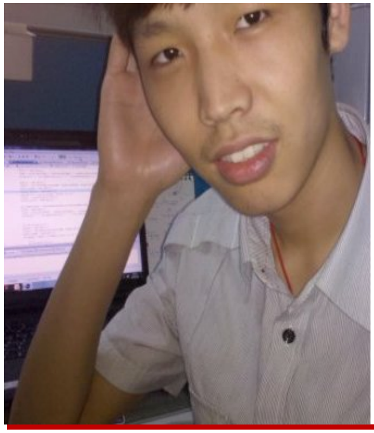
QUINT: On Query-Specific Optimal Networks

Presenter: **Liangyue Li**

Joint work with

Yuan Yao
(NJU)



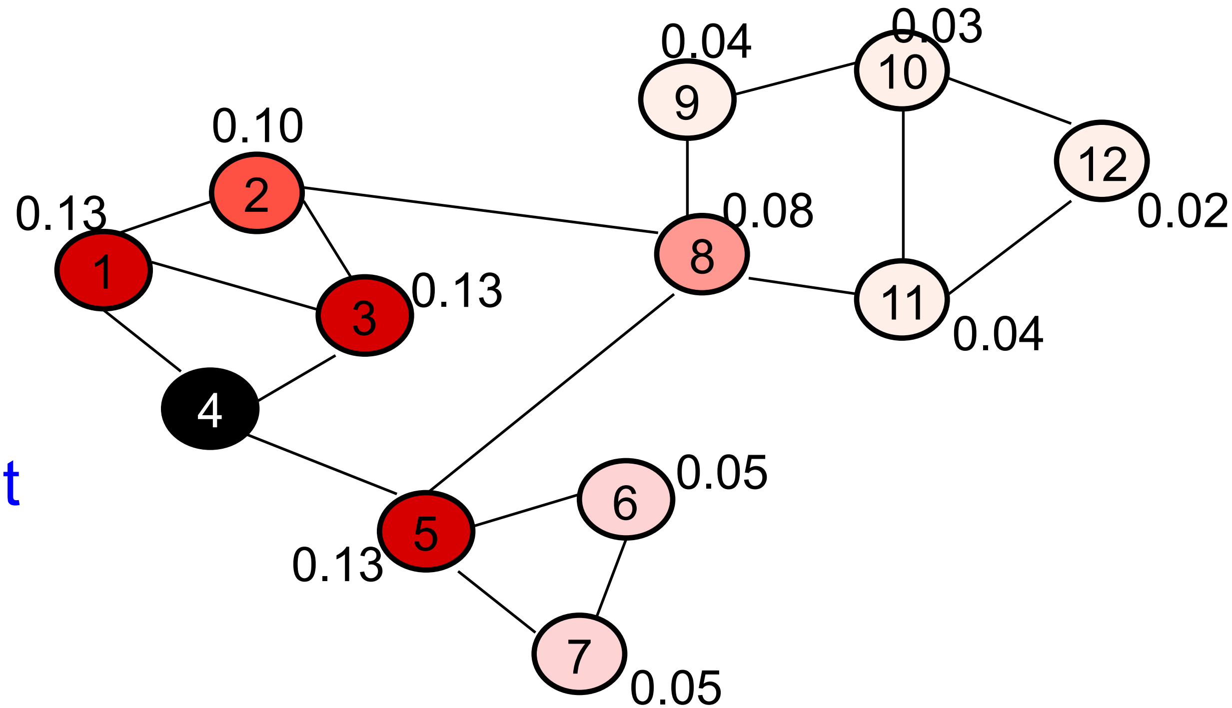
Jie Tang
(Tsinghua)

Wei Fan
(Baidu)

Hanghang Tong
(ASU)

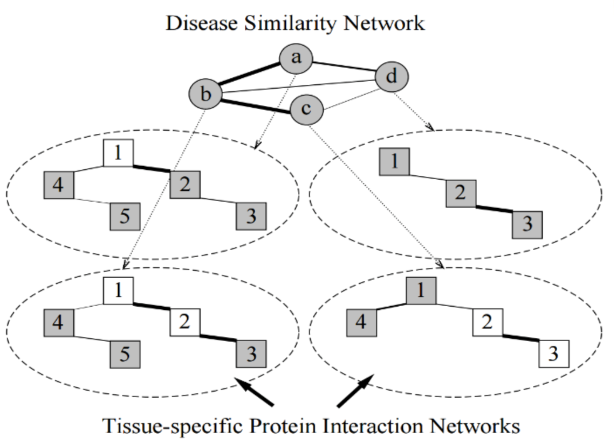
Node Proximity: What?

- **Node proximity**: the closeness (a.k.a., relevance, or similarity) between two nodes



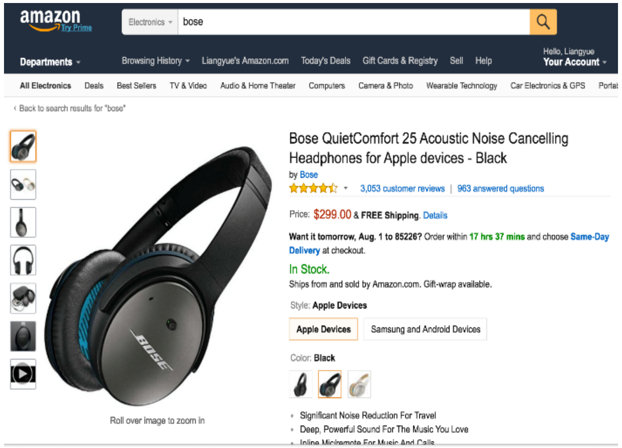
What is the closest node to 4?

Node Proximity: Why?



Biology [Ni+]

Social Network [Lerman+]



‹ Back to search results for "bose"

Bose QuietComfort 25 Acoustic Noise Cancelling Headphones for Apple devices - Black

by Bose

3,053 customer reviews | 963 answered questions

Price: $299.00 & FREE Shipping. Details

Want it tomorrow, Aug. 1 to 85226? Order within 17 hrs 37 mins and choose Same-Day Delivery at checkout.

In Stock.

Ships from and sold by Amazon.com. Gift-wrap available.

Style: Apple Devices

Apple Devices | Samsung and Android Devices

Color: Black

Roll over image to zoom in

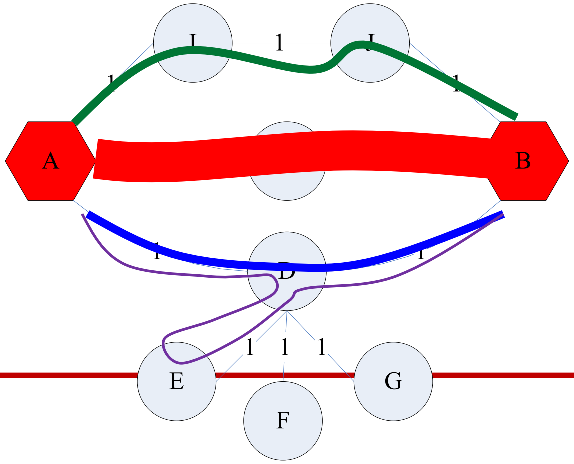
- Significant Noise Reduction For Travel
- Deep, Powerful Sound For The Music You Love
- Inline Mic/remote For Music And Calls

E-commerce [Chen+]

Disaster Mgtm [Zheng+]

