



UNIVERSITÀ
DEGLI STUDI
FIRENZE

FLORE

Repository istituzionale dell'Università degli Studi di Firenze

On Computing Large Temporal (Unilateral) Connected Components

Questa è la versione Preprint (Submitted version) della seguente pubblicazione:

Original Citation:

On Computing Large Temporal (Unilateral) Connected Components / Costa I.L.; Lopes R.; Marino A.; Silva A.. - STAMPA. - 13889:(2023), pp. 282-293. (34th International Workshop on Combinatorial Algorithms, IWOCA 2023 twn 2023) [10.1007/978-3-031-34347-6_24].

Availability:

The webpage <https://hdl.handle.net/2158/1333416> of the repository was last updated on 2023-10-12T10:32:55Z

Publisher:

Springer Science and Business Media Deutschland GmbH

Published version:

DOI: 10.1007/978-3-031-34347-6_24

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

Publisher copyright claim:

La data sopra indicata si riferisce all'ultimo aggiornamento della scheda del Repository FloRe - The above-mentioned date refers to the last update of the record in the Institutional Repository FloRe

(Article begins on next page)

On Computing Large Temporal (Unilateral) Connected Components

Isnard Lopes Costa¹ Raul Lopes^{2,3} Andrea Marino⁴
Ana Silva^{1,4}

¹Universidade Federal do Ceará, Brazil.

`isnard.lopes@alu.ufc.br; anasilva@mat.ufc.br`

²Université Paris-Dauphine, CNRS UMR7243, France.

³DIENS, École normale supérieure de Paris, CNRS, France.

`raul.wayne@gmail.com`

⁴Università degli Studi di Firenze, Italy.

`andrea.marino@unifi.it`

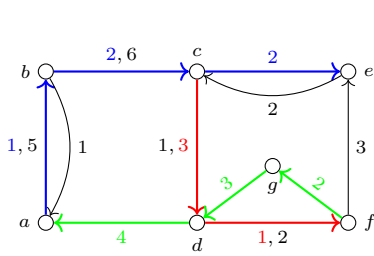
Abstract

A temporal (directed) graph is a graph whose edges are available only at specific times during its lifetime, τ . Paths are sequence of adjacent edges whose appearing times are either strictly increasing or non-strictly increasing (i.e., non-decreasing) depending on the scenario. Then, the classical concept of connected components and also of unilateral connected components in static graphs naturally extends to temporal graphs. In this paper, we answer the following fundamental questions in temporal graphs. (i) What is the complexity of deciding the existence of a component of size k , parameterized by τ , by k , and by $k + \tau$? We show that this question has a different answer depending on the considered definition of component and whether the temporal graph is directed or undirected. (ii) What is the minimum running time required to check whether a subset of vertices are pairwise reachable? A quadratic algorithm is known but, contrary to the static case, we show that a better running time is unlikely unless SETH fails. (iii) Is it possible to verify whether a subset of vertices is a component in polynomial time? We show that depending on the definition of temporal component this test is NP-complete.

1 Introduction

A (*directed*) *temporal graph* (G, λ) with *lifetime* τ consists of a (directed) graph G together with a *time-function* $\lambda : E(G) \rightarrow 2^{[\tau]}$ which tells when each edge $e \in E(G)$ is available along the discrete time interval $[\tau]$. Given $i \in [\tau]$, the *snapshot* G_i refers to the subgraph of G containing exactly the edges available in time i . Temporal graphs, also appearing in the literature under different names [14, 8, 32], have attracted a lot of attention in the past decade, as many works have extended classical notions of Graph Theory to temporal graphs (we refer the reader to the surveys [32, 36] and the seminal paper [31]).

A crucial characteristic of temporal graphs is that a u, v -walk/path in G is valid only if it traverses a sequence of adjacent edges e_1, \dots, e_k at non-decreasing times $t_1 \leq \dots \leq t_k$, respectively, with $t_i \in \lambda(e_i)$ for every $i \in [k]$. Similarly, one can



$A' = \{a, b\}$ is a closed connected set, as a and b reach each other without using external vertices.

$A = \{a, b, c, d\}$ is a maximal closed connected set, i.e. a CLOSED TCC.

$B = \{a, b, c, d, e\}$ is a CLOSED TUCC but not a CLOSED TCC as, using only vertices in B , a, b, c, d reach each other, e reaches all the vertices in B and vice versa, except for d , which does not reach e . B is also a TCC, as d can reach e using the external vertex f .

$C = \{a, b, c, d, e, f\}$ is a TUCC as B forms a CLOSED TUCC, f is able to reach every other vertex directly or via the external vertex g . However, C is not a TCC as a, b, c, e cannot reach f .

Figure 1: On the left a temporal graph, where on each edge e we depict $\lambda(e)$. Some of its components according to the non-strict model are reported on the right.

consider strictly increasing sequences, i.e. with $t_1 < \dots < t_k$. The former model is referred to as *non-strict* model, while the latter as *strict*. In both settings, we call such sequence a *temporal u, v -walk/path*, and we say that u reaches v . For instance, in Figure 1, both blue and green paths are valid in the non-strict model, but only the green one is valid in the strict model, as the blue one traverses two edges with label 2. The red path is not valid in both models.

The non-strict model is more appropriate in situations where the time granularity is relatively big. This is the case in a disease-spreading scenario [48], where the spreading speed might be unclear or in “time-varying graphs”, as in in [38], where a single snapshot corresponds to all the edges available in a time interval, e.g. the set of all the streets available in a day. As for the strict model, it can represent the connections of the public transportation network of a city which are available only at precise scheduled times. All in all, there is a rich literature on both models (see [24, 15, 42, 48, 27]), and this is why we explore both settings.

Connected Sets and Components. Given a temporal graph $\mathcal{G} = (G, \lambda)$, we say that $X \subseteq V(G)$ is a *temporal connected set* if u reaches v and v reaches u , for every $u, v \in X$. Extending the classical notion of connected components in static graphs, in [6] the authors define a *temporal connected component* (TCC for short) as a maximal connected set of \mathcal{G} . Such constraint can be strengthened to the existence of such paths using only vertices of X . Formally, X is a *closed temporal connected component* (CLOSED TCC for short) if, for every $u, v \in X$, we have that u reaches v and v reaches u through temporal paths contained in X . See Figure 1 for an example of TCC and CLOSED TCC.

Unilateral Connected Components. In the same fashion, also the concept of *unilateral connected components* can be extended to temporal graphs. In static graph theory, they are a well-studied relaxation of connected components which asks for a path from u to v or vice versa, for every pair u, v in the component [2, 9]. More formally, in a directed graph G , we say that $X \subseteq V(G)$ is a *unilateral connected set* if either u reaches v or v reaches u , for every $u, v \in X$. X is a *unilateral connected component* if it is maximal. In this paper, we introduce the definition of a *(closed) unilateral temporal connected set/component*, which can be seen as the immediate translation of unilateral connected component to the temporal context. Formally, $X \subseteq V(G)$ is a *temporal unilateral connected set* if u reaches v or v reaches u , for every $u, v \in X$, and it is a *closed unilateral connected set* if this holds using paths contained in X . Finally, a *(closed) temporal unilateral connected component* ((CLOSED) TUCC for short) is a maximal (closed) temporal unilateral connected set. See again Figure 1 for an example.

Problems. In this paper, we deal with four different definitions of temporal connected components, depending on whether they are unilateral or not, and whether they are closed or not. In what follows, we pose three questions, and we comment on partial knowledge about each of them. Later on, we discuss our results, which close almost all the gaps found in the literature. We start by asking the following.

Question 1 (Parameterized complexity). *Deciding the existence of temporal components of size at least k parameterized by (i) τ , i.e. the lifetime, (ii) k , and (iii) $k + \tau$.*

In order to answer Question 1 for the strict model, there is a very simple parameterized reduction from k -clique, known to be W[1]-hard when parameterized by k [22], to deciding the existence of connected components (both closed or not and both unilateral or not) of size at least k in undirected temporal graphs. This reduction has appeared in [12]. Given an undirected graph G , we can simply consider the temporal graph $\mathcal{G} = (G, \lambda)$ where $\lambda(uv) = \{1\}$ for all $uv \in E(G)$ (i.e., \mathcal{G} is equal to G itself). As u temporally reaches v if and only if $uv \in E(G)$, one can see that all those problems are now equivalent to deciding the existence of a k -clique in G . Observe that we get W[1]-hardness when parameterized by k or $k + \tau$, and para-NP-completeness when parameterized by τ , both in the undirected and the directed case.¹ However, this reduction does not work in the case of the *non-strict* model, leaving Question 1 open. Indeed the reductions in [6] and in [13] for (CLOSED) TCCs, which work indistinctly for both the strict or the non-strict models, are not parameterized reductions. We also observe that the aforementioned reductions work on the non-strict model only for $\tau \geq 4$.

Another question of interest is the following. Letting n be the number of vertices in \mathcal{G} and M be the number of *temporal edges*,² it is known that, in order to verify whether $X \subseteq V(G)$ is a connected set in \mathcal{G} , we can simply apply $O(n)$ single source “best” path computations (see e.g. [11, 46]), resulting in a time complexity of $O(n \cdot M)$. This is $O(M^2)$ if \mathcal{G} has no isolated vertices, a natural assumption when dealing with connectivity problems. As in static graphs testing connectivity can be done in linear time [28], we ask whether the described algorithm can be improved.

Question 2 (Lower bound on checking connectivity). *Given a temporal graph \mathcal{G} and a subset $X \subseteq V(\mathcal{G})$, what is the minimum running time required to check whether X is a (unilateral) connected set?*

Finally we focus on one last question.

Question 3 (Checking maximality). *Given a temporal graph \mathcal{G} and a subset $X \subseteq V(\mathcal{G})$, is it possible to verify, in polynomial time, whether X is a component, i.e. a maximal (closed) (unilateral) connected set?*

For Question 3, we first observe that the property of being a temporal (unilateral) connected set is hereditary (forming an independence system [33], see [19] for a survey about set systems), meaning that every subset of a (unilateral) connected set is still a (unilateral) connected set. For instance, in Figure 1, every subset of the connected set $B = \{a, b, c, d, e\}$ is a connected set. Also, checking whether $X' \subseteq V(G)$ is a temporal (unilateral) connected set can be done in time $O(n \cdot M)$, as discussed above. We can then check whether X is a maximal such set in time $O(n^2 \cdot M)$: it suffices

¹In the directed case, it suffices to replace each edge of the input graph with two opposite directed edges between the same endpoints.

² $M = \sum_{e \in E(\mathcal{G})} |\lambda(e)|$.

to test, for every $v \in V(G) \setminus X$, whether by adding v to X we still get a temporal (unilateral) connected set. On the other hand, *closed* connected (unilateral) sets are not hereditary, because by removing vertices from the set we could destroy the paths between other members of the set. This is the case for the closed connected set $A = \{a, b, c, d\}$ in Figure 1, since by removing d there are no temporal paths from c to a nor b anymore. This implies that the same approach as before does not work, i.e., we cannot check whether X is maximal by adding to X a *single* vertex at a time, then checking for connectivity. For instance, the closed connected set $A' = \{a, b\}$ in Figure 1 cannot be grown into the closed connected set A by adding one vertex at a time, since both $A' \cup \{c\}$ and $A' \cup \{d\}$ are not closed connected sets. Hence, the answer to Question 3 for closed sets does not seem easy, and until now was still open.

We remark the important practical consequences of the latter question. Indeed, in practice, when trying to find structures of maximum size, a usual viable strategy is modifying *backtracking listing algorithms* for efficient generating maximal structures (eventually with pruning strategies) and choosing the largest structures found [23, 10]. Such algorithms typically solve the so-called *extension problem*, that is generating all (or some of) the maximal solutions enlarging a partial one [30, 18, 3, 19]. Question 3 implicitly asks whether efficient generation of CLOSED TCCs or CLOSED TUCCs is likely to exist or not.

	PAR. τ	PAR. k	PAR. $k + \tau$
TCC	p-NP $\tau \geq 2$ (Th. 4)	W[1]-h Dir. $\tau \geq 2$ (Th. 6) and Undir. (Th. 5)	W[1]-h Dir. (Th. 6)
TUCC		W[1]-h Dir. $\tau \geq 2$ (Th. 6) FPT Undir. (Th. 8)	FPT Undir. (Th. 8)
CLOSED TCC		W[1]-h Dir. $\tau \geq 3$ (Th. 7)	W[1]-h Dir. (Th. 7)
CLOSED TUCC		W[1]-h Dir. $\tau \geq 3$ (Th. 7) FPT Undir. (Th. 8)	FPT Undir. (Th. 8)

Table 1: A summary of our results for the parameterized complexity of computing components of size at least k of a temporal graph \mathcal{G} having lifetime τ in the *non-strict* model. “W[1]-h” stands for W[1]-hardness and “pNP” stands for para-NP-completeness. For the strict model the entries are W[1]-h in the third and fourth columns and p-NP in the second one already for $\tau = 1$, both for the directed and the undirected case.

