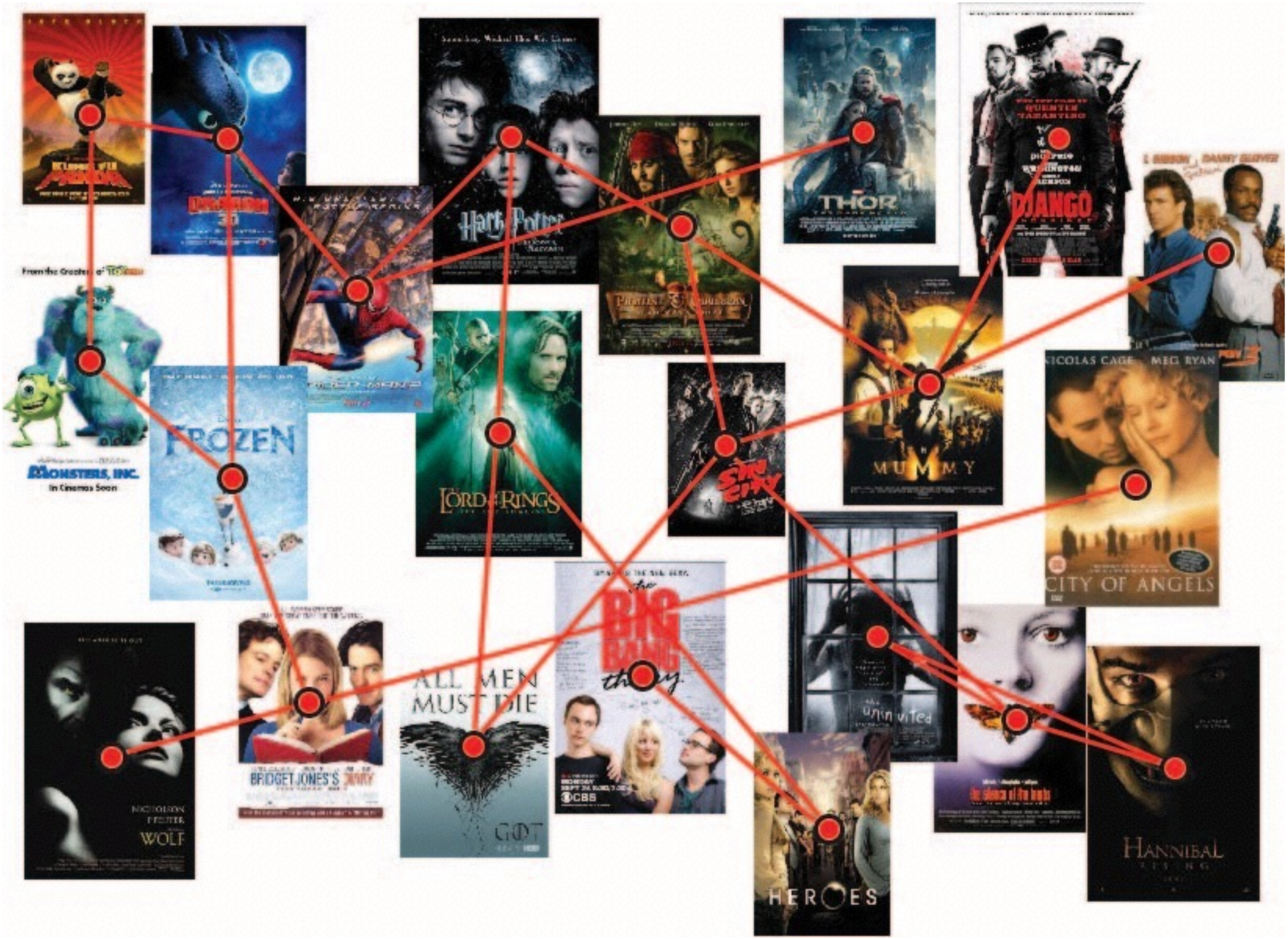
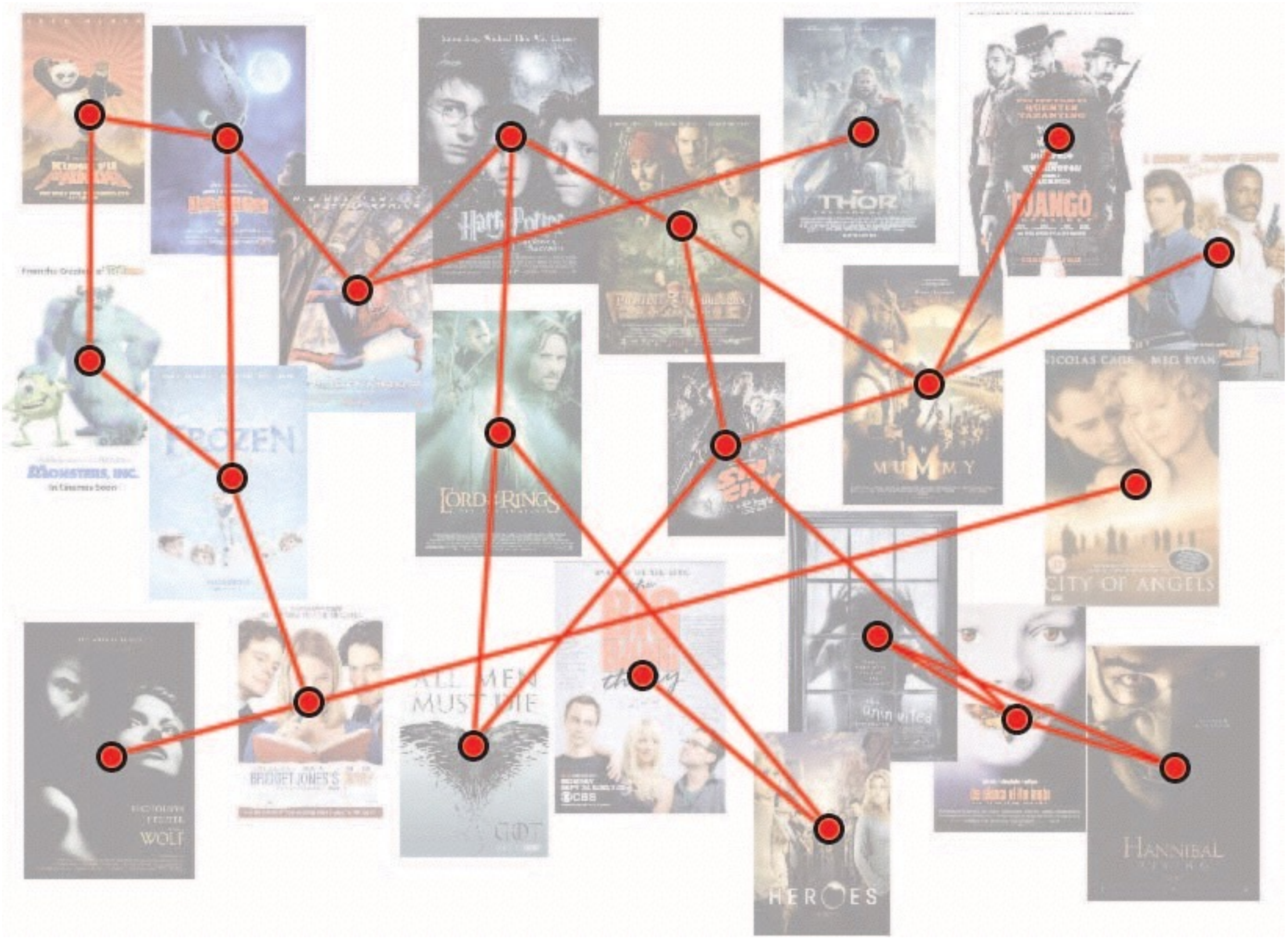
GRAPH BANDITS

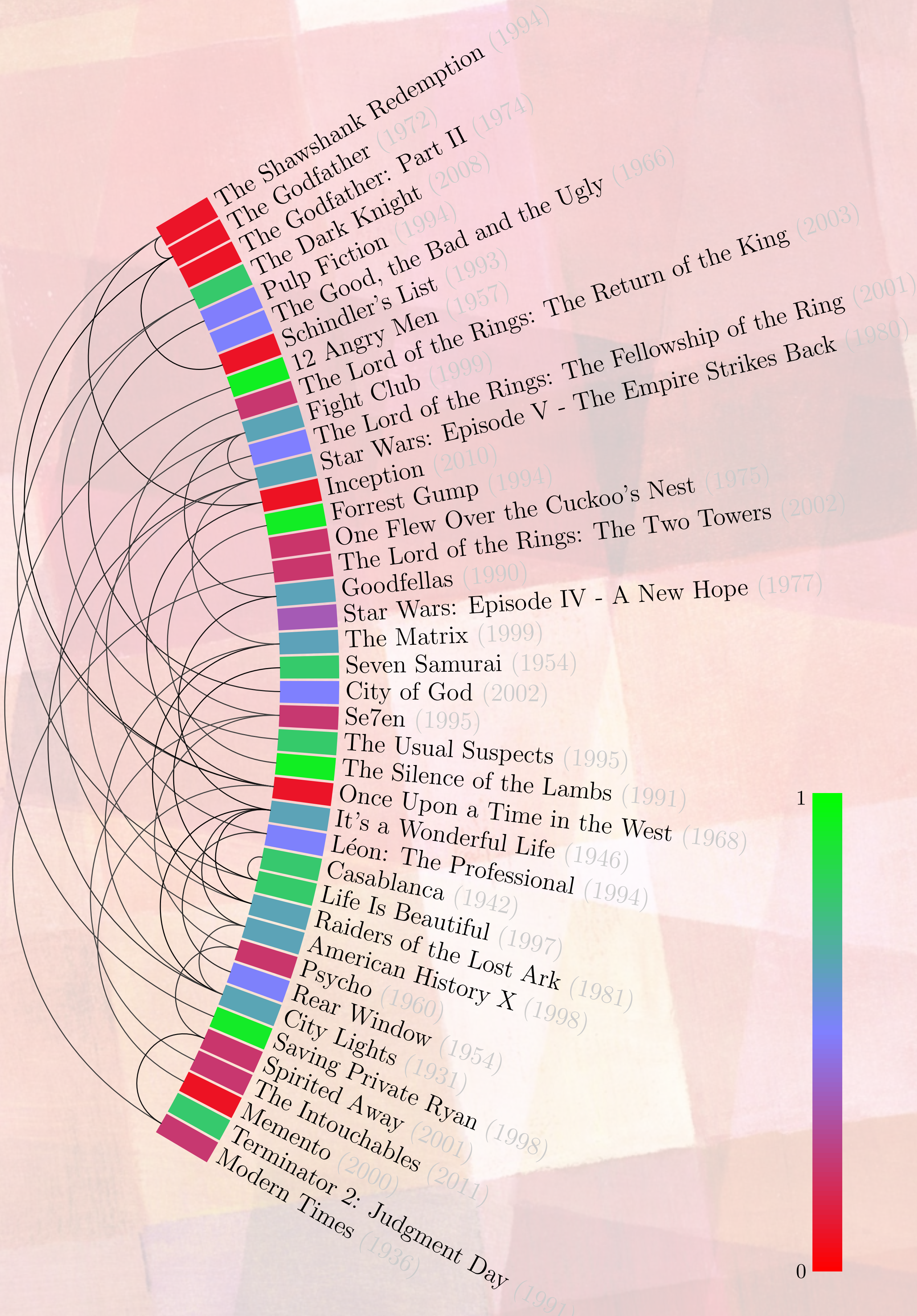
Michal Valko, SequeL, Inria Lille - Nord Europe


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



MONSTERS, INC.
In Cinemas Soon
FROZEN
Harry Potter
THOR
DJANGO
NICOLAS CAGE MEG RYAN
CITY OF ANGELS
MUMMY
ALL MEN MUST DIE
GOT
BRIDGET JONES'S DIARY
NICHOLSON
PFEIFFER
WOLF
HEROES
HANNIBAL
CBS


Harry Potter
THOR
DJANGO
NICOLAS CAGE MEG RYAN
CITY OF ANGELS
FROZEN
MONSTERS, INC.
LORD OF THE RINGS
MUMMY
ALL MEN MUST DIE
HEROES
HANNIBAL
WOLF



Example of a graph bandit problem

movie recommendation

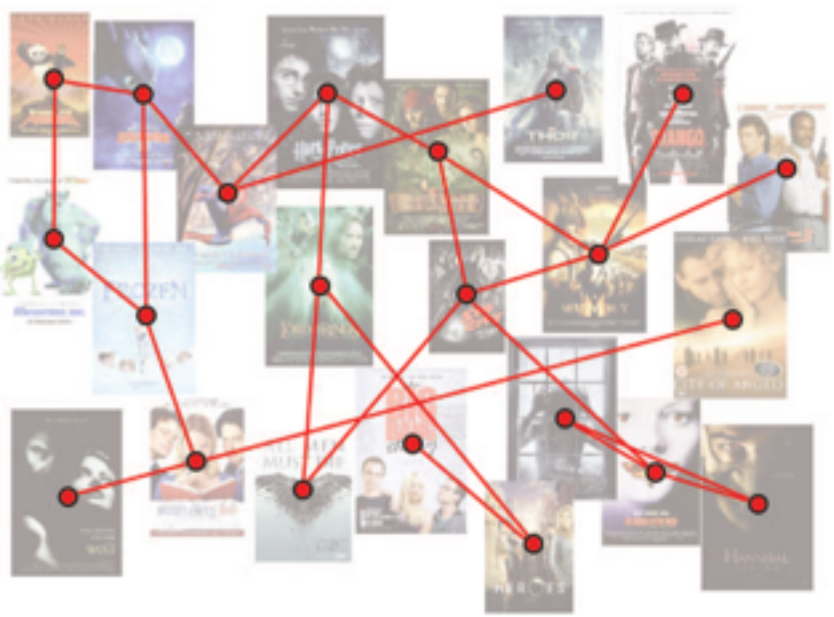
- recommend movies to a **single** user
- **goal:** maximise the sum of the ratings (minimise regret)
- good prediction after just a few steps

$$T \ll N$$

- extra information
  - ratings are **smooth** on a graph
- main question: can we learn **faster**?

GETTING REAL

Let's be lazy and ignore the structure



$\rightarrow$

Multi-armed bandit problem!

Worst case regret (to the best fixed strategy)

#actions

#rounds

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

How big is N? Number of movies on **http://www.imdb.com/stats**: 3,589,057

Problem: Too many actions!

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

- **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

JOINT WORK WITH...