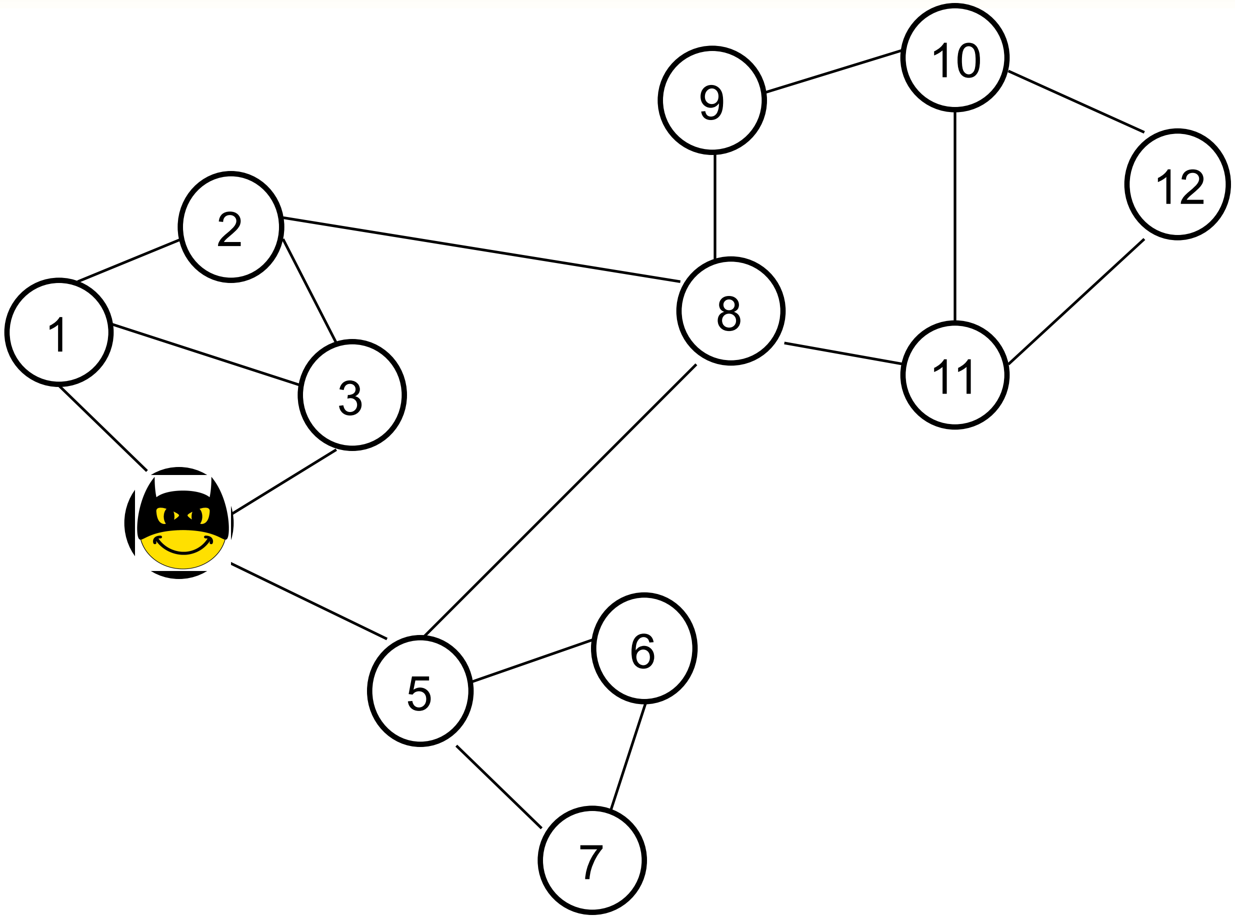
Node Proximity: How?

- **Random Walk with Restart** (RWR)
  - **Idea**: summarize multiple weighted relationships btw nodes
  - **Variants**:
    - Electric networks: SAEC[Faloutsos+]
    - Katz [Katz], [Huang+]
    - Matrix-Forest-based Alg [Chobotarev+]



Prox (A, B) =
Score (Red Path) +
Score (Green Path) +
Score (Blue Path) +
Score (Purple Path) + …

Node Proximity: RWR



Node Proximity -- RWR

- **Detail:** a random walker starts from *s*
  - (a) transmit to one neighbor with $p \sim cA_{ij}$
  - (b) go back to *s* with prob $(1-c)$
- **Formulation**

$$\underset{\text{Ranking vector}}{\underset{\downarrow}{\mathbf{r}_s}} = c\underset{\text{Adjacent matrix}}{\underset{\downarrow}{\mathbf{A}}}\mathbf{r}_s + \underset{\text{Restart prob}}{\underset{\downarrow}{(1-c)}}\underset{\text{Starting vector}}{\underset{\downarrow}{\mathbf{e}_s}}$$

- **Assumption**
  - How to best leverage the fixed input graph $\mathbf{A}$

Node Proximity: Learning RWR

$$\mathbf{Q} = (\mathbf{I} - c\mathbf{A})^{-1}$$

- **Goal**
  - Use side information to learn better graph
  - Side info: user feedback, node attributes
- **Key Idea:** Infer optimal edge weights

$$\min_{w} \underbrace{\|w\|^2}_{\text{Map edge attributes to weights}} + \lambda \sum_{x\in\mathcal{P}, y\in\mathcal{N}} \underbrace{h(\mathbf{Q}(y,s) - \mathbf{Q}(x,s))}_{\text{Match user preferences}}$$

- **Limitation:** Fixed topology

J. Tang, T. Lou and J. Kleinberg. Transfer Link Prediction across Heterogeneous Social Networks. TOIS, 2015.

L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. WSDM, 2011.

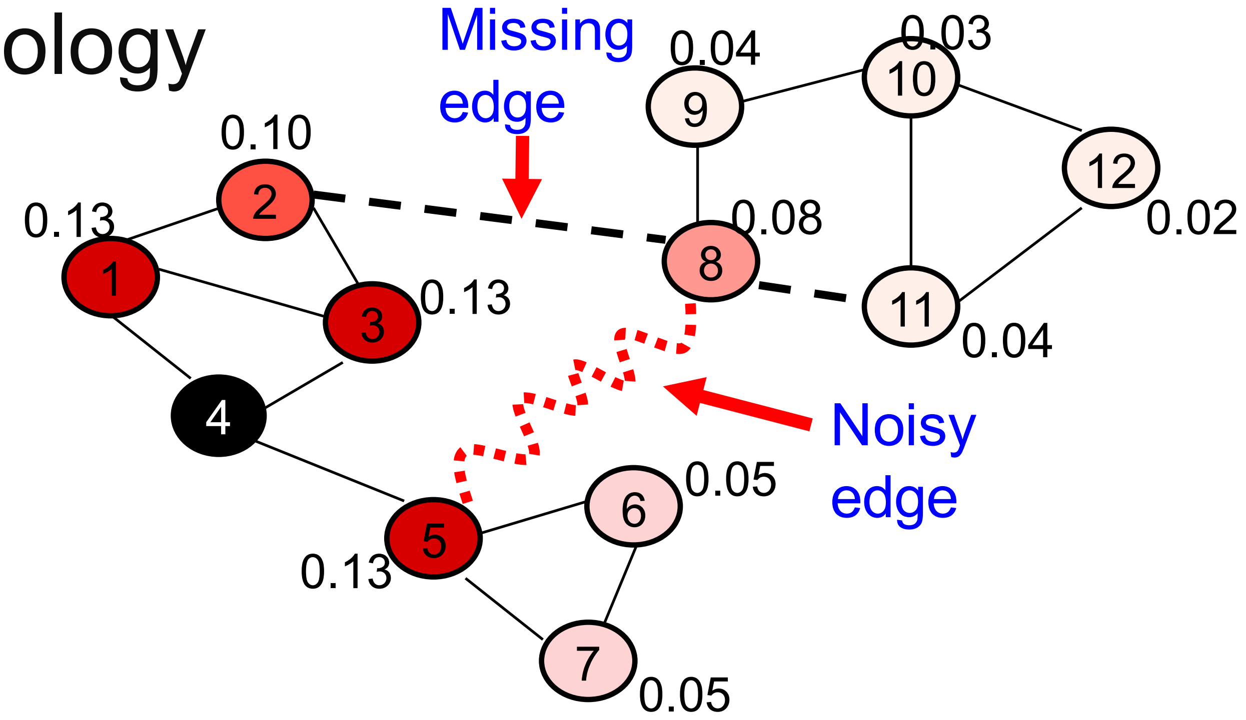
A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked entities. KDD, 2006.

