GSP 2026
Madrid, Spain

Subspace Projection Methods for Fast Spectral Embeddings of Evolving Graphs

Mohammad Eini [†], Dr. Abdullah Karaaslanli [†], Dr. V. Kalantzis*, Prof. P. Traganitis[†]

* IBM Research, Thomas J. Watson Research Center

† Dept. of ECE, Michigan State University

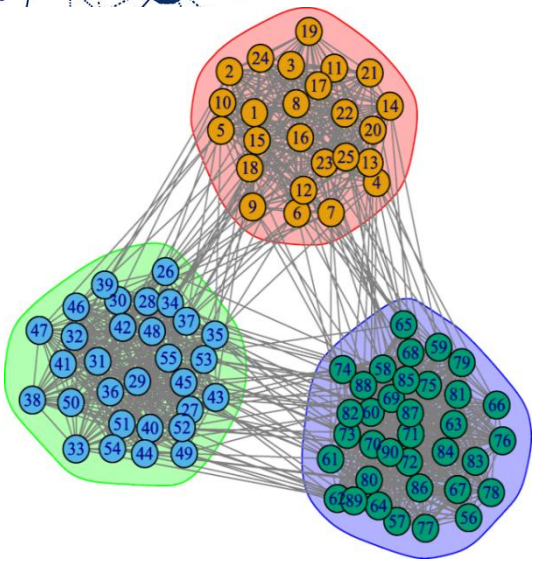
Acknowledgements: NSF Grant 2312546

arXiv pre-print: [2603.19439] - https://arxiv.org/abs/2603.19439

IBM

MICHIGAN STATE UNIVERSITY

Motivation



Graphs are ubiquitous

- ❖ Model social/economic/communication networks
- ❖ Describe complex/dynamical systems
- ❖ Capture correlation and dependencies between data

❑ Plethora of methods for graph analysis and learning

- ❖ Graph SP / Graph Neural Networks / Laplacian regularization

❑ **Graphs may vary over time/update dynamically**

- ❖ Evolving social networks
- ❖ Time-varying Communication networks
- ❖ Temporal structure of brain connectivity networks

Q: Can we efficiently update graph operators/descriptors as the graph changes?

Focus: Adjacency/Laplacian eigenvector updates

Eigenembeddings of graphs

- Consider an *N*-node graph $\mathcal{G} := (\mathcal{V}, \mathcal{E}) \quad N = |\mathcal{V}|$

  $\mathcal{V}$: vertex/node set $\qquad$ $\mathcal{E}$: edge set

- Key graph operator – $N \times N$ adjacency matrix $\mathbf{A}$

  $$[\mathbf{A}]_{ij} = \begin{cases} 1 & (i,j) \in \mathcal{E} \\ 0 & \text{o.w.} \end{cases}$$

  - Undirected graph – symmetric ***A***

- *K* leading eigenvalues $\{\lambda_i\}_{i=1}^K$ and corresponding eigenvectors $\{\mathbf{x}_i\}_{i=1}^K$ of $\mathbf{A}$

  $$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i \qquad |\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_K|$$

  $$\mathbf{X}_K := [\mathbf{x}_1, \ldots, \mathbf{x}_K]$$
  $$\mathbf{\Lambda}_K := \text{diag}(\lambda_1, \ldots, \lambda_K)$$

- **Applications:**
  - Node centrality
  - PageRank
  - Subgraph centrality
  - Katz Centrality
  - Spectral Clustering

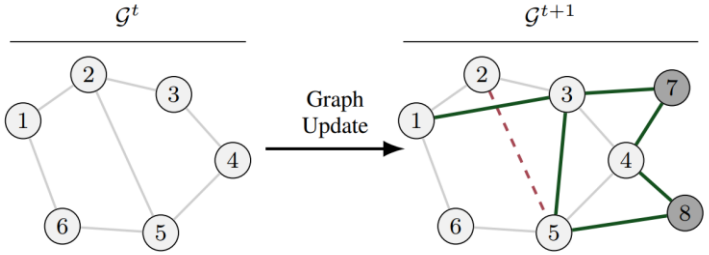
  $$\text{evecs of } \mathbf{L} = \text{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A}$$

Q: What if the graph changes over time? Recomputing EVD is expensive!

