The Graph Motif problem parameterized by the structure of the input graph

Édouard Bonnet[1*] **and Florian Sikora**[2]

1 **Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary**
bonnet.edouard@sztaki.mta.hu

2 **PSL, Université Paris-Dauphine, LAMSADE UMR CNRS 7243, France**
florian.sikora@dauphine.fr

Abstract

The Graph Motif problem was introduced in 2006 in the context of biological networks. It consists of deciding whether or not a multiset of colors occurs in a connected subgraph of a vertex-colored graph. Graph Motif has been analyzed from the standpoint of parameterized complexity. The main parameters which came into consideration were the size of the multiset and the number of colors. Though, in the many applications of Graph Motif, the input graph originates from real-life and has structure. Motivated by this prosaic observation, we systematically study its complexity relatively to graph structural parameters. For a wide range of parameters, we give new or improved FPT algorithms, or show that the problem remains intractable. Interestingly, we establish that Graph Motif is W[1]-hard (while in W[P]) for parameter max leaf number, which is, to the best of our knowledge, the first problem to behave this way.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Parameterized Complexity, Structural Parameters, Graph Motif, Computational Biology

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

The Graph Motif problem has received a lot of attention during the last decade. Informally, Graph Motif is defined as follows: given a graph with arbitrary colors on the nodes and a multiset of colors called the motif, the goal is to decide if there exists a subset of vertices of the graph such that (1) the subgraph induced by this subset is connected and (2) the colors on the subset of vertices match the motif, i.e. each color appears the same number of times as in the motif. Originally, this problem is motivated by applications in biological network analysis [27]. However, it proves useful in social or technical networks [4] or in the context of mass spectrometry [8].

Studying biological networks allows a better characterization of species, by determining small recurring subnetworks, often called *motifs*. Such motifs can correspond to a set of nodes realizing some function, which may have been evolutionary preserved. Thus, it is crucial to determine these motifs to identify common elements between species and transfer the biological knowledge. Graph Motif corresponds to topology-free queries and can be

* This work is partially supported by ERC Starting Grant PARAMTIGHT (n. 280152).

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–20

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

seen as a variant of a graph pattern matching problem with the sole topological requirement of connectedness. Such queries were also studied extensively for sequences during the last thirty years, and with the increase of knowledge about biological networks, it is relevant to extend these queries to networks [33].

2 Preliminaries and previous work

For any two integers $x < y$, we set $[x, y] := \{x, x+1, \ldots, y-1, y\}$, and for any positive integer x, $[x] := [1, x]$. If $G = (V, E)$ is a graph and $S \subseteq V$ a subset of vertices, $G[S]$ denotes the subgraph of G induced by S. For a vertex $v \in V$, the set of neighbors of v in G is denoted by $N_G(v)$, or simply $N(v)$, and $N_G(S) := (\bigcup_{v \in S} N(v)) \setminus S$ and will often be written just $N(S)$. We define $N[v] := N(v) \cup \{v\}$ and $N[S] := N(S) \cup S$. We say that a vertex v *dominates* a set of vertices S if $S \subseteq N[v]$. A set of vertices R *dominates* another set of vertices S if $S \subseteq N[R]$. If $G = (V, E)$ is a graph and $V' \subseteq V$, $G - V'$ denotes the graph $G[V \setminus V']$. A *universal vertex* v, in a graph $G = (V, E)$, is such that $N_G[v] = V$. A *matching* of a graph is a mutually disjoint set of edges. In an explicitly bipartite graph $G = (V_1 \cup V_2, E)$, we call a matching of size $\min(|V_1|, |V_2|)$ a *perfect matching*. A *cluster graph* (or simply, *cluster*) is a disjoint union of cliques. A *co-cluster graph* (or, *co-cluster*) is the complement graph of a cluster graph. If $\mathcal{C}$ is a class of graphs, the *distance to* $\mathcal{C}$ of a graph G is the minimum number of vertices to remove from G to get a graph in $\mathcal{C}$.

If $f : A \to B$ is a function and $A' \subseteq A$, $f_{|A'}$ denotes the restriction of f to A', that is $f_{|A'} : A' \to B$ such that $\forall x \in A'$, $f_{|A'}(x) := f(x)$. Similarly, if E is a set of edges on vertices of V and $V' \subseteq V$, $E_{|V'}$ is the subset of edges of E having both endpoints in V'.

Graph Motif and multisets. Graph Motif is defined as follows:

Graph Motif

- **Input**: A triple (G, c, M), where $G = (V, E)$ is a graph, $c : V \to \mathcal{C}$ gives some color of $|\mathcal{C}|$ to the vertices, and M is a multiset of colors of $\mathcal{C}$.
- **Output**: A subset $P \subseteq V$ such that (1) $G[P]$ is connected and (2) $c(P) = M$.

We will refer to condition (1) as the *connectivity constraint* and to condition (2) as the *multiset constraint*. For convenience, if $S \subseteq V$, $c(S)$ will denote the multiset of colors of vertices in S.

The *multiplicity* of element x in multiset M, denoted by $m_M(x)$ is the number of occurences of x in M. The cardinality of a multiset M denoted by $|M|$ is its number of elements *with their multiplicity*: $\Sigma_{x \in M} m_M(x)$. If M and N are two multisets, $M \cup N$ is the multiset A such that $\forall x$, $m_A(x) = m_M(x) + m_N(x)$, and $M \setminus N$ is the multiset D such that $\forall x \in M$, $m_D(x) = \max(0, m_M(x) - m_N(x))$ (and $\forall x \notin M$, $m_D(x) = 0$). We write $M \subseteq N$ iff $M \setminus N = \emptyset$ and $M \subset N$ iff $M \subseteq N$ and $M \neq N$.

▶ **Example 1.** Let $M = \{1, 2, 2, 4, 5, 5, 5\}$ and $N = \{1, 1, 1, 2, 2, 3, 3, 4, 5, 5, 5, 5\}$. $|M| = 7$, $|N| = 12$, $M \setminus N = \emptyset$, $N \setminus M = \{1, 1, 3, 3, 5\}$, and $M \subseteq N$.

Parameterized Complexity and ETH. A parameterized problem (I, k) is said *fixed-parameter tractable* (or in the class FPT) w.r.t. (with respect to) parameter k if it can be solved in $f(k) \cdot |I|^c$ time (in *fpt-time*), where f is any computable function and c is a constant (see [31, 16] for more details about fixed-parameter tractability). The parameterized complexity hierarchy is composed of the classes $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{W}[\mathsf{P}] \subseteq \mathsf{XP}$. The class XP contains problems solvable in time $|I|^{f(k)}$, where f is an unrestricted function.

A powerful technique to design parameterized algorithms is *kernelization*. In short, kernelization is a polynomial-time self-reduction algorithm that takes an instance (I, k) of a parameterized problem P as input and computes an equivalent instance (I', k') of P such that $|I'| \leqslant h(k)$ for some computable function h and $k' \leqslant k$. The instance (I', k') is called a *kernel* in this case. If the function h is polynomial, we say that (I', k') is a polynomial kernel.

The *Exponential Time Hypothesis* (ETH) is a conjecture by Impagliazzo et al. [23] asserting that there is no $2^{o(n)}$-time algorithm for 3-SAT on instances with n variables. The so-called sparsification lemma, also proved in [23], shows that if ETH turns out to be true, then there is no $2^{o(n+m)}$-time algorithm solving 3-SAT where m is the number of clauses.

Previous work. Many results about the complexity of Graph Motif are known. The problem is NP-hard even with strong restrictions. For instance, it remains NP-hard for bipartite graphs of maximum degree 4 and motifs containing two colors only [17], or for trees of maximum degree 3 and when the motif is colorful (that is, no color occurs more than once) [17], or for rooted trees of depth 2 [2]. However, the problem is solvable in polynomial time when the graph is a caterpillar [2], or when both the number of colors in the motif and the treewidth of the graph are bounded by a constant [17].

As Graph Motif is intractable even for very restricted classes of graphs, and considering that, in practice, the motif is supposed to be small compared to the graph, the parameterized complexity of Graph Motif relatively to the size of the motif has been tackled. It is indeed in FPT when parameterized by the size of the motif. At least seven different papers gave an FPT algorithm [17, 4, 22, 26, 5, 33, 32]. The best (randomized) algorithm runs in time $O^*(2^k)$ where the O^* notation suppresses polynomial factors [5, 33] and works well in practice for small values of k, even with hundreds of millions of edges [6]. The current best deterministic algorithm takes time $O^*(5.22^k)$ [32]. However, an algorithm running in time $O^*((2-\epsilon)^k)$ would break the 2^n barrier in solving Set Cover instances with n elements [5]. Besides, it is unlikely that Graph Motif admits a polynomial kernel, even on a restricted class of trees [2]. Ganian also proved that the problem is in FPT when the parameter is the size of a minimum vertex cover of the graph [19]. Actually, his algorithm is given for a smaller parameter called twin-cover. Ganian also show that Graph Motif can be solved in $O^*(2^k)$ for graphs with neighborhood diversity k [20]. On the negative side, the problem is W[1]-hard relatively to the number of colors, even for trees [17]. To deal with the huge rate of noise in the biological data, many variants of the problem has been introduced. For example, the approach of Dondi *et al.* requires a solution with a minimum number of connected components [15], while the one of Betzler *et al.* asks for a 2-connected solution [4]. In other variants stemming purely from bio-informatics, some colors can be added to, substituted or subtracted from the solution [10, 15].

In light of the previous paragraphs, it is clear that the complexity of Graph Motif is well known for different versions and constraints on the problem itself. However, only few works take into account the structure of the input graph. We believe that this an interesting direction since Graph Motif has applications in real-life problems, where the input is not random. For example, some biological networks have been shown scale-free or with small diameter [1]. We will therefore introduce a systematic study with respect to structural graph parameters [25, 18]. We believe that this is also of theoretical interest, to understand how a given parameter influences the complexity of the problem.

