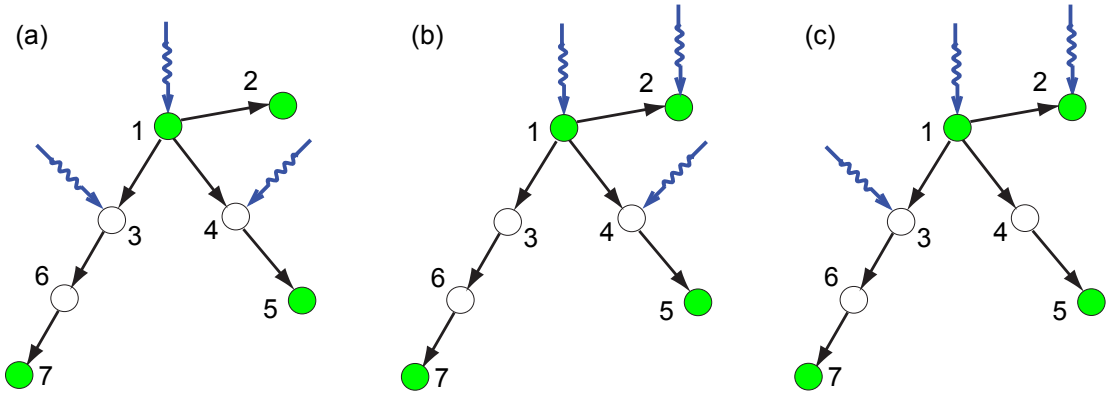
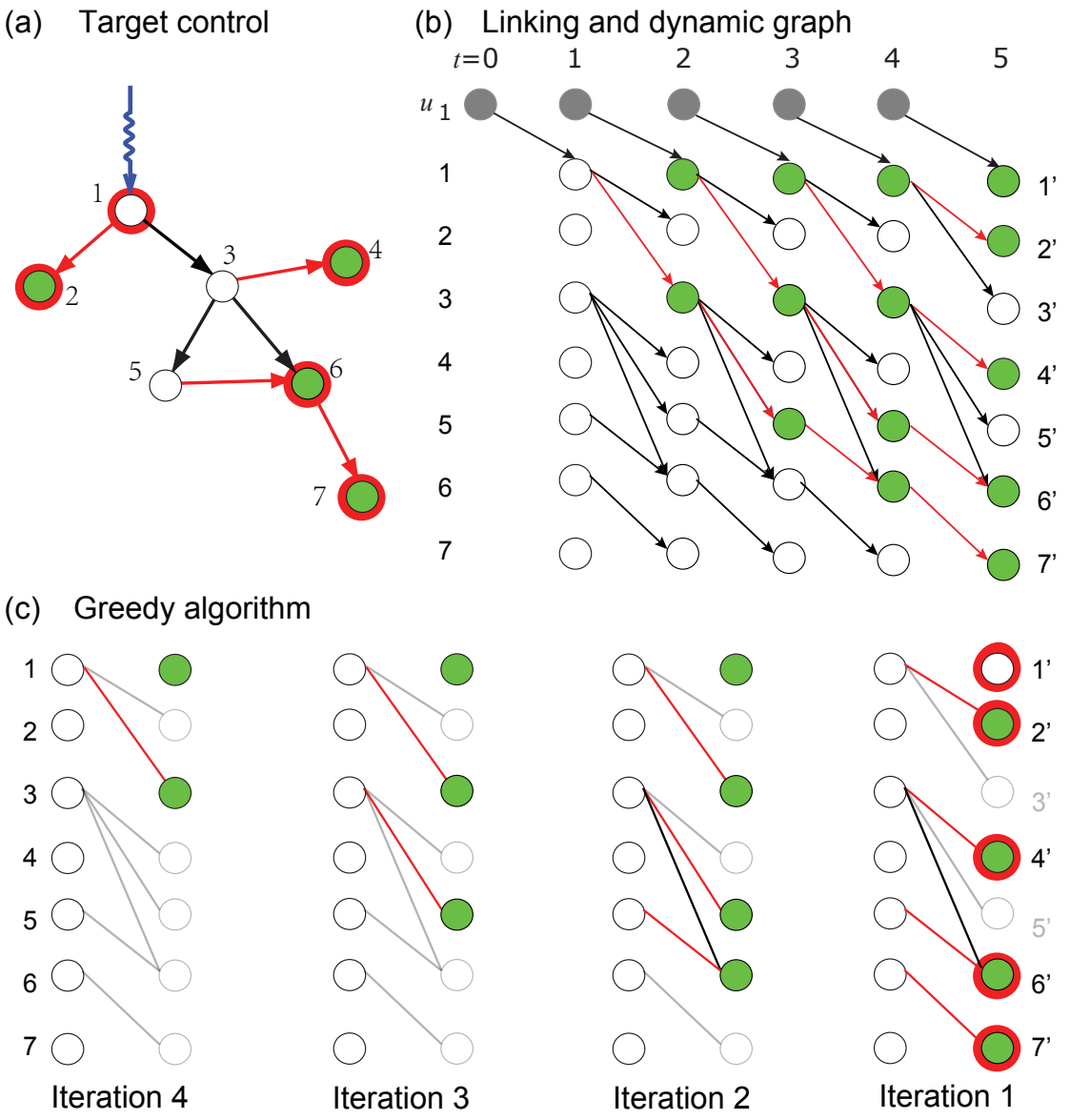
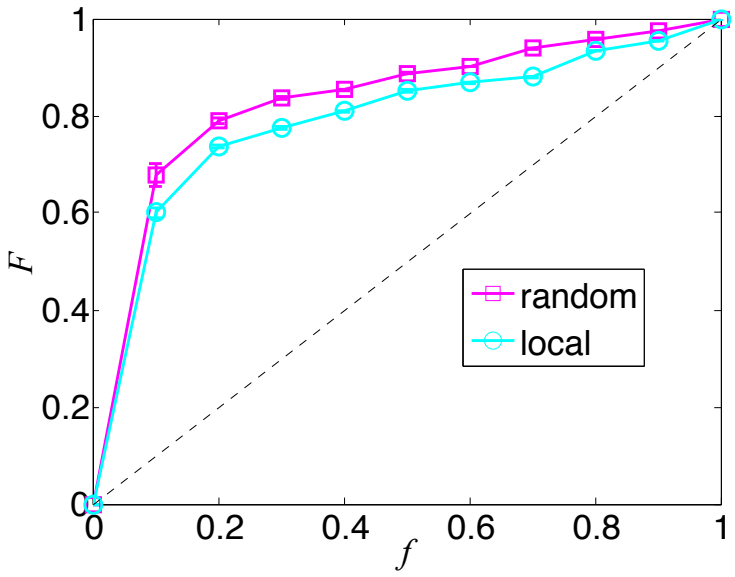


