The Communication Complexity of Pattern Matching with Edits Revisited

Tomasz Kociumaka
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Jakob Nogler
Massachusetts Institute of Technology, Cambridge, MA, USA

Philip Wellnitz
National Institute of Informatics, Tokyo, Japan
The Graduate University for Advanced Studies, SOKENDAI, Tokyo, Japan

Abstract

The decades-old *Pattern Matching with Edits* problem, given a length-n string T (the text), a length-m string P (the pattern), and a positive integer k (the threshold), asks to list the k-error occurrences of P in T, that is, all fragments of T whose edit distance to P is at most k. The one-way communication complexity of this problem is the minimum number of bits that Alice, given an instance (P, T, k) of the problem, must send to Bob so that Bob can reconstruct the answer solely from that message.

In recent work [STOC'24], we showed that, in the natural parameter regime $0 < k < m < n/2$, $\Omega(n/m \cdot k \log(m/k))$ bits are necessary and $\mathcal{O}(n/m \cdot k \log^2 m)$ bits are sufficient for this problem. More generally, for strings over an alphabet Σ, we gave an $\mathcal{O}(n/m \cdot k \log m \log(m|\Sigma|))$-bit encoding that allows one to recover a shortest sequence of edits for every k-error occurrence of P in T.

In this paper, we revisit the original proof and improve the encoding size to $\mathcal{O}(n/m \cdot k \log(m|\Sigma|/k))$, which matches the lower bound for constant-sized alphabets. We further establish a new tight lower bound of $\Omega(n/m \cdot k \log(m|\Sigma|/k))$ for the edit sequence reporting variant we solve. Our encoding size also matches the communication complexity established for the simpler Pattern Matching with Mismatches problem in the context of streaming algorithms [Clifford, Kociumaka, Porat; SODA'19].

2012 ACM Subject Classification Theory of computation → Pattern matching; Theory of computation → Communication complexity; Mathematics of computing → Combinatorics on words

Keywords and phrases Edit distance, Pattern matching, Communication complexity

Digital Object Identifier 10.4230/LIPIcs.CPM.2026.26

Related Version *Full Version*: `https://arxiv.org/abs/2604.15601` [16]

1 Introduction

In the more-than-classic *Pattern Matching with Edits* problem (PMWE) [22], we are given a text T of length n, a pattern P of length m, and a threshold k, and the task is to compute the starting positions of all substrings of T whose edit distance to P is at most k. Formally, we aim to compute the set

$$\mathrm{Occ}_k^E(P,T) \coloneqq \{i \in [0\,..\,n] : \exists_{j \in [i\,..\,n]}\, \delta_E(P, T[i\,..\,j)) \le k\},$$

where $\delta_E(\star,\star)$ denotes the edit (Levenshtein) distance [21]. This metric quantifies the (dis)similarity between strings by counting the minimum number of single-character edits (insertions, deletions, and substitutions) required to transform one string into the other.

Research into this problem has expanded far beyond the classical setting [22, 19, 20, 9, 5, 6], with modern variants exploring compressed [11, 24, 3, 5], dynamic [5], streaming [23, 17, 2], weighted [7], and quantum [13, 15] settings. We recently investigated the problem's one-way communication complexity [13], which involves determining the minimum space required to encode an instance so that the set of k-error occurrences can be reconstructed without further access to the original input.

The *communication complexity* framework models this as a two-party game: Alice, who holds the problem instance, transmits a single message to Bob. Bob's task is to derive the full output solely from this message. Because Bob lacks any initial input of his own, the problem reduces to a one-way single-round protocol, and the primary objective is to minimize the number of bits Alice must transmit.

In [13], we established that, in the natural parameter regime of $0 < k < m < n/2$, $\Omega(n/m \cdot k \log(m/k))$ bits are necessary for this problem, while $\mathcal{O}(n/m \cdot k \log^2 m)$ bits are sufficient. More generally, we showed that if the shortest sequence of edits must be recovered for each k-error occurrence, the upper bound increases to $\mathcal{O}(n/m \cdot k \log m \log(m|\Sigma|))$, where Σ is the input alphabet. The former result is implied by the latter by performing a simple alphabet reduction: all characters of T not present in P are mapped to a single character. The result of [13] is significant for three reasons.

1. A central open question [9, 6] for PMWE is whether a static algorithm can match the $\mathcal{O}(n + n/m \cdot k^{2-\Omega(1)})$ conditional lower bound [1]. The current state-of-the-art algorithm by Charalampopoulos, Kociumaka, and Wellnitz [6] runs in $\tilde{\mathcal{O}}(n + n/m \cdot k^{3.5})$ time, and relies on structural results [5] that allow it to return the required output $\mathrm{Occ}_k^E(P, T)$ using $\tilde{\mathcal{O}}(n/m \cdot k^3)$ bits. This encoding is fundamentally incompatible with reaching the lower bound; however, the results of [13] demonstrate that this barrier can indeed be overcome.
2. A similar question was previously settled for the simpler Pattern Matching with Mismatches problem (PMWM) [8]; here $\Omega(n/m \cdot k \log(m/k))$ bits are necessary and $\mathcal{O}(n/m \cdot k \log(m|\Sigma|/k))$ bits suffice to also report the mismatches for all k-mismatch occurrences. Thus, by [13] and up to essentially an $\mathcal{O}(\log m)$ factor, the edit distance setting behaves the same as the mismatch setting in terms of information density.
3. Said result of [13] also paves the way for better algorithms in other computational models (e.g., the quantum algorithms for PMWE of [13, 15], which crucially rely on this result). The prior work on PMWM [8] suggests that one-way communication complexity is a stepping stone toward $\tilde{\mathcal{O}}(k)$-space algorithms in the streaming and semi-streaming models. This is particularly relevant given the concurrent development of $\mathcal{O}(k \cdot n^{o(1)})$-size edit distance sketches [18]. Although the state-of-the-art sketches do not yet offer the space-efficient sketch construction needed for a full streaming implementation, the advances in [13] represent significant steps toward it.

Our Results. We re-examine the encoding provided in [13] and, through a series of refinements, eliminate the $\Theta(\log m)$ term separating the upper and lower bounds. Thereby, we place PMWE on par with PMWM. Formally, we show the following result.

▶ **Theorem 1.1.** *Fix integers n, m, k with $n \geq m \geq k > 0$ and an input alphabet Σ. The Pattern Matching with Edits problem admits a one-way deterministic communication protocol that sends $\mathcal{O}(n/m \cdot k \log(m|\Sigma|/k))$ bits. Within the same communication complexity, one can also reconstruct the set of all fragments $T[\,i\,..\,j\,)$ satisfying $\delta_E(P, T[\,i\,..\,j\,)) \leq k$ and, for each such fragment, the edit information (the positions and values of edited characters) of all optimal edit sequences transforming P into the fragment $T[\,i\,..\,j\,)$.*

▶ Remark 1.2. Our result extends routinely beyond $n \geq m \geq k > 0$. If $n < m$, it suffices to apply Theorem 1.1 with the text T replaced by $\mathtt{0}^{m-n} \cdot T$, where $\mathtt{0} \in \Sigma$ is an arbitrary character. Applying Theorem 1.1 with threshold $k = m$ allows one to reconstruct the entire pattern from the edit sequence for an empty fragment of the text and the entire text from the edit sequences for single-character fragments; this covers $k > m$. Finally, the case $k = 0$ is no harder than $k = 1$. Thus, in full generality, the communication complexity becomes

$$\mathcal{O}\left(\max\left(1, \tfrac{n}{m}\right) \cdot \min(m, k+1) \cdot \log \tfrac{m|\Sigma|}{\min(m,k+1)}\right).$$

Through the aforementioned alphabet reduction, we also achieve $\mathcal{O}(n/m \cdot k \log m)$ bits for the variant where only the positions (but not the values) of edited characters are reported. This suffices for the baseline problem asking for the starting positions of k-error occurrences, which means that we match the lower bound of [13] unless both $k \geq m^{1-o(1)}$ and $|\Sigma| \geq \omega(1)$.

It is not difficult to show that a single optimal edit sequence for a single k-error occurrence can be encoded using $\mathcal{O}(k \log(m|\Sigma|/k))$ bits. While the previous approach in [13] required storing information proportional to $\mathcal{O}(\log m)$ edit sequences for each length-$\Theta(m)$ block of the text, our improvement achieves the information content equivalent to $\mathcal{O}(1)$ edit sequences. Hence, in a $\Theta(m)$-length block of T, the cost of encoding all k-error occurrences and their respective edit sequences is asymptotically no greater than the cost of encoding one!

We also establish a new lower bound proving that our protocol is optimal for edit retrieval.

▶ **Theorem 1.3.** *Fix integers n, m, k with $n \geq m \geq k > 0$ and an input alphabet Σ with $\mathtt{0} \in \Sigma$ and $|\Sigma| \geq 2$. Consider an encoding that, for each fragment $T[\,i\,..\,j\,)$ with $\delta_E(P, T[\,i\,..\,j\,)) \leq k$, allows to reconstruct an edit sequence $P \rightsquigarrow T[\,i\,..\,j\,)$ and the corresponding edit information. Such an encoding must use $\Omega(n/m \cdot k \log(|\Sigma|m/k))$ bits for $P = \mathtt{0}^m$ and some $T \in \Sigma^n$.*

Our Techniques. On a very high level, in [13], the text T is first partitioned into $\mathcal{O}(n/m)$ partially overlapping blocks, each of length $\mathcal{O}(m)$ and such that each k-error occurrence of P in T is fully contained in some block. For each such block, the encoding contains:

1. a set S of $\mathcal{O}(\log m)$ k-error occurrences of P in T, each stored together with an optimal edit sequence using $\mathcal{O}(k \log(m|\Sigma|/k))$ bits;
2. the Lempel–Ziv LZ77 [25] compressed representation of a collection of fragments of T selected based on S. The total number of LZ77 phrases is proportional to the total cost of k-error occurrences in S, so they can be encoded using $\mathcal{O}(k \log m \log(m|\Sigma|/k))$ bits.

