An Improved Sketching Bound for Edit Distance

Ce Jin* Jelani Nelson† Kewen Wu‡

Abstract

We provide improved upper bounds for the simultaneous sketching complexity of edit distance. Consider two parties, Alice with input $x \in \Sigma^n$ and Bob with input $y \in \Sigma^n$, that share public randomness and are given a promise that the edit distance $\mathsf{ed}(x, y)$ between their two strings is at most some given value k. Alice must send a message sx and Bob must send sy to a third party Charlie, who does not know the inputs but shares the same public randomness and also knows k. Charlie must output $\mathsf{ed}(x, y)$ precisely as well as a sequence of $\mathsf{ed}(x, y)$ edits required to transform x into y. The goal is to minimize the lengths $|sx|, |sy|$ of the messages sent.

The protocol of Belazzougui and Zhang (FOCS 2016), building upon the random walk method of Chakraborty, Goldenberg, and Koucký (STOC 2016), achieves a maximum message length of $\tilde{O}(k^8)$ bits, where $\tilde{O}(\cdot)$ hides $\text{poly}(\log n)$ factors. In this work we build upon Belazzougui and Zhang's protocol and provide an improved analysis demonstrating that a slight modification of their construction achieves a bound of $\tilde{O}(k^3)$.

1 Introduction

The edit distance $\mathsf{ed}(x, y)$ between two strings is defined to be the minimum number of character insertions, deletions, or substitutions required to transform x into y. It is one of the most well-studied distance measures on strings, with applications in information retrieval, natural language processing, and bioinformatics. If x, y are each at most length n, the textbook Wagner-Fischer algorithm computes $\mathsf{ed}(x, y)$ exactly in $O(n^2)$ time, with the only improvement since being by a $\log n$ factor due to Masek and Paterson [MP80]. It has since been shown that an $O(n^{2-\epsilon})$ time algorithm does not exist for any constant $\epsilon > 0$ unless the *Strong Exponential Time Hypothesis* fails [BI18]. Since the work of [MP80], several subsequent works have considered setups beyond offline exact algorithms for edit distance, such as faster approximation algorithms [AO12, AKO10, CDG+18, BR20, KS20b, AN20], metric embeddings [OR07, CK06, KN06, KR09], smoothed complexity [AK12, BSS20], quantum algorithms [BEG+18], sublinear time algorithms for gap versions [BJKK04, GKS19, BCR20, KS20a], and communication complexity and sketching/streaming [CGK16, BZ16, CJLW18, Hae19, CL20]. In this work we focus on communication complexity, and specifically *simultaneous communication complexity*.

In the communication model, Alice has input string x and Bob has y. They, or a third party, would like to compute $\mathsf{ed}(x, y)$ as well as a minimum length sequence of edits for transforming x into y. We consider the setting of shared public randomness amongst all parties. The one-way setting in which Alice sends a single message to Bob, who must then output $\mathsf{ed}(x, y)$, is known as the *document exchange problem* and has a long history. In the promise version of the problem for which we are promised $\mathsf{ed}(x, y) \le k$, Orlitsky [Orl91] gave a deterministic protocol in which Alice only sends $O(k \log(n/k))$ bits in the case of binary strings, which is optimal, with the downside that Bob's running time to process her message is exponential. Haeupler recently used public randomness to improve Bob's running time to polynomial with the same asymptotic message length, and it is now known that a polynomial-time recovery algorithm is achievable deterministically if one increases the message length to $O(k \log^2(n/k))$ [CJLW18, Hae19]. Belazzougui and Zhang [BZ16] studied the harder *simultaneous communication* model in which Alice and Bob each send messages to a third party Charlie, who knows neither string but shares knowledge of the public randomness, and Charlie must output

*MIT. `cejin@mit.edu`. Supported by an Akamai Presidential Fellowship.

†UC Berkeley. `minilek@berkeley.edu`. Supported by NSF award CCF-1951384, ONR grant N00014-18-1-2562, ONR DORECG award N00014-17-1-2127, and a Google Faculty Research Award.

‡UC Berkeley. `shlw_kevin@hotmail.com`.

$\mathsf{ed}(x, y)$ as well as the edits required to transform x into y. In this model they gave a protocol in which each player sends $O(k^8 \log^5 n) = \tilde{O}(k^8)$ bits.[1]

Definition 1.1 (Problem $\mathscr{Q}_{n,k,\delta}$)**.** Alice and Bob and a referee share public randomness. Alice (resp., Bob) gets a length-n input string x (resp., y) over alphabet Σ, and then sends a "sketch" $sx \in \{0,1\}^*$ (resp., sy) to the referee. We say the *size* of the sketch is maximum length of strings sx and sy. After receiving the sketches sx and sy,

- if $\mathsf{ed}(x, y) \le k$, the referee needs to compute $\mathsf{ed}(x, y)$ as well as an optimal edit sequence from x to y, with success probability at least $1 - \delta$;
- if $\mathsf{ed}(x, y) > k$, the referee needs to report "error", with success probability at least $1 - \delta$.

Main contribution. We build upon and improve techniques developed in [BZ16] to show that a very slight modification of their protocol needs a sketch size of only $\tilde{O}(k^3)$ bits to solve problem $\mathscr{Q}_{n,k,\delta}$. More precisely, the bound is $O(k^3 \log^2(n/\delta) \log n)$ bits.[2]

1.1 Proof Overview

We provide a high-level description of the previous results [CGK16,BZ16] that we build on, and then briefly describe our new ideas.

CGK random walk. The previous sketching result [BZ16] uses a random walk technique developed in [CGK16]. Given two input strings x, y of length n, we append them with infinitely many zeros and initialize two pointers $i = 1, j = 1$. In each step t, we first append $x[i]$ to Alice's output tape (and append $y[j]$ to Bob's output tape), and then increment i by $r_t(x[i]) \in \{0,1\}$, and increment j by $r_t(y[j]) \in \{0,1\}$, where $r_t \colon \Sigma \to \{0,1\}$ is a random function. The process continues for $3n$ steps and we consider the evolution of $i - j$, i.e., the distance between the two pointers during this random process. Observe that when $x[i] \ne y[j]$, the change of $i - j$ is a mean-zero random variable in $\{-1, 0, +1\}$ (and we call this a *progress step*); while when $x[i] = y[j]$, the difference $i - j$ will not change.

The main result of [CGK16] shows that the number of progress steps in this random process is at least $\mathsf{ed}(x, y)/2$, and at most $O(\mathsf{ed}(x, y)^2)$ with constant probability. This property was used to design a sketching protocol (with public randomness for generating r_t) for estimating $\mathsf{ed}(x, y)$ up to a quadratic factor error by applying an approximate Hamming distance sketching protocol to the two strings generated by the random walk (where a progress step corresponds to a Hamming mismatch between Alice's and Bob's output strings).

[BZ16] algorithm. The key idea of [BZ16] is the following. A CGK random walk naturally induces a non-intersecting matching between the input strings: we view x and y as a bipartite graph, where if (i, j) is an edge then $x[i] = y[j]$ and i, j are the pointers in some step of the walk. In particular, this matching can be viewed as an edit sequence where a character is unchanged if it is matched.

Using an exact Hamming sketch protocol (with sketch size near-linear in the number of Hamming errors), the referee can recover this matching, as well as all the unmatched characters. Although this matching may not correspond to an edit sequence of optimal length, [BZ16] shows: suppose we obtain *multiple* matchings by running i.i.d. CGK random walks. Then,

(a) if the *intersection* of their matched edges is contained in an optimal matching, then one can extract enough information from the matchings and unmatched characters to recover an optimal edit sequence using dynamic programming;

(b) if we generate $\mathrm{poly}(\mathsf{ed}(x, y), \log n)$ many i.i.d. CGK random walks, then the precondition of Item (a) is satisfied with constant probability.

[1] We use $\tilde{O}(f)$ throughout this paper to denote $f \cdot \mathrm{polylog}(n)$.

[2] We remark that both the algorithm of [BZ16] and our improved algorithm are time-efficient, and work in the more restrictive setting where Alice and Bob have only $\mathrm{poly}(k \log(n/\delta))$ memory and receive the input strings in a *streaming* fashion.

Our improvements. We obtain our result by improving the dependence on $\mathsf{ed}(x, y)$ in Item (b) described above. In particular, we reduce the number of required random walks. Our improvements come from two parts.

To obtain the first improvement, we observe that [BZ16]'s algorithm relies on the following two events happening. The first is that, for every edge that does appear in a (fixed) optimal matching, there should be one of the sampled CGK random walks that misses this edge. The second is that the CGK random walks should have few progress steps. In [BZ16], they pay a union bound over the two events to make sure *all CGK random walks* are good for the decoder. This introduces a large dependence on $\mathsf{ed}(x, y)$, mainly due to the fact that the number of Hamming errors in a CGK random walk has a heavy-tailed distribution. We manage to avoid this by arguing that these two events happen simultaneously (see Lemma 3.6) with decent probability, and then modifying the decoding algorithm to only consider *those good CGK random walks*.

The second improvement comes from improved analysis for Item (b), which depends on the following property of the CGK random walk [BZ16, Lemma 16] (see Subsection 3.3 for how this property can be used): informally, if a string $X[1..L]$ has a certain kind of self-similarity (for example, it is periodic), then with some nontrivial probability, a CGK random walk on X itself starting with two pointers $i = 2, j = 1$ will not pass through the state $(i = L, j = L)$. To be more precise, if there is a non-intersecting matching between $X[1..L]$ and itself, where every matched edge (I, J) satisfies $I > J$, and the number of singletons (unmatched characters) is at most K, then the CGK random walk will miss $(i = L, j = L)$ with $\Omega(1/K^2)$ probability.[3]

We use a more technical analysis to improve the bound to $\Omega(1/K)$ (see Proposition 3.9). Now we informally describe our main idea. Starting from the state $(i = 2, j = 1)$, with at least $\Omega(1/K)$ probability it will first reach a state (i, j) with $i - j > d_0 = \Theta(K)$ before reaching $i - j = 0$ (note that $i - j$ can never become negative). Then we will show that with good probability $i - j$ will remain in the range $[d_0/2, 3d_0/2]$. To do this, we show an $O(d_0^2)$ upper bound on the expected total number of progress steps, and use the fact that the expected deviation produced by a P-step one-dimensional random walk is $O(\sqrt{P})$.

To bound the expected total number of progress steps, we divide the evolution of the state (i, j) into several phases, where in each phase the pointers move from a *stable state* (i, j) to another *stable state* (i', j'), satisfying $j' \geq i$ and $i' \geq 2j - i$. Here, a *stable state* (i, j) informally means that we have a good upper bound of $\mathsf{ed}(X[j..i - 1], X[i..2i - j - 1])$ in terms of the number of singletons in the range $[j..2i - j - 1]$ (for example, if X is "close" to a string with period p, and $i - j$ is approximately a multiple of p, then (i, j) is a stable state). We will bound the expected number of progress steps in one phase by $O\left((i - j) \cdot S + S^2\right)$, where S denotes the number of singletons in the range $[j..i' - 1]$. We can see the sum of S over all phases is at most $2K$ since each singleton is counted at most twice. Hence, summing up over all phases would give the desired $O(K^2)$ upper bound, if we assume $i - j = \Theta(d_0)$. Although this assumption may lead to circular reasoning, we can get around this issue by a more careful argument.

Organization. We give several needed definitions in Section 2. In Section 3 we state and analyze our sketching algorithm, which as mentioned, is mostly similar to [BZ16] but with small modifications. Section 4 is devoted to our main technical lemma. In comparison with the proof overview, Section 3 is for the first improvement and Section 4 is for the second improvement. Then we discuss limits on our approach and further problems in Section 5. The lower bounds and some of the technical proofs are deferred to the appendix.

2 Preliminaries

In this section we introduce formal definitions.

[3] There is a subtle gap in the proof of [BZ16, Lemma 16]. On page 18 of their full version, they bounded the number of progress steps in two cases: (1) at least one of the pointers is not in any cluster, and (2) both of the two pointers are in the same cluster. (Their terminology *cluster* refers to a contiguous sequence of matched edges with no singletons in-between.) However, they did not analyze the case where the two pointers are separated in different clusters, and it was not clear to us how to repair that gap using the techniques developed in [BZ16].

2.1 Notations

Let $[n]$ denote $\{1, 2, \ldots, n\}$, and let $[l..r]$ denote $\{l, l+1, \ldots, r\}$. Let $\circ$ denote string concatenation. Let $\mathbb{N}$ denote the set of natural numbers $\{0, 1, \ldots\}$. We consider sketching protocols for strings in Σ^n in this work, where Σ denotes the alphabet. We assume $|\Sigma| \le \mathrm{poly}(n)$ and $0 \in \Sigma$.[4]

For a string $s \in \Sigma^n$ and index $1 \le i \le n$, $s[i]$ (or sometimes s_i) denotes the i-th character of s. For $1 \le i \le j \le n$, $s[i..j]$ denotes the substring $s[i] \circ s[i+1] \circ \cdots \circ s[j]$. If $i > j$ then $s[i..j]$ is the empty string.

2.2 Edit Distance

Definition 2.1 (Edit distance $\mathsf{ed}(\cdot,\cdot)$)**.** The *edit distance* between two strings x and y, denoted by $\mathsf{ed}(x, y)$, is the minimum number of edits (insertions, deletions, and substitutions[5]) required to transform x to y.

We note the following simple facts about edit distance.

Fact 2.2. *Let x and y be two strings of length n and m respectively. Then $\mathsf{ed}(x, y) \ge |n - m|$.*

Proposition 2.3. *Let x, y be two length-n strings. Let x' be any (not necessarily contiguous) subsequence of x. Then $\mathsf{ed}(x, y) \le 2 \cdot \mathsf{ed}(x', y)$.*

Proof. Since $\mathsf{ed}(x', y) \ge n - \mathsf{LCS}(x', y) \ge n - \mathsf{LCS}(x, y)$, we have $\mathsf{ed}(x, y) \le 2 \cdot (n - \mathsf{LCS}(x, y)) \le 2 \cdot \mathsf{ed}(x', y)$. $\square$

Definition 2.4 (Matching induced by edit sequence $\mathcal{M}(S)$)**.** Given strings x, y and an edit sequence S, we can construct a bipartite graph between x and y, where every character in x that is not substituted nor deleted is connected by an edge to its counterpart in y. These edges form a non-intersecting matching, which we denote by $\mathcal{M}(S)$. Moreover, when S achieves optimal edit distance, we say $\mathcal{M}(S)$ is an *optimal matching*.

We show the following properties of an optimal matching, the proof of which is deferred to Appendix C.

Lemma 2.5. *Let x, y be two strings. Let S be an optimal edit sequence and $\mathcal{M}(S)$ be its corresponding optimal matching.*

(1) If $(i, j) \in \mathcal{M}(S)$, then $|i - j| \le \mathsf{ed}(x, y)$.

(2) If $u' \le u$ and $v' \le v$ and $u - u' + 1 = v - v' + 1 =: L$, then the number of matched edges with both endpoints in $x[u'..u]$ and $y[v'..v]$ is at least $L - 3 \cdot \mathsf{ed}(x, y) - |u - v|$, i.e.,

$$\left|\mathcal{M}(S) \cap \left([u'..u] \times [v'..v]\right)\right| \ge L - 3 \cdot \mathsf{ed}(x, y) - |u - v|.$$

Though there may be multiple optimal matchings, the following definition specifies a canonical one.

Definition 2.6 (Greedy optimal matching $\mathcal{M}$, [BZ16])**.** Let x, y be two strings. For each edit sequence S achieving optimal edit distance, let $\mathcal{M}(S)$ be the matching induced by S. Then the *greedy optimal matching* $\mathcal{M}$ is defined to be the smallest $\mathcal{M}(S)$ in lexicographical order. Specifically, we represent $\mathcal{M}(S)$ as a sequence of (i, j) pairs then sort the sequence lexicographically, and the greedy optimal matching is such that this sorted sequence is as lexicographically small as possible.

This greedy optimal matching enjoys some extra properties, which can be easily proved.

Lemma 2.7 ([BZ16])**.** *Let x, y be two strings and $\mathcal{M}$ be their greedy optimal matching.*

(1) If $(i, j) \in \mathcal{M}$ and $x[i+1] = y[j+1]$, then $(i+1, j+1) \in \mathcal{M}$.

(2) If $x[u'..u] = y[v'..v]$ and $(i, j), (i', j') \in \mathcal{M} \cap \left([u'..u] \times [v'..v]\right)$ are two matched edges in $x[u'..u], y[v'..v]$, then $((i - j) - (u - v)) \cdot ((i' - j') - (u - v)) \ge 0$. Moreover, when the equality holds we have $(u, v) \in \mathcal{M}$.

[4] For larger alphabet the algorithm still works but some $\log n$ terms in the bounds become $\log|\Sigma|$. For example, the sketch size will be $O\left(k^3 \log(n|\Sigma|/\delta) \log(n/\delta) \log n\right)$. Alternatively, the parties can hash Σ into a new alphabet of size $O(n^2/\gamma)$ and have no hash collisions on the characters appearing in x, y with probability at least $1 - \gamma$.

[5] There is another definition of edit distance, denoted by $\mathsf{ed}'(x, y)$, where only insertions and deletions are allowed. We have $\mathsf{ed}(x, y) \le \mathsf{ed}'(x, y) \le 2 \cdot \mathsf{ed}(x, y)$, and $\mathsf{ed}'(x, y) = |x| + |y| - \mathsf{LCS}(x, y)$, where LCS stands for *longest common subsequence*. The algorithm in [BZ16], as well as our modification of it, can be easily adapted to work for this variant of edit distance as well.

2.3 The CGK Random Walk

We review a useful random process called the *CGK random walk*, which was first introduced by Chakraborty, Goldenberg, and Koucký [CGK16], and played a central role in the sketching algorithm of [BZ16].

Definition 2.8 (CGK random walk $\lambda_r(s)$, [CGK16])**.** Given a string $s \in \Sigma^n$, an integer $m \geq 0$, and a sequence of $m \cdot |\Sigma|$ random coins interpreted as a random function $r\colon [m] \times \Sigma \to \{0,1\}$, the m-step CGK random walk is a length-m string $\lambda_r(s) \in \Sigma^m$ defined by the following process:

- Append s with infinitely many zeros.
- Initialize the pointer $p \leftarrow 1$ and the output string $s' \leftarrow \emptyset$.
- For each step $i = 1, \ldots, m$:
  - Append $s[p]$ to s'.
  - Update $p \leftarrow p + r(i, s[p])$.
- Output $s' =: \lambda_r(s)$.

For a contiguous segment of the output string $\lambda_r(s)$, the *pre-image* of this segment refers to the corresponding substring in the original input string s (which may also include the appended trailing zeros if the walk extends beyond s).

Due to its usefulness in the two-party setting with public randomness, we also frequently use the term *CGK random walk* to refer to a *pair* of random walks (as defined in Definition 2.8) performed on two input strings x, y using the *shared* random string r.

Consider a CGK random walk λ on two input strings x, y. We use p_i (resp., q_i) to denote the pointer on string x (resp., y) at the beginning of step i. We refer to the pair (p_i, q_i) as the *state* of λ at the i-th step, and we write $(p, q) \in \lambda$ if λ passes through the state (p, q), i.e., there exists some i for which $p_i = p$ and $q_i = q$. We say the i-th step of λ is a *progress step* if the i-th characters of the output strings $\lambda_r(x)$ and $\lambda_r(y)$ differ, or equivalently, $x[p_i] \neq y[q_i]$.[6] We say λ *walks through* x, y, if in the end the two pointers satisfy $p_m \geq |x|$ and $q_m \geq |y|$.

The following theorem established the connection between CGK random walks and edit distance. Informally, when $\mathsf{ed}(x, y)$ is small, with good probability the number of progress steps in λ is also small (or equivalently, the Hamming distance between the output strings $\lambda_r(x), \lambda_r(y)$ is small).