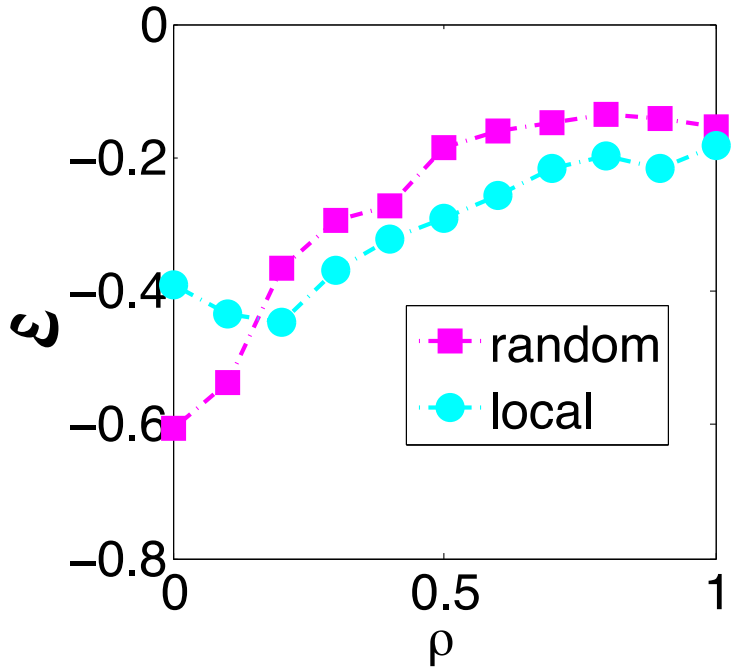
Supplementary Figure 1: **Controlling a directed tree of seven nodes**. To control the whole network we need at least 3 driver nodes, which can be either $\{1, 3, 4\}$ (a), $\{1, 2, 4\}$ (b) or $\{1, 2, 3\}$ (c), predicted by the structural control theory. If instead we want to control a subset of nodes, e.g. $\{1, 2, 5, 7\}$ (the green nodes) with a minimum set of nodes, we need to solve the target control problem. The upper bound obtained by structural control theory indicates that we need at least three driver nodes (the same sets shown in (a), (b), and (c)). But, in reality, we only need one driver node (node 1), which can be obtained from both the k-walk theory and the greedy algorithm.



