

Timeline Problems in Temporal Graphs: Vertex Cover vs. Dominating Set

Anton Herrmann ✉ 

Algorithmics and Computational Complexity, Technische Universität Berlin, Germany

Christian Komusiewicz ✉ 

Institute of Computer Science, Friedrich Schiller University Jena, Germany

Nils Morawietz ✉ 

Institute of Computer Science, Friedrich Schiller University Jena, Germany

LaBRI, Université de Bordeaux, France

Frank Sommer ✉ 

Institute of Logic and Computation, TU Wien, Austria

Institute of Computer Science, Friedrich Schiller University Jena, Germany

Abstract

A temporal graph is a finite sequence of graphs, called snapshots, over the same vertex set. Many temporal graph problems turn out to be much more difficult than their static counterparts. One such problem is TIMELINE VERTEX COVER (also known as $\text{MINTIMELINE}_\infty$), a temporal analogue to the classical VERTEX COVER problem. In this problem, one is given a temporal graph $\mathcal{G}$ and two integers k and ℓ , and the goal is to cover each edge of each snapshot by selecting for each vertex at most k activity intervals of length at most ℓ each. Here, an edge uv in the i th snapshot is covered, if an activity interval of u or v is active at time i . In this work, we continue the algorithmic study of TIMELINE VERTEX COVER and introduce the TIMELINE DOMINATING SET problem where we want to dominate all vertices in each snapshot by the selected activity intervals.

We analyze both problems from a classical and parameterized point of view and also consider partial problem versions, where the goal is to cover (dominate) at least t edges (vertices) of the snapshots. With respect to the parameterized complexity, we consider the temporal graph parameters vertex-interval-membership-width (vimw) and interval-membership-width (imw). We show that all considered problems admit FPT-algorithms when parameterized by $\text{vimw} + k + \ell$. This provides a smaller parameter combination than the ones used for previously known FPT-algorithms for TIMELINE VERTEX COVER. Surprisingly, for $\text{imw} + k + \ell$, TIMELINE DOMINATING SET turns out to be easier than TIMELINE VERTEX COVER, by also admitting an FPT-algorithm, whereas the vertex cover version is NP-hard even if $\text{imw} + k + \ell$ is constant. We also consider parameterization by combinations of n , the vertex set size, with k or ℓ and parameterization by t . Here, we show for example that both partial problems are fixed-parameter tractable for t which significantly improves and generalizes a previous result for a special case of PARTIAL TIMELINE VERTEX COVER with $k = 1$.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Parameterized complexity and exact algorithms; Theory of computation $\rightarrow$ Graph algorithms analysis

Keywords and phrases NP-hard problem, FPT-algorithm, interval-membership-width, Color coding

Related Version Some of the results of this work are also contained in the first author's Master's thesis [23].

Funding *Nils Morawietz*: Supported by the French ANR, project ANR-22-CE48-0001 (TEMPOGRAL).

Frank Sommer: Supported by the Alexander von Humboldt Foundation.

1 Introduction

A crucial task in the management of wireless sensor networks is to monitor the network by selecting few dedicated sensors that can monitor themselves and other sensors which are close enough to have a direct wireless connection [29, 32]. In graph-theoretic terms, this is the famous NP-hard DOMINATING SET problem where we say that a vertex dominates itself and all its neighbors and the task is to select few vertices of the graph such that every vertex is dominated by some selected vertex. In some applications of sensor networks, the network may change over time. Then, instead of a static graph G the input would be a *temporal graph* $\mathcal{G}$, consisting of a set of *snapshots* G_i , each reflecting the connections at timestep i . Moreover, in such a scenario each sensor could carry out the surveillance only for a bounded duration, for example due to limited battery capacities. Thus, instead of selecting few sensors to monitor the network, we would like to select, for each sensor, an active time interval of bounded length ℓ , during which it monitors itself and all its current neighbors. To make the model more general, we may also allow for each sensor to select up to k such active intervals, for example because each sensor carries k batteries. By calling a collection of active intervals for all vertices of the graph a *k -activity ℓ -timeline*, the scenario described above corresponds to the following problem.

TIMELINE DOMINATING SET (TIMELINE DS)

Input: A temporal graph $\mathcal{G}$ and integers $k \geq 1, \ell \geq 0$.

Question: Is there a k -activity ℓ -timeline $\mathcal{T}$ which dominates all temporal vertices of $\mathcal{G}$?

For an example input and solution for TIMELINE DS, refer to [Figure 1](#). As in the static DOMINATING SET problem, we may further generalize the problem to handle scenarios where we are not able to monitor the whole network over the full time period but instead we want to maximize the number of monitored sensors under the resource limitations.

TIMELINE PARTIAL DOMINATING SET (TIMELINE PDS)

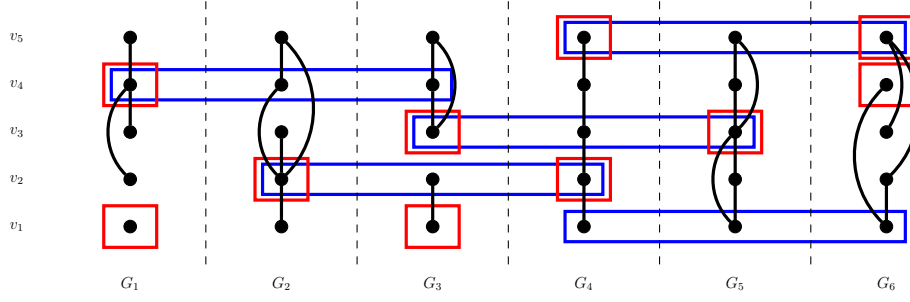
Input: A temporal graph $\mathcal{G}$ and integers $k \geq 1, \ell \geq 0, t \geq 0$.

Question: Is there a k -activity ℓ -timeline $\mathcal{T}$ which dominates at least t temporal vertices of $\mathcal{G}$?

In this work, we initialize the study of TIMELINE (PARTIAL) DOMINATING SET in particular from a computational complexity perspective. As we show, TIMELINE (PARTIAL) DOMINATING SET is NP-hard. To cope with this intractability, we consider mostly the parameterized complexity of the problem, with a particular focus on structural parameterizations of temporal graphs.

The idea to consider timelines in a temporal graph setting is not new: Rozenshtein et al. [30] introduced it in TIMELINE VERTEX COVER,¹ the second main problem studied in this work.

¹ Rozenshtein et al. denote this problem as $\text{MINTIMELINE}_\infty$.



■ **Figure 1** The short red boxes represent a 2-activity 0-timeline dominating all temporal vertices and the long blue boxes represent a 1-activity 2-timeline covering all temporal edges. Observe that the interval length is defined in such a way that an interval starting and ending in the same snapshot has length 0.

TIMELINE VERTEX COVER (TIMELINE VC)

Input: A temporal graph $\mathcal{G}$ and integers $k \geq 1, \ell \geq 0$.

Question: Is there a k -activity ℓ -timeline $\mathcal{T}$ which covers all temporal edges of $\mathcal{G}$?

For an example input and solution for TIMELINE VC, refer again to [Figure 1](#). The model behind TIMELINE VC is that edges in a temporal graph arise only when at least one of their endpoints is active and the task of the problem is to provide an explanation of all edges such that the vertices are only active a few times, each time only for a short duration [30]. Dondi et al. [13] later analyzed the partial variant of the problem.

TIMELINE PARTIAL VERTEX COVER (TIMELINE PVC)

Input: A temporal graph $\mathcal{G}$ and integers $k \geq 1, \ell \geq 0, t \geq 0$.

Question: Is there a k -activity ℓ -timeline $\mathcal{T}$ which covers at least t temporal edges of $\mathcal{G}$?

We continue the study of TIMELINE (PARTIAL) VERTEX COVER with two aims: First, some of the parameters considered in our study have not yet been studied for this problem. Second, we seek a comprehensive complexity overview for these two fundamental timeline problems, highlighting their similarities and differences.

Previous results. Rozenshtein et al. [30] showed that TIMELINE VC is NP-hard in general, but solvable in polynomial time for $k = 1$. Froese et al. [19] continued the algorithmic study of TIMELINE VC. They proved NP-hardness even if the input consists of at most three snapshots, $k = 2$ and $\ell = 0$ and studied the parameterized complexity with respect to different parameters. Froese et al. [19] showed fixed-parameter tractability for $n + k$ and W[1]-hardness for $n + \ell$. Herein, n is the number of vertices of the underlying graph which in turn is the static graph obtained by taking the union of all edge sets. Moreover, TIMELINE VC admits an XP-algorithm for n and is fixed-parameter tractable with respect to n if $\ell = 0$ [19]. Schubert [31] later analyzed the parameterized complexity of TIMELINE VC with respect to combinations of input parameters with structural parameters of the underlying graph. Finally, Dondi et al. [13] obtained two results for TIMELINE PVC with $k = 1$: First, it was shown that this case is NP-hard, in contrast to TIMELINE VC. Second, Dondi et al. [13] gave

FPT-algorithms for parameterization by the number of covered edges and by the number of uncovered edges.

Our results. We start off with a new hardness result for TIMELINE VC: We show that the problem remains NP-hard even if the total number of snapshots T , k , and ℓ are constant and additionally the maximum degree in every snapshot is at most one. Then, we show the NP-hardness of TIMELINE DS, again for the case that T , k , and ℓ are constant and the maximum snapshot degree is one.

We then study the parameterized complexity of the problems; an overview of the results is given in Table 1. As noted above, TIMELINE VC is fixed-parameter tractable with respect to $n + k$ [19]. Since n is a rather large parameter, Froese et al. [19] asked whether there are FPT-algorithms also for parameters that are smaller than n . We make progress in this direction by considering the two parameters *interval-membership-width* (imw) and *vertex-interval-membership-width* (vimw) which were introduced by Bumpus and Meeks [3]. The parameters imw and vimw are interesting in the sense that they are among the relatively few truly temporal structural graph parameters because they are sensitive to the order of the snapshots [3]. Informally, vimw can be defined as follows: We say the *lifetime* of a vertex is the time interval between its first and last non-isolated appearance in a snapshot. For a timestep i , the bag of i is the set of vertices whose lifetime contains timestep i , and vimw is the size of the largest bag of $\mathcal{G}$.

We first show that TIMELINE PVC is fixed-parameter tractable with respect to $\text{vimw} + k + \ell$. To further improve on this, we introduce a hierarchy of parameters with non-increasing size. The idea, inspired by the h -index of static graphs [17], is that a temporal graph may have only few large bags. In that case, we would rather parameterize by the size of the x th largest bag, denoted by $\text{vimw}[x]$, for some $x > 1$. With this definition, we have $\text{vimw} = \text{vimw}[1]$ and $\text{vimw}[x] \geq \text{vimw}[x + 1]$ for all $x \in [T]$. We show that TIMELINE PVC is fixed-parameter tractable for $\text{vimw}[x] + k + \ell$ with $x = k \cdot (\ell + 1)$. Informally, this means that the larger k and ℓ get, the more large bags can be ignored for the parameter. We then consider TIMELINE (PARTIAL) DOMINATING SET in the same setting. We first showing fixed-parameter tractability of TIMELINE PDS for $\text{vimw} + k + \ell$. Then we show that for TIMELINE DS it can be improved to fixed-parameter tractability for $\text{vimw}[x] + k + \ell$ for $x = \ell + 2$ while for TIMELINE PDS parameterization by $\text{vimw}[2] + k + \ell$ is not useful. The latter result is obtained by showing NP-hardness of the extremely restricted special case when there are two snapshots, one of which is edgeless, $k = 1$ and $\ell = 0$.

Afterwards, we consider the interval-membership-width (imw). Informally, this is the edge-version of vimw, that is, the bag at timestep i contains the edges which occur at timestep i and those that occur both before and after i . Note that imw can be considered to be a smaller parameter than vimw: For all instances, we have $\text{vimw} \in \Omega(\sqrt{\text{imw}})$ and there are instances with constant values of imw and unbounded values of vimw. We show that TIMELINE DS is fixed-parameter tractable for the parameter $\text{imw} + k + \ell$ while all three other problems are NP-hard for constant values of $\text{imw} + k + \ell$. Given the latter hardness results, we refrain from analyzing a hierarchy of imw-based parameters. Altogether, the results show that in our setting, vimw is a much more powerful parameter than imw.

We then conclude by considering the more standard parameters n , k , ℓ , and t , where t denotes the number of vertices to dominate or edges to cover, respectively. For TIMELINE DS, we show fixed-parameter tractability for $n + k$ and W[1]-hardness for $n + \ell$, thus showing that the complexity is similar to the one of TIMELINE VC. Finally, we show that TIMELINE PDS and TIMELINE PVC can be solved in $2^{\mathcal{O}(t)} \cdot (n + T)^{\mathcal{O}(1)}$ time. This improves and extends a

■ **Table 1** Overview of our results on the classic and parameterized complexity for TIMELINE (PARTIAL) VERTEX COVER and TIMELINE (PARTIAL) DOMINATING SET. Herein, vimw denotes the vertex-interval-membership-width, imw denotes the interval-membership-width, and vimw[x] denotes the size of the x -largest bag in the vertex-interval-membership sequence. Recall that vimw = vimw[1].

Parameter	TIMELINE VC	TIMELINE PVC	TIMELINE DS	TIMELINE PDS
$\text{vimw} + k + \ell$	FPT Thm. 4.1	FPT Thm. 4.1	FPT Thm. 4.3	FPT Thm. 4.3
$\text{imw} + k + \ell$	NP-h Thm. 4.7	NP-h Thm. 4.7	FPT Thm. 4.6	NP-h Thm. 4.8
$k + \ell + \text{vimw}[x]$	$x = k(\ell + 1) + 1$ FPT Prop. 4.2	$x = k(\ell + 1) + 1$ FPT Prop. 4.2	$x = \ell + 2$ FPT Prop. 4.4	$x = 2$ NP-h Thm. 4.5
$n + k$	FPT [19, Thm. 8]	?	FPT Thm. 5.1	?
$n + \ell$	W[1]-h [19, Thm. 12]	W[1]-h [19, Thm. 12]	W[1]-h Thm. 5.2	W[1]-h Thm. 5.2
t	-	FPT Cor. 5.6	-	FPT Thm. 5.5

previous result of Dondi et al. [13, Thm. 6] who provided an algorithm for TIMELINE PVC with $k = 1$ that has running time $2^{t \cdot \log(t)} \cdot (n + T)^{\mathcal{O}(1)} = t^t \cdot (n + T)^{\mathcal{O}(1)}$.

Further related work. Akrida et al. [1] introduced the problems TEMPORAL VERTEX COVER (TVC) and SLIDING WINDOW TEMPORAL VERTEX COVER (SW-TVC), where the goal is to find a minimum number of temporal vertices that cover the edges of the temporal graph. In TVC each edge needs to be covered in at least one of the snapshots and in SW-TVC, the input contains an integer Δ , and the aim is to cover every edge at least once at every Δ consecutive time steps. Akrida et al. [1] showed that TVC remains NP-hard when the underlying graph is a star graph. Moreover, they studied the approximability of both problems, provided (S)ETH-based running time lower bounds and designed an exact dynamic programming algorithm with exponential running time. The research on these problems was then continued by Hamm et al. [22], who showed that TVC is solvable in polynomial time if the underlying graph is a path or cycle while SW-TVC remains NP-hard in this case. Besides that, they provided several (exact and approximation) algorithms for the sliding window variant. The TEMPORAL DOMINATING SET (TDS) problem is defined analogously, here the task is to dominate every vertex in at least one of the snapshots of the temporal graph by selecting few temporal vertices. TDS is NP-hard in very restricted cases, for example when the maximum degree in every snapshot is 2 [24]. There are further temporal variants of VERTEX COVER and DOMINATING SET which have been discussed and studied. These include a multistage variant of VERTEX COVER [18], a reachability-based variant for DOMINATING SET [25] and different conditions on how and when a vertex should be dominated [4, 33].

2 Preliminaries

We denote the set of integers $\{i, i+1, \dots, j-1, j\}$ by $[i, j]$, and $[1, i]$ simply by $[i]$. Given a set V and an integer $k > 0$, we write $\binom{V}{k}$ for the collection of all size- k subsets of V .

Graph theory. A (static) graph $G = (V, E)$ consists of a set of *vertices* V and a set of edges $E \subseteq \binom{V}{2}$. We denote by $V(G)$ and $E(G)$ the vertex and edge set of G , respectively. Furthermore, we let $n := |V(G)|$ and $m := |E(G)|$. For an edge $\{u, v\}$ we write uv and call u and v *endpoints* of the edge. Further, we say that the vertex $v \in V$ is *incident* with $e \in E$ if $v \in e$. If $uv \in E$, then u and v are *adjacent* and they are *neighbors* of each other. The set $N_G(v) := \{u \in V : u \text{ and } v \text{ are adjacent}\}$ is the (*open*) *neighborhood* of v . Moreover, $N_G[v] := N_G(v) \cup \{v\}$ is the *closed neighborhood* of v . For a set of vertices S , we let $N_G[S] := \bigcup_{s \in S} N_G[s]$ and $N_G(S) = N_G[S] \setminus S$. By $\deg_G(v) := |N_G(v)|$ we denote the *degree* of v . If $\deg_G(v) = 0$, we say that v is *isolated*. The *maximum degree* is $\Delta(G) := \max_{v \in V} \deg_G(v)$. For vertex sets S and T , we let $E_G(S, T) := \{uv \in E(G) : u \in S \text{ and } v \in T\}$ and we let $E(S) := E(S, S)$. A graph G is *bipartite* if $V(G)$ can be partitioned into two sets S and T such that $E(G) = E(S, T)$.

A *star* is a bipartite graph where $|S| = 1$. We may drop the subscript $\cdot_G$ when it is clear from context. A *path* in G is a sequence of distinct vertices $(v_1, v_2, \dots, v_{z+1})$ such that $v_i v_{i+1} \in E$ for each $i \in [z]$. In this case we say that each vertex v_i is *connected* to each vertex v_j .

For more details on graph theory, we refer to the book by Diestel [9].

Temporal graphs. A *temporal graph* $\mathcal{G}$ is a finite sequence of graphs $(G_1, \dots, G_T)$ which all have the same set of vertices $V(\mathcal{G}) := V(G_1) = \dots = V(G_T)$. The graphs G_i are called *snapshots*, T is the *lifetime* of $\mathcal{G}$, and $i \in [T]$ is a *time step*. We may write V instead of $V(\mathcal{G})$ and E_i instead of $E(G_i)$. Moreover, for $u, v \in V$ and $i \in [T]$, (v, i) is a *temporal vertex* and $(\{u, v\}, i) = (uv, i)$ with $uv \in E_i$ is a *temporal edge* in snapshot G_i . We say edge e *appears* in G_i if $e \in E_i$. Moreover, we say that a vertex v is *incident* with the temporal edge (e, i) if $v \in e$. For a temporal vertex (v, i) , we define the (*open*) *neighborhood* by $N_{\mathcal{G}}((v, i)) := N_{G_i}(v)$ and the *closed neighborhood* by $N_{\mathcal{G}}[(v, i)] := N_{G_i}[v]$. By $G_{\downarrow}(\mathcal{G}) := (V_{\downarrow} := V(\mathcal{G}), E_{\downarrow} := \bigcup_{i=1}^T E_i)$ we denote the *underlying graph* of $\mathcal{G}$. We let $\Delta_i(\mathcal{G}) = \Delta(G_i)$. Furthermore, by $\Delta_{\max}(\mathcal{G}) := \max_{i=1}^T \Delta_i(\mathcal{G})$ we denote the *maximum snapshot degree* of $\mathcal{G}$. If $E_i = \emptyset$ we say that G_i is *empty*. We may drop the subscript $\cdot_{\mathcal{G}}$ when it is clear from context.

Parameter definitions. We give the precise definitions of the parameters related to the (vertex)-interval-membership width: vimw , $\text{vimw}[x]$, and imw .

