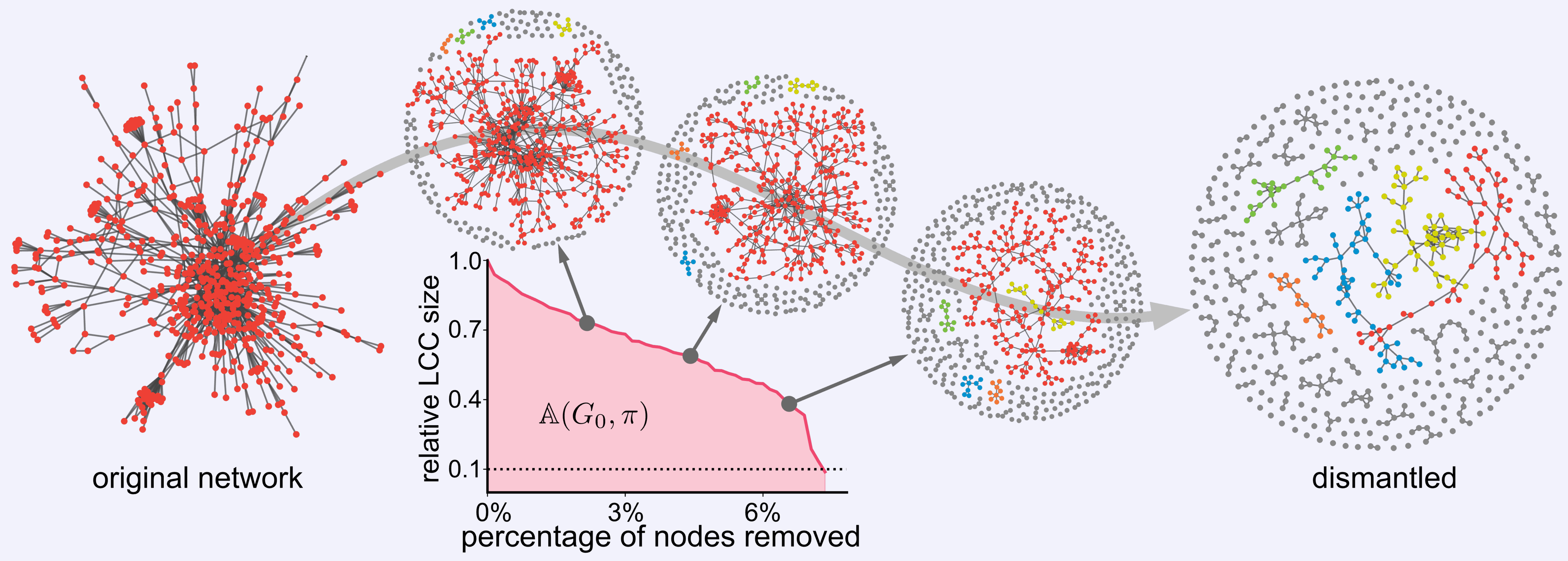
IMPERIAL

Learning Network Dismantling Without Handcrafted Features

Haozhe Tian,
Pietro Ferraro,
Robert Shorten,
Mahdi Jalili,
Homayoun Hamedmoghadam