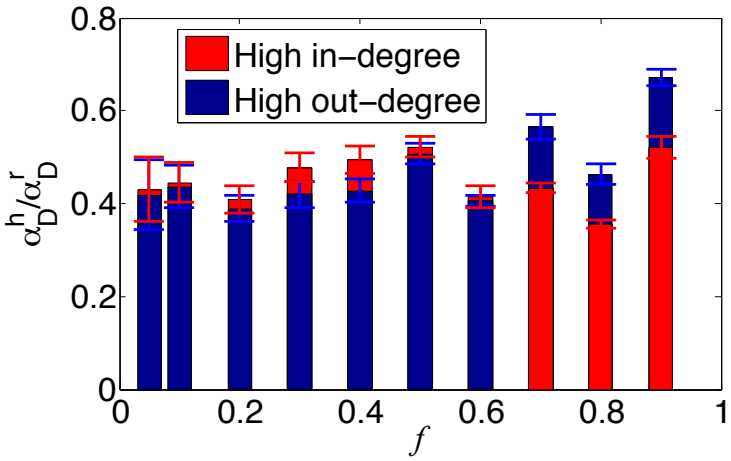
Supplementary Figure 2: **Greedy algorithm is closely related to the concept of linking in dynamic graphs.** (a) Node set $\{1, 2, 4, 6, 7\}$ can be controlled by node 1. (b) Node 1 can control node set $\{1, 2, 4, 6, 7\}$ because there are 4 disjoint linkings from node 1 to nodes $\{1, 2, 4, 6, 7\}$. (c) Node set $\{1, 2, 4, 6, 7\}$ can be controlled by node 1 identified from the greedy algorithm.



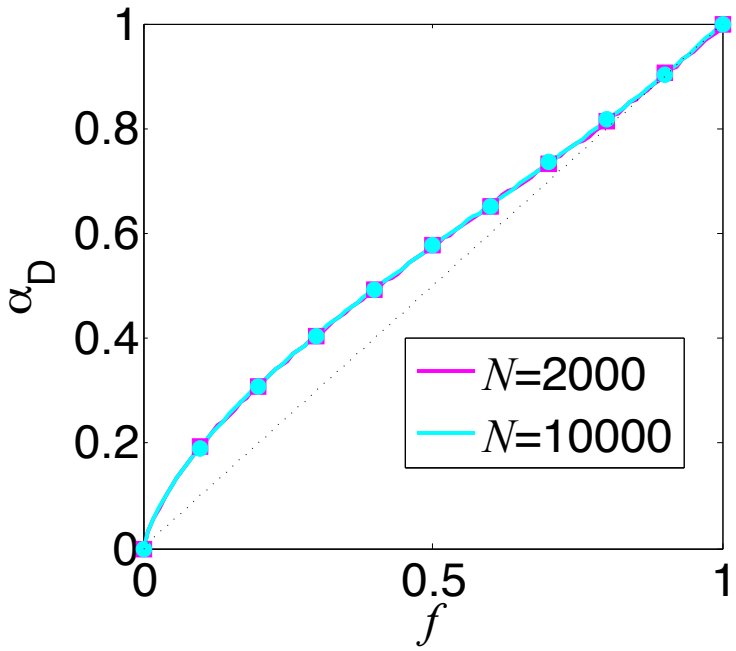
Supplementary Figure 3: **Relative size of the controllable subsystem vs. the fraction of target nodes.** F denotes the relative size of the controllable subsystems. f denotes the target node fraction. The calculation is done for ER networks with $N = 1000$ and mean degree $\langle k \rangle = 4$. The result is averaged over 6 realizations. The error bars are of the size of the symbols.



