Northeastern University
Network Science Institute
Khoury College of Computer Sciences

@tinaeliassi

Task-driven Network Discovery via Deep RL on Embedded Spaces

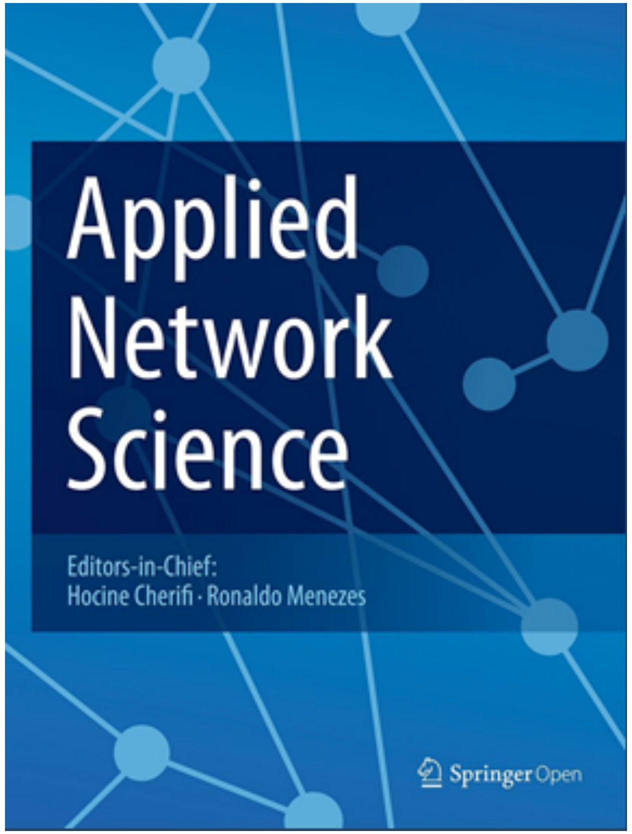
Tina Eliassi-Rad
tina@eliassi.org

Joint work with

Rajmonda Caceres
MIT LL

Peter Morales
MIT LL ➟ Microsoft

Forthcoming at

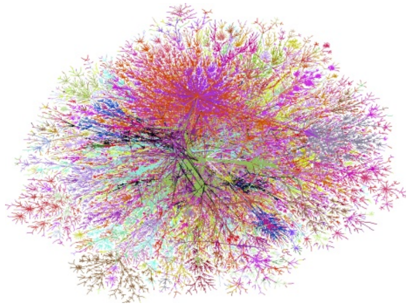


Preprint at https://bit.ly/3sozou8

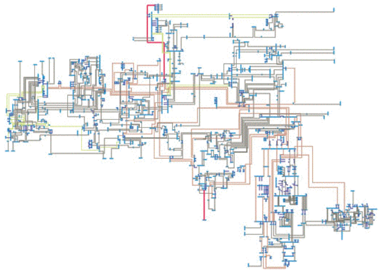
Complex networks are ubiquitous

Technological Networks

Internet

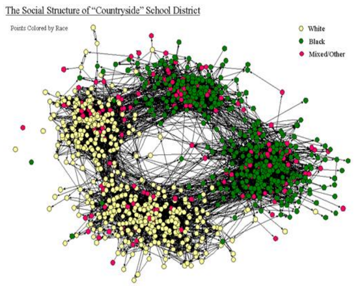


NY State Power Grid

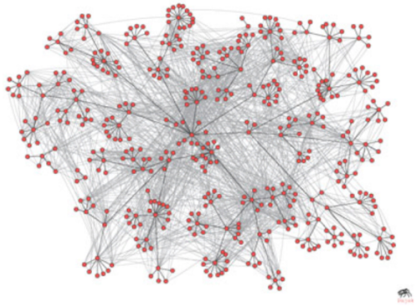


Social Networks

Friendship

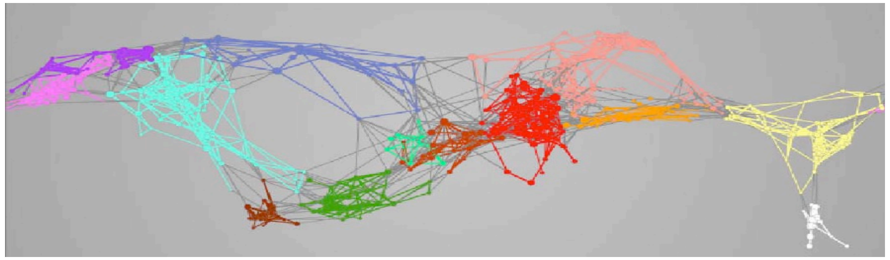


HP Emails



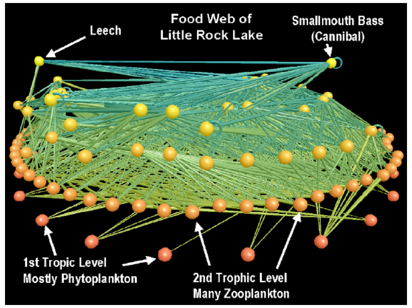
Information Networks

Map of Science

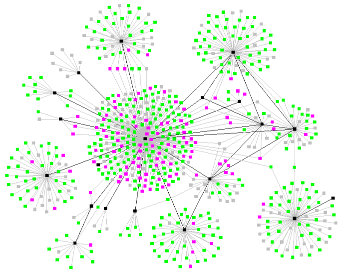


Biological networks

Food Web



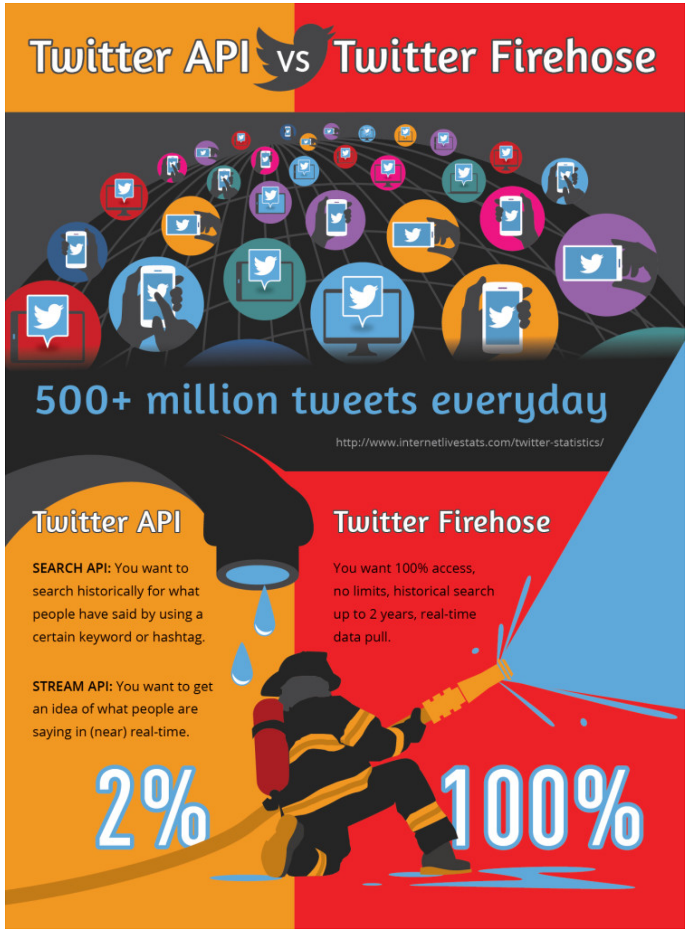
Contagion of TB



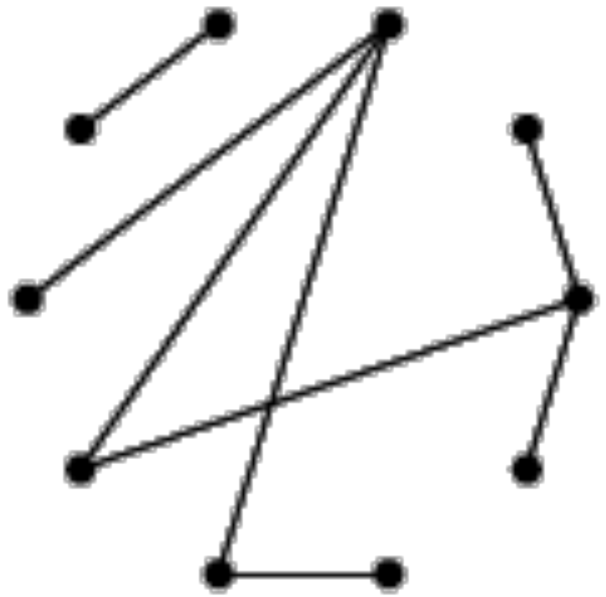
Partially observed

complex networks are ubiquitous

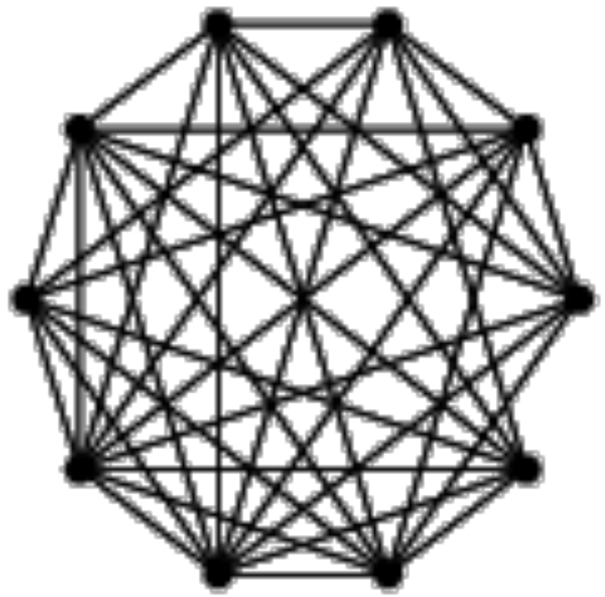
- Networked representations of real-world phenomena are often partially observed
- Acquiring more network data is often expensive and/or hard
- Even when your data is complete, you may not have the computational resources to examine all of it



Working with incomplete data can **skew** analyses



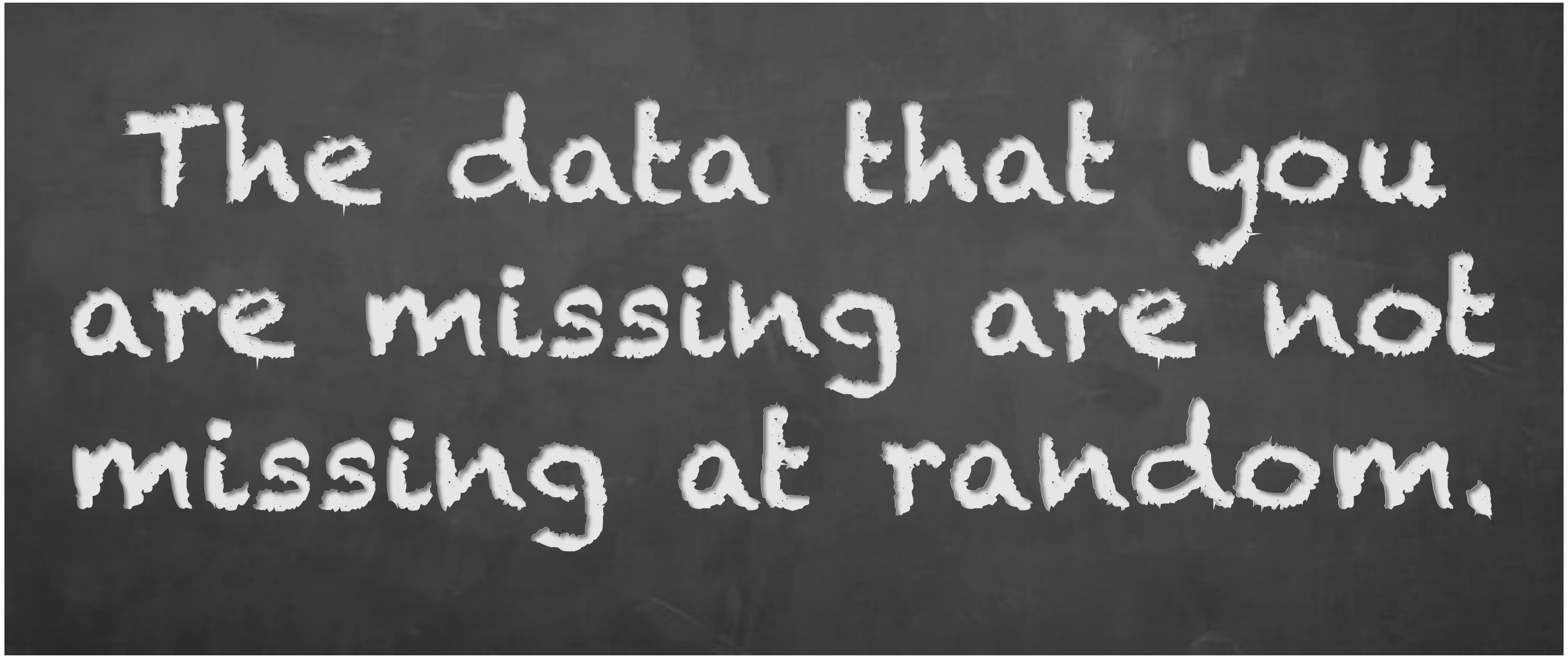
Partially observed network



Fully observed network

Average degree = 2 |E| / |V|

Working with incomplete data can **skew** analyses



The network discovery question

Given a query budget for identifying additional nodes and edges, how can one get a more accurate representation of the fully observed network?

