Column subset selection and Markov chain compression

Probability Seminar

Brown

March 31, 2026

Michael Lindsey

UC Berkeley

Based on joint work with **Mark Fornace** (LBNL)

Part I: Background on CSSP

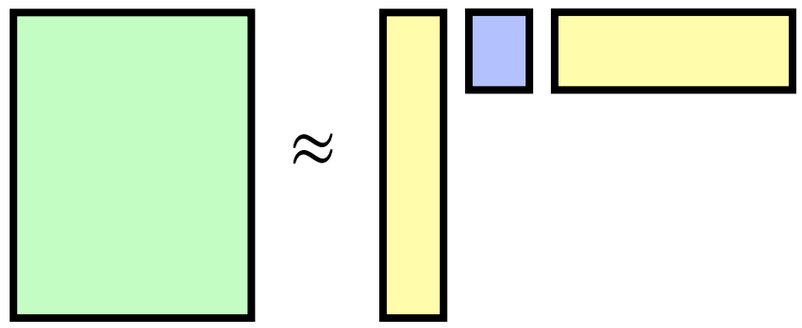
Column selection

- **Rough idea:** want to pick columns of a matrix that span its (effective) column space as parsimoniously as possible
  - Structured low-rank decomposition
- **CX decomposition** (a.k.a. *interpolative decomposition*)**:** given $A \in \mathbb{R}^{m \times n}$ want to choose columns $C = A_{:,\mathcal{J}}$ such that we can recover $A \approx CX$
  - Given column selection C, can choose X to minimize $\|A - CX\|_{\mathrm{F}}^2$
  - Yields $X = C^{\dagger}A = (C^{\mathsf{T}}C)^{-1}C^{\mathsf{T}}A$
- **CUR decomposition:** $A \approx CUR$ where R denote selected rows
  - Choose $U = C^{\dagger}AR^{\dagger}$
  - Can choose columns and rows independently without sacrificing much

Nyström approximation

- **Nyström approximation:** for a positive definite ("kernel") matrix, approximately factorize

$$K = K_{:,\mathcal{J}} [K_{\mathcal{J},\mathcal{J}}]^{-1} K_{\mathcal{J},:}$$



- CX factorization and Nyström approximation are closely related!
- Consider $K = A^\top A$ and choose $C = A_{:,\mathcal{J}}$. Then:

$$\min_X \|A - CX\|_{\mathrm{F}}^2 = \mathrm{Tr}\left[\underbrace{K - K_{:,\mathcal{J}} [K_{\mathcal{J},\mathcal{J}}]^{-1} K_{\mathcal{J},:}}_{=: \tilde{K}(\mathcal{J}) \geq 0} \right]$$

- **Frobenius norm error of CX ↔ nuclear norm error of Nyström**
- We will mostly focus the discussion on Nyström approximation, without loss of generality
  - Can reduce to interpolative decomposition via a **feature map**, e.g., random Fourier features

Example applications

- **Many applications!**
  - Interpretable / structured low-rank factorizations [Mahoney and Drineas (2008)]
    - Identify descriptive data points / features
  - Data reduction
  - Efficient kernel regression
  - Experimental design
  - Spectral clustering [Damle, Minden, Ying (2018)]
  - Model order reduction (DEIM) [Chaturantabut and Sorensen (2009)]
  - Tensor cross interpolation (tensor trains from queries) [Oseledets and Tyrtyshnikov]
  - Computational quantum chemistry (SCDM and ISDF) [Damle, Lin, Lu, Ying]
  - Low-rank compression of neural networks [Chee et al (2022)]
  - New application: **Markov chain compression**

Existing approaches

- Column subset selection problem (CSSP) is NP-complete [Shitov 2021]
- May or may not care a lot about speed of the selection algorithm… provided that it's tractable (you can rarely solve exactly)
- **Many approaches!**
  - Uniform sampling
  - Leverage score sampling [Clarkson, Drineas, Kannan, Mahoney, Woodruff, …]
  - QR with column pivoting (QRCP) / greedy pivoted Cholesky
  - DEIM [Chaturantabut and Sorensen (2009)]
  - Strong RRQR [Gu and Eisenstat]
  - Randomly pivoted QR [Deshpande and Vempala] / randomly pivoted Cholesky [Chen, Epperly, Tropp, Webber (2023)]
  - Determinantal point process sampling (DPP)
- *Wide range of theoretical and practical pros and cons*

Some adaptive approaches

- ***s*-DPP sampling:**
  - Sample size-s subset $\mathcal{J}$ from (unnormalized) density $\det(K_{\mathcal{J},\mathcal{J}})$
  - For reviews, see [Dereziński and Mahoney; Kulesza]
- **Greedy pivoted Cholesky** (<u>diagonal maximization</u>):
  - Augment $\mathcal{J}$ with ℓ maximizing $\left[\tilde{K}(\mathcal{J})\right]_{\ell,\ell}$
- **Randomly pivoted Cholesky** (<u>diagonal sampling</u>):
  - Augment $\mathcal{J}$ with ℓ sampled from weights $\left[\tilde{K}(\mathcal{J})\right]_{\ell,\ell}$

- **Nuclear maximization** (<u>this work</u>):
  - Augment $\mathcal{J}$ with index ℓ minimizing the objective $\mathrm{Tr}\left[K - \tilde{K}(\mathcal{J} \cup \{\ell\})\right]$
  - See [Farahat et al (2011)] , [Ordozgoit et al (2018)], and [Altschuler et al (2016)]
  - Various algorithmic improvements and extensions, stronger theory

Part II: Algorithms for nuclear selection

Nuclear maximization: basic algorithm

- Some algebra reveals that given $\mathcal{I}$ and $\tilde{K} = \tilde{K}(\mathcal{I})$, we augment with the index ℓ that maximizes

$$\mathscr{L}_{\tilde{K}}(\ell) := \frac{[\tilde{K}^2]_{\ell,\ell}}{\tilde{K}_{\ell,\ell}} = \frac{\|\tilde{K}_{:,\ell}\|^2}{\tilde{K}_{\ell,\ell}}$$

- We develop a new sparsity-exploiting algorithm for exact implementation (see the preprint for full pseudocode)

Nuclear maximization: basic algorithm

- Achieves the following scaling for k-selection in Nyström kernel approximation ($n \times n$) and CX / CUR ($m \times n$)

Algorithm setting	Deterministic scoring
Kernel approximation	$O(nk^2 + k\,\mathrm{nnz}(K))$
CUR decomposition	$O((m+n)k^2 + k\,\mathrm{nnz}(A) + \chi)$

χ is the cost of computing both $AA^\top$ and $A^\top A$

- For CX / CUR, we will do much better (avoid construction of $AA^\top$ and $A^\top A$)
- For Nyström, pivoted Cholesky approaches cost $O(nk^2)$, assuming cost $O(1)$ per entry
  - Avoid forming whole matrix K (only need to update diagonal of $\tilde{K}$)
- If fast matvecs by K are available (cf. EFGP [Greengard, Rachh, Barnett] and extension [Kielstra and Lindsey], KeOps [Charlier et al]), we can improve to $\approx O(nk^2)$
- Memory bottleneck of matvec can be avoided (cf. KeOps)

Nuclear maximization: matrix-free algorithm

- Recall the nuclear scores:
-

$$\mathscr{L}_{\tilde{K}}(\ell) := \frac{[\tilde{K}^2]_{\ell,\ell}}{\tilde{K}_{\ell,\ell}} = \frac{\|\tilde{K}_{:,\ell}\|^2}{\tilde{K}_{\ell,\ell}}$$