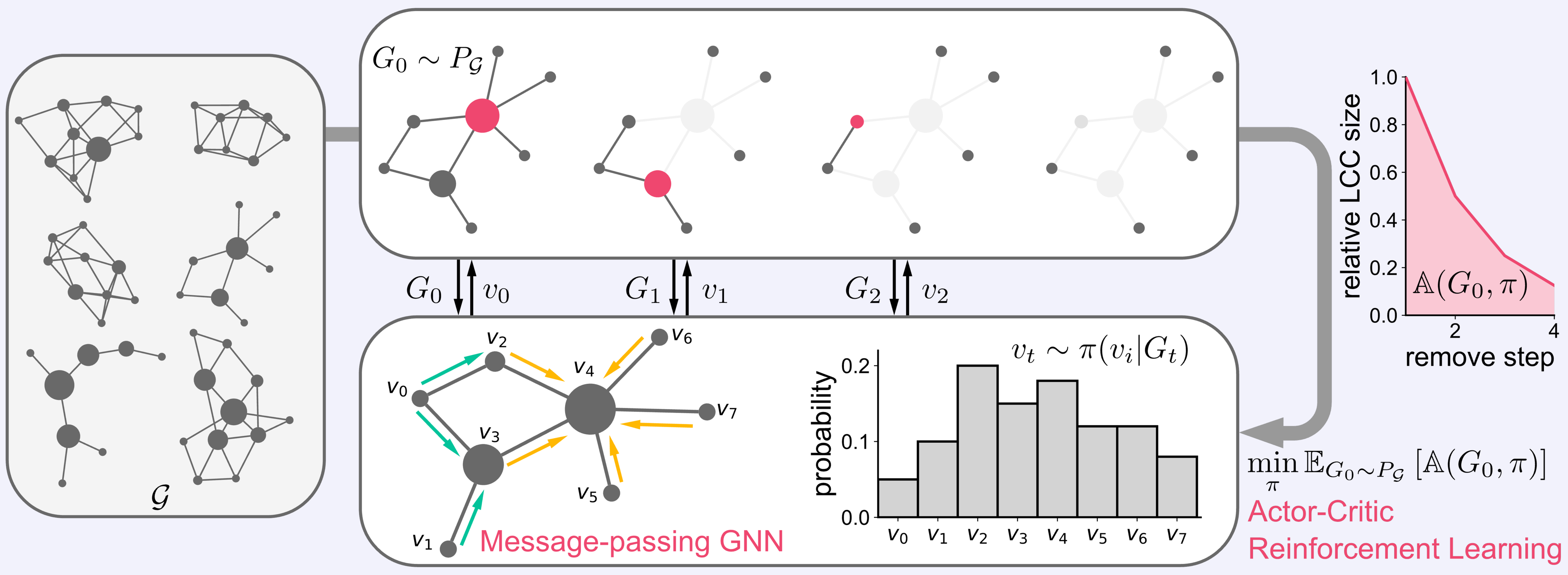
Network Dismantling



Network dismantling (ND) is a classic NP-hard problem that aims to find the order of node removals that minimize the area under the dismantling curve, denoted by $\mathbb{A}(G_0, \pi)$. This problem is extremely challenging:

- Huge policy search space ($|V|$ factorial for a network with $|V|$ nodes).
- Requires deep understanding of nodes' roles (one remove changes other nodes' roles).

We propose an efficient, data-driven planner: **Message Iteration Network Dismantler (MIND)**:



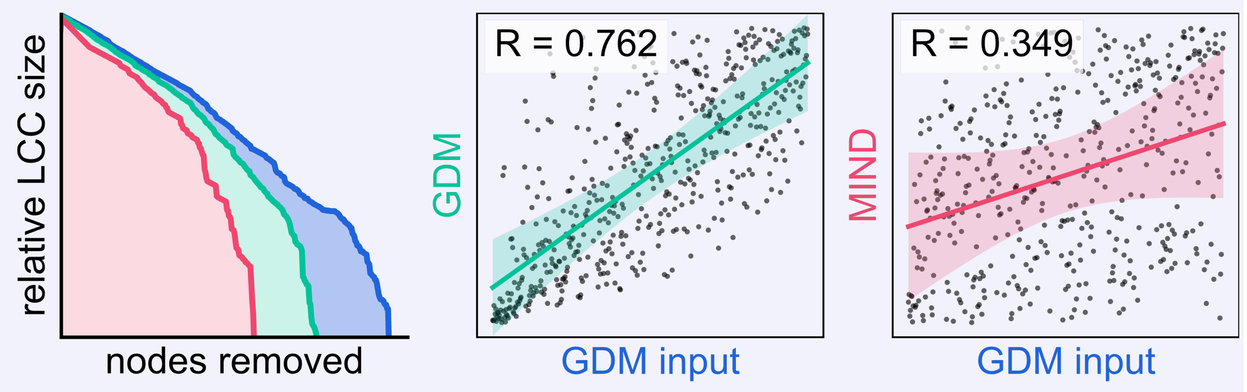
Why featureless?

1. Handcrafting input is computationally expensive

Fiedler vector as input	~6 s
Random walk as input	NA (OOM)
MIND forward pass	**~0.25s**

Computation time on the Hyves network (1.5 million nodes)

2. Input features can bias the dismantling policy



- MIND does not use features (uses all-ones).
- GDM is a SoTA baseline that uses input features.
- GDM input is derived using PCA on the GDM inputs.

GDM solution resembles its un-trained input and is less effective than MIND's data-driven solution.

Methodology

1. All-to-one attention mechanism and message profiles:

- Graph Attention Network:

$$\hat{e}_i^h = \sum_{j \in i \cup \mathcal{N}(i)} \alpha_{i,j}^h W^h e_j^h,$$

where e_i^h and $\alpha_{i,j}^h$ are the node embedding and the attention coefficient for the h-th head.

