

# Reducing Exposure to Harmful Content via Graph Rewiring

Corinna Coupette, Stefan Neumann, and Aristides Gionis



# Real-World Motivation

# Real-World Motivation

The image shows a YouTube video player interface. The main video is a Fox News broadcast featuring a man speaking. A large, semi-transparent overlay with the text "POLITICAL PERSECUTION" is positioned over the right side of the video. Below the video, a news ticker reads: "THE LEFT'S LATEST WITCH HUNT TRUMP INDICTED IN BRAGG'S POLITICALLY-CHARGED PROBE" and "BREAKING NEWS". A small inset for "JOE TACOPINA COMING UP" is visible in the bottom right corner of the video player. To the right of the video player is a list of recommended videos with thumbnails and titles, including "Tucker Carlson Tonight 4/2/23 FULL | BREAKING FOR NEWS...", "China Is Killing Americans" - Reaction To Xi Jinping (Infytv...)", "America Has 'Crossed the Rubicon': Mark Levin", "In less than 24 Hours the US dollar changes FOREVER...", "Trump Indictment: It is now OPEN SEASON on any and all...", and "Inside Olympic US \$4 Billion Submarine Probing the...". The YouTube search bar and "Sign In" button are visible at the top of the page.

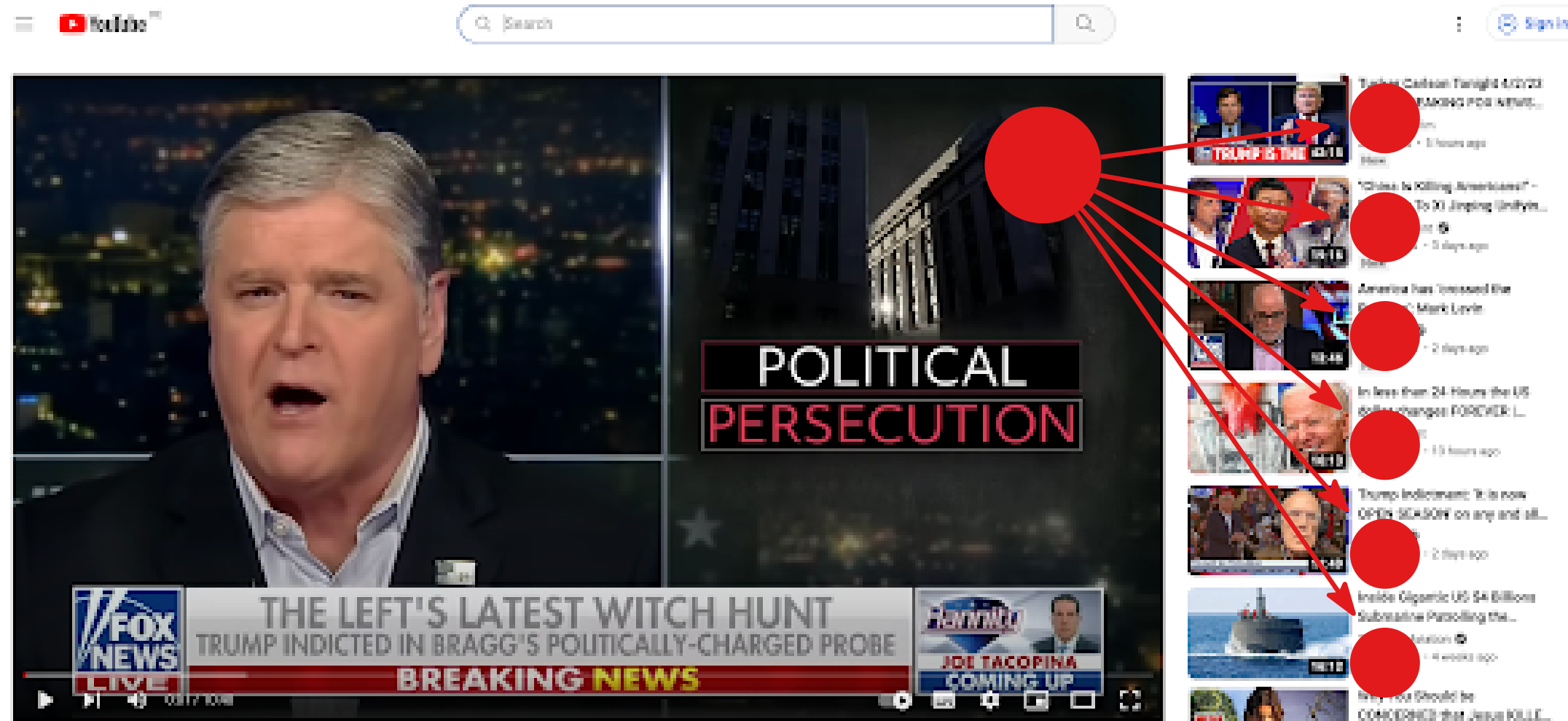
# Real-World Motivation



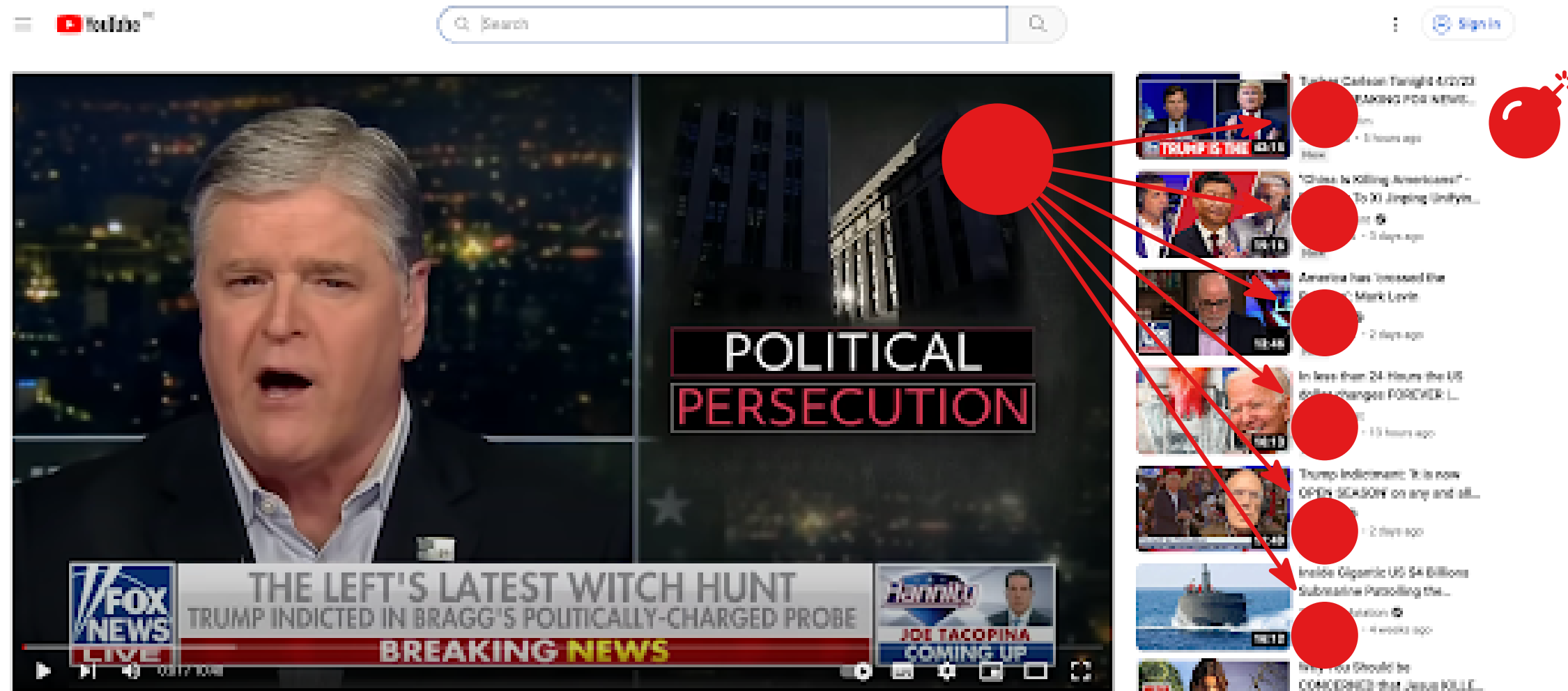
# Real-World Motivation



# Real-World Motivation



# Real-World Motivation



# Real-World Motivation

The image shows a YouTube interface with a video player and a sidebar. The video player displays a Fox News broadcast with a large red and white text overlay that reads "POLITICAL PERSECUTION". Below the video, a news ticker reads "THE LEFT'S LATEST WITCH HUNT TRUMP INDICTED IN BRAGG'S POLITICALLY-CHARGED PROBE" and "BREAKING NEWS". The sidebar on the right contains several video thumbnails with titles such as "Trump's Cabinet Tonight 4/21/23", "China is Killing Americans", and "America has Unleashed the". A large red circle is positioned over the video player, with red arrows pointing from it to several of the video thumbnails in the sidebar. A red bomb icon is also present in the top right corner of the sidebar area.



# Real-World Motivation

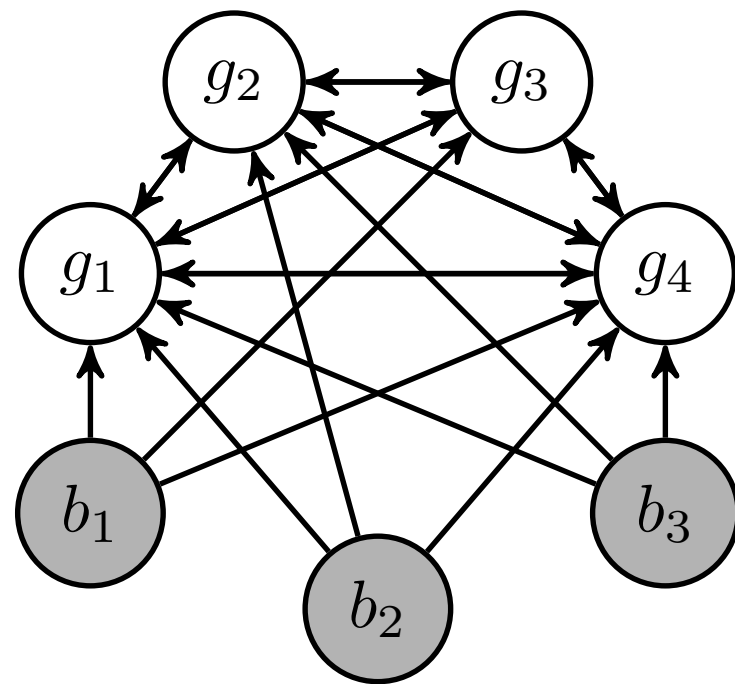
The image shows a YouTube interface with a video player on the left and a list of recommended videos on the right. The video player displays a Fox News broadcast with a large red and white text overlay that reads "POLITICAL PERSECUTION". Below the video, a news ticker reads "THE LEFT'S LATEST WITCH HUNT TRUMP INDICTED IN BRAGG'S POLITICALLY-CHARGED PROBE" and "BREAKING NEWS". A red circle is drawn around the video player, with several red arrows pointing from it to a heart icon in the top right corner of the YouTube interface. Below the heart icon is a list of recommended videos, each with a red circle next to its thumbnail. The recommended videos include titles such as "Trump's Cabinet Tonight 4/21/23", "China is Killing Americans", "America has Unleashed the", "In less than 24 Hours the US", "Trump Indictment: It is now", and "Inside Olympic US \$4 Billion".

# Real-World Motivation

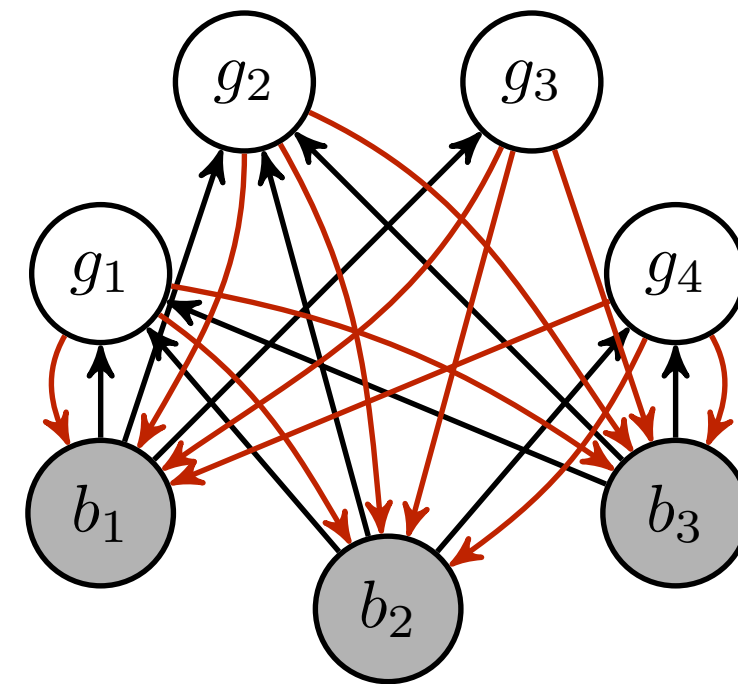
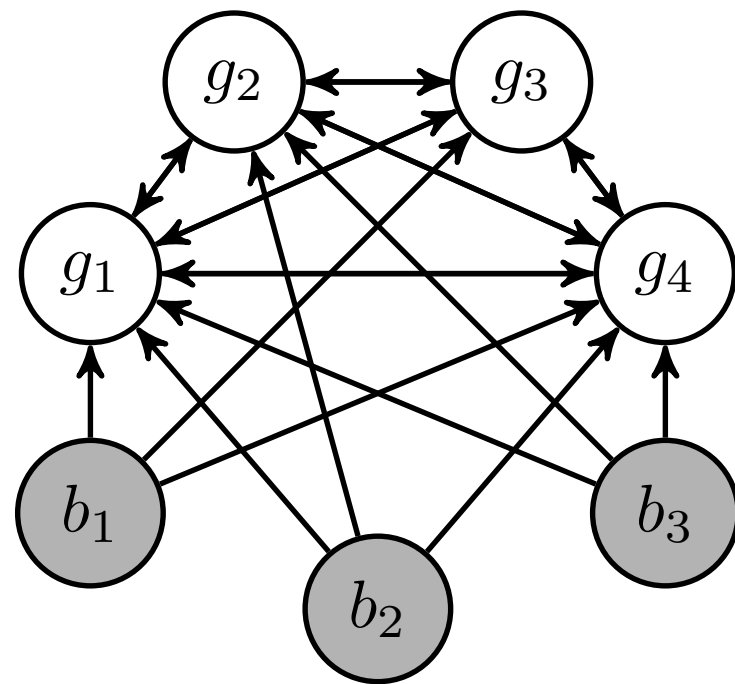
The image shows a YouTube interface. The main video player displays a Fox News broadcast with a large red and white text overlay that reads "POLITICAL PERSECUTION". Below the video, a news ticker reads "THE LEFT'S LATEST WITCH HUNT TRUMP INDICTED IN BRAGG'S POLITICALLY-CHARGED PROBE" and "BREAKING NEWS". To the right of the video player is a list of recommended videos. A red circle highlights one of these videos, and several red arrows point from this circle to the video player and to a heart icon above the recommendations. The heart icon is a blue outline of a hand holding a heart, next to a solid blue heart icon. A "Sign In" button is visible in the top right corner of the YouTube interface.

# Research-World Motivation

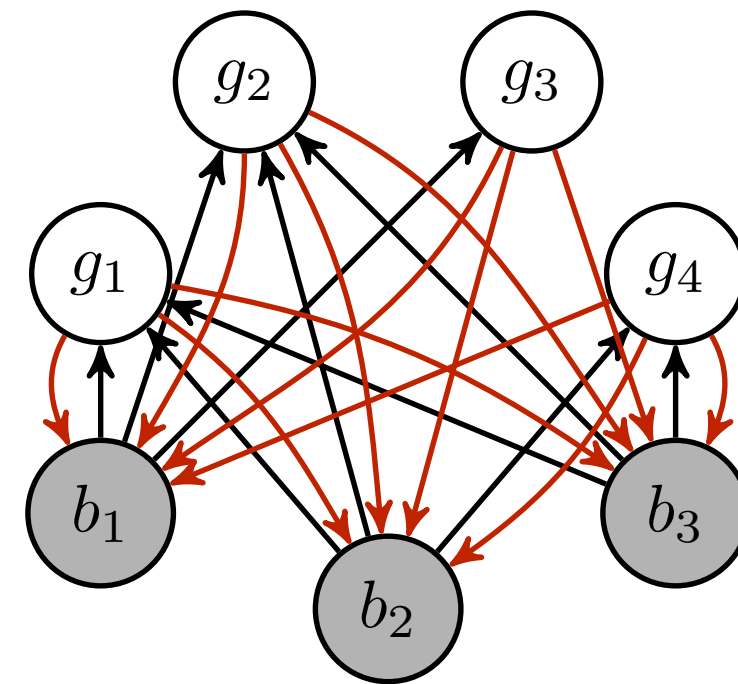
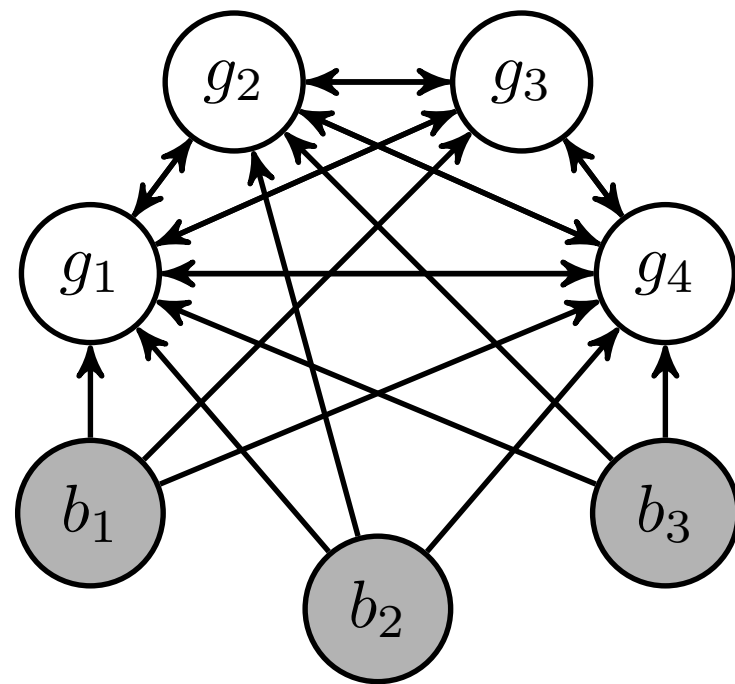
# Research-World Motivation



# Research-World Motivation

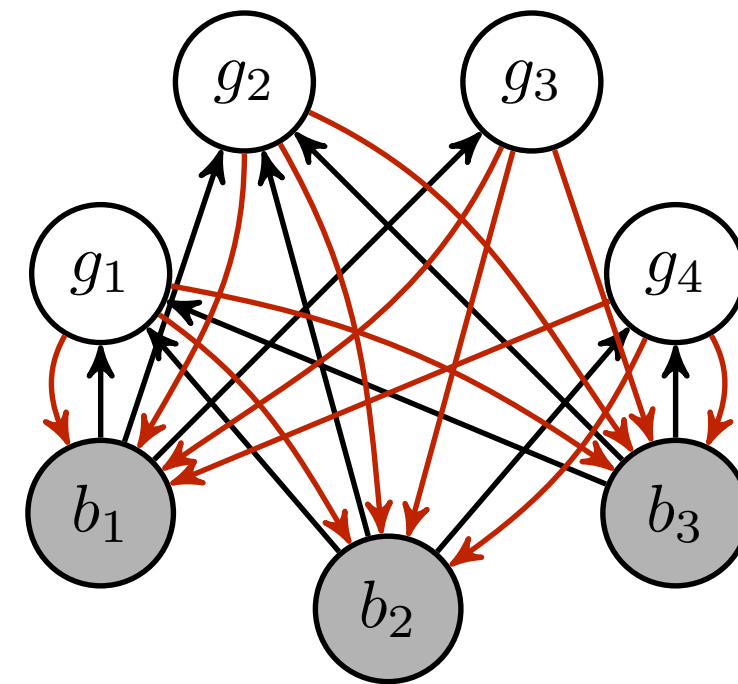
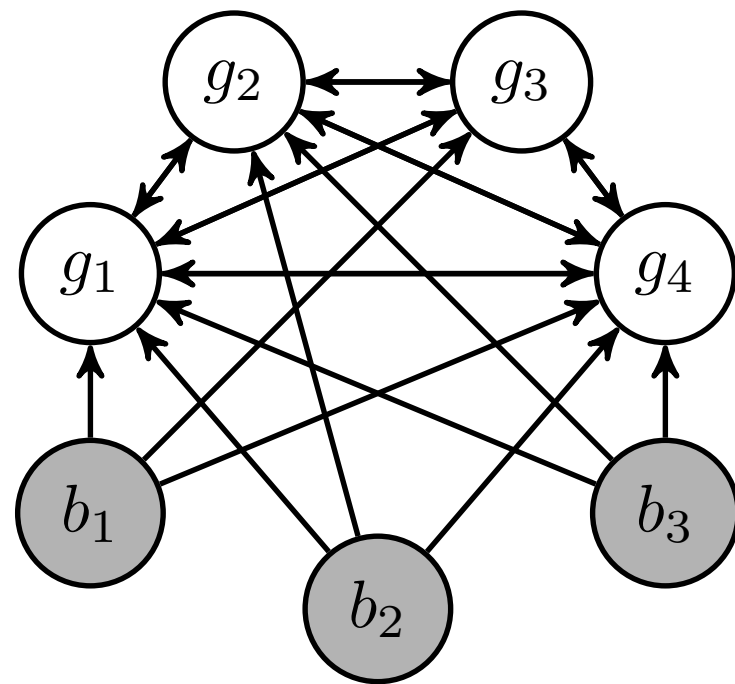


# Research-World Motivation



Both graphs minimize the *segregation* objective used in prior work (Fabbri et al. 2022)

# Research-World Motivation



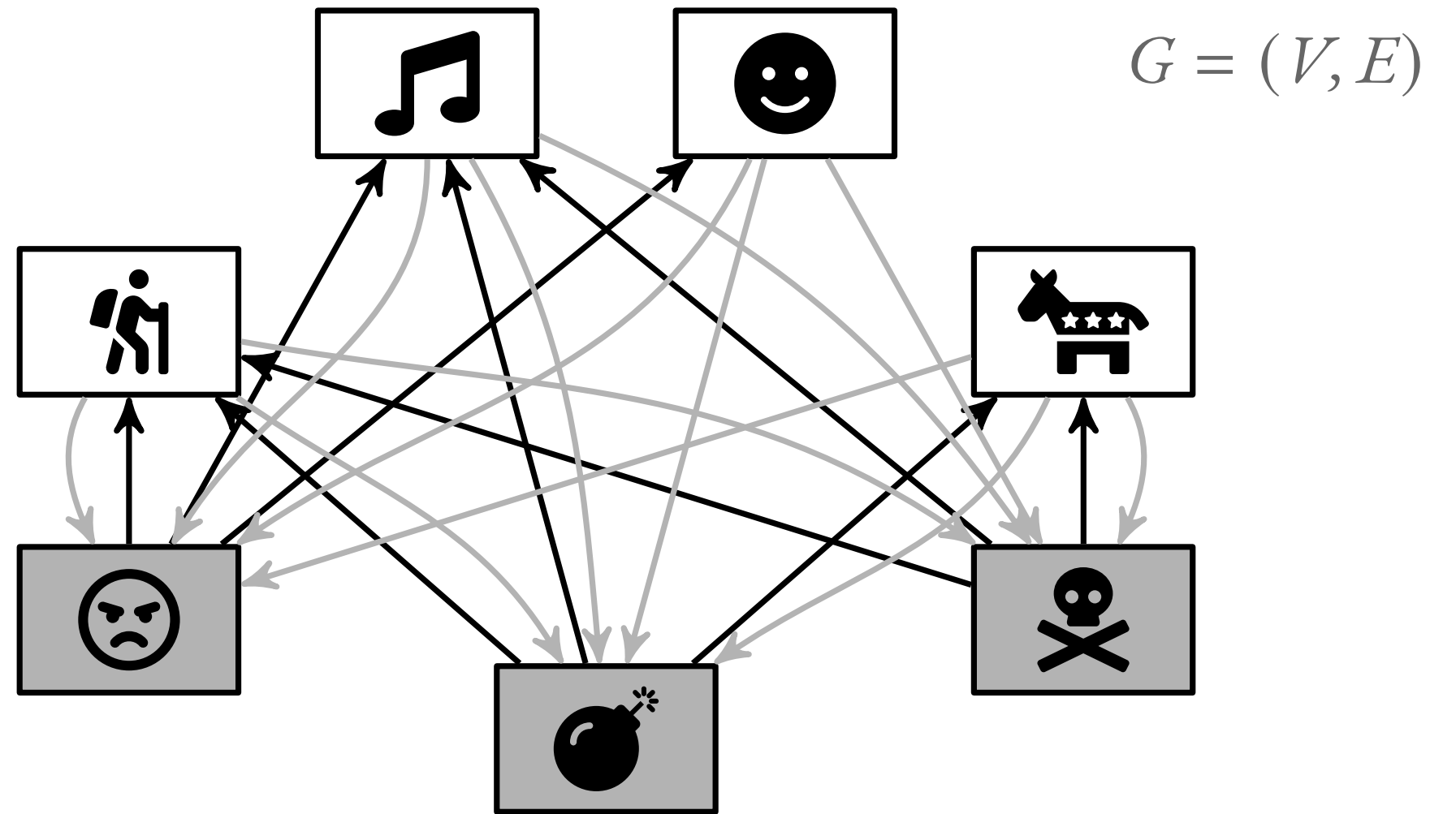
Both graphs minimize the *segregation* objective used in prior work (Fabbri et al. 2022)