The network discovery question

Given a query budget for identifying additional nodes and edges, how can one get a more accurate representation of the fully observed network?

A more accurate representation

Active Exploration

A more accurate representation involves **growing** the network by adding nodes and edges

Active Learning

A more accurate representation involves **learning the best performing function** on the network for a down-stream task such as selective harvesting

A more accurate representation

Active Exploration

A more accurate representation involves **growing** the network by adding nodes and edges

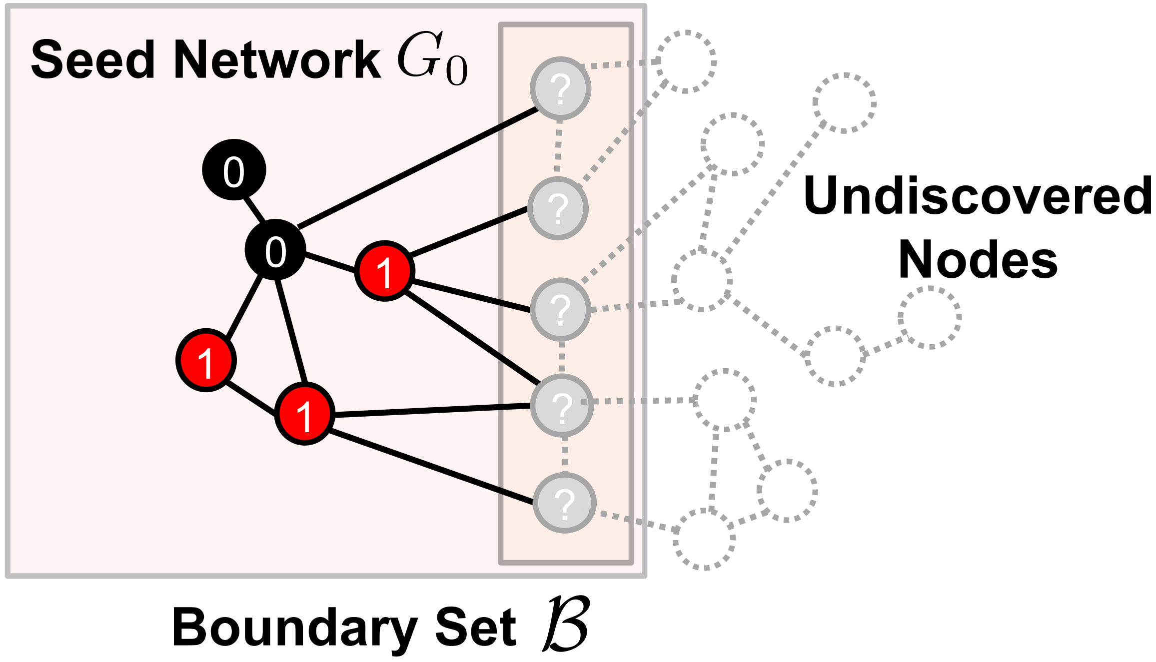
Active Learning

A more accurate representation involves **learning the best performing function** on the network for a down-stream task such as selective harvesting

Task-driven network discovery

Selective harvesting

Given a **seed network** G_0 and a **budget** b, grow the network to discover as many nodes of a particular type as possible

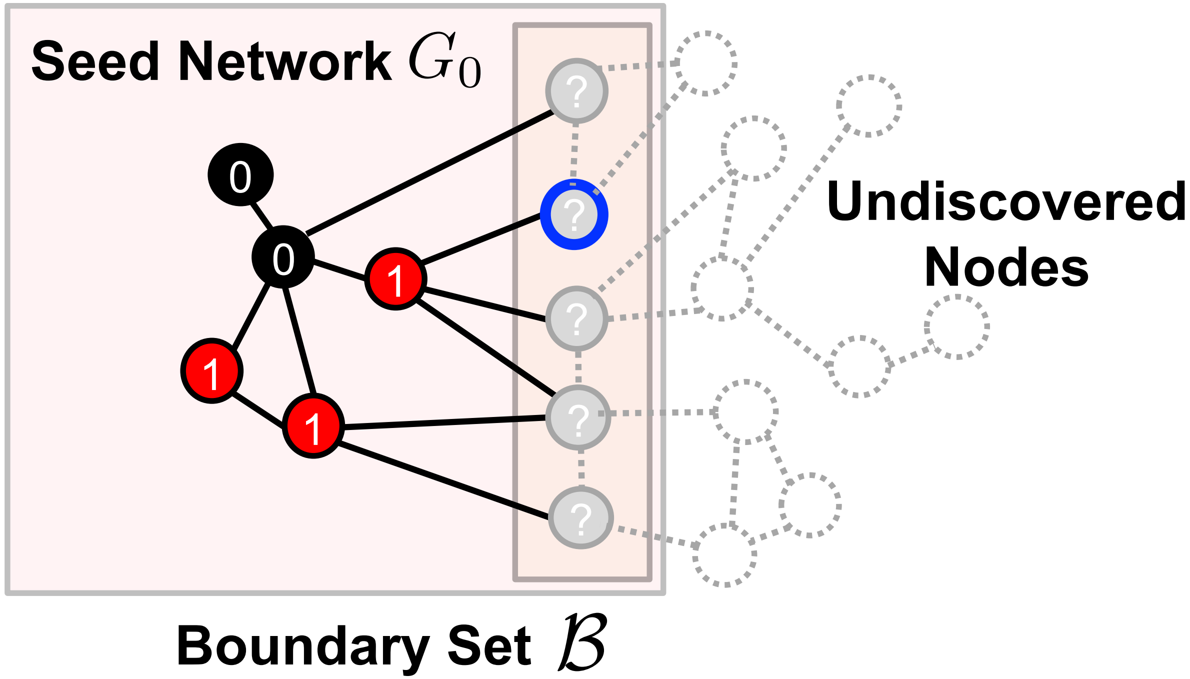


Selective harvesting

Given a **seed network** G_0 and a **budget** b, grow the network to discover as many nodes of a particular type as possible

Access mechanism

- Query a node that is observed but whose label is not known (boundary set)

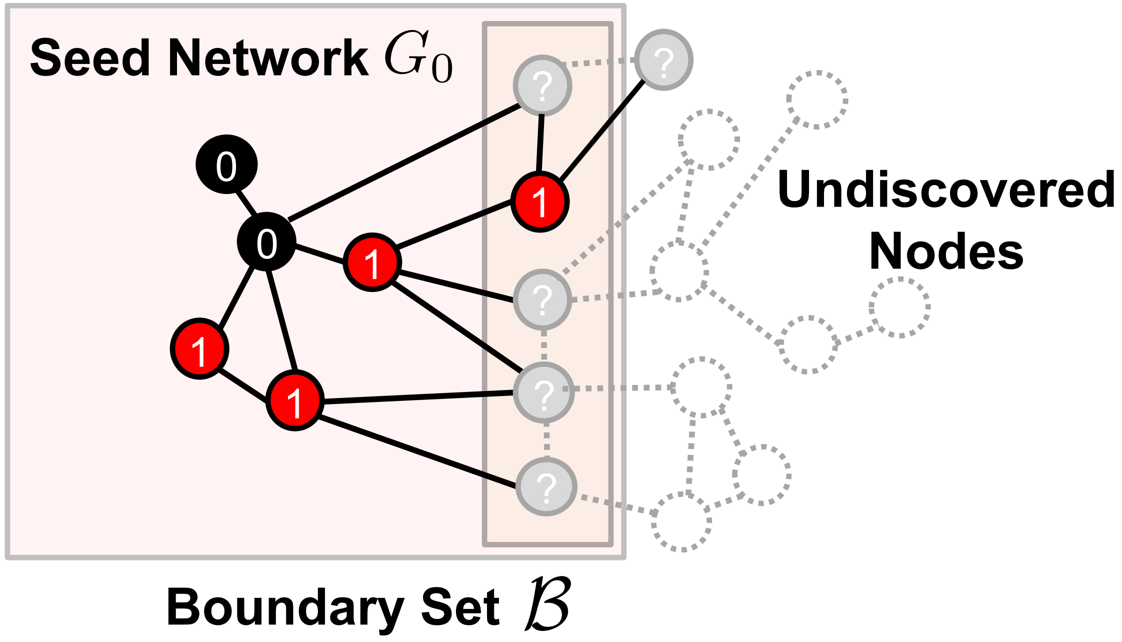


Selective harvesting

Given a **seed network** G_0 and a **budget** b, grow the network to discover as many nodes of a particular type as possible

<u>Access mechanism</u>

- Query a node that is observed but whose label is not known (boundary set)
- Query returns the label of the node and all of its neighbors (but not their labels)

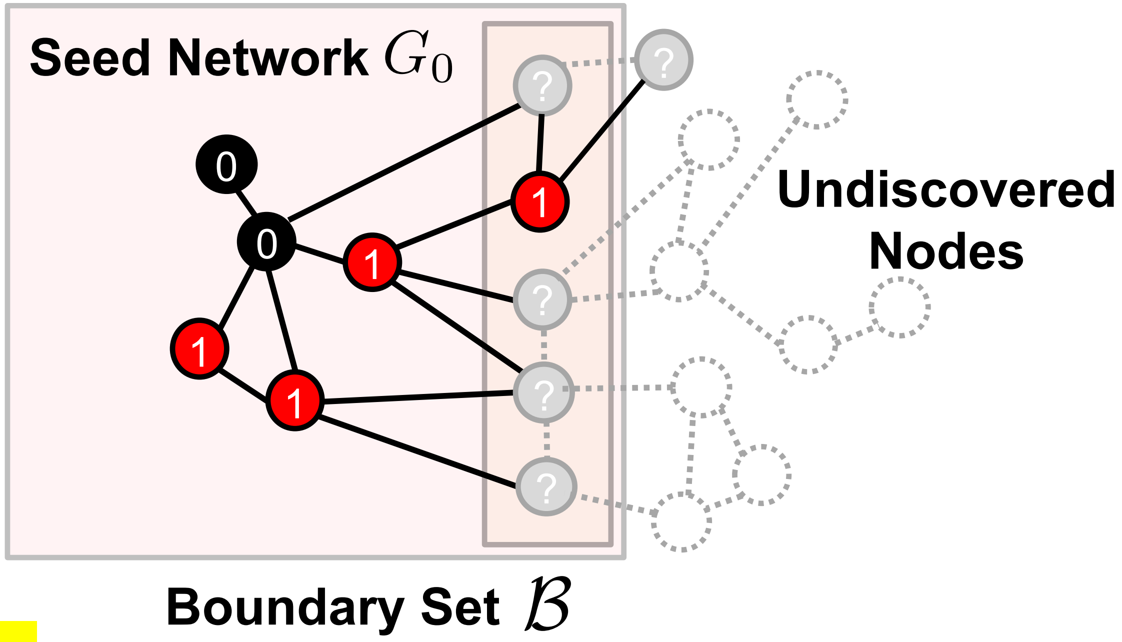


Selective harvesting

Given a **seed network** G_0 and a **budget** b, grow the network to discover as many nodes of a particular type as possible

Access mechanism

- Query a node that is observed but whose label is not known (boundary set)
- Query returns the label of the node and all of its neighbors (but not their labels)

