scientific reports

OPEN

Network control by a constrained external agent as a continuous optimization problem

Jannes Nys[1,2], Milan van den Heuvel[3]✉, Koen Schoors[3] & Bruno Merlevede[3]

Social science studies dealing with control in networks typically resort to heuristics or solely describing the control distribution. Optimal policies, however, require interventions that optimize control over a socioeconomic network subject to real-world constraints. We integrate optimisation tools from deep-learning with network science into a framework that is able to optimize such interventions in real-world networks. We demonstrate the framework in the context of corporate control, where it allows to characterize the vulnerability of strategically important corporate networks to sensitive takeovers, an important contemporaneous policy challenge. The framework produces insights that are relevant for governing real-world socioeconomic networks, and opens up new research avenues for improving our understanding and control of such complex systems.

Networks are ubiquitous in modern society. The interconnectedness of networks has the potential to magnify the impact of a local event or intervention on the system as a whole. Such impact has been shown across multiple fields, from cascades in financial systems[1], to service adoption in social networks[2]. This raises the question how (a set of) nodes can gain influence or control over certain states in socioeconomic networks (e.g. decisions in companies, opinions of people). Across a wide range of systems (e.g. financial systems, governmental systems), insights into this process of gaining influence/control would improve our understanding of system stability, facilitate the design of optimal interventions to stabilize these systems or steer them in desirable directions. What is the optimal targeting strategy to maximize adoption of products or services in a socioeconomic network? How vulnerable is the control over strategic sectors to hostile takeovers? In this paper, we present a framework that reformulates these questions as a continuous optimization problem. This allows us to combine network science with state-of-the-art optimisation methods from deep learning to analyze the control of complex networks in a scalable and flexible manner.

Related literature. In the social sciences, and especially the field of social influence analysis, control is often viewed as a property that (instantly) flows through the system from node to node, like influence on product adoption in social networks, or corporate control through share holdings in ownership networks[2–4]. In social influence analysis, influence refers to the ability of an entity to change the probabilities of states of other entities in the network, (e.g. influencers that make their followers buy items). Here, we use the term control in a broad sense to refer to influence over the distribution of states in the system, unless explicitly mentioned otherwise.

Information diffusion models model how information flows through the network[5]. The literature uses algorithms on these networks to solve the social influence maximization (SIM) problem[6], defined as: *the problem of finding the minimal subset of nodes, the seed nodes, in a socioeconomic network that could maximize the spread of influence*, which is NP-hard. This has caused the literature to mostly focus on solutions that are either approximate or use heuristics. These solutions do not scale well for large networks[4].

Another domain where the topic of network control has been studied is (structural) network controllability. Herein, the goal is to find a optimal set of nodes, the driver nodes, that have the potential to independently control all states (on the nodes or edges) in the network in finite time[7–9], or to independently control a target subset of these states[10,11]. The control is exerted by injecting an external signal into the driver nodes, which then propagates throughout the network, as dictated by a set of linear differential equations. In this literature the driver nodes are said to have control if they are capable of independently steering the states of all (target[10]) nodes, or edges[12] in the network to any desired final state in finite time. Control in this strict sense seems a useful concept for designing policy interventions in socioeconomic networks, and the method has been demonstrated on such networks in

[1]IDLab, Department of Computer Science, University of Antwerp - imec, 2000 Antwerp, Belgium. [2]Department of Physics and Astronomy, Ghent University, 9000 Ghent, Belgium. [3]Department of Economics, Ghent University, 9000 Ghent, Belgium. ✉email: Milan.vandenHeuvel@UGent.be

the past[13–15]. Controllability, however, holds implicit communication assumptions which cause the results to have little meaning in socioeconomic networks[15]. The assumption originates from the independence condition that is imposed on the control[16] and entails that, for node controllability, nodes are only able to send a single linearly independent signal to all their neighbors, which in most socioeconomic contexts does not make sense. In the context of corporate control, a widely used example in both the node and edge controllability literature[7,8], this would unrealistically imply that a company can only give a single instruction to all owned subsidiaries.

A sub-field of the network controllability literature, called synchronizability or pinning control[17], also focuses on controlling the states on the network. In contrast to (structural) controllability, pinning control looks to achieve spatio-temporal symmetries in the states and thus only looks at a small section of the phase space at any given time. This approach is often used in consensus and epidemiology(-like) dynamics where the goal is for everyone to end up in the same state (e.g. reaching consensus, eradicating diseases, adopting a policy). Note that this goal is fundamentally different from (structural) controllability's goal of enabling access to any configuration in the state space.

Lastly, there exists a growing literature around estimating the treatment effect of an intervention in a network. This literature sets itself apart from the classic treatment effect estimation literature[18], which looks at the treatment effect at the individual level, by taking into account the connectedness of the entities. Herein, they try to estimate the true treatment effect of an observed intervention in a network by accounting for the spillover effects[19] that propagate from the individuals through the network, and the potential changes to the network itself[20] caused by this intervention. Alternatively, such estimations have also been achieved through simulating potential interventions and trying to approximate the whole propagation dynamics of these interventions[2,21].

The current literature around network interventions thus focuses on either figuring out if and by whom the states on the network could be maximally influenced (or controlled)—typically imposing strong assumptions or using heuristics—or estimating the effect of an observed intervention in a network. In practice, however, policymakers are often faced with the added difficulty of highly *non-linear dynamics*, as well as real-world *constraints* when designing intervention strategies (e.g. legal restrictions on which entities can be targeted, budget restrictions). To the best of our knowledge, no research currently exists that addresses these different aspects simultaneously in a single framework. This paper fills this gap by integrating insights from literature and deep-learning optimization tools into a scalable and flexible optimization framework for network interventions.

Reformulating the control question.

Taking inspiration from the external signal perspective in controllability (where an external node is connected to the driver nodes to steer the system), and the flow of control in social influence analysis (where models are utilised that describe the diffusion of control), we reformulate the question of optimal (static) control as the continuous optimisation of an objective function of a constrained external agent. This allows us to construct a framework that, given a differentiable control objective, leverages scalable automatic differentiation and gradient-based optimization strategies to compute the optimal (or most efficient) intervention an external agent can perform on the network to maximize its control. The objective function to be optimized consists of two parts: a first term capturing the agent's control in a given network configuration, and a second term capturing the cost (e.g. effort, monetary value) to obtain this control. Optimizing this objective function then essentially boils down to maximizing the control at minimal cost. The external agent perspective does not limit the scope of our approach, as one can let the external agent coincide with an internal node and initialize it with its edges as a starting point of the optimisation.

Traditionally, techniques for optimizing an objective function can be classified into three categories (1) symbolic differentiation, (2) numerical differentiation through the finite difference approximation of gradients, and (3) automatic differentiation. However, in network science applications, symbolic differentiation becomes intractable, and numerical differentiation becomes unstable, due to the many variables and interacting components. In deep learning, the literature was faced with a similar challenge in the analysis of large systems[22], which has been tackled by automatic differentiation. While automatic differentiation has received little attention in network science, we show its usefulness by demonstrating that, under the condition that the objective function is continuous and differentiable almost everywhere (the proof in[23] holds for our method as well), automatic differentiation and gradient-based optimization strategies can be used to find an optimum in the context of network control.

