# Interface of Complexity and Data science

Dr. Nino Antulov-Fantulin[1]

[1]Computational Social Science, ETH Zurich, Switzerland

September 9, 2020

ETH zürich

# Interface

**Complexity science** studies complex systems and problems that are composed of many components which may interact with each other in a dynamic and non-linear way. In particular, I am interested in complex networks.

Interface of **complexity** and **data science** is what? Place/problems where mathematics of Machine Learning and Complexity Science are "coupled" in such a way to provide insights.

# About me

- Senior scientist at Computational Social Science, ETH Zurich
- Visiting research associate at Courant, NYU.
- Phd Supervisor and Panel member of the "Data Science" joint doctorate (in collaboration with Scuola Normale Superiore, Sant'Anna School, University of Pisa and National Research Council)
- Co-founder and Head of Research at Aisot, real-time predictive analytics for markets.
- Previously, I was at Institute Rudjer Boskovic and Faculty of Electrical Engineering and Computing in Zagreb.

# Research questions

Structure

- How to select the set of nodes in a network that, when removed or (de)activated, can stop the spread of (dis)information, mitigate an epidemic, or disrupt a malicious system by fragmenting it into small components at the minimum overall cost
- What is the connection between directed graphs and geometry of statistical manifolds?

Dynamics:

- How graph geodesics can unify continuous and discrete time stochastic susceptible-infected-recovered processes on networks?
- Can Neural Networks be used for controlling dynamical processes on complex networks?

# Outline

- Mapping spreading dynamics (Collab)

$$\frac{d}{dt}F(\mathcal{G}) \to \mathcal{G}^*(V, E)$$

- Statistical embedding for directed graphs (Collab)

$$V \to R^k$$

- Neural Network Control of Dynamics (Collab)

$$\frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t), \mathcal{G}, \mathbf{u}(t))$$

- Generalized network dismantling (Optional)

$$\mathcal{G}(V, E) \to \mathcal{G}^*(V^*, E^*)$$

# Dynamical process – Spreading process

# Epidemicdatathon.com

# Epidemic model


**Problem formulation**: For a static, unweighted graph $G = (E, V)$ estimate final outcome of the SIR process.

- $\beta dt$ – probability that an Infected node transmits the disease to Susceptible neighbouring node in $dt$,
- $\gamma dt$ – probability that an Infected node recovers in $dt$.

# Mapping of spreading dynamics

SCIENTIFIC REPORTS

Article | OPEN | Published: 26 April 2018

## Simulating SIR processes on networks using weighted shortest paths

Dijana Tolić, Kaj-Kolja Kleineberg & Nino Antulov-Fantulin

*Scientific Reports* **8**, Article number: 6562 (2018) | Download Citation

# Unifying susceptible-infected-recovered processes on networks

PHYSICAL REVIEW RESEARCH

Welcome Recent Subjects Authors Referees Search About Scope Staff

Open Access

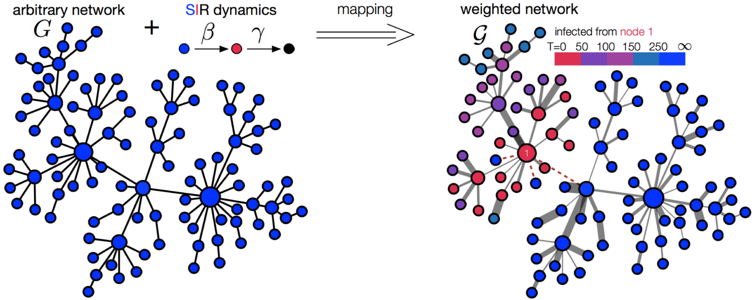
Unifying continuous, discrete, and hybrid susceptible-infected-recovered processes on networks

Lucas Böttcher and Nino Antulov-Fantulin
Phys. Rev. Research **2**, 033121 – Published 22 July 2020

https://arxiv.org/pdf/2002.11765.pdf

# Mapping of spreading dynamics

Mapping to the ensemble of weighted networks $\{\mathcal{G}_k\}$



# Mapping of spreading dynamics

# Mapping of spreading dynamics

In the special case of the Poissonian SIR process one obtains

$$\rho_{i,j} = \begin{cases} -\ln(x)/\beta : & -\ln(x)/\beta \leq -\ln(y)/\gamma \\ \infty : & -\ln(x)/\beta > -\ln(y)/\gamma \end{cases}. \qquad (1)$$

# Geodesic equivalence

# Geodesic equivalence

We denote the distance as shortest paths on weighted networks, $d_{\mathcal{G}_k}(v_i, v_j) = \min_{\chi_{ij}} \sum_{(k,l)\in\chi_{ij}} \rho_{k,l}$.

This distance is equivalent to the propagation time from node $i$ to $j$, i.e. $t(v_i \rightarrow v_j) = d_{\mathcal{G}_k}(v_i, v_j)$.

Approximation or exact?
Complexity .... $O(E + N \log N)$ find solution from any node.

# Source Inference

For a given snapshot $r^*$ of a spreading process at time $t_0$ on a network, how does one detect the source node that generated the snapshot?

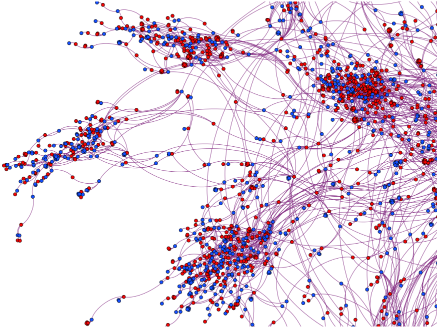
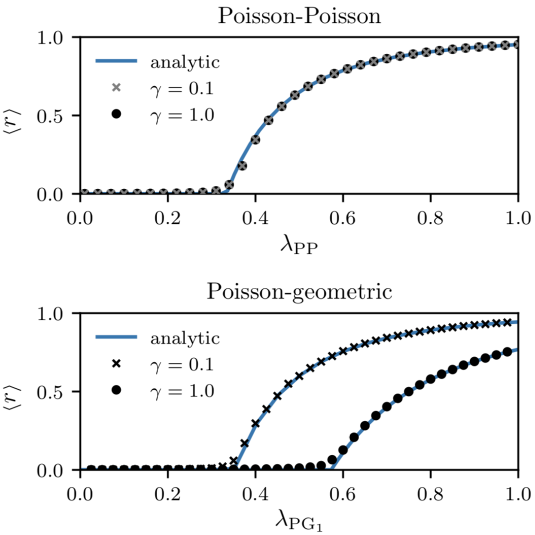