Theorem 2.9 ([CGK16, Theorem 4.1])**.** *Let λ be an m-step CGK random walk on x, y. Then*

(1) if $m \geq 3 \cdot \max\{|x|, |y|\}$, then λ walks through x, y with probability at least $1 - e^{\Omega(m)}$;

(2) given $\lambda_r(x)$ and r, we can reconstruct the pre-image of $\lambda_r(x)$;

(3) $\mathbf{Pr}\left[\#\text{progress steps in } \lambda \geq (T \cdot \mathsf{ed}(x, y))^2\right] \leq O(1/T)$.

We provide a simpler proof for Item (3) of this theorem in Appendix A.

2.4 Random Walks

We frequently relate the CGK random walk to the following one-dimensional random walk.

Definition 2.10 (One-dimensional unbiased and self-looped random walk)**.** A stochastic process $X = (X_t)_{t \in \mathbb{N}}$ on integers is a one-dimensional unbiased and self-looped random walk if its transition satisfies

$$X_i = \begin{cases} X_{i-1} - 1 & \text{w.p., } 1/4, \\ X_{i-1} & \text{w.p., } 1/2, \\ X_{i-1} + 1 & \text{w.p., } 1/4. \end{cases}$$

[6] Our definition of "progress step" is different from that of [BZ16], which additionally requires at least one of the two pointers moves forward in that step.

Remark 2.11. Let λ be a CGK random walk on two strings and (p, q) be its state. Define $\Delta = p - q$. Then Δ can be viewed as a one-dimensional unbiased and self-looped random walk, which makes a transition when and only when λ makes a progress step.

Fact 2.12 (e.g. [LPW09, Proposition 2.1])**.** *Let a, b be two non-negative integers and X be a one-dimensional unbiased and self-looped random walk. Suppose the walk starts at $X_0 = 0$ and stops when $(X_i = -a) \vee (X_i = b)$. Then*

(1) if $(a, b) \neq (0, 0)$, then $\mathbf{Pr}[X \text{ stops at } b] = a/(a+b)$ and $\mathbf{Pr}[X \text{ stops at } a] = b/(a+b)$;

(2) $\mathbb{E}[\#\text{steps until } X \text{ stops}] = 2 \cdot ab$.

By Remark 2.11 and the martingale property, we have the following lemma, the proof of which is deferred to Appendix D.

Lemma 2.13. *Consider an ∞-step CGK random walk λ on x, y, where p, q are the pointers on x, y respectively. Let u be an index and let $U, V \geq u - 1$ be any integers. Then the following hold.*

(1) Let T_0 be the first time that $p_{T_0} \geq u$. Then $\mathbb{E}[|p_{T_0} - q_{T_0}|] \leq 4 \cdot \mathsf{ed}(x[1..U], y[1..V])$.

(2) Let T_1 be the first time that $(p_{T_1} \geq u) \wedge (q_{T_1} \geq u)$. Then $\mathbb{E}[|p_{T_1} - q_{T_1}|] \leq 4 \cdot \mathsf{ed}(x[1..U], y[1..V])$.

3 Sketches for Edit Distance

For the rest of the paper, we use the following notational conventions:

- n is the length of the input strings; $m := 3n$ is the number of steps in a CGK random walk.
- x, y are the input strings of length n, which is appended with infinitely many zeros; we are promised $\mathsf{ed}(x, y) \leq k$.[7]
- when we use $(\cdot, \cdot)$ to denote a CGK state or an edge between x, y, the first coordinate is a pointer on x and the second is on y.
- $\mathcal{M}$ is the greedy optimal matching of x, y.

Our goal is to prove the following theorem.

Theorem 3.1. *There exists a sketching algorithm for $\mathcal{Q}_{n,k,\delta}$ with sketch size $O\left(k^3 \log^2(n/\delta) \log n\right)$ bits. Moreover, the algorithm has the following properties.*

- *The encoding algorithm used by Alice (resp., Bob) only assumes one-pass streaming access to the input string x (resp., y). The time complexity per character is $\mathrm{poly}(k \log(n/\delta))$, and the space complexity is $O\left(k^3 \log^2(n/\delta) \log n\right)$ bits.* [8]
- *The decoding algorithm used by the referee has time complexity $\mathrm{poly}(k \log(n/\delta))$.*

In Subsection 3.1, we review the general framework of [BZ16]'s sketching protocol, and highlight our key improvement in Lemma 3.6. We will prove this key lemma in Subsection 3.2 and Subsection 3.3. In Subsection 3.4 we present the detailed construction of sketches.

[7] We will also analyze the behaviour of our algorithms when $\mathsf{ed}(x, y) > k$.

[8] The algorithm may use a large number of shared random bits, which can be reduced using Nisan's generator [Nis92]. The main cost, as we can see from the proof, comes from the CGK random walk. Hence we refer readers to [CGK16] for more details on reducing randomness for the CGK random walk.

3.1 General Framework

We adopt the definition of *effective alignments* from [BZ16]. Intuitively, an effective alignment between two strings x, y contains the information of an edit sequence from x to y, but does not contain the information of unchanged characters.

Definition 3.2 (Effective alignment $\mathcal{A}$, [BZ16])**.** For two strings $x, y \in \Sigma^n$, an *effective alignment* $\mathcal{A}$ between x and y is a triplet (G, g_x, g_y), where

- $G = (V_x, V_y, E)$ is a bipartite matching where nodes $V_x = [n], V_y = [n]$ correspond to indices of characters in x and y respectively, and every matched edge $(i, j) \in E$ satisfies $x[i] = y[j]$. Moreover, the matched edges are non-intersecting, i.e., for every pair of distinct edges $(i, j), (i', j') \in E$, we have $i < i'$ iff $j < j'$.

- g_x (resp., g_y) is a partial function defined on the set of unmatched nodes $U_x \subseteq V_x$ (resp., $U_y \subseteq V_y$). For each $i \in U_x$ (resp., $j \in U_y$), define $g_x(i) = x[i]$ (resp., $g_y(j) = y[j]$).

Definition 3.3 (Effective alignments consistent with a CGK random walk, [BZ16])**.** Let λ be a CGK random walk on x, y, where p, q are the pointers on x and y respectively. If λ walks through x, y, then we say an effective alignment $\mathcal{A} = (G, g_x, g_y)$ is *consistent with* λ if for every matched edge $(p, q) \in G$, we have $(p, q) \in \lambda$.

As mentioned in Subsection 1.1, Alice and Bob use public randomness to instantiate $\tau = O(k \log(n/\delta))$ independent CGK random walks $\lambda_1, \ldots, \lambda_\tau$ on x, y. Then, for each CGK random walk λ_i, Alice constructs a sketch sx_i based on her part of the random walk $\lambda_i(x)$, and Bob similarly constructs sy_i based on his part of the random walk $\lambda_i(y)$. The referee receives sx_i, sy_i, and tries to extract an effective alignment $\mathcal{A}_i$ from the sketches. Each sx_i (and sy_i) has length $O(k^2 \log(n/\delta) \log n)$. The properties of this protocol are summarized as follows.

Construction 3.4 (Sketch for each random walk, adapting [BZ16])**.** Let $C \geq 1$ be some large constant and $\eta \in (0, 1)$. There exists an efficient sketching algorithm such that the following holds. Let λ be an m-step CGK random walk on x (and y). Then,

- the sketch size and encoding space are $O\left(k^2 \log(n/\eta) \log n\right)$ bits;
- the encoding time per character and decoding time are both $\mathrm{poly}(k \log(n/\eta))$;
- for fixed λ, x, y the following hold with success probability at least $1 - \eta$:
  - the decoder either (a) reports "error", or (b) outputs an effective alignment $\mathcal{A}$ consistent with λ;
  - when λ walks through x, y and contains at most $C \cdot k^2$ progress steps, (b) occurs.

We present a formal proof of Construction 3.4 in Subsection 3.4.

The final sketches are simply $sx = sx_1 \circ \cdots \circ sx_\tau$ and $sy = sy_1 \circ \cdots \circ sy_\tau$. The referee tries to obtain an effective alignment from every (sx_i, sy_i), and then uses the following lemma to compute $\mathsf{ed}(x, y)$ and recover an optimal edit sequence.

Lemma 3.5 ([BZ16, Lemma 14 and Lemma 19])**.** *There exists a deterministic algorithm taking (sx, sy) as input such that the following holds.*

- *The running time of the algorithm is $\mathrm{poly}(|sx| + |sy|) = \mathrm{poly}(k \log(n/\delta))$.*
- *Let $\mathcal{A}_{i_1}, \ldots, \mathcal{A}_{i_w}$ be the effective alignments*[9] *decoded from $(sx_1, sy_1), \ldots, (sx_\tau, sy_\tau)$.*

  If $w \geq 1$ and each $\mathcal{A}_{i_j}$ is consistent with λ_{i_j}, then the algorithm outputs a valid edit sequence. If, additionally, $\mathcal{M}$ goes through all edges that are common to $\mathcal{A}_{i_1}, \ldots, \mathcal{A}_{i_w}$, then the edit sequence is optimal.

[9] Although we can check if $\mathcal{A}_{i_j}$ is an effective alignment, there is no guarantee that $\mathcal{A}_{i_j}$ is an effective alignment consistent with λ_{i_j}. This subtle difference comes from that in Construction 3.4 we do not give any guarantee outside the $1 - \eta$ success probability.

Now we state our key lemma.

Lemma 3.6 (Key Lemma)**.** *There exist some large constants $C_1, C_2 \geq 1$ such that the following holds. Let λ be an ∞-step CGK random walk on x, y. Then for any fixed $(u,v) \notin \mathcal{M}, x[u] = y[v]$, we have*

$$\mathbf{Pr}\left[(u,v) \notin \lambda \bigwedge \#\textit{progress steps in } \lambda \leq C_1 \cdot k^2\right] \geq \frac{1}{C_2 \cdot k}.$$

Here we reiterate that Lemma 3.6 summarizes our improvement over the previous work of [BZ16] in two aspects (as mentioned in Subsection 1.1): (1) The previous work only gave a lower bound on $\mathbf{Pr}[(u,v) \notin \lambda]$, while we bound the probability of two events happening simultaneously; (2) The previous work only gave a bound of $\Omega(1/k^2)$, while we give an $\Omega(1/k)$ bound. The proof of this Lemma 3.6 is divided into two parts in Subsection 3.2 and Subsection 3.3, in which a technical proposition that leads to the improvement in Item (2) will be proved in Section 4.

Assuming Lemma 3.6, we can prove Theorem 3.1.

Proof of Theorem 3.1. Let C_3 be a large constant.

For the encoding part, we instantiate $\tau = C_2 k \cdot C_3 \log(n/\delta) = O(k \cdot \log(n/\delta))$ independent m-step CGK random walks $\lambda_i, i \in [\tau]$; and construct each sx_i, sy_i using Construction 3.4 with parameter $C = C_1, \eta = \delta/(2\tau)$.

For the decoding part, we run the decoding procedure in Construction 3.4 to obtain $\mathcal{A}_{i_1}, \ldots, \mathcal{A}_{i_w}$ for Lemma 3.5. If $w = 0$ or the edit sequence from Lemma 3.5 has more than k edits, we report "error"; otherwise we output the edit sequence together with the corresponding edit distance.

Bounds on the parameters. By constructing each sx_i (and sy_i) in parallel, the final sketch size and encoding space[10] are

$$\tau \cdot O\left(k^2 \log(n/\eta) \log n\right) = O\left(k^3 \log^2(n/\delta) \log n\right).$$

The encoding time per character is then

$$\tau \cdot \mathrm{poly}\left(k \log(n/\eta)\right) = \mathrm{poly}(k \log(n/\delta)).$$

The decoding time follows immediately from Lemma 3.5.

Analysis of the algorithm when $\mathsf{ed}(x,y) > k$. Since $\eta = \delta/(2\tau)$ in Construction 3.4 and by union bound, the decoder, with probability at least $1 - \delta/2$, for each (sx_i, sy_i) either reports "error", or outputs an effective alignment $\mathcal{A}_i$ consistent with λ_i. Conditioning on this, when we apply Lemma 3.5, either $w = 0$ or it outputs a valid edit sequence which has more than k edits. In either case, we report "error".

Analysis of the algorithm when $\mathsf{ed}(x,y) \leq k$. Let $S = \left\{(u,v) \in [n]^2 \mid (u,v) \notin \mathcal{M}, x[u] = y[v]\right\}$ and define events

- $\mathcal{E}_i$: λ_i walks through x, y.
- $\mathcal{E}_i'$: $(u,v) \notin \lambda_i \bigwedge \#$progress steps in $\lambda_i \leq C_1 \cdot k^2$.

Then

$$\begin{aligned}
&\mathbf{Pr}\left[\forall (u,v) \in S,\ \exists i \in [\tau],\ \mathcal{E}_i \wedge \mathcal{E}_i'\right] \\
&\geq 1 - \sum_{(u,v)\in S} \mathbf{Pr}\left[\mathcal{E}_1 \wedge \mathcal{E}_1'\right]^\tau \\
&\geq 1 - \sum_{(u,v)\in S} \left(1 - \mathbf{Pr}\left[\neg\mathcal{E}_1\right] - \mathbf{Pr}\left[\neg\mathcal{E}_1'\right]\right)^\tau \\
&\geq 1 - n^2 \cdot \left(1 - e^{\Omega(n)} - \frac{1}{C_2 \cdot k}\right)^\tau && \text{(due to Theorem 2.9 and Lemma 3.6)} \\
&\geq 1 - \frac{\delta}{2}. && (1)
\end{aligned}$$

[10]We omit the space for storing auxiliary information (e.g., pointers) in the calculation, since these are minor terms.

Let $\lambda_{i_1}, \ldots, \lambda_{i_w}$ be the random walks walking through x, y and containing at most $C_1 \cdot k^2$ progress steps. Since $\eta = \delta/(2\tau)$ in Construction 3.4 and by union bound, the decoder, with probability at least $1-\delta/2$, for each (sx_i, sy_i) either reports "error", or outputs an effective alignment $\mathcal{A}_i$ consistent with λ_i. Conditioning on this, Construction 3.4 must at least obtain effective alignments $\mathcal{A}_{i_j}, \ldots, \mathcal{A}_{i_w}$ that are consistent with the corresponding random walks. Combined with (1), with probability at least $1-\delta$, for any $(u, v) \in S$ there exists some λ_{i_j} missing it. Then the edit sequence from Lemma 3.5 is optimal. □

3.2 Proof of Lemma 3.6: Case $|u - v| > 100 \cdot k$

Proof of Lemma 3.6: Case $|u-v| > 100 \cdot k$. Assume without loss of generality $u > v$. We stop λ when it meets u. Then by Item (1) in Lemma 2.13, at this time the state (p, q) satisfies $\mathbb{E}[|p-q|] \le 4 \cdot k$. Hence by Markov's inequality,

$$\mathbf{Pr}\left[(u, v) \notin \lambda\right] \ge \mathbf{Pr}\left[p - q \le 100 \cdot k\right] = 1 - \mathbf{Pr}\left[p - q > 100 \cdot k\right] \ge 1 - \frac{4 \cdot k}{100 \cdot k} = 0.96. \tag{2}$$

On the other hand, by setting C_1 large enough we know from Theorem 2.9

$$\mathbf{Pr}\left[\#\text{progress steps in } \lambda \le C_1 \cdot k^2\right] \ge 0.99.$$

Hence, by setting C_2 large enough, we have

$$\mathbf{Pr}\left[(u, v) \notin \lambda \bigwedge \#\text{progress steps in } \lambda \le C_1 \cdot k^2\right] \ge 0.96 + 0.99 - 1 \ge \frac{1}{C_2 \cdot k}.$$

□

3.3 Proof of Lemma 3.6: Case $|u - v| \le 100 \cdot k$

First we need the following definition.

Definition 3.7 (Stable zone $\mathcal{Z}$, [BZ16])**.** The *stable zone* $\mathcal{Z}$ of (u, v) consists of substrings $x[u'..u], y[v'..v]$ of equal length $L = u - u' + 1 = v - v' + 1$, where $L \le \min\{u, v\}$ is the maximum possible length satisfying $x[u'..u] = y[v'..v]$. In particular, $u - v = u' - v'$; and $(u', v') \ne (1, 1)$ as $(u, v) \notin \mathcal{M}$.

Moreover, we say a state (p, q) *enters* $\mathcal{Z}$ if $p \ge u'$ and $q \ge v'$.

We will find the following claim useful. It is proved in [BZ16], and we give a slightly different proof here for completeness.

Claim 3.8 ([BZ16, Claim 21])**.** Consider an ∞-step CGK random walk λ on x, y, where p, q are the pointers on x, y respectively. Let T be the first time that λ enters $\mathcal{Z}$, i.e., $(p_T \ge u') \wedge (q_T \ge v')$. Then

$$\mathbf{Pr}\left[p_T - q_T \ne u - v\right] = \mathbf{Pr}\left[p_T - q_T \ne u' - v'\right] \ge 2/3.$$

Proof. Note that event "$p_T - q_T = u' - v'$" is exactly "$(p_T, q_T) = (u', v')$". Hence one of the following three cases must happen at some step $t < T$ in order to make $(p_T, q_T) = (u', v')$ possible:

- $(p_t, q_t) = (u', v'-1)$. Then the transitions must satisfy $r(i, x[u']) = r(i, y[v'-1]) = 0$ for all $i \in [t..T-2]$ and $r(T-1, x[u']) = 1, r(T-1, y[v'-1]) = 0$, which happens with probability $1/3$ if $x[u'] \ne y[v'-1]$ and with probability 0 if not.
- $(p_t, q_t) = (u'-1, v')$. Similar analysis.
- $(p_t, q_t) = (u'-1, v'-1)$. If the state after time t does not fall into the previous two cases, the transitions must satisfy $r(i, x[u'-1]) = r(i, y[v'-1]) = 0$ for all $i \in [t..T-2]$ and $r(T-1, x[u'-1]) = r(T-1, y[v'-1]) = 1$, which happens with probability $1/3$. □

We will also rely on the following technical result, the proof of which is in Section 4.

Proposition 3.9. *There exists a universal constant $C_4 \ge 1$ such that the following holds. Assume X, Y are two identical length-L strings over alphabet Σ. Assume there exists a size-M matching $(i_1, j_1), \ldots, (i_M, j_M) \in [L]^2$ such that*

- *$i_t > j_t$ and $X[i_t] = Y[j_t]$ hold for all $t \in [M]$;*
- *$i_1 < i_2 < \cdots < i_M$ and $j_1 < j_2 < \cdots < j_M$.*

Let $\rho = C_4 \cdot (L - M)$ and $(\hat{I}, \hat{J})$ be any state satisfying $\hat{I} - \hat{J} \geq \rho$. Then a CGK random walk on X, Y starting from $(\hat{I}, \hat{J})$ will miss (L, L) with probability at least 0.5.

By symmetry, we derive the following corollary.

Corollary 3.10. *Let $C_4 \geq 1$ be the same constant in Proposition 3.9. Assume X, Y are two identical length-L strings over alphabet Σ. Assume there exists a size-d matching $(i_1, j_1), \ldots, (i_M, j_M) \in [L]^2$ such that*

- *$i_t > j_t$ holds for all $t \in [M]$, or $i_t < j_t$ holds for all $t \in [M]$;*
- *$X[i_t] = Y[j_t]$ holds for all $t \in [M]$;*
- *$i_1 < i_2 < \cdots < i_M$ and $j_1 < j_2 < \cdots < j_M$.*

Let $\rho = C_4 \cdot (L - M)$ and $(\hat{I}, \hat{J})$ be any state satisfying $|\hat{I} - \hat{J}| \geq \rho$. Then a CGK random walk on X, Y starting from $(\hat{I}, \hat{J})$ will miss (L, L) with probability at least 0.5.

Proof of Lemma 3.6: Case $|u - v| \leq 100 \cdot k$. Let $C_5 \geq 1$ be a large constant. We will apply Proposition 3.9 with parameter $M \geq L - 103 \cdot k$; and let $\rho = C_4 \cdot 103k$ be the corresponding bound in it.