Given the many challenges to empirically measuring control in real-world socioeconomic networks[24] we will, as a first step, demonstrate our framework in the context of corporate ownership networks, where direct bilateral control can be defined through (partial) ownership of companies.

The challenge of global corporate ownership.

The decrease in communication and information costs has made businesses more interdependent in global value chains[25]. The associated globalisation of the corporate control market has resulted in more complex and internationalized corporate ownership. This has prompted several developed countries, including the EU and the US, to introduce or sharpen foreign investment screening mechanisms on the grounds of security or national interest[26]. At the same time governing such networks has become more cumbersome. Establishing who has controlling power in a company has become a nontrivial task, further complicated by different accounting rules and company structures across countries[27]. Individuals may control a holding company that acts as a store of ownership of other companies. Companies may control other companies indirectly through intermediary subsidiaries[3]. A network approach has therefore become prerequisite for obtaining a deeper understanding of corporate control problems. To account for indirect control along longer paths, ownership has to be propagated and consolidated throughout the different ownership paths between firms. Vitali et al.[28] proposed an adequate algorithm that summarizes the control of a given company on the network (how many subsidiaries a company controls, how much monetary value they represent, etc.). Using our framework we leverage this bilateral control matrix between companies and the cost of control, here the cost of buying shares, to find the optimal intervention by an external agent (company, government, etc.) to maximize its control over (part) of the network. Our ownership data is drawn from the Orbis dataset by Bureau van Dijk-A Moody's Company. This data has been used widely to study various aspects of business networks, such as their role in the international transmission of shocks[29], the relationship between group structure and subsidiary autonomy[30], the effect of contracting institutions on subsidiary location and ownership share[31], ownership concentration and types of control[32], and hierarchical differentiation across business groups[33]. While these papers study various aspects of business group networks, they all restrict themselves to static analyses of network characteristics or their implications. By consequence, none of these works yield methods that can be directly applied to finding optimal intervention strategies in corporate networks. We are therefore the first to provide an scalable optimization framework that enables the study of a new class of problems revolving around optimal interventions to control networks, such as vulnerability of strategic companies or sectors to hostile takeovers.

The remainder of the paper is structured as follows. First, we present our framework to optimize interventions in a network to gain optimal control. We detail the construction and optimisation of the objective function and show how we can account for real-world constraints on interventions such as budgets and target groups. Second, we illustrate the power of our framework by applying it in the context of the corporate ownership network. We start by explaining the intervention problem in this context and applying our framework to a simple synthetic graph to gain insights into the algorithm. We end this section by using data from Orbis Europe on Great Britain's biotech research sector to show how the framework can be used to gain crucial insights into the vulnerability of this sector to hostile takeovers in different policy environments. In the final part of this paper we discuss new research opportunities that can be explored with our framework, potential future extensions, and broader applicability.

Optimal control framework

In most use cases, the focus is explicitly on obtaining control over the nodes in a network (rather than e.g. over the edges). We therefore introduce our methodology in this context. A more general introduction to our core method, and an extension for constrained optimization, and pseudocode for the implementation of the different routines is given in the "Methods" section.

Node control optimization.

We are interested in optimizing the control over nodes (individuals, companies, etc.) in the network. For this reason, we focus on how and to what extent a single external agent (or node), representing, for instance, a hostile company, can take control of a network. Consider a network G of N nodes $n_i \in \mathscr{V}$ with corresponding node values v_i, and edges $e_{ij} \in \mathscr{E}$. The weighted edges $\mathscr{E}$ between the nodes in $\mathscr{V}$ are represented by the weighted adjacency matrix $A = \left[a_{ij}\right]$, with a_{ij} the weight of the edge e_{ij} from node n_i to node n_j. To represent the agent (whose intervention on the network we aim to optimize), we add an additional node x to the network, which is attached to G through a set of edges $e_{xi} \in \mathscr{O}$ that point to a selection of nodes $n_i \in \mathscr{S} \subseteq \mathscr{V}$. G' denotes the extended graph with nodes $\mathscr{V} \cup \{x\}$ and edges $\mathscr{E} \cup \mathscr{O}$. A schematic example is shown in Fig. 1. To differentiate between edges within G and edges originating from x, we use edge weight o_j to represent the direct control of x on node $n_j \in \mathscr{S}$, with $\mathbf{o} = [o_1, \ldots, o_{|\mathscr{S}|}]^T \in [0,1]^{|\mathscr{S}|}$. We assume the existence of a backbone algorithm (BB) that can propagate the (indirect) control through the network. Propagation of control refers to the fact that the control of node n_i over node n_j also indirectly introduces control of n_i over the children of n_j, and more generally, all of its descendants. The latter, however, requires a careful definition of how control is computed, e.g. in the case of cycles[28], and depends strongly on the chosen BB. The result of the BB must be a vector $\mathbf{c} = [c_1, \ldots, c_N]^T \in [0,1]^N$ that determines the total control (direct + indirect) of x into all (reachable) nodes (see Fig. 1). A node n_j is reachable from another node n_i if there is a directed path from n_i to n_j. For any node n_j that is unreachable from the external node x, $c_j = 0$. The BB algorithm must be continuous and differentiable in the edge weights o_j. Any method that meets the mentioned requirements (continuous, differentiable and producing a control vector) can be used as a backbone in our method.

Our goal is to optimize the (indirect) control over a targeted set of nodes $\mathscr{T} \subseteq \mathscr{V}$, by optimizing the direct control weights o_j over a subset of nodes $\mathscr{S} \subseteq \mathscr{V}$. Note that when $\mathscr{S} \subset \mathscr{V}$ or $\mathscr{T} \subset \mathscr{V}$ are strict subsets, we use the terms *source* and *target* restricted control. Gaining control in node n_j also comes with a cost, which is determined by the associated node's value v_j. The trade-off between cost and control can be tuned with the parameter λ, which gives preference to solutions with a lower cost. Hence, our optimization boils down to finding $\mathbf{o}^*$ for which

$$\mathbf{o}^* = \underset{\mathbf{o}}{\operatorname{argmin}}\, \mathscr{L}_{control}(\mathbf{o}) + \lambda \mathscr{L}_{cost}(\mathbf{o}). \tag{1}$$

As control-loss function, we define (assuming $c_j \in [0,1]$)

$$\mathscr{L}_{control}(\mathbf{o}) = \sum_{n_j \in \mathscr{T}} 1 - c_j(\mathbf{o}) \tag{2}$$

which describes the *lack* of control. Since we minimize the loss, this maximizes the control of x over $\mathscr{T}$. For most use cases the complexity and non-linearity are dictated by the functional dependence of $c_j(\mathbf{o})$. For example, as we will show in the next section for the case of corporate control, the intervention of an external agent (given by $\mathbf{o}$) will change the adjacency matrix A, i.e. $A(\mathbf{o})$, which in turn affects the control: $c_j(A(\mathbf{o}), \mathbf{o})$. In general, our framework does not impose linearity constraints on this functional dependence, and can handle nonlinear intervention effects on the network properties. The details of the loss function depend strongly on the use case. One may, for example, simply impose a cost that penalizes the number of takeovers

