Fast Percolation Centrality Approximation with Importance Sampling

ICDM 2025

Antonio Cruciani
antonio.cruciani@aalto.fi

Aalto University
School of Science

Leonardo Pellegrina
leonardo.pellegrina@unipd.it

Università degli Studi di Padova

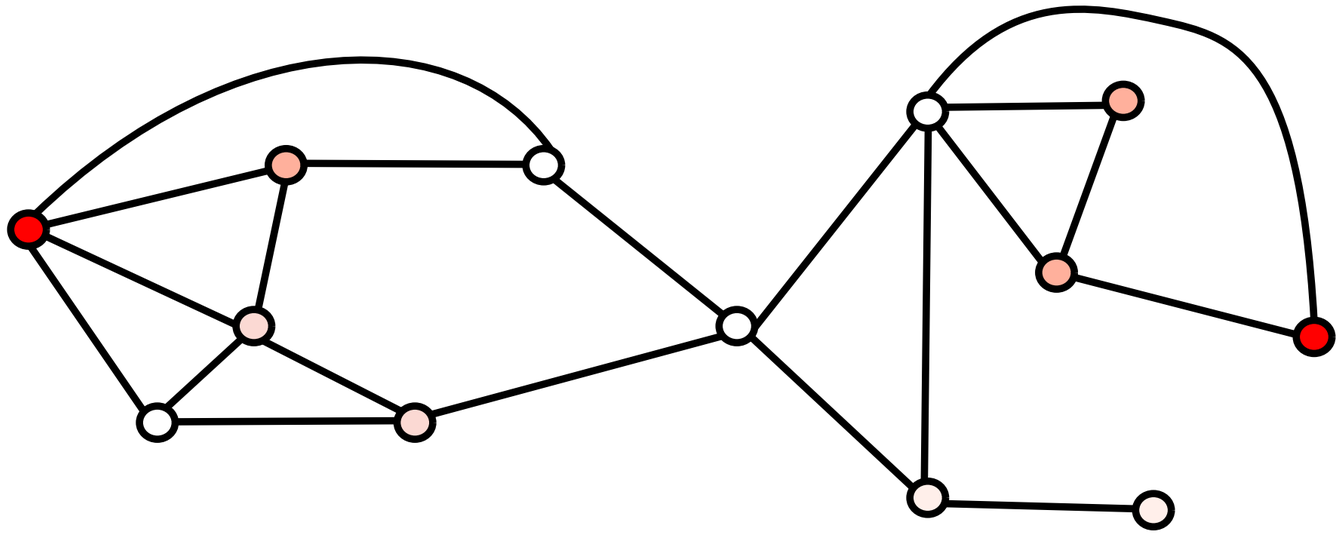
Our Problem

Input: We are given a graph $G = (V, E)$ and a percolation states vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in [0, 1]^n$

Goal: Compute the *percolation centrality* of each node $v \in V$

States values

- 1.0
- 0.5
- 0.4
- 0.1
- 0.0



Our Problem

Input: We are given a graph $G = (V, E)$ and a percolation states vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in [0, 1]^n$

Goal: Compute the *percolation centrality* of each node $v \in V$

[Piraveenan et al., PloS one]

$$p(v) = \sum_{s \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \cdot \kappa(s, t, v) \in [0, 1]$$

Where

- $\sigma_{st}(v)$ is the number of shortest paths between s and t *passing through* v
- σ_{st} is the number of shortest paths between s and t
- $\kappa(s, t, v)$ is defined as

$$\kappa(s, t, v) = \frac{R(x_s - x_t)}{\sum_{u \neq v \neq w} R(x_u - x_w)}$$

- $R(x) = \max(0, x)$

Our Problem

Practical Issue: The *exact* computation of the percolation centrality requires $\Omega(n^2)$ time (lower bound)!

IMPRACTICAL!

