Math. Stat. Learn. 2 (2019), 1–35
DOI 10.4171/MSL/xxx

Mathematical Statistics and Learning

Subpolynomial trace reconstruction for random strings and arbitrary deletion probability

Nina Holden, Robin Pemantle, Yuval Peres and Alex Zhai

Abstract. The insertion-deletion channel takes as input a bit string $\mathbf{x} \in \{0,1\}^n$, and outputs a string where bits have been deleted and inserted independently at random. The trace reconstruction problem is to recover $\mathbf{x}$ from many independent outputs (called "traces") of the insertion-deletion channel applied to $\mathbf{x}$. We show that if $\mathbf{x}$ is chosen uniformly at random, then $\exp(O(\log^{1/3} n))$ traces suffice to reconstruct $\mathbf{x}$ with high probability. For the deletion channel with deletion probability $q < 1/2$ the earlier upper bound was $\exp(O(\log^{1/2} n))$. The case of $q \geq 1/2$ or the case where insertions are allowed has not been previously analyzed, and therefore the earlier upper bound was as for worst-case strings, i.e., $\exp(O(n^{1/3}))$. We also show that our reconstruction algorithm runs in $n^{1+o(1)}$ time.

A key ingredient in our proof is a delicate two-step alignment procedure where we estimate the location in each trace corresponding to a given bit of $\mathbf{x}$. The alignment is done by viewing the strings as random walks and comparing the increments in the walk associated with the input string and the trace, respectively.

Mathematics Subject Classification (2010). 68Q17, 68Q25, 68Q87, 68W32.

Keywords. Trace reconstruction, deletion channel.

1. Introduction

Learning a parameter from a sequence of noisy observations is a basic problem in statistical inference and machine learning. The amount of data required (known as the *sample complexity*) to learn the parameter is of fundamental interest. A natural problem in this class where the missing parameter is a bit string and it is unknown whether the sample complexity is polynomial, is the trace reconstruction problem for the *insertion-deletion channel*. This channel takes as input a string $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1}) \in \{0,1\}^n$ and outputs a noisy version of it, where bits have been randomly inserted and deleted. Let $q \in [0,1)$ be the deletion probability and let $q' \in [0,1)$ be the insertion probability. First, for each j, before the jth bit of $\mathbf{x}$ we insert $G_j - 1$ uniform and independent bits, where the independent geometric random variables $G_j \geq 1$ have parameter $1 - q'$. Then we delete each bit of the resulting

string independently with probability q. The output string $\widetilde{\mathbf{x}}$ is called a *trace*. An example is shown in Figure 1.

$$\underset{\mathbf{x}}{1\,0\,0\,1\,1\,1\,0} \to 1\,1\,0\,0\,1\,0\,1\,0\,1\,1\,1\,1\,0\,0 \to 1\,1\,0\,0\,1\,0\,1\,0\,1\,1\,1\,1\,0\,0 \to \underset{\widetilde{\mathbf{x}}}{1\,1\,1\,0\,0\,1\,0}$$

Figure 1. We obtain a trace $\widetilde{\mathbf{x}}$ by sending $\mathbf{x}$ through the insertion-deletion channel. Inserted bits are shown in green and deleted bits are shown in red.

Suppose that the input string $\mathbf{x}$ is unknown. The *trace reconstruction problem* asks the following: How many i.i.d. copies of the trace $\widetilde{\mathbf{x}}$ do we need in order to determine $\mathbf{x}$ with high probability? (A more formal problem description will be given in Section 3.)

There are two variants of this problem: the "worst case" and the "average case" (also referred to as the "random case"). In the worst case variant, we want to obtain bounds which hold uniformly over all possible input strings $\mathbf{x}$. In the average case variant, the input string is chosen uniformly at random.

In this paper, we study the average case. Holenstein, Mitzenmacher, Panigrahy, and Wieder [7] gave an algorithm for reconstructing random strings from the deletion channel using polynomially many traces, assuming the deletion probability q is sufficiently small. Peres and Zhai [15] proved that $\exp(O(\log^{1/2} n))$ many traces suffice for the deletion channel when the deletion probability q is below $1/2$. For $q \geq 1/2$, the previous best bound was the same as for worst case strings, i.e., $\exp(O(n^{1/3}))$, see [4, 14].

The following theorem is our main result, which improves the upper bound for all $q \in [0, 1)$ and also holds when we allow insertions. In particular, we answer the part of the first open question in [13, Section 9] which concerns random strings.

Theorem 1. *For $n \in \mathbb{N}$ let $\mathbf{x} \in \{0,1\}^n$ be a bit string where the bits are chosen uniformly and independently at random. Given $q, q' \in [0, 1)$ there exists $M > 0$ such that for all n we can reconstruct $\mathbf{x}$ with probability $1 - o_n(1)$ using $\lceil \exp(M \log^{1/3} n) \rceil$ traces from the insertion-deletion channel with parameters q and q'. Moreover, our algorithm runs in $n^{1+o(1)}$ time.*

An earlier version of this work appeared in COLT [6]; in this updated version, we have simplified the proof and added an analysis of the algorithm's running time.

We remark that the trace reconstruction problem is significantly more difficult for $q > 1/2$ [1] and that the alignment algorithm used by Peres and Zhai fails fundamentally

[1] Suppose that $q > 1/2$, the string $\mathbf{w}$ is an arbitrary string of length $(1-q)n$, and $\mathbf{x}$ is a random string of length n. Then it holds with probability at least $1 - \exp(-cn)$ that $\mathbf{w}$ is a subsequence of $\mathbf{x}$. To see this, observe that the number of bits in $\mathbf{x}$ until we see w_0 is a geometric random variable of mean 2. Iterating, existence (with probability $1 - \exp(-cn)$) of an appropriate subsequence holds by concentration for the sum of independent geometric random variables of mean 2. Therefore, by a union bound, if $q > 1/2$,

in this case. Moreover, the upper bound $\exp(O(\log^{1/3} n))$ in Theorem 1 is the best one can obtain without also improving the upper bound $\exp(O(n^{1/3}))$ for worst case strings. Indeed, given an arbitrary string of length $m = \log_{2+\varepsilon} n$ for $\varepsilon > 0$, this string will appear in a random length n string with probability converging to 1 as $n \to \infty$. In particular, a given worst case string of length m is likely to appear in our random string, and the best known algorithm for reconstructing this string requires $\exp(\Omega(m^{1/3})) = \exp(\Omega(\log^{1/3} n))$ traces. See Lemma 10 in [12] for the details of this reduction.

We note also that our methods can be adapted easily to certain other reconstruction problems, e.g., to the case where one allows substitutions in addition to deletions and insertions, and the case where the bits in the input $\mathbf{x}$ are independent Bernoulli(r) random variables for arbitrary $r \in (0, 1)$, instead of $r = 1/2$. There is also a simple reduction (described in e.g. [12] and [4]) of the trace reconstruction problem for larger alphabets to the case of bits. Moreover, as shown in [12], trace reconstruction becomes much easier if the alphabet size grows as $\Omega(\log n)$.

In Section 2 we present some background and literature on the trace reconstruction problem, before we give a precise definition of the trace reconstruction problem in Section 3. We give an outline of the proof of Theorem 1 in Section 4 and we present some notation in Section 5. In Sections 6 to 9 we prove the various ingredients which are needed for the proof of Theorem 1, and in Section 10 we conclude the proof.

2. Related work

The trace reconstruction problem dates back to the early 2000's [1, 10, 11]. Batu, Kannan, Khanna, and McGregor, who were partially motivated by the study of genetic mutations, considered the case where the deletion probability q is decreasing in n. They proved that if the original string $\mathbf{x}$ is random and the deletion probability $q = O(1/\log n)$, then $\mathbf{x}$ can be constructed with high probability using $O(\log n)$ samples. Furthermore, they proved that if $q = O(n^{-(1/2+\varepsilon)})$, then every string $\mathbf{x}$ can be reconstructed with high probability with $O(n \log n)$ samples.

Holenstein, Mitzenmacher, Panigrahy, and Wieder [7] considered the case of random strings and constant deletion probability. They gave an algorithm for reconstruction with polynomially many traces when the deletion probability q is less than some small threshold c. The threshold c is not given explicitly in the work of [7], but was estimated in [15] to be at most 0.07.

The result of [7] was improved by [15]. They showed that a subpolynomial number of traces $\exp(O(\log^{1/2} n))$ is sufficient for reconstruction, and they extended the range of allowed q to the interval $[0, 1/2)$.

---

then any subexponential collection of strings of length $(1-q)n$ (typical for traces of the deletion channel) are, with high probability, all substrings of a random string $\mathbf{x}$ of length n.

Our work improves the above results in three ways. First, we improve the upper bound to $\exp(O(\log^{1/3} n))$. Second, we allow for any deletion and insertion probabilities in $[0, 1)$. Third, unlike [15], our method works not only for the deletion channel, but also for the case where we allow insertions and substitutions.

It is shown by [7] that $\exp(O(n^{1/2} \log n))$ traces suffice for reconstruction with high probability with worst case input. This was improved to $\exp(O(n^{1/3}))$ independently by De, O'Donnell, and Servedio [4] and by Nazarov and Peres [14]. Until the current work, the average case upper bound was equal to the worst case upper bound for $q \geq 1/2$. The techniques developed by [4, 14] are applied in the current work and the work of [15] to certain shorter substrings of our random string.

A lower bound of $\Omega(\log^2 n)$ was obtained in the average case in [12], and a lower bound of $\Omega(n)$ was obtained in the worst case in [1]. These bounds were improved to $\Omega(\log^{9/4} n/\sqrt{\log\log n})$ and $\Omega(n^{5/4}/\sqrt{\log n})$, respectively, by Holden and Lyons [5], and further to $\Omega(\log^{5/2} n/(\log\log n)^7)$ and $\Omega(n^{3/2}/\log^7 n)$ by Chase [3]. Trace reconstruction for the setting which allows insertions and substitution in addition to deletions was considered in [4, 8, 14, 16]. We refer to the introduction of [4] and the survey [13] for further background on the deletion channel.

3. The trace reconstruction problem

To simplify notation, throughout the paper we will implicitly pad any finite-length bit strings with infinitely many zeroes on the right. Thus, expressions such as $\mathbf{x}(i : j)$ are well-defined for $i < j$ even if j is larger than the length of $\mathbf{x}$. Let $\mathbb{N} = \{0, 1, \dots\}$, and let $\mathcal{S} := \{0, 1\}^{\mathbb{N}}$ denote the space of infinite sequences of zeroes and ones. We denote elements of $\mathcal{S}$ by $\mathbf{x} := (x_0, x_1, \dots)$. If $I \subset \mathbb{R}$ we will sometimes write $\mathbf{x}(I)$ instead of $\mathbf{x}(I \cap \mathbb{N})$, and we use a similar convention for functions which are defined on (subsets of) $\mathbb{N}$.

Fix a deletion probability q and an insertion probability q' in $[0, 1)$, and let $p = 1-q$ and $p' = 1-q'$. We construct $\widetilde{\mathbf{x}}$ from $\mathbf{x}$ by the procedure described above, i.e., first, for each $j \in \mathbb{N}$ we insert $G_j - 1$ uniform and independent bits before the jth bit of $\mathbf{x}$. The geometric random variables G_j are independent and satisfy

$$\mathbb{P}[G_j = v] = (q')^{v-1}(1 - q'), \quad \forall v \in \{1, 2, \dots\}.$$

Then we delete each bit of the resulting string independently with probability q.

Let μ be the law of i.i.d. Bernoulli random variables with parameter $1/2$. We write $\mathbb{P}_{\mathbf{x}} := \mathbb{P}_{\delta_{\mathbf{x}}}$ for the law of $\widetilde{\mathbf{x}}$ when $\mathbf{x}$ is fixed and write $\mathbb{P} := \mathbb{P}_\mu$ for the law of $\widetilde{\mathbf{x}}$ when $\mathbf{x}$ is picked according to μ. We call the string $\widetilde{\mathbf{x}}$ a trace. An example is given in Figure 1.

Worst case reconstruction problem. Let $q, q' \in [0, 1)$. For any $N \in \mathbb{N}$ let $\mathbb{P}_{\mathbf{x}}^N$ denote the probability measure associated with N independent outputs of the

insertion-deletion channel $\mathbb{P}_{\mathbf{x}}$ with deletion (resp. insertion) probability q (resp. q'). For $n \in \mathbb{N}$ and $\mathbf{x} \in \{0,1\}^n$ let $\mathfrak{X}$ denote a collection of $N_n \in \mathbb{N}$ traces sampled independently at random. We say that worst case strings of length n can be reconstructed with probability $1 - o_n(1)$ from N_n traces, if there is a function $G\colon \mathcal{S}^{N_n} \to \{0,1\}^n$, such that for all $\mathbf{x} \in \mathcal{S}$,

$$\mathbb{P}_{\mathbf{x}}^{N_n}[G(\mathfrak{X}) = \mathbf{x}(0 : n-1)] = 1 - o_n(1).$$

Average case reconstruction problem. Let μ_n denote uniform measure on $\{0,1\}^n$. We say that uniformly random strings of length n can be reconstructed with probability $1-o_n(1)$ from N_n traces if we can find a set $\mathcal{S}_n \subset \{0,1\}^n$ with $\mu_n(\mathcal{S}_n) = 1-o_n(1)$, and a function $G\colon \mathcal{S}^{N_n} \to \{0,1\}^n$, such that for all $\mathbf{x} \in \mathcal{S}$ for which $\mathbf{x}(0 : n-1) \in \mathcal{S}_n$, we have

$$\mathbb{P}_{\mathbf{x}}^{N_n}[G(\mathfrak{X}) = \mathbf{x}(0 : n-1)] = 1 - o_n(1).$$

In particular, Theorem 1 says that uniformly random strings can be reconstructed from $N_n := \lceil \exp(M \log^{1/3} n) \rceil$ traces with probability $1 - o_n(1)$.

4. Outline of proof

We give here an informal description of the algorithm used to achieve the bound in Theorem 1. The bits of $\mathbf{x}$ will be recovered one by one: for any $k, n \in \mathbb{N}$ with $k < n$ we assume $\mathbf{x}(0 : k)$ is already known, and we show that with probability $1 - O(n^{-2})$ we can use $\lceil \exp(M \log^{1/3} n) \rceil$ traces to determine the subsequent bit x_{k+1}. We will reuse these same traces for each step (i.e., for all values of k). Even with this reuse, by a union bound, the reconstruction will succeed at every step with high probability.