Our results. Our results concerning Question 1 are reported in Table 1 for the non-strict model, since for the strict model all the entries would be W[1]-hard or para-NP-complete already for $\tau = 1$, as we argued before. In the non-strict model, we observe instead that the situation is much more granulated. If $\tau = 1$, then all the problems become the corresponding traditional ones in static graphs, which are all polynomial (see Paragraph “Related works”). As for bigger values of τ , the complexity depends on the definition of component being considered, and whether the temporal graph is directed or not. Table 1 considers $\tau > 1$, reporting on negative results “ $\tau \geq x$ ” for some x meaning that the negative result starts to hold for temporal graphs of lifetime at least x .

The second column of Table 1 addresses Question 1(i), i.e., parameterization by τ . We prove that, for all the definitions of components being considered, the related problem becomes immediately para-NP-complete as soon as τ increases from 1 to 2; this is done in Theorem 4. This reduction improves upon the reduction of [6], which holds only for $\tau \geq 4$.

Question 1(ii) (parameterization by k) is addressed in the third column of Table 1. Considering first directed temporal graphs, we prove that all the problems are W[1]-hard. In particular, deciding the existence of a TCC or TUCC of size at least k is

W[1]-hard already for $\tau \geq 2$ (Theorem 6). As for the existence of closed components, W[1]-hardness also holds as long as $\tau \geq 3$ (Theorem 7). Observe that, since τ is constant in both results, these also imply the W[1]-hardness results presented in the last column, thus answering also Question 1(iii) (parameterization by $k + \tau$) for directed graphs. On the other hand, if the temporal graph is undirected, then the situation is even more granulated. Deciding the existence of a TCC of size at least k remains W[1]-hard, but only if τ is unbounded. This is complemented by the answer to Question 1(iii), presented in the last column of Table 1: TCC and (even) CLOSED TCC are FPT on undirected graphs when parameterized by $k + \tau$ (Theorem 8). We also give FPT algorithms when parameterized by k for unilateral components, namely TUCC and CLOSED TUCC. Observe how this differs from TCC, whose corresponding problem is W[1]-hard, meaning that unilateral and traditional components behave very differently when parameterized by k .

In summary, Table 1 answers Question 1 for almost all the definitions of components, both for directed and undirected temporal graphs, leaving open only the problems of, given an undirected temporal graph, deciding the existence of a CLOSED TCC of size at least k when parameterized by k , and solving the same problem for CLOSED TCC and CLOSED TUCC in directed temporal graphs where $\tau = 2$.

Concerning Questions 2 and 3, our results are summarized in Table 2. All these results hold both for the strict and the non-strict models. For Question 2, we prove that the trivial $O(M^2)$ algorithm to test whether S is a (closed) (unilateral) connected set is best possible, unless the Strong Exponential Time Hypothesis (SETH) fails [29]. For Question 3, in the case of TCC and TUCC, we have already seen that checking whether a set $X \subseteq V$ is a component can be done in $O(n^2 \cdot M)$. Interestingly, for CLOSED TCC and CLOSED TUCC, we answer negatively (unless P=NP) to Question 3.

	CHECK WHETHER $X \subseteq V$ IS A CONNECTED SET	CHECK WHETHER $X \subseteq V$ IS A COMPONENT
TCC	$\Theta(M^2)$ (Th. 9)	$O(n^2 \cdot M)$
TUCC		
CLOSED TCC		NP-c (Th. 10)
CLOSED TUCC		

Table 2: Our results for Question 2 and Question 3, holding for both the *strict* and the *non-strict* models. Recall that a component is a (inclusion-wise) maximal connected set. The $O(\cdot)$ result is easy and explained in the introduction. M (resp. n) denotes the number of temporal edges (resp. nodes) in \mathcal{G} .

Related work. The known reductions for temporal connected components in the literature [13, 6] which considers the non-strict setting are not parameterized and leave open the case when $\tau = 2$ or 3. The reductions we give here are parameterized (Theorems 6 and 7) and Theorem 4 closes also the cases $\tau = 2$ and 3. Furthermore, in [13] they show a series of interesting transformations but none of them allows us to apply known negative results for the strict model to the non-strict one.

There are many other papers about temporal connected components in the literature, including in [12], where they give an example where there can be an exponential number of temporal connected components in the strict model. In [38], the authors show that the problem of computing TCCs is a particular case of finding cliques in the so-called *affine graph*. This does not imply that the problem is NP-complete as claimed. Further related works include recent papers on giant components and connectivity in randomized temporal graphs [5, 16] and on networks with continuous varying-time [1].

Other notions of temporal components in the literature include *temporal out-component* (resp. *in-component*) in [37], Δ -component in [26], weakly connected

components in [37]. The latter applies if the temporal graph (G, λ) is directed and, as in the analogue case in static directed graphs, it simply ignore directions of the edges and consider the undirected version of the underlying graph G . In this paper we implicitly consider also weakly temporal connected components as studying how to compute them in directed temporal graphs is the same as studying how to compute (CLOSED) TCCs in undirected temporal graphs.

Finally, we remark that there are many results in the literature concerning unilateral components in static graphs, also with applications to community detection [34]. Even though the number of unilateral components in a graph is exponential [2], deciding whether there is one of size at least k is polynomial. In [2, Theorem 3], they prove that this corresponds to deciding whether a DAG with weights on its vertices has a path of weight at least k , which in turn can be done in polynomial time by slightly modifying the algorithm for longest paths [43]. Additionally, in [2, 17], the authors propose a listing algorithm, and in [4], a characterization of unilaterally connected graphs is presented. Further related works include [35], and [25]. There were no results about unilateral components in temporal graphs until now.

Preliminaries. For further formal definitions, we refer to Appendix A. For basic graph theory concepts and notation, we refer to [44], and to [21, 20] for basic background on parameterized complexity.

Structure of the Paper. In what follows, we present sketches of proofs of all results, with the results concerning Question 1 presented in Section 2, and the ones related to Questions 2 and 3 presented in Section 3. In Section 4, we present our concluding remarks.

2 Parameterized Complexity Results

This section is devoted to answer Question 1 and prove the results summarized in Table 1.

Parameterization by τ . We start by proving the result in the first column of Table 1 about para-NP-completeness wrt the lifetime τ , which applies to all the definitions of components. For (CLOSED) TCC, we do a reduction from MAXIMUM EDGE BICLIQUE (MEBP for short), which consists in, given a bipartite graph G and an integer k , deciding whether G has a biclique with at least k edges. It was proved to be NP-complete in [41]. Using the same construction, we prove hardness of (CLOSED) TUCC reducing from $2K_2$ -FREE EDGE SUBGRAPH, which consists in, given a bipartite graph G and an integer k , deciding whether G has a $2K_2$ -free subgraph with at least k edges. This was proved to be NP-complete in [47].

The main idea of the reductions is to generate a temporal graph \mathcal{G} whose base graph is the line graph L of a bipartite graph H with parts X, Y . We make active in timestep 1 every edge of a clique in L related to vertices in X , and in timestep 2 every edge of a clique related to vertices in Y . Doing so, we ensure that any pair of vertices of \mathcal{G} associated with a biclique in H reach one another in \mathcal{G} . We prove that there exists a biclique in H with at least k edges if and only if there exists a CLOSED TCC in \mathcal{G} of size at least k . The result extends to TCCs, as every TCC is also a CLOSED TCC. For the unilateral case, we can relax the biclique to a $2K_2$ -free graph since only one reachability relation is needed. As a result, we get the following, whose formal constructions and correctness are proven in Appendix C.1.

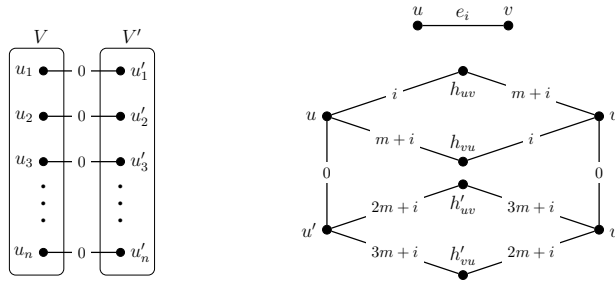


Figure 2: Construction used in the proof of Theorem 5. On the left, the two copies of $V(G)$ and the edges between them, active in timestep 0. On the right, the edge $e_i \in E(G)$ and the associated gadget in \mathcal{G} .

Theorem 4. *For every fixed $\tau \geq 2$ and given a temporal graph \mathcal{G} of lifetime τ and an integer k , it is NP-complete to decide if \mathcal{G} has a (CLOSED) TCC or a (CLOSED) TUCC of size at least k , even if the base graph of \mathcal{G} is the line graph of a bipartite graph.*

W[1]-hardness by k . We now focus on proving the W[1]-hardness results in the second column of Table 1 concerning parameterization by k , which also imply some of the results of the third column. The following W[1]-hardness results (Theorem 5, 7, and 6) are parameterized reductions from k -CLIQUE. The general objective is constructing a temporal graph \mathcal{G} in a way that vertices in \mathcal{G} are in the same component if and only if the corresponding nodes in the original graph are adjacent. Notice that we have to do this while: (i) ensuring that the size of the desired component is $f(k)$ for some computable function k (i.e., this is a parameterized reduction); and (ii) avoiding that the closed neighborhood of a vertex forms a component, so as to not have a false “yes” answer to k CLIQUE. To address these tasks, we rely on different techniques. The first reduction concerns TCC in undirected graphs and requires τ to be unbounded, as for τ bounded we show that the problem is FPT by $k + \tau$ (Theorem 8). The technique used is a parameterized evolution of the so-called *semaphore* technique used in [6, 13], which in general replaces edges by labeled diamonds to control paths of the original graph. However, while the original reduction gives labels in order to ensure that paths longer than one are broken, the following one allows the existence of paths longer than one. But if a temporal path from u to v exists for $uv \notin E(G)$, then the construction ensures the non-existence of temporal paths from v to u . Because of this property, the reduction does not extend to TUCCs, which we prove to be FPT when parameterized by k instead (Theorem 8). The interested reader can skip directly to Appendix C.2 for the complete proof.

Theorem 5. *Given a temporal graph \mathcal{G} and an integer k , deciding if \mathcal{G} has a TCC of size at least k is W[1]-hard with parameter k .*

Proof. We make a parameterized reduction from k -CLIQUE. Let G be graph and $k \geq 3$ be an integer. We construct the temporal graph $\mathcal{G} = (G', \lambda)$ as follows. See Figure 2 to follow the construction. First, add to G' every vertex in $V(G)$ and make $V = V(G)$. Second, add to G' a copy u' of every vertex $u \in V$ and define $V' = \{u' \mid u \in V\}$. Third, for every pair u, u' with $u \in V$ and $u' \in V'$ add the edge uu' to G' and make all such edges active at timestep 0. Fourth, consider an arbitrary ordering e_1, \dots, e_m of the edges of G and, for each edge $e_i = uv$, create four new vertices $\{h_{uv}, h_{vu}, h'_{uv}, h'_{vu} \mid uv \in E(G)\}$, adding edges:

- uh_{uv} and vh_{vu} , active at time i ;
- $u'h'_{uv}$ and $v'h'_{vu}$, active at time $2m + i$;
- $h_{vu}u$ and $h_{uv}v$, active at time $m + i$; and
- $h'_{vu}u'$ and $h'_{uv}v'$, active at time $3m + i$.

Denote the set $\{h_{uv}, h_{vu} \mid uv \in E(G)\}$ by H , and the set $\{h'_{uv}, h'_{vu} \mid uv \in E(G)\}$ by H' . We now prove that G has a clique of size at least k if and only if \mathcal{G} has a TCC of size at least $2k$. Given a clique C in G , it is easy to check that $C \cup \{u \in V' \mid u \in C\}$ is a TCC, and because of space constraints we present the formal argument only in Appendix C.2.

Now, let $S \subseteq V(G')$ be a TCC of \mathcal{G} of size at least $2k$. We want to show that either $C = \{u \in V(G) \mid u \in S \cap V\}$ or $C' = \{u \in V(G) \mid u' \in S \cap V'\}$ is a clique of G of size at least k . This part of the proof combines a series of useful facts, which we cannot include here due to space constraints. The full proof can be found Appendix C.2, and in what follows we present a sketch of it.

First, we argue that both C and C' are cliques in G . Then, by observing that the only edges between $V \cup H$ and $V' \cup H'$ are those incident to V and V' at timestep 0, we conclude that either $S \subseteq V \cup H$ or $S \subseteq V' \cup H'$. Since the cases are similar, we assume the former. If $|S \cap V| \geq k$, then C contains a clique of size at least k and the result follows. Otherwise, we define $E_S = \{uv \in E(G) \mid \{h_{uv}, h_{vu}\} \cap S \neq \emptyset\}$. That is, E_S is the set of edges of G related to vertices in $S \cap H$. We then prove the following claim.

Claim. Let $a, b \in S \cap H$ be associated with distinct edges g, g' of G sharing an endpoint v . If u and w are the other endpoints of g and g' , respectively, then u and w are also adjacent in G . Additionally, either $|S \cap \{h_{xy}, h_{yx}\}| \leq 1$ for every $xy \in E(G)$, or $|S \cap H| \leq 2$.