Supplementary Figure 4: **The effect of an emerging hub on the target control efficiency.** Starting from an ER random network with mean degree $\langle k \rangle = 10$ and number of nodes $N = 1000$ for 200 realizations, we rewire a ρ fraction of nodes to a particular node i. Node i hence will emerge as a hub if ρ is very high. The error bars are of the size of the symbols.



Supplementary Figure 5: **Controlling hubs in scale-free networks.** Here we calculate the ratio between $\alpha_{\mathrm{D}}^{\mathrm{h}}$ (the target controllability parameter of controlling the top f fraction of highest degree nodes) and $\alpha_{\mathrm{D}}^{\mathrm{r}}$ (the target controllability parameter of random control a f fraction of nodes).



Supplementary Figure 6: **Finite size effect on target controllability.** For ER networks with mean degree $\langle k \rangle = 5.6$ and two different sizes, we show the normalized fraction of driver nodes (α_{D}) in function of the target node fraction f for random node selection scheme.

Supplementary Note 1: Target Control

We consider linear time-invariant (LTI) systems of the form [1, 2]

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx. \end{cases} \tag{1}$$

where $x \in \mathbb{R}^N$, $y \in \mathbb{R}^S$ and $u \in \mathbb{R}^M$ are the state vector, output vector and control inputs respectively. The state matrix, output matrix and input matrix are given, respectively, by $A \in \mathbb{R}^{N\times N}$, $C \in \mathbb{R}^{S\times N}$ and $B \in \mathbb{R}^{N\times M}$. We will denote the linear control system Supplementary Eq. (1) as a triplet (A, B, C). The dimension of its controllable subspace $\mathcal{C}$ is denoted as $\dim(\mathcal{C}) = d(A, B, C)$.

Definition 1 (Output controllability)**.** *A system is output controllable if we can move its output from any initial condition to any final condition in a finite time interval with a suitable control input.*

Theorem 1 (Output controllability theorem [2])**.** *The LTI system (A, B, C) is output controllable if and only if its output controllability matrix has full row rank*

$$d(A, B, C) = \text{rank}[C(B, AB, A^2B, ..., A^{N-1}B)] = S. \tag{2}$$

Target control can be viewed as a special output control problem, where $y = Cx$ is the state of a target node set $\{x_{c_1}, \ldots, x_{c_s}\}$. In other words, the matrix $C \in \mathbb{R}^{S\times N}$ satisfies $C_{i,c_i} = 1$ and all other elements are zeros, where c_i $(i = 1, 2, \ldots, S)$ is ith target node. In practical terms, target controllability can be posed as identifying the minimum set of driver nodes such that Supplementary Eq. (2) is satisfied. To directly apply Supplementary Eq. (2) we need to know all the matrix elements in A, B and C, which for most networks are either unknown or known only approximately. Even if we know all the matrix elements in A, B and C, it is still a computationally prohibitive task to identify the minimum set of driver nodes for large networks, requiring to test $2^N - 1$ distinct node combinations. To bypass the need to know the link weights, we adopt the structural control theory developed decades ago [3].

The system (A, B) is structurally controllable if it is possible to choose the non-zero elements (or weights) in A and B such that the system satisfies Kalman's rank condition [3]. A structurally controllable system can be shown to be controllable for almost all weight combinations, except for some pathological cases with zero measure. Thus, structural controllability helps us to overcome our inherently incomplete knowledge of the link weights in A and B.