Algorithmic Questions

- Q1: optimal weights or optimal topology?
- Q2: one-fits-all or one-fits-one?
- Q3: offline learning or online learning?

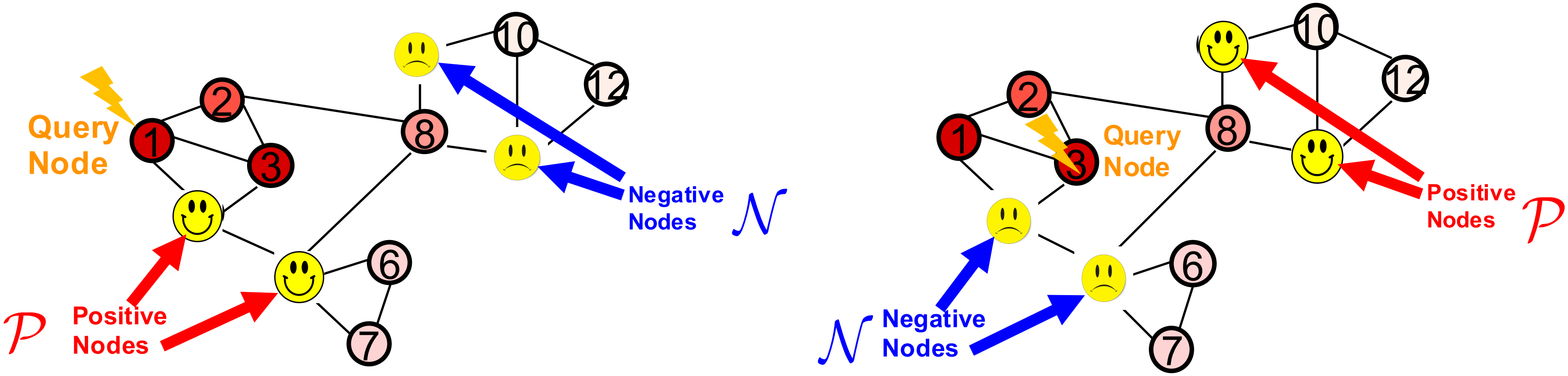
Q1: Optimal Weights or Topology?

- **Observation**: real network is noisy and incomplete
- **Challenge**: learn optimal weights and topology



Q2: One-fits-all, or one-fits-one?

- **Observation**: optimal network for different queries might be different

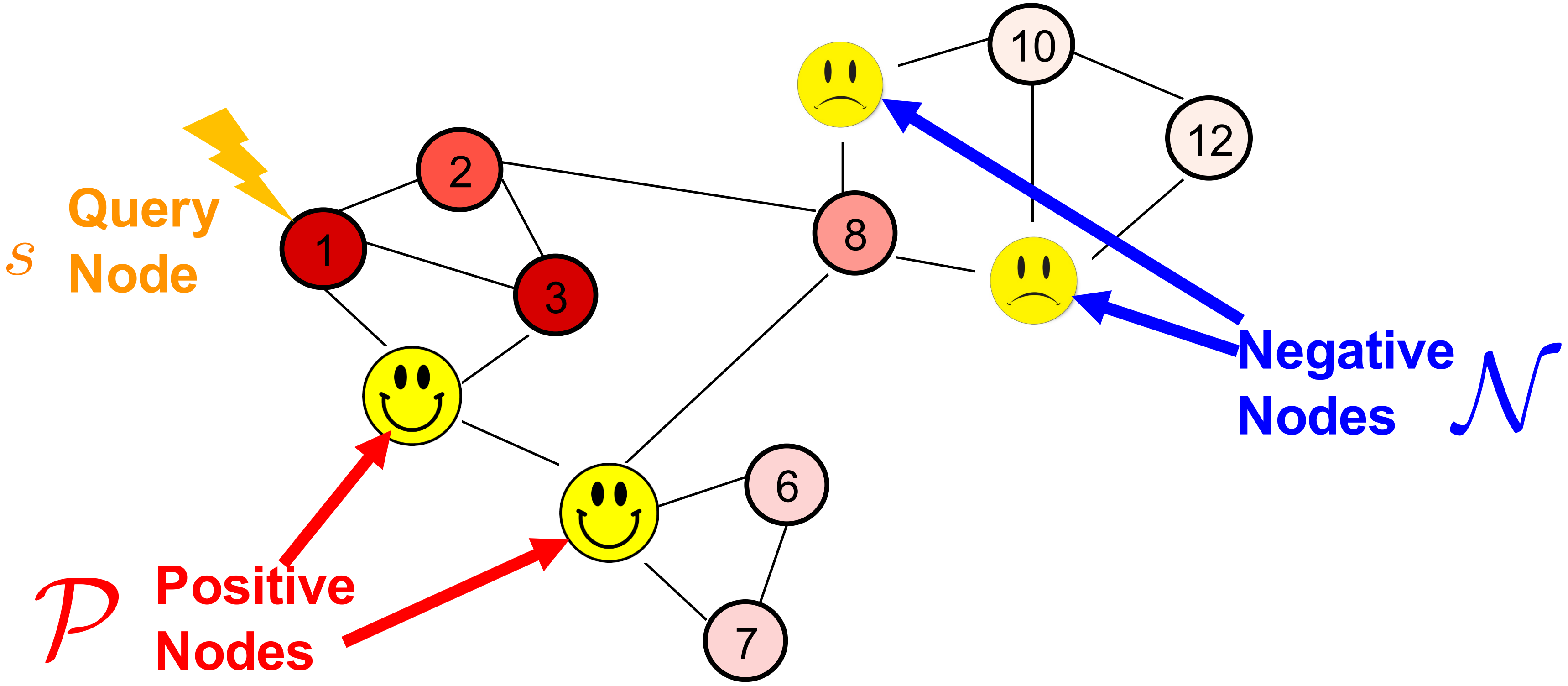


- **Challenge**:
  - How to tailor learning for each query

Q3: Offline or Online Learning

- **Observation**:
  - Learning RWR: costly iterative sub-routine to compute a single gradient vector
  - Learning topology: parameter space expands to $O(n^2)$
  - One-fits-one: one optimal network for each query
- **Challenge**:
  - How to perform query-specific online learning?