We expect λ to have the following three phases:

- $\mathcal{E}_1$: λ enters $\mathcal{Z}$ in a state (p_1, q_1) within $C_5 \cdot k^2$ progress steps, where $0 < |(p_1 - q_1) - (u - v)| \leq 200 \cdot k$.
- $\mathcal{E}_2$: Starting from (p_1, q_1) and within $2 \cdot \rho^2$ progress steps, λ reaches a state (p_2, q_2) where either $(p_2, q_2) > (n, n)$ or $|(p_2 - q_2) - (u - v)| \geq \rho$. Also, during the walk from (p_1, q_1) to (p_2, q_2), λ never reaches some state (p, q) satisfying $(p - q) - (u - v) = 0$.
- $\mathcal{E}_3$: $(u, v) \notin \lambda$ and #progress steps in $\lambda \leq 2 \cdot \rho^2 + C_5 \cdot \left(k^2 + (\rho + 301 \cdot k)^2\right)$.

Claim 3.11. $\mathbf{Pr}[\mathcal{E}_1] \geq 0.5$.

Claim 3.12. $\mathbf{Pr}[\mathcal{E}_2 \mid \mathcal{E}_1] \geq 1/(2 \cdot \rho)$.

Claim 3.13. $\mathbf{Pr}[\mathcal{E}_3 \mid \mathcal{E}_1 \wedge \mathcal{E}_2] \geq 1/4$.

Assuming Claim 3.11, Claim 3.12, and Claim 3.13, we show the following desired bound

$$\mathbf{Pr}\left[(u, v) \notin \lambda \bigwedge \#\text{progress steps in } \lambda \leq C_1 \cdot k^2\right] \geq \mathbf{Pr}[\mathcal{E}_3] \geq \mathbf{Pr}[\mathcal{E}_1 \wedge \mathcal{E}_2 \wedge \mathcal{E}_3] \geq \frac{1}{C_2 \cdot k}$$

by setting $C_1 = 2 \cdot (103 \cdot C_4)^2 + C_5 \cdot \left(1 + (301 + 103 \cdot C_4)^2\right)$ and $C_2 = 16$. $\square$

Proof of Claim 3.11. By Theorem 2.9 with C_5 large enough, we ensure

$$\mathbf{Pr}\left[\#\text{progress steps in } \lambda \leq C_5 \cdot k^2\right] \geq 0.99.$$

Then combining Claim 3.8 and (2), we have

$$\begin{aligned}
&\mathbf{Pr}[\mathcal{E}_1] \\
&\geq \mathbf{Pr}\left[\#\text{progress steps in } \lambda \text{ before } \mathcal{Z} \leq C_5 \cdot k^2\right] \\
&\quad + \mathbf{Pr}[(p_1, q_1) \neq (u', v')] + \mathbf{Pr}[|(p_1 - q_1) - (u - v)| \leq 200 \cdot k] - 2 \\
&\geq \mathbf{Pr}\left[\#\text{progress steps in } \lambda \leq C_5 \cdot k^2\right] + \mathbf{Pr}[(p_1, q_1) \neq (u', v')] + \mathbf{Pr}[|p_1 - q_1| \leq 100 \cdot k] - 2 \\
&\geq 0.99 + 2/3 + 0.96 - 2 \\
&> 0.5.
\end{aligned}$$

$\square$

Proof of Claim 3.12. Let $a = |(p_1 - q_1) - (u - v)|$. Then by conditioning on $\mathcal{E}_1$, we know $a \geq 1$. If $a \geq \rho$, then $(p_2, q_2) = (p_1, q_1)$ and the claims holds immediately. Therefore we assume $a < \rho$.

For convenience, if the case $(p_2, q_2) > (n, n)$ happens, we replace $x[p_2+1..\infty]$ and $y[q_2+1..\infty]$ with random strings[11] and continue the walk. Denote this new walk as λ' and we stop it when $(p - q) - (u - v) = 0$ or $|(p - q) - (u - v)| \geq \rho$ happens. Let π be a one-dimensional unbiased and self-looped random walk starting from a. Due to the correspondence between transitions in π and progress steps in λ', we can stop π when it reaches 0 or ρ; and hence

$$\begin{aligned}
\mathbf{Pr}\left[\mathcal{E}_2 \mid \mathcal{E}_1\right] &\geq \mathbf{Pr}\left[\lambda' \text{ stops at the second case within } 2 \cdot \rho^2 \text{ progress steps}\right] \\
&= \mathbf{Pr}\left[\pi \text{ stops at } \rho \text{ in } 2 \cdot \rho^2 \text{ steps}\right] \\
&= \mathbf{Pr}\left[\pi \text{ stops at } \rho\right] \cdot \mathbf{Pr}\left[\pi \text{ stops } \rho \text{ in } 2 \cdot \rho^2 \text{ steps} | \pi \text{ hits } \rho \text{ first}\right] \\
&= \mathbf{Pr}\left[\pi \text{ stops at } \rho\right] \cdot \left(1 - \mathbf{Pr}\left[\pi \text{ stops } \rho \text{ after } 2 \cdot \rho^2 \text{ steps} | \pi \text{ hits } \rho \text{ first}\right]\right) \\
&\geq \mathbf{Pr}\left[\pi \text{ stops at } \rho\right] \cdot \left(1 - \frac{\mathbb{E}\left[\#\text{steps until } \pi \text{ stops} | \pi \text{ hits } \rho \text{ first}\right]}{2 \cdot \rho^2}\right). \\
& \hspace{6cm} \text{(due to Markov's inequality)}
\end{aligned}$$

By Fact 2.12, we have

$$\begin{aligned}
a \cdot (\rho - a) &= \mathbb{E}\left[\#\text{steps until } \pi \text{ stops}\right] \\
&\geq \mathbf{Pr}\left[\pi \text{ stops at } \rho\right] \mathbb{E}\left[\#\text{steps until } \pi \text{ stops} | \pi \text{ stops at } \rho\right] \\
&= \frac{a}{\rho} \cdot \mathbb{E}\left[\#\text{steps until } \pi \text{ stops} | \pi \text{ stops at } \rho\right].
\end{aligned}$$

Hence

$$\mathbf{Pr}\left[\mathcal{E}_2 \mid \mathcal{E}_1\right] \geq \frac{a}{\rho} \cdot \left(1 - \frac{\rho \cdot (\rho - a)}{2 \cdot \rho^2}\right) \geq \frac{1}{2 \cdot \rho}. \qquad \square$$

Proof of Claim 3.13. Observe that

$$\mathbf{Pr}\left[\mathcal{E}_3 \mid \mathcal{E}_1 \wedge \mathcal{E}_2\right] \geq \mathbf{Pr}\left[\text{after } (p_2, q_2),\ (u, v) \notin \lambda \bigwedge \#\text{progress steps} \leq C_5 \cdot (\rho + 301 \cdot k)^2\right].$$

If $(p_2, q_2) > (n, n)$ then there is nothing to prove. So we focus on the case $|(p_2 - q_2) - (u - v)| \geq \rho$.

Let $X = Y = x[u'..u] = y[v'..v]$, $L = u - u' + 1 = v - v' + 1$, and $(\hat{I}, \hat{J}) = (p_2 - u' + 1, q_2 - v' + 1)$. Then $|\hat{I} - \hat{J}| \geq \rho$. Recall that $\mathcal{M}$ is the greedy optimal matching between x, y. We retain those edges $(p, q) \in \mathcal{M}$ that is inside $\mathcal{Z}$ to form a matching $\mathcal{M}'$ between X, Y, i.e.,

$$\mathcal{M}' = \left\{(p - u' + 1, q - v' + 1) \mid (p, q) \in \mathcal{M}, u' \leq p \leq u, v' \leq q \leq v\right\}.$$

By Lemma 2.5, we have

$$M := |\mathcal{M}'| \geq L - 3 \cdot \mathsf{ed}(x, y) - |u - v| \geq L - 103 \cdot k.$$

By Lemma 2.7 and $(u, v) \notin \mathcal{M}$, the condition in Corollary 3.10 holds and we get

$$\mathbf{Pr}\left[\text{after } (p_2, q_2),\ (u, v) \notin \lambda\right] \geq 0.5.$$

Therefore, it suffices to prove

$$\mathbf{Pr}\left[\text{after } (p_2, q_2),\ \#\text{progress steps} \leq C_5 \cdot (\rho + 301 \cdot k)^2\right] \geq \frac{3}{4}. \tag{3}$$

Note that

$$\mathsf{ed}(x[p_2..\infty], y[q_2..\infty]) \leq \mathsf{ed}(x, y) + |p_2 - q_2| \leq \mathsf{ed}(x, y) + |(p_2 - q_2) - (u - v)| + |u - v|.$$

Thus (3) follows from Theorem 2.9 with C_5 large enough and the following estimate:

[11] The random strings here are not essential. The only purpose of this is to generate infinitely many progress steps.

- If $|(p_1 - q_1) - (u - v)| \leq \rho$, then $|(p_2 - q_2) - (u - v)| = \rho$. Hence

$$\mathsf{ed}(x[p_2..\infty], y[q_2..\infty]) \leq k + \rho + 100 \cdot k \leq \rho + 301 \cdot k.$$

- If $\rho < |(p_1 - q_1) - (u - v)| \leq 200 \cdot k$, then $(p_2, q_2) = (p_1, q_1)$ and $|(p_2 - q_2) - (u - v)| \leq 200 \cdot k$. Hence

$$\mathsf{ed}(x[p_2..\infty], y[q_2..\infty]) \leq k + 200 \cdot k + 100 \cdot k \leq \rho + 301 \cdot k. \qquad \square$$

3.4 Sketch Construction

In this section, we provide a detailed description of Construction 3.4 and prove its correctness. The construction is mostly based on [BZ16, Section 4.1], with a few simplifying modifications.

Proof of Construction 3.4. Let $x', y' \in \Sigma^m$ be the outputs of the random walks on x and y respectively, i.e., $x' = \lambda_r(x)$ and $y' = \lambda_r(y)$, where r is the (public) randomness.

The encoding algorithm. We only describe the encoding algorithm for x. The algorithm for y is analogous.

Build a full binary tree $\mathcal{T}_x$ of depth $d = \log m$ on top of x', where the segments in depth i have length $m/2^i$. We use $\mathcal{T}_x(i,j)$ to denote the j-th segment in depth i. More precisely, for $i \in [0..d], j \in [2^i]$,

$$\mathcal{T}_x(i,j) = x'\left[1 + \frac{(j-1)\cdot m}{2^i} .. \frac{j \cdot m}{2^i}\right].$$

Let $f : \Sigma^{\leq m} \to \Gamma$ be a randomized hash function[12] in Theorem 3.14 by setting $\eta_1 = \eta/(2(d+1)\cdot m)$, and define $h^x_{i,j} = f(\mathcal{T}_x(i,j))$.

Theorem 3.14 (Rolling hash, [KR87])**.** *Let $\eta_1 \in (0,1)$. There exists a randomized hash function $f\colon \Sigma^{\leq m} \to \Gamma$ where*

(a) $|\Gamma| = O(|\Sigma| + (m/\eta_1))$ *and* $\Sigma \subseteq \Gamma$;

(b) for any distinct $z, z' \in \Sigma^{\leq m}$, $\mathbf{Pr}[f(z) = f(z')] \leq \eta_1$; *and for any single character* $c \in \Sigma$, $f(c) = c$;[13]

(c) f *can be computed with* $O(\log|\Gamma|)$ *bits of space and* $\mathrm{poly}(\log|\Gamma|)$ *time per character.*

In addition, we define the following quantities for each segment $\mathcal{T}_x(i,j)$ which will be used later to identify an effective alignment:

- let $\ell^x_{i,j} \in [m]$ be the length of the pre-image of $\mathcal{T}_x(i,j)$ in $x[1..\infty]$;
- let $\alpha^x_{i,j} \in \{0,1\}$ be the indicator of whether the pre-image of the first character of $\mathcal{T}_x(i,j)$ and the last character of $\mathcal{T}_x(i,j-1)$[14] are identical;
- let $\beta^x_{i,j} \in \{0,1\}$ be the indicator of whether the pre-image of the last character of $\mathcal{T}_x(i,j)$ and the first character of $\mathcal{T}_x(i,j+1)$[15] are identical.

That is, assume the pointers are p_1, p_2, p_3, p_4 for $\mathcal{T}_x(i,j-1)$'s last character, $\mathcal{T}_x(i,j)$'s first character, $\mathcal{T}_x(i,j)$'s last character, and $\mathcal{T}_x(i,j+1)$'s first character respectively. Then

- $\ell^x_{i,j} = p_3 - p_2 + 1$.
- If $p_1 = p_2$ then $\alpha^x_{i,j} = 1$; otherwise $\alpha^x_{i,j} = 0$.

[12] The same f is used for both x and y in this random walk λ, but it may be different among different walks.

[13] When we interpret Σ as a set of numbers, the rolling hash is $f(x) = \sum_{i=1}^{|x|} x_i \cdot r^{i-1} \mod p$, where r is a random number and p is a large prime. Therefore $f(\cdot)$ is an identity map on a single character.

[14] If $j = 1$, $\mathcal{T}_x(i,j-1)$ is not well-defined. Then we simply define $\alpha^x_{i,j} = 0$ in this case.

[15] If $j = 2^i$, $\mathcal{T}_x(i,j+1)$ is not well-defined. Then we simply define $\beta^x_{i,j} = 0$ in this case.

- If $p_3 = p_4$ then $\beta^x_{i,j} = 1$; otherwise $\beta^x_{i,j} = 0$.

Let U_x be the set of 6-tuples $(i, j, h^x_{i,j}, \ell^x_{i,j}, \alpha^x_{i,j}, \beta^x_{i,j}) \in [0..d] \times [m] \times \Gamma \times [m] \times \{0,1\} \times \{0,1\}$. We write u_x as the indicator vector for U_x, which is a binary vector of length $4(d+1) \cdot m^2 \cdot |\Gamma|$. We now apply Theorem 3.15 with $L = 4(d+1) \cdot m^2 \cdot |\Gamma|, \Delta = 6(d+1) \cdot Ck^2, \eta_2 = \eta/2$ to obtain the sketch $\mathcal{S}(u_x)$.

Theorem 3.15 ([KNP+17, Section A.3]). *Let L, Δ be two positive integers and $\eta_2 \in (0,1)$. Let $z, z' \in \{0,1\}^L$. There exists an efficient randomized linear sketching algorithm $\mathcal{S}$ where*

- *the sketch size and the encoding space are $O(\Delta \log L + \log(L/\eta_2))$;*
- *the encoding time per bit and the decoding time are $\mathrm{poly}(\Delta \log(L/\eta_2))$;*
- *if $z - z'$ has at most Δ non-zero coordinates, the decoder recovers $z - z'$ exactly;*
- *if $z - z'$ has more than Δ non-zero coordinates, the decoder reports "fail" with probability at least $1 - \eta_2$.*

The final sketch is $(\ell^x_{0,1}, \mathcal{S}(u_x))$. Since $\ell^x_{0,1} \in [m]$ is the length of the pre-image of the whole λ, it in particular indicates if λ walks through x.

The decoding algorithm. Let $\mathcal{S}_x = (\ell^x_{0,1}, \mathcal{S}(u_x))$ and $\mathcal{S}_y = (\ell^y_{0,1}, \mathcal{S}(u_y))$ be the sketches for x, y in this round of random walk respectively. We first condition on the event $\mathcal{E}$ that the following two bullets simultaneously occur:

- For all i, j, if $\mathcal{T}_x(i,j) \neq \mathcal{T}_y(i,j)$ then $h^x_{i,j} \neq h^y_{i,j}$.
- If u_x and u_y differ in more than Δ coordinates, the decoder from Theorem 3.15 reports "fail".

By a union bound, this conditioning costs at most η probability loss.

If $\ell^x_{0,1} < n$ or $\ell^y_{0,1} < n$ or the decoder from Theorem 3.15 reports "fail", we directly report "error". Let $\mathcal{E}'$ be the event that we haven't reported "error". Then it suffices to verify the following two claims.

Claim 3.16. Conditioning on $\mathcal{E}$ and λ containing at most $C \cdot k^2$ progress steps, u_x and u_y differ in at most Δ coordinates.

Proof. Let

$$V_1 = \{(i,j) \mid \mathcal{T}_x(i,j) \neq \mathcal{T}_y(i,j)\}, \qquad V_2 = \left\{(i,j) \mid \ell^x_{i,j} \neq \ell^y_{i,j}\right\},$$
$$V_3 = \left\{(i,j) \mid \alpha^x_{i,j} \neq \alpha^y_{i,j}\right\}, \qquad V_4 = \left\{(i,j) \mid \beta^x_{i,j} \neq \beta^y_{i,j}\right\}.$$

Conditioning on $\mathcal{E}$, the number of coordinates where u_x, u_y differ equals twice the size of $V_1 \cup V_2 \cup V_3 \cup V_4$.

- **For V_1.** Observe that $(i,j) \in V_1$ iff there is some progress step among $1 + (j-1) \cdot m/2^i$ and $j \cdot m/2^i$. Hence by the structure of the full binary tree, we have $|V_1| \le (d+1) \cdot \#\text{progress steps in } \lambda \le (d+1) \cdot Ck^2$.
- **For V_2.** By Item (2) in Theorem 2.9, if $\mathcal{T}_x(i,j) = \mathcal{T}_y(i,j)$ then their pre-image equals, which implies $\ell^x_{i,j} = \ell^y_{i,j}$. Thus $V_2 \subseteq V_1$.
- **For V_3.** By Item (2) in Theorem 2.9, if $\mathcal{T}_x(i,j-1) \circ \mathcal{T}_x(i,j) = \mathcal{T}_y(i,j-1) \circ \mathcal{T}_y(i,j)$ then their pre-image equals, which implies the random walk on this part is identical for x, y and hence $\alpha^x_{i,j} = \alpha^y_{i,j}$. Thus $V_3 \subseteq V_1 \cup \{(i,j) \mid (i,j-1) \in V_1\}$.
- **For V_4.** Similar analysis as for V_3. We have $V_4 \subseteq V_1 \cup \{(i,j) \mid (i,j+1) \in V_1\}$.

In all, $|V_1 \cup V_2 \cup V_3 \cup V_4| = |V_1 \cup V_3 \cup V_4| \le |V_1| + |V_3 \setminus V_1| + |V_4 \setminus V_1| \le 3 \cdot |V_1| \le 3(d+1) \cdot Ck^2 = \Delta/2$. ☐

Claim 3.17. Conditioning on $\mathcal{E}, \mathcal{E}'$, we can compute in $\mathrm{poly}(k \log(n/\eta))$ time an effective alignment $\mathcal{A}$ consistent with λ.

