Supplementary Materials for

The fundamental advantages of temporal networks

A. Li, S. P. Cornelius, Y.-Y. Liu, L. Wang,* A.-L. Barabási*

*Corresponding author. Email: barabasi@gmail.com (A.-L.B.); longwang@pku.edu.cn (L.W.)

Published 24 November 2017, Science 358, 1042 (2017)
DOI: 10.1126/science.aai7488

This PDF file includes:

- Materials and Methods
- Figs. S1 to S25
- References
Materials and Methods

S1  Controllable space  
   S1.1 Temporal networks  .................................................. 3  
   S1.2 Static networks  ....................................................... 6  
   S1.3 Relation between the controllable spaces of static versus temporal networks  6

S2  Description of empirical data sets  6

S3  Network randomization and null models  8

S4  Relationship between $S_t$ and $S_s$  9

S5  Control energy  11
   S5.1 Derivation of control energy for temporal networks  .................. 11  
   S5.2 Solving the quadratic problem  ...................................... 13  
   S5.3 Minimum energy needed to control temporal networks  .......... 15  
   S5.4 Minimum energy needed to control static networks  ............ 16

S6  Analysis of the control energy  17

S7  Scaling behavior of the lower bound of minimum control energy  18  
   S7.1 A simple case  ....................................................... 18  
   S7.2 General case  ......................................................... 22  
      S7.2.1 Two snapshots and arbitrary number of driver nodes . . . 22  
      S7.2.2 Arbitrary numbers of snapshots and driver nodes ........... 25

S8  Use of the Laplacian matrix for $A_m$  25

S9  Locality of the optimal control trajectories for temporal networks  26

S10 Figures  29

S11 References and Notes  49
S1 Controllable space

S1.1 Temporal networks

For a dynamical system of the form $\dot{x}(t) = Ax(t) + Bu(t)$, the system state at time $t$ is given by $x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-s)}Bu(s)ds$, with the initial state $x(t_0) = x_0$. For temporal networks we can write the system state after the $m$th snapshot as $x(t_m) = e^{A_m \Delta t_m}x(t_{m-1}) + \int_{t_{m-1}}^{t_m} e^{A_m(t_m-s)}B_m u_m(s)ds$. After $M$ snapshots, the final state $x_f$ at time $t_M$ is

$$
\begin{align*}
\mathbf{x}_f &= \prod_{m=M}^{1} e^{A_m \Delta t_m}x_0 + \sum_{m=1}^{M-1} \left( \prod_{j=M}^{m+1} e^{A_j \Delta t_j} \int_{t_{m-1}}^{t_m} e^{A_m(t_m-s)}B_m u_m(s)ds \right) + \int_{t_{M-1}}^{t_M} e^{A_M(t_M-s)}B_M u_M(s)ds.
\end{align*}
$$

Hence, we can write all states $x_f$ that can be reached from $x_0 = 0$ as

$$
\begin{align*}
\mathbf{x}_f &= \sum_{m=1}^{M-1} \left( \prod_{j=M}^{m+1} e^{A_j \Delta t_j} \int_{t_{m-1}}^{t_m} e^{A_m(t_m-s)}B_m u_m(s)ds \right) + \int_{t_{M-1}}^{t_M} e^{A_M(t_M-s)}B_M u_M(s)ds.
\end{align*}
$$

Similarly, we can write all states $x_0$ that can reach $x_f = 0$ as

$$
\begin{align*}
\mathbf{x}_0 &= -\sum_{m=1}^{M} \prod_{j=1}^{m} e^{-A_j \Delta t_j} \int_{t_{m-1}}^{t_m} e^{A_m(t_m-s)}B_m u_m(s)ds.
\end{align*}
$$

Taken together, the set of states $x_f$ from $x_0 = 0$ under control input $u_m(t)$ ($m = 1, 2, \cdots, M$) for a temporal network defined by $\{(A_m, B_m, \Delta t_m)\}_{m=1}^{M}$ is

$$
\begin{align*}
\mathbf{\Omega} &= \sum_{m=1}^{M-1} \prod_{j=M}^{m+1} e^{A_j \Delta t_j} \left\{ \mathbf{x} \bigg| \mathbf{x} = \int_{t_{m-1}}^{t_m} e^{A_m(t_m-s)}B_m u_m(s)ds, \text{ for } \forall u_m \right\} \\
&+ \left\{ \mathbf{x} \bigg| \mathbf{x} = \int_{t_{M-1}}^{t_M} e^{A_M(t_M-s)}B_M u_M(s)ds, \text{ for } \forall u_M \right\},
\end{align*}
$$

and the corresponding set of states $x_0$ to $x_f = 0$ is

$$
\begin{align*}
\sum_{m=1}^{M} \prod_{j=1}^{m} e^{-A_j \Delta t_j} \left\{ \mathbf{x} \bigg| \mathbf{x} = \int_{t_{m-1}}^{t_m} e^{A_m(t_m-s)}B_m u_m(s)ds, \text{ for } \forall u_m \right\}.
\end{align*}
$$
We can simplify the integration term above using the following lemma.

**Lemma 1**: Given matrices $A \in \mathbb{R}^{N \times N}$ and $B \in \mathbb{R}^{N \times p}$, for any $0 \leq t_0 < t_f < +\infty$, we have

$$\left\{ x \mid x = \int_{t_0}^{t_f} e^{A(t_f-s)}Bu(s)\,ds, \forall \ u \right\} = \langle A|B \rangle,$$

where

$$\langle A|B \rangle = \langle A|R(B) \rangle = R(B) + AR(B) + \cdots + A^{N-1}R(B) = \sum_{i=0}^{N-1} A^iR(B),$$

with ‘+’ corresponding to the direct summation of vector spaces and $A^0$ being the identity matrix. The column space of $B_{N \times p}$ is defined as $R(B) = \{ Bv \mid v \in \mathbb{R}^p \}$.

The above lemma can be derived from Lemma 2.10 in Ref. [31]. For readers’ convenience, we offer a proof here.

**Proof**: Denoting $S = \left\{ x \mid x = \int_{t_0}^{t_f} e^{A(t_f-s)}Bu(s)\,ds, \forall \ u \right\}$, from

$$e^A = \sum_{i=0}^{\infty} \frac{1}{i!} A^i$$

we have

$$S = \left\{ x \mid x = \int_{t_0}^{t_f} \sum_{i=0}^{\infty} \frac{1}{i!} A^i(t_f-s)^iBu(s)\,ds, \forall \ u \right\}$$

$$= \left\{ x \mid x = \sum_{i=0}^{\infty} A^iB \int_{t_0}^{t_f} \frac{1}{i!} (t_f-s)^iBu(s)\,ds, \forall \ u \right\}$$

$$\subset \sum_{i=0}^{N-1} A^iR(B) = \langle A|B \rangle.$$

Note that for $i > N - 1$, $A^i$ can be obtained as linear combinations of $A^0, A^1, \cdots, A^{N-1}$. Hence we have $S \subset \langle A|B \rangle$.

Consider the matrix

$$S_0 = \int_{t_0}^{t_f} e^{A(t_f-s)}BB^Te^{A^T(t_f-s)}ds.$$
If \( x \) belongs to the null space of \( S_0 \), we have
\[
0 = x^T S_0 x = \int_{t_0}^{t_f} x^T e^{A(t-t-s)} BB^T e^{A^T(t-t-s)} x ds
\]
\[
= \int_{t_0}^{t_f} \| B^T e^{A^T(t-t-s)} x \| ds,
\]
which induces \( B^T e^{A^T(t-t-s)} x = 0 \) for all \( s \in [t_0, t_f] \). This requires that all derivatives of \( B^T e^{A^T(t-t-s)} x \) equal to 0 at \( t_f \), that is
\[
B^T x = 0, B^T A x = 0, \ldots, B^T (A^T)^m x = 0, \ldots
\]
which gives
\[
x \in N(B^T) \cap N(B^T A^T) \cap \cdots \cap N(B^T (A^T)^m) \cap \cdots
\]
\[
= [R(B) + R(AB) + \cdots + R(A^{N-1}B)]_\perp
\]
\[
= \langle A|B \rangle_\perp,
\]
since \( N(B^T) = \{ x | B^T x = 0 \} = [R(B)]_\perp \), where \( [Q]_\perp \) means the orthogonal complementary space of \( Q \). Conversely, if \( x \in \langle A|B \rangle_\perp \), we have \( x \in N(S_0) \). Hence
\[
N(S_0) = \langle A|B \rangle_\perp,
\]
or equivalently,
\[
R(S_0) = \langle A|B \rangle.
\]
If \( x \in \langle A|B \rangle \), there exists a vector \( z \) such that \( x = S_0 z \). Then using the control signal, \( u(s) = B^T e^{A^T(t-t-s)} z \), we have
\[
x = S_0 z = \int_{t_0}^{t_f} e^{A(t-t-s)} BB^T e^{A^T(t-t-s)} z ds = \int_{t_0}^{t_f} e^{A(t-t-s)} Bu(s) ds \in S.
\]
Thus we obtain \( \langle A|B \rangle \subset S \).