Let $\mathcal{G} = (G_1, \dots, G_T)$ be a temporal graph. The *vertex-interval-membership sequence* of $\mathcal{G}$ is the sequence $(F_i)_{i \in [T]}$ of vertex-subsets $F_i \subseteq V(\mathcal{G})$ (called *bags*), where each F_i is defined as

$$F_i := \{v \in V(\mathcal{G}) : \exists p, q \in [T] \text{ such that } p \leq i \leq q, \deg_{G_p}(v) \geq 1 \text{ and } \deg_{G_q}(v) \geq 1\}.$$

The *vertex-interval-membership-width* of $\mathcal{G}$, denoted by $\text{vimw}(\mathcal{G})$, is the maximum size of a bag in the vertex-interval-membership sequence, that is, $\text{vimw}(\mathcal{G}) := \max\{|F_i| : i \in [T]\}$.

For given $x \in [T]$ we introduce the parameter $\text{vimw}[x]$ as the size of the x th largest bag of the vertex-interval-membership sequence. Formally, let $(F_{i_1}, \dots, F_{i_T})$ be an ordering

of $(F_i)_{i \in [T]}$ such that $|F_{i_1}| \geq \dots \geq |F_{i_T}|$. Then we define

$$\text{vimw}[x](\mathcal{G}) := |F_{i_x}|.$$

Note that $\text{vimw}[i]$ can be much smaller than $\text{vimw}[1] = \text{vimw}$: for example consider a temporal graph where every vertex only appears non-isolated for a short period of time. If all snapshots have only a few edges, then vimw is small. However, a single snapshot with many edges is enough to make vimw arbitrarily large. If only a few such snapshots exist, then $\text{vimw}[i]$ is already much smaller than vimw for small i .

The *interval-membership sequence* of $\mathcal{G}$ is the sequence $(F_i)_{i \in [T]}$ of edge-subsets $F_i \subseteq E_\downarrow$ (again called *bags*), where each F_i is defined as

$$F_i := \{e \in E_\downarrow : \exists p, q \in [T] \text{ such that } p \leq i \leq q \text{ and } e \in E_p \cap E_q\}.$$

The *interval-membership-width* of $\mathcal{G}$, denoted by $\text{imw}(\mathcal{G})$, is the maximum size of a bag in the interval-membership sequence, that is, $\text{imw}(\mathcal{G}) := \max\{|F_i| : i \in [T]\}$.

We remark that $\text{vimw} = \Omega(\sqrt{\text{imw}})$ and that there exist temporal graphs with $\text{imw} = 1$ but arbitrary large vimw [3]. Finally, note that the vertex-interval-membership sequence and the interval-membership sequence can be computed in polynomial time with respect to the input size.

Timelines. Given a temporal graph $\mathcal{G} = (G_1, \dots, G_T)$, an *activity interval* of a vertex v is a triple $(v, a, b) \in V(\mathcal{G}) \times [T] \times [T]$ such that $a \leq b$. We say a that is the *starting time* and b is the *end time* of an activity interval (v, a, b) . The *length* of the activity interval is defined by $b - a$. A *k-activity timeline* of the temporal graph $\mathcal{G}$ is a set of activity intervals $\mathcal{T} \subseteq \{(v, a, b) \in V(\mathcal{G}) \times [T] \times [T] : a \leq b\}$, which contains at most k activity intervals for each vertex. A *k-activity ℓ -timeline* is a k -activity timeline in which each activity interval has length at most ℓ . Given a k -activity timeline $\mathcal{T}$, a vertex v is called *active* in snapshot G_i , if there exists $(v, a, b) \in \mathcal{T}$ such that $a \leq i \leq b$. We say that the activity interval (v, a, b) *intersects* snapshot G_i , if $a \leq i \leq b$. A k -activity timeline $\mathcal{T}$ *dominates* the temporal vertex (v, i) if $v \in N_{G_i}[\{u : u \text{ is active in } G_i\}]$. A k -activity timeline $\mathcal{T}$ *covers* the temporal edge (e, i) if at least one of the two endpoints of e is active in snapshot G_i .

Parameterized complexity. Let $L \subseteq \Sigma^*$ be a computational problem specified over some alphabet Σ and let $p : \Sigma^* \rightarrow \mathbb{N}$ be a *parameter*, that is, p assigns to each instance of L an integer parameter value (which we simply denote by p if the instance is clear from the context). We say that L is *fixed-parameter tractable (FPT) with respect to p* if L can be decided in $f(p) \cdot |I|^{\mathcal{O}(1)}$ time where $|I|$ is the length of the input. The corresponding hardness concept related to fixed-parameter tractability is $\text{W}[t]$ -hardness, $t \geq 1$; if a problem L is $\text{W}[t]$ -hard with respect to p , then L is assumed to *not* be fixed-parameter tractable. Moreover, L is *slice-wise polynomial (XP) with respect to p* if L can be decided in $f(p) \cdot |I|^{g(p)}$ time. Let (I, p) and (I', p') be two instances of L . A *reduction to a (problem) kernel for L* is a polynomial-time algorithm that computes an instance (I', p') such that

- $p' + |I'| \leq g(p)$, and
- (I, p) is a yes-instance of L if and only if (I', p') is a yes-instance of L .

The instance (I', p') is referred to as *the problem kernel* and g is the *size* of the kernel. Furthermore, if g is a polynomial, we say that the kernel is a *polynomial problem kernel*. For more details about parameterized complexity we refer to the standard monographs [7, 15].

3 NP-Hardness Results

We show NP-hardness for TIMELINE VC and TIMELINE DS even when they are restricted to temporal graphs with a maximum snapshot degree of 1 and constant lifetime T , interval number k , and interval length ℓ . First, we present the reduction for TIMELINE VC.

► **Theorem 3.1.** *TIMELINE VC is NP-hard even if $T = 23, k = 2, \ell = 4$, and the maximum snapshot degree is one.*

Proof. We reduce from the NP-hard problem 3-COLORING on graphs with maximum degree four [21]. In 3-COLORING the input is a graph and the question is whether the vertices can be colored with three colors such that no two adjacent vertices have the same color.

Intuition: The basic idea is to construct a temporal graph with three parts C_1, C_2 and C_3 (call them *color blocks*), each consisting of five snapshots and representing one of the three colors. The part in which a vertex v is *not active* corresponds to the assigned color of v . As each of the three parts will contain every edge of G and only vertices of degree at most one, this ensures that all neighbors are active in the part where v is not active. Hence, they are assigned a different color. The basic idea for encoding a given graph G into snapshots of maximum degree one is to find a proper edge coloring with five colors and split the edge set with respect to this coloring, which can be done in polynomial time via Vizing's theorem. Our aim is that no activity interval of any vertex can hit more than one color block. To ensure this, we add sufficiently many empty snapshots between any two parts.

Construction: Let $(G = (V = \{v_1, \dots, v_n\}, E))$ be an instance of 3-COLORING on graphs with maximum degree four. We construct the following temporal graph $\mathcal{G}$ where $V(\mathcal{G}) = V$.

The temporal graph $\mathcal{G}$ consists of 23 snapshots. It can be seen as a sequence of five parts, namely C_1, H_1, C_2, H_2, C_3 . Parts H_1 and H_2 consist of 4 empty snapshots and each of C_1, C_2, C_3 consists of 5 snapshots. Thus, snapshots 6 to 9 and 15 to 18 are empty. We denote H_1 and H_2 as the *empty blocks* and we call C_1, C_2 and C_3 *color blocks* as they correspond to the three colors. In the following we formally define the snapshots of the color blocks. All three color blocks C_1, C_2 and C_3 will be identical. Hence, we only formally define C_1 , that is, snapshots 1 to 5.

Here we exploit the fact that the graph G has maximum degree four, because this implies that there exists a proper edge coloring with at most five colors due to Vizing's theorem [34], which can be computed in polynomial time [27]. Let $E(G) = F_1 \cup \dots \cup F_5$ be the partition of $E(G)$ into five colors of some proper edge coloring. Now, snapshot j contains exactly the edges of F_j . Finally, observe that by construction, the maximum degree in each snapshot is at most one. By setting $k = 2$ and $\ell = 4$, we obtain the TIMELINE VC instance $(\mathcal{G}, k, \ell)$.

Correctness: We show that G is 3-colorable if and only if $\mathcal{G}$ has a 2-activity 4-timeline $\mathcal{T}$ covering all temporal edges of $\mathcal{G}$. In the following we say a vertex v is *active* during C_j for $j \in [3]$ to indicate that an activity interval of v starts at the first time step and ends and the last time step of the corresponding part.

($\Rightarrow$) Let $\phi : V(G) \rightarrow [3]$ be a proper 3-coloring of G . We construct an activity timeline of $\mathcal{G}$ which covers all temporal edges. For each $i \in [n]$, the vertex $v \in V(G)$ is active in the two color blocks, which do not correspond to the assigned color of v . Clearly, this yields a 2-activity timeline $\mathcal{T}$. It remains to argue that each temporal edge is covered by $\mathcal{T}$. Since ϕ is a proper 3-coloring, both endpoints of each edge $uw \in E(G)$ have a different color. Moreover, since in color block C_j all vertices are active which do not have color j , we conclude that all temporal edges of $\mathcal{G}$ are covered by $\mathcal{T}$.

($\Leftarrow$) Let $\mathcal{T}$ be a 2-activity 4-timeline covering all temporal edges of $\mathcal{G}$. Observe that since parts H_1 and H_2 corresponding to snapshots 6 to 9 and 15 to 18 are empty, no activity

interval of any vertex can hit more than one color block. Consequently, since $k = 2$, for each vertex v there exists at least one color $j \in [3]$ such that v is not active during any snapshot of color block C_j . Let $\phi(v) \in [3]$ be such a color. Note that if $\phi(v)$ is not unique, we pick any color fulfilling the property. We now argue that ϕ is a proper 3-coloring of G . Assume towards a contradiction that this is not the case, that is, there exists at least one edge $uw \in E(G)$ such that both endpoints u and w have the same color j . This implies that both u and w are not active during C_j . But since exactly one of the 5 snapshots corresponding to color block C_j contains the edge uw , timeline $\mathcal{T}$ is not covering all temporal edges of $\mathcal{G}$, a contradiction. Thus, ϕ is a proper 3-coloring of G . ◀

Next, we extend the ideas of this reduction to TIMELINE DS for which the reduction is more involved since we also need to deal with isolated temporal vertices.

► **Theorem 3.2.** *TIMELINE DS is NP-hard even if $T = 35, k = 3, \ell = 6$ and the maximum snapshot degree is one.*

Proof. We reduce from 3-COLORING on graphs with maximum degree four. This problem is known to be NP-hard [21]. In 3-COLORING the input is a graph and the question is whether the vertices can be colored with three colors such that no two adjacent vertices have the same color. Figure 2 sketches the following reduction. We start by giving an intuition.

Intuition: The basic idea is to construct a temporal graph with three parts C_1, C_2 and C_3 (call them *color blocks*), each consisting of seven snapshots and representing one of the three colors. The part in which a vertex v is not active corresponds to the color assigned to v . As each of the three parts will contain every edge of G and only vertices of degree at most one, this ensures that all neighbors are active in the part where v is not active. Hence, they are assigned a different color. The basic idea for encoding a given graph G into snapshots of maximum degree one is to find a proper edge coloring and split the edge set with respect to this coloring. In the following reduction we unfortunately need in addition to C_1, C_2 and C_3 another part H consisting of 14 snapshots and some additional vertices. This leads to $k = 3$ for the following reasons. On the one hand we need to ensure that some activity intervals of the vertices coincide exactly with the color blocks. On the other hand we need to take care of possible isolated vertices in the snapshots.

Construction: Suppose $(G = (V = \{v_1, \dots, v_n\}, E))$ is a given instance of 3-COLORING on graphs with maximum degree four. We construct the following temporal graph $\mathcal{G}$. For each vertex $v_i \in V(G)$ we introduce five vertices. Formally, we set

$$V(\mathcal{G}) := \bigcup_{i=1}^n \{v_i, v'_i, u_i, u'_i, u''_i\}.$$

The vertex v_i represents the vertex v_i from G in some snapshots of the temporal graph. The vertex v'_i ensures that two activity intervals of v_i coincide exactly with two of the three color blocks and that v_i is not isolated in any of the snapshots of the color blocks. The vertices u_i, u'_i and u''_i ensure that v'_i is not isolated in any of the color blocks. However, the introduction of new vertices also leads to an increasing number of activity intervals. The later introduced part H of our temporal graph forces us to use them.

The temporal graph $\mathcal{G}$ consists of 35 snapshots. It can be seen as a sequence of four parts, namely H, C_1, C_2 and C_3 . Part H consists of 14 snapshots and each of C_1, C_2, C_3 consists of seven snapshots. We call C_1, C_2 and C_3 *color blocks* as they correspond to the three colors. In the following we formally define the snapshots.

The first part of the temporal graph is the part H in which all snapshots have the same set of edges. We set

$$E_1 = \dots = E_{14} := \bigcup_{i=1}^n \{v_i u_i, u'_i u''_i\}$$

while v'_i remains isolated in all these snapshots for each $i \in [n]$.

The three color blocks C_1, C_2 and C_3 are very similar to each other. We formally define C_1 and state how C_2 and C_3 differ from C_1 . The first snapshot (say L_1) and last snapshot (say R_1) of C_1 are equal. They ensure that some activity intervals coincide exactly with the color block. The five snapshots between L_1 and R_1 encode our graph G with snapshots of maximum degree one. Here we need the fact that our graph G has maximum degree four, because this implies that there exists a proper edge coloring with at most five colors due to Vizing's theorem [34], which can be computed in polynomial time [27]. Let $E(G) = F_1 \cup \dots \cup F_5$ be the partition of $E(G)$ into five colors of some proper edge coloring. With this we are now ready to formally define the snapshots of C_1 . Recall that the first 14 snapshots are part of H , so the first and last snapshot of C_1 are G_{15} and G_{21} respectively. We define

$$E_{15} = E_{21} := \bigcup_{i=1}^n \{v_i v'_i, u'_i u''_i\}.$$

The five snapshots between them encode G , ensure that v_i and v'_i are not isolated and contain the edge $u'_i u''_i$. Formally, we define for each $j \in [5]$ the edge set of G_{15+j} by

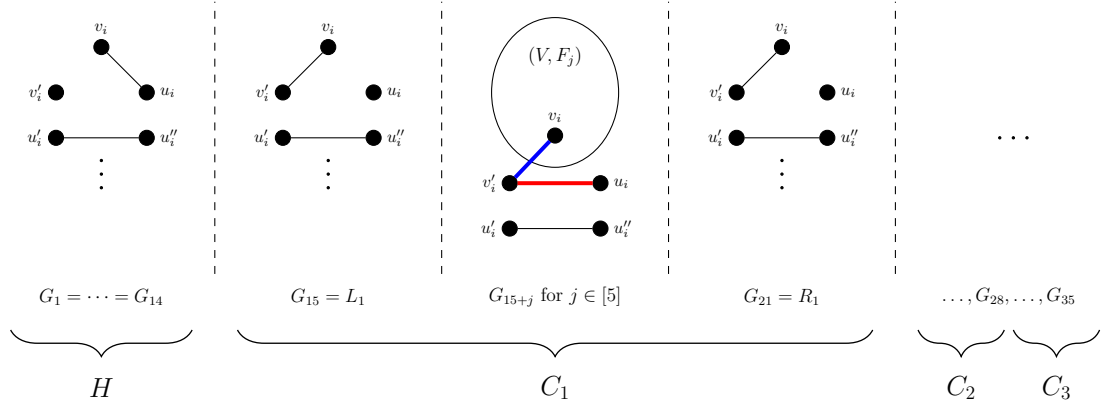
$$E_{15+j} := F_j \cup \bigcup_{i=1}^n \{u'_i u''_i\} \cup E_{i,j}$$

where we set $E_{i,j} = \{v_i v'_i\}$ if v_i is an isolated vertex in the graph $(V(G), F_j)$, and $E_{i,j} = \{v'_i u_i\}$, otherwise.

As mentioned above, C_2 and C_3 are very similar to C_1 . Both of them consist of seven snapshots, where the first and last snapshot are equal and the five snapshots between them encode G . The color part C_2 starts at snapshot G_{22} and ends at snapshot G_{28} . The color part C_3 starts at snapshot G_{29} and ends at snapshot G_{35} . The only difference between C_1 and the other color blocks is that u_i swaps roles with u'_i in C_2 and with u''_i in C_3 respectively. The first snapshots of C_2 and C_3 are called L_2 and L_3 respectively and the last snapshots of C_2 and C_3 are called R_2 and R_3 respectively. Finally, observe that by construction, the maximum degree in each snapshot is at most one. By setting $k = 3$ and $\ell = 6$, we obtain the TIMELINE DS instance $(\mathcal{G}, k, \ell)$.

Correctness: We show that G is 3-colorable if and only if there is a 3-activity 6-timeline $\mathcal{T}$ dominating all temporal vertices of $\mathcal{G}$. From now on we call the first part of H , consisting of the first seven snapshots, H_1 and we call the second part, consisting of the other seven snapshots, H_2 . The reason for this is simply because the activity intervals have length six (so they intersect seven snapshots). In the following we say a vertex v is active during H_i ($i \in [2]$) or during C_j ($j \in [3]$) to indicate that an activity interval of v starts at the first time step and ends and the last time step of the corresponding part.

($\Rightarrow$) Suppose $\phi : V(G) \rightarrow [3]$ is a proper 3-coloring of G . We construct an activity timeline $\mathcal{T}$ dominating all temporal vertices of $\mathcal{G}$. For each $i \in [n]$, the vertex v_i is active in the two color blocks, which do not correspond to the assigned color of v_i and the vertex v'_i is active in the color block, which corresponds to the assigned color of v_i . All the remaining activity intervals of v_i, v'_i, u_i, u'_i and u''_i ensure that all of their temporal vertices are dominated. In particular, these remaining activity intervals do not depend on the color of v_i . Informally,



■ **Figure 2** A sketch of the reduction from [Theorem 3.2](#). Let G be a graph with maximum degree at most four, where $E(G) = F_1 \cup \dots \cup F_5$ is a partition of the edges into the colors. The blue edge in the figure exists in G_{15+j} if and only if v_i is isolated in the graph (V, F_j) . Otherwise, the red edge exists. The parts C_2 and C_3 are constructed in a similar way as C_1 , but u_i swaps roles with u'_i and u''_i respectively.

our activity timeline $\mathcal{T}$ is constructed in the following way. For each $i \in [n]$ with $\phi(v_i) = j$, let

- v_i be active in H_1 and in C_r for $r \neq j$,
- v'_i be active in H_1, H_2, C_j ,
- u_i be active in H_2, C_1, C_2 ,
- u'_i be active in H_1, C_2, C_3 ,
- u''_i be active in H_2, C_1, C_3 .

Formally, we define for each color $j \in [3]$ the set $\mathcal{T}_j$ by

$$\mathcal{T}_1 := \bigcup_{\substack{v_i \in V: \\ \phi(v_i)=1}} \{(v_i, 22, 28), (v_i, 29, 35), (v'_i, 15, 21)\},$$

$$\mathcal{T}_2 := \bigcup_{\substack{v_i \in V: \\ \phi(v_i)=2}} \{(v_i, 15, 21), (v_i, 29, 35), (v'_i, 22, 28)\},$$

$$\mathcal{T}_3 := \bigcup_{\substack{v_i \in V: \\ \phi(v_i)=3}} \{(v_i, 15, 21), (v_i, 22, 28), (v'_i, 29, 35)\}$$

and we define for each $i \in [n]$ the set

$$\mathcal{H}_i := \{(v_i, 1, 7), (v'_i, 1, 7), (v'_i, 8, 14), (u_i, 8, 14), (u_i, 15, 21), (u_i, 22, 28) \\ (u'_i, 1, 7), (u'_i, 22, 28), (u'_i, 29, 35), (u''_i, 8, 14), (u''_i, 15, 21), (u''_i, 29, 35)\}.$$

This allows us to define

$$\mathcal{T} := \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3 \cup \bigcup_{i=1}^n \mathcal{H}_i.$$

Note that by construction $\mathcal{T}$ is a 3-activity 6-timeline and it remains to show that $\mathcal{T}$ dominates all temporal vertices in $\mathcal{G}$.