Overall, this yields an encoding cost of $\mathcal{O}(k \log m \log(m|\Sigma|/k))$ bits per block and $\mathcal{O}(n/m \cdot k \log m \log(m|\Sigma|/k))$ bits for the entire text. The underlying construction is relatively complex, relying on recent insights relating edit distance to compressibility [4, 12] via the so-called *self-edit distance*. Nevertheless, our key improvement is simple to describe at a high level: In (1), we partition the i-th k-error occurrence into $\Theta(2^i)$ pieces and store the edit information only for the "cheapest" piece, incurring only $k_i = \mathcal{O}(k/2^i)$ edits, for a total of $\mathcal{O}(k)$ instead of $\mathcal{O}(k \log m)$. The fragments in (2) are defined essentially the same way and, thanks to the decreased total cost in (1), they can be encoded in $\mathcal{O}(k \log(m|\Sigma|/k))$ bits.

Generalizing the relevant concepts to capture pieces of P requires several delicate steps, which we discuss in Section 3. Moreover, our approach breaks for $k = o(\log m)$ since 0-error pieces still need a $\Theta(\log m)$-bit representation, which becomes a bottleneck when $|S| \geq \omega(k)$. In that case, through the structural characterization of Charalampopoulos, Kociumaka, and Wellnitz [5], the existence of $\omega(k)$ occurrences implies that P and the relevant part of T are at edit distance $\mathcal{O}(k)$ from highly periodic strings. We then use this rigid structure to show that just three $\mathcal{O}(k)$-error occurrences of P in T can play the role of S in (1).

2 Preliminaries

Sets. For integers $i, j \in \mathbb{Z}$, we write $[\,i\,..\,j\,]$ to denote the set $\{i, \ldots, j\}$ and $[\,i\,..\,j\,)$ to denote the set $\{i, \ldots, j-1\}$; we define the sets $(\,i\,..\,j\,]$ and $(\,i\,..\,j\,)$ similarly.

For a set S of integers and a parameter $k > 0$, we also define $\lfloor S/k \rfloor \coloneqq \{\lfloor s/k \rfloor : s \in S\}$.

Strings. An *alphabet* Σ is a set of characters. We write $X = X[\,0\,]\,X[\,1\,] \cdots X[\,n-1\,] \in \Sigma^n$ to denote a *string* of length $|X| = n$ over Σ. For a *position* $i \in [\,0\,..\,n\,)$, we say that $X[\,i\,]$ is the i-th character of X. A string Y is a *substring* of another string X if $Y = X[\,i\,] \cdots X[\,j-1\,]$

holds for some integers $0 \le i \le j \le |X|$. In this case, we say that there is an (*exact*) *occurrence* of Y starting at position i in X. The occurrence is a *fragment* of X denoted $X[\,i\,..\,j\,)$; formally, the fragment can be interpreted as a tuple (X, i, j) consisting of a (reference to) X and the two positions i and j. We may also write $X[\,i\,..\,j-1\,]$, $X(\,i-1\,..\,j-1\,]$, or $X(\,i-1\,..\,j\,)$ for the fragment $X[\,i\,..\,j\,)$. A *prefix* of a string X is a fragment of the form $X[\,0\,..\,j\,)$, and a *suffix* of a string X is a fragment of the form $X[\,i\,..\,|X|\,)$.

For two strings A and B, we write AB for their concatenation. We write A^k for the concatenation of $k \in \mathbb{Z}_{\ge 0}$ copies of the string A. Moreover, A^∞ is an infinite string (indexed with non-negative integers) formed as the concatenation of an infinite number of copies of A.

An integer $p \in [\,1\,..\,n\,]$ is *a period* of a string $X \in \Sigma^n$ if we have $X[\,i\,] = X[\,i+p\,]$ for all $i \in [\,0\,..\,n-p\,)$. In this case, we also say that the string $X[\,0\,..\,p\,)$ is a *string period* of X. In other words, a string P is a string period of a string X if X is a prefix of P^∞ and $|P| \le |X|$. *The period* of a non-empty string X, denoted $\mathrm{per}(X)$, is the smallest period of X. A non-empty string X is *periodic* if $\mathrm{per}(X) \le |X|/2$.

Periodicity Lemma. For completeness, we restate Fine and Wilf's Periodicity Lemma [10].

▶ **Lemma 2.1** (Periodicity Lemma [10]). *If p, q are periods of a string X of length $|X| \ge p + q - \gcd(p,q)$, then $\gcd(p,q)$ is a period of X.*

Edit Distance and Alignments. The *edit distance* (*Levenshtein distance* [21]) between two strings X and Y, denoted by $\delta_E(X,Y)$, is the minimum number of character insertions, deletions, and substitutions required to transform X into Y. Formally, we first define an *alignment* between string fragments.

▶ **Definition 2.2** ([6, Definition 2.1]). *A sequence $\mathcal{A} = (x_i, y_i)_{i=0}^{\ell}$ is an* alignment *of $X[\,x\,..\,x'\,)$ onto $Y[\,y\,..\,y'\,)$, denoted by $\mathcal{A} : X[\,x\,..\,x'\,) \rightsquigarrow\!\!\!\!\rightarrow Y[\,y\,..\,y'\,)$, if it satisfies $(x_0, y_0) = (x, y)$, $(x_{i+1}, y_{i+1}) \in \{(x_i+1, y_i+1), (x_i+1, y_i), (x_i, y_i+1)\}$ for $i \in [\,0\,..\,\ell\,)$, and $(x_\ell, y_\ell) = (x', y')$. Moreover, for $i \in [\,0\,..\,\ell\,)$:*

- *If $(x_{i+1}, y_{i+1}) = (x_i+1, y_i)$, we say that $\mathcal{A}$* deletes *$X[\,x_i\,]$.*
- *If $(x_{i+1}, y_{i+1}) = (x_i, y_i+1)$, we say that $\mathcal{A}$* inserts *$Y[\,y_i\,]$.*
- *If $(x_{i+1}, y_{i+1}) = (x_i+1, y_i+1)$, we say that $\mathcal{A}$* aligns *$X[\,x_i\,]$ to $Y[\,y_i\,]$. If also $X[\,x_i\,] = Y[\,y_i\,]$, then $\mathcal{A}$* matches *$X[\,x_i\,]$ and $Y[\,y_i\,]$; otherwise, $\mathcal{A}$* substitutes *$X[\,x_i\,]$ with $Y[\,y_i\,]$.*

The *cost* of an alignment $\mathcal{A}$ of $X[\,x\,..\,x'\,)$ onto $Y[\,y\,..\,y'\,)$, denoted by $\mathrm{cost}(\mathcal{A})$ or $\delta_E^{\mathcal{A}}(X[\,x\,..\,x'\,), Y[\,y\,..\,y'\,))$, is the total number of characters that $\mathcal{A}$ inserts, deletes, or substitutes. The edit distance $\delta_E(X,Y)$ is the minimum cost of an alignment of $X[\,0\,..\,|X|\,)$ onto $Y[\,0\,..\,|Y|\,)$. An alignment of X onto Y is *optimal* if its cost is equal to $\delta_E(X,Y)$.

Given an alignment $\mathcal{A} : X[\,x\,..\,x'\,) \rightsquigarrow\!\!\!\!\rightarrow Y[\,y\,..\,y'\,)$ and a fragment $X[\,\bar{x}\,..\,\bar{x}'\,)$ contained in $X[\,x\,..\,x'\,)$, we write $\mathcal{A}(X[\,\bar{x}\,..\,\bar{x}'\,))$ for the fragment $Y[\,\bar{y}\,..\,\bar{y}'\,)$ of $Y[\,y\,..\,y'\,)$ that $\mathcal{A}$ aligns against $X[\,\bar{x}\,..\,\bar{x}'\,)$. As insertions and deletions may render this definition ambiguous, we set

$$\bar{y} \coloneqq \min\{\hat{y} : (\bar{x}, \hat{y}) \in \mathcal{A}\} \quad \text{and} \quad \bar{y}' \coloneqq \begin{cases} y' & \text{if } \bar{x}' = x', \\ \min\{\hat{y}' : (\bar{x}', \hat{y}') \in \mathcal{A}\} & \text{otherwise.} \end{cases}$$

This particular choice satisfies the following decomposition property.

▶ **Observation 2.3** ([6, Fact 2.2]). *For an alignment $\mathcal{A} : X \rightsquigarrow\!\!\!\!\rightarrow Y$ and a decomposition $X = X_1 \cdots X_t$ into t fragments, $Y = \mathcal{A}(X_1) \cdots \mathcal{A}(X_t)$ is a decomposition into t fragments with $\delta_E^{\mathcal{A}}(X,Y) = \sum_{i=1}^{t} \delta_E^{\mathcal{A}}(X_i, \mathcal{A}(X_i))$. If $\mathcal{A}$ is optimal, then $\delta_E(X,Y) = \sum_{i=1}^{t} \delta_E(X_i, \mathcal{A}(X_i))$.*

We use the following *edit information* notion to encode alignments.

▶ **Definition 2.4** (Edit information of an alignment). *For an alignment* $(x_i, y_i)_{i=0}^{\ell} = \mathcal{A} : X[x..x') \rightsquigarrow Y[y..y')$, *the* edit information *is the set of 4-tuples* $\mathsf{E}_{X,Y}(\mathcal{A}) = \{(x_i, \mathsf{cx}_i : y_i, \mathsf{cy}_i) : i \in [0..\ell)$ *and* $\mathsf{cx}_i \neq \mathsf{cy}_i\}$, *where*

$$\mathsf{cx}_i = \begin{cases} X[x_i] & \text{if } x_{i+1} = x_i + 1, \\ \varepsilon & \text{otherwise;} \end{cases} \quad \text{and} \quad \mathsf{cy}_i = \begin{cases} Y[y_i] & \text{if } y_{i+1} = y_i + 1, \\ \varepsilon & \text{otherwise.} \end{cases}$$

By monotonicity of the alignment, the order of the tuples $(x_i, \mathsf{cx}_i : y_i, \mathsf{cy}_i)$ *with respect to* i *coincides with their lexicographic order by* (x_i, y_i) *and by* (y_i, x_i).