Figure: Visualization of aggregated empirical temporal network ($\approx 3500$ nodes) of sexual contacts in Brazil, used for source detection (Antulov-Fantulin et al. , Phys. Rev. Lett. 114, 248701)

# Unifying SIR processes



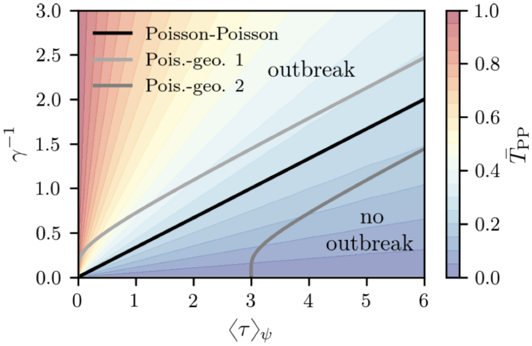
# Mapping of spreading dynamics

$$\rho_{i,j} = \begin{cases} \Psi^{-1}(x) : & \Psi^{-1}(x) \leq \Phi^{-1}(y) \\ \infty : & \Psi^{-1}(x) > \Phi^{-1}(y) \end{cases}, \tag{2}$$

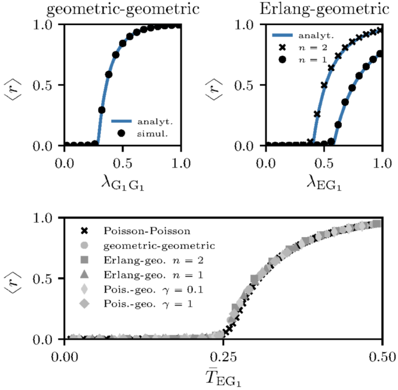
where $x$ and $y$ are uniform random numbers $\in [0,1]$, $\Phi^{-1}(x)$ and $\Psi^{-1}(y)$ are inverse functions of the cumulative inter-event distributions
$\Phi(t) = \int_0^t dt' \phi(t')$ (recovery time) and
$\Psi(t) = \int_0^t dt' \psi(t')$ (infection propagation delay).

Exact and mean-field mapping!

# Unifying SIR processes



# Unifying SIR processes



# Mapping SIR – Collab example

https://colab.research.google.com/drive/
1CmSegGOFODjdbMJOcyH69m2tVI9OLWo4?usp=sharing

# Statistical manifold embedding

Circle Limit III by M. C. Escher. Wood engraving, 1959.

# Statistical embedding



Published as a conference paper at ICLR 2020

## Low-dimensional statistical manifold embedding of directed graphs

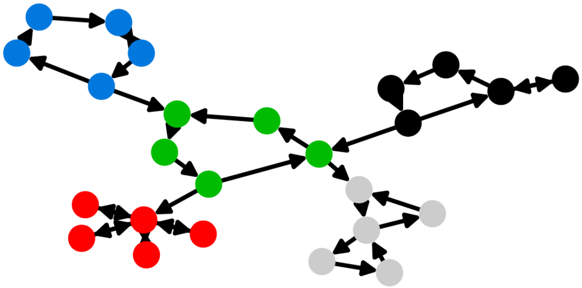
**Thorben Funke**
L3S Research Center
Leibniz University Hannover
Hannover, Germany

**Tian Guo**
Computationl Social Science
ETH Zürich
Zurich, Switzerland

**Alen Lancic**
Faculty of Science
Department of Mathematics
University of Zagreb, Croatia

**Nino Antulov-Fantulin**
Computationl Social Science
ETH Zürich
Zurich, Switzerland
anino@ethz.ch

# Statistical embedding



# Statistical embedding

Table 1: Properties of datasets

| Name | $\|V\|$ | $\|E\|$ | $\|\{d_{u,v} : d_{u,v} = \infty\}\|/\|V\|^2$ | Reciprocity |
|---|---|---|---|---|
| Synthetic example | 25 | 30 | 0.48 | 34.3% |
| Political blogs | 1224 | 19,025 | 0.34 | 24.3% |
| Cora | 23,166 | 91,500 | 0.83 | 5.1% |
| arXiv hep-th | 27,770 | 352,807 | 0.71 | 0.3% |

# Statistical embedding

We propose a low-dimensional node embedding of directed graphs to probability distributions, by using divergence as a measurement to quantify distorsion w.r.t. graph geodesics.

# Statistical embedding



Our embedding space $X$ is the space of k-variate exponential power distributions (also called generalized error distributions), described by the following probability density function for $\lambda > 0, \psi_\lambda(x|\boldsymbol{\Sigma}, \boldsymbol{\mu}) =$

$$\frac{\lambda \Gamma(\frac{k}{2})}{2^{1+\frac{k}{\lambda}} \pi^{\frac{k}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}} \Gamma(\frac{k}{\lambda})} \exp\left(-\frac{[(x-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (x-\boldsymbol{\mu})]^{\frac{\lambda}{2}}}{2}\right)$$

# Statistical embedding

Now, in order to learn these statistical manifold embeddings, we can define the loss function over asymmetric distances $D = (\mathrm{d}_{u,v})_{u,v \in V}$ of a directed graph and $\{(\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u)\}_u$ as:

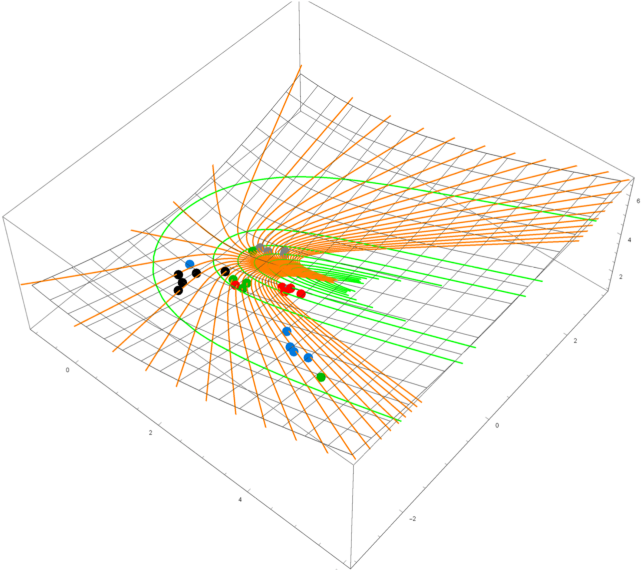
$$\mathcal{L}(\{(\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u)\}_u) = \sum_{u \neq v} ||(1 + \tau\, \mathrm{KL}_{u,v})^{-1} - \mathrm{d}_{u,v}^{-\beta}\,||_2^2,$$

where $\tau \in \mathbb{R}_+$ is a free (trainable) parameter and $\beta \in \mathbb{R}_+$ a fixed value.

