

# CrashSim: An Efficient Algorithm for Computing SimRank over Static and Temporal Graphs

Mo Li<sup>1,2</sup>, Farhana M. Choudhury<sup>2</sup>, Renata Borovica-Gajic<sup>2</sup>, Zhiqiong Wang<sup>1</sup>, Junchang Xin<sup>1,\*</sup>, Jianxin Li<sup>3</sup>

<sup>1</sup>Northeastern University, CN

<sup>2</sup>University of Melbourne, AU

<sup>3</sup>Deakin University, AU

# Outline

---

## Background

- SimRank Overview
- Motivation

## Our Approach

- CrashSim Algorithm — static graphs
- CrashSim-T Algorithm — temporal graphs



## Problem Definition

- Preliminaries
- Problem Definition

## Experiments and Conclusion

- Experimental Evaluation
- Conclusion

---

# Background

- SimRank Overview
- Motivation



/01

---

# Background

- Similarity assessment plays a vital role in our lives.

Customers Who Bought This Item Also Bought Page 1 of 15



The screenshot shows a grid of seven book covers with their titles, authors, ratings, and prices. The books are: 'Data Science from Scratch: First Principles with Python' by Joel Grus; 'Python for Data Analysis: Data Wrangling with Pandas, NumPy, and...' by Wes McKinney; 'Data Science for Business: What You Need to Know about Data Mining and...' by Foster Provost; 'Reproducible Research with R and R Studio, Second Edition...' by Christopher Gandrud; 'An Introduction to Statistical Learning with Applications in R...' by Gareth James; 'Data Smart: Using Data Science to Transform Information into Insight' by John W. Foreman; and 'The Statistical Sleuth: A Course in Methods of Data Analysis' by Fred Ramsey.

Recommender System



Citation Graph

The graph consists of many small circular nodes connected by thin lines, representing a network of citations between academic papers. The nodes are arranged in a roughly circular pattern with many internal connections.



Collaboration Network

The graph shows a network of stylized human icons connected by dashed lines, representing a collaboration network. The nodes are arranged in a circular pattern with many internal connections, similar to the citation graph but with more distinct nodes.

# Background

---

- SimRank
  - Node-to-node measurement based on the topology of graphs (KDD'02)
  - Basic assumption
    - *Two nodes will be similar if they are both highly relevant to similar nodes*

- Two Forms
  - Original definition (KDD'02)

$$s(u, v) = \begin{cases} 1, & \text{if } u = v \\ \frac{c}{|I(u)| \cdot |I(v)|} \sum_{x \in I(u), y \in I(v)} s(x, y), & \text{otherwise.} \end{cases}$$

- $\sqrt{c}$ - walk (SIGMOD'16)

$$s(u, v) = \Pr [W(u) \text{ and } W(v) \text{ meet}]$$

# Background

- Temporal Graph

The image shows a screenshot of an Amazon product page for a book. The section is titled "Customers Who Bought This Item Also Bought" and displays seven recommended books. A diagram on the right side of the screenshot illustrates a Recommender System. It consists of two nodes, F and H, connected by a double-headed arrow. Node F is positioned above node H. Three arrows point from the recommended books to node F, and one arrow points from node H to node F. The text "Recommender System" is written in orange to the right of the diagram.

Book Title	Author	Format	Price
Data Science from Scratch: First Principles with Python	Joel Grus	Mining Paperback	\$33.99
Python for Data Analysis: Data Wrangling with Pandas, NumPy, and...	Wes McKinney	Paperback	\$27.68
Data Science for Business: What You Need to Know about Data Mining and...	Foster Provost	Paperback	\$37.99
Reproducible Research with R and R Studio, Second Edition...	Christopher Gandrud	Paperback	\$51.97
An Introduction to Statistical Learning with Applications in R...	Garth James	Hardcover	\$68.35
Data Smart: Using Data Science to Transform Information into Insight	John W. Foreman	Simulation Paperback	\$28.16
The Statistical Sleuth: A Course in Methods of Data Analysis	Fred Ramsey	Hardcover	\$294.42

- Temporal SimRank queries: *threshold* and *trend*

# Problem Definition

- Preliminaries
- Problem Definition

/02

# Preliminaries

---

**Definition 1** ( $\sqrt{c}$ -walk). Let  $c$  denotes the decay factor in the definition of SimRank, a  $\sqrt{c}$ -walk in  $G$  is defined such that:

- In each step of the random walk, we have  $1 - \sqrt{c}$  probability to stop.
- For the remaining  $\sqrt{c}$  probability, one of the in-neighbors of the current node is selected uniformly at random as the next step.

$$s(u, v) = \Pr [W(u) \text{ and } W(v) \text{ meet}]$$

SLING algorithm (SIGMOD'16)

$$= \sum_i \Pr [W(u) \text{ and } W(v) \text{ first meet at } u_i].$$

ProbeSim algorithm (VLDB'17)



# Problem Definition

---

**Problem** (Temporal SimRank Queries)

**Given:**  $G, u, [T_1, T_t]$

**Return:** node set  $\Omega$ , such that the SimRank of  $u$  and each node  $v \in \Omega$  continuously meet a certain query requirement during the entire query interval  $[T_1, T_t]$

**Problem** (Temporal SimRank Trend Query)

**Given:**  $G, u, [T_1, T_t]$

**Return:** node set  $\Omega$ , such that the SimRank of  $u$  and each node  $v \in \Omega$  is continuously **increasing (or decreasing)** during the entire query interval  $[T_1, T_t]$

**Problem** (Temporal SimRank Threshold Query)

**Given:**  $G, u, [T_1, T_t], \theta$

**Return:** node set  $\Omega$ , such that the SimRank of  $u$  and each node  $v \in \Omega$  is **greater than  $\theta$**  during the entire query interval  $[T_1, T_t]$

## Our Approach

- CrashSim Algorithm — static graphs
- CrashSim-T Algorithm — temporal graphs

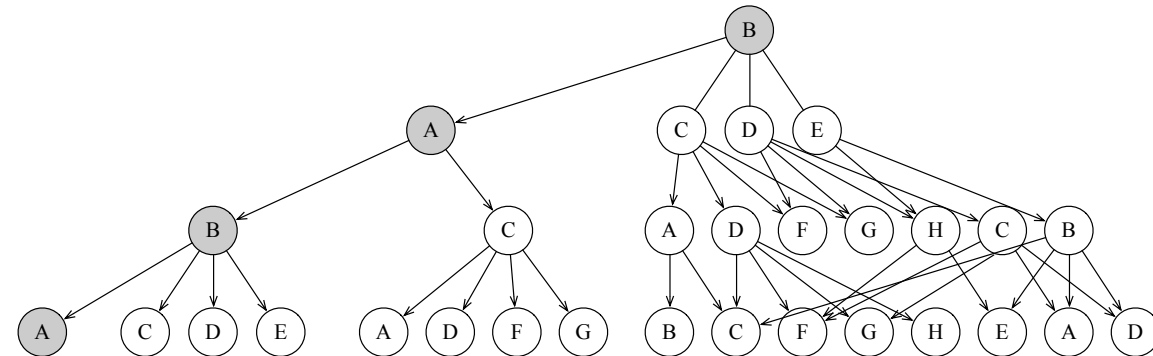
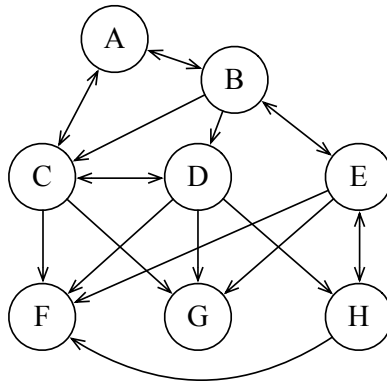
# CrashSim Algorithm

- Motivation

- ProbeSim (VLDB'17) is the state-of-the-art algorithm for SimRak computation over static graph

$$s(u, v) = \Pr [W(u) \text{ and } W(v) \text{ meet}]$$

$$= \sum_i \Pr [W(u) \text{ and } W(v) \text{ first meet at } u_i].$$



- Drawbacks

- redundant computations*
- the length of  $\sqrt{c}$ -walk determine the computation costs*