Our contribution. In Section 3, we improve the known FPT algorithms with parameter distance to clique, vertex cover number, and edge clique cover number. We also give a parameterized algorithm for the parameter distance to co-cluster which nicely reuses the

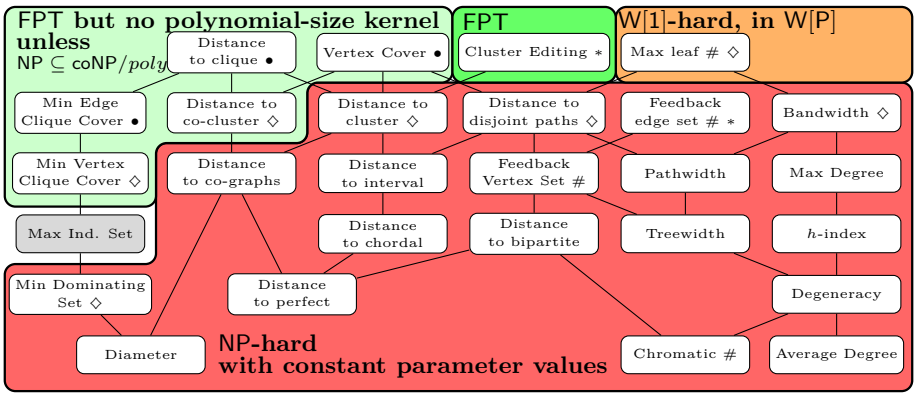

Figure 1 Hasse diagram of the relationship between different parameters ([25]). Two parameters are connected by a line if the parameter below can be polynomially upper-bounded in the parameter above. For example, *vertex cover* is above *distance to disjoint paths* since deleting a vertex cover produces an independent set, hence a set of disjoint paths. Therefore, positive results propagate upwards, while negative results propagate downwards. Results marked by ♢ are obtained in this paper, those marked with • are improvement of existing results, and those marked with ∗ are corollaries of existing results.

FPT algorithms for both vertex cover number and distance to clique and another algorithm for parameter vertex clique cover number. These last two algorithms are noteworthy since a bounded distance to co-cluster or a bounded vertex clique cover number do not imply a bounded neighborhood diversity, a parameter for which Graph Motif was already known to be in FPT. We also show that a polynomial kernel for the aforementioned parameters is unlikely. In Section 4, we show that Graph Motif remains hard on graphs of constant distance to disjoint paths, or constant bandwidth, or constant distance to cluster, or constant dominating set number. More surprisingly, we establish that Graph Motif is W[1]-hard (but in W[P]) for the parameter max leaf number. To the best of our knowledge, there is no previously known problem behaving similarly when parameterized by max leaf number. Indeed, graphs with bounded max leaf number are really simple and, for instance, all the problems studied in [18] are FPT for this parameter. These positive and negative results draw a tight line between tractability and intractability (see Figure 1). Due to space constraints, some proofs (marked with ★) are deferred to the appendix.

3 FPT algorithms and lower bound in the size of kernels

In this section, we improve or establish new FPT algorithms for several parameters. We also give a lower bound on the size of the kernel for all those parameters except *cluster editing number*. Figure 1 summarizes those results.

3.1 Cluster editing and linear neighborhood diversity

The cluster editing number of a graph is the number of edge deletions or additions required to get a cluster graph. It can be computed in time $O^*(1.62^k)$ [7]. We will use a known result involving another parameter called neighborhood diversity introduced by Lampis [28]. A graph has neighborhood diversity k if there is a partition of its vertices into at most k sets such that all the vertices in each set *have the same type*. And, two vertices u and v *have*

the same type iff $N(v) \setminus \{u\} = N(u) \setminus \{v\}$. We say that a graph parameter κ has *linear* (resp. *exponential*) *neighborhood diversity* if, for every positive integer k, all the graphs G such that $\kappa(G) \leqslant k$ have neighborhood diversity ck (resp. c^k) for some constant c. We say that a parameter κ has *unbounded neighborhood diversity*, if there is *no* function f such that all graphs G with $\kappa(G) \leqslant k$ have neighborhood diversity $f(k)$.

▶ **Theorem 2** ([20]). Graph Motif *can be solved in* $O^*(2^k)$ *on graphs with neighborhood diversity* k.

The following result is a direct consequence of the fact that, restricted to connected graphs, cluster editing has linear neighborhood diversity.

▶ **Corollary 3.** Graph Motif *can be solved in* $O^*(8^k)$*, where* k *is the cluster editing number.*

Proof. Let $(G = (V, E), c, M)$ be any instance of Graph Motif. We can assume that G is connected, otherwise we run the algorithm in each connected component of G. Let X be the set of vertices which are an endpoint of an edited edge (deleted or added) and let G' be the cluster graph obtained by the k edge editions. We may observe that $|X| \leqslant 2k$ and that the number of maximal cliques $C_1, \ldots, C_l$ in G' is bounded by k (otherwise, G could not be connected). For each $i \in [l]$, and for each vertex $v \in C_i \setminus X$, $N[x] = C_i$. Thus the neighborhood diversity of G is bounded by $|X| + l \leqslant 2k + k = 3k$. So, we can run the algorithm for bounded neighborhood diversity [20] and it takes time $O^*(2^{3k})$. ◀

3.2 Parameters with exponential neighborhood diversity

The next three parameters that we consider are *distance to clique*, *size of a minimum vertex cover*, and *size of a minimum edge clique cover*. For the first two, a value of k entails that the neighborhood diversity is at most $k + 2^k$; and neighborhood diversity 2^k for the third one. Therefore, Ganian has already given an algorithm running in double exponential time for these parameters ($O^*(2^{k+2^k})$ or $O^*(2^{2^k})$, see Theorem 2, [19, 20]). We improve this bound to single exponential time $2^{O(k)}$ (more precisely $O^*(4^k)$) for distance to clique and to $2^{O(k \log k)}$ for the vertex cover and edge clique cover numbers. The latter running time is sometimes called *slightly superexponential* FPT time [29]. Then, we prove that for each of those three parameters, a polynomial kernel is unlikely.

As a preparatory lemma for the algorithm parameterized by distance to clique, we show that a variant of Set Cover with thresholds is solvable in time $O^*(2^n)$, where n is the size of the universe. In the problem that we call here Colored Set Cover with Thresholds, one is given a triple $(\mathcal{U}, \mathcal{S} = \mathcal{C}_1 \uplus \ldots \uplus \mathcal{C}_l, (a_1, \ldots, a_l))$ where $\mathcal{U}$ is a ground set of n elements, $\mathcal{S}$ is a set of subsets of $\mathcal{U}$ partitioned into l classes called *colors* and $(a_1, \ldots, a_l)$ is a tuple of l positive integers called *threshold vector*. The goal is to find a set cover $\mathcal{T} \subseteq \mathcal{S}$ (not necessarily minimum) such that for each $i \in [l]$, the number of sets with color i (that is, in $\mathcal{C}_i$) in $\mathcal{T}$ is at most a_i.

▶ **Lemma 4.** Colored Set Cover with Thresholds *with* n *elements and* m *sets can be solved in time* $O(nm2^n + nm)$.

Proof. We order the sets of $\mathcal{S}$ such that sets of the same color appear consecutively, say, first the sets of $\mathcal{C}_1$, then the sets of $\mathcal{C}_2$, and so on. The order within the sets of a same color is not important and is chosen arbitrarily. We denote the sets resultantly ordered by $S_1, \ldots, S_m$ and function c maps the index of a set to its color. Therefore, $c(j) = i$ means that set S_j has color i ($S_j \in C_i$). We fill by dynamic programming the table T, where $T[U, j]$ is meant to

contain the minimum number of sets in $\mathcal{C}_{c(j)}$ among any subset of $\{S_1, \ldots, S_j\}$ that covers $U \subseteq \mathcal{U}$ and respects the threshold vector.

As an initialization step, for each $U \subseteq \mathcal{U}$, we set $T[U,1] = 1$ if $U \subseteq S_1$, and $T[U,1] = \infty$ otherwise. For each $j \in [2,m]$, assuming that $T[U', j-1]$ was already filled for every $U' \subseteq \mathcal{U}$, we distinguish two cases to fill $T[U,j]$. If S_j is the first set of the color class $\mathcal{C}_{c(j)}$ then:

$$T[U,j] = \begin{cases} 0 \text{ if } T[U,j-1] < \infty & (* \text{ discard } S_j *) \\ 1 \text{ if } T[U,j-1] = \infty \text{ and } T[U \setminus S_j, j-1] < \infty & (* \text{ add } S_j *) \\ \infty \text{ otherwise} \end{cases}$$

Otherwise S_j is not the first set in $\mathcal{C}_{c(j)}$ and:

$$T[U,j] = \min \begin{cases} T[U,j-1] & (* \text{ discard } S_j *) \\ v+1 \text{ if } v < a_{c(j)} \text{ and } \infty \text{ otherwise} & (* \text{ add } S_j *) \end{cases}$$

with $v = T[U \setminus S_j, j-1]$.