Three ingredients are required, as follows:

1. A Boolean test $T(\mathbf{w}, \widetilde{\mathbf{w}})$ on pairs $(\mathbf{w}, \widetilde{\mathbf{w}})$ of bit strings indicating whether $\widetilde{\mathbf{w}}$ is a plausible match for the string $\mathbf{w}$ sent through the insertion-deletion channel.

2. An alignment procedure that uses the test T repeatedly to produce for each of the independent traces $\widetilde{\mathbf{x}}$ an estimate τ for the position in $\widetilde{\mathbf{x}}$ corresponding to the k-th bit of $\mathbf{x}$.

3. A bit recovery procedure based on a method of [4, 14, 15] to produce from the approximately aligned traces an estimate of the subsequent bit or bits.

The argument of [15] follows the same overall structure, with an alignment step followed by a reconstruction step for each bit in the original string. However, the greedy alignment step of [15] relies crucially on the assumption that the deletion probability $q < 1/2$, and that no insertions are allowed. We overcome this problem by introducing a new kind of test for the alignment based on studying correlations between blocks in the input string and in the trace.

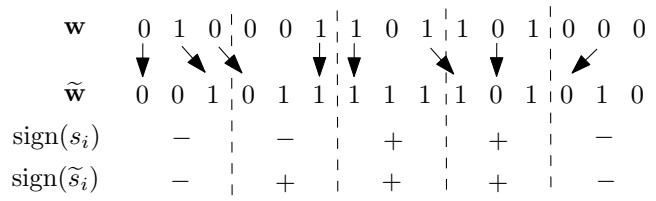

Figure 2. Illustration of the test $T_{\ell,\lambda}(\mathbf{w}, \widetilde{\mathbf{w}})$ with $\ell = 15$ and $\lambda = 3$. The arrows indicate that a bit in $\mathbf{w}$ was copied to a bit in $\widetilde{\mathbf{w}}$.

4.1. The test T. We describe here a simplified version of the test $T(\mathbf{w}, \widetilde{\mathbf{w}})$, which returns 1 if there is a likely match and 0 otherwise. Assume for simplicity that $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ have the same length and that $q = q'$, so the expected output length of the insertion-deletion channel is the same as the input length. The test involves two parameters: the length ℓ of the strings to test and another parameter $\lambda \leq \sqrt{\ell}$. We subdivide both $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ into roughly ℓ/λ blocks of size λ. We use $T_{\ell,\lambda}$ to denote the test using these parameters.

For each i, let s_i be the number of 1's minus the number of 0's in the i-th block of $\mathbf{w}$, and similarly define $\widetilde{s}_i$ for $\widetilde{\mathbf{w}}$. For some fixed constant $c > 0$ to be specified later, we declare that

$$T_{\ell,\lambda}(\mathbf{w}, \widetilde{\mathbf{w}}) = \begin{cases} 1 & \text{if } \displaystyle\sum_{i=1}^{\lceil \ell/\lambda \rceil} \operatorname{sign}(s_i) \cdot \operatorname{sign}(\widetilde{s}_i) > c \cdot \frac{\ell}{\lambda}, \\ 0 & \text{otherwise.} \end{cases}$$

See Figure 2 for an illustration. The idea here is that if the bits in the ith block of $\widetilde{\mathbf{w}}$ did not come from the ith block of $\mathbf{w}$, then they will be independent random bits, and so each $\operatorname{sign}(s_i) \cdot \operatorname{sign}(\widetilde{s}_i)$ will be 1 or -1 with equal probability (if we ignore the small probability event on which s_i or $\widetilde{s}_i$ is equal to 0, in which case we have $\operatorname{sign}(s_i) \cdot \operatorname{sign}(\widetilde{s}_i) = 0$). In this case, by Hoeffding's inequality, the test declares a match with probability $e^{-\Omega(\ell/\lambda)}$. Let us call this situation a "spurious match".

On the other hand, if one of the bits in the i-th block of $\widetilde{\mathbf{w}}$ was copied over from somewhere in the i-th block of $\mathbf{w}$ for some i, we consider the match to be a "true match". Let us describe one relatively likely way in which this could happen. Suppose that a positive fraction of the bits in the i-th block of $\widetilde{\mathbf{w}}$ came from the i-th block of $\mathbf{w}$. There is roughly a $e^{-O(\ell/\lambda^2)}$ chance that this will happen for all i (comparable to the probability that a simple random walk stays within $[-\lambda, \lambda]$ for ℓ steps). In this case, it is quite likely for a match to occur, because each $\operatorname{sign}(s_i)$ and $\operatorname{sign}(\widetilde{s}_i)$ will be positively correlated.

To summarize, the main feature of our test is that it has a true match rate of at least $e^{-O(\ell/\lambda^2)}$ and a spurious match rate of at most $e^{-\Omega(\ell/\lambda)}$, which is much lower than the true match rate as long as λ is large enough. In order to make these statements

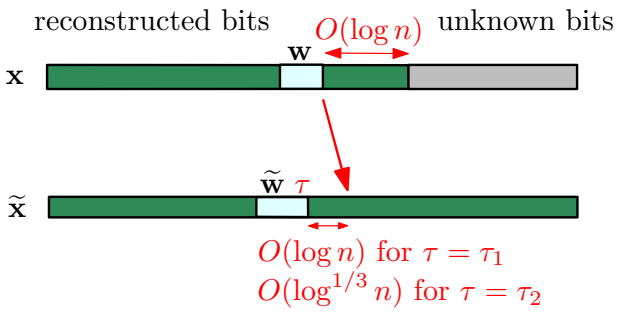

Figure 3. Illustration of the two alignment steps. In each step we fix a substring $\mathbf{w}$ of $\mathbf{x}$ which has distance $O(\log n)$ from the first unknown bit. Assuming the test gives a positive result with some substring $\widetilde{\mathbf{w}}$ of the trace, the alignment error (indicated by the lower horizontal arrow) is the difference between the right end-point τ of $\widetilde{\mathbf{w}}$ and the position in $\widetilde{\mathbf{x}}$ corresponding to the right end-point of $\mathbf{w}$ (indicated by the vertical red arrow).

rigorous, however, the real test we use (as well as the precise notions of "true" or "spurious" match) is slightly more complicated than what is described above. The test is defined formally in Section 6.

4.2. Alignment. We first remark that our alignment procedure will actually fail for most (all but about $e^{-O(\log^{1/3} n)}$) of the traces, and we will disregard these failed traces for purposes of reconstructing the current bit. Nevertheless, by taking enough total traces (more precisely, by choosing the constant M in Theorem 1 sufficiently large), we will still have a sufficient number of successful alignments to work with.

The alignment τ is computed in two steps. See Figure 3. We first compute a preliminary alignment position τ_1 in the output which corresponds to position $k_1 := k - C\log n$ in the input (where C is some large enough constant). This is done by performing the test $T_{\ell,\lambda}$ with $(\ell,\lambda) = (C\log^{5/3} n, C^{1/2}\log^{2/3} n)$. In particular, we declare τ_1 to be the first index in the output for which

$$T_{\ell,\lambda}(\mathbf{x}(k_1-\ell+1:k_1), \widetilde{\mathbf{x}}(\tau_1-\ell+1:\tau_1)) = 1.$$

Assuming that such a τ_1 exists, we claim that τ_1 is very likely to have alignment error of order at most $O(C\log n)$.

The probability of a spurious match is

$$e^{-\Omega(\ell/\lambda)} = e^{-\Omega(C^{1/2}\log n)} = n^{-\Omega(C^{1/2})},$$

which is negligible for large enough C. This probability is so small that even taking a union bound over all substrings of length ℓ, we are unlikely to find a single length-ℓ substring that produces spurious matches at a high rate.

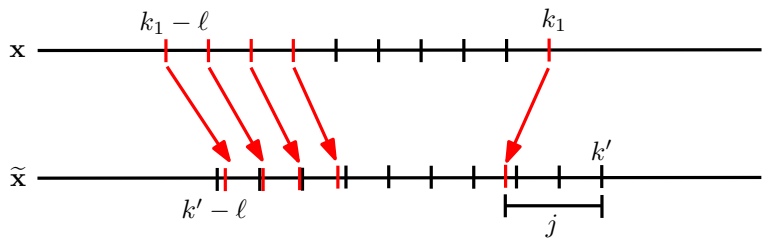

Figure 4. The red arrows indicate to what position a few bits (represented by the vertical red line segments) in the input string are copied in the trace. The vertical line segments (black or red at the top; black at the bottom) indicate the blocks of length λ which are used by the test $T_{\ell,\lambda}$. In the example we may get a positive test with the intervals $\mathbf{x}(k_1 - \ell + 1 : k_1)$ and $\widetilde{\mathbf{x}}(k' - \ell + 1 : k')$ although the alignment error j is rather large, since the number of deletions minus the number of insertions in the latter part of the trace happens to be particularly large, which gives a good overlap between the blocks at the beginning of the string but not at the end.

Meanwhile, the probability of a true match is at least

$$e^{-O(\ell/\lambda^2)} \geq e^{-O(\log^{1/3} n)}.$$

A true match in this case means that for some i with $1 \leq i \leq \ell$, the bit in position $k_1 - \ell + i$ of the input was copied to somewhere very close to position $\tau_1 - \ell_i$ in the output (as will be made precise later).

However, this does not guarantee that the alignment error of τ_1 is $O(C \log n)$: it could happen that between positions $k_1 - \ell + i$ and k_1, the difference between the number of insertions and deletions is more than $C \log n$. See Figure 4 for an illustration of this effect. By Hoeffding's inequality, the probability this happens is at most

$$e^{-\Omega\left((C \log n)^2/\ell\right)} = e^{-\Omega(C \log^{1/3} n)},$$

so that with C large enough, this misalignment scenario happens rarely compared to the true match probability described above.

4.3. Fine alignment and bit statistics. Once we have aligned to within $O(\log n)$, we will perform a second alignment step to align within roughly $O(\log^{1/3} n)$. This involves performing the test $T_{\ell,\lambda}$ with $(\ell, \lambda) = (C^{2/3} \log^{1/3} n, C^{1/12})$. This gives a true match rate of

$$e^{-O(\ell/\lambda^2)} = e^{-O(C^{1/2} \log^{1/3} n)}$$

and spurious match rate of

$$e^{-\Omega(\ell/\lambda)} = e^{-\Omega(C^{7/12} \log^{1/3} n)},$$

so the true matches dominate the spurious ones.

However, the chance of spurious matches is not low enough to union bound over all substrings of length $\ell = C^{2/3}\log^{1/3} n$ that we might align to. In fact, it is actually quite likely that there will be some "bad" substring of length ℓ in the input that produces a high rate of spurious matches (for example, imagine having two nearby runs of ℓ consecutive 0's in the input). Getting around this problem requires some care; the main idea is that although there may be some bad length-ℓ substrings, it is very unlikely that every single length-ℓ substring in an interval of size $\log n$ is bad, and we use one of the not bad strings to align.

The end result of our two alignment steps is some position τ_2 in the output which has an alignment error of $O(\log^{1/3} n)$ to some specified location k_* in the input string $\mathbf{x}$ (i.e., the input position corresponding to τ_2 is within $O(\log^{1/3} n)$ of k_*). Furthermore, k_* is within the last $O(\log n)$ positions of what we have reconstructed so far (i.e., $k - k_* = O(\log n)$).

We can then use a variant of the worst case reconstruction algorithm from [4, 14] (which was also used in [15]) to reconstruct $\mathbf{x}(k_* : k_* + C\log n)$ using $e^{O(\log^{1/3} n)}$ (approximate) traces $\widetilde{\mathbf{x}}(\tau_2 : \infty)$, where shifting of $O(\log^{1/3} n)$ can be tolerated. The algorithm is based on looking at individual bit statistics, and the key property can be roughly stated as follows: consider two possibilities for the string $\mathbf{x}$ which match the first k bits reconstructed so far but disagree on the $(k+1)$-th bit. Then, there will be a noticeable ($e^{-O(\log^{1/3} n)}$) difference in the expected value of one of the positions in $\widetilde{\mathbf{x}}(\tau_2 : \infty)$, which allows us to statistically distinguish the two possibilities. A precise formulation is given in Lemma 20.

4.4. Implementing the algorithm efficiently. While we have thus far given a complete description of how to do trace reconstruction using $e^{O(\log^{1/3} n)}$ traces, there are several obstacles to making this algorithm run in $n^{1+o(1)}$ time.

First, during the alignment stage as described so far, we perform our test T on a sliding window that potentially passes over the whole output string. This is at least $O(n)$ work needed for reconstructing even a single bit, which would lead to an overall running time no better than $O(n^2)$. However, it is quite wasteful to compute our alignment from scratch each time we reconstruct a new bit. Instead, we can use previous alignments as a rough guide, allowing us to skip past all but $n^{o(1)}$ bits of the output string.

Next, to determine the good index k_2, we need some way of assessing whether a given string of length $\log^{1/3} n$ behaves well with our test T in terms of having a high true match rate and low spurious match rate. While explicitly calculating these probabilities is not straightforward, we can estimate them by Monte-Carlo simulation. Recall that the probabilities in question are of order $e^{-O(\log^{1/3} n)} = n^{-o(1)}$, so only $n^{o(1)}$ samples are required to achieve a good enough accuracy.

Finally, in the actual reconstruction step (based on bit statistics as described in the last paragraph of Section 4.3), a naive implementation requires us to test every

possibility for the first roughly $\log n$ unreconstructed bits. However, as observed in [7], the comparison of bit statistics may be formulated as a linear program, which can be solved much more efficiently.

5. Notation for the insertion-deletion channel and Markov properties

Let us introduce some general notation and conventions that will be used for the rest of the paper. To lighten notation, we fix once and for all two values $q, q' \in [0, 1)$ for the deletion and insertion probability, respectively. In order to simplify notation, we will further assume throughout that

$$q = q', \tag{1}$$

so that the expected length of the output equals the length of the input. The same arguments carry through in a straightforward way for general q, q' with appropriate scaling of output lengths. We will use big-O notation, and all implicit constants in $O(\cdot)$ and $\Omega(\cdot)$ expressions may depend on q and q' but nothing else.

We will also need to control the relative sizes of various constant factors. To this end, we introduce a parameter C which will appear in some of our bounds, which should be thought of as a "large constant". We will ultimately complete our argument by choosing C to be sufficiently large (where the threshold for being large enough depends only on q and q').