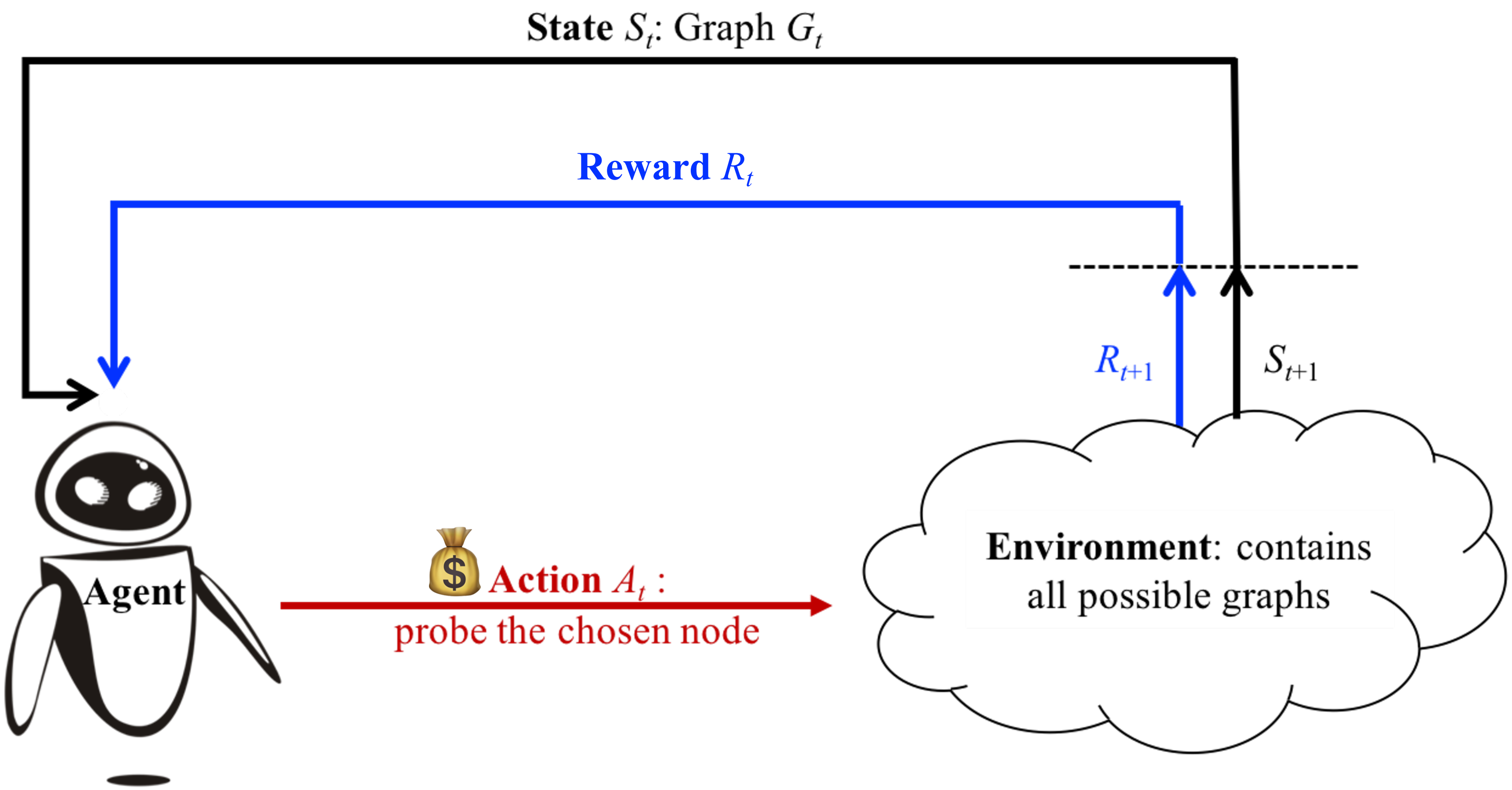


Goal: Learn how to select an appropriate boundary node to query

Some issues that make the problem difficult

- **Sparse signals**: you see relatively few examples of the relevant structure
- **Network effects**: relevant features/structures become evident in aggregate
- **Context-specific relevance**: there are multi-faceted notions of relevance
- **Complexity of invariant features**: you need to observe many variations of topology to understand relevant features

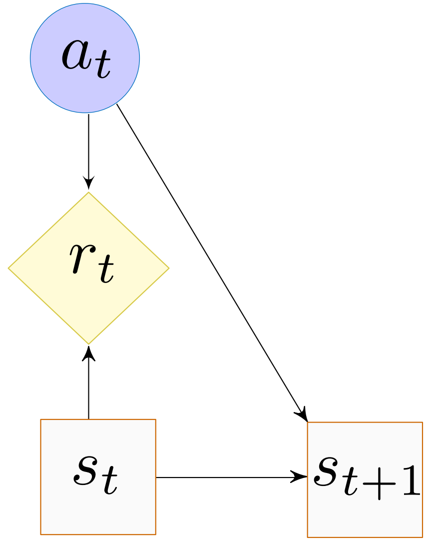
Selective harvesting via reinforcement learning



- Learn through an interaction paradigm
- ⟨State/action, reward⟩ pairs versus ⟨node, label⟩ pairs

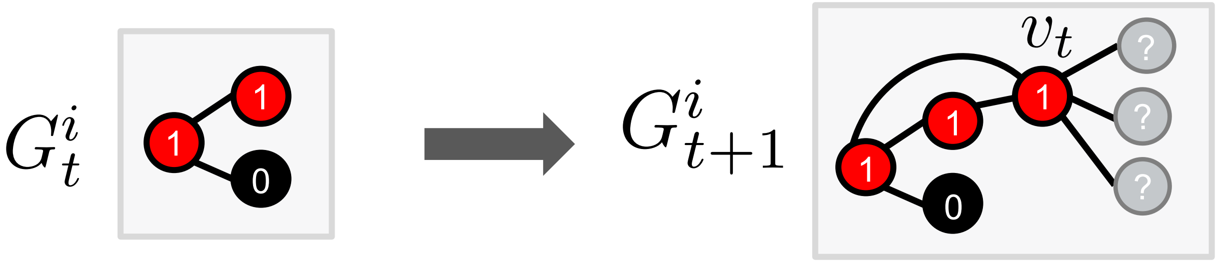
Selective harvesting as a MDP

State space *S*: Set of all intermediate networks defined over a set of vertices *V*, a set of random graph models $\{M^i\}$, and a labeling function $C(v) \in \{0, 1, *\}$



$$\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$$

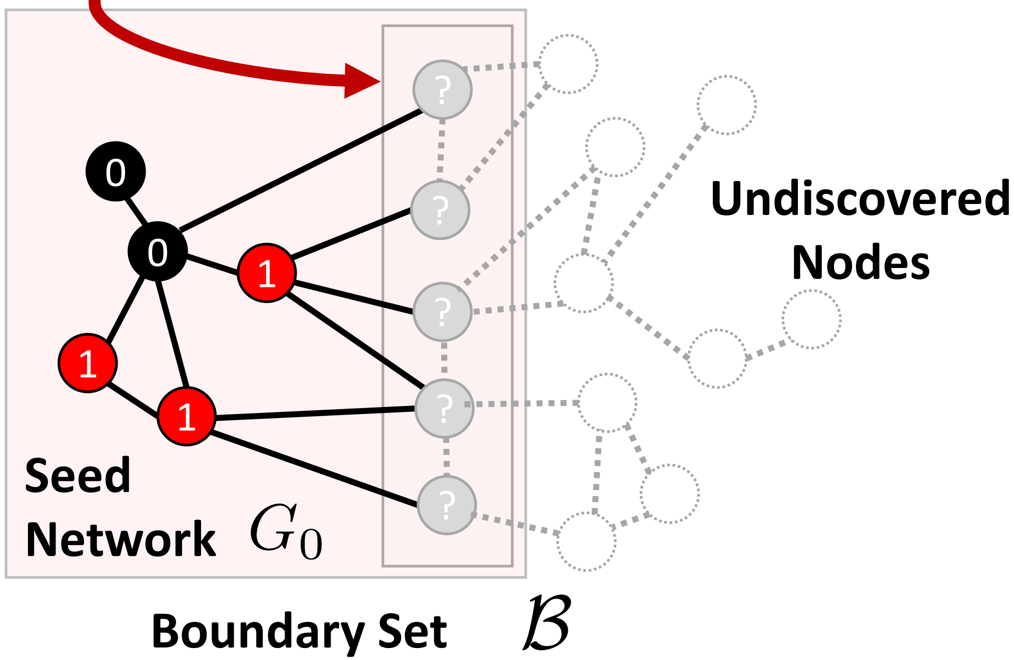
$$\mathcal{S} = \bigcup_{M^i} \{s_t = G_t^i\} \qquad G_t^i = \{V_t^i, E_t^i\}$$

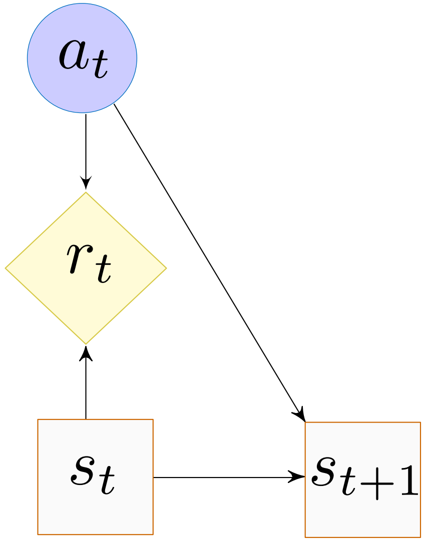


Selective harvesting as a MDP

Action space *A*: Nodes on the boundary set

$$\mathcal{A} = \{a_t\} = \{v_t\} \text{ where } v_t \in \mathcal{B}$$

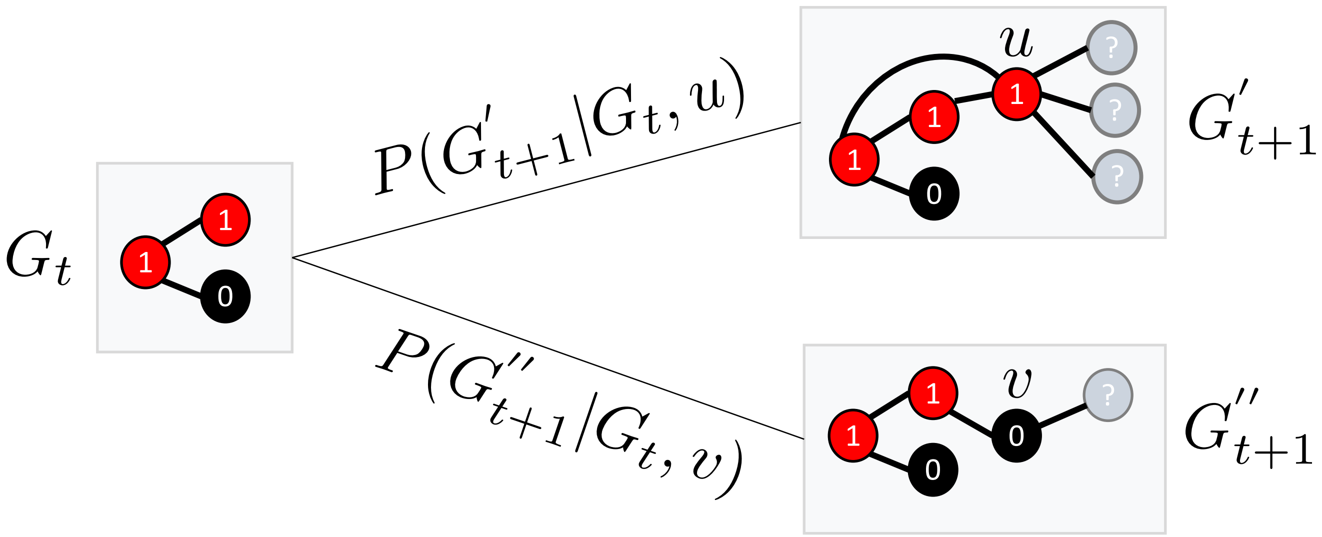


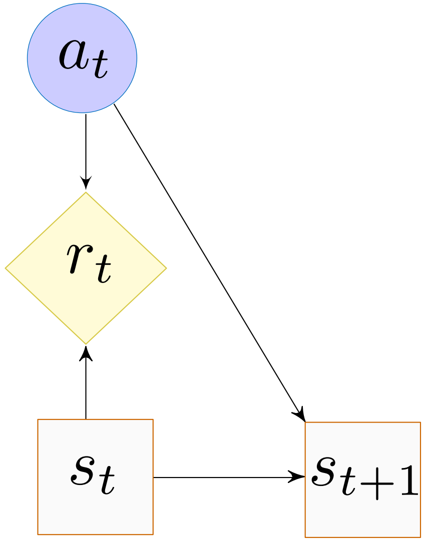


$$\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$$

Selective harvesting as a MDP

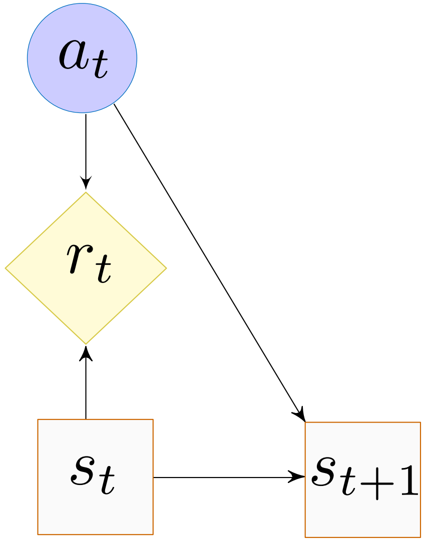
Transition function *T*: $T(s_t, a_t, s_{t+1}) = P\big(s_{t+1} \big| s_t, a_t\big)$





