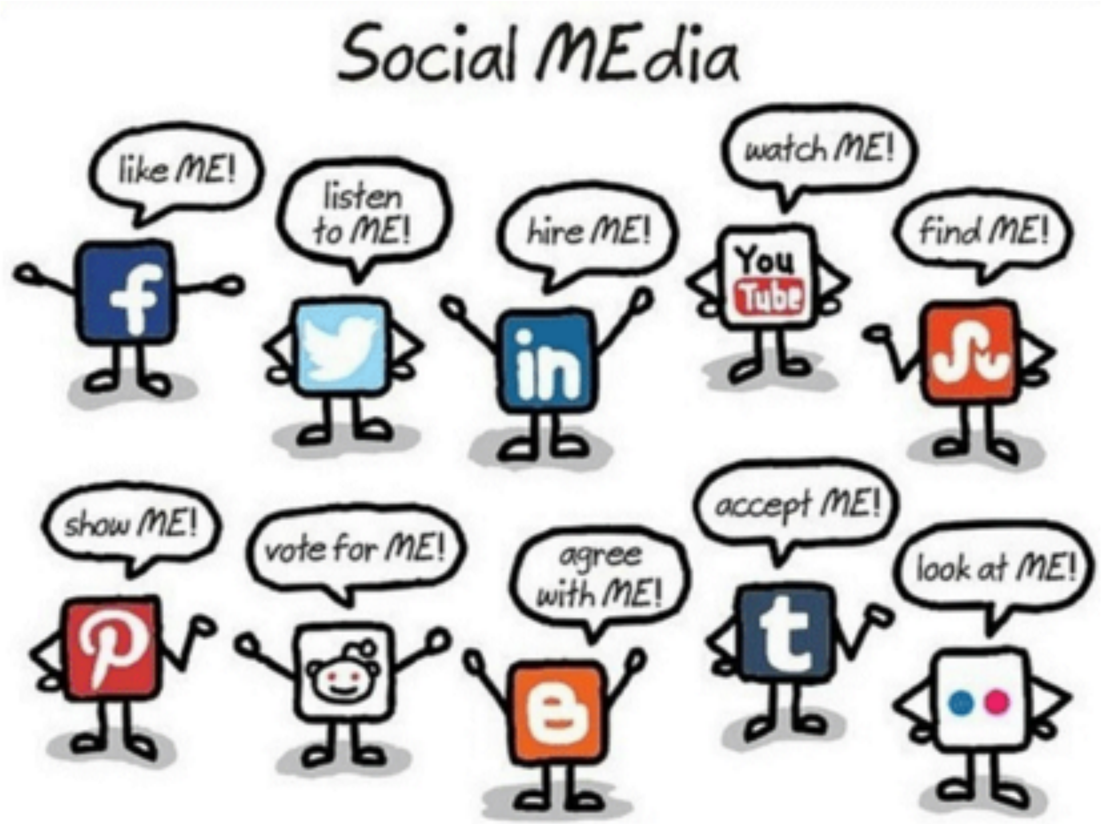
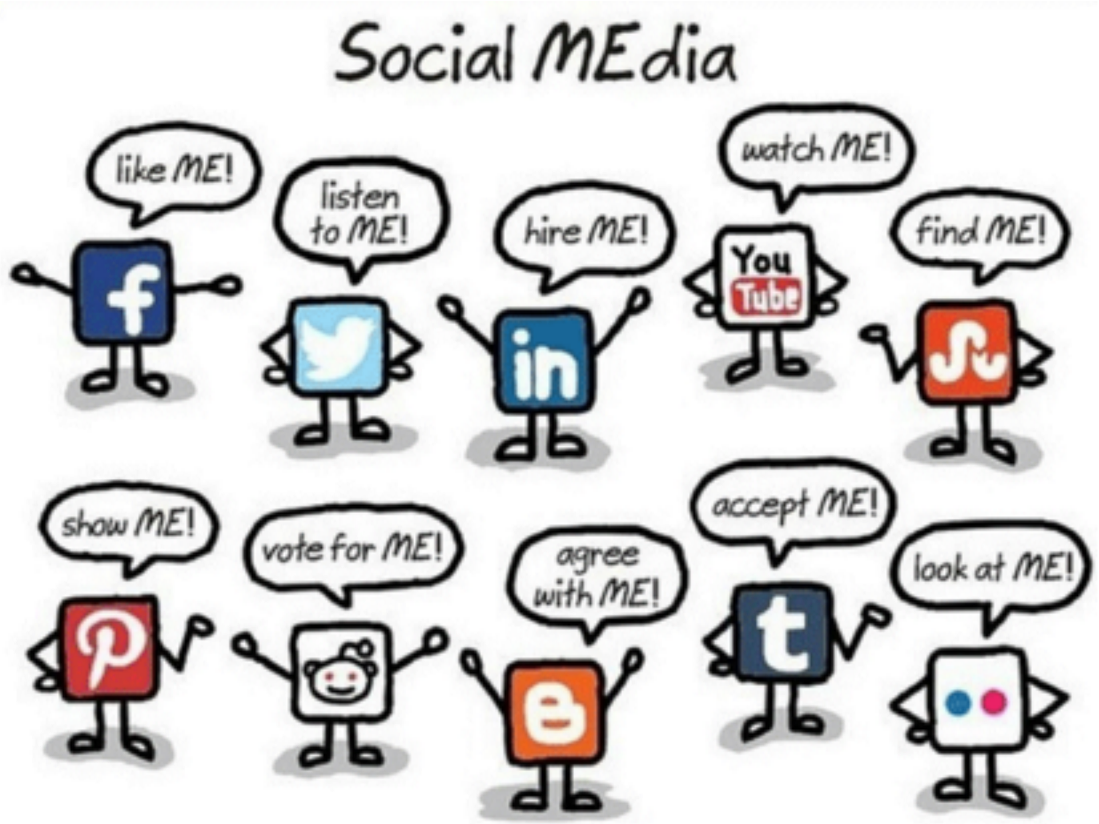
BANDITS ON GRAPHS AND STRUCTURES

Michal Valko, SequeL, Inria Lille - Nord Europe (HdR defense)

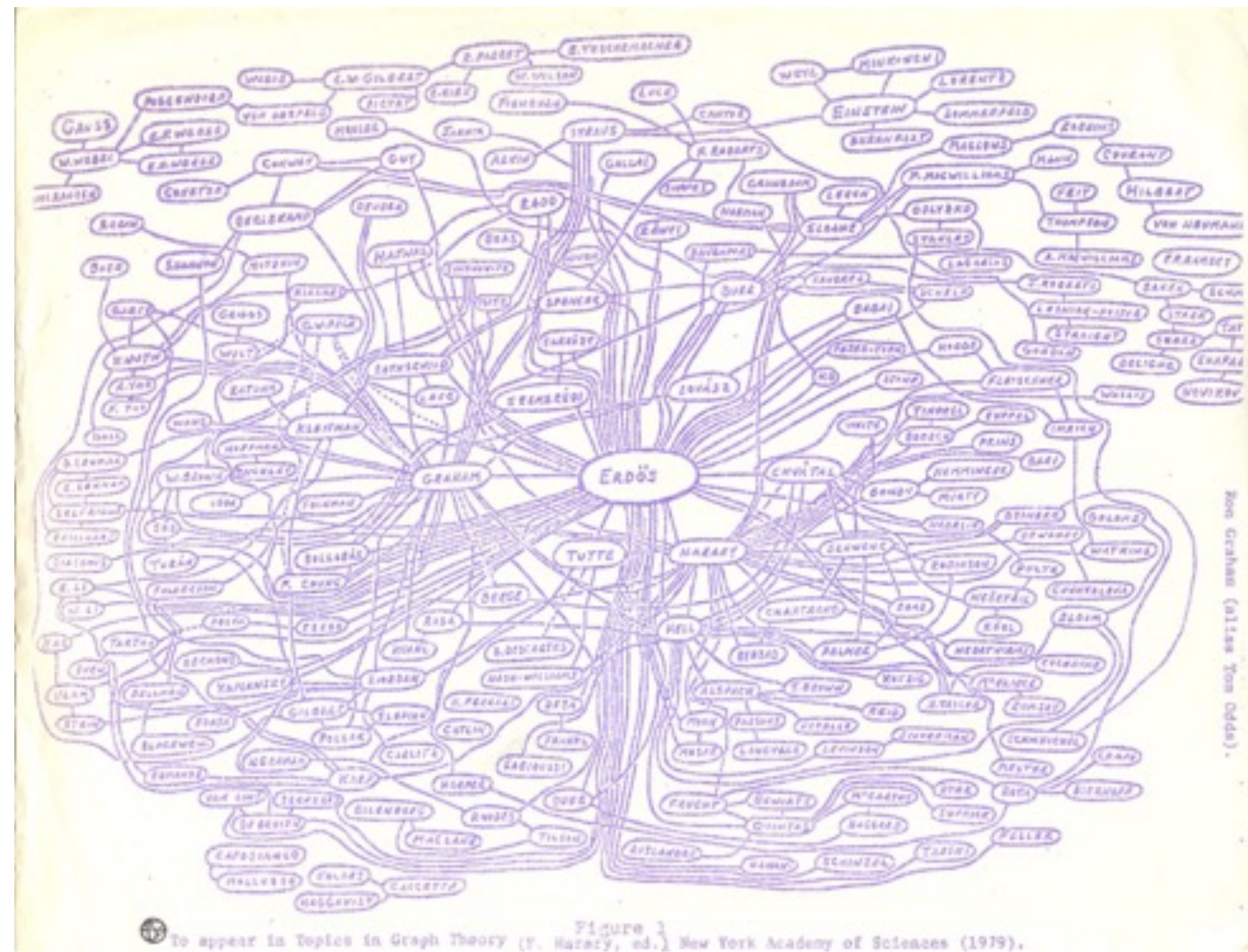
M.	Nicolas	**VAYATIS**	ENS de Cachan	Garant & Examinateur
M.	Aurélien	**GARIVIER**	Université Toulouse	Rapporteur
M.	Gábor	**LUGOSI**	Universitat Pompeu Fabra	Rapporteur
M.	Vianney	**PERCHET**	ENSAE ParisTech	Rapporteur
M.	Nicolò	**CESA-BIANCHI**	Università di Milano	Examinateur
M.	Mark	**HERBSTER**	University College London	Examinateur
M.	Rémi	**MUNOS**	DeepMind & Inria	Examinateur



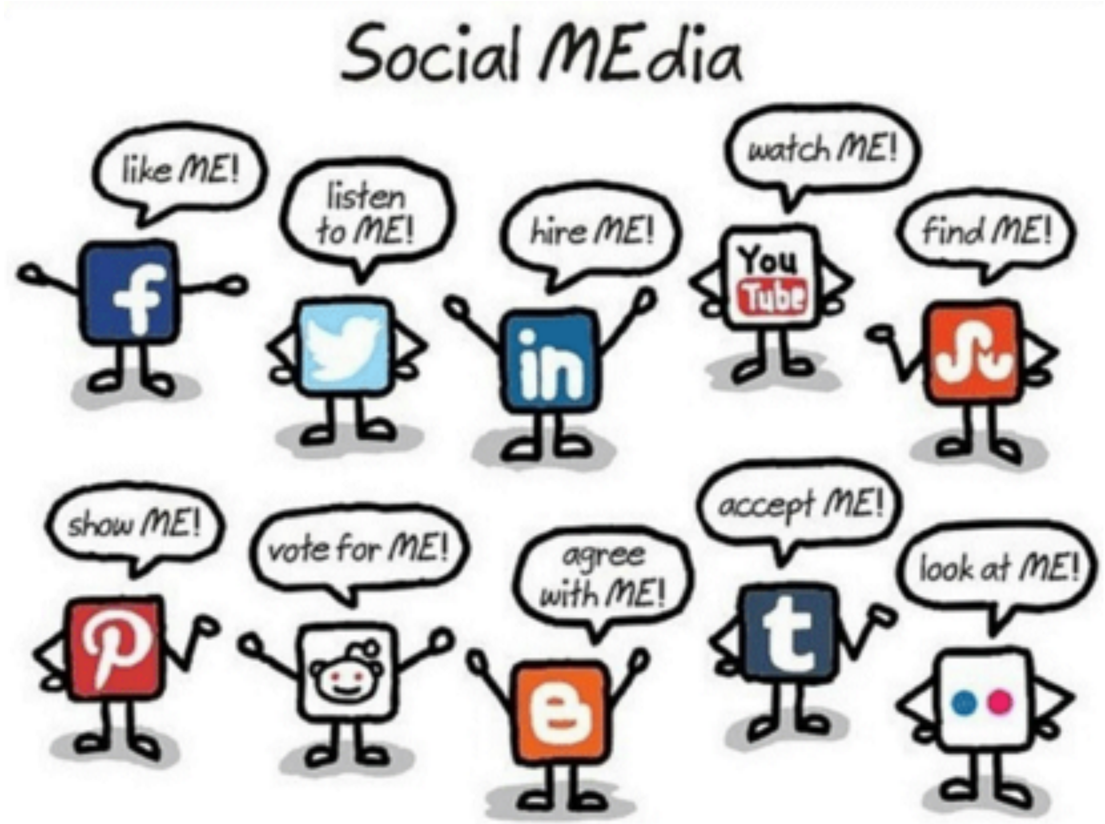
online social networks



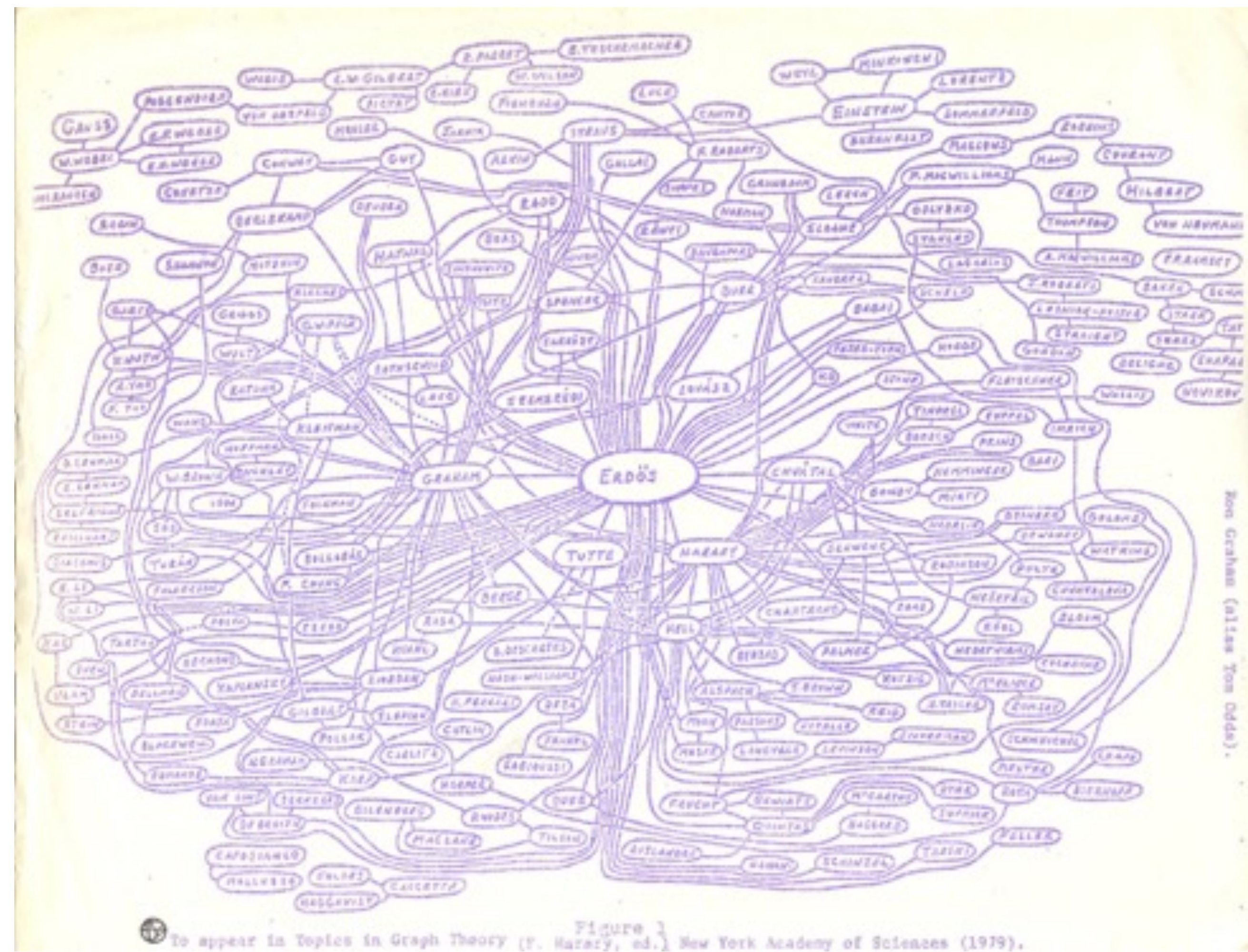
online social networks



Erdös number project



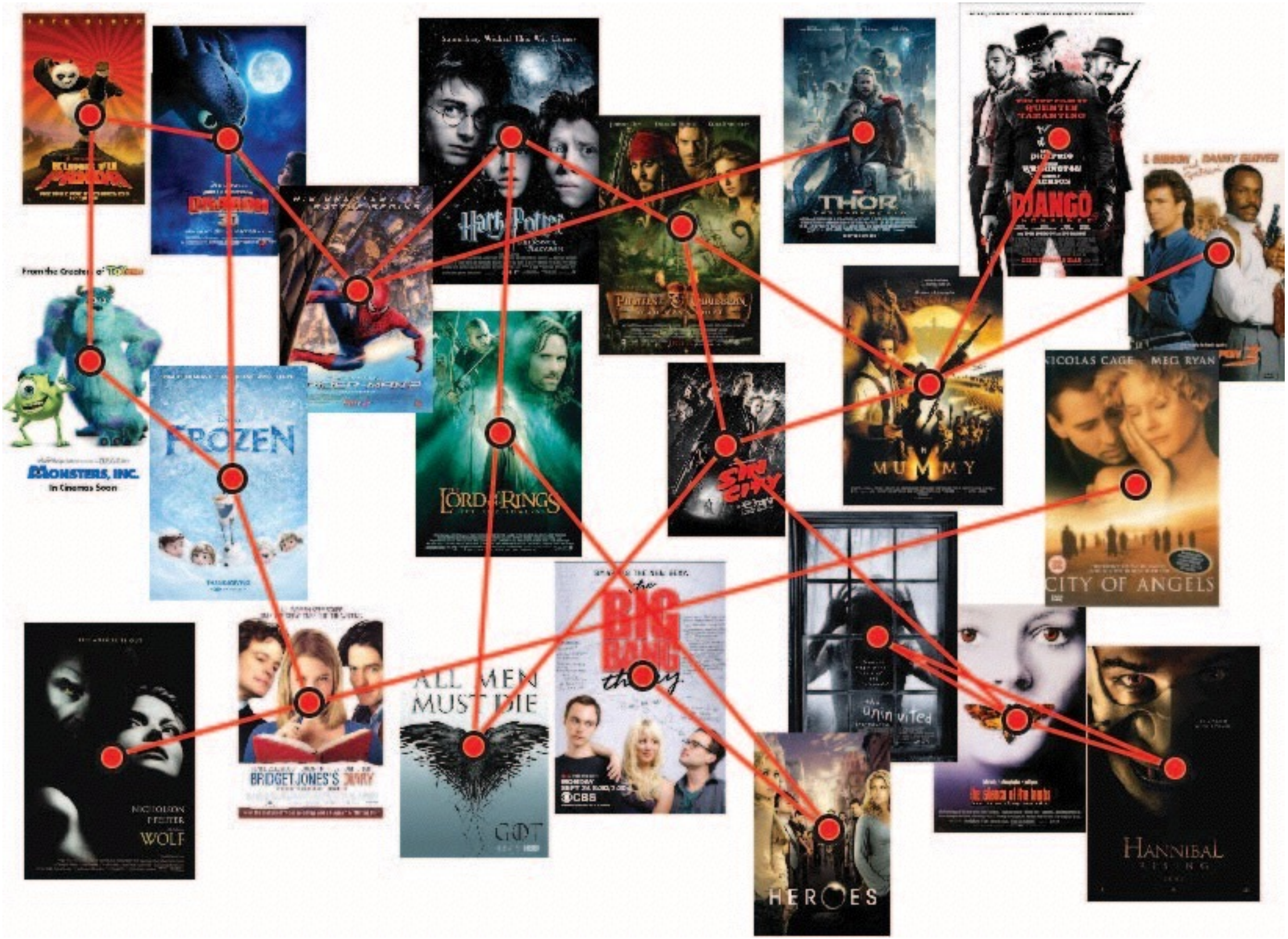
online social networks

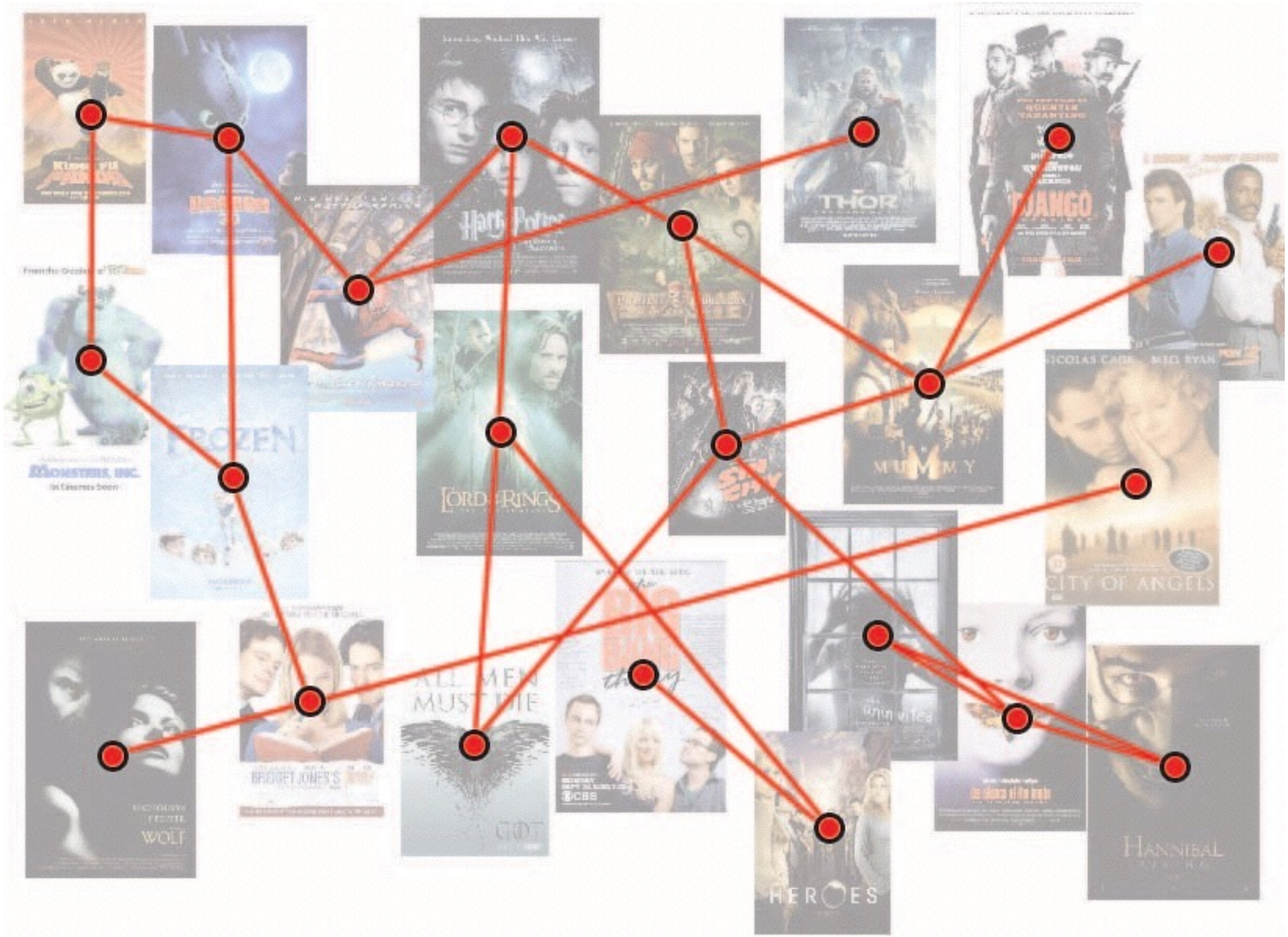


Erdös number project

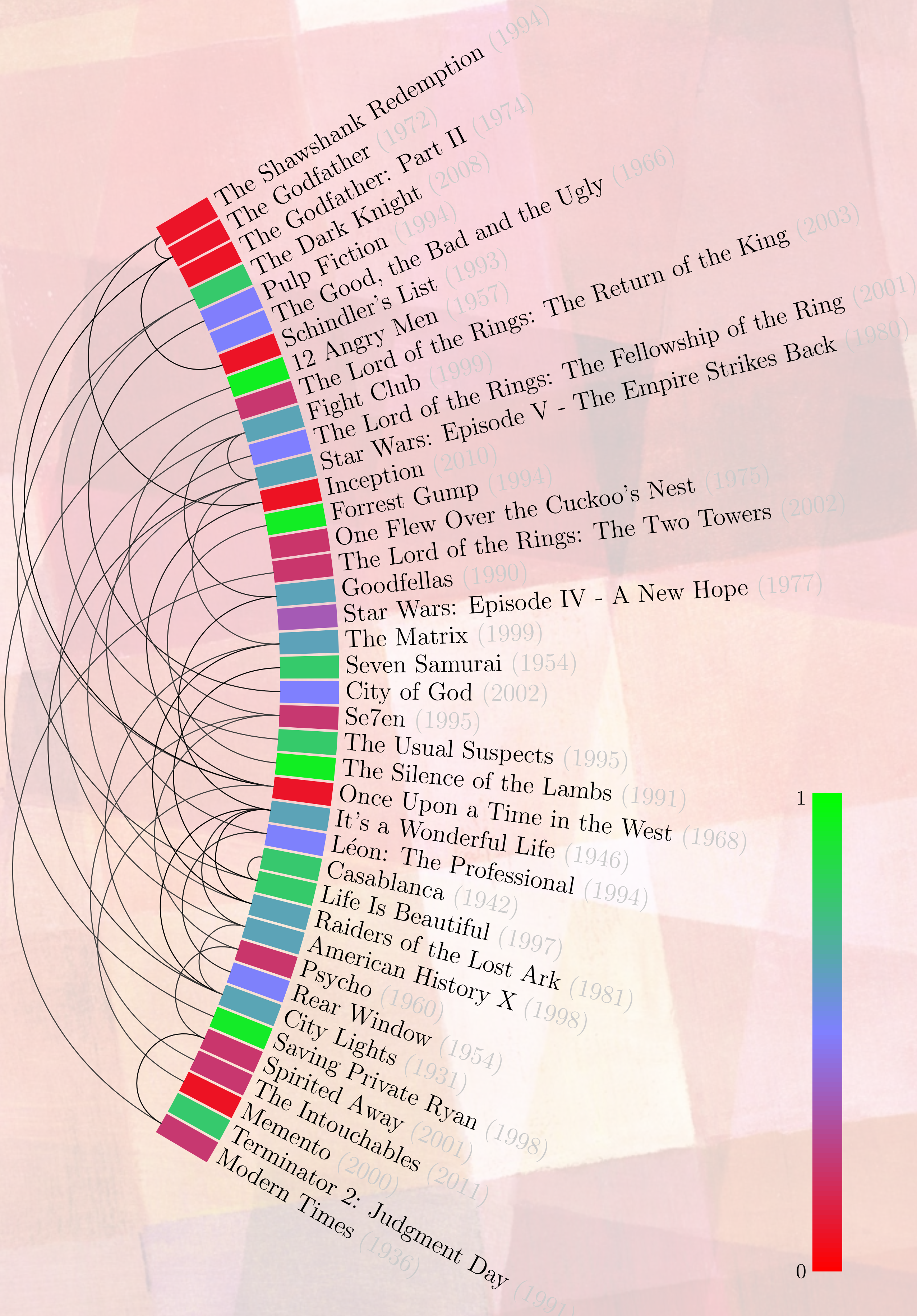
Berkeley's floating sensor network


KUNG FU PANDA
DRAGON 3D
Harry Potter
THOR
DJANGO
MONSTERS, INC.
In Cinemas Soon
FROZEN
LORD OF THE RINGS
SIN CITY
MUMMY
NICOLAS CAGE MEG RYAN
CITY OF ANGELS
NICHOLSON
PFEIFFER
WOLF
BRIDGET JONES'S DIARY
ALL MEN
MUST DIE
GOT
the BIG BANG theory
CBS
the uninvited
HEROES
HANNIBAL
RISING




THOR
DJANGO
MONSTERS, INC.
FROZEN
NICOLAS CAGE
MEG RYAN
CITY OF ANGELS
HEROES
HANNIBAL
WOLF

Example of a graph bandit problem



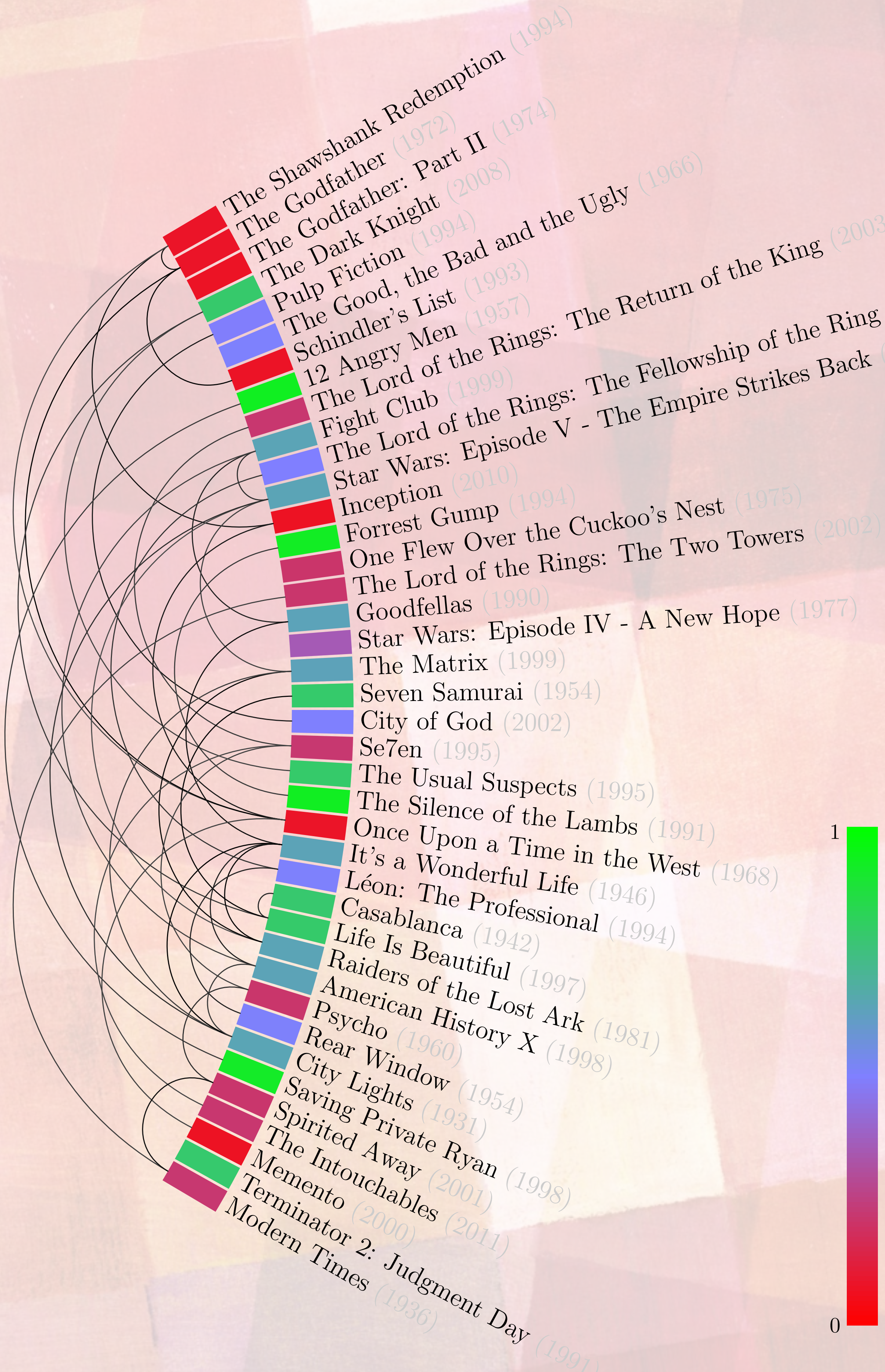
Example of a graph bandit problem

movie recommendation

Example of a graph bandit problem

movie recommendation

- recommend movies to a **single** user



Example of a graph bandit problem

movie recommendation

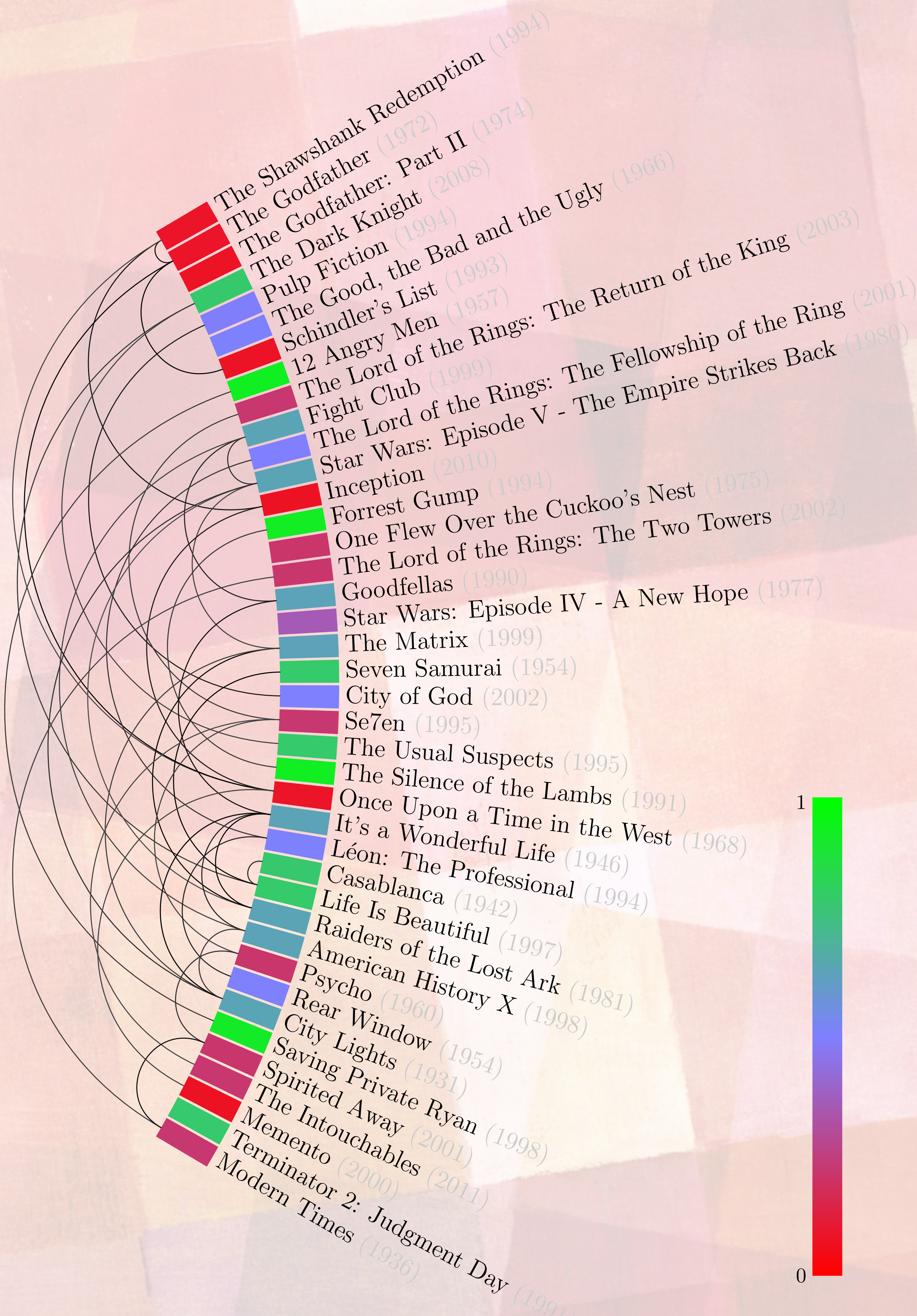
- recommend movies to a
- **goal:** maximise the sum of the ratings (minimise regret)

Example of a graph bandit problem

movie recommendation

- recommend movies to a
- **goal:**
  (minimise regret)
- good prediction after just a few steps

$$T \ll N$$



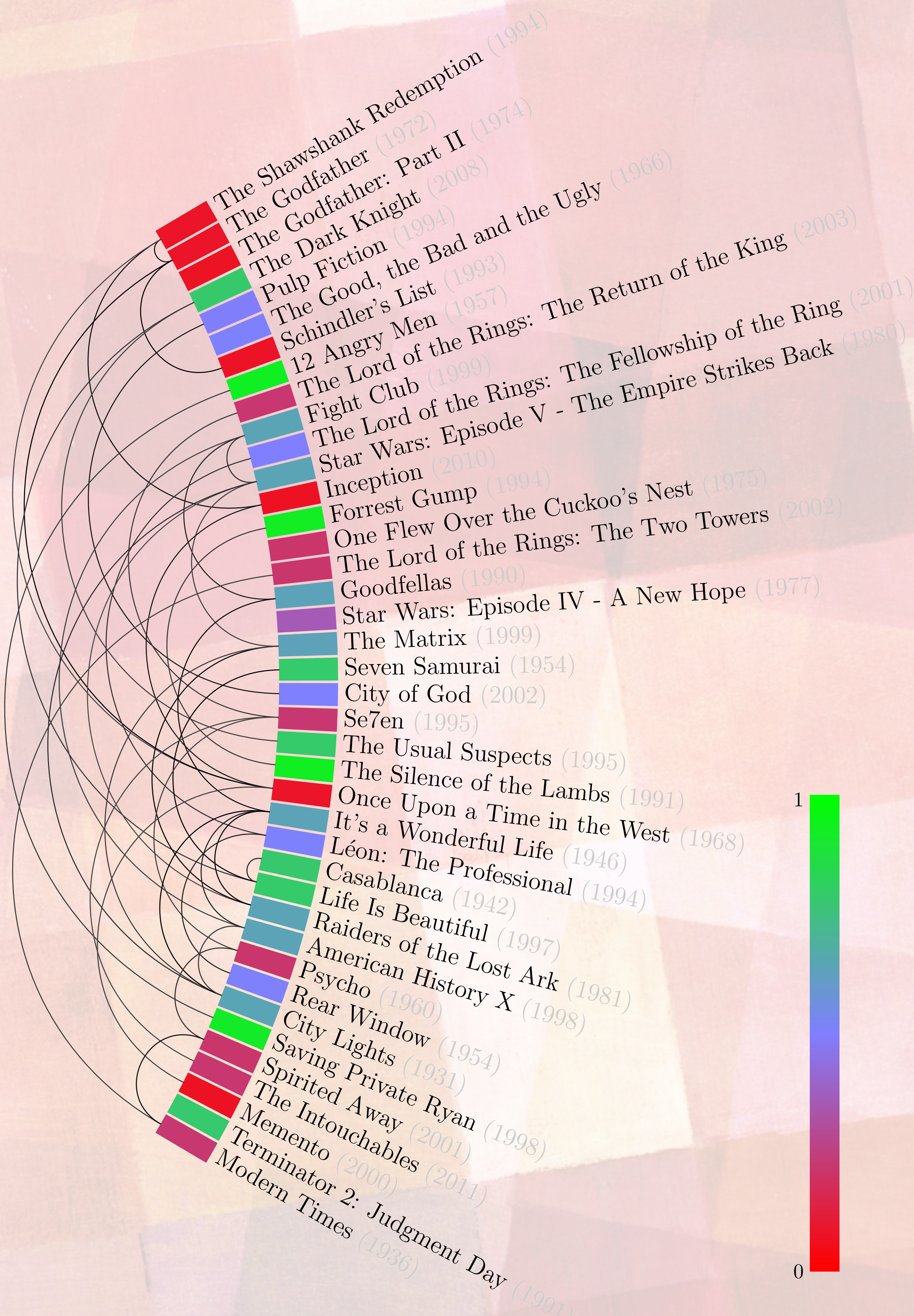
Example of a graph bandit problem

movie recommendation

- recommend movies to a
- goal:
  (minimise regret)
- good prediction after just a few steps

$$T \ll N$$

- extra information



Example of a graph bandit problem

movie recommendation

- recommend movies to a
- **goal:**
  (minimise regret)
- good prediction after just a few steps

$$T \ll N$$

- extra information
  - ratings are **smooth** on a graph

Example of a graph bandit problem

movie recommendation

- recommend movies to a
- **goal:**
  (minimise regret)
- good prediction after just a few steps

$$T \ll N$$

- extra information
  - ratings are
- main question: can we learn **faster**?

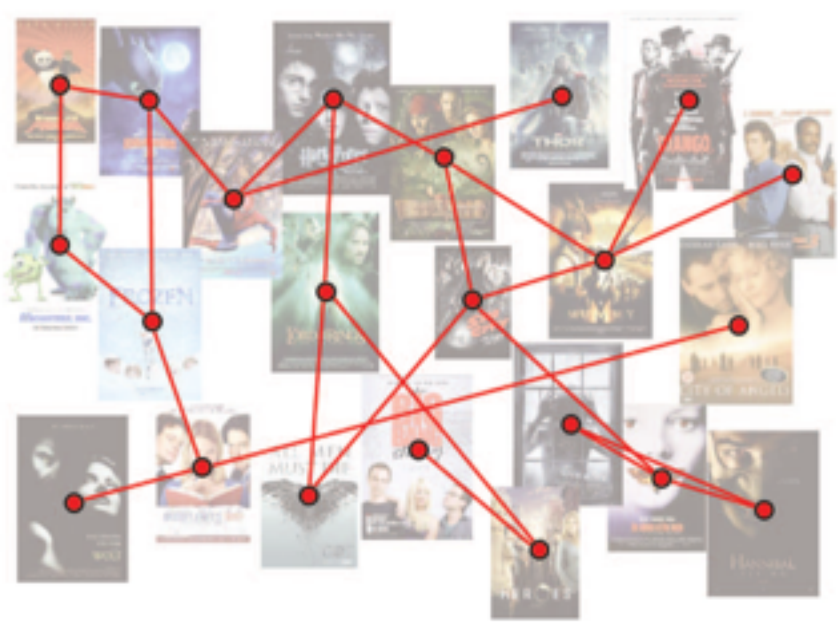
GETTING REAL

GETTING REAL

Let's be lazy and ignore the structure

GETTING REAL

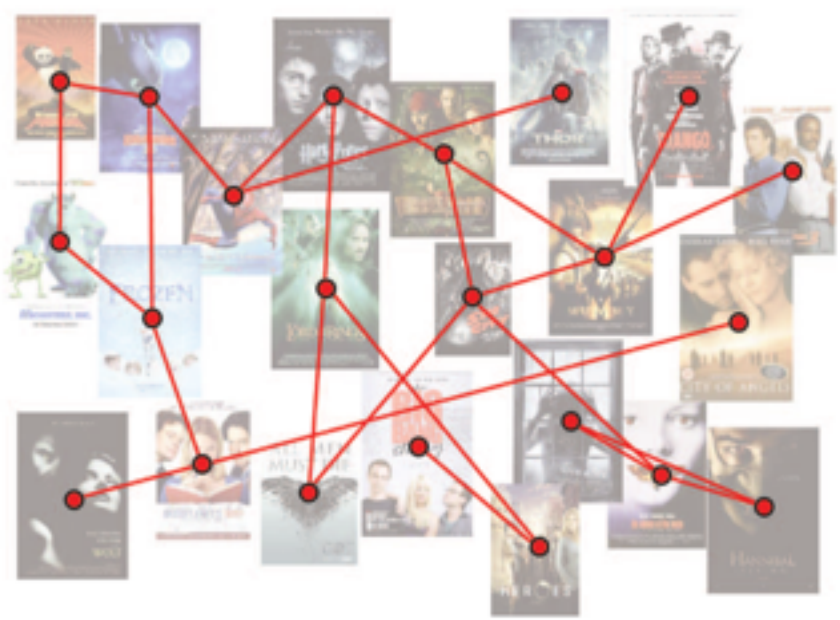
Let's be lazy and ignore the structure



→

GETTING REAL

Let’s be lazy and ignore the structure

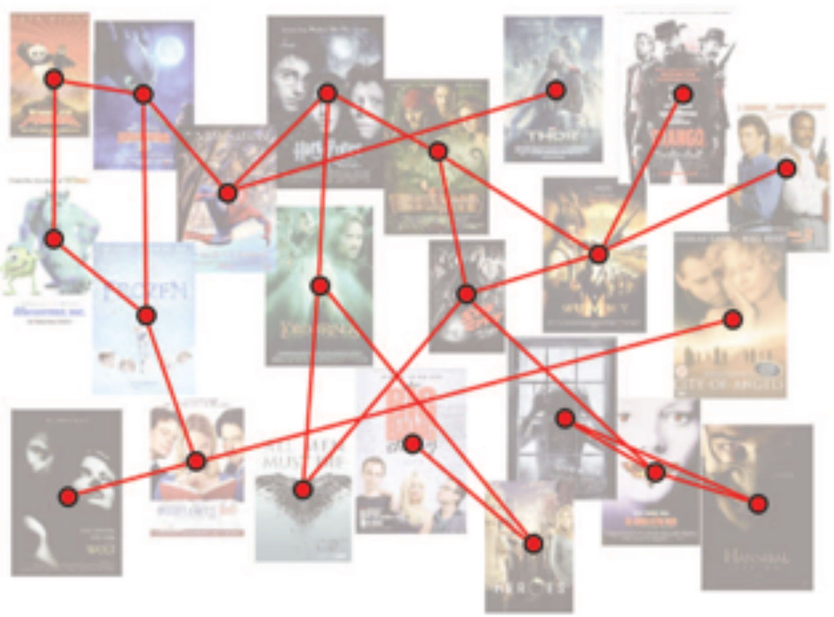


→

Multi-armed bandit problem!

GETTING REAL

Let's be lazy and ignore the structure



→

Multi-armed bandit problem!

Worst case regret (to the best fixed strategy)

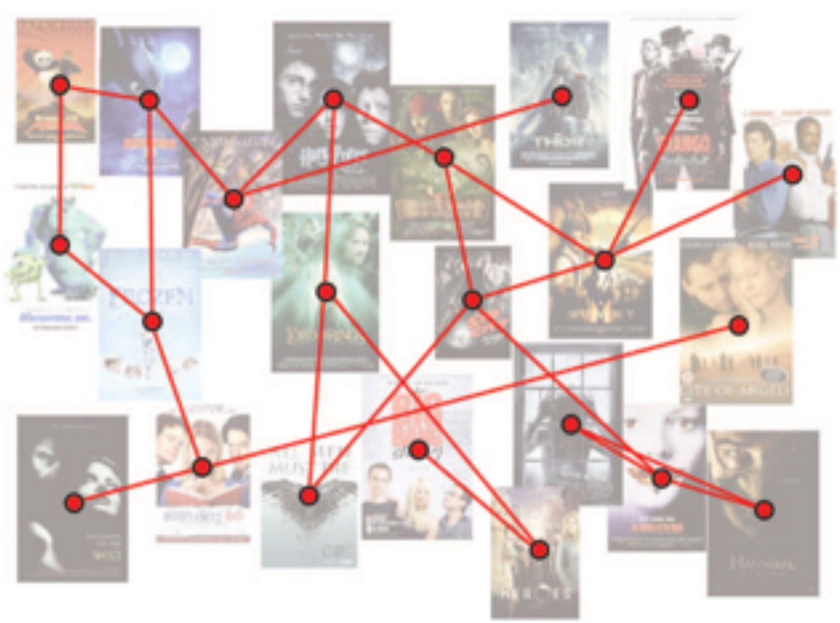
$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions (N)

#rounds (T)

GETTING REAL

Let's be lazy and ignore the structure



→

Multi-armed bandit problem!

Worst case regret (to the best fixed strategy)

Matching lower bound (Auer, Cesa-Bianchi, Freund, Schapire 2002)

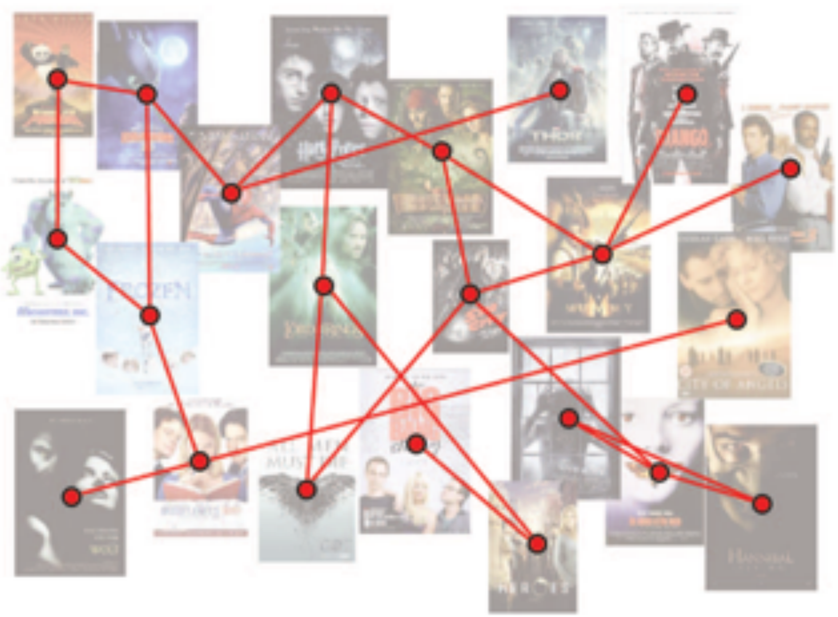
$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions

#rounds

GETTING REAL

Let's be lazy and ignore the structure



→

Multi-armed bandit problem!