Next, let us introduce some notation relating to strings and their traces. Recall that $\mathcal{S} := \{0, 1\}^{\mathbb{N}}$ denotes the space of infinite sequences of zeroes and ones. Let $\Omega = \mathcal{S} \times [0, 1]^{\mathbb{N}}$. We denote the first coordinate function on Ω by $\mathbf{x} := (x_0, x_1, \ldots)$ and the second by $\omega := (\omega_0, \omega_1, \ldots)$. Let U be the product uniform measure on $[0, 1]^{\mathbb{N}}$. If ρ is any measure on $\{0, 1\}^{\mathbb{N}}$, let $\mathbb{P}_\rho := \rho \times U$. Thus, our previous notation can be expressed as $\mathbb{P}_{\mathbf{x}} := \mathbb{P}_{\delta_{\mathbf{x}}}$ and $\mathbb{P} := \mathbb{P}_\mu$, where μ is the law of i.i.d. Bernoulli random variables with parameter $1/2$.

We can construct the output $\widetilde{\mathbf{x}} = (\widetilde{x}_0, \widetilde{x}_1, \ldots)$ of the insertion-deletion channel as a function of $\mathbf{x}$ and ω, where ω represents the (random) pattern of insertions and deletions. The construction proceeds as follows. Temporarily denote $a := q(1 - q')/(1 - qq')$ and $b := q'(1 - q)/(1 - qq')$. For each $m \in \mathbb{N}$ we define quantities $s(m), s'(m) \in \mathbb{N}$, where $s(m)$ (resp. $s'(m)$) represents a position in $\mathbf{x}$ (resp. $\widetilde{\mathbf{x}}$) associated with the randomness of $\omega(m)$. We make the definition by setting $s(0) = s'(0) = 0$ and proceeding inductively for $m \geq 0$:

- If $\omega(m) \in [0, a]$, then define $s(m + 1) = s(m) + 1$ and $s'(m + 1) = s'(m)$ (deletion).
- If $\omega(m) \in (a, a + b/2]$, then set $s(m + 1) = s(m)$, $s'(m + 1) = s'(m) + 1$, and $\widetilde{x}_{s'(m)} = 0$ (insertion of 0).

- If $\omega(m) \in (a + b/2, a + b]$, then set $s(m + 1) = s(m)$, $s'(m + 1) = s'(m) + 1$, and $\widetilde{x}_{s'(m)} = 1$ (insertion of 1).
- If $\omega(m) \in (a + b, 1]$, then set $s(m + 1) = s(m) + 1$, $s'(m + 1) = s'(m) + 1$, and $\widetilde{x}_{s'(m)} = x_{s(m)}$ (copy).

We will now justify briefly why this definition of the deletion-insertion channel is equivalent to the one given in Section 3. The channel described in Section 3 is equivalent to the following: (i) before bit x_j insert G_j uniform and independent bits, where G_j is as before, (ii) delete each of the inserted bits independently with probability q, and (iii) delete each of the original bits independently with probability q. The combined effect of (i) and (ii) is to insert $\widehat{G}_j$ uniform and independent bits before bit x_j, where $\widehat{G}_j$ is a geometric random variable with parameter b. From this we conclude equivalence with the channel as described in the bullet points above, because in this channel we insert a geometric number of bits with parameter b between each copy or deletion, and since the fraction of bits in the input string which are deleted is given by $b/(1 - a - b) = q$.

Let us now introduce some notation for corresponding positions in input strings with positions in their traces. Define

$$\psi(j) := \inf\{t \geq 0 : s(t) = j\}, \quad \widetilde{\psi}(j) := \sup\{t \geq 0 : s'(t) = j\}.$$

In other words, $\psi(j)$ is the index of the first coordinate in ω that decides whether to insert a bit before x_j. Similarly, $\widetilde{\psi}(j)$ is the index in ω that determines the value of $\widetilde{x}_j$. Next, define

$$f(k) = s'(\psi(j)), \quad g(k') = s(\widetilde{\psi}(k')).$$

In other words, $f(k)$ is the value of $s'(m)$ at the first time when $s(m) = k$, and $g(k')$ is the value of $s(m)$ at the first time when $s'(m) = k'$. See Figure 5 for an illustration. Roughly speaking, position k in the input gets mapped to position $f(k)$ in the output, and position k' in the output was mapped to from position $g(k')$ in the input. In particular, g is an approximate inverse of f. We also define the function

$$d(k, k') = \max\big(|f(k) - k'|, |g(k') - k|\big), \tag{2}$$

which measures the failure of position k in the input to correspond to position k' in the output.

The next proposition records the Markov property that is satisfied by our insertion-deletion process. It will be convenient to use the notation θ to denote the shift operator on bit strings, i.e., $\theta(\mathbf{x}) = \mathbf{x}(1 : \infty)$ and more generally, $\theta^k(\mathbf{x}) = \mathbf{x}(k : \infty)$.

Proposition 2. *For any $m \geq 0$, conditioned on the values of $s(m)$ and $s'(m)$, the string $\theta^{s'(m)}(\widetilde{\mathbf{x}})$ has the same law as a trace from $\theta^{s(m)}(\mathbf{x})$ through the insertion-deletion channel. In particular, for any $t \geq 0$, the law of $\theta^t(\widetilde{\mathbf{x}})$ is $\mathbb{P}_{\theta^{g(t)}(\mathbf{x})}$.*

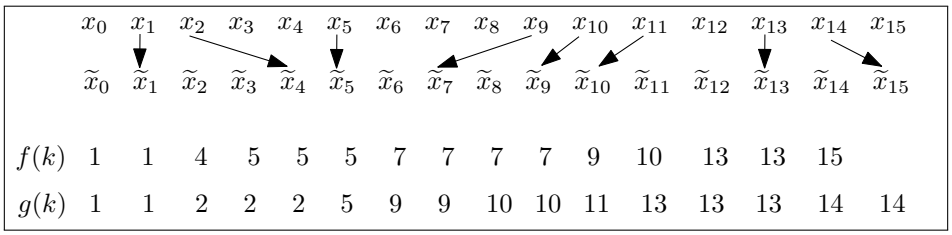

Figure 5. Illustration of the functions f and g. The arrows indicate bits which are copied from $\mathbf{x}$ to $\widetilde{\mathbf{x}}$.

Proof. This is almost immediate from the way we contructed the insertion-deletion channel in terms of ω. It is clear that the increments of $(s(m), s'(m))$ are i.i.d. Thus, starting $(s(m), s'(m))$ from a specified value simply amounts to ignoring the first $s(m)$ bits of the input and writing to the output starting from position $s'(m)$. □

6. Clear robust bias test

We now give a formal definition of the test $T_{\ell,\lambda}$. Recall that the test is designed to answer whether a substring $\widetilde{\mathbf{w}}$ of length ℓ in a trace is likely to have come from a substring $\mathbf{w}$ of the same length in the already recovered part of the input. The test involves subdivision into "blocks" of size approximately $\lambda \le \sqrt{\ell}$. We remark that the test will be applied for (ℓ, λ) on two different scales, namely of order $(\log^{5/3} n, \log^{2/3} n)$ and $(\log^{1/3} n, 1)$.

Given a string $\mathbf{w} := \mathbf{x}(k-\ell+1 : k)$, let $d := \lceil \ell/\lambda \rceil$ denote the number of blocks. The right endpoints of the blocks $\{u_i\}$ will be given by $u_i := k - \ell + \lceil i\ell/d \rceil$. Because $\lambda \le \sqrt{\ell} \le d$, this definition makes $\{(u_{i-1}, u_i] : 1 \le i \le d\}$ a partition of $(k-\ell, k]$ into consecutive intervals of length λ or $\lambda - 1$.

Let us define the *robust bias* of a block $\mathbf{x}(u_{i-1}+1 : u_i)$ to be

$$\lambda^{-1/2} \inf_{\substack{t_1, t_2 \in \mathbb{N}: \\ |t_1 - u_{i-1}| < \lambda/100 \\ |t_2 - u_i| < \lambda/100}} \Big| \sum_{j=t_1}^{t_2} (2x_j - 1) \Big|. \tag{3}$$

We say that a block has a *clear robust bias* if its robust bias is at least 1. See Figure 6 for an illustration.

For some $\theta \in (0, 1/10)$ let $\mathcal{I} \subset \{1, \dots, d\}$ be the indices of the $\lceil \theta d \rceil$ blocks for which the robust bias is largest (with ties resolved in some arbitrary way). By Donsker's theorem, for θ sufficiently small and λ sufficiently large compared to θ, it holds with high probability for large ℓ that all blocks in $\mathcal{I}$ have a clear robust bias.

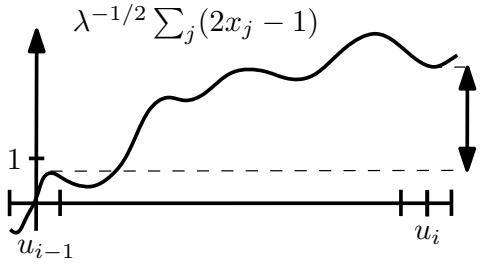

Figure 6. The length of the vertical arrow describes the robust bias associated with the block $\mathbf{x}(u_{i-1}+1:u_i)$. The curve represents the partial sums $\lambda^{-1/2}\sum_j(2x_j-1)$, renormalized to equal 0 at u_{i-1}. We say that the robust bias is clear if it is at least 1, such as shown in the given example.

We fix such a choice of θ as follows: for B a standard Brownian motion, let

$$\theta := \frac{1}{10}\mathbb{P}\left[\inf_{t_1\in[0,1/50],t_2\in[1,1+1/50]}|B_{t_2}-B_{t_1}|>1\right]>0\,. \tag{4}$$

For each i, define the quantity

$$s_i := \sum_{j=u_{i-1}+1}^{u_i}(2x_j-1) \tag{5}$$

which counts the number of 1's minus the number of 0's in the i-th block. Define $\widetilde{s}_i$ similarly for a string $\widetilde{\mathbf{w}}$ of the same length as $\mathbf{w}$. We first define our test using an extra parameter $c\in(0,1)$. The test is given by

$$T^c_{\ell,\lambda}(\mathbf{w},\widetilde{\mathbf{w}}) = \begin{cases}1 & \text{if } \sum_{i\in\mathcal{I}}\operatorname{sign}(s_i)\cdot\operatorname{sign}(\widetilde{s}_i) > c|\mathcal{I}|,\\ 0 & \text{otherwise,}\end{cases}$$

where $|\mathcal{I}|$ denotes the cardinality of $\mathcal{I}$. We will only apply the test for a particular value of c, which will be chosen depending on the insertion/deletion probabilities as described in the next subsection.

6.1. Estimates for the test T.

Definition 3. We say that a string $\widetilde{\mathbf{w}}$ of length ℓ *has clear robust bias at scale* λ if at least $\lceil\theta d\rceil = |\mathcal{I}|$ of its blocks have clear robust bias.

First, we formally state our earlier claim that due to our sufficiently small choice of θ, a random string has clear robust bias with high probability.

Lemma 4. *Let* $\mathbf{w}$ *be a random string of length* ℓ*. Then,* $\mathbf{w}$ *fails to have clear robust bias at scale* λ *with probability at most* $e^{-\Omega(\ell/\lambda)}$.

Proof. Within a single block, by Donsker's theorem, the partial sums of the bits converge to Brownian motion as $\lambda \to \infty$. Thus, for sufficiently large λ, the probability that this block has clear robust bias is at least 2θ. Consequently, the probability that the proportion of blocks with clear robust bias is less than θ is exponentially small in the number of blocks, i.e., it is of order $e^{-\Omega(\ell/\lambda)}$. □

Lemma 5. *Let* $\mathbf{w}$ *be a string of length* ℓ *which exhibits robust bias at scale* λ*. Suppose that we take a trace* $\widetilde{\mathbf{w}}$ *of* $\mathbf{w}$ *through the insertion-deletion channel. Then, for a small enough constant* $c > 0$ *depending only on the insertion/deletion probabilities, we have*

$$\mathbb{P}\big(T^c_{\ell,\lambda}(\mathbf{w}, \widetilde{\mathbf{w}}(0 : \ell - 1)) = 1\big) \geq e^{-O(\ell/\lambda^2)}.$$

In order to prove Lemma 5, we first define a condition describing when $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ are unusually well aligned. We will then show that $T^c_{\ell,\lambda}$ is reasonably likely to test positive conditioned on this good alignment.

Definition 6. Let $\mathbf{w}$ be a string of length ℓ and $\widetilde{\mathbf{w}}$ a trace of $\mathbf{w}$ through the insertion-deletion channel. Following the notation of this section, for a block size λ, we let $d = \lceil \ell/\lambda \rceil$, and we let $\{u_i\}_{i=0}^d$ denote the endpoints of the blocks. We say that the trace is *s-aligned* if for each $0 \leq i \leq d$, it holds that

$$|(u_i - u_0) - (f(u_i) - f(u_0))| < s.$$

If the trace is s-aligned, we say that we have *s-alignment*.

Proof of Lemma 5. Let c_0 be a small constant to be specified later. We first show that the probability of $c_0\lambda$-alignment is at least $e^{-O(\ell/c_0\lambda^2)}$. For convenience, define the function

$$\zeta(x) := (x - u_0) - (f(x) - f(u_0)).$$

Consider any λ consecutive blocks numbered $i + 1$ through $i + \lambda$. Note that the sequence $\{\frac{\zeta(k)}{\lambda}\}_{k=u_i}^{u_{i+\lambda}}$ is a mean-zero simple random walk (since we assume $q = q'$), where the distribution of the increments is determined by the insertion and deletion probabilities and has finite variance.

By Donsker's theorem, $\{\frac{\zeta(k)}{\lambda}\}_{k=u_i}^{u_{i+\lambda}}$ converges to a constant multiple of standard Brownian motion as $\lambda \to \infty$ (where time is scaled by λ^2). For a standard Brownian motion $\{W_t\}_{0\leq t\leq 1}$, we have that

$$\mathbb{P}\big(W_0, W_1 \in [-c_0/2, c_0/2] \text{ and } |W_t| \leq c_0 \text{ for all } 0 \leq t \leq 1\big) = e^{-O(1/c_0)}.$$

Thus, we have a similar statement for the quantities $\frac{\zeta(k)}{\lambda}$, where

$$\mathbb{P}\left(\frac{|\zeta(u_i)|}{\lambda} \leq \frac{c_0}{2}, \ \frac{|\zeta(u_{i+\lambda})|}{\lambda} \leq \frac{c_0}{2}, \ \text{and} \ \max_{i<j<i+\lambda} \frac{|\zeta(u_{i+\lambda})|}{\lambda} \leq c_0\right) \geq e^{-O(1/c_0)}.$$

Chaining together these events for each group of λ consecutive blocks (of which there are roughly ℓ/λ^2), we see that the overall probability of being $c_0\lambda$-aligned is at least $e^{-O(\ell/c_0\lambda^2)}$.

