(\sqrt{n/\mu})$ in any product space. For the case of uniform random bits and probabilities corresponding to Hamming balls, our Theorem 2, using polynomial time algorithms, shows how to obtain results that match the corresponding lower bound on the vertex isoperimetry [Har66]. See Theorems 44 for a formalization.

1.2 Technical overview

Here, we describe the key ideas behind our main results of Theorems 1 and 2 at a high level.

1.2.1 Theorem 1: attacking protocols with arbitrary message length

We prove Theorem 1 by giving a novel inductive analysis (over adversary's budget k) for a variant of the attack of [MM19]. Interestingly, even though the attack of [EMM20] improves [MM19] for many-replacing regime, we are not able to build our few-replacing attacks on that of [EMM20]! We also do a modification to the [MM19] (by *always* looking at a message before changing or not changing it) that allows us to significantly improve the exact bound. Our modification of the attack of [MM19] makes the attack's description simpler and allows for sharper analysis (even in the many-replacing regime of [MM19], but that is not our focus here). In fact, that change is crucial to obtain our Theorem 2 which gives an *optimal* bounds for uniform binary messages.

Our proof of Theorem 2 is inspired by the recent work of Khorasgani et al. [KMW21] who studied 1-*replacing information-theoretic non-targeted* attacks, but we still use ideas from their work in our setting. In particular, we use a concave function as the lower bound of the success probability of our attack and use induction over the number of bits n. The exact attack and the details of our inductive proof, however, are quite different from the work of Khorasgani et al. [KMW21].

Outline. We first describe our ideas for Theorem 1 and then will do so for Theorem 2. For Theorem 1, we will first sketch the proofs of [KKR18, MM19, EMM20][6] and explain why they require $k = \Omega(k)$ replacements to give a meaningful bound. Then, we explain our new ideas that allow bypassing the barrier of $k = \Omega(k)$.

In the following, we explain our new ideas behind the proof of Theorems 1 and 2.

Why the previous attacks need $k = \Omega(\sqrt{n})$ corruptions. The targeted attacks of [MM19, KKR18, EMM20] have a similar core that make them rely on many $k = \Omega(\sqrt{n})$ number of corruptions to achieve bias towards 1. These attacks first show that certain specific attacks with *unlimited* budget can significantly bias the output of the function towards 1. Then, in the second step, they show that the number of corruptions of such ∞-replacing attacks will not be more than $O(\sqrt{n})$. To contrast our approach, the analysis of our attack for proving Theorem 1 starts from $k = 1$ and increases k, while those of [MM19, KKR18, EMM20] start from $k = \infty$ and show that it does not have to be more than $k = \Theta(\sqrt{n})$.

[6] In case [KKR18], here we refer to their proof for the case of bitwise messages. Their attack for the long-message setting is (inherently) an non-targeted attack, and not a PPT one.

Notation. Let w_i be the i'th message sent by the i'th party, and let v_i be the possible modified version ($v_i \neq w_i$ if the adversary corrupts the i^{th} party and changes its message). We let $w_{\leq i} = (w_1, \ldots, w_i)$ and $v_{\leq i}$ is defined similarly. Let $f(v_1, \ldots, v_n) = b$ be the Boolean function that determines the final output bit b. Also $\mu = \Pr[b = 1]$ holds in the original (no-attack) protocol. (See Section 2 for all the definitions.)

The attack below summarizes the approach of [MM19, KKR18, EMM20]. At a high level, it tracks the expected value $\bar{f}(v_{\leq i-1}) = \Pr[b = 1 \mid v_{\leq i-1}]$ of the final bit b conditioned on the current messages $v_{\leq i-1}$ (which forms a Doob martingale).

Construction 3 (The attacks of [MM19, KKR18, EMM20])**.** Let w_i be the honestly prepared message of the i'th party that is about to be sent in round i. If the number of corruptions has not reached k yet, with the attack parameter $\lambda \in [0, 1]$, do as follows.

1. Even before looking at w_i, if there is *some* v_i that increases the expected value of b by λ (i.e., $\bar{f}(v_{\leq i}) > \bar{f}(v_{\leq i-1}) + \lambda$) then corrupt the i'th party and send v_i instead.
2. Otherwise, look at w_i. If, it is going to decrease the expected value of b by more than λ (i.e., $\bar{f}(v_{\leq i}, w_i) < \bar{f}(v_{\leq i-1}) - \lambda$), then again corrupt message w_i to v_i.
3. Otherwise, do not corrupt the i'th party, and let $v_i = w_i$ remain unchanged.

Analysis of the attack of Construction 3. The main two ideas are as follows.

1. Ignoring the number of corruptions, the ∞-replacing attack achieves expected value $1 - \text{err}(\lambda, \mu, n)$, where $\text{err}(\lambda, \mu, n) = e^{-\Omega(\mu^2/(n\lambda^2))}$ is an "Azuma error".
2. For every corruption, the expected value of the output jumps up by at least λ.

Relying on the above two keys, it can be proved that the total expected number of corruptions cannot be larger than $1/\lambda$, so by choosing $\lambda \approx \mu/\sqrt{n}$, they can achieve both (1) high expected value $1 - \text{err}(\lambda, \mu, n)$ and (2) few corruptions $k \leq 1/\lambda \approx \sqrt{n}/\mu$.

A candidate one-replacing targeted PPT attack. We now propose our new one-replacing attack that we will analyze using new ideas. The first version of our attack follows the attack of Construction 3 and immediately stops as soon as the first corruption happens. We then show how to combine steps 2 and 3 of the attack to further improve it. We emphasize that even for the basic version of our attack, the previous analysis by [MM19] does not say anything about the power of a 1-replacing variant, as this attack is cut prematurely for the analysis of [MM19] to go through.

Idea 1: we gain as soon as the corruption happens. Our first key idea is that, the additive attack of [MM19] (as opposed to the "multiplicative" attack of [EMM20]) *always* gains by λ, whenever a corruption happens. So, to analyze our 1-replacing attacks, all we need is to lower bound the probability p_1 of one corruption.

Idea 2: 1-replacing is as good as ∞-replacing if no corruptions happens. As long as no corruption has happened, our one-replacing attacker is actually *identical* to an attack with no limit on the number of corruptions. Also, note that the probability of outputting 1 in the ∞-replacing attack of [MM19] is $1 - \text{err}(\lambda, \mu, n)$. Therefore, we conclude that if we run the one-replacing attack, with probability $1 - \text{err}(\lambda, \mu, n)$ we *either* output 1 (which is good enough) *or* do at least one corruption (which is also good for us!). Since the probability of outputting 1 *without* any attacks is exactly μ, we can now lower bound p_1 and conclude that

$$p_1 \geq 1 - \text{err}(\lambda, \mu, n) - \mu.$$

Having the above bound on p_1, we lower bond output's expected value μ_1 under our 1-replacing attack is

$$\mu_1 \geq \mu + \lambda \cdot (1 - \text{err}(\lambda, \mu, n) - \mu).$$

We can now choose $\lambda = \Theta(\mu/\sqrt{n})$ which leads to up bias $\Omega(\mu/\sqrt{n})$. This attack can be made polynomial time by approximating output's Doob martingale.

Induction on k to obtain k-replacing targeted attack. Having the 1-replacing attack above, it is now tempting to apply them recursively to get k-replacing attacks. Note that this is possible only because we have a *targeted* attack, and so we can recursively apply such attack k times, each of which is a one-replacing attack, and increase the expected value of the output bit gradually. This approach, however, remains polynomial time only for $k = O(1)$. Here, we take a different approach and directly analyze the k-replacing attack of [MM19] using induction on k.

The idea is to allow the ∞-replacing attack of [MM19] run for k corruptions in total rather than one, and then trying to analyze it by induction on k. Suppose p_k is the probability that the k-replacing attack reaches its k'th corruption. Also, let μ_i be the expected value of the output b under the i-replacing targeted attack. A key idea is that all we have to do is to lower bound the probability of the corruptions happening, and by linearity of expectation we will indeed gain by at least $\lambda \cdot k$ in expected value of the outcome. In fact, we go one step further and relate the gain in the k'th corruption directly to the gain already obtained through $k-1$ corruptions. I.e., by linearity of expectation, we have:

$$\mu_k \geq \mu_{k-1} + \lambda \cdot p_k.$$

The intuition is that before reaching the k'th corruption, the two attack are the same, and once the k'th corruption happens, the k-replacing attack gets a jump of λ up compared to the $(k-1)$-replacing attack. Again, all we need is to lower bound p_k. To do so, we again use a generalization of the idea that we described for the case of one-replacing above. Namely, we note that as long as the k'th corruption does not happen in the k-replacing attack, it is again *indistinguishable* from the ∞-replacing attack of [MM19]. Also, the $(k-1)$-replacing attack reaches $b = 1$ with probability μ_{k-1} already. Using a union bound, we get:

$$p_k \geq 1 - \mathrm{err}(\lambda, \mu, n) - \mu_{k-1},$$

using which we can get that the expected value of b under the k-replacing attack is

$$\mu_k \geq \mu + \lambda \cdot (1 - \mathrm{err}(\lambda, \mu, n) - \mu_{k-1}).$$

Solving the recursive inequalities above, we lower bound μ_k as in Theorems 1 and 15.

1.2.2 Theorem 2: optimal attacks for uniform binary messages

We now describe some of the key ideas behind our proof of Theorem 2, which deals with uniform binary messages. In this section, we mainly focus on showing the core ideas that lead to the information theoretic optimal k-replacing attacks of Theorem 2, which deals with online attacks. In Section 4.1we show how to use similar ideas (by approximating the Doob martingale of the final output bit) used for the polynomial-time attacks for Theorem 1 to also extend our information theoretic attacks for Theorem 2 to polynomial time variants.

Notation. First, we define the key notations that are needed for our overview of the ideas behind the proof of our Theorem 2. Here, all the original messages are *independent and uniform* random bits, which we denote with $(u_1, \ldots, u_n)$. Also, we let $\mathcal{S}$ be the set of input sequences that lead to output 1, namely $\mathcal{S} = \{x \mid f(x) = 1\}$. We know that $\Pr[(u_1, \ldots, u_n) \in \mathcal{S}] = \Pr[\sum u_i \geq t] = \beta_n^{(t)}$ is that of a Hamming ball. The goal of the adversary is to maximize the probability of falling into $\mathcal{S}$ through k-replacements in an online

way. We now define the "online expansion" under optimal online k-replacing attacks, both as a function of sets, or as a function of set probabilities. (See Definition 25 for more details.) Let A be an online k-replacing adversary over the uniform distribution over $\{0,1\}^n$. Let $\mathsf{OnExp}^{(\mathsf{A})}(\mathcal{S})$ be the probability that A can map a random input to $\mathcal{S}$ through its online k-replacing attack. Let $\mathsf{OnExp}^{(k)}(\mathcal{S})$ be the maximum over $\mathsf{OnExp}^{(\mathsf{A})}(\mathcal{S})$ among all k-replacing attacks, and let the following be the minimum of $\mathsf{OnExp}^{(k)}(\mathcal{S})$ among all sets of measure μ.

$$\mathsf{OnExp}_n^{(k)}(\mu) = \inf_{\mathcal{S},\Pr[\mathcal{S}]\geq\mu} \mathsf{OnExp}^{(k)}(\mathcal{S}).$$

Our key idea is to show that the following piecewise-linear function is a *lower bound* on the power of k-replacing attacks. We prove this by induction on n. In comparison, [KMW21] also used similar piecewise-linear functions, but their goal was to obtain *1-replacing information-theoretic non-targeted* attacks. It is possible that using similar techniques, one can make the attack of [KMW21] also polynomial time, but the key differences are due to the fact that [KMW21] aims for a non-targeted attack, and hence it ends up with a completely different recursive relation and induction on n.

Definition 4 (The piecewise-linear lower bound – informal)**.** For any non-negative integers k, n, the function $\ell_n^{(k)}\colon [0,1] \to [0,1]$ is defined as follows.

- If $\mu = \beta_n^{(t)}$ for any $t \in [n]$, it holds that $\ell_n^{(k)}\left(\beta_n^{(t)}\right) = \beta_n^{(t-k)}$. Namely, when the input probability is that of an exact Hamming balls, $\ell_n^{(k)}$ returns their probability after expanding them to include anything within their k Hamming distance (which is also a Hamming ball).
- Connect all the $n+2$ points above to obtain a piecewise-linear function $\ell_n^{(k)}$.

See Definition 32 for a formal definition of the function above.

Recursive relation for $\mathsf{OnExp}_n^{(k)}(\mu)$. We then use a recursive relation that can be used to exactly compute $\mathsf{OnExp}_n^{(k)}(\mu)$ for all k, n, μ (see Definition 28). The idea of the recursive relation is to model adversary's decision based on optimal decisions. In fact, if an adversary is given a bit $u_i = 0$, and it holds that $\Pr[(0,u_2,\ldots,u_n) \in \mathcal{S}] = \mu_0, \Pr[(1,u_2,\ldots,u_n) \in \mathcal{S}] = \mu_1$. Then, an optimal online adversary shall decide between changing it to 1 or not, and if it knows the optimal solutions for $\mathsf{OnExp}_{n-1}^{(k)}(\mu_0)$ (reflecting the "no change" decision) and $\mathsf{OnExp}_{n-1}^{(k-1)}$ (reflecting the "change" decision) it can make the optimal decision.

Using lower bounds lead to lower bounds. We prove by induction on n, that if one uses *lower bounds* (e.g., $\ell_{n-1}^{(k)}$ and $\ell_{n-1}^{(k-1)}$) instead of $\mathsf{OnExp}_{n-1}^{(k)}(\mu_0)$ and $\mathsf{OnExp}_{n-1}^{(k-1)}(\mu_1)$ in the recursive relation that computes $\mathsf{OnExp}_n^{(k)}$, then one obtains a lower bound on $\mathsf{OnExp}_n^{(k)}(\mu)$. This part of the proof follows from the monotonicity of the recursive relation for $\mathsf{OnExp}_n^{(k)}$.

Function $\mathsf{OnExp}_n^{(k)}(\mu)$ *remains a lower bound for* $\ell_n^{(k)}$. We also show that when we apply the recursive relation over $\ell_{n-1}^{(k)}$ and $\ell_{n-1}^{(k-1)}$, the result will be an *upper bound* on $\ell_n^{(k)}$. This, together with the step above implies that $\ell_n^{(k)}$ remains a lower bound $\ell_n^{(k)}$. This is the most technical step of the proof that goes through a careful case study and heavily relies on the concavity and monotonicity of $\ell_n^{(k)}$.

Making the attack polynomial time. In the actual polynomial time attack, the adversary approximates μ, and it uses $\ell_n^{(k)}$ (which is efficiently computable) instead of $\mathsf{OnExp}_n^{(k)}(\mu)$ in the recursive relation and decides to change or not change the bits. See Section 4.1 for more details.

1.3 Further related work

Many of the related works were already discussed in previous sections. In this section, we discuss other works related to ours, mostly in the context of coin tossing protocols.

Adaptive corruption. As explained above, our results are proved in the *strong* adaptive corruption model. However, many works study the power of standard adaptive corruption in coin tossing protocols. The main result in [KKR18] indeed proves the existence of such attacks that achieve *non-targeted* biasing that controls the output fully when the number of corruptions is $k \geq \sqrt{n}$. Haitner and Karidi-Heller [HKH20] further generalized this result to multi-turn protocols, resolving a long-standing open problem of Ben-Or and Linial [BOL90]. Dodis [Dod01] previously proved that certain black-box methods cannot break this conjecture. The recent work of Khorasgani, Maji, and Wang [KMW21, KMM19] showed that for the case of 1 replacing, (computationally unbounded) adaptive adversaries can achieve *non-targeted* bias $\Omega(1/\sqrt{n})$ in single-turn protocols.

Static corruption. A static adversary chooses the corrupted parties independently of the execution of the protocol, and hence can fix the corrupted set ahead of the execution. The previously mentioned works of [BOL89, BGZ16, MM17, MDM18, MMM19] all fall into this framework and prove that corrupting k parties can lead to bias $\Omega(\mu k/n)$ statically. These results hold even if the statically corrupted set is chosen *at random*. For single-round protocols in which each party sends a single bit, Kahn, Kalai and Linial [KAH88] showed that any protocol is susceptible to $\Omega(n/\log n)$ corruptions. A long line of exciting works (see [RSZ02]) showed how to achieve robustness to $(1-\delta)\cdot n$ static corruption for any $\delta < 1$.

Fair coin tossing. Another line of work in coin tossing protocols aims to study the power of *fair* protocols in which the parties *need* to output a bit even if the other party is caught cheating (e.g., by aborting in the middle of the protocol). The work of Cleve [Cle86] showed that in any such protocol with r rounds between two parties, there is a PPT attacker that biases the output of the other party by at least $\Omega(1/r)$. The work of Moran, Naor, and Segev [MNS09] showed how to *match* this bound assuming oblivious transfer, leading to an "optimally fair" protocol. A sequence of works [DSLMM11, DMM14, HNO+18, HMO18] showed barriers for doing so from one-way functions, and finally, the beautiful work of Maji and Wang [MW20] completely resolved this question for black-box constructions. For works on fair coin tossing in the multi-party settings see [HT17, BHMO18].

2 Preliminaries

General notation. We use calligraphic letters (e.g., $\mathcal{X}$) for sets. All distributions and random variables in this work are discrete. We use bold letters (e.g., $\mathbf{w}$) to denote random variables that return a sample from a corresponding discrete distribution. By $w \leftarrow \mathbf{w}$ we denote sampling w from the random variable $\mathbf{w}$. By $\mathrm{Supp}(\mathbf{w})$ we denote the support set of $\mathbf{w}$. For an event $\mathcal{S} \subseteq \mathrm{Supp}(\mathbf{w})$, the probability function of $\mathbf{w}$ for $\mathcal{S}$ is denoted as $\Pr[\mathbf{w} \in \mathcal{S}] = \Pr_{w \leftarrow \mathbf{w}}[w \in \mathcal{S}]$ or simply as $\Pr[\mathcal{S}]$ when $\mathbf{w}$ is clear from the context. By $\mathbf{u} \equiv \mathbf{v}$ we denote that the random variables $\mathbf{u}$ and $\mathbf{v}$ have the same distributions. Unless stated otherwise, we denote vectors by using a bar over a variable. By $(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$ we refer to a sequence of n *jointly sampled* random variables. For a vector $(w_1 \ldots w_n)$, we use $w_{\leq i}$ to denote the prefix $(w_1, \ldots, w_i)$, and we use the same notation $\mathbf{w}_{\leq i}$ for jointly distributed random variables. For vector $x = u_{\leq i-1}$ and $y = u_i$, by, by xy we denote the vector $u_{\leq i-1}$ that appends u_i as the last coordinate of x. For a jointly distributed random variables $(\mathbf{u}, \mathbf{v})$, by $(\mathbf{u} \mid \mathbf{v} = v)$ or we denote the random variable $\mathbf{u}$ conditioned on $\mathbf{v} = v$. When it is clear from the context, we simply write $(\mathbf{u} \mid v)$ or $\mathbf{u}[v]$ instead. By $\mathbf{u} \times \mathbf{v}$ we refer to the product distribution in

which $\mathbf{u}$ and $\mathbf{v}$ are sampled independently. $\mathrm{HD}(u_{\le n}, v_{\le n}) = |\{i \mid u_i \neq v_i\}|$ denotes the Hamming distance for vectors of n coordinates.

Random processes. Let $\mathbf{w}_{\le n} \equiv (\mathbf{w}_1, \ldots, \mathbf{w}_n)$ be a sequence of jointly distributed random variables. We can interpret the distribution of $\mathbf{w}_{\le i}$ as a random process in which the i^{th} block w_i is sampled from the marginal distribution $(\mathbf{w}_i \mid w_{\le i-1}) \equiv (\mathbf{w}_i \mid \mathbf{w}_{\le i-1} = w_{\le i-1}) \equiv \mathbf{w}_i[w_{\le i-1}]$. We also use $\mathbf{w}_{\le n}[\cdot]$ to denote an oracle sampling algorithm that given $w_{\le i}$ returns a sample from $\mathbf{w}_{\le n}[w_{\le i}]$.

Attack model. Our adversaries *replace* a message/block in a random process. Namely, they observe the blocks one by one and sometimes intervene to replace them with a new value. (The new values will subsequently change the way the random process will proceed.) Hence, we refer to them as *replacing* adversaries. Such adversaries are equivalent to *strongly adaptive corrupting* adversaries as defined in [GKP15].

Definition 5 (Online replacing attacks on random processes)**.** Let $\mathbf{w}_{\le n} \equiv (\mathbf{w}_1, \ldots, \mathbf{w}_n)$ be a random process. Suppose $\mathsf{A}(x, \sigma) \to (x', \sigma')$ is a (potentially randomized) algorithm with the following syntax. It takes as input some (randomness,) x and σ, where σ is interpreted as a "state", and it outputs (x', σ'). We call such algorithm an *online replacing adversary* and define the following properties for it.

We define the following notions for $\mathbf{w}_{\le n}$.

- **The generated and output random processes under replacing attacks.** Suppose A is an replacing algorithm. We now define two random processes that result from running the replacing adversary A to influence the original random process $\mathbf{w}_{\le n}$. For $i = 1, 2, \ldots, n$, we first sample $u_i \leftarrow (\mathbf{w}_i \mid \mathbf{w}_{\le i-1} = v_{\le i-1})$, and then we obtain $(v_i, \sigma_i) \leftarrow \mathsf{A}(u_i, \sigma_{i-1})$. If at any point during this process $\Pr[\mathbf{w}_{\le i} = v_{\le i}] = 0$, we will output $u_{i+1} = \cdots = u_n = v_{i+1} = \cdots = v_n = \perp$. We call $(\mathbf{u}_{\le n}, \mathbf{v}_{\le n})$ the *jointly generated* random processes under the attack. We also refer to $u_{\le n}$ as the *original values* and $v_{\le n}$ as the *output* of the random process under the attack A.

- **Online replacing.** We call A a *valid* (online replacing) attack on $\mathbf{w}_{\le n}$, if with probability 1 over the generation of $u_{\le n}, v_{\le n}$, it holds that none of the coordinates are $\perp$ (i.e., $\Pr[\mathbf{w}_{\le i} = v_{\le i}] \neq 0$.) In this work we always work with valid online replacing attacks, even if they are not called valid.

- **Budget of replacing attacks.** Replacing adversary A has *budget* k, if

$$\Pr[\mathrm{HD}(\mathbf{u}_{\le n}, \mathbf{v}_{\le n}) \le k] = 1,$$

  where $(\mathbf{u}_{\le n}, \mathbf{v}_{\le n})$ are the jointly generated random processes that are also jointly distributed.

- **Algorithmic efficiency of attacks.** If $\mathbf{w}_{\le n}$ is indexed by n as a member of a *family* of joint distributions defined for all $n \in \mathbb{N}$, then we call an online or offline replacing algorithm *efficient*, if its running time is at most $\mathrm{poly}(N)$ where N is the total bit-length representation of any $w_{\le n} \in \mathrm{Supp}(\mathbf{w}_{\le n})$. We would also consider efficiency where the replacing algorithm uses an oracle. In particular, we say an attack $\mathsf{A}^{\mathbf{w}_{\le n}[\cdot]}$ with oracle access to sampler $\mathbf{w}_{\le n}[\cdot]$ is efficient if it runs in time $\mathrm{poly}(N)$.

We now recall the so-called Doob martingale of a (Boolean-output) random process.