A standard induction shows that the instance is positive iff $T[\mathcal{U}, m] \neq \infty$. The only costly operation in filling one entry of table T is the set difference which can be done in $O(n)$. If we want to produce an actual solution (and not solely decide the problem), we can add one bit in each entry $T[U,j]$ signaling whether or not S_j should be taken. Should the instance be positive, it then takes time $O(nm)$ to reconstruct a solution from a filled table T. Therefore, the running time is $O(n|T| + nm) = O(nm2^n + nm)$. ◀

▶ **Theorem 5.** Graph Motif *can be solved in $O^*(4^k)$, where k is the distance to clique.*

Proof. Let $(G = (V,E), c : V \to \mathcal{C}, M)$ be any instance of Graph Motif and assume R is a solution, that is $G[R]$ is connected and $c(R) = M$. If there is no solution, our algorithm will detect it eventually. We first compute a set $S \subseteq V$ of size k such that $C := V \setminus S$ is a clique. This can be done in time $O^*(2^k)$ by branching over the two endpoints of a *non-edge*, or even in $O^*(1.2738^k)$ by applying the state-of-the-art algorithm for Vertex Cover on the complementary graph [12]. Running through all the 2^k subsets of S, one can guess the subset $S' = R \cap S$ of S which is in the solution R, and $S_1, S_2, \ldots, S_{k'}$ be the $k' \leqslant k$ connected components of $G[S']$. It must hold that $c(S') \subseteq M$, otherwise R would not be a solution. Now, the problem boils down to finding a non-empty (an empty subset would mean that $S' = R$ which can be easily checked) subset $C' \subseteq C$ such that $G[S' \cup C']$ is connected and $c(C') \subseteq M \setminus c(S')$. Then, the set $S' \cup C'$ can be extended into a solution by adding vertices of $C \setminus C'$ with the right colors. The graph $G[S' \cup C']$ is connected iff each connected component S_j of $G[S']$ has at least one neighbor in $N(C')$. We build an equivalent instance of Colored Set Cover with Thresholds in the following way. The ground set $\mathcal{U}$ is of size k' with one element x_j per connected component S_j of $G[S']$. For each vertex v in C colored by i, there is a set S_v colored by i such that $x_j \in S_v$ iff $N(v) \cap S_j \neq \emptyset$. For each color i, the threshold a_i is set to the multiplicity of i in $M \setminus c(S')$. If there are more than one set with the same color and the same elements, we keep only one copy of this colored set. The number of sets is therefore at most $2^{k'}|\mathcal{C}|$. So, it takes time $O(k'2^{k'}|\mathcal{C}|(2^{k'}+1)) = O^*(4^{k'})$ to solve this instance, hence an overall worst case running time of $O^*(4^k)$. ◀

▶ **Theorem 6.** Graph Motif *can be solved in $O^*(2^{2k \log k})$ on graphs with a vertex cover of size k.*

Proof. We start similarly to the previous algorithm. We compute a minimum vertex cover S of G in time $O^*(2^k)$ (or $O^*(1.2738^k)$ [12]), and then guess in time $O^*(2^k)$ the subset $S' = S \cap R$, where R is a fixed solution. Again, we denote by $S_1, S_2, \ldots, S_{k'}$ the connected components of $G[S']$. We remove $c(S')$ from the motif and we remove from V the set I' of the vertices of the independent set $I := V \setminus C$ which have no neighbor in S'. Now,

by the transformation presented in the algorithm parameterized by distance to clique, the problem could be made equivalent to a constrained version of Colored Set Cover with Thresholds where the intersection graph (with an edge between two sets if they have a non-empty intersection) of the solution has to be connected. Unfortunately, it is not clear whether or not this variant can be solved in time $2^{O(n)}$. Thus, at this point, we have to do something different.

Let $R_d = \{r_1, r_2, \ldots, r_l\} \subseteq R \setminus S'$ be a minimal (inclusion-wise) set of vertices such that $G[S' \cup R_d]$ is connected. We can observe that $l \leqslant k' \leqslant k$. We guess in time $O^*(l!B_l)$ (where B_l is the l-th Bell number, i.e., the number of partitions of a set of size l) an ordered partition $P := \langle A_1, A_2, \ldots, A_l \rangle$ of the connected components $\{S_1, \ldots, S_{k'}\}$ such that, for each $i \in [l]$, (1) r_i has at least one neighbor in each connected component of A_i and (2) if $i \geqslant 2$, r_i has at least one neighbor in a connected component of $\bigcup_{1 \leqslant j < i} A_j$. Note that such an ordered partition always exists since $G[S' \cup R_d]$ is connected. Now, we build the bipartite graph $B = (P \cup M', F)$, where $M' = M \setminus c(S')$ and there is an edge between $A_i \in P$ and each copy of color $c \in M'$ iff there is a vertex $v \in I$ colored by c in the original graph G and such that (1) v has at least one neighbor in each connected component of A_i and (2) if $i \geqslant 2$, v has at least one neighbor in a connected component of $\bigcup_{1 \leqslant j < i} A_j$. By construction, $\{\{A_i, c(r_i)\} \mid i \in [l]\}$ is a maximum matching of size $|P| = l$ in graph B. Thus, we compute in polynomial time a maximum matching $\{\{A_i, c_i\} \mid i \in [l]\}$ in B. Then, we obtain a solution to the Graph Motif instance by taking, for each $i \in |l]$ any vertex v_i colored by c_i and having (1) at least one neighbor in each connected component of A_i and (2) if $i \geqslant 2$, at least one neighbor in a connected component of $\bigcup_{1 \leqslant j < i} A_j$. This can also be done in polynomial time and the existence of such a v_i is guaranteed by the construction of graph B. Then, we complete set $S' \cup \bigcup_{i \in [l]} \{v_i\}$ into a solution by taking any vertices in $I \setminus I'$ with the right colors. As $l! \leqslant l^l$, $B_l \leqslant (\frac{l}{2})^l$ (even $B_l < (\frac{0.792l}{\ln(l+1)})^l$ [3]), and $l \leqslant k$ the overall running time is $O^*(2^k + 2^k k! B_k) = O^*(k^k k^k) = O^*(2^{2k \log k})$. ◀

In the Edge Clique Cover problem, one asks, given a graph $G = (V, E)$ and an integer k, for k subsets $C_1, \ldots, C_k \subseteq V$, such that $\forall i \in [k]$, $G[C_i]$ is a clique, and $\forall e \in E$, e lies in a clique C_i for some $i \in [k]$. The set $\{C_1, \ldots, C_k\}$ is called an *edge clique cover* of G. The *edge clique cover number* of a graph G is the smallest k such that G has an edge clique cover of size k. Edge Clique Cover admits a kernel of size 2^k [21] and, as observed in [13], it can be solved by dynamic programming in time $2^{O(n+m)}$. Therefore, it can be solved in time $2^{O(2^k + 2^{2k})}$, that is $2^{2^{O(k)}}$. On the negative side, Edge Clique Cover cannot be solved in time $2^{2^{o(k)}}$ under ETH [13]. Thus, the algorithm of Ganian [20] is essentially optimal if the edge clique cover is not given. But, we may imagine that the instance comes with an optimal or close to optimal edge clique cover, or that we have a good heuristic to compute it (a polynomial time approximation with sufficiently good ratio is unlikely [30]).

▶ **Theorem 7.** Graph Motif *can be solved in time* $2^{2^{O(k)}}$*, where* k *is the edge clique cover number, and in time* $O^*(2^{2k \log k + k})$ *if an edge clique cover of size* k *is given as part of the input.*

Proof. Let $(G = (V, E), c, M)$ be any instance of Graph Motif. If not given, we first compute an edge clique cover $\{C_1, \ldots, C_k\}$ of size k in G, in time $2^{2^{O(k)}}$ [21]. We guess in time $O^*(2^k)$ the exact subset $\{C'_1, \ldots, C'_{k'}\} \subseteq \{C_1, \ldots, C_k\}$ of cliques C_i such that $C_i \cap R$ is non-empty, for a fixed solution R. Now, we turn the instance into an equivalent instance where the motif has size $|M| + k'$ and the graph has at most $|V| + k'$ vertices and a vertex cover of size k'. The new graph is a bipartite graph $B = (A \cup W, F)$ such that A contains one vertex $v(C'_i)$ per clique C'_i (so, A is a vertex cover of graph B of size $k' \leqslant k$), $W = C'_1 \cup \ldots \cup C'_{k'} \subseteq V$,

and there is an edge in F between $v(C'_i) \in A$ and $w \in W$ iff $w \in C'_i$. Each vertex in W keeps the color it had in G. A fresh color c is given to the k' vertices of A, and color c is added to the motif M with multiplicity k'. Then, we run the algorithm parameterized by the vertex cover number of Theorem 6. This algorithm has an overall running time of $O^*(2^k 2^{2k \log k})$, if the edge clique cover is given, and $2^{2^{O(k)}}$ otherwise. ◀

Ganian [19], Theorem 6 and Theorem 5 prove that Graph Motif is in FPT if the parameter is the vertex cover number or the distance to clique. Therefore, the problem has a kernel [31]. Though, the size of this kernel is *a priori* not known. We show that the corresponding kernels cannot be polynomial unless NP $\subseteq$ coNP/*poly*.