Our next step is to estimate the probability of a positive test conditioned on the trace being $c_0\lambda$-aligned. We further condition on the specific values of the $f(u_i)$. Note that the substring $\mathbf{w}(u_i + 1 : u_{i+1})$ must be transformed into the substring $\widetilde{\mathbf{w}}(f(u_i) + 1 : f(u_{i+1}))$. The insertion/deletion patterns of these transformations are all independent, and they have the same distribution as the insertion-deletion channel applied to $\mathbf{w}(u_i : u_{i+1} - 1)$ conditioned on the output trace having length $f(u_{i+1}) - f(u_i)$.

Let m_i denote the number of bits copied to $\widetilde{\mathbf{w}}(f(u_i) : f(u_{i+1}) - 1)$ from $\mathbf{w}(u_i : u_{i+1} - 1)$, and note that we have $m_i = \Omega(\lambda)$ with probability $1 - e^{-\Omega(\lambda)}$. Also, since we assume that our trace is $c_0\lambda$-aligned, only $O(c_0\lambda)$ bits in $\widetilde{\mathbf{w}}(u_i : u_{i+1} - 1)$ were copied from outside of $\mathbf{w}(u_i : u_{i+1} - 1)$. Finally, note that any bits in $\widetilde{\mathbf{w}}(u_i : u_{i+1} - 1)$ which were not copied from $\mathbf{w}$ are i.i.d. uniformly random.

Recall that $\widetilde{\mathbf{w}}$ was assumed to have robust bias at scale λ. Thus, if $i \in \mathcal{I}$, then with probability $1 - e^{-\Omega(\lambda)}$ we will have $\Omega(\sqrt{\lambda})$ bias in the sum of bits in $\widetilde{\mathbf{w}}(u_{i-1} : u_i)$ copied from $\mathbf{w}$. It follows that for large enough λ and small enough c_0, the correlation between $\mathrm{sign}(\widetilde{s}_i)$ and $\mathrm{sign}(s_i)$ is $\Omega(1)$. This means that (after all our conditioning) as long as c is small enough, the test $T^c_{\ell,\lambda}$ will succeed except on an event with probability $e^{-\Omega(\ell/\lambda)}$.

In summary, there is at least a $e^{-O(\ell/\lambda^2)}$ chance to have λ-alignment, and conditioning on this, a positive match occurs with high probability. This proves the lemma. □

From now on, we will always run our tests using the value of c as in Lemma 5. Thus, we will henceforth simply write $T_{\ell,\lambda}$ instead of $T^c_{\ell,\lambda}$.

Definition 7. Let $\mathbf{w} = \mathbf{x}(a + 1 : a + \ell)$ be a substring of the input, and let $\widetilde{\mathbf{w}} = \widetilde{\mathbf{x}}(b + 1 : b + \ell)$ be a substring of the same length taken from the trace. We say that $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ are *s-mismatched* if for any $0 \le i \le \ell$ it holds that $d(a + i, b + i) > s$, where d is defined in (2).

Lemma 8. *Let $\mathbf{x}$ be a random string, and suppose we sample a trace $\widetilde{\mathbf{x}}$ from the insertion-deletion channel.*

Consider two length-ℓ substrings $\mathbf{w} = \mathbf{x}(a+1 : a+\ell)$ and $\widetilde{\mathbf{w}} = \widetilde{\mathbf{x}}(b+1 : b+\ell)$. Let ω^0 be a realization of the randomness of the insertion-deletion channel (i.e., an insertion/deletion pattern) for which $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ are λ-mismatched. Then,

$$\mathbb{P}\left(T_{\ell,\lambda}(\mathbf{w}, \widetilde{\mathbf{w}}) = 1 \mid \omega = \omega^0\right) \le e^{-\Omega(\ell/\lambda)}.$$

Proof. Note that after conditioning on ω, the remaining randomness is to sample the values of the bits. We do this incrementally block by block (where the blocks are of

size λ). Suppose that we have already sampled the first $i-1$ blocks of $\mathbf{w}$ and $\widetilde{\mathbf{w}}$; let us consider the conditional distribution on bits in the i-th blocks of $\mathbf{w}$ and $\widetilde{\mathbf{w}}$.

Note that there are only two ways dependency can occur between bits in the i-th blocks of $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ and the already sampled bits: either a bit in the i-th block of $\widetilde{\mathbf{w}}$ came from one of the first $i-1$ blocks of $\mathbf{w}$ or a bit in one of the first $i-1$ blocks of $\widetilde{\mathbf{w}}$ came from a bit in the i-th block of $\mathbf{w}$. Moreover, these two scenarios are mutually exclusive. It follows that the i-th block of at least one of $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ is completely independent of the already sampled bits.

Suppose for example that the i-th block of $\mathbf{w}$ is independent of the already sampled bits. Since $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ are λ-mismatched, the i-th blocks of $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ are also independent of each other. It follows that we may first sample the i-th block of $\widetilde{\mathbf{w}}$, and conditioned on that, $\operatorname{sign}(s_i)$ is equally likely to be 1 or -1. A similar argument holds in the other case, where the i-th block of $\widetilde{\mathbf{w}}$ is independent of the first $i-1$ blocks of $\mathbf{w}$ and $\widetilde{\mathbf{w}}$.

Either way, the end result is that $\operatorname{sign}(s_i)\cdot\operatorname{sign}(\widetilde{s}_i)$ is a uniform random sign independent of the previous blocks. It follows that the quantity

$$\sum_{i\in\mathcal{I}}\operatorname{sign}(s_i)\cdot\operatorname{sign}(\widetilde{s}_i)$$

has the law of a sum of $|\mathcal{I}|$ independent random signs, so by Hoeffding's inequality, the chance that it exceeds the threshold $c|\mathcal{I}|$ required for our test $T_{\ell,\lambda}$ is

$$e^{-\Omega(|\mathcal{I}|)} = e^{-\Omega(\ell/\lambda)},$$

as desired. □

7. The "good" set of strings

Definition 9. Let ℓ and λ be given positive integers. Let $\mathbf{x}$ be an input string, let I be an interval of length ℓ, and write $\mathbf{w}=\mathbf{x}(I)$. Let J be another interval (often we will have $I\subseteq J$).

Suppose we take a trace of $\mathbf{x}$ through the insertion-deletion channel. We say that an (I,J)-*spurious match* occurs if for some substring $\widetilde{\mathbf{w}}=\widetilde{\mathbf{x}}(i_1:i_2)$ of the output such that $g([i_1,i_2])\subseteq J$, we have $T_{\ell,\lambda}(\mathbf{w},\widetilde{\mathbf{w}})=1$, but $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ are λ-mismatched. We use

$$\mathcal{Q}_{T_{\ell,\lambda}}(I,J)$$

to denote the event that an (I,J)-spurious match occurs.

Lemma 10. *Let ℓ and λ be given positive integers. Let I be an interval of length ℓ, and let $J \supseteq I$ be an interval containing I. Suppose we have an input string $\mathbf{x}$ all of whose bits are determined except those in J, which are drawn i.i.d. uniformly. Then, letting $|J|$ denote the length of J,*

$$\mathbb{P}(\mathcal{Q}_{T_{\ell,\lambda}}(I, J)) \leq |J| e^{-\Omega(\ell/\lambda)} + e^{-\Omega(|J|)}.$$

Proof. We will first condition on an insertion/deletion pattern ω from the insertion-deletion channel. For each interval I' of length ℓ, if ω^0 is an insertion/deletion pattern which causes I' and I to be λ-mismatched, then we know by Lemma 8 that

$$\mathbb{P}(T_{\ell,\lambda}(\mathbf{x}(I), \widetilde{\mathbf{x}}(I')) = 1 \mid \omega = \omega^0) \leq e^{-\Omega(\ell/\lambda)}. \tag{6}$$

Let J' denote the minimal interval containing $f(J)$, and note that J' is a function of the insertion/deletion pattern ω. Then, conditioning on J' and taking a union bound over all possible $I' \subseteq J'$ in (6) gives

$$\mathbb{P}(\mathcal{Q}_{\ell,\lambda}(I, J) \mid J') \leq |J'| e^{-\Omega(\ell/\lambda)}.$$

Since we also have $\mathbb{P}(|J'| \geq 2|J|) \leq e^{-\Omega(|J|)}$, this yields our final bound

$$\mathbb{P}(\mathcal{Q}_{T_{\ell,\lambda}}(I, J)) \leq |J| e^{-\Omega(\ell/\lambda)} + e^{-\Omega(|J|)}. \qquad \square$$

Lemma 11. *Let ℓ and λ be given positive integers. Let I be an interval of length ℓ, and let J be another disjoint interval whose distance from I is at least $|J|$. Suppose we have an input string $\mathbf{x}$ all of whose bits are determined except those in I, which are drawn i.i.d. uniformly. Then,*

$$\mathbb{P}(\mathcal{Q}_{T_{\ell,\lambda}}(I, J)) \leq |J| e^{-\Omega(\ell/\lambda)} + e^{-\Omega(|J|)}.$$

Proof. Let J' be the minimal interval containing $f(J)$, and define the event

$$E = \big\{|J'| \leq 2|J| \text{ and not all bits between } I \text{ and } J \text{ were deleted}\big\}.$$

Note that E is measurable with respect to the σ-field generated by ω (i.e., the randomness of the insertion-deletion channel), and $\mathbb{P}(E) \geq 1 - e^{-\Omega(|J|)}$.

Meanwhile, conditioned on E, consider any subinterval $I' \subseteq J'$ of length ℓ. None of the bits in I' come from I, so the random bits in I are independent of the bits in I'. Consequently, we have

$$\mathbb{P}(T_{\ell,\lambda}(\mathbf{x}(I), \widetilde{\mathbf{x}}(I')) = 1 \mid E) \leq e^{-\Omega(\ell/\lambda)}.$$

Taking a union bound over possible choices of I' given that E occurs, we conclude that

$$\mathbb{P}(\mathcal{Q}_{T_{\ell,\lambda}}(I, J)) \leq |J| e^{-\Omega(\ell/\lambda)} + \mathbb{P}(E^c) \leq |J| e^{-\Omega(\ell/\lambda)} + e^{-\Omega(|J|)}. \qquad \square$$

7.1. Coarsely well-behaved strings.

Definition 12. Let $\mathbf{x}$ be a string of length at least n. Let $\ell = C\log^{5/3} n$ and $\lambda = C^{1/2}\log^{2/3} n$. We say that $\mathbf{x}$ is *coarsely well-behaved* if for each interval $I \subseteq [0,n]$ of length ℓ, it holds that $\mathbf{x}(I)$ has robust bias at scale λ and

$$\mathbb{P}_{\mathbf{x}}(\mathcal{Q}_{\ell,\lambda}(I,[0,n])) \le n^{-2}.$$

Lemma 13. *Let* $\mathbf{x}$ *be a random string. Then,* $\mathbf{x}$ *is coarsely well-behaved with probability at least* $1-n^{-2}$.

Proof. Consider first a particular interval $I = (a, a+\ell]$ of length ℓ. By Lemma 4, we know that $\mathbf{x}(I)$ has robust bias at scale λ with probability at least

$$1 - e^{-\Omega\left(\ell/\lambda\right)} = 1 - e^{-\Omega(C^{1/2}\log n)} \ge 1 - n^{-4}$$

for large enough C.

We also have from Lemma 10 that

$$\mathbb{P}(\mathcal{Q}_{\ell,\lambda}(I,[0,n])) \le e^{-\Omega(C^{1/2}\log n)}.$$

Thus, for large enough C, we can ensure by Markov's inequality that

$$\mathbb{P}\left(\mathbb{P}_{\mathbf{x}}(\mathcal{Q}_{\ell,\lambda}(I,[0,n])) \ge n^{-2}\right) \le n^{-4}.$$

Taking a union bound over at most n possible values of I completes the proof. □

7.2. Finely well-behaved strings.

Recall Lemmas 10 and 11. Let $c_0 > 0$ be such that the terms $\Omega(\ell/\lambda)$ and $\Omega(|J|)$ in these lemmas could have been replaced by $10c_0\ell/\lambda$ and $10c_0|J|$, respectively.

Definition 14. Let $\mathbf{x}$ be a string of length n, and let $\ell = C^{2/3}\log^{1/3} n$ and $\lambda = C^{1/12}$. We say that $\mathbf{x}$ is *finely well-behaved* if for each interval $J := [a, a + C\log n] \subseteq [0,n]$ of length $C\log n$, there exists a subinterval

$$I \subset \left[a + \tfrac{1}{3}C\log n, a + \tfrac{2}{3}C\log n\right]$$

of size ℓ such that $\mathbf{x}(I)$ has robust bias at scale λ and

$$\mathbb{P}_{\mathbf{x}}\left(\mathcal{Q}_{\ell,\lambda}(I,J)\right) \le e^{-c_0 C^{7/12}\log^{1/3} n}.$$

Lemma 15. *Let* $\mathbf{x}$ *be a random string. Then,* $\mathbf{x}$ *is finely well-behaved with probability at least* $1-n^{-2}$.

Proof. Throughout the proof the implicit constants in $\Omega(\cdot)$ and $O(\cdot)$ may depend on c_0 but not on C. Fix a particular interval $J = [a, a + C\log n] \subseteq [0,n]$ of length $C\log n$, and let $\ell = C^{2/3}\log^{1/3} n$ and $\lambda = C^{1/12}$ be as in Definition 14.

Consider $m := \frac{1}{3}C^{1/3}\log^{2/3} n$ disjoint length-ℓ subintervals

$$I_1, I_2, \ldots, I_m \subset \left[a + \tfrac{1}{3}C\log n, a + \tfrac{2}{3}C\log n\right].$$

For a given realization of $\mathbf{x}$, we say that I_i is *bad* if either it does not have clear robust bias at scale λ or it holds that

$$\mathbb{P}_{\mathbf{x}}\big(\mathcal{Q}_{\ell,\lambda}(I_i, J)\big) \geq e^{-c_0 C^{7/12}\log^{1/3} n}. \tag{7}$$

Let $\ell' := C^{1/6}\ell = C^{5/6}\log^{1/3} n$ and define the event

$$H = \left\{\begin{array}{l}\text{there exists some } t \text{ such that} \\ \quad g(t), g(t+\ell) \in I \text{ and } |g(t) - g(t+\ell)| \geq \ell'\end{array}\right\},$$

which roughly says that a substring of length ℓ' had so many deletions that only ℓ or fewer bits were left in the output. We have that $\mathbb{P}(H) \leq e^{-\Omega(\ell')}$.

