Circular Trace Reconstruction

Shyam Narayanan, Michael Ren

Abstract

Trace Reconstruction is the problem of learning an unknown string $x$ from independent traces of $x$, where traces are generated by independently deleting each bit of $x$ with some deletion probability $q$. In this paper, we initiate the study of Circular Trace Reconstruction, where the unknown string $x$ is circular and traces are now rotated by a random cyclic shift. Trace reconstruction is related to many computational biology problems studying DNA, which is a primary motivation for this problem as well, as many types of DNA are known to be circular.

Our main results are as follows. First, we prove that we can reconstruct arbitrary circular strings of length $n$ using $\exp(\tilde{O}(n^{1/3}))$ traces for any constant deletion probability $q$, as long as $n$ is prime or the product of two primes. For $n$ of this form, this nearly matches the best known bound of $\exp(O(n^{1/3}))$ for standard trace reconstruction. Next, we prove that we can reconstruct random circular strings with high probability using $n^{O(1)}$ traces for any constant deletion probability $q$. Finally, we prove a lower bound of $\tilde{\Omega}(n^3)$ traces for arbitrary circular strings, which is greater than the best known lower bound of $\tilde{\Omega}(n^{3/2})$ in standard trace reconstruction.

1 Introduction

The trace reconstruction problem asks one to recover an unknown string $x$ of length $n$ from independent noisy samples of the string. In the original setting, $x$ is a binary string in $\{0,1\}^n$, and a random subsequence $\tilde{x}$ of $x$, called a trace, is generated by sending $x$ through a deletion channel with deletion probability $q$, which removes each bit of $x$ independently with some fixed probability $q$. The main question is to determine how many independent traces are needed to recover the original string with high probability. This question has become very well studied over the past two decades [Lev01a, Lev01b, BKKM04, KM05, HMPW08, VS08, MPV14, DOS19, NP17, PZ17, HHP18, HL20, HPP18, Cha19, CDL20], with many results over various settings. For instance, people have studied the case where we wish to reconstruct $x$ for any arbitrarily chosen $x \in \{0,1\}^n$ (worst-case) or the case where we just wish to reconstruct a randomly chosen string $x$ (average-case). People have also studied the trace reconstruction problem for various values of the deletion probability $q$, such as if $q$ is a fixed constant between 0 and 1 or decays as some function of $n$. People have also studied variants where the traces allow for insertions of random bits, rather than just deletion of bits, and variants where the string is no longer binary but from a larger alphabet.

Finally, various generalizations or variants of the trace reconstruction problem have also been developed. These include error-correcting codes over the deletion channel (i.e., “coded” trace reconstruction) [CGMR19, BLS20], reconstructing matrices [KMMP19] and trees [DRR19] from traces, and reconstructing mixtures of strings from traces [BCF19, BCSS19, Nar20].

In this paper, we develop and study a new variant of trace reconstruction that we call Circular Trace Reconstruction. In this variant, there is again an unknown string $x \in \{0,1\}^n$ that we can sample traces from, but this time, the string $x$ is a cyclic string, meaning that there is no beginning.
Figure 1: An example of a circular trace. We start with an unknown circular string (top left). Each bit of the string is randomly deleted (red bits are deleted, black bits are retained) and the order of the retained bits is preserved, so we are left with the smaller circular string. However, since there is no beginning or end of the circular string, we assume the string is seen in clockwise order starting from a randomly chosen bit.

or end to the string. Equivalently, one can imagine a linear string that undergoes a random cyclic shift before a trace is returned. See Figure 1 for an example. Our goal, like in the normal trace reconstruction, is to reconstruct the original circular string using as few random traces as possible.

1.1 Main Results and Comparison to Linear Trace Reconstruction

Perhaps the first natural question about circular trace reconstruction is the following: how does the sample complexity of circular trace reconstruction compare to the sample complexity of standard (linear) trace reconstruction? Intuitively, one should expect circular trace reconstruction to be at least as difficult as standard trace reconstruction, since given any trace of a linear string, we can randomly rotate it to get a trace of the corresponding circular string. This reasoning, however, is slightly flawed. For instance, if we wish to distinguish between two strings \( x \) and \( y \) which are different as linear strings but equivalent up to a cyclic shift, then one cannot distinguish between traces of random rotations of \( x \) and traces of random rotations of \( y \). However, by padding the trace with extra bits before randomly rotating, one can show that circular trace reconstruction is at least as hard as linear trace reconstruction in both the worst-case and average-case. Indeed, we have the following proposition – as its proof is quite simple, we defer it to Appendix A.

**Proposition 1.1.** Suppose that we can solve worst-case circular trace reconstruction over length \( m \) strings with deletion probability \( q \) using \( T_1(m, q) \) traces. Then, we can solve worst-case linear trace reconstruction over length \( n \) strings with deletion probability \( q \) using \( \min_{m \geq 2n} T_1(m, q) \) traces.

Likewise, suppose that we can solve average-case circular trace reconstruction over length \( m \) strings with deletion probability \( q \) using \( T_2(m, q) \) traces. Then, we can solve average-case linear trace reconstruction over length \( n \) strings with deletion probability \( q \) using \( \min_{m \geq 2n} T_2(m, q) \) traces.

Given Proposition 1.1, any upper bounds for circular trace reconstruction imply nearly equivalent upper bounds for the linear trace reconstruction, and any lower bounds for linear trace reconstruction imply nearly equivalent lower bounds for circular trace reconstruction. This raises two natural questions. First, can we match or nearly match the best linear trace reconstruction upper bounds for circular trace reconstruction? Second, can we beat the best linear trace reconstruction lower bounds for circular trace reconstruction?

The first main result we prove is for worst-case circular strings. The best known upper bound for worst-case linear trace reconstruction with deletion probability \( q \), where \( q \) is a fixed constant
between 0 and 1, is \(\exp(O(n^{1/3}))\), where the unknown string has length \(n\). Our first main result, proven in Section 3, provides a nearly matching upper bound for the circular trace reconstruction problem, but only if the length \(n\) has at most 2 prime factors.

**Theorem 1.2.** Let \(x\) be an unknown, arbitrary circular string of length \(n\), let \(q\) be the deletion probability of each element in the string, and let \(p = 1 - q\) be the retention probability. Then, if \(n\) is either a prime or a product of two (possibly equal) primes, using \(\exp(O(n^{1/3}(\log n)^{2/3}p^{-2/3}))\) random traces, we can determine \(x\) with failure probability at most \(2^{-n}\).

The primary reason why our theorem fails for \(n\) having 3 or more prime factors is that we prove the following number theoretic result which is crucial in our algorithm.

**Theorem 1.3.** For any fixed integer \(n \geq 2\), the following statement is true if and only if \(n\) has at most 2 prime factors, counting multiplicity.

Define \(\omega := e^{2\pi i/n}\), and suppose that \(a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}\) are all integers in \(\{0, 1\}\). Also, suppose that for all \(0 \leq k \leq n - 1\), there is some integer \(c_k\) such that \(\sum a_i \omega^{i-k} = \omega^{c_k} \cdot \sum b_i \omega^{i-k}\). Then, the sequences \(\{a_i\}\) and \(\{b_i\}\) are cyclic shifts of each other.

The next main result we prove is for average-case circular strings: we show that a random circular string can be recovered using a polynomial number of traces. Formally, we prove the following theorem, done in section 4.

**Theorem 1.4.** Let \(x\) be an unknown but randomly chosen circular string of length \(n\) and let \(0 < q < 1\) be the deletion probability of each element. Then, there exists a constant \(C_q\) depending only on \(q\) such that we can determine \(x\) with failure probability at most \(n^{-10}\) using \(O(n^{C_q})\) traces.

The main lemma we need to prove Theorem 1.4 is actually a result that is true for worst-case strings. Specifically, we show how to recover the multiset of all consecutive substrings of length \(O(\log n)\) using a polynomial number of strings. While this does not guarantee that we can recover an arbitrary circular string, it does allow us to recover what we will call regular strings, which we show comprise the majority of circular strings. The following lemma may be of independent interest for studying worst-case strings as well, as it allows one to gain information about all “consecutive chunks” of the unknown string using only a polynomial number of queries.

**Lemma 1.5.** Let \(x = x_1 \cdots x_n\) be an arbitrary circular string of length \(n\) and let \(0 < q < 1\) be the deletion probability of each element. Then, for \(k = 100\log n\), we can recover the multiset of all substrings \(\{x_{i}x_{i+1} \cdots x_{i+k-1}\}_{i=1}^{n}\), where indices are modulo \(n\), using \(O(n^{C_q})\) traces with failure probability \(n^{-10}\), where \(C_q\) is a constant that only depends on \(q\).

The best known upper bound for average-case linear trace reconstruction is only \(\exp(O((\log n)^{1/3}))\). Unfortunately, we were not able to adapt their argument to circular strings. One major reason why we are unable to do so is that in the argument of [HPP18] (as well as [PZ17], which provides an \(\exp(O((\log n)^{1/2}))\) sample algorithm), the authors recover the \((k+1)\)st bit of the string assuming the first \(k\) bits are known using a small number of traces, and by reusing traces, they inductively recover the full string. However, since we are dealing with circular strings, even recovering the “first” bit does not make much sense. However, we note that even a polynomial-sample algorithm is quite nontrivial. In the linear case, a polynomial-sample algorithm for average-case strings was first proven by [HMPW08], and their algorithm only worked as long as the deletion probability \(q\) was at most some small constant, which when optimized is only about 0.07 [PZ17].
Our final main result regards lower bounds for worst-case strings. For linear worst-case strings, the best known lower bound for trace reconstruction is $\tilde{\Omega}(n^{3/2})$ \cite{Cha19}. For circular trace reconstruction, we show an improved lower bound of $\Omega(n^3)$, although the proof of our lower bound is actually much simpler and cleaner than those of the known lower bounds for standard trace reconstruction \cite{Cha19, HL20}. Specifically, we prove the following theorem, done in Section 5:

**Theorem 1.6.** Let $n \geq 1$, $2 \leq k \leq 4$, and let $x$ be the string $10^n10^{n+1}10^{n+k} = 10 \ldots 0 \ 1 \ 0 \ldots \ 0 \ 1 \ 0 \ldots$.

Likewise, let $y$ be the string $y = 10^n10^{n+k}10^{n+1}$. Then, the strings $x, y$ are not equivalent up to cyclic rotations, but for any constant deletion probability $q$, one requires $\Omega(n^3 / \log^3 n)$ random traces to distinguish between the original string being $x$ or $y$. Thus, for all integers $n$, worst-case circular trace reconstruction requires at least $\tilde{\Omega}(n^3)$ random traces.

1.1.1 Concurrent Work

We note that a very similar statement to Lemma 1.5, but for linear strings, was proven in independent concurrent work by Chen et. al. \cite[Theorem 2]{CDL20}, which provides a polynomial-sample algorithm for a “smoothed” variant of worst-case linear trace reconstruction. Many ideas in our proof of Lemma 1.5 and their proof appear to overlap, though our proof is substantially shorter.