Query-specific Optimal Network Learning



Given: An input network $\mathbf{A}$, a query node s, positive nodes $\mathcal{P}$ and negative nodes $\mathcal{N}$

Learn: An optimal network $\mathbf{A}_s$ specific to the query

Roadmap

- Motivations
- Proposed Solutions: **QUINT**
- Empirical Evaluations
- Conclusions

$\mathbf{Q} = (\mathbf{I} - c\mathbf{A})^{-1}$

QUINT - Formulations

- **Optimization Formulation** (hard version)

Matching Input Network

Positive nodes

Negative nodes

$$\arg\min_{\mathbf{A}_s} \quad \|\mathbf{A}_s - \mathbf{A}\|_F^2$$

$$\text{s.t.,} \quad \mathbf{Q}(x, s) > \mathbf{Q}(y, s), \forall x \in \mathcal{P}, \forall y \in \mathcal{N}$$

Matching Preference(hard)

- **Remarks**
  - Larger parameter space $O(n^2)$
  - Query-specific Optimal Network
  - No exception is allowed in the constraint

$\mathbf{Q} = (\mathbf{I} - c\mathbf{A})^{-1}$

QUINT - Formulations

- **Optimization Formulation** (soft version)

$$\arg\min_{\mathbf{A}_s} \mathcal{L}(\mathbf{A}_s) = \lambda \|\mathbf{A}_s - \mathbf{A}\|_F^2 + \sum_{x \in \mathcal{P}, y \in \mathcal{N}} g(\mathbf{Q}(y,s) - \mathbf{Q}(x,s))$$

Loss function

Penalty to the violation of preferences

- **Remarks**
  - Characteristic $\mathbf{Q}(y,s) < \mathbf{Q}(x,s) \Rightarrow g(\cdot) = 0$

    $\mathbf{Q}(y,s) > \mathbf{Q}(x,s) \Rightarrow g(\cdot) > 0$
  - Wilcoxon-Mann-Whitney (WMW) loss

$$\mathbf{Q} = (\mathbf{I} - c\mathbf{A})^{-1}$$

QUINT -- Optimization

- **Gradient Descent Based Solution**
  - Gradient

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{A}_s)}{\partial \mathbf{A}_s} &= 2\lambda(\mathbf{A}_s - \mathbf{A}) + \sum_{x\in\mathcal{P},y\in\mathcal{N}} \frac{\partial g(\mathbf{Q}(y,s)-\mathbf{Q}(x,s))}{\partial \mathbf{A}_s} \\ &= 2\lambda(\mathbf{A}_s - \mathbf{A}) + \sum_{x,y} \frac{\partial g(d_{yx})}{\partial d_{yx}} \Big(\underset{\overset{\uparrow}{\text{Differentiable}}}{\frac{\partial \mathbf{Q}(y,s)}{\partial \mathbf{A}_s}} - \boxed{\frac{\partial \mathbf{Q}(x,s)}{\partial \mathbf{A}_s}} \Big)\end{aligned}$$

  - Derivative of an Inverse

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{A}_s(i,j)} = -\mathbf{Q}\frac{\partial(\mathbf{I}-c\mathbf{A}_s)}{\partial \mathbf{A}_s(i,j)}\mathbf{Q} = c\mathbf{Q}\mathbf{J}^{ij}\mathbf{Q}$$

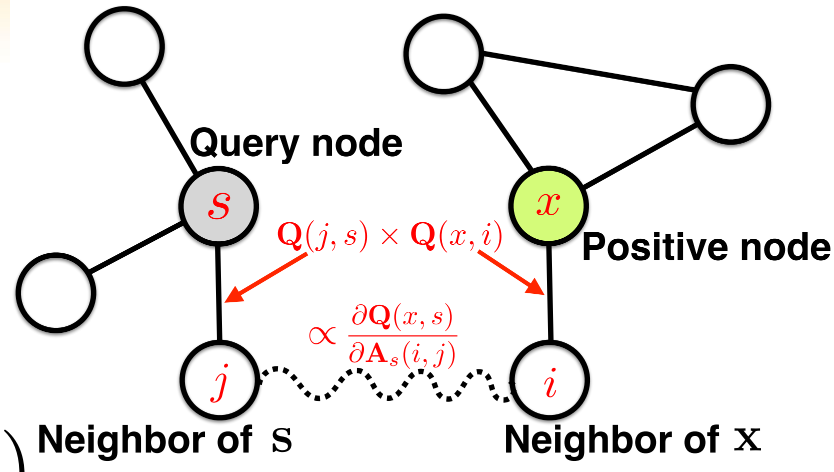
$$\frac{\partial \mathbf{Q}(x,s)}{\partial \mathbf{A}_s(i,j)} = c\mathbf{Q}(x,i)\mathbf{Q}(j,s)$$

$$\mathbf{Q} = (\mathbf{I} - c\mathbf{A})^{-1}$$

QUINT -- Optimization

- **Intuition**

$$\frac{\partial \mathbf{Q}(x,s)}{\partial \mathbf{A}_s(i,j)} = c\mathbf{Q}(x,i)\mathbf{Q}(j,s)$$



- **Complexity**

$$O(T_1|\mathcal{P}| \cdot |\mathcal{N}|(T_2 m + n^2))$$

- **Observation**
  - Usually $T_1, T_2, |\mathcal{P}|, |\mathcal{N}| \ll m, n$
  - Complexity: quadratic

Q: how to scale up?

$\mathbf{Q} = (\mathbf{I} - c\mathbf{A})^{-1}$

QUINT – Scale-up

- **Key idea:** Optimal network is rank-one perturbation to original network
- **Details**:

$$\begin{aligned} \arg\min_{\mathbf{f},\mathbf{g}} \mathcal{L}(\mathbf{f},\mathbf{g}) &= \lambda \|\mathbf{f}\mathbf{g}'\|_F^2 + \beta(\|\mathbf{f}\|^2 + \|\mathbf{g}\|^2) \\ &+ \sum_{x\in\mathcal{P},y\in\mathcal{N}} g(\mathbf{Q}(y,s) - \mathbf{Q}(x,s)) \end{aligned}$$

- **Optimization**: alternating gradient descent
- **Complexity**: $O(T_1|\mathcal{P}| \cdot |\mathcal{N}|(T_2 m + n))$

QUINT – Variant #1

- **Key idea**: apply Taylor Approximation for $\mathbf{Q}$
- **Details**:

$$\begin{aligned}\mathbf{Q} &= (\mathbf{I} - c\mathbf{A})^{-1} \\ &\approx \mathbf{I} + \sum_{i=1}^{k} c^k \mathbf{A}^k\end{aligned}$$

- **Complexity:** using 1st order Taylor

$$O(T_1|\mathcal{P}| \cdot |\mathcal{N}|n)$$

- **Benefit**: accessing $\mathbf{Q}(i,j)$ faster

QUINT – Variant #2

- **Key idea**: Only update neighborhood of the query node and the pos/neg nodes (*Localized Rank-One Perturbation*)
- **Complexity**

$$O(T_1|\mathcal{P}| \cdot |\mathcal{N}| \max(|\mathbb{N}(s)|, |\mathbb{N}(\mathcal{P},\mathcal{N})|))$$