Alexandra Carpentier
Universität Potsdam

Branislav Kveton
Adobe Research

Gergely Neu
Universitat Pompeu Fabra

Manjesh Hanawal
Boston University

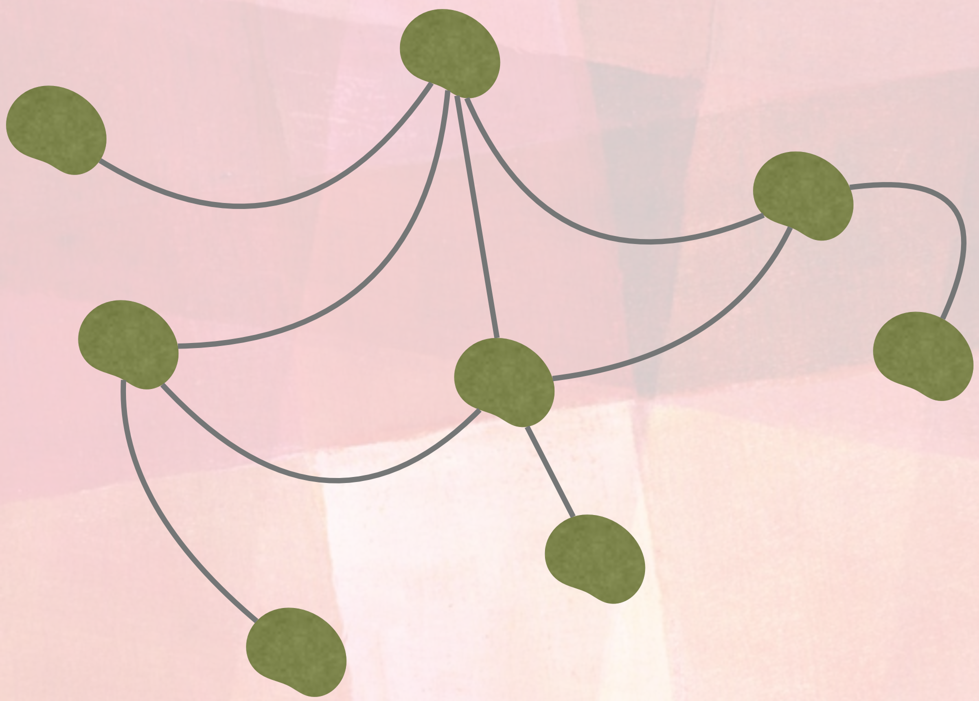
Rémi Munos
Google DeepMind

Shipra Agrawal
Columbia University

Tomáš Kocák
SequeL, Inria Lille
graduating in 2016

Venkatesh Saligrama
Boston University

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 setups differ in

1. loss
2. feedback
3. guarantees

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

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

independence number

influence maximisation
revealing bandits

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

detectable dimension

noisy side observations
on graphs

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

effective independence number

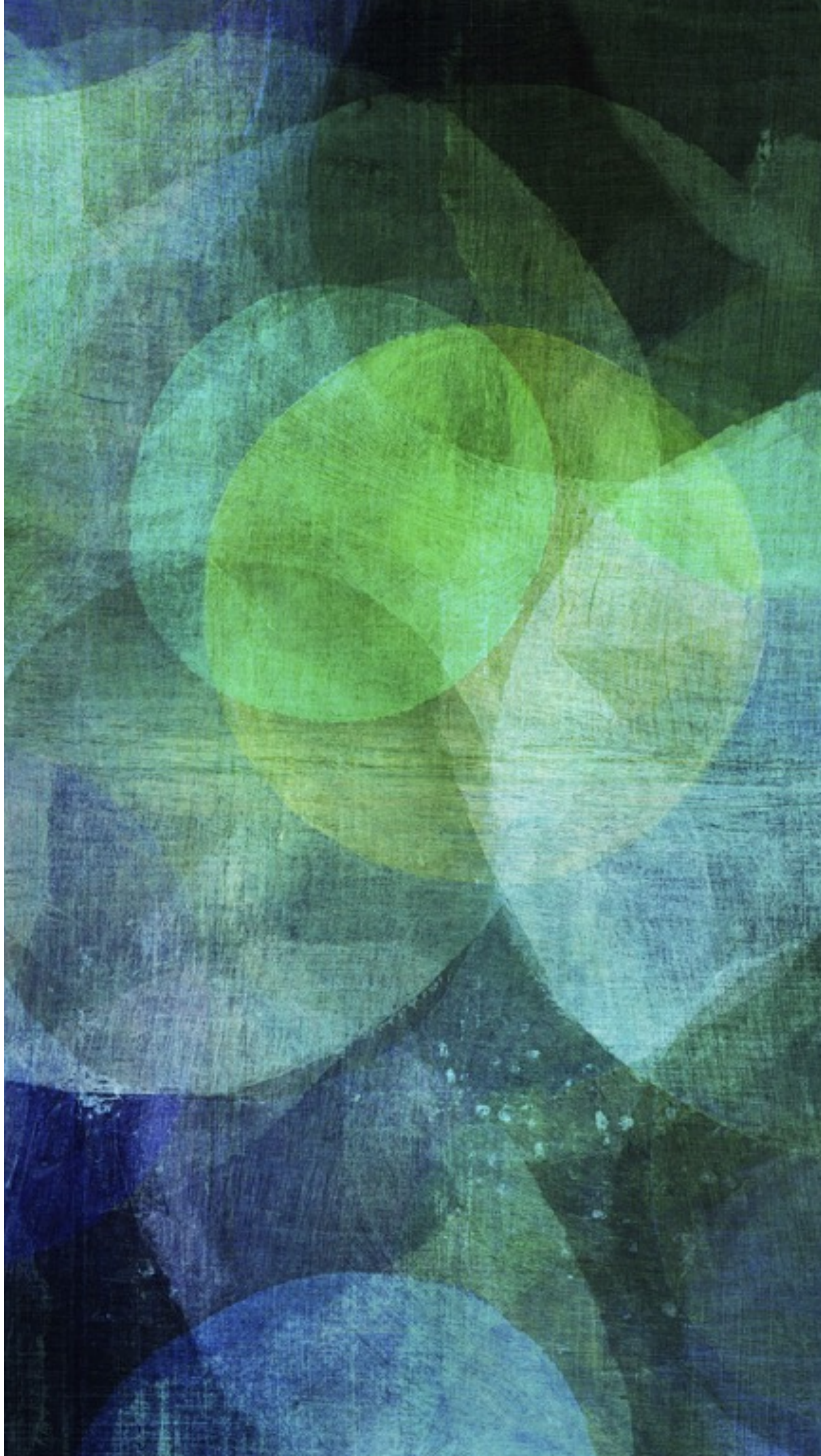
MV, Munos, Kveton, Kocák: **Spectral Bandits for Smooth Graph Functions**, ICML 2014

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



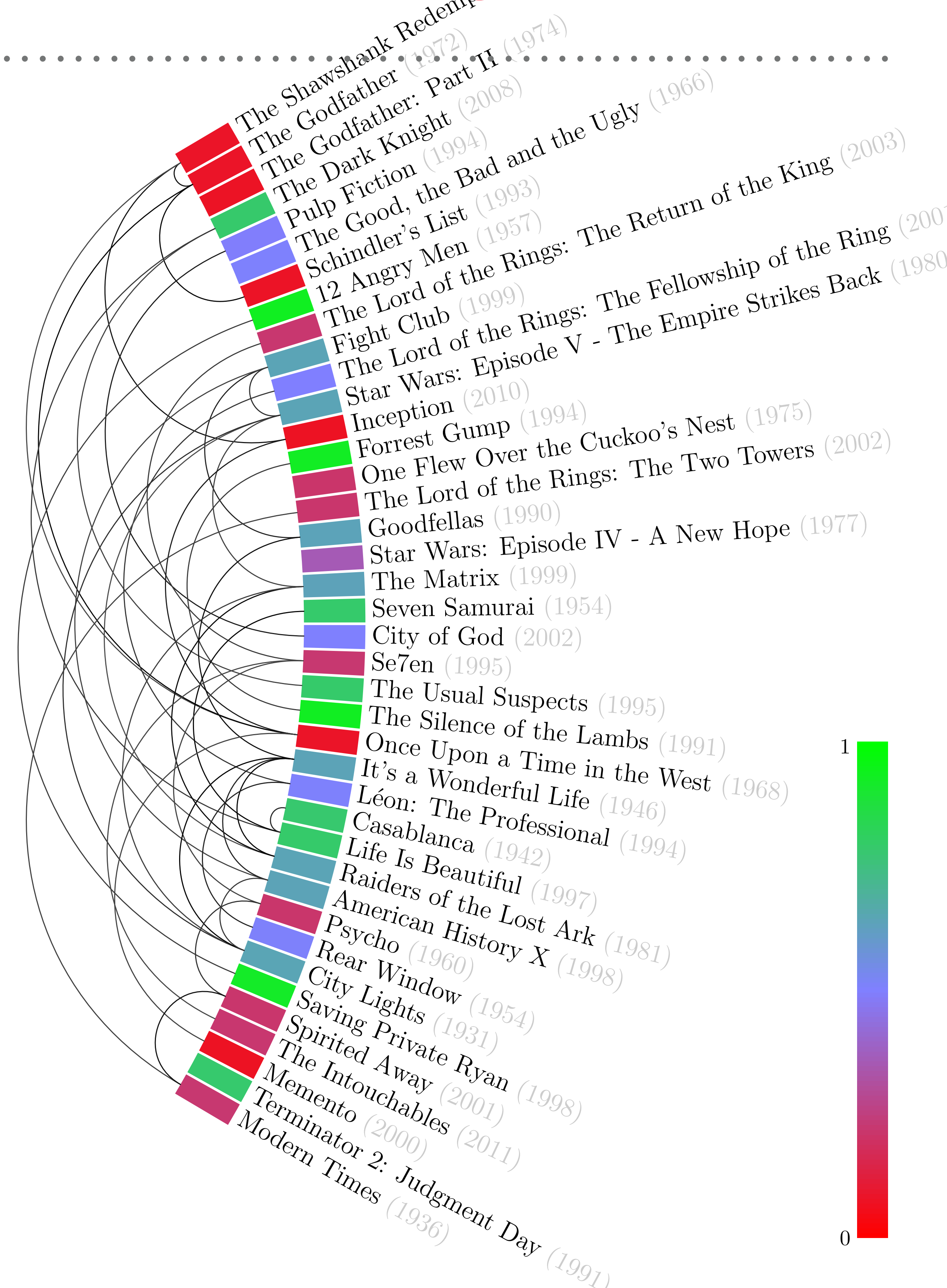
SPECTRAL BANDITS

Assumptions

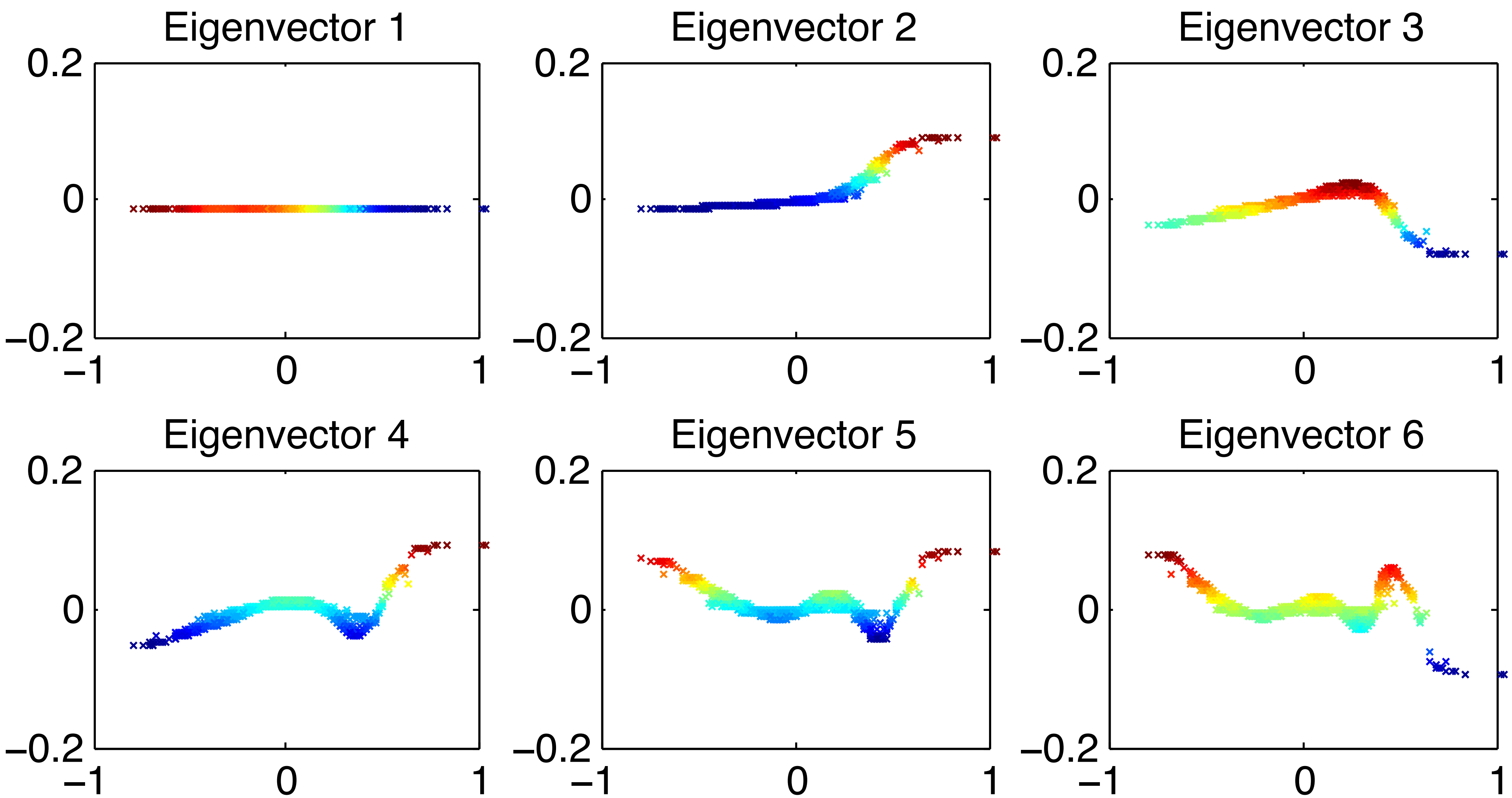
- Unknown reward function $f : V(G) \to \mathbb{R}$.
- Function f is **smooth** on a graph.
- Neighboring movies $\Rightarrow$ similar preferences.
- Similar preferences $\not\Rightarrow$ neighboring movies.

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.

