Trace Reconstruction from Local Statistical Queries

Xi Chen
Columbia University, New York, NY, USA

Anindya De
University of Pennsylvania, Philadelphia, PA, USA

Chin Ho Lee
North Carolina State University, Raleigh, NC, USA

Rocco A. Servedio
Columbia University, New York, NY, USA

Abstract

The goal of *trace reconstruction* is to reconstruct an unknown n-bit string x given only independent random *traces* of x, where a random trace of x is obtained by passing x through a deletion channel. A *Statistical Query* (SQ) algorithm for trace reconstruction is an algorithm which can only access statistical information about the distribution of random traces of x rather than individual traces themselves. Such an algorithm is said to be ℓ*-local* if each of its statistical queries corresponds to an ℓ-junta function over some block of ℓ consecutive bits in the trace. Since several – but not all – known algorithms for trace reconstruction fall under the local statistical query paradigm, it is interesting to understand the abilities and limitations of local SQ algorithms for trace reconstruction.

In this paper we establish nearly-matching upper and lower bounds on local Statistical Query algorithms for both worst-case and average-case trace reconstruction. For the worst-case problem, we show that there is an $\tilde{O}(n^{1/5})$-local SQ algorithm that makes all its queries with tolerance $\tau \geq 2^{-\tilde{O}(n^{1/5})}$, and also that any $\tilde{O}(n^{1/5})$-local SQ algorithm must make some query with tolerance $\tau \leq 2^{-\tilde{\Omega}(n^{1/5})}$. For the average-case problem, we show that there is an $O(\log n)$-local SQ algorithm that makes all its queries with tolerance $\tau \geq 1/\mathrm{poly}(n)$, and also that any $O(\log n)$-local SQ algorithm must make some query with tolerance $\tau \leq 1/\mathrm{poly}(n)$.

2012 ACM Subject Classification Mathematics of computing → Probabilistic inference problems

Keywords and phrases trace reconstruction, statistical queries, algorithmic statistics

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2024.52

Category RANDOM

Funding *Xi Chen*: Supported by NSF grants CCF-1703925, IIS-1838154, CCF-2106429 and CCF-2107187.
Anindya De: Supported by NSF grants CCF-1926872, CCF-1910534 and CCF-2045128.
Chin Ho Lee: Supported by Madhu Sudan's and Salil Vadhan's Simons Investigator Awards while at Harvard University.
Rocco A. Servedio: Supported by NSF grants IIS-1838154, CCF-2106429, CCF-2211238 and by the Simons Collaboration on Algorithms and Geometry.

1 Introduction

In the *trace reconstruction* problem, the goal is to reconstruct an unknown string $x \in \{0,1\}^n$ given access to independent random *traces* of x, where a random trace of x is a string obtained by passing x through a deletion channel that independently deletes each bit with probability δ and concatenates the surviving bits. Trace reconstruction has been a well-studied problem since the early 2000s [30, 29, 2], and some combinatorial variants of the problem were already considered in the 1970s [26]. Over the past decade, a wide range of algorithmic results and lower bounds have been established for many variants of the trace reconstruction problem,

including worst-case [32, 19, 36, 22, 10, 11], average-case [37, 23, 24, 38], and smoothed analysis [13] versions, the low deletion rate regime [12], approximate trace reconstruction [18, 8, 9, 14], coded trace reconstruction [16, 6], variants in which different bits of the source string have different deletion probabilities [21], circular trace reconstruction [35], trace reconstruction on trees [17, 5], population recovery variants [1, 33, 34], connections to other problems such as mixture distribution learning [28], and more [20, 39].

The original, and arguably most fundamental, versions of the problem are the "worst-case" and "average-case" versions with constant deletion rate $\delta \in (0,1)$. In the worst-case problem the source string x is an arbitrary (worst-case) element of $\{0,1\}^n$, and in the average-case problem the source string x is selected uniformly at random from $\{0,1\}^n$; equivalently, an average-case algorithm is only required to succeed for a $1 - o_n(1)$ fraction of all 2^n possible source strings $x \in \{0,1\}^n$. These two problems are the focus of our work, so in the rest of this paper we consider worst-case and average-case trace reconstruction and we always assume that the deletion rate δ is an arbitrary (known) constant in $(0,1)$.

Despite much effort, there are mildly exponential gaps between the best known upper bounds and lower bounds for both worst-case and average-case trace reconstruction. Improving on earlier $2^{\tilde{O}(n^{1/2})}$-trace and $2^{\tilde{O}(n^{1/3})}$-trace algorithms of [25, 19, 36], in [11] Chase gave an algorithm for worst-case trace reconstruction that uses $2^{\tilde{O}(n^{1/5})}$ traces. The best known lower bound, also due to Chase [10], is $\tilde{\Omega}(n^{3/2})$ traces (improving on earlier $\tilde{\Omega}(n^{5/4})$ and $\Omega(n)$ lower bounds [22, 2]). For the average-case problem, improving on earlier $\exp(O((\log n)^{1/2}))$-trace and $\exp(O((\log n)^{1/3}))$-trace algorithms [37, 23, 24], Rubinstein [38] recently gave an $\exp(\tilde{O}((\log n)^{1/5}))$-trace algorithm. The best known average-case lower bound, due to Chase [10], is $\tilde{\Omega}((\log n)^{5/2})$ traces, improving on an earlier $\tilde{\Omega}((\log n)^{9/4})$ lower bound [22].

These substantial gaps naturally suggest the study of restricted classes of algorithms for trace reconstruction, with the hope that it may be possible to obtain sharper results. This is the starting point of our work: we propose to study the trace reconstruction problem from the vantage point of *statistical query* algorithms. As our main contribution we obtain fairly sharp upper and lower bounds on *local* statistical query algorithms for trace reconstruction, as described below.

Statistical Query trace reconstruction algorithms. The *Statistical Query* (SQ) model [27] was first introduced by Kearns as a means to obtain PAC learning algorithms that can tolerate random classification noise. In the decades since then, the SQ model has emerged as a major topic of study in its own right in computational learning theory and related fields such as differential privacy and optimization. An attractive feature of the SQ model is that it is powerful enough to capture state-of-the-art algorithms in a variety of different settings, yet it is also amenable to proving *unconditional* lower bounds.

SQ algorithms can only access data through noisy estimates of the expected values of user-generated query functions. In the context of trace reconstruction, an SQ oracle takes as input a bounded *query function* $q : \{0,1\}^n \to [-1,1]$ and a *tolerance parameter* $\tau \in (0,1)$ that are provided by the reconstruction algorithm. It returns a value $\widehat{P}_q$ which satisfies $|\widehat{P}_q - P_q| \leq \tau$, where P_q is the expected value of q on a random trace, i.e. $P_q := \mathbf{E}_{\boldsymbol{y} \sim \mathbf{Del}_\delta(x)}[q(\boldsymbol{y})]$.[1] Thus an SQ algorithm for trace reconstruction does not receive any actual traces of x; rather, it can only use aggregate statistical information about the overall distribution of traces.

[1] Since the length of each trace is at most n, we view each trace $\boldsymbol{y}$ as padded with a suffix of $n - |\boldsymbol{y}|$ zeros, so the argument to q is actually $\boldsymbol{y}0^{n-|\boldsymbol{y}|}$. This is equivalent to assuming that the n-bit source string x is padded with an infinite suffix of 0-bits.

To the best of our knowledge, the current paper is among the first works that explicitly considers the trace reconstruction problem from the perspective of statistical queries (see also [15], which we discuss in more detail below). However, in hindsight the earliest nontrivial algorithms for worst-case trace reconstruction [25, 19, 36] already made it evident that SQ algorithms – in fact, SQ algorithms which use extremely simple query functions – could be effective for trace reconstruction. The algorithms of [25, 19, 36] all work by using the traces from $\mathbf{Del}_\delta(x)$ only to obtain high-accuracy estimates of the n values $\mathbf{E}_{\boldsymbol{y}\sim\mathbf{Del}_\delta(x)}[\boldsymbol{y}_i]$ for $i \in [n]$ and then doing some subsequent computation on those estimated values; thus they correspond to SQ algorithms in which each query function is simply a Boolean dictator function, i.e. a 1-junta. On the other hand, the highly efficient average-case trace reconstruction algorithms of [37, 23, 24, 38], which use a sub-polynomial number of traces, involve various "alignment" routines which attempt to identify locations in individual received traces that correspond to specific locations in the source string. These algorithms seem to make essential use of individual traces and do not seem to be compatible with the SQ model. So given that some, but not all, known trace reconstruction algorithms correspond to SQ algorithms, it is of interest to study both the abilities and limitations of SQ algorithms for trace reconstruction.

In this work we consider a natural class of SQ algorithms, which we call *ℓ-local* SQ algorithms. An ℓ-local query function $q : \{0,1\}^n \to [-1,1]$ is an ℓ-junta over some ℓ consecutive bits of its input string, i.e. for all y, q satisfies $q(y) = q'(y_i, y_{i+1}, \ldots, y_{i+\ell-1})$ for some index i and some function $q' : \{0,1\}^\ell \to [-1,1]$. We say that an algorithm is an *ℓ-local SQ algorithm with tolerance* τ_0 if all of its calls to the SQ oracle are made with ℓ-local query functions and the tolerance parameter for each call is at least τ_0.

The results of [19, 36] already show that 1-local SQ algorithms with tolerance $\tau_0 = 2^{-\tilde{O}(n^{1/3})}$ can successfully perform worst-case trace reconstruction, and moreover [19, 36] additionally show that tolerance $\tau_0 = 2^{-\tilde{\Omega}(n^{1/3})}$ is required for any 1-local SQ worst-case trace reconstruction algorithm. Thus, in analyzing the abilities and limitations of ℓ-local algorithms for trace reconstruction for a particular value of ℓ, our goal is to determine the tolerance which is necessary and sufficient for such algorithms to succeed in worst-case or average-case trace reconstruction. A simple argument which we give in Section 2.1 shows that any ℓ-local SQ algorithm (which may be adaptive) using tolerance τ_0 can be converted to a nonadaptive SQ algorithm that makes at most $n2^\ell$ queries, all of which are ℓ-local "subword" queries (defined in Section 2.1) of tolerance $\tau_0 2^{-\ell}$. Moreover, a standard argument shows that any nonadaptive SQ algorithm which makes M statistical queries, each with tolerance at least τ_0, can be simulated in the obvious way by a standard trace reconstruction algorithm that uses $\mathrm{poly}(\log M, 1/\tau_0)$ independent traces from $\mathbf{Del}_\delta(x)$. Thus, we will be particularly interested in identifying the value ℓ of the locality parameter for which tolerance (roughly) $2^{-\ell}$ is both necessary and sufficient for trace reconstruction. As we explain next, our main results do precisely this, for both worst-case and average-case trace reconstruction.

1.1 Our results

We give upper and lower bounds on local SQ algorithms for both worst-case and average-case trace reconstruction. Our upper and lower bounds match each other up to fairly small factors for both the worst-case and average-case versions of the problem.

The worst-case problem. Our main lower bound is the following result, which gives a lower bound on the tolerance for $n^{1/5}$-local SQ algorithms performing worst-case trace reconstruction:

▶ **Theorem 1** (Worst-case lower bound, informal version of Theorem 6). *Fix any constant deletion rate $0 < \delta < 1$. For $\ell = \tilde{\Theta}(n^{1/5})$, any ℓ-local SQ algorithm for worst-case trace reconstruction must have tolerance $\tau_0 = \exp(-\tilde{\Omega}(n^{1/5}))$.*

Our algorithmic result for the worst-case problem shows that this lower bound is essentially optimal:

▶ **Theorem 2** (Worst-case upper bound, informal version of Theorem 15). *Fix any constant deletion rate $0 < \delta < 1$. There is a $\tilde{O}(n^{1/5}))$-local SQ algorithm for the worst-case trace reconstruction problem with tolerance $\tau_0 = \exp(-\tilde{O}(n^{1/5}))$.*

The average-case problem. As mentioned earlier, the state-of-the-art average-case trace reconstruction algorithms of [37, 23, 24, 38] do not seem to be compatible with the SQ model. Recall that those algorithms use $2^{O((\log n)^c)}$ traces, for $c \in \{1/5, 1/3, 1/2\}$, and thus any SQ analogue of those algorithms would have tolerance $\approx 2^{-O((\log n)^c)}$. We show that no $O(\log n)$-local (or even $n^{0.49}$-local) SQ algorithm for average-case trace reconstruction can succeed with such a coarse tolerance parameter:

▶ **Theorem 3** (Average-case lower bound, informal version of Theorem 23). *Fix any constant deletion rate $0 < \delta < 1$. Any ℓ-local SQ algorithm for average-case trace reconstruction must have tolerance $\tau_0 \leq \ell/\sqrt{n}$.*

Finally, we give an average-case $O(\log n)$-local SQ algorithm that has inverse polynomial tolerance:

▶ **Theorem 4** (Average-case upper bound, informal version of Theorem 25). *Fix any constant deletion rate $0 < \delta < 1$. There is an $O(\log n)$-local SQ algorithm for average-case trace reconstruction with tolerance $\tau_0 = 1/\mathrm{poly}(n)$.*

Our results can be summarized as follows: As discussed immediately before Section 1.1, we may say that an ℓ-local SQ algorithm with tolerance τ_0 has overall complexity $\mathrm{poly}(n2^\ell, 1/\tau_0)$. Theorems 1 and 2 together say that the optimal complexity of worst-case local SQ trace reconstruction is $2^{\tilde{\Theta}(n^{1/5})}$, and Theorems 3 and 4 together say that the optimal complexity of average-case local SQ trace reconstruction is $n^{\Theta(1)}$.