# Statistical embedding – problem formulation

- we use
$$KL_{p,q} = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{dx},$$
where $p$ and $q$ are the densities of node $P$, $Q$ to minimize the distorsion w.r.t. distance $\mathrm{d}_G(P, Q)$

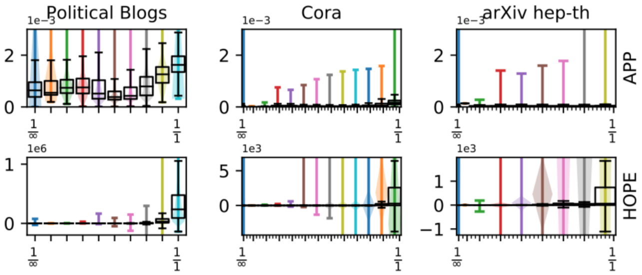
# Statistical embedding



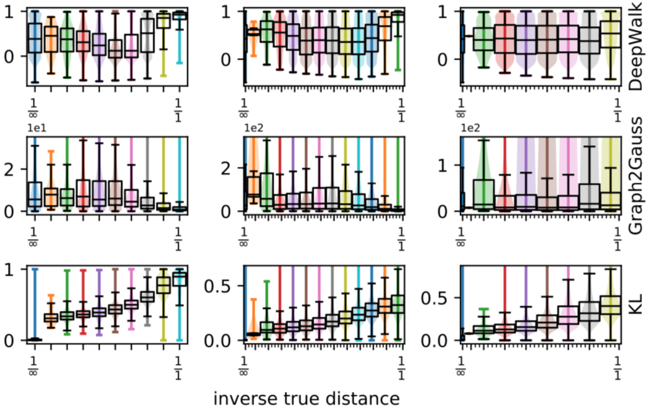
# Statistical embedding

| Network | | APP | HOPE | DeepWalk | Graph2Gauss | KL (10) | KL (100) | KL (full) |
|---|---|---|---|---|---|---|---|---|
| Political Blogs | $\rho$ | .16 | .45 | .25 | $-.17$ | .73 | .77 | **.88** |
| | $r$ | .29 | .45 | .24 | $-.33$ | .71 | .72 | **.89** |
| | avg. MI | .15 | .65 | 0.12 | .09 | .47 | .59 | **.85** |
| | std. MI | .006 | .007 | .005 | .005 | .005 | .005 | .006 |
| Cora | $\rho$ | .10 | .17 | .07 | $-.004$ | .53 | .66 | **.77** |
| | $r$ | .01 | .41 | .02 | .013 | .56 | .62 | **.65** |
| | avg. MI | .018 | .13 | .013 | .01 | .23 | .35 | **.43** |
| | std. MI | 002 | .005 | .004 | .003 | .005 | .006 | .007 |
| arXiv hep-th | $\rho$ | .01 | .20 | .08 | $-.05$ | .53 | .62 | **.68** |
| | $r$ | .12 | .28 | .04 | $-.06$ | .59 | .68 | **.72** |
| | avg. MI | .033 | .15 | .014 | .01 | .25 | **.37** | .36 |
| | std. MI | .003 | .004 | .003 | .003 | .005 | .005 | .005 |

# Statistical embedding



# Statistical embedding



# Statistical embedding – geometry (advanced)

**Theorem 1.** *Let $\lambda$ be an even number. Then the distributions with density of Eq. (1) and parametrization $(\sigma^1, \dots \sigma^k, \mu^1, \dots, \mu^k) \mapsto \boldsymbol{X} \sim \psi_\lambda(x|\boldsymbol{\Sigma} = \mathrm{diag}(\sigma^1, \dots, \sigma^k), \boldsymbol{\mu} = (\mu^1, \dots, \mu^k))$ with $\sigma^i > 0$ are a statistical manifolds $\mathcal{S}$ with the following properties:*

1. *For univariate exponential power distribution, the curvature is constant and equal to $-1/\lambda$.*
2. *the Fisher information matrix, i.e. the Riemannian metric tensor, in this coordinate system at point $\psi_\lambda(x|\boldsymbol{\Sigma}, \boldsymbol{\mu})$ with $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma^1, \dots, \sigma^k)$ and $\boldsymbol{\mu} = (\mu^1, \dots, \mu^k)$ is given by*
$$(g_{ij})_{ij} = \mathrm{diag}\left(\frac{c_1}{(\sigma^1)^2}, \dots, \frac{c_1}{(\sigma^k)^2}, \frac{c_2}{(\sigma^1)^2}, \dots, \frac{c_2}{(\sigma^k)^2}\right),$$
*where*
$$c_1 = \frac{\Gamma(1-\frac{1}{\lambda})\lambda(\lambda-1)}{\Gamma(\frac{1}{\lambda})}, \quad c_2 = \lambda.$$
3. *the Riemannian distance between $\psi_\lambda(x|\boldsymbol{\Sigma}, \boldsymbol{\mu})$ and $\psi_\lambda(x|\tilde{\boldsymbol{\Sigma}}, \tilde{\boldsymbol{\mu}})$ is*
$$\mathrm{d}_F(\psi_\lambda(x|\boldsymbol{\Sigma}, \boldsymbol{\mu}), \psi_\lambda(x|\tilde{\boldsymbol{\Sigma}}, \tilde{\boldsymbol{\mu}})) = \sqrt{\lambda \sum_{i=1}^{k} \mathit{arcosh}^2 \left(1 + \frac{\left(\frac{\mu^i - \widetilde{\mu}^i}{c_3}\right)^2 + (\sigma^i - \widetilde{\sigma}^i)^2}{2\sigma^i \widetilde{\sigma}^i}\right)}$$
*with $c_3 = \sqrt{\frac{\Gamma(\frac{1}{\lambda})}{(\lambda-1)\Gamma(1-\frac{1}{\lambda})}}$.*

# Statistical embedding – learning (advanced)



**Scalable learning procedure.** The full objective function Eq. (3) consists of $|V|(|V|-1)$ terms and only graphs with up to the magnitude of around $10^4$ nodes can be applied. To extend our method beyond this limit, we propose an approximated solution, where the size of the training data scales linearly in the number of nodes and thus can be applied to large graphs.

