Consensus Patterns (Probably) Has no EPTAS

Christina Boucher* Christine Lo* Daniel Lokshantov*

June 28, 2012

Abstract

Given n length-L strings $S = \{s_1, \ldots, s_n\}$ over a constant size alphabet Σ together with an integer ℓ, where $\ell \leq L$, the objective of *Consensus Patterns* is to find a length-ℓ string s, a substring t_i of each s_i in S such that $\sum_{\forall i} d(t_i, s)$ is minimized. Here $d(x, y)$ denotes the Hamming distance between the two strings x and y. *Consensus Patterns* admits a PTAS [Li et al., STOC 1999, JCSS 2002] is fixed parameter tractable when parameterized by D [Marx, FOCS 2005, SICOMP 2008], and although it is a well-studied problem, improvement of the PTAS to an EPTAS seemed elusive. We prove that *Consensus Patterns* does not admit an EPTAS unless FPT=W[1], answering an open problem from [Fellows et al., STACS 2002, Combinatorica 2006]. To the best of our knowledge, *Consensus Patterns* is the first problem that admits a PTAS, and is fixed parameter tractable when parameterized by the value of the objective function but does not admit an EPTAS under plausible complexity assumptions. The proof of our hardness of approximation result combines parameterized reductions and gap preserving reductions in a novel manner. As an intermediate result used in the proof of our main theorem, we show that *Consensus String with Outliers*, a problem that was recently introduced by the authors in the context of RNA splice site prediction [6], also admits no EPTAS unless FPT=W[1].

*Department of Computer Science and Engineering, University of California, San Diego

1 Introduction

Lanctot et al. [14] initiated the study of *distinguishing string selection problems* in bioinformatics, where we seek a representative string satisfying some distance constraints from each of the input strings. The *Consensus Patterns* problem falls within this broad class of stringology problems. Given n length-L strings $S = \{s_1, \ldots, s_n\}$ over a constant size alphabet Σ together with an integer ℓ, where $\ell \leq L$, the objective of *Consensus Patterns* is to find a length-ℓ string s, a length-ℓ substring t_i of each s_i in S such that $\sum_{\forall i} d(t_i, s)$ is minimized. Here $d(x, y)$ denotes the Hamming distance between the two strings x and y. One specific application of *Consensus Patterns* in bioinformatics is the problem of finding transcription factor binding sites [14, 21]. Transcription factors are proteins that bind to promoter regions in the genome and have the effect of regulating the expression of one or more genes. Hence, the region where a transcription factor binds is very well-conserved, and the problem of detecting such regions can be extrapolated to the problem of finding the substrings $\{t_1, \ldots, t_n\}$.

Consensus Patterns is NP-hard even when the alphabet is binary [15], so we do not expect a polynomial-time algorithm for the problem. On the other hand, the problem admits a *polynomial time approximation scheme* (PTAS), which finds a solution that is at most a factor $(1 + \epsilon)$ worse than the optimum [15] in $n^{O(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon})}$-time. While a superpolynomial dependence of the running time on $\frac{1}{\epsilon}$ is implied by the NP-hardness of *Consensus Patterns*, there is still room for faster approximation schemes for the problem and so a significant effort has been invested in attempting on proving tighter bounds on the running time of the PTAS [4, 5]. If the exponent of the polynomial in the running time of a PTAS is independent of ϵ then the PTAS is called an *efficient PTAS* (EPTAS). An interesting question, posed by Fellows et al. [8] is whether *Consensus Patterns* admits an EPTAS.

The difference in running time of a PTAS and an EPTAS can be quite dramatic. For instance, running a $O(2^{1/\epsilon}n)$-time algorithm is reasonable for $\epsilon = \frac{1}{10}$ and $n = 1000$, whereas running a $O(n^{1/\epsilon})$-time algorithm is infeasible on this same input. Hence, considerable effort has been devoted to improving PTASs to EPTASs, and showing that such an improvement is unlikely for some problems. For example, Arora [2] gave a $n^{O(1/\epsilon)}$-time PTAS for *Euclidean TSP*, which was then improved to a $O(2^{O(1/\epsilon^2)}n^2)$-time algorithm in the journal version of the paper [3]. On the other hand *Independent Set* admits a PTAS on unit disk graphs [13] but Marx [17] showed that it does not admit an EPTAS assuming FPT$\neq$W[1]—a widely believed assumption from parameterized complexity. Many more examples of PTASs that have been improved to EPTASs, and problems for which there exists a PTAS but the existence of an EPTAS has been ruled out under the assumption that FPT$\neq$W[1] can be found in the survey of Marx [18]. In this paper we show that assuming FPT$\neq$W[1], *Consensus Patterns* does not admit an EPTAS, resolving the open problem of Fellows et al. [8]. Since *Consensus Patterns* has a PTAS and is FPT, standard methods for ruling out an EPTAS cannot applied. We discuss this in more details in Section 1.1. Our proof avoids this obstacle by combining gap preseving reductions and parameterized reductions in a novel manner.

In an intermediate step of the proof of our main theorem, we consider *Consensus String with Outliers*, a problem that was recently introduced by the authors in the context of RNA splice site prediction [6]. Here we are given n length-ℓ strings $S = \{s_1, \ldots, s_n\}$ over a finite alphabet Σ and a nonnegative integer k. The objective is to find a consensus string s and subset of $S^* \subset S$, where $n - |S^*| = k$ and $\sum_{\forall t \in S^*} d(s, t)$ is minimal. This problem also admits a PTAS [6], we show that unless FPT=W[1], *Consensus String with Outliers* does not admit an EPTAS.

1.1 Methods

Our lower bounds are proved under the assumption FPT$\neq$W[1], a standard assumption in parameterized complexity that we will briefly discuss here. In a parameterized problem every instance $\mathcal{I}$ comes with a *parameter* k. A parameterized problem is said to be *fixed parameter tractable* (FPT) if there is an algorithm solving instances of the problem in time $f(k)|\mathcal{I}|^{O(1)}$ for some function f depending only on k and not on $|\mathcal{I}|$. The class of all fixed parameter tractable problems is denoted by FPT. The class

W[1] of parameterized problems is the basic class for fixed parameter intractability, FPT $\subseteq$ W[1] and the containment is believed to be proper. A parameterized problem Π with the property that an FPT algorithm for Π would imply that FPT=W[1] is called W[1]-hard. Thus demonstrating W[1]-hardness of a parameterized problem implies that it is unlikely that the problem is FPT. We refer the reader to the textbooks [7, 10, 20] for a more thorough discussion of parameterized complexity.

W[1]-hardness is frequently used to rule out EPTAS's for optimization problems, since an EPTAS for an optimization problem automatically yields a FPT algorithm for the corresponding decision problem parameterized by the value of the objective function [18]. More specifically, if we set $\epsilon = \frac{1}{2\alpha}$, where α is the value of the objective function, then a $(1+\epsilon)$-approximation algorithm would distinguish between "yes" and "no" instances of the problem. Hence, an EPTAS could be used to solve the problem in $O(f(\epsilon)n^{O(1)}) = O(g(\alpha)n^{O(1)})$-time. Hence, if a problem is W[1]-hard when parameterized by the value of the objective function then the corresponding optimization problem does not admit an EPTAS unless FPT=W[1]. To the best of our knowledge, *all* known results ruling out EPTASs for problems for which a PTAS is known use this approach. However, this approach cannot be used to rule out an EPTAS for *Consensus Patterns* because *Consensus Patterns* parameterized by d has been shown to be FPT by Marx [16]. Thus, different methods are required to rule out an EPTAS for *Consensus Patterns*.

In his survey, Marx [18] introduces a hybrid of FPT reductions and gap preserving reductions and argues that it is conceivable that such a reduction could be used to prove that a problem that has a PTAS and is FPT parameterized by the value of the objective function does not admit an EPTAS unless FPT=W[1]. We show that *Consensus Patterns* does not admit an EPTAS unless FPT=W[1], giving the first example of this phenomenon. At the core of our reduction is an analysis of one-dimensional random walks where some of the steps are "double steps" that are taken in the same direction. The results on random walks could turn out useful in other hardness proofs, and thus, might be of independent interest.

Preliminaries

A PTAS for a minimization problem finds a $(1+\epsilon)$-approximate solution in time $|\mathcal{I}|^{f(1/\epsilon)}$ for some function f. An approximation scheme where the exponent of $|\mathcal{I}|$ in the running time is independent of ϵ is called an *efficient* polynomial time approximation scheme (EPTAS). Formally, an EPTAS is a PTAS whose running time is $f(1/\epsilon)^{O(1)}|\mathcal{I}|^{O(1)}$.

Let $L, L' \subseteq \sum^* \times \mathbb{N}$ be two parameterized problems. We say that L *fpt-reduces* to L' if there are functions $f, g : \mathbb{N} \to \mathbb{N}$, and an algorithm that given an instance $(\mathcal{I}, k)$ runs in time $f(k)|\mathcal{I}|^{f(k)}$ and outputs an instance $(\mathcal{I}', k')$ such that $k' \leq g(k)$ and $(\mathcal{I}, k) \in L \iff (\mathcal{I}', k') \in L'$. These reductions work as expected; if L fpt-reduces to L' and L' is FPT then so is L'. Furthermore, if L fpt-reduces to L' and L is W[1]-hard then so is L'.