1.2 Discussion and techniques

The worst-case setting. Theorem 1 and Theorem 2 should be contrasted with recent results of Cheng et al. [15], which consider a restricted class of local SQ algorithms known as *ℓ-mer based* algorithms. As defined by Mazooji and Shomorony [31], the *ℓ-mer density map* is a certain vector of statistics about the frequency of length-ℓ subwords[2] of the source string $x \in \{0,1\}^n$. [31] gave an algorithm which, for constant deletion rate $0 < \delta < 1/2$, constructs an ε-accurate (in ℓ_∞ distance) estimate of the ℓ-mer density map using $\mathrm{poly}(n, 2^\ell, 1/\varepsilon)$ traces. Cheng et al. [15] defined a trace reconstruction algorithm to be *ℓ-mer based* if it only uses the ℓ-mer density map of x, and observed that the algorithm of [31] (see in particular Lemma 6 of [31] and its proof) only uses local statistical information about traces, and hence is a local SQ algorithm.

[2] Recall that a *subword* of x is a sequence of bits that occur consecutively in x, i.e. $x_i x_{i+1} \cdots x_{i+\ell-1}$, whereas a *substring* of x is a subsequence of bits that need not occur consecutively, i.e. $x_{i_1} x_{i_2} \cdots x_{i_\ell}$.

The main result of Cheng et al. is a proof that any $n^{1/5}$-mer based algorithm for worst-case trace reconstruction must have tolerance $\tau_0 = 2^{-\tilde{\Omega}(n^{1/5})}$. Our Theorem 1 generalizes this result because it gives a lower bound for the entire class of $n^{1/5}$-local SQ algorithms, which includes the class of $n^{1/5}$-mer based algorithms by the results described above. We remark that Theorem 1 also has a shorter and simpler proof than Theorem 2 of [15].

At a high-level, we obtain Theorem 1 by a reduction to proving a 1-local SQ lower bound on n-bit source strings that are "gappy." These are strings in which every two 1s are separated by $\gg n^{1/5}$ zeros (see Definition 8 for the precise definition). The intution here is that for a gappy string, any $n^{1/5}$-bit subword in its traces is very unlikely to contain two 1s, and so all the useful information is contained in subword queries of Hamming weight at most 1, which can then be further reduced to 1-bit queries. Then we can adapt the lower bound arguments in [19] to gappy strings to obtain our lower bound.

Turning to Theorem 2, Cheng et al. observed that the $2^{\tilde{O}(n^{1/5})}$-trace algorithm of [11] for worst-case trace reconstruction can be interpreted as a $\tilde{O}(n^{1/5})$-mer based algorithm with tolerance $\tau_0 = 2^{-\tilde{O}(n^{1/5})}$. By the earlier observation of Cheng et al. mentioned in the first paragraph and the [31] algorithm, which works provided that the deletion rate δ lies in $(0, 1/2)$, this means that Chase's algorithm can be expressed as a $\tilde{O}(n^{1/5})$-local SQ algorithm which has tolerance $\tau_0 = 2^{-\tilde{O}(n^{1/5})}$ when $\delta \in (0, 1/2)$. Our Theorem 15 is based on a similar observation about Chase's algorithm, but applied directly to the local SQ model without going through the notion of k-mer statistics. Our approach is based on techniques and arguments from [13]; using these techniques allows our argument to apply more generally to the entire range of deletion rates $\delta \in (0, 1)$.

Average-case. The average-case lower bound of Theorem 3 is proved using a fairly simple argument based on "hiding" a bit which might be either 0 or 1 in the middle of the source string. We turn to the average-case upper bound.

The average-case SQ algorithm described in Theorem 4 is obtained by adapting an algorithm for *smoothed* trace reconstruction to the SQ model. The [13] paper gives an algorithm for "smoothed" trace reconstruction, which is a generalization of the average-case trace reconstruction problem. While the algorithm of [13] only interacts with the input traces by using them to form empirical estimates of subword frequencies in traces, it is not trivially an SQ algorithm. This is because the [13] algorithm estimates these subword frequencies across a range of different deletion probabilities $\delta, \delta + \Delta, \delta + 2\Delta, \ldots$ up to $(\delta + 1)/2$. In the usual (non-SQ) trace reconstruction setting where traces are available, it is trivial to simulate access to $\mathbf{Del}_{\delta'}(x)$ given access to $\mathbf{Del}_\delta(x)$ for any $\delta' > \delta$, simply by drawing $\boldsymbol{y} \sim \mathbf{Del}_\delta(x)$ and deleting each bit of $\boldsymbol{y}$ independently with probability $\frac{1-\delta'}{1-\delta}$. But in the SQ setting, we only have access to statistical queries of traces drawn from $\mathbf{Del}_\delta(x)$ rather than individual traces. We circumvent this issue by showing that any algorithm that makes ℓ-local statistical queries with tolerance τ to $\mathbf{Del}_{\delta'}(x)$, for $\delta' > \delta$, can be simulated by an algorithm that makes only ℓ'-local statistical queries with tolerance τ' to $\mathbf{Del}_\delta(x)$, where (roughly speaking) $\ell' \approx \ell/(1 - \delta')$ and $\tau' = \Theta(\tau)$. With this ingredient in hand, the algorithm of [13] is easily adapted to give Theorem 4.

1.3 Future work

Several natural questions suggest themselves for future work. Perhaps the foremost among these is the following: Given Theorem 2, the current state-of-the-art unrestricted algorithm for the general worst-case trace reconstruction problem is an $\tilde{O}(n^{1/5})$-local SQ algorithm. Might it be the case that this is in fact an optimal algorithm for trace reconstruction? We

currently seem quite far from being able to resolve this (recall that the state of the art in lower bounds for unrestricted worst-case trace reconstruction algorithms is only $\tilde{\Omega}(n^{3/2})$ traces [11]).

A partial step towards answering the above bold question would be to establish lower bounds on general SQ algorithms for worst-case trace reconstruction, i.e. SQ algorithms that are not assumed to have bounded locality. It is difficult to imagine how queries that depend on far-separated portions of an input trace could be useful, but proving this seems quite challenging.

As a concrete first goal along these lines, a generalization of the notion of an ℓ-local SQ is the notion of a *size-s* SQ. A size-s SQ is an SQ which asks for the expected value of some s-junta function $q'(\boldsymbol{y}_{i_1}, \ldots, \boldsymbol{y}_{i_s})$ of a random trace $\boldsymbol{y}$, but unlike an ℓ-local SQ the input bits of the junta do not need to form a consecutive block of positions in $\boldsymbol{y}$. Similar to Lemma 5, a size-s SQ algorithm can be assumed without loss of generality to use only query functions of the form $\mathbf{1}[y_{i_1}, \ldots, y_{i_s}] = w$ as $(i_1, \ldots, i_s)$ ranges over $\binom{[n]}{s}$ and w ranges over $\{0,1\}^s$. Even the following goal appears to be quite challenging:

> Show that any SQ algorithm for the worst-case trace reconstruction problem that makes only size-2 queries must have tolerance $\tau = 1/n^{\omega(1)}$.

We believe that this is an interesting target problem for future work.

2 Preliminaries

Notation. Given integers $a \le b$ we write $[a:b]$ to denote $\{a, \ldots, b\}$. It will be convenient for us to index a binary string $x \in \{0,1\}^n$ using $[0:n-1]$ as $x = (x_0, \ldots, x_{n-1})$. We write ln to denote natural logarithm and log to denote logarithm to the base 2. We write $|x|$ to denote the length of a string x.

We denote the set of non-negative integers by $\mathbb{Z}_{\ge 0}$. We write $D_r(z)$ to denote the closed disk in the complex plane of radius r centered at $z \in \mathbb{C}$, and $\partial D_r(z)$ to denote the circle which is the boundary of that disk.

Subwords. Fix a string $x \in \{0,1\}^n$ and an integer $k \in [n]$. A k-subword of x is a (contiguous) subword of x of length k, given by $(x_a, x_{a+1}, \ldots, x_{a+k-1})$ for some $a \in [0:n-k]$. Given such a string x and integers $0 \le a < b \le n-1$, we write $x[a:b]$ to denote the subword $(x_a, x_{a+1}, \ldots, x_b)$. For a string $w \in \{0,1\}^k$, let $\#(w, x)$ denote the number of occurrences of w as a subword of x.

Distributions. We use bold font letters to denote probability distributions and random variables, which should be clear from the context. We write "$\boldsymbol{x} \sim \mathbf{X}$" to indicate that random variable $\boldsymbol{x}$ is distributed according to distribution $\mathbf{X}$.

Deletion channel and traces. Throughout this paper the parameter $\delta : 0 < \delta < 1$ denotes the *deletion probability*, and we write ρ to denote the retention probability $\rho = 1 - \delta$. Given a string $x \in \{0,1\}^n$, we write $\mathbf{Del}_\delta(x)$ to denote the distribution of the string that results from passing x through the δ-deletion channel (so the distribution $\mathbf{Del}_\delta(x)$ is supported on $\{0,1\}^{\le n}$), and we refer to a string drawn from $\mathbf{Del}_\delta(x)$ as a *trace* of x. Recall that a random trace $\boldsymbol{y} \sim \mathbf{Del}_\delta(x)$ is obtained by independently deleting each bit of x with probability δ and concatenating the surviving bits. [3]

[3] For simplicity in this work we assume that the deletion probability δ is known to the reconstruction algorithm.

For $x \in \{0,1\}^n$ (recall that we index the bits of x as $(x_0, \ldots, x_{n-1})$) we view a draw of a trace $\boldsymbol{y} \sim \mathbf{Del}(x)$ as corresponding to a ρ-biased random draw of a subset $\mathbf{R} \subseteq [0 : n-1]$, where the elements of $\mathbf{R}$ are the bits that are *retained* in x to obtain $\boldsymbol{y}$. So if the sorted elements of $\mathbf{R}$ are $\mathbf{R} = \{\boldsymbol{r}_0 < \boldsymbol{r}_1 < \cdots < \boldsymbol{r}_{\boldsymbol{m}-1}\}$ for some $\boldsymbol{m} \leq n$, then the bits of the trace $\boldsymbol{y} = (\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\boldsymbol{m}-1})$ are $\boldsymbol{y}_0 = x_{\boldsymbol{r}_0}, \boldsymbol{y}_1 = x_{\boldsymbol{r}_1}$, and so on.

2.1 Local Statistical Query algorithms

As described earlier, an ℓ-local query function $q : \{0,1\}^n \to [-1,1]$ is a function

$$q(y) = q'(y_i, y_{i+1}, \ldots, y_{i+\ell-1})$$

for some index i and some function $q' : \{0,1\}^\ell \to [-1,1]$, i.e. a real-valued bounded ℓ-junta over consecutive input variables. An algorithm is an *ℓ-local SQ algorithm with tolerance τ_0* if all of its calls to the SQ oracle are made with ℓ-local query functions and the tolerance parameter for each call is at least τ_0.

Let us say that an ℓ-local query function is a *subword query* if it is of the form

$$q'(y) = \mathbf{1}\left[(y_i, \ldots, y_{i+\ell-1}) = w\right] \tag{1}$$

for some string $w \in \{0,1\}^\ell$. The following simple lemma shows that without loss of generality, every ℓ-local SQ algorithm makes at most $n2^\ell$ (non-adaptive) queries, corresponding to all possible length-ℓ subword queries:

▶ **Lemma 5.** *Let A be an ℓ-local SQ algorithm with tolerance τ_0 (note that A may make any number of calls to the SQ oracle and may be adaptive, i.e. the choice of later queries may depend on the responses received on earlier queries). Then there is an algorithm A' with the same behavior as A which makes $n2^\ell$ queries (all possible length-ℓ subword queries), each with tolerance $\tau_0/2^\ell$.*

Proof. The algorithm A' makes all $n2^\ell$ subword queries of the form given in Equation (1), where i ranges over $[n]$ and w ranges over $\{0,1\}^\ell$. It makes each such subword query with tolerance parameter $\tau_0/2^\ell$. Let $p_{i,w} = \mathbf{Pr}_{\boldsymbol{y}\sim\mathbf{Del}(x)}[(\boldsymbol{y}_i, \ldots, \boldsymbol{y}_{i+\ell-1}) = w]$ and let $\widehat{p}_{i,w}$ be the value received from the SQ oracle in response to the query (1), so $|\widehat{p}_{i,w} - p_{i,w}| \leq \tau_0/2^\ell$.

