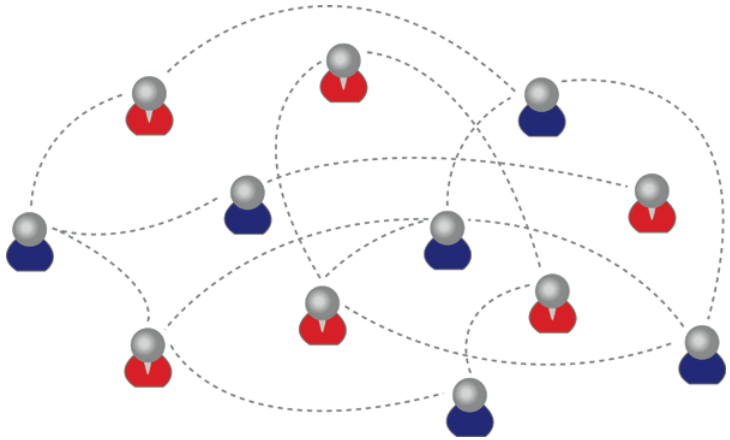
Incremental and Accuracy-aware Personalized Pagerank through Scheduled Approximation

Fanwei Zhu, Yuan Fang, Kevin Chang, Jing Ying

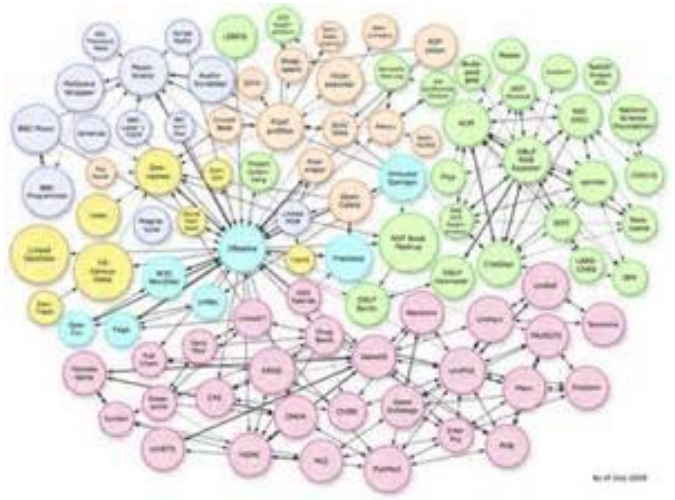
Zhejiang University
University of Illinois at Urbana-Champaign