Algorithm 1: Construct effective alignment $\mathcal{A}$

Input: $\ell^x_{0,1}, \ell^y_{0,1}$ and $(h^x_{i,j}, \ell^x_{i,j}, \alpha^x_{i,j}, \beta^x_{i,j}), (h^y_{i,j}, \ell^y_{i,j}, \alpha^y_{i,j}, \beta^y_{i,j})$ for marked nodes (i, j)
Output: An effective alignment $\mathcal{A}$ consistent with λ

Initialize $\mathcal{A} = (G, g_x, g_y)$ with an empty graph G and empty functions g_x, g_y
`DFS`$(1, 0, 1, \ell^x_{0,1}, 1, \ell^y_{0,1})$
Procedure `DFS`$(i, j, s_x, e_x, s_y, e_y)$
    /* (i, j) is the current node on the binary tree. s_x (resp., s_y) is the pointer for $\mathcal{T}_x(i, j)$'s (resp., $\mathcal{T}_y(i, j)$'s) first character. e_x (resp., e_y) is the pointer for $\mathcal{T}_x(i, j)$'s (resp., $\mathcal{T}_y(i, j)$'s) last character. */
    if (i, j) *is not marked* **then**
        foreach *edge* $(p_x, p_y) \in \{(s_x, s_y), (s_x + 1, s_y + 1), \ldots, (e_x, e_y)\}$ **do**
            if $p_x \le n$ *and* $p_y \le n$ **then** Add edge (p_x, p_y) to G
            if $p_x \le n$ *and* $p_y > n$ **then** Update $g_x(p_x) = 0$
            if $p_x > n$ *and* $p_y \le n$ **then** Update $g_y(p_y) = 0$
        end
    else
        if (i, j) *is a leaf* **then**
            Update $g_x(s_x) = h^x_{i,j}$ and $g_y(s_y) = h^y_{i,j}$ // due to Item (b) in Theorem 3.14
        else
            $(i_1, j_1) \leftarrow$ left child of (i, j) and $(i_2, j_2) \leftarrow$ right child of (i, j)
            if (i_1, j_1) *is marked* **then**
                $m_x \leftarrow s_x + \ell^x_{i_1,j_1} - 1$ and $m_y \leftarrow s_y + \ell^y_{i_1,j_1} - 1$
                $o_x \leftarrow 1 - \beta^x_{i_1,j_1}$ and $o_y \leftarrow 1 - \beta^y_{i_1,j_1}$
                `DFS`$(i_1, j_1, s_x, m_x, s_y, m_x)$
                `DFS`$(i_2, j_2, m_x + o_x, e_x, m_y + o_y, e_y)$
            else   // (i_2, j_2) must be marked
                $m_x \leftarrow e_x - \ell^x_{i_2,j_2} + 1$ and $m_y \leftarrow e_y - \ell^y_{i_2,j_2} + 1$
                $o_x \leftarrow 1 - \alpha^x_{i_1,j_1}$ and $o_y \leftarrow 1 - \alpha^y_{i_1,j_1}$
                `DFS`$(i_1, j_1, s_x, m_x - o_x, s_y, m_x - o_x)$
                `DFS`$(i_2, j_2, m_x, e_x, m_y, e_y)$
            end
        end
    end

Proof. Conditioning on $\mathcal{E}, \mathcal{E}'$, we use the decoder of Theorem 3.15 to mark those (i, j) that

$$(h^x_{i,j}, \ell^x_{i,j}, \alpha^x_{i,j}, \beta^x_{i,j}) \neq (h^y_{i,j}, \ell^y_{i,j}, \alpha^y_{i,j}, \beta^y_{i,j}).$$

We construct $\mathcal{A}$ by performing a DFS on the tree as in Algorithm 1. Its correctness is guaranteed by $\ell^x_{0,1}, \ell^y_{0,1} \ge n$.

Observe that the number of recursions is at most the number of marked nodes, which is at most Δ. Hence the running time of Algorithm 1 is $\mathrm{poly}(k \log(n/\eta))$. □

Bounds on the parameters. The decoding time is obvious, so we only calculate the parameters for encoding. By Theorem 3.15, the sketch size is

$$O(\log m) + O(\Delta \log L + \log(L/\eta_2)) = O\left(k^2 \log(n/\eta) \log n\right).$$

When doing encoding, we work on each depth of $\mathcal{T}$ in parallel. For a fixed depth, the hashes can be computed sequentially. Hence by Theorem 3.14 and Theorem 3.15, the encoding space[16] is

$$(d + 1) \cdot O(\log |\Gamma|) + O(\Delta \log L + \log(L/\eta_2)) = O\left(k^2 \log(n/\eta) \log n\right).$$

[16]We omit the space for storing auxiliary information (e.g., current nodes) in the calculation, since these are minor term.

Note that when a character arrives, we generate at most $(d+1)$ 6-tuples, hence the encoding time per character is bounded by

$$(d+1) \cdot (\text{poly}(\log |\Gamma|) + \text{poly}(\Delta \log(L/\eta_2))) = \text{poly}(k \log(n/\eta)). \qquad \square$$

4 CGK Random Walks on Self-similar Strings

This section is devoted to the proof of Proposition 3.9. It characterizes CGK random walks on strings of certain self-similarity, which may be interesting on its own.

Proposition (Proposition 3.9 restated)**.** *There exists a universal constant $C_4 \geq 1$ such that the following holds. Assume X, Y are two identical length-L strings over alphabet Σ. Assume there exists a size-M matching $(i_1, j_1), \ldots, (i_M, j_M) \in [L]^2$ such that*

- *$i_t > j_t$ and $X[i_t] = Y[j_t]$ hold for all $t \in [M]$;*
- *$i_1 < i_2 < \cdots < i_M$ and $j_1 < j_2 < \cdots < j_M$.*

Let $\rho = C_4 \cdot (L - M)$ and $(\hat{I}, \hat{J})$ be any state satisfying $\hat{I} - \hat{J} \geq \rho$. Then a CGK random walk on X, Y starting from $(\hat{I}, \hat{J})$ will miss (L, L) with probability at least 0.5.

We will first provide necessary definitions and explore basic properties in Subsection 4.1. Then we relate them with edit distances in Subsection 4.2, and present the main proof in Subsection 4.3 and Subsection 4.4. The proof of a technical lemma is deferred to Subsection 4.5.

4.1 Stable States

We fix the matching in Proposition 3.9, so when we say (i, j) is a matched edge it means (i, j) is an edge in the matching. We extend X, Y to $X[-\infty..\infty], Y[-\infty..\infty]$ by adding dummy characters $X[i] = Y[i] = X[L]$ for all $i > L$, and $X[i] = Y[i] = X[1]$ for all $i < 1$. We also add matched edges $(i, i-1)$ for all $i > L$ as well as $i \leq 1$. Note that all the edges are still non-intersecting. Though the added characters may not be consistent with the original input strings x, y, it does not change the probability of the walk missing (L, L). Since $X = Y$ and the initial state satisfies $\hat{I} \geq \hat{J} + \rho \geq \hat{J}$, any future state (I, J) must still satisfy $I \geq J$.

We introduce the notion of *stable segment*.

Definition 4.1 (Stable segment)**.** We say $[l..r]$ is a *stable segment*, if for every matched edge (I, J) (where we must have $I > J$), exactly one of the following two conditions hold:

- $J < l$ and $I \leq r$.
- $J \geq l$ and $I > r$.

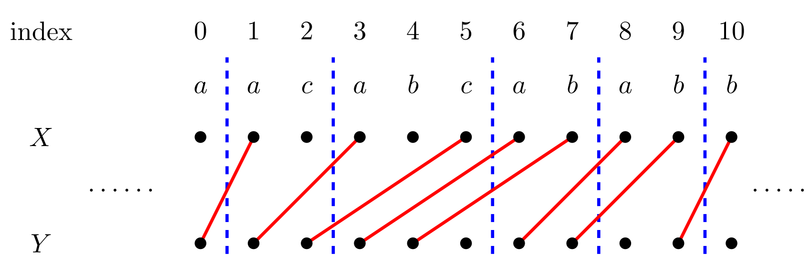

Figure 1: A stable partition for $X[1..L] = Y[1..L] = acabcabab$ $(L = 9)$.

For example in Figure 1, every segment separated by blue dashed lines is a stable segment.

Remark 4.2. To gain a better intuition of the definition, consider the special case where the string $X[1..L]$ has period p and every matched edge (I, J) inside segment $[1..L]$ satisfies $I - J = p$. In this periodic case, a segment contained in $[2..L-1]$ is stable if and only if its length is p.

Our motivation is that, when there are few unmatched characters, using our more generalized definition we can approximately preserve the nice properties of periodic strings. For example, when X has period p, the strings $X[i..i+tp-1]$ and $X[i+tp..i+2tp-1]$ must be identical. In a non-periodic case, we can similarly prove that $X[i..j-1]$ and $X[j..j+(j-i)-1]$ have small edit distance if $[i..j-1]$ can be divided into several stable segments (see Lemma 4.15 for a more formal statement). In the remaining part of the section, readers are encouraged to use the periodic case for a more intuitive understanding.

The following lemma says that every character can be the beginning/ending of some stable segment.

Lemma 4.3. *The following hold.*

(1) For every $r \in \mathbb{Z}$, there exists some $l \le r$ such that $[l..r]$ is a stable segment.

(2) For every $l \in \mathbb{Z}$, there exists some $r \ge l$ such that $[l..r]$ is a stable segment.

Proof. We prove Item (1), and the proof of Item (2) is analogous. Let

$$l' = \min\{l' : \text{there exists } r' \ge r+1 \text{ such that } (r', l') \text{ is a matched edge}\}.$$

We claim $[\min\{r, l'\}..r]$ must be a stable segment. If not, then there are two possible cases:

- There exists a matched edge (I, J) such that $J < l'$ and $I > r$. This contradicts the minimality of l'.
- There exists a matched edge (I, J) such that $l' \le J < I \le r$. By the definition of l', there is another matched edge (r', l') where $r' \ge r+1 > I$. Since these two edges are non-intersecting, we must have $l' > J$. A contradiction. □

Definition 4.4 (Stable partition $\mathcal{P}$ and stable states)**.** Consider a partition $\mathcal{P} = (\mathcal{P}_i)_i$ of the integers into segments, where $\mathcal{P}_i = [p_i..p_{i+1}-1]$ and $p_i < p_{i+1}$. We say $\mathcal{P}$ is a *stable partition* if every $\mathcal{P}_i$ is a stable segment. Then we say

- state (I, J) is a *$(\mathcal{P}, b)$-stable state*, if there exists some i such that $J = p_i$ and $I = p_{i+b}$;
- state (I, J) is a *b-stable state*, if there exists a stable partition $\mathcal{P}$ such that (I, J) is a $(\mathcal{P}, b)$-stable state;
- state (I, J) is a *stable state*, if there exists some $b \ge 0$ such that (I, J) is a b-stable state. In particular, when $I \ge J > L$, (I, J) is always a stable state.

Note that, given a partition of $[J..I-1]$ into several stable segments, we can apply Lemma 4.3 and extend it into a stable partition. Hence we have the following fact.

Fact 4.5. *For any $b \ge 1$, a state (I, J) is a b-stable state iff $[J..I-1]$ can be partitioned into b stable segments. Moreover, a state (I, J) is a 0-stable state iff $I = J$.*

From the definition of stable partition, we immediately have the following result.

Proposition 4.6. *Let $(\mathcal{P}_i)_i$ be a stable partition. Then for every matched edge (I, J), there exists some i such that $J \in \mathcal{P}_{i-1}$ and $I \in \mathcal{P}_i$.*

Proof. Let i, j be such that $I \in \mathcal{P}_i$ and $J \in \mathcal{P}_j$. Since $I > J$, we must have $i \ge j$. If $i > j+1$, then $\mathcal{P}_{j+1}$ cannot be a stable segment. If $i = j$, then $\mathcal{P}_j$ cannot be a stable segment. Hence we must have $i = j+1$. □

Lemma 4.7 (Stable predecessors $\mathsf{pred}_L(\cdot), \mathsf{pred}_R(\cdot)$)**.** *For every $I \in \mathbb{Z}$, there exists $\mathsf{pred}_L(I) \le \mathsf{pred}_R(I) \le I-1$ such that $[J..I-1]$ is a stable segment iff $\mathsf{pred}_L(I) \le J \le \mathsf{pred}_R(I)$.*

Moreover, the following hold:

(1) $\mathsf{pred}_L(I) \le \mathsf{pred}_L(I+1)$;

(2) $\mathsf{pred}_R(I) \le \mathsf{pred}_R(I+1)$.

Proof. To prove the first part, suppose $J_0 \le J \le J_1$, and both $[J_0..I-1]$ and $[J_1..I-1]$ are stable segments. If $[J..I-1]$ is not a stable segment, then there are two possibilities:

- There is a matched edge (I', J') such that $J' < J$ and $I - 1 < I'$. Then $J' < J_1 \le I - 1 < I'$, contradicting that $[J_1..I-1]$ is a stable segment.
- There is a matched edge (I', J') such that $J \le J' < I' \le I - 1$. Then $J_0 < J' < I' \le I_1$, contradicting that $[J_0..I-1]$ is a stable segment.

Hence, $[J..I-1]$ is also a stable segment, which means $\mathsf{pred}_L(I), \mathsf{pred}_R(I)$ are well-defined.

We prove Item (1). Let $J_0 = \mathsf{pred}_L(I)$, then there must be a matched edge $(I', J_0 - 1)$ where $J_0 \le I' \le I - 1$. If $\mathsf{pred}_L(I+1) < J_0$, then $[\mathsf{pred}_L(I+1)..I]$ is a stable segment, which contains matched edge $(I', J_0 - 1)$. A contradiction.

Now we turn to Item (2). Let $J_1 = \mathsf{pred}_R(I+1)$. Suppose that $\mathsf{pred}_R(I) > J_1$. Then $[J_1+1..I-1]$ is also a stable segment due to Item (1). By the definition of J_1, there must be a matched edge (I', J_1) where $I' \ge I + 1$. This contradicts that $[J_1+1..I-1]$ is a stable segment. $\square$

For example in Figure 1, we have $\mathsf{pred}_L(8) = 5, \mathsf{pred}_R(8) = 6$.

Lemma 4.8 (b-stable predecessors $\mathsf{pred}_L^{(b)}(\cdot), \mathsf{pred}_R^{(b)}(\cdot)$)**.** *Define*

$$\mathsf{pred}_L^{(b)}(I) = \begin{cases} I & b = 0, \\ \mathsf{pred}_L(\mathsf{pred}_L^{(b-1)}(I)) & b \ge 1, \end{cases} \quad \textit{and} \quad \mathsf{pred}_R^{(b)}(I) = \begin{cases} I & b = 0, \\ \mathsf{pred}_R(\mathsf{pred}_R^{(b-1)}(I)) & b \ge 1. \end{cases}$$

Then the following hold for all I, b:

(1) $\mathsf{pred}_L^{(b)}(I) \le \mathsf{pred}_L^{(b)}(I+1)$*, and* $\mathsf{pred}_R^{(b)}(I) \le \mathsf{pred}_R^{(b)}(I+1)$*, and* $\mathsf{pred}_L^{(b)}(I) \le \mathsf{pred}_R^{(b)}(I)$*.*

(2) For all $J \le I - 1$, (I, J) is a b-stable state iff $\mathsf{pred}_L^{(b)}(I) \le J \le \mathsf{pred}_R^{(b)}(I)$*.*

(3) $\mathsf{pred}_R^{(b)}(I) \ge \mathsf{pred}_L^{(b)}(I+1) - 1$*.*

Proof. Item (1) follows immediately from Lemma 4.7.

We now prove Item (2) and (3) together by induction on b. Suppose both statements hold for $b-1$, where $b \ge 1$.

We first prove the "only if" part of Item (2). We partition $[J..I-1]$ into b stable segments, the last of which is $[I_1..I-1]$. Then we have $\mathsf{pred}_L^{(b-1)}(I_1) \le J \le \mathsf{pred}_R^{(b-1)}(I_1)$ by induction hypothesis on Item (2), and $\mathsf{pred}_L(I) \le I_1 \le \mathsf{pred}_R(I)$. Then we have $J \ge \mathsf{pred}_L^{(b-1)}(I_1) \ge \mathsf{pred}_L^{(b-1)}(\mathsf{pred}_L(I)) = \mathsf{pred}_L^{(b)}(I)$, and similarly $J \le \mathsf{pred}_R^{(b)}(I)$.

Now we prove the "if" part of Item (2). We show there exists some I_1 such that $\mathsf{pred}_L(I) \le I_1 \le \mathsf{pred}_R(I)$ and $\mathsf{pred}_L^{(b-1)}(I_1) \le J \le \mathsf{pred}_R^{(b-1)}(I_1)$, which by induction hypothesis on Item (2) implies a partition of the segment $[J..I-1]$ into b stable segments. Suppose there is no such I_1, then by the monotonicity of $\mathsf{pred}_L^{(b-1)}$ and $\mathsf{pred}_R^{(b-1)}$, there must exist $\mathsf{pred}_L(I) \le I' < \mathsf{pred}_R(I)$ such that $\mathsf{pred}_R^{(b-1)}(I') < J < \mathsf{pred}_L^{(b-1)}(I'+1)$, contradicting the induction hypothesis on Item (3).

Finally we prove Item (3). Suppose there exists some I such that $\mathsf{pred}_R^{(b)}(I) < \mathsf{pred}_L^{(b)}(I+1) - 1$. By Lemma 4.3, there exists I' such that $[\mathsf{pred}_R^{(b)}(I)+1..I'-1]$ can be partitioned into b stable segments. Hence by Item (2), $\mathsf{pred}_L^{(b)}(I') \le \mathsf{pred}_R^{(b)}(I) + 1 \le \mathsf{pred}_R^{(b)}(I')$. Then, $\mathsf{pred}_L^{(b)}(I') \le \mathsf{pred}_R^{(b)}(I) + 1 < \mathsf{pred}_L^{(b)}(I+1)$ implies $I' < I + 1$; while $\mathsf{pred}_R^{(b)}(I') \ge \mathsf{pred}_R^{(b)}(I) + 1$ implies $I' > I$. A contradiction. $\square$

Given a stable partition $\mathcal{P}$, we can define a predecessor function for $\mathcal{P}$ as follows.

Lemma 4.9 (Stable predecessor for a stable partition)**.** *Let $\mathcal{P} = (\mathcal{P}_i)_i$ be a stable partition where $\mathcal{P}_i = [p_i..p_{i+1} - 1]$ and $p_i < p_{i+1}$. Then there exists a non-decreasing function* $\mathsf{pred}_{\mathcal{P}} \colon \mathbb{Z} \to \mathbb{Z}$ *such that the following hold:*

- *For every i,* $\mathsf{pred}_{\mathcal{P}}(p_{i+1}) = p_i$*.*

- *For every I, we have $\mathsf{pred}_{\mathcal{P}}(I) \le I-1$, and $[\mathsf{pred}_{\mathcal{P}}(I)..I-1]$ is a stable segment.*

Proof. For any fixed i, it suffices to determine the values of $\mathsf{pred}_{\mathcal{P}}(p_i+j)$ for $j = 1, 2, \ldots, p_{i+1}-p_i-1$.

We simply define $\mathsf{pred}_{\mathcal{P}}(p_i+j) = \max\{\mathsf{pred}_{\mathcal{P}}(p_i+j-1), \mathsf{pred}_L(p_i+j)\}$. By Lemma 4.7 and induction on j, we have $\mathsf{pred}_L(p_i+j) \le \mathsf{pred}_{\mathcal{P}}(p_i+j) \le \mathsf{pred}_R(p_i+j)$. □

Similarly as in Lemma 4.8, we can define the b-stable predecessor $\mathsf{pred}^{(b)}_{\mathcal{P}}(\cdot)$ for a stable partition $\mathcal{P}$, which is sandwiched between $\mathsf{pred}^{(b)}_L(\cdot)$ and $\mathsf{pred}^{(b)}_R(\cdot)$, and also satisfies the Item (1) in Lemma 4.8. As an example, in Figure 1 $\mathsf{pred}^{(3)}_{\mathcal{P}}(8) = 1$.

Corollary 4.10 (b-stable predecessors $\mathsf{pred}^{(b)}_{\mathcal{P}}(\cdot)$)**.** *Let $\mathcal{P}$ be a stable partition. Define*

$$\mathsf{pred}^{(b)}_{\mathcal{P}}(I) = \begin{cases} I & b = 0, \\ \mathsf{pred}_{\mathcal{P}}(\mathsf{pred}^{(b-1)}_{\mathcal{P}}(I)) & b \ge 1. \end{cases}$$

Then for any I, b, we have $\mathsf{pred}^{(b)}_{\mathcal{P}}(I) \le \mathsf{pred}^{(b)}_{\mathcal{P}}(I+1)$.

Moreover, $(I, \mathsf{pred}^{(b)}_{\mathcal{P}}(I))$ is a $(\mathcal{P}, b)$-stable state and $\mathsf{pred}^{(b)}_L(I) \le \mathsf{pred}^{(b)}_{\mathcal{P}}(I) \le \mathsf{pred}^{(b)}_R(I)$.

4.2 Edit Distances for Stable States

We will bound the edit distance between stable states using the number of singletons.

Definition 4.11 (Singleton)**.** Every unmatched $X[i]$ or $Y[j]$ is called a *singleton*.