Let s be a string over the alphabet Σ. We denote the length of s as $|s|$, and the jth character of s as $s[j]$. Hence, $s = s[1]s[2]\ldots s[|s|]$. For a set S of strings of the same length we denote by $S[i]$ as $\{s[i] \ : \ s \in S\}$. Thus, if the same character appears at position i in several strings it is counted several times in $S[i]$. For an interval $P = \{i, i+1, \ldots, j-1, j\}$ of integers, define $s[P]$ to be the substring $s[i]s[i+1]\ldots s[j]$ of s. For a set S of strings and interval P define $S[P]$ to be the (multi)set $\{s[P] \ : \ s \in S\}$. For a set S of length-ℓ strings we define the *consensus string* of S, denoted as $c(S)$, as the sequence where $c(S)[i]$ is the most-frequent character in $S[i]$ for all $i \leq \ell$. Ties are broken by selecting the lexicographically first such character, however, we note that the tie-breaking will not affect our arguments.

We denote the sum Hamming distance between a string, s, and a set of strings, S, as $d(S, s)$. Observe that the consensus string $c(S)$ minimizes $d(S, c(S))$—implying that no other string x is closer to S than $c(S)$. However, some $x \neq c(S)$ could achieve $d(S, x) = d(S, c(S))$ and we refer to such strings as *majority strings* because they are obtained by picking a most-frequent character at every position with ties broken arbitrarily.

We will use standard concentration bounds for sums of independent random variables. In particular,

the following variant of the Hoeffding's bound [12] given by Grimmett and Stirzaker [11, p. 476] will be needed.

Proposition 1. (Hoeffding's bound) *Let $X_1, X_2, ...X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ for all i. Let $X = \Sigma_i X_i$ and the expected value of X be $E[X]$ then it follows that:*

$$\Pr[X - E[X] \geq t] \leq \exp\left(\frac{-2t^2}{\Sigma_{i=1}^{n}(b_i - a_i)^2}\right).$$

2 Hardness of Approximating Consensus String with Outliers

In *Consensus String with Outliers* we are given a set S of strings and an integer k, the objective is to find a subset $S^* \subseteq S$ of size $n^* = n - k$ minimizing $d(S^*, c(S^*))$. In [6] we show that the problem admits a PTAS. We now show that unless FPT=W[1], *Consensus String with Outliers* does not admit an EPTAS. In section 3 we use this result as a starting point to prove that unless FPT=W[1], *Consensus Patterns* does not admit a EPTAS either. Specifically, we prove the following theorem.

Theorem 1. *There exists no EPTAS for* Consensus String With Outliers*, unless FPT = W[1].*

In order to prove theorem 1 we show that a *gap* version of *Consensus String With Outliers* is W[1]-hard. We follow the notation of Marx [18].

Gap-Consensus String with Outliers	
Input:	A set[1] of n length-ℓ strings $S = \{s_1, \ldots, s_n\}$ over a finite alphabet Σ, an integer $n^* \leq n$, a rational ϵ and two integers D_{yes} and D_{no} with $D_{no} \geq D_{yes}(1+\epsilon)$ such that either there exists a set S^* of size n^* such that $d(S^*, c(S^*)) \leq D_{yes}$ or for every S^* of size n^*, $d(S^*, c(S^*)) \geq D_{no}$.
Parameter:	$\lceil 1/\epsilon \rceil$
Question:	Is there an S^* such that $d(S^*, c(S^*)) \leq D_{yes}$?

Clearly an EPTAS for *Consensus String with Outliers* could be used to solve *Gap-Consensus String with Outliers* in time $f(\epsilon)(n\ell)^{O(1)}$. Hence Theorem 1 is a direct consequence of the following lemma, which is proved over the remainder of this section.

Lemma 1. Gap-Consensus String with Outliers *is W[1]-hard.*

The proof of Lemma 1 is by reduction from the *MultiColored Clique (MCC)* problem. Here input is a graph G, an integer k and a partition of $V(G)$ into $V_1 \uplus V_2 \ldots V_k$ such that for each i, $G[V_i]$ is an independent set. The task is to determine whether G contains a clique C of size k. Observe that such a clique must contain exactly one vertex from each V_i, since for each i we have $C \cap V_i \leq 1$. It is well-known that MCC is W[1]-hard [9].

Given an instance (G, k) of MCC we produce in time $f(k)n^{O(1)}$ an instance (S, n^*) of *Gap-Consensus String with Outliers* with the following property. If G has a k-clique then there exists an $S' \subset S$ of size n^* such that $d(S', c(S')) \leq D_{yes}$, whereas if no k-clique exists in G then for each $S' \subset S$ of size n^* we have $d(S', c(S')) \geq D_{no}$. The values of D_{yes} and D_{no} will be chosen later in the proof, but the crux of the construction is that $D_{no} \geq \left(1 + \frac{1}{h(k)}\right) D_{yes}$. Hence, an $f(\epsilon)(n\ell)^{O(1)}$ time algorithm for *Gap-Consensus String with Outliers* could be used to solve to solve the MCC problem in time $g(k)n^{O(1)}$ by setting $\epsilon = \frac{1}{2h(k)}$. Thus the reduction is a parameterized, gap-creating reduction where the size of gap decreases as k increases but the decrease is a function of k only.

Construction. We describe how the instance (S, n^*) is constructed from (G, k). Our construction is randomized, and will succeed with probability $\frac{2}{3}$. To prove Theorem 1 we have to change the construction to make it deterministic but for now let us not worry about that. We start by considering the instance (G, k) and let $E(G) = \{e_1, e_2, \dots e_m\}$. We partition the edge set $E(G)$ into sets $E_{p,q}$ where $1 \leq p < q \leq k$ as follows; $e_i \in E_{p,q}$ if $e_i = uv$, $u \in V_p$ and $v \in V_q$.

Edges of G are unordered pairs uv of vertices of G. An *edge endpoint* $\hat{e}$ is an *ordered* pair (u, v) of vertices of G such that uv is an edge of G. We denote the set of all edge endpoints of G by $\hat{E}(G) = \{\hat{e}_1, \hat{e}_2, \dots \hat{e}_{2m}\}$. There are two edge endpoints that correspond to the same edge. For two edge endpoints $\hat{e}_p$ and $\hat{e}_q$ that both correspond to the edge e_r we say that $\hat{e}_p \sim \hat{e}_q$, $\hat{e}_p \sim e_r$ and that $\hat{e}_q \sim e_r$. For every $i \leq k$ define the set $\hat{E}_i = \{(u, v) \in \hat{E} \ : \ u \in V_i\}$.

Based on G and k, we select two integers ℓ_1 and ℓ_2, that satisfy the following proerties; $\ell_1 = f \cdot \log n$, $\ell_2 = g \cdot \ell_1$ for some $f \geq 1$ and $g \geq 1$ that depend only on k. The exact value of ℓ_1 and ℓ_2 will be discussed later in the proof. We construct a set $Z = z_1, z_2, \dots z_{2m}$ of strings, Z will act as a "pool of random bits" in our construction. For each endpoint $\hat{e}_i \in \hat{E}(G)$ we make a string z_i as follows.

$$z_i = \overline{a}_i^1 \circ \overline{a}_i^2 \dots \circ \overline{a}_i^k \circ \overline{b}_i^{1,2} \circ \overline{b}_i^{1,3} \dots \overline{b}_i^{1,k} \circ \overline{b}_i^{2,3} \circ \overline{b}_i^{2,4} \dots \circ \overline{b}_i^{k-1,k}$$

For every p, $\overline{a}_i^p$ is a random binary string of length ℓ_1. For every p and q, $\overline{b}_i^{p,q}$ is a random binary string of length ℓ_2. For each p and vertex $u \in V_p$ we make an identification string $id(u)$ of length ℓ_1. Let i be the smallest integer such that the edge endpoint $\hat{e}_i$ is (u, v) for some v. We set $id(u) = \overline{a}_i^p$. Similarly, for every pair of integers $p \leq q$ and each edge $e \in E_{p,q}$ make an identification string $id(e)$ of length ℓ_2. Let i be the smallest integer such that $\hat{e}_i \sim e$. We set $id(e) = \overline{b}_i^{p,q}$. We now make the set S of strings in our instance. For each endpoint $\hat{e}_i \in \hat{E}(G)$ we make a string s_i as follows.

$$s_i = a_i^1 \circ a_i^2 \dots \circ a_i^k \circ b_i^{1,2} \circ b_i^{1,3} \dots b_i^{1,k} \circ b_i^{2,3} \circ b_i^{2,4} \dots \circ b_i^{k-1,k}$$

Here $a_i^p = id(u)$ if $\hat{e}_i = (u, v) \in \hat{E}_p$ and $a_i^p = \overline{a}_i^p$ otherwise. Also, $b_i^{p,q} = id(uv)$ if $\hat{e}_i \sim uv$, $u \in V_p$ and $v \in V_q$. Otherwise $b_i^{p,q} = \overline{b}_i^{p,q}$. We refer to a_i^1 through a_i^k as the *vertex blocks* of s_i and the $b_i^{p,q}$'s are the *edge blocks* of s_i. We refer to a_i^ps as the p'th vertex block and to the the $b_i^{p,q}$s as the (p, q)'th edge block. We set $n^* = 2\binom{k}{2}$, $L = k \cdot \ell_1 + \binom{k}{2} \cdot \ell_2$, and $N = |S| = 2m$, this concludes the construction. Recall that n^* is the size of the solution S^* sought for and observe that L is the length of the constructed strings in S.