# CrashSim Algorithm

---

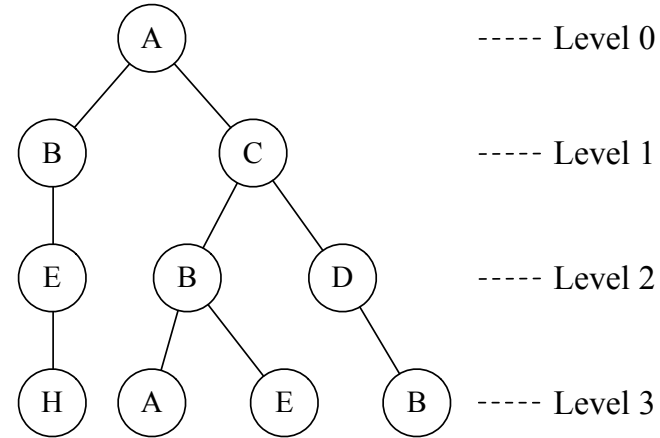
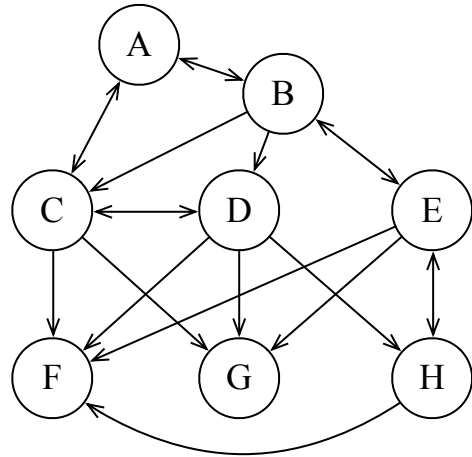
- Key idea
  - *Constrain the length of  $\sqrt{c}$ -walk to  $l_{max}$*
  - *A reverse reachable tree of source  $u$  with the limited length of  $\sqrt{c}$ -walk,  $l_{max}$*
  - *Still obtain SimRank estimators with the same guaranteed error bound of the ProbeSim*

**Problem** (Approximation Guarantee)

**Given:**  $G, u, \varepsilon, \delta$

**Return:**  $s(u, v)$  such that  $|s(u, v) - sim(u, v)| \leq \varepsilon$  with at least  $1 - \delta$  probability

# CrashSim Algorithm



$$U(0, A) = 1$$

$$U(1, B) = U(0, A) \times \frac{\sqrt{c}}{|I(B)|} = 1 \times \frac{0.5}{2} = 0.25$$

$$U(1, C) = U(0, A) \times \frac{\sqrt{c}}{|I(C)|} = 1 \times \frac{0.5}{3} = 0.167$$

$$U(2, E) = 0.0625, U(2, B) = 0.0417, U(2, D) = 0.0417$$

$$U(3, H) = 0.0156, U(3, A) = 0.0104$$

$$U(3, E) = 0.0104, U(3, B) = 0.0104$$

In the  $k$ -th trial,  $W(C) = (C, D, B, A)$

$$\begin{aligned} s_k(A, C) &= U(0, C) + U(1, D) + U(2, B) + U(3, A) \\ &= 0 + 0 + 0.0417 + 0.0104 = 0.0521 \end{aligned}$$

# CrashSim Algorithm

---

**Theorem 1.** For any node  $v \in \Omega$ ,  $\text{sim}(u, v)$  and its estimator  $s(u, v)$  satisfies  $\Pr \{ |\text{sim}(u, v) - s(u, v)| \leq \varepsilon \} \geq 1 - \delta$ , where  $s(u, v) = \frac{1}{n_r} \sum_{k=1}^{n_r} \sum_{i=2}^{\min(l, l_{\max})} P(v, W(u, i))$ ,  
 $l_{\max} = \frac{1 + \sqrt{c}}{(1 - \sqrt{c})^2}$ ,  $n_r = \frac{3c}{(\varepsilon - p\varepsilon_t)^2} \log \frac{n}{\delta}$ ,  $\varepsilon_t = (\sqrt{c})^{l_{\max}}$ ,  
 $p = \sum_{k=1}^{l_{\max}} (\sqrt{c})^{k-1} (1 - \sqrt{c})$ .

**Time Complexity:**  $O(m + n_r \cdot |\Omega|) = O(m + \frac{3c}{(\varepsilon - p\varepsilon_t)^2} \log \frac{n}{\delta} \cdot |\Omega|)$ .