Let $\mathsf{sing}_X[l, r)$ (resp., $\mathsf{sing}_Y[l, r)$) denote the number of singletons in $X[l..r-1]$ (resp., $Y[l..r-1]$). Let $\mathsf{sing}[l, r) := \mathsf{sing}_X[l, r) + \mathsf{sing}_Y[l, r)$.

We start with 1-stable states, i.e., stable segments.

Lemma 4.12. *Let $\mathcal{P}$ be a stable partition. For $I < I'$, let $J = \mathsf{pred}_{\mathcal{P}}(I), J' = \mathsf{pred}_{\mathcal{P}}(I')$. Then*

(a) $\mathsf{ed}(X[I..I'-1], Y[J..J'-1]) \le \mathsf{sing}_X[I, I') + \mathsf{sing}_Y[J, J') \le \mathsf{sing}[J, I')$*;*

(b) $|(I'-J') - (I-J)| = |(I'-I) - (J'-J)| \le \mathsf{sing}_X[I, I') + \mathsf{sing}_Y[J, J') \le \mathsf{sing}[J, I')$*.*

Proof. For every fixed $i \in [I, I']$, let $j_0 = \mathsf{pred}_{\mathcal{P}}(i), j_1 = \mathsf{pred}_{\mathcal{P}}(i+1)$. By Lemma 4.9, we observe that $X[i]$ is either unmatched or matched to one of the characters $Y[j_0], Y[j_0+1], \ldots, Y[j_1-1]$; and also conversely these characters can only be matched to $X[i]$. Hence,

$$\mathsf{ed}(X[i..i], Y[j_0..j_1-1]) \le \mathsf{sing}_X[i, i+1) + \mathsf{sing}_Y[j_0, j_1).$$

Summing up over all $i = I, I+1, \ldots, I'$, we obtain

$$\mathsf{ed}(X[I..I'-1], Y[J..J'-1]) \le \mathsf{sing}_X[I, I') + \mathsf{sing}_Y[J, J').$$

On the other hand, by Fact 2.2 we have $\mathsf{ed}(X[I..I'-1], Y[J..J'-1]) \ge |(I'-I) - (J'-J)|$. Hence Item (b) follows immediately from Item (a). □

Corollary 4.13. *Let $\mathcal{P}$ be a stable partition. For any I, let $J = \mathsf{pred}_{\mathcal{P}}(I)$, and let $I^{(b)} = \mathsf{pred}^{(b)}_{\mathcal{P}}(I), J^{(b)} = \mathsf{pred}^{(b+1)}_{\mathcal{P}}(I)$ for $b \ge 0$. Then*

(a) $\mathsf{ed}(X[J..I-1], Y[J^{(b)}..I^{(b)}-1]) \le \mathsf{sing}_X[I^{(b)}, I) + \mathsf{sing}_Y[J^{(b)}, J) \le \mathsf{sing}[J^{(b)}, I)$*;*

(b) $|(I-I^{(b)}) - (J-J^{(b)})| = |(I-J) - (I^{(b)}-J^{(b)})| \le \mathsf{sing}_X[I^{(b)}, I) + \mathsf{sing}_Y[J^{(b)}, J) \le \mathsf{sing}[J^{(b)}, I)$*.*

Proof. For every $0 \le c < b$, by applying Lemma 4.12 to $J^{(c)}$ and $I^{(c)}$, we obtain

$$\mathsf{ed}(X[J^{(c)}..I^{(c)}-1], Y[J^{(c+1)}..I^{(c+1)}-1]) \le \mathsf{sing}_X[J^{(c)}, I^{(c)}) + \mathsf{sing}_Y[J^{(c+1)}, I^{(c+1)}).$$

Then by triangle inequality,

$$\begin{aligned}
\mathsf{ed}(X[J..I-1], Y[J^{(b)}..I^{(b)}-1]) &\le \sum_{0\le c<b} \mathsf{ed}(X[J^{(c)}..I^{(c)}-1], Y[J^{(c+1)}..I^{(c+1)}-1]) \\
&\le \sum_{0\le c<b} \left(\mathsf{sing}_X[J^{(c)}, I^{(c)}) + \mathsf{sing}_Y[J^{(c+1)}, I^{(c+1)})\right) \\
&= \mathsf{sing}_X[I^{(b)}, I) + \mathsf{sing}_Y[J^{(b)}, J) \\
&\le \mathsf{sing}[J^{(b)}, I).
\end{aligned}$$

On the other hand, Item (b) follows from Item (a) by Fact 2.2. ◻

Now we extend it to b-stable states.

Corollary 4.14. *Let $\mathcal{P}$ be a stable partition. For $I_1 \le I_2$ and any fixed $b \ge 0$, let $I_1^{(b)} = \mathsf{pred}_{\mathcal{P}}^{(b)}(I_1), I_2^{(b)} = \mathsf{pred}_{\mathcal{P}}^{(b)}(I_2)$. Then*

$$\left|(I_1 - I_1^{(b)}) - (I_2 - I_2^{(b)})\right| \le b \cdot \mathsf{sing}[I_1^{(b)}, I_2).$$

Moreover, the following generalization also holds. Suppose $I_1 \le I_2 \le \cdots \le I_m$, then

$$\sum_{j=1}^{m-1} \left|(I_j - I_j^{(b)}) - (I_{j+1} - I_{j+1}^{(b)})\right| \le b \cdot \mathsf{sing}[I_1^{(b)}, I_m).$$

Proof. Since

$$I_j - I_j^{(b)} = \sum_{c=0}^{b-1}(I_j^{(c)} - I_j^{(c+1)}),$$

we have

$$\begin{aligned}
\left|(I_j - I_j^{(b)}) - (I_{j+1} - I_{j+1}^{(b)})\right| &\le \sum_{c=0}^{b-1} \left|(I_j^{(c)} - I_j^{(c+1)}) - (I_{j+1}^{(c)} - I_{j+1}^{(c+1)})\right| \\
&\le \sum_{c=0}^{b-1} \left(\mathsf{sing}_X[I_j^{(c)}, I_{j+1}^{(c)}) + \mathsf{sing}_Y[I_j^{(c+1)}, I_{j+1}^{(c+1)})\right). \qquad \text{(due to Lemma 4.12)}
\end{aligned}$$

Summing up over all $j = 1, 2, \ldots, m-1$, we obtain

$$\begin{aligned}
\sum_{j=1}^{m-1} \left|(I_j - I_j^{(b)}) - (I_{j+1} - I_{j+1}^{(b)})\right| &\le \sum_{j=1}^{m-1}\sum_{c=0}^{b-1} \left(\mathsf{sing}_X[I_j^{(c)}, I_{j+1}^{(c)}) + \mathsf{sing}_Y[I_j^{(c+1)}, I_{j+1}^{(c+1)})\right) \\
&= \sum_{c=0}^{b-1} \left(\mathsf{sing}_X[I_1^{(c)}, I_m^{(c)}) + \mathsf{sing}_Y[I_1^{(c+1)}, I_m^{(c+1)})\right) \\
&\le b \cdot \mathsf{sing}_X[I_1^{(b-1)}, I_m) + b \cdot \mathsf{sing}_Y[I_1^{(b)}, I_m^{(1)}) \\
&\le b \cdot \mathsf{sing}[I_1^{(b)}, I_m).
\end{aligned}$$

◻

We end this subsection with the following lemma, which will be used in Subsection 4.3.

Lemma 4.15. *Let $\mathcal{P}$ be a stable partition. For any I and $b \ge 1$, let $J = \mathsf{pred}_{\mathcal{P}}^{(b)}(I)$, and let $d = I - J, I' = I + d$. Then $\mathsf{ed}(X[I..I'-1], Y[J..I-1]) \le 4b \cdot \mathsf{sing}[J, I')$.*

Proof. The function $\mathsf{pred}_{\mathcal{P}}$ divides the string $Y[J..I-1]$ into b stable segments, which correspond to substrings $\sigma_{-b}, \sigma_{-b+1}, \ldots, \sigma_{-1}$ from left to right, where $\sigma_{-i} := Y[\mathsf{pred}_{\mathcal{P}}^{(i)}(I)..\mathsf{pred}_{\mathcal{P}}^{(i-1)}(I) - 1]$. Suppose $\mathcal{P}$ induces the substrings $\sigma_0, \sigma_1, \sigma_2, \ldots$ immediately after σ_{-1} (see Figure 2).

Assume I' is contained in the stable segment corresponding to substring σ_q for some $q \ge 0$. We divide the proof into the following two cases.

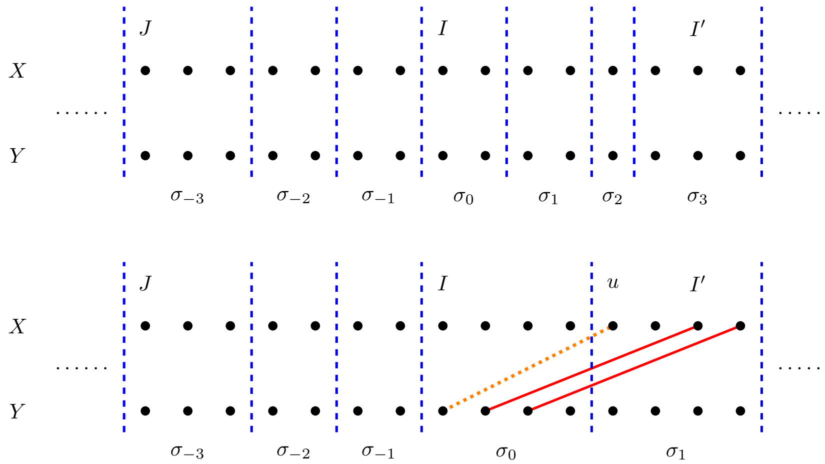

Figure 2: Visualization for the proof of Lemma 4.15, where the top one is for Case 1 and the bottom one is for Case 2. In both cases $b = 3$. In Case 1, $q = 3$. In Case 2, $q = 1$.

Case 1: $q \geq b$. There are at least b stable segments in $X[I..I'-1]$, which corresponds to substrings $\sigma_0, \sigma_1, \ldots, \sigma_{b-1}$. By Corollary 4.13, $\mathsf{ed}(\sigma_{i-b}, \sigma_i) \leq \mathsf{sing}[J, I')$ holds for every $0 \leq i \leq b-1$. Then,

$$\mathsf{ed}(\sigma_0 \circ \sigma_1 \circ \cdots \circ \sigma_{b-1}, Y[J..I-1]) \leq \sum_{i=0}^{b-1} \mathsf{ed}(\sigma_{i-b}, \sigma_i) \leq b \cdot \mathsf{sing}[J, I').$$

By Proposition 2.3, we immediately have

$$\mathsf{ed}(X[I..I'-1], Y[J..I-1]) \leq 2b \cdot \mathsf{sing}[J, I').$$

Case 2: $0 \leq q < b$. Suppose σ_q starts at $X[u]$. Then $X[u..I'-1]$ is a prefix of σ_q. There are $r = (I'-u) - \mathsf{sing}_X[u, I')$ matched edges (i, j) that satisfy $u \leq i < I'$ (see the dotted orange line in Figure 2). Since $Y[j]$ must be contained in σ_{q-1}, we have

$$\begin{aligned}
\mathsf{ed}(X[u..I'-1], \sigma_{q-1}) &\leq (I'-u) + |\sigma_{q-1}| - 2r \\
&= 2 \cdot \mathsf{sing}_X[u, I') + |\sigma_{q-1}| - (I'-u) \\
&\leq 2 \cdot \mathsf{sing}[u, I') + |\sigma_{q-1}| - (I'-u).
\end{aligned}$$

Note that

$$\begin{aligned}
I' - u &= (I' - I) - (|\sigma_0| + |\sigma_1| + \cdots + |\sigma_{q-1}|) \\
&= (|\sigma_{-b}| + |\sigma_{-b+1}| + \cdots + |\sigma_{-1}|) - (|\sigma_0| + |\sigma_1| + \cdots + |\sigma_{q-1}|) \\
&\geq |\sigma_{-1}| + \sum_{0 \leq j < q} (|\sigma_{-b+j}| - |\sigma_j|) \\
&\geq |\sigma_{-1}| - q \cdot \mathsf{sing}[J, u). && \text{(due to Corollary 4.13)}
\end{aligned}$$

Combining two inequalities, we have

$$\begin{aligned}
\mathsf{ed}(X[u..I'-1], \sigma_{-1}) &\leq \mathsf{ed}(X[u..I'-1], \sigma_{q-1}) + \mathsf{ed}(\sigma_{q-1}, \sigma_{-1}) \\
&\leq 2 \cdot \mathsf{sing}[u, I') + q \cdot \mathsf{sing}[J, u) + |\sigma_{q-1}| - |\sigma_{-1}| + \mathsf{ed}(\sigma_{q-1}, \sigma_{-1}) \\
&\leq 2 \cdot \mathsf{sing}[u, I') + q \cdot \mathsf{sing}[J, u) + 2 \cdot \mathsf{sing}[J, u) && \text{(due to Corollary 4.13)} \\
&= 2 \cdot \mathsf{sing}[J, I') + q \cdot \mathsf{sing}[J, u).
\end{aligned}$$

Hence, by Proposition 2.3, we have

$$\begin{aligned}\mathsf{ed}(X[I..I'-1], Y[J..I-1]) &\le 2\cdot \mathsf{ed}(X[I..I'-1], \sigma_{-q-1}\circ\cdots\circ\sigma_{-1})\\ &\le 2\cdot \mathsf{ed}(\sigma_0\circ\cdots\circ\sigma_{q-1}, \sigma_{-q-1}\circ\cdots\circ\sigma_{-2}) + 2\cdot \mathsf{ed}(X[u, I'-1], \sigma_{-1})\\ &\le 2q\cdot \mathsf{sing}[J,u] + 2\cdot(2\cdot \mathsf{sing}[J,I'] + q\cdot \mathsf{sing}[J,u])\\ &\le 4b\cdot \mathsf{sing}[J,I']. \end{aligned}$$

□

4.3 Catch-up and Stabilize

This subsection is devoted for Lemma 4.16, which shows a CGK random walk goes from a stable state to a distant stable state with low cost. This process consists of a "catch-up phase" (i.e., from a stable state to a distant non-stable state) and then a "stabilization phase" (i.e., from a non-stable state to a nearby stable state).

Lemma 4.16 (From stable to stable)**.** *Consider a CGK random walk starting from a stable state $(I_0, J_0), I_0 > J_0$. Let D be a distance bound satisfying $D \ge I_0 - J_0$.*

Consider the first time $T > 0$ that either $I_T - J_T > D$, or the following three conditions hold simultaneously: $J_T \ge I_0$, and $I_T \ge 2I_0 - J_0$, and (I_T, J_T) is a stable state.[17] *Let P be the number of progress steps before time T and let $S = \mathsf{sing}[J_0, I_T)$. Then*

$$\mathbb{E}\left[P - 2000\cdot\left(S\cdot D + S^2\right)\right] \le 0.$$

The proof of this lemma relies on Lemma 4.15 and Lemma 2.13 to control the catch-up phase, and the following technical lemma to control the stabilization phase. The proof of Lemma 4.17 is deferred to Subsection 4.5.

Lemma 4.17 (From non-stable to stable)**.** *Consider a CGK random walk starting from a non-stable state $(\tilde{I}_0, \tilde{J}_0), \tilde{I}_0 > \tilde{J}_0$. Let $\mathcal{P}$ be a stable partition and let b' be such that $\tilde{L}_0 < \tilde{J}_0 < \tilde{R}_0$ where $\tilde{L}_0 = \mathsf{pred}_{\mathcal{P}}^{(b')}(\tilde{I}_0), \tilde{R}_0 = \mathsf{pred}_{\mathcal{P}}^{(b'-1)}(\tilde{I}_0)$. Let D be a distance bound satisfying $D \ge \tilde{I}_0 - \tilde{J}_0$.*

Consider the first time T' that either $(\tilde{I}_{T'}, \tilde{J}_{T'})$ is a stable state or $\tilde{I}_{T'} - \tilde{J}_{T'} > D$.[18] *Let P' be the number of progress steps before time T' and let $S' = \mathsf{sing}[\tilde{L}_0, \tilde{I}_{T'})$. Then*

$$\mathbb{E}\left[P' - 50\cdot\left((\tilde{R}_0 - \tilde{J}_0)(\tilde{J}_0 - \tilde{L}_0) + S'\cdot D + S'^2\right)\right] \le 0.$$

Now we present the proof of Lemma 4.16.

Proof of Lemma 4.16. If $S = 0$, then $X[J_0..I_T - 1] = Y[J_0..I_T - 1]$ is periodic string and the period divides $I_0 - J_0$; hence $P = 0$. Therefore we assume $S \ge 1$.

Catch-up phase. Consider the first time Q that either $I_Q - J_Q > D$, or $J_Q \ge I_0$ and $I_Q \ge 2I_0 - J_0$. Let $\hat{P}$ be the number of progress steps before time Q. Then

$$\mathbb{E}[\hat{P}] \le 4\cdot(I_0 - J_0) \le 4\cdot D \le 4\cdot \mathbb{E}[S\cdot D]. \tag{4}$$

If $I_Q - J_Q > D$ or (I_Q, J_Q) is a stable state, then $T = Q$ and $P = \hat{P}$; hence the bound holds naturally. Thus we focus on the case $J_Q \ge I_0$ and $I_Q \ge 2I_0 - J_0$ and (I_Q, J_Q) is non-stable from now on.

Stabilization phase. By the definition of Q, we know $T > Q$ is the first time that either (I_T, J_T) is a stable state, or $I_T - J_T > D$. Let P' be the number of progress steps from time Q to time T. Applying Lemma 4.17 with $\tilde{I}_0 = I_Q, \tilde{J}_0 = J_Q$ and inheriting its notations, we have

$$\mathbb{E}\left[P' - 50\cdot\left((\tilde{R}_0 - \tilde{J}_0)(\tilde{J}_0 - \tilde{L}_0) + S'\cdot D + S'^2\right)\right] \le 0. \tag{5}$$

To relate $(\tilde{R}_0 - \tilde{J}_0)(\tilde{J}_0 - \tilde{L}_0)$ with S and D, we will prove the following claim.

Claim 4.18. $\mathbb{E}\left[\left(\tilde{R}_0 - \tilde{J}_0\right)\left(\tilde{J}_0 - \tilde{L}_0\right)\right] \le 20\cdot \mathbb{E}\left[S\cdot D\right]$.

[17] This time T is almost surely well-defined, as any state (I, J) satisfying $I \ge J > L$ is a stable state.

[18] This time T' is almost surely well-defined, as any state (I, J) satisfying $I \ge J > L$ is a stable state.