Taken together, we have Lemma 1: \( \left\{ x \mid x = \int_{t_0}^{t_f} e^{A(t-t-s)} Bu(s) ds, \forall u \right\} = \langle A|B \rangle \). }
Based on the above lemma, for the temporal network $\{(A_m, B_m, \Delta t_m)\}_{m=1}^M$, we define

$$\Omega = \langle A_M | B_M \rangle + \sum_{m=1}^{M-1} \prod_{j=M}^{m+1} e^{A_j \Delta t_j} \langle A_m | B_m \rangle,$$

as the controllable space, and

$$\sum_{m=1}^{M} \prod_{j=1}^{m} e^{-A_j \Delta t_j} \langle A_m | B_m \rangle = \langle A_1 | B_1 \rangle + \sum_{m=2}^{M} \prod_{j=1}^{m-1} e^{-A_j \Delta t_j} \langle A_m | B_m \rangle,$$

as the reachable space. In the area of complex networks, for convenience, we set $x_0 = 0$, and note that here the dimensions of controllable and reachable spaces are equal.

### S1.2 Static networks

When all snapshots of a temporal network are identical, i.e. $A_m = A_s$, the temporal network reduces to a static one [22], and the spaces (S3) and (S4) reduce to $\langle A_s | B \rangle$. It follows that a static network is controllable if and only if

$$\langle A_s | B \rangle = \mathbb{R}^N,$$

which is the classic Kalman’s rank condition for controllability [8].

### S1.3 Relation between the controllable spaces of static versus temporal networks

It is natural to ask what is the relation between (S3) and its counterpart (S5). The answer is that the relationship between the controllable spaces of temporal ($\Omega_t$) and static ($\Omega_s$) networks is not determinate. Indeed, Fig. S1 shows a simple example illustrating that theoretically both $\Omega_t \supset \Omega_s$ and $\Omega_t \subset \Omega_s$ are possible.

### S2 Description of empirical data sets

We construct temporal networks from four different kinds of empirical data (Table S1):
ACM conference: This data set is from the ACM Hypertext 2009 conference provided by the SocioPatterns collaboration [32], where the 113 conference attendees wearing radio badges were monitored for face-to-face communications. Every communication between a pair of attendees is stored as a triplet \((t, i, j)\) (the number of face-to-face communications is 20,818), meaning that people with anonymous IDs \(i\) and \(j\) chatted with each other during the 20-second interval \([t-20s, t]\). The data spans a time period of about 2.5 days (212,340s) starting from 8am on Jun 29th, 2009 [25]. The snapshot duration \(\Delta t\) is chosen from 1000s to 212,340s for each temporal network, yielding different temporal sequences with different numbers of snapshots.

Student contacts: This data set consists of a sequence of time-stamped contacts between 126 students in three classes in a high school in Marseilles, France over 4 days in Dec. 2011 [33]. The format of the data is same as that of ACM conference, and it is also provided by the SocioPatterns collaboration [32]. Here \(\Delta t\) is chosen between 1000s to 326,450s.

Ant interactions: The interactions in this data set represent antenna-body contacts of four colonies of the ant Temnothorax rugatulus [26]. We adopt the largest colony (colony 1) as our data set, comprising 1,911 interactions between 89 ants over 1,438 seconds. Here the duration of each antenna-body is neglected (Fig. S2). As with the ACM conference data, we generate the snapshots of the temporal network with \(\Delta t\) chosen between 10s to 1438s.

Protein network: This dataset is based on the protein-protein interaction network of Saccharomyces cerevisiae from the Database of Interacting Proteins (DIP), consisting of 5,023 proteins and 22,570 interactions. From different gene expression datasets, researchers construct the dynamic protein-protein interactions by identifying the active time of each protein [27]. According to gene ontology, networks of proteins are then constructed based on three domains: cellular component (CC), molecular function (MF), and biological process (BP). Here we have 33, 50, and 50 snapshots for CC, MF, and BP, where the number of related proteins is 84, 74, and 85, respectively.

Technological network: Here we consider three datasets consisting of data packet exchanges between 25 wireless radios in an emulated mobile ad-hoc network (MANET) that experiences a denial-of-service cyber-attack. Each dataset covers a time period of 900s, with the cyber-attack occurring after approximately 300s of operation. The packets are generated
using real application and networking software running in a special test environment that emulates the packet loss characteristics of a wireless communication channel between radios in motion. A packet exchange is recorded by each network protocol that handles the packet. The applications generating the packets are specially designed test applications that are configured to model communication in a mobile wireless network that could be seen in a search and rescue mission. We construct the temporal network with 50 snapshots from three datasets named 1-ip6, 2-ip6, and 3-ip6, respectively. Each network contains the same set of 34 nodes.

Figures S2-S5 show, as a function of time, the interaction activity, degree distribution, average degree, and number of components of the aggregated networks for each of these datasets.

Table S1: Characteristics of the empirical data sets. \( N \) is the number of nodes, and \( M \) is the (maximum) number of snapshots. For human and animal data sets stored in terms of sequence of interactions, \( M \) is acquired from the time window we choose to aggregate the networks. Considering the basic attributes of temporal networks highly depend on \( \Delta t \), here we only list \( N \) and \( M \), and other information of the data is shown in Figs. S2 to S5.

<table>
<thead>
<tr>
<th></th>
<th>ACM conference contacts</th>
<th>Student interactions</th>
<th>Ant interactions</th>
<th>Protein network</th>
<th>Technological network</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>113</td>
<td>126</td>
<td>89</td>
<td>84</td>
<td>85</td>
</tr>
<tr>
<td>( M )</td>
<td>212, 340/( \Delta t )</td>
<td>326, 430/( \Delta t )</td>
<td>1, 438/( \Delta t )</td>
<td>33</td>
<td>50</td>
</tr>
</tbody>
</table>

S3 Network randomization and null models

Each data set used in our study can be represented by a sequence of contacts, namely triplets of the form \((t, i, j)\), indicating that two individuals \( i \) and \( j \) interact with each other at time \( t \) (ant interactions) or from \( t \) for a duration of 20s (ACM conference). There are many different methods to randomize temporal networks, yielding different null models [19]. Here are the typical null models:

**Time Reversal (TR):** The temporal order of the contacts is reversed, *e.g.* with the first becoming the last and vice versa. Note that under this procedure the duration of a given interaction could be changed, especially for high resolution data sets. For example, if we have a sequence \((t_1, i, j), (t_2, i, j), (t_3, i', j')\), and \((t_3, i', j)\) (which is common in the ACM conference data set), after we implement TR, we may have \((t_3, i, j), (t_2, i, j), (t_1, i', j')\), and \((t_1, i', j)\) where the duration of every link is retained, or \((t_3, i, j), (t_3, i, j), (t_2, i', j')\), and \((t_1, i', j)\) where the
duration of the interaction between $i$ and $j$ is now reduced by half. In our data, we run TR according to the two cases above, and find our results are robust. This model is designed to assess the causality between individual interactions, for example, whether the latter contacts are triggered by the former [19], or if there are strong temporal correlations embedded in the original data.

**Randomly Permuted Times (RPT):** Here we shuffle the timestamps of the contacts, leaving the sources and targets of the links unaltered. Note that RPT has the effect of destroying temporal patterns and erasing time correlations between contacts.

**Randomized Edges (RE):** In this model, we iteratively choose pairs of contacts $(i, j)$ and $(i', j')$, and replace them with $(i, i')(j, j')$ or $(i, j')(j, i')$ with equal probability provided the change results in neither self loops nor multiple edges. Here duration of interactions is maintained from the point of whole static network, while the numbers and durations of an individual node’s contacts are generally altered. For example, for two contiguous contacts $(t_1, i, j)$ and $(t_2, i, j)$, RE may change $(t_2, i, j)$ into $(t_2, i, j')$. A given node’s degree may be conserved in data with low temporal resolution, while it is likely to change in high resolution data.

**Randomized Edges and Randomly Permuted Times (RERPT):** Equivalent to RE followed by RPT.

Since the raw PPI and technology data are already represented as network snapshots, we first extract the corresponding contact sequence and use the index of a contact’s containing snapshot as its interaction time. That is, each link (between $i$ and $j$) in snapshot $m$ is represented by the triplet $(m, i, j)$.

For the empirical data we considered, the effects of the above randomizations on the average degree and number of components of the aggregated networks are shown in Figs. S4 and S5. The schematic illustration of methods used to randomize temporal networks is shown in Fig. S7.

**S4  Relationship between $S_t$ and $S_s$**

For a given (sub)sequence of $M$ snapshots, we define the corresponding static network $A_s$ by aggregating all the $M$ snapshots. This reflects the widely-used convention [34, 35, 36] that a static network represents an aggregation of a temporal sequence. As we have shown in the
main text, the mechanism we illustrate in Fig. 2 is the rule rather than the exception, which makes real temporal networks typically more controllable than their static counterparts. Below we show additional analyses to check the robustness of our results.

As we only show temporal networks corresponding to a single $\Delta t$ for the ACM conference and ant interactions data in main text, more cases of $\Delta t$ are given in Fig. S10. The result demonstrated in Fig. 3 (namely that $S_t < S_s$) holds for other values of $\Delta t$.

For the supplementary student interactions data we analyzed, the corresponding values of $S_t$ and $S_s$ are shown in Fig. S9C. As with the ant interactions data, the interaction times are a sequence of discrete time points rather than a time interval like the communications between conference attendees and students. We give the results based on the original data format in Fig. 2, assigning a small, finite duration to each contact, which generates the result in Fig. S10. The robustness of the results for other values of the snapshot duration has also been verified. The results shown in Fig. S10 corroborate those in Fig. 2B, suggesting that the result given in main text does not depend on the duration of the interactions.