$\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$

Selective harvesting as a MDP

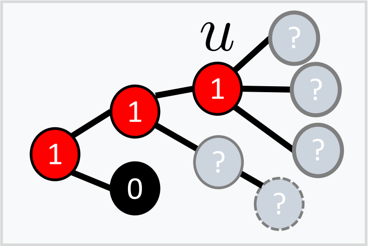


$$\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$$

Reward function *R*: $R: S \times A \rightarrow \mathbb{R}$

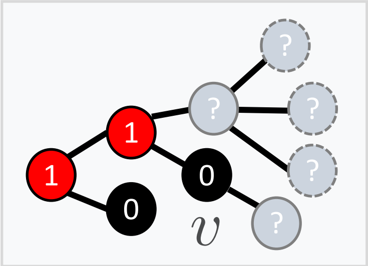
- In our case: reward for discovering a node with the relevant type

$$r_{t+1}(G'_{t+1} | G_t, u) = 1$$



G'_{t+1}

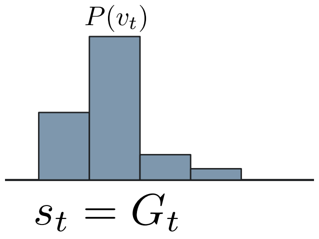
$$r_{t+1}(G''_{t+1} | G_t, v) = 0$$



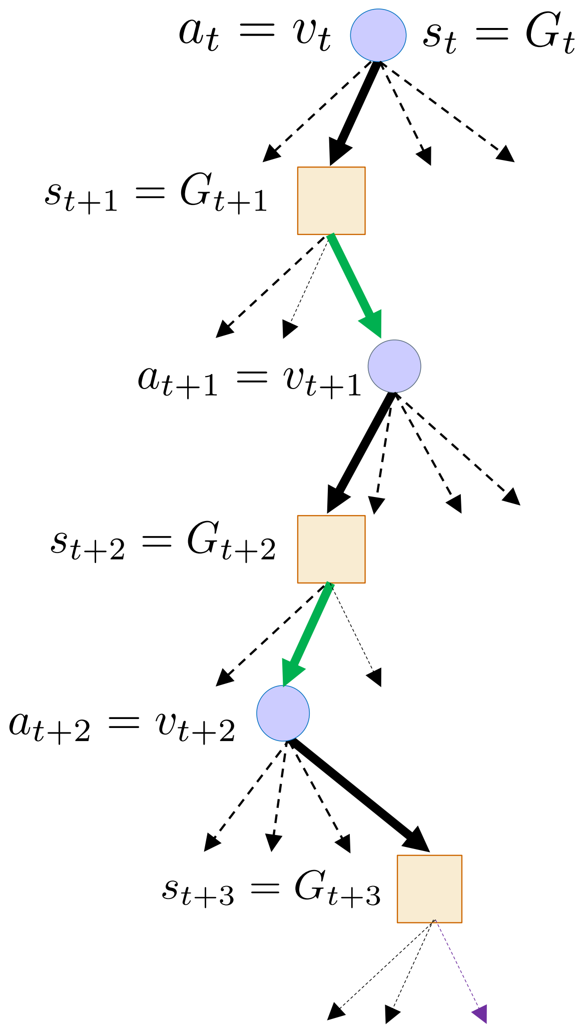
G''_{t+1}

Policy function

- **Policy** $\pi: S \times A \to \mathbb{R}$
  - $\pi(s,a) = P(a|s)$



- A **trajectory** τ_h is an instantiation of a policy over horizon *h*
  - $\tau_h = \{\langle a_t, G_{t+1}\rangle, \langle a_{t+1}, G_{t+2}\rangle, \cdots, \langle a_{t+h-1}, G_{t+h}\rangle\}$
- Our budget *b* imposes an upper bound on the length of these trajectories



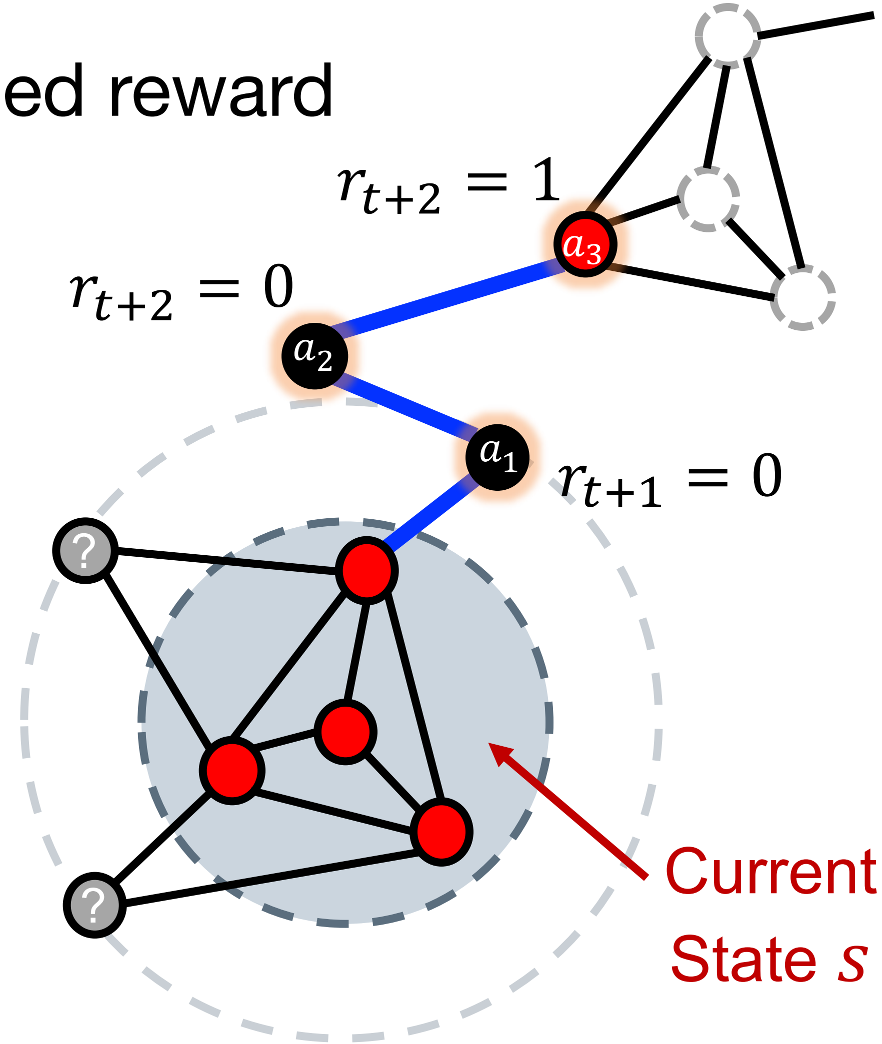
Modeling future reward: Return function

- **Return function** R_t: Cumulative discounted reward over a trajectory of length h

$$R_t = \sum_{k+1}^{h} \underbrace{\gamma^{k-1}}_{\text{Discount Factor}} \underbrace{r_{t+k}}_{\text{Reward}}$$

- Example: $\gamma = 0.5,\ h = 3$

$$R_t = 1 \cdot \underbrace{0}_{a_1} + \frac{1}{2} \cdot \underbrace{0}_{a_2} + \frac{1}{4} \cdot \underbrace{1}_{a_3} = \frac{1}{4}$$



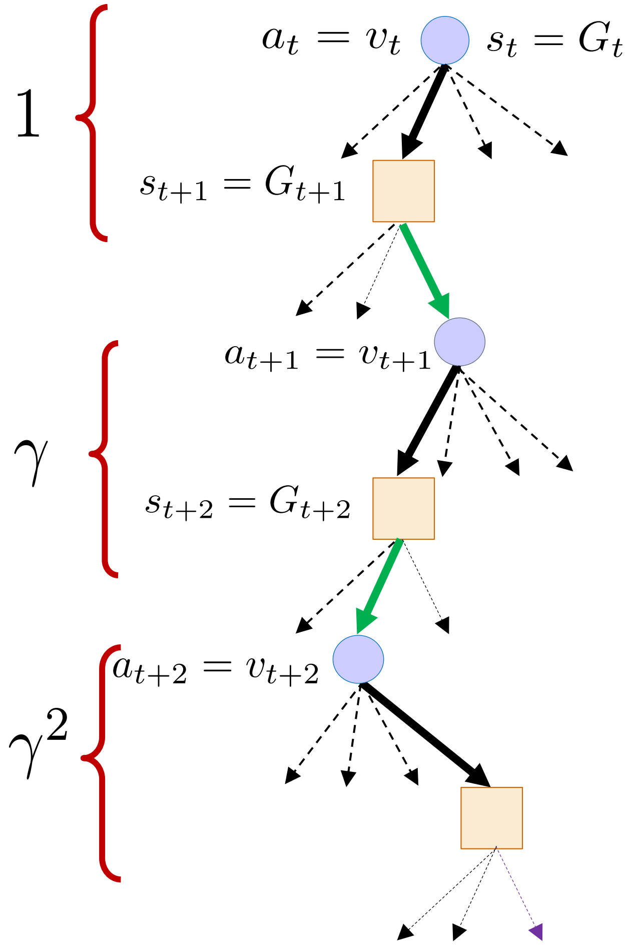
Value function

- **Value function** $V^{\pi}(s_t)$ of a state s_t is the expected discounted sum of future rewards, starting at s_t and following policy π and using discount factor $\gamma \in [0,1]$:

$$V^{\pi}(s_t) = \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s_t]$$

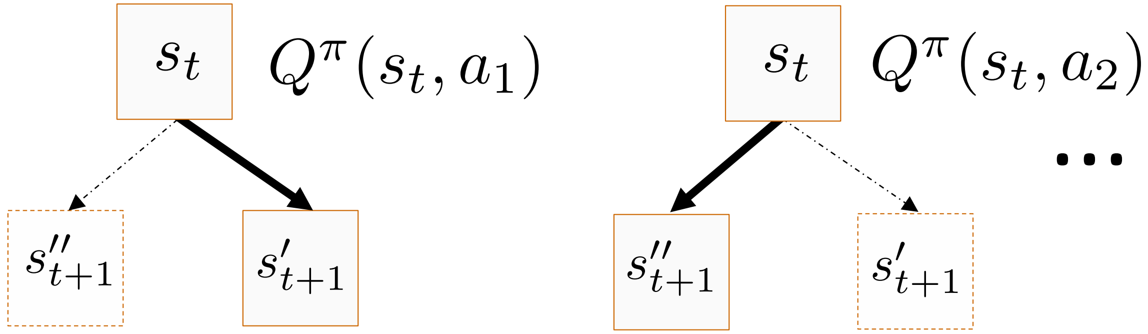
- The **goal is to maximize the value function**

$$\pi^* = \arg\max_{\pi} V^{\pi}(s_t)$$



Q function

- The value function assumes access to a transition function (with knowledge of the system dynamics)
- We optimize by decomposing the state value into action specific values:



$$V^\pi(s) = \sum_a \pi(a, s) Q^\pi(s, a)$$

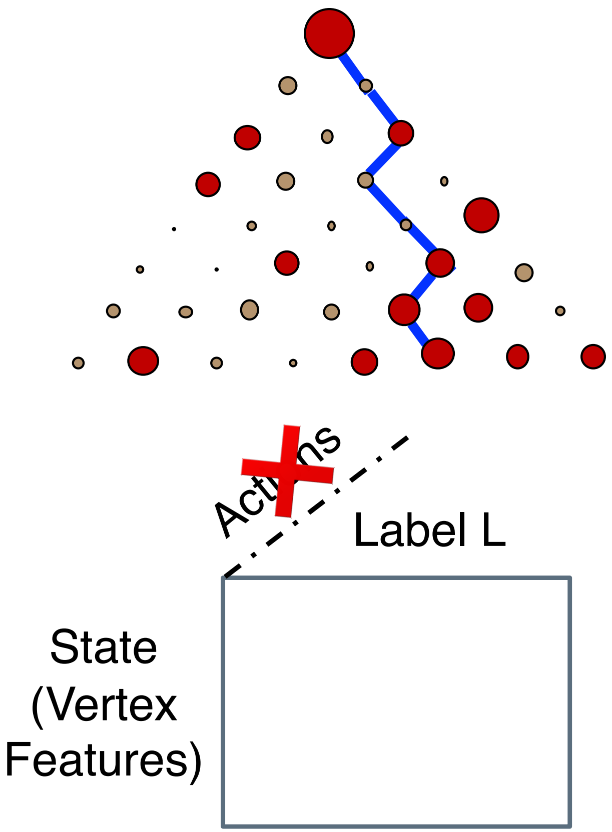
$\pi(s, a) = P(a|s)$

- **Action-Value (Q) Function**

$$Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s, a]$$

What are current approaches missing?