Our Goal: given an $\varepsilon \in (0, 1)$, compute an ***ε*-approximation** $\{\tilde{p}(v), v \in V\}$ of the percolation centrality for each node:

$$|\tilde{p}(v) - p(v)| \leq \varepsilon, \quad \forall v \in V$$

State of the art

Estimating the Percolation Centrality of Large Networks through Pseudo-dimension Theory [de Lima et al,KDD'20]

Their results in a nutshell

They use **uniform** sampling (UNIF) to approximate:

$$p^\star(v) = \frac{1}{n(n-1)} \sum_{s \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \cdot \kappa(s,t,v) \in \left[0, \frac{1}{n(n-1)}\right]$$

Sample size of

$$\ell = \frac{0.5}{\varepsilon^2} \left(\lceil \log(D) - 2 \rceil + 1 - \ln \delta \right)$$

To achieve an ε-approximation with probability $\geq 1 - \delta$

Some issues with the SOTA

$$p^\star(v) = \frac{1}{n(n-1)} \sum_{s \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \cdot \kappa(s,t,v) \in \left[0, \frac{1}{n(n-1)}\right]$$

$\varepsilon \geq \frac{1}{n(n-1)}$ ⟶ Is uninformative! Is enough to directly output $\{\tilde{p}^\star(v) = 0, \forall v \in V\}$

$\varepsilon < \frac{1}{n(n-1)}$ ⟶ We need $\ell \in \Omega(n^4)$ samples!

No truly effective algorithm exists to approximate the percolation centrality.

Why IS and not UNIF?

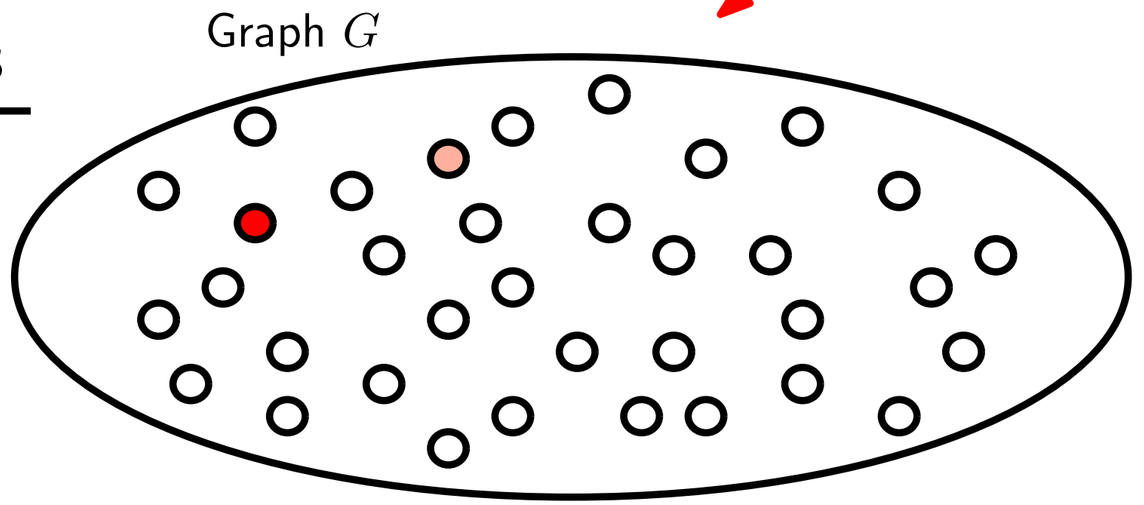
Observation: given s and t in G, if $x_s \leq x_t$ then:

$$\kappa(s,t,v) = \frac{R(x_s - x_t)}{\sum_{u \neq v \neq w} R(x_u - x_w)} = 0$$

$\longrightarrow$ Sampling s, t with $x_s \leq x_t$ is useless!

Toy example:

	Percolation states
●	1.0
●	0.5
○	0.0



We need a "big" sample size to pick the red or the light-red nodes as sources using UNIF!

IS boosts the sampling of such points, obtaining more accurate estimates

Uniform sampling has high variance

Our Approach

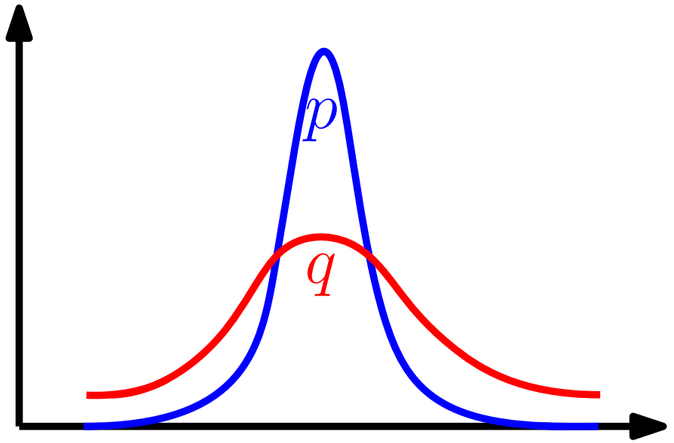
A new algorithm called **PercIS** based on **Importance Sampling** that returns an ε-approximation with probability $\geq 1 - \delta$ efficiently, thanks to new sharp sample size bounds.