1.2 Motivation and Relation to Other Work

From a theoretical perspective, circular trace reconstruction can bring many novel insights to the theory of reconstruction algorithms, some of which may be useful even in the standard trace reconstruction problem. For instance, the proof of Theorem 1.2 combines analytic, statistical, and combinatorial approaches as in previous trace reconstruction papers, but now also uses ideas from number theory and results about cyclotomic integers. To the best of our knowledge, this paper is the first paper on trace reconstruction that utilizes number theoretic ideas, though there is work on other problems about cyclic strings that uses ideas from number theory. Also, Lemma 1.5 shows a way to recover all contiguous sequences in the original string of length $O(\log n)$ for arbitrary circular strings, which is a new result even in the linear case (concurrent with \cite{CDL20}) and has applications to problems in linear trace reconstruction (as done in \cite{CDL20}).

From an applications perspective, trace reconstruction is closely related to the multiple sequence alignment problem in computational biology. In the multiple sequence alignment problem, one is given DNA sequences from several related organisms, and the goal is to align the sequences to determine what mutations each descendant underwent from their common ancestor: the trace reconstruction problem is analogous to actually recovering the common ancestor. See \cite{BKKM04} for more about the relation between multiple sequence alignment and trace reconstruction.

The multiple sequence alignment problem is also a key motivation for studying circular trace reconstruction. Many important types of DNA, such as mitochondrial DNA in humans and other eukaryotes, chloroplast DNA, bacterial DNA, and DNA in plasmids, are predominantly circular (see, e.g., \cite[pp. 313, 397, 516-517]{RUC11}, or \cite{Wik}). Therefore, understanding circular trace reconstruction could prove useful in reconstructing ancestral sequences for mitochondrial or bacterial DNA. Another problem in computational biology that trace reconstruction may be applicable to is the DNA Data Storage problem, where data is stored in DNA and can be recovered through sequencing, though the stored DNA may mutate over time \cite{CGK12, OAC18}. Recently, long-term
DNA data storage in plasmids has been successfully researched [NPP+18], which further motivates the study of circular trace reconstruction.

Besides the linear trace reconstruction problem, circular trace reconstruction is also closely related to the problem of population recovery from the deletion channel [BCF+19, BCSS19, Nar20], where the goal is to recover an unknown mixture of $\ell$ strings from random traces. Indeed, receiving traces from a circular string is equivalent to receiving traces from a uniform mixture of a linear string along with all of its cyclic shifts, so circular trace reconstruction can be thought of as an instance of population recovery from the deletion channel with mixture size $\ell = n$.

Unfortunately, the best known algorithm for population recovery over worst-case strings requires $\exp \left( \tilde{O}(n^{1/3}) \cdot \ell^2 \right)$ traces [Nar20], which is not useful if $\ell = n$. However, to prove our worst-case upper bound, we will use ideas based on [DOS19, NP17, Nar20] to estimate certain polynomials that depend on the unknown circular string $x$. For the average case problem, i.e., if given a mixture over $\ell$ random strings, population recovery can be done with poly $(\ell, \exp (\log n)^{1/3})$ random traces. While this seemingly implies a poly$(n)$-sample algorithm for average-case circular trace reconstruction, the $n$ cyclic shifts of the circular string are quite similar to each other and thus do not behave like a collection of $n$ independent random strings. Indeed, our techniques for average-case circular trace reconstruction are very different from those developed in [BCSS19].

While circular strings have not been studied before in the context of trace reconstruction, people have studied circular strings and cyclic shifts in the context of edit distance [Mae90, AGMP13], multi-reference alignment [BCSZ19, BNWR19, PWB+19], and other pattern matching problems [CKP+21]. We note that [AGMP13] also applies results from number theory and about cyclotomic polynomials, though the techniques overall are not very similar to ours.

1.3 Proof Overview

In this subsection, we highlight some of the ideas used in Theorems 1.2 and 1.6.

The proof of Theorem 1.2 is partially based on ideas from [DOS19, NP17, Nar20]. In [DOS19, NP17], the authors consider two strings $x, y \in \{0, 1\}^n$ and show how to distinguish between random traces of $x$ and random traces of $y$. To do so, they construct an unbiased estimator for $P(z; x) := \sum x_i z^i$ (or $P(z; y) := \sum y_i z^i$) solely based on the random trace of either $x$ or $y$, for some $z \in \mathbb{C}$. By showing that the unbiased estimator is never “too” large and that $P(z; x)$ and $P(z; y)$ differ enough for an appropriate choice of $z$, they can estimate this quantity using many random traces to distinguish between $x$ and $y$. Unfortunately, in our case, applying the same estimator will give us an unbiased estimator for $P'(z; x) := \mathbb{E}_i[P(z; x^{(i)})]$, where $x^{(i)}$ is the $i$th cyclic shift of $x$: it turns out that $P(z; x) = P(z; y)$ as polynomials in $z$ even if $x, y$ have the same number of 1’s. Our goal will then be to establish some other polynomial $Q(z; x)$ such that we can construct a good unbiased estimator, but at the same time $Q'(z; x) := \mathbb{E}_i[Q(z; x^{(i)})]$ and $Q'(z; y) := \mathbb{E}_i[Q(z; y^{(i)})]$ are distinct polynomials for any distinct cyclic strings $x, y$. We show that the polynomial $Q(z; x) := z^{kn} P(z; x)^k P(z^{-k}; x)$ will do the job, for some some small integer $k$. We provide a (significantly more complicated) unbiased estimator of $Q(z; x)$ using a random trace: the construction is similar to that of [Nar20], which shows how to estimate $P(z; x)^k$ for some integer $k$. To show that $Q(z; x) \neq Q(z; y)$ as polynomials, we first show that $P(z; x)^k P(z^{-k}; x)$ has the special property that if $z$ is a cyclotomic $n$th root of unity, this polynomial is in fact invariant under cyclic shifts! Thus, it just suffices to show that if $x, y \in \{0, 1\}^n$ are not cyclic shifts of each other, there is some $n$th root of unity $\omega$ such that $P(\omega; x)^k P(\omega^{-k}; x) \neq P(\omega; y)^k P(\omega^{-k}; y)$. This will require significant number theoretic
exists some absolute constant $c$ significant amount, if $M$ coefficients. Suppose there is some positive integer $n$ and we wish to recover $\tilde{c}$ trace $\leq 0$ very close to the original polynomial, but differs from $i\sum_{j=0}^n c_i (1 - q)^{100\log n} q^i$, and we wish to recover $c_0$. The $(1 - q)^{100\log n}$ term is a constant that equals $1/\text{poly}(n)$, so it is easy to recover an approximation to $\sum_{i\geq 0} c_i q^i$. We truncating this polynomial at an appropriate degree (approximately $C \log n$ for some large $C$) and show that the truncated polynomial $\sum_{i=0}^{C\log n} c_i x^i$ is very close to the original polynomial, but differs from $\sum_{i=0}^{C\log n} c'_i x^i$ for some $x \in [q, (1 - q)/2]$ by a significant amount, if $c'_0 \neq c_0$, using ideas based on [BEK99]. We can also simulate a trace with deletion probability $x > q$ by taking a “trace of the trace.” This will be sufficient in determining $c_0$, and therefore, the (multi)-set of all consecutive substrings of length $100 \log n$.

The proof of Theorem 2.6 proceeds by showing that the laws of the traces of $x = 10^n 10^{n+1} 10^{n+1}$ and $y = 10^n 10^{n+k} 10^{n+1}$ are close to each other in the sense of Hellinger distance and concluding by a lemma in [H20] that was used in a similar fashion to show a lower bound for linear trace reconstruction. It is first shown that conditioned on a 1 being deleted, a trace from $x$ is equidistributed as a trace from $y$. Then explicit expressions for the probabilities that the trace is $10^n 10^b 10^c$ are computed and compared, yielding an upper bound on the Hellinger distance. The difference between the probabilities for $x$ and $y$ is proportional to the product of $(a-b)(b-c)(a-c)$ and a symmetric polynomial in $a,b,c$. Both $x$ and $y$ consist of three 1’s separated by runs of 0’s of approximate length $n$, so with high probability we have that $a, b, c$ are approximately $np$, with square root fluctuations. The contribution of the $(a-b)(b-c)(a-c)$ term allows us to recover a $\Omega(n^3)$ bound.

\section{Preliminaries}

First, we explain a basic definition we will use involving complex numbers.

**Definition 2.1.** For $z \in \mathbb{C}$, let $|z|$ be the magnitude of $z$, and if $z \neq 0$, let $\arg z$ be the argument of $z$, which is the value of $\theta \in (-\pi, \pi]$ such that $\frac{z}{|z|} = e^{i\theta}$.

Next, we state a Littlewood-type result about bounding polynomials on arcs of the unit circle.

**Theorem 2.2.** [BE97] Let $f(z) = \sum_{j=0}^n a_j z^j$ be a nonzero polynomial of degree $n$ with complex coefficients. Suppose there is some positive integer $M$ such that $|a_0| \geq 1$ and $|a_j| \leq M$ for all $0 \leq j \leq n$. Then, if $A$ is an arc of the unit circle $\{z \in \mathbb{C} : |z| = 1\}$ with length $0 < a < 2\pi$, there exists some absolute constant $c_1 > 0$ such that

$$\sup_{z \in A} |f(z)| \geq \exp \left( -c_1 (1 + \log M) \frac{1}{a} \right).$$

Next, we state two well known results about roots of unity in cyclotomic fields.
Lemma 2.3. [Mar77] Let $\omega = e^{2\pi i/n}$. Then, the set of $\{\omega^k\}$ for $k \in \mathbb{Z}$, $\gcd(k, n) = 1$ are all Galois conjugates. This means that if $P(x)$ is an integer polynomial, then $P(\omega^k) = 0$ if and only if $P(\omega) = 0$ for any $k \in \mathbb{Z}$ with $\gcd(k, n) = 1$. Moreover, $P(\omega) = 0$ if and only if $P$ is a multiple of the $n$th Cyclotomic polynomial.

Lemma 2.4. [Mar77] Let $\omega = e^{2\pi i/n}$ be an $n$th root of unity, and let $\mathbb{Q}[\omega]$ be the $n$th degree cyclotomic field. Then, if $z \in \mathbb{Q}[\omega]$ such that $z^r = 1$ for some integer $r \geq 1$, $z$ must equal $\omega^k$ or $-\omega^k$ for some integer $k$.

Finally, we define the Hellinger distance between two probability measures and state a folklore bound on distinguishing between distributions based on samples in terms of the Hellinger distance.

Definition 2.5. Let $\mu$ and $\nu$ be discrete probability measures over some set $\Omega$. In other words, for $x \in \Omega$, $\mu(x)$ is the probability of selecting $x$ when drawing from the measure $\mu$. Then, the Hellinger distance is defined as

$$d_H(\mu, \nu) = \left( \sum_{x \in \Omega} \left( \sqrt{\mu(x)} - \sqrt{\nu(x)} \right)^2 \right)^{1/2}.$$

The following proposition is quite well-known (see for instance, [HL20, Lemma A.5]).

Proposition 2.6. If $\mu, \nu$ are discrete probability measures, then if given i.i.d. samples from either $\mu$ or $\nu$, one must see at least $\Omega(d_H(\mu, \nu)^{-2})$ i.i.d. samples to determine whether the distribution is $\mu$ or $\nu$ with at least $2/3$ success probability.

3 Worst Case: Upper Bound

In this section, we prove Theorem 1.2, i.e., we provide an $\exp\left(\tilde{O}(n^{1/3})\right)$-sample algorithm for circular trace reconstruction when the length $n$ is a prime or product of two primes.