- **Online methods** explore one possible path over the space of potential graphs
  - Insufficient training samples to capture the complexity of relevant graph features
  - Risk missing the appropriate structure given the sparse signal
  - Can not rely on models or access to history, so they are too suspectable to:
    - Phase shifts in reward signals
    - Tunnel vision over longer explorations



Our perspective: offline + online learning

- Can we learn useful strategies by practicing offline on various tasks and topologies?

State space representation

- Network state spaces are combinatorically large and require strategies for efficient exploration
- Assumption: Some states are more useful in helping us estimate the value/policy functions
- Questions
  1. Can graph embedding(s) of the state space reduce its complexity?
  2. What kind of embedding(s) are better for planning over graphs for the given task?

Map network states into canonical representations

1. **Embed** the graph representing s_t using *embed*(G_t)
2. **Re-order** rows of G_t's adjacency matrix based on embedding distance to node(s) with label of interest
   - Closer nodes get ranked higher ➔ a prioritized set of boundary nodes
3. **Truncate** the reordered adjacency matrix to retain the graph induced by the top k nodes
   - Hyperparameter k defines the graph for computing potential trajectories and long-term reward

Training set generation for offline learning

- Background Models
  - Stochastic Block Model (SBM)
  - Lancichinetti–Fortunato–Radicchi (LFR)
  - Block Two-level Erdös-Renyi (BTER)
  - …

Foreground Model

  - Erdös-Renyi (ER)
  - Barabasi-Albert (BA)
  - …

Training instance generation

1. Generate **background** instance by randomly selecting a background model & its parameters
2. Generate **foreground** (target) instances by randomly selecting a foreground model & its parameters
3. Insert several foreground instances a few hops a way in the background instance

Episodic training

- Learn by growing and discovering similar network instances, where we have access to ground truth node labels

$$\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, s_3, \ldots\}$$

- Build a training set of {X=(state, action), Y=reward} tuples

$$X = \{\langle s_0, a_0\rangle, \langle s_1, a_1\rangle, \ldots, \langle s_L, a_L\rangle\},$$

where L = learning episode length

$$Y = \{r_0, R_1, \ldots, R_L\},$$

where $R_i = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \gamma^i r_i$

The learning objective

Learn a policy that maximizes our objective function

$$J(\theta) = \sum_{s \in S} p_{\pi_\theta}(s) \sum_{a \in A} \pi_\theta(s,a) \underbrace{\left(Q_\phi(s,a) - \sum_{a \in A} Q_\phi(s,a) \right)}_{\text{Exploitation}} + \underbrace{cH\big(s, \pi_\theta(s,a)\big)}_{\text{Exploration}}$$

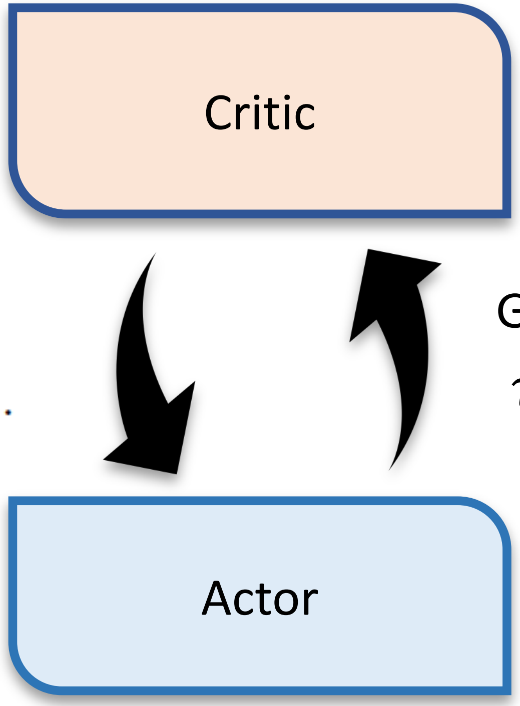
Exploitation
Estimated advantage of action *a* from state *s* given policy π_θ

Exploration

Optimization algorithm: we use a gradient-based proximal policy method (PPO)

Our model: Network Actor Critic (NAC)

$$L(\phi) = \left|\left|y_t - Q_\phi(x_t)\right|\right|_2^2$$



Estimated advantage:

$$Q_\phi(s_t, a_t) - \sum_{a \in \mathcal{A}} Q_\phi(s_t, a).$$

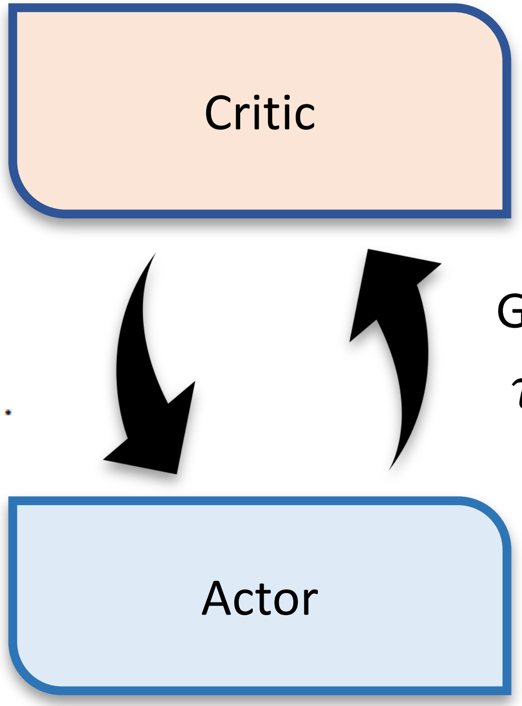
Generate samples:

$$\tau = \{s_0, a_0, r_0, s_1, \ldots\}$$

$$\hat{\theta} \leftarrow \arg\max_\theta J(\theta)$$

Our model: Network Actor Critic (NAC)

$$L(\phi) = \left|\left|y_t - Q_\phi(x_t)\right|\right|_2^2$$



Estimated advantage:

$$Q_\phi(s_t, a_t) - \sum_{a \in \mathcal{A}} Q_\phi(s_t, a).$$

Generate samples:

$$\tau = \{s_0, a_0, r_0, s_1, \dots\}$$

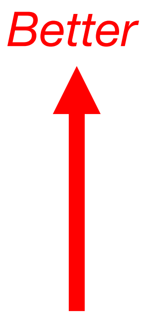
$$\hat{\theta} \leftarrow \arg\max_\theta J(\theta)$$

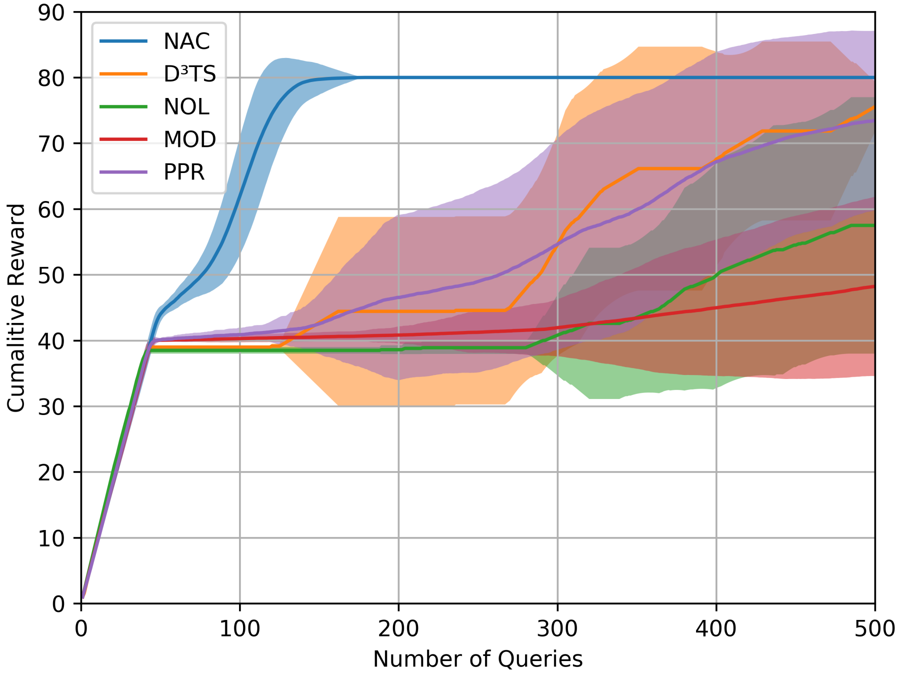
Side note: The policy and value functions are learned via CNNs with 3 convolutional layers, 64 hidden channels, and a final fully connected layer.

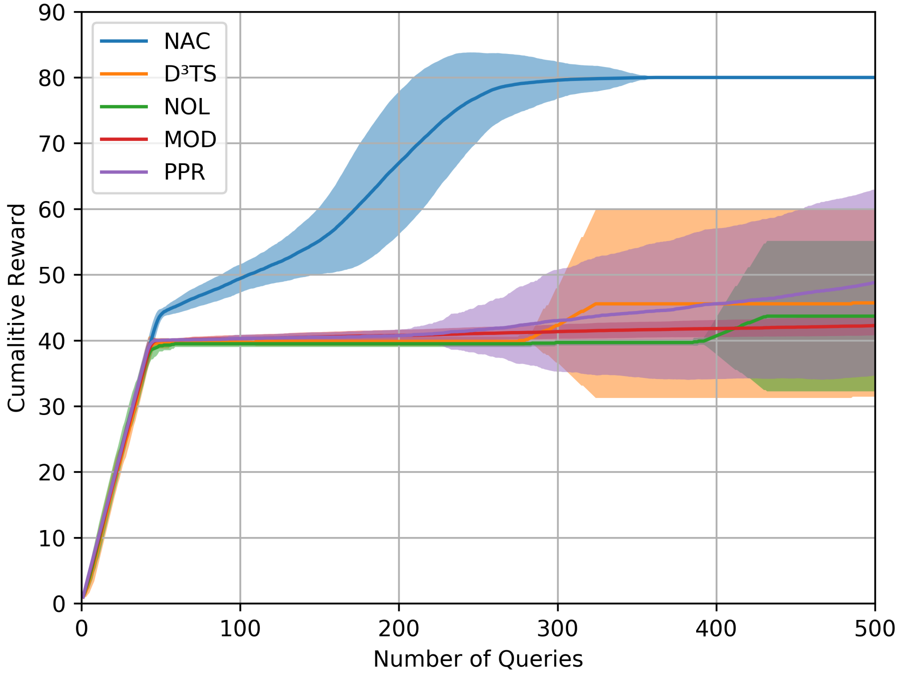
Experiments: Baselines & competitors

- Competitors:
  - Directed Diversity Dynamic Thompson Sampling (**D^{3}TS**) [Murai et al, *Data Mining and Knowledge Discovery* 2017]
    - Multi-armed bandit approach that leverages different node classifiers and Thompson sampling to diversify the selection of a boundary nodes
  - Network Online Learning (**NOL**) [LaRock et al, *Applied Network Science* 2020]
    - Learns an online regression function that maximizes discovery of previously unobserved nodes for a given number of queries
- Baselines:
  - Maximum Observed Degree (**MOD**) [Avrachenkov et al, *INFOCOM Workshops* 2014]
    - Selects the node with the highest number of observed neighbors that have the desired label
  - Personalized Page Rank (**PPR**)
    - Selects the highest scored node via PPR