Analysis. We consider the constructed strings s_i as random variables, and for every j the character $s_i[j]$ is also a random variable which takes value 1 with probability $1/2$ and 0 with probability $1/2$. Observe that for $j \neq j'$ and any i and i' the random variables $s_i[j]$ and $s_{i'}[j']$ are independent. On the other hand $s_i[j]$ and $s_{i'}[j]$ could be dependent. However, if $s_i[j]$ and $s_{i'}[j]$ are dependent then, by construction $s_i[j] = s_{i'}[j]$. Let $S^* \subset S$ such that $|S^*| = n^*$. Here we consider S^* as a set of random string variables, rather than a set of strings. We are interested in studying $d(S^*, c(S^*))$ for different choices of the set S^*. We can write out $d(S^*, c(S^*))$ as $\sum_{p=1}^{L} d(S^*[p], c(S^*)[p])$ and so $d(S^*, c(S^*))$ is the sum of L independent random variables, each taking values from 0 to n^*. Thus, when L is large enough $d(S^*, c(S^*))$ is sharply concentrated around $E[d(S^*, c(S^*))]$.

We turn our attention to $E[d(S^*, c(S^*))]$ for different choices of S^*. The two main cases that we distinguish between is whether S^* corresponds to the set of edge endpoints of a clique in G or not. Before proceeding to these cases, we need some additional definitions. Let $\vec{v}$ be a vector of positive integers. We define the random variable $X_{\vec{v}} = \vec{W} \cdot \vec{v}$ where $\vec{W}$ is a random vector with same dimension as $\vec{v}$, such that each coordinate of $\vec{W}$ is drawn from $\{-1, 1\}$ uniformly at random. The variable $X_{\vec{v}}$ is interpreted as follows: start a one-dimensional random walk at 0, in each step of the walk we go left or right with probability $1/2$. However, the length of the different steps varies, in step i the walk jumps $\vec{v}[i]$ to the left or right. The value of $X_{\vec{v}}$ is the offset from the origin at the end of the walk. The total *length* of the random walk is $\sum_i \vec{v}[i]$ whereas the *number of steps* of the walk is the dimension of $\vec{v}$

Let j be a position in an edge block. What we mean by this is that $s_i[j]$ is a character in $y_i^{p,q}$. Suppose no two strings of S^* correspond to edge endpoints of the same edge. Then $d(S^*[j], c(S^*[j]))$ is distributed as $n^*/2 - |X_{\vec{v}}|$ where $\vec{v}$ is a n^*-dimensional vector of 1s. Specifically for all $s_i \in S^*$ the $s_i[j]$s are independent so $c(S^*[j])$ is the majority character out of n^* characters independently drawn from $\{0, 1\}$, and $d(c(S^*[j], S^*[j]))$ is the number of occurrences of the minority character. This is distributed as $n^*/2 - |X_{\vec{v}}|$.

Again, let j be a position in the (p, q)-edge block, but now suppose that S^* contains t pairs of edge endpoints that correspond to the same edge in $E_{p,q}$. S^* can also contain single endpoints of edges from $E_{p,q}$ or both endpoints of edges in $E_{p',q'}$ for $(p', q') \neq (p, q)$ but we do not count these. From the construction of the (p, q)-edge block it follows that $d(S^*[j], c(S^*[j]))$ is distributed as $n^*/2 - |X_{\vec{v}}|$ where $\vec{v}$ is a $n^* - t$ dimensional vector with t entries of value 2 and $n^* - 2t$ entries with value 1. We define the random variable $X_{r,t}^i = i + X_{\vec{v}}$ where v is a vector with $r - 2t$ entries that are 1 and t entries that are 2. Intuitively $X_{r,t}^i$ is the offset from 0 of a random walk starting at i of length r, with t steps of length 2 and the remaining steps of length 1. We set $x_{r,t}^i = E[|X_{r,t}^i|]$. Finally, we define E_{yes} as

$$E_{yes} = k \cdot \ell_1 \cdot (n^*/2 - x_{n^*-k+1,0}^{k-1}) + \binom{k}{2} \cdot \ell_2 \cdot (n^*/2 - x_{n^*,1}^0) \tag{1}$$

Lemma 2. *Let S^* be a subset of S of size n^* that corresponds to the set of edge endpoints of a k-clique in G. Then $E[d(S^*, c(S^*))] = E_{yes}$.*

Proof. For each position j in a vertex block, consider the distribution of $d(S^*[j], c(S^*)[j])$. There are $k - 1$ edge endpoints in S^* which are all incident to the same vertex v, so the strings corresponding to these endpoints all have the same character at position j. The remaining strings all have random characters at this position. Hence $d(S^*[j], c(S^*)[j])$ is distributed as $n^*/2 - |X_{\vec{v}}|$ where $\vec{v}$ is a $n^* - (k-2)$ dimensional vector with $n^* - (k-1)$ entries of value 1 and one entry with value $k-1$. It is easy to see that $|X_{\vec{v}}|$ is in fact distributed as $|X_{n^*-k+1,0}^{k-1}|$ since we can make the step corresponding to the entry of value $k-1$ first, and this step will take the random walk to position $k-1$ or $-(k-1)$, but with respect to distance from 0 these positions are symmetric. Since there are $k \cdot \ell_1$ positions in vertex blocks this accounts for the first term of the equation.

For each position j in an edge block (p, q) there are two strings in S^* that correspond to edge endpoints of the same edge in $E_{p,q}$. These two strings have the same character at position j. All the other strings in S^* correspond to edge endpoints of strings in $E_{p',q'}$ where $p' \neq p$ or $q' \neq q$. The characters at position j for these strings are drawn independently. Hence $d(S^*[j], c(S^*)[j])$ is distributed as $n^*/2 - E[|X_{n^*,1}^0|]$. Since there are $\binom{k}{2} \cdot \ell_2$ positions in edge blocks this accounts for the second term of the equation. □

We now proceed to show that for any set S^* that does not correspond to a set of edge endpoints of a k-clique in G, $E[d(S^*, c(S^*))]$ is at least factor ϵ greater than E_{yes}, where ϵ depends only on k. Let $\hat{E}^*$ be the set of edge endpoints corresponding to S^*. Define E^* to be the set of edges $uv \in E(G)$ such that $(u, v) \in \hat{E}^*$ and $(v, u) \in \hat{E}^*$. Clearly, $|E^*| \leq \binom{k}{2}$, hence if $E_{p,q} \cap E^* \neq \emptyset$ for every p, q then $|E_{p,q} \cap E^*| = 1$ for every p,q. We start by proving that if there exists a p, q such that $E_{p,q} \cap E^* = \emptyset$ then $E[d(S^*, c(S^*))]$ is big. This proof is based on "differentiating" $x_{n^*,t}^0$ with respect to t. In particular for integers i, r, t such that $r \geq 1$ and $t \geq 2$ define $\delta x_{r,t}^i = x_{r,t}^i - x_{r,t-1}^i$.

Claim 1. $x_{n^*,0}^0 < x_{n^*,1}^0$. *If n^* is divisible by 4 then $\delta x_{n^*,1}^0 > \delta x_{n^*,t}^0$ for all $t > 1$. Furthermore, for every i,t and r we can compute $x_{r,t}^i$ in time polynomial in i and r.*

The intuition of Claim 1 is as follows. A random walk with double steps is just the sum of independent random variables, with variables corresponding to single steps taking values from $\{-1, 1\}$ and variables corresponding to double steps taking values from $\{-2, 2\}$. A double step has higher variance than the sum of two single steps. Hence, if we do a random walk starting from 0 of total length n^*

with t double steps, then the expected distance from 0 should increase as t increases. Furthermore, as t increases the variance of the random walk increses linearly, so the standard deviation increases less and less with each increment of t. Thus it is natural to expect that as t increases, each successive step increases the expected offset from 0 less and less. Quite surprisingly this does not hold in general (we do not prove this, as it is not important for our results). However, when the length of the random walk is a multiple of 4, the claim does hold.

Proof of Claim 1. Recall that $x^i_{r,t} = E[|X^i_{r,t}|]$ where $X^i_{r,t}$ is a random variable denoting the final position of a random walk of length r, with t double steps, starting at i. Here i is an integer and might be negative. Conditional expectation yields the following recurrence for $x^i_{r,t}$, $r \geq 2t \geq 0$.

$$x^i_{r,t} = \begin{cases} |i| & \text{if } r = 0, \\ (x^{i+1}_{r-1,t} + x^{i-1}_{r-1,t})/2 & \text{if } r > 2t, \\ (x^{i+2}_{r-2,t-1} + x^{i-2}_{r-2,t-1})/2 & \text{if } t \geq 1. \end{cases}$$

It is easy to see that one of the three cases must apply when $r \geq 2t \geq 0$ - and $x^i_{r,t}$ is only defined for these values. Observe that if $r > 2t$ and $t \geq 1$ then both the second and the third case apply. The recurrence above also yields a polynomial time algorithm to compute $x^i_{r,t}$. The recurrence above together with definition of $\delta x^i_{r,t}$ yields the following recurrence for $\delta x^i_{r,t}$, for $r \geq 2t$ and $t \geq 1$.

$$\delta x^i_{r,t} = \begin{cases} 0 & \text{if } r = 2, |i| \geq 2, \\ 1/2 & \text{if } r = 2, |i| = 1, \\ 1 & \text{if } r = 2, |i| = 0, \\ (\delta x^{i+1}_{r-1,t} + \delta x^{i-1}_{r-1,t})/2 & \text{if } r > 2t, \\ (\delta x^{i+2}_{r-2,t-1} + \delta x^{i-2}_{r-2,t-1})/2 & \text{if } t \geq 2. \end{cases}$$