For a temporal network with $M$ snapshots, we define $S_t$ ($S_s$) to be equal to $M$ if the corresponding temporal (static) networks are not controllable even upon reaching (aggregating) the final snapshot $M$. In this case, the number of snapshots required for control is larger than $M$, or equivalently, more driver nodes are needed. For the protein and technological networks, we find many cases where $S_s = M$ in Fig. 2, thus we performed additional analysis by adding more driver nodes and thereby decreasing $S_t$ and $S_s$. The results are shown in Fig. S11.

Here we employ a toy model to explore the above insight we got from empirical datasets. We first generate a set of $M$ snapshots randomly and independently according to the $G(N,p)$ model [37], where the link weights are assigned independently and randomly from $(0,1)$. We then randomly generate $N_s$ snapshot sequences, calculating $S_t$ and $S_s$ for each under the same set of driver nodes. We find that with more snapshots (larger $M$) the likelihood that $S_t > S_s$ drops to zero (Fig. S12).

This result can be qualitatively understood as follows. We know that the emergence (or aggregation) of new snapshots with independent edge weights will never shrink the controllable subspace for temporal (or static) networks. In the temporal case, the controllable subspace will typically expand after the emergence of a new snapshot. Yet, in static networks, aggregating
a new snapshot usually switches the controllable subspace from one to another without necessarily adding additional dimensions (Fig. 1C and 1D). With increasing $M$, temporal (static) networks will have more snapshots to explore (aggregate), which offers temporal networks more opportunities to expand the controllable subspace. By definition, when the dimension of the controllable subspace reaches the size of the system, the system becomes fully controllable. Hence temporal networks will typically need fewer snapshots to achieve full controllability than static networks, implying that $S_t < S_s$ dominates for large $M$, consistent with our numerical findings.

Finally, considering that the previous four randomization models presented in Sec. S3 mainly alter the network—not the timing of the events, here we analyse a new model, Randomly Distributed Times (RDT, Fig. S7), by keeping the network unchanged but changing the timing of each event. In RDT, we randomly distribute time stamps using Poisson, uniform, and normal distributions. After generating a series of time stamps that follow the given distribution, we adjust every time stamp $t'_i$ to the observation time window $[t_{min}, t_{max}]$ of the original data with an appropriate transformation: $(t'_i - t'_{min})(t_{max} - t_{min})/(t'_{max} - t'_{min}) + t_{min}$ for $t'_{min} > 0$, where $t'_{min}$ ($t'_{max}$) is the minimal (maximal) number in the generated time stamps. Note that when $t'_{min} \leq 0$ the expression becomes $(t'_i + t'_{min})(t_{max} - t_{min})/(t'_{max} - t'_{min}) + t_{min}$. After performing RDT on all datasets, we find that our main conclusion that temporal networks reach controllability faster than the corresponding static networks still holds (Fig. S13).

S5 Control energy

S5.1 Derivation of control energy for temporal networks

For a single snapshot $(A, B)$ (or equivalently, a static network), the minimum energy for controlling the system from $x_0$ at $t_0$ to $x_f$ at $t_f$ corresponds to the unique input of the form $u(t) = B^T e^{A^T(t_f-t)}c_s$, where $c_s = W_s^{-1}(x_f - e^{At}x_0)$ and $W_s = \int_{t_0}^{t_f} e^{A(t_f-s)}BB^T e^{A^T(t_f-s)} ds$ [38, 39]. Here $c_s$ is a constant vector determined by $x_0$, $x_f$, $t_0$, and the system’s dynamics.

For simplicity, we consider the case where each snapshot is controllable, which actually makes it possible to compare the energy for controlling temporal networks and corresponding static networks (otherwise we cannot ensure that the static and temporal versions of the network are
According to the principle of optimality, if \( u(t) \) is the energy-optimal input to control a temporal network, then the energy accumulated over each snapshot must also be minimal for the control sub-problem of traveling between the states at the beginning and end of that snapshot. Hence we can write the candidate energy optimal control signals for a temporal network as

\[
u(t) = B^T_m e^{A^T_m (t_m - t)} c_m \quad \text{for} \quad t_{m-1} \leq t < t_m, \ m = 1, 2, \cdots, M.
\]

Based on the above, the solution for the temporal network is

\[
x_f - e^{A_M \Delta t_M} \cdots e^{A_1 \Delta t_1} x_0 = e^{A_M \Delta t_M} \cdots e^{A_2 \Delta t_2} \int_{t_0}^{t_1} e^{A_1 (t_1 - s)} B_1 u(s) ds + \cdots + \int_{t_{M-1}}^{t_M} e^{A_M (t_M - s)} B_M u(s) ds
\]

\[
= e^{A_M \Delta t_M} \cdots e^{A_2 \Delta t_2} \int_{t_0}^{t_1} e^{A_1 (t_1 - s)} B_1 B_1^T e^{A^T_1 (t_1 - s)} ds \cdot c_1 + \cdots + \int_{t_{M-1}}^{t_M} e^{A_M (t_M - s)} B_M B_M^T e^{A^T_M (t_M - s)} ds \cdot c_M.
\]

(S6)

Note that the above form of the optimal input over each snapshot is derived based on the principle of optimality combined with the (known) form of the optimal control signals in static networks.

Using the following notation

\[
d = x_f - e^{A_M \Delta t_M} \cdots e^{A_1 \Delta t_1} x_0,
\]

\[
W_m = \int_{t_{m-1}}^{t_m} e^{A_m (t_m - s)} B_m B_m^T e^{A^T_m (t_m - s)} ds
\]

\[
= \int_{0}^{\Delta t_m} e^{A_{m+1} \Delta t_{m+1}} B_{m+1} B_{m+1}^T e^{A^T_{m+1} \Delta t_{m+1}} ds,
\]

\[
c = (c_1^T, c_2^T, \cdots, c_M^T)^T,
\]

\[
H = (e^{A_M \Delta t_M} \cdots e^{A_2 \Delta t_1} \mathbf{W}_1, \ldots, e^{A_M \Delta t_M} \cdots e^{A_{m+1} \Delta t_{m+1}} \mathbf{W}_m, \ldots, \mathbf{W}_M) = \mathbf{S} \mathbf{W},
\]

\[
S = \left( e^{A_M \Delta t_M} \cdots e^{A_2 \Delta t_1}, \ldots, e^{A_M \Delta t_M} \cdots e^{A_{m+1} \Delta t_{m+1}}, \mathbf{I}_N \right)
\]

\[
= \left( \prod_{l=M}^{1} e^{A_l \Delta t_l}, \ldots, \prod_{l=M}^{m+1} e^{A_l \Delta t_l}, \mathbf{I}_N \right),
\]

\[
\mathbf{W} = \text{diag}(\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_M),
\]
we can write (S6) as \( d = Hc \).

The energy to control temporal networks from \( x_0 \) at \( t_0 \) to \( x_f \) at \( t_f \) can be written as

\[
E(x_0, x_f) = \frac{1}{2} \int_{t_0}^{t_f} u^T(t)u(t)dt = \frac{1}{2}c^TWc.
\]

Hence, the minimum energy could be obtained by solving the quadratic programming problem

\[
\min \ E(x_0, x_f) = \frac{1}{2}c^TWc
\]

s.t. \( Hc = d \) \quad (S7)

for the unknown \( c \).

**S5.2 Solving the quadratic problem**

Since \( W \) is positive definite and symmetric, we have \( W = UΛU^T \), where \( UU^T = U^TU = I \) and \( Λ \) is diagonal. Using

\[
x = \sqrt{Λ}U^Tc
\]

(i.e. \( c = U\left( \sqrt{Λ} \right)^{-1}x \)) and

\[
K = HU\left( \sqrt{Λ} \right)^{-1}
\]

we can write the quadratic programming problem (S7) as

\[
\min \ E(x_0, x_f) = \frac{1}{2}c^TWc = \frac{1}{2}c^TUΛU^Tc = \frac{1}{2}x^Tx
\]

s.t. \( HU\left( \sqrt{Λ} \right)^{-1}x = Kx = d \) \quad (S8)

To solve (S8), let

\[
f(x, v) = \frac{1}{2}x^Tx + v^T(Kx - d)
\]

and minimize \( f(x, v) \), where \( v \) is a set of Lagrange multipliers. At the point \((x^*, v^*)\) where
\( f(x, v) \) reaches the minimum, the following relations must be satisfied

\[
\frac{\partial f(x, v)}{\partial x^*} = x^* + K^T v^* = 0 \tag{S9}
\]
\[
\frac{\partial f(x, v)}{\partial v^*} = Kx^* - d = 0. \tag{S10}
\]

Multiplying both sides of (S9) by \( K \) on the left, we have

\[
Kx^* + KK^Tv^* = 0
\]
\[
Kx^* = d.
\]

If \( KK^T \) is non-singular, \( v^* = - (KK^T)^{-1} d \), and then according to (S9) we have

\[
x^* = K^T (KK^T)^{-1} d. \tag{S11}
\]

Since \( K = HU \left( \sqrt{\Lambda} \right)^{-1} \), if we prove \( KK^T \) is non-singular, then the problem (S8) can be solved according to the expression (S11). To do so, we employ the following Lemma 2.

**Lemma 2**: If \( K \) is a matrix over real numbers with size \( n \times m \), then the rank of \( K \) and \( KK^T \) is equal.

It is a simple exercise of advanced matrix theory to prove the above lemma, and we would like to give a proof as follows.