$\mathbb{N}(s)$: Neighbors of *s*

$\mathbb{N}(\mathcal{P},\mathcal{N})$: Neighbors of *pos/neg nodes*

$$\max(|\mathbb{N}(s)|, |\mathbb{N}(\mathcal{P},\mathcal{N})|) \ll n$$

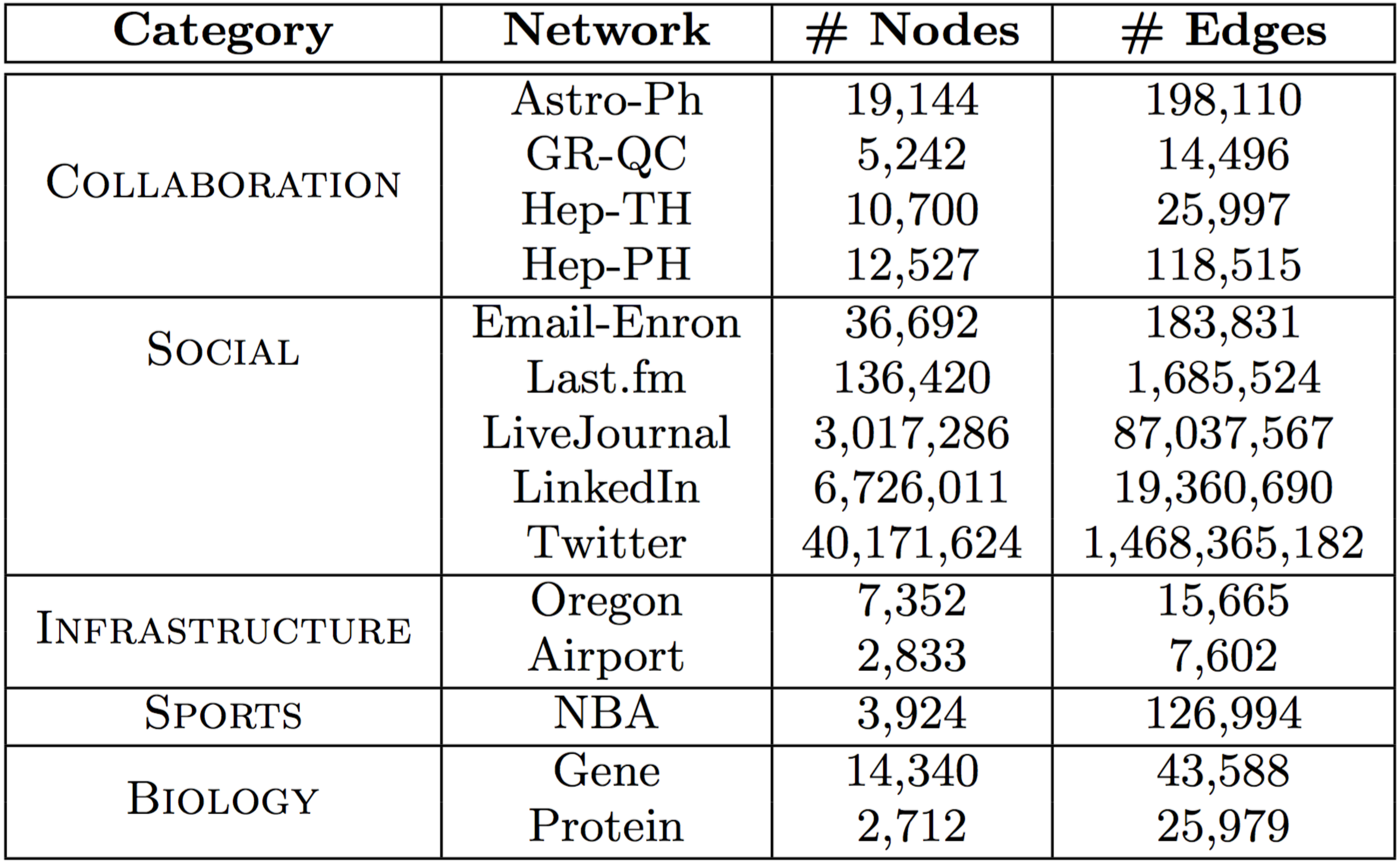
- **Benefit:** usually sub-linear to *n*

Roadmap

- Motivations
- Proposed Solutions: **QUINT**
- Empirical Evaluations
- Conclusions

Datasets

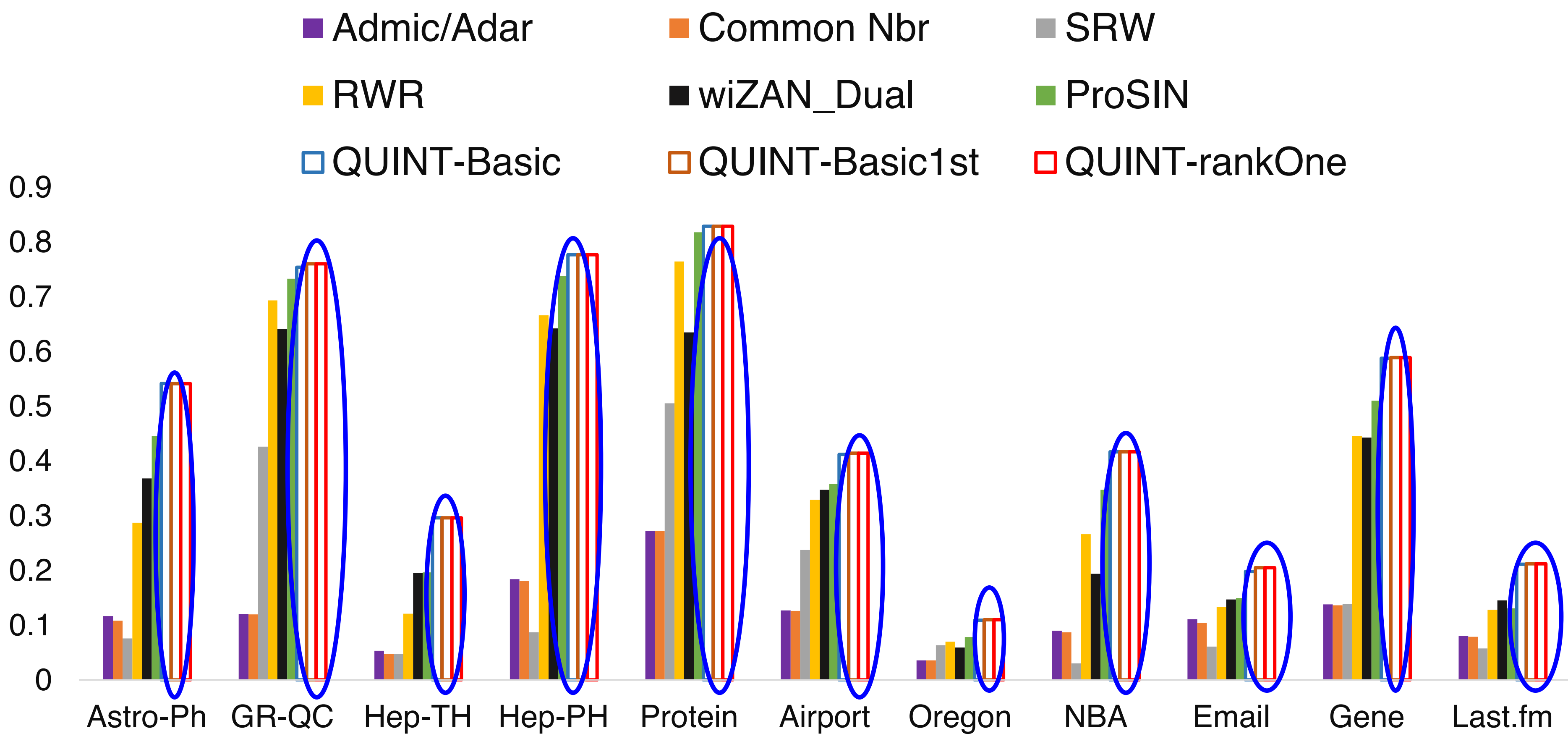
10+ diverse networks



Category	Network	# Nodes	# Edges
Collaboration	Astro-Ph	19,144	198,110
	GR-QC	5,242	14,496
	Hep-TH	10,700	25,997
	Hep-PH	12,527	118,515
Social	Email-Enron	36,692	183,831
	Last.fm	136,420	1,685,524
	LiveJournal	3,017,286	87,037,567
	LinkedIn	6,726,011	19,360,690
	Twitter	40,171,624	1,468,365,182
Infrastructure	Oregon	7,352	15,665
	Airport	2,833	7,602
Sports	NBA	3,924	126,994
Biology	Gene	14,340	43,588
	Protein	2,712	25,979