SMOOTH GRAPH FUNCTIONS

- $\mathbf{f} = (f_1, \ldots, f_N)^\mathsf{T}$: Vector of function values.
- Let $\mathbf{L} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}$ be the eigendecomposition of the Laplacian.
  - Diagonal matrix $\boldsymbol{\Lambda}$ whose diagonal entries are eigenvalues of $\mathbf{L}$.
  - Columns of $\mathbf{Q}$ are eigenvectors of $\mathbf{L}$.
  - Columns of $\mathbf{Q}$ form a basis.
- $\boldsymbol{\alpha}^*$: Unique vector such that $\mathbf{Q}\boldsymbol{\alpha}^* = \mathbf{f}$ Note: $\mathbf{Q}^\mathsf{T}\mathbf{f} = \boldsymbol{\alpha}^*$

$\frac{1}{2}\sum_{i,j\leq n} w_{i,j}(f_i - f_j)^2$

$$S_G(\mathbf{f}) = \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} = \mathbf{f}^\mathsf{T}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}\mathbf{f} = \boldsymbol{\alpha}^{*\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{\alpha}^* = \|\boldsymbol{\alpha}^*\|_{\boldsymbol{\Lambda}}^2 = \sum_{i=1}^{N} \lambda_i (\alpha_i^*)^2$$

Smoothness and regularization: Small value of

(a) $S_G(\mathbf{f})$ **(b)** $\boldsymbol{\Lambda}$ norm of $\boldsymbol{\alpha}^*$ **(c)** α_i^* for large λ_i

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

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

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

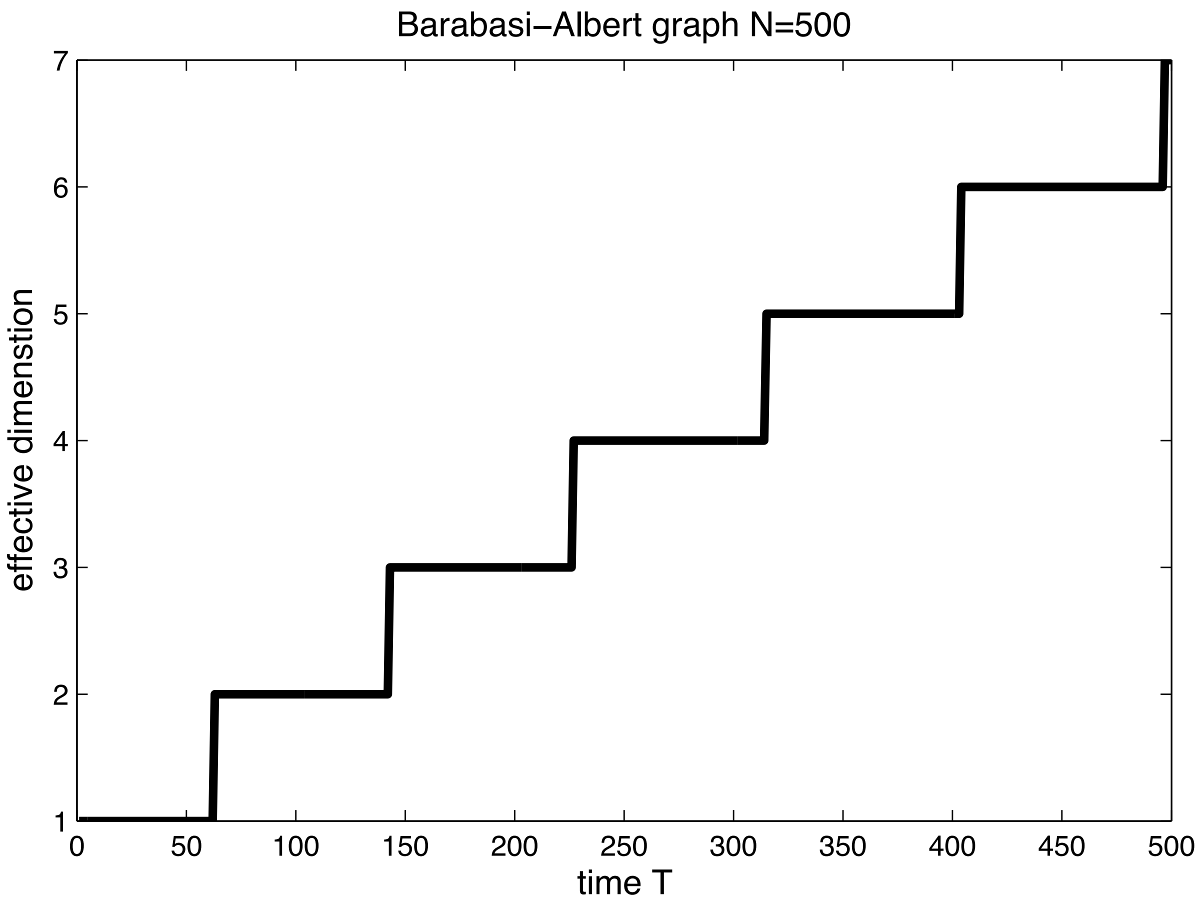
$$(d-1)\lambda_d \leq \frac{T}{\log(1+T/\lambda)}.$$

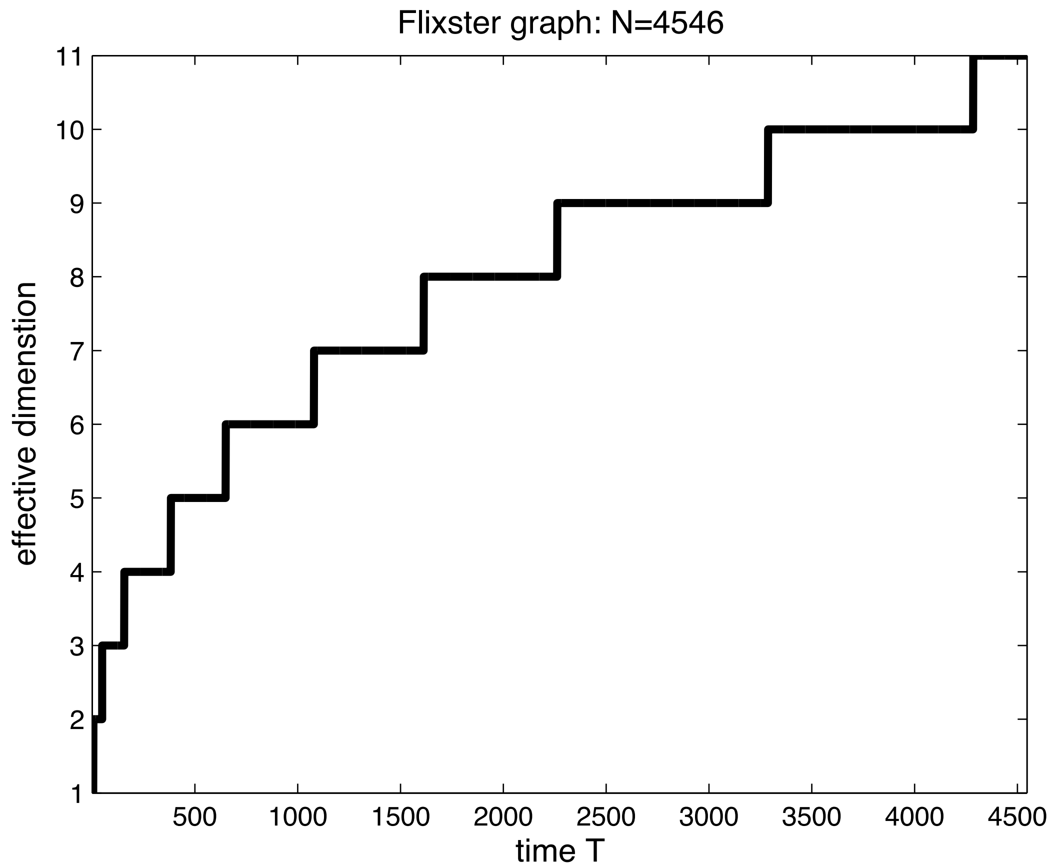
- Function of time horizon and graph properties
- λ_i : i-th smallest eigenvalue of $\mathbf{\Lambda}$.
- λ: Regularization parameter of the algorithm.

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 $T < N = D$.

Given a vector of weights $\boldsymbol{\alpha}$, we define its $\boldsymbol{\Lambda}$ norm as

$$\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} = \sqrt{\sum_{k=1}^{N} \lambda_k \alpha_k^2} = \sqrt{\boldsymbol{\alpha}^\mathsf{T} \boldsymbol{\Lambda} \boldsymbol{\alpha}},$$

and fit the ratings r_v with a (regularized) least-squares estimate

$$\widehat{\boldsymbol{\alpha}}_t = \arg\min_{\boldsymbol{\alpha}} \left(\sum_{v=1}^{t} \left[\langle \mathbf{x}_v, \boldsymbol{\alpha} \rangle - r_v \right]^2 + \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}^2 \right).$$

$\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}$ is a penalty for non-smooth combinations of eigenvectors.

SPECTRALUCB PSEUDOCODE

Input:
    $N,\ T,\ \{\boldsymbol{\Lambda}_\mathbf{L}, \mathbf{Q}\},\ \lambda,\ \delta,\ R,\ C$
Run:
    $\boldsymbol{\Lambda} \leftarrow \boldsymbol{\Lambda}_\mathbf{L} + \lambda \mathbf{I}$
    $d \leftarrow \max\{d : (d-1)\lambda_d \leq T/\ln(1+T/\lambda)\}$
for $t = 1$ **to** T **do**
    Update the basis coefficients $\widehat{\boldsymbol{\alpha}}$:
    $\mathbf{X}_t \leftarrow [\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(t-1)}]^\mathsf{T}$
    $\mathbf{r} \leftarrow [r_1, \ldots, r_{t-1}]^\mathsf{T}$
    $\mathbf{V}_t \leftarrow \mathbf{X}_t \mathbf{X}_t^\mathsf{T} + \boldsymbol{\Lambda}$
    $\widehat{\boldsymbol{\alpha}}_t \leftarrow \mathbf{V}_t^{-1} \mathbf{X}_t^\mathsf{T} \mathbf{r}$
    $c_t \leftarrow 2R\sqrt{d \ln(1+t/\lambda) + 2\ln(1/\delta)} + C$
    $\pi(t) \leftarrow \arg\max_a \left(\mathbf{x}_a^\mathsf{T} \widehat{\boldsymbol{\alpha}} + c_t \|\mathbf{x}_a\|_{\mathbf{V}_t^{-1}} \right)$
    Observe the reward r_t
end for

SPECTRALUCB REGRET BOUND

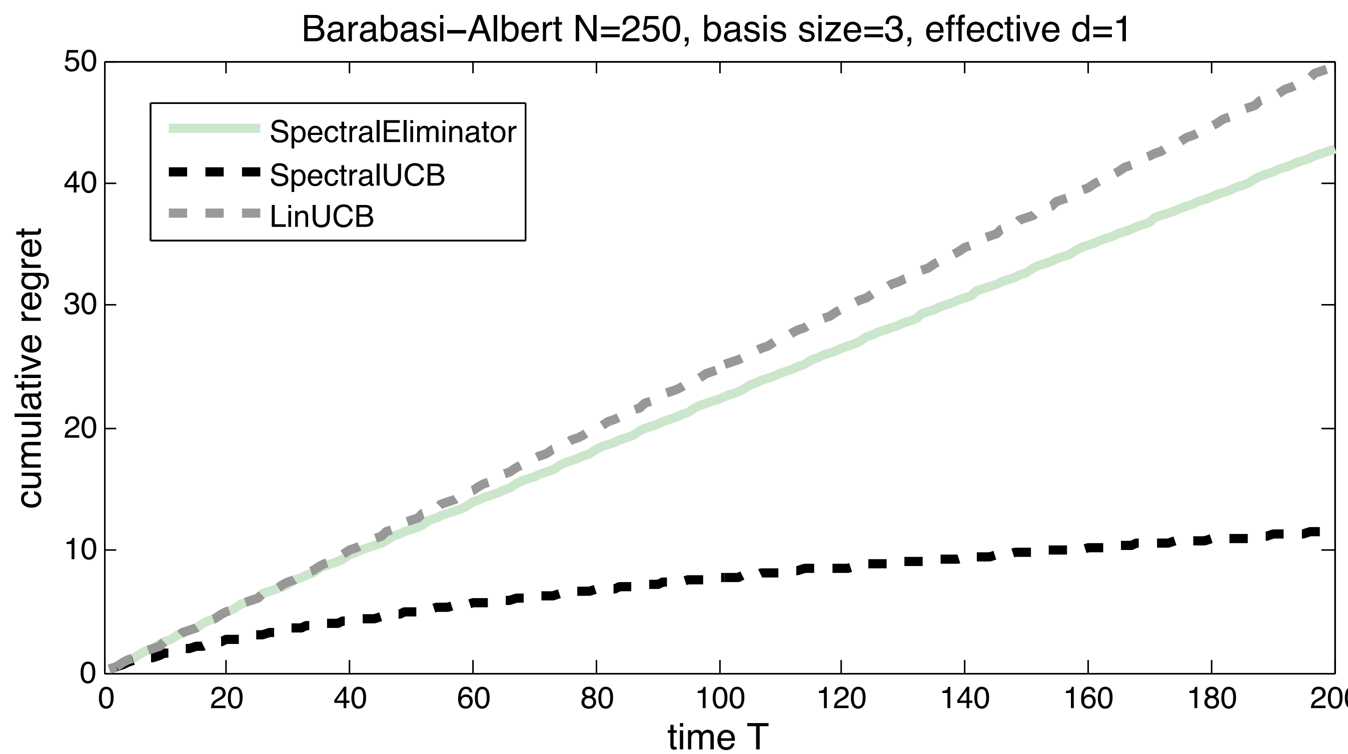
- d: Effective dimension.
- λ: Minimal eigenvalue of $\mathbf{\Lambda} = \mathbf{\Lambda_L} + \lambda \mathbf{I}$.
- C: Smoothness upper bound, $\|\boldsymbol{\alpha}^*\|_{\mathbf{\Lambda}} \leq C$.
- $\mathbf{x}_i^\mathsf{T} \boldsymbol{\alpha}^* \in [-1, 1]$ for all i.

