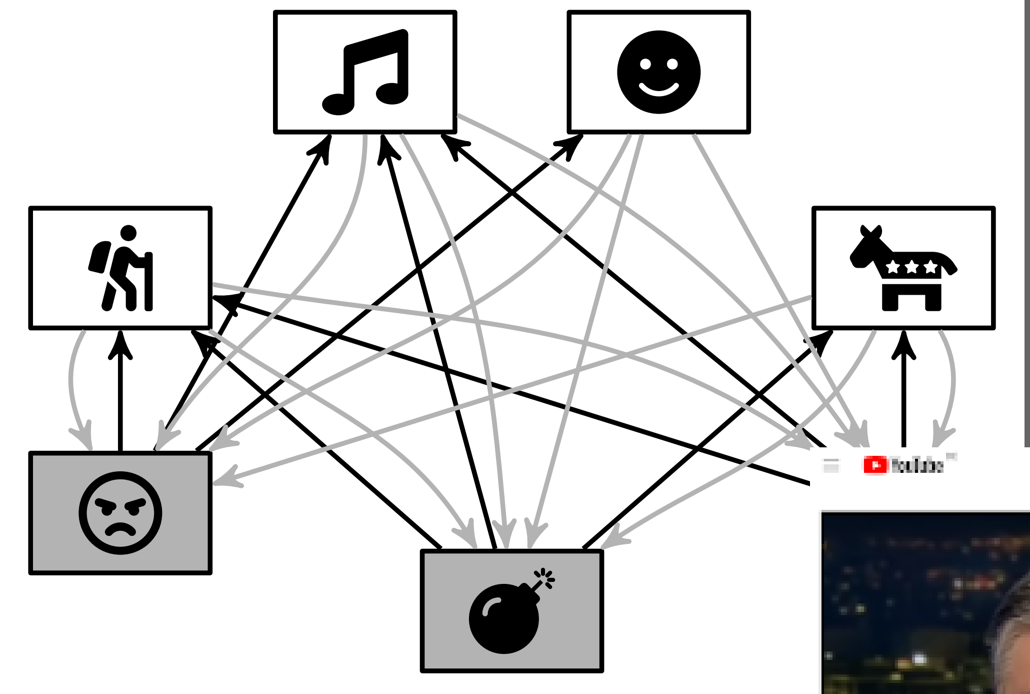
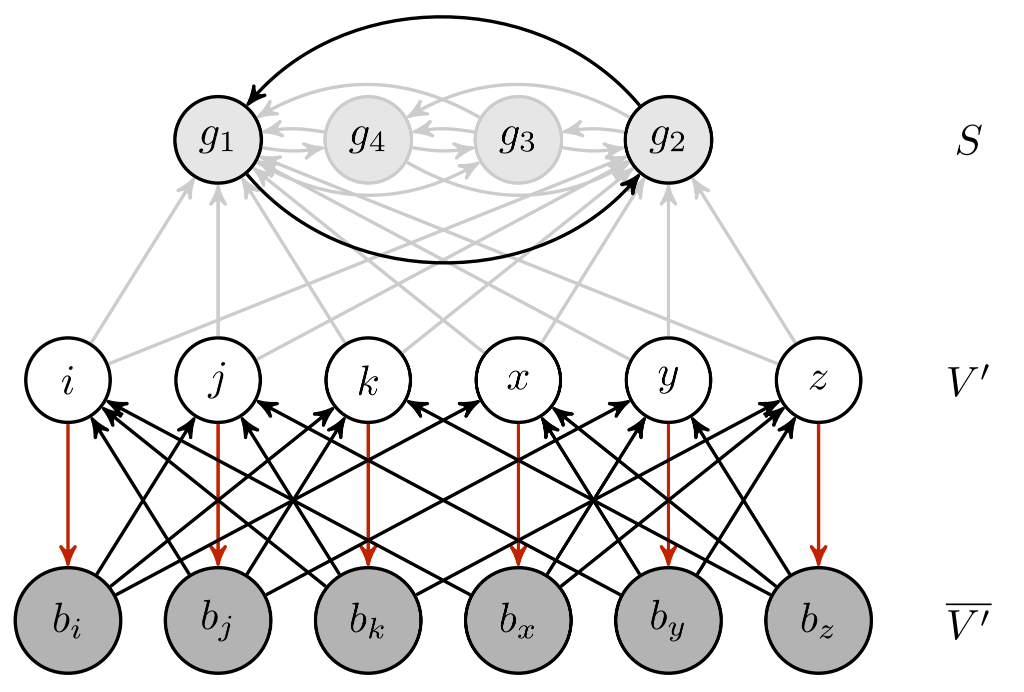




KTH VETENSKAP OCH KONST

Reducing Exposure to Harmful Content via Graph Rewiring

Corinna Coupette, Stefan Neumann, and Aristides Gionis

ACADEMY OF FINLAND
erc European Research Council Established by the European Commission
WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM
Social Mining & Big Data Ecosystem SoBigData RESEARCH INFRASTRUCTURE