U. Kang, S. Papadimitriou, J. Sun, and H. Tong, "Centralities in large networks: Algorithms and observations," in Proceedings of the 2011 SIAM international conference on data mining, 2011, pp. 119–130.
U. Von Luxburg, "A tutorial on spectral clustering," Statistics and Computing, vol. 17, no. 4, pp. 395–416, 2007.

Time-varying graphs

- Consider a sequence of graphs $\{\mathcal{G}^{(t)}\}_{t=1}^{T}$
  - Graph at time *t:* $\mathcal{G}^{(t)} := (\mathcal{V}^{(t)}, \mathcal{E}^{(t)})$, $N^{(t)} = |\mathcal{V}^{(t)}| = |\mathcal{V}^{(t-1)}| + S^{(t)}$ with $\mathbf{A}^{(t)}$
  - Leading eigenpairs of $\mathbf{A}^{(t)} \rightarrow \{(\lambda_i^{(t)}, \mathbf{x}_i^{(t)})\}_{i=1}^{K}$



Topological changes

$$\mathbf{A}^{(t+1)} = \overline{\mathbf{A}}^{(t)} + \mathbf{\Delta}^{(t+1)} = \begin{bmatrix} \mathbf{A}^{(t)} & \mathbf{0}_{N^{(t)},S^{(t+1)}} \\ \mathbf{0}_{S^{(t+1)},N^{(t)}} & \mathbf{0}_{S^{(t+1)},S^{(t+1)}} \end{bmatrix} + \begin{bmatrix} \mathbf{K}^{(t+1)} & \mathbf{G}^{(t+1)} \\ \mathbf{G}^{(t+1)\top} & \mathbf{C}^{(t+1)} \end{bmatrix}$$

Graph expansion

- **Goal:** Update *K* leading eigenpairs of $\mathbf{A}^{(t)}$ to the *K* leading eigenpairs of $\mathbf{A}^{(t+1)}$
  - Single time-step update: $\mathbf{A} \rightarrow \widehat{\mathbf{A}} = \overline{\mathbf{A}} + \mathbf{\Delta} = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{N,S} \\ \mathbf{0}_{S,N} & \mathbf{0}_{S,S} \end{bmatrix} + \begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^{\top} & \mathbf{C} \end{bmatrix}$

$$\overline{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{0}_S \end{bmatrix} \qquad \{(\lambda_i, \overline{\mathbf{x}}_i)\}_{i=1}^{K} \rightarrow \{(\widehat{\lambda}_i, \widehat{\mathbf{x}}_i)\}_{i=1}^{K}$$

Prior art

- EVD at every time-step — Complexity per time-step: $O(|\mathcal{E}^{(t+1)}|K + (N^{(t)} + S^{(t+1)})K^2)$
- Perturbation-based methods [Chen & Tong'15, Mitz & Shkolnisky'22]
  - Based on first-order analysis

  Fast and scalable

  Information loss when graph expands
- SVD tracking for Spectral Clustering [Dhanjal et al'14] — Information loss when graph expands
- SVD tracking with restarts (TIMERS) [Zhang et al'17]

  High accuracy

  Computationally expensive

- **Contribution:** Computationally and memory efficient algorithm for tracking the adjacency of an expanding graph, that can tradeoff accuracy for efficiency

Perturbation-based updates

- Given eigenpairs $\{(\lambda_i, \mathbf{x}_i)\}_{i=1}^{K}$ of $\mathbf{A}$ a first-order sensitivity analysis yields

$$\widetilde{\lambda}_j = \lambda_j + \overline{\mathbf{x}}_j^\top \mathbf{\Delta} \overline{\mathbf{x}}_j + O(\|\mathbf{\Delta}\|^2)$$

$$\widetilde{\mathbf{x}}_j = \overline{\mathbf{x}}_j + \sum_{i=1, i\neq j}^{i=N} \frac{\overline{\mathbf{x}}_i^\top \mathbf{\Delta} \overline{\mathbf{x}}_j}{\lambda_j - \lambda_i} \overline{\mathbf{x}}_i + O(\|\mathbf{\Delta}\|^2)$$

$$\widetilde{\mathbf{x}}_j = \overline{\mathbf{X}}_K \mathbf{a}_j$$

Tracking Eigenpairs (TRIP) – Basic [Chen & Tong'15]