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

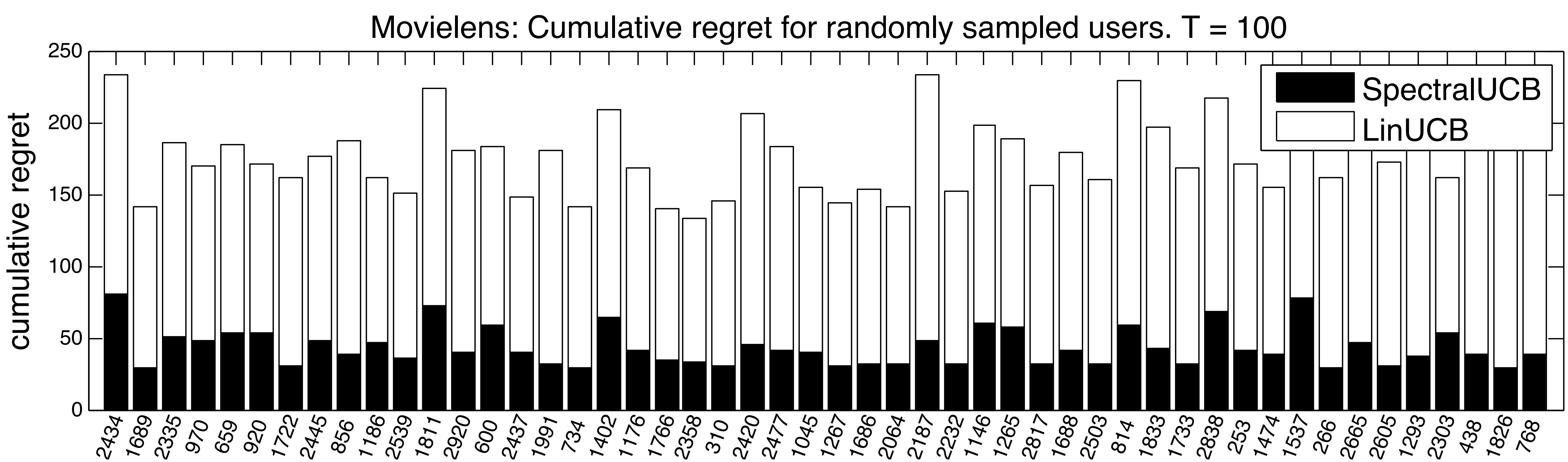
$$R_T \leq \left(8R\sqrt{d \ln \frac{\lambda + T}{\lambda} + 2 \ln \frac{1}{\delta}} + 4C + 4 \right) \sqrt{dT \ln \frac{\lambda + T}{\lambda}}.$$

$$R_T \approx d\sqrt{T \ln T}$$

EVALUATION



Barabasi–Albert N=250, basis size=3, effective d=1
SpectralEliminator
SpectralUCB
LinUCB
cumulative regret
time T
0 10 20 30 40 50
0 20 40 60 80 100 120 140 160 180 200

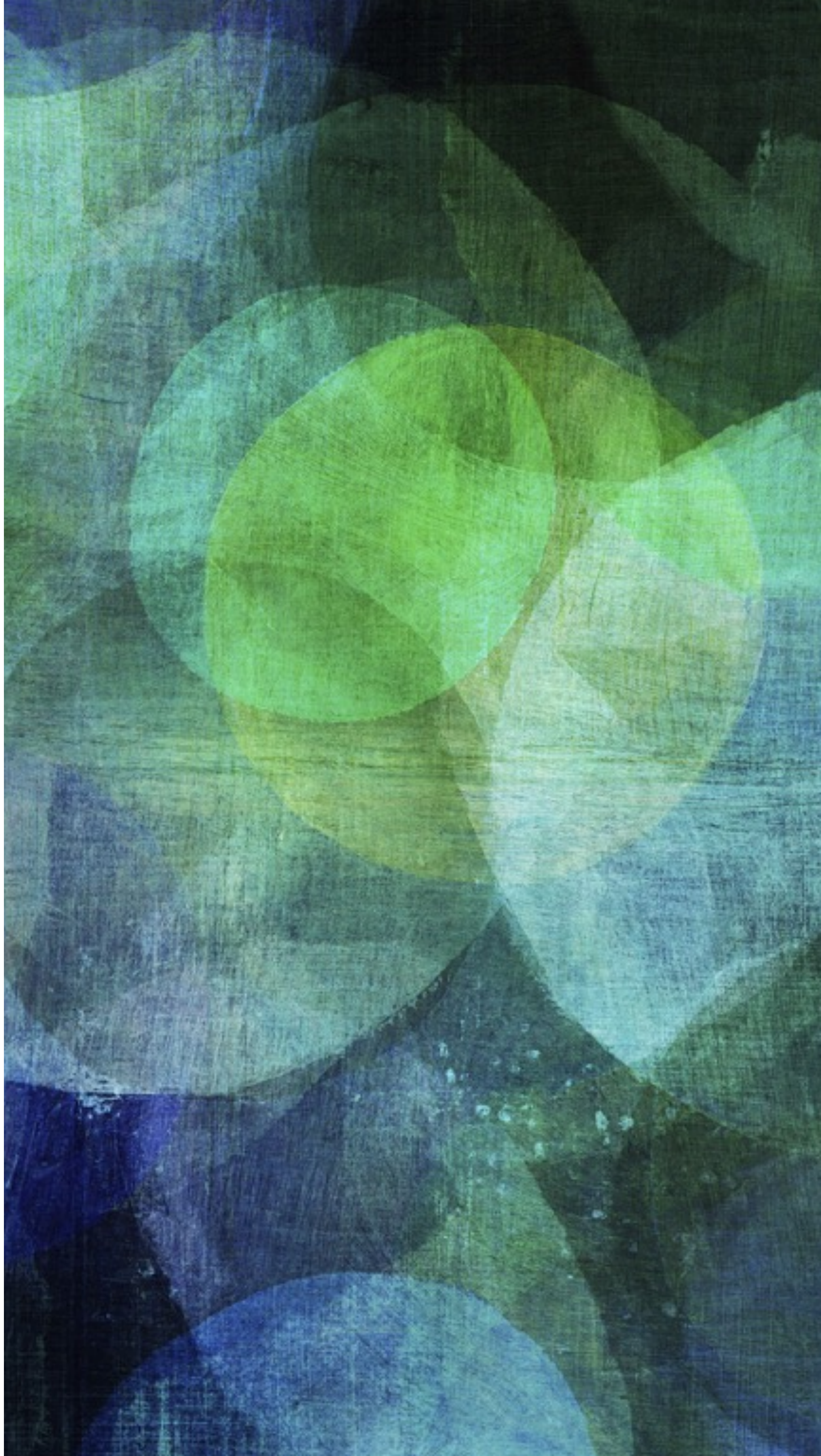


Movielens: Cumulative regret for randomly sampled users. T = 100
SpectralUCB
LinUCB
cumulative regret
0 50 100 150 200 250
2434 1689 2335 970 659 920 1722 2445 856 1186 2539 1811 2920 600 2437 1991 734 1402 1176 1766 2358 310 2420 2477 1045 1267 1686 2064 2187 2232 1146 1265 2817 1688 2503 814 1833 1733 2838 253 1474 1537 266 2665 2605 1293 2303 438 1826 768

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

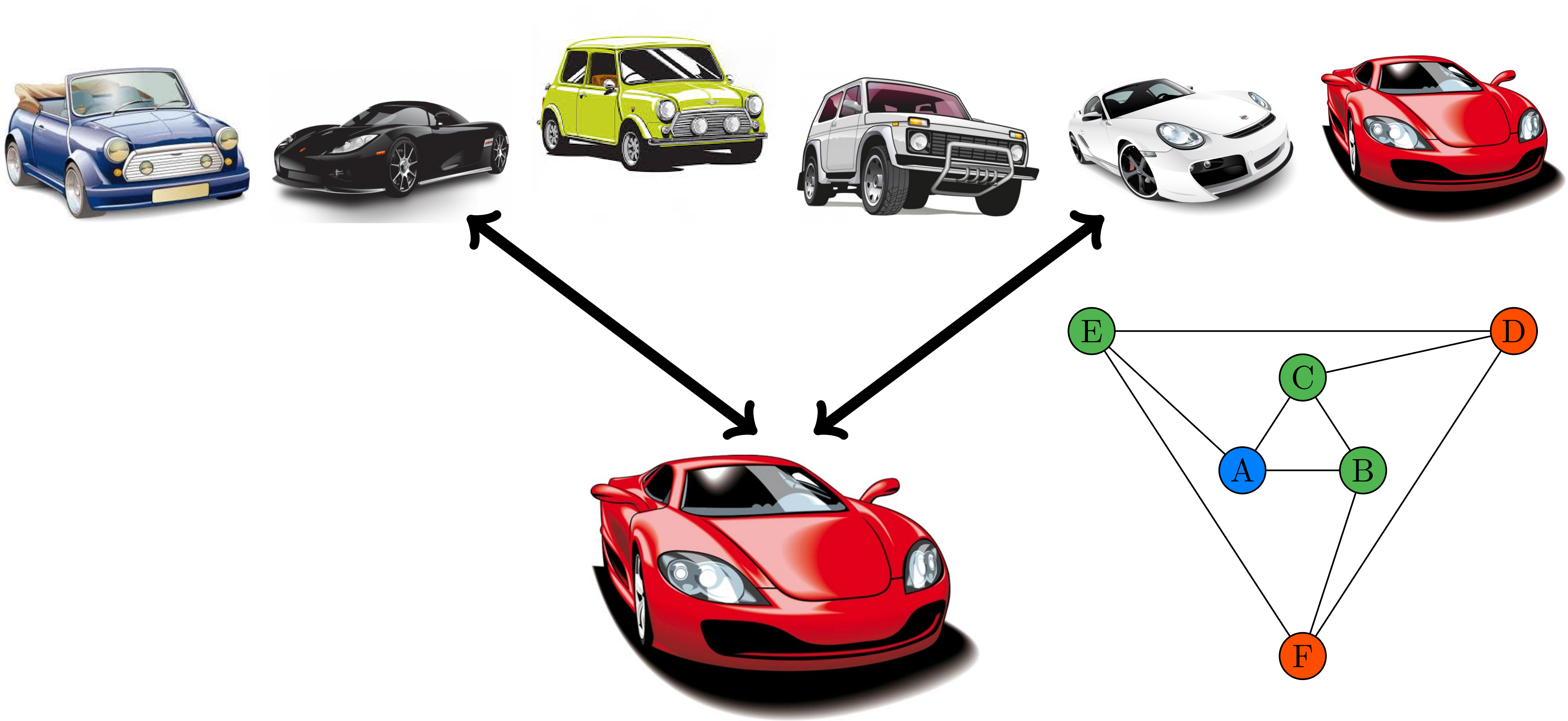
GRAPH BANDITS WITH SIDE OBSERVATIONS

exploiting **free** observations from neighbouring nodes

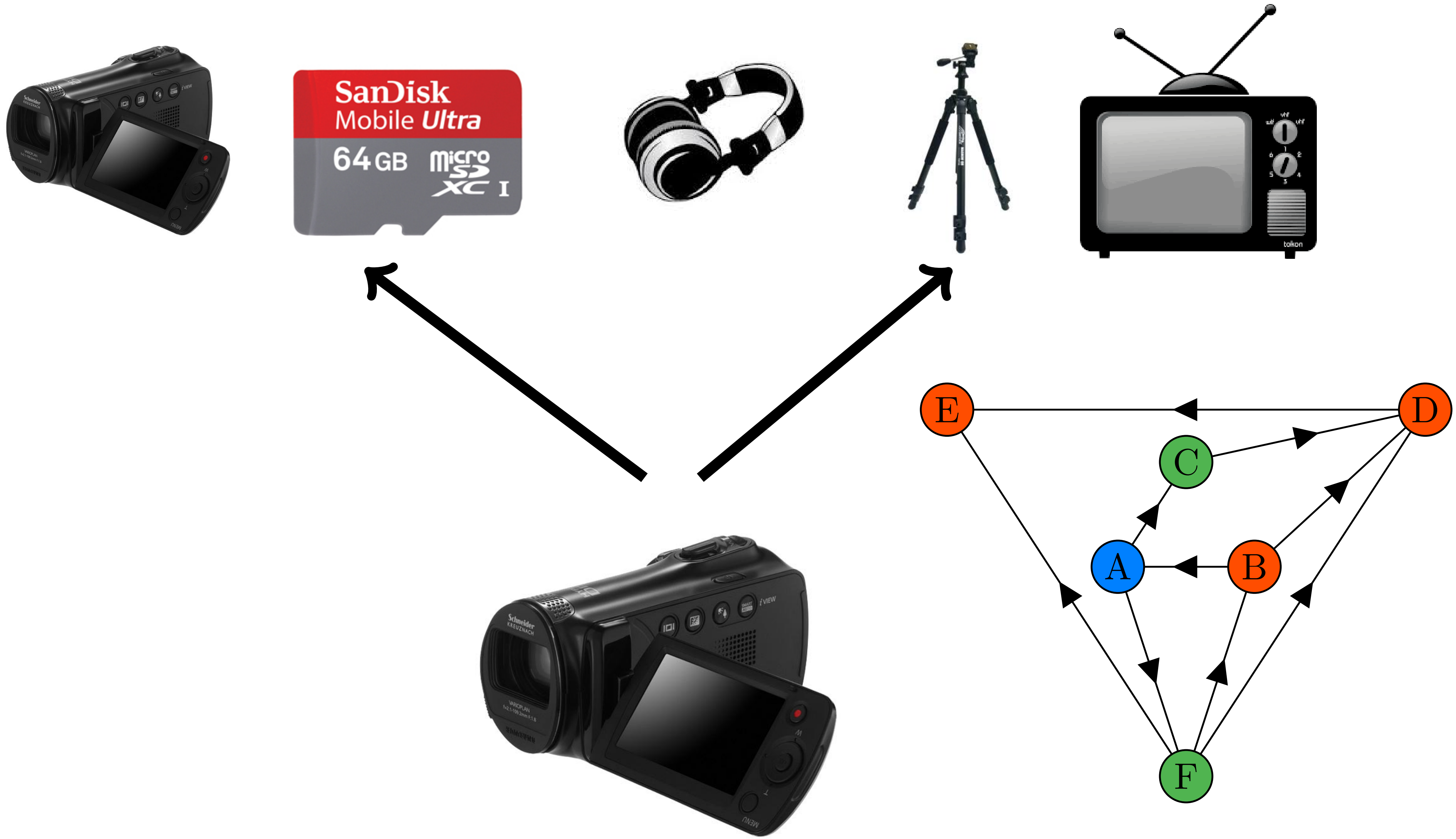


SIDE OBSERVATIONS: UNDIRECTED

Example 1: undirected observations



Example 2: Directed observation



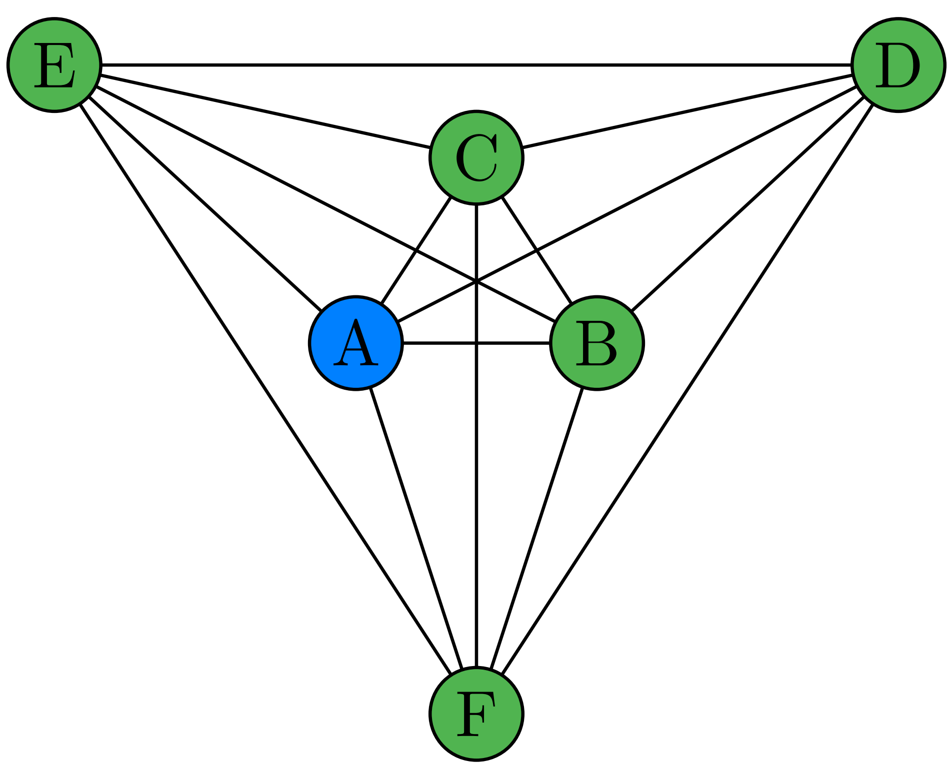
SIDE OBSERVATION: ADVERSARIAL SETTING

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

- **Environment (adversary):**
  - Privately assigns losses to actions
  - Generates an observation graph
    - Undirected / Directed
    - Disclosed / Not disclosed
- **Learner:**
  - Plays action $I_t \in [N]$
  - Obtain loss ℓ_{t,I_t} of action played
  - Observe losses of neighbors of I_t
    - Graph: disclosed