Observe that, given two strings X and Y, the endpoints x, x', y, y', and the edit information $\mathsf{E}_{X,Y}(\mathcal{A})$ of an alignment $\mathcal{A} : X[x..x') \rightsquigarrow Y[y..y')$, we are able to fully reconstruct $\mathcal{A}$. Indeed, each tuple of $\mathsf{E}_{X,Y}(\mathcal{A})$ specifies a non-matching operation, whereas all pairs of $\mathcal{A}$ before the first tuple, between consecutive tuples, and after the last tuple must be matches. Moreover, given the characters of $X[x..x')$ and the edit information $\mathsf{E}_{X,Y}(\mathcal{A})$, one can recover the characters of $Y[y..y')$. Indeed, the tuples of $\mathsf{E}_{X,Y}(\mathcal{A})$ reveal all characters of $Y[y..y')$ created by insertions and substitutions, while the remaining characters of $Y[y..y')$ correspond to matches and can therefore be copied from $X[x..x')$.

Pattern Matching with Edits. In the context of two strings P (referred to as the pattern) and T (referred to as the text), along with a positive integer k (referred to as the threshold), we say that $T[t..t')$ is a *k-error occurrence* of P in T if $\delta_E(P, T[t..t')) \leq k$ holds.

3 Upper Bound: Proof Overview

In this section, we provide an overview of the proof of Theorem 1.1[1]. To this end, we fix two strings $P \in \Sigma^m$ and $T \in \Sigma^n$, and a positive threshold k. Throughout this section, we assume that $k = o(m)$, $n \leq 2m - 2k$, and P has k-error occurrences as a prefix and as a suffix of T. In [13, Claim 4.31[2]], a standard block-splitting argument is employed to demonstrate that a protocol using $\mathcal{O}(k \log m \log(m|\Sigma|))$ bits for this specific case is sufficient to achieve $\mathcal{O}(n/m \cdot k \log m \log(m|\Sigma|))$ bits in general. We improve the former communication bound to $\mathcal{O}(k \log(m|\Sigma|/k))$ bits and a similar proof to [13, Claim 4.31] to drop the assumptions.

The remainder of this section is structured as follows. In Section 3.1, we provide a concise overview of the proof from [13], focusing specifically on the components we adapt to achieve our improvements. Subsequently, in Section 3.2, we outline our improved construction.

3.1 Previous Encoding

3.1.1 The Graph G_S and the Induced Periodic Structure

The main ingredient of the encoding of [13] is a set S of $\mathcal{O}(\log m)$ alignments of P onto fragments of T with cost at most k each. The starting point of [13] is to analyze how much information such a set of alignments carries, in order to better understand how one should choose which alignments to include and what additional information is needed to fully encode $\mathrm{Occ}_k^E(P, T)$. To enable this analysis, the set S is associated with a graph $\mathbf{G}_S$, which we call the *inference graph*.

[1] This paper provides an overview of the full version [16].
[2] All references to numbers of theorems, lemmas, and definitions of [13] refer to their full version [14].

▶ **Definition 3.1** ([13, Definition 4.1]). *Let S be a set of alignments $\mathcal{X} : P[\,p\,..\,p'\,) \rightsquigarrow T[\,t\,..\,t'\,)$. We define the undirected graph $\mathbf{G}_S = (V, E)$ as follows. The vertex set V contains*

1. *$|P|$ vertices representing characters of P;*
2. *$|T|$ vertices representing characters of T; and*
3. *one special vertex $\perp$.*

The edge set E contains the following edges for each alignment $\mathcal{X} \in S$.

1. *$\{P[\,x\,], \perp\}$ for every character $P[\,x\,]$ that $\mathcal{X}$ deletes;*
2. *$\{\perp, T[\,y\,]\}$ for every character $T[\,y\,]$ that $\mathcal{X}$ inserts;*
3. *$\{P[\,x\,], T[\,y\,]\}$ for every pair of characters $P[\,x\,]$ and $T[\,y\,]$ that $\mathcal{X}$ aligns.*

We say that an edge $\{P[\,x\,], T[\,y\,]\}$ is black *if $\mathcal{X}$ matches $P[\,x\,]$ and $T[\,y\,]$. All other edges are* red*.*

A connected component of $\mathbf{G}_S$ is red *if it contains at least one red edge; otherwise, the connected component is* black*. We denote the number of black components with* $\mathrm{bc}(\mathbf{G}_S)$*.*

Note that all vertices contained in black components correspond to characters of P and T. Moreover, all characters of a single black component are the same, because the presence of a black edge indicates that some alignment in S matches the two corresponding characters.

The inference graph $\mathbf{G}_S$ is represented implicitly via the edit information $\mathsf{E}_{P,T}(\mathcal{X})$ for each alignment $\mathcal{X} : P[\,p\,..\,p'\,) \rightsquigarrow T[\,t\,..\,t'\,)$ in S. In [13] we naively use the fact that $\mathcal{O}(k \log(m|\Sigma|))$ bits suffice to encode this information; here, we provide a (slightly) more efficient encoding argument that is required to obtain our tight results for small alphabets.

▶ **Lemma 3.2.** *Let S be a set of alignments $\mathcal{X} : P[\,p\,..\,p'\,) \rightsquigarrow T[\,t\,..\,t'\,)$. The set $\{\mathsf{E}_{P,T}(\mathcal{X}) : \mathcal{X} \in S\}$ together with the starting/ending points for each $\mathcal{X} \in S$ can be encoded using*

$$\mathcal{O}\left(|S| \log m + \sum\nolimits_{\mathcal{X} \in S : \mathrm{cost}(\mathcal{X}) > 0} \mathrm{cost}(\mathcal{X}) \cdot \log\left(\tfrac{m|\Sigma|}{\mathrm{cost}(\mathcal{X})}\right)\right) \text{ bits.}$$

This information, together with the bit encoding of $n = |T|$ and $m = |P|$, suffices to fully reconstruct the complete edge set of $\mathbf{G}_S$ and the color of each edge. Moreover, this information suffices to identify the character $\sigma \in \Sigma$ for every node in a red component.

Proof. For each $\mathcal{X} \in S$ with $\mathrm{cost}(\mathcal{X}) > 0$, the elements in $\mathsf{E}_{P,T}(\mathcal{X})$ are monotone in their first and third components. Using a "stars and bars" encoding, these components take $\mathcal{O}(\log \binom{m+\mathrm{cost}(\mathcal{X})}{\mathrm{cost}(\mathcal{X})})$ bits. If $\mathrm{cost}(\mathcal{X}) \le k = o(m)$, this simplifies to $\mathcal{O}(\mathrm{cost}(\mathcal{X}) \log(m/\mathrm{cost}(\mathcal{X})))$. The second and fourth components are encoded separately using $\mathcal{O}(\mathrm{cost}(\mathcal{X}) \log |\Sigma|)$ bits. For each $\mathcal{X} \in S$ with $\mathrm{cost}(\mathcal{X}) = 0$ we only need to store the endpoints using $\mathcal{O}(\log m)$ bits each.

Finally, for every character in P and T within a red component, there exists a path of black edges (possibly of length zero) connecting that character to one incident to a red edge. Since black edges connect identical characters, the character value is invariant along this path. Because we explicitly store all characters incident to red edges, we can deduce the character at the origin of any such path. ◀

Lemma 3.2 provides an encoding that reveals all characters contained in red components of in the inference graph $\mathbf{G}_S$. Thus, the case $\mathrm{bc}(\mathbf{G}_S) = 0$ is easy as we can then fully retrieve P and T.

Consequently, we assume $\mathrm{bc}(\mathbf{G}_S) > 0$ for the rest of the section, as this is the remaining case. Even though one cannot learn the characters in black components via the edit information $\mathsf{E}_{P,T}(\mathcal{X})$ for each alignment $\mathcal{X} : P[\,p\,..\,p'\,) \rightsquigarrow T[\,t\,..\,t'\,)$ in S (and storing all of them would be prohibitive), one can still infer which character belongs to which component.

The key observation of [13] is that black components are extremely structured and appear in a periodic fashion. To make this structure more formal, we define two strings, $T_{|S}$ and $P_{|S}$, obtained by retaining only the characters that belong to black connected components.

▶ **Definition 3.3** ([13, Definition 4.3]). *Let $T_{|S}$ and $P_{|S}$ denote the subsequences of T and P, respectively, consisting of the characters contained in black components of $\mathbf{G}_S$. We denote the lengths of $T_{|S}$ and $P_{|S}$ by n_S and m_S, respectively.*

The first structural observation is that the black edges of the inference graph $\mathbf{G}_S$ induced by a single alignment $\mathcal{X} \in S$ correspond to an *exact occurrence* of $P_{|S}$ in $T_{|S}$.

▶ **Lemma 3.4** ([13, Claim 4.5]). *Let $\mathcal{X} : P \rightsquigarrow T[y\,..\,y')\in S$, and define $y_{\mathcal{X}}, y'_{\mathcal{X}} \in [0\,..\,n_S]$ as the number of characters of $T_{|S}$ contained in $T[0\,..\,y)$ and $T[0\,..\,y')$, respectively. Then, we have $P_{|S} = T_{|S}[y_{\mathcal{X}}\,..\,y'_{\mathcal{X}})$, and $\mathcal{X}$ induces edges between $P_{|S}[p]$ and $T_{|S}[y_{\mathcal{X}}+p]$ for every $p \in [0\,..\,|P_{|S}|)$, and no other edges incident to characters of $P_{|S}$ or $T_{|S}$.*

Many overlapping exact occurrences induce periods of the pattern; thus, in [13] we enforce the following condition on S to ensure the induced exact matchings overlap.