Worst case regret (to the best fixed strategy)

$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions

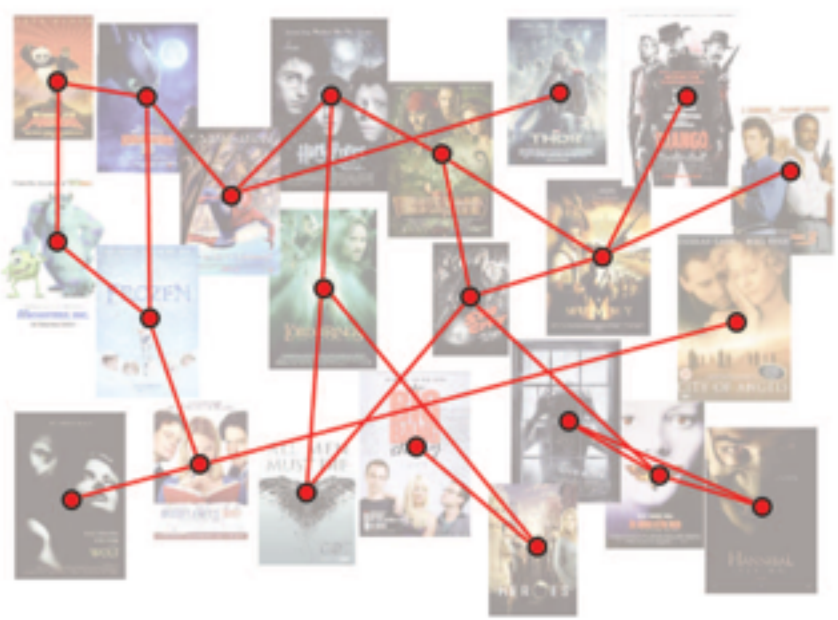
#rounds

Matching lower bound (Auer, Cesa-Bianchi, Freund, Schapire 2002)

How big is N? Number of movies on **http://www.imdb.com/stats**: 3,792,257

GETTING REAL

Let's be lazy and ignore the structure



→

Multi-armed bandit problem!

Worst case regret (to the best fixed strategy)

$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions

#rounds

Matching lower bound (Auer, Cesa-Bianchi, Freund, Schapire 2002)

How big is N? Number of movies on **http://www.imdb.com/stats**: 3,792,257

Problem: Too many actions!

LEARNING FASTER

LEARNING FASTER

$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions

#rounds

LEARNING FASTER

$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions (N)

#rounds (T)

- Arm independence is too strong and unnecessary

LEARNING FASTER

$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions

#rounds

- Arm independence is too strong and unnecessary
- Replace **N** with something much smaller

LEARNING FASTER

$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions

#rounds

- Arm independence is too strong and unnecessary
- Replace **N** with something much smaller
  - problem/instance/data dependent

LEARNING FASTER

#actions

$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#rounds

- Arm independence is too strong and unnecessary
- Replace **N** with something much smaller
  - problem/instance/data dependent
  - example: linear bandits **N** to **D**

#dimensions

LEARNING FASTER

$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions

#rounds

- Arm independence is too strong and unnecessary
- Replace **N** with something much smaller
  - problem/instance/data dependent
  - example: linear bandits **N** to **D**

#dimensions

- **In this talk: Graph** Bandits!

LEARNING FASTER

$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions

#rounds

- Arm independence is too strong and unnecessary
- Replace **N** with something much smaller
  - problem/instance/data dependent
  - example: linear bandits **N** to **D**

#dimensions

- **In this talk: Graph** Bandits!
  - sequential problems where **actions are nodes** on a graph

LEARNING FASTER

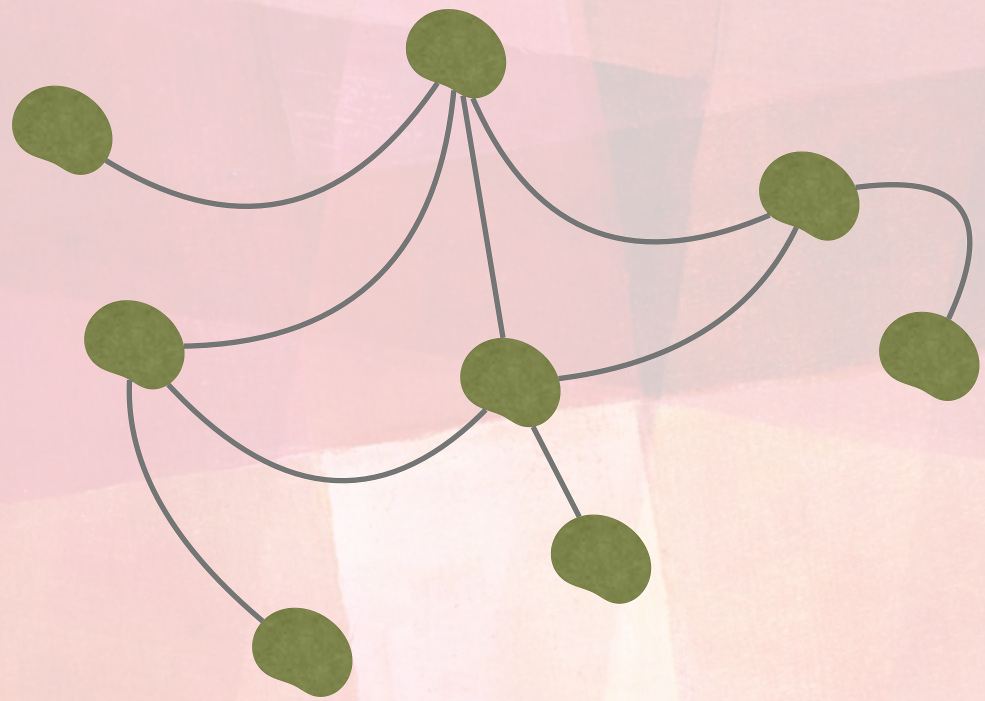
$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions

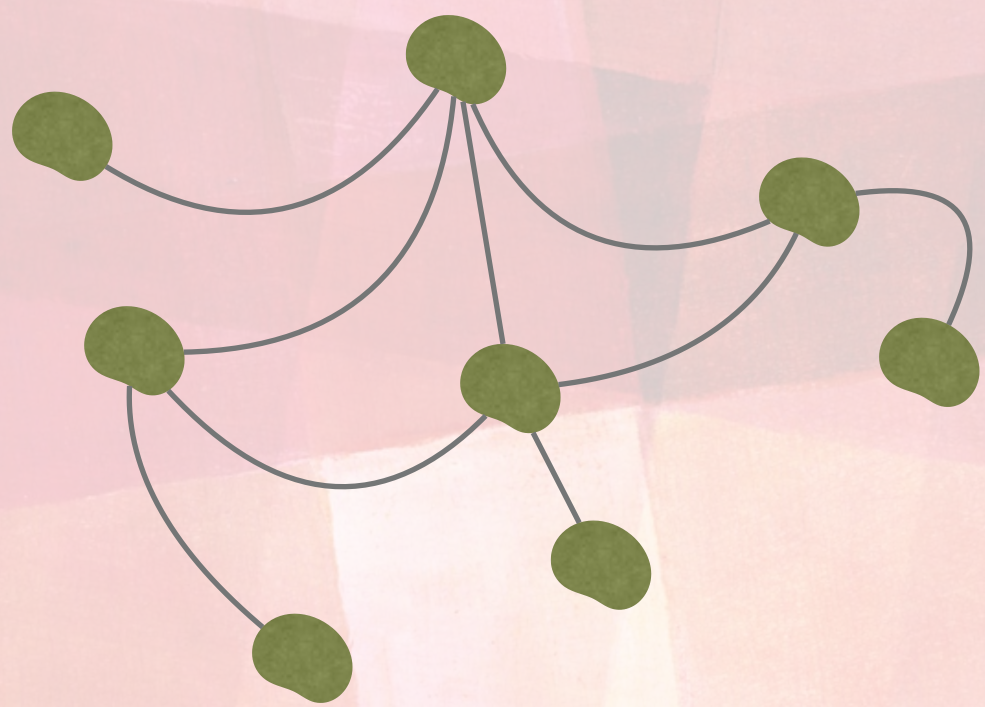
#rounds

#dimensions

- Arm independence is too strong and unnecessary
- Replace **N** with something much smaller
  - problem/instance/data dependent
  - example: linear bandits **N** to **D**
- **In this talk: Graph** Bandits!
  - sequential problems where **actions are nodes** on a graph
  - find strategies that replace **N** with a smaller **graph-dependent** quantity

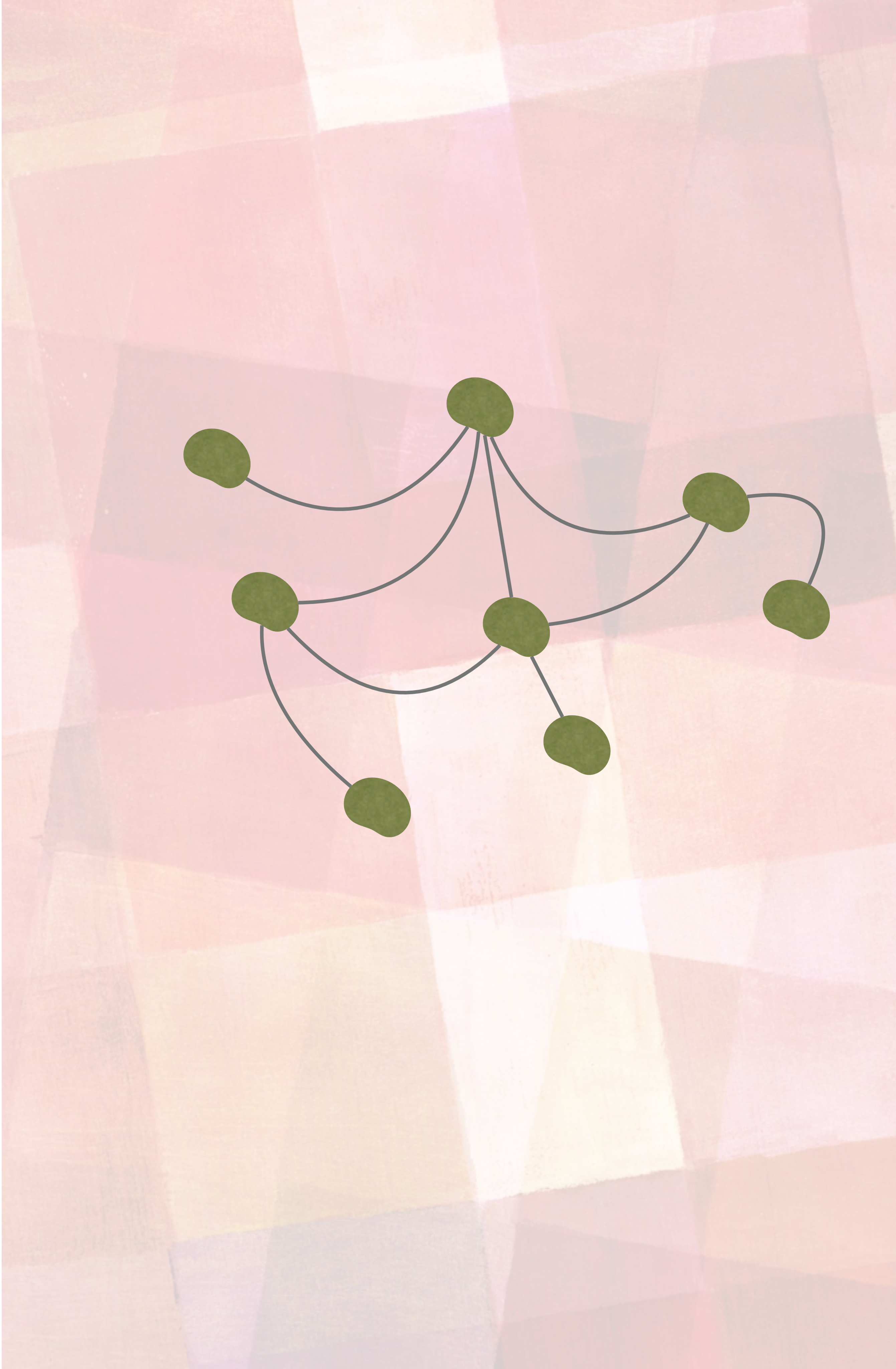


GRAPH BANDITS: GENERAL SETUP



GRAPH BANDITS: GENERAL SETUP

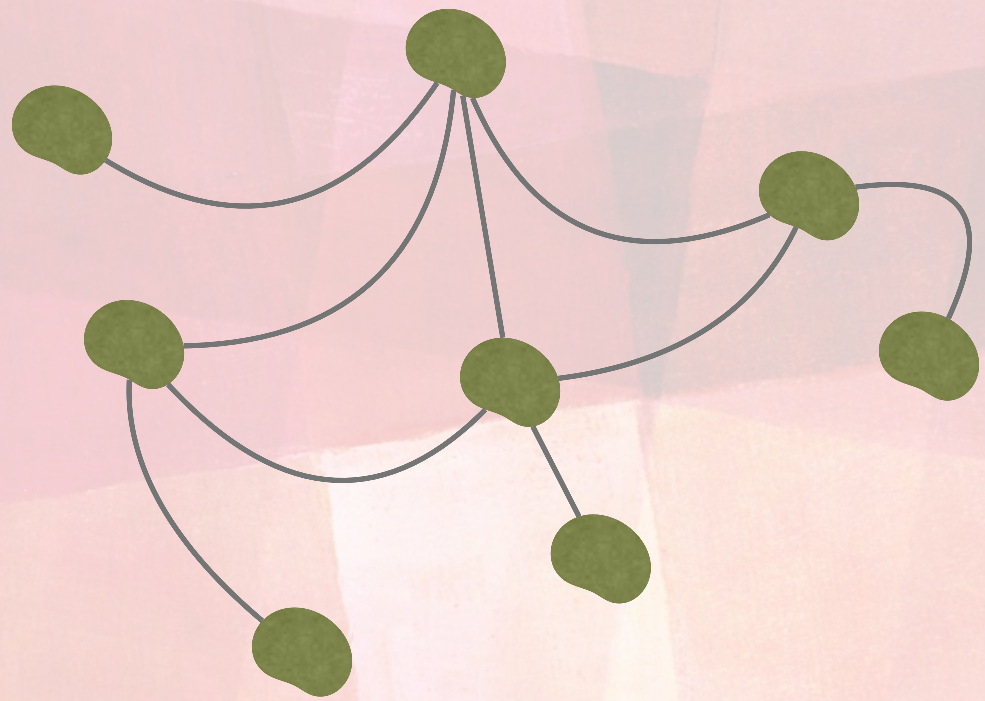
Every round **t** the learner



GRAPH BANDITS: GENERAL SETUP

Every round **t** the learner

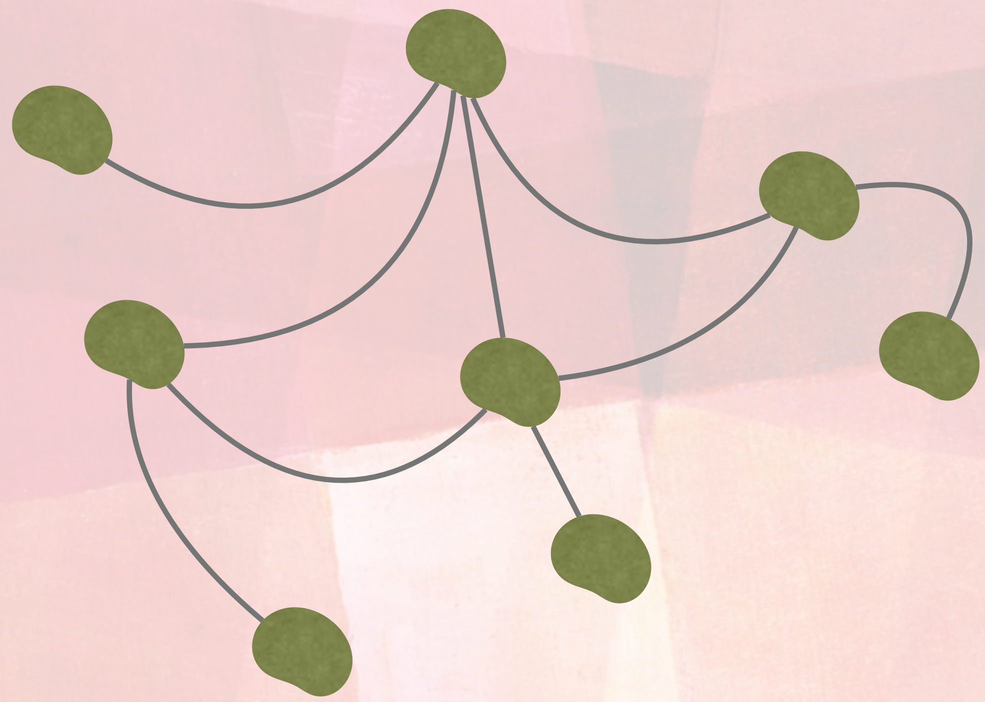
- picks a node $I_t \in [N]$



GRAPH BANDITS: GENERAL SETUP

Every round **t** the learner

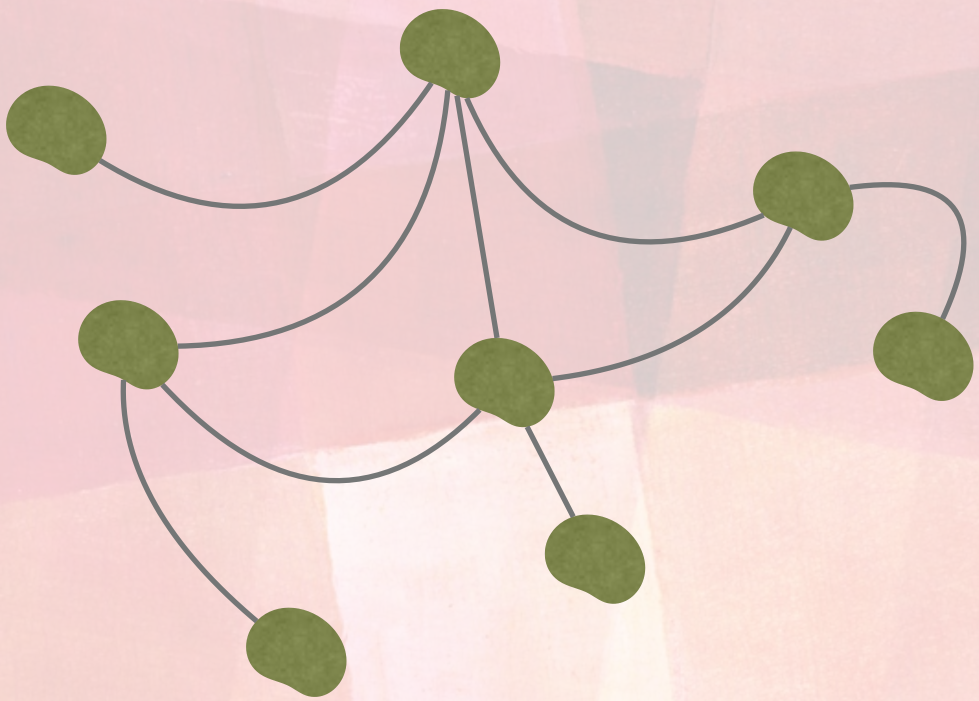
- picks a node $I_t \in [N]$
- incurs a loss ℓ_{t,I_t}



GRAPH BANDITS: GENERAL SETUP

Every round **t** the learner

- picks a node $I_t \in [N]$
- incurs a loss ℓ_{t,I_t}
- optional feedback



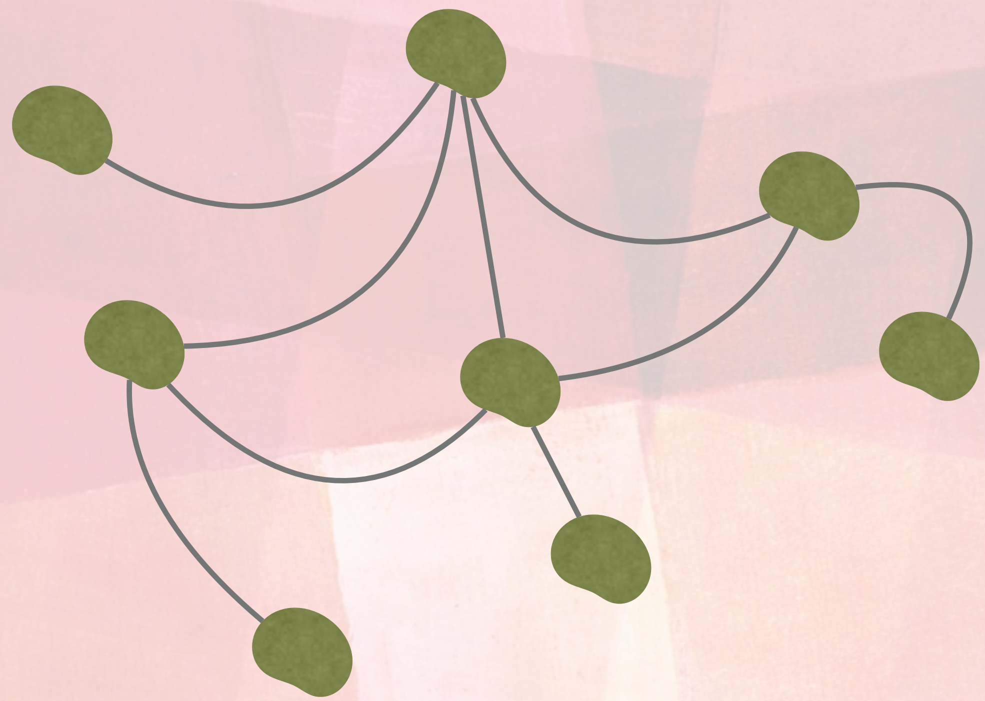
GRAPH BANDITS: GENERAL SETUP

Every round **t** the learner

- picks a node $I_t \in [N]$
- incurs a loss ℓ_{t,I_t}
- optional feedback

The performance is total expected regret

$$R_T = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T} (\ell_{t,I_t} - \ell_{t,i})\right]$$



GRAPH BANDITS: GENERAL SETUP

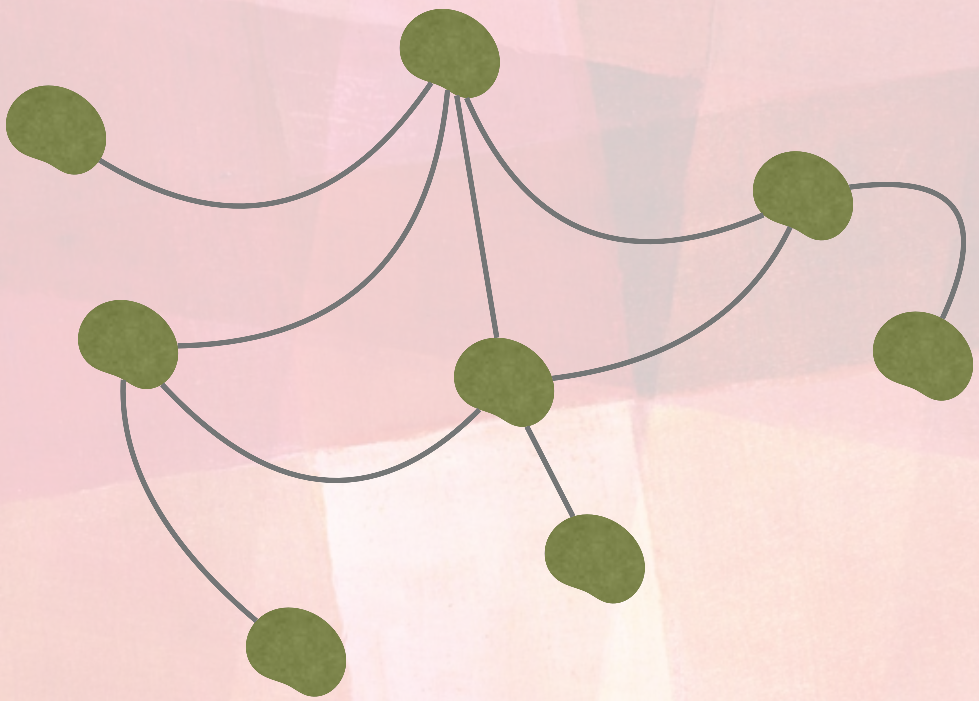
Every round **t** the learner

- picks a node $I_t \in [N]$
- incurs a loss ℓ_{t,I_t}
- optional feedback

The performance is total expected regret

$$R_T = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T} (\ell_{t,I_t} - \ell_{t,i})\right]$$

Specific problems differ in



GRAPH BANDITS: GENERAL SETUP

Every round **t** the learner

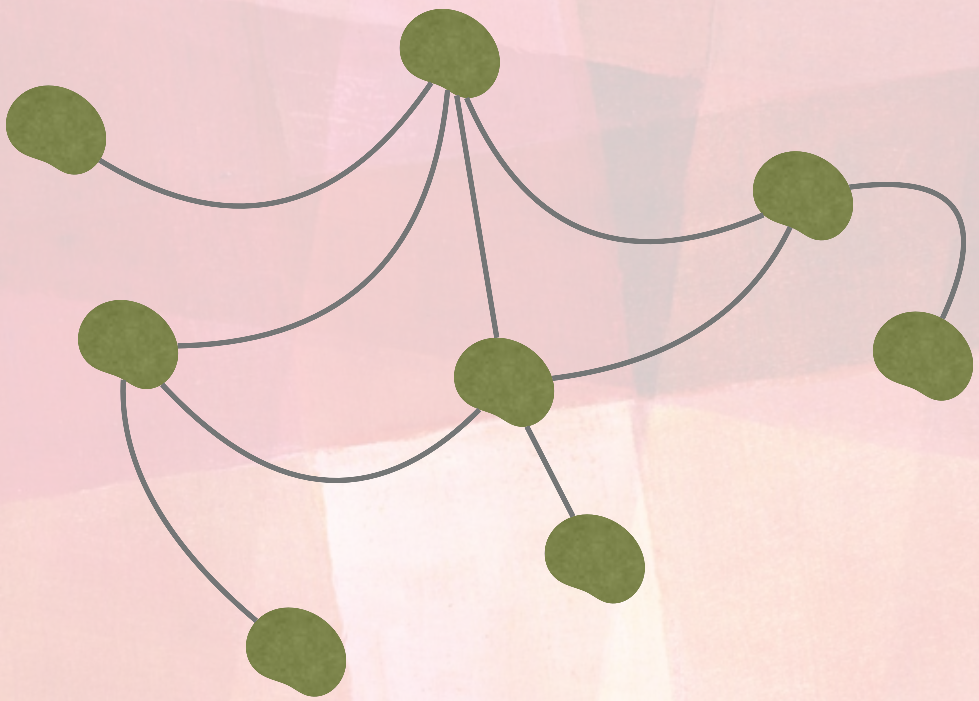
- picks a node $I_t \in [N]$
- incurs a loss ℓ_{t,I_t}
- optional feedback

The performance is total expected regret

$$R_T = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T}(\ell_{t,I_t} - \ell_{t,i})\right]$$

1. loss

Specific problems differ in



GRAPH BANDITS: GENERAL SETUP

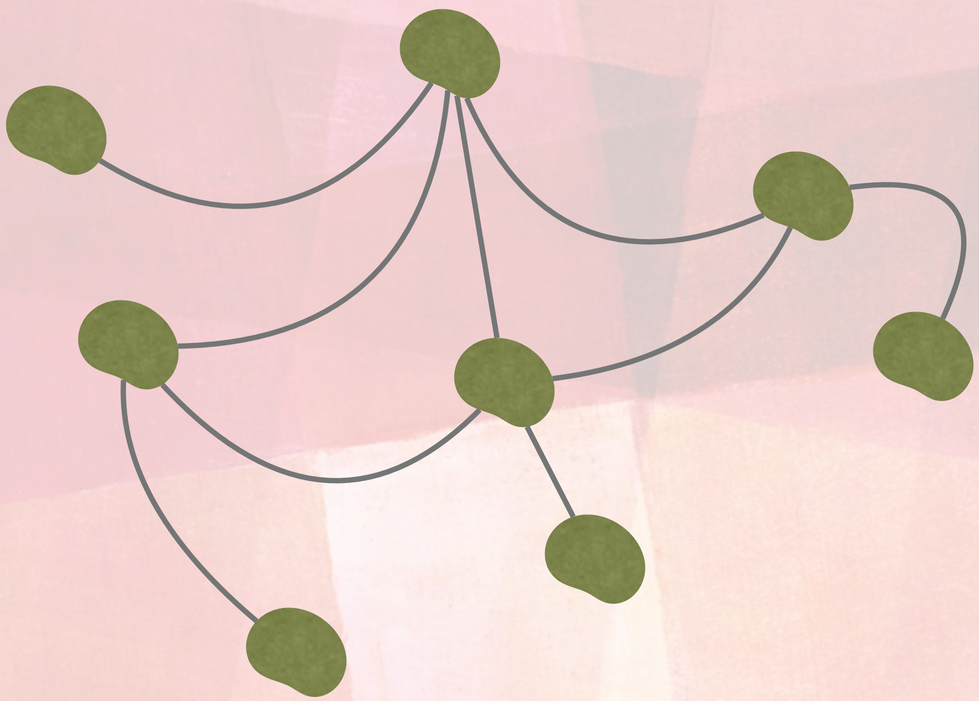
Every round **t** the learner

- picks a node $I_t \in [N]$
- incurs a loss ℓ_{t,I_t}
- optional feedback

The performance is total expected regret

$$R_T = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T} (\ell_{t,I_t} - \ell_{t,i})\right]$$

Specific problems differ in
1. loss
2. feedback



GRAPH BANDITS: GENERAL SETUP

Every round **t** the learner

- picks a node $I_t \in [N]$
- incurs a loss ℓ_{t,I_t}
- optional feedback

The performance is total expected regret

$$R_T = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T} (\ell_{t,I_t} - \ell_{t,i})\right]$$

Specific problems differ in

1. loss
2. feedback
3. guarantees

STRUCTURES IN BANDIT PROBLEMS

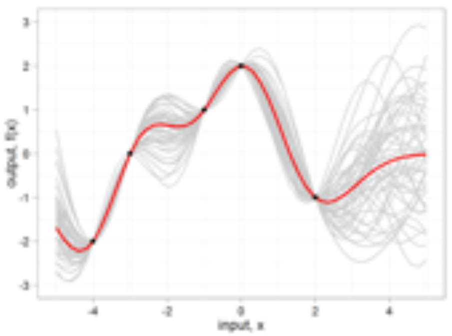
STRUCTURES IN BANDIT PROBLEMS

GRAPHS

STRUCTURES IN BANDIT PROBLEMS

GRAPHS

KERNELS

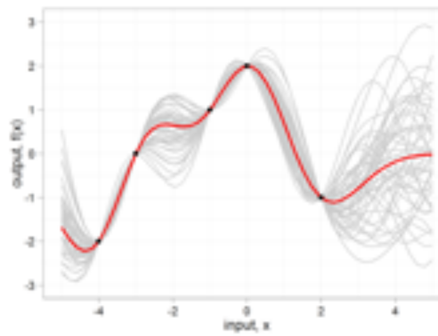


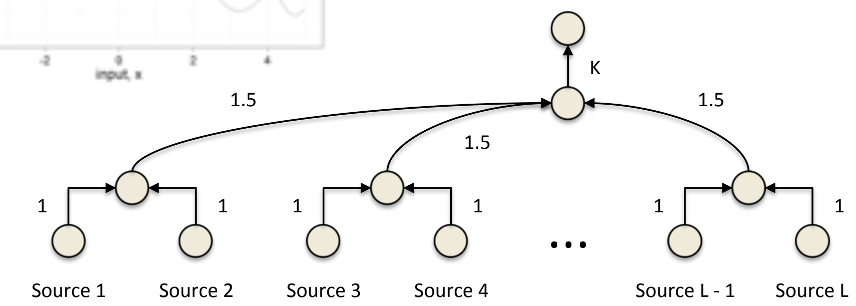
STRUCTURES IN BANDIT PROBLEMS

GRAPHS

KERNELS

POLYMATROIDS





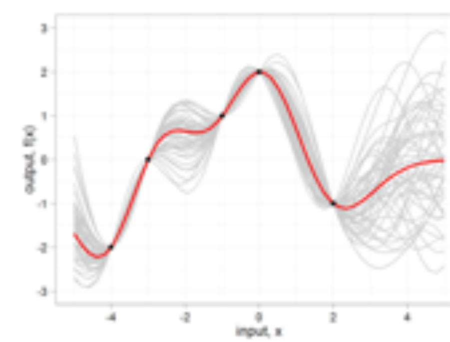
STRUCTURES IN BANDIT PROBLEMS

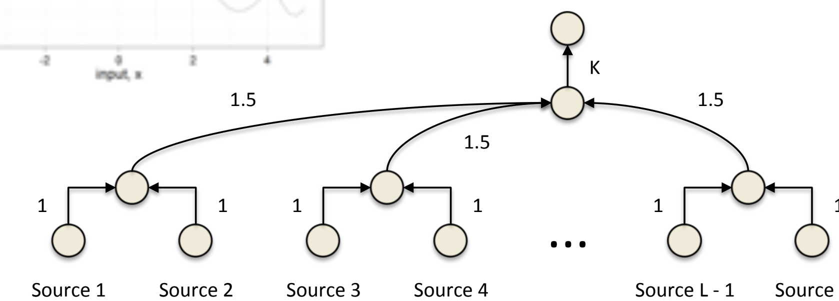
GRAPHS

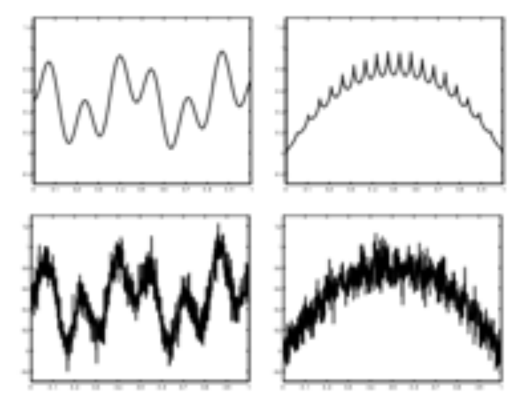
KERNELS

POLYMATROIDS

CONTINUOUS FUNCTIONS







STRUCTURES IN BANDIT PROBLEMS

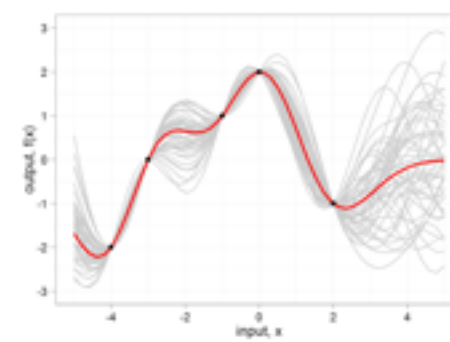
GRAPHS

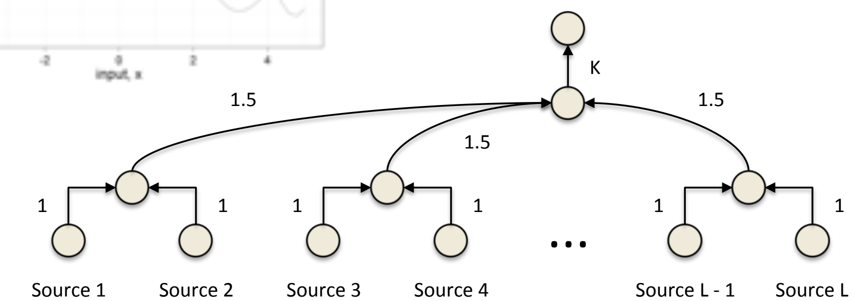
KERNELS

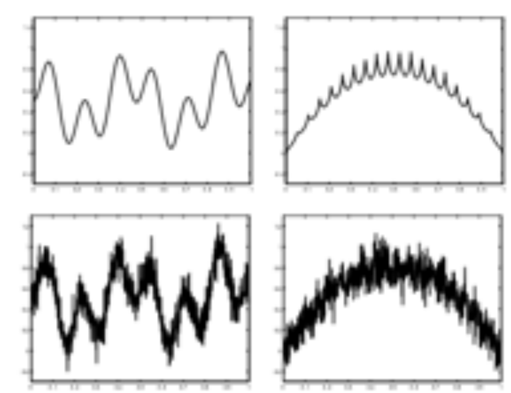
POLYMATROIDS

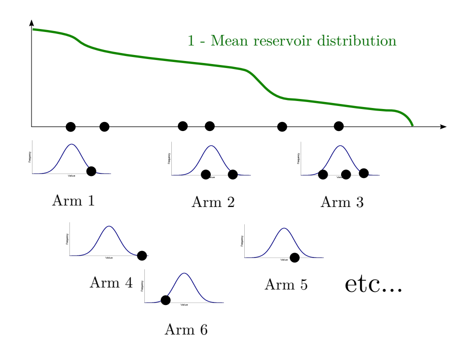
CONTINUOUS FUNCTIONS

STRUCTURES WITHOUT TOPOLOGY



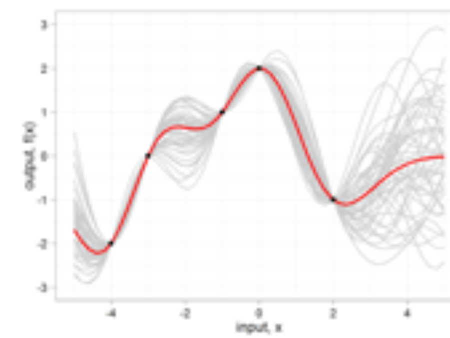


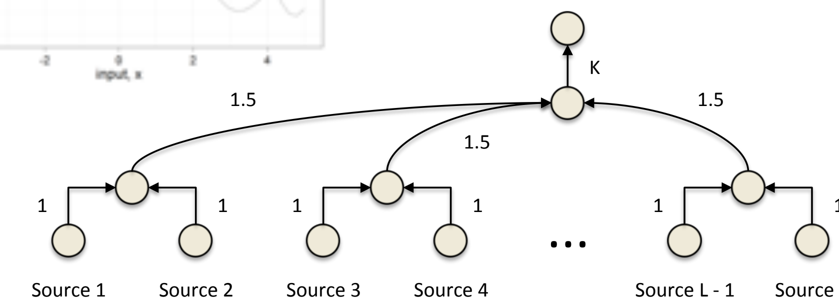


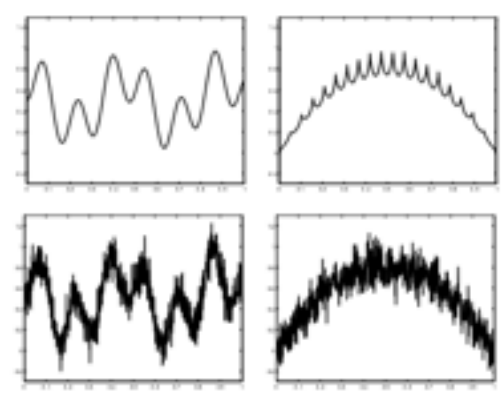


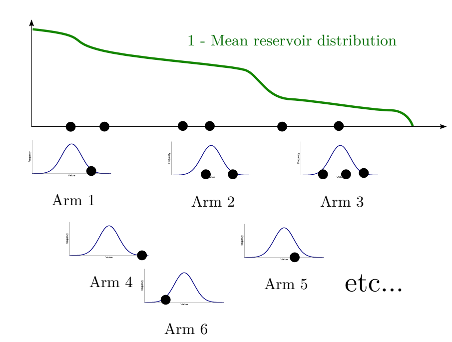
STRUCTURES IN BANDIT PROBLEMS

- GRAPHS
- KERNELS
- POLYMATROIDS
- CONTINUOUS FUNCTIONS
- STRUCTURES WITHOUT TOPOLOGY
- ...









STRUCTURES IN BANDIT PROBLEMS

GRAPHS

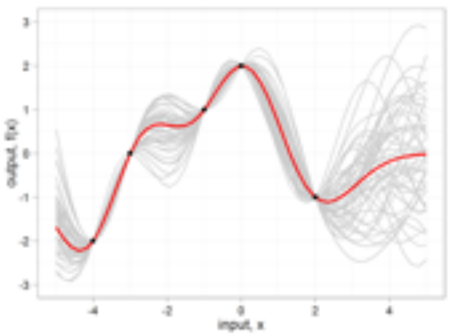
KERNELS

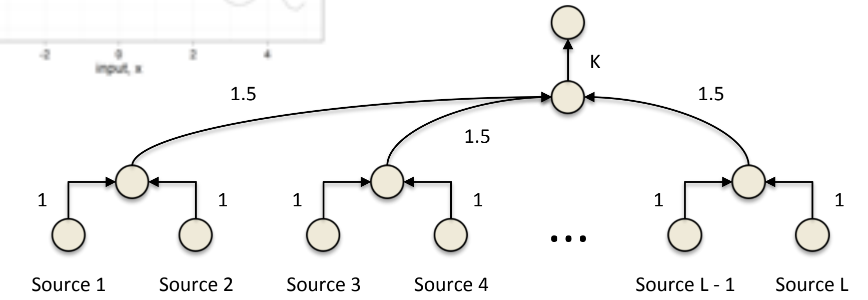
POLYMATROIDS

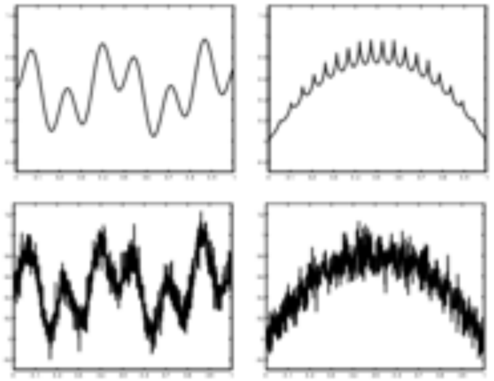
CONTINUOUS FUNCTIONS

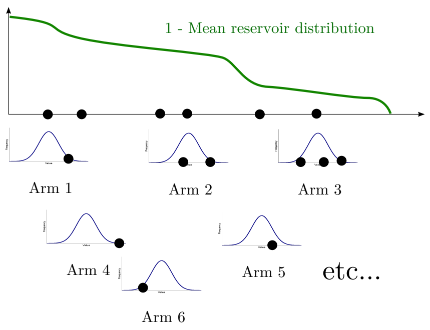
STRUCTURES WITHOUT TOPOLOGY

...