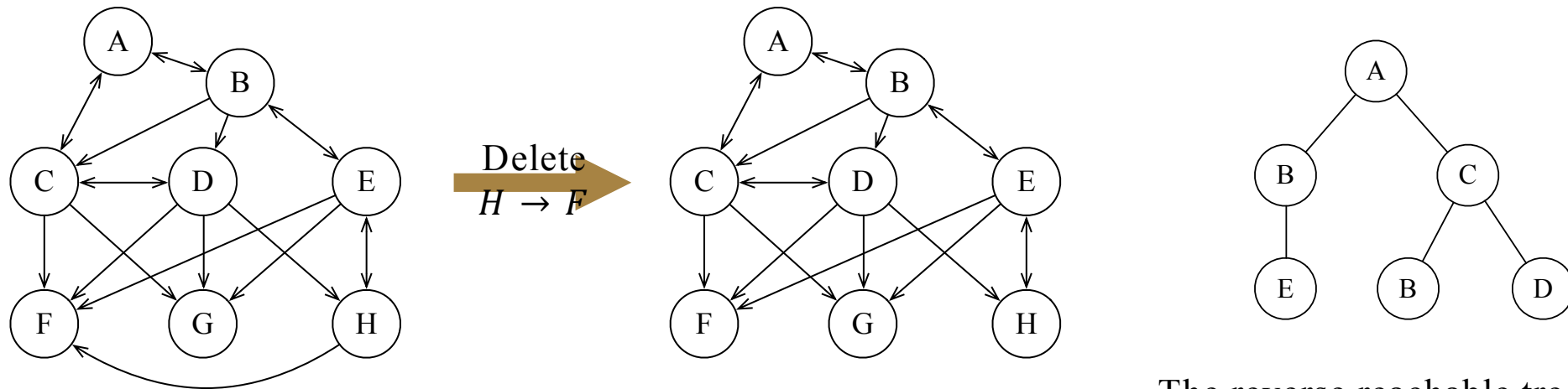
# CrashSim-T Algorithm

---

- Two opportunities
  - *Unnecessary to compute the SimRank between  $u$  and the candidate node set  $\Omega$  at each time instant*
  - *The size of node set  $\Omega$  can only gradually reduce over time*
- CrashSim naturally supports the computation of SimRank of the source  $u$  and a partial set of nodes.

# CrashSim-T Algorithm --- Delta Pruning

- Affected area of a changed edge  $x \rightarrow y$ 
  - *The altered nodes in the reverse reachable tree of  $u$*
  - *$l_{max} - 1$  length reachable nodes of  $y$*
- Delta pruning: ignore the nodes of an unaffected area



The reverse reachable tree of A remains unchanged.



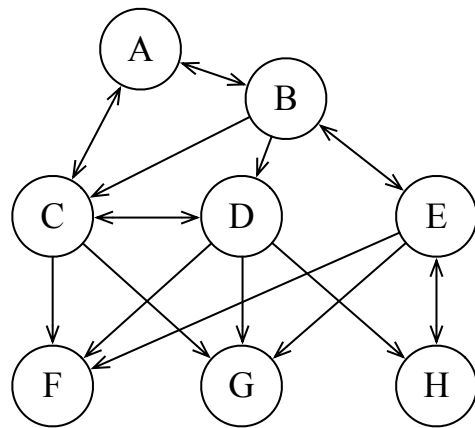
# CrashSim-T Algorithm --- Difference Pruning