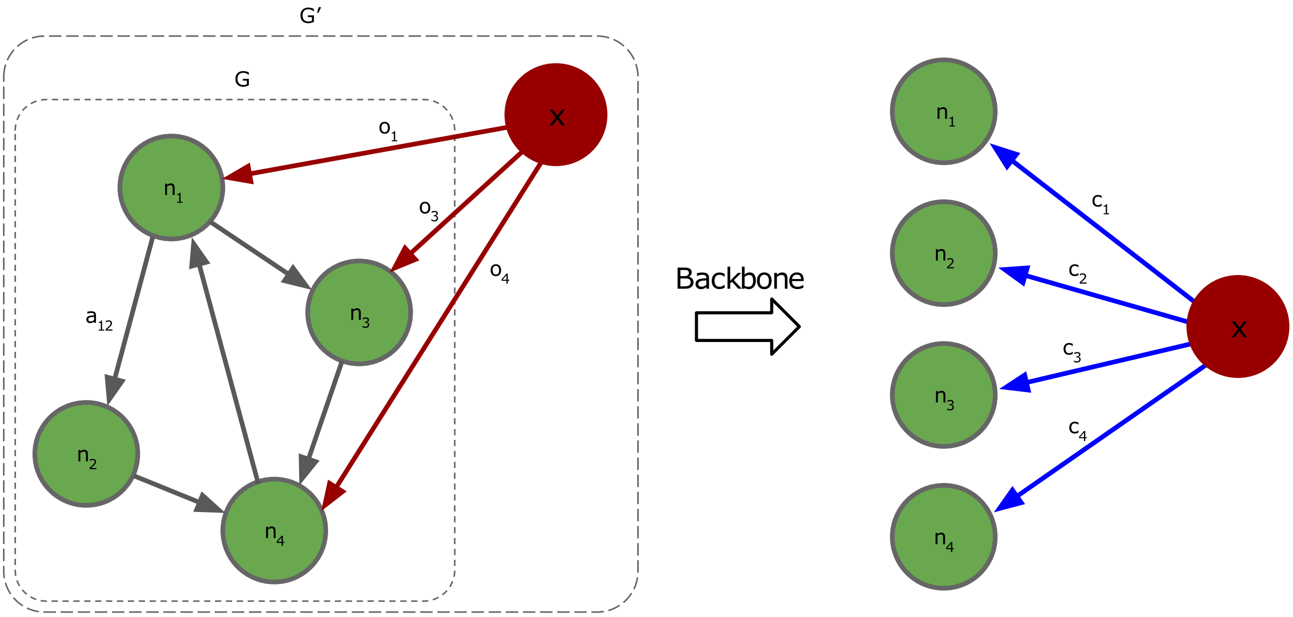

Figure 1. High-level illustration of the proposed framework. We show an example of a network G with nodes $n_i \in \mathscr{V}$ (green) with values v_i, and edges e_{ij} (grey) with their weights determined by the weighted adjacency matrix $A = [a_{ij}]$. We also show the extended network G' with the external node x (red), which is connected to the nodes of G by the edges e_{xj} (red) with weights o_j. Here, the directly controlled nodes are $\mathscr{S} = \{n_1, n_3, n_4\}$. After applying the backbone algorithm to propagate control, we obtain the total indirect control c_j (blue edges) of x in all reachable nodes. Hence, even though $n_2 \notin \mathscr{S}$ is not directly controlled by x, it can be indirectly controlled, e.g. through n_1.

$$\mathscr{L}_{cost}(\mathbf{o}) = \|\mathbf{o}\|_p \tag{3}$$

where $p = 1, 2$ and $\| \, \|_p$ indicates the l_p-norm, thereby inducing sparseness in $\mathbf{o}$. To relate the problem to SIM and (exact) controllability: this optimization aims at maximizing the control, with as few takeovers as possible, while ignoring the node values. By defining a cutoff C_{cut} for c_i or using proximal gradient descent, we can obtain a proxy for the size of the typical control set. Another useful loss function may be

$$\mathscr{L}_{cost}(\mathbf{o}) = \sum_{n_j \in \mathscr{S}} o_j v_j \tag{4}$$

which represents the cost of the direct control in the set $\mathscr{S}$, weighted by the node values.

We have explicitly introduced an external agent for clarity. In practice, this node might coincide with an internal node, be more connected, or contain an alliance of nodes. Our framework, however, is generic, and can be easily extended to these specific cases. While λ effectively constrains the budget, it might not be evident from its value what the budget exactly is. To accommodate situations where the optimisation needs to be run for a specific monetary value for the budget, we also implemented a constrained optimisation version of our framework (see "Methods" section).

Results

To show the power of our optimal control framework, we apply it to the corporate control problem and demonstrate its ability to provide actionable insights which were hitherto intractable to calculate. First, we explain the data and the relevant processing and modelling choices, definitions, and assumptions. Second, we apply the method together with the assumptions within a simple synthetic graph to demonstrate the inner workings and interpretation of the output of the algorithm. Lastly, we apply the method to a real-world example, Great Britain's biotech research sector, where we show how the application of our framework can lead to policy relevant insights.

The global corporate ownership network. In the context of global corporate control, information on the corporate ownership and corporate financials is typically available from government mandated financial reports that must be filed periodically. Companies like Bureau van Dijk collect these data and compile them into databases, such as the Orbis Europe database used in this paper. Using such data, Vitali et al.[28] have designed an algorithm to calculate the global control of companies by propagating and consolidating direct and indirect ownership throughout the ownership network. Their algorithm essentially assumes a multiplicative process of control along the paths in the networks and allows one to compute the control distribution of a snapshot of a network. It does not, however, provide an answer to governance questions relevant to investment screening mechanisms like"How vulnerable is a given sector to takeovers?", or "Which policy limits the sector's vulnerabil-

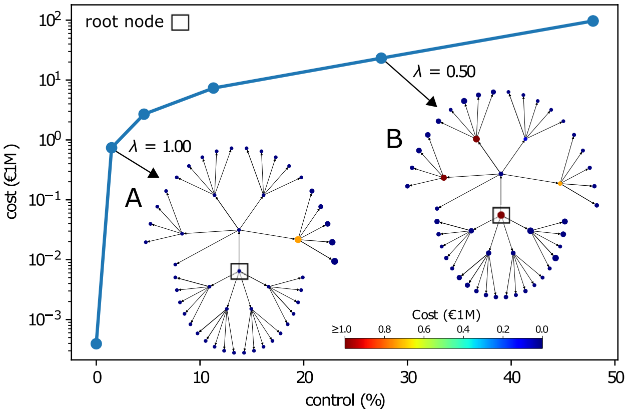

Figure 2. Control and cost loss as a function of λ-representing the importance of the budget in the control optimization—for the synthetic extended star graph. Control shows the % of total shares of the target nodes that the external agent controls. Cost is the monetary value that the agent spends to acquire a given control (by buying source nodes). The insets **A**, and **B** show the intervention strategy of the external agent for $\lambda = 1$ and $\lambda = 0.5$ (in units of control per €1M). The color of the node represents the cost the external agent invests in that specific company, while the size represents the control the external agent acquires.

ity to takeovers most effectively?". Answering such questions requires inferring the optimal perturbation of the edge weights in the network, which can be achieved with our methodological framework.