This approximation solution is based on a decomposition of the loss function Eq. (3) into a neighborhood term and a singularity term as:

$$\mathcal{L}(\{(\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u)\}_u) = \underbrace{\sum_{u\neq v, \mathrm{d}_{u,v}<\infty} ||(1+\tau\,\mathrm{KL}_{u,v})^{-1} - \mathrm{d}_{u,v}^{-\beta}\,||_2^2}_{\text{neighborhood term}} + \underbrace{\sum_{u\neq v, \mathrm{d}_{u,v}=\infty} ||(1+\tau\,\mathrm{KL}_{u,v})^{-1} - \mathrm{d}_{u,v}^{-\beta}\,||_2^2}_{\text{singularity term}}.$$

# Statistical embedding – learning (advanced)

**Lemma 1.** *Let $G = (V, E)$ be a graph. Then our embedding has $2k|V| + 1$ degrees of freedom and one hyperparameter ($\beta$). Generating the training samples for the full method, is equivalent to the all pair shortest path problem, which can be solved within $O(|V|^2 \log |V| + |V||E|)$ for sparse graphs and evaluating the loss function has time complexity $O(|V|^2)$. The scalable variant has, for $B \ll |V|$, a time complexity of $O((B+1)|V| + |E|)$ and the loss function has $O(B|V|)$ terms.*

# Statistical embedding – general KL (advanced)

As there exists no closed form of KL divergence for the generalized exponential power distributions, we propose an efficient importance sampling Monte Carlo estimation.

$$\mathrm{KL}_{u,v} = \mathrm{KL}(p_u^\lambda(x), q_v^\lambda(x)) = \int p_u^\lambda(x) \log \frac{p_u^\lambda(x)}{q_v^\lambda(x)} \, \mathrm{d}x =$$

$$= \mathbb{E}_{x \sim p_u^\lambda(x)} \left[ \log \frac{p_u^\lambda(x)}{q_v^\lambda(x)} \right] = \mathbb{E}_{x \sim \psi_{\lambda_*}(x)} \left[ \frac{p_\lambda(x)}{\psi_{\lambda_*}(x)} \log \frac{p_\lambda(x)}{q_\lambda(x)} \right].$$

# Statistical embedding – learning (advanced)



### A.8 Correction of Euclidean gradient

In section 3, we have deduced that the steepest direction is given by $\tilde{\nabla} L = G^{-1} \nabla L$. In other words, in each step of our optimization, we need to correct [2] the Euclidean gradient by the inverse of Fisher information matrix at the respective point. To be more specific, to update the representation of a node with $\sigma^1, \ldots \sigma^k, \mu^1, \ldots, \mu^k$ we apply

$$\tilde{\nabla}_{\frac{\partial}{\partial \sigma^i}} L = \frac{(\sigma^i)^2}{c_1} \nabla_{\frac{\partial}{\partial \sigma^i}} L, \quad \tilde{\nabla}_{\frac{\partial}{\partial \mu^i}} L = \frac{(\sigma^i)^2}{c_2} \nabla_{\frac{\partial}{\partial \mu^i}} L,$$

where $c_1$ and $c_2$ are the constants from Theorem 1. As we can see in Figure 4, the relative improvement of the corrected gradient decreases with the network size.

# Statistical embedding – Collab example

https://colab.research.google.com/drive/1RI_FuDc76H12BdNc2NEfL51j7cfRXqd7?usp=sharing

# NNC of graph dynamics

We describe networked dynamical systems by ODEs of the form

$$\dot{\mathbf{x}}(t) = f\left(\mathbf{x}(\mathbf{t}), \mathcal{A}, \mathbf{u}(\mathbf{t})\right),$$

where $\mathbf{x}(\mathbf{t}) \in \mathbb{R}^{\mathbf{N}}$ denotes the state vector and $\mathbf{u}(\mathbf{t}) \in \mathbb{R}^{\mathbf{M}}$ $(M \leq N)$ an external control, and $\mathcal{A}$ denotes the adjacency matrix.
We want to learn control signal $\mathbf{u}(\mathbf{t})$ with Neural Network.

# NNC of graph dynamics

Published: 11 May 2011

## Controllability of complex networks

Yang-Yu Liu, Jean-Jacques Slotine & Albert-László Barabási

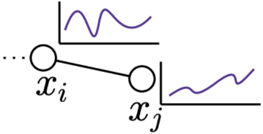
*Nature* **473**, 167–173(2011) | Cite this article



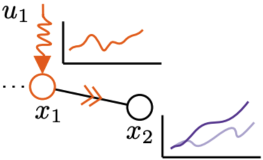
### Abstract

The ultimate proof of our understanding of natural or technological systems is reflected in our ability to control them. Although control theory offers mathematical tools for steering

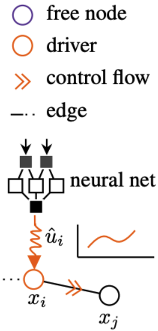
# NNC of graph dynamics




**(a)** No control.




**(b)** Control.




**(c)** Neural network control.

# NNC of graph dynamics

Example of Binary ODE dynamics with no control

$$\dot{x}_i(t) = L(x_i(t)) + \sum_{j=1}^{N} \mathcal{A}_{i,j} Q(x_i(t), x_j(t)), \qquad (3)$$

where $L(x_i(t))$ describes self-interactions and $Q(x_i(t), x_j(t))$ accounts for pair-wise interactions between neighbours.

With appropriate choices of functions $L(\cdot)$ and $Q(\cdot)$, the ODE system can model epidemic processes, biochemical dynamics, birth–death processes, and regulatory dynamics.

# NNC of graph dynamics



**Algorithm 1:** A generic algorithm that describes the parameter learning of NNC.

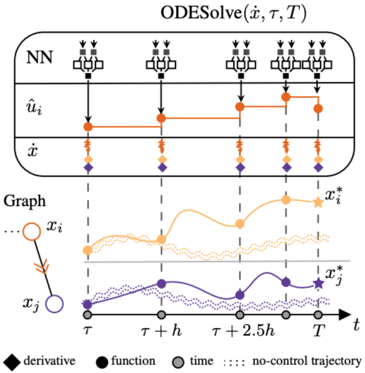
**Result:** $\boldsymbol{w}$
**Init::** $\boldsymbol{x_0}, \boldsymbol{w}, f(\cdot)$, ODESolve$(\cdot)$, $J(\cdot), \boldsymbol{x}^*$,use_adjoint;
**Params::** $\eta$, epoch;
epoch $\leftarrow$ 0;
**while** epoch $<$ epochs **do**
  $t \leftarrow 0$ ;
  $\boldsymbol{x} \leftarrow \boldsymbol{x_0}$;
  // Generate a trajectory based on NNC.
  $\boldsymbol{x} \leftarrow$ ODESolve$(\boldsymbol{x}, 0, T, f, \hat{\boldsymbol{u}}(\boldsymbol{x}, t; \boldsymbol{w}))$;
  // gradient descent update
  $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla_{\boldsymbol{w}} J(\boldsymbol{x}, \boldsymbol{x}^*)$;
  // or Quasi-Newton with Hessian:
  // $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta H^{-1} \nabla_{\boldsymbol{w}} J(\boldsymbol{x}, \boldsymbol{x}^*)$
**end**