Real-World Motivation

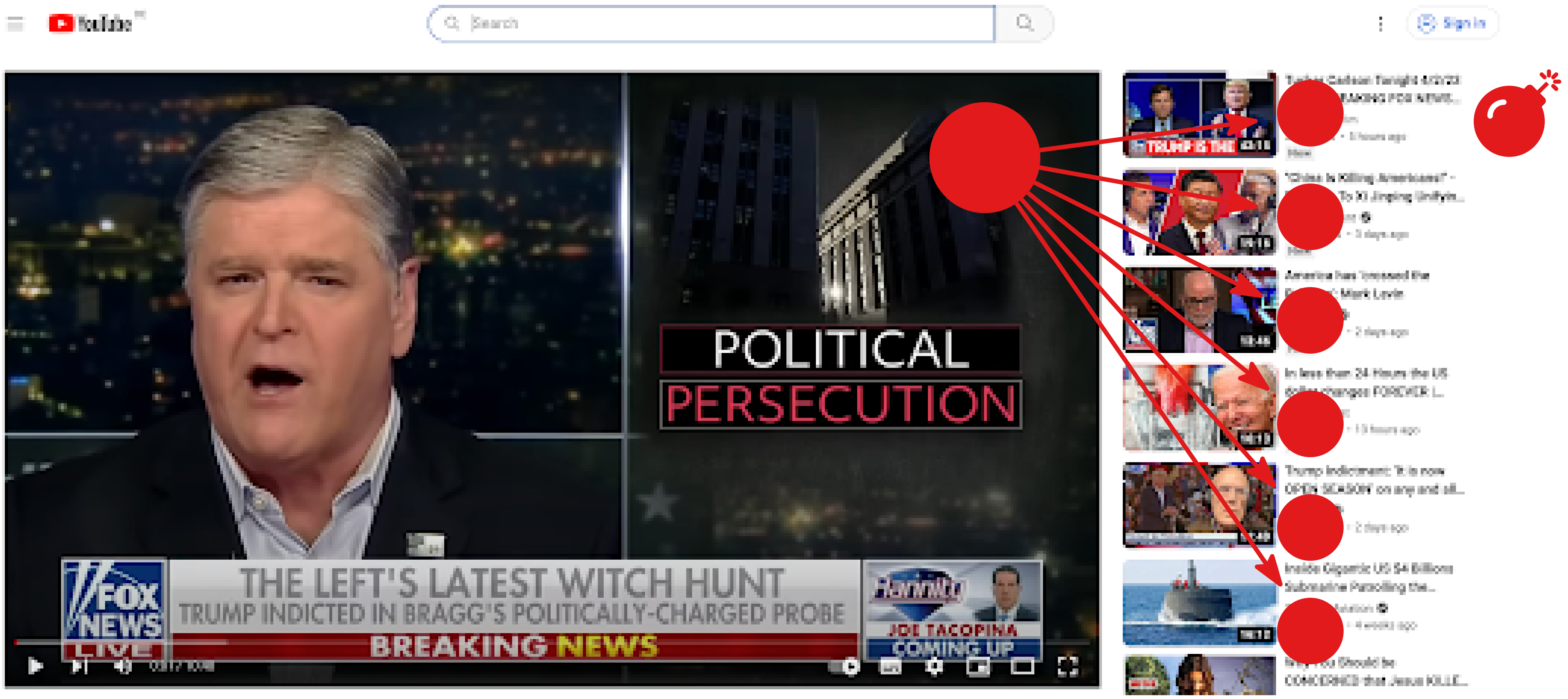
Real-World Motivation

Real-World Motivation

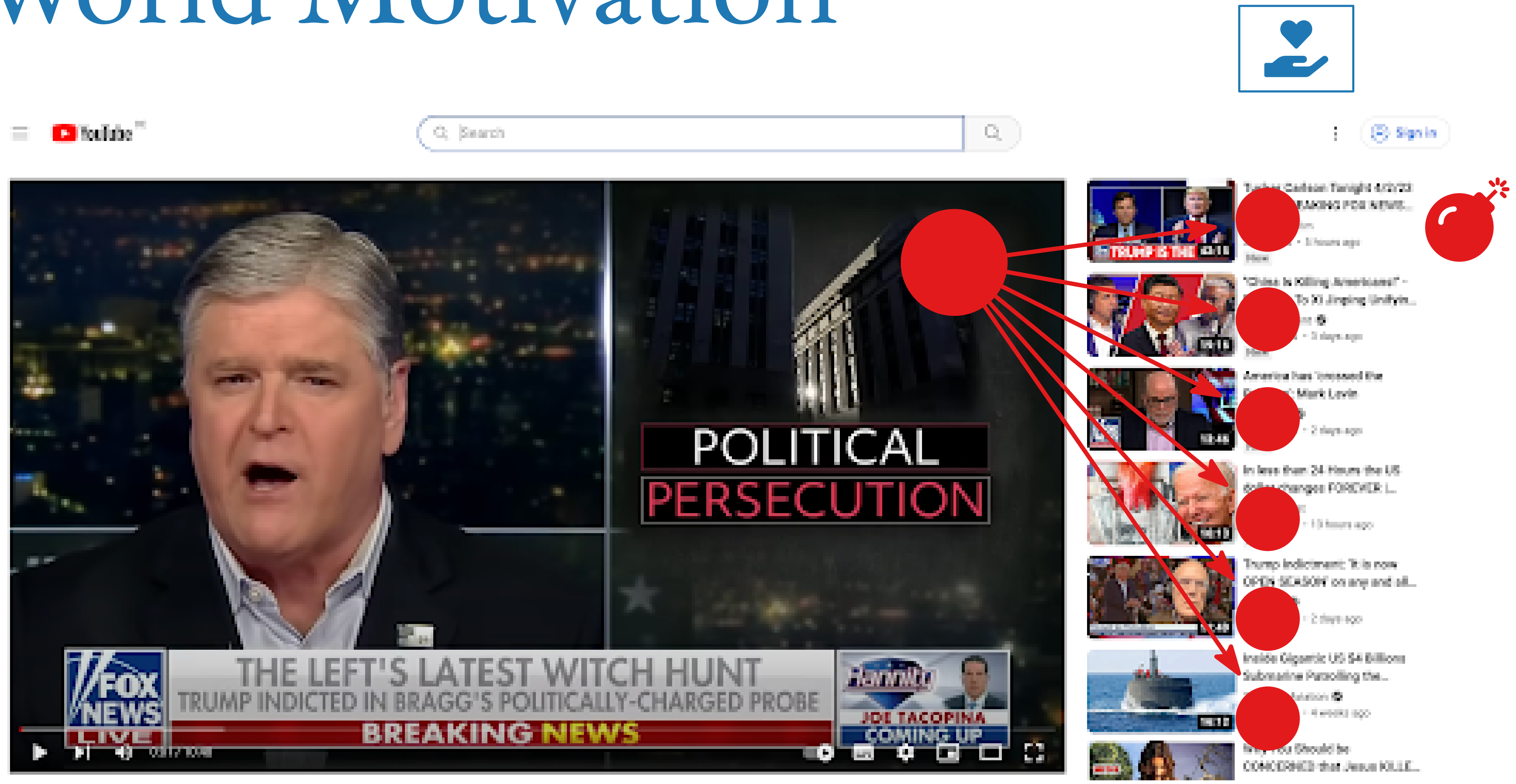
Real-World Motivation

Real-World Motivation

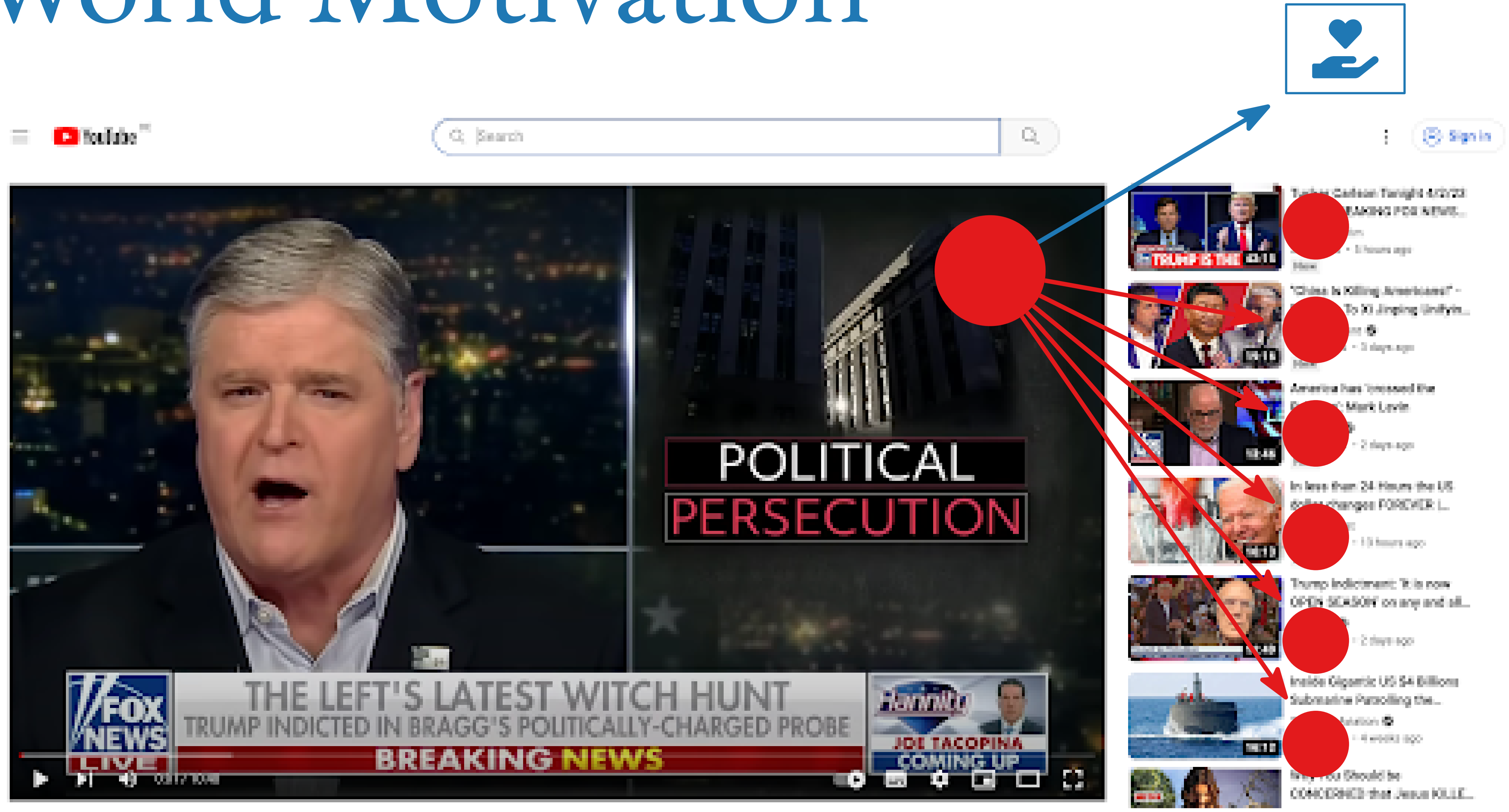
Real-World Motivation



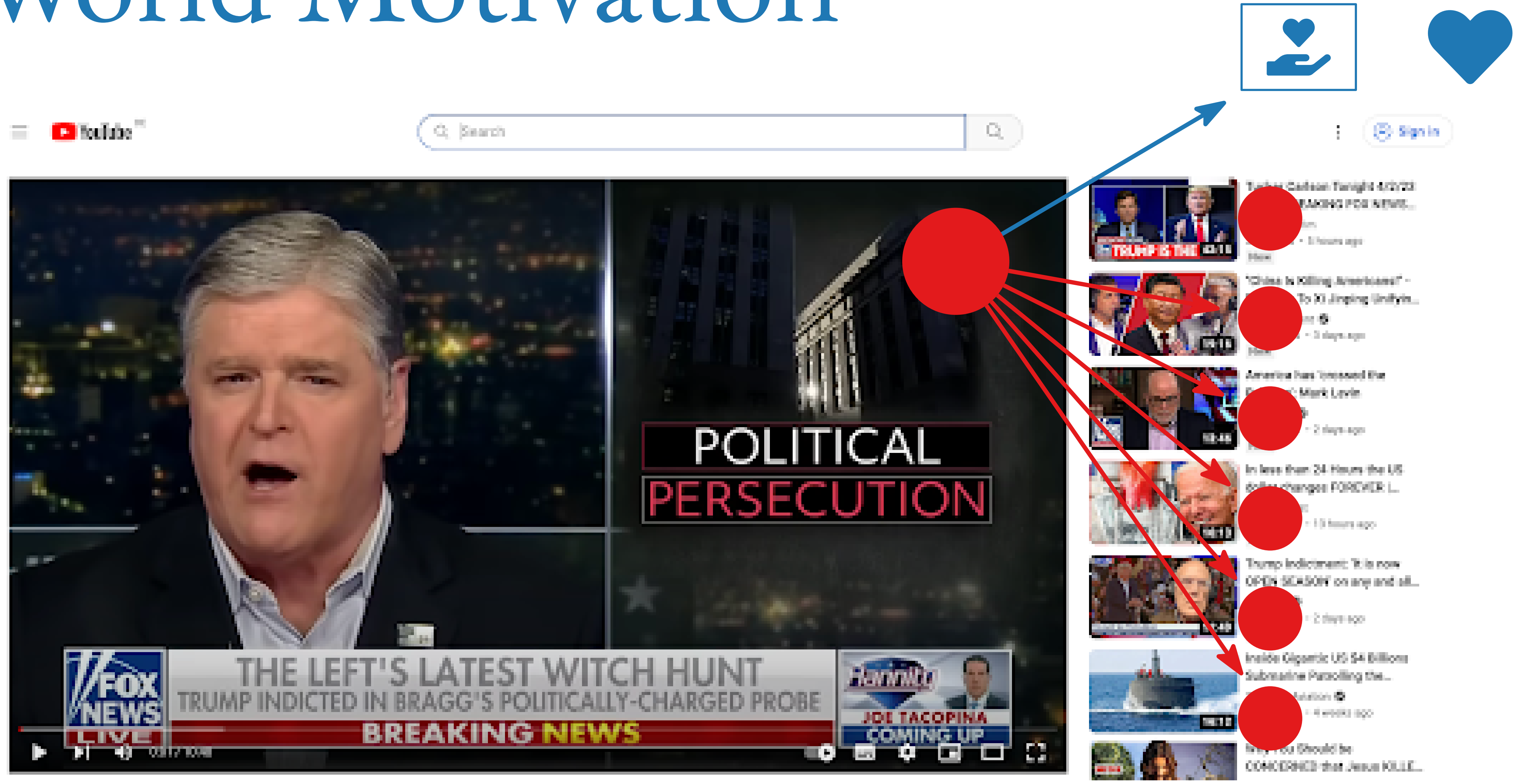
Real-World Motivation



Real-World Motivation

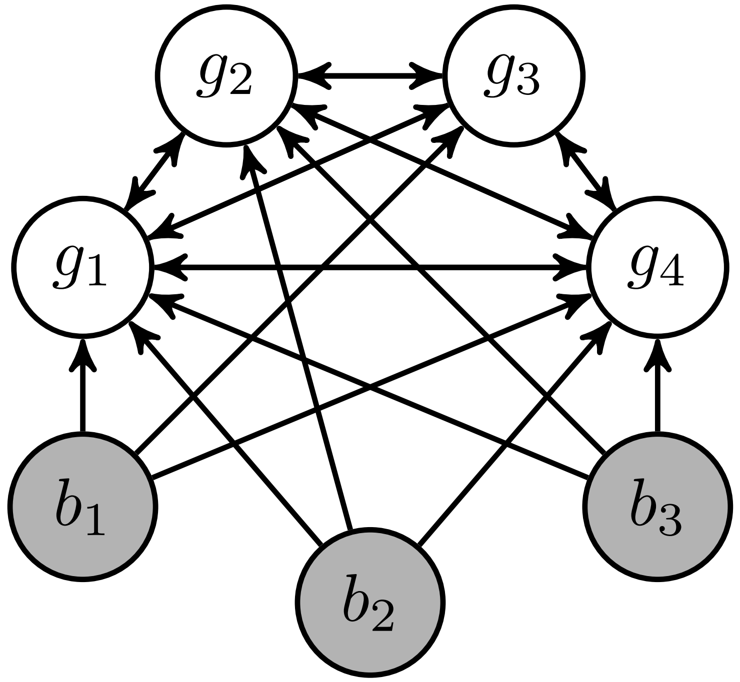


Real-World Motivation

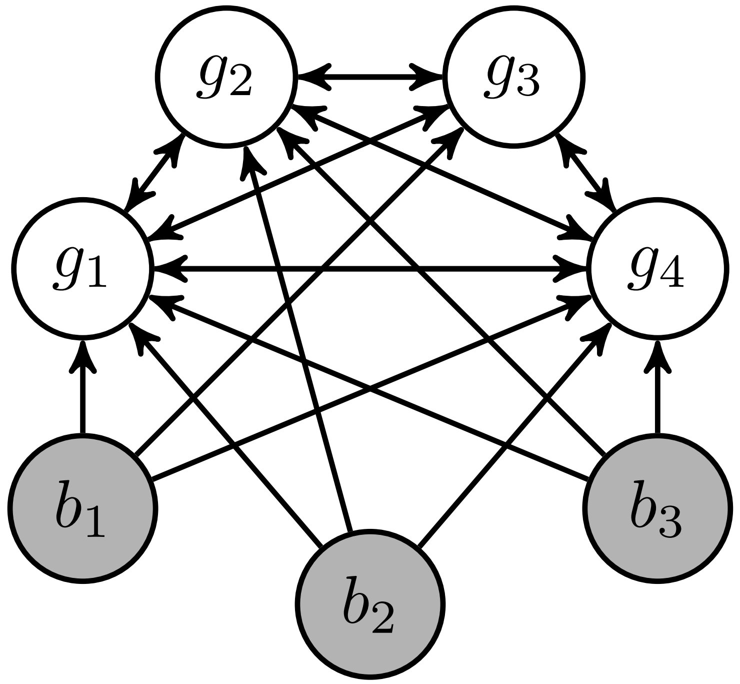


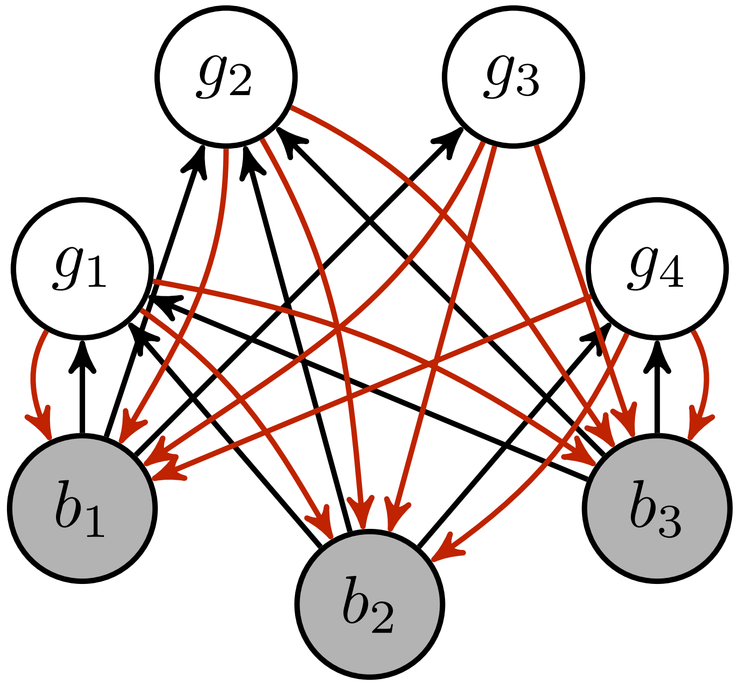
Research-World Motivation

Research-World Motivation

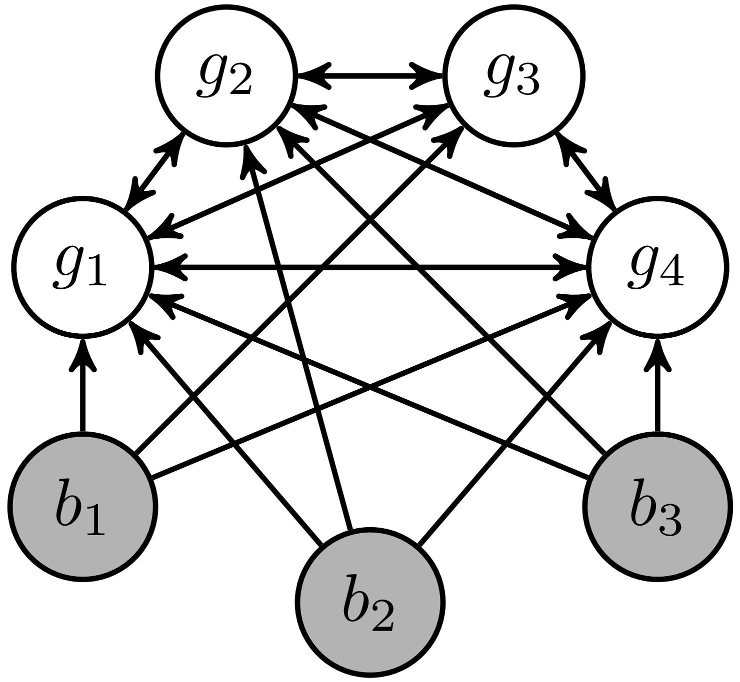


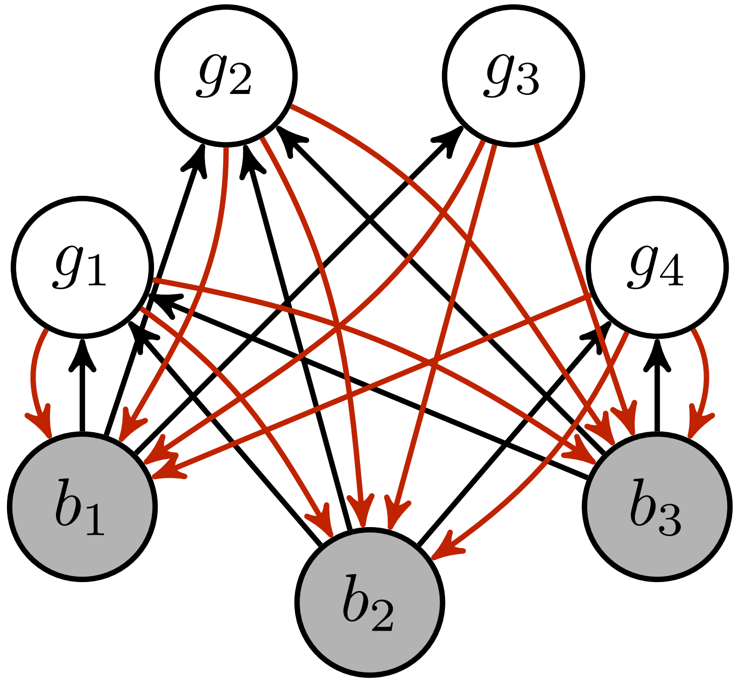
Research-World Motivation





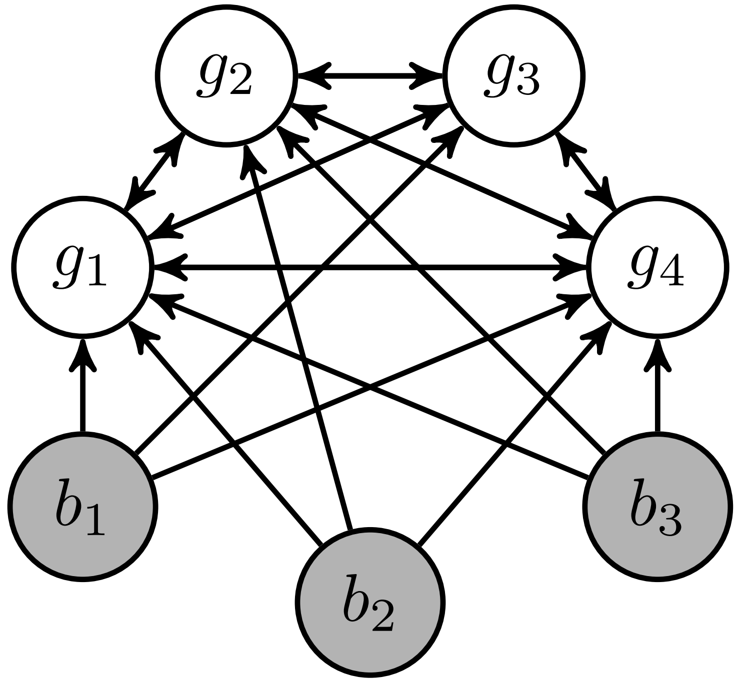
Research-World Motivation

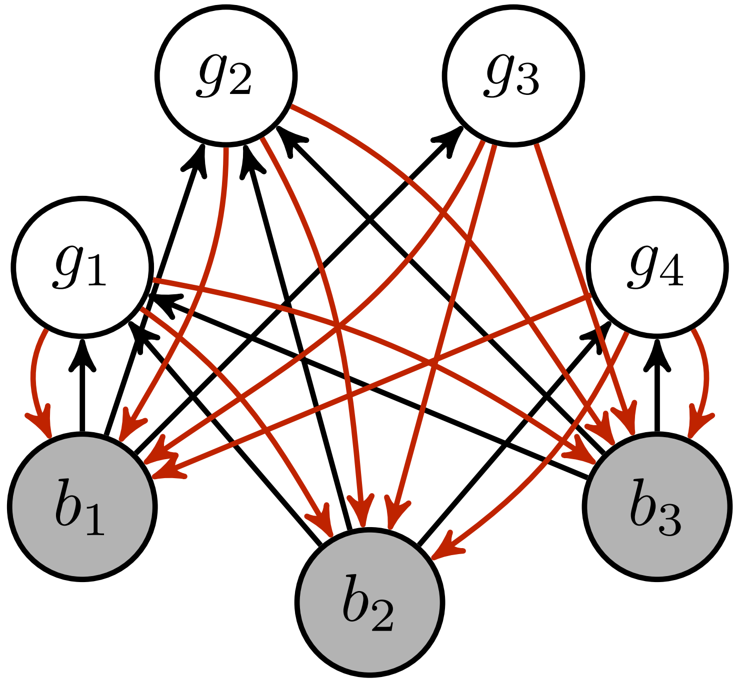




Both graphs minimize the *segregation* objective used in prior work (Fabbri et al. 2022)

Research-World Motivation





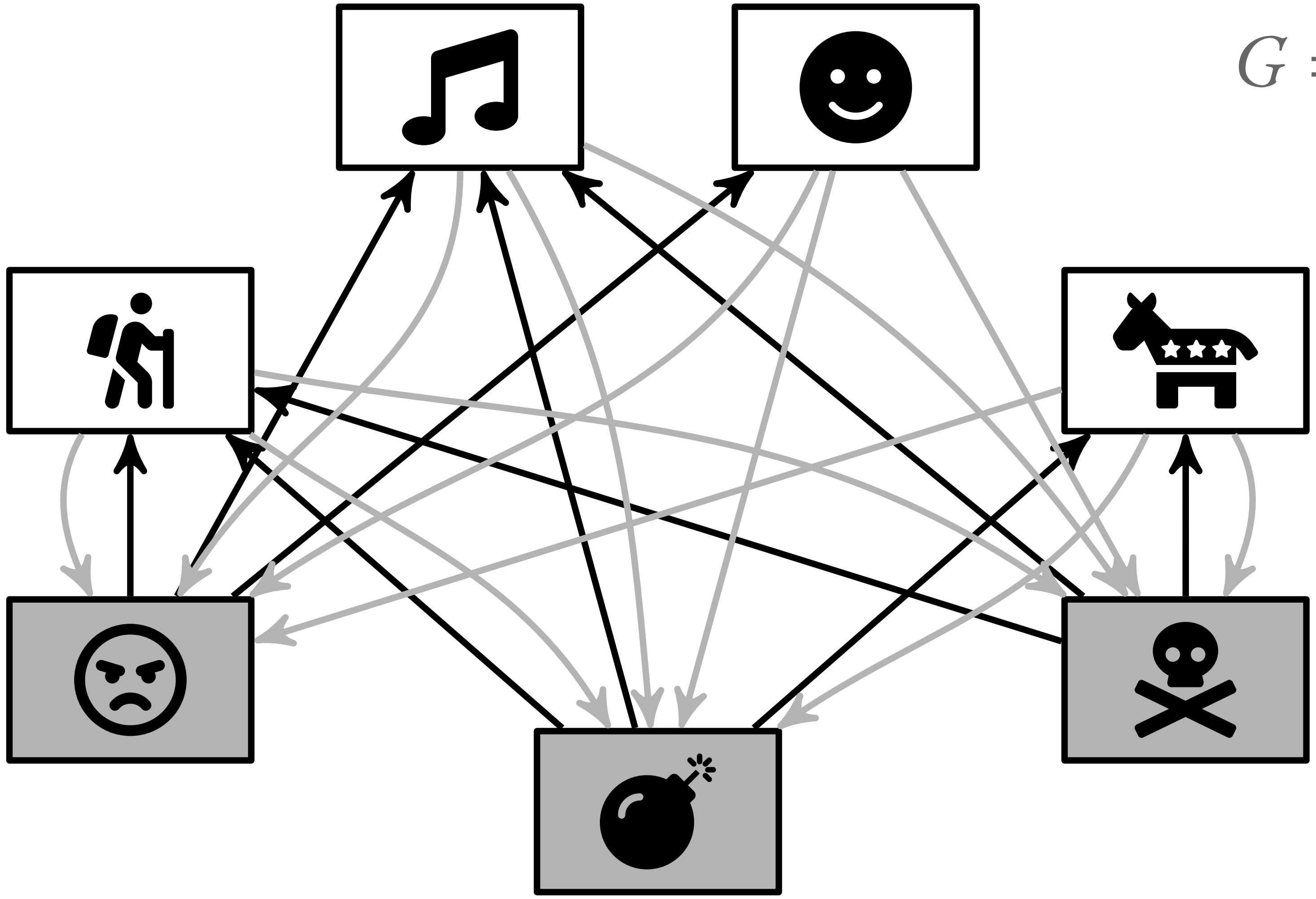
Both graphs minimize the *segregation* objective used in prior work (Fabbri et al. 2022)