We design an *exposure* objective that is minimized only by the left graph

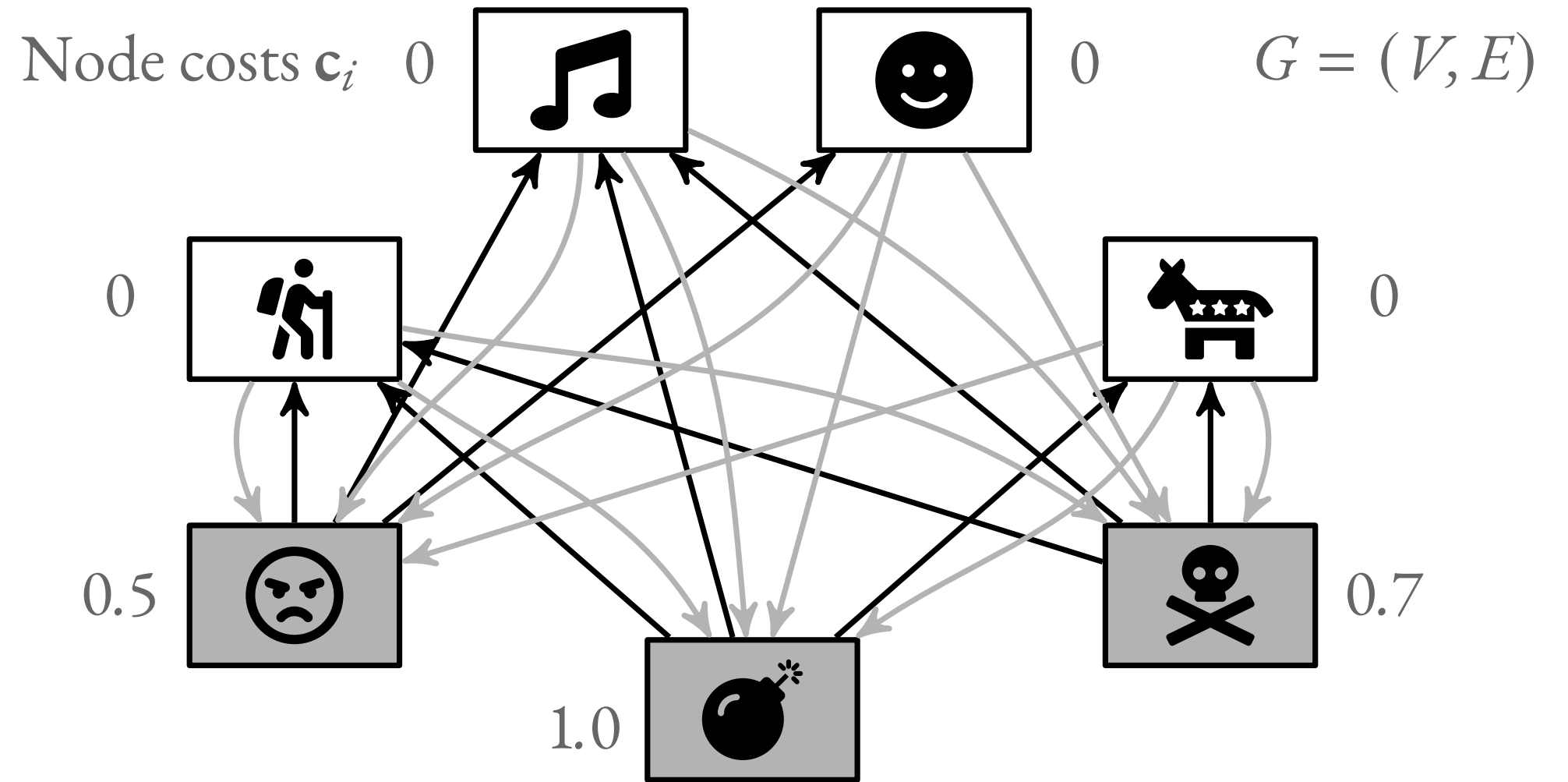
# Model



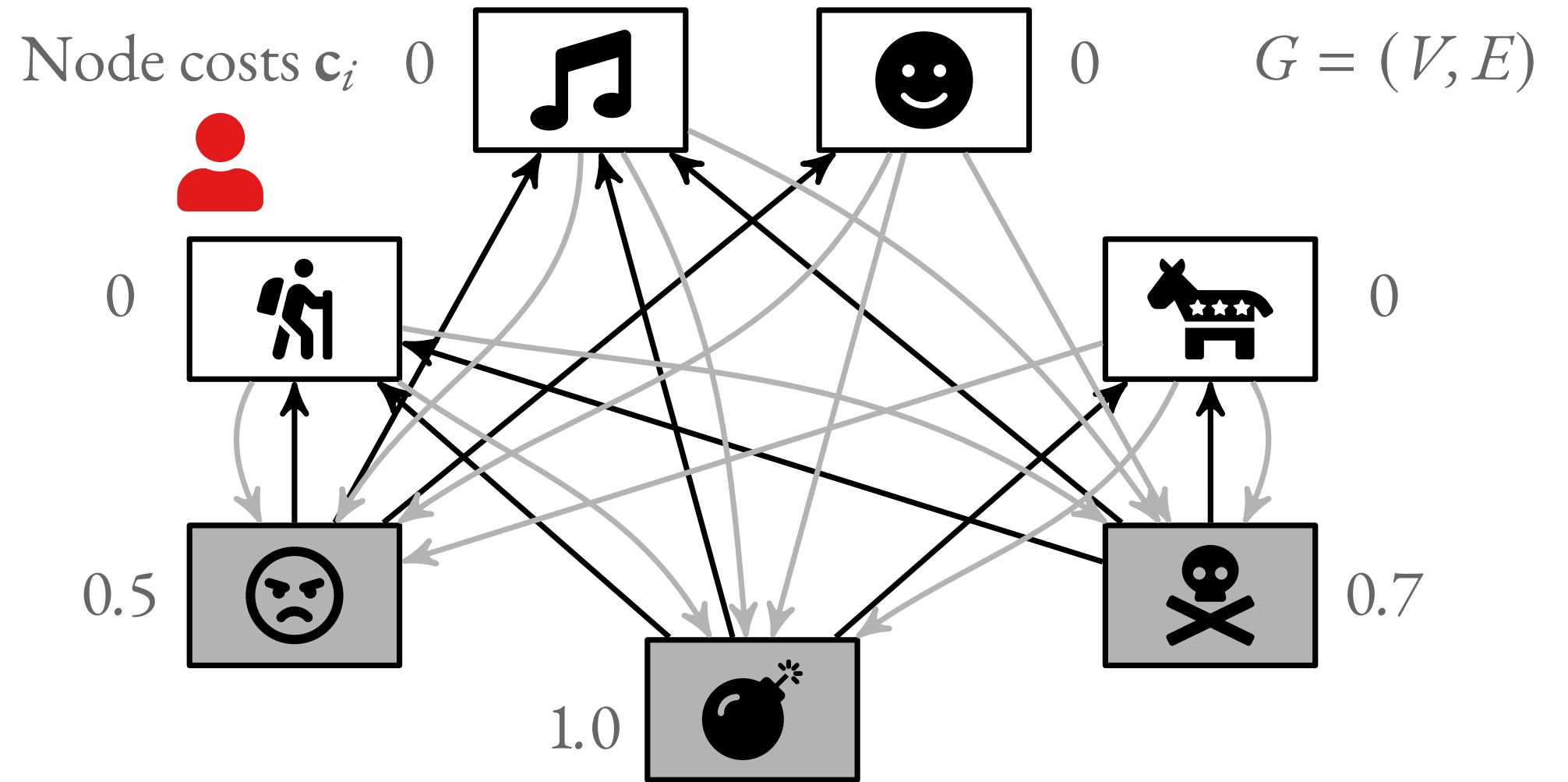
# Model



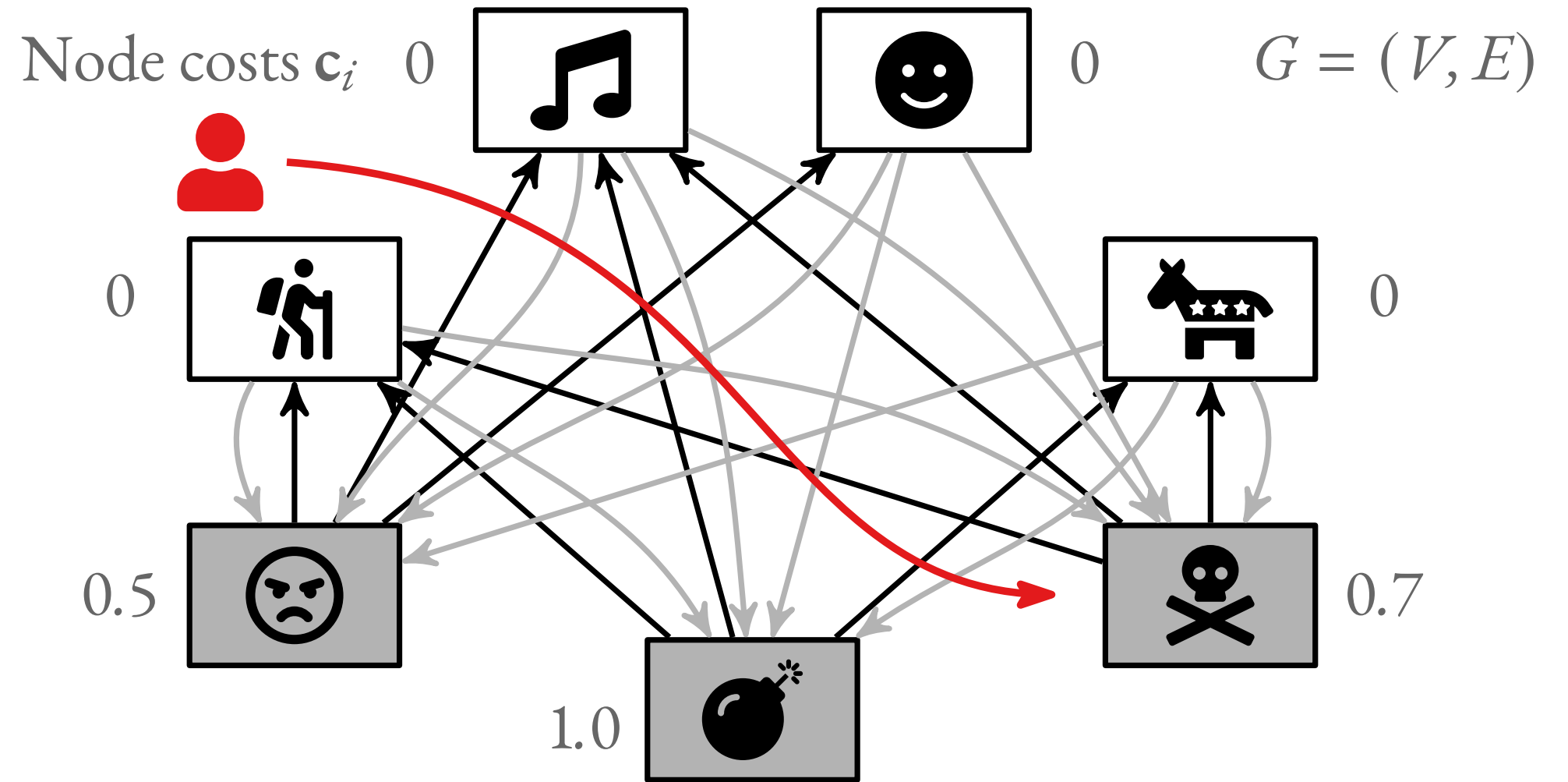
# Model



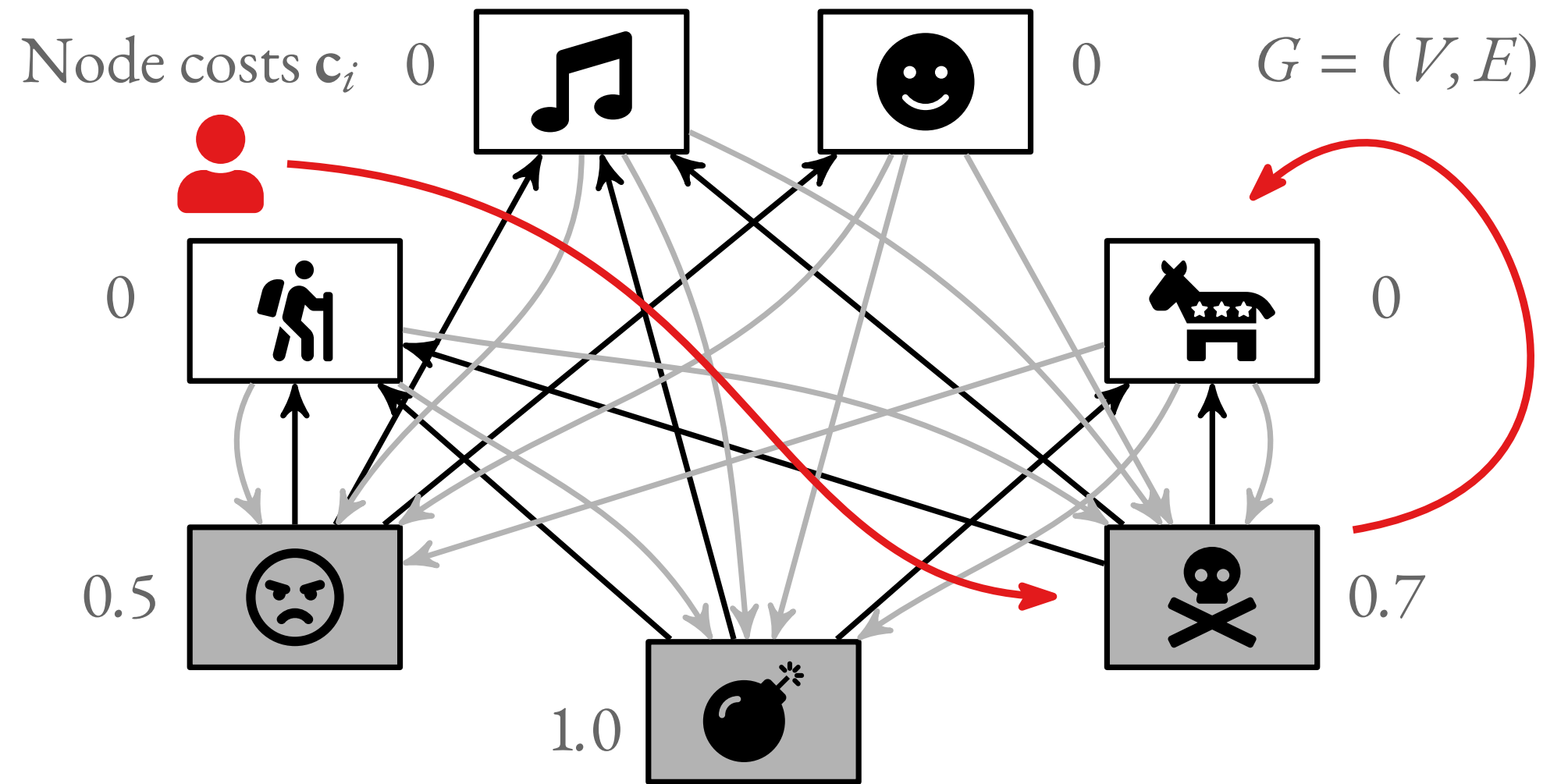
# Model



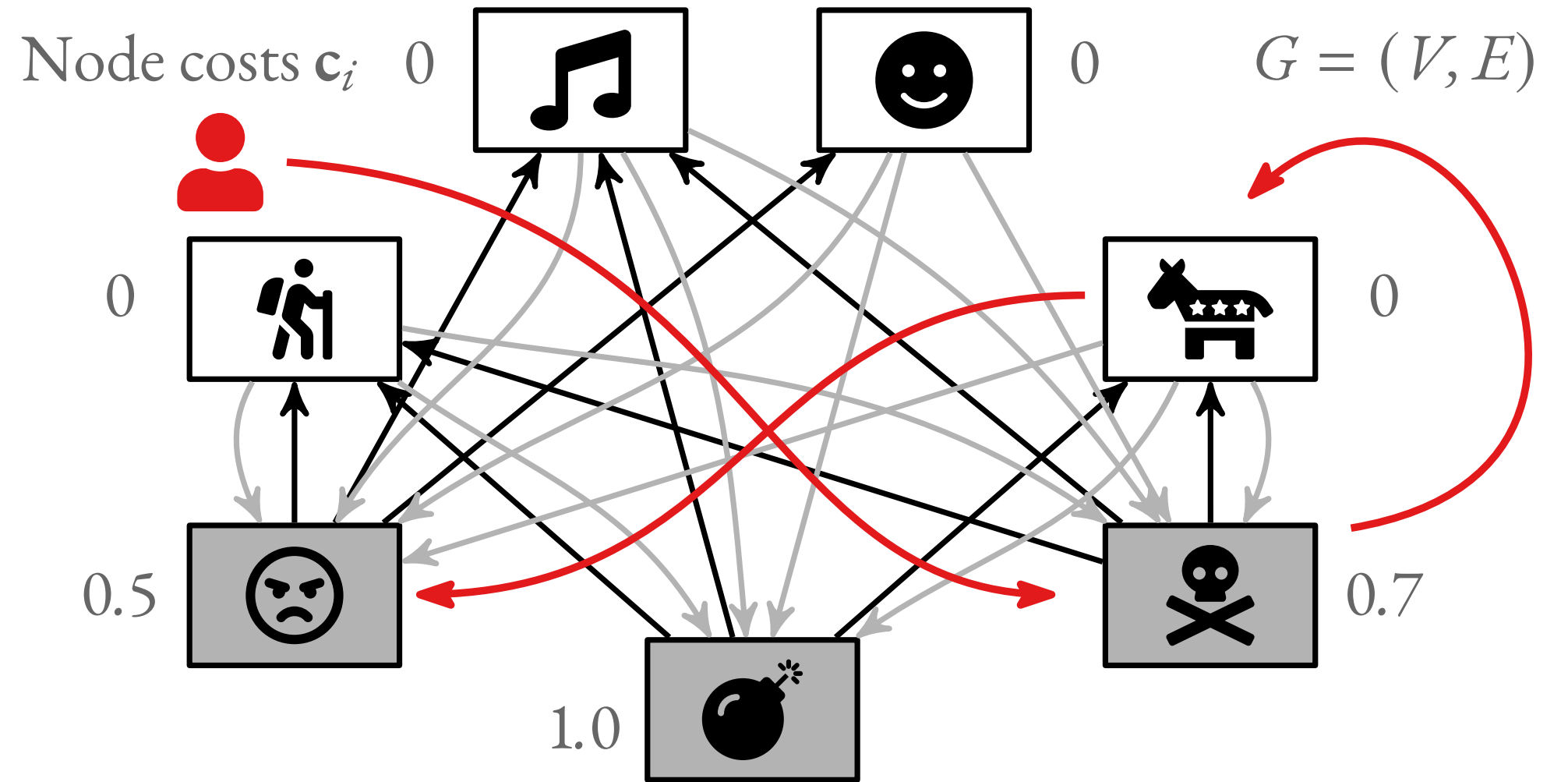
# Model



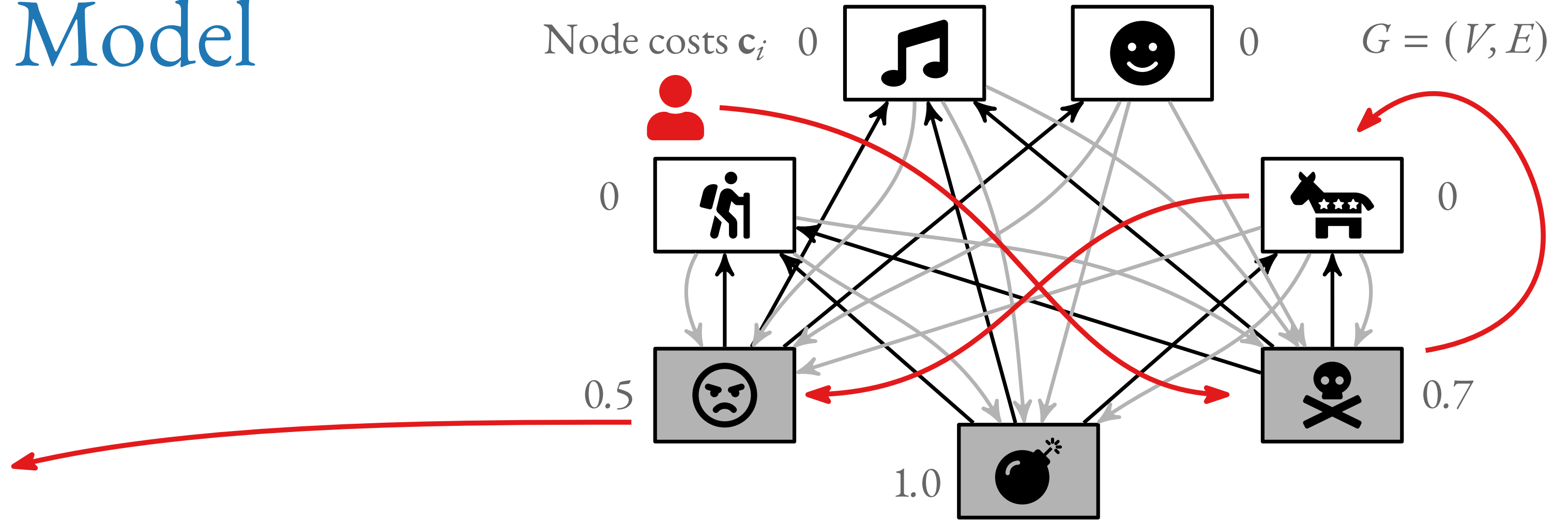
# Model



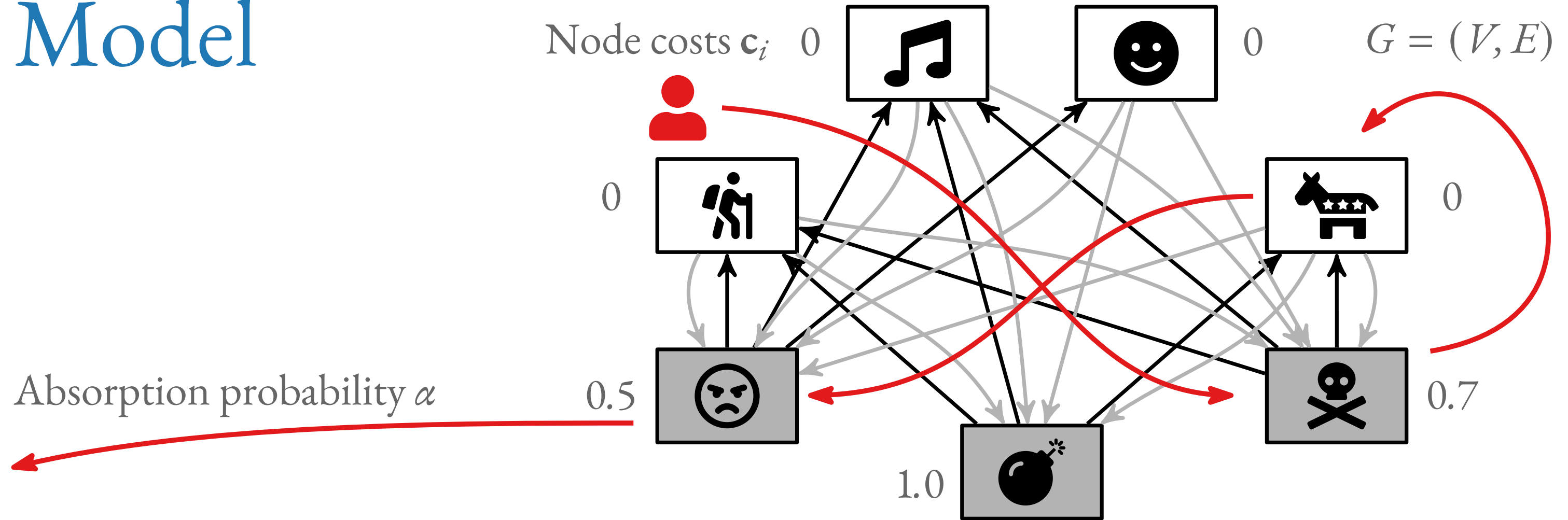
# Model



# Model

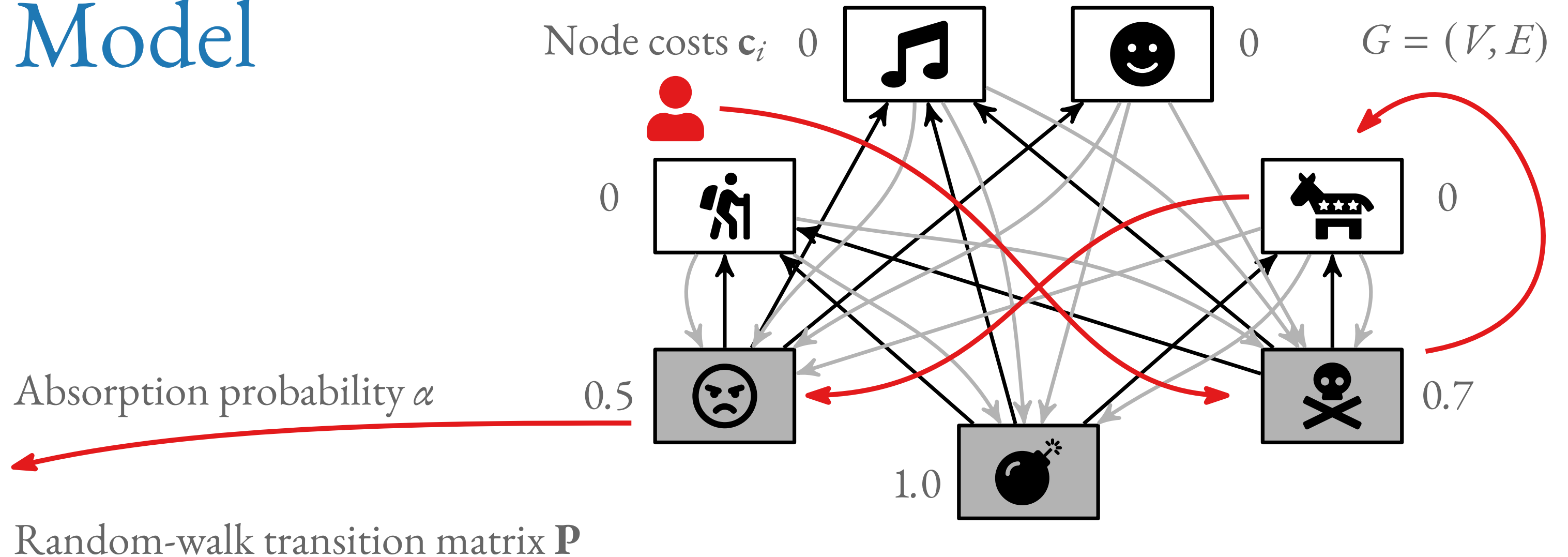


# Model

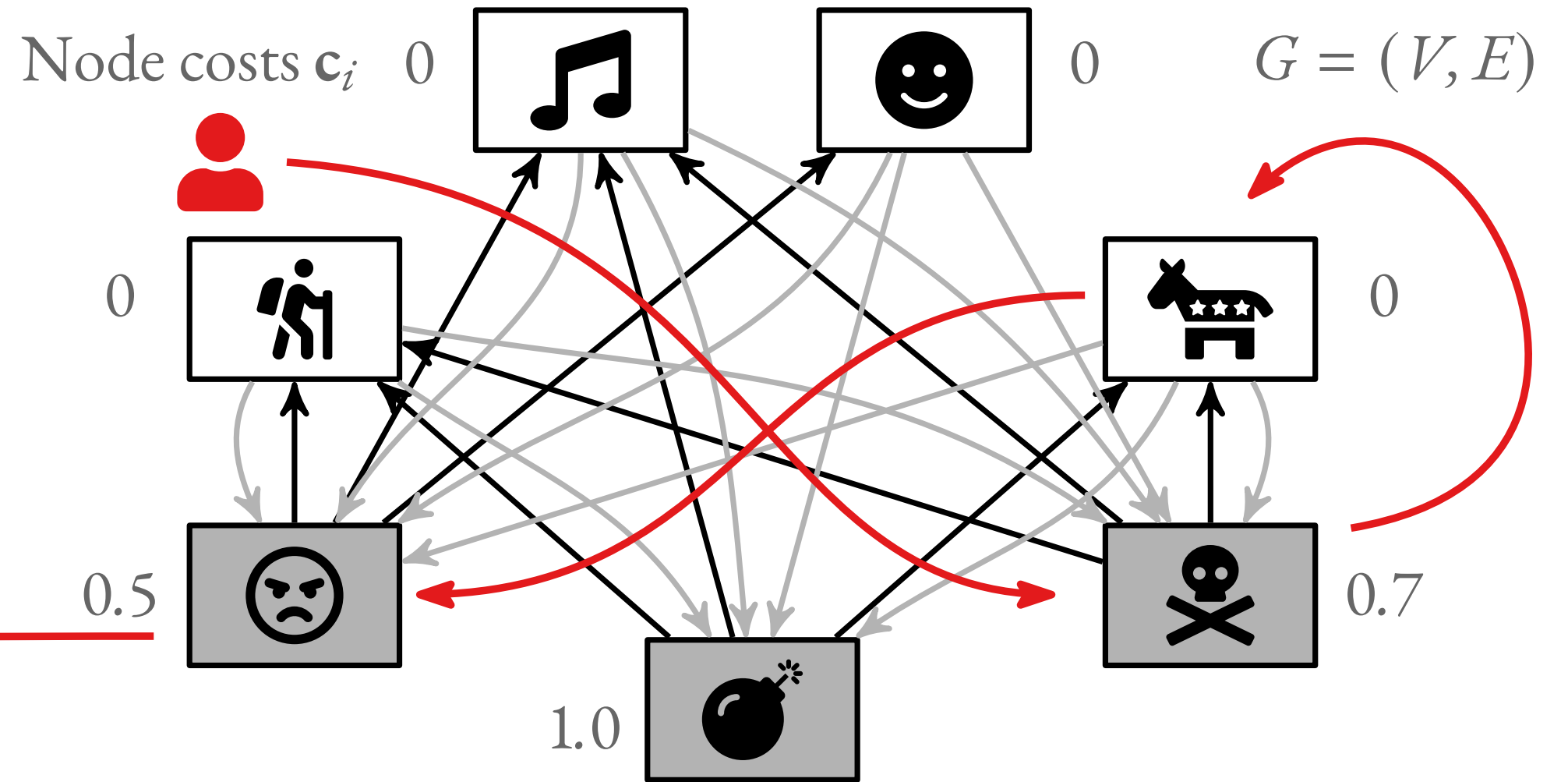




# Model



# Model

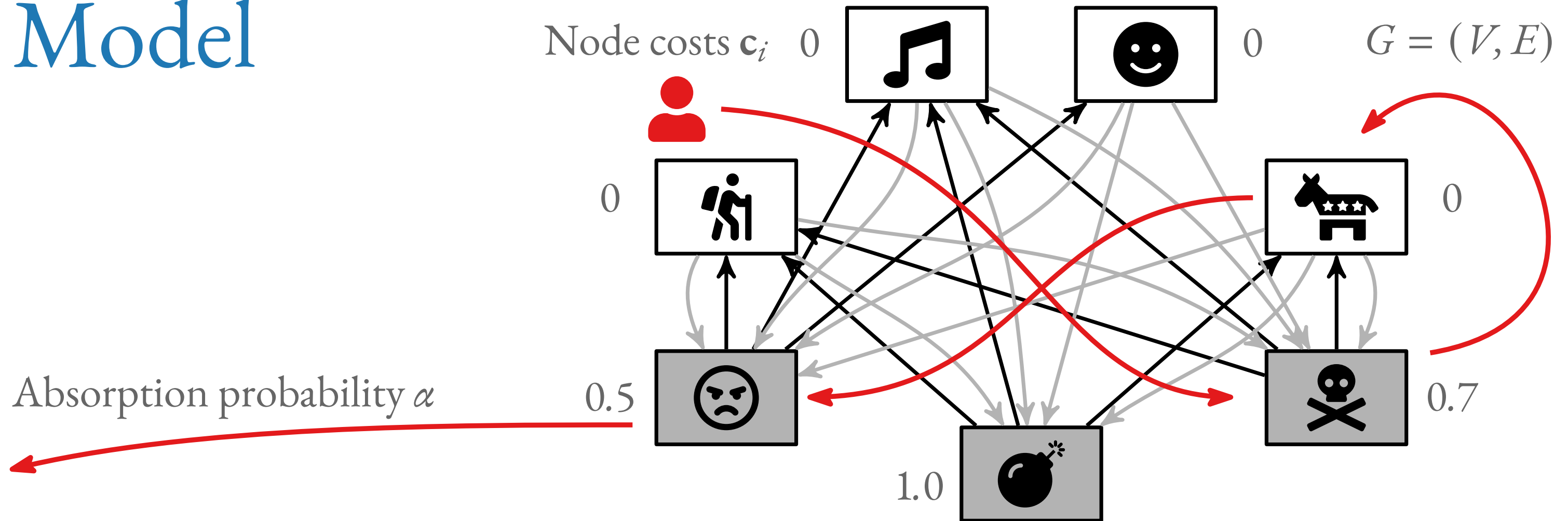


Absorption probability  $\alpha$

Random-walk transition matrix  $\mathbf{P}$

Fundamental matrix  $\mathbf{F} = (\mathbf{I} - \mathbf{P})^{-1}$

# Model

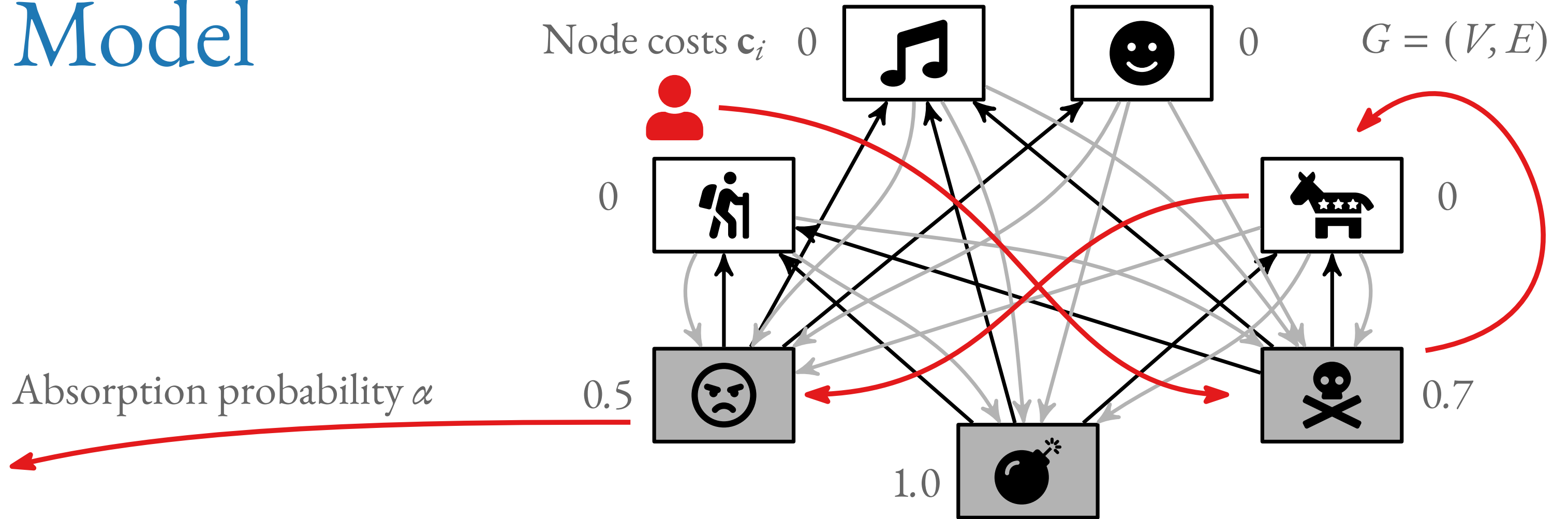


Random-walk transition matrix  $\mathbf{P}$

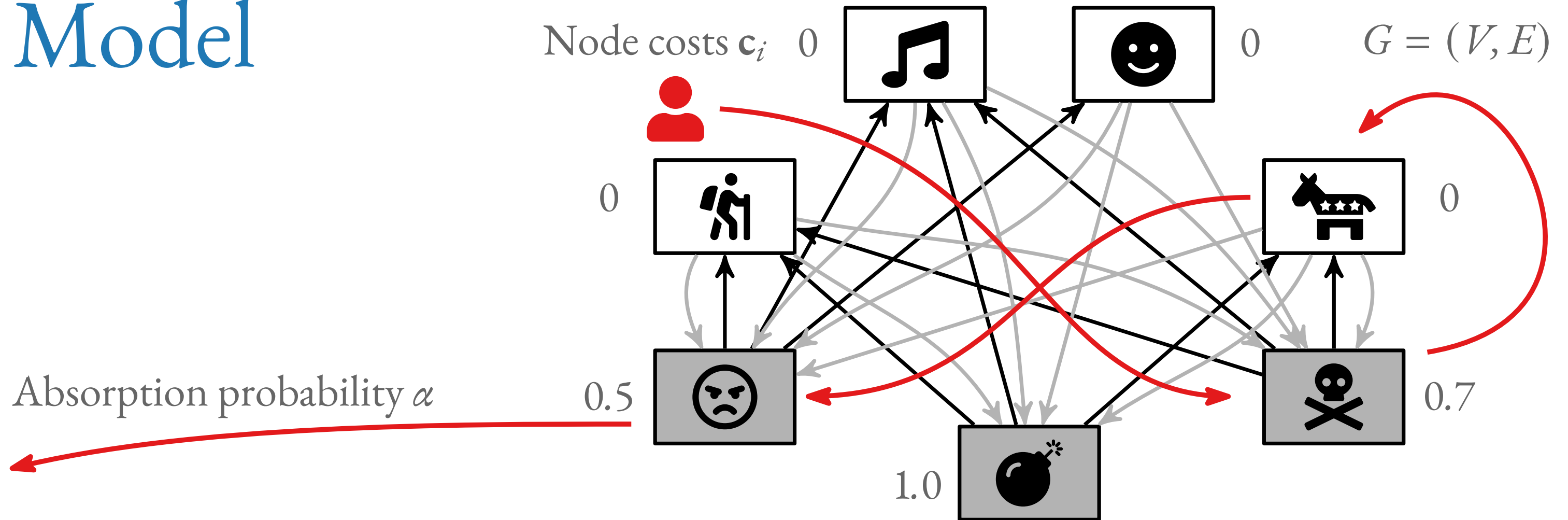
Fundamental matrix  $\mathbf{F} = (\mathbf{I} - \mathbf{P})^{-1}$

$\sum_{j \in V} \mathbf{F}_{ij} \mathbf{c}_j = \mathbf{e}_i^T \mathbf{F} \mathbf{c}$ : Expected exposure starting at  $i$

# Model



# Model



Random-walk transition matrix  $\mathbf{P}$

Fundamental matrix  $\mathbf{F} = (\mathbf{I} - \mathbf{P})^{-1}$

$\sum_{j \in V} \mathbf{F}_{ij} \mathbf{c}_j = \mathbf{e}_i^T \mathbf{F} \mathbf{c}$ : Expected exposure starting at  $i$

$f(G) = \sum_{i \in V} \mathbf{e}_i^T \mathbf{F} \mathbf{c} = \mathbf{1}^T \mathbf{F} \mathbf{c}$ : Expected total exposure

How can we reduce  $f(G)$ ?

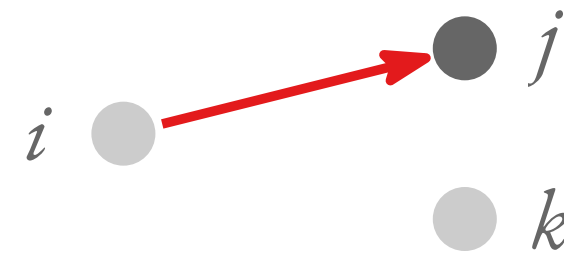
# Problem Statement

$f(G) = \mathbf{1}^T \mathbf{F} \mathbf{c}$ : Expected total exposure

# Problem Statement

$f(G) = \mathbf{1}^T \mathbf{F} \mathbf{c}$ : Expected total exposure

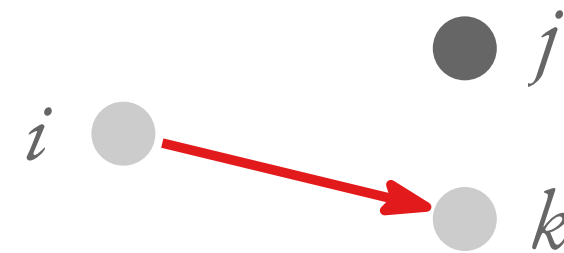
Edit operation: *Edge rewiring*



# Problem Statement

$f(G) = \mathbf{1}^T \mathbf{F} \mathbf{c}$ : Expected total exposure

Edit operation: *Edge rewiring*



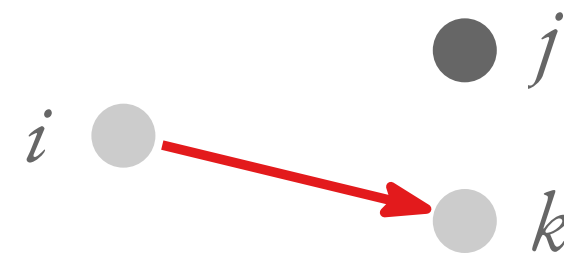


# Problem Statement

$f(G) = \mathbf{1}^T \mathbf{F} \mathbf{c}$ : Expected total exposure

$G_r$ :  $G$  after  $r$  rewirings

Edit operation: *Edge rewiring*



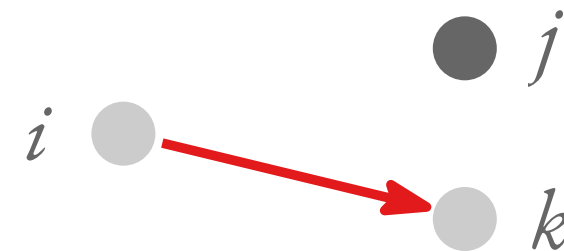
# Problem Statement

$f(G) = \mathbf{1}^T \mathbf{F} \mathbf{c}$ : Expected total exposure

$G_r$ :  $G$  after  $r$  rewirings

$r$ -Rewiring Exposure Minimization (REM)

Edit operation: *Edge rewiring*



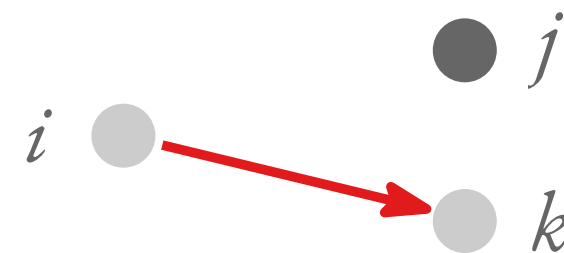
# Problem Statement

$f(G) = \mathbf{1}^T \mathbf{F} \mathbf{c}$ : Expected total exposure

$G_r$ :  $G$  after  $r$  rewirings

$r$ -Rewiring Exposure Minimization (REM)  
 $\min f(G_r)$

Edit operation: *Edge rewiring*

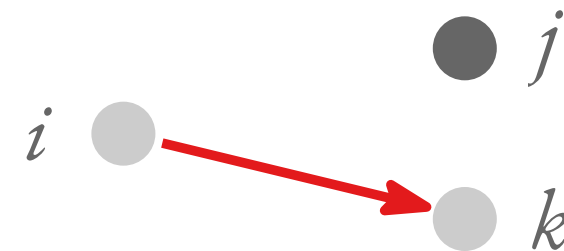


# Problem Statement

$f(G) = \mathbf{1}^T \mathbf{F} \mathbf{c}$ : Expected total exposure

Edit operation: *Edge rewiring*

$G_r$ :  $G$  after  $r$  rewirings



*r*-Rewiring Exposure Minimization (REM)

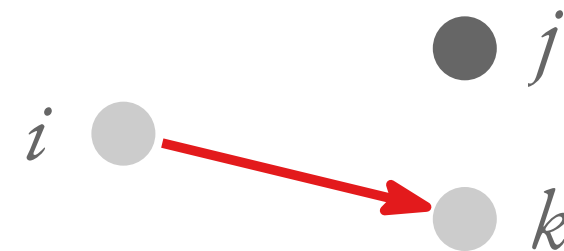
$$\min f(G_r) \Leftrightarrow \max f_{\Delta}(G, G_r) = f(G) - f(G_r)$$

# Problem Statement

$f(G) = \mathbf{1}^T \mathbf{F} \mathbf{c}$ : Expected total exposure

Edit operation: *Edge rewiring*

$G_r$ :  $G$  after  $r$  rewirings



*$r$ -Rewiring Exposure Minimization (REM)*

$$\min f(G_r) \Leftrightarrow \max f_{\Delta}(G, G_r) = f(G) - f(G_r)$$

More realistic & involved variant:  *$q$ -Relevant  $r$ -Rewiring Exposure Minimization (QREM)*

# Theoretical Analysis

# Theoretical Analysis

REM objective:  $\mathbf{1}^T \mathbf{F} \mathbf{c}$

# Theoretical Analysis

REM objective:  $\mathbf{1}^T \mathbf{F} \mathbf{c}$

Minimization problem:  $\min f(G_r)$



# Theoretical Analysis

REM objective:  $\mathbf{1}^T \mathbf{F} \mathbf{c}$

Minimization problem:  $\min f(G_r)$

NP-hard via reduction from Minimum Vertex Cover

# Theoretical Analysis

REM objective:  $\mathbf{1}^T \mathbf{F} \mathbf{c}$

Minimization problem:  $\min f(G_r)$

NP-hard via reduction from Minimum Vertex Cover

...likely also hard to approximate (based on Unique Games Conjecture)

# Theoretical Analysis

REM objective:  $\mathbf{1}^T \mathbf{F} \mathbf{c}$

Minimization problem:  $\min f(G_r)$

NP-hard via reduction from Minimum Vertex Cover

...likely also hard to approximate (based on Unique Games Conjecture)

Equivalent maximization problem:  $\max f_{\Delta}(G, G_r) = f(G) - f(G_r)$

# Theoretical Analysis

REM objective:  $\mathbf{1}^T \mathbf{F} \mathbf{c}$

Minimization problem:  $\min f(G_r)$

NP-hard via reduction from Minimum Vertex Cover

...likely also hard to approximate (based on Unique Games Conjecture)

Equivalent maximization problem:  $\max f_{\Delta}(G, G_r) = f(G) - f(G_r)$

General knowledge: Submodularity of max objective  $\rightarrow$  Greedy  $(1 - \frac{1}{e})$ -approximation

# Theoretical Analysis

REM objective:  $\mathbf{1}^T \mathbf{F} \mathbf{c}$

Minimization problem:  $\min f(G_r)$

NP-hard via reduction from Minimum Vertex Cover

...likely also hard to approximate (based on Unique Games Conjecture)

Equivalent maximization problem:  $\max f_{\Delta}(G, G_r) = f(G) - f(G_r)$

General knowledge: Submodularity of max objective  $\rightarrow$  Greedy  $(1 - \frac{1}{e})$ -approximation

Challenge: Rewiring problems not *generally* submodular

# Theoretical Analysis

REM objective:  $\mathbf{1}^T \mathbf{F} \mathbf{c}$

Minimization problem:  $\min f(G_r)$

NP-hard via reduction from Minimum Vertex Cover

...likely also hard to approximate (based on Unique Games Conjecture)

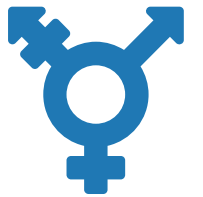
Equivalent maximization problem:  $\max f_{\Delta}(G, G_r) = f(G) - f(G_r)$

General knowledge: Submodularity of max objective  $\rightarrow$  Greedy  $(1 - \frac{1}{e})$ -approximation

Challenge: Rewiring problems not *generally* submodular

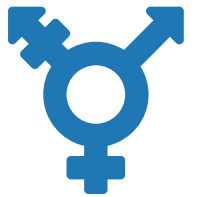
Idea: *Conditional* submodularity

# Greedy Algorithm: GAMINE



# Greedy Algorithm: GAMINE

Naïve implementation:  $O(rn^2(n + m))$

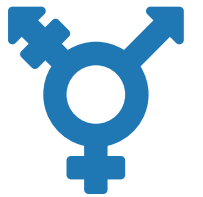




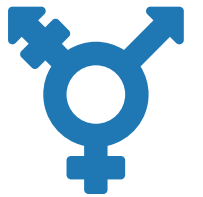
# Greedy Algorithm: GAMINE

Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion



# Greedy Algorithm: GAMINE

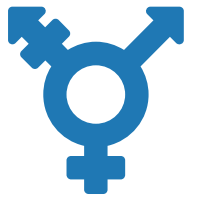


Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

# Greedy Algorithm: GAMINE



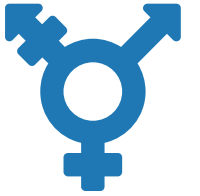
Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

---

**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

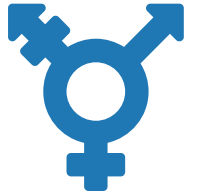
**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

```
1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Precompute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F} \mathbf{c}$   $\triangleright O(\kappa m)$ 
4:   Precompute  $\mathbf{1}^T \mathbf{F} \mathbf{u}$  for  $(i, j) \in E$   $\triangleright O(m)$ 
5:   Precompute  $\mathbf{F} \mathbf{u}$  for  $(i, j) \in E$   $\triangleright O(n^2)$ 
6:   Precompute  $\mathbf{v}^T \mathbf{F} \mathbf{c}$  for  $j \neq k \in V$   $\triangleright O(\kappa n^2)$ 
7:   1-REM()
8: return  $X$ 
```

```
9: function 1-REM()
10:    $\Delta, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:   for  $(i, j) \in E$  do  $\triangleright O(m)$ 
12:     for  $k \in V \setminus (\Gamma^+(i) \cup \{i\})$  do  $\triangleright O(n)$ 
13:        $\mathbf{u} \leftarrow \mathbf{P}[i, j] \mathbf{e}_i$ 
14:        $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$ 
15:        $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T \mathbf{F} \mathbf{u})(\mathbf{v}^T \mathbf{F} \mathbf{c})}{1 + \mathbf{v}^T \mathbf{F} \mathbf{u}}$   $\triangleright O(1)$ 
16:       if  $\Delta_{ijk} > \Delta$  then
17:          $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$ 
18:    $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:    $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:    $\mathbf{P}[i', j'] \leftarrow 0$ 
21:    $X \leftarrow X \cup \{(i', j', k')\}$ 
```

---

# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

New bottleneck: Number of candidate rewirings

---

**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

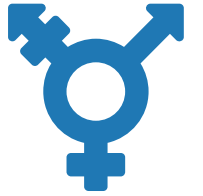
**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

```
1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Precompute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F} \mathbf{c}$  ▷  $O(\kappa m)$ 
4:   Precompute  $\mathbf{1}^T \mathbf{F} \mathbf{u}$  for  $(i, j) \in E$  ▷  $O(m)$ 
5:   Precompute  $\mathbf{F} \mathbf{u}$  for  $(i, j) \in E$  ▷  $O(n^2)$ 
6:   Precompute  $\mathbf{v}^T \mathbf{F} \mathbf{c}$  for  $j \neq k \in V$  ▷  $O(\kappa n^2)$ 
7:   1-REM()
8: return  $X$ 

9: function 1-REM()
10:   $\Delta, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:  for  $(i, j) \in E$  do ▷  $O(m)$ 
12:    for  $k \in V \setminus (\Gamma^+(i) \cup \{i\})$  do ▷  $O(n)$ 
13:       $\mathbf{u} \leftarrow \mathbf{P}[i, j] \mathbf{e}_i$ 
14:       $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$ 
15:       $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T \mathbf{F} \mathbf{u})(\mathbf{v}^T \mathbf{F} \mathbf{c})}{1 + \mathbf{v}^T \mathbf{F} \mathbf{u}}$  ▷  $O(1)$ 
16:      if  $\Delta_{ijk} > \Delta$  then
17:         $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$ 
18:   $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:   $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:   $\mathbf{P}[i', j'] \leftarrow 0$ 