**Algorithm 2:** A simple ODESolve implementation.

**Function** *ODESolve*$(\boldsymbol{x}, t, T, f, \hat{\boldsymbol{u}}(\boldsymbol{x}, t; \boldsymbol{w}))$:
  // Euler Method
  **while** $t \leq T$ **do**
    // Computational graph is
    // preserved through time
    // gradients flow through $\boldsymbol{x}$
    $\hat{\boldsymbol{u}} \leftarrow \hat{\boldsymbol{u}}(\boldsymbol{x}, t; \boldsymbol{w})$;
    $\boldsymbol{x} \leftarrow \boldsymbol{x} + \tau f(\boldsymbol{x}, \boldsymbol{u})$;
    // Step $\tau$ could be adaptive
    $t \leftarrow t + \tau(t)$;
  **end**
  **return** $\boldsymbol{x}$;
**end**

Where the loss function is MSE between reached
$J(\mathbf{x(t)}, \mathbf{x}^*) = \frac{1}{\mathbf{N}} ||\mathbf{x(t)} - \mathbf{x}^*||_2^2$

# NNC of graph dynamics



# NNC of graph dynamics

PRL **108,** 218703 (2012) PHYSICAL REVIEW LETTERS week ending 25 MAY 2012

## Controlling Complex Networks: How Much Energy Is Needed?

Gang Yan,[1] Jie Ren,[2,3] Ying-Cheng Lai,[4] Choy-Heng Lai,[2] and Baowen Li[2,5]

[1]*Temasek Laboratories, National University of Singapore, 117411, Singapore*

[2]*Department of Physics and Centre for Computational Science and Engineering, National University of Singapore, 117542, Singapore*

[3]*Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

[4]*School of Electrical, Computer and Energy Engineering, Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

[5]*Center for Phononics and Thermal Energy Science, Department of Physics, Tongji University, 200092, Shanghai, China*

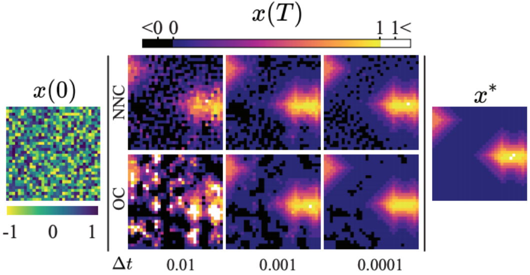
(Received 28 November 2011; published 23 May 2012)

The outstanding problem of controlling complex networks is relevant to many areas of science and engineering, and has the potential to generate technological breakthroughs as well. We address the physically important issue of the energy required for achieving control by deriving and validating scaling laws for the lower and upper energy bounds. These bounds represent a reasonable estimate of the energy cost associated with control, and provide a step forward from the current research on controllability toward ultimate control of complex networked dynamical systems.

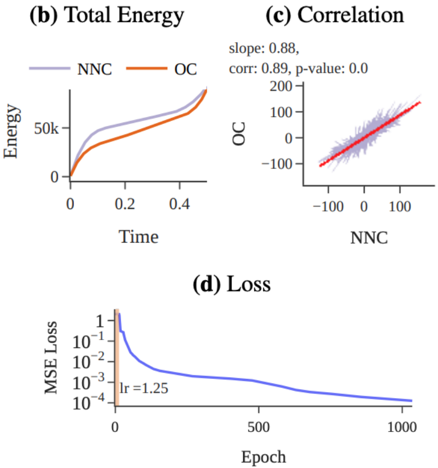
DOI: 10.1103/PhysRevLett.108.218703 PACS numbers: 89.75.Fb, 64.60.aq

# LTI and optimal control

**(a)** Reached states for NNC, OC under interaction intervals of varying length.



# LTI and optimal control



# Why NNC works?

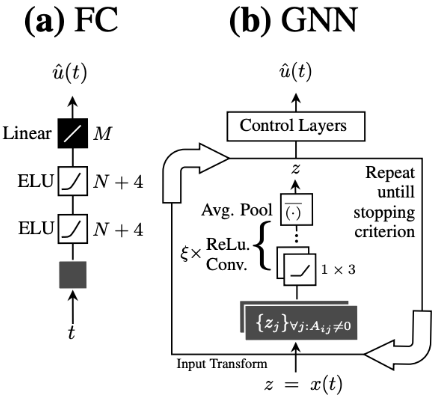


Why NNC produces near-optimal control signals if loss function $J(.)$ is not taking explicitly the control energy into the account? Optimal control functional

$$\int_0^T [u(t)]^2 dt + J(x(T), x^*), \text{s.t.} \dot{x} = f(u(t), \mathcal{A}, x(t))$$

- Neural network initialization
- Induced gradient property

# Which architecture? Does it matter?




Yes, for the number of parameters. GNN are more parameter efficient for non-linear dynamics.

# What happens with highly non-linear systems (SIRX) ?

$$\dot{S}_i(t) = -\beta S_i(t) \sum_j \mathcal{A}_{i,j} I_j(t) - \left(\kappa_{0,i} + \hat{u}_i(t) \mathbb{1}_{i \in \mathbb{B}}\right) S_i(t)$$

$$\dot{I}_i(t) = \beta S_i(t) \sum_j \mathcal{A}_{i,j} I_j(t) - \gamma I_i(t) - (\kappa_{0,i} + \hat{u}_i(t) \mathbb{1}_{i \in \mathbb{B}}) I_i(t)$$

$$\dot{R}_i(t) = \gamma I_i(t) + \left(\kappa_{0,i} + \hat{u}_i(t) \mathbb{1}_{i \in \mathbb{B}}\right) S_i(t),$$

$$\dot{X}_i(t) = \left(\kappa_{0,i} + \hat{u}_i(t) \mathbb{1}_{i \in \mathbb{B}}\right) I_i(t),$$

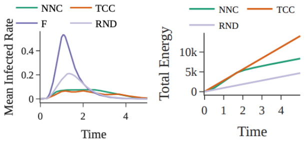
SIRX model (Maier and Brockmann, Science 2020) extended with networks and controls.