As long as H does not occur, then any spurious match counted in (7) must have come from an interval J' of length ℓ', i.e., we have

$$\mathcal{Q}_{\ell,\lambda}(I_i, J) \subseteq H \cup \left(\bigcup_{\substack{J' \subset J \text{ an interval} \\ \text{of length } \ell'}} \mathcal{Q}_{\ell,\lambda}(I_i, J')\right).$$

Thus, by the pigeonhole principle, if I_i is bad due to (7), then there must be some interval J_i' of length ℓ' for which

$$\mathbb{P}_{\mathbf{x}}\big(\mathcal{Q}_{\ell,\lambda}(I_i, J_i')\big) \geq e^{-c_0 C^{7/12}\log^{1/3} n}. \tag{8}$$

We say that (I_i, J_i') is a *bad pair* if either I_i does not have clear robust bias at scale λ, or (8) holds. In particular, the above discussion shows that if I_i is bad, then there is some interval J_i' of length ℓ' for which (I_i, J_i') is a bad pair.

Suppose for the sake of contradiction that with probability at least n^{-2} in the randomness of $\mathbf{x}$, all the I_i are bad (i.e., $\mathbf{x}$ is not finely well-behaved). Each I_i is part of a bad pair (I_i, J_i'), and note that there are at most $(C\log n)^m = n^{o(1)}$ possible values for $(J_1', \ldots, J_m')$. Thus, by the pigeonhole principle, it must hold for some specific choice of $(J_1', \ldots, J_m')$ that

$$\mathbb{P}\big((I_i, J_i') \text{ is a bad pair for } i = 1, 2, \ldots, m\big) \geq n^{-3}. \tag{9}$$

Fixing this choice of $(J_1', \ldots, J_m')$, we will derive a contradiction.

Let $r := 0.01C^{1/6}\log^{2/3} n$. To carry out the analysis, we inductively define a sequence $i_1, i_2, \ldots, i_r$ as follows. We take $i_1 = 1$, and for $k \geq 1$, let

$$N_k = \bigcup_{j=1}^{k} (I_{i_j} \cup J_{i_j}').$$

Then, choose i_{k+1} so that $I_{i_{k+1}}$ is distance at least $2\ell'$ from N_k. Note that the $2\ell'$-neighborhood of N_k intersects at most $2k \cdot \lceil 5\ell'/\ell \rceil \le 12C^{1/6}k$ of the I_i, so such a choice is always possible as long as $k \le r$.

Let $\mathcal{G}_k$ be the σ-field generated by the bits of $\mathbf{x}$ whose positions are in N_k, and let E_k denote the event that (I_{i_k}, J_{i_k}) is a bad pair. Note that E_k is measurable with respect to $\mathcal{G}_k$.

First of all, note that whether I_{i_k} has clear robust bias at scale λ is independent of $\mathcal{G}_{k-1}$, so by Lemma 4, we have

$$\mathbb{P}\left(\begin{matrix} I_{i_k} \text{ does not have clear} \\ \text{robust bias at scale } \lambda \end{matrix} \,\middle|\, \mathcal{G}_{k-1} \right) \le e^{-\Omega(\ell/\lambda)} = e^{-\Omega(C^{7/12}\log^{1/3} n)}.$$

Next, we will estimate $\mathbb{P}\big(\mathcal{Q}_{\ell,\lambda}(I_{i_k}, J'_{i_k}) \mid \mathcal{G}_{k-1}\big)$. Suppose first that I_{i_k} and J'_{i_k} are disjoint and distance at least ℓ' apart. Then, by Lemma 11, we have

$$\mathbb{P}\big(\mathcal{Q}_{\ell,\lambda}(I_{i_k}, J'_{i_k}) \mid \mathcal{G}_{k-1}\big) \le \ell' e^{-10c_0\ell/\lambda} + e^{-10c_0\ell'} \le e^{-9c_0C^{7/12}\log^{1/3} n}.$$

If instead I_{i_k} and J'_{i_k} are within distance ℓ' of each other, then let J be the interval formed by extending I_{i_k} on both sides by $2\ell'$, so that $J'_{i_k} \subset J$. By our construction, it is also guaranteed that J is disjoint from N_{k-1}, and so when conditioning on $\mathcal{G}_{k-1}$, none of its bits have been determined yet. Then, we may apply Lemma 10 to obtain

$$\begin{aligned}\mathbb{P}\big(\mathcal{Q}_{\ell,\lambda}(I_{i_k}, J'_{i_k}) \mid \mathcal{G}_{k-1}\big) &\le \mathbb{P}\big(\mathcal{Q}_{\ell,\lambda}(I, J) \mid \mathcal{G}_{k-1}\big) \\ &\le 3\ell' e^{-10c_0\ell/\lambda} + e^{-10c_0\ell'} = e^{-9c_0C^{7/12}\log^{1/3} n}.\end{aligned}$$

Thus, the above bound holds in either case, and by Markov's inequality, this implies

$$\mathbb{P}\big(\mathbb{P}_{\mathbf{x}}\big(\mathcal{Q}_{\ell,\lambda}(I_{i_k}, J'_{i_k})\big) \ge e^{-c_0C^{7/12}\log^{1/3} n} \mid \mathcal{G}_{k-1}\big) \le e^{-\Omega(C^{7/12}\log^{1/3} n)}.$$

It follows that

$$\begin{aligned}\mathbb{P}(E_k \mid \mathcal{G}_{k-1}) &\le \mathbb{P}\left(\begin{matrix} I_{i_k} \text{ does not have clear} \\ \text{robust bias at scale } \lambda \end{matrix} \,\middle|\, \mathcal{G}_{k-1} \right) \\ &\quad + \mathbb{P}\big(\mathbb{P}_{\mathbf{x}}\big(\mathcal{Q}_{\ell,\lambda}(I_{i_k}, J'_{i_k})\big) \ge e^{-c_0C^{7/12}\log^{1/3} n} \mid \mathcal{G}_{k-1}\big) \\ &\le e^{-\Omega(C^{7/12}\log^{1/3} n)}.\end{aligned}$$

Iterating this over $k = 1, \dots, r$, we finally obtain

$$\mathbb{P}(E_1 \cap E_2 \cap \cdots \cap E_r) \le e^{-\Omega(C^{7/12}\log^{1/3} n)\cdot r} = e^{-\Omega(C^{3/4}\log n)},$$

which is smaller than n^{-3} for large enough C. This gives our desired contradiction of (9), completing the proof. □

8. Alignment rules

For the rest of the paper, let Ξ_{bad} denote the set of strings that either fail to be coarsely well-behaved or finely well-behaved.

Lemma 16. *Let $k \geq \ell$ be a given positive integer, and let $\ell = C\log^{5/3} n$ and $\lambda = C^{1/2}\log^{2/3} n$ as in Definition 12. Define*

$$\tau_1^k = \tau_1^k(\widetilde{\mathbf{x}}) := \inf\left\{k' \in [\ell, 2n] : T_{\ell,\lambda}(\mathbf{x}(k-\ell+1:k), \widetilde{\mathbf{x}}(k'-\ell+1:k')) = 1\right\},$$

where we set $\tau_1^k = \infty$ if no such k' exists. If $\mathbf{x} \notin \Xi_{\text{bad}}$, then

$$\mathbb{P}\left(\tau_1^k < \infty,\ d(k,\tau_1^k) > \tfrac{1}{10}C\log n\right) \leq e^{-\Omega(C\log^{1/3} n)}$$

and

$$\mathbb{P}\left(\tau_1^k < \infty,\ d(k,\tau_1^k) \leq \tfrac{1}{10}C\log n\right) \geq e^{-O(\log^{1/3} n)}.$$

Proof. Let $I = (k-\ell, k]$, and define the events

$$E = \{\tau_1^k < \infty\}, \quad E' = \left\{\tau_1^k < \infty,\ d(k,\tau_1^k) > \tfrac{1}{10}C\log n\right\},$$
$$F = \left\{\begin{array}{c}\text{the difference between the number of inserted and}\\ \text{deleted bits among those in } I \text{ is at least } \frac{1}{20}C\log n\end{array}\right\}.$$

Note that by Hoeffding's inequality, we have

$$\mathbb{P}_{\mathbf{x}}(F) = e^{-\Omega\left(\frac{(C\log n)^2}{C\log^{5/3} n}\right)} = e^{-\Omega(C\log^{1/3} n)}.$$

To show the first inequality, which amounts to bounding $\mathbb{P}_{\mathbf{x}}(E')$, suppose that E holds but $\mathcal{Q}_{\ell,\lambda}(I,[0,n])$ does not. Then, it must be that the matched string $\widetilde{\mathbf{x}}(\tau_1^k-\ell+1:\tau_1^k)$ is not λ-mismatched. However, if additionally $d(k,\tau_1^k) > C\log n$, then F must hold. Thus,

$$\begin{aligned}\mathbb{P}_{\mathbf{x}}(E') &\leq \mathbb{P}_{\mathbf{x}}(\mathcal{Q}_{\ell,\lambda}(I,[0,n])) + \mathbb{P}_{\mathbf{x}}(E' \setminus \mathcal{Q}_{\ell,\lambda}(I,[0,n]))\\ &\leq n^{-2} + e^{-\Omega(C\log^{1/3} n)} = e^{-\Omega(C\log^{1/3} n)},\end{aligned}$$

establishing the first inequality.

Next, note that by Lemma 5, a positive match will be found (i.e., E will hold) with probability at least $e^{-O(\ell/\lambda^2)} = e^{-O(\log^{1/3} n)}$. Thus,

$$\mathbb{P}_{\mathbf{x}}(E \setminus E_1) \geq e^{-O(\log^{1/3} n)} - e^{-\Omega(C\log^{1/3} n)} = e^{-O(\log^{1/3} n)},$$

establishing the second inequality. □

Lemma 17. *Let $\mathbf{x} \notin \Xi_{\text{bad}}$ be a string, and let k be a positive integer. Suppose that we know $\mathbf{x}(0:k)$, and let $\varphi(t) = te^{t/(C^{2/3}\log^{2/3} n)}$. Let $\mathcal{F}_j$ denote the σ-algebra generated by $\mathbf{x}(0:k)$ and $\widetilde{\mathbf{x}}(0:j)$. Then, there is a position $k_* \in [k - C\log n, k]$*

and a stopping time $\tau_2^k(\widetilde{\mathbf{x}})$ for $\mathcal{F}_j$ for which the following properties hold: defining the event

$$F^k = F^k_{\mathbf{x}} := \big\{\tau_2^k(\widetilde{\mathbf{x}}) < \infty,\ g(\tau_2^k(\widetilde{\mathbf{x}})) \in [k - C\log n, k]\big\},$$

we have

1. $\mathbb{P}\big[\{\tau_2^k(\widetilde{\mathbf{x}}) < \infty\} \setminus F^k\big] \le n^{-2}$,
2. $\mathbb{P}\big[F^k\big] \ge e^{-O(C^{1/2}\log^{1/3} n)}$,
3. $\mathbb{E}\big[\varphi\big(|g(\tau_2^k(\widetilde{\mathbf{x}})) - k_*|\big) \mid F^k\big] \le O(C^{2/3}\log^{1/3} n)$.

Remark 18. In fact, the proof below implies that the same result holds if instead of requiring $\mathbf{x} \notin \Xi_{\text{bad}}$, we only require that its first k bits match some string not in Ξ_{bad}.

Proof. Let $\ell = C^{2/3}\log^{1/3} n$ and $\lambda = C^{1/12}$, and let

$$I \subset \big[k - \tfrac{2}{3}C\log n, k - \tfrac{1}{3}C\log n\big]$$

be the interval guaranteed by Definition 14 (since $\mathbf{x}$ is finely well-behaved). Let τ_1^k be as in Lemma 16, and consider the interval

$$I' = \big[\tau_1^k(\widetilde{\mathbf{x}}) + \tfrac{1}{6}C\log n, \tau_1^k(\widetilde{\mathbf{x}}) + \tfrac{5}{6}C\log n\big].$$

We define

$$\tau_2^k(\widetilde{\mathbf{x}}) := \inf\big\{k' : [k'-\ell, k'] \subset I' \text{ and } T(\mathbf{x}(I), \widetilde{\mathbf{x}}(k'-\ell : k')) = 1\big\},$$

where as usual we set $\tau_2^k(\widetilde{\mathbf{x}}) = \infty$ if no such k' exists (or if $\tau_1^k(\widetilde{\mathbf{x}}) = \infty$). Our choice of k_* is then the right endpoint of I.

To lighten notation, in the rest of the proof we write $\tau_1^k = \tau_1^k(\widetilde{\mathbf{x}})$ and $\tau_2^k = \tau_2^k(\widetilde{\mathbf{x}})$. In several of our calculations, it will be convenient to exclude the event

$$E := \big\{|g(\tau_1^k + t) - (k - C\log n + t)| > \tfrac{1}{8}C\log n \text{ for some } 0 \le t \le C\log n\big\}.$$

We first show that E is a rare event. Recall that by the properties of τ_1^k established in Lemma 16, we have that

$$\mathbb{P}\big(|g(\tau_1^k) - k + C\log n| \ge \tfrac{1}{10}C\log n\big) \le n^{-2}.$$

On the other hand, if $|g(\tau_1^k) - k + C\log n| \le \frac{1}{10}C\log n$, then in order for E to occur, among the bits with input positions between $k - \frac{11}{10}C\log n$ and k, the difference between the number of deletions and insertions must have been at least $\Omega(C\log n)$. This occurs with probability at most $n^{-\Omega(C)} \le n^{-2}$ for large enough C. Thus, we see that $\mathbb{P}(E) \le O(n^{-2})$.

Let us now establish the properties stated in the lemma. The first property immediately follows from our bound on $\mathbb{P}(E)$, since whenever $\tau_2^k < \infty$, we have

$$g(\tau_1^k) + \tfrac{1}{6}C \log n \le g(\tau_2^k) \le g(\tau_1^k) + \tfrac{5}{6}C \log n.$$

Thus, we have $\{\tau_2^k < \infty\} \setminus F^k \subseteq E$, and so

$$\mathbb{P}\big(\{\tau_2^k < \infty\} \setminus F^k\big) \le \mathbb{P}(E) \le O(n^{-2}).$$

Recall from Lemma 5 that with probability at least $e^{-O(\ell/\lambda^2)} = e^{-O(C^{1/2}\log^{1/3} n)}$, the bits coming from I will form a positive match for the test $T_{\ell,\lambda}$. Outside of the event E, this match will be detected by our procedure, and so

$$\mathbb{P}(\tau_2^k < \infty) \ge e^{-O(C^{1/2}\log^{1/3} n)} - \mathbb{P}(E) = e^{-O(C^{1/2}\log^{1/3} n)}.$$

Subtracting the bound from the first property yields the second property.