- $\tilde{K} = \tilde{K}(\mathscr{I})$ permits fast matvecs, we can estimate $\mathrm{diag}\big[\,\tilde{K}^2\,\big]$ using randomized technique
  - Interpretable as either a trace estimator or a subspace embedding
  - Use the identity

$$\mathrm{diag}\big[\,\tilde{K}^2\,\big] = \mathbb{E}_{z\sim\mathscr{N}}[(\tilde{K}z)\odot(\tilde{K}z)]$$

  - Require $O(\log n)$ samples for constant relative error control
- Similarly, if $\tilde{K} = \tilde{C}\tilde{C}^{\mathsf{T}}$ admits a symmetric factorization, we can use identity

$$\mathrm{diag}[\,\tilde{K}\,] = \mathbb{E}_{z\sim\mathscr{N}}[(\tilde{C}z)\odot(\tilde{C}z)]$$

Nuclear maximization: matrix-free algorithm

- Useful identity / interpretation of the remainder:

$$\tilde{K}(\mathcal{I}) = K - K_{:,\mathcal{I}}\,[K_{\mathcal{I},\mathcal{I}}]^{-1}\,K_{\mathcal{I},:} = P(\mathcal{I})\,K\,P(\mathcal{I})^{\mathsf{T}}$$

where $P(\mathcal{I})$ is the "oblique projector":

$$P(\mathcal{I}) = I - K_{:,\mathcal{I}}\,[K_{\mathcal{I},\mathcal{I}}]^{-1}\,I_{\mathcal{I},:}$$

which satisfies $P(\mathcal{I})^2 = P(\mathcal{I}),\ P(\mathcal{I})^{\mathsf{T}}\,K_{:,\mathcal{I}} = 0$

- Therefore a symmetric factorization for $K = CC^{\mathsf{T}}$ yields a symmetric factorization $\tilde{K} = \tilde{C}\tilde{C}^{\mathsf{T}}$ via

$$\tilde{C}(\mathcal{I}) = P(\mathcal{I})\,C$$

- Note that $K = A^{\mathsf{T}}A$ and $K = AA^{\mathsf{T}}$ are automatically equipped with symmetric factorization for CUR column/row selection, yielding complexity

$$O((m+n)k^2z + kz\,\mathrm{nnz}(A))$$

$z = O(\log n)$ denotes number of random samples in randomized diagonal estimation

Part III: Error analysis

Preliminaries

- Recall we want to choose subset minimizing the nuclear norm error:

$$\mathcal{E}_K(\mathcal{I}) \coloneqq \mathrm{Tr}[\tilde{K}(\mathcal{I})] = \mathrm{Tr}[K] - \mathrm{Tr}[K_{:,\mathcal{I}}(K_{\mathcal{I},\mathcal{I}})^{-1}K_{\mathcal{I},:}]$$

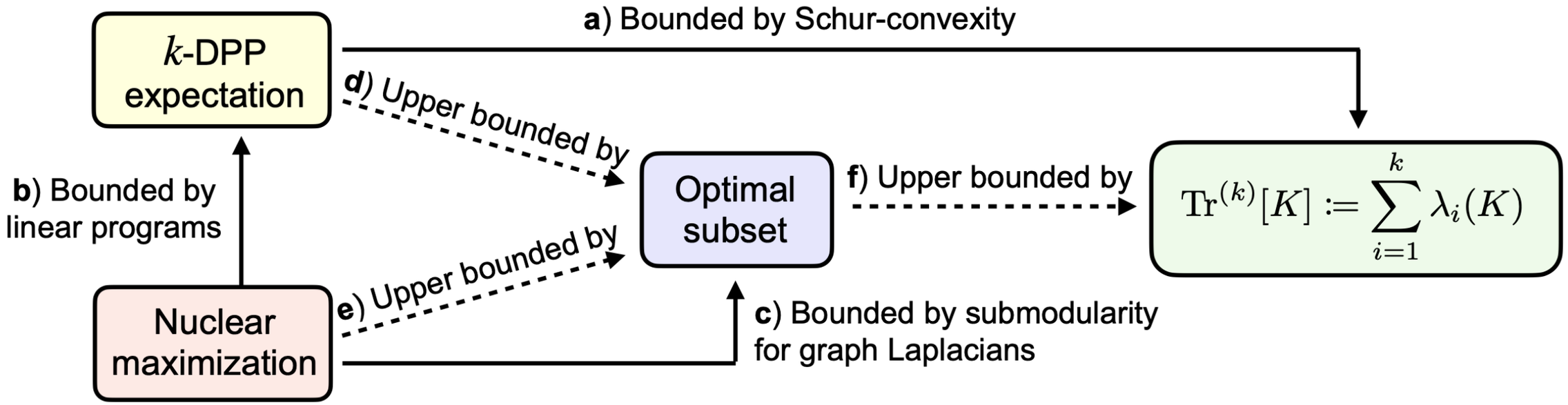
- Equivalently, maximize the following objective:

$$\mathcal{L}_K(\mathcal{I}) \coloneqq \mathrm{Tr}[K_{:,\mathcal{I}}(K_{\mathcal{I},\mathcal{I}})^{-1}K_{\mathcal{I},:}]$$

- Trivial upper bound offered by sum of top k eigenvalues:

$$\mathrm{Tr}^{(k)}\left[K\right] \coloneqq \sum_{i=1}^{k} \lambda_i(K)$$

Schematic of error bounds

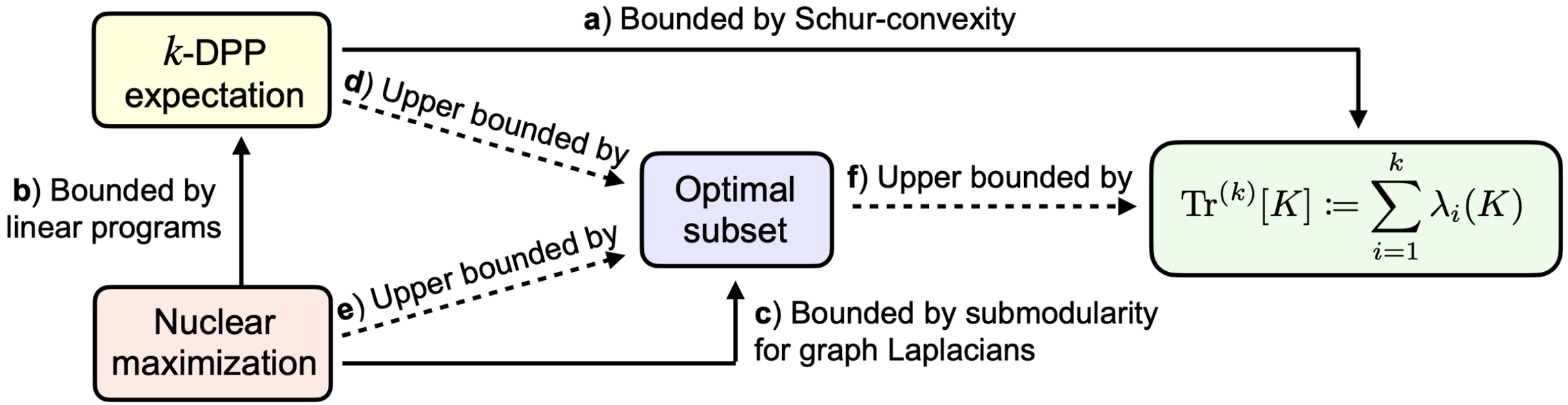


- Useful to recall the performance guarantee on DPP sampling [Guruswami and Sinop (2012)]

$$\mathrm{Tr}[K] - \mathcal{D}_s(K) \leq \left(1 + \frac{r}{s-r+1}\right)\left(\mathrm{Tr}[K] - \mathrm{Tr}^{(r)}\,[K]\,\right)$$