SPECIFIC GRAPH BANDIT SETTINGS

SPECIFIC GRAPH BANDIT SETTINGS

smoothness
spectral bandits

SPECIFIC GRAPH BANDIT SETTINGS

smoothness
spectral bandits

side observations
on graphs

SPECIFIC GRAPH BANDIT SETTINGS

smoothness
spectral bandits

side observations
on graphs

noisy side
observations
on graphs

SPECIFIC **GRAPH** BANDIT SETTINGS

smoothness
spectral bandits

side observations
on graphs

influence maximisation
revealing bandits

noisy side observations
on graphs

SPECIFIC GRAPH BANDIT SETTINGS

smoothness
spectral bandits

$R_T = \widetilde{\mathcal{O}}\left(d\sqrt{T \ln T}\right)$

#relevant eigenvectors

side observations
on graphs

influence maximisation
revealing bandits

noisy side observations
on graphs

SPECIFIC GRAPH BANDIT SETTINGS

smoothness
spectral bandits

$$R_T = \widetilde{\mathcal{O}}\left(d\sqrt{T \ln T}\right)$$

#relevant eigenvectors

side observations
on graphs

$$R_T = \widetilde{\mathcal{O}}\left(\sqrt{\overline{\alpha} T \ln N}\right)$$

independence number

influence maximisation
revealing bandits

noisy side observations
on graphs

SPECIFIC GRAPH BANDIT SETTINGS

smoothness
spectral bandits
$R_T = \widetilde{\mathcal{O}}\left(d\sqrt{T \ln T}\right)$

#relevant eigenvectors

side observations
on graphs
$R_T = \widetilde{\mathcal{O}}\left(\sqrt{\overline{\alpha}\, T \ln N}\right)$

independence number

influence maximisation
revealing bandits

noisy side observations
on graphs
$R_T = \widetilde{\mathcal{O}}\left(\sqrt{\overline{\alpha}^{\star}\, T \ln N}\right)$

effective independence number

SPECIFIC GRAPH BANDIT SETTINGS

smoothness
spectral bandits
$R_T = \widetilde{\mathcal{O}}\left(d\sqrt{T \ln T}\right)$
#relevant eigenvectors

side observations
on graphs
$R_T = \widetilde{\mathcal{O}}\left(\sqrt{\overline{\alpha} T \ln N}\right)$
independence number

influence maximisation
revealing bandits
$R_T = \widetilde{\mathcal{O}}\left(\sqrt{r_* T D_*}\right)$
detectable dimension

noisy side observations
on graphs
$R_T = \widetilde{\mathcal{O}}\left(\sqrt{\overline{\alpha}^{\star} T \ln N}\right)$
effective independence number

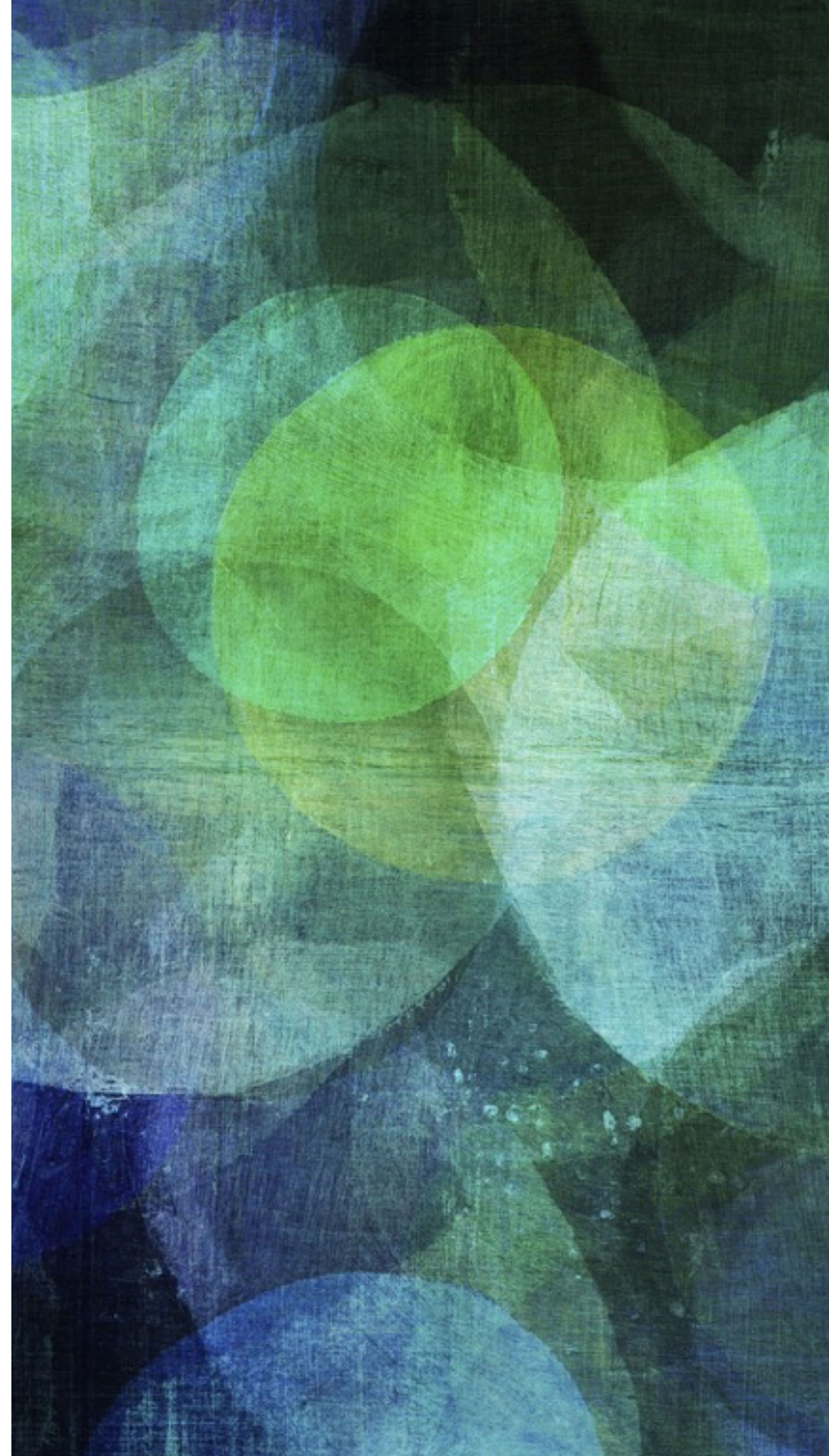
MV, Munos, Kveton, Kocák: **Spectral Bandits for Smooth Graph Functions**, ICML 2014

Kocák, MV, Munos, Agrawal: **Spectral Thompson Sampling**, AAAI 2014

Hanawal, Saligrama, MV, Munos: **Cheap Bandits**, ICML 2015

SPECTRAL BANDITS

exploiting smoothness of rewards on graphs



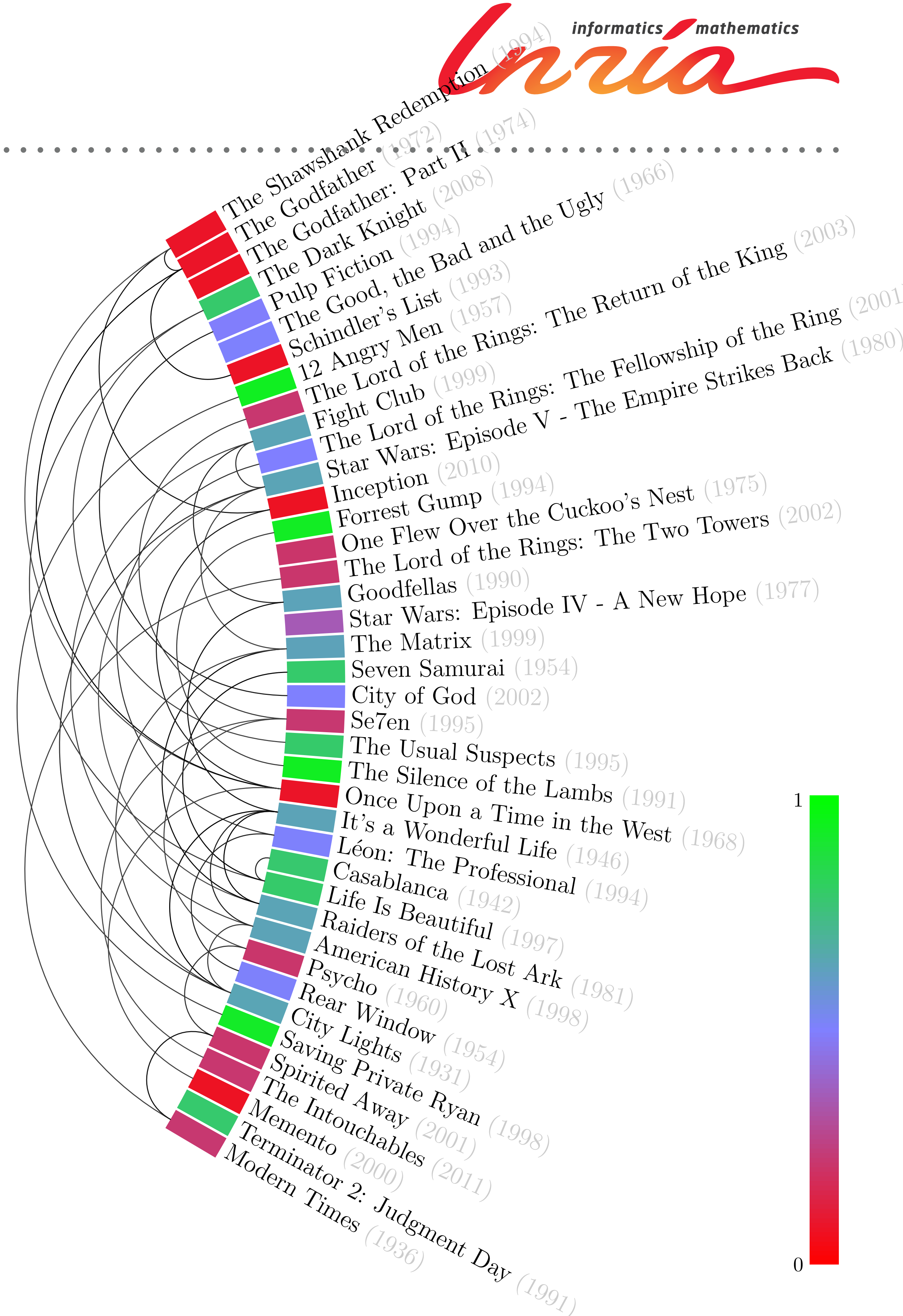
SPECTRAL BANDITS

Assumptions

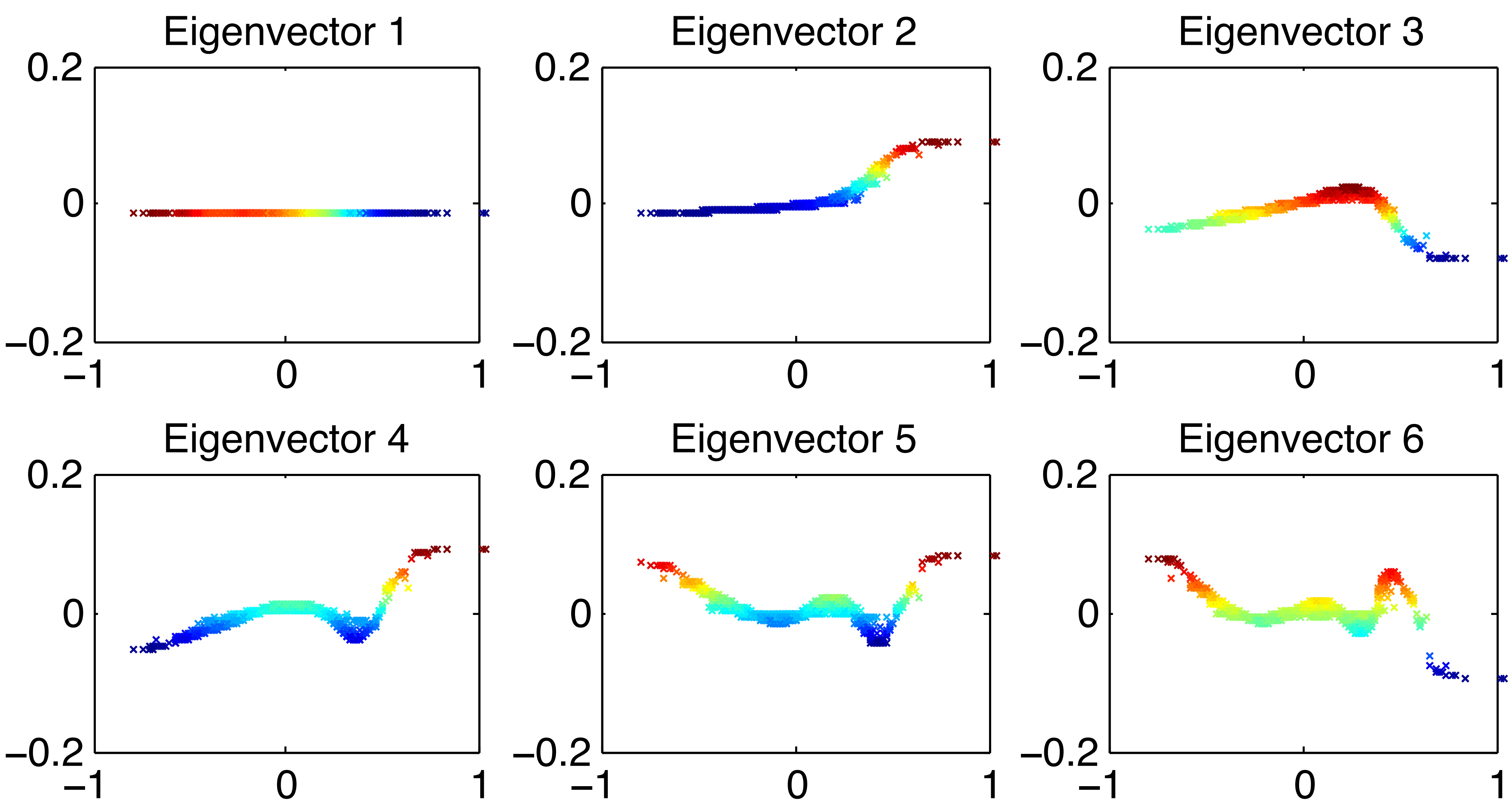
- Unknown reward function $f : V(G) \to \mathbb{R}$.
- Function f is **smooth** on a graph.
- Neighboring movies $\Rightarrow$ similar preferences.
- Similar preferences $\not\Rightarrow$ neighboring movies.

Desiderata

An algorithm useful in the case $T \ll N$!



FLIXSTER DATA



Eigenvectors from the Flixster data corresponding to the smallest few eigenvalues of the graph Laplacian projected onto the first principal component of data. Colors indicate the values.

SPECTRAL BANDIT: LEARNING SETTING

Learning setting for a bandit algorithm π

- In each time t step choose a node $\pi(t)$.
- the $\pi(t)$-th row $\mathbf{x}_{\pi(t)}$ of the matrix $\mathbf{Q}$ corresponds to the arm $\pi(t)$.
- Obtain noisy reward $r_t = \mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^* + \varepsilon_t$. Note: $\mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^* = f_{\pi(t)}$
  - ε_t is R-sub-Gaussian noise. $\forall\xi \in \mathbb{R}, \mathbb{E}[e^{\xi\varepsilon_t}] \leq \exp\left(\xi^2 R^2/2\right)$
- Minimize cumulative regret

$$R_T = T \max_a \left(\mathbf{x}_a^{\mathsf{T}}\boldsymbol{\alpha}^*\right) - \sum_{t=1}^{T} \mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^*.$$

Learning setting for a bandit algorithm π

- In each time t step choose a node $\pi(t)$.
- the $\pi(t)$-th row $\mathbf{x}_{\pi(t)}$ of the matrix $\mathbf{Q}$ corresponds to the arm $\pi(t)$.
- Obtain noisy reward $r_t = \mathbf{x}_{\pi(t)}^\mathsf{T}\boldsymbol{\alpha}^* + \varepsilon_t$. Note: $\mathbf{x}_{\pi(t)}^\mathsf{T}\boldsymbol{\alpha}^* = f_{\pi(t)}$
  - ε_t is R-sub-Gaussian noise. $\forall \xi \in \mathbb{R},\ \mathbb{E}[e^{\xi\varepsilon_t}] \leq \exp\left(\xi^2 R^2/2\right)$
- Minimize cumulative regret

$$R_T = T \max_a \left(\mathbf{x}_a^\mathsf{T}\boldsymbol{\alpha}^*\right) - \sum_{t=1}^{T} \mathbf{x}_{\pi(t)}^\mathsf{T}\boldsymbol{\alpha}^*.$$

Can we just use linear bandits?

LINEAR VS. SPECTRAL BANDITS

- **Linear bandit algorithms**
  - **LinUCB** (Li et al., 2010)
    - Regret bound $\approx D\sqrt{T \ln T}$
  - **LinearTS** (Agrawal and Goyal, 2013)
    - Regret bound $\approx D\sqrt{T \ln N}$

Note: D is ambient dimension, in our case N, length of x_i.
Number of actions, e.g., all possible movies $\rightarrow$ **HUGE!**

- **Spectral bandit algorithms**
  - **SpectralUCB** (Valko et al., ICML 2014)
    - Regret bound $\approx d\sqrt{T \ln T}$
    - Operations per step: $D^2 N$
  - **SpectralTS** (Kocák et al., AAAI 2014)
    - Regret bound $\approx d\sqrt{T \ln N}$
    - Operations per step: $D^2 + DN$

Note: d is effective dimension, usually much smaller than D.

SPECTRAL BANDITS – EFFECTIVE DIMENSION

- **Effective dimension:** Largest d such that

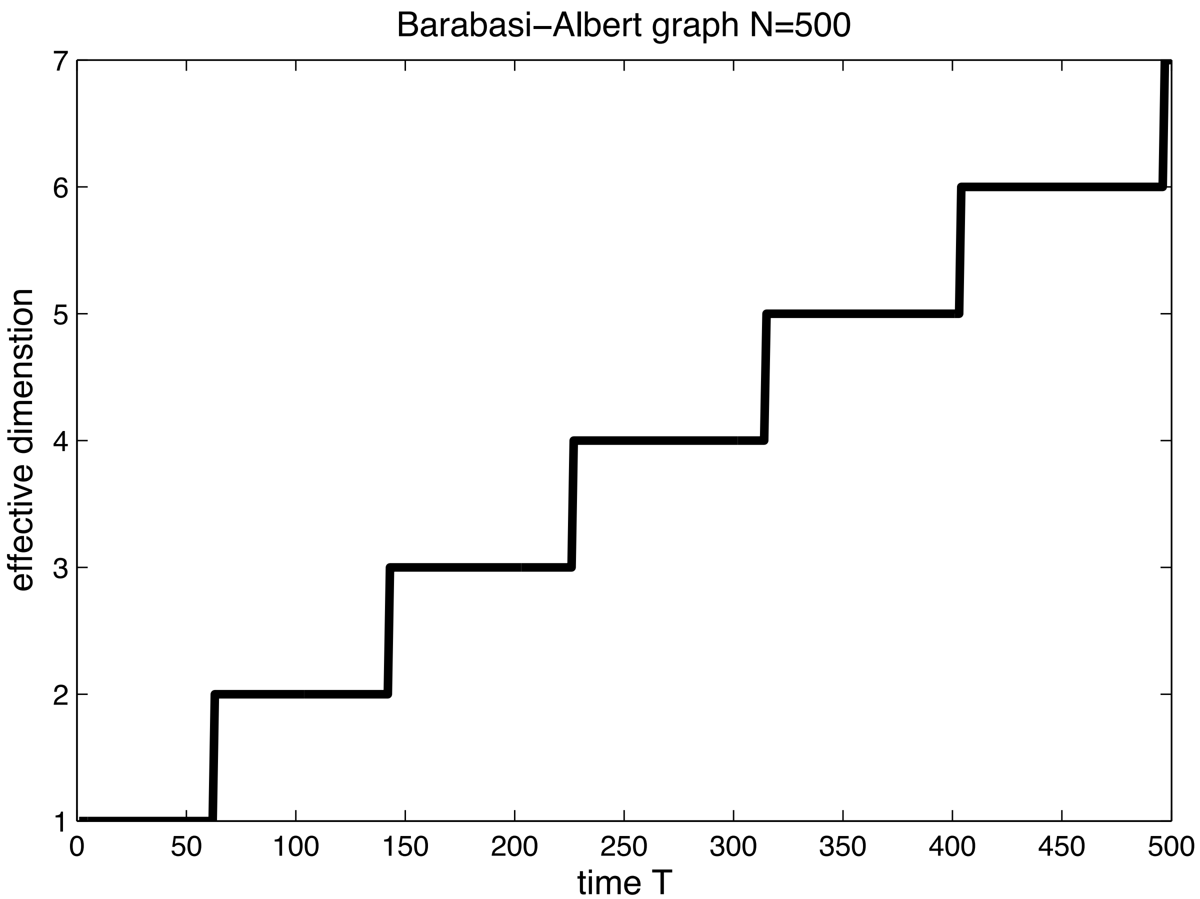
$$(d-1)\lambda_d \leq \frac{T}{\log(1+T/\lambda)}.$$

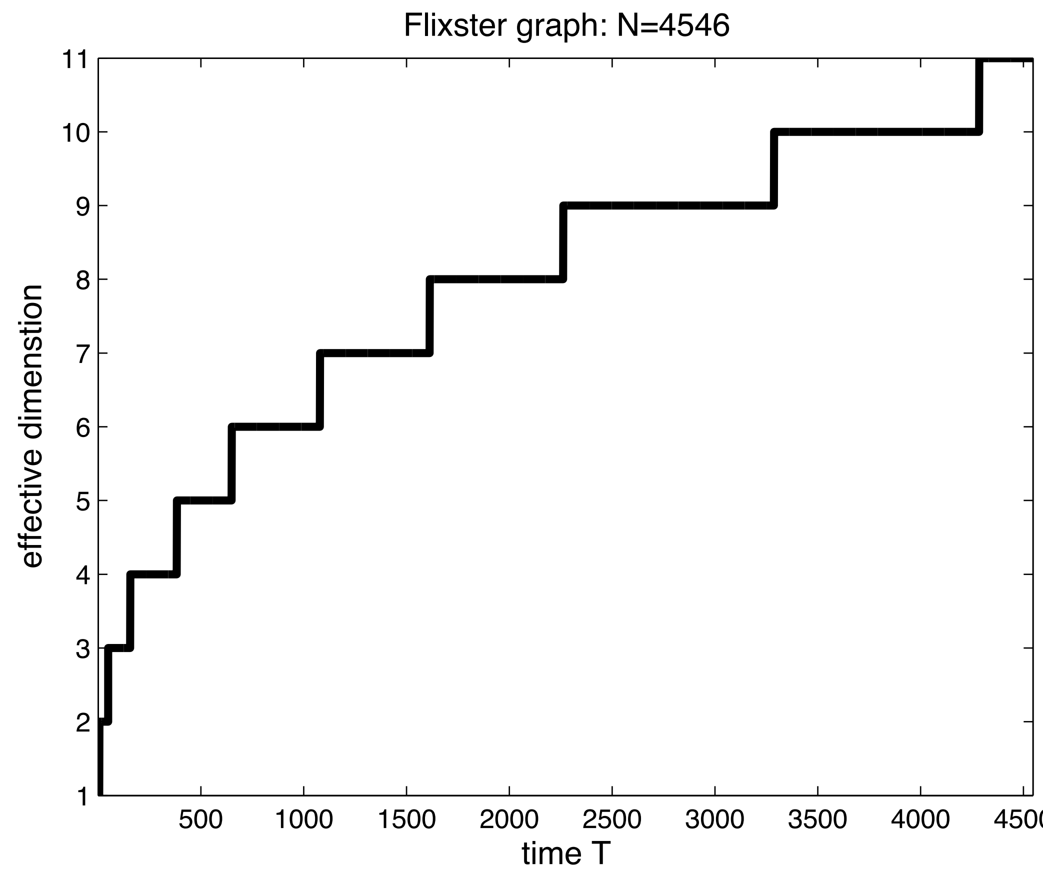
- Function of time horizon and graph properties
- λ_i : i-th smallest eigenvalue of $\mathbf{\Lambda}$.
- λ: Regularization parameter of the algorithm.

Properties:

- d is small when the coefficients λ_i grow rapidly above time.
- d is related to the number of "non-negligible" dimensions.
- Usually d is much smaller than D in real world graphs.
- Can be computed beforehand.

SPECTRAL BANDITS - EFFECTIVE DIMENSION





$$d \ll D$$

Note: In our setting $T < N = D$.

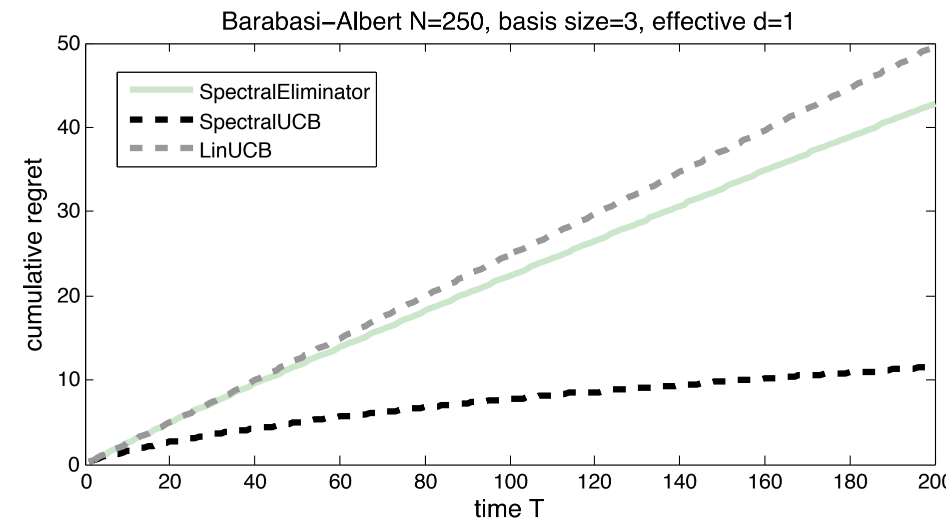
SPECTRALUCB REGRET BOUND

- d: Effective dimension.
- λ: Minimal eigenvalue of $\mathbf{\Lambda} = \mathbf{\Lambda_L} + \lambda\mathbf{I}$.
- C: Smoothness upper bound, $\|\boldsymbol{\alpha}^*\|_{\mathbf{\Lambda}} \leq C$.
- $\mathbf{x}_i^\mathsf{T}\boldsymbol{\alpha}^* \in [-1, 1]$ for all i.

The **cumulative regret** R_T of **SpectralUCB** is with probability $1-\delta$ bounded as

$$R_T \leq \left(8R\sqrt{d\ln\frac{\lambda+T}{\lambda} + 2\ln\frac{1}{\delta}} + 4C + 4\right)\sqrt{dT\ln\frac{\lambda+T}{\lambda}}.$$

SPECTRALUCB REGRET BOUND



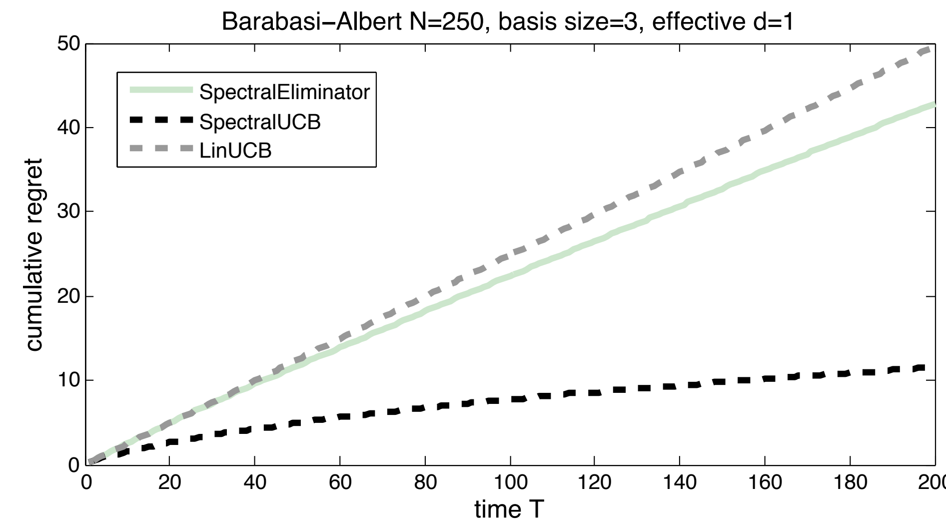
- d: Effective dimension.
- λ: Minimal eigenvalue of $\mathbf{\Lambda} = \mathbf{\Lambda_L} + \lambda \mathbf{I}$.
- C: Smoothness upper bound, $\|\boldsymbol{\alpha}^*\|_{\mathbf{\Lambda}} \leq C$.
- $\mathbf{x}_i^\mathsf{T} \boldsymbol{\alpha}^* \in [-1, 1]$ for all i.

The **cumulative regret** R_T of **SpectralUCB** is with probability $1 - \delta$ bounded as

$$R_T \leq \left(8R\sqrt{d \ln \frac{\lambda + T}{\lambda} + 2 \ln \frac{1}{\delta}} + 4C + 4 \right) \sqrt{dT \ln \frac{\lambda + T}{\lambda}}.$$

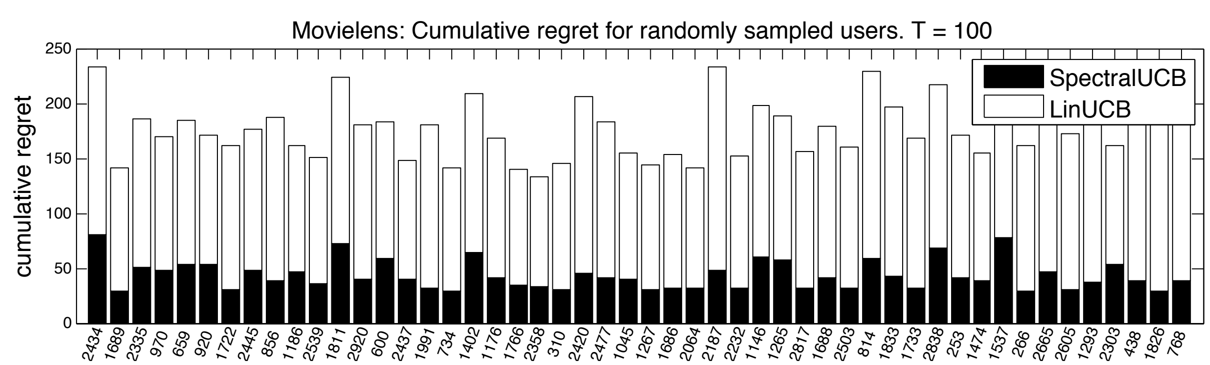
SPECTRALUCB REGRET BOUND

- d: Effective dimension.
- λ: Minimal eigenvalue of $\mathbf{\Lambda} = \mathbf{\Lambda_L} + \lambda \mathbf{I}$.
- C: Smoothness upper bound, $\|\boldsymbol{\alpha}^*\|_{\mathbf{\Lambda}} \leq C$.
- $\mathbf{x}_i^\mathsf{T} \boldsymbol{\alpha}^* \in [-1, 1]$ for all i.



The **cumulative regret** R_T of **SpectralUCB** is with probability $1 - \delta$ bounded as

$$R_T \leq \left(8R\sqrt{d \ln \frac{\lambda + T}{\lambda} + 2 \ln \frac{1}{\delta}} + 4C + 4 \right) \sqrt{dT \ln \frac{\lambda + T}{\lambda}}.$$

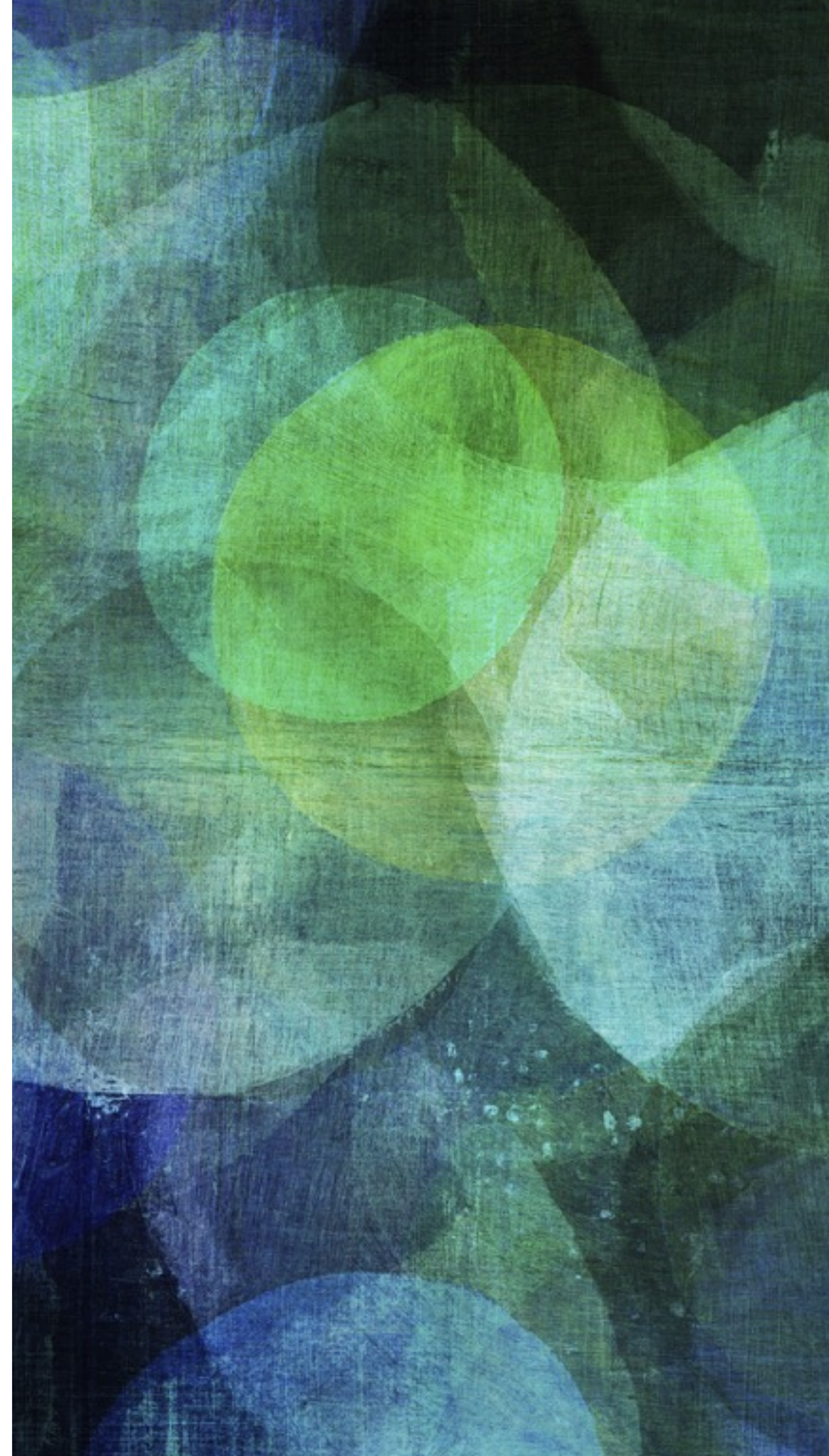


Kocák, Neu, MV, Munos: **Efficient learning by implicit exploration in bandit problems with side observations**, NIPS 2014

Kocák, Neu, MV: **Online learning with Erdos-Rényi side-observation graphs** UAI 2016 (to appear)

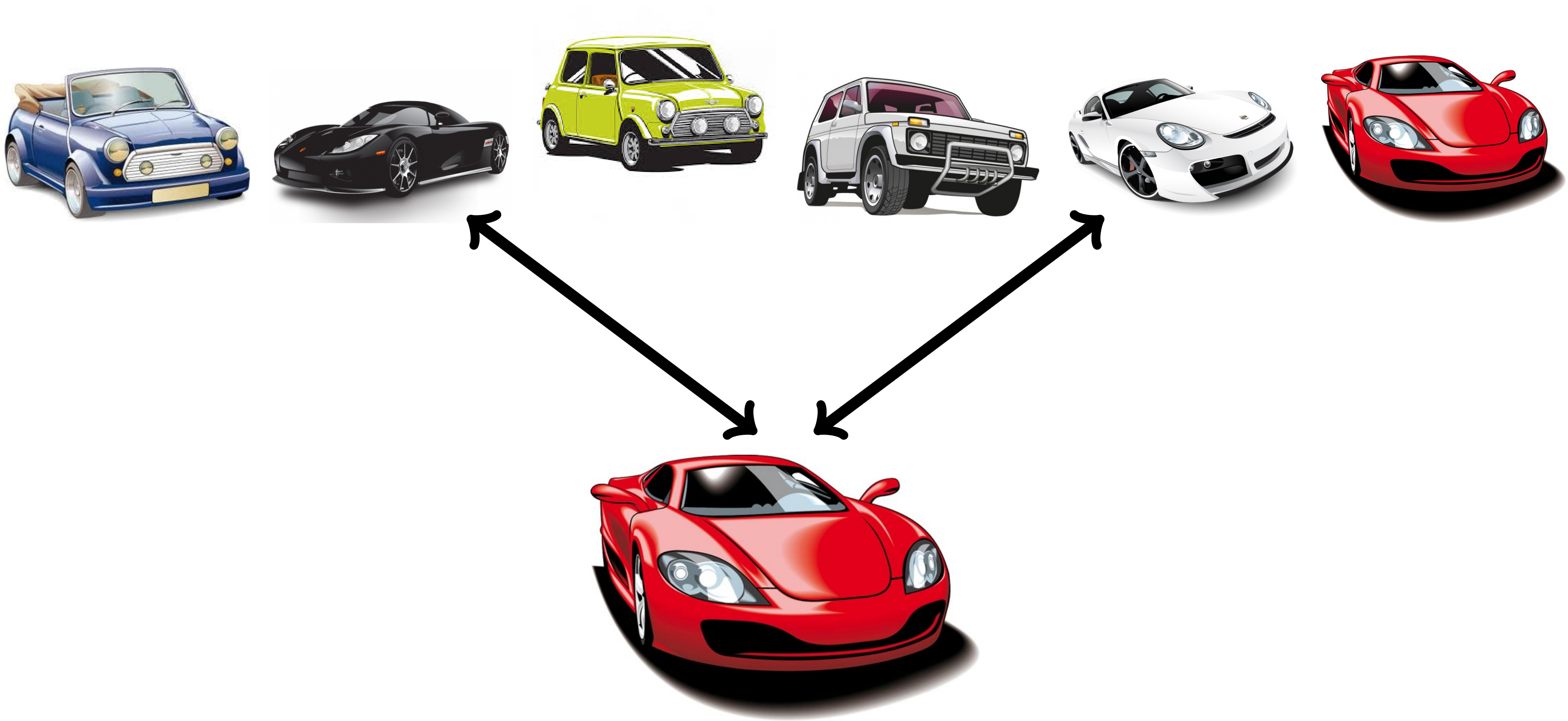
GRAPH BANDITS WITH SIDE OBSERVATIONS

exploiting **free** observations from neighbouring nodes



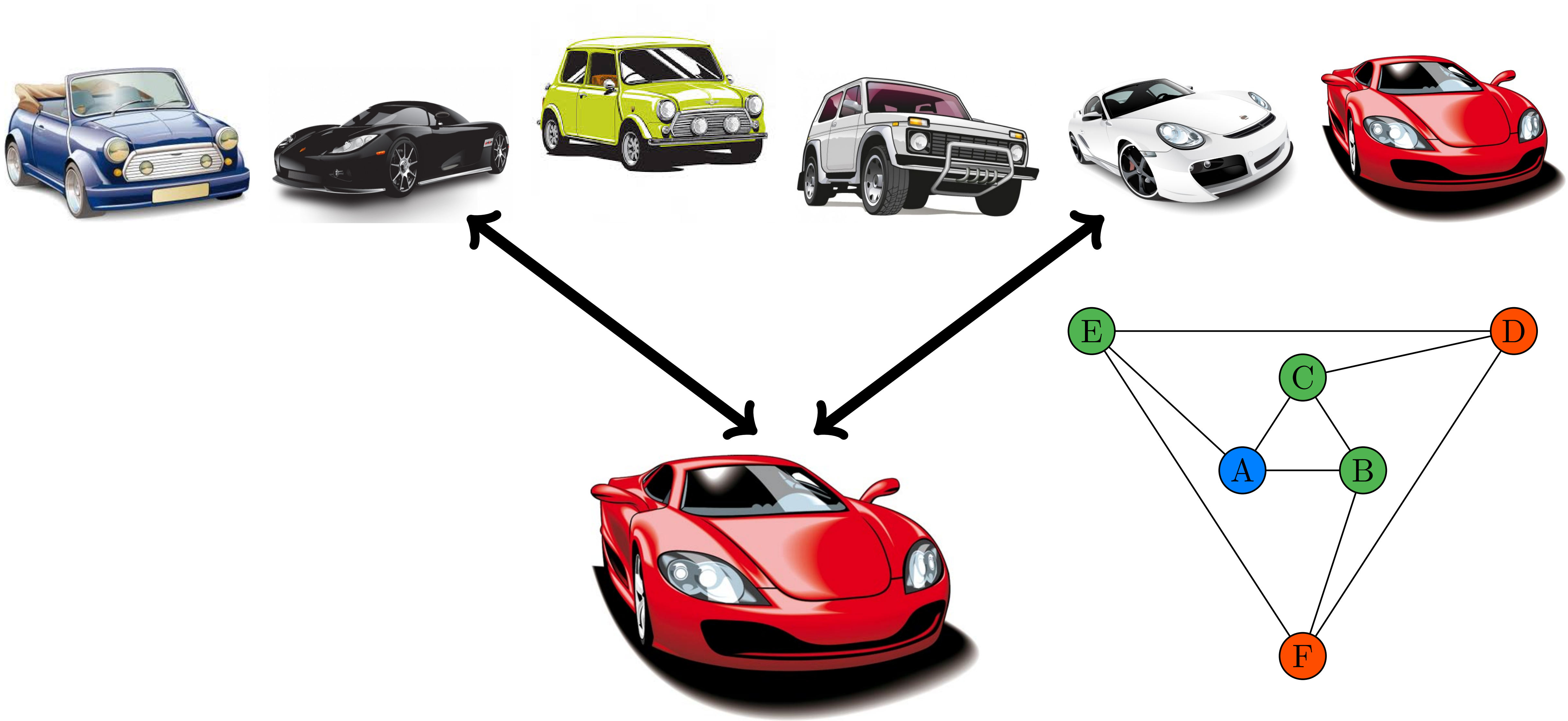
SIDE OBSERVATIONS: UNDIRECTED

Example 1: undirected observations



SIDE OBSERVATIONS: UNDIRECTED

Example 1: undirected observations

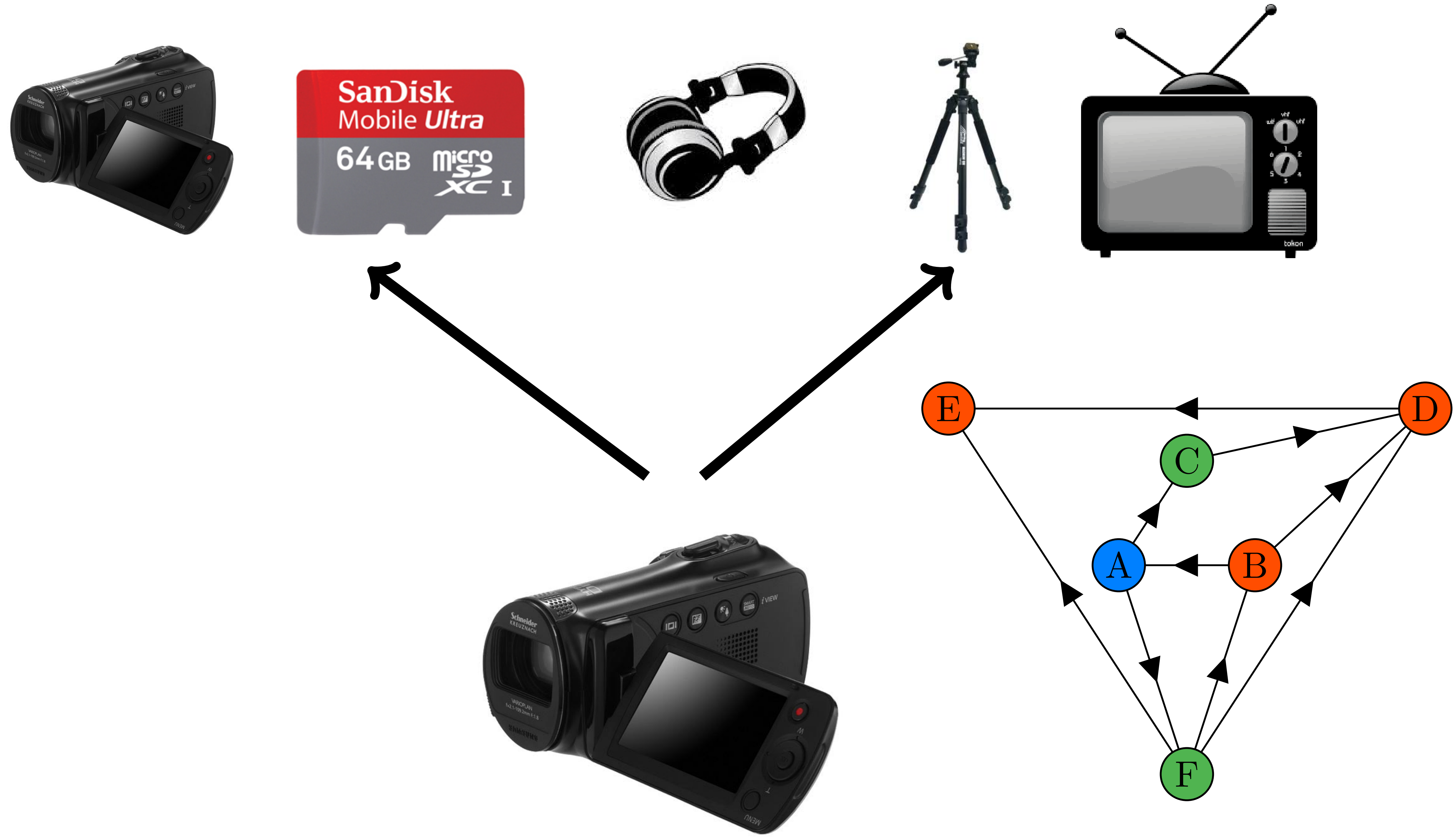


SIDE OBSERVATIONS: DIRECTED

Example 2: Directed observation

SIDE OBSERVATIONS: DIRECTED

Example 2: Directed observation

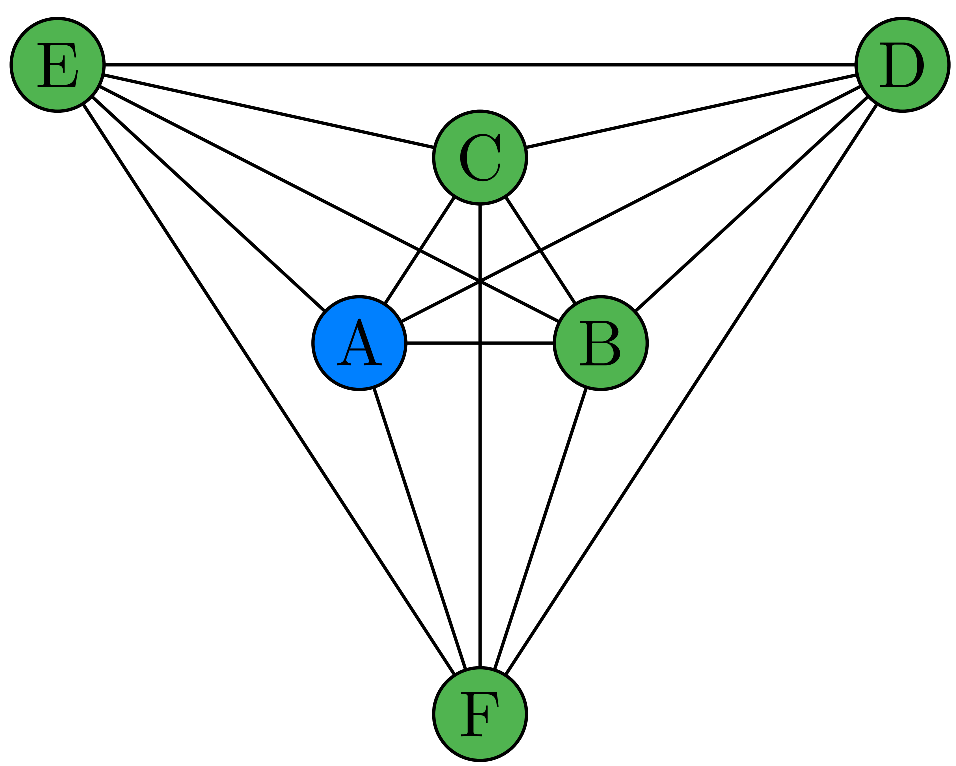


SIDE OBSERVATIONS – AN INTERMEDIATE GAME

Full Information setting

- Pick an action (e.g. action A)
- Observe losses of all actions
- $R_T = \widetilde{\mathcal{O}}(\sqrt{T})$

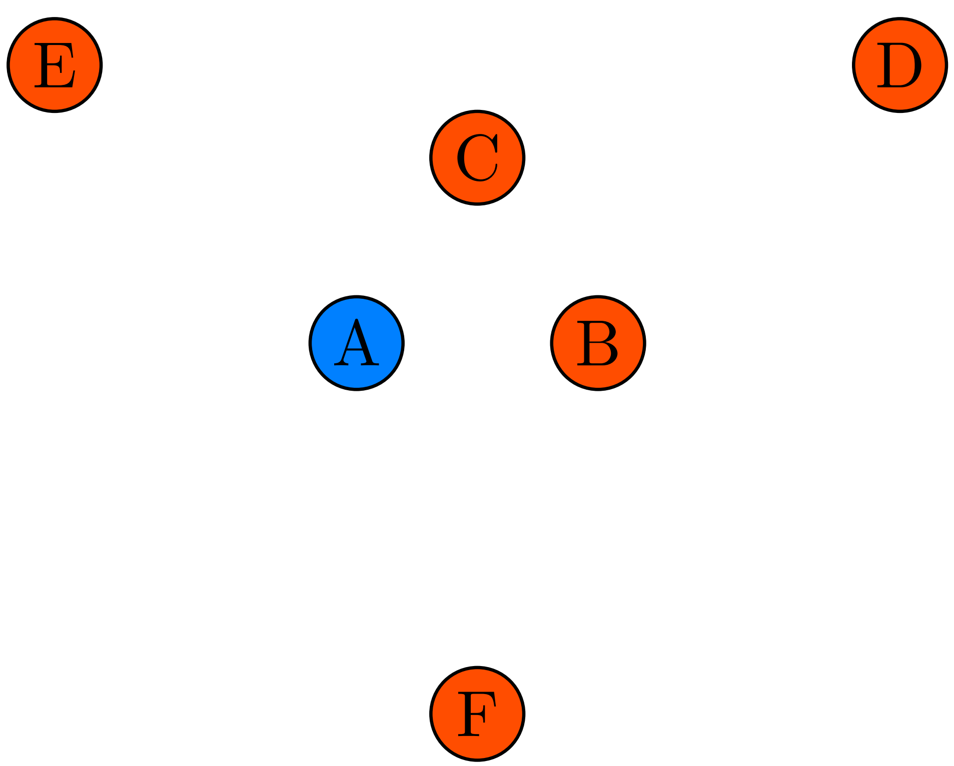
Hedge



Bandit setting

- Pick an action (e.g. action A)
- Observe loss of a chosen action
- $R_T = \widetilde{\mathcal{O}}(\sqrt{NT})$

EXP3



KNOWLEDGE OF OBSERVATION GRAPHS

KNOWLEDGE OF OBSERVATION GRAPHS

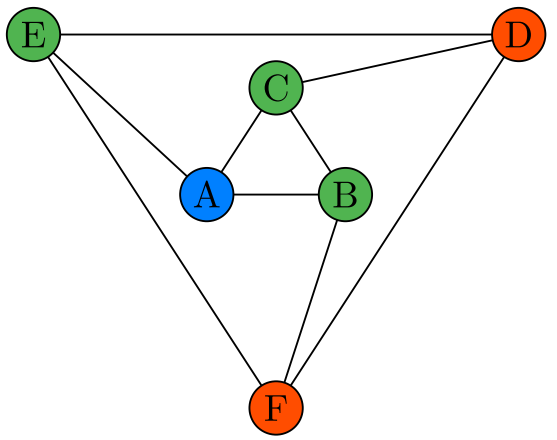
- ELP (Mannor and Shamir 2011)

KNOWLEDGE OF OBSERVATION GRAPHS

- ELP (Mannor and Shamir 2011)
    - **EXP3** - with **"LP balanced exploration"**

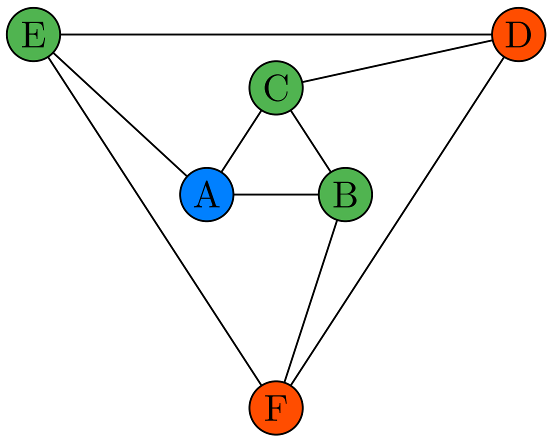
KNOWLEDGE OF OBSERVATION GRAPHS

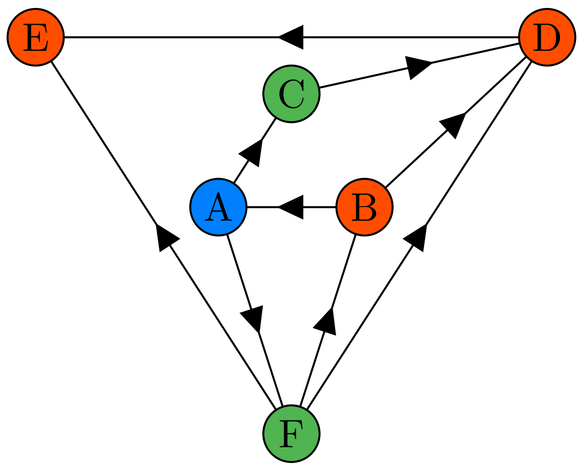
- ELP (Mannor and Shamir 2011)
  - **EXP3** - with **"LP balanced exploration"**
  - undirected $O(\sqrt{(\alpha T)})$ ✅ – needs to know G_t



KNOWLEDGE OF OBSERVATION GRAPHS

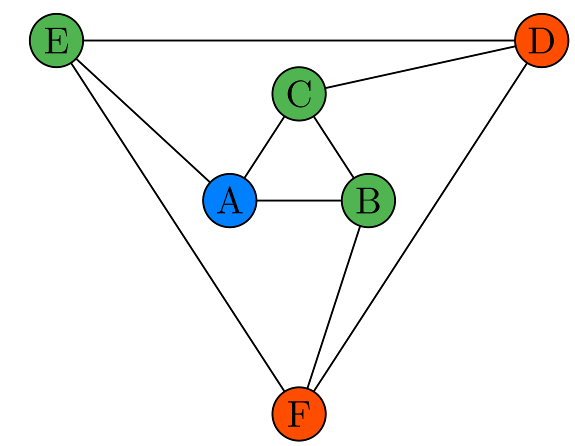
- ELP (Mannor and Shamir 2011)
  - **EXP3** - with **“LP balanced exploration”**
  - undirected $O(\sqrt{(\alpha T)})$ ✅ – needs to know G_t
  - directed case $O(\sqrt{(cT)})$ – needs to know G_t