Let (q, τ_0) be any (query function, tolerance) pair that A may make in the course of its execution. We show that a $\pm\tau_0$-accurate estimate $\widehat{P}_q$ of P_q can be computed from the responses to the $n2^\ell$ queries of A'. This is easily seen to imply the lemma.

Since A is ℓ-local, the expected value P_q is

$$P_q = \mathop{\mathbf{E}}_{\boldsymbol{y}\sim\mathbf{Del}_\delta(x)}[q'(\boldsymbol{y}_i, \ldots, \boldsymbol{y}_{i+\ell-1})]$$

for some $q' : \{0,1\}^\ell \to [-1,1]$ and some $i \in [n]$. Since

$$P_q = \sum_{w\in\{0,1\}^\ell} p_{i,w} \cdot q'(w),$$

by setting $\widehat{P}_q$ to be

$$P_q = \sum_{w\in\{0,1\}^\ell} \widehat{p}_{i,w} \cdot q'(w),$$

recalling that $|q'(w)| \le 1$ for all w, the triangle inequality gives

$$|\widehat{P}_q - P_q| = \left| \sum_{w \in \{0,1\}^\ell} (\widehat{p}_{i,w} - p_{i,w}) \cdot q'(w) \right| \le \max_w |q'(w)| \cdot \sum_w |\widehat{p}_{i,w} - p_{i,w}| \le \sum_w \tau_0 / 2^\ell \le \tau_0$$

as desired. ◀

3 Worst-case lower bounds

In this section we prove the following lower bound on local SQ algorithms for the worst-case trace reconstruction problem:

▶ **Theorem 6** (Worst-case lower bound). *Fix any constant deletion rate $0 < \delta < 1$. For a suitable absolute constant c_0, any $c_0 n^{1/5}/(\log n)^{2/5}$-local SQ algorithm for worst-case trace reconstruction must have tolerance $\tau_0 < \exp(-\Omega(n^{1/5}/(\log n)^{2/5}))$.*

Setup. Fix any $0 < \delta < 1$. For notational clarity let us write $\ell := c_0 n^{1/5}/(\log n)^{2/5}$. Given an n-bit source string x, an index $i \in [0 : n-1]$, and an ℓ-bit string w, we define the value

$$p_{x,i,w} := \Pr_{\boldsymbol{y} \sim \mathbf{Del}_\delta(x)} [(\boldsymbol{y}_i, \dots, \boldsymbol{y}_{i+\ell-1}) = w], \tag{2}$$

so $p_{x,i,w}$ is the probability that a random trace of x has w as the subword starting in position i. We refer to the vector $(p_{x,i,w})_{i \in [0:n-1], w \in \{0,1\}^\ell}$ as the *ℓ-subword signature* of x.

We will prove the following:

▶ **Lemma 7.** *For a suitable absolute constant c_0, there are distinct n-bit strings $a \ne a' \in \{0,1\}^n$ whose ℓ-subword signatures are very close to each other in ℓ_∞-distance: more precisely,*

$$\text{For all } i \in [0 : n-1], w \in \{0,1\}^\ell, \text{ we have } |p_{a,i,w} - p_{a',i,w}| \le \exp(-2c_0 n^{1/5}/(\log n)^{2/5}). \tag{3}$$

To see why Lemma 7 implies Theorem 6, let A be any ℓ-local SQ algorithm with tolerance $\exp(-c_0 n^{1/5}/(\log n)^{2/5})$. By Lemma 5, there is an algorithm A' with the same behavior as A which makes only subword queries for subwords of length ℓ, where each query of A' has tolerance $\exp(-c_0 n^{1/5}/(\log n)^{2/5})/2^\ell > \exp(-2c_0 n^{1/5}/(\log n)^{2/5})$. By Equation (3), a query for the value of $p_{x,i,w}$ can be answered with the value $q_{i,w} = \frac{p_{a,i,w} + p_{a',i,w}}{2}$ whether the source string x is a or a'. But this means that it is impossible for A to be an algorithm which successfully solves the worst-case trace reconstruction problem.

In the rest of this section, we focus on establishing Lemma 7.

We require the following simple definition:

▶ **Definition 8.** *Given $t > 1$, we say that a string $x \in \{0,1\}^n$ is* t-gappy *if it is of the form*

$$x = b_0 0^{t-1} b_1 0^{t-1} \cdots b_{n/t-1} 0^{t-1}$$

for some string $b_0, b_1, \dots, b_{n/t-1} \in \{0,1\}^{n/t}$.

Recall $\rho = 1 - \delta$. Fix

$$t := \frac{100 \log(n) \ell}{\rho} = \Theta(n^{1/5} (\log n)^{3/5}). \tag{4}$$

The two strings a, a' whose existence is asserted by Lemma 7 will both be t-gappy. (We note that the argument of Cheng et al. [15] also used gappy strings.)

One reason that gappy strings are useful for us because they make it very easy to handle almost all of the ℓ-bit strings $w \in \{0,1\}^\ell$ that we need to consider in order to establish Lemma 7. To see this, observe that any string w containing at least two ones is very unlikely to be a length-ℓ subword of a random trace $\boldsymbol{y}$: since the source string is t-gappy, we expect consecutive ones in a random trace $\boldsymbol{y}$ to be at least $\rho t \gg \ell$ positions apart from each other. More precisely, we have the following lemma:

▶ **Lemma 9.** *Let $x \in \{0,1\}^n$ be any t-gappy string, and let $w \in \{0,1\}^\ell$ be any string with at least two ones. Then for any $i \in [0 : n-1]$ we have $p_{x,i,w} \le 1/n^{49\ell}$.*

Proof. Fix $0 \le \alpha < \beta \le \ell - 1$ to be any two positions in w such that $w_\alpha = w_\beta = 1$, and let $\boldsymbol{y} \sim \mathbf{Del}_\delta(x)$. Observe that we have $p_{x,i,w} \le \mathbf{Pr}[\boldsymbol{y}_{i+\alpha} = \boldsymbol{y}_{i+\beta} = 1]$.

Let $\mathbf{R} = \{\boldsymbol{r}_0 < \boldsymbol{r}_2 < \cdots < \boldsymbol{r}_{\boldsymbol{m}-1}\} \subseteq [0 : n-1]$ be the ρ-biased random subset of $[0 : n-1]$ consisting of the indices that are retained in x to obtain $\boldsymbol{y}$. We may view the draw of $\mathbf{R}$ as being carried out sequentially in independent stages $0, 1, \ldots$, where in each stage s the element s is included in $\mathbf{R}$ with probability ρ. Fix any outcome of stages $0, 1, \ldots$ up until $r_{i+\alpha}$ has been included in $\mathbf{R}$. Even supposing that $x_{r_{i+\alpha}} = 1$ (so that $y_{i+\alpha} = 1$), the probability that $x_{\boldsymbol{r}_{i+\beta}} = 1$ (which equals $\mathbf{Pr}[\boldsymbol{y}_{i+\beta} = 1]$) is at most (writing k for $\beta - \alpha$)

$$\sum_{j \ge 1} \mathbf{Pr}[\text{exactly } k \text{ of the } jt \text{ indices in } r_{i+\alpha}+1, \ldots, r_{i+\alpha}+jt \text{ are retained}] \tag{5}$$

$$= \sum_{j \ge 1} \binom{jt}{k} \rho^k \delta^{jt-k}$$

$$\le \sum_{j \ge 1} (\rho j t)^k \cdot \delta^{jt/2} = (\rho t)^k \sum_{j \ge 1} j^k \delta^{jt/2} \qquad \text{(since } k \le jt/2\text{)}$$

$$\le (\rho t)^\ell \sum_{j \ge 1} j^\ell (1-\rho)^{jt/2} \qquad \text{(using } k \le \ell\text{)}$$

$$\le (100 \log(n) \ell)^\ell \sum_{j \ge 1} j^\ell e^{-50 \log(n) \ell j}. \quad \text{(by the choice of } t\text{, and using } (1-\rho)^{1/\rho} \le e^{-1}\text{)}$$

When $j = 1$ the first term of the sum $\sum_{j \ge 1} j^\ell e^{-50 \log(n) \ell j}$ is $e^{-50 \log(n) \ell}$. The ratio of successive terms of the sum is

$$\frac{(j+1)^\ell e^{-50 \log(n) \ell (j+1)}}{j^\ell e^{-50 \log(n) \ell j}} \le 2^\ell e^{-50 \log(n) \ell} = (2/n^{50})^\ell \ll 1/2.$$

So the sum $\sum_{j \ge 1} j^\ell e^{-50 \log(n) \ell j}$ is at most $2e^{-50 \log(n) \ell} = 2/n^{50\ell}$, and since $100 \log(n) \ell < n$ for n sufficiently large, we get that (5) $\le 1/n^{49\ell}$. It follows that $p_{x,i,w} \le \mathbf{Pr}[\boldsymbol{y}_{i+\alpha} = \boldsymbol{y}_{i+\beta} = 1] \le 1/n^{49\ell}$ as claimed. ◀

Given Lemma 9 it remains to argue about the $\ell + 1$ strings $w \in \{0,1\}^\ell$ of Hamming weight 0 or 1. We handle the weight-1 strings by reducing their analysis to the analysis of *one-bit* strings as follows: fix any $\alpha \in [0 : \ell - 1]$ and let $w = e_\alpha \in \{0,1\}^\ell$ be the string with a single 1 coordinate in position α. The following lemma, which we prove using Lemma 9, shows that for any gappy source string x the value of p_{x,i,e_α} is very close to the expected value of a *single* location in a random trace. (A sharper bound could be obtained with a bit more work, but the bound given by Lemma 10 is sufficient for our purposes.)

▶ **Lemma 10.** *Let $x \in \{0,1\}^n$ be any t-gappy string, and let $w = e_\alpha \in \{0,1\}^\ell$ be the string containing a single 1 in coordinate α. Then for any $i \in [0 : n-1]$ we have*

$$\left| \mathop{\mathbf{Pr}}_{\boldsymbol{y} \sim \mathbf{Del}_\delta(x)} [\boldsymbol{y}_{i+\alpha} = 1] - p_{x,i,e_\alpha} \right| \le 2^{\ell-1}/n^{49\ell}.$$

Proof. We have

$$\Pr_{\boldsymbol{y}\sim\mathbf{Del}_\delta(x)}[\boldsymbol{y}_{i+\alpha}=1]=\sum_{w\in\{0,1\}^\ell:w_\alpha=1}p_{x,i,w},\quad\text{so}$$

$$0\le\Pr_{\boldsymbol{y}\sim\mathbf{Del}_\delta(x)}[\boldsymbol{y}_{i+\alpha}=1]-p_{x,i,e_\alpha}=\sum_{w\in\{0,1\}^\ell:w_\alpha=1,|w|\ge2}p_{x,i,w}\le(2^{\ell-1}-1)/n^{49\ell}$$

where the inequality is Lemma 9. ◀

The one remaining ℓ-bit string to consider is $w=0^\ell$. However, if all $2^\ell-1$ other strings have been handled successfully then this string is automatically handled as well:

▶ **Lemma 11.** *Fix $a,a'\in\{0,1\}^n$ and $i\in[0:n-1]$. Suppose that for all $w\in\{0,1\}^\ell\setminus\{0^\ell\}$ we have $|p_{a,i,w}-p_{a',i,w}|\le\kappa$. Then $|p_{a,i,0^\ell}-p_{a',i,0^\ell}|\le(2^\ell-1)\kappa$.*

Proof. This is an immediate consequence of $\sum_{w\in\{0,1\}^\ell}p_{x,i,w}=1$, which holds for every x and i. ◀

Thus, it suffices to construct two t-gappy strings a,a' whose one-bit statistics are very close:

▶ **Lemma 12.** *For $x\in\{0,1\}^n$ and $i\in[0:n-1]$ define*

$$p_{x,i}:=\Pr_{\boldsymbol{y}\sim\mathbf{Del}_\delta(x)}[\boldsymbol{y}_i=1]. \tag{6}$$