First, consider the part H (the first 14 snapshots). The four vertices v_i, u_i, u'_i and u''_i are dominated in every snapshot of H , because the edges $v_i u_i, u'_i u''_i$ exist in every snapshot of H and $\{(v_i, 1, 7), (u_i, 8, 14), (u'_i, 1, 7), (u''_i, 8, 14)\} \subseteq \mathcal{T}$. The remaining vertex v'_i is dominated in these snapshots, because it is active in all snapshots of them.

Next, we consider the color part C_1 . The two vertices u'_i and u''_i are dominated, because the edge $u'_i u''_i$ exists in every snapshot of C_1 and $(u'_i, 15, 21) \in \mathcal{T}$. The vertex u_i is dominated, because it is active in all of C_1 . For v_i there are two cases. If $\alpha(v_i) = 1$, then v_i is not active in C_1 and v'_i is active in all of C_1 . Consequently, v'_i is dominated in C_1 and v_i is dominated in each snapshot of C_1 either by v'_i or by a neighbor with respect to G . For this note that each neighbor u of v_i in G is active in C_1 , since $\alpha(u) \neq 1$. If $\alpha(v_i) \neq 1$, then v_i is active in C_1 and therefore dominated. In this case the vertex v'_i is not active in C_1 and is dominated in each snapshot of C_1 either by v_i or by u_i . Therefore all temporal vertices in the color block C_1 are dominated by $\mathcal{T}$. The same reasoning holds for the color blocks C_2 and C_3 by swapping the roles of u_i with u'_i and u''_i respectively.

($\Leftarrow$) For the other direction suppose $\mathcal{T}$ is a 3-activity 6-timeline dominating all temporal vertices of $\mathcal{G}$. We want to construct a valid 3-coloring for G . The idea is to assign v_i the color j for which v_i is not active in C_j . For this purpose, we need to show that two of the three activity intervals of v_i correspond to exactly two of the color blocks. We split the proof into smaller parts for more clarity.

Since v'_i is isolated in the first fourteen snapshots, we observe the following.

► **Observation 1.** *At least two activity intervals of v'_i are completely contained in the part H .*

Using **Observation 1**, we can prove the next claim.

▷ **Claim 3.3.** The vertex v_i has exactly one activity interval, which intersects part H . This activity interval either corresponds to the time steps of H_1 or H_2 .

Proof. First, note that in each first snapshot L_j and last snapshot R_j of a color block C_j ($j \in [3]$), one of the two vertices v_i and v'_i needs to be active. By **Observation 1**, the vertex v'_i has at most one activity interval, which intersects the color blocks. Further, observe that each activity interval can intersect at most one snapshot L_j and one snapshot R_j at the same time (and this only if the activity interval does not intersect H), because the length of an activity interval is at most six. This implies that v'_i is active in at most one L_j and in at most R_j . Moreover, it follows that v_i has at most one activity interval which intersects H , because the activity intervals from v_i need to intersect in sum at least two first and two last snapshots of the color blocks and any activity interval intersecting H can only intersect L_1 and no R_j .

Now we show that at most one activity interval of u_i intersects H . For contradiction, assume that two activity intervals of u_i intersect part H . This implies that the remaining third activity interval of u_i has to intersect R_1 , since u_i is isolated in this snapshot (and no activity interval can intersect H and R_1 at the same time). Consequently, u_i is not active in both R_2 and R_3 . Since u_i is adjacent to u'_i and u''_i in R_2 and R_3 respectively, it follows that both u'_i and u''_i need to be active in both R_2 and R_3 (once for dominating u_i and once they are isolated). No single activity interval can intersect both R_2 and R_3 and therefore at most one activity interval from each u'_i and u''_i can intersect the parts H and C_1 . As $u'_i u''_i$ is an isolated edge in all snapshots of H and C_1 , we obtain a contradiction as not every temporal vertex of u'_i and u''_i can be dominated in H and C_1 .

Hence, we conclude that u_i and v_i have at most one activity interval which intersects H . This directly implies that v_i is active either in exactly H_1 or H_2 (recall that $v_i u_i$ is an isolated edge in all snapshots of H). $\triangleleft$

From [Observation 1](#) and [Claim 3.3](#) we conclude that there are in total three activity intervals of v_i and v'_i (two of them are from v_i), which intersect the color blocks. As mentioned above, in each first snapshot L_j and in each last snapshot R_j of a color block C_j , one of the vertices v_i or v'_i needs to be active. Also recall that any activity interval intersects at most one snapshot L_j and at most one snapshot R_j at the same time. Consequently, each of these three remaining activity intervals needs to intersect a different L_j and different R_j than the other ones. This directly implies that the three remaining activity intervals of v_i and v'_i occupy exactly the three color blocks C_1, C_2 and C_3 .

We can define a 3-coloring of G by assigning v_i the color j if and only if v_i is not active in the color block C_j . If v_i is not active in C_j , then by construction, each $w \in N_G(v_i)$ has to be active in C_j (recall that $v_i w$ is an edge in some of the snapshots of C_j and the maximum snapshot degree is one, so v_i has to be dominated by w in some snapshot of the color block C_j) and therefore is assigned a different color than v_i . ◀

4 The Influence of Membership-Width Based Parameters

Now, we investigate the membership-width based parameters vimw and imw . More precisely, we study the parameter combinations $\text{vimw} + k + \ell$ and $\text{imw} + k + \ell$. These combinations are motivated by the fact that **TIMELINE VC** and **TIMELINE DS** are $\text{W}[1]$ -hard with respect to $n + \ell$; for **TIMELINE VC** this follows from previous work [\[19\]](#) and for **TIMELINE DS**, we will show this in [Section 5](#).

4.1 The Parameter Vertex-Interval-Membership-Width

In this section, we provide FPT-algorithms with respect to $\text{vimw} + k + \ell$ for **TIMELINE PVC** and **TIMELINE PDS**. Moreover, we show that vimw can be replaced by smaller parameters $\text{vimw}[x]$ for all problems except **TIMELINE PDS** (the precise value of x differs for **TIMELINE PVC** and **TIMELINE DS**). Simultaneously, these algorithms also are XP-algorithms for **TIMELINE PVC** and **TIMELINE PDS** parameterized by vimw alone. Initially, we focus on **TIMELINE (PARTIAL) VERTEX COVER** and we show that **TIMELINE PVC** admits an FPT-algorithm for $\text{vimw} + k + \ell$.

► **Theorem 4.1.** ***TIMELINE PVC** is solvable in $((k+1)(\ell+2))^{2 \cdot \text{vimw}} \cdot (n+T)^{\mathcal{O}(1)}$ time.*

Proof. Let $(\mathcal{G} = (G_1, \dots, G_T), k, \ell, t)$ be an instance of **TIMELINE PVC**. Further, suppose that $T \geq k \cdot (\ell + 1) + 1$ (otherwise, it is a trivial instance) and that the underlying graph $G_\downarrow$ contains no isolated vertex v . Otherwise, since v does not cover any temporal edge, v can be safely removed and consequently t remains unchanged. Moreover, we assume that each vertex v is contained in at least $k \cdot (\ell + 1) + 1$ bags (recall that these are consecutive) because otherwise we can simply greedily pick k activity intervals of v which contain all snapshots where v is non-isolated and thus we cover all temporal edges incident with v . Thus, it is safe to remove v from $\mathcal{G}$ and reduce t by the number of temporal edges incident with v over all snapshots.

The idea is to use dynamic programming over the lifetime of the temporal graph. Each table entry corresponds to the maximal number of covered temporal edges where a specific set of vertices in the bag is active. In other words, the dynamic program keeps track of the activity of the vertices which are contained in the bag F_i of the currently considered time step $i \in [T]$. While iterating over the lifetime of the temporal graph, we need to ensure that the activities at a time step are compatible with the activities at the neighboring time steps. Moreover, for a vertex v let i and j be the index of the smallest and largest snapshot where v

is incident to an edge, respectively. By definition, $v \in F_x$ for any $i \leq x \leq j$. Furthermore, since v is isolated in all snapshots G_x where $x < i$ or $x > j$, it is safe to assume that all k activity intervals of v are used when v is contained in the bags F_x .

Recall that the vertex-interval-membership sequence can be computed in polynomial time. Since we assumed that the underlying graph contains no isolated vertex, it follows that each vertex is contained in at least one bag of the sequence.

Table Definition: We denote $\overleftarrow{F}_i := F_1 \cup \dots \cup F_i$ and $\overline{F}_i := \overleftarrow{F}_i \setminus F_i$. In other words, $\overline{F}_i$ is the set of vertices which are isolated in all snapshots $G_i, \dots, G_T$ since we assumed that $G_\downarrow$ contains no isolated vertices. Moreover, for $i \in [T]$ we define functions $\text{curr} : F_i \rightarrow [0, k]$, and $\text{pos} : F_i \rightarrow [0, \ell + 1]$. The functions curr and pos indicate for each vertex $v \in F_i$ whether an activity interval of v is active during snapshot G_i . More precisely, the function value $\text{curr}(v)$ determines the number of the current activity interval of v and the function value $\text{pos}(v)$ determines the position in this activity interval. Here, $\text{curr}(v) = 0$ means that the current time step is before the first activity interval and $\text{pos}(v) = 0$ means that v is currently not active, that is, the $\text{curr}(v)$ -th activity interval of v is already over and the next one has not started yet. Now the entry $\text{DP}[i, \text{curr}, \text{pos}]$ contains the maximum number of covered temporal edges in the first i snapshots $G_1, \dots, G_i$, while i is at the $\text{pos}(v)$ -th position of the $\text{curr}(v)$ -th activity interval of $v \in F_i$.

The formal definition of the table now is

$$\text{DP}[i, \text{curr}, \text{pos}] \triangleq \max_{\mathcal{T}} |\{(e, j) \in E_\downarrow \times [i] : (e, j) \text{ is covered by } \mathcal{T}\}|,$$

where the maximum is taken over all k -activity ℓ -timelines

$$\mathcal{T} \subseteq (\overline{F}_i \times [i - 1] \times [T]) \cup (F_i \times [i] \times [T]),$$

such that for each $v \in F_i$

- (i) there are at most $\text{curr}(v)$ activity intervals of v in $\mathcal{T}$,
- (ii) $\text{pos}(v) \leq i$, and
- (iii) $(v, i - \text{pos}(v) + 1, b) \in \mathcal{T}$ for $b = \min(i - \text{pos}(v) + 1 + \ell, T)$ if $\text{pos}(v) > 0$.

We need Conditions (i)-(iii) for the following reasons: Condition (i) ensures that each vertex has at most k activity intervals. Moreover, condition (ii) ensures that no activity interval starts before time step one. Finally, condition (iii) ensures that each activity interval has length ℓ , except if the remaining number of snapshots is too small.

Initialization: For all $\text{curr} : F_1 \rightarrow [0, k]$ and $\text{pos} : F_1 \rightarrow [0, 1]$, for which $\text{curr}(v) = 0$ implies $\text{pos}(v) = 0$, the table is initialized by

$$\text{DP}[1, \text{curr}, \text{pos}] = |E_{G_1}(\{v \in F_1 : \text{pos}(v) = 1\}, F_1)|.$$

Note that $\overline{F}_1 = \emptyset$ and therefore only activity timelines $\mathcal{T} \subseteq F_1 \times [1] \times [T]$ are considered in the definition of $\text{DP}[1, \text{curr}, \text{pos}]$. The initialization is correct, since the table entry $\text{DP}[1, \text{curr}, \text{pos}]$ contains the number of temporal edges which are covered by the active vertices from F_1 in G_1 . Observe that the other endpoints of the covered edges are also contained in F_1 by definition.

Recurrence: The remaining table entries are computed recursively. Intuitively, $\text{DP}[i, \text{curr}, \text{pos}]$ is computed by trying all activity interval choices for the vertices of F_{i-1} , such that there is no conflict with curr and pos . Formally, we set

$$\text{DP}[i, \text{curr}, \text{pos}] = \max_{\text{curr}', \text{pos}'} \text{DP}[i - 1, \text{curr}', \text{pos}'] + |E_{G_i}(\{v \in F_i : \text{pos}(v) \geq 1\}, F_i)|$$

where the maximum is taken over all $\text{curr}' : F_{i-1} \rightarrow [0, k]$ and $\text{pos}' : F_{i-1} \rightarrow [0, \ell + 1]$ which are *compatible* with curr and pos . Informally, curr' and pos' are *compatible* with curr and pos if they provide a correct extension of the activities of $F_{i-1} \cap F_i$ at time step i to time step $i - 1$. Formally, four conditions need to be fulfilled: First, if an interval of v is already active at time step i , that is, if $\text{pos}(v) \geq 2$, then an activity interval of v is already active during time step $i - 1$. This can be formalized as follows.

$$\text{If } \text{pos}(v) \geq 2, \text{ then } \text{curr}'(v) \leq \text{curr}(v) \text{ and } \text{pos}'(v) = \text{pos}(v) - 1. \quad (1)$$

Second, if a new activity interval of v starts at time step i either v was inactive at time step $i - 1$ or the previous activity interval of v ended at time step $i - 1$. This can be formalized as follows.

$$\text{If } \text{curr}(v) > 1 \text{ and } \text{pos}(v) = 1, \text{ then } \text{curr}'(v) \leq \text{curr}(v) - 1 \text{ and } \text{pos}'(v) \in \{\min(i-1, \ell+1), 0\}. \quad (2)$$

Third, if the first activity interval of v starts at time step i , then in the previous time step $i - 1$ vertex v cannot be active. This can be formalized as follows.

$$\text{If } \text{curr}(v) = 1 \text{ and } \text{pos}(v) = 1, \text{ then } \text{curr}'(v) = 0 \text{ and } \text{pos}'(v) = 0. \quad (3)$$

Finally, if v is not active during time step i , either v is also not active during time step $i - 1$ or an activity interval of v ended at time step $i - 1$. This can be formalized as follows.

$$\text{If } \text{curr}(v) \geq 1 \text{ and } \text{pos}(v) = 0, \text{ then } \text{curr}'(v) \leq \text{curr}(v) \text{ and } \text{pos}'(v) \in \{\min(i-1, \ell+1), 0\}. \quad (4)$$

The minimum in Conditions 2 and 4 ensures that the activity intervals do not have a starting point smaller than one.

Correctness: We show the correctness of the computation by proving inequalities in both directions.

($\leq$) Suppose the maximum in the definition of the table entry $\text{DP}[i, \text{curr}, \text{pos}]$ is attained for $\mathcal{T}$ and assume that no activity intervals from $\mathcal{T}$ overlap. We define $\mathcal{T}'$ by taking all activity intervals from $\mathcal{T}$ which are relevant for a table entry of $i - 1$. Formally, we define

$$\begin{aligned} \mathcal{T}' := & \{(v, a, b) \in \mathcal{T} : a \leq i - 2, v \in \overline{F}_{i-1}\} \\ & \cup \{(v, a, b) \in \mathcal{T} : a \leq i - 1, v \in F_{i-1}\}, \end{aligned}$$

and, in accordance with $\mathcal{T}'$, functions $\text{curr}' : F_{i-1} \rightarrow [0, k]$, $\text{pos}' : F_{i-1} \rightarrow [0, \ell + 1]$ by

$$\begin{aligned} \text{curr}'(v) &:= |\{(v, a, b) \in \mathcal{T}' : a, b \in [T]\}| \\ \text{pos}'(v) &:= \begin{cases} i - a & \text{if } (v, a, b) \in \mathcal{T}' \text{ with } a \leq i \leq b \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Note that $\mathcal{T}' \subseteq \mathcal{T}$. We show that curr' and pos' satisfy Conditions (1)–(4). This means that the table entry $\text{DP}[i - 1, \text{curr}', \text{pos}']$ is considered in the maximum on the right side of the recursive computation. By definition of curr' and pos' it is clear that $\mathcal{T}'$ is then considered in the definition of the table entry $\text{DP}[i - 1, \text{curr}', \text{pos}']$. Now let $v \in F_{i-1} \cap F_i$.

- (1) If $\text{pos}(v) \geq 2$, then $(v, a, b) \in \mathcal{T}$ for some $a \leq i - 1, b \in [T]$ and by definition of $\mathcal{T}'$, each such activity interval of v in $\mathcal{T}$ is also contained in $\mathcal{T}'$. Consequently, we have $\text{curr}'(v) \leq \text{curr}(v)$ and $\text{pos}'(v) = \text{pos}(v) - 1$.

- (2) If $\text{curr}(v) > 1$ and $\text{pos}(v) = 1$, then $(v, i, b) \in \mathcal{T} \setminus \mathcal{T}'$ for some $b \in [T]$. So the activity timeline $\mathcal{T}'$ contains at most $\text{curr}(v) - 1$ activity intervals of v and therefore $\text{curr}'(v) \leq \text{curr}(v) - 1$. Now either $(v, a, i-1) \in \mathcal{T}'$ for some $a \in [i-1]$ or v is not active in G_{i-1} with respect to the activity intervals from $\mathcal{T}'$. This means that either $\text{pos}'(v) = \min(i-1, \ell+1)$ or $\text{pos}'(v) = 0$.
- (3) If $\text{curr}(v) = 1$ and $\text{pos}(v) = 1$, then $(v, i, b) \in \mathcal{T} \setminus \mathcal{T}'$ for some $b \in [T]$. Consequently, $\mathcal{T}'$ contains no activity interval of v and therefore $\text{curr}'(v) = \text{pos}'(v) = 0$.
- (4) If $\text{curr}(v) \geq 1$ and $\text{pos}(v) = 0$, then either $(v, a, i-1) \in \mathcal{T}'$ for some $a \in [i-1]$ or v is not active in G_{i-1} with respect to the activity intervals from $\mathcal{T}'$. So by definition $\text{curr}'(v) \leq \text{curr}(v)$ and either $\text{pos}'(v) = \min(i-1, \ell+1)$ or $\text{pos}'(v) = 0$.

Now consider the set $\mathcal{A}$ of temporal edges until time step i that are covered by $\mathcal{T}$ and the set $\mathcal{A}'$ of temporal edges until time step $i-1$ that are covered by $\mathcal{T}'$. There are two cases for a temporal edge $(vw, j) \in \mathcal{A} \setminus \mathcal{A}'$.

Case 1: $j = i$. Since the temporal edge (vw, j) is covered by $\mathcal{T}$ we can conclude that $vw \in E_{G_i}(\{v \in F_i : \text{pos}(v) \geq 1\}, F_i)$.

Case 2: $j < i$. We have $(vw, j) \in \binom{F_{i-1}}{2} \times [i-1]$. This implies the existence of an activity interval (v, a, b) or (w, a, b) in $\mathcal{T}$ with $a \leq i-1$. However, then the activity interval is by definition contained in $\mathcal{T}'$ and consequently $(vw, j) \in \mathcal{A}'$.

Hence, any temporal edge in $\mathcal{A} \setminus \mathcal{A}'$ is contained in $E_{G_i}(\{v \in F_i : \text{pos}(v) \geq 1\}, F_i)$. We thus have the desired inequality

$$\begin{aligned} \text{DP}[i, \text{curr}, \text{pos}] &= |\mathcal{A}| \leq |\mathcal{A}'| + |E_{G_i}(\{v \in F_i : \text{pos}(v) \geq 1\}, F_i)| \\ &\leq \text{DP}[i-1, \text{curr}', \text{pos}'] + |E_{G_i}(\{v \in F_i : \text{pos}(v) \geq 1\}, F_i)|. \end{aligned}$$

($\geq$) Suppose the maximum on the right side of the computation is attained for $\text{curr}', \text{pos}'$ and the maximum in the definition of $\text{DP}[i-1, \text{curr}', \text{pos}']$ is attained for $\mathcal{T}'$. We define $\mathcal{T}$ by extending $\mathcal{T}'$ in the following way: For $v \in F_{i-1} \cap F_i$ the activity interval $(v, i, \min(i+\ell, T))$ is added if and only if $\text{pos}(v) = 1$. These new intervals together with the already active ones ($\text{pos}(v) \geq 2$) cover all edges of $E_{G_i}(\{v \in F_i : \text{pos}(v) \geq 1\}, F_i)$. Hence, the total number of edges covered by $\mathcal{T}$ is $\text{DP}[i-1, \text{curr}', \text{pos}'] + |E_{G_i}(\{v \in F_i : \text{pos}(v) \geq 1\}, F_i)|$.