21:   $X \leftarrow X \cup \{(i', j', k')\}$ 
```

---

# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

New bottleneck: Number of candidate rewirings

Reducing candidate rewirings:  $O(r\kappa(\Delta^+ n + m))$

---

**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

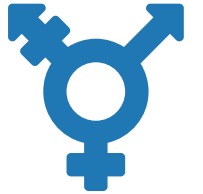
**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

```
1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Precompute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F} \mathbf{c}$   $\triangleright O(\kappa m)$ 
4:   Precompute  $\mathbf{1}^T \mathbf{F} \mathbf{u}$  for  $(i, j) \in E$   $\triangleright O(m)$ 
5:   Precompute  $\mathbf{F} \mathbf{u}$  for  $(i, j) \in E$   $\triangleright O(n^2)$ 
6:   Precompute  $\mathbf{v}^T \mathbf{F} \mathbf{c}$  for  $j \neq k \in V$   $\triangleright O(\kappa n^2)$ 
7:   1-REM()
8: return  $X$ 
```

```
9: function 1-REM()
10:    $\Delta, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:   for  $(i, j) \in E$  do  $\triangleright O(m)$ 
12:     for  $k \in V \setminus (\Gamma^+(i) \cup \{i\})$  do  $\triangleright O(n)$ 
13:        $\mathbf{u} \leftarrow \mathbf{P}[i, j] \mathbf{e}_i$ 
14:        $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$ 
15:        $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T \mathbf{F} \mathbf{u})(\mathbf{v}^T \mathbf{F} \mathbf{c})}{1 + \mathbf{v}^T \mathbf{F} \mathbf{u}}$   $\triangleright O(1)$ 
16:       if  $\Delta_{ijk} > \Delta$  then
17:          $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$ 
18:    $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:    $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:    $\mathbf{P}[i', j'] \leftarrow 0$ 
21:    $X \leftarrow X \cup \{(i', j', k')\}$ 
```

---

# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

New bottleneck: Number of candidate rewirings

Reducing candidate rewirings:  $O(r\kappa(\Delta^+ n + m))$

Only consider  $\Delta^+ + 2$  most promising targets

---

**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

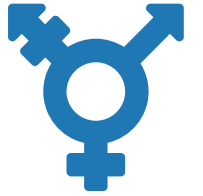
**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

```
1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Precompute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F} \mathbf{c}$  ▷  $O(\kappa m)$ 
4:   Precompute  $\mathbf{1}^T \mathbf{F} \mathbf{u}$  for  $(i, j) \in E$  ▷  $O(m)$ 
5:   Precompute  $\mathbf{F} \mathbf{u}$  for  $(i, j) \in E$  ▷  $O(n^2)$ 
6:   Precompute  $\mathbf{v}^T \mathbf{F} \mathbf{c}$  for  $j \neq k \in V$  ▷  $O(\kappa n^2)$ 
7:   1-REM()
8: return  $X$ 

9: function 1-REM()
10:   $\Delta, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:  for  $(i, j) \in E$  do ▷  $O(m)$ 
12:    for  $k \in V \setminus (\Gamma^+(i) \cup \{i\})$  do ▷  $O(n)$ 
13:       $\mathbf{u} \leftarrow \mathbf{P}[i, j] \mathbf{e}_i$ 
14:       $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$ 
15:       $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T \mathbf{F} \mathbf{u})(\mathbf{v}^T \mathbf{F} \mathbf{c})}{1 + \mathbf{v}^T \mathbf{F} \mathbf{u}}$  ▷  $O(1)$ 
16:      if  $\Delta_{ijk} > \Delta$  then
17:         $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$ 
18:   $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:   $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:   $\mathbf{P}[i', j'] \leftarrow 0$ 
21:   $X \leftarrow X \cup \{(i', j', k')\}$ 
```

---

# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

New bottleneck: Number of candidate rewirings

Reducing candidate rewirings:  $O(r\kappa(\Delta^+ n + m))$

Only consider  $\Delta^+ + 2$  most promising targets

Still suffices to identify rewiring maximizing  $\sigma\tau = (\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})$

---

**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

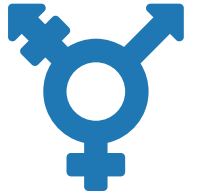
```
1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Precompute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F}\mathbf{c}$   $\triangleright O(\kappa m)$ 
4:   Precompute  $\mathbf{1}^T \mathbf{F}\mathbf{u}$  for  $(i, j) \in E$   $\triangleright O(m)$ 
5:   Precompute  $\mathbf{F}\mathbf{u}$  for  $(i, j) \in E$   $\triangleright O(n^2)$ 
6:   Precompute  $\mathbf{v}^T \mathbf{F}\mathbf{c}$  for  $j \neq k \in V$   $\triangleright O(\kappa n^2)$ 
7:   1-REM()
8: return  $X$ 

9: function 1-REM()
10:   $\Delta, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:  for  $(i, j) \in E$  do  $\triangleright O(m)$ 
12:    for  $k \in V \setminus (\Gamma^+(i) \cup \{i\})$  do  $\triangleright O(n)$ 
13:       $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$ 
14:       $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$ 
15:       $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})}{1 + \mathbf{v}^T \mathbf{F}\mathbf{u}}$   $\triangleright O(1)$ 
16:      if  $\Delta_{ijk} > \Delta$  then
17:         $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$ 
18:   $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:   $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:   $\mathbf{P}[i', j'] \leftarrow 0$ 
21:   $X \leftarrow X \cup \{(i', j', k')\}$ 
```

---



# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

New bottleneck: Number of candidate rewirings

Reducing candidate rewirings:  $O(r\kappa(\Delta^+ n + m))$

Only consider  $\Delta^+ + 2$  most promising targets

Still suffices to identify rewiring maximizing  $\sigma\tau = (\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})$

Can no longer afford to compute  $\varrho = 1 + \mathbf{v}^T \mathbf{F}\mathbf{u}$ , but...

**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

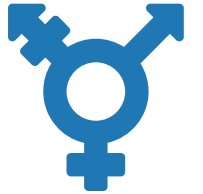
```

1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Precompute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F}\mathbf{c}$   $\triangleright O(\kappa m)$ 
4:   Precompute  $\mathbf{1}^T \mathbf{F}\mathbf{u}$  for  $(i, j) \in E$   $\triangleright O(m)$ 
5:   Precompute  $\mathbf{F}\mathbf{u}$  for  $(i, j) \in E$   $\triangleright O(n^2)$ 
6:   Precompute  $\mathbf{v}^T \mathbf{F}\mathbf{c}$  for  $j \neq k \in V$   $\triangleright O(\kappa n^2)$ 
7:   1-REM()
8: return  $X$ 

9: function 1-REM()
10:   $\Delta, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:  for  $(i, j) \in E$  do  $\triangleright O(m)$ 
12:    for  $k \in V \setminus (\Gamma^+(i) \cup \{i\})$  do  $\triangleright O(n)$ 
13:       $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$ 
14:       $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$ 
15:       $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})}{1 + \mathbf{v}^T \mathbf{F}\mathbf{u}}$   $\triangleright O(1)$ 
16:      if  $\Delta_{ijk} > \Delta$  then
17:         $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$ 
18:   $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:   $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:   $\mathbf{P}[i', j'] \leftarrow 0$ 
21:   $X \leftarrow X \cup \{(i', j', k')\}$ 

```

# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

New bottleneck: Number of candidate rewirings

Reducing candidate rewirings:  $O(r\kappa(\Delta^+ n + m))$

Only consider  $\Delta^+ + 2$  most promising targets

Still suffices to identify rewiring maximizing  $\sigma\tau = (\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})$

Can no longer afford to compute  $\rho = 1 + \mathbf{v}^T \mathbf{F}\mathbf{u}$ , but...

Correlation between  $\sigma\tau$  and  $\sigma\tau/\rho$  almost perfect

**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

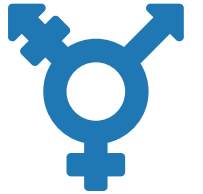
```

1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Precompute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F}\mathbf{c}$  ▷  $O(\kappa m)$ 
4:   Precompute  $\mathbf{1}^T \mathbf{F}\mathbf{u}$  for  $(i, j) \in E$  ▷  $O(m)$ 
5:   Precompute  $\mathbf{F}\mathbf{u}$  for  $(i, j) \in E$  ▷  $O(n^2)$ 
6:   Precompute  $\mathbf{v}^T \mathbf{F}\mathbf{c}$  for  $j \neq k \in V$  ▷  $O(\kappa n^2)$ 
7:   1-REM()
8: return  $X$ 

9: function 1-REM()
10:   $\Delta, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:  for  $(i, j) \in E$  do ▷  $O(m)$ 
12:    for  $k \in V \setminus (\Gamma^+(i) \cup \{i\})$  do ▷  $O(n)$ 
13:       $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$ 
14:       $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$ 
15:       $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})}{1 + \mathbf{v}^T \mathbf{F}\mathbf{u}}$  ▷  $O(1)$ 
16:      if  $\Delta_{ijk} > \Delta$  then
17:         $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$ 
18:   $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:   $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:   $\mathbf{P}[i', j'] \leftarrow 0$ 
21:   $X \leftarrow X \cup \{(i', j', k')\}$ 

```

# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

New bottleneck: Number of candidate rewirings

Reducing candidate rewirings:  $O(r\kappa(\Delta^+ n + m))$

Only consider  $\Delta^+ + 2$  most promising targets

Still suffices to identify rewiring maximizing  $\sigma\tau = (\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})$

Can no longer afford to compute  $\rho = 1 + \mathbf{v}^T \mathbf{F}\mathbf{u}$ , but...