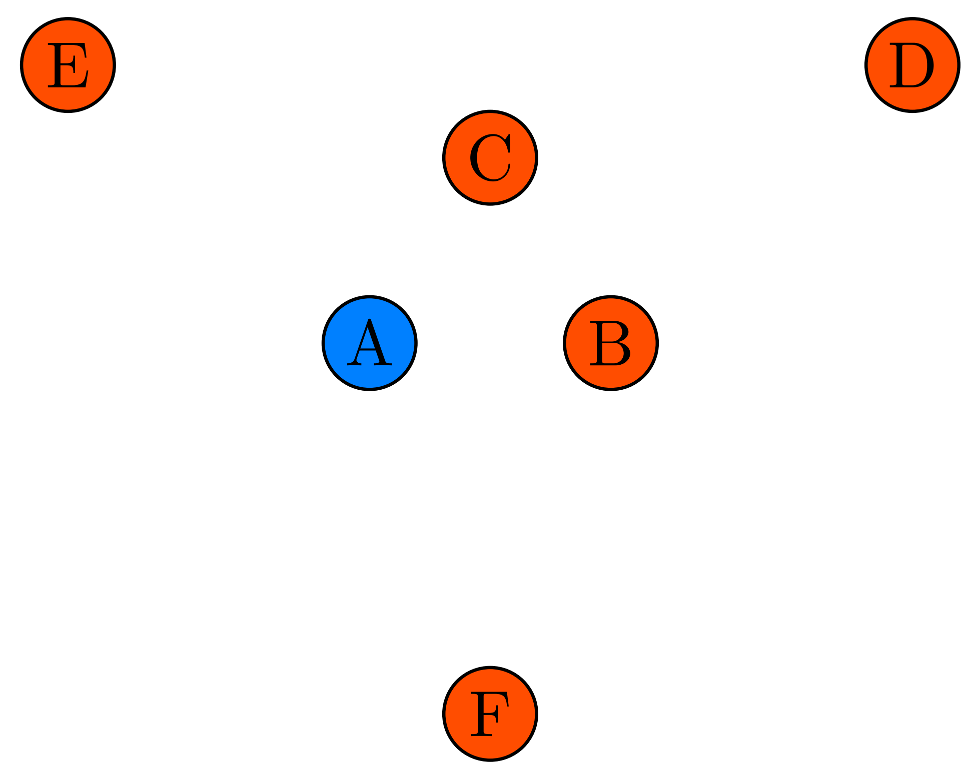
Full Information setting

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



Bandit setting

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



UNDIRECTED GRAPHS

Side observation (Undirected case)

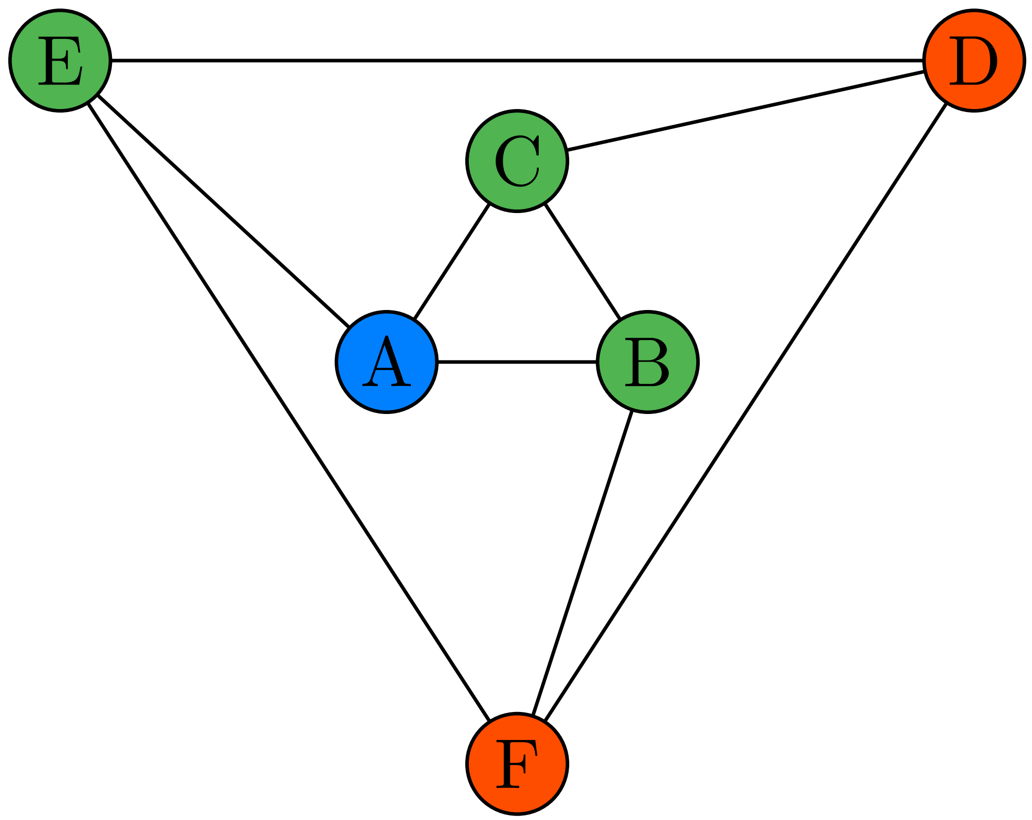
- Pick an action (e.g. action A)
- Observe losses of neighbors

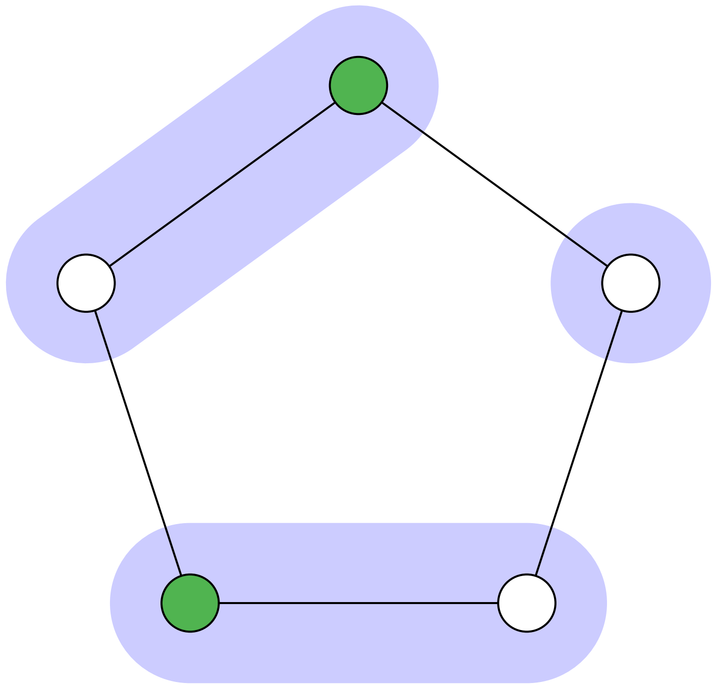
Mannor and Shamir (ELP algorithm)

- Need to know the graph
- Clique decomposition (c cliques)
- $R_T = \widetilde{\mathcal{O}}(\sqrt{cT})$

Alon, Cesa-Bianchi, Gentile, Mansour

- No need to know the graph
- Independence set of α actions
- $R_T = \widetilde{\mathcal{O}}(\sqrt{\alpha T})$





DIRECTED GRAPHS

Side observation (Directed case)

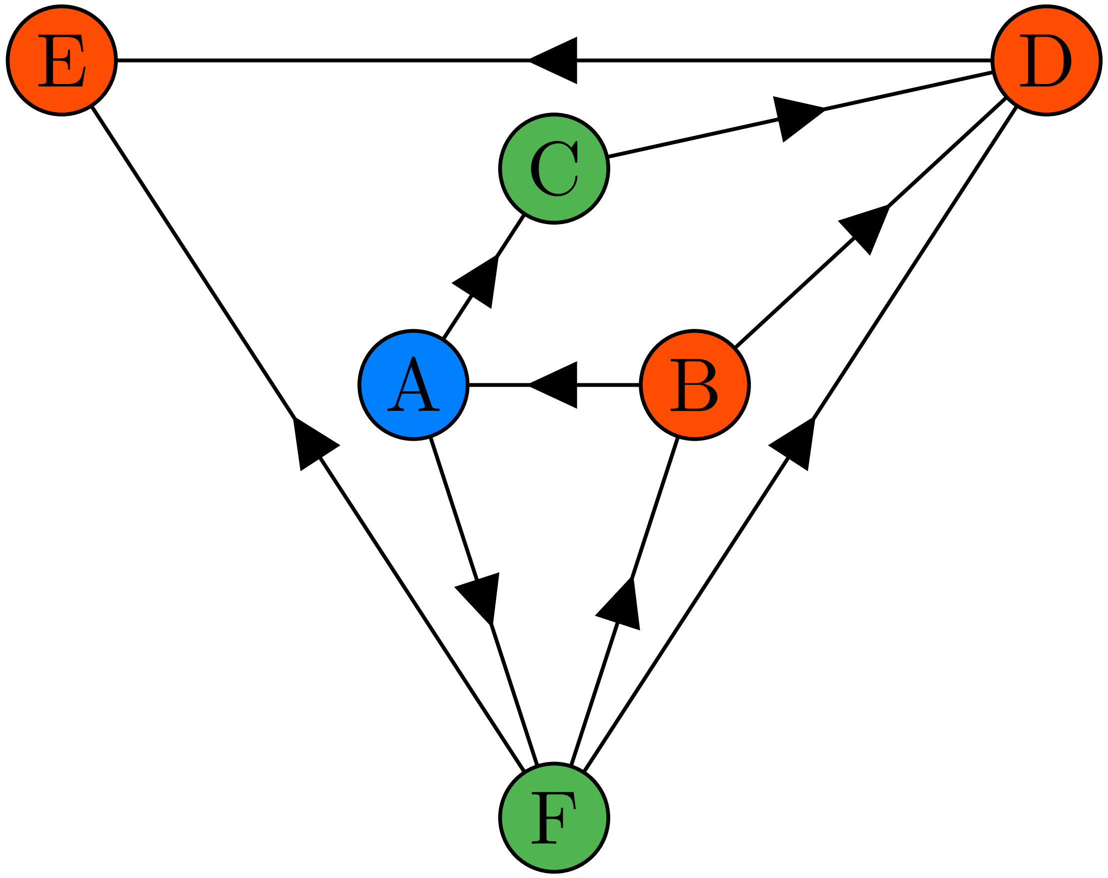
- Pick an action (e.g. action A)
- Observe losses of neighbors

Alon, Cesa-Bianchi, Gentile, Mansour

- **Exp3-DOM**
- Need to know graph
- Need to find dominating set
- $R_T = \widetilde{\mathcal{O}}(\sqrt{\alpha T})$

Exp3-IX - Kocák et. al

- No need to know graph
- $R_T = \widetilde{\mathcal{O}}(\sqrt{\alpha T})$



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**

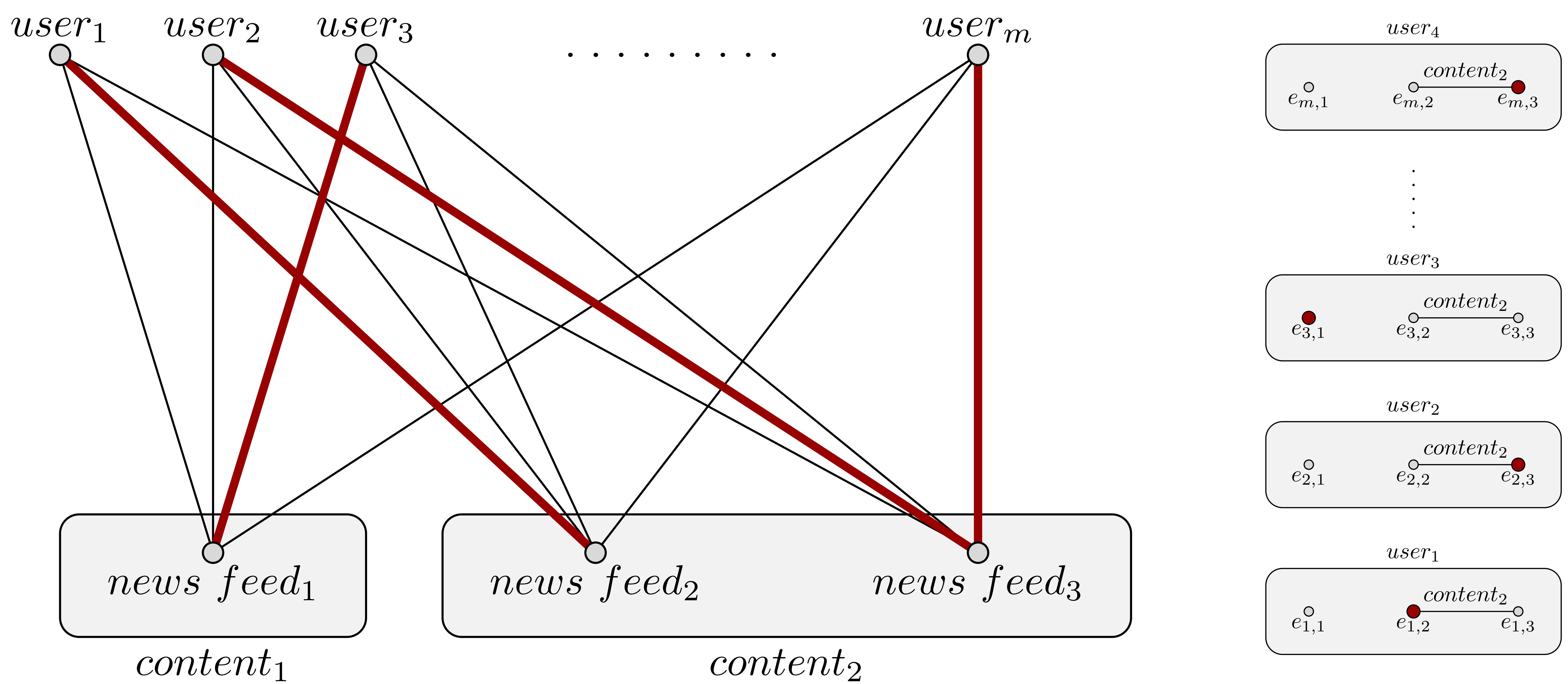
- **no need to know the graph before**
- no need to estimate dominating set
- no need for doubling trick
- no need for aggregation