Importance Sampling

We use **Importance Sampling**

We want to approximate an expectation

$$\mu = \mathbb{E}_p[f(X)] = \sum_x f(x)p(x)$$



Problem: Sampling from p might be inefficient

Idea: Sample from an importance distribution q which emphasizes "important" regions.

$$\mathbb{E}_p[f(X)] = \mathbb{E}_q\left[f(X)\frac{p(X)}{q(X)}\right]$$

The quality of our importance distribution is

$$\hat{d} = \max_{x:q(x)>0} \frac{p(x)}{q(x)}$$

PercIS: Importance Sampling Distribution

Idea: sample s and t with probability $\kappa(s,t,v)$

Challenge: we want to estimate n averages (the set $\{p(v),\, v \in V\}$) simultaneously, but the weights $\kappa(s,t,v)$ depend on v!

We define $\tilde{\kappa} : V \times V \to [0,1]$

$$\tilde{\kappa}(s,t) = \frac{R(x_s - x_t)}{\sum_{u \neq w} R(x_u - x_w)}$$

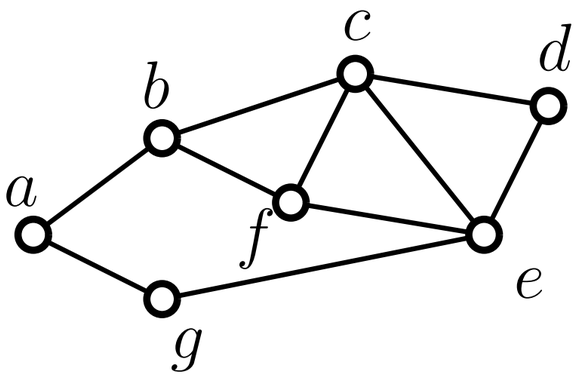
$\tilde{\kappa}$ is a valid distribution over all couples of nodes

For any shortest path τ_{st} between nodes s and t, we define the importance distribution q as:

$$q(\tau_{st}) = \frac{\tilde{\kappa}(s,t)}{\sigma_{st}}, \quad \text{For all shortest paths } \tau_{st} \text{ of } G$$

PercIS: sampling from q

Problem: sampling s, t with probability $\tilde{\kappa}(s, t)$

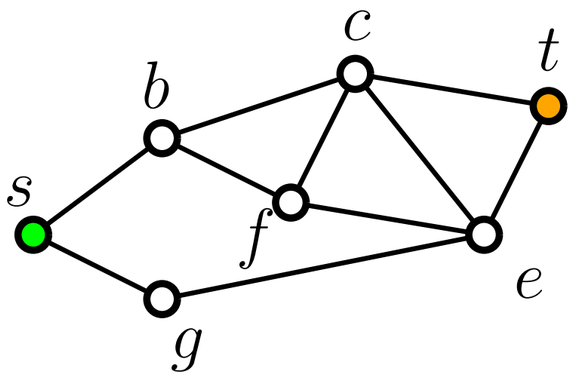


PercIS: sampling from q

Problem: sampling s, t with probability $\tilde{\kappa}(s, t)$

1) Sample s with marginal $\Pr(s) = \sum_u \tilde{\kappa}(s, u)$

2) Sample t with $\Pr(t \mid s) = \frac{\tilde{\kappa}(s, t)}{\sum_u \tilde{\kappa}(s, u)}$