▶ **Definition 3.5** ([13, Definition 4.2]). *We say S* encloses *T if $|T| \le 2|P| - 2k$ and there exist two alignments $\mathcal{X}_{\mathsf{pref}}, \mathcal{X}_{\mathsf{suf}} \in S$ such that $\mathcal{X}_{\mathsf{pref}}$ aligns P with a prefix of T and $\mathcal{X}_{\mathsf{suf}}$ aligns P with a suffix of T, or equivalently $(0,0) \in \mathcal{X}_{\mathsf{pref}}$ and $(|P|,|T|) \in \mathcal{X}_{\mathsf{suf}}$.*

Given this condition on S, we show that not only are $P_{|S}$ and $T_{|S}$ periodic, but even the membership of their characters in black components follows a periodic structure.

▶ **Lemma 3.6** ([13, Lemma 4.4]). *If S encloses T, then, for every $c \in [0\,..\,\mathrm{bc}(\mathbf{G}_S))$, there is a black connected component with node set $\{P_{|S}[i] : i \equiv_{\mathrm{bc}(\mathbf{G}_S)} c\} \cup \{T_{|S}[i] : i \equiv_{\mathrm{bc}(\mathbf{G}_S)} c\}$, that is, a black connected component containing all characters of $P_{|S}$ and $T_{|S}$ appearing at positions congruent to c modulo $\mathrm{bc}(\mathbf{G}_S)$. Moreover, the last characters of $P_{|S}$ and $T_{|S}$ are contained in the same black connected component, that is, $|T_{|S}| \equiv_{\mathrm{bc}(\mathbf{G}_S)} |P_{|S}|$.*

Lemma 3.6 allows us to define the following quantities related to the periodic structure:

- For $c \in [0\,..\,\mathrm{bc}(\mathbf{G}_S))$, we define the *$c$-th black connected component* as the black connected component containing $P_{|S}[c]$ and set $m_c := \lceil (m_S - c)/\mathrm{bc}(\mathbf{G}_S)\rceil$ and $n_c := \lceil (n_S - c)/\mathrm{bc}(\mathbf{G}_S)\rceil$ as the number of characters in P and T, respectively, that belong to the c-th black connected component.
- For $c \in [0\,..\,\mathrm{bc}(\mathbf{G}_S))$ and $j \in [0\,..\,m_c)$, we define $\pi^c_j \in [0\,..\,m)$ as the position of $P_{|S}[c + j\cdot \mathrm{bc}(\mathbf{G}_S)]$ in P. Similarly, for $c \in [0\,..\,\mathrm{bc}(\mathbf{G}_S))$ and $i \in [0\,..\,n_c)$, we define $\tau^c_i \in [0\,..\,n)$ as the position of $T_{|S}[c + i\cdot \mathrm{bc}(\mathbf{G}_S)]$ in T. Note that, for any $c \in [0\,..\,\mathrm{bc}(\mathbf{G}_S))$, the characters $\{T[\tau^c_i]\}_{i=0}^{n_c-1} \cup \{P[\pi^c_j]\}_{j=0}^{m_c-1}$ are exactly those in the c-th black component. Consequently, they are all identical.

3.1.2 Constructing S and the Encoding

In [13], based on S and a weight $w = \mathcal{O}(k|S|)$, we identify a subset of black components $C_S \subseteq [0\,..\,\mathrm{bc}(\mathbf{G}_S))$ so that the characters $P[\pi^c_0]$ with $c \in C_S$ belong to fragments of T whose LZ77 parses consist of $w = \mathcal{O}(k|S|)$ phrases in total. The construction ensures several desirable properties specified below. This is the most technical component of [13], and it is not significantly affected by our modifications, so we omit deeper insights here.

These desirable properties concern all fragments $T[t\,..\,t')$ such that if we were to align P onto $T[t\,..\,t')$, then π^0_0 would be close enough to τ^0_i for some $i \in [0\,..\,n_0)$. More formally:

▶ **Definition 3.7** ([13, Definition 4.28]). *We say that S* captures *$T[\,t\,..\,t'\,)$ if S encloses T and either $\mathrm{bc}(\mathbf{G}_S) = 0$ or $|\tau_i^0 - t - \pi_0^0| \leq w + 3k$ holds for some $i \in [\,0\,..\,n_0\,)$.*

For all such $T[\,t\,..\,t'\,)$, the information carried by S and C_S is sufficient to infer whether $\delta_E(P, T[\,t\,..\,t'\,)) \leq k$ and, if so, provide the exact distance and infer the edit sequence.

▶ **Theorem 3.8** ([13, Corollary 4.30]). *Let S be a set of k-edit alignments of P onto fragments of T such that S encloses T and $\mathrm{bc}(\mathbf{G}_S) > 0$. Construct $P^\#$ and $T^\#$ by replacing, for every $c \notin C_S$, every character in the c-th black component with a unique character $\#_c$.*

If S captures all k-error occurrences, then

1. *$\delta_E(P, T[\,t\,..\,t'\,)) = \delta_E(P^\#, T^\#[\,t\,..\,t'\,))$ and $\mathsf{E}_{P,T}(\mathcal{X}) = \mathsf{E}_{P^\#,T^\#}(\mathcal{X})$ for all optimal alignments $\mathcal{X} : P \rightsquigarrow T[\,t\,..\,t'\,)$ of cost at most k.*
2. *$\delta_E(P, T[\,t\,..\,t'\,)) \leq \delta_E(P^\#, T^\#[\,t\,..\,t'\,))$ for all integers $0 \leq t \leq t' \leq n$.*

Theorem 3.8 makes sure that one can encode the desired information for all captured k-error occurrences. For those that are not captured, the following result is shown in [13].

▶ **Lemma 3.9** ([13, Lemma 4.27]). *Let $\mathcal{Y} : P \rightsquigarrow T[\,t\,..\,t'\,)$ be an alignment of cost at most k. If $|\tau_i^0 - t - \pi_0^0| > w + 2k$ holds for every $i \in [\,0\,..\,n_0 - m_0\,]$, then there is no $c \in [\,0\,..\,\mathrm{bc}(\mathbf{G}_S)\,)$ such that $\mathcal{Y}$ aligns $P[\,\pi_0^c\,]$ with $T[\,\tau_i^c\,]$ for some $i \in [\,0\,..\,n_c\,)$.*

An alignment $\mathcal{Y}$ as in Lemma 3.9 satisfies $\mathrm{bc}(\mathbf{G}_{S\cup\{\mathcal{Y}\}}) \leq \mathrm{bc}(\mathbf{G}_S)/2$, since each black component becomes red or is merged with another black component. Thus, adding $\mathcal{Y}$ to S allows for significant progress towards $\mathrm{bc}(\mathbf{G}_S) = 0$. Altogether, the construction proceeds as follows.

- In the beginning, we set $S = \{\mathcal{X}_{\mathsf{pref}}, \mathcal{X}_{\mathsf{suf}}\}$.
- While S does not capture all k-error occurrences, select an uncaptured k-error occurrence $T[\,t\,..\,t'\,)$ and add to S an optimal alignment $\mathcal{Y} : P \rightsquigarrow T[\,t\,..\,t'\,)$.
- Return the edit information $\mathsf{E}_{P,T}(\mathcal{X})$ for alignments $\mathcal{X} \in S$, encoded using Lemma 3.2, and the set $\{(c, P[\,\pi_0^c\,]) : c \in C_S\}$, encoded using the LZ77-compressed fragments of P.

Since the second step can be executed at most $\mathcal{O}(\log m)$ times before S captures all k-error occurrences, the entire encoding requires $\mathcal{O}(k \log m \log(m|\Sigma|/k))$ bits. To decode, it suffices to retrieve the inference graph $\mathbf{G}_S$ from the edit information and, using the encoding of C_S as described in Theorem 3.8, construct $P^\#$ and $T^\#$, which preserve the solution to the original problem.

3.2 Improved Encoding

To tighten our upper bound on the communication complexity, we revisit the step where an uncaptured alignment $\mathcal{Y}$ is added to S. For this, we strengthen the conclusion of Lemma 3.9, which applies only to the initial character $P[\,\pi_0^c\,]$ of each black component $c \in [\,0\,..\,\mathrm{bc}(\mathbf{G}_S)\,)$. In the full version version, we prove that $\mathcal{Y}$ cannot match any characters within the same black component.

▶ **Lemma 3.10.** *Let $\mathcal{Y} : P \rightsquigarrow T[\,t\,..\,t'\,)$ be an optimal alignment with $\mathrm{cost}(\mathcal{Y}) \leq k$. If $C_S \neq [\,0\,..\,\mathrm{bc}(\mathbf{G}_S)\,)$ and $|\tau_i^0 - t - \pi_0^0| > w + 2k$ holds for every $i \in [\,0\,..\,n_0\,)$, then there is no $c \in [\,0\,..\,\mathrm{bc}(\mathbf{G}_S)\,)$ such that $\mathcal{Y}$ aligns $P[\,\pi_j^c\,]$ with $T[\,\tau_i^c\,]$ for some $i \in [\,0\,..\,n_c\,)$ and $j \in [\,0\,..\,m_c\,)$.*

Lemma 3.10 brings us to the main idea of our improvement: rather than adding an entire uncaptured alignment $\mathcal{Y}$ to S, it suffices to add only the subset of edges required to merge every black component with another component. More specifically, if we decompose P into

$$P[\,0\,..\,\pi_0^0\,) \circ \left(\bigcirc_{j=0}^{m_0-2} P[\,\pi_j^0\,..\,\pi_{j+1}^0\,)\right) \circ P[\,\pi_{m_0-1}^0\,..\,|P|\,),$$

then at least one of the middle $m_0 - 1$ fragments of P in this decomposition carries at most $\mathcal{O}(k/m_0)$ edits in $\mathcal{Y}$. So ideally, we would like to add to S and the corresponding edges $\mathcal{Y}$ restricted to this fragment. If we managed to show that the periodic structure is preserved already by adding such a subset, then not only $\mathrm{bc}(\mathbf{G}_S)$ would halve, but also the parameter m_0 would at least double in each iteration. Hence, with this new strategy, we could hope that the total cost of alignment or partial alignment in S is bounded by $\sum_{i=1}^{\infty} k/2^i = \mathcal{O}(k)$.