For a (linear) string $x \in \{0, 1\}^n$ and $z \in \mathbb{C}$, we define $P(z; x) := \sum_{i=1}^n x_i z^i$. The first lemma we require creates an unbiased estimator for $\prod_{i=1}^n P(z_i; x)$ for some complex numbers $z_1, \ldots, z_m$, using only random traces of $x$. The proof of the following lemma greatly resembles the proof of [Nat20, Lemma 4.1], so we defer the proof to Appendix A.

Lemma 3.1. Let $x$ be a linear string of length $n$. Fix $q$ as the deletion probability and $p = 1 - q$ as the retention probability. Then, for any integer $m \geq 1$ and any $Z = (z_1, \ldots, z_m)$ for $z_1, \ldots, z_m \in \mathbb{C}$, there exists some function $g_m(\tilde{x}, Z)$ such that

$$\mathbb{E}_{\tilde{x}}[g_m(\tilde{x}, Z)] = \prod_{k=1}^m \left( \sum_{i=1}^n x_i z_k^i \right),$$

where the expectation is over traces drawn from $x$. Moreover, for any $L \geq 1$, and for all $\tilde{x} \in \{0, 1\}^n$ and all $Z$ such that $|z_1|, \ldots, |z_m| = 1$ and $|\arg z_i| \leq \frac{L}{2}$ for all $1 \leq i \leq m$,

$$|g_m(\tilde{x}, Z)| \leq (p^{-1}mn)^{O(m)} \cdot e^{O(m^2n/(p^2L^2))}.$$
For $x \in \{0, 1\}^n$ and $z \in \mathbb{C}$, let $P(z; x) := \sum_{i=1}^{n} x_i z^i$. Our main goal will be to determine the value of $f_t(z; x) := P(z; x)^t \cdot P(z^{-t}; x)$ for some integer $t$, where $z$ is an $n$th root of unity. Importantly, we note that $f_t(z; x)$ is invariant under rotations of $x$, since for $z = e^{2\pi ik/n}$,

$$\sum_{i=1}^{n} x_{i+1 \pmod n} z^i = \sum_{i=1}^{n} x_i z^{i-1} = P(z; x) \cdot z^{-1}$$

whereas

$$\sum_{i=1}^{n} x_{i+1 \pmod n} z^{-ti} = \sum_{i=1}^{n} x_i z^{-t(i-1)} = P(z^{-t}; x) \cdot z^t$$

Therefore, if we define $x^{(j)}$ as the string $x$ rotated by $j$ places (so $x^{(j)}_i = x_{(i+j) \pmod n}$), then $f(z; x) = f(z; x^{(j)})$ for all $z = e^{2\pi ik/n}$ and $0 \leq j \leq n - 1$.

Now, choose some $z$ with $|z| = 1$ and $|\arg z| \leq \frac{1}{L}$. Also, fix some integer $t$, let $m = t + 1$, and let $Z = (z, \ldots, z, z^{-t})$. Then, if $j$ is randomly chosen in $\{0, 1, \ldots, n - 1\}$ and $\tilde{x}$ is a random trace,

$$\mathbb{E}_x[nz^{tn} \cdot g_m(\tilde{x}, Z)] = (n \cdot z^{tn}) \left(\frac{1}{n} \cdot \sum_{j=0}^{n-1} P(z; x^{(j)})^t \cdot P(z^{-t}; x^{(j)})\right) = \sum_{j=0}^{n-1} z^{tn} \cdot P(z; x^{(j)})^t \cdot P(z^{-t}; x^{(j)}).$$

Note that $\sum_{j=0}^{n-1} z^{tn} \cdot P(z; x^{(j)})^t \cdot P(z^{-t}; x^{(j)})$ is a polynomial of $z$ of degree at most $(t+1)n$ and all coefficients bounded by $n^{t+1}$. We write this polynomial as $Q_t(z; x)$. Thus, if we define $h_t(\tilde{x}, z) := nz^{tn}g_m(\tilde{x}, Z)$, we have that $\mathbb{E}_x[h_t(\tilde{x}, z)] = Q_t(z; x)$ for $\tilde{x}$ a trace of a randomly shifted $x$, and that $|h_t(\tilde{x}; z)| \leq (p^{-1}tn)^O(t) \cdot e^{O(t^2n/(p^2L^2))}$ whenever $|z| = 1$ and $|\arg z| \leq \frac{1}{L}$ for $L \geq 2$, since $m = t + 1$.

Now, we will state two important results that will lead to the proof of the main result.

**Lemma 3.2.** Let $n \geq 2$, and suppose that $x, x'$ are strings in $\{0, 1\}^n$ such that $Q_t(z; x) \neq Q_t(z; x')$ as polynomials in $z$. Then, there is a uniform constant $c_2$ such that for any $L \geq 2$, there exists $z$ such that $|z| = 1$, $|\arg z| \leq \frac{1}{L}$, and

$$|Q_t(z; x) - Q_t(z; x')| \geq n^{-c_2tL}.$$

**Proof.** Note that $Q_t(z; x) - Q_t(z; x')$ is a nonzero polynomial in $z$ of degree at most $(t+1)n$ and with all coefficients bounded by $2n^{t+1}$. Therefore, by Theorem 2.2

$$\sup_{|z|=1, |\arg z| \leq 1/L} |Q_t(z; x) - Q_t(z; x')| \geq \exp\left(-c_1(1 + \log(2n^{t+1}))\right) \geq \exp(-c_2 \cdot L \cdot t \cdot \log n) = n^{-c_2tL},$$

where we note that the arc $\{ z : |z| = 1, |\arg z| \leq \frac{1}{L} \}$ has length $\frac{2}{L}$. \hfill $\square$

The next important result we need will be Theorem 3.3. We defer the full proof of Theorem 3.3 to Subsection A.3 but as the proof of the case where $n$ is prime is simpler, we prove this special case here. Using this, we can get an $\exp\left(O(n^{1/3})\right)$ sample upper bound at least for $n$ prime.

**Proposition 3.3.** Suppose that $n = p$ is prime, and $a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1} \in \{0, 1\}$ such that for all $0 \leq k < p$, there is some integer $c_k$ such that $\sum_{i=0}^{p-1} a_i = \omega^{ck} \cdot \sum_{i=0}^{p-1} b_i$. Then, the sequences $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_n\}$ are equivalent up to a cyclic permutation.
Proof. First, \( \sum_{i=0}^{p-1} a_i = \omega^0 \cdot \sum_{i=0}^{p-1} b_i \). Since \( \sum_{i=0}^{p-1} a_i \) and \( \sum_{i=0}^{p-1} b_i \geq 0 \) are both positive real numbers, and since \( \omega^0 \) is a root of unity, we must have that \( \sum_{i=0}^{p-1} a_i = \sum_{i=0}^{p-1} b_i \). In the case \( p = 2 \), this alone proves the proposition, so we now assume \( p \) is odd.

Now, we have that \( \sum_{i=0}^{p-1} a_i \omega^i = \omega^{p-1} \cdot \sum_{i=0}^{p-1} b_i \omega^i \). Letting \( b'_i = b_i(i-1) \pmod{p} \), we have that \( b' \) is a cyclic shift of \( b \), and \( \sum_{i=0}^{p-1} a_i = \sum_{i=0}^{p-1} b'_i \) and \( \sum_{i=0}^{p-1} a_i \omega^i = \sum_{i=0}^{p-1} b'_i \omega^i \). Letting \( Q(x) = \sum_{i=0}^{p-1} (a_i - b'_i)x^i \), we have that \( \omega \) and \( 1 \) are both roots of \( Q(x) \). Since \( Q(x) \) is an integer-valued polynomial, this implies that all Galois conjugates of \( \omega \) are roots, so \( 1, \omega, \omega^2, \ldots, \omega^{p-1} \) are roots of \( Q(x) \). Thus, \( x^p - 1 \) divides \( Q(x) \). But since \( Q(x) \) has degree at most \( p - 1 \), \( Q(x) \) must equal \( 0 \), so \( a_i = b'_i \) for all \( i \). Since the sequence \( b' \) is just a shift of \( b \), we are done. \( \square \)

Finally, we are ready to prove Theorem 1.4.

Proof of Theorem 1.4. Let \( L = \Theta(n^{1/3}(\log n)^{-1/3}p^{-2/3}) \), and suppose that we are trying to distinguish between the original circular string being \( a = a_1a_2 \cdots a_n \) or \( b = b_1b_2 \cdots b_n \), where \( a, b \) are distinct, even up to cyclic shifts. First, we claim that for some \( 0 \leq t \leq n-1 \), some \( 2 \leq t \leq 5 \), and \( z = \omega^t \), we have that \( P(z; a)^t P(z^{-t}; a) \neq \frac{P(z; b)^t P(z^{-t}; b)}{\omega^k} \), where we recall that \( \omega := e^{2\pi i/n} \).

To prove this, first choose \( k \) such that \( \sum_{i=1}^{n} a_i \omega^{-i} \neq \omega^k \cdot \sum_{i=1}^{n} b_i \omega^{-i} \) for all integers \( c_k \), which exists by Theorem 1.3. If \( k = 0 \), then \( P(\omega^k; a) = P(1; a) \) and \( P(\omega^k; b) = P(1; b) \) are distinct nonnegative integers, so we trivially have \( P(1; a)^t P(1; a) \neq \frac{P(1; b)^t P(1; b)}{\omega^k} \). Otherwise, let \( t \) be the smallest prime that doesn’t divide \( \frac{n}{\gcd(n,k)} \) (so \( t \leq 5 \) as \( n \) has at most 2 prime factors). If \( \sum_{i=1}^{n} a_i \omega^{ik} = 0 \), then \( \sum_{i=1}^{n} b_i \omega^{ik} \neq 0 \). Now, since \( \omega^{-t} \) is a Galois conjugate of \( \omega^k \) (since \( t \nmid n \)), we also have that \( \sum_{i=1}^{n} b_i \omega^{-ti} \neq 0 \). This means that \( P(\omega^k; a) = 0 \) and \( P(\omega^k; b) = 0 \). Otherwise, let \( P(\omega^k; a) = \sum_{i=1}^{n} a_i \omega^{ik} \) and \( P(\omega^k; b) = \sum_{i=1}^{n} b_i \omega^{ik} \) be both nonzero. This means that for all \( r \geq 0 \), \( P(\omega^{(-t)^r}; a) \) and \( P(\omega^{(-t)^r}; b) \) are both nonzero, since \( \omega^{(-t)^r} \) and \( \omega^k \) are Galois conjugates. This means that if \( P(z; a)^t P(z^{-t}; a) = P(t; b)^2 P(z^{-t}; b) \) for all \( z = \omega^{(-t)^r} \), then

\[
\frac{P(\omega^{(-t)^r}; a)}{P(\omega^{(-t)^r}; a)^{-t}} = \frac{P(z^{-t}; a)}{P(z; a)^{-t}} = \frac{P(z^{-t}; b)}{P(z; b)^{-t}} = \frac{P(\omega^{(-t)^r+1}; b)}{P(\omega^{(-t)^r}; b)^{-t}}
\]

for all \( r \geq 0 \), so we inductively have that

\[
\frac{P(\omega^{(-t)^r}; a)}{P(\omega^{(-t)^r}; a)^{-t}} = \frac{P(\omega^{(-t)^r}; b)}{P(\omega^{(-t)^r}; b)^{-t}}.
\]