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

Optimistic bias for the loss estimates

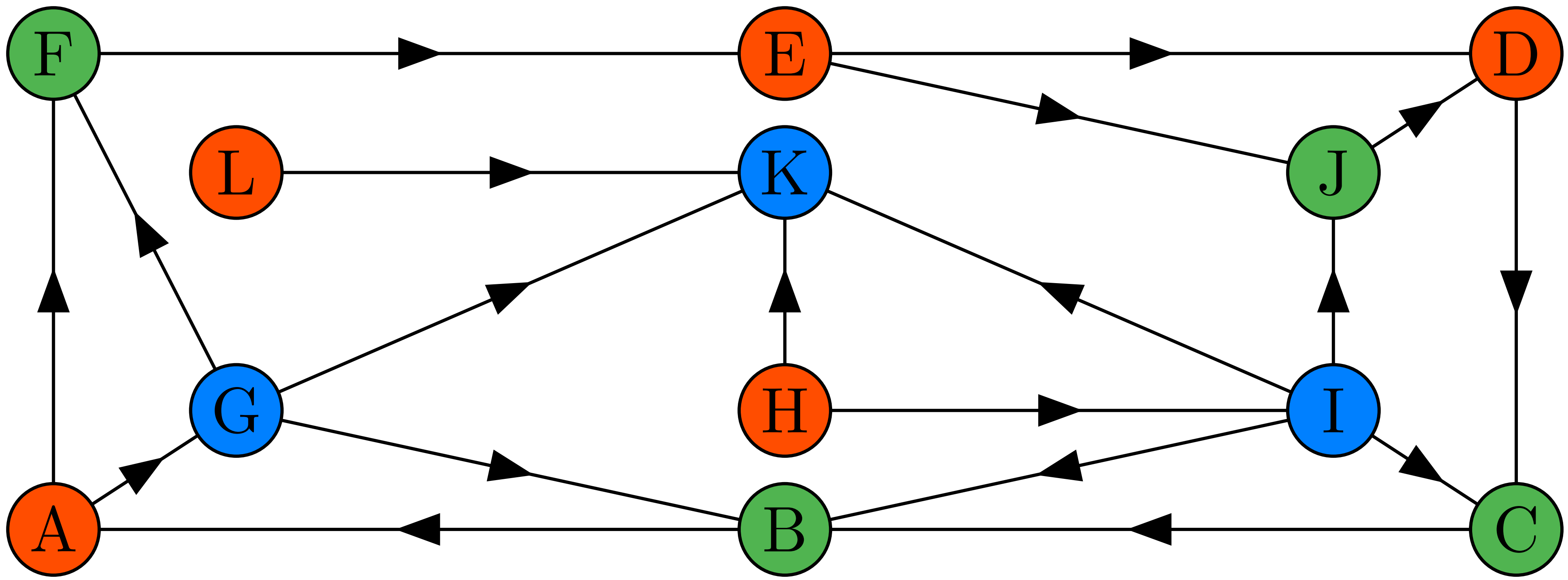
$$\mathbb{E}[\hat{\ell}_{t,i}] = \frac{\ell_{t,i}}{o_{t,i} + \gamma} o_{t,i} + 0(1 - o_{t,i}) = \ell_{t,i} - \ell_{t,i} \frac{\gamma}{o_{t,i} + \gamma} \leq \ell_{t,i}$$

COMPLEX ACTIONS: NEWS FEEDS



- Play m out of N nodes (combinatorial structure)
- Obtain losses of all played nodes
- Observe losses of all neighbors of played nodes

COMPLEX GRAPH ACTIONS



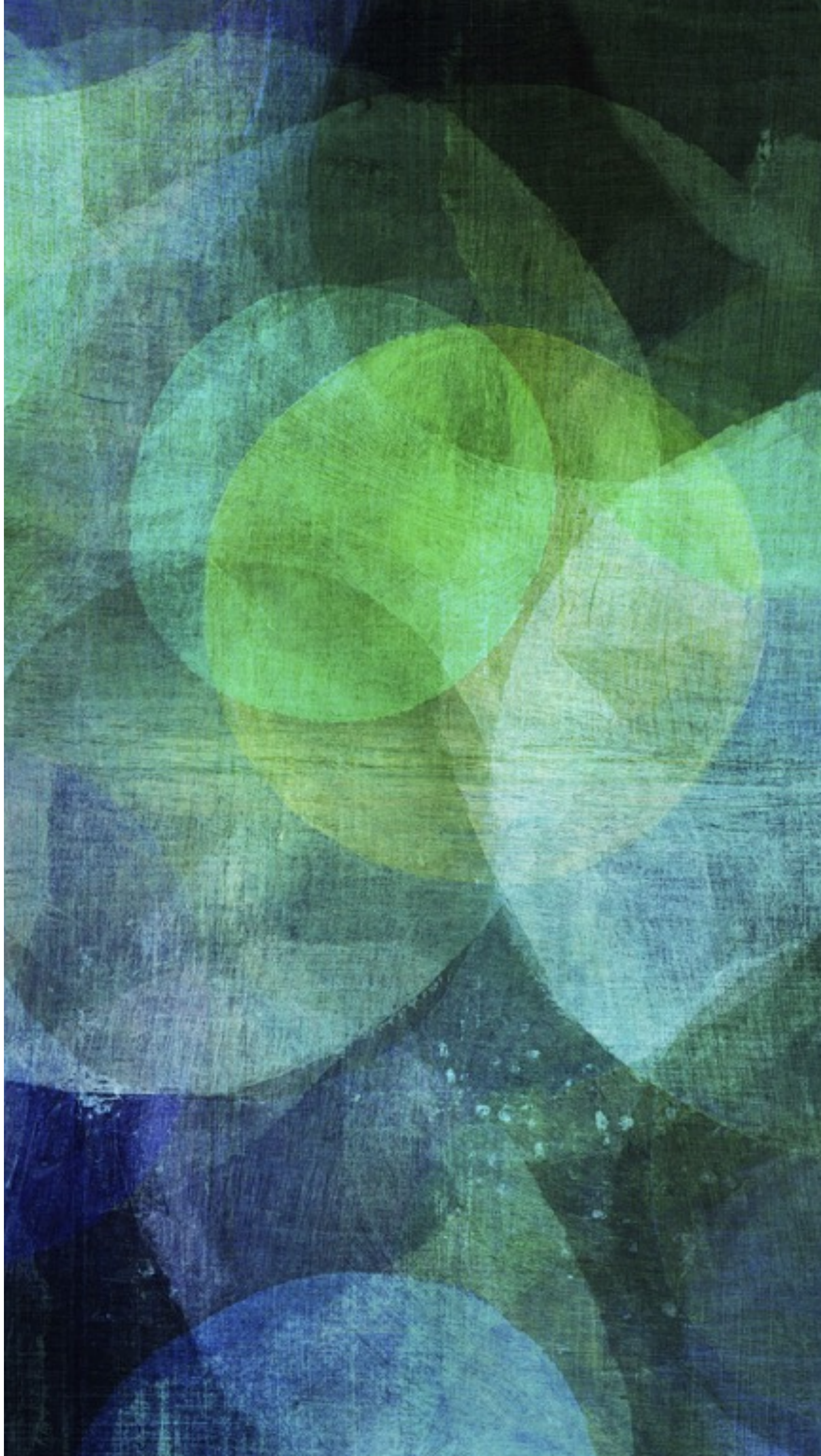
- Play action $\mathbf{V}_t \in S \subset \{0,1\}^N$, $\|\mathbf{v}\|_1 \leq m$ from all $\mathbf{v} \in S$
- Obtain losses $\mathbf{V}_t^\mathsf{T}\boldsymbol{\ell}_t$
- Observe additional losses according to the graph

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

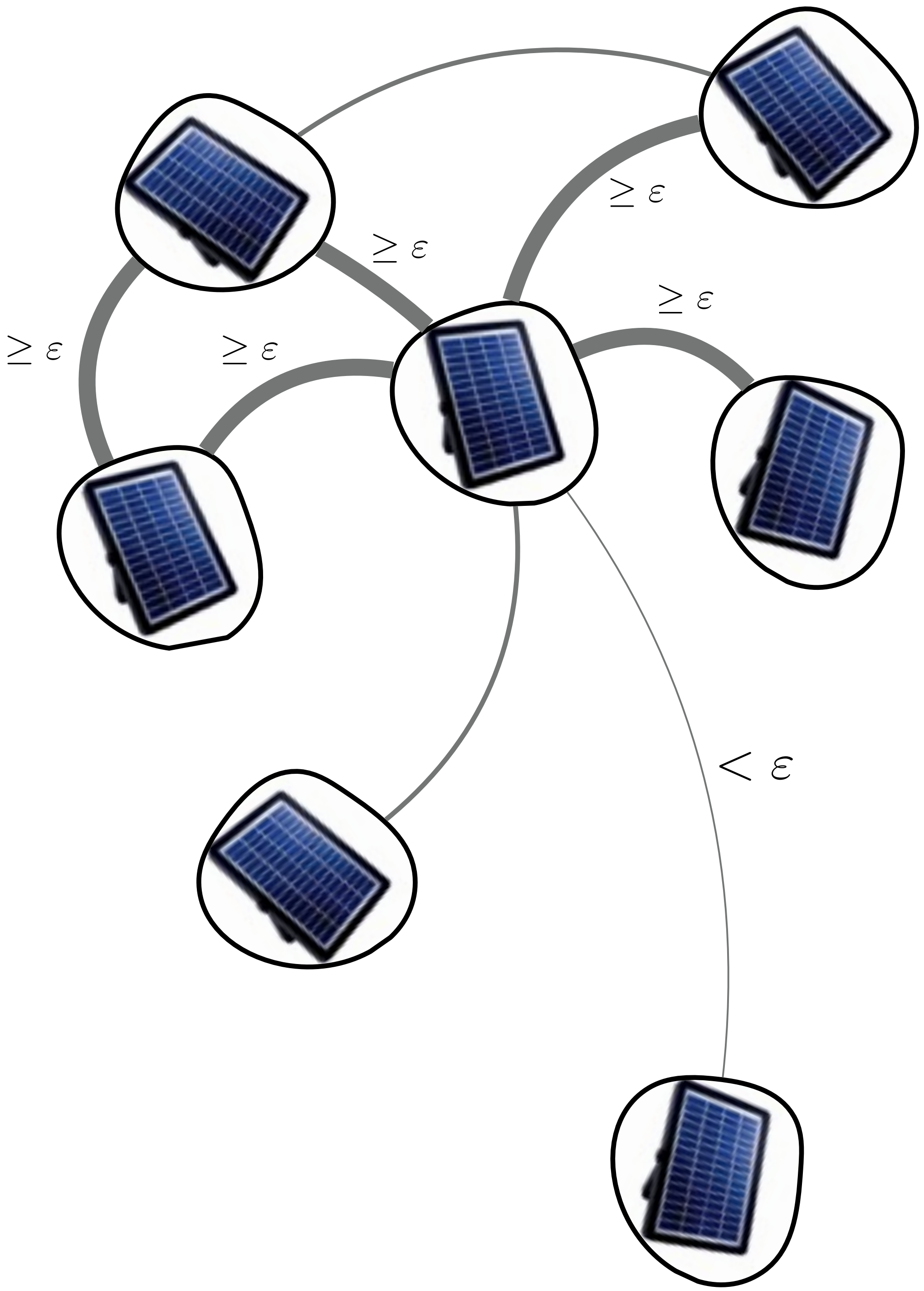
Kocák, Neu, MV: **Online learning with noisy side observations**, AISTATS 2016 (to appear)

GRAPH BANDITS WITH NOISY SIDE OBSERVATIONS

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



NOISY SIDE OBSERVATIONS



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?