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

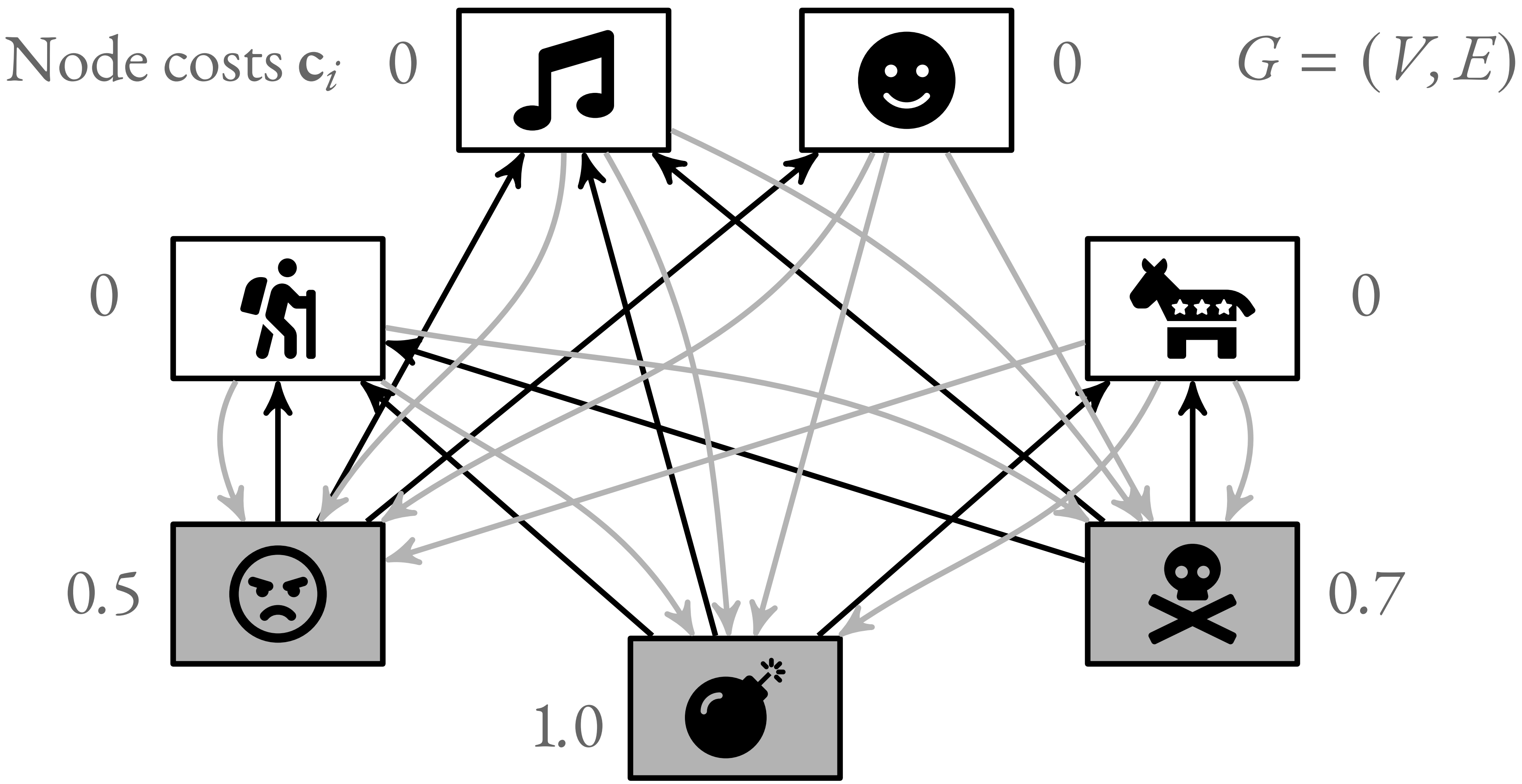
Model

Model

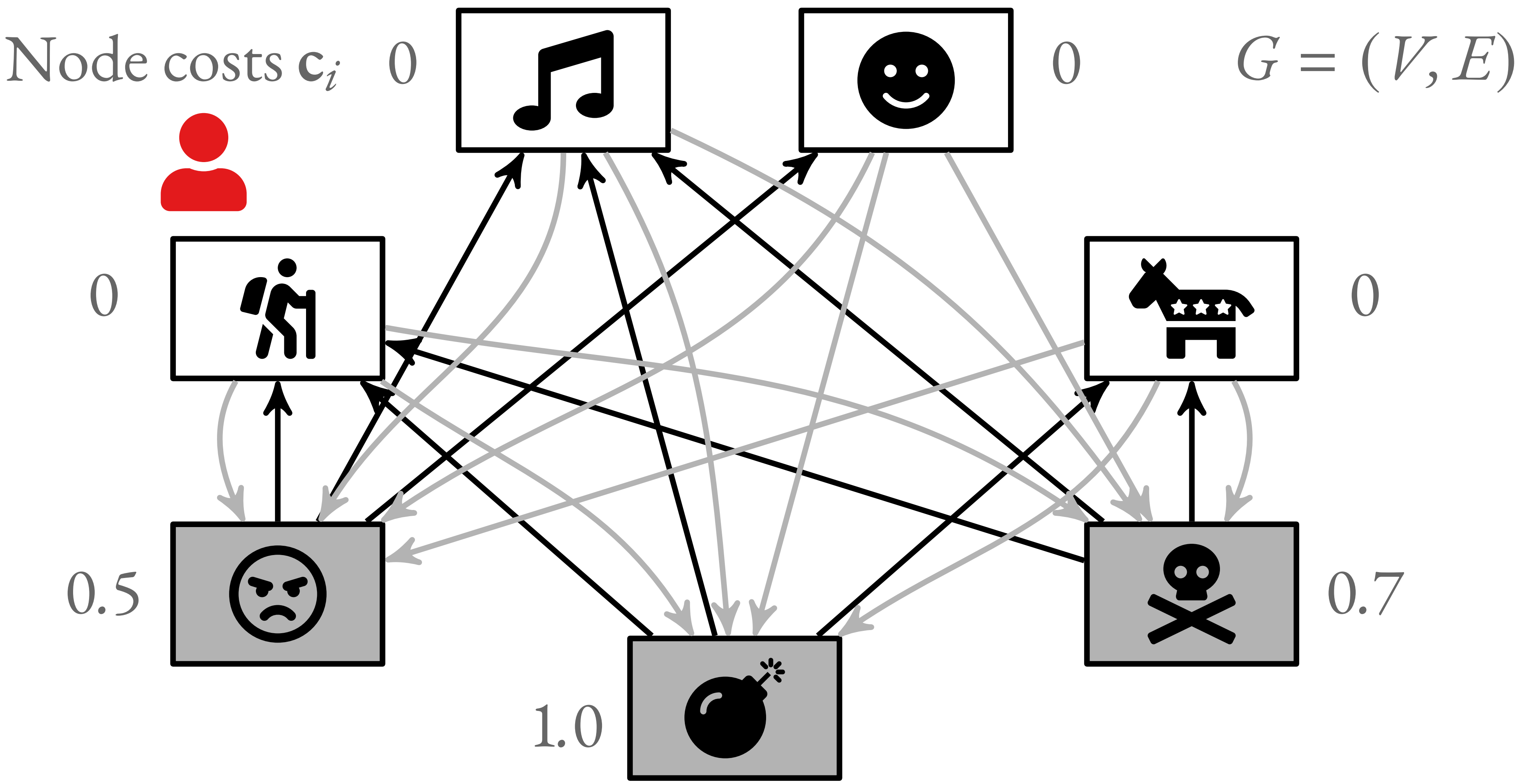
$G = (V, E)$



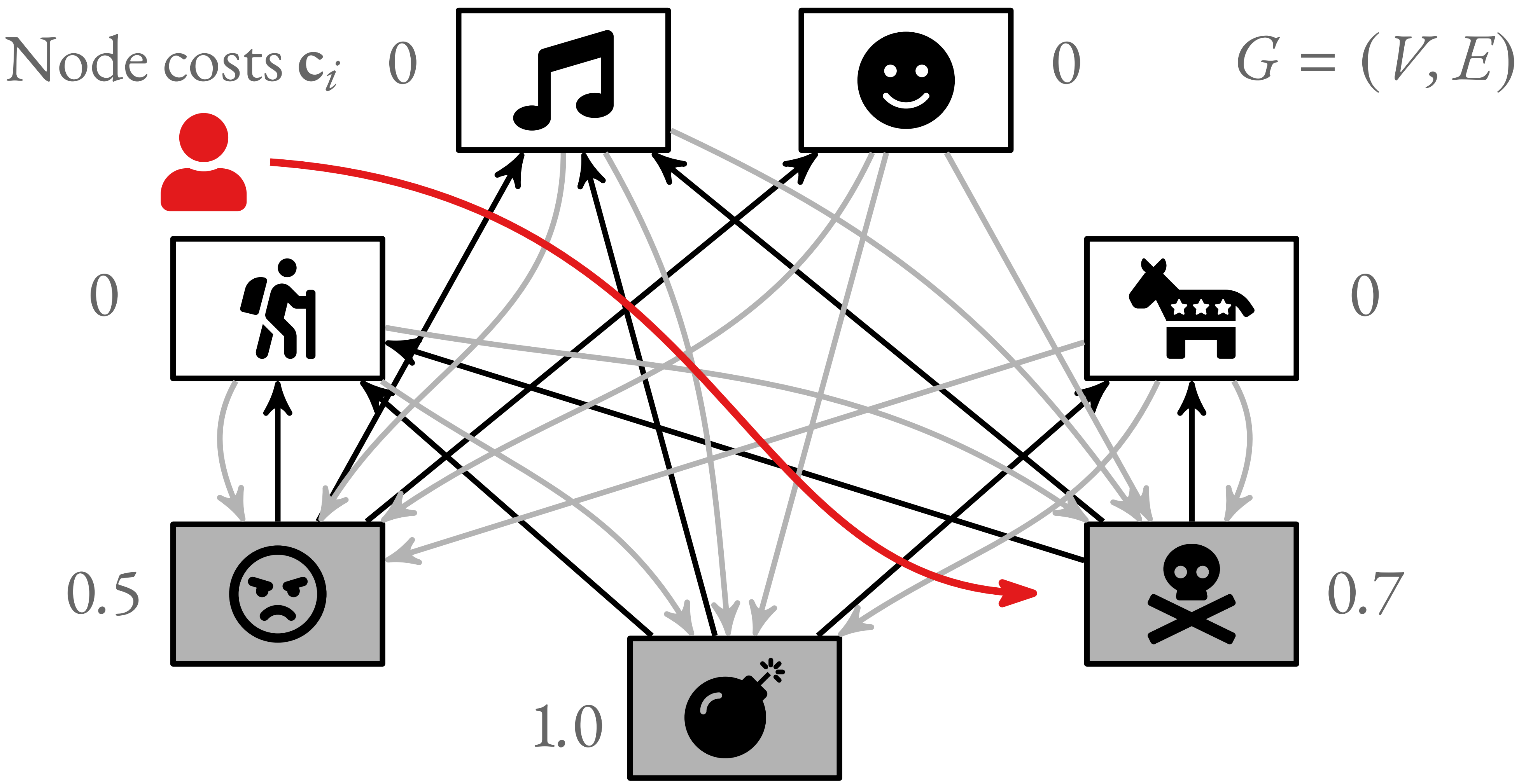
Model



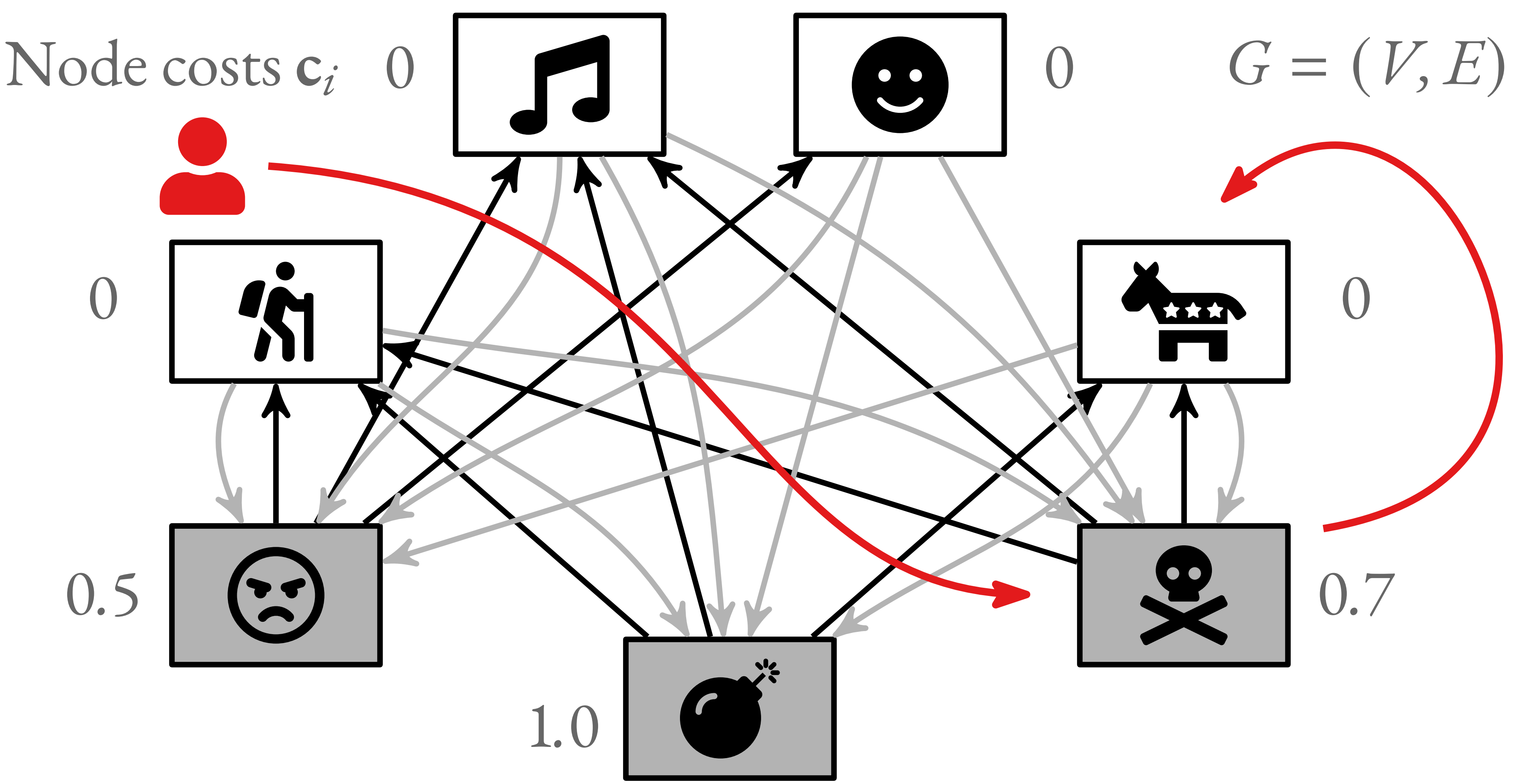
Model



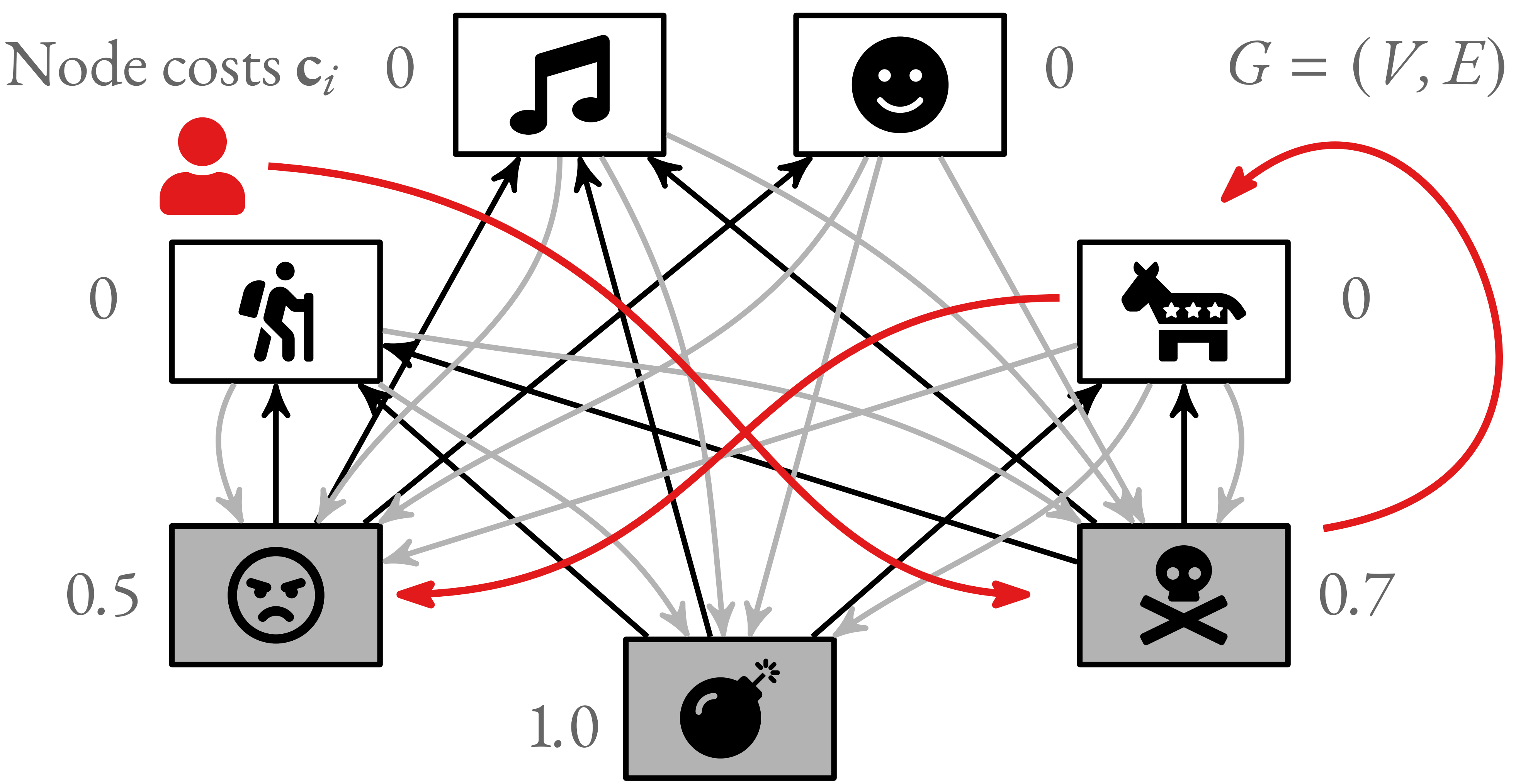
Model



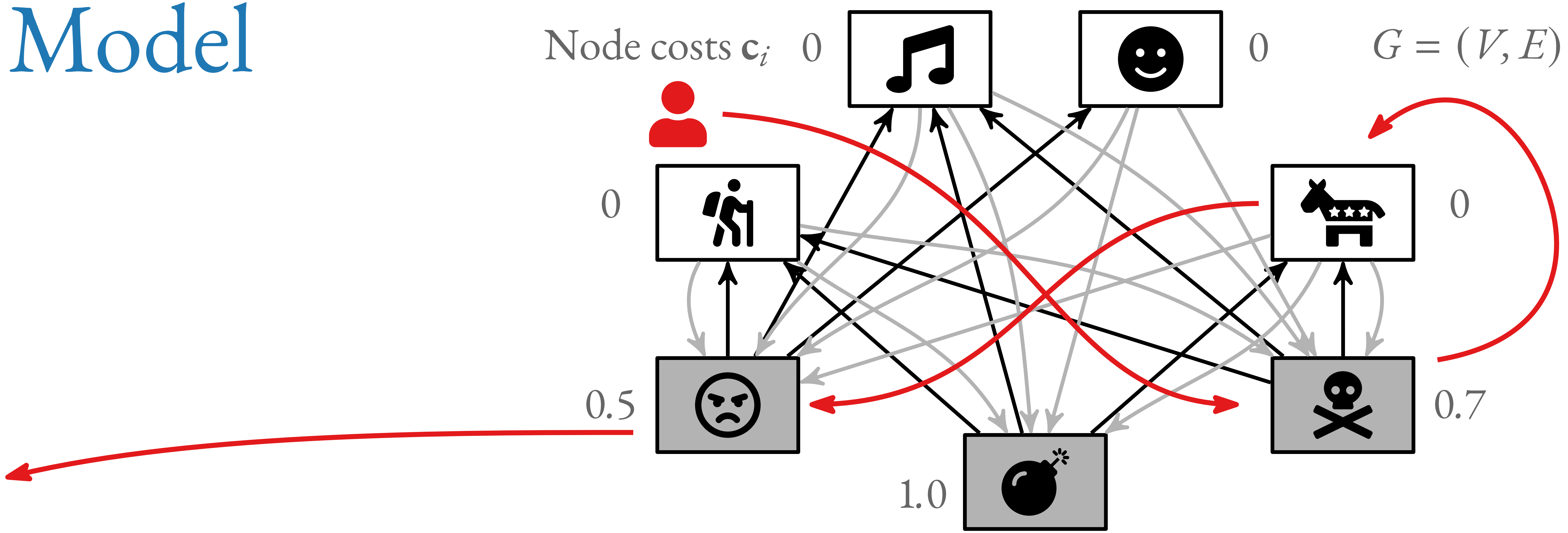
Model



Model



Model

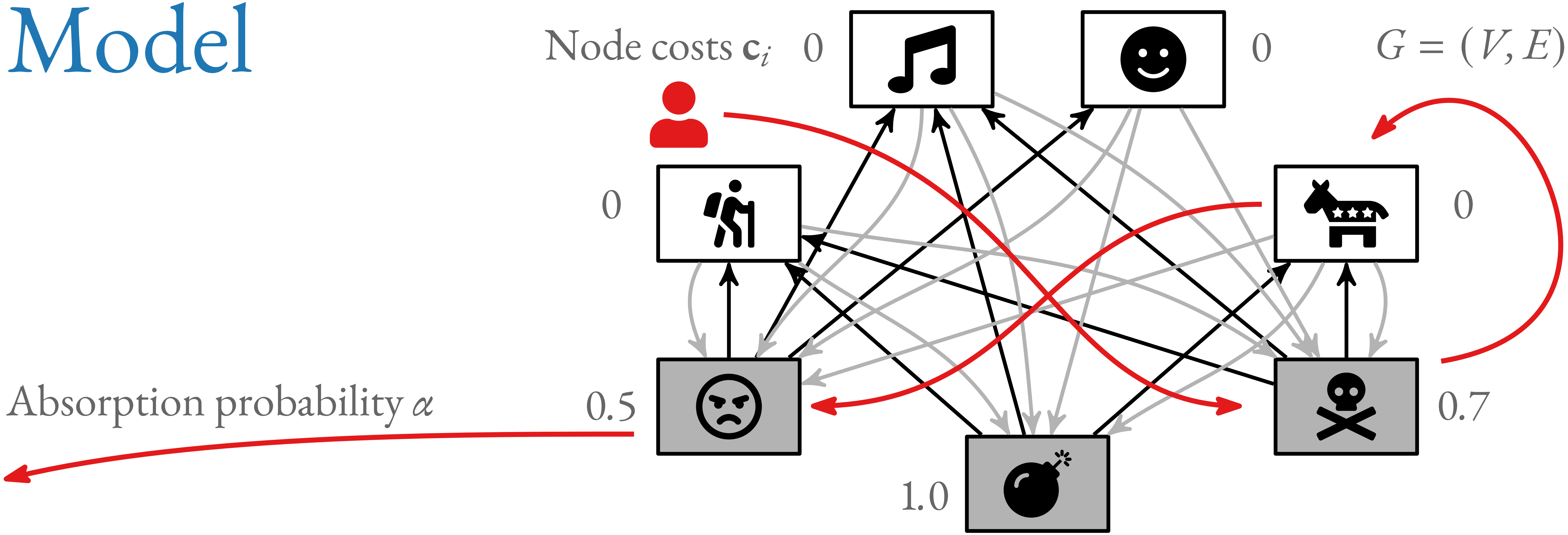


Model

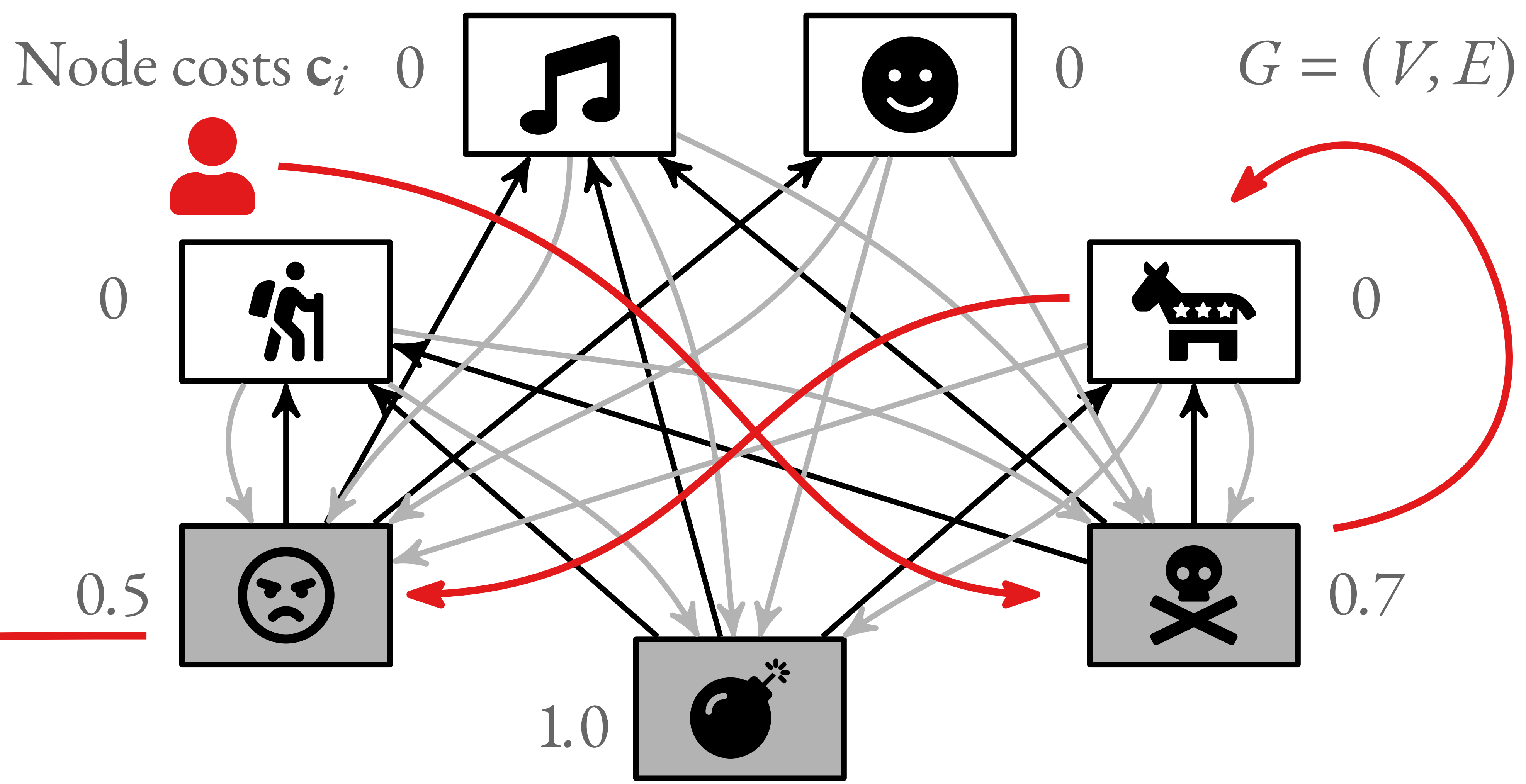
Node costs $\mathbf{c}_i$

$G = (V, E)$

Absorption probability α



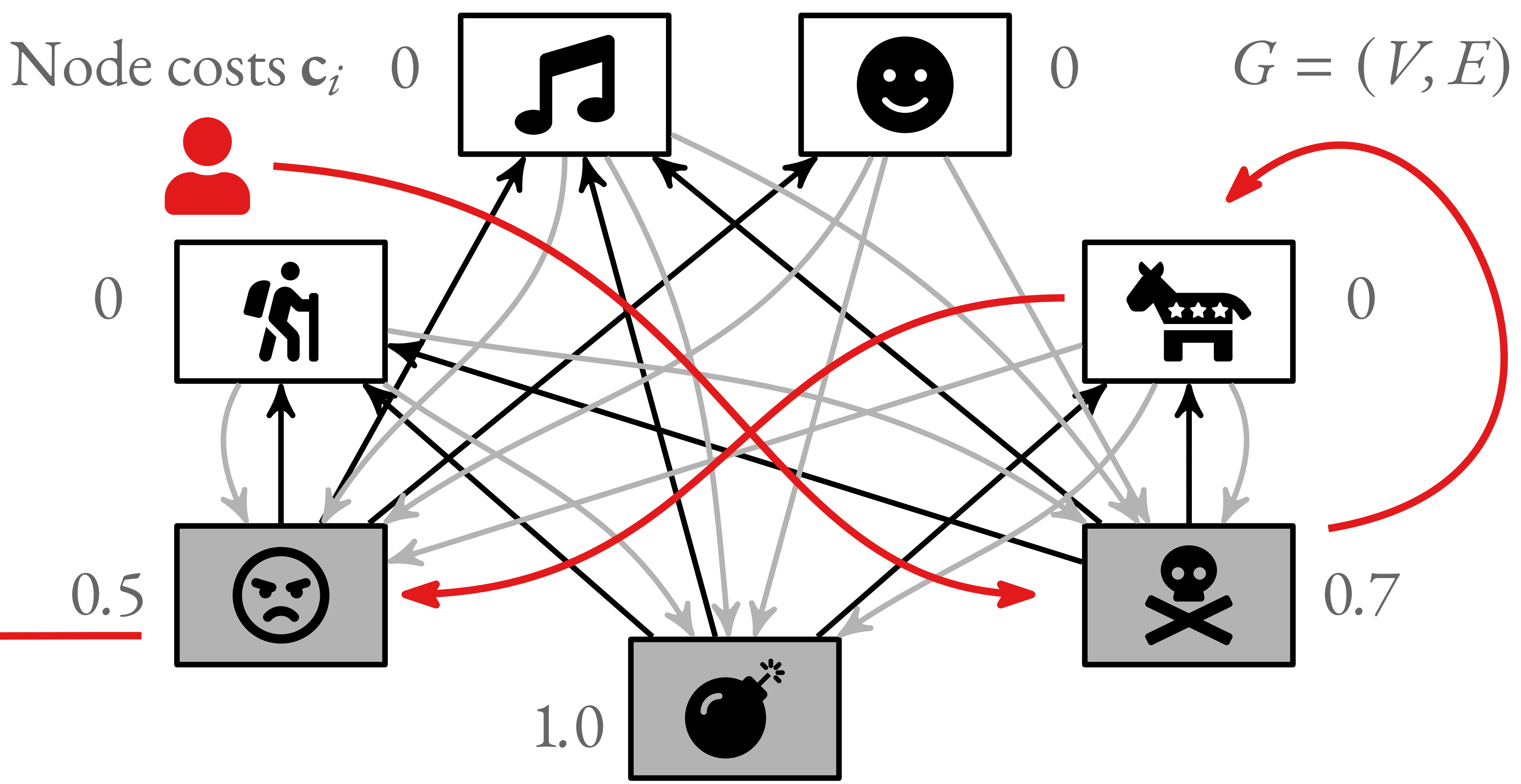
Model



Absorption probability α

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

Model

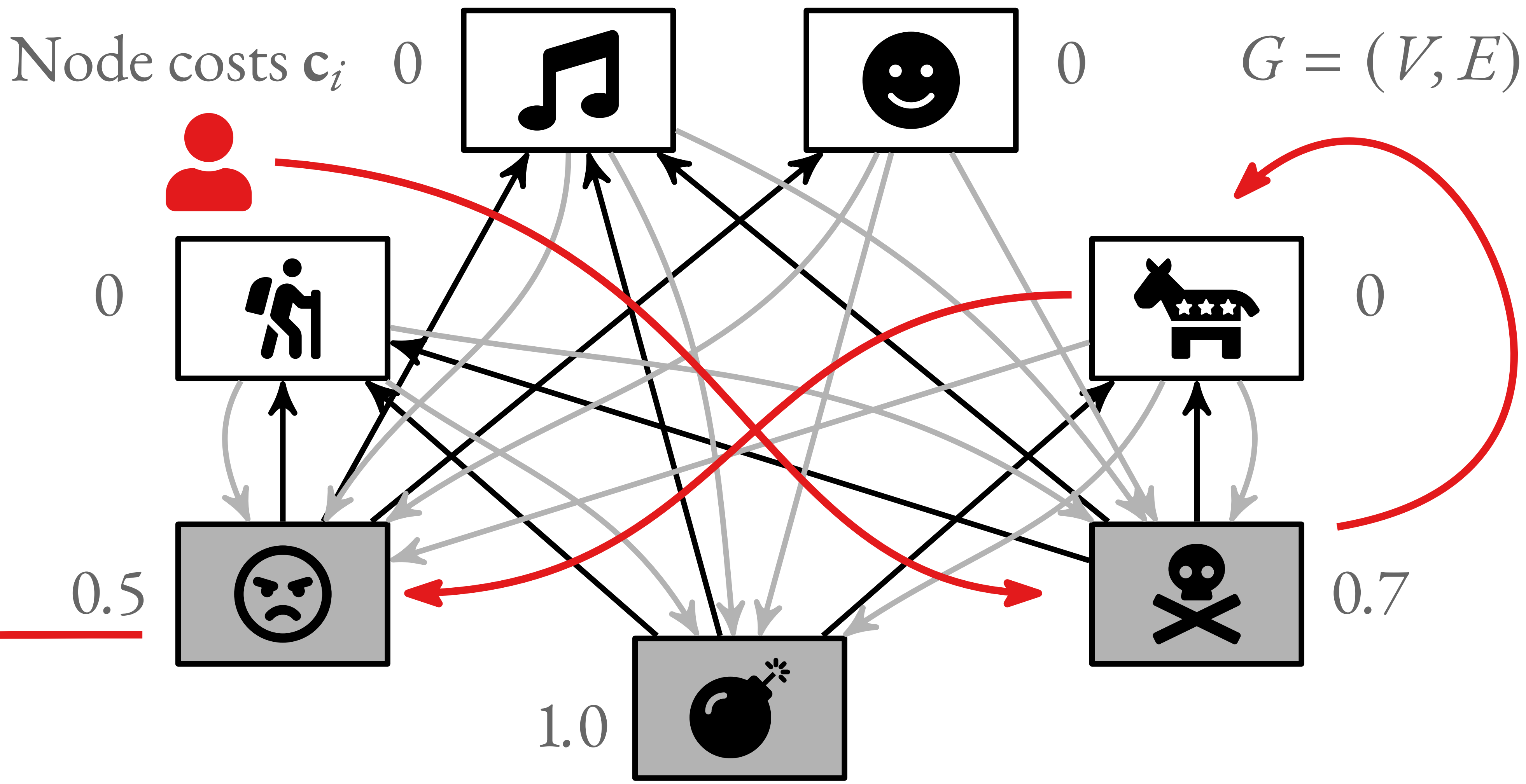


Absorption probability α

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

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

Model



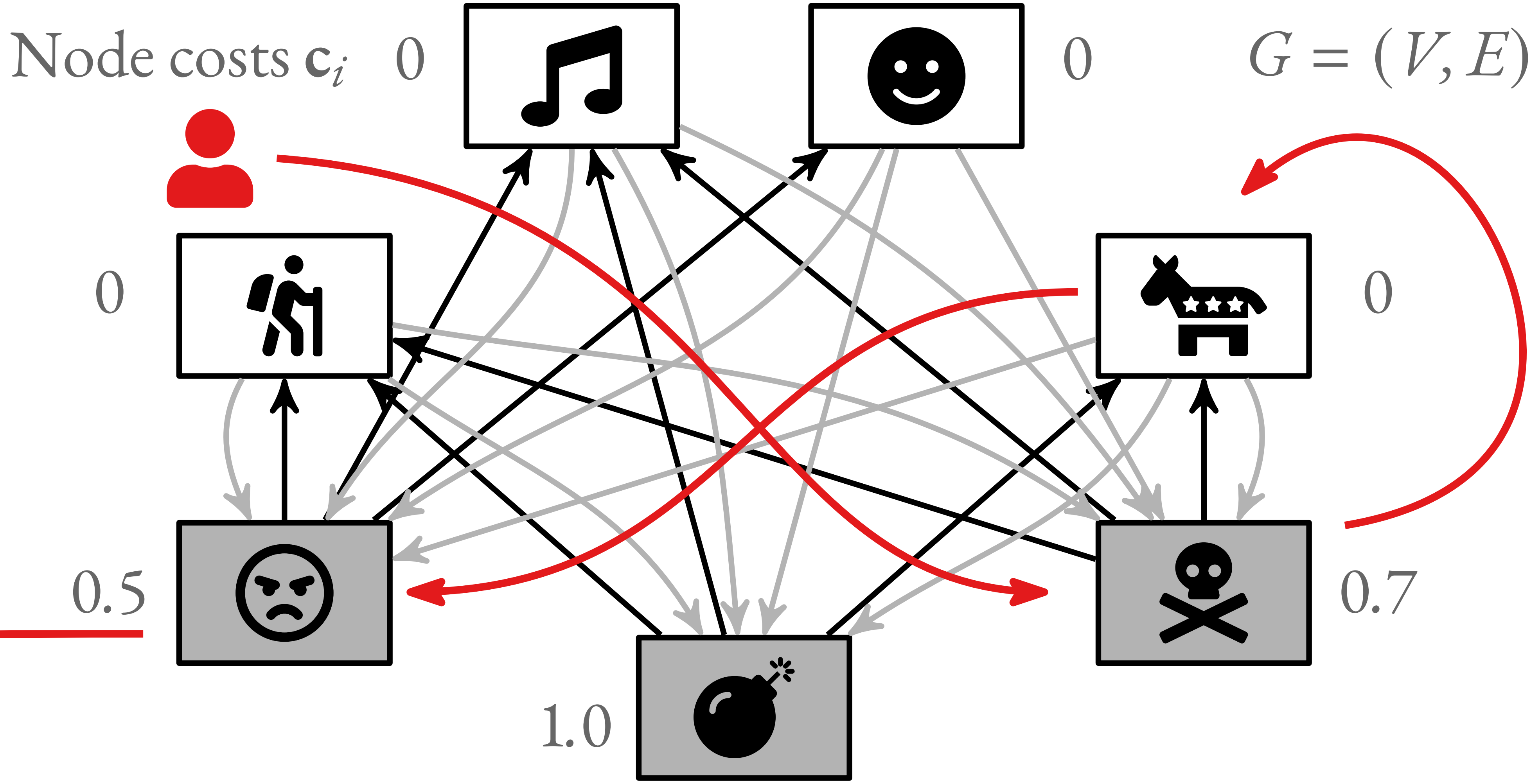
Absorption probability α

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



Absorption probability α

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

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

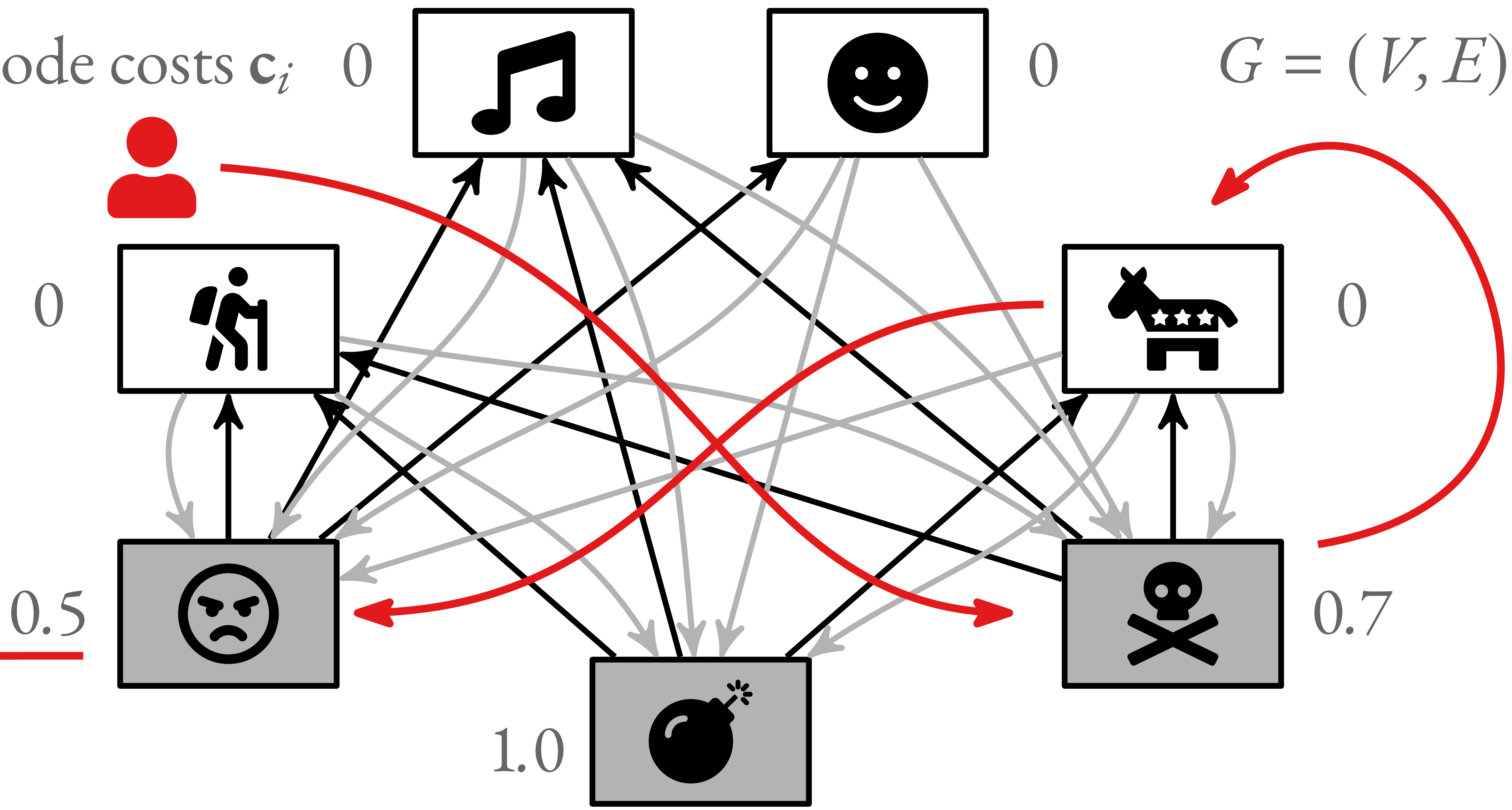
$\sum_{j \in V} \mathbf{F}_{ij}\mathbf{c}_j = \mathbf{e}_i^T \mathbf{F}\mathbf{c}$: Expected exposure starting at i

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

Model

Node costs $\mathbf{c}_i$

$G = (V, E)$



Absorption probability α

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 $f(G)$?