Motivation:
Useful for ranking, expensive to compute

- Graphs are everywhere, calling for graph-based ranking algorithm



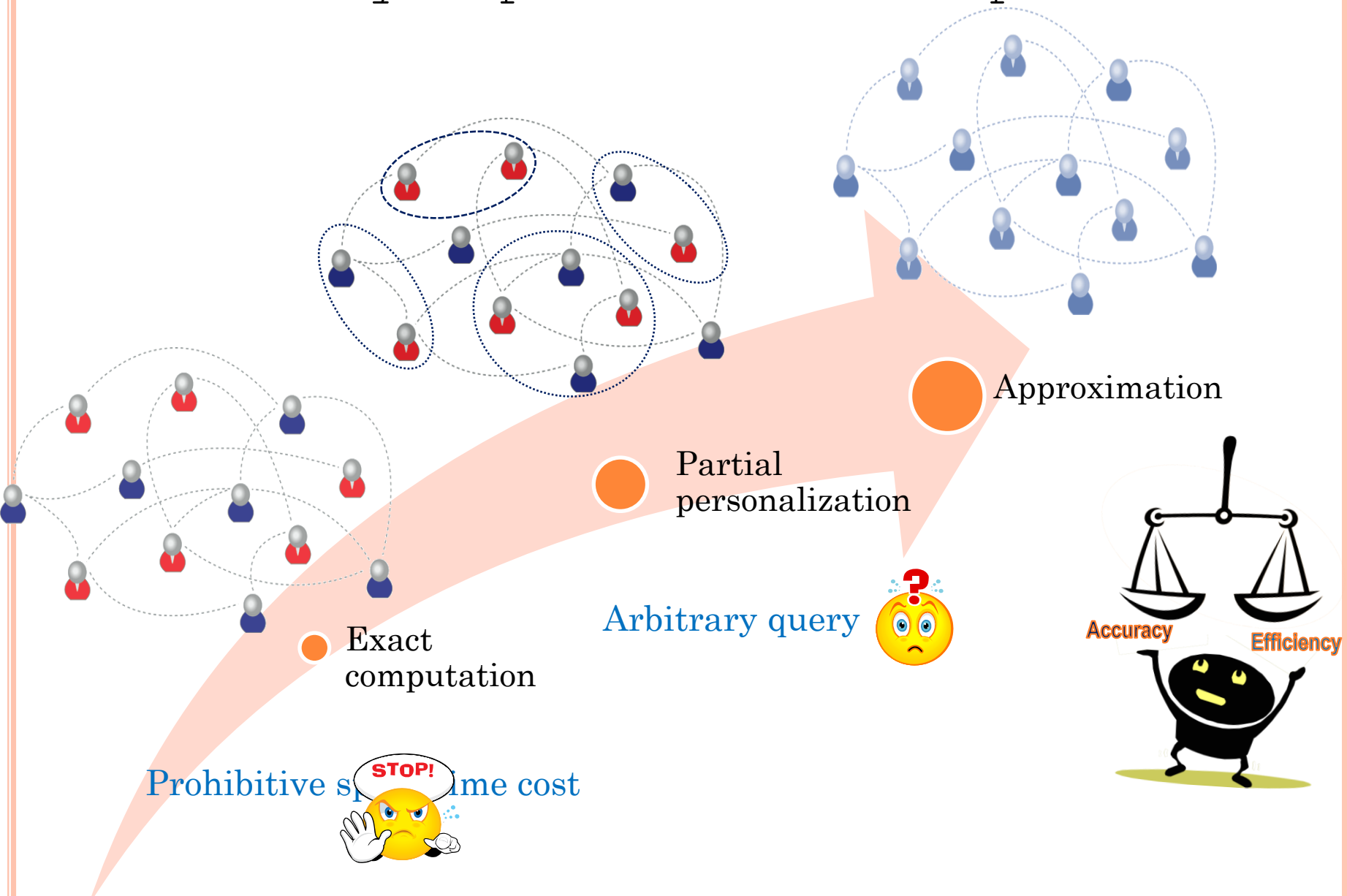
social network



DBLP citation network

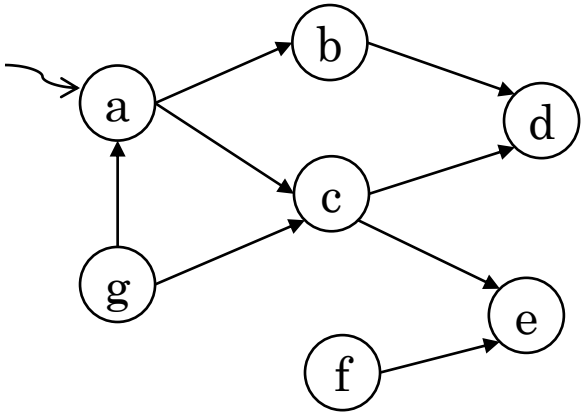
- Personalized Pagerank (PPV)
  - Effective for ranking
  - Expensive to compute

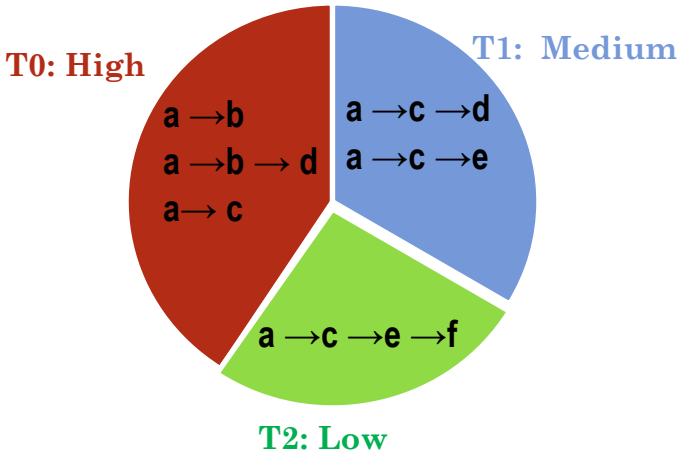
Focus:
Efficiency aspect of PPV computation