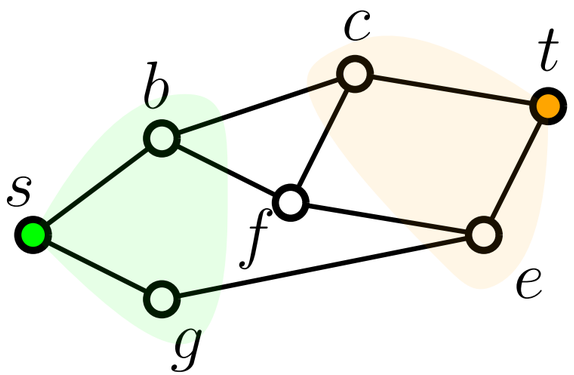
PercIS: sampling from q

Problem: sampling s, t with probability $\tilde{\kappa}(s, t)$

1) Sample s with marginal $\Pr(s) = \sum_u \tilde{\kappa}(s, u)$

2) Sample t with $\Pr(t \mid s) = \frac{\tilde{\kappa}(s, t)}{\sum_u \tilde{\kappa}(s, u)}$

3) Perform a Bidirectional Balanced BFS from s and t



PercIS: sampling from q

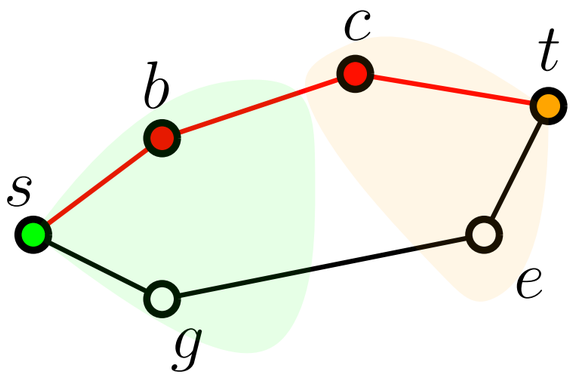
Problem: sampling s, t with probability $\tilde{\kappa}(s,t)$

1) Sample s with marginal $\Pr(s) = \sum_u \tilde{\kappa}(s,u)$

2) Sample t with $\Pr(t \mid s) = \frac{\tilde{\kappa}(s,t)}{\sum_u \tilde{\kappa}(s,u)}$

3) Perform a Bidirectional Balanced BFS from s and t

4) Pick a shortest path τ_{st} u.a.r.

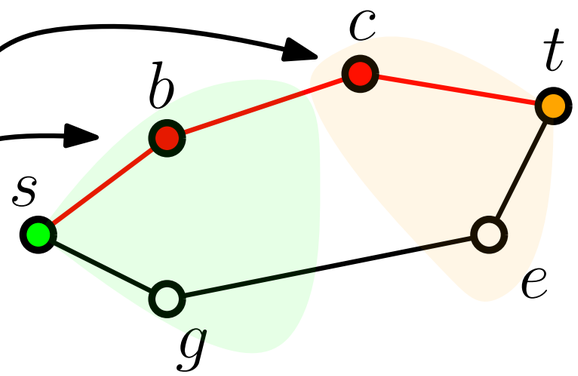


PercIS correctly draws ℓ samples from q in time $\mathcal{O}(n + \ell(\log n + T_{BBFS}))$ and space $\mathcal{O}(n+m)$

PercIS: the Estimator

Given a collection of shortest paths $\mathcal{S} = \{\tau^1, \tau^2, \ldots, \tau^\ell\}$ drawn from q

$$\tilde{p}(v) = \frac{1}{\ell} \sum_{\tau_{st} \in \mathcal{S}} \frac{\kappa(s,t,v)}{\tilde{\kappa}(s,t)} \cdot \underbrace{\mathbb{1}\left[v \in \mathcal{I}(\tau_{st})\right]}_{\text{Internal Nodes}}$$



Our estimator is *unbiased*

Upperbound on the variance $\mathrm{Var}_q[\tilde{p}(v)] \leq \hat{d} \cdot p(v)$

PercIS: sample complexity

New **data-dependent** upper bound on the sample size!

Given $\varepsilon, \delta \in (0, 1)$

$$\ell \approx \frac{(2\hat{v} + \frac{2}{3}\varepsilon\hat{d})}{\varepsilon^2} \cdot \left(\ln(\hat{d}\hat{\rho}/\hat{v}) + \ln(2/\delta)\right)$$

we obtain a ε-approximation with probability $\geq 1 - \delta$.