For the last property, let us estimate the probability

$$\mathbb{P}\big[F^k \cap \{|g(\tau_2^k) - k_*| \ge 2\ell\}\big].$$

There are two possible cases to consider:

1. A spurious match event $\mathcal{Q}_{\ell,\lambda}(I, [k - C\log n, k])$ may occur.

2. If there is no spurious match event but F^k holds, then it means there is some $t \in [\tau_2^k - \ell, \tau_2^k]$ for which $g(t) \in [k_* - \ell, k_*]$. Then, the only way to have

$$|g(\tau_2^k) - k_*| \ge 2\ell$$

is if there were at least ℓ more deletions than insertions in the length-2ℓ input interval $[k_*, k_* + 2\ell]$.

By our choice of the interval I, we can estimate the probability of the first scenario by

$$\mathbb{P}(\mathcal{Q}_{\ell,\lambda}(I, [k - C\log n, k])) \le e^{-\Omega(C^{7/12}\log^{1/3} n)},$$

and the last scenario has probability $e^{-\Omega(\ell)} = e^{-\Omega(C^{2/3}\log^{1/3} n)}$. Thus, the overall probability is

$$\mathbb{P}\big[F^k \cap \{|g(\tau_2^k) - k_*| \ge 2\ell\}\big] \le e^{-\Omega(C^{7/12}\log^{1/3} n)},$$

and so

$$\frac{\mathbb{P}\big[F^k \cap \{|g(\tau_2^k) - k_*| \ge 2\ell\}\big]}{\mathbb{P}\big[F^k \cap \{|g(\tau_2^k) - k_*| \le 2\ell\}\big]} \le e^{-\Omega(C^{1/2}\log^{1/3} n)}.$$

To calculate the relevant expectation, we can divide into cases depending on whether $|g(\tau_2^k) - k_*| \le 2\ell$ or not (note that on the event F^k, we always have $|g(\tau_2^k) - k_*| \le C \log n$). This yields

$$\begin{aligned}\mathbb{E}\big[\varphi\big(|g(\tau_2^k) - k_*|\big) \mid F^k\big] &\le \varphi(2\ell) \cdot \mathbb{P}\big[|g(\tau_2^k) - k_*| \le 2\ell \mid F^k\big] \\ &\quad + \varphi(C \log n) \cdot \mathbb{E}\big[|g(\tau_2^k) - k_*| \ge 2\ell \mid F^k\big] \\ &= \varphi(2\ell) + e^{-\Omega(C^{1/2} \log^{1/3} n)} \cdot \varphi(C \log n) \\ &= O(\ell) = O(C^{2/3} \log^{1/3} n),\end{aligned}$$

as desired. □

9. Reconstruction from approximately aligned strings

Recall from Proposition 2 that we can use $\widetilde{\mathbf{x}}(\tau_2^k(\widetilde{\mathbf{x}}) : \infty)$ as a trace of $\mathbf{x}(g(\tau_2^k(\widetilde{\mathbf{x}})) : \infty)$. If we had exactly $g(\tau_2^k(\widetilde{\mathbf{x}})) = k$, then the problem would be reduced to worst case reconstruction of $\mathbf{x}(k : \infty)$. This section adapts the methods of [4, 14] to handle imperfect alignment using a similar approach as [15]. We start with the following definition.

Definition 19. Let τ_2^k be as in Lemma 17. For a bit string $\mathbf{x}$ and positive integer k, let

$$V(\widetilde{\mathbf{x}}) := \widetilde{\mathbf{x}}\big(\tau_2^k(\widetilde{\mathbf{x}}) : \infty\big).$$

(We will only be concerned with $V(\widetilde{\mathbf{x}})$ in cases where $\tau_2^k(\widetilde{\mathbf{x}}) < \infty$.) For any string $\mathbf{w}$, let $\mathbf{x}_\mathbf{w}$ denote the concatenation $\mathbf{x}(0 : k) : \mathbf{w}$, and consider a trace $\widetilde{\mathbf{x}}_\mathbf{w}$ drawn from the insertion-deletion channel applied to $\mathbf{x}_\mathbf{w}$. Then, define

$$v(\mathbf{w}) := \mathbb{E}_{\mathbf{x}_\mathbf{w}}\big[V(\widetilde{\mathbf{x}}_\mathbf{w}) \mid F^k_{\mathbf{x}_\mathbf{w}}\big].$$

Note that v is a linear function of $\mathbf{w}$, so we may extend this definition to any $\mathbf{w} \in [0,1]^{\mathbb{N}}$. Finally, for $m \in \mathbb{N}$ and $\mathbf{a} \in [0,1]^{\mathbb{N}}$ set

$$\|\mathbf{a}\|_{m,\infty} := \max_{j \le m} |a_j|. \tag{10}$$

Our reconstruction is based on the following lemma.

Lemma 20. *Fix a string $\mathbf{x} \notin \Xi_{\text{bad}}$ and consider a constant $C > 0$. Suppose that $\mathbf{w} \in [0,1]^{\mathbb{N}}$ satisfies*

$$|v_j(\mathbf{w}) - v_j(\mathbf{x}(k+1 : \infty))| \le e^{-\Omega(C \log^{1/3} n)} \tag{11}$$

for all $j \le C^2 \log n$. Then, for large enough C, we must have

$$|w_0 - x_{k+1}| < \frac{1}{2}.$$

Moreover, if $\mathbf{w} = \mathbf{x}(k+1 : k+2C^2 \log n)$*, then*

$$\|v(\mathbf{w}) - v(\mathbf{x}(k+1:\infty))\|_{C^2 \log n,\infty} \leq O(n^{-2}). \tag{12}$$

The core of the proof of Lemma 20 is contained in the following lemma about bit statistics for randomly shifted strings.

Lemma 21. *Let* n *be a positive integer, and let* $\mathbf{a} \in [-1,1]^{\mathbb{N}}$ *be a sequence of real numbers for which* $a_i = 0$ *for* $i \leq n$ *but for which* $|a_{n+1}| \geq \frac{1}{2}$*. Let* S *be a random variable taking integer values between* 0 *and* $n-1$*.*

Let $\varphi(t) = te^{t/n^{2/3}}$*. Then, there exists an index* $j = O(n)$ *such that for a trace* $\widetilde{\mathbf{a}}$ *from the shifted sequence* $\theta^S(\mathbf{a})$*, we have*

$$|\mathbb{E}\big[\mathbb{E}_{\theta^S(\mathbf{a})}(\widetilde{a}_j)\big]| \geq \exp\Big(-O\big(n^{1/3} + \min_{0\leq t\leq n} \mathbb{E}\big[\varphi\big(|S-t|\big)\big]\big)\Big).$$

Let us first deduce Lemma 20 from Lemma 21.

Proof of Lemma 20. For the first claim, note that $v(\mathbf{w})$ and $v(\mathbf{x}(k+1):\infty)$ are calculated by taking expectations involving traces from $\mathbf{x}_\mathbf{w}$ and $\mathbf{x}$. Let $\widetilde{\mathbf{x}}_\mathbf{w}$ and $\widetilde{\mathbf{x}}$ denote these traces, and for purposes of our analysis, we may suppose that these traces were sampled using the same insertion/deletion pattern ω. In this case, the two events $F^k_\mathbf{x}$ and $F^k_{\mathbf{x}_\mathbf{w}}$ coincide, because they involve only the first k bits of $\mathbf{x}$ and $\mathbf{x}_\mathbf{w}$, which are identical. Thus, we will subsequently use F^k to refer to either $F^k_\mathbf{x}$ or $F^k_{\mathbf{x}_\mathbf{w}}$.

Note also that when F^k holds, we have

$$\tau_2^k(\widetilde{\mathbf{x}}) = \tau_2^k(\widetilde{\mathbf{x}}_\mathbf{w}) \quad \text{and} \quad g(\tau_2^k(\widetilde{\mathbf{x}})) = g(\tau_2^k(\widetilde{\mathbf{x}}_\mathbf{w})).$$

Accordingly, we will write $\tau = \tau_2^k(\widetilde{\mathbf{x}}) = \tau_2^k(\widetilde{\mathbf{x}}_\mathbf{w})$ when conditioning on F^k. Let S be a random variable with the same distribution as $g(\tau) - k + C\log n$ conditioned on F^k. Then, by Proposition 2, we have

$$\begin{aligned} v_j(\mathbf{w}) - v_j(\mathbf{x}(k+1:\infty)) &= \mathbb{E}\big[\widetilde{x}_{\mathbf{w},\tau+j+1} - \widetilde{x}_{\tau+j+1} \mid F^k\big] \\ &= \mathbb{E}_{\theta^{k-C\log n+S(\mathbf{x}_\mathbf{w})}}[\widetilde{x}_{j+1}] - \mathbb{E}_{\theta^{k-C\log n+S(\mathbf{x})}}[\widetilde{x}_{j+1}]. \end{aligned} \tag{13}$$

We are now in a position to apply Lemma 21. Take

$$\mathbf{a} = \mathbf{x}_\mathbf{w}(k - C\log n : \infty) - \mathbf{x}(k - C\log n : \infty).$$

Note that $a_i = 0$ for $i \leq C\log n$, and suppose that $|a_{C\log n+1}| \geq \frac{1}{2}$. Recall that by Lemma 17, we have

$$\mathbb{E}\big[\varphi\big(|g(\tau) - k_*|\big) \mid F^k\big] \leq O(C^{2/3}\log^{1/3} n).$$

Then, Lemma 21 gives some index $j = O(C\log n)$ for which

$$\begin{aligned} \big|\mathbb{E}\big[\mathbb{E}_{\theta^S(\mathbf{a})}(\widetilde{a}_j)\big]\big| &\geq e^{-O(C^{1/3}\log^{1/3} n) - O(C^{2/3}\log^{1/3} n)} \\ &= e^{-O(C^{2/3}\log^{1/3} n)}. \end{aligned}$$

Substituting into (13), this is a contradiction of (11) for large enough C. We conclude that if (11) holds, then we must have $|w_0 - x_{k+1}| = |a_{C\log n+1}| > \frac{1}{2}$.

For the second claim, let $\mathbf{w} = \mathbf{x}(k+1 : k+2C^2\log n)$. Note that if on the event F^k we have $\|V(\widetilde{\mathbf{x}}_{\mathbf{w}}) - V(\widetilde{\mathbf{x}})\|_{C^2\log n,\infty} \neq 0$, then it means that there had to have been at least $C^2\log n$ more deletions than insertions in the interval $[k, k+2C^2\log n]$. Thus,

$$\mathbb{P}\big(\{\|V(\widetilde{\mathbf{x}}_{\mathbf{w}}) - V(\widetilde{\mathbf{x}})\|_{C^2\log n,\infty} \neq 0\} \cap F^k\big) \leq e^{-\Omega(C^2\log n)} = n^{-\Omega(C^2)},$$

and by Lemma 17, we have

$$\mathbb{P}\big(\|V(\widetilde{\mathbf{x}}_{\mathbf{w}}) - V(\widetilde{\mathbf{x}})\|_{C^2\log n,\infty} \neq 0 \mid F^k\big) \leq \frac{n^{-\Omega(C^2)}}{\mathbb{P}(F^k)} \leq n^{-\Omega(C^2)}.$$

Since the entries of $V(\widetilde{\mathbf{x}})$ and $V(\widetilde{\mathbf{x}}_{\mathbf{w}})$ are all 0 or 1, this proves the second claim upon taking expectations. □

9.1. Proof of Lemma 21. The remainder of this section is devoted to proving Lemma 21.

Lemma 22. *Let S be a bounded $\mathbb{N}$-valued random variable. Let $\mathbf{a} = (a_0, a_1, \dots) \in [-1,1]^{\mathbb{N}}$, and let $\widetilde{\mathbf{a}}$ be the output from the insertion-deletion channel with deletion (resp. insertion) probability q (resp. q'), applied to the randomly shifted string $\theta^S(\mathbf{a})$. Let $\phi_1(w) = pw + q$, $\phi_2(w) = \frac{p'w}{1-q'w}$, and $\sigma(s) = \mathbb{P}[S = s]$ for $s \in \mathbb{N}$. Define*

$$P(z) := \sum_{s=0}^{d} \sigma(s) z^s, \quad Q(z) := \sum_{j=0}^{\infty} a_j z^j.$$

Then, for any $|w| < 1$,

$$\mathbb{E}\Big[\sum_{j\geq 0} \widetilde{a}_j w^j\Big] = \frac{pp'}{1-q'\phi_1(w)} \cdot P\left(\frac{1}{\phi_2\circ\phi_1(w)}\right) \cdot Q(\phi_2\circ\phi_1(w)). \tag{14}$$

Proof. Recall the construction of $\widetilde{\mathbf{x}}$ from $\mathbf{x}$ given in Section 3, where we first insert a geometric number (minus one) bits before each bit of $\mathbf{x}$ and then delete each bit independently with probability q. From this description we see that we can sample $\widetilde{\mathbf{a}}$ by first setting $\widetilde{\mathbf{a}}^{(2)} = \theta^S(\mathbf{a})$, then letting $\mathbf{a}^{(3)}$ be the string we get when sending $\mathbf{a}^{(2)}$ through the insertion channel with insertion probability q' (and no deletions), and finally obtain $\widetilde{\mathbf{a}}$ by sending $\widetilde{\mathbf{a}}^{(3)}$ through the deletion channel with

deletion probability q (and no insertions). Three elementary generating function manipulations (see, respectively, [15, Lemma 4.2], [14, Lemma 5.2 and 2.1]) give

$$\mathbb{E}\Big[\sum_{j\geq 0} a_j^{(2)} w^j\Big] = P(w^{-1})Q(w), \quad \mathbb{E}\Big[\sum_{j\geq 0} a_j^{(3)} w^j \,\Big|\, \mathbf{a}^{(2)}\Big] = \frac{\phi_2(w)}{w}\sum_{j\geq 0} a_j^{(2)}\phi_2(w)^j,$$

$$\mathbb{E}\Big[\sum_{j\geq 0} \widetilde{a}_j w^j \,\Big|\, \mathbf{a}^{(3)}\Big] = p\sum_{j\geq 0} a_j^{(3)}\phi_1(w)^j.$$

Combining these identifies we get (14):

$$\begin{aligned}\mathbb{E}\Big[\sum_{j\geq 0} \widetilde{a}_j w^j\Big] &= p\mathbb{E}\Big[\sum_{j\geq 0} a_j^{(3)}\phi_1(w)^j\Big] \\ &= p\frac{\phi_2\circ\phi_1(w)}{\phi_1(w)}\mathbb{E}\Big[\sum_{j\geq 0} a_j^{(2)}\big(\phi_2\circ\phi_1(w)\big)^j\Big] \\ &= p\frac{\phi_2\circ\phi_1(w)}{\phi_1(w)}P\left(\frac{1}{\phi_2\circ\phi_1(w)}\right)Q(\phi_2\circ\phi_1(w)). \qquad \square\end{aligned}$$

The following result is Corollary 3.2 of [2] with $M = 1$, $a = \ell$ and $c_1 = C_{\mathrm{BE}}$, observing that the class of polynomials whose coefficients have modulus at most 1 are in their class $\mathcal{K}_1^1$ and that their statement $\mathcal{K}_M := \mathcal{K}_M^0$ after their definition of $\mathcal{K}_M^\mu$ should be ignored in favor of the correct statement $\mathcal{K}_M := \mathcal{K}_M^1$ occurring in their Corollary 3.2.