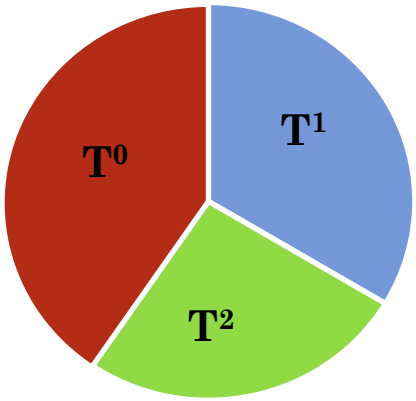
Key insight: Scheduled approximation

- Partitioning by importance





- Prioritizing computation

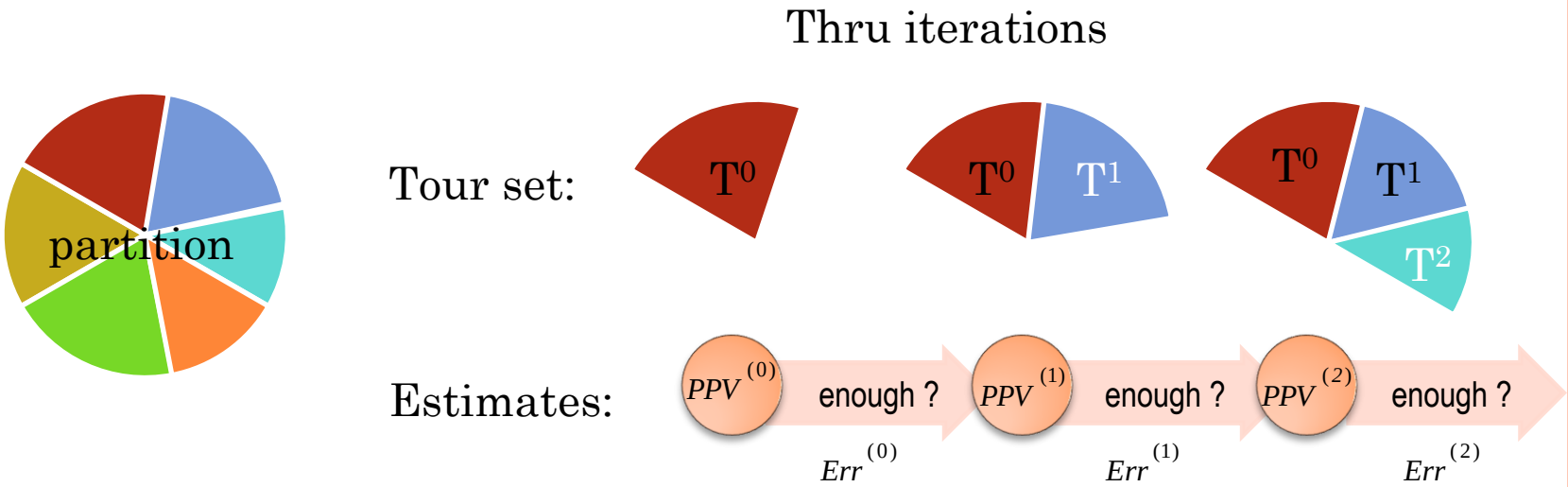


b	0.098
c	0.056
d	0.063
e	0.021
f	0.016

$$PPV_a^{(0)} \quad PPV_a^{(1)} \quad PPV_a^{(2)} = PPV_a$$

Novelty: Incremental & accuracy-aware

- Incremental query processing



- Accuracy aware

$$Err^{(i)} = \sum_q |\, PPV(p) - PPV^{(i)}(p) \,| = 1 - \sum_q PPV^{(i)}(p)$$

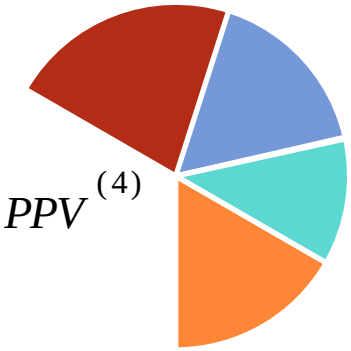
sum of current estimates

Challenges: Efficient implementation

- Challenge 1: How to effectively partition tours?

importance of tours w.r.t query?