- Computing $\widetilde{\lambda}_j$'s first: $\widetilde{\mathbf{x}}_j = \overline{\mathbf{X}}_K \mathbf{b}_j$

TRIP [Chen & Tong'15]

$$(\mathrm{diag}(\widetilde{\lambda}_j - \lambda_1, \ldots, \widetilde{\lambda}_j - \lambda_K) - \overline{\mathbf{X}}_K^\top \mathbf{\Delta} \overline{\mathbf{X}}_K)\mathbf{b}_j = \overline{\mathbf{X}}_K^\top \mathbf{\Delta} \overline{\mathbf{x}}_j .$$

- Approximating *N-K* trailing eigenvectors, using some scalar *μ*

$$\widetilde{\mathbf{x}}_j = \overline{\mathbf{x}}_j + \sum_{i=1, i\neq j}^{K} \frac{\overline{\mathbf{x}}_i^\top \mathbf{\Delta} \overline{\mathbf{x}}_j}{\lambda_j - \lambda_i} \overline{\mathbf{x}}_i + \sum_{i=K+1}^{N} \frac{\overline{\mathbf{x}}_i^\top \mathbf{\Delta} \overline{\mathbf{x}}_j}{\lambda_j - \mu} \overline{\mathbf{x}}_i$$

$$\widetilde{\mathbf{x}}_j = \left[\overline{\mathbf{X}}_K, (\mathbf{I} - \overline{\mathbf{X}}_K \overline{\mathbf{X}}_K^\top)\mathbf{\Delta} \overline{\mathbf{x}}_j\right] \mathbf{c}_j$$

$$\widetilde{\mathbf{x}}_j = \left[\overline{\mathbf{X}}_K, (\mathbf{I} - \overline{\mathbf{X}}_K \overline{\mathbf{X}}_K^\top) \begin{bmatrix} \mathbf{K} \\ \mathbf{G}^\top \end{bmatrix} \mathbf{x}_j \right] \mathbf{c}_j$$

Residual modes (RM) [Mitz & Shkolnisky'22]

Perturbation-based methods approximate eigenvalues via $\widetilde{\mathbf{x}}_j = \mathbf{Z}\mathbf{v}_j$

- ✓ Fast and closed-form solution
- ✓ Require only $\mathbf{X}_K, \mathbf{\Delta}$ [low memory]
- x Coefficients $\mathbf{v}_j$ may be suboptimal
- x Do not handle graph expansion

T. Kato, Perturbation theory for linear operators, vol. 132. Springer Science & Business Media, 2013.

A Rayleigh-Ritz (RR) approximation scheme

- Given orthonormal basis matrix $\mathbf{Z} \in \mathbb{R}^{N+S \times D}$ coefficients $\mathbf{v}_j$ for $\widetilde{\mathbf{x}}_j$ can be estimated via RR

  S1. Compute the *K* leading eigenpairs $\{(\theta_j, \mathbf{v}_j)\}_{j=1}^{K}$ of $\mathbf{Z}^\top \widehat{\mathbf{A}} \mathbf{Z} = \mathbf{Z}^\top \overline{\mathbf{A}} \mathbf{Z} + \mathbf{Z}^\top \boldsymbol{\Delta} \mathbf{Z}$

  S2. Form approximate eigenpairs $\{(\theta_j, \mathbf{Z}\mathbf{v}_j)\}_{j=1}^{K}$ of $\widehat{\mathbf{A}}$: $\widehat{\mathbf{x}}_j = \mathbf{Z}\mathbf{v}_j$

  - Minimize residual $\|\widehat{\mathbf{A}}\mathbf{Z}\mathbf{v}_j - \theta_j \mathbf{Z}\mathbf{v}_j\|$
  - Approximation exact when: $\widehat{\mathbf{x}}_j \in \mathrm{Ran}(\mathbf{Z})$

- Basis matrix **Z** affects
  - Approximation accuracy
  - Computational complexity

- Reduced memory footprint using existing eigendecomposition

$$\mathbf{Z}^\top \widehat{\mathbf{A}} \mathbf{Z} \approx \mathbf{Z}^\top \overline{\mathbf{X}}_K \overline{\boldsymbol{\Lambda}}_K \overline{\mathbf{X}}_K^\top \mathbf{Z} + \mathbf{Z}^\top \boldsymbol{\Delta} \mathbf{Z}$$

G. H. Golub and C. F. Van Loan, Matrix computations, JHU press, 2013.
V. Kalantzis, G. Kollias, S. Ubaru, A. N. Nikolakopoulos, L. Horesh, and K. Clarkson, “Projection techniques to update the truncated SVD of evolving matrices with applications,” in International Conference on Machine Learning, 2021, pp. 5236–5246.