Definition 6 (Doob martingale, partial averages, and their approximate variant)**.** For random process $\mathbf{w}_{\le n} \equiv (\mathbf{w}_1, \ldots, \mathbf{w}_n)$, let $f\colon \mathrm{Supp}(\mathbf{w}_{\le n}) \mapsto \mathbb{R}$, $i \in [n]$, and $w_{\le i} \in \mathrm{Supp}(\mathbf{w}_{\le i})$. Then we use the notation $\bar{f}(w_{\le i}) = \mathbb{E}_{w_{\le n} \leftarrow (\mathbf{w}_{\le n} | w_{\le i})}[f(w_{\le n})]$ to define the expected value of f for a sample from $\mathbf{w}_{\le n}$ conditioned on the prefix $w_{\le i}$ and refer to it as a *partial-average* of f. In particular, using notation $w_{\le 0} = \varnothing$, we have $\bar{f}(\varnothing) = \mathbb{E}[f(\mathbf{w}_{\le n})]$. The random process $(\bar{f}(\mathbf{w}_{\le 1}), \ldots, \bar{f}(\mathbf{w}_{\le n}))$ is called the Doob martingale of the function f over the random process $\mathbf{w}_{\le n}$. For the same $\mathbf{w}_{\le n}$ and $\bar{f}(\cdot)$, we call $\tilde{f}(\cdot)$ an (additive) ε-approximation of $\bar{f}(\cdot)$, if for all $w_{\le i} \in \mathrm{Supp}(\mathbf{w}_{\le i})$, it holds that $\tilde{f}(w_{\le i}) \in \bar{f}(w_{\le i}) \pm \varepsilon$.

If one is given oracle access to ℓ samples from $(\mathbf{w}_i \mid w_{\leq i})$, then by averaging them, one can obtain (due to the Hoeffding inequality) an ε-approximation of $\tilde{f}(w_{\leq i})$ for with probability $1 - \exp(-\ell/\varepsilon^2)$.

2.1 Useful facts

We use the following variant of the Azuma inequality which is proved in [HMRAR13].

Lemma 7 (Azuma's inequality for dynamic interval lengths (Theorem 2.5 in [HMRAR13]))**.** *Let $\mathbf{t}_{\leq n} \equiv (\mathbf{t}_1, \ldots, \mathbf{t}_n)$ be a sequence of n jointly distributed random variables such that for all $i \in [n]$, and for all $t_{\leq i-1} \leftarrow \mathbf{t}_{\leq i-1}$, we have*

$$\exists t^*, \quad \Pr_{t_i \leftarrow \mathbf{t}_i | t_{\leq i-1}} [t^* + \eta_i \geq t_i \geq t^* - \eta_i] = 0$$

and $\mathbb{E}[\mathbf{t}_i \mid t_{\leq i-1}] \geq 1$. Then, we have

$$\Pr\left[\sum_{i=1}^{n} \mathbf{t}_i \leq -s\right] \leq \mathrm{e}^{\frac{-s^2}{2\sum_{i=1}^{n} \eta_i^2}}$$

Lemma 8 (Azuma's inequality for dynamic interval lengths under approximate conditions)**.** *Let $\mathbf{t}_{\leq n} \equiv (\mathbf{t}_1, \ldots, \mathbf{t}_n)$ be a sequence of n jointly distributed random variables such that for all $i \in [n]$, and for all $t_{\leq i-1} \leftarrow \mathbf{t}_{\leq i-1}$, we have*

$$\exists t^*, \quad \Pr_{t_i \leftarrow \mathbf{t}_i | t_{\leq i-1}} [|t_i| \geq 1] = 0$$

$$\exists t^*, \quad \Pr_{t_i \leftarrow \mathbf{t}_i | t_{\leq i-1}} [t^* + \eta_i \geq t_i \geq t^* - \eta_i] \geq 1 - \gamma$$

and $\mathbb{E}[\mathbf{t}_i \mid t_{\leq i-1}] \geq -\gamma$. Then, we have

$$\Pr\left[\sum_{i=1}^{n} \mathbf{t}_i \leq -s\right] \leq \mathrm{e}^{\frac{-(s-2n\gamma)^2}{2\sum_{i=1}^{n} \eta_i^2}} + n \cdot \gamma$$

Proof. If we let $\gamma = 0$, Lemma 8 becomes equivalent to Lemma 7. Here we sketch why Lemma 8 can also be reduced to the case that $\gamma = 0$ (i.e., Azuma inequality). We build a sequence $\mathbf{t}'_i$ from $\mathbf{t}_i$ as follows: Sample $t_i \leftarrow \mathbf{t}_i \mid t_{\leq i-1}$, if $|t_i - t^*| \leq \eta_i$, output $t'_i = t_i + 2\gamma$ otherwise output $t^* + 2\gamma$. We have $\mathbb{E}[\mathbf{t}'_i \mid t'_{\leq i-1}] \geq 0$ and $\Pr[|t'_i - t^* - 2\gamma| > \tau_i] = 0$. Now we can use Lemma 8 for the basic case of $\gamma = 0$ for the sequence $\mathbf{t}'_i$ and use it to get a looser bound for sequence $\mathbf{t}_i$, using the fact that $\exists i \in [n], |t_i - t^*| \geq \eta_i$ happens with probability at most $n \cdot \gamma$. □

Lemma 9 (Composition of concave functions)**.** *Suppose ℓ_1 and ℓ_2 are two non-decreasing concave functions. Then $\ell_1(\ell_2)$ is also non-decreasing and concave.*

3 Attacking protocols with arbitrary message length

In this section, we design and analyze our k-replacing up-biasing attack on random processes with arbitrary alphabet size. We first describe our attack in an idealized model in which the partial-average oracle $\bar{f}(\cdot)$ and "maximum child" of a prefix of the process are available for free. In Section 3.1, we show that our attack can be made polynomial-time using an approximation of the partial-average oracle that can be obtained in polynomial time.

Construction 10 (k-replacing attack using exact oracles)**.** This attack uses the exact partial-average oracle $\bar{f}(\cdot)$ and another oracle that returns "the best choice" for the next block (see u^*_{i+1} defined below). The attack is also parameterized by a vector $\lambda_{\le k} = (\lambda_1, \ldots, \lambda_k) \in [0,1]^k$ for some integer $k \le n$ which is adversary's budget. The attack will keep state $\sigma_i = (u_{\le i}, v_{\le i})$ where $u_{\le i}$ are the original values and $v_{\le i}$ are the output values under attack.[7] Having state $(u_{\le i}, v_{\le i})$ and for given u_{i+1} the algorithm A will decide on whether to keep or replace u_{i+1}, using $u^*_{i+1} = \mathrm{argmax}_{u'_{i+1}} \bar{f}(v_{\le i}, u'_{i+1})$, $\bar{f}^* = \bar{f}(v_{\le i}, u^*_{i+1})$, and $d = \mathrm{HD}(u_{\le i}, v_{\le i})$ as follows.

- (Case 0) If $d \ge k$, do not change u_{i+1} and output $v_{i+1} = u_{i+1}$.
- (Case 1) if Case 0 does not happen and $\bar{f}(v_{\le i}, u_{i+1}) < \bar{f}^* - \lambda_{d+1}$, then $\mathsf{A}[\lambda_{\le k}](u_{i+1})$ will return the output $v_{i+1} = u^*_{i+1}$ which is different from u_{i+1}.
- (Case 2) If Cases 0, 1 do not happen, do not change u_{i+1} and output $v_{i+1} = u_{i+1}$.

In all the cases above, A will also update the state as $\sigma_{i+1} = (u_{\le i+1}, v_{\le i+1})$.

Notation. Suppose we run the attack $\mathsf{A}[\lambda_{\le k}]$ on random process $\mathbf{w}_{\le n}$ through the process described in Definition 5. (In particular, u_{i+1} will be sampled from $(\mathbf{w}_{i+1} \mid \mathbf{w}_{\le i} = v_{\le i})$.) We use $(\mathbf{u}^{(k)}_{\le n}, \mathbf{v}^{(k)}_{\le n})$ to denote the jointly generated random processes under the attack $\mathsf{A}[\lambda_{\le k}]$. (This notation allows us to distinguish between the generated random processes under attacks with different budget.) We sometimes use $(\mathbf{u}^{(\infty)}_{\le n}, \mathbf{v}^{(\infty)}_{\le n})$ to denote $(\mathbf{u}^{(n)}_{\le n}, \mathbf{v}^{(n)}_{\le n})$ as they are the same distributions. Also, let

$$\mu_k = \mathop{\mathbb{E}}_{(u_{\le n}, v_{\le n}) \leftarrow (\mathbf{u}^{(k)}_{\le n}, \mathbf{v}^{(k)}_{\le n})} [f(v_{\le n})]$$

denotes the expected value of f over the sequence that is the output of k-replacing attack of Construction 10. For $k = 0$ we have and $\mu_0 = \mu = \mathbb{E}[f(\mathbf{w}_{\le n})]$.

Lemma 11 below shows that the increase in μ_k compared with μ_{k-1} can be related to the "threshold parameter" λ_k and the probability that an attack with *unlimited* (or equivalently just n) budget with threshold parameters $\lambda_1, \ldots, \lambda_k, \lambda'_{k+1}, \ldots, \lambda'_n$ makes at least k replacements.

Lemma 11. *We have*

$$\mu_k \ge \mu_{k-1} + \lambda_k \cdot \Pr_{(u_{\le n}, v_{\le n}) \leftarrow (\mathbf{u}^{(\infty)}_{\le n}, \mathbf{v}^{(\infty)}_{\le n})} [\mathrm{HD}(u_{\le n}, v_{\le n}) \ge k].$$

Proof. For any $j \in \{0,1,2\}$, let C^k_j be the Boolean random variable over (u_{i+1}, σ_i) that determines which case of the attack A with budget k happens on prefix $(v_{\le i}, u_{i+1})$ where $v_{\le i}$ is the finalized output prefix, $u_{\le i}$ is the original prefix and u_{i+1} is the original sampled block at round $i+1$. For all $(v_{\le i}, u_{\le i}, u_{i+1})$ we have $\sum_{j=0}^{2} C^k_j(u_{i+1}, \sigma_i) = 1$ because the cases complement each other.

In the rest of the proof, whenever $u_{\le i}$ and $v_{\le i}$ are clear from the context, we will use $C^k_j(u_{i+1})$ instead of $C^k_j(u_{i+1}, \sigma_i)$. In the following, when the threshold parameters $\lambda_1, \ldots, \lambda_k$ are clear from the context, we will use A instead of $\mathsf{A}[\lambda_{\le k}]$.

For all $u_{\le i}, v_{\le i} \in \mathrm{Supp}(\mathbf{u}_{\le i}, \mathbf{v}_{\le i})$ we have the following qualities for different cases of the attack.

[7] Attack would need $v_{\le i}$ and the "used part of the budget" $\mathrm{HD}(u_{\le i}, v_{\le i})$. Both of these can be obtained from $\sigma_i = (u_{\le i}, v_{\le i})$.

- Case 0:

$$\mathop{\mathbb{E}}_{(u_{i+1},v_{i+1})\leftarrow(\mathbf{u}^k_{i+1},\mathbf{v}^k_{i+1})[u_{\le i},v_{\le i}]}\left[\left(\bar{f}(v_{\le i},v_{i+1})-\bar{f}(v_{\le i},u_{i+1})\right)\cdot C^k_0(u_{i+1})\right]=0. \tag{1}$$

- Case 1:

$$C^k_1(u_{i+1})=(C^\infty_1(u_{i+1})\wedge \mathrm{HD}(u_{\le i},v_{\le i})<k). \tag{2}$$

  This is because as long as the number of replacements is fewer than k, Case 1 of the attack with budget k would go through whenever A with budget of n does so.

- Case 2:

$$\mathop{\mathbb{E}}_{(u_{i+1},v_{i+1})\leftarrow(\mathbf{u}^k_{i+1},\mathbf{v}^k_{i+1})[u_{\le i},v_{\le i}]}\left[\left(\bar{f}(v_{\le i},v_{i+1})-\bar{f}(v_{\le i},u_{i+1})\right)\cdot C^k_2(u_{i+1}))\right]=0. \tag{3}$$

  This is correct because either $C^k_2(v_{\le i},u_{i+1})=0$ or $u_{i+1}=v_{i+1}$.

We define a notation $g(v_{\le i+1},u_{\le i+1})=\bar{f}(v_{\le i+1})-\bar{f}(v_{\le i},u_{i+1})$. In the following We use the shorten forms of $\mathbb{E}_{(\mathbf{u}_{\le i},\mathbf{v}_{\le i})}$ and $\mathbb{E}_{(\mathbf{u}_{\le n},\mathbf{v}_{\le n})[u_{\le i},v_{\le i}]}$ to refer to $\mathbb{E}_{(u_{\le i},v_{\le i})\leftarrow(\mathbf{u}_{\le i},\mathbf{v}_{\le i})}$ and $\mathbb{E}_{(u,v)\leftarrow(\mathbf{u}_{\le n},\mathbf{v}_{\le n})[u_{\le i},v_{\le i}]}$. We have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\le n},\mathbf{v}^{(k)}_{\le n})}[f(v_{\le n})]-\mu=\mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\le n},\mathbf{v}^{(k)}_{\le n})}\left[\sum_{i=0}^{n-1}(\bar{f}(v_{\le i+1})-\bar{f}(v_{\le i}))\right]\\
&=\mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\le n},\mathbf{v}^{(k)}_{\le n})}\left[\sum_{i=0}^{n-1}(\bar{f}(v_{\le i+1})-\bar{f}(v_{\le i},u_{i+1}))\right]\quad\text{(by the definition of }\bar{f}) &(4)\\
&=\sum_{i=0}^{n-1}\mathop{\mathbb{E}}_{(\mathbf{u}^k_{\le i},\mathbf{v}^k_{\le i})}\mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\le n},\mathbf{v}^{(k)}_{\le n})[u_{\le i},v_{\le i}]}\left[g(v_{\le i+1},u_{\le i+1})\cdot\left(\sum_{j=0}^{2}C^k_j(u_{i+1})\right)\right]\\
&=\sum_{i=0}^{n-1}\mathop{\mathbb{E}}_{(\mathbf{u}^k_{\le i},\mathbf{v}^k_{\le i})}\mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\le n},\mathbf{v}^{(k)}_{\le n})[u_{\le i},v_{\le i}]}\left[g(v_{\le i+1},u_{\le i+1})\cdot C^k_1(u_{i+1})\right]\quad\text{(by (3) and (1))} &(5)\\
&=\sum_{i=0}^{n-1}\mathop{\mathbb{E}}_{(\mathbf{u}^k_{\le i},\mathbf{v}^k_{\le i})}\mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\le n},\mathbf{v}^{(k)}_{\le n})[u_{\le i},v_{\le i}]}\left[g(v_{\le i+1},u_{\le i+1})\cdot(C^{(\infty)}_1(u_{i+1})\wedge(\mathrm{HD}(u_{\le i},v_{\le i})<k)\right]\quad\text{(by (2))}\\
&=\sum_{i=0}^{n-1}\mathop{\mathbb{E}}_{(\mathbf{u}^\infty_{\le i},\mathbf{v}^\infty_{\le i})}\mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\le n},\mathbf{v}^{(k)}_{\le n})[u_{\le i},v_{\le i}]}\left[g(v_{\le i+1},u_{\le i+1})\cdot(C^\infty_1(u_{i+1})\wedge(\mathrm{HD}(u_{\le i},v_{\le i})<k))\right]. &(6)
\end{aligned}$$

The last equality above holds, because for all $u_{\le i},v_{\le i}$ where $\mathrm{HD}(u_{\le i},v_{\le i})<k$,

$$\Pr[(\mathbf{u}^k_{\le i},\mathbf{v}^k_{\le i})=(u_{\le i},v_{\le i})]=\Pr[(\mathbf{u}^{(\infty)}_{\le i},\mathbf{v}^{(\infty)}_{\le i})=(u_{\le i},v_{\le i})].$$

The reason for this is that as long as we have not used the full budget k, the k-replacing attack will behave as if its budget is infinite.

Similarly, for the adversary A with budget $k-1$ we have

$$\mathop{\mathbb{E}}_{(\mathbf{u}_{\le n}^{(k-1)},\mathbf{v}_{\le n}^{(k-1)})}[f(v_{\le n})] - \mu = \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(\mathbf{u}_{\le i}^{(\infty)},\mathbf{v}_{\le i}^{(\infty)})\le i} \mathop{\mathbb{E}}_{(\mathbf{u}_{\le n}^{(k-1)},\mathbf{v}_{\le n}^{(k-1)})[u_{\le i},v_{\le i}]} [\eta(u_{\le i+1}, v_{\le i+1})]\,. \tag{7}$$

where $\eta(u_{\le i+1}, v_{\le i+1}) = g(v_{\le i+1}, u_{\le i+1}) \cdot \big(C_1^{\infty}(u_{i+1}) \wedge (\mathrm{HD}(u_{\le i}, v_{\le i}) < k-1)\big)$. Therefore, by combining Equations (6) and (7) we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{u}_{\le n}^{(k)},\mathbf{v}_{\le n}^{(k)})}[f(v_{\le n})] - \mathop{\mathbb{E}}_{(u_{\le n},v_{\le n})\leftarrow(\mathbf{u}_{\le n}^{(k-1)},\mathbf{v}_{\le n}^{(k-1)})}[f(v_{\le n})] = \\
&\sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(\mathbf{u}_{\le i}^{(\infty)},\mathbf{v}_{\le i}^{(\infty)})} \mathop{\mathbb{E}}_{(\mathbf{u}_{\le n}^{(k)},\mathbf{v}_{\le n}^{(k)})[u_{\le i},v_{\le i}]} [g(v_{\le i+1}, u_{\le i+1}) \cdot C_1^{\infty}(u_{i+1}) \cdot (\mathrm{HD}(u_{\le i}, v_{\le i}) = k-1)] \\
&\ge \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(\mathbf{u}_{\le i}^{(\infty)},\mathbf{v}_{\le i}^{(\infty)})} \left[\lambda_k \cdot \mathop{\mathbb{E}}_{(\mathbf{u}_{\le n}^{(k)},\mathbf{v}_{\le n}^{(k)})[u_{\le i},v_{\le i}]} [C_1^{\infty}(u_{i+1}) \cdot (\mathrm{HD}(u_{\le i}, v_{\le i}) = k-1)]\right] \\
&= \lambda_k \cdot \Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[\mathrm{HD}(u_{\le n}, v_{\le n}) \ge k].
\end{aligned}$$

The last equality above holds because whenever $C_1^{(\infty)}$ holds, we know that A will replace u_{i+1} with $v_{i+1} \neq u_{i+1}$ and this makes the hamming distance of $u_{\le i+1}$ from $v_{\le i+1}$ equal to k. □

Now we prove the following lemma about the power of attacks with infinite budget. Claim 19 in [MM19] also prove a similar bound for their attack but our attack achieves a better bound because of the fact that our attack has only one step in which the replacement might happen which allows us to make a better use of Azuma's inequality with dynamic interval (See Lemma 7).

Lemma 12. *If* $\mu_\infty = \mathbb{E}_{(u_{\le n},v_{\le n})\leftarrow(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[f(v_{\le n})]$ *and* $\lambda = \max_{i\in[n]} \lambda_i$*, then*

$$\mu_\infty \ge 1 - e^{-\frac{2\mu^2}{n\lambda^2}}.$$

Proof. We define a sequence of random variables $\mathbf{t}_{\le n} = (\mathbf{t}_1, \ldots, \mathbf{t}_n)$, where $t_{i+1} = \bar{f}(v_{\le i+1}) - \bar{f}(v_{\le i})$ is a random variable that is dependent on $v_{\le i+1}$. Then we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})[u_{\le i},v_{\le i}]} [\bar{f}(v_{\le i+1}) - \bar{f}(v_{\le i})] \\
&\ge \mathop{\mathbb{E}}_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})[u_{\le i},v_{\le i}]} [\bar{f}(v_{\le i}, u_{i+1}) - \bar{f}(v_{\le i})] = 0.
\end{aligned}$$

Therefore, $\mathbf{t}_{\le n}$ defines a sub-martingale. Furthermore, we have

$$\bar{f}^* \ge \bar{f}(v_{\le i+1}) \ge \bar{f}^* - \lambda.$$

Therefore, t_i always falls in an interval of size λ. Hence, applying the right variant of Azuma's Inequality (as stated in Lemma 7) over $\mathbf{t}_{\le n}$, we have

$$\Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[f(v_{\le n}) = 0] = \Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[f(v_{\le n}) - \mu \le -\mu] = \Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}\left[\sum_{i=1}^{n} t_i \le -\mu\right] \le e^{-\frac{2\mu^2}{n\lambda^2}}. \tag{8}$$

Now, leveraging the fact that f outputs in $\{0,1\}$ and relying on Inequality (8), we have

$$\Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[f(v_{\le n})=1] = 1 - \Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[f(v_{\le n})-\mu \le -\mu] \ge 1 - e^{-\frac{2\mu^2}{n\lambda^2}}.$$

□

Lemma 13. *If $\lambda = \max_{i\in[k]} \lambda_i$, then*

$$\Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[\mathrm{HD}(u_{\le n}, v_{\le n}) \ge k] \ge 1 - e^{-\frac{2\mu^2}{n\lambda^2}} - \mu_{k-1}.$$

Proof. First we have

$$\begin{aligned}
&\Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[\big(f(v_{\le n})=1 \wedge \mathrm{HD}(u_{\le n}, v_{\le n}) < k\big) \vee (\mathrm{HD}(u_{\le n}, v_{\le n}) \ge k)] \\
&= \Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[f(v_{\le n})=1 \vee \mathrm{HD}(u_{\le n}, v_{\le n}) \ge k] \\
&\ge \Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[f(v_{\le n})=1] \\
&= \mu_\infty \ge 1 - e^{-\frac{2\mu^2}{n\lambda^2}} \text{ (by Lemma 12).}
\end{aligned} \tag{9}$$

On the other hand, by a union bound we have

$$\begin{aligned}
&\Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[\big(f(v_{\le n})=1 \wedge \mathrm{HD}(u_{\le n}, v_{\le n}) < k\big) \vee (\mathrm{HD}(u_{\le n}, v_{\le n}) \ge k)] \le \\
&\Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[f(v_{\le n})=1 \wedge \mathrm{HD}(u_{\le n}, v_{\le n}) < k] + \Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[\mathrm{HD}(u_{\le n}, v_{\le n}) \ge k].
\end{aligned} \tag{10}$$

The generated process under $k-1$ replacing attack is same as n-replacing attack as long as the number of replacements is less than k. Therefore, it holds that

$$\Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[\big(f(v_{\le n})=1 \wedge \mathrm{HD}(u_{\le n}, v_{\le n}) < k\big)] \le \Pr_{(\mathbf{u}_{\le n}^{(k-1)},\mathbf{v}_{\le n}^{(k-1)})}[f(v_{\le n})=1] = \mu_{k-1}. \tag{11}$$

Now, combining Inequalities (9), (10) and (11) we get

$$\Pr_{(\mathbf{u}_{\le n}^{(\infty)},\mathbf{v}_{\le n}^{(\infty)})}[\mathrm{HD}(u_{\le n}, v_{\le n}) \ge k] \ge 1 - e^{-\frac{2\mu^2}{n\lambda^2}} - \mu_{k-1}.$$

□

Corollary 14. *If $\lambda = \max_{i\in[k]} \lambda_i$, then we have*

$$\mu_k \ge \mu_{k-1} + \lambda_k \cdot \left(1 - e^{\frac{-2\mu^2}{n\cdot\lambda^2}} - \mu_{k-1}\right).$$

Proof. Combining Lemmas 13 and 11 we have

$$\begin{aligned}\mu_k &\geq \mu_{k-1} + \lambda_k \cdot \Pr_{(\mathbf{u}_{\leq n}^{(\infty)}, \mathbf{v}_{\leq n}^{(\infty)})}[\mathrm{HD}(u_{\leq n}, v_{\leq n}) \geq k] \quad \text{(by Lemma 11)} \\ &\geq \mu_{k-1} + \lambda_k \cdot \left(1 - e^{\frac{-2\mu^2}{n \cdot \lambda^2}} - \mu_{k-1}\right) \quad \text{(by Lemma 13).}\end{aligned}$$

□

Theorem 15. *If $\lambda = \max_{i \in [k]} \lambda_i$, then we have*

$$\mu_k \geq \mu + \left(1 - \prod_{i=1}^{k}(1 - \lambda_i)\right) \cdot \left(1 - e^{\frac{-2\mu^2}{n \cdot \lambda^2}} - \mu\right).$$

In particular, by setting all $\lambda_i = \frac{\mu}{\sqrt{n}}$ we get

$$\mu_k \geq \mu + \left(1 - \left(1 - \frac{\mu}{\sqrt{n}}\right)^k\right) \cdot \left(1 - e^{-2} - \mu\right).$$

Note that the choice of $\lambda_i = \mu/\sqrt{n}$ above is not optimal when we want to maximize μ_i. The optimal choice does not have a compact closed form and leads to different λ_i's for different remaining budgets.

