“Collect, Commit, Expand”: A Strategy for Faster CPQR-Based Column Selection on Short, Wide Matrices

Robin Armstrong

PhD Candidate, Cornell University, Center for Applied Mathematics

Mid-Atlantic Numerical Analysis Day, November 15th, 2024

Joint work with…

Anil Damle
Assistant Professor, Cornell University
Department of Computer Science

The Column Subset Selection Problem

- Let $A \in \mathbb{R}^{m \times n}$, $k \geq 1$. **Find k columns that are large and linearly independent.**
- Formalized as volume maximization:

$$\max_{S \in [n]^k} \mathrm{vol}(A(:, S)),$$

  where $\mathrm{vol}(X) = \sqrt{\det(X^T X)}$.

- Formalized as singular value maximization:

$$\max_{S \in [n]^k} \sigma_{\min}\big(A(:, S)\big)$$

- **These problems are NP-hard** [2].

[2] Civril and Magdon-Ismail, Theoretical Computer Science (2009).

CSSP in Matrix Approximation

- How to approximate a matrix in terms of its own columns? We want:

