Distance to Transitivity: New Parameters for Taming Reachability in Temporal Graphs

Arnaud Casteigts
Department of Computer Science, University of Geneva, Switzerland

Nils Morawietz
Friedrich Schiller University Jena, Institute of Computer Science, Germany

Petra Wolf
LaBRI, CNRS, Université de Bordeaux, Bordeaux INP, France

Abstract

A temporal graph is a graph whose edges only appear at certain points in time. Reachability in these graphs is defined in terms of paths that traverse the edges in chronological order (temporal paths). This form of reachability is neither symmetric nor transitive, the latter having important consequences on the computational complexity of even basic questions, such as computing temporal connected components. In this paper, we introduce several parameters that capture how far a temporal graph $\mathcal{G}$ is from being transitive, namely, *vertex-deletion distance to transitivity* and *arc-modification distance to transitivity*, both being applied to the reachability graph of $\mathcal{G}$. We illustrate the impact of these parameters on the temporal connected component problem, obtaining several tractability results in terms of fixed-parameter tractability and polynomial kernels. Significantly, these results are obtained without restrictions of the underlying graph, the snapshots, or the lifetime of the input graph. As such, our results isolate the impact of non-transitivity and confirm the key role that it plays in the hardness of temporal graph problems.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Mathematics of computing → Discrete mathematics

Keywords and phrases Temporal graphs, Parameterized complexity, Reachability, Transitivity.

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.28

Funding *Arnaud Casteigts*: supported by French ANR, project ANR-22-CE48-0001 (TEMPOGRAL).
Petra Wolf: supported by French ANR, project ANR-22-CE48-0001 (TEMPOGRAL).

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 28; pp. 28:1–28:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Temporal graphs have gained attention lately as appropriate tools to capture time-dependent phenomena in fields as various as transportation, social networks analysis, biology, robotics, scheduling, and distributed computing. On the theoretical side, these graphs generate interest mostly for their intriguing features. Indeed, many basic questions are still open, with a general feeling that existing techniques from graph theory typically fail on temporal graphs. In fact, most of the natural questions considered in static graphs turn out to be intractable when formulated in a temporal version, and likewise, most of the temporal analogs of classical structural properties are false.

One of the earliest examples is that the natural analog of Menger's theorem does not hold in temporal graphs [21]. Another early result is that deciding if a temporal connected component (set of vertices that can reach each other through temporal paths) of a certain size exists is NP-complete [6]. A more recent and striking result is that there exist temporally connected graphs on $\Theta(n^2)$ edges in which every edge is critical for connectivity; in other words, no temporal analog of sparse spanners exist unconditionally [5] (though they do, probabilistically [11]). Moreover, minimizing the size of such spanners is APX-hard [2, 5]. Further hardness results for problems whose static versions are generally tractable include separators [19], connectivity mitigation [16], exploration [4, 17], flows [1], Eulerian paths [7], and even spanning trees [9].

Faced by these difficulties, the algorithmic community has focused on special cases, and tools from parameterized complexity were employed with moderate success. A natural approach here is to apply the range of classical graph parameters to restrict either the underlying graph of the temporal graph (i.e. which edges can exist at all) or its snapshots (i.e. which edges may exist simultaneously). For example, finding temporal paths with bounded waiting time at each node (which is NP-hard in general) turns out to be FPT when parameterized by treedepth or vertex cover number of the underlying graph. But the problem is already W[1]-hard for pathwidth (let alone treewidth) [10]. In fact, as observed in [18], most temporal graphs problems remain hard even when the underlying graph has bounded treewidth (sometimes, even a tree or a star [3, 4, 16]).

A possible explanation for these results is that temporal graph problems are *very* hard. Another one is that parameters based on static graph properties are not adequate. Some parameters whose definition is based on that of a temporal graph include timed feedback vertex sets (counting the cumulative distance to trees over all snapshots) [10] and the $p(\mathcal{G})$ parameter from [4], that measures in a certain way how dynamic the temporal graph is and enables polynomial kernels for the exploration problem. While these parameters represent some progress towards finer-grained restrictions, they remain somewhat structural in the sense that their definition is stable under re-shuffling of the snapshots.

A key aspect of temporal graphs is that the *ordering of events* matters. Arguably, a truly temporal parameter should be sensitive to that. An interesting step in this direction was recently made by Bumpus and Meeks [7], introducing interval-membership-width, a parameter that quantifies the extent to which the set of intervals defined by the first and last appearance of an edge at each vertex can overlap (with application to Eulerian paths). In a sense, this parameter measures how complex the interleaving of events could be. Another, perhaps even more fundamental feature of temporal graphs is that the reachability relation based on temporal paths is not guaranteed to be symmetric or transitive. While the former is a well-known limitation of directed graphs, the latter is specific to temporal graphs (directed or not), and it has been suspected to be one of the main sources of intractability since the onset of the theory. (Note that a temporal graph of bounded interval-membership-width may still be arbitrarily non-transitive.) In the present work, we explore new parameters that control how transitive a temporal graph is, thereby isolating, and confirming, the role that this aspect plays in the tractability of temporal reachability problems.

Our Contributions. We introduce and investigate two parameters that measure how far a temporal graph is from having transitive reachability. For a temporal graph $\mathcal{G}$, our parameters directly address the reachability features of $\mathcal{G}$, and as such, they are formulated in terms of its reachability graph $G_R = (V, \{(u, v) : u \rightsquigarrow v\})$, a directed graph whose arcs represent the existence of temporal paths in $\mathcal{G}$, whether $\mathcal{G}$ itself is directed or undirected. Indeed, the reachability of $\mathcal{G}$ is transitive if and only if the arc relation of G_R is transitive. Two natural ways of measuring this distance are in terms of *vertex deletion* and *arc modification*, namely:

- *Vertex-deletion distance to transitivity* (δ_{vd}) is the minimum number of vertices whose deletion from G_R makes the resulting graph transitive.
- *Arc-modification distance to transitivity* (δ_{am}) is the minimum number of arcs whose addition or deletion from G_R makes the resulting graph transitive.

As for the arc-modification distance, we may occasionally consider its restriction to arc-addition only (δ_{aa}).

Among the many problems that were shown intractable in temporal graphs, one of the first, and perhaps most iconic one, is the computation of temporal connected components [6] (see also [13, 23]). In order to benchmark our new parameters, we investigate their impact on the computational complexity of this problem. Informally, given a temporal graph $\mathcal{G}$ (defined later) on a set of vertices V, a temporal connected component is a subset $V' \subseteq V$ such that for all u and v in V', u can reach v by a temporal path. Interestingly, the non-transitive nature of reachability here makes it possible for such vertices to reach each other through temporal paths that travel outside the component, without absorbing the intermediate vertices into the component. This gives rise to two distinct notions of components: *open* temporal connected components (Open-TCC) and *closed* temporal connected components (Closed-TCC), the latter requiring that only internal vertices are used in the temporal paths, and both being NP-hard to compute.

The statement of our results requires a few more facts. Both algorithmic and structural results in temporal graphs are highly sensitive to subtle definitional variations, called *settings*. In the *non-strict* setting, the labels along a temporal path are only required to be non-decreasing, whereas in the *strict* setting, they must be increasing. It turns out that both settings are sometimes incomparable in difficulty, and the techniques developed for each may be different. Some temporal graphs, called *proper*, have the property that no two adjacent edges share a common time label, making it possible to ignore the distinction between strict and non-strict temporal paths. Whenever possible, hardness results should preferably be obtained for proper temporal graphs, so that they apply in both settings at once. Finally, with a few exceptions, our results hold for both directed and undirected temporal graphs.

Bearing these notions in mind, our results are the following. For Open-TCC, we obtain an FPT algorithm with parameter δ_{vd}, running in time $3^{\delta_{\mathrm{vd}}} \cdot n^{\mathcal{O}(1)}$ (in all the settings). Unfortunately, δ_{vd} turns out to be too small for obtaining a kernel of polynomial size. In fact, we show that under reasonable computational complexity assumptions, no polynomial kernel in $\delta_{\mathrm{vd}} + \mathrm{vc} + k$ exists (except possibly for the non-strict undirected setting), where k denotes the size of the sought tcc and where vc denotes the vertex cover number of the underlying graph. Next, we obtain an FPT algorithm running in time $4^{\delta_{\mathrm{am}}} \cdot n^{\mathcal{O}(1)}$ for the mostly larger parameter δ_{am}, and show that Open-TCC admits a polynomial kernel of size

$|M|^3$, where M is a given arc set for which $(V, A(G_R)\Delta M)$ is transitive. It also admits a polynomial kernel of size δ_{aa}^2 when restricting modification to addition-only (again, all these results hold in all the settings). Closed-TCC, in comparison, seems to be a harder problem, at least with respect to our parameters. In particular, we show that it remains NP-hard even if $\delta_{\mathrm{am}} = \delta_{\mathrm{vd}} = 1$ in all the settings (through proper graphs). It is also W[1]-hard when parameterized by $\delta_{\mathrm{vd}} + \delta_{\mathrm{am}} + k$ in all the settings, except possibly in the non-strict undirected setting. In fact, these two results hold even for temporal graphs whose reachability graph misses a single arc towards being a bidirectional clique.

Put together, these results establish clearly that non-transitivity is a genuine source of hardness for Open-TCC. The case of Closed-TCC is less clear. On the one hand, the parameters do not suffice to make this particular version of the problem tractable. This is not so surprising, as the reachability graph itself does not encode which paths are responsible for reachability, in particular, whether these paths are internal or external in a component. On the other hand, this gives us a separation between both versions of the problem and provides some support for the fact that Closed-TCC may be harder than Open-TCC, which was not known before. Finally, the negative results for Closed-TCC can serve as a landmark result for guiding future efforts in defining transitivity parameters that exploit more sophisticated structures than the reachability graph.

Organization of the Work. The main definitions are given in Section 2. Then, we investigate each parameter in a dedicated section (δ_{vd} in Section 3 and δ_{am} in Section 4). The limitations of these parameters in the case of Closed-TCC are presented in Section 5. Finally, Section 6 concludes the paper with some remarks and open questions. Due to space limitations, the proofs of statements marked with ($\star$) are deferred to a full version.

2 Preliminaries

For concepts of parameterized complexity, like FPT, W[1]-hardness, and polynomial kernels, we refer to the standard monographs [14, 15]. A reduction g between two parameterized problems is called a *polynomial parameter transformation*, if the reduction can be computed in polynomial time and, if for every input instance (I, k), we have that $(I', k') = g(I, k)$ with $k' \in k^{\mathcal{O}(1)}$. We call a polynomial time reduction from a problem L to L itself a *self-reduction*.

Notation. Let j be a positive integer, we denote with $[j]$ the set $\{1, 2, \ldots, j\}$. Moreover, for $1 \le i \le j$, we define $[i, j] := [j] \setminus [i-1]$. For a decision problem L, we say that two instances I_1, I_2 of L are *equivalent* if I_1 is a yes-instance of L if and only if I_2 is a yes-instance of L. For two sets A and B, we denote with $A \Delta B$ the symmetric difference of A and B.

Graphs. We consider a graph $G = (V, E)$ to be a static graph. If not indicated otherwise, we assume G to be undirected. Given a (directed) graph G, we denote by $V(G)$ the set of vertices of G, by $E(G)$ (respectively, $A(G)$) the set of edges (arcs) of G. Let $G = (V, E)$ be a graph and let $X \subseteq V(G)$ be a set of vertices. We denote by $E_G(X) = \{\{u, v\} \in E \mid u \in X, v \in X\}$ the edges in G between the vertices of S. Moreover, we define the following operations on G: $G[X] = (X, E_G[X])$, $G - X = G[V \setminus X]$. We call a sequence $\rho = v_0, v_1, \ldots, v_r$ of vertices a *path* in graph G if $v_0, \ldots, v_r \in V(G)$ and for each $i \in [r]$, $\{v_{i-1}, v_i\} \in E(G)$. We denote with $N_G[v]$ the closed neighborhood of the vertex $v \in V(G)$. A vertex set $S \subseteq V$ is a *clique* in an undirected graph, if each pair of vertices in S is adjacent in G. For a directed graph $G = (V, A)$, we call a set $S \subseteq V$ a *bidirectional clique*, if for every pair of distinct vertices u, v in S, we have $(u, v) \in A$ and $(v, u) \in A$. Let $G = (V, A)$ be a directed graph.

A *strongly connected component (scc)* in G is an inclusion maximal vertex set $S \subseteq V$ under the property that there is a directed path in G between any two vertices of S. For each directed graph G, there is a unique partition of the vertex set of G into sccs. Moreover, this partition can be computed in linear time [24].

Temporal graphs. A *temporal graph* $\mathcal{G}$ over a set of vertices V is a sequence $\mathcal{G} = (G_1, G_2, \dots, G_L)$ of graphs such that for all $t \in [L], V(G_t) = V$. We call L the *lifetime* of $\mathcal{G}$ and for $t \in [L]$, we call $G_t = (V, E_t)$ the *snapshot graph* of $\mathcal{G}$ at *time step* t. We call $G = (V, E)$ with $E = \bigcup_{t\in[L]} E_t$ the *underlying graph* of $\mathcal{G}$. We denote by $V(\mathcal{G})$ the set of vertices of $\mathcal{G}$. We write V if the temporal graph is clear from context. We call an undirected temporal graph $\mathcal{G} = (G_1, G_2, \dots, G_L)$ *proper*, if for each vertex $v \in V(\mathcal{G})$ the degree of v in G_t is one, for each $t \leq L$. We call a directed temporal graph $\mathcal{G} = (G_1, G_2, \dots, G_L)$ *proper*, if for each vertex $v \in V(\mathcal{G})$ the out-degree or the in-degree of v in G_t is zero, for each $t \leq L$. We further call a (directed) temporal graph $\mathcal{G}$ *simple*, if each edge (arc) exists in exactly one snapshot. We call a sequence $v_0, v_1, \dots, v_r$ of vertices that form a path in the underlying graph G of $\mathcal{G}$ a *strict (non-strict) temporal path* in $\mathcal{G}$ if for each $i \in [r]$, there exists an $j_i \in [L]$ such that $\{v_{i-1}, v_i\} \in E(G_{j_i})$ and the sequence of indices j_i is increasing (non-decreasing).

For a temporal graph $\mathcal{G}$, we say that a vertex $u \in V$ *strictly (non-strictly) reaches* a vertex $v \in V$ if there is a strict (non-strict) temporal path from u to v, i.e., with $v_0 = u$ and $v_r = v$. We define the *strict (non-strict) reachability relation* $R \subseteq V \times V$ as: for all $u, v \in V$, $(u, v) \in R$ if and only if u strictly (non-strictly) reaches v. We call the directed graph $G_R = (V, R)$ the *strict (non-strict) reachability graph* of $\mathcal{G}$. We say that G_R is transitive, if and only if R is transitive. More generally, we say that a directed graph G is *transitive*, if its set of arcs forms a transitive relation. For a directed graph $G = (V, A)$ we call a set of vertices $S \subseteq V$ a *transitivity modulator* if $G - S$ is transitive.

▶ **Observation 1.** *Let G be a transitive directed graph. Then, for each vertex $v \in V(G)$, $G[V \setminus \{v\}]$ is also transitive.*

Next we define our main problems of interest in this work: Finding open and closed temporal connected components.

> Open Temporal Connected Component (Open-TCC)
> **Input:** Temporal graph $\mathcal{G} = (G_1, G_2, \dots, G_L)$ and integer k.
> **Question:** Does there exists an open temporal connected component of size at least k, i.e., a subset $C \subseteq V(\mathcal{G})$ with $|C| \geq k$, such that for each $u, v \in C$, u reaches v, and vice versa.

We differentiate between the strict vs. non-strict and directed vs. undirected version of Open-TCC depending on whether we consider strict vs. non-strict reachability and directed vs. undirected temporal graphs. We define the problem Closed Temporal Connected Component (Closed-TCC) similarly with the additional restriction that at least one temporal path over which u reaches v is fully contained in C. We abbreviate a temporal connected component as tcc.

Distance to transitivity. We introduce two parameters that measure how far the reachability graph $G_R = (V, A)$ of a temporal graph is from being transitive. The first parameter, *vertex-deletion distance to transitivity*, δ_{vd}, counts how many vertices need to be deleted from G_R in order to obtain a transitive reachability graph, i.e., the size of a minimum transitivity modulator. This parameter is especially suited for temporal graphs for which the reachability graph consists of cliques with small overlaps. The second parameter, *arc-modification distance* *to transitivity*, δ_{am}, counts how many arcs need to be added to or removed from G_R in order to obtain a transitive reachability graph and is especially suited for directed temporal graphs or temporal graphs for which the reachability graph consists of cliques with large overlaps. Formally, we define the parameters as follows.

$$\delta_{\mathrm{vd}} = \min_{S \subseteq V}(|S|) \text{ for which } G'_R = G_R - S \text{ is transitive.}$$

$$\delta_{\mathrm{am}} = \min_{M \subseteq V \times V}(|M|) \text{ for which } G'_R = (V, A \Delta M) \text{ is transitive.}$$

For δ_{am}, we call the set M an *arc-modification set*. Note that $\delta_{\mathrm{vd}} \leq 2 \cdot \delta_{\mathrm{am}}$, since the endpoints of an arc-modification set form a transitivity modulator.

2.1 Basic Observations

Next, we present basic observations that motivate the study of the considered parameters.

▶ **Lemma 2** ([6]). *Let $\mathcal{G}$ be a temporal graph with reachability graph G_R. Then a set $S \subseteq V(\mathcal{G})$ is a tcc in $\mathcal{G}$ if and only if S is a bidirectional clique in G_R.*

▶ **Lemma 3** ($\star$). *Let G be a transitive directed graph. Then every vertex set $S \subseteq V(G)$ is a bidirectional clique in G if and only if each pair of vertices of S can reach each other.*

Note that this implies the following.

▶ **Corollary 4.** *Let G be a transitive directed graph. Then every scc in G is also a maximal bidirectional clique and vice versa.*

The previous observations thereby imply that both Open-TCC and Closed-TCC can be solved in polynomial time on temporal graphs with transitive reachability graphs.

3 Vertex-Deletion Distance to Transitivity

We first focus on the parameter δ_{vd}. Note that computing this parameter is NP-hard: In a strict temporal graph $\mathcal{G}$ with lifetime 1, the reachability graph G_R of $\mathcal{G}$ is exactly the directed graph obtained from orienting each edge of the underlying graph in both directions. Hence, on such a temporal graph, computing δ_{vd} is exactly the cluster vertex deletion number of the underlying graph, that is, the minimum size of any vertex set to remove, such that no induced path of length 2 remains. Since computing the latter parameter is NP-hard [22], this hardness also translates to computing the parameter δ_{vd}.

Moreover, note that computing this parameter can be done similarly to computing the cluster vertex deletion number of a graph: If a directed graph $G = (V, A)$ is not transitive, then there are vertices u, v, and w in V, such that (u, v) and (v, w) are arcs of A and (u, w) is not an arc of A. Hence, each transitivity modulator for G has to contain at least one of the vertices u, v, or w. This implies, that a standard branching algorithm that considers each of these three vertices to be removed from the graph to obtain a transitive graph, finds a minimum size transitivity modulator in $3^{\delta_{\mathrm{vd}}} \cdot n^{\mathcal{O}(1)}$ time.

▶ **Proposition 5.** *Let $\mathcal{G}$ be a temporal graph with reachability graph G_R. Then, we can compute in time $3^{\delta_{\mathrm{vd}}} \cdot n^{\mathcal{O}(1)}$ a minimal-size transitivity modulator of G_R.*

Based on this result, we now present an FPT-algorithm for Open TCC when parameterized by δ_{vd}.

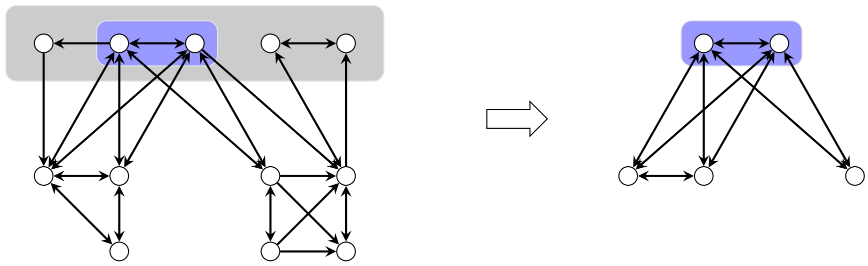

Figure 1 Illustration of the algorithm in Lemma 6. On the left: reachability graph G_R with transitivity modulator S in gray and the chosen subset $S' \subseteq S$ to extend in blue. On the right: The subset S' together with the vertices V' that are bidirectionally connected to all vertices in S'.

▶ **Lemma 6.** *Let $I := (\mathcal{G}, k)$ be an instance of* OPEN-TCC *with reachability graph G_R. Let S be a given transitivity modulator of G_R. Then, we can solve I in time $2^{|S|} \cdot n^{\mathcal{O}(1)}$.*

Proof. By Lemma 4, every scc in $G_R[V \setminus S]$ is a bidirectional clique, since S is a transitivity modulator for G_R. Lemma 2 then implies that each tcc C in $\mathcal{G}$ with $C \cap S = \emptyset$ is an scc in $G_R[V \setminus S]$ and vice versa.

The FPT-algorithm then works as follows: We iterate over all subsets S' of S with the idea to find a tcc that extends S'. If S' is not a bidirectional clique in G_R, we discard the current set and continue with the next subset of S, as no superset of S' is a bidirectional clique and thus also not a tcc. Hence, assume that S' is a bidirectional clique. If S' has size at least k, I is a trivial yes-instance of OPEN-TCC. Otherwise, we do the following: Let V' be the vertices of $V \setminus S$ that are bidirectional connected to every vertex in S'. As $G_R[V \setminus S]$ is transitive, Observation 1 implies that $G_R[V']$ is also transitive. Hence, the sccs in $G_R[V']$ correspond to tccs in $\mathcal{G}$ by Corollary 4 and Lemma 2. Since every vertex in S' is bidirectional connected to every other vertex in $S' \cup V'$ in G_R, for each bidirectional clique $C \subseteq V'$ in $G_R[V']$, $C \cup S'$ is a tcc in $\mathcal{G}$. Hence, it remains to check, whether any scc in $G_R[V']$ has size at least $k - |S'|$. Figure 1 illustrates the sets S, S', and V'.

Finding the strongly connected components of a graph and identifying whether a set of vertices forms a bidirectional clique can be done in polynomial time. Hence, our algorithm runs in time $2^{\delta_{\mathrm{vd}}} \cdot n^{\mathcal{O}(1)}$, since we iterate over each subset S' of S. ◀

Based on Proposition 5 and Lemma 6, we thus derive our FPT-algorithm for OPEN-TCC when parameterized by δ_{vd}.

▶ **Theorem 7.** *OPEN-TCC can be solved in $3^{\delta_{\mathrm{vd}}} \cdot n^{\mathcal{O}(1)}$ time.*

Kernelization Lower Bounds

In this section, we show that a polynomial kernel for OPEN-TCC when parameterized by $\delta_{\mathrm{vd}} + \mathrm{vc} + k$ is unlikely, where vc is the vertex cover number of the underlying graph. Note that δ_{vd} and vc are incomparable: On the one hand, consider a temporal graph $\mathcal{G}$ where the underlying graph G is a star with leaf set $X \cup Y$ and center c, such that the edges from X to c exist in snapshots G_1 and G_3 and the edges from Y to c exist in snapshot G_2. Then, each vertex of X can reach each other vertex, but in the strict setting, no vertex of Y can reach any other vertex of Y. Hence, each minimum transitivity modulator has to contain all vertices of X or all but one vertex of Y, which implies that for $|X| = |Y|$, $\delta_{\mathrm{vd}} \in \Theta(|V(\mathcal{G})|)$, whereas the vertex cover number of G is only 1. On the other hand, consider a temporal graph $\mathcal{G}$ with only one snapshot G_1, such that G_1 is a clique. Then, the underlying graph of $\mathcal{G}$ is exactly G_1 and has a vertex cover number of $|V(\mathcal{G})| - 1$, but the strict reachability graph of $\mathcal{G}$ is a bidirectional clique, which is a transitive graph. Hence, $\delta_{\text{vd}}(\mathcal{G}) = 0$.

We now present our kernelization lower bound for the strict undirected version of Open-TCC.

▶ **Theorem 8.** *The strict undirected version of Open-TCC does not admit a polynomial kernel when parameterized by* $\text{vc} + \delta_{\text{vd}} + k$*, unless* NP ⊆ coNP/poly*, where* vc *denotes the vertex cover number of the underlying graph.*

Proof. This result immediately follows from the known [21] reduction from Clique which, in fact, is as a polynomial parameter transformation.

Clique
Input: An undirected graph $G = (V, E)$ and integer k.
Question: Is there a clique of size k in G?

For the sake of completeness, we recall the reduction. Let $I := (G = (V, E), k)$ be an instance of Clique and let $\mathcal{G}$ be the temporal graph with lifetime 1, where G is the unique snapshot of $\mathcal{G}$.

Then, for each vertex set $X \subseteq V$, X is a clique in G if and only if X is a strict tcc in $\mathcal{G}$. Hence, I is a yes-instance of Clique if and only if $(\mathcal{G}, k)$ is a yes-instance of the strict undirected version of Open-TCC. It is known that Clique does not admit a polynomial kernel when parameterized by k plus the vertex cover number of G, unless NP ⊆ coNP/poly [12]. Let S be a minimum size vertex cover of G and let G_R be the strict reachability graph of $\mathcal{G}$. Note that G_R contains an arc (u, v) with $u \neq v$ if and only if $\{u, v\}$ is an edge of G. Hence, $V \setminus S$ is an independent set in G_R, which implies that S is a transitivity modulator of G_R. Consequently, $\delta_{\text{vd}} \leq |S|$. Recall that Clique does not admit a polynomial kernel when parameterized by $k + |S|$, unless NP ⊆ coNP/poly [12]. This implies that the strict undirected version of Open-TCC does not admit a polynomial kernels when parameterized by $\text{vc} + \delta_{\text{vd}} + k$, unless NP ⊆ coNP/poly. ◀

Next, we present the same lower bound for both directed versions of Open-TCC.

▶ **Theorem 9.** *The directed version of Open-TCC does not admit a polynomial kernel when parameterized by* $\text{vc} + \delta_{\text{vd}} + k$*, unless* NP ⊆ coNP/poly*, where* vc *denotes the vertex cover number of the underlying graph. This holds both for the strict and the non-strict version of the problem.*

Proof. Again, we present a polynomial parameter transformation from Clique.

Recall that Clique does not admit a polynomial kernel when parameterized by the size of a give minimum size vertex cover S of G plus k, unless NP ⊆ coNP/poly [12]. This holds even if $G[S]$ is $(k-1)$-partite [20], which implies that each clique of size k in G contains exactly $k - 1$ vertices of S and exactly one vertex of $V \setminus S$, since $V \setminus S$ is an independent set.

Construction. Let $I := (G := (V, E), k)$ be an instance of Clique and let S be a given minimum size vertex cover S of G, such that $G[S]$ is $(k-1)$-partite. Assume that $k > 6$.

We obtain an equivalent instance of Open-TCC in two steps: First, we perform an adaptation of a known reduction [6] from the instance $(G[S], k-1)$ of Clique to an instance $(\widetilde{\mathcal{G}}, k-1)$ of the directed version of Open-TCC where each sufficiently large (of size at least 5) vertex set X of $\widetilde{\mathcal{G}}$ is a tcc in $\widetilde{\mathcal{G}}$ if and only if X is a clique in $G[S]$. Second, we

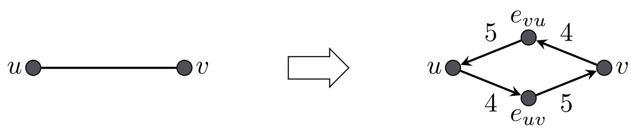

Figure 2 For two adjacent vertices u and v of S the vertices and arcs added to the temporal graph $\widetilde{\mathcal{G}}$ in the proof of Theorem 9.

extend $\widetilde{\mathcal{G}}$ by the vertices of $V \setminus S$ and some additional connectivity-gadgets, to ensure that the resulting temporal graph has a tcc of size k if and only if there is a vertex from $V \setminus S$ for which the neighborhood in G contains a clique of size $k-1$.

Let $(\widetilde{\mathcal{G}}, k-1)$ be the instance of Open-TCC constructed as follows: We initialize $\widetilde{\mathcal{G}}$ as an edgeless temporal graph of lifetime 5 with vertex set $S \cup \{e_{uv}, e_{vu} \mid \{u,v\} \in E_G(S)\}$. Next, for each edge $\{u,v\} \in E$, we add the arcs (u, e_{uv}) and (v, e_{vu}) to time step 4 and add the arcs (e_{uv}, v) and (e_{vu}, u) to time step 5. This completes the construction of $\widetilde{\mathcal{G}}$. An example of the arcs added to $\widetilde{\mathcal{G}}$ is shown in Figure 2. Note that the first three snapshots of $\widetilde{\mathcal{G}}$ are edgeless. This construction is an adaptation of the reduction presented by Bhadra and Ferreira [6] to the case of directed temporal graphs. Note that the temporal graph $\widetilde{\mathcal{G}}$ has the following properties that we make use of in our reduction:

1) $\widetilde{\mathcal{G}}$ is a proper and simple directed temporal graph,
2) the vertex set $\mathcal{V}$ of $\widetilde{\mathcal{G}}$ has size $\mathcal{O}(|S|^2)$ and contains all vertices of S,
3) each tcc of size at least $k-1$ in $\widetilde{\mathcal{G}}$ contains only vertices of S, and
4) each vertex set $X \subseteq S$ of size at least $k-1$ is a tcc in $\widetilde{\mathcal{G}}$ if and only if X is a clique in $G[S]$.

Note that the two last properties imply that the largest tcc of $\widetilde{\mathcal{G}}$ has size at most $k-1$, since $G[S]$ is $(k-1)$-partite.

Next, we describe how to extend the temporal graph $\widetilde{\mathcal{G}}$ to obtain a temporal graph $\mathcal{G}'$ which has a tcc of size k if and only if I is a yes-instance of Clique. Let $n := |V|$. Moreover, let $\mathcal{G}'$ be a copy of $\widetilde{\mathcal{G}}$. We extend the vertex set of $\mathcal{G}'$ by all vertices of $V \setminus S$, and a vertex v_{in} for each vertex $v \in S$.

For each vertex $v \in S$, we add the arc (v_{in}, v) to time step 3. For each vertex $v \in S$ and each neighbor $w \in V \setminus S$ of v in G, we add the arc (v, w) to time step 2 and the arc (w, v_{in}) to time step 1. This completes the construction of $\mathcal{G}'$. Let V' denote the newly added vertices, that is, $V' := (V \setminus S) \cup \{v_{\text{in}} \mid v \in S\}$.

Next, we show that there is a clique of size k in G if and only if there is a tcc of size k in $\mathcal{G}'$.

($\Rightarrow$) Let $K \subseteq V$ be a clique of size k in G. We show that K is a tcc in $\mathcal{G}'$. As discussed above, K contains exactly $k-1$ vertices of S and exactly one vertex w^* of $V \setminus S$. By construction of $\widetilde{\mathcal{G}}$, $K \setminus \{w^*\}$ is a tcc in $\widetilde{\mathcal{G}}$ and thus also a tcc in $\mathcal{G}'$. It thus remains to show that each vertex $K \setminus \{w^*\}$ can reach vertex w^* in $\mathcal{G}'$ and vice versa. Since each vertex of $K \setminus \{w^*\}$ is adjacent to w^* in G, by construction, w^* is an out-neighbor of each vertex of $K \setminus \{w^*\}$ in $\mathcal{G}'$. Hence, it remains to show that w^* can reach each vertex of $K \setminus \{w^*\}$ in $\mathcal{G}'$. Let v be a vertex of $K \setminus \{w^*\}$. Since v is adjacent to w^* in G, there is an arc (w^*, v_{in}) in $\mathcal{G}'$ that exists at time step 1. Hence, there is a temporal path from w^* to v in $\mathcal{G}'$, since the arc (v_{in}, v) exists at time step 3. Concluding, K is a tcc in $\mathcal{G}'$.

($\Leftarrow$) Let X be a tcc of size k in $\mathcal{G}'$. We show that X is a clique of size k in G. To this end, we first show that X contains only vertices of V. Afterwards, we show that X is a clique

Table 1 For each vertex $v \in V(\mathcal{G}')$ a lower bound for $\text{out}_v^{\min}$ and an upper bound for $\text{in}_v^{\max}$.

	$\text{out}_v^{\min}$	$\text{in}_v^{\max}$
$v \in S$	2	5
$v \in V(\widetilde{\mathcal{G}}) \setminus S$	4	5
$v \in V \setminus S$	1	2
$v \in \{u_{\text{in}} \mid u \in S\}$	3	1

in G.

To show that X contains only vertices of V, we first analyze the reachability of vertices of $V(\mathcal{G}')$. For a vertex $v \in V(\mathcal{G}')$, we denote

- by $\text{out}_v^{\min}$ the smallest time label of any arc exiting v and
- by $\text{in}_v^{\max}$ the largest time label of any arc entering v.

Note that a vertex v cannot reach a distinct vertex w in $\mathcal{G}'$ if $\text{in}_w^{\max} < \text{out}_v^{\min}$. Table 1 shows for each vertex $v \in V(\mathcal{G}')$ a lower bound for $\text{out}_v^{\min}$ and an upper bound for $\text{in}_v^{\max}$.

Based on Table 1, we can derive the following properties about reachability in $\mathcal{G}'$.

▷ Claim 10. **a)** No vertex of $V(\widetilde{\mathcal{G}}) \setminus S$ can reach any vertex of V' in $\mathcal{G}'$.
b) No vertex of $\{v_{\text{in}} \mid v \in S\}$ can reach any other vertex of $\{v_{\text{in}} \mid v \in S\}$ in $\mathcal{G}'$.
c) No vertex of $\{v_{\text{in}} \mid v \in S\}$ can reach any vertex of $V \setminus S$ in $\mathcal{G}'$.
d) No vertex of S can reach any vertex of $\{v_{\text{in}} \mid v \in S\}$ in $\mathcal{G}'$.
e) No vertex of $V \setminus S$ can reach any other vertex of $V \setminus S$ in $\mathcal{G}'$.

Proof. Based on Table 1, we derive Items a) to d). It remains to show Item e). To this end, observe that each arc with a vertex of $V \setminus S$ as source has a vertex of $\{v_{\text{in}} \mid v \in S\}$ as sink. Due to Item c), no vertex of $\{v_{\text{in}} \mid v \in S\}$ can reach any vertex of $V \setminus S$ in $\mathcal{G}'$. Hence, no vertex $V \setminus S$ can reach any other vertex of $V \setminus S$ in $\mathcal{G}'$. This implies that Item e) holds. ◁

Since X is a tcc in $\mathcal{G}'$, Claim 10 implies that X contains at most one vertex of $V \setminus S$ (due to Item e)) and at most one vertex of $\{v_{\text{in}} \mid v \in S\}$ (due to Item b)). In other words, X contains at most two vertices of V'. Since $k > 6$, this then implies that X contains at least one vertex of $V(\widetilde{\mathcal{G}})$. Claim 10 thus further implies that X contains no vertex of $\{v_{\text{in}} \mid v \in S\}$ (due to Items a) and d)). This then implies that X contains at least $k-1$ vertices of $V(\widetilde{\mathcal{G}})$.

To show that X contains only vertices of V and is a clique in G we now show that the reachability between any two vertices of $V(\widetilde{\mathcal{G}})$ in $\mathcal{G}'$ is the same as in $\widetilde{\mathcal{G}}$. Let P be a temporal path between two distinct vertices of $V(\widetilde{\mathcal{G}})$ in $\mathcal{G}'$. We show that P is also a temporal path in $\widetilde{\mathcal{G}}$. Assume towards a contradiction that this is not the case. Hence, P visits at least one vertex of V'. Since no vertex of $V(\widetilde{\mathcal{G}})$ can reach any vertex of $\{v_{\text{in}} \mid v \in S\}$ (due to Items a) and d)), P visits no vertex of $\{v_{\text{in}} \mid v \in S\}$. Moreover, since each vertex of $V \setminus S$ has only out-neighbors in $\{v_{\text{in}} \mid v \in S\}$, P visits no vertex of $V \setminus S$ either. Consequently, P contains no vertex of V'; a contradiction.

Hence, P is a temporal path in $\widetilde{\mathcal{G}}$, which implies that for each vertex set $Y \subseteq V(\widetilde{\mathcal{G}})$, Y is a tcc in $\widetilde{\mathcal{G}}$ if and only if Y is a tcc in $\mathcal{G}'$. Recall that X contains at least $k-1$ vertices of $V(\widetilde{\mathcal{G}})$. Since the largest tcc in $\widetilde{\mathcal{G}}$ has size at most $k-1$ and each tcc of size $k-1$ in $\widetilde{\mathcal{G}}$ is a clique in G, this implies that $X \cap V(\widetilde{\mathcal{G}})$ is a clique of size $k-1$ in $G[S]$. Since X contains no vertex of $\{v_{\text{in}} \mid v \in S\}$, this implies that X contains exactly one vertex w^* of $V \setminus S$. Hence, it remains to show that each vertex $v \in X \setminus \{w^*\}$ is adjacent to w^* in G. Since X is a tcc in $\mathcal{G}'$, v can reach w^* in $\mathcal{G}'$. By construction and illustrated in Table 1, $\text{out}_v^{\min} \geq 2 \geq \text{in}_{w^*}^{\max}$. Since v reaches w^* and $\mathcal{G}'$ is a proper temporal graph, the arc (v, w^*) is contained in $\mathcal{G}'$. By construction, this implies that v and w^* are adjacent in G. Consequently, X is a clique in G. This completes the correctness proof of the reduction.

Parameter bounds. It thus remains to show that $\delta_{\text{vd}}(\mathcal{G}')$ and the vertex cover of the underlying graph of $\mathcal{G}'$ are at most $|S|^{\mathcal{O}(1)}$ each. Let $V^* := V(\mathcal{G}') \setminus (V \setminus S)$. Note that V^* has size $|V(\widetilde{\mathcal{G}})| + |S| \in \mathcal{O}(|S|^2)$ and is a vertex cover of the underlying graph of $\mathcal{G}'$. Hence, the vertex cover number of the underlying graph of $\mathcal{G}'$ is $\mathcal{O}(|S|^2)$. To show the parameter bounds, it thus suffices to show that V^* is a transitivity modulator of the reachability graph G_R of $\mathcal{G}'$. Due to Claim 10, $G_R - V^* = G_R[V \setminus S]$ is an independent set. Consequently, V^* is a transitivity modulator of G_R. Hence, $\delta_{\text{vd}}(\mathcal{G}') \in \mathcal{O}(|S|^2)$. By the fact that CLIQUE does not admit a polynomial kernel when parameterized by $|S| + k$, unless NP $\subseteq$ coNP/poly, OPEN-TCC does not admit a polynomial kernel when parameterized by $\delta_{\text{vd}}(\mathcal{G}')$ plus the vertex cover number of the underlying graph of $\mathcal{G}'$ plus k, unless NP $\subseteq$ coNP/poly. ◀

Note that our kernelization lower bounds do not include the non-strict undirected version of OPEN-TCC. An modification of Theorem 9 seems difficult, unfortunately. This is due to the fact that undirected edges can be traversed in both direction, which makes it very difficult to limit the possible reachable vertices in the temporal graph, while preserving a small transitivity modulator.

4 Arc-Modification Distance to Transitivity - A Polynomial Kernel

Next, we focus on the parameterized complexity of OPEN-TCC when parameterized by the size of a given arc-modification set towards a transitive reachability graph. As discussed earlier, for each arc-modification set M towards a transitive reachability graph, δ_{vd} does not exceed $2 \cdot |M|$, since removing the endpoints of all edges of M results in a transitivity modulator. This implies the following due to Theorem 7 and the fact that a minimum size arc-modification set towards a transitive graph can be computed in $2.57^{\delta_{\text{am}}} \cdot n^{\mathcal{O}(1)}$ time [25].