Proof. We prove this by induction on k. The case of $k = 1$ directly follows from Corollary 14. For $k > 1$, by Corollary 14 we have

$$\mu_k \geq \mu_{k-1} + \lambda_k \cdot \left(1 - e^{\frac{-2\mu^2}{n \cdot \lambda^2}} - \mu_{k-1}\right),$$

which implies that

$$\mu_k \geq (1 - \lambda_k) \cdot \mu_{k-1} + \lambda_k \cdot \left(1 - e^{\frac{-2\mu^2}{n \cdot \lambda^2}}\right).$$

Now we can use the induction's hypothesis and replace μ_{k-1} with $\mu + \left(1 - \prod_{i=1}^{k-1}(1 - \lambda_i)\right) \cdot \left(1 - e^{-2\mu^2/(n \cdot \lambda^2)} - \mu\right)$ which implies that

$$\mu_k \geq \mu + \left(1 - \prod_{i=1}^{k}(1 - \lambda_i)\right) \cdot \left(1 - e^{\frac{-2\mu^2}{n \cdot \lambda^2}} - \mu\right),$$

and that proves the claim. □

3.1 Making the attack run in polynomial time

In this section, we explain why the attack of the Section 3 can be implemented in polynomial time. In particular, we show how we can modify Construction 10 so that it runs in polynomial time, if one can efficiently sample form the random process conditioned on any prefix. (This is true, e.g., when the random process models a single-turn coin tossing protocol, as the original protocol shall run in polynomial time.)

At a high level, the proofs of this section closely follow the steps of the proofs in Section 3, and in each step we show that the proof is robust to using "approximate" values for what we previously assumed to be

known exactly. Therefore, for a reader who is not primarily concerned with the polynomial-time aspect of the attack, we suggest reading Section 3, which is simpler.

Note that the attack of Construction 10 is not polynomial time mainly because calculating $\bar{f}(\cdot)$ and $f^*(\cdot)$ oracles is not a polynomial-time task. In order to make the attack polynomial time, we first show that if the algorithm has access to the approximated version of these oracles, it still can achieve almost the same bias towards 1. Then, we show that calculating the approximation of these approximate oracles is actually possible in polynomial-time.

First, we first state our new construction that uses the approximate oracles.

Construction 16 (k-replacing using approximate partial-average and maximum-child oracles)**.** This attack uses the approximate oracle $\tilde{f}(\cdot)$ (see Definition 6) so that for all $v_{\leq i}$ we have

$$|\tilde{f}(v_{\leq i}) - \bar{f}(v_{\leq i})| \leq \tau. \tag{12}$$

The attack also uses an additional oracle for returning an approximate best choice for next block, such that

$$u^*_{i+1} \in \left\{ u'_{i+1} \mid \Pr\left[\tilde{f}(v_{\leq i}, u_{i+1}) > \tilde{f}(v_{\leq i}, u'_{i+1})\right] \leq \tau \right\} \tag{13}$$

and also let $\tilde{f}^*(v_{\leq i}) = \tilde{f}(\tilde{u}^*_{i+1})$ and $d = \mathrm{HD}(u_{\leq i}, v_{\leq i})$. The attack is parameterized by a vector $\lambda_{\leq k} = (\lambda_1, \ldots, \lambda_k) \in [0,1]^k$ for some integer $k \leq n$ which is adversary's budget. The attack will keep a state $\sigma_i = (u_{\leq i}, v_{\leq i})$ where $u_{\leq i}$ are the original values and $v_{\leq i}$ are the output values under attack. Having state $(u_{\leq i}, v_{\leq i})$ and for given u_{i+1} the approximate attacker App will decide on whether to keep or replace u_{i+1}, using u^*_{i+1}, $\tilde{f}^*$, and $d = \mathrm{HD}(u_{\leq i}, v_{\leq i})$ as follows.

- (Case 0) If $d \geq k$ do nothing and output $v_{i+1} = u_{i+1}$.
- (Case 1) if Case 0 does not happen and $\tilde{f}(v_{\leq i}, u_{i+1}) < \tilde{f}^*(v_{\leq i}) - \lambda_z$, then output $v_{i+1} = u^*_{i+1}$.
- (Case 2) If Case 0 and Case 1 not happen, do nothing and set $v_{i+1} = u_{i+1}$.

In all the cases above, App will also update the state σ_{i+1} accordingly by setting $\sigma_{i+1} = (u_{\leq i+1}, v_{\leq i+1})$.

Now we show that if conditions 12 and 13 hold, then the attack of Construction 16 can achieve the desired bias.

Lemma 17. *Let μ_k be the average of the output bit after applying the attack of Construction 16. Then, for $\lambda = \max_{i \in [k]} \lambda_i$ we have*

$$\mu_k \geq \mu + \left(1 - \prod_{i=1}^{k}(1-\lambda_i)\right) \cdot \left(1 - e^{\frac{-2(\mu - 5n\tau)^2}{n \cdot \lambda^2}} - \mu\right) - 6n \cdot k \cdot \tau.$$

In particular, by setting all $\lambda_i = \frac{\mu}{\sqrt{n}}$ and $\tau \leq \min(\frac{\mu}{10000n}, \frac{\varepsilon}{12 \cdot n^2})$ we get

$$\mu_k \geq \mu + \left(1 - \left(1 - \mu/\sqrt{n}\right)^k\right) \cdot (1 - e^{-1.99} - \mu) - \varepsilon/2.$$

Before proving the theorem above, we mention the following corollary about the power of polynomial time attacks.

Theorem 18. *For any ε there is a k-replacing $\mathsf{App}^{\mathbf{w}_{\leq n}[.]}$ attack with oracle access to the sampler from random process that runs in time $\mathrm{poly}(N/(\varepsilon \cdot \mu))$ and achieves bias at least*

$$\left(1 - \left(1 - \frac{\mu}{\sqrt{n}}\right)^k\right) \cdot (1 - e^{-1.99} - \mu) - \varepsilon,$$

where N is the total bit representation of the process.

Proof. We shall only show how to implement the approximate oracles $\tilde{f}(\cdot)$ and $\tilde{f}^*(\cdot)$ in polynomial time. Doing so is possible by continuing the random process (i.e., the coin flipping protocol) many times and taking their average, which is possible to be done efficiently because the attack has oracle access to the process (in the context of coin tossing, this is possible if the protocol is single turn and turns in polynomial time). Using Chernoff-Hoefding bound, we can bound the probability of having error $|\tilde{f} - \bar{f}|$ larger than τ to be at most $\varepsilon/4n$, by making $\mathrm{poly}(n/(\varepsilon \cdot \tau))$ random continuations. We can also ensure the condition for $\tilde{f}^*$ to happen with probability at least $1 - \varepsilon/4n$, by making $\mathrm{poly}(\log(n/\varepsilon)/\tau)$ samples. Of course, there is a chance of our approximation failing but that does not hurt the proof as we bound the probability of the failure for both type of queries by $\varepsilon/4n$ and in total we make at most $2n$ such queries during the course of the protocol. Therefore, by union bound, the final probability of any of these queries failing is at most $\varepsilon/2$. In the worst-case, for the failure scenarios the average becomes 0 and we lose an additive $\varepsilon/2$ in the final bias. Since our attack at each round makes $\mathrm{poly}(n/(\varepsilon \cdot \tau))$ random continuations, the running time of the attack given oracle access to the sampler is equal to $O(\mathrm{poly}(n/\varepsilon \cdot \tau)) = O(\mathrm{poly}(n/\varepsilon \cdot \mu))$. □

Now we prove Lemma 17.

Proof of Lemma 17. The proof is similar to the proof of Theorem 15. We first need to show that the approximate attack would achieve high bias, in the case of infinite number of replacements. We start by showing a variation of Lemma 12 with the approximated oracle.

Lemma 19. *Let $\lambda = \max_{i \in [n]} \lambda_i$. Then, we have*

$$\mu_\infty \geq 1 - e^{-\frac{2(\mu - 5n\tau)^2}{n\lambda^2}} - 2\tau n$$

Proof. We define a sequence of random variables $\mathbf{t}_{\leq n} = (\mathbf{t}_1, \ldots, \mathbf{t}_n)$, where $t_{i+1} = \tilde{f}(v_{\leq i+1}) - \tilde{f}(v_{\leq i})$ is a random variable that is dependent on $v_{\leq i}$.

Then we have

$$\mathop{\mathbb{E}}_{(\mathbf{u}^\infty, \mathbf{v}^\infty)[u_{\leq i}, v_{\leq i}]} [\tilde{f}(v_{\leq i+1}) - \tilde{f}(v_{\leq i})] \geq \mathop{\mathbb{E}}_{(\mathbf{u}^\infty, \mathbf{v}^\infty)[u_{\leq i}, v_{\leq i}]} [\bar{f}(v_{\leq i}, u_{i+1}) - \bar{f}(v_{\leq i}) - 2\tau] = -2\tau. \tag{14}$$

Therefore, $\mathbf{t}_{\leq n}$ defines an approximate sub-martingale.

Furthermore, by the guarantee of $\tilde{f}^*$ and the way the attack works we have

$$\Pr_{(\mathbf{u}^\infty, \mathbf{v}^\infty)[u_{\leq i}, v_{\leq i}]} [\tilde{f}^*(v_{\leq i}) \geq \tilde{f}(v_{\leq i+1}) \geq \tilde{f}^*(v_{\leq i}) - \lambda] \geq 1 - \tau \tag{15}$$

Therefore, t_i always falls in an interval of size λ and by Applying the approximate Azuma's inequality (as stated in Lemma 8) over t_i, we have

$$\Pr\left[\sum_{i=1}^{n} t_i \leq -\mu + \tau\right] \leq e^{-\frac{2(\mu - 5n\tau)^2}{n\lambda^2}} + 2n\tau.$$

Then we have

$$\Pr[f(v_{\le n})=1] \ge \Pr[\tilde{f}(v_{\le n})>\tau] = \Pr\left[\sum t_i \ge \tau-\mu\right] \ge 1-e^{-\frac{2(\mu-5n\tau)^2}{n\lambda^2}}-2n\tau.$$

□

We then have the approximated version of Lemma 11.

Lemma 20. *We have*

$$\mu_k \ge \mu_{k-1}+\lambda_k \cdot \Pr_{(u_{\le n},v_{\le n})\leftarrow(\mathbf{u}^{(\infty)}_{\le n},\mathbf{v}^{(\infty)}_{\le n})}[\mathrm{HD}(u_{\le n},v_{\le n})\ge k]-4n\tau.$$

Proof. The proof is very similar to the proof of Lemma 11. The only difference is that in Equation (4) we switch from $\bar{f}$ to $\tilde{f}$ and we only loose $2\tau n$ because of the approximation error. Namely,

$$\left|\mathop{\mathbb{E}}_{(\mathbf{u}^k_{\le n},\mathbf{v}^k_{\le n})[u_{\le i},v_{\le i}]}\left[\sum_{i=1}^{n}\left(\tilde{f}(v_{\le i+1})-\tilde{f}(v_{\le i})\right)\right]-\mathop{\mathbb{E}}_{(\mathbf{u}^k_{\le n},\mathbf{v}^k_{\le n})[u_{\le i},v_{\le i}]}\left[\sum_{i=1}^{n}\left(\bar{f}(v_{\le i+1})-\bar{f}(v_{\le i})\right)\right]\right|\le 2\tau n. \tag{16}$$

All other equations will remain the same because Equations 1, 2 and 3 still hold when we use $\tilde{f}$ instead of $\bar{f}$.

This will change Equation (5) and will add an additive term of $2\tau n$ and we have

$$\mathop{\mathbb{E}}_{(\mathbf{u}^k_{\le n},\mathbf{v}^k_{\le n})}[f(v_{\le n})]-\mu \ge$$
$$\sum_{i=0}^{n-1}\mathop{\mathbb{E}}_{(\mathbf{u}^\infty_{\le i},\mathbf{v}^\infty_{\le i})}\left[\mathop{\mathbb{E}}_{(\mathbf{u}^k_{\le n},\mathbf{v}^k_{\le n}))[u_{\le i},v_{\le i}]}\left[g(v_{\le i+1},u_{\le i+1})\cdot\left(C_1^\infty(u_{i+1})\wedge(\mathrm{HD}(u_{\le i},v_{\le i})<k)\right)\right]\right]-2n\tau.$$

Similarly, for $k-1$ we have

$$\mathop{\mathbb{E}}_{(\mathbf{u}^{k-1}_{\le n},\mathbf{v}^{k-1}_{\le n})}[f(v_{\le n})]-\mu \le$$
$$\sum_{i=0}^{n-1}\mathop{\mathbb{E}}_{(\mathbf{u}^\infty_{\le i},\mathbf{v}^\infty_{\le i})}\left[\mathop{\mathbb{E}}_{(\mathbf{u}^{k-1}_{\le n},\mathbf{v}^{k-1}_{\le n})[u_{\le i},v_{\le i}]}\left[g(v_{\le i+1},u_{\le i+1})\cdot\left(C_1^\infty(u_{i+1})\wedge(\mathrm{HD}(u_{\le i},v_{\le i})<k-1)\right)\right]+2n\tau.$$

By subtracting the two inequalities above the proof of Lemma is complete. □

Lemma 21. *If* $\lambda=\max_{i\in[k]}\lambda_i$, *then*

$$\Pr_{(\mathbf{u}^\infty_{\le n},\mathbf{v}^\infty_{\le n})}[\mathrm{HD}(u_{\le n},v_{\le n})\ge k]\ge 1-e^{-\frac{2(\mu-5n\tau)^2}{n\lambda^2}}-2n\tau-\mu_{k-1}.$$

Proof. The proof steps of this lemma are exactly as those of Lemma 13. First we have

$$\begin{aligned}
&\Pr_{(\mathbf{u}_{\le n}^{\infty},\mathbf{v}_{\le n}^{\infty})}[\big(f(v_{\le n})=1\wedge \mathrm{HD}(u_{\le n},v_{\le n})<k\big)\vee(\mathrm{HD}(u_{\le n},v_{\le n})\ge k)]\\
&\quad=\Pr_{(\mathbf{u}_{\le n}^{\infty},\mathbf{v}_{\le n}^{\infty})}[f(v_{\le n})=1\vee \mathrm{HD}(u_{\le n},v_{\le n})\ge k]\\
&\quad\ge\Pr_{(\mathbf{u}_{\le n}^{\infty},\mathbf{v}_{\le n}^{\infty})}[f(v_{\le n})=1]\\
&\quad=\mu_{\infty}\\
&\text{(by Lemma 19)}\ge 1-e^{-\frac{2(\mu-5n\tau)^2}{n\lambda^2}}+2n\tau. \qquad (17)
\end{aligned}$$

On the other hand, by union bound we have

$$\begin{aligned}
&\Pr_{(\mathbf{u}_{\le n}^{\infty},\mathbf{v}_{\le n}^{\infty})}[\big(f(v_{\le n})=1\wedge \mathrm{HD}(u_{\le n},v_{\le n})<k\big)\vee(\mathrm{HD}(u_{\le n},v_{\le n})\ge k)]\\
&\le\Pr_{(\mathbf{u}_{\le n}^{\infty},\mathbf{v}_{\le n}^{\infty})}[f(v_{\le n})=1\wedge \mathrm{HD}(u_{\le n},v_{\le n})<k]+\Pr_{(\mathbf{u}_{\le n}^{\infty},\mathbf{v}_{\le n}^{\infty})}[\mathrm{HD}(u_{\le n},v_{\le n})\ge k]. \qquad (18)
\end{aligned}$$

It also holds that

$$\Pr_{(\mathbf{u}_{\le n}^{\infty},\mathbf{v}_{\le n}^{\infty})}[\big(f(v_{\le n})=1\wedge \mathrm{HD}(u_{\le n},v_{\le n})<k\big)]\le\Pr_{(\mathbf{u}_{\le n}^{k-1},\mathbf{v}_{\le n}^{k-1})}[f(v_{\le n})=1]=\mu_{k-1}. \qquad (19)$$

Now, combining Inequalities 17, 18 and 19 we get

$$\Pr_{(\mathbf{u}_{\le n}^{\infty},\mathbf{v}_{\le n}^{\infty})}[\mathrm{HD}(u_{\le n},v_{\le n})\ge k]\ge 1-e^{-\frac{2(\mu+5n\tau)^2}{n\lambda^2}}-2n\tau-\mu_{k-1},$$

which finishes the proof. □

Corollary 22. *If $\lambda=\max_{i\in[k]}\lambda_i$, then we have*

$$\mu_k\ge\mu_{k-1}+\lambda_k\cdot\left(1-e^{\frac{-2(\mu-5n\tau)^2}{n\cdot\lambda^2}}-2n\tau-\mu_{k-1}\right)-4n\cdot\tau.$$

Proof. This corollary follows by combining Lemmas 20 and 21. □

Putting things together. Now that we have all the required Lemmas we can prove Theorem using the exact same inductive argument of Theorem 15. The case of $k=1$ directly follows from Corollary 22. For $k>1$, by Corollary 22 we have

$$\mu_k\ge\mu_{k-1}+\lambda_k\cdot\left(1-e^{\frac{-2(\mu-5n\tau)^2}{n\cdot\lambda^2}}-2n\tau-\mu_{k-1}\right)-4n\cdot\tau,$$

which implies that

$$\mu_k\ge(1-\lambda_k)\cdot\mu_{k-1}+\lambda_k\cdot\left(1-e^{\frac{-2(\mu-5n\tau)^2}{n\cdot\lambda^2}}\right)-6n\cdot\tau.$$

Now we can use induction hypothesis and replace μ_{k-1} with

$$\mu+\left(1-\prod_{i=1}^{k-1}(1-\lambda_i)\right)\cdot\left(1-e^{\frac{-2(\mu-5n\tau)^2}{n\cdot\lambda^2}}-\mu\right)-6n\cdot(k-1)\cdot\tau$$

which implies that

$$\mu_k \geq \mu + \left(1 - \prod_{i=1}^{k}(1-\lambda_i)\right) \cdot \left(1 - e^{\frac{-2(\mu-5n\tau)^2}{n\cdot\lambda^2}} - \mu\right) - 6nk\tau,$$

and that proves the claim.

□

4 Optimal attacks for uniform binary messages

In this section, we focus on the setting in which n parties each send a uniform random bit and then a final bit is chosen based on the published messages. We will show how to obtain *optimal* online k-replacing attacks that *match* the power of *offline* attacks.

Notation. $\mathbf{u}_{\leq n} \equiv (\mathbf{u}_1 \times \cdots \times \mathbf{u}_n)$ denotes the uniform random variable over $\{0,1\}^n$, where each $\mathbf{u}_i$ is a uniform and independent random bit. In this section, for simplicity we use notation $\mathbf{U}_n$ for this distribution. We will study k-replacing attacks on $\mathbf{U}_n$.[8] $\mathrm{HW}(x) = \mathrm{HD}(x, 0^n)$ denotes Hamming weight of $x \in \{0,1\}^n$. We let $[n] = \{1,\ldots,n\}$, $\langle n] = \{0,\ldots,n\}$ and $\langle n\rangle = \{0,\ldots,n+1\}$. For $t \in \langle n\rangle$, we define the threshold function $\tau_t \colon \{0,1\}^n \to \{0,1\}$ as $\tau_t(x) = 1$ iff $\mathrm{HW}(x) \geq t$. (τ_0 is the constant function 1 function and τ_{n+1} is the constant 0 function.) We let $\beta_n^{(t)} = 2^{-n} \cdot \sum_{i=t}^{n} \binom{n}{i}$ be the probability of the Hamming ball defined by τ_t, and when n is clear from the context we write it as $\beta^{(t)}$. We also let $s_n^{(t)} = 2^n \cdot \beta_n^{(t)}$ be the size of the same Hamming ball. For set $\mathcal{S} \subset \mathbb{R}, r \in \mathbb{R}$, we use the notation $r\mathcal{S} = \{rx \mid x \in \mathcal{S}\}$, e.g., $r\langle n] = \{0, r, 2r, \ldots, nr\}$. We let $\binom{n}{k} = 0$ if $k < 0$ or $k > n$. For a set $\mathcal{S} \subseteq \{0,1\}^n$ and $r \in \{0,1\}^d$ for $d \in [n]$, we let

$$\mathcal{S}[r] = \left\{x' \mid x \in \mathcal{S} \wedge \exists x' \in \{0,1\}^{n-d} \text{ such that } x = (r, x')\right\}$$

be the set of suffixes of strings in $\mathcal{S}$ of length $n-d$ with r as their prefix.

We first define the isoperimetry function that capture the power of "offline" attacks.

Definition 23 (The offline expansion and isoperimetry functions)**.** For $k \in [n], \mathcal{S} \subseteq \{0,1\}^n$, the offline *$k$-expansion* (probability) of $\mathcal{S}$ is the probability of all points within Hamming distance k of $\mathcal{S}$

$$\mathsf{OffExp}^{(k)}(\mathcal{S}) = \frac{|\{y \in \{0,1\}^n \mid \exists x \in \mathcal{S}, \mathrm{HD}(x,y) \leq k\}|}{2^n}.$$

For a given probability μ, the *k-expansion* of μ is equal to:

$$\mathsf{OffExp}^{(k)}(\mu) = \inf_{\mathcal{S}, \Pr[\mathcal{S}] \geq \mu} \mathsf{OffExp}^{(k)}(\mathcal{S}).$$

Finally, for a set $\mathcal{S}$ and probability μ, we define the (offline) *k-isoperimetry* function

$$\mathsf{OffIso}^{(k)}(\mathcal{S}) = \mathsf{OffExp}^{(k)}(\mathcal{S}) - \Pr[\mathcal{S}], \quad \mathsf{OffIso}_n^{(k)}(\mu) = \mathsf{OffExp}_n^{(k)}(\mu) - \mu.$$

Note that whenever the input is a set $\mathcal{S} \subseteq \{0,1\}^n$, it already determines n on its own, and hence we do not need to state it explicitly, but when the input is $\mu \in \mathbb{R}$, we explicitly state n as the index of the function.

[8] In Sections 2 and 3, we called the original random process $\mathbf{w}_{\leq n}$ and $\mathbf{U}_n$ was one of the generated random processes (modeling the original samples). However, since we are starting from a *product* distribution, it would hold that $\mathbf{U}_n \equiv \mathbf{w}_{\leq n}$, and thus we simply call the original distribution $\mathbf{u}$.

Theorem 24 (Implied by the vertex isoperimetric inequality in Boolean hypercube [Har66])**.** *For any $t \in \langle n]$, it holds that* $\mathsf{OffExp}^{(k)}(\beta^{(t)}) = \beta^{(t-k)}$.

Online attacks vs. offline attacks. Suppose an adversary wants to increase the probability of falling into a set $\mathcal{S}$ in an "offline" attack, in which the adversary gets a point $x \leftarrow \mathbf{U}_n$ and then can replace k of the bits of x. It is easy to see that the adversary can increase the probability of falling into $\mathcal{S}$ exactly by $\mathsf{OffIso}^{(k)}(\mathcal{S})$. Accordingly, we can define the *online* variant of such attacks as defined in Section 5. In such online attacks, the adversary gets to see the independent and uniformly sampled random bits $(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ one by one, and after seeing $u_i \leftarrow \mathbf{u}_i$, it can decide to keep or change it.

Definition 25 (The online expansion OnExp and isoperimetry OnIso functions)**.** Let A be an online adversary of budget k over the uniform distribution $\mathbf{U}_n$ over $\{0,1\}^n$. Let $\mathbf{v}_{\leq n}$ be the generated output random process (distributed over $\{0,1\}^n$) under attack A (as defined in Definition 5). We define $\mathsf{OnExp}^{(\mathsf{A})}(\mathcal{S}) = \Pr[\mathbf{v}_{\leq n} \in \mathcal{S}]$. Let $\mathcal{A}_k$ be the set of *all* k-replacing attacks on $\mathbf{U}_n$. We define $\mathsf{OnExp}^{(k)}(\mathcal{S})$ as the maximum probability of points in $\{0,1\}^n$ that any online adversary can map to $\mathcal{S}$ by up to k changes to a stream of n uniformly random bits. Namely,

$$\mathsf{OnExp}^{(k)}(\mathcal{S}) = \max_{\mathsf{A} \in \mathcal{A}_k} \mathsf{OnExp}^{(\mathsf{A})}(\mathcal{S}).$$

Also, for any $\mu \in [0,1]$, we define

$$\mathsf{OnExp}_n^{(k)}(\mu) = \inf_{\mathcal{S}, \Pr[\mathcal{S}] \geq \mu} \mathsf{OnExp}^{(k)}(\mathcal{S})$$

as the minimum $\mathsf{OnExp}^{(k)}(\mathcal{S})$ among all sets of probability at least μ. Finally, for any set $\mathcal{S}$ and probability μ, we define the *online k-isoperimetry* functions as follows

$$\mathsf{OnIso}^{(k)}(\mathcal{S}) = \mathsf{OnExp}^{(k)}(\mathcal{S}) - \Pr[\mathcal{S}], \quad \mathsf{OnIso}_n^{(k)}(\mu) = \mathsf{OnExp}_n^{(k)}(\mu) - \mu$$

as the *growth* in probability of falling into sets (of probability μ) under optimal online k-replacing attacks.

Since offline adversaries know as much as online adversaries when making decision to change or not, it always holds that $\mathsf{OffIso}(\mathcal{S}) \geq \mathsf{OnIso}(\mathcal{S})$, and hence $\mathsf{OffIso}_n^{(k)}(\mu) \geq \mathsf{OnIso}_n(\mu)$ for all n, $\mathcal{S} \subseteq \{0,1\}^n$, and $\mu \in [0,1]$. The surprising phenomenon stated in the next theorem is that when μ is the probability of a Hamming ball, online and offline attacks have the *same exact* power as a function of the measure μ, and consequently the online and offline k-isoperimetry functions would be equal.

Theorem 26 (Power of online vs. offline attacks for the uniform distribution over $\{0,1\}^n$)**.** *For all $n \in \mathbb{N}, t \in [n], k \leq t$, if $\beta^{(t)} = \Pr[\mathrm{HW}(\mathbf{U}_n) \geq t]$ be the probability of a Hamming ball. Then it holds that*

$$\mathsf{OnExp}_n^{(k)}(\beta^{(t)}) = \mathsf{OffExp}_n^{(k)}(\beta^{(t)}) = \beta^{(t-k)}.$$

In words, if $\mu = \beta^{(t)}$, then the power of online k-replacing adversaries to increase the probability of falling into a set $\mathcal{S}$, in the minimum over all sets of probability at least $\beta^{(t)}$, is equal to that of offline attacks.

Reaching a target probability. Suppose $\Pr[\mathcal{S}] = \mu$, and suppose we want to increase the probability of falling into $\mathcal{S}$ to $\mu' > \mu$. How much budget an adversary needs? Theorem 26 shows that as long as μ is the probability of a Hamming ball (i.e., $\mu = \beta^{(t)}$), then in the worst case (among all possible sets $\mathcal{S}$

of probability μ) the power of online and offline attacks are *exactly the same*. Therefore, this brings up the natural question of what happens in general, when μ is *not* exactly the probability of a Hamming ball. As stated in Corollary 27 below, Theorem 26 already shows that the power of offline and online attacks is different by at most one. In fact, as we will see later, these quantities are not equal in general. In particular, Figure 1 compares $\mathsf{OnIso}_n(\mu)$ and $\mathsf{OffIso}_n^{(k)}(\mu)$ for all μ when $n = 10$ (and $k = 1$).

Corollary 27 (Budget of online vs. offline attacks to reach a target probability)**.** *For $0 < \mu < \mu' \leq 1$, let*

$$\mathsf{OfBud}_n(\mu \to \mu') = \min_{k \in [n]} [\mathsf{OffExp}_n^{(k)}(\mu) \geq \mu']$$

be the minimum budget k that an offline adversary needs to increase the probability of falling into any set $\mathcal{S}$ of probability at least μ to μ'. Let

$$\mathsf{OnBud}_n(\mu \to \mu') = \min_{k \in [n]} [\mathsf{OnExp}_n^{(k)}(\mu) \geq \mu']$$

be the similar quantity for online attacks. Then, it always holds that

$$\mathsf{OfBud}_n(\mu \to \mu') \leq \mathsf{OnBud}_n(\mu \to \mu') \leq \mathsf{OfBud}_n(\mu \to \mu') + 1$$

and $\mathsf{OfBud}_n(\beta^{(t)} \to \mu') = \mathsf{OnBud}_n(\beta^{(t)} \to \mu')$ *for all* $t \in [n+1]$.

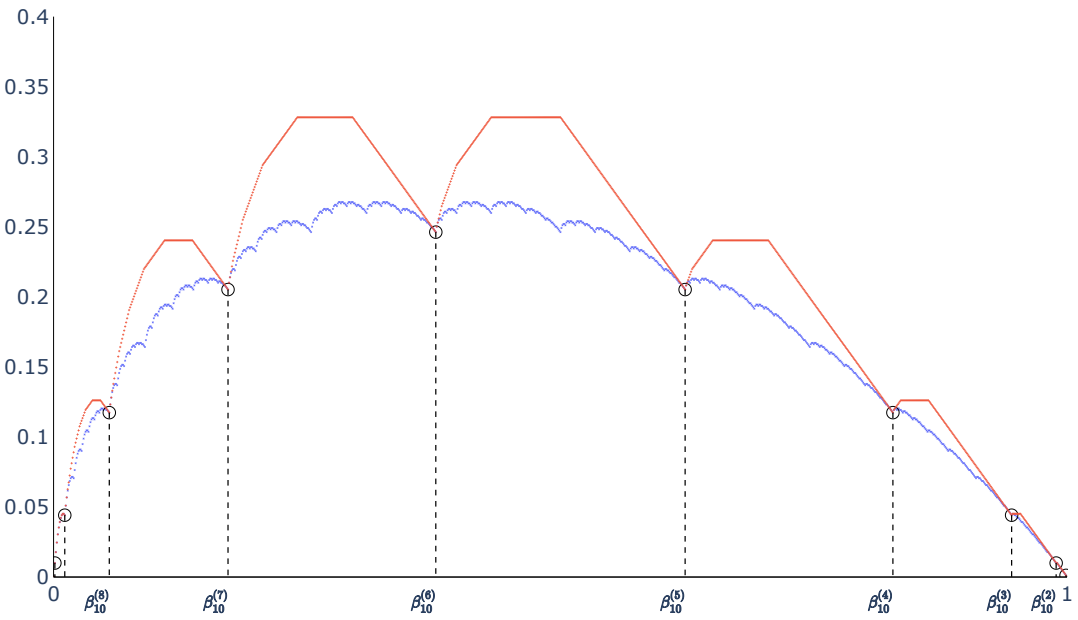

Figure 1: Comparing the online isoperimetric function OnIso (blue) versus the offline isoperimetric function OffIso (red) for $n = 10$.

We first prove Corollary 27 using Theorem 26 and then will prove Theorem 26.

Proof of Corollary 27. Let $k = \mathsf{OfBud}_n(\mu \to \mu')$, we have $\mathsf{OffExp}_n^{(k)}(\mu) \geq \mu'$ and $\mathsf{OffExp}_n^{(k-1)}(\mu) < \mu'$. Let $t \in \langle n \rangle$ be the minimum such t that $\beta^{(t)} \geq \mu$, and so we have $\beta^{(t+1)} \leq \mu \leq \beta^{(t)}$. By the monotonicity of $\mathsf{OnExp}_n^{(k)}$ function, we have

$$\mathsf{OnExp}_n^{(k+1)}(\beta^{(t+1)}) \leq \mathsf{OnExp}_n^{(k+1)}(\mu). \tag{20}$$

By Theorem 26 it holds that $\mathsf{OnExp}_n^{(k+1)}(\beta^{(t+1)}) = \mathsf{OffExp}_n^{(k+1)}(\beta^{(t+1)})$. Now, because $\beta^{(t+1)} \leq \mu \leq \beta^{(t)}$, by the monotonicity of $\mathsf{OffExp}_n^{(k)}(\mu)$ we have

$$\mathsf{OffExp}_n^{(k)}(\mu) \leq \mathsf{OffExp}_n^{(k)}(\beta^{(t)}) = \mathsf{OffExp}_n^{(k+1)}(\beta^{(t+1)}) = \mathsf{OnExp}_n^{(k+1)}(\beta^{(t+1)}). \tag{21}$$

Combining (20) and (21), we have

$$\mathsf{OffExp}_n^{(k)}(\mu) \leq \mathsf{OnExp}_n^{(k+1)}(\beta^{(t+1)}) \leq \mathsf{OnExp}_n^{(k+1)}(\mu).$$

Therefore we have,

$$\begin{aligned}
\mathsf{OfBud}_n(\mu \to \mu') + 1 &= \min_{k \in [n]} [\mathsf{OffExp}_n^{(k)}(\mu) \geq \mu'] + 1 \\
&\geq \min_{k \in [n]} [\mathsf{OnExp}_n^{(k+1)}(\mu) \geq \mu'] + 1 \\
&= \min_{k+1 \in [n]} [\mathsf{OnExp}_n^{(k+1)}(\mu) \geq \mu'] \\
&= \mathsf{OnBud}_n(\mu \to \mu').
\end{aligned}$$

The inequality holds because let $k' = \min_{k \in [n]} [\mathsf{OffExp}_n^{(k)}(\mu) \geq \mu']$, we have $\mathsf{OnExp}_n^{(k'+1)}(\mu) \geq \mu'$, and therefore $\min_{k \in [n]} [\mathsf{OnExp}_n^{(k+1)}(\mu) \geq \mu'] \leq k'$. Since we also have $\mathsf{OffExp}_n^{(k)}(\mu) \geq \mathsf{OnExp}_n^{(k)}(\mu)$ for any μ, $\mathsf{OfBud}_n(\mu \to \mu') \leq \mathsf{OnBud}_n(\mu \to \mu') \leq \mathsf{OfBud}_n(\mu \to \mu') + 1$.

Finally, because $\mathsf{OffExp}_n^{(k)}(\beta^{(t)}) \geq \mathsf{OnExp}_n^{(k)}(\beta^{(t)})$ holds for any k and t, we have $\mathsf{OfBud}_n(\beta^{(t)} \to \mu') = \mathsf{OnBud}_n(\beta^{(t)} \to \mu')$ for all $t \in [n+1]$. □

In the rest of this section, we prove Theorem 26.

Proof of Theorem 26. To Theorem 26, we start by deriving a recursive relation for $\mathsf{OnExp}_n^{(k)}(\cdot)$. Before doing so, we define some mathematical notation.

Definition 28 (Definitions related to the recursive relation of online expansion)**.** For $s \in \langle 2^n]$, let

$$\mathcal{D}iv_{n-1}(s) = \left\{(s_0, s_1) \mid s_0, s_1 \in \langle 2^{n-1}], 0 \leq s_0 \leq s_1 \leq 2^{n-1}, s = s_0 + s_1\right\}$$

be the set of ways in which a "set size" $s \in \langle 2^{n-1}]$ can be divided into two sizes. For $(s_0, s_1) \in \mathcal{D}iv_{n-1}(s)$ and a *fixed* pair of integers n, k let $\mathsf{Rec}_n^{(k)}(\cdot, \cdot)$ be defined as

$$\mathsf{Rec}_n^{(k)}(s_0, s_1) = \frac{\mathsf{Rec}_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\mathsf{Rec}_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right), \mathsf{Rec}_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \tag{22}$$

based on functions $\mathsf{Rec}_{n-1}^{(k)}, \mathsf{Rec}_{n-1}^{(k-1)}$ to be specified later. Finally, for $\mu \in 2^{-n}\langle 2^n]$ let

$$\mathsf{Rec}_n^{(k)}(\mu) = \inf_{(s_0, s_1) \in \mathcal{D}iv_n(2^n \cdot \mu)} \mathsf{Rec}_n^{(k)}(s_0, s_1). \tag{23}$$

Transformation $\mathsf{Rec}_n^{(k)}[p,q]$**.** For functions p, q defined on input space $2^{-n}\langle 2^n]$. Suppose we use p instead of $\mathsf{Rec}_{n-1}^{(k)}$ and q instead of $\mathsf{Rec}_{n-1}^{(k-1)}$ in Equation (22). Then by $\mathsf{Rec}_n^{(k)}[p,q](\cdot,\cdot)$ (resp. $\mathsf{Rec}_n^{(k)}[p,q](\cdot)$) we denote the function that one obtains in Equation (22) (resp. Equation (23)).

Interpretation. $\mathsf{Rec}_n^{(k)}(s_0,s_1)$ represents the optimal choice that a tampering adversary can make to increase the probability of falling into a set of size s, when $\mathcal{S}[0]=\mathcal{S}_0, \mathcal{S}[1]=\mathcal{S}_1$ are adversarially chosen based on their sizes s_0, s_1 where $s_0 \le s_1$, and when the optimal online expansions for s_0, s_1 can be applied by (appropriate use of) functions $\mathsf{Rec}_{n-1}^{(k)}, \mathsf{Rec}_{n-1}^{(k-1)}$.

Notation. Let f, g be defined over the same input domain $\mathcal{D}$. We say $f \le g$, if $\forall \mu \in \mathcal{D}, f(\mu) \le g(\mu)$.

We now show that the transformation of Definition 28 has some desired properties.

Claim 29 (Transformation of Definition 28 is monotone)**.** *Let* $\mathsf{u}_{n-1}^{(k)} \le \mathsf{v}_{n-1}^{(k)}$ *and* $\mathsf{u}_{n-1}^{(k-1)} \le \mathsf{v}_{n-1}^{(k-1)}$*, and let*

$$\mathsf{u}_n^{(k)} = \mathsf{Rec}_n^{(k)}[\mathsf{u}_{n-1}^{(k)}, \mathsf{u}_{n-1}^{(k-1)}], \quad \mathsf{v}_n^{(k)} = \mathsf{Rec}_n^{(k)}[\mathsf{v}_{n-1}^{(k)}, \mathsf{v}_{n-1}^{(k-1)}]$$

as defined in Definition 28. Then, it holds that $\mathsf{u}_n^{(k)} \le \mathsf{v}_n^{(k)}$.

Proof. We first show that for any $s_0, s_1 \in \mathcal{D}iv_n(2^n \cdot \mu)$, we have $\mathsf{u}_n^{(k)}(s_0,s_1) \le \mathsf{v}_n^{(k)}(s_0,s_1)$. Because $\mathsf{u}_{n-1}^{(k)} \le \mathsf{v}_{n-1}^{(k)}$ and $\mathsf{u}_{n-1}^{(k-1)} \le \mathsf{v}_{n-1}^{(k-1)}$, we have $\mathsf{u}_{n-1}^{(k)}\left(s_1/2^{n-1}\right) \le \mathsf{v}_{n-1}^{(k)}\left(s_1/2^{n-1}\right)$ and

$$\max\{\mathsf{u}_{n-1}^{(k)}\left(s_0/2^{n-1}\right), \mathsf{u}_{n-1}^{(k)}\left(s_1/2^{n-1}\right)\} \le \max\{\mathsf{v}_{n-1}^{(k)}\left(s_0/2^{n-1}\right), \mathsf{v}_{n-1}^{(k)}\left(s_1/2^{n-1}\right)\}.$$

Therefore, $\mathsf{u}_n^{(k)}(s_0,s_1) \le \mathsf{v}_n^{(k)}(s_0,s_1)$ holds for any s_0, s_1.

From Eq. (23), let $(s_0', s_1') = \arg\inf_{(s_0,s_1)\in\mathcal{D}iv_n(2^n\cdot\mu)} \mathsf{v}_n^{(k)}(s_0,s_1)$ be the partition where $\mathsf{v}_n^{(k)}(\mu)$ achieves its minimum. Then we have $\mathsf{u}_n^{(k)}(\mu) \le \mathsf{u}_n^{(k)}(s_0',s_1') \le \mathsf{v}_n^{(k)}(s_0',s_1') = \mathsf{v}_n^{(k)}(\mu)$. □

Claim 30 (Recursive relation for online expansion)**.** *One can recursively compute* $\mathsf{OnExp}_n^{(k)}(\mu)$ *for all* $\mu \in 2^{-n}\langle 2^n]$ *as follows.*

- *If* $k = 0$ *and* $n \ge 0$*, then* $\mathsf{OnExp}_n^{(0)}(\mu) = \mu$.
- *If* $k \ge 1$ *and* $k \ge n$*, then:* $\mathsf{OnExp}_n^{(k)}(0) = 0$ *and* $\mathsf{OnExp}_n^{(k)}(\mu) = 1$ *for* $\mu > 0$.
- *If* $k \ge 1$ *and and* $k < n$*, then* $\mathsf{OnExp}_n^{(k)} = \mathsf{Rec}_n^{(k)}[\mathsf{OnExp}_{n-1}^{(k)}, \mathsf{OnExp}_{n-1}^{(k-1)}]$ *as in Definition 28.*

Proof sketch. The extremal cases of the recursive relation stated in the first two bullets hold trivially. Below we argue why the inductive step as stated in the third bullet holds as well.

Suppose by fixing the first bit to b we get a subset of size s_b, and $s_0 \le s_1$, and suppose in both cases the residual subsets $\mathcal{S}[0], \mathcal{S}[1]$ are chosen in the "worst" case (against the adversary) based on their sizes s_0, s_1, minimizing the success probability of an online adversary. Since $\mathsf{OnExp}(\cdot)$ is a monotone function, then when the first bit is selected to be 1, the adversary has no motivation to replace it with 0. When the first bit is selected to be 0, the adversary has choose between maximum of the expansions that arise from changing or not changing the bit to 1. Once we consider all ways that s can be split into $s = s_0 + s_1$, this leads to the definition of the recursion of Eq. (23) and the transformation of Definition 28. □

Claim 31. *Suppose* $p \leq \mathsf{OnExp}_{n-1}^{(k)}, q \leq \mathsf{OnExp}_{n-1}^{(k-1)}$ *for functions* p, q*. Then, it holds that* $\mathsf{Rec}_n^{(k)}[p,q] \leq \mathsf{OnExp}_n^{(k)}$ *(see Definition 28).*

Proof. The proof directly follows from Claims 30 and 29. □

We now define a piecewise-linear function $\ell_n^{(k)}$ to later prove to be a lower bound for $\mathsf{OnExp}_n^{(k)}$.

Definition 32 (The piecewise-linear (lower bound) function)**.** For any non-negative integers k, n, the function $\ell_n^{(k)} \colon [0,1] \to [0,1]$ is defined as follows.

- If $\mu = \beta_n^{(t)}$ for any $t \in \langle n \rangle$, it holds that $\ell_n^{(k)}\left(\beta_n^{(t)}\right) = \mathsf{OffExp}_n^{(k)}\left(\beta_n^{(t)}\right)$. Namely, $\ell_n^{(k)}\left(\beta_n^{(n+1)}\right) = \mathsf{OffExp}_n^{(k)}(0) = 0$, and for any $t \in \langle n]$, $\ell_n^{(k)}\left(\beta_n^{(t)}\right) = \beta_n^{(t-k)} = \Pr\left[\mathrm{HW}(\mathbf{U}_n) \geq t-k\right]$.
- If $\mu = \alpha\beta_n^{(t)} + (1-\alpha)\beta_n^{(t-1)}$ for $0 < \alpha < 1$ and any $t \in \langle n \rangle$, then $\ell_n^{(k)}(\mu) = \alpha \cdot \ell_n^{(k)}\left(\beta_n^{(t)}\right) + (1-\alpha) \cdot \ell_n^{(k)}\left(\beta_n^{(t-1)}\right)$.

Proposition 33 (Composition of the lower bound function)**.** *For any* $k_1, k_2, n \geq 0$ *and* $\mu \in [2^{-n}, 1]$*, it hold that* $\ell_n^{(k_1+k_2)}(\mu) = \ell_n^{(k_1)}\left(\ell_n^{(k_2)}(\mu)\right)$.

Proof. Consider every case,

- If $\mu = \beta_n^{(t)}$. By Definition 32 we have $\ell_n^{(k_1)}\left(\ell_n^{(k_2)}(\mu)\right) = \ell_n^{(k_1)}\left(\mathsf{OffExp}_n^{(k_2)}\left(\beta_n^{(t)}\right)\right)$. As $\mu \in [2^{-n}, 1]$, we have $t \leq n$. Therefore, $\mathsf{OffExp}_n^{(k_2)}\left(\beta_n^{(t)}\right) = \beta_n^{(t-k_2)}$. Therefore, we have

$$\ell_n^{(k_1)}\left(\ell_n^{(k_2)}\left(\beta_n^{(t)}\right)\right) = \ell_n^{(k_1)}\left(\beta_n^{(t-k_2)}\right) = \beta_n^{(t-(k_2+k_1))} = \ell_n^{(k_1+k_2)}\left(\beta_n^{(t)}\right).$$

- If $\mu = \alpha\beta_n^{(t)} + (1-\alpha)\beta_n^{(t-1)}$ for $0 < \alpha < 1$, In this case, by Definition 32 we have $\ell_n^{(k_2)}(\mu) = \alpha \cdot \ell_n^{(k_2)}\left(\beta_n^{(t)}\right) + (1-\alpha) \cdot \ell_n^{(k_2)}\left(\beta_n^{(t-1)}\right)$. As $\mu \in [2^{-n}, 1]$, we have $t \leq n$. Therefore, we have $\ell_n^{(k_2)}(\mu) = \alpha \cdot \beta_n^{(t-k_2)} + (1-\alpha) \cdot \beta_n^{(t-1-k_2)}$. We then have

$$\begin{aligned}
\ell_n^{(k_1)}\left(\ell_n^{(k_2)}(\mu)\right) &= \ell_n^{(k_1)}\left(\alpha \cdot \beta_n^{(t-k_2)} + (1-\alpha) \cdot \beta_n^{(t-1-k_2)}\right) \\
&= \alpha \cdot \beta_n^{(t-k_2-k_1)} + (1-\alpha) \cdot \beta_n^{(t-1-k_2-k_1)} \\
&= \ell_n^{(k_1+k_2)}(\mu).
\end{aligned}$$

□

Lemma 34. $\ell_n^{(k)}$ *is concave for all* $n, k \geq 0$.