Where:

- $\hat{\rho}$ is a bound to the average path length (observation: $\hat{\rho} \leq D$)
- $\hat{v}$ is a bound to the max empirical variance ($\hat{v} \geq \max_v \mathsf{Var}[\hat{p}(v)]$)
- $\hat{d}$ is the max likelihood ratio $\max_{v \in V} \max_{s \neq t} \frac{\kappa(s,t,v)}{\tilde{\kappa}(s,t)}$

PercIS vs UNIF: theoretical results

Define the **State Gap** as

$$\Delta = \max_{v \in V} \left\{ \max_{s \neq v \neq t} (x_s - x_t) \right\}$$

Then:

- When $\Delta \in \Omega(1)$, the likelihood ratio $\hat{d}$ of the Importance Distribution q is $\mathcal{O}(1)$
- There exist instances with $\Delta \in \Omega(1)$ where $\hat{d}$ for UNIF is $\Omega(n)$.
- There exists instances with $\Delta \in \Omega(1)$ where we need $\Omega(n^2)$ random samples for UNIF, while $\mathcal{O}(n)$ random samples for PercIS!

Experimental Analysis

Graph	$\|V\|$	$\|E\|$	D	ρ	Type
P2P-Gnutella31	62586	147892	31	7.199	D
Cit-HepPh	34546	421534	49	5.901	D
Soc-Epinions	75879	508837	16	2.755	D
Soc-Slashdot	82168	870161	13	2.135	D
Web-Notredame	325729	1469679	93	9.265	D
Web-Google	875713	5105039	51	9.713	D
Musae-Facebook	22470	170823	15	2.974	U
Email-Enron	36692	183831	13	2.025	U
CA-AstroPH	18771	198050	14	2.194	U

We assign percolation states using different settings:

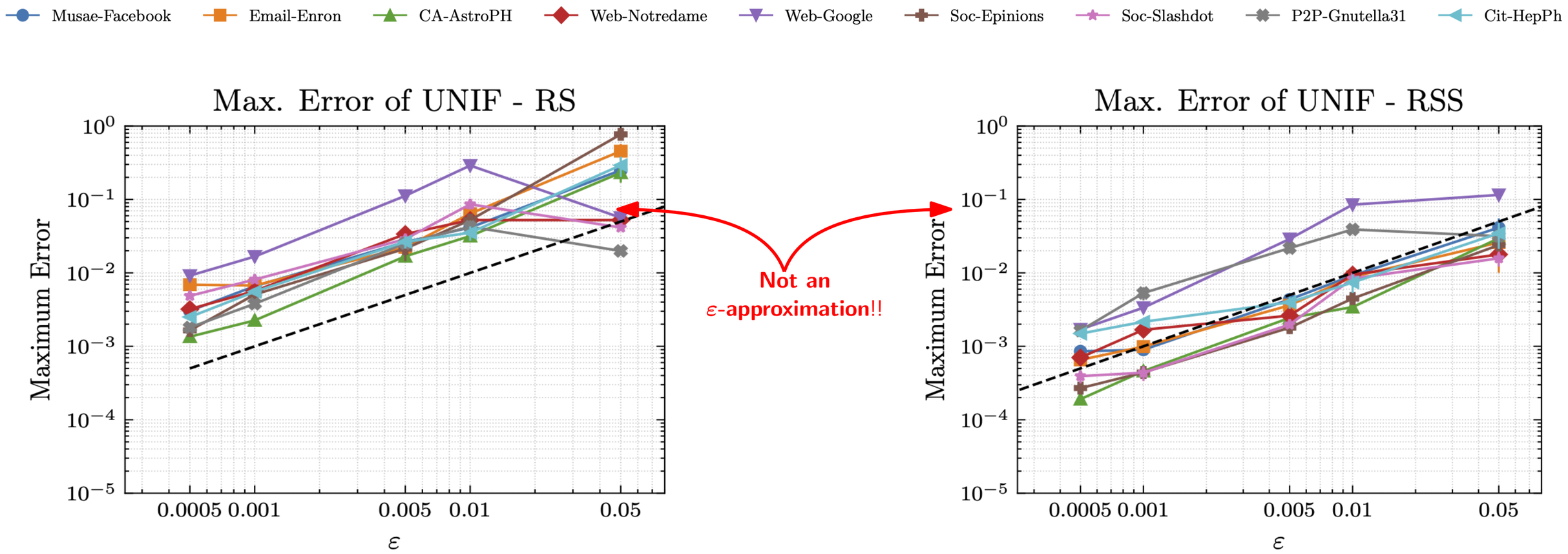
Random Seeds (RS): $\mathcal{O}(1)$ number of nodes v with $x_v = 1$ and the rest is set to 0

Random Seed Spread (RSS): $\log n$ random initiators v with $x_v = 1$ and simulation of infection spreading process from them.