Suppose that $a\ne a'\in\{0,1\}^n$ are two t-gappy strings such that for each $i\in[0:n-1]$ we have $|p_{a,i}-p_{a',i}|\le\exp(-\Omega(n^{1/5}/(\log n)^{2/5}))$. Then for all $i\in[0:n-1]$, $w\in\{0,1\}^\ell$ we have

$$|p_{a,i,w}-p_{a',i,w}|\le2^\ell\cdot\exp(-\Omega(n^{1/5}/(\log n)^{2/5}))+4^\ell/n^{49\ell}\le\exp(-2c_0n^{1/5}/(\log n)^{2/5}).$$

Proof. Lemma 9 gives $|p_{a,i,w}-p_{a',i,w}|\le1/n^{49\ell}$ for $|w|\ge2$. Lemma 10 and the assumption on $|p_{a,i}-p_{a',i}|$ gives $|p_{a,i,w}-p_{a',i,w}|\le\exp(-\Omega(n^{1/5}/(\log n)^{2/5}))+2^\ell/n^{49\ell}$ for $|w|=1$. Given these bounds, Lemma 11 gives $|p_{a,i,0^\ell}-p_{a',i,0^\ell}|\le2^\ell\cdot\exp(-\Omega(n^{1/5}/(\log n)^{2/5}))+4^\ell/n^{49\ell}$. ◀

3.1 Establishing closeness of one-bit statistics

Let us write $p_x=(p_{x,0},\dots,p_{x,n-1})$ to denote the n-dimensional vector in $[0,1]^n$ whose coordinates are given by Equation (6). From the results in the previous subsection it suffice to prove the following:

▶ **Lemma 13.** *There are two distinct t-gappy strings $a,a'\in\{0,1\}^n$ such that for all $i\in[0:n-1]$ we have $\|p_a-p_{a'}\|_\infty\le\exp(-\Omega(n^{1/5}/(\log n)^{2/5}))$.*

This is very similar to the main lower bound statement that was established in the two works [19, 36] (independently of each other); those papers considered "one-bit statistics" which correspond precisely to our $p_{x,i}$ quantities, and showed that there are two distinct strings $x,x'\in\{0,1\}^n$ (not restricted to be gappy) such that $|p_{x,i}-p_{x',i}|\le\exp(-\Omega(n^{1/3}))$ for all $i\in[0:n-1]$. In what follows we adapt their techniques to deal with t-gappy source strings.

Following [19], given a pair of source strings $a,a'\in\{0,1\}^n$ we define the corresponding *deletion-channel polynomial* (over $\mathbb{C}$) to be

$$P_{a,a'}(z):=\sum_{i=0}^{n-1}(p_{a,i}-p_{a',i})\cdot z^i. \tag{7}$$

We have

$$\|p_a - p_{a'}\|_\infty \le \|p_a - p_{a'}\|_1 \le \sqrt{n} \max_{z \in \partial D_1(0)} |P_{a,a'}(z)|, \tag{8}$$

where the second inequality is by Proposition 3.5 of [19] (the proof is a simple and standard computation about complex polynomials). Thus our goal is to establish the existence of two distinct t-gappy strings $a \neq a' \in \{0,1\}^n$ for which $\max_{z \in \partial D_1(0)} |P_{a,a'}(z)|$ is small. To do this, we begin by observing that since bit j of a source string ends up in location i of a trace with probability $\binom{j}{i}\rho^{i+1}\delta^{j-i}$, we have

$$p_{a,i} = \mathop{\mathbf{Pr}}_{\boldsymbol{y} \sim \mathbf{Del}_\delta(a)}[\boldsymbol{y}_i = 1] = \sum_{j=0}^{n-1} \binom{j}{i} \rho^{i+1}\delta^{j-i} a_j, \text{ and hence } p_{a,i} - p_{a',i} = \sum_{j=0}^{n-1} (a_j - a'_j)\binom{j}{i}\rho^{i+1}\delta^{j-i}.$$

Hence (following [19, 36]) we get

$$\begin{aligned} P_{a,a'}(z) &= \sum_{i=0}^{n-1} \left(\sum_{j=0}^{n-1} (a_j - a'_j)\binom{j}{i}\rho^{i+1}\delta^{j-i} \right) z^i = \rho \sum_{j=0}^{n-1} (a_j - a'_j)\delta^j \sum_{i=0}^{n-1} \binom{j}{i} \left(\frac{\rho z}{\delta}\right)^i \\ &= \rho \sum_{j=0}^{n-1} (a_j - a'_j) w^j \quad (\text{taking } w = 1 - \rho + \rho z) \end{aligned} \tag{9}$$

where the last line used the binomial theorem and $\delta = 1 - \rho$. Now, let us write the t-gappy strings a, a' as

$$a := b_0 0^{t-1} b_1 0^{t-1} \cdots b_{n/t} 0^{t-1}, \qquad a' := b'_0 0^{t-1} b'_1 0^{t-1} \cdots b'_{n/t} 0^{t-1} \tag{10}$$

for some $b, b' \in \{0,1\}^{n/t}$. From Equation (9) we get that

$$P_{a,a'}(z) = \rho \cdot \sum_{j=0}^{n/t-1} (b_j - b'_j) w^{jt} \tag{11}$$

(the structure afforded by Equation (11) is another reason why t-gappy strings are useful for us). Since $0 < \rho = 1 - \delta < 1$ is a constant, recalling Equation (8) our goal is to establish the existence of a string $0^{n/t} \neq v = (v_0, \ldots, v_{n/t-1}) \in \{-1, 0, 1\}^{n/t}$ such that

$$\max_{\theta \in (-\pi, \pi]} \left| \sum_{j=0}^{n/t-1} v_j \left((1 - \rho + \rho e^{i\theta})^t \right)^j \right| \tag{12}$$

is small.

As described in Theorem 6.2 of [19], a result of Borwein and Erdélyi [3] (specifically, the first proof of Theorem 3.3 in the "special case" on p. 11 of [3]) establishes the following:

▶ **Theorem 14** ([3]). *There are universal constants $c_1, c_2, c_3 > 0$ such that the following holds: For all $0 < a \le c_1$ there exists an integer $2 \le k \le c_2/a^2$ and a nonzero vector $u \in \{-1, 0, 1\}^{k+1}$ such that $\max_{w \in D_{6a}(1)} |\sum_{j=0}^{k} u_j w^j| \le \exp(-c_3/a)$.*

Let $m = n^{1/5}/(\log n)^{2/5} = 1/a$, so $a = 1/m = (\log n)^{2/5}/n^{-1/5}$. Recalling Equation (4), we have that $c_2/a^2 = c_2 m^2 \ll n/(2t)$, so we get that there exists a vector $0^{n/(2t)} \neq u \in \{-1, 0, 1\}^{n/(2t)}$ such that

$$\max_{w \in D_{6/m}(1)} \left| \sum_{j=0}^{n/(2t)-1} u_j w^j \right| \le \exp(-c_3 m). \tag{13}$$

Routine geometry shows that if $|\theta| \le \frac{1}{mt}$ then $|1-(1-\rho+\rho e^{i\theta})^t| \le 6/m$, so we get that

$$\max_{|\theta|\le 1/(mt)} \left| \sum_{j=0}^{n/(2t)-1} u_j \left((1-\rho+\rho e^{i\theta})^t\right)^j \right| \le \exp(-c_3 m). \tag{14}$$

Now we can describe our final desired string $v \in \{-1,0,1\}^{n/t}$: it is obtained by padding u with a prefix of $n/(2t)$ many zeros. We thus have

$$\begin{aligned}(12) &= \max_{\theta\in(-\pi,\pi]} \sum_{j=0}^{n/t-1} v_j \left((1-\rho+\rho e^{i\theta})^t\right)^j \\ &= \max_{\theta\in(-\pi,\pi]} \left(\overbrace{(1-\rho+\rho e^{i\theta})^{n/2}}^{A} \cdot \overbrace{\sum_{j=0}^{n/(2t)-1} u_j \left((1-\rho+\rho e^{i\theta})^t\right)^j}^{B} \right). \end{aligned} \tag{15}$$

Since $|1-\rho+\rho e^{i\theta}| \le 1$ for all $\theta \in (-\pi,\pi]$, we have that $|A|$ is always at most 1 and $|B|$ is always at most $n/(2t)$. We bound Equation (15) by considering two possible ranges for $|\theta|$. If $|\theta| \le 1/(mt)$, then since $|A| \le 1$, from Equation (14) we have that (15) $\le 1 \cdot |B| \le \exp(-c_3 m)$. On the other hand, if $|\theta| > 1/(mt)$ then since $|B| \le n/(2t)$ and ρ is a constant between 0 and 1, we get that $|1-\rho+\rho e^{i\theta}| \le 1 - \frac{c_\rho}{(mt)^2}$, and hence

$$(15) \le \frac{n}{2t} \cdot |A| \le \frac{n}{2t} \cdot \left(1 - \frac{c_\rho}{(mt)^2}\right)^{n/2} \le \exp(-c'_\rho n/(mt)^2)$$

for two constants $c_\rho, c'_\rho > 0$ that depend only on ρ. Since $m = n^{1/5}/(\log n)^{2/5}$ and $n/(mt)^2 = \Theta(n^{1/5}/(\log n)^{2/5})$, for all $\theta \in (-\pi,\pi]$ we have that (12) $\le \exp(-\Omega(n^{1/5}/(\log n)^{2/5}))$, so the proof of Lemma 13 and hence of Theorem 6 is complete.

4 Worst-case upper bounds

In this section we will give a local SQ algorithm for worst-case trace reconstruction, proving Theorem 2.

▶ **Theorem 15** (Worst-case upper bound). *Fix any constant deletion rate $0 < \delta < 1$. There is a worst-case SQ trace reconstruction algorithm that makes only $(O(n^{1/5}\log^5 n))$-local queries with tolerance $\tau = 2^{-O(n^{1/5}\log^5 n)}$.*

Overview. As discussed in the introduction, [15] showed that the state-of-the-art worst-case trace reconstruction algorithm of Chase [11] can be interpreted as a $\tilde{O}(n^{1/5})$-mer based algorithm, and further observed that the work [31] implicitly showed that for deletion rate $\delta < 1/2$, any k-mer based algorithm only relies on local statistics of random traces. The same observation can also be inferred from the work [13]; more generally, that work implicitly showed that for *any* deletion rate $0 < \delta < 1$ (not just $\delta < 1/2$), Chase's algorithm can be interpreted as a local SQ algorithm. We obtain Theorem 15 by making this interpretation explicit, without going through the notion of k-mer statistics.

In the case of $\delta < 1/2$, the observation in [13, 31] is the following. Chase's algorithm is based on estimating (from below) a certain univariate polynomial $Q_x(z_0)$ at some point z_0 inside the shifted complex disc $D := \{\frac{z-\delta}{1-\delta} : |z| \le 1\}$. Moreover, the degree-$\ell$ coefficient of Q_x can be estimated using ℓ-local statistics. When δ is bounded away from $1/2$, these

works observed that the magnitude of the degree-ℓ term of Q_x decays exponentially in ℓ, and so the contribution from the high-degree terms is negligible and can be truncated from the evaluation.

In the case of $\delta \geq 1/2$, a point in D can have magnitude 1 or more, and so the high-degree terms in Q_x need not decay in magnitude. Instead of evaluating the polynomial on some point in D, [13] applies a result by Borwein, Erdélyi, and Kós [4] (see Lemma 18 below) which shows that there exists a value t_0 in the real interval $[\delta, \frac{1}{4} + \frac{3}{4}\delta]$ such that $Q_x(t_0)$ is almost as large as $Q_x(z_0)$, and as a result, we can estimate the truncation of $Q_x(t_0)$ instead.