Proof. $\ell_n^{(0)}$ is linear, and hence concave, so suppose $k \geq 1$. Let fix n, and define $\hat{\ell}(\mu) = \ell_n^{(1)}(\mu) - \mu$ for $\mu \in [0,1]$. To prove that $\ell_n^{(k)}(\mu)$ is concave over $[2^{-n}, 1]$, it is sufficient to show that $\hat{\ell}(\mu)$ is concave over $[2^{-n}, 1]$, because:

1. If $\hat{\ell}(\mu)$ is concave, then $\hat{\ell}(\mu) + \mu = \ell_n^{(1)}(\mu)$ is concave as well.

2. If $\ell_n^{(1)}(\mu)$ is concave, since it is non-decreasing, by repeated applications of Lemma 9 and Proposition 33, it follows that $\ell_n^{(k)}$ is also concave for all $k \geq 1$ as well, when we limit the inputs to $\mu \geq 2^{-n}$.

Therefore, in the following, we only aim to prove that (1) $\hat{\ell}(\mu)$ is concave over $[2^{-n}, 1]$, and (2) the left and right derivatives of $\hat{\ell}(\mu)$ over $\mu = 2^{-n}$ do not violate its concavity.

In the following, we will fix n and $k = 1$. Because n, k are both fixed, in the rest of the proof of Lemma 34 we do not represent them explicitly as indexes anymore.

It holds that $\hat{\ell}(\beta^{(t)}) = \mathsf{OffIso}(\beta^{(t)})$ for all $t \in \langle n \rangle$. Also, for $\mu \in (\beta^{(t)}, \beta^{(t-1)})$ (recall that $\beta^{(t)} < \beta^{(t-1)}$) where $\mu = \alpha\beta^{(t)} + (1-\alpha)\beta^{(t-1)}$, we have

$$\begin{aligned}\hat{\ell}(\mu) &= \alpha\ell(\beta^{(t)}) + (1-\alpha)\ell(\beta^{(t-1)}) - \alpha\beta^{(t)} - (1-\alpha)\beta^{(t-1)} \\ &= \alpha\mathsf{OffIso}(\beta^{(t)}) + (1-\alpha)\mathsf{OffIso}(\beta^{(t-1)}).\end{aligned}$$

Since the curve $\hat{\ell}$ is linear over every interval $\mu \in [\beta^{(t)}, \beta^{(t-1)}]$ for all $t \in [n+1]$, to prove its concavity, we only have to compare its left and right derivatives at every $\beta^{(t)}, t \in [n]$, where it holds that $\hat{\ell}(\beta^{(t)}) = \mathsf{OffIso}(\beta^{(t)})$. Hence, for all $t \in [n]$, we need to prove the following.

$$\frac{\mathsf{OffIso}(\beta^{(t)}) - \mathsf{OffIso}(\beta^{(t+1)})}{\beta^{(t)} - \beta^{(t+1)}} \geq \frac{\mathsf{OffIso}(\beta^{(t-1)}) - \mathsf{OffIso}(\beta^{(t)})}{\beta^{(t-1)} - \beta^{(t)}} \tag{24}$$

Note that by letting $t = n$ in Inequality (24), we have $\hat{\ell}$ is still concave for point 2^{-n}. We first verify Inequality (24) for extreme cases of $t = 1, n$. If $t = 1$, then Inequality (24) holds because

$$\frac{1-n}{n} = \frac{\mathsf{OffIso}(\beta^{(t)}) - \mathsf{OffIso}(\beta^{(t+1)})}{\beta^{(t)} - \beta^{(t+1)}} \geq \frac{\mathsf{OffIso}(\beta^{(t-1)}) - \mathsf{OffIso}(\beta^{(t)})}{\beta^{(t-1)} - \beta^{(t)}} = \frac{0-1}{1}.$$

If $t = n$, a generalization of Inequality 24 for any k holds because

$$\frac{\sum_{i=0}^{k}\binom{n}{i} - 0}{1-0} = \frac{\ell^{(k)}(\beta^{(t)}) - \ell^{(k)}(\beta^{(t+1)})}{\beta^{(t)} - \beta^{(t+1)}} \geq \frac{\ell^{(k)}(\beta^{(t-1)}) - \ell^{(k)}(\beta^{(t)})}{\beta^{(t-1)} - \beta^{(t)}} = \frac{\binom{n}{k+1}}{n},$$

which in turn is correct because $\sum_{i=0}^{k}\binom{n}{i} > \binom{n}{k} \geq \binom{n}{k+1}/n$.

For the intermediate cases, for all $t \in \{n-1, \ldots, 2\}$, we have to prove:

$$\begin{aligned}\frac{\binom{n}{t-k} - \binom{n}{t}}{\binom{n}{t}} &= \frac{\mathsf{OffIso}(\beta^{(t)}) - \mathsf{OffIso}(\beta^{(t+1)})}{\beta^{(t)} - \beta^{(t+1)}} \\ &\geq \frac{\mathsf{OffIso}(\beta^{(t-1)}) - \mathsf{OffIso}(\beta^{(t)})}{\beta^{(t-1)} - \beta^{(t)}} = \frac{\binom{n}{t-2} - \binom{n}{t-1}}{\binom{n}{t-1}}\end{aligned}$$

which is equivalent to proving the following true statement

$$\begin{aligned}\frac{t}{n-t+1} &= \frac{t!(n-t)!}{(t-1)!(n-t+1)!} = \frac{\binom{n}{t-1}}{\binom{n}{t}} \geq \frac{\binom{n}{t-2}}{\binom{n}{t-1}} \\ &= \frac{(t-1)!(n-t+1)!}{(t-2)!(n-t+2)!} = \frac{t-1}{n-t+2}.\end{aligned}$$

□

The main step of the proof of Theorem 26 is to show the following claim.

Claim 35. *It holds that* $\ell_n^{(k)} \leq \mathsf{Rec}_n^{(k)}\left[\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}\right]$.

Proof. In the following, for simplicity we let $\mathsf{Rec} = \mathsf{Rec}_n^{(k)}\left[\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}\right]$.

Case of exact Hamming ball probabilities. We first prove

$$\forall t \in \langle n \rangle,\ \ell_n^{(k)}(\beta^{(t)}) \leq \mathsf{Rec}(\beta^{(t)}) \tag{25}$$

and then will extend the proof of this inequality to an arbitrary $\mu \in 2^{-n}\langle 2^n]$. We only need to prove Inequality 25 for $t \in [n]$, because $\beta_n^{(n+1)} = 0, \beta^{(0)} = 1$, and so

$$\ell_n^{(k)}(0) = \mathsf{Rec}(0) = 0, \quad \ell_n^{(k)}(1) = \mathsf{Rec}(1) = 1.$$

Recall that $\ell_n^{(k)}\left(\beta_n^{(t)}\right) = \mathsf{OffExp}_n^{(k)}\left(\beta_n^{(t)}\right) = \beta_n^{(t-k)}$. Hence, for $s = s_n^{(t)} = \beta^{(t)} \cdot 2^n$ where $t \in [n]$, our goal is to prove the following

$$\beta_n^{(t-k)} \leq \inf_{(s_0,s_1)\in \mathcal{D}iv_n(s)} \mathsf{Rec}(s_0, s_1). \tag{26}$$

Case studies. Note that $\beta_n^{(t)} 2^n = s_n^{(t)} = s_{n-1}^{(t)} + s_{n-1}^{(t-1)}$ because of the Pascal equality. Also by the definition of $\mathcal{D}iv_n$, we have $s_0 \leq s_1$ and $s_0 + s_1 = s_n^{(t)}$ for any choice of $(s_0, s_1) \in \mathcal{D}iv_n(s)$ in the right hand side of Equation 26. Then, one of the following three cases must hold: (1) $s_0 = s_{n-1}^{(t)}$, (2) $s_0 < s_{n-1}^{(t)}$, or (3) $s_0 > s_{n-1}^{(t)}$. Hence, we divide our analysis to the same three cases, and then prove that $\beta_n^{(t-k)} \leq \mathsf{Rec}(s_0, s_1)$ holds in all of them. We will also use the Pascal equality in the form of $\beta_n^{(t-k)} = (\beta_{n-1}^{(t-k-1)} + \beta_{n-1}^{(t-k)})/2$.

1. $s_{n-1}^{(t)} = s_0 < s_1 = s_{n-1}^{(t-1)}$. In this case, we have

$$\ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right) = \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right) = \beta_{n-1}^{(t-k)}$$

   which, informally speaking means that, it does not matter if the adversary intervenes to change 0 to 1 when the first bit is fixed to 0. Formally, we have

$$\begin{aligned}\mathsf{Rec}(s_0, s_1) &= \mathsf{Rec}(s_{n-1}^{(t)}, s_{n-1}^{(t-1)}) \\ &= \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right), \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \\ &= \frac{\beta_{n-1}^{(t-k-1)} + \beta_{n-1}^{(t-k)}}{2} = \beta_n^{(t-k)}.\end{aligned}$$

2. $s_{n-1}^{(t)} < s_0 \leq s_1 < s_{n-1}^{(t-1)}$. Informally speaking, in this case the adversary does not change the bit and

we use the piece-wise linearity of the ℓ function on $[\beta_{n-1}^{(t)}, \beta_{n-1}^{(t-1)}]$. More formally,

$$\begin{aligned}
\mathsf{Rec}(s_0, s_1) &= \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right), \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \\
&\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right)}{2} \\
&= \frac{\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right) + \ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t)}}{2^{n-1}}\right)}{2} \text{ (by piece-wise linearity of } \ell_{n-1}^{(k)}) \\
&= \mathsf{Rec}\left(s_{n-1}^{(t)}, s_{n-1}^{(t-1)}\right) = \beta_n^{(t-k)}.
\end{aligned}$$

3. $s_0 < s_{n-1}^{(t)} < s_{n-1}^{(t-1)} < s_1$. Informally speaking, in this case the adversary does change the bit 0 into 1, and we also use the fact that $\ell_{n-1}^{(k)}$ is monotone. More formally,

$$\begin{aligned}
\mathsf{Rec}(s_0, s_1) &= \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right), \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \\
&\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)}{2} \\
&\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right) + \ell_{n-1}^{(k-1)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right)}{2} \text{ (by monotonicity of } \ell_{n-1}^{(k)}) \\
&= \mathsf{Rec}\left(s_{n-1}^{(t)}, s_{n-1}^{(t-1)}\right) = \beta_n^{(t-k)}.
\end{aligned}$$

Case of other probabilities. Here we no longer assume that $\mu = \beta^{(t)}$ for some $t \in [n]$, and assume $\mu = \alpha\beta_n^{(t)} + (1-\alpha)\beta_n^{(t-1)}$ for some $t \in [n+1]$ and $0 < \alpha < 1$. Recall that $\beta^{(t)}2^n = s_n^{(t)} = s_{n-1}^{(t)} + s_{n-1}^{(t-1)}$ and $\beta^{(t-1)}2^n = s_n^{(t-1)} = s_{n-1}^{(t-1)} + s_{n-1}^{(t-2)}$. We define

$$s_0' = \alpha \cdot s_{n-1}^{(t)} + (1-\alpha) \cdot s_{n-1}^{(t-1)}, \quad s_1' = \alpha \cdot s_{n-1}^{(t-1)} + (1-\alpha) \cdot s_{n-1}^{(t-2)}.$$

By the definition of μ, it holds that $\mu \cdot 2^n = s = s_0' + s_1'$ because

$$s_0' + s_1' = \alpha \cdot \left(s_{n-1}^{(t)} + s_{n-1}^{(t-1)}\right) + (1-\alpha) \cdot \left(s_{n-1}^{(t-1)} + s_{n-1}^{(t-2)}\right) = \alpha \cdot s_n^{(t)} + (1-\alpha) \cdot s_n^{(t-1)} = s.$$

In general, s_0', s_1' are not integers, but intuitively, $s_0' + s_1'$ gives the critical way of splitting s into two numbers at which the replacing and no-replacing strategies give the same bound and we can do the case studies. (In particular s_0', s_1' take the role of $s_{n-1}^{(t)}, s_{n-1}^{(t-1)}$ when we previously assumed that $\mu = \beta^{(t)}$.)

Useful observations. By the piecewise linearity of $\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}$ we have

$$\ell_{n-1}^{(k)}\left(\frac{s_0'}{2^{n-1}}\right) = \alpha\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t)}}{2^{n-1}}\right) + (1-\alpha)\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right),$$

$$\ell_{n-1}^{(k)}\left(\frac{s_1'}{2^{n-1}}\right) = \alpha\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right) + (1-\alpha)\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t-2)}}{2^{n-1}}\right),$$

$$\ell_{n-1}^{(k-1)}\left(\frac{s_1'}{2^{n-1}}\right) = \alpha\ell_{n-1}^{(k-1)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right) + (1-\alpha)\ell_{n-1}^{(k-1)}\left(\frac{s_{n-1}^{(t-2)}}{2^{n-1}}\right),$$

$$\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t)}}{2^{n-1}}\right) = \beta_{n-1}^{(t-k)} = \ell_{n-1}^{(k-1)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right),$$

$$\text{and} \quad \ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right) = \beta_{n-1}^{(t-k-1)} = \ell_{n-1}^{(k-1)}\left(\frac{s_{n-1}^{(t-2)}}{2^{n-1}}\right).$$

Therefore, we get the following.

$$\ell_{n-1}^{(k)}\left(\frac{s_0'}{2^{n-1}}\right) = \ell_{n-1}^{(k-1)}\left(\frac{s_1'}{2^{n-1}}\right) = \alpha\beta_{n-1}^{(t-k)} + (1-\alpha)\beta_{n-1}^{(t-k-1)}, \tag{27}$$

$$\ell_{n-1}^{(k)}\left(\frac{s_1'}{2^{n-1}}\right) = \alpha\beta_{n-1}^{(t-k-1)} + (1-\alpha)\beta_{n-1}^{(t-k-2)}. \tag{28}$$

Case studies. We now again partition into three different categories and separately prove that $\ell_n^{(k)}(\mu) \leq \mathsf{Rec}(s_0, s_1)$ holds for each category.

1. $s_0' = s_0 < s_1 = s_1'$. In this case, using Equations (27) and (28) we get

$$\begin{aligned}
&\mathsf{Rec}(s_0', s_1') \\
&= \frac{\ell_{n-1}^{(k)}\left(\frac{s_1'}{2^{n-1}}\right)}{2} + \frac{\max\left\{\ell_{n-1}^{(k)}\left(\frac{s_0'}{2^{n-1}}\right), \ell_{n-1}^{(k-1)}\left(\frac{s_1'}{2^{n-1}}\right)\right\}}{2} \\
&= \frac{\alpha\cdot\beta_{n-1}^{(t-k-1)} + (1-\alpha)\cdot\beta_{n-1}^{(t-k-2)}}{2} + \frac{\alpha\cdot\beta_{n-1}^{(t-k)} + (1-\alpha)\cdot\beta_{n-1}^{(t-k-1)}}{2} \\
&= \alpha\cdot\beta_n^{(t-k)} + (1-\alpha)\cdot\beta_n^{(t-k-1)} \\
&= \alpha\cdot\ell_n^{(k)}(\beta^{(t)}) + (1-\alpha)\cdot\ell_n^{(k)}(\beta^{(t-1)}) = \ell_n^{(k)}(\mu).
\end{aligned}$$

2. $s_0' < s_0 \leq s_1 < s_1'$. Informally speaking, in this case the adversary does not tamper and leave the bit 0 unchanged. We will use the fact that $\ell_{n-1}^{(k)}$ is *concave*, which was proved in Lemma 34. Note that in the corresponding Case 2 when the probability μ was that of an exact ball ($\mu = \beta_n^{(t)}$) we could have also used the fact that $\ell_{n-1}^{(k)}$ is concave, but in that case we only used the concavity over a linear part of $\ell_{n-1}^{(k)}$. However, in our current case, we could no longer only rely on the piecewise linearity of $\ell_{n-1}^{(k)}$

and we would use its concavity. More formally,

$$\begin{aligned}
\mathsf{Rec}(s_0, s_1) &= \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right), \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \\
&\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right)}{2} \\
&\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s'_1}{2^{n-1}}\right) + \ell_{n-1}^{(k)}\left(\frac{s'_0}{2^{n-1}}\right)}{2} \text{ (by } \textbf{concavity} \text{ of } \ell_{n-1}^{(k)}) \\
&= \mathsf{Rec}(s'_0, s'_1) = \ell_n^{(k)}(\mu).
\end{aligned}$$

3. $s_0 < s'_0 < s'_1 < s_1$. Informally speaking, in this case the adversary does change the bit 0 into 1, and we rely on the monotonicity of $\ell_{n-1}^{(k)}$. More formally,

$$\begin{aligned}
\mathsf{Rec}(s_0, s_1) &= \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right), \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \\
&\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)}{2} \\
&\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s'_1}{2^{n-1}}\right) + \ell_{n-1}^{(k-1)}\left(\frac{s'_1}{2^{n-1}}\right)}{2} \text{ (by monotonicity of } \ell_{n-1}^{(k)}) \\
&= \mathsf{Rec}(s'_0, s'_1) = \ell_n^{(k)}(\mu).
\end{aligned}$$

□

Claim 36. $\ell_n^{(k)} \leq \mathsf{OnExp}_n^{(k)}$.

Proof. The proof is by induction on n. The claim hold for $n = 0$. Using Claim 31 and 35 and induction we get:

$$\mathsf{OnExp}_n^{(k)} \geq \mathsf{Rec}_n^{(k)}\left[\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}\right] \geq \ell_n^{(k)}.$$

□

Now we can finish the proof of Theorem 26. If $\mu = \beta_n^{(t)}$ for some $t \in \langle n \rangle$, it then always holds that $\ell_n^{(k)}(\mu) \geq \mathsf{OnExp}_n^{(k)}(\mu)$ simply because $\ell_n^{(k)}(\mu)$ describes how much one particular protocol (i.e., τ_t) can bound adversary's power, while $\mathsf{OnExp}_n^{(k)}(\mu)$ is equal to the *minimum* of the same quantity among all protocols. Therefore, by Claim 36, $\mathsf{OnExp}_n^{(k)}\left(\beta_n^{(t)}\right) = \ell_n^{(k)}\left(\beta_n^{(t)}\right) = \beta_n^{(t-k)}$. □

Relaxing the last message to non-binary. Here we discuss an extension to Theorem 26 that follows essentially from the same proof. Theorem 26 shows that online attacks are as powerful as offline attacks when we focus on protocols with uniform binary messages. Now, suppose we allow the last message of the protocol to be an arbitrary long message, while every other message is supposed to be a uniform bit. We refer to such protocols as *binary-except-last-message* (BELM) protocols. Note that BELM protocols constitute a *larger* set of protocols, and hence they potentially could include more robust protocols that further limits the power of (offline or online) attacks. We observe that, essentially the same proof as that of Theorem 26 shows that we can strengthen Theorem 26 as follows.

Theorem 37 (Informally stated: extending Theorem 26 to BELM protocols)**.** *Suppose a random process* $\mathbf{w}_{\leq n} = (\mathbf{w}_1, \ldots, \mathbf{w}_n)$ *has the property that all the first* $n-1$ *blocks are independent and uniform random bits, and suppose* f *is a Boolean function defined over this random process. Suppose* $\Pr[f(\mathbf{w}_{\leq n}) = 1] = \beta_n^{(t)}$ *for some* $t \in [n]$*. Then, there is an online* k*-replacing adversary over* $\mathbf{u}_{\leq n}$ *that generates joint random process* $(\mathbf{u}_{\leq n}, \mathbf{v}_{\leq n})$ *with* $\mathbf{v}_{\leq n}$ *being the output process, such that* $\Pr[f(\mathbf{w}_{\leq n}) = 1] \geq \beta_n^{(t-k)}$*. Note that this is optimal in a strong sense: there is a fully binary protocol (i.e., the threshold function* τ_t*) for which even* offline k*-replacing adversaries are limited to achieve offline expansion at most* $\beta_n^{(t-k)}$*.*

Proof Sketch. The proof of the above improved variant of Theorem 26 relies on two observations. One of them is the basis of the induction, when $n = 1$, and the other one is the improved induction step which follows from the improve variant of Claim 35 as explained below.

Relaxing transformation of Definition 28. Claim 35 was the heart of the proof of Theorem 26. In this claim, we deal with the recursion of Eq. (23) which is defined by splitting integer s into smaller integers, computing some recursive expansions and taking the minimum. It is easy to see that Claim 35 holds even if we relax the way we split s into smaller quantities and pick such pairs as *real* values

$$\widetilde{\mathcal{D}iv}_{n-1}(s) = \left\{(s_0, s_1) \mid s_0, s_1 \in \mathbb{R}, 0 \leq s_0 \leq s_1 \leq 2^{n-1}, s = s_0 + s_1\right\}.$$

In particular, let $\widetilde{\mathsf{Rec}}_n^{[k]}$ be the similar transformation using this relaxed variant $\widetilde{\mathcal{D}iv}_{n-1}(s)$ instead. First, note that by this relaxation instead, we might end up getting *smaller* expansions; namely, $\widetilde{\mathsf{Rec}}_n^{(k)} \leq \mathsf{Rec}_n^{(k)}$. Yet, the same proof shows that Claim 35 holds even if we use $\widetilde{\mathsf{Rec}}_n^{(k)}$ instead of $\mathsf{Rec}_n^{(k)}$. Moreover, in (both variants of) Case 1, it is now always possible to achieve the equality using some pair in $\widetilde{\mathcal{D}iv}_{n-1}(s)$. Therefore, this time we obtain a slightly stronger statement than that of Claim 35 for BELM protocols as follows.

Claim 38 (Variant of Claim 35 for BELM protocols)**.** $\ell_n^{(k)} = \widetilde{\mathsf{Rec}}_n^{(k)}[\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}]$.

The proof of the claim above is identical to that of Claim 35. □

4.1 Making the attack run in polynomial time

In this subsection, we describe how using more ideas, one can extend the attacks of Section 4 to run in polynomial time.

Theorem 39 (Polynomial-time variant of Theorem 26)**.** *For all* $n \in \mathbb{N}, k \leq t$*, let* $\ell_n^{(k)}(\cdot)$ *(see Definition 32) be the lower bound of the online expansion achieved by the attack of Theorem 26. Then, there is an online* k*-replacing adversary* A *who, for every* $\varepsilon, \mu \in (0,1)$*, runs in time* $\mathrm{poly}(\log(1/\delta)n/\varepsilon)$ *and achieves the following: with probability* $1-\delta$ *over the randomness* r *of* A*, if we fix* r *we get the following*

$$\mathsf{OnExp}_n^{(\mathsf{A})}(\mu) \geq \ell_n^{(k)}(\mu - \varepsilon).$$

Before proving Theorem 39 we derive some corollaries.