- Related area: the  $l_{max}$  length reverse reachable tree of  $u$  and  $v$

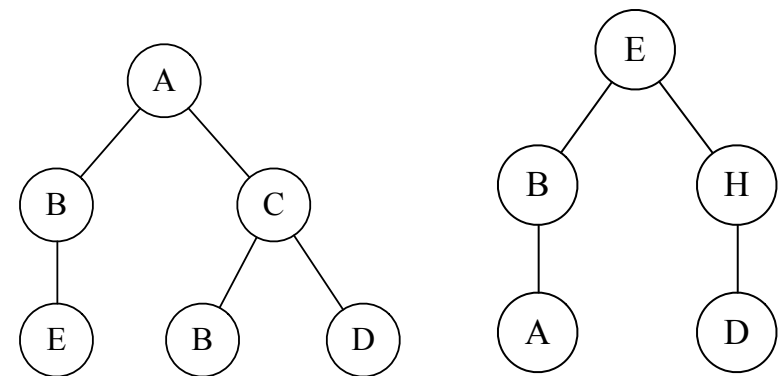
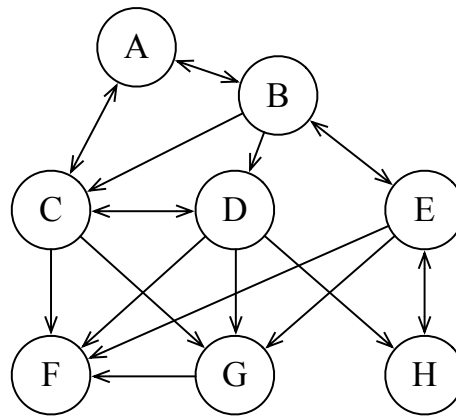
$$s(u, v) = Pr[W(u) \text{ and } W(v) \text{ meet}]$$

$$= \sum_i Pr[W(u) \text{ and } W(v) \text{ first meet at } u_i].$$

- Difference pruning: filter out those nodes whose related area is unchanged



Add  
 $G \rightarrow F$



The reverse reachable tree of A and E remains unchanged.

# CrashSim-T Algorithm

---

- Main idea
  - *Check whether the conditions of delta and difference pruning are satisfied*
  - *If so, disregard those nodes as part of the candidate node set*
  - *Invoke CrashSim algorithm to compute  $u$  and residual nodes*
  - *According to different query requirements to filter out unsatisfied nodes*

---

# Experiments and Conclusion

- Experimental Evaluation
- Conclusion

# Experimental Evaluation

---

- Datasets

Datasets	Type	n	m	t
AS-733	Undirected	6,474	13,233	733
AS-Caidi	Directed	26,475	106,762	122
Wiki-Vote	Directed	7,155	103,689	100
HepTh	Undirected	9,877	25,998	100
HepPh	Directed	34,546	421,578	100

- Comparison baselines

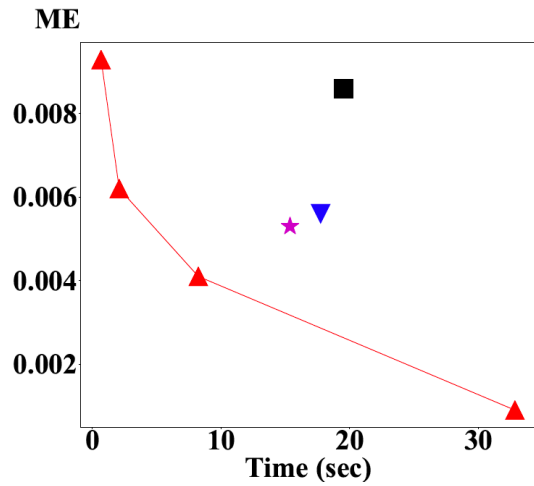
- SLING (SIGMOD'16), ProbeSim (VLDB'17), READS (VLDB'17)

- Setting and metrics

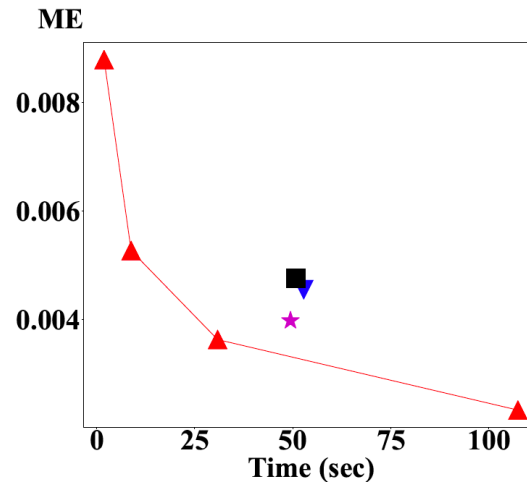
- $\varepsilon$  varies from 0.0125, 0.025, 0.05 to 0.1
- $ME = \max|s(u, v) - sim(u, v)| (v \in V)$
- $precision = \frac{v(k_1) \cap v(k_2)}{\max(k_1, k_2)}$