To finish the proof, we first recall that we are in the case $|S \cap H| \geq k + 1$. By our assumption that $k \geq 3$, note that the above claim gives us that $|S \cap \{h_{xy}, h_{yx}\}| \leq 1$ for every $xy \in E(G)$, which in turn implies that $|E_S| = |S \cap H|$. Additionally, observe that, since $|S \cap H| \geq 4$, the same claim also gives us that there must exist $w \in V$ such that e is incident to w for every $e \in E_S$. Indeed, the only way that 3 distinct edges can be mutually adjacent without being all incident to a same vertex is if they form a triangle. Supposing that 3 edges in E_S form a triangle $T = (a, b, c)$, since $|E_S| \geq 4$, there exists an edge $e \in E_S \setminus E(T)$. But now, since G is a simple graph, e is incident to at most one between a, b and c , say a . We get a contradiction wrt the aforementioned claim as in this case e is not incident to edge $bc \in E_S$. Finally, by letting $C'' = \{v_1, \dots, v_k\}$ be any choice of k distinct vertices such that $\{wv_1, \dots, wv_k\} \subseteq E_S$, our claim gives us that v_i and v_j are adjacent in G , for every $i, j \in [k]$; i.e., C'' is a k -clique in G . \square

The following result concerns TCC and TUCC in directed temporal graphs. It is important to remark that for TCC and τ unbounded, we already know that the problem is W[1]-hard because of Theorem 5 which holds for undirected graphs and extends to directed ones. However, the following reduction applies specifically for directed ones already for $\tau = 2$. The technique used here is the previously mentioned semaphore technique, made parameterized by exploiting the direction of the edges. For the sake of space, the construction is shown only in Figure 3, which shows how to obtain the temporal graph \mathcal{G} in Figure 3(b) from the graph G in Figure 3(a) in a way that, for every integer $k \geq 3$, graph G has a clique of size at least k if and

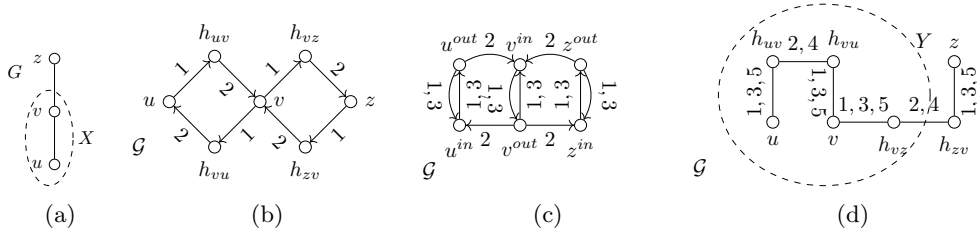


Figure 3: Examples for some of our reductions. Given the graph in (a), Theorem 6 constructs the directed temporal graph in (b), Theorem 7 constructs the directed temporal graph in (c), and, given additionally set X in (a), Theorem 10 constructs the temporal graph \mathcal{G} and set Y in (d).

only if \mathcal{G} has a TCC of size at least k . The formal construction and proof are given in Appendix C.3

Theorem 6. *Given a directed temporal graph \mathcal{G} and an integer k , deciding if \mathcal{G} has a TCC of size at least k is $W[1]$ -hard with parameter k , even if \mathcal{G} has lifetime 2. The same holds for TUCC.*

The next result concerns closed TCCs and TUCCs. In this case, we also reduce from k -CLIQUE, but we cannot apply the semaphore technique as before. Indeed, as we are dealing with closed components, nodes must be reachable using vertices inside the components, while the semaphore technique would make them reachable via additional nodes, which do not necessarily reach each other. For this reason, in the following we introduce a new technique subdividing nodes, instead of edges, in order to break paths of the original graph of length longer than one, being careful to allow that these additional nodes reach each other. The construction is shown in Figure 3, which shows how to construct temporal graph \mathcal{G} in Figure 3(c), given graph G in Figure 3(a) in a way that graph G has a clique of size k if and only if \mathcal{G} has a CLOSED TCC (TUCC) of size at least $2k$. The formal construction and proof are given in Appendix C.4.

Theorem 7. *Given a directed temporal graph \mathcal{G} and an integer k , deciding if \mathcal{G} has a CLOSED TCC of size at least k is $W[1]$ -hard with parameter k , even if \mathcal{G} has lifetime 3. The same holds for CLOSED TUCC.*

FPT algorithms. We now show our FPT algorithms to find (CLOSED) TCCs and (CLOSED) TUCCs in undirected temporal graphs, as for directed temporal graphs we have proved $W[1]$ -hardness. In particular, we prove the following result, whose proof is shown in Appendix C.5.

Theorem 8. *Given a temporal graph $\mathcal{G} = (G, \lambda)$ on n vertices and with lifetime τ , and a positive integer k , there are algorithms running in time*

1. $O(k^{k \cdot \tau} \cdot n)$ that decides whether there is a TCC of size at least k ;
2. $O(2^{k^\tau} \cdot n)$ that decides whether there is a CLOSED TCC of size at least k ;
3. $O(k^{k^2} \cdot n)$ that decides whether there is a TUCC of size at least k ; and
4. $O(2^{k^k} \cdot n)$ that decides whether there is a CLOSED TUCC of size at least k .

Sketch. The *reachability digraph* R associated to \mathcal{G} is a directed graph with the same vertex set as \mathcal{G} , and such that uv is an edge in R if and only if u reaches v in \mathcal{G} and $u \neq v$. This is related to the *affine* graph in [38]. Observe that finding a TCC (TUCC) in \mathcal{G} of size at least k is equivalent to finding a set $S \subseteq V(\mathcal{G})$ in R of size *exactly* k such that $uv \in E(R)$ and (or) $vu \in E(R)$ for every pair $u, v \in V(R)$. As for finding a CLOSED TCC (CLOSED TUCC), we need to have the same property, except that all subsets of size *at least* k must be tested (recall that being a closed connected (unilateral) set is not hereditary). Therefore, if $\Delta = \Delta(R)$, then testing connectivity takes time $O(k^\Delta \cdot n)$ (it suffices to test all subsets of size $k - 1$ in $N(u)$, for all $u \in V(R)$), while testing closed connectivity takes time $O(2^\Delta \cdot n)$ (it suffices to test all subsets of size *at least* $k - 1$ in $N(u)$, for all $u \in V(R)$). The proofs then consist in bounding the value Δ in each case. \square \square

It is important to observe that, for unilateral components, these bounds depend only on k , while for TCCs and CLOSED TCCs they depend on both k and τ . This is consistent with the fact that we have proved that for TCC the problem is $W[1]$ -hard when parameterized just by k (Theorem 5).

3 Checking Connectivity and Maximality

This section is focused on Questions 2 and Question 3. The former is open for all definitions of components for both the strict and the non-strict models. We answer to the question providing the following conditional lower bound, which holds for both models, where the notation $\tilde{O}(\cdot)$ ignores polylog factors. Due to space constraints, its proof is shown in Appendix D. We apply the technique used for instance in [7, 40, 45] to prove lower bounds for polynomial problems, falling within the fine-grained complexity framework.

Theorem 9. *Consider a temporal graph \mathcal{G} on M temporal edges. There is no algorithm running in time $\tilde{O}(M^{2-\epsilon})$, for some ϵ , that decides whether G is temporally (unilaterally) connected, unless SETH fails.*

We now focus on Question 3. We prove the results in the second column of Table 2, about the problem of deciding whether a subset of vertices Y of a temporal graph is a component, i.e. a maximal connected set. The question is open both for the strict and the non-strict model. We argued already in the introduction that this is polynomial for TCC and TUCC for both models. In the following we prove NP-completeness for CLOSED TCC and CLOSED TUCC on undirected graphs. The results extend to directed graphs as well.

Theorem 10. *Let \mathcal{G} be a (directed) temporal graph, and $Y \subseteq V(\mathcal{G})$. Deciding whether Y is a CLOSED TCC is NP-complete. The same holds for CLOSED TUCC.*

Proof. We reduce from the problem of deciding whether a subset of vertices X of a given a graph G is a maximal 2-club, where a 2-club is a set of vertices C such that $G[C]$ has diameter at most 2. This problem has been shown to be NP-complete in [39]. Let us first focus on the strict model. In this case, given G we can build a temporal graph \mathcal{G} with only two snapshots, both equal to G . Observe that X is a 2-club in G if and only if X is a CLOSED TCC in \mathcal{G} . Indeed, because we can take only one edge in each snapshot and $\tau = 2$, we get that temporal paths will always have length at most 2. This also extends to CLOSED TUCC by noting that all paths in \mathcal{G} can be temporally traversed in both directions.

In the case of the non-strict model, the situation is more complicated as in each snapshot we can take an arbitrary number of edges resulting in paths arbitrarily long. We show the construction for CLOSED TCC in what follows, proving its correctness in Appendix E. Let \mathcal{G} be obtained from G by subdividing each edge uv of G twice, creating vertices h_{uv} and h_{vu} , with $\lambda(uh_{uv}) = \lambda(vh_{vu}) = \{1, 3, 5\}$, and $\lambda(h_{uv}h_{vu}) = \{2, 4\}$. See Figure 3 (d) for an illustration.

Given (G, X) , the instance of maximal 2-club, we prove that X is a maximal 2-club in G iff $Y = X \cup N_H(X)$ is a CLOSED TCC in \mathcal{G} . For this, it suffices to prove that, given $X' \subseteq V(G)$ and defining Y' similarly as before w.r.t. X' , we have that $G[X']$ has diameter at most 2 iff Y' is a closed temporal connected set. The proof extends to CLOSED TUCC by proving that every CLOSED TCC is also a CLOSED TUCC and vice-versa. \square

4 Concluding remarks

In this paper, we revisit the notion of connected components in temporal graphs introduced in [6] from the point of view of parameterized complexity. We then consider unilateral connectivity in temporal graphs, and investigate all related problems, in both the strict and the non-strict setting, as well as both for directed and undirected temporal graphs, parameterizing by the size k of the desired component, the lifetime τ of the considered (directed) temporal graph \mathcal{G} , and by $k + \tau$. We classify all possible entries in Table 1, leaving open just the following questions.

Question 11. *Given an undirected temporal graph \mathcal{G} , and considering parameterization by k , the size of the searched component, what is the complexity of deciding the existence of a CLOSED TCC?*

Question 12. *Given a directed temporal graph \mathcal{G} with lifetime 2, and considering parameterization by k , the size of the searched component, what is the complexity of deciding the existence of a CLOSED TCC (TUCC)?*

We additionally prove a lower bound for testing connectivity, and prove that deciding maximality of closed (unilateral) connectivity is NP-complete.

References

- [1] Eleni C Akrida and Paul G Spirakis. On verifying and maintaining connectivity of interval temporal networks. *Parallel Processing Letters*, 29(02):1950009, 2019.
- [2] Eshrat Arjomandi. On finding all unilaterally connected components of a digraph. *Information Processing Letters*, 5(1):8–10, 1976.
- [3] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete applied mathematics*, 65(1-3):21–46, 1996.
- [4] Jørgen Bang-Jensen and Gregory Z Gutin. *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.
- [5] Ruben Becker, Arnaud Casteigts, Pierluigi Crescenzi, Bojana Kodric, Malte Renken, Michael Raskin, and Viktor Zamaraev. Giant components in random temporal graphs. *CoRR*, abs/2205.14888, 2022.

- [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-NOW 2003 Montreal, Canada, October 8-10, 2003, Proceedings*, pages 259–270, 2003.
- [7] Michele Borassi, Pierluigi Crescenzi, and Michel Habib. Into the square: On the complexity of some quadratic-time solvable problems. In *ICTCS*, volume 322 of *Electronic Notes in Theoretical Computer Science*, pages 51–67. Elsevier, 2015.
- [8] Pierre Borgnat, Eric Fleury, Jean-Loup Guillaume, Clémence Magnien, Céline Robardet, and Antoine Scherrer. Evolving networks. In *Mining Massive Data Sets for Security*, pages 198–203, 2007.
- [9] Allan B Borodin and Ian Munro. Notes on efficient and optimal algorithms. Technical report, U. of Toronto and U. of Waterloo, Canada, 1972.
- [10] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.
- [11] Marco Calamai, Pierluigi Crescenzi, and Andrea Marino. On computing the diameter of (weighted) link streams. *ACM J. Exp. Algorithmics*, 27:4.3:1–4.3:28, 2022.
- [12] Arnaud Casteigts. Finding structure in dynamic networks. *arXiv preprint arXiv:1807.07801*, 2018.
- [13] Arnaud Casteigts, Timothée Corsini, and Writika Sarkar. Simple, strict, proper, happy: A study of reachability in temporal graphs. *arXiv preprint arXiv:2208.01720*, 2022.
- [14] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.
- [15] Arnaud Casteigts, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. Finding temporal paths under waiting time constraints. *Algorithmica*, 83(9):2754–2802, 2021.
- [16] Arnaud Casteigts, Michael Raskin, Malte Renken, and Viktor Zamaraev. Sharp thresholds in random simple temporal graphs. In *FOCS*, pages 319–326. IEEE, 2021.
- [17] Grant A. Cheston. A correction to a unilaterally connected components algorithm. *Inf. Process. Lett.*, 7(2):125, 1978.
- [18] Alessio Conte, Roberto Grossi, Mamadou Moustapha Kanté, Andrea Marino, Takeaki Uno, and Kunihiro Wasa. Listing induced steiner subgraphs as a compact way to discover steiner trees in graphs. In *MFCS*, volume 138 of *LIPICs*, pages 73:1–73:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- [19] Alessio Conte, Roberto Grossi, Andrea Marino, and Luca Versari. Listing maximal subgraphs satisfying strongly accessible properties. *SIAM J. Discret. Math.*, 33(2):587–613, 2019.