Theorem 39 above shows that a polynomial time adversary can get essentially the same lower bound on the online k-expansion probability, but the lower bound is applied to $\mu - \varepsilon$ rather than μ. Hence, one can hope to get arbitrarily close to $\ell_n^{(k)}(\mu)$, if the derivative of $\ell_n^{(k)}(\mu)$ is bounded around μ. In particular, one might hope to prove this when μ is not too small. (This is inherent, as μ could be exponentially small, in which case a polynomial-time adversary might have no way to find any point $x \leftarrow \mathbf{U}_n$ that falls into the target set $\mathcal{S}$ (or equivalently $f(x) = 1$ when f is the characteristic function of $\mathcal{S}$).

Lemma 40 (Bound on derivative of the lower bound function)**.** *Let $k \leq n$, $t \in [n+1]$, and $\mu \in (\beta_n^{(k)}(t), \beta_n^{(k)}(t-1))$. Then, for any $\mu' \geq \mu - \varepsilon$, it holds that $\ell_n^{(k)}(\mu') \geq \ell_n^{(k)}(\mu) - \varepsilon/\mu$. (In particular, the left derivative of $\ell_n^{(k)}$ at μ is at most $1/\mu$.)*

Proof. In fact, a stronger condition holds: $\ell_n^{(k)}(\mu') \geq \ell_n^{(k)}(\mu) \cdot (1 - \varepsilon/\mu)$. We will only use that (1) $\ell_n^{(k)}(\cdot)$ is concave (by Lemma 34), (2) $\ell_n^{(k)}(0) = 0$, and that (3) $\ell_n^{(k)}(\mu) \leq 1$.

Suppose we connect the origin $(0,0)$ to $(\mu, \ell_n^{(k)}(0))$ and denotes this line segment as curve e. It holds that $e \leq c$, where c is the curve $\ell_n^{(k)}(\alpha)$ for $\alpha \in [0, \mu]$, because c is concave. Furthermore, the left derivative of the curve c at μ would be upper bounded by the derivative of line e at μ. Because $\ell_n^{(k)}(0) = 0$ and that $\ell_n^{(k)}(\mu) \leq 1$, the derivative of curve e is $\ell_n^{(k)}(\mu)/\mu$. Moreover, since the curve e is below c, it holds that

$$\ell_n^{(k)}(\mu') \geq e(\mu') \geq e(\mu - \varepsilon) = \ell_n^{(k)}(\mu) \cdot (1 - \varepsilon/\mu).$$

□

Theorem 39 and Lemma 40 imply the following corollary.

Corollary 41. *In the same setting of Theorem 39, there is an adversary* A *who also gets μ as input, runs in time* $\mathrm{poly}(\log(1/\delta)n/(\mu\varepsilon))$ *and achieves the following for all $\mu \in [0,1]$. With probability $1-\delta$ over the randomness r of* A*, if we fix r, it holds that:*

$$\mathsf{OnExp}_n^{(\mathsf{A})}(\mu) \geq \ell_n^{(k)}(\mu) - \varepsilon.$$

Proof. By Lemma 40, if $\mu' \geq \mu - \varepsilon'$, for $\varepsilon' = \varepsilon \cdot \mu$, it implies that $\ell_n^{(k)}(\mu') \geq \ell_n^{(k)}(\mu) - \varepsilon'/\mu = \ell_n^{(k)}(\mu) - \varepsilon$. So, to obtain the attack of Corollary 41 with parameter ε, all we need to do is to run the adversary of Theorem 39 with parameter ε' equal to $\varepsilon \cdot \mu$. This leads to running time $\mathrm{poly}(\log(1/\delta)n/(\varepsilon\mu))$ as well. □

In the rest of this subsection, we prove Theorem 39. At a high level, the polynomial time attack follows the following ideas. Note that we already have established in Theorem 26 that $\ell_n^{(k)}$ is a lower bound on the expansion of optimal information theoretic k-replacing attacks. So, we use this in our polynomial time attacks. We will use $\ell_n^{(k)}$ as the guaranteed bound in our decision to change or not to change a bit during the attack. Then, we will use Claim 35 and Claim 29 as black-box and show that the attack still achieves what we want. One big catch is that we do not have access to the true partial averages of a prefix during the attack. We will only use approximate valued, and show that they still lead to reasonable bounds, by induction.

Construction 42 (Polynomial k-replacing attack A_k on uniform bit messages)**.** This attack uses the additive ε'-approximate oracle $\tilde{f}(\cdot)$, so that for all $u_{\leq i} \in \{0,1\}^i$, it holds that $\tilde{f}(u_{\leq i}) \in \bar{f}(u_{\leq i}) \pm \varepsilon'$.

Given a state $\sigma_i = (u_{\leq i}, v_{\leq i})$ where $u_{\leq i}$ are the original values, $v_{\leq i}$ are the output values under attack , and u_{i+1} is sampled uniformly at random from $\{0,1\}$, the algorithm A at time $i+1$ decides whether to keep or to replace u_{i+1}. Let $d = \mathrm{HD}(u_{\leq i}, v_{\leq i})$ be the number of replacements already made to the sequence.

- (Case 0) If $d \geq k$, do not change u_{i+1} and output $v_{i+1} = u_{i+1}$.
- (Case 1) if Case 0 does not happen and

$$\ell^{(k-d)}\left(\tilde{f}(v_{\leq i}, u_{i+1}) - \varepsilon' \cdot (2(n-i)-1)\right) < \ell^{(k-1-d)}\left(\tilde{f}(v_{\leq i}, \bar{u}_{i+1}) - \varepsilon' \cdot (2(n-i)-1)\right),$$

  make a replacement and output $v_{i+1} = \bar{u}_{i+1}$.

- (Case 2) If Cases 0 and 1 do not happen, do not change u_{i+1} and output $v_{i+1} = u_{i+1}$.

Here, $\bar{u}_{i+1} = 1 - u_{i+1}$ is the complement of u_{i+1}.

We then use the adversary A in Construction 42 to prove Theorem 39.

Proof of Theorem 39. We prove $\mathsf{OnExp}_n^{(A_k)}(\mu) \geq \ell_n^{(k)}(\mu - \varepsilon' \cdot 2n)$ (i.e., the online expansion of adversary A in Construction 42 holds) by performing induction on n and k, where $\mu = f(v_{\leq i})$ is the current expansion with no adversary. Then by letting $\varepsilon' = \varepsilon/2n$, we obtain Theorem 39.

For the base case, no attack can possibly happen when $k = 0$, which indicates $\ell_n^{(0)}(\mu) = \ell_n^{(\mathsf{A}_0)}(\mu) = \mu$. Similarly for $n = 0$, $\ell_0^{(k)}(\mu) = \ell_0^{(\mathsf{A}_k)}(\mu) = \mu$. Therefore, $\mathsf{OnExp}_n^{(\mathsf{A}_k)}(\mu) \geq \ell_n^{(k)}(\mu - \varepsilon' \cdot 2n)$ holds for $n = 0$ and $k = 0$.

Now, consider the adversary A has total budget k' on a sequence of n' uniform bits, and has state $\sigma_i = (u_{\leq i}, v_{\leq i})$ and a new bit u_{i+1}. Let $k = k' - \mathrm{HD}(u_{\leq i}, v_{\leq i})$ be the remaining budget of A, and $n = n' - i$ be the number of random variables left to sample. Now, let $\mu_0 = \bar{f}(v_{\leq i}, 0)$ be the expected value when $v_{i+1} = 0$, and $\mu_1 = \bar{f}(v_{\leq i}, 1)$ be the expected value when $v_{i+1} = 1$. Note that A only has access to $\tilde{f}(\cdot)$ instead of $\bar{f}(\cdot)$. We then let $\tilde{\mu}_0 = \tilde{f}(v_{\leq i}, 0)$ and $\tilde{\mu}_1 = \tilde{f}(v_{\leq i}, 1)$. By the induction hypothesis, we have $\mathsf{OnExp}_n^{(\mathsf{A}_k)}(\mu_0) \geq \ell_n^{(k)}(\mu_0 - \varepsilon' \cdot 2(n-1)) \geq \ell_n^{(k)}(\tilde{\mu}_0 - \varepsilon' \cdot (2n-1))$ and $\mathsf{OnExp}_n^{(\mathsf{A}_k)}(\mu_1) \geq \ell_n^{(k)}(\mu_1 - \varepsilon' \cdot 2(n-1)) \geq \ell_n^{(k)}(\tilde{\mu}_1 - \varepsilon' \cdot (2n-1))$. Similarly, we have $\mathsf{OnExp}_n^{(\mathsf{A}_{k-1})}(\mu_0) \geq \ell_n^{(k-1)}(\tilde{\mu}_0 - \varepsilon' \cdot (2n-1))$ and $\mathsf{OnExp}_n^{(\mathsf{A}_{k-1})}(\mu_1) \geq \ell_n^{(k-1)}(\tilde{\mu}_1 - \varepsilon' \cdot (2n-1))$.

Finally, by applying Claim 35 and Claim 29, we have

$$
\begin{aligned}
\mathsf{OnExp}_n^{(\mathsf{A}_k)}(\mu) &= \mathsf{Rec}_n^{(k)}[\mathsf{OnExp}_{n-1}^{(\mathsf{A}_k)}, \mathsf{OnExp}_{n-1}^{(\mathsf{A}_{k-1})}](\mu) \\
\text{(by Definition 28)} \quad &\geq \mathsf{Rec}_n^{(k)}[\mathsf{OnExp}_{n-1}^{(\mathsf{A}_k)}, \mathsf{OnExp}_{n-1}^{(\mathsf{A}_{k-1})}](\mu_0, \mu_1) \\
\text{(by Claim 29)} \quad &\geq \mathsf{Rec}_n^{(k)}[\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}]\left(\tilde{\mu}_0 - \varepsilon' \cdot (2n-1), \tilde{\mu}_1 - \varepsilon' \cdot (2n-1)\right) \\
\text{(by Claim 35)} \quad &\geq \ell_n^{(k)}\left(\frac{\tilde{\mu}_0 + \tilde{\mu}_1}{2} - \varepsilon' \cdot (2n-1)\right) \\
&\geq \ell_n^{(k)}\left(\frac{\mu_0 - \varepsilon' + \mu_1 - \varepsilon'}{2} - \varepsilon' \cdot (2n-1)\right) \\
&= \ell_n^{(k)}\left(\mu - \varepsilon' \cdot 2n\right).
\end{aligned}
$$

□

References

[AM80] D Amir and VD Milman. Unconditional and symmetric sets in n-dimensional normed spaces. *Israel Journal of Mathematics*, 37(1-2):3–20, 1980. 5

[BGZ16] Iddo Bentov, Ariel Gabizon, and David Zuckerman. Bitcoin beacon. *arXiv preprint arXiv:1605.04559*, 2016. 10

[BHMO18] Amos Beimel, Iftach Haitner, Nikolaos Makriyannis, and Eran Omri. Tighter bounds on multi-party coin flipping via augmented weak martingales and differentially private sampling. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 838–849. IEEE, 2018. 10

[Blu84] Manuel Blum. How to exchange (secret) keys. *ACM Transactions on Computer Systems*, 1:175–193, 1984. 2

[BNS+06] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25. ACM, 2006. 39

[BOL89] M. Ben-Or and N. Linial. Collective coin flipping. *Advances in Computing Research*, 5:91–115, 1989. 4, 10

[BOL90] Michael Ben-Or and Nathan Linial. Collective coin flipping. *Advances in Computing Research*, 5:91–115, 1990. 2, 10

[CI93] Richard Cleve and Russell Impagliazzo. Martingales, collective coin flipping and discrete control processes. *Manuscript*, 1993. 4, 5, 44, 45

[Cle86] Richard Cleve. Limits on the security of coin flips when half the processors are faulty. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 364–369. ACM, 1986. 10

[DKK+16] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 655–664. IEEE, 2016. 40

[DMM14] Dana Dachman-Soled, Mohammad Mahmoody, and Tal Malkin. Can optimally-fair coin tossing be based on one-way functions? In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 217–239, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany. 10

[Dod01] Yevgeniy Dodis. New imperfect random source with applications to coin-flipping. In *International Colloquium on Automata, Languages, and Programming*, pages 297–309. Springer, 2001. 10

[DSLMM11] Dana Dachman-Soled, Yehuda Lindell, Mohammad Mahmoody, and Tal Malkin. On the black-box complexity of optimally-fair coin tossing. In *Theory of Cryptography Conference*, pages 450–467. Springer, 2011. 10

[EMM20] Omid Etesami, Saeed Mahloujifar, and Mohammad Mahmoody. Computational concentration of measure: Optimal bounds, reductions, and more. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 345–363. SIAM, 2020. 3, 4, 5, 6, 7, 40

[GKP15] Shafi Goldwasser, Yael Tauman Kalai, and Sunoo Park. Adaptively secure coin-flipping, revisited. In *International Colloquium on Automata, Languages, and Programming*, pages 663–674. Springer, 2015. 2, 5, 11

[Har66] Lawrence H Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1(3):385–393, 1966. 5, 6, 23

[HKH20] Iftach Haitner and Yonatan Karidi-Heller. A tight lower bound on adaptively secure full-information coin flip. *Foundations of Computer Science (FOCS), IEEE 61st Annual Symposium on*, 2020. 3, 5, 10

[HMO18] Iftach Haitner, Nikolaos Makriyannis, and Eran Omri. On the complexity of fair coin flipping. In *Theory of Cryptography Conference*, pages 539–562. Springer, 2018. 10

[HMRAR13] Michel Habib, Colin McDiarmid, Jorge Ramirez-Alfonsin, and Bruce Reed. *Probabilistic methods for algorithmic discrete mathematics*, volume 16. Springer Science & Business Media, 2013. 12

[HNO+18] Iftach Haitner, Kobbi Nissim, Eran Omri, Ronen Shaltiel, and Jad Silbak. Computational two-party correlation: A dichotomy for key-agreement protocols. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 136–147. IEEE, 2018. 10

[HT17] Iftach Haitner and Eliad Tsfadia. An almost-optimally fair three-party coin-flipping protocol. *SIAM Journal on Computing*, 46(2):479–542, 2017. 10

[KAH88] J KAHN. The influence of variables on boolean functions. In *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science, 1988*, 1988. 10

[KKR18] Yael Tauman Kalai, Ilan Komargodski, and Ran Raz. A lower bound for adaptively-secure collective coin-flipping protocols. In *32nd International Symposium on Distributed Computing*, 2018. 3, 5, 6, 7, 10

[KMM19] Hamidreza Amini Khorasgani, Hemanta K Maji, and Tamalika Mukherjee. Estimating gaps in martingales and applications to coin-tossing: constructions and hardness. In *Theory of Cryptography Conference*, pages 333–355. Springer, 2019. 4, 5, 10, 45

[KMW21] Hamidreza Amini Khorasgani, Hemanta K Maji, and Mingyuan Wang. Optimally-secure coin-tossing against a byzantine adversary. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 2858–2863. IEEE, 2021. 4, 5, 6, 9, 10

[LLS89] David Lichtenstein, Nathan Linial, and Michael Saks. Some extremal problems arising from discrete control processes. *Combinatorica*, 9(3):269–287, 1989. 3, 5

[LRV16] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 665–674. IEEE, 2016. 40

[Mar74] Grigorii Aleksandrovich Margulis. Probabilistic characteristics of graphs with large connectivity. *Problemy peredachi informatsii*, 10(2):101–108, 1974. 5

[McD89] Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989. 5

[MDM18] Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. Learning under p-Tampering Attacks. In *ALT*, pages 572–596, 2018. 5, 10, 39, 40

[MDM19] Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. *AAAI Conference on Artificial Intelligence*, 2019. 39

[MM17] Saeed Mahloujifar and Mohammad Mahmoody. Blockwise p-tampering attacks on cryptographic primitives, extractors, and learners. In *Theory of Cryptography Conference*, pages 245–279. Springer, 2017. 5, 10, 39

[MM19] Saeed Mahloujifar and Mohammad Mahmoody. Can adversarially robust learning leverage computational hardness? In Aurélien Garivier and Satyen Kale, editors, *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, pages 581–609, Chicago, Illinois, 22–24 Mar 2019. PMLR. 3, 4, 5, 6, 7, 8, 15, 39, 40

[MMM19] Saeed Mahloujifar, Mohammad Mahmoody, and Ameer Mohammed. Universal multi-party poisoning attacks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 4274–4283, 2019. 5, 10, 39, 40

[MNS09] Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. In *TCC*, pages 1–18, 2009. 10

[MS86] Vitali D Milman and Gideon Schechtman. *Asymptotic theory of finite dimensional normed spaces*, volume 1200. Springer Verlag, 1986. 5

[MW20] Hemanta K. Maji and Mingyuan Wang. Black-box use of one-way functions is useless for optimal fair coin-tossing. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 593–617. Springer, 2020. 10

[RSZ02] Alexander Russell, Michael Saks, and David Zuckerman. Lower bounds for leader election and collective coin-flipping in the perfect information model. *SIAM Journal on Computing*, 31(6):1645–1662, 2002. 10

[SHN+18] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792*, 2018. 39

[STS16] Shiqi Shen, Shruti Tople, and Prateek Saxena. A uror: defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 508–519. ACM, 2016. 39

[Tal95] Michel Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, 81(1):73–205, 1995. 5, 40

A Applications

In this section, we provide more details for applications of our main results of Theorem 1 and 15.

A.1 Targeted poisoning attacks on learners

Let L be a learning algorithm that gets a data set $\mathcal{S}$ and produces a model θ. A data poisoning attack [BNS+06] is one that aims to damage θ (this can be modeled in many different ways) through minimal changes to $\mathcal{S}$. Now, let $P_1, \ldots, P_n$ be n parties that, perhaps interactively, share their (training) data sets $\mathcal{S}_1, \ldots, \mathcal{S}_n$ one by one in n rounds, and at the end a central algorithm L deterministically produces a model θ based on $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_n\}$. Let $b = 1$ if some bad Boolean property holds over θ (e.g., failing to correctly classify a particular point x, leading to a *targeted poisoning attack* [BNS+06, STS16], or have certain level of overall risk). Then, if $\Pr[b = 1] = \mu$ holds before any attacks (in this case μ can be a small value to begin with, something like 0.01 or $1/n$), then our targeted attack can increase $\Pr[b = 1]$ (i.e., that the bad property B holds over θ) by at least $\Omega(k \cdot \mu/\sqrt{n})$ while changing the messages of $k = O(\sqrt{n})$ of the parties.

One key feature of the above attack is that, similarly to the works of [MM17, MDM18, SHN+18, MDM19, MM19], it only uses so-called "clean labels" in its poison data, simply because the adversary never steps outside the support set of the messages supported by the random process of the coin tossing protocol. Also, since our attack is polynomial-time, it can even rule out the possibility of achieving *computational* robustness by relying on computational intractability assumptions. As a special case (when we combine the parties into a single party), the same argument above can be applied to the *single* party learning systems as well, in which our attacker can change $k/\sqrt{n}$ of the examples of a single learning deterministic algorithm and increase the probability of any bad even $\mathcal{B}$ from μ to $\mu + \Omega(k \cdot \mu/\sqrt{n})$. More formally, we obtain the following theorem as a corollary.

Theorem 43 (Clean-label targeted poisoning with few tamperings on any learner)**.** *Let $\mathcal{X}$ be an input space, $\mathcal{Y}$ a label space, L a deterministic learning algorithm, $\mathcal{H}$ the hypothesis space (i.e., L produces a function in $\mathcal{H}$), D a distribution over trained examples $\mathcal{X} \times \mathcal{Y}$, $m \in \mathbb{N}$ a sample complexity bound. Suppose the training data $\mathcal{S}$ is partitioned into n subsets $\mathcal{S} = \cup \mathcal{S}_i$ which are generated in n rounds, aggregated and then fed to the learner L. Suppose $k < n$ the budget of the poisoning adversary. Suppose, for some predicate $\mathcal{B}$, the probability of L outputting a function that satisfies $\mathcal{B}$ is at least μ:*

$$\Pr_{\mathcal{S} \leftarrow D^n, h = L(\mathcal{S})}[h \in \mathcal{B}] \geq \mu$$

Then there is an online *adversary* A *who get to see the training examples in $\mathcal{S}$ one by one, and then changes at most k of the examples, and increase the probability of h having property $\mathcal{B}$ by at least $\Omega(k/\sqrt{n})$. Moreover, if $\mathcal{B}$ is PPT testable, given oracle access to samples from D, the attack can be implemented in polynomial.*

The proof of the theorem above is immediate, once we interpret the i'th training subset as the message w_i, output 1 if $h \in \mathcal{B}$, and apply our k-replacing attack of Theorem 15. In particular, the sampling oracle for D will allow A to do random continuations of the random process, and the polynomial-time testability of $\mathcal{B}$ allows A to have an oracle access to the Boolean function determining the "output bit".

Related work on poisoning attacks. The connection between targeted poisoning attacks and their connection to adaptive coin tossing was studied previously in [MM19, MDM19]. However, all the previous attacks require at least $O(\sqrt{n})$ number of corrupting to get meaningful bounds. Prior to that, the work of [MDM18, MMM19] also studied the effect of static coin tossing attacks on machine learning protocols. Importantly,

similar to our result, the attack of [MM19, EMM20, MMM19, MDM18] also run in polynomial time given access to a data sampler and the training algorithm. Another line of work that tackles with computational aspects of robust machine learning, started with the exciting works of [DKK+16, LRV16] that shows how to deal with outliers for certain statistical tasks, and importantly do so in polynomial time. The distinction of our work with this line of work on robust statistics is that we care about computational aspects of the "attacker" while they care about making the learning algorithms polynomial time.

A.2 Computational isoperimetry in product spaces

Consider a random variable $\mathbf{w}_{\leq n}$ consisting of n independently sampled random variables $(\mathbf{w}_1 \times \cdots \times \mathbf{w}_n) \equiv \mathbf{w}_{\leq n}$, and let $(\mathbf{w}_{\leq n}, \mathrm{HD})$ be the n-dimensional metric probability space in which HD be the Hamming distance between any two $w_{\leq n} = (w_1, \ldots, w_n)$ and $w'_{\leq n} = (w'_1, \ldots, w'_n)$, defined as $\mathrm{HD}(w_{\leq n}, w'_{\leq n}) = |\{i \mid w_i \neq w'_i\}|$. Then, a basic question about such spaces is how quickly events will expand. Namely, let $\mathcal{E} \subseteq \mathrm{Supp}(\mathbf{w}_{\leq n})$ be an event of measure μ, and let $\mathcal{E}_k = \{w_{\leq n} \mid \exists\, v_{\leq n} \in \mathcal{E}, \mathrm{HD}(w_{\leq n}, v_{\leq n}) \leq k\}$ be its k expansion. How big is the probability of $\mathcal{E}_k$ based on n, μ? Concentration of measure in product spaces [Tal95] state that the probability of $\mathcal{E}_k$ quickly converges to ≈ 1 for $k \approx \sqrt{n}$ (ignoring logarithmic terms) as long as $\mu \geq 1/\operatorname{poly}(n)$. Such results are tightly related to isoperimetric inequalities in such spaces [Tal95] in which only a lower bound on the "boundary" of the set $\mathcal{E}$ is proved. In the discrete setting, the boundary simply becomes the same as $\mathcal{E}_1 \setminus \mathcal{E}$, and lower bounding its measure becomes a special case of measure concentration for small $k = o(\sqrt{n})$, which here we refer to as shallow expansions. Mahloujifar and Mahmoody [MM19] introduced a computational variant of the measure concentration phenomenon (and its related notion of isoperimetric inequalities) in which one is interested in mapping a sampled point $w_{\leq n} \leftarrow \mathbf{w}_{\leq n}$ to a k-close point in $\mathcal{E}$ *efficiently*.