▶ **Theorem 8.** *Unless* NP $\subseteq$ coNP/*poly*, Graph Motif *has no polynomial kernel when parameterized by the vertex cover number or the distance to clique, even for (i) motifs with only 3 colors and (ii) when the motif is colorful.*

Proof. We only give the proof for (i). The second item (ii) can be proven similarly following the ideas of [5]. We will define an OR-cross-composition [9] from the NP-complete X3C problem, stated as follows: given and integer q, a set $X = \{x_1, x_2, \ldots, x_{3q}\}$ and a collection $\mathcal{S} = \{S_1, \ldots, S_{|\mathcal{S}|}\}$ of 3-elements subsets of X, the goal is to decide if $\mathcal{S}$ contains a sub-collection $\mathcal{T} \subseteq \mathcal{S}$ such that $|\mathcal{T}| = q$ and each element of X occurs in exactly one element of $\mathcal{T}$. Given t instances, $(X_1, \mathcal{S}_1), (X_2, \mathcal{S}_2), \ldots, (X_t, \mathcal{S}_t)$, of X3C, we define our equivalence relation $\mathcal{R}$ such that any strings that are not encoding valid instances are equivalent, and $(X_i, \mathcal{S}_i), (X_j, \mathcal{S}_j)$ are equivalent iff $|X_i| = |X_j|$ and $|\mathcal{S}_i| = |\mathcal{S}_j|$. Hereafter, we assume that $X_i = [3q]$ and $\mathcal{S}_i = \{S_1, \ldots, S_{|\mathcal{S}_i|}\}$, for any $i \in [t]$. We will build an instance (G, c, M) of Graph Motif parameterized by the vertex cover or the distance to clique, where G is the input graph, c the coloring function and M the motif, such that there is a solution for Graph Motif iff there is an $i \in [t]$ such that there is a solution for $(X_i, \mathcal{S}_i)$. We will now describe how to build such instance of Graph Motif. The graph G consists of t nodes $r_1, r_2, \cdots, r_t$ forming a clique. There are also $O((3q)^3)$ nodes $s_{x,y,z}, 1 \leqslant x < y < z \leqslant 3q$, with an edge between r_i and $s_{x,y,z}$ iff the 3-element subset $\{x, y, z\}$ exists in $\mathcal{S}_i$. Finally, there are $3q$ nodes $x_i, 1 \leqslant i \leqslant 3q$, and there is an edge between x_i and every subset $s_{x,y,z}$ where x_i occurs (see also Figure 2). The coloration is $c(r_i) = 1$, for all $1 \leqslant i \leqslant t$, $c(s_{x,y,z}) = 2$ for all $1 \leqslant x < y < z \leqslant 3q$, and $c(x_i) = 3, 1 \leqslant i \leqslant 3q$. The multiset M consists of 1 occurrence of the color 1, q occurrences of color 2 and $3q$ occurrences of color 3.

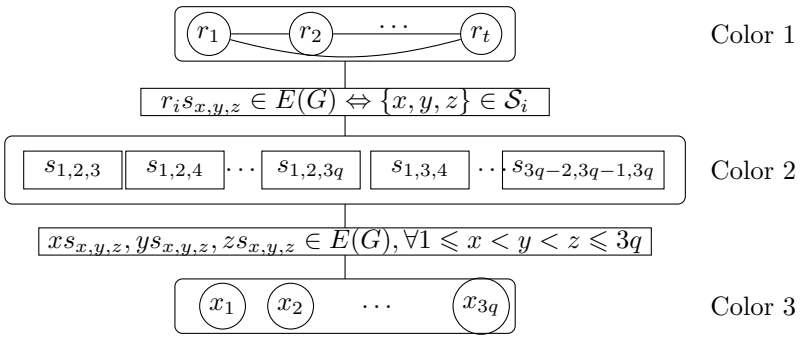

Figure 2 Illustration of the construction of G. The motif consists of 1 occurrence of color 1, q of color 2 and $3q$ of color 3.

It is easy to see that $\{s_{x,y,z} | 1 \leqslant x < y < z \leqslant 3q\} \cup \{x_i | 1 \leqslant i \leqslant 3q\}$ is a vertex cover for G and that its removal leaves only a clique, and that its size is polynomial in $3q$ and hence in the size of the largest instance.

Let us show that there is a solution for our instance of Graph Motif iff at least one of the $(X_i, \mathcal{S}_i)$'s has a solution of size q.

($\Leftarrow$) Suppose that $(X_i, \mathcal{S}_i)$ has a solution $\mathcal{T}_i$ of size q. We set $P = \{r_i\} \cup \{s_{x,y,z} \mid \{x,y,z\} \in \mathcal{T}_i\} \cup \{x_i | 1 \leqslant i \leqslant 3q\}$. One can easily check that $G[P]$ is connected and that $c(P) = M$.

($\Rightarrow$) Suppose that there is a solution $P \subseteq V$ such that $G[P]$ is connected and $c(P) = M$. Due to the motif, only one of the nodes r_i is in P and all nodes x_i are in P. We claim that there is then a solution $\mathcal{T}_i$ in $(X_i, \mathcal{S}_i)$, where i is the index of the only node r_i in P. We add in $\mathcal{T}_i$ the q sets $\{x,y,z\}$ such that $s_{x,y,z} \in P$. By the connectivity constraint, these sets all occurs in the instance i s.t. $r_i \in P$. Let us now prove that $\mathcal{T}_i$ covers exactly all the elements of X_i. Since P is a solution, the nodes $s_{x,y,z}$ in P correspond to a partition of X. Otherwise, one of the node x_i will not be connected. ◀

3.3 Parameters with unbounded neighborhood diversity

This section disproves the idea that Graph Motif is only tractable for classes with bounded neighborhood diversity. Indeed, we show that Graph Motif is in FPT parameterized by the size of a *vertex clique cover* or by the distance to co-cluster. The former algorithm creates a win/win based on König's theorem applied to a bounded number of auxiliary bipartite graphs. The latter is simpler and use as subroutines the algorithms parameterized by vertex cover number and distance to clique.

In the Vertex Clique Cover problem (also known as Clique Partition), one asks, given a graph $G = (V, E)$ and an integer k, for a *partition* of the vertices into k subsets $C_1, \ldots, C_k \subseteq V$, such that $\forall i \in [k]$, $G[C_i]$ is a clique. The set $\{C_1, \ldots, C_k\}$ is called an *vertex clique cover* of G. The *vertex clique cover number* of a graph G is the smallest k such that G has an vertex clique cover of size k. This problem is equivalent to the Graph Coloring problem since a graph as a vertex clique cover of size k iff its complement is k-colorable. Therefore, Vertex Clique Cover is unlikely to be in XP. However, if a vertex clique cover comes with the input, we show that Graph Motif is in FPT for parameter vertex clique cover number. One can notice that Graph Motif is NP-hard in 2-colorable graphs. This is a striking example of how easier can Graph Motif be on the denser counterpart of two complementary classes.

To realize that vertex clique cover number has unbounded neighborhood diversity, think of the complement of a bipartite graph. The vertex clique cover is of size 2 but the neighborhood diversity could be arbitrary; for parameter distance to co-cluster, think of the complementary of a cluster graph with an unbounded number of cliques.

▶ **Theorem 9 (★).** Graph Motif *can be solved in time* $O^*(2^{4k \log(2k)})$ *where* k *is the vertex clique cover number, provided that the vertex clique cover is given as part of the input.*

▶ **Theorem 10 (★).** Graph Motif *can be solved in* $O^*(2^{2k \log k})$*, where* k *is the distance to co-cluster.*

4 Parameters for which Graph Motif is hard

In this section, we provide several parameters for which Graph Motif is not in XP, unless P = NP. In other words, the problem is NP-hard even for fixed values of the parameter. We also prove that the problem remains W[1]-hard for parameter max leaf number. Figure 1 summarizes these results.

4.1 Deletion set numbers

We study parameters which correspond to the minimum number of vertices to remove to make the graph belong to a restricted class. We will show that Graph Motif remains NP-hard for constant values of those parameters. More precisely, the colorful restriction of Graph Motif is hard even if we can obtain a set of disjoint paths by removing 1 vertex, a cluster graph by removing 1 vertex, and an acyclic graph by removing 0 edge.

▶ **Theorem 11** ([17]). Graph Motif *is* NP*-hard even when G is a tree of maximum degree 3 and the motif is colorful.*

▶ **Corollary 12.** Graph Motif *is* NP*-hard even for graphs with feedback edge set 0 and when the motif is colorful.*

▶ **Theorem 13 (★).** Graph Motif *is* NP*-hard even (i) for graphs with distance* 1 *to disjoint paths and when the motif is colorful and (ii) for graphs with bandwidth* 4 *and when the motif is colorful.*

▶ **Theorem 14 (★).** Graph Motif *is* NP*-hard even for graphs with distance* 1 *to cluster and when the motif is colorful.*

4.2 Dominating set number

Being given a small dominating set of the graph cannot help in solving Graph Motif. For any instance $(G = (V, E), c, M)$, one may add a universal new vertex v to G, and color it with a color which does not appear in motif M. The minimum dominating set $\{v\}$ is of size 1. Vertex v cannot be part of the solution due to its color, so answering the new problem is as hard as solving the original instance. Though, this could be considered as cheating since a vertex whose color is not in M can immediately be discarded from the graph. We show that even when $\forall v \in V$, $c(v) \in M$, graphs with dominating set of size 2 can be hard to solve.

▶ **Theorem 15 (★).** Graph Motif *is* NP*-hard even for graphs with a minimum dominating set of size* 2 *and when the motif is colorful.*

4.3 Max leaf number

The *max leaf number* of a graph G, denoted $\mathrm{ml}(G)$ is the maximum number of leaves (i.e., vertices of degree 1) in a spanning tree of G. Therefore, if G is itself a tree, then $\mathrm{ml}(G)$ is simply the number of leaves of G. We will show that Graph Motif is in XP (even in W[P]) and is W[1]-hard with parameter max leaf number. In fact, we will even prove that it is W[1]-hard on trees with parameter *number of leaves in the tree plus number of distinct colors in the motif*. This strenghtens the previously known result that the problem is W[1]-hard on trees with parameter number of distinct colors in the motif [17].

▶ **Theorem 16 (★).** Graph Motif *can be solved in time $O^*(16^k n^{10k}) = n^{O(k)}$, where $k = ml(G)$ and is even in* W[P] *with respect to that parameter.*

▶ **Theorem 17 (★).** Graph Motif *is* W[1]*-hard with respect to the max leaf number plus the number of colors, even on trees.*

5 Conclusion and open problems

Figure 1 sums up the parameterized complexity landscape of Graph Motif with respect to structural parameters. For parameter maximum independent set the complexity status of Graph Motif remains unknown. Even when the problem is in FPT, polynomial kernels tend to be unlikely; be it for the natural parameter even on comb graphs [2] or for the vertex cover number or the distance to clique (Theorem 8). Is it also the case for parameter cluster editing number?