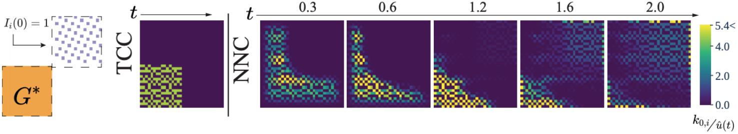
# NNC on network SIRX

**Table 2:** Energy and peak infection evaluation.

| Control | $\max_t(\bar{I}(t))$ | $E(T)$ |
|---|---|---|
| TCC | 0.068 | 14059.7 |
| NNC | 0.078 | 8354.7 |
| RND | 0.210 | 4687.0 |
| F | 0.430 | 0.0 |

**Figure 5:** SIRX control evaluation.



# NNC example – Collab example

https://colab.research.google.com/drive/17v2AOm0a8lMDUaQ7GkWGdn7G5Ulp1Wbt?usp=sharing

# Network dismantling

# Network Dismantling

# NEWS & VIEWS

For News & Views online, go to nature.com/newsandviews

NETWORK SCIENCE

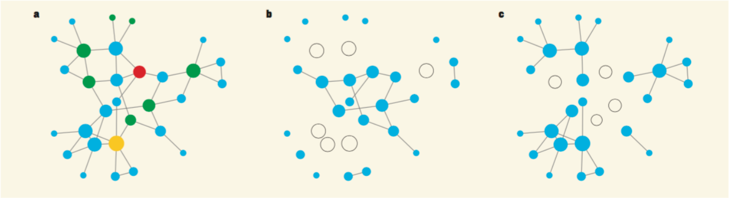
## Destruction perfected

**Pinpointing the nodes whose removal most effectively disrupts a network has become a lot easier with the development of an efficient algorithm. Potential applications might include cybersecurity and disease control.** **SEE LETTER P.65**

ISTVÁN A. KOVÁCS
& ALBERT-LÁSZLÓ BARABÁSI

which deletion would cause maximum damage is a non-deterministic polynomial-time hard

(known as influencers) and recalculating the collective influence of the rest following each



# Network Dismantling

## Network dismantling

**Alfredo Braunstein[a,b,c], Luca Dall'Asta[a,c], Guilhem Semerjian[d], and Lenka Zdeborová[e,1]**

[a]Department of Applied Science and Technology, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy; [b]Human Genetics Foundation, Via Nizza 52, 10126 Turin, Italy; [c]Collegio Carlo Alberto, Via Real Collegio 30, 10024 Moncalieri, Italy; [d]Laboratoire de Physique Théorique, Ecole Normale Supérieure, Paris Sciences et Lettres (PSL) Research University, Sorbonne Universités, Université Pierre et Marie Curie (UPMC), CNRS, 75005 Paris, France; and [e]Institut de Physique Théorique, CNRS, Commissariat à l'Energie Atomique (CEA) and Université Paris Saclay, 91191 Gif-sur-Yvette, France

Edited by Giorgio Parisi, University of Rome, Rome, Italy, and approved September 16, 2016 (received for review March 29, 2016)

**We study the network dismantling problem, which consists of determining a minimal set of vertices in which removal leaves the network broken into connected components of subextensive size. For a large class of random graphs, this problem is tightly connected**

information bits or ensure possibility of transportation). An adversary might be able to destroy a set of nodes with the goal of destroying this functionality. It is thus important to understand what an optimal attack strategy is, possibly as a first step in the

# 2. Cost of attack

**Current assumption***:

cost of removing any node in a network is the same.

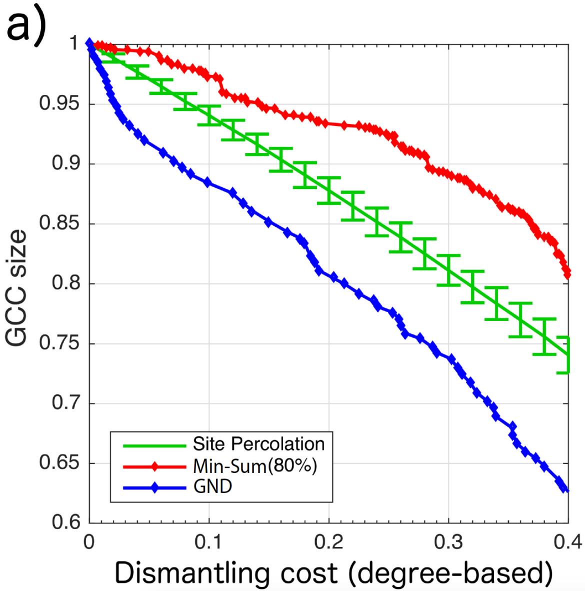
* Braunstein A, Dall'Asta L, Semerjian G, Zdeborová L (2016) Network dismantling. Proceedings of the National Academy of Sciences 113(44):12368–12373.
Morone F, Makse HA (2015) Influence maximization in complex networks through optimal percolation. Nature 524(7563):65–68
Kovács IA, Barabási AL (2015) Network science: Destruction perfected. Nature 524(7563):38–39.
Zdeborová L, Zhang P, Zhou HJ (2016) Fast and simple decycling and dismantling of networks. Scientific Reports 6.
Mugisha S, Zhou HJ (2016) Identifying optimal targets of network attack by belief propagation. Phys. Rev. E 94(1):012305.
Morone F, Min B, Bo L, Mari R, Makse HA (2016) Collective influence algorithm to find influencers via optimal percolation in massively large social media. Scientific Reports 6(1). Tian L, Bashan A, Shi DN, Liu YY (2017) Articulation points in complex networks. Nature Communications 8:14223.
Zhou HJ (2013) Spin glass approach to the feedback vertex set problem. The European Physical Journal B 86(11):455.

# 2. Cost of attack

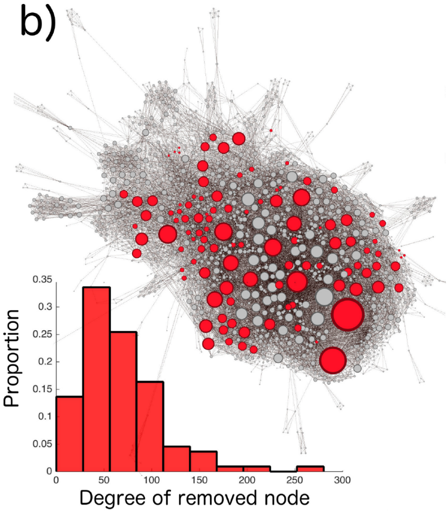
**Our assumption****:
cost of removing any node in a network is **not the same**. It can be arbitrary non-negative weight e.g. degree.