A straightforward induction using this recurrence shows that $\delta x^0_{r,1} > 0$ for all $r \geq 0$, proving that $x^0_{n^*,0} < x^0_{n^*,1}$. Define $\delta^2 x^i_{r,t} = \delta x^i_{r,t} - \delta x^i_{r,t-1}$. Observe that $\delta^2 x^i_{r,t}$ is only well defined when $r \geq 2t$ and $t \geq 2$. Inserting the recurrence for $\delta x^i_{r,t}$ into the definition of $\delta^2 x^i_{r,t}$ yields the following recurrence for $\delta^2 x^i_{r,t}$.

$$\delta^2 x^i_{r,t} = \begin{cases} 0 & \text{if } r = 4, |i| \geq 4, \\ 1/8 & \text{if } r = 4, |i| = 3, \\ 1/4 & \text{if } r = 4, |i| = 2, \\ -1/8 & \text{if } r = 4, |i| = 1, \\ -1/2 & \text{if } r = 4, |i| = 0, \\ (\delta^2 x^{i+1}_{r-1,t} + \delta^2 x^{i-1}_{r-1,t})/2 & \text{if } r > 2t, \\ (\delta^2 x^{i+2}_{r-2,t-1} + \delta^2 x^{i-2}_{r-2,t-1})/2 & \text{if } t \geq 3. \end{cases} \tag{2}$$

We prove that if r is divisible by 4 then $\delta^2 x^0_{r,2} < 0$ and for all $t > 2$ we have $\delta^2 x^0_{r,t} \leq 0$. These two facts prove that for $t \geq 2$ we have

$$\delta x^0_{r,t} = \delta x^0_{r,1} + \sum_{j=2}^{t} \delta^2 x^0_{r,j} < \delta x^0_{r,1},$$

which is precisely the last statement of the claim.

For integers i, $r \geq 0$, t such that $r \geq 2t$ define $w^i_{r,t}$ to be the number of one dimensional walks of length r with t double steps and $r - 2t$ unit steps that start in 0 and end in i. Observe that $w^i_{r,t} = w^{-i}_{r,t}$. For even $r \geq 4$, expanding Equation 2 for $\delta^2 x^0_{r,t}$ exhaustively yields the following expression.

$$\delta^2 x^0_{r,t} = \Big[-\frac{1}{2} w^0_{r-4,t-2} + \frac{1}{4} w^2_{r-4,t-2} + \frac{1}{4} w^{-2}_{r-4,t-2} \Big] \Big/ 2^{r-4} = \Big[-w^0_{r-4,t-2} + w^2_{r-4,t-2} \Big] \Big/ 2^{r-3}.$$

Hence to prove the statement of the claim it suffices to show that if r is divisible by 4 then $w^0_{r,0} > w^2_{r,0}$ and $w^0_{r,t} \geq w^2_{r,t}$ for $t \geq 1$. For non-negative i, the number of walks satisfies the following recurrence. The number of walks satisfies the following recurrence.

$$w^i_{r,t} = \begin{cases} 1 & \text{if } r = 0, i = 0, \\ 0 & \text{if } r = 0, i \neq 0, \\ w^{i-1}_{r-1,t} + w^{i+1}_{r-1,t} & \text{if } r > 2t, \\ w^{i-2}_{r-2,t-1} + w^{i+2}_{r-2,t-1} & \text{if } t > 1, i \geq 2 \end{cases} \tag{3}$$

It is easy to see that when r is even, $w^{2i}_{r,0} = \binom{r}{\frac{r}{2}-i}$. Since $\binom{r}{x} > \binom{r}{x-1}$ when $x \leq r/2$ it follows that

$$w^{2i}_{r,0} > w^{2(i+1)}_{r,0} \text{for all } i \tag{4}$$

Equation 4 directly implies $w^0_{r,0} > w^2_{r,0}$.

It remains to prove that if r is divisible by 4 then $w^0_{r,t} \geq w^2_{r,t}$. For the case that $r = 2t$, expanding Equation 3 exhaustively yields the following expression.

$$w^i_{2t,t} = \begin{cases} \binom{t}{(2t-i)/4} & \text{if } i \equiv 2t \pmod 4, \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Most importantly, if $0 \leq i \leq i'$ and $w^i_{2t,t}$ is non-zero, then $w^i_{2t,t} \geq w^{i'}_{2t,t}$.

We now prove that when $r - 2t$ is an even, positive integer and $i \geq 0$ then $w^{2i}_{r,t} \geq w^{2(i+1)}_{r,t}$. A special case of this inequality is that when r is divisible by 4 then $w^0_{r,t} \geq w^2_{r,t}$. Observe that when $t = 0$ the inequality follows by Equation 4. We prove the inequality by induction on $r - t$. Observe that when r decreases by 2 while t decreases by 1, $r - t$ decreases. Hence for $t \geq 1$ and $i \geq 1$ we have

$$w^{2i}_{r,t} = w^{2i-2}_{r-2,t-1} + w^{2i+2}_{r+2,t-1} \geq w^{2i}_{r-2,t-1} + w^{2i+4}_{r+2,t-1} = w^{2(i+1)}_{r,t}.$$

Now, for $t \geq 1$ and $i = 0$ we have that $w^0_{r,t} = 2w^0_{r-2,t} + 2w^2_{r-2,t}$ and $w^2_{r,t} = w^0_{r-2,t} + 2w^2_{r-2,t} + w^4_{r-2,t}$. Hence to prove that $w^0_{r,t} \geq w^2_{r,t}$ it suffices to prove $w^0_{r-2,t} \geq w^4_{r-2,t}$. If $r - 2t = 2$ then by Equation 5 we have that either $w^0_{r-2,t} = w^4_{r-2,t} = 0$ or $w^0_{r-2,t} \geq w^4_{r-2,t}$. In both cases this implies $w^0_{r,t} \geq w^2_{r,t}$. Finally, if $r - 2t > 2$ then the induction hypothesis yields

$$w^0_{r,t} = 2w^0_{r-2,t} + 2w^2_{r-2,t} \geq w^0_{r-2,t} + 2w^2_{r-2,t} + w^4_{r-2,t} \geq w^2_{r,t}.$$

Hence, when r is divisible by 4 then $w^0_{r,t} \geq w^2_{r,t}$, concluding the proof of the claim. □

Set $\Delta = \min_{i \leq n^*}(\delta x^0_{n^*,1} - \delta x^0_{n^*,i})$. By Claim 1, $\Delta > 0$. Define

$$E^1_{no} = \binom{k}{2} \cdot \ell_2 \cdot (n^*/2 - x^0_{n^*,1}) + \ell_2 \Delta \tag{6}$$

Observe that if $\ell_2 > \frac{\ell_1 \cdot k \cdot (n^*/2)}{\Delta}$ then $E^1_{no} > E_{yes}$. Selecting ℓ_2 slightly larger than this will ensure the desired gap between E^1_{no} and E_{yes}, so we set

$$\ell_2 = \ell_1 \cdot \left\lceil \frac{k \cdot n^*}{\Delta} \right\rceil. \tag{7}$$

Observe that the ratio between ℓ_2 and ℓ_1 is a function of k.

Lemma 3. *Let S^* be a subset of S of size n^*, where the corrresponding edge set E^* has the property that $|E^* \cap E_{p,q}| \neq 1$ for at least one pair $p, q \leq k$. Then $E[d(S^*, c(S^*))] \geq E^1_{no}$.*

Proof. For any position j in an edge block number p, q, $d(S^*[j], c(S^*)[j])$ is distributed as $X^0_{n^*,t[p,q]}$ where $t[p,q]$ is the number of edges e in $E_{p,q}$ such that for both endpoints of e the strings corresponding to them are in S^*. It follows that $E[d(S^*, c(S^*))] \geq \ell_2 \cdot \sum_{p,q} x^0_{n^*,t[p,q]}$ since here we are just counting the contribution of the edge block positions to the expectation. Since $|S^*| = n^*$ it follows that $\sum_{p,q} x^0_{n^*,t[p,q]} \leq \binom{k}{2}$. We can now use Claim 1 to lower bound the expectation of $d(S^*, c(S^*))$. In particular, we have that

$$\begin{aligned}
E[d(S^*, c(S^*))] &\geq \sum_{p,q} \left(\ell_2 \cdot (n^*/2 - x^0_{n^*,t[p,q]})\right) = \sum_{p,q} \left(\ell_2 \cdot (n^*/2 - x^0_{n^*,1} + x^0_{n^*,1} - x^0_{n^*,t[p,q]})\right) \\
&= \binom{k}{2} \cdot \ell_2 \cdot (n^*/2 - x^0_{n^*,1}) + \ell_2 \cdot (\binom{k}{2} x^0_{n^*,1} - \sum_{p,q} x^0_{n^*,t[p,q]}) \\
&= \binom{k}{2} \cdot \ell_2 \cdot (n^*/2 - x^0_{n^*,1}) + \ell_2 \cdot \left(\binom{k}{2} x^0_{n^*,1} - \binom{k}{2} x^0_{n^*,0} - \sum_{p,q} \sum_{t=1}^{t[p,q]} \delta x^0_{n^*,t}\right) \\
&= \binom{k}{2} \cdot \ell_2 \cdot (n^*/2 - x^0_{n^*,1}) + \ell_2 \cdot \left(\binom{k}{2} \delta x^0_{n^*,1} - \sum_{p,q} \sum_{t=1}^{t[p,q]} \delta x^0_{n^*,t}\right)
\end{aligned}$$