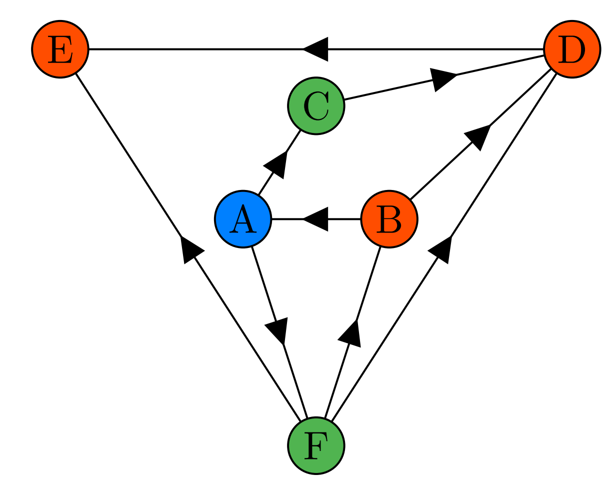




KNOWLEDGE OF OBSERVATION GRAPHS

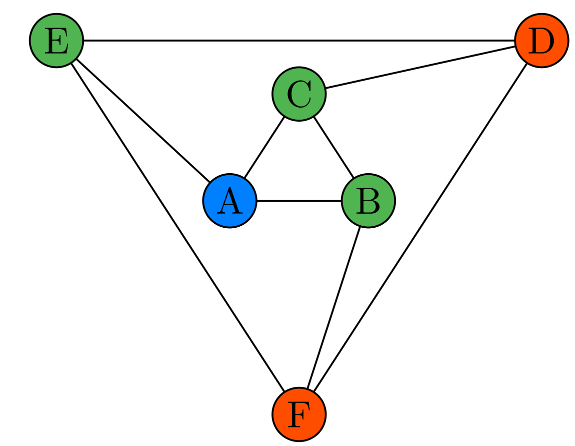
- ELP (Mannor and Shamir 2011)
  - **EXP3** - with **"LP balanced exploration"**
  - undirected $O(\sqrt{(\alpha T)})$ ✅ – needs to know G_t
  - directed case $O(\sqrt{(cT)})$ – needs to know G_t
- EXP3-SET (Alon, Cesa-Bianchi, Gentile, Mansour, 2013)

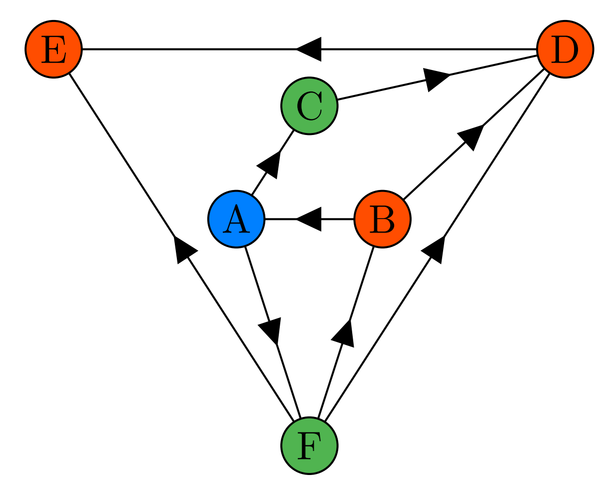




KNOWLEDGE OF OBSERVATION GRAPHS

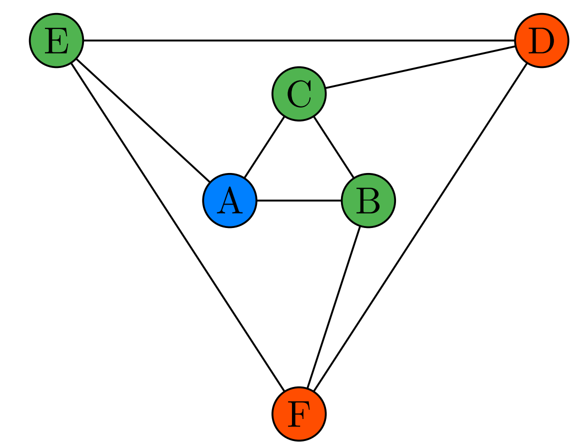
- ELP (Mannor and Shamir 2011)
  - **EXP3** - with **"LP balanced exploration"**
  - undirected $O(\sqrt{(\alpha T)})$ ✅ – needs to know G_t
  - directed case $O(\sqrt{(cT)})$ – needs to know G_t
- EXP3-SET (Alon, Cesa-Bianchi, Gentile, Mansour, 2013)
  - undirected $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅

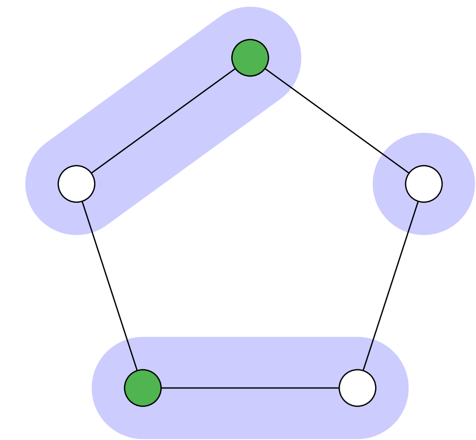


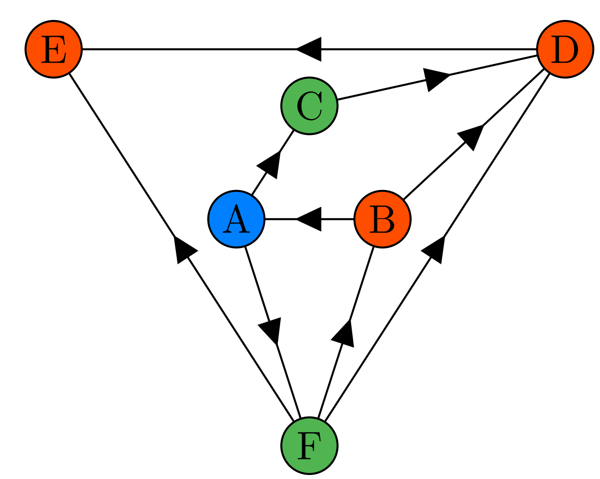


KNOWLEDGE OF OBSERVATION GRAPHS

- ELP (Mannor and Shamir 2011)
  - **EXP3** - with **"LP balanced exploration"**
  - undirected $O(\sqrt{(\alpha T)})$ ✅ – needs to know G_t
  - directed case $O(\sqrt{(cT)})$ – needs to know G_t
- EXP3-SET (Alon, Cesa-Bianchi, Gentile, Mansour, 2013)
  - undirected $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
- EXP3-DOM (Alon, Cesa-Bianchi, Gentile, Mansour, 2013)

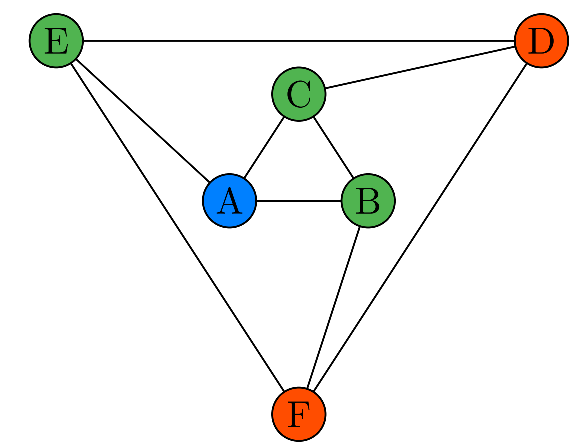


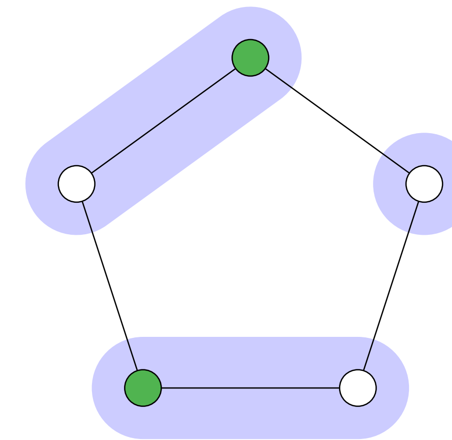


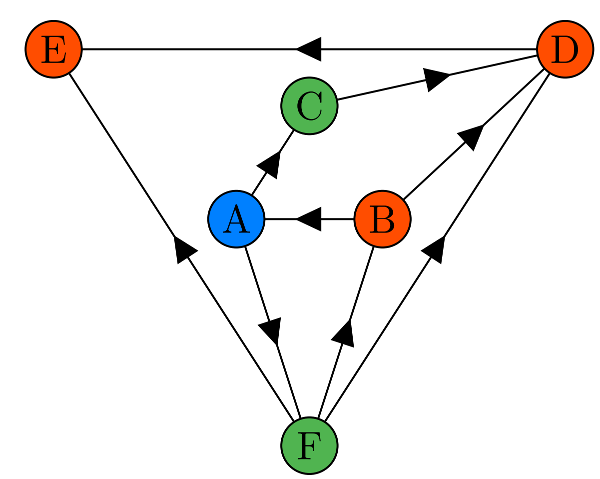


KNOWLEDGE OF OBSERVATION GRAPHS

- ELP (Mannor and Shamir 2011)
  - **EXP3** - with **"LP balanced exploration"**
  - undirected $O(\sqrt{(\alpha T)})$ ✅ – needs to know G_t
  - directed case $O(\sqrt{(cT)})$ – needs to know G_t
- EXP3-SET (Alon, Cesa-Bianchi, Gentile, Mansour, 2013)
  - undirected $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
- EXP3-DOM (Alon, Cesa-Bianchi, Gentile, Mansour, 2013)
  - directed $O(\sqrt{(\alpha T)})$ ✅ – need to know G_t

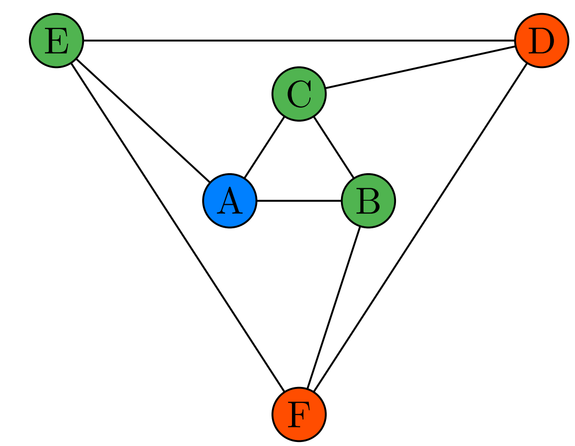


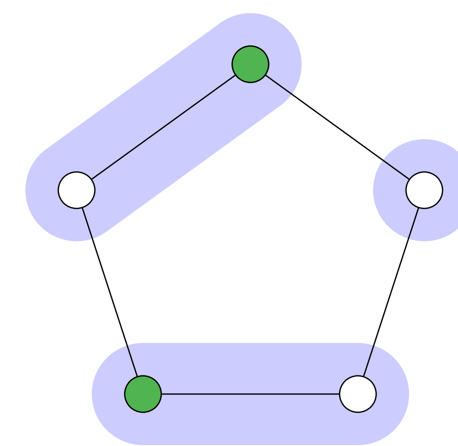


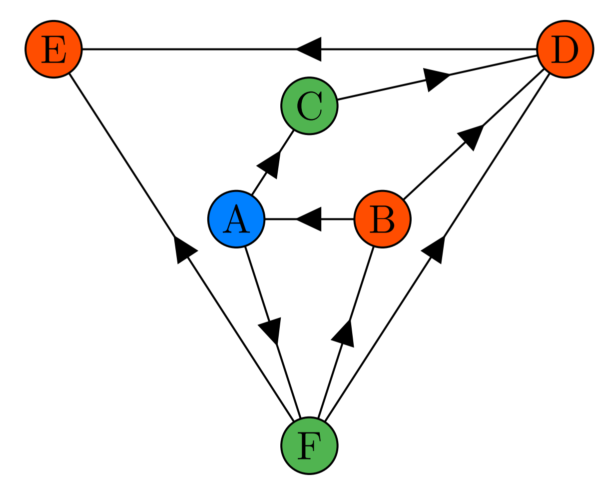


KNOWLEDGE OF OBSERVATION GRAPHS

- ELP (Mannor and Shamir 2011)
  - **EXP3** - with **"LP balanced exploration"**
  - undirected $O(\sqrt{(\alpha T)})$ ✅ – needs to know G_t
  - directed case $O(\sqrt{(cT)})$ – needs to know G_t
- EXP3-SET (Alon, Cesa-Bianchi, Gentile, Mansour, 2013)
  - undirected $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
- EXP3-DOM (Alon, Cesa-Bianchi, Gentile, Mansour, 2013)
  - directed $O(\sqrt{(\alpha T)})$ ✅ – need to know G_t
  - **calculates dominating set**







EXP3-IX: IMPLICIT EXPLORATION

Algorithm 1 EXP3-IX

Input: Set of actions $\mathcal{S} = [d]$,
    parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
    $w_{t,i} \leftarrow (1/d)\exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [d]$
    An adversary privately chooses losses $\ell_{t,i}$ for $i \in [d]$ and generates a graph G_t
    $W_t \leftarrow \sum_{i=1}^{d} w_{t,i}$
    $p_{t,i} \leftarrow w_{t,i}/W_t$
    Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,d})$
    Observe graph G_t
    Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
    $o_{t,i} \leftarrow \sum_{(j\to i)\in G_t} p_{t,j}$ for $i \in [d]$
    $\hat{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{o_{t,i}+\gamma_t} \mathbb{1}_{\{(I_t \to i)\in G_t\}}$ for $i \in [d]$
end for

EXP3-IX: IMPLICIT EXPLORATION

Algorithm 1 EXP3-IX

Input: Set of actions $\mathcal{S} = [d]$,
  parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
  $w_{t,i} \leftarrow (1/d) \exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [d]$
  An adversary privately chooses losses $\ell_{t,i}$ for $i \in [d]$ and generates a graph G_t
  $W_t \leftarrow \sum_{i=1}^{d} w_{t,i}$
  $p_{t,i} \leftarrow w_{t,i}/W_t$
  Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,d})$
  Observe graph G_t
  Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
  $o_{t,i} \leftarrow \sum_{(j \to i) \in G_t} p_{t,j}$ for $i \in [d]$
  $\hat{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{o_{t,i} + \gamma_t} \mathbb{1}_{\{(I_t \to i) \in G_t\}}$ for $i \in [d]$
end for

Optimistic bias for the loss estimates

EXP3-IX: IMPLICIT EXPLORATION

Algorithm 1 EXP3-IX

Input: Set of actions $\mathcal{S} = [d]$,
parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
$w_{t,i} \leftarrow (1/d) \exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [d]$
An adversary privately chooses losses $\ell_{t,i}$ for $i \in [d]$ and generates a graph G_t
$W_t \leftarrow \sum_{i=1}^{d} w_{t,i}$
$p_{t,i} \leftarrow w_{t,i}/W_t$
Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \dots, p_{t,d})$
Observe graph G_t
Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
$o_{t,i} \leftarrow \sum_{(j \to i) \in G_t} p_{t,j}$ for $i \in [d]$
$\hat{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{o_{t,i} + \gamma_t} \mathbb{1}_{\{(I_t \to i) \in G_t\}}$ for $i \in [d]$
end for

Optimistic bias for the loss estimates

$$\mathbb{E}[\hat{\ell}_{t,i}] = \frac{\ell_{t,i}}{o_{t,i} + \gamma} o_{t,i} + 0(1 - o_{t,i}) = \ell_{t,i} - \ell_{t,i} \frac{\gamma}{o_{t,i} + \gamma} \leq \ell_{t,i}$$

EXP3-IX: IMPLICIT EXPLORATION

Algorithm 1 EXP3-IX

Input: Set of actions $\mathcal{S} = [d]$,
    parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
    $w_{t,i} \leftarrow (1/d) \exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [d]$
    An adversary privately chooses losses $\ell_{t,i}$ for $i \in [d]$ and generates a graph G_t
    $W_t \leftarrow \sum_{i=1}^{d} w_{t,i}$
    $p_{t,i} \leftarrow w_{t,i}/W_t$
    Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,d})$
    Observe graph G_t
    Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
    $o_{t,i} \leftarrow \sum_{(j \to i) \in G_t} p_{t,j}$ for $i \in [d]$
    $\hat{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{o_{t,i} + \gamma_t} \mathbb{1}_{\{(I_t \to i) \in G_t\}}$ for $i \in [d]$
end for

Benefits of the **implicit exploration**

Optimistic bias for the loss estimates

$$\mathbb{E}[\hat{\ell}_{t,i}] = \frac{\ell_{t,i}}{o_{t,i} + \gamma} o_{t,i} + 0(1 - o_{t,i}) = \ell_{t,i} - \ell_{t,i} \frac{\gamma}{o_{t,i} + \gamma} \leq \ell_{t,i}$$

EXP3-IX: IMPLICIT EXPLORATION

Algorithm 1 EXP3-IX

Input: Set of actions $\mathcal{S} = [d]$,
    parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
    $w_{t,i} \leftarrow (1/d) \exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [d]$
    An adversary privately chooses losses $\ell_{t,i}$ for $i \in [d]$ and generates a graph G_t
    $W_t \leftarrow \sum_{i=1}^{d} w_{t,i}$
    $p_{t,i} \leftarrow w_{t,i}/W_t$
    Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,d})$
    Observe graph G_t
    Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
    $o_{t,i} \leftarrow \sum_{(j \to i) \in G_t} p_{t,j}$ for $i \in [d]$
    $\hat{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{o_{t,i}+\gamma_t} \mathbb{1}_{\{(I_t \to i) \in G_t\}}$ for $i \in [d]$
end for

Benefits of the **implicit exploration**

- **no need to know the graph before**

Optimistic bias for the loss estimates

$$\mathbb{E}[\hat{\ell}_{t,i}] = \frac{\ell_{t,i}}{o_{t,i} + \gamma} o_{t,i} + 0(1 - o_{t,i}) = \ell_{t,i} - \ell_{t,i} \frac{\gamma}{o_{t,i} + \gamma} \leq \ell_{t,i}$$

EXP3-IX: IMPLICIT EXPLORATION

Algorithm 1 EXP3-IX

Input: Set of actions $\mathcal{S} = [d]$,
  parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
  $w_{t,i} \leftarrow (1/d) \exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [d]$
  An adversary privately chooses losses $\ell_{t,i}$ for $i \in [d]$ and generates a graph G_t
  $W_t \leftarrow \sum_{i=1}^{d} w_{t,i}$
  $p_{t,i} \leftarrow w_{t,i} / W_t$
  Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,d})$
  Observe graph G_t
  Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
  $o_{t,i} \leftarrow \sum_{(j \to i) \in G_t} p_{t,j}$ for $i \in [d]$
  $\hat{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{o_{t,i} + \gamma_t} \mathbb{1}_{\{(I_t \to i) \in G_t\}}$ for $i \in [d]$
end for

Benefits of the **implicit exploration**

- **no need to know the graph before**
- no need to estimate dominating set

Optimistic bias for the loss estimates

$$\mathbb{E}[\hat{\ell}_{t,i}] = \frac{\ell_{t,i}}{o_{t,i} + \gamma} o_{t,i} + 0(1 - o_{t,i}) = \ell_{t,i} - \ell_{t,i} \frac{\gamma}{o_{t,i} + \gamma} \leq \ell_{t,i}$$

EXP3-IX: IMPLICIT EXPLORATION

Algorithm 1 EXP3-IX

Input: Set of actions $\mathcal{S} = [d]$,
    parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
    $w_{t,i} \leftarrow (1/d)\exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [d]$
    An adversary privately chooses losses $\ell_{t,i}$ for $i \in [d]$ and generates a graph G_t
    $W_t \leftarrow \sum_{i=1}^{d} w_{t,i}$
    $p_{t,i} \leftarrow w_{t,i}/W_t$
    Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,d})$
    Observe graph G_t
    Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
    $o_{t,i} \leftarrow \sum_{(j \to i) \in G_t} p_{t,j}$ for $i \in [d]$
    $\hat{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{o_{t,i}+\gamma_t} \mathbb{1}_{\{(I_t \to i) \in G_t\}}$ for $i \in [d]$
end for

Benefits of the **implicit exploration**

- **no need to know the graph before**
- no need to estimate dominating set
- no need for doubling trick

Optimistic bias for the loss estimates

$$\mathbb{E}[\hat{\ell}_{t,i}] = \frac{\ell_{t,i}}{o_{t,i}+\gamma} o_{t,i} + 0(1 - o_{t,i}) = \ell_{t,i} - \ell_{t,i}\frac{\gamma}{o_{t,i}+\gamma} \leq \ell_{t,i}$$

EXP3-IX: IMPLICIT EXPLORATION

Algorithm 1 EXP3-IX

Input: Set of actions $\mathcal{S} = [d]$,
    parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
    $w_{t,i} \leftarrow (1/d)\exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [d]$
    An adversary privately chooses losses $\ell_{t,i}$ for $i \in [d]$ and generates a graph G_t
    $W_t \leftarrow \sum_{i=1}^{d} w_{t,i}$
    $p_{t,i} \leftarrow w_{t,i}/W_t$
    Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,d})$
    Observe graph G_t
    Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
    $o_{t,i} \leftarrow \sum_{(j \to i) \in G_t} p_{t,j}$ for $i \in [d]$
    $\hat{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{o_{t,i} + \gamma_t} \mathbb{1}_{\{(I_t \to i) \in G_t\}}$ for $i \in [d]$
end for

Benefits of the **implicit exploration**

- **no need to know the graph before**
- no need to estimate dominating set
- no need for doubling trick
- no need for aggregation

Optimistic bias for the loss estimates

$$\mathbb{E}[\hat{\ell}_{t,i}] = \frac{\ell_{t,i}}{o_{t,i} + \gamma} o_{t,i} + 0(1 - o_{t,i}) = \ell_{t,i} - \ell_{t,i} \frac{\gamma}{o_{t,i} + \gamma} \leq \ell_{t,i}$$

EXP3-IX: IMPLICIT EXPLORATION

Algorithm 1 EXP3-IX

Input: Set of actions $\mathcal{S} = [d]$,
  parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
  $w_{t,i} \leftarrow (1/d)\exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [d]$
  An adversary privately chooses losses $\ell_{t,i}$ for $i \in [d]$ and generates a graph G_t
  $W_t \leftarrow \sum_{i=1}^{d} w_{t,i}$
  $p_{t,i} \leftarrow w_{t,i}/W_t$
  Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,d})$
  Observe graph G_t
  Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
  $o_{t,i} \leftarrow \sum_{(j \to i) \in G_t} p_{t,j}$ for $i \in [d]$
  $\hat{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{o_{t,i} + \gamma_t} \mathbb{1}_{\{(I_t \to i) \in G_t\}}$ for $i \in [d]$
end for

Benefits of the **implicit exploration**

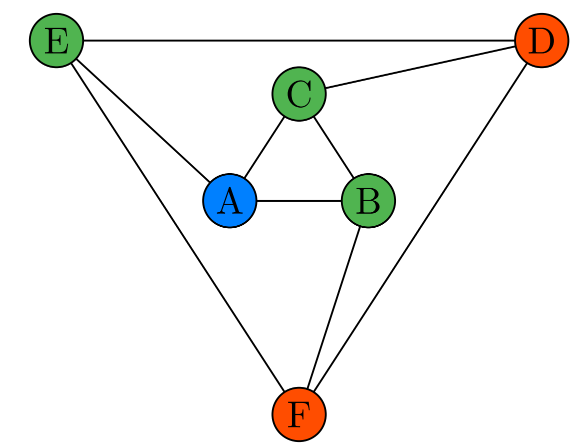
- **no need to know the graph before**
- no need to estimate dominating set
- no need for doubling trick
- no need for aggregation

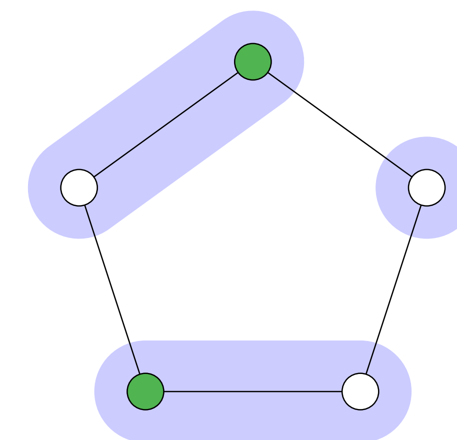
$$R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha T \ln N}\right)$$

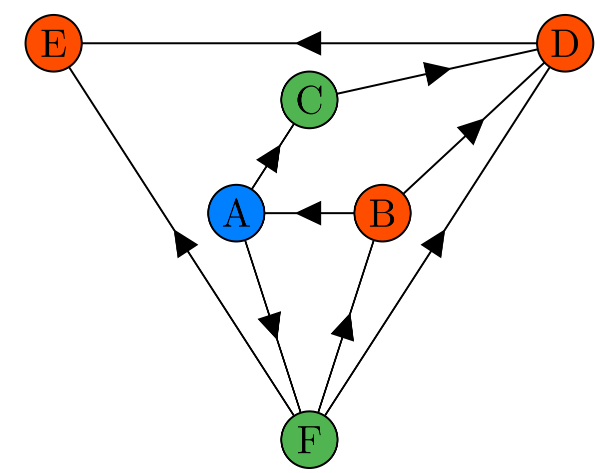
Optimistic bias for the loss estimates

$$\mathbb{E}[\hat{\ell}_{t,i}] = \frac{\ell_{t,i}}{o_{t,i} + \gamma} o_{t,i} + 0(1 - o_{t,i}) = \ell_{t,i} - \ell_{t,i}\frac{\gamma}{o_{t,i} + \gamma} \leq \ell_{t,i}$$

FOLLOW UPS

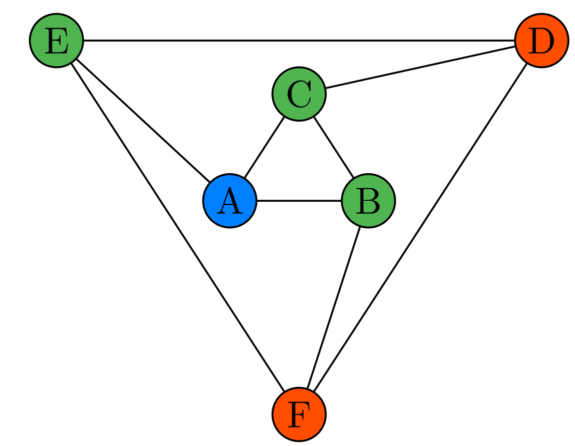


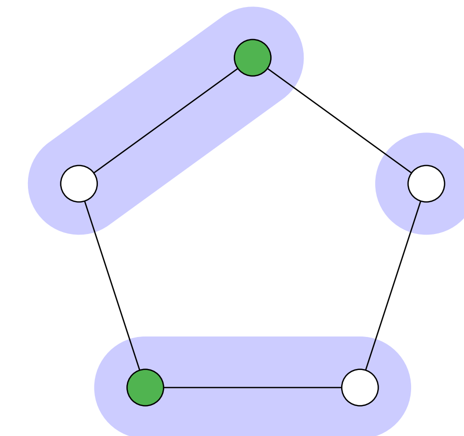


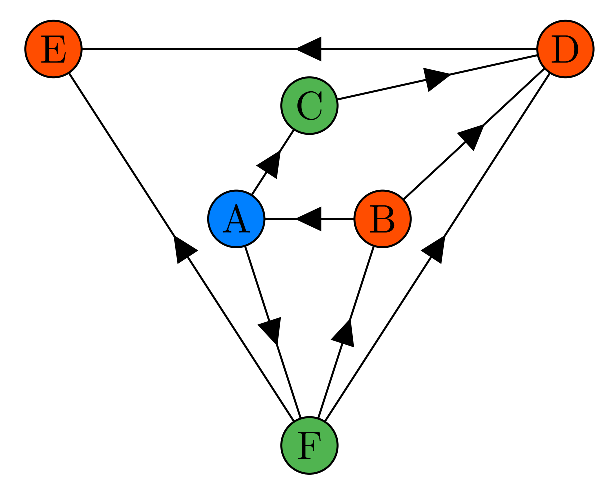


FOLLOW UPS

- EXP3-IX (Kocák, Neu, MV, Munos, 2014)

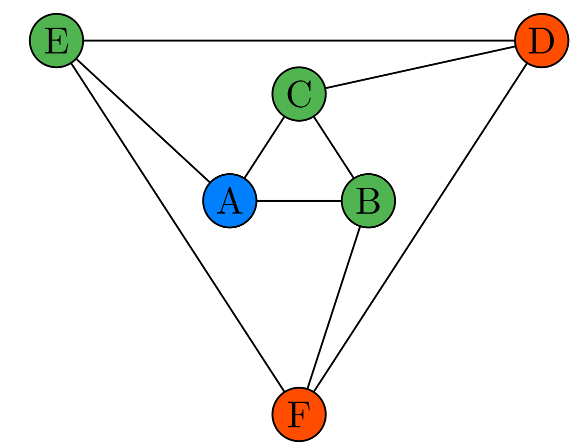


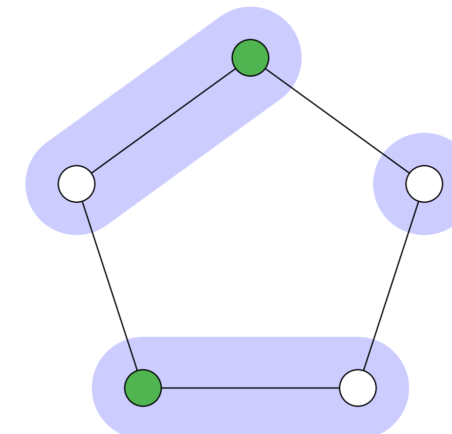


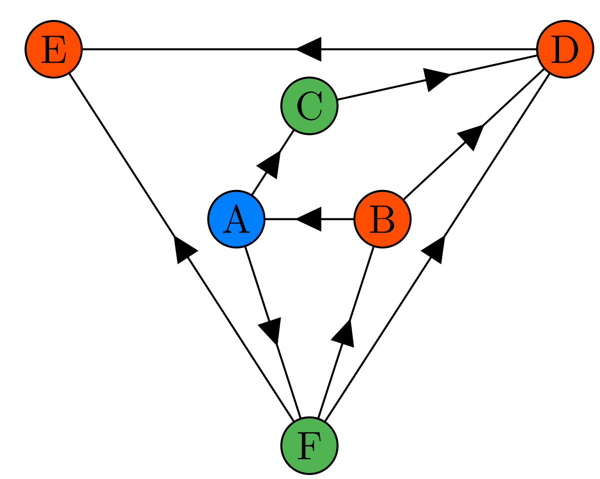


FOLLOW UPS

- EXP3-IX (Kocák, Neu, MV, Munos, 2014)
    - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅

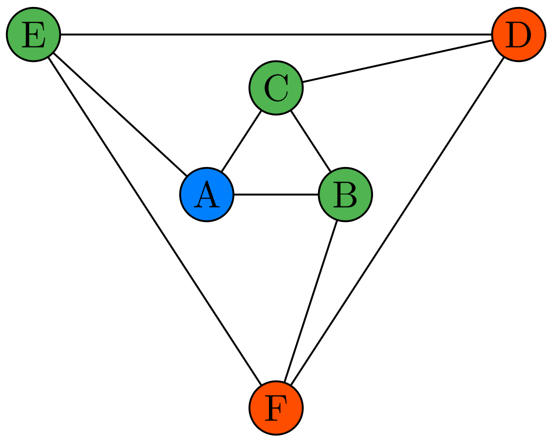


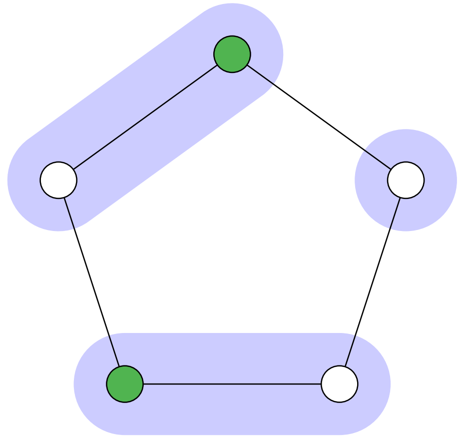


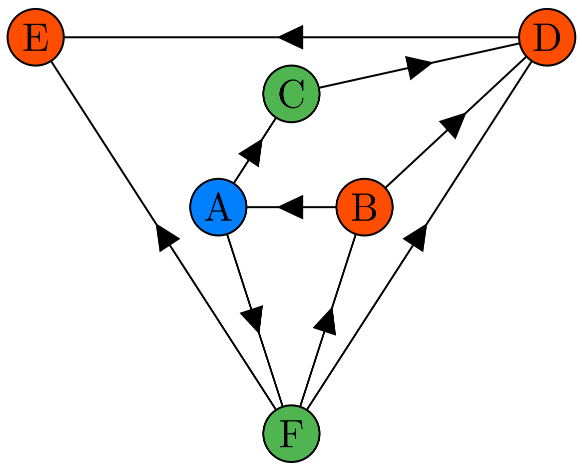


FOLLOW UPS

- EXP3-IX (Kocák, Neu, MV, Munos, 2014)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
- EXP3.G (Alon, Cesa-Bianchi, Dekel, Koren, 2015)

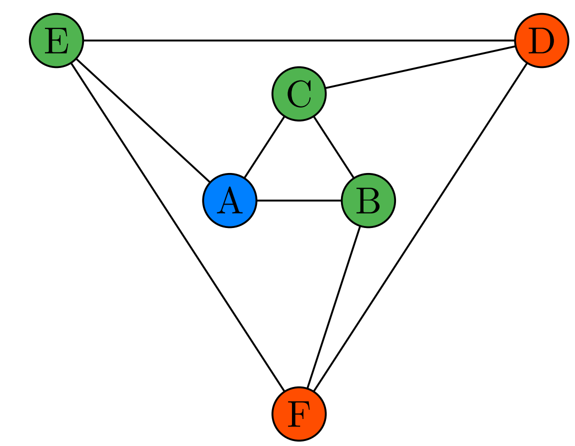


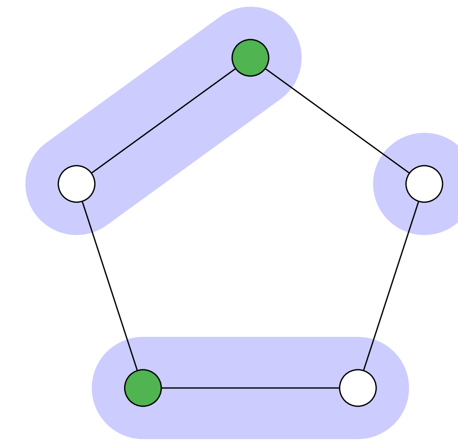


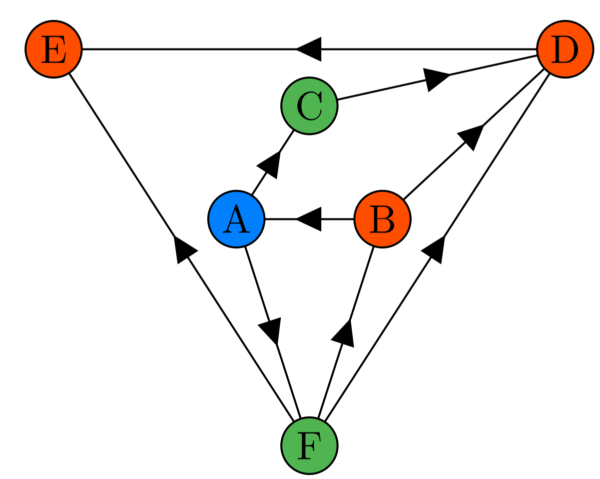


FOLLOW UPS

- EXP3-IX (Kocák, Neu, MV, Munos, 2014)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
- EXP3.G (Alon, Cesa-Bianchi, Dekel, Koren, 2015)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅

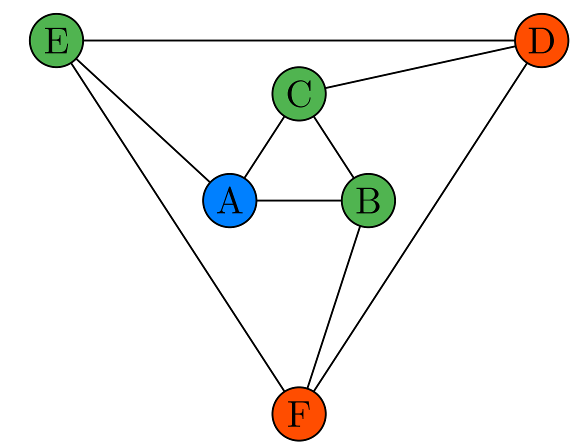


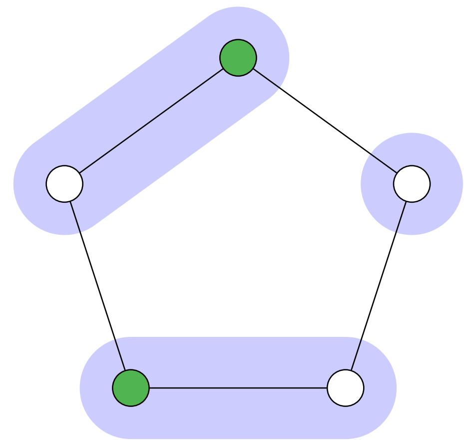


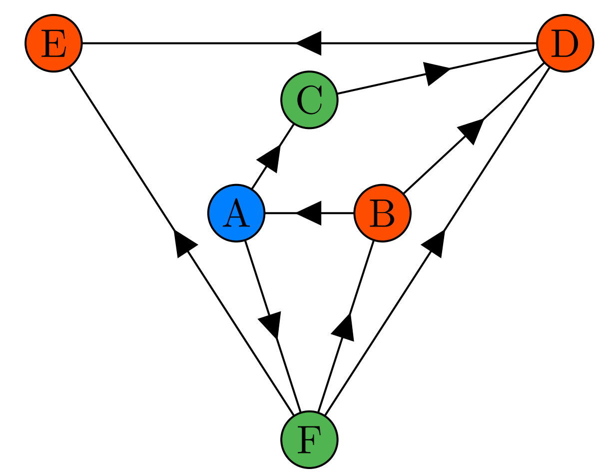


FOLLOW UPS

- EXP3-IX (Kocák, Neu, MV, Munos, 2014)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
- EXP3.G (Alon, Cesa-Bianchi, Dekel, Koren, 2015)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
  - mixes uniform distribution

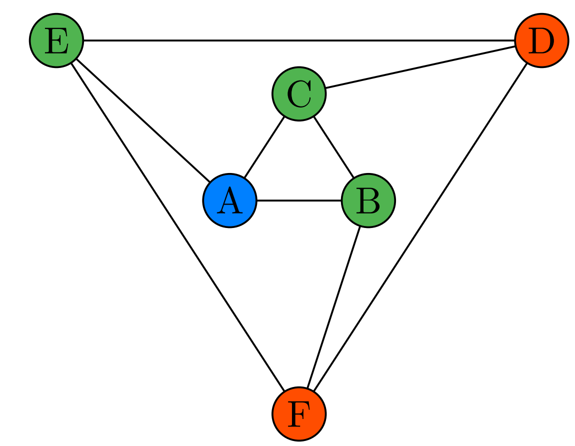


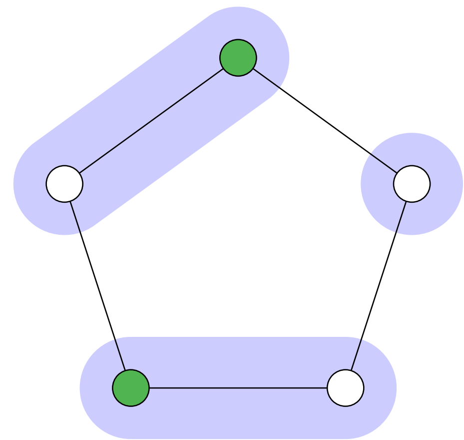


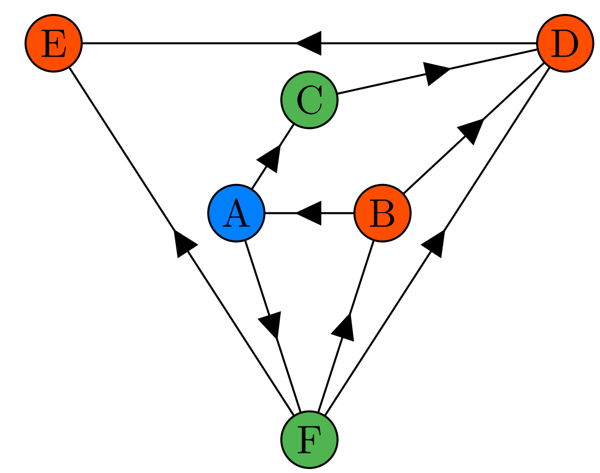


FOLLOW UPS

- EXP3-IX (Kocák, Neu, MV, Munos, 2014)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
- EXP3.G (Alon, Cesa-Bianchi, Dekel, Koren, 2015)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
  - mixes uniform distribution
  - more general algorithm for settings **beyond bandits**

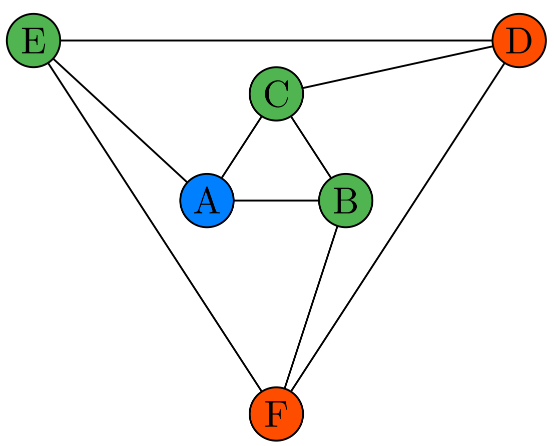


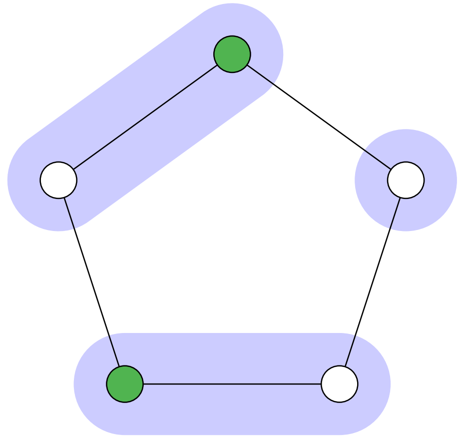


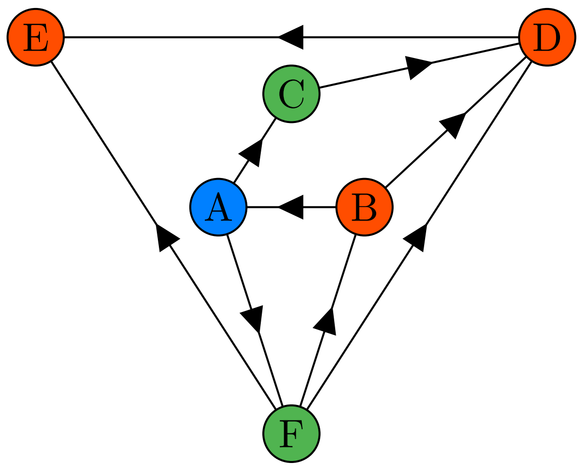


FOLLOW UPS

- EXP3-IX (Kocák, Neu, MV, Munos, 2014)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
- EXP3.G (Alon, Cesa-Bianchi, Dekel, Koren, 2015)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
  - mixes uniform distribution
  - more general algorithm for settings **beyond bandits**
  - high-probability bound

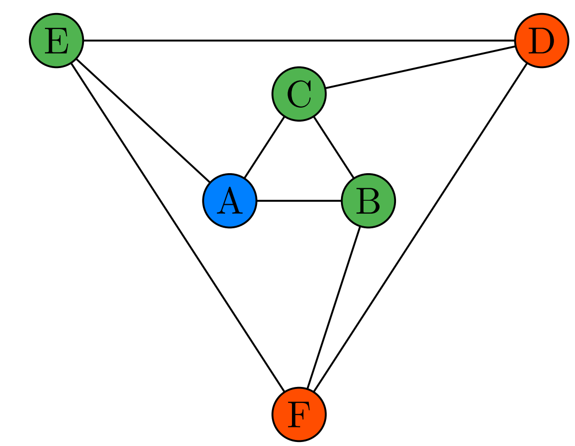


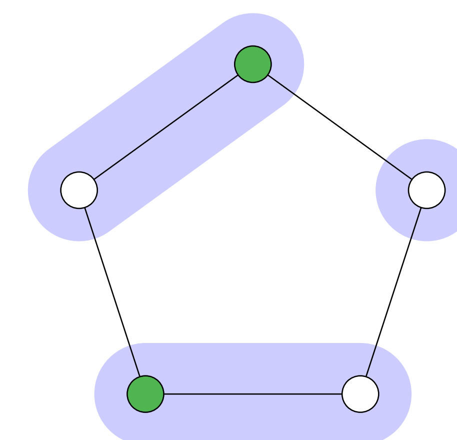


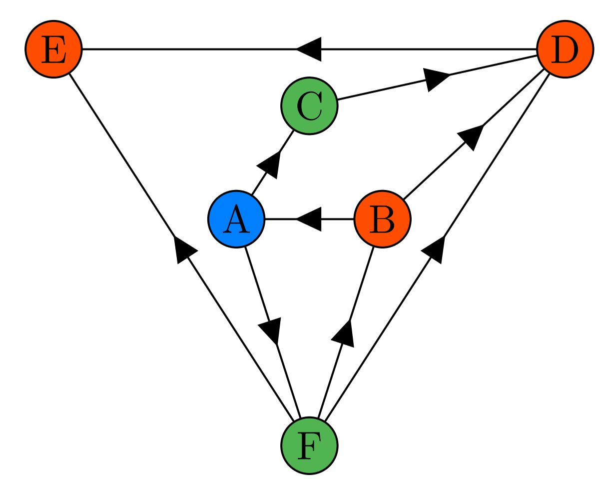


FOLLOW UPS

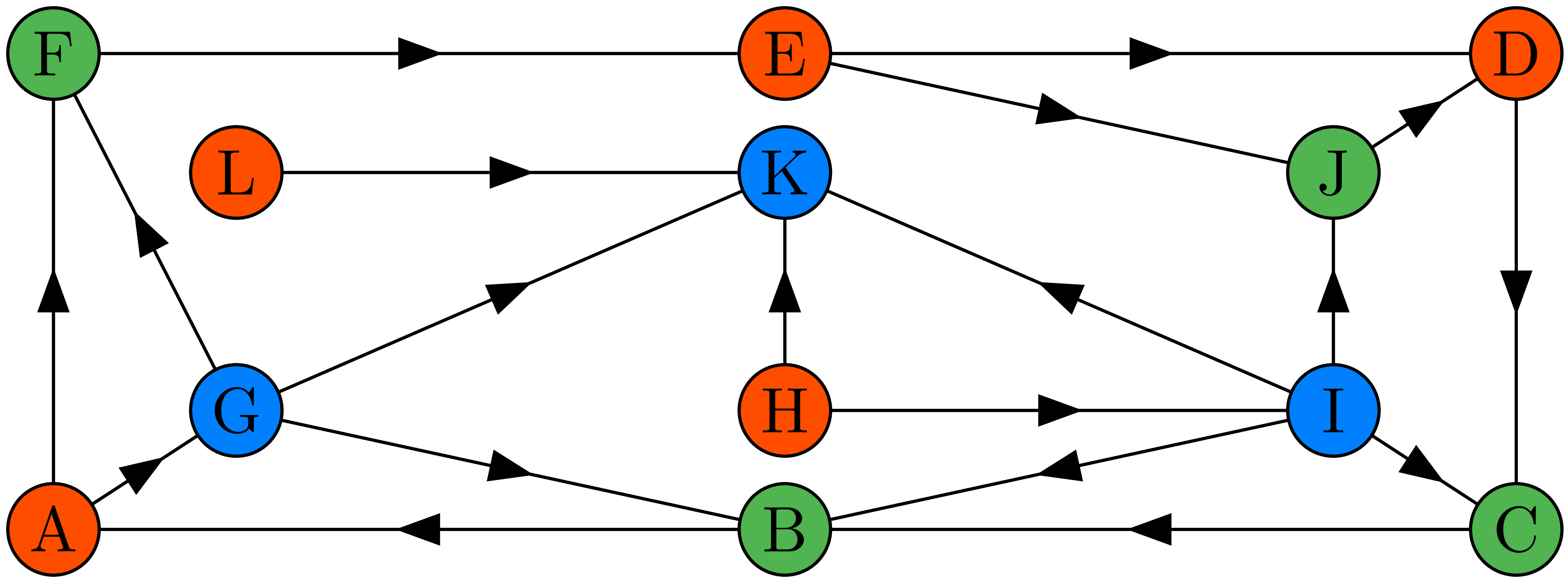
- EXP3-IX (Kocák, Neu, MV, Munos, 2014)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
- EXP3.G (Alon, Cesa-Bianchi, Dekel, Koren, 2015)
  - directed $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
  - mixes uniform distribution
  - more general algorithm for settings **beyond bandits**
  - high-probability bound
- Neu 2015: high-probability bound for EXP3-IX