**Weights taken into the account:
Patron A, Cohen R, Li D, Havlin S (2017) Optimal cost for strengthening or destroying a given network. Phys. Rev. E 95(5):052305.
Ren XL, Gleinig N, Tolic D, Antulov-Fantulin N (2017) Underestimated cost of targeted attacks on complex networks, arxiv:1710.03522

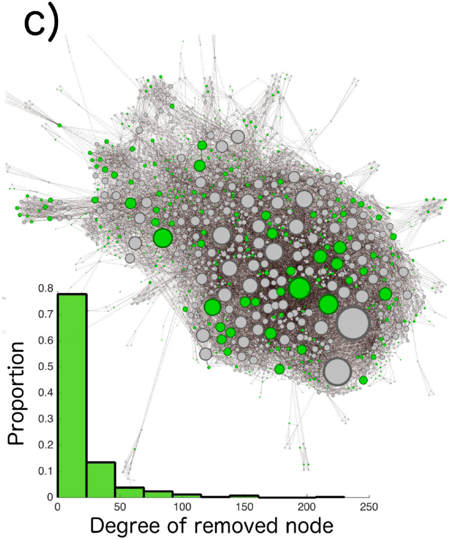
# Example 1: Online social network



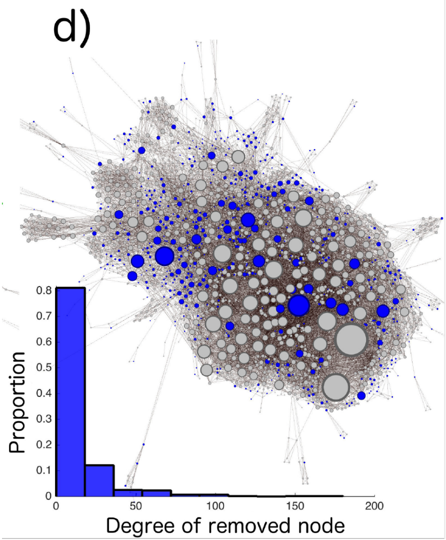
# Example 1: Online social network



# Example 1: Online social network



# Example 1: Online social network



# 3. New approach – objective function

Then the upper bound for the cost of removing a subset of nodes that are adjacent to the edges separating clusters $M$ and $\overline{M}$ is:

$$\frac{1}{2}\sum_{i,j} -\frac{1}{2}\left(v_i v_j - 1\right) A_{i,j}\left(w_i + w_j - 1\right), \tag{4}$$

where the matrix $A$ denotes the adjacency matrix of the network.

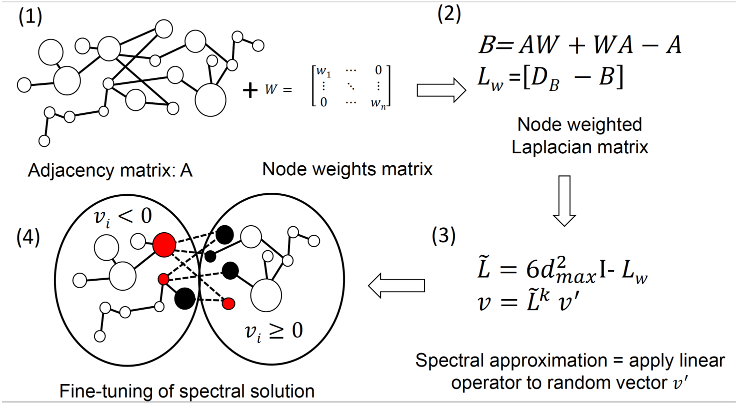
# 3. New approach – objective function

Upper bound of the cost (number of removed edges)

$$\min \frac{1}{4} x^t L_w x \tag{5}$$

where $L_w$ is modified Laplacian, which taked into the account the weights. where $L_w = D_B - B$, $B = AW + WA - A$, matrices $W$, $D_B$ are diagonal matrices with elements: $W_{ii} = w_i$, $(D_B)_{ii} = \sum_{j=1}^{n} B_{ij}$, respectively.

# Overall




We show $|\lambda_n| \leq 4d_{max}(w_{max} + 1)$, where $w_{max}$ is the largest cost and $d_{max}$ is the maximal degree.

# Convergence (optional)

**Theorem 1** *If $v$ is the output of the previous algorithm, then*

*a)*

$$|\lambda_2 - \frac{v^T L_w v}{v^T v}| \leq \frac{12 \cdot d_{max}^2 \cdot n}{\psi_2^2} \cdot |\frac{\tilde{\lambda}_3}{\tilde{\lambda}_2}|^{2\eta(n)} = \frac{12 \cdot d_{max}^2 \cdot n}{\psi_2^2} \cdot |\frac{6 \cdot d_{max}^2 - \lambda_3}{6 \cdot d_{max}^2 - \lambda_2}|^{2\eta(n)} \qquad [11]$$

*b) If $\eta(n)$ grows assymptotically faster than*

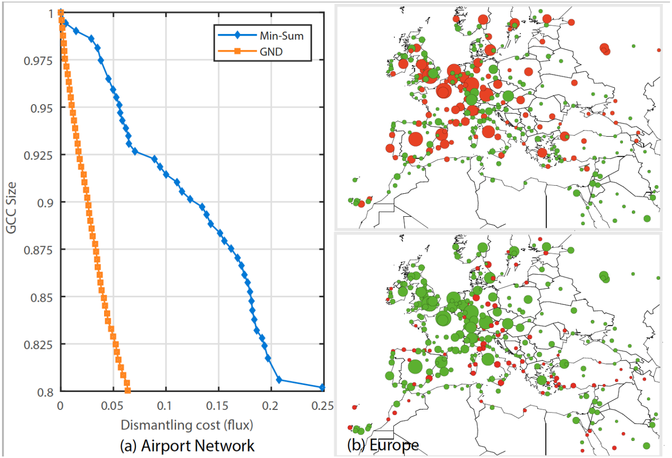
$$\frac{11}{2\log\left(|\frac{\tilde{\lambda}_2}{\tilde{\lambda}_3}|\right)} \log(n), \qquad [12]$$

*then*

$$\mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}|\right] \to 0. \qquad [13]$$

To dismantle large-scale networks, we propose an efficient iterative node-weighted spectral bisection method, which has complexity $O(n \log^{2+\epsilon}(n))$