**Proof**: The null space of \( K^T \) is given by vectors \( x \) satisfying \( K^T x = 0 \). And the null space of \( KK^T \) is given by vectors \( y \) satisfying \( KK^T y = 0 \). Since \( K^T x = 0 \), we have \( KK^T x = 0 \), i.e. \( x \) belongs to the null space of \( KK^T \). From \( KK^T y = 0 \), we have \( y^T KK^T y = 0 = (K^T y)^T K^T y \), i.e., \( K^T y = 0 \) and \( y \) belongs to the null space of \( K^T \). Thus, the two equations \( K^T x = 0 \) and \( KK^T y = 0 \) have same solutions. As such, the number of independent vectors in the fundamental system is also the same, i.e. \( n - \text{rank}(K^T) = n - \text{rank}(KK^T) \). Hence we have \( \text{rank}(K) = \text{rank}(K^T) = \text{rank}(KK^T) \). \( \square \)

Based on the above **Lemma 2**, we have \( \text{rank}(KK^T) = \text{rank}(K) = \text{rank}(HU \left( \sqrt{\Lambda} \right)^{-1}) \),
and $K$ is a matrix with size $N \times NM$, and

$$
HU \left( \sqrt{\Lambda} \right)^{-1} = SWU \left( \sqrt{\Lambda} \right)^{-1} = SUU^T U \left( \sqrt{\Lambda} \right)^{-1} = SU \sqrt{\Lambda} \\
= \left( e^{A_1 \Delta t_1} \cdots e^{A_{j-1} \Delta t_{j-1}} \cdots e^{A_{j+1} \Delta t_{j+1}} \cdots , I_N \right) U \sqrt{\Lambda}.
$$

We know that $U = (u_1, u_2, \cdots , u_N)$, where $u_i$ is an eigenvector of one of the eigenvalues of $W$, and $\text{rank}(U) = N$. In addition, we have $\text{rank}(\sqrt{\Lambda}) = N$. Thus we obtain $\text{rank}(SU \sqrt{\Lambda}) = \text{rank}(SU) = \text{rank}(S)$. Since the last block of $S$ is $I_N$, we have $\text{rank}(S) = N$. Hence we have $\text{rank}(KK^T) = \text{rank}(K) = \text{rank} \left( HU \left( \sqrt{\Lambda} \right)^{-1} \right) = N$, and $KK^T$ is a non-singular square matrix with size $N$.

Thus the solution of the problem (S8) is $x^* = K^T (KK^T)^{-1} d$, where $K = HU \left( \sqrt{\Lambda} \right)^{-1}$.

Hence we have

$$
E(x_0, x_f) = \frac{1}{2} x^T x^* = \frac{1}{2} d^T \left[ K^T (KK^T)^{-1} \right]^T \left[ K^T (KK^T)^{-1} \right] d \\
= \frac{1}{2} d^T (KK^T)^{-1} d = \frac{1}{2} d^T \left( SUU^T U \Lambda^{-1} U^T U \Lambda U^T S \right)^{-1} d \\
= \frac{1}{2} d^T (SWS^T)^{-1} d.
$$

### S5.3 Minimum energy needed to control temporal networks

Taken together, the quadratic programming problem given in Eq. (S7) is solved analytically by the optimal solution

$$
c^* = S^T (SWS^T)^{-1} d, \quad (S12)
$$

with the corresponding minimum control energy

$$
E^*(x_0, x_f) = \frac{1}{2} d^T W_{\text{eff}}^{-1} d, \quad (S13)
$$

where the $N \times N$ matrix $W_{\text{eff}} = SWS^T$ is an “effective” gramian matrix, encoding the energy structure of the temporal network. Hereafter, we refer to the minimum control energy $E^*(x_0, x_f)$ as simply the control energy $E$. 

15
For controllability in the case $x_0 = 0$, the above results reduce to

$$c^*_c = S^T (SWST)^{-1} x_f,$$

and

$$E = \frac{1}{2} x_f^T W_{\text{eff}}^{-1} x_f. \quad (S14)$$

**S5.4 Minimum energy needed to control static networks**

When all snapshots are identical ($A_m = A_s$), our results reduce to the case of static networks. Indeed, for static networks, the quadratic programming (S7) becomes

$$\min \quad E(x_0, x_f) = \frac{1}{2} c^T W c$$

s.t. \quad $W_s c = d = x_f - e^{A_s(t_f-t_0)} x_0.$

If the system is controllable, $W_s$ is nonsingular, and so there is a unique solution $c = W_s^{-1} d$. Hence, the optimal solution from above quadratic programming determines the optimal input as

$$u(t) = B^T e^{A_s(T_f-t)} W_s^{-1} (x_f - e^{A_s(t_f-t_0)} x_0),$$

from which we have

$$E = \frac{1}{2} (x_f - e^{A_s(t_f-t_0)} x_0)^T W_s^{-1} (x_f - e^{A_s(t_f-t_0)} x_0), \quad (S15)$$

where $W_s = \int_{t_0}^{t_f} e^{A_s(s-t)} B B^T e^{A_s(T_f-s)} ds$, and $e^{-A_s t_f} W_s e^{-A_s T_f}$ is the canonical controllability gramian matrix [38]. The result for $M = 1$ is same as that given in [39].
As $A_m = A_s$ for $m = 1, 2, \ldots, M$, we have

\[
SWS^T = \sum_{i=1}^{M-1} \left( \prod_{k=M}^{i+1} e^{A_k \Delta t_k} W_i \prod_{l=i+1}^{M} e^{A_l^T \Delta t_l} \right) + W_M
\]

\[
= \sum_{i=1}^{M-1} \int_0^{\Delta t_i} e^{A(s+\sum_{k=i+1}^{M} \Delta t_k)} BB^T e^{A^T(s+\sum_{k=i+1}^{M} \Delta t_k)} ds + W_M
\]

\[
= \int_0^{\sum_{k=1}^{M} \Delta t_k} e^{A_s BB^T e^{A^T_s} s} ds
\]

\[
= W_s.
\]

Thus the energy for controlling temporal networks (S14) reduces to the known result for static networks (S15).

**S6 Analysis of the control energy**

To account for the fact that the control energy generally grows as $x_0$ and $x_f$ become farther apart, we introduce the normalized control energy \[39\] from $x_0 = 0$ to $x_f$, namely

\[
E = \frac{x^T_f (SWS^T)^{-1} x_f}{2x^2_f x_f}.
\]

We can obtain bounds for this normalized energy as

\[
\underline{E} = \frac{1}{2\lambda_{\text{max}}} \leq E \leq \overline{E} = \frac{1}{2\lambda_{\text{min}}},
\]

where $\underline{E}$ and $\overline{E}$ are the lower and upper bounds of $E$; $\lambda_{\text{max}}$ and $\lambda_{\text{min}}$ are the maximum and minimum eigenvalues of $W_{\text{eff}}$, respectively. Note that $W_{\text{eff}}$ is symmetric and positive definite because the underlying temporal (or static) network is taken to be controllable in our analysis. As such, the minimum eigenvalue of $W_{\text{eff}}$ is bigger than 0, meaning all quantities are well defined.

We can demonstrate numerically that $\lambda_{\text{min}}$ is generally greater in temporal networks, and hence $\overline{E}$ is usually smaller, often much smaller, than in their static counterparts (Fig. S14). This implies that the average control energy $\langle E \rangle$ is typically much less in a temporal network,
despite the fact that $\bar{E}$ may correspond to different “worst-case” directions in the static versus temporal case. Indeed a typical control direction $d$ will have some component lying along the eigenvector of $W_{\text{eff}}$ corresponding to $\lambda_{\text{min}}$. Since the eigenvalues of $W_{\text{eff}}$ typically vary many orders of magnitude (Fig. S14), this worst-case direction dominates the control energy, and $\bar{E}$ is expected to be representative of $\langle E \rangle$, an expectation borne out by our results (Fig. 3). This also explains why the scaling of $\langle E \rangle$ is determined by that of $\bar{E}$, which we can show decreases according to $\langle E \rangle \sim \Delta t^{-\gamma}$ for small $\Delta t$ before reaching a plateau (Fig. 3).

The robustness of these results has been checked for other networks and shown in Figs. S16 - S18. To test the extent to which our results rely on knowing the exact sequence of snapshots we generate a set of $M$ random snapshots, and enumerate all the possible switching sequences of the snapshots from the set. We find that the control energy does not vary notably as the switching sequence changes, both for real and synthetic networks (Fig. S19 and Fig. S20). This indicates that this control cost is principally a function of the set of possible networks, not their orders.

S7 Scaling behavior of the lower bound of minimum control energy

For ease of understanding, we first derive the scaling behavior of $\bar{E}$ for small $\Delta t$ for the simple case of two snapshots $A_1$, $A_2$ and one driver node. The general case will be derived later.

S7.1 A simple case

For the simple case with two snapshots and one driver node, we employ the notation $A(i, j) = A_{ij} = a_{ij}$ to represent the entry at $i$th row and $j$th column in matrix $A$, and let $A_1 = (a_{ij})_{NN}$ and $A_2 = (b_{ij})_{NN}$. As only the $c$th node receives input directly (a single driver node), we have $B^T = (0, \cdots, 1, \cdots, 0)$ where the $c$th entry is 1 while others are 0.