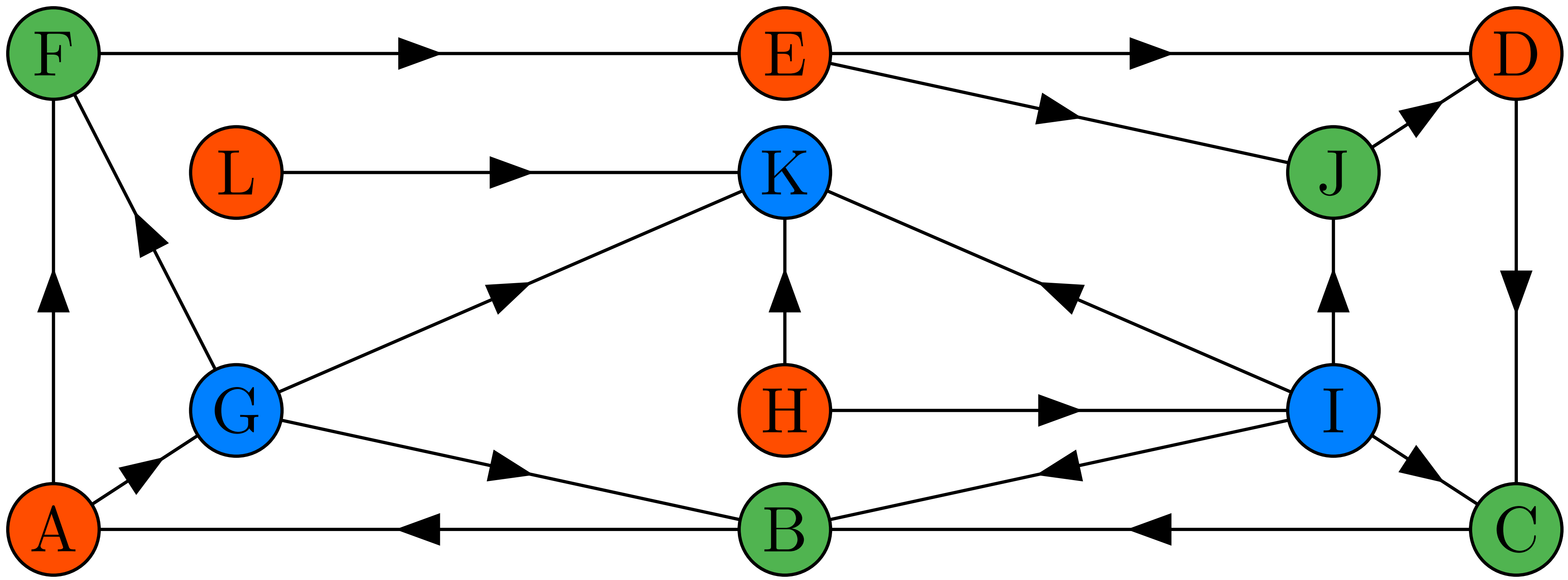


COMPLEX GRAPH ACTIONS



- Play action $\mathbf{V}_t \in S \subset \{0,1\}^N$, $\|\mathbf{v}\|_1 \leq m$ from all $\mathbf{v} \in S$
- Obtain losses $\mathbf{V}_t^\mathsf{T} \boldsymbol{\ell}_t$
- Observe additional losses according to the graph

COMPLEX GRAPH ACTIONS



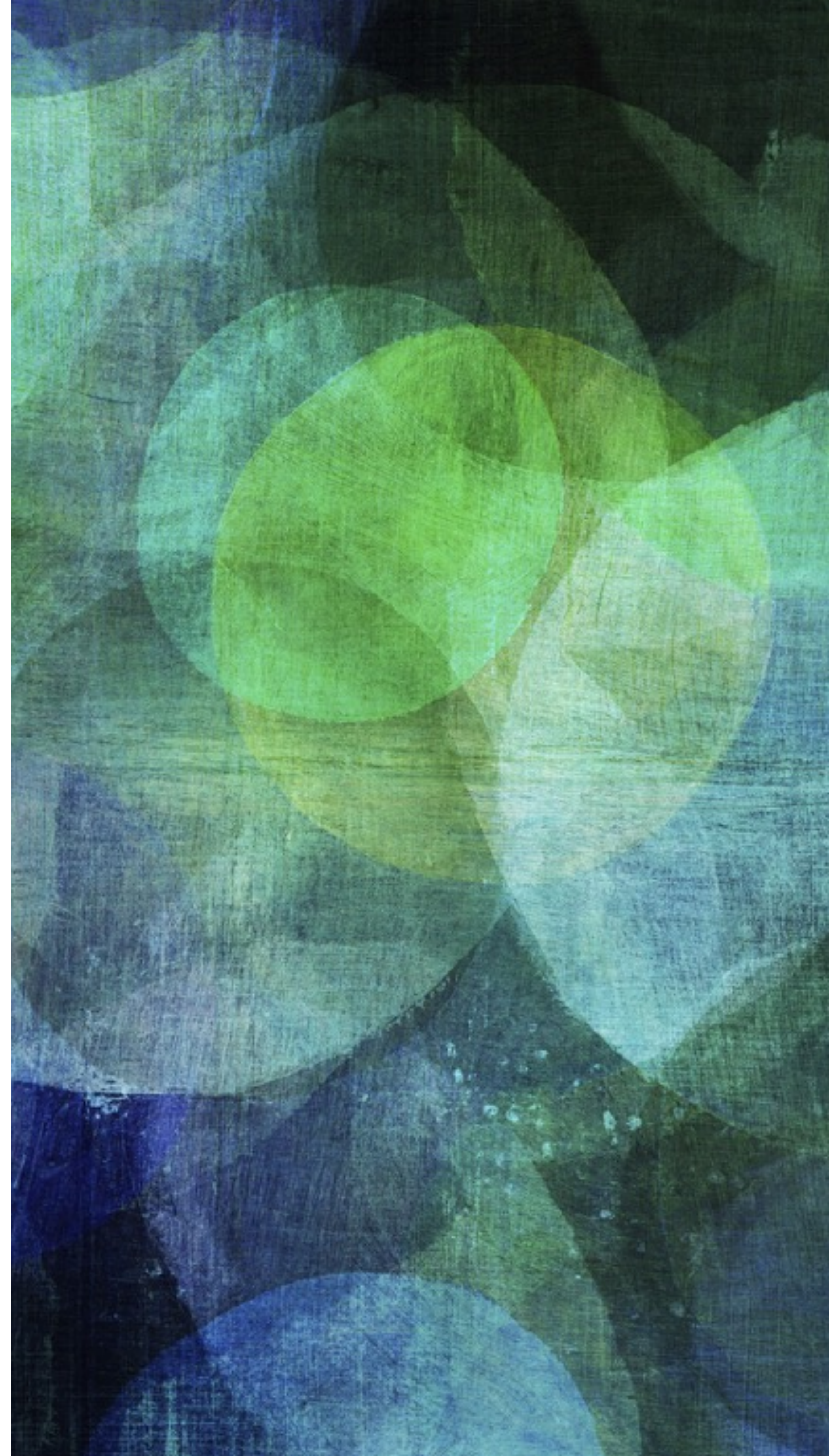
- Play action $\mathbf{V}_t \in S \subset \{0,1\}^N$, $\|\mathbf{v}\|_1 \leq m$ from all $\mathbf{v} \in S$
- Obtain losses $\mathbf{V}_t^\mathsf{T} \boldsymbol{\ell}_t$
- Observe additional losses according to the graph

$$R_T = \widetilde{\mathcal{O}}\left(m^{3/2}\sqrt{\sum_{t=1}^{T}\alpha_t}\right) = \widetilde{\mathcal{O}}\left(m^{3/2}\sqrt{\alpha}T\right)$$

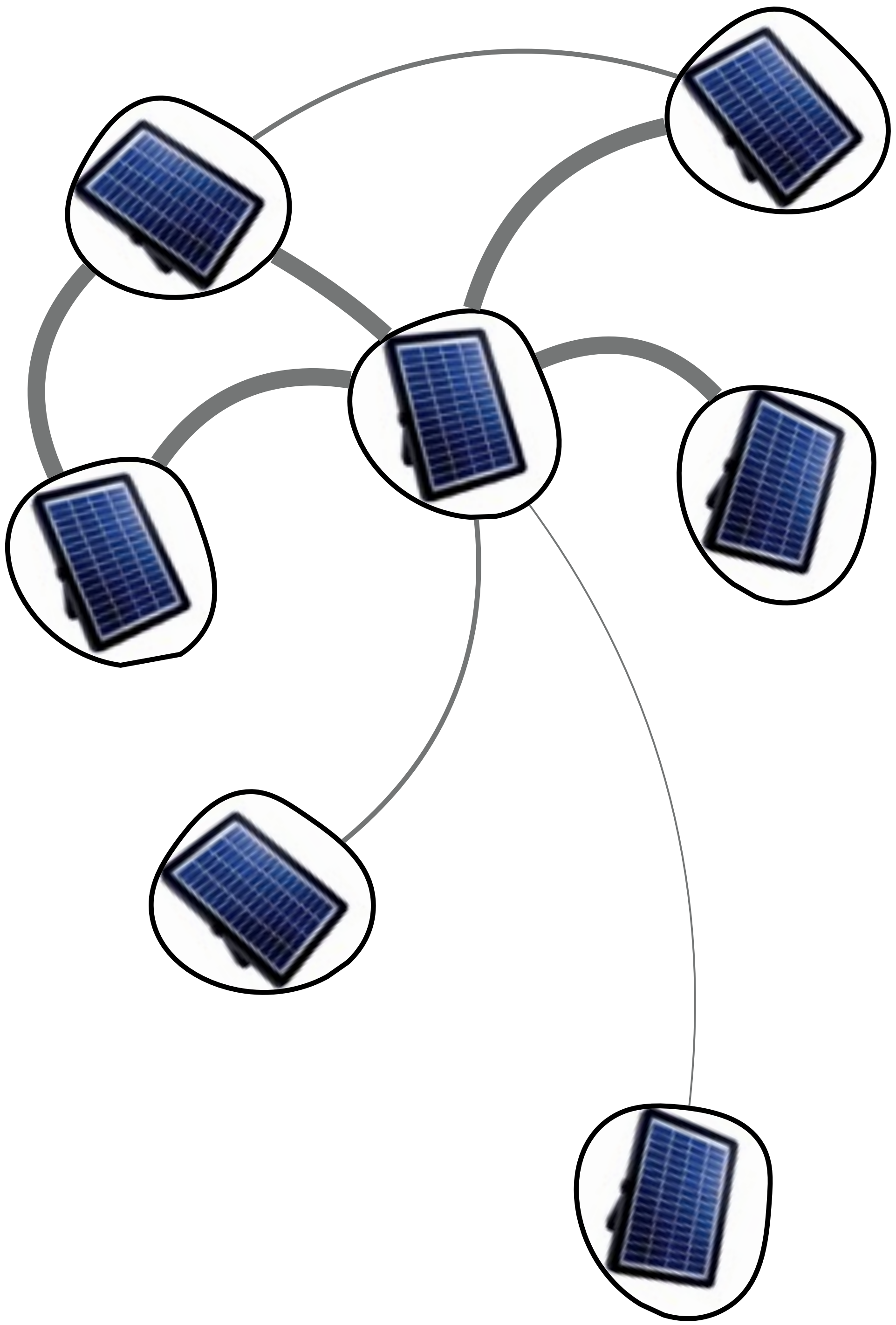
Kocák, Neu, MV: **Online learning with noisy side observations**, AISTATS 2016

GRAPH BANDITS WITH NOISY SIDE OBSERVATIONS

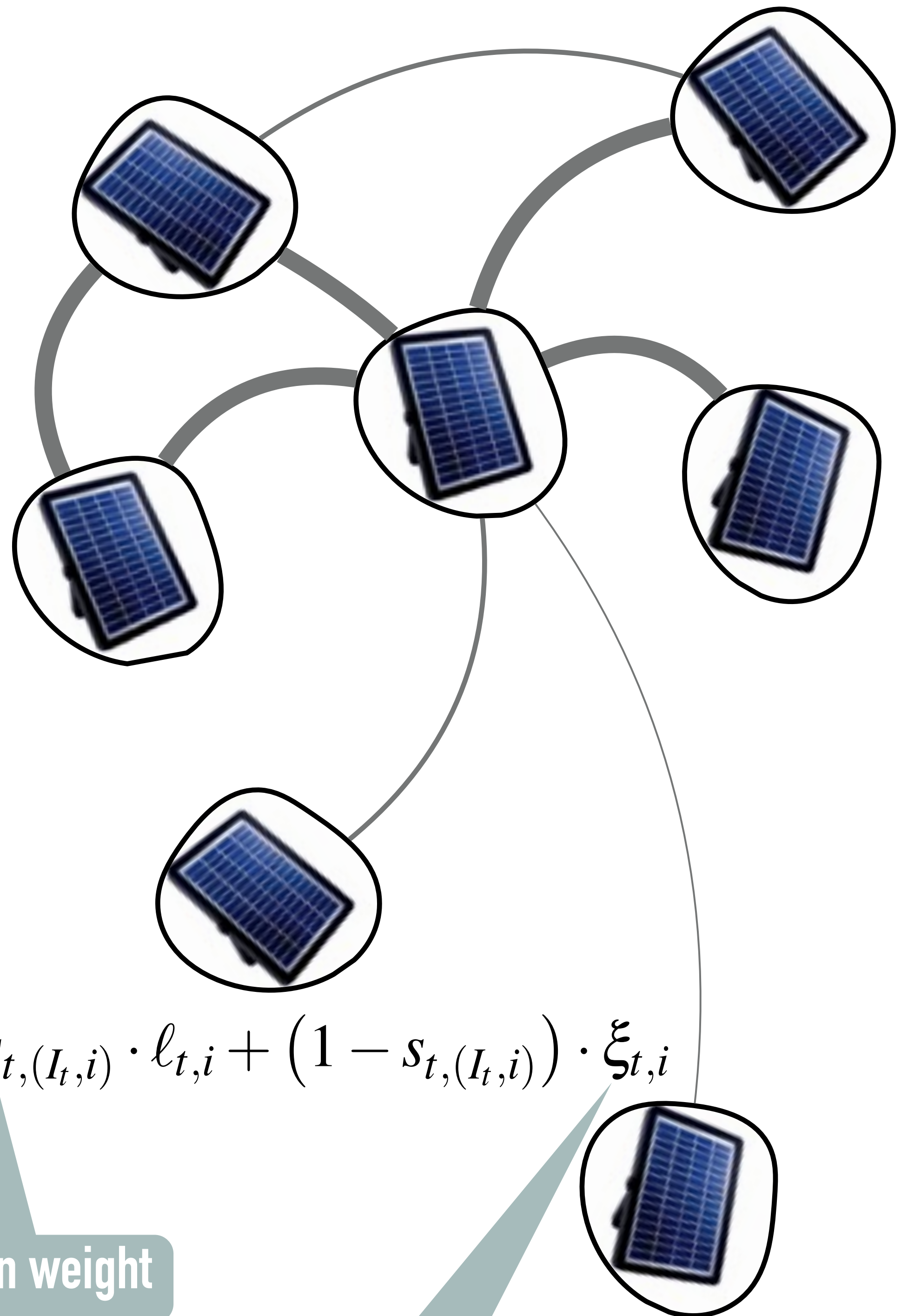
exploiting side observations that can be perturbed by certain level of noise



NOISY SIDE OBSERVATIONS



NOISY SIDE OBSERVATIONS

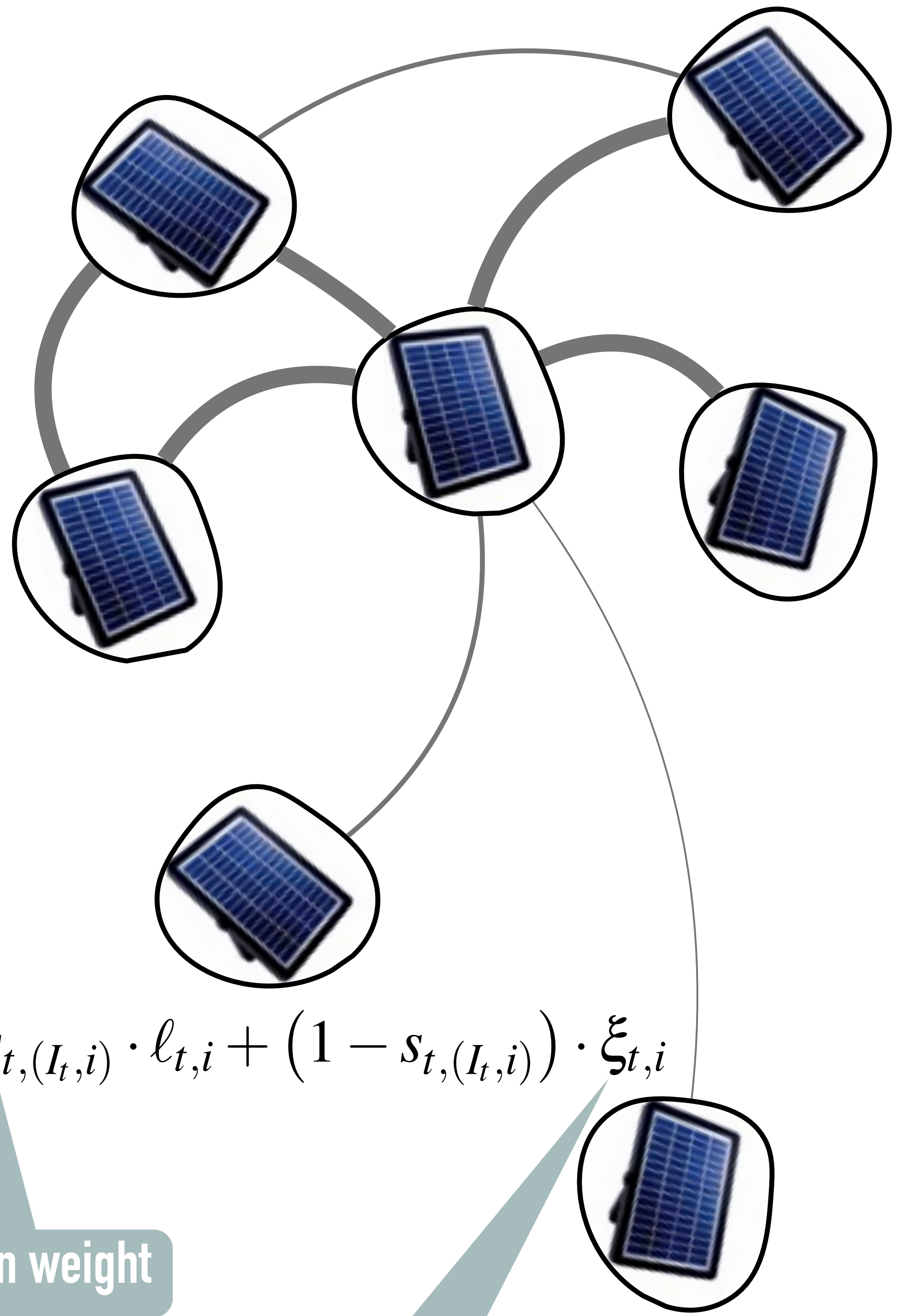


$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

known weight

zero-mean noise

NOISY SIDE OBSERVATIONS



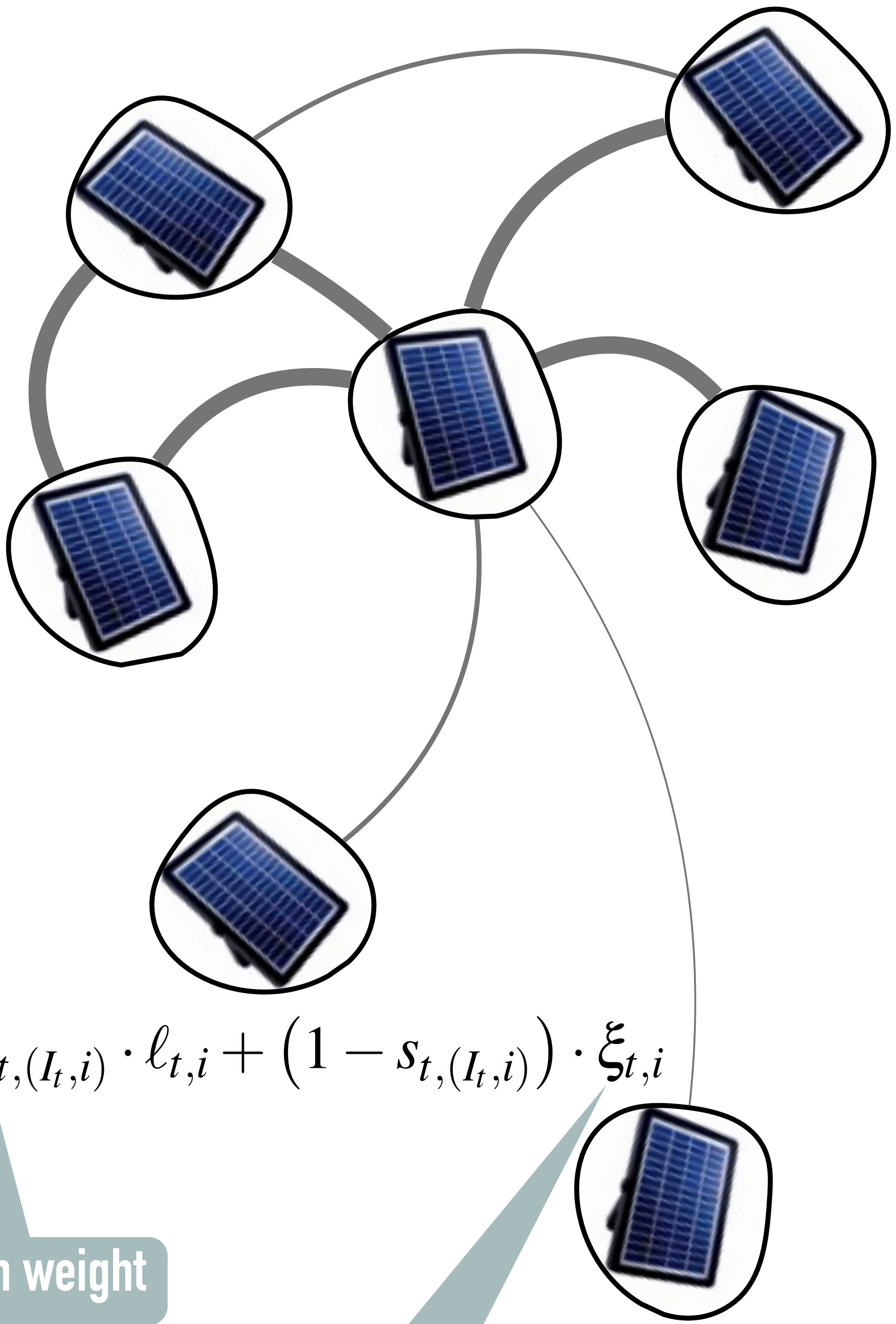
Want: only reliable information!

$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

known weight

zero-mean noise

NOISY SIDE OBSERVATIONS



$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

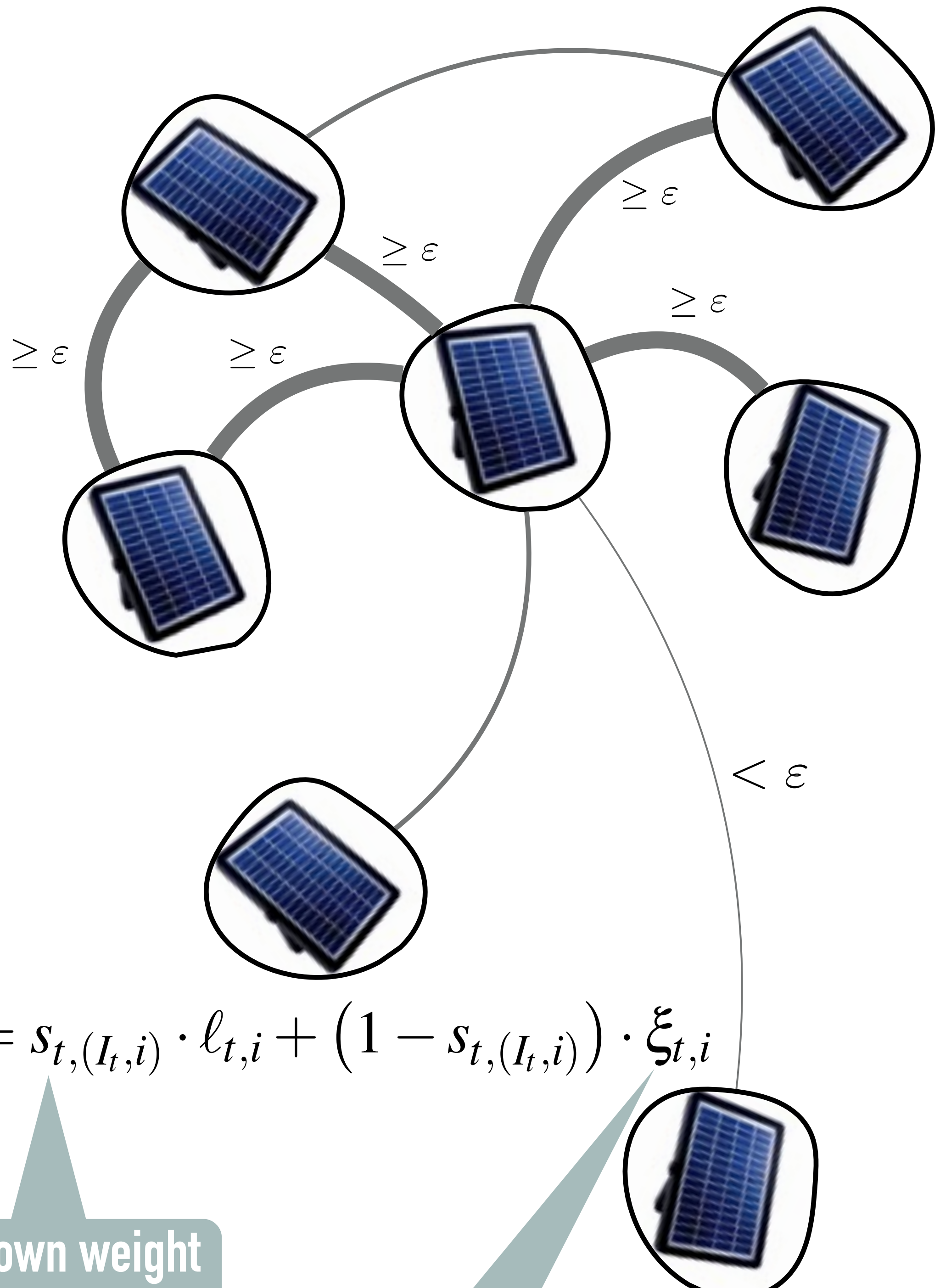
known weight

zero-mean noise

Want: only reliable information!

1) If we know the perfect cutoff ε

NOISY SIDE OBSERVATIONS



$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

known weight

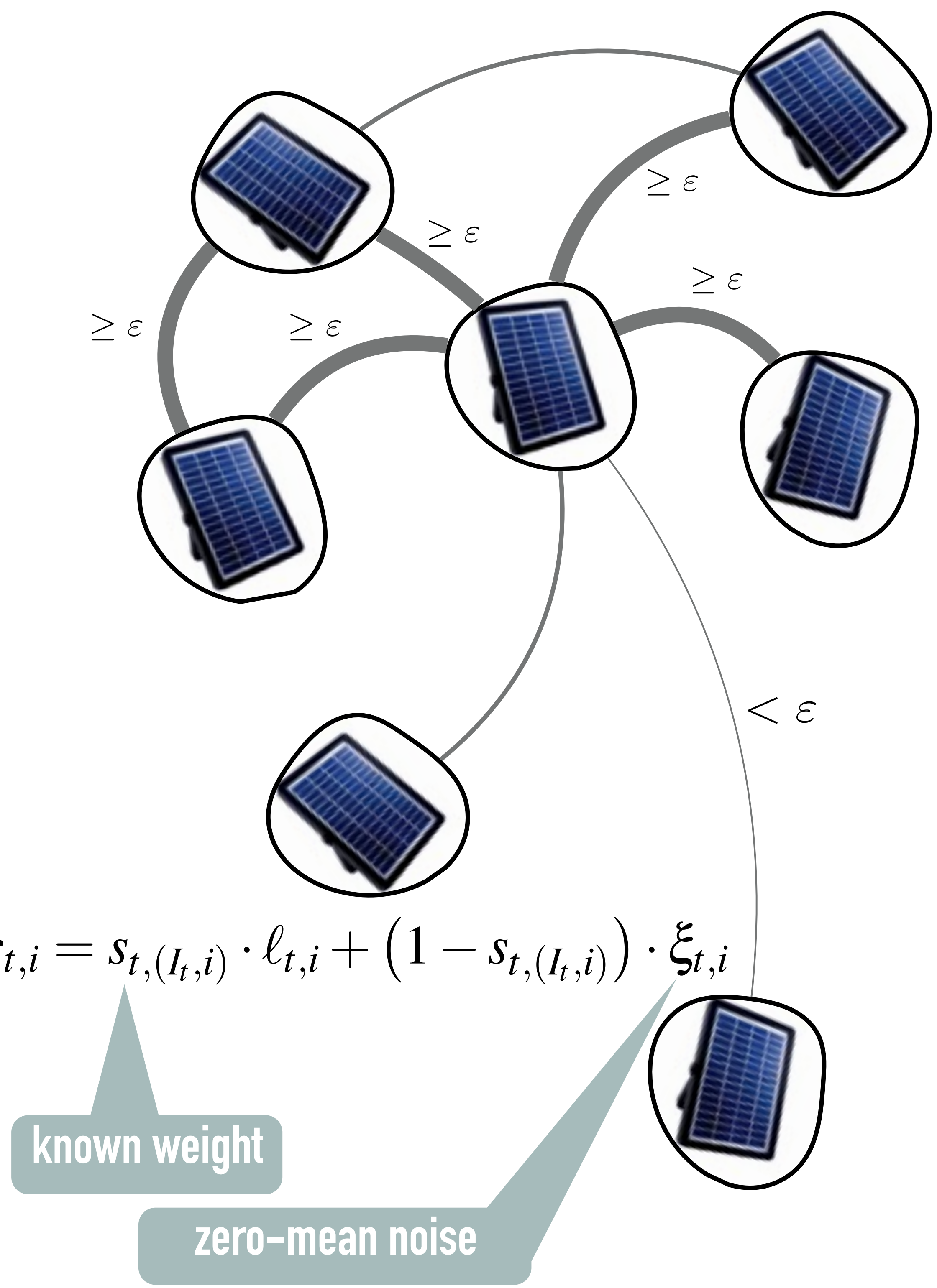
zero-mean noise

Want: only reliable information!

1) If we know the perfect cutoff ε

- reliable: use as exact

NOISY SIDE OBSERVATIONS



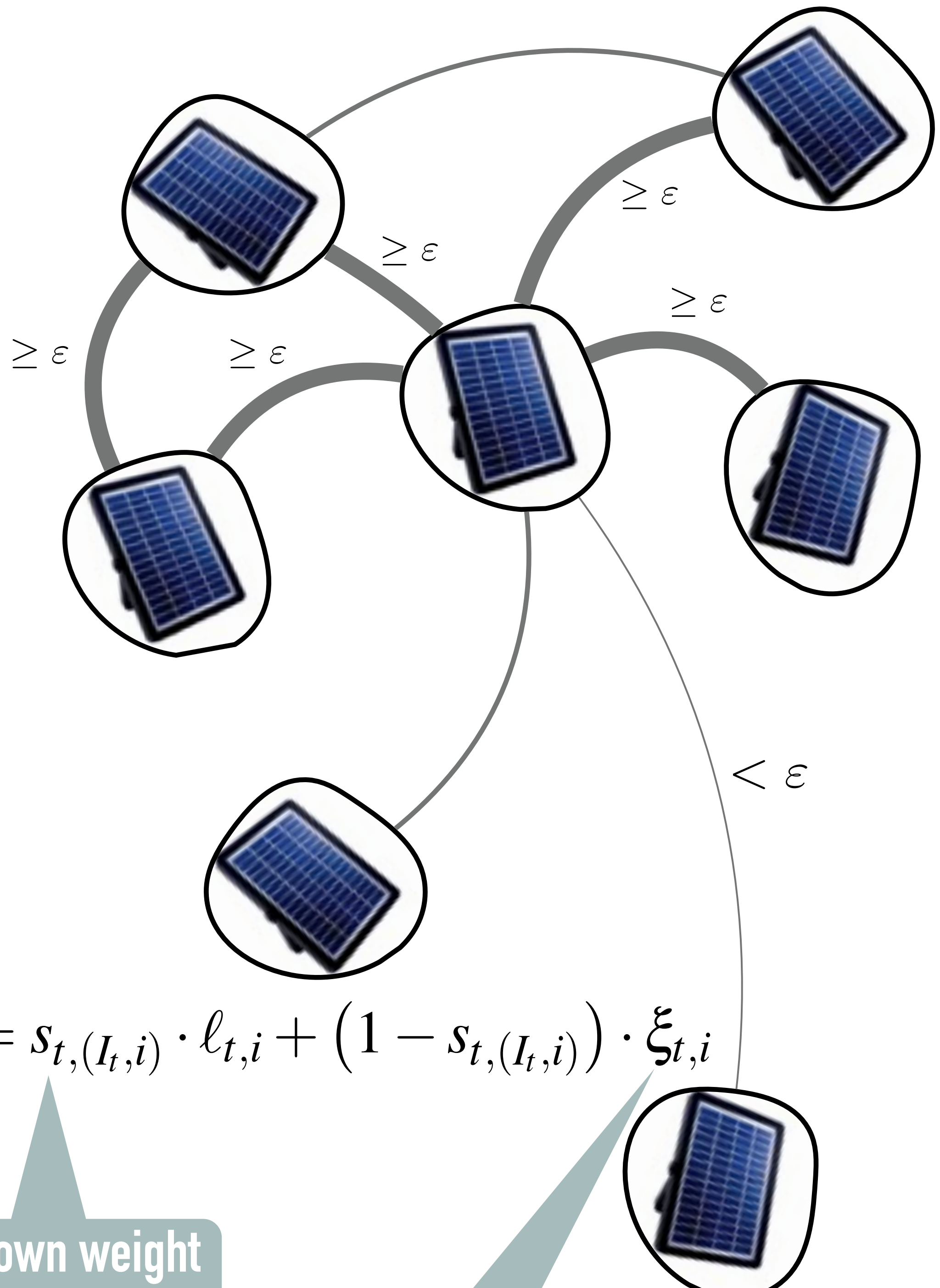
$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

Want: only reliable information!

1) If we know the perfect cutoff ε

- reliable: use as exact
- unreliable: rubbish

NOISY SIDE OBSERVATIONS



$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

known weight

zero-mean noise

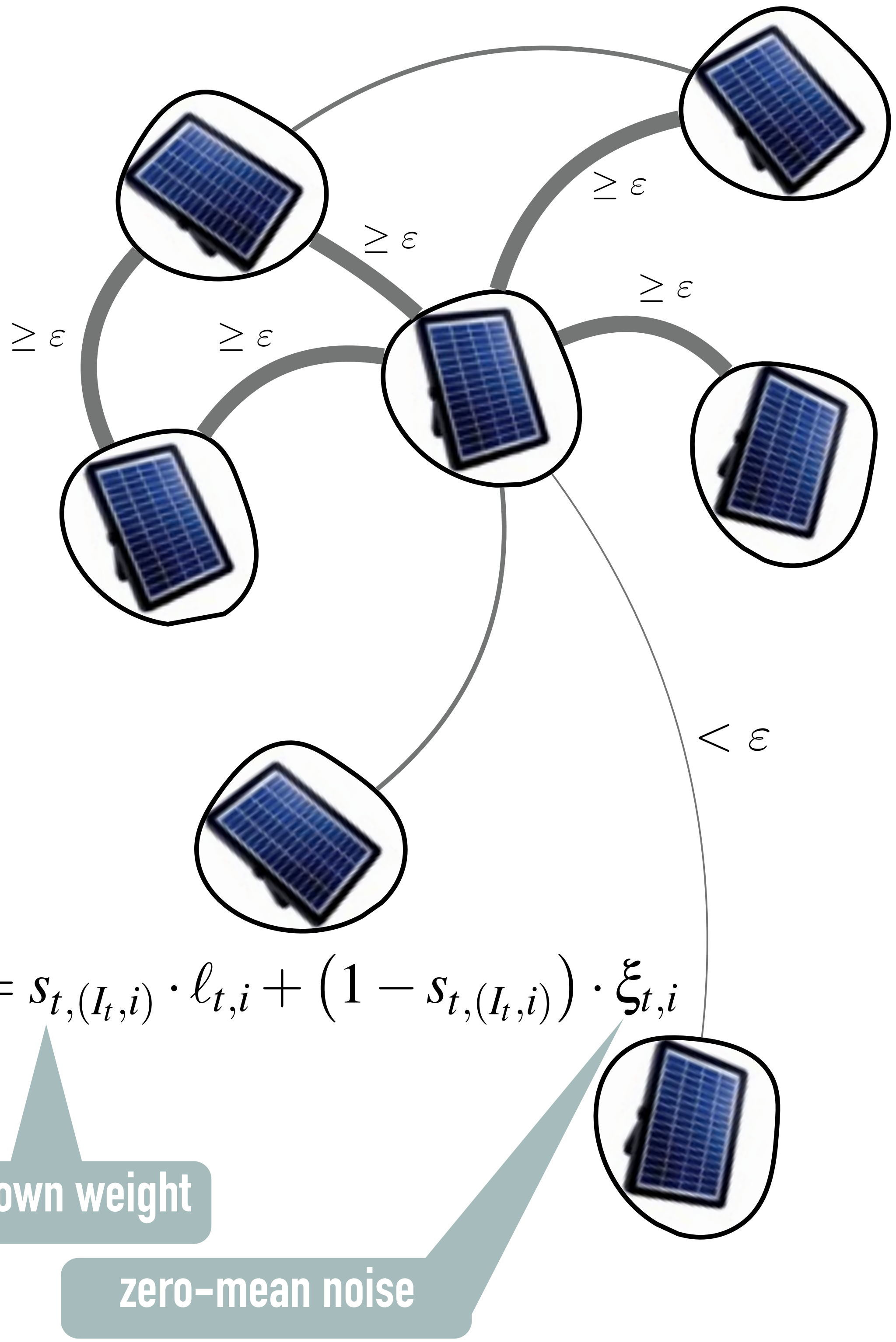
Want: only reliable information!

1) If we know the perfect cutoff ε

- reliable: use as exact
- unreliable: rubbish

then we can improve over pure bandit setting!

NOISY SIDE OBSERVATIONS



$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

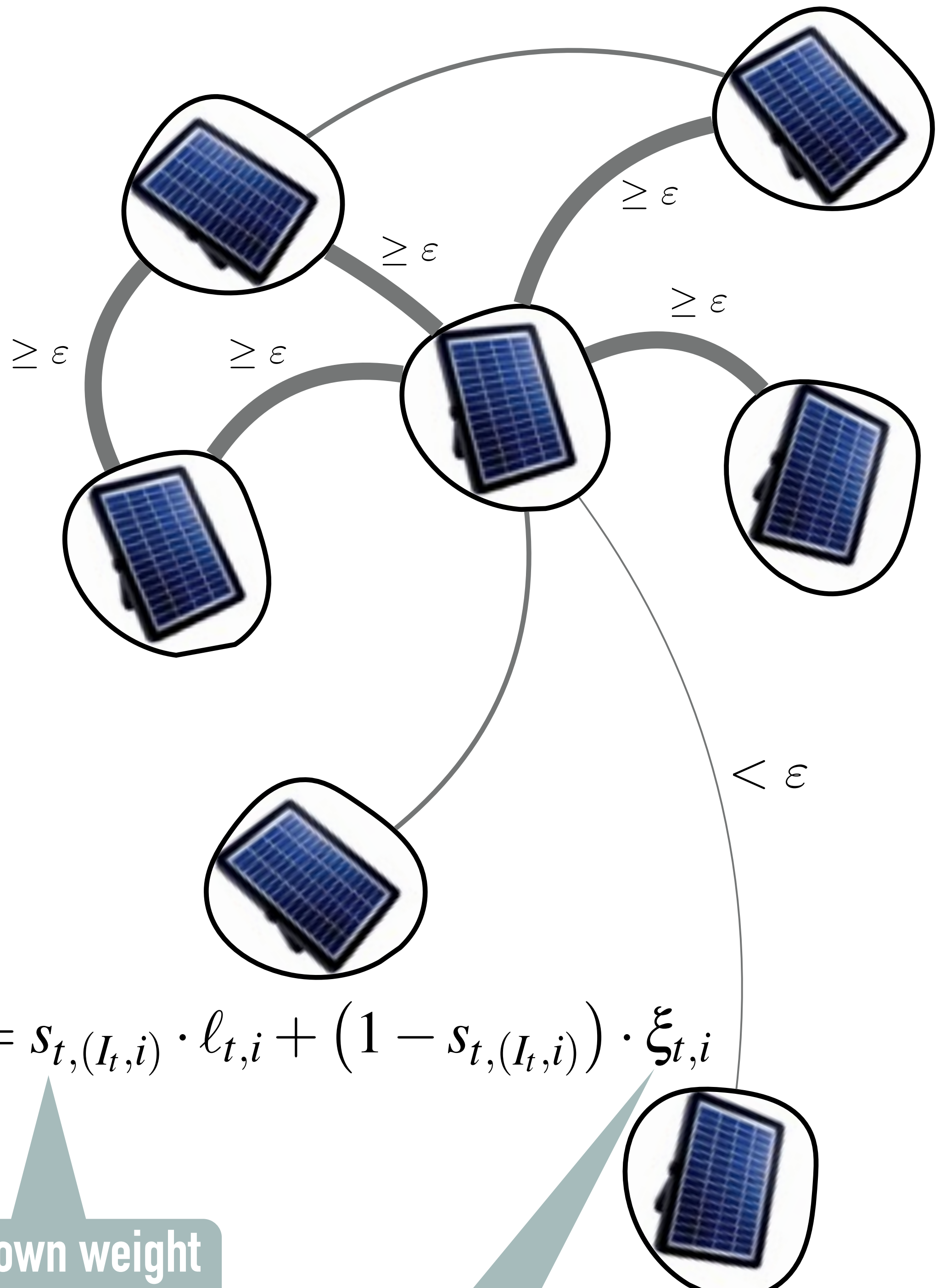
Want: only reliable information!

1) If we know the perfect cutoff ε

- reliable: use as exact
- unreliable: rubbish

then we can improve over pure bandit setting!

NOISY SIDE OBSERVATIONS



$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

known weight

zero-mean noise

Want: only reliable information!

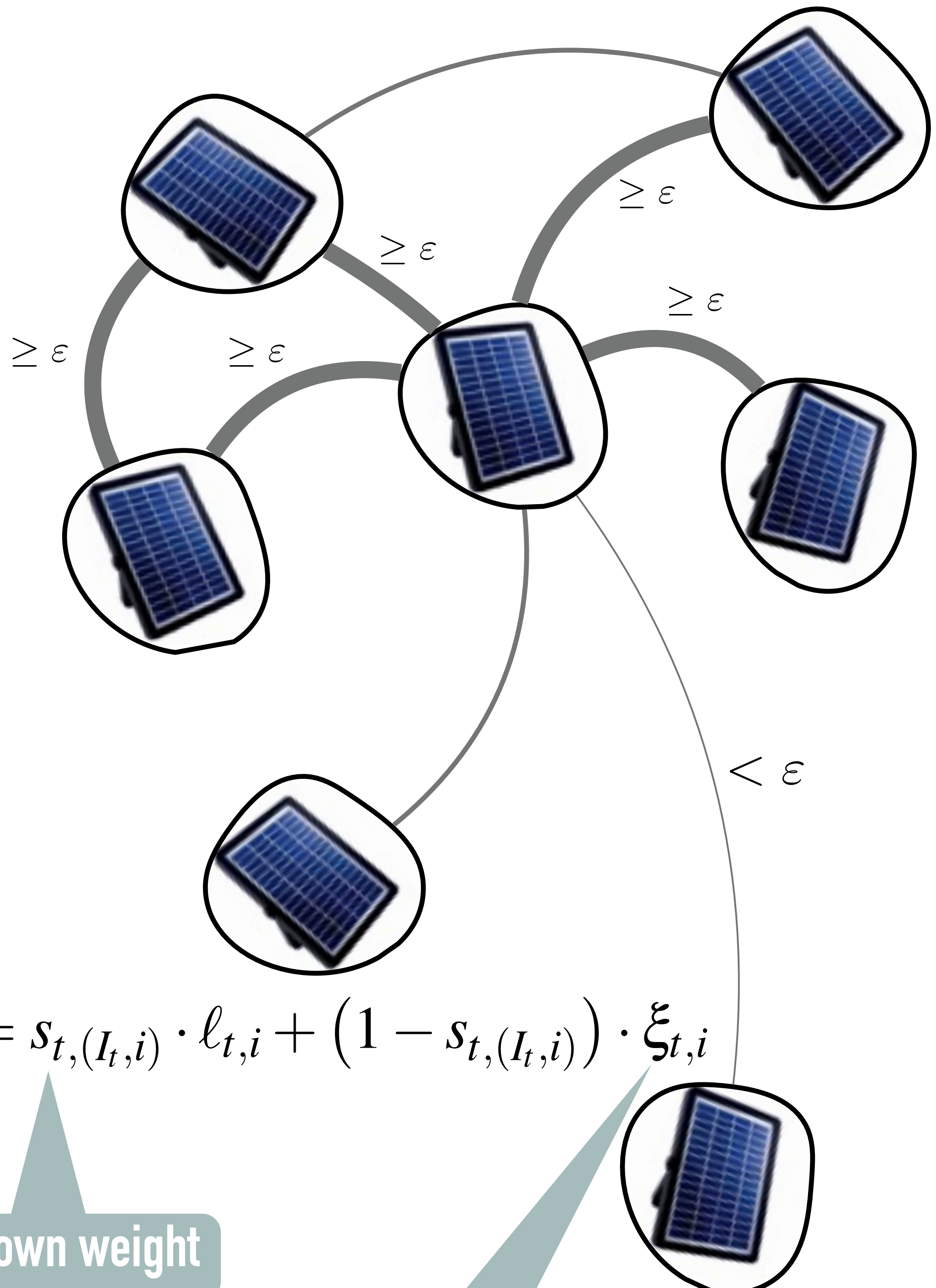
1) If we know the perfect cutoff ε

- reliable: use as exact
- unreliable: rubbish

then we can improve over pure bandit setting!

2) Treating noisy observation induces bias

NOISY SIDE OBSERVATIONS



$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

Want: only reliable information!

1) If we know the perfect cutoff ε

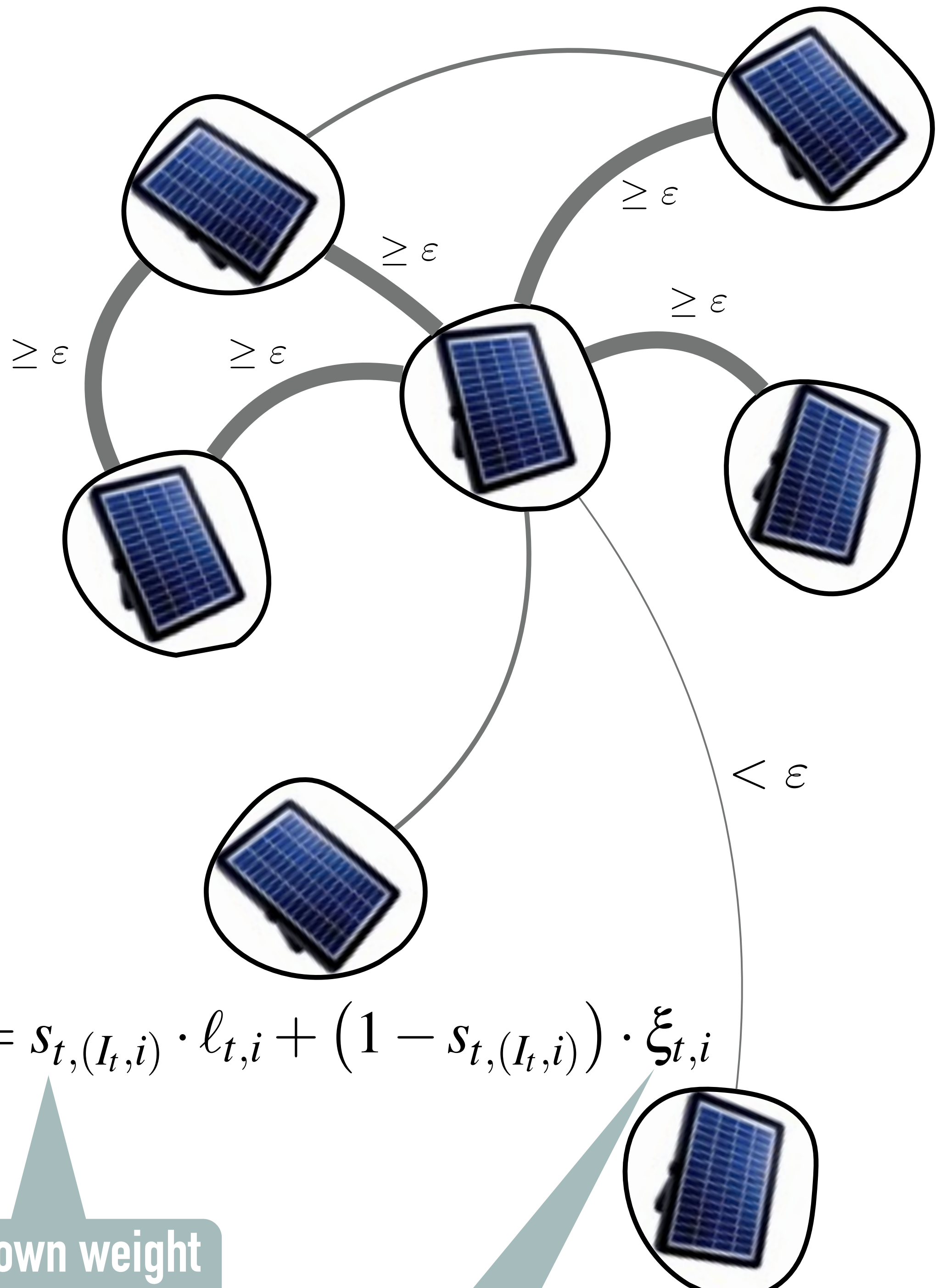
- reliable: use as exact
- unreliable: rubbish

then we can improve over pure bandit setting!

2) Treating noisy observation induces bias

What can we hope for?

NOISY SIDE OBSERVATIONS



$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

known weight

zero-mean noise

Want: only reliable information!