- Here $\mathscr{D}_s(K)$ is the expectation value of s-DPP sampling

Our main result



Discrepancy between nuclear maximization and DPP sampling

For $\mathcal{G}_k^{(\zeta)}$, a subset of $k \geq s$ columns greedily selected with approximation error ζ

$$1 - \frac{\mathcal{L}_K(\mathcal{G}_k^{(\zeta)})}{\mathcal{D}_s(K)} < e^{-k/((1+\zeta)s)}$$

ζ is the relative approximation error for the nuclear scores

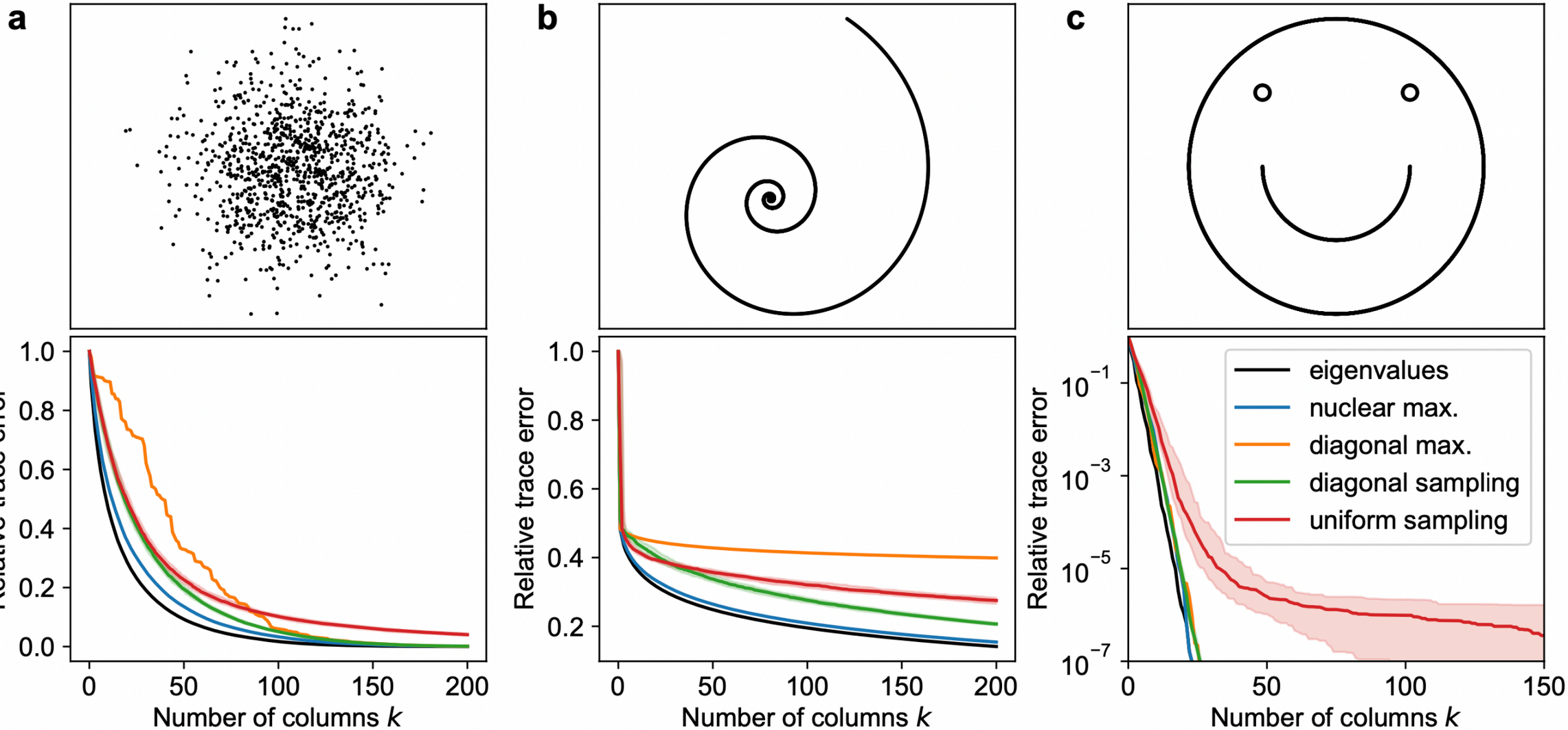
In randomized matrix-free approach, guaranteed to be $O(\epsilon)$ with $O(\epsilon^2 \log n)$ matvecs