In the context of corporate control, the edge weights a_{ij} represent the direct control of company n_i over company n_j, while the node value v_i represents the total value of company n_i. These edge weights give an indication of how easy it is for company n_i to influence the decisions made in company n_j. While multiple voting systems exist within the corporate world (e.g. one share one vote, majority votes with different cut-offs), we maximize the total control. We thus view a higher percentage of control to equal more influence over decisions in the company, with 100% control meaning no restrictions on what decisions it can influence. Other approaches can be easily implemented by transforming the ownership values in the network. In the applications that follow, we use the algorithm in Vitali et al.[28] (see "Methods" section for technical details) as the BB in our framework to compute control in corporate ownership networks.

Given a trial external control/ownership vector $\mathbf{o}$, the network properties are affected by this external ownership. We therefore adjust the ownership matrix A using the following Assumptions:

1. x obtains a fraction o_j of ownership in a company n_j by uniformly buying stock from all companies n_i that have ownership in n_j: $a_{ij} \rightarrow (1 - o_j)a_{ij}$ (maximum entropy method). We define $A(\mathbf{o})$ as the adjusted ownership matrix for a given $\mathbf{o}$. To obtain ownership in root nodes of G, A remains unaffected.
2. we assume that x can only obtain as much ownership in a node n_j as explained by G: $o_j^{max} = \sum_{n_i \in \mathcal{V}} a_{ij}$ (except for root nodes of G, where we artificially set $o_j^{max} = 1$ in our examples).

Assumption 1 introduces non-linearities in our external agent optimization problem, since we introduce $\mathbf{o}$ -dependence into the non-linear BB algorithm by adjusting the adjacency matrix (B in (5)). In other words, non-linearities are introduced due to the fact that the agent's interventions affect the properties of the network which it is trying to control.

Application to a synthetic graph.

We first walk through the simple synthetic example of an extended star graph. This kind of topology appears frequently in corporate networks due to mergers and acquisitions between star networks consisting of companies and their subsidiaries. An example of this is Microsoft Corp, which owns Xbox, Skype, Linkedin, which in turn own other subsidiaries. For details see Fig. 1 in Rungi et al.[3]. The ownership weights are sampled from a uniform distribution $\mathcal{U}(0, 1)$. The node values follow the simple rule that the central nodes (root nodes) have the highest value, while the value of other nodes declines with their distance from these root nodes. We implement this feature using $v_i = 2^{D-d_i+1}$ (in €1M), where d_i and D are the depth of node n_i and the star (tree) respectively.

To study the vulnerability of the network to external control, we optimize Eq. (1) for several values of λ, the results are shown in Fig. 2. The λ-curve in Fig. 2 demonstrates exactly how expensive it is to obtain control over the network. Its changing slope, which reflects the changing trade-off between cost vs control, gives us an indication of vulnerability or susceptibility of the network to external control (e.g. hostile takeovers). We visualise the strategy of the external agent along the curve in insets **A** and **B** for $\lambda = 1$ and 0.5 respectively. The external agent prefers to take over companies closer to the root node up to the point where the budget is large enough to take over the root node itself, resulting in maximum control over the network. Note that not necessarily all shares of every company are available in the network (e.g. restricted shares). For this reason, the upper bound for external control might be less than 100%.

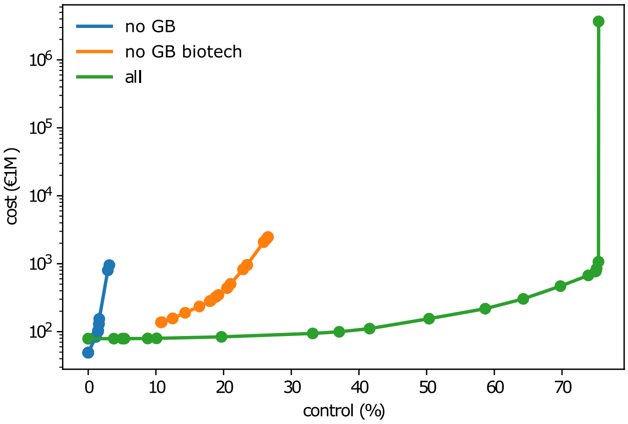

Figure 3. Control and cost for the target restricted control optimization. GB constraint (*no GB*, in orange), for the GB biotech constraint (*no GB biotech*, in blue), and for the unconstrained case (*all*, in green). Control shows the % of total shares of the target nodes that the external agent controls. Note that not necessarily all shares of every company are available in the network (e.g. restricted shares). Cost is the monetary value that the agent spends to acquire a given control (in €1M).

In this network the preference for proximity to the root node reflects that nodes closer to the root node offer a more significant amount of indirect control (see Fig. S1). We can conclude that for the extended star graph it quickly becomes very expensive to gain control over the network. However, once the budget allows to gain control over the root node, maximum control is quickly reached. More detailed investigation on the exact distribution of costs over the nodes is possible with our methodological framework but falls outside the scope of this article. Still, we provide an example in Fig. S2 and Fig. S3 in the SI.

This simple example shows how the framework can be used to perform a detailed analysis of both network features such as susceptibility to control, and node-specific features such as the centrality of the nodes that offer maximal control.

Application to the Great Britain (GB) biotech research sector.

To demonstrate the method in a real-world setting, we study GB's biotech sector. We extract the corporate ownership network from the Orbis Europe database provided by Bureau van Dijk, a private publisher of business information. We retrieve all the biotechnology research centers registered in GB and their in-component, which are all the nodes that can reach them through a directed path, from the 2017 ownership snapshot of the European corporate network (see "Methods" section for details). For simplicity we focus on the largest subgraph in which each pair of nodes is connected with each other via at least one path. The resulting network consists of 1109 nodes and 1506 edges, of which 114 biotechnology research companies from GB. We take the biotech research centers as the target nodes ($\mathscr{T}$) over which we want to maximize control with a given budget. We simulate different restrictions on the source nodes ($\mathscr{S}$), reflecting different GB policies with respect to foreign takeovers, to evaluate these policies' impact on the vulnerability of the biotechnology sector. We study three policy options:

1. GB constraint: the external agent cannot buy a company registered in GB directly but can buy any other European company in the network. In this case the nodes in $\mathscr{S}$ contain *no GB* companies.
2. GB biotech constraint: the external agent cannot buy a biotech research center registered in GB directly but can buy any other European or GB company in the network. In this case the nodes in $\mathscr{S}$ contain *no GB biotech* research centres.
3. Unconstrained: the external agent can buy any company in the network. In this case the nodes in $\mathscr{S}$ contain *all* companies.