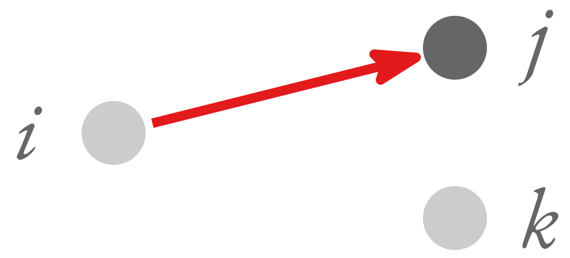
Problem Statement

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

Problem Statement

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

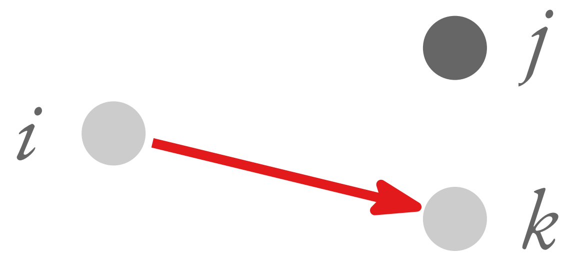
Edit operation: *Edge rewiring*



Problem Statement

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

Edit operation: *Edge rewiring*

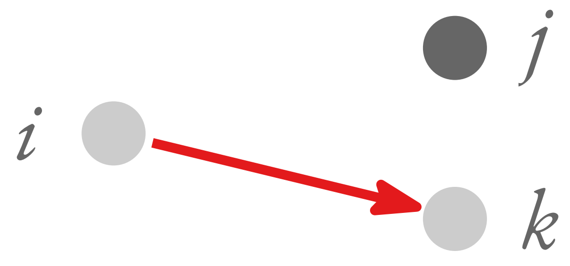


Problem Statement

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

G_r: G after r rewirings

Edit operation: *Edge rewiring*



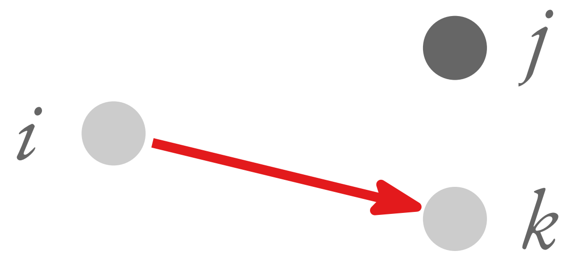
Problem Statement

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

G_r: G after r rewirings

r-Rewiring Exposure Minimization (REM)

Edit operation: *Edge rewiring*



Problem Statement

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

Edit operation: *Edge rewiring*

G_r: G after r rewirings

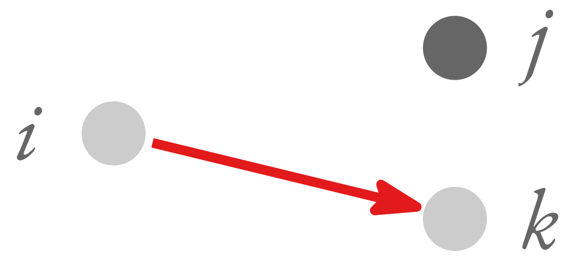
r-Rewiring Exposure Minimization (REM)

$$\min f(G_r)$$

Problem Statement

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

Edit operation: *Edge rewiring*



G_r: G after r rewirings

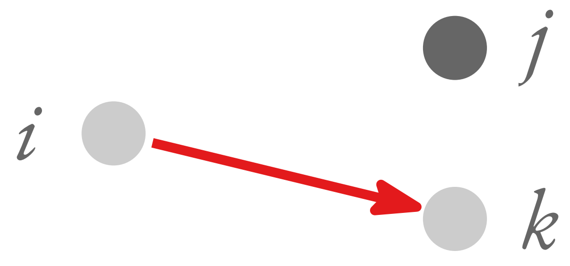
r-Rewiring Exposure Minimization (REM)

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

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: $\mathbf{1}^T \mathbf{Fc}$

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: $\mathbf{1}^T\mathbf{Fc}$

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 → 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 → 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 κ 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 κ 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)

$X \leftarrow \emptyset$
for $i \in \mathbb{N}_{\leq r}$ **do**
  Precompute $\mathbf{1}^T\mathbf{F}$ and $\mathbf{Fc}$ ▹ $O(\kappa m)$
  Precompute $\mathbf{1}^T\mathbf{Fu}$ for $(i, j) \in E$ ▹ $O(m)$
  Precompute $\mathbf{Fu}$ for $(i, j) \in E$ ▹ $O(n^2)$
  Precompute $\mathbf{v}^T\mathbf{Fc}$ for $j \neq k \in V$ ▹ $O(\kappa n^2)$
  1-REM()
return X

function 1-REM()
  $\Delta, i', j', k' \leftarrow 0, \perp, \perp, \perp$
  for $(i, j) \in E$ **do** ▹ $O(m)$
    for $k \in V \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▹ $O(n)$
      $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$
      $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$
      $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T\mathbf{Fu})(\mathbf{v}^T\mathbf{Fc})}{1+\mathbf{v}^T\mathbf{Fu}}$ ▹ $O(1)$
      if $\Delta_{ijk} > \Delta$ **then**
        $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$
  $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$
  $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$
  $\mathbf{P}[i', j'] \leftarrow 0$
  $X \leftarrow X \cup \{(i', j', k')\}$

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 κ 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)

$X \leftarrow \emptyset$
for $i \in \mathbb{N}_{\leq r}$ **do**
    Precompute $\mathbf{1}^T\mathbf{F}$ and $\mathbf{Fc}$ ▷ $O(\kappa m)$
    Precompute $\mathbf{1}^T\mathbf{Fu}$ for $(i, j) \in E$ ▷ $O(m)$
    Precompute $\mathbf{Fu}$ for $(i, j) \in E$ ▷ $O(n^2)$
    Precompute $\mathbf{v}^T\mathbf{Fc}$ for $j \neq k \in V$ ▷ $O(\kappa n^2)$
    1-REM()
return X

function 1-REM()
    $\Delta, i', j', k' \leftarrow 0, \bot, \bot, \bot$
    for $(i, j) \in E$ **do** ▷ $O(m)$
        for $k \in V \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▷ $O(n)$
            $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$
            $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$
            $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T\mathbf{Fu})(\mathbf{v}^T\mathbf{Fc})}{1+\mathbf{v}^T\mathbf{Fu}}$ ▷ $O(1)$
            if $\Delta_{ijk} > \Delta$ **then**
                $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$
    $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$
    $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$
    $\mathbf{P}[i', j'] \leftarrow 0$
    $X \leftarrow X \cup \{(i', j', k')\}$

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 κ 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)

$X \leftarrow \emptyset$
for $i \in \mathbb{N}_{\leq r}$ **do**
  Precompute $\mathbf{1}^T\mathbf{F}$ and $\mathbf{Fc}$ ▹ $O(\kappa m)$
  Precompute $\mathbf{1}^T\mathbf{Fu}$ for $(i, j) \in E$ ▹ $O(m)$
  Precompute $\mathbf{Fu}$ for $(i, j) \in E$ ▹ $O(n^2)$
  Precompute $\mathbf{v}^T\mathbf{Fc}$ for $j \neq k \in V$ ▹ $O(\kappa n^2)$
  1-REM()
return X

function 1-REM()
  $\Delta, i', j', k' \leftarrow 0, \bot, \bot, \bot$
  for $(i, j) \in E$ **do** ▹ $O(m)$
    for $k \in V \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▹ $O(n)$
      $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$
      $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$
      $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T\mathbf{Fu})(\mathbf{v}^T\mathbf{Fc})}{1+\mathbf{v}^T\mathbf{Fu}}$ ▹ $O(1)$
      if $\Delta_{ijk} > \Delta$ **then**
        $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$
  $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$
  $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$
  $\mathbf{P}[i', j'] \leftarrow 0$
  $X \leftarrow X \cup \{(i', j', k')\}$

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 κ 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)

$X \leftarrow \emptyset$
for $i \in \mathbb{N}_{\leq r}$ **do**
  Precompute $\mathbf{1}^T\mathbf{F}$ and $\mathbf{Fc}$ ▷ $O(\kappa m)$
  Precompute $\mathbf{1}^T\mathbf{Fu}$ for $(i, j) \in E$ ▷ $O(m)$
  Precompute $\mathbf{Fu}$ for $(i, j) \in E$ ▷ $O(n^2)$
  Precompute $\mathbf{v}^T\mathbf{Fc}$ for $j \neq k \in V$ ▷ $O(\kappa n^2)$
  1-REM()
return X

function 1-REM()
  $\Delta, i', j', k' \leftarrow 0, \bot, \bot, \bot$
  for $(i, j) \in E$ **do** ▷ $O(m)$
    for $k \in V \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▷ $O(n)$
      $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$
      $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$
      $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T\mathbf{Fu})(\mathbf{v}^T\mathbf{Fc})}{1+\mathbf{v}^T\mathbf{Fu}}$ ▷ $O(1)$
      if $\Delta_{ijk} > \Delta$ **then**
        $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$
  $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$
  $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$
  $\mathbf{P}[i', j'] \leftarrow 0$
  $X \leftarrow X \cup \{(i', j', k')\}$

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 κ 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)

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

function 1-REM()
  $\Delta, i', j', k' \leftarrow 0, \bot, \bot, \bot$
  for $(i, j) \in E$ **do** ▹ $O(m)$
    for $k \in V \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▹ $O(n)$
      $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$
      $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$
      $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T\mathbf{F}\mathbf{u})(\mathbf{v}^T\mathbf{F}\mathbf{c})}{1+\mathbf{v}^T\mathbf{F}\mathbf{u}}$ ▹ $O(1)$
      if $\Delta_{ijk} > \Delta$ **then**
        $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$
  $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$
  $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$
  $\mathbf{P}[i', j'] \leftarrow 0$
  $X \leftarrow X \cup \{(i', j', k')\}$

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 κ 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)

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

function 1-REM()
    $\Delta, i', j', k' \leftarrow 0, \bot, \bot, \bot$
    for $(i, j) \in E$ **do** ▹ $O(m)$
        for $k \in V \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▹ $O(n)$
            $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$
            $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$
            $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T\mathbf{F}\mathbf{u})(\mathbf{v}^T\mathbf{F}\mathbf{c})}{1+\mathbf{v}^T\mathbf{F}\mathbf{u}}$ ▹ $O(1)$
            if $\Delta_{ijk} > \Delta$ **then**
                $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$
    $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$
    $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$
    $\mathbf{P}[i', j'] \leftarrow 0$
    $X \leftarrow X \cup \{(i', j', k')\}$

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 κ 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...
  - Correlation between $\sigma\tau$ and $\sigma\tau/\varrho$ 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)

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

function 1-REM()
    $\Delta, i', j', k' \leftarrow 0, \bot, \bot, \bot$
    for $(i, j) \in E$ **do** ▷ $O(m)$
        for $k \in V \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▷ $O(n)$
            $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$
            $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$
            $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T\mathbf{F}\mathbf{u})(\mathbf{v}^T\mathbf{F}\mathbf{c})}{1+\mathbf{v}^T\mathbf{F}\mathbf{u}}$ ▷ $O(1)$
            if $\Delta_{ijk} > \Delta$ **then**
                $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$
    $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$
    $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$
    $\mathbf{P}[i', j'] \leftarrow 0$
    $X \leftarrow X \cup \{(i', j', k')\}$

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 κ 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. . .

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

$\sigma\tau > \sigma\tau'$ *almost* always implies $\sigma\tau/\varrho > \sigma\tau'/\varrho'$

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)

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

function 1-REM()
  $\Delta, i', j', k' \leftarrow 0, \bot, \bot, \bot$
  for $(i, j) \in E$ **do** ▷ $O(m)$
    for $k \in V \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▷ $O(n)$
      $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$
      $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_k$
      $\Delta_{ijk} \leftarrow \frac{(\mathbf{1}^T\mathbf{F}\mathbf{u})(\mathbf{v}^T\mathbf{F}\mathbf{c})}{1+\mathbf{v}^T\mathbf{F}\mathbf{u}}$ ▷ $O(1)$
      if $\Delta_{ijk} > \Delta$ **then**
        $\Delta, i', j', k' \leftarrow \Delta_{ijk}, i, j, k$
  $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$
  $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$
  $\mathbf{P}[i', j'] \leftarrow 0$
  $X \leftarrow X \cup \{(i', j', k')\}$

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 κ 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…
  - Correlation between $\sigma\tau$ and $\sigma\tau/\varrho$ almost perfect
  - $\sigma\tau > \sigma\tau'$ *almost* always implies $\sigma\tau/\varrho > \sigma\tau'/\varrho'$

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)

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

function 1-REM()
  $\hat{\Delta}, i', j', k' \leftarrow 0, \bot, \bot, \bot$
  for $(i,j) \in E$ **do** ▹ $O(m)$
    for $k_{ij} \in K \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▹ $O(\Delta^+)$
      $\mathbf{u} \leftarrow \mathbf{P}[i,j]\mathbf{e}_i$
      $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_{k_{ij}}$
      $\hat{\Delta}_{ijk} \leftarrow (\mathbf{1}^T\mathbf{F}\mathbf{u})(\mathbf{v}^T\mathbf{F}\mathbf{c})$ ▹ $O(1)$
      if $\hat{\Delta}_{ijk} > \hat{\Delta}$ **then**
        $\hat{\Delta}, i', j', k' \leftarrow \hat{\Delta}_{ijk}, i, j, k_{ij}$
  $E \leftarrow (E \setminus \{(i',j')\}) \cup \{(i',k')\}$
  $\mathbf{P}[i',k'] \leftarrow \mathbf{P}[i',j']$
  $\mathbf{P}[i',j'] \leftarrow 0$
  $X \leftarrow X \cup \{(i',j',k')\}$

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 κ 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. . .

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

$\sigma\tau > \sigma\tau'$ *almost* always implies $\sigma\tau/\varrho > \sigma\tau'/\varrho'$

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)

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

function 1-REM( )
  $\hat{\Delta}, i', j', k' \leftarrow 0, \bot, \bot, \bot$
  for $(i, j) \in E$ **do** ▷ $O(m)$
    for $k_{ij} \in K \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▷ $O(\Delta^+)$
      $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$
      $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_{k_{ij}}$
      $\hat{\Delta}_{ijk} \leftarrow (\mathbf{1}^T\mathbf{F}\mathbf{u})(\mathbf{v}^T\mathbf{F}\mathbf{c})$ ▷ $O(1)$
      if $\hat{\Delta}_{ijk} > \hat{\Delta}$ **then**
        $\hat{\Delta}, i', j', k' \leftarrow \hat{\Delta}_{ijk}, i, j, k_{ij}$
  $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$
  $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$
  $\mathbf{P}[i', j'] \leftarrow 0$
  $X \leftarrow X \cup \{(i', j', k')\}$

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 κ 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...
  - Correlation between $\sigma\tau$ and $\sigma\tau/\varrho$ almost perfect
  - $\sigma\tau > \sigma\tau'$ *almost* always implies $\sigma\tau/\varrho > \sigma\tau'/\varrho'$

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

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

function 1-REM()
  $\hat{\Delta}, i', j', k' \leftarrow 0, \bot, \bot, \bot$
  for $(i, j) \in E$ **do** ▷ $O(m)$
    for $k_{ij} \in K \setminus (\Gamma^+(i) \cup \{i\})$ **do** ▷ $O(\Delta^+)$
      $\mathbf{u} \leftarrow \mathbf{P}[i, j]\mathbf{e}_i$
      $\mathbf{v} \leftarrow \mathbf{e}_j - \mathbf{e}_{k_{ij}}$
      $\hat{\Delta}_{ijk} \leftarrow (\mathbf{1}^T\mathbf{F}\mathbf{u})(\mathbf{v}^T\mathbf{F}\mathbf{c})$ ▷ $O(1)$
      if $\hat{\Delta}_{ijk} > \hat{\Delta}$ **then**
        $\hat{\Delta}, i', j', k' \leftarrow \hat{\Delta}_{ijk}, i, j, k_{ij}$
  $E \leftarrow (E \setminus \{(i', j')\}) \cup \{(i', k')\}$
  $\mathbf{P}[i', k'] \leftarrow \mathbf{P}[i', j']$
  $\mathbf{P}[i', j'] \leftarrow 0$
  $X \leftarrow X \cup \{(i', j', k')\}$

Experimental Evaluation

Experimental Evaluation

Dataset	d	n	m	$f(G)/n$
SU, SH (2·4·36 graphs)	5	10^i for $i \in \{2, 3, 4, 5\}$	5×10^i	[1.291, 15.231]
YT-100K (3 · 6 graphs)	5	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-COV19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		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	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-Jan06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-All (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		934 550	[5.315, 11.660]
	20		1 869 100	[5.138, 11.517]

Synthetic data

Experimental Evaluation

Dataset	d	n	m	$f(G)/n$
SU, SH (2·4·36 graphs)	5	10^i	$5{\times}10^i$ for $i \in \{2, 3, 4, 5\}$	[1.291, 15.231]
YT-100K (3 · 6 graphs)	5	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-COV19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		934 550	[5.315, 11.660]
	20		1 869 100	[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	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-COV19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		934 550	[5.315, 11.660]
	20		1 869 100	[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	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-COV19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		934 550	[5.315, 11.660]
	20		1 869 100	[5.138, 11.517]

Synthetic data

YouTube video data

NELA-GT news data

Competition

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	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-Jan06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-All (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		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)

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	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-COV19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		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

Experimental Evaluation

Dataset	d	n	m	$f(G)/n$
SU, SH (2·4·36 graphs)	5	10^i for $i \in \{2, 3, 4, 5\}$	5×10^i	[1.291, 15.231]
YT-100K (3 · 6 graphs)	5	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-Jan06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-All (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		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

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	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-Jan06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-Cov19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-All (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		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

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	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-COV19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		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

Experimental Evaluation

Dataset	d	n	m	$f(G)/n$
SU, SH (2·4·36 graphs)	5	10^i for $i \in \{2, 3, 4, 5\}$	5×10^i	[1.291, 15.231]
YT-100K (3 · 6 graphs)	5	40 415	202 075	[0.900, 8.475]
	10		404 150	[0.938, 8.701]
	20		808 300	[0.989, 9.444]
YT-10K (3 · 6 graphs)	5	150 572	752 860	[0.806, 5.785]
	10		1 505 720	[0.883, 7.576]
	20		3 011 440	[0.949, 8.987]
NF-JAN06 (3 · 6 graphs)	5	11 931	59 655	[4.217, 9.533]
	10		119 310	[4.248, 9.567]
	20		238 620	[4.217, 9.533]
NF-COV19 (3 · 6 graphs)	5	57 447	287 235	[4.609, 11.068]
	10		574 470	[4.392, 10.769]
	20		1 148 940	[4.329, 10.741]
NF-ALL (3 · 6 graphs)	5	93 455	467 275	[5.565, 11.896]
	10		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

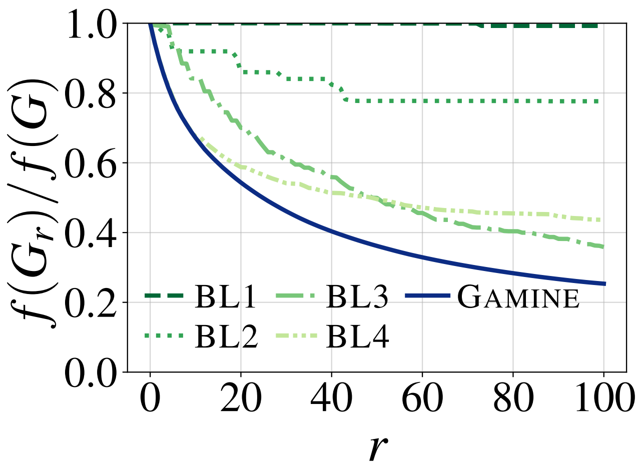
Python 3.10

Commodity hardware

Basic parallelization

Gamine…

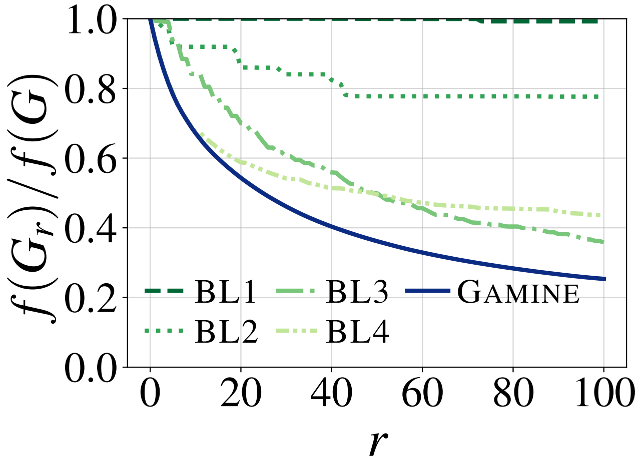
Gamine...

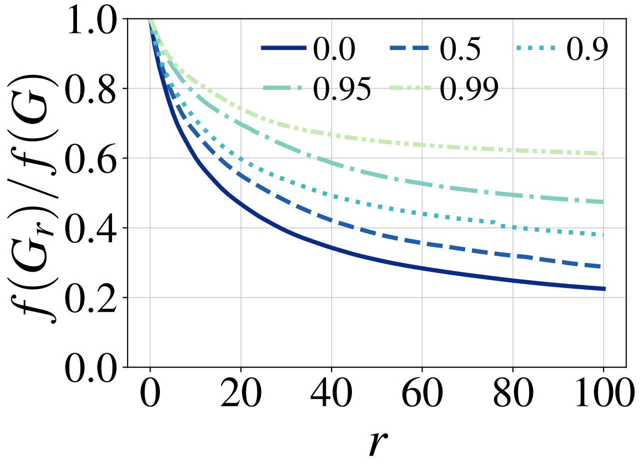


...outperforms all *baselines*

Gamine...