1) If we know the perfect cutoff ε

- reliable: use as exact
- unreliable: rubbish

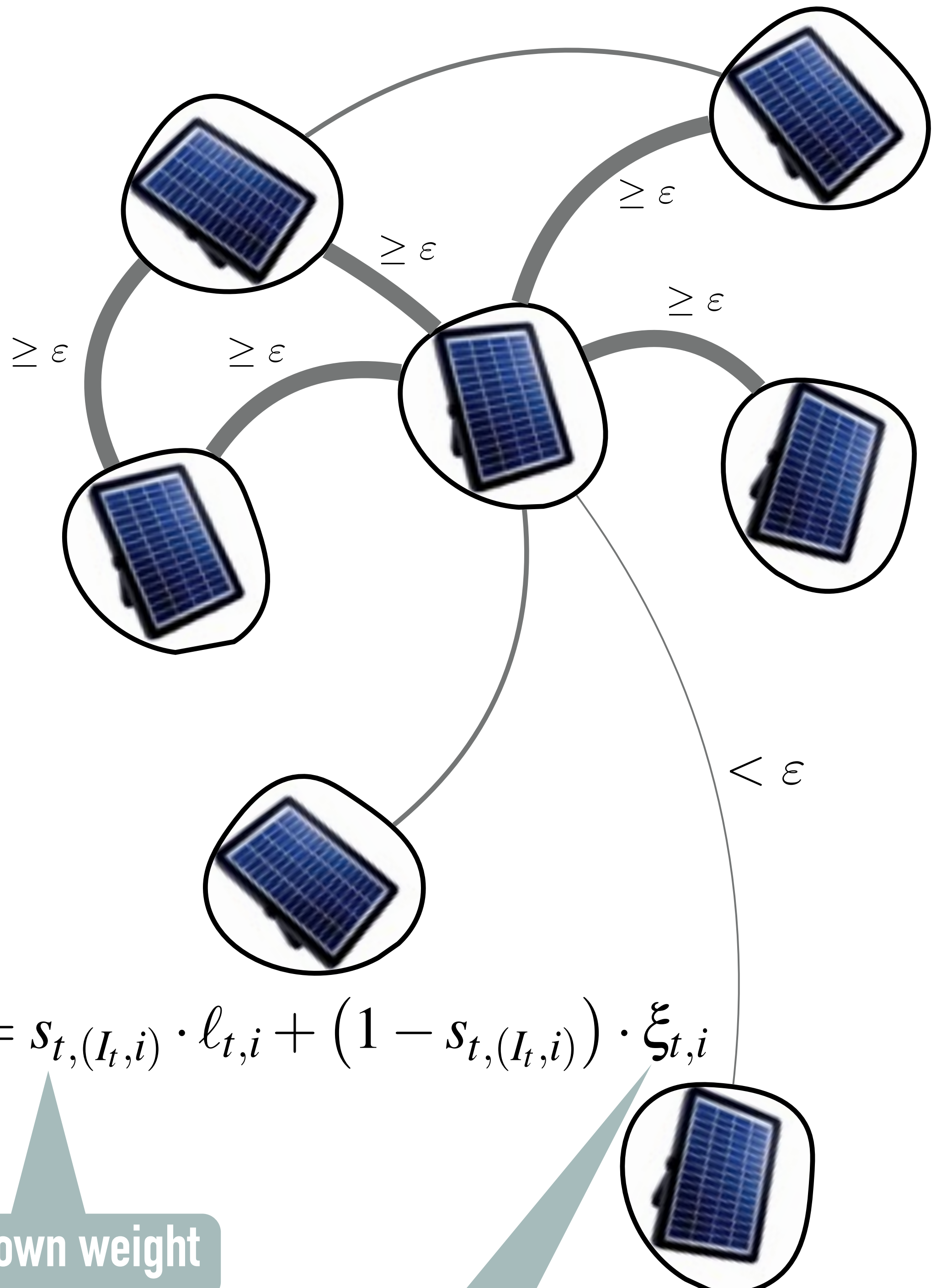
then we can improve over pure bandit setting!

2) Treating noisy observation induces bias

What can we hope for?

$\tilde{\mathcal{O}}\left(\sqrt{1T}\right) \leq \qquad \leq \tilde{\mathcal{O}}\left(\sqrt{NT}\right)$

NOISY SIDE OBSERVATIONS



$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

known weight

zero-mean noise

Want: only reliable information!

1) If we know the perfect cutoff ε

- reliable: use as exact
- unreliable: rubbish

then we can improve over pure bandit setting!

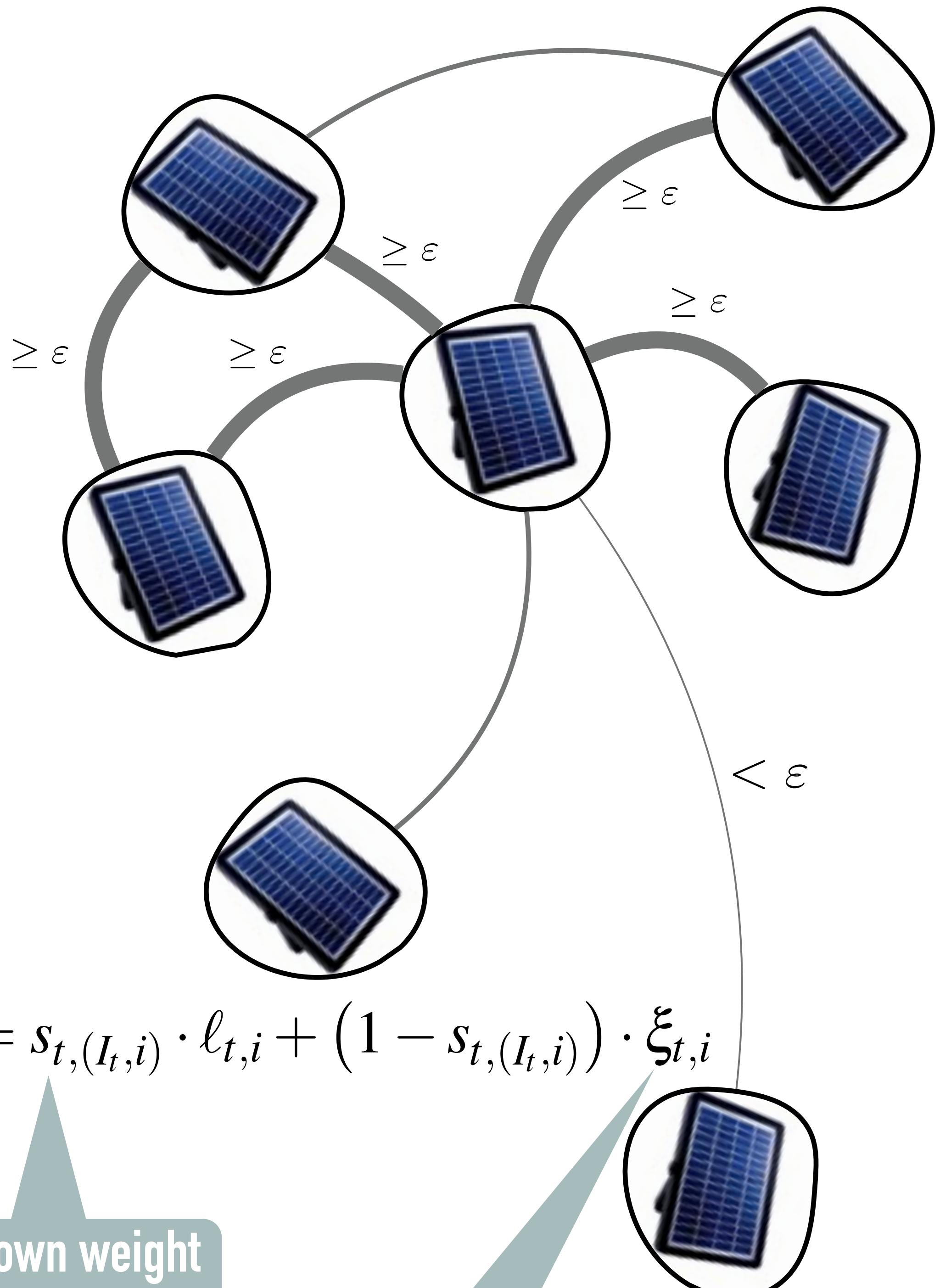
2) Treating noisy observation induces bias

What can we hope for?

$$\tilde{\mathcal{O}}\left(\sqrt{1T}\right) \leq \tilde{\mathcal{O}}\left(\sqrt{\overline{\alpha}^\star T}\right) \leq \tilde{\mathcal{O}}\left(\sqrt{NT}\right)$$

effective independence number

NOISY SIDE OBSERVATIONS



$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

known weight

zero-mean noise

Want: only reliable information!

1) If we know the perfect cutoff ε

- reliable: use as exact
- unreliable: rubbish

then we can improve over pure bandit setting!

2) Treating noisy observation induces bias

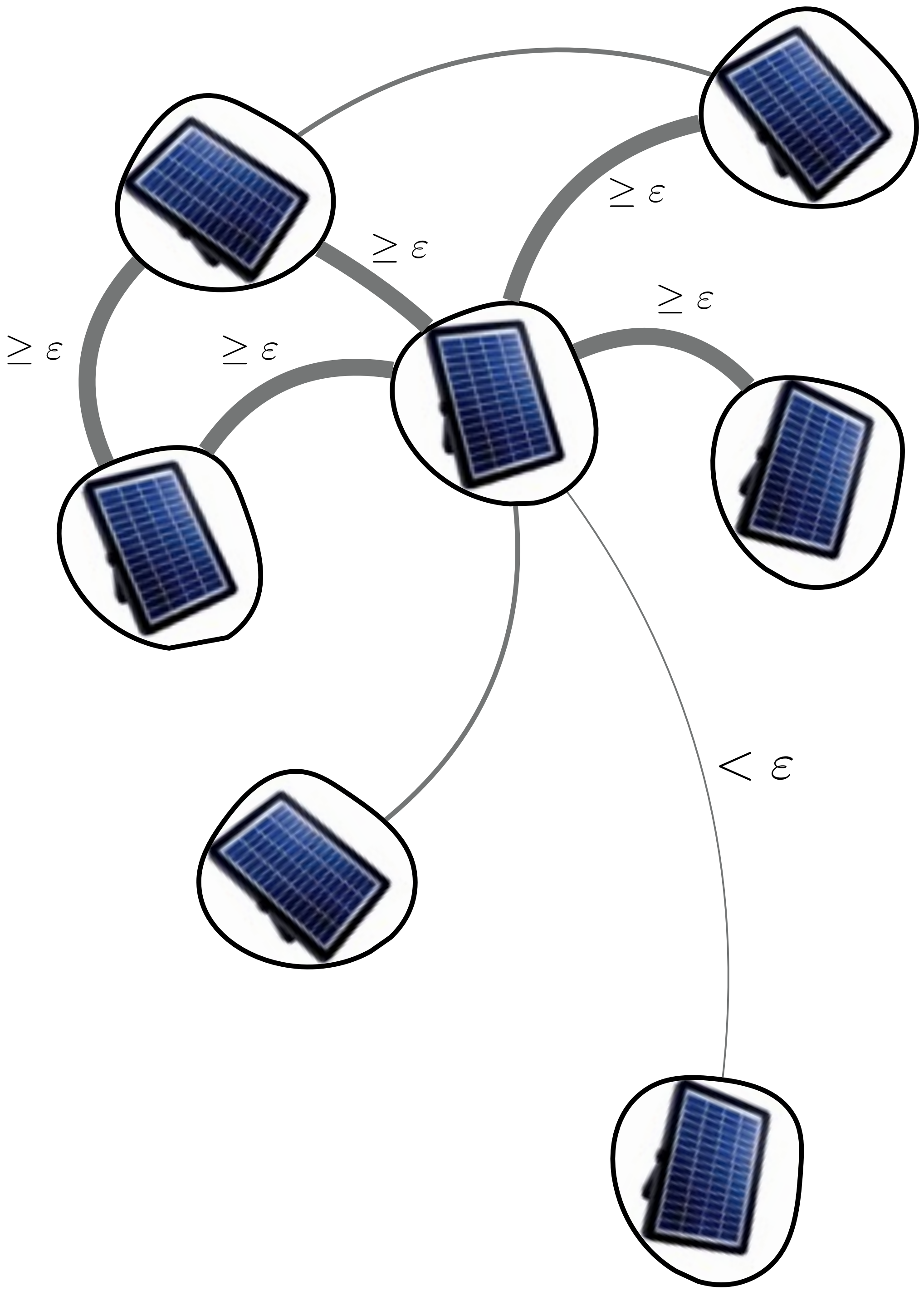
What can we hope for?

$$\tilde{\mathcal{O}}\left(\sqrt{1T}\right) \leq \tilde{\mathcal{O}}\left(\sqrt{\overline{\alpha}^{\star} T}\right) \leq \tilde{\mathcal{O}}\left(\sqrt{NT}\right)$$

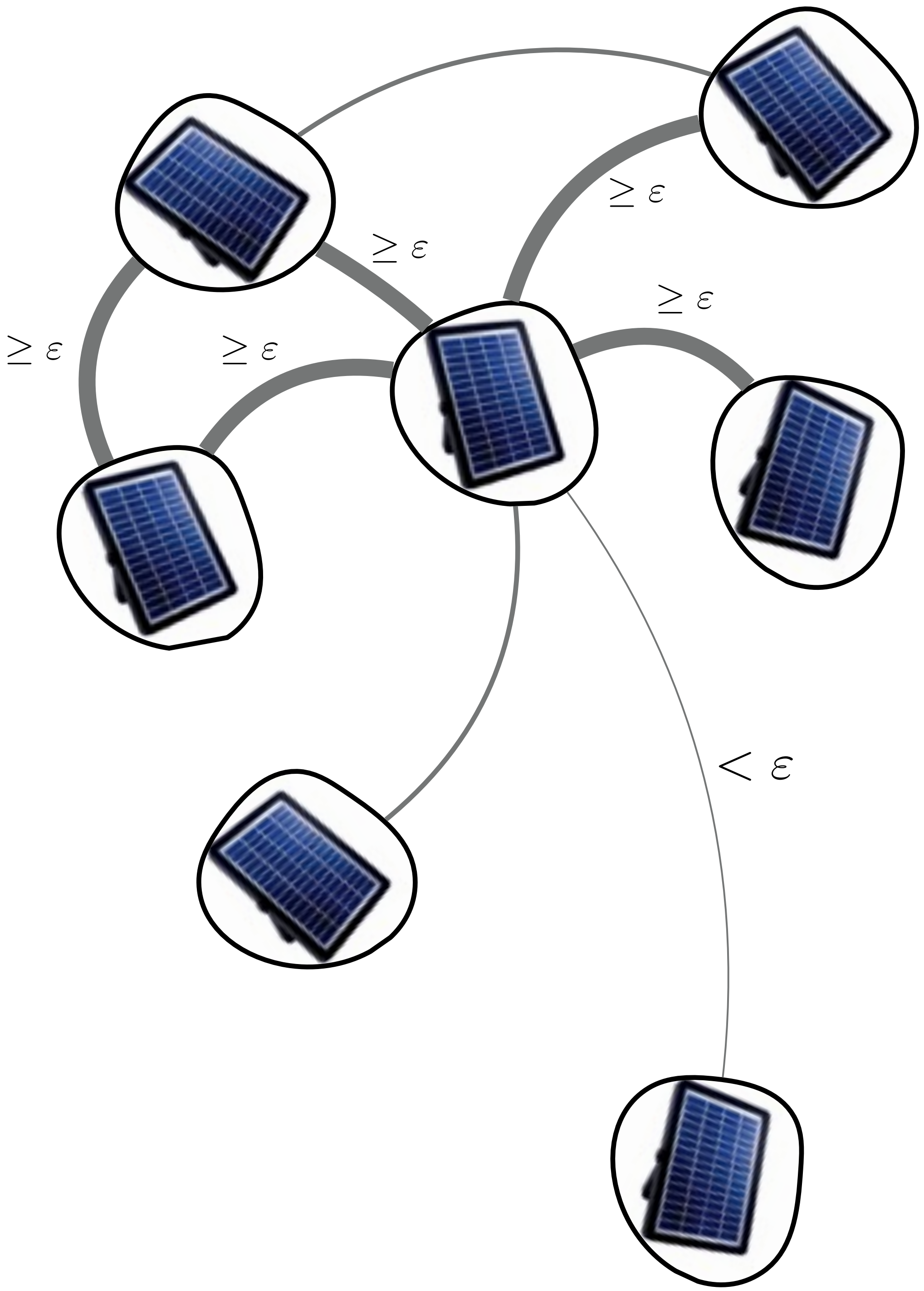
effective independence number

Can we learn without knowing either ε or α^*?

NOISY SIDE OBSERVATIONS

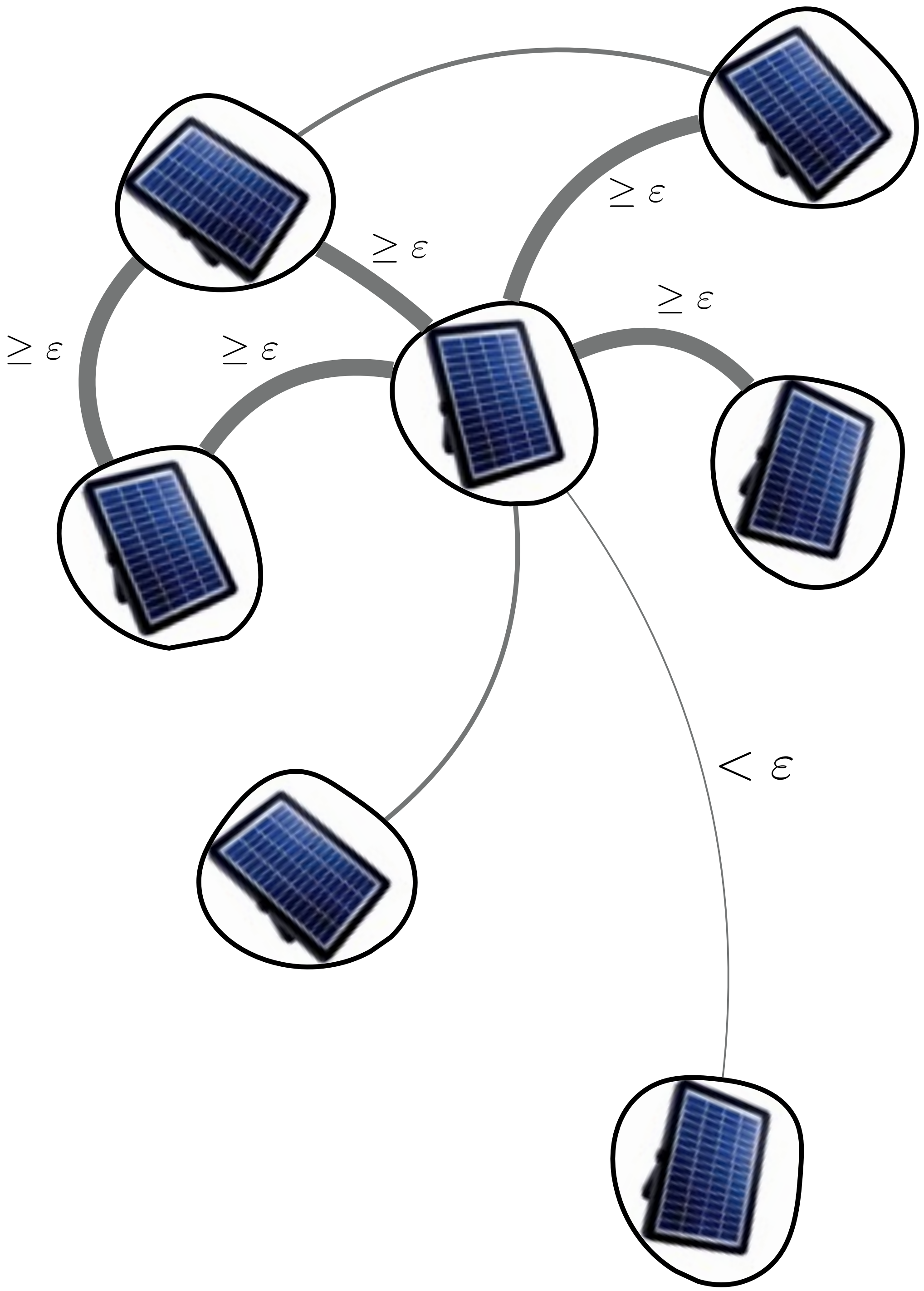


NOISY SIDE OBSERVATIONS



- **G**: weighted graph
- **G(ε)**: graph with only ≥ε edges
- **α(ε)**: independence number of G(ε)
- **effective independence number of G:**

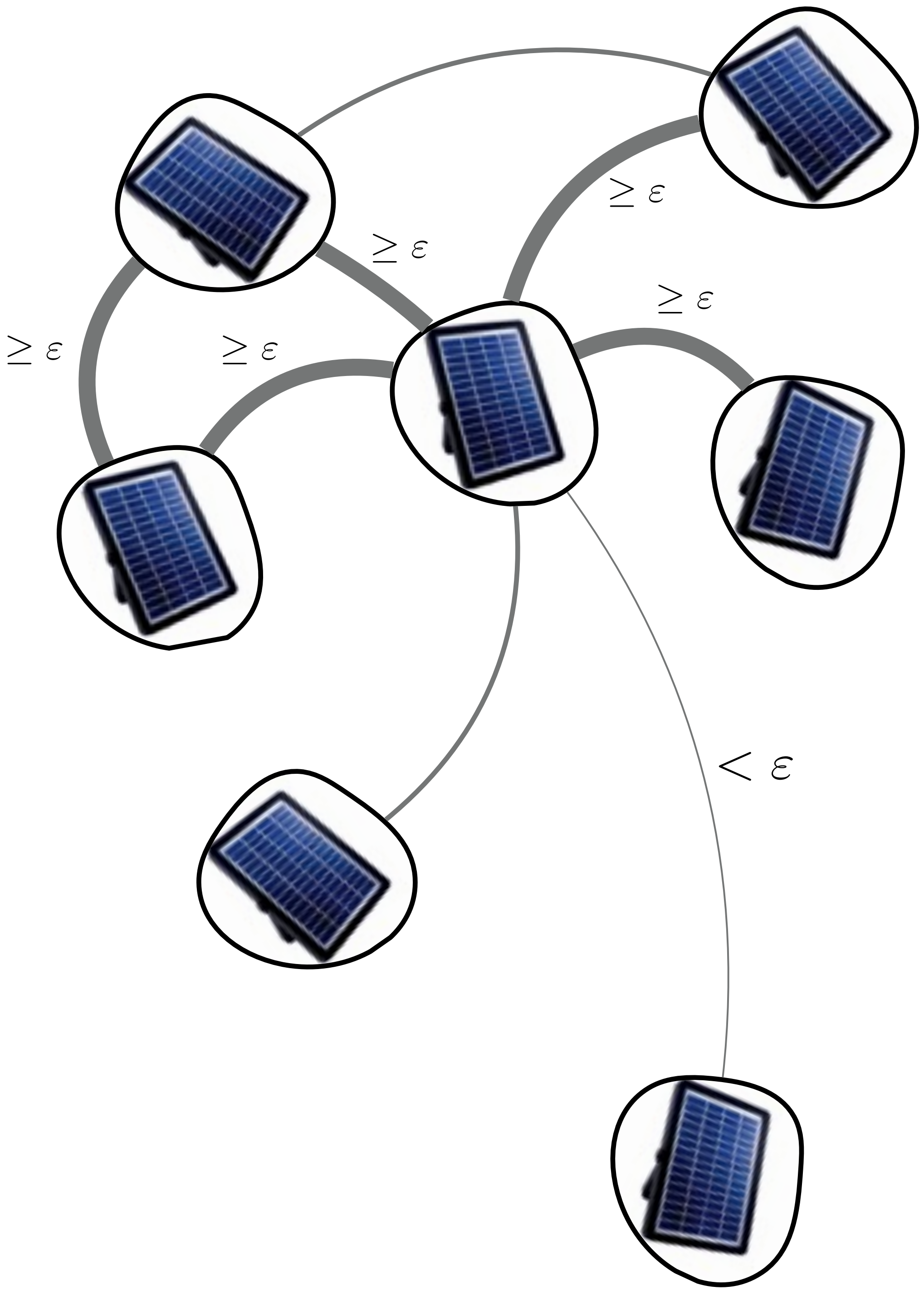
NOISY SIDE OBSERVATIONS



- **G**: weighted graph
- **G(ε)**: graph with only ≥ε edges
- **α(ε)**: independence number of G(ε)
- **effective independence number of G:**

$$\alpha^* = \min_{\varepsilon \in [0,1]} \frac{\alpha(\varepsilon)}{\varepsilon^2}$$

NOISY SIDE OBSERVATIONS



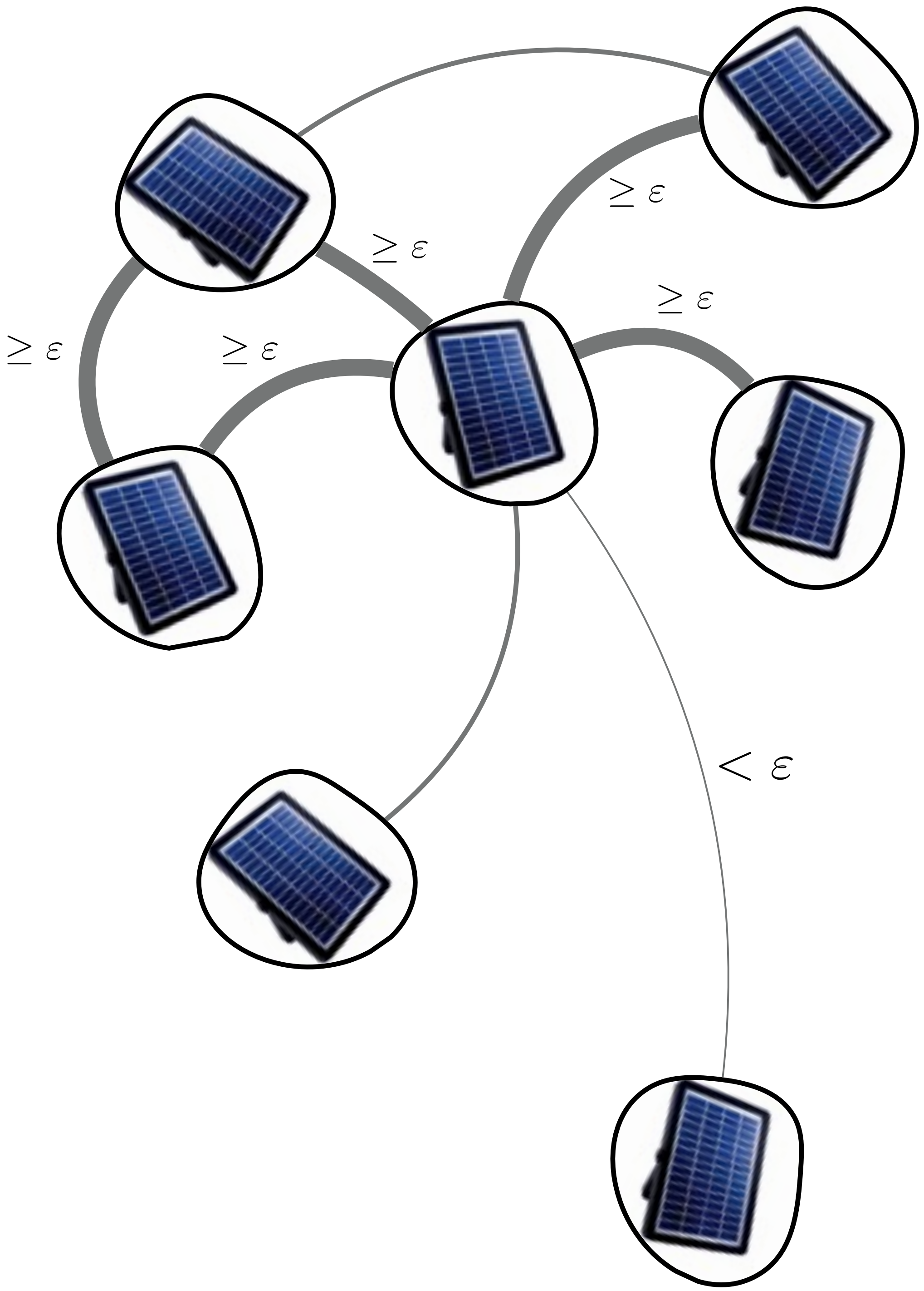
Threshold estimate $R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T}\right)$

$$\widehat{\ell}^{(\mathrm{T})}_{t,i} = \frac{c_{t,i}\mathbb{I}_{\{s_{t,(I_t,i)}\geq \varepsilon_t\}}}{\sum_{j=1}^N p_{t,j}s_{t,(j,i)}\mathbb{I}_{\{s_{t,(j,i)}\geq \varepsilon_t\}} + \gamma_t}$$

- **effective independence number of G:**

$$\alpha^* = \min_{\varepsilon\in[0,1]} \frac{\alpha(\varepsilon)}{\varepsilon^2}$$

NOISY SIDE OBSERVATIONS



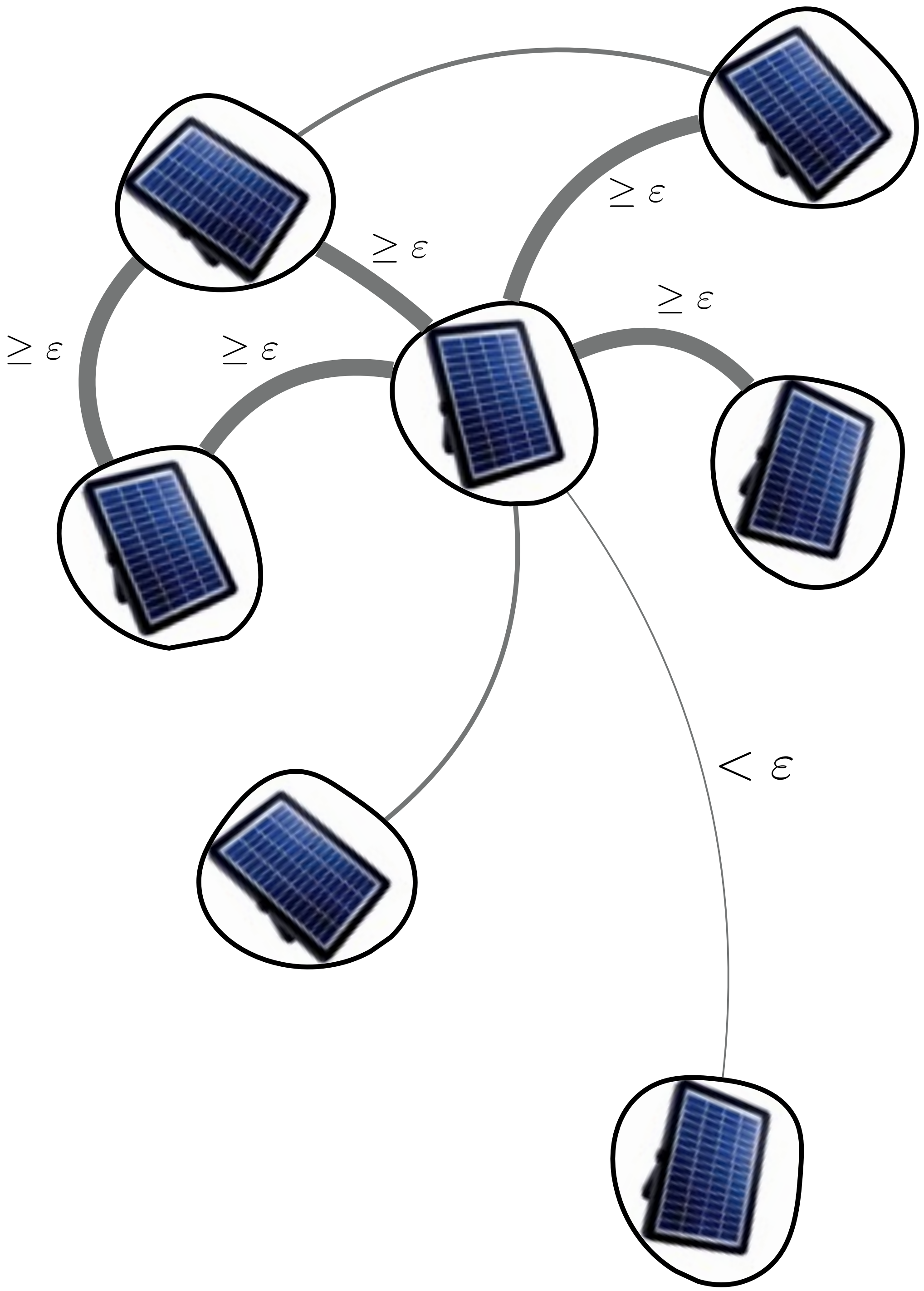
Threshold estimate $R_T = \widetilde{\mathcal{O}}\left(\sqrt{\overline{\alpha}^\star T}\right)$

$$\widehat{\ell}^{(\mathrm{T})}_{t,i} = \frac{c_{t,i}\mathbb{I}_{\{s_{t,(I_t,i)} \geq \varepsilon_t\}}}{\sum_{j=1}^N p_{t,j} s_{t,(j,i)} \mathbb{I}_{\{s_{t,(j,i)} \geq \varepsilon_t\}} + \gamma_t}$$

WIX estimate $R_T = \widetilde{\mathcal{O}}\left(\sqrt{\overline{\alpha}^\star T}\right)$

$$\widehat{\ell}_{t,i} = \frac{s_{t,(I_t,i)} \cdot c_{t,i}}{\sum_{j=1}^N p_{t,j} s^2_{t,(j,i)} + \gamma_t}$$

NOISY SIDE OBSERVATIONS



Threshold estimate $R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T}\right)$

$$\widehat{\ell}^{(\mathrm{T})}_{t,i} = \frac{c_{t,i}\mathbb{I}_{\{s_{t,(I_t,i)}\geq\varepsilon_t\}}}{\sum_{j=1}^N p_{t,j}s_{t,(j,i)}\mathbb{I}_{\{s_{t,(j,i)}\geq\varepsilon_t\}}+\gamma_t}$$

WIX estimate $R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T}\right)$

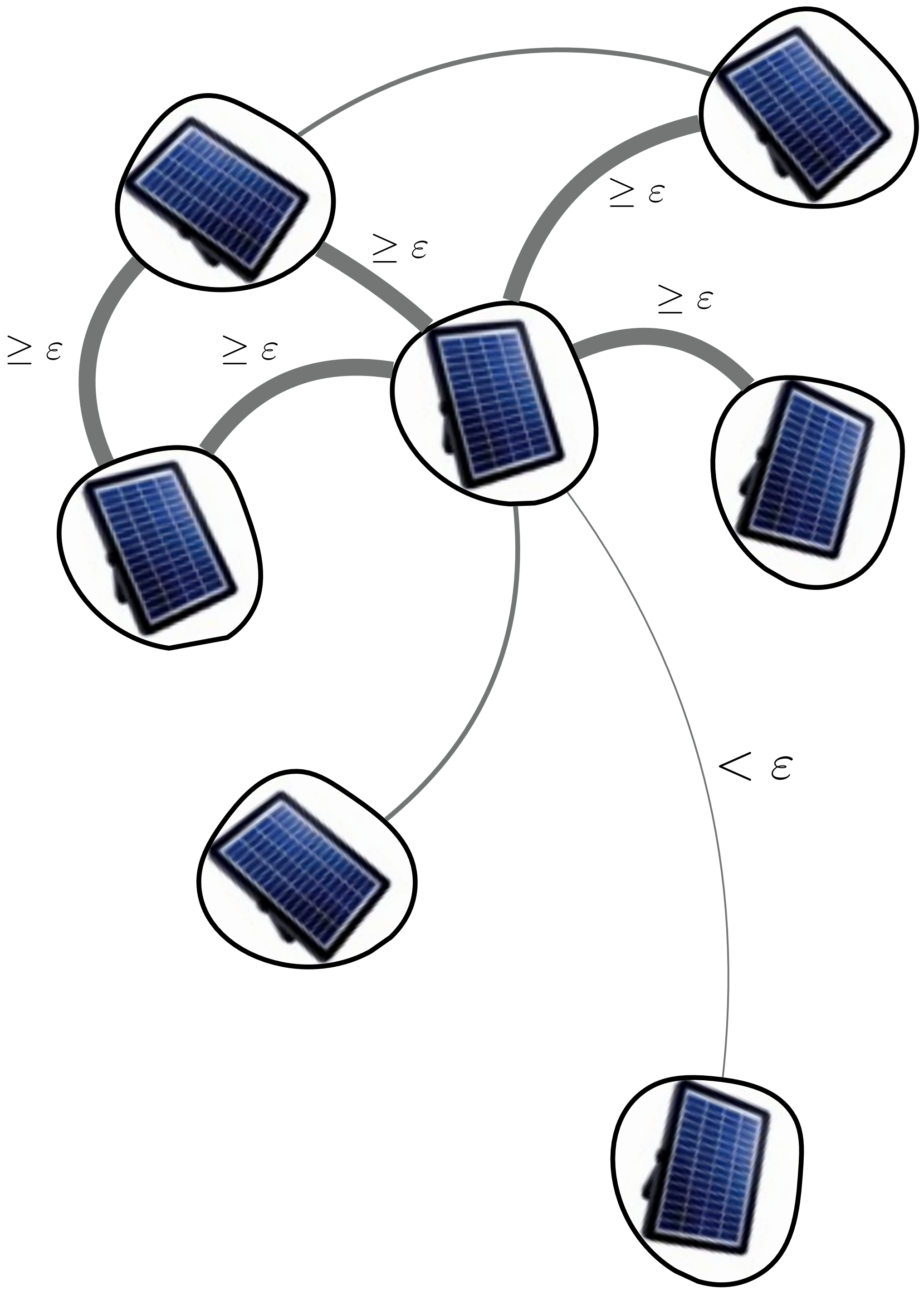
$$\widehat{\ell}_{t,i} = \frac{s_{t,(I_t,i)}\cdot c_{t,i}}{\sum_{j=1}^N p_{t,j}s^2_{t,(j,i)}+\gamma_t}$$

Since $\alpha^* \leq \alpha(1)/1 \leq N$

incorporating noisy observations does not hurt

$$\widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T}\right) \leq \widetilde{\mathcal{O}}\left(\sqrt{NT}\right)$$

NOISY SIDE OBSERVATIONS



Threshold estimate $R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T}\right)$

$$\widehat{\ell}^{(\mathrm{T})}_{t,i} = \frac{c_{t,i}\mathbb{I}_{\{s_{t,(I_t,i)} \geq \varepsilon_t\}}}{\sum_{j=1}^N p_{t,j} s_{t,(j,i)} \mathbb{I}_{\{s_{t,(j,i)} \geq \varepsilon_t\}} + \gamma_t}$$

WIX estimate $R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T}\right)$

$$\widehat{\ell}_{t,i} = \frac{s_{t,(I_t,i)} \cdot c_{t,i}}{\sum_{j=1}^N p_{t,j} s^2_{t,(j,i)} + \gamma_t}$$

Since $\alpha^* \leq \alpha(1)/1 \leq N$

incorporating noisy observations does not hurt

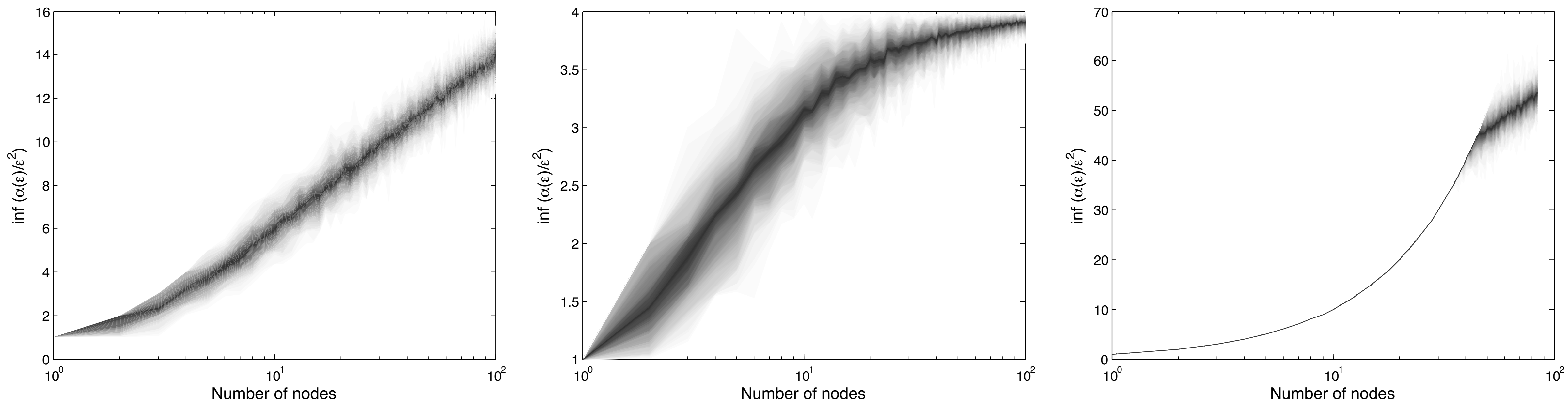
$$\widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T}\right) \leq \widetilde{\mathcal{O}}\left(\sqrt{N T}\right)$$

But how much does it help?

EMPIRICAL RESULTS

EMPIRICAL RESULTS

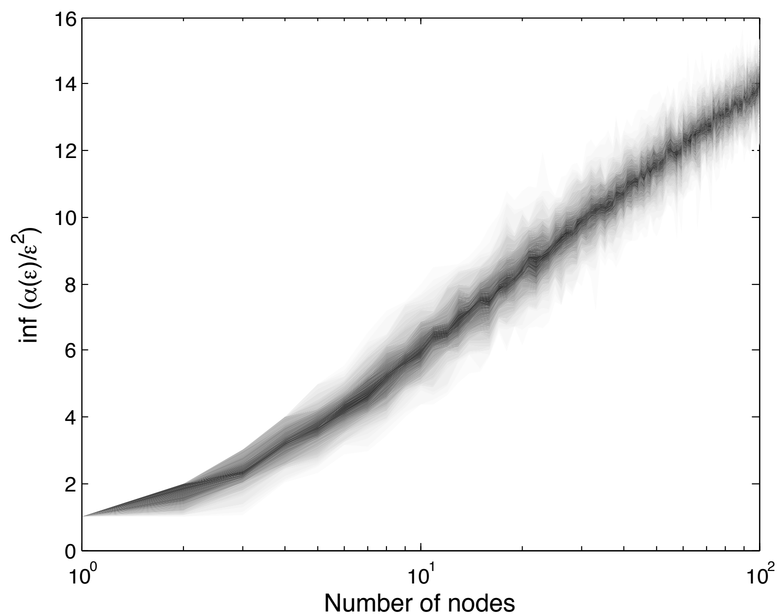
EMPIRICAL α^* FOR RANDOM GRAPHS WITH IID WEIGHTS



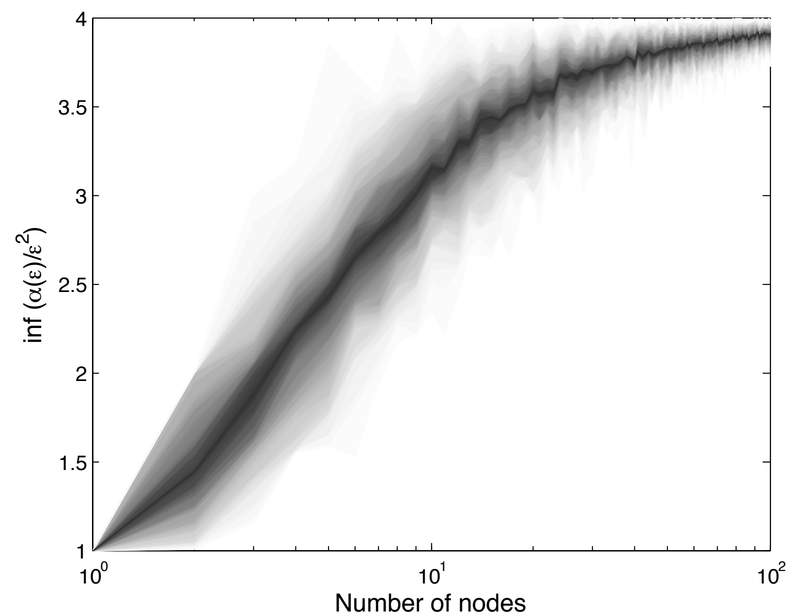
(a) $U(0,1)$ weights (b) $U(\frac{1}{2},1)$ weights (c) $U(0,\frac{1}{2})$ weights

EMPIRICAL RESULTS

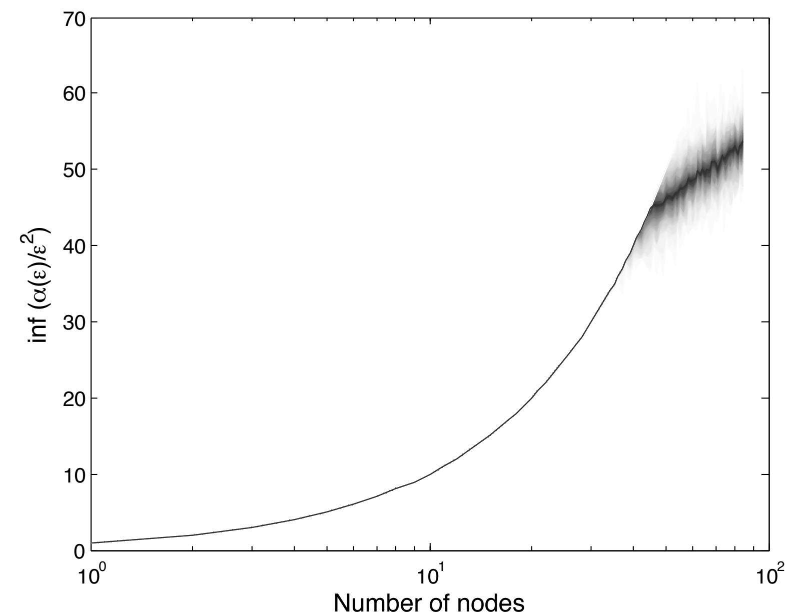
EMPIRICAL α^* FOR RANDOM GRAPHS WITH IID WEIGHTS



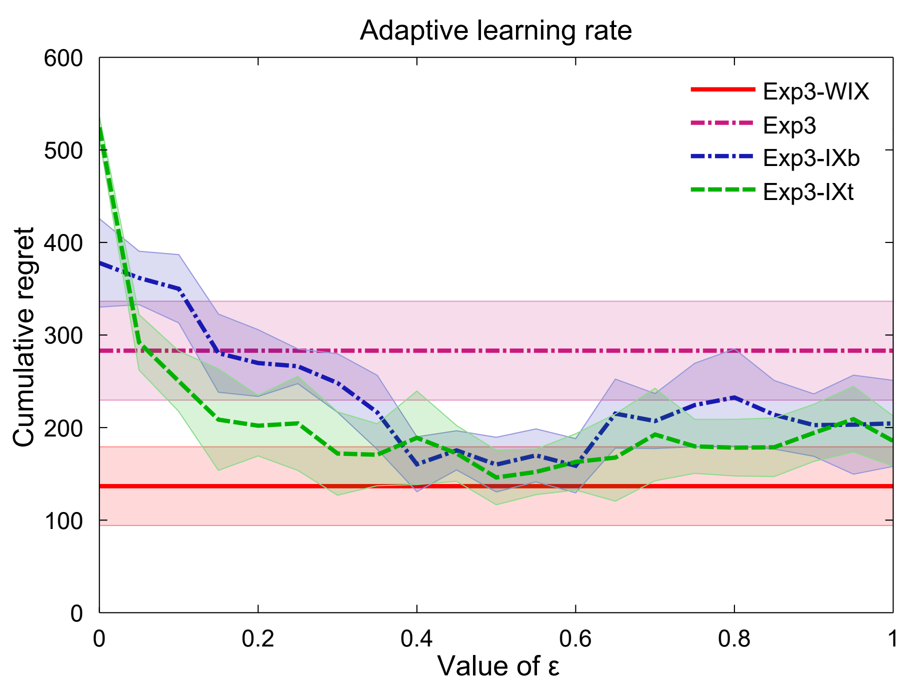
(a) $U(0,1)$ weights



(b) $U(\frac{1}{2},1)$ weights

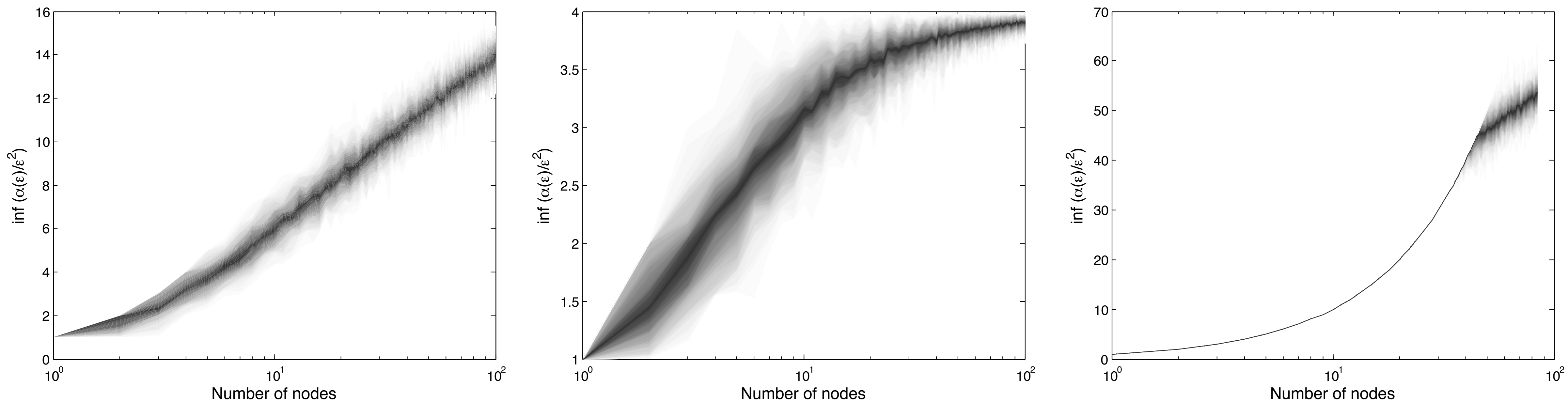


(c) $U(0,\frac{1}{2})$ weights

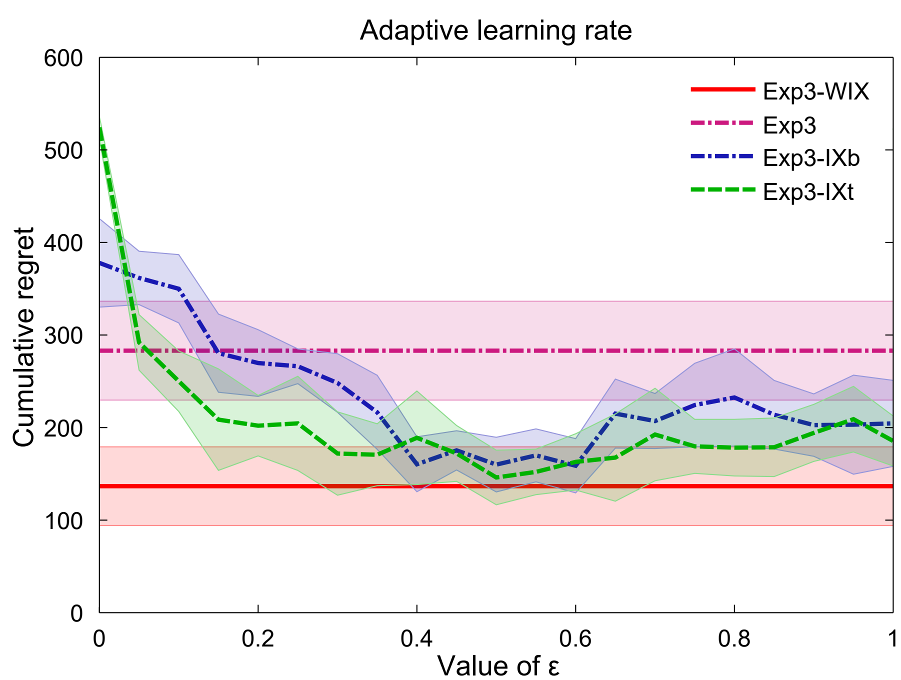


EMPIRICAL RESULTS

EMPIRICAL α^* FOR RANDOM GRAPHS WITH IID WEIGHTS



(a) $U(0,1)$ weights (b) $U(\frac{1}{2},1)$ weights (c) $U(0,\frac{1}{2})$ weights



- **special case:** if s_{ij} is either 0 or ε than $\alpha^* = \alpha/\varepsilon^2$
  - For this special case, there is a matches $\Theta(\sqrt{(\alpha T)}/\varepsilon)$ by Wu, György, Szepesvári, 2015.