- [20] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [21] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- [22] Rod G Downey and Michael R Fellows. Fixed-parameter tractability and completeness ii: On completeness for w [1]. *Theoretical Computer Science*, 141(1-2):109–131, 1995.
- [23] John D Eblen, Charles A Phillips, Gary L Rogers, and Michael A Langston. The maximum clique enumeration problem: algorithms, applications, and implementations. In *BMC bioinformatics*, volume 13, pages 1–11. Springer, 2012.
- [24] Jessica A. Enright, Kitty Meeks, and Hendrik Molter. Counting temporal paths. *CoRR*, abs/2202.12055, 2022.
- [25] John Frederick Fink and Linda Lesniak-Foster. Graphs for which every unilateral orientation is traceable. *Ars. Combinatoria*, 9:113–118, 1980.
- [26] Carlos Gómez-Calzado, Arnaud Casteigts, Alberto Lafuente, and Mikel Larrea. A connectivity model for agreement in dynamic systems. In *European Conference on Parallel Processing*, pages 333–345. Springer, 2015.
- [27] Roman Haag, Hendrik Molter, Rolf Niedermeier, and Malte Renken. Feedback edge sets in temporal graphs. *Discrete Applied Mathematics*, 307:65–78, 2022.
- [28] John Hopcroft and Robert Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.
- [29] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.
- [30] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, Lhouari Nourine, and Takeaki Uno. A polynomial delay algorithm for enumerating minimal dominating sets in chordal graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 138–153. Springer, 2015.
- [31] David Kempe, Jon M. Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 504–513. ACM, 2000.
- [32] Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(1):1–29, 2018.
- [33] Eugene L. Lawler, Jan Karel Lenstra, and AHG Rinnooy Kan. Generating all maximal independent sets: Np-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9(3):558–565, 1980.
- [34] Vincent Levorato and Coralie Petermann. Detection of communities in directed networks based on strongly p-connected components. In *2011 international conference on computational aspects of social networks (CASoN)*, pages 211–216. IEEE, 2011.

- [35] Tamara Mchedlidze and Antonios Symvonis. Unilateral orientation of mixed graphs. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 588–599. Springer, 2010.
- [36] Othon Michail. An introduction to temporal graphs: An algorithmic perspective. *Internet Mathematics*, 12(4):239–280, 2016.
- [37] Vincenzo Nicosia, John Tang, Cecilia Mascolo, Mirco Musolesi, Giovanni Russo, and Vito Latora. Graph metrics for temporal networks. In *Temporal networks*, pages 15–40. Springer, 2013.
- [38] Vincenzo Nicosia, John Tang, Mirco Musolesi, Giovanni Russo, Cecilia Mascolo, and Vito Latora. Components in time-varying graphs. *Chaos: An interdisciplinary journal of nonlinear science*, 22(2):023101, 2012.
- [39] Foad Mahdavi Pajouh and Balabhaskar Balasundaram. On inclusionwise maximal and maximum cardinality k -clubs in graphs. *Discret. Optim.*, 9(2):84–97, 2012.
- [40] Mihai Pătraşcu and Ryan Williams. On the possibility of faster sat algorithms. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1065–1075. SIAM, 2010.
- [41] René Peeters. The maximum edge biclique problem is np-complete. *Discrete Applied Mathematics*, 131(3):651–654, 2003.
- [42] Maciej Rymar, Hendrik Molter, André Nichterlein, and Rolf Niedermeier. Towards classifying the polynomial-time solvability of temporal betweenness centrality. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 219–231. Springer, 2021.
- [43] Robert Sedgewick and Kevin Wayne. *Algorithms: Part I*. Addison-Wesley Professional, 2014.
- [44] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.
- [45] Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 645–654. IEEE, 2010.
- [46] Huanhuan Wu, James Cheng, Silu Huang, Yiping Ke, Yi Lu, and Yanyan Xu. Path problems in temporal graphs. *Proceedings of the VLDB Endowment*, 7(9):721–732, 2014.
- [47] Mihalis Yannakakis. Computing the minimum fill-in is np-complete. *SIAM Journal on Algebraic Discrete Methods*, 2(1):77–79, 1981.
- [48] Philipp Zschoche, Till Fluschnik, Hendrik Molter, and Rolf Niedermeier. The complexity of finding small separators in temporal graphs. *Journal of Computer and System Sciences*, 107:72–92, 2020.

A Preliminaries and notation

Given a graph $G = (V, E)$, directed or not, and a set $X \subseteq V(G)$ we write $G[X]$ for the subgraph of G induced by X . If e is an edge of a directed or undirected graph with *endpoints* u and v , we may refer to e as (u, v) and say that e is *incident* to u and v . If e is an edge³ from u to v of a directed graph, we say that e has *tail* u , *head* v , and is *oriented* from u to v . The *degree* $d_G(v)$ of a vertex v of a (directed) graph G is the number of edges of G incident to v . We denote by $\Delta(G)$ the maximum degree of a vertex of G . The *neighborhood* $N_G(v)$ of v is the set $\{u \in V(G) \mid (u, v) \in E(G)\}$. If D is a digraph, the *in-neighborhood* $N_D^-(v)$ of v is the set $\{u \in V(D) \mid (u, v) \in E(D)\}$, and the *out-neighborhood* $N_D^+(v)$ is the set $\{u \in V(D) \mid (v, u) \in E(D)\}$.

An undirected graph G is said to be *simple* if there is at most one edge between every pair of vertices of G (i.e., there are no *parallel* edges). We say that G is *bipartite* if there is a partition of $V(G)$ into two non-empty sets X and Y such that every edge of G has one endpoint in X and the other endpoint in Y . A *matching* of G is a set of edges $M \subseteq E(G)$ such that no two edges of M share an endpoint; i.e., they all are pairwise *independent*. This definition also applies to oriented edges in directed graphs. A *clique* in an graph H is a subset $C \subseteq V(H)$ such that all vertices of C are pairwise adjacent. If H is directed, we say that C is a clique (resp. *full clique*) if one of (resp. both) the two possible edges exist between u and v , for every pair $u, v \in C$, $u \neq v$. A *biclique* in a bipartite graph H is a disjoint pair of sets $A, B \subseteq V(H)$ such that there is an edge from every $a \in A$ to every $b \in B$. A graph is *$2K_2$ -free* if it does not contain a pair of edges uv and xy such that $G[\{u, v, x, y\}]$ contains exactly the two edges (i.e., is isomorphic to a $2K_2$).

A *walk* in a (directed) graph G is an alternating sequence W of vertices and edges that starts and ends with a vertex, and such that for every edge (u, v) in the walk, vertex u (resp. vertex v) is the element right before (resp. right after) edge (u, v) in W . If the first vertex in a walk is u and the last one is v , then we say this is a *walk from u to v* . A *path* is a (directed) graph containing exactly a walk that contains all of its vertices and edges without repetition. A directed graph D is *strongly connected* if, for every pair of vertices $u, v \in V(D)$, there is a walk from u to v and a walk from v to u in D . We say that D is *weakly connected* if the underlying graph of D is connected. A *strong component* of D is a maximal induced subgraph of D that is strongly connected, and a *weak component* of D is a maximal induced subgraph of D that is weakly connected. For simplicity, we call a strong connected component of a directed graph simply “connected component”.

An *orientation* of an undirected graph G is a digraph D obtained from G by choosing an orientation for each edge $e \in E(G)$. The undirected graph G formed by ignoring the orientation of the edges of a digraph D is the *underlying graph* of D .

The *line graph* of an undirected graph G is the graph L with vertex set $E(G)$ where two vertices $e, f \in V(L)$ associated with edges of G are linked by an edge if and only if e and f share an endpoint in G .

For an undirected (static) graph G and $u \in V(G)$, let $\delta(u)$ denote the set $\{e \in E(G) \mid u \in e\}$. For a set $S \subseteq E(G)$, the *edge-induced subgraph* $G[S]$ is a graph whose edge set is S and vertex set consists of all endpoints of the edges in S .

Given a temporal (directed) graph $\mathcal{G} = (G, \lambda)$ and a subset $S \subseteq V(G)$, we say that S is *temporal connected* (in \mathcal{G}) if there is a temporal u, v -path in \mathcal{G} for every ordered pair $(u, v) \in S \times S$.

³We refer to arcs of directed graphs as edges (following the notation in [44])

A.1 Parameterized complexity

A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$. For an instance $I = (x, k) \in \Sigma^* \times \mathbb{N}$, k is called the *parameter*.

A parameterized problem L is *fixed-parameter tractable* (FPT) if there exists an algorithm \mathcal{A} , a computable function f , and a constant c such that given an instance $I = (x, k)$, \mathcal{A} (called an *FPT algorithm*) correctly decides whether $I \in L$ in time bounded by $f(k) \cdot |I|^c$. For instance, the VERTEX COVER problem parameterized by the size of the solution is FPT.

A parameterized problem L is in XP if there exists an algorithm \mathcal{A} and two computable functions f and g such that given an instance $I = (x, k)$, \mathcal{A} (called an *XP algorithm*) correctly decides whether $I \in L$ in time bounded by $f(k) \cdot |I|^{g(k)}$. For instance, the CLIQUE problem parameterized by the size of the solution is in XP.

Within parameterized problems, the class W[1] may be seen as the parameterized equivalent to the class NP of classical decision problems. Without entering into details (see [21, 20] for the formal definitions), a parameterized problem being W[1]-hard can be seen as a strong evidence that this problem is *not* FPT. CLIQUE parameterized by the size of the solution is the canonical example of a W[1]-hard problem.

Parameterized reductions are used to transfer fixed-parameter tractability or hardness between parameterized problems. Namely, a parameterized reduction is an algorithm that, given an instance (x, k) of a parameterized problem L , runs in time $f(k) \cdot |x|^{O(1)}$ and outputs an instance (x', k') of a parameterized problem L' such that $k' \leq g(k)$ for some computable function g and (x, k) is positive if and only if (x', k') is positive. For example, if L is W[1]-hard and there is a parameterized reduction from L to L' , then L' is also W[1]-hard and thus unlikely to admit an FPT algorithm.

A.2 Temporal paths and components

Given a temporal (directed) graph $\mathcal{G} = (G, \lambda)$ and vertices $v_0, v_q \in V(G)$, a *temporal v_0, v_q -walk* in \mathcal{G} is defined as a sequence of vertices and temporal edges, $(v_0, \alpha_1, v_1, \dots, \alpha_q, v_q)$ such that, for each $i \in [q]$, α_i has endpoints $v_{i-1}v_i$ and is active in a timestep t_i which is at most equal to the timestep where α_{i+1} is active. Sometimes we abuse notation and write $P = (v_0, t_1, v_1, \dots, t_q, v_q)$ instead, where t_i is equal to the timestep where α_i is active, for every $i \in [q]$. We then say that P *starts in time* t_1 and *finishes in time* t_q . Given $i, j \in \{0, \dots, q\}$, we denote by $v_i P v_j$ the v_i, v_j -walk $(v_i, t_{i+1}, v_{i+1}, \dots, t_j, v_j)$. Additionally, if no vertices of G are repeated in P , then we say that P is a *temporal v_0, v_q -path*. It is important to mention that distinctions between paths and walks are important for some problems, but since it is not the case in this work, the reader should not worry about interchangeable uses of “walks” and “paths” along the text. Given two vertices $u, v \in V(G)$, we say that u *reaches* v in \mathcal{G} if there exists a temporal u, v -walk in \mathcal{G} .

Given a temporal (directed) graph $\mathcal{G} = (G, \lambda)$ and a subset $S \subseteq V(G)$, we say that S is *temporal connected* (in \mathcal{G}) if there is a temporal u, v -path in \mathcal{G} for every ordered pair $(u, v) \in S \times S$. In [6], the authors define a *temporal connected component* (TCC for short) as a maximal subset $S \subseteq V(G)$ such that S is temporal connected. Similarly, a *closed temporal connected component* (CLOSED TCC for short) was defined as a maximal subset $S \subseteq V(G)$ for which, for every ordered pair $(u, v) \in S \times S$, there is a temporal u, v -path in \mathcal{G} using only vertices of S . In other words, a CLOSED TCC is a maximal subset $S \subseteq V(G)$ such that $\mathcal{G}[S]$ (the temporal subgraph induced by S) is temporal connected. We say that S is a *temporal unilaterally connected set* if for every pair $u, v \in S$ there is a temporal u, v -path or a temporal v, u -path in \mathcal{G} . If all such paths use only vertices in S then we say that S is a *closed temporal unilaterally*

connected set. If S is maximal such set, then we say that S is a *temporal unilaterally connected component* (TUCC) in the first case, and a *closed temporal unilaterally connected component* (CLOSED TUCC) in the second case.

