Algorithmica (2023) 85:2028–2064
https://doi.org/10.1007/s00453-022-01077-w

Multistage *s–t* Path: Confronting Similarity with Dissimilarity

Till Fluschnik[1] · Rolf Niedermeier[1] · Carsten Schubert[1] · Philipp Zschoche[1]

Received: 5 November 2020 / Accepted: 28 November 2022 / Published online: 3 January 2023

Abstract

Addressing a quest by Gupta et al. (in: Proceedings of the 41st international colloquium on automata, languages, and programming (ICALP 2014), vol 8572 of LNCS. Springer, pp 563–575, 2014), we provide a first, comprehensive study of finding a short *s–t* path in the multistage graph model, referred to as the MULTISTAGE *s–t* PATH problem. Herein, given a sequence of graphs over the same vertex set but changing edge sets, the task is to find short *s–t* paths in each graph ("snapshot") such that in the found path sequence the consecutive *s–t* paths are "similar". We measure similarity by the size of the symmetric difference of either the vertex set (vertex-similarity) or the edge set (edge-similarity) of any two consecutive paths. We prove that these two variants of MULTISTAGE *s–t* PATH are already NP-hard for an input sequence of only two snapshots and maximum vertex degree four. Motivated by this fact and natural applications of this scenario e.g. in traffic route planning, we perform a parameterized complexity analysis. Among other results, for both variants, vertex- and edge-similarity, we prove parameterized hardness (W[1]-hardness) regarding the parameter path length (solution size). As a further conceptual investigation, we then modify the multistage model by asking for *dissimilar* consecutive paths. As one of the main technical results (employing so-called representative sets known from non-temporal settings), we prove that

Till Fluschnik was supported by the DFG, project TORE (NI 369/18).
An extended abstract appears in the Proceedings of the 31st International Symposium on Algorithms and Computation (ISAAC 2020). This full version now contains all proofs and details.

✉ Philipp Zschoche
zschoche@tu-berlin.de

Till Fluschnik
till.fluschnik@tu-berlin.de

Rolf Niedermeier
rolf.niedermeier@tu-berlin.de

Carsten Schubert
carsten.gm.schubert@campus.tu-berlin.de

1 Faculty IV, Algorithmics and Computational Complexity, Technische Universität Berlin, Berlin, Germany

dissimilarity allows for fixed-parameter tractability for the parameter solution size, contrasting with our W[1]-hardness proof of the corresponding similarity case. We also provide partially positive results concerning efficient and effective data reduction (kernelization).

Keywords Temporal graphs · Shortest paths · Parameterized complexity · Kernelization · Representative sets in temporal graphs

1 Introduction

Finding short paths is perhaps the most fundamental task in algorithmic graph theory and network analysis. There are numerous applications, including operations research, robotics, social network analysis, traffic and transportation, and VLSI design. More specifically, we are concerned with finding a short path connecting two designated vertices s and t. It is fair to say that for *static* graphs the algorithmics (also from a practical side) of finding short(est) paths is very well understood. This is much less so when considering path finding in *temporal* graphs, that is, graphs whose edge sets change over time,[1] a framework that in recent years received more and more attention in the field of network science. For instance, models concerned with disease spreading or traffic routing typically are more realistic when taking into account that links between network nodes change over time. In this work, we study path finding in temporal graphs with the additional ("multistage") assumption that s–t-paths for consecutive snapshots of the temporal graph shall be sufficiently "similar". We confront this with the opposite view that s–t-paths for consecutive snapshots of the temporal graph shall be significantly "dissimilar". Herein, similarity can naturally be measured both by comparing the edge sets of the s–t paths or by comparing the vertex sets of the s–t paths. Altogether, we end up with four natural problem variants.

A few words on motivation. Both scenarios address different aspects of robustness in an environment changing over time. Let us first look at the dissimilarity scenario. Here one may think of a situation where because of necessary recovery or cleansing costs (in pandemic times one may think of disinfection measures) one wants to avoid that subsequent "agents" on the way from start to goal share too many parts of their routing paths. Moreover, one may also think of applications in the context of so-called VIP routing, which address security aspects [21, 22]. As to the similarity scenario, one may think of robustness in the sense of "path maintenance": every deviation from the path used before causes additional costs (set up, preparation, checking) and thus shall be kept at a minimum. This can be interpreted in the spirit of incremental changes (evolutionary rather than radical changes) [11, 30].

Formally, a temporal graph $\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau)$ consists of a set V of vertices and lifetime τ many edge sets $E_1, E_2, \ldots, E_\tau$ over V. Finding an s–t path over time, also known as temporal s–t path, has already been studied [5, 45]. There, however, a path may use edges from $\bigcup_{i=1}^{\tau} E_i$, while in our setting we search for path sequences consisting of τ paths, one for each E_i. With focusing on similar *and* dissimilar paths

[1] Holme and Saramäki [32, 33] and Michail [40] survey algorithmic aspects of temporal graphs.

here, however, we introduce a new view on finding paths in temporal graphs. More specifically, addressing a quest of Gupta et al. [29], one of the first studies on multistage problems, this paper initiates a study of finding short s–t paths in the *multistage* model, that is, finding a short s–t path in each *snapshot* (V, E_i) of the temporal graph $\mathcal{G}$ such that consecutive s–t paths do not differ too much; formally, we have the following (where Π refers to a requested property of two consecutive paths in the solution):

Π MULTISTAGE s–t PATH (Π-MSTP)

Input: A temporal graph $\mathcal{G} = (V, E_1, E_2, \dots, E_\tau)$, two distinct vertices $s, t \in V$, and two integers $k, \ell \in \mathbb{N}_0$.

Question: Is there a sequence $(P_1, P_2, \dots, P_\tau)$ such that P_i is an s–t path in (V, E_i) with $|V(P_i)| \leq k$ for all $i \in \{1, \dots, \tau\}$, and $\text{dist}_\Pi(P_i, P_{i+1}) \leq \ell$ for all $i \in \{1, \dots, \tau - 1\}$?

The multistage model requests snapshot solutions such that (with respect to time) consecutive ones are similar to each other. Herein, similarity is measured by the symmetric difference of the sets describing the consecutive snapshot solutions. For paths, there are two natural choices for comparing: the sets of vertices and the sets of edges. Thus, we obtain two distance measures defined as follows.

$$\text{dist}_{\text{V}\triangle\text{V}}(P_i, P_{i+1}) := |V(P_i) \triangle V(P_{i+1})| \qquad (\text{V}\triangle\text{V-MstP}),$$
$$\text{dist}_{\text{E}\triangle\text{E}}(P_i, P_{i+1}) := |E(P_i) \triangle E(P_{i+1})| \qquad (\text{E}\triangle\text{E-MstP}).$$

Confronting the similarity request of the multistage framework with a dissimilarity request instead leads to the following.

$$\text{dist}_{\text{V}\cap\text{V}}(P_i, P_{i+1}) := |(V(P_i) \cap V(P_{i+1})) \setminus \{s, t\}| \qquad (\text{V} \cap \text{V-MstP}),$$
$$\text{dist}_{\text{E}\cap\text{E}}(P_i, P_{i+1}) := |E(P_i) \cap E(P_{i+1})| \qquad (\text{E} \cap \text{E-MstP}).$$

Note that we can easily compute each of the four distances in linear time.

In the following, we study the classical and parameterized complexity of all four variants E△E-MSTP, V△V-MSTP, V∩V-MSTP, and E∩E-MSTP. When performing a parameterized complexity analysis, we do not only aim for a better understanding of the influence of several natural problem parameters like path length $k - 1$ or the upper bound ℓ on the distance values between consecutive snapshots, but we also want to find out where (and why) the problem variants are potentially different from each other; in particular, this means confronting the similarity (also known as classical multistage) view with the dissimilarity view.

Our Contributions. We introduce four natural variants of the MULTISTAGE s–t PATH problem by employing four different ways to measure the distance between consecutive solutions. Doing so, seemingly for the first time for multistage models in general, we provide a systematic study on the impact on the algorithmic complexity when switching between edge and vertex distances on the one hand, and similarity versus dissimilarity distance measurements on the other hand.

We prove all four problems to be NP-complete, even in the restricted case of only two snapshots, each snapshot being series–parallel and the underlying graph

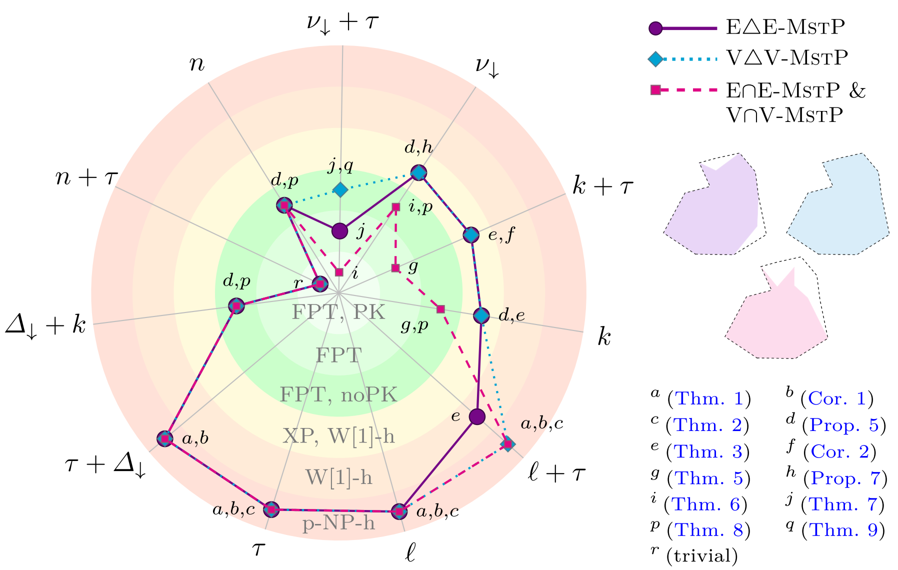

Fig. 1 Overview of our results. "p-NP-h.", "W[1]-h.", "FPT", "PK", and "noPK" respectively abbreviate para-NP-hard, W[1]-hard, fixed-parameter tractable, polynomial kernel, and "no polynomial kernel unless NP $\subseteq$ coNP / poly". Note that $\ell \leq 2k$ and $k \leq 2\nu_\downarrow + 1$

being of maximum degree four. We provide an extensive study on the parameterized complexity landscape of the problems regarding the parameters k (path length), ℓ (maximum path distance between consecutive snapshots), τ (lifetime), n (number of graph vertices), $\nu_\downarrow$ (vertex cover number of the "underlying graph"), and $\Delta_\downarrow$ (maximum vertex degree in the underlying graph); see Fig. 1 for an overview. The results of our parameterized complexity analysis reveal a clear distinction between similarity and dissimilarity. When parameterized by the maximum number k of vertices in each s–t path, while E△E-MSTP and V△V-MSTP are W[1]-hard, E∩E-MSTP and V∩V-MSTP are fixed-parameter tractable. To this end, we develop one of the first uses of the technique of representative sets [24, 41] in the context of temporal graphs. In addition, we show that, under standard complexity-theoretic assumptions, the similarity problem V△V-MSTP parameterized by the number of vertices has no polynomial kernel, while the dissimilarity problem V∩V-MSTP has one.

Related Work. We studies are within algorithmic temporal graph theory and, more specifically, contribute and extend a series of studies on the multistage model. Notably, all previous studies (on various basic computational problems) within the multistage framework adhere to the "similarity view"; we extend this by introducing also a "dissimilarity view".

To the best of our knowledge, the *multistage* model (which is a temporal model not necessarily only applying to graph problems) first appeared in 2014 in works of Eisenstat et al. [14] and Gupta et al. [29]. In a nutshell, the model considers a sequence $(I_1, \ldots, I_\tau)$ of instances of some problem P as input, and it asks for a "robust" sequence of solutions to the instances in the sense that any two consecutive solutions are similar. Several classical problems have been studied in the multistage model, both from an approximate [1–4] and from a parameterized [8, 19, 20, 23, 31, 34] algorithmics point of view. While E△E-MstP and V△V-MstP adhere to the original multistage model, our two problems E∩E-MstP and V∩V-MstP can be seen as a novel and natural variation of the multistage model by replacing the goal of consecutive similarity with consecutive dissimilarity.

Several basic temporal graph problems are closely related to the task of finding a (short) temporal s–t path (finding an s–t path over time, that is, an s–t path where the edges along the path have non-decreasing time stamps) [5, 9, 10, 15–18, 35, 36, 45, 47, 48]. While these problems typically are concerned with temporal s–t paths that may span over several snapshots of the temporal graph, in our multistage-inspired framework we aim at finding an s–t path in *each* snapshot.

We mention in passing that there is also somewhat related work on short paths in multiplex networks (also known as multilayer or multimodal networks) [27]. The main difference to our scenario is that the temporal aspect imposes an ordering of the layers whereas the multiplex view does not; in addition, Ghariblou et al. [27] perform a multiobjective optimization, being particularly interested in Pareto efficiency.

2 Preliminaries

We denote by $\mathbb{N}$ and $\mathbb{N}_0$ the natural numbers excluding and including 0, respectively. By $\log(\cdot)$ we denote the logarithm to base two. We use basic notation from graph theory and parameterized algorithmics.

Graph Theory. An undirected graph $G = (V, E)$ is a tuple consisting of a set V of vertices and a set $E \subseteq \{\{v, w\} \mid v, w \in V, v \neq w\}$ of edges. For a graph G, we also denote by $V(G)$ and $E(G)$ the vertex and edge set of G, respectively. For a vertex set $W \subseteq V$, the induced subgraph $G[W]$ is defined as the graph $(W, \{\{v, w\} \in E \mid v, w \in W\})$. A (simple) path $P = (V, E)$ is a graph with a set $V = \{v_1, \dots, v_k\}$ of distinct vertices and edge set $E = \{\{v_i, v_{i+1}\} \mid 1 \leq i < k\}$ (we often represent path P by the tuple $(v_1, v_2, \dots, v_k)$); we say that P is a v_1–v_k path. The length of a path is its number of edges. For two vertices $s, t \in V(G)$, an s–t separator $S \subseteq V(G) \setminus \{s, t\}$ is a set of vertices such that there is no s–t path in $G - S$, where $G - S = G[V \setminus S]$. We denote by $N_G(v) = \{w \in V \mid \{w, v\} \in E\}$ the neighborhood of a vertex v in G, and by $\deg(v) = |N_G(v)|$ the degree of v in G. Moreover, we denote by Δ (or $\Delta(G)$) the maximum vertex-degree of G, that is, $\Delta(G) = \max_{v \in V} \deg(v)$. A *vertex cover* of G is a set W of vertices such that $G - W$ contains no edge; we denote by ν (or $\nu(G)$) the smallest size of a vertex cover in G. A graph with distinct terminal vertices s, t is series–parallel if it can be turned into a single edge by a sequence of contractions of degree-two vertices except s and t while removing any parallel edge that appears [13].

Temporal Graph Theory. A temporal graph $\mathcal{G} = (V, E_1, E_2, \dots, E_\tau)$ consists of a set V of vertices and lifetime τ many edge sets $E_1, E_2, \dots, E_\tau$ over V. We also denote by $\tau(\mathcal{G})$ the lifetime of $\mathcal{G}$. The size of $\mathcal{G}$ is $|\mathcal{G}| := |V| + \sum_{i=1}^{\tau} |E_i|$. The static graph (V, E_i) is called the i-th snapshot. The *underlying graph* $\mathcal{G}_\downarrow$ of $\mathcal{G}$ is the static graph $(V, E_1 \cup \cdots \cup E_\tau)$. The underlying vertex cover number $\nu_\downarrow$ is $\nu(\mathcal{G}_\downarrow)$. The underlying maximum degree $\Delta_\downarrow$ is $\Delta(\mathcal{G}_\downarrow)$.

Parameterized Complexity. Let Σ denote a finite alphabet. A parameterized problem $L \subseteq \{(x, k) \in \Sigma^* \times \mathbb{N}_0\}$ is a subset of all instances (x, k) from $\Sigma^* \times \mathbb{N}_0$, where k denotes the parameter. A parameterized problem L is (i) fixed-parameter tractable if there is an algorithm that decides every instance (x, k) for L in $f(k) \cdot |x|^{O(1)}$ time, (ii) contained in the class XP if there is an algorithm that decides every instance (x, k) for L in $|x|^{f(k)}$ time, and (iii) para-NP-hard if the problem for some constant value of the parameter is NP-hard, where f is some computable function only depending on the parameter. For two parameterized problems L, L', an instance $(x, k) \in \Sigma^* \times \mathbb{N}_0$ of L is equivalent to an instance $(x', k') \in \Sigma^* \times \mathbb{N}_0$ for L' if $(x, k) \in L \iff (x', k') \in L'$. A problem L is hard for the class W[1] (W[1]-hard) if for every problem $L' \in$ W[1] there is an algorithm that maps any instance (x, k) in $f(k) \cdot |x|^{O(1)}$ time to an equivalent instance (x', k') with $k' = g(k)$ for some computable functions f, g. It holds true that FPT $\subseteq$ W[1] $\subseteq$ XP, where FPT denotes the class of all fixed-parameter tractable parameterized problems. It is believed that FPT $\neq$ W[1], and that hence no W[1]-hard problem is fixed-parameter tractable. A problem kernelization for a parameterized problem L is a polynomial-time algorithm that maps any instance (x, k) of L to an equivalent instance (x', k') of L (the kernel) such that $|x'|+k \leq f(k)$ for some computable function f; If f is a polynomial, we say that the problem kernelization (and kernel) is polynomial. It is well-known that a decidable parameterized problem is fixed-parameter tractable if and only if it admits a problem kernelization.

3 Relation Between Distance Measures: From Edges to Vertices

We show that there are polynomial-time algorithms that, given an instance of E$\triangle$E-MstP or of E$\cap$E-MstP, construct an equivalent instance of the respective vertex-counterpart.

Proposition 1 *There is an algorithm that, on every input* $(\mathcal{G}, s, t, k, \ell)$ *to* E$\triangle$E-MstP, *computes in* $\mathcal{O}(|\mathcal{G}| \cdot \ell)$ *time an equivalent instance* $(\mathcal{G}', s, t, k', \ell')$ *of* V$\triangle$V-MstP *such that* $k' \in O(k \cdot \ell)$, $\ell' \in O(\ell^2)$, $\Delta(\mathcal{G}'_\downarrow) = \max\{\Delta(\mathcal{G}_\downarrow), 2\}$, *and* $\tau(\mathcal{G}) = \tau(\mathcal{G}')$.

Proof Let $I = (\mathcal{G} = (V, E_1, \ldots, E_\tau), s, t, k, \ell)$ be an instance of E$\triangle$E-MstP. Let initially $V' = V$. For each edge $e = \{a, b\} \in E := E_1 \cup \cdots \cup E_\tau$, add the set $V_e = \{v_e^1, \ldots, v_e^{\ell+1}\}$ of $\ell + 1$ vertices to V'. For each $i \in \{1, \ldots, \tau\}$, set E'_i to $\bigcup_{e \in E_i} P_e$, where $P_e = \{\{a, v_e^1\}, \{v_e^{\ell+1}, b\}\} \cup \bigcup_{1 \leq j \leq \ell} \{\{v_e^j, v_e^{j+1}\}\}$. This finishes the construction of $\mathcal{G}' = (V', E'_1, \ldots, E'_\tau)$. Finally, set $k' = k + (k-1)(\ell+1)$ and $\ell' = (\ell+1)^2 - 1$. We claim that I is a yes-instance if and only if $I' := (\mathcal{G}', s, t, k', \ell')$ is a yes-instance.