We now proceed to a detailed proof of Theorem 15. Let $\ell := 2n^{1/5}$. Our $(O(n^{1/5}\log^5 n))$-local SQ algorithm in Theorem 15 is based on the following two lemmas. (Throughout this section, it will be more convenient for us to phrase various quantities in terms of the retention rate $\rho = 1 - \delta$.) For a source string $x \in \{0,1\}^n$ and an ℓ-bit pattern $w \in \{0,1\}^\ell$, let $P_{x,w}(z,t)$ be the following bivariate polynomial:

$$P_{x,w}(z,t) := \sum_{0 \leq i_1 < \cdots < i_\ell \leq n-1} \prod_{k=1}^{\ell} \mathbf{1}\left[x_{i_k} = w_k\right] z^{i_1} \cdot t^{i_\ell - i_1 - (\ell-1)}.$$

▶ **Lemma 16.** *For every deletion rate $\delta \in (0,1)$, there is a constant C_ρ such that the following holds. For every distinct pair of source strings $x, x' \in \{0,1\}^n$, there is a pattern $w \in \{0,1\}^\ell$, a point $z_0 \in \{e^{i\theta} : |\theta| \leq n^{-2/5}\} \cup [1-\rho, 1-\frac{3}{4}\rho]$, and a real value $t_0 \in [1-\rho, 1-\frac{3}{4}\rho]$, such that*

$$|P_{x,w}(z_0,t_0) - P_{x',w}(z_0,t_0)| \geq \exp\left(-C_\rho n^{1/5}\log^5 n\right).$$

▶ **Lemma 17.** *For every deletion rate $\delta \in (0,1)$, there exists an SQ algorithm that makes $C_\rho n^{1/5}\log^5 n$-local queries with tolerance $\exp(-C_\rho n^{1/5}\log^5 n)$ such that for every $w \in \{0,1\}^\ell$, $z \in \{e^{i\theta} : |\theta| \leq n^{-2/5}\} \cup [1-\rho, 1-\frac{3}{4}\rho]$, and $t \in [1-\rho, 1-\frac{3}{4}\rho]$ it outputs an estimate $\widehat{P_{x,w}}(z,t)$ of $P_{x,w}(z,t)$ that is accurate to within $\pm 0.1 \cdot \exp(-C_\rho n^{1/5}\log^5 n)$.*

Our ℓ-local SQ algorithm (Proof of Theorem 15 assuming Lemmas 16 and 17)

Given an unknown source string $x \in \{0,1\}^n$, our reconstruction algorithm enumerates every pair of distinct strings $x_1 \neq x_2 \in \{0,1\}^n$. For each such pair, it considers the triple (w, z_0, t_0) for that pair whose existence is given by Lemma 16. (Hence there are at most 2^{2n} many such triples (w, z_0, t_0) considered in total.) Then it uses the SQ algorithm in Lemma 17 to obtain an accurate estimate $\widehat{P_{x,w}}(z_0,t_0)$ of $P_{x,w}(z_0,t_0)$ for each w within an additive factor of $\pm 0.1 \cdot \exp(-C_\rho n^{1/5}\log^5 n)$, and outputs the x' such that $\widehat{P_{x,w}}(z_0,t_0)$ and $P_{x',w}(z_0,t_0)$ are $\pm 0.5 \cdot \exp(-C_\rho n^{1/5}\log^5 n)$-close to each other for every w, z_0, t_0. The correctness follows immediately from Lemma 16, because if $x' \neq x$, then by that lemma there is some (w, z_0, t_0) such that by the triangle inequality we have

$$\begin{aligned}|\widehat{P_{x,w}}(z_0,t_0) - P_{x',w}(z_0,t_0)| &\geq |P_{x,w}(z_0,t_0) - P_{x',w}(z_0,t_0)| - |\widehat{P_{x,w}}(z_0,t_0) - P_{x,w}(z_0,t_0)| \\ &\geq 0.9 \cdot \exp(-C_\rho n^{1/5}\log^5 n).\end{aligned}$$

4.1 Proof of Lemma 16

In this subsection we prove Lemma 16. We first recall the following result from [4].

▶ **Lemma 18** (Theorem 5.1 in [4]). *There are constants $c_1, c_2 > 0$ such that for every analytic function f on the open unit disc $\{z : |z| < 1\}$ with $|f(z)| < \frac{1}{1-|z|}$ and every $a \in (0,1]$, we have*

$$|f(0)|^{\frac{c_1}{a}} \leq \exp(c_2/a) \sup_{t \in [1-a, 1-\frac{3}{4}a]} |f(t)|.$$

Note that polynomials with coefficients bounded by 1 are clearly analytic and satisfy the condition that $|f(z)| < \frac{1}{1-|z|}$ on the open unit disc $\{z : |z| < 1\}$.

We note that in the actual statement in [4, Theorem 5.1], the interval containing t is $[1-a, 1]$. However, a close inspection of the proof reveals that the interval can be restricted to be $[1-a, 1-\frac{3}{4}a]$. Specifically, their Theorem 5.1 is based on their Corollary 5.3, which in turn is based on their Corollary 5.2, where the interval is taken to be $[1-a, 1-a+\frac{1}{4}a]$. A self-contained proof using essentially the same argument can also be found in [13, Theorem 9].

We further note that the difference between $[1-a, 1]$ and $[1-\frac{3}{4}a]$ is crucial in showing that the contribution of the high-degree terms of the relevant polynomial (Equation (16)) is negligible. Had t been 1, then $\frac{t-(1-\rho)}{\rho} = 1$ and there would have been no exponential decay in the high-degree terms.

Lemma 16 follows from two cases below.

Case 1: $x_i \neq x'_i$ for some $0 \leq i \leq \ell - 1$

In this case, we consider the ℓ-bit pattern $w := x[0 : \ell-1]$. Note that $P_{x,w}(0,0) - P_{x',w}(0,0) = \mathbf{1}[x[0:\ell-1] = w] - \mathbf{1}[x'[0:\ell-1] = w] = 1$. We now apply Lemma 18 twice. The first application is to the polynomial $Q_1(z_1) := P_{x,w}(z_1, 0) - P_{x',w}(z_1, 0)$, which implies that there exists some $z_0 \in [1-\rho, 1-\frac{3}{4}\rho]$ such that

$$|Q_1(z_0)| \geq e^{-c_2/\rho}|Q_1(0)|^{c_1/\rho} = e^{-c_2/\rho}|P_{x,w}(0,0) - P_{x',w}(0,0)|^{c_1/\rho} = e^{-c_2/\rho}.$$

We now apply Lemma 18 again to the polynomial

$$Q_2(z_2) := \frac{P_{x,w}(z_0, z_2) - P_{x',w}(z_0, z_2)}{\binom{n}{\ell}}.$$

Note that all coefficients in Q_2 have magnitude at most 1. This implies the existence of some $t_0 \in [1-\rho, 1-\frac{3}{4}\rho]$ such that

$$\begin{aligned}|P_{x,w}(z_0,t_0) - P_{x',w}(z_0,t_0)| = \binom{n}{\ell}|Q_2(t_0)| &\geq \binom{n}{\ell}e^{-c_2/\rho}|Q_2(0)|^{c_1/\rho} = \binom{n}{\ell}e^{-c_2/\rho}\left(\frac{|Q_1(z_0)|}{\binom{n}{\ell}}\right)^{c_1/\rho} \\ &\geq \frac{e^{-\frac{c_2}{\rho} - \frac{c_1 c_2}{\rho^2}}}{\binom{n}{\ell}^{\frac{c_1}{\rho}-1}} \geq e^{-\Omega_\rho(\ell \log n)} = e^{-\Omega_\rho(n^{1/5}\log n)},\end{aligned}$$

where the last inequality used $\binom{n}{\ell} \geq (n/\ell)^\ell$, and the last equality follows from our choice of $\ell = 2n^{1/5}$. To conclude, there exists some $(z_0, t_0) \in [1-\rho, 1-\frac{3}{4}\rho]^2$ such that $|P_{x,w}(z_0,t_0) - P_{x',w}(z_0,t_0)| \geq \Omega_\rho(\binom{n}{\ell}^{-c_1/\rho})$.

Case 2: $x_i = x'_i$ for all $0 \leq i \leq \ell - 1$

For this case, [11, Corollary 6.1] (with the interval $[1-2\rho, 1]$ replaced with $[1-\rho, 1-\frac{3}{4}\rho]$) can be restated, using Lemma 18 in a similar fashion as Case 1, as follows:

▶ **Lemma 19** (Corollary 6.1 in [11], slightly rephrased and refined). *For every $\rho > 0$, there exists a constant C_ρ such that the following holds. Let $\ell = 2n^{1/5}$. For every distinct $x, x' \in \{0,1\}^n$ where $x_i = x'_i$ for every $0 \leq i < \ell - 1$, there exists a pattern $w \in \{0,1\}^\ell$, a $z_0 = e^{i\theta}$ for some $\theta \in [-n^{-2/5}, n^{-2/5}]$ and a $t_0 \in [1-\rho, 1-\frac{3}{4}\rho]$ such that*

$$\left|\sum_{0 \leq i_1 < \cdots < i_\ell \leq n-1} \left(\prod_{k=1}^{\ell} \mathbf{1}[x_{i_k} = w_k] - \prod_{k=1}^{\ell} \mathbf{1}\left[x'_{i_k} = w_k\right]\right) z_0^{i_1} \cdot t_0^{i_\ell - i_1 - (\ell-1)}\right| \geq \exp\left(-C_\rho n^{1/5}\log^5 n\right).$$

Combining the two cases proves Lemma 16.

4.2 Proof of Lemma 17

We now prove Lemma 17. We first state the following identity relating two multivariate polynomials, each of which is defined in terms of an arbitrary $f : \{0,1\}^\ell \to \mathbb{C}$. One of these involves the evaluation of f on the ℓ-bit (not necessarily consecutive) substrings of the source string x, and the other involves the expectation of f evaluated on the ℓ-bit substrings of a random trace $\boldsymbol{y} \sim \mathbf{Del}_\delta(x)$. This identity has now appeared in several places such as [13, 11] (see [13, Section 5.2] for a proof).

▶ **Fact 20.** *For every $f\colon \{0,1\}^\ell \to \mathbb{C}$, $x \in \{0,1\}^n$, $\rho \in [0,1]$, and $z \in \mathbb{C}^\ell$,*

$$\rho^\ell \sum_{0\le i_1<\cdots<i_\ell\le n-1} f(x_{i_1},\dots,x_{i_\ell})\big((1-\rho)+\rho z_1\big)^{i_1}\prod_{k=2}^{\ell}\big((1-\rho)+\rho z_k\big)^{i_k-i_{k-1}-1}$$
$$= \sum_{0\le j_1<\cdots<j_\ell\le n-1} \mathop{\mathbf{E}}_{\boldsymbol{y}\sim\mathbf{Del}_{1-\rho}(x)}\big[f(\boldsymbol{y}_{j_1},\dots,\boldsymbol{y}_{j_\ell})\big] z_1^{j_1}\prod_{k=2}^{\ell} z_k^{j_k-j_{k-1}-1}.$$

Letting $f(u_1,\dots,u_\ell)$ be the indicator function $\mathbf{1}[u=w]$ for some pattern $w \in \{0,1\}^\ell$, then performing a simple change of variable $z_i \mapsto \frac{z_i-(1-\rho)}{\rho}$, and then identifying the variables $z_3,\dots,z_\ell$ with the variable z_2, we obtain the following corollary.

▶ **Corollary 21.** *For every $\rho \in (0,1]$, $x \in \{0,1\}^n$, $w \in \{0,1\}^\ell$, and $(z_1,z_2) \in \mathbb{C}^2$,*

$$P_{x,w}(z_1,z_2) =$$
$$\rho^{-\ell} \sum_{0\le j_1<\cdots<j_\ell\le n-1} \mathop{\mathbf{E}}_{\boldsymbol{y}\sim\mathbf{Del}_{1-\rho}(x)}\Big[\prod_{k=1}^{\ell}\mathbf{1}\left[\boldsymbol{y}_{j_k}=w_k\right]\Big]\Big(\frac{z_1-(1-\rho)}{\rho}\Big)^{j_1}\Big(\frac{z_2-(1-\rho)}{\rho}\Big)^{j_\ell-j_1-(\ell-1)}. \tag{16}$$

Let $Q(z_1,z_2)$ be the bivariate polynomial on the right hand side of Equation (16). Observe that for every fixed z_1, viewing $Q(z_1,z_2)$ as a univariate polynomial in z_2, its z_2-coefficient of degree d (a univariate polynomial in z_1) can be estimated using d-local SQs. We will first prove that Q, as a univariate polynomial in the second variable z_2, is close to its low-degree truncation $Q_{\le d}$ (for a suitable choice of d), defined by

$$Q_{\le d}(z_1,z_2) :=$$
$$\rho^{-\ell} \sum_{\substack{0\le j_1<\cdots<j_\ell\le n-1:\\ j_\ell-j_1-(\ell-1)\le d}} \mathop{\mathbf{E}}_{\boldsymbol{y}}\Big[\prod_{k=1}^{\ell}\mathbf{1}\left[\boldsymbol{y}_{j_k}=w_k\right]\Big]\Big(\frac{z_1-(1-\rho)}{\rho}\Big)^{j_1}\Big(\frac{z_2-(1-\rho)}{\rho}\Big)^{j_\ell-j_1-(\ell-1)}, \tag{17}$$

when both z_1, z_2 belong to the domain in Lemma 16.