The previous works of [MM19, EMM20] already showed that once $k > \Omega(\sqrt{n})$, then almost all the measure of $\mathbf{w}_{\leq n}$ is *computationally* concentrated around any event $\mathcal{E}$ of measure at least $1/\operatorname{poly}(n)$, matching the information theoretic results of [Tal95] up to a constant factor. However, the results of [MM19, EMM20] do not say anything about the "shallow depth" surfaces around $\mathcal{E}$, e.g., when $k = 1$.

A direct corollary of our Theorem 1 implies such computational concentration for small k, including computational (lower bounds on the) isoperimetry of $\mathcal{E}$ as special case. This can be observed by designing a protocol in which the i^{th} party samples $w_i \leftarrow \mathbf{w}$, broadcasts it in round i, and $b = 1$ if $w_{\leq n} \in \mathcal{E}$. Then, a k-replacing adversary that increases the probability of $\mathcal{E}$ by *efficiently* changing the messages of k parties simply gives a method of mapping $\mu + \Omega(\mu k/\sqrt{n})$ measure of $\mathbf{w}_{\leq n}$ into $\mathcal{E}$ by changing them in at most k coordinates, proving a computational isoperimetric inequality for such metric probability spaces (i.e., case of $k = 1$) and more generally computational concentration for shallow ($k = o(\sqrt{n})$) expansions. One might wonder that when $k = O(1)$ one can obtain computational (polynomial time) concentration trivially by trying all possible ways to change k blocks and see if we will fall into the target set $\mathcal{E}$. We emphasize that this only works in the *offline* setting (in which the adversary gets to see the full sequence before changing k bits of it) and only when the support set of $\mathbf{w}_i$ is of polynomial size. Therefore, making such algorithms online *or* polynomial time are both nontrivial when the support set of $\mathbf{w}_i$ is large.

More formally, we obtain the following theorem as a corollary to Theorem 15.

Theorem 44 (Computational isoperimetry in product spaces)**.** *Let $\mathbf{w}_{\leq n} \equiv \mathbf{w}_1 \times \cdots \times \mathbf{w}_n$ be any product probability space of dimension n, and let $\mathrm{HD}(\cdot,\cdot)$ be the Hamming distance of dimension n defined over $\mathrm{Supp}(\mathbf{w}_{\leq n})$. We call an algorithm A k-replacing, if given any $u_{\leq n} \leftarrow \mathbf{w}_{\leq n}$ maps it to some $v_{\leq n}$ such that $\mathrm{HD}(u_{\leq n}, v_{\leq n}) \leq k$ (the mapped instance could be $u_{\leq n}$ itself). Then, there is a PPT k-replacing algorithm A such that, given oracle access to membership queries to any set $\mathcal{S} \subseteq \mathrm{Supp}(\mathbf{w}_{\leq n})$ of measure at least*

$\mu \leq 1 - \Omega(1)$ *(according to* $\mathbf{w}_{\leq n}$*) it holds that,* A *runs in time* $\mathrm{poly}(N/\mu)$*, where* N *is the bit length of any* $w_{\leq n} \in \mathrm{Supp}(\mathbf{w}_{\leq n})$*, and it holds that:*

$$\Pr_{u_{\leq n} \leftarrow \mathbf{w}_{\leq n}} \left[\mathsf{A}^{\mathbf{w}_{\leq n}[\cdot], \mathcal{S}}(u_{\leq n}) \in \mathcal{S} \right] \geq \mu + \Omega(\mu \cdot k/\sqrt{n}).$$

Also, we note that if we remove the PPT condition, the existence of such A is *equivalent* to lower bounds on the measure of k-expansions of sets of probability μ in product spaces under Hamming distance. For $k = 1$, this notion exactly captures the isoperimetric inequality.

For the special case that $\mathbf{w}_{\leq n}$ is the uniform distribution over $\{0,1\}^n$, Theorem 39 gives a tight computational isoperimetric inequality for the Boolean hypercube.

B Improved online attacks through recursive composition

In this section, we study how much one can improve the biasing bounds proved in Section 3 using information-theoretic (i.e., computationally unbounded) attacks while the attack is still *online*. Note that if the attacks are off-line and computationally unbounded, then the problem becomes equivalent to proving concentration bounds (or isoperimetric inequalities) for product spaces under Hamming distance. In other words, by putting the computational aspects aside, we focus on finding lower bounds on the *online* isoperimetry in product spaces (or more generally, of random processes) under Hamming distance.

In summary, the our approach is as follows. We observe that any 1-replacing targeted attack can be used in a recursive way to improve the bias using more corruptions. Then we will apply this idea to our 1-corruption attack of Construction 10 by optimizing the choice of λ based on n, μ, which leads to a recursive attack with a better bias than that of Theorem 15, because we can indeed choose λ_i's differently for different i. On the down side, because this attack is recursive, its running time will *not* be polynomial if $k = \omega(1)$.

Theorem 45 (Recursively composing attacks)**.** *Suppose* A *is an* 1*-replacing targeted attack such that for every protocol with expected output at least* $\geq \mu$ *before the attack, its expected output after the attack will be at least* $\mu' = g(\mu) \geq \mu$*. Then, there is a* k*-replacing attack* IndA_k *that increases the bias of any protocol from at least* μ *to at least* $g^{(k)}(\mu)$ *where* $g^{(k)}(\cdot)$ *means applying* g k *times. This result is general, as it applies to both online and offline attacks.*

Proof. The key assumption in Theorem 45 is that it uses a *targeted* attack, and it uniformly applies to both offline and online attacks. However, we state the proof for the case of single-turn online attacks. We prove the theorem by induction on k. The basis of the induction by k is trivially true. Now suppose we have a k-tampering adversary IndA_{k-1} who is attacking the random process $\mathbf{w}_{\leq n}$ and generates the joint processes $(\mathbf{u}_{\leq n}, \mathbf{v}_{\leq n})$ as a result. A crucial property of such attacks, which we will use, is that $\mathrm{Supp}(\mathbf{v}_{\leq n}) \subseteq \mathrm{Supp}(\mathbf{w}_{\leq n})$. Now, define the random process $\mathbf{w}'_{\leq n} = (\mathbf{w}'_1, \ldots, \mathbf{w}'_n)$ as $w'_i = (u_i, v_i)$. Namely, $\mathbf{w}_{\leq n}$ contains pairs that are sampled according to the joint process $(\mathbf{u}_{\leq n}, \mathbf{v}_{\leq n})$. We write $(u_{\leq n}, v_{\leq n}) \leftarrow \mathbf{w}'_{\leq n}$ to denote the sampling of $((u_1, v_1), \ldots, (u_n, v_n))$ from $\mathbf{w}_{\leq n}$.

By the induction hypothesis we have $\Pr_{(u_{\leq n}, v_{\leq n}) \leftarrow \mathbf{w}'_{\leq n}}[\mathrm{HD}(u_{\leq n}, v_{\leq n}) \leq k-1] = 1$. In other words, for all $(u_{\leq n}, v_{\leq n}) \in \mathrm{Supp}(\mathbf{w}'_{\leq n})$ it holds that $\mathrm{HD}(u_{\leq n}, v_{\leq n}) \leq k-1$. We also define $f(w'_1, \ldots, w'_n) = f(v_1, \ldots, v_n)$, which simply means the function is effectively applied to the (finalized values) $v_1, \ldots, v_n$.

Now, we apply the 1-replacing attack A to the random process $\mathbf{w}'_{\leq n}$ according to function f. Note that this is *different* from applying the attack to $(v_1, \ldots, v_n)$, as this process has even different alphabet size. Let $(\mathbf{u}'_{\leq n}, \mathbf{v}'_{\leq n})$ be the resulting joint processes, where $\mathbf{u}'_{\leq n}$ captures the initial samples from $\mathbf{w}'_{\leq n}$, and $\mathbf{v}'_{\leq n}$ captures the finalized values. (Note that each u'_i, as well as each v'_i, is a *pair*.)

Now, we see how the combination of IndA_{k-1} and A, which we call IndA_k, can be interpreted as an attack on the original random process $\mathbf{w}_{\leq n}$. Let $u'_i = (a_i, b_i)$ and $v'_i = (c_i, d_i)$. Then in each step (a_i, b_i) is sampled first conditioned on $v'_1, \ldots, v'_{i-1}$. This means a_i is first sampled from $(\mathbf{w}_i \mid \mathbf{w}_{\leq i-1} = d_{\leq i-1})$, then the IndA_{k-1} generates b_i, and then A takes u'_i and generates $v'_i = (c_i, d_i)$. Therefore, we can interpret a_i as the original sample for the ith block (before the attack), and d_i as the resulting finalized value after the attacker IndA_k intervenes.

We need to prove two things about the attack A_{k+1}: its achieved bias and the budget.

Bias of IndA_k*:* By induction we have $\mathbb{E}[f(\mathbf{w}'_{\leq n})] \geq g^{(k-1)}(\mu)$. Furthermore, by the property of A, we have the expected output under the attack IndA_k to satisfy

$$\mathbb{E}[f(\mathbf{d}_{\leq n})] = \mathbb{E}[f(\mathbf{v}'_{\leq n})] \geq g(g^{(k-1)}(\mu)) = g^{(k)}(\mu).$$

Budget of A_{k+1}*:* We first note that because our replacing attacks (in this IndA_{k-1}) always stay in the support set of the original random process, therefore $\mathrm{Supp}(\mathbf{v}'_{\leq n}) \subseteq \mathrm{Supp}(\mathbf{w}'_{\leq n})$. Therefore, for all sampled $(c_{\leq n}, d_{\leq n}) \leftarrow \mathbf{v}'_{\leq n}$, it holds that $\mathrm{HD}(c_{\leq n}, d_{\leq n}) \leq k-1$. Furthermore, by applying the same property to A, we conclude that $\mathrm{HD}(u'_{\leq n}, v'_{\leq n}) \leq 1$. By these two, we conclude that $\mathrm{HD}(a_{\leq n}, d_{\leq n}) \leq k$. □

Remark 46 (Running time of recursive attacks)**.** If the original targeted 1-replacing attack A runs in polynomial time, then the recursive k-replacing attack IndA_k (of depth/budget k) also can be implemented in polynomial time as long as $k = O(1)$. The reason is that any recursive algorithm of constant depth, with a polynomial time base case, runs in polynomial time. However, for $k = \omega(1)$, the running time of such attacks become super-polynomial time, and can only be used as an information-theoretic attack.

The following lemma follows as a direct special case of Theorem 15.

Lemma 47 (Targeted 1-tampering attacks)**.** *Let $k = 1$ in Construction 10. Then we have*

$$\mu_1 - \mu \geq (1 - \mu - e^{\frac{-2\mu^2}{n\lambda_1^2}}) \cdot \lambda_1.$$

We can now use the 1-replacing attack of Lemma 47 by optimizing λ_1 based on μ, n. We first show how to find a closed-form formula for an optimized attack that improves the bound of Theorem 15.

Theorem 48 (Optimized recursive attack)**.** *Let μ_i be the average of process when in the i^{th} stage of the inductive attack. Also let $c = \min_{i\in[k]} 1 - e^{\frac{-2\mu_{i-1}^2}{n\lambda_i^2}}$. We then have*

$$\mu_k \geq \mu + (1 - \prod_{i=1}^{k}(1 - \lambda_i))(c - \mu).$$

In particular, (by setting $\lambda_i = \mu_{i-1}/\sqrt{n}$) we obtain

$$\mu_k \geq \mu + \left(1 - \left(1 - \frac{1 - e^{-2}}{\sqrt{n}} + \frac{(1 - e^{-2} - \mu)\left(1 - (1 - \mu/\sqrt{n})^k\right)}{k\mu}\right)^k\right) \cdot (1 - e^{-2} - \mu).$$

Remark 49. Note that one can show that the first bound of Theorem 48 is always better than the bound of Theorem 15. The reason is that μ_i is increasing and for all i we have $1 - e^{-\frac{\mu_i^2}{\lambda_i^2 \cdot n}} \geq 1 - e^{-\frac{\mu^2}{\lambda_i^2 \cdot n}}$.

Proof of Theorem 48. Let μ_k to be the bias of the recursive application of the attack according to Theorem 45 with k recursions, when using λ_i in the ith stage of the recursion. We inductively prove that

$$c - \mu_k \leq (c - \mu) \prod_{i=1}^{k} (1 - \lambda_i).$$

Define $g(\mu, \lambda)$ to be the minimum expectation of the 1-replacing attack with parameter λ when applied to a random process with average μ. By Lemma 47 we have

$$g(\mu, \lambda) \geq \mu + (c - \mu)\lambda$$

Therefore, by Theorem 45 we know that the minimum bias of the recursive algorithm with k replacements is at least $u_k = g^k(\mu, \lambda_{\leq k})$. Now we inductively show that $(c - \mu_k) \leq (c - \mu) \prod_{i=1}^{k} (1 - \lambda_k)$. By definition of g, for any i we have $c - \mu_i \leq c - g(\mu_{i-1}, \lambda_i) \leq (c - \mu_{i-1})(1 - \lambda_i)$. Therefore, by setting $i = 1$, we get the base of induction. For $i > 1$ we using induction hypothesis we get $c - g(\mu_i) \leq (c - \mu_{i-1})(1 - \lambda_i) \leq (c - \mu) \prod_{j=1}^{i} (1 - \lambda_i)$ which proves the hypothesis. Therefore, we have

$$\mu_k \geq \mu + (1 - \prod_{i=1}^{k} (1 - \lambda_i))(c - \mu)$$

which finishes the first part of theorem. Now we will analyze this bound to get the closed form (part two of theorem). We first set each $\lambda_i = \mu_{i-1}/\sqrt{n}$ which implies

$$\mu_k \geq \mu + (1 - \prod_{i=1}^{k} (1 - \lambda_i))(1 - e^{-2} - \mu).$$

A relax approximation of this bound could be to replace all λ_is with λ_1 here, but doing this approximation at this stage would be sub-optimal. define $t_i = 1 - e^{-2} - u_i$. We first use the AM-GM inequality to get

$$t_k/t_0 \leq \prod_{i=1}^{k} (1 - \lambda_i) \leq (1 - (\sum_{i=1}^{k} \lambda_i)/k)^k. \tag{29}$$

We need to bound $\sum_{i=1}^{k} \lambda_i$ from bellow. We have

$$\sum_{i=1}^{k} \lambda_i = \sum_{i=0}^{k-1} (1 - e^{-2} - t_i)/\sqrt{n} = k(1 - e^{-2})/\sqrt{n} - \sum_{i=0}^{k-1} t_i/\sqrt{n}. \tag{30}$$

Only now, we use the trick of replacing λ_i with λ_1. Note that t_i is always smaller than the geometric series $Y_i = Y_{i-1} \cdot (1 - \lambda_1)$ with $Y_0 = t_0$, as λ_is are increasing. Therefore we can upper bound $\sum_{i=1}^{k} t_i$ with sum of series $\sum_{i=1}^{k} Y_i$ which is

$$\sum_{i=1}^{k} Y_i = Y_0(1 - (1 - \lambda_1)^k)/\lambda_1 = t_0(1 - (1 - \lambda_1)^k)/\lambda_1 \leq t_0/\lambda_1.$$

By replacing this in (30), we get

$$\sum_{i=1}^{k} \lambda_i = \sum_{i=1}^{k} (1 - e^{-2} - t_i)/\sqrt{n} \geq k((1 - e^{-2})/\sqrt{n} - (1 - (1 - \lambda_1)^k) \cdot t_0/\mu, \tag{31}$$

which together with (29) they imply

$$t_k/t_0 \leq \left(1 - \frac{1 - e^{-2}}{\sqrt{n}} + \frac{t_0\big(1 - (1 - \lambda_1)^k\big)}{k(1 - e^{-2} - t_0)}\right)^k.$$

Therefore, we have

$$\mu_k \geq \mu + \left(1 - \left(1 - \frac{1 - e^{-2}}{\sqrt{n}} + \frac{(1 - e^{-2} - \mu)\big(1 - (1 - \lambda_1)^k\big)}{k\mu}\right)^k\right) \cdot (1 - e^{-2} - \mu).$$

□

Optimization without a closed-form formula. This bound of Theorem 48 still suffers from the fact that by increasing k, the final expected value still reaches $1 - e^2$ rather than 1. However, if λ is indeed optimized in each iteration of the recursive attack, then the expected output approaches 1. For that, λ_i, μ_i are inductively defined as follows. We let $\mu_0 = \mu$, and then for $i \geq 1$,

$$\lambda_i = \operatorname*{argmax}_{\lambda} \ (1 - \mu_{i-1} - e^{\frac{-2\mu_{i-1}^2}{n\lambda^2}}) \cdot \lambda, \text{ and } \mu_i = \mu_{i-1} + (1 - \mu_{i-1} - e^{\frac{-2\mu_{i-1}^2}{n\lambda_i^2}}) \cdot \lambda_i.$$

C Targeted biasing attacks from martingale gap finders

In Section B, we showed that finding 1-replacing attacks are essentially all we need for obtaining *information theoretic* k-replacing attacks. The work of Cleve and Impagliazzo [CI93] showed how to obtain 1-replacing *non-targeted* attacks using a general argument about Doob martingales. In particular, they showed that for any n-step martingale in which the final value is in $\{0, 1\}$, with $\Omega(1)$ probability there is a step in the martingale with jump at least $\Omega(1/\sqrt{n})$. We refer to the latter as *finding gaps* in Doob martingales.

In this section, we show a reduction from *targeted* attacks on coin flipping protocols to the task of finding gaps in their corresponding Doob martingale. This shows that the result of Cleve and Impagliazzo can also be used to obtain targeted 1-replacing attacks with bias $\Omega(1/\sqrt{n})$. Note that, their result is only proved for almost unbiased final bits in which both $\{0, 1\}$ happen with probability $\Omega(1)$. Therefore, we cannot use their result to recover the results of Section B, however if one wants to get an attack of budget 1, as we show in this section their gap-finding argument would be useful and relevant.

We first define the notion of a "stopping" adversary who simply stops a random process.

Definition 50 (Online stopping)**.** Let $\mathbf{w}_{\leq n} \equiv (\mathbf{w}_1, \ldots, \mathbf{w}_n)$ be a sequence of jointly distributed random variables, and let $\mathbf{w}[w_{\leq i-1}]$ be the online sampler for $\mathbf{w}_{\leq n}$ for all $i \in [n]$ and all $w_{\leq i-1} \in \mathrm{Supp}(\mathbf{w}_{\leq i-1})$. We call a (potentially randomized) algorithm Stop an *online stopping* algorithm for $\mathbf{w}_{\leq n}$, if for any randomness r for Stop the following holds:

- $\mathsf{Stop}_r(w_{\leq i}) \in \{0, 1\}$ for any $x_{\leq i}$, where outputting 1 is interpreted as "stop".

- If $\mathsf{Stop}_r(w_{\le i}) = 1$, then $\mathsf{Stop}_r(w_{\le j}) = 1$ for any $w_{\le j}$ where $j \ge i$. Namely, if Stop announces that stop has happened at moment i, then it will say so afterwards.
- If $\mathsf{Stop}_r(w_{\le n}) = 1$ for $w_{\le n} = (w_1, \ldots, w_n)$, then we let $\mathsf{StopTime}_r(w_{\le n}) = i$ if $i \in [n]$ is the smallest number such that $\mathsf{Stop}(w_{\le i}) = 1$. If $\mathsf{Stop}(w_{\le n}) = 0$, we let $\mathsf{StopTime}_r(w_{\le n}) = n + 1$. By $\tau \leftarrow \mathsf{StopTime}(\mathbf{w}_{\le n})$, we denote the random process of sampling $w_{\le n} \leftarrow \mathbf{w}_{\le n}$, choosing r at random, and letting $\tau = \mathsf{StopTime}_r(w_{\le n})$.

Definition 51 (Gap finders for martingales)**.** Let $\mathbf{w}_{\le n} \equiv (\mathbf{w}_1, \ldots, \mathbf{w}_n)$ be a martingale, and extend it by defining $w_0 = \mathbb{E}[\mathbf{w}_1], w_{n+1} = w_n$. We call an online stopping algorithm Stop for $\mathbf{w}_{\le n}$ a (ρ, α) *gap finder* if with probability at least ρ over $w_{\le n} \leftarrow \mathbf{w}_{\le n}$ and the randomness r for Stop it holds that $|w_\tau - w_{\tau-1}| \ge \alpha$, where $\tau = \mathsf{StopTime}_r(w_{\le n})$. Namely, with probability at least ρ, the stop time chosen by Stop shows a jump of at least α. We simply call Stop a α gap finder if $\mathbb{E}_{w_{\le n} \leftarrow \mathbf{w}_{\le n}, r}[|w_\tau - w_{\tau-1}|] \ge \alpha$.

We now recall two results proved about gap finders.

Theorem 52 (Gap finders for martingales)**.** *Let $\mathbf{w}_{\le n}$ be an n-step Martingale such that $w_0 = \mathbb{E}[\mathbf{w}_1] = \mu$. Then, the following holds:*

- [CI93] *There exists an $(\Omega(\mu), \Omega(\mu/\sqrt{n}))$ gap finder for $\mathbf{w}_{\le n}$.*
- [KMM19] *There exists an α gap finder for $\mathbf{w}_{\le n}$, where $\alpha = 2\mu(1-\mu)/\sqrt{2n-1}$.*

The first part is proved in [CI93] for $\mu = 1/2$, but the same proof carries over for an arbitrary μ with the bound stated in Theorem 52. The two results of Theorem 52 are incomparable, but if one ignores the constants, then the first part implies the second part when $\mu = \Omega(1)$.

From gap finders to non-targeted attacks. In [CI93, KMM19] it was shown how to obtain *non-targeted* attacks from gap finders. In particular, (ρ, α) (resp. α) gap finders can be used to obtain non-targeted with bias $\Omega(\rho\alpha)$ [CI93] (resp. $\Omega(\alpha)$ [KMM19]). The idea is quite simple: a (ρ, α) gap finder either achieves *positive* gaps of α with probability at least $\rho/2$ or they achieve *negative* gaps of α with probability at least $\rho/2$. In the former case, the adversary can wait for a moment when $+\alpha$ jump is about to happen (with nonzero probability) in which case it corrupts the sender of the message and sends the message that achieves this jump. If the latter case happens, the adversary can wait for an jump of magnitude $-\alpha$ and then it will corrupt the party and resets its message. The same trick can be applied to α gap finders as well. This left open whether gap finders can be used to obtain *targeted* biasing attacks.

From gap finders to *targeted* attacks. In this section, we show how to obtain targeted attacks using (ρ, α) gap finders. We do not know how to find (expected) α gap finders for this purpose. This means we cannot use the result of [KMM19] (see Theorem 52). In particular, we show how to obtain targeted bias $\Omega(\rho\alpha)$ using any (ρ, α) gap finder Unfortunately, the bound of the (ρ, α) gap finder of [CI93] degrades with μ, which means we cannot use their result to obtain an asymptotically similar result to that of our Lemma 47. This motivates the following question, which as far as we know is open.