The sparsification lemma [23] together with a straightforward reduction from 3-SAT shows that, under ETH, Graph Motif cannot be solved in time $2^{o(n)}$ on graphs with n vertices. Thus, for every parameter k bounded by n, an algorithm solving Graph Motif in $2^{o(k)}$ would disprove ETH. This is the case of four out of six parameters for which we have given an FPT algorithm; cluster editing and edge clique cover numbers are only bounded by n^2. On the one hand, it says that our algorithm running in $2^{O(k)}$ for parameter distance to clique is probably close to optimal. On the other hand, for parameter vertex cover number, for instance, we have still some room for improvement between the $2^{O(k \log k)}$-upper bound and the $2^{o(k)}$-lower bound under ETH. Can we improve the algorithm to time $2^{O(k)}$, or, on the contrary, show a stronger lower bound of $2^{o(k \log k)}$ (potentially using [29])?

References

1 E. Alm and A. P. Arkin. Biological Networks. *Current Opinion in Structural Biology*, 13(2):193–202, 2003.

2 A. M. Ambalath, R. Balasundaram, C. Rao H., V. Koppula, N. Misra, G. Philip, and M. S. Ramanujan. On the Kernelization Complexity of Colorful Motifs. In *Proc. of the 5th IPEC*, volume 6478 of *LNCS*, pages 14–25. Springer, 2010.

3 D. Berend and T. Tassa. Improved bounds on bell numbers and on moments of sums of random variables. *Probab. and Math. Statist.*, 30(2):185–205, 2010.

4 N. Betzler, R. van Bevern, M. R. Fellows, C. Komusiewicz, and R. Niedermeier. Parameterized algorithmics for finding connected motifs in biological networks. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 8(5):1296–1308, 2011.

5 A. Björklund, P. Kaski, and L. Kowalik. Probably optimal graph motifs. In *Proc. of the 30th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 20 of *LIPIcs*, 2012.

6 A. Björklund, P. Kaski, L. Kowalik, and J. Lauri. Engineering motif search for large graphs. In U. Brandes and D. Eppstein, editors, *Proc. of the 17th ALENEX*, pages 104–118. SIAM, 2015.

7 S. Böcker. A golden ratio parameterized algorithm for cluster editing. *J. Discrete Algorithms*, 16:79–89, 2012.

8 S. Böcker, F. Rasche, and T. Steijger. Annotating Fragmentation Patterns. In *Proc. of the 9th International Workshop Algorithms in Bioinformatics (WABI)*, volume 5724 of *LNCS*, pages 13–24. Springer, 2009.

9 H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014.

10 S. Bruckner, F. Hüffner, R. M. Karp, R. Shamir, and R. Sharan. Topology-Free Querying of Protein Interaction Networks. *Journal of Computational Biology*, 17(3):237–252, 2010.

11 M. Cesati. The Turing way to parameterized complexity. *J. Comput. Syst. Sci.*, 67(4):654–685, 2003.

12 J. Chen, I. A. Kanj, and G. Xia. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411(40–42):3736 – 3756, 2010.

13 M. Cygan, M. Pilipczuk, and M. Pilipczuk. Known algorithms for EDGE CLIQUE COVER are probably optimal. In *Proc. of Symposium on Discrete Algorithms, SODA 2013*, pages 1044–1053. SIAM, 2013.

14 M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. Kernelization hardness of connectivity problems in d-degenerate graphs. *Discrete Applied Mathematics*, 160(15):2131–2141, 2012.

15 R. Dondi, G. Fertin, and S. Vialette. Complexity issues in vertex-colored graph pattern matching. *J Discr Algo*, 9(1):82–99, 2011.

16 R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

17 M. R. Fellows, G. Fertin, D. Hermelin, and S. Vialette. Upper and lower bounds for finding connected motifs in vertex-colored graphs. *J. Comput. Syst. Sci.*, 77(4):799–811, 2011.

18 M. R. Fellows, D. Lokshtanov, N. Misra, M. Mnich, F. A. Rosamond, and S. Saurabh. The complexity ecology of parameters: An illustration using bounded max leaf number. *Theory Comput. Syst.*, 45(4):822–848, 2009.

19 R. Ganian. Twin-cover: Beyond vertex cover in parameterized algorithmics. In *Proc. of the 6th International Symposium Parameterized and Exact Computation IPEC 2011*, volume 7112 of *LNCS*, pages 259–271. Springer, 2011.

20 R. Ganian. Using neighborhood diversity to solve hard problems. *CoRR*, abs/1201.3091, 2012.

21 J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Data reduction and exact algorithms for clique cover. *ACM Journal of Experimental Algorithmics*, 13, 2008.

22 S. Guillemot and F. Sikora. Finding and counting vertex-colored subtrees. *Algorithmica*, 65(4):828–844, 2013.

23 R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

24 D. J. Kleitman and D. B. West. Spanning trees with many leaves. *SIAM J. Discrete Math.*, 4(1):99–106, 1991.

25 C. Komusiewicz and R. Niedermeier. New races in parameterized algorithmics. In *Proc. of Mathematical Foundations of Computer Science MFCS 2012*, volume 7464 of *LNCS*, pages 19–30. Springer, 2012.

26 I. Koutis. Constrained multilinear detection for faster functional motif discovery. *Inf. Process. Lett.*, 112(22):889–892, 2012.

27 V. Lacroix, C. G. Fernandes, and M.-F. Sagot. Motif search in graphs: application to metabolic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 3(4):360–368, 2006.

28 M. Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012.

29 D. Lokshtanov, D. Marx, and S. Saurabh. Slightly superexponential parameterized problems. In *Proc. of SODA 2011*, pages 760–776, 2011.

30 C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

31 R. Niedermeier. *Invitation to Fixed Parameter Algorithms*. Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006.

32 R. Y. Pinter, H. Shachnai, and M. Zehavi. Deterministic parameterized algorithms for the graph motif problem. In *Proc. of Mathematical Foundations of Computer Science MFCS 2014*, volume 8635 of *LNCS*, pages 589–600. Springer, 2014.

33 R. Y. Pinter and M. Zehavi. Algorithms for topology-free and alignment network queries. *J. Discrete Algorithms*, 27:29–53, 2014.

A Appendix

A.1 Preliminaries

A W[1]-hard problem is not fixed-parameter tractable (unless FPT = W[1]) and one can prove W[1]-hardness by means of a *parameterized reduction* from a W[1]-hard problem. This is a mapping of an instance (I, k) of a problem A_1 in $g(k) \cdot |I|^{O(1)}$ time (for any computable function g) into an instance (I', k') for A_2 such that $(I, k) \in A_1 \Leftrightarrow (I', k') \in A_2$ and $k' \leq h(k)$ for some function h.

A.2 Preliminaries

It is well known that a problem is in FPT iff it has a kernel, but this equivalence yields super-polynomial kernels (in general). To design efficient parameterized algorithms, a kernel of polynomial (or even linear) size in k is important. However, some lower bounds on the size of the kernel can be shown unless some polynomial hierarchy collapses. To show this result, we will use the cross composition technique developed by Bodlaender et al. [9]. Due to space constraints, we defer the useful definitions and theorems to appendix (but can also be found in [9]).

A.3 Preliminaries

▶ **Definition 18** (Polynomial equivalence relation [9]). An equivalence relation $\mathcal{R}$ on Σ^* is said to be *polynomial* if the following two conditions hold: (i) There is an algorithm that given two strings $x, y \in \Sigma^*$ decides whether x and y belong to the same equivalence class in time $(|x| + |y|)^{O(1)}$. (ii) For any finite set $S \subseteq \Sigma^*$ the equivalence relation $\mathcal{R}$ partitions the elements of S into at most $(\max_{x \in S} |x|)^{O(1)}$ classes.

▶ **Definition 19** (OR-cross-composition [9]). Let $L \subseteq \Sigma^*$ be a set and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. We say that L *cross-composes* into Q if there is a polynomial equivalence relation $\mathcal{R}$ and an algorithm which, given t strings $x_1, x_2, \ldots, x_t$ belonging to the same equivalence class of $\mathcal{R}$, computes an instance $(x^*, k^*) \in \Sigma^* \times \mathbb{N}$ in time polynomial in $\sum_{i=1}^{t} |x_i|$ such that: (i) $(x^*, k^*) \in Q \Leftrightarrow x_i \in L$ for some $1 \leqslant i \leqslant t$. (ii) k^* is bounded by a polynomial in $\max_{i=1}^{t} |x_i| + \log t$.

▶ **Theorem 20** ([9]). *Let $L \subseteq \Sigma^*$ be a set which is* NP*-hard under Karp reductions. If L cross-composes into the parameterized problem Q, then Q has no polynomial kernel unless* NP $\subseteq$ coNP/*poly*.

A.4 Proof 1 (Theorem 9)

Proof. Let $(G = (V, E), c, M)$ be the instance and suppose that the partition into cliques $\{C_1, \ldots, C_k\}$ of the graph G is given. We remove all the vertices whose color does not belong to M, since they cannot be part of a solution. We also remove all the edges between two vertices of the same color c, if c has multiplicity exactly 1 in M. This is safe because at most one endpoint of such an edge can be in a solution. First, we guess in time $O^*(2^k)$ which of the cliques $\mathcal{S} = \{C'_1, \ldots, C'_{k'}\} \subseteq \{C_1, \ldots, C_k\}$ have a non-empty intersection with a fixed solution R, and we remove from G the cliques which are not in $\mathcal{S}$.