# Experimental Evaluation

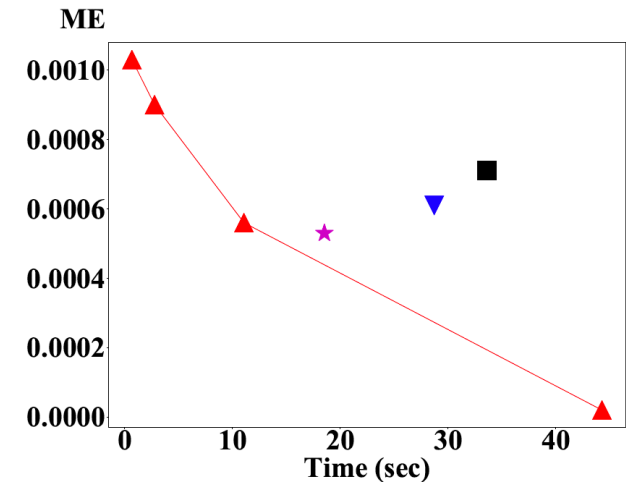
▲ CrashSim    ★ ProbeSim    ▼ SLING    ■ READS



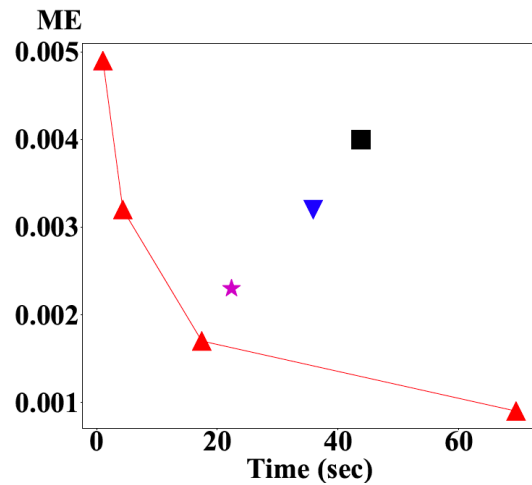
(a) AS-733



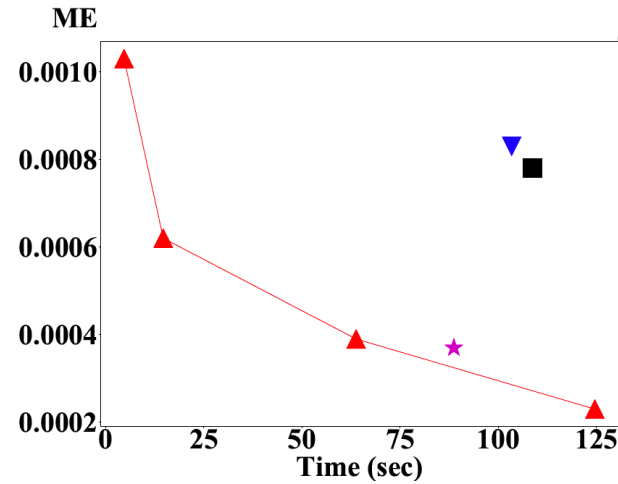
(b) AS-Caidi



(c) Wiki-Vote

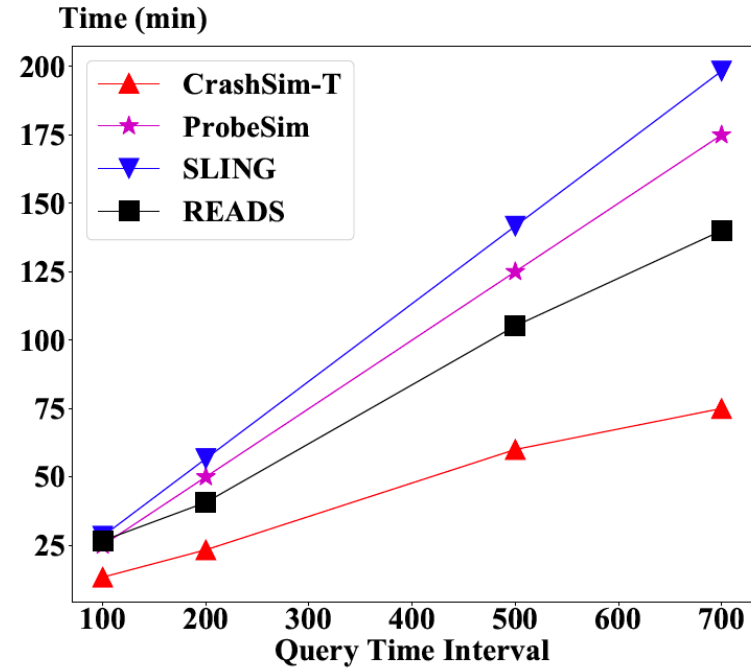
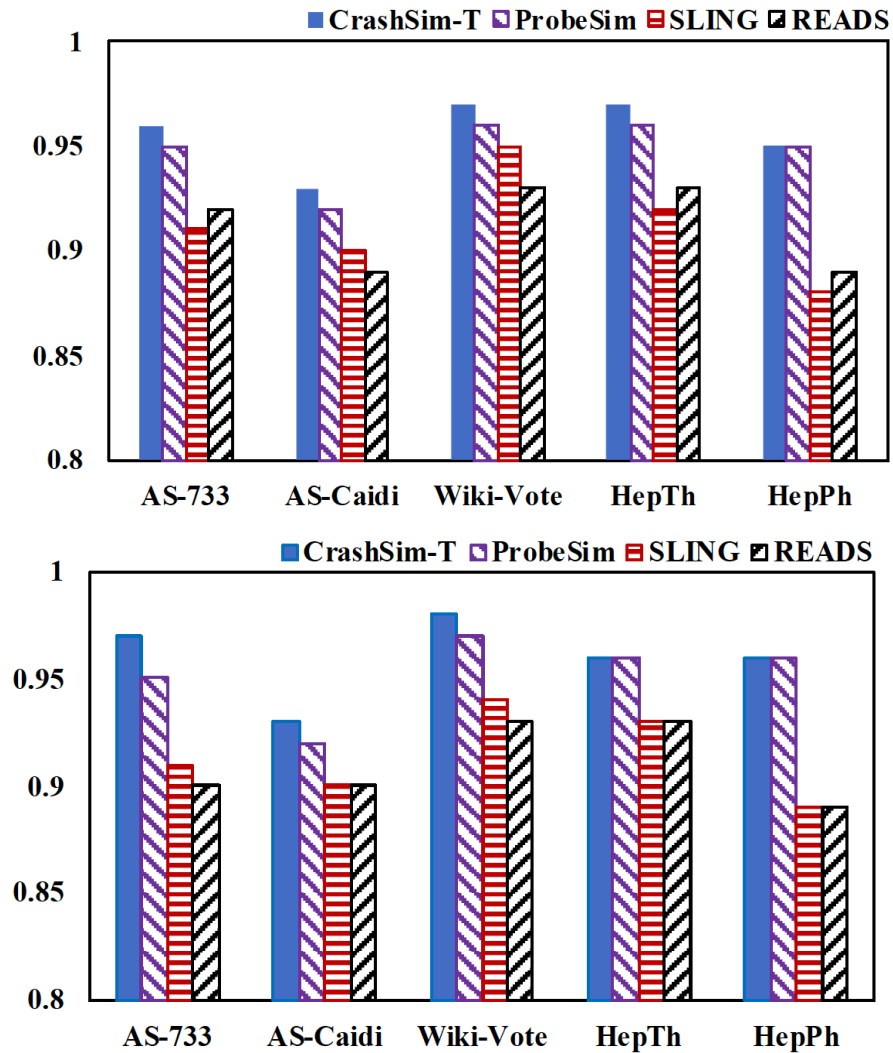


(d) HepTh



(e) HepPh

# Experimental Evaluation



The impact of the query interval on the response time of the algorithms

# Conclusion

---

- Propose **CrashSim** algorithm, an index-free algorithm for single-source and partial SimRank computation in static graphs
- Introduce **CrashSim-T** --- an extension to CrashSim to solve SimRank queries over temporal graphs
- Experiments show that both CrashSim and CrashSim-T outperform the state-of-the-art algorithms.

**THANKS**

**Thanks.**

Mo Li

limo\_neucse@hotmail.com