($\Rightarrow$) Let $\mathcal{P} = (P_1, \ldots, P_\tau)$ be a solution to I. For each $i \in \{1, \ldots, \tau\}$, construct P'_i with $V(P'_i) = V(P_i) \cup \{V_e \mid e \in E(P_i)\}$ and $E(P'_i) = \{P_e \mid e \in E(P_i)\}$. Clearly P'_i is an s–t path in (V', E'_i). Moreover, $|V(P'_i)| = |V(P_i)| + |\{V_e \mid e \in E(P_i)\}| \leq k + (k-1) \cdot (\ell+1) = k'$ and $|V(P'_i) \triangle V(P'_{i+1})| \leq \ell + (\ell+1) \cdot |E(P_i) \triangle E(P_{i+1})| \leq \ell + (\ell+1)\ell = \ell'$.

($\Leftarrow$) Let $\mathcal{P}' = (P'_1, \ldots, P'_\tau)$ be a solution to I'. For each $i \in \{1, \ldots, \tau\}$, construct P_i with $V(P_i) = V(P'_i) \setminus \{V_e \mid P_e \subseteq E(P'_i)\}$ and $E(P_i) = \{e \mid P_e \subseteq E(P'_i)\}$. Clearly P_i is an s–t path in (V, E_i). Moreover, note that $k^* := |V(P'_i) \cap V| \leq k$, since otherwise we have too many vertices in P'_i, contradicting $\mathcal{P}'$ to be a solution. Hence, we have that $|V(P_i)| = |V(P'_i) \cap V| \leq k$. Further note that $|\{e \in E \mid V_e \subseteq V(P'_i) \triangle V(P'_{i+1})\}| \leq \ell$, since otherwise $|V(P'_i) \triangle V(P'_{i+1})| \geq (\ell + 1) \cdot (\ell + 1) > \ell'$. Hence, $|E(P_i) \triangle E(P_{i+1})| = |\{e \in E \mid V_e \subseteq V(P'_i) \triangle V(P'_{i+1})\}| \leq \ell$. $\square$

Proposition 2 *There is an algorithm that, on every input $(\mathcal{G}, s, t, k, \ell)$ to* E∩E-MSTP, *computes in $\mathcal{O}(|\mathcal{G}|)$ time an equivalent instance $(\mathcal{G}', s, t, k', \ell')$ of* V∩V-MSTP *such that $k' = 2k - 1$, $\ell' = \ell$, $\Delta(\mathcal{G}_\downarrow) = \max\{\Delta(\mathcal{G}'_\downarrow), 4\}$, and $\tau(\mathcal{G}) = \tau(\mathcal{G}')$.*

Proof Let $I = (\mathcal{G} = (V, E_1, \ldots, E_\tau), s, t, k, \ell)$ be an instance of E∩E-MSTP, and denote by E the set $E_1 \cup \cdots \cup E_\tau$. Define for each $v \in V \setminus \{s, t\}$ the set $V_v = V_v^0 \cup V_v^1$, where $V_v^i = \{v^i\}$ for each $i \in \{0, 1\}$, and define $V_s = \{s\}$ and $V_t = \{t\}$. Set $V^* = \bigcup_{v \in V} V_v$. We set $V' = V^* \cup \{x_e \mid e \in E\}$. Next, for each edge $e = \{v, w\} \in E$ with $v, w \notin \{s, t\}$, let $E_e^0 = \{\{v^0, x_e\}, \{w^0, x_e\}\}$ and $E_e^1 = \{\{v^1, x_e\}, \{w^1, x_e\}\}$, and for each edge $e = \{v, w\} \in E$ with $v \in \{s, t\}$ and $w \notin \{s, t\}$, let $E_e^0 = \{\{v, x_e\}, \{w^0, x_e\}\}$ and $E_e^1 = \{\{v, x_e\}, \{w^1, x_e\}\}$. If $e = \{s, t\} \in E$, then set $E_e^0 = E_e^1 = \{\{\{s, x_e\}, \{x_e, t\}\}$. Finally, let $E_e = E_e^0 \cup E_e^1$ and $E'_i = \bigcup_{e \in E_i} E_e$. Set $k' = 2k - 1$ and $\ell' = \ell$. This finishes the construction of instance $I' := (\mathcal{G}' = (V', E'_1, \ldots, E'_\tau), s, t, k', \ell')$ of V∩V-MSTP. Note that I' can be constructed in $\mathcal{O}(|\mathcal{G}|)$ time. We claim that I is a `yes`-instance if and only if I' is a `yes`-instance.

($\Rightarrow$) Let $(P_1, \ldots, P_\tau)$ be a solution to I. We claim that the sequence $(P'_1, \ldots, P'_\tau)$ with $V(P'_i) = \bigcup_{v \in V(P_i)} V_v^{i \bmod 2} \cup \{x_e \mid e \in E(P_i)\}$ and $E(P'_i) = \bigcup_{e \in E(P_i)} E_e^{i \bmod 2}$ is a solution to I'. First, observe that each P'_i is an s–t path, and $|V(P'_i)| = |V(P_i)| + |E(P_i)| \leq 2k - 1$. Moreover, $|(V(P'_i) \cap V(P'_{i+1})) \setminus \{s, t\}| = |\{x_e \mid e \in E(P_i) \cap E(P_{i+1})\}| \leq \ell = \ell'$.

($\Leftarrow$) Let $(P'_1, \ldots, P'_\tau)$ be a solution to I' such that for each P'_i it holds true that $|V_v \cap V(P'_i)| \leq 1$. Note that $V(P'_i) = \{s, t\} \uplus W_i \uplus X_i$ with $W_i \subseteq V^* \setminus \{s, t\}$ and $X_i \subseteq \{x_e \mid e \in E\}$. We claim that $(P_1, \ldots, P_\tau)$ with $V(P_i) = \{v \mid v^i \in W_i\} \cup \{s, t\}$ and $E(P_i) = \{e \mid x_e \in X_i\}$ is a solution to I. First, observe that each P_i is an s–t path, and $|V(P_i)| \leq k$. Moreover, $|E(P_i) \cap E(P_{i+1})| \leq |X_i \cap X_{i+1}| \leq \ell' = \ell$. $\square$

Due to Propositions 1 and 2, often we just may prove lower bounds for E△E-MSTP and E∩E-MSTP, and upper bounds for V△V-MSTP and V∩V-MSTP, and transfer the results to their respective counterparts.

4 NP-Hardness Even for Two Snapshots of Maximum Degree Four

In this section, we prove that all four problems are NP-hard even for only two snapshots and the maximum underlying vertex-degree being four.

Theorem 1 E△E-MSTP *and* E∩E-MSTP, *the latter with $\ell = 0$, are* NP-*hard even if $\mathcal{G}$ consists of two snapshots both being series–parallel graphs and $\Delta(\mathcal{G}_\downarrow) = 4$.*

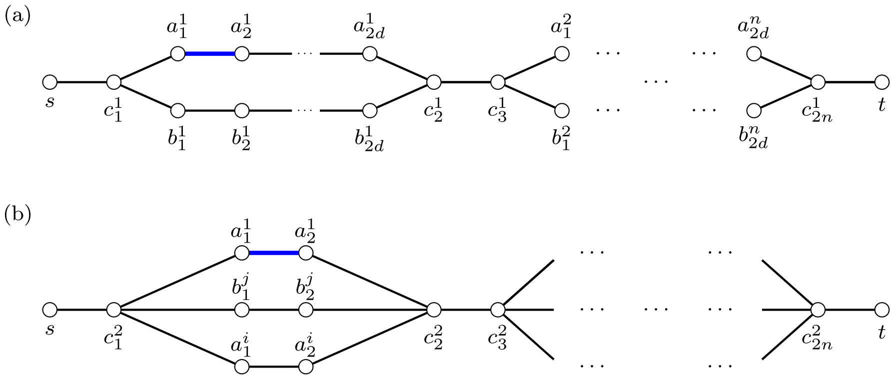

Fig. 2 Illustration of Constructions 1 with **a** illustrating the first snapshot and **b** illustrating the second snapshot, exemplified for clause $C_1 = (x_1 \vee \overline{x_j} \vee x_i)$. The edge $\{a_1^1, a_2^1\}$ is highlighted in both (**a**) and (**b**)

Proof The theorem follows directly from the forthcoming Propositions 3 and 4. □

We give two polynomial-time many-one reductions from the NP-complete 3-SAT problem, each employing the following.

Construction 1 Let $(X = \{x_1, \ldots, x_n\}, \mathcal{C} = (C_1, \ldots, C_n))$ be an instance of 3-SAT where w.l.o.g. the number n of variables equals the number of clauses, and let $d \geq 2$ denote the most frequent appearance (along the clause sequence) of any literal of some variable in X. We construct a temporal graph $\mathcal{G} = (V, E_1, E_2)$ as follows (see Fig. 2 for an illustration).

Let $V := \{s, t\} \cup \{c_1^i, \ldots, c_{2n}^i \mid i \in \{1, 2\}\} \cup \{a_1^i, \ldots, a_{2d}^i \mid x_i \in X\} \cup \{b_1^i, \ldots, b_{2d}^i \mid x_i \in X\}$. Let $E_{i,a} := \bigcup_{1 \leq j < 2d} \{\{a_j^i, a_{j+1}^i\}\}$ and let $E_{i,b} := \bigcup_{1 \leq j < 2d} \{\{b_j^i, b_{j+1}^i\}\}$. Then E_1 contains

- the edge $\left\{s, c_1^1\right\}$,
- the edge set $\bigcup_{1 \leq i \leq n} \left\{\left\{c_{2i-1}^1, a_1^i\right\}, \left\{c_{2i-1}^1, b_1^i\right\}\right\}$,
- the edge set $\bigcup_{1 \leq i \leq n} \left\{\left\{c_{2i}^1, a_{2d}^i\right\}, \left\{c_{2i}^1, b_{2d}^i\right\}\right\}$,
- the edge $\{t, c_{2n}^1\}$,
- the edge set $\bigcup_{1 \leq i < n} \left\{\left\{c_{2i}^1, c_{2i+1}^1\right\}\right\}$, and
- the edge sets $\bigcup_{1 \leq i \leq n} E_{i,a}$ and $\bigcup_{1 \leq i \leq n} E_{i,b}$.

For E_2, for each clause $C_q \in \mathcal{C}$ we define the vertex set V_{C_q} and edge set E_{C_q} as follows. If C_q contains the j-th appearance of the positive literal x_i, then add the vertices a_{2j-1}^i, a_{2j}^i to V_{C_q} and the edges $\{a_{2j-1}^i, a_{2j}^i\}, \{c_{2q-1}^2, a_{2j-1}^i\}, \{c_{2q}^2, a_{2j}^i\}$ to E_{C_q}. If C_q contains the j-th appearance of the negative literal $\overline{x_i}$, then add b_{2j-1}^i, b_{2j}^i to V_{C_q} and the edges $\{b_{2j-1}^i, b_{2j}^i\}, \{c_{2q-1}^2, b_{2j-1}^i\}, \{c_{2q}^2, b_{2j}^i\}$ to E_{C_q}. Then, we have that E_2 contains the edges $\{s, c_1^2\}$, $\{t, c_{2n}^2\}$, the edge set $\bigcup_{1 \leq i < n} \{\{c_{2i}^2, c_{2i+1}^2\}\}$, and E_{C_q} for each $q \in \{1, \ldots, n\}$. This finishes the construction of $\mathcal{G}$. It is not difficult to see that (V, E_1) and (V, E_2) are series–parallel graphs. Moreover, $\Delta(\mathcal{G}_\downarrow) = 4$. Set $k = 2 + 2n + 2d \cdot n$.

Intuitively, if an instance resulting from Constructions 1 is a yes-instance for E△E-MSTP, then the s–t path in the first snapshot selects setting variables to true or false such that the s–t path in the second snapshot can pass a literal for each clause. It follows that Constructions 1 is a polynomial-time many-one reduction.

The next two results, Propositions 3 and 4, together prove Theorem 1.

Proposition 3 E△E-MSTP *is* NP*-hard even if $\mathcal{G}$ consists of two snapshots both being series–parallel graphs and $\Delta(\mathcal{G}_{\downarrow}) = 4$.*

Proof Let $I = (X = \{x_1, \ldots, x_n\}, \mathcal{C} = (C_1, \ldots, C_n))$ be an instance of 3-SAT such that the number n of variables equals the number of clauses, and let d denote the largest number of appearances of any literal of some variable in X. Let $I' = (\mathcal{G} = (V, E), s, t, k, \ell)$ with $\ell = 5n + 2dn + 2$ and $k = 2 + 2n + 2d \cdot n$ be the instance of E△E-MSTP obtained from I using Constructions 1. We claim that I is a yes-instance if and only if I' is a yes-instance.

($\Rightarrow$) Let $X' \subseteq X$ be a solution. We construct the paths (P_1, P_2) as follows. Vertex set $V(P_1)$ contains $\{s, t\} \cup \{c^1_1, \ldots, c^1_{2n}\}$ and $V(P_2)$ contains $\{s, t\} \cup \{c^2_1, \ldots, c^2_{2n}\}$. For each $i \in \{1, \ldots, n\}$, if $x_i \in X'$, then $V(P_1)$ contains the vertices $\{a^i_1, \ldots, a^i_{2d}\}$, and if $x_i \notin X'$, then $V(P_1)$ contains $\{b^i_1, \ldots, b^i_{2d}\}$. Set $E(P_1) = E(G[V(P_1)])$. Note that P_1 is an s–t path and $|V(P_1)| = 2 + 2n + 2d \cdot n = k$. Observe that for any clause C_q we have $V_{C_q} \cap V(P_1) \neq \emptyset$, since X' is a solution. For $E(P_2)$, for each $q \in \{1, \ldots, n\}$, let $h^i_{2j-1}, h^i_{2j} \in V_{C_q} \cap V(P_1)$ with $h \in \{a, b\}$ be with smallest $i \in \{1, \ldots, n\}$, then $E(P_2)$ contains the edges $\{c^2_{2q-1}, h^i_{2j-1}\}$, $\{h^i_{2j-1}, h^i_{2j}\}$, and $\{h^i_{2j}, c^2_{2q}\}$. Note that P_2 is an s–t path in (V, E_2) with $|V(P_2)| = 2+2n+2n < k$. It remains to consider $E(P_1)\triangle E(P_2)$. Let $B = \{\{v, w\} \mid v, w \in V_{C_q} \cap V(P_2), q \in \{1, \ldots, n\}\}$. Observe that $E(P_1) \cap E(P_2) = B$, since for all other edges in $E(P_2)\setminus B$ we have that at least one endpoint is in $\{c^2_1, \ldots, c^2_{2d}\}$, which is disjoint from $V(P_1)$. Hence $|E(P_1)\triangle E(P_2)| \leq |E(P_1) \cup E(P_2)| - |E(P_1) \cap E(P_2)| = (2 + 2n + 2dn + 2 + 2n + 2n - 2) - n = 5n + 2dn + 2 = \ell$.

($\Leftarrow$) Let (P_1, P_2) be a solution to I'. Observe that for all $i \in \{1, \ldots, n\}$, $V(P_1)$ contains as a subset either the set $\{a^i_1, \ldots, a^i_{2d}\}$ or the set $\{b^i_1, \ldots, b^i_{2d}\}$. Let $X' = \{x_i \in X \mid a^i_1, \ldots, a^i_{2d} \in V(P_1)\}$. We claim that the formula of I is true when the variables in X' are set to true. Let C_q be an arbitrary clause from $\mathcal{C}$. Let $\{c^2_{2q-1}, v, w, c^2_{2q}\}$ be the vertices on the subpath from P_2 connecting c^2_{2q-1} with c^2_{2q}, where $v, w \in V_{C_q}$. Note that $\{v, w\} \in E(P_1)$, since otherwise $|E(P_1) \cup E(P_2)| - |E(P_1) \cap E(P_2)| > (2 + 2n + 2dn + 2 + 2n + 2n - 2) - n = \ell$. Hence, if $\{v, w\} = \{a^i_{2j-1}, a^i_{2j}\}$ for some $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, 2d - 1\}$, then $x_i \in X'$, setting C_q to true. Otherwise, if $\{v, w\} = \{b^i_{2j-1}, b^i_{2j}\}$ for some $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, 2d - 1\}$, then $x_i \notin X'$, setting C_q to true (x_i is negated in C_q). Since C_q was chosen arbitrarily, it follows that X' is a solution to I. □

Interestingly, Constructions 1 also gives a polynomial-time many-one reduction for E∩E-MSTP. Here the intuition is opposite: the first snapshot path selects setting the variables to the *complement* of a satisfying assignment such that the second snapshot path can pass the "clause gadgets" without passing any edge contained in the first snapshot path.

Proposition 4 E$\cap$E-MSTP *is* NP-*hard even if* $\mathcal{G}$ *consists of two snapshots both being series–parallel graphs,* $\Delta(\mathcal{G}_\downarrow) = 4$*, and* $\ell = 0$.

Proof Let $I = (X = \{x_1, \ldots, x_n\}, \mathcal{C} = (C_1, \ldots, C_n))$ be an instance of 3-SAT such that the number n of variables equals the number of clauses, and let d denote the largest appearance of any literal of some variable in X. Let $I' = (\mathcal{G} = (V, E), s, t, k, \ell)$ with $\ell = 0$ and $k = 2 + 2n + 2d \cdot n$ be the instance of E$\cap$E-MSTP obtained from I using Constructions 1. We claim that I is a `yes`-instance if and only if I' is a `yes`-instance. The proof works analogously to the proof of Proposition 1, except for the fact that P_1 selects the complement of a satisfying assignment.

($\Rightarrow$) Let $X' \subseteq X$ be a solution. We construct the paths (P_1, P_2) as follows. Vertex set $V(P_1)$ contains $\{s, t\} \cup \{c_1^1, \ldots, c_{2n}^1\}$ and $V(P_2)$ contains $\{s, t\} \cup \{c_1^2, \ldots, c_{2n}^2\}$. Let H be an auxiliary, initially empty vertex set. For each $i \in \{1, \ldots, n\}$, if $x_i \in X'$, then $V(P_1)$ contains $\{b_1^i, \ldots, b_{2d}^i\}$ and H contains $\{a_1^i, \ldots, a_{2d}^i\}$, and if $x_i \notin X'$, then $V(P_1)$ contains $\{a_1^i, \ldots, a_{2d}^i\}$ and H contains $\{b_1^i, \ldots, b_{2d}^i\}$. Note that $H \cap V(P_1) = \emptyset$. Set $E(P_1) = E(G[V(P_1)])$. Note that P_1 is an s–t path and $|V(P_1)| = 2 + 2n + 2d \cdot n = k$. Observe that $V_{C_q} \cap H \neq \emptyset$, since X' is a solution. For P_2, for each $q \in \{1, \ldots, n\}$, let $h_{2j-1}^i, h_{2j}^i \in V_{C_q} \cap H$ with $h \in \{a, b\}$ with smallest $i \in \{1, \ldots, n\}$, then $V(P_2)$ contains h_{2j-1}^i, h_2^i and $E(P_2)$ contains the edges $\{c_{2q-1}^2, h_{2j-1}^i\}$, $\{h_{2j-1}^i, h_{2j}^i\}$, and $\{c_{2q}^2, h_{2j}^i\}$. Note that P_2 is an s–t path in (V, E_2) with $|V(P_2)| = 2 + 2n + 2n < k$. It remains to consider $E(P_1) \cap E(P_2)$. Note that $E(P_1) \cap E(P_2) = \emptyset$, since $V(P_1) \cap H = \emptyset$, and $V(P_2) \cap V_{C_q} \subseteq H$ for all $q \in \{1, \ldots, n\}$.