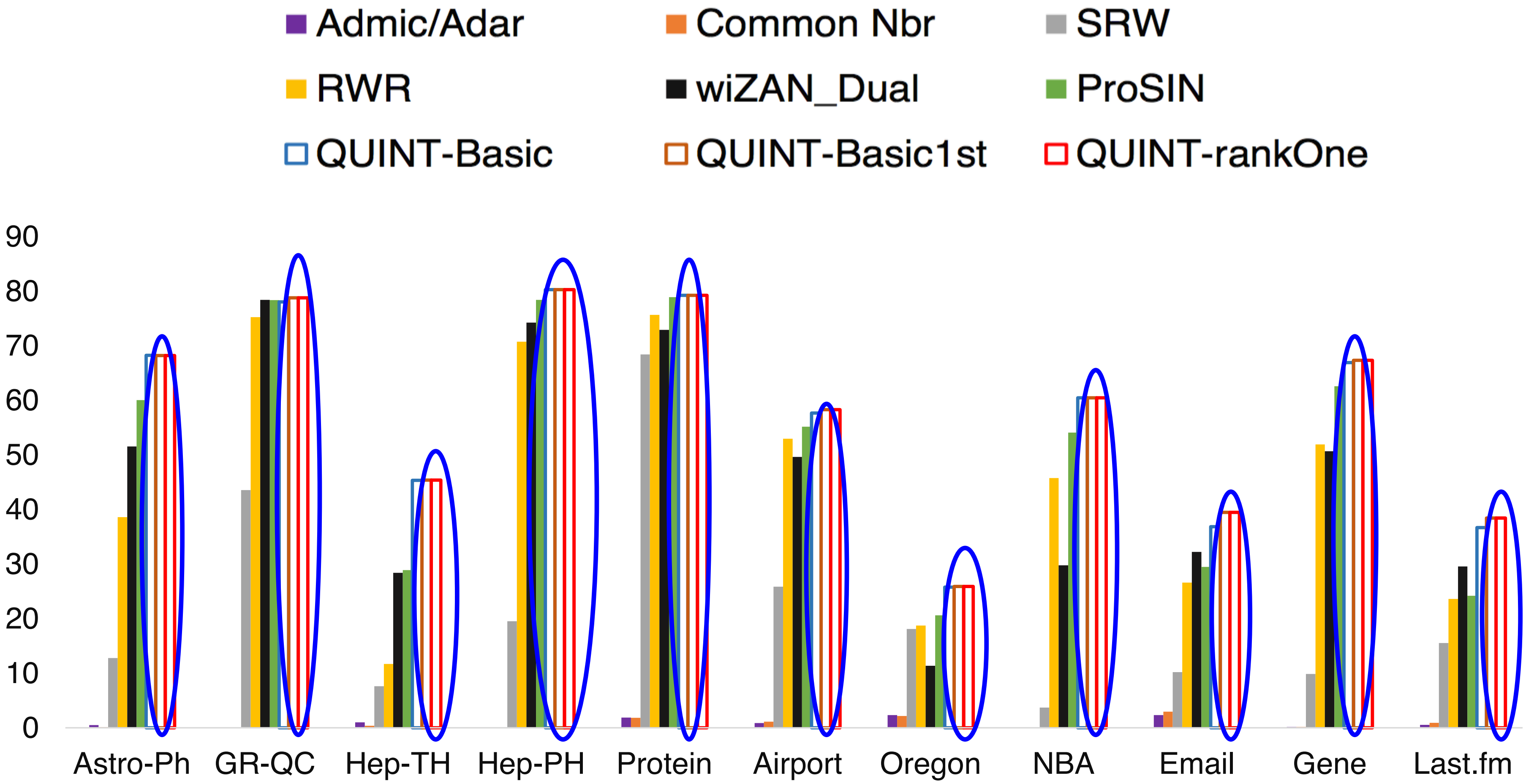
Effectiveness: MAP (Higher is better)

MAP: Mean Average Precision

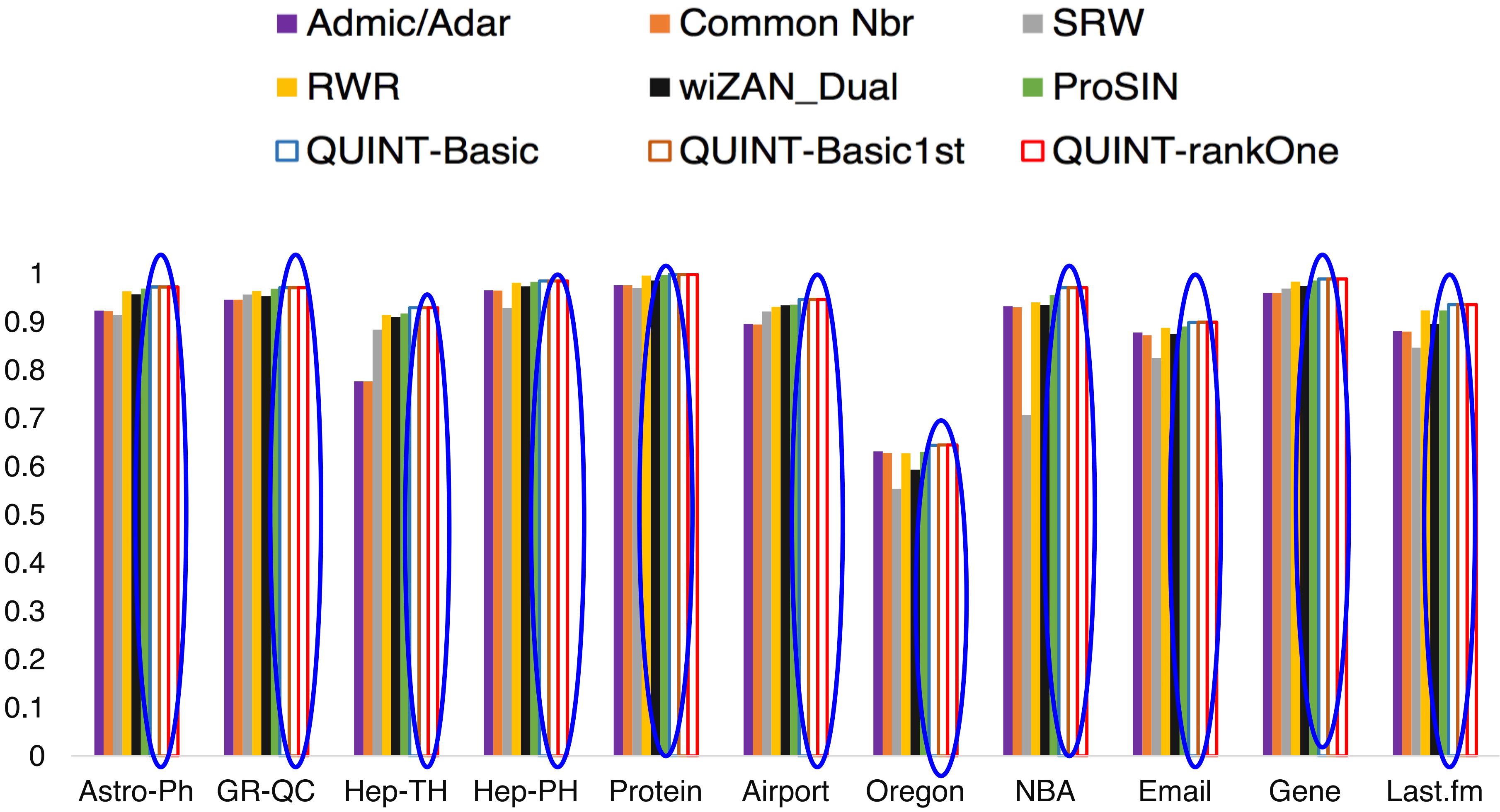


Effectiveness: HLU (Higher is better)