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

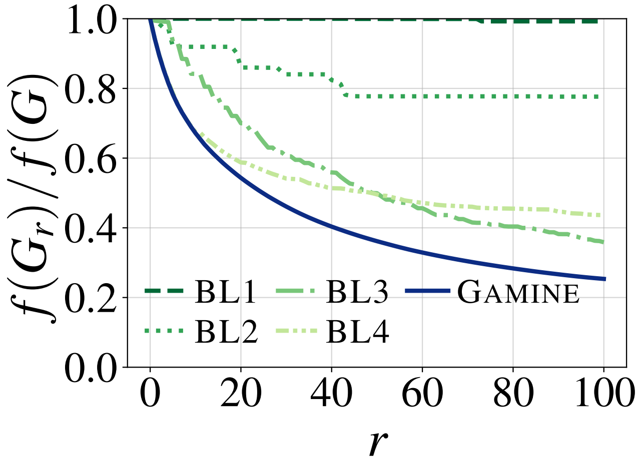


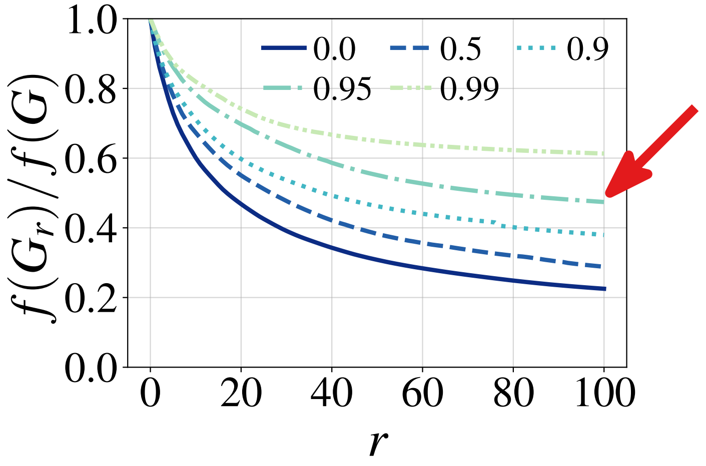


...outperforms all *baselines*

GAMINE…

…is robust to changes in the *quality threshold*





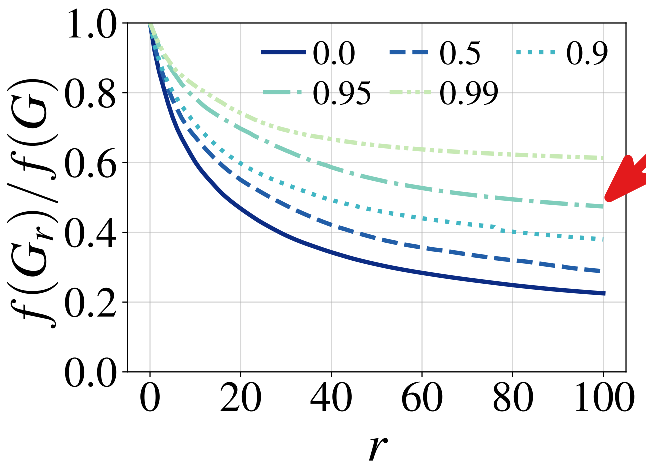
50% exposure reduction

5% drop in recommendation quality

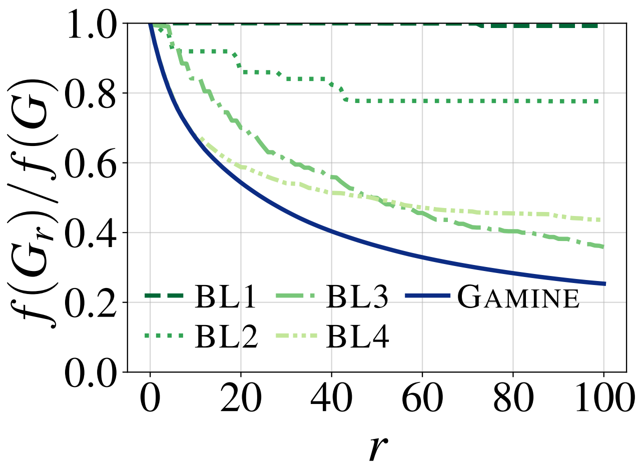
…outperforms all *baselines*

Gamine...

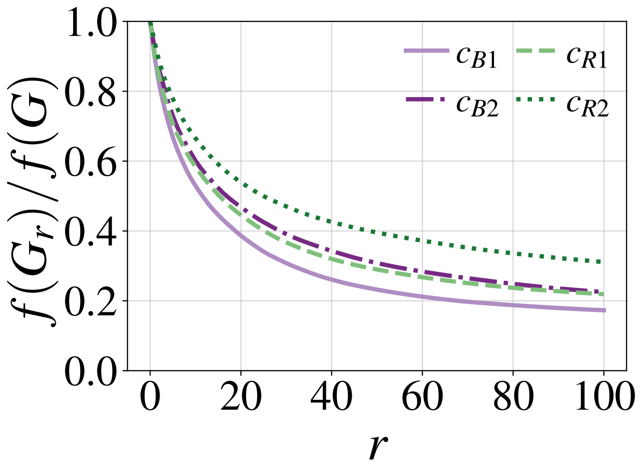
...is robust to changes in the ***quality threshold***



50% exposure reduction
5% drop in recommendation quality



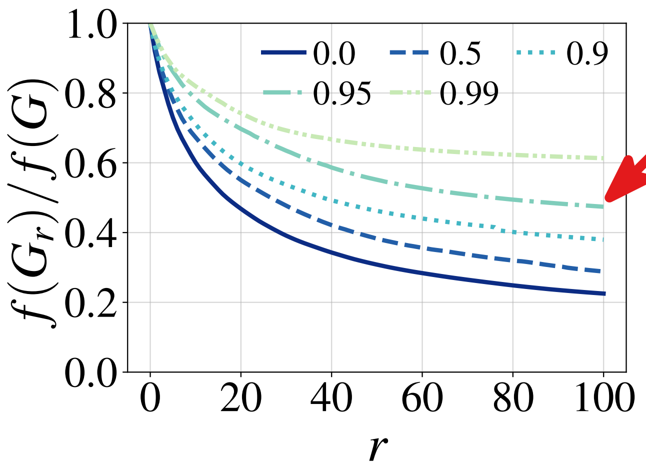
...outperforms all *baselines*



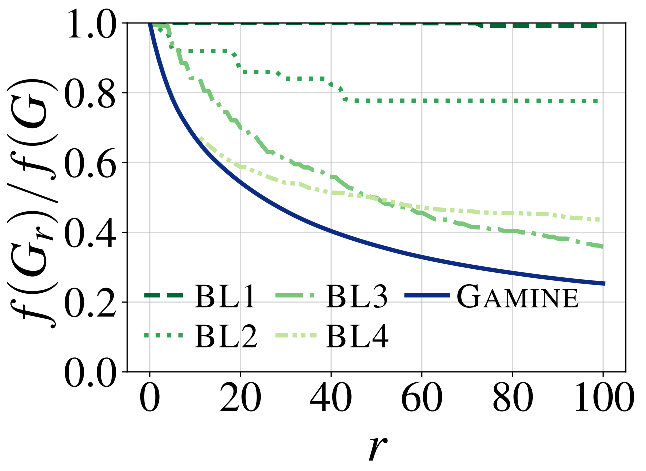
...is robust to changes in the *cost function*

Gamine…

…is robust to changes in the *quality threshold*

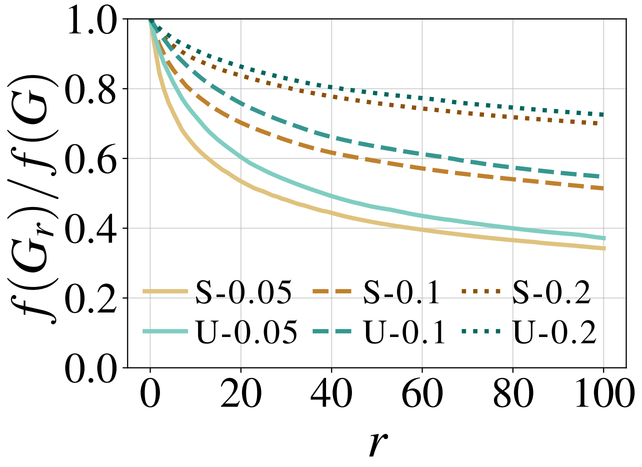


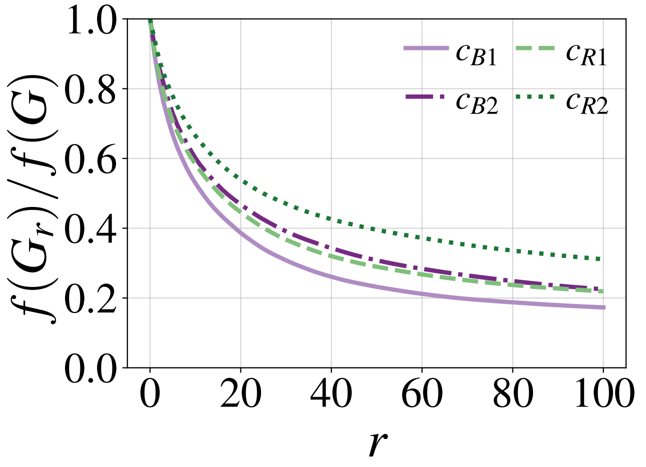
50% exposure reduction
5% drop in recommendation quality



…outperforms all *baselines*

…is robust to changes in the *user model*

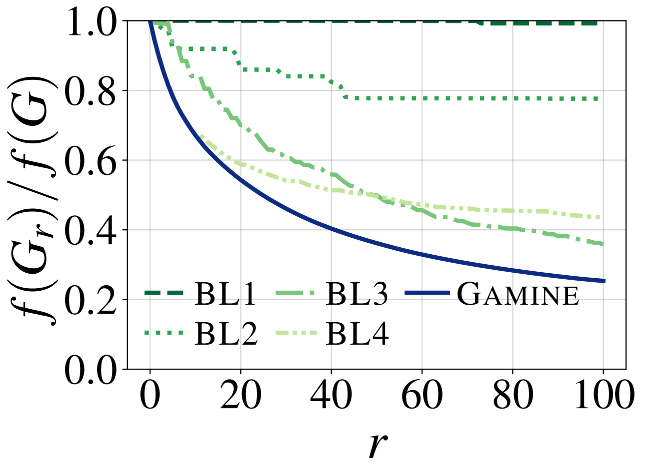


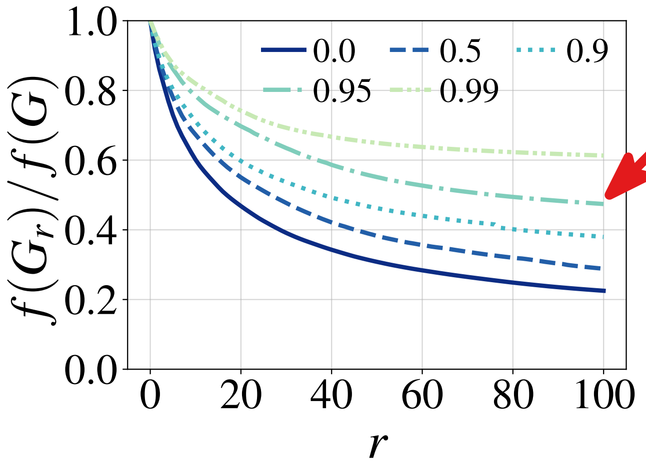


…is robust to changes in the *cost function*

GAMINE…

…is robust to changes in the *quality threshold*



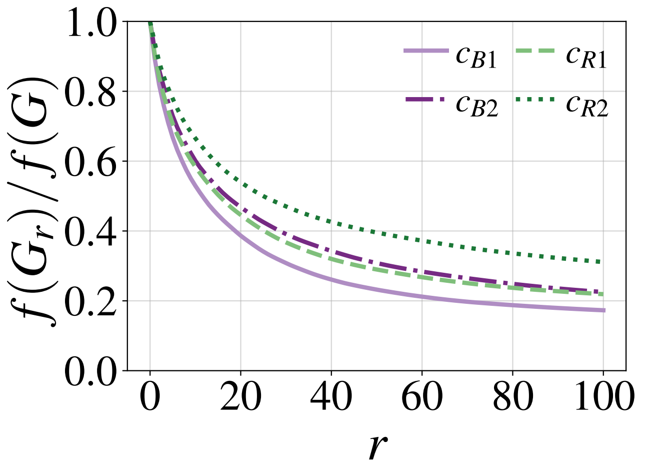


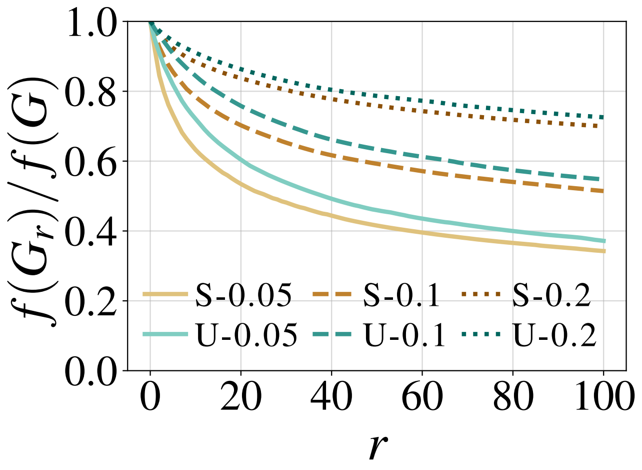
50% exposure reduction

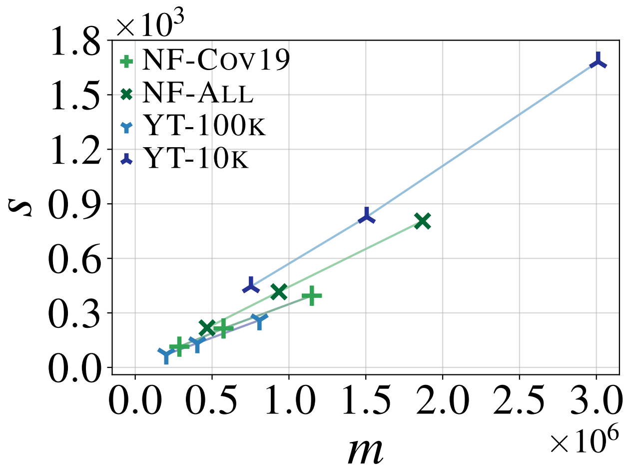
5% drop in recommendation quality

…outperforms all *baselines*

…is robust to changes in the *user model*



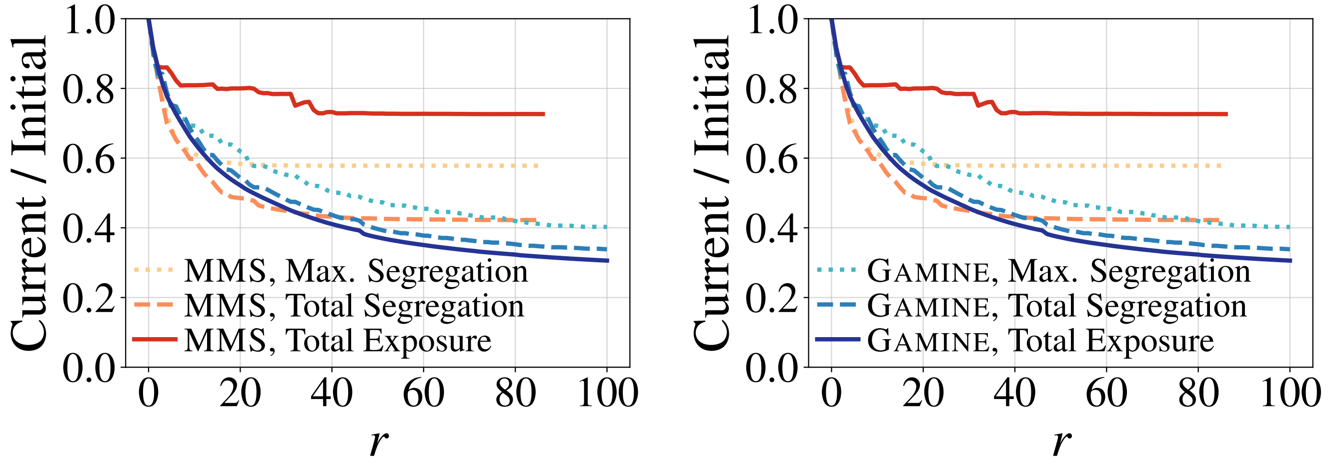




…is robust to changes in the *cost function*

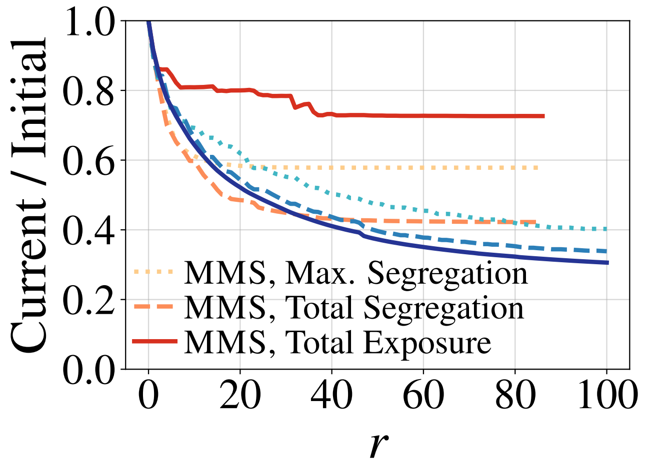
…and scales linearly

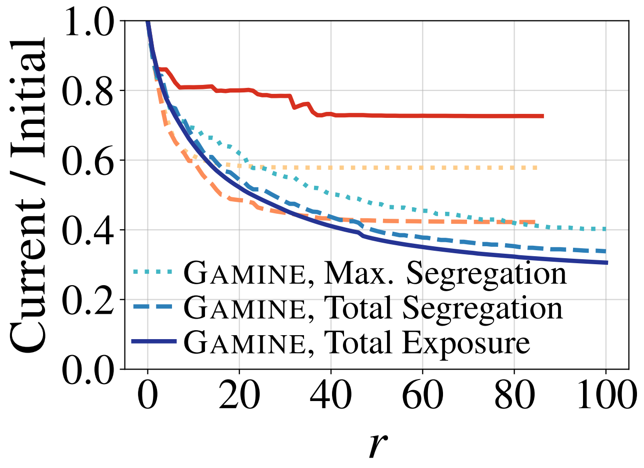
Gamine…



…outperforms MMS on the video data and…

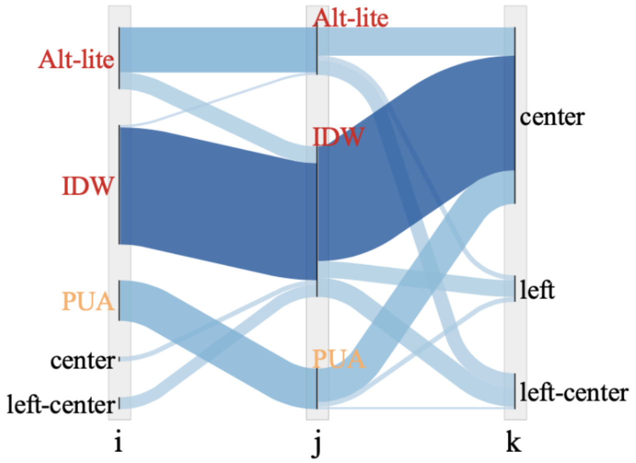
Gamine...

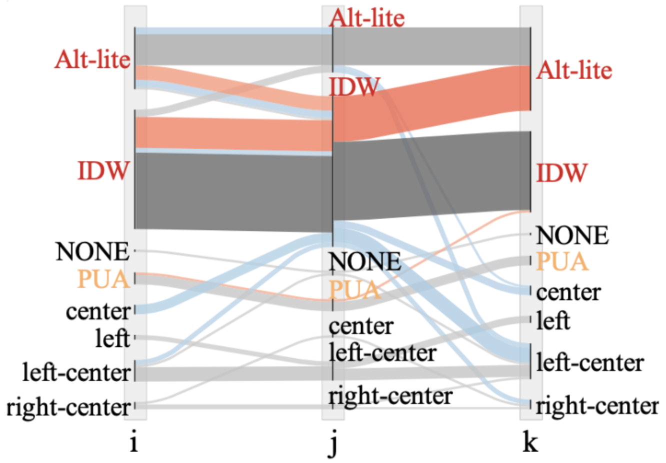




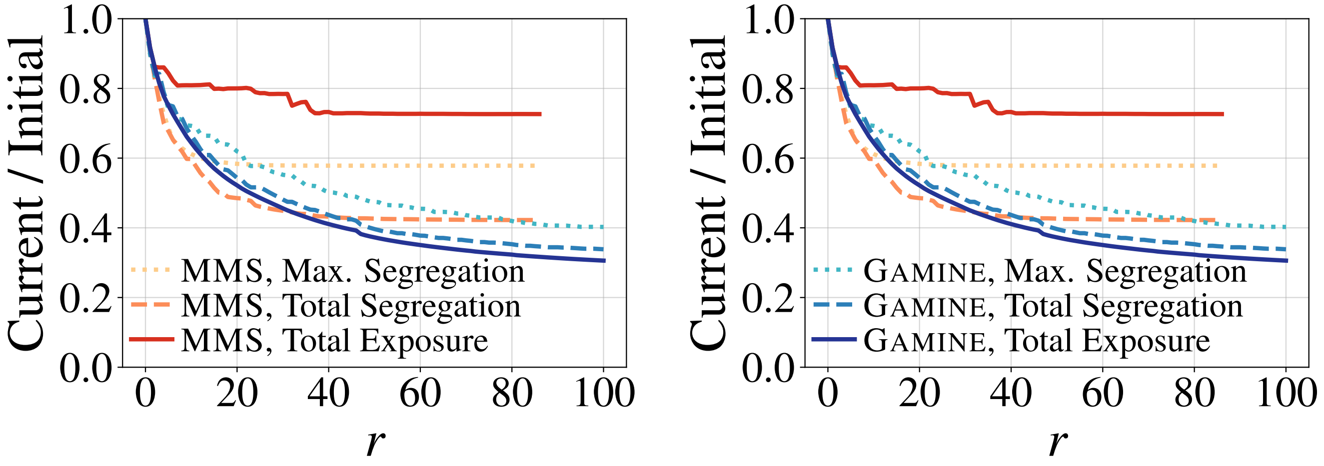
...outperforms MMS on the video data and...

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

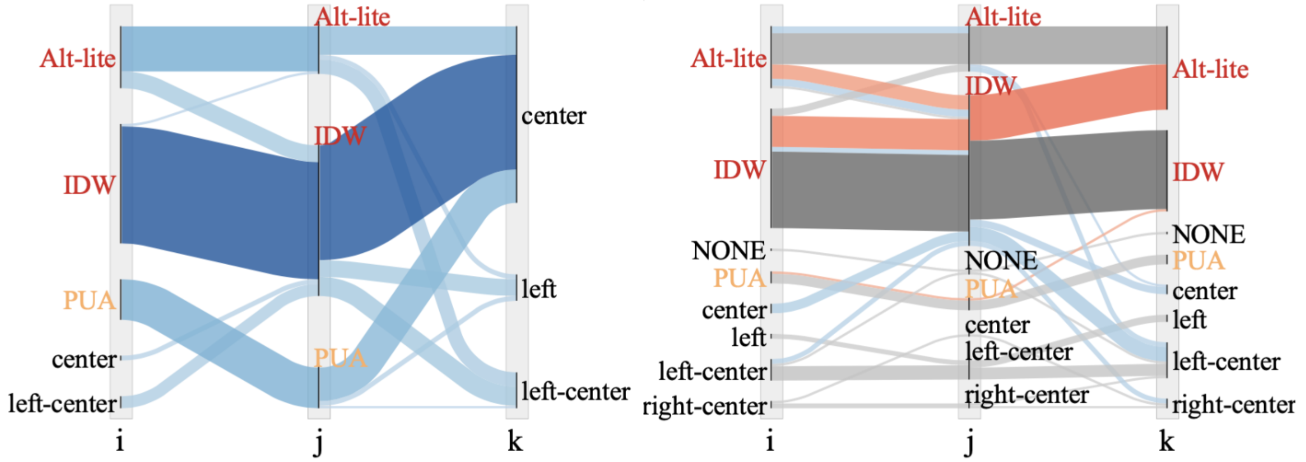




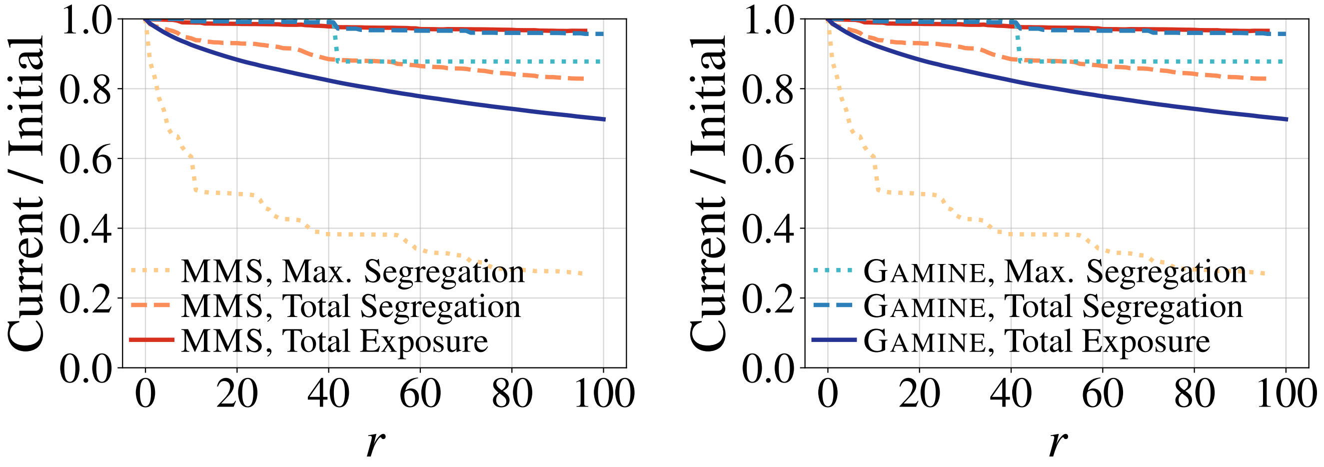
GAMINE...