- All-to-one attention mechanism (**MIND-AM**):

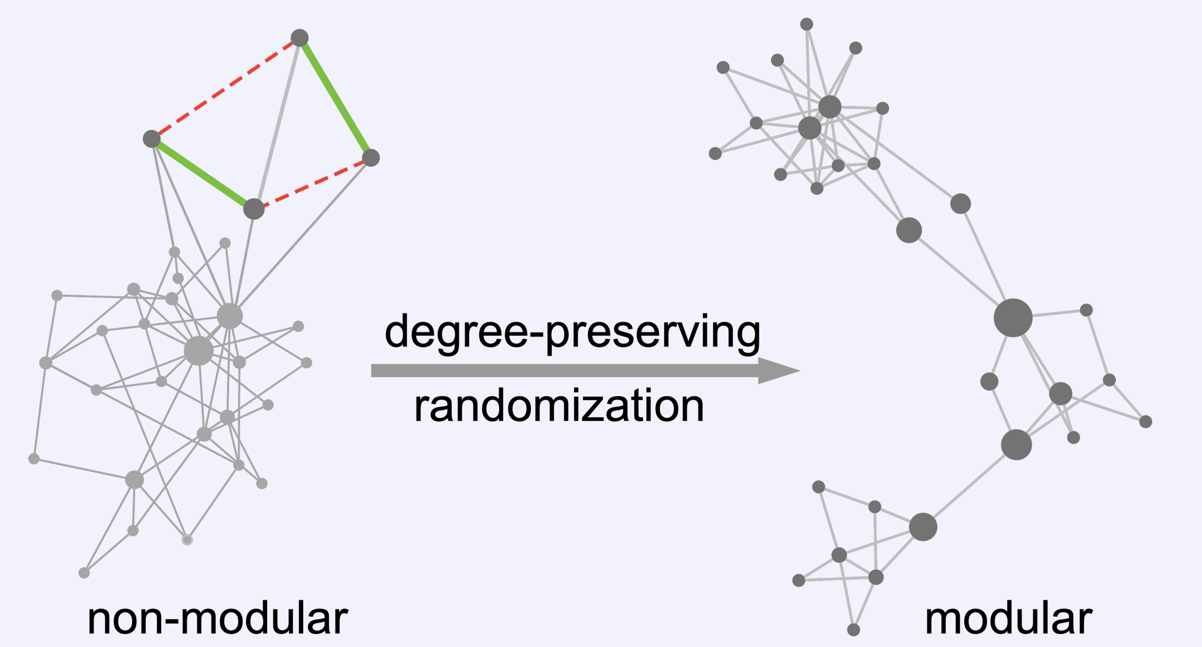
$$\alpha_{i,j}^h = \mathrm{MLP}_\alpha\left(\left[W^h e_i^1 \mid \cdots \mid W^h e_i^H \mid W^h e_j^1 \mid \cdots \mid W^h e_j^H\right]\right).$$

- Message profiles (**MIND-MP**):

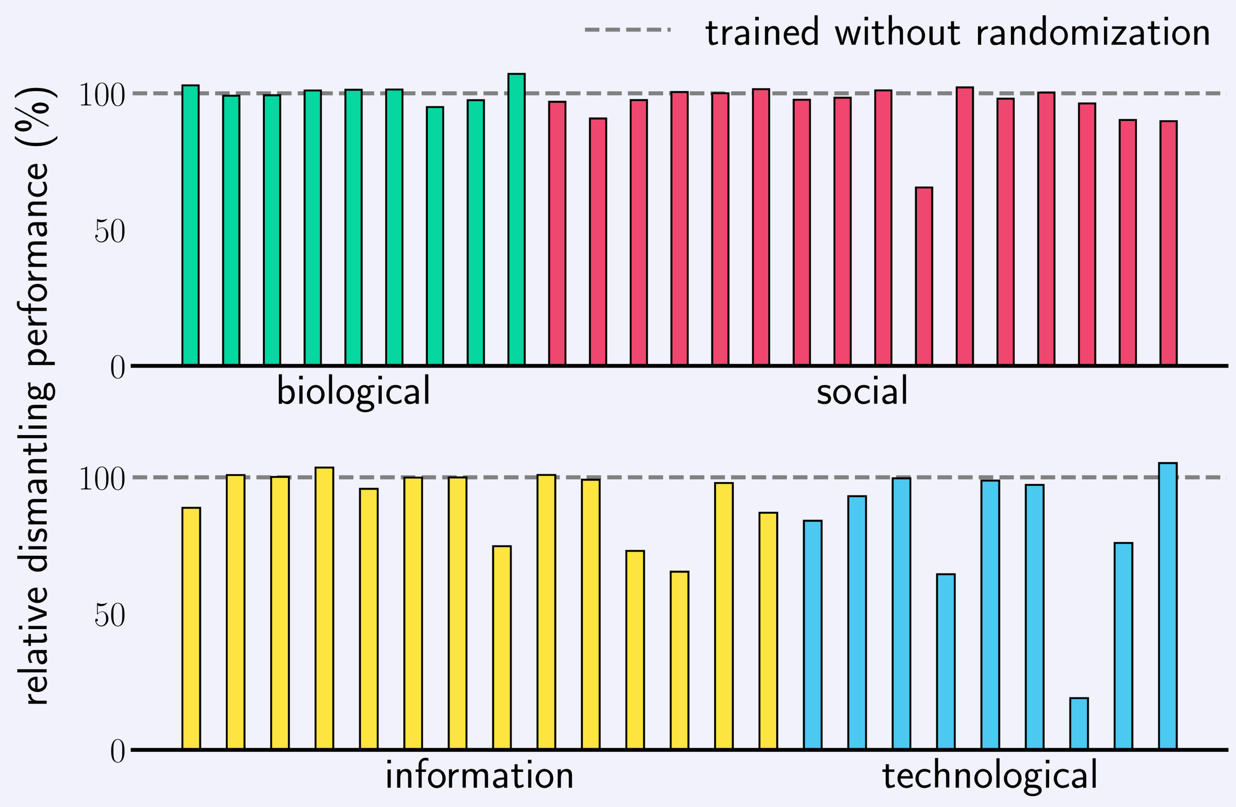
$$\pi(v_i \mid G_t) = \mathrm{MLP}_\pi\left(\left[e_i^{(1)} \mid \cdots \mid e_i^{(K)}\right]\right).$$