Now, letting \( r = \varphi \left( \frac{n}{\gcd(n,k)} \right) \), we know that \( k \cdot (-t)^r \equiv k \pmod{n} \) by Euler’s theorem, which means that \( \omega^{(-t)^r} = \omega^k \). Thus,

\[
P(\omega^k; a)^{1-(-t)^r} = P(\omega^k; b)^{1-(-t)^r}.
\]

Since \( k \neq 0 \), we have that \( \frac{n}{\gcd(n,k)} > 1 \) so \( r \geq 1 \). Thus, since \( t \geq 2 \), \( 1 - (-t)^r \neq 0 \). Now, since \( P(\omega^k; a), P(\omega^k; b) \) are nonzero, we have that \( \frac{P(\omega^k; a)}{P(\omega^k; b)} \) is a \( |1 - (-t)^r| \)th root of unity. Also, \( P(\omega^k; a), P(\omega^k; b) \in \mathbb{Q}[\omega] \), which means \( \frac{P(\omega^k; a)}{P(\omega^k; b)} \in \mathbb{Q}[\omega] \). However, all roots of unity in \( \mathbb{Q}[\omega] \) are of the form \( \pm \omega^i \) for some \( i \), and since \( (-t)^r - 1 \) is odd if \( n \) is odd (since \( t = 2 \)), we must have that \( \frac{P(\omega^k; a)}{P(\omega^k; b)} = \omega^{ck} \) for some integer \( c_k \). This is a contradiction, so we must have that \( P(z; a)^t P(z^{-t}; a) \neq \frac{P(z; b)^t P(z^{-t}; b)}{\omega^k} \), for some \( z = \omega^{(-t)^r}, r \geq 0 \).
Next, as we have already noted, \( P(z; a)^t P(z^{-t}; a) \) is invariant under rotation of \( a \), and \( P(z; b)^t P(z^{-t}; b) \) is invariant under rotation of \( b \). Thus, by our definition of \( \mathcal{Q}_t(z; x) \), we have that \( \mathcal{Q}_t(z; a) \neq \mathcal{Q}_t(z; b) \). Thus, by Lemma 3.2, there is some \( z \) such that \( |z| = 1, |\arg z| \leq \frac{1}{4} \), and

\[
|\mathcal{Q}_t(z; a) - \mathcal{Q}_t(z; b)| \geq n^{-c_2 t L} \geq n^{-5c_2 L}.
\]

Therefore, for \( L = \Theta(n^{1/3}(\log n)^{-1/3} p^{-2/3}) \), there exists some \( z \) with \( |z| = 1 \) and \( |\arg z| \leq \frac{1}{4} \) and some \( 2 \leq t \leq 5 \) such that

\[
|\mathcal{Q}_t(z; a) - \mathcal{Q}_t(z; b)| \geq n^{-5c_2 L} \geq \exp\left(-c_3 \cdot n^{1/3}(\log n)^{2/3} p^{-2/3}\right)
\]

but

\[
|\mathcal{h}_t(x, z)| \leq (p^{-1} n)^{O(1)} \cdot \exp\left(\frac{n}{p^2 L^2}\right) \leq \exp\left(c_4 \cdot n^{1/3}(\log n)^{2/3} p^{-2/3}\right).
\]

Therefore, by choosing \( z \) and \( t \) appropriately, taking \( R = \exp\left(O\left(n^{1/3}(\log n)^{2/3} p^{-2/3}\right)\right) \) traces \( \hat{x}^{(1)}, \ldots, \hat{x}^{(R)} \), and letting \( \mathcal{h}_t(x) \) denote the average of \( \mathcal{h}_t(\hat{x}^{(i)}, z) \) for all \( i \), the Chernoff bound tells us that with probability at least \( 1 - 10^n \), \( |\mathcal{h}_t(z) - \mathcal{Q}_t(z; a)| \leq \frac{1}{2} \cdot \exp\left(c_4 \cdot n^{1/3}(\log n)^{2/3} p^{-2/3}\right) \) if the original string were \( a \), and \( |\mathcal{h}(z) - \mathcal{Q}_t(z; b)| \leq \frac{1}{2} \cdot \exp\left(c_4 \cdot n^{1/3}(\log n)^{2/3} p^{-2/3}\right) \) if the original string were \( b \). Thus, by returning \( h(z) \) if \( h(z) \) is closer to \( \mathcal{Q}_t(z; a) \) and returning \( b \) otherwise, we can distinguish between the original string being \( a \) or \( b \) using \( \exp\left(O\left(n^{1/3}(\log n)^{2/3} p^{-2/3}\right)\right) \) traces, with \( 1 - 10^n \) failure probability.

Thus, to reconstruct the original string \( x \), we simply run the distinguishing algorithm for all pairs \( a, b \in \{0, 1\}^n \) such that \( a \neq b \), using the same \( R \) traces \( \hat{x}^1, \ldots, \hat{x}^R \). With probability at least \( 1 - (4/10)^n \geq 1 - 2^{-n} \), the true string \( x \) will be the only string such that the distinguishing algorithm will successfully choose \( x \) over all other strings. Thus, for \( n \) a prime or a product of two primes, the circular trace reconstruction problem can be solved using \( \exp\left(O\left(n^{1/3}(\log n)^{2/3} p^{-2/3}\right)\right) \) traces. \( \square \)

### 4 Average Case: Upper Bound

We now consider the situation in which the unknown circular string \( x \) is random. We will suppose that \( x \) is equidistributed as a random circular string in which each bit is 0 or 1 with \( \frac{1}{2} \) probability. Note that this distribution is not uniform over all possible circular strings. However, our arguments can easily be modified to handle such a situation. We use the randomness to rule out certain problematic strings with high probability, and this can be done for uniform random circular strings as well as other distributions, for example if independently each bit is biased towards 0 or 1.

**Theorem 4.1.** Let \( x \) be a random (in the sense described above) unknown circular string of length \( n \) and let \( q \) be the deletion probability of each element. Then there exists a constant \( C_q \) depending only on \( q \) such that we can determine \( x \) with failure probability at most \( n^{-10} \) using \( O(n^{C_q}) \) traces.

In what follows, we will let \( x = x_1 \cdots x_n \) and take indices of bits in \( x \) modulo \( n \). Let \( k = 100 \log n \). We first note that with high probability, all of the consecutive substrings of \( x \) of length \( k \) and \( k - 1 \) are pairwise distinct. We will refer to such strings as *regular* strings. Indeed, the probability that \( x_i \cdots x_{i+k-1} = x_j \cdots x_{j+k-1} \) for \( i \neq j \) is \( 2^{-k} \) (where indices are taken modulo \( n \)), and union bounding over all \( i, j \) as well as both \( k \) and \( k - 1 \) gives a failure probability of at most \( O(n^2 2^{-k}) \ll n^{-10} \).

If we assume that \( x \) is regular, the length \( k \) consecutive substrings of \( x \) uniquely determine \( x \). Indeed, given \( x_i \cdots x_{i+k-1} \), we can uniquely determine \( x_{i+k} \) as there is a unique length \( k \) consecutive
substring of $x$ that begins with $x_{i+1} \cdots x_{i+k-1}$. Iteratively applying this allows us to recover the entire string $x$. Thus, to prove Theorem 4.3 it suffices to prove Lemma 1.3 i.e., to determine how many times each length $k$ substring appears consecutively in $x$ using $O(n^{C_k})$ traces, which will allow us to recover $x$ if $x$ is regular.

We will show the existence of $C_q$ so that for any string $s$ of length $k$, we can distinguish between strings $x$ and $y$ correctly using $O(n^{C_q})$ samples with failure probability $10^{-n}$, if the number of consecutive occurrences of $s$ in $x$ and in $y$ differ, from which a union bound over all strings $s$ of length $k$ and all pairs of strings $x,y$ of length $n$ shows the result. Let $\alpha$ denote a sufficiently large constant only depending on $q$ that we will determine later. For $0 \leq i \leq n - k$, let $c_i$ denote the number of (not necessarily consecutive) occurrences of $s$ in $x$ contained in a consecutive substring of $x$ of length at most $i + k$. Similarly, let $d_i$ denote the number of (not necessarily consecutive) occurrences of $s$ in $y$ contained in a consecutive substring of $y$ of length at most $i + k$.

By assumption, we have that $c_0 \neq d_0$. By casework on the last bit of the occurrence of $s$, we have that $c_i, d_i \leq n(i+k \choose k)$. Let $P(t) = \sum_{i=0}^{\alpha k} c_i t^i$ and $Q(t) = \sum_{i=0}^{\alpha k} d_i t^i$. Moreover, the following is true:

**Lemma 4.2.** The probability that a trace of $x$ starts with $s$ (where a random bit in the string is chosen as the beginning before bits are deleted) is $\frac{1}{n}(1-q)^kP(q) + O(q^{\alpha k}(\alpha + 1)^k e^k)$. Similarly, the probability that a trace of $y$ starts with $s$ is $\frac{1}{n}(1-q)^kQ(q) + O(q^{\alpha k}(\alpha + 1)^k e^k)$.

**Proof.** To compute the probability that a trace of $x$ starts with $s$, we do casework on how many bits are deleted before the last bit in the occurrence of $s$. If $i$ bits are deleted, then note that there are $c_i$ ways for it to be done by definition. Each such way has a probability of $\frac{1}{n}(1-q)^k q^i$ to occur. Indeed, for each way there is a $\frac{1}{n}$ probability that the correct starting bit is chosen, and the probability that only the bits corresponding to the specific instance of $s$ are kept is $(1-q)^k q^i$. It follows that the probability is exactly $\frac{1}{n}(1-q)^k \sum_{i=0}^{\alpha k} c_i q^i$.

It remains to show that $\frac{1}{n}(1-q)^k \sum_{i=0}^{\alpha k} c_i q^i = O(q^{\alpha k}(\alpha + 1)^k e^k)$. As mentioned before, we have that $c_i \leq n(i+k \choose k)$. Thus, this term is at most $\sum_{i>\alpha k} (i+k \choose k) q^i \leq (\alpha k \choose k) q^{\alpha k} \sum_{i>\alpha k} \left(\frac{q(\alpha + 1)}{\alpha}\right)^i$. Indeed, the ratio of consecutive terms in the sequence $(i+k \choose k) q^i$ is equal to $q^{i+k} \frac{i+k}{\alpha}$ for a sufficiently large choice of $\alpha$, $\frac{q(\alpha + 1)}{\alpha} < 1$, so $\sum_{i>\alpha k} (i+k \choose k) q^i = O((\alpha k \choose k) q^{\alpha k}) = O(q^{\alpha k}(\alpha + 1)^k e^k)$ by Stirling’s approximation.

The argument for $y$ is analogous. □

Lemma 4.2 allows us to estimate $P(q)$ and $Q(q)$ up to an $O(n(1-q)^{-k}q^{\alpha k}(\alpha + 1)^k e^k)$ error by looking at how often traces of $x$ or $y$ begin with $s$, and then dividing by $\frac{1}{n}(1-q)^k$. So long as $P(q)$ and $Q(q)$ are sufficiently far apart, a Chernoff bound allows us to determine with high probability if the traces came from $x$ or $y$. However, it may be the case that $P(q)$ and $Q(q)$ are quite close. To remedy this, we observe that it is possible to simulate higher deletion probabilities $q' > q$. Indeed, this can be achieved by deleting each bit in traces received independently with probability $\frac{q}{q'}$. Thus, it suffices to find $q' \in [q,r]$ with $P(q')$ and $Q(q')$ far apart for some $q < r < 1$. The existence of such a $q'$ is proven by the following Littlewood-type result of Borwein, Erdélyi, and Kós.