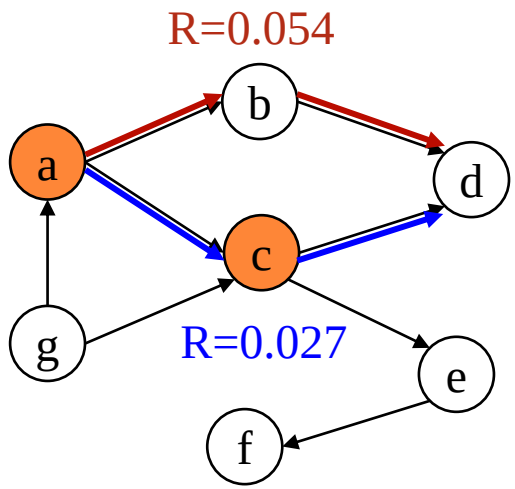
- Challenge 2: How to efficiently compute each PPV increment?



$= inc^0 + inc^1 + inc^2 + inc^3$

Solution: Hub-based realization

- Hub nodes
  - Discriminating: high out-degree decaying reachability
  - Sharing: popularity segments shared by tours



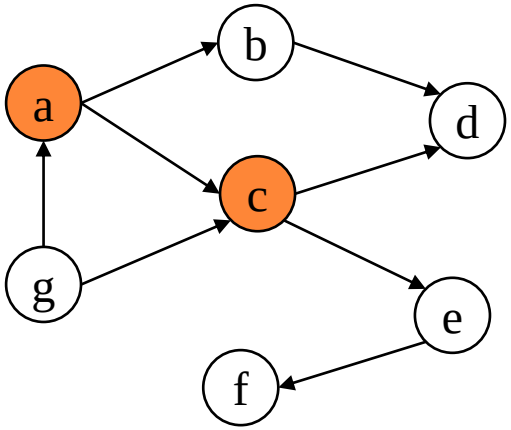
$$H = \{a, c\}$$

$$R(a \to b \to d) = \frac{1}{2} \times 0.85^2 \times 0.15 = 0.054$$

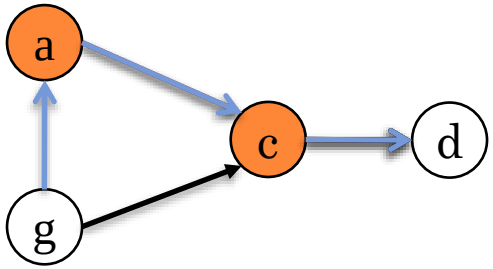
$$R(a \to c \to d) = \frac{1}{2} \times 0.85^2 \times \frac{1}{2} \times 0.15 = 0.027$$

Solution: Hub-based realization

- Hub nodes
  - Discriminating: high out-degree decaying reachability
  - Sharing: popular segments shared by tours



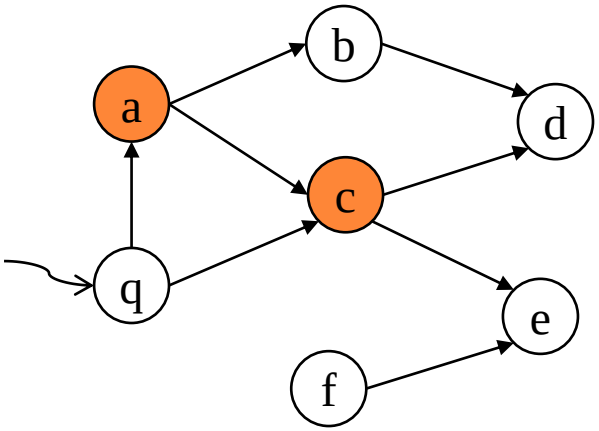
$H = \{a, c\}$



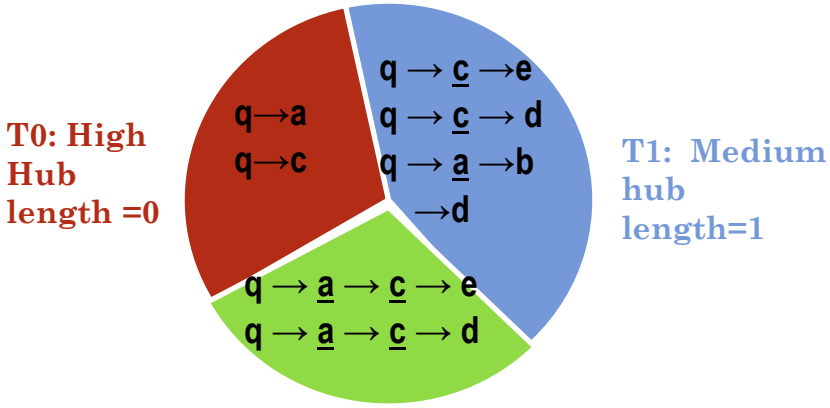
$c \to d$ shared by $a \to c \to d$, $g \to c \to d$, $g \to a \to c \to d$