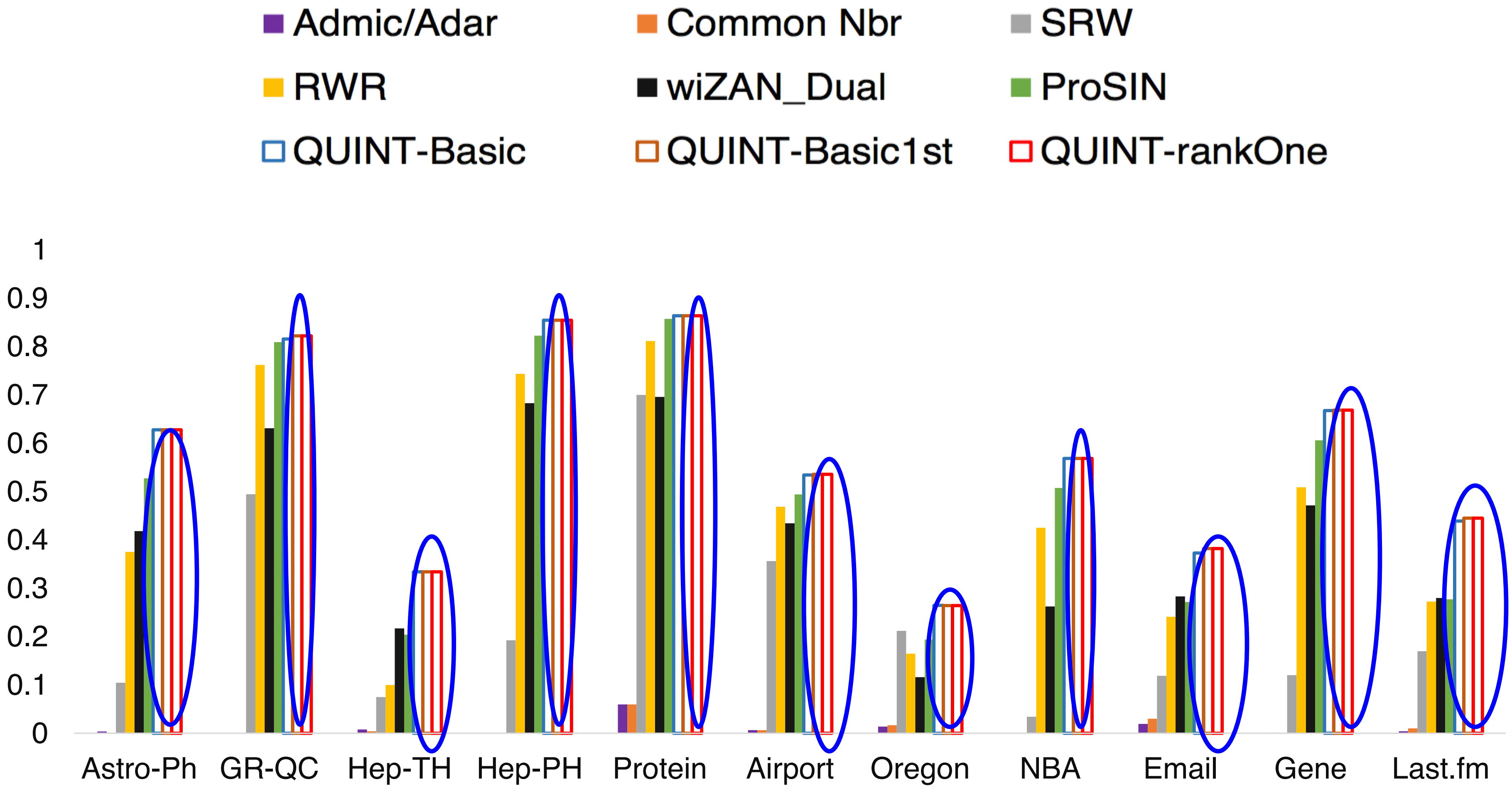
HLU: Half-life Utility



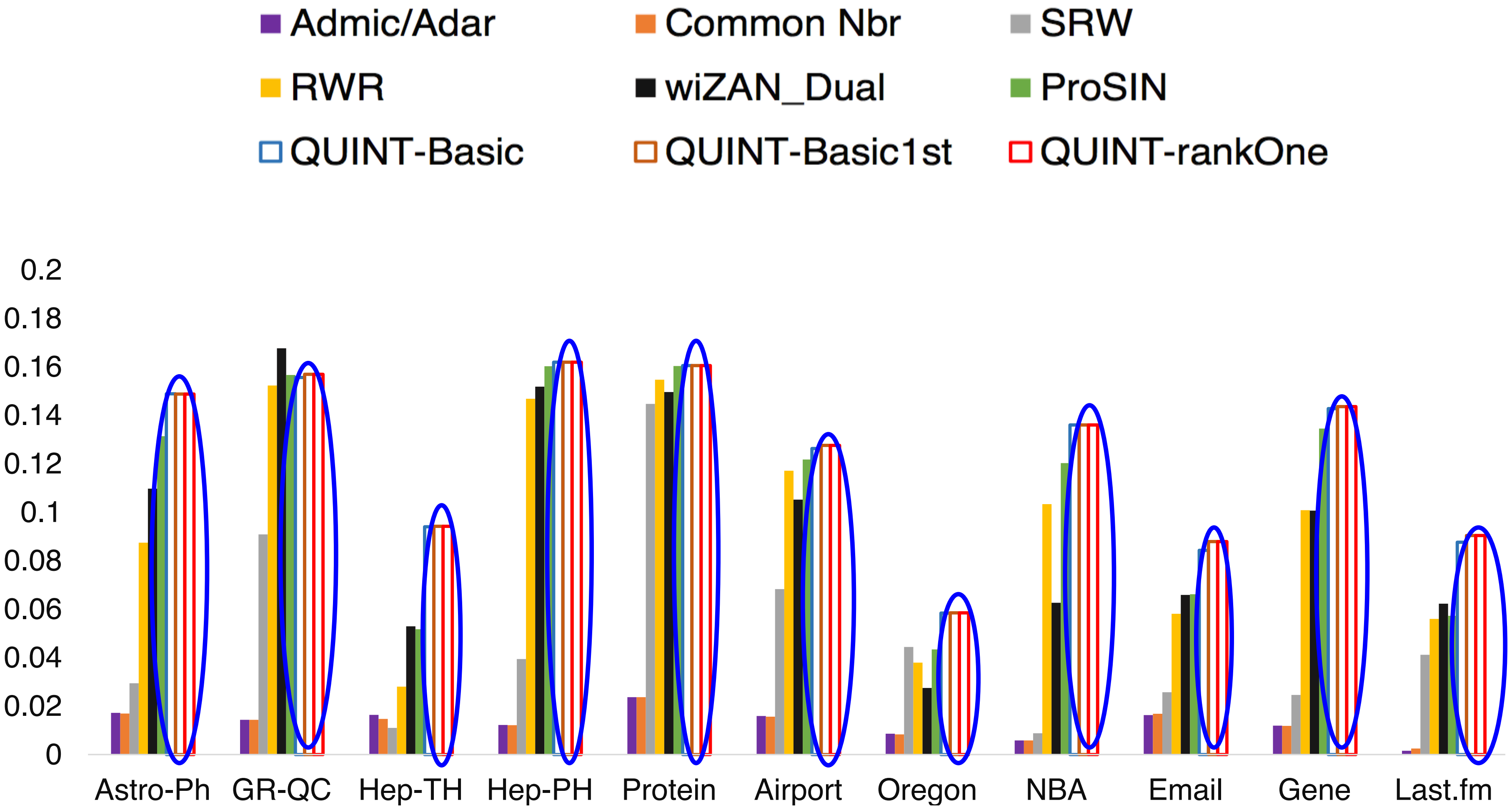
Effectiveness: AUC (Higher is better)



Effectiveness: Precision@20 (Higher is better)