Part IV: Numerical results

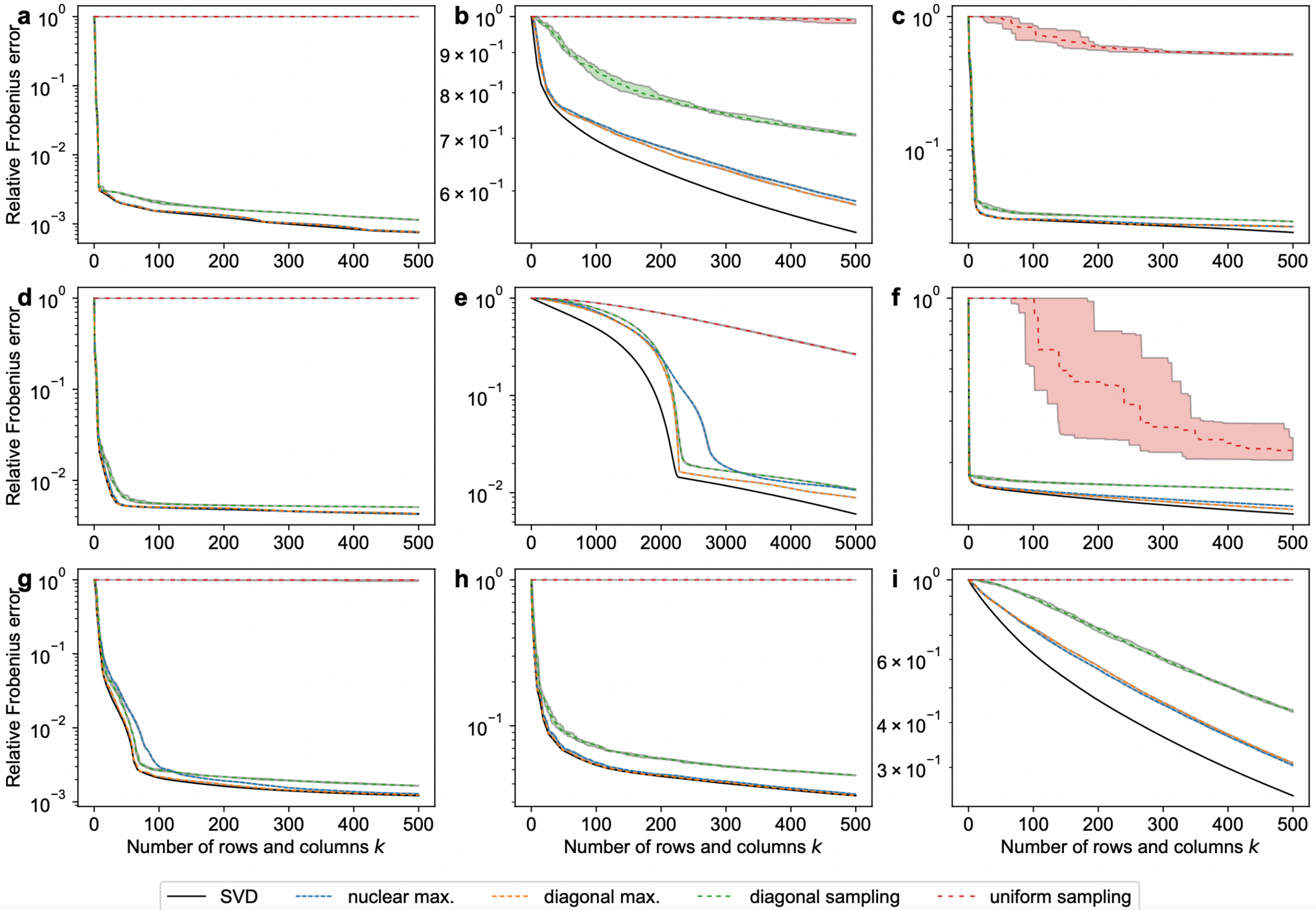
Summary of findings

- Careful comparison of nuclear maximization (*this work*), uniform selection, diagonal maximization (a.k.a. QRCP / greedy pivoted Cholesky), and diagonal sampling (a.k.a. randomly pivoted QR / Cholesky)
- Across all experiments, nuclear maximization performs as well or better than the other methods
  - Diagonal maximization tends to work well in many real-world settings, but can be badly broken
- We design counterexamples for which *only nuclear maximization* succeeds qualitatively (see preprint for details)
- Moreover, for counterexamples previously designed to break diagonal maximization, nuclear selection performs best even though it is also greedy

Some kernel approximation results

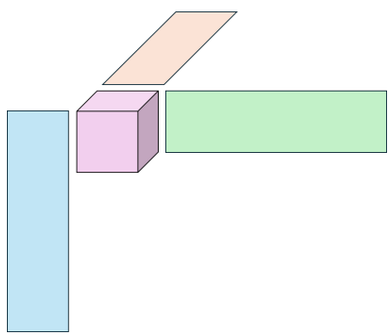


CUR decomposition results (SuiteSparse)

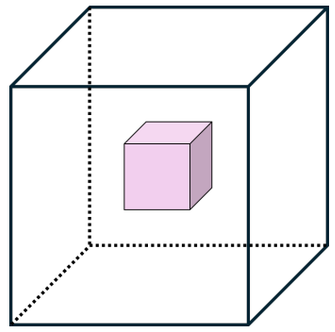


Structure preserving tensor approximation

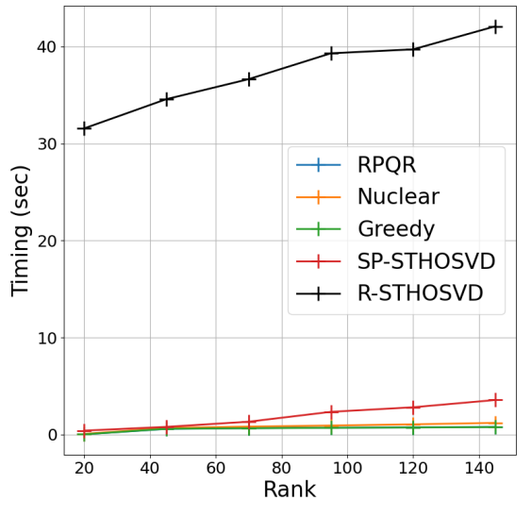
- Extend column / row selection tools to structured tensor approximation
  - Algorithmically exploit sparsity or CP / empirical moment structure
- Work with Mark Fornace and **Yifan Zhang** (UT Austin): arXiv:2503.18921