▶ **Corollary 11.** *OPEN-TCC can be solved in $4^{\delta_{\text{am}}} \cdot n^{\mathcal{O}(1)}$ time.*

In the remainder of this section, we thus consider this parameter with respect to kernelization algorithms. In contrast to parameterizations by δ_{vd}, we now show that a polynomial kernelization algorithm can be obtained for OPEN-TCC when parameterized by the size of a given arc-modification set towards a transitive reachability graph.

In fact, we show an even stronger result, since our kernelization algorithm does not need to know the actual arc-modification set but only its endpoints. To formulate this more general result, we need the following definition: Let $G = (V, A)$ be a directed graph. A transitivity modulator $S \subseteq V$ of G is called *inherent*, if there is an arc-modification set M with $M \subseteq S \times S$ for which $(V, A \Delta M)$ is a transitive graph. Note that the set of endpoint of an arc-modification set towards a transitive graph always forms an inherent transitivity modulator.

▶ **Theorem 12.** *Let $I = (\mathcal{G}, k)$ be an instance of OPEN-TCC and let $G_R = (V, A)$ be the reachability graph of $\mathcal{G}$. Moreover, let $B \subseteq V$ be an inherent transitivity modulator of G_R. Then, for each version of OPEN-TCC, one can compute in polynomial time an equivalent instance of total size $\mathcal{O}(|B|^3)$.*

Proof. We first present a compression to CLIQUE. Let $\widehat{G} = (V, E)$ be an undirected graph that contains an edge $\{u, v\}$ if and only if (u, v) and (v, u) are arcs of G_R. Due to Lemma 2, I is a

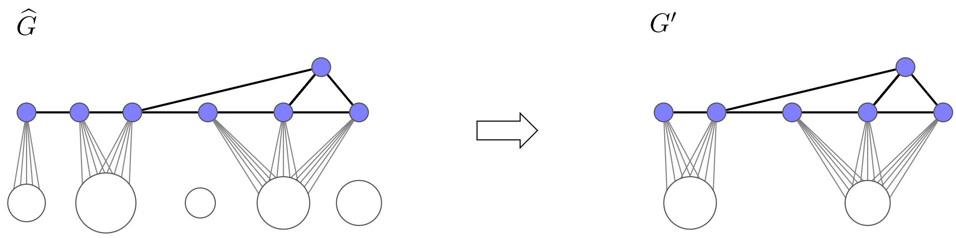

Figure 3 Left: the original instance of CLIQUE from Theorem 12 constructed from the reachability graph of the considered temporal graph. Right: the obtained compressed instance of CLIQUE after exhaustive application of all reduction rules. In both parts, the blue vertices are the vertices from the inherent transitivity modulator B and the cycles at the bottom indicate the white clusters. Note that in both graphs, each blue vertex has neighbors in at most one white cluster (see Claim 14). Intuitively, RR 1 ensures that small clusters are removed, RR 1 and RR 2 ensure that there are no isolated white clusters, and RR 3 reduces the size of each white cluster to at most $|B| + 1$.

yes-instance of OPEN-TCC if and only if $(\widehat{G}, k)$ is a yes-instance of CLIQUE. Let $W := V \setminus B$. We call the vertices of B *blue* and the vertices of W *white*. Note that $G_R[W]$ is a transitive graph, since B is a transitivity modulator of G_R. Moreover, there exists an arc set $M \subseteq B \times B$ such that $G'_R = (V, A \Delta M)$ is transitive, since B is an inherent transitivity modulator of G_R. In the following, we present reduction rules to remove vertices from $\widehat{G}$ to obtain an equivalent instance (G', k') of CLIQUE with $\mathcal{O}(|B|^2)$ vertices and where G' is an induced subgraph of $\widehat{G}$. The graphs $\widehat{G}$ and G' are conceptually depicted in Figure 3.

To obtain this smaller instance of CLIQUE, we initialize G' as a copy of $\widehat{G}$ and k' as k and exhaustively applying three reduction rules. Our first two reduction rules are the following:

- RR 1: Remove a vertex v from G', if v has degree less than $k' - 1$ in G'.
- RR 2: If a white vertex has at least $k' - 1$ white neighbors in G', output a constant size yes-instance.

Note that the first reduction rule is safe, since no vertex of degree less than $k' - 1$ can be part of a clique of size at least k'. Moreover, each connected component in G' has size at least k' after this reduction rule is exhaustively applied. The safeness of the second reduction rule relies on the following observation.

▷ Claim 13. If two white vertices u and v are adjacent in G', then they are real twins in G'. That is, $N_{G'}[u] = N_{G'}[v]$.

Proof. Assume that u and v are adjacent in G' and assume towards a contradiction that there is a vertex w in G' which is adjacent to u in G' but not adjacent to v in G'. Since w and v are not adjacent in G', G_R contains at most one of the arcs (w, v) or (v, w). Assume without loss of generality that (w, v) is not an arc of G_R. Since u is adjacent to both v and w in G', G_R contains the arcs (v, u) and (u, w). Recall that both u and v are white vertices. This implies that the arc-modification set M contains no arc incident with any of these two vertices. Hence, M contains none of the arcs of $\{(v, u), (u, w), (v, w)\}$, which implies that $G'_R = (V, A \Delta M)$ is not a transitive graph; a contradiction. ◁

Note that this implies that each connected component in $G'[W]$ is a clique of real twins in G'. We call each such connected component in $G'[W]$ a *white cluster*.

Note that after exhaustive applications of the first two reduction rules, each white cluster has size at most $k' - 1$ and each connected component in G' has size at least k'. This implies that each connected component in G' contains at least one blue vertex. Since G' contains at most $|B|$ blue vertices, this implies that G' has at most $|B|$ connected components.

In the following, we show that no blue vertex has neighbors in more than one white cluster. This then implies that G' contains at most $|B| \cdot k'$ vertices. In a final step, we then show how to reduce the value of k'.

▷ Claim 14. No blue vertex has neighbors in more than one white cluster.

Proof. Assume towards a contradiction that there is a blue vertex w which is adjacent to two white vertices u and v in G', such that u and v are not part of the same white cluster. Since u and v are not part of the same white cluster, u and v are not adjacent in G' due to Claim 13. This implies that G_R contains at most one of the arcs (u, v) or (v, u). Assume without loss of generality that (u, v) is not an arc of G_R. Since w is adjacent to both u and v in G', G_R contains the arcs (u, w) and (w, v). By the fact that both u and v are white vertices, M contains no arc of $\{(u, w), (w, v), (u, v)\}$, which implies that $G'_R = (V, A\Delta M)$ is not a transitive graph; a contradiction. ◁

As mentioned above, this implies that G' contains at most $|B| \cdot k'$ vertices. Next, we show how to reduce the size of the white clusters if $k' > |B|$. To this end, we introduce our last reduction rule:

- RR 3: If $k' > |B| + 1$, remove an arbitrary white vertex from each white cluster and reduce k' by 1.

Note that RR 3 is safe: If $k' > |B| + 1$, a clique of size k' in G' has to contain at least two white vertices, since G' contains at most $|B|$ blue vertices. Since no clique in G' can contain vertices of different white clusters, we reduce the size of a maximal clique of size at least k' in G' by exactly one, when removing one vertex of each white cluster.

Hence, after all reduction rules are applied exhaustively, the resulting instance (G', k') of Clique contains at most $|B|$ blue vertices and at most $|B|$ white clusters. Each such white cluster hast size at most $|B| + 1$. This implies that the resulting graph G' contains $\mathcal{O}(|B|^2)$ vertices and $\mathcal{O}(|B|^3)$ edges, since each vertex has degree $\mathcal{O}(|B|)$.

Based on a known polynomial-time reduction [6], we can compute for an instance (G^*, k^*) of Clique, an equivalent instance $(\mathcal{G}^*, k^*)$ of Open-TCC, where $\mathcal{G}^*$ is a proper temporal graph and has $\mathcal{O}(n + m)$ vertices and edges, where n and m denote the number of vertices and the number of edges of G^*, respectively. Since G' has $\mathcal{O}(|B|^2)$ vertices and $\mathcal{O}(|B|^3)$ edges, this implies that we can obtain an equivalent instance of Open-TCC of total size $\mathcal{O}(|B|^3)$ in polynomial time. By the fact that $\mathcal{G}^*$ is a proper temporal graph, this works for all problem versions of Open-TCC. ◀

Based on the fact that the set B of endpoints of any arc-modification set M towards a transitive graph is an inherent transitivity modulator of size at most $2 \cdot |M|$, this implies the following for kernelization algorithms with respect to arc-modification sets towards a transitive graph.