Because $\mathcal{T}$ is considered in the definition of $\text{DP}[i, \text{curr}, \text{pos}]$, we have

$$\text{DP}[i, \text{curr}, \text{pos}] \geq \text{DP}[i-1, \text{curr}', \text{pos}'] + |E_{G_i}(\{v \in F_i : \text{pos}(v) \geq 1\}, F_i)|.$$

Hence, the desired inequality holds.

Result: Finally, we return yes if and only if $\text{DP}[T, \text{curr}, \text{pos}] \geq t$ for $\text{curr} : F_T \rightarrow \{k\}$ and some function $\text{pos} : F_T \rightarrow \{0, \ell+1\}$. The restriction to these functions is correct, since we can assume without loss of generality that a vertex is active in G_T if and only if its k -th activity interval ends at time step T (recall that we assume that each vertex is in at least $k(\ell+1)+1$ bags). Consequently, all k -activity ℓ -timelines of $\mathcal{G}$ are considered in the definition of the table entry $\text{DP}[T, \text{curr}, \text{pos}]$ and all covered temporal edges of $\mathcal{G}$ are counted in the respective maximum of the definition.

Running Time: For each time step $i \in [T]$, there are $((k+1)(\ell+2))^{\text{vimw}}$ choices for the functions $\text{curr} : F_i \rightarrow [0, k]$ and $\text{pos} : F_i \rightarrow [0, \ell+1]$. This upper-bounds the size of our dynamic programming table by $T \cdot ((k+1)(\ell+2))^{\text{vimw}}$. In each recursive computation of a table entry, there are also $\mathcal{O}(((k+1)(\ell+2))^{\text{vimw}})$ choices for curr' and pos' . The remaining summands of the recursive formula can be computed in $\mathcal{O}(n)$ time. Hence, the overall running time can be bounded by $((k+1)(\ell+2))^{2 \cdot \text{vimw}} \cdot (n+T)^{\mathcal{O}(1)}$. $\blacktriangleleft$

Now, we show that for TIMELINE PVC the parameter $\text{vimw} = \text{vimw}[1]$ can be replaced by an even smaller parameter, $\text{vimw}[k(\ell + 1) + 1]$. The algorithm for this parameter has two steps: First it applies preprocessing step that handles the large bags. Then, it invokes the algorithm of [Theorem 4.1](#) for $\text{vimw} = \text{vimw}[1]$.

► **Proposition 4.2.** *TIMELINE PVC is solvable in $((k + 1)(\ell + 2))^{4 \cdot \text{vimw}[x]} \cdot (n + T)^{\mathcal{O}(1)}$ time where $x = k(\ell + 1) + 1$.*

Proof. Let $(F_i)_{i \in [T]}$ be the bags of the vertex-interval-membership sequence of $\mathcal{G}$. Similar to the proof of [Theorem 4.1](#), it is safe to assume that each vertex v is contained in at least $k \cdot (\ell + 1)$ bags (by definition the bags containing v need to be consecutive) because otherwise, we can greedily pick the k -activity intervals of v such that v is active during all these snapshots and thus all temporal edges incident to v are covered and we can reduce t accordingly.

We call the $k \cdot (\ell + 1)$ largest bags of $(F_i)_{i \in [T]}$ *large*. The idea is to pick the intervals greedily for all vertices that are only contained in large bags. Intuitively, this works since the large bags appear consecutively. Afterwards, we remove all these vertices and adapt t accordingly. Finally, we can invoke the algorithm of [Theorem 4.1](#).

Now, let $[i, j] \subset [T]$ be a sequence of consecutive indices such that each bag F_x for each $x \in [i, j]$ is large. By definition $j - i \leq k \cdot (\ell + 1) - 1$. Now, let $\mathcal{A} = \bigcup_{x \in [i, j]} F_x \setminus (F_{i-1} \cup F_{j+1})$. If $i = 0$ or $j = T$, then F_{i-1} or F_{j+1} is simply the empty set. Suppose $\mathcal{A} \neq \emptyset$. By definition, each vertex in $\mathcal{A}$ is isolated in $G_1, \dots, G_{i-1}$ and also in $G_{j+1}, \dots, G_T$. By our assumption that each vertex is contained in at least $k \cdot (\ell + 1)$ bags, we can conclude that $j - i = k \cdot (\ell + 1) - 1$. Now observe that by picking the k -activity intervals for each vertex in $\mathcal{A}$ greedily, we can cover all temporal edges which are incident to some vertex of $\mathcal{A}$. Hence, it is safe to remove $\mathcal{A}$ from $\mathcal{G}$ and to reduce t by the number of temporal edges covered by the vertices in $\mathcal{A}$. For the remaining instance, we use the algorithm of [Theorem 4.1](#). After removing $\mathcal{A}$, we have $|F_x| \leq |F_{i-1}| + |F_{j+1}| \leq 2 \text{vimw}[k(\ell + 1) + 1]$ for every $x \in [i, j]$ and consequently [Theorem 4.1](#) yields the desired running time. Now, suppose $\mathcal{A} = \emptyset$ and let F_x be a large bag contained in a sequence of large bags with indices $[i, j]$. In this case, we can conclude $|F_x| \leq |F_{i-1}| + |F_{j+1}| \leq 2 \text{vimw}[k(\ell + 1) + 1]$ and again apply [Theorem 4.1](#). ◀

Now, we focus on TIMELINE (PARTIAL) DOMINATING SET. Initially, we show how to adapt the algorithm of [Theorem 4.1](#) for TIMELINE PDS to obtain an algorithm with the same running time for TIMELINE PDS. However, we need to be more careful since isolated vertices in the snapshots can no longer be ignored as for TIMELINE PVC. This means that vertices which are not contained in the current bag of the vertex-interval-membership sequence might have to be active. Hence, the update of the dynamic program additionally needs to consider the case that activity intervals of vertices appear before or after their first or last occurrence in a bag.

► **Theorem 4.3.** *TIMELINE PDS is solvable in $((k + 1)(\ell + 2))^{2 \cdot \text{vimw}} \cdot (n + T)^{\mathcal{O}(1)}$ time.*

Proof. Let $(\mathcal{G} = (G_1, \dots, G_T), k, \ell, t)$ be an instance of TIMELINE PDS. Again, suppose that $T \geq k \cdot (\ell + 1)$ (otherwise, it is a trivial instance) and that the underlying graph $G_\downarrow$ contains no isolated vertex (otherwise, decrease t by $k \cdot (\ell + 1)$ and delete the vertex).

The idea of the dynamic program over the lifetime of the temporal graph is analogous to the algorithm of [Theorem 4.1](#). Recall that the dynamic program keeps track of the activity of the vertices which are contained in the bag F_i of the currently considered time step $i \in [T]$. Again, while iterating over the lifetime of the temporal graph, we need to ensure that the activities at a time step are compatible with the activities at the neighboring time steps.

Moreover, we need to be careful as it might be optimal for some vertices to be active at time steps where they are not contained in the respective bag to dominate some isolated vertices.

Recall that the vertex-interval-membership sequence can be computed in polynomial time. Since we assumed that the underlying graph contains no isolated vertex, it follows that each vertex is contained in at least one bag of the sequence.

Table Definition: Again we let $\overleftarrow{F}_i := F_1 \cup \dots \cup F_i$ and $\overline{F}_i := \overleftarrow{F}_i \setminus F_i$. Recall that $\overline{F}_i$ is the set of vertices which are isolated in all snapshots $G_i, \dots, G_T$. Moreover, again for $i \in [T]$ we define functions $\text{curr} : F_i \rightarrow [0, k]$, and $\text{pos} : F_i \rightarrow [0, \ell + 1]$ which indicate for each vertex $v \in F_i$ whether an activity interval of v is active during snapshot G_i . Recall that the function value $\text{curr}(v)$ determines the current activity interval of v and the function value $\text{pos}(v)$ determines the position in this activity interval.

Now, the entry $\text{DP}[i, \text{curr}, \text{pos}]$ contains the combined maximum number of dominated temporal vertices of $\overline{F}_i$ in the whole temporal graph and of F_i in the snapshots $G_1, \dots, G_i$, while i is at the $\text{pos}(v)$ -th position of the $\text{curr}(v)$ -th activity interval of $v \in F_i$. We are now ready to present the formal definition of the table.

$$\text{DP}[i, \text{curr}, \text{pos}] \triangleq \max_{\mathcal{T}} |\{(v, j) \in \overline{F}_i \times [T] : (v, j) \text{ is dominated by } \mathcal{T}\} \cup \{(v, j) \in F_i \times [i] : (v, j) \text{ is dominated by } \mathcal{T}\}|,$$

where the maximum is taken over all k -activity ℓ -timelines

$$\mathcal{T} \subseteq (\overline{F}_i \times [T] \times [T]) \cup (F_i \times [i] \times [T]),$$

such that for each $v \in F_i$

- (i) there are at most $\text{curr}(v)$ activity intervals of v in $\mathcal{T}$,
- (ii) $\text{pos}(v) \leq i$, and
- (iii) $(v, i - \text{pos}(v) + 1, b) \in \mathcal{T}$ for some $b \geq 1$ if $\text{pos}(v) > 0$.

We need conditions (i)-(iii) for the identical reasons as for TIMELINE PVC: Condition (i) ensures that each vertex has at most k activity intervals. Moreover, condition (ii) ensures that no activity interval starts before time step one. Finally, condition (iii) ensures that each activity interval has length ℓ , except if the remaining number of snapshots is too small.

Initialization: For all $\text{curr} : F_1 \rightarrow [0, k]$ and $\text{pos} : F_1 \rightarrow [0, 1]$, for which $\text{curr}(v) = 0$ implies $\text{pos}(v) = 0$, the table is initialized by

$$\text{DP}[1, \text{curr}, \text{pos}] = |N_{G_1}[\{v \in F_1 : \text{pos}(v) = 1\}]|.$$

Note that $\overline{F}_1 = \emptyset$ and therefore only activity timelines $\mathcal{T} \subseteq F_1 \times [1] \times [T]$ are considered in the definition of $\text{DP}[1, \text{curr}, \text{pos}]$. The initialization is correct, since the table entry $\text{DP}[1, \text{curr}, \text{pos}]$ contains the number of vertices which are dominated by active vertices from F_1 in G_1 and the dominated vertices are also contained in F_1 by definition.

Recurrence: Intuitively, $\text{DP}[i, \text{curr}, \text{pos}]$ is computed by trying all activity interval choices for the vertices of F_{i-1} , such that there is no conflict with curr and pos . Clearly, all vertices which are either active during time step i or a neighbor of an active vertex in snapshot G_i get dominated (summand (A1)). However, we also need to consider how many of the vertices from F_i and F_{i-1} are isolated in $G_1, \dots, G_{i-1}$ and $G_i, \dots, G_T$ respectively. In particular, we need to count how many of these isolated temporal vertices are dominated by a solution candidate. More precisely, two cases are possible: First, if $v \in F_i$ is isolated in $G_1, \dots, G_{i-1}$, then in each time step in $[i-1]$, which intersects an activity interval of v , exactly one vertex, that is, v itself is dominated (summands (A2) and (A3)). Second, if $v \in F_i$ is isolated

in $G_{i+1}, \dots, G_T$, then in each time step in $[i+1, T]$, which intersects an activity interval of v , exactly one vertex, that is, v itself is dominated (summands (A4) and (A5)). Consequently, the formal update is as follows:

$$\begin{aligned}
 \text{DP}[i, \text{curr}, \text{pos}] = & \max_{\text{curr}', \text{pos}'} \text{DP}[i-1, \text{curr}', \text{pos}'] \\
 & + |N_{G_i}[\{v \in F_i : \text{pos}(v) \geq 1\}]| \quad (A1) \\
 & + \sum_{\substack{v \in F_i \setminus F_{i-1}: \\ \text{pos}(v)=0}} \min(\text{curr}(v)(\ell+1), i-1) \quad (A2) \\
 & + \sum_{\substack{v \in F_i \setminus F_{i-1}: \\ \text{pos}(v)>0}} \min((\text{curr}(v)-1)(\ell+1) + \text{pos}(v)-1, i-1) \quad (A3) \\
 & + \sum_{\substack{v \in F_{i-1} \setminus F_i: \\ \text{pos}'(v)=0}} \min((k - \text{curr}'(v))(\ell+1), T-i+1) \quad (A4) \\
 & + \sum_{\substack{v \in F_{i-1} \setminus F_i: \\ \text{pos}'(v)>0}} \min((k - \text{curr}'(v))(\ell+1) + \ell+1 - \text{pos}'(v), T-i+1) \quad (A5),
 \end{aligned}$$

where the maximum is taken over all $\text{curr}' : F_{i-1} \rightarrow [0, k]$ and $\text{pos}' : F_{i-1} \rightarrow [0, \ell+1]$ which are *compatible* with curr and pos . Here, compatibility has the identical meaning as in the algorithm in [Theorem 4.1](#) for TIMELINE PVC: Informally, curr' and pos' are compatible with curr and pos if they provide a correct extension of the activities of $F_{i-1} \cap F_i$ at time step i to time step $i-1$. Formally, four conditions need to be fulfilled: First, if an interval of v is already active at time step i , that is, if $\text{pos}(v) \geq 2$, then an activity interval of v is already active during time step $i-1$. This can be formalized as follows.

$$\text{If } \text{pos}(v) \geq 2, \text{ then } \text{curr}'(v) \leq \text{curr}(v) \text{ and } \text{pos}'(v) = \text{pos}(v) - 1. \quad (1)$$

Second, if a new activity interval of v starts at time step i either v was inactive at time step $i-1$ or the previous activity interval of v ended at time step $i-1$. This can be formalized as follows.

$$\text{If } \text{curr}(v) > 1 \text{ and } \text{pos}(v) = 1, \text{ then } \text{curr}'(v) \leq \text{curr}(v) - 1 \text{ and } \text{pos}'(v) \in \{\min(i-1, \ell+1), 0\}. \quad (2)$$

Third, if the first activity interval of v starts at time step i , then in the previous time step $i-1$ vertex v cannot be active. This can be formalized as follows.

$$\text{If } \text{curr}(v) = 1 \text{ and } \text{pos}(v) = 1, \text{ then } \text{curr}'(v) = 0 \text{ and } \text{pos}'(v) = 0. \quad (3)$$

Finally, if v is not active during time step i , either v is also not active during time step $i-1$ or an activity interval of v ended at time step $i-1$. This can be formalized as follows.

$$\text{If } \text{curr}(v) \geq 1 \text{ and } \text{pos}(v) = 0, \text{ then } \text{curr}'(v) \leq \text{curr}(v) \text{ and } \text{pos}'(v) \in \{\min(i-1, \ell+1), 0\}. \quad (4)$$

The minimum in two of the conditions ensures that the activity intervals do not have a starting point smaller than one.

Correctness: ($\leq$) Suppose the maximum in the definition of the table entry $\text{DP}[i, \text{curr}, \text{pos}]$ is attained for $\mathcal{T}$ and assume that no activity intervals from $\mathcal{T}$ overlap. We define $\mathcal{T}'$ by taking all activity intervals from $\mathcal{T}$, which are relevant for a table entry of $i - 1$. Formally, we define

$$\begin{aligned} \mathcal{T}' := & \{(v, a, b) \in \mathcal{T} : v \in \overline{F}_{i-1}\} \\ & \cup \{(v, a, b) \in \mathcal{T} : a \leq i - 1, v \in F_{i-1}\}, \end{aligned}$$

and, in accordance with $\mathcal{T}'$, functions $\text{curr}' : F_{i-1} \rightarrow [0, k], \text{pos}' : F_{i-1} \rightarrow [0, \ell + 1]$ by

$$\begin{aligned} \text{curr}'(v) &:= |\{(v, a, b) \in \mathcal{T}' : a, b \in [T]\}| \\ \text{pos}'(v) &:= \begin{cases} i - a & \text{if } (v, a, b) \in \mathcal{T}' \text{ with } a \leq i \leq b \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Note that $\mathcal{T}' \subseteq \mathcal{T}$. We show that curr' and pos' satisfy conditions (1)-(4). This means that the table entry $\text{DP}[i - 1, \text{curr}', \text{pos}']$ is considered in the maximum on the right side of the recursive computation. By definition of curr' and pos' it is clear that $\mathcal{T}'$ is then considered in the definition of the table entry $\text{DP}[i - 1, \text{curr}', \text{pos}']$. Now let $v \in F_{i-1} \cap F_i$.

- (1) If $\text{pos}(v) \geq 2$, then $(v, a, b) \in \mathcal{T}$ for some $a \leq i - 1, b \in [T]$ and by definition of $\mathcal{T}'$, each such activity interval of v in $\mathcal{T}$ is also contained in $\mathcal{T}'$. Consequently, we have $\text{curr}'(v) \leq \text{curr}(v)$ and $\text{pos}'(v) = \text{pos}(v) - 1$.
- (2) If $\text{curr}(v) > 1$ and $\text{pos}(v) = 1$, then $(v, i, b) \in \mathcal{T} \setminus \mathcal{T}'$ for some $b \in [T]$. So the activity timeline $\mathcal{T}'$ contains at most $\text{curr}(v) - 1$ activity intervals of v and therefore $\text{curr}'(v) \leq \text{curr}(v) - 1$. Now either $(v, a, i - 1) \in \mathcal{T}'$ for some $a \in [i - 1]$ or v is not active in G_{i-1} with respect to the activity intervals from $\mathcal{T}'$. This means that either $\text{pos}'(v) = \min(i - 1, \ell + 1)$ or $\text{pos}'(v) = 0$.
- (3) If $\text{curr}(v) = 1$ and $\text{pos}(v) = 1$, then $(v, i, b) \in \mathcal{T} \setminus \mathcal{T}'$ for some $b \in [T]$. Consequently, $\mathcal{T}'$ contains no activity interval of v and therefore $\text{curr}'(v) = \text{pos}'(v) = 0$.
- (4) If $\text{curr}(v) \geq 1$ and $\text{pos}(v) = 0$, then either $(v, a, i - 1) \in \mathcal{T}'$ for some $a \in [i - 1]$ or v is not active in G_{i-1} with respect to the activity intervals from $\mathcal{T}'$. So by definition $\text{curr}'(v) \leq \text{curr}(v)$ and either $\text{pos}'(v) = \min(i - 1, \ell + 1)$ or $\text{pos}'(v) = 0$.

It remains to compare the set

$$\begin{aligned} \mathcal{A} := & \{(v, j) \in \overline{F}_i \times [T] : (v, j) \text{ is dominated by } \mathcal{T}\} \\ & \cup \{(v, j) \in F_i \times [i] : (v, j) \text{ is dominated by } \mathcal{T}\} \end{aligned}$$

with the set

$$\begin{aligned} \mathcal{A}' := & \{(v, j) \in \overline{F}_{i-1} \times [T] : (v, j) \text{ is dominated by } \mathcal{T}'\} \\ & \cup \{(v, j) \in F_{i-1} \times [i - 1] : (v, j) \text{ is dominated by } \mathcal{T}'\}. \end{aligned}$$

Since $\mathcal{T}' \subseteq \mathcal{T}$, $\overline{F}_{i-1} \subseteq \overline{F}_i$ and $F_{i-1} \subseteq F_i \cup \overline{F}_i$, it is clear that $\mathcal{A}' \subseteq \mathcal{A}$. We distinguish between three possible cases, when temporal vertices could be contained in $\mathcal{A} \setminus \mathcal{A}'$: **a)** $(u, j) \in F_i \times \{i\}$, **b)** $(u, j) \in F_i \times [i - 1]$, and **c)** $(u, j) \in \overline{F}_i \times [T]$. These cases are analog to the proof of [Theorem 4.1](#) for TIMELINE PVC; but now they are more involved since also isolated vertices are important for domination.