Final bounds. Since $P = \hat{P} + P'$ and $S' \le S$, by (4), (5), and Claim 4.18 we have

$$\mathbb{E}\left[P - 2000 \cdot \left(S \cdot D + S^2\right)\right] \le \mathbb{E}\left[P - 4S \cdot D - 50 \cdot \left(20S \cdot D + S \cdot D + S^2\right)\right] \le 0. \qquad \square$$

Proof of Claim 4.18. Assume (I_0, J_0) is $(\mathcal{P}, b)$-stable where $b \ge 1$. By Lemma 4.9 we know $J_0 = \mathsf{pred}_{\mathcal{P}}^{(b)}(I_0)$. By Item (2) in Lemma 2.13, we have

$$\begin{aligned}
\mathbb{E}\left[\left|(\tilde{I}_0 - I_0) - (\tilde{J}_0 - J_0)\right|\right] &\le 4 \cdot \mathsf{ed}(X[I_0..2I_0 - J_0 - 1], Y[J_0..I_0 - 1]) \\
&\le 16b \cdot \mathsf{sing}[J_0, 2I_0 - J_0) && \text{(due to Lemma 4.15)} \\
&\le 16b \cdot S.
\end{aligned}$$

Let $\tilde{J}_0' := \mathsf{pred}_{\mathcal{P}}^{(b)}(\tilde{I}_0)$. By Corollary 4.14, we have $\left|(\tilde{I}_0 - \tilde{J}_0') - (I_0 - J_0)\right| \le b \cdot \mathsf{sing}[J_0, \tilde{I}_0) \le b \cdot S$. Hence

$$\mathbb{E}\left[\left|\tilde{J}_0 - \tilde{J}_0'\right|\right] \le \mathbb{E}\left[\left|(\tilde{I}_0 - I_0) - (\tilde{J}_0 - J_0)\right| + \left|(\tilde{I}_0 - \tilde{J}_0') - (I_0 - J_0)\right|\right] \le 17b \cdot \mathbb{E}\left[S\right]. \tag{6}$$

On the other hand, since $\tilde{I}_0 \ge \tilde{J}_0, \tilde{J}_0'$, we have

$$|\tilde{J}_0 - \tilde{J}_0'| \le (\tilde{I}_0 - \tilde{J}_0) + (\tilde{I}_0 - \tilde{J}_0') \le (\tilde{I}_0 - \tilde{J}_0) + (\tilde{I}_0 - I_0) + (I_0 - J_0) \le 3 \cdot D, \tag{7}$$

For $\tilde{J}_0$, we know $\tilde{L}_0$ and $\tilde{R}_0$ are the closest boundaries of stable segments induced by $\tilde{I}_0$ and $\mathcal{P}$, hence

$$\min\left\{\tilde{R}_0 - \tilde{J}_0, \tilde{J}_0 - \tilde{L}_0\right\} \le |\tilde{J}_0 - \tilde{J}_0'|. \tag{8}$$

Since $\tilde{R}_0 \ge \tilde{J}_0 \ge I_0$, we apply Lemma 4.12 and obtain

$$\left(\tilde{R}_0 - \tilde{L}_0\right) - \left(\mathsf{pred}_{\mathcal{P}}^{(j)}(I_0) - \mathsf{pred}_{\mathcal{P}}^{(j+1)}(I_0)\right) \le \mathsf{sing}[\mathsf{pred}_{\mathcal{P}}^{(j+1)}(I_0), \tilde{R}_0) \le S,$$

for all $0 \le j \le b-1$. Taking average over j, we obtain

$$\tilde{R}_0 - \tilde{L}_0 \le S + \frac{I_0 - J_0}{b} \le S + \frac{D}{b}. \tag{9}$$

Therefore we have

$$\begin{aligned}
&\mathbb{E}\left[\left(\tilde{R}_0 - \tilde{J}_0\right)\left(\tilde{J}_0 - \tilde{L}_0\right)\right] \\
&\le \mathbb{E}\left[\left(\tilde{R}_0 - \tilde{L}_0\right) \cdot \min\left\{\tilde{R}_0 - \tilde{J}_0, \tilde{J}_0 - \tilde{L}_0\right\}\right] \\
&\le \mathbb{E}\left[\left|\tilde{J}_0 - \tilde{J}_0'\right| \cdot \left(\frac{D}{b} + S\right)\right] && \text{(due to (8) and (9))} \\
&\le 17 \cdot \mathbb{E}\left[S \cdot D\right] + 3 \cdot \mathbb{E}\left[S \cdot D\right] && \text{(due to (6) and (7))} \\
&= 20 \cdot \mathbb{E}\left[S \cdot D\right].
\end{aligned} \qquad \square$$

4.4 Proof of Proposition 3.9

Given previous lemmas to control progress steps, we now prove Proposition 3.9.

Proposition (Proposition 3.9 restated)**.** *There exists a universal constant $C_4 \ge 1$ such that the following holds. Assume X, Y are two identical length-L strings over alphabet Σ. Assume there exists a size-M matching $(i_1, j_1), \ldots, (i_M, j_M) \in [L]^2$ such that*

- *$i_t > j_t$ and $X[i_t] = Y[j_t]$ hold for all $t \in [M]$;*
- *$i_1 < i_2 < \cdots < i_M$ and $j_1 < j_2 < \cdots < j_M$.*

Let $\rho = C_4 \cdot (L - M)$ and $(\hat{I}, \hat{J})$ be any state satisfying $\hat{I} - \hat{J} \geq \rho$. Then a CGK random walk on X, Y starting from $(\hat{I}, \hat{J})$ will miss (L, L) with probability at least 0.5.

Proof. Since $X[1]$ and $Y[L]$ are matched to dummy characters after we extend X, Y, there are $K := \mathsf{sing}[-\infty, +\infty) = 2 \cdot (L - M - 1)$ singletons in total. Let $d := \hat{I} - \hat{J}$ be the initial distance between the two pointers and let $D := 2 \cdot d$. For a state $(I, J), I \geq J$,

- if $I = J \leq L$ or $I - J > D$, then we say it is a *failure state*;
- if it is not a failure state and $I > L$, then we say it is a *success state*.

We stop the CGK random walk when it reaches a success state or a failure state. The former case implies that the random walk misses (L, L). So it suffices to prove that we stop at a success state with probability at least 0.5.

Phases in the CGK random walk. Let $I_0 = \hat{I}, J_0 = \hat{J}$ and $t_0 = 0$. Let $t_1 \geq 0$ be the first time that either (I_{t_1}, J_{t_1}) is a stable state or $I_{t_1} - J_{t_1} > D$.[19]

For every $i \geq 2$, if $(I_{t_{i-1}}, J_{t_{i-1}})$ is neither a success state nor a failure state, we know $J_{t_{i-1}} < I_{t_{i-1}} \leq L$ and $I_{t_{i-1}} - J_{t_{i-1}} \leq D$. Then we recursively define $t_i > t_{i-1}$ to be the first time that either $I_{t_i} - J_{t_i} > D$, or the following three conditions hold simultaneously: $I_{t_i} \geq 2I_{t_{i-1}} - J_{t_{i-1}}$, and $J_{t_i} \geq I_{t_{i-1}}$, and (I_{t_i}, J_{t_i}) is a stable state.

Assume we stop at (I_{t_m}, J_{t_m}), which is either a success state or a failure state.[20] Let P_i be the number of progress steps made during the time interval $[t_i, t_{i+1})$. Then $P := \sum_{i=0}^{m-1} P_i$ is the total number of progress steps before we stop.

Bounds on $\mathbb{E}[P_0]$. Let $\mathcal{P}$ be an arbitrary stable partition and let b be such that $\mathsf{pred}_{\mathcal{P}}^{(b)}(I_0) \leq J_0 < \mathsf{pred}_{\mathcal{P}}^{(b-1)}(I_0)$. Let $L_0 = \mathsf{pred}_{\mathcal{P}}^{(b)}(I_0), R_0 = \mathsf{pred}_{\mathcal{P}}^{(b-1)}(I_0)$. Since $X[1]$ is matched to $Y[0]$, we know $\mathsf{pred}_{\mathcal{P}}(1) = 0$. Hence applying Lemma 4.12 with $I' = R_0, I = 1$, we have

$$(R_0 - L_0) - (1 - 0) \leq \mathsf{sing}[0, R_0) \leq \mathsf{sing}[0, I_0).$$

Therefore, let $S_0 = \mathsf{sing}[0, I_{t_1})$ and we have

$$(J_0 - L_0)(R_0 - J_0) \leq \left\lfloor \frac{R_0 - L_0}{2} \right\rfloor \cdot \left\lceil \frac{R_0 - L_0}{2} \right\rceil \leq (\mathsf{sing}[0, I_0))^2 \leq S_0^2.$$

Thus by Lemma 4.17, we have $\mathbb{E}\left[P_0 - 50 \cdot \left(2 \cdot S_0^2 + S_0 \cdot D\right)\right] \leq 0$.

Bounds on $\mathbb{E}[P_i], 1 \leq i \leq m - 1$. Let $S_i := \mathsf{sing}[J_{t_i}, I_{t_{i+1}})$. By Lemma 4.16, we have

$$\mathbb{E}\left[P_i - 2000 \cdot \left(S_i \cdot D + S_i^2\right)\right] \leq 0.$$

Final bounds. Note that $\sum_{0 \leq i < m} S_i \leq 2 \cdot \mathsf{sing}[0, I_{t_m}) \leq 2 \cdot K$. This is because $J_{t_{i+1}} \geq I_{t_i}$ for all $i \geq 1$, implying each singleton is counted at most twice. Hence

$$\mathbb{E}[P] = \mathbb{E}\left[\sum_{i=0}^{m-1} P_i\right] \leq \mathbb{E}\left[2000 \sum_{i=0}^{m-1} (S_i D + S_i^2)\right] \leq 2000 \cdot \mathbb{E}\left[D \sum_{i=0}^{m-1} S_i + \left(\sum_{i=0}^{m-1} S_i\right)^2\right] \leq 8000 \cdot (K \cdot d + K^2).$$

For $1 \leq j < +\infty$, let r_j be the deviation brought by the j-th progress step.[21] Then r_j are i.i.d. random variables with

$$\mathbf{Pr}[r_j = 0] = 1/2, \quad \mathbf{Pr}[r_j = +1] = \mathbf{Pr}[r_j = -1] = 1/4.$$

[19] By our definition, any state $(I, J) > (L, L)$ is a stable state. Hence t_1 is almost surely well-defined.

[20] Since the random walk walks through $X[1..L], Y[1..L]$ almost surely, m is almost surely well-defined.

[21] Though we will only use $r_1, \ldots, r_P$, we define it in this way to make the next Cauchy-Schwarz inequality easier to understand.

Hence by Cauchy-Schwarz inequality, we have

$$\mathbb{E}\left[\left|\sum_{j=1}^{P} r_j\right|\right] = \mathbb{E}\left[\left|\sum_{j=1}^{+\infty} r_j \cdot 1_{\{j\le P\}}\right|\right] \le \sqrt{\mathbb{E}\left[\left(\sum_{j=1}^{+\infty} r_j \cdot 1_{\{j\le P\}}\right)^2\right]} = \sqrt{\mathbb{E}\left[\frac{P}{2}\right]} \le \sqrt{4000\cdot(K\cdot d + K^2)}.$$

Observe that in the end we have $I_{t_m} - J_{t_m} = d + \sum_{j=1}^{P} r_j$. By setting $\rho = C_4 \cdot (L - M)$ for some large enough constant C_4, we have $d \ge \rho \ge C_4 \cdot K/2$ and

$$d \ge 4 \cdot \sqrt{4000\cdot(K\cdot d + K^2)} \ge 4 \cdot \mathbb{E}\left[\left|\sum_{j=1}^{P} r_j\right|\right].$$

Then by Markov's inequality, with probability at least 0.5 we have $|I_{t_m} - J_{t_m} - d| \le d/2$, which indicates (I_{t_m}, J_{t_m}) is not a failure state. Hence we stop at some success state with probability at least 0.5. ☐

4.5 Proof of Lemma 4.17

This subsection is devoted to the proof of Lemma 4.17. We first prove the following useful lemma.

Lemma 4.19. *Let (I, J) and (I', J') be two states where $I' \ge I$ and $J' \ge J$. Suppose at least one of the following two conditions holds:*

(1) $J \le \mathsf{pred}_R^{(b)}(I)$ and $J' \ge \mathsf{pred}_L^{(b)}(I')$.

(2) $J \ge \mathsf{pred}_L^{(b)}(I)$ and $J' \le \mathsf{pred}_R^{(b)}(I')$.

Then every possible CGK random walk from (I, J) to (I', J') must contain a b-stable state.

Proof. Let $(I_0, J_0) = (I, J)$ be the starting state. Then there are four possible transitions during the walk

$$(I_{t+1}, J_{t+1}) \in \{(I_t, J_t), (I_t, J_t+1), (I_t+1, J_t), (I_t+1, J_t+1)\}.$$

We first prove Item (1). If there is no b-stable state on the walk, then by Lemma 4.8 for all t, either $J_t \le \mathsf{pred}_L^{(b)}(I_t) - 1$ or $J_t \ge \mathsf{pred}_R^{(b)}(I_t) + 1$. Hence there must exist some t such that $J_t \le \mathsf{pred}_L^{(b)}(I_t) - 1$ and $J_{t+1} \ge \mathsf{pred}_R^{(b)}(I_{t+1}) + 1$. Then

$$J_{t+1} - 1 \ge \mathsf{pred}_R^{(b)}(I_{t+1}) \ge \mathsf{pred}_R^{(b)}(I_t) \ge \mathsf{pred}_L^{(b)}(I_t) \ge J_t + 1.$$

A contradiction.

Now we prove Item (2). Similarly, if there is no b-stable state on the walk, then by Lemma 4.8 there must exist some t such that $J_t \ge \mathsf{pred}_R^{(b)}(I_t) + 1$ and $J_{t+1} \le \mathsf{pred}_L^{(b)}(I_{t+1}) - 1$. Hence

$$J_t - 1 \ge \mathsf{pred}_R^{(b)}(I_t) \ge \mathsf{pred}_L^{(b)}(I_t + 1) - 1 \ge \mathsf{pred}_L^{(b)}(I_{t+1}) - 1 \ge J_{t+1}.$$

A contradiction. ☐

Corollary 4.20. *Let (I, J) and (I', J') be two non-stable states where $I' \ge I$ and $J' \ge J$. Let $\mathcal{P}$ be a stable partition and let b be such that $\mathsf{pred}_{\mathcal{P}}^{(b)}(I) < J < \mathsf{pred}_{\mathcal{P}}^{(b-1)}(I)$.*

If there exists some CGK random walk from (I, J) to (I', J') which does not contain any stable state, then $\mathsf{pred}_{\mathcal{P}}^{(b)}(I') < J' < \mathsf{pred}_{\mathcal{P}}^{(b-1)}(I')$.

Proof. By Corollary 4.10, we have

$$\mathsf{pred}_L^{(b)}(I) \le \mathsf{pred}_{\mathcal{P}}^{(b)}(I) < J < \mathsf{pred}_{\mathcal{P}}^{(b-1)}(I) \le \mathsf{pred}_R^{(b-1)}(I).$$

As there is no b-stable state or $(b-1)$-stable state on the walk, by Lemma 4.19 and Corollary 4.10 we have

$$\mathsf{pred}_{\mathcal{P}}^{(b)}(I') \le \mathsf{pred}_R^{(b)}(I') < J' < \mathsf{pred}_L^{(b-1)}(I') \le \mathsf{pred}_{\mathcal{P}}^{(b-1)}(I').$$ ☐

Now we are ready to prove Lemma 4.17. We restate it here with simpler notations.

Lemma (Lemma 4.17 restated)**.** *Consider a CGK random walk starting from a non-stable state* $(I_0, J_0), I_0 > J_0$*. Let* $\mathcal{P}$ *be a stable partition and let* b *be such that* $L_0 < J_0 < R_0$ *where* $L_0 = \mathsf{pred}_{\mathcal{P}}^{(b)}(I_0), R_0 = \mathsf{pred}_{\mathcal{P}}^{(b-1)}(I_0)$*. Let* D *be a distance bound satisfying* $D \geq I_0 - J_0$*.*

Consider the first time T *that either* (I_T, J_T) *is a stable state or* $I_T - J_T > D$*. Let* P *be the number of progress steps before time* T *and let* $S = \mathsf{sing}[L_0, I_T)$*. Then*

$$\mathbb{E}\left[P - 50 \cdot \left((R_0 - J_0)(J_0 - L_0) + S \cdot D + S^2\right)\right] \leq 0.$$

Proof. Let $t_0 = 0$. Starting from state (I_{t_0}, J_{t_0}), consider the first time t_1 that we reach a state (I_{t_1}, J_{t_1}) that one of the following three conditions holds: $I_{t_1} - J_{t_1} = I_{t_0} - R_{t_0}$, or $I_{t_1} - J_{t_1} = I_{t_0} - L_{t_0}$, or $I_{t_1} \geq J_{t_1} > L$.[22] By Fact 2.12, the expected number of progress steps is at most $2 \cdot (R_{t_0} - J_{t_0})(J_{t_0} - L_{t_0})$.

Note that if $T \leq t_1$, then the bounds hold naturally. Therefore, from now on we focus on the case $T > t_1$, which means $I_{t_1} - J_{t_1} \leq D$ and we have not reached a stable state during these steps. Let $L_{t_1} := \mathsf{pred}_{\mathcal{P}}^{(b)}(I_{t_1})$ and $R_{t_1} := \mathsf{pred}_{\mathcal{P}}^{(b-1)}(I_{t_1})$. By Corollary 4.20, we have $L_{t_1} < J_{t_1} < R_{t_1}$. Then similarly we consider the first time $t_2 > t_1$ that we reach a state (I_{t_2}, J_{t_2}) satisfying one of the following three conditions: $I_{t_2} - J_{t_2} = I_{t_1} - R_{t_1}$, or $I_{t_2} - J_{t_2} = I_{t_1} - L_{t_1}$, or $I_{t_2} \geq J_{t_2} > L$, which takes at most

$$2 \cdot (R_{t_1} - J_{t_1})(J_{t_1} - L_{t_1}) \tag{10}$$

progress steps in expectation. Now we give an upper bound for (10).

- If $I_{t_1} - J_{t_1} = I_{t_0} - R_{t_0}$, then $R_{t_1} - J_{t_1} = (I_{t_0} - R_{t_0}) - (I_{t_1} - R_{t_1})$. Thus
$$(10) \leq 2 \cdot (R_{t_1} - L_{t_1})(R_{t_1} - J_{t_1}) = 2 \cdot (R_{t_1} - L_{t_1})\left((I_{t_0} - R_{t_0}) - (I_{t_1} - R_{t_1})\right).$$
- If $I_{t_1} - J_{t_1} = I_{t_0} - L_{t_0}$, then $J_{t_1} - L_{t_1} = (I_{t_1} - L_{t_1}) - (I_{t_0} - L_{t_0})$. Thus
$$(10) \leq 2 \cdot (R_{t_1} - L_{t_1})(J_{t_1} - L_{t_1}) = 2 \cdot (R_{t_1} - L_{t_1})\left((I_{t_1} - L_{t_1}) - (I_{t_0} - L_{t_0})\right).$$

In either case, we have

$$(10) \leq 2 \cdot (R_{t_1} - L_{t_1}) \cdot \max\left\{(I_{t_0} - R_{t_0}) - (I_{t_1} - R_{t_1}), (I_{t_1} - L_{t_1}) - (I_{t_0} - L_{t_0})\right\}.$$

Similarly we repeatedly generate (I_{t_i}, J_{t_i}) as well as $L_{t_i} = \mathsf{pred}_{\mathcal{P}}^{(b)}(I_{t_i}), R_{t_i} = \mathsf{pred}_{\mathcal{P}}^{(b-1)}(I_{t_i})$, where we have never reached a stable state, or a state with distance greater than D before and including (I_{t_m}, J_{t_m}); while starting from (I_{t_m}, J_{t_m}) we reach (I_T, J_T) which is a stable state, or a state with distance at least D. The expected total number of progress steps is

$$\begin{aligned}
&2 \cdot (R_{t_0} - J_{t_0})(J_{t_0} - L_{t_0}) + 2\sum_{i=1}^{m}(R_{t_i} - J_{t_i})(J_{t_i} - L_{t_i}) \qquad (11)\\
&\leq 2 \cdot (R_{t_0} - J_{t_0})(J_{t_0} - L_{t_0})\\
&\quad + 2\sum_{i=1}^{m}(R_{t_i} - L_{t_i}) \cdot \max\left\{(I_{t_{i-1}} - R_{t_{i-1}}) - (I_{t_i} - R_{t_i}), (I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\right\}.
\end{aligned}$$

Claim 4.21. For any possible outcome of $\{(I_{t_i}, J_{t_i})\}_{i=0}^{m}$ and (I_T, J_T), let $S' = \mathsf{sing}[L_{t_0}, I_{t_m})$ and $S = \mathsf{sing}[L_{t_0}, I_T)$, then we have

$$(11) \leq 50 \cdot \left((R_{t_0} - J_{t_0})(J_{t_0} - L_{t_0}) + {S'}^2 + S' \cdot D\right) \leq 50 \cdot \left((R_0 - J_0)(J_0 - L_0) + S^2 + S \cdot D\right).$$

[22] The third condition implies (I_{t_1}, J_{t_1}) is a stable state.