Definition 2 (Structurally equivalent)**.** *Two matrices $A = (a_{ij})$ and $\hat{A} = (\hat{a}_{ij})$ of the same size are said to be structurally equivalent if their non-zero entries coincide in position, i.e., $a_{ij} = 0$ iff $\hat{a}_{ij} = 0$ for all i and j. Two systems (A, B, C) and $(\hat{A}, \hat{B}, \hat{C})$ are said to be structurally equivalent if the corresponding pairs of matrices are.*

Definition 3 (Generic dimension)**.** *The generic dimension $gd(A, B, C)$ of the output state space is defined as*

$$gd(A, B, C) = \max_{\hat{A},\hat{B},\hat{C}} \{d(\hat{A}, \hat{B}, \hat{C})\}, \tag{3}$$

where $\hat{A}, \hat{B}, \hat{C}$ are structurally equivalent of A, B, C respectively.

Consider a directed network $\mathcal{G}(V, E)$ with $N = |V|$ nodes and $L = |E|$ links. If there exists a directed link from node i to node j, then $a_{ji} \neq 0$ in the state matrix A. A target node set of size S is denoted as $\mathcal{C} = \{c_1, c_2, ..., c_S\} \subseteq V$. In order to control the S target nodes, we need to drive M nodes $\mathcal{B} = \{b_1, b_2, \cdots, b_M\} \subseteq V$.

Without loss of generality, we consider $\mathcal{C} = \{1, 2, ..., S\}$ and the output state vector $y = [x_1, x_2, ..., x_S]^\top$, then the output matrix can be written as $C = [\mathbf{I}, \mathbf{0}]$, where $\mathbf{I}$ is an identity matrix of $S \times S$, and $\mathbf{0}$ is an $S \times (N - S)$ matrix with all entries zero. We denote the state variables of the remaining (non-target) nodes as $z = [x_{S+1}, x_{S+2}, ..., x_N]^\top$. Then we can decompose $\dot{x} = Ax + Bu$ as

$$\begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A^{(11)} & A^{(12)} \\ A^{(21)} & A^{(22)} \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \tag{4}$$

where $A = \begin{bmatrix} A^{(11)} & A^{(12)} \\ A^{(21)} & A^{(22)} \end{bmatrix}$, $A^{(11)}$ represents the topology of S target nodes in $\mathcal{C}$, $A^{(22)}$ represents the topology of the $N - S$ non-target nodes in set $\bar{\mathcal{C}} := V \setminus \mathcal{C}$. The non-zero entries in $A^{(21)}$ and $A^{(12)}$ represent the links between target and non-target nodes.

Supplementary Note 2: k-walk theory

Consider a directed tree-like network that has at most one directed path from any node u to any other node v. (Note that if there is only one node with in-degree 0, i.e., a root node, such a directed tree is called an *arborescence* in graph theory.) The main result of k-walk theory is that for linear time-invariant dynamics on such directed trees *a single node u can fully control a set of target nodes provided the path length from node u to each target node is unique*. This result enables us to develop an efficient algorithm to identify the controllable subsystems of any single node in directed trees. Here, controllable subsystems of node i mean the maximum sets of nodes that can be fully controlled by directly controlling node i only. Note that for directed tree-like networks k-walk theory can find some controllable subsystems that would be totally missed by the previous method based on control centrality [4]. For example, as shown in Supplementary Figure 1, by calculating the control centrality of node 1 we can only obtain one controllable subsystem $\{1, 3, 6, 7\}$. Using k-walk theory, however, we can identify the following controllable subsystems $\{1, 3, 6, 7\}$, $\{1, 2, 5, 7\}$, $\{1, 2, 6, 7\}$, $\{1, 4, 5, 7\}$, $\{1, 4, 6, 7\}$, and $\{1, 3, 5, 7\}$.

In this section, we prove the main result of k-walk theory and provide an efficient algorithm to find all controllable subsytems of any single node in a directed tree-like network. To achieve that, we first introduce some basic concepts. Though some of the concepts are originally defined on undirected networks, in this section we focus on directed networks or digraphs.