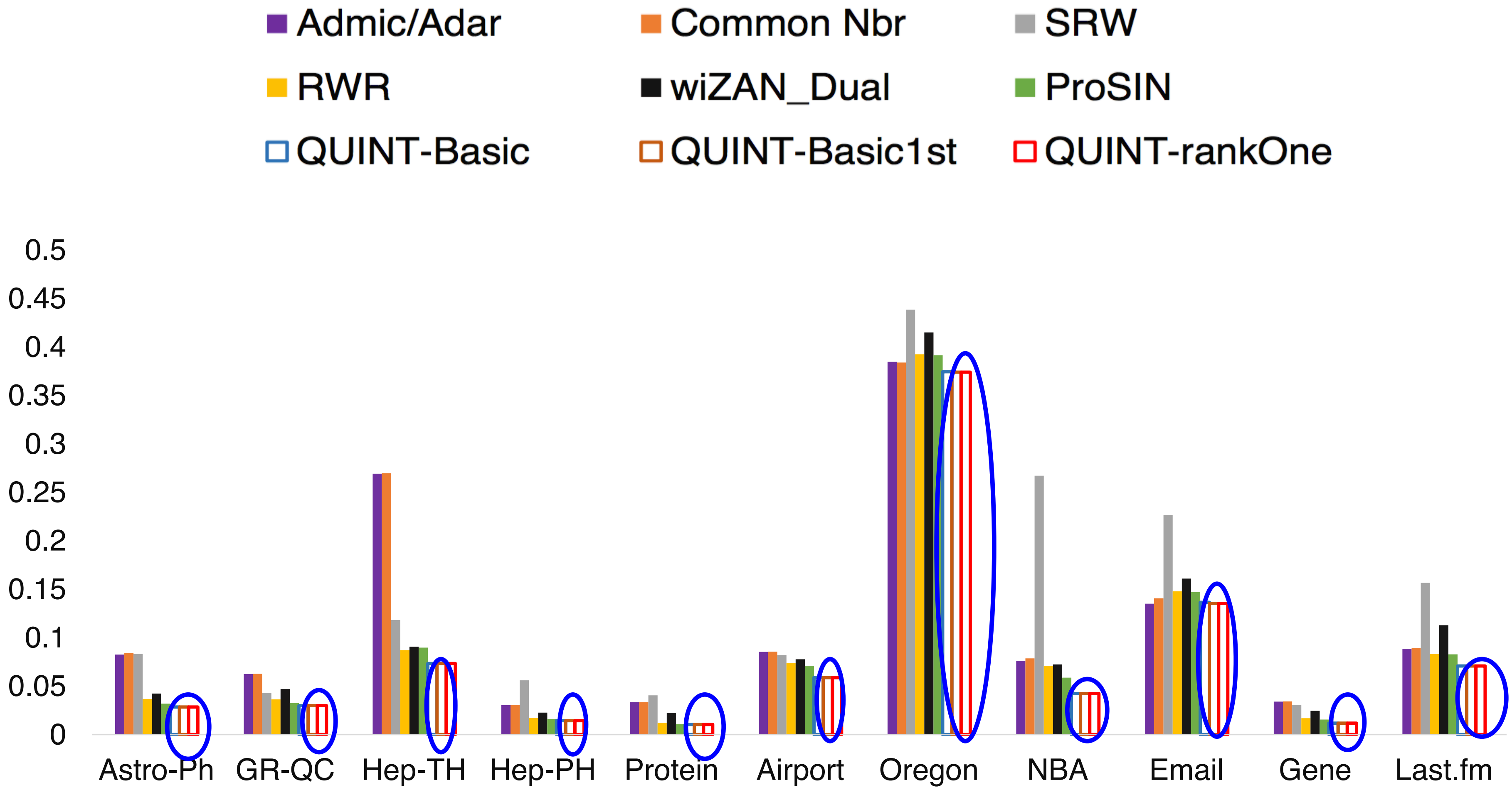


Effectiveness: Recall@5 (Higher is better)

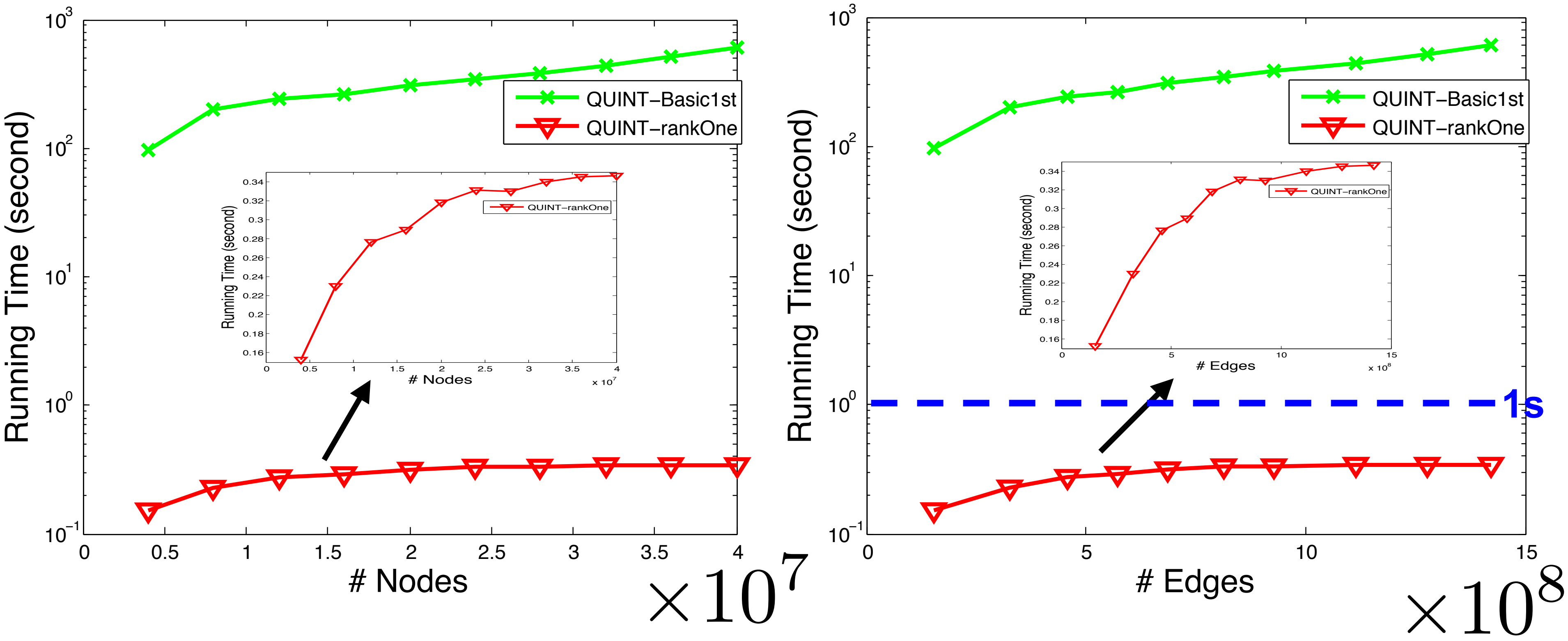


Effectiveness: MPR (Lower is better)

MPR: Mean Percentile Ranking



Efficiency -- Twitter



QUINT-rankOne scales sub-linearly

Roadmap

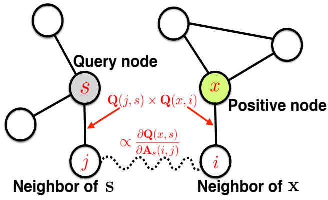
- Motivations
- Proposed Solutions: **QUINT**
- Empirical Evaluations
- Conclusions

Conclusion: QUINT

- **Goals**: Learn Optimal network (for Node Proximity)

	Q1	Q2	Q3
Existing	Optimal weights	One-fit-all	offline
QUINT	**Optimal topology**	**One-fit-one**	**online**

- **Algorithms**: VERY efficient way to compute $\frac{\partial \mathbf{Q}(x,s)}{\partial \mathbf{A}_s(i,j)}$
  - Rank-1 approx + Taylor approx + local search



- **Results:**
  - consistently better on 10+ networks & 6 metrics
  - sublinear scalability, near real-time response on billion-scale networks