▶ **Theorem 15.** *Let $I = (\mathcal{G}, k)$ be an instance of Open-TCC and let $G_R = (V, A)$ be the reachability graph of $\mathcal{G}$. Moreover, let $M \subseteq V \times V$ be a set of arcs such that $G'_R = (V, A\Delta M)$ is transitive. Then, for each version of Open-TCC, one can compute in polynomial time an equivalent instance of total size $\mathcal{O}(|M|^3)$.*

Moreover, if the arc-modification set M only adds arcs to the reachability graph, we can obtain the further even better kernelization result.

▶ **Lemma 16** (⋆). *Let $I = (\mathcal{G}, k)$ be an instance of* Open-TCC *and let $G_R = (V, A)$ be the reachability graph of $\mathcal{G}$. Moreover, let $M \subseteq V \times V$ be a set with $A \cap M = \emptyset$ of arcs such that $G'_R = (V, A \Delta M)$ is transitive. Then, for each version of* Open-TCC*, one can compute in polynomial time an equivalent instance of total size $\mathcal{O}(|M|^2)$.*

Hence, if we are given an arc-modification set M of size δ_{am}, we can compute a polynomial kernel. Unfortunately, finding a minimum-size arc-modification set of a given directed graph is NP-hard [25] and no polynomial-factor approximations are known that run in polynomial time. Hence, we cannot derive a polynomial kernel for the parameter δ_{am}. Positively, if we only consider arc-additions, we can compute the transitive closure of a given directed graph in polynomial time. This implies that we can find a minimum-size arc-modification set towards a transitive reachability graph in polynomial time among all such sets that only add arcs to to reachability graph. Consequently, we derive the following.

▶ **Corollary 17.** Open-TCC *admits a kernel of size $\mathcal{O}(\delta_{\text{aa}}^2)$, where δ_{aa} denotes the minimum number of necessary arc-additions to make the respective reachability graph transitive.*

5 Limits of these Parametrizations for Closed TCCs

So far, the temporal paths that realize the reachability between two vertices in a tcc could lie outside of the temporal connected component. If we impose the restriction that those temporal paths must be contained in the tcc, the problem of finding a large tcc becomes NP-hard even when the reachability graph is missing only a single arc to become a complete bidirectional clique: In other words, the problem becomes NP-hard even if $\delta_{\text{vd}} = \delta_{\text{am}} = 1$. On general temporal graphs, all versions of Closed-TCC are known to be NP-hard [8, 13].

▶ **Theorem 18.** *For each version of* Closed-TCC*, there is a polynomial time self-reduction that transforms an instance $(\mathcal{G}, k)$ with $k > 4$ of that version of* Closed-TCC *into an equivalent instance $(\mathcal{G}', k)$, such that the reachability graph of $\mathcal{G}'$ is missing only a single arc to be a complete bidirectional clique.*

Proof. Let $I := (\mathcal{G}, k)$ be an instance of Closed-TCC with underlying graph $G = (V, E)$ and let L be the lifetime of $\mathcal{G}$. Moreover, assume for simplicity that the vertices of V are exactly the natural numbers from $[1, n]$ with $n := |V|$. To obtain an equivalent instance $I' := (\mathcal{G}', k)$ of Closed-TCC, we extends $\mathcal{G}$ as follows: We initialize $\mathcal{G}'$ as a copy of $\mathcal{G}$ and for each vertex $v \in V$, we add a new vertex v' to $\mathcal{G}'$. Additionally, we add three vertices x_1, x_2, and x_3 to $\mathcal{G}'$. Furthermore, we append $2n + 4$ empty snapshots to the end of $\mathcal{G}'$ and add the following edges to $\mathcal{G}'$: For each vertex $v \in V$, we add the edge $\{v, v'\}$ to time steps $L + 1$ and $L + 2n + 4$, the edge $\{v', x_1\}$ to time step $L + 1 + v$, and the edge $\{v', x_3\}$ to time step $L + n + 3 + v$. Finally, we add the edge $\{x_1, x_2\}$ to time step $L + n + 2$ and the edge $\{x_2, x_3\}$ to time step $L + n + 3$. This completes the construction of $\mathcal{G}'$. An illustration of the additional vertices and edges is given in Figure 4.

Before we show the equivalence between the two instances of Closed-TCC, we first show that the reachability graph G'_R of $\mathcal{G}'$ only misses a single arc to be a bidirectional clique.

▷ Claim 19. The arc (x_3, x_1) is the only arc that is missing in G'_R.

Proof. First, we show that (x_3, x_1) is not an arc of G'_R. By construction of $\mathcal{G}'$, (i) no edge of $\mathcal{G}'$ that is incident with x_1 exists in any time step larger than $L + n + 2$, and (ii) no edge of $\mathcal{G}'$ that is incident with x_3 exists in any time step smaller than $L + n + 3$. This implies that no temporal path in $\mathcal{G}'$ that starts in x_3 can reach x_1. Hence, (x_3, x_1) is not an arc of G'_R.

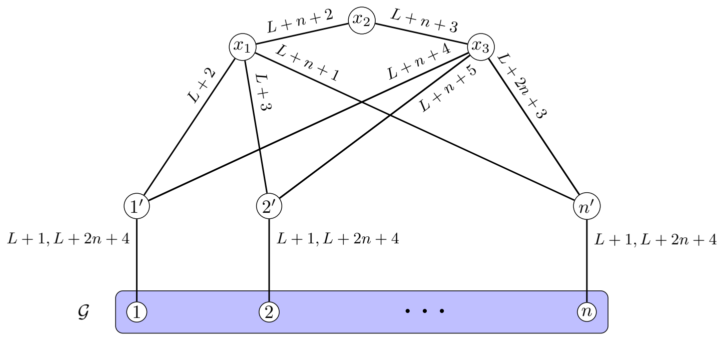

Figure 4 An illustration of the additional vertices and edges that are added to $\mathcal{G}$ in the reduction of Theorem 18. Here, L denotes the lifetime of $\mathcal{G}$ and the labels on the edges indicate in which snapshots the respective edges exist in the constructed temporal graph.

Next, we show that G'_R contains all other possible arcs. To this end, we present strict temporal paths in $\mathcal{G}'$ that guarantee the existence of these arcs in G'_R. Let u and v be vertices of V. Consider the temporal path P that starts in vertex u and traverses

- the edge $\{u, u'\}$ in time step $L + 1$,
- the edge $\{u', x_1\}$ in time step $L + 1 + u$,
- the edge $\{x_1, x_2\}$ in time step $L + n + 2$,
- the edge $\{x_2, x_3\}$ in time step $L + n + 3$,
- the edge $\{x_3, v'\}$ in time step $L + n + 3 + v$, and
- the edge $\{v', v\}$ in time step $L + 2n + 4$.

Since each vertex of V is a natural number of $[1, n]$, the times steps in which these edges are traversed by P are strictly increasing, which implies that P is a strict temporal path of $\mathcal{G}'$. Note that each suffix and each prefix of P is also a strict temporal path in $\mathcal{G}'$. Hence, this implies that G'_R contains all the arcs $\{(u, v'), (u, v), (u', v'), (u', v)\} \cup \{(u', x_1), (u', x_1), (x_1, v'), (x_1, v) \mid i \in \{1, 2, 3\}\} \cup \{(x_1, x_3)\}$. Moreover, since $\{x_1, x_2\}$ and $\{x_2, x_3\}$ are edges in $\mathcal{G}'$, G'_R also contains the arcs $\{(x_1, x_2), (x_2, x_1), (x_2, x_3), (x_3, x_2)\}$. This implies that (x_3, x_1) is the unique arc that is missing in G'_R. ◁

Next, we show that I is a yes-instance of Closed-TCC if and only if I' is a yes-instance of Closed-TCC.

($\Rightarrow$) This direction follows directly by the fact that $\mathcal{G}'$ is obtained by extending $\mathcal{G}$. Hence, a closed tcc S of size k in $\mathcal{G}$ is also a closed tcc in $\mathcal{G}'$.

($\Leftarrow$) Let S be a closed tcc of size k in $\mathcal{G}'$. Recall that $k > 4$. We show that S only contains vertices of V. To this end, we first show that S does not contain x_2.