Isolated Component (IC): Only a isolated constant sized component has percolation states > 0

Uniform Sates (UN): Each $x_v \sim \text{Uniform}([0, 1])$

UNIF and ε-approximation



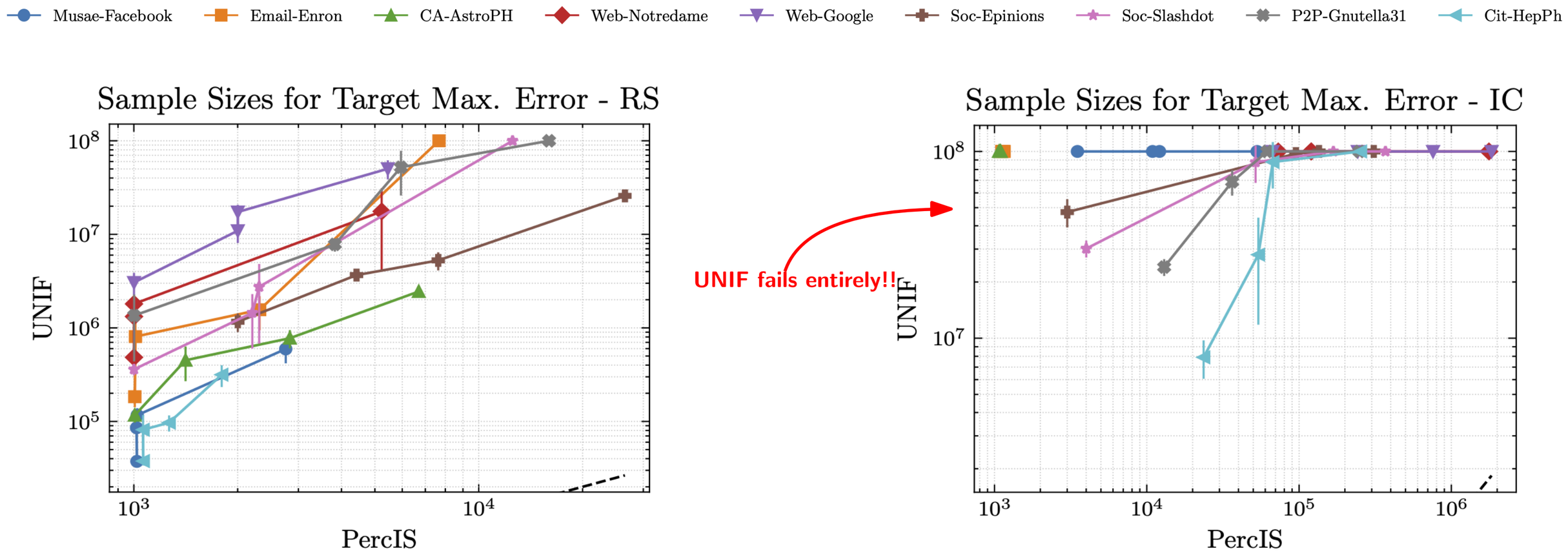
Maximum Error of UNIF on random samples of size $\mathcal{O}(\ln(D/\delta)/\varepsilon^2)$

Maximum (absolute) Error PercIS vs UNIF

Maximum Errors of PercIS (x axes) and UNIF (y axes) on random samples of *fixed* sizes $\ell \in [10^3, 10^6]$.

PercIS significantly outperforms UNIF on every graph and every setting!