Observe that if $t[p,q] = 1$ for all p, q then

$$\sum_{p,q} \sum_{t=1}^{t[p,q]} \delta x^0_{n^*,t} = \binom{k}{2} \delta x^0_{n^*,1}$$

and the second term of the last equation cancels. However this is not the case, since S^* is assumed *not* to correspond to the set of endpoints of a set of edges that intersects with every $E_{p,q}$. It follows that

$$\sum_{p,q} \sum_{t=1}^{t[p,q]} \delta x^0_{n^*,t} \leq \binom{k}{2} \delta x^0_{n^*,1} - \Delta$$

which in turn implies that

$$E[d(S^*, c(S^*))] \geq \binom{k}{2} \cdot \ell_2 \cdot (n^*/2 - x^0_{n^*,1}) + \ell_2 \Delta = E^1_{no}$$

□

By Lemma 3 we know that any set S^* such that $E[d(S^*, c(S^*))] < E^1_{no}$ corresponds to all the endpoints of an edge set E^* such that for every $p, q \leq k$ we have $E^* \cap E_{p,q} \neq \emptyset$. It remains to prove that if E^* does not correspond to the edge set of a k-clique in G then $E[d(S^*, c(S^*))] \geq E^2_{no}$ for an integer E^2_{no} which is sufficiently large compared to E_{yes}. Observe that for each $i \leq k$ there are exactly $k-1$ edges in E^* that are incident to vertices of V_i. What we prove is that if $E[d(S^*, c(S^*))] < E^2_{no}$ then for every i, the set of edges in E^* that have an endpoint in V_i all come from the same vertex. Just as for the proof of Lemma 3 we need a preliminary claim about the properties of certain random walks. Let $\mathcal{V}$ be the set of all vectors with all positive integer entries such that the sum of the entries is exactly n^* and the sum of all the entries that are not 1 is at most $k-1$. Let $\mathcal{V}' = \mathcal{V} \setminus \vec{u}$ where $\vec{u}$ is the vector in $\mathcal{V}$ with one entry equal to $k-1$. Observe that for this choice of $\vec{u}$, $E[|X_{\vec{u}}|] = x^{k-1}_{n^*-k+1,0}$. Set $\Delta_2 = \min_{\vec{v} \in \mathcal{V}'} x^{k-1}_{n^*-k+1,0} - E[|X_{\vec{v}}|]$

Claim 2. *For every $\vec{v} \in \mathcal{V}'$, $x^{k-1}_{n^*-k+1,0} - E[|X_{\vec{v}}|] > 0$. Hence Δ_2 is positive. Furthermore, Δ_2 can be computed in time $f(k)$ for some function f.*

Proof. To see that Δ_2 can be computed in time $f(k)$ for some function f it is sufficient to observe that the size of the sample space of any variable $X_{\vec{u}}$ is bounded by a function of k and that $|\mathcal{V}|$ is upper bounded by a function of k as well. We now prove that Δ_2 is positive. Consider a vector $\vec{v} \in \mathcal{V}$ and let $\vec{v}'$ be a vector that contains all the entries of $\vec{v}$ that are greater than 1, and possibly some entries that are 1 such that the sum of the entries in $\vec{v}'$ is exactly $k-1$. Conditional expectation yields:

$$E[|X_{\vec{v}}|] = \sum_{i=-k+1}^{k-1} P[X_{\vec{v}'} = i] \cdot x^i_{n^*-k+1,0}.$$

Observe that for every i whose parity is not the same as $k-1$ we have that $P[X_{\vec{v}'} = i] = 0$ since every entry of $\vec{v}'$ is either added or subtracted to get $X_{\vec{v}'}$ and $-1 \equiv 1(\mod 2)$. Furthermore, a simple induction (see below) shows that for every non-negative i' such that $|i'| \leq |i| - 2$, $x^i_{r,0} > x^{i'}_{r,0}$. Together these two facts imply that for all $i \notin \{k-1, -k+1\}$ for which $P[X_{\vec{v}'} = i]$ is non-zero we have $x^i_{n^*-k+1,0} < x^{k-1}_{n^*-k+1,0}$. Since such an i must exist for any $\vec{v} \in \mathcal{V}'$ we have that $x^{k-1}_{n^*-k+1,0} - E[|X_{\vec{v}}|] > 0$.

Finally, we have to prove that for every non-negative $i' \leq i-2$, $x^i_{r,0} > x^{i'}_{r,0}$. We do this by induction on r. For $r = 0$ this clearly holds as $x^i_{0,0} = |i|$. For $r \geq 1$ conditional expectation yields that $x^i_{r,0} = 1/2(x^{i-1}_{r-1,0} + x^{i+1}_{r-1,0}) > 1/2(x^{i'-1}_{r-1,0} + x^{i'+1}_{r-1,0}) = x^{i'}_{r,0}$ if $i' \neq 0$ and $x^i_{i,0} = 1/2(x^{i-1}_{r-1,0} + x^{i+1}_{r-1,0}) > 1/2(2x^1_{r-1,0}) = x^{i'}_{r,0}$ if $i' = 0$. This concludes the proof. □

We are now ready to prove the last part of the reduction. Let

$$E^2_{no} = \binom{k}{2} \cdot \ell_2 \cdot (n^*/2 - x^0_{n^*,1}) + k \cdot \ell_1 \cdot (n^*/2 - x^{k-1}_{n^*-k+1,0}) + \ell_1 \Delta_2. \tag{8}$$

Lemma 4. *Let S^* be a subset of S of size n^* that corresponds to all edge endpoints of a set E^* of edges such that for every $p, q \leq k$ we have $|E^* \cap E_{p,q}| = 1$. If there exist two distinct vertices $v_1, v_2 \in V_i$ such that E^* contains edges incident to both v_1 and v_2 then $E[d(S^*, c(S^*))] \geq E^2_{no}$.*

Proof. Consider a set S^* satisfying the conditions of the lemma. The contribution to $E[d(S^*, c(S^*))]$ of the edge blocks is exactly $\binom{k}{2} \cdot \ell_2 \cdot (n^*/2 - x^0_{n^*,1})$. Now, consider a vertex block i such that there is exactly one vertex $v_i \in V_i$ that is incident to edges of E^*. This block contributes exactly $\ell_1 \cdot (n^*/2 - x^{k-1}_{n^*-k+1,0})$ to $E[d(S^*, c(S^*))]$. Finally, consider a block i such that there are two distinct vertices $v_1, v_2 \in V_i$ such that E^* contains edges incident to both v_1 and v_2. Let $\vec{v}$ be a vector that for each vertex $v \in V_i$ which is incident to at least one edge in E^*, contains an entry which is exactly the number of edges in E^* that v is incident to. For each edge which is not incident to any vertices in V_i the vector $\vec{v}$ contains a entry with value 1. Hence the sum of the entries of $\vec{v}$ is n^*, the sum of all entries in $\vec{v}$ that are greater than 1 is at most $k-1$, and $\vec{v}$ contains no entry which is $k-1$. This is because exactly $k-1$ edges are incident to vertices in V_i and two such edges are incident to distinct vertices. Hence $\vec{v} \in \mathcal{V}'$. Now, observe that the vertex block i contributes exactly $\ell_1 \cdot (n^*/2 - E[|X_{\vec{v}}|])$ to $E[d(S^*, c(S^*))]$. By Claim 2, $E[|X_{\vec{v}}|] \leq x^{k-1}_{n^*-k+1,0} - \Delta_2$. Thus,

$$E[d(S^*, c(S^*))] \geq \binom{k}{2} \cdot \ell_2 \cdot (n^*/2 - x^0_{n^*,1}) + k \cdot \ell_1 \cdot (n^*/2 - x^{k-1}_{n^*-k+1,0}) + \ell_1 \Delta_2 = E^2_{no}$$

□

Set $E_{no} = \min(E^1_{no}, E^2_{no}) = E^2_{no}$. From Equations 1, 6, 8 and 7 we conclude that there exist constants κ_{yes}, κ_{no} and κ_L depending only on k such that $E_{yes} = \kappa_{yes}\ell_1$, $E_{no} = \kappa_{no}\ell_1$ and $L = \kappa_L \ell_1$. Furthermore, $\kappa_{yes} < \kappa_{no}$ and the value of κ_{yes}, κ_{no} and κ_L can be computed in time $f(k)$ for some function f. Set $\kappa'_{yes} = (2\kappa_{yes} + \kappa_{no})/3$ and $\kappa'_{no} = (\kappa_{yes} + 2\kappa_{no})/3$. Then $\kappa_{yes} < \kappa'_{yes} < \kappa'_{no} < \kappa_{no}$. We set $D_{yes} = \kappa'_{yes}\ell_1$ and $D_{no} = \kappa'_{no}\ell_1$.

A randomized analogue of Lemma 1. Before proving Lemma 1 we argue that the randomized construction works. Specifically, we show that if *Gap-Consensus String With Outliers* is W[1]-hard under randomized FPT-reductions. The results proved in this section are not used in the proof of Lemma 1, but they provide useful insights on how the deterministic construction works.