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

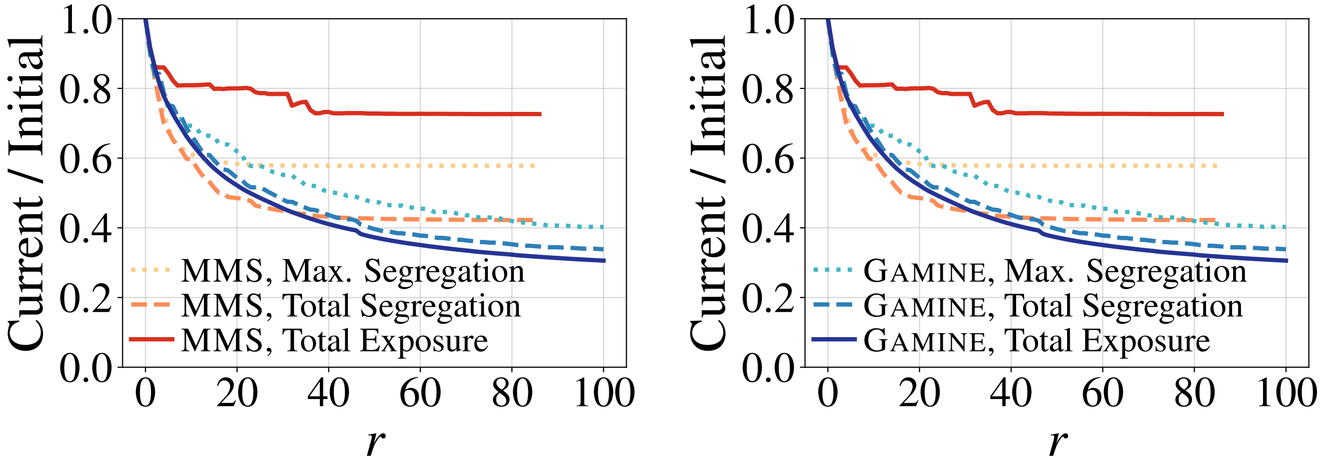


...outperforms MMS on the video data and...

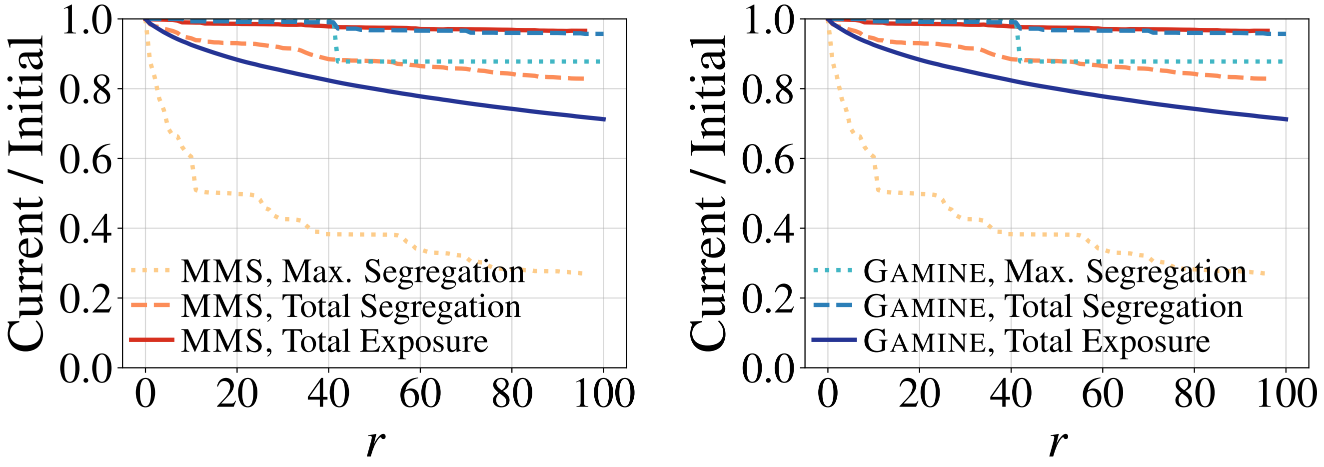


...it doesn't do nearly as well on the news data...

GAMINE…

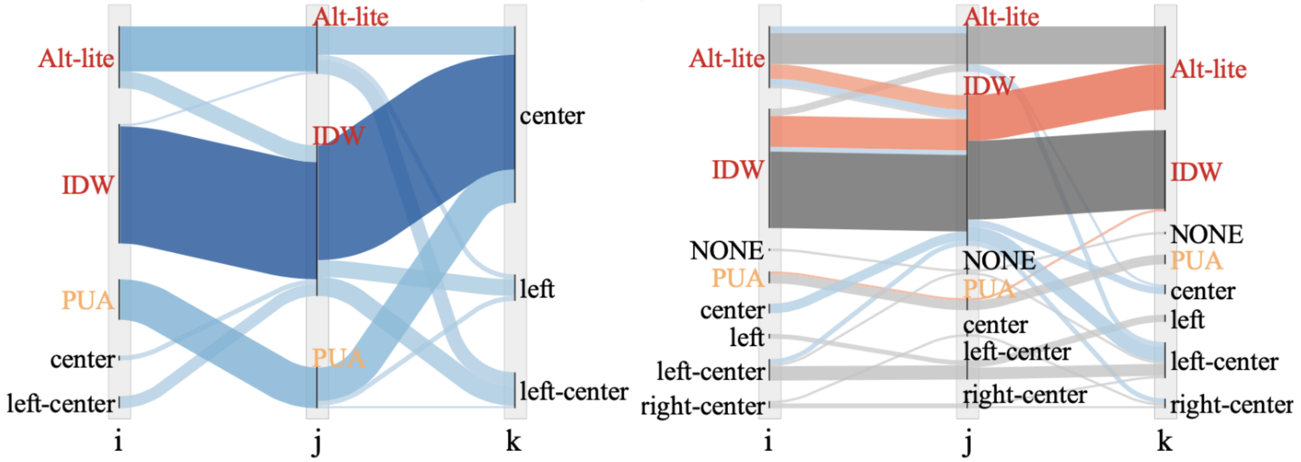


…outperforms MMS on the video data and…

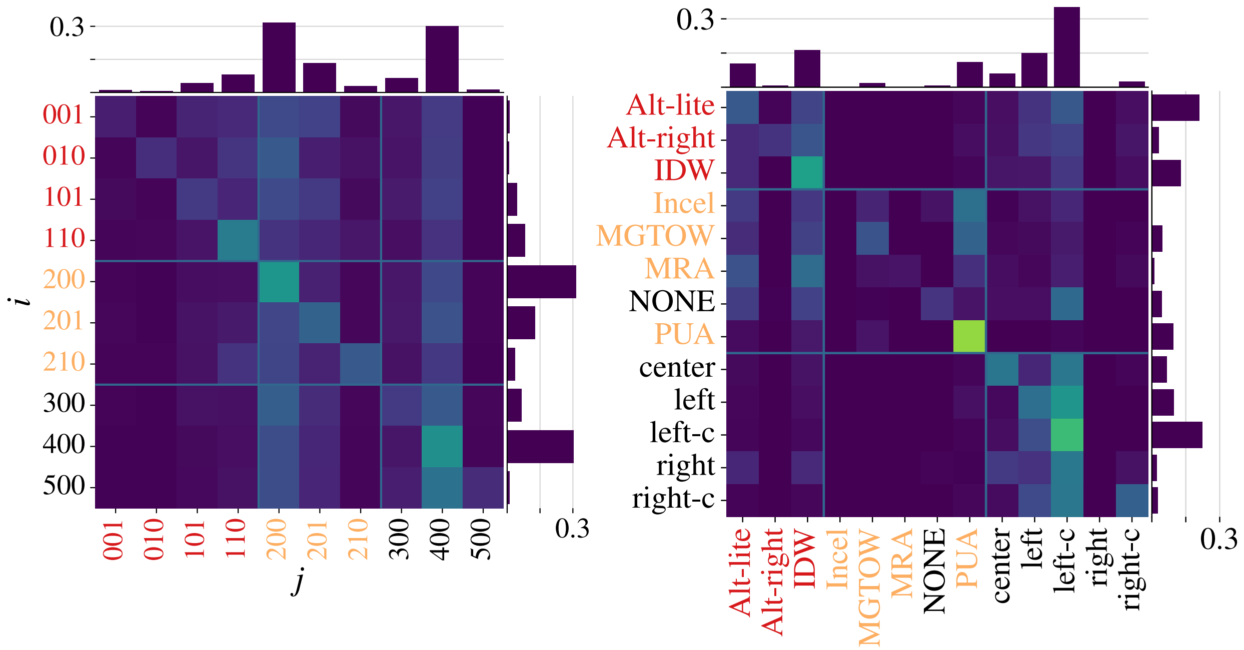


…it doesn't do nearly as well on the news data…

…generates *interpretable* rewirings, but…



…because the news data is more *complex*!



Summary

Summary

Real-world problem + modeling assumptions → r-rewiring exposure minimization (REM)

Summary

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

Summary

Real-world problem + modeling assumptions → 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 $\longrightarrow$ 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 → 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!

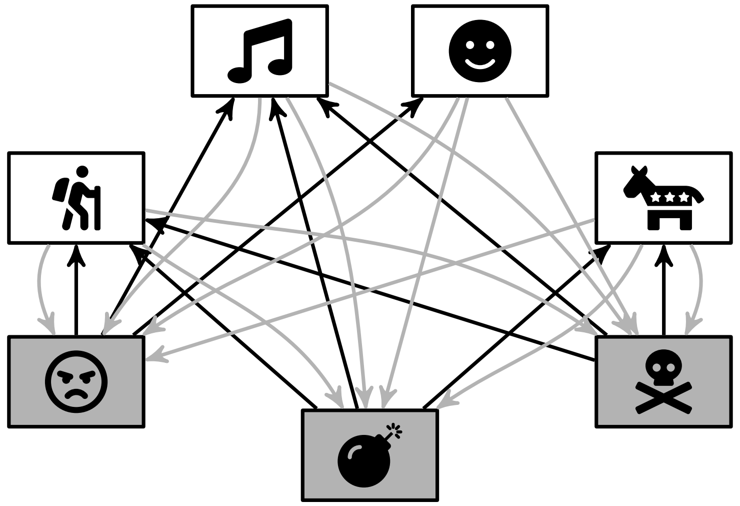
Real-world problem + modeling assumptions → r-rewiring exposure minimization (REM)
= Optimizing random walks via graph editing under budget constraints

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

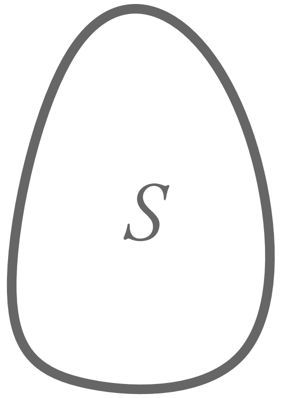
Conditional Submodularity

Partition V into two classes

Conditional Submodularity

Partition V into two classes

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

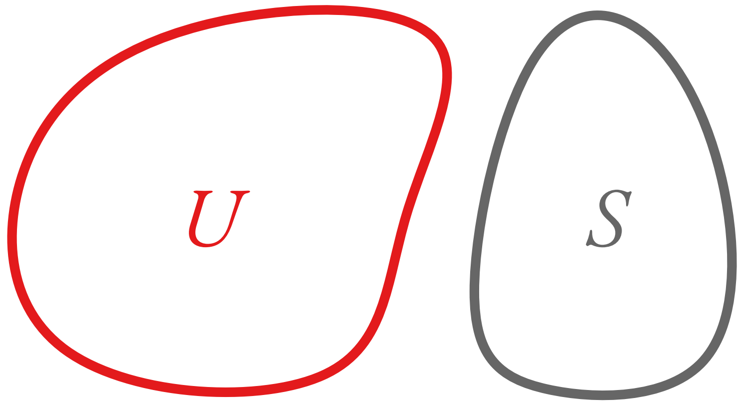


Conditional Submodularity

Partition V into two classes

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

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



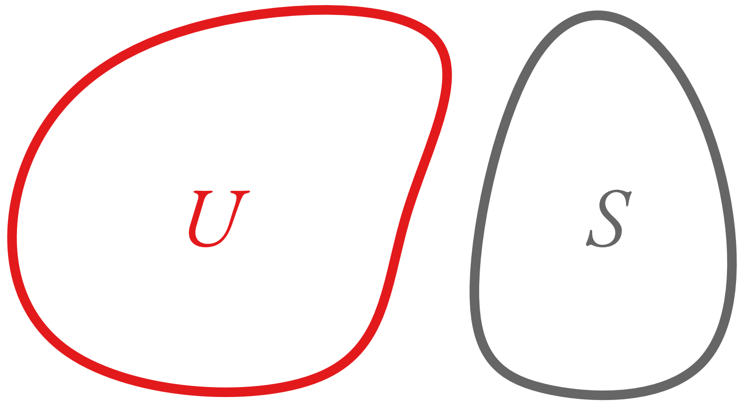
Conditional Submodularity

Partition V into two classes

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

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

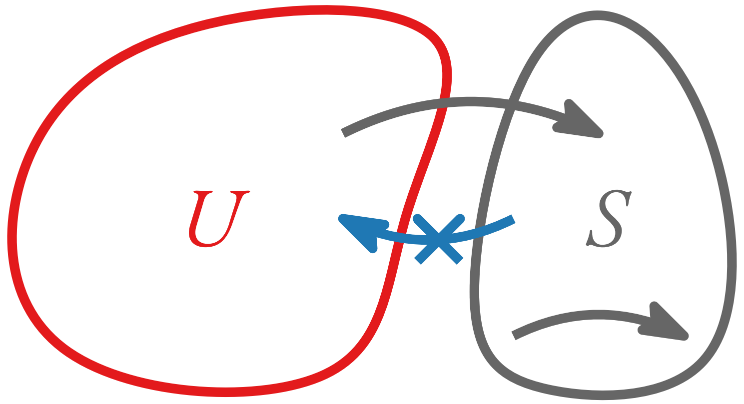
Partition V into two classes

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

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

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

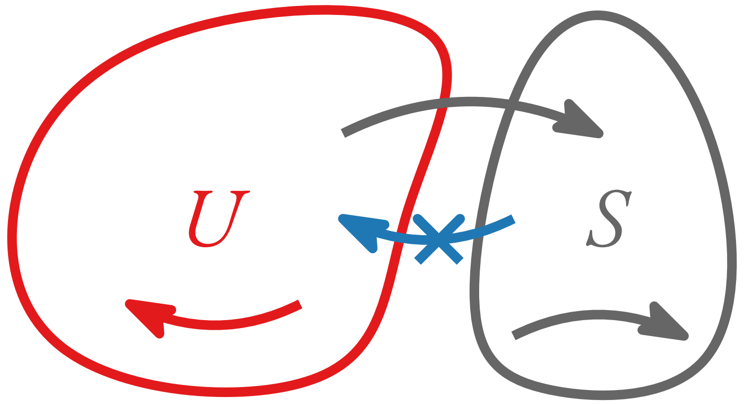


Conditional Submodularity

Partition V into two classes

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

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

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

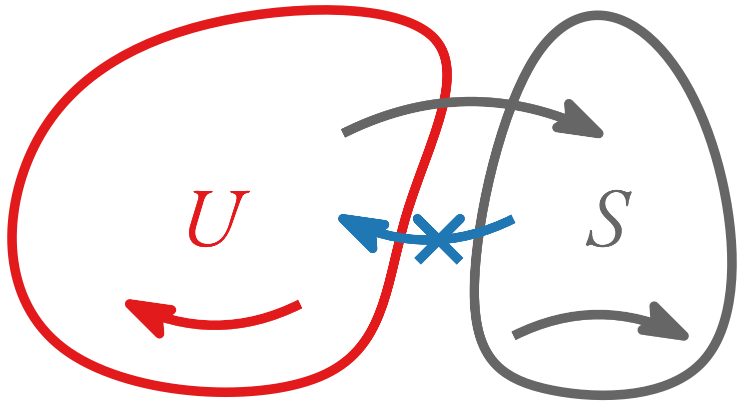
Only problematic class: $E_{UU} = \{(i, j) \mid i \in U \wedge j \in U\}$

Conditional Submodularity

Partition V into two classes

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

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



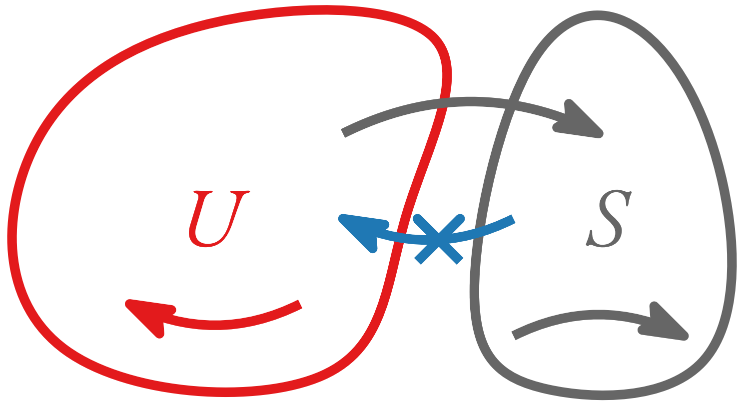
Partition E into four classes based on S and U

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

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

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

Conditional Submodularity



Partition V into two classes

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

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

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

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

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

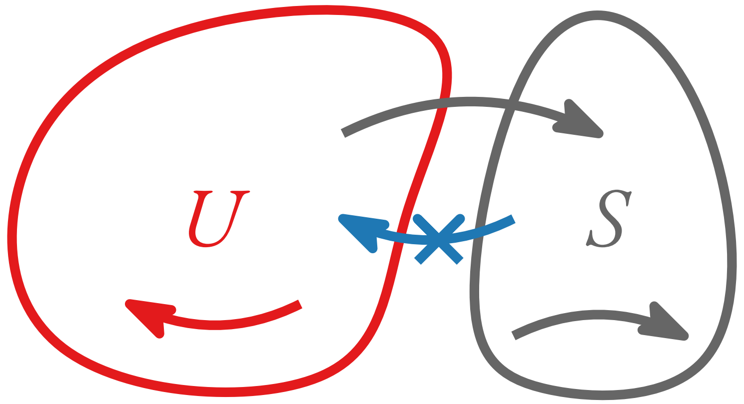
If $|S| \geq \Lambda^+$, then $\max f_\Delta(G, G_r)$ is equivalent to maximizing submodular function over E_{UU}

Conditional Submodularity

Partition V into two classes

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

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

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

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}

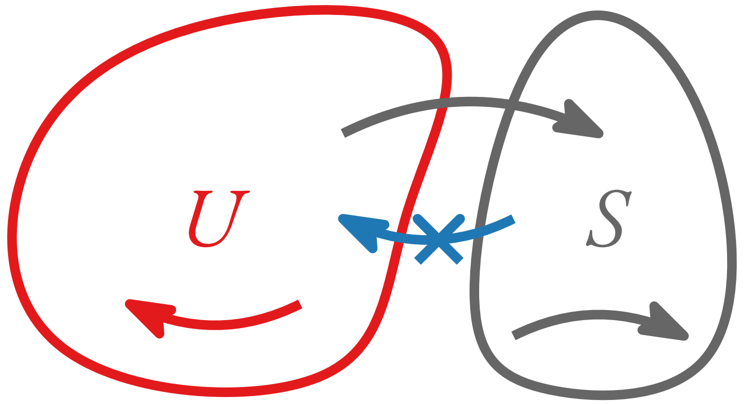
"G has the safety property"

Conditional Submodularity

Partition V into two classes

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

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



Partition E into four classes based on S and U

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

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

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 ↔ 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 $\mathbf{P}$

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

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

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

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

Greedy Rewiring

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

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

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

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

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

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

Greedy Rewiring

Greedy step ↔ 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]