We first show the λ-curves for these policy environments in Fig. 3. The unconstrained case can be viewed as an upper limit. Here the external agent is allowed to take over all available shares of the target companies directly, provided sufficient budget is available. From this unconstrained case, we see that just below 80% represents the upper bound for control an external company can achieve in the GB biotech research centers. Because buying direct control in the biotech research centers is not allowed in the GB and GB biotech constraint cases, control over the target companies can only be obtained indirectly through other companies. The optimal strategy to gain control over the target set in these cases is thus to leverage the ownership network and consolidate indirect ownership into control, a strategy that is often not accounted for in policy because of the complexity of ownership networks. Even for the relatively small network surrounding the biotech research centres, indirect control is non-trivial to account for (see Fig. S2 for a snapshot of the network). Being able to do so highlights one of the main benefits of our optimal control framework.

For the GB biotech constraint, i.e. GB specifically protects their biotech research centers from direct foreign investments. We, however, find that a foreign investor can still obtain up to 30% control over the sector. Most of

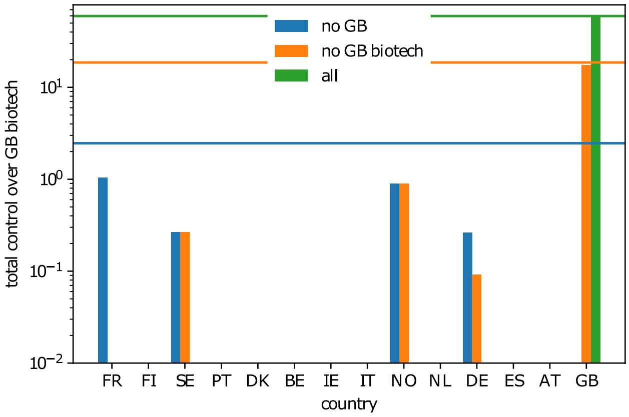

Figure 4. Distribution of the source of control over the GB biotech research sector along the various countries for a budget-restricted optimization. The budget is chosen to be $B = €901M$, which equals the total value of the target set of GB-biotech firms. The horizontal line indicate the total control achieved in the different policy environments.

this indirect control can be realized through non-biotech companies in GB (see Fig. 4), who thus seem to be more closely connected to the biotech research centres than (biotech) companies in the EU. If the constraint is extended to restrict investments in all GB companies (GB constraint), we find that only 4% of the sector is exposed to foreign control through EU companies. Note that these exposures represent lower bounds since we currently do not allow the external investor to change the outgoing edge weights (ownership) of the companies it invests in.

To fully protect the target set from unwanted external control, it is thus clear that policy should take into account the complete network of ownership and especially the large indirect exposure of the network.

Discussion

Controlling the many socioeconomic networks that greatly influence our daily lives has been a longstanding challenge in network science and across many fields. To go from observing and describing towards governing, there is a need to calculate optimal interventions while accounting for real-world constraints. Inspired by building blocks from network controllability and social influence analysis, we reformulate the problem of optimal interventions in networks into an optimization of an external agent's objective function. Herein, a first term captures the control objective, and the second term the cost of the control. This objective function is subsequently optimized using automatic differentiation and gradient-based optimization strategies. To the best of our knowledge, our framework represents the first occasion where automatic differentiation and gradient-based optimisation are integrated into a coherent framework for interventions in the context of (social) networks. Probably closest to our work are Fan et al.[34] who use deep reinforcement learning to find key nodes in tasks such as network destruction. While they search for optimal strategies to alter network connectivity (network property), our work allows to optimize a property on the network (node properties). We also point out that in machine learning, a similar reformulation of learning causal interactions as a continuous optimization problem gave rise to an entirely new field of causal deep learning and causal representation learning[35].

By building on a well-established backbone algorithm to calculate corporate control[28], we show that our framework is capable of characterizing the vulnerability of corporate networks in strategic sectors to sensitive takeovers, an important policy challenge for many countries (e.g. the EU framework for screening foreign direct investment started October 2020). Insights derived from our method could facilitate the protection of strategic sectors from unwanted foreign control.

Though we restricted the demonstration of our framework to the specific context of corporate control, it is directly applicable with the same BB algorithm by Vitali et al.[28] to a wide class of problems that require maximizing the diffusion of control, information, or influence throughout a network. This is because Vitali et al.[28] is valid in any context with a multiplicative process along the edges of a network (e.g. probability-based propagation). For example, re-imagining the synthetic graph in Fig. 2 as a specific socioeconomic network, the insights can then be reinterpreted for use in a marketing or information strategy. Further, any other propagation algorithm that is continuous and differentiable can be leveraged in the proposed framework, making it flexible to different use cases where propagation might not be multiplicative.

Our research shows that combining state-of-the-art gradient-based optimisation techniques and differentiation tools (that are mainly used in deep learning) with network theory opens up new opportunities for complex systems research to better understand, design, and govern networks. The present work represents an important first step in exploring these opportunities. In future work, we plan to extend the framework to calculate optimal interventions in a multi-agent context, where the different agents can have competing objectives in a shared environment. This will further broaden the applicability of the framework to more complicated scenarios, like in the case of foreign entities and the government as two separate agents competing for control. We expect that in the future, our method can be used to increasingly larger networks, where the current development of tools (such as PyTorch) for sparse matrices is of primary importance.

Methods

Vitali backbone algorithm for corporate control. As an example backbone algorithm to compute the indirect control in corporate networks, we use the corporate control algorithm by Vitali et al.[28]. We briefly outline their methodology. We first discuss how to compute corporate control between any pair of nodes $n_j, n_i \in \mathscr{V}$ within a network G (see e.g. Fig. 1), and then show how to obtain the control of the external agent.

Corporate control is approximated through corporate ownership. We assume we have a corporate ownership network, where the adjacency matrix $A = [a_{ij}]$ quantifies the factional ownership of company or node n_i in company n_j. When company n_k owns a fraction a_{ki} of company n_i, it is common to assume in the corporate control literature that n_k indirectly owns $a_{ki}a_{ij}$ shares of firm n_j through the path $n_k \to n_i \to n_j$.

We now aim to compute the total control of a node n_i over any other node in the network. We introduce the matrix B, which is obtained by selecting the rows and columns from (the direct control matrix) A that correspond to nodes that are reachable from n_i, after removing incoming links in n_i. The corresponding direct control of n_i into any n_j is given by a row of B, corresponding to paths of length $l = 1$: $C_{l=1} = [B]_{ij}$. Next, we consider all directed paths of length 2 starting from n_i. Proceeding as in the previous example, the corresponding control of n_i in n_j is given by $C_{l=2} = [B^2]_{ij}$. Continuing this way, and taking into account all possible path lengths, we find that the propagated control c_{ij} of any node $n_i \in \mathscr{V}$ in any other node $n_j \in V$ via the network G is given by the matrix $C = [c_{ij}]$ as

$$C = \sum_{l=1}^{+\infty} C_l = B(\mathbb{I} - B)^{-1} \tag{5}$$

For a more in-depth discussion, we refer to Vitali et al.[28].