Open question about asymptotically optimal (ρ, α) gap finders. Let $\mathbf{w}_{\le n}$ be an n-step Martingale such that $\mathbb{E}[\mathbf{w}_1] = \mu \le 0.5$. Then, can one always obtain an (ρ, α) gap finder for $\mathbf{w}_{\le n}$ such that $\rho\alpha = \Omega(\mu/\sqrt{n})$?

A positive answer to the following question above would have allowed us to use the derived targeted attack instead of our Lemma 47 and obtain (information-theoretic) k-replacing attacks through recursive compositing, as stated in Section B.

We now describe our construction of 1-replacing targeted adversaries based on (ρ, α) gap finders.

Theorem 53 (Targeted 1-replacing attacks from (ρ, α) gap finders)**.** *Suppose* Stop *is an online stopping algorithm for the random process* $\mathbf{w}_{\leq n}$*. Suppose* f *is a Boolean function defined over* $\mathrm{Supp}(\mathbf{w}_{\leq n})$ *and that* Stop *is a* (ρ, α) *gap finder for Doob martingale* $\left(\bar{f}(\mathbf{w}_{\leq i})\right)_{i\in[n]}$*. Then, given oracle access to* Stop *and* $\bar{f}(\cdot)$*, there is an* efficient 1*-replacing algorithm* $\mathsf{A}^{\mathsf{Stop}(\cdot),\bar{f}(\cdot)}$ *that transforms* $\mathbf{w}_{\leq n}$ *into* $\mathbf{v}_{\leq n}$ *such that* $\mathbb{E}[f(\mathbf{v}_{\leq n})] \geq \mathbb{E}[f(\mathbf{w}_{\leq n})] + \Omega(\alpha\rho)$.

To prove Theorem 53, we construct two 1-replacing algorithms. We show that for every random process, one of the 1-replacing algorithms will bias the output by $\Omega(\alpha\rho)$.

Construction 54 (Targeted 1-replacing attacks using negative gap finders)**.** Given a prefix $v_{\leq i-1}$ that is the output of previous steps of the algorithm and a candidate sample w_i and a Boolean variable Abort that indicates whether the algorithm has aborted or not, let $\mathsf{Stop}(v_{\leq i-1}, w_i)$ denote the output of Stop algorithm. Initialize $\mathsf{Abort} = 0$ for $i = 1$. StopA_1^- at time i decides whether to change or not change w_i as follows.

- (Case 0) If $\mathsf{Abort} = 1$, leaves with $v_i = w_i$.
- (Case 1) If Case 0 does not happen, suppose $\mathsf{Stop}(v_{\leq i-1}, w_i)$ and $\bar{f}(v_{\leq i-1}) \geq \bar{f}(v_{\leq i-1}, w_i)$, then StopA_1^- will replace and return a randomly sample $v_i \leftarrow (\mathbf{w}_i \mid \mathbf{w}_{\leq i-1} = v_{\leq i-1})$ and set $\mathsf{Abort} = 1$.
- (Case 2) If both Case 0 and Case 1 do not happen, leave with $v_i = w_i$.

Construction 55 (Targeted 1-replacing attacks using positive gap finders)**.** This is identical to Construction 54, with the only difference being in Case 1:

- (Case 1) if Case 0 does not happen, the algorithm randomly samples a $v_i \leftarrow (\mathbf{w}_i \mid \mathbf{w}_{\leq i-1} = v_{\leq i-1})$. If $\mathsf{Stop}(v_{\leq i-1}, v_i)$ and $\bar{f}(v_{\leq i-1}) < \bar{f}(v_{\leq i-1}, v_i)$, then StopA_1^+ will replace w_i with v_i. Set $\mathsf{Abort} = 1$.

We prove either Construction 54 or the Construction 55 can increase the output bit by at least $\Omega(\alpha\rho)$.

Claim 56. *Suppose* Stop *is an* (ρ, α) *gap finder for random process* $\mathbf{w}_{\leq n}$*, where* $\mu = \mathbb{E}[f(w_{\leq n})]$*. Given oracle access to* Stop *and* $\bar{f}(\cdot)$*, either Construction 54 or Construction 55 will transform* $\mathbf{w}_{\leq n}$ *to* $\mathbf{v}_{\leq n}$ *such that* $\mathbb{E}[f(v_{\leq n})] \geq \frac{\alpha\rho}{4} + \mu$ *with only 1 replacement.*

Proof. We start by showing every replacement increases the expectation of $\bar{f}(v_{\leq i})$

- For any Case-1 replacement in Construction 54, we have $\mathbb{E}[\bar{f}(v_{\leq i})] = \bar{f}(v_{\leq i-1}) \geq \bar{f}(v_{\leq i-1}, w_i)$. Also, $\mathbb{E}[\bar{f}(v_{\leq i})]$ increases by $\bar{f}(v_{\leq i-1}) - \bar{f}(v_{\leq i-1}, w_i) \geq \alpha$.
- For any replacement made by Case 1 in Construction 55, we have $\bar{f}(v_{\leq i}) > \bar{f}(v_{\leq i-1})$. For the prefix $v_{\leq i-1}$, $\mathbb{E}[\bar{f}(v_{\leq i})]$ increase by $\bar{f}(v_{\leq i}) - \bar{f}(v_{\leq i-1}) \geq \alpha$.

Let $S(u_{\leq i-1}, u_i)$ be a Boolean function that is True when the gap finder Stop returns the gap on the prefix $u_{\leq i-1}$ and sample u_i. By definition, we have the following $\mathbb{E}_{u_{\leq n}\leftarrow\mathbf{w}_{\leq n}}\left[\sum_{i=1}^{n} S(u_{\leq i-1}, u_i)\right] \geq \rho$. A gap can either be positive or negative. Let

$$\rho_1 = \mathop{\mathbb{E}}_{u_{\leq n}\leftarrow\mathbf{w}_{\leq n}}\left[\sum_{i=1}^{n} S(u_{\leq i-1}, u_i) \wedge \left(\bar{f}(u_{\leq i-1}) \geq \bar{f}(u_{\leq i-1}, u_i)\right)\right]$$

be the probability of the negative gap and

$$\rho_2 = \mathop{\mathbb{E}}_{u_{\le n} \leftarrow \mathbf{w}_{\le n}} \left[\sum_{i=1}^{n} S(u_{\le i-1}, u_i) \wedge \left(\bar{f}(u_{\le i-1}) < \bar{f}(u_{\le i-1}, u_i) \right) \right]$$

be the probability of the positive gap. Then by definition we have $\rho_1 + \rho_2 \ge \rho$. Therefore, either $\rho_1 \ge \frac{\rho}{2}$ or $\rho_2 \ge \frac{\rho}{2}$. We discuss these two cases separately.

Assuming $\rho_1 \ge \frac{\rho}{2}$: In this case we prove StopA_1^- achieves the gain of at least $\frac{\alpha\rho}{2}$. Let Boolean variable $C(v_{\le i}, u)$ denote the attack StopA_1^- is performed on prefix $v_{\le i}$ with sample u (Case 1 in Construction 54). We have

$$C(v_{\le i}, u) \implies S(v_{\le i}, u) \wedge \left(\bar{f}(v_{\le i}) \ge \bar{f}(v_{\le i}, u) \right). \tag{32}$$

by the definition of Construction 54. Also, if $S(v_{\le i}, u)$ is True, by the definition of Stop we have $\forall j < i, S(v_{\le j}, v_{j+1}) =$ False, which indicates Construction 54 won't abort at the prefix $v_{\le j}$. Therefore,

$$S(v_{\le i}, u) \wedge \left(\bar{f}(v_{\le i}) \ge \bar{f}(v_{\le i}, u) \right) \implies C(v_{\le i}, u). \tag{33}$$

Combining 32 and 33 we have

$$C(v_{\le i}, u) = S(v_{\le i}, u) \wedge \left(\bar{f}(v_{\le i}) \ge \bar{f}(v_{\le i}, u) \right). \tag{34}$$

For non-replacing case $\bar{C}(v_{\le i}, u)$, we have

$$\forall v_{\le i}, \mathop{\mathbb{E}}_{(u,v) \leftarrow (\mathbf{w},\mathbf{v})[v_{\le i}]} \left[(\bar{f}(v_{\le i}, v) - \bar{f}(v_{\le i})) \cdot \bar{C}(v_{\le i}, u) \right] = 0. \tag{35}$$

For replacement case $C(v_{\le i}, u)$, we have

$$\forall v_{\le i}, \mathop{\mathbb{E}}_{(u,v) \leftarrow (\mathbf{w},\mathbf{v})[v_{\le i}]} \left[(\bar{f}(v_{\le i}, v) - \bar{f}(v_{\le i})) \cdot C(v_{\le i}, u) \right] \ge \mathop{\mathbb{E}}_{(u,v) \leftarrow (\mathbf{w},\mathbf{v})[v_{\le i}]} \left[\alpha C(v_{\le i}, u) \right]. \tag{36}$$

Using StopA_1^- as the attacker, we also have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(u_{\le n}, v_{\le n}) \leftarrow (\mathbf{u}_{\le n}, \mathbf{v}_{\le n})} [f(v_{\le n})] - \mu \\
&= \mathop{\mathbb{E}}_{(u_{\le n}, v_{\le n}) \leftarrow (\mathbf{u}_{\le n}, \mathbf{v}_{\le n})} \left[\sum_{i=1}^{n-1} (\bar{f}(v_{\le i+1}) - \bar{f}(v_{\le i})) \right] \\
&= \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{v_{\le i} \leftarrow \mathbf{v}_{\le i}} \left[\mathop{\mathbb{E}}_{(u,v) \leftarrow (\mathbf{w},\mathbf{v})[v_{\le i}]} \left[(\bar{f}(v_{\le i}, v) - \bar{f}(v_{\le i})) \cdot \left(C(v_{\le i}, u) + \bar{C}(v_{\le i}, u) \right) \right] \right] \\
&= \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{v_{\le i} \leftarrow \mathbf{v}_{\le i}} \left[\mathop{\mathbb{E}}_{(u,v) \leftarrow (\mathbf{w},\mathbf{v})[v_{\le i}]} \left[(\bar{f}(v_{\le i}, v) - \bar{f}(v_{\le i})) \cdot C(v_{\le i}, u) \right] + \right. \\
&\qquad \left. \mathop{\mathbb{E}}_{(u,v) \leftarrow (\mathbf{w},\mathbf{v})[v_{\le i}]} \left[(\bar{f}(v_{\le i}, v) - \bar{f}(v_{\le i})) \cdot \bar{C}(v_{\le i}, u) \right] \right] \\
\text{(By (35, 36))} &\ge \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{v_{\le i} \leftarrow \mathbf{v}_{\le i}} \left[\mathop{\mathbb{E}}_{(u,v) \leftarrow (\mathbf{w},\mathbf{v})[v_{\le i}]} \left[\alpha \cdot C(v_{\le i}, u) \right] \right] \\
\text{(By (34))} &= \alpha \mathop{\mathbb{E}}_{u_{\le n} \leftarrow \mathbf{w}_{\le n}} \left[\sum_{i=1}^{n} S(u_{\le i-1}, u_i) \wedge \left(\bar{f}(v_{\le i-1}) \ge \bar{f}(v_{\le i-1}, v_i) \right) \right] \\
&= \alpha\rho_1 \ge \frac{\alpha\rho}{2}.
\end{aligned}$$

Therefore, StopA_1^- have $\mathbb{E}[f(v_{\le n})] \ge \frac{\alpha\rho}{2} + \mu$.

Assuming $\rho_2 \ge \frac{\rho}{2}$: In this case we prove StopA_1^+ increase the expectation $\mathbb{E}[f(v_{\le n})]$ by at least $\frac{\alpha\rho}{4}$. Let Boolean variable $C(v_{\le i}, u, u')$ denote whether the attack StopA_1^+ is performed on prefix $v_{\le i}$, candidate sample u, and an additional sample u' generated by the adversary (Case 1 in Construction 55).

Similar to the previous case, we have the following equations.

$$\forall v_{\le i}, \mathop{\mathbb{E}}_{(u,u',v)\leftarrow(\mathbf{w},\mathbf{w},\mathbf{v})[v_{\le i}]} \left[(\bar{f}(v_{\le i}, v) - \bar{f}(v_{\le i})) \cdot \bar{C}(v_{\le i}, u, u')\right] = 0. \tag{37}$$

and

$$\begin{aligned} \forall v_{\le i}, \mathop{\mathbb{E}}_{(u,u',v)\leftarrow(\mathbf{w},\mathbf{w},\mathbf{v})[v_{\le i}]} &\left[(\bar{f}(v_{\le i}, v) - \bar{f}(v_{\le i})) \cdot C(v_{\le i}, u, u')\right] \\ &\ge \mathop{\mathbb{E}}_{(u,u',v)\leftarrow(\mathbf{w},\mathbf{w},\mathbf{v})[v_{\le i}]} \left[\alpha C(v_{\le i}, u, u')\right]. \end{aligned} \tag{38}$$

Also, from the definition of StopA_1^+ we have

$$C(v_{\le i}, u, u') \implies (v_{i+1} = u') \wedge S(v_{\le i}, u') \wedge \bar{f}(v_{\le i}, u') - \bar{f}(v_{\le i}) \ge \alpha. \tag{39}$$

We then construct a Hybrid stopping algorithm based on Stop. We prove that Hybrid also find large positive gaps by Claim 58, and then we connect Hybrid with the StopA_1^+ by Claim 59.

Construction 57 (Stopping algorithm Hybrid)**.** Given a prefix $v_{\le i}$ that is the output of previous steps of the algorithm and a candidate sample w_i, the Hybrid algorithm will generate two samples, u and u' from $\mathbf{w}_{\le n}$.

- (Case 0) If $\mathsf{Stop}(v_{\le i}, u') \wedge \left(\bar{f}(v_{\le i}, u') - \bar{f}(v_{\le i}) \ge \alpha\right)$, then it returns u'.
- (Case 1) Otherwise, if $\mathsf{Stop}(v_{\le i}, u) \wedge \left(\bar{f}(v_{\le i}, u) - \bar{f}(v_{\le i}) \ge \alpha\right)$, then the algorithm returns u.
- (Case 2) Otherwise, return u or u' at random and continue the random process.

Hybrid is a stopping algorithm as it only stops whenever Stop also stops.

Claim 58. *Hybrid is a gap finder that finds positive gap with probability at least ρ_2.*

Let a Boolean variable $H(v_{\le i}, u)$ denotes whether the Hybrid algorithm stops when the prefix is $v_{\le i}$ and the sample is u. Using this definition we can prove Claim 58.

Proof. Intuitively the probability that Hybrid stops for $v_{\le i}$ is larger than that of Stop:

$$H(v_{\le i}, u, u') = S(v_{\le i}, u) \vee S(v_{\le i}, u'). \tag{40}$$

To formally prove the claim, we use induction. The claim trivially holds when $n = 1$. Now suppose the

claim holds for $n = k-1$. For $n = k$, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{u_{\le n}, u'_{\le n} \leftarrow \mathsf{Hybrid}} \left[\sum_{i=0}^{k-1} H(u_i, u, u') \right] \\
&= \mathop{\mathbb{E}}_{u_{\le n}, u'_{\le n} \leftarrow \mathsf{Hybrid}} \left[H(u_0, u, u') + \bar{H}(u_0, u, u') \cdot \mathop{\mathbb{E}}_{u_{\le n} \leftarrow \mathsf{Hybrid}(u_1)} \left[\sum_{i=1}^{k-1} H(u_1, u, u') \right] \right] \\
(\text{by induction}) &\ge \mathop{\mathbb{E}}_{u_{\le n}, u'_{\le n} \leftarrow \mathsf{Hybrid}} \left[H(u_0, u, u') + \bar{H}(u_0, u, u') \cdot \mathop{\mathbb{E}}_{u_{\le n} \leftarrow \mathbf{w}_{\le n}(u_1)} \left[\sum_{i=1}^{k-1} S(u_1, u) \right] \right] \\
(\text{by (40)}) &= \mathop{\mathbb{E}}_{u_{\le n}, u'_{\le n} \leftarrow \mathsf{Hybrid}} \left[S(u_0, u) + \bar{S}(u_0, u) S(u_0, u') + \bar{S}(u_0, u) \bar{S}(u_0, u') \cdot \mathop{\mathbb{E}}_{u_{\le n} \leftarrow \mathbf{w}_{\le n}(u_1)} \left[\sum_{i=1}^{k-1} S(u_1, u) \right] \right] \\
&\ge \mathop{\mathbb{E}}_{u_{\le n}, u'_{\le n} \leftarrow \mathsf{Hybrid}} \left[S(u_0, u) + \bar{S}(u_0, u) S(u_0, u') \cdot \mathop{\mathbb{E}}_{u_{\le n} \leftarrow \mathbf{w}_{\le n}(u_1)} \left[\sum_{i=1}^{k-1} S(u_1, u) \right] + \right. \\
&\qquad \left. \bar{S}(u_0, u) \bar{S}(u_0, u') \cdot \mathop{\mathbb{E}}_{u_{\le n} \leftarrow \mathbf{w}_{\le n}(u_1)} \left[\sum_{i=1}^{k-1} S(u_1, u) \right] \right] \\
&= \mathop{\mathbb{E}}_{u_{\le n}, u'_{\le n} \leftarrow \mathsf{Hybrid}} \left[S(u_0, u) + \bar{S}(u_0, u) \cdot \mathop{\mathbb{E}}_{u_{\le n} \leftarrow \mathbf{w}_{\le n}(u_1)} \left[\sum_{i=1}^{k-1} S(u_1, u) \right] \right] \\
&= \mathop{\mathbb{E}}_{u_{\le n} \leftarrow \mathbf{w}_{\le n}} \left[\sum_{i=0}^{k-1} S(u_i, u) \right].
\end{aligned}$$

□

Claim 59. *Construction 55 increase the expectation* $\mathbb{E}[f(v_{\le n})]$ *by at least* $\frac{\rho_2}{2}$.

Proof. We first show that the sequence $v_{\le n}$ generated by StopA_1^+ follows the random process Hybrid.

Consider the sampling process in (Case 1) of StopA_1^+. Two samples u and u' are generated following $\mathbf{w}_{\le n}$, although one of them, u' is generated by the adversary. If u' triggers Stop with a positive gap, the algorithm return u', otherwise, it return u. Then, for Hybrid, the algorithm also draw two samples u and u' and the two samples are equal to each other. Then the algorithm examine which sample triggers Stop with a large positive gap. Clearly by symmetry, the examination order doesn't matter. So we can assume the model check u' first. Also, if neither of u and u' can trigger Stop, we return either u and u'. By the symmetry, the distribution of the return sample when neither samples trigger Stop is similar with just return one of them. Therefore, the distribution Hybrid is identical to the distribution of StopA_1^+. We then prove

$$\forall v_{\le i}, \Pr[C(v_{\le i}, u, u')] \ge \frac{1}{2} \Pr[H(v_{\le i}, u, u')]. \tag{41}$$

By the definition of Hybrid, we have

$$C(v_{\le i}, u, u') \vee C(v_{\le i}, u', u) \implies H(v_{\le i}, u, u'). \tag{42}$$

Therefore,

$$\Pr[C(v_{\le i}, u, u')] + \Pr[C(v_{\le i}, u', u)] \ge \Pr[H(v_{\le i}, u, u')], \tag{43}$$

which implies (41). □

With Claim 58 and Claim 59, we can finish the proof of Claim 56 as follows:

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{(u_{\le n},v_{\le n})\leftarrow(\mathbf{u}_{\le n},\mathbf{v}_{\le n})}[f(v_{\le n})]-\mu\\
&=\mathop{\mathbb{E}}_{(u_{\le n},v_{\le n})\leftarrow(\mathbf{u}_{\le n},\mathbf{v}_{\le n})}[\sum_{i=1}^{n-1}(\bar{f}(v_{\le i+1})-\bar{f}(v_{\le i}))]\\
&=\sum_{i=0}^{n-1}\mathop{\mathbb{E}}_{v_{\le i}\leftarrow\mathbf{v}_{\le i}}\left[\mathop{\mathbb{E}}_{(u,u',v)\leftarrow(\mathbf{w},\mathbf{w},\mathbf{v})[v_{\le i}]}\left[(\bar{f}(v_{\le i},v)-\bar{f}(v_{\le i}))\cdot\left(C(v_{\le i},u,u')+\bar{C}(v_{\le i},u,u')\right)\right]\right]\\
\text{(By (37, 38))}&\ge\sum_{i=0}^{n-1}\mathop{\mathbb{E}}_{v_{\le i}\leftarrow\mathbf{v}_{\le i}}\left[\mathop{\mathbb{E}}_{(u,u',v)\leftarrow(\mathbf{w},\mathbf{w},\mathbf{v})[v_{\le i}]}\left[\alpha\cdot C(v_{\le i},u,u')\right]\right]\\
\text{(By (41))}&\ge\frac{\alpha}{2}\cdot\sum_{i=0}^{n-1}\mathop{\mathbb{E}}_{v_{\le i}\leftarrow\mathbf{v}_{\le i}}\left[\mathop{\mathbb{E}}_{(u,u',v)\leftarrow(\mathbf{w},\mathbf{w},\mathbf{v})[v_{\le i}]}\left[H(v_{\le i},u,u')\right]\right]\\
&\ge\frac{\alpha\rho_2}{2}\ge\frac{\alpha\rho}{4}.
\end{aligned}
$$

Combining the two cases above, we conclude that either StopA_1^- or StopA_1^+ will achieve

$$\mathbb{E}[v_{\le n}]\ge\frac{\alpha\rho}{4}+\mu.$$

□

The following extension of Theorem 53 allows the reduction from targeted attacks (i.e., online 1-isoperimetry) to gap-finding attacks to be polynomial time, assuming the random process is online-samplable. The reason is that using online samplers and function f as oracles, one can approximate the martingale $\bar{f}(\cdot)$ up to arbitrarily small $\varepsilon=1/\operatorname{poly}$ additive error.

Theorem 60 (Robust targeted attacks from noisy gap finders)**.** *If in Theorem 53,* A *is given an* ε*-approximation oracle* $\tilde{f}(\cdot)$ *instead of the exact oracle* $\bar{f}(\cdot)$*, it can still run in time* $\operatorname{poly}(N/\tau)$ *where* N *is an upper bound on the bit-length of all* $u_{\le n}\in\operatorname{Supp}(\mathbf{w}_{\le n})$*, and it still holds that* $\mathbb{E}[f(\mathbf{v}_{\le n})]\ge\mathbb{E}[f(\mathbf{w}_{\le n})]+\Omega\left((\alpha-\tau)\rho\right)$.

The proof of Theorem 60 follows directly from Construction 54 and Construction 55. By using the exact same construction with the approximated oracle $\tilde{f}(\cdot)$ to replace $\bar{f}(\cdot)$, we have the gap found by StopA_1 algorithms with the approximated oracle is equivalent to gap found by $\bar{f}(\cdot)$ when $\tau<\alpha$, only the size of gap will be at least $\alpha-\tau$ instead. In fact, even if we are not given oracle access to Stop, and need to implement it ourselves, we can still use the ε approximate oracle $\bar{f}(\cdot)$ to find out the existence of $\alpha-O(\varepsilon)$ gaps, and by choosing $\varepsilon=\alpha/10$, we still find a gap of size $\Omega(\alpha)$.