Definition 4 (Reachable set)**.** *The reachable set $\mathcal{R}_i$ of node i contains all the nodes that node i can reach through directed paths. We denote $r_i = |\mathcal{R}_i|$ as the total number of nodes that node i can reach.*

Definition 5 (Controllable set)**.** *A controllable set $\mathcal{C}_{i,\alpha}$ of node i contains a maximum set of nodes that can be fully controlled by node i. Here, $\alpha \in [1, s_i]$ and s_i is the totoal number of controllable sets of node i.*

Note that $|\mathcal{C}_{i,\alpha}| = c_i, \forall \alpha \in [1, s_i]$ and c_i is the control centrality of node i, i.e., the dimension of the largest controllable subspace if we control node i only. One can show that $\cup_{\alpha=1}^{s_i} \mathcal{C}_{i,\alpha} \subseteq \mathcal{R}_i$, i.e., the union of all the controllable sets of node i is just the reachable set of node i. The reachable set $\mathcal{R}_i$ of node i can be obtained by a breadth first search, from which r_i can be calculated as well. The control centrality and one controllable set of node i can be calculated by solving the maximum-weight cycle partition problem via linear programming [4]. Yet, there was no efficient method to enumerate all the controllable sets of a given node.

Definition 6 (Walks)**.** *A walk is an alternating sequence of nodes and links in the form* $v_0, e_1, v_1, ..., v_{n-1}, e_n, v_n$ *such that for* $1 \leq i \leq n$*, the link* $e_i = (v_{i-1} \to v_i)$ *has source node* v_{i-1} *and target node* v_i*. A walk is closed if its first and last nodes are the same, and open if they are different. The length of a walk is its number of links.*

Definition 7 (Path, simple path, cycle)**.** *(i) A path is an open walk. (ii) A simple path is an open walk in which no nodes are repeated. (iii) A cycle is a closed walk that starts and ends at the same node but otherwise has no repeated nodes or links.*

For a directed network G with adjacency matrix $A = [a_{ij}]$, $a_{ij} = 1$ if there is a link $(v_i \to v_j)$ and 0 otherwise. We denote d_{ij} as the length of a walk from node i to node j. We have $d_{ij} = k$ if the (i, j) entry of matrix A^k is non-zero. For general directed networks, d_{ij} might have multiples values, because there could be many different walks connecting node i and node j. For a directed tree, all the paths are simple and consequently d_{ij} is single-valued for any node pair (i, j) in T.

Theorem 2. *For a directed tree, node* i *can control all the* c_i *nodes in the set* $\mathcal{C}_{i,\alpha} = \{i_1, ..., i_{c_i}\}$*, where* $d_{ii_k} = k - 1$ *and* $k \in [1, c_i]$*. Note that* i_1 *denotes node* i *itself.*

Proof. According to structural output controllability theorem [5], node i can control all the nodes in $\mathcal{C}_{i,\alpha}$, if the generic dimension of the output controllability matrix $C := c_{i,\alpha}[b_i, Ab_i, A^2b_i, ..., A^{N-1}b_i]$ has rank c_i, i.e.,

$$gd(C) = gd(c_{i,\alpha}[b_i, Ab_i, A^2b_i, ..., A^{N-1}b_i]) = c_i. \tag{5}$$

Here, $c_{i,\alpha} = \mathbf{I}(\mathcal{C}_{i,\alpha})$ denotes a $c_i \times N$ matrix that contains the $\{i_1, ..., i_{c_i}\}$th rows of the identify matrix $\mathbf{I}$, b_i is ith column of the identity matrix. Note that the $N \times 1$ vector $A^k b_i$ contains non-zero entries corresponding to those nodes with $d_{ij} = k - 1$. Since the network is a directed tree and the set $\mathcal{C}_{i,\alpha}$ contains only one node that satisfies $d_{ij} = k - 1$, we have $c_{i,\alpha}A^k b_i = \beta_k \mathbf{I}_{i_k}$ where β_k is a non-zero constant, and $\mathbf{I}_{i_k}$ represents the i_k-column of the identity matrix. Hence, we have

$$gd(C) = gd[\beta_1 \mathbf{I}_{i_1}, ..., \beta_{c_i} \mathbf{I}_{i_{c_i}}]. \tag{6}$$