Correlation between  $\sigma\tau$  and  $\sigma\tau/\rho$  almost perfect

$\sigma\tau > \sigma\tau'$  almost always implies  $\sigma\tau/\rho > \sigma\tau'/\rho'$

**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

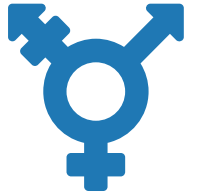
```

1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Precompute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F}\mathbf{c}$  ▷  $O(\kappa m)$ 
4:   Precompute  $\mathbf{1}^T \mathbf{F}\mathbf{u}$  for  $(i, j) \in E$  ▷  $O(m)$ 
5:   Precompute  $\mathbf{F}\mathbf{u}$  for  $(i, j) \in E$  ▷  $O(n^2)$ 
6:   Precompute  $\mathbf{v}^T \mathbf{F}\mathbf{c}$  for  $j \neq k \in V$  ▷  $O(\kappa n^2)$ 
7:   1-REM()
8: return  $X$ 

9: function 1-REM()
10:   $\Delta, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:  for  $(i, j) \in E$  do ▷  $O(m)$ 
12:    for  $k \in V \setminus (\Gamma^+(i) \cup \{i\})$  do ▷  $O(n)$ 
13:       $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$ 
14:       $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$ 
15:       $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})}{1 + \mathbf{v}^T \mathbf{F}\mathbf{u}}$  ▷  $O(1)$ 
16:      if  $\Delta_{ijk} > \Delta$  then
17:         $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$ 
18:   $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:   $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:   $\mathbf{P}[i', j'] \leftarrow 0$ 
21:   $X \leftarrow X \cup \{(i', j', k')\}$ 

```

# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

New bottleneck: Number of candidate rewirings

Reducing candidate rewirings:  $O(r\kappa(\Delta^+ n + m))$

Only consider  $\Delta^+ + 2$  most promising targets

Still suffices to identify rewiring maximizing  $\sigma\tau = (\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})$

Can no longer afford to compute  $\rho = 1 + \mathbf{v}^T \mathbf{F}\mathbf{u}$ , but...

Correlation between  $\sigma\tau$  and  $\sigma\tau/\rho$  almost perfect

$\sigma\tau > \sigma\tau'$  almost always implies  $\sigma\tau/\rho > \sigma\tau'/\rho'$

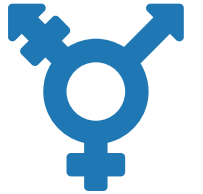
**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

```
1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Compute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F}\mathbf{c}$  ▷  $O(\kappa m)$ 
4:   Compute  $\mathbf{1}^T \mathbf{F}\mathbf{u}$  for all  $(i, j) \in E$  ▷  $O(m)$ 
5:    $K \leftarrow \{k \mid \mathbf{F}\mathbf{c}[k] \in \{(\Delta^+ + 2) \text{ smallest } \mathbf{F}\mathbf{c}\text{-values}\}\}$  ▷  $O(n)$ 
6:   Compute  $\mathbf{v}^T \mathbf{F}\mathbf{c}$  for  $j \in V$  and  $k \in K$  ▷  $O(\Delta^+ n)$ 
7:   1-REM()
8: return  $X$ 

9: function 1-REM()
10:   $\hat{\Delta}, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:  for  $(i, j) \in E$  do ▷  $O(m)$ 
12:    for  $k_{ij} \in K \setminus (\Gamma^+(i) \cup \{i\})$  do ▷  $O(\Delta^+)$ 
13:       $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$ 
14:       $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_{k_{ij}}$ 
15:       $\hat{\Delta}_{ijk} \leftarrow (\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})$  ▷  $O(1)$ 
16:      if  $\hat{\Delta}_{ijk} > \hat{\Delta}$  then
17:         $\hat{\Delta}, i', j', k' \leftarrow \hat{\Delta}_{ijk}, i, j, k_{ij}$ 
18:   $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:   $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:   $\mathbf{P}[i', j'] \leftarrow 0$ 
21:   $X \leftarrow X \cup \{(i', j', k')\}$ 
```

# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

New bottleneck: Number of candidate rewirings

Reducing candidate rewirings:  $O(r\kappa(\Delta^+ n + m))$

Only consider  $\Delta^+ + 2$  most promising targets

Still suffices to identify rewiring maximizing  $\sigma\tau = (\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})$

Can no longer afford to compute  $\rho = 1 + \mathbf{v}^T \mathbf{F}\mathbf{u}$ , but...

Correlation between  $\sigma\tau$  and  $\sigma\tau/\rho$  almost perfect

$\sigma\tau > \sigma\tau'$  almost always implies  $\sigma\tau/\rho > \sigma\tau'/\rho'$

**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

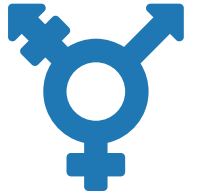
**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

```
1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Compute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F}\mathbf{c}$  ▷  $O(\kappa m)$ 
4:   Compute  $\mathbf{1}^T \mathbf{F}\mathbf{u}$  for all  $(i, j) \in E$  ▷  $O(m)$ 
5:    $K \leftarrow \{k \mid \mathbf{F}\mathbf{c}[k] \in \{(\Delta^+ + 2) \text{ smallest } \mathbf{F}\mathbf{c}\text{-values}\}\}$  ▷  $O(n)$ 
6:   Compute  $\mathbf{v}^T \mathbf{F}\mathbf{c}$  for  $j \in V$  and  $k \in K$  ▷  $O(\Delta^+ n)$ 
7:   1-REM()
8: return  $X$ 

9: function 1-REM()
10:   $\hat{\Delta}, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:  for  $(i, j) \in E$  do ▷  $O(m)$ 
12:    for  $k_{ij} \in K \setminus (\Gamma^+(i) \cup \{i\})$  do ▷  $O(\Delta^+)$ 
13:       $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$ 
14:       $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_{k_{ij}}$ 
15:       $\hat{\Delta}_{ijk} \leftarrow (\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})$  ▷  $O(1)$ 
16:      if  $\hat{\Delta}_{ijk} > \hat{\Delta}$  then
17:         $\Delta, i', j', k' \leftarrow \hat{\Delta}_{ijk}, i, j, k_{ij}$ 
18:   $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:   $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:   $\mathbf{P}[i', j'] \leftarrow 0$ 
21:   $X \leftarrow X \cup \{(i', j', k')\}$ 
```



# Greedy Algorithm: GAMINE



Naïve implementation:  $O(rn^2(n + m))$

Bottleneck: Matrix inversion

Forgoing matrix inversion:  $O(r\kappa n(n + m))$

Approximate inverse via  $\kappa$  power iterations

New bottleneck: Number of candidate rewirings

Reducing candidate rewirings:  $O(r\kappa(\Delta^+ n + m))$

Only consider  $\Delta^+ + 2$  most promising targets

Still suffices to identify rewiring maximizing  $\sigma\tau = (\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})$

Can no longer afford to compute  $\rho = 1 + \mathbf{v}^T \mathbf{F}\mathbf{u}$ , but...

Correlation between  $\sigma\tau$  and  $\sigma\tau/\rho$  almost perfect

$\sigma\tau > \sigma\tau'$  almost always implies  $\sigma\tau/\rho > \sigma\tau'/\rho'$

$O(r\kappa(\Delta^+ n + m)) = O(n)$  under realistic assumptions

**Input:** Graph  $G = (V, E)$ , transition matrix  $\mathbf{P}$ , costs  $\mathbf{c}$ , budget  $r$

**Output:** Set of  $r$  rewirings  $X$  of shape  $(i, j, k)$

```
1:  $X \leftarrow \emptyset$ 
2: for  $i \in \mathbb{N}_{\leq r}$  do
3:   Compute  $\mathbf{1}^T \mathbf{F}$  and  $\mathbf{F}\mathbf{c}$  ▷  $O(\kappa m)$ 
4:   Compute  $\mathbf{1}^T \mathbf{F}\mathbf{u}$  for all  $(i, j) \in E$  ▷  $O(m)$ 
5:    $K \leftarrow \{k \mid \mathbf{F}\mathbf{c}[k] \in \{(\Delta^+ + 2) \text{ smallest } \mathbf{F}\mathbf{c}\text{-values}\}\}$  ▷  $O(n)$ 
6:   Compute  $\mathbf{v}^T \mathbf{F}\mathbf{c}$  for  $j \in V$  and  $k \in K$  ▷  $O(\Delta^+ n)$ 
7:   1-REM()
8: return  $X$ 
```

```
9: function 1-REM()
10:   $\hat{\Delta}, i', j', k' \leftarrow 0, \perp, \perp, \perp$ 
11:  for  $(i, j) \in E$  do ▷  $O(m)$ 
12:    for  $k_{ij} \in K \setminus (\Gamma^+(i) \cup \{i\})$  do ▷  $O(\Delta^+)$ 
13:       $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$ 
14:       $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_{k_{ij}}$ 
15:       $\hat{\Delta}_{ijk} \leftarrow (\mathbf{1}^T \mathbf{F}\mathbf{u})(\mathbf{v}^T \mathbf{F}\mathbf{c})$  ▷  $O(1)$ 
16:      if  $\hat{\Delta}_{ijk} > \hat{\Delta}$  then
17:         $\hat{\Delta}, i', j', k' \leftarrow \hat{\Delta}_{ijk}, i, j, k_{ij}$ 
18:   $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$ 
19:   $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$ 
20:   $\mathbf{P}[i', j'] \leftarrow 0$ 
21:   $X \leftarrow X \cup \{(i', j', k')\}$ 
```

# Experimental Evaluation

# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5		202 075	[0.900, 8.475]
	10	40 415	404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10k (3 · 6 graphs)	5		752 860	[0.806, 5.785]
	10	150 572	1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5		59 655	[4.217, 9.533]
	10	11 931	119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5		287 235	[4.609, 11.068]
	10	57 447	574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5		467 275	[5.565, 11.896]
	10	93 455	934 550	[5.315, 11.660]
	20		1 869 100	[5.138, 11.517]



# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5 10 20	40 415	202 075 404 150 808 300	[0.900, 8.475] [0.938, 8.701] [0.989, 9.444]
YT-10k (3 · 6 graphs)	5 10 20	150 572	752 860 1 505 720 3 011 440	[0.806, 5.785] [0.883, 7.576] [0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5 10 20	11 931	59 655 119 310 238 620	[4.217, 9.533] [4.248, 9.567] [4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5 10 20	57 447	287 235 574 470 1 148 940	[4.609, 11.068] [4.392, 10.769] [4.329, 10.741]
NF-ALL (3 · 6 graphs)	5 10 20	93 455	467 275 934 550 1 869 100	[5.565, 11.896] [5.315, 11.660] [5.138, 11.517]

| Synthetic data

# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5 10 20	40 415	202 075 404 150 808 300	[0.900, 8.475] [0.938, 8.701] [0.989, 9.444]
YT-10k (3 · 6 graphs)	5 10 20	150 572	752 860 1 505 720 3 011 440	[0.806, 5.785] [0.883, 7.576] [0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5 10 20	11 931	59 655 119 310 238 620	[4.217, 9.533] [4.248, 9.567] [4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5 10 20	57 447	287 235 574 470 1 148 940	[4.609, 11.068] [4.392, 10.769] [4.329, 10.741]
NF-ALL (3 · 6 graphs)	5 10 20	93 455	467 275 934 550 1 869 100	[5.565, 11.896] [5.315, 11.660] [5.138, 11.517]

Synthetic data

YouTube video data

# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5 10 20	40 415	202 075 404 150 808 300	[0.900, 8.475] [0.938, 8.701] [0.989, 9.444]
YT-10k (3 · 6 graphs)	5 10 20	150 572	752 860 1 505 720 3 011 440	[0.806, 5.785] [0.883, 7.576] [0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5 10 20	11 931	59 655 119 310 238 620	[4.217, 9.533] [4.248, 9.567] [4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5 10 20	57 447	287 235 574 470 1 148 940	[4.609, 11.068] [4.392, 10.769] [4.329, 10.741]
NF-ALL (3 · 6 graphs)	5 10 20	93 455	467 275 934 550 1 869 100	[5.565, 11.896] [5.315, 11.660] [5.138, 11.517]

Synthetic data

YouTube video data

NELA-GT news data

# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5		202 075	[0.900, 8.475]
	10	40 415	404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10k (3 · 6 graphs)	5		752 860	[0.806, 5.785]
	10	150 572	1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5		59 655	[4.217, 9.533]
	10	11 931	119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5		287 235	[4.609, 11.068]
	10	57 447	574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5		467 275	[5.565, 11.896]
	10	93 455	934 550	[5.315, 11.660]
	20		1 869 100	[5.138, 11.517]

Synthetic data

Competition

YouTube video data

NELA-GT news data

# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5		202 075	[0.900, 8.475]
	10	40 415	404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10k (3 · 6 graphs)	5		752 860	[0.806, 5.785]
	10	150 572	1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5		59 655	[4.217, 9.533]
	10	11 931	119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5		287 235	[4.609, 11.068]
	10	57 447	574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5		467 275	[5.565, 11.896]
	10	93 455	934 550	[5.315, 11.660]
	20		1 869 100	[5.138, 11.517]

Synthetic data

Competition

Four baselines (ablations)

YouTube video data

NELA-GT news data

# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5 10 20	40 415	202 075 404 150 808 300	[0.900, 8.475] [0.938, 8.701] [0.989, 9.444]
YT-10k (3 · 6 graphs)	5 10 20	150 572	752 860 1 505 720 3 011 440	[0.806, 5.785] [0.883, 7.576] [0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5 10 20	11 931	59 655 119 310 238 620	[4.217, 9.533] [4.248, 9.567] [4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5 10 20	57 447	287 235 574 470 1 148 940	[4.609, 11.068] [4.392, 10.769] [4.329, 10.741]
NF-ALL (3 · 6 graphs)	5 10 20	93 455	467 275 934 550 1 869 100	[5.565, 11.896] [5.315, 11.660] [5.138, 11.517]

Synthetic data

YouTube video data

NELA-GT news data

Competition

Four baselines (ablations)

One competitor  
(Fabbri et al., “MMS”) for QREM

# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5 10 20	40 415	202 075 404 150 808 300	[0.900, 8.475] [0.938, 8.701] [0.989, 9.444]
YT-10k (3 · 6 graphs)	5 10 20	150 572	752 860 1 505 720 3 011 440	[0.806, 5.785] [0.883, 7.576] [0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5 10 20	11 931	59 655 119 310 238 620	[4.217, 9.533] [4.248, 9.567] [4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5 10 20	57 447	287 235 574 470 1 148 940	[4.609, 11.068] [4.392, 10.769] [4.329, 10.741]
NF-ALL (3 · 6 graphs)	5 10 20	93 455	467 275 934 550 1 869 100	[5.565, 11.896] [5.315, 11.660] [5.138, 11.517]

Synthetic data

YouTube video data

NELA-GT news data

Competition

Four baselines (ablations)

One competitor  
(Fabbri et al., “MMS”) for QREM

Implementation



# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5 10 20	40 415	202 075 404 150 808 300	[0.900, 8.475] [0.938, 8.701] [0.989, 9.444]
YT-10k (3 · 6 graphs)	5 10 20	150 572	752 860 1 505 720 3 011 440	[0.806, 5.785] [0.883, 7.576] [0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5 10 20	11 931	59 655 119 310 238 620	[4.217, 9.533] [4.248, 9.567] [4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5 10 20	57 447	287 235 574 470 1 148 940	[4.609, 11.068] [4.392, 10.769] [4.329, 10.741]
NF-ALL (3 · 6 graphs)	5 10 20	93 455	467 275 934 550 1 869 100	[5.565, 11.896] [5.315, 11.660] [5.138, 11.517]

Synthetic data

YouTube video data

NELA-GT news data

Competition

Four baselines (ablations)

One competitor  
(Fabbri et al., “MMS”) for QREM

Implementation

Python 3.10



# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5 10 20	40 415	202 075 404 150 808 300	[0.900, 8.475] [0.938, 8.701] [0.989, 9.444]
YT-10k (3 · 6 graphs)	5 10 20	150 572	752 860 1 505 720 3 011 440	[0.806, 5.785] [0.883, 7.576] [0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5 10 20	11 931	59 655 119 310 238 620	[4.217, 9.533] [4.248, 9.567] [4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5 10 20	57 447	287 235 574 470 1 148 940	[4.609, 11.068] [4.392, 10.769] [4.329, 10.741]
NF-ALL (3 · 6 graphs)	5 10 20	93 455	467 275 934 550 1 869 100	[5.565, 11.896] [5.315, 11.660] [5.138, 11.517]

Synthetic data

YouTube video data

NELA-GT news data

Competition

Four baselines (ablations)

One competitor  
(Fabbri et al., “MMS”) for QREM

Implementation

Python 3.10

Commodity hardware

# Experimental Evaluation

Dataset	$d$	$n$	$m$	$f(G)/n$
SU, SH (2 · 4 · 36 graphs)	5	$10^i$ for $i \in \{2, 3, 4, 5\}$	$5 \times 10^i$	[1.291, 15.231]
YT-100k (3 · 6 graphs)	5		202 075	[0.900, 8.475]
	10	40 415	404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10k (3 · 6 graphs)	5		752 860	[0.806, 5.785]
	10	150 572	1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5		59 655	[4.217, 9.533]
	10	11 931	119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5		287 235	[4.609, 11.068]
	10	57 447	574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5		467 275	[5.565, 11.896]
	10	93 455	934 550	[5.315, 11.660]
	20		1 869 100	[5.138, 11.517]

Synthetic data

YouTube video data

NELA-GT news data

Competition

Four baselines (ablations)

One competitor  
(Fabbri et al., “MMS”) for QREM

Implementation

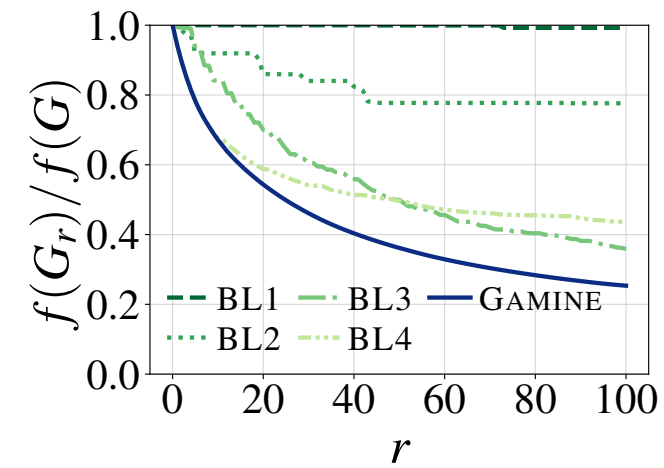
Python 3.10

Commodity hardware

Basic parallelization

# GAMINE...

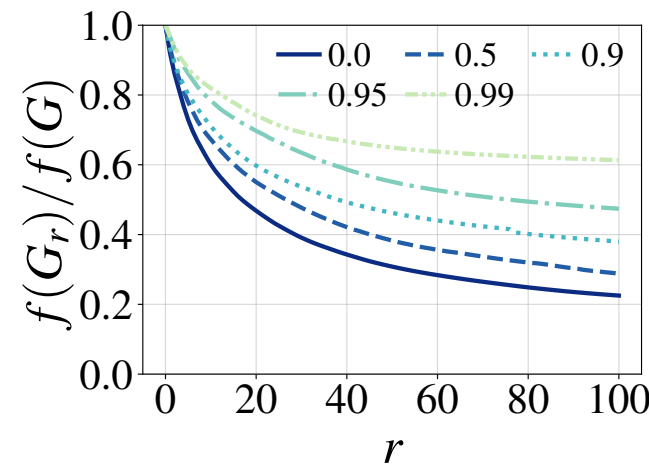
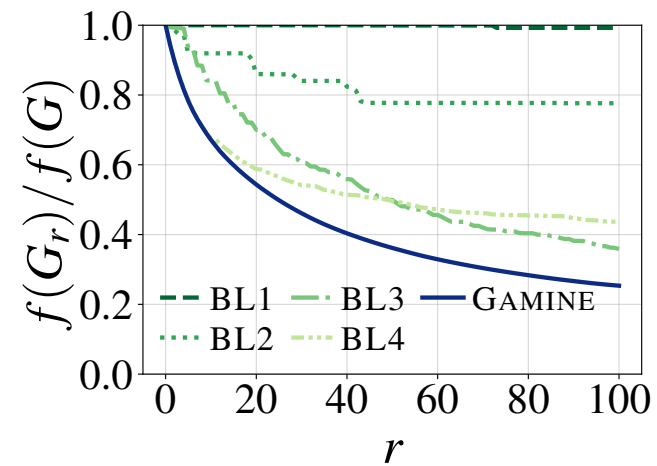
# GAMINE...



...outperforms all *baselines*

# GAMINE...

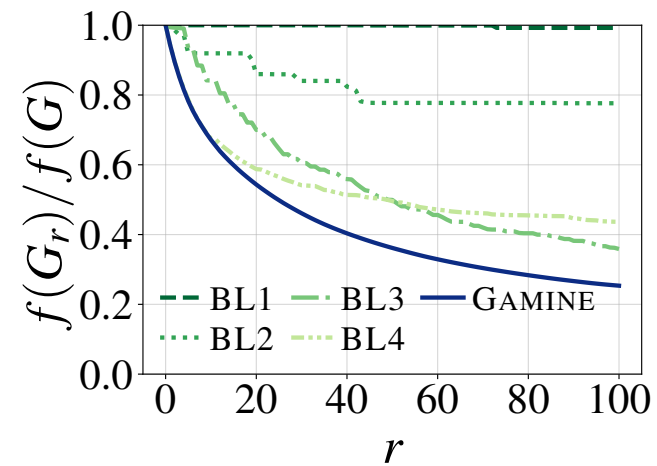
...is robust to changes in the *quality threshold*



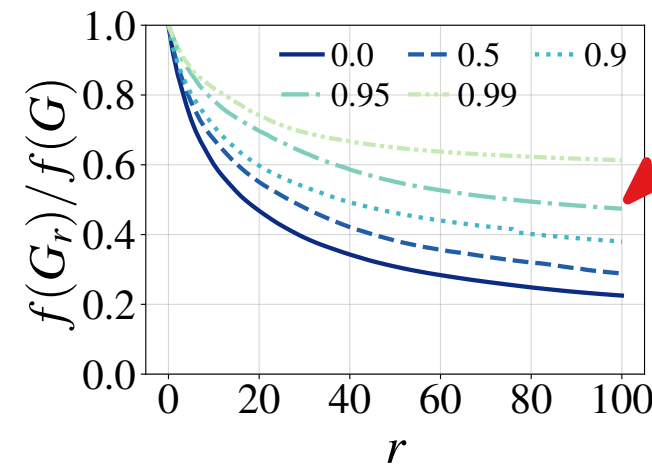
...outperforms all *baselines*

# GAMINE...

...is robust to changes in the *quality threshold*



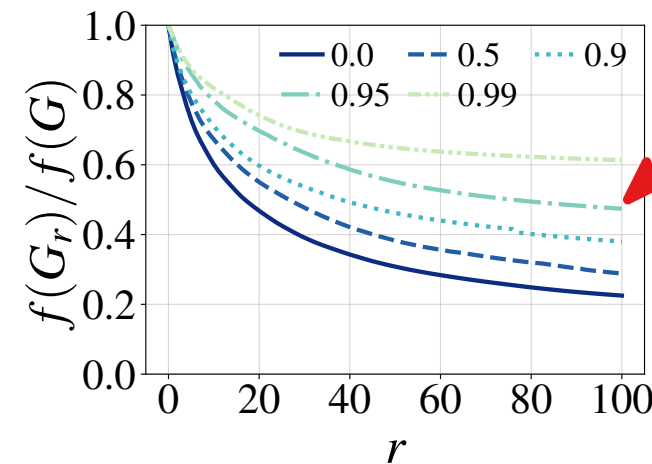
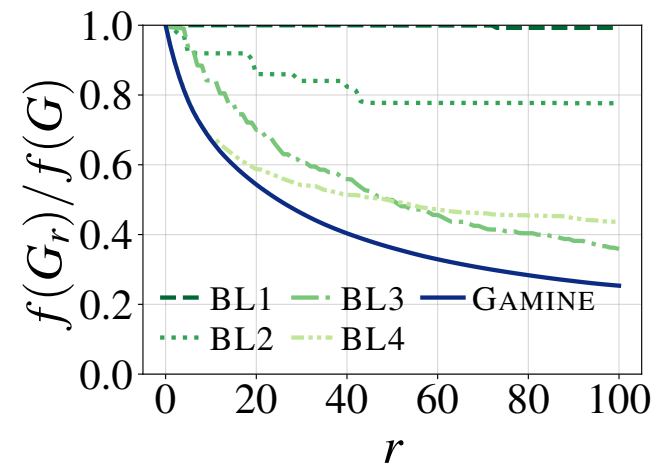
...outperforms all *baselines*



50% exposure reduction  
5% drop in recommendation quality

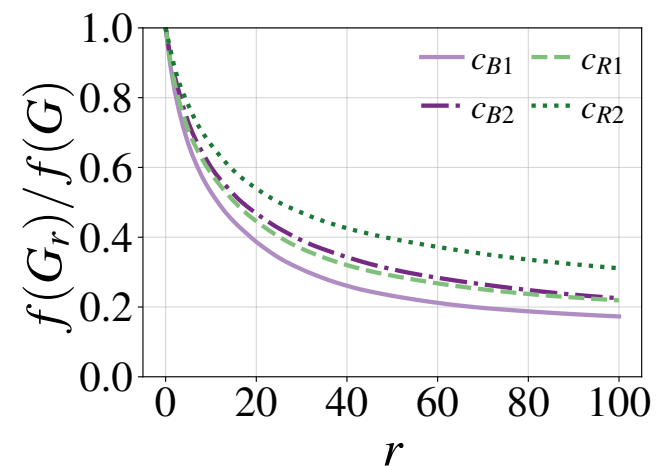
# GAMINE...

...is robust to changes in the *quality threshold*



50% exposure reduction  
5% drop in recommendation quality

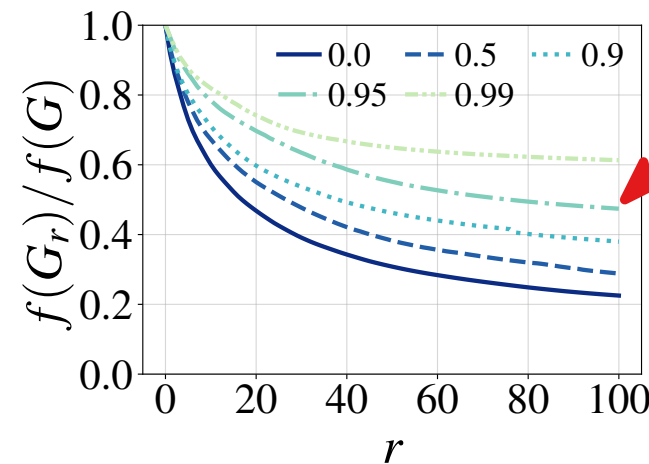
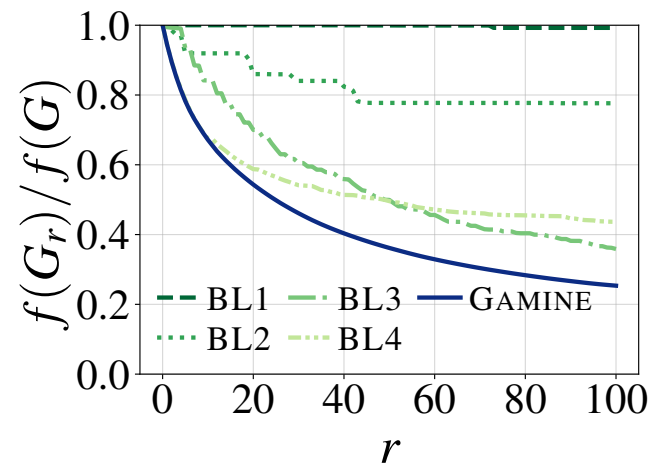
...outperforms all *baselines*



...is robust to changes in the *cost function*

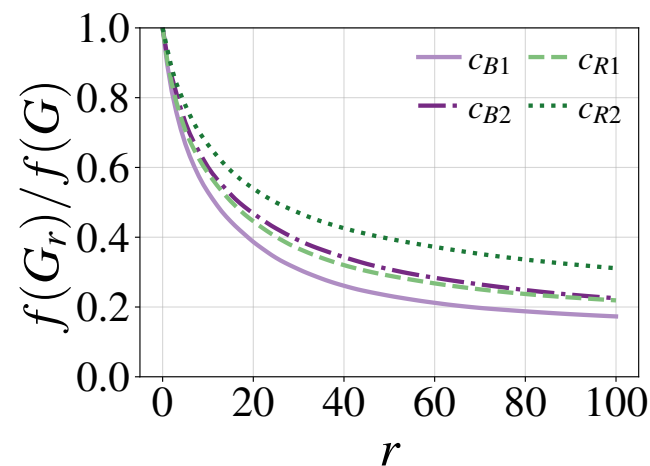
# GAMINE...