Challenge 1: Discriminating provides partition metric

- More hubs, less important
- Partition tours by hub length (# of hubs)

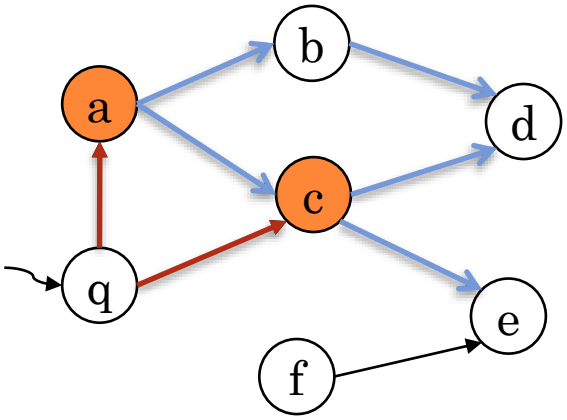


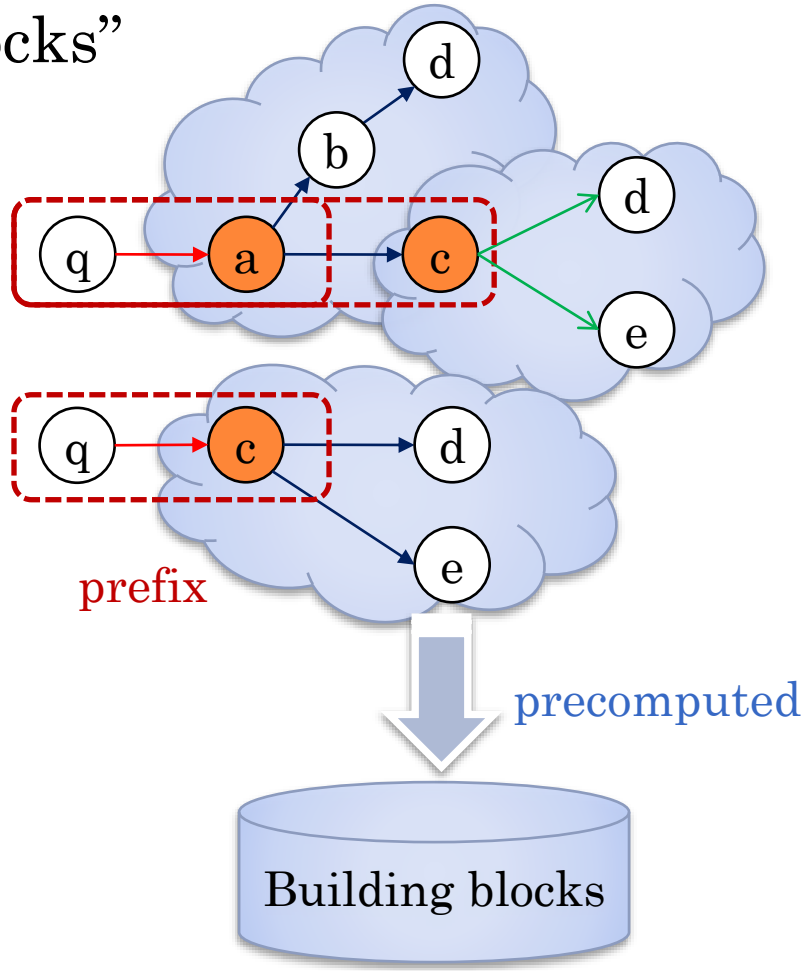
$H = \{a, c\}$



Challenge 2: Sharing enables reusing overlaps

- Reuse "prefix" among iterations
- Precompute "building blocks"