In our paper, we showed that these designs enabled MIND to extract commonly-used structural measures in the literature.

2. Systematic diversification of training networks:

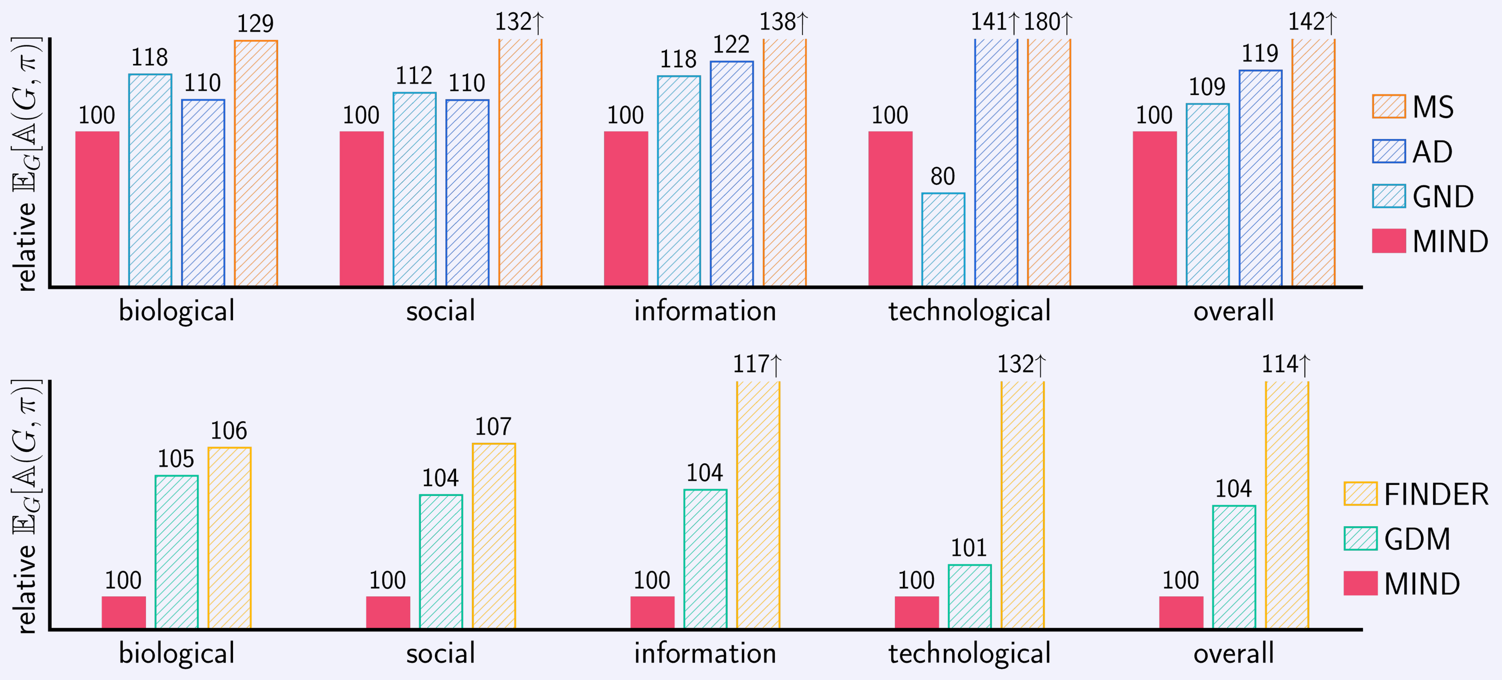


Randomly generated training networks are similar in properties. We use stochastic rewiring to make a more diverse training set.