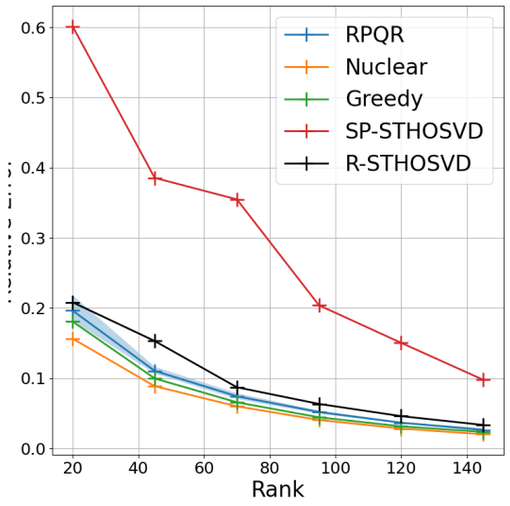
Tensor interpolative decomposition



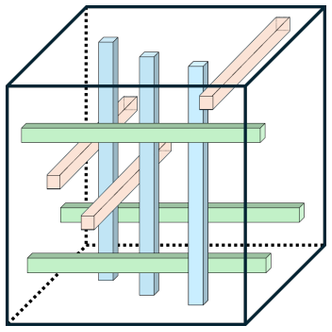
Core ID



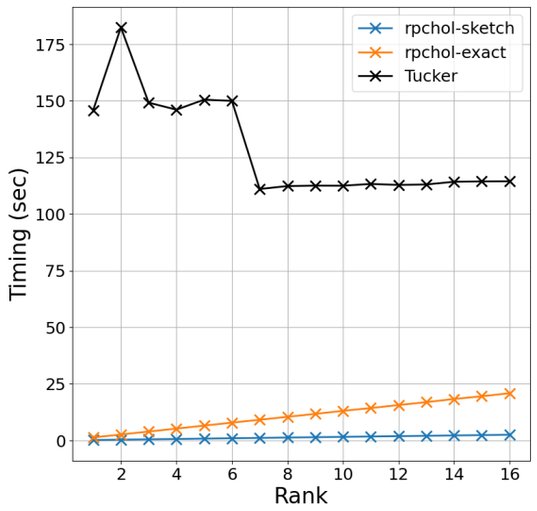
Speedups vs alternatives



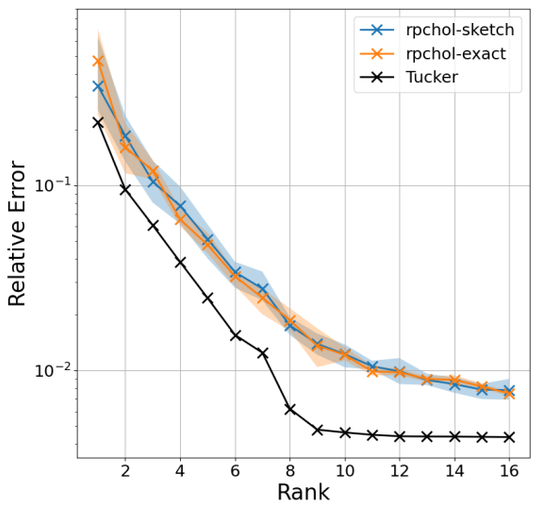
Error vs alternatives



Satellite ID



Speedups vs Tucker and via sketching

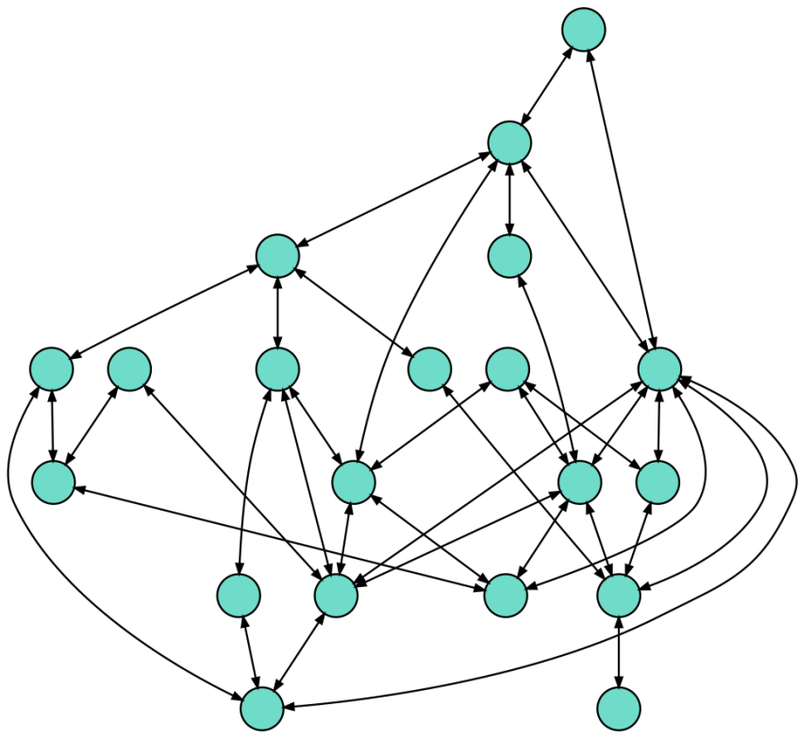


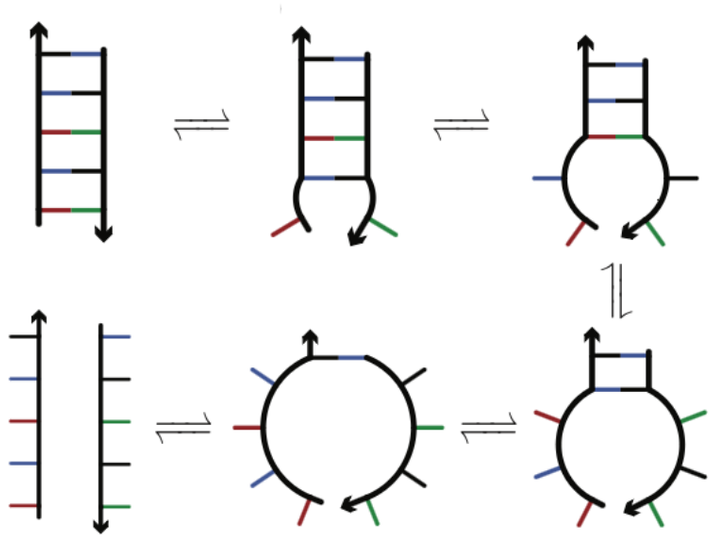
Preservation of accuracy when sketching

Part V: Markov chain compression

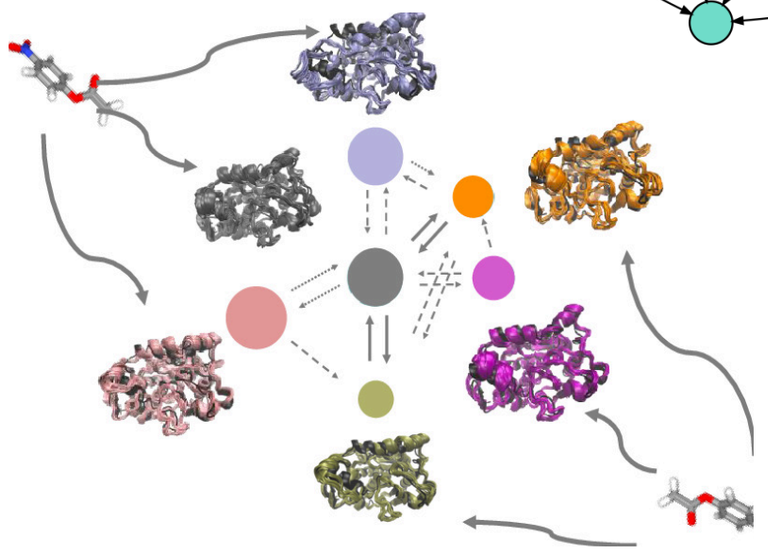
Macrostate model construction

- Given large Markov state model with sparse transitions, want to construct *macrostate model*
  - Many applications, esp. in chemical kinetics
- Want a principled, automatic framework
  - Can we select a few key states, then reduce Markov model to the resulting "marked process"





DNA and RNA secondary structures[1]



Competing enzymatic pathways[2]

[1] [Fornace et al. *ACS Synth Biol*, 2020]

[2] [Lu et al. *ACS Omega*, 2020]

Markov chains

Mathematical setup

$$\mathrm{R}_{i,j} = \begin{cases} r_{i \to j} & i \neq j \\ -\sum_{k \neq i} r_{i \to k} & i = j \end{cases}$$

- Transition rate matrix $R \in \mathbb{R}^{n \times n}$
  – probability is conserved (row stochastic): $R\mathbf{1} = \mathbf{0}$
  – $R_{i,j} \geq 0$ for all $i \neq j$

- Resulting probability evolution:

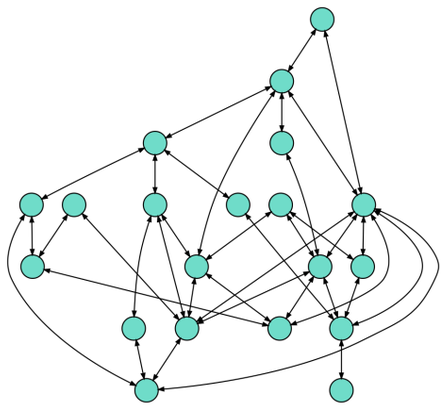
$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\phi}(t) = R^{\top}\tilde{\phi}(t) \implies \tilde{\phi}(t) = \tilde{P}^{\top}(t)\,\tilde{\phi}(0) \text{ where } \tilde{P}(t) := e^{Rt}$$

- Irreducible with stationary probability distribution π:

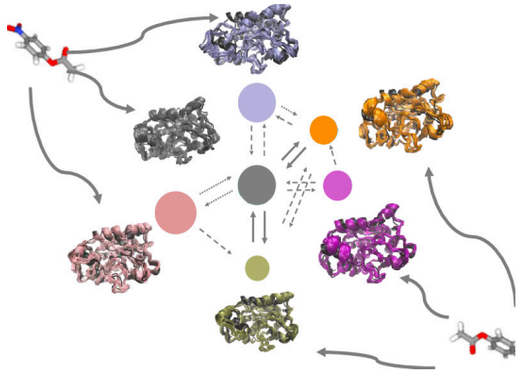
$$\lim_{t \to \infty} \tilde{\phi}(t) = \pi$$

- Reversible (detailed balance): $\mathrm{diag}(\pi)R = R^{\top}\,\mathrm{diag}(\pi)$

Overall motivation



Large weighted graph describing dynamics in chemistry, statistical mechanics, etc.