($\Leftarrow$) Let (P_1, P_2) be a solution to I'. Observe that for all $i \in \{1, \ldots, n\}$, P_1 contains as a subset either the set $\{a_1^i, \ldots, a_{2d}^i\}$ or the set $\{b_1^i, \ldots, b_{2d}^i\}$. Let $X' = \{x_i \in X \mid b_1^i, \ldots, b_{2d}^i \in V(P_1)\}$. We claim that X' is a solution to I. Let C_q be an arbitrary clause from $\mathcal{C}$. Let $\{c_{2q-1}^2, v, w, c_{2q}^2\}$ be the vertices on the subpath from P_2 connecting c_{2q-1}^2 with c_{2q}^2, where $v, w \in V_{C_q}$. Note that $\{v, w\} \notin E(P_1)$, since otherwise $|E(P_1) \cap E(P_2)| > 0$. Hence, if $\{v, w\} = \{a_{2j-1}^i, a_{2j}^i\}$ for some $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, 2d - 1\}$, then $\{b_{2j-1}^i, b_{2j}^i\} \subseteq V(P_1)$ and hence $x_i \in X'$, setting C_q to true. Otherwise, if $\{v, w\} = \{b_{2j-1}^i, b_{2j}^i\}$ for some $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, 2d - 1\}$, then $\{a_{2j-1}^i, a_{2j}^i\} \subseteq V(P_1)$ and hence $x_i \notin X'$ setting C_q to true (x_i is negated in C_q). Since C_q was chosen arbitrarily, it follows that X' is a solution to I. $\square$

Due to Propositions 1 and 2, we get the following from Theorem 1.

Corollary 1 V$\triangle$V-MSTP *and* V$\cap$V-MSTP *with* $\ell = 0$ *are* NP-*hard even if* $\tau = 2$ *and* $\Delta(\mathcal{G}_\downarrow) = 4$.

We proved E$\cap$E-MSTP and V$\cap$V-MSTP to remain NP-hard even if $\ell = 0$ and $\tau = 2$. This leads us to ask whether for a constant value of $\ell + \tau$, E$\triangle$E-MSTP or V$\triangle$V-MSTP remain NP-hard. In fact, we prove this to be true for the vertex-variant. while leaving open whether E$\triangle$E-MSTP is contained in XP regarding $\ell + \tau$.

Theorem 2 *Even if* $\ell = 0$ *and* $\tau = 2$, V$\triangle$V-MSTP *is* NP*-hard and admits no* $2^{o(k)} \cdot (|\mathcal{G}|)^{O(1)}$*-time algorithm unless the Exponential Time Hypothesis fails.*

We give a polynomial-time reduction from the following NP-complete [26] problem.

HAMILTONIAN PATH
Input: An undirected graph $G = (V, E)$.
Question: Is there a Hamiltonian path in G, i.e., a path in G that contains every vertex of G exactly once?

Construction 2 Let $(G = (V, E))$ be an instance of HAMILTONIAN PATH and let $V = \{v_1, v_2, \dots, v_n\}$. We construct the temporal graph $\mathcal{G} = (V', E_1, E_2)$ with $V' := V \cup \{s, t\}$ as follows. Set

$$E_1 := \{\{s, v_1\}\} \cup \{\{v_n, t\}\} \cup \bigcup_{i=1}^{n-1} \{\{v_i, v_{i+1}\}\}, \text{ and}$$
$$E_2 := E \cup \bigcup_{i=1}^{n} \{\{s, v_i\}, \{t, v_i\}\}.$$

Finally, set $k = n + 2$ and $\ell = 0$.

Proof of Theorem 2 Let $I = (G = (V, E))$ be an instance of HAMILTONIAN PATH and let $V = \{v_1, \dots, v_n\}$ be enumerated. Moreover, let $I' = (\mathcal{G} = (V', E_1, E_2), s, t, k, \ell)$ be the instance obtained from I using Construction 2. We claim that I is a `yes`-instance if and only if I' is a `yes`-instance.

($\Rightarrow$) Let P be a Hamiltonian path in G with endpoints v_i and v_j. Construct (P_1, P_2) as follows. Let $V(P_1) = V'$ and $E(P_1) = E_1$. Let $V(P_2) = V'$ and $E(P_2) = E(P) \cup \{\{s, v_i\}, \{t, v_j\}\}$. Since $V(P_1) = V(P_2) = V'$, we have that $|V(P_1)| = n + 2 = k$ and $V(P_1) \triangle V(P_2) = \emptyset$. Hence, (P_1, P_2) is a solution to I'.

($\Leftarrow$) Let I' be a `yes`-instance of V$\triangle$V-MSTP and let (P_1, P_2) be a solution. By the construction of (V, E_1) and the fact that (P_1, P_2) is a solution to I', we know that $V(P_1) = V(P_2) = V'$. We construct a Hamiltonian path $P = (V_P, E_P)$ from P_2 as follows. Let $V_P = V(P_2) \setminus \{s, t\}$, and let $E_P = \{e \in E(P_2) \mid e \cap \{s, t\} = \emptyset\}$. That is, P is the subpath of P_2 where the neighbors of s and t on P_2 form the endpoints. It follows that P is a path in G containing all vertices in V, and hence, I is a `yes`-instance.

Finally, note that since $k = n + 2$, and by the fact that HAMILTONIAN PATH admits no $2^{o(n)} \cdot (n + m)^{O(1)}$-time algorithm unless the Exponential Time Hypothesis fails, the second part of the theorem follows. $\square$

5 The Role of the Parameter Path Length

In this section, we focus on the parameter k, the maximum number of vertices in any s–t path. It is not hard to see that all variants allow for an XP-algorithm when parameterized by the number k of maximal vertices in each path.

Proposition 5 V△V-MSTP, V∩V-MSTP, E△E-MSTP, *and* E∩E-MSTP, *are solvable in* $\Delta_{\max}^{O(k)} \cdot |\mathcal{G}|^{O(1)}$ *time, where* $\Delta_{\max} = \max_{i \in \{1,\ldots,\tau\}} \Delta((V, E_i))$.

Proof The proof is in line with the proof of [23, Proposition 4.2]. We sketch the proof in the general setup Π-MSTP.

Given an instance $I = (\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau), s, t, k, \ell)$, construct a directed graph $D = (V', A)$ with vertex set $V' = V_1' \uplus \ldots \uplus V_\tau' \cup \{s', t'\}$ and arc set A together with a mapping $\gamma : V' \to (2^V, 2^{\binom{V}{2}})$ as follows. For each $i \in \{1, \ldots, \tau\}$ and each s–t path P of length at most $k-1$ in (V, E_i) add a vertex v to V_i' and set $\gamma(v) = P$. It is easy to verify that a straight-forward search tree algorithm (starting in s and exploring edges until the path has length $k-1$) can enumerate all s–t paths of length $k-1$ in (V, E_i) in $O(\Delta_{\max}^k \cdot |E_i|)$ time, for any $i \in \{1, \ldots, \tau\}$. Next, for each $i \in \{1, \ldots, \tau - 1\}$, if for two vertices $v \in V_i'$ and $w \in V_{i+1}'$ it holds true that $\mathrm{dist}_\Pi(\gamma(v), \gamma(w)) \leq \ell$, then add the arc $\{v, w\}$. Finally make s' adjacent with all vertices in V_1', and t' adjacent with all vertices in V_τ'. This finishes the construction. It is not difficult to see that I is a `yes`-instance if and only if there is an s'–t' path in D (which can be checked in time linear in the size of D). □

We will prove that the parameterization with k distinguishes similarity from dissimilarity: While E△E-MSTP and V△V-MSTP are W[1]-hard regarding k (even regarding $k + \tau$), each of E∩E-MSTP and V∩V-MSTP turn out to be fixed-parameter tractable.

5.1 W[1]-Hardness for the Similarity Variant Regarding $k + \tau$ and $\nu_\downarrow$

We prove that E△E-MSTP is W[1]-hard regarding $k + \tau$ even if the upper bound ℓ on the sizes of consecutive symmetric differences is constant. Due to Proposition 1, we then obtain the same result for V△V-MSTP. The proof is by a parameterized reduction from the W[1]-complete problem MULTICOLORED CLIQUE parameterized by the clique size.

Theorem 3 *Even if* $\ell = 4$ *and each snapshot is bipartite,* E△E-MSTP *is* NP-*hard and* W[1]-*hard when parameterized by* $k + \tau$.

To prove Theorem 3, we reduce from the W[1]-complete problem MULTICOLORED CLIQUE parameterized by r.

MULTICOLORED CLIQUE
Input: An undirected, r-partite graph $G = (V_1, \ldots, V_r, E)$.
Question: Is there an r-vertex clique in G?

Intuitively, in each snapshot we order the r parts differently such that any two colors appear at least once consecutively. Hence, if there is a sequence of s–t paths through all r parts in each snapshot over the same vertex set, then this witnesses the existence of each edge of any two vertices from distinct parts. For the ordering of the r parts in the snapshots, we define the following sequence of permutations where two consecutive permutations only differ by a swap of two consecutive elements.

Definition 1 For all $1 \leq i \leq 1 + \binom{r}{2}$, let π_i^r be a permutation of $(1, \ldots, r)$ as follows. Let $\pi_1^r = (1, \ldots, r)$. For $i > 1$, let π_i^r be obtained from π_{i-1}^r as follows. Let j be the index such that $\pi_{i-1}^r(j) < \pi_{i-1}^r(j+1)$ and there is no $j' \neq j$ such that $\pi_{i-1}^r(j') < \pi_{i-1}^r(j)$ and $\pi_{i-1}^r(j') < \pi_{i-1}^r(j'+1)$. Then set $\pi_i^r(j) = \pi_{i-1}^r(j+1)$, $\pi_i^r(j+1) = \pi_{i-1}^r(j)$, and $\pi_i^r(j') = \pi_{i-1}^r(j')$ for all $j' \in \{1, \ldots, r\} \setminus \{j, j+1\}$.

Note that each pair is swapped exactly once, hence we have that $\pi^r_{1+\binom{r}{2}} = (r, r-1, \ldots, 1)$. Moreover, we have the following.

Observation 1 *For all distinct $r_1, r_2 \in \{1, \ldots, r\}$, there is an $i \in \{1, \ldots, 1 + \binom{r}{2}\}$ such that $|j_1 - j_2| = 1$, where $\pi_i^r(j_1) = r_1$ and $\pi_i^r(j_2) = r_2$.*

Next we describe the construction used in the reduction.

Construction 3 Let $(G = (V_1, \ldots, V_r, E))$ be an instance of Multicolored Clique. Let $E_{i,j} \subseteq E$ denote the set of all edges between V_i and V_j. We construct an instance $(\mathcal{G} = (V, E_1, \ldots, E_\tau), s, t, k, \ell)$ with $\tau = \binom{r}{2} + 1$ of E$\triangle$E-MstP as follows. Let $V = \{s, t\} \cup V_1 \cup \cdots \cup V_r$. Add the edge sets $\bigcup_{v \in V_{\pi_i^r(1)}} \{\{s, v\}\}$ and $\bigcup_{v \in V_{\pi_i^r(r)}} \{\{t, v\}\}$ to E_i. Moreover, add $E_{\pi_i^r(j), \pi_i^r(j+1)}$ for all $1 \leq j < r$. Set $k = r + 2$ and $\ell = 4$.

Proof of Theorem 3 Let $I = (G = (V_1, \ldots, V_r, E)$ be an instance of Multicolored Clique. Let $E_{i,j} \subseteq E$ denote the set of all edges between V_i and V_j. Let $I' = (\mathcal{G} = (V, E_1, \ldots, E_\tau), s, t, k, \ell)$ be the instance obtained from I using Constructions 3. We claim that I is a `yes`-instance if and only if I' is a `yes`-instance.

($\Rightarrow$) Let I be a `yes`-instance, and let $C \subseteq V_1 \cup \cdots \cup V_r$ form a multicolored clique in G. We claim that $(P_1, \ldots, P_\tau)$ with $V(P_i) = C \cup \{s, t\}$ and $E(P_i) = E(G_i[V(P_i)])$ is a solution to I'. Note that each P_i is an s–t path with $k = r + 2$ vertices, since in G_i the edge set $E_{\pi_i^r(j), \pi_i^r(j+1)}$ exists for $j \in \{1, \ldots, r-1\}$. Moreover, $E(P_i) \triangle E(P_{i+1})$ contains at most four edges, since $\pi_i^r = (\ldots, a, b, c, d, \ldots)$ and $\pi_{i+1}^r = (\ldots, a, c, b, d, \ldots)$, where b, c denote the two unique indices that are swapped from π_i^r to π_{i+1}^r.

($\Leftarrow$) Let $\mathcal{P} = (P_1, \ldots, P_\tau)$ be a solution to I'. Note that $|V(P_i) \cap V_x| = 1$ for all $x \in \{1, \ldots, r\}$, since each V_x forms an s–t separator and $|V(P)| \leq k = r + 2$. We claim that $V(P_i) = V(P_j)$ for all $i, j \in \{1, \ldots, \tau\}$. Suppose not, then there exists an i such that $V(P_i) \neq V(P_{i+1})$. Then there are at least five edges in $E(P_i) \triangle E(P_{i+1})$: Let $\pi_i^r = (\ldots, a, b, c, d, \ldots)$ and $\pi_{i+1}^r = (\ldots, a, c, b, d, \ldots)$, then $E(P_i) \triangle E(P_{i+1})$ is a superset of the edge set E' containing one edge in $E_{a,b}$, one edge in $E_{a,c}$, one edge in $E_{c,d}$, and one edge in $E_{b,d}$. Moreover, let x be the (smallest) index such that $V(P_i) \cap V_x \ni v \neq v' \in V(P_{i+1}) \cap V_x$. Then $E(P_i) \triangle E(P_{i+1})$ contains two edges incident with v and two edges with v', where at most two edges intersect with E' (in the case of $x \in \{b, c\}$). This contradicts the fact that $\mathcal{P}$ is a solution. Let $C = V(P_1) \setminus \{s, t\}$. We claim that C forms a multicolored clique in G. First, recall that $|C \cap V_i| = 1$ for all $i \in \{1, \ldots, r\}$. Suppose there are $v, w \in C$, $v \neq w$, such that $\{v, w\} \notin E$. Let $v \in V_i$ and $w \in V_j$. Due to Observation 1, there is a snapshot G_x that contains $E_{i,j}$. Then P_x is not an s–t path in G_x, contradicting $\mathcal{P}$ being a solution. Hence, $\{v, w\} \in E$ for all $v, w \in C$, $v \neq w$. That is, C forms a multicolored clique in G. $\square$

Due to Proposition 1, we get the following.

Corollary 2 V△V-MSTP *is* W[1]*-hard when parameterized by* $k + \tau$*, even if* ℓ *is constant.*

By Proposition 5 and since $k \leq n$, we know that E△E-MSTP and V△V-MSTP are fixed-parameter tractable regarding the number n of graph vertices. Regarding the parameter number k of path vertices (and even for $k+\tau$), by Theorem 3 and Corollary 2 we know that both problems are in XP yet W[1]-hard. Since we can assume $k \leq 2\nu_{\downarrow}+1$ (recall that $\nu_{\downarrow}$ is the vertex cover number of the underlying graph) in every instance and naturally $\nu_{\downarrow} \leq n$, we can settle the parameterized complexity regarding $\nu_{\downarrow}$:

Theorem 4 *When parameterized by* $\nu_{\downarrow}$*,* V△V-MSTP *with* $\ell = 1$ *and* E△E-MSTP *are* W[1]*-hard.*

We prove each statement of Theorem 4 separately, both proofs rely on parameterized reductions from MULTICOLORED CLIQUE.

Proposition 6 E△E-MSTP *when parameterized by* $\nu_{\downarrow}$ *is* W[1]*-hard.*

For the subsequent construction, we employ the following.

Definition 2 For $r \in \mathbb{N}$, we define for all $i, j \in \{1, \ldots, r\}$, $i \neq j$, the bijection $\pi^r_{i,j} : \{1, \ldots, r\}\setminus\{i, j\} \to \{1, \ldots, r-2\}$ such that for $x, y \in \{1, \ldots, r\}\setminus\{i, j\}$ if $x < y$, then $\pi^r_{i,j}(x) < \pi^r_{i,j}(y)$.

We now describe the construction in the reduction behind Proposition 6.

Construction 4 Let $(G = (V_1, \ldots, V_r, E))$ be an instance of MULTICOLORED CLIQUE with $n = |V_1| = \cdots = |V_r|$ and let $N := n \cdot \binom{r}{2}$. We construct a temporal graph $\mathcal{G} = (V', E_1, \ldots, E_\tau)$ with $\tau = 2N$ as follows (see Fig. 3 for an illustration). Let V' initially contain $V_1, \ldots, V_r$ and s, t. Moreover, V' contains the sets $A = \{a_1, \ldots, a_r\}$ and $B = \{b_1, \ldots, b_r\}$. Finally, V' contains the sets $C^1 = \{c^1_i \mid 0 \leq i \leq r\}$ and $C^2 = \{c^2_i \mid 0 \leq i \leq r\}$. We construct the edge set E_{odd} as follows. It contains the edges $\{s, c^1_0\}$, $\{c^1_0, a_1\}$, $\{c^1_r, b_r\}$, and $\{c^1_r, t\}$. Moreover, it contains the edges $\{b_i, c^1_i\}$, $\{c^1_i, a_{i+1}\}$ for every $1 \leq i < r$. Finally, it contains the edge set $\bigcup_{v\in V_i}\{\{a_i, v\}, \{b_i, v\}\}$ for every $i \in \{1, \ldots, r\}$. We set $E_i := E_{\text{odd}}$ for each odd $i \in \{1, \ldots, \tau\}$. Next, let ϕ be a bijection that maps each (i, v, j) to a distinct integer in $\{1, \ldots, N\}$, where $i < j$, $i, j \in \{1, \ldots, r\}$, $v \in V_i$. We construct the edge set $E_{2\phi(i,v,j)}$ as follows. We add the edges $\{s, c^2_r\}$, $\{c^2_r, b_i\}$. Then, b_i is connected with all $w \in V_i$. Next, vertex v is adjacent with a_j, and all $w \in V_i\setminus\{v\}$ are adjacent with b_j. Next, a_j is adjacent to a vertex in $w \in V_j$ if and only if $\{w, v\} \in E$. Vertices b_j and a_i are adjacent with all vertices in V_j, and vertex a_i is also adjacent with c^2_0. Let $\pi = \pi^r_{i,j} : \{1, \ldots, r\}\setminus\{i, j\} \to \{1, \ldots, r-2\}$ (see Definition 2). Then c^2_0 is adjacent with $a_{\pi^{-1}(1)}$ and c^2_{r-2} is adjacent with $b_{\pi^{-1}(r-2)}$ and with c^2_{r-1} which in turn is adjacent with t. Moreover, for all $p \in \{1, \ldots, r-3\}$ the vertex c^2_p is adjacent with $a_{\pi^{-1}(p+1)}$ and $b_{\pi^{-1}(p)}$. Finally, $a_{\pi^{-1}(p)}$ and $b_{\pi^{-1}(p)}$ are adjacent to all vertices in $V_{\pi^{-1}(p)}$. This finishes the construction of $E_{2\phi(i,v,j)}$. Set $k = 4r + 3$ and $\ell = 4r + 7$.