Experiments: Results on synthetic data





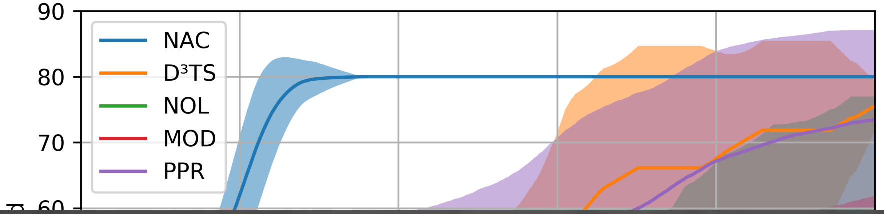


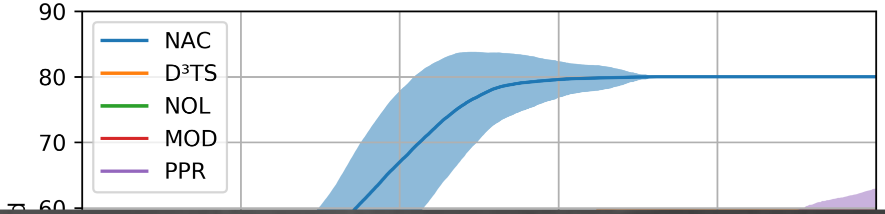
- 2 embedded **cliques** (40 vertices each)
- Cliques are on average 3 hops away from each other

- 2 embedded **dense subgraphs** (40 vertices each)
- Subgraphs are on average 3 hops away from each other

Experiments: Results on synthetic data

Better





- NAC can recover its performance even when search starts outside an anomalous graph region
- NAC gives consistent behavior across various random graph models

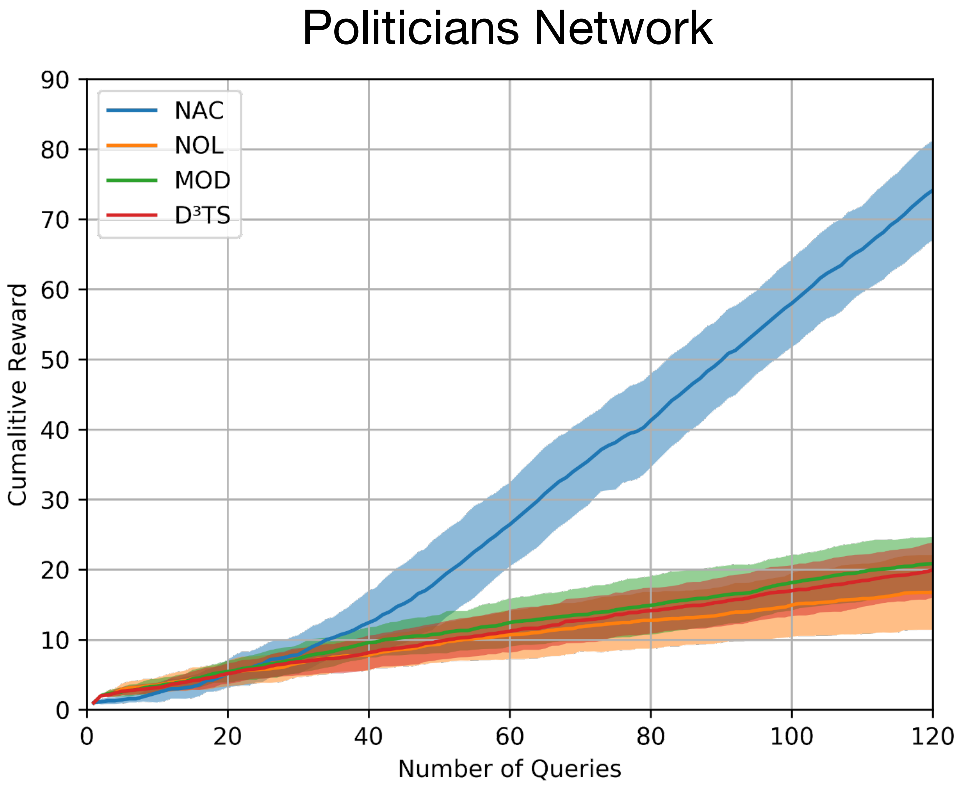
)

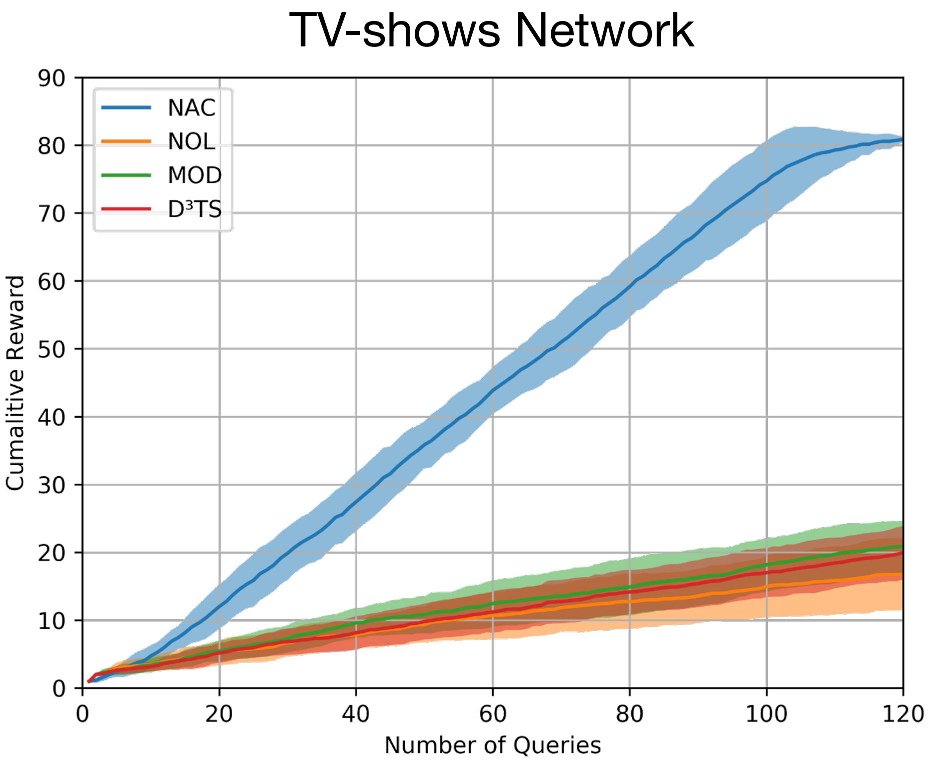
from each other

from each other

Experiments: Results on real data

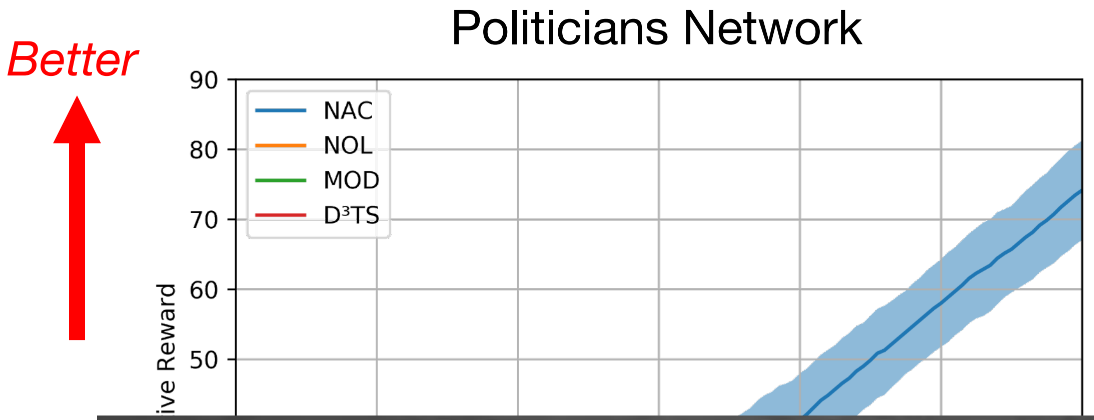
Better

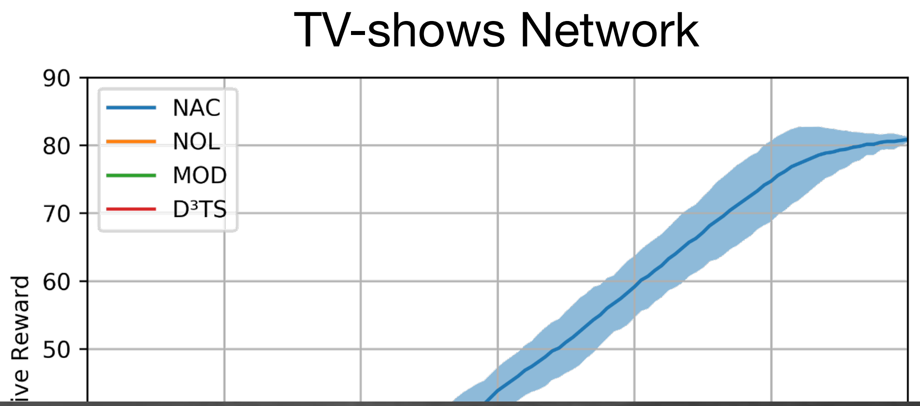




- Facebook pages dataset*: nodes are pages, edges are likes between the pages
  - Number of nodes: 4000-6000 nodes
  - Sparse networks with high clustering clustering coefficient
- Embed small subgraphs with density of 0.8

Experiments: Results on real data



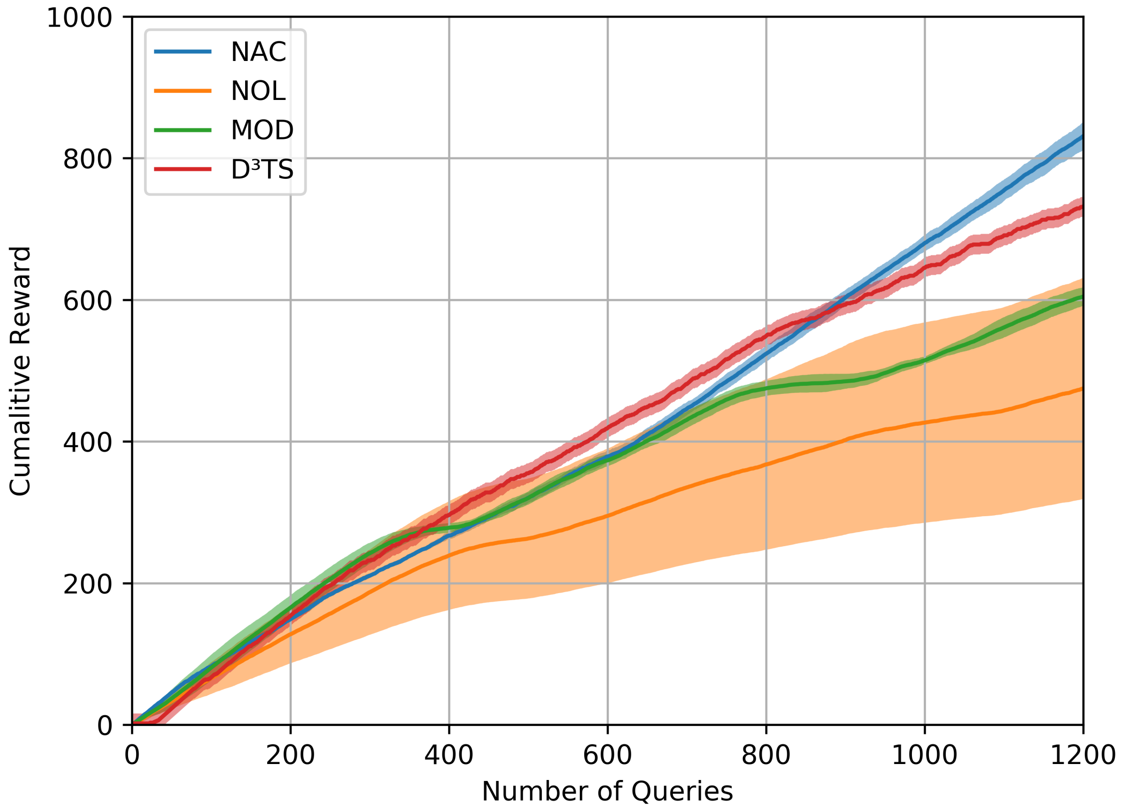


Even though NAC was trained on synthetic graphs, it generalizes to real network topologies