Small reduced rate model

Lu et al. ACS Omega, 2020

Graph Laplacians

$$\mathrm{R}_{i,j} = \begin{cases} r_{i\to j} & i \neq j \\ -\sum_{k\neq i} r_{i\to k} & i = j \end{cases} \quad \Rightarrow \quad L = -\mathrm{Diag}(\pi^{1/2}) R \, \mathrm{Diag}(\pi^{-1/2})$$

Symmetrized treatment

L is (1) symmetric, (2) positive semidefinite, (3) an M-matrix, (4) has single zero eigenvector $h = \sqrt{\pi}$ ($\mathbf{1}^\top \pi = 1$ so $\|h\|_2 = 1$)

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi(t) = -L\phi(t) \quad \Rightarrow \quad P(t) := e^{-Lt} = \mathrm{diag}(h) e^{Rt} \, \mathrm{diag}^{-1}(h)$$

Main operator that we will approximate

Relation to graph Laplacians

- Let $\Delta = \mathrm{diag}(h) L \, \mathrm{diag}(h)$. Then Δ is a graph Laplacian (still symmetric but now $\Delta\mathbf{1} = 0$)
- If $h = \mathbf{1}/\sqrt{n}$, then $L = \Delta = -R$ without rescaling
- For us, L and h are known in advance; we work with these and just call L "Laplacian"

Graph Laplacian CSSP

- Roughly, want to perform **column selection** for the kernel L^{-1} (ill-defined)
  - In L^{-1}, modes weighted inversely by timescale
  - **Intuition:** many rapid/transient modes, few slow modes $\Longrightarrow$ low rank

- Instead, selection for $K_\gamma := L_\gamma^{-1} = (L + \gamma h h^\top)^{-1}$ in limit $\gamma \to 0$

Note: $K_\gamma = \int_0^\infty P_\gamma(t)\, dt$ where $P_\gamma(t) = e^{-L_\gamma t}$

(similarity-transformed occupation time of killed process)

- **Markov chain compressibility $\leftrightarrow$ low-rankness of K_γ**
  (modulo rank-one contribution of stationary coefficient blowing up as $\gamma \to 0$)

Implementation and guarantees for Laplacian CSSP

- Fast matrix-free implementation requires manipulation of $K = L^+$
  - Symmetric factorization yielded by $K = \sqrt{L^+}\sqrt{L^+}$
  - Multiplication by $\sqrt{L^+}$ using fast Laplacian preconditioner, polynomial approximation of a preconditioned matrix square root
  - Scaling is essentially optimal: $\tilde{O}(nk^2 + k\,\mathrm{nnz}(L))$
- Can show using probabilistic interpretation that subset selection objective is ***submodular***, allowing us to establish bound relative to *optimal* subset
  (better than DPP expectation)

Compressed dynamics

- Given an $V \in \mathbb{R}^{n \times k}$ with orthonormal columns, compress

$$P_V(t) = V e^{-(V^\top L V)t} V^\top$$

- **Theorem:**

$$\|P_V(t) - P(t)\| \leq \frac{3\sqrt{3}}{2\pi} \frac{\varepsilon}{t}$$

where norm is nuclear or spectral, $\varepsilon = \|L^+ - V(V^\top L V)^+ V^\top\|$

(ε is roughly slowest timescale not captured by approximation)

- Proof by contour integration argument

Compressed dynamics

$$P_V(t) = V\, e^{-(V^\top L V)t}\, V^\top$$

$$\|P_V(t) - P(t)\| \leq \frac{3\sqrt{3}}{2\pi}\,\frac{\varepsilon}{t}$$

- **Special case:** $V_\gamma = \mathrm{orth}\left([K_\gamma]_{:,\mathcal{I}}\right)$ in $\gamma \to 0$ limit
  - ε recovers limiting Nyström approximation error for subset $\mathcal{I}$
  - V can in this case be related to the committor matrix C for selected sites

$$\int_0^\infty [P_\gamma(t) - P_{\mathcal{I},\gamma}(t)]\,\mathrm{d}t = K_\gamma - (K_\gamma)_{:,\mathcal{I}}(K_\gamma)^{-1}_{\mathcal{I},\mathcal{I}}(K_\gamma)_{\mathcal{I},:}$$

Contour integral argument

- **Resolvent analysis:** Express matrix exponentials via contour integrals of the resolvents ($z \in \mathbb{C}$)

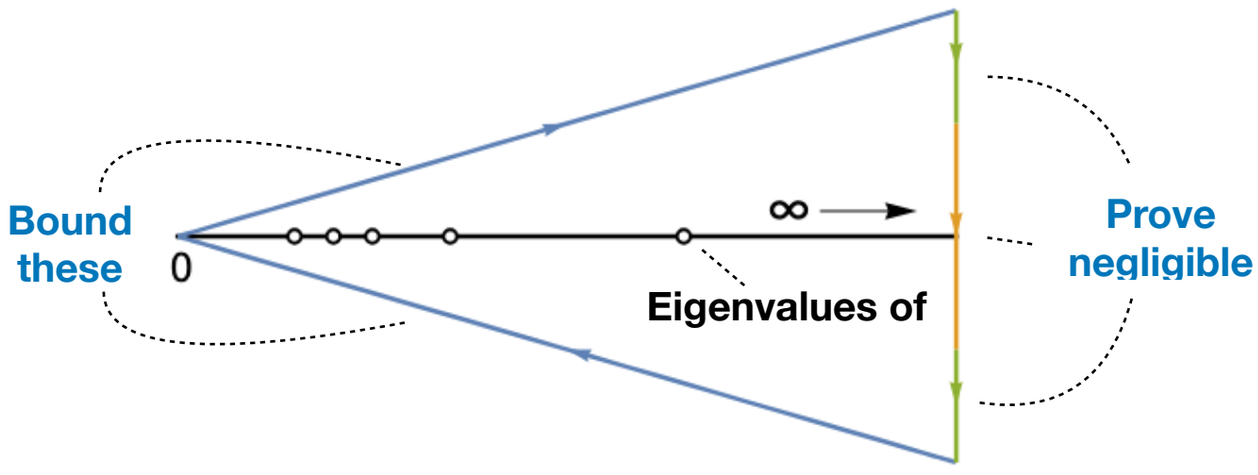
  Full: $$P(t) = \frac{1}{2\pi i} \oint_{\mathcal{C}} e^{-1/z}\, \mathcal{Q}_t(z)\, \mathrm{d}z \quad \text{where } \mathcal{Q}_t(z) \coloneqq (z\mathbf{I} - K/t)^{-1}$$

  Projective: $$P_{\mathcal{I}}(t) = \frac{1}{2\pi i} \oint_{\mathcal{C}} e^{-1/z}\, \mathcal{R}_t(z)\, \mathrm{d}z \quad \text{where } \mathcal{R}_t(z) = \left(z\mathbf{I} - (K_{:,\mathcal{I}} K_{\mathcal{I},\mathcal{I}}^{-1} K_{\mathcal{I},:})/t\right)^{-1}$$

- Subtract integrands, apply resolvent identities, and substitute so that:

$$\mathcal{Q}_t(z) - \mathcal{R}_t(z) = \mathcal{Q}_t(z)\left[\mathcal{R}_t(z)^{-1} - \mathcal{Q}_t(z)^{-1}\right]\mathcal{R}_t(z) \quad \Rightarrow \quad \|\mathcal{Q}_t(z) - \mathcal{R}_t(z)\|_{\{2,*\}} \leq \frac{\varepsilon_{\{2,*\}}}{t\,|\mathrm{Im}(z)|^2}$$

- **Contour integral argument:** Construct a contour, bound on all segments, take limits carefully



$$\|P(t) - P_{\mathcal{I}}(t)\|_{\{2,*\}} \leq \frac{3\sqrt{3}}{2\pi} \frac{\varepsilon_{\{2,*\}}(\mathcal{I})}{t}$$

Main projective compression bound

Structure-preserving approximation

- Structure-preserving compression is *not* an orthogonal projection (harder to bound!)

$$P_{\mathcal{I}}(t) = C \exp\left(-C^{+} L C t\right) C^{+}$$

Projective

$$P_{\mathcal{I}}^{\mathrm{sp}}(t) := C e^{-\hat{L} t} C^{\top}, \quad \text{where } \hat{L} := C^{\top} L C$$

Structure-preserving

- Swapping C^{+} for $C^{\top}$ introduces a much more complicated problem
- But it preserves the intuition and structure of the input Laplacian in a nice way:
  - $\hat{L}$ is a valid Laplacian with fluxes and stationary distribution determined by C
  - Straightforwardly the reduction desired by chemists, etc.
  - Empirically, not much worse or different than projective…but we want to make this precise