By attaching the external party x to node $n_i \in \mathscr{S}$ with an edge o_j, x obtains control in a company v_j through all paths connecting n_i to n_j. The control C_{ij} is weighted by the edge control o_j of x in n_i, which corresponds to $C_{l=1} = \mathbf{o}$ in the discussion above. Hence, the total control of x over nodes $\mathscr{V}$ is given by

$$\mathbf{c}^T = \mathbf{o}^T(\mathbb{I} - B)^{-1} \tag{6}$$

One main difficulty in strongly connected corporate control networks is the presence of cycles, which the method of Vitali et al.[28] solves correctly. Nodes that are not reachable from x are of course not controllable by x: $c_j = 0$. We point out that the form of (5) can be used in a more general context than corporate control, since it accumulates the total signal by tracing all paths of all lengths between two nodes in the network.

Core method. In general, we take a network G, with node set $\mathscr{V}$ and edge set $\mathscr{E}$, to which we associate a set of (continuous-valued) node and edge property vectors: $V = \{v_i | n_i \in \mathscr{V}\}$ and $A = \{a_{ij} | e_{ij} \in \mathscr{E}\}$. In the specific case of node control in corporate networks, $v_i \in \mathbb{R}$ is the company value and $a_{ij} \in \mathbb{R}$ is fraction of company n_j that is owned by company n_i. We point out, however, that both v_i and e_{ij} can be vectors. Furthermore, their values can be fixed for some i, j. In our case of node control with an external agent, only a subset of the e_{ij} is varied directly, namely those that are outgoing of the external agent. The other e_{ij} are adjusted deterministically based on the latter, as discussed in Assumption 1. Depending on the application, we define a cost function with two terms. A first term is a control loss function $\mathscr{L}_{\text{control}}(V, E; G)$ representing the amount of control a set of nodes has over the network, for a given V and E. A second term $\mathscr{L}_{\text{cost}}(V, E; G)$ penalizes domains of the phase space (V, E) that carry a high cost. Given that the total loss function, $\mathscr{L}_{\text{control}} + \lambda \mathscr{L}_{\text{cost}}$ (with λ a constant factor that balances the importance of control and cost) is continuous and differentiable (almost everywhere) in the node and/or edge properties that we aim to optimize for, we determine the optimum via automatic differentiation and gradient descent.

In the above λ quantifies importance of cost versus control and enables us to gain insights into the general cost to control the system (by varying λ). In many applications the optimisation will involve a monetary budget constraint instead of a relative importance measure. In this case, the total cost that corresponds to a given set of properties V, E is subject to the constraint of a (maximum) budget M. We then need to optimize

$$\begin{aligned} &\min_{\mathbf{o}} \mathscr{L}_{control}(V, E; G) \\ s.t.\quad &H_{budget}(V, E; G) = 0 \end{aligned} \tag{7}$$

where $H_{budget}(V, E; G) = \mathscr{L}_{cost}(V, E; G) - M$ (the above may also be an inequality constraint). To solve the latter problem, we use the augmented Lagrangian approach[35,36], which uses an unconstrained optimization problem (with a quadratic penalty for the constraint) to solve the constraint optimization task. To account for budget constraints, we can thus optimize the dual loss function with Langrange multiplier α:

$$\mathscr{L}_{augm}(V, E; G) = \mathscr{L}_{control}(V, E; G) + \frac{\rho}{2} H_{budget}(V, E; G)^2 + \alpha \left| H_{budget}(V, E; G) \right|. \tag{8}$$

Optimization procedure. In practice, we parametrize the optimization problem with a set of unbounded variables $\mathbf{p}_u \in \mathbb{R}^{|\mathscr{S}|}$ and use the sigmoid function (σ) to obtain an ownership fraction $\mathbf{p} = \sigma(\mathbf{p}_u) \in [0, 1]^{|\mathscr{S}|}$. Furthermore, Assumption 2 suggests that $o_j \in [0, o_j^{max}]$. Therefore, we parametrize $o_j = o_j^{max} p_j$ to remain within these bounds. Hence, o_j represents to total fraction of company n_j owned by x, while p_j represents the fraction of the *available* stock owned by x. By implementing the procedure in deep-learning toolboxes such as PyTorch, one can perform network optimization use-cases efficiently using autograd. More specifically, the gradients are computed through a reverse-mode gradient accumulation scheme[37]. We use learning rates $\eta \in \{1, 0.1, 0.001\}$ and the Adam optimizer with default parameters in the PyTorch implementation. Notice that PyTorch also offers the possibility for sparse matrices, which would allow to scale the approach to large networks. The parameters p_{uj} are initialized by generating random values from a normal distribution $N(\mu = -7, \sigma = 10^{-4})$, or by estimating the values of **o** by the fraction of a company value with respect to the available total budget. The optimization stops when the loss changes by less than 10^{-8} in 5 subsequent optimization steps, or after a maximum of 3k steps, whichever occurs first. As an example, a complete single optimization, for a given value of λ, takes about 10 min for the Orbis Europe dataset of 1048 nodes on a laptop with a 1.4 GHz Quad-Core Intel Core i5 processor. For the constrained optimization, we optimize the augmented Lagrangian in (8) with $\rho = 1$ and $\alpha = 0$. We iteratively increase the value of α by adding $\rho|H_{budget}|$ and re-optimize. We then increase ρ by scaling it with a factor of 10, whenever we obtain a new value for the constraint $|H_{budget}|$ larger than 0.25 times the last optimal value of $|H_{budget}|$. This process is repeated until $|H_{budget}|$ is below a given tolerance threshold. Part of the procedure has been outlined in more detail in the pseudocode of the algorithm, see Algorithms 1–4.

Orbis Europe biotech data. The Orbis Europe database is compiled by Bureau van Dijk-A Moody's Company, a private publisher of business information who collects ownership structure and financial information on almost 20 million companies in Europe. This database provides for each company information on all its shareholders, corporate or non-corporate, and identifies direct and indirect voting rights. Widely held small shareholdership in listed firms is not identified separately but indicated as a separate category. Reference shareholders are indicated separately. If a company (or other type of shareholder) outside of the EU owns part of a European company, this information is also included in the data, while detailed financial information is not available for such a non-EU owner. From this database we extract all European and British companies for which total assets are reported and take this as our base network. From this network, we select the companies registered in Great Britain with NACE-code (standard European industrial classification code) 7211, whose activities are defined as: *"Research and experimental development on biotechnology"*. This resulted in 260 companies. Taking the in-component of all these companies (the companies that via some ownership path can reach any of the 260 biotech companies) resulted in a network of 1381 nodes and 1695 edges. From this we only take nodes with a known NACE-code so that we only extract the corporate network and drop for instance the individual owners. In line with international definitions used e.g. by the OECD, Eurostat, or UNCTAD, we focus on the corporate shareholder network through which control can be obtained of the network. For simplicity and plotting purposes, we further only keep the largest connected component in this network. This final selection resulted in a network of 1109 edges and 1506 nodes, with 114 of them being biotech research centres in Great Britain.