- a) If $(u, j) \in F_i \times \{i\}$ is contained in $\mathcal{A}$, then $u \in N_{G_i}[\{v \in F_i : \text{pos}(v) \geq 1\}]$ and therefore either $u \in F_{i-1} \cap F_i$ or $u \in F_i \setminus F_{i-1}$. If $u \in F_{i-1} \cap F_i$, then we have $(u, i) \notin \mathcal{A}'$ because

$\mathcal{A}'$ only contains temporal vertices of F_{i-1} from the first $i-1$ snapshots. Otherwise, if $u \in F_i \setminus F_{i-1}$, then $u \notin \overleftarrow{F}_{i-1}$ and subsequently $(u, i) \notin \mathcal{A}'$. In both cases, (u, i) is counted in summand (A1).

- b) If $u \in F_i \cap \overleftarrow{F}_{i-1}$, then u is dominated by some activity interval (w, a, b) with $a \leq i-1$ and $w \in \overleftarrow{F}_{i-1}$. Therefore $\mathcal{T}'$ dominates u and therefore $u \in \mathcal{A}'$. If $u \in F_i \setminus \overleftarrow{F}_{i-1}$, then $\mathcal{A} \setminus \mathcal{A}'$ possibly contains temporal vertices (u, j) for some $j \in [i-1]$. Since u is isolated in these time steps, we conclude that $(u, j) \in \mathcal{A} \setminus \mathcal{A}'$ for $j \in [i-1]$ if and only if $(u, a, b) \in \mathcal{T}$ with $a \leq j \leq b$. If $\text{pos}(u) = 0$, then there are at most $\text{curr}(u)$ activity intervals of u from $\mathcal{T}$ in $G_1, \dots, G_{i-1}$ and therefore at most $\min(\text{curr}(u)(\ell+1), i-1)$ isolated temporal vertices of u from $G_1, \dots, G_{i-1}$ are contained in $\mathcal{A} \setminus \mathcal{A}'$ (summand (A2)). If $\text{pos}(u) > 0$, then there are at most $\text{curr}(u) - 1$ activity intervals and a part of the $\text{curr}(u)$ -th activity interval of u from $\mathcal{T}$ in $G_1, \dots, G_{i-1}$. So in this case there are at most $\min((\text{curr}(u) - 1)(\ell+1) + \text{pos}(u) - 1, i-1)$ isolated temporal vertices of u from $G_1, \dots, G_{i-1}$ contained in $\mathcal{A} \setminus \mathcal{A}'$ (summand (A3)).
- c) If $u \in \overleftarrow{F}_i \cap \overleftarrow{F}_{i-1}$, then u is dominated by some activity interval of itself or by some (w, a, b) with $a \leq i-1$ and $w \in \overleftarrow{F}_{i-1}$. Again, by definition $\mathcal{T}'$ dominates u and therefore $u \in \mathcal{A}'$. If $u \in \overleftarrow{F}_i \setminus \overleftarrow{F}_{i-1} = F_{i-1} \setminus F_i$, then $\mathcal{A} \setminus \mathcal{A}'$ possibly contains temporal vertices (u, j) for $j \in [i, T]$. Since u is isolated in these time steps, we have $(u, j) \in \mathcal{A} \setminus \mathcal{A}'$ for $j \in [i, T]$ if and only if $(u, a, b) \in \mathcal{T}$ with $a \leq j \leq b$. If $\text{pos}(u) = 0$, then there are at most $(k - \text{curr}'(u))$ activity intervals of u from $\mathcal{T}$ in $G_i, \dots, G_T$ and therefore at most $\min((k - \text{curr}'(u))(\ell+1), T-i+1)$ isolated temporal vertices of u in $G_i, \dots, G_T$ are contained in $\mathcal{A} \setminus \mathcal{A}'$ (summand (A4)). If $\text{pos}(u) > 0$, then there are at most $(k - \text{curr}'(u))$ activity intervals and a part of the $(\text{curr}(u) - 1)$ -th activity interval of u from $\mathcal{T}$ in $G_i, \dots, G_T$. Consequently, in this case there are at most $\min((k - \text{curr}'(u))(\ell+1) + \ell + 1 - \text{pos}(u), T-i+1)$ isolated temporal vertices of u in $G_i, \dots, G_T$ contained in $\mathcal{A} \setminus \mathcal{A}'$ (summand (A5)).

Combining the previous observations, we have

$$\text{DP}[i, \text{curr}, \text{pos}] = |\mathcal{A}| \leq |\mathcal{A}'| + (A1) + (A2) + (A3) + (A4) + (A5)$$

and therefore the desired inequality holds.

($\geq$) Suppose the maximum on the right side of the computation is attained for $\text{curr}', \text{pos}'$ and the maximum in the definition of $\text{DP}[i-1, \text{curr}', \text{pos}']$ is attained for $\mathcal{T}'$. We define $\mathcal{T}$ by extending $\mathcal{T}'$ in the following way.

- For $v \in F_{i-1} \cap F_i$ the activity interval $(v, i, \min(i+\ell, T))$ is added if and only if $\text{pos}(v) = 1$. With these intervals $\mathcal{T}$ dominates all of $N_{G_i}[\{v \in F_i : \text{pos}(v) \geq 1\}]$.
- For $v \in F_i \setminus F_{i-1}$ (so v is isolated in the snapshots $G_1, \dots, G_{i-1}$) there are two cases. If $\text{pos}(v) = 0$, then $\text{curr}(v)$ activity intervals of v are added in the time interval $[i-1]$ such that $\min(\text{curr}(v)(\ell+1), i-1)$ isolated temporal vertices of v in $G_1, \dots, G_{i-1}$ are dominated by $\mathcal{T}$. Otherwise, if $\text{pos}(v) > 0$, then $(\text{curr}(v) - 1)$ activity intervals of v are added in the time interval $[i-1]$ such that $\min((\text{curr}(v) - 1)(\ell+1) + \text{pos}(v) - 1, i-1)$ isolated temporal vertices of v in $G_1, \dots, G_{i-1}$ are dominated by $\mathcal{T}$.
- For $v \in \overleftarrow{F}_i \setminus \overleftarrow{F}_{i-1} = F_{i-1} \setminus F_i$ (so v is isolated in $G_i, \dots, G_T$) there are also two cases. If $\text{pos}'(v) = 0$, then $(k - \text{curr}'(v))$ activity intervals of v are added in the time interval $[i, T]$ such that $\min((k - \text{curr}'(v))(\ell+1), T-i+1)$ isolated temporal vertices of v in $G_i, \dots, G_T$ are dominated by $\mathcal{T}$. If $\text{pos}'(v) > 0$, then $(k - \text{curr}'(v))$ activity intervals of v are added in the time interval $[i, T]$ so that $\min((k - \text{curr}'(v))(\ell+1) + \ell + 1 - \text{pos}(v), T-i+1)$ isolated temporal vertices of v in $G_i, \dots, G_T$ are dominated by $\mathcal{T}$.

By definition of the table entries we have

$$\text{DP}[i, \text{curr}, \text{pos}] \geq \text{DP}[i-1, \text{curr}', \text{pos}'] + (A1) + (A2) + (A3) + (A4) + (A5),$$

because $\mathcal{T}$ is considered in the definition of $\text{DP}[i, \text{curr}, \text{pos}]$. Therefore the desired inequality holds.

Result: Finally, we return yes if and only if $\text{DP}[T, \text{curr}, \text{pos}] \geq t$ for $\text{curr} : F_T \rightarrow \{k\}$ and some function $\text{pos} : F_T \rightarrow \{0, \ell+1\}$. The restriction to these functions is correct, since we can assume without loss of generality that a vertex is active in G_T if and only if its k -th activity interval ends at time step T . Further, note that $\bar{F}_T \cup F_T = V(\mathcal{G})$. Consequently, all k -activity ℓ -timelines of $\mathcal{G}$ are considered in the definition of the table entry $\text{DP}[T, \text{curr}, \text{pos}]$ and all dominated temporal vertices of $\mathcal{G}$ are counted in the respective maximum of the definition.

Running Time: For each time step $i \in [T]$, there are $((k+1)(\ell+2))^{\text{vimw}}$ choices for the functions $\text{curr} : F_i \rightarrow [0, k]$ and $\text{pos} : F_i \rightarrow [0, \ell+1]$. This upper-bounds the size of our dynamic programming table by $T \cdot ((k+1)(\ell+2))^{\text{vimw}}$. In each recursive computation of a table entry, there are also $\mathcal{O}(((k+1)(\ell+2))^{\text{vimw}})$ choices for curr' and pos' . The remaining summands of the recursive formula can be computed in $\mathcal{O}(n)$ time. Hence, the overall running time is bounded by $((k+1)(\ell+2))^{2 \cdot \text{vimw}} \cdot (n+T)^{\mathcal{O}(1)}$. $\blacktriangleleft$

Recall that for TIMELINE PVC we showed that $\text{vimw} = \text{vimw}[1]$ can be replaced by the much smaller parameter $\text{vimw}[k \cdot (\ell+1) + 1]$, the size of the $k \cdot (\ell+1) + 1$ largest bag. Hence, one may ask whether vimw can be replaced by $\text{vimw}[i]$ with some suitably chosen i for TIMELINE DS and TIMELINE PDS. We show that both problems behave quite differently: First, we show we that vimw can be replaced by $\text{vimw}[\ell+2]$ for TIMELINE DS by having a preprocessing step and then adapting the algorithm behind [Theorem 4.3](#). Second, we show that replacing $\text{vimw} = \text{vimw}[1]$ by $\text{vimw}[2]$ is not possible for TIMELINE PDS. More precisely, we show NP-hardness even if there are only two snapshots one of which is edgeless (which implies $\text{vimw}[2] = 0$).

We start with our result for TIMELINE DS.

For TIMELINE PDS, we show that we cannot replace $\text{vimw} = \text{vimw}[1]$ by $\text{vimw}[2]$, since we prove para-NP-hardness for $\text{vimw}[2] + k + \ell$. We start with our algorithmic result for TIMELINE DS.

► **Proposition 4.4.** *TIMELINE DS is solvable in $((k+1)(\ell+2))^{4 \cdot \text{vimw}[x]} \cdot (n+T)^{\mathcal{O}(1)}$ time where $x = \ell+2$.*

Proof. Similarly to the proofs before, it is safe to assume that $T \geq k \cdot (\ell+1) + 1$ because otherwise each vertex can be active in each snapshot and thus every vertex is dominated. Let $(F_i)_{i \in [T]}$ be the bags of the vertex-interval-membership sequence of $\mathcal{G}$. Analogously to [Proposition 4.2](#), we denote the $\ell+1$ largest bags of $(F_i)_{i \in [T]}$ as *large* and each remaining bag is denoted as *small*. The observation is that for each vertex v which is only contained in large bags, there are a huge number of snapshots where v is isolated. To dominate v in such snapshots, v has to be active during them. This essentially fixes all k -activity intervals of v . Afterwards, we use a slight modification of the dynamic program of [Theorem 4.3](#) to find the best activity intervals for the remaining vertices.

Next, let $[i, j] \subseteq [T]$ be a maximal consecutive sequence of indices such that each bag F_x is large for any $x \in [i, j]$. Furthermore, let $\mathcal{A}_{ij} = \bigcup_{x \in [i, j]} F_x \setminus (F_{i-1} \cup F_{j+1})$. The set $\mathcal{A}_{ij}$ contains the vertices which are isolated in $G_1, \dots, G_{i-1}$ and in $G_{j+1}, \dots, G_T$. Note that if $i = 1$ or $j = T$ the respective sets F_{i-1} and F_{j+1} are empty. Moreover, if $\mathcal{A}_{ij}$ is empty, then

we can conclude $|F_x| \leq |F_{i-1}| + |F_{j+1}| \leq 2 \cdot \text{vimw}[\ell + 2]$ for all $x \in [i, j]$ and subsequently **Theorem 4.3** yields the desired running time. Hence, we assume that at least one such $\mathcal{A}_{ij}$ is non-empty.

Observe that since $v \in \mathcal{A}_{ij}$ is isolated in $G_1, \dots, G_{i-1}$ and in $G_{j+1}, \dots, G_T$, the vertex v must be active during $[i - 1]$ and during $[j + 1, T]$ to ensure that it is dominated at these time steps. Since by definition $j - i \leq \ell$, in total these are at least $T - (\ell + 1) \geq k \cdot (\ell + 1) + 1 - (\ell + 1) = (k - 1) \cdot (\ell + 1) + 1$ snapshots. Since $(k - 1)$ activity intervals can dominate at most $(k - 1)(\ell + 1)$ of these isolated vertices, we conclude that all k -activity intervals of v are necessary to dominate v in these snapshots. In particular, if $T \geq k \cdot (\ell + 1) + (\ell + 1) + 1$, then we deal with a no-instance since in at least one snapshot where v is isolated, no activity interval of v can be active.

Now, we can assign the k -activity intervals of v greedily: To dominate v in $G_1, \dots, G_{i-1}$ we let the first activity interval of v start at snapshot 1, the second at snapshot $\ell + 2$ and so on. By the observation above, we know that all k activity intervals of v are used in this way. (If there is still some snapshot where v is isolated and not dominated, then return no.)

Recall that for TIMELINE PVC in the proof of **Proposition 4.2** we could simply remove the vertices in $\mathcal{A}_{ij}$ and then call the dynamic program of **Theorem 4.1** for the smaller bag size. But for TIMELINE DS, however, we cannot simply remove the vertices in $\mathcal{A}_{ij}$, since each vertex in $\mathcal{A}_{ij}$ also needs to be dominated in snapshots $G_i, \dots, G_j$ and may also dominate some other vertices in $G_i, \dots, G_j$ if it is active. **1)** Note that in some of these snapshots the vertices of $\mathcal{A}_{ij}$ are active due to our greedy choice of the activity intervals for the vertices in $\mathcal{A}_{ij}$; but since $T \geq k \cdot (\ell + 1) + 1$, in at least one snapshot the vertices still need to be dominated. **2)** Moreover, in each snapshot G_x where $x \in [i, j]$ and the vertices in $\mathcal{A}_{ij}$ are active, all temporal vertices in $N_{G_x}[\mathcal{A}_{ij}]$ are already dominated. Also note that in snapshot G_x for any $x \in [i, j]$ either all vertices of $\mathcal{A}_{ij}$ are active or none of them.

Due to Properties **1)** and **2)** we adapt the dynamic program of the algorithm of **Theorem 4.3** as follows: For each consecutive sequence $F_i, \dots, F_j$ of large bags we let $\mathcal{A}_{ij} = \bigcup_{x \in [i, j]} F_x \setminus (F_{i-1} \cup F_{j+1})$ be the vertices which are isolated in $G_1, \dots, G_{i-1}$ and $G_{j+1}, \dots, G_T$. Also, let $\mathcal{A}$ be the union of all sets $\mathcal{A}_{ij}$, that is, the set of all vertices which are only non-isolated in some large bags. In the new dynamic program we only consider the subset $Q_i = F_i \setminus \mathcal{A}$ of vertices. Consequently, for a small bag there is no change. But for a large bag F_x we now only consider vertices $Q_x = F_x \setminus \mathcal{A}$. Observe that each vertex in Q_x is either contained in F_{i-1} or in F_{j+1} . Thus, $|Q_x| \leq |F_{i-1}| + |F_{j+1}| \leq 2 \cdot \text{vimw}[\ell + 2]$. We next describe how we change the update for bag F_x to respect **1)** and **2)**. Recall that in G_x all vertices of $\mathcal{A} \setminus \mathcal{A}_{ij}$ are isolated and thus active and that the vertices in $\mathcal{A}_{ij}$ are either all active or not. **1)** if no vertex in $\mathcal{A}_{ij}$ is active, the active vertices in $F_x \setminus \mathcal{A}_{ij}$ need to dominate $\mathcal{A}_{ij}$. Let $S_x = \{v \in F_x : \text{pos}(v) \geq 1\}$. If $\mathcal{A}_{ij} \subseteq N_{G_x}[S]$, then nothing changes, but otherwise if $\mathcal{A}_{ij} \not\subseteq N_{G_x}[S]$, then we set the corresponding entry $\text{DP}[i, \text{curr}, \text{pos}]$ to $-\infty$ since at least one vertex in $\mathcal{A}_{ij}$ does not get dominated in G_x . **2)** if all vertices in $\mathcal{A}_{ij}$ are active, then all vertices of $N_{G_x}[\mathcal{A}_{ij}]$ are already dominated, and thus we change summand (A1) to $|N_{G_x}[\mathcal{A}_{ij} \cup \{v \in F_x : \text{pos}(v) \geq 1\}]|$.

The correctness and the running time of the updated dynamic program can be shown analogously as in the proof of **Theorem 4.3**. Only the exponent of the new running time is different since in each big bag we consider at most $2 \cdot \text{vimw}[\ell + 2]$ vertices. ◀

Finally, we show hardness for constant k, ℓ , and $\text{vimw}[2]$ for TIMELINE PDS.

► **Theorem 4.5.** *TIMELINE PDS is NP-hard even if $T = 2$, $k = 1$, $\ell = 0$, the underlying graph is bipartite, planar, has maximum degree 3, and one snapshot is edgeless.*

Proof. We reduce from the classic DOMINATING SET problem, where the input consists of a graph H and an integer k and the goal is to find a dominating set of size most k . Here, a vertex set S is a dominating set if $N[S] = V(H)$. DOMINATING SET is NP-hard even if the graph is bipartite, planar and has a maximum degree of 3 [20].

Now, we construct a TIMELINE PDS instance $(\mathcal{G}, k = 1, \ell = 0, t)$ as follows: We set $T = 2$. The first snapshot G_1 of $\mathcal{G}$ is a copy of H and the second snapshot G_2 is edgeless. Clearly, the underlying graph of $\mathcal{G}$ has maximum degree 3. Finally, we set $t = 2n - k$ where $n = |V(H)|$.

If S is a dominating set, then letting S be active in G_1 , and letting the vertices $V(H) \setminus S$ be active in G_2 dominates $t = 2n - k$ vertices of $\mathcal{G}$. Conversely, suppose we have a solution for $(\mathcal{G}, k = 1, \ell = 0, t = 2n - k)$. Let A be the vertices which are active in G_1 and let B be the vertices which are active in G_2 . Clearly, $|A| + |B| \leq n$. Let $\alpha = |N[A]|$ and $\beta = |B|$. By definition $\alpha + \beta \geq t = 2n - k$. Since $\alpha \leq n$ and $\beta \leq n$, we obtain $\alpha \geq n - k$, $\beta \geq n - k$ and thus $|B| \geq n - k$ and $|A| \leq k$. Hence, if $\alpha = n$, A is a dominating set of size at most k and we are done. Consequently, in the following it is safe to assume that $\alpha < n$. We let $Z = V(H) \setminus N[A]$ be the vertices of G_1 which are not dominated by A . Moreover, we let $\gamma = \beta - (n - k)$. Note that $0 \leq \gamma \leq k$. Now, observe that if $|Z| > \gamma$, A and B dominate only $n - |Z| + \gamma + n - k \leq 2n - k - 1$ vertices, a contradiction. Thus, $|Z| \leq \gamma$. Now, we claim that $S = A \cup Z$ is a dominating set of H of size at most k . By definition, S is a dominating set. For the size bound, observe that $k = \gamma + n - \beta$ and since $\gamma \geq 0$ we obtain that S has size at most k . $\blacktriangleleft$

4.2 The Parameter Interval-Membership-Width

Now, we study the influence of imw instead of vimw. Initially, we show that the parameter $\text{imw} + k + \ell$ yields a polynomial kernel for TIMELINE DS. This also implies a polynomial kernel for $\text{vimw} + k + \ell$ and $n + k + \ell$ for TIMELINE DS. More precisely, we show an even stronger result: We provide a polynomial kernel for $q + k + \ell$, where q is the maximal number of edges in any snapshot. Note that q is never larger than imw, since any bag F_i contains E_i . Moreover, in the temporal graph $\mathcal{G}$ with a star on n leaves as underlying graph and lifetime $T = 2n$, where the edge towards the i -th leaf is active at snapshots i and $i + n$, we have $\text{imw} = n$ and $q = 1$.