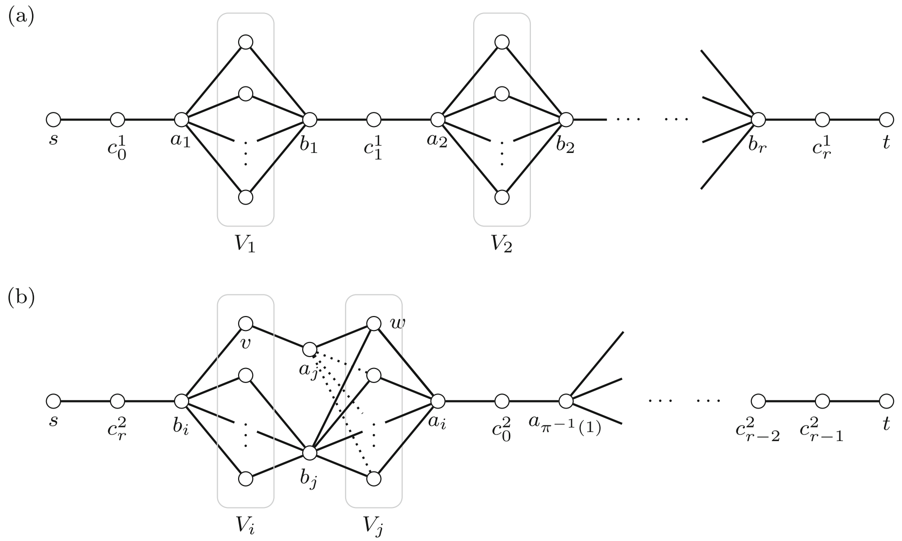

Fig. 3 Illustration of Constructions 4 with **a** showing an odd snapshot and **b** showing the even snapshot $G_{2\phi(i,v,j)}$ with edge $\{a_j, w\}$ being present assuming $\{v, w\} \in E$, and dotted edges may or may not be present (depending on E)

Observation 2 *Let $p \in \{1, \dots, N\}$. In (V, E_{2p-1}), each vertex in $A \cup B \cup C^1$, and each set V_i is an s–t separator, and in (V, E_{2p}) with $p = \phi(i, v, j)$ each vertex in $(A \backslash \{a_j\}) \cup (B \backslash \{b_j\}) \cup C^2$, each set V_i, and the set $\{a_j, b_j\}$ is an s–t separator.*

Observation 3 *Let $p \in \{1, \dots, N\}$. Every s–t path in (V, E_{2p}) with at most k' vertices contains exactly one vertex from each V_i.*

Proof For every odd snapshot, the statement is clear by construction. Consider $p = \phi(i, v, j)$ and (V, E_{2p}), and let P be an arbitrary s–t path with at most k vertices. We know from Observations 2 that every s–t path in (V, E_{2p}) contains every vertex in $(A \backslash \{a_j\}) \cup (B \backslash \{b_j\}) \cup C^2$, one vertex from each set V_i, and one vertex from $\{a_j, b_j\}$. It follows that $|V(P)| \geq 2 + |(A \backslash \{a_j\}) \cup (B \backslash \{b_j\}) \cup C^2| + r + 1 = 2 + (2r - 2 + (r + 1)) + r + 1 = 4r + 2$. Moreover, with the same argument as for the odd snapshots, it contains exactly one vertex from each set V_q with $q \in \{1, \dots, r\} \backslash \{i, j\}$. So, suppose P contains one more vertex from V_i or V_j. Then P must contain both a_j and b_j, since b_i and a_i can only appear once on any s–t path and $\{a_j, b_j\}$ separates V_i from V_j. Hence $|V(P)| = (4r + 2) + 2 = 4r + 4 > k$, yielding a contradiction. □

Since in every snapshot each vertex from $C^1 \cup C^2$ is of degree two or zero, we have the following.

Observation 4 *Let $p \in \{1, \dots, N\}$. Every s–t path in (V, E_{2p-1}) contains the edge set E'_{2p-1} consisting of all edges incident with a vertex in $\{c_0^1, \dots, c_r^1\}$. Every s–t path in (V, E_{2p}) contains the edge set E'_{2p} consisting of all edges incident with $\{c_0^2, \dots, c_r^2\}$. Hence, we have that $E_{2p-1} \triangle E_{2p} \supseteq E'_{2p-1} \cup E'_{2p}$ and $|E'_{2p-1} \cup E'_{2p}| = 4r+3 = \ell-4$, and $E_{2p} \triangle E_{2p+1} \supseteq E'_{2p} \cup E'_{2p+1}$ and $|E'_{2p} \cup E'_{2p+1}| = 4r + 3 = \ell - 4$.*

Lemma 1 *Let $\mathcal{P} = (P_1, \ldots, P_\tau)$ be a solution to the instance obtained using Constructions 4. Then $V(P_p) \cap V = V(P_q) \cap V$ for all $p, q \in \{1, \ldots, \tau\}$.*

Proof Assume towards a contradiction that there is $q = \phi(i, v, j)$ such that $V(P_{2q-1}) \cap V \neq V(P_{2q}) \cap V$ or $V(P_{2q}) \cap V \neq V(P_{2q+1}) \cap V$. We consider the first case (the second case is analogous). We know that each V_x is an s–t separator in (V, E_p) for every $x \in \{1, \ldots, r\}$ and $p \in \{1, \ldots, \tau\}$. Moreover, we know from Observations 3 that each of P_{2q-1} and P_{2q} contains exactly one vertex from each V_x, $x \in \{1, \ldots, r\}$. So, there is a $z \in \{1, \ldots, r\}$ such that there are distinct v' and v'' in V_z such that $v' \in V(P_{2q-1})$ and $v'' \in V(P_{2q})$. If $z \notin \{i, j\}$, then $\{v', a_z\}, \{v', b_z\}, \{v'', a_z\}, \{v'', b_z\} \in E_{2p-1} \triangle E_{2p}$. If $z = i$, then $\{v', b_z\}, \{v'', b_z\} \in E_{2p-1} \triangle E_{2p}$. If $z = j$, then $\{v', a_j\}, \{v', b_j\} \in E_{2p-1} \triangle E_{2p}$. Let $u \in V(P_{2q}) \cap V_i$ and let $w \in V(P_{2q}) \cap V_j$. By construction, we know that $\{u, a_j\}, \{w, a_i\}, \{u, a_i\}, \{w, b_j\} \in E_{2p-1} \triangle E_{2p}$. Hence, $E_{2p-1} \triangle E_{2p}$ contains $\ell - 4$ edges each being incident with a vertex in $C^1 \cup C^2$, and at least six further edges, amounting to $\ell + 2$ edges, contradicting the fact that $\mathcal{P}$ is a solution. □

Proof of Proposition 6 Let $I = (G = (V_1, \ldots, V_r, E)$ be an instance of Multicolored Clique with $|V_1| = \cdots = |V_r|$, and let $I' = (\mathcal{G} = (V', E_1, \ldots, E_\tau), s, t, k, \ell)$ be the instance obtained from I using Constructions 4 in polynomial time. Note that every edge in $\bigcup_{p=1}^{\tau} E_p$ is incident with $M{:} = A \cup B \cup C^1 \cup C^2 \cup \{s, t\}$, and hence M is a vertex cover of the underlying graph of size $|M| = 2r + 2r + 2 + 3 = 4r + 5$. Denote by $G_p = (V, E_p)$ the p-th snapshot of $\mathcal{G}$ for every $p \in \{1, \ldots, \tau\}$. We claim that I is a `yes`-instance if and only if I' is a `yes`-instance.

($\Rightarrow$) Let $W \subseteq V$ form a multicolored clique. Let P_{odd} be the path in $G_{\text{odd}}{:} = (V, E_{\text{odd}})$ with vertex set $V(P_{\text{odd}}) = A \cup B \cup C^1 \cup \{s, t\} \cup W$, and the edge set $E(P_{\text{odd}}) = E(G_{\text{odd}}[V(P_{\text{odd}})])$. Note that $|V(P_{\text{odd}})| = 3r+1+2+r = 4r+3 = k$. Set $P_{2p-1}{:} = P_{\text{odd}}$ for every $p \in \{1, \ldots, \tau/2\}$. Next we construct P_{2p} for every $p \in \{1, \ldots, \tau/2\}$. Let $p = \phi(i, v, j)$. We distinguish two cases whether $v \in W$ or not.

Case 1: $v \in W$. Let $V(P_{2p}) = A \cup B \setminus \{b_j\} \cup C^2 \cup \{s, t\} \cup W$, and $E(P_{2p}) = E(G_{2p}[V(P_{2p})])$. Note that $|V(P_{2p})| = 4r+2 \leq k$. Moreover, P_{2p} is an s–t path since the edges $\{v, a_j\}, \{a_j, w\}$ are contained in G_{2p}, where $w \in W \cap V_j$, since $\{v, w\} \in E$.

Case 2: $v \notin W$. Let $V(P_{2p}) = A \cup B \setminus \{a_j\} \cup C^2 \cup \{s, t\} \cup W$, and $E(P_{2p}) = E(G_{2p}[V(P_{2p})])$. Note that $|V(P_{2p})| = 4r+2 \leq k$. Moreover, P_{2p} is an s–t path since the edges $\{u, b_j\}, \{b_j, w\}$ are contained in G_{2p}, where $u \in W \cap V_i$ and $w \in W \cap V_j$ since b_j is adjacent to every vertex in $V_i \setminus \{v\}$ and V_j.

It remains to show that $|E(P_{2p-1}) \triangle E(P_{2p})| \leq \ell$ for all $p \in \{1, \ldots, \tau/2\}$, and that $|E(P_{2p}) \triangle E(P_{2p+1})| \leq \ell$ for all $p \in \{1, \ldots, \tau/2 - 1\}$. We prove the former, as the latter follows analogously. Let $p = \phi(i, v, j)$. By construction, $E(P_{2p-1}) \triangle E(P_{2p})$ contains all edges incident with C^1 and C^2. Let $u \in V_i \cap W$, and $w \in V_j \cap W$. We consider two cases:

Case 1: $u = v$. Note that P_{2p} has the subpath $b_i u a_j w a_i$, and hence we have that $E(P_{2p-1}) \triangle E(P_{2p})$ contains the edges $\{u, a_j\}, \{w, a_i\} \in E(P_{2p})$ and the edges $\{u, a_i\}, \{w, b_j\} \in E(P_{2p-1})$. Note that all other edges in $E(P_{2p-1}) \cup E(P_{2p})$ not incident to a vertex in $C^1 \cup C^2$ are also in $E(P_{2p-1}) \cap E(P_{2p})$. Hence, $|E(P_{2p-1}) \triangle E(P_{2p})| = 2(r + 1) + 2(r + 1) - 1 + 4 = 4r + 7 = \ell$.

Case 2: $u \neq v$. Note that P_{2p} has the subpath $b_i u b_j w a_i$, and hence we have that $E(P_{2p-1}) \triangle E(P_{2p})$ contains the edges $\{u, b_j\}, \{w, a_i\} \in E(P_{2p})$ and the edges $\{u, a_i\}, \{w, a_j\} \in E(P_{2p-1})$. Note that all other edges in $E(P_{2p-1}) \cup E(P_{2p})$ not incident to a vertex in $C^1 \cup C^2$ are also in $E(P_{2p-1}) \cap E(P_{2p})$. Hence, $|E(P_{2p-1}) \triangle E(P_{2p})| = 2(r+1) + 2(r+1) - 1 + 4 = 4r + 7 = \ell$.

It follows that $(P_1, \ldots, P_\tau)$ is a solution to I'.

($\Leftarrow$) Let $(P_1, \ldots, P_\tau)$ be a solution to I'. Due to Lemma 1, we know that $V(P_p) \cap V = V(P_q) \cap V =: W$ for all $p, q \in \{1, \ldots, \tau\}$. We claim that W forms a multicolored clique in G. By Observations 3, we know that $|W \cap V_i| = 1$, for all $i \in \{1, \ldots, r\}$. Let $w_i \in W \cap V_i$ denote the corresponding vertex, for all $i \in \{1, \ldots, r\}$. It remains to show that for each distinct pair w_i, w_j, we have that $\{w_i, w_j\} \in E$. Assume without loss of generality that $i < j$, and let $p = \phi(i, w_i, j)$. Since P_{2p} is an s–t path in G_{2p}, it contains the subpath $w_i a_j w_j$, since w_i is only adjacent to b_i and a_j. By construction of snapshot G_{2p}, we know that $\{a_j, w_j\} \in E(G_{2p})$ if and only if $\{w_i, w_j\} \in E$. Hence, the claim follows. □

For V$\triangle$V-MSTP, we have an even stronger result: the problem is W[1]-hard regarding $\nu_\downarrow$ even if the size of any symmetric difference of the vertex sets of consecutive paths is at most one. The proof is, however, similar to the proof of Proposition 6.

Proposition 7 V$\triangle$V-MSTP *when parameterized by* $\nu_\downarrow$ *is* W[1]*-hard, even if* $\ell = 1$.

Construction 5 Let $(G = (V_1, \ldots, V_r, E))$ be an instance of MULTICOLORED CLIQUE with $n = |V_1| = \cdots = |V_r|$ and let $N := n \cdot \binom{r}{2}$. We construct a temporal graph $\mathcal{G} = (V', E_1, \ldots, E_\tau)$ with $\tau = 2N$ as follows. Let V' initially contain $V_1, \ldots, V_r$ and s, t. Finally, V' contains the sets $A = \{a_0, \ldots, a_r\}$ and two special vertices x and y. We construct the edge set E_{odd} as follows. It contains the edges $\{s, a_0\}$ and $\{a_r, t\}$. Finally, it contains the edge set $\bigcup_{v \in V_i} \{\{a_{i-1}, v\}, \{a_i, v\}\}$ for every $i \in \{1, \ldots, r\}$. We set $E_i := E_{\text{odd}}$ for each odd $i \in \{1, \ldots, \tau\}$. Next, let ϕ be a bijection that maps (i, v, j) to $\{1, \ldots, N\}$, where $i < j$, $i, j \in \{1, \ldots, r\}$, $v \in V_i$. We construct the edge set $E_{2\phi(i,v,j)}$ as follows. We add the edge $\{s, a_i\}$. Then, a_i is connected with all $w \in V_i$. Next, v is adjacent with x, and all $w \in V_i \setminus \{v\}$ are adjacent with y. Next, x is adjacent to a vertex in $w \in V_j$ if and only if $\{w, v\} \in E$. Vertices y and a_j are adjacent with all vertices in V_j, and vertex a_j is also adjacent with $a_{\pi^{-1}(1)}$, where $\pi = \pi^r_{i,j} : \{1, \ldots, r\} \setminus \{i, j\} \to \{1, \ldots, r-2\}$ (see Definition 2). Then t is adjacent with a_0 which in turn is also adjacent with $a_{\pi^{-1}(r-2)}$, and for each $p \in \{1, \ldots, r-3\}$, $a_{\pi^{-1}(p)}$ and $a_{\pi^{-1}(p+1)}$ are adjacent to all vertices in $V_{\pi^{-1}(p)}$. This finishes the construction of $E_{2\phi(i,v,j)}$. Set $k = 2r + 4$ and $\ell = 1$.

Observation 5 *In* (V, E_{2p-1})*, each vertex in* A*, and each set* V_i *is an* s–t *separator, and in* (V, E_{2p}) *with* $p = \phi(i, v, j)$ *each vertex in* A*, each set* V_i*, and the set* $\{x, y\}$ *is an* s–t *separator.*

We know that each s–t path in an even snapshot contains s and t, and $r + 1$ vertices from A, and one of x and y, leaving r vertices. Since each V_i forms an s–t separator, we have the following.

Observation 6 *Every* s–t *path in* (V, E_p) *with at most* k *vertices contains exactly one vertex from each* V_i.

Proof of Proposition 7 Let $I = (G = (V_1, \dots, V_r, E))$ be an instance of MULTICOLORED CLIQUE with $|V_1| = \dots = |V_r|$, and let $I' = (\mathcal{G} = (V', E_1, \dots, E_\tau), s, t, k, \ell)$ be the instance obtained from I using Constructions 5 in polynomial time. Note that every edge in $\bigcup_{p=1}^{\tau} E_p$ is incident with $M := A \cup \{x, y\} \cup \{s, t\}$, and hence M is a vertex cover of the underlying graph of size $|M| = r + 5$. Denote by $G_p = (V, E_p)$ the p-th snapshot of $\mathcal{G}$ for every $p \in \{1, \dots, \tau\}$. We claim that I is a `yes`-instance if and only if I' is a `yes`-instance.

($\Rightarrow$) Let $W \subseteq V$ be a multicolored clique. Define P_{odd} as the path in $G_{\text{odd}} = (V, E_{\text{odd}})$ with vertex set $V(P_{\text{odd}}) = \{s, t\} \cup A \cup W$ and edge set $E(G_{\text{odd}}[V(P_{\text{odd}})])$. Note that P_{odd} is an s–t path with $2r + 3$ vertices. Set $P_{2p-1} := P_{\text{odd}}$. For P_{2p} with $p = \phi(i, v, j)$, we set

$$V(P_{2p}) = V(P_{\text{odd}}) \cup \begin{cases} \{x\}, & \text{if } v \in W \\ \{y\}, & \text{otherwise,} \end{cases} \quad \text{and} \quad E(P_{2p}) = E(G_{2p}[V(P_{2p})]).$$

Note that P_{2p} is an s–t path, since if $v \in W$, then the edge $\{x, w\}$ with $w \in W \cap V_j$ exists. Moreover, $|V(P_{2p})| = 2r + 4$, and by construction we have that $|V(P_p) \triangle V(P_{p+1})| = 1$ for all $p \in \{1, \dots, \tau - 1\}$.

($\Leftarrow$) Let $(P_1, \dots, P_\tau)$ be a solution to I'. Due to Observations 6, we know that each P_i contains exactly one vertex from V_i. In fact, it holds true that $V(P_i) \cap V = V(P_j) \cap V$ for all $i, j \in \{1, \dots, \tau\}$: Suppose not, that is, there is an $i \in \{1, \dots, \tau - 1\}$ such that $w \in V \cap (V(P_i) \triangle V(P_{i+1}))$. In both cases ($w \in V(P_i) \setminus V(P_{i+1})$ or $w \in V(P_{i+1}) \setminus V(P_i)$) we get a contradiction to Observations 6. Let $W := V \cap V(P_1)$. We claim that W is a multicolored clique in G. Let $v \in V_i \cap W$ and $w \in V_j \cap W$ with $i, j \in \{1, \dots, r\}$, $i < j$, be arbitrary but fixed. Then, path $P_{2\phi(i,v,j)}$ contains the subpath (v, x, w), proving that $\{v, w\} \in E$. It follows that W is a multicolored clique in G. □

We will see in the next section that similar hardness results as Theorems 3 and 4, Corollary 2 and Proposition 7 are unlikely for E∩E-MSTP or V∩V-MSTP.

5.2 Fixed-Parameter Tractability for Dissimilarity Variant Regarding *k*

In stark contrast to Theorems 3 and Corollary 2, we show in this section that V∩V-MSTP and E∩E-MSTP can be solved in linear time for constant path lengths; put differently, they are linear-time fixed-parameter tractable when parameterized by path length $k - 1$.

Theorem 5 V∩V-MSTP *and* E∩E-MSTP *can be solved in* $2^{O(k)} \cdot |\mathcal{G}|$ *time.*

We defer the proof of Theorems 5 towards the end of this section and, moreover, only describe the algorithm for V∩V-MSTP. In a nutshell, the algorithm behind Theorems 5 computes for each snapshot *sufficiently many* s–t paths such that no matter which vertices are used in the snapshots beforehand and afterwards, one of these s–t paths has a small intersection with these vertices. To this end, we introduce q-robust sets[2] of s–t paths.

Definition 3 Let $G = (V, E)$ be a graph, $s, t \in V$ two distinct vertices, $\mathcal{F}$ be a set of s–t paths of length at most $k - 1$, and $q \in N_0$. We call $\mathcal{F}$ *q-robust* if for each set $X \subseteq (V(G) \backslash \{s, t\})$ of size at most q the following holds: if there is an s–t path in $G - X$ of length at most $k - 1$, then there is an s–t path $P \in \mathcal{F}$ which is an s–t path in $G - X$.

To find a solution, it is sufficient to have a $2(k - \ell)$-robust set of s–t paths of length at most $k - 1$ for each snapshot of the temporal graph:

Lemma 2 *Let $I = (\mathcal{G} = (V, (E_i)_{i=1}^{\tau}), s, t, k, \ell)$ be an instance of* V∩V-MSTP *and $\mathcal{F}_i$ be a $2(k - \ell)$-robust set of s–t paths of length at most $k - 1$ in $G_i = (V, E_i)$, for all $i \in \{1, \ldots, \tau\}$. Then, I is a* `yes`*-instance if and only if there is a solution $(P_1, \ldots, P_\tau)$ such that $P_i \in \mathcal{F}_i$, for all $i \in \{1, \ldots, \tau\}$.*

Proof Since the converse is trivially true, we only show that if I is a `yes`-instance, then there is a solution $(P_1, \ldots, P_\tau)$ for I such that for all $i \in \{1, \ldots, \tau\}$ we have $P_i \in \mathcal{F}_i$.