We now show Claim 4.21 implies the lemma. Consider a tree with the root node representing the initial state (I_{t_0}, J_{t_0}). For each node p in the tree, and every possible walk λ from state (I_{t_i}, J_{t_i}) represented by p to a subsequent state $(I_{t_{i+1}}, J_{t_{i+1}})$, we draw a directed edge, labeled with λ, from p to a child node p' representing $(I_{t_{i+1}}, J_{t_{i+1}})$. We continue this process and include leaf nodes representing stable states or states with distance greater than D. Hence the random process that we are studying is equivalent to a root-to-leaf path on the tree.

Let $\pi(p, p')$ be the probability of arriving at child node p' if we start from node p, and let $\delta(p, p')$ denote the number of progress steps along the walk specified by the edge from p to p'.

- For every non-leaf node p, let $\sigma(p) = \sum_{p' \text{ is a child of } p} \pi(p, p') \cdot \delta(p, p')$. We have seen $\sigma(p) \le 2 \cdot (J_{t_i} - L_{t_i})(R_{t_i} - J_{t_i})$, where (I_{t_i}, J_{t_i}) is the state represented by node p, and L_{t_i}, R_{t_i} are the boundaries of the stable segment containing J_{t_i}.
- For every leaf node p, let $\sigma(p) = -50 \cdot \big((R_0 - J_0)(J_0 - L_0) + S^2 + S \cdot D\big)$, where S is defined for the path from root to p.

Using this formulation, Claim 4.21 states that for every root-to-leaf path $(p_0, p_1, p_2, \ldots, p_m, p_{m+1})$ where p_0 is the root and p_{m+1} is a leaf node, we have $\sum_{i=0}^{m+1} \sigma(p_i) \le 0$. Then,

$$
\begin{aligned}
&\mathbb{E}\left[P - 50 \cdot \big((R_0 - J_0)(J_0 - L_0) + S \cdot D + S^2\big)\right] \\
&= \mathop{\mathbb{E}}_{(p_0, p_1, \ldots, p_m, p_{m+1})} \left[\delta(p_0, p_1) + \delta(p_1, p_2) + \cdots + \delta(p_m, p_{m+1}) + \sigma(p_{m+1})\right] \\
&= \sum_{(p_0, p_1, \ldots, p_m, p_{m+1})} \pi(p_0, p_1)\pi(p_1, p_2) \cdots \pi(p_m, p_{m+1}) \cdot \big(\delta(p_0, p_1) + \delta(p_1, p_2) + \cdots + \delta(p_m, p_{m+1}) + \sigma(p_{m+1})\big) \\
&= \sigma(p_0) + \sum_{(p_0, p_1)} \pi(p_0, p_1)\sigma(p_1) + \cdots + \sum_{(p_0, p_1, \ldots, p_m)} \pi(p_0, p_1)\pi(p_1, p_2) \cdots \pi(p_{m-1}, p_m)\sigma(p_m) \\
&\qquad + \sum_{(p_0, p_1, \ldots, p_m, p_{m+1})} \pi(p_0, p_1)\pi(p_1, p_2) \cdots \pi(p_m, p_{m+1})\sigma(p_{m+1}) \\
&= \mathop{\mathbb{E}}_{(p_0, p_1, \ldots, p_m, p_{m+1})} \left[\sigma(p_0) + \sigma(p_1) + \cdots + \sigma(p_m) + \sigma(p_{m+1})\right] \\
&\le 0.
\end{aligned}
$$

□

It remains to prove Claim 4.21, which we divide into two cases.

Proof of Claim 4.21: Case $b = 1$. Note that when $b = 1$, we can only have $I_{t_i} - J_{t_i} = I_{t_{i-1}} - L_{t_{i-1}}$, since otherwise $I_{t_i} - J_{t_i} = 0$ and hence (I_{t_i}, J_{t_i}) is a 0-stable state. Thus

$$
\begin{aligned}
&\sum_{i=1}^{m} (R_{t_i} - J_{t_i})(J_{t_i} - L_{t_i}) \\
&\le \sum_{i=1}^{m} (R_{t_i} - L_{t_i})(J_{t_i} - L_{t_i}) \\
&= \sum_{i=1}^{m} (R_{t_i} - L_{t_i}) \cdot \big((I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\big) \\
&\le \max_{1 \le i \le m} \{R_{t_i} - L_{t_i}\} \cdot \sum_{i=1}^{m} \big((I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\big) \\
&\le \max_{1 \le i \le m} \{R_{t_i} - L_{t_i}\} \cdot \mathsf{sing}[L_{t_0}, I_{t_m}]. && \text{(due to Corollary 4.14)}
\end{aligned}
$$

For any $i \in [m]$, by Lemma 4.12 we have $(R_{t_i} - L_{t_i}) - (R_{t_0} - L_{t_0}) \le \mathsf{sing}[L_{t_0}, R_{t_i}] = \mathsf{sing}[L_{t_0}, I_{t_i}]$. In addition, since $b = 1$ we have $R_{t_i} = I_{t_i}$, which implies

$$
\max_{1 \le i \le m} \{R_{t_i} - L_{t_i}\} \le (R_{t_0} - L_{t_0}) + \mathsf{sing}[L_{t_0}, R_{t_m}] = (I_{t_1} - J_{t_1}) + \mathsf{sing}[L_{t_0}, I_{t_m}] \le D + \mathsf{sing}[L_{t_0}, I_{t_m}].
$$

Hence by writing $S' = \mathsf{sing}[I_{t_0}, I_{t_m})$, we have

$$(11) \le 2 \cdot (R_{t_0} - J_{t_0})(J_{t_0} - L_{t_0}) + 2 \cdot D \cdot S' + 2 \cdot {S'}^2. \qquad \square$$

Proof of Claim 4.21: Case $b \ge 2$. We partition $[m]$ into groups where each group contains a contiguous interval of indices. The first group starts from 1. The second group starts from $i = \min\{i \mid L_{t_i} \ge I_{t_1}\}$. Similarly, the third group starts from $i' = \min\{i' \mid L_{t_{i'}} \ge I_{t_i}\}$, and so on (see Figure 3).

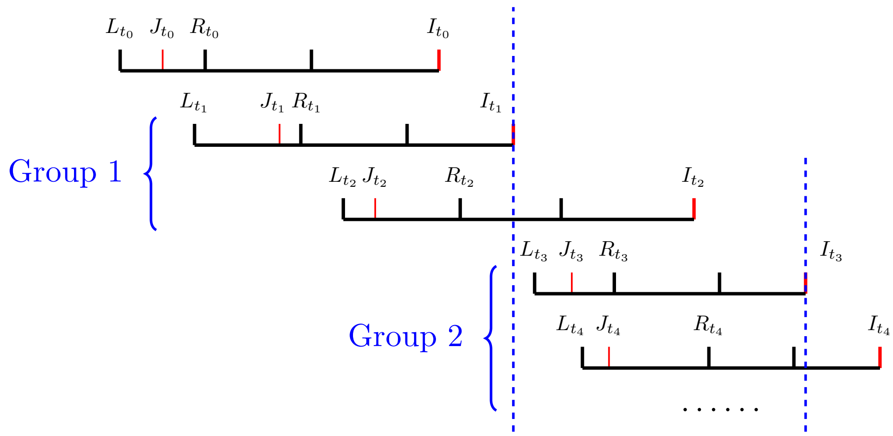

Figure 3: Visualization of the grouping in the proof of Claim 4.21. Here $b = 3$.

For a particular group $[p..q]$, we have

$$\begin{aligned}
&\sum_{i=p}^{q} (R_{t_i} - L_{t_i}) \cdot \max\left\{(I_{t_{i-1}} - R_{t_{i-1}}) - (I_{t_i} - R_{t_i}), (I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\right\} \\
\le &\left(\max_{p \le i \le q} \{R_{t_i} - L_{t_i}\}\right) \cdot \sum_{i=p}^{q} \max\left\{(I_{t_{i-1}} - R_{t_{i-1}}) - (I_{t_i} - R_{t_i}), (I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\right\}. \qquad (12)
\end{aligned}$$

We now bound the first factor of (12). For every $0 \le j \le b$, let $I_{t_i}^{(j)} = \mathsf{pred}_{\mathcal{P}}^{(j)}(I_{t_i})$. Hence we have $L_{t_i} = I_{t_i}^{(b)}, R_{t_i} = I_{t_i}^{(b-1)}$. For any $1 \le j \le b-1$, by Lemma 4.12 we have

$$(R_{t_i} - L_{t_i}) - (I_{t_i}^{(j-1)} - I_{t_i}^{(j)}) \le \mathsf{sing}[L_{t_i}, I_{t_i}^{(j-1)}).$$

Taking average over j, we obtain

$$R_{t_i} - L_{t_i} \le \frac{1}{b-1}(I_{t_i} - R_{t_i}) + \mathsf{sing}[L_{t_i}, I_{t_i}) \le \frac{1}{b-1}(I_{t_i} - J_{t_i}) + \mathsf{sing}[L_{t_i}, I_{t_i}) \le \frac{D}{b-1} + \mathsf{sing}[L_{t_i}, I_{t_i}),$$

which immediately implies

$$\max_{p \le i \le q} \{R_{t_i} - L_{t_i}\} \le \frac{D}{b-1} + \mathsf{sing}[L_{t_p}, I_{t_q}). \qquad (13)$$

Hence by (12), we have

$$\begin{aligned}
&\sum_{i=1}^{m} (R_{t_i} - L_{t_i}) \cdot \max\left\{(I_{t_{i-1}} - R_{t_{i-1}}) - (I_{t_i} - R_{t_i}), (I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\right\} \\
\le &\frac{D}{b-1} \cdot \sum_{i=1}^{m} \max\left\{(I_{t_{i-1}} - R_{t_{i-1}}) - (I_{t_i} - R_{t_i}), (I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\right\} \qquad (14) \\
+ &\sum_{\text{group } g = [p..q]} \mathsf{sing}[L_{t_p}, I_{t_q}) \cdot \sum_{i=p}^{q} \max\left\{(I_{t_{i-1}} - R_{t_{i-1}}) - (I_{t_i} - R_{t_i}), (I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\right\}. \qquad (15)
\end{aligned}$$

Bound (14). Applying Corollary 4.14 and observing $S' = \mathsf{sing}[I_{t_0}, I_{t_m})$, we obtain

$$\begin{aligned}
&\sum_{i=1}^{m} \max\left\{(I_{t_{i-1}} - R_{t_{i-1}}) - (I_{t_i} - R_{t_i}), (I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\right\} \\
&\leq \sum_{i=1}^{m} \left(\left|(I_{t_{i-1}} - R_{t_{i-1}}) - (I_{t_i} - R_{t_i})\right| + \left|(I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\right|\right) \\
&\leq (b-1) \cdot \mathsf{sing}[R_{t_0}, I_{t_m}) + b \cdot \mathsf{sing}[L_{t_0}, I_{t_m}) \\
&\leq (2b-1) \cdot \mathsf{sing}[L_{t_0}, I_{t_m}) \\
&= (2b-1) \cdot S'.
\end{aligned} \tag{16}$$

Bound (15). Note that

$$\begin{aligned}
&\sum_{i=p+1}^{q} \max\left\{(I_{t_{i-1}} - R_{t_{i-1}}) - (I_{t_i} - R_{t_i}), (I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\right\} \\
&= \sum_{i=p+1}^{q} \max\left\{(R_{t_i} - R_{t_{i-1}}) - (I_{t_i} - I_{t_{i-1}}), (I_{t_i} - I_{t_{i-1}}) - (L_{t_i} - L_{t_{i-1}})\right\} \\
&\leq \sum_{i=p+1}^{q} \left((R_{t_i} - R_{t_{i-1}}) + (I_{t_i} - I_{t_{i-1}})\right) \\
&= (R_{t_q} - R_{t_p}) + (I_{t_q} - I_{t_p}) \\
&\leq (R_{t_q} - L_{t_q}) + (I_{t_p} - R_{t_p}) + (I_{t_q} - R_{t_q}) + (R_{t_q} - L_{t_q}). && \text{(due to } I_{t_p} \geq L_{t_q}\text{)}
\end{aligned}$$

We also have

$$\begin{aligned}
&\max\left\{(I_{t_{p-1}} - R_{t_{p-1}}) - (I_{t_p} - R_{t_p}), (I_{t_p} - L_{t_p}) - (I_{t_{p-1}} - L_{t_{p-1}})\right\} \\
&\leq (I_{t_{p-1}} - R_{t_{p-1}}) + (I_{t_p} - L_{t_p}) \\
&= (I_{t_{p-1}} - R_{t_{p-1}}) + (I_{t_p} - R_{t_p}) + (R_{t_p} - L_{t_p}).
\end{aligned}$$

Summing up the above two inequalities and using $I_{t_i} - R_{t_i} \leq I_{t_i} - J_{t_i} \leq D$, we obtain

$$\begin{aligned}
&\sum_{i=p}^{q} \max\left\{(I_{t_{i-1}} - R_{t_{i-1}}) - (I_{t_i} - R_{t_i}), (I_{t_i} - L_{t_i}) - (I_{t_{i-1}} - L_{t_{i-1}})\right\} \\
&\leq 4 \cdot D + 3 \cdot \max\left\{R_{t_q} - L_{t_q}, R_{t_p} - L_{t_p}\right\} \\
&\leq 4 \cdot D + 3 \cdot \left(\frac{D}{b-1} + \mathsf{sing}[L_{t_p}, I_{t_q})\right) && \text{(due to (13))} \\
&\leq 7 \cdot D + 3 \cdot \mathsf{sing}[L_{t_p}, I_{t_q}).
\end{aligned}$$

Hence (15) can be bounded by

$$\begin{aligned}
&\sum_{\text{group } g = [p..q]} \mathsf{sing}[L_{t_p}, I_{t_q}) \cdot \left(7 \cdot D + 3 \cdot \mathsf{sing}[L_{t_p}, I_{t_q})\right) \\
&\leq 7D \cdot \left(\sum_{\text{group } g = [p..q]} \mathsf{sing}[L_{t_p}, I_{t_q})\right) + 3 \cdot \left(\sum_{\text{group } g = [p..q]} \mathsf{sing}[L_{t_p}, I_{t_q})\right)^2 \\
&\leq 14D \cdot S' + 12 \cdot {S'}^2,
\end{aligned} \tag{17}$$

where the last inequality follows from $\sum_{\text{group } g = [p..q]} \mathsf{sing}[L_{t_p}, I_{t_q}) \leq 2 \cdot \mathsf{sing}[L_{t_1}, I_{t_m}) \leq 2 \cdot S'$, as our grouping rule ensures that each singleton appears in at most two groups.

Final bounds. Combining (14), (15), (16), and (17), we have

$$
\begin{aligned}
(11) &\le 2\cdot(R_{t_0}-J_{t_0})(J_{t_0}-L_{t_0}) + \frac{2D}{b-1}\cdot(2b-1)\cdot S' + \left(28D\cdot S' + 24\cdot {S'}^2\right)\\
&\le 2\cdot(R_{t_0}-J_{t_0})(J_{t_0}-L_{t_0}) + 34D\cdot S'\cdot + 24\cdot {S'}^2. \qquad \square
\end{aligned}
$$

5 Discussion

Building upon [BZ16], we present an improved sketching algorithm for edit distance with sketch size $\tilde{O}(k^3)$. Although the algorithm itself is essentially the same as in [BZ16], the analysis is more involved. We conclude the paper with a few remarks on further problems.

- **Lower bounds.** We conjecture the lower bound for this problem (i.e., $\mathscr{Q}_{n,k,\delta}$) is $\tilde{\Omega}(k^2)$, since $\Theta(k^2)$ is the distortion of the CGK random walk embedding [CGK16]. However, to the best of our knowledge, there is no lower bound beyond $\tilde{\Omega}(k)$. (Since we do not find any paper formally stating the lower bounds, we present them in Appendix B.)

- **Edit distance.** It is natural to wonder if current framework can be pushed further. For example, is it possible that we only run $\tau = O(1)$ rounds of CGK random walks and there will be an optimal matching going through all edges that are common to these walks? Unfortunately this is not true, and we can show $\tau = \Omega(\sqrt{k})$ with the following example:

$$
\begin{aligned}
x &= Ac_1c_2\cdots c_{k-1}Bc_1c_2\cdots c_{k-1}\underbrace{d\cdots d}_{2k}Ac_1c_2\cdots c_{k-1},\\
y &= Bc_1c_2\cdots c_{k-1}\underbrace{d\cdots d}_{2k}Ac_1c_2\cdots c_{k-1}Bc_1c_2\cdots c_{k-1}.
\end{aligned}
$$

  Then with probability $1-\Theta(1/\sqrt{k})$, a CGK random walk walks through (k,k). Note that $\mathsf{ed}(x,y)\le 2\cdot k$ by deleting $x[1..k]$ and inserting $y[4k+1..5k]$. However any edit sequence leaving (k,k) matched will have at least $(2\cdot k+1)$ edits, where the one more edit comes from substituting $x[1]$ with $y[1]$. Moreover, this example may generalize to the binary alphabet by replacing each symbol with a short random binary string.

- **Ulam distance.** The Ulam distance is the edit distance on two permutations, i.e., $x\in[n]^n$ (resp., $y\in[n]^n$) and $x_i\ne x_j$ (resp., $y_i\ne y_j$) for distinct i,j. Our algorithm (as well as the algorithm in [BZ16]) works for Ulam distance with an improved bound $\tilde{O}(k^{2.5})$. This comes from the following observation: there is no matched edge in the stable zone, hence the length of stable zone is at most k, which means we can set $\rho = O(\sqrt{L})$ in Proposition 3.9. It would be interesting to improve the algorithm for Ulam distance.

- **Only the distance.** Though our algorithm computes edit distance as well as an optimal edit sequence, it is reasonable to relax the problem by simply asking for the distance or even a constant approximation of the distance. However, we are not aware of any result achieving better sketch size in this setting.

Acknowledgements

We thank Qin Zhang for answering several questions about [BZ16]. C. J. thanks Virginia Vassilevska Williams for several helpful discussions.