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

effective independence number

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

PROTOCOL FOR NOISY OBSERVATIONS

Parameters:
set of arms $[N]$, number of rounds T.
For all $t = 1, 2, \dots, T$ **repeat**

1. The environment picks a loss function $\ell_t : [N] \to [0,1]$ and a directed weighted graph G_t with edge weights in $[0,1]$.
2. Based on its previous observations (and possibly some source of randomness), the learner picks an action $I_t \in [N]$.
3. The learner suffers loss ℓ_{t,I_t}.
4. The learner observes G_t and the feedback

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

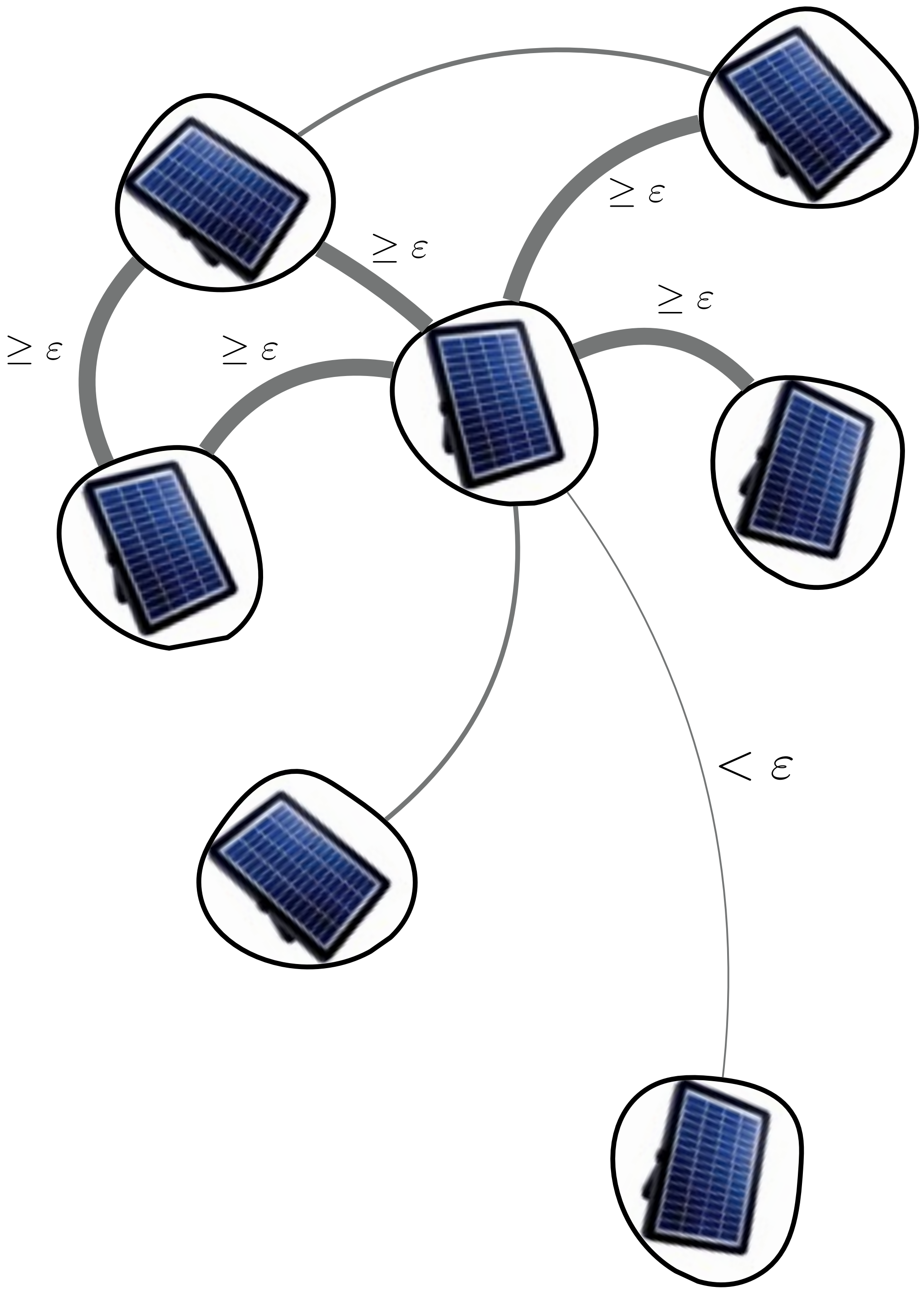
for every arm $i \in [N]$.

Nothing is revealed to the learner

The weights **s** are revealed

The noise is bounded $\xi \leq R$

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:**

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

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

incorporating noisy observations does not hurt

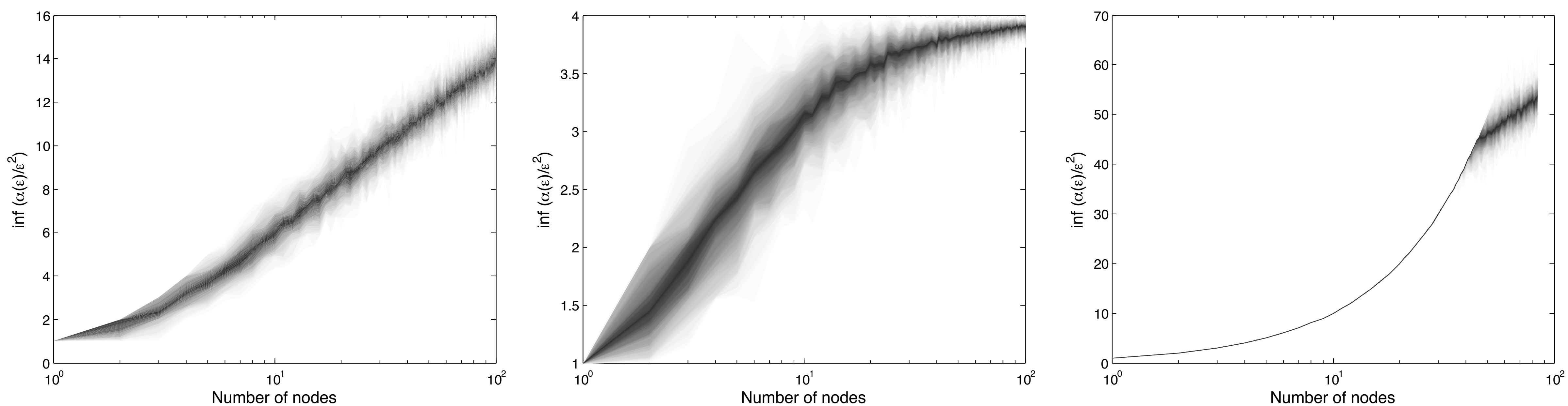
$$\tilde{\mathcal{O}}\left(\sqrt{\overline{\alpha}^\star T}\right) \leq \tilde{\mathcal{O}}\left(\sqrt{N T}\right)$$

But how much does it help?

EMPIRICAL α^* FOR SOME GRAPHS

- k x k grid graphs with weights $1/(1+d_{ij}^2)$ have α^* empirically bounded by a constant
- **special case:** if s_{ij} is either 0 or ε than $\alpha^* = \alpha/\varepsilon^2$
  - For this special case, there is a minimax regret $\Theta(\sqrt{(\alpha T)}/\varepsilon)$ by Wu, **György**, Szepesvári: **Online Learning with Gaussian Payoffs and Side Observations**, NIPS 2015.

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

ALGORITHM

Algorithm 1 Algorithm template: EXP3 (Auer et al., 2002a)

Initialization: $\widehat{L}_{0,i} = 0$ for all $i \in [N]$.
for $t = 1$ **to** T **do**
    Set η_t and γ_t.
    Construct the probability distribution $\boldsymbol{p}_t$ with.

$$p_{t,i} = \frac{\exp\big(-\eta_t \widehat{L}_{t-1,i}\big)}{\sum_{j=1}^N \exp\big(-\eta_t \widehat{L}_{t-1,j}\big)}.$$

    Play random arm I_t according to $\boldsymbol{p}_t$.
    Incur loss ℓ_{t,I_t}.
    Observe $c_{t,i} = s_{t,(I_t,i)}\ell_{t,i} + (1 - s_{t,(I_t,i)})\xi_{t,i}$ for all $i \in [N]$.
    Observe graph G_t.
    Construct loss estimates $\widehat{\ell}_{t,i}$.
    Set $\widehat{L}_{t,i} = \widehat{L}_{t-1,i} + \widehat{\ell}_{t,i}$.
end for

Naïve estimate $R_T = ?$

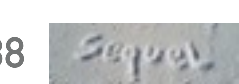
$$\widehat{\ell}_{t,i}^{(\mathrm{B})} = \frac{c_{t,i}}{\sum_{j=1}^N p_{t,j} s_{t,(j,i)} + \gamma_t}$$

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

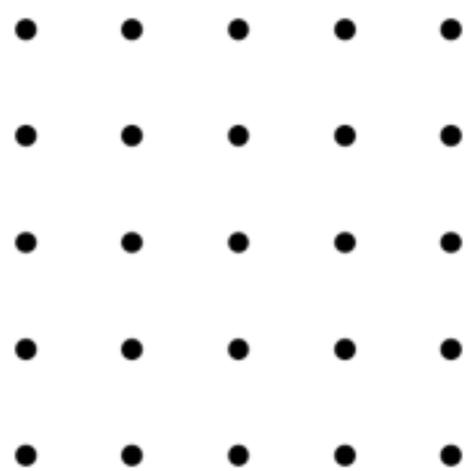
$$\widehat{\ell}_{t,i}^{(\mathrm{T})} = \frac{c_{t,i}\mathbb{I}_{\{s_{t,(I_t,i)} \geq \varepsilon_t\}}}{\sum_{j=1}^N p_{t,j} s_{t,(j,i)}\mathbb{I}_{\{s_{t,(j,i)} \geq \varepsilon_t\}} + \gamma_t}$$

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

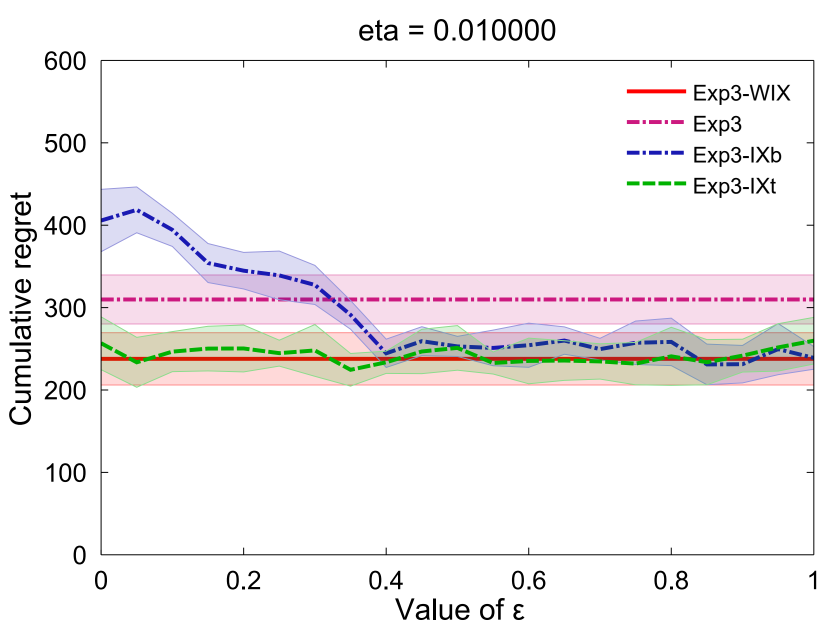
$$\widehat{\ell}_{t,i} = \frac{s_{t,(I_t,i)} \cdot c_{t,i}}{\sum_{j=1}^N p_{t,j} s_{t,(j,i)}^2 + \gamma_t}$$

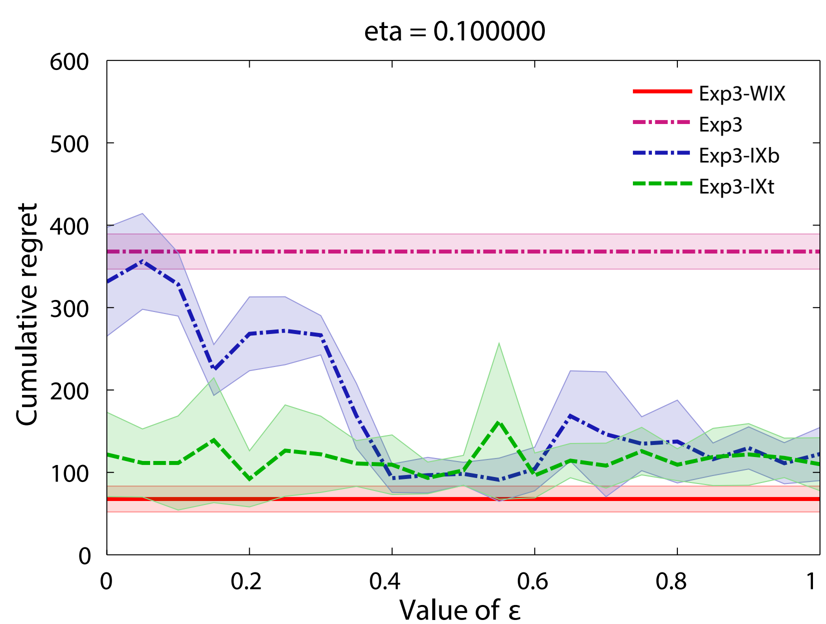


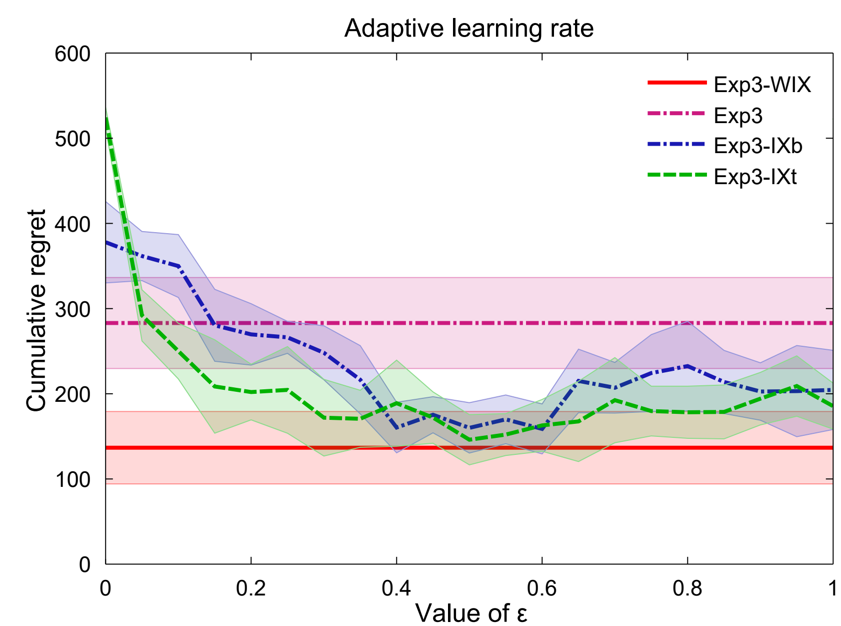
EMPIRICAL RESULTS FOR 5X5 GRID



- **nodes:** 25 actions on a 5x5 grid
- **weight:** $\min\{3/d^2,1\}$
  - d is the euclidean distance
- **loss:** alternating random walks