Error bound summary

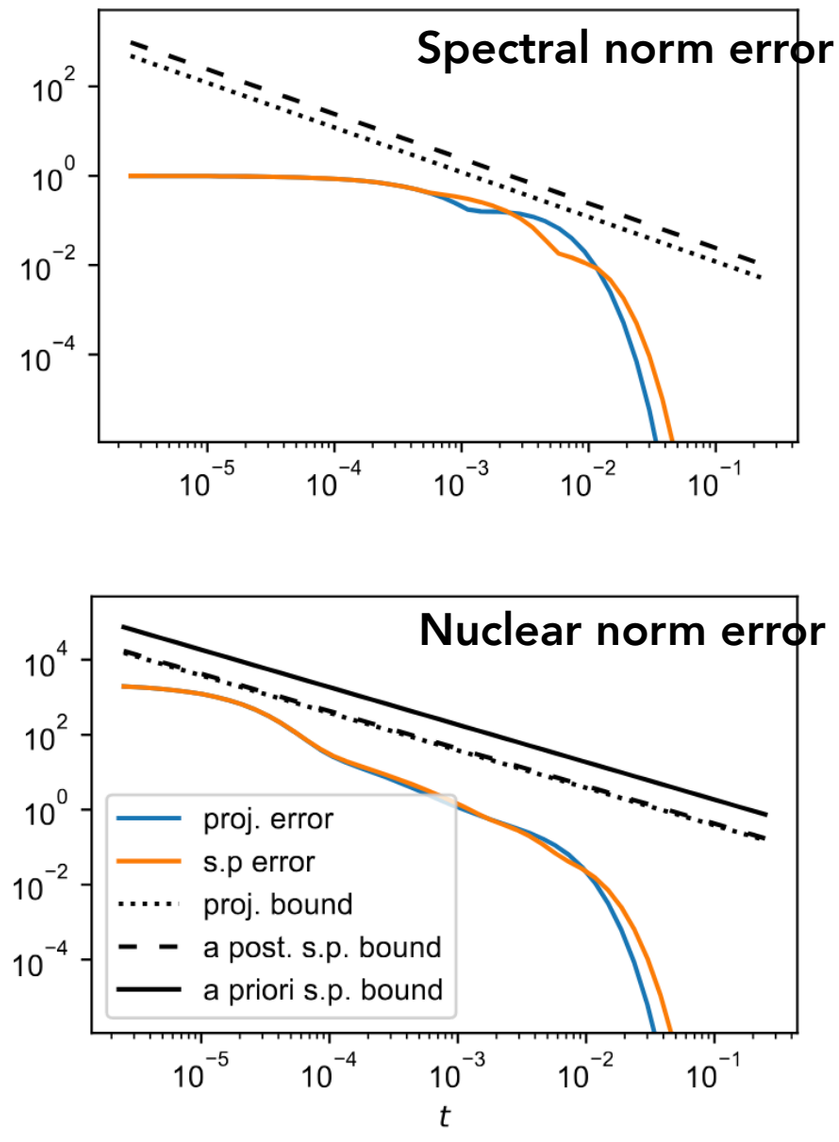
New and simple bounds for both approximations:

Projective approximation

$$\|P(t) - P_{\mathcal{I}}(t)\|_{\{2,*\}} \leq \frac{3\sqrt{3}}{2\pi} \frac{\varepsilon_{\{2,*\}}(\mathcal{I})}{t}$$

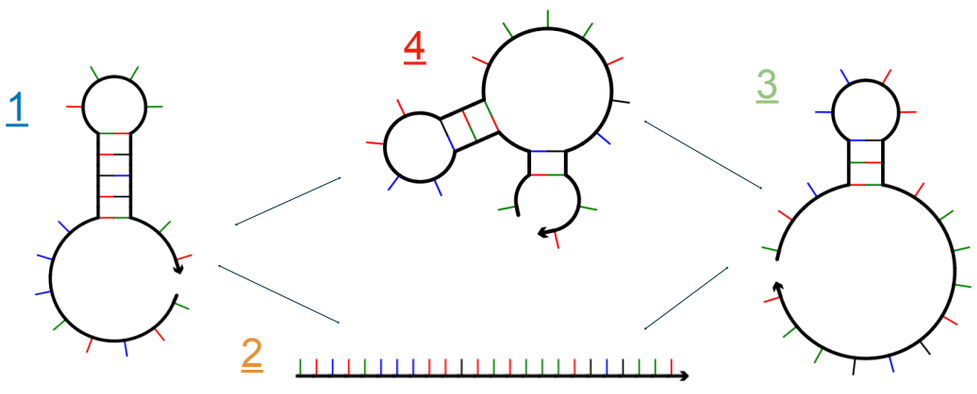
Structure-preserving approximation

$$\|P^{\mathrm{sp}}_{\mathcal{I}}(t) - P(t)\|_{*} \leq \left(\frac{3\sqrt{3}}{2\pi} + |\mathcal{I}| \frac{2}{\pi} \right) \frac{\varepsilon_{*}(\mathcal{I})}{t}$$