On a technical level, the proof is delicate: The selection of the fragment $P[\pi_j^0 \mathinner{..} \pi_{j+1}^0)$, as described so far, depends on $\mathcal{Y}$ and the black components defined relative to S. However, the structure of the black components changes when going from S to $S \cup \mathcal{Y}$, and the fragment is not guaranteed to contain the same black components as in $S \cup \mathcal{Y}$. Thus, for our proof of correctness, we need to be careful to get rid of any such undesired dependence on the future.

Finally, we note that Lemma 3.10 is not a strict strengthening of Lemma 3.9, as it introduces three additional conditions: now $i \in [0 \mathinner{..} n_0)$ instead of $i \in [0 \mathinner{..} n_0 - m_0]$, $\mathcal{Y}$ is optimal, and that $C_S \neq [0 \mathinner{..} \mathrm{bc}(\mathbf{G}_S))$. We observe that all conditions are relatively harmless. Since we are only concerned with capturing all optimal alignments, the first two conditions are perfectly acceptable. Moreover, whenever $C_S = [0 \mathinner{..} \mathrm{bc}(\mathbf{G}_S))$, we have $P = P^{\#}$ and $T = T^{\#}$, where $P^{\#}$ and $T^{\#}$ are defined as in Theorem 3.8. At this point, we know that P and T already have a sufficiently cheap encoding to be sent directly.

3.2.1 Redefining S

Let us start making our improvement formal. To this end, we relax the definition of S: we let S be a set of alignments of *fragments of* P onto fragments of T with cost at most k, and we denote by $\mathrm{cost}(S) \coloneqq \sum_{\mathcal{X} \in S} \mathrm{cost}(\mathcal{X})$ the sum of all costs of the alignments contained in S.

Based on such S, we can define the inference graph $\mathbf{G}_S$ as before and argue that all characters in red components follow from the edit information for all $\mathcal{X} \in S$. We conclude again that $\mathrm{bc}(\mathbf{G}_S) = 0$ is easy, and we assume that $\mathrm{bc}(\mathbf{G}_S) > 0$ in this (sub)section. We also define $P_{|S}$ and $T_{|S}$ as before. Lemma 3.4 can be extended quite easily to fragments of P; see the full version.

▶ **Lemma 3.11.** *Let $\mathcal{X} : P[x \mathinner{..} x') \rightsquigarrow T[y \mathinner{..} y') \in S$ and define $y_{\mathcal{X}}, y'_{\mathcal{X}} \in [0 \mathinner{..} n_S]$ as the number of characters of $T_{|S}$ contained in $T[0 \mathinner{..} y)$ and $T[0 \mathinner{..} y')$. Similarly, define $x_{\mathcal{X}}, x'_{\mathcal{X}} \in [0 \mathinner{..} m_S]$ as the number of characters of $P_{|S}$ contained in $P[0 \mathinner{..} x)$ and $P[0 \mathinner{..} x')$.*

Then, $P_{|S}[x_{\mathcal{X}} \mathinner{..} x'_{\mathcal{X}}) = T_{|S}[y_{\mathcal{X}} \mathinner{..} y'_{\mathcal{X}})$, and $\mathcal{X}$ induces edges between $P_{|S}[x_{\mathcal{X}} + p]$ and $T_{|S}[y_{\mathcal{X}} + p]$ for $p \in [0 \mathinner{..} x'_{\mathcal{X}} - x_{\mathcal{X}})$, and no other edges incident to characters of $P_{|S}$ or $T_{|S}$.

We next give the new condition on S needed to observe the periodic structure.

▶ **Definition 3.12.** *We say S* encloses *T if $|T| \leq 2 \cdot |P| - 2k$ and S can be written as*

$$S = \{\mathcal{X}_{\mathsf{pref}}, \mathcal{X}_{\mathsf{suf}}, \mathcal{A}_1, \ldots, \mathcal{A}_a, \mathcal{B}_1, \ldots, \mathcal{B}_b\}$$

such that the following conditions hold:

(a) *$\mathcal{X}_{\mathsf{pref}}$ aligns the whole P with a prefix of T;*
(b) *$\mathcal{X}_{\mathsf{suf}}$ aligns the whole P with a suffix of T;*
(c) *$\mathcal{A}_1, \ldots, \mathcal{A}_a$ align the whole P with fragments of T.*

We say S is degenerate *if $y_{\mathcal{X}_{\mathsf{pref}}} = y_{\mathcal{X}_{\mathsf{suf}}} = y_{\mathcal{A}_1} = \ldots = y_{\mathcal{A}_a} = 0$. Moreover, we define*

$$g_0 \coloneqq \begin{cases} m_S & \text{if } S \text{ is degenerate,} \\ \gcd\{y_{\mathcal{X}_{\mathsf{pref}}}, y_{\mathcal{X}_{\mathsf{suf}}}, y_{\mathcal{A}_1}, \ldots, y_{\mathcal{A}_a}\} & \text{otherwise.} \end{cases}$$

For each $i \in [1\,..\,b]$, we define $g_i \coloneqq \gcd(g_{i-1}, y_{\mathcal{B}_i} - x_{\mathcal{B}_i})$. We further say that S succinctly encloses *T if the following condition holds for every $i \in [1\,..\,b]$.*

(d) *The alignment $\mathcal{B}_i : P[x_i\,..\,x'_i) \rightsquigarrow T[y_i\,..\,y'_i)$ satisfies $x'_{\mathcal{B}_i} - x_{\mathcal{B}_i} \geq g_{i-1} + 1$.*

If S encloses T, then each of the alignments $\mathcal{X}_{\mathsf{pref}}, \mathcal{X}_{\mathsf{suf}}, \mathcal{A}_1, \ldots, \mathcal{A}_a$ aligns the entire string P with a fragment of T, and the edges they induce connect to all vertices of $P_{|S}$. Consequently, we have $x_{\mathcal{X}_{\mathsf{pref}}} = x_{\mathcal{X}_{\mathsf{suf}}} = x_{\mathcal{A}_1} = \cdots = x_{\mathcal{A}_a} = 0$ and $x'_{\mathcal{X}_{\mathsf{pref}}} = x'_{\mathcal{X}_{\mathsf{suf}}} = x'_{\mathcal{A}_1} = \cdots = x'_{\mathcal{A}_a} = m_S$. Moreover, note that whenever S succinctly encloses T and S is degenerate, then $b = 0$, since the condition (d) cannot hold for $i = 1$ if $b > 0$.

We can also prove something more about the exact matching induced by $\mathcal{X}_{\mathsf{pref}}$ and $\mathcal{X}_{\mathsf{suf}}$.

▶ **Lemma 3.13.** *If S encloses T, then $y_{\mathcal{X}_{\mathsf{pref}}} = 0$, $y_{\mathcal{X}_{\mathsf{suf}}} = n_S - m_S$, and $n_S \leq 2m_S$.*

The proof, provided for completeness in the full version, is the same as [13, Claim 4.6]. This is because the argument relies exclusively on the alignments $\mathcal{X}_{\mathsf{pref}}$ and $\mathcal{X}_{\mathsf{suf}}$ (and not any $\mathcal{B}_i$).

Lemma 3.13 also implies that, when S is degenerate, then $m_S = n_S$ and every alignment $\mathcal{X} \in \{\mathcal{X}_{\mathsf{pref}}, \mathcal{X}_{\mathsf{suf}}, \mathcal{A}_1, \ldots, \mathcal{A}_a\}$ induces in $\mathbf{G}_S$ an edge $\{P_{|S}[p], T_{|S}[t]\}$ iff $p = t$. In this case, for each $p \in [0\,..\,m_S)$, there is a black component consisting solely of $P_{|S}[p]$ and $T_{|S}[p]$.

The following generalization of the inference graph $\mathbf{G}_S$ lets us work with graphs of more predictable structure.

▶ **Definition 3.14.** *Suppose S encloses T. We define the sets $S^0 \coloneqq \{\mathcal{X}_{\mathsf{pref}}, \mathcal{X}_{\mathsf{suf}}, \mathcal{A}_1, \ldots, \mathcal{A}_a\}$ and $S^i \coloneqq \{\mathcal{X}_{\mathsf{pref}}, \mathcal{X}_{\mathsf{suf}}, \mathcal{A}_1, \ldots, \mathcal{A}_a, \mathcal{B}_1, \ldots, \mathcal{B}_i\}$ for $i \in [1\,..\,b]$.*

Moreover, for $i \in [0\,..\,b]$, we define the graph $\mathbf{G}^i_S$ obtained from $\mathbf{G}_S$ as follows. We remove vertices (along with their incident edges) that are not in black components in $\mathbf{G}_S$ (so only vertices in $P_{|S}$ and $T_{|S}$ remain), and we additionally remove any edge that is not induced by alignments in the set S^i.

We now proceed to argue this definition successfully preserves the periodic structure.

▶ **Lemma 3.15.** *Suppose S encloses T. Then, the following hold for each $i \in [0\,..\,b)$.*

1. *If condition* (d) *in Definition 3.12 holds for all indices in $[1\,..\,i]$, then, for each $c \in [0\,..\,g_i)$, the graph $\mathbf{G}^i_S$ has a connected component with node set $C^i_c \coloneqq \{P_{|S}[j] : j \equiv_{g_i} c\} \cup \{T_{|S}[j] : j \equiv_{g_i} c\}$, where j ranges over $[0\,..\,m_S)$ and $[0\,..\,n_S)$, respectively.*
   In particular, in $\mathbf{G}_S = \mathbf{G}^b_S$, for every $c \in [0\,..\,\mathrm{bc}(\mathbf{G}_S))$, there exists a black connected component with node set $\{P_{|S}[i] : i \equiv_{\mathrm{bc}(\mathbf{G}_S)} c\} \cup \{T_{|S}[i] : i \equiv_{\mathrm{bc}(\mathbf{G}_S)} c\}$.
2. *The last characters of $P_{|S}$ and $T_{|S}$ are in the same component of $\mathbf{G}^i_S$.*

Proof. We first prove (1). If S is degenerate (and hence $b = 0$), we have already argued that $\mathbf{G}_S = \mathbf{G}_{S^0} = \mathbf{G}^0_S$ has the claimed structure. Therefore, we assume that S is not degenerate.