We call *transversal edge* an edge in $E(C'_i, C'_j)$ with $i \neq j \in [k']$, where $E(X, Y)$ denotes the set of edges of E having one endpoint in X and the other in Y. Such a transversal edge

is said to have *type* $\{i, j\}$. An *inner edge* is an edge which lies within the same clique C'_i for some $i \in [k']$. As $G[R]$ is connected, one may observe that there is a set $E_c \subseteq E(G[R])$ of $k' - 1$ transversal edges such that between every pair of vertices u, $v(\in R)$, there is a path made only of edges in E_c and inner edges. Informally, E_c is a spanning tree of the k' cliques of $\mathcal{S}$ seen as vertices.

We guess in time $O^*(k'^{2(k'-1)})$ the type of each edge in E_c. We denote by T_c the corresponding set of $k' - 1$ types. We also guess in time $O^*(2^{k'})$ if two edges in E_c of types $\{i, j\}$ and $\{i, j'\}$, happen to have a common endpoint (the same vertex in C'_i). As R is a solution, $M \subseteq c(C'_1 \cup \ldots \cup C'_{k'})$ holds. Therefore, it all boils down to finding $k'-1$ transversal edges whose set of types is precisely T_c and such that the multiset of colors of their at most $2(k' - 1)$ endpoints is included in M.

For each type $\{i, j\} \in T_c$, we build the bipartite graph $B_{i,j} = (H_i \uplus H_j, F)$ where H_i (resp. H_j) are all the colors of the vertices of C'_i (resp. C'_j). There is an edge in F between color $c \in H_i$ and color $c' \in H_j$ whenever there is a transversal edge of type $\{i, j\}$ whose endpoint in C'_i is colored by c and whose endpoint in C'_j is colored by c'. The rest of the algorithm is a win/win based on the classic König's theorem which states that in a bipartite graph the size of a minimum vertex cover is equal to the size of a maximum matching. The core idea is that either there is a large diversity of colors for the endpoints of a transversal edge, and a suitable transversal edge can always be found at the end, or there is only a limited choice of colors for those endpoints and one can branch over those possibilities. By branching, we commit ourselves to find a transversal edge uv whose endpoint, say, u has a specific color c. In that case, we say that the endpoint u has its color *fixed*. In a first step, we will branch until the endpoints of all the transversal edges are fixed (or can always be fixed). In a second step, we will build a solution respecting the fixed colors.

We distinguish two cases. Either, there is a matching $S_{i,j}$ in $B_{i,j}$ with at least $2k' - 3$ edges. Then, for any multiset of colors M_o of size at most $2k' - 4$, there is an edge $\{c, c'\}$ in $S_{i,j}$ such that $M_o \cup \{c, c'\} \subseteq M$. Indeed, since $|S_{i,j}| > |M_o|$, there is at least one edge of $S_{i,j}$ whose endpoints are not colored by an element of M_o. Recall also that there can be an edge between two vertices of the same color only if the multiplicity of that color in M is at least 2. Therefore, whatever the multiset $M_o \subseteq M$ of colors at the endpoints of the $k' - 2$ other transversal edges is, one can always find a transversal edge of type $\{i, j\}$ colored by c and c' such that $M_o \cup \{c, c'\} \subseteq M$. Thus, we can forget about this particular transversal edge, and we say that the transversal edge of type $\{i, j\}$ is *abundant*.

Otherwise, there is a vertex cover of $B_{i,j}$ with at most $2k' - 4$ vertices. Note that a vertex $c \in H_i$ (resp. H_j) in the graph $B_{i,j}$ corresponds to choosing color c for the endpoint in C'_i (resp. C'_j) of the transversal edge of type $\{i, j\}$. Therefore, we branch on those at most $2k' - 4$ possibilities of coloring one of the endpoint of the transversal edge of type $\{i, j\}$.

This describes what we do when no endpoint of the transversal edge has its color fixed. Now, suppose we have a transversal edge of type $\{i, j\}$ such that the color of the endpoint in, say, C'_i is fixed to color c. If the number of neighbors of vertex $c \in H_i$ in the graph $B_{i,j}$ is at least $2k' - 3$, we declare this edge abundant and no longer care about this edge. Otherwise, if this number is at most $2k' - 4$, we branch on the at most $2k' - 4$ ways of coloring the endpoint in C'_j of the transversal edge of type $\{i, j\}$.

Note also that when we fix the color of an endpoint in C'_i of a transversal edge of type $\{i, j\}$, it also fixes the color of the endpoints in C'_i of potential transversal edges of type $\{i, j'\}$ which we have guessed to share a common endpoint (in C'_i) with the transversal edge of type $\{i, j\}$. Naturally, this potential set of transversal edges might very well be empty. After a branching of depth at most $2k' - 2$ and arity at most $2k' - 4$, we reach a situation

where each transversal edge is either abundant or both its endpoints have fixed colors. We fix arbitrary colors to the endpoints of the abundant transversal edges, which are not fixed yet. We explained above why this is always possible.

Now, all the endpoints of the transerval edges have their color fixed. By guessing the set T_c of types of the transversal edges and whether or not two transversal edges are incident, we have in fact guessed the shape of a forest that those edges constitute in the original graph G. In each tree of this abstract forest, we compute actual transversal edges in a bottom-up manner. Let us see this tree as labeled over pairs clique/color (C'_i, c). We associate each leaf labeled by (C'_i, c) with the subset $J_{i,c} \subseteq C'_i$ of vertices colored by c (that is, $\forall u \in C'_i$, $u \in J_{i,c} \Leftrightarrow c(u) = c$). We associate each inner node labeled by (C'_i, c) whose r children are associated with sets $J_{i_1,c_1}, \ldots, J_{i_r,c_r}$ with the subset $J_{i,c} \subseteq C'_i$ of vertices colored by c which have at least one neighbor in J_{i_h,c_h} for each $h \in [r]$. When the last node e of the tree gets its set J, this set is non empty if we have made all our guesses accordingly to solution R. We define e as the root of the tree. Now, in a top-down manner we find the corresponding transversal edges. We take in the solution an arbitrary vertex $u \in J$. In each set associated with a child of e we take arbitrarily a neighbor of u; and so on, up to the leaves. By construction, this is always possible. It is possible that while doing this process on two different trees of the forest, we take "twice" the same vertex in some C'_i. This can only help since the goal is not to exceed the multiplicities of M. Equivalently, we could have guessed the forest of transversal edges with the least number of connected components, to forbid this possibility.

The running time of the algorithm is $O^*(2^k k^{2k-2} 2^k (2k-4)^{2k-2}) = O^*((2k)^{4k})$. ◀

A.5 Proof 2 (Theorem 10)

Proof. Let $(G = (V, E), c, M)$ be any instance of Graph Motif and let R be a solution. Let X be a minimum subset (of size k) whose deletion makes the graph G a co-cluster. Co-cluster graphs are exactly the $\overline{P_3}$-free graphs. We can apply a bounded-depth branching algorithm by finding a $\overline{P_3}$ and branching on which of the three vertices to put into the solution. This leads to a $O^*(3^k)$ algorithm. Let $S_1, S_2, ..., S_q$ be the partition of the co-cluster $G[V \setminus X]$ into maximal independent sets. The idea is to run the algorithm parameterized by the vertex cover number if at most one S_i is inhabited by solution R, and the one parameterized by distance to clique otherwise. Therefore, we distinguish two cases:
(A) $|\{i \in [q] \mid R \cap S_i \neq \emptyset\}| \leqslant 1$ or (B) $|\{i \in [q] \mid R \cap S_i \neq \emptyset\}| \geqslant 2$.

In case (A) holds, we will find a solution by solving, for each $i \in [q]$, the instance $(G[X \cup S_i], c_{|X \cup S_i}, M)$. As X is a vertex cover of size k in $G[X \cup S_i]$, this could be done in time $O^*(2^{2k \log k})$ by Theorem 6.

In case (B) holds, we can guess in time n^2 one vertex $s \in S_i \cap R$ and one vertex $t \in S_j \cap R$ with $i \neq j \in [q]$. Then, we will find a solution by solving $(G' = (V \setminus \{s,t\}, E'), c_{V \setminus \{s,t\}}, M \setminus c(\{s,t\}))$ where $E' = (E \cup \{\{u,v\} \mid u \in S_a, v \in S_b, a \neq b \in [q]\})_{|V \setminus \{s,t\}}$. Indeed, if $Y \subseteq V \setminus \{s,t\}$ induces a connected subgraph in G', then $G[Y \cup \{s,t\}]$ is connected. As $G' - X$ is a clique, this can be done in time $O^*(4^k)$ by Theorem 5. The overall running time is $O^*(3^k + q2^{2k \log k} + n^2 4^k) = O^*(2^{2k \log k})$. ◀

A.6 Proof 3 (Theorem 13)

Proof. We will detail only (i). We propose a reduction from Exact Cover by 3-Sets (X3C). This special case of Set Cover is known to be NP-complete. Recall that X3C is

stated as follows, given a set $X = \{x_1, x_2, \ldots, x_{3q}\}$ and a collection $\mathcal{S} = \{S_1, \ldots, S_{|\mathcal{S}|}\}$ of 3-elements subsets of X, the goal is to decide if $\mathcal{S}$ contains a subcollection $\mathcal{T} \subseteq \mathcal{S}$ such that each element of X occurs in exactly one element of $\mathcal{T}$. The size of X must be a multiple of three since a solution is a set of triplets where each element of X must appears exactly once.