▷ Claim 22. Let C''_ρ be a constant, and $d_0 \ge C''_\rho(\ell+n^{1/5})+2\log n$. For every $z \in \{e^{i\theta} : |\theta| \le n^{-2/5}\} \cup [1-\rho, 1-\frac{3}{4}\rho]$ and $t \in [1-\rho, 1-\frac{3}{4}\rho]$, we have $|Q_{\le d_0}(z,t) - Q(z,t)| \le 4\cdot 2^{-d_0/2}$.

Proof. It suffices to show that for every $d \ge d_0$, the homogeneous degree-d (in the variable t) term of Q, that is,

$$\rho^{-\ell} \sum_{\substack{0\le j_1<\cdots<j_\ell\le n-1\\ 0\le j_\ell-j_1-(\ell-1)=d}} \mathbf{E}\Big[\prod_{k=1}^{\ell}\mathbf{1}\left[\boldsymbol{y}_{j_k}=w_k\right]\Big]\Big(\frac{z-(1-\rho)}{\rho}\Big)^{j_1}\Big(\frac{t-(1-\rho)}{\rho}\Big)^{j_\ell-j_1-(\ell-1)}, \tag{18}$$

is bounded by $2^{-d/2}$, as then we have $|Q(z,t) - Q_{\le d_0}(z,t)| \le \sum_{d>d_0} 2^{-d/2} = 4\cdot 2^{-d_0/2}$, as desired.

We now bound Equation (18) as follows. First, the expectation in each term of the summation can be bounded by 1. Second, writing z as $e^{i\theta}$ for some $|\theta| \le n^{-2/5}$, and using $|\cos\theta| \ge 1 - \theta^2/2$, we have

$$\begin{aligned}|z-(1-\rho)|^2 &= \big(\cos\theta - (1-\rho)\big)^2 + \sin^2\theta = 1 - 2(1-\rho)\cos\theta + (1-\rho)^2 \\ &= 2(1-\rho)(1-\cos\theta) + \rho^2 \le (1-\rho)\theta^2 + \rho^2.\end{aligned}$$

Using $|\theta| \le n^{-2/5}$ and $j_1 \le n$, when $z = e^{i\theta}$ for some $|\theta| \le n^{-2/5}$ we have that

$$\Big|\frac{z-(1-\rho)}{\rho}\Big|^{j_1} \le \left(1 + (1-\rho)\Big(\frac{\theta}{\rho}\Big)^2\right)^{j_1/2} \le e^{C'_\rho n^{1/5}} \tag{19}$$

for some constant C'_ρ. And when $z \in [1-\rho, 1-\frac{3}{4}\rho]$ we have $0 \le \frac{z-(1-\rho)}{\rho} \le 1/4$ and so Equation (19) is again satisfied (with room to spare). Similarly, for $t \in [1-\rho, 1-\frac{3}{4}\rho]$ we have $0 \le \frac{t-(1-\rho)}{\rho} \le 1/4$, and so $|\Big(\frac{t-(1-\rho)}{\rho}\Big)^d| \le 4^{-d}$.

Finally, the number of indices $0 \le j_1 < \cdots < j_\ell \le n-1$ with $j_\ell - j_1 - (\ell-1) = d$ is at most $n \cdot \binom{(\ell-2)+d}{\ell-2} \le n \cdot 2^{d+(\ell-2)}$. So the degree-$d$ term (18) can be bounded by

$$\rho^{-\ell} \cdot n \cdot 2^{d+\ell-2} \cdot e^{C'_\rho n^{1/5}} \cdot 4^{-d} \le n \cdot (2/\rho)^\ell \cdot e^{C'_\rho n^{1/5}} \cdot 2^{-d},$$

which is at most $2^{-d/2}$ whenever $d \ge C''_\rho(\ell + n^{1/5}) + 2\log n$, for some constant C''_ρ. ◁

We now describe our local SQ algorithm to approximate the low-degree polynomial $Q_{\le d}(z,t)$, for any $(z,t) \in \{e^{i\theta} : |\theta| \le n^{-2/5}\} \cup [1-\rho, 1-\frac{3}{4}\rho] \times [1-\rho, 1-\frac{3}{4}\rho]$. Set $d_0 := C''_\rho n^{1/5}\log^5 n \ge C''_\rho(\ell + n^{1/5}) + 2\log n$. Our d_0-local algorithm makes the following d_0-local queries:

$$\mathop{\mathbf{E}}_{\boldsymbol{y}\sim\mathbf{Del}_{1-\rho}(x)}\Big[\mathbf{1}\left[\boldsymbol{y}[j : j+d_0-1] = u\right]\Big] \text{ for every } u \in \{0,1\}^{d_0} \text{ and } j \in \{0,\dots,n-1\}.$$

Let $\widehat{p_{u,j}}$ be the estimate of $\mathbf{E}[\mathbf{1}\left[\boldsymbol{y}[j : j+d_0-1] = u\right]]$ that is received as a response to the query. For every fixed tuple $0 \le j_1 < \cdots < j_\ell \le n-1$ such that $j_\ell - j_1 - (\ell-1) \le d_0$, using the identity

$$\mathbf{E}\Big[\prod_{k=1}^{\ell}\mathbf{1}\left[\boldsymbol{y}_{j_k} = w_k\right]\Big] = \sum_{u\in\{0,1\}^{d_0}:\forall k\in[\ell]:u_{j_k-j_1+1}=w_k} \mathbf{E}\Big[\mathbf{1}\left[\boldsymbol{y}[j_1 : j_1+d_0-1] = u\right]\Big],$$

which is a sum of $2^{d_0-\ell}$ terms, the algorithm computes the estimate $\widehat{p_{u,j_1,\dots,j_\ell}}$ of $\mathbf{E}\left[\prod_{k=1}^{\ell}\mathbf{1}\left[\boldsymbol{y}_{j_k} = w_k\right]\right]$ (using the estimates $\widehat{p_{u,j_1}}$ of $\mathbf{E}[\mathbf{1}\left[\boldsymbol{y}[j_1 : j_1+d_0-1] = u\right]]$) by

$$\widehat{p_{w,j_1,\dots,j_\ell}} := \sum_{u\in\{0,1\}^{d_0}:\forall k\in[\ell]:u_{j_k-j_1+1}=w_k} \widehat{p_{u,j_1}},$$

for each $w \in \{0,1\}^\ell$ and tuple of indices $0 \le j_1 < \cdots < j_\ell \le n-1$ such that $j_\ell - j_1 - (\ell-1) \le d_0$. If the tolerance for each query is τ_0, then the error of each estimate $\widehat{p_{w,j_1,\dots,j_\ell}}$ is $\pm 2^{d_0-\ell}\cdot\tau_0$. Finally, the algorithm computes the estimate $\widehat{Q_{\le d_0}}(z,t)$ of $Q_{\le d_0}(z,t)$ using Equation (17), as

$$\widehat{Q_{\le d_0}}(z,t) := \rho^{-\ell} \sum_{\substack{0\le j_1<\cdots<j_\ell\le n-1:\\ j_\ell-j_1-(\ell-1)\le d_0}} \widehat{p_{w,j_1,\dots,j_\ell}}\Big(\frac{z-(1-\rho)}{\rho}\Big)^{j_1}\Big(\frac{t-(1-\rho)}{\rho}\Big)^{j_\ell-j_1-(\ell-1)}.$$

There are at most $n \cdot \binom{d_0+(\ell-1)}{\ell-1} \leq n \cdot 2^{d_0+(\ell-1)}$ such tuples. So the total error is $n \cdot 2^{d_0+(\ell-1)} \cdot 2^{d_0-\ell} \cdot \tau_0 \leq n \cdot 2^{2d_0} \cdot \tau_0$.

By Claim 22, we have that for every (z,t) in the domain specified in Lemma 17

$$\begin{aligned}
|\widehat{Q_{\leq d_0}}(z,t) - P_{x,w}(z,t)| &= |\widehat{Q_{\leq d_0}}(z,t) - Q(z,t)| \\
&\leq |\widehat{Q_{\leq d_0}}(z,t) - Q_{\leq d_0}(z,t)| + |Q_{\leq d_0}(z,t) - Q(z,t)| \\
&\leq n \cdot 2^{2d_0} \cdot \tau_0 + 4 \cdot 2^{-d_0/2} \\
&= 2^{2C''_\rho n^{1/5} \log^5 n} \tau_0 + \exp(-C''_\rho n^{1/5} \log^5 n).
\end{aligned}$$

Setting the tolerance parameter τ_0 to be $\exp(-C_\rho n^{1/5} \log^5 n)$ proves Lemma 17.

5 Average-case lower bounds

▶ **Theorem 23** (Average-case lower bound). *Fix any constant deletion rate $0 < \delta < 1$. Any ℓ-local SQ algorithm for average-case trace reconstruction must have tolerance $\tau_0 \leq O(\ell/\sqrt{n})$.*

Let x be an arbitrary fixed string in $\{0,1\}^n$ and let x' be the string obtained from x by flipping the bit $x_{n/2}$ in the middle. Let $q : \{0,1\}^n \to [-1,1]$ be any ℓ-junta query (which is not necessarily ℓ-local), i.e., there are $0 \leq i_1 < \ldots < i_\ell < n$ such that $q(x) = q'(x_{i_1}, \ldots, x_{i_\ell})$ for some $q' : \{0,1\}^\ell \to [-1,1]$. We will prove the following claim:

▷ Claim 24. Let $P_q := \mathbf{E}_{\boldsymbol{y} \sim \mathbf{Del}_\delta(x)}[q(\boldsymbol{y})]$ and $P'_q := \mathbf{E}_{\boldsymbol{y} \sim \mathbf{Del}_\delta(x')}[q(\boldsymbol{y})]$. Then $|P_q - P'_q| \leq O(\ell/\sqrt{n})$.

Proof. Let $\mathbf{R}$ be a ρ-biased random draw of a subset of $[0 : n-1]$ with $\mathbf{R} = \{\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{\boldsymbol{m}-1}\}$ for some $\boldsymbol{m} \leq n$. Given that the only difference between x and x' is the middle bit, we have

$$\left|P_q - P'_q\right| \leq 2 \cdot \mathbf{Pr}_{\mathbf{R}}\left[\boldsymbol{r}_{i_j} = n/2 \text{ for some } j \in [\ell]\right] \leq 2 \sum_{j \in [\ell]} \mathbf{Pr}_{\mathbf{R}}\left[\boldsymbol{r}_{i_j} = n/2\right].$$

Since $\delta \in (0,1)$ is a constant, $\mathbf{Pr}_{\mathbf{R}}[\boldsymbol{r}_i = n/2] \leq O(1/\sqrt{n})$ for any i, from which the claim follows. ◁

We now prove Theorem 23:

Proof. (of Theorem 23) Indeed we will show that any SQ algorithm for average-case trace reconstruction that uses ℓ-junta queries with tolerance τ must satisfy $\tau \leq O(\ell/\sqrt{n})$. To see this, consider any SQ algorithm for trace reconstruction that uses ℓ-junta queries with tolerance τ that is larger than the $O(\ell/\sqrt{n})$ in Claim 24. It follows from Claim 24 that, for any string $x \in \{0,1\}^n$, such an algorithm cannot distinguish between x and x'. As a result, such an algorithm fails to reconstruct $\boldsymbol{x} \sim \{0,1\}^n$ with probability at least $1/2$. ◀

6 Average-case upper bounds

▶ **Theorem 25** (Average-case upper bound). *Fix any constant deletion rate $0 < \delta < 1$. There is an SQ algorithm for average-case trace reconstruction that uses $\ell = O(\log n)$-local queries with tolerance $\tau = 1/\mathrm{poly}(n)$.*

We will prove Theorem 25 by showing that the algorithm in [14] can be simulated with local SQ queries. To do so, we will need to recall the smoothed analysis model.

▶ **Definition 26.** *Let x^{worst} be an unknown and arbitrary string in $\{0,1\}^n$ and $0 < \sigma < 1$ be a "smoothening parameter." Let $\boldsymbol{x}$ be generated by flipping every bit of x^{worst} independently with probability σ.*

*For parameters $\eta, \tau > 0$, a (T, η, τ)-*trace reconstruction algorithm in the smoothed analysis model (with smoothening parameter σ) *has the following guarantee: With probability at least $1 - \eta$ (over the random generation of $\boldsymbol{x}$ from x^{worst}), it is the case that the algorithm, given access to independent traces drawn from $\mathbf{Del}_\delta(\boldsymbol{x})$, outputs the string $\boldsymbol{x}$ with probability at least $1 - \tau$ (over the random traces drawn from $\mathbf{Del}_\delta(\boldsymbol{x})$). The time complexity as well as the number of traces is bounded by T.*

Observe that the average case trace reconstruction setting corresponds to the smoothed analysis setting with $\sigma = 1/2$ and x^{worst} set to the all zeros string (though any fixed choice of x^{worst} works equally well).