Let us assign a unique label $\$_C$ to each component C of $\mathbf{G}^i_S$ and define strings P_{cc} and T_{cc} of length m_S and n_S, respectively, as follows. For $i \in [0\,..\,m_S)$, set $P_{\mathrm{cc}}[i] = \$_C$, where C is the component of $\mathbf{G}^i_S$ containing $P_{|S}[i]$. Similarly, for $i \in [0\,..\,n_S)$, set $T_{\mathrm{cc}}[i] = \$_C$, where C is the black connected component of $\mathbf{G}^i_S$ containing $T_{|S}[i]$.

By Lemmas 3.11 and 3.13, $\{0, |T_{\mathrm{cc}}| - |P_{\mathrm{cc}}|\} \subseteq \{y_{\mathcal{X}_{\mathsf{pref}}}, y_{\mathcal{X}_{\mathsf{suf}}}, y_{\mathcal{A}_1}, \ldots, y_{\mathcal{A}_a}\} \subseteq \mathrm{Occ}(P_{\mathrm{cc}}, T_{\mathrm{cc}})$. Since $|T_{\mathrm{cc}}| \leq 2|P_{\mathrm{cc}}|$, we can use the following claim already proved in [13].

▷ Claim 3.16 ([13, Lemma 3.2]). Consider a non-empty pattern P' and a text T' with $|T'| \leq 2|P'| + 1$. If $\{0, |T'| - |P'|\} \subseteq \mathrm{Occ}(P', T')$, that is, P' occurs both as a prefix and as a suffix of T', then $\gcd(\mathrm{Occ}(P', T'))$ is a period of T'.

It follows that $\gcd(\mathrm{Occ}(P_{\mathrm{cc}}, T_{\mathrm{cc}}))$ is a period of T_{cc}. Hence, $\gcd\{y_{\mathcal{X}_{\mathsf{pref}}}, y_{\mathcal{X}_{\mathsf{suf}}}, y_{\mathcal{A}_1}, \ldots, y_{\mathcal{A}_a}\}$ is a period of both T_{cc} and its prefix P_{cc}, and thus g_0 is a period of both T_{cc} and P_{cc}.

We now prove by induction that T_{cc} and P_{cc} have period g_i for all $i \in [0\,..\,b]$. The base case $i = 0$ follows from the argument above. For $i > 0$, assuming T_{cc} and P_{cc} have period g_{i-1}, we observe that the condition on $\mathcal{B}_i$ ensures $x'_{\mathcal{B}_i} - x_{\mathcal{B}_i} \geq g_{i-1} + 1 \geq g_{i-1}$. The following new claim allows us to conclude that T_{cc} and P_{cc} have period $g_i = \gcd(g_{i-1}, y_{\mathcal{B}_i} - x_{\mathcal{B}_i})$.

▷ Claim 3.17. Consider strings P' and T' with a common string period Q. If there are matching fragments $P'[\,x\,..\,x'\,) = T'[\,y\,..\,y'\,)$ of length $x' - x \geq |Q|$, then P' and T' also have a common period $\gcd(|Q|, y - x)$.

Proof. The claim is trivial if $y = x$, so we henceforth assume otherwise. By symmetry between the fragments, we can also assume without loss of generality that $y > x$. Since P' and T' are prefixes of Q^∞, we have $Q^\infty[\,x\,..\,x+|Q|\,) = Q^\infty[\,y\,..\,y+|Q|\,)$. Observe that $y-x$ is a period of Q^∞: for each $i \in \mathbb{Z}_{\geq 0}$, we indeed have $Q^\infty[\,i\,] = Q^\infty[\,x + (i-x) \bmod |Q|\,] = Q^\infty[\,y+(i-x) \bmod |Q|\,] = Q^\infty[\,i+y-x\,]$. From the Periodicity Lemma (Lemma 2.1) applied for $Q^\infty[\,0\,..\,|Q|+y-x\,)$, we conclude that $\gcd(|Q|, y-x)$ is a period of $Q^\infty[\,0\,..\,|Q|+y-x\,)$ and hence of Q. As a divisor of $|Q|$, it is also a common period of Q^∞, P', and T'. ◁

Consequently, for each $c \in [\,0\,..\,g_i\,)$, the set C^i_c belongs to a single connected component of $\mathbf{G}^i_S$. It remains to prove $g_i = \mathrm{bc}(\mathbf{G}^i_S)$. We do this by demonstrating that no edge of $\mathbf{G}^i_S$ leaves C^i_c. Note that Lemma 3.11 further implies that every edge incident to $P_{|S}$ or $T_{|S}$ connects $P_{|S}[\,p\,]$ with $T_{|S}[\,t\,]$ such that $t = p + (y_{\mathcal{X}} - x_{\mathcal{X}})$ for some $\mathcal{X} \in S$ and $p \in [\,x_{\mathcal{X}}\,..\,x'_{\mathcal{X}}\,)$. In particular, $t \equiv_{g_i} p$, so $P_{|S}[\,p\,] \in C^i_c$ holds if and only if $T_{|S}[\,t\,] \in C^i_c$.

As for (2), since P_{cc} is a suffix of T_{cc}, the last characters of $P_{|S}$ and $T_{|S}$ are connected. ◀

3.2.2 New Iterative Construction of S

For the new construction of S, we use the following key lemma.

▶ **Lemma 3.18.** *Suppose S encloses T and condition* (d) *of Definition 3.12 holds up to i (inclusive) for some $i \in [\,0\,..\,b-1\,)$ (if $i = 0$, then this statement is void). Further, suppose that there exists an integer z such that all characters of $P_{|S^i}[\,z \cdot \mathrm{bc}(\mathbf{G}_{S^i})\,..\,(z+2) \cdot \mathrm{bc}(\mathbf{G}_{S^i})\,)$ are in the pre-image of $\mathcal{B}_{i+1}$. Then, either $\mathrm{bc}(\mathbf{G}_S) = 0$ or condition* (d) *also holds for $i+1$.*

Proof. By Lemma 3.15(1), in the inference graph $\mathbf{G}_{S^i}$, for each $c \in [\,0\,..\,\mathrm{bc}(\mathbf{G}_{S^i})\,)$, there is a black component with node set $\{P_{|S^i}[j] : j \equiv_{\mathrm{bc}(\mathbf{G}_{S^i})} c\} \cup \{T_{|S^i}[j] : j \equiv_{\mathrm{bc}(\mathbf{G}_{S^i})} c\}$. Consider the graph $\mathbf{G}^i_S$. This graph is identical to $\mathbf{G}_{S^i}$ except that the components that become red in $\mathbf{G}_S$ are removed. Let $R \subseteq [\,0\,..\,\mathrm{bc}(\mathbf{G}_{S^i})\,)$ index these removed components.

If $R = [\,0\,..\,\mathrm{bc}(\mathbf{G}_{S^i})\,)$, then $\mathrm{bc}(\mathbf{G}_S) = 0$, and the proof is complete. Otherwise, we proceed as follows. Choose an arbitrary $r \in [\,0\,..\,\mathrm{bc}(\mathbf{G}_{S^i})\,) \setminus R$, and consider the fragment

$$P_{|S^i}[\,z \cdot \mathrm{bc}(\mathbf{G}_{S^i}) + r\,..\,(z+1) \cdot \mathrm{bc}(\mathbf{G}_{S^i}) + r\,] \subseteq P_{|S^i}[\,z \cdot \mathrm{bc}(\mathbf{G}_{S^i})\,..\,(z+2) \cdot \mathrm{bc}(\mathbf{G}_{S^i})\,),$$

whose characters, by our assumption, are also contained in the pre-image of $\mathcal{B}_{i+1}$. The characters $P_{|S^i}[\,z \cdot \mathrm{bc}(\mathbf{G}_{S^i}) + r\,]$ and $P_{|S^i}[\,(z+1) \cdot \mathrm{bc}(\mathbf{G}_{S^i}) + r\,]$ belong to the same black component of $\mathbf{G}_{S^i}$. Since $r \notin R$, these characters survive in $\mathbf{G}^i_S$ and correspond to some characters $P_{|S}[\,p\,]$ and $P_{|S}[\,p'\,]$. Note that we must have $|p' - p| \geq g_i$ as, by Lemma 3.15(1), characters in the same component of $\mathbf{G}^i_S$ lie at least g_i positions apart. Thus, $|P_{|S}[\,p\,..\,p'\,]| \geq g_i + 1$. Since $P_{|S}[\,p\,..\,p'\,]$ is contained in $P_{|S}[\,x_{\mathcal{B}_{i+1}}\,..\,x'_{\mathcal{B}_{i+1}}\,)$, we get $x'_{\mathcal{B}_{i+1}} - x_{\mathcal{B}_{i+1}} \geq g_i + 1$. ◀

Algorithm 1 Construction of S that succinctly encloses T.

Input: Strings P and T, and a threshold k such that P has k-error occurrences as both a prefix and a suffix of T.
Output: A set S that succinctly encloses T such that $\mathrm{cost}(S) = \mathcal{O}(k)$.