# Airports



1
0.975
0.95
0.925
0.9
0.875
0.85
0.825
0.8
GCC Size
0
0.05
0.1
0.15
0.2
0.25
Dismantling cost (flux)
Min-Sum
GND
(a) Airport Network
(b) Europe

# Publication

## Generalized network dismantling

Xiao-Long Ren[a,1], Niels Gleinig[b,1], Dirk Helbing[a], and Nino Antulov-Fantulin[a,2]

[a]Computational Social Science, ETH Zürich, 8092 Zürich, Switzerland; and [b]Department of Computer Science, ETH Zürich, 8092 Zürich, Switzerland

Edited by H. Eugene Stanley, Boston University, Boston, MA, and approved February 15, 2019 (received for review April 12, 2018)

**Finding an optimal subset of nodes in a network that is able to efficiently disrupt the functioning of a corrupt or criminal organization or contain an epidemic or the spread of misinformation is a highly relevant problem of network science. In this paper, we address the generalized network-dismantling problem, which**

find good approximations of the optimal dismantling strategy. For example, novel approximations (17–23) have been proposed based on spin-glass and optimal percolation theory. However, all these methods make the implicit assumption that the cost of removing nodes is the same. Only recently have people been

# Publication * Ensemble approach

## Ensemble approach for generalized network dismantling

Xiao-Long Ren and Nino Antulov-Fantulin

ETH Zurich, Zrich CH-8092, Switzerland
To whom correspondence should be addressed: `anino@ethz.ch`

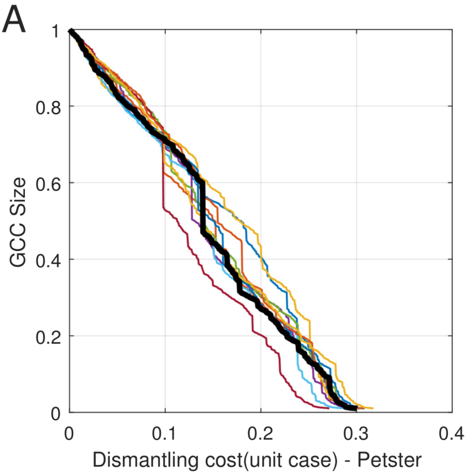


**Abstract.** Finding a set of nodes in a network, whose removal fragments the network below some target size at minimal cost is called network dismantling problem and it belongs to the NP-hard computational class. In this paper, we explore the (generalized) network dismantling problem by exploring the spectral approximation with the variant of the power-iteration method. In particular, we explore the network dismantling solution landscape by creating the ensemble of possible solutions from different initial conditions and a different number of iterations of the spectral approximation.

**Keywords:** network dismantling, spectral partitioning, robustness

# Publication * Ensemble approach

How many iterations?



# Publication * Ensemble approach



By design allow variability in the dismantling solutions by exploring the solution landscape from different initial conditions.

In the ensemble approach, we will produce $K$ different dismantling solutions $S_1$,$S_2$,...$S_K$, and take the one with the with the **minimum cost**:

$$S^* = \arg\min \{cost(S1), cost(S2), ..., cost(S_K)\}.$$

.

# Publication * Ensemble approach

| Unweighted case | BPD | Min-Sum | GND | Ensemble-GND | GNDR | Ensemble-GNDR |
|---|---|---|---|---|---|---|
| Crime Network | 101 | 120 | 110 | 103 | 103 | **99** |
| Petster Network | 474 | 485 | 601 | 510 | 467 | **441** |
| RoadEU Network | 151 | 160 | 193 | 159 | 171 | **144** |
| Political-blogs | **375** | 380 | 494 | 435 | 404 | 386 |

| Weighted case | BPD | Min-Sum | GND | Ensemble-GND | GNDR | Ensemble-GNDR |
|---|---|---|---|---|---|---|
| Crime Network | 0.594 | 0.644 | 0.642 | 0.624 | 0.584 | **0.572** |
| Petster Network | 0.829 | 0.837 | 0.914 | 0.873 | 0.810 | **0.792** |
| RoadEU Network | 0.463 | 0.491 | 0.523 | 0.470 | 0.464 | **0.417** |
| Political-blogs | 0.978 | 0.979 | 0.995 | 0.993 | 0.984 | **0.977** |

# GraphDQN

Published as a conference paper at ICLR 2019

## DISMANTLE LARGE NETWORKS THROUGH DEEP REINFORCEMENT LEARNING

**Changjun Fan**
National University of Defense Technology
fanchangjun@nudt.edu.cn

**Yizhou Sun**
University of California, Los Angeles
yzsun@cs.ucla.edu

**Li Zeng**
National University of Defense Technology
crack521@163.com

**Yang-Yu Liu**
Harvard Medical School
yyl@channing.harvard.edu

**Muhao Chen**
University of California, Los Angeles
muhaochen@ucla.edu

**Zhong Liu**
National University of Defense Technology
phillipliu@263.net

# Publication * Ensemble approach

| Unweighted case | GraphDQN | Ensemble-GND | Ensemble-GNDR |
|---|---|---|---|
| Crime2 Network | 185 | 183 | **161** |
| HI-II-14 Network | 553 | 483 | **412** |
| DBLP Network | 2496 | 2499 | **2064** |
| Digg Network | 4335 | 4826 | **3920** |
| **Weighted case** | **GraphDQN** | **Ensemble-GND** | **Ensemble-GNDR** |
| Crime2 Network | 0.989 | 0.802 | **0.718** |
| HI-II-14 Network | 0.977 | 0.942 | **0.831** |
| Digg Network | 0.975 | 0.930 | **0.806** |

- T. Funke, T; Guo, A. Lancic, N. Antulov-Fantulin, Low-dimensional statistical manifold embedding of directed graphs, 8th International Conference on Learning Representations (ICLR 2020)
- L. Bottcher, N. Antulov-Fantulin, Unifying susceptible-infected-recovered processes on networks, Phys. Rev. Research 2, 033121 – Published 22 July 2020
- X.L. Ren, N. Gleinig, D. Helbing, N. Antulov-Fantulin, Generalized network dismantling, PNAS April 2, 2019 116 (14) 6554-6559;
- T. Asikis, L. Bottcher, N. Antulov-Fantulin, NNC: Neural-Network Control of Dynamical Systems on Graphs, (2020), `https://arxiv.org/abs/2006.09773`

Photo credits:
Photo by Alex Chambers on Unsplash
Photo by Jeremy Perkins on Unsplash
Photo by Philip Brown on Unsplash
Photo by Sanwal Deen on Unsplash
Photo by Alex Guillaume on Unsplash
Photo by Keith Hardy on Unsplash