Lemma 5. *For any $S^* \subset S$ such that $|S^*| = n^*$,*

$$P\Big[|d(S^*, c(S^*)) - E[d(S^*, c(S^*))]| > x \cdot \ell_1\Big] \leq 2\exp\left(-2\frac{x^2}{\kappa_L (n^*)^2}\ell_1\right).$$

Proof. We have that $d(S^*, c(S^*)) = \sum_{p=1}^{L} d(S^*[p], c(S^*)[p])$. The $d(S^*[p], c(S^*)[p])$'s are independent random variables taking values from 0 to n^*. Since $L = \kappa_L \ell_1$ it follows that $P[|d(S^*, c(S^*)) - E[d(S^*, c(S^*))]| > x \cdot \ell_1] = P[|d(S^*, c(S^*)) - E[d(S^*, c(S^*))]| > \frac{x}{\kappa_L} \cdot L]$. By Hoeffding's inequality (Proposition 1) it follows that

$$\begin{aligned} P\left[|d(S^*, c(S^*)) - E[d(S^*, c(S^*))]| > \frac{x}{\kappa_L} \cdot L\right] &\leq 2\exp\left(-2\left(\frac{x}{\kappa_L n^*}\right)^2 L\right) \\ &= 2\exp\left(-2\frac{x^2}{\kappa_L (n^*)^2}\ell_1\right) \end{aligned}$$

□

We now define ℓ_1. This value for ℓ_1 is only valid for the randomized construction, and a different value for ℓ_1 is used in the proof of Theorem 1.

$$\ell_1 = \frac{(n^*)^2 \kappa_L}{2(\kappa'_{yes} - \kappa_{yes})} \ln\left(20(2m)^{n^*}\right). \tag{9}$$

Recall that m is the number of edges in the graph G, so $m \leq n^2$ and hence $\ell_1 \leq f \cdot \log n$ for some f depending only on k.

Lemma 6. *If G has a k-clique C, let S^* be the set of strings corresponding to edge endpoints of edges in C. Then $P[d(S^*, c(S^*)) > D_{yes}] \leq \frac{1}{10(2m)^{n^*}}$. If G does not contain a k-clique, then the probability that S contains a subset S^* of size n^* such that $d(S^*, c(S^*)) < D_{no}$ is at most $1/10$.*

Proof. If G has a k-clique C, let S^* be the set of strings corresponding to edge endpoints of edges in C. Then by Lemma 2, $E[d(S^*, c(S^*))] = D_{yes}$. Now, $D_{yes} - E_{yes} = (\kappa'_{yes} - \kappa_{yes})\ell_1$ and hence, by Lemma 5,

$$P[d(S^*, c(S^*)) > D_{yes}] \leq P[|d(S^*, c(S^*)) - E_{yes}| > (\kappa'_{yes} - \kappa_{yes})\ell_1] \leq \frac{1}{10(2m)^{n^*}}.$$

On the other hand, consider a set S^* of size n^* that does not correspond to the edge endpoints of a clique. If S^* does not correspond to a set E^* of edges such that $|E^* \cap E_{p,q}| = 1$ for every p,q, then $E[d(S^*, c(S^*))] \geq E^1_{no} > E_{no}$. If S^* corresponds to a set E^* of edges such that $|E^* \cap E_{p,q}| = 1$, but E^* is not the edge set of a clique in G then there exists an i and $v_1, v_2 \in V_i$ such that E^* contains edges incident to v_1 and to v_2. In this case Lemma 4 yields that $E[d(S^*, c(S^*))] \geq E^2_{no} = E_{no}$. Hence $E[d(S^*, c(S^*))] \geq E_{no}$. Finally, $E_{no} - D_{no} = (\kappa_{no} - \kappa'_{no})\ell_1$ and $(\kappa_{no} - \kappa'_{no})\ell_1 = (\kappa'_{yes} - \kappa_{yes})\ell_1$ and hence, by Lemma 5,

$$P[d(S^*, c(S^*)) \leq D_{no}] = P[E_{no} - d(S^*, c(S^*)) > (\kappa'_{yes} - \kappa_{yes})\ell_1] \leq \frac{1}{10(2m)^{n^*}}.$$

Thus, if G does not contain a clique of size k, the union bound yields that the probability that S contains a subset S^* of size n^* such that $d(S^*, c(S^*)) < D_{no}$ is at most $1/10$. □

We now prove a randomized analogue of Lemma 1.

Lemma 7. *If* Gap-Consensus String With Outliers *is FPT then W[1] $\subseteq$ randomized FPT.*

Proof. Assiming that *Gap-Consensus String With Outliers* has an algorithm with running time $f(\epsilon)(n\ell)^{O(1)}$ we give a randomized fixed parameter tractable algorithm for *MCC* with two sided error. We construct the instance to *Gap-Consensus String With Outliers* as described, here $\epsilon = \frac{D_{no}}{D_{yes}} - 1 = \frac{\kappa'_{no}}{\kappa'_{yes}} - 1$. If the algorithm for *Gap-Consensus String With Outliers* concludes that there is a set S^* such that $d(S^*, c(S^*)) \leq D_{yes}$ the algorithm returns that the input graph G contains a k-clique, otherwise we return that G has no k-clique. The construction takes time $O(f(k)n^{O(1)})$ for some function f, and ϵ depends only on k. Hence the total running time is $g(k)n^c$ for some function g. Thus the algorithm terminates in FPT time.

If G contains a k-clique, then by Lemma 6, with probability at least $1 - \frac{1}{10(2m)^{n^*}} \geq 1 - \frac{1}{n^k}$ there is a set S^* of size n^* such that $d(S^*, c(S^*)) \leq D_{yes}$. If this event occurs, the algorithm for *Gap-Consensus String With Outliers* will correctly find such a set and correctly return "yes". Hence the probability of false negatives is at most $\frac{1}{n^k}$.

If G does not contain a k-clique, then by Lemma 6, with probability at least $9/10$ for every set S^* of size n^* we hae $d(S^*, c(S^*)) > D_{no}$. If this event occurs the algorithm correctly returns "no" and hence the probability if false positives is at most $1/10$. This implies that there is a randomized fixed parameter tractable algorithm for *MCC*, which in turn shows that W[1] $\subseteq$ randomized FPT. □

A Deterministic Construction and Proof of Lemma 1. In order to prove Lemma 1 we need to make the construction deterministic. We only used randomness to construct the set Z, all other steps are deterministic. We now show how Z can be computed deterministically instead of being selected at random, preserving the properties of the reduction. For this, we need the concept of near p-wise independence defined by Naor and Naor [19]. The original definition of near p-wise independence is in terms of sample spaces, we define near p-wise independence in terms of collections of binary strings. This is only a notational difference, and one may freely translate between the two variants.

Definition 1 ([19])**.** *A set $C = \{c_1, c_2, \ldots c_t\}$ of length ℓ binary strings is (ϵ, p)-independent if for any subset C' of C of size p, if a position $i \leq t$ is selected uniformly at random, then*

$$\sum_{\alpha \in \{0,1\}^p} |P[C'[i] = \alpha] - 2^{-p}| \leq \epsilon.$$

Naor and Naor [19] and Alon et al. [1] give determinsitic constructions of small nearly k-wise independent sample spaces. Reformulated in our terminology, Alon et al. prove a slightly stronger version of the following theorem.

Theorem 2 ([1])**.** *For every t, p, and ϵ there is a (ϵ, p)-independent set $C = \{c_1, c_2, \ldots c_t\}$ of binary strings of length ℓ, where $\ell = O(\frac{2^k \cdot k \log t}{\epsilon})$. Furthermore, C can be computed in time $O(|C|^{O(1)})$.*

We use Theorem 2 to construct the set Z. We set

$$\epsilon = \frac{\kappa'_{yes} - \kappa_{yes}}{\kappa_L \cdot n^*}$$

and construct an (ϵ, n^*)-independent set C of $2m$ strings. These strings have length $\ell = f \cdot \log(n)$ for some f depending only on k, and C can be constructed in time $O(gn^{O(1)})$ for some g depending only on k. We set $\ell_1 = \ell$. Observe that since ℓ_2 is an integer multiple of ℓ_1, the length of the strings in Z is an integer multiple of ℓ_1. For every i we set $z_i = c_i \circ c_i \circ \ldots \circ c_i$, where we used κ_L copies of c_i such that z_i is a string of length L. The remaining part of the construction, i.e the construction of S from Z remains unchanged. To distinguish between the deterministically constructed S and the

randomized construction, we refer to the deterministically constructed S as S_{det}. We now prove that for every $S^*_{det} \subseteq S_{det}$ of size n^*, if S^* is the set of strings in the randomized construction that corresponds to the same edge endpoints as S^*_{det}, then $d(S^*_{det}, c(S^*_{det}))$ is almost equal to $E[d(S^*, c(S^*))]$.