▷ Claim 20 ($\star$). The set S does not contain x_2.

Next, we show that the vertex x_2 is required to have pairwise temporal paths between distinct vertices of $\{w' \mid w \in V\}$ in $\mathcal{G}'$.

▷ Claim 21 ($\star$). Let u and v be distinct vertices of V with $u < v$. Then, each temporal path from v' to u' in $\mathcal{G}'$ visits x_2.

As a consequence, S contains at most one vertex of $\{w' \mid w \in V\}$, since S is a closed tcc that does not contain vertex x_2. Based on the above two claims, we now show that S contains only vertices of V.

▷ Claim 22 ($\star$). Only vertices of V are contained in S.

Since no edge between any two vertices of V was added while constructing $\mathcal{G}'$ from $\mathcal{G}$, each temporal path in $\mathcal{G}'$ that visits only vertices of V is also a temporal path in $\mathcal{G}$. Together with Claim 22, this implies that S is a closed tcc in $\mathcal{G}$, which implies that I is a yes-instance of Closed-TCC. ◀

Recall that all versions of Closed-TCC are NP-hard [8, 13]. Moreover the strict undirected version of Closed-TCC is W[1]-hard when parameterized by k [8] and both directed versions of Closed-TCC are W[1]-hard when parameterized by k [13]. Together with Theorem 18, this implies the following intractability results for Closed-TCC.

▶ **Theorem 23.** *All versions of Closed-TCC are NP-hard even if $\delta_{\text{vd}} = \delta_{\text{am}} = 1$. More precisely, this hardness holds on instances where the reachability graph is missing only a single arc to be a complete bidirectional clique. Excluding the undirected non-strict version of Closed-TCC, all versions of Closed-TCC are W[1]-hard when parameterized by k under these restrictions*

6 Conclusion

We introduced two new parameters δ_{vd} and δ_{am} that capture how far the reachability graph of a given temporal graph is from being transitive. We demonstrated their applicability when the goal is to find open tccs in a temporal graph, presenting FPT-algorithms for each parameter individually, and a polynomial kernel with respect to δ_{am}, assuming that the corresponding arc-modification set of size δ_{am} is given. Computing such a set is NP-hard in general directed graphs [25]. An interesting question, also formulated in that paper, is whether this parameter is at least approximable to within a polynomial factor of δ_{am}. If so, our result implies a polynomial kernel for Open-TCC when parameterized by δ_{am}. Alternatively, the existence of a proper polynomial kernel for Open-TCC when parameterized by δ_{am} could also be shown by finding an approximation for a minimum-size inherent transitivity modulator due to Theorem 12 and the fact that the size of a minimum-size inherent transitivity modulator never exceeds $2 \cdot \delta_{\text{am}}$.

Another natural question is to identify what are other (temporal) reachability problems for which our transitivity parameters could be useful. For instance, consider a variant of the Open-TCC problem where we search for *d-tccs*, that is, tccs such that the fastest temporal path between the vertices has duration at most d. It is plausible that our positive results carry over to this version when applied to the *d-reachability graph*, i.e., the graph whose arcs represent temporal paths of duration at most d.

Regarding Closed-TCC, our intractability results show that neither δ_{vd} nor δ_{am} suffice to make this problem tractable. However, our results do not preclude the existence of an FPT-algorithm in the case that the arc modification operations are restricted to deletion only, which remains to be investigated. Nonetheless, we still believe that transitivity is a key aspect of the problem. The problem with Closed-TCC is that the reachability graph itself does not encode whether the paths responsible for reachability travel through internal or external vertices.

We would like to initiate the idea of considering further transitivity parameters based on modifications of the temporal graph itself, not only of the reachability graph. In particular, could FPT-algorithms for such parameters be achieved for reachability problems such as Open-TCC with similar performance as our FPT-algorithms for δ_{vd} and δ_{am}, and could these parameters make Closed-TCC tractable as well? And if so, how difficult is the computation of such parameters?

References

1 Eleni C Akrida, Jurek Czyzowicz, Leszek Gasieniec, Łukasz Kuszner, and Paul G Spirakis. Temporal flows in temporal networks. *Journal of Computer and System Sciences*, 103:46–60, 2019.

2 Eleni C Akrida, Leszek Gąsieniec, George B Mertzios, and Paul G Spirakis. The complexity of optimal design of temporally connected graphs. *Theory of Computing Systems*, 61:907–944, 2017.

3 Eleni C Akrida, George B Mertzios, Paul G Spirakis, and Christoforos Raptopoulos. The temporal explorer who returns to the base. *Journal of Computer and System Sciences*, 120:179–193, 2021.

4 Emmanuel Arrighi, Fedor V. Fomin, Petr A. Golovach, and Petra Wolf. Kernelizing temporal exploration problems. In Neeldhara Misra and Magnus Wahlström, editors, *18th International Symposium on Parameterized and Exact Computation, IPEC 2023, September 6-8, 2023, Amsterdam, The Netherlands*, volume 285 of *LIPIcs*, pages 1:1–1:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. URL: `https://doi.org/10.4230/LIPIcs.IPEC.2023.1`, `doi:10.4230/LIPICS.IPEC.2023.1`.

5 Kyriakos Axiotis and Dimitris Fotakis. On the size and the approximability of minimum temporally connected subgraphs. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

6 Sandeep Bhadra and Afonso Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *Ad-Hoc, Mobile, and Wireless Networks: Second International Conference, ADHOC-NOW2003, Montreal, Canada, October 8-10, 2003. Proceedings 2*, pages 259–270. Springer, 2003.

7 Benjamin Merlin Bumpus and Kitty Meeks. Edge exploration of temporal graphs. *Algorithmica*, 85(3):688–716, 2023.

8 Arnaud Casteigts. Finding structure in dynamic networks. *CoRR*, abs/1807.07801, 2018. URL: `http://arxiv.org/abs/1807.07801`, `arXiv:1807.07801`.

9 Arnaud Casteigts and Timothée Corsini. In search of the lost tree: Hardness and relaxation of spanning trees in temporal graphs. In *31th Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2024.

10 Arnaud Casteigts, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. Finding temporal paths under waiting time constraints. *Algorithmica*, 83(9):2754–2802, 2021.

11 Arnaud Casteigts, Michael Raskin, Malte Renken, and Viktor Zamaraev. Sharp thresholds in random simple temporal graphs. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–326. IEEE Computer Society, 2022.

12 Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Information and Computation*, 201(2):216–231, 2005.

13 Isnard Lopes Costa, Raul Lopes, Andrea Marino, and Ana Silva. On Computing Large Temporal (Unilateral) Connected Components. In *Combinatorial Algorithms - 34th International Workshop, IWOCA 2023, Tainan, Taiwan, June 7-10, 2023, Proceedings*, volume 13889 of *Lecture Notes in Computer Science*, pages 282–293. Springer, 2023. `doi:10.1007/978-3-031-34347-6\_24`.

14 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

15 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

16 Jessica Enright, Kitty Meeks, George B Mertzios, and Viktor Zamaraev. Deleting edges to restrict the size of an epidemic in temporal networks. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

17 Thomas Erlebach and Jakob T Spooner. Parameterized temporal exploration problems. In *1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

18 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. As time goes by: reflections on treewidth for temporal graphs. In *Treewidth, Kernels, and Algorithms: Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, pages 49–77. Springer, 2020.

19 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal graph classes: A view through temporal separators. *Theoretical Computer Science*, 806:197–218, 2020.

20 Niels Grüttemeier and Christian Komusiewicz. On the Relation of Strong Triadic Closure and Cluster Deletion. *Algorithmica*, 82(4):853–880, 2020.

21 David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 504–513, 2000.

22 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.

23 Jason Schoeters, Eric Sanlaville, Stefan Balev, and Yoann Pigné. Temporally connected components. *Available at SSRN 4590651*, 2024.

24 Robert Endre Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972. doi:10.1137/0201010.

25 Mathias Weller, Christian Komusiewicz, Rolf Niedermeier, and Johannes Uhlmann. On making directed graphs transitive. *Journal of Computer and System Sciences*, 78(2):559–574, 2012. URL: https://doi.org/10.1016/j.jcss.2011.07.001, doi:10.1016/J.JCSS.2011.07.001.