Let us now describe the construction of an instance $\mathcal{I}' = (G = (V, E), c, M)$ of Graph Motif from an arbitrary instance $\mathcal{I} = (X, \mathcal{S})$ of X3C (see also Figure 3). The graph $G = (V, E)$ is built as follows: there is a distinct root r, for each $S_i \in \mathcal{S}$, there are two paths built from r, the first one is made of a node a_i^1, three nodes representing the elements in S_i and a node b_i^1, the other one is made of two nodes a_i^2 and b_i^2. The graph is thus a tree such that removing r gives a collection of $2|\mathcal{S}|$ paths.

The set of colors is $\mathcal{C} = \{1, 2, \cdots, 2|\mathcal{S}| + 3q + 1\}$. The coloration of G is such that $c(a_i^1) = c(a_i^2) = i$ and $c(b_i^1) = c(b_i^2) = |\mathcal{S}| + i$ for $1 \leqslant i \leqslant |\mathcal{S}|$, the $3q$ colors $2|\mathcal{S}| + 1, \cdots, 2|\mathcal{S}| + 3q$ are assigned to vertices corresponding to X, and $c(r) = 3q + 2|\mathcal{S}| + 1$. The motif is equal to the set of colors and is thus colorful. This construction is clearly done in polynomial-time in regards of $\mathcal{I}$.

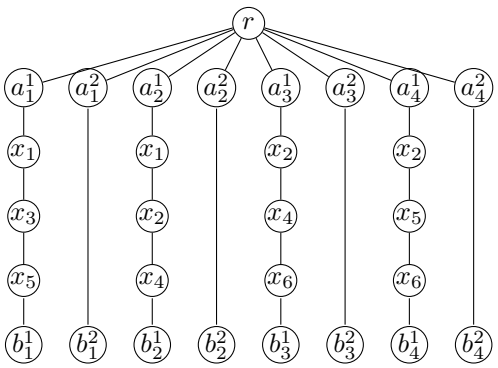

Figure 3 The graph G built from $X = \{x_1, x_2, \ldots, x_6\}$ (thus with $q = 2$) and $\mathcal{S} = \{\{x_1, x_3, x_5\}, \{x_1, x_2, x_4\}, \{x_2, x_4, x_6\}, \{x_2, x_5, x_6\}\}$.

Let us now prove that if there is a solution for an instance $\mathcal{I}$ of X3C, then there is solution for the instance $\mathcal{I}'$ of Graph Motif. Given a solution $\mathcal{T} \subseteq \mathcal{S}$ for $\mathcal{I}$, a solution P for $\mathcal{I}'$ is built as follows: take the root, for each $S_i \in \mathcal{T}$, take the whole path from a_i^1 to b_i^1, and for each $S_i \notin \mathcal{T}$, take the path $a_i^2 b_i^2$. Informally speaking, for each set, either the set is in $\mathcal{T}$ and thus the path with the nodes corresponding to the elements is taken, otherwise the path with only two nodes is taken. By definition of a solution for $\mathcal{I}$, each color $2|\mathcal{S}| + 1, \cdots, 2|\mathcal{S}| + 3q$ is taken only once, and for each color $1, \cdots, 2|\mathcal{S}|$, exactly one of the two occurrences is taken. The root is also taken and thus the solution is connected.

Conversely, let us now prove that there is a solution for the instance $\mathcal{I}$ of X3C if there is a solution for the instance $\mathcal{I}'$ of Graph Motif. First observe that the root r must be in the solution since it is the only node with this color. Also, for each $1 \leqslant i \leqslant |\mathcal{S}|$, either a_i^1 or a_i^2 must be in the solution since it is the only node with color i. The same holds for b_i^1 and b_i^2. Also, observe that if a_i^1 is in the solution, then b_i^1 must also be in the solution, with the three element nodes along the path. Indeed, if it is not the case, the color $c(b_i^1)$ will never be in the solution since the only other node with this color is b_i^2. However, in order to add b_i^2 in the solution, a_i^2 must be in the solution to respect the connectivity constraint, which is impossible since $c(a_i^1) = c(a_i^2)$. Therefore, either the three element nodes corresponding

to a set $S_i \in \mathcal{S}$ are entirely in the solution P, or none are. The solution is built as follows: $\mathcal{T} = \{S_i : a_i^1 \in P\}$. Since P is a solution, colors of P appears exactly once. Therefore, each element of X appears exactly once in $\mathcal{T}$.

For (ii), we slightly modify the graph G. Instead of having one vertex r linked to each a_i^j (for $i \in [|\mathcal{S}|]$ and $j \in [2]$), we now have a path $S = r_1^1 r_1^2 r_2^1 r_2^2 \dots r_{|\mathcal{S}|}^1 r_{|\mathcal{S}|}^2$, and for each $i \in [|\mathcal{S}|]$ and $j \in [2]$, there is an edge between r_i^j and a_i^j. We call that new graph H. We may observe that H is a comb graph whose spine is S. The set of colors is now $\mathcal{C} = [4|\mathcal{S}| + 3q]$. All the vertices in $G - r$ keep the same colors, and for each $i \in [|\mathcal{S}|]$ and $j \in [2]$, $c(r_i^j) = 2|\mathcal{S}| + 3q + 2(i-1) + j$. In other words, we give a fresh and distinct color to each vertex of S. Again, the motif M is the entire set of colors $\mathcal{C}$. The correctness is the same as for (i), since all vertices of S must be in any solution because they are the only occurrences of their respective color. Since the maximal paths having exactly one vertex in the spine S, called *teeth*, are of length at most 6, the bandwidth of H is bounded by 6, too. Indeed, one can number the vertices increasingly tooth by tooth. A more careful analysis shows that the bandwidth of H is actually 5.

Equivalently, we could have followed the reduction of [14] starting from a version of Sat where each literal appears in at most two clauses. This variant is also NP-complete, and the graph produced would have bandwidth 4. ◀

A.7 Proof 4 (Theorem 14)

Sketch. To prove this theorem, one can use the reduction from Colorful Set Cover to Graph Motif where the input graph is a tree of diameter at most 4 (called superstar) [2]. The idea is just to replace each subtree representing a set S_i by a clique of size $|S_i| + 1$. Removing the root of the former superstar in this new graph yields a disjoint union of cliques and the rest of the proof carries over. ◀

A.8 Proof 5 (Theorem 15)

Proof. We reduce from a rooted variant of Graph Motif, where the solution should contain a special vertex r. As checking for every possible root if rooted Graph Motif has a solution would answer the standard Graph Motif problem, rooted Graph Motif is also NP-hard.

We will now prove that the problem remains hard with a small dominating set. The informal idea is to add a universal node u such that the dominating set is small, but with a gadget to avoid the possibility of having this universal node in a solution (making the problem easy since any subset will be connected due to u). More formally, from any instance $\mathcal{I} = (G = (V, E), c, M)$, and any fixed vertex r in V, we build the instance $\mathcal{I}' = (G' = (V \cup \{u, s, t\}, E'), c', M')$, where $E' = E \cup \{\{s, t\}, \{t, r\}\} \cup \{\{u, w\} \mid w \in V\}$, $c'(w) = c(w)$ for each $w \in V$, $c'(t) = c'(u) = x$, $c'(s) = y$, with x and y being two distinct fresh colors, and $M' = M \cup \{x, y\}$. $\{u, t\}$ is a dominating set in G' of size 2. Let R be a solution of Graph Motif for instance $\mathcal{I}'$. Vertex s is the only vertex with color y, so it has to be in R. But then, as the only neighbor of s is t (and $|M'| \geqslant 2$), t should also be in R. Only one vertex with color x can be in R, so u cannot be part of the solution. Now, the problem is as hard as solving instance $\mathcal{I}$ rooted in r. ◀

A.9 Proof 6 (Theorem 16)

Proof. Let $(G = (V, E), c, M)$ be any instance of Graph Motif, $k = \mathrm{ml}(G)$, and S the set of vertices with degree stricly greater than 2 in G. Again, we may assume that G is connected and also that G is not a cycle, since otherwise Graph Motif is trivially solvable in time $O(n^2)$.

It is known that $|S| \leqslant 4k$ (even $4k-2$) [24]. First, we can exhaustively find in time $2^{4k} = 16^k$ the intersection $T = S \cap R$, where R is a fixed solution. By definition, $V \setminus S$ are vertices of degree at most 2. In particular, $G[V \setminus S]$ is a disjoint union of paths (some of the paths may be reduced to a single vertex). Indeed, there cannot be a cycle in $G[V \setminus S]$ since this cycle could not be connected to the rest of G.

We can show that the number of paths in $G[V \setminus S]$ is at most $5k$. As G is connected, we can find $s-1$ paths $P_1, \ldots, P_{s-1}$ of $G[V \setminus S]$ such that $G[S \cup P_1 \cup \ldots \cup P_{s-1}]$ is connected, where s is the number of connected components of $G[S]$. Therefore, we build the following spanning tree of G: we start by taking the edges of any spanning forest of $G[S]$, plus all the edges incident to at least one vertex of a path P_i (for $i \in [s-1]$). Now, all the remaining paths in $G[V \setminus S]$ will provide (at least) one leaf each. As $s \leqslant |S| \leqslant 4k$, if the number of paths in $G[V \setminus S]$ were larger than $5k$, then we could exhibit a spanning tree with at least $k+1$ leaves, which is a contradiction to $k = \mathrm{ml}(G)$.

To satisfy the connectivity constraint, solution R can intersect each of the at most $5k$ paths of $G[V \setminus S]$ in at most n^2 different ways (more precisely in at most $\binom{l}{2} + l + 1$ where l is the number of vertices in the path). So, we can guess the intersection $R \cap (V \setminus S)$ in time $(n^2)^{5k} = n^{10k}$. Overall, we can decide Graph Motif in time $O^*(16^k n^{10k})$ where k is the max leaf number.