For all $p \in \{1, \ldots, \tau + 1\}$, let $\mathcal{S}_p$ be the set which contains each solution $(P_1, \ldots, P_\tau)$ for I where for all $j < p$ we have $P_j \in \mathcal{F}_j$. Let $i := \max\{p \in \{1, \ldots, \tau + 1\} \mid \mathcal{S}_p \neq \emptyset\}$ and $(P_1, \ldots, P_\tau) \in \mathcal{S}_i$. If $i = \tau + 1$, then we are done. Hence, assume towards a contradiction that $i \leq \tau$.

(Case 1): Suppose $1 < i < \tau$. Let $X_1 = V(P_{i-1}) \backslash V(P_i)$ and $X_2 = V(P_{i+1}) \backslash V(P_i)$. If $X \in \{X_1, X_2\}$ is larger than $k - \ell$, then remove arbitrary vertices from X such that $|X| = k - \ell$. Note that $|V(P_{i-1}) \backslash X_1| \leq \ell$ and $|V(P_{i+1}) \backslash X_2| \leq \ell$. Observe that P_i is an s–t path of length at most $k - 1$ in $G_i - (X_1 \cup X_2)$. Since $\mathcal{F}_i$ is $2(k - \ell)$-robust, there is an s–t path $P \in \mathcal{F}_i$ of length at most $k - 1$ in $G_i - (X_1 \cup X_2)$, see Fig. 4 for an illustration. Hence, $|V(P) \cap V(P_{i-1})| \leq |V(P) \cap (V(P_{i-1}) \backslash X_1)| \leq \ell$ and $|V(P) \cap V(P_{i+1})| \leq |V(P) \cap (V(P_{i+1}) \backslash X_2)| \leq \ell$. Thus, $S = (P_1, \ldots, P_{i-1}, P, P_{i+1}, \ldots, P_\tau)$ is a solution for I. This contradicts i being maximal.

(Case 2): If $i = 1$ ($i = \tau$), then we set $X_1 = \emptyset$ ($X_2 = \emptyset$) and conclude analogously to Case 1 that i is not maximized.

□

The main tool of our algorithm is a fast ("linear-time FPT") computation of small sets of s–t paths of length at most $k - 1$ which are q-robust. We believe that such a use of representative families may become a general algorithmic tool being potentially helpful for other multistage problems. Formally, we show the following.

Lemma 3 *Let $G = (V, E)$ be a graph with two distinct vertices $s, t \in V$, and $k, q \in \mathbb{N}_0$. We can compute, in $2^{O(k+q)} \cdot |E|$ time, a q-robust set $\mathcal{F}$ of s–t paths of length at most $k - 1$ such that $|\mathcal{F}| \leq 2^{q+k}$.*

[2] Briefly put, q-robust sets are q-representative families [41], just explicitly coined to s–t paths of length at most k. This notion shall avoid confusion with the later defined q-representatives of independent sets.

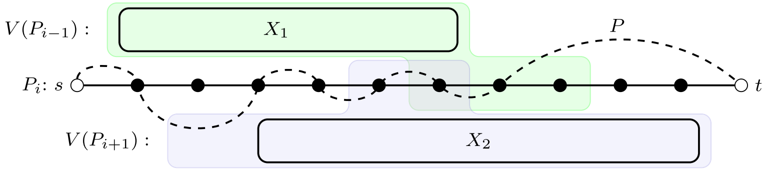

Fig. 4 Illustration of Case 1 in the proof of Lemma 2, where $|V(P_{i+1})\setminus V(P_i)| > k - \ell$

In order to prove Lemma 3, we extend the "representative-family-based" algorithm for k-Path of Fomin et al. [24] such that we can find s–t paths avoiding a size-at-most-q set of vertices. The proof of Lemma 3 is deferred to the end of this section.

We use standard terminology from matroid theory [42]. A pair $(U, \mathcal{I})$, where U is the *ground set* and $\mathcal{I} \subseteq 2^U$ is a family of *independent sets*, is a *matroid* if the following holds:

- $\emptyset \in \mathcal{I}$;
- if $A' \subseteq A$ and $A \in \mathcal{I}$, then $A' \in \mathcal{I}$;
- if $A, B \in \mathcal{I}$ and $|A| < |B|$, then there is an $x \in B\setminus A$ such that $A \cup \{x\} \in \mathcal{I}$.

An inclusion-wise maximal independent set $A \in \mathcal{I}$ of a matroid $M = (U, \mathcal{I})$ is a *basis*. The cardinality of the bases of M is called the *rank* of M. The *uniform matroid of rank* r on U is the matroid $(U, \mathcal{I})$ with $\mathcal{I} = \{S \subseteq U \mid |S| \le r\}$. A matroid (U, I) is *linear* or *representable over a field* $\mathbb{F}$ if there is a matrix A with entries in $\mathbb{F}$ and the columns labeled by the elements of U such that $S \in \mathcal{I}$ if and only if the columns of A with labels in S are linearly independent over $\mathbb{F}$. The matrix A is called a *representation* of (U, I).

Definition 4 (q-representative family of independent sets) Given a matroid $(U, \mathcal{I})$, a family $\mathcal{S} \subseteq \mathcal{I}$ of independent sets, we say that a subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is a *q-representative of* $\mathcal{S}$ if for each set $Y \subseteq U$ of size at most q it holds that if there is a set $X \in \mathcal{S}$ with $X \uplus Y \in \mathcal{I}$, then there is a set $\widehat{X} \in \widehat{\mathcal{S}}$ such that $\widehat{X} \uplus Y \in \mathcal{I}$.

We are only interested in uniform matroids, hence, to simplify matters we reformulate the definition of representative families.

Definition 5 (q-representative family) Let $\mathcal{S} = \{S_1, \dots, S_t\}$ be a family of sets of size p over a universe U. A subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is a q-representative of $\mathcal{S}$ if for every set $Y \subseteq U$ of size at most q it holds that if there is a set $X \in \mathcal{S}$ disjoint from Y, then there is a set $\widehat{X} \in \widehat{\mathcal{S}}$ disjoint from Y.

For linear matroids, there are fixed-parameter algorithms parameterized by rank that compute small representatives for large families of independent sets.

Lemma 4 (Fomin et al. [24, Theorem 1.1]) *Let $M = (U, \mathcal{I})$ be a linear matroid of rank $p + q$ given together with its representation matrix A_M over a field $\mathbb{F}$. Let $\mathcal{S} = \{S_1, \dots, S_t\}$ be a family of independents sets of M of size p. For a given q, a q-representative family $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ of size $\binom{p+q}{p}$ can be computed in $O\left(\binom{p+q}{p} t p^{\omega} + t \binom{p+q}{p}^{\omega-1}\right)$ time. Here, $\omega < 2.373$ is the matrix multiplication exponent.*

Lemma 5 *Given a set U and an integer r, we can compute in $O(r \cdot |U|)$ time a representation A of the uniform matroid of rank r on U, where $p \in O(|U|)$ and A is over a prime field $\mathbb{F}_p$.*

Proof A Vandermonde matrix of size $r \times |U|$ in a field with at least $|U|$ distinct elements suffices as representation of the uniform matroid of rank r on U [39, Section 3.4].

Let $p \in \{|U|, \dots, 2|U|\}$ be a prime number. Such a prime exists by the folklore Bertrand-Chebyshev theorem and can be computed in $O(|U|^{1/2+o(1)}) \le O(|U|)$ time using the Lagarias-Odlyzko method [44]. Observe that we can perform a primitive operation in the prime field $\mathbb{F}_p$ by first performing the operation in $\mathbb{Z}$ and then taking the result modulo p. Since we only need $O(\log |U|)$ many bits to store one element of $\mathbb{F}_p$, each element of $\mathbb{F}_p$ fits into one memory cell of the Word RAM computation model. Hence, we can perform a primitive operation over $\mathbb{F}_p$ in constant time.

Finally, we can compute the Vandermonde matrix of size $r \times |U|$ in $O(r \cdot |U|)$ time, because each entry is either 1 or an elementary element of $\mathbb{F}_p$ or can be computed by one multiplication from another entry calculated earlier. $\square$

In a nutshell, we extend the representative family based algorithm for k-Path of Fomin et al. [24] such that we find s–t paths which can avoid a set of vertices of size at most q.

Algorithm 1 Let $G = (V, E)$ be a graph with two distinct vertices $s, t \in V$, and $k, q \in \mathbb{N}_0$. Define $\mathcal{N}_v^i$ to be a $(q+k-i)$-representative of the family of all sets $A \subseteq V$ such that there is an s–v path P in G of length $i-1$ with $V(P) = A$.

Our goal is to compute $\mathcal{N}_t^k$, as we will construct the desired q-robust set of s–t paths from it later on. We start by setting $\mathcal{N}_s^1 := \{s\}$ and $\mathcal{N}_v^1 := \emptyset$ for all $v \in V' \setminus \{s\}$. Then, we compute for all $i \in \{2, \dots, k\}$ (in ascending order)

$$\mathcal{T}_v^i := \bigcup_{\{v,w\} \in E'} \; \bigcup_{X \in \mathcal{N}_w^{i-1} : v \notin X} (X \cup \{v\}). \tag{1}$$

Then (using Lemma 4) we compute a $(q+k-i)$-representative $\mathcal{N}_v^i$ of $\mathcal{T}_v^i$.

Lemma 6 *For all $i \in \{1, \dots, k\}$, the family $\mathcal{N}_v^i$ (from Algorithm 1) is of size at most $\binom{q+k-i}{i}$ and a $(q+k-i)$-representative of the family of all sets $A \subseteq V$ such that there is an s–v path P in G of length $i-1$ with $V(P) = A$.*

Proof We will prove this claim by induction. Observe that $\mathcal{N}_v^1$ is correctly computed for all $v \in V$. Now assume that for all $j < i \le k$ the family $\mathcal{N}_v^j$ is of size at most $\binom{q+k-j}{j}$ and $\mathcal{N}_v^j$ is a $(q+k-j)$-representative of the family of all sets $A \subseteq V$ such that there is an s–v path P in G of length $j-1$ with $V(P) = A$.

Let $Y \subseteq V'$ be a set of size at most $(q+k-i)$ and $v \in V$. Assume there is an s–v path P of length $i-1$ such that $Y \cap V(P) = \emptyset$. Let $w \in V(P)$ be the vertex which is visited by P directly before v (starting from s). Let P' be the s–w path of length $i-2$ induced by P without v. Since $(Y \cup \{v\}) \cap V(P') = \emptyset$ and $Y \cup \{v\}$ is a set of size $q+k-(i-1)$, we know, by induction hypothesis, that there is an $A \in \mathcal{N}_w^{i-1}$

and an s–w path P'' of length $i-2$ with $V(P'') = A$ and $(Y \cup \{v\}) \cap V(P'') = \emptyset$. Hence, by Algorithm 1, $V(P'') \cup \{v\} \in \mathcal{T}_v^i$. Since $Y \cap V(P'') = \emptyset$ and $\mathcal{N}_v^i$ is an $(q+k-i)$-representative of $\mathcal{T}_v^i$, we know that $\mathcal{N}_v^i$ contains a set B such that there is an s–v path P''' of length $i-1$ with $V(P''') = B$ and $B \cap Y = \emptyset$. Hence, $\mathcal{N}_v^j$ is indeed a $(q+k-i)$-representative of the family of all sets $A \subseteq V$ such that there is an s–v path P in G of length $i-1$ with $V(P) = A$.

The upper bound on the size of $\mathcal{N}_v^i$ follows from Lemma 4. This completes the proof. □

Lemma 7 *The family $\mathcal{N}_t^k$ from Algorithm 1 can be computed in $2^{O(q+k)} \cdot |E|$ time.*

Proof As a preprocessing step, we remove in $O(|E|)$ time via breadth-first search all vertices which are not on an s–t path. Hence, $|V| \leq |E|$. Furthermore, we use Lemma 5 to compute a representation of the uniform matroid M of rank $q+k$ on V in $O(|E| \cdot (q+k))$ time. Then, for each $i \in \{1, \dots, k\}$ and each $v \in V$ we compute $\mathcal{T}_v^i$ in $O(\deg(v) \cdot 2^{q+k})$ time, since for all $w \in V$ the family $\mathcal{N}_w^{i-1}$ is of size at most 2^{q+k}, see Lemma 4. Hence, $\mathcal{T}_v^i$ is of size at most $2^{q+k} \deg(v)$. Computing (with Lemma 4) the $(q+k-i)$-representative $\mathcal{N}_v^i$ of $\mathcal{T}_v^i$ takes $2^{O(k+q)} \cdot \deg(v)$ time. Hence, by the Handshaking Lemma, this yields an overall running time of $2^{O(k+q)} \cdot |E|$ time. □

In the proof of Lemma 7, one could use Theorem 1.2 instead of Theorem 1.1 from Fomin et al. [24] to improve the constant hidden in the Big-O notation. However, we would lose the linear dependency in $|E|$ by doing so.

We are now ready to prove Lemma 3.

Proof of Lemma 3 First, we construct the graph $G' = (V', E')$ where we add k new dummy vertices $d_1, \dots, d_k$ to G. Hence, $V' := V \cup \{d_1, \dots, d_k\}$ and

$$\begin{aligned} E' := E &\cup \{\{v, d_i\} \mid \{v, t\} \in E, i \in \{1, \dots, k\}\} \\ &\cup \{\{d_i, d_{i+1}\} \mid i \in \{1, \dots, k-1\}\} \\ &\cup \{\{d_i, t\} \mid i \in \{1, \dots, k\}\}. \end{aligned}$$

Note that for each s–t path P in G of length at most $k-1$ there is an s–t path P' in G' of length exactly $k-1$ such that $V(P) = V(P') \setminus \{d_1, \dots, d_k\}$. Furthermore, for each s–t path P' in G' of length exactly $k-1$ there is an s–t path P in G of length at most $k-1$ such that $V(P) = V(P') \setminus \{d_1, \dots, d_k\}$.

Using Algorithm 1, we compute in $2^{O(q+k)} \cdot |E|$ time (Lemma 7) $\mathcal{N}_t^{k+1}$ for G', s, t, k, and q. By Lemma 6, we know that $\mathcal{N}_t^k$ is of size at most $\binom{q+k}{k}$ and a q-representative of the family of all sets $A \subseteq V'$ such that there is an s–v path P in G of length $k-1$ with $V(P) = A$.

Now we compute the desired set $\mathcal{F}$, which we initialize by $\mathcal{F} := \emptyset$. Observe, that during the execution of Algorithm 1, we can store for each set $A \in \mathcal{T}_v^i$ a corresponding s–v path P in G with $V(P) = A$, where $i \in \{1, \dots, k\}$, $v \in V'$. We now go over all $A \in \mathcal{N}_t^k$ and their corresponding s–t paths P_A of length $k-1$ in G'. Next, we store in $\mathcal{F}$ an s–t path P' in G of length at most $k-1$ such that $V(P') = V(P_A) \setminus \{d_1, \dots, d_k\}$. The whole procedure ends after $2^{O(q+k)} \cdot |E|$ time and $\mathcal{F}$ is of size at most $|\mathcal{F}| \leq 2^{q+k}$.

It remains to show that $\mathcal{F}$ is q-robust. Let $X \subseteq V$ of size at most q such that there is an s–t path P of length at most $k-1$ in $G - X$. Hence, there is an s–t path P' in G' of length exactly k such that $V(P) = V(P') \setminus \{d_1, \dots, d_k\}$. Since $X \cap V(P') = \emptyset$, we know that there is an $A \in \mathcal{N}_t^k$ such that there is an s–v path P'' in G' of length k with $V(P'') = A$ and $A \cap X = \emptyset$. Thus, we added an s–t path P^* to $\mathcal{F}$ with $V(P^*) = A \setminus \{d_1, \dots, d_k\}$. Hence, $V(P^*) \cap X$ and it thus is an s–t path in $G - X$. □

Having Lemmas 2, and 3, we are set to prove Theorems 5.

Proof of Theorem 5 We only show the proof for V∩V-MSTP. The fixed-parameter tractability of E∩E-MSTP follows from Proposition 2.

Given an instance $I = (\mathcal{G} = (V, (E_i)_{i=1}^{\tau}), s, t, k, \ell)$ of V∩V-MSTP, we first check whether there is an empty E_i. If this is the case, then I is a `no`-instance. Afterwards, we can assume that $\tau \le |\mathcal{G}|$. For each $i \in \{1, \dots, \tau\}$, we compute in $2^{O(k+2(k-\ell))}|E_i| = 2^{O(k)}|E_i|$ time a $2(k-\ell)$-robust set $\mathcal{F}_i$ of s–t paths of length at most $k-1$ in $G_i = (V, E_i)$ such that $|\mathcal{F}_i| \le 2^{O(k)}$, see Lemma 3.

Next, we construct a directed graph $G' = (V', E')$, where beside s, t each path in $\mathcal{F}_i$ has a corresponding vertex, for all $i \in \{1, \dots, \tau\}$. Formally, that is, $V' := \{s, t\} \cup \bigcup_{i=1}^{\tau} \mathcal{F}_i$, and $E' := \{(P, P') \mid P \in \mathcal{F}_i, P' \in \mathcal{F}_{i+1}, |V(P) \cap V(P')| \le \ell,$ for some $i \in \{1, \dots, \tau-1\}\} \cup \{(s, P) \mid P \in \mathcal{F}_1\} \cup \{(P, t) \mid P \in \mathcal{F}_\tau\}$. Observe that $|V'| + |E'| \le 2^{O(k)} \cdot \tau$. Since $\sum_{i=1}^{\tau} |E_i| \le |\mathcal{G}|$, this yields an overall running time of $2^{O(k)} \cdot \max\{\tau, |\mathcal{G}|\} = 2^{O(k)} \cdot |\mathcal{G}|$.

It remains to show that I is a `yes`-instance if and only if there is an s–t path in G'. We only show that if I is a `yes`-instance, then there is an s–t path in G' since the converse is easy to verify from the definition of G'. Let I be a `yes`-instance. Then, by Lemma 2, there is a solution $(P_1, \dots, P_\tau)$ such that $P_i \in \mathcal{F}_i$, for all $i \in \{1, \dots, \tau\}$. For each $i \in \{1, \dots, \tau-1\}$, we have that $|V(P_i) \cap V(P_{i+1})| \le \ell$. It follows that G' has an edge from the vertex corresponding to P_i to the vertex corresponding to P_{i+1}. Hence, there is an s–t path in G' because s is adjacent to all vertices corresponding to a path in $\mathcal{F}_1$ and each vertex corresponding to a path in $\mathcal{F}_\tau$ is adjacent to t. □

6 Looking Through the Lens of Efficient Data Reduction

In this section, we study whether (polynomial) problem kernels for our four multistage s–t path problems exist. We start from the simple observation that every problem trivially admits a problem kernel of size polynomial in $n + \tau$. When strengthening n to $\nu_\downarrow$, that is, when parameterizing by $\nu_\downarrow + \tau$, where $\nu_\downarrow$ denotes the vertex cover number of the underlying graph, for E∩E-MSTP and V∩V-MSTP we prove a polynomial-size problem kernel (Sect. 6.1) and for E△E-MSTP and V△V-MSTP we prove a single-exponential-size problem kernel (Sect. 6.2). We prove that, unless NP ⊆ coNP / poly, the latter cannot be improved to polynomial size for V△V-MSTP and that, when parameterized by n (i.e., dropping τ from $n + \tau$), none of the four problems admits a polynomial kernel (Sect. 6.3).

6.1 Polynomial Kernel for the Dissimilarity Variant Regarding $\nu_\downarrow + \tau$

In this section, we prove V∩V-MSTP and E∩E-MSTP to admit problem kernels of polynomial size in $\nu_\downarrow + \tau$.

Theorem 6 *Each of* V∩V-MSTP *and* E∩E-MSTP *admits a problem kernel with at most* $\tau \cdot (2\nu_\downarrow + 2 + \binom{2\nu_\downarrow}{2}(3k-3)) \in O(\tau\nu_\downarrow^3)$ *vertices and* τ *snapshots.*

The kernelization behind Theorems 6 basically relies on the following data reduction rule.