Constructing an expressive basis matrix

- Recall update structure $\widehat{\mathbf{A}} = \overline{\mathbf{A}} + \boldsymbol{\Delta} = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{N,S} \\ \mathbf{0}_{S,N} & \mathbf{0}_{S,S} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{K} \\ \mathbf{G}^\top \end{bmatrix}}_{\boldsymbol{\Delta}_1} \underbrace{\begin{bmatrix} \mathbf{G} \\ \mathbf{C} \end{bmatrix}}_{\boldsymbol{\Delta}_2}$

- Residual mode basis matrix $\mathbf{Z}_{\mathrm{RM}} = \left[\overline{\mathbf{X}}_K, (\mathbf{I} - \overline{\mathbf{X}}_K \overline{\mathbf{X}}_K^\top)\boldsymbol{\Delta}_1 \mathbf{X}_K\right]$
  - ✗ Missing information from $\boldsymbol{\Delta}_2$

Idea: Explicitly include $\boldsymbol{\Delta}_2$ in basis $\mathbf{Z}$

$$\mathbf{Z} = \mathrm{Ran}\left(\left[\overline{\mathbf{X}}_K, (\mathbf{I} - \overline{\mathbf{X}}_K \overline{\mathbf{X}}_K^\top)\left[\boldsymbol{\Delta}_1 \mathbf{X}_K, \boldsymbol{\Delta}_2\right]\right]\right)$$

  - ✓ Encodes all graph changes
  - ✗ Increases dimensionality of basis by *S*

- Reducing complexity by Randomized SVD (RSVD)
  - Rank-*L* truncated SVD of $\mathbf{Y} = (\mathbf{I} - \overline{\mathbf{X}}_K \overline{\mathbf{X}}_K^\top)\boldsymbol{\Delta}_2 \boldsymbol{\Omega}_\top$ yields approximate orthonormal basis of $(\mathbf{I} - \overline{\mathbf{X}}_K \overline{\mathbf{X}}_K)\boldsymbol{\Delta}_2$

$$\boldsymbol{\Omega} \in \mathbb{R}^{S \times L+P}$$
$$[\boldsymbol{\Omega}]_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

N. Halko, P.-G. Martinsson, and J. A. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions,” SIAM review, vol. 53, no.2, pp. 217–288, 2011.

Graph-Rayleigh Ritz Eigenspace Tracking (G-REST)

- Initial graph adjacency $\mathbf{A}^{(0)} \approx \mathbf{X}_K \mathbf{\Lambda}_K \mathbf{X}_K^\top$

For *t = 0,…, T-1*

- Receive graph update $\mathbf{\Delta}^{(t+1)}$
- Construct orthonormal basis $\mathbf{Z}^{(t+1)}$
- Compute *K* leading eigenpairs $\{(\theta_j, \mathbf{v}_j)\}_{j=1}^{K}$ of

$$\mathbf{Z}^{(t+1)^\top} \mathbf{A}^{(t+1)} \mathbf{Z}^{(t+1)} \approx \mathbf{Z}^{(t+1)^\top} \overline{\mathbf{X}}_K^{(t)} \mathbf{\Lambda}_K^{(t)} \overline{\mathbf{X}}_K^{(t)^\top} \mathbf{Z}^{(t+1)} + \mathbf{Z}^{(t+1)^\top} \mathbf{\Delta}^{(t+1)} \mathbf{Z}^{(t+1)}.$$