For undirected temporal network, both $A_1$ and $A_2$ are symmetric with real entries, allowing us to write $A_1 = P\Theta P^T$ and $A_2 = Q\Gamma Q^T$, where $P = (P_{ij})_{NN}$, $Q = (Q_{ij})_{NN}$, $\Theta =$
\[
\begin{pmatrix}
\theta_1 \\
\vdots \\
\theta_N
\end{pmatrix}, \quad \text{and} \quad \Gamma = \begin{pmatrix}
\gamma_1 \\
\vdots \\
\gamma_N
\end{pmatrix}, \quad \text{and we assume} \quad \lambda(A_1) : \theta_1 > \theta_2 > \cdots > \theta_N, \\
\text{and} \quad \lambda(A_2) : \gamma_1 > \gamma_2 > \cdots > \gamma_N.
\]

As there are two snapshots \( A_1 \) and \( A_2 \), we have

\[
W_{\text{eff}} = \text{SWS}^T = \begin{pmatrix}
e^{A_2 \Delta t} & e^{A_1 \Delta t} \end{pmatrix} \begin{pmatrix}
e^{A_2 \Delta t} & e^{A_1 \Delta t} \end{pmatrix} dt.
\]

Then we obtain

\[
W_{\text{eff1}} = Q e^{\Gamma_\Delta t} P \int_0^{\Delta t} e^{\Theta_p d_t} BB^T e^{\Theta_p d_t} Q, \\
W_{\text{eff2}} = Q \int_0^{\Delta t} e^{\Gamma_\Delta t} BB^T Q e^{\Gamma_\Delta t} Q.
\]

and

\[
W_{\text{eff1}}(i,j) = \sum_{r=1}^N \sum_{s=1}^N \sum_{m=1}^N \sum_{n=1}^N Q_{ir} e^{\gamma \Delta t} Q_{sr} \sum_{k=1}^N \sum_{l=1}^N \frac{P_{sk} P_{ck} P_{cl} P_{ml}}{\theta_k + \theta_l} \left[ e^{(\theta_k + \theta_l) \Delta t} - 1 \right] Q_{mn} e^{\gamma \Delta t} Q_{jn}\]

\[
= \sum_{r=1}^N \sum_{s=1}^N \sum_{m=1}^N \sum_{n=1}^N Q_{ir} e^{(\gamma + \gamma) \Delta t} Q_{sr} Q_{mn} Q_{jn} \sum_{k=1}^N \sum_{l=1}^N \frac{P_{sk} P_{ck} P_{cl} P_{ml}}{\theta_k + \theta_l} \left[ e^{(\theta_k + \theta_l) \Delta t} - 1 \right]. \tag{S16}
\]

\[
W_{\text{eff2}}(i,j) = \sum_{k=1}^N \sum_{l=1}^N \frac{Q_{ik} Q_{cl} Q_{jl}}{\gamma_k + \gamma_l} \left[ e^{(\gamma_k + \gamma_l) \Delta t} - 1 \right]. \tag{S17}
\]
Generally, as $\Delta t \to 0$, $e^{(\gamma_k + \gamma_l)\Delta t} \approx 1 + (\gamma_k + \gamma_l)\Delta t$. Then we have

$$W_{\text{eff1}}(i, j) \approx \sum_{r=1}^{N} \sum_{s=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} \sum_{r=1}^{N} \sum_{n=1}^{N} Q_{ir}Q_{cr}Q_{cn}Q_{jn} \Delta t$$

$$= \sum_{r=1}^{N} \sum_{n=1}^{N} Q_{ir}Q_{cr}Q_{cn}Q_{jn} [1 + (\gamma_r + \gamma_n)\Delta t] \Delta t$$

$$= \Delta t \sum_{r=1}^{N} \sum_{n=1}^{N} Q_{ir}Q_{cr}Q_{cn}Q_{jn} + \Delta t^2 \sum_{r=1}^{N} \sum_{n=1}^{N} Q_{ir}Q_{cr}Q_{cn}Q_{jn} \gamma_r$$

$$+ \Delta t^2 \sum_{r=1}^{N} \sum_{n=1}^{N} Q_{ir}Q_{cr}Q_{cn}Q_{jn} \gamma_n, \Omega_1 \Omega_2 \Omega_3$$

and

$$W_{\text{eff2}}(i, j) \approx \sum_{k=1}^{N} \sum_{l=1}^{N} Q_{ik}Q_{ck}Q_{cl}Q_{jl} \Delta t$$

$$= \begin{cases} 
\Delta t \sum_{k=1}^{N} Q_{ik}Q_{ck} & \text{if } j = c \\
\Delta t \sum_{l=1}^{N} Q_{cl}Q_{jl} & \text{if } i = c \\
0 & \text{otherwise}
\end{cases} = \begin{cases} \Delta t & \text{if } i = j = c \\
0 & \text{otherwise}
\end{cases} .$$

Furthermore, we obtain

$$\Omega_1 = \begin{cases} 
\Delta t & \text{if } i = j = c \\
0 & \text{otherwise}
\end{cases}$$

$$\Omega_2 = \begin{cases} 
\Delta t^2 \sum_{r=1}^{N} Q_{ir}Q_{cr} \gamma_r = \Delta t^2 b_{ic} & \text{if } j = c \\
0 & \text{otherwise}
\end{cases}$$

$$\Omega_3 = \begin{cases} 
\Delta t^2 \sum_{n=1}^{N} Q_{cn}Q_{jn} \gamma_n = \Delta t^2 b_{cj} & \text{if } i = c \\
0 & \text{otherwise}
\end{cases} .$$
Thus we have

$$W_{\text{eff}}(i,j) = W_{\text{eff}1}(i,j) + W_{\text{eff}2}(i,j) = \begin{cases} 2\Delta t + 2b_{cc}\Delta t^2 & \text{if } i = j = c \\ b_{ij}\Delta t^2 & \text{if } i = c \text{ and } j \neq c \\ b_{ic}\Delta t^2 & \text{if } j = c \text{ and } i \neq c \end{cases}$$

and we get

$$W_{\text{eff}} = \begin{pmatrix} 0 & \cdots & 0 & b_{1c}\Delta t^2 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & b_{c-1,c}\Delta t^2 & 0 & \cdots & 0 \end{pmatrix}.$$

To determine the eigenvalues of $W_{\text{eff}}$, we just need to solve the following set of linear equations

$$|W_{\text{eff}} - \lambda I| = \begin{vmatrix} -\lambda & \cdots & 0 & b_{1c}\Delta t^2 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -\lambda & b_{c-1,c}\Delta t^2 & 0 & \cdots & 0 \\ b_{c1}\Delta t^2 & \cdots & b_{c,c-1}\Delta t^2 & 2\Delta t + 2b_{cc}\Delta t^2 - \lambda & b_{c,c+1}\Delta t^2 & \cdots & b_{cN}\Delta t^2 \\ 0 & \cdots & 0 & b_{c+1,c}\Delta t^2 & -\lambda & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & b_{Nc}\Delta t^2 & 0 & \cdots & -\lambda \end{vmatrix} = \lambda^2 - (2\Delta t + 2b_{cc}\Delta t^2)\lambda - \Delta t^4 \sum_{i=1,i\neq c}^{N} b_{ic}^2 (-\lambda)^{N-2} = 0.$$

It follows that the maximum eigenvalue of $W_{\text{eff}}$ is given by

$$\lambda_{\text{max}} = \Delta t + b_{cc}\Delta t^2 + \sqrt{\Delta t^2 + 2b_{cc}\Delta t^3 + \Delta t^4 \sum_{i=1}^{N} b_{ic}^2}.$$
Therefore, in the limit $\Delta t \to 0$, we have

$$E \approx \frac{1}{2 \left( \Delta t + b_{cc} \Delta t^2 + \sqrt{\Delta t^2 + 2b_{cc} \Delta t^3 + \Delta t^4 \sum_{i=1}^{N} b_{ic}^2} \right)} \sim \Delta t^{-1}. $$

### S7.2 General case

For the general case, we first assume that there are $p \leq N$ driver nodes, and the set of nodes receiving inputs is $\mathcal{I} = \{i_1, i_2, \cdots, i_p\}$. Without loss of generality, we can relabel all nodes such that the driver nodes are $j = 1, 2, \cdots, p$ (i.e. $i_j = j$), with node $j$ receiving the input $u_j(t)$. Thus we have $B_{ii} = 1$ for $i = 1, 2, \cdots, p$, with other entries of $B$ equal to 0. We shall first analyze the control energy with two snapshots, and later derive the more general case.

#### S7.2.1 Two snapshots and arbitrary number of driver nodes

For the control gramian entries $W_{\text{eff}1}(i,j)$ and $W_{\text{eff}2}(i,j)$ given in equations (S16) and (S17), we have

$$W_{\text{eff}1}(i,j) = \sum_{r=1}^{N} \sum_{s=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} Q_{ir} e^{i(\gamma_r + \gamma_n)\Delta t} Q_{sr} Q_{mn} Q_{jn} \sum_{k=1}^{N} \sum_{x=1}^{N} \sum_{s=1}^{p} \sum_{y=1}^{N} \sum_{l=1}^{N} P_{sk} P_{yk} B_{xs} B_{ys} P_{yl} P_{ml} \theta_k \theta_l \left[ e^{i(\theta_k + \theta_l)\Delta t} - 1 \right],$$

$$W_{\text{eff}2}(i,j) = \sum_{k=1}^{N} \sum_{x=1}^{N} \sum_{s=1}^{p} \sum_{y=1}^{N} \sum_{l=1}^{N} Q_{ik} Q_{xk} B_{xs} B_{ys} Q_{jl} Q_{yl} \gamma_k \gamma_l \left[ e^{i(\gamma_k + \gamma_l)\Delta t} - 1 \right],$$

for $p$ driver nodes. Denoting $BB^T = (B'_{xy})_{NN}$, we have

$$B'_{xy} = \sum_{s=1}^{p} B_{xs} B_{ys} = \begin{cases} 1 & \text{if } 1 \leq x = y \leq p, \\ 0 & \text{otherwise} \end{cases},$$

and we have $B = \begin{pmatrix} I \\ 0 \end{pmatrix}$ and $BB^T = \begin{pmatrix} I \\ 0 \end{pmatrix} \begin{pmatrix} I & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}$, where $I$ is the identity matrix with size $p$.  