[14] gave a polynomial-time algorithm for trace reconstruction in the smoothed analysis setting. Taking $\sigma = 1/2$, the main result of [14] gives the following:

▶ **Theorem 27** (Theorem 1 in [14])**.** *There is an algorithm for trace reconstruction which for any η, τ and $\delta > 0$, has the following guarantee: With probability $1-\eta$ over $\boldsymbol{x}$ drawn uniformly at random from $\{0,1\}^n$, it is the case that the algorithm, given access to independent traces drawn from $\mathbf{Del}_\delta(\boldsymbol{x})$, outputs the string $\boldsymbol{x}$ with probability at least $1 - \tau$ (over the random traces drawn from $\mathbf{Del}_\delta(\boldsymbol{x})$). Its running time and sample complexity are upper bounded by*

$$T = \left(\frac{n}{\eta}\right)^{O\left(\frac{1}{(1-\delta)} \cdot \log\left(\frac{2}{(1-\delta)}\right)\right)}.$$

We begin with a short description of the algorithm in [14] (page 27 of the Arxiv version of [14]), giving only the level of detail necessary for the current paper. We set the following parameters:

$$\begin{aligned} k &= O(\log(n/\eta)), \quad \kappa = \left(\frac{1}{n} \cdot \left(\frac{1-\delta}{2}\right)^k\right)^{O(1/(1-\delta))}, \qquad \theta = (1-\delta)^2/2, \\ d &= \frac{C}{\theta}\left(\ln n + k \ln \frac{C}{\theta}\right), \qquad \Delta = \frac{\kappa}{2d^2 \cdot n \cdot \binom{d+k-2}{k-2}}. \end{aligned} \tag{20}$$

Set L to be the largest integer such that $\delta + L \cdot \delta \leq (1+\delta)/2$.

Given two strings $x \in \{0,1\}^n$ and $w \in \{0,1\}^k$, [14] define a univariate polynomial $\mathsf{SW}_{x,w}(\cdot)$. The precise formal definition of this polynomial is not important for us; rather, the following relation (Equation 6 in the Arxiv version of [14]) is sufficient for our purposes:

$$\mathop{\mathbf{E}}_{\boldsymbol{y} \sim \mathbf{Del}_{\delta'}(x)}[\#(w, \boldsymbol{y})] = (1-\delta')^k \cdot \mathsf{SW}_{x,w}(\delta'), \tag{21}$$

where $\#(w, \boldsymbol{y})$ is the number of times w appears a subword of $\boldsymbol{y}$. We remark to the reader that $\#(w, \boldsymbol{y})$ is a sum of k-local query functions (we will elaborate on this shortly). The algorithm in [14] proceeds as follows:

1. Define set $S := \{\delta, \delta + \Delta, \delta + 2\Delta, \ldots, \delta + L\Delta\}$.
2. For every $w \in \{0,1\}^k$ and $\delta' \in S$, the algorithm computes $\pm\kappa$-accurate estimates of $\mathsf{SW}_{x,w}(\delta')$, using Equation (21).
3. With these estimates of $\mathsf{SW}_{x,w}(\delta')$ (for $\delta' \in S$ and $w \in \{0,1\}^k$), the algorithm runs a linear program followed by a greedy algorithm to reconstruct the original string.

In particular, excluding the part of Step (2) in which the estimates of $\mathsf{SW}_{x,w}(\delta')$ are computed, the rest of the reconstruction algorithm is deterministic and does not use the traces. Note that the reason why the [14] algorithm does not immediately translate to a local SQ algorithm for us is the following: in our model the permissible statistical queries are with respect to $\boldsymbol{y} \sim \mathbf{Del}_\delta(x)$, whereas the [14] algorithm, as sketched above, uses estimates of probabilities corresponding to statistical queries over $\boldsymbol{y} \sim \mathbf{Del}_{\delta'}(x)$ for various values of $\delta' > \delta$.

Thus, to obtain a local SQ algorithm, it suffices to show that the values $\{\mathsf{SW}_{x,w}(\delta')\}_{\delta' \in S,\ w \in \{0,1\}^k}$ can be estimated using local SQ queries corresponding to $\mathbf{Del}_\delta(x)$. More precisely, we have the following claim whose proof is immediate from the description of the above algorithm and (21).

▷ Claim 28. For any $\delta' \in S$, to compute $\mathsf{SW}_{x,w}(\delta')$ to error κ, it is sufficient to estimate the value of $\mathbf{E}_{\boldsymbol{y} \sim \mathbf{Del}_{\delta'}(x)}[\#(w, \boldsymbol{y})]$ up to error τ', where

$$\tau' := \left(\frac{1}{n}\left(\frac{1-\delta}{2}\right)^k\right)^{O(1/(1-\delta))}. \tag{22}$$

Proof. We need to compute $\mathsf{SW}_{x,w}(\delta')$ for $\delta' \in S$ to error κ. By (21), it suffices to compute $\mathbf{E}_{\boldsymbol{y} \sim \mathbf{Del}_{\delta'}(x)}[\#(w, \boldsymbol{y})]$ to error $\kappa \cdot (1-\delta')^k$; noting that $\delta' \leq (1+\delta)/2$, the claim follows. ◁

The main technical lemma of this section is the following.

▶ **Lemma 29.** *For the parameters defined as above, the following holds: Given the values of all subword queries of length ℓ with tolerance $\tau'/2$ (with τ' defined in (22)) corresponding to $\mathbf{Del}_\delta(x)$, we can compute $\mathsf{SW}_{x,w}(\delta')$ for all $\delta' \in S$ and $w \in \{0,1\}^k$ to within error $\pm\kappa$. Here ℓ is defined to be*

$$\ell = \Theta\left(\frac{k}{1-\delta} \cdot \ln\left(\frac{2}{1-\delta}\right) + \frac{\ln n}{(1-\delta)}\right).$$

Before proving this lemma, we observe that Theorem 25 follows immediately from the lemma:

Proof. (of Theorem 25) For any constant $0 < \delta < 1$ and $\eta = n^{-\Theta(1)}$, by our choice of parameters we have $k = O(\log n)$ (see (20)). With this choice, $\ell = O(\log n)$ and $\tau' = n^{-\Theta(1)}$. By Lemma 29, using the values of all subword queries of length ℓ (with tolerance $\tau'/2$), we can compute $\mathsf{SW}_{x,w}(\delta')$ for all $\delta' \in S$ and $w \in \{0,1\}^k$ to within error $\pm\kappa$. By the guarantee of the algorithm in [14], this suffices to recover x. Thus, we get Theorem 25. ◀

6.1 Proof of Lemma 29

We start with the following observation.

▶ **Fact 30.** *For $\delta' \geq \delta$, let $0 \leq \beta_r = (1-\delta')/(1-\delta) \leq 1$. Then $\mathbf{Del}_{\delta'}(x) = \mathbf{Del}_{\beta_r}(\mathbf{Del}_\delta(x))$. In other words, we can simulate traces from the deletion channel $\mathbf{Del}_{\delta'}(\cdot)$ by first getting a trace from $\mathbf{Del}_\delta(\cdot)$ and then passing it through the deletion channel $\mathbf{Del}_{\beta_r}$.*

As stated earlier, we will assume that our original string x is padded with infinitely many 0-symbols to its right. This means that for any i, the i^{th} position of the trace is well-defined. We now consider the process of getting a trace $\boldsymbol{y}$ from $\mathbf{Del}_{\delta'}(x)$ given a trace $\boldsymbol{z} \sim \mathbf{Del}_\delta(x)$. We will do this by thinking of $\mathbf{Del}_{\beta_r}(\cdot)$ as a "selector process". We start with the following definition.

▶ **Definition 31.** *For a parameter $p \in (0,1)$ and integers $k > 0$ and $\ell \geq 0$, we define the distribution* $\mathsf{Hypernb}(p,k,\ell)$ *as follows: Define an infinite random string $\boldsymbol{w} = (\boldsymbol{w}_0, \dots)$ in $\{0,1\}^*$ where each bit is independently 0 with probability p and 1 with probability $(1-p)$. Let $\mathbf{i}_s$ be the location of the s^{th} one in $\boldsymbol{w}$. Then a sample from* $\mathsf{Hypernb}(p,k,\ell)$ *is given by* $(\mathbf{i}_k, \dots, \mathbf{i}_{k+\ell-1})$.

Finally, we say that an outcome from $\mathsf{Hypernb}(p,k,\ell)$ *is* t-bounded *if* $|\mathbf{i}_{k+\ell-1} - \mathbf{i}_k| \leq t$.

We note that for any fixed s, the process generating $\mathbf{i}_s$ is *memoryless*, in the sense that for any fixed r and s (with $r \geq s$), the random variable $\mathbf{i}_r - \mathbf{i}_s$ is distributed as a negative binomial random variable.

With the above definition, we can now state the following claim:

▷ Claim 32. Fix $\delta' \geq \delta$, $k \geq 1$, and $\ell \geq 0$. Let $\boldsymbol{y} \sim \mathbf{Del}_{\delta'}(x)$ and $\boldsymbol{z} \sim \mathbf{Del}_{\delta}(x)$. For $\beta_r = (1-\delta')/(1-\delta)$, let $(\mathbf{i}_k, \dots, \mathbf{i}_{k+\ell-1}) \sim \mathsf{Hypernb}(\beta_r, k, \ell)$. Then the distribution of $(\boldsymbol{y}_k, \dots, \boldsymbol{y}_{k+\ell-1})$ is identical to the distribution of $(\boldsymbol{z}_{\mathbf{i}_k}, \dots, \boldsymbol{z}_{\mathbf{i}_{k+\ell-1}})$.

Proof. The proof is essentially obvious from Fact 30 and Definition 31. In particular, from Fact 30, given $\boldsymbol{z} \sim \mathbf{Del}_{\delta}(x)$, to get $\boldsymbol{y} \sim \mathbf{Del}_{\delta'}(x)$, we need to simulate the deletion channel $\mathbf{Del}_{\beta_r}$ on the string $\boldsymbol{z}$. By definition of the deletion channel $\mathbf{Del}_{\beta_r}$, the location of the positions $(k, \dots, k+\ell-1)$ is given by $(\mathbf{i}_k, \dots, \mathbf{i}_{k+\ell-1})$ sampled from $\mathsf{Hypernb}(p,k,\ell)$. This finishes the proof. ◁

We next need a lower bound on the probability that $\mathsf{Hypernb}(p,k,\ell)$ is bounded. To obtain this, we first state a tail bound on negative binomial random variables:

▷ Claim 33 ([7]). Let $\mathsf{Negbin}(m,p)$ be a negative binomially distributed random variable with parameters m and p, i.e. it is the number of trials needed to get m heads from independent coin tosses with heads probability p. Then $\mathbf{E}[\mathsf{Negbin}(m,p)] = m/p$ and furthermore, for any $t > 1$,

$$\mathbf{Pr}\big[\mathsf{Negbin}(m,p) > tm/p\big] \leq \exp\left(-\frac{tm(1-1/t)^2}{2}\right).$$

From this we can obtain the following claim which lower bounds the probability that $\mathsf{Hypernb}(\beta_r,k,\ell)$ is s-bounded.

▷ Claim 34. For any k, an outcome $(\mathbf{i}_k, \dots, \mathbf{i}_{k+\ell-1}) \sim \mathsf{Hypernb}(\beta_r,k,\ell)$ is s-bounded with probability at least $1-\xi$ for $s = t(\ell-1)/\beta_r$, where $\xi = \exp(-t(\ell-1)/8)$ for $t \geq 2$.

Proof. The gap $\mathbf{i}_{k+\ell-1} - \mathbf{i}_k$ is a negative binomial random variable which is distributed as $\mathsf{Negbin}(\ell-1, \beta_r)$. Thus, by Claim 33, it follows that

$$\mathbf{Pr}\Big[\mathbf{i}_{k+\ell-1} - \mathbf{i}_k > \frac{t(\ell-1)}{\beta_r}\Big] \leq \exp\left(\frac{-t(\ell-1)(1-1/t)^2}{2}\right).$$

For $t \geq 2$, we can simplify the upper bound as

$$\mathbf{Pr}\Big[\mathbf{i}_{k+\ell-1} - \mathbf{i}_k > \frac{t(\ell-1)}{\beta_r}\Big] \leq \exp\left(\frac{-t(\ell-1)}{8}\right).$$

Defining ξ as $\exp(\frac{-t(\ell-1)}{8})$, we get the claim. ◁

We now state the following technical claim.