Results and Applications

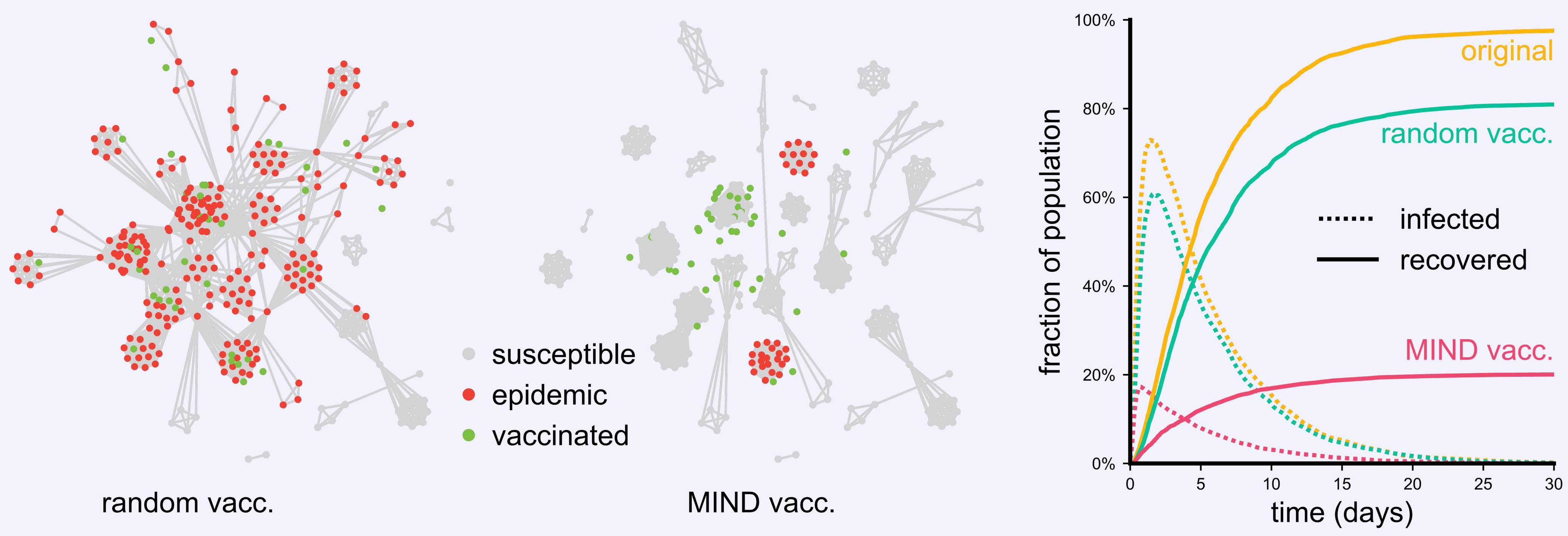
Deploying MIND trained on synthetic networks (150-250 nodes) to 48 real networks (up to 1.5 million nodes). MIND is faster than the baselines (with linear complexity) but achieves the strongest overall performance.



MS	GND	FINDER	GDM	MIND
$\mathcal{O}(\lvert V\rvert \log\lvert V\rvert) + \mathcal{O}(\lvert E\rvert)$	$\mathcal{O}(\lvert V\rvert \log^{2+\epsilon}\lvert V\rvert)$	$\mathcal{O}(\lvert V\rvert \log\lvert V\rvert) + \mathcal{O}(\lvert E\rvert)$	$\mathcal{O}(\lvert V\rvert \langle d^2\rangle^* + \lvert E\rvert)$	$\mathcal{O}(\lvert V\rvert + \lvert E\rvert)$

* from calculating the input feature of clustering coefficient

The SoTA performance of MIND directly translates to real-world solutions. E.g., controlling infectious disease using minimal, targeted vaccines. Compared to random vaccination, MIND breaks way more infection chains.



haozhe.tian21@imperial.ac.uk