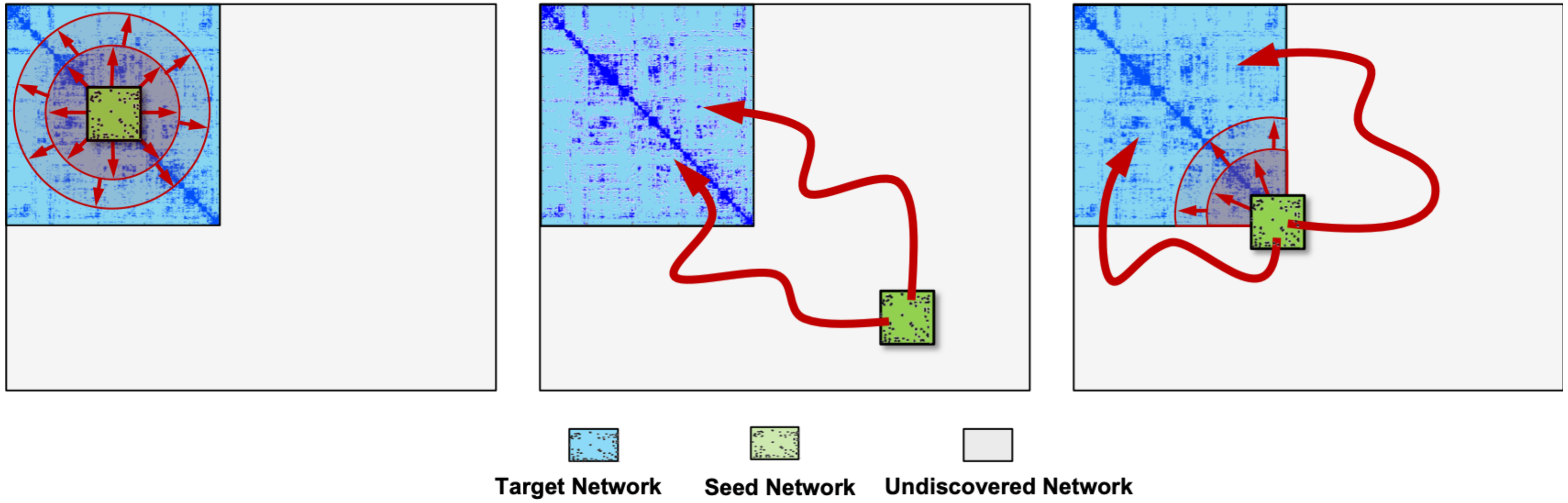
- Facebook pages dataset*: nodes are pages, edges are likes between the pages
  - Number of nodes: 4000-6000 nodes
  - Sparse networks with high clustering clustering coefficient
- Embed small subgraphs with density of 0.8

Experiments: Results on real data

- LiveJournal dataset
- Nodes are users; edges are friendship relationships; attributes are group memberships
- Target attribute is the top group (new task)
- # of nodes = 3,997,962
- # of edges = 34,681,189
- Average degree = 17.35
- Size of target group ~= 1400

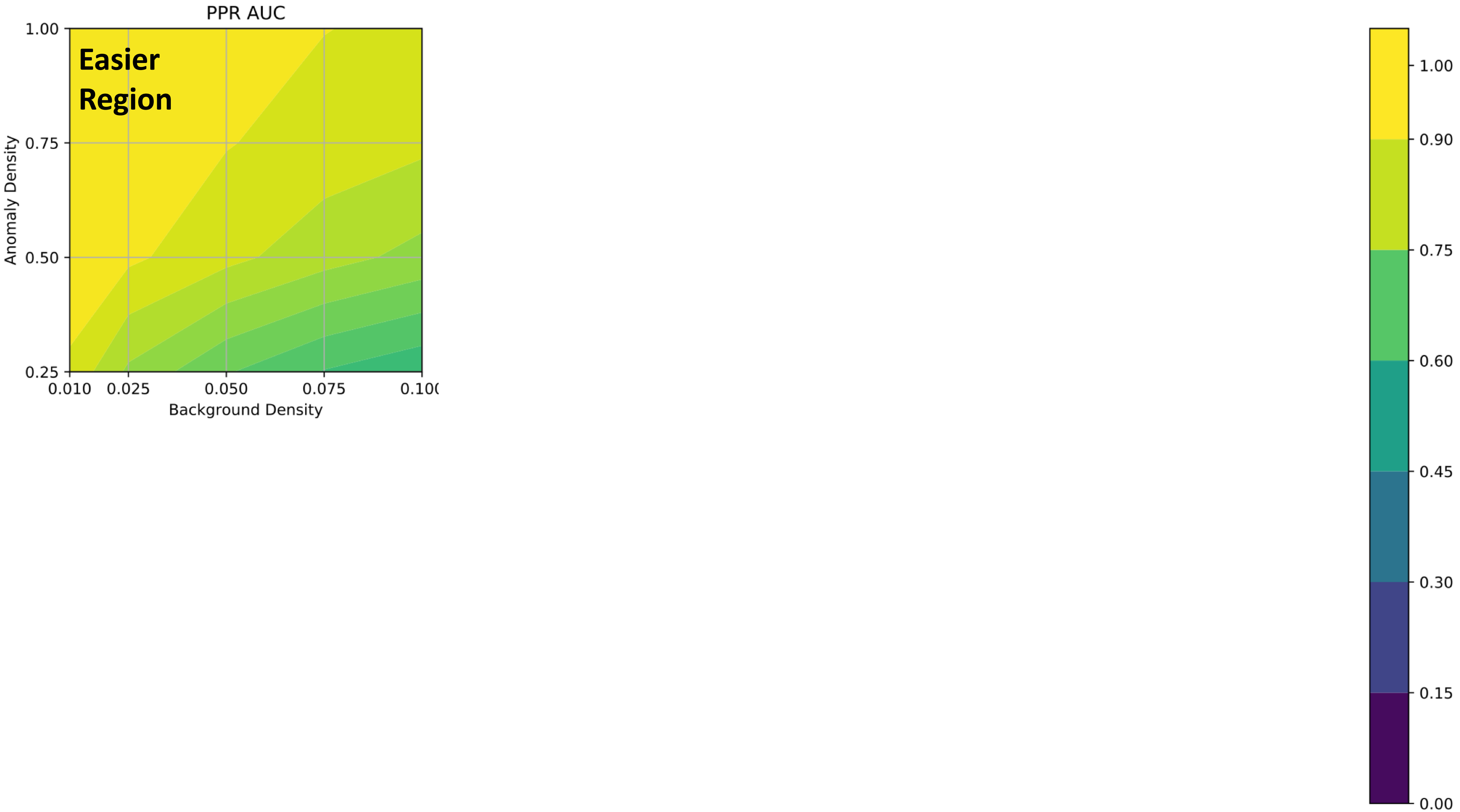


Which graph embedding to choice?

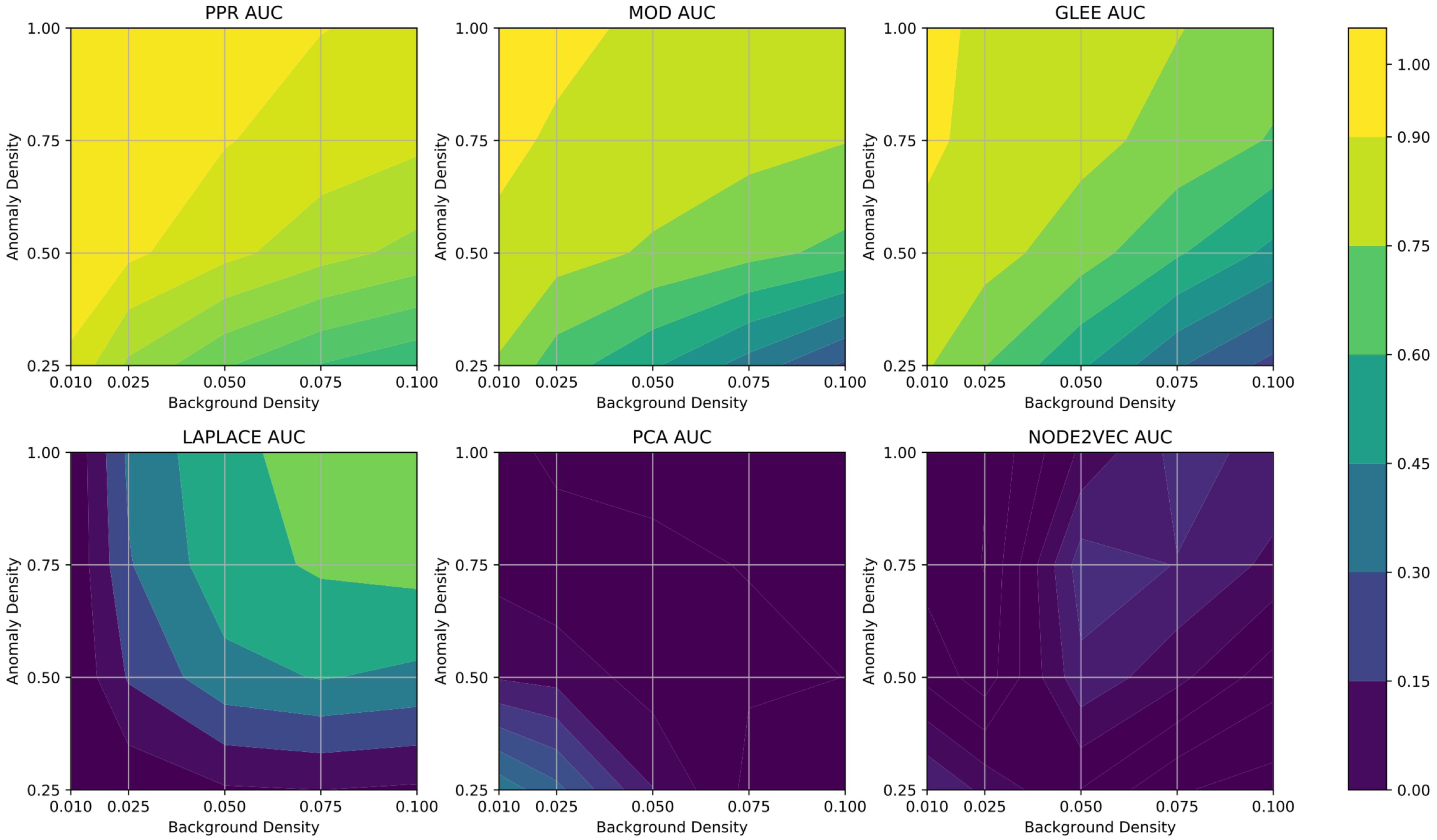


- Consistent embedding
- Compressible state space
- Robustness to increasing signal complexity
- Faster learning convergence time

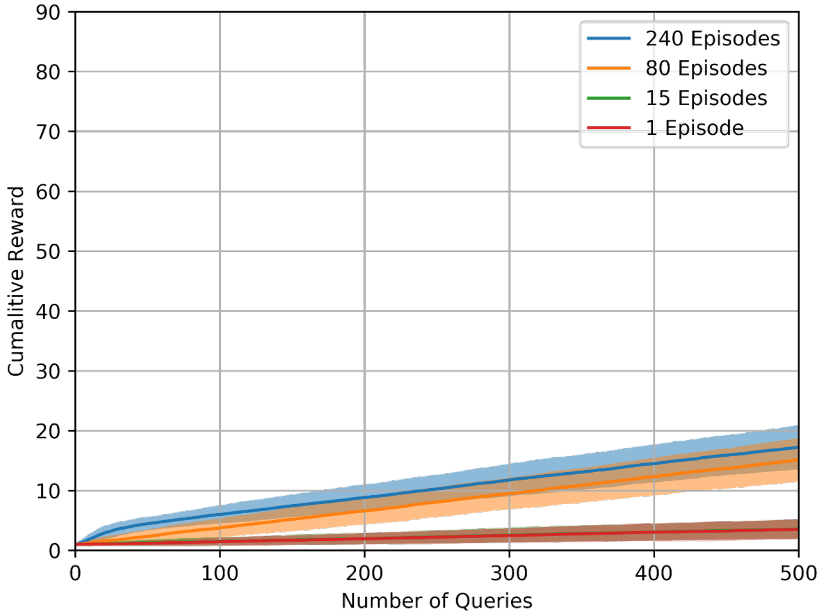
Robustness to increasing signal complexity