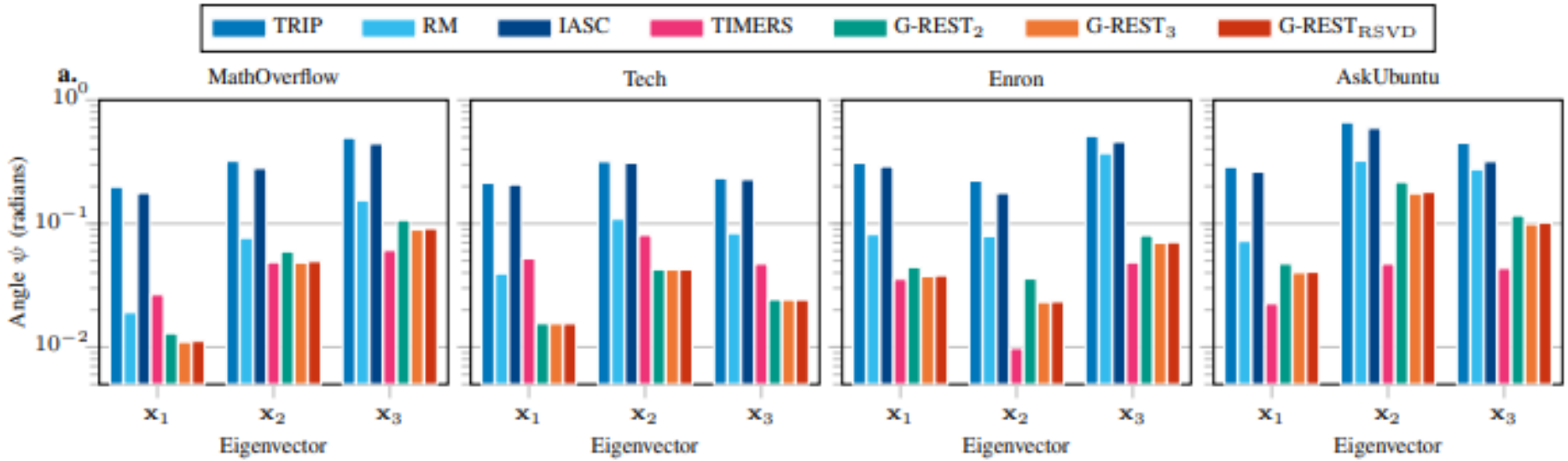
- Set $\mathbf{X}_K^{(t+1)} = \mathbf{Z}^{(t+1)} [\mathbf{v}_1, \ldots, \mathbf{v}_K], \quad \mathbf{\Lambda}_K^{(t+1)} = \mathrm{diag}(\theta_1, \ldots, \theta_K)$

- Updating Laplacian $\mathbf{L} = \mathrm{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A}$ eigenembeddings:
  - Same algorithm can be applied to $\mathbf{T}^{(t)} = \alpha^{(t)} \mathbf{I} - \mathbf{L}^{(t)}$