Reduction Rule 1 Let $I = (\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau), s, t, k, \ell)$ be an instance of V∩V-MSTP or E∩E-MSTP with underlying graph $\mathcal{G}_\downarrow$.

1. Compute a vertex cover V' of $\mathcal{G}_\downarrow$ of size at most $2\nu_\downarrow$.
2. For each pair of distinct vertices $v, w \in V'$ and each $i \in \{1, \ldots, \tau\}$, in $N^i_{vw} := (N_{(V,E_i)}(v) \cap N_{(V,E_i)}(w)) \setminus V'$ mark $\min\{3k-3, |N^i_{vw}|\}$ vertices.
3. Construct a set V'' containing $\{s,t\} \cup V'$ and all marked vertices, and then construct the temporal graph $\mathcal{G}' = (V'', E'_1, \ldots, E'_\tau)$, where $E'_i = \{\{v, w\} \in E_i \mid v, w \in V''\}$, for all $i \in \{1, \ldots, \tau\}$.
4. Output the instance $O = (\mathcal{G}', s, t, k, \ell)$.

First, we prove that we can efficiently execute Reduction Rules 1.

Lemma 8 *Reduction Rules 1 is correct and can be executed in* $O(n\tau \cdot \nu_\downarrow^2)$ *time.*

Proof We can compute a 2-approximate vertex cover of the underlying graph in linear time via a maximal matching (Step 1). Next, we compute for each of the at most $\binom{2\nu_\downarrow}{2}$ pairs of vertices in V', in each of the τ snapshots, their neighborhood and mark a subset therein in linear time. Finally, we can compute the set V'', then $\mathcal{G}'$, and then O to output, each in linear time. Hence, this procedure ends after $O(n\tau \cdot \nu_\downarrow^2)$ time.

Let $I = (\mathcal{G} = (V, (E_i)_{i=1}^{\tau}), s, t, k, \ell)$ be an instance of V∩V-MSTP or E∩E-MSTP, and let $O = (\mathcal{G}', s, t, k, \ell)$ be the output instance of Reduction Rules 1 on I. Furthermore, for all $i \in \{1, \ldots, \tau\}$, let G_i and G'_i respectively denote the i-th snapshot of $\mathcal{G}$ and of $\mathcal{G}'$.

($\Leftarrow$) Since each path in a snapshot of $\mathcal{G}'$ is also a path in $\mathcal{G}$, we have that if O is a yes-instance, then I is a yes-instance as well.

($\Rightarrow$) Now let $(P_1, \ldots, P_\tau)$ be a solution for I. Clearly, if for each $i \in \{1, \ldots, \tau\}$ we have that P_i is a path in G'_i, then $(P_1, \ldots, P_\tau)$ is also a solution for O. For all $p \in \{1, \ldots, \tau\}$ let $\mathcal{S}_p$ be the set of solutions for I such that P_j is a path in G'_j, for all $j \in \{1, \ldots, p-1\}$. Note that if $\mathcal{S}_{\tau+1}$ is not empty, then O is clearly a yes-instance. Let $i = \max\{p \in \{1, \ldots, \tau\} \mid \mathcal{S}_p \neq \emptyset\}$ and let $S = (P_1, \ldots, P_\tau) \in \mathcal{S}_i$, $P_i = (v_0, v_1, \ldots, v_{k'})$, $s = v_0$, and $t = v_{k'}$ such that j is maximum under the condition that $v_0, \ldots, v_{j-1}$ is a path in G'_i. We can conclude that v_j is not a vertex in $\mathcal{G}'$. Let $V^* = V' \cup \{s, t\}$ where V' is the vertex cover we computed during the execution of Reduction Rules 1. Hence, $v_j \notin V^*$ but $v_{j-1}, v_{j+1} \in V^*$, otherwise V^* is not a vertex cover. Let $N = (N_{(V,E'_i)}(v_{j-1}) \cap N_{(V,E'_i)}(v_{j+1})) \setminus V'$. From Reduction Rules 1, we know that N is of size at least $3k-3$, as v_j is not in $\mathcal{G}'$. Now we distinguish into four cases:

1. If $1 = i = \tau$, then set $X = V(P_i)\setminus\{s, t, v_j\}$.
2. If $1 = i < \tau$, then set $X = (V(P_i) \cup V(P_{i+1}))\setminus\{s, t, v_j\}$.
3. If $1 < i = \tau$, then set $X = (V(P_{i-1}) \cup V(P_i))\setminus\{s, t, v_j\}$.
4. If $1 < i < \tau$, then set $X = (V(P_{i-1}) \cup V(P_i) \cup V(P_{i+1}))\setminus\{s, t, v_j\}$.

Since all paths in S are of length at most k, we know that X is of size at most $3k - 4$. Hence, there is a vertex $w \in N\setminus X$ such that $P' = (s = v_0, v_1, \ldots, v_{j-1}, w, v_{j+1}, \ldots, v_{k'} = t)$ is an s–t path in G_i' of length $k' \leq k$. Moreover, we note that

- if $i > 1$, then $|V(P_{i-1}) \cap V(P')| \leq |V(P_{i-1}) \cap V(P_i)|$ and $|E(P_{i-1}) \cap E(P')| \leq |E(P_{i-1}) \cap E(P_i)|$;
- if $i < \tau$, then $|V(P') \cap V(P_{i+1})| \leq |V(P_i) \cap V(P_{i+1})|$ and $|E(P') \cap E(P_{i+1})| \leq |E(P_i) \cap E(P_{i+1})|$.

Hence, in either case of I and O both being instances of V∩V-MSTP or E∩E-MSTP, $(P_1, \ldots, P_{i-1}, P, P_{i+1}, \ldots, P_\tau)$ is a solution for O. □

Proof of Theorem 6 Given an instance $I = (\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau), s, t, k, \ell)$, we apply Reduction Rules 1 in polynomial time to obtain the instance $O = (\mathcal{G}', s, t, k, \ell)$ being equivalent to I (Lemma 8), containing τ snapshots and at most $\tau \cdot (2\nu_\downarrow + 2 + \binom{2\nu_\downarrow}{2}(3k - 3))$ vertices. □

6.2 Single-Exponential Kernel for the Similarity Variant Regarding $\nu_\downarrow + \tau$

We prove that E△E-MSTP and V△V-MSTP admit problem kernels of single-exponential size in $\nu_\downarrow + \tau$, proving containment in FPT. As we will see later, unless NP $\subseteq$ coNP / poly this result for V△V-MSTP cannot be improved to size polynomial in $\nu_\downarrow + \tau$.

Theorem 7 *Each of* E△E-MSTP *and* V△V-MSTP *admits a problem kernel with at most* $2\nu_\downarrow + 4^{\nu_\downarrow \tau}(2\nu_\downarrow + 1)$ *vertices and* τ *snapshots.*

To prove Theorem 7, we lift the well-known graph-theoretic notion of (false) twins to temporal graphs as follows.

Definition 6 Two vertices v, w in a temporal graph $\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau)$ are called *(false) temporal twins* if $N_{(V,E_i)}(v) = N_{(V,E_i)}(w)$ for every $i \in \{1, \ldots, \tau\}$.

Note that Definition 6 implies an equivalence relation $\sim$ on the vertex set V, where $v \sim w$ if and only if they are temporal twins, and, hence, a partition of the vertex set into classes of temporal twins. Moreover, every pair of vertices in the same temporal twin class is non-adjacent. We show that such a partition is efficiently computable.

Lemma 9 *For a temporal graph* $\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau)$*, a partition* $V = (V_1, \ldots, V_p)$ *of* V *into temporal twin classes is computable in* $O(\tau \cdot |V|^2)$ *time.*

Proof Firstly, we compute all (false) twin classes in the first snapshot (V, E_1) in time linear in $|V| + |E_1|$. Next, for each vertex $v \in V$, check for each w with $v \sim w$ whether w is a false twin in each snapshot $(V, E_2), \ldots, (V, E_\tau)$, and adjust $\sim$ accordingly. □

In a nutshell, given a vertex cover X of our underlying graph, we aim for having few (i.e., upper-bounded by some single-exponential function in $\nu_\downarrow + \tau$) temporal twin classes in the independent set $Y = V \setminus X$, where each temporal twin class in turn contains only few vertices. By definition we have only few temporal twin classes.

Observation 7 *Let $\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau)$ be a temporal graph with partition $V = (X, Y)$ of V such that Y is an independent set in each snapshot. Then the size of every partition of Y into temporal twin classes is at most $2^{|X|\tau}$.*

Proof There are at most $2^{|X|}$ different neighborhoods for any vertex in Y per snapshot. As there are τ snapshots, there are at most $(2^{|X|})^\tau$ many temporal twin classes. □

We next aim for shrinking temporal twin classes. Note that for every temporal twin class with q neighbors, any s–t path contains at most $q-1$ vertices from the temporal twin class: recall that each temporal twin class forms an independent set, and hence every s–t path must "alternate" between the class and its neighboring vertices. In fact, temporal twin classes that are significantly larger than their neighborhood can be shrunk.

Reduction Rule 2 Let S be a temporal twin class with $|S \setminus \{s, t\}| \geq \max_{1 \leq i \leq \tau} |N_{(V,E_i)}(S)| + 2$. Then delete a vertex $v \in S \setminus \{s, t\}$.

Lemma 10 *Reduction Rules 2 is correct and exhaustively applicable in $O(\tau \cdot |V|^3)$ time.*

Proof The reduction is clearly applicable in $O(\tau \cdot |V|^3)$ time. We prove its correctness. To this end, let $\mathcal{G}$ and $\mathcal{G}'$ respectively denote the temporal graphs before and after application of Reduction Rules 2, and let $S' := S \setminus \{v, s, t\}$. Note that $|S'| \geq \max_{1 \leq i \leq \tau} |N_{(V,E_i)}(S')| - 1$. Moreover, observe that due to Lemma 9 we can exhaustively apply Reduction Rules 2 in polynomial time. We claim that $I = (\mathcal{G}, s, t, k, \ell)$ is a `yes`-instance if and only if $I' = (\mathcal{G}', s, t, k, \ell)$ is a `yes`-instance.

($\Leftarrow$) As $\mathcal{G}' = \mathcal{G} - v$, every sequence of s–t paths forming a solution for I' is also a solution to I.

($\Rightarrow$) Let I be a `yes`-instance, and assume that every solution to I contains the vertex v (otherwise we are done). Let $\mathcal{P} = (P_1, \ldots, P_\tau)$ be a solution to I such that v appears latest in the sequence among all solutions. Let P_{r_1} be the first s–t path that contains v, and let $r_1, \ldots, r_p$ be a maximal sequence such that $v \in V(P_{r_q})$ for each $1 \leq q \leq p$. Since $|S \setminus \{s, t\}| \geq \max_{1 \leq i \leq \tau} |N_{(V,E_i)}(S)| + 2$ and S forms an independent set, there is a vertex $w \in S'$ such that $w \notin V(P_r)$. We claim that "replacing" v by w in $P_{r_1}, \ldots, P_{r_p}$ forms a solution to I where v appears later than in $\mathcal{P}$, yielding a contradiction. Let $r_s > r_1$ denote the smallest index such that $w \in V(P_{r_s+1})$, or $r_s = r_p$ if no such index exists. For all $1 \leq q \leq s$, let P'_{r_q} be the s–t path with $V(P'_{r_q}) = (V(P_{r_q}) \setminus \{v\}) \cup \{w\}$ and $E(P'_{r_q}) = (E(P_{r_q}) \setminus \{\{v, u\} \mid u \in N_{P_{r_q}}(v)\}) \cup \{\{w, u\} \mid u \in N_{P_{r_q}}(v)\}$. For each $i \in \{1, \ldots, \tau\} \setminus \{r_1, \ldots, r_s\}$, we set $P'_i = P_i$. Observe that $|V(P'_{r_q})| = |V(P_{r_q})|$ and $|E(P'_{r_q})| = |E(P_{r_q})|$. Moreover, for all $1 \leq q < r_s$ we have that $|V(P'_{r_q}) \triangle V(P'_{r_{q+1}})| = |V(P_{r_q}) \triangle V(P_{r_{q+1}})|$ and $|E(P'_{r_q}) \triangle E(P'_{r_{q+1}})| = |E(P_{r_q}) \triangle E(P_{r_{q+1}})|$. If $r_1 > 1$, then it also holds true that $|V(P'_{r_1-1})\triangle V(P'_{r_1})| = |V(P_{r_1-1})\triangle V(P_{r_1})|$ and $|E(P'_{r_1-1})\triangle E(P'_{r_1})| = |E(P_{r_1-1})\triangle E(P_{r_1})|$. Finally, we consider the case of $r_s < \tau$, the cases herein whether or not $w \in V(P_{r_s+1})$.

Case 1: $w \notin V(P_{r_s+1})$, $r_s \leq r_q$. Then for the vertices we have that $V(P'_{r_s})\triangle V(P'_{r_s+1}) = ((V(P_{r_s})\triangle V(P_{r_s+1}))\setminus\{v\}) \cup \{w\}$. For the edges, we have that

$$E\big(P'_{r_s}\big)\triangle E\big(P'_{r_s+1}\big) = ((E(P_{r_s})\triangle E(P_{r_s+1}))\setminus\{\{v,u\} \mid u \in N_{P_{r_s+1}}(v)\}) \\ \cup \{\{w,u\} \mid u \in N_{P'_{r_s}}(w)\}.$$

Case 2: $w \in V(P_{r_s+1})$, $r_s < r_q$. Then for the vertices we have that $V(P'_{r_s})\triangle V(P'_{r_s+1}) = ((V(P_{r_s})\triangle V(P_{r_s+1}))\setminus\{w\}) \cup \{v\}$. For the edges, we have that

$$E\big(P'_{r_s}\big)\triangle E\big(P'_{r_s+1}\big) = (E(P_{r_s})\triangle E(P_{r_s+1})\setminus\{\{w,u\} \mid u \in N_{P_{r_s+1}}(w)\}) \\ \cup \{\{v,u\} \mid u \in N_{P_{r_s+1}}(v)\}.$$

Case 3: $w \in V(P_{r_s+1})$, $r_s = r_q$. Then for the vertices we have that $V(P'_{r_s})\triangle V(P'_{r_s+1}) = (V(P_{r_s})\triangle V(P_{r_s+1}))\setminus(\{w\} \cup \{v\})$. For the edges, we have that

$$E\big(P'_{r_s}\big)\triangle E\big(P'_{r_s+1}\big) = \big((E(P_{r_s})\triangle E(P_{r_s+1}))\setminus \\ (\{\{v,u\} \mid u \in N_{P_{r_s}}(v)\} \cup \{\{w,u\} \mid u \in N_{P_{r_s+1}}(w)\})\big) \\ \cup (\{\{w,u\} \mid u \in N_{P'_{r_s}}(w)\} \cup \{\{w,u\} \mid u \in N_{P_{r_s+1}}(w)\}).$$

Hence, in either case we have that the sizes of the symmetric differences both for vertex and edge sets are not increased. It follows that $\mathcal{P}' = (P'_1, \ldots, P'_\tau)$ is a solution in which v appears later than in $\mathcal{P}$, contradicting the choice of $\mathcal{P}$. □

Proof of Theorem 7 First, in $\mathcal{G}_\downarrow$ compute (via a maximal matching) a vertex cover X of size at most $2\nu_\downarrow$ in linear time. Let $V = (X, Y)$, where $Y = V\setminus X$ is an independent set. Next, compute all temporal twin classes of Y in polynomial time (Lemma 9). Apply Reduction Rules 2 exhaustively on every temporal twin class. Due to Lemma 10, this returns an equivalent instance in polynomial time where every temporal twin class contains at most $|X| + 1$ vertices. Due to Observations 7, there are at most $2^{|X|\tau}$ many temporal twin classes. In total, the obtained temporal graph contains at most $|X| + 2^{|X|\tau}(|X| + 1)$ vertices and τ snapshots. □

6.3 Lower Bounds on Kernelization Regarding n and $\nu_\downarrow + \tau$

We know that relaxing n to $\nu_\downarrow$ in $n + \tau$ allows for polynomial and single-exponential kernelization for dissimilarity and similarity, respectively. We know that dropping n is not possible (as to para-NP-hardness regarding τ, see Theorem 1). In this section, we prove that, unless NP $\subseteq$ coNP / poly, dropping τ is not possible either.

Theorem 8 *Unless* NP $\subseteq$ coNP / poly, *none of* E△E-MstP, V△V-MstP, E∩E-MstP, *and* V∩V-MstP *admits a problem kernel of size polynomial in n.*

Theorem 8 will follow from the forthcoming Propositions 8 and 9.

For proving that kernels of polynomial size are unlikely to exist, we use the cross-composition framework of Bodlaender et al. [7]. The framework, like the original framework [6, 25], bases upon the complexity-theoretic assumption that the polynomial time hierarchy does not collapse to its third level, which implies that NP $\nsubseteq$ coNP / poly [46]. The central notions of the framework are OR- and AND-cross-compositions, which require the notion of polynomial equivalence relations [7]: we call $\mathcal{R}$ a polynomial equivalence relation on Σ^* if we can decide in polynomial time whether any two $x, y \in \Sigma^*$ are $\mathcal{R}$-equivalent, and the number of equivalence classes in any finite set $S \subseteq \Sigma^*$ is in $(\max_{x \in S} |x|)^{O(1)}$.

Definition 7 ([7]) Given an NP-hard problem $L \subseteq \Sigma^*$, a parameterized problem $P \subseteq \Sigma^* \times \mathbb{N}$, and a polynomial equivalence relation $\mathcal{R}$ on the instances of L, an OR-cross-composition of L into P (with respect to $\mathcal{R}$) is an algorithm that takes p $\mathcal{R}$-equivalent instances $x_1 \dots, x_p$ of L and constructs in time $(\sum_{i=1}^{p} |x_i|)^{O(1)}$ an instance (x, k) of P such that (i) $k \in (\max_{1 \le i \le p} |x_i| + \log(p))^{O(1)}$ and (ii) $(x, k) \in P \iff x_i \in L$ for at least one $i \in \{1, \dots, p\}$.

An AND-cross-composition is an OR-cross-composition where (ii) is replaced by $(x, k) \in P \iff x_i \in L$ for all $i \in \{1, \dots, p\}$.

The connection is now the following: If a parameterized problem admits an OR-cross-composition (or AND-cross-composition) and a polynomial problem kernelization, then NP $\subseteq$ coNP / poly and the polynomial hierarchy collapses to its third level [7, 12].

We call two instances $I = (\mathcal{G}, s, t, k, \ell), I' = (\mathcal{G}', s', t', k', \ell')$ $\mathcal{R}$-equivalent if $|V(\mathcal{G})| = |V(\mathcal{G}')|$, $\tau(\mathcal{G}) = \tau(\mathcal{G}')$, $k = k'$, and $\ell = \ell'$.

Proposition 8 *There is an algorithm that given p $\mathcal{R}$-equivalent instances $I_1, \dots, I_p$ of* E△E-MstP, *computes in polynomial time an instance I of* E△E-MstP *such that the number of vertices of I is polynomial upper-bounded in the maximum number of vertices among $I_1 \dots, I_p$ and I is a* yes*-instance if and only if each of $I_1, \dots, I_p$ is a* yes*-instance.*

Construction 6 Let $I_1 = (\mathcal{G}_1 = (V, E_1^1, \dots E_\tau^1), s_1, t_1, k, \ell), \dots, I_p = (\mathcal{G}_1 = (V, E_1^p, \dots E_\tau^p), s_p, t_p, k, \ell)$ be p $\mathcal{R}$-equivalent instances of E△E-MstP (note that by relabeling, we can assume identical vertex sets). Let τ be the lifetime of $\mathcal{G}_i$, for all $i \in \{1, \dots, p\}$. We construct an instance $I = (\mathcal{G}', s, t, k', \ell)$ with $\mathcal{G}' = (V', E_1, \dots, E_{\tau'})$ and $k' = k+2$ and $\tau' = p(\tau + k')$ as follows. Let $V' = \{s, t\} \cup V$ with two new distinct vertices s and t. Let $E_{\text{trans}} = \{\{v, w\} \mid v, w \in V'\}$, that is, E_{trans} describes the edge set of a clique on V'. Next, let $\widehat{E}_r^q = E_r^q \cup \{\{s, s_q\}, \{t, t_q\}\}$ for every $r \in \{1, \dots, \tau\}$ and $q \in \{1, \dots, p\}$. For $1 \le q \le p$ and $1 \le j \le \tau + k'$, we set $E_{(q-1)(\tau+k')+j} = \widehat{E}_j^q$ if $j \le \tau$, and $E_{(q-1)(\tau+k')+j} = E_{\text{trans}}$ if $j > \tau$. This finishes the construction. Note that the construction is computable in polynomial time.