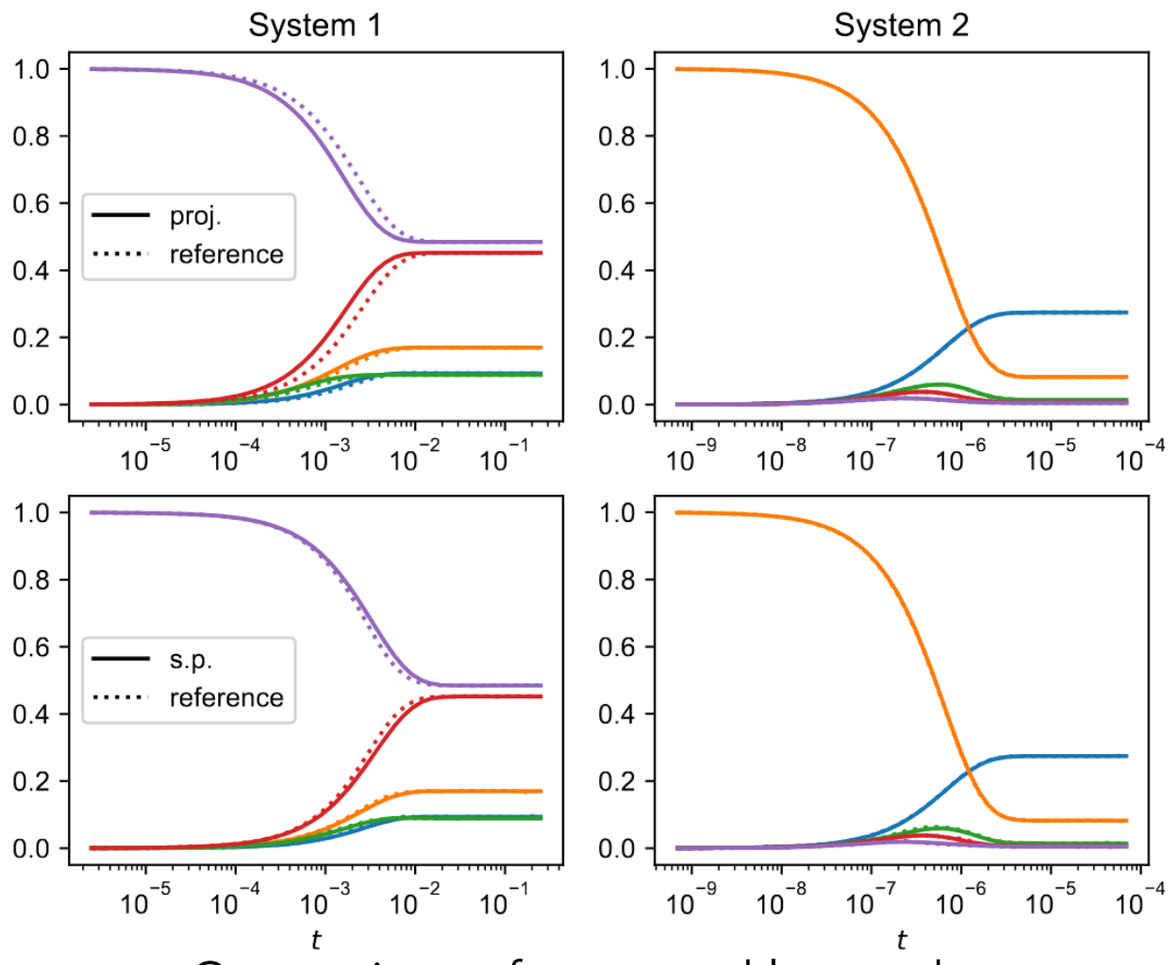


DNA secondary structure chemical kinetics

- We verify that subsets via nuclear maximization do a good job of minimizing nuclear and spectral norm errors
  - Systems up to $n = 10^6$
- The error in the projected subspace is also very small (right)

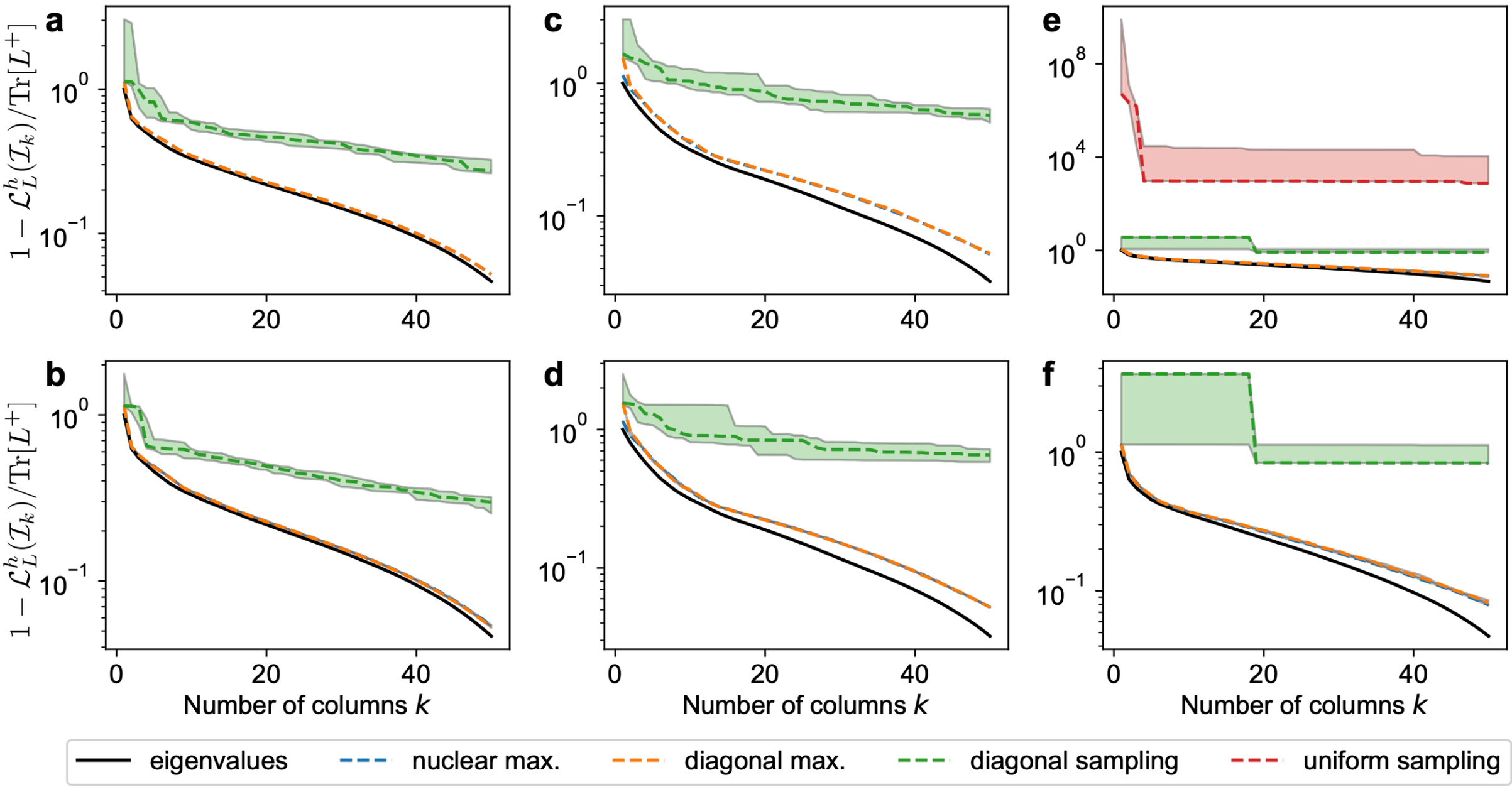


Reduced Markov state model (RNA folding)



Comparison of exact and low-rank dynamics in the projected subspace (1 macrostate per color)

DNA secondary structure chemical kinetics



Conclusions

- **Nuclear maximization:** strong theoretical guarantees and strong practical performance for CSSP
  - General case: at least about as good as DPP (which is almost as good as the eigenvalue bound)
  - Laplacian case: nearly combinatorially optimal
- **Markov chain compression:** can be connected to CSSP!
  - Nyström error controls recovery error and sets timescale of effective approximation
  - Full dynamics recovery
  - No assumption of time scale separation
- **Ongoing work:** construct Markov state model from sampled trajectories on continuous space (e.g., molecular dynamics), compatible with further compression via column selection

References

Joint work with
Mark Fornace (LBNL)

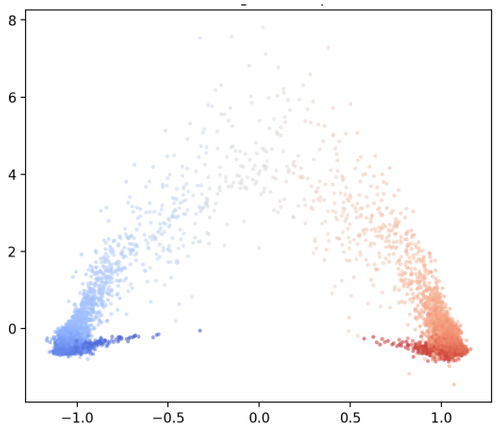
CSSP:

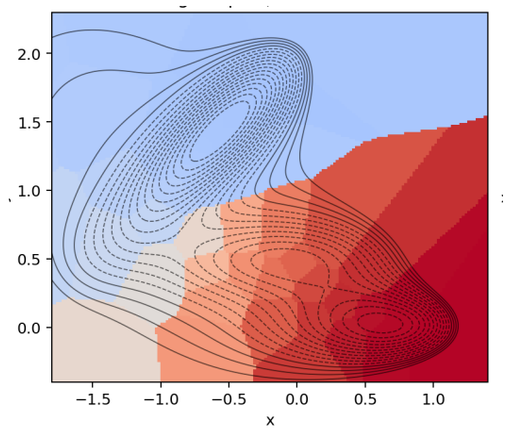
arXiv:2407.01698

Markov chain compression:

arXiv:2506.22918

learn slow modes





embed and cluster

peptide helix formation (messy!)