The *reachability digraph* $R(\mathcal{G})$ associated to \mathcal{G} is a directed graph with the same vertex set as \mathcal{G} , and such that uv is an edge in $R(\mathcal{G})$ if and only if u reaches v in \mathcal{G} , $u \neq v$. This is related to the *affine graph* in [38]. This is a slight generalization of the *affine graph* introduced in [38]. There, since they are interested only in the TCC variants, they consider pairs that are mutually reachable from each other, ignoring the edges uv of $R(\mathcal{G})$ that are not symmetric (i.e., for which vu is not present).

B Auxiliary results

The following result is an immediate consequence of the definition of reachability graph $R(\mathcal{G})$.

Lemma 13. *Given a temporal (directed) graph $\mathcal{G} = (G, \lambda)$, then the following hold:*

1. C is a TCC in \mathcal{G} if and only if C is a maximal full clique in $R(\mathcal{G})$;
2. C is a TUCC in \mathcal{G} if and only if C is a maximal clique in $R(\mathcal{G})$;
3. If C is a CLOSED TCC in \mathcal{G} , then C is a full clique in $R(\mathcal{G})$; and
4. If C is a CLOSED TUCC in \mathcal{G} , then C is a clique in $R(\mathcal{G})$.

For each $i \in [\tau]$ and $u \in V(\mathcal{G})$, denote by $C_i(u)$ the set of vertices in the same connected component of G_i as u , and by $R_i(u)$ the set of vertices in G_i reachable from u (i.e., $v \in R_i(u)$ if and only if there is a u, v -path in G_i). Observe that, if G is undirected, then $C_i(u) = R_i(u)$. Note also that $\mathcal{R}_\tau(u)$ is exactly equal to $N_{R(\mathcal{G})}^-(u)$. For the sake of completeness, we now show that we can recursively define the set of vertices reachable from u by a temporal path finishing at time at most i . We apply the following lemma in the context of non-strict reachability, but also holds for strict.

Lemma 14. *Let \mathcal{G} be a (directed) temporal graph, and let $\mathcal{R}_i(u)$ be recursively defined as:*

$$\mathcal{R}_i(u) = \begin{cases} R_1(u) & , \text{ if } i = 1 \\ \bigcup_{v \in \mathcal{R}_{i-1}(u)} R_i(v) & , \text{ otherwise} \end{cases}$$

Then $\mathcal{R}_i(u)$ is equal to the set of vertices reachable from u by a temporal path finishing at time at most i .

Proof. We want to prove that $v \in \mathcal{R}_i(u)$ if and only if there exists a temporal u, v -walk finishing in time at most i . First, let $v \in \mathcal{R}_i(u)$. If $i = 1$, then $v \in R_1(u)$ and u reaches v in G_1 by definition. So suppose $i > 1$. Again by definition, we have $v \in \bigcup_{v' \in \mathcal{R}_{i-1}(u)} R_i(v')$. Consider then $w \in \mathcal{R}_{i-1}(u)$ such that $v \in R_i(w)$. By induction hypothesis, there exists a temporal u, w -path P finishing in time at most $i - 1$. And because w reaches v in G_i , such path can be extended to a temporal u, v -walk finishing in time at most i .

Now, let v be such that there exists a temporal u, v -path P finishing in time at most i . If P finishes in time at most $i - 1$, we are done by induction hypothesis. Otherwise, let $w \in V(P)$ be closest to v in P such that the temporal edges incident to w in P occur in time $j < i$ and i . Observe that wPv is contained in G_i , and hence $v \in R_i(w)$. Additionally, uPw finishes in time at most $j \leq i - 1$, and by induction hypothesis $w \in \mathcal{R}_{i-1}(u)$. By definition we then get $v \in \mathcal{R}_i(u)$, as we wanted to show. \square

The following easy proposition tells us that deciding the existence of large components (i.e. maximal connected sets) is equivalent to deciding the existence of large connected sets.

Proposition 15. *Let \mathcal{G} be a temporal (directed) graph. Then \mathcal{G} has a (closed) temporal (unilaterally) connected component of size at least k if and only if (closed) temporal (unilaterally) connected set of size at least k .*

C Parameterized Complexity Results: Proofs

C.1 Proof of Theorem 4

The MAXIMUM EDGE BICLIQUE Problem (MEBP for short) consists in, given a bipartite graph G and an integer k , deciding whether G has a biclique with at least k edges. It was proved to be NP-complete in [41]. Problem $2K_2$ -FREE EDGE SUBGRAPH consists in, given a bipartite graph G and an integer k , deciding whether G has a $2K_2$ -free subgraph with at least k edges. This was proved to be NP-complete in [47].

For an undirected (static) graph G and $u \in V(G)$, let $\delta(u)$ denote the set $\{e \in E(G) \mid u \in e\}$. And for a set $S \subseteq E(G)$, the *edge-induced subgraph* $G[S]$ is a graph whose edge set is S and vertex set consists of all endpoints of the edges in S .

As stated in the main text, we prove NP-completeness of (CLOSED) TCC and (CLOSED) TUCC using the same construction and reducing from the above problems. We start by proving hardness of (CLOSED) TCC.

Proof. Consider an instance (H, k) of MEBP, consisting of a bipartite graph $H = (X \cup Y, E)$ and an integer k . Let $X = \{x_1, \dots, x_p\}$ and $Y = \{y_1, \dots, y_q\}$. We construct a temporal graph $\mathcal{G} = (G, \lambda)$ with lifetime 2 such that $V(G) = E(H)$ and snapshot G_1 is the graph whose connected components are precisely $\delta(x_i)$ for each $i \in \{1, \dots, p\}$, while snapshot G_2 is the graph whose connected components are precisely $\delta(y_j)$ for each $j \in \{1, \dots, q\}$. We consider that the components are cliques; clearly G is the line graph of H . We claim that there exists a biclique (A, B) in H with at least k edges if and only if there exists a CLOSED TCC in \mathcal{G} of size at least k . Then we prove that every TCC is also a CLOSED TCC, finishing this part of the proof.

Suppose first that there exists a biclique (A, B) in H with at least k edges, and let $C = E(H[A, B])$. We want to show that C is a closed temporal connected set of \mathcal{G} . Let $e = xy$ and $e' = x'y'$ be two elements of C , with $\{x, x'\} \subseteq X$ and $\{y, y'\} \subseteq Y$. If $x = x'$ then $e, e' \in \delta(x)$ and hence are contained in the same component of G_1 (i.e., they reach each other by a direct edge); the analogous holds in case $y = y'$, so suppose $x \neq x'$ and $y \neq y'$. Since (A, B) is a biclique in H , we have that $\{xy, xy', x'y, x'y'\} \subseteq C$. Denote xy' by f and $x'y$ by f' . Now, in \mathcal{G} we can reach f from e at timestep 1 and e' from f at timestep 2. Similarly, we can also reach e from e' in \mathcal{G} . Because f, f' are also in C , and since this holds for any two such edges, we get that C is a closed temporal connected set, and by Proposition 15, we get that \mathcal{G} has a CLOSED TCC of size at least k .

For the converse, suppose that \mathcal{G} has a TCC C with $|C| \geq k$. We want to show that C forms a biclique in H with at least k edges. Let $A \subseteq X$ contain all vertices incident to some $e \in C$, and define B similarly with relation to Y . First we show that (A, B) is a biclique in H . Observe that, combined with the previous paragraph, we get that any TCC is also a CLOSED TCC; hence the proof will follow also for both problems. Let $x \in A$ and $y \in B$. We need to show that xy is an edge of H . Note that since $x \in A$, it must be an endpoint of some edge $e_x \in C$; analogously, since $y \in B$, it must be an endpoint of some edge $e_y \in C$. Let y' be the other endpoint of e_x and let

x' be the other endpoint of e_y . Since H is bipartite, we have $x' \in A$ and $y' \in B$. If $x' = x$ or $y' = y$ we are done, so suppose otherwise. Since e_x and e_y are in C , there exists a temporal e_x, e_y -path. Note that, by the construction of \mathcal{G} , this means that there exists an edge $e' \in \delta(x) \cap \delta(y)$. The only possibility is $e' = xy$, as we wanted. To finish the proof just observe that $|E(H[A, B])| \geq |C| \geq k$.

For (CLOSED) TUCC, we make a reduction from $2K_2$ -FREE EDGE SUBGRAPH. This was shown to be equivalent to the MINIMUM FILL-IN problem in co-bipartite graphs in [47], where the authors also showed that this problem is NP-complete. Given a bipartite H , the proof follows similarly to the first case, using exactly the same construction for the temporal graph \mathcal{G} .

Assume that H contains a $2K_2$ -free subgraph H' with at least k edges. Given edges $e = xy$ and $e' = x'y'$ of H' , since H' is $2K_2$ -free either $xy' \in E(H')$, or $x'y \in E(H')$, or both. Thus in \mathcal{G} there is a temporal path from e to e' , or a temporal path from e to e' , or both. Since in this case we only care about unilateral components, the first part of the proof follows. The second part also follows with similar arguments, just noticing that an unilateral component only generates a $2K_2$ -free subgraph in H .

In both cases, for higher values of τ it suffices to add snapshots with empty edge sets. \square

C.2 Proof of Theorem 5

Proof. We make a parameterized reduction from k -CLIQUE. Let G be graph and $k \geq 3$ be an integer. We construct the temporal graph $\mathcal{G} = (G', \lambda)$ as follows. First, add to G' every vertex in $V(G)$ and make $V = V(G)$. Second, add to G' a copy u' of every vertex $u \in V$ and define $V' = \{u' \mid u \in V\}$. Third, for every pair u, u' with $u \in V$ and $u' \in V'$ add the edge uu' to G' and make all such edges active at timestep 0. Fourth, consider an arbitrary ordering e_1, \dots, e_m the edges of G and for each edge $e_i = uv$ create for new vertices $\{h_{uv}, h_{vu}, h'_{uv}, h'_{vu} \mid uv \in E(G)\}$, and add edges:

- uh_{uv} and vh_{vu} , active at time i ;
- $u'h'_{uv}$ and $v'h'_{vu}$, active at time $2m + i$;
- $h_{vu}u$ and $h_{uv}v$, active at time $m + i$; and
- $h'_{vu}u'$ and $h'_{uv}v'$, active at time $3m + i$.

Denote the set $\{h_{uv}, h_{vu} \mid uv \in E(G)\}$ by H , and the set $\{h'_{uv}, h'_{vu} \mid uv \in E(G)\}$ by H' . We now prove that G has a clique of size at least k if and only if \mathcal{G} has a temporal connected set of size at least $2k$. The theorem follows by Proposition 15.

First, let $C \subseteq V$ be a clique of size at least k in G and $C' = \{u' \mid u \in C\}$. We show that $C \cup C'$ is a TCC of \mathcal{G} . For this, let $u, v \in C$. Since $\lambda(uh_{uv}) = i < m + i = \lambda(h_{uv}v)$, we get that u reaches v in \mathcal{G} through h_{uv} . Because C is a clique in G we conclude that C is a temporal connected set of \mathcal{G} , and similarly the same holds for C' . Thus it remains to show that pairs of vertices of the form u, u' with $u \in C$ and $u' \in C'$ are also connected in \mathcal{G} . This is true due to the choice of timestep 0 for the edges forming the matching between V and V' of \mathcal{G} .

Now, let $S \subseteq V(G')$ be a TCC of \mathcal{G} of size at least $2k$. We want to show that either $C = \{u \in V(G) \mid u \in S \cap V\}$ or $C' = \{u' \in V(G) \mid u' \in S \cap V'\}$ is a clique of G of size at least k . For this, we first prove a series of useful facts.

Claim 1. *Let P be a temporal path in $G'[V \cup H]$. Then P has at most one internal vertex of H , and hence $|V(P)| \leq 5$. The same holds if P is contained in $G'[V' \cup H']$.*

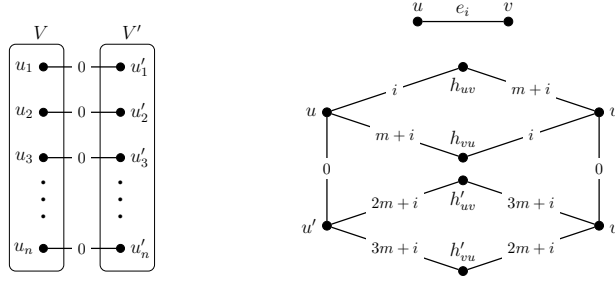


Figure 4: This appears as Figure 2 in the main text. Construction used in the proof of Theorem 5. On the left, the two copies of $V(G)$ and the edges between them, active in timestep 0. On the right, the edge $e_i \in E(G)$ and the associated gadget in \mathcal{G} .

Proof. It suffices to observe that every $e \in H$ is incident to exactly two edges of G' , one active at time at most m and the other one active at time at least $m + 1$. The second part follows because $G'[V \cup P]$ is a bipartite graph. \square

Claim 2. C and C' are cliques in G .

Proof. Let $u, v \in C$. Since C is a temporal connected set, there is a temporal path from u to v . Such path must contain only edges of $G'[V \cup H]$ since the edges between V and V' are only active in timestep 0, and all other edges are active in a later time (i.e., there is no way to leave u to u' at time 0, then go back to v). By Claim 1 and the fact that $G'[V \cup H]$ is bipartite, it follows that u and v must be adjacent. The argument for $u, v \in C'$ is analogous by taking their copies, u', v' in S . \square

Note that if $S \subseteq V \cup V'$, then Claim 2 and the fact that $|S| \geq 2k$ directly imply that either C or C' is a clique of size at least k in G . Assume now that $S \cap (H \cup H') \neq \emptyset$. In this case, it is not ensured that C or C' contains a clique of size at least k , but the following claims allow us to obtain another clique.