For a subset I of $\{1, 2, \ldots, 2m\}$ define $S^*(I) = \{s_i \in S \ : \ i \in I\}$ and $S^*_{det}(I) = \{s_i \in S_{det} \ : \ i \in I\}$. For every $j \leq \kappa_L$, define $P_j = \{\kappa_L \cdot j + 1, \kappa_L \cdot (j+1)\}$. Hence for every i and j, $z_i[P_j] = c_i$. The construction of S_{det} (and S) from Z implies that for every j, the substring there exists a function $f_j : \mathbb{N} \to \mathbb{N}$ such that for any $i \leq 2m$, $s_i[P_j] = z_{f(i)}[P_j]$. For any $I \subseteq \{1, 2, \ldots, 2m\}$ and $j < \kappa_L$ we define $Z^*(I, j) = \{z_{f_j(i)} \ : \ i \in I\}$. This means that for a subset $I \subseteq \{1, 2, \ldots, 2m\}$ of size n^*, the set $Z^*(I, j)$ is the set of n^* strings in Z which $S^*(I)[P_j]$ and $S^*_{det}(I)[P_j]$ depend on. For every set $I \subseteq \{1, 2, \ldots, 2m\}$ of size n^* and integer $j < \kappa_L$ define $d^I_j : \{0,1\}^{n^*} \to \{0, 1, \ldots, n^*\}$ to be a function such that for any $p \in P_j$, if $Z^*(I) = \alpha$ then $d(S^*(I)[p], c(S^*(I)[p])) = d^I_j(\alpha)$ and $d(S^*_{det}(I)[p], c(S^*_{det}(I)[p])) = d^I_j(\alpha)$. Since $S^*(I)[p]$ depends in exactly the same way on $Z^*(I)[p]$ for all $p \in P_j$ the function d^I_j is well defined. For every set $I \subseteq \{1, 2, \ldots, 2m\}$ of size n^* and integer $j < \kappa_L$ we have the following expression for $d(S^*(I)_{det}[P_j], c(S^*(I)_{det}[P_j]))$.

$$d\left(S^*_{det}(I)[P_j], c(S^*_{det}(I)[P_j])\right) = \ell_1 \cdot \sum_{\alpha \in \{0,1\}^{n^*}} P[Z^*(I)[p] = \alpha] \cdot d^I_j(\alpha) \tag{10}$$

Here the probability $P[Z^*(I)[p] = \alpha]$ is taken when p is selected from P_j uniformly at random. For the randomized construction we have that $P[Z^*(I)[p] = \alpha] = \frac{1}{2^{n^*}}$, which yields the following expression.

$$E\left[d\left(S^*(I)[P_j], c(S^*(I)[P_j])\right)\right] = \ell_1 \cdot \sum_{\alpha \in \{0,1\}^{n^*}} \frac{1}{2^{n^*}} \cdot d^I_j(\alpha) \tag{11}$$

Combining Equations 10 and 11 yields the following bound.

$$\begin{aligned}
&\left|d\left(S^*_{det}(I)[P_j], c(S^*_{det}(I)[P_j])\right) - E[d(S^*(I)[P_j], c(S^*(I)[P_j]))]\right| \\
&\qquad = \ell_1 \cdot \left|\sum_{\alpha \in \{0,1\}^{n^*}} \left(P[Z^*(I)[p] = \alpha] - \frac{1}{2^{n^*}}\right) \cdot d^I_j(\alpha)\right| \qquad (12) \\
&\qquad \leq \ell_1 \cdot \sum_{\alpha \in \{0,1\}^{n^*}} \left|\left(P[Z^*(I)[p] = \alpha] - \frac{1}{2^{n^*}}\right)\right| \cdot n^* \\
&\qquad \leq \ell_1 \cdot \epsilon \cdot n^*
\end{aligned}$$

Summing Equation 12 over $0 \leq j < \kappa_L$ yields the desired bound for every $I \subseteq \{1, 2, \ldots, 2m\}$ of size n^*.

$$\left|d\left(S^*_{det}(I), c(S^*_{det}(I))\right) - E[d(S^*(I), c(S^*(I)))]\right| \leq \ell_1 \cdot \kappa_L \cdot \epsilon \cdot n^* \leq \ell_1 \cdot (\kappa_{yes'} - \kappa_{yes}) \tag{13}$$

Equation 13 allows us to finish the proof of Lemma 1. For any set S^* of size n^* that corresponds to a clique in G, we have that $E[d(S^*(I), c(S^*(I)))] = E_{yes} = \ell_1 \kappa_{yes}$, and so by Equation 13, $d\left(S^*_{det}(I), c(S^*_{det}(I))\right) \leq \ell_1 \kappa'_{yes} = D_{yes}$. For any set S^* of size n^* that does not correspond to a clique in G, we have that $E[d(S^*(I), c(S^*(I)))] \geq E_{no} = \ell_1 \kappa_{no}$, and so by Equation 13, $d\left(S^*_{det}(), c(S^*_{det}(I))\right) \geq \ell_1 \kappa'_{no} = D_{no}$. Since $\frac{D_{no}}{D_{yes}} \geq 1 + \delta$ for some δ depending only on k, the construction is an fpt-reduction from MCC to *Gap-Consensus String With Outliers*, completing the proof of Lemma 1. □

3 Hardness of Approximating Consensus Patterns

To show that *Consensus Patterns* does not admit an EPTAS we will first demonstrate hardness of a colored variant of *Gap-Consensus String with Outliers* where the input multiset S is partitioned into n^*

parts, that is $S = S_1 \uplus S_2 \uplus \ldots \uplus S_{n^*}$, and the solution set S^* is required to satisfy $s_i^* \in S_i$. We call this variant of the problem *Gap-Colored Consensus String with Outliers*.

In the proof of Lemma 1 we reduced from MCC to*Gap-Consensus String with Outliers*. The input set S of strings constructed by the reduction can be partitioned into n^* parts, $S = \bigcup_{p,q} S_{p,q}$ for p,$q \in \{1, \ldots, k\}$ and $p \neq q$, such that each string in $S_{p,q}$ corresponds to an edge endpoint (u, v) with $u \in V_p$ and $v \in V_q$. The images of "yes" instances of MCC have the property that any set S^* that corresponds to the edge endpoints of a clique satisfy $d(S^*, c(S^*)) \leq D_{yes}$, and $|S^* \cap S_{p,q}| = 1$ for every p,$q \leq k$ (and $p \neq q$). On the other hand in the images of "no" instances of MCC, every set S^* of size n^* satisfies $d(S^*, c(S^*)) \geq D_{no}$. If the construction in Section 2 outputs the partition $S = \bigcup_{p,q} S_{p,q}$ together with the set S, the reduction of Section 2 demonstrates the W[1]-hardness of *Gap-Colored Consensus String with Outliers*

Lemma 8. Gap-Colored Consensus String with Outliers *is W[1]-hard.*

The hardness result of Lemma 8 remains valid even if we demand that all of the (multi) sets S_i contain the same number of strings; if $|S_i| < |S_j|$ for some i, j we can make duplicates of strings in S_i until equality is obtained. To show that *Consensus Patterns* does not have an EPTAS unless $FPT = W[1]$ we introduce the following gap variant of the problem.

Gap-Consensus Patterns	
Input:	A set $S = \{s_1, \ldots, s_n\}$ of length-L strings over a constant size alphabet Σ together with an integer ℓ, where $\ell \leq L$, a rational ϵ and intgers D_{yes} and D_{no} with $D_{no} \geq D_{yes}(1 + \epsilon)$ such that the following holds. Either there is a length-ℓ substring t_i of each s_i in S such that $\sum_{\forall i} d(t_i, s) \leq D_{yes}$ or for every collection $t_1, \ldots t_n$ such that t_i is a length-ℓ substring s_i we have $\sum_{\forall i} d(t_i, s) \geq D_{yes}$.
Parameter:	$\lceil 1/\epsilon \rceil$
Question:	Is there a length-ℓ substring t_i of each s_i in S such that $\sum_{\forall i} d(t_i, s) \leq D_{yes}$?

We will now give a fpt-reduction from *Gap-Colored Consensus String with Outliers* to *gap-Consensus Patterns*. The main ingredient in our reduction is a gadget string w. The string w has length L_1 (to be determined later), and for every $i \geq 1$, $w[i] = 1$ if $i = j^2$ for an integer j and $w[i] = 0$ otherwise. We will say that an integer i is a *square* if $i = j^2$ for some integer j. Thus $w[i]$ is 1 if and only if i is a square.

Lemma 9. *For positive integers x, y and z such that $z \geq \frac{L_1}{4}$, $x < y$ and $y + z \leq L_1$ we have $d(w[\{x, x+1, \ldots, x+z\}], w[\{y, y+1, \ldots, y+z\}]) \geq \lfloor \frac{\sqrt{L_1}}{16} \rfloor$*

Proof. To lower bound $d(w[\{x, x+1, \ldots, x+z\}], w[\{y, y+1, \ldots, y+z\}])$ it is sufficient to find the number of values for i between 0 and z such that $w[x+i] = 1$ but $w[y+i] = 0$, that is $x+i$ is a square but $y+i$ is not. Let $i_1, i_2, \ldots i_t$ be all the values for i such that $x+i$ is square, in increasing order. We prove that if $y + i_j$ is square then $y + i_{j+1}$ is not. In particular, suppose $y + i_j$ is square. Let r_x and r_y be the integers such that $x + i_j = r_x^2$ and $y + i_j = r_y^2$. Since $x < y$ we have $r_x < r_y$. Furthermore, $x + i_{j+1} = (r_x + 1)^2$. Hence

$$y + i_{j+1} = r_y^2 + i_{j+1} - i_j = r_y^2 + ((r_x + 1)^2 - r_x^2) < r_y^2 + ((r_y + 1)^2 - r_y^2) = (r_y + 1)^2.$$

But then $y + i_{j+1}$ can't be square. It follows that there are at least $\lfloor \frac{t}{2} \rfloor$ values for i such that $x + i$ is a square but $y + i$ is not. It remains to lower bound t.