► **Theorem 4.6.** *TIMELINE DS has a kernel where both the number of snapshots and vertices are cubic in $q + k + \ell$ which can be computed in linear time with respect to the input size.*

Proof. In order to provide the kernel we bound the number of snapshots T and the number n of vertices of the underlying graph based on the following case distinction.

Case 1: $q \geq n/2$. By definition each k -activity ℓ -timeline has active vertices in at most $n \cdot k \cdot (\ell + 1)$ snapshots. Hence, if $T > 2 \cdot q \cdot k \cdot (\ell + 1) \geq n \cdot k \cdot (\ell + 1)$, then the instance is a no-instance.

Case 2.1: $q < n/2$ and $T > 2 \cdot k \cdot (\ell + 1) + 1$. We show that in this case, we are dealing with a no-instance. Since each snapshot contains at most q edges, any set of p vertices can dominate at most $p + q$ vertices in this snapshot. As $q < n/2$, it follows that at least $n/2$ vertices must be active in order to dominate all temporal vertices of the corresponding snapshot G_x . Moreover, note that the total number of active vertices in any k -activity ℓ -timeline is at most $z = n \cdot k \cdot (\ell + 1)$. Consequently, $T \leq z/(n/2) \leq 2 \cdot k \cdot (\ell + 1)$ for any yes-instance.

Case 2.2: $q < n/2$ and $T \leq 2 \cdot k \cdot (\ell + 1) + 1$. If $n \leq 4 \cdot q \cdot k \cdot (\ell + 1)$, then we have the desired bound. Otherwise, if $n \geq 4 \cdot q \cdot k \cdot (\ell + 1) + 1$, we can solve the instance in polynomial time as follows. Since each snapshot has at most q edges, at most $2q$ vertices are non-isolated in each snapshot. Moreover, since there are at most $2 \cdot k \cdot (\ell + 1)$ snapshots, in

total at most $4 \cdot q \cdot k \cdot (\ell + 1)$ vertices are non-isolated in the temporal graph. Consequently, since $n \geq 4 \cdot q \cdot k \cdot (\ell + 1) + 1$ there exists at least one vertex v which is isolated in every snapshot. Thus, if $T \geq k \cdot (\ell + 1) + 1$ we deal with a trivial no-instance and if $T \leq k \cdot (\ell + 1)$ we deal with a trivial yes-instance.

Thus, in each case we either solve the instance (giving a kernel of constant size) or obtain a kernel with at most $2 \cdot q \cdot k \cdot (\ell + 1)$ snapshots and at most $4 \cdot q \cdot k \cdot (\ell + 1)$ vertices. Moreover, the kernel can be computed in linear time as it involves only computing n , q , and T . ◀

We show that the remaining problems are NP-hard if $\text{imw} + k + \ell \in \mathcal{O}(1)$.

► **Theorem 4.7.** *TIMELINE VC is NP-hard even if $k = 2$, $\ell = 0$, $\text{imw} = 4$, each edge of the underlying graph exists in exactly one snapshot, and the underlying graph has a constant maximum degree.*

Proof. We reduce from the NP-hard 3-COLORING on graphs with maximum degree four [21].

Construction: Let $(G = (V, E))$ be an instance of 3-COLORING on graphs with maximum degree four. We construct the following temporal graph $\mathcal{G}$. For each vertex $v \in V$, we fix an arbitrary ordering of the edges incident with v and add 16 vertices. Formally, we set

$$V(\mathcal{G}) := \bigcup_{v \in V} \{r_v^i, y_v^i, b_v^i, x_v^i : i \in [4]\}.$$

The vertex sets $R_v := \{r_v^i : i \in [4]\}$, $Y_v := \{y_v^i : i \in [4]\}$, and $B_v := \{b_v^i : i \in [4]\}$ will indicate the color choices of the vertex. The temporal graph $\mathcal{G}$ consists of $12|V| + 3|E|$ snapshots. In the following we formally define the snapshots. Note that the order of the snapshots does not matter for the instance, since $\ell = 0$ and each edge will appear only in a single snapshot.

For each vertex $v \in V$, each color $c \in \{r, y, b\}$, and each $i \in [4]$, we add a snapshot $G_{v,c,i}$ with edge set $\{x_v^i c_v^j : j \in [4]\}$. For each edge $e = \{u, v\} \in E$ and each color $c \in \{r, y, b\}$, we add a snapshot $G_{e,c}$ that has only the single edge $c_u^i c_v^j$, such that e is the i th edge incident with u in G and the j th edge incident with v in G . This completes the construction of $\mathcal{G}$. Note that each edge of the underlying graph exists in exactly one snapshot: Edges incident with a vertex x_v^i only exist in the three snapshots $G_{v,r,i}$, $G_{v,y,i}$, and $G_{v,b,i}$, and all these three snapshots are pairwise edge disjoint. All other edges are part of only a single snapshot by construction of the snapshots $G_{e,c}$. Moreover, note that each vertex q of $R_v \cup Y_v \cup B_v$ has at most one snapshot outside of $\{G_{v,c,i} : c \in \{r, y, b\}, i \in [4]\}$ in which it has an active incident edge. If such a snapshot exists, we may call it the *relevant* snapshot of q . Finally, we set $k = 2$ and $\ell = 0$.

Intuition: Let $v \in V$. For each $i \in [4]$, to cover all edges of the snapshots $\{G_{v,c,i} : c \in \{r, y, b\}\}$ the temporal vertex x_v^i can be used only twice, which implies that for some $C \in \{R, Y, B\}$, all vertices of C_v are part of the timeline $\mathcal{T}$ which covers all temporal edges of $\mathcal{G}$ for at least one of the snapshots $\{G_{v,c,i} : c \in \{r, y, b\}\}$. Since i has 4 possible values and each vertex can be part of $\mathcal{T}$ for at most two snapshots, there is some $C \in \{R, Y, B\}$, such that all vertices of C_v in $\mathcal{T}$ are from the snapshots $\{G_{v,c,i} : c \in \{r, y, b\}, i \in [4]\}$. As we will show, this resembles a color choice for which we can show that it implies a proper 3-coloring of G . Intuitively, this is the case since for the selected color c , for each edge $e = vw$ of G , the unique temporal edge of $G_{e,c}$ must be covered, which can only be achieved by a vertex of C_w . Thus, the color of w cannot be c simultaneously.

Correctness: We now show that G is 3-colorable if and only if there is a k -activity ℓ -timeline covering all temporal edges of $\mathcal{G}$.

($\Rightarrow$) Let $\pi : V \rightarrow \{r, y, b\}$ be a proper 3-coloring of G . We define a k -activity ℓ -timeline $\mathcal{T}$ covering all temporal edges of $\mathcal{G}$ as follows. We initialize $\mathcal{T} = \emptyset$. For a fixed vertex $v \in V$

we let $c := \pi(v)$, and we let $\{\alpha, \beta\} = \{r, y, b\} \setminus \{c\}$. We add the following temporal vertices to $\mathcal{T}$:

1. the vertices of $\{c_v^i : i \in [4]\}$ from both snapshots $G_{v,c,1}$ and $G_{v,c,2}$,
2. the vertex x_v^1 from both snapshots $G_{v,\alpha,1}$ and $G_{v,\beta,1}$,
3. the vertex x_v^2 from both snapshots $G_{v,\alpha,2}$ and $G_{v,\beta,2}$,
4. the vertex x_v^3 from both snapshots $G_{v,c,3}$ and $G_{v,\alpha,3}$,
5. the vertex x_v^4 from both snapshots $G_{v,c,4}$ and $G_{v,\beta,4}$,
6. the vertices of $\{\alpha_v^i : i \in [4]\}$ from snapshot $G_{v,\alpha,4}$,
7. the vertices of $\{\beta_v^i : i \in [4]\}$ from snapshot $G_{v,\beta,3}$, and
8. each vertex q of $\{\alpha_v^i, \beta_v^i : i \in [4]\}$ from its relevant snapshots (if such a snapshot exists).

Note that each vertex of the underlying graph is part of at most two snapshots in $\mathcal{T}$. We now show that in each snapshot, for each temporal edge, at least one endpoint is in $\mathcal{T}$.

To this end, observe that for each vertex v , each temporal edge of any of the snapshots of $\{G_{v,c,i} : c \in \{r, y, b\}, i \in [4]\}$ is covered by exactly one temporal vertex described in **Items 1–7**. Thus, consider a temporal edge (e', t) that is not incident with any vertex x_w^i . By definition of all snapshots outside of $\{G_{v,c,i} : c \in \{r, y, b\}, i \in [4]\}$, $e' = c_u^i c_v^j$ for some edge $e = uv \in E$ and a color $c \in \{r, y, b\}$, such that e is the i th edge incident with u in G and the j th edge incident with v in G . In particular, $G_{e,c}$ is the unique snapshot in which e' is active, implying that this is the relevant snapshot of both c_u^i and c_v^j . Since π is a proper coloring of G , there at least one $w \in \{u, v\}$, such that $\pi(w) \neq c$. Without loss of generality assume that $w = v$. Thus, by **Item 8**, $(c_v^j, t) \in \mathcal{T}$. Hence, the temporal edge (e', t) is covered, implying that $\mathcal{T}$ is a k -activity ℓ -timeline covering all temporal edges of $\mathcal{G}$.

($\Leftarrow$) Let $\mathcal{T}$ be a k -activity ℓ -timeline that covers all temporal edges of $\mathcal{G}$. First, we show that for each vertex $v \in V$, there is at least one color $C \in \{R, Y, B\}$, such that none of the vertices of C_v is part of $\mathcal{T}$ in its respective relevant snapshot. To this end, note that for each $i \in [4]$, x_v^i can only be part of $\mathcal{T}$ for at most two of the three snapshots $G_{v,r,i}$, $G_{v,y,i}$, and $G_{v,b,i}$. This is due to the fact that $k = 2$. Since in each of these three snapshots, the edge set corresponds to a star with center x_v^i and leaf set $\{c_v^i : i \in [4]\}$ for each $c \in \{r, y, b\}$, this implies that, in order to have a vertex cover in each snapshot, for at least one set $C_v \in \{R_v, Y_v, B_v\}$, all vertices of C_v are part of $\mathcal{T}$ for at least one snapshot of $G_{v,r,i}$, $G_{v,y,i}$, and $G_{v,b,i}$. Since this property is true for each $i \in [4]$, for at least one of the sets $C_v \in \{R_v, Y_v, B_v\}$, all vertices of C_v use both possible appearances in $\mathcal{T}$ in the snapshots of $\{G_{v,c,i} : c \in \{r, y, b\}, i \in [4]\}$.

With this property at hand, we can now define a 3-coloring of G . Let $\pi : V \rightarrow \{r, y, b\}$ where for each vertex $v \in V$, $\pi(v)$ is an arbitrary fixed color $c \in \{r, y, b\}$ for which all vertices of $\{c_v^j : j \in [4]\}$ use both possible appearances in $\mathcal{T}$ in the snapshots of $\{G_{v,c,i} : c \in \{r, y, b\}, i \in [4]\}$. By the above argumentation, π is properly defined. It remains to show that for each edge $uv \in E$, $\pi(u) \neq \pi(v)$. Assume towards a contradiction that there is an edge $e = uv \in E$ with $\pi(u) = \pi(v)$. Let $c := \pi(u)$, and let i and j be such that e is the i th edge incident with u in G and e is the j th edge incident with v in G . Then, by definition, $G_{e,c}$ is a snapshot of $\mathcal{G}$ that contains exclusively the edge $c_u^i c_v^j$. By definition of π and since $\pi(u) = \pi(v) = c$, for each of the two vertices c_u^i and c_v^j , both possible appearances in $\mathcal{T}$ are only in the snapshots of $\{G_{v,c,i} : c \in \{r, y, b\}, i \in [4], v \in V\}$. Hence, $c_u^i c_v^j$ is not covered by $\mathcal{T}$. This contradicts the fact that $\mathcal{T}$ covers all temporal edges of $\mathcal{G}$. Thus, we obtain $\pi(u) \neq \pi(v)$ which implies that π is a proper 3-coloring of G . ◀

► **Theorem 4.8.** *TIMELINE PDS is NP-hard even if $k = 1$, $\ell = 0$, $\text{imw} = 36$ and the underlying graph has a constant maximum degree.*

Proof. We give a reduction from 3-SAT-(2,2). In this 3-SAT variant each variable appears exactly two times positively and exactly two times negatively. This problem is NP-hard [8].

Construction: Let ϕ be an instance of 3-SAT-(2,2) with variable set $X := \{x_1, \dots, x_n\}$ and clause set $C := \{C_1, \dots, C_m\}$. We fix some order on the set of clauses which induces an order on the (non-)negated appearances of each variable. As a result, we can denote the first and second nonnegated appearance of x_i by x_i^1 and x_i^2 and the first and second negated appearances by $\bar{x}_{i,1}$ and $\bar{x}_{i,2}$. We construct the following temporal graph $\mathcal{G}$. For each variable x_i we introduce two sets with seven vertices

$$V_i^1 := \{x_i^1, \bar{x}_i^1, p_i^1, q_i^1, r_i^1, s_i^1, t_i^1\}, \quad V_i^2 := \{x_i^2, \bar{x}_i^2, p_i^2, q_i^2, r_i^2, s_i^2, t_i^2\}$$

and add six snapshots $G_{6i-5}, \dots, G_{6i}$. The first three of these six snapshots only contain a crown graph on $(V_i^1 \cup V_i^2) \setminus \{\bar{x}_i^1, \bar{x}_i^2\}$, that is, a biclique with partite sets $V_i^1 \setminus \{\bar{x}_i^1\}$ and $V_i^2 \setminus \{\bar{x}_i^2\}$ without the edges of the form $v_i^1 v_i^2$ where $v \in \{x, p, q, r, s, t\}$. We remark that each pair of the form $v^1 v^2$ in the crown graph dominates the whole crown graph. For the remaining three snapshots we do the same but for the set $(V_i^1 \cup V_i^2) \setminus \{x_i^1, x_i^2\}$. In the following, these 6 snapshots for variable x_i are denoted by X_i . We denote these as the *variable snapshots*. Note that this constructs the first $6n$ snapshots of our temporal graph $\mathcal{G}$.

For each clause C_j we add six vertices

$$U_j := \{a_j, a'_j, b_j, b'_j, c_j, c'_j\}$$

and introduce one snapshot G_{6n+j} to which we refer by G_{C_j} . This snapshot contains a triangle on the three vertices corresponding to the literal appearances in C_j (recall that the vertex x_i^j stands for the j -th appearance of x_i). Furthermore, we make each of the three vertices adjacent to two different vertices of U_j . This constructs the snapshots $G_{6n+1}, \dots, G_{6n+m}$ which we call *clause snapshots*.

Finally, for every $j \in [m]$ we add six more snapshots $G_{6n+m+6j-5}, \dots, G_{6n+m+6j}$. Here, the first three of the six snapshots only contain the triangle on $\{a_j, b_j, c_j\}$ and the remaining three only contain the triangle on $\{a'_j, b'_j, c'_j\}$. We refer to these six snapshots by Z_j . These are the *dummy snapshots*. This finishes the construction of our temporal graph $\mathcal{G} = (G_1, \dots, G_{6n+7m})$.

Observe that the interval-membership-width of $\mathcal{G}$ is exactly the number of edges in the crown graphs and therefore we have $\text{imw} = 36$.

Finally, we set $k = 1, \ell = 0, t = 76n + 21m$ and obtain the TIMELINE PDS-instance $\mathcal{I} = (\mathcal{G}, k, \ell, t)$.

Intuition: Intuitively, the six variable snapshots for each variable form a selection gadget. In particular, the size of the crown graphs and the existence of the other vertices from $V_i^1 \cup V_i^2$ force that either both x_i^1 and x_i^2 or both $\bar{x}_i^1$ and $\bar{x}_i^2$ are active in X_i and therefore not in the clause snapshots. Dominating the triangle in the snapshot G_{C_j} corresponds to satisfying the clause C_j and the vertices from U_j ensure that it is also worth to set two or three literals to true in one clause. The last $6m$ dummy snapshots are constructed with the purpose to take care of the vertices from U_j such that they do not influence the clause snapshots.

Correctness: We proceed by showing that ϕ is satisfiable if and only if there is a k -activity ℓ -timeline $\mathcal{T}$ for $\mathcal{G}$ which dominates at least $t = 76n + 21m$ temporal vertices.

($\Rightarrow$) Let α be a satisfying assignment of ϕ . If $\alpha(x_i) = 1$, then we place the activity intervals of $(V_i^1 \cup V_i^2) \setminus \{x_i^1, x_i^2\}$ such that all constructed crown graphs in X_i are dominated.

Note that this is possible since $|V_i^1 \cup V_i^2| = 14$ and there are six snapshots where one pair of vertices $\{v_i^1, v_i^2\}$ for $v \in \{\bar{x}, p, q, r, s, t\}$ in each snapshot is enough to dominate the crown graphs. Moreover, we let x_i^1 and x_i^2 be active in the snapshot corresponding to the clause in which they are contained. This means if $x_{i,1} \in C_j$, then we let x_i^1 be active in the snapshot G_{C_j} . If $\alpha(x_i) = 0$, then we do the same but for $V_i^1 \cup V_i^2 \setminus \{\bar{x}_i^1, \bar{x}_i^2\}$ and $\bar{x}_i^1, \bar{x}_i^2$. In particular, this already dominates 76 temporal vertices for each variable via the crown graphs in all variable snapshots and the four temporal vertices from the sets U_j which are dominated in the clause snapshots.

Since α is a satisfying assignment, we conclude that in all clause snapshots G_{C_j} the triangle, which corresponds to the clause, is dominated. This gives us $3m$ additionally dominated temporal vertices in the clause snapshots.

Finally, for every $j \in [m]$ we use the activities of the six vertices from U_j to dominate the triangle in every of the six snapshots in Z_j . This yields 18 dominated temporal vertices for each $j \in [m]$ in the dummy snapshots and therefore the solution dominates $76n + 21m$ temporal vertices in total.

($\Leftarrow$) Let $\mathcal{T}$ be a solution to our TIMELINE PDS-instance and suppose $\mathcal{T}$ dominates at least $76n + 21m$ temporal vertices.

First of all, observe that for each $j \in [m]$ we can assume that every vertex from U_j is active in some dummy snapshot in Z_j . This is due to the fact that every vertex from U_j has degree at most two in all snapshots and in Z_j each of them can dominate three temporal vertices without affecting any vertex outside of U_j . This yields $18m$ dominated temporal vertices. Consequently, no vertex of any U_j is active during any variable snapshot or any clause gadget.

Second, observe that it is safe to assume that all vertices from $V_i^1 \cup V_i^2$ are either active during a variable snapshot in X_i or in some clause snapshot since in every other snapshot each vertex in $V_i^1 \cup V_i^2$ is isolated. We now show that in every snapshot in X_i exactly one pair $\{v_i^1, v_i^2\} \subseteq V_i^1 \cup V_i^2$ for some $v \in \{x, \bar{x}, p, q, r, s, t\}$ is active.

If there is a snapshot G_j associated with X_i in which no vertex of $V_i^1 \cup V_i^2$ is active, then there exists a vertex $v_i^1 \in V_i^1$ from the crown graph which is active a) in a snapshot outside of X_i or b) in a snapshot in X_i together with some other vertex of V_i^1 since exactly 6 snapshots are associated with X_i . In this case, making v_i^1 active in G_j yields a solution which dominates strictly more vertices since in case a) only one temporal vertex is dominated and in case b) making v_i^1 inactive reduced the number of dominated temporal vertices by only one. Thus, in the following, it is safe to assume that in each snapshot G_j associated with X_i at least one vertex of V_i^1 is active.