Pseudocode

A simplified and high-level overview of the budget-constrained optimization procedure is outlined in Algorithm 4 for a budget M. The core of our method is outlined in Algorithm 1, where we show how to compute the loss functions and its gradient. Helper sub routines are defined in Algorithms 2 and 3. The same notation is used as in the main text, and we refer the reader to the definitions therein, rather than recapitulating their definition. The gradients are computed using an automatic differentiation implementation (such as Pytorch[37], where one calls `torch.grad`) and therefore must not be computed analytically or through finite differentiation. The function `BackBone` returns the control vector for a given intervention. AdamOptimizerUpdate computes a new parameter vector based on the previous trial parameters with the Adam optimizer[38] with the default parameters in PyTorch[37].

Algorithm 1 Function ControlAndGradients($\mathbf{p}_u; \mathbf{o}_{max}, A, \rho, \alpha, M$) computes the value and gradient of the (augmented) loss function with respect to trial parameters $\mathbf{p}_u$.

```
1:  $\mathbf{p} \leftarrow \mathrm{sigmoid}(\mathbf{p}_u)$
2:  $\mathbf{o} \leftarrow \mathbf{o}^{max} \odot \mathbf{p}$
3:  $B \leftarrow A(\mathbf{1} - \mathbf{o})$                                        (Assumption 1)
4:  $\mathbf{c} \leftarrow \mathrm{BackBone}(\mathbf{o}, B)$                  (for example, Vitali et al.[28], see Eq. 6)
5:  $\mathscr{L}_{control} \leftarrow \sum_{n_j \in \mathscr{T}} 1 - c_j$
6:  $\mathscr{L}_{cost} \leftarrow \sum_{n_j \in \mathscr{S}} o_j v_j$
7:  $H_{budget} \leftarrow \mathscr{L}_{cost} - M$
8:  $\mathscr{L}_{augm} \leftarrow \mathscr{L}_{control} + \frac{\rho}{2} H^2_{budget} + \alpha \left|H_{budget}\right|$        (see Eq. 8)
9:  $\mathbf{g}_{\mathbf{p}_u} \leftarrow \nabla_{\mathbf{p}_u} \mathscr{L}_{augm}(\mathbf{p}_u)$    (Gradients: use automatic differentiation to propagate back to line 1.)
10: return $\mathscr{L}_{augm}, H_{budget}, \mathbf{g}_{\mathbf{p}_u}$
```

Algorithm 2 Function OptimizeControl($\mathbf{p}_u; A, \mathbf{o}_{max}, \rho, \alpha, M$) optimizes the (augmented) loss with learning rate $\eta = 10^{-1}$ and tolerance $\text{tol}_{\mathscr{L}} = 10^{-8}$, by varying the trial parameters $\mathbf{p}_u$.

```
$\mathscr{L}_{augm,old} \leftarrow +\infty$
$\mathscr{L}_{augm} \leftarrow 0$
while $\left|\mathscr{L}_{augm} - \mathscr{L}_{augm,old}\right| > \text{tol}_{\mathscr{L}}$ do        (Can be relaxed to a condition over multiple subsequent steps.)
    $\mathscr{L}_{augm,old} \leftarrow \mathscr{L}_{augm}$
    $\mathscr{L}_{augm}, H_{budget}, \mathbf{g}_{\mathbf{p}_u} \leftarrow \text{ControlAndGradients}(\mathbf{p}_u; A, \mathbf{o}_{max}, \rho, \alpha, M)$
    $\mathbf{p}_u \leftarrow \text{AdamOptimizerUpdate}(\mathbf{p}_u, \mathbf{g}_{\mathbf{p}_u}, \eta)$
end while
return $\mathbf{p}_u, \mathscr{L}_{augm}, H_{budget}$
```

Algorithm 3 Function ComputeMaximumOwnership(A).

```
for $j \in \mathscr{S}$ do
    if IsRootNode($n_j$, $G$) then
        $o_j^{max} \leftarrow 1$        (Assumption 2)
    else
        $o_j^{max} \leftarrow \sum_{n_i \in \mathscr{V}} a_{ij}$
    end if
end for
return $\mathbf{o}^{max}$
```

Algorithm 4 Optimization of network control with budget M and constraint tolerance $\text{tol}_H = 10^{-8}$

```
$\mathbf{p}_u \leftarrow \text{Normal}(\mu = -7, \sigma = 10^{-4})$        (Expert-based initial values result in better convergence.)
$\mathbf{o}^{max} \leftarrow$ ComputeMaximumOwnership(A)
$\rho \leftarrow 1$
$\alpha \leftarrow 0$
$H_{budget,old} \leftarrow +\infty$
while True do
    while $\rho < 10^{20}$ do
        $\mathbf{p}_u, \mathscr{L}_{augm}, H_{budget} \leftarrow \text{OptimizeControl}(\mathbf{p}_u; A, \mathbf{o}_{max}, \rho, \alpha, M)$
        if $\left|H_{budget}\right| > \frac{1}{4}\left|H_{budget,old}\right|$ then
            $\rho \leftarrow 10\rho$
        else
            break
        end if
    end while
    $H_{budget,old} \leftarrow H_{budget}$
    $\alpha \leftarrow \alpha + \left|\rho H_{budget}\right|$
    if $\left|H_{budget,old}\right| < \text{tol}_H$ then
        break
    end if
end while
return $\mathbf{p}_u$
```

Data availability

All data analysed in this paper, including synthetic graphs and real-world networks, are anonymised according to the conditions of the data provider (Bureau van Dijk) and included in the GitHub repository.

Code availability

The code of the Optimal Network Control framework and for the reproduction of the figures is detailed and available in the following GitHub repository: https://github.com/CSI-ADS/OptimControl.

Received: 25 August 2021; Accepted: 18 January 2022
Published online: 10 February 2022