**Theorem 4.3 ([BEK99], Theorem 5.1).** There exist absolute constants $c_1 > 0$ and $c_2 > 0$ such that if $f$ is a polynomial with coefficients in $[-1,1]$ and $a \in (0,1]$, then

$$|f(0)|^{c_1/a} \leq \exp \left(\frac{c_2}{a} \sup_{z \in [1-a,1]} |f(z)|\right).$$
Proof of Theorem 4.4. Let $r = \frac{2 + \epsilon}{2}$. We first apply Theorem 4.3 to $(\binom{ak}{k})^{-1}(P(rx) - Q(rx))$ and $a = 1 - q/r$. Here, we are using the fact that the coefficients of $P$ and $Q$ are bounded in magnitude by $(\binom{ak}{k})$ by previous observations, and that $|P(0) - Q(0)| \geq 1$. Theorem 4.3 tells us that

$$
\binom{ak + k}{k}^{-\alpha/2} \leq \exp \left( \frac{c_2}{a} \binom{ak + k}{k}^{-1} \sup_{z \in [1-a,1]} |P(rz) - Q(rz)| \right)
$$

$$
= \exp \left( \frac{c_2}{a} \binom{ak + k}{k}^{-1} \sup_{q' \in [q,r]} |P(q') - Q(q')| \right),
$$

or

$$
\sup_{q' \in [q,r]} |P(q') - Q(q')| \geq c_3 \binom{ak + k}{k}^{-c_4}
$$

for some constants $c_3$ and $c_4$ that only depend on $q$.

In particular, this is much larger than $10^k n(1 - r)^{-k} r^\alpha (\alpha + 1)^k e^k$ for sufficiently large values of $\alpha$ ($\alpha$ may depend on $q$). Indeed, after taking $k$th roots and using Stirling’s approximation this reduces to showing that $e(\alpha + 1)^k e^k$ for sufficiently large $\alpha$ where $c_5$ is some constant that only depends on $q$, which is clear (since $0 < r < 1$ is fixed and $n^{1/k} < 2$). Thus, for any $q' \in [q,r]$, the error term $\frac{1}{n} (1 - q')^k \sum_{i=q+1}^{n-k} c_i (q')^i = O((q')^\alpha (\alpha + 1)^k e^k)$ is at most $10^{-k}$ times $\frac{1}{n} (1 - q')^k \sup_{q' \in [q,r]} |P(q') - Q(q')|$. Hence, for some $q' \in [q,r]$, the probability that a trace begins with $s$ under bit deletion with probability $q'$ differs between $x$ and $y$ by $\Omega(10^k n(1 - r)^{-k} r^\alpha (\alpha + 1)^k e^k)$ is at most $10^{-k}$ times $\frac{1}{n} (1 - q')^k \sup_{q' \in [q,r]} |P(q') - Q(q')|$. Hence, we can distinguish between $x$ and $y$ using $O(n^C n)$ traces with failure probability at most $\exp(-\Omega(n))$, so the theorem follows.

5 Worst Case: Lower Bound

In this section, we prove Theorem 1.6 and demonstrate that worst-case circular trace reconstruction requires $\Omega(n^3)$ traces. We first record the following lemma from [HL20] expressing the number of independent samples required to distinguish between two probability measures $\mu$ and $\nu$ in terms of their Hellinger distance $d_H(\mu, \nu)$, defined to be $(\sum_{x \in X} (\mu(\{x\}) - \nu(\{x\}))^2)^{1/2}$ where the sum is over all events in some discrete sample space $X$. Let $d_{TV}(\mu, \nu)$ denote the total variation distance between $\mu$ and $\nu$ and $\mu^m$ denote the law of $n$ independent samples from $\mu$.

Lemma 5.1 ([HL20], Lemma A.5). If $\mu$ and $\nu$ are probability measures satisfying $d_H(\mu, \nu) \leq 1/2$, then for $m \geq 1/(4d_H^2(\mu, \nu))$, we have that $1 - d_{TV}(\mu^m, \nu^m) \geq e$ if $m \leq \frac{\log(1/e)}{d_H^2(\mu, \nu)}$.

Note that the number of samples $m$ required to distinguish between $\mu$ and $\nu$ is given by the total variation distance between $\mu^m$ and $\nu^m$. Thus, it requires $\Omega(d_H^{-2}(\mu, \nu))$ samples to distinguish between two probability measures $\mu$ and $\nu$.

Proof of Theorem 1.6. We now specialize to the case of distinguishing between $x = 10^n 10^{n+1} 10^{n+k}$ and $y = 10^n 10^{n+k} 10^{n+1}$ from independent traces. Let $\mu$ and $\nu$ respectively denote the laws of traces from $x$ and $y$. We will show that $d_H^2(\mu, \nu) = O((n \log n)^{3/2})$, which establishes the result by Lemma 5.1.
First, we note that conditional on the first 1 in \( x \) being deleted, the resulting trace is equidistributed as a trace from \( y \) conditioned on the second 1 being deleted, as in both cases we obtain a trace from the circular string \( 10^{a+1}10^{2n+k} \). Similar arguments for other cases show that conditioned on any 1 being deleted, traces from \( x \) and \( y \) are equal in law. Thus, the resulting string must have three 1’s to contribute to the Hellinger distance. We will henceforth assume that the resulting trace is of the form \( 10^a10^b10^c \) for some nonnegative integers \( a, b, c \).

We now compute the ratio \( \frac{\mu(\{10^a10^b10^c\})}{\nu(\{10^a10^b10^c\})} \), and show that it is typically \( 1 + O((n/\log n)^{3/2}) \). We have that

\[
\frac{\mu(\{10^a10^b10^c\})}{\nu(\{10^a10^b10^c\})} = \frac{(n)(n+1)(n+k) + (n)(n+1)(n+k) + (n)(n+1)(n+k)}{(n)(n+1)(n+k) + (n)(n+1)(n+k) + (n)(n+1)(n+k)}.
\]

It follows that

\[
\frac{\mu(\{10^a10^b10^c\})}{\nu(\{10^a10^b10^c\})} = \frac{\frac{1}{(n+1-c)(n+1-a)(n+k-c)} + \frac{1}{(n+1-c)(n+1-a)(n+k-a)} + \frac{1}{(n+1-c)(n+1-a)(n+k-b)}}{(n+1-a)(n+1-b)(n+k-c)} + \frac{1}{(n+1-a)(n+1-b)(n+k-a)} + \frac{1}{(n+1-a)(n+1-b)(n+k-b)}.
\]

Multiplying the numerator and denominator by \( \prod_{i=1}^k (n+i-a)(n+i-b)(n+i-c) \) results in

\[
S_1 = \prod_{i=1}^k (n+i-a) \prod_{i=2}^k (n+i-b) \prod_{i=2}^k (n+i-c) \prod_{i=1}^k (n+i-a) \prod_{i=2}^k (n+i-c),
\]

and

\[
S_2 = \prod_{i=2}^k (n+i-b) \prod_{i=1}^k (n+i-a) \prod_{i=1}^k (n+i-c) \prod_{i=2}^k (n+i-b) \prod_{i=2}^k (n+i-c),
\]

respectively. We have that \( S_1 - S_2 = (a-b) \prod_{i=2}^k (n+i-a)(n+i-b) + (b-c) \prod_{i=2}^k (n+i-b)(n+i-c) + (c-a) \prod_{i=2}^k (n+i-c)(n+i-a) \). This is an alternating polynomial in \( a, b, c \), i.e. applying a permutation \( \sigma \) to \( a, b, c \) changes the sign of the polynomial by the sign of \( \sigma \). Hence, it can be written in the form \( (a-b)(b-c)(a-c)P_k(n, a, b, c) \), where \( P_k \) is a polynomial in \( n, a, b, c \) of degree \( 2k - 4 \) since \( S_1 \) and \( S_2 \) have degree \( 2k - 1 \).

By a standard Chernoff bound, there exists a constant \( C \) such that with probability at least \( 1 - n^{-100} \), \( a, b, c \in [np - C\sqrt{n\log n}, np + C\sqrt{n\log n}] \). When this occurs, we have that \( S_2 = \Omega(n^{2k-1}) \) and \( |S_1 - S_2| = O((n\log n)^{3/2} n^{2k-4}) \), so \( \frac{\mu(\{10^a10^b10^c\})}{\nu(\{10^a10^b10^c\})} \in [1 - (c \log n/n)^{3/2}, 1 + (c \log n/n)^{3/2}] \) for some constant \( c \). We thus have that

\[
d_H^2(\mu, \nu) = \sum_{a,b,c \geq 0} \left( \mu(\{10^a10^b10^c\}) - \nu(\{10^a10^b10^c\}) \right)^2 \leq 2n^{-100}
\]

\[
+ \sum_{a,b,c \in [np - C\sqrt{n\log n}, np + C\sqrt{n\log n}]} \nu(\{10^a10^b10^c\})^2 \left( 1 - \frac{\mu(\{10^a10^b10^c\})}{\nu(\{10^a10^b10^c\})} \right)^2 = O((\log n/n)^3).
\]

It follows by Lemma 5.1 that it requires \( \Omega(n^3/\log^3 n) \) samples to distinguish between traces from \( x \) and \( y \), as desired.
Acknowledgments

This research was partially supported by the MIT Akamai Fellowship, the NSF Graduate Fellowship, and by NSF-DMS grant 1949884 and NSA grant H98230-20-1-0009. The first author thanks Piotr Indyk for many helpful discussions and feedback, as well as Mehtaab Sawhney for pointers to some references. The second author thanks Professor Joe Gallian for running the Duluth REU at which part of this research was conducted, as well as program advisors Amanda Burcroff, Colin Defant, and Yelena Mandelshtam for providing a supportive environment.

References

[AGMP13] Alexandr Andoni, Assaf Goldberger, Andrew McGregor, and Ely Porat. Homomorphic fingerprints under misalignments: Sketching edit and shift distances. In *Proceedings of the 45th Annual ACM SIGACT Symposium on Theory of Computing*, pages 931–940, 2013.

[BCF19+] Frank Ban, Xi Chen, Adam Freilich, Rocco A. Servedio, and Sandip Sinha. Beyond trace reconstruction: Population recovery from the deletion channel. In *60th Annual IEEE Symposium on Foundations of Computer Science*, pages 745–768, 2019.

[BCSS19] Frank Ban, Xi Chen, Rocco A. Servedio, and Sandip Sinha. Efficient average-case population recovery in the presence of insertions and deletions. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, pages 44:1–44:18, 2019.

[BCSZ19] Afonso S. Bandeira, Moses Charikar, Amit Singer, and Andy Zhu. Multireference alignment using semidefinite programming. In *Fifth Annual ACM Conference on Innovations in Theoretical Computer Science*, pages 745–768, 2019.

[BE97] Peter Borwein and Tamás Erdélyi. Littlewood-type polynomials on subarcs of the unit circle. *Indiana University Mathematics Journal*, 46(4):1323–1346, 1997.

[BEK99] Peter Borwein, Tamás Erdélyi, and Géza Kós. Littlewood-type problems on [0, 1]. *Proceedings of the London Mathematical Society*, 3(79):22–46, 1999.