22
Hence we obtain

\[ W_{\text{eff}_1}(i,j) = \sum_{r=1}^{N} \sum_{s=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} Q_{ir} e^{(\gamma_r + \gamma_n)\Delta t} Q_{sr} Q_{mn} Q_{jn} \sum_{k=1}^{N} \sum_{p=1}^{p} \sum_{l=1}^{N} P_{sk} P_{ck} P_{pl} P_{ml} \frac{e^{(\theta_k + \theta_l)\Delta t} - 1}{\theta_k + \theta_l} \]

\[ W_{\text{eff}_2}(i,j) = \sum_{k=1}^{N} \sum_{c=1}^{N} \sum_{l=1}^{N} Q_{ik} Q_{ck} Q_{cl} Q_{jl} \frac{e^{(\gamma_k + \gamma_l)\Delta t} - 1}{\gamma_k + \gamma_l} \]

As \( \Delta t \to 0 \), we get

\[ W_{\text{eff}_1}(i,j) \approx \Omega_1 + \Omega_2 + \Omega_3 \quad W_{\text{eff}_2}(i,j) \approx \begin{cases} \Delta t & \text{if } i = j \in \mathcal{I} \\ 0 & \text{otherwise} \end{cases} \]

and

\[ \Omega_1 = \begin{cases} \Delta t & \text{if } i = j \in \mathcal{I} \\ 0 & \text{otherwise} \end{cases} \]

\[ \Omega_2 = \begin{cases} \Delta t^2 \sum_{r=1}^{N} Q_{ir} Q_{jr} \gamma_r = \Delta t^2 b_{ij} & \text{if } j \in \mathcal{I} \\ 0 & \text{otherwise} \end{cases} \]

\[ \Omega_3 = \begin{cases} \Delta t^2 \sum_{n=1}^{N} Q_{in} Q_{jn} \gamma_n = \Delta t^2 b_{ij} & \text{if } i \in \mathcal{I} \\ 0 & \text{otherwise} \end{cases} \]

For details of the notation, we refer the reader to equations (S19) and (S18). Furthermore, we have

\[ W_{\text{eff}}(i,j) = W_{\text{eff}_1}(i,j) + W_{\text{eff}_2}(i,j) = \begin{cases} 2\Delta t + 2b_{ii} \Delta t^2 & \text{if } i = j \in \mathcal{I} \\ b_{ij} \Delta t^2 & \text{if } i \in \mathcal{I} \text{ and } j \neq i \\ b_{ij} \Delta t^2 & \text{if } j \in \mathcal{I} \text{ and } i \neq j \end{cases} \]
and

\[
W_{\text{eff}} = \begin{pmatrix}
2\Delta t + 2b_{1,1}\Delta t^2 & \cdots & b_{1,p}\Delta t^2 & b_{1,p+1}\Delta t^2 & \cdots & b_{1N}\Delta t^2 \\
\vdots & & \vdots & \vdots & & \vdots \\
b_{p,1}\Delta t^2 & \cdots & 2\Delta t + 2b_{p,p}\Delta t^2 & b_{p,p+1}\Delta t^2 & \cdots & b_{pN}\Delta t^2 \\
b_{p+1,1}\Delta t^2 & \cdots & b_{p+1,p}\Delta t^2 & 0 & \cdots & 0 \\
\vdots & & \vdots & \vdots & & \vdots \\
b_{N,1}\Delta t^2 & \cdots & b_{N,p}\Delta t^2 & 0 & \cdots & 0
\end{pmatrix}.
\]

To obtain the eigenvalues of \( W_{\text{eff}} \), we must solve the following equations

\[
|W_{\text{eff}} - \lambda I| = \begin{vmatrix}
2\Delta t + 2b_{1,1}\Delta t^2 - \lambda & \cdots & b_{1,p}\Delta t^2 & b_{1,p+1}\Delta t^2 & \cdots & b_{1N}\Delta t^2 \\
\vdots & & \vdots & \vdots & & \vdots \\
b_{p,1}\Delta t^2 & \cdots & 2\Delta t + 2b_{p,p}\Delta t^2 - \lambda & b_{p,p+1}\Delta t^2 & \cdots & b_{pN}\Delta t^2 \\
b_{p+1,1}\Delta t^2 & \cdots & b_{p+1,p}\Delta t^2 & -\lambda & \cdots & 0 \\
\vdots & & \vdots & \vdots & & \vdots \\
b_{N,1}\Delta t^2 & \cdots & b_{N,p}\Delta t^2 & 0 & \cdots & -\lambda
\end{vmatrix}

= |2\Delta t + 2b_{1,1}\Delta t^2 - \lambda + \sum_{l=p+1}^{N} \frac{b_{l,1}^2\Delta t^4}{\lambda} & \cdots & b_{1,p}\Delta t^2 + \sum_{l=p+1}^{N} \frac{b_{l,p}b_{1,1}\Delta t^4}{\lambda} \\
\vdots & & \vdots & \vdots & & \vdots \\
b_{p,1}\Delta t^2 + \sum_{l=p+1}^{N} \frac{b_{l,1}b_{1,1}\Delta t^4}{\lambda} & \cdots & 2\Delta t + 2b_{p,p}\Delta t^2 - \lambda + \sum_{l=p+1}^{N} \frac{b_{l,p}\Delta t^4}{\lambda} | (-\lambda)^{N-p}.
\]

As \( \Delta t \to 0 \), the determinant of \( W_{\text{eff}} - \lambda I \) can be approximated by the first order expansion in \( \Delta t \), yielding \( |W_{\text{eff}} - \lambda I| \approx (2\Delta t - \lambda)^p (-\lambda)^{N-p} \). We thus obtain \( E \sim \Delta t^{-1} \).
### S7.2.2 Arbitrary numbers of snapshots and driver nodes

For each snapshot, we write $A_j = U_j \Lambda_j U_j^T$ with $U_j = \left(U_{r_i}^{(j)}\right)_{NN}$ and $\Lambda_j = \begin{pmatrix} \lambda_j^{(j)} \\ \vdots \\ \lambda_j^{(N)} \end{pmatrix}$.

Then we have

$$W_{\text{eff}}(i, j) = \sum_{i_M, O_M, \ldots, O_{l+2}, i_{l+1}, O_{l+1}, O_{l+2}, \ldots, O_M, i_M} U_i^{(M)} e^{\lambda_i^{(M)} \Delta t} U_{O_{l+1}, i_{l+1}}^{(l+1)} \sum_{c \in L} \sum_{i_c=1}^{N} \sum_{i'_c=1}^{N} U_c^{(l)} U_{c,i_c}^{(l)} U_{i_c,i'_c}^{(l)} e^{\lambda_{i_c}^{(l)} + \lambda_{i'_c}^{(l)}} \Delta t - 1 \cdot U_{O_{l+1}, i_{l+1}}^{(l+1)} e^{\lambda_{i_{l+1}}^{(l+1)} \Delta t} U_{O_{l+2}, i_{l+1}}^{(l+1)} \ldots U_M^{(M)} e^{\lambda_M^{(M)} \Delta t} U_{j,i_M}^{(M)}$$

where $W_{\text{eff}} = \sum_{l=1}^{M} W_{\text{eff}}(i, j)$.

For small $\Delta t$, we have

$$\frac{1}{\lambda_i^{(l)} + \lambda_{i'_c}^{(l)}} \left[e^{\left(\lambda_i^{(l)} + \lambda_{i'_c}^{(l)}\right) \Delta t} - 1\right] \approx \Delta t,$$

which leads to $E \sim \Delta t^{-1}$ for small $\Delta t$ according to derivation provided in Sec. S7.2.1.

In summary, for short $\Delta t$ we prove that for temporal networks, the lower bound of control energy follows

$$E \sim \Delta t^{-1},$$

regardless of the number of driver nodes and snapshots. Note that this includes static networks as a special case, recapitulating the earlier observation that $E \sim \Delta t^{-1}$ [39].

### S8 Use of the Laplacian matrix for $A_m$

To study the control energy and locality, we create synthetic temporal networks by generating each snapshot randomly and independently according to the $G(n, p)$ model. We discard and regenerate any snapshot that is disconnected and assign link weights independently and randomly from $(0, 1)$. For the control energy we add self-loops with identical weight $a_m$ to all nodes, where
$a_m$ is chosen to stabilize the standalone dynamics of each snapshot $m$, reflecting the fact that most real systems have a stable state corresponding to the system’s mode of normal operation [40]. Note, however, that our underlying theory also works for unstable dynamics.