Numerical tests: Eigenvector approximation

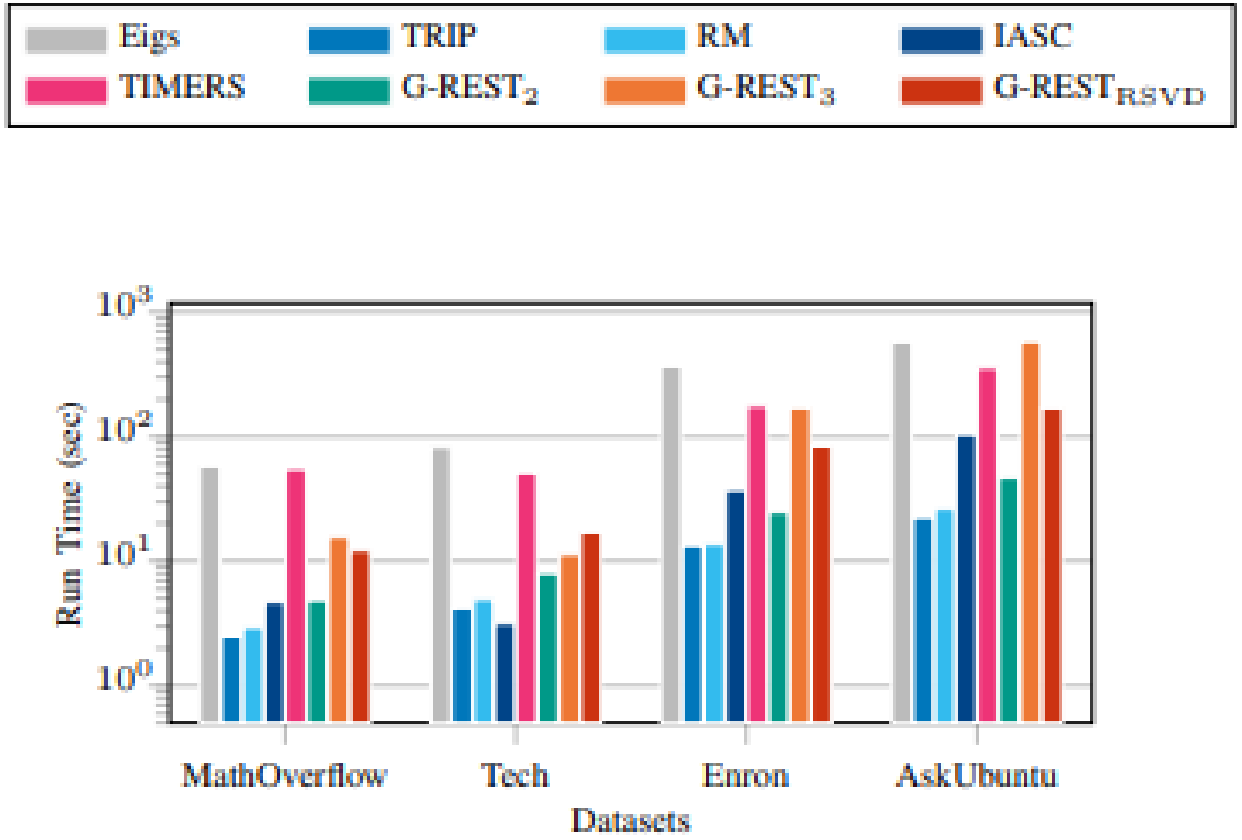
- Dynamic graph datasets [time-stamped edges] from SNAP
  - At each *t* new vertices and edges added. *T* total updates
  - Figures of merit: eigenvector angle w.r.t. **eigs,** after *T* updates

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$
MathOverflow	24,818	187,986
Tech	34,761	107,720
Enron	87,273	297,456
AskUbuntu	159,316	455,691



J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection." http://snap.stanford.edu/data, June 2014.

Numerical tests: Runtime

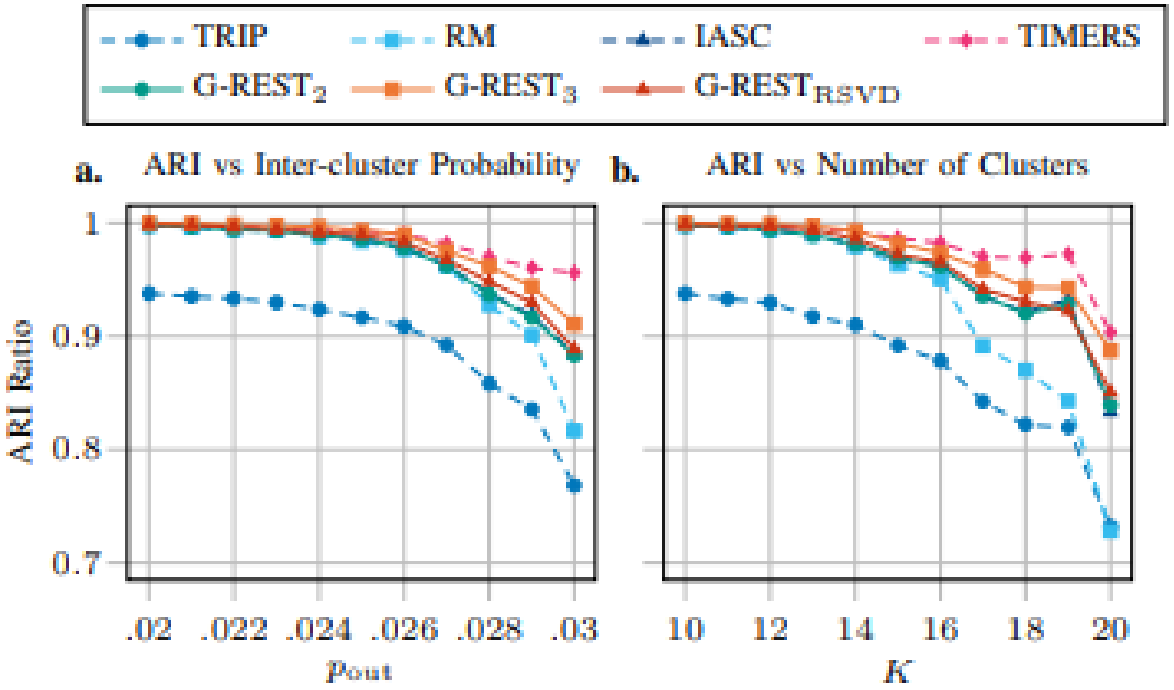


Numerical tests: Spectral clustering

- Synthetic Stochastic Block Model graph, *N=10,000, K* communities, $N^{(0)} = 9,500, T = 10$