[BKKM04] Tugkan Batu, Sampath Kannan, Sanjeev Khanna, and Andrew McGregor. Reconstructing strings from random traces. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 910–918, 2004.

[BLS20] Joshua Brakensiek, Ray Li, and Bruce Spang. Coded trace reconstruction in a constant number of traces. In *61st IEEE Annual Symposium on Foundations of Computer Science, to appear*, 2020.

[BNWR19] Afonso S. Bandeira, Jonathan Niles-Weed, and Philippe Rigollet. Optimal rates of estimation for multi-reference alignment. *Mathematical Statistics and Learning*, 2(1):25–75, 2019.
[CDL⁺20] Xi Chen, Anindya De, Chin Ho Lee, Rocco A. Servedio, and Sandip Sinha. Polynomial-time trace reconstruction in the smoothed complexity model. CoRR, abs/2008.12386, 2020.

[CGK12] George M. Church, Yuan Gao, and Sriram Kosuri. Next-generation digital information storage in dna. Science, 337(6102):1628, 2012.

[CGMR19] Mahdi Cheraghchi, Ryan Gabrys, Olgica Milenkovic, and João Ribeiro. Coded trace reconstruction. In Information Theory Workshop, 2019.

[Cha19] Zachary Chase. New lower bounds for trace reconstruction. CoRR, abs/1905.03031, 2019.

[CKP⁺21] Panagiotis Charalampopoulos, Tomasz Kociumaka, Solon P. Pissis, Jakub Radoszewski, Wojciech Rytter, Juliusz Straszynski, Tomasz Walen, and Wiktor Zuba. Circular pattern matching with k mismatches. Journal of Computer and System Sciences, 115:73–85, 2021.

[DOS19] Anindya De, Ryan O’Donnell, and Rocco A. Servedio. Optimal mean-based algorithms for trace reconstruction. Annals of Applied Probability, 29(2):851–874, 2019.

[DRR19] Sami Davies, Miklos Racz, and Cyrus Rashtchian. Reconstructing trees from traces. In Conference On Learning Theory, pages 961–978, 2019.

[HHP18] Lisa Hartung, Nina Holden, and Yuval Peres. Trace reconstruction with varying deletion probabilities. In Proceedings of the Fifteenth Workshop on Analytic Algorithmics and Combinatorics, pages 54–61, 2018.

[HL20] Nina Holden and Russell Lyons. Lower bounds for trace reconstruction. Annals of Applied Probability, 30(2):503–525, 2020.

[HMPW08] Thomas Holenstein, Michael Mitzenmacher, Rina Panigrahy, and Udi Wieder. Trace reconstruction with constant deletion probability and related results. In Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 389–398, 2008.

[HPP18] Nina Holden, Robin Pemantle, and Yuval Peres. Subpolynomial trace reconstruction for random strings and arbitrary deletion probability. In Conference On Learning Theory, pages 1799–1840, 2018.

[KM05] Sampath Kannan and Andrew McGregor. More on reconstructing strings from random traces: insertions and deletions. In Proceedings of the 2005 IEEE International Symposium on Information Theory, pages 297–301, 2005.

[KMMP19] Akshay Krishnamurthy, Arya Mazumdar, Andrew McGregor, and Soumyabrata Pal. Trace reconstruction: Generalized and parameterized. In Proceedings of the 27th Annual European Symposium on Algorithms, pages 68:1–68:25, 2019.

[Lev01a] Vladimir I. Levenshtein. Efficient reconstruction of sequences. IEEE Trans. Information Theory, 47(1):2–22, 2001.
Vladimir I. Levenshtein. Efficient reconstruction of sequences from their subsequences or supersequences. *J. Comb. Theory, Ser. A*, 93(2):310–332, 2001.

Maurice Maes. On a cyclic string-to-string correction problem. *Information Processing Letters*, 35(2):73–78, 1990.

Daniel A. Marcus. *Number Fields*. Springer International Publishing, 1977.

Andrew McGregor, Eric Price, and Sofya Vorotnikova. Trace reconstruction revisited. In *Proceedings of the 22nd Annual European Symposium on Algorithms*, pages 689–700, 2014.

Shyam Narayanan. Population recovery from the deletion channel: Nearly matching trace reconstruction bounds. *CoRR*, abs/2004.06828, 2020.

Fedor Nazarov and Yuval Peres. Trace reconstruction with $\exp(o(n^{1/3}))$ samples. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1042–1046, 2017.

Hoang Hiep Nguyen, Jeho Park, Seon Joo Park, Chang-Soo Lee, Seungwoo Hwang, Yong-Beom Shin, Tai Hwan Ha, and Moonil Kim. Long-term stability and integrity of plasmid-based dna data storage. *Polymers*, 10(1):28, 2018.

Lee Organick, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z Racz, Govinda Kamath, Parikshit Gopalan, Bichlien Nguyen, Christopher N Takahashi, Sharon Newman, Hsing-Yeh Parker, Cyrus Rashtchian, Kendall Stewart, Gagan Gupta, Robert Carlson, John Mulligan, Douglas Carmean, Georg Seelig, Luis Ceze, , and Karin Strauss. Random access in large-scale dna data storage. *Nature Biotechnology*, 36:242–248, 2018.

Amelia Perry, Jonathan Weed, Afonso S. Bandeira, Philippe Rigollet, and Amit Singer. The sample complexity of multireference alignment. *SIAM Journal on Mathematics of Data Science*, 1(3):497–517, 2019.

Yuval Peres and Alex Zhai. Average-case reconstruction for the deletion channel: Subpolynomially many traces suffice. In *58th IEEE Annual Symposium on Foundations of Computer Science*, pages 228–239, 2017.

Jane B. Reece, Lisa A. Urry, Michael L. Cain, Steven A. Wasserman, Peter V. Minorsky, and Robert B. Jackson. *Campbell Biology*. Pearson, 9th edition, 2011.

Krishnamurthy Viswanathan and Ram Swaminathan. Improved string reconstruction over insertion-deletion channels. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 399–408, 2008.

Circular DNA. Available at [https://en.wikipedia.org/wiki/Circular_DNA](https://en.wikipedia.org/wiki/Circular_DNA)
A Omitted Proofs

A.1 Proof of Proposition 1.1

Here, we prove Proposition 1.1 which shows that circular trace reconstruction is at least as hard as linear trace reconstruction in both the worst-case and average case models for any choice of q.

Proof of Proposition 1.1. Let \( m \geq 2n \), and suppose that using \( T_1 = T_1(m, q) \) traces, we can solve worst-case circular trace reconstruction over length \( m \) strings with failure probability \( \delta \). Then, suppose we are given \( T_1 \) traces of some unknown linear string \( x \) of length \( n \). We will reconstruct \( x \) as follows. First, the algorithm creates a random binary string \( y \) of length \( m - n \). Then, the algorithm lets \( x' \) be the circular string \( x \circ y \), i.e. \( x \) concatenated with \( y \), which has length \( m \). While we do not know \( x' \), given a random trace \( \tilde{x}_i \) of \( x \), we can create a random trace \( \tilde{x}_i' \) of \( x' \) by creating a random trace of \( y \) (with deletion probability \( q \)) and appending it to \( \tilde{x}_i \), and then randomly rotating it. Doing this for each trace gives us \( T_1 \) random traces of the circular string \( x' \), which allows us to reconstruct \( x' \) with probability \( 1 - \delta \). Now, the string \( y \) appears exactly once (consecutively) in the circular string \( x' \) with failure probability exponentially small in \( n \) since \( m \geq 2n \), and since we know \( y \), we would be able to find the unique copy of \( y \) in \( x' \) and thus recover the linear string \( x \) with failure probability \( \delta + e^{-\Omega(n)} \).

The same argument works in the average case. Suppose using \( T_2 = T_2(m, q) \) traces, we can solve average-case circular trace reconstruction with probability \( \delta \), where the average string is generated by creating a uniformly random binary (linear) string and making it circular. Then, if given \( T_2 \) random traces of a random linear string \( x \) of length \( n \), our algorithm works the same way: creating a random string \( y \) of length \( m - n \), appending it to \( x \), and then recovering \( x \) with 1 - \( e^{-\Omega(n)} \) probability, there is a unique copy of \( y \) in \( x' \).

A.2 Proof of Lemma 3.1

Here, we prove Lemma 3.1, which gives us the unbiased estimator of \( \prod_{i=1}^{m} P(z_i; x) \). To do so, we first note a simple proposition about complex numbers.

Proposition A.1. \([\text{Nar20}]\) Let \( z \) be a complex number with \(|z| = 1\) and \(|\arg z| \leq \theta\). Then, for any \( 0 < p < 1 \), \(|\frac{z^{(1-p)} \theta}{p} \leq 1 + \frac{\theta^2}{p}\).

Proof of Lemma 3.1. For some \( 1 \leq k \leq m \), fix some complex numbers \( w_1, \ldots, w_k \) and consider the random variable

\[
    f(\tilde{x}, w) := \sum_{1 \leq i_1 < i_2 < \cdots < i_k \leq n} \tilde{x}_{i_1} \cdots \tilde{x}_{i_k} w_1^{i_1} w_2^{i_2-1} \cdots w_k^{i_k-1}
\]

for \( w = (w_1, \ldots, w_k) \), which is a random variable since \( \tilde{x} \) is random.

We first describe \( E[f(\tilde{x}, w)] \) and choose appropriate values for \( w_1, \ldots, w_k \). First, we can rewrite

\[
    f(\tilde{x}, w) = \sum_{i_1, \ldots, i_k \geq 1} \tilde{x}_{i_1} \tilde{x}_{i_1+i_2} \cdots \tilde{x}_{i_1+i_2+\cdots+i_k} w_1^{i_1} w_2^{i_2} \cdots w_k^{i_k}.
\]

For any \( j_1, \ldots, j_k \), note that \( \tilde{x}_{i_1} \) coming from \( x_{j_1} \), \( \tilde{x}_{i_1+i_2} \) coming from \( x_{j_1+j_2} \), etc. means that \( j_1 \geq i_1, j_2 \geq i_2, \ldots, j_k \geq i_k \). Moreover, even in this case, this will only happen with probability

\[
    \prod_{r=1}^{k} \left( p \cdot \binom{j_r - 1}{i_r - 1} q^{j_r - i_r} \right) = \prod_{r=1}^{k} \binom{j_r - 1}{i_r - 1}.
\]

17
Therefore, we have that

\[
\mathbb{E}[f(\tilde{x}, w)] = \sum_{i_1, \ldots, i_k \geq 1 \atop j_1 + \cdots + j_k \leq n} \prod_{r=1}^{k} \left( \frac{(j_{r} - 1)}{(i_{r} - 1)} \right) p^{i_r} q^{j_r - i_r} x_{j_1 + \cdots + j_r} w_{i_r}^{i_r}
\]

\[
= \sum_{j_1, \ldots, j_k \geq 1 \atop j_1 + \cdots + j_k \leq n} \prod_{r=1}^{k} (pw_{i_r} x_{j_1 + \cdots + j_r} \cdot \sum_{i_r=1}^{j_r - 1} (j_r - 1) p^{i_r - 1} q^{j_r - i_r} w_{i_r}^{i_r - 1})
\]

\[
= \sum_{j_1, \ldots, j_k \geq 1 \atop j_1 + \cdots + j_k \leq n} \prod_{r=1}^{k} (pw_{i_r} x_{j_1 + \cdots + j_r} \cdot (pw_{i_r} + q)^{j_r - 1})
\]

\[
= p^k \frac{w_1 \cdots w_k}{(pw_1 + q) \cdots (pw_k + q)} \sum_{j_1, \ldots, j_k \geq 1 \atop j_1 + \cdots + j_k \leq n} x_{j_1} \cdots x_{j_1 + \cdots + j_k} (pw_1 + q)^{j_1} \cdots (pw_k + q)^{j_k}.
\]