▷ Claim 35. Given the value of all ℓ-local subword queries for deletion channel $\mathbf{Del}_\delta(x)$ with tolerance η, we can compute the value of all ℓ'-local subword queries for $\mathbf{Del}_{\delta'}(x)$ with tolerance τ' where

$$\eta = \tau'/2; \quad \ell = C \cdot \left(\frac{\ell'}{1-\delta'} \cdot \ln\left(\frac{2}{1-\delta}\right) + \frac{\ln n}{(1-\delta')} \right), \quad \text{for a suitably large constant } C.$$

Proof. Fix any $w \in \{0,1\}^{\ell'}$ and consider the quantity

$$p'_{x,k,w} := \mathop{\mathbf{Pr}}_{\boldsymbol{y} \sim \mathbf{Del}_{\delta'}(x)} [(\boldsymbol{y}_k, \ldots, \boldsymbol{y}_{k+\ell'-1}) = w].$$

Then, by Claim 32, it follows that

$$p'_{x,k,w} := \mathop{\mathbf{Pr}}_{\boldsymbol{z} \sim \mathbf{Del}_\delta(x), (\mathbf{i}_k, \ldots, \mathbf{i}_{k+\ell'-1}) \sim \mathsf{Hypernb}(\beta_r, k, \ell')} [\boldsymbol{z}_{\mathbf{i}_1}, \ldots, \boldsymbol{z}_{\mathbf{i}_{k+\ell'-1}} = w], \tag{23}$$

where $\beta_r = (1-\delta')/(1-\delta)$. Define the parameter $t = C \cdot \left(\frac{1}{1-\delta} \cdot \ln\left(\frac{2}{1-\delta}\right) + \frac{\ln n}{\ell'(1-\delta)} \right)$, where the constant C is set so that $\exp\left(\frac{t(\ell'-1)}{8}\right) = \frac{\tau'}{2}$. As $\ell = t(\ell'-1)/\beta_r$, by Claim 34 we have

$$\mathbf{Pr}[\mathbf{i}_{\ell'+k-1} - \mathbf{i}_{\ell'} > \ell] \leq \exp\left(\frac{t(\ell'-1)}{8}\right) = \frac{\tau'}{2}. \tag{24}$$

Now, define $\mathcal{E}$ as the event (over the samples $(\mathbf{i}_k, \ldots, \mathbf{i}_{k+\ell'-1})$) that $|\mathbf{i}_{k+\ell'-1} - \mathbf{i}_k| \leq \ell$. We now re-express

$$\begin{aligned} p'_{x,k,w} \quad &= \quad \mathop{\mathbf{Pr}}_{\boldsymbol{z} \sim \mathbf{Del}_\delta(x), (\mathbf{i}_k, \ldots, \mathbf{i}_{k+\ell'-1})} \left[\boldsymbol{z}_{\mathbf{i}_1}, \ldots, \boldsymbol{z}_{\mathbf{i}_{k+\ell'-1}} = w \ \wedge \mathcal{E}\right] \\ &+ \quad \mathop{\mathbf{Pr}}_{\boldsymbol{z} \sim \mathbf{Del}_\delta(x), (\mathbf{i}_k, \ldots, \mathbf{i}_{k+\ell'-1})} \left[\boldsymbol{z}_{\mathbf{i}_1}, \ldots, \boldsymbol{z}_{\mathbf{i}_{k+\ell'-1}} = w \ \wedge \overline{\mathcal{E}}\right]. \end{aligned}$$

From the bound (24), the second term is at most $\tau'/2$ in magnitude and thus,

$$\left| p'_{x,k,w} - \mathop{\mathbf{Pr}}_{\boldsymbol{z} \sim \mathbf{Del}_\delta(x), (\mathbf{i}_k, \ldots, \mathbf{i}_{k+\ell'-1})} \left[\boldsymbol{z}_{\mathbf{i}_1}, \ldots, \boldsymbol{z}_{\mathbf{i}_{k+\ell'-1}} = w \ \wedge \mathcal{E}\right] \right| \leq \tau'/2.$$

Furthermore, for any particular outcome of $(\mathbf{i}_k, \ldots, \mathbf{i}_{k+\ell'-1})$ for which event $\mathcal{E}$ happens, the quantity $\mathbf{Pr}_{\boldsymbol{z} \sim \mathbf{Del}_\delta(x)}[\boldsymbol{z}_{\mathbf{i}_1}, \ldots, \boldsymbol{z}_{\mathbf{i}_{k+\ell'-1}} = w]$ is a ℓ-local subword query. Since we have the value of all ℓ-local subword queries up to error $\tau'/2$, we can compute $p'_{x,k,w}$ to error τ'. ◁

Proof. (of Lemma 29) By Claim 32, to compute $\mathsf{SW}_{x,w}(\delta')$ to error $\pm\kappa$, it suffices to compute $\mathbf{E}_{\boldsymbol{y} \sim \mathbf{Del}_{\delta'}(x)}[\#(w, \boldsymbol{y})]$ for every $w \in \{0,1\}^{\ell'}$ up to error $\pm\tau'$ where τ' is defined in (22). Now, by Claim 35, for any given $\delta' \geq \delta$, to compute $\mathbf{E}_{\boldsymbol{y} \sim \mathbf{Del}_{\delta'}(x)}[\#(w, \boldsymbol{y})]$ to error τ, it suffices to have the value of all ℓ-local subword queries to error $\tau'/2$ where

$$\ell = C \cdot \left(\frac{\ell'}{1-\delta'} \cdot \ln\left(\frac{2}{1-\delta}\right) + \frac{\ln n}{(1-\delta')} \right).$$

Since $\delta' \leq (1+\delta)/2$, it follows that

$$\ell \leq C \cdot \left(\frac{2\ell'}{1-\delta} \cdot \ln\left(\frac{2}{1-\delta}\right) + \frac{2\ln n}{(1-\delta)} \right).$$

Thus, if we have the value of all k-local subword queries to error $\tau'/2$, where k is set to

$$k = \Theta\left(\frac{\ell'}{1-\delta} \cdot \ln\left(\frac{2}{1-\delta}\right) + \frac{\ln n}{(1-\delta)} \right),$$

we can recover x. This finishes the proof. ◀

References

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

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

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

4 Peter Borwein, Tamás Erdélyi, and Géza Kós. Littlewood-type problems on $[0, 1]$. *Proc. London Math. Soc. (3)*, 79(1):22–46, 1999. doi:10.1112/S0024611599011831.

5 Tatiana Brailovskaya and Miklós Z. Rácz. Tree trace reconstruction using subtraces. *J. Appl. Probab.*, 60(2):629–641, 2023. doi:10.1017/jpr.2022.81.

6 Joshua Brakensiek, Ray Li, and Bruce Spang. Coded trace reconstruction in a constant number of traces. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 482–493, 2020. doi:10.1109/FOCS46700.2020.00052.

7 Daniel G. Brown. How I wasted too long finding a concentration inequality for sums of geometric variables. Available at https://uwspace.uwaterloo.ca/bitstream/handle/10012/17210/negbin.pdf?sequence=1, 2011.

8 Diptarka Chakraborty, Debarati Das, and Robert Krauthgamer. Approximate trace reconstruction via median string (in average-case). In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 213 of *LIPIcs*, pages 11:1–11:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

9 Z. Chase and Y. Peres. Approximate trace reconstruction of random strings from a constant number of traces. Available at arXiv:2107.06454, 2021.

10 Zachary Chase. New lower bounds for trace reconstruction. *Ann. Inst. H. Poincaré Probab. Statist.*, 57(2):627–643, 2021.

11 Zachary Chase. Separating words and trace reconstruction. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 21–31. ACM, 2021.

12 Xi Chen, Anindya De, Chin Ho Lee, Rocco A. Servedio, and Sandip Sinha. Polynomial-time trace reconstruction in the low deletion rate regime. In *12th Innovations in Theoretical Computer Science Conference*, volume 185, pages 20:1–20:20, 2021.

13 Xi Chen, Anindya De, Chin Ho Lee, Rocco A. Servedio, and Sandip Sinha. Polynomial-time trace reconstruction in the smoothed complexity model. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 54–73, 2021.

14 Xi Chen, Anindya De, Chin Ho Lee, Rocco A. Servedio, and Sandip Sinha. Near-optimal average-case approximate trace reconstruction from few traces. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA 2022)*, pages 779–821, 2022.

15 Kuan Cheng, Elena Grigorescu, Xin Li, Madhu Sudan, and Minshen Zhu. On k-mer-based and maximum likelihood estimation algorithms for trace reconstruction. *CoRR*, abs/2308.14993, 2023. doi:10.48550/arXiv.2308.14993.

16 Mahdi Cheraghchi, Ryan Gabrys, Olgica Milenkovic, and João Ribeiro. Coded trace reconstruction. *IEEE Trans. Inform. Theory*, 66(10):6084–6103, 2020. doi:10.1109/TIT.2020.2996377.

17 Sami Davies, Miklós Z. Rácz, and Cyrus Rashtchian. Reconstructing trees from traces. In Alina Beygelzimer and Daniel Hsu, editors, *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 961–978. PMLR, 2019. URL: http://proceedings.mlr.press/v99/davies19a.html.

18 Sami Davies, Miklos Z. Rácz, Cyrus Rashtchian, and Benjamin G. Schiffer. Approximate trace reconstruction: Algorithms. In *IEEE International Symposium on Information Theory*, 2021.

19 Anindya De, Ryan O'Donnell, and Rocco A. Servedio. Optimal mean-based algorithms for trace reconstruction. In *Proceedings of the 49th ACM Symposium on Theory of Computing (STOC)*, pages 1047–1056, 2017.

20 Elena Grigorescu, Madhu Sudan, and Minshen Zhu. Limitations of mean-based algorithms for trace reconstruction at small distance. In *IEEE International Symposium on Information Theory*, 2021.

21 Lisa Hartung, Nina Holden, and Yuval Peres. Trace reconstruction with varying deletion probabilities. In *Proceedings of the Fifteenth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2018, New Orleans, LA, USA, January 8-9, 2018.*, pages 54–61, 2018.

22 Nina Holden and Russell Lyons. Lower bounds for trace reconstruction. *Ann. Appl. Probab.*, 30(2):503–525, 2020. doi:10.1214/19-AAP1506.

23 Nina Holden, Robin Pemantle, and Yuval Peres. Subpolynomial trace reconstruction for random strings and arbitrary deletion probability. In *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 1799–1840. PMLR, 2018.

24 Nina Holden, Robin Pemantle, Yuval Peres, and Alex Zhai. Subpolynomial trace reconstruction for random strings and arbitrary deletion probability. *Mathematical Statistics and Learning*, 2(3/4):275–309, 2019.

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

26 V. V. Kalashnik. Reconstruction of a word from its fragments. *Computational Mathematics and Computer Science (Vychislitel'naya matematika i vychislitel'naya tekhnika), Kharkov*, 4:56–57, 1973.

27 M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

28 Akshay Krishnamurthy, Arya Mazumdar, Andrew McGregor, and Soumyabrata Pal. Trace reconstruction: Generalized and parameterized. In *27th Annual European Symposium on Algorithms, ESA 2019*, volume 144 of *LIPIcs*, pages 68:1–68:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

29 Vladimir Levenshtein. Efficient reconstruction of sequences. *IEEE Transactions on Information Theory*, 47(1):2–22, 2001.

30 Vladimir Levenshtein. Efficient reconstruction of sequences from their subsequences or supersequences. *Journal of Combinatorial Theory Series A*, 93(2):310–332, 2001.

31 Kayvon Mazooji and Ilan Shomorony. Substring density estimation from traces. In *IEEE International Symposium on Information Theory, ISIT 2023, Taipei, Taiwan, June 25-30, 2023*, pages 803–808. IEEE, 2023. doi:10.1109/ISIT54713.2023.10206758.

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

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

34 Shyam Narayanan. Improved algorithms for population recovery from the deletion channel. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1259–1278. SIAM, 2021. doi:10.1137/1.9781611976465.77.

35 Shyam Narayanan and Michael Ren. Circular Trace Reconstruction. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, pages 18:1–18:18, 2021.

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

37 Yuval Peres and Alex Zhai. Average-case reconstruction for the deletion channel: Subpolynomially many traces suffice. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 228–239. IEEE Computer Society, 2017.

38 Ittai Rubinstein. Average-case to (shifted) worst-case reduction for the trace reconstruction problem. In *50th International Colloquium on Automata, Languages, and Programming*, volume 261 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 102, 20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/lipics.icalp.2023.102.

39 Jin Sima and Jehoshua Bruck. Trace reconstruction with bounded edit distance. In *IEEE International Symposium on Information Theory*, 2021. Manuscript, available at arXiv:2102.05372.