The gap between a square number i and the next square number i' is less than $2\sqrt{i'} \leq 2\sqrt{L_1}$. Thus the number of square numbers in $\{x, x+1, \ldots, x+z\}$ is at least $\frac{L_1}{4} \cdot \frac{1}{2\sqrt{L_1}} \geq \lfloor \frac{\sqrt{L_1}}{8} \rfloor$. Hence $d(w[\{x, x+1, \ldots, x+z\}], w[\{y, y+1, \ldots, y+z\}]) \geq \lfloor \frac{\sqrt{L_1}}{16} \rfloor$. □

Given an instance n^*, $S = S_1 \uplus S_2 \uplus \ldots \uplus S_{n^*}$ of *Gap-Colored Consensus String with Outliers* we construct an instance of *Gap-Cosensus Patterns* as follows. Let ℓ be the length of all the strings in S. We choose L_1 such that $\lfloor \frac{\sqrt{L_1}}{16} \rfloor > n^* \cdot \ell$ and construct a gadget string w of length L_1. For every $i \leq n^*$ we make a string $\hat{s}_i$ from the set S_i. Let $S_i = s_i^1, s_i^2, \ldots, s_i^t$. We define

$$\hat{s}_i = w \circ s_i^1 \circ w \circ s_i^2 \circ w \ldots \circ w \circ s_i^t.$$

and set $L = L_1 + \ell$. We keep the values of D_{yes} and D_{no}. This concludes the construction.

Lemma 10. *For every $S^* = \{s_1^*, \ldots, s_{n^*}^*\} \subset S$ such that $s_i^* \in S_i$ for all i there is a collection $T^* = t_1^*, \ldots t_{n^*}^*$ such that t_i^* is a length L substring of $\hat{s}_i$ and $d(c(T^*), T^*) \leq d(C(S^*), S^*)$.*

Proof. For every i, set $t_1^* = w \circ s_i^*$. Since $s_i^* \in S_i$ we have that t_1^* is a length L substring of $\hat{s}_i$. Set $c = w \circ c(S^*)$, we have that $d(c(T^*), T^*) \leq d(c, T^*) \leq d(C(S^*), S^*)$. □

Lemma 11. *For every collection $T^* = t_1^*, \ldots t_{n^*}^*$ such that t_i^* is a length L substring of $\hat{s}_i$ and $d(c(T^*), T^*) \leq n^* \cdot \ell$ there is a subset $S^* = \{s_1^*, \ldots, s_{n^*}^*\} \subset S$ such that $s_i^* \in S_i$ for all i and $d(C(S^*), S^*) \leq d(c(T^*), T^*)$.*

Proof. For every i we can decompose t_i^* into $t_i^* = w[\{a_i + 1, \ldots, L\}] \circ s_i^* \circ w[\{1, \ldots, a_i\}]$ for a non-negative integer $a_i \leq L$, where $s_i^* \in S_i$. If $a_i = 0$ then $t_i^* = w \circ s_i^*$ while $a_i = L$ gives $t_i^* = s_i^* \circ w$. Set $S^* = \{s_1^*, \ldots, s_{n^*}^*\}$. We need to show that $d(C(S^*), S^*) \leq d(c(T^*), T^*)$. It is sufficient to show that for every i, j we have $a_i = a_j$ because then all the s_i^*'s align in the decomposition of the t_i^*'s and so $d(C(S^*), S^*) \leq d(c(T^*), T^*)$ holds.

We prove that if $a_i \neq a_j$ for some i, j then $d(c(T^*), T^*) \geq d(t_i^*, t_j^*) > d(t_i^*, t_j^*) > n^* \cdot \ell$, contradicting the assumption of the lemma. If $a_i \neq a_j$, without loss of generality $a_i < a_j$. Then we can decompose $t_i^* = w_i^1 \circ z_i \circ w_i^2 \circ s_i^* \circ w_i^3$ and $t_j^* = w_j^1 \circ s_j \circ w_j^2 \circ s_j^* \circ w_j^3$ such that the following properties hold. The lengths of w_i^1, w_i^2 and w_i^3 equals the lengths of w_j^1, w_j^2 and w_j^3 respectively, z_i and z_j both have length ℓ, and w_i^1, w_i^2, w_i^3, w_j^1, w_j^2, w_j^3 are all substrings of w. Since $\ell \leq \frac{L_1}{4}$ we have that one of w_i^1, w_i^2, w_i^3 have length at least $\frac{L_1}{4}$. Without loss of generality this is w_i^1. We have that $d(t_i^*, t_j^*) \geq d(w_i^1, w_j^1)$. Furthermore, since $a_i \neq a_j$ we have $w_i^1 \neq w_j^1$ and hence by Lemma 9 it follows that $d(w_i^1, w_j^1) \geq \lfloor \frac{\sqrt{L_1}}{16} \rfloor > n^* \cdot \ell$. But this implies that $d(t_i^*, t_j^*) > n^* \cdot \ell$. By the triangle inequality we have $d(c(T^*), T^*) \geq d(t_i^*, t_j^*) > n^* \cdot \ell$ yielding the desired contradiction. □

The construction, together with Lemmata 8, 10 and 11 yield the following result.

Lemma 12. Gap-Consensus Patterns *is W[1]-hard.*

Since an EPTAS for *Consensus Patterns* could be used to solve *Gap-Consensus Patterns* in time $f(\epsilon)(nL)^{O(1)}$, Lemma 12 implies our main result.

Theorem 3. Consensus Patterns *does not have an EPTAS unless FPT=W[1].*

4 Conclusions and Future Work

We have shown that two stringology problems with applications in computational biology, *Consensus Patterns* and *Consensus String With Outliers*, do not admit EPTASs unless FPT=W[1]. Our results rule out the possibility of a $(1+\epsilon)$ approximation algorithms with running time $f(1/\epsilon)n^{O(1)}$, while the best PTASes for the two problems have running time $n^{O(1/\epsilon^4)}$ (*Consensus Patterns*) and $n^{O(1/\epsilon^2)}$ (*Consensus String With Outliers*) respectively. Hence there is still a significant gap between known upper and lower bounds, and obtaining tighter bounds warrants further investigation.

References

[1] N. Alon, O. Goldreich, J. Håstad and R. Peralta, Simple Construction of Almost k-wise Independent Random Variables. *Random Struct. Algor.*, 3(3): 289–304, 1992.

[2] S. Arora. Polynomial Time Approximation Schemes for Euclidean TSP and Other Geometric Problems. *Proc of 37th FOCS*, pages 2-11, 1996.

[3] S. Arora, Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and other Geometric Problems. *J. ACM*, 45, 5:753–782, 1998.

[4] B. Brejová, D.G. Brown, I.M. Harrower, and T. Vinar. New Bounds for Motif Finding in Strong Instances. *Proc. of 17th CPM*, pages 94–105, 2006.

[5] B. Brejová, D.G. Brown, I.M. Harrower, A. López-Ortiz and T. Vinar. Sharper Upper and Lower Bounds for an Approximation Scheme for Consensus-Pattern. *Proc. of 16th CPM*, pages 1–10, 2005.

[6] C. Lo, B. Kakaradov, D. Lokshtanov, and C. Boucher. SeeSite: Efficiently Finding Co-occurring Splice Sites and Exon Splicing Enhancers. arXiv:1206.5846v1.

[7] R.G. Downey, and M.R. Fellows. Parameterized complexity. *Springer*, 1999.

[8] M.R. Fellows, J. Gramm, and R. Niedermeier. On the parameterized intractability of motif search problems. *Combinatorica*, 26:141–167, 2006.

[9] M.R. Fellows, D. Hermelin, F.A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009

[10] J. Flum, and M. Grohe. Parameterized Complexity Theory. *Springer-Verlag*, 2006.

[11] F. Grimmett, and D. Stirzaker. Probability and random processes. *Oxford University Press*, 3 edition, 2001.

[12] W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *J. Amer. Statistical Assoc.*, 58(301): 13–30, 1963.

[13] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, S.S. Ravi, D.J. Rosenkrantz, and R.E. Stearns, NC-Approximation Schemes for NP- and PSPACE-Hard Problems for Geometric Graphs. *J. Algorithms*, 26(2):238–274, 1998

[14] J.K. Lanctot, M. Li, B. Ma, S. Wang, and L. Zhang. Distinguishing string selection problems. *Inform. Comput.*, 185(1):41–55, 2003. Preliminary version appeared in Proc. 10th SODA, pages 41-55, 1999.

[15] M. Li, B. Ma, and L. Wang. Finding similar regions in many sequences. *J. Comput. System Sci.*, 65(1):73–96, 2002.

[16] D. Marx. Closest Substring Problems with Small Distances. *SIAM J. Comput.*, 38(4):1283–1410, 2008.

[17] D. Marx. Efficient Approximation Schemes for Geometric Problems? *Proc. of 13th ESA*, 51(1): 448–459, 2005.

[18] D. Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1): 60–78, 2008.

[19] J. Naor and M. Naor. Small-Bias Probability Spaces: Efficient Constructions and Applications. *SIAM J. Comput.*, 22(4): 838–856, 1993.

[20] R. Niedermeier. Invitation to Fixed-Parameter Algorithms. *Oxford University Press*, 2006.

[21] P. Pevzner and S. Sze. Combinatorial approaches to finding subtle signals in DNA strings. In *Proc. of the 8th ISMB*, pages 269–278, 2000.