1 Initialize $S \coloneqq \{\mathcal{X}_{\mathsf{pref}}, \mathcal{X}_{\mathsf{suf}}\}$, where $\mathcal{X}_{\mathsf{pref}}, \mathcal{X}_{\mathsf{suf}}$ are alignments of cost at most k aligning P with a prefix and a suffix of T, which exist by the input assumptions
2 **for** $\ell \in [1\,..\,2]$ **do**
3 Construct $\mathbf{G}_S$, $P_{|S}$, $T_{|S}$ and C_S for S, and introduce the corresponding notation
4 **if** *S captures all k-error occurrences* **or** $C_S = [0\,..\,\mathrm{bc}(\mathbf{G}_S))$ **then return** S
5 Let $\mathcal{Y} : P \rightsquigarrow\!\!\!\!\rightarrow T[t\,..\,t')$ be an optimal k-edit alignment that S does not capture
6 Add the full alignment $\mathcal{Y}$ to S as the alignment $\mathcal{A}_\ell$
7 **for** $\ell \in \mathbb{Z}_{\geq 1}$ **do**
8 Construct $\mathbf{G}_S$, $P_{|S}$, $T_{|S}$ and C_S for S, and introduce the corresponding notation
9 **if** *S captures all k-error occurrences* **or** $C_S = [0\,..\,\mathrm{bc}(\mathbf{G}_S))$ **then return** S
10 Let $\mathcal{Y} : P \rightsquigarrow\!\!\!\!\rightarrow T[t\,..\,t')$ be an optimal k-edit alignment that S does not capture
11 Select $j \in [0\,..\,m_0 - 2)$ minimizing the cost of the partial alignment $\mathcal{Y}_j \subseteq \mathcal{Y}$, where $\mathcal{Y}_j : P[\pi^0_j\,..\,\pi^0_{j+2}) \rightsquigarrow\!\!\!\!\rightarrow \mathcal{Y}(P[\pi^0_j\,..\,\pi^0_{j+2}))$
12 Add $\mathcal{Y}_j$ to S as the alignment $\mathcal{B}_\ell$

Finally, we can provide the new construction of S. As in [13], we construct, from S, a set C_S of black components and a parameter w based on a weight function covering S. We use the following facts (proved in the full version):

1. Lemma 3.10 still holds, and after replacing "enclosure" with "succinct enclosure" in Definition 3.7, Theorem 3.8 still holds.
2. We can improve the bound for encoding $\{(c, T[\tau^c_0]) : c \in C_S\}$ from $\mathcal{O}(w \log m)$ additional bits (on top of the edit information of S) to $\mathcal{O}\left((w+k)\log(1 + m|\Sigma|/(w+k))\right)$ bits.
3. All of this is possible with $w = \mathcal{O}(\mathrm{cost}(S))$ instead of $w = \mathcal{O}(k|S|)$.

▶ **Lemma 3.19.** *Suppose that $n \leq 2m - 2k$ and that there are k-error occurrences of P appearing as both a prefix and a suffix of T. Then, Algorithm 1 computes a set S of size $|S| = \mathcal{O}(\min(\log m, \lfloor |\mathrm{Occ}^E_k(P,T)|/k \rfloor))$ and $\mathrm{cost}(S) = \mathcal{O}(k)$ such that the following hold:*

1. *S succinctly encloses T,*
2. *$C_S = [0\,..\,\mathrm{bc}(\mathbf{G}_S))$ or S captures all k-error occurrences of P in T,*
3. *the set $\{\mathsf{E}_{P,T}(\mathcal{X}) : S \ni \mathcal{X} : P[p\,..\,p') \rightsquigarrow\!\!\!\!\rightarrow T[t\,..\,t')\}$ together with all starting/ending points of the alignments in S can be encoded in $\mathcal{O}(k\log(m|\Sigma|/k) + |S|\log m)$ bits.*

Proof sketch (full proof in the full version). In Algorithm 1, we maintain the invariant that S succinctly encloses T at the start of every iteration of the loops in Lines 2 and 7.

This invariant holds at the beginning as we set $S \coloneqq \{\mathcal{X}_{\mathsf{pref}}, \mathcal{X}_{\mathsf{suf}}\}$. We proceed by adding at most two alignments $\mathcal{A}_\ell$ to S corresponding to uncaptured k-error occurrences in the loop of Line 2. After the ℓ-th iteration, one of $\mathrm{bc}(\mathbf{G}_S) = 0$, $C_S = [0\,..\,\mathrm{bc}(\mathbf{G}_S))$, or $s_P \geq 2^\ell$ holds, where s_P is the minimum number of characters of P in a black component of $\mathbf{G}_S$. Indeed, by Lemma 3.10, after adding $\mathcal{Y}$ to S, each black component of $\mathbf{G}_S$ either becomes red or is merged with another component. Hence, unless $\mathrm{bc}(\mathbf{G}_{S\cup\{\mathcal{Y}\}}) = 0$, every black component of $\mathbf{G}_{S\cup\{\mathcal{Y}\}}$ contains at least two black components of $\mathbf{G}_S$, so s_P at least doubles.

Adding the two alignments of the form $\mathcal{A}_\ell$ ensures that $m_0 \geq s_P \geq 2^2 = 4$ when we enter the loop at Line 7 (so that $m_0 \geq 3$ in Line 11, and the quantifier $j \in [0\,..\,m_0 - 2)$ is well-defined). There, we start adding alignments of the form $\mathcal{B}_\ell$ to S. Each time we add

such an alignment, we use Lemmas 3.10 and 3.18 to prove that S still succinctly encloses T. Using a similar argument as before, $\mathrm{bc}(\mathbf{G}_S) = 0$, $C_S = [0\,..\,\mathrm{bc}(\mathbf{G}_S))$, or $m_0 \geq s_P \geq 2^{2+\ell}$ at the end of the ℓ-th iteration. Since $s_P \leq m$, this ensures that the loop finishes after $\mathcal{O}(\log m)$ iterations and $|S| = \mathcal{O}(\log m)$. When it does, all k-error occurrences are captured.

Next, we argue why the bound on $\mathrm{cost}(S)$ holds. Since each of the alignments $\mathcal{Y}_j$ can overlap only with at most two others, we obtain that the sum of the costs of the alignments $\mathcal{Y}_j$ for all $j \in [0\,..\,m_0 - 2)$ is at most $2k$. Consequently, in the ℓ-th iteration the cost of the alignment $\mathcal{Y}_j$ selected at Line 11 is at most $\mathrm{cost}(\mathcal{Y}_j) \leq 2k/(m_0-2) \leq 2k/(2^{\ell+1}-2) \leq k/2^{\ell-1}$. Thus, $\mathrm{cost}(S) \leq 4k + k \cdot \sum_{\ell=1}^{\mathcal{O}(\log m)} \frac{1}{2^{\ell-1}} \leq 6k = \mathcal{O}(k)$.

Lastly, we give the bound on the encoding size. The endpoints of the alignments $\mathcal{X} \in S$ can be encoded in $\mathcal{O}(|S| \log m)$ bits. On top of that, by Lemma 3.2, the edit information $\{\mathsf{E}_{P,T}(\mathcal{X}) : \mathcal{X} \in S\}$ can be encoded in space asymptotically bounded as follows, using $S^+ := \{\mathcal{X} \in S : \mathrm{cost}(\mathcal{X}) > 0\}$ and the monotonicity of $x \log(k/x)$ for $0 < x \leq k/e$:

$$\sum_{\mathcal{X}\in S^+} \mathrm{cost}(\mathcal{X}) \cdot \log\left(\tfrac{m|\Sigma|}{\mathrm{cost}(\mathcal{X})}\right) = \mathrm{cost}(S) \cdot \log\left(\tfrac{m|\Sigma|}{k}\right) + \sum_{\mathcal{X}\in S^+} \mathrm{cost}(\mathcal{X}) \cdot \log\left(\tfrac{k}{\mathrm{cost}(\mathcal{X})}\right)$$
$$\leq k \log\left(\tfrac{m|\Sigma|}{k}\right) + \mathcal{O}(k) + k \cdot \sum_{\ell=1}^{\mathcal{O}(\log m)} \tfrac{\ell-1}{2^{\ell-1}} \leq \mathcal{O}\left(k \log\left(\tfrac{m|\Sigma|}{k}\right)\right) \text{ bits.}$$

To show $|S| = \mathcal{O}(|\lfloor \mathrm{Occ}_k^E(P,T)/k \rfloor|)$, we argue in the full proof that the alignments $\mathcal{Y}$ in Line 10 start at least k positions away from each other. ◀

Now, following the approach in [13], we can encode the set $\{\mathsf{E}_{P,T}(\mathcal{X}) : \mathcal{X} \in S\}$ and the pairs $\{(c, T[\tau_0^c]) : c \in C_S\}$ using $\mathcal{O}(k \log(m|\Sigma|/k) + |S| \log m)$ bits. Decoding would then proceed via an appropriately modified version of Theorem 3.8. Unfortunately, this falls short of proving Theorem 1.1 because the $|S| \log m$ term might dominate the encoding size; consequently, we must develop an alternative for cases where this term is prohibitively large.

Workaround. Note that $|S| \log m = \mathcal{O}(|S| \log \frac{m}{\log m}) \leq \mathcal{O}(\log m \cdot \log \frac{m}{\log m})$, so $|S| \log m = \mathcal{O}(k \log \frac{m}{k})$ holds as long as $k \geq \log m$ or $|S| = \mathcal{O}(k)$. Thus, the critical scenario arises when $k < \log m$ and $|S| = \omega(k)$. Not coincidentally, we also proved $|S| = \mathcal{O}(|\lfloor \mathrm{Occ}_k^E(P,T)/k \rfloor|)$ in Lemma 3.19, which lets us derive $|\lfloor \mathrm{Occ}_k^E(P,T)/k \rfloor| = \omega(k)$ in this case. This condition characterizes one of the two fundamental structural results for PMwE from [5]: in a setting (almost) identical to ours, there is a primitive string Q with $|Q| \leq m/128k$ such that the edit distance between P and a prefix of Q^∞ does not exceed $2k$. Moreover, [5, Theorem 5.2] upper-bounds the distance between T and a prefix of Q^∞ by $6k$. The subsequent characterization in [5, Section 5] implies that, for every k-error occurrence $T[t\,..\,t')$ of P, there exists an exact occurrence of Q in P that is matched perfectly both in the alignment from P to the prefix of Q^∞ and in the alignment from P via $T[t\,..\,t')$ to a fragment of Q^∞.