...is robust to changes in the *quality threshold*

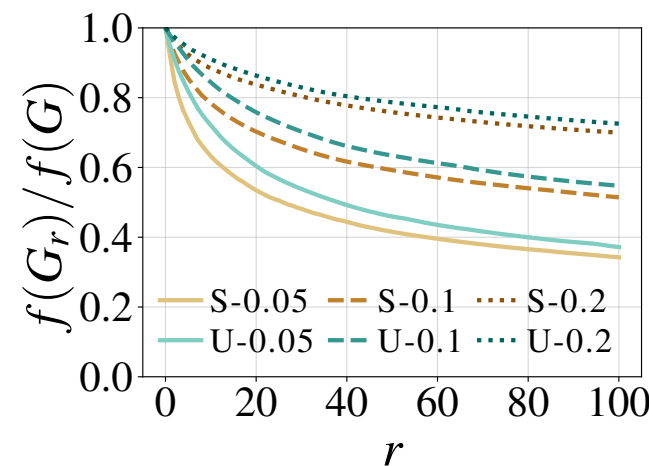


50% exposure reduction  
5% drop in recommendation quality

...outperforms all *baselines*



...is robust to changes in the *user model*

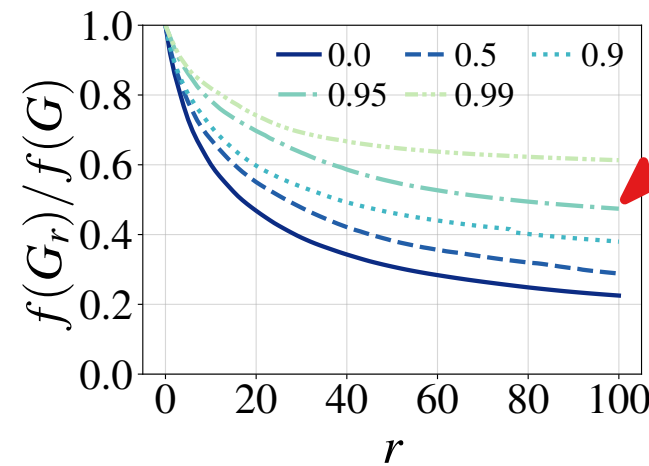
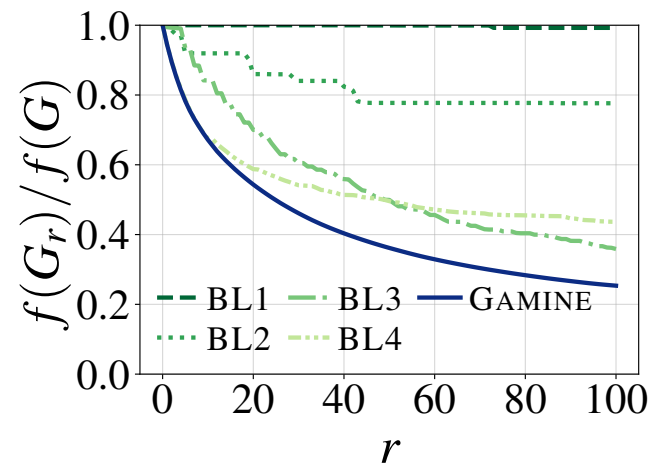


...is robust to changes in the *cost function*



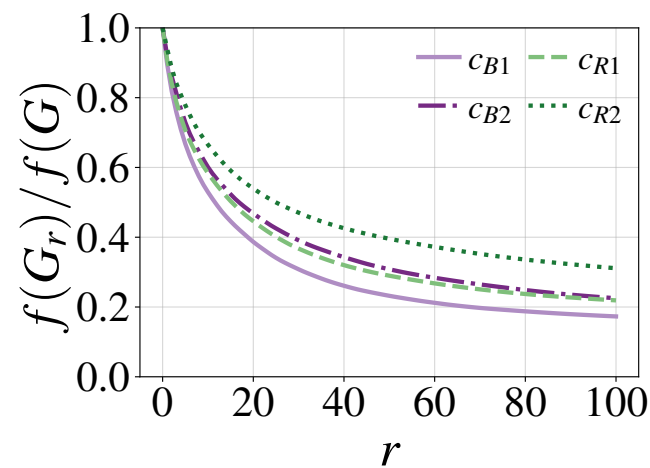
# GAMINE...

...is robust to changes in the *quality threshold*

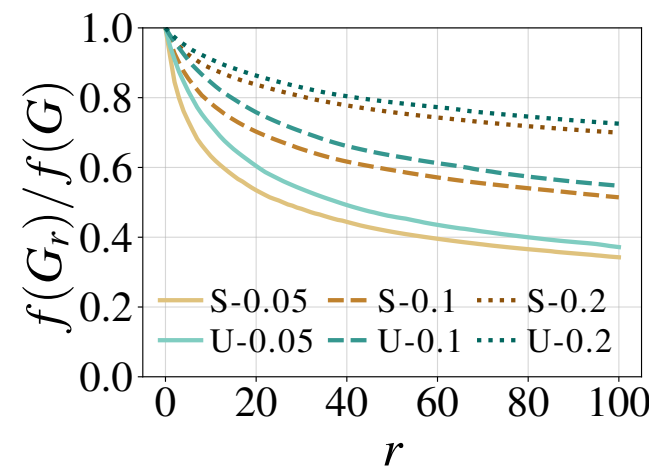


50% exposure reduction  
5% drop in recommendation quality

...outperforms all *baselines*

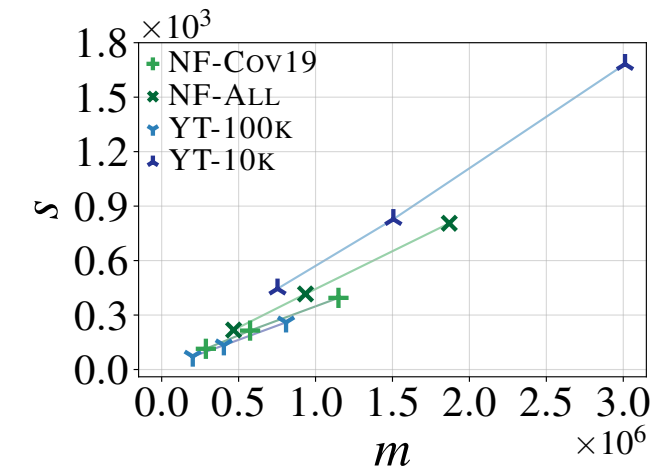


...is robust to changes in the *user model*

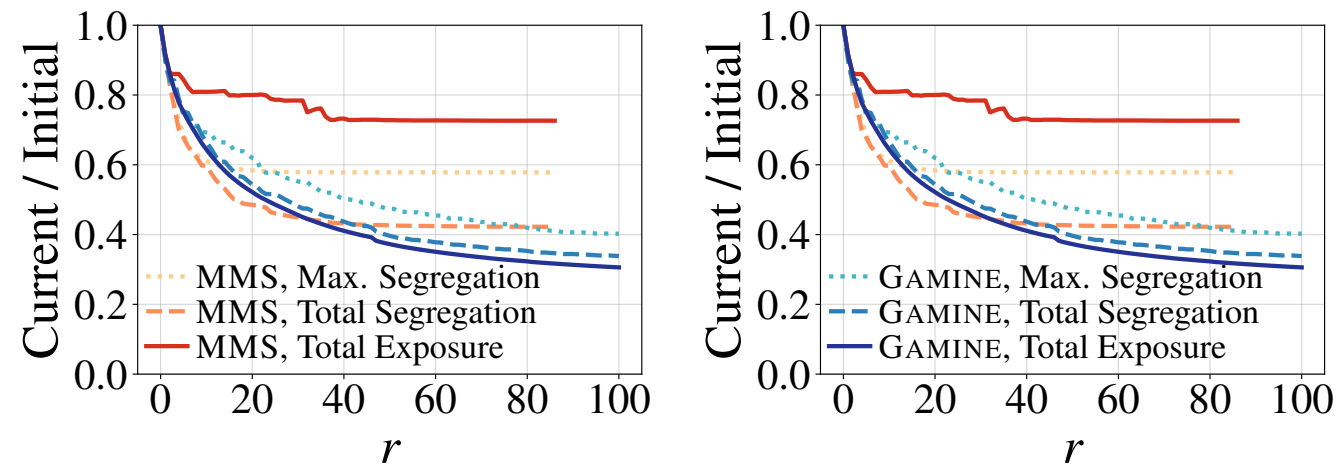


...is robust to changes in the *cost function*

...and scales linearly

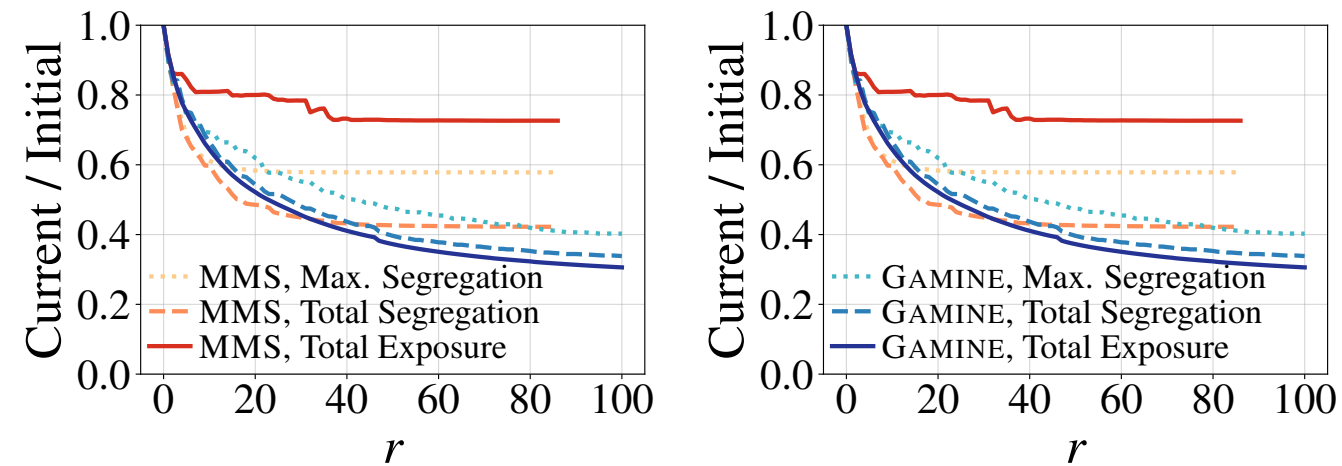


# GAMINE...

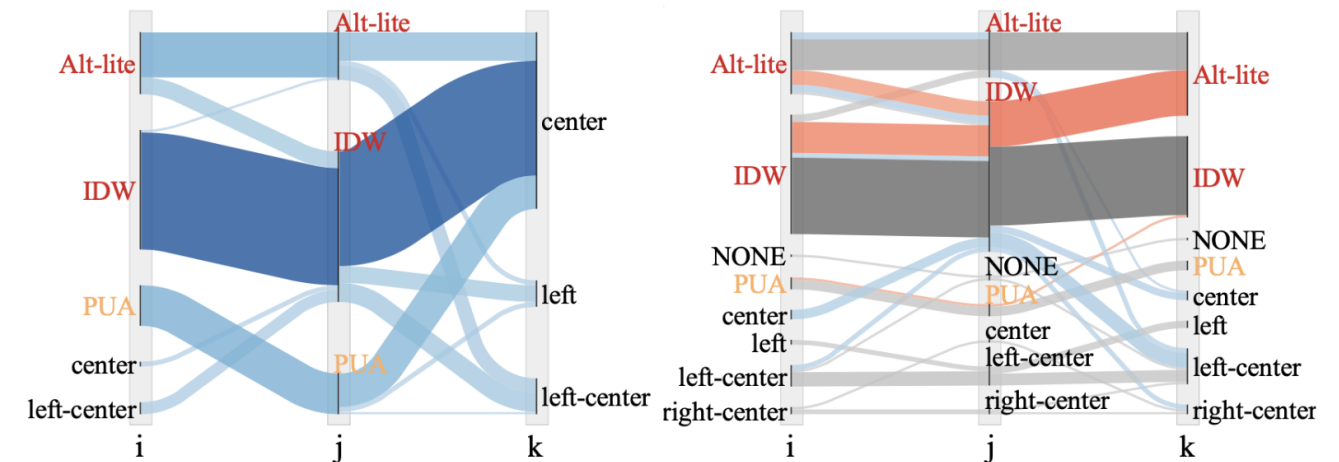


...outperforms MMS on the video data and...

# GAMINE...



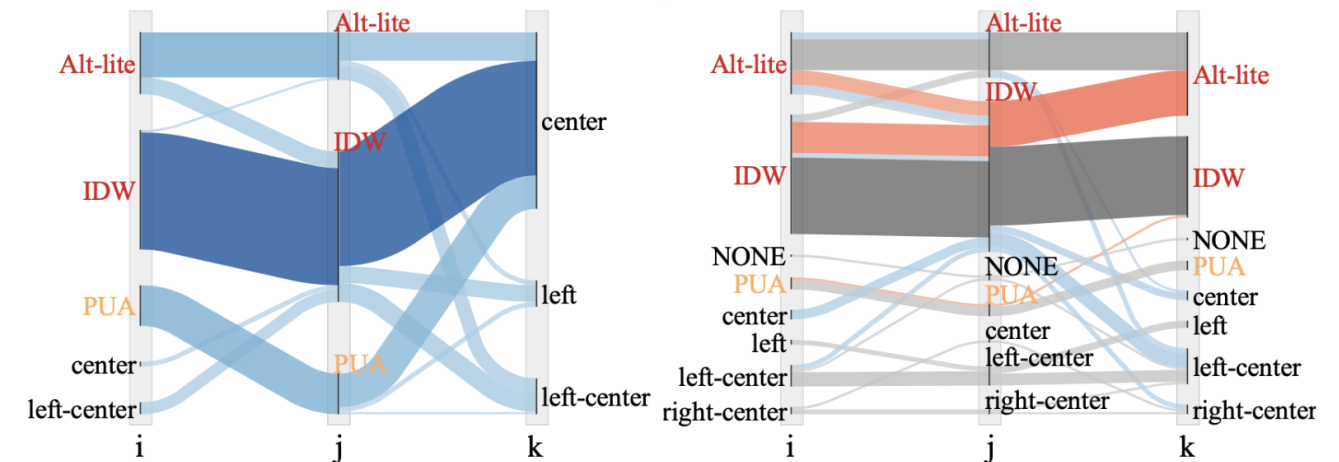
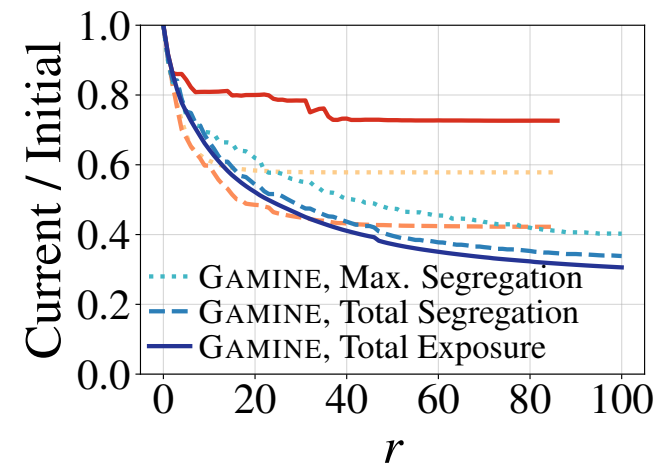
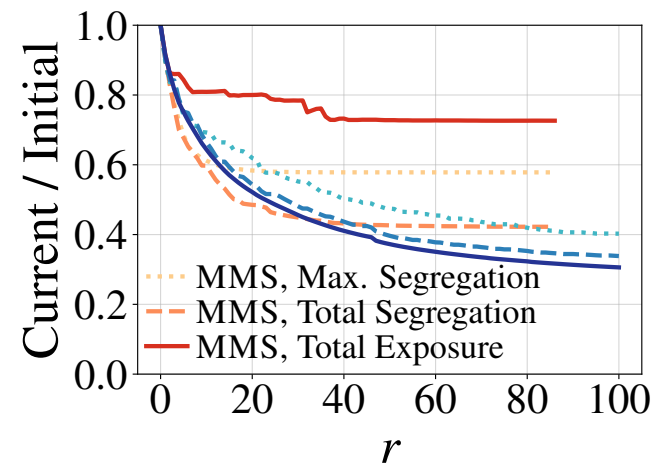
...generates *interpretable* rewirings, but...



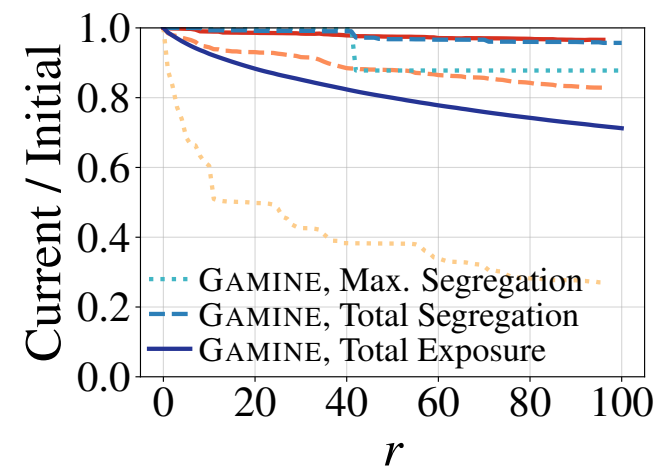
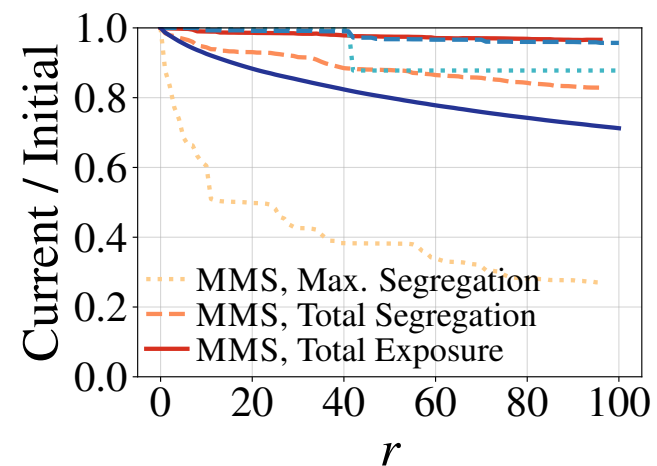
...outperforms MMS on the video data and...

# GAMINE...

...generates *interpretable* rewirings, but...



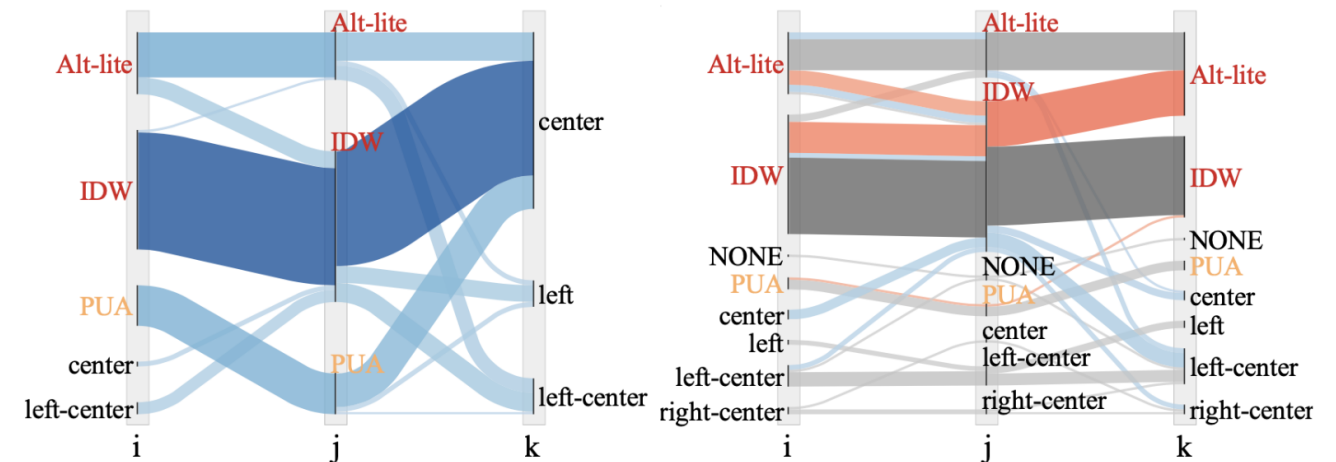
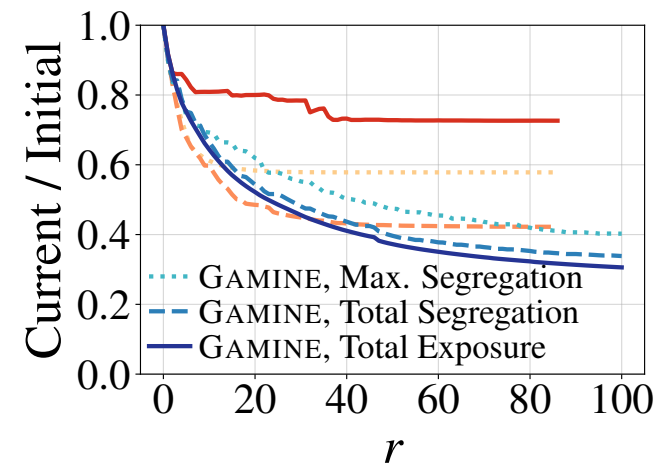
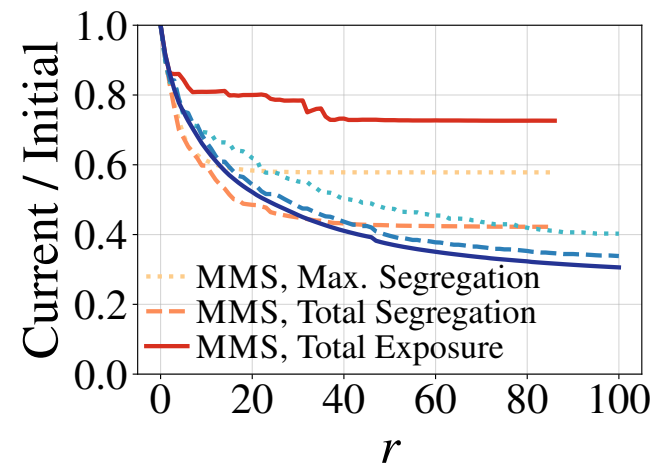
...outperforms MMS on the video data and...



...it doesn't do nearly as well on the news data...

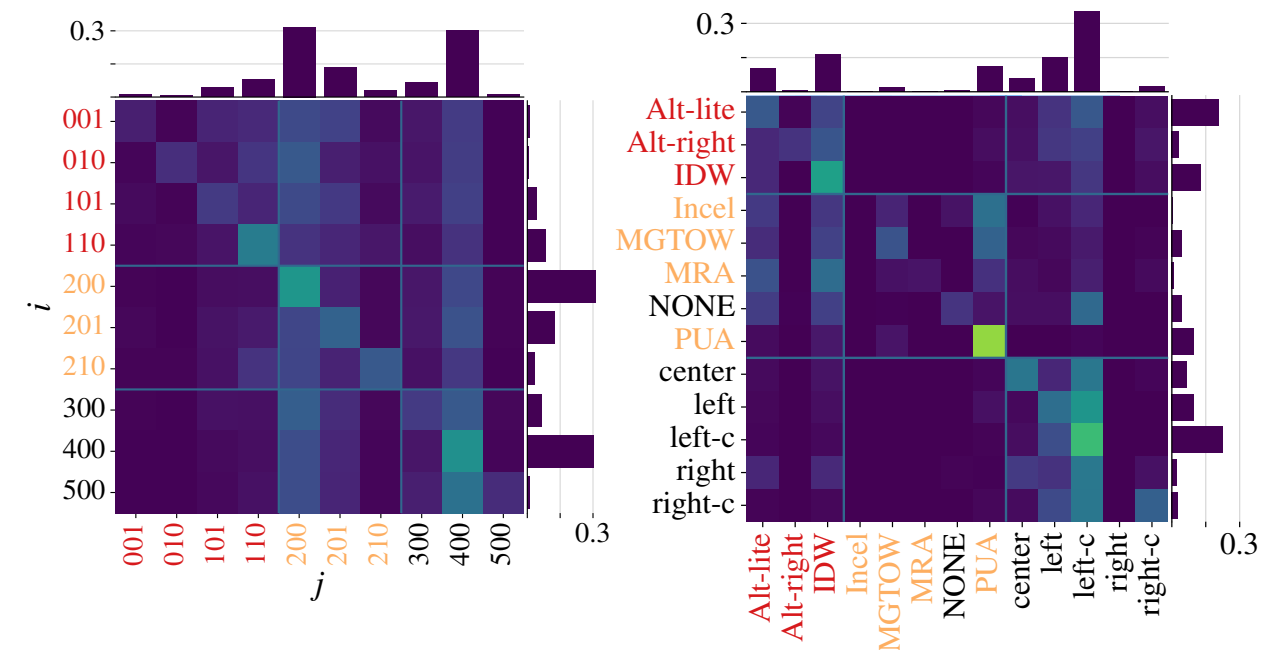
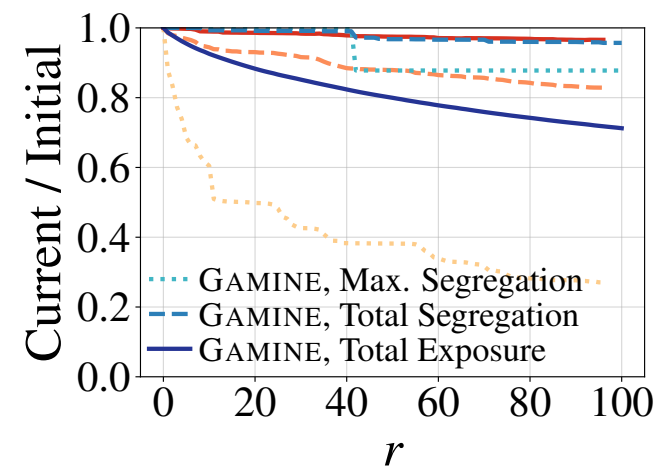
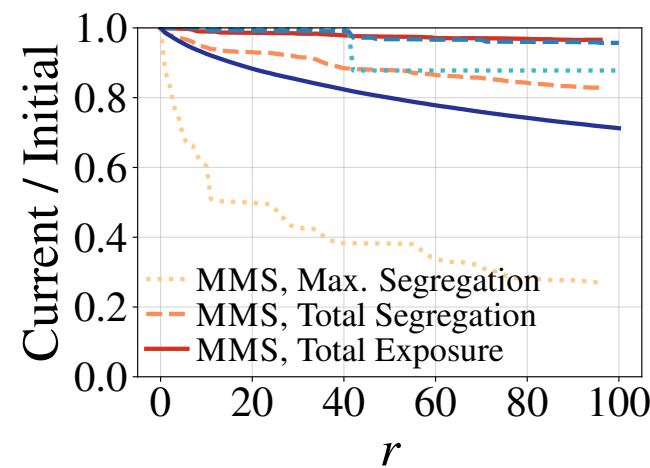
# GAMINE...

...generates *interpretable* rewirings, but...



...outperforms MMS on the video data and...

...because the news data is more *complex*!



...it doesn't do nearly as well on the news data...