Probability of intra-cluster edge: $p_{\text{in}} = 0.05$ Probability of inter-cluster edge: p_{out}

- Figure of merit: Adjusted Rand Index (ARI) ratio



L. Hubert and P. Arabie, “Comparing partitions,” Journal of classification, vol. 2, no. 1, pp. 193–218, 1985

Conclusions and research directions

- **Take home**: Adjacency eigenvector embeddings can be efficiently updated via a Rayleigh-Ritz approximation procedure
  - Judiciously designed basis matrix yields accurate updates
  - Randomized SVD can speed up proposed method – accuracy/efficiency tradeoff
- Future research directions:
  - Node removal
  - Performance analysis
  - Applications in GSP and GNNs

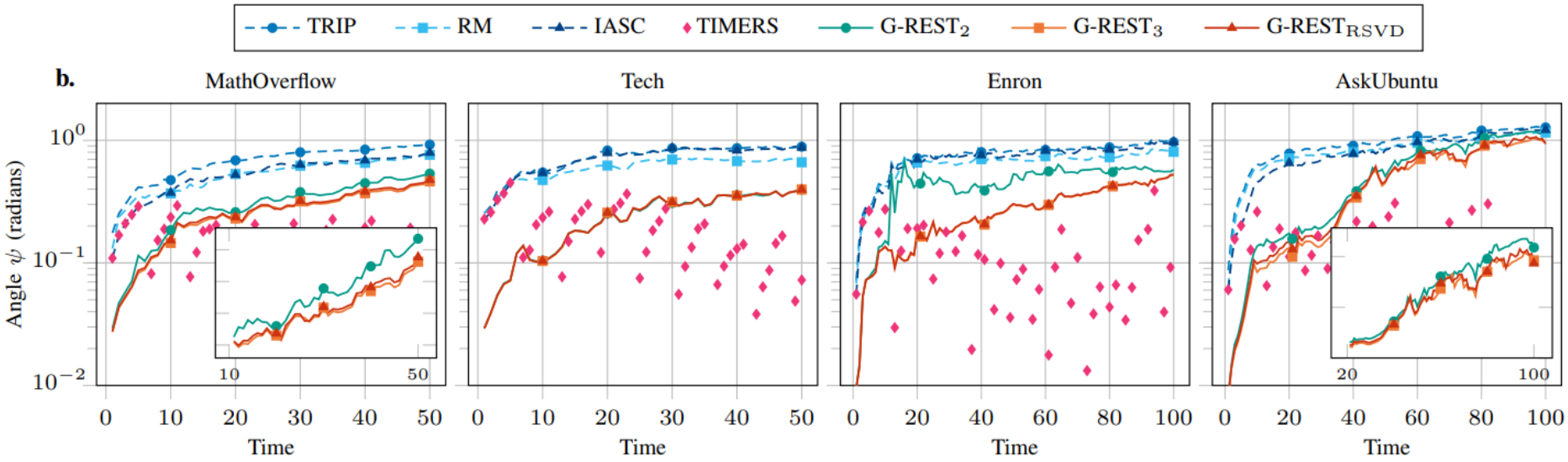
arXiv pre-print: [2603.19439] - https://arxiv.org/abs/2603.19439

Thank you!

Numerical tests: Eigenvector approximation

- Dynamic graph datasets [time-stamped edges] from SNAP
  - At each *t* new vertices and edges added. *T* total updates
  - Figures of merit: eigenvector angle w.r.t. **eigs,** after *T* updates

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$
MathOverflow	24,818	187,986
Tech	34,761	107,720
Enron	87,273	297,456
AskUbuntu	159,316	455,691



J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection." http://snap.stanford.edu/data, June 2014.