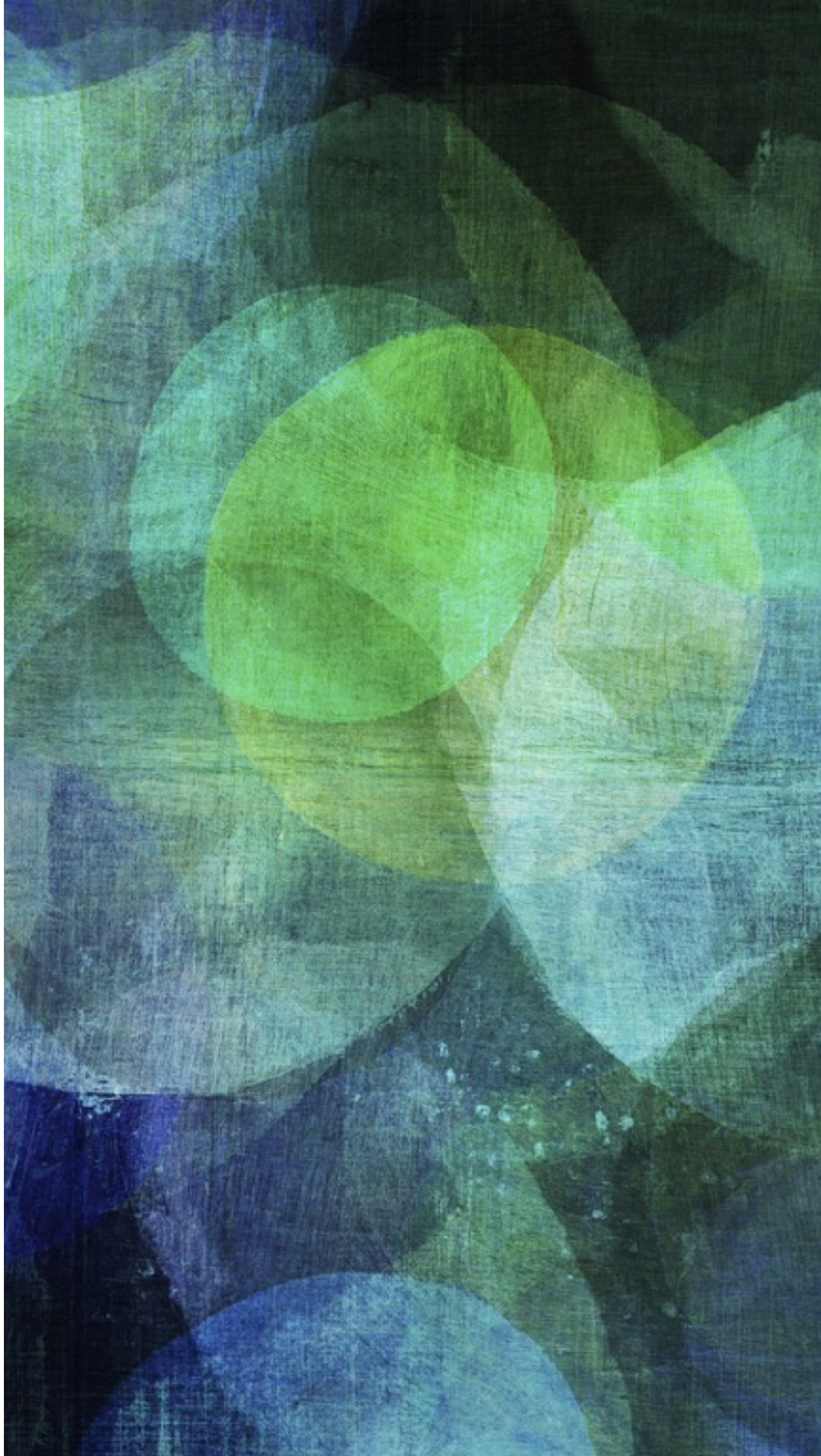




Carpentier, MV: **Revealing Graph Bandits for Maximising Local Influence**, AISTATS 2016 (to appear)

INFLUENCE MAXIMISATION

looking for the influential nodes while exploring the graph

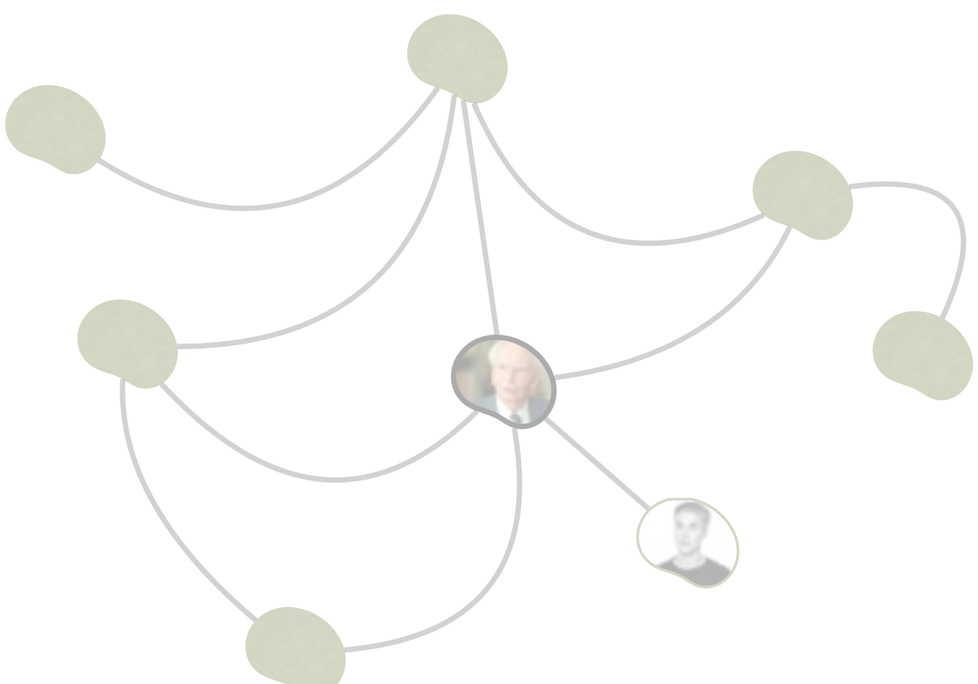


REVEALING BANDITS FOR LOCAL INFLUENCE

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

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

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

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

Ignoring the structure again? The best we can do is $\widetilde{\mathcal{O}}\left(\sqrt{r_* T N}\right)$

reward of the best node

We aim to do better: $R_T = \widetilde{\mathcal{O}}\left(\sqrt{r_* T D_*}\right)$

D_* - detectable dimension dependent 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

Idea of the algorithm:

- *exploration phase:* sample randomly to find out $\approx D_*$ nodes
- *bandit case:* use any bandit algorithm on these nodes

EMPIRICAL RESULTS

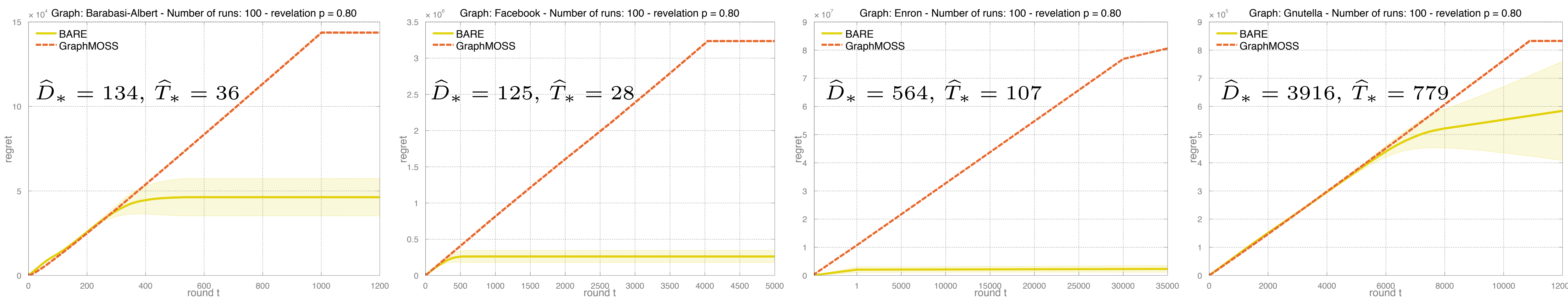


Figure 1: *Left*: Barabási-Albert. *Middle left*: Facebook. *Middle right*: Enron. *Right*: Gnutella.

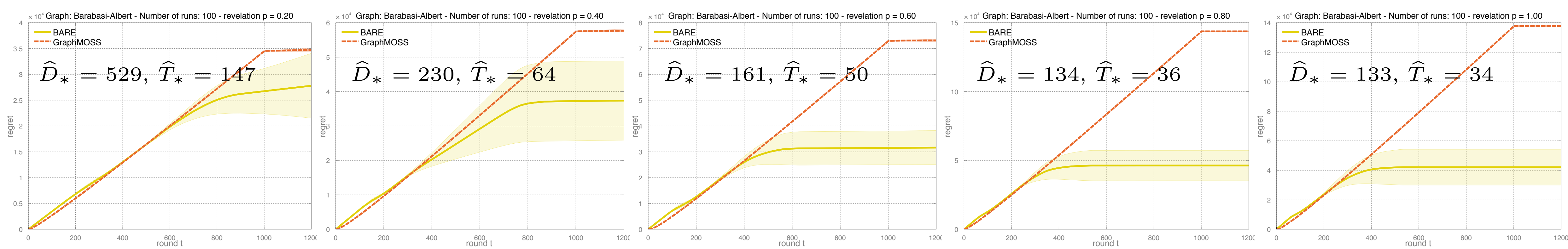


Figure 2: Barabási-Albert model with varying p between 0.2 and 1

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

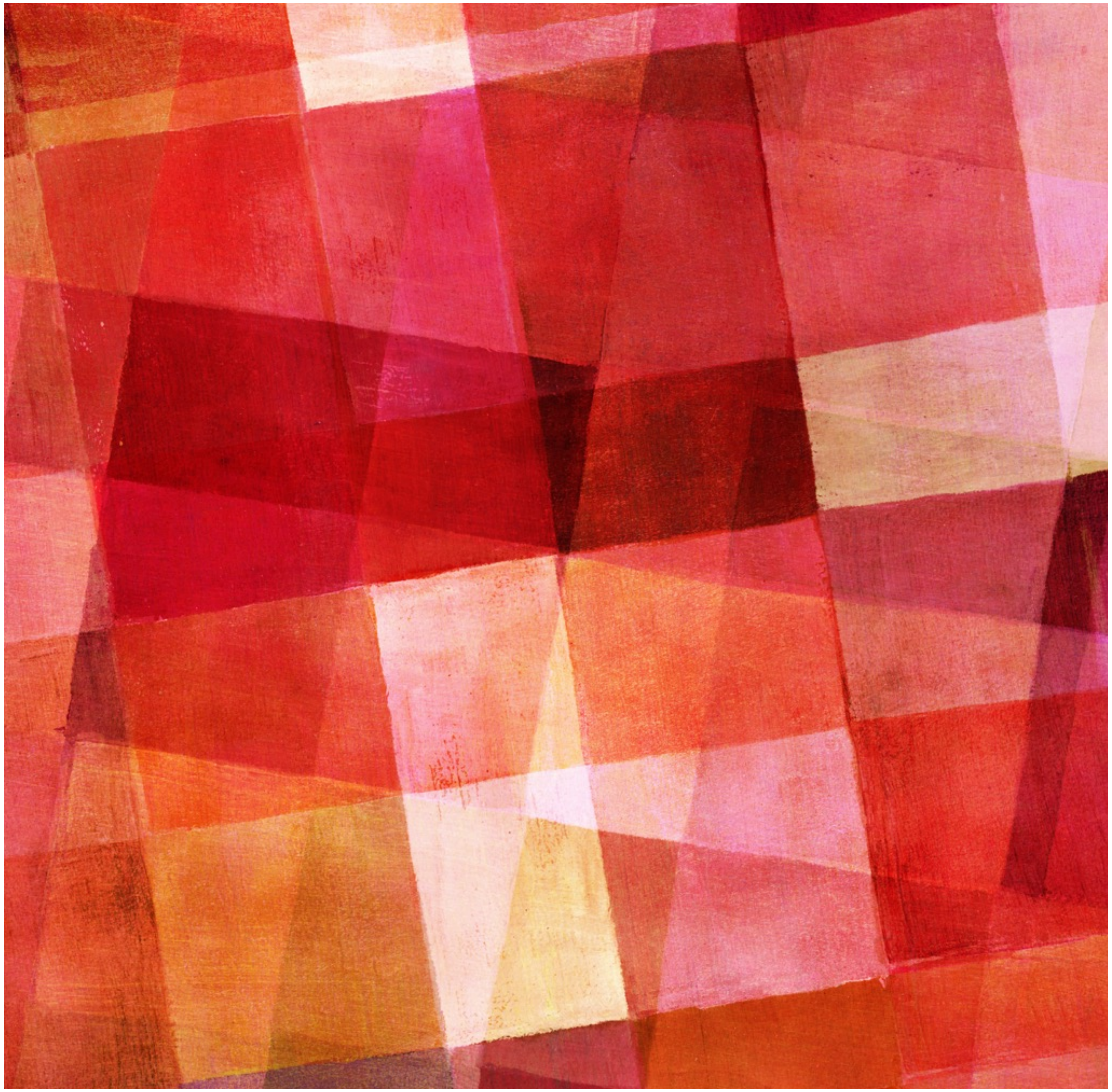
CONCLUSION AND NEW DIRECTIONS

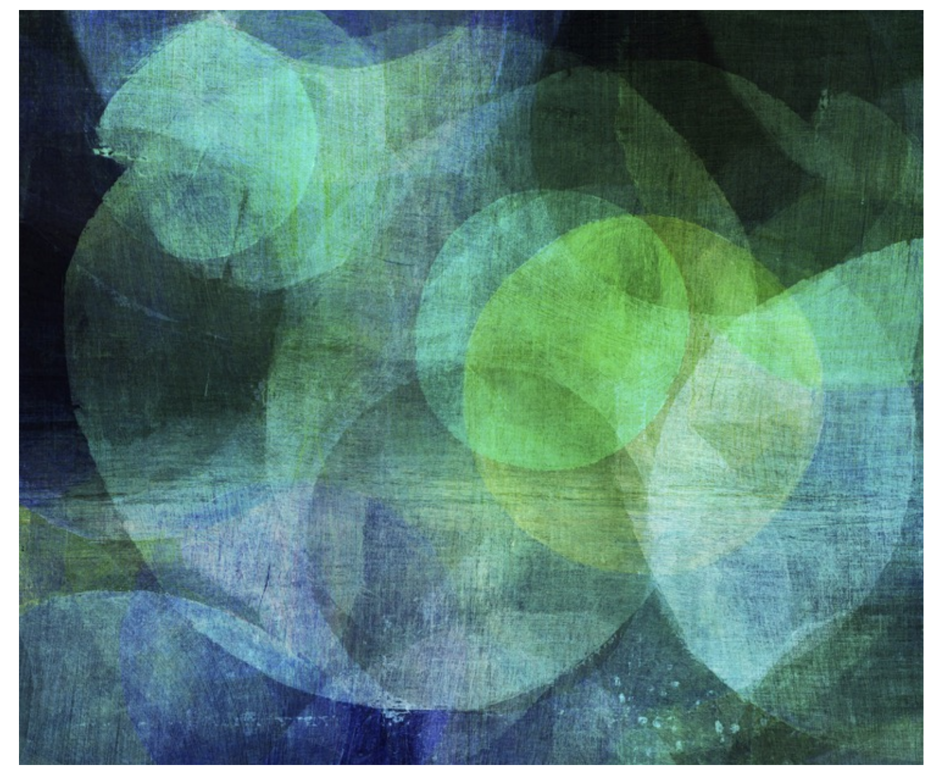
Graph Bandits

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

New directions

- graph generators (BA, ER, ...)
- learning (with) communities - SBM
- crawling strategies
- reducing assumption on graph knowledge





Multi-armed Bandit Workshop 2016 at STOR-i, Lancaster University, UK

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

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