For the control energy and locality analyses of the main text, we employ the Laplacian matrix with self loops for the system matrix $A_m$ of each snapshot. Specifically, $L = (l_{ij})_{NN}$, where

$$l_{ij} = \begin{cases} w_{ij} & i \neq j \\ -\sum_{j=1,j\neq i}^{N} w_{ij} & i = j \end{cases}$$

and $w_{ij}$ is (randomly-chosen) weight of the edge from node $j$ to node $i$. For an arbitrary vector $\xi = (\xi_1, \xi_2, \cdots, \xi_N)^T$, we have

$$\xi^T L \xi = \sum_{i=1}^{N} \sum_{j=1}^{N} l_{ij} \xi_i \xi_j = \sum_{i=1}^{N} \sum_{j=1,j\neq i}^{N} (w_{ij} \xi_i \xi_j - w_{ij} \xi_i^2) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} (\xi_i - \xi_j)^2.$$ 

Thus we know when all $w_{ij} > 0$ ($w_{ij} < 0$), $L$ is negative (positive) semi-definite. Here, we let $A_m = a_m I + L$, where $a_m$ is chosen to stabilize the dynamics of each individual snapshot:

* When $w_{ij} > 0$, we can tune $a_m$ to make $A_m$ negative ($a_m < 0$), negative semi- ($a_m = 0$), and non-negative definite ($a_m > 0$, here when $a_m$ is sufficiently positive, $A_m$ can be positive definite) matrix $A_m$ since $a_m$ is the maximum eigenvalue of $A_m$.

* When $w_{ij} < 0$, we can tune $a_m$ to make $A_m$ positive ($a_m > 0$), positive semi- ($a_m = 0$), and non-positive definite ($a_m < 0$, here when $a_m$ is sufficiently negative, $A_m$ can be negative definite) matrix $A_m$ since $a_m$ is the minimum eigenvalue of $A_m$.

**S9  Locality of the optimal control trajectories for temporal networks**

For the optimal solution $c^*$ shown in (S12), we calculate the minimum control energy together with the optimal inputs to control a temporal network. Substituting the optimal inputs $u_m(t) = B_m^T e^{A_m^T (t_m-t)} c^*_m$ for $t_{m-1} \leq t < t_m$ over the snapshot $m$ where $m = 1, 2, \cdots, M$ into the

26
corresponding trajectory
\[ x(t_m) = e^{A_m \Delta t_m} x(t_{m-1}) + \int_{t_{m-1}}^{t_m} e^{A_m(t_m-s)} B_m u_m(s) \, ds, \]
we obtain the optimal trajectory
\[ x(t) = e^{A_m(t-t_{m-1})} x(t_{m-1}) + W_m[t_{m-1}, t] c^*_m \]
for \( t \in [t_{m-1}, t_{m-1} + \Delta t_m] \), along which the control cost is minimum, where \( W_m[t_{m-1}, t] = \int_{t_{m-1}}^{t} e^{A_m(t-s)} B_m B_m^T e^{A_m(t_m-s)} \, ds \) and \( e^{-A_m t_m} W_m[t_{m-1}, t_m] e^{-A_m t_m} \) is the gramian matrix of snapshot \( m \), and \( c^*_m \) is a constant vector of dimension \( N \). Note that \( c^*_m \) is given by \( c^* = S^T (S W S^T)^{-1} d \) with \( c^* = (c_1^T, c_2^T, \ldots, c_M^T)^T \).

In this section, we show control trajectories for temporal and static systems in two to three dimensions to give a visual understanding of the control non-locality of static networks. Figure S21 shows the full trajectories for an example two-dimensional system with two and five snapshots, for 100 different final states. Figure S22 shows the same for a three dimensional system. In agreement with the results presented in Fig. 4, we find that the length of trajectories for temporal networks is considerably less than that for static networks, independent of the choice of \( x_0 \) and the value of control distance.

We calculate \( L \) numerically according to
\[
L = \int_{t_0}^{t_f} \| \dot{x}(t) \| \, dt = \int_{t_0}^{t_f} \sqrt{x_1'^2(t) + x_2'^2(t) + \cdots + x_N'^2(t)} \, dt
\approx \frac{1}{t_{\text{step}}} \sum_{j=0}^{N} \sqrt{\sum_{i=1}^{N} \left[ x_i(t_j + t_{\text{step}}) - x_i(t_j) \right]^2},
\]
where \( t_{\text{step}} = 0.025 \) is the discretization time step.

In Fig. 4C for the technological network, we give the maximum magnitude of the state components (which dominates the length of the corresponding control trajectory). This can be expressed as
\[
L_i^* = \max_i \left| \sum_{j=0}^{1/t_{\text{step}}} \left[ x_i(t_j + t_{\text{step}}) - x_i(t_j) \right] \right|.
\]
Since in our numerical examples, $L_{i^*}$ is on the order of $10^{35}$ for the temporal network and $10^{64}$ for the static counterpart, looking at $x_1(t)$ is sufficient to demonstrate that temporal networks exhibit more local trajectories in this network. The corresponding state components in the case of 1, 2, and 3 driver nodes are shown in Fig. S23.

Similar to the earlier analysis of control energy in terms of different switching sequences (Sec. S6), for the different snapshot sequences shown in Fig. S20, we here give the corresponding control trajectories in Fig. S25. For the real (technological) network we adopted, Fig. S24 shows the locality as the order of snapshots is changed. As with the control energy, the locality is shown to be largely a function of the set of snapshots rather than their precise order.
Fig. S1: There is no theoretically determinate relation between the controllability of temporal and static networks. (A), A controllable temporal network corresponding to an uncontrollable static network. (B), A controllable static network corresponding to an uncontrollable temporal network. Here as in the main text, we assume that one input corresponds to exactly one driver node, meaning that $B$ is diagonal upon row permutation with a single entry equal to one in each column.
Fig. S2: Contact activity in empirical data. The curves show the contact activity (number of contacts over a 300s time window) for the ACM conference and student contacts, and over 10 seconds for ant interactions. For human interactions we observe the rhythm of day and night, while for ants the number of interactions shows little temporal variation, i.e. with no bursts or lulls.
Fig. S3: Degree distribution of the static networks corresponding to four kinds of empirical datasets. The static networks are aggregated from all contacts for the ACM conference and ant interactions. For protein and technological networks, the static networks are aggregated from all snapshots.
Fig. S4: Average degree of static networks as a function of snapshots aggregated. For the ACM conference and ant interactions network, we set $\Delta t = 1000s$ and 10s, which generates 213 and 144 snapshots, respectively. For the protein network (CC) we have 33 snapshots, with 50 for each of MF and BP. “ORI” denotes the original data sets. The different data sets show different patterns in how nodes acquire links as the system evolves, i.e. how $\langle k \rangle$ depends on $m$. For the ACM conference, $\langle k \rangle$ shows long plateaus corresponding to break periods in the conference (Fig. S2). For each network, we also show the effect of the randomization procedures discussed in Sec. S3: only RE changes the ultimate value of $\langle k \rangle$; TR and RPT change the pattern of increase of $\langle k \rangle$ but not the final value.
Fig. S5: Number of components of static networks as a function of snapshots aggregated. Dashed lines correspond to 20% of the number of nodes in the network. The intersections with the colored lines therefore denote the number of snapshots that must elapse to achieve full controllability using 20% of the network as driver nodes. Here, the number of components in the static counterpart can be no larger than the number of driver nodes when the temporal network is controllable.
Step 0: 
Downloading the time course gene expression data

Step 1: 
Filtering the gene expression data

Step 2: 
Constructing the PPI network

Step 3: 
Calculating $S_t$ and $S_s$ according to the Eq. (2)
Fig. S6: Schematic illustration of the calculation process over temporal and static protein-protein interaction (PPI) networks. Step 0: Download the raw time-series microarray data from the gene expression array (GSE4987) in the Gene Expression Omnibus (GEO). This dataset is in the form of a $6,297 \times 50$ matrix, including the expression profiles of 6,228 probes at 50 different time points. The probe sets are mapped to gene symbols according to the annotation file provided by Affymetrix and thus obtain 4,915 budding yeast *Saccharomyces cerevisiae* gene products [27]. Step 1: Filter the raw gene expression data by comparing the expression levels of genes at every time point to the active threshold obtained from the three-sigma principle [27]. Step 2: Construct the PPI network (snapshot) at every time point by only considering those interacting proteins in the global PPI network of *S. cerevisiae* that are present at those time points. The global PPI network of *S. cerevisiae* is downloaded from the Database of Interacting Proteins (DIP). This network consists of 5,023 proteins and 22,570 interactions. The static PPI network is obtained by aggregating all the snapshots. This figure is adapted from Ref. [41], Fig. 1. Step 3: Based on the sequences of the snapshots, the number of snapshots that temporal (or static) network needs to elapse (or aggregate) to reach fully controllable space can be calculated based on Eq. (2) in the main text. Note that in this work, we consider three small temporal PPI networks based on three gene ontology (GO) terms: cellular component (CC), molecular function (MF), biological process (BP), and in each temporal network, all the proteins share the GO term. Those three small temporal networks (denoted as CC, MF, BP) have 33, 50, and 50 snapshots, and 84, 74, and 85 proteins, respectively.
Fig. S7: Schematic illustration of methods used to randomize temporal networks. The rows depict six events (triplets) \((t,i,j)\) occurring over six discrete timestamps \(t_1\) to \(t_6\), involving four individuals (A, B, C, D) who are represented by red, green, blue, and gold nodes, respectively. For each method, a conceivable randomized instance of the original network is drawn with the time of each contact (or interaction) indicated on the corresponding edge. In Time Reversal (TR), the times of the contacts are reversed. In Randomly Permuted Times (RPT), we shuffle the times of the contacts. In Randomized Edges (RE), we iteratively choose pairs of contacts \((i,j)\) and \((i',j')\), and replace them with \((i,i')\) \((j,j')\) or \((i,j')\) \((i',j)\) with equal probability, provided the change results in neither self loops nor multiple edges. In Randomized Edges and Randomly Permuted Times (RERPT), we perform RE first and then RPT. In Randomly Distributed Times (RDT), we assign an entirely new set of timestamps \(t'\) (drawn from a given distribution) to the contacts. Note that, in TR and RPT, the inter-event times for a given node will change, but the total number of events for that node is unchanged. In RE, the degree and the contact times of each node may change. In RDT, the contact time of each event may change, while the degree of each node is unchanged. For more details, see Sec. S3.
Fig. S8: Faster paths to controllability in temporal networks when the original data is randomized. We find that $S_t < S_s$ for various $\Delta t$ both for the original sequence of snapshots (Fig. 2), and when the sequence of interactions is randomized using several null models [19]: TR (Time Reversal), RPT (Randomly Permuted Times), RE (Randomized Edges) and RERPT (Randomized Edges and Randomly Permuted Times) (see Sec. S3 for the randomization procedure). Parameters and other details of this analysis are the same as those used in Fig. 2 of the main text.
Fig. S9: Temporal networks reach controllability faster independent of the value of $\Delta t$. Shown are $S_t$ and $S_s$ for the ACM conference (A), ant interactions (B), and student contacts (C) networks. Our result that temporal networks reach controllability faster holds over a wide range of $\Delta t$. Parameters and other details of this analysis are the same as those used in Fig. 2 of the main text.