NEW DIRECTIONS

NEW DIRECTIONS: UNKNOWN GRAPHS!

NEW DIRECTIONS: UNKNOWN GRAPHS!

- **Learning on the graph while learning the graph?**

NEW DIRECTIONS: UNKNOWN GRAPHS!

- Learning on the graph
  - most of algorithms require (some) knowledge of the graph

NEW DIRECTIONS: UNKNOWN GRAPHS!

- Learning on the graph
  - most of algorithms require (some) knowledge of the graph
  - not always available to the learner

NEW DIRECTIONS: UNKNOWN GRAPHS!

- Learning on the graph
  - most of algorithms require (some) knowledge of the graph
  - not always available to the learner
- Question: Can we learn faster without knowing the graphs?

NEW DIRECTIONS: UNKNOWN GRAPHS!

- Learning on the graph
  - most of algorithms require (some) knowledge of the graph
  - not always available to the learner
- Question: Can we learn faster without knowing the graphs?
  - example: social network provider has little incentive to reveal the graphs to advertisers

NEW DIRECTIONS: UNKNOWN GRAPHS!

- Learning on the graph
  - **most of algorithms require (some) knowledge of the graph**
  - not always available to the learner
- Question: Can we learn faster without knowing the graphs?
  - example: social network provider has little incentive to reveal the graphs to advertisers
- Answer: **Cohen, Hazan, and Koren:** Online learning with **feedback** graphs without the graphs (ICML 2016, to appear)

NEW DIRECTIONS: UNKNOWN GRAPHS!

- **Learning on the graph**
  - **most of algorithms require (some) knowledge of the graph**
  - not always available to the learner
- Question: Can we learn faster without knowing the graphs?
  - example: social network provider has little incentive to reveal the graphs to advertisers
- Answer:
  the graphs
  - **NO!** (in general we cannot, but possible in the stochastic case)

NEW DIRECTIONS: UNKNOWN GRAPHS!

- Learning on the graph
  - **most of algorithms require (some) knowledge of the graph**
  - not always available to the learner
- Question: Can we learn faster without knowing the graphs?
  - example: social network provider has little incentive to reveal the graphs to advertisers
- Answer:
  the graphs
  - **NO!**
- Rest of the talk:

NEW DIRECTIONS: UNKNOWN GRAPHS!

- Learning on the graph
  - **most of algorithms require (some) knowledge of the graph**
  - not always available to the learner
- Question: Can we learn faster without knowing the graphs?
  - example: social network provider has little incentive to reveal the graphs to advertisers
- Answer:
  the graphs
  - **NO!**
- Rest of the talk:
  - **Erdös-Rényi side observation graphs**

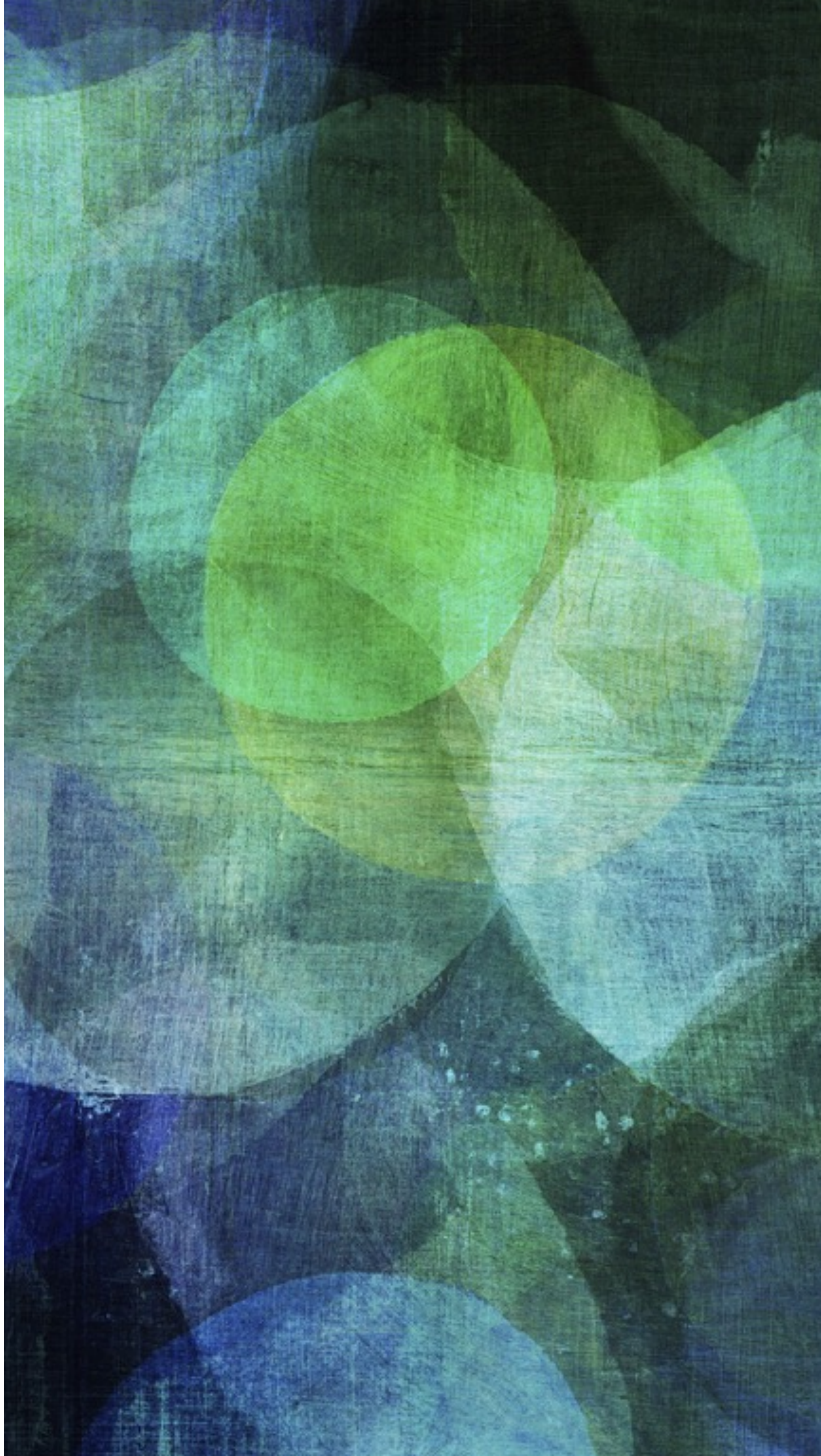
NEW DIRECTIONS: UNKNOWN GRAPHS!

- Learning on the graph
  - **most of algorithms require (some) knowledge of the graph**
  - not always available to the learner
- Question: Can we learn faster without knowing the graphs?
  - example: social network provider has little incentive to reveal the graphs to advertisers
- Answer:
  the graphs
  - **NO!**
- Rest of the talk:
  - **Erdös-Rényi side observation graphs**
  - **Influence Maximisation**

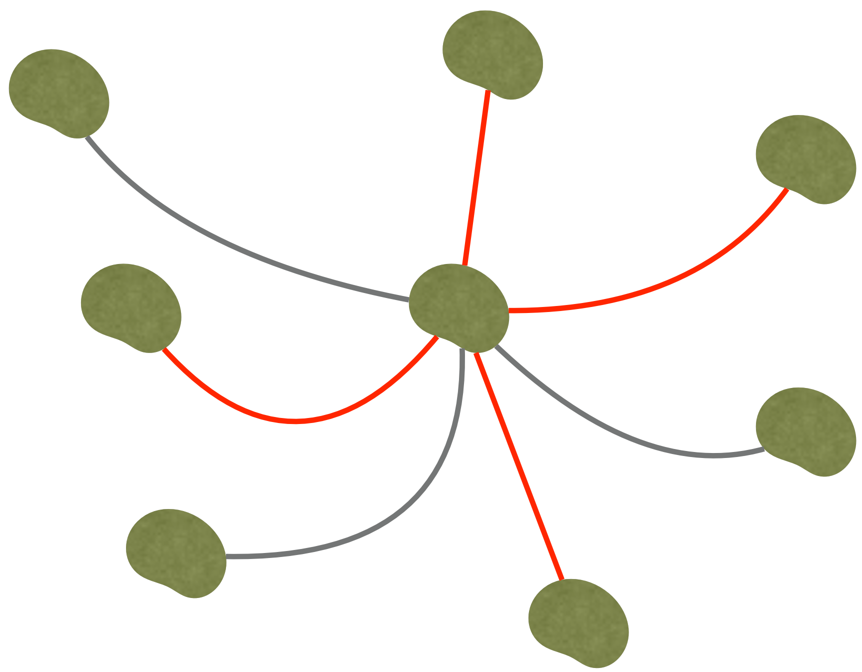
Kocák, Neu, MV: **Online learning with Erdos-Rényi side-observation graphs**
UAI 2016 (to appear)

GRAPH BANDITS WITH ERDÖS-RÉNYI OBSERVATIONS

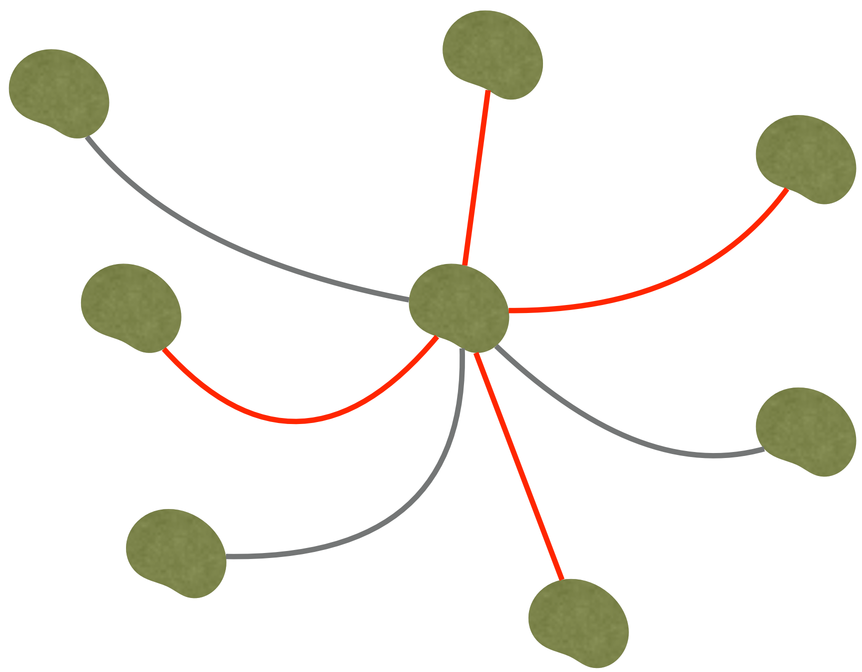
side observations from graph generators



PROTOCOL FOR ERDÖS-RÉNYI GRAPHS

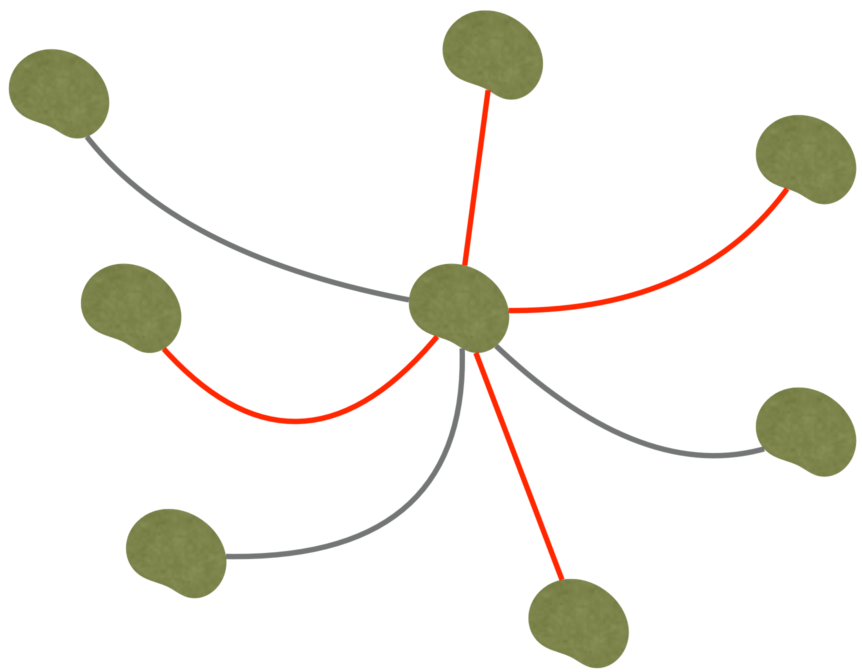


PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

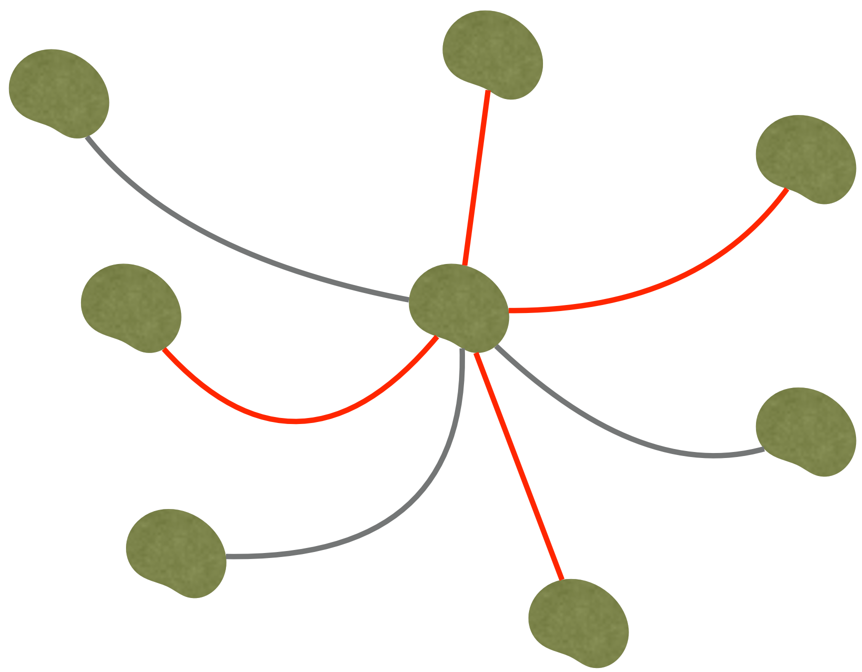
PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

- picks a node I_t

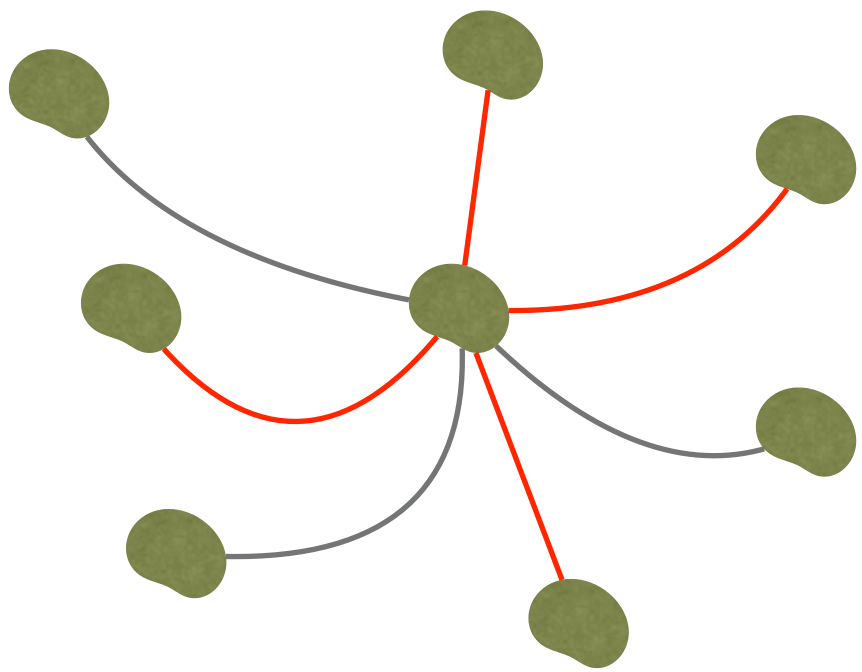
PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

- picks a node I_t
- suffers loss for I_t

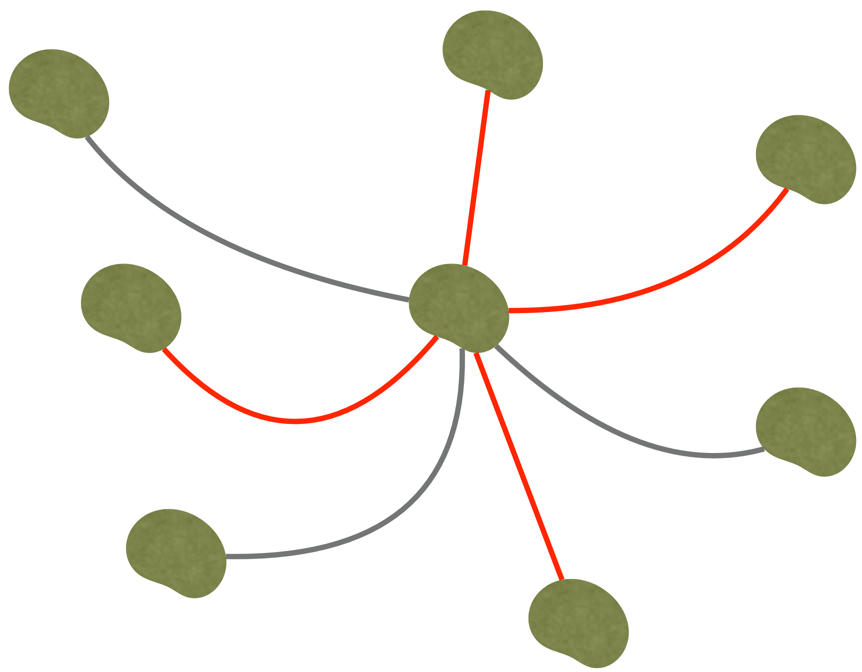
PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

- picks a node I_t
- suffers loss for I_t
- receives feedback

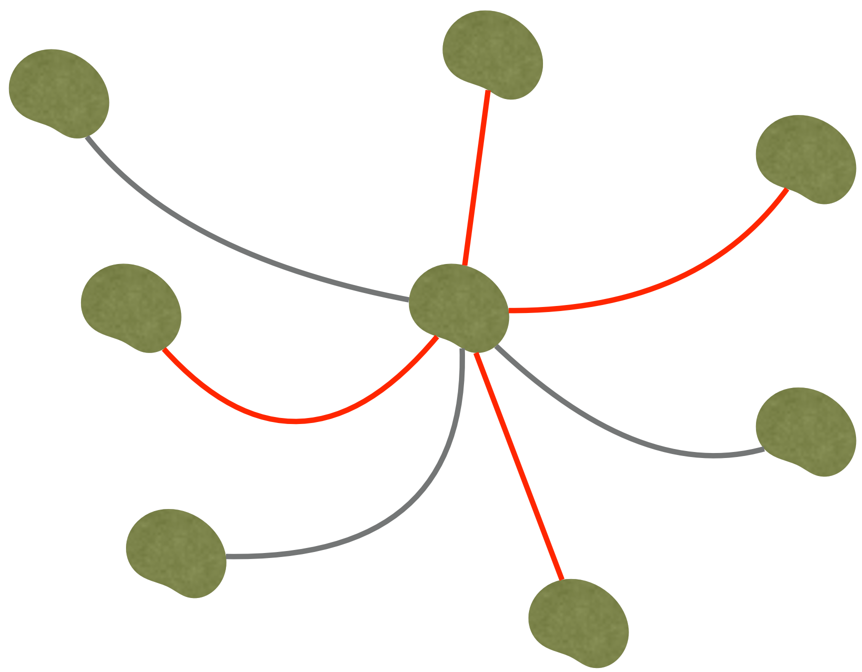
PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

- picks a node I_t
- suffers loss for I_t
- receives feedback
    - for I_t

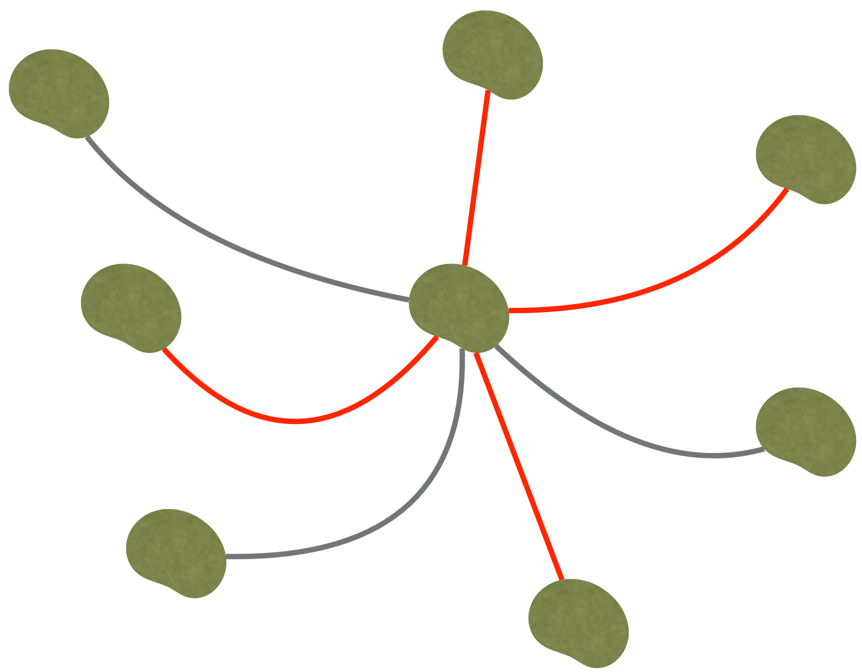
PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

- picks a node I_t
- suffers loss for I_t
- receives feedback
  - for I_t
  - for every other node with probability r_t

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

- picks a node I_t
- suffers loss for I_t
- receives feedback
  - for I_t
  - for every other node with probability r_t

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

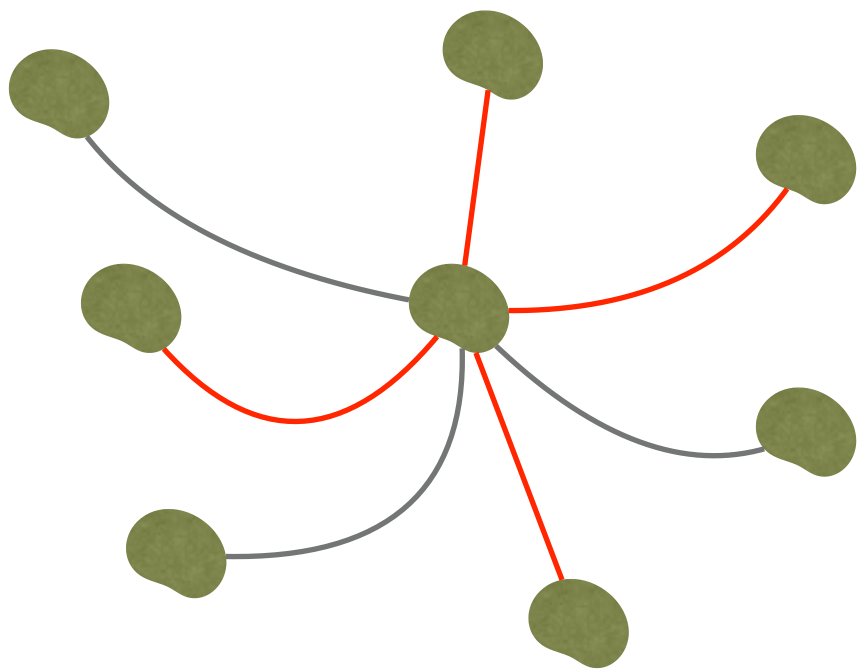
is loss of i observed?

true loss

probability of picking i

probability of side observation

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

- picks a node I_t
- suffers loss for I_t
- receives feedback
  - for I_t
  - for every other node with probability r_t

Regret of Exp3-SET (Alon et al. 2013):

$$\mathcal{O}\left(\sqrt{\sum_t (1/r_t)(1-(1-r_t)^N)\log N}\right)$$

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1-p_{t,i})r_t}$$

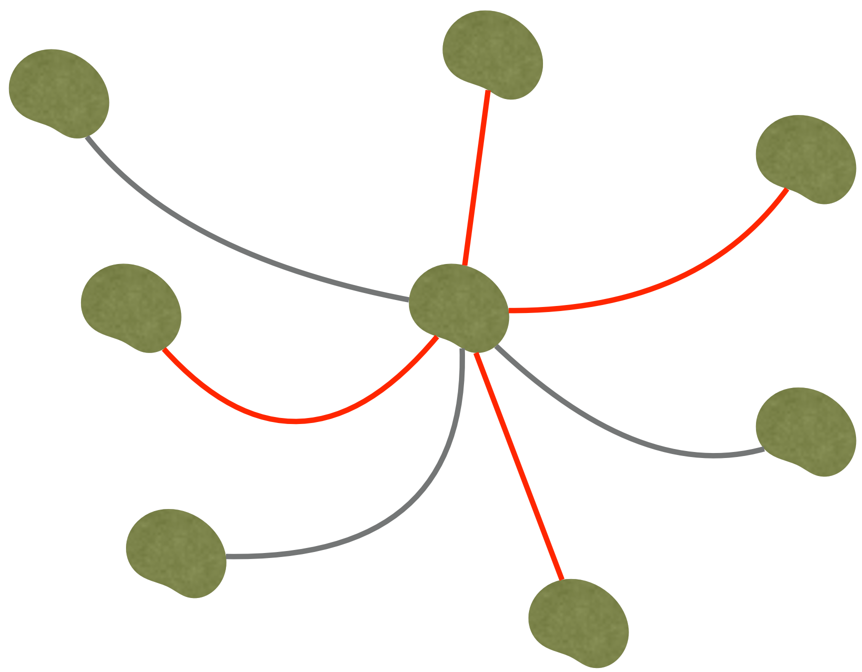
is loss of i observed?

true loss

probability of picking i

probability of side observation

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

- picks a node I_t
- suffers loss for I_t
- receives feedback
  - for I_t
  - for every other node with probability r_t

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

is loss of i observed?

true loss

probability of picking i

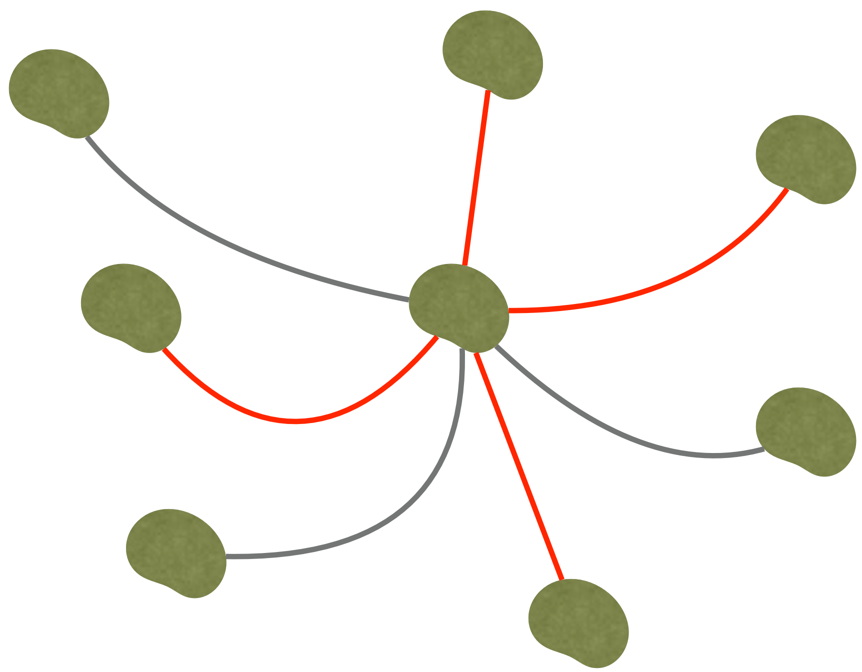
probability of side observation

Regret of Exp3-SET (Alon et al. 2013):

$$\mathcal{O}\left(\sqrt{\textstyle\sum_t (1/r_t)(1-(1-r_t)^N)\log N}\right)$$

How to estimate r_t in every round when it is changing?

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

- picks a node I_t
- suffers loss for I_t
- receives feedback
  - for I_t
  - for every other node with probability r_t

Regret of Exp3-SET (Alon et al. 2013):

$$\mathcal{O}\left(\sqrt{\sum_t (1/r_t)(1-(1-r_t)^N)\log N}\right)$$

How to estimate r_t in every round when it is changing?

How to estimate losses without the knowledge of r_t?

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1-p_{t,i})r_t}$$

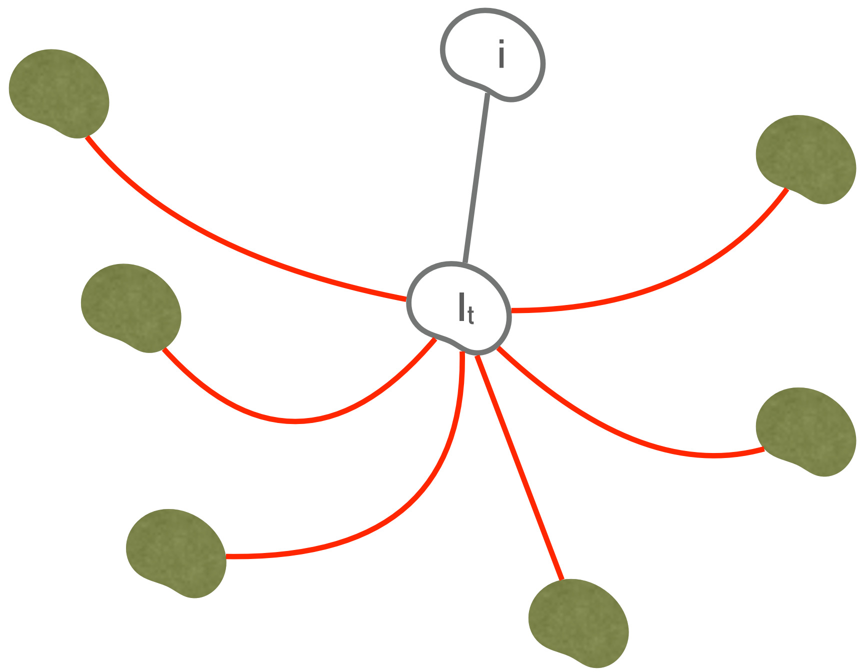
is loss of i observed?

true loss

probability of picking i

probability of side observation

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

is loss of i observed?

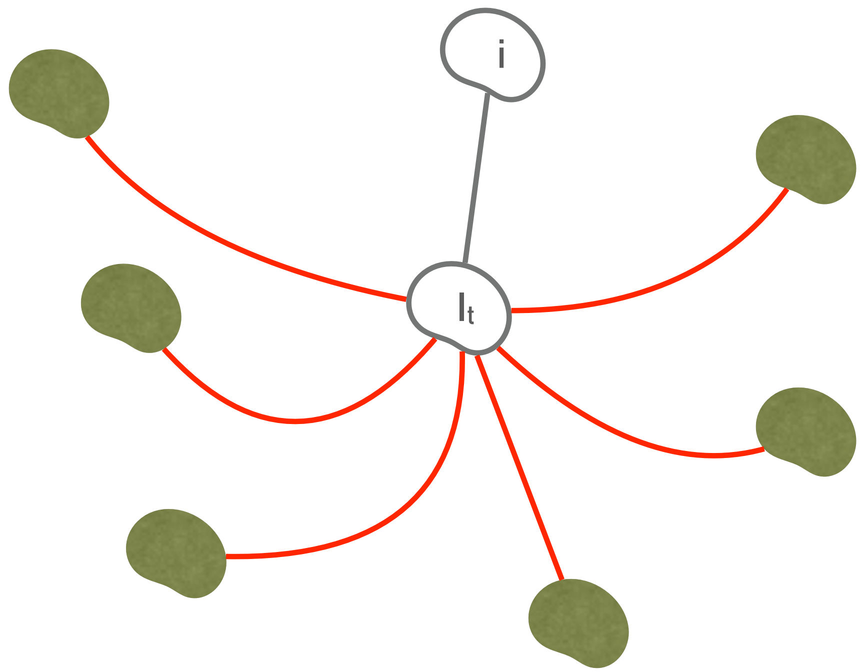
true loss

probability of picking i

probability of side observation

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS

- N-2 samples from Bernoulli(r_t) ... R(k)



is loss of i observed?

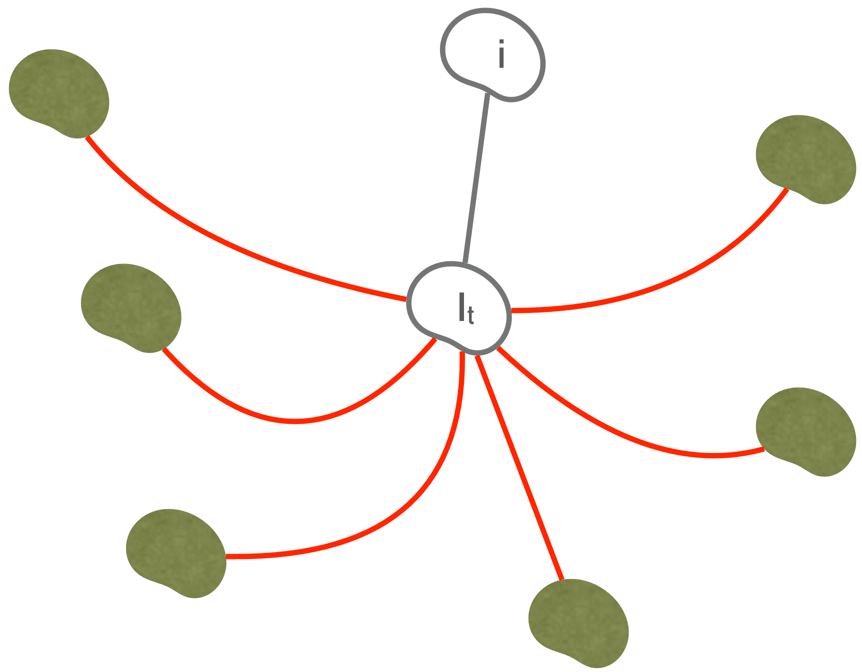
true loss

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

probability of picking i

probability of side observation

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



- N-2 samples from Bernoulli(r_t) ... R(k)
- N-2 samples from p_{ti} ... P(k)

is loss of i observed?

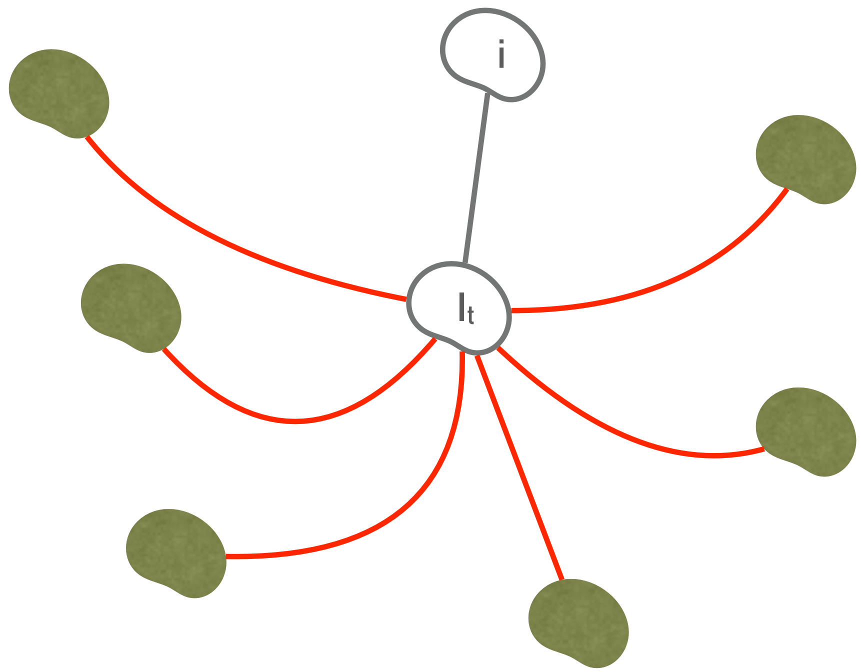
true loss

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

probability of picking i

probability of side observation

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



- N-2 samples from Bernoulli(r_t) ... R(k)
- N-2 samples from p_{ti} ... P(k)
- O'(k) = P(k) + (1-P(k))R(k)

is loss of i observed?

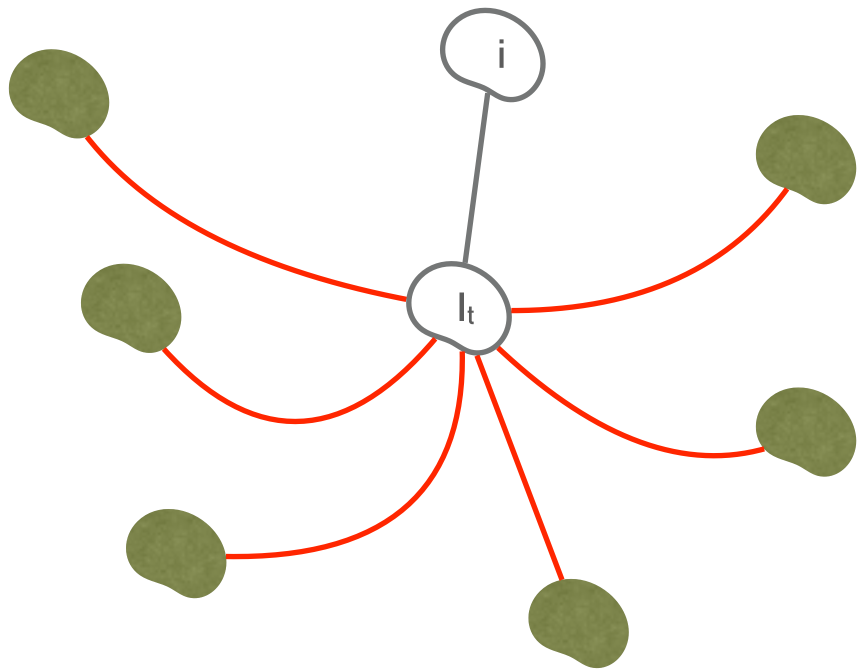
true loss

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

probability of picking i

probability of side observation

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



- N-2 samples from Bernoulli(r_t) ... R(k)
- N-2 samples from p_{ti} ... P(k)
- O'(k) = P(k) + (1-P(k))R(k)
- G_{ti} = min{k : O'(k) = 1} U {N-1}

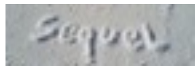
is loss of i observed?

true loss

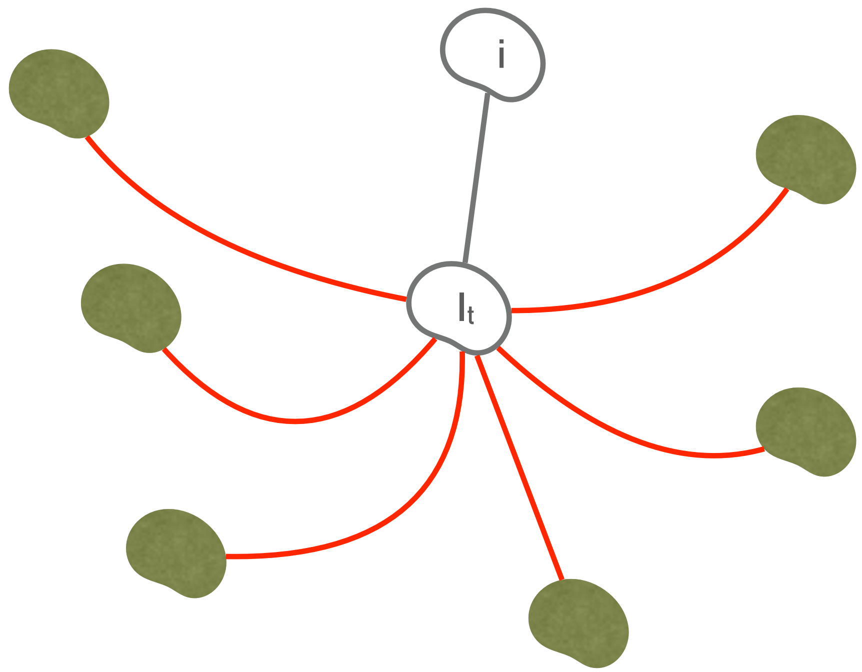
$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

probability of picking i

probability of side observation



PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



- N-2 samples from Bernoulli(r_t) ... R(k)
- N-2 samples from p_{ti} ... P(k)
- O'(k) = P(k) + (1-P(k))R(k)
- G_{ti} = min{k : O'(k) = 1} U {N-1}
- E[G_{ti}] ≈ 1/(p_{ti} +(1-p_{ti})r_t)

is loss of i observed?

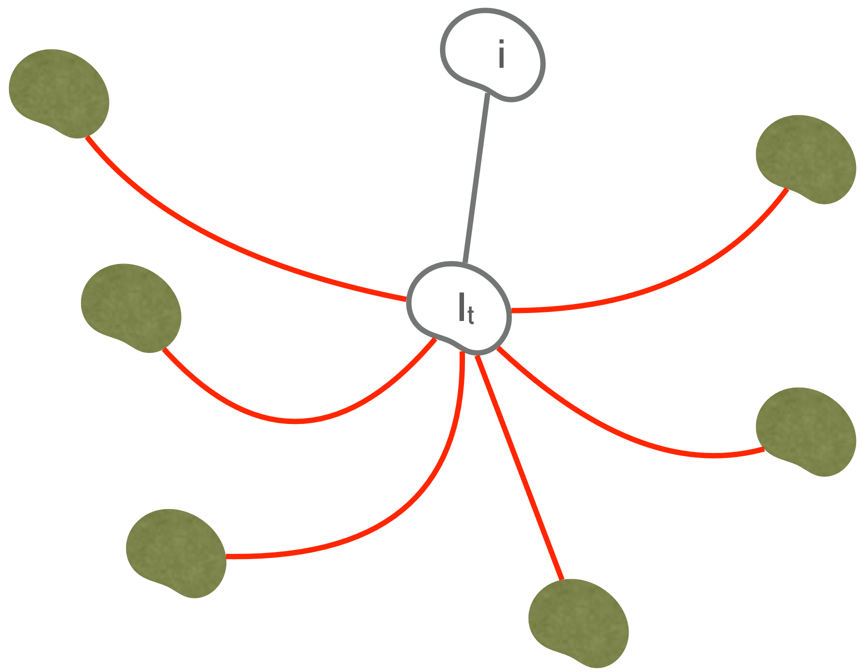
true loss

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

probability of picking i

probability of side observation

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



- N-2 samples from Bernoulli(r_t) ... R(k)
- N-2 samples from p_{ti} ... P(k)
- O'(k) = P(k) + (1-P(k))R(k)
- G_{ti} = min{k : O'(k) = 1} U {N-1}
- E[G_{ti}] ≈ 1/(p_{ti} +(1-p_{ti})r_t)

$$\widehat{\ell}_{t,i} = G_{t,i} O_{t,i} \ell_{t,i}$$

is loss of i observed?

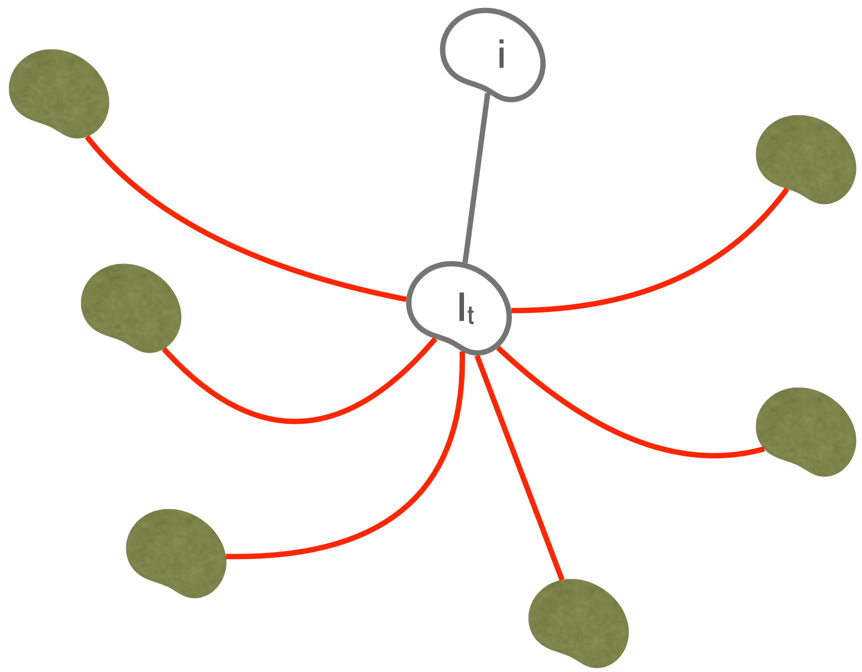
true loss

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

probability of picking i

probability of side observation

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



- N-2 samples from Bernoulli(r_t) ... R(k)
- N-2 samples from p_{ti} ... P(k)
- O'(k) = P(k) + (1-P(k))R(k)
- G_{ti} = min{k : O'(k) = 1} U {N-1}
- $E[G_{ti}] \approx 1/(p_{ti} + (1-p_{ti})r_t)$

$$\widehat{\ell}_{t,i} = G_{t,i} O_{t,i} \ell_{t,i}$$

If $r_t \geq (\log T)/(2N-2)$ then

$$\mathcal{O}\left(\sqrt{\log N \sum_{t=1}^{T} \frac{1}{r_t}}\right)$$

is loss of i observed?