Lemma 23 (Borwein and Erdélyi (1997)). *There is a universal constant C_{BE} such that for any polynomial f satisfying $|f(0)| = 1$ and whose coefficients have modulus at most 1, and for any arc α of the unit circle whose angular length is denoted $s \in (0, 2\pi)$, we have*

$$\sup_{z\in\alpha} |f(z)| \geq e^{-C_{\mathrm{BE}}/s}.$$

Proof of Lemma 21. Let

$$t^* = \operatorname*{arg\,min}_{0\leq t\leq n} \mathbb{E}\big[\varphi\big(|S-t|\big)\big].$$

Define the quantities

$$\begin{aligned}\mathbf{a} &= \mathbf{x}^{(1)} - \mathbf{x}^{(2)} \in \{-1, 0, 1\}^{\mathbb{N}}, \\ L &= \max\big(n^{1/3}, \mathbb{E}\big[\varphi\big(|S-t^*|\big)\big]\big), \\ \rho &= 1 - 1/L^2.\end{aligned}$$

We first establish a general bound for Möbius transformations appearing in Lemma 22.

Claim. There is a constant $c_2 \in (0, 1/20)$ depending only on q, q' such that if $|\arg(z)| \le c_2/L$, $|z| = 1$, and $w = \phi_1^{-1}(\phi_2^{-1}(\rho \cdot z))$, then $|w| \le 1 - c_2/L^2$.

Proof of the claim. Observe that ϕ_2 (resp. ϕ_1) is a Möbius transformation mapping $\mathbb{D}$ to a smaller disk which is contained in $\overline{\mathbb{D}}$, which is tangent to $\partial\mathbb{D}$ at 1, and which maps $\mathbb{R}$ to $\mathbb{R}$. In particular, defining $\Psi := \phi_1^{-1} \circ \phi_2^{-1}$, we get by linearizing the map around $z = 1$ that $\Psi(1 + \widetilde{z}) = 1 + a\widetilde{z} + O(|\widetilde{z}|^2)$ for $a > 1$ depending only on q, q'. Writing $z = e^{i\theta}$, we have

$$\begin{aligned} w &= \Psi(\rho e^{i\theta}) \\ &= 1 + a(\rho e^{i\theta} - 1) + O\big(|\rho e^{i\theta} - 1|^2\big) \\ &= 1 + a\big((1 - L^{-2})(1 + i\theta) - 1\big) + O(\theta^2 + L^{-4}) \\ &= 1 + a(-L^{-2} + i\theta) + O(\theta^2 + L^{-4}), \end{aligned}$$

so $|w| < 1 - c_2/L^2$ when $c_2 = c_2(q, q')$ is sufficiently small, and the claim is proved. □

Let P and Q be as in Lemma 22, and write $\widetilde{Q}(z) := z^{-n-1}Q(z)$, where our assumption that $\mathbf{x}$ and $\mathbf{x}'$ first differ in the $(n+1)$-th bit implies that $\widetilde{Q}(z)$ is a polynomial with $|\widetilde{Q}(0)| \ge 1/2$.

Observe that $z \mapsto \widetilde{Q}(\rho \cdot z)$ has coefficients of modulus at most 1, so we may apply Lemma 23 to find $z_0 = e^{i\theta}$ with $|\theta| \le c_2/L$ such that $|\widetilde{Q}(\rho z_0)| \ge e^{-C_{\mathrm{BE}}L/c_2}$. By definition of c_2, we see that $w_0 := \Psi(\rho \cdot z_0)$ satisfies $|w_0| \le 1 - c_2/L^2$. An illustration of the points z_0 and w_0 is given in Figure 7.

We next show that

$$\left|P\left(\frac{1}{\rho z_0}\right)\right| \ge \frac{1}{2}. \tag{15}$$

To see this, define $\widetilde{P}(z) = z^{-t^*}P(z)$, which is an analytic function in the right half-plane. For all z in the right half-plane satisfying $1 \le |z| \le \rho^{-1}$, differentiating $\widetilde{P}$ gives

$$\begin{aligned} |\widetilde{P}'(z)| &= \left|\sum_{j=0}^{d}(j - t^*)\sigma(j)z^{j-t^*-1}\right| \le \sum_{j=0}^{d}|j - t^*| \cdot \sigma(j) \cdot |z|^{j-t^*-1} \\ &\le \mathbb{E}\big[|S - t^*| \cdot \rho^{-|S-t^*|}\big] \le 4\mathbb{E}\big[|S - t^*| \cdot e^{|S-t^*|/L^2}\big] \\ &\le 4\mathbb{E}\big[|S - t^*| \cdot e^{|S-t^*|/n^{2/3}}\big] = 4\mathbb{E}\big[\varphi\big(|S - t^*|\big)\big] \le 4L. \end{aligned}$$

We also have

$$|\rho^{-1}z_0^{-1} - 1| = \rho^{-1}|1 - \rho z_0| \le |z_0 - 1| + \rho^{-1}(1 - \rho) \le \frac{c_2}{L} + \frac{2}{L^2}.$$

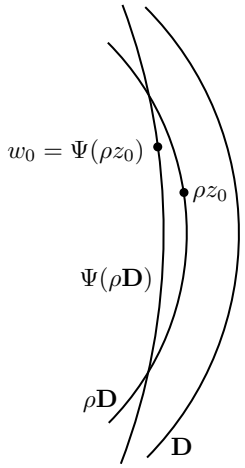

Figure 7. Illustration of the points $z_0, w_0 \in \mathbb{C}$ defined in the proof of Lemma 21. We first choose $z_0 = e^{i\theta}$ for $|\theta| \le c_2/L$, such that $|\widetilde{Q}(\rho \cdot z_0)|$ is bounded from below. Then we observe that $|w_0| < 1 - c_2/L^2$, which helps us to bound the modulus of $\mathbb{E}[\sum_{j\ge 0} \widetilde{a}_j w_0^j]$ from below.

Therefore, for all sufficiently large m,

$$\begin{aligned} |P(\rho^{-1}z_0^{-1})| &= \rho^{-\mathbb{E}S}|\widetilde{P}(\rho^{-1}z_0^{-1})| \ge 1 - |\widetilde{P}(\rho^{-1}z_0^{-1}) - 1| \\ &= 1 - \left| \int_1^{\rho^{-1}z_0^{-1}} \widetilde{P}'(z)\, dz \right| \ge 1 - |\rho^{-1}z_0^{-1} - 1| \cdot 4L \\ &\ge 1 - \left(\frac{c_2}{L} + \frac{2}{L^2} \right) \cdot 4L \ge \frac{1}{2}, \end{aligned}$$

proving (15).

Also note that the following quantity is bounded from below by a constant depending only on p and p':

$$\frac{pp'}{|1 - q\phi_1(w_0)|}.$$

Using Lemma 22 and the above estimates, it follows that:

$$\begin{aligned} \left| \mathbb{E}\left[\sum_{j\ge 0} \widetilde{a}_j w_0^j \right] \right| &\ge \frac{pp'}{|1 - q\phi_1(w_0)|} \cdot \left| P\left(\frac{1}{\rho z_0} \right) \right| \cdot \rho^{n+1} \cdot |\widetilde{Q}(\rho z_0)| \qquad (16) \\ &\ge \frac{pp'}{|1 - q\phi_1(w_0)|} \cdot \frac{1}{2} \left(1 - \frac{1}{L^2} \right)^{n+1} e^{-C_{\mathrm{BE}} L/c_2} \\ &\ge e^{-O(L) - O(n/L^2)} = e^{-O(L)}. \end{aligned}$$

Since $|w_0| \le 1 - c_2/L^2$, for any $C_{\text{fwd}} > 1$,

$$\left| \sum_{j \ge C_{\text{fwd}} n} \mathbb{E}[\widetilde{a}_j] w_0^j \right| \le \left| \sum_{j \ge C_{\text{fwd}} n} \left(1 - \frac{c_2}{L^2}\right)^j \right| \le L^2 c_2^{-1} e^{-C_{\text{fwd}} L/c_2} = e^{-\Omega(C_{\text{fwd}} L)}. \tag{17}$$

Combining (16) and (17), by taking C_{fwd} sufficiently large (depending only on q and q'), we have

$$\mathbb{E}\left[\sum_{j=0}^{\lceil C_{\text{fwd}} n \rceil - 1} |\widetilde{a}_j w_0^j| \right] \ge \left| \mathbb{E}\left[\sum_{j=0}^{\lceil C_{\text{fwd}} n \rceil - 1} \widetilde{a}_j w_0^j \right] \right| \ge e^{-O(L)}.$$

It follows by the pigeonhole principle that there is a $j < C_{\text{fwd}} n$ for which

$$|\mathbb{E}[\widetilde{a}_j]| \ge |\mathbb{E}[\widetilde{a}_j] w_0^j| \ge \left(2\lceil C_{\text{fwd}} n \rceil\right)^{-1} e^{-O(L)} = e^{-O(L)},$$

as desired. □

10. Proof of Theorem 1

For each k, we will describe how to reconstruct x_{k+1} assuming we know $\mathbf{x}(0:k)$. This will involve applying the rules τ_1^k and τ_2^k. For purposes of analyzing the computational cost, we also assume that the results of τ_1^i and τ_2^i have been saved for all $i < k$. We assume throughout the proof that $k \ge 2C \log n$; the case $k < 2C \log n$ is easier since we can reconstruct the first $2C \log n$ bits directly by the results of Section 9 (with no shift).

We will use $n^{o(1)}$ traces to do the reconstruction of x_{k+1} with success probability at least $1 - n^{-2}$. Thus, even if we reuse the sampled traces at each step, by a union bound, reconstruction will succeed for all bits with high probability. The computational cost for reconstructing this bit will also be $n^{o(1)}$ with probability at least $1 - n^{-2}$, so that the overall computational cost is $n^{1+o(1)}$ with high probability.

10.1. Computing τ_1^k and τ_2^k in $n^{o(1)}$ time. For each trace $\widetilde{\mathbf{x}}$, we want to first find τ_1^k and then find τ_2^k. If we wanted to determine τ_1^k with probability 1 we would need to perform a sliding window of tests $T_{\ell,\lambda}(\mathbf{x}(k-\ell+1:k), \widetilde{\mathbf{x}}(k'-\ell+1:k'))$, where k' potentially ranges from $\ell - 1$ to $2n$. Each test only takes polylog(n) time to perform, but as described, we are performing $\Theta(n)$ tests, which exceeds our goal of $n^{o(1)}$.

To save on computation, we will only find estimates for τ_1^k and τ_2^k, which are correct with high probability. Observe that if previously we had τ_1^i close to $f(i)$ for some i, then to find a (non-spurious) match, we need only test bits in $\widetilde{\mathbf{x}}$ close to or after position $f(i)$. To carry out this argument formally, we begin with the following definition.

Definition 24. Suppose that $\mathbf{x} \notin \Xi_{\text{bad}}$ and $\widetilde{\mathbf{x}}$ is a trace from $\mathbf{x}$. We say that $\widetilde{\mathbf{x}}$ is *progressively alignable* if the following hold:

1. For any $i \in \{1, \dots, n\}$ such that $\tau_1^i < \infty$, we have $d(i, \tau_1^i) \le \log^2 n$ (recall (2)).
2. For any t with $0 \le t \le n - e^{\log^{1/2} n}$, there is some $i \in [t, t + e^{\log^{1/2} n}]$ such that $\tau_1^i < \infty$.

The thresholds in the above definition have been chosen so that traces are progressively alignable with very high probability, as shown by the next lemma.

Lemma 25. *Suppose that* $\mathbf{x} \notin \Xi_{\text{bad}}$ *and* $\widetilde{\mathbf{x}}$ *is a trace from* $\mathbf{x}$. *Then* $\widetilde{\mathbf{x}}$ *is progressively alignable with probability at least* $1 - O(n^{-2})$.

Proof. Let $\ell = C \log^{5/3} n$ and $\lambda = C \log^{2/3} n$ as in Lemma 16. To see that the first property occurs with probability $1 - O(n^{-2})$, we may use the same argument as in the proof of Lemma 16, with the only modification being that in the definitions of the events E' and F, the quantities $\frac{1}{10} C \log n$ and $\frac{1}{20} C \log n$ should be changed to $\log^2 n$ and $\frac{1}{2} \log^2 n$, respectively.

For the second property, note that since $\mathbf{x}$ is coarsely well-behaved, each of its substrings $\mathbf{w}$ of length ℓ exhibits robust bias at scale λ. Thus, we may apply Lemma 5 to conclude that with probability at least $e^{-O(\ell/\lambda^2)} \ge e^{-O(\log^{1/3} n)}$, the part of the trace coming from $\mathbf{w}$ contains a match for the test $T_{\ell,\lambda}(\mathbf{w}, \cdot)$.

In any interval $[t, t + e^{\log^{1/2} n}]$, we have at least $e^{\Omega(\log^{1/2} n)}$ disjoint intervals of length ℓ, and each of these has independently a $e^{-O(\log^{1/3} n)}$ chance of producing a match. The chance of not having a single match over this whole interval is therefore at most

$$\left(1 - e^{-O(\log^{1/3} n)}\right)^{e^{\Omega(\log^{1/2} n)}} = \exp\left(- \frac{e^{\Omega(\log^{1/2} n)}}{e^{O(\log^{1/3} n)}} \right) \le n^{-3}.$$

Taking a union bound over all t establishes the second property. □

Let us now specify our algorithm for computing (an estimate of) τ_1^k, which is only guaranteed to give the right value (i.e., the value defined in Lemma 16) if the trace is progressively alignable, but also costs only $n^{o(1)}$ operations. The algorithm is to first look for $i \in [k - 2e^{\log^{1/2} n}, k - e^{\log^{1/2} n}]$ such that $\tau_1^i < \infty$. Then, we evaluate τ_1^k as

$$\inf \left\{ \tau_1^i \le k' \le \tau_1^i + 3e^{\log^{1/2} n} : T_{\ell,\lambda}(\mathbf{x}(k - \ell : k), \widetilde{\mathbf{x}}(k' - \ell : k')) = 1 \right\}.$$

As long as the trace is progressively alignable and our stored values of τ_1^i are correct, this gives us the right answer, i.e., our estimate for τ_1^k is equal to the true value of τ_1^k. The above test takes only $e^{O(\log^{1/2} n)} = n^{o(1)}$ operations, and by Lemma 25 and a union bound, the overall probability of getting at least one wrong result is less than n^{-1}.