References

1. Roukny, T., Bersini, H., Pirotte, H., Caldarelli, G. & Battiston, S. Default cascades in complex networks: Topology and systemic risk. *Sci. Rep.* **3**, 1–8. https://doi.org/10.1038/srep02759 (2013).
2. Banerjee, A., Chandrasekhar, A. G., Duflo, E. & Jackson, M. O. The diffusion of microfinance. *Science* **341**, 1236498. https://doi.org/10.1126/science.1236498 (2013).
3. Rungi, A., Morrison, G. & Pammolli, F. Global ownership and corporate control networks. *IMT Lucca EIC WP Series*, Vol. 7 (2017).
4. Banerjee, S., Jenamani, M. & Pratihar, D. K. A survey on influence maximization in a social network. *Knowl. Inf. Syst.* **62**, 3417–3455. https://doi.org/10.1007/s10115-020-01461-4 (2020).
5. Li, Y., Fan, J., Wang, Y. & Tan, K.-L. Influence maximization on social graphs: A survey. *IEEE Trans. Knowl. Data Eng.* **30**, 1852–1872. https://doi.org/10.1109/TKDE.2018.2807843 (2018).
6. Kempe, D., Kleinberg, J. & Tardos, É. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 137–146 (2003).
7. Liu, Y.-Y., Slotine, J.-J. & Barabási, A.-L. Controllability of complex networks. *Nature* **473**, 167–173. https://doi.org/10.1038/nature10011 (2011).
8. Nepusz, T. & Vicsek, T. Controlling edge dynamics in complex networks. *Nat. Phys.* **8**, 568–573. https://doi.org/10.1038/nphys2327 (2012).
9. Zhang, Y., Garas, A. & Schweitzer, F. Control contribution identifies top driver nodes in complex networks. *Adv. Complex Syst.* **22**, 1950014 (2019).
10. Gao, J., Liu, Y.-Y., D'Souza, R. M. & Barabási, A.-L. Target control of complex networks. *Nat. Commun.* **5**, 1–8. https://doi.org/10.1038/ncomms6415 (2014).
11. Pang, S.-P. & Hao, F. Target control of edge dynamics in complex networks. *Phys. A Stat. Mech. Appl.* **512**, 14–26. https://doi.org/10.1016/j.physa.2018.08.011 (2018).
12. Pang, S.-P., Wang, W.-X., Hao, F. & Lai, Y.-C. Universal framework for edge controllability of complex networks. *Sci. Rep.* **7**, 1–12. https://doi.org/10.1038/s41598-017-04463-5 (2017).
13. Galbiati, M., Delpini, D. & Battiston, S. The power to control. *Nat. Phys.* **9**, 126–128. https://doi.org/10.1038/nphys2581 (2013).
14. Egerstedt, M. Degrees of control. *Nature* **473**, 158–159 (2011).
15. van den Heuvel, M. & Nys, J. Communication coordination in network controllability. arXiv, https://arxiv.org/pdf/2105.04164.pdf. arXiv:2105.04164v1 (2021).
16. Kalman, R. E. Mathematical description of linear dynamical systems. *J. Soc. Ind. Appl. Math. Ser. A Control* **1**, 152–192. https://doi.org/10.1137/0301010 (1963).
17. Wang, X. F. & Chen, G. Pinning control of scale-free dynamical networks. *Phys. A Stat. Mech. Appl.* **310**, 521–531. https://doi.org/10.1016/s0378-4371(02)00772-0 (2002).
18. Abadie, A. & Cattaneo, M. D. Econometric methods for program evaluation. *Annu. Rev. Econ.* **10**, 465–503. https://doi.org/10.1146/annurev-economics-080217-053402 (2018).
19. Bargagli-Stoffi, F. J., Tortù, C. & Forastiere, L. Heterogeneous treatment and spillover effects under clustered network interference. arXiv:2008.00707 (2020).
20. Comola, M. & Prina, S. Treatment effect accounting for network changes. *Rev. Econ. Stat.* **103**, 597–604. https://doi.org/10.1162/rest_a_00908 (2021).
21. Badham, J., Kee, F. & Hunter, R. F. Simulating network intervention strategies: Implications for adoption of behaviour. *Netw. Sci.* **6**, 265–280. https://doi.org/10.1017/nws.2018.4 (2018).
22. Baydin, A. G., Pearlmutter, B. A., Radul, A. A. & Siskind, J. M. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.* **18**, 1–43 (2018).
23. Lee, W., Yu, H., Rival, X. & Yang, H. On correctness of automatic differentiation for non-differentiable functions. In *Advances in Neural Information Processing Systems*, Vol. 33 (2020).
24. Peng, S. *et al.* Influence analysis in social networks: A survey. *J. Netw. Comput. Appl.* **106**, 17–32. https://doi.org/10.1016/j.jnca.2018.01.005 (2018).
25. Baldwin, R. *The Great Convergence: Information Technology and the New Globalization* (Harvard University Press, 2016).
26. Karl, J., Zhan, J., Weber, J. & Kuc, O. *National Security-related Screening Mechanisms for Foreign Investment: An Analysis of Recent Policy Developments* (United Nations Conference on Trade and Development, 2019).
27. Garcia-Bernardo, J., Fichtner, J., Takes, F. W. & Heemskerk, E. M. Uncovering offshore financial centers: Conduits and sinks in the global corporate ownership network. *Sci. Rep.* **7**, 1–10. https://doi.org/10.1038/s41598-017-06322-9 (2017).
28. Vitali, S., Glattfelder, J. B. & Battiston, S. The network of global corporate control. *PLoS ONE* **6**, e25995 (2011).
29. Cravino, J. & Levchenko, A. A. Multinational firms and international business cycle transmission*. *Q. J. Econ.* **132**, 921–962. https://doi.org/10.1093/qje/qjw043 (2016).
30. Belenzon, S., Hashai, N. & Patacconi, A. The architecture of attention: Group structure and subsidiary autonomy. *Strateg. Manag. J.* **40**, 1610–1643. https://doi.org/10.1002/smj.3059 (2019).
31. Eppinger, P. & Kukharskyy, B. Contracting institutions and firm integration around the world. *Eur. Econ. Rev.* **137**, 103815. https://doi.org/10.1016/j.euroecorev.2021.103815 (2021).
32. Aminadav, G. & Papaioannou, E. Corporate control around the world. *J. Finance* **75**, 1191–1246. https://doi.org/10.1111/jofi.12889 (2020).
33. Altomonte, C., Ottaviano, G. I., Rungi, A. & Sonno, T. *Business Groups as Knowledge-based Hierarchies of Firms*. CEPR Discussion Paper 16677 (2021).
34. Fan, C., Zeng, L., Sun, Y. & Liu, Y.-Y. Finding key players in complex networks through deep reinforcement learning. *Nat. Mach. Intell.* **2**, 317–324. https://doi.org/10.1038/s42256-020-0177-2 (2020).
35. Zheng, X., Aragam, B., Ravikumar, P. K. & Xing, E. P. Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems* 9472–9483 (2018).
36. Nemirovsky, A. *Optimization II. Numerical Methods for Nonlinear Continuous Optimization* (Technion-Israel Institute of Technology, 1999).
37. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32* (eds Wallach, H. *et al.*) 8024–8035 (Curran Associates, Inc., 2019).
38. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

Acknowledgements

This research was supported by BOF Postdoctoral Fellowships (BOF20/PDO/030 and c7d2cd3a-6e3e-42a5- 81f4-84315fdce944), the National Bank of Belgium (DOZA/IL/DDC/AM/133b-2021), and Hercules

Foundation-FWO (AUGE/17/01). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Author contributions

J.N., M.v.d.H, K.S. designed research; J.N., M.v.d.H. performed research; J.N. implemented the model; J.N., M.v.d.H. analyzed and processed results; J.N., M.v.d.H wrote the original draft; J.N., M.v.d.H, K.S, and B.M reviewed and edited the final draft; and K.S., B.M. provided data.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-06144-4.

Correspondence and requests for materials should be addressed to M.v.d.