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

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

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

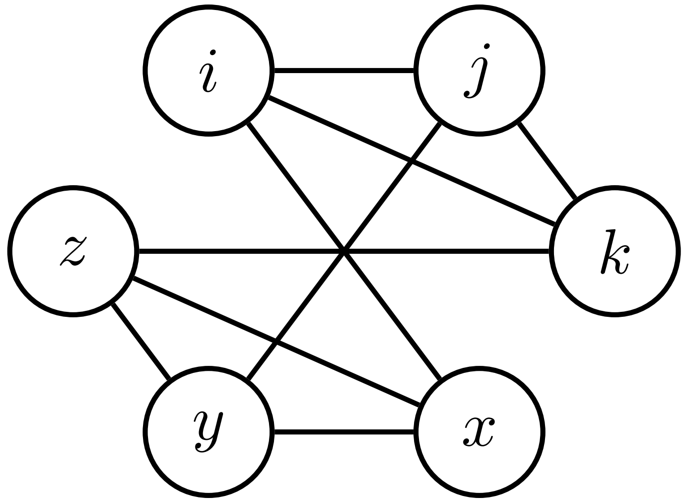
NP-Hardness: Setup

NP-Hardness: Setup

Reduction from Minimum Vertex Cover

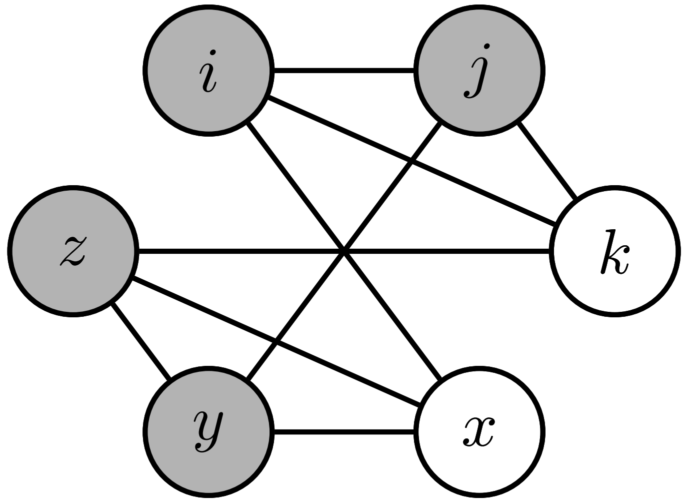
NP-Hardness: Setup

Reduction from Minimum Vertex Cover



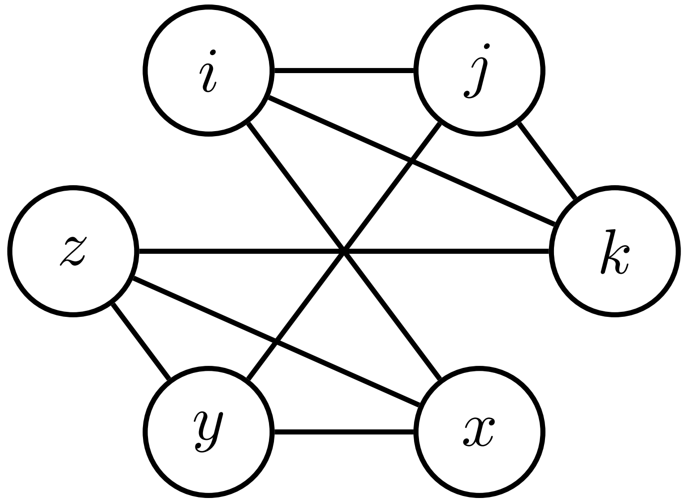
NP-Hardness: Setup

Reduction from Minimum Vertex Cover



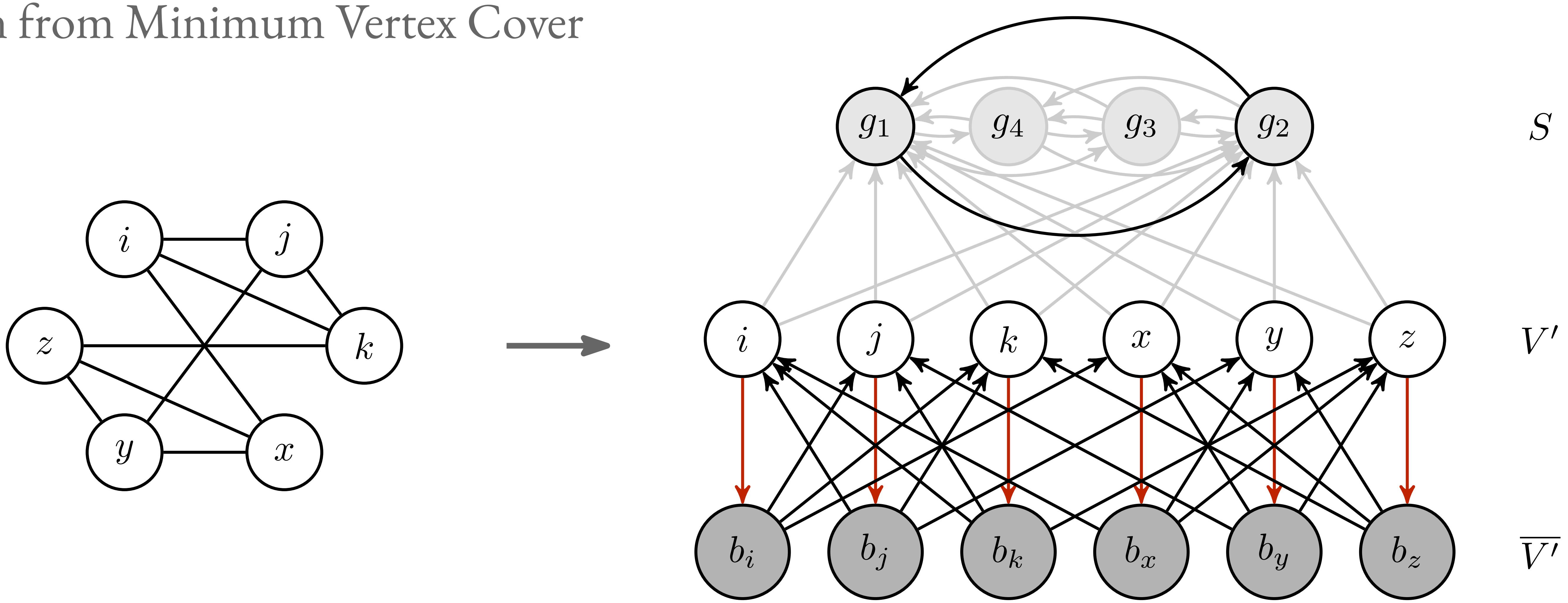
NP-Hardness: Setup

Reduction from Minimum Vertex Cover



NP-Hardness: Setup

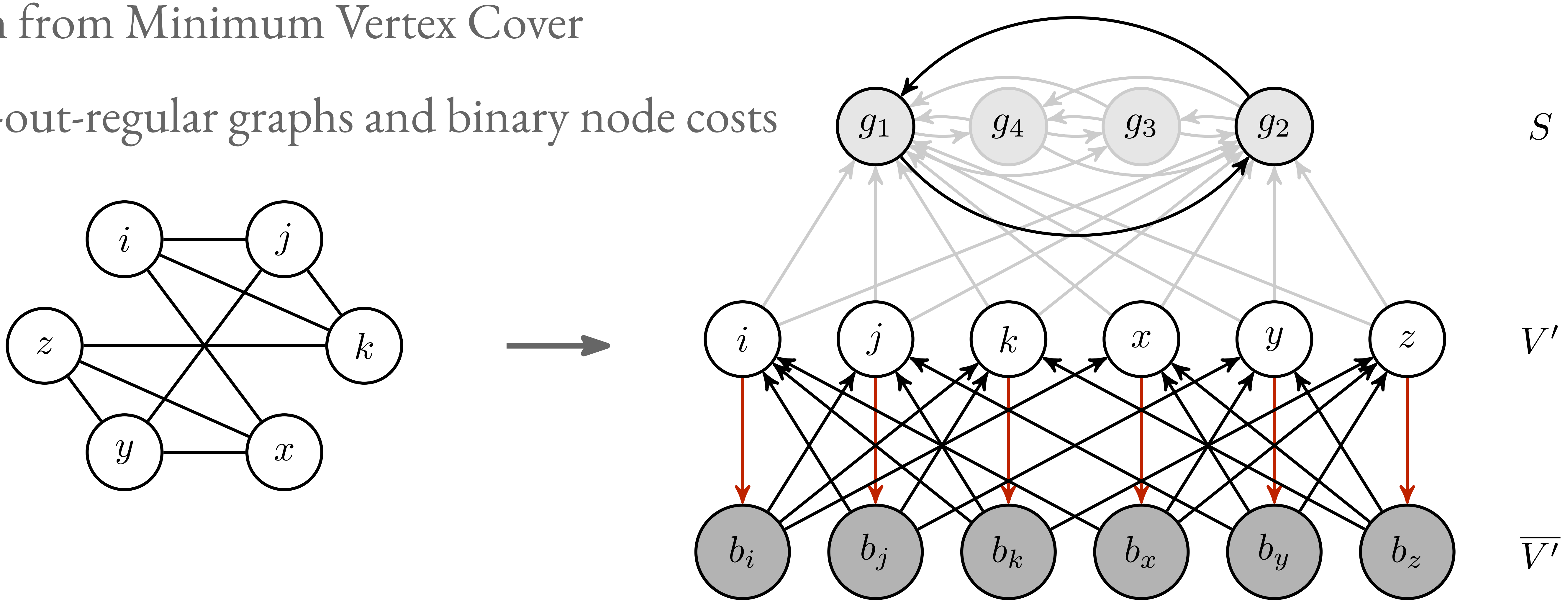
Reduction from Minimum Vertex Cover



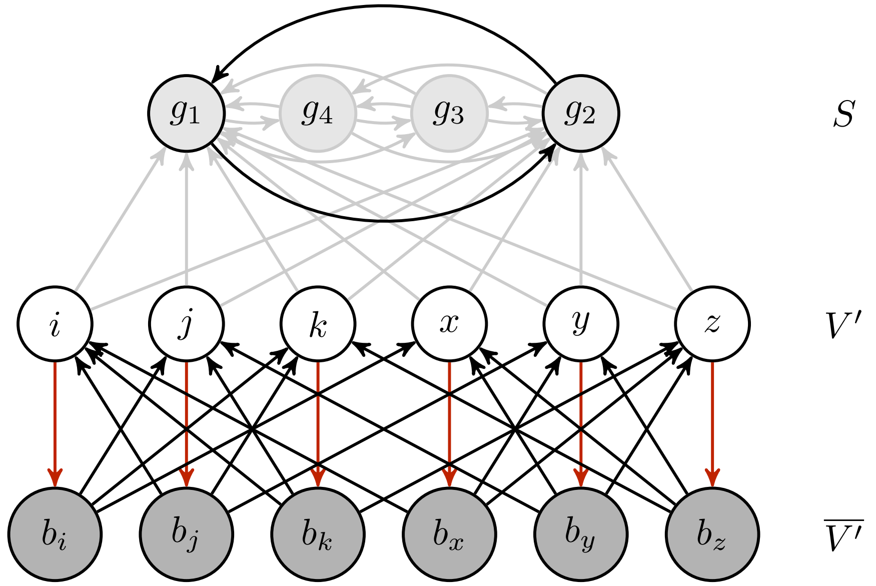
NP-Hardness: Setup

Reduction from Minimum Vertex Cover

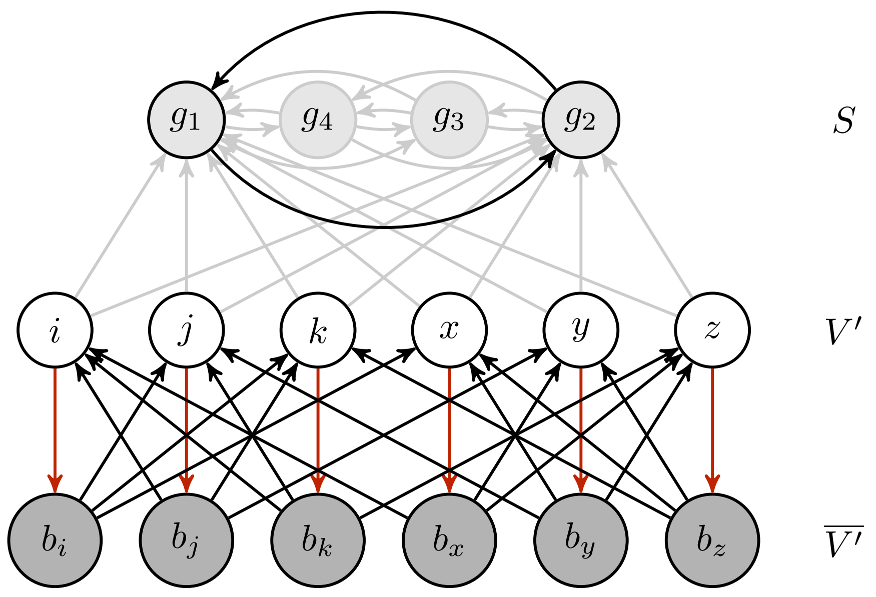
Even for 3-out-regular graphs and binary node costs



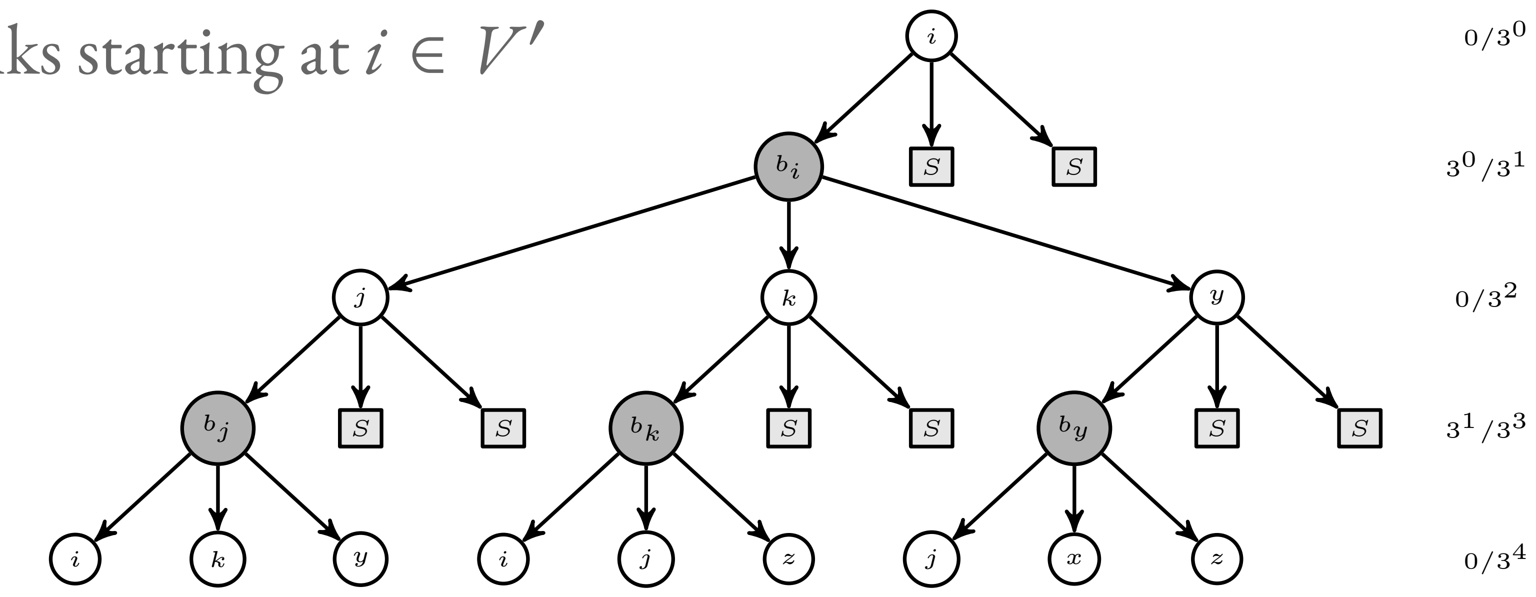
NP-Hardness: Intuition



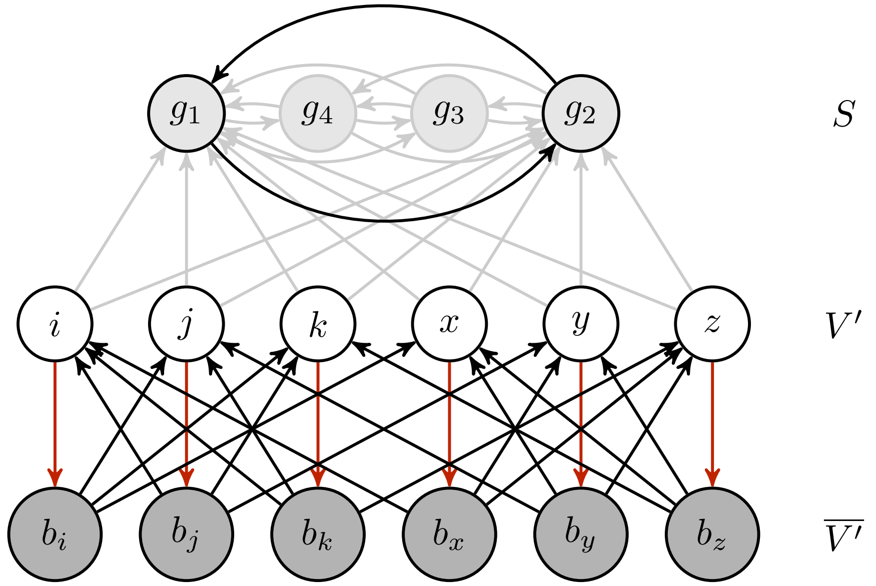
NP-Hardness: Intuition



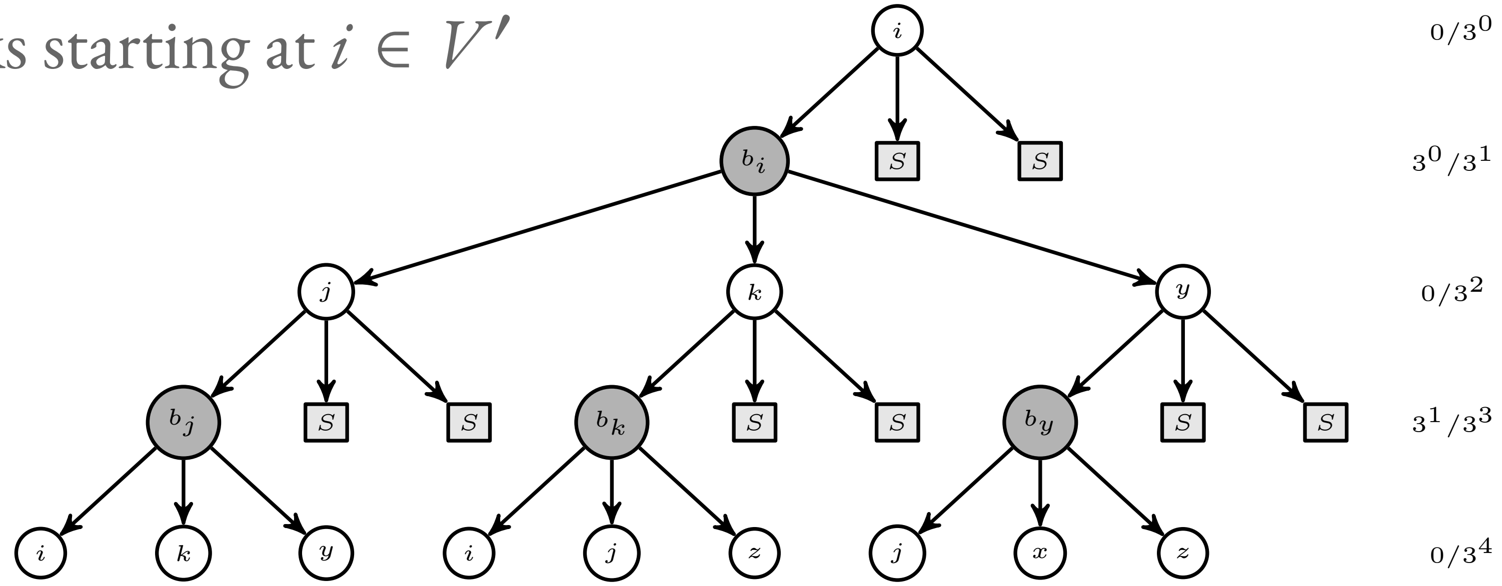
Walks starting at $i \in V'$



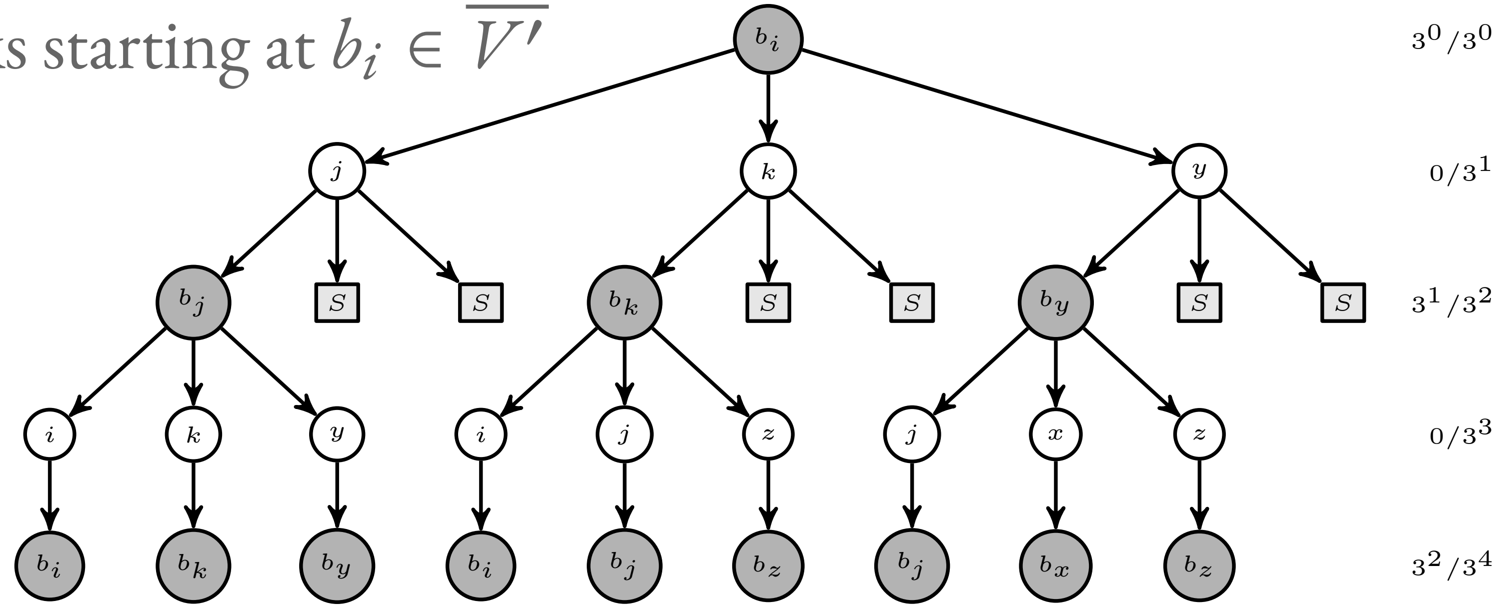
NP-Hardness: Intuition



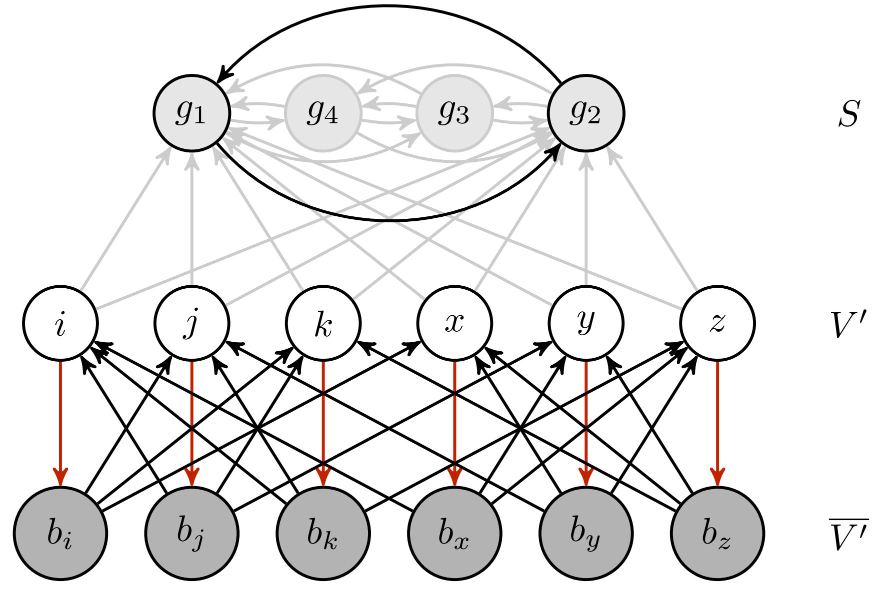
Walks starting at $i \in V'$



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

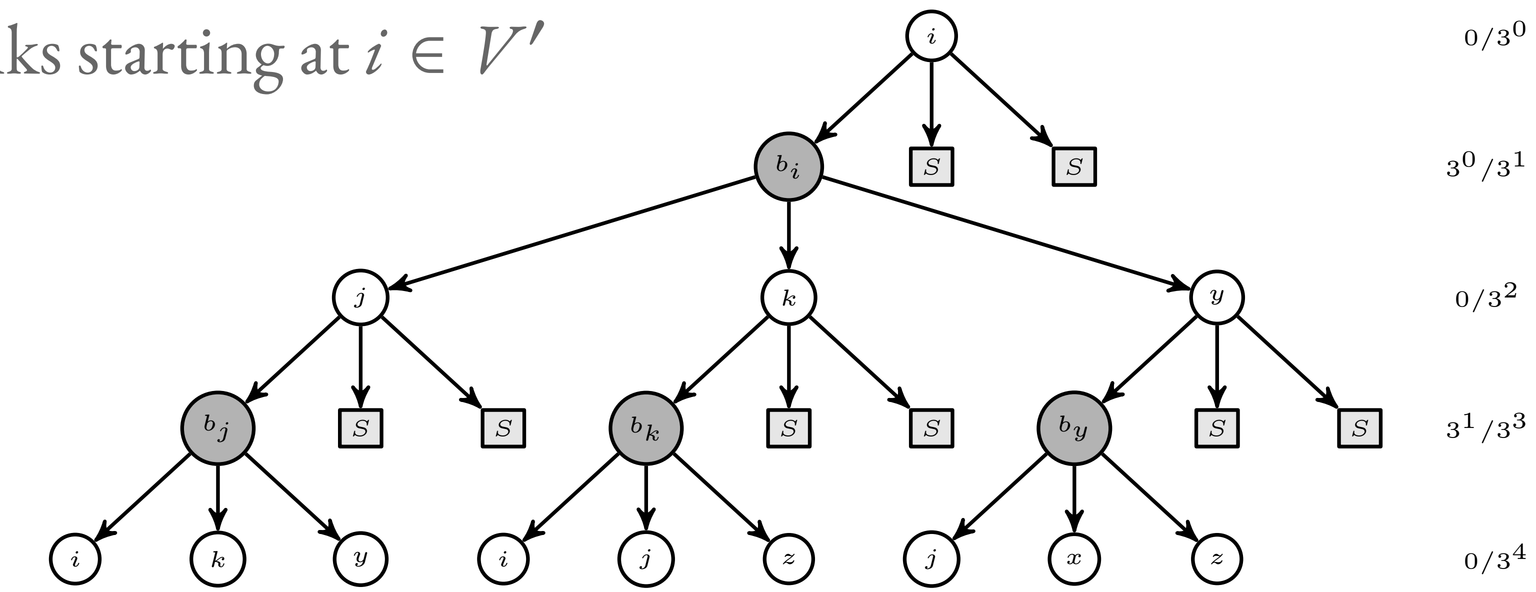


NP-Hardness: Intuition

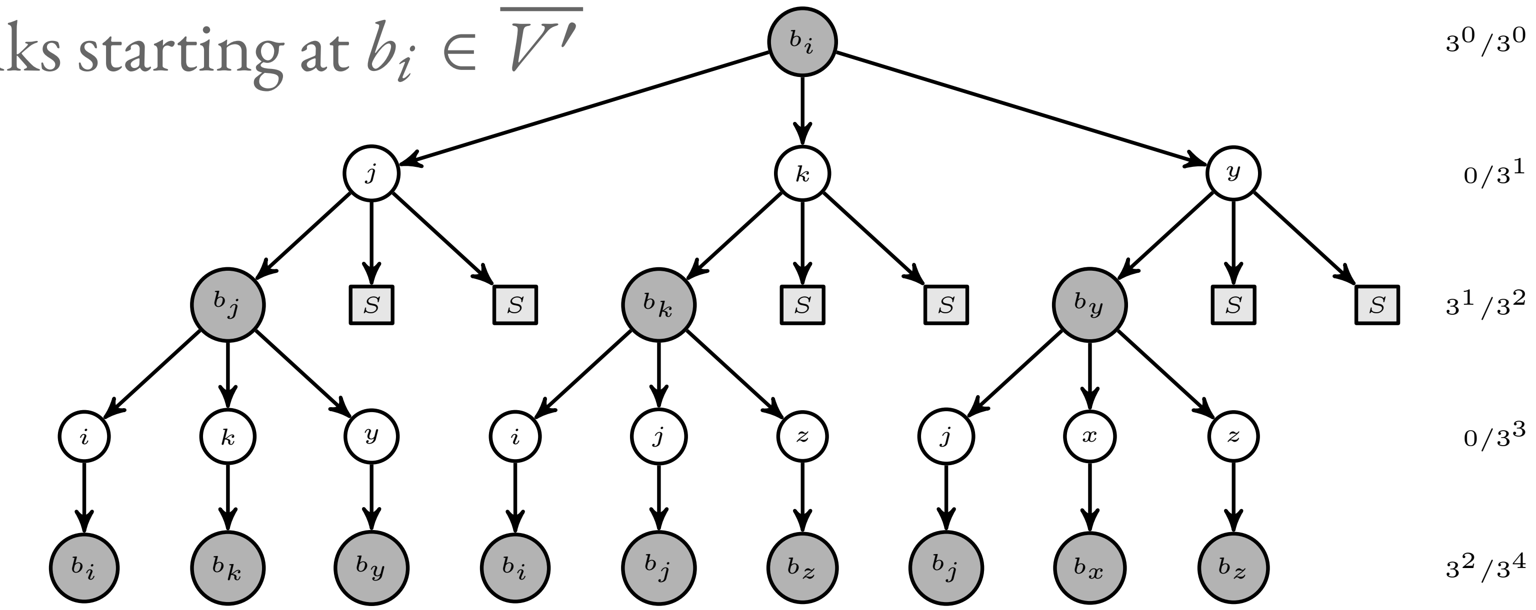


$$f(G) = n' \underbrace{\sum_{t=0}^{\infty} 3^{-t}(1-\alpha)^{2t}}_{\text{Contributions from } \overline{V'}}$$