Next, in order to calculate τ_2^k, it is necessary to identify the "good" interval I in Definition 14. For a given interval I, it can be easily checked whether $\mathbf{x}(I)$ has robust bias at scale λ, but it is not as straightforward to explicitly calculate

$$\mathbb{P}_{\mathbf{x}}\big(\mathcal{Q}_{\ell,\lambda}(I,J)\big).$$

However, we can estimate this probability to high accuracy by Monte-Carlo simulation. Since the relevant interval J is only of logarithmic size, each simulated sample can be produced in $n^{o(1)}$ time. By Hoeffding's inequality, $e^{O(C^2\log^{1/3}n)}$ samples are enough get an accuracy of $e^{-\Omega(C^2\log^{1/3}n)}$ with probability $1-n^{-2}$. This level of accuracy is small compared to the probability bound in Definition 14, so it is accurate enough for all of our analysis to carry through.

10.2. Determining the next bit. Our algorithm will work by sampling $N := e^{C^2\log^{1/3}n}$ traces. Let N_1 denote the number of traces for which $\tau_2^k < \infty$, and let us number these traces $\widetilde{\mathbf{x}}^{(1)},\dots,\widetilde{\mathbf{x}}^{(N_1)}$. Note that by Lemma 17, we have $N_1 = e^{\Omega(C^2\log^{1/3}n)}$ with probability at least

$$1 - e^{-\Omega\left(e^{\Omega(\log^{1/3}n)}\right)} \geq 1 - n^{-2}.$$

Our reconstruction strategy is based on Lemma 20. Let $m := C^2\log n$. Recall from (10) the notation $\|\mathbf{a}\|_{m,\infty}$ for any $\mathbf{a} \in [0,1]^{\mathbb{N}}$. Suppose we are able to (approximately) solve the following minimization problem (where v is as in Definition 19):

$$\min_{\mathbf{w}\in[0,1]^{2m}} \|v(\mathbf{w}) - v(\mathbf{x}(k+1:\infty))\|_{m,\infty}. \tag{18}$$

Then, by Lemma 20, we could recover x_{k+1} by rounding w_0 to the nearest integer (either 0 or 1). Note that (18) is a linear program with $O(\log n)$ variables and constraints, and so it can be solved in polylog(n) time by e.g. [9].[2]

However, two issues arise: we do not have direct access to the quantity $v(\mathbf{x}(k+1:\infty))$, nor are we able to evaluate quantities like $v(\mathbf{w})$ directly.

To address the first issue, we can estimate $v(\mathbf{x}(k+1:\infty))$ using our traces. Consider the empirical mean

$$\widehat{v} = \frac{1}{N_1}\sum_{i=1}^{N_1} V\big(\widetilde{\mathbf{x}}^{(i)}\big).$$

[2]More precisely, in [9, Section 1.6] it is proved that if L is the number of bits in the input and $2m$ is the number of variables then the problem can be solved in time $O(m^{3.5}L^2\log L\log\log L)$. We can round the coefficients of v to the nearest multiple of $m^{-1}e^{-K\log^{1/3}n} = e^{-\Theta(K\log^{1/3}n)}$. For $K \gg 1$, this gives $L = O(mK\log^{1/3}n)$, so we can find a solution which differs from the optimal solution by $e^{-\Omega(K\log^{1/3}n)}$ and has running time of order $m^{3.5}L^2\log L\log\log L \ll \log^7 n$.

This is a sum of i.i.d. vectors with entries in $[0,1]$ whose expectation is the desired vector $v(\mathbf{x}(k+1:\infty))$. Thus, by Hoeffding's inequality, we have with probability at least $1-n^{-2}$ that

$$\|\widehat{v} - v(\mathbf{x}(k+1:\infty))\|_{m,\infty} \le e^{-\Omega(C^2 \log^{1/3} n)}.$$

To address the second issue, we will estimate $v(\mathbf{w})$ by Monte-Carlo simulation. Let $\mathbf{e}_i$ denote the vector with 1 in the i-th entry and 0 elsewhere; we first estimate the quantities $v(\mathbf{e}_i)$. Since we already know $\mathbf{x}(0:k)$, we can simulate drawing a trace from $\mathbf{x}_{\mathbf{e}_i}$ (Definition 19) and compute τ_2^k. However, this once again takes $O(n)$ time, because we have to scan through the whole string.

Instead, we sample a trace $\widetilde{\mathbf{x}}'_{\mathbf{e}_i}$ from the shortened string $\mathbf{x}_{\mathbf{e}_i}(k-\log^2 n:\infty)$ and evaluate $V(\widetilde{\mathbf{x}}'_{\mathbf{e}_i})$. Note that the trace $\widetilde{\mathbf{x}}'_{\mathbf{e}_i}$ is equivalent to removing the first $f(k-\log^2 n)$ bits from a trace $\widetilde{\mathbf{x}}_{\mathbf{e}_i}$ of the full string $\mathbf{x}_{\mathbf{e}_i}$. Coupling $\widetilde{\mathbf{x}}'_{\mathbf{e}_i}$ and $\widetilde{\mathbf{x}}_{\mathbf{e}_i}$ in this way, $V(\widetilde{\mathbf{x}}'_{\mathbf{e}_i})$ is usually the exact same as $V(\widetilde{\mathbf{x}}_{\mathbf{e}_i})$; the only way they can differ is if $g(\tau_1^k(\widetilde{\mathbf{x}}_{\mathbf{e}_i})-\ell) \le k-\log^2 n$, which by Lemma 16 happens with probability $O(n^{-2})$. Note that we do not know that $\mathbf{x}_{\mathbf{e}_i} \notin \Xi_{\text{bad}}$ (in fact, we typically have $\mathbf{x}_{\mathbf{e}_i} \in \Xi_{\text{bad}}$). However, we have $\mathbf{x}_{\mathbf{e}_i}(0:k) = \mathbf{x}(0:k)$, and the initial part of the string is the most relevant part when we do the alignment, since the string we use to align is chosen as a substring of $\mathbf{x}_{\mathbf{e}_i}(0:k) = \mathbf{x}(0:k)$. Furthermore, adding the string $\mathbf{e}_i$ at the end will cause false positives with very small probability since the bit statistics of this string are very different from those of the string we use to align. It follows that

$$\begin{aligned}
&\big\|\mathbb{E}\big[V(\widetilde{\mathbf{x}}'_{\mathbf{e}_i}) \mid \tau_2^k(\widetilde{\mathbf{x}}'_{\mathbf{e}_i}) < \infty\big] - \mathbb{E}\big[V(\widetilde{\mathbf{x}}_{\mathbf{e}_i}) \mid \tau_2^k(\widetilde{\mathbf{x}}_{\mathbf{e}_i}) < \infty\big]\big\|_{m,\infty} \\
&= \left\| \frac{\mathbb{E}\big[V(\widetilde{\mathbf{x}}_{\mathbf{e}_i})\mathbf{1}_{\tau_2^k(\widetilde{\mathbf{x}}_{\mathbf{e}_i})<\infty}\big] + O(n^{-2})}{\mathbb{P}\big[\tau_2^k(\widetilde{\mathbf{x}}_{\mathbf{e}_i}) < \infty\big] + O(n^{-2})} - \frac{\mathbb{E}\big[V(\widetilde{\mathbf{x}}_{\mathbf{e}_i})\mathbf{1}_{\tau_2^k(\widetilde{\mathbf{x}}_{\mathbf{e}_i})<\infty}\big]}{\mathbb{P}\big[\tau_2^k(\widetilde{\mathbf{x}}_{\mathbf{e}_i}) < \infty\big]} \right\|_{m,\infty} = O(n^{-1.5}).
\end{aligned}$$

We can also see from Lemma 17 that

$$\begin{aligned}
&\big\|v(\mathbf{e}_i) - \mathbb{E}\big[V(\widetilde{\mathbf{x}}_{\mathbf{e}_i}) \mid \tau_2^k(\widetilde{\mathbf{x}}_{\mathbf{e}_i}) < \infty\big]\big\|_{m,\infty} \\
&\qquad = \big\|\mathbb{E}\big[V(\widetilde{\mathbf{x}}_{\mathbf{e}_i}) \mid F^k\big] - \mathbb{E}\big[V(\widetilde{\mathbf{x}}_{\mathbf{e}_i}) \mid \tau_2^k(\widetilde{\mathbf{x}}_{\mathbf{e}_i}) < \infty\big]\big\|_{m,\infty} \le O(n^{-1.5}).
\end{aligned}$$

Thus, by performing this simulation $e^{C^2 \log^{1/3} n}$ times and averaging the results, we are able to provide an estimate $v'(\mathbf{e}_i)$ of $v(\mathbf{e}_i)$ to within $e^{-\Omega(C^2 \log^{1/3} n)}$ error with probability at least $1-O(n^{-2})$. We can extend this linearly to all possible inputs $\mathbf{w} \in [0,1]^{2m}$ by setting

$$v'(\mathbf{w}) = \sum_{i=1}^{2m} w_i v'(\mathbf{e}_i),$$

and we see that overall $\|v'(\mathbf{w}) - v(\mathbf{w})\|_{m,\infty} \le e^{-\Omega(C^2 \log^{1/3} n)}$.

We can then solve the modified optimization problem

$$\min_{\mathbf{w}\in[0,1]^{2m}} \|v'(\mathbf{w}) - \widehat{v}\|_{m,\infty},$$

which is an approximation of our original problem. With probability at least $1-n^{-2}$, the objective function in the above problem is within $e^{-\Omega(C^2\log^{1/3} n)}$ of the objective in (18). In this case, as long as C is large enough, our minimizer $\mathbf{w}^*$ will satisfy the hypothesis of Lemma 20, and so we can correctly extract the next bit x_{k+1} as the closer of 0 or 1 to w_1^*. This completes our analysis and establishes Theorem 1.

References

[1] T. Batu, S. Kannan, S. Khanna, and A. McGregor, Reconstructing strings from random traces, in *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 910–918, ACM, New York, 2004. Zbl 1318.68206 MR 2290981

[2] P. Borwein and T. Erdélyi, Littlewood-type problems on subarcs of the unit circle, *Indiana Univ. Math. J.*, **46** (1997), no. 4, 1323–1346. Zbl 0930.30005 MR 1631600

[3] Z. Chase, New Lower Bounds for Trace Reconstruction, 2019. arXiv:1905.03031

[4] A. De, R. O'Donnell, and R. Servedio, Optimal mean-based algorithms for trace reconstruction, in *STOC'17–Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, 1047–1056, ACM, New York, 2017. Zbl 1369.68202 MR 3678250

[5] N. Holden and R. Lyons, Lower bounds for trace reconstruction, *Ann. Appl. Probab.*, **30** (2020), no. 2, 503–525. MR 4108114

[6] N. Holden, R. Pemantle, and Y. Peres, Subpolynomial trace reconstruction for random strings and arbitrary deletion probability, in *Proceedings of the 31st Conference On Learning Theory (COLT)*, 1799–1840, Proceedings of Machine Learning Research, 75, PMLR, 2018.

[7] T. Holenstein, M. Mitzenmacher, R. Panigrahy, and U. Wieder, Trace reconstruction with constant deletion probability and related results, in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 389–398, ACM, New York, 2008. Zbl 1192.94064 MR 2487606

[8] S. Kannan and A. McGregor, More on reconstructing strings from random traces: insertions and deletions, *Proceedings of the International Symposium on Information Theory (ISIT)*, 297–301, IEEE, 2005.

[9] N. Karmarkar, A new polynomial-time algorithm for linear programming, in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 302–311, ACM, New York, 1984.

[10] V. I. Levenshtein, Efficient reconstruction of sequences, *IEEE Trans. Inform. Theory*, **47** (2001), no. 1, 2–22. Zbl 1029.94019 MR 1819952

[11] V. I. Levenshtein, Efficient reconstruction of sequences from their subsequences or supersequences, *J. Combin. Theory Ser. A*, **93** (2001), no. 2, 310–332. Zbl 0992.68155 MR 1805300

[12] A. McGregor, E. Price, and S. Vorotnikova, Trace reconstruction revisited, in *Algorithms–ESA 2014*, 689–700, Lecture Notes in Comput. Sci., 8737, Springer, Heidelberg, 2014. Zbl 1425.68470 MR 3253172

[13] M. Mitzenmacher, A survey of results for deletion channels and related synchronization channels, *Probab. Surv.*, **6** (2009), 1–33. Zbl 1189.94058 MR 2525669

[14] F. Nazarov and Y. Peres, Trace reconstruction with $\exp(O(n^{1/3}))$ samples, in *STOC'17–Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, 1042–1046, ACM, New York, 2017. Zbl 1370.68087 MR 3678249

[15] Y. Peres and A. Zhai, Average-case reconstruction for the deletion channel: subpolynomially many traces suffice, in *58th Annual IEEE Symposium on Foundations of Computer Science–FOCS 2017*, 228–239, IEEE Computer Soc., Los Alamitos, CA, 2017. MR 3734232

[16] K. Viswanathan and R. Swaminathan, Improved string reconstruction over insertion-deletion channels, in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 399–408, ACM, New York, 2008. Zbl 1192.94072 MR 2487607

Received 07 January, 2020

N. Holden, Institute for Theoretical Studies, ETH Zürich,
Clausiusstrasse 47, 8092 Zürich, Switzerland
E-mail: ninahold@gmail.com

R. Pemantle, David Rittenhouse Laboratories, University of Pennsylvania,
209 South 33rd Street, Philadelphia, PA 19104, USA
E-mail: pemantle@math.upenn.edu

Y. Peres, Microsoft Research, 14820 NE 36th St,
Redmond, WA 98052, USA
E-mail: yuval@yuvalperes.com

A. Zhai, Stanford University, 450 Jane Stanford Way,
Stanford, CA 94305, USA
E-mail: alexlinzhai@gmail.com