Claim 3. For every $h \in H$ and every $x' \in V' \cup H'$, h does not reach x' . Similarly, for every $h' \in H'$ and every $x \in V \cup H$, h' does not reach x .

Proof. The only edges between $V \cup H$ and $V' \cup H'$ are those incident to V and V' at timestep 0. Since every edge incident to $h \in H \cup H'$ is active only at a later timestep, the claim follows. \square

Claim 4. If $a, b \in S \cap H$, then a and b are related to the same edge, or to edges adjacent to each other. The same holds for $a, b \in S \cap H'$.

Proof. Suppose, without loss of generality, that a reaches b . Suppose also by contradiction that a, b are related to distinct edges, say e_i and e_j , respectively. Write e_i as uv and e_j as xy and assume that $\{u, v\} \cap \{x, y\} = \emptyset$. Because every temporal path between a and b must alternate between V and H , as $G'[V \cup H]$ is bipartite, and since by Claim 1 every temporal path contains at most one internal vertex of H , we get that the temporal a, b -path must use vertices (a, v, h_{vx}, x, b) . This gives us that ev and vh_{vx} must be active in timestep at most m , while $h_{vx}x$ and xb must be active in timestep at least $m + 1$. Hence, by letting vx be equal to e_ℓ , we must have that $i < \ell < j$. We apply an analogous argument to a temporal b, a -path to obtain that j

must be smaller than i , a contradiction. A similar argument can clearly be applied to $e, f \in S \cap H'$, and the claim follows. \square

Now suppose that $S \cap H \neq \emptyset$. By Claim 3 we get that $S \subseteq V \cup H$. Since V and H are disjoint and $|S| \geq 2k$, we get that either $|S \cap V| \geq k$ or $|S \cap H| \geq k$. If the former occurs, then C contains a clique of size at least k by Claim 2. Otherwise, denote by E_S the set of edges of G related to vertices in $S \cap H$ (i.e. $E_S = \{uv \in E(G) \mid \{h_{uv}, h_{vu}\} \cap S \neq \emptyset\}$). The following is the last ingredient of the proof.

Claim 5. *Let $a, b \in S \cap H$ be associated with distinct edges g, g' of G sharing an endpoint v . If u and w are the other endpoints of g and g' , respectively, then u and w are also adjacent in G . Additionally, either $|S \cap \{h_{xy}, h_{yx}\}| \leq 1$ for every $xy \in E(G)$, or $|S \cap H| \leq 2$.*

Proof. Suppose, without loss of generality, that a reaches b . By contradiction suppose that u and w are not adjacent in G . This gives us that every uw -path in G' contains two internal vertices of H , and therefore is not a temporal path by Claim 1. Because every subpath of a temporal path is also a temporal path, this means that there is no temporal a, b -path passing by u and w . By construction, and since G is a simple graph (i.e., there is only one edge with endpoints u and v , and only one with endpoints v and w), we get from Claim 1 that, if P is a temporal a, b -path not containing both u and w , then P is one of the following paths: $P_1 = (a, v, b)$; $P_2 = (a, u, a', v, b)$ where $\{a, a'\} = \{h_{uv}, h_{vu}\}$; or $P_3 = (a, v, f', w, f)$ where $\{f, f'\} = \{h_{vw}, h_{wv}\}$. Note that the same argument can be applied to a temporal b, a -path, except that we arrive to the reverses of the above paths. Note that, since all paths are strictly increasing, we get that at least one between P_2 or P_3 (or their reverse) is a temporal path. But observe that neither the subpath (a, u, a', v) nor its reverse can ever be temporal paths by construction, which means that neither P_2 nor its reverse can be temporal paths. A similar argument can be applied to P_3 , thus leading to a contradiction.

For the second part, suppose by contradiction that $\{h_{xy}, h_{yx}\} \subseteq S$ and $|S \cap H| > 2$. Let $a \in (S \cap H) \setminus \{h_{xy}, h_{yx}\}$. By Claim 4 we can suppose, without loss of generality, that $a \in \{h_{xw}, h_{wx}\}$ for some $w \neq y$. Observe also that the previous paragraph tells us that one of the temporal paths between $\{h_{xy}, h_{yx}\}$ and a must contain (y, f, w) or its reverse, where $f \in \{h_{yw}, h_{wy}\}$. Since such a path contains 4 edges, by letting xy be equal to e_i , yw be equal to e_j and wx be equal to e_ℓ , we get $i < j < \ell$. Thus in this case we have that we is active in time at least $m + 1$, which in turn gives us that $a = h_{xw}$. We can now verify that a does not reach h_{xy} . Indeed, every a, h_{xy} -path starting with edge aw must contain some internal vertex h of H , in which case it cannot be a temporal path as it starts with an edge active at time at least $m + 1$ (namely aw) and contains an edge active in time at most m (namely one of the edges incident to h). A similar argument can be applied if the path starts with edge ax , since it must be distinct from (e, x, h_{xy}) (recall that $\lambda(ax) = \ell > i = \lambda(xh_{xy})$). \square

Now, recall that we are in the case $|S \cap H| \geq k + 1$. By our assumption that $k \geq 3$, note that Claim 5 gives us that $|S \cap \{h_{xy}, h_{yx}\}| \leq 1$ for every $xy \in E(G)$, which in turn implies that $|E_S| = |S \cap H|$. Additionally, observe that, since $|S \cap H| \geq 4$, Claim 5 also gives us that there must exist $w \in V$ such that e is incident to w for every $e \in E_S$. Indeed, the only way that 3 distinct edges can be mutually adjacent without being all incident to a same vertex is if they form a triangle. Supposing that 3 edges in E_S form a triangle $T = (a, b, c)$, since $|E_S| \geq 4$, there exists an edge $e \in E_S \setminus E(T)$. But now, since G is a simple graph, e is incident to at most one between a, b and c , say a . We get a contradiction to Claim 5 as in this case e is not incident to edge $bc \in E_S$. Finally, by letting $C'' = \{v_1, \dots, v_k\}$ be any choice of k

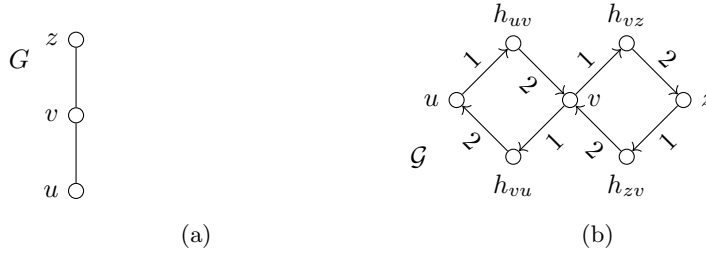


Figure 5: Given the graph in (a), Theorem 6 constructs the directed temporal graph in (b).

distinct vertices such that $\{wv_1, \dots, wv_k\} \subseteq E_S$, Claim 5 gives us that v_i and v_j are adjacent in G , for every $i, j \in [k]$; i.e., C'' is a clique of size at least k in G . This finishes the proof as the case $S \cap H' \neq \emptyset$ is clearly analogous. \square

C.3 Proof of Theorem 6

Proof. See Figure 5 to follow the construction. Let G be a graph and consider the directed graph D_G constructed as follows. First, add to D_G every vertex of G . Then, for each $uv \in E(G)$, add to D_G vertices h_{uv} and h_{vu} , directed edges uh_{uv} and vh_{vu} , and directed edges $h_{uv}v$ and $h_{vu}u$. Denote by H the set $\{h_{uv}, h_{vu} \mid uv \in E(G)\}$. To construct the directed temporal graph \mathcal{G} we start from D_G and for every $uv \in E(G)$

- make edges uh_{uv} and vh_{vu} active in timestep 1; and
- make edges $h_{uv}v$ and $h_{vu}u$ active in timestep 2.

Assume $k \geq 3$. We now prove that G has a clique of size k if and only if \mathcal{G} has a temporal connected set of size at least k . The theorem follows by Proposition 15. Notice that every vertex of G is contained in $V(\mathcal{G})$, and that \mathcal{G} has lifetime 2.

If C is a clique in G , then for every $u, v \in C$, we get that u reaches v and v reaches u in \mathcal{G} because of the paths $(u, 1, h_{uv}, 2, v)$ and $(v, 1, h_{vu}, 2, u)$. It remains to show that if \mathcal{G} has a temporal connected set of size at least k , then G has a clique of size at least k . Let C' be such a temporal connected set. We prove that $C' \subseteq V(G)$ and $uv \in E(G)$ for every $u, v \in C'$. First observe that G_1 has only edges from $V(G)$ to H , and G_2 , from H to $V(G)$. This implies that a temporal path must be of length at most 2, which in turn implies that $C' \cap V(G)$ must be a clique. Now suppose that there exists $h_{uv} \in C' \cap H$. Observe that h_{uv} has exactly one incoming edge, active in timestep 1, and exactly one outgoing edge, active in timestep 2. Additionally, observe that every edge outgoing from v is active in timestep 1. This means that v is the only vertex of $V(\mathcal{G})$ reachable from h_{uv} , contradicting the fact that $k \geq 3$. Thus we conclude that $C' \subseteq V(G)$ and the result follows.

Now, for the unilateral case, observe that every TCC is also a TUCC, hence from the above paragraph we get that if G has a clique of size at least k , then \mathcal{G} has a TUCC of size at least k . Now, if \mathcal{G} has a TUCC of size at least k , then observe that the same arguments as before can be applied. Indeed, if $u, v \in C' \cap V(G)$, then it must be that either u reaches v or v reaches u , and in any case we have $uv \in E(G)$. Additionally, we know that C' cannot contain any vertex of H , as $k \geq 3$ and v is the only vertex reachable by h_{uv} for every $h_{uv} \in H$. \square

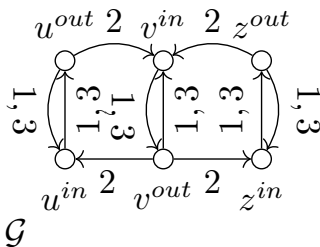


Figure 6: Temporal graph constructed in the Proof of Theorem 7, given the graph given in Figure 5a.

C.4 Proof of Theorem 7

Proof. Observe Figure 6 to follow the construction. Let G be a graph and consider the directed graph D_G constructed as follows. For every $u \in V(G)$, add to D_G vertices u^{in} and u^{out} , an edge from u^{in} to u^{out} , and an edge from u^{out} to u^{in} (notice that each pair $u^{\text{in}}, u^{\text{out}}$ induce a cycle in \mathcal{G}). Then, for each edge $uv \in E(G)$, add to D_G an edge from u^{out} to v^{in} and an edge from v^{out} to u^{in} . The directed temporal graph $\mathcal{G} = (D_G, \lambda)$ is such that λ is defined as follows.

- For every $u \in V(G)$, make edges between u^{in} and u^{out} active in timesteps 1 and 3 in both directions; and
- For every $uv \in E(G)$, make the edges from u^{out} to v^{in} and from v^{out} to u^{in} active in timestep 2.

We now prove that G has a clique of size at least k if and only if \mathcal{G} has a closed temporal connected set of size at least $2k$. The theorem follows by Proposition 15. Notice that \mathcal{G} has lifetime 3. Let C be a clique of size at least k in G , and let $C' = \{u^{\text{in}}, u^{\text{out}} \in V(\mathcal{G}) \mid u \in C\}$. We prove that, for every $u, v \in C$ with $u \neq v$, the set $\{u^{\text{in}}, u^{\text{out}}, v^{\text{in}}, v^{\text{out}}\}$ is a closed temporal connected set; note that this implies that C' itself is a closed temporal connected set, as desired. By construction, for every $w \in V(G)$ there are temporal paths from w^{in} to w^{out} and the other way around, in other words u^{in} reaches u^{out} , and vice-versa, and v^{in} reaches v^{out} and vice-versa. Moreover, u^{in} reaches v^{in} in \mathcal{G} through the path $(u^{\text{in}}, 1, u^{\text{out}}, 2, v^{\text{in}})$. Observe that this also implies that u^{out} reaches v^{in} , and by symmetry, that both v^{in} and v^{out} reach u^{in} . Finally, note that the path $(u^{\text{in}}, 1, u^{\text{out}}, 2, v^{\text{in}}, 3, v^{\text{out}})$ implies that both u^{in} and u^{out} reach v^{out} , and by symmetry we also get that v^{in} and v^{out} reach u^{out} . This finishes this part of the proof.