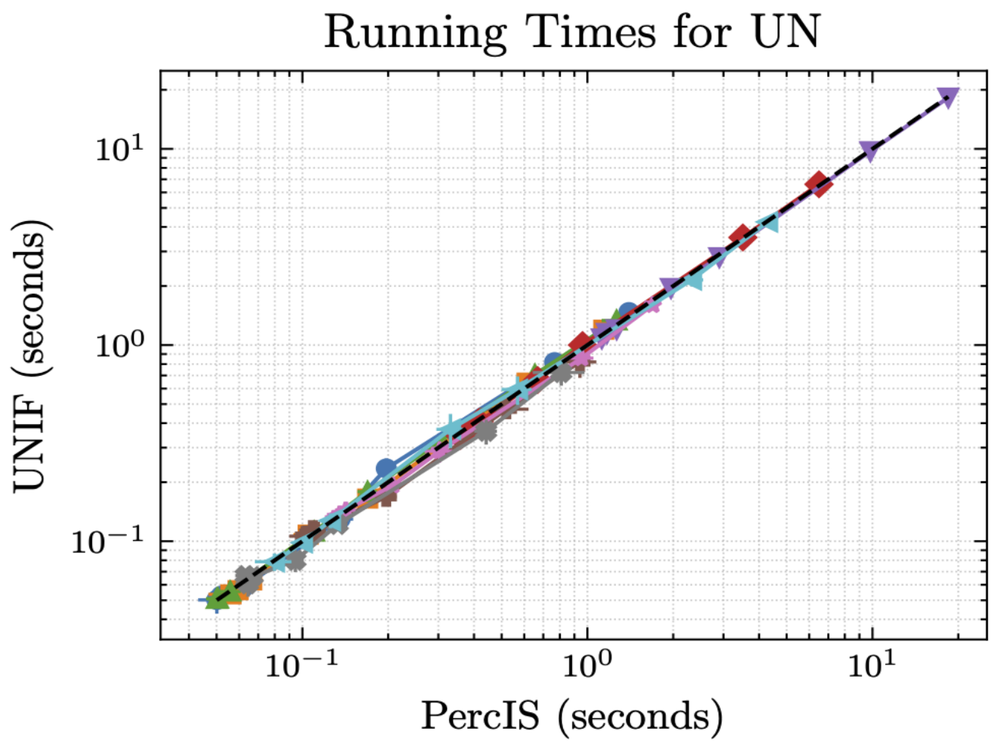
Target Maximum Error

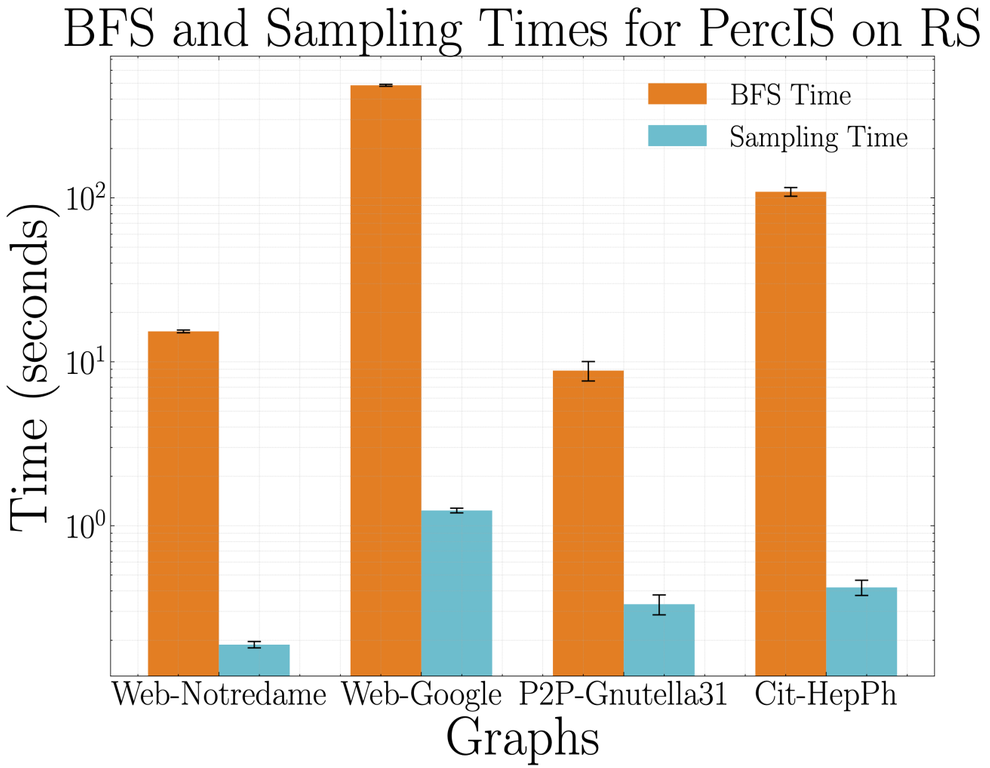


Sample sizes required to obtain a Maximum Error $\leq \varepsilon$ by UNIF (y axes) and PercIS (x axes). We set the cap to 10^8