Now, suppose there exist a snapshot G_j associated with X_i such that only vertices of V_i^1 are active and no vertex of V_i^2 . Since in each snapshot associated with X_i at least one vertex of V_i^1 is active, in G_j at most 2 vertices of V_i^1 are active, without loss of generality say p_i^1 and q_i^1 . Now, consider vertex p_i^2 . Making p_i^2 active during G_j dominates the 5 temporal vertices $V_i^1 \setminus \{p_i^1, q_i^1\}$. Next, consider the snapshot G_z where p_i^2 is currently active. If G_z is not associated with X_i , then p_i^2 is isolated in G_z . Otherwise, if G_z is associated with X_i at least one vertex of $V_i^1 \setminus \{p_i^1, q_i^1\}$, say s_i^1 , is active during G_z . Hence, making p_i^2 inactive at G_z only makes (a subset of the) 5 temporal vertices $V_i^1 \setminus \{p_i^1, s_i^1\}$ undominated. Consequently, it is safe to assume that p_i^2 is active during G_j . Thus, we have now shown that it is safe to assume that in every snapshot in X_i at least one pair $\{v_i^1, v_i^2\} \subseteq V_i^1 \cup V_i^2$ for some $v \in \{x, \bar{x}, p, q, r, s, t\}$ is active.

Since any pair $\{v_i^1, v_i^2\} \subseteq V_i^1 \cup V_i^2$ for some $v \in \{x, \bar{x}, p, q, r, s, t\}$ dominates all temporal vertices of the crown, it is safe to assume that per associated snapshot with X_i exactly

one pair $\{v_i^1, v_i^2\} \subseteq V_i^1 \cup V_i^2$ is active. As this holds for every $i \in [n]$, we know that by construction at most $4n + 3m$ temporal vertices in the clause snapshots can be dominated. Consequently, the solution $\mathcal{T}$ has to dominate in total $76n$ temporal vertices in the parts X_i . However, this can only be the case if exactly one pair $\{v^1, v^2\} \subseteq V_i^1 \cup V_i^2$ is active in every snapshot of X_i . Since each vertex from $V_i^1 \cup V_i^2 \setminus \{x_i^1, x_i^2, \bar{x}_i^1, \bar{x}_i^2\}$ appears only isolated outside of X_i and each vertex from $\{x_i^1, x_i^2, \bar{x}_i^1, \bar{x}_i^2\}$ dominates at least two vertices from U_j in the corresponding clause snapshot, it follows that either $\{x_i^1, x_i^2\}$ or $\{\bar{x}_i^1, \bar{x}_i^2\}$ are not active in X_i since $\mathcal{T}$ maximizes the number of dominated temporal vertices.

Finally, this allows us to define an assignment α by setting $\alpha(x_i) = 1$ if and only if $\{x_i^1, x_i^2\}$ are not active in X_i . To see that this is a satisfying assignment, recall that $\mathcal{T}$ dominates at least $76n + 21m$ temporal vertices. By the reasoning above it follows that $\mathcal{T}$ dominates in total at least $4n + 3m$ temporal vertices in the clause snapshots. Clearly, this can only be the case if in every clause snapshot one of the vertices of the contained triangle is active, which by definition of α means that α sets at least one literal appearance in every clause to 1. ◀

5 Complexity with respect to Input Parameters

Finally, we study the influence of the input parameters n, k, ℓ , and t on the complexity of TIMELINE (PARTIAL) VERTEX COVER and TIMELINE (PARTIAL) DOMINATING SET. Both TIMELINE VC and TIMELINE DS are NP-hard for constant values of k and ℓ . Consequently, larger parameters need to be considered to obtain FPT-algorithms or XP-algorithms. Froese et al. [19, Theorem 8] showed that TIMELINE VC admits an FPT-algorithm for $n + k$. A similar algorithm also works for TIMELINE DS.

► **Theorem 5.1.** *TIMELINE DS is solvable in time $\mathcal{O}(n^{2+nk}T)$.*

Proof. Let $(\mathcal{G} = (G_1, \dots, G_T), k, \ell)$ be an instance of TIMELINE DS.

The algorithm starts with an empty solution candidate $\mathcal{T} = \emptyset$ and recursively adds activity intervals to this set. Furthermore, it keeps one counter k_v for each vertex $v \in V(\mathcal{G})$ for the number of activity intervals.

If $k_v > k$ for some $v \in V$, then return no. If $\mathcal{T}$ dominates all temporal vertices in $\mathcal{G}$, then return yes. Otherwise, let i be the minimum time step such that G_i contains an undominated vertex v . For each $u \in N_{G_i}[v]$ the algorithm branches by adding $(u, i, i + \ell)$ to $\mathcal{T}$, increasing k_u by one and continuing recursively.

The size of the closed neighborhood of v in G_i is at most n . In each recursive call the algorithm selects one activity interval and a solution can contain at most nk activity intervals. Consequently, the search tree is of size $\mathcal{O}(n^{nk})$. For each recursive call, the algorithm needs to find the first undominated temporal vertex. This can be done in $\mathcal{O}(n^2T)$ time and therefore we obtain an overall running time of $\mathcal{O}(n^{nk} \cdot n^2T)$. ◀

Froese et al. [19, Theorem 12] showed that TIMELINE VC is W[1]-hard for n even if $\ell = 1$ with a reduction from UNARY BIN PACKING and using a multicolored and a nonuniform variant of TIMELINE VC as intermediate problems in the reduction. In NONUNIFORM TIMELINE VC we are not given a single number of permitted activity intervals k , but instead a number k_v for each vertex v . In a similar way we introduce NONUNIFORM TIMELINE DS, which we will use as an intermediate problem to show hardness for $n + \ell$ for TIMELINE DS.

► **Theorem 5.2.** *TIMELINE DS parameterized by n is W[1]-hard even if $\ell = 1$.*

We start by proving the hardness of the nonuniform variant.

► **Lemma 5.3.** *NONUNIFORM TIMELINE DS is $W[1]$ -hard with respect to n even if $\ell = 1$.*

Proof. In the $W[1]$ -hardness proof of TIMELINE VC, Froese et. al [19] use NONUNIFORM TIMELINE VC as an intermediate problem. In particular, they show that this problem, when parameterized by n , is $W[1]$ -hard even if $\ell = 1$. We now reduce from this problem. A sketch of the idea can be found in Figure 3.

A common reduction from VERTEX COVER to DOMINATING SET is to introduce a new vertex for each edge and make this new vertex adjacent to both endpoints of the edge. In the following reduction we will make use of this idea. However, we also need to take care of isolated temporal vertices. For this purpose, we introduce another vertex which is connected to all isolated vertices in each snapshot.

Suppose $(\mathcal{G} = (G_1, \dots, G_T), (k_v)_{v \in V}, \ell = 1)$ is a given instance of NONUNIFORM TIMELINE VC and let $V := V(\mathcal{G})$. We construct a new temporal graph $\mathcal{G}'$ with vertex set

$$V' := V \cup \{w_e : e \in E_{\downarrow}(\mathcal{G})\} \cup \{z\}.$$

For each snapshot G_i (with edge set E_i) of $\mathcal{G}$ we introduce the snapshot G'_i for $\mathcal{G}'$ with the edge set

$$\begin{aligned} E'_i := & E_i \cup \{vw_e : e \in E_i, v \in e\} \\ & \cup \{vz : v \in V, \deg_{G_i}(v) = 0\} \\ & \cup \{w_e z : e \in E_{\downarrow}(\mathcal{G}) \setminus E_i\}. \end{aligned}$$

In other words, the edge set of G'_i consists of the edges E_i , the edges between the new vertices, which were introduced for the edges from E_i and the respective endpoints and the edges, which connect all remaining isolated vertices to z . It remains to fix the individual number of activity intervals for each vertex. For each $v \in V'$ we set

$$k'_v := \begin{cases} k_v & \text{if } v \in V, \\ \lceil \frac{T}{2} \rceil & \text{if } v = z, \\ 0 & \text{otherwise.} \end{cases}$$

The idea is that z is active in each snapshot and dominates all isolated temporal vertices, while the vertices from V dominate all the remaining temporal vertices.

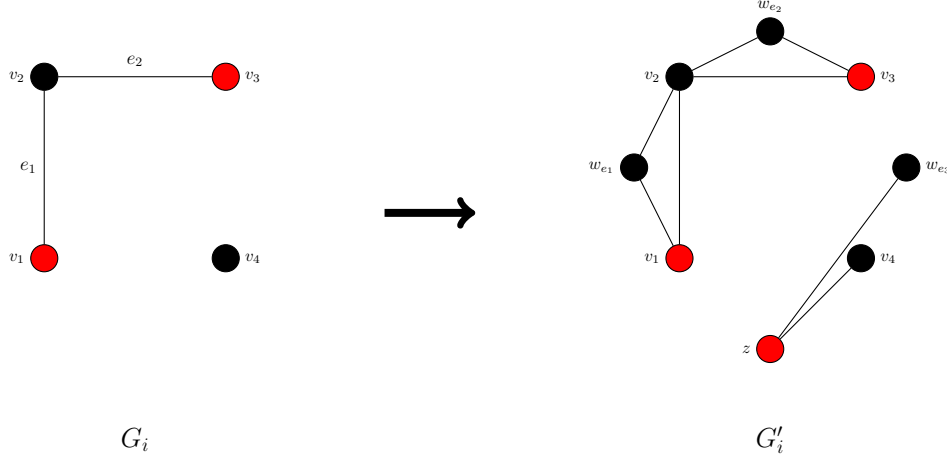
By the previous construction, the NONUNIFORM TIMELINE DS instance is given by $(\mathcal{G}' = (G'_1, \dots, G'_T), (k'_v)_{v \in V'}, \ell = 1)$. Since $|V'| \in O(|V|^2)$, the $W[1]$ -hardness follows if the reduction is correct.

Correctness: We show that $(\mathcal{G}, (k_v)_{v \in V}, \ell = 1)$ is a yes-instance of NONUNIFORM TIMELINE VC if and only if $(\mathcal{G}', (k'_v)_{v \in V'}, \ell = 1)$ is a yes-instance of NONUNIFORM TIMELINE DS.

($\Rightarrow$) Suppose the activity timeline $\mathcal{T}$ is a solution to $(\mathcal{G}, (k_v)_{v \in V}, \ell)$. In particular, $\mathcal{T}$ covers all temporal edges in $\mathcal{G}$. Consider

$$\mathcal{T}' := \begin{cases} \mathcal{T} \cup \bigcup_{i=1}^{\frac{T}{2}} \{(z, 2i-1, 2i)\} & \text{if } T \text{ is even,} \\ \mathcal{T} \cup \bigcup_{i=1}^{\lfloor \frac{T}{2} \rfloor} \{(z, 2i-1, 2i)\} \cup \{(z, T, T)\} & \text{if } T \text{ is odd.} \end{cases}$$

Intuitively, $\mathcal{T}'$ is basically $\mathcal{T}$ and additionally the vertex z is active in each snapshot. The constraints $(k'_v)_{v \in V'}$ for the number of activity intervals of each vertex are clearly satisfied. Since z is active in every snapshot, it remains to show that all temporal vertices, which are not a neighbor of z , are dominated. Assume (v, i) is such a temporal vertex. We have to distinguish between two cases. If $v \in V$ and $v \notin N_{G_i}(z)$, then $uv \in E_i$ for some $u \in V$. This



■ **Figure 3** A sketch of the reduction from NONUNIFORM TIMELINE VC to NONUNIFORM TIMELINE DS in [Lemma 5.3](#). For each snapshot G_i of $\mathcal{G}$ one snapshot G'_i is introduced for $\mathcal{G}'$. In the example shown in the figure, it is assumed that the underlying graph of $\mathcal{G}$ has the edge set $E_{\downarrow} = \{e_1, e_2, e_3\}$. For each of these edges one vertex is introduced (with zero activity intervals) and connected to the respective endpoints in G'_i if and only if the edge appears in G_i . Further, all the isolated vertices are connected to z , which is assigned enough activity intervals to be active in every snapshot. The vertices marked in red are active.

implies that u or v is active in snapshot G_i with respect to $\mathcal{T}$ (because the temporal edge (uv, i) has to be covered) and therefore (v, i) is dominated in $\mathcal{G}'$ because $\mathcal{T} \subseteq \mathcal{T}'$. If $v = w_e$ for some $e \in E_i$ and $v \notin N_{G_i}(z)$, then by construction $w_e u, w_e u' \in E'_i$ for some $u, u' \in V$ where $e = uu' \in E_i$. In particular, u or u' is active with respect to $\mathcal{T}$ and therefore (v, i) is dominated in $\mathcal{G}'$ by $\mathcal{T}'$.

($\Leftarrow$) Suppose the activity timeline $\mathcal{T}'$ is a solution to $(\mathcal{G}', (k'_v)_{v \in V'}, \ell)$ and define

$$\mathcal{T} := \{(v, a, b) \in \mathcal{T}' : v \in V\},$$

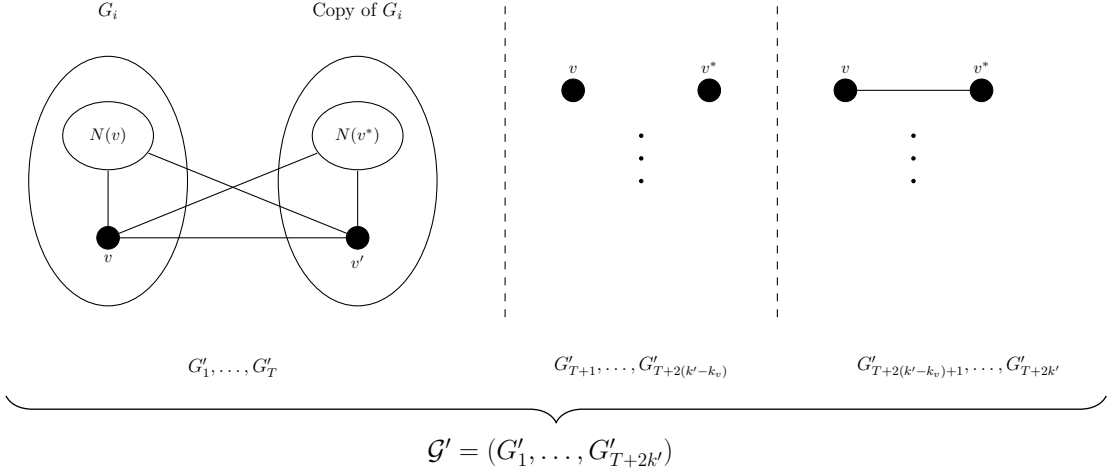
that is, $\mathcal{T}$ is simply the set of activity intervals of vertices from V , which are contained in $\mathcal{T}'$. By construction, it is clear that the constraints on the number of activity intervals are satisfied. We show that $\mathcal{T}$ covers all temporal edges in $\mathcal{G}$. If $e = uv \in E_i$, then w_e is dominated by u or v in G'_i , because w_e is not adjacent to z in G'_i and has no own activity intervals. So in particular, u or v is active in G'_i with respect to $\mathcal{T}'$. By definition $\mathcal{T}$ contains the same activity intervals of u and v as $\mathcal{T}'$. It follows that u or v is active in G_i with respect to $\mathcal{T}$ and therefore the temporal edge (e, i) is covered by $\mathcal{T}$. $\blacktriangleleft$

With [Lemma 5.3](#) we are now ready to prove [Theorem 5.2](#), that is, the $W[1]$ -hardness of TIMELINE DS with respect to n , even if $\ell = 1$.

Proof of Theorem 5.2. We want to make use of [Lemma 5.3](#), this means we provide a reduction from NONUNIFORM TIMELINE DS with $\ell = 1$ to TIMELINE DS with $\ell' = 1$. A sketch of the reduction can be found in [Figure 4](#).

Let $I := (\mathcal{G} = (G_1, \dots, G_T), (k_v)_{v \in V}, \ell = 1)$ be an instance of NONUNIFORM TIMELINE DS with vertex set $V := V(\mathcal{G})$. The maximum number of permitted activity intervals $k' := \max_{v \in V} k_v$ will be the permitted number of activity intervals in the TIMELINE DS instance $I' := (\mathcal{G}', k', \ell' = 1)$. We construct a new temporal graph $\mathcal{G}'$ on the vertex set

$$V' := V(\mathcal{G}') = V \cup \{v^* : v \in V\}.$$



■ **Figure 4** A sketch of the reduction from NONUNIFORM TIMELINE DS to TIMELINE DS in **Theorem 5.2**. If $\mathcal{G} = (G_1, \dots, G_T)$ is the given temporal, then the newly constructed temporal graph $\mathcal{G}'$ consists of $T + 2k'$ snapshots, where k' is the maximum number of permitted activity intervals in the nonuniform variant. The reduction introduces for each vertex v a copy v^* and the figure sketches how these two vertices are linked in each snapshot. Note that the number k_v of permitted activity intervals may be different for each vertex $v \in V(\mathcal{G})$.

For this we introduce $T + 2k'$ snapshots to $\mathcal{G}'$ where snapshot G'_i has the edge set

$$E'_i := \begin{cases} \bigcup_{v \in V} \{vv^*\} \cup \bigcup_{vw \in E_i} \{vw, vw^*, v^*w, v^*w^*\} & \text{if } i \in [T], \\ \{vv^* : v \in V \text{ such that } i > T + 2(k' - k_v)\} & \text{if } i \in [T + 1, T + 2k']. \end{cases}$$

Intuitively, for $i \in [T]$ the snapshot G'_i is constructed from G_i by taking G_i and a copy of G_i and making v and its copy v^* adjacent to the same vertices. In the next $2(k' - k_v)$ snapshots ($G'_{T+1}, \dots, G'_{T+2(k'-k_v)}$) the vertex v and its copy v^* remain isolated and in the last $2k_v$ snapshots ($G'_{T+2(k'-k_v)+1}, \dots, G'_{T+2k'}$) they are only incident with the edge vv^* . Consequently, v and v^* dominate the same set of vertices in each every snapshot of $\mathcal{G}'$. Further, observe that for $i \in [T]$ a set of vertices from V is a dominating set of G_i if and only if it is a dominating set of G'_i .

Correctness: We show that I is a yes-instance of NONUNIFORM TIMELINE DS if and only if I' is a yes-instance of TIMELINE DS. Recall that $\ell = \ell' = 1$.

($\Rightarrow$) Suppose the activity timeline $\mathcal{T}$ is a solution for I and consider

$$\begin{aligned} \mathcal{T}' := \mathcal{T} \cup \bigcup_{v \in V} \bigg(& \bigcup_{i=1}^{k'-k_v} \{(v, T+2i-1, T+2i), (v', T+2i-1, T+2i)\} \\ & \cup \bigcup_{i=k'-k_v+1}^{k'} \{(v', T+2i-1, T+2i)\} \bigg). \end{aligned}$$

The activity timeline $\mathcal{T}'$ has the same activity intervals as $\mathcal{T}$ in the first T snapshots. Moreover, for each vertex $v \in V$ the vertices v and v^* are both active in the $2(k' - k_v)$ snapshots $G'_{T+1}, \dots, G'_{T+2(k'-k_v)}$ and v^* is also active in the last $2k_v$ snapshots $G'_{T+2(k'-k_v)+1}, \dots, G'_{T+2k'}$. Note that each vertex has at most k' activity intervals (since v has at most k_v activity intervals in $\mathcal{T}$) and each of them is of length at most one. It remains to show that $\mathcal{T}'$ dominates all temporal vertices in $\mathcal{G}'$. For $i \in [T]$, any dominating

set in G_i is also a dominating set in G'_i and therefore $\mathcal{T}'$ dominates all temporal vertices in the first T snapshots (because $\mathcal{T}$ does). For $i \in [T+1, T+2(k'-k_v)]$, the vertices v and v^* are both active and therefore dominated. For $i \in [T+2(k'-k_v)+1, T+2k']$ they are dominated by v^* , because the edge vv^* exists and v^* is active. Consequently, all temporal vertices are dominated and $\mathcal{T}'$ is a valid solution to I' .