Now, fix \(k \leq m\) and fix a sequence \(B = (B_1, \ldots, B_k)\) of strictly nested nonempty subsets of \([m]\) with \(B_1 = [m]\). By this, we mean that \([m] = B_1 \supseteq B_2 \supseteq \cdots \supseteq B_k = \emptyset\). Now, for \(1 \leq r \leq k\), define \(w_{B,r} := \frac{1}{p}((\prod_{i \in B_r} z_i) - q)\), \(w_B := (w_{B,1}, \ldots, w_{B,k})\), and \(C_r := B_r \setminus B_{r+1}\) for \(1 \leq r \leq k - 1\) and \(C_k := B_k\). Finally, for any set \(S \subset [m]\), define \(z_S := \prod_{i \in S} z_i\). Then,

\[
\mathbb{E}[f(\tilde{x}, w_B)] = p^k \cdot \frac{w_{B,1} \cdots w_{B,k}}{z_{B_1} z_{B_2} \cdots z_{B_k}} \cdot \sum_{j_1, \ldots, j_k \geq 1 \atop j_1 + \cdots + j_k \leq n} x_{i_1} \cdots x_{i_1 + \cdots + i_k} \frac{z_{C_1} \cdots z_{C_k}}{z_{B_1} z_{B_2} \cdots z_{B_k}} \cdot \mathbb{E}[f(\tilde{x}, w_B)]
\]

where we have written \(i_r = j_1 + j_2 + \cdots + j_r\) for all \(1 \leq r \leq k\). This implies that

\[
\prod_{k=1}^{m} \left( \sum_{i=1}^{n} x_i z_i^i \right) = \sum_{1 \leq i_1 < i_2 < \cdots < i_k \leq n}^{[m] = B_1 \supseteq \cdots \supseteq B_k \neq \emptyset} x_{i_1} \cdots x_{i_k} z_{C_1} \cdots z_{C_k} \cdot \mathbb{E}[f(\tilde{x}, w_B)]
\]

The first line is true by expanding and using the fact that \(x_i = x_i^{b_i}\) for all \(b_i \in \mathbb{N}\), as \(x \in \{0, 1\}\).

Now, let

\[
g_m(\tilde{x}, Z) := \sum_{1 \leq k \leq m}^{B=(B_1, \ldots, B_k)} p^{-k} \cdot \frac{z_{B_1} \cdots z_{B_k}}{w_{B,1} \cdots w_{B,k}} \cdot f(\tilde{x}, w_B).
\]

For fixed \(p, q, n\), note that \(g_m(\tilde{x}, Z)\) is indeed only a function of \(\tilde{x}, Z\), and \(m\), as the \(w_{B,r}\)'s are determined given \(Z\). Importantly, there is no dependence of \(g\) on \(x\). Then,

\[
\mathbb{E}[g_m(\tilde{x}, Z)] = \prod_{k=1}^{m} \left( \sum_{i=1}^{n} x_i^i \right).
\]
Finally, we provide bounds on \( f(\bar{x}, w_B) \) that will give us our bounds on \( g_m(\bar{x}, Z) \). If \( |z_i| = 1 \) and \( |\arg z_i| \leq \frac{1}{6} \) for all \( i \), then for \( B = (B_1, \ldots, B_r) \), \( z_{B_r} = \prod_{i \in B_r} z_i \) has magnitude 1 and argument at most \( \frac{r}{6} \) in absolute value. Therefore, \( |w_{B,r}| = \frac{1}{p} \left( (\prod_{i \in B_r} z_i) - q \right) \leq 1 + O \left( \frac{m^2}{p^2 L^2} \right) = \exp \left( O \left( \frac{m^2}{p^2 L^2} \right) \right) \), by Proposition A.1. This means that for any \( i_1 + \cdots + i_k \leq n \),

\[
|w_{B,1}^{i_1} w_{B,2}^{i_2} \cdots w_{B,k}^{i_k}| \leq \exp \left( O \left( \frac{m^2}{p^2 L^2} \cdot n \right) \right).
\]

Since \( f(\bar{x}, w_B) \) is the sum of \( w_{B,1}^{i_1} w_{B,2}^{i_2} \cdots w_{B,k}^{i_k} \) over all \( i_1, \ldots, i_k \geq 1 \) with \( i_1 + \cdots + i_k \leq n \), there are at most \( n^k \) choices of \( i_1, \ldots, i_k \), so we have that

\[
|f(\bar{x}, w_B)| \leq \exp \left( O \left( \frac{m^2}{p^2 L^2} \cdot n \right) \right) \cdot n^k.
\]

Therefore,

\[
|g_m(\bar{x}, Z)| \leq \sum_{1 \leq k \leq m} p^{-k} \cdot |w_{B,1}^{i_1} \cdots w_{B,k}^{i_k}| \cdot |f(\bar{x}, w_B)| \leq \sum_{1 \leq k \leq m} p^{-k} \cdot \exp \left( O \left( \frac{m^2 n}{p^2 L^2} \right) \right) \cdot n^k
\]

\[
\leq (p^{-1} nm)^{O(m)} \cdot \exp \left( O \left( \frac{m^2 n}{p^2 L^2} \right) \right).
\]

The final equation follows from the observation that the number of sequences \( (B_1, \ldots, B_k) \) is at most \( m^m \), since each \( i \in [m] \) has some final subset \( j \) such that \( i \in B_j \) but \( i \notin B_{j+1} \). \( \square \)

### A.3 Proof of Theorem 1.3

We let \( \omega_k \) denote \( e^{2\pi i/k} \) for \( k \geq 1 \). When dealing with a string of length \( n \), we write \( \omega := \omega_n \).

In the case where \( n \) is prime, we already proved it in Proposition 3.3.

Next, we prove it in the case that \( n = p \cdot q \) for \( p, q \) odd primes. We will first need a simple lemma, which is likely folklore, though we give a proof regardless.

**Lemma A.2.** Let \( p, q \) be distinct primes, and suppose that \( (1 - \omega_p)|\sum_{i=0}^{q-1} b_i \omega_q^i \) in \( \mathbb{Q}[\omega_{pq}] \). Then, we in fact have that \( p|\sum_{i=0}^{q-1} b_i \omega_q^i \) in \( \mathbb{Q}[\omega_{pq}] \).

**Proof.** Note that \( (1 - \omega_p)|\sum_{i=0}^{q-1} b_i \omega_q^i \) implies that \( (1 - \omega_p)^p|\sum_{i=0}^{q-1} b_i \omega_q^i \)\(^p \). But \( p|1 - \omega_p \)\(^p \), so \( p|\sum_{i=0}^{q-1} b_i \omega_q^i \)\(^p \). Now, using Frobenius Endomorphism, we have that \( \sum_{i=0}^{q-1} b_i \omega_q^i \equiv \sum_{i=0}^{q-1} b_i \omega_q^{i \cdot p} \) (mod \( p \)), so \( p|\sum_{i=0}^{q-1} b_i \omega_q^i \)\(^p \). But since \( p \neq q \), we have that \( \omega_q^p \) and \( \omega_q \) are Galois conjugates, so we therefore have that \( p|\sum_{i=0}^{q-1} b_i \omega_q^i \), as desired. \( \square \)

**Lemma A.3.** Theorem 1.3 is true when \( n = p \cdot q \), where \( p, q \) are distinct odd primes.

**Proof.** First, we have that \( \sum_{i=0}^{n-1} a_i = \omega^a \sum_{i=0}^{n-1} b_i \) and since \( a_1, \ldots, a_n, b_1, \ldots, b_n \in \{0,1\} \), this means that \( \omega^a \) is real and thus equals 1. So, \( \sum_{i=0}^{n-1} a_i = \sum_{i=0}^{n-1} b_i \). Next, we have that \( \sum_{i=0}^{n-1} a_i \omega_p^i = \omega^a \cdot \sum_{i=0}^{n-1} b_i \omega_p^i \), since \( \omega_p = \omega^a \). Therefore, since the \( a_i \)'s are all integers, this implies that either

19
\[\sum_{i=0}^{n-1} a_i \omega_p^i = \sum_{i=0}^{n-1} b_i \omega_p^i = 0 \text{ or } \omega^e = \frac{\sum a_i \omega_p^i}{\sum b_i \omega_p^i} \in \mathbb{Q}[\omega_p].\] Thus, by Theorem 2.2, \(\omega^e\) actually equals \(\omega_p^k\) for some \(k\). Likewise, \(\omega^{e_p}\) actually equals \(\omega_q^\ell\) for some \(\ell\), so we have that

\[\sum_{i=0}^{n-1} a_i \omega_p^i = \omega_p^k \sum_{i=0}^{n-1} b_i \omega_p^i, \quad \sum_{i=0}^{n-1} a_i \omega_q^i = \omega_q^\ell \sum_{i=0}^{n-1} b_i \omega_q^i.\]

Therefore, by the Chinese Remainder theorem, we can cyclically shift \(\{b_i\}\) by something that is \(k\) modulo \(p\) and \(\ell\) modulo \(q\) to get some sequence \(\{b'_i\}\) so that

\[\sum_{i=0}^{n-1} a_i = \sum_{i=0}^{n-1} b'_i, \quad \sum_{i=0}^{n-1} a_i \omega_p^i = \sum_{i=0}^{n-1} b'_i \omega_p^i, \quad \text{and} \quad \sum_{i=0}^{n-1} a_i \omega_q^i = \sum_{i=0}^{n-1} b'_i \omega_q^i.\]

Without loss of generality, we can therefore pretend that \(k = \ell = 0\), so in fact we have \(b'_i = b_i\) for all \(i\). Now, suppose that \(\sum a_i \omega_i^j = \omega^m \cdot \sum b_i \omega_i^j\). Our goal is to show that \(p|m\) and \(q|m\), so that \(\omega^m = 1\). Assume the contrary, WLOG that \(q \nmid m\). Then, we can write

\[\sum_{i=0}^{n-1} (a_i - b_i) \omega^j = (\omega^m - 1) \cdot \sum_{i=0}^{n-1} b_i \omega^j.\] (1)

Now, choose integers \(r, s\) so that \(r \cdot q + 1 = s \cdot p\). Then, we have that \(\omega_q^{i+s} - \omega^i = \omega_q^{i+s-p} - \omega^i = \omega^j (\omega_p^{i-q} - 1) = \omega^j (\omega_p^{i-r} - 1)\), which is a multiple of \(\omega_p - 1\). Therefore, we have that \(1 - \omega_p\) divides

\[\sum_{i=0}^{n-1} (a_i - b_i) \cdot (\omega^i - \omega_q^{i+s}) = \sum_{i=0}^{n-1} (a_i - b_i) \omega^i - \sum_{i=0}^{n-1} (a_i - b_i) \omega_q^{i+s} = \sum_{i=0}^{n-1} (a_i - b_i) \omega^i.\]

The last equality in the above line follows since \(\sum_{i=0}^{n-1} a_i \omega_q^i = \sum_{i=0}^{n-1} b_i \omega_q^i\), and since \(s\) is relatively prime to \(q\), this means \(\omega_q^s\) is a Galois conjugate of \(\omega_q\), so \(\sum_{i=0}^{n-1} a_i \omega_q^{i+s} = \sum_{i=0}^{n-1} b_i \omega_q^{i+s}\).