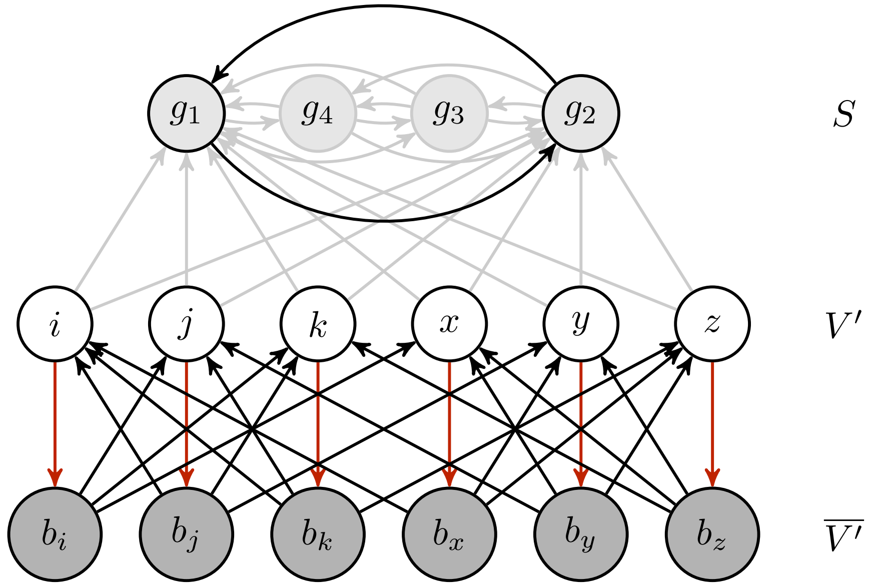
Walks starting at $i \in V'$



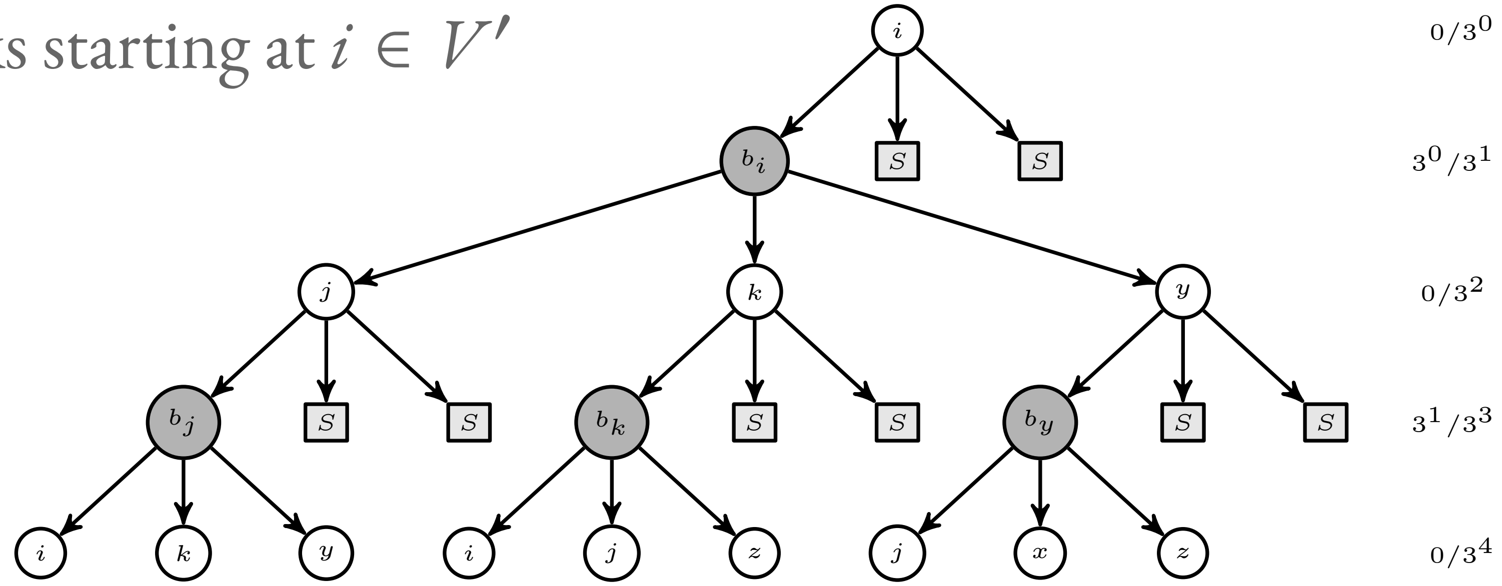
Walks starting at $b_i \in \overline{V'}$



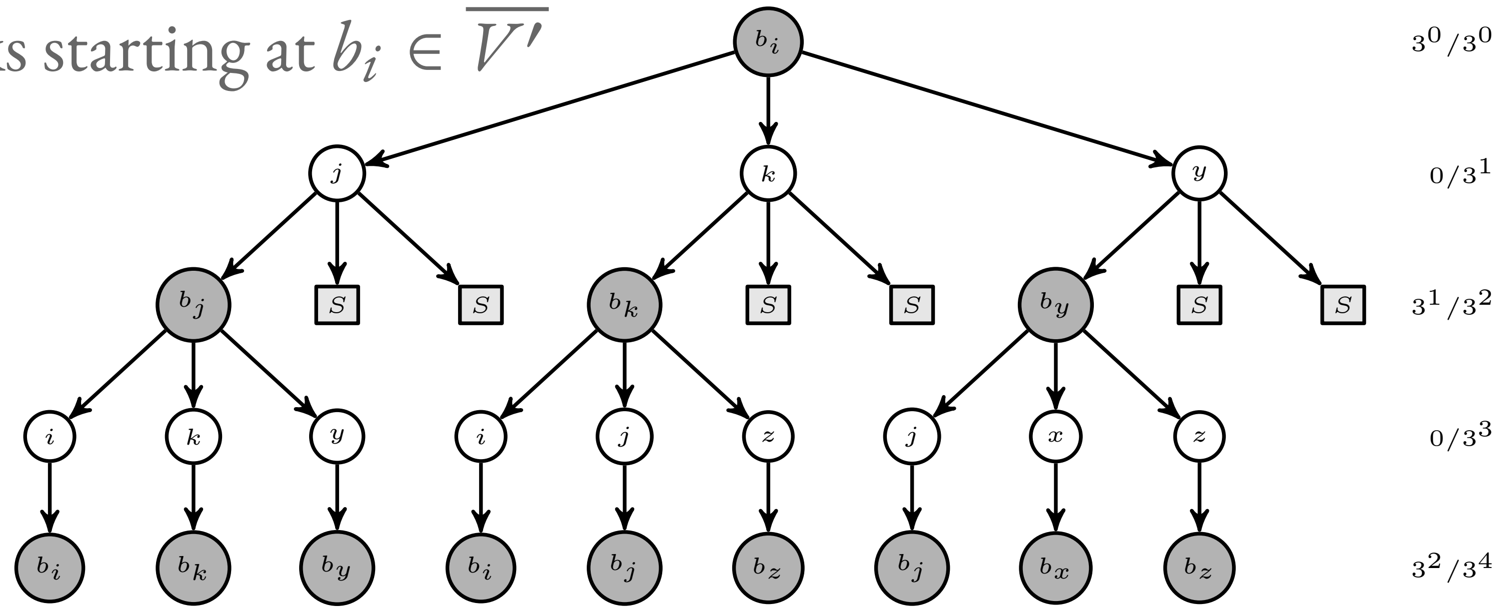
NP-Hardness: Intuition



Walks starting at $i \in V'$

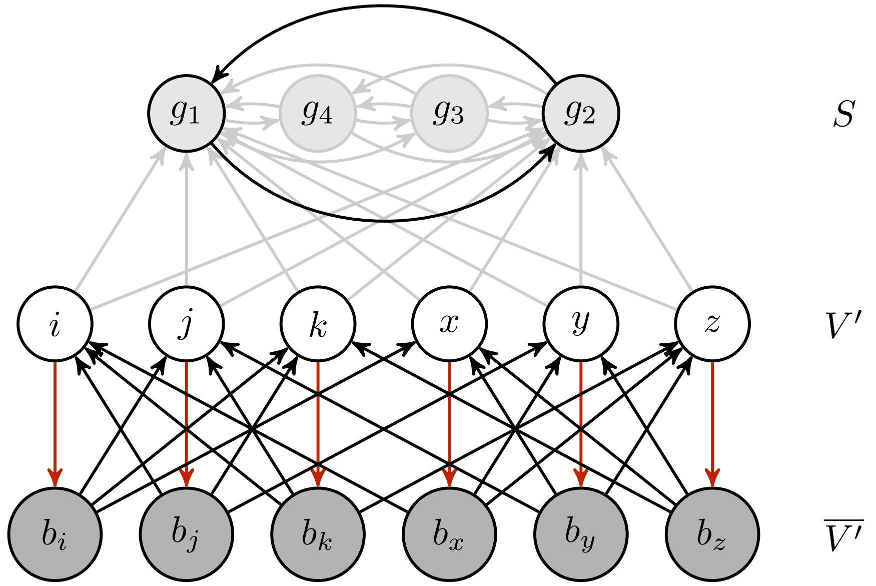


Walks starting at $b_i \in \overline{V'}$



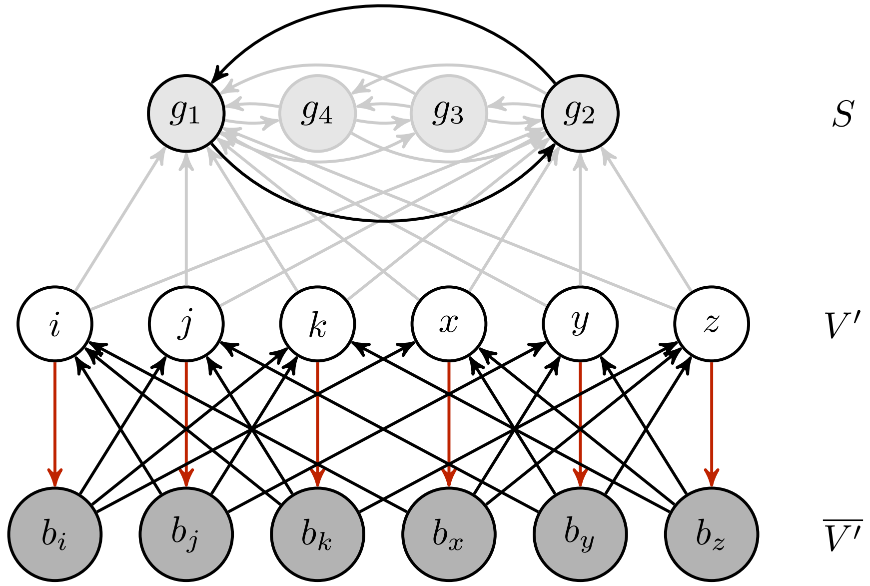
$$f(G) = n' \underbrace{\sum_{t=0}^{\infty} 3^{-t}(1-\alpha)^{2t}}_{\text{Contributions from } \overline{V'}} + n' \underbrace{\sum_{t=0}^{\infty} 3^{-t-1}(1-\alpha)^{2t+1}}_{\text{Contributions from } V'}$$

NP-Hardness: Sketch



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

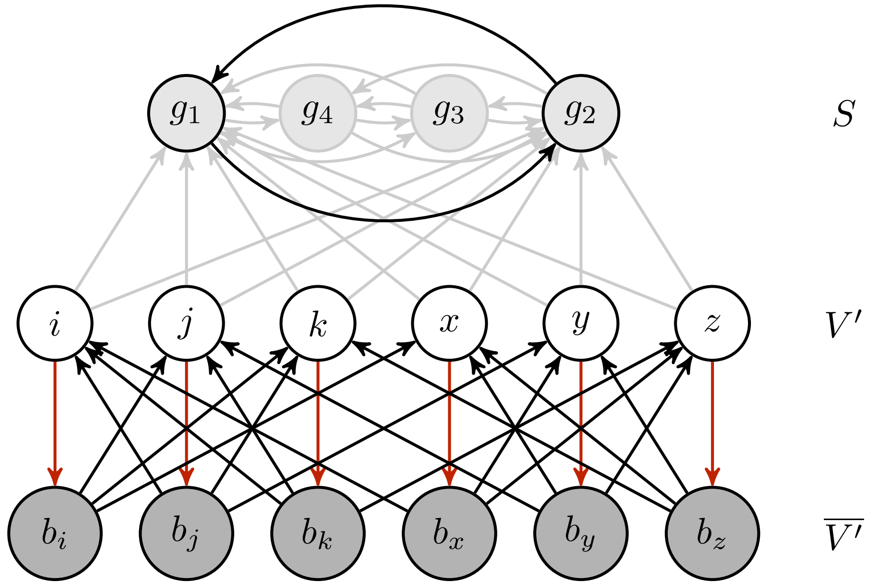
NP-Hardness: Sketch



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

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

NP-Hardness: Sketch

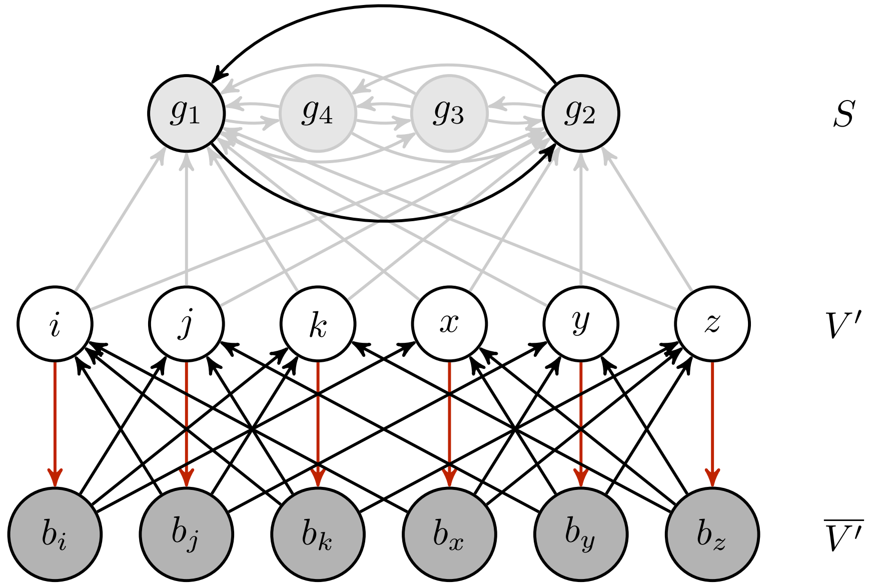


Candidate edges to rewire are $\{(a, b_a) \mid a \in V'\}$

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

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

NP-Hardness: Sketch



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

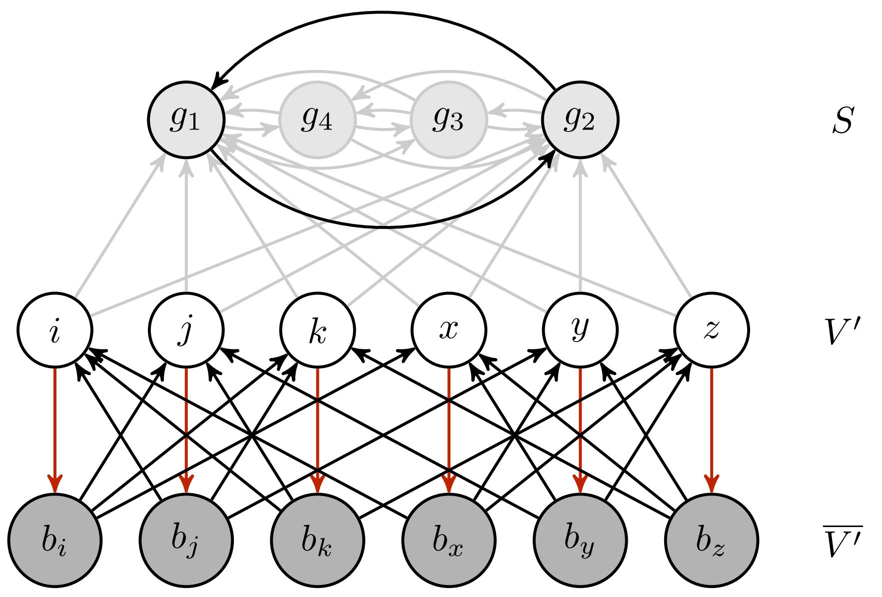
Rewiring $(i, b_i) \to (i, g_n)$ in $G \leftrightarrow i \in \mathrm{VC}(G)$

Reduction of $f(G)$ by individual rewiring

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

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

NP-Hardness: Sketch



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

Rewiring $(i, b_i) \to (i, g_n)$ in $G \leftrightarrow i \in \mathrm{VC}(G)$

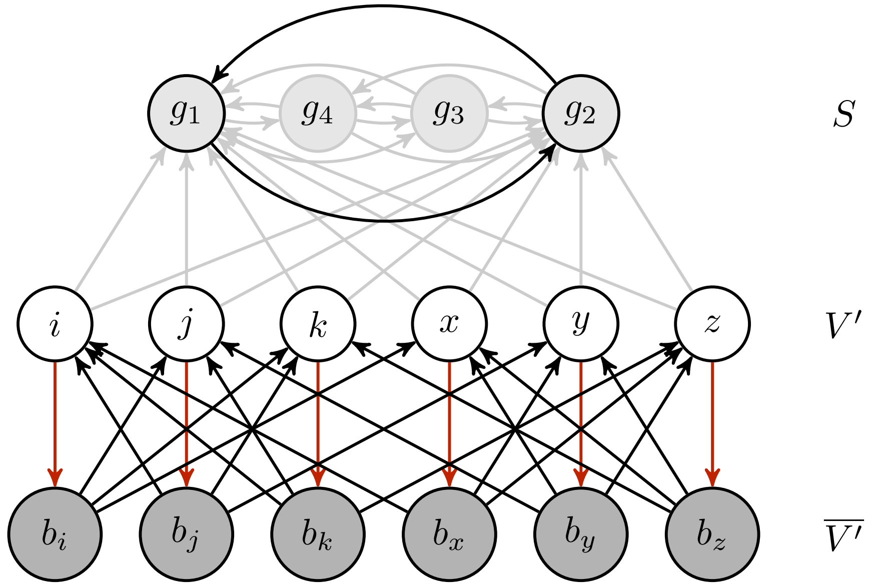
Reduction of $f(G)$ by individual rewiring

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

γ': Number of newly covered edges in G'

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

NP-Hardness: Sketch



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

Rewiring $(i, b_i) \longrightarrow (i, g_n)$ in $G \leftrightarrow i \in \mathrm{VC}(G)$

Reduction of $f(G)$ by individual rewiring

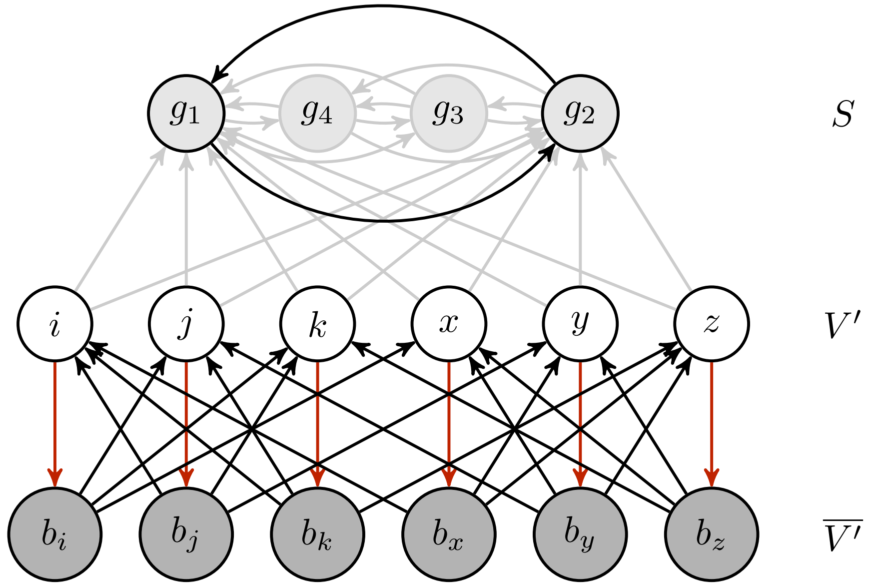
$$\frac{1}{3}(1-\alpha) + \frac{1}{9}(1-\alpha)^2 + \frac{2\gamma'}{27}(1-\alpha)^3 + \varepsilon'$$

γ': Number of newly covered edges in G'

ε': Contributions of walks beyond step 3

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

NP-Hardness: Sketch



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

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

Rewiring $(i, b_i) \to (i, g_n)$ in $G \leftrightarrow i \in \mathrm{VC}(G)$

Reduction of $f(G)$ by individual rewiring

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

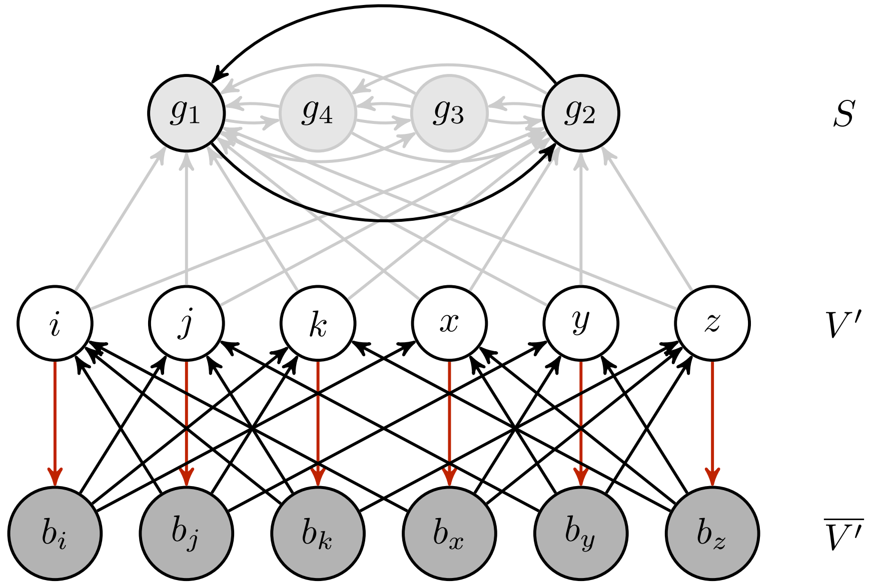
γ': Number of newly covered edges in G'

ε': Contributions of walks beyond step 3

MVC of size at most r

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

NP-Hardness: Sketch



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

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

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

γ': Number of newly covered edges in G'

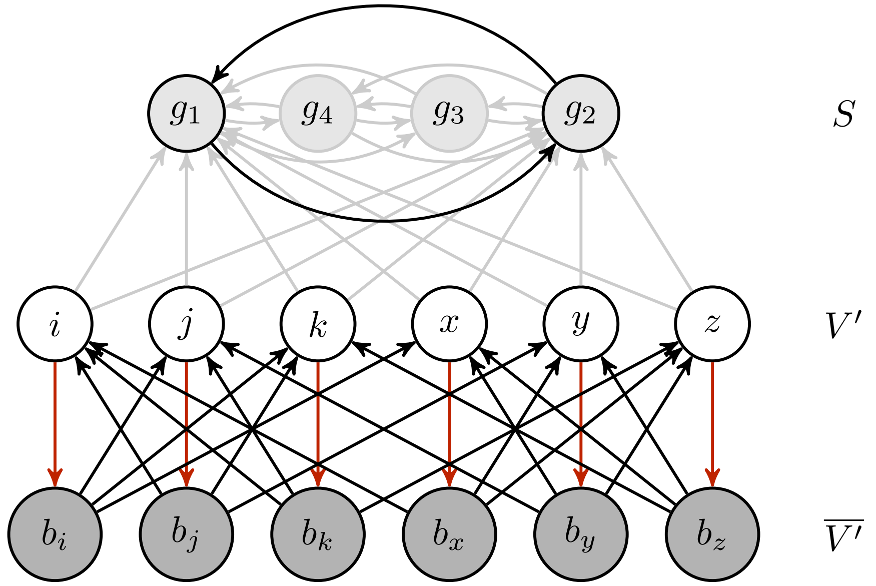
ε': Contributions of walks beyond step 3

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 $\xi < \frac{2}{27}(1-\alpha)^3$

NP-Hardness: Sketch



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

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

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

Reduction of $f(G)$ by individual rewiring

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

γ': Number of newly covered edges in G'

ε': Contributions of walks beyond step 3

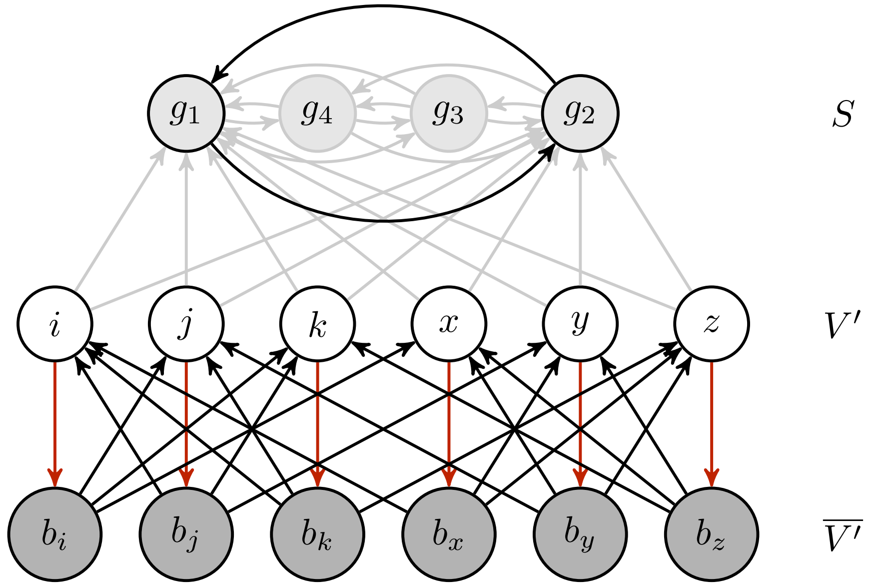
MVC of size at most r

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

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

ξ: *All* harm beyond step 3

NP-Hardness: Sketch



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

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

Rewiring $(i, b_i) \to (i, g_n)$ in $G \leftrightarrow i \in \mathrm{VC}(G)$

Reduction of $f(G)$ by individual rewiring

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

γ': Number of newly covered edges in G'

ε': Contributions of walks beyond step 3

MVC of size at most r

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

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

ξ: *All* harm beyond step 3

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