We can also show that Graph Motif parameterized by $\mathrm{ml}(G)$ is in $\mathsf{W}[\mathsf{P}]$ with the characterisation of this class by Turing machines with bounded non-determinism [11]. ◀

A.10 Proof 7 (Theorem 17)

Proof. We show the stronger result that Graph Motif is $\mathsf{W}[1]$-hard on subdivisions of the star $K_{1,k}$ with parameter $k + |\mathcal{C}|$ where $\mathcal{C}$ is the set of colors. From any instance $H = (H_1 \uplus H_2 \uplus \ldots \uplus H_k, E)$ of the $\mathsf{W}[1]$-hard problem Multicolored k-Clique, we construct an equivalent instance $(T = (V, E'), c : V \to \mathcal{C}, M)$ of Graph Motif where T is a tree with $k + 2\binom{k}{2} + 1$ leaves and $\mathcal{C}$ contains $2\binom{k}{2} + 3$ colors. More precisely, T is a subdivision of the star $K_{1,k+2\binom{k}{2}+1}$. We recall that the Multicolored k-Clique problem asks for a k-clique in H hitting each H_i (exactly once). By potentially adding some isolated vertices, we can assume that each H_i contains the same number t of vertices.

The set of colors $\mathcal{C}$ is $\{c_m, c_b, c_e\} \cup \bigcup_{i<j\in[k]} \{c_{i,j,+}, c_{i,j,-}\}$ ($|\mathcal{C}| = 2\binom{k}{2} + 3$). The motif M contains c_m with multiplicity $k+1$, both c_b and c_e with multiplicity $t^2k^2 + tk + \binom{k}{2}((t+1)^2 + t)$, and for each $i < j \in [k]$, both $c_{i,j,+}$ and $c_{i,j,-}$ with multiplicity $(t+1)^2 + t$. The tree T is a subdivision of $K_{1,k+2\binom{k}{2}+1}$ whose center v is colored by c_m (as **m**andatory). Exactly k other vertices than v have color c_m, so they all should be in the solution. By definition, $T[V \setminus \{v\}]$ is a disjoint union of $k + 2\binom{k}{2} + 1$ paths.

First, there are k paths $P_1, P_2, \ldots, P_k$ where P_j is meant to encode color class H_j, for each $j \in [k]$. The *start* of path P_j is the unique neighbor of v in P_j and the *end* is the other endpoint of the path. For each $j \in [k]$, we describe P_j from its start to its end. For each $i < j$, there are $t+1$ vertices with color $c_{i,j,+}$ (in any order), and then a vertex with color c_m. We denote by $P_{j,m}$ this first part of path P_j, and observe that $\{v\} \cup \bigcup_{j\in[k]} P_{j,m}$ has to

be in any solution. We may also notice that $P_{1,m}$ is reduced to a single vertex with color c_m. The rest of P_j encodes the vertices $v_1^j, v_2^j, \dots, v_t^j$ of H_j. For each $h \in [t]$, we encode vertex v_h^j by a path starting with a vertex colored by c_b (as **b**egin) and ending with a vertex colored by c_e (as **e**nd), and in between we put (in any order) $t+1$ vertices colored by $c_{i,j,+}$ for each $i \in [j-1]$, and one vertex colored by $c_{j,l,+}$ for each $l \in [j+1,k]$. We call this gadget *block* encoding v_h^j and denote it by $B(v_h^j)$. In general, we call *block* a subpath of T minimal for the following property: the endpoints of the subpath are x colored by c_b and y colored by c_e, and x is in the path from center v to y. The blocks $B(v_1^j), B(v_2^j) \dots, B(v_t^j)$ are put one after the other along path P_j and that ends the construction of the P_js.

We now describe the $\binom{k}{2}$ paths $P_{i,j}$ (with $i < j \in [k]$) of $T[V \setminus \{v\}]$ each encoding the edges of $E_{i,j} = E(H_i, H_j)$. To construct $P_{i,j}$, we compute the list of $|E_{i,j}|$ positive integers $L_{i,j} = \{(t+1)(h'+1)+h \mid v_h^i v_{h'}^j \in E_{i,j}\}$. Indeed, for $h, h' \in [t]$, the $(t+1)h'+h$ are pairwise distinct integers (having different quotient and/or remainder in the euclidian division by $t+1$), and the $(t+1)(h'+1)+h$ are just the same numbers with an offset of $+(t+1)$. We sort $L_{i,j}$ and we define a new (non sorted) list of $|E_{i,j}|$ positive integers $D_{i,j}$ of differences of two consecutive entries of the sorted $L_{i,j}$. More precisely, the first entry of $D_{i,j}$ is the smallest value in $L_{i,j}$ and its r-th entry ($r \in [2, |E_{i,j}|]$) is the r-th entry of sorted $L_{i,j}$ minus the $r-1$-th entry of sorted $L_{i,j}$. Let us denote by $d_1^{i,j}, d_2^{i,j}, \dots, d_{|E_{i,j}|}^{i,j}$ the successive entries of $D_{i,j}$. $P_{i,j}$ consists of $|E_{i,j}|$ blocks $B^{i,j}(s)$ with a vertex colored by c_b, then $d_s^{i,j}$ vertices colored by $c_{i,j,-}$, and finally a vertex colored by c_e. Here, the order of the blocks matters. We put $B^{i,j}(1)$ first, and then $B^{i,j}(2)$ up to $B^{i,j}(|E_{i,j}|)$.

In $T[V \setminus \{v\}]$, there is also a path P_B (as **B**lock) of length $2(t^2k^2 + tk + \binom{k}{2}((t+1)^2+t))$ regularly alternating a vertex colored by c_b and a vertex colored by c_e, and such that the neighbor of v in P_B is colored by c_b.

Last, for each $i < j \in [k]$, $T[V \setminus \{v\}]$ comprises a path $P'_{i,j}$ of length $4((t+1)^2+t)$ made of $(t+1)^2+t$ consecutive copies of the same block of size 4: a vertex colored by c_b, then two vertices colored by $c_{i,j,+}$ and $c_{i,j,-}$, and finally a vertex colored by c_e. This ends the construction of T.

As observed earlier, a solution has to contain $\{v\} \cup \bigcup_{j \in [k]} P_{j,m}$ since it is necessary to satisfy both the multiplicity of color c_m and the connectivity constraint. A connected subgraph of a subdivision of star $K_{1,k+2\binom{k}{2}+1}$ containing its center v, is entirely defined by its at most $k + 2\binom{k}{2} + 1$ leaves (at most 1 leaf per path in $T[V \setminus \{v\}]$). We can therefore describe a solution as where to stop in each of those $k + 2\binom{k}{2} + 1$ paths.

A solution cannot stop in the middle of a block. Indeed, no matter where we stop in the other paths, the number of vertices colored by c_b is at least the number of vertices colored by c_e. In total, such a solution would have at least one more vertex with color c_b than with color c_e, and therefore would not satisfy the motif M. Conversely, if we respect this rule of not stopping in the middle of a block, we can satisfy the multiplicity of c_b and c_e. Indeed, the number of vertices with color c_b (resp. c_e) outside P_B is smaller than $t^2k^2 + tk + \binom{k}{2}((t+1)^2+t)$ (tk for the P_js, $\Sigma_{i<j\in[k]}|E_{i,j}| = |E| \leqslant t^2k^2$ for the $P_{i,j}$s, and $\binom{k}{2}((t+1)^2+t)$ for the $P'_{i,j}$s). So, if we take p vertices of color c_b (resp. c_e) outside P_B, we can always complete with the first $2(t^2k^2 + tk + \binom{k}{2}((t+1)^2+t) - p)$ vertices of path P_B.

For any $i < j \in [k]$ the number of vertices colored by $c_{i,j,+}$ outside $P'_{i,j}$ is $(t+1)^2+t$ (that is $(t+1)+(t+1)t = (t+1)^2$ in path P_j and t in path P_i). And similarly, if we stop the paths P_i, P_j and $P_{i,j}$ in such a way that we have the same number p' of vertices colored by $c_{i,j,+}$ and by $c_{i,j,-}$, then we can complete our solution by taking the first $4((t+1)^2+t-p')$ vertices of $P'_{i,j}$.

So far, we have shown that the problem is equivalent to stopping each path P_i and each

path $P_{i,j}$ such that (1) none block is only partially taken (2) the number of vertices colored by $c_{i,j,+}$ is equal to the number of vertices colored by $c_{i,j,-}$ for any $i < j \in [k]$. As, we were first forced to take $\{v\} \cup \bigcup_{j \in [k]} P_{j,m}$, the number of vertices colored by $c_{i,j,+}$ was *initially* greater than the number of vertices colored by $c_{i,j,-}$. The only *legal* amounts of vertices colored by $c_{i,j,-}$ (recall that we cannot stop in the middle of a block $B^{i,j}(s)$) are $d_1^{i,j}$, $d_1^{i,j} + d_2^{i,j}$ up to $\Sigma_{l \in [|E_{i,j}|]} d_l^{i,j}$. By construction, those values correspond to the number of vertices colored by $c_{i,j,+}$ if we stop path P_i after block $B(v_h^i)$ and path P_j after block $B(v_{h'}^j)$ such that $v_h^i v_{h'}^j \in E_{i,j}$. Again, the keypoint is that the mapping $(h, h') \in [t]^2 \mapsto (h'+1)(t+1)+h$ is one-to-one. To equalize the number of vertices colored by $c_{i,j,+}$ and $c_{i,j,-}$ for each $i < j \in [k]$, one therefore has to find a multicolored clique C in H and stop each path P_l just after the vertex of color l in C (for any $l \in [k]$). Conversely, if there is such a multicolored clique C in graph H, then stopping the paths P_is just after the block of the one vertex in C, leads to a solution, as explained in the previous paragraphs. Thus, there is a solution to the Graph Motif instance iff there is a multicolored k-clique in H. ◀