Assume now that that C' is a closed temporal connected set of \mathcal{G}' of size at least $2k$. Let $C = \{u \in V(G) \mid \{u^{\text{in}}, u^{\text{out}}\} \cap C' \neq \emptyset\}$. Clearly $|C| \geq k$ since $|C'| \geq 2k$. To show that C is a clique in G , observe that \mathcal{G} consists of a matching at times 1 and 3, containing only edges of the form $u^{\text{in}}u^{\text{out}}$ and of the form $u^{\text{out}}u^{\text{in}}$, together with edges in timestep 2 that go only from $O = \{u^{\text{out}} \mid u \in V(G)\}$ to $I = \{u^{\text{in}} \mid u \in V(G)\}$. This implies that any temporal path in \mathcal{G} contains at most one edge from O to I , which are only defined if the corresponding vertices are adjacent in G . We then get that, if $u, v \in C$ with $u \neq v$, then it must be the case that $uv \in E(G)$.

The proof for CLOSED TUCC is similar, except that, for every $uv \in E(G)$, we only need to add either $u^{\text{out}}v^{\text{in}}$ or $v^{\text{out}}u^{\text{in}}$ to \mathcal{G} . \square

C.5 Proof of Theorem 8

Proof. Let \mathcal{G} be a temporal graph and k be a positive integer. We first prove items 1 and 2, namely, that there exist algorithms running in time:

- $O(k^{k \cdot \tau} \cdot n)$ that decides whether there is a TCC of size at least k ; and
- $O(2^{k^\tau} \cdot n)$ that decides whether there is a CLOSED TCC of size at least k .

Denote by F the graph obtained from the reachability digraph $R(\mathcal{G})$ by removing all edges that are not symmetric and taking the underlying graph. Lemma 13 tells us that TCC is equivalent to finding a clique of size at least k in F , while CLOSED TCC is equivalent to finding a clique S of size at least k such that $\mathcal{G}[S]$ is connected. Observe that if $\Delta(F) \leq \Delta$, then the former can be solved by testing, for every $u \in V(F)$ and every $S \subseteq N_F(u)$ with $|S| = k - 1$, whether $S \cup \{u\}$ is a clique in F ; this takes time $O(\Delta^k \cdot k^2 \cdot n)$. Now for the latter, we need to test for the existence of such sets of bigger sizes. This is because CLOSED TCC's are not closed under inclusion. Nevertheless, since $\Delta(G) \leq \Delta$ and testing whether $\mathcal{G}[S]$ is connected can be done in time $O(|S| \cdot |E(\mathcal{G}[S])|)$, we can test for the existence of a CLOSED TCC in time $O(2^\Delta \cdot \Delta^3 \cdot n)$ by searching all cliques of size at least k in $N[u]$, for every $u \in V(G)$. We finish the proof by bounding the value of Δ .

Now, we show that $\Delta \leq (k - 1)^\tau$, which combined with the previous paragraph gives us the stated running time. For this, first notice that, for every $i \in [\tau]$, the vertex set of any connected component C of G_i is a clique in F and a closed temporal connected set of \mathcal{G} . This means that we can suppose that the size of any connected component of G_i is at most $k - 1$, for every $i \in [\tau]$, as otherwise we have a trivial yes-instance for both problems. Now, since $C_i(u) = R_i(u)$ when G is undirected, apply Lemma 14 to see that $|\mathcal{R}_\tau(u)| \leq (k - 1)^\tau$. Additionally, by definition we know that $\mathcal{R}_\tau(u)$ contains exactly the set of vertices reachable by u in \mathcal{G} . Since $v \in N_F(u)$ if and only if u reaches v and v reaches u , it follows that $N_F(u) \subseteq \mathcal{R}_\tau(u)$. This finishes the proof of items 1 and 2.

Now we turn our attention to items 3 and 4, namely, algorithms running in time:

- $O(k^{k^2} \cdot n)$ that decides whether there is a TUCC of size at least k ; and
- $O(2^{k^k} \cdot n)$ that decides whether there is a CLOSED TUCC of size at least k .

Again by applying Lemma 13, a similar argument as the one used for items 1 and 2 can be applied directly to the reachability graph $F = R(\mathcal{G})$ to say that, if $\Delta(F) \leq \Delta$, then TUCC can be solved in time $O(\Delta^k \cdot k^2 \cdot n)$, while CLOSED TUCC can be solved in time $O(2^\Delta \cdot \Delta^3 \cdot n)$. hence it remains to bound Δ .

We prove first that $d_G(u) \leq k - 2$ for every $u \in V(G)$. This holds because, given any pair $v, w \in N(u)$, and any choice of values $i \in \lambda(uv)$ and $j \in \lambda(uw)$, either we have $i \leq j$, in which case (v, i, u, j, w) is a temporal path, or $i > j$, in which case (w, j, u, i, v) is a temporal path. In other words, for every $v, w \in N(u)$, either v reaches w or w reaches v , which implies that $N_G[u]$ is a clique in F , for every $u \in V(G)$. Hence if $d_G(u) \geq k - 1$, we are in a trivial yes instance. To finish, just observe that any temporal path forms a CLOSED TUCC, which is always contained in a TUCC. Therefore, we can suppose that any vertex reachable from u is reached by a temporal path containing at most $k - 1$ edges. Because $d_G(v) \leq k - 2$ for every $v \in V(G)$, we get that $d_F(u) = |\mathcal{R}_\tau(u)| \leq (k - 2)^{k-1}$ and the result follows. \square

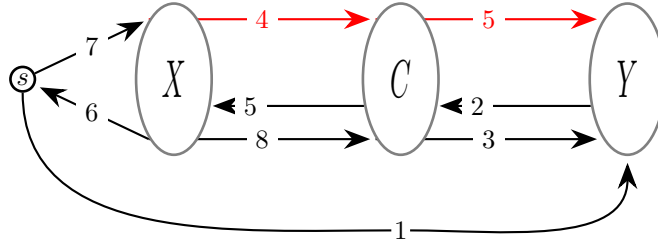


Figure 7: General structure of the constructed graph in the reduction for the temporal connectivity testing problem. A black edge denotes the existence of all possible edges. A red edge denotes the temporal edges whose existence is conditioned to the assignment not satisfying the clause.

D Checking Connectivity: proof of Theorem 9

This section is focused on Question 2, which is open for all definitions of components for both the strict and the non-strict models. We answer to the question providing the conditional lower bound in Theorem 9, which holds for both models, where the notation $\tilde{O}(\cdot)$ ignores poly-logarithmic factors.

We apply the technique used for instance in [7, 40, 45] to prove lower bounds for polynomial problems, falling within the fine-grained complexity framework. We use *quasilinear Karp reduction*, i.e. Karp reductions running in quasilinear time, whose formal definition is given in [7]. In the following we will use $\tilde{O}(\cdot)$ to neglect poly-logarithmic factors.

The key idea is to reduce a starting problem that is known not to be solvable in subquadratic time to our problem using such kind of reduction. This seed problem is the following formulation of the k -SAT* problem. Let ϕ be a CNF formula on variables $\mathcal{X} = \{x_1, \dots, x_n\}$ and $\mathcal{Y} = \{y_1, \dots, y_m\}$, with m clauses of size at most k . Let X denote the set of all 2^n possible truth assignments for \mathcal{X} , and similarly let Y denote the set of all 2^m possible truth assignments for \mathcal{Y} . In the k -SAT* problem, given $I = (\phi, X, Y)$, the goal is to decide if ϕ is satisfiable. The main difference with relation to the classical k -SAT problem is the size of the input, which is $|I| = O(2^n)$.

Remark 16 ([7]). *k -SAT* with input I cannot be solved in time $O(|I|^{2-\epsilon})$ for some ϵ , unless SETH fails.*

By presenting a quasilinear Karp reduction from k -SAT*, and applying Remark 16, we obtain that, unless SETH fails, there is no subquadratic algorithm that decides if a given temporal graph is temporal (unilaterally) connected.

In this section, it is helpful to formally define the following two problems.

Problem TEMPORAL CONNECTED.

Input: A temporal graph \mathcal{G} .

Question: Is \mathcal{G} temporal connected?

Problem TEMPORAL UNILATERALLY CONNECTED.

Input: A temporal graph \mathcal{G} .

Question: Is \mathcal{G} temporal unilaterally connected?

For both problems, given an instance $I = (\phi, X, Y)$ of k -SAT*, we construct a temporal graph $\mathcal{G} = (G, \lambda)$ such that \mathcal{G} is not temporal (unilaterally) connected if and only if ϕ has a satisfying assignment. As in the obtained temporal graph all non-strict paths are also strict, the result holds on both models.

We first present a reduction from k -SAT* to the complement of TEMPORAL CONNECTED. See Figure 7 to follow the construction. Also, let $C = \{c_1, \dots, c_m\}$ be the set of clauses in ϕ . Let $V(G) = X \cup C \cup Y \cup \{s\}$. Add all edges from s to X and let them be active in time 7; all edges between X and s and let them be active in time 6; all edges between X and C , active in time 8; all edges between C and X , active in time 5; all edges between C and Y , active in time 3; all edges between Y and C , active in time 2; and all edges between s and Y , active in time 1. Finally, for each pair $f \in X$ and $c_i \in C$, we add edge fc_i , active in time 4, if and only if f does *not* satisfy c_i . Similarly, for each pair $f \in Y$ and $c_i \in C$, we add edge c_if , active in time 5, if and only if f does *not* satisfy c_i . See Figure 8 for an example.

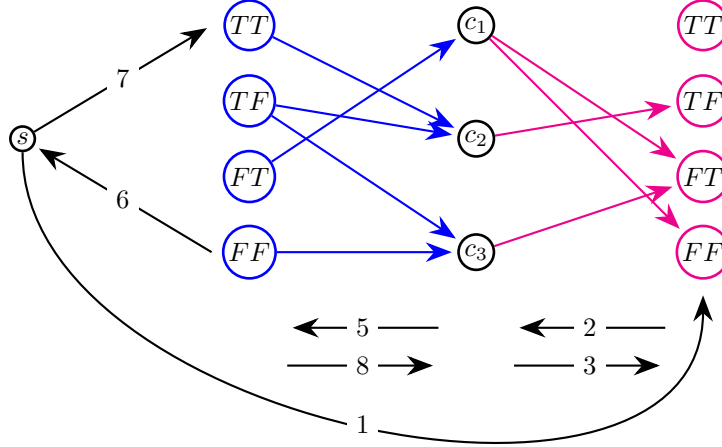


Figure 8: Graph in the reduction for the temporal connectivity testing problem, related to the formula $\phi = (x_1 \vee \neg x_2 \vee y_1) \wedge (\neg x_1 \vee \neg y_1 \vee y_2) \wedge (x_2 \vee y_1 \vee \neg y_2)$. Blue nodes denote assignments of x_1 and x_2 (e.g., node TT blue denotes the assignment $x_1 = True$ and $x_2 = True$), while magenta nodes denote assignments of y_1 and y_2 . Again, black edges denote existence of all possible edges. We put them outside the vertices in order to make the figure clean. Blue edges are active in time 4 and magenta edges, in time 5.

We now argue that this is a quasilinear Karp reduction. The reduction is quasilinear since $\tau = 8$, $|V(G)| = 2^n + 2^n + m + 1 = O(|I|)$, and $|E(G)| = 2^n m + 2^n m + 2^n + 2^n = \tilde{O}(|I|)$, since m in Remark 16 can be assumed to be $polylog(|I|)$ [7]. It remains to prove correctness. Before we do that, we first argue that the reachability graph of \mathcal{G} always contains uv for every u and v such that either $u \notin X$ or $v \notin Y$. For this, we analyse all cases below:

- s reaches f for every $f \in X \cup Y$ by direct edges;
- s reaches c_i for every $c_i \in C$ through a path $(s, 1, f, 2, c_i)$ for any $f \in Y$;
- s is reachable by f for every $f \in X$ by direct edges;
- s is reachable by every $c_i \in C$ through a path $(c_i, 5, f, 6, s)$ for any $f \in X$;

- s is reachable by every $f \in Y$ through a path $(f, 2, c_i, 5, f', 6, s)$ for any $c_i \in C_i$ and any $f' \in X$;
- Every $f \in X$ reaches every $c_i \in C$ and is reached by it through direct edges;
- Every $f \in X$ is reachable by every $f' \in Y$ through a path $(f', 2, c_i, 5, f)$ for any $c_i \in C_i$;
- Every $c_i \in C$ reaches every $f \in Y$ and is reached by it through direct edges;
- Every $f \in X$ reaches every $f' \in X$ through the path $(f, 6, s, 7, f')$;
- Every $c_i \in C$ reaches every $c_j \in C$ through the path $(c_i, 5, f, 8, c_j)$ for any $f \in X$;
- Every $f \in Y$ reaches every $f' \in Y$ through the path $(f, 2, c_i, 3, f')$ for any $c_i \in C$.

Now we prove that ϕ is satisfiable if and only if there exists $f_X \in X$ and $f_Y \in Y$ such that f_X does *not* reach f_Y (i.e., \mathcal{G} is *not* temporal connected). First, suppose that ϕ is satisfiable and consider a satisfying assignment f of ϕ . Then let $f_X \in X$ be equal to f restricted to $\{x_1, \dots, x_n\}$ and $f_Y \in Y$ be equal to f restricted to $\{y_1, \dots, y_n\}$. Observe that for every $c_i \in C$, either f_X satisfies c_i , and hence $(f_X c_i, 4) \notin E^T(\mathcal{G})$, or f_Y satisfies c_i , and hence $(c_i f_Y, 5) \notin E^T(\mathcal{G})$. Observe also that the only possible temporal paths between f_X and f_Y are of the type $(f_X, 4, c_i, 5, f_Y)$ for some $c_i \in C$. It thus follows that f_X does not reach f_Y . Now suppose that f_X does not reach f_Y for some pair $f_X \in X$ and $f_Y \in Y$. This must be because for every $c_i \in C$, either $(f_X c_i, 4) \notin E^T(\mathcal{G})$, and hence f_X satisfies c_i , or $(c_i f_Y, 5) \notin E^T(\mathcal{G})$, and hence f_Y satisfies c_i . Therefore $f_X \cup f_Y$ is a satisfying assignment for ϕ .