Since $\mathbf{I}_{i_1}, ..., \mathbf{I}_{i_{c_i}}$ are all independent, $gd(C) = c_i$ and the subsystem represented by the set $\mathcal{C}_{i,\alpha}$ is controllable by controlling node i only. □

Now we propose an efficient algorithm to find all the controllable subsystems of node i in a directed tree: (1) Calculate the distance d_{ij} between node i and any other node j in the tree. (2) According to its distance from i, classify a node j from a distance-class $\mathcal{D}$ where all the nodes in

$\mathcal{D}$ have the same distance to node i. (3) Assume there are in total P distance-classes $\mathcal{D}_1, \cdots, \mathcal{D}_P$, with size $D_1, \cdots, D_P$ respectively. Choosing one node from each distance-class $\mathcal{D}$ will form a controllable set of node i. Note that the total number of controllable sets of node i is given by $D_1 \times \cdots \times D_P$.

Supplementary Note 3: Upper bound, lower bound, and greedy algorithm

In this section, we derive the upper and lower bounds of the number of driver nodes required for target control. We also provide a greedy algorithm to find an approximately minimum set of driver nodes for target control.

Theorem 3 (Structural state variable controllability theorem [6]). *Consider a structural system in the form of Supplementary Eq. (4). The target state y is structurally controllable if the target nodes are covered by a cactus structure underlying the directed network corresponding to the controlled system (A, B).*

Theorem 3 enables us to derive the upper bound of the minimum number of control inputs needed for target control.

Algorithm 1 (Upper bound). *(1) According to the minimum input theorem [7], we can find at least one minimum set $\mathcal{D}$ of driver nodes to control the whole network. Each driver node is connected to a root of a cactus. (2) Calculate the minimal number of cacti needed to cover all the target nodes.*

The lower bound of the minimum number of driver nodes for target control can be derived based on the concept of dilation in structural control theory [3]: two target nodes that share the same incoming neighbor set need at least two independent control inputs. Hence we can formalize the target control problem as a bipartite matching problem, see Figure 1(d) in the main paper.

Algorithm 2 (Lower bound). *(1) Build a bipartite graph $\mathcal{B}$, where the right side $\mathcal{R}$ consists of all the target nodes, and the left side $\mathcal{L}$ consists of all the nodes that can reach the target nodes. There is a link between node $u \in \mathcal{L}$ and $v \in \mathcal{R}$ if there is a link $u \to v$ in the original directed network $\mathcal{G}$. (2) Find the maximum matching in $\mathcal{B}$. The unmatched nodes in $\mathcal{R}$ are just the driver nodes.*

One can show that the minimal number of driver nodes for target control is no less than the number of driver nodes derived from the lower bound algorithm.

The greedy algorithm is based on the lower bound algorithm. Actually, the latter can be considered as the first step of the former. Denote that target node set as $\mathcal{C}^0$.

Algorithm 3 (Greedy algorithm). *(1) At step $t = 0$, use algorithm 2 to find the lower bound of driver nodes to control the target nodes, i.e., the unmatched nodes in the right side of the bipartite graph, denoted as D_0. Then we can find all the matched nodes on the left side as the new target nodes $\mathcal{C}^1$.*

(2) If $\mathcal{C}^1 = \emptyset$, stop, we obtain the driver node set D_0, and number of driver nodes $P_{\mathrm{D}} = |D_0|$. If $\mathcal{C}^1 \neq \emptyset$, go to (3).

(3) At step $t \geq 1$, use algorithm 3 to find the lower bound of driver nodes for the new target nodes $\mathcal{C}^t$, the unmatched nodes in the right side are the driver node set D_t. Then we can find all the matched nodes on the left side as the new target nodes $\mathcal{C}^{t+1}$. Go to (4).

(4) If $\mathcal{C}^{t+1} = \emptyset$, stop, we obtain the driver node set $\cup_{j=0}^{t} D_j$, and number of driver nodes $P_{\mathrm{D}} = |\cup_{j=0}^{t} D_j|$. If $\mathcal{C}^{t+1} \neq \emptyset$, go to (3).