($\Leftarrow$) Now suppose $\mathcal{T}'$ is a solution to I' . Informally, the activity timeline $\mathcal{T}$ for the nonuniform instance will contain the activity intervals of a vertex v , which correspond to the parts of activity intervals of v or v^* , which intersect the first T snapshots. Formally, we set

$$\mathcal{T} := \{(v, a, \min(b, T)) : (v, a, b) \in \mathcal{T}' \text{ or } (v^*, a, b) \in \mathcal{T}' \text{ where } v \in V, a \leq T\}.$$

We show that in total at most k_v activity intervals of v and v^* intersect the first T snapshots. Recall that the activity intervals have length one and therefore each of them intersects exactly two snapshots. In the last $2k_v - 1$ snapshots the vertices v and v^* are only adjacent to each other and therefore in total k_v activity intervals from them have to intersect these snapshots. Since they have length one, none of them can intersect the $2(k' - k_v) - 1$ snapshots $G'_{T+2}, \dots, G'_{T+2(k'-k_v)}$. In these snapshots both v and v^* are isolated, so $(k' - k_v)$ activity intervals of each of them have to intersect these snapshots. Again they have length one, which implies that they cannot intersect the first T snapshots. Consequently, there are in total $2(k' - k_v) + k_v$ activity intervals of v and v^* , which do not intersect the first T snapshots. It follows that $\mathcal{T}$ contains at most k_v activity intervals of v . We show that $\mathcal{T}$ dominates all temporal vertices in $\mathcal{G}$. For this recall that v and v^* dominate the exact same vertices in every snapshot $G'_1, \dots, G'_T$ and that a set of vertices from V is a dominating set of G_i if and only if it is a dominating set of G'_i . $\blacktriangleleft$

The W[1]-hardness results for both problems do not apply to the case $\ell = 0$. For TIMELINE VC, this case is fixed-parameter tractable because there exists an ILP formulation in which the number of variables is bounded by some function of n [19, Lemma 10]. It is also straightforward to extend this ILP formulation to TIMELINE PVC. To complete the picture, we extend the ILP formulation also to TIMELINE PDS, thus showing that it is fixed-parameter tractable for $\ell = 0$ as well.

► **Theorem 5.4.** *TIMELINE PDS is fixed-parameter tractable with respect to n if $\ell = 0$.*

Proof. Suppose $(\mathcal{G} = (G_1, \dots, G_T), k, \ell)$ is an instance of TIMELINE PDS with $\ell = 0$. We provide an ILP formulation (adaption from [19]), where the number of variables is upper-bounded by some function of n . For this we define the following

- variables X_E^S with $S \subseteq V$ and $E \subseteq \binom{V}{2}$, which correspond to the number of times the vertex set S is active in a snapshot with edge set E ,
- constants $\alpha(E)$ storing the number of snapshots with edge set E ,
- constants t_E^S storing the number of dominated vertices, if S is the set of active vertices and the snapshot has edge set E .

Note that the number of these variables and constants only depends on n . In particular, there are $2^{\binom{n}{2}+n}$ variables X_E^S . With this we are now able to formulate our integer linear

program:

$$\sum_{E \subseteq \binom{V}{2}} \sum_{S \subseteq V} t_E^S X_E^S \geq t \quad (1)$$

$$\sum_{S \subseteq V} X_E^S = \alpha(E), \quad \text{for all } E \subseteq \binom{V}{2} \quad (2)$$

$$\sum_{E \subseteq \binom{V}{2}} \sum_{\substack{S \subseteq V: \\ v \in S}} X_E^S \leq k, \quad \text{for all } v \in V \quad (3)$$

$$X_E^S \in \mathbb{N}_0 \quad (4)$$

Correctness: We show that this ILP is feasible if and only if $(\mathcal{G}, k, \ell = 0)$ is a yes-instance of TIMELINE PDS.

($\Rightarrow$) Suppose a solution to the ILP is given. We construct an activity timeline $\mathcal{T}$ for $(\mathcal{G}, k, \ell = 0)$. For this let $S_1, S_2, \dots$ be some fixed order on the subsets of V . Now use $X_E^{S_1}$ activity intervals of exactly the vertices from S_1 in the first $X_E^{S_1}$ snapshots with edge set E . Then use $X_E^{S_2}$ activity intervals of exactly the vertices from S_2 for the next $X_E^{S_2}$ snapshots with edge set E , and so on. By doing this for every edge set E , we obtain a k -activity 0-timeline $\mathcal{T}$. Constraint (1) of the ILP ensures that at least t temporal vertices are dominated, while constraint (3) ensures that at most k activity intervals of each vertex are selected. Finally (2) guarantees that the number of snapshots with edge set E is respected.

($\Leftarrow$) Consider an activity timeline $\mathcal{T}$, which is a solution to $(\mathcal{G}, k, \ell = 0)$. Set X_E^S to the number of times the vertex set S is active in a snapshot with edge set E (with respect to the solution $\mathcal{T}$). Condition (1) of the ILP is satisfied, because $\mathcal{T}$ dominates at least t temporal vertices, and condition (3) holds, because $\mathcal{T}$ contains at most k activity intervals of each vertex. Finally, the conditions (2) and (4) are direct consequences of the definition of X_E^S . $\blacktriangleleft$

Finally, we show that both TIMELINE PDS and TIMELINE PVC can be solved in $2^{\mathcal{O}(t)} \cdot (n+T)^{\mathcal{O}(1)}$ time, where t denotes the number of temporal vertices/edges, which need to be dominated/covered. Recall that Dondi et al. [13, Theorem 6] provided an FPT-algorithm for TIMELINE PVC with running time $t^t \cdot (n+T)^{\mathcal{O}(1)}$ if $k = 1$. Hence, our approach improves upon their running time and is not limited to $k = 1$. The idea is to use color coding [2], a very popular technique to obtain FPT-algorithms [7]. To provide deterministic algorithm, we use (n, k) -perfect hash families which can be computed efficiently [28].

Initially, we show our result for TIMELINE PDS.

► **Theorem 5.5.** *TIMELINE PDS is solvable in $2^{\mathcal{O}(t)} \cdot (n+T)^{\mathcal{O}(1)}$ time.*

Proof. Let $(\mathcal{G} = (G_1, \dots, G_T), k, \ell, t)$ be a TIMELINE PDS-instance. We assume that $T \geq k(\ell + 1)$ (otherwise the instance is trivial) and fix some order $v_1, \dots, v_n$ of the vertices from $V(\mathcal{G})$. The idea is to use color coding, so we assume that our temporal vertices are colored randomly with t colors. We say that an activity timeline $\mathcal{T}$ dominates a color, if it dominates a temporal vertex of the respective color and we denote with $C_{a,b}^j$ the colors, which are dominated by the activity interval (v_j, a, b) . Now, consider the following dynamic programming table.

Table Definition: For a set $S \subseteq \{1, \dots, t\}$ of colors, and integers $j \in [0, n], k' \in [0, k]$ we define

$$\text{DP}[S, j, k'] \doteq \text{true}$$

if and only if there exists a k -activity ℓ -timeline $\mathcal{T} \subseteq \{v_1, \dots, v_j\} \times [T] \times [T]$ such that

- (i) $|\mathcal{T} \cap (\{v_j\} \times [T] \times [T])| = k'$, and
- (ii) $\mathcal{T}$ dominates the colors from S .

In other words, $\text{DP}[S, j, k'] \triangleq \text{true}$ if and only if the colors from S can be dominated by using k activity intervals of $v_1, \dots, v_{j-1}$ and k' activity intervals of v_j .

Initialization: We set the entry $\text{DP}[S, 0, k]$ to **true** if and only if $S = \emptyset$.

Recurrence: We consider the case where $k' = 0$ and $k' > 0$ individually:

$$\text{DP}[S, j, k'] = \begin{cases} \bigvee_{a \in [T-\ell]} \text{DP}[S \setminus C_{a,a+\ell}^j, j, k' - 1], & \text{if } k' > 0, \\ \text{DP}[S, j - 1, k], & \text{if } k' = 0. \end{cases}$$

Correctness: We prove the correctness of the computation by showing two directions.

($\Rightarrow$) Suppose $\text{DP}[S, j, k'] = \text{true}$. By definition, there exists a k -activity ℓ -timeline $\mathcal{T} \subseteq \{v_1, \dots, v_j\} \times [T] \times [T]$, which satisfies (i) and (ii). Otherwise, if $k' > 0$, then there exists an activity interval $(v_j, a, a + \ell) \in \mathcal{T}$. Consequently, $\mathcal{T} \setminus \{(v_j, a, a + \ell)\}$ is considered in the definition of $\text{DP}[S \setminus C_{a,a+\ell}^j, j, k' - 1]$ and satisfies (i) and (ii). Therefore, we have $\text{DP}[S \setminus C_{a,a+\ell}^j, j, k' - 1] = \text{true}$. If $k' = 0$, then $\mathcal{T}$ only contains activity intervals of $v_1, \dots, v_j$. This means that $\mathcal{T}$ considered in the definition of the table entry $\text{DP}[S, j - 1, k]$ and subsequently $\text{DP}[S, j - 1, k] = \text{true}$.

($\Leftarrow$) Let $k' > 0$ and $\text{DP}[S \setminus C_{a,a+\ell}^j, j, k' - 1] = \text{true}$ for some $a \in [T - \ell]$. If $\mathcal{T}$ is considered in the definition of $\text{DP}[S \setminus C_{a,a+\ell}^j, j, k' - 1]$ and satisfies (i) and (ii), then the activity timeline $\mathcal{T} \cup \{(v_j, a, a + \ell)\}$ is considered in the definition of $\text{DP}[S, j, k']$ and also satisfies (i) and (ii). This implies $\text{DP}[S, j, k'] = \text{true}$. If $k' = 0$, then each k -activity ℓ -timeline, which is considered in the definition of $\text{DP}[S, j - 1, k]$ and satisfies (i) and (ii), is also considered in the table entry $\text{DP}[S, j, k' = 0]$ and satisfies (i) and (ii) for this entry. Therefore, $\text{DP}[S, j - 1, k] = \text{true}$ implies $\text{DP}[S, j, k] = \text{true}$.

Result: Finally, we return yes if and only if $\text{DP}[\{1, \dots, t\}, n, k] = \text{true}$. In particular, we return yes if and only if there exists a k -activity ℓ -timeline, which dominates all colors from $S = \{1, \dots, t\}$.

Running Time: The table has size $\mathcal{O}(2^t \cdot n \cdot k)$ and each update takes $\mathcal{O}(n \cdot T^2)$ time (to determine the set $C_{a,a+\ell}^j$ for all $a \in [T]$). This yields a running time of $\mathcal{O}(2^t \cdot n^2 \cdot T^2 \cdot k)$. In the beginning of the algorithm we assumed that some random coloring of the temporal vertices is given. Instead of using random colorings, we can use derandomization based on (n, k) -perfect hash families; see [7] for details. This guarantees that for each set of t temporal vertices, we deterministically find a coloring, which colors these t temporal vertices with exactly t colors. This means, there is a solution to $(\mathcal{G}, k, \ell, t)$ if and only if the algorithm from above returns yes for one of the constructed colorings. So, running the algorithm for each of the colorings of the construction, solves our problem instance. The overall running time can then be bounded by $(2e)^t t^{\mathcal{O}(\log t)} \cdot n^3 \log(n) \cdot T^3 k$, which is $2^{\mathcal{O}(t)} \cdot (n + T)^{\mathcal{O}(1)}$ time. $\blacktriangleleft$

The algorithm of [Theorem 5.5](#) also works analogously for TIMELINE PVC: We only need to color the edges instead of the vertices.

► **Corollary 5.6.** *TIMELINE PVC is solvable in $2^{\mathcal{O}(t)} \cdot (n + T)^{\mathcal{O}(1)}$ time.*

6 Conclusion

We studied the classical and parameterized complexity of TIMELINE (PARTIAL) VERTEX COVER and TIMELINE (PARTIAL) DOMINATING SET. We showed that all problems admit FPT-algorithms for $\text{vimw} + k + \ell$, where vimw is the vertex-interval-membership width.

Our running time bounds also give XP-algorithms for vimw alone. For **TIMELINE VC** this improves upon an XP-algorithm for n [19]. Moreover, we showed that the smaller parameter $\text{imw} + k + \ell$ leads to para-NP-hardness for all problems except **TIMELINE DS**. Hence, we discovered a parameter where a dominating set problem is tractable while the corresponding vertex cover variant is not.

For future work it is interesting to investigate whether already the parameter $\text{vimw} + k$ yields an FPT-algorithm for any problem in our study. Moreover, it is open whether **TIMELINE PVC** and **TIMELINE PDS** admit an FPT-algorithm with respect to $n + k$. Even for $k = 1$ fixed-parameter tractability with respect to n is still open. Moreover, one could investigate the complexity with respect to other temporal graph parameters such as the temporal neighborhood diversity [16] or temporal graph parameters that are sensitive to the order of the snapshots [5, 6].

A further problem related to **TIMELINE VC** is **MINTIMELINE₊**. In this problem, the value of ℓ bounds the sum of the lengths of the activity intervals instead of the length of the longest activity interval. **MINTIMELINE₊** was also introduced by Rozhenstein et al. [30] and studied in several works [10–12, 14, 19, 26, 31]. To distinguish this problem from the other problems in our naming convention, **TIMELINE VC** (**MINTIMELINE_∞**) could be called **MINMAX TIMELINE VC** and **MINTIMELINE₊** could be called **SUM TIMELINE VC**, and the timeline variants of **DOMINATING SET** could be named analogously. It is open which of our positive results for vimw and imw can be transferred to the **SUM** variants of the problems. Finally, it is interesting to study other classic problems as **FEEDBACK VERTEX SET** in the **MINMAX TIMELINE** and **SUM TIMELINE** setting.

References

- 1 Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Viktor Zamaraev. Temporal vertex cover with a sliding time window. *J. Comput. Syst. Sci.*, 107:108–123, 2020.
- 2 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.
- 3 Benjamin Merlin Bumpus and Kitty Meeks. Edge exploration of temporal graphs. *Algorithmica*, 85(3):688–716, 2023.
- 4 Arnaud Casteigts. *A Journey through Dynamic Networks (with Excursions)*. Habilitation thesis, Université de Bordeaux, 2018.
- 5 Arnaud Casteigts, Nils Morawietz, and Petra Wolf. Distance to Transitivity: New Parameters for Taming Reachability in Temporal Graphs. In *Proceedings of the 49th International Symposium on Mathematical Foundations of Computer Science (MFCS '24)*, volume 306 of *LIPICs*, pages 36:1–36:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
- 6 Filippas Christodoulou, Pierluigi Crescenzi, Andrea Marino, Ana Silva, and Dimitrios M. Thilikos. Making the interval membership width of temporal graphs connected and bidirectional. In *Proceedings of the 35th International Workshop on Combinatorial Algorithms (IWOCA '24)*, volume 14764 of *Lecture Notes in Computer Science*, pages 247–258. Springer, 2024.
- 7 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 8 Andreas Darmann and Janosch Döcker. On simplified NP-complete variants of monotone3-sat. *Discret. Appl. Math.*, 292:45–58, 2021.
- 9 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2012.
- 10 Riccardo Dondi. Insights into the Complexity of Disentangling Temporal Graphs. In *Proceedings of the 23rd Italian Conference on Theoretical Computer Science (ICTCS '22)*, volume 3284 of *CEUR Workshop Proceedings*, pages 1–13. CEUR-WS.org, 2022.
- 11 Riccardo Dondi. Untangling temporal graphs of bounded degree. *Theor. Comput. Sci.*, 969:114040, 2023.

- 12 Riccardo Dondi and Manuel Lafond. An FPT algorithm for timeline cover. *J. Comput. Syst. Sci.*, 154:103679, 2025.
- 13 Riccardo Dondi, Fabrizio Montecchiani, Giacomo Ortali, Tommaso Piselli, and Alessandra Tappini. Partial temporal vertex cover with bounded activity intervals. In *Proceedings of the 3rd Symposium on Algorithmic Foundations of Dynamic Networks (SAND '24)*, volume 292 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
- 14 Riccardo Dondi and Alexandru Popa. Exact and approximation algorithms for covering timeline in temporal graphs. *Ann. Oper. Res.*, 351(1):609–628, 2025.
- 15 Rodney G. Downey and Michael Ralph Fellows. *Fundamentals of Parameterized Complexity*. Springer Science & Business Media, 2013.
- 16 Jessica A. Enright, Samuel D. Hand, Laura Larios-Jones, and Kitty Meeks. Structural parameters for dense temporal graphs. In *Proceedings of the 49th International Symposium on Mathematical Foundations of Computer Science (MFCS '24)*, volume 306 of *LIPIcs*, pages 52:1–52:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
- 17 David Eppstein and Emma S. Spiro. The h-index of a graph and its application to dynamic subgraph statistics. *J. Graph Algorithms Appl.*, 16(2):543–567, 2012.
- 18 Till Fluschnik, Rolf Niedermeier, Valentin Rohm, and Philipp Zschoche. Multistage vertex cover. *Theory Comput. Syst.*, 66(2):454–483, 2022.
- 19 Vincent Froese, Pascal Kunz, and Philipp Zschoche. Disentangling the computational complexity of network untangling. *Theory Comput. Syst.*, 68(1):103–121, 2024.
- 20 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- 21 M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the 6th Annual ACM Symposium on Theory of Computing*, pages 47–63. ACM, 1974.
- 22 Thekla Hamm, Nina Klobas, George B. Mertzios, and Paul G. Spirakis. The complexity of temporal vertex cover in small-degree graphs. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI '22)*, pages 10193–10201. AAAI Press, 2022.
- 23 Anton Herrmann. On the complexity of computing optimal dominating sets in temporal graphs. Master's thesis, Friedrich-Schiller-Universität Jena, 2024.
- 24 Anton Herrmann, Christian Komusiewicz, Nils Morawietz, and Frank Sommer. Temporal dominating set and temporal vertex cover under the lense of degree restrictions. In *Proceedings of the 4th Symposium on Algorithmic Foundations of Dynamic Networks (SAND '25)*, volume 330 of *LIPIcs*, pages 16:1–16:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.
- 25 David C. Kutner and Laura Larios-Jones. Temporal reachability dominating sets: Contagion in temporal graphs. *J. Comput. Syst. Sci.*, 155:103701, 2026.
- 26 Giorgio Lazzarinetti, Sara Manzoni, Italo Zoppis, and Riccardo Dondi. FastMinTC+: A Fast and Effective Heuristic for Minimum Timeline Cover on Temporal Networks. In *Proceedings of the 31st International Symposium on Temporal Representation and Reasoning (TIME '24)*, volume 318 of *LIPIcs*, pages 20:1–20:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
- 27 Jayadev Misra and David Gries. A constructive proof of Vizing's theorem. *Inf. Process. Lett.*, 41(3):131–133, 1992.
- 28 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS '95)*, pages 182–191. IEEE Computer Society, 1995.
- 29 Tayler Pino, Salimur Choudhury, and Fadi Al-Turjman. Dominating set algorithms for wireless sensor networks survivability. *IEEE Access*, 6:17527–17532, 2018.
- 30 Polina Rozenshtein, Nikolaj Tatti, and Aristides Gionis. The network-untangling problem: from interactions to activity timelines. *Data Min. Knowl. Discov.*, 35(1):213–247, 2021.

- 31 Carsten Schubert. Leveraging graph structure to untangle temporal networks efficiently. Master's thesis, TU Berlin, Institute of Software Engineering and Theoretical Computer Science, 2023.
- 32 I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Trans. Parallel Distributed Syst.*, 13(1):14–25, 2002.
- 33 M Verheije. Algorithms for domination problems on temporal graphs. Master's thesis, Utrecht University, Graduate School of Natural Sciences, 2021.
- 34 Vadim G Vizing. On an estimate of the chromatic class of a p -graph. *Discret Analiz*, 3:25–30, 1964.