References

[AK12] Alexandr Andoni and Robert Krauthgamer. The smoothed complexity of edit distance. *ACM Trans. Algorithms*, 8(4):44:1–44:25, 2012. URL: https://doi.org/10.1145/2344422.2344434, doi:10.1145/2344422.2344434. 1

[AKO10] Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Polylogarithmic approximation for edit distance and the asymmetric query complexity. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 377–386, 2010. URL: https://doi.org/10.1109/FOCS.2010.43, doi:10.1109/FOCS.2010.43. 1

[AN20] Alexandr Andoni and Negev Shekel Nosatzki. Edit distance in near-linear time: it's a constant factor. *CoRR*, abs/2005.07678, 2020. To appear in FOCS 2020. URL: https://arxiv.org/abs/2005.07678, arXiv:2005.07678. 1

[AO12] Alexandr Andoni and Krzysztof Onak. Approximating edit distance in near-linear time. *SIAM J. Comput.*, 41(6):1635–1648, 2012. URL: https://doi.org/10.1137/090767182, doi:10.1137/090767182. 1

[BCR20] Joshua Brakensiek, Moses Charikar, and Aviad Rubinstein. A simple sublinear algorithm for gap edit distance. *CoRR*, abs/2007.14368, 2020. URL: https://arxiv.org/abs/2007.14368, arXiv:2007.14368. 1

[BEG+18] Mahdi Boroujeni, Soheil Ehsani, Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, and Saeed Seddighin. Approximating edit distance in truly subquadratic time: Quantum and MapReduce. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1170–1189, 2018. URL: https://doi.org/10.1137/1.9781611975031.76, doi:10.1137/1.9781611975031.76. 1

[BI18] Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). *SIAM J. Comput.*, 47(3):1087–1097, 2018. URL: https://doi.org/10.1137/15M1053128, doi:10.1137/15M1053128. 1

[BJKK04] Ziv Bar-Yossef, T. S. Jayram, Robert Krauthgamer, and Ravi Kumar. Approximating edit distance efficiently. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 550–559, 2004. URL: https://doi.org/10.1109/FOCS.2004.14, doi:10.1109/FOCS.2004.14. 1

[BR20] Joshua Brakensiek and Aviad Rubinstein. Constant-factor approximation of near-linear edit distance in near-linear time. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 685–698, 2020. URL: https://doi.org/10.1145/3357713.3384282, doi:10.1145/3357713.3384282. 1

[BSS20] Mahdi Boroujeni, Masoud Seddighin, and Saeed Seddighin. Improved algorithms for edit distance and LCS: beyond worst case. In *Proceedings of the 31st ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pages 1601–1620. SIAM, 2020. URL: https://doi.org/10.1137/1.9781611975994.99, doi:10.1137/1.9781611975994.99. 1

[BZ16] Djamal Belazzougui and Qin Zhang. Edit distance: Sketching, streaming, and document exchange. In *Proceedings of the 57th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 51–60. IEEE Computer Society, 2016. URL: https://doi.org/10.1109/FOCS.2016.15, doi:10.1109/FOCS.2016.15. 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 29

[CDG+18] Diptarka Chakraborty, Debarati Das, Elazar Goldenberg, Michal Koucký, and Michael E. Saks. Approximating edit distance within constant factor in truly sub-quadratic time. In *Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 979–990. IEEE Computer Society, 2018. URL: https://doi.org/10.1109/FOCS.2018.00096, doi:10.1109/FOCS.2018.00096. 1

[CGK16] Diptarka Chakraborty, Elazar Goldenberg, and Michal Koucký. Streaming algorithms for embedding and computing edit distance in the low distance regime. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 712–725. ACM, 2016. URL: https://doi.org/10.1145/2897518.2897577, doi:10.1145/2897518.2897577. 1, 2, 5, 6, 29, 32

[CJLW18] Kuan Cheng, Zhengzhong Jin, Xin Li, and Ke Wu. Deterministic document exchange protocols, and almost optimal binary codes for edit errors. In *Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 200–211, 2018. URL: https://doi.org/10.1109/FOCS.2018.00028, doi:10.1109/FOCS.2018.00028. 1

[CK06] Moses Charikar and Robert Krauthgamer. Embedding the ulam metric into ℓ_1. *Theory Comput.*, 2(11):207–224, 2006. URL: https://doi.org/10.4086/toc.2006.v002a011, doi:10.4086/toc.2006.v002a011. 1

[CL20] Kuan Cheng and Xin Li. Efficient document exchange and error correcting codes with asymmetric information. *CoRR*, abs/2007.00870, 2020. To appear in SODA 2021. URL: https://arxiv.org/abs/2007.00870, arXiv:2007.00870. 1

[GKS19] Elazar Goldenberg, Robert Krauthgamer, and Barna Saha. Sublinear algorithms for gap edit distance. In *Proceedings of the 60th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1101–1120, 2019. URL: https://doi.org/10.1109/FOCS.2019.00070, doi:10.1109/FOCS.2019.00070. 1

[GS01] Geoffrey R. Grimmett and David R. Stirzaker. *Probability and random processes*. Oxford University Press, third edition, 2001. 34

[Hae19] Bernhard Haeupler. Optimal document exchange and new codes for insertions and deletions. In *Proceedings of the 60th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 334–347, 2019. URL: https://doi.org/10.1109/FOCS.2019.00029, doi:10.1109/FOCS.2019.00029. 1

[HSZZ06] Wei Huang, Yaoyun Shi, Shengyu Zhang, and Yufan Zhu. The communication complexity of the hamming distance problem. *Inf. Process. Lett.*, 99(4):149–153, 2006. URL: https://doi.org/10.1016/j.ipl.2006.01.014, doi:10.1016/j.ipl.2006.01.014. 33

[KN06] Subhash Khot and Assaf Naor. Nonembeddability theorems via Fourier analysis. *Mathematische Annalen*, 334:821–852, 2006. 1

[KNP+17] Michael Kapralov, Jelani Nelson, Jakub Pachocki, Zhengyu Wang, David P. Woodruff, and Mobin Yahyazadeh. Optimal lower bounds for universal relation, and for samplers and finding duplicates in streams. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 475–486. IEEE Computer Society, 2017. URL: https://doi.org/10.1109/FOCS.2017.50, doi:10.1109/FOCS.2017.50. 13

[KR87] Richard M. Karp and Michael O. Rabin. Efficient randomized pattern-matching algorithms. *IBM J. Res. Dev.*, 31(2):249–260, 1987. URL: https://doi.org/10.1147/rd.312.0249, doi:10.1147/rd.312.0249. 12

[KR09] Robert Krauthgamer and Yuval Rabani. Improved lower bounds for embeddings into l_1. *SIAM J. Comput.*, 38(6):2487–2498, 2009. URL: https://doi.org/10.1137/060660126, doi:10.1137/060660126. 1

[KS20a] Tomasz Kociumaka and Barna Saha. Sublinear-time algorithms for computing & embedding gap edit distance. *CoRR*, abs/2007.12762, 2020. To appear in FOCS 2020. URL: https://arxiv.org/abs/2007.12762, arXiv:2007.12762. 1

[KS20b] Michal Koucký and Michael E. Saks. Constant factor approximations to edit distance on far input pairs in nearly linear time. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 699–712. ACM, 2020. URL: https://doi.org/10.1145/3357713.3384307, doi:10.1145/3357713.3384307. 1

[LPW09] David A Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, first edition, 2009. 6, 32

[MP80] William J. Masek and Mike Paterson. A faster algorithm computing string edit distances. *J. Comput. Syst. Sci.*, 20(1):18–31, 1980. URL: https://doi.org/10.1016/0022-0000(80)90002-1, doi:10.1016/0022-0000(80)90002-1. 1

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Comb.*, 12(4):449–461, 1992. URL: https://doi.org/10.1007/BF01305237, doi:10.1007/BF01305237. 6

[OR07] Rafail Ostrovsky and Yuval Rabani. Low distortion embeddings for edit distance. *J. ACM*, 54(5):23, 2007. URL: https://doi.org/10.1145/1284320.1284322, doi:10.1145/1284320.1284322. 1

[Orl91] Alon Orlitsky. Interactive communication: Balanced distributions, correlated files, and average-case complexity. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 228–238, 1991. URL: https://doi.org/10.1109/SFCS.1991.185373, doi:10.1109/SFCS.1991.185373. 1

A Simpler Analysis of CGK

We first restate Item (3) in Theorem 2.9 here and set the number of random walk steps to infinity.

Theorem A.1 ([CGK16, Theorem 4.1])**.** *Let λ be an ∞-step random walk on x, y, where p, q are the pointers on x, y. Then*

$$\mathbf{Pr}\left[\#\textit{progress steps in } \lambda \geq (T \cdot \mathsf{ed}(x,y))^2\right] \leq O\left(\frac{1}{T}\right).$$

Proof. Let z be the longest common subsequence of x, y. Hence x (and y) can be obtained from z by at most $\mathsf{ed}(x,y)$ insertions. We perform a CGK random walk on z with pointer w using the same randomness. Let $\lambda_{x,z}$ (resp., $\lambda_{y,z}$) be view of λ on x, z (resp., y, z). Since $\lambda_{x,z}$ and $\lambda_{y,z}$ are projections of λ, by triangle inequality it suffices to prove

$$\mathbf{Pr}\left[\#\text{progress steps in } \lambda_{x,z} \geq (T \cdot \mathsf{ed}(x,y))^2\right] \leq O\left(\frac{1}{T}\right), \tag{18}$$

$$\mathbf{Pr}\left[\#\text{progress steps in } \lambda_{y,z} \geq (T \cdot \mathsf{ed}(x,y))^2\right] \leq O\left(\frac{1}{T}\right). \tag{19}$$

Now we prove (18), and the proof of (19) is analogous. We will find the following classical result useful.

Fact A.2 (e.g. [LPW09, Theorem 2.17])**.** *Let k be some non-negative integer. Let π be a one-dimensional unbiased and self-looped random walk (See Definition 2.10) starting from 0. Let T_0 be the first time π hits k. Then*

$$\mathbf{Pr}\left[T_0 > (M \cdot k)^2\right] \leq O\left(\frac{1}{M}\right).$$

Let $k_{x,z}$ be the number of insertions required to get x from z. Observe that if p is between the i-th insertion and the $(i+1)$-th insertion from z, we have $p - w \leq i$; and when the equality holds the random walk will not have progress steps before p arrives at the $(i+1)$-th insertion. Hence, we can safely truncate $\lambda_{x,z}$ at time t for which we have $p_t = w_t + k_{x,z}$. Though t is not necessarily bounded since no progress step occurs outside x, z, we can conceptually keep adding progress steps after p, w are outside x, z until $p_t = w_t + k_{x,z}$, which will only increase the count. Considering the correspondence between progress steps and transitions in a one-dimensional unbiased and self-looped random walk (see Remark 2.11), by Fact A.2 we have

$$\begin{aligned}
&\mathbf{Pr}\left[\#\text{progress steps in } \lambda_{x,z} \geq (T \cdot \mathsf{ed}(x,y))^2\right] \\
\leq\, &\mathbf{Pr}\left[\#\text{progress steps in } \lambda_{x,z} \geq (T \cdot k_{x,z})^2\right] \\
\leq\, &\mathbf{Pr}\left[\#\text{transitions before } p_t - w_t = k_{x,z} \text{ is at least } (T \cdot k_{x,z})^2\right] \\
\leq\, &O\left(\frac{1}{T}\right).
\end{aligned}$$

□

B Lower Bounds

In this section we will prove the following lower bounds. Though they are just simple counting arguments and reductions, yet we can't find any paper explicitly stating them. Hence we decide to include the proof here.

Theorem B.1. *The sketch size is $\Omega(k\log(n|\Sigma|/k))$ bits if we want to compute edit distance and an optimal edit sequence with probability at least $2/3$.*

Proof. Let sy be the sketch of y and R be the maximum number of random bits used. Let S be the set of triple (r,x,y) where $r\in\{0,1\}^R$ is the randomness, x,y are the inputs of length n with $\mathsf{ed}(x,y)\le k$, and we succeed in computing an optimal edit sequence. By the assumption, for any fixed x,y there are at least $2/3$ fraction of r such that $(r,x,y)\in S$. Hence

$$|S|\ge \underbrace{\frac{2}{3}\cdot 2^R}_{\text{number of } r}\cdot\underbrace{|\Sigma|^n}_{\text{number of } x}\cdot\underbrace{\binom{n}{k}\cdot(|\Sigma|-1)^k}_{\text{number of } y}.$$

On the other hand, when $(r,x,y)\in S$, we can recover y using x and the edit sequence. Hence we have an injection from S to $\{0,1\}^R\times\Sigma^n\times\{0,1\}^{|sy|}$, which means

$$|S|\le 2^R\cdot|\Sigma|^n\cdot 2^{|sy|}.$$

By rearranging terms, we have $|sy|=\Omega(k\log(n|\Sigma|/k))$. □

Theorem B.2. *Assume $k\le O(\sqrt{n}),|\Sigma|\ge 2$ or $k\le O(n),|\Sigma|\ge 2n$. The sketch size is $\Omega(k)$ bits if we only want to compute edit distance with probability at least $2/3$.*

Proof. We show two reductions from the following theorem.

Theorem B.3 ([HSZZ06])**.** *Let $1\le d\le 3N/8$ be a parameter and $K=2d/3$. Assume Alice gets $X\in\{0,1\}^N$ and Bob gets $Y\in\{0,1\}^N$, and both X and Y have exactly K ones. Their goal is to distinguish with probability $2/3$ whether $\mathsf{Ham}(X,Y)\le d$, where $\mathsf{Ham}(\cdot,\cdot)$ is the Hamming distance. Then the number of communication bits is $\Omega(d)$.*

Now we present the reduction.

- **Case $k\le O(\sqrt{n}),|\Sigma|\ge 2$.**

  Construct x from X by replacing 0 with $\underbrace{0..0}_{n}1\underbrace{0..0}_{2n-1}\underbrace{1..1}_{3n}$ and replacing 1 with $\underbrace{0..0}_{2n-1}1\underbrace{0..0}_{n}\underbrace{1..1}_{3n}$. The construction for y is the same. Let $n=6N^2,k=4K$.

  Then $\mathsf{ed}(x,y)=2\cdot\mathsf{Ham}(X,Y)\le 4K$. Therefore we can tell if $\mathsf{Ham}(X,Y)\le d$ when we successfully compute $\mathsf{ed}(x,y)$. Hence by Theorem B.3, the sketch size is $\Omega(d)=\Omega(k)$. Note that $d\le O(N)$, we have $k\le O(\sqrt{n})$.

- **Case $k\le O(n),|\Sigma|\ge 2n$.**

  Construct x from X by replacing the i-th bit of X with $2\cdot i-X_i\in[2N]$. The construction for y is the same. Let $n=N,k=2K$.

  Then $\mathsf{ed}(x,y)=\mathsf{Ham}(X,Y)\le 2K$. Therefore we can tell if $\mathsf{Ham}(X,Y)\le d$ when we successfully computes $\mathsf{ed}(x,y)$. Hence by Theorem B.3, the sketch size is $\Omega(d)=\Omega(k)$. Note that $d\le O(N)$, we have $k\le O(n)$. □

We conjecture the conditions in Theorem B.2 can be relaxed to $k\le O(n),|\Sigma|\ge 2$ with a better reduction.

C Proof of Lemma 2.5

Lemma (Lemma 2.5 restated). *Let x, y be two strings. Let S be an optimal edit sequence and $\mathcal{M}(S)$ be its corresponding optimal matching.*

(1) If $(i, j) \in \mathcal{M}(S)$, then $|i - j| \leq \mathsf{ed}(x, y)$.

(2) If $u' \leq u, v' \leq v$ and $u - u' + 1 = v - v' + 1 =: L$, then the number of matched edges in $x[u'..u], y[v'..v]$ is at least $L - 3 \cdot \mathsf{ed}(x, y) - |u - v|$, i.e.,

$$\left|\mathcal{M}(S) \cap \left([u'..u] \times [v'..v]\right)\right| \geq L - 3 \cdot \mathsf{ed}(x, y) - |u - v|.$$

Proof. Let $k := \mathsf{ed}(x, y)$. We first prove Item (1). Imagine we start with x, y and perform S to make them equal. Let (i_t, j_t) be the edge (i, j) after performing t edits, then $i_k = j_k$. Note that each edit can change the difference of i_t, j_t by at most 1, hence $|i - j| = |i_0 - j_0| \leq k$.

Now we turn to Item (2). Let U be the set of matched edges in $x[u'..u], y[v'..v]$ projected on y, then

$$U = \{j \in [v'..v] \mid (j, i_j) \text{ is a matched edge for some } i_j \text{ and } i_j \in [u'..u]\}.$$

Assume without loss of generality $u \geq v$ (or equivalently $u' \geq v'$). Since $\mathcal{M}(S)$ is non-intersecting, it suffices to prove $|U| \geq L - 3 \cdot k - (u - v)$. By Item (1), for any $u' + k \leq j \leq v - k$, if (j, i_j) is a matched edge, then $i_j \in [u'..u]$. On the other hand, there are at most k characters on y that are not covered by $\mathcal{M}(S)$. Hence

$$|U| \geq ((v - k) - (u' + k) + 1) - k = (u - u' + 1) - 3 \cdot k - (u - v) = L - 3 \cdot k - (u - v). \qquad \square$$

D Proof of Lemma 2.13

We will use the following versions of optional stopping theorem and Borel-Cantelli theorem.

Theorem D.1 (Optional stopping theorem, e.g. [GS01, Section 12.5]). *Let $X = (X_t)_{t \in \mathbb{N}}$ be a discrete-time martingale and T is a stopping time with values in $\mathbb{N}$. If T is almost surely bounded, then $\mathbb{E}[X_T] = \mathbb{E}[X_0]$.*

Theorem D.2 (Borel-Cantelli theorem, e.g. [GS01, Section 7.3]). *Let T be a non-negative random variable. If*

$$\sum_{i=0}^{+\infty} \mathbf{Pr}[T > i] < +\infty,$$

then T is almost surely bounded.

Now we prove Lemma 2.13.

Lemma (Lemma 2.13 restated). *Consider an ∞-step CGK random walk λ on x, y, where p, q are the pointers on x, y respectively. Let u be an index and let $U, V \geq u - 1$ be any integers. Then the following hold.*

(1) Let T_0 be the first time that $p_{T_0} \geq u$. Then $\mathbb{E}[|p_{T_0} - q_{T_0}|] \leq 4 \cdot \mathsf{ed}(x[1..U], y[1..V])$.

(2) Let T_1 be the first time that $(p_{T_1} \geq u) \wedge (q_{T_1} \geq u)$. Then $\mathbb{E}[|p_{T_1} - q_{T_1}|] \leq 4 \cdot \mathsf{ed}(x[1..U], y[1..V])$.

Proof. Let T be the first time that $(p_T \geq u) \vee (q_T \geq u)$. We first verify T, T_0, T_1 are almost surely bounded. By Chernoff's bound, we know for any $i \geq 6 \cdot u$,

$$\mathbf{Pr}[T > i], \ \mathbf{Pr}[T_0 > i], \ \mathbf{Pr}[T_1 > i] \leq e^{-\Omega(i)}.$$

Hence

$$\sum_{i=0}^{+\infty} \mathbf{Pr}[T > i] \leq 6 \cdot u + \sum_{i=6 \cdot u}^{+\infty} e^{-\Omega(i)} < +\infty.$$

The same calculation holds for T_0, T_1. Therefore, by Theorem D.2 they are almost surely bounded.

Let $k = \mathsf{ed}(x[1..U], y[1..V])$ and let z be the longest common subsequence of $x[1..U], y[1..V]$. Hence $x[1..U]$ (and $y[1..V]$) can be obtained from z by at most k insertions. In particular, the length of z is at least $U - k \geq u - k - 1$.

We perform a CGK random walk on z with pointer w using the same randomness.

Claim D.3. $\mathbb{E}\left[|p_T - q_T|\right] \leq 4 \cdot k$.

Proof. By Theorem D.1, $\mathbb{E}\left[p_T - w_T\right] = 0$. Observe that $p_T \leq u$ and $p_{T-1} \leq u - 1$, hence $p_T - w_T \leq k$ and

$$\mathbb{E}\left[|p_T - w_T|\right] = \mathbb{E}\left[|p_T - w_T| + (p_T - w_T)\right] = 2 \cdot \mathbb{E}\left[\max\left\{p_T - w_T, 0\right\}\right] \leq 2 \cdot k.$$

Similarly $\mathbb{E}\left[|q_T - w_T|\right] \leq 2 \cdot k$. Hence

$$\mathbb{E}\left[|p_T - q_T|\right] \leq \mathbb{E}\left[|p_T - w_T|\right] + \mathbb{E}\left[|q_T - w_T|\right] \leq 4 \cdot k. \qquad \square$$

We now prove Item (1). Observe that if $p_T \geq u$, then $T_0 = T$. Otherwise $p_T < u = q_T$ and $T_0 > T$, which means $q_{T_0} \geq p_{T_0}$. By Theorem D.1,

$$\mathbb{E}\left[|p_{T_0} - q_{T_0}| \,\middle|\, p_T, q_T\right] = \mathbb{E}\left[q_{T_0} - p_{T_0} \,\middle|\, p_T, q_T\right] = q_T - p_T = |p_T - q_T|.$$

Hence, $\mathbb{E}\left[|p_{T_0} - q_{T_0}|\right] = \mathbb{E}\left[|p_T - q_T|\right] \leq 4 \cdot k$.

Let T_0' be the first time that $q_{T_0'} \geq u$. Then by symmetry $|p_{T_0'} - q_{T_0'}|$ shares the same bound as $|p_{T_0} - q_{T_0}|$. Hence Item (2) follows from $T_1 = \max\left\{T_0, T_0'\right\}$. $\square$