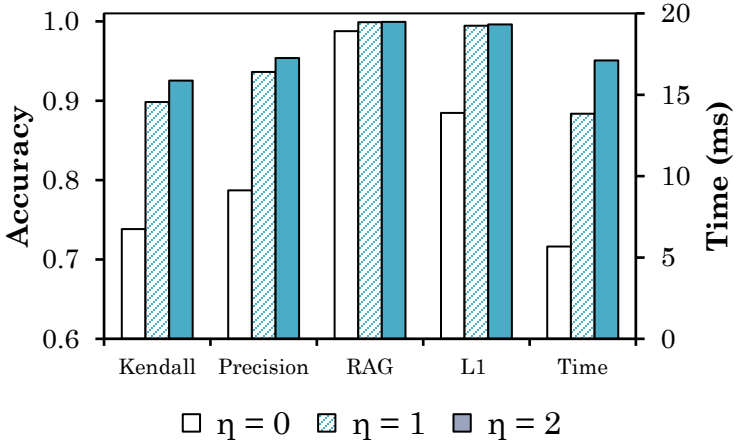




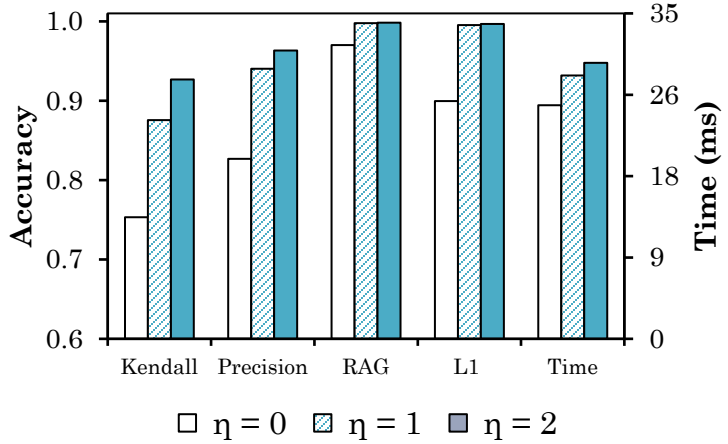
Results: Fast with accuracy control

- More iterations render better accuracy

(a) DBLP

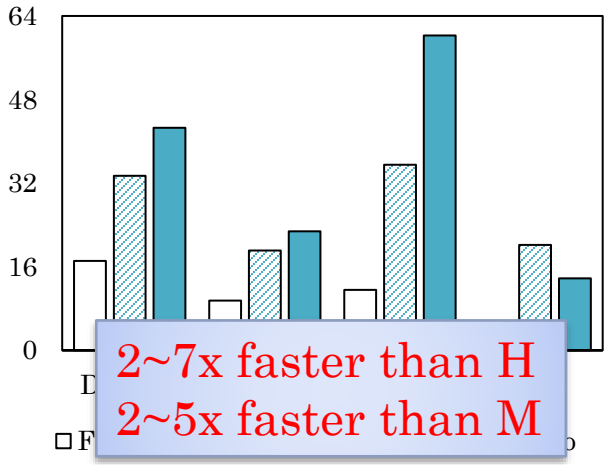


(b) LiveJournal

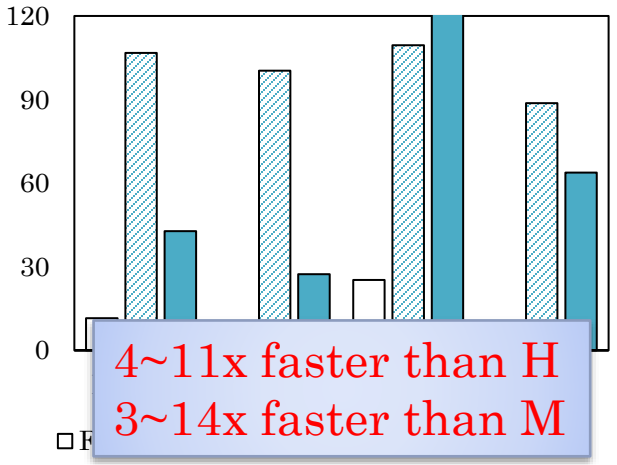


- Faster online/offline computation

(a) Online query time



(b) Offline precomputation time



Conclusion and future work

- Conclusion
  - a scheduled approximation strategy to approximate PPVs
  - an efficient hub-based realization
  - up to 7x faster with accuracy control

- Future work
  - automatic parameter configuration
  - tackling dynamic, evolving graph
  - generalizing to other graph algorithms

Thank you!