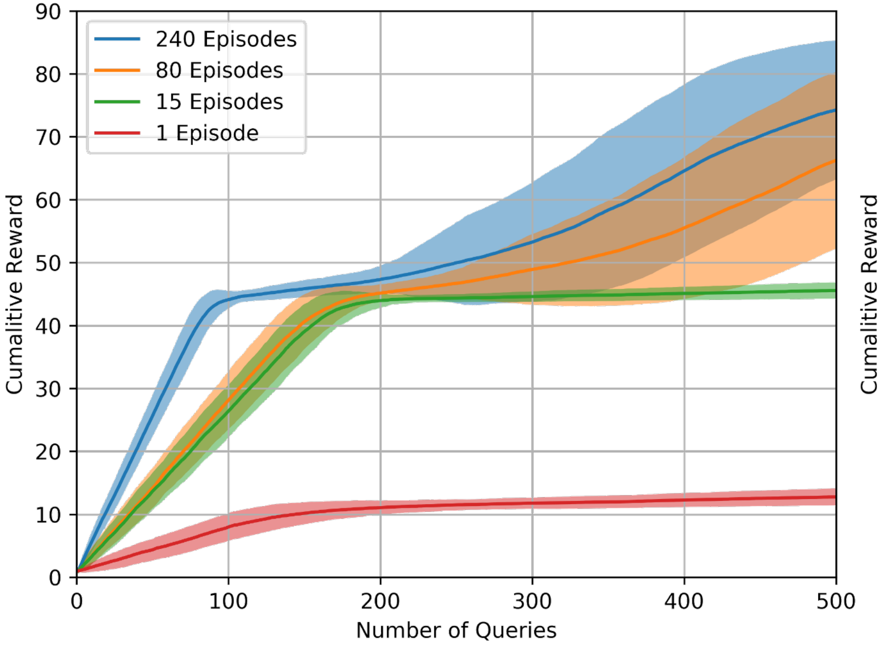
Robustness to increasing signal complexity



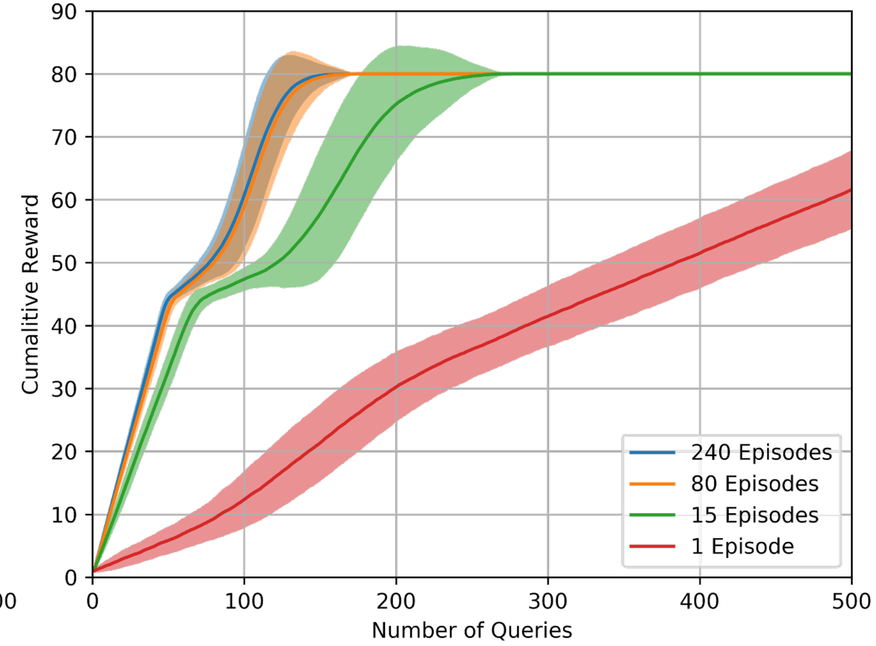
Learning convergence rates



No Embedding



Laplace Embedding



Pagerank Embedding

State approximation has a substantial impact on training time

Wrap-up: Network Actor-Critic (NAC)

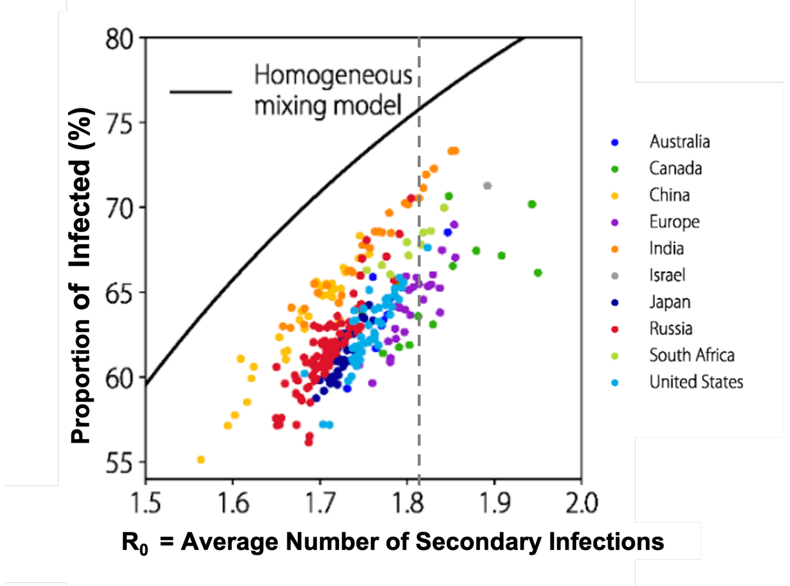
Given a query budget for identifying additional nodes and edges, how can one get a more accurate representation of the fully observed network for selective harvesting?

Method	State Space	Action Space	Observ -ability	Learning Goal	Learning Framework	Policy Training	State Embedding
NOL [9]	Large	Dynamic	Partial	Vertex Property	MDP	Online	No
D^3TS [14]	Large	Dynamic	Partial	Vertex Property	Supervised	Online	No
$GCPN$ [16]	Small	Fixed	Full	Graph Property	MDP	Offline on Given Dataset	No
NAC	Large	Dynamic	Partial	Vertex Property	MDP	Offline & Online on Designed Dataset	Yes

NOL: LaRock et al, *Applied Network Science* 2020; D^3TS: Murai et al, *Data Mining and Knowledge Discovery* 2017; GCPN: You et al, *NeurIPS* 2018.

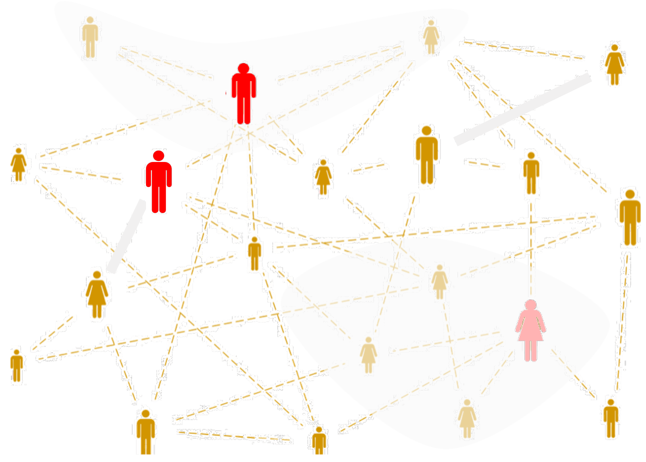
Control of pandemics

Complex Network Effects



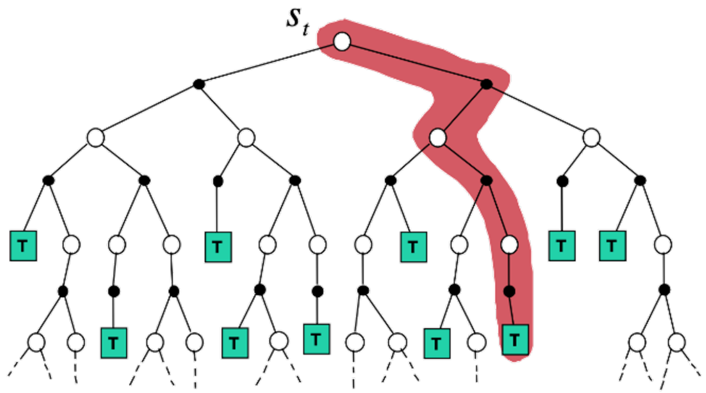
- Same initial conditions lead to diverse disease progression
- Variance in secondary infections

Partial Observability



- Asymptomatic transmission
- Privacy concerns
- Lack of sensing

Large Decision Space

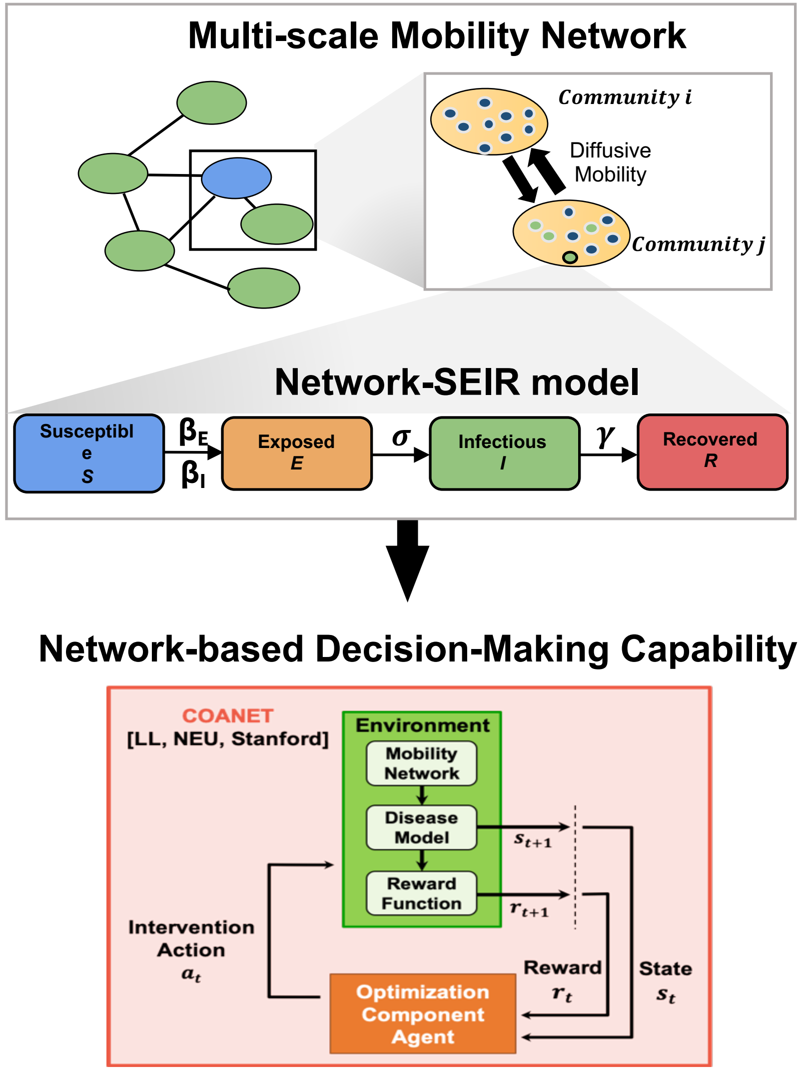


- Limited resources available
- Uncertain consequences

Left figure adapted from Mistry, D., Litvinova, M., Pastore y Piontti, A. et al. Inferring high-resolution human mixing patterns for disease modeling. *Nat Commun* 12, 323 (2021). https://doi.org/10.1038/s41467-020-20544-y

Problem and high-level overview of our system: COANET

Problem: *Given a budget on available resources and/or associated cost, obtain an optimal sequence of interventions that reduces the rate of spread*



Questions

tina@eliassi.org
NAC Preprint at https://bit.ly/3sozou8