Now, since \(q \nmid m\), we have that either \(\omega^m - 1\) is a unit in \(\mathbb{Z}[\omega]\) (if \(p \nmid m\)) or \(\omega^m - 1 \equiv q (\text{if } p|m)\). Therefore, by Equation (1), we have that

\[(1 - \omega_p) q \cdot \sum_{i=0}^{n-1} b_i \omega^i \Rightarrow (1 - \omega_p) \sum_{i=0}^{n-1} b_i \omega^i,\]

since \((1 - \omega_p), (q)\) are relatively prime as ideals. Now, recalling that \(\omega^i \equiv \omega_q^{i+s} \pmod{1 - \omega_p}\), we have that \((1 - \omega_p))|\sum_{i=0}^{n-1} b_i \omega_i^{i+s}\).

By Lemma A.2, we have that \(p|\sum_{i=0}^{n-1} b_i \omega_i^{i+s}\). Since \(\omega_q^s\) and \(\omega_q\) are Galois conjugates, this also means that \(p|\sum_{i=0}^{n-1} b_i \omega_i^{i+s}\). Now, for \(0 \leq j \leq q - 1\), let \(d_j = b_j + b_{j+q} + \cdots + b_{j+(p-1)q}\).

We have that \(p|\sum_{j=0}^{q-1} d_j \omega_j^q\), so \(\sum_{j=0}^{q-1} d_j \omega_j^q\) is an algebraic integer in \(\mathbb{Q}[\omega_q]\). Therefore, \(d_0 \equiv d_1 \equiv \cdots \equiv d_{q-1} \pmod{p}\). Since \(0 \leq d_i \leq p\) for all \(i\), we either have that \(d_0 = d_1 = \cdots = d_{q-1}\), or \(d_0, d_1, \ldots, d_{q-1} \in \{0, p\}\).

Likewise, we also have that \(\sum b_i \omega^i = \omega^m \cdot \sum a_i \omega^i\), where \(q \nmid (-m)\). Therefore, if \(c_j = a_j + a_{j+q} + \cdots + a_{j+(p-1)q}\) for each \(0 \leq j \leq q - 1\), we either have that \(c_0 = c_1 = \cdots = c_{q-1}\), or \(c_0, c_1, \ldots, c_{q-1} \in \{0, p\}\).
Now, suppose that $d_0, d_1, \ldots, d_{q-1} \in \{0, p\}$. This means that for all $0 \leq j \leq q - 1$, $b_j = b_{j+q} = \cdots = b_{j+(p-1)q}$, so $b_j \omega^i + b_{j+q} \omega^{i+q} + \cdots + b_{j+(p-1)q} \omega^{i+(p-1)q} = 0$ for all $0 \leq j \leq p - 1$. Importantly, this means $\sum b_j \omega^i = 0$. But since $\sum a_i \omega^i = \omega^n \cdot \sum b_i \omega^i$ for some $m \in \mathbb{Z}$, this also means that $\sum a_i \omega^i = 0$, so in fact we do have that $\sum b_j \omega^i = \sum a_i \omega^i$. Likewise, if $a_0, a_1, \ldots, a_{q-1} \in \{0, p\}$ we also have that $\sum b_j \omega^i = \sum a_i \omega^i = 0$ by a symmetric argument.

Otherwise, we are dealing with the case where $d_0 = d_1 = \cdots = d_{q-1}$ and $c_0 = c_1 = \cdots = c_{q-1}$. But then, $0 = \sum_{j=0}^{q-1} d_j \omega^i_q = \sum_{i=0}^{n-1} b_i \omega^i_q$ and $0 = \sum_{j=0}^{q-1} c_j \omega^i = \sum_{i=0}^{n-1} a_i \omega^i_q$. Recall that $\sum a_i \omega^i = \omega^n \cdot \sum b_i \omega^i$ and that we assumed $q \nmid m$. If $p \nmid m$, then if $m = p \cdot t$, we have $\sum a_i \omega^i = \sum b_i \omega^{i \mod t}$, where $i \equiv t \mod n$ is done modulo $n$. Moreover, we have that $\sum b_i \omega^i = \sum b_i \omega^{i \mod t} = \sum b_i \omega^i_q$, $\sum b_i \omega^i = \sum b_i \omega^{i \mod t} = \sum b_i \omega^i_q$, $\sum a_i \omega^i = \sum b_i \omega^i_q$, and $0 = \sum \sum a_i \omega^i = \sum b_i \omega^i_q$. Thus, the same argument applied to the last two equations implies that $0 = \sum_{i=0}^{n-1} a_i \omega^i_q = \sum_{i=0}^{n-1} b_i \omega^i_q$. As a result, we can shift the sequence by $m$, since $\sum a_i \omega^i = \sum b_i \omega^i$, but we will still have that $\sum a_i = \sum b_i$, $\sum a_i \omega^i = \sum b_i \omega^i_q$, $\sum b_i \omega^i = 0$, and $\sum a_i \omega^i = \sum b_i \omega^i_q = \sum b_i \omega^i_q$.

We now show that Theorem 1.3 is true when $n$ is the square of an odd prime.

**Proposition A.4.** Theorem 1.3 is true if when $n = p^2$ is the square of an odd prime.

**Proof.** By shifting, we may without loss of generality assume that $\sum a_i \omega^i = \sum b_i \omega^i$, so $P(x) = \sum (a_i - b_i) x^i$ has $\omega$ as a root. Thus, $1 + x^p + \cdots + x^{n-p} \mid P(x)$, which means that $a_i - b_i = a_{i+p} - b_{i+p} = \cdots = a_{i+n-p} - b_{i+n-p}$, where indices are taken mod $n$. Thus, if it is not the case that $a_i = b_{i+p} = \cdots = b_{i+n-p}$, equivalently that $b_i = b_{i+p} = \cdots = b_{i+n-p}$, then we must have that $a_i = b_i$, $a_{i+p} = b_{i+p}$, $\ldots$, $a_{i+n-p} = b_{i+n-p}$ since $a_j, b_j \in \{0, 1\}$.

Let $z = \omega^p$. We have that

$$\sum a_i z^i = (a_0 + a_p + \cdots + a_{n-p}) + (a_1 + a_{p+1} + \cdots + a_{n-p-1}) z + \cdots + (a_{p-1} + a_{2p-1} + \cdots + a_{n-1}) z^{p-1},$$

$$\sum b_i z^i = (b_0 + b_p + \cdots + b_{n-p}) + (b_1 + b_{p+1} + \cdots + b_{n-p-1}) z + \cdots + (b_{p-1} + b_{2p-1} + \cdots + b_{n-1}) z^{p-1}.$$

Thus, $\frac{\sum a_i z^i}{\sum b_i z^i} \in \mathbb{Q}[z]$, so $p \mid c_p$ and we have that $\sum a_i z^i = z^m \sum b_i z^i$ for some $m \in \mathbb{Z}$. It follows that $\{a_i + a_{i+p} + \cdots + a_{i+n-p}\}$ and $\{b_i + b_{i+p} + \cdots + b_{i+n-p}\}$ are cyclic shifts of each other. Since these sequences are of length $p$, this means that they are equal. We already know that $a_i + a_{i+p} + \cdots + a_{i+n-p} \not\in \{0, p\}$ implies $a_i \neq a_{i+p}$ for all $\ell$. But it is also the case that $a_i + a_{i+p} + \cdots + a_{i+n-p} = b_i + b_{i+p} + \cdots + b_{i+n-p} \in \{0, p\}$ implies $a_i \neq a_{i+p}$ for all $\ell$ since $a_j, b_j \in \{0, 1\}$. Thus, we have shown that $a_i = b_i$ for all $i$, so we are done.

**Proposition A.5.** Theorem 1.3 is true if $n = 2p$, i.e., $n$ is twice a prime.
Proof. If \( p = 2 \), i.e. \( n = 4 \), then if \( \sum a_i = \sum b_i \) but the sequences \( \{a_i\} \) and \( \{b_i\} \) are not equal up to a cyclic rotation, then up to cyclic rotations, we either have \( \{a_i\} = \{1, 0, 1, 0\} \) and \( \{b_i\} = \{1, 1, 0, 0\} \) or vice versa. But then, \( \sum a_i(-1)^i = \pm 2 \) and \( \sum b_i(-1)^i = 0 \).

If \( p \) is an odd prime, then note that the minimal polynomial of \( \omega = \omega_n \) is \( 1 + x^2 + \cdots + x^{2(p-1)} \). Now, suppose that \( a, b \) are rotated so that if \( \omega \) and \( \omega^k \) or vice versa. But then, \( \sum x(i | R) = 0 \). Since \( a_i - b_i \in \{-1, 0, 1\} \) for all \( i \), we must have that \( \sum_{i=0}^{2p-1} (a_i - b_i)x^i = (1 + x^2 + \cdots + x^{2(p-1)}) \cdot R(x) \), where \( R(x) \) must be either 0, \( \pm 1 \), and \( \pm x \pm 1 \). However, since \( \sum a_i = \sum b_i \), we have that \( P(1) - Q(1) = 0 \neq (p - 1) \cdot R(1) \), so \( R(1) = 0 \). Thus, \( R(x) \) must equal either 0, \( x - 1 \), or \( 1 - x \).

If \( R(x) = x - 1 \), then \( a_i - b_i = 1 \) for all odd \( i \) and \( -1 \) for all even \( i \), which means that \( a_i = 1 \) if and only if \( i \) is odd, but \( b_i = 0 \) if and only if \( i \) is even. Since \( n \) is even, this means that \( \{a_i\} \) and \( \{b_i\} \) are the same sequence, up to a rotation by 1. The same is true if \( R(x) = 1 - x \) by symmetry between \( a \) and \( b \). Finally, if \( R(x) = 0 \), then \( a_i - b_i = 0 \) for all \( i \), so \( a_i = b_i \) for all \( i \), and thus the sequences \( \{a_i\} \) and \( \{b_i\} \) are the same. \( \square \)

Finally, we remark that the statement is false for numbers with 3 or more prime factors, which concludes the proof of Theorem 1.3. Suppose that \( n = abc \) with \( a, b, c > 1 \). Let \( A = \{1, a+1, \ldots, ab - a + 1, a, ab + a, \ldots, abc - ab + a\} \) and \( B = \{1, a+1, \ldots, ab - a + 1, 0, ab, \ldots, abc - ab\} \). Consider circular strings \( a \) and \( b \) of length \( n \) with 1s in positions given by \( A \) and \( B \), respectively. Let \( P(x) = \sum_{i \in A} x^i \) and \( Q(x) = \sum_{i \in B} x^i \). We have that \( P(x) - Q(x) = (x^a - 1) \cdot \frac{x^b - 1}{x - 1} \) and \( P(x) - x^a Q(x) = x(1 - x^b) \). Thus, for all \( k \), \( \frac{P(\omega^k)}{Q(\omega^k)} \) is a power of \( \omega \), so the conditions of Theorem 1.3 hold. However, \( a \) and \( b \) are not cyclic shifts of each other.