Consider now the complement of TEMPORALLY UNILATERALLY CONNECTED. We make a similar reduction. Observe Figure 9 to follow the construction. Let $V(G) = X \cup C \cup Y \cup \{x, y, c\}$. For each $z \in \{x, y, c\}$, add all edges from Z to z and let them be active in time 1, and all edges from z to Z and let them be active in time 2. Add also all edges from x to C and from c to Y , active in time 6. Finally, add $\{xc, xy, cy\}$ active in time 7 and, for each pair $f \in X$ and $c_i \in C$, we add edge $f c_i$, active in time 4, if and only if f does *not* satisfy c_i . Similarly, for each pair $f \in Y$ and $c_i \in C$, we add edge $c_i f$, active in time 5, if and only if f does *not* satisfy c_i .

Let $\mathcal{G} = (G, \lambda)$ be the constructed temporal graph. Similarly as before, we get $\tau = 7$, $|V(G)| = 2^{n+1} + m + 3 = O(|I|)$, and $|E(G)| = 2^{n+1}m + 2^{n+1} + m + 3 = \tilde{O}(|I|)$. It remains to prove that ϕ is not satisfiable if and only if \mathcal{G} is unilaterally temporal connected. As before, we first prove that the only missing pairs are of the type $f_X f_Y$ with $f_X \in X$ and $f_Y \in Y$. Recall that we only need at least one of the edges uv or vu for every pair $u, v \in V(G)$. Below we analyse only the necessary edges. See Figure 10 to follow the proof.

- x reaches every $u \in V(G)$ directly through edges. This implies edges xu in $R(\mathcal{G})$ for every $u \in V(G)$;
- Every $f_X \in X$ reaches every $z \in \{c, y\}$ through the path $(f_X, 1, x, 7, z)$. This implies edges $f_X c$ and $f_X y$ in $R(\mathcal{G})$;
- Every $f_X \in X$ reaches every $c_i \in C$ through the path $(f_X, 1, x, 6, c_i)$. This implies edges $f_X c$ in $R(\mathcal{G})$;
- Every $f_X \in X$ reaches every $f'_X \in X$, $f'_X \neq f_X$, through the path $(f_X, 1, x, 2, f'_X)$;

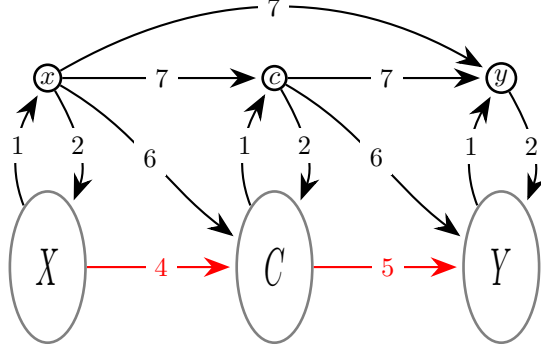


Figure 9: General structure of the constructed graph in the reduction for the unilateral temporal connectivity testing problem. A black edge denotes the existence of all possible edges. A red edge denotes the temporal edges whose existence is conditioned to the assignment not satisfying the clause.

- c reaches every $u \in C \cup Y \cup \{y\}$ directly through edges. This implies edges cu in $R(\mathcal{G})$;
- Every $c_i \in C$ reaches y through the path $(c_i, 1, c, 7, y)$. This implies edges $c_i y$ in $R(\mathcal{G})$;
- Every $c_i \in C$ reaches every $f_Y \in Y$ through the path $(c_i, 1, c, 6, f_Y)$. This implies edges $c_i f_Y$ in $R(\mathcal{G})$;
- Every $c_i \in C$ reaches every $c_j \in C$, $i \neq j$, through the path $(c_i, 1, c, 2, c_j)$. This implies edges $c_i c_j$ in $R(\mathcal{G})$;
- y reaches every $f_Y \in Y$ directly through an edge. This implies edges $y f_Y$ in $R(\mathcal{G})$;
- Every $f_Y \in Y$ reaches every $f'_Y \in Y$, $f'_Y \neq f_Y$, through the path $(f_Y, 1, y, 2, f'_Y)$. This implies edges $f_Y f'_Y$ in $R(\mathcal{G})$.

The proof of correctness is analogous to the previous one. It relies on the fact that the only possible temporal paths between the sets X and Y are of the type $(f_X, 4, c_i, 5, f_Y)$, for some $c_i \in C$. Indeed, no path from Y to X may exist, since each edge that leaves Y goes to y , and there are no edges from y to $V(G) \setminus Y$. Additionally, the only temporal edges leaving X that do not go directly to C are of type $(f_X, 1)$. If a temporal path starting with such temporal edge uses an edge at time 7, then it cannot arrive to Y , as all edges arriving in Y have smaller timestamps. Hence such a path must use an edge $(x c_i, 6)$ for some $c_i \in C$, and again we get stuck as all edges leaving C occur before time 6. Now, since the path $(f_X, 4, c_i, 5, f_Y)$ exists if and only if both f_X and f_Y do not satisfy c_i , the results follows. That is, if ϕ is satisfiable, then this fails for some pair, and vice-versa.

Now observe that we have made reductions from $k\text{-SAT}^*$ to the *complements* of our problems. However, since a subquadratic algorithm that solves the complement $\bar{\Pi}$ of a problem Π , also solves Π (indeed I is a positive instance of Π if and only if I is a negative instance of $\bar{\Pi}$), we get that Theorem 9 follows.

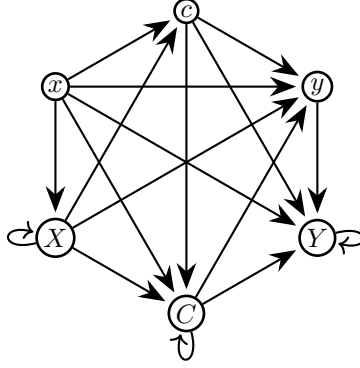


Figure 10: Subgraph of $R(\mathcal{G})$ shown above. An edge between two sets/vertices means that all such edges exist in $R(\mathcal{G})$. Note that more edges might exist but that this is enough to prove that we miss only an edge between X and Y (which means all edges between these sets).

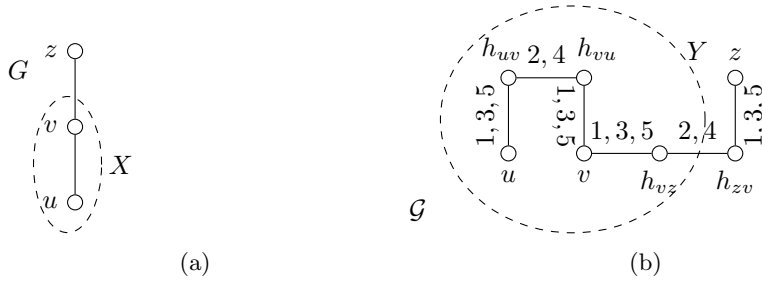


Figure 11: Construction in the proof of Theorem 10.

E Checking Maximality: proof of Theorem 10

Proof. It remains to prove that, in the non-strict model, given a (directed) temporal graph \mathcal{G} and a subset $Y \subseteq V(\mathcal{G})$, deciding whether Y is a CLOSED TCC (CLOSED TUCC) is NP-complete. The strict case is already treated in the main text. As before, we make a reduction from the problem of, given a graph G and $X \subseteq V(G)$, deciding whether X is a 2-club.

Observe Figure 11. We obtain \mathcal{G} from G by subdividing each edge $uv \in E(G)$ twice, creating vertices h_{uv} and h_{vu} , with $\lambda(uh_{uv}) = \lambda(vh_{vu}) = \{1, 3, 5\}$, and $\lambda(h_{uv}h_{vu}) = \{2, 4\}$. Observe that the first vertex in the subscript of h_{xy} tells us which between x and y is adjacent to h_{xy} . Denote by H the set $\{h_{uv}, h_{vu} \mid uv \in E(G)\}$. We now prove that X is a maximal 2-club in G if and only if $Y = X \cup N_H(X)$ is a CLOSED TCC in \mathcal{G} . In fact, we prove that:

1. If $X \subseteq V(G)$ is such that $G[X]$ has diameter at most 2, then $Y = X \cup N_H(X)$ is a closed connected set in \mathcal{G} ; and
2. If $Y \subseteq V(\mathcal{G})$ is a closed connected set, then $X = Y \cap V(G)$ is such that $G[X]$ has diameter at most 2.

We argue that indeed 1 and 2 above imply what we want, i.e., that X is a maximal 2-club in G if and only if $Y = X \cup N_H(X)$ is a CLOSED TCC in \mathcal{G} . Observe that,

supposing that 1 and 2 hold, if X is a maximal 2-club, then $Y = X \cup N_H(X)$ must be a CLOSED TCC. Indeed, if $Y \subset Y'$ and Y' is a closed connected set (i.e., Y is not maximal), then by 2 we get that $X' = Y' \cap V(G)$ has diameter 2. Since X' contains X , this contradicts the choice of X . Conversely, if Y is a CLOSED TCC, then X must be a maximal 2-club, as otherwise we could apply 1 to get a closed connected set strictly containing Y .

We first prove 1. So, consider $X \subseteq V(G)$ such that $G[X]$ has diameter at most 2, and define Y as above. Let $u, v \in Y \cap V(G)$. If $uv \in E(G)$, then $(u, 1, h_{uv}, 2, h_{vu}, 3, v)$ and $(v, 1, h_{vu}, 2, h_{uv}, 3, u)$ witness that u reaches v and v reaches u in $\mathcal{G}[Y]$. And if $uv \notin E(G)$, then, since $G[X]$ has diameter 2, let $w \in N(u) \cap N(v)$ in G . We get the paths: $(u, 1, h_{uw}, 2, h_{wu}, 3, w, 3, h_{wv}, 4, h_{vw}, 5, v)$ and $(v, 1, h_{vw}, 2, h_{wv}, 3, w, 3, h_{wu}, 4, h_{uw}, 5, u)$. Therefore, u reaches v and v reaches u in $\mathcal{G}[Y]$. Now, consider $u \in X \cap Y$ and $h \in H \cap Y$. Let $v \in X$ be such that $h \in N(v)$ (observe that v is uniquely defined). If $v = u$, then there is nothing to prove, so suppose otherwise. Because X has diameter at most 2, there exists a u, v -path P in $G[X]$ of length at most 2, say (u, w, v) , with possibly $w = v$. Then either $(u, 1, h_{uw}, 2, h_{wu}, 3, w, 3, h)$ is a temporal u, h -path, in case $w = v$, or $(u, 1, h_{uw}, 2, h_{wu}, 3, w, 3, h_{wv}, 4, h_{vw}, 5, v, 5, h)$ is a temporal u, h -path, in case $w \neq v$. One can check that the symmetric path between h and u ensures that also h reaches u in $\mathcal{G}[Y]$. Now, let $h, h' \in H$, and let $u \in N(h) \cap X$ and $v \in N(h') \cap X$. One can observe that a similar argument can be applied, by possibly starting the previous path with $(h, 1, u)$, in case $h = h_{ux}$ for some x not within the u, v -path P taken in $G[X]$.

Now, assume that Y is a closed connected set of \mathcal{G} , and consider $X = Y \cap V(G)$. We want to show that $G[X]$ has diameter at most 2. Suppose by contradiction that u and v are a distance 3 in $G[X]$. Observe that, since each $h \in H$ has degree exactly 2 in \mathcal{G} , we get that every temporal path in \mathcal{G} is related to exactly one path in G , and vice-versa. One can then verify that u and v cannot reach each other in $\mathcal{G}[Y]$ as the traversal of any edge in G is related to the traversal of a strictly increasing path of length 3 in $\mathcal{G}[Y]$. In other words, no u, v -path in \mathcal{G} is a valid temporal path, as the lifetime of \mathcal{G} is 5.

Finally, we prove that every closed unilaterally connected set is also a closed connected set. Since the reverse trivially holds, we get that it is also NP-complete to decide whether $Y \subseteq V(\mathcal{G})$ is a CLOSED TUCC. So, consider $Y \subseteq V(\mathcal{G})$ a closed unilaterally connected set, and suppose that $x, y \in Y$ are such that x reaches y in $\mathcal{G}[Y]$. Let $(x = x_1, t_1, x_2, \dots, x_q, t_q, x_{q+1} = y)$ be a temporal x, y -path in $\mathcal{G}[Y]$. By a case analysis, one can check that there exist t'_1, \dots, t'_q for which $(y = x_{q+1}, t'_1, x_{q-1}, \dots, x_2, t'_q, x_1 = x)$ is a valid temporal y, x -path. \square