Fig. S10: Temporal networks reach controllability faster in the ant interaction network. $S_t$ is not bigger than $S_s$ for ants interactions even when each contact is equipped with a finite duration. Here each time point is scaled up by a factor of 60 and every antenna-body interaction is assumed to last 20s. Parameters and other details of this analysis are the same as those used in Fig. 2 of the main text.
**Fig. S11:** Temporal networks reach controllability faster regardless of the number of driver nodes used. The static versions of the technological and protein networks sometimes remain uncontrollable at the final snapshot using sets of driver nodes corresponding to 20% of the network, as done in Fig. 2. Here we calculate $S_t$ and $S_s$ by instead using sets of driver nodes corresponding to 80% of the network size. Our demonstration that temporal networks reach controllability faster than their static counterparts remains true for these larger sets of driver nodes. Predictably, both $S_t$ and $S_s$ decrease relative to Fig. 3C and 3D. Nonetheless, we still observe cases where $S_s = M$, meaning that the static network remains uncontrollable even after the final snapshot is aggregated. This is true even though a full 80% of the nodes are directly controlled. In contrast, the temporal version of the network is controllable, and with only 20% of the network as driver nodes. Each bar corresponds to $10^3$ random sets of driver nodes.

**Fig. S12:** Relation between $S_s$ and $S_t$ based on a toy model. For $N_s$ sequences randomly selected from a set of $M$ snapshots, we calculate the $S_t$ and $S_s$ with fixed setting of the driver nodes. And the probability of $S_t < S_s$ and $S_t > S_s$ is calculated over all the selected sequences. Here $N = 50$, $N_s = 500$, $p = 0.03$ and $N_d = 3$. 
Fig. S13: Faster paths to controllability in temporal networks. For all datasets, we randomly distribute time stamps within the observation time window \([t_{\text{min}}, t_{\text{max}}]\) of the original dataset, using Poisson, uniform, and normal distributions. These distributions were selected for illustrative purposes, and do not represent the event times generated by any specific point process. Regardless of how the timestamps are distributed, we find that temporal networks reach controllability faster than the corresponding static networks on all datasets we considered. Note that after generating a series of time stamps that obey Poisson (mean: 3) and normal (mean: 0.5, standard deviation: 1) distributions, we adjust every time stamp to the time window \([t_{\text{min}}, t_{\text{max}}]\) by appropriate transformation (see Sec. S4 for details). Other parameters are the same as those in Fig. 2 of the main text. The robustness of the results has been checked.
Fig. S14: The minimum eigenvalue of $W_{\text{eff}}$ dominates the control energy. For different $\Delta t$, all eigenvalues of $W_{\text{eff}}$ are given by gray points for (A) temporal and (B) static networks, where the minimum eigenvalues are enlarged in red and blue, respectively. The eigenvalues of $W_{\text{eff}}$ vary over many orders of magnitude, implying that the average control energy is dominated by the worst-case direction (corresponding to the $\lambda_{\text{min}}$) of $W_{\text{eff}}$ (Eq. (S13)). Since $\lambda_{\text{min}}$ is much greater for the temporal network, the energy required to move in typical control directions is thus expected to be less than in the corresponding static network. Here the system parameters are the same as those used in Fig. 3A of the main text.

Fig. S15: Temporal networks require less control energy compared to static networks. Counterpart to Fig. 3 of the main text with more fractions of nodes are controlled: (A) 0.3 and (B) 1. For the technological network, the number of driver nodes is 1 in (C) and 7 in (D). Here the system parameters are the same as those used in Fig. 3 of the main text.
Fig. S16: Temporal networks require less control energy compared to static networks. Counterpart to Fig. 3 of the main text with $N = 10, M = 2, \bar{k} = 4, a_1 = -3,$ and $a_2 = -1$.

Fig. S17: Temporal networks require less control energy compared to static networks. Counterpart to Fig. 3 of the main text with $N = 10, M = 5, \bar{k} = 4,$ and $a_i = -2$ for $i = 1, 2, \cdots, 5$. 
Fig. S18: Difference in control energy for a real network. We aggregate the total $M = 50$ snapshots of the 1-ip6 network into $M = 2$ snapshots. We show the distribution of the control energy over 300 randomly-selected final states with unit distance away from $x_0 = 0$, for varying $\Delta t$ and numbers of driver nodes $N_d$ (blue: static, red: temporal). The corresponding average energies $\langle E \rangle$ are denoted in each panel. We find that control energy decreases as either $N_d$ or $\Delta t$ increases. Here we choose $a_1 = -1$ and $a_2 = -2$. 
Fig. S19: Difference in control energy for a real network. Here we change the order of the snapshots in the temporal network studied in Fig. S18. All other parameters and notations are the same as those in Fig. S18.
Fig. S20: The energy for controlling temporal networks shows little variability with respect to the switching sequence. For a set of $M$ snapshots, we enumerate all the possible sequences of snapshots, and for every sequence, we calculate the control energy distribution (calculation method is the same as that in Fig. S18). Here $N = 10$, $M = 3$, $N_d = 1$, $\Delta t = 10^{-6}$, and we choose $a_1 = -1$, $a_2 = -2$, $a_3 = -3$. We present the distribution of the control energy (red: temporal, blue: static) for six sequences here ((A): $A_1, A_2, A_3$, (B): $A_1, A_3, A_2$, (C): $A_2, A_1, A_3$, (D): $A_2, A_3, A_1$, (E): $A_3, A_1, A_2$, (F): $A_3, A_2, A_1$). The numbers on each panel are the average control energy over the final states we choose, for which we find that the control energy varies less than 10% over the possible switching sequences.
Fig. S21: Additional control trajectories for temporal and static networks with two and five snapshots. We select 100 trajectories from \( x_0 = 0 \) (indicated by a star) to \( x_f \) with \( \| x_f \| = 10^{-3} \) (i.e. uniformly along the gray curve), for a randomly-generated temporal network and its static counterpart with (A) \( N = 2, M = 2 \), and (B) \( N = 2, M = 5 \).

Fig. S22: Control trajectories in a three dimensional system. Each trajectory is from \( x_0 \) (star) to a given \( x_f \) over \( 0 \leq t \leq 1 \) for a static (A) and a temporal (B) network. We consider a total of 100 such randomly chosen \( x_f \) located on a sphere (grey) centered on \( x_0 \) with a radius of \( \delta = 10^{-3} \). We only plot those final states on the equator for clarity. Here we consider three dimension systems and two snapshots for visualization, and the robustness of the results has been tested.
Fig. S23: Locality of control trajectories in a real network. The panels show, for different numbers $N_d$ of driver nodes, the node states $x_i(t)$ as a function of time for control in the temporal (red) and static (blue) version of the ad hoc mobile communication network (1-ip6). Here $x_0 = 0$ and $x_f$ is taken to be $(1, ..., 1)^T/\sqrt{N}$ for $N = 34$ nodes. The total length of the control trajectory $L$ is denoted in red (temporal) and blue (static). We see that the true temporal version of this network exhibits considerably more local control trajectories than its aggregated counterpart, in line with the results shown for synthetic networks in the main text (Fig. 4C). Moreover, $L$ decreases as $N_d$ increases for both the temporal and static network. Here $a_i = -1$ and $M = 2$. 
Fig. S24: Locality of control trajectories in a real network. Counterpart of Fig. S23, and here we change the order of the snapshots. All other parameters and notations are the same as those in Fig. S23.
Fig. S25: Locality of control trajectories for different switching sequences of a temporal network. Counterpart of Fig. S20, where here we calculate the corresponding control trajectories for each different snapshot sequence. We find that the temporal network exhibits more local control trajectories regardless of the precise order of snapshots.
References and Notes