Observation 8 *Let G be a clique with two distinct vertices s, t, and let P, P' be two s–t paths each with at most $k \in \mathbb{N}$ vertices. Then there is a polynomial-time computable sequence $(P = P_1, \dots, P_k = P')$ of k s–t paths each with at most k vertices, such that $|E(P_i) \triangle E(P_{i+1})| \le 4$ for all $i \in \{1, \dots, k-1\}$.*

Proof Let $P = (s, v_1, \ldots, v_x, t)$ and $P' = (s, v'_1, \ldots, v'_{x'}, t)$. We consider two cases:

Case 1: $x \le x'$. Set $P_i = (s, v'_1, \ldots, v'_{i-1}, v_i, \ldots, v_x, t)$ for every $2 \le i \le x$. Note that $|E(P_i) \triangle E(P_{i+1})| \le 4$ as we switch two vertices yielding four edges. If $x = x'$, then $P_x = P'$. Otherwise, for $1 \le i \le x' - x$, let $P_{x+i} = (s, v'_1, \ldots, v'_x, v'_{x+1}, \ldots, v'_{x+i}, t)$. Note that $|E(P_{x+i}) \triangle E(P_{x+i+1})| \le 4$ as we replace the edge $\{v'_{x+i}, t\}$ by the edges $\{v'_{x+i}, v'_{x+i+1}\}$ and $\{v'_{x+i+1}, t\}$.

Case 2: $x > x'$. Set $P_i = (s, v'_1, \ldots, v'_{i-1}, v_i, \ldots, v_x, t)$ for every $2 \le i \le x'$. Note that $|E(P_i) \triangle E(P_{i+1})| \le 4$ as we switch two vertices yielding four edges. For $1 \le i \le x - x'$, let $P_{x+i} = (s, v'_1, \ldots, v'_{x'}, v_{x+1}, \ldots, v_{x-i}, t)$. Note that $|E(P_{x+i}) \triangle E(P_{x+i+1})| \le 4$ as we replace the edges $\{v_{x-i}, v_{x-i-1}\}$ and $\{v_{x-i}, t\}$ by the edge $\{v_{x-i-1}, t\}$.

Finally, if $r = \max\{x, x'\} < k$, then set $P_i = P_r$ for all $r < i \le k$ (note that since the paths are identical, their symmetric difference is zero). The sequence is computable in polynomial time. □

Proof of Proposition 8 Let $I_1 = (\mathcal{G}_1, s_1, t_1, k, \ell), \ldots, I_p = (\mathcal{G}_p, s_p, t_p, k, \ell)$ be p $\mathcal{R}$-equivalent instances of E△E-MstP with $\mathcal{G}_q = (V, E^q_1, \ldots E^q_\tau)$ for every $q \in \{1, \ldots, p\}$ and $\ell = 4$, and let $I = (\mathcal{G}', s, t, k', \ell)$ with $\mathcal{G}' = (V', E_1, \ldots, E_{\tau'})$ and $k' = k + 2$ be the instance obtained from $I_1, \ldots, I_p$ using Constructions 6. Note that $|V(\mathcal{G}')| = |V| + 2$. We claim that I is a `yes`-instance if and only if each of $I_1, \ldots, I_p$ is a `yes`-instance.

($\Rightarrow$) Let $(P_1, \ldots, P_{\tau'})$ be a solution to I. For $1 \le q \le p$ and $1 \le j \le \tau$, we define $P^q_j = P_{(q-1)(\tau+k')+j} - \{s, t\}$ as the path obtained from $P_{(q-1)(\tau+k')+j}$ when deleting s and t. with vertex set $V(P_{(q-1)(\tau+k')+j}) \setminus \{s, t\}$ and edge set $E(P_{(q-1)(\tau+k')+j}) \setminus \{\{s, s_q\}, \{t, t_q\}\}$. We claim that for each $1 \le q \le p$, $(P^q_1, \ldots, P^q_\tau)$ is a solution for I_q. First note that for every $j \in \{1, \ldots, \tau\}$, P^q_j is an s_q–t_q path in (V, E^q_j) and $|V(P^q_j)| = |V(P_{(q-1)(\tau+k')+1}) \setminus \{s, t\}| \le k' - 2 = k$. Moreover, for every $j \in \{1, \ldots, \tau - 1\}$, $|E(P^q_j) \triangle E(P^q_{j+1})| = |E(P_{(q-1)(\tau+k')+j}) \triangle E(P_{(q-1)(\tau+k')+j+1})| \le \ell$ (recall that s is only adjacent with s_q and t is only adjacent with t_q). Hence, the claim follows.

($\Leftarrow$) Let $(P^q_1, \ldots, P^q_\tau)$ be a solution for I_q for every $q \in \{1, \ldots, p\}$. For each $q \in \{1, \ldots, p\}$ and each $i \in \{1, \ldots, \tau\}$, let $\widehat{P}^q_i$ be the path obtained from P^q_i with $V(\widehat{P}^q_i) = V(P^q_i) \cup \{s, t\}$ and $E(\widehat{P}^q_i) = E(P^q_i) \cup \{\{s, s_q\}, \{t_q, t\}\}$. Note that $\widehat{P}^q_i$ is an s–t path and $|V(\widehat{P}^q_i)| = |V(P^q_i)| + 2 \le k'$, and we have that $|E(\widehat{P}^q_i) \triangle E(\widehat{P}^q_{i+1})| = |E(P^q_i) \triangle E(P^q_{i+1})| \le \ell$. Due to Observations 8, for each $q \in \{1, \ldots, p - 1\}$, we can compute for $\widehat{P}^q_\tau$ and $\widehat{P}^{q+1}_1$ a sequence $(\widehat{P}^q_\tau = P^{q,q+1}_1, \ldots, P^{q,q+1}_{k'} = \widehat{P}^{q+1}_1)$ of k' s–t paths such that each path has at most k' vertices and $|E(P^{q,q+1}_i) \triangle E(P^{q,q+1}_{i+1})| \le 4 = \ell$ for all $i \in \{1, \ldots, k' - 1\}$. Next we construct the path sequence $\mathcal{P} = (P_1, \ldots, P_{\tau'})$. For each $q \in \{1, \ldots, p\}$, we set $P_{(q-1)(\tau+k')+j} = \widehat{P}^q_j$ for $1 \le j \le \tau$, and we set $P_{(q-1)(\tau+k')+\tau+j} = P^{q,q+1}_j$ for $1 \le j \le k'$. Clearly, $|E(P_{(q-1)(\tau+k')+\tau}) \triangle E(P_{(q-1)(\tau+k')+\tau+1})| = |E(P_{(q-1)(\tau+k')+\tau+k'}) \triangle E(P_{q(\tau+k')+1})| = 0$ by construction for all $q \in \{1, \ldots, p\}$. It follows that for every $i \in \{1, \ldots, \tau'\}$, P_i is an s–t path with at most k' vertices, and for every $i \in \{1, \ldots, \tau' - 1\}$, it holds true that $|E(P_i) \triangle E(P_{i+1})| \le \ell$. Hence, $\mathcal{P}$ is a solution to I, and the claim follows. □

Proposition 9 *There is an algorithm that given p $\mathcal{R}$-equivalent instances $I_1, \ldots, I_p$ of* E∩E-MSTP, *computes in polynomial time an instance I of* E∩E-MSTP *such that $n \in (|V_1|)^{O(1)}$ and I is a* `yes`*-instance if and only if each of $I_1, \ldots, I_p$ is a* `yes`*-instance.*

Construction 7 Let $I_1 = (\mathcal{G}_1, s_1, t_1, k, \ell), \ldots, I_p = (\mathcal{G}_p, s_p, t_p, k, \ell)$ be p $\mathcal{R}$-equivalent instances of E∩E-MSTP with $\mathcal{G}_q = (V, E_1^q, \ldots E_\tau^q)$ for all $q \in \{1, \ldots, p\}$ and $\ell = 0$. We construct an instance $I = (\mathcal{G}', s, t, k', \ell)$ with $\mathcal{G}' = (V', E_1, \ldots, E_{\tau'})$ and $k' = k + 2$. Let $V' = \{s, t\} \cup V$ with two new distinct vertices s, t. Let $E_{\text{trans}} = \{\{s, t\}\}$, that is, E_{trans} only contains the edge s, t. Next, let $\widehat{E}_r^q = E_r^q \cup \{\{s, s_q\}, \{t, t_q\}\}$ for every $r \in \{1, \ldots, \tau\}$ and $q \in \{1, \ldots, p\}$. For $1 \le q \le p$ and $1 \le j \le \tau + 1$, we set $E_{(q-1)(\tau+1)+j} = \widehat{E}_j^q$ if $j \le \tau$, and $E_{(q-1)(\tau+1)+j} = E_{\text{trans}}$ if $j = \tau + 1$. This finishes the construction. Note that the construction runs in polynomial time.

Proof of Proposition 9 Let $I_1 = (\mathcal{G}_1, s_1, t_1, k, \ell), \ldots, I_p = (\mathcal{G}_p, s_p, t_p, k, \ell)$ be p $\mathcal{R}$-equivalent instances of E∩E-MSTP with $\mathcal{G}_q = (V, E_1^q, \ldots E_\tau^q)$ for every $q \in \{1, \ldots, p\}$ and $\ell = 0$, and let $I = (\mathcal{G}', s, t, k', \ell)$ with $\mathcal{G}' = (V', E_1, \ldots, E_{\tau'})$ and $k' = k + 2$ be the instance obtained from $I_1, \ldots, I_p$ using Constructions 7. Note that $|V(\mathcal{G}')| = |V| + 2$ We claim that I is `yes`-instance if and only if each of $I_1, \ldots, I_p$ is a `yes`-instance.

($\Rightarrow$) Let $(P_1, \ldots, P_{\tau'})$ be a solution to I. For $1 \le q \le p$ and $1 \le j \le \tau$, we define $P_j^q = P_{(q-1)(\tau+1)+j} - \{s, t\}$ as the path obtained from $P_{(q-1)(\tau+1)+j}$ when deleting s and t, which has vertex set $V(P_{(q-1)(\tau+1)+j}) \setminus \{s, t\}$ and edge set $E(P_{(q-1)(\tau+1)+j}) \setminus \{\{s, s_q\}, \{t, t_q\}\}$. We claim that for each $1 \le q \le p$, $(P_1^q, \ldots, P_\tau^q)$ is a solution for I_q. First note that for every $j \in \{1, \ldots, \tau\}$, P_j^q is an s_q–t_q path in (V, E_j^q) and $|V(P_j^q)| = |V(P_{(q-1)(\tau+k')+1}) \setminus \{s, t\}| \le k' - 2 = k$. Moreover, for every $j \in \{1, \ldots, \tau - 1\}$, $|E(P_j^q) \cap E(P_{j+1}^q)| = |E(P_{(q-1)(\tau+1)+j}) \cap E(P_{(q-1)(\tau+1)+j+1})| \le \ell$ (recall that s is only adjacent with s_q and t is only adjacent with t_q). Hence, the claim follows.

($\Leftarrow$) Let $(P_1^q, \ldots, P_\tau^q)$ be a solution for I_q for every $q \in \{1, \ldots, p\}$. For each $q \in \{1, \ldots, p\}$ and each $i \in \{1, \ldots, \tau\}$, let $\widehat{P}_i^q$ be the path obtained from P_i^q with $V(\widehat{P}_i^q) = V(P_i^q) \cup \{s, t\}$ and $E(\widehat{P}_i^q) = E(P_i^q) \cup \{\{s, s_q\}, \{t_q, t\}\}$. Note that $\widehat{P}_i^q$ is an s–t path and $|V(\widehat{P}_i^q)| = |P_i^q| + 2 \le k'$, and $|E(\widehat{P}_i^q) \cap E(\widehat{P}_{i+1}^q)| = |E(P_i^q) \cap E(P_{i+1}^q)| \le \ell$. Let $P = (s, t)$ be the s–t path with vertex set $V(P) = \{s, t\}$ and edge set $E(P) = \{\{s, t\}\}$. Next we construct the path sequence $\mathcal{P} = (P_1, \ldots, P_{\tau'})$. For each $q \in \{1, \ldots, p\}$, we set $P_{(q-1)(\tau+1)+j} = \widehat{P}_j^q$ for $1 \le j \le \tau$, and we set $P_{(q-1)(\tau+1)+\tau+1} = P$. Clearly, $|E(P_{(q-1)(\tau+1)+\tau}) \cap E(P_{(q-1)(\tau+1)+\tau+1})| = |E(P_{(q-1)(\tau+1)+\tau+1}) \cap E(P_{q(\tau+k')+1})| = 0$ by construction for every $q \in \{1, \ldots, p\}$, since P is the only path using only the edge $\{s, t\}$. It follows that for every $i \in \{1, \ldots, \tau'\}$, P_i is an s–t path with at most k' vertices, and for every $i \in \{1, \ldots, \tau' - 1\}$, it holds true that $|E(P_i) \cap E(P_{i+1})| \le \ell$. Hence, $\mathcal{P}$ is a solution to I, and the claim follows. □

While Theorem 8 is proven via an AND-cross-composition [7], we prove that V△V-MSTP admits no problem kernel of size polynomial in $\tau + \nu_\downarrow$ (unless NP $\subseteq$ coNP / poly) via an OR-cross-composition. Recall that $\nu_\downarrow$ denotes the vertex cover number of the underlying graph, and the result can be understood as that relaxing n in $n + \tau$ does not allow for efficient preprocessing.

We prove that, unless NP $\subseteq$ coNP / poly, improving the single-exponential kernel for V$\triangle$V-MSTP regarding $\nu_{\downarrow}+\tau$ to polynomial size is not possible.

Theorem 9 *Unless* NP $\subseteq$ coNP / poly, V$\triangle$V-MSTP *admits no problem kernel of size polynomial in* $\nu_{\downarrow}+\tau$.

To prove Theorem 9, we OR-cross-compose [7] from the following NP-complete [43] problem.

POSITIVE 1-IN-3 SAT
Input: A set X of variables and a set $\mathcal{C}$ of clauses each containing three positive literals over X.
Question: Is there $X' \subseteq X$ such that setting exactly the variables in X' to true results in each clause having exactly one variable set to true?

We call two instances $(X, \mathcal{C}), (X', \mathcal{C}')$ of POSITIVE 1-IN-3 SAT $\mathcal{R}$-equivalent if $|X| = |X'|$ and $|\mathcal{C}| = |\mathcal{C}'|$. Note that $\mathcal{R}$ defines a polynomial equivalence relation [7]. In particular, we show the following.

Proposition 10 *There is an algorithm that given a power p of two $\mathcal{R}$-equivalent instances $I_1 = (X_1, \mathcal{C}_1), \ldots, I_p = (X_p, \mathcal{C}_p)$ of* POSITIVE 1-IN-3 SAT, *computes in polynomial time an instance I of* V$\triangle$V-MSTP *such that $k+\tau+\nu_{\downarrow} \in (\max_{i\in\{1,\ldots,p\}} |X_i| + |\mathcal{C}_i| + \log(p))^{O(1)}$ and I is a* yes*-instance if and only if at least one of $I_1, \ldots, I_p$ is a* yes*-instance.*

We use the following Constructions 8 to show Proposition 10, see Fig. 5 for an illustration. The basic idea of the construction is that the temporal graph has, among other vertices, a vertex set $D = \bigcup_{q=1}^{p} D^q$, where D^q has one vertex for each variable in the q-th input instance. If we use a vertex from D^q in the s–t path, then we set the corresponding variable to true. In the first $\log(p)$ snapshots, we ensure that each s–t path can only use vertices from D which come from the same input instance. The remainder of the snapshots ensures that the clauses are satisfied. Here, the $(\log(p)+r)$-th snapshot ensures that the r-th clause of some input instance is satisfied with exactly one variable (vertex). Since we only use variables from one instance, Proposition 10 follows.

Construction 8 Let $I_1 = (X_1, \mathcal{C}_1), \ldots, I_p = (X_p, \mathcal{C}_p)$ be p, where p is a power of two, $\mathcal{R}$-equivalent instances of POSITIVE 1-IN-3 SAT where $N = |X_i|$ and $M = |\mathcal{C}_i|$ for all $i \in \{1, \ldots, p\}$. Let $D^q = \{v_i^q \mid i \in \{1, \ldots, N\}\}$ for all $q \in \{1, \ldots, p\}$, and $D = \bigcup_{q\in\{1,\ldots,p\}} D^q$. Let $A = \{a_0^i, a_1^i \mid i \in \{0, \ldots, N\}\}$ and $B = \{b_0^i, b_1^i \mid i \in \{0, \ldots, N\}\}$. Set $V = \{s, t\} \cup D \cup A \cup B$. Define for each $d \in \{0, 1\}$ the auxiliary function

$$h_d(i, r) := \begin{cases} a_d^i, & r \text{ odd} \\ b_d^i, & r \text{ even}. \end{cases}$$

We next describe the edge sets $E_1, \ldots, E_{\log(p)}$ and $E_{\log(p)+1}, \ldots, E_{\log(p)+M}$.

For edge set E_r with $r \leq \log(p)$, let E_r contain for each $d \in \{0, 1\}$

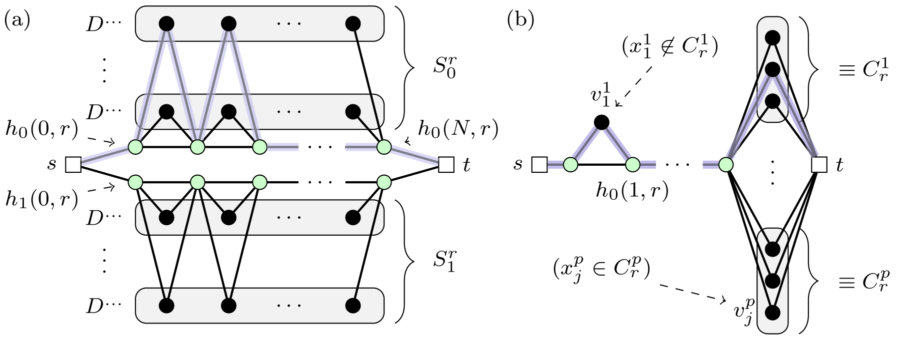

Fig. 5 Illustration of Constructions 8 with p input instances. **a** shows a snapshot (V, E_r) with $r \leq \log(p)$. **b** shows a snapshot $(V, E_{\log(p)+r})$ for the r-th clause of each input instance. Observe that the green (bright) vertices (including s, t) form a vertex cover of the underlying graph

– the edges $\{s, h_d(0, r)\}$, $\{t, h_d(N, r)\}$, and
– the edge set $\bigcup_{1 \leq i \leq N}\{\{h_d(i-1, r), h_d(i, r)\}\}$.

These sets form two s–t paths in (V, E_r). Finally, let S_0^r be the union of D^q with the r-th bit of the binary encoding of $q-1$ being 0, and S_1^r be the union of D^q with the r-th bit of the binary encoding of $q-1$ being 1. For $v_i^q \in S_0^r$, add the edges $\{h_0(i-1, r), v_i^q\}$ and $\{h_0(i, r), v_i^q\}$. Similarly, for $v_i^q \in S_1^r$, add the edges $\{h_1(i-1, r), v_i^q\}$ and $\{h_1(i, r), v_i^q\}$.