# Summary

# Summary

Real-world problem + modeling assumptions  $\rightarrow$   $r$ -rewiring exposure minimization (REM)

# Summary

Real-world problem + modeling assumptions  $\rightarrow$   $r$ -rewiring exposure minimization (REM)  
= Optimizing random walks via graph editing under budget constraints



# Summary

Real-world problem + modeling assumptions  $\rightarrow$   $r$ -rewiring exposure minimization (REM)  
= Optimizing random walks via graph editing under budget constraints

NP-Hardness by reduction from Minimum Vertex Cover

# Summary

Real-world problem + modeling assumptions  $\rightarrow$   $r$ -rewiring exposure minimization (REM)  
= Optimizing random walks via graph editing under budget constraints

NP-Hardness by reduction from Minimum Vertex Cover  
 $(1 - \frac{1}{e})$ -Approximability under conditional submodularity

# Summary

Real-world problem + modeling assumptions  $\rightarrow$   $r$ -rewiring exposure minimization (REM)  
= Optimizing random walks via graph editing under budget constraints

NP-Hardness by reduction from Minimum Vertex Cover  
 $(1 - \frac{1}{e})$ -Approximability under conditional submodularity

Efficient GAMINE algorithm ( $O(n)$  under realistic assumptions)

# Summary

Real-world problem + modeling assumptions  $\rightarrow$   $r$ -rewiring exposure minimization (REM)  
= Optimizing random walks via graph editing under budget constraints

NP-Hardness by reduction from Minimum Vertex Cover  
 $(1 - \frac{1}{e})$ -Approximability under conditional submodularity

Efficient GAMINE algorithm ( $O(n)$  under realistic assumptions)

Extensive evaluation on synthetic and real-world data

# Summary

Real-world problem + modeling assumptions  $\rightarrow$   $r$ -rewiring exposure minimization (REM)  
= Optimizing random walks via graph editing under budget constraints

NP-Hardness by reduction from Minimum Vertex Cover  
 $(1 - \frac{1}{e})$ -Approximability under conditional submodularity

Efficient GAMINE algorithm ( $O(n)$  under realistic assumptions)

Extensive evaluation on synthetic and real-world data  
Effective, efficient, and robust in practice

# Summary

Real-world problem + modeling assumptions  $\rightarrow$   $r$ -rewiring exposure minimization (REM)  
= Optimizing random walks via graph editing under budget constraints

NP-Hardness by reduction from Minimum Vertex Cover  
 $(1 - \frac{1}{e})$ -Approximability under conditional submodularity

Efficient GAMINE algorithm ( $O(n)$  under realistic assumptions)

Extensive evaluation on synthetic and real-world data  
Effective, efficient, and robust in practice

Favorite future work: *Data complexity*

# Thank you!



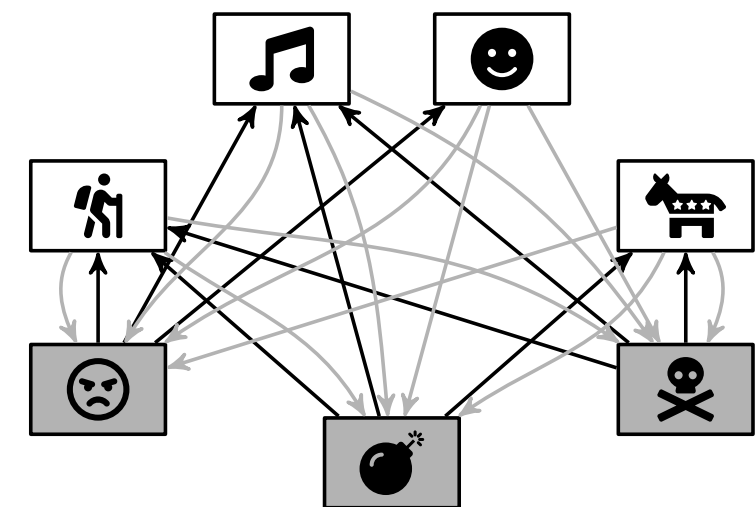
Real-world problem + modeling assumptions  $\rightarrow$   $r$ -rewiring exposure minimization (REM)  
= Optimizing random walks via graph editing under budget constraints

NP-Hardness by reduction from Minimum Vertex Cover  
 $(1 - \frac{1}{e})$ -Approximability under conditional submodularity

Efficient GAMINE algorithm ( $O(n)$  under realistic assumptions)

Extensive evaluation on synthetic and real-world data  
Effective, efficient, and robust in practice

Favorite future work: *Data complexity*



# Conditional Submodularity



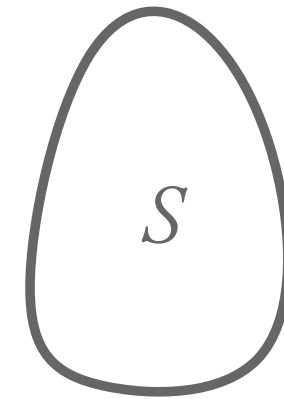
# Conditional Submodularity

Partition  $V$  into two classes

# Conditional Submodularity

Partition  $V$  into two classes

$$\text{Safe nodes } S = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} = 0\}$$

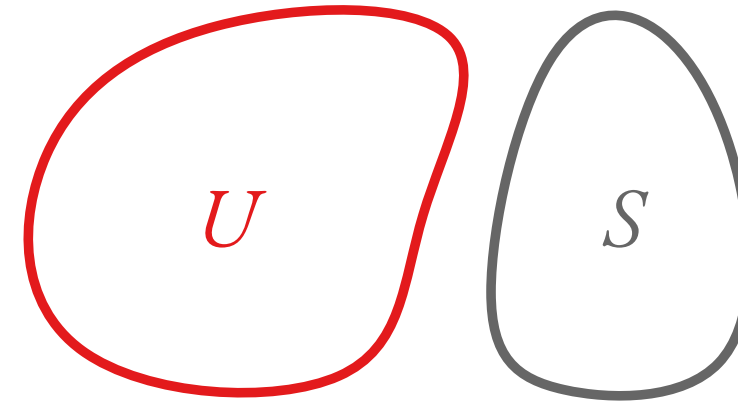


# Conditional Submodularity

Partition  $V$  into two classes

$$\text{Safe nodes } S = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} = 0\}$$

$$\text{Unsafe nodes } U = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} > 0\}$$



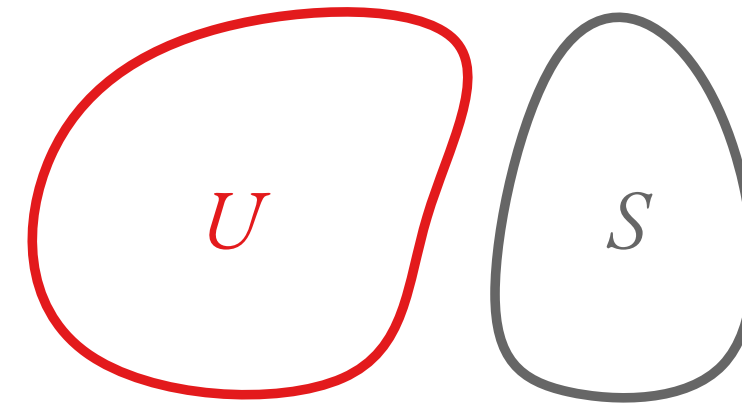
# Conditional Submodularity

Partition  $V$  into two classes

$$\text{Safe nodes } S = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} = 0\}$$

$$\text{Unsafe nodes } U = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} > 0\}$$

Partition  $E$  into four classes based on  $S$  and  $U$



# Conditional Submodularity

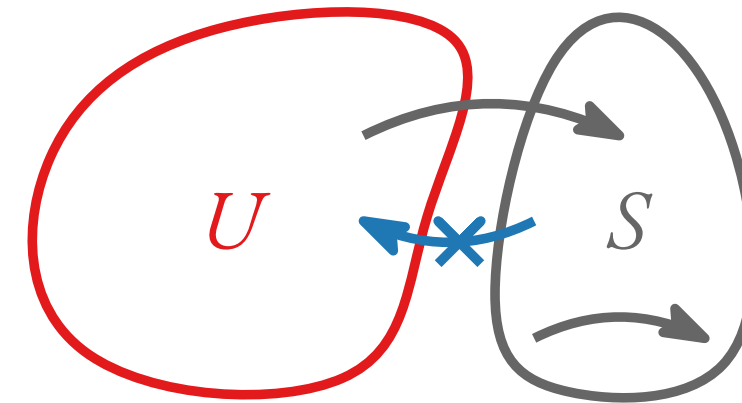
Partition  $V$  into two classes

$$\text{Safe nodes } S = \{i \in V \mid \mathbf{e}_i^T \mathbf{F}\mathbf{c} = 0\}$$

$$\text{Unsafe nodes } U = \{i \in V \mid \mathbf{e}_i^T \mathbf{F}\mathbf{c} > 0\}$$

Partition  $E$  into four classes based on  $S$  and  $U$

$$E_{AB} = \{(i, j) \in E \mid i \in A \wedge j \in B\}$$



# Conditional Submodularity

Partition  $V$  into two classes

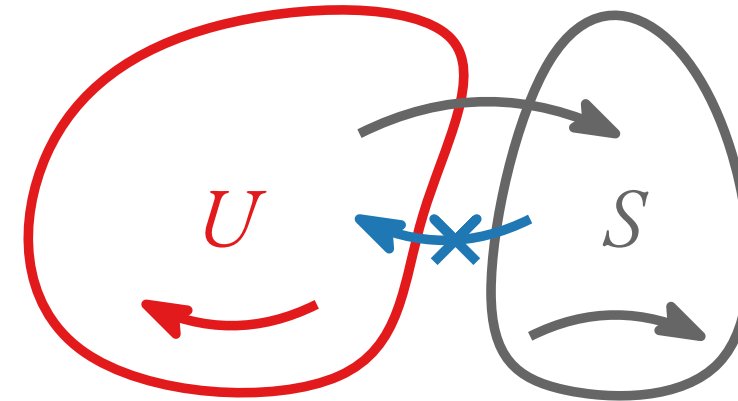
$$\text{Safe nodes } S = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} = 0\}$$

$$\text{Unsafe nodes } U = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} > 0\}$$

Partition  $E$  into four classes based on  $S$  and  $U$

$$E_{AB} = \{(i, j) \in E \mid i \in A \wedge j \in B\}$$

$$\text{Only problematic class: } E_{UU} = \{(i, j) \mid i \in U \wedge j \in U\}$$



# Conditional Submodularity

Partition  $V$  into two classes

$$\text{Safe nodes } S = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} = 0\}$$

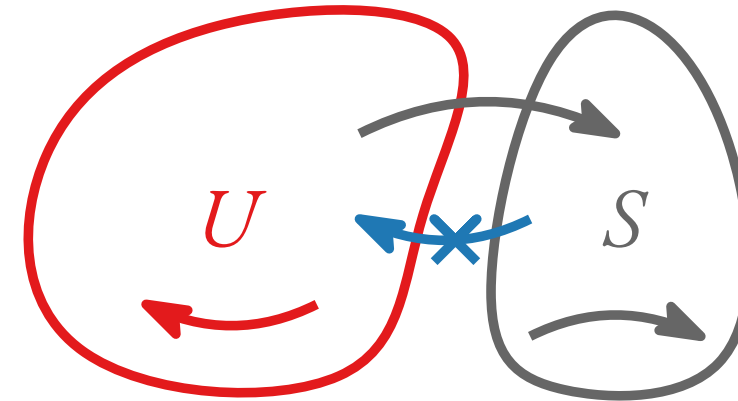
$$\text{Unsafe nodes } U = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} > 0\}$$

Partition  $E$  into four classes based on  $S$  and  $U$

$$E_{AB} = \{(i, j) \in E \mid i \in A \wedge j \in B\}$$

$$\text{Only problematic class: } E_{UU} = \{(i, j) \mid i \in U \wedge j \in U\}$$

Define  $\Lambda^+ = \max\{\delta^+(i) \mid i \in U\}$

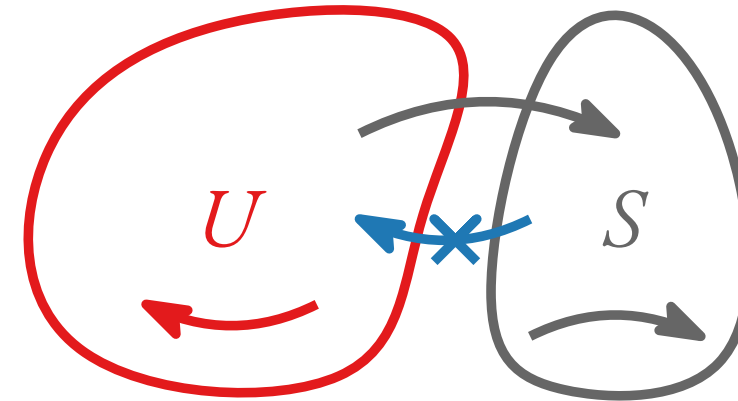


# Conditional Submodularity

Partition  $V$  into two classes

$$\text{Safe nodes } S = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} = 0\}$$

$$\text{Unsafe nodes } U = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} > 0\}$$



Partition  $E$  into four classes based on  $S$  and  $U$

$$E_{AB} = \{(i, j) \in E \mid i \in A \wedge j \in B\}$$

$$\text{Only problematic class: } E_{UU} = \{(i, j) \mid i \in U \wedge j \in U\}$$

Define  $\Lambda^+ = \max\{\delta^+(i) \mid i \in U\}$

If  $|S| \geq \Lambda^+$ , then  $\max f_{\Delta}(G, G_r)$  is equivalent to maximizing submodular function over  $E_{UU}$

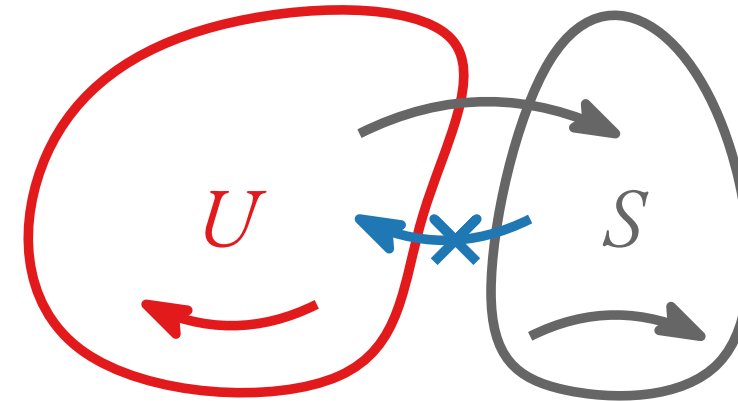


# Conditional Submodularity

Partition  $V$  into two classes

$$\text{Safe nodes } S = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} = 0\}$$

$$\text{Unsafe nodes } U = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} > 0\}$$



Partition  $E$  into four classes based on  $S$  and  $U$

$$E_{AB} = \{(i, j) \in E \mid i \in A \wedge j \in B\}$$

$$\text{Only problematic class: } E_{UU} = \{(i, j) \mid i \in U \wedge j \in U\}$$

Define  $\Lambda^+ = \max\{\delta^+(i) \mid i \in U\}$

If  $|S| \geq \Lambda^+$ , then  $\max f_{\Delta}(G, G_r)$  is equivalent to maximizing submodular function over  $E_{UU}$

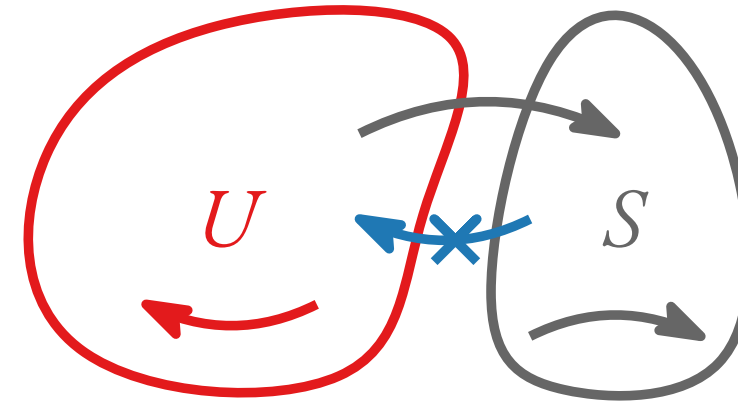
“ $G$  has the safety property”

# Conditional Submodularity

Partition  $V$  into two classes

$$\text{Safe nodes } S = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} = 0\}$$

$$\text{Unsafe nodes } U = \{i \in V \mid \mathbf{e}_i^T \mathbf{F} \mathbf{c} > 0\}$$



Partition  $E$  into four classes based on  $S$  and  $U$

$$E_{AB} = \{(i, j) \in E \mid i \in A \wedge j \in B\}$$

$$\text{Only problematic class: } E_{UU} = \{(i, j) \mid i \in U \wedge j \in U\}$$

Define  $\Lambda^+ = \max\{\delta^+(i) \mid i \in U\}$

If  $|S| \geq \Lambda^+$ , then  $\max f_{\Delta}(G, G_r)$  is equivalent to maximizing submodular function over  $E_{UU}$

“ $G$  has the safety property”

Intuition: Pick edges from  $E_{UU}$  to rewire into  $S$

# Greedy Rewiring

# Greedy Rewiring

Greedy step  $\leftrightarrow$  Rank-one update of transition matrix  $\mathbf{P}$

# Greedy Rewiring

Greedy step  $\leftrightarrow$  Rank-one update of transition matrix  $\mathbf{P}$

$$\mathbf{u} = \mathbf{P}_{ij} \mathbf{e}_i$$

# Greedy Rewiring

Greedy step  $\leftrightarrow$  Rank-one update of transition matrix  $\mathbf{P}$

$$\mathbf{u} = \mathbf{P}_{ij}\mathbf{e}_i, \quad \mathbf{v} = \mathbf{e}_j - \mathbf{e}_k$$

# Greedy Rewiring

Greedy step  $\leftrightarrow$  Rank-one update of transition matrix  $\mathbf{P}$

$$\mathbf{u} = \mathbf{P}_{ij}\mathbf{e}_i, \quad \mathbf{v} = \mathbf{e}_j - \mathbf{e}_k$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{u}(-\mathbf{v})^T$$

# Greedy Rewiring

Greedy step  $\leftrightarrow$  Rank-one update of transition matrix  $\mathbf{P}$

$$\mathbf{u} = \mathbf{P}_{ij}\mathbf{e}_i, \quad \mathbf{v} = \mathbf{e}_j - \mathbf{e}_k$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{u}(-\mathbf{v})^T$$

$$\mathbf{F}' = (\mathbf{I} - \mathbf{P} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{F} - \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1+\mathbf{v}^T\mathbf{F}\mathbf{u}}$$

[Sherman-Morrison-Formula]



# Greedy Rewiring

Greedy step  $\leftrightarrow$  Rank-one update of transition matrix  $\mathbf{P}$

$$\mathbf{u} = \mathbf{P}_{ij}\mathbf{e}_i, \quad \mathbf{v} = \mathbf{e}_j - \mathbf{e}_k$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{u}(-\mathbf{v})^T$$

$$\mathbf{F}' = (\mathbf{I} - \mathbf{P} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{F} - \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1 + \mathbf{v}^T\mathbf{F}\mathbf{u}}$$

[Sherman-Morrison-Formula]

$$f(G') = f(G) - \mathbf{1}^T \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1 + \mathbf{v}^T\mathbf{F}\mathbf{u}} \mathbf{c} = f(G) - \frac{\sigma\tau}{\rho}$$

# Greedy Rewiring

Greedy step  $\leftrightarrow$  Rank-one update of transition matrix  $\mathbf{P}$

$$\mathbf{u} = \mathbf{P}_{ij}\mathbf{e}_i, \quad \mathbf{v} = \mathbf{e}_j - \mathbf{e}_k$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{u}(-\mathbf{v})^T$$

$$\mathbf{F}' = (\mathbf{I} - \mathbf{P} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{F} - \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1 + \mathbf{v}^T\mathbf{F}\mathbf{u}}$$

[Sherman-Morrison-Formula]

$$f(G') = f(G) - \mathbf{1}^T \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1 + \mathbf{v}^T\mathbf{F}\mathbf{u}} \mathbf{c} = f(G) - \frac{\sigma\tau}{\rho}$$

$$\rho = 1 + \mathbf{v}^T\mathbf{F}\mathbf{u} > 0$$

[Normalization factor, always positive]

# Greedy Rewiring

Greedy step  $\leftrightarrow$  Rank-one update of transition matrix  $\mathbf{P}$

$$\mathbf{u} = \mathbf{P}_{ij}\mathbf{e}_i, \quad \mathbf{v} = \mathbf{e}_j - \mathbf{e}_k$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{u}(-\mathbf{v})^T$$

$$\mathbf{F}' = (\mathbf{I} - \mathbf{P} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{F} - \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1 + \mathbf{v}^T\mathbf{F}\mathbf{u}} \quad [\text{Sherman-Morrison-Formula}]$$

$$f(G') = f(G) - \mathbf{1}^T \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1 + \mathbf{v}^T\mathbf{F}\mathbf{u}} \mathbf{c} = f(G) - \frac{\sigma\tau}{\rho}$$

$$\rho = 1 + \mathbf{v}^T\mathbf{F}\mathbf{u} > 0 \quad [\text{Normalization factor, always positive}]$$

$$\sigma = \mathbf{1}^T\mathbf{F}\mathbf{u} > 0 \quad [\text{Expected number of visits to } i, \text{ scaled by } \mathbf{P}_{ij}, \text{ always positive}]$$

# Greedy Rewiring

Greedy step  $\leftrightarrow$  Rank-one update of transition matrix  $\mathbf{P}$

$$\mathbf{u} = \mathbf{P}_{ij}\mathbf{e}_i, \quad \mathbf{v} = \mathbf{e}_j - \mathbf{e}_k$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{u}(-\mathbf{v})^T$$

$$\mathbf{F}' = (\mathbf{I} - \mathbf{P} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{F} - \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1 + \mathbf{v}^T\mathbf{F}\mathbf{u}} \quad [\text{Sherman-Morrison-Formula}]$$

$$f(G') = f(G) - \mathbf{1}^T \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1 + \mathbf{v}^T\mathbf{F}\mathbf{u}} \mathbf{c} = f(G) - \frac{\sigma\tau}{\rho}$$

$$\rho = 1 + \mathbf{v}^T\mathbf{F}\mathbf{u} > 0 \quad [\text{Normalization factor, always positive}]$$

$$\sigma = \mathbf{1}^T\mathbf{F}\mathbf{u} > 0 \quad [\text{Expected number of visits to } i, \text{ scaled by } \mathbf{P}_{ij}, \text{ always positive}]$$

$$\tau = \mathbf{v}^T\mathbf{F}\mathbf{c} \quad [\text{Difference between exposure of } j \text{ and exposure of } k, \text{ can have any sign}]$$

# Greedy Rewiring

Greedy step  $\leftrightarrow$  Rank-one update of transition matrix  $\mathbf{P}$

$$\mathbf{u} = \mathbf{P}_{ij}\mathbf{e}_i, \quad \mathbf{v} = \mathbf{e}_j - \mathbf{e}_k$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{u}(-\mathbf{v})^T$$

$$\mathbf{F}' = (\mathbf{I} - \mathbf{P} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{F} - \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1 + \mathbf{v}^T\mathbf{F}\mathbf{u}} \quad [\text{Sherman-Morrison-Formula}]$$

$$f(G') = f(G) - \mathbf{1}^T \frac{\mathbf{F}\mathbf{u}\mathbf{v}^T\mathbf{F}}{1 + \mathbf{v}^T\mathbf{F}\mathbf{u}} \mathbf{c} = f(G) - \frac{\sigma\tau}{\rho}$$

$$\rho = 1 + \mathbf{v}^T\mathbf{F}\mathbf{u} > 0 \quad [\text{Normalization factor, always positive}]$$

$$\sigma = \mathbf{1}^T\mathbf{F}\mathbf{u} > 0 \quad [\text{Expected number of visits to } i, \text{ scaled by } \mathbf{P}_{ij}, \text{ always positive}]$$

$$\tau = \mathbf{v}^T\mathbf{F}\mathbf{c} \quad [\text{Difference between exposure of } j \text{ and exposure of } k, \text{ can have any sign}]$$

$\rightarrow$  A rewiring  $(i, j) \rightarrow (i, k)$  is greedily permissible iff  $\tau > 0 \leftrightarrow j$  is more exposed than  $k$

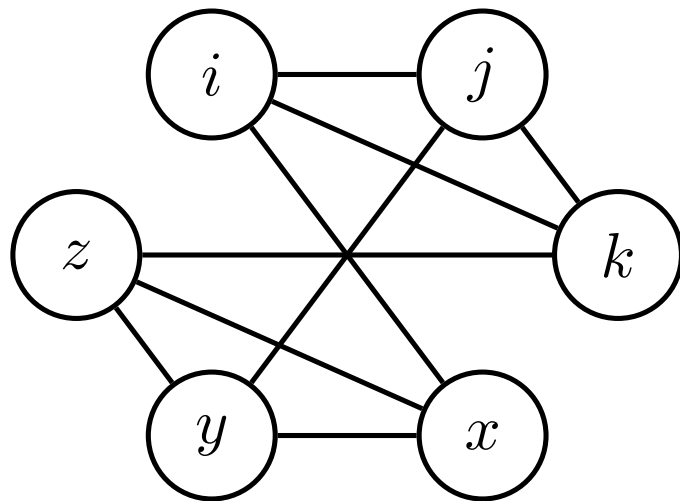
# NP-Hardness: Setup

# NP-Hardness: Setup

Reduction from Minimum Vertex Cover

# NP-Hardness: Setup

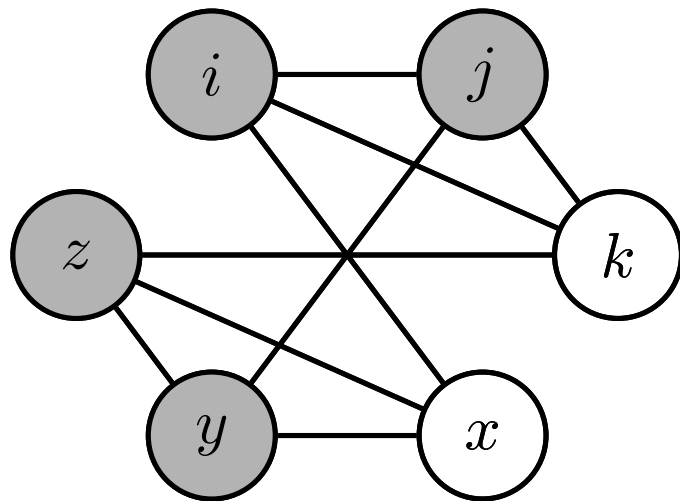
Reduction from Minimum Vertex Cover





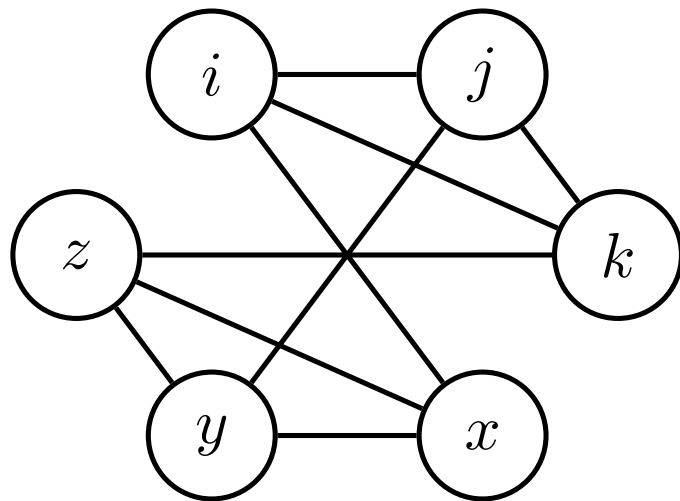
# NP-Hardness: Setup

Reduction from Minimum Vertex Cover



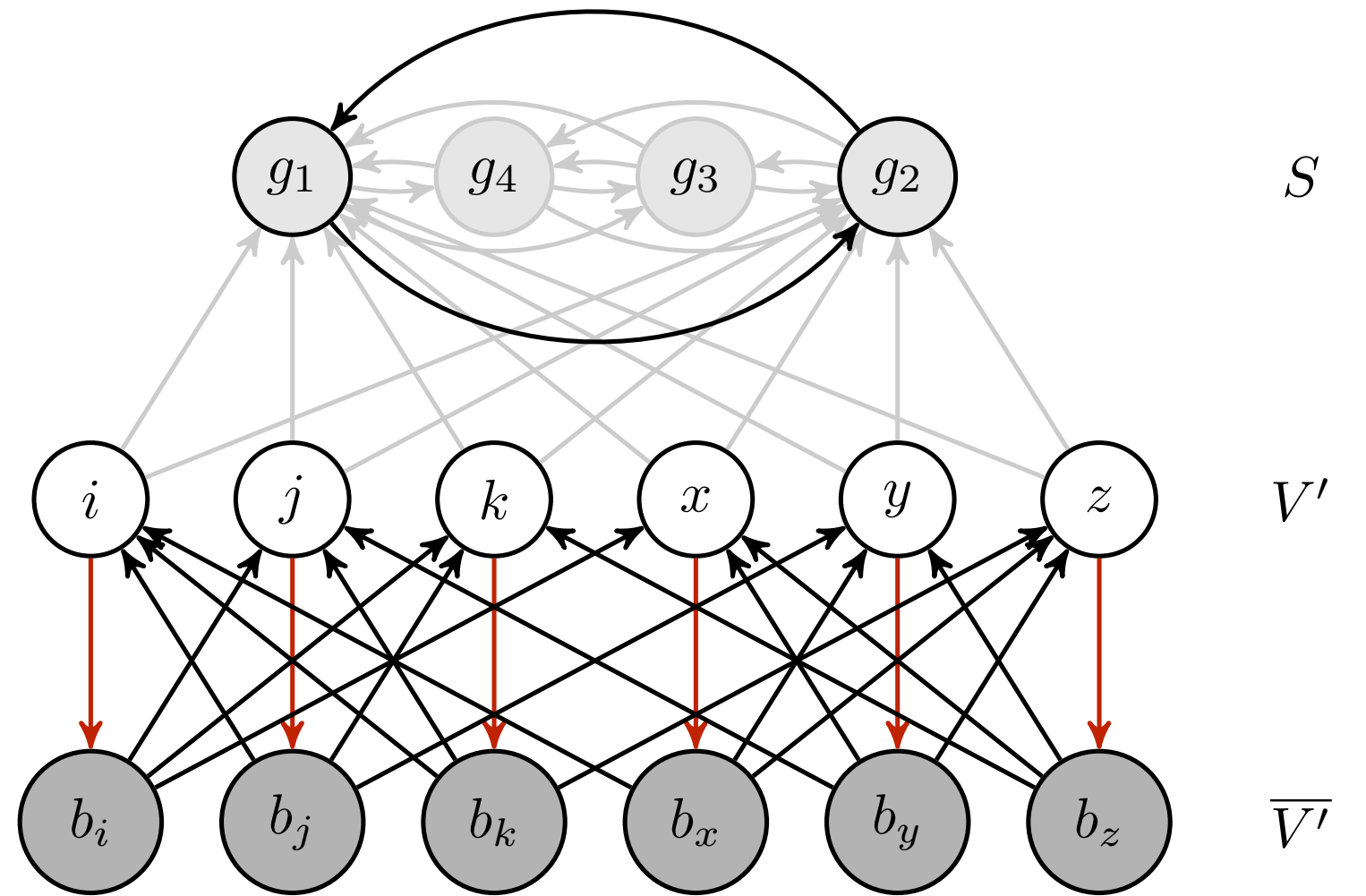
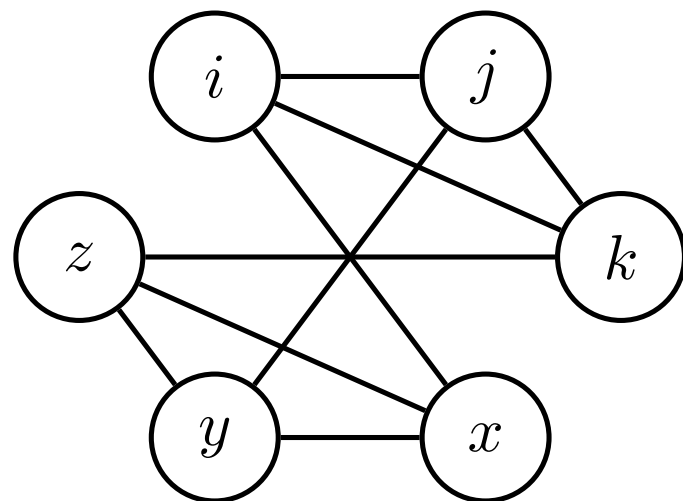
# NP-Hardness: Setup