Figure 1(d) in the main paper illustrates the process of greedy algorithm. Note that the greey algorithm is closely related to the concepts of linking and dynamic graph in structural control theory [8].

Definition 8 (Dynamic graph and linking [8])**.** *A directed graph associated with A, B, and N is the dynamic graph $\bar{G} = \bar{G}_N$ defined on the set of nodes $\bar{V} = \bar{V}_A \cup \bar{V}_B$ where $\bar{V}_A = V_{A,1} \cup \ldots \cup V_{A,N}$, $\bar{V}_B = V_{B,0} \cup \ldots \cup V_{B,N-1}$, $V_{A,t} = \{v_{it} : i = 1, \ldots, n\}, t = 1, \ldots, N$, and $V_{B,t} = \{v_{n+j,t} : j = 1, \ldots, m\}, t = 0, \ldots, N-1$. The set of directed edges of $\bar{G}$ is given by $\{v_{jt}v_{i,t+1} : a_{ij} \neq 0, t = 1, \ldots, N-1\} \cup \{v_{n+j,t}v_{i,t+1} : b_{ij} \neq 0, t = 0, \ldots, N-1\}$. A collection of node disjoint paths from $\bar{V}_B$ to $V_{A,N}$ is called linking. The size of a linking is the number of paths in it.*

The sufficiency of the above greedy algorithm can be proved by invoking Theorem 1 of [8] (Supplementary Fig. 2). Note that the greedy algorithm may find a set of driver nodes that can actually control a larger node set than the target node set itself.

Supplementary Note 4: The effect of hubs and network size

In order to understand if the peaks observed in Figure 5 of the main text might be due to the emergence of hubs, we perform the following analysis.

First, we start from an ER network with node set V and randomly select one node $i \in V$. Then we randomly select a ρ fraction nodes from the node set $V \setminus i$ and rewire those nodes to node i, preserving the in-degree and out-degree of those nodes. Hence the mean degree of the network is fixed. We calculate the target control efficiency of the rewired network. As shown in Supplementary Figure 4, the result implies that the presence of hubs increases the target control efficiency.

Second, we consider scale-free networks and study the case of choosing the top f fraction of highest in- or out-degree nodes as the target nodes. The results are shown in Supplementary Figure 5, where $\alpha_{\mathrm{D}}^{\mathrm{h}}$ denotes the target controllability parameter of controlling the top f fraction of the highest in- or out-degree nodes and $\alpha_{\mathrm{D}}^{\mathrm{r}}$ denote the target controllability parameter of randomly chosen f fraction of nodes. We find that, in general, controlling hubs requires less driver nodes. Interestingly, if we choose a small fraction of nodes ($f \leq 60\%$), controlling high out-degree nodes is easier than controlling high in-degree nodes, but the opposite is true if we control a large fraction of nodes ($f \geq 70\%$).

Supplementary References

[1] Aström, K. J. and Murray, R. M. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, (2010).

[2] Dorf, R. C. *Modern control systems*. Addison-Wesley Longman Publishing Co., Inc., (1991).

[3] Lin, C.-T. Structural controllability. *Automatic Control, IEEE Transactions on* **19**(3), 201–208 (1974).

[4] Liu, Y.-Y., Slotine, J.-J., and Barabási, A.-L. Control centrality and hierarchical structure in complex networks. *Plos one* **7**(9), e44459 (2012).

[5] Murota, K. and Poljak, S. Note on a graph-theoretic criterion for structural output controllability. *Automatic Control, IEEE Transactions on* **35**(8), 939–942 (1990).

[6] Blackhall, L. and Hill, D. J. On the structural controllability of networks of linear systems. In *2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 245–250, (2010).

[7] Liu, Y.-Y., Slotine, J.-J., and Barabási, A.-L. Controllability of complex networks. *Nature* **473**, 167–173 (2011).

[8] Poljak, S. On the generic dimension of controllable subspaces. *Automatic Control, IEEE Transactions on* **35**(3), 367–369 (1990).