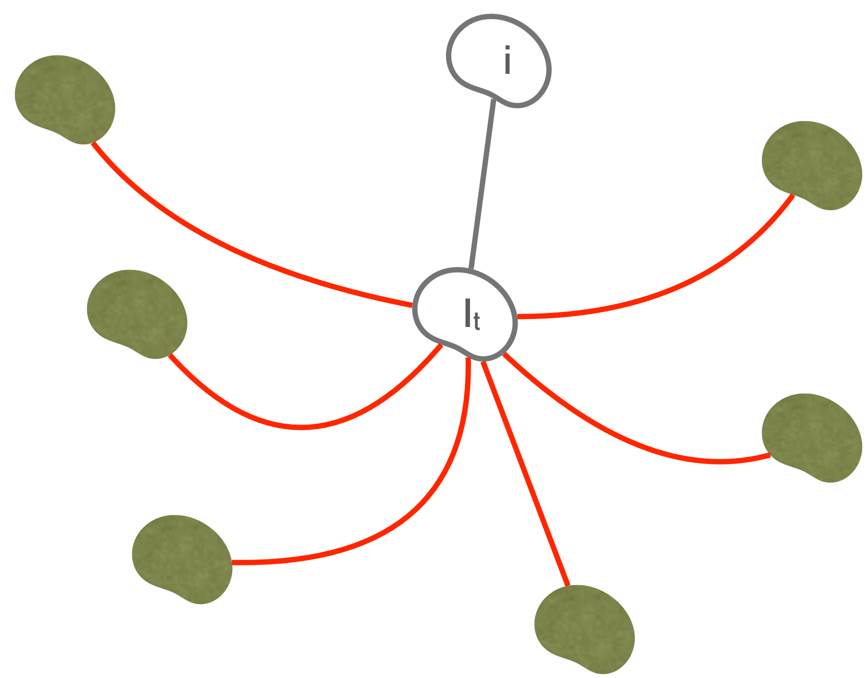
true loss

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

probability of picking i

probability of side observation

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



- N-2 samples from Bernoulli(r_t) ... R(k)
- N-2 samples from p_{ti} ... P(k)
- O'(k) = P(k) + (1-P(k))R(k)
- G_{ti} = min{k : O'(k) = 1} U {N-1}
- E[G_{ti}] ≈ 1/(p_{ti} + (1-p_{ti})r_t)

$$\widehat{\ell}_{t,i} = G_{t,i} O_{t,i} \ell_{t,i}$$

is loss of i observed?

true loss

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i}) r_t}$$

probability of picking i

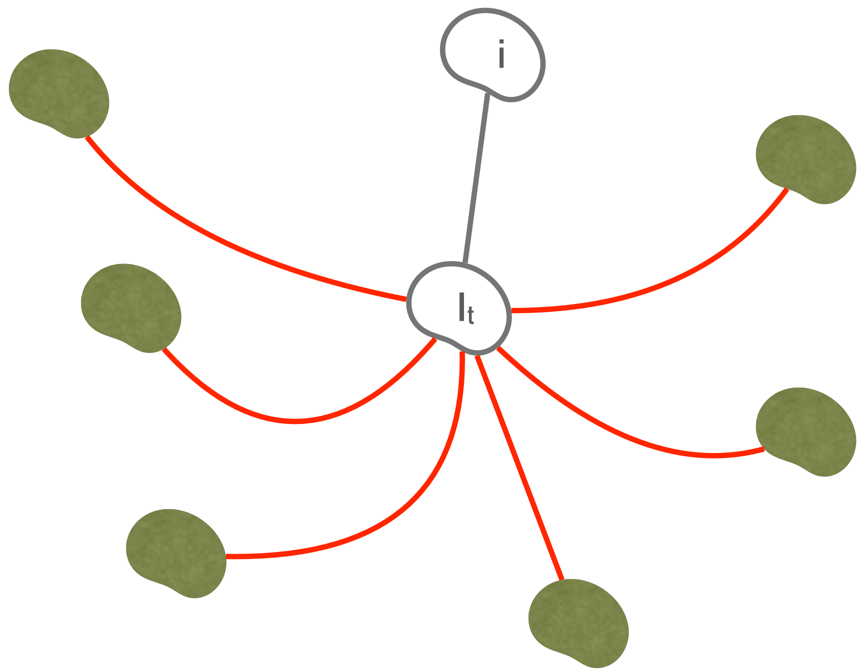
probability of side observation

If $r_t \geq (\log T)/(2N-2)$ then

$$\mathcal{O}\left(\sqrt{\log N \sum_{t=1}^{T} \frac{1}{r_t}}\right)$$

Lower bound (Alon et al. 2013) $\Omega(\sqrt{T/r})$

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



- N-2 samples from Bernoulli(r_t) ... R(k)
- N-2 samples from p_{ti} ... P(k)
- O'(k) = P(k) + (1-P(k))R(k)
- G_{ti} = min{k : O'(k) = 1} U {N-1}
- E[G_{ti}] ≈ 1/(p_{ti} + (1-p_{ti})r_t)

$$\widehat{\ell}_{t,i} = G_{t,i} O_{t,i} \ell_{t,i}$$

is loss of i observed?

true loss

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

probability of picking i

probability of side observation

If r_t ≥ (log T)/(2N-2) then

$$\mathcal{O}\left(\sqrt{\log N \sum_{t=1}^{T} \frac{1}{r_t}}\right)$$

Lower bound (Alon et al. 2013) $\Omega(\sqrt{T/r})$

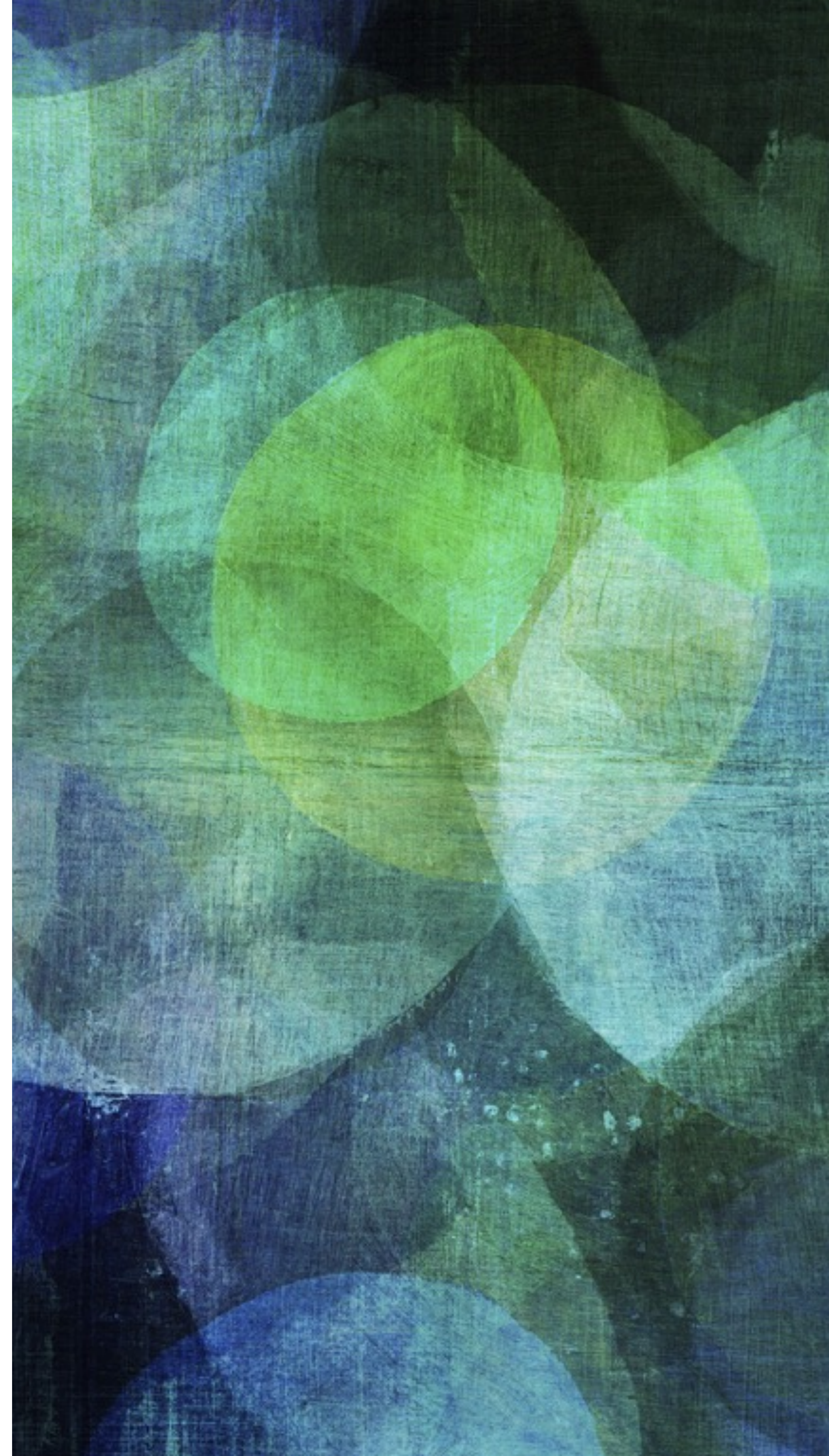
Get rid of r_t ≥ (log T)/(2N-2)?

Carpentier, MV: **Revealing Graph Bandits for Maximising Local Influence**, AISTATS 2016

Wen, Kveton, MV: **Influence Maximization with Semi-Bandit Feedback,** (arXiv:1605.06593)

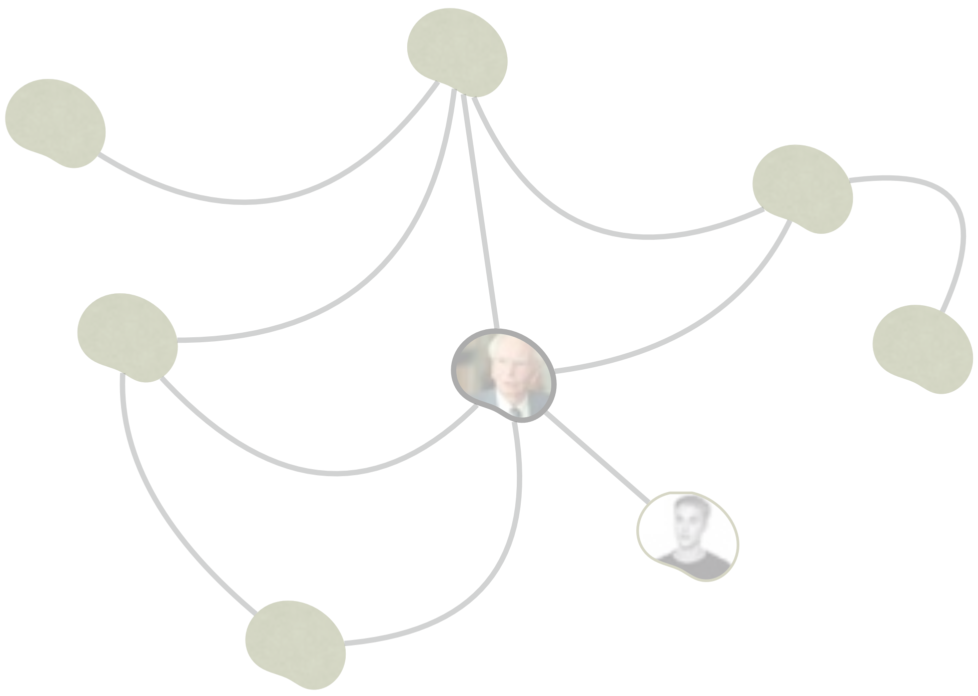
INFLUENCE MAXIMISATION

looking for the influential nodes **while** exploring the graph



REVEALING BANDITS FOR LOCAL INFLUENCE

Unknown $\mathbf{M} = (p_{i,j})_{i,j}$ symmetric matrix of influences

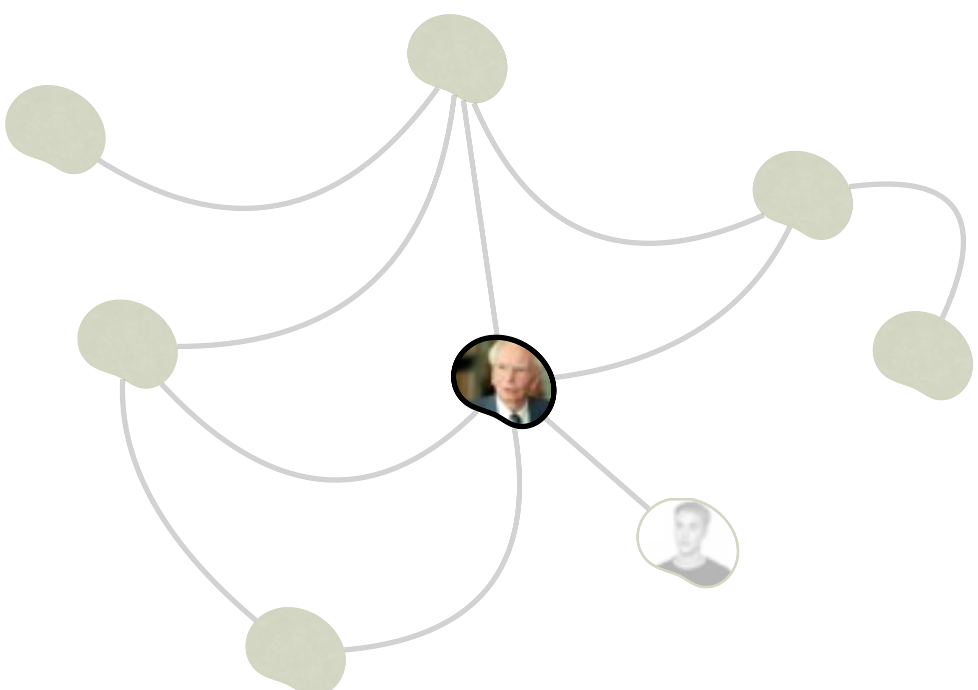


REVEALING BANDITS FOR LOCAL INFLUENCE

Unknown $\mathbf{M} = (p_{i,j})_{i,j}$ symmetric matrix of influences

In each time step $t = 1, \ldots, T$

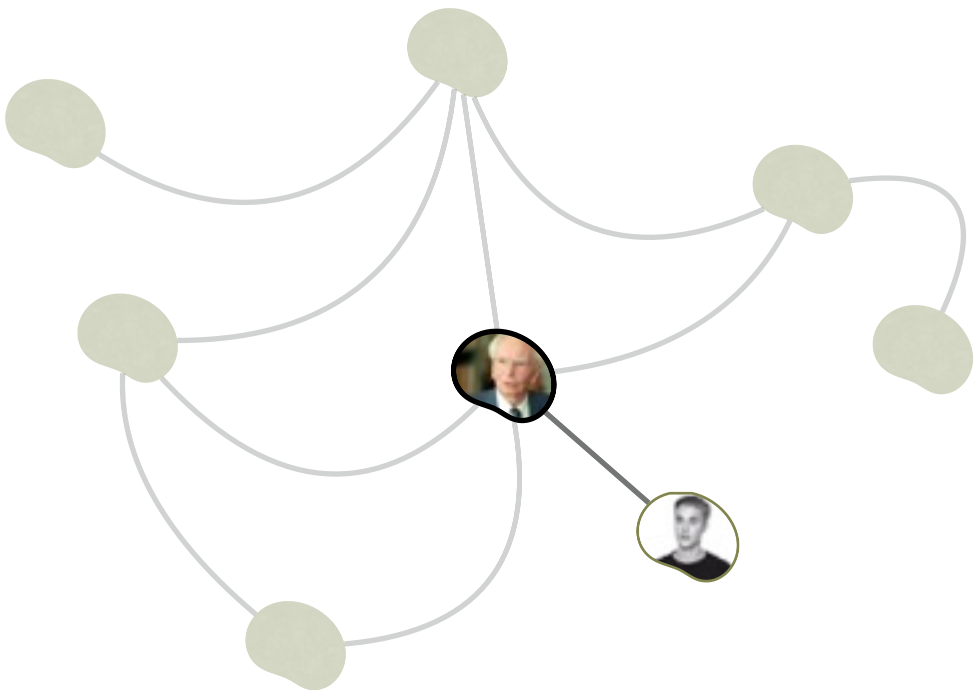
- learners picks a node k_t
- set $S_{k_t,t}$ of influenced nodes is *revealed*



Unknown $\mathbf{M} = (p_{i,j})_{i,j}$ symmetric matrix of influences

In each time step $t = 1, \ldots, T$

- learners picks a node k_t
- set $S_{k_t,t}$ of influenced nodes is *revealed*

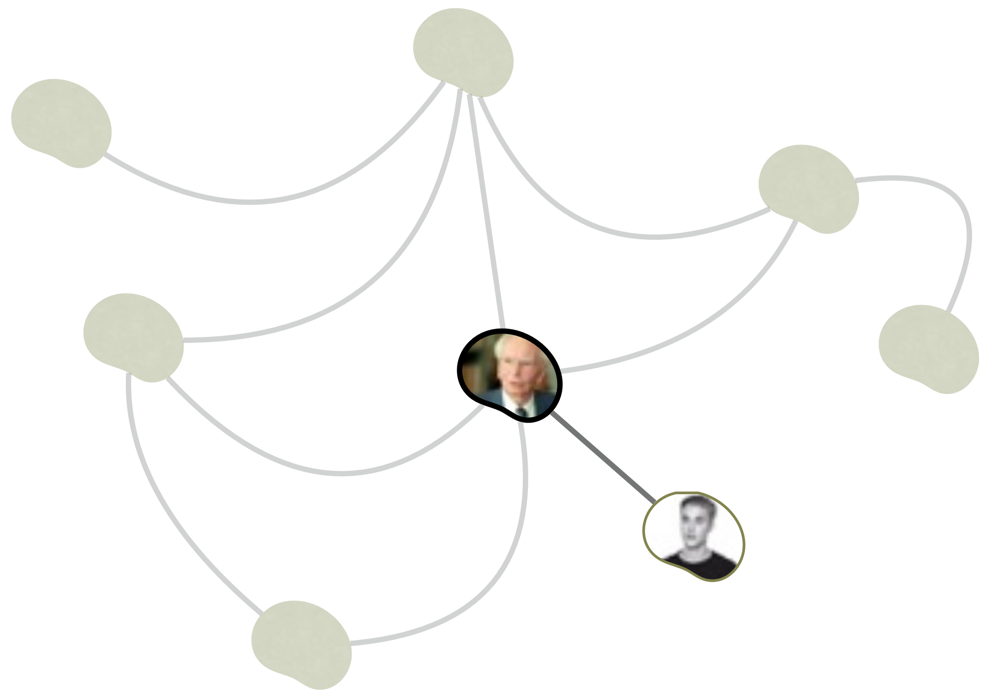


REVEALING BANDITS FOR LOCAL INFLUENCE

Unknown $\mathbf{M} = (p_{i,j})_{i,j}$ symmetric matrix of influences

In each time step $t = 1, \ldots, T$

- learners picks a node k_t
- set $S_{k_t,t}$ of influenced nodes is *revealed*



Select influential people = Find the strategy maximising

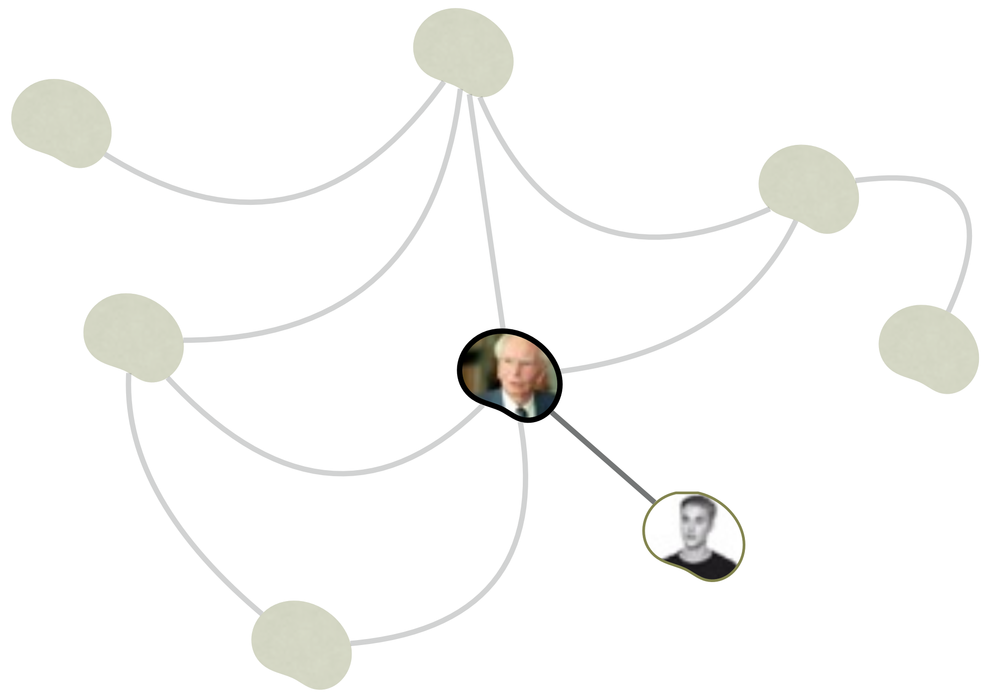
$$L_T = \sum_{t=1}^{T} |S_{k_t,t}|$$

REVEALING BANDITS FOR LOCAL INFLUENCE

Unknown $\mathbf{M} = (p_{i,j})_{i,j}$ symmetric matrix of influences

In each time step $t = 1, \ldots, T$

- learners picks a node k_t
- set $S_{k_t,t}$ of influenced nodes is *revealed*



Select influential people = Find the strategy maximising

$$L_T = \sum_{t=1}^{T} |S_{k_t,t}|$$

The number of expected influences of node **k** is by definition

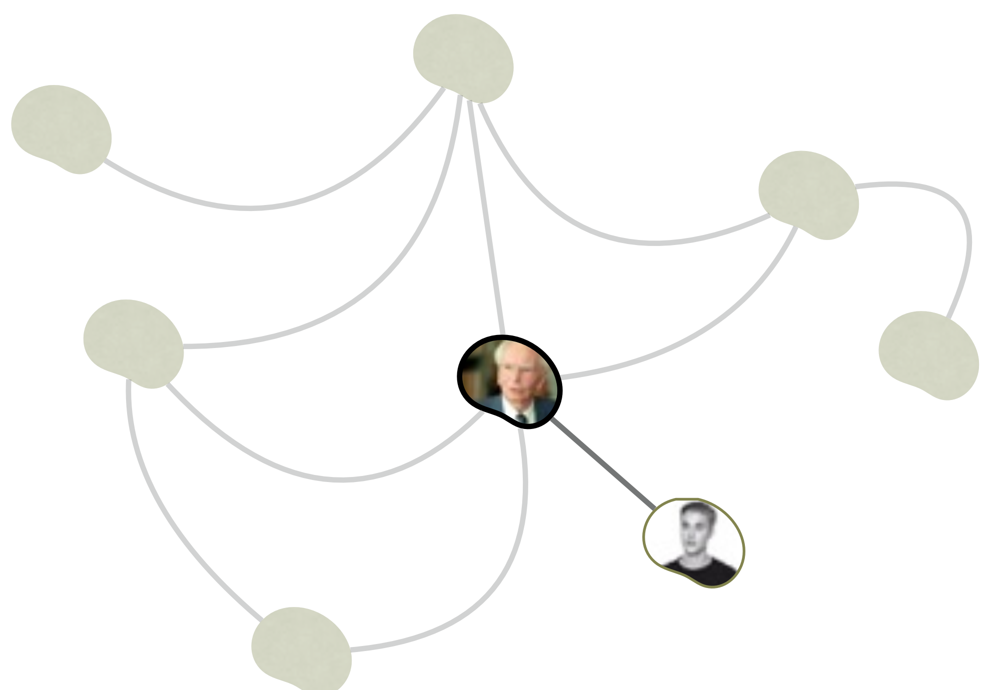
$$r_k = \mathbb{E}\left[|S_{k,t}|\right] = \sum_{j \leq N} p_{k,j}$$

REVEALING BANDITS FOR LOCAL INFLUENCE

Unknown $\mathbf{M} = (p_{i,j})_{i,j}$ symmetric matrix of influences

In each time step $t = 1, \ldots, T$

- learners picks a node k_t
- set $S_{k_t,t}$ of influenced nodes is *revealed*



Select influential people = Find the strategy maximising

$$L_T = \sum_{t=1}^{T} |S_{k_t,t}|$$

The number of expected influences of node **k** is by definition

$$r_k = \mathbb{E}\left[|S_{k,t}|\right] = \sum_{j \leq N} p_{k,j}$$

Oracle strategy always selects the best

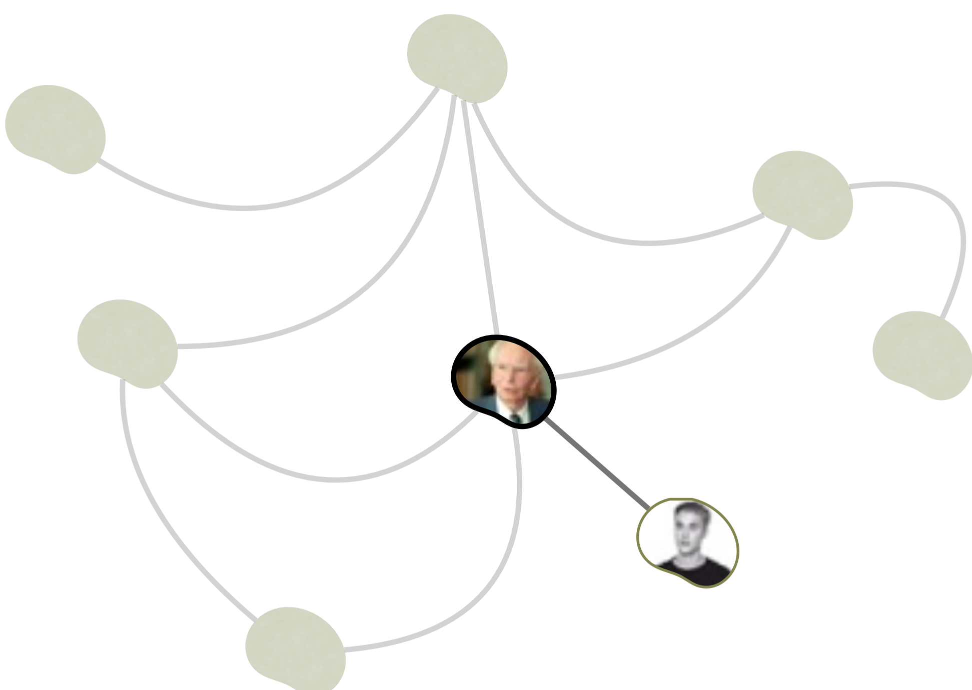
$$k^* = \arg\max_k \mathbb{E}\left[\sum_{t=1}^{T} |S_{k,t}|\right] = \arg\max_k Tr_k$$

REVEALING BANDITS FOR LOCAL INFLUENCE

Unknown $\mathbf{M} = (p_{i,j})_{i,j}$ symmetric matrix of influences

In each time step $t = 1, \ldots, T$

- learners picks a node k_t
- set $S_{k_t,t}$ of influenced nodes is *revealed*



Select influential people = Find the strategy maximising

$$L_T = \sum_{t=1}^{T} |S_{k_t,t}|$$

The number of expected influences of node **k** is by definition

$$r_k = \mathbb{E}\left[|S_{k,t}|\right] = \sum_{j \leq N} p_{k,j}$$

Oracle strategy always selects the best

$$k^* = \arg\max_k \mathbb{E}\left[\sum_{t=1}^{T} |S_{k,t}|\right] = \arg\max_k Tr_k$$

Expected regret of any adaptive, non-oracle strategy unaware of **M**

$$\mathbb{E}\left[R_T\right] = \mathbb{E}\left[L_T^*\right] - \mathbb{E}\left[L_T\right]$$

REVEALING BANDITS

REVEALING BANDITS

- Ignoring the structure again?

REVEALING BANDITS

- Ignoring the structure again? $\widetilde{\mathcal{O}}\left(\sqrt{r_* T N}\right)$

reward of the best node

REVEALING BANDITS

- Ignoring the structure again? $\widetilde{\mathcal{O}}\left(\sqrt{r_* T N}\right)$
- **BAndit REvelator:** 2-phase algorithm

reward of the best node

REVEALING BANDITS

- Ignoring the structure again? $\widetilde{\mathcal{O}}\left(\sqrt{r_* T N}\right)$

reward of the best node

- **BAndit REvelator:** 2-phase algorithm
- **global** exploration phase
  - super-efficient exploration
  - linear regret — needs to be short!
  - extracts **D_*** nodes

REVEALING BANDITS

- Ignoring the structure again? $\widetilde{\mathcal{O}}\left(\sqrt{r_* T N}\right)$

reward of the best node

- **BAndit REvelator:** 2-phase algorithm
- **global** exploration phase
  - super-efficient exploration
  - linear regret — needs to be short!
  - extracts $\mathbf{D}_*$ nodes
- **bandit** phase
  - uses a minimax-optimal bandit algorithm (GraphMOSS)
  - has a “square root” regret on $\mathbf{D}_*$ nodes

REVEALING BANDITS

- Ignoring the structure again? $\widetilde{\mathcal{O}}\left(\sqrt{r_* T N}\right)$

  reward of the best node

- **BAndit REvelator:** 2-phase algorithm
- **global** exploration phase
  - super-efficient exploration
  - linear regret — needs to be short!
  - extracts **D**$_*$ nodes
- **bandit** phase
  - uses a minimax-optimal bandit algorithm (GraphMOSS)
  - has a "square root" regret on **D**$_*$ nodes
- **D**$_*$ realizes the optimal trade-off !
  - different from exploration/exploitation tradeoff

REVEALING BANDITS

- Ignoring the structure again? $\widetilde{\mathcal{O}}\left(\sqrt{r_* T N}\right)$

  reward of the best node

- **BAndit REvelator:** 2-phase algorithm
- **global** exploration phase
  - super-efficient exploration
  - linear regret — needs to be short!
  - extracts **D**$_*$ nodes
- **bandit** phase
  - uses a minimax-optimal bandit algorithm (GraphMOSS)
  - has a "square root" regret on **D**$_*$ nodes
- **D**$_*$ realizes the optimal trade-off !
  - different from exploration/exploitation tradeoff

Regret of BARE

$$\widetilde{\mathcal{O}}\left(\sqrt{r_* T D_*}\right)$$

REVEALING BANDITS

- Ignoring the structure again? $\tilde{\mathcal{O}}\left(\sqrt{r_* T N}\right)$
  - reward of the best node
- **BAndit REvelator:** 2-phase algorithm
- **global** exploration phase
  - super-efficient exploration
  - linear regret — needs to be short!
  - extracts D_* nodes
- **bandit** phase
  - uses a minimax-optimal bandit algorithm (GraphMOSS)
  - has a "square root" regret on D_* nodes
- D_* realizes the optimal trade-off !
  - different from exploration/exploitation tradeoff

Regret of BARE

$$\tilde{\mathcal{O}}\left(\sqrt{r_* T D_*}\right)$$

- **D*** - detectable dimension (depends on T and the structure)
  - **good case**: star-shaped graph can have D* = 1
  - **bad case:** a graph with many small cliques.
  - **the worst case:** all nodes are disconnected except 2

EMPIRICAL RESULTS

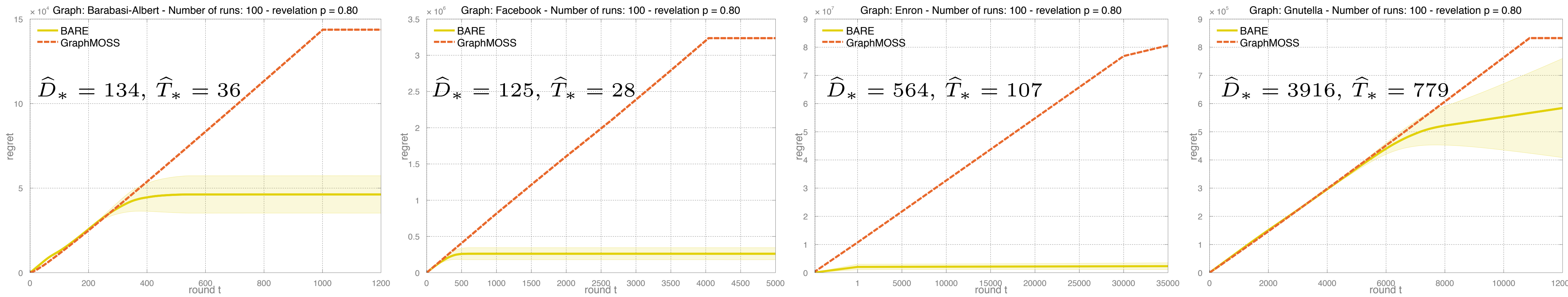


Figure 1: *Left*: Barabási-Albert. *Middle left*: Facebook. *Middle right*: Enron. *Right*: Gnutella.

- Enron and Facebook **vs.** Gnutella (decentralised)

GLOBAL INFLUENCE MODELS

GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015: **Independence Cascades**, Linear Threshold models

GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015:
  - **global** and **multiple-source** models

GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015:
  - global
- Different feed-back models

GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015:
  - global
- Different feed-back models
  - **Full bandit** (only the number of influenced nodes)

GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015:
  - global
- Different feed-back models
  - Full bandit
  - **Node-level semi-bandit** (identities of influenced nodes)

GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015:
  - global
- Different feed-back models
  - Full bandit
  - Node-level semi-bandit
  - **Edge-level semi-bandit** (identities of influenced edges)

GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015:
  - global
- Different feed-back models
  - Full bandit
  - Node-level semi-bandit
  - Edge-level semi-bandit
    - http://arxiv.org/abs/1605.06593 (Wen, Kveton, MV)

GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015:
  - global
- Different feed-back models
  - Full bandit
  - Node-level semi-bandit
  - Edge-level semi-bandit
    - http://arxiv.org/abs/1605.06593
    - IMLinUCB with linear parametrization of edge weights

GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015:
  - global
- Different feed-back models
  - Full bandit
  - Node-level semi-bandit
  - Edge-level semi-bandit
    - http://arxiv.org/abs/1605.06593
    - IMLinUCB with linear parametrization of edge weights
    - Regret analysis for subset of graphs (forests, circles, ...)

GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015:
  - global
- Different feed-back models
  - Full bandit
  - Node-level semi-bandit
  - Edge-level semi-bandit
    - http://arxiv.org/abs/1605.06593
    - IMLinUCB with linear parametrization of edge weights
    - Regret analysis for subset of graphs (forests, circles, ...)

$$\tilde{O}\left(dC_*\sqrt{E_* n}\right)$$

number of edge features

maximum cardinality of the reachable edges

maximum observed relevance

CONCLUSION AND NEW DIRECTIONS

CONCLUSION AND NEW DIRECTIONS

Graph Bandits

CONCLUSION AND NEW DIRECTIONS

Graph Bandits

- specific way of exploiting the problem **structure** to learn **faster**

CONCLUSION AND NEW DIRECTIONS

Graph Bandits

- specific way of exploiting the problem **structure** to learn **faster**
  - smooth rewards - **spectral** bandits, **cheap** bandits

CONCLUSION AND NEW DIRECTIONS

Graph Bandits

- specific way of exploiting the problem **structure** to learn **faster**
  - smooth rewards - **spectral** bandits, **cheap** bandits
  - (noisy) side observations - **informed** bandits

CONCLUSION AND NEW DIRECTIONS

Graph Bandits

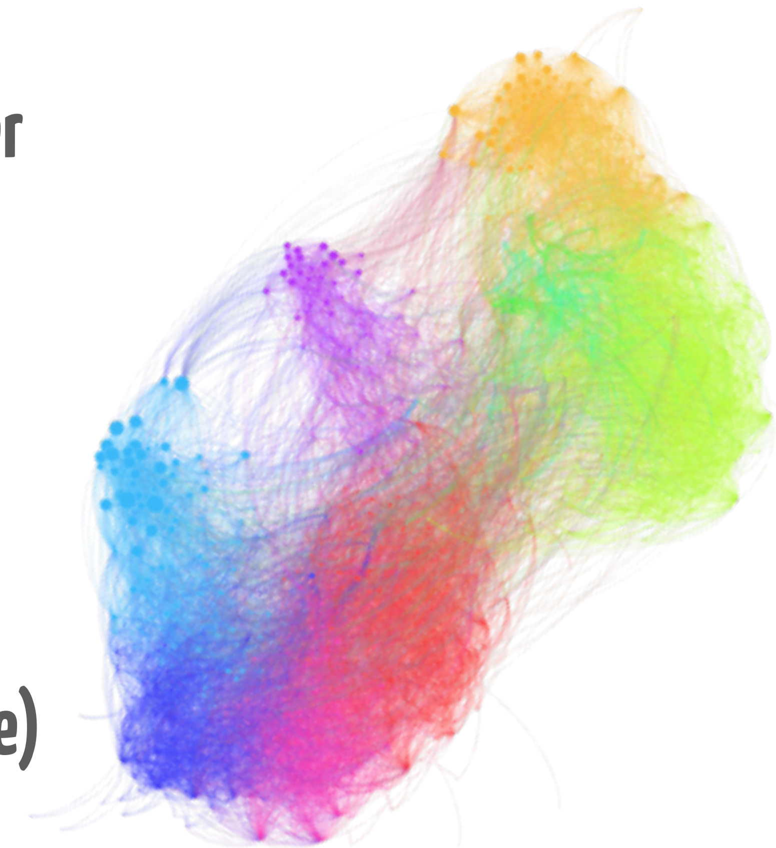
- specific way of exploiting the problem **structure** to learn **faster**
    - smooth rewards - **spectral** bandits, **cheap** bandits
    - (noisy) side observations - **informed** bandits
    - influence maximisation - **revealing** bandits

CONCLUSION AND NEW DIRECTIONS

Graph Bandits

- specific way of exploiting the problem **structure** to learn **faster**
  - smooth rewards - **spectral** bandits, **cheap** bandits
  - (noisy) side observations - **informed** bandits
  - influence maximisation - **revealing** bandits

New directions (reducing assumption on graph knowledge even more)



CONCLUSION AND NEW DIRECTIONS

Graph Bandits

- specific way of exploiting the problem **structure** to learn **faster**
  - smooth rewards - **spectral** bandits, **cheap** bandits
  - (noisy) side observations - **informed** bandits
  - influence maximisation - **revealing** bandits

New directions (reducing assumption on graph knowledge even more)

- graph generators (BA, ...), learning (with) communities - SBM

CONCLUSION AND NEW DIRECTIONS

Graph Bandits

- specific way of exploiting the problem **structure** to learn **faster**
  - smooth rewards - **spectral** bandits, **cheap** bandits
  - (noisy) side observations - **informed** bandits
  - influence maximisation - **revealing** bandits

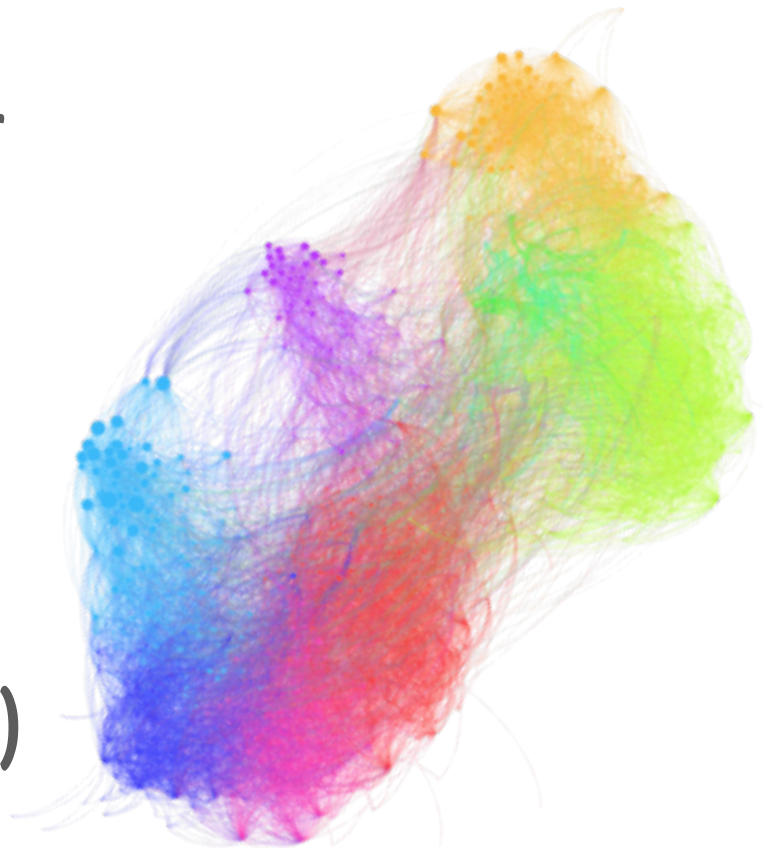
New directions (reducing assumption on graph knowledge even more)

- graph generators (BA, ...), learning (with) communities - SBM
- crawling strategies (Krause et al.)

CONCLUSION AND NEW DIRECTIONS

Graph Bandits

- specific way of exploiting the problem **structure** to learn **faster**
  - smooth rewards - **spectral** bandits, **cheap** bandits
  - (noisy) side observations - **informed** bandits
  - influence maximisation - **revealing** bandits



New directions (reducing assumption on graph knowledge even more)

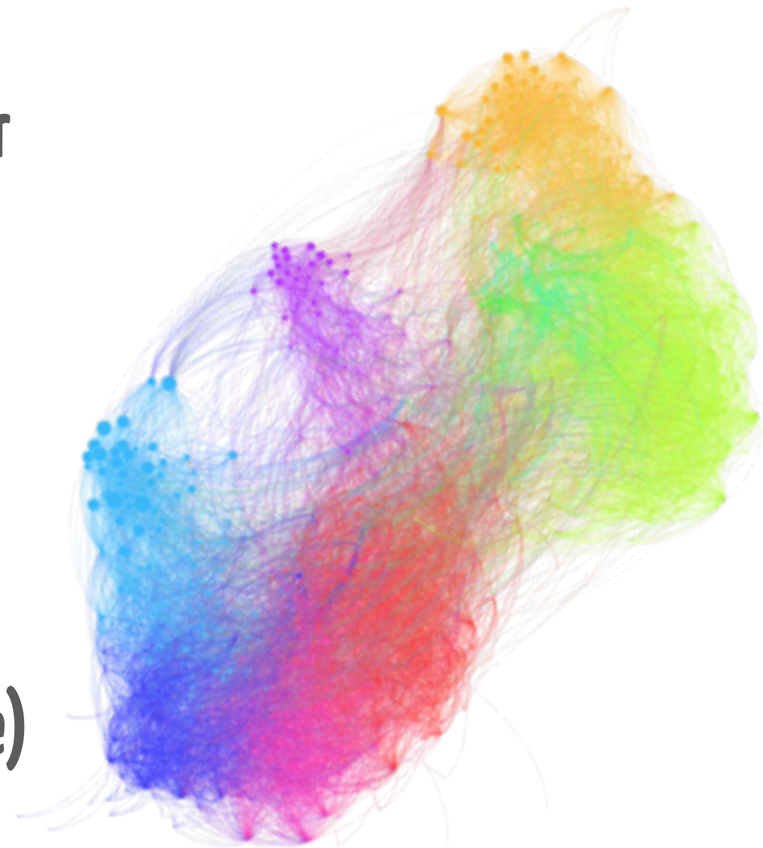
- graph generators (BA, ...), learning (with) communities - SBM
- crawling strategies (Krause et al.)

Not every structure is a graph

CONCLUSION AND NEW DIRECTIONS

Graph Bandits

- specific way of exploiting the problem **structure** to learn **faster**
  - smooth rewards - **spectral** bandits, **cheap** bandits
  - (noisy) side observations - **informed** bandits
  - influence maximisation - **revealing** bandits



New directions (reducing assumption on graph knowledge even more)

- graph generators (BA, ...), learning (with) communities - SBM
- crawling strategies (Krause et al.)

Not every structure is a graph

- some examples: polymatroids, kernels, (smooth) functions, no-topology structures

JOINT WORK WITH…

Inria informatics mathematics

Akram Erraqabi
U Montréal

Alessandro Lazaric
SequeL, Inria

Alexandra Carpentier
Universität Potsdam

Azin Ashkan
Technicolor Labs

Branislav Kveton
Adobe Research

Daniele Calandriello
SequeL, Inria

Gergely Neu
U Pompeu Fabra

Ilias Flaounas
Atlassian

Jean-Bastien Grill
SequeL, Inria

Julien Audiffren
CMLA, ENS Cachan

Manjesh Hanawal
Boston University

Mohammad Ghavamzadeh
Adobe Research

Nello Cristianini
University of Bristol

Odalric-Ambrym Maillard
Tao, Inria

Philippe Preux
SequeL, Inria

Rémi Munos
Google DeepMind

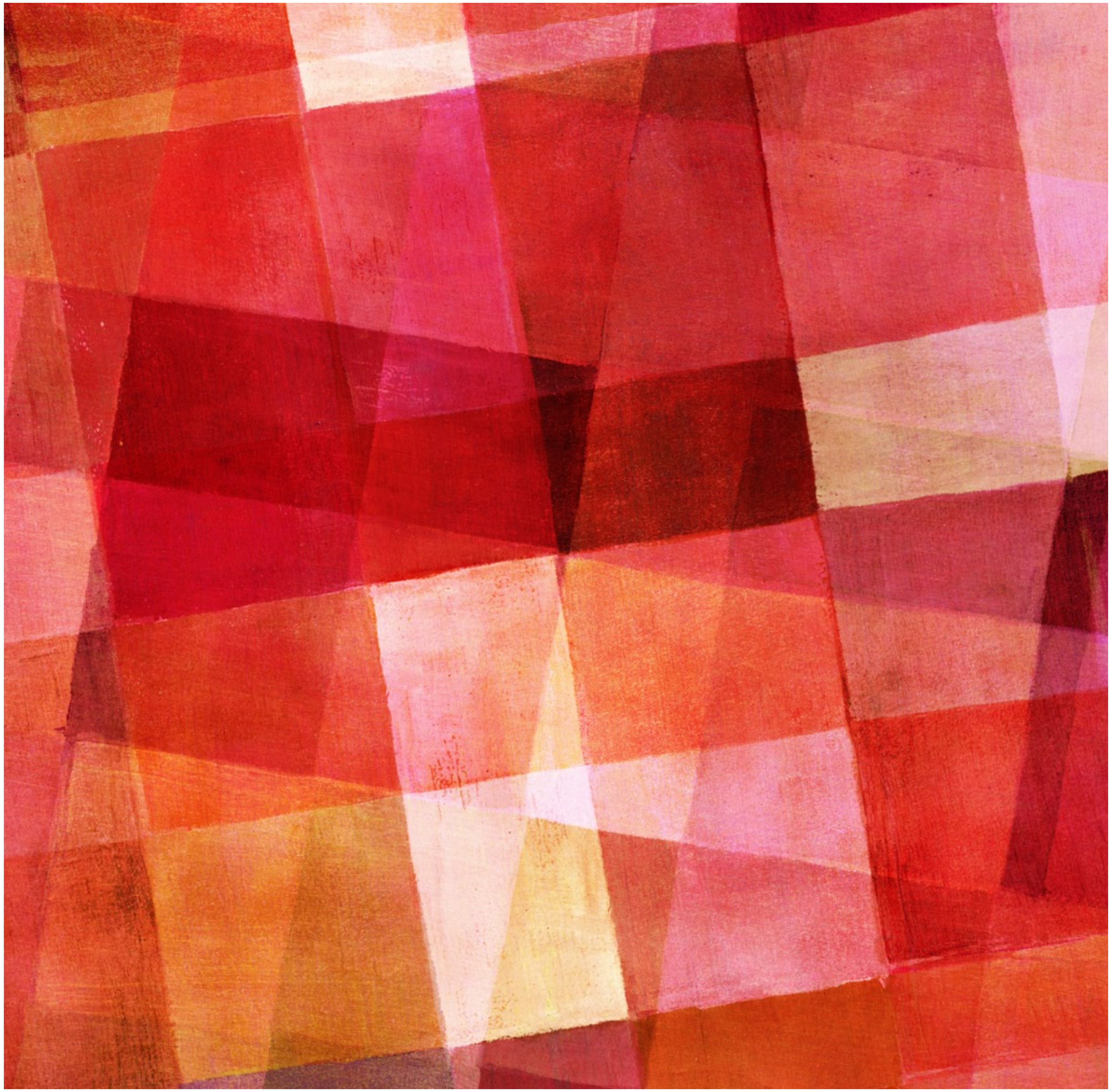
Shipra Agrawal
Columbia U

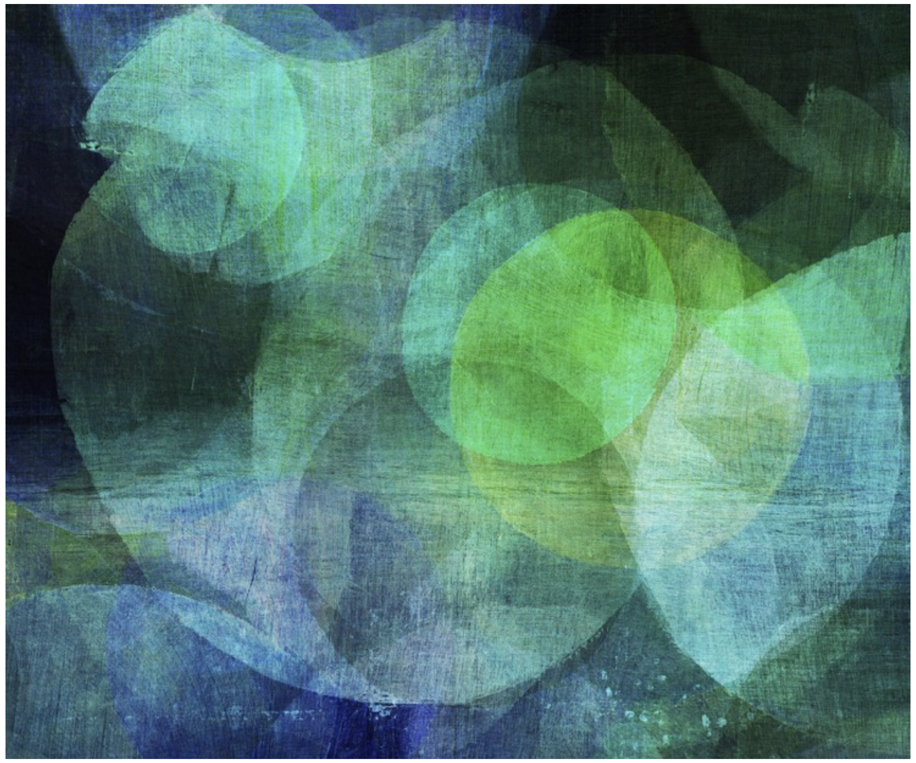
Tomáš Kocák
SequeL, Inria

Venkatesh Saligrama
Boston University

Yaakov Engel
Rafael

Zheng Wen
Adobe Research





Michal Valko, SequeL, Inria Lille - Nord Europe, michal.valko@inria.fr

http://researchers.lille.inria.fr/~valko/hp/