Reduction from Minimum Vertex Cover



# NP-Hardness: Setup

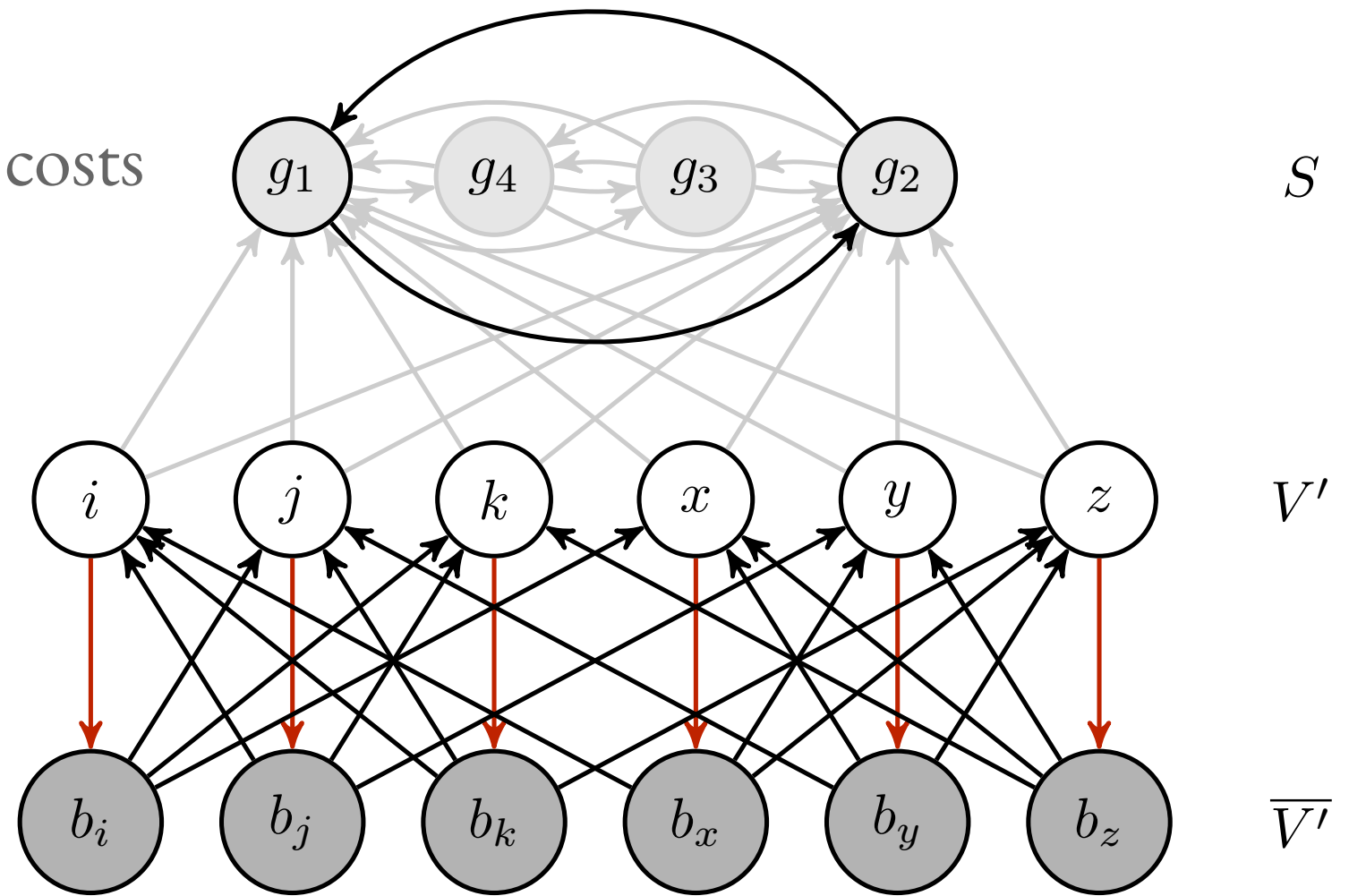
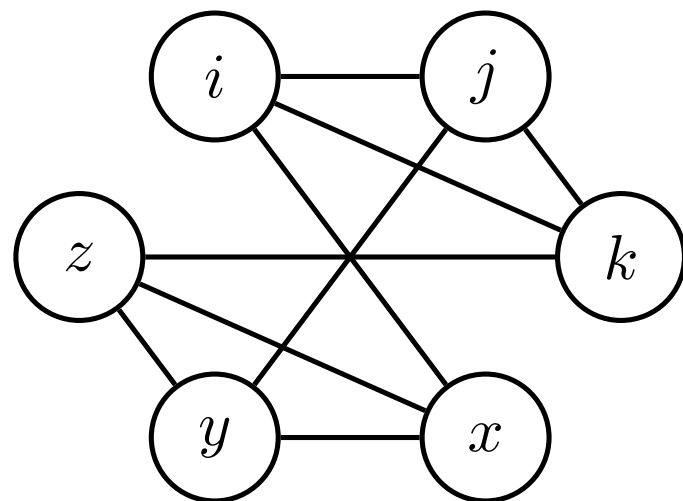
Reduction from Minimum Vertex Cover



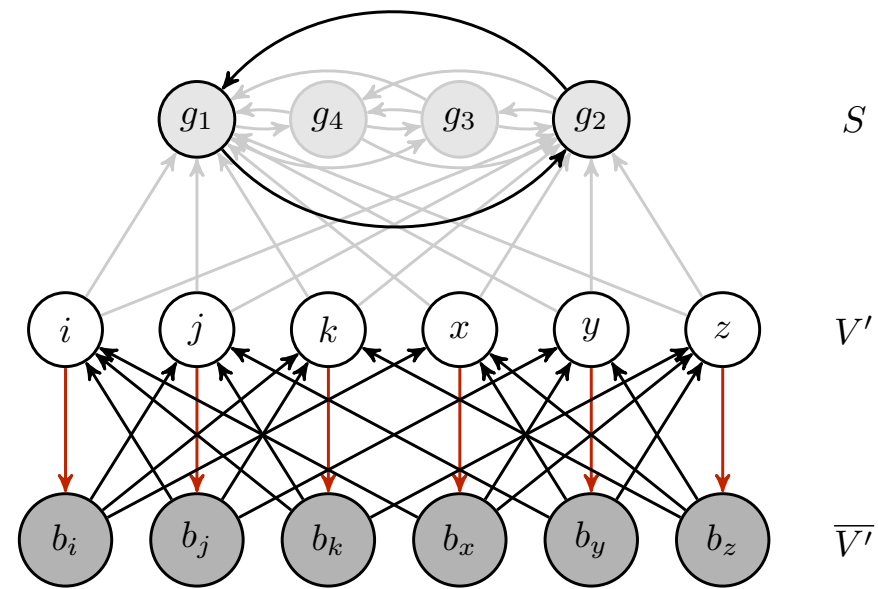
# NP-Hardness: Setup

Reduction from Minimum Vertex Cover

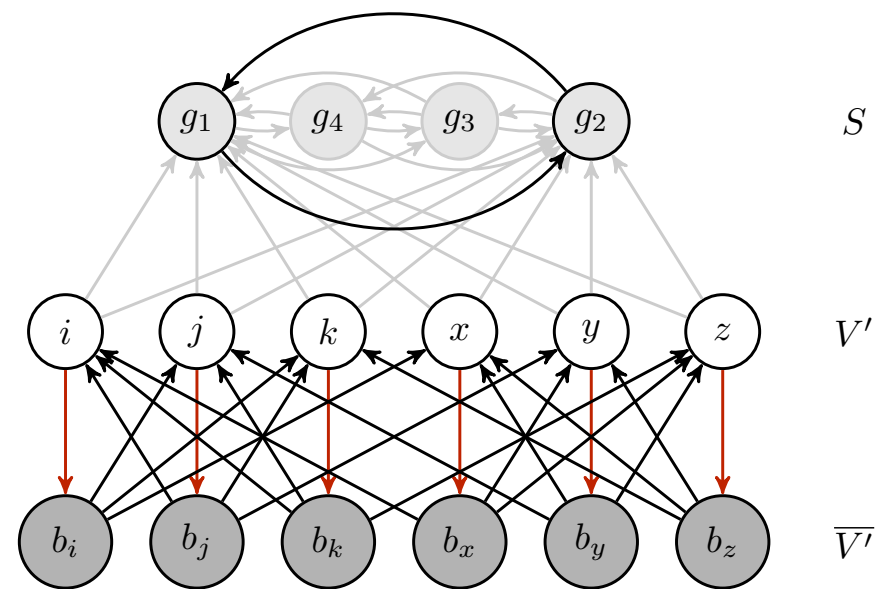
Even for 3-out-regular graphs and binary node costs



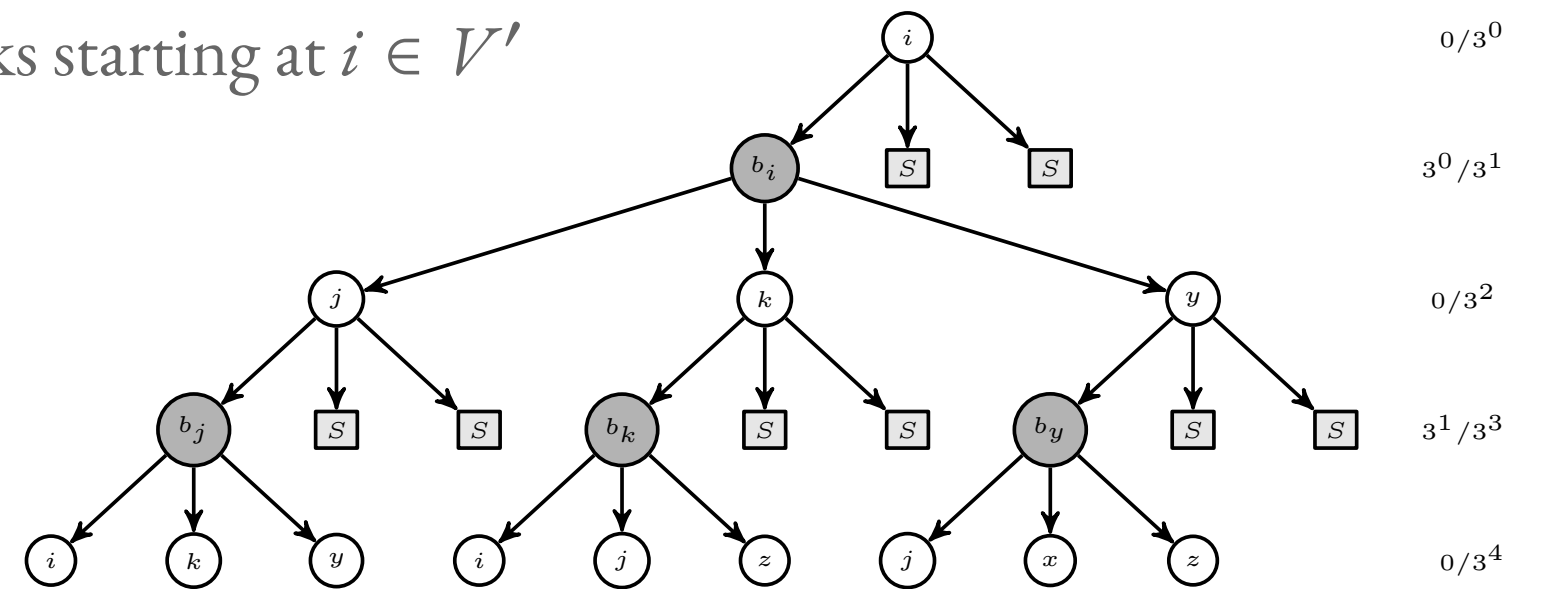
# NP-Hardness: Intuition



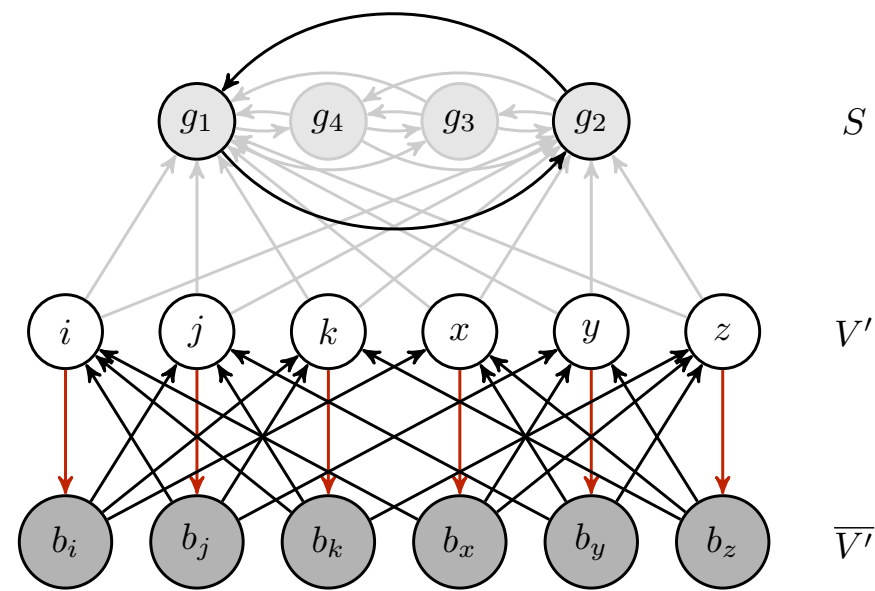
# NP-Hardness: Intuition



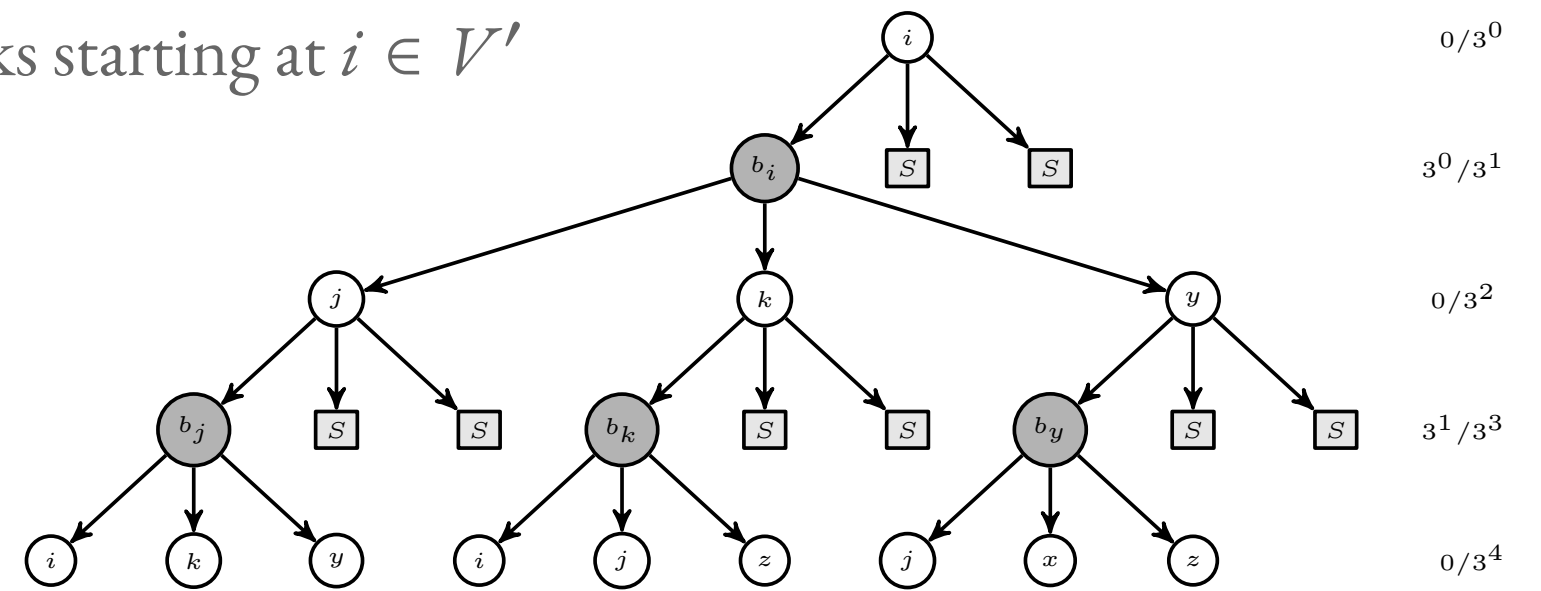
Walks starting at  $i \in V'$



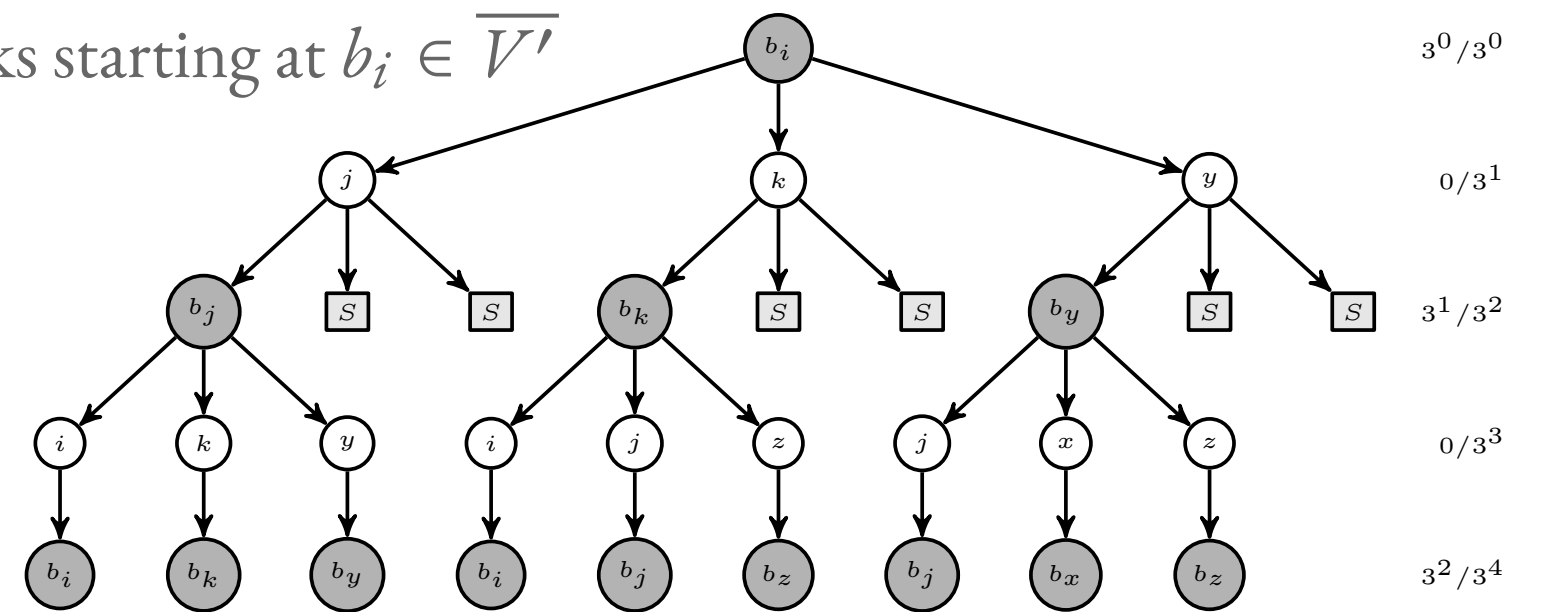
# NP-Hardness: Intuition



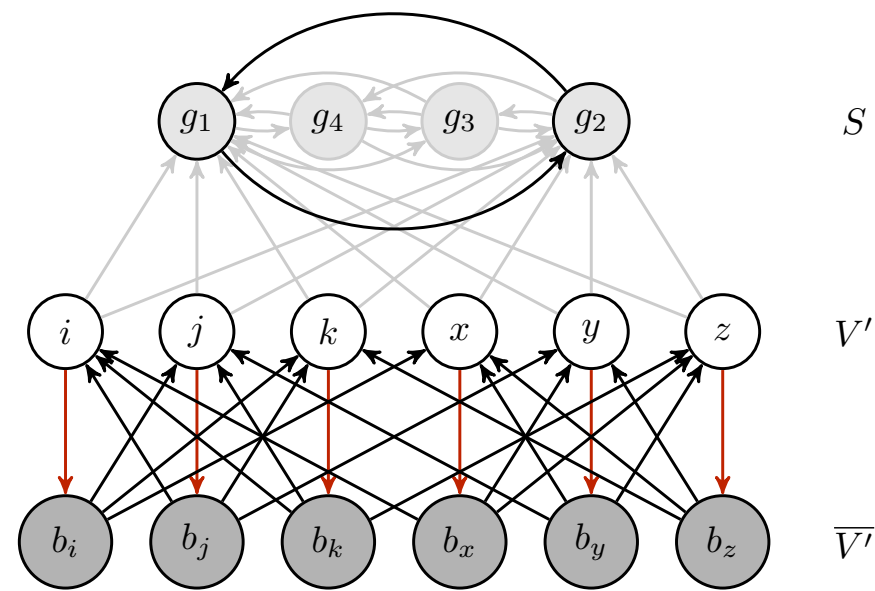
Walks starting at  $i \in V'$



Walks starting at  $b_i \in \overline{V'}$

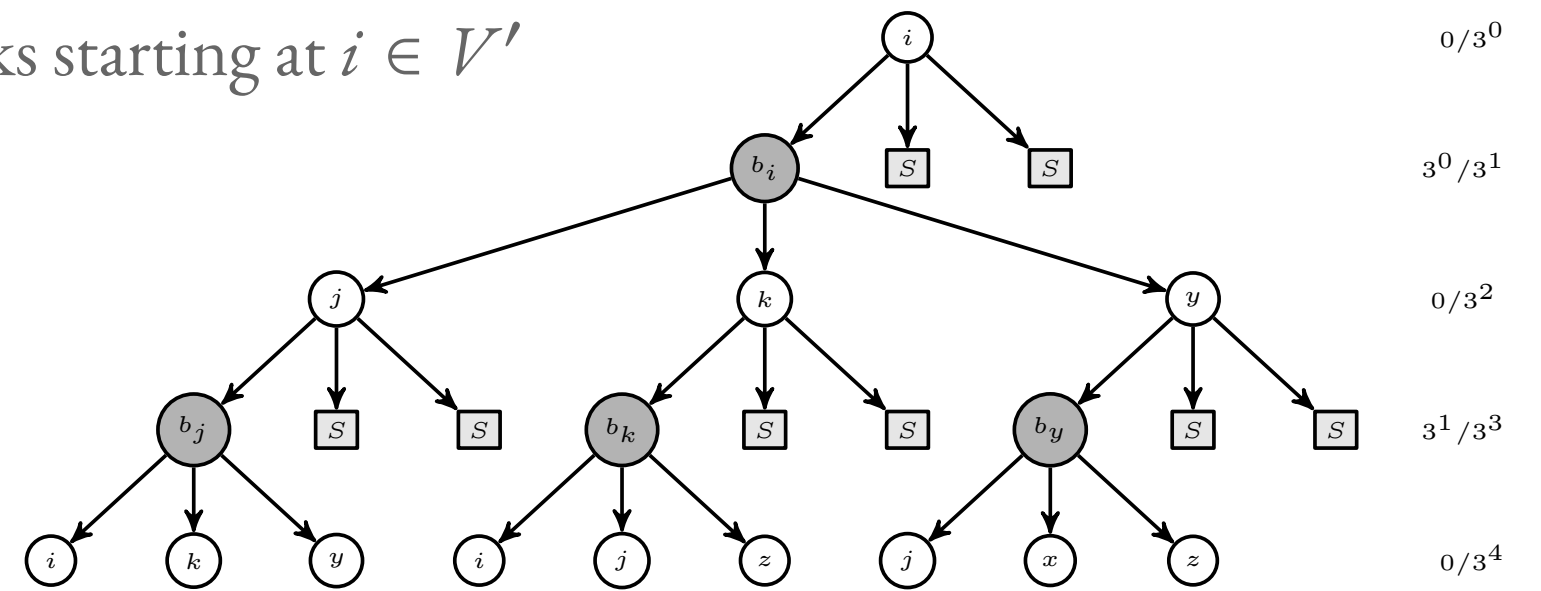


# NP-Hardness: Intuition

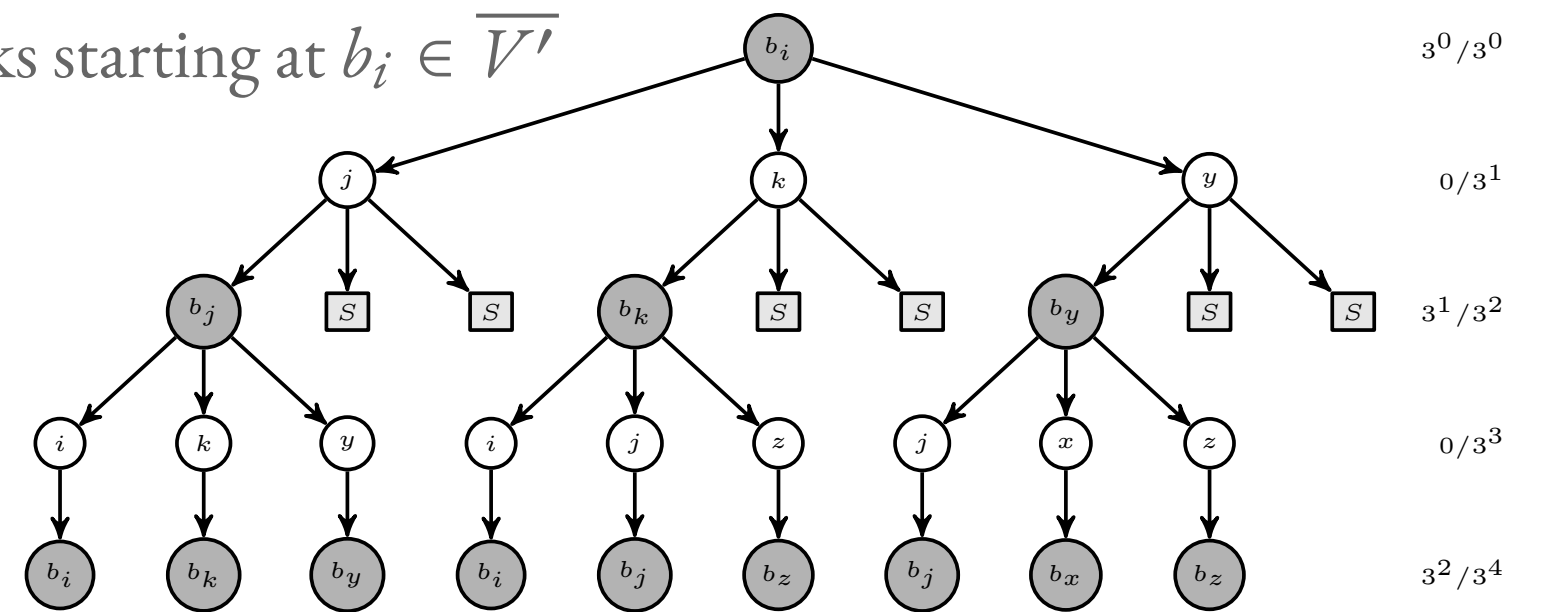


$$f(G) = n' \underbrace{\sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \overline{V'}}$$