For edge set $E_{\log(p)+r}$ with $r \leq M$, let $E_{\log(p)+r}$ contain the edge $\{s, h_0(0, r)\}$ and the edge set $\bigcup_{1 \leq i \leq N}\{\{h_0(i-1, r), h_0(i, r)\}\}$. Consider the clauses $C_r^1, \dots, C_r^p$. For each C_r^q, if $x_i^q \in C_r^q$, then add the edges $\{h_0(N, r), v_i^q\}$, $\{v_i^q, t\}$, and if $x_i^q \notin C_r^q$, then add the edges $\{h_0(i-1, r), v_i^q\}$, $\{h_0(i, r), v_i^q\}$.

Set $k = 2N + 3$ and $\ell = 2(N + 1)$. This finishes the construction.

Observation 9 *If $(P_1, \dots, P_\tau)$ is a solution to I of Constructions 8, then for every $r \in \{1, \dots, \tau - 1\}$*

(i) $|V(P_r) \triangle V(P_{r+1})| = \ell$,
(ii) $V(P_r) \triangle V(P_{r+1}) \subseteq A \cup B$, *and*
(iii) $V(P_r) \cap D = V(P_{r'}) \cap D$ *for all* $r' \in \{1, \dots, \tau\}$.

Proof Let $r \in \{1, \dots, \tau - 1\}$. Note that in (V, E_r), $\{h_0(i, r), h_1(i, r)\}$ is an s–t separator for each $0 \leq i \leq N$. Hence, P_r must contain for each $0 \leq i \leq N$ a vertex from $\{h_0(i, r), h_1(i, r)\}$. The same holds for P_{r+1}: $\{h_0(i, r+1), h_1(i, r+1)\}$ is an s–t separator for each $0 \leq i \leq N$, and hence P_{r+1} must contain for each $0 \leq i \leq N$ a vertex from $\{h_0(i, r+1), h_1(i, r+1)\}$. Since $h_d(i, r) \neq h_{d'}(i', r+1)$ for all $i, i' \in \{0, \dots, N\}$ and $d, d' \in \{0, 1\}$, it follows that $|V(P_r) \triangle V(P_{r+1})| \geq 2(N+1) = \ell$. Since $(P_1, \dots, P_\tau)$ is a solution, it also holds true that $|V(P_r) \triangle V(P_{r+1})| \leq \ell$, and hence $V(P_r) \triangle V(P_{r+1}) \subseteq A \cup B$. This in turn implies that $D \cap V(P_r) \triangle V(P_{r+1}) = \emptyset$, and hence $V(P_r) \cap D = V(P_{r'}) \cap D$ for all $r' \in \{1, \dots, \tau\}$. □

Lemma 11 *If $(P_1, \dots, P_\tau)$ is a solution to I of Constructions 8, then for all $r \in \{1, \dots, \tau\}$ it holds true that $\emptyset \neq V(P_r) \cap D \subseteq D^q$ for some $q \in \{1, \dots, p\}$.*

Proof Observe that for each $r \in \{1, \dots, M\}$, we have that D is an s–t separator in the snapshot $(V, E_{\log(p)+r})$, and hence every s–t path must contain a vertex from D. Due to Observation 9, we know that $V(P_r) \cap D = V(P_{r'}) \cap D$ for all $r, r' \in \{1, \dots, \tau\}$. Suppose that each path from $P_1, \dots, P_\tau$ contains a vertex $v \in D^q$ and a vertex $v' \in D^{q'}$ for $q \neq q'$ in $V(P_r)$. Let $r \leq \log(p)$ be such that the r-th bit of q is d and of q' is $1-d$ with $d \in \{0, 1\}$ (that is, where their r-th bits differ). For $G_r = (V, E_r)$ it holds by construction that $G_r - \{s, t\}$ contains two connected components, one containing the vertex set $\bigcup_{i=0}^{N} h_d(i, r)$, and the other containing the vertex set $\bigcup_{i=0}^{N} h_{1-d}(i, r)$. Note that in G_r, $v \in D^q$ is only connected to two vertices from $\bigcup_{i=0}^{N} h_d(i, r)$, and $v' \in D^{q'}$ is only connected to two vertices from $\bigcup_{i=0}^{N} h_{1-d}(i, r)$. Hence, $P_r - \{s, t\}$ contains vertices from two connected components, contradicting the fact that P_r is an s–t path in G_r. $\square$

Proof of Proposition 10 Let $I_1 = (X_1, \mathcal{C}_1), \dots, I_p = (X_p, \mathcal{C}_p)$ be p, p being a power of two, $\mathcal{R}$-equivalent instances of Positive 1-in-3 SAT where $N = |X|$ and $M = |\mathcal{C}|$. Let $I = (\mathcal{G} = (V, E_1, E_2, \dots, E_\tau), s, t, k, \ell)$ be the instance obtained by Constructions 8 from $I_1, \dots, I_p$. Observe that $A \cup B \cup \{s, t\}$ is a vertex cover of the underlying graph of $\mathcal{G}$. Hence, we have that $k + \tau + \nu_\downarrow \leq 2N + 3 + \log(p) + M + N + 4$.

We claim that I is a `yes`-instance if and only if at least one of $I_1, \dots, I_p$ is a `yes`-instance.

($\Leftarrow$) Let $X \subseteq X_q$ be a solution to I_q, for some $q \in \{1, \dots, p\}$.

We construct a solution $(P_1, \dots, P_\tau)$ to I as follows. Set for each $r \in \{1, \dots, \log(p)\}$, where $d = 0$ if the r-th bit of $q - 1$ is 0, and 1 otherwise,

$$
\begin{aligned}
V(P_r) &= \bigcup_{x_i^q \in X} \{v_i^q\} \cup \{s, t\} \cup \bigcup_{0 \leq i \leq N} h_d(i, r)\\
E(P_r) &= \{\{s, h_d(0, r)\}\} \cup \{\{t, h_d(N, r)\}\}\\
&\quad \cup \bigcup_{x_i^q \in X} \{\{h_d(i-1, r), v_i^q\}, \{h_d(i, r), v_i^q\}\}\\
&\quad \cup \bigcup_{x_i^q \in X_q \setminus X} \{\{h_d(i-1, r), h_d(i, r)\}\}.
\end{aligned}
$$

Moreover, for each $r \in \{1, \dots, M\}$ set, where $x_j^q \in X^q \cap C_r^q$,

$$
\begin{aligned}
V(P_{\log(p)+r}) &= \bigcup_{x_i^q \in X} \{v_i^q\} \cup \{s, t\} \cup \bigcup_{0 \leq i \leq N} h_0(i, r),\\
E(P_{\log(p)+r}) &= \{\{s, h_0(0, r)\}, \{h_0(j-1, r), h_0(j, r)\}, \{h_0(N, r), v_j^q\}, \{t, v_j^q\}\}\\
&\quad \cup \bigcup_{x_i^q \in X \setminus \{x_j^q\}} \{\{h_0(i-1, r), v_i^q\}, \{h_0(i, r), v_i^q\}\}\\
&\quad \cup \bigcup_{x_i^q \in X_q \setminus X} \{\{h_0(i-1, r), h_0(i, r)\}\}.
\end{aligned}
$$

First observe that $|V(P_r)| \leq N+2+N+1$, for all $r \in \{1, \ldots, \tau\}$. Second, observe that $|V(P_r) \triangle V(P_{r+1})| = \ell$, for all $r \in \{1, \ldots, \tau-1\}$. Finally, we claim that P_r is an s–t path in (V, E_r) for each $r \in \{1, \ldots, \tau\}$. For P_r with $r \leq \log(p)$, this follows by construction. Consider $P_{\log(p)+r}$ with $1 \leq r \leq M$. Note that X contains exactly one x_j^q with $x_j^q \in C_r^q$ ($j \in \{1, \ldots, N\}$) and hence the subpath $(h_0(N, r), v_i^q, t)$ of $P_{\log(p)+r}$ exists in $(V, E_{\log(p)+r})$. By construction the subpath of $P_{\log(p)+r}$ from s to $h_0(N, r)$ also exists.

($\Rightarrow$) Let $(P_1, \ldots, P_\tau)$ be a solution to I. Due to Lemma 11, we know that for all $r \in \{1, \ldots, \tau\}$ it holds true that $\emptyset \neq V(P_r) \cap D \subseteq D^q$ for some $q \in \{1, \ldots, p\}$. Let $X = \{x_i^q \mid v_i^q \in V(P_1)\}$. We claim that X is a solution to I_q, that is, for every clause C_r^q there is an $x \in X$ with $x \in C_r^q$. Consider the snapshot $G_{\log(p)+r} = (V, E_{\log(p)+r})$. Since $P_{\log(p)+r}$ is an s–t path in $G_{\log(p)+r}$ and D is an s–t separator in $G_{\log(p)+r}$, there is exactly one $v \in D$ such that subpath $(h_0(N, r), v, t)$ is a subpath of $P_{\log(p)+r}$. We know that $v \in D^q$, and hence there is an $x \in X$ such that $x \in C_r^q$. $\square$

Proposition 10 describes an OR-cross-composition from an NP-hard problem to V$\triangle$V-MSTP parameterized by $\nu_\downarrow + \tau$, and hence Theorems 9 follows [7]. We leave open whether E$\triangle$E-MSTP allows for a problem kernel of size polynomial in $\nu_\downarrow + \tau$.

7 Conclusion

On the one extreme, our hardness results exploit that the temporal graph can change dramatically from one time step to another. On the other extreme, the NP-hard (and typically parameterized hard) LENGTH-BOUNDED DISJOINT PATH problem [28] easily reduces to all four MSTP variants with each snapshot having the same edge set. This leads to the natural question for further islands of computational tractability between these two extremes. Moreover, for the similarity case, we leave open whether working with edge distances decisively differs from working with vertex distances.

The models we introduced (and future, more refined models based upon these) may find several applications as they naturally capture time-dependent route-querying tasks. Herein, additionally considering edge-lengths may be necessary. Besides resolving questions we explicitly stated as open throughout the text, future work could address generalizing the "consecutiveness" property by requiring that also short sequences (as in the time-window model of temporal graphs [37, 38]) of consecutive paths are (pairwise) similar or dissimilar. Furthermore, with introducing the "dissimilarity view" we entered new territory in the context of multistage problems; it seems natural to also study it for other problems beyond s–t PATH. Finally, to analyze s–t PATH in the *global multistage*[3] setting is well-motivated as well [31].

Acknowledgements We thank the referees for their careful reading and constructive comments.

Funding Open Access funding enabled and organized by Projekt DEAL.

[3] That is, the total sum over all differences between consecutive paths in the solution is upper-bounded.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Bampis, E., Escoffier, B., Lampis, M., Paschos, V.T.: Multistage matchings. In: Proceedings of the 16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018), vol. 101 of LIPIcs, pp. 7:1–7:13. Schloss Dagstuhl—Leibniz-Zentrum für Informatik (2018)
2. Bampis, E., Escoffier, B., Schewior, K., Teiller, A.: Online multistage subset maximization problems. In: Proceedings of the 27th the Annual European Symposium on Algorithms (ESA 2020), vol. 144 of LIPIcs, pp .11:1–11:14. Schloss Dagstuhl—Leibniz-Zentrum für Informatik (2019a)
3. Bampis, E., Escoffier, B., Teiller, A.: Multistage knapsack. In: Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019), vol. 138 of LIPIcs, pp. 22:1–22:14. Schloss Dagstuhl—Leibniz-Zentrum für Informatik (2019b)
4. Bampis, E., Escoffier, B., Kononov, A.V.: LP-based algorithms for multistage minimization problems. In: Proceedings of the 18th International Workshop on Approximation and Online Algorithms (WAOA 2020), vol. 12806 of LNCS, pp. 1–15. Springer (2020). https://doi.org/10.1007/978-3-030-80879-2_1
5. Bentert, M., Himmel, A.-S., Nichterlein, A., Niedermeier, R.: Efficient computation of optimal temporal walks under waiting-time constraints. Appl. Netw. Sci. **5**, 1–26 (2020)
6. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. J. Comput. Syst. Sci. **75**(8), 423–434 (2009)
7. Bodlaender, H.L., Jansen, B.M.P., Kratsch, S.: Kernelization lower bounds by cross-composition. SIAM J. Discrete Math. **28**(1), 277–305 (2014)
8. Bredereck, R., Fluschnik, T., Kaczmarczyk, A.: When votes change and committees should (not). In: Proceedings of the 31th International Joint Conference on Artificial Intelligence (IJCAI 2022), pp. 144–150. ijcai.org (2022). https://doi.org/10.24963/ijcai.2022/21
9. Buß, S., Molter, H., Niedermeier, R., Rymar, M.: Algorithmic aspects of temporal betweenness. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2020), pp. 2084–2092 (2020)
10. Casteigts, A., Himmel, A.-S., Molter, H., Zschoche, P.: Finding temporal paths under waiting time constraints. Algorithmica **83**(9), 2754–2802 (2021). https://doi.org/10.1007/s00453-021-00831-w
11. Charikar, M., Chekuri, C., Feder, T., Motwani, R.: Incremental clustering and dynamic information retrieval. SIAM J. Comput. **33**(6), 1417–1440 (2004). https://doi.org/10.1137/S0097539702418498
12. Drucker, A.: New limits to classical and quantum instance compression. SIAM J. Comput. **44**(5), 1443–1479 (2015)
13. Duffin, R.J.: Topology of series–parallel networks. J. Math. Anal. Appl. **10**(2), 303–318 (1965)
14. Eisenstat, D., Mathieu, C., Schabanel, N.: Facility location in evolving metrics. In: Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP 2014), vol. 8572 of LNCS, pp. 459–470. Springer (2014)
15. Enright, J., Meeks, K.: Deleting edges to restrict the size of an epidemic: a new application for treewidth. Algorithmica **80**(6), 1857–1889 (2018)

16. Enright, J., Meeks, K., Mertzios, G.B., Zamaraev, V.: Deleting edges to restrict the size of an epidemic in temporal networks. J. Comput. Syst. Sci. **119**, 60–77 (2021). https://doi.org/10.1016/j.jcss.2021.01.007
17. Erlebach, T., Spooner, J.T.: Faster exploration of degree-bounded temporal graphs. In: Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018), vol. 117 of LIPIcs, pp. 36:1–36:13. Schloss Dagstuhl—Leibniz-Zentrum für Informatik (2018)
18. Erlebach, T., Kammer, F., Luo, K., Sajenko, A., Spooner, J.T.: Two moves per time step make a difference. In: Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019), vol. 132 of LIPIcs, pp. 141:1–141:14. Schloss Dagstuhl—Leibniz-Zentrum für Informatik (2019)
19. Fluschnik, T.: A multistage view on 2-satisfiability. In: Proceedings of the 12th International Conference on Algorithms and Complexity (CIAC 2021), vol. 12701 of LNCS, pp. 231–244. Springer (2021). https://doi.org/10.1007/978-3-030-75242-2_16
20. Fluschnik, T., Kunz, P.: Bipartite temporal graphs and the parameterized complexity of multistage 2-coloring. In: Proceedings of the 1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2022), vol. 221 of LIPIcs, pp. 16:1–16:18. Schloss Dagstuhl—Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPIcs.SAND.2022.16
21. Fluschnik, T., Kratsch, S., Niedermeier, R., Sorge, M.: The parameterized complexity of the minimum shared edges problem. J. Comput. Syst. Sci. **106**, 23–48 (2019)
22. Fluschnik, T., Morik, M., Sorge, M.: The complexity of routing with collision avoidance. J. Comput. Syst. Sci. **102**, 69–86 (2019)
23. Fluschnik, T., Niedermeier, R., Rohm, V., Zschoche, P.: Multistage vertex cover. Theory Comput. Syst. **66**(2), 454–483 (2022). https://doi.org/10.1007/s00224-022-10069-w
24. Fomin, F.V., Lokshtanov, D., Panolan, F., Saurabh, S.: Efficient computation of representative families with applications in parameterized and exact algorithms. J. ACM **63**(4), 29:1-29:60 (2016)
25. Fortnow, L., Santhanam, R.: Infeasibility of instance compression and succinct PCPs for NP. J. Comput. Syst. Sci. **77**(1), 91–106 (2011)
26. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
27. Ghariblou, S., Salehi, M., Magnani, M., Jalili, M.: Shortest paths in multiplex networks. Nat. Sci. Rep. **7**, 2142 (2017)
28. Golovach, P.A., Thilikos, D.M.: Paths of bounded length and their cuts: parameterized complexity and algorithms. Discrete Optim. **8**(1), 72–86 (2011)
29. Gupta, A., Talwar, K., Wieder, U.: Changing bases: multistage optimization for matroids and matchings. In: Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP 2014), vol. 8572 of LNCS, pp. 563–575. Springer (2014)
30. Hartung, S., Niedermeier, R.: Incremental list coloring of graphs, parameterized by conservation. Theor. Comput. Sci. **494**, 86–98 (2013)
31. Heeger, K., Himmel, A.-S., Kammer, F., Niedermeier, R., Renken, M., Sajenko, A.: Multistage graph problems on a global budget. Theor. Comput. Sci. **868**, 46–64 (2021). https://doi.org/10.1016/j.tcs.2021.04.002
32. Holme, P., Saramäki, J. (eds.): Temporal Networks. Springer, Berlin (2013)
33. Holme, P., Saramäki, J. (eds.): Temporal Network Theory. Springer, Berlin (2019)
34. Kellerhals, L., Renken, M., Zschoche, P.: Parameterized algorithms for diverse multistage problems. In: Proceedings of the 29th Annual European Symposium on Algorithms (ESA 2021), vol. 204, pp. 55:1–55:17. Schloss Dagstuhl—Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.ESA.2021.55
35. Kempe, D., Kleinberg, J.M., Kumar, A.: Connectivity and inference problems for temporal networks. J. Comput. Syst. Sci. **64**(4), 820–842 (2002)
36. Klobas, N., Mertzios, G.B., Molter, H., Niedermeier, R., Zschoche, P.: Interference-free walks in time: temporally disjoint paths. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021), pp. 4090–4096. ijcai.org (2021). https://doi.org/10.24963/ijcai.2021/563
37. Latapy, M., Viard, T., Magnien, C.: Stream graphs and link streams for the modeling of interactions over time. Soc. Netw. Anal. Min. **8**(1), 61:1-61:29 (2018)
38. Latapy, M., Fiore, M., Ziviani, A.: Link streams: methods and applications. Comput. Netw. **150**, 263–265 (2019)

39. Marx, D.: A parameterized view on matroid optimization problems. Theor. Comput. Sci. **410**(44), 4471–4479 (2009)
40. Michail, O.: An introduction to temporal graphs: an algorithmic perspective. Internet Math. **12**(4), 239–280 (2016)
41. Monien, B.: How to find long paths efficiently. Discrete Math. **25**, 239–254 (1985)
42. Oxley, J.G.: Matroid Theory. Oxford University Press, Oxford (1992)
43. Schaefer, T.J.: The complexity of satisfiability problems. In: Proceedings of the 10th ACM Symposium on Theory of Computing (STOC 1978), pp. 216–226 (1978)
44. Tao, T., Croot, E., III., Helfgott, H.: Deterministic methods to find primes. Math. Comput. **81**(278), 1233–1246 (2012)
45. Wu, H., Cheng, J., Ke, Y., Huang, S., Huang, Y., Hejun, W.: Efficient algorithms for temporal path computation. IEEE Trans. Knowl. Data Eng. **28**(11), 2927–2942 (2016)
46. Yap, C.-K.: Some consequences of non-uniform conditions on uniform classes. Theor. Comput. Sci. **26**, 287–300 (1983)
47. Zschoche, P.: Restless temporal path parameterized above lower bounds. CoRR (2022). https://doi.org/10.48550/arXiv.2203.15862
48. Zschoche, P., Fluschnik, T., Molter, H., Niedermeier, R.: The complexity of finding small separators in temporal graphs. J. Comput. Syst. Sci. **107**, 72–92 (2020)