PercIS always converges with a smaller number of samples

Running Times (for equal sample size)





PercIS has a running time comparable to UNIF

Equal sample size
$\ell \in [10^3, 10^6]$

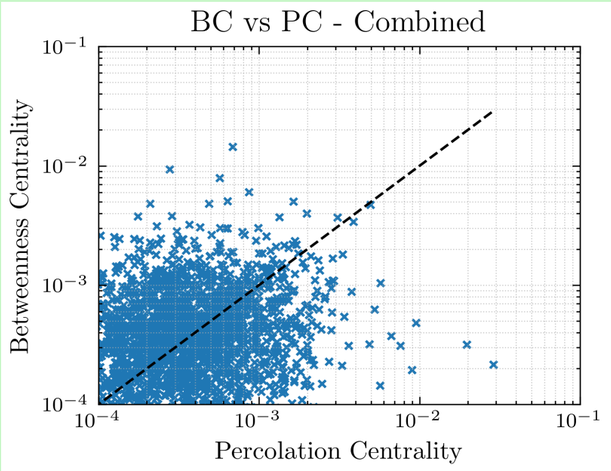
ImportanceSampler overhead is negligible!

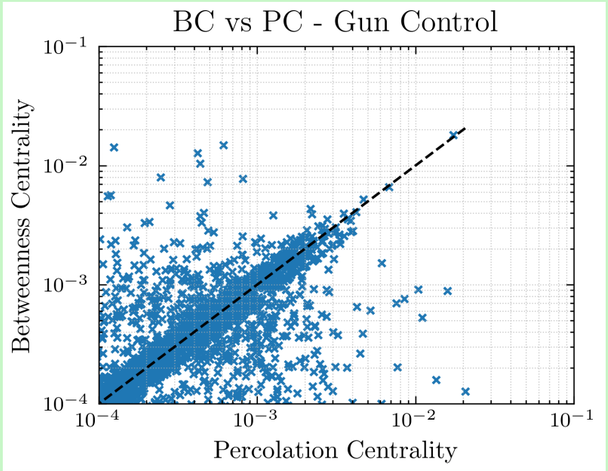
Experiments for Labeled Networks

More potential for:

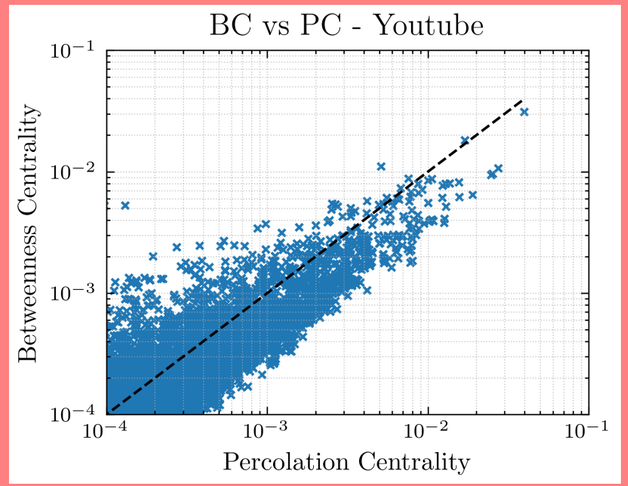
- identifying bridges among users with opposing views/opinions
- flagging content that comes form radicalization pathways

Opinion Networks





Harmful Contents



	Jaccard Similarity Top-K		
Graph	**10**	**50**	**100**
Guns	0.053	0.087	0.117
Combined	0.0	0.031	0.015
Youtube	0.429	0.369	0.504

Jaccard similarity of the top k nodes for betweenness and percolation.

Conclusions

- We presented PercIS, a novel approximation algorithm for the PC
- Novel Importance Sampling Distribution
- Tight theoretical guarantees
- PercIS consistently outperform the state-of-the-art

Thank You!

Our paper