Walks starting at  $i \in V'$

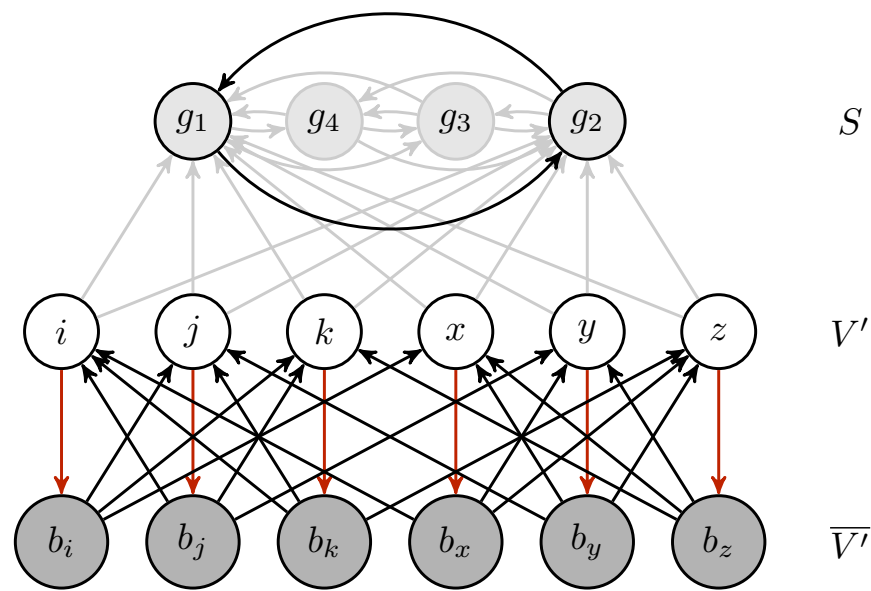


Walks starting at  $b_i \in \overline{V'}$

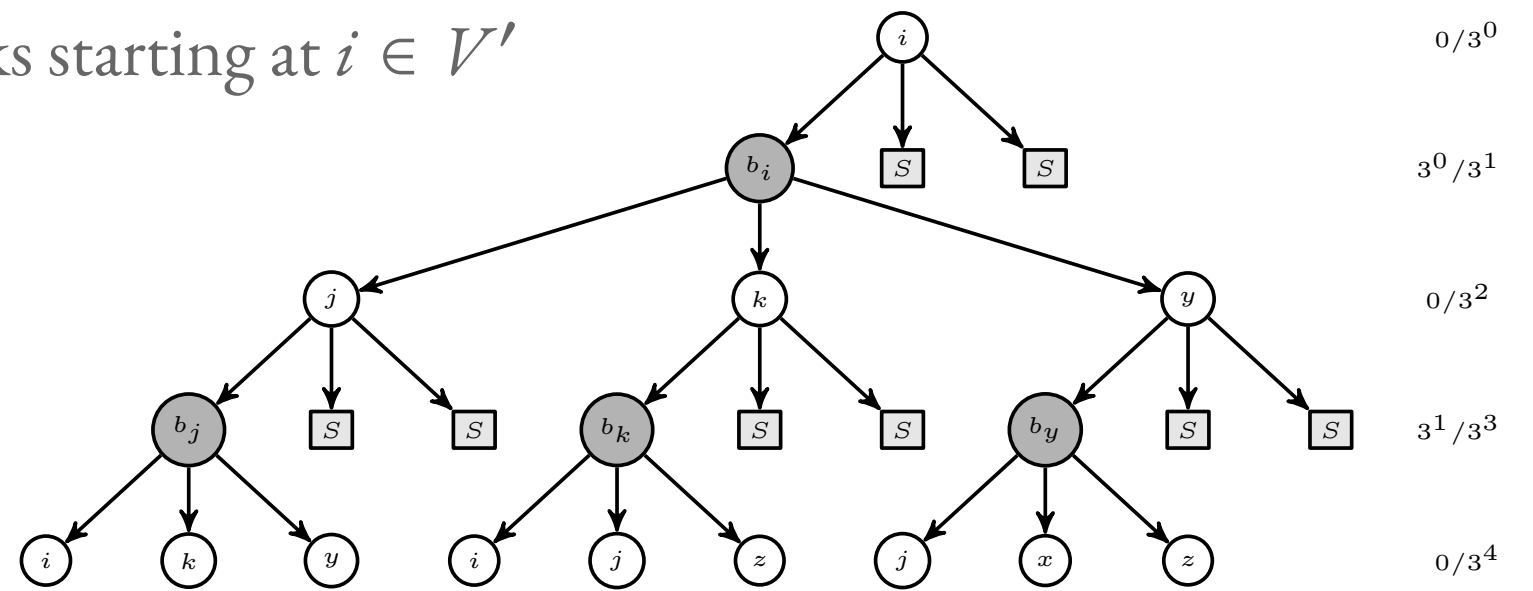




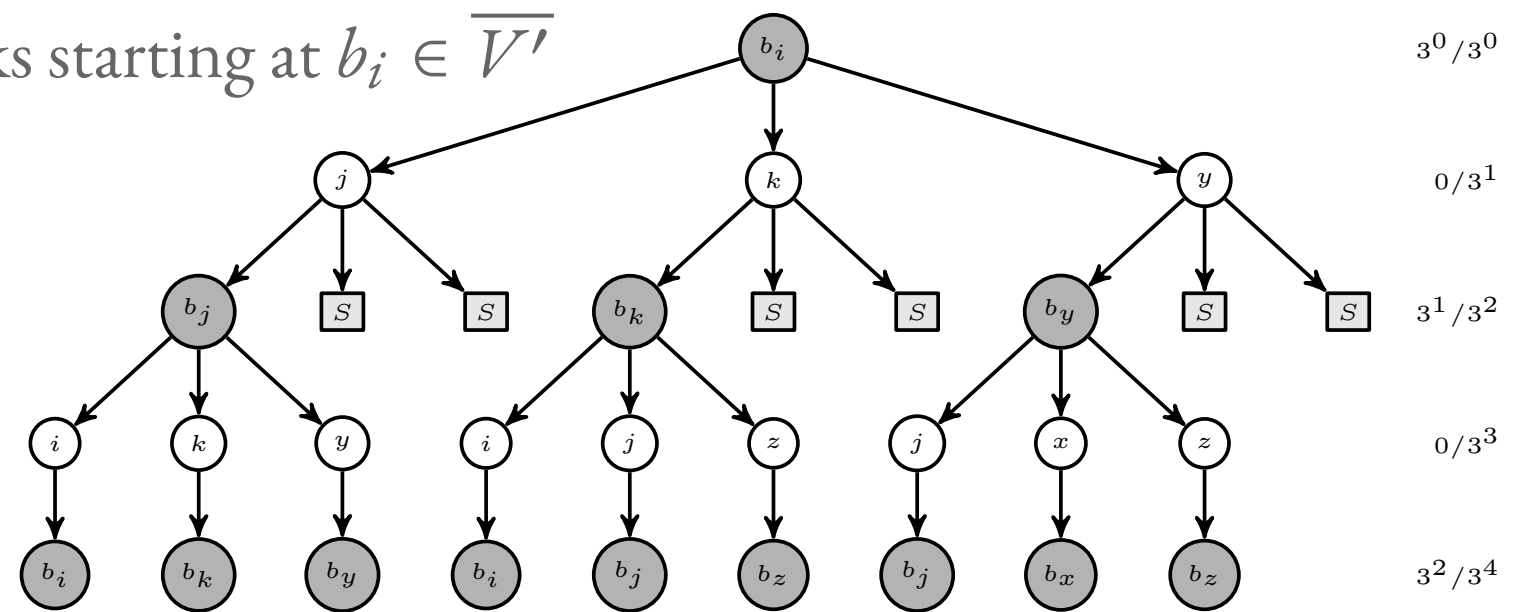
# NP-Hardness: Intuition



Walks starting at  $i \in V'$

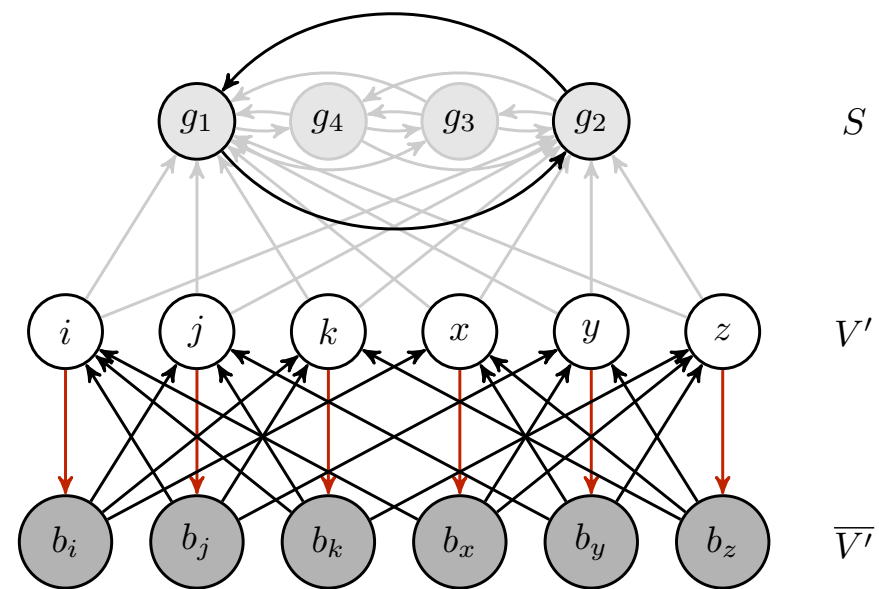


Walks starting at  $b_i \in \overline{V'}$



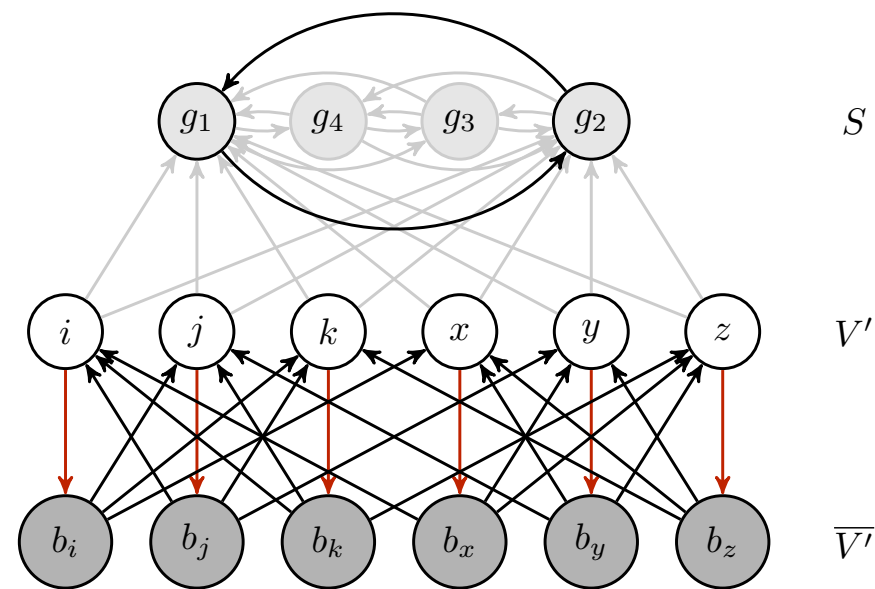
$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1-\alpha)^{2t}}_{\text{Contributions from } \overline{V'}} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1-\alpha)^{2t+1}}_{\text{Contributions from } V'}$$

# NP-Hardness: Sketch



$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \bar{V}'} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1 - \alpha)^{2t+1}}_{\text{Contributions from } V'}$$

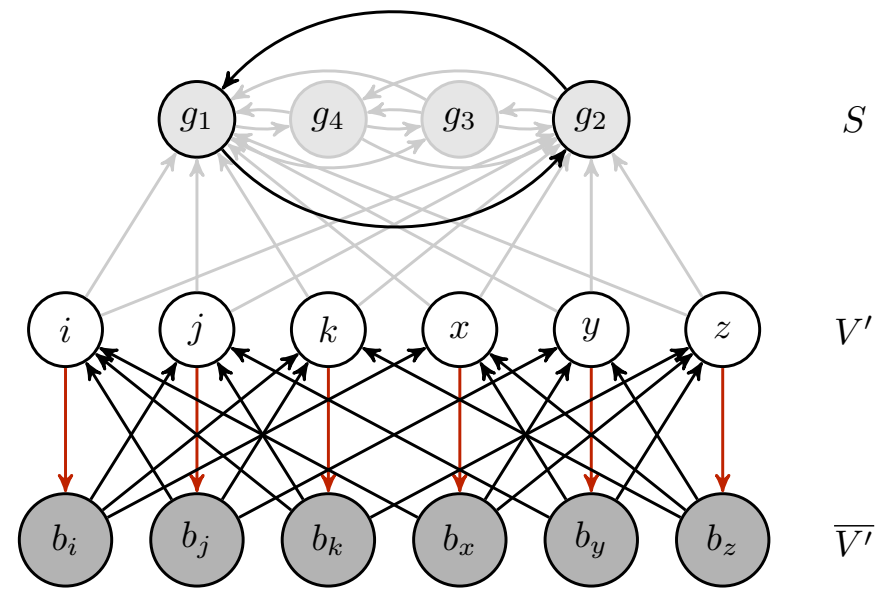
# NP-Hardness: Sketch



Candidate edges to rewire are  $\{(a, b_a) \mid a \in V'\}$

$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \bar{V}'} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1 - \alpha)^{2t+1}}_{\text{Contributions from } V'}$$

# NP-Hardness: Sketch

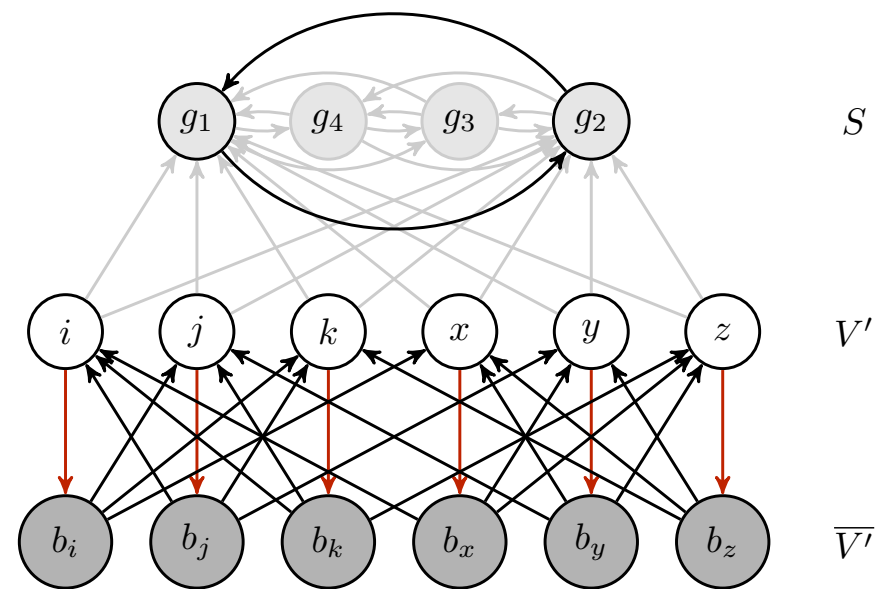


Candidate edges to rewire are  $\{(a, b_a) \mid a \in V'\}$

Rewiring  $(i, b_i) \rightarrow (i, g_n)$  in  $G \leftrightarrow i \in \text{VC}(G)$

$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \bar{V}'} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1 - \alpha)^{2t+1}}_{\text{Contributions from } V'}$$

# NP-Hardness: Sketch



Candidate edges to rewire are  $\{(a, b_a) \mid a \in V'\}$

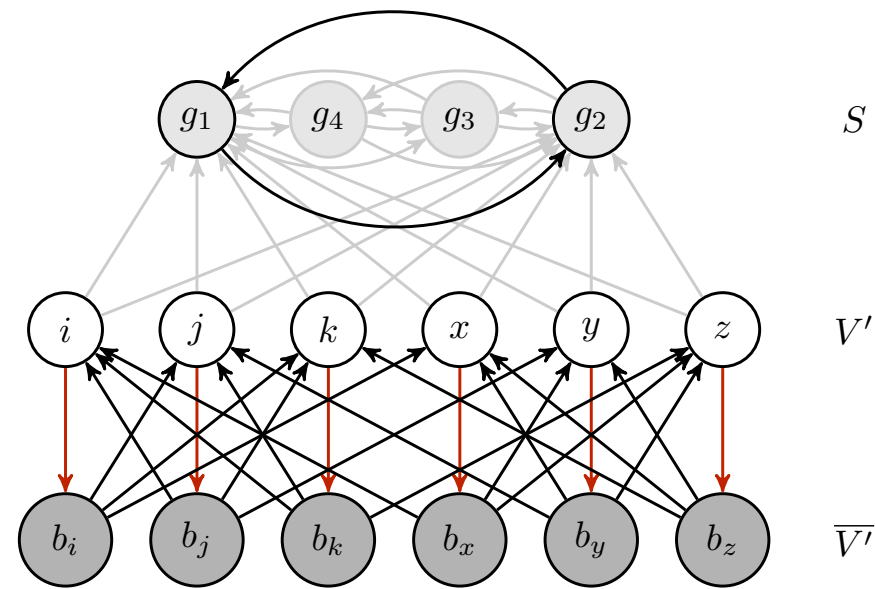
Rewiring  $(i, b_i) \rightarrow (i, g_n)$  in  $G \leftrightarrow i \in \text{VC}(G)$

Reduction of  $f(G)$  by individual rewiring

$$\frac{1}{3}(1 - \alpha) + \frac{1}{9}(1 - \alpha)^2 + \frac{2\gamma'}{27}(1 - \alpha)^3 + \varepsilon'$$

$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \bar{V}'} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1 - \alpha)^{2t+1}}_{\text{Contributions from } V'}$$

# NP-Hardness: Sketch



Candidate edges to rewire are  $\{(a, b_a) \mid a \in V'\}$

Rewiring  $(i, b_i) \rightarrow (i, g_n)$  in  $G \leftrightarrow i \in \text{VC}(G)$

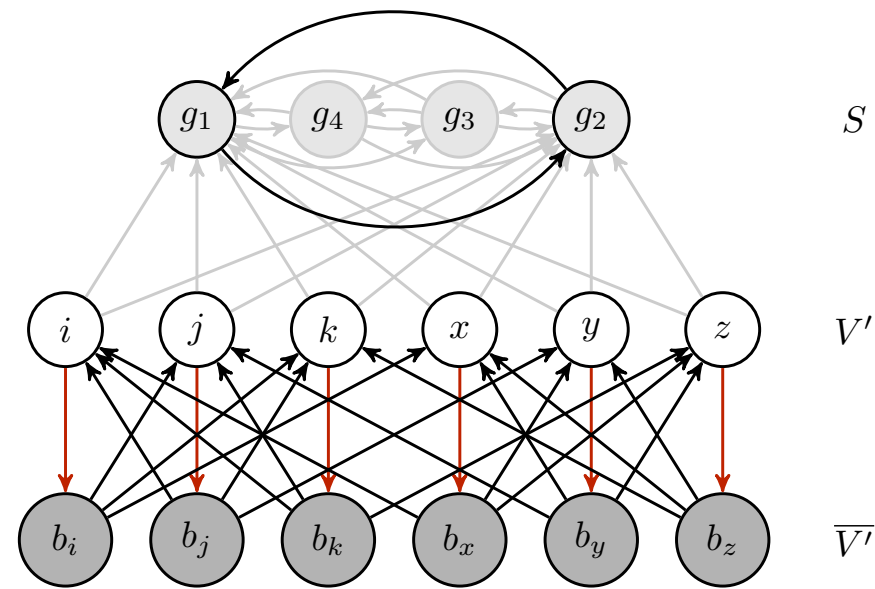
Reduction of  $f(G)$  by individual rewiring

$$\frac{1}{3}(1 - \alpha) + \frac{1}{9}(1 - \alpha)^2 + \frac{2\gamma'}{27}(1 - \alpha)^3 + \varepsilon'$$

$\gamma'$ : Number of newly covered edges in  $G'$

$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \bar{V}'} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1 - \alpha)^{2t+1}}_{\text{Contributions from } V'}$$

# NP-Hardness: Sketch



Candidate edges to rewire are  $\{(a, b_a) \mid a \in V'\}$

Rewiring  $(i, b_i) \rightarrow (i, g_n)$  in  $G \leftrightarrow i \in \text{VC}(G)$

Reduction of  $f(G)$  by individual rewiring

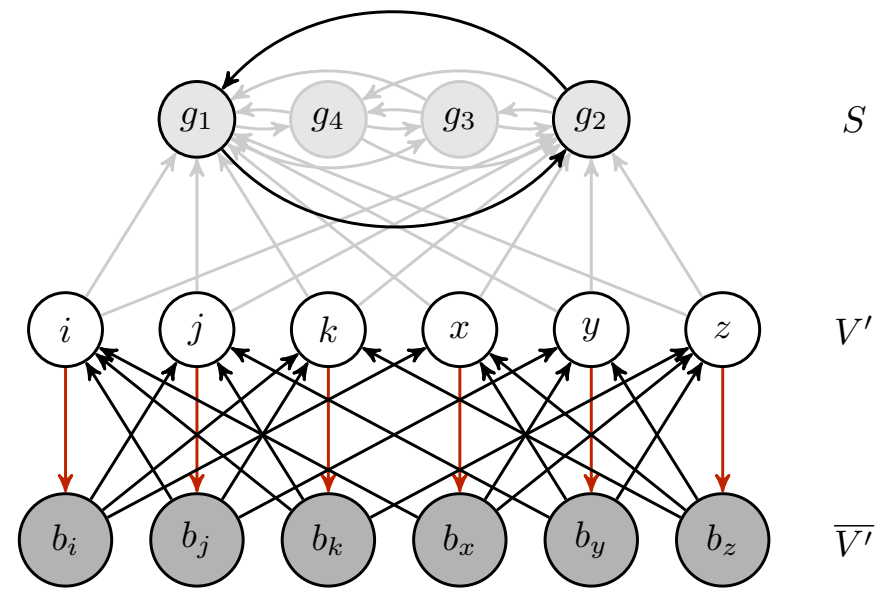
$$\frac{1}{3}(1 - \alpha) + \frac{1}{9}(1 - \alpha)^2 + \frac{2\gamma'}{27}(1 - \alpha)^3 + \varepsilon'$$

$\gamma'$ : Number of newly covered edges in  $G'$

$\varepsilon'$ : Contributions of walks beyond step 3

$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \bar{V}'} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1 - \alpha)^{2t+1}}_{\text{Contributions from } V'}$$

# NP-Hardness: Sketch



Candidate edges to rewire are  $\{(a, b_a) \mid a \in V'\}$

Rewiring  $(i, b_i) \rightarrow (i, g_n)$  in  $G \leftrightarrow i \in \text{VC}(G)$

Reduction of  $f(G)$  by individual rewiring

$$\frac{1}{3}(1 - \alpha) + \frac{1}{9}(1 - \alpha)^2 + \frac{2\gamma'}{27}(1 - \alpha)^3 + \varepsilon'$$

$\gamma'$ : Number of newly covered edges in  $G'$

$\varepsilon'$ : Contributions of walks beyond step 3

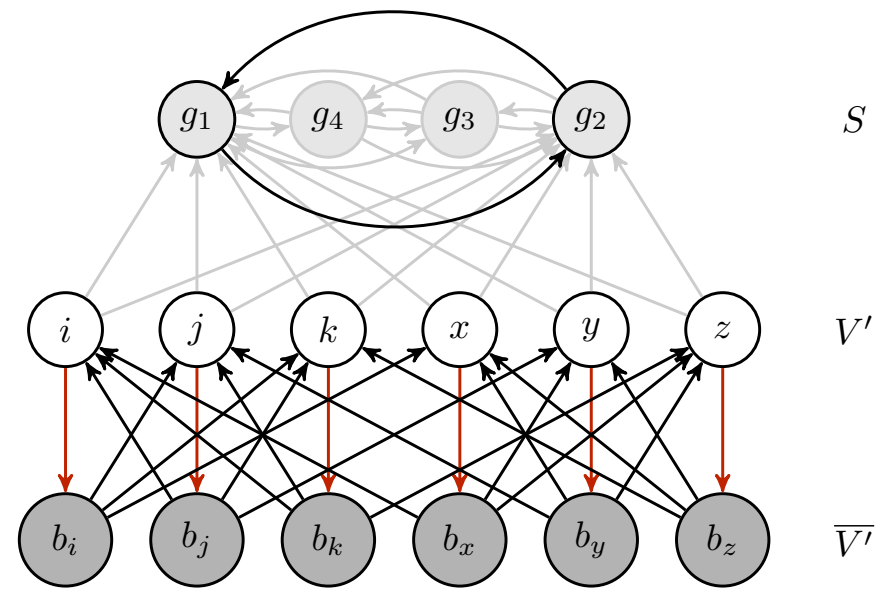
MVC of size at most  $r$

$$\leftrightarrow f_{\Delta}(G, G_r) = \frac{r}{3}(1 - \alpha) + \frac{3r}{9}(1 - \alpha)^2 + \frac{2m'}{27}(1 - \alpha)^3 + \varepsilon$$

$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \bar{V}'} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1 - \alpha)^{2t+1}}_{\text{Contributions from } V'}$$



# NP-Hardness: Sketch



Candidate edges to rewire are  $\{(a, b_a) \mid a \in V'\}$

Rewiring  $(i, b_i) \rightarrow (i, g_n)$  in  $G \leftrightarrow i \in \text{VC}(G)$

Reduction of  $f(G)$  by individual rewiring

$$\frac{1}{3}(1 - \alpha) + \frac{1}{9}(1 - \alpha)^2 + \frac{2\gamma'}{27}(1 - \alpha)^3 + \varepsilon'$$

$\gamma'$ : Number of newly covered edges in  $G'$

$\varepsilon'$ : Contributions of walks beyond step 3

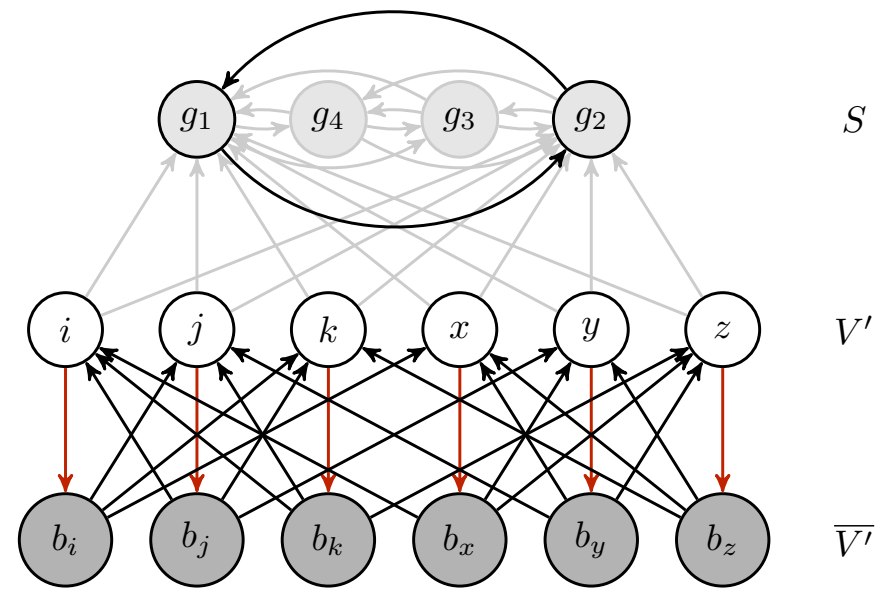
MVC of size at most  $r$

$$\leftrightarrow f_{\Delta}(G, G_r) = \frac{r}{3}(1 - \alpha) + \frac{3r}{9}(1 - \alpha)^2 + \frac{2m'}{27}(1 - \alpha)^3 + \varepsilon$$

For  $0.5 \leq \alpha \leq 1$ , we have  $\xi < \frac{2}{27}(1 - \alpha)^3$

$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \bar{V}'} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1 - \alpha)^{2t+1}}_{\text{Contributions from } V'}$$

# NP-Hardness: Sketch



Candidate edges to rewire are  $\{(a, b_a) \mid a \in V'\}$

Rewiring  $(i, b_i) \rightarrow (i, g_n)$  in  $G \leftrightarrow i \in \text{VC}(G)$

Reduction of  $f(G)$  by individual rewiring

$$\frac{1}{3}(1 - \alpha) + \frac{1}{9}(1 - \alpha)^2 + \frac{2\gamma'}{27}(1 - \alpha)^3 + \varepsilon'$$

$\gamma'$ : Number of newly covered edges in  $G'$

$\varepsilon'$ : Contributions of walks beyond step 3

MVC of size at most  $r$

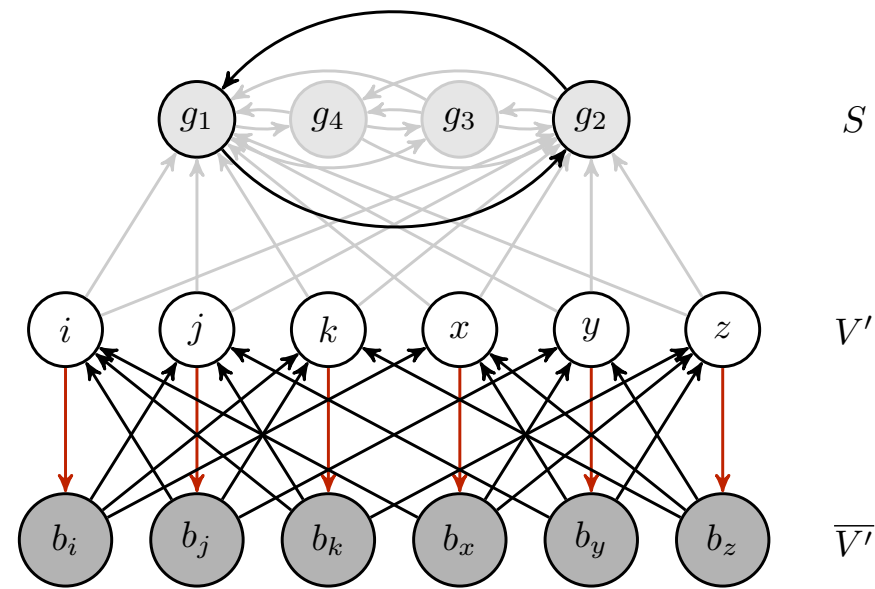
$$\leftrightarrow f_{\Delta}(G, G_r) = \frac{r}{3}(1 - \alpha) + \frac{3r}{9}(1 - \alpha)^2 + \frac{2m'}{27}(1 - \alpha)^3 + \varepsilon$$

For  $0.5 \leq \alpha \leq 1$ , we have  $\xi < \frac{2}{27}(1 - \alpha)^3$

$\xi$ : All harm beyond step 3

$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \bar{V}'} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1 - \alpha)^{2t+1}}_{\text{Contributions from } V'}$$

# NP-Hardness: Sketch



Candidate edges to rewire are  $\{(a, b_a) \mid a \in V'\}$

Rewiring  $(i, b_i) \rightarrow (i, g_n)$  in  $G \leftrightarrow i \in \text{VC}(G)$

Reduction of  $f(G)$  by individual rewiring

$$\frac{1}{3}(1 - \alpha) + \frac{1}{9}(1 - \alpha)^2 + \frac{2\gamma'}{27}(1 - \alpha)^3 + \varepsilon'$$

$\gamma'$ : Number of newly covered edges in  $G'$

$\varepsilon'$ : Contributions of walks beyond step 3

MVC of size at most  $r$

$$\leftrightarrow f_{\Delta}(G, G_r) = \frac{r}{3}(1 - \alpha) + \frac{3r}{9}(1 - \alpha)^2 + \frac{2m'}{27}(1 - \alpha)^3 + \varepsilon$$

For  $0.5 \leq \alpha \leq 1$ , we have  $\xi < \frac{2}{27}(1 - \alpha)^3$

$\xi$ : All harm beyond step 3

→ Can check “truncated”  $f(G_r)$  to decide MVC

$$f(G) = \underbrace{n' \sum_{t=0}^{\infty} 3^{-t} (1 - \alpha)^{2t}}_{\text{Contributions from } \bar{V}'} + \underbrace{n' \sum_{t=0}^{\infty} 3^{-t-1} (1 - \alpha)^{2t+1}}_{\text{Contributions from } V'}$$