Our workaround is to initialize S so that the periodic structure of $\mathbf{G}_S$ *essentially coincides* with the structure induced by Q. Namely, every black component corresponds to a position $r \in [0\,..\,|Q|)$ and consists of all characters of P and T that the alignments with prefixes of Q^∞ match with characters of the form $Q^\infty[r + j|Q|]$ for $j \in \mathbb{Z}_{\geq 0}$. By Lemma 3.10 and the above characterization of k-error occurrences, this guarantees that all k-error occurrences are captured by S. To achieve such $\mathbf{G}_S$, we include in S alignments for the two approximate occurrences of P as a prefix and a suffix of T, and a third $14k$-edit alignment that, for every j, aligns the jth copy of Q in P with the $(j+1)$th copy of Q in T. This yields the following lemma proved in the full version.

▶ **Lemma 3.20.** *Suppose that $n \le 3/2 \cdot m - 28k$, that there are k-error occurrences of P appearing as both a prefix and a suffix of T, and that there is a primitive string Q with $|Q| \le m/128k$ and $\delta_E(P, Q^*) \le 2k$. Then, we can construct a set S of at most three $14k$-edit alignments of P onto fragments of T such that, when the notions of* succinctly enclosing *and* capturing *are interpreted with threshold $14k$, all of the following hold:*

1. *S succinctly encloses T,*
2. *S captures all k-error occurrences of P in T,*
3. *the set $\{\mathsf{E}_{P,T}(\mathcal{X}) : S \ni \mathcal{X} : P[p..p') \rightsquigarrow T[t..t')\}$ together with all starting/ending points of the alignments in S can be encoded in $\mathcal{O}(k \log(m|\Sigma|/k))$ bits.*

4 Lower Bound

▶ **Theorem 1.3.** *Fix integers n, m, k with $n \ge m \ge k > 0$ and an input alphabet Σ with $0 \in \Sigma$ and $|\Sigma| \ge 2$. Consider an encoding that, for each fragment $T[i..j)$ with $\delta_E(P, T[i..j)) \le k$, allows to reconstruct an edit sequence $P \rightsquigarrow T[i..j)$ and the corresponding edit information. Such an encoding must use $\Omega(n/m \cdot k \log(|\Sigma|m/k))$ bits for $P = 0^m$ and some $T \in \Sigma^n$.*

Proof. Set $p := \lfloor n/m \rfloor$ and $T := S_0 \cdot S_1 \cdots S_{p-1} \cdot 0^{n-pm}$, where $S_0, \ldots, S_{p-1} \in \Sigma^m$ are strings that contain at most k characters from $\Sigma \setminus \{0\}$ and all remaining characters are equal to 0. Clearly, $\delta_E(P, T[qm..(q+1)m)) \le k$ for every $q \in [0..p)$. Moreover, for every $q \in [0..p)$, there is a unique optimal alignment $\mathcal{A}_q : P \rightsquigarrow T[qm..(q+1)m)$ that performs substitutions exactly in the positions where the non-zero characters appear (it is not difficult to verify that any deletions or insertions would increase the cost). Thus, $\mathcal{A}_q$ allows us to fully recover S_q from the edit information $\mathsf{E}_{P,T}(\mathcal{A}_q)$, and consequently T can be fully recovered.

Finally, the number of possibilities for each block S_q is $\sum_{i=0}^{k} \binom{m}{i}(|\Sigma|-1)^i$. If $k \le m/2$, then this quantity is at least $\binom{m}{k}(|\Sigma|-1)^k$, and the standard estimate $\binom{m}{k} = 2^{\Omega(k \log(m/k))}$ yields $\binom{m}{k} = 2^{\Omega(k \log(|\Sigma|m/k))}$ for $|\Sigma| = 2$ and $\binom{m}{k}(|\Sigma|-1)^k \ge \binom{m}{k}(|\Sigma|/2)^k = 2^{\Omega(k \log(|\Sigma|m/k))}$ for $|\Sigma| > 2$. Otherwise, the sum contains the term $\binom{m}{\lceil m/2 \rceil}(|\Sigma|-1)^{\lceil m/2 \rceil}$. If $|\Sigma| = 2$, then this term is simply $\binom{m}{\lceil m/2 \rceil} \ge 2^m/m \ge 2^{\Omega(m)} \ge 2^{\Omega(m \log |\Sigma|)}$. If $|\Sigma| > 2$, then the term is at least $(\Sigma-1)^{\lceil m/2 \rceil} \ge (|\Sigma|/2)^{m/2} = 2^{\Omega(m \log |\Sigma|)}$. In either case, we have $2^{\Omega(m \log |\Sigma|)} \ge 2^{\Omega(k \log(|\Sigma|m/k))}$ when $m/2 < k \le m$. Therefore, the number of possibilities for T is $2^{\Omega(n/m \cdot k \log(|\Sigma|m/k))}$. ◀

References

1 Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). *SIAM Journal on Computing*, 2018. doi:10.1137/15M1053128.

2 Sudatta Bhattacharya and Michal Koucký. Streaming k-edit approximate pattern matching via string decomposition. In *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023*, 2023. doi:10.4230/LIPIcs.ICALP.2023.22.

3 Philip Bille, Gad M. Landau, Rajeev Raman, Kunihiko Sadakane, Srinivasa Rao Satti, and Oren Weimann. Random access to grammar-compressed strings and trees. *SIAM Journal on Computing*, 2015. doi:10.1137/130936889.

4 Alejandro Cassis, Tomasz Kociumaka, and Philip Wellnitz. Optimal algorithms for bounded weighted edit distance. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023*, 2023. doi:10.1109/FOCS57990.2023.00135.

5 Panagiotis Charalampopoulos, Tomasz Kociumaka, and Philip Wellnitz. Faster approximate pattern matching: A unified approach. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, 2020. doi:10.1109/FOCS46700.2020.00095.

6 Panagiotis Charalampopoulos, Tomasz Kociumaka, and Philip Wellnitz. Faster pattern matching under edit distance: A reduction to dynamic puzzle matching and the seaweed monoid of permutation matrices. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022*, 2022. doi:10.1109/FOCS54457.2022.00072.

7 Panagiotis Charalampopoulos, Tomasz Kociumaka, and Philip Wellnitz. Pattern matching under weighted edit distance, 2025. Accepted at FOCS 2025. doi:10.48550/arXiv.2510.17752.

8 Raphaël Clifford, Tomasz Kociumaka, and Ely Porat. The streaming k-mismatch problem. In *30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, 2019. doi:10.1137/1.9781611975482.68.

9 Richard Cole and Ramesh Hariharan. Approximate string matching: A simpler faster algorithm. *SIAM Journal on Computing*, 2002. doi:10.1137/S0097539700370527.

10 Nathan J. Fine and Herbert S. Wilf. Uniqueness theorems for periodic functions. *Proceedings of the American Mathematical Society*, 1965. doi:10.1090/S0002-9939-1965-0174934-9.

11 Paweł Gawrychowski and Damian Straszak. Beating $\mathcal{O}(nm)$ in approximate LZW-compressed pattern matching. In *24th International Symposium on Algorithms and Computation, ISAAC 2013*, LNCS, 2013. doi:10.1007/978-3-642-45030-3_8.

12 Daniel Gibney, Ce Jin, Tomasz Kociumaka, and Sharma V. Thankachan. Near-optimal quantum algorithms for bounded edit distance and Lempel-Ziv factorization. In *35th ACM-SIAM Symposium on Discrete Algorithms, SODA 2023*, 2024. doi:10.1137/1.9781611977912.11.

13 Tomasz Kociumaka, Jakob Nogler, and Philip Wellnitz. On the communication complexity of approximate pattern matching. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, 2024. doi:10.1145/3618260.3649604.

14 Tomasz Kociumaka, Jakob Nogler, and Philip Wellnitz. On the communication complexity of approximate pattern matching, 2024. doi:10.48550/arXiv.2403.18812.

15 Tomasz Kociumaka, Jakob Nogler, and Philip Wellnitz. Near-optimal-time quantum algorithms for approximate pattern matching. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 517–534. SIAM, 2025. doi:10.1137/1.9781611978322.15.

16 Tomasz Kociumaka, Jakob Nogler, and Philip Wellnitz. The communication complexity of pattern matching with edits revisited, 2026. arXiv:2604.15601.

17 Tomasz Kociumaka, Ely Porat, and Tatiana Starikovskaya. Small-space and streaming pattern matching with k edits. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021*, 2021. doi:10.1109/FOCS52979.2021.00090.

18 Michal Koucký and Michael Saks. Almost linear size edit distance sketch. In *56th Annual ACM Symposium on the Theory of Computing; STOC 2024*, 2024.

19 Gad M. Landau and Uzi Vishkin. Fast string matching with k differences. *Journal of Computer and System Sciences*, 1988. doi:10.1016/0022-0000(88)90045-1.

20 Gad M. Landau and Uzi Vishkin. Fast parallel and serial approximate string matching. *Journal of Algorithms*, 1989. doi:10.1016/0196-6774(89)90010-2.

21 Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 1965.

22 Peter H. Sellers. The theory and computation of evolutionary distances: Pattern recognition. *Journal of Algorithms*, 1980. doi:10.1016/0196-6774(80)90016-4.

23 Tatiana Starikovskaya. Communication and streaming complexity of approximate pattern matching. In *28th Annual Symposium on Combinatorial Pattern Matching, CPM 2017*, 2017. doi:10.4230/LIPIcs.CPM.2017.13.

24 Alexander Tiskin. Threshold approximate matching in grammar-compressed strings. In *Prague Stringology Conference, PSC 2014*, 2014. URL: http://www.stringology.org/event/2014/p12.html.

25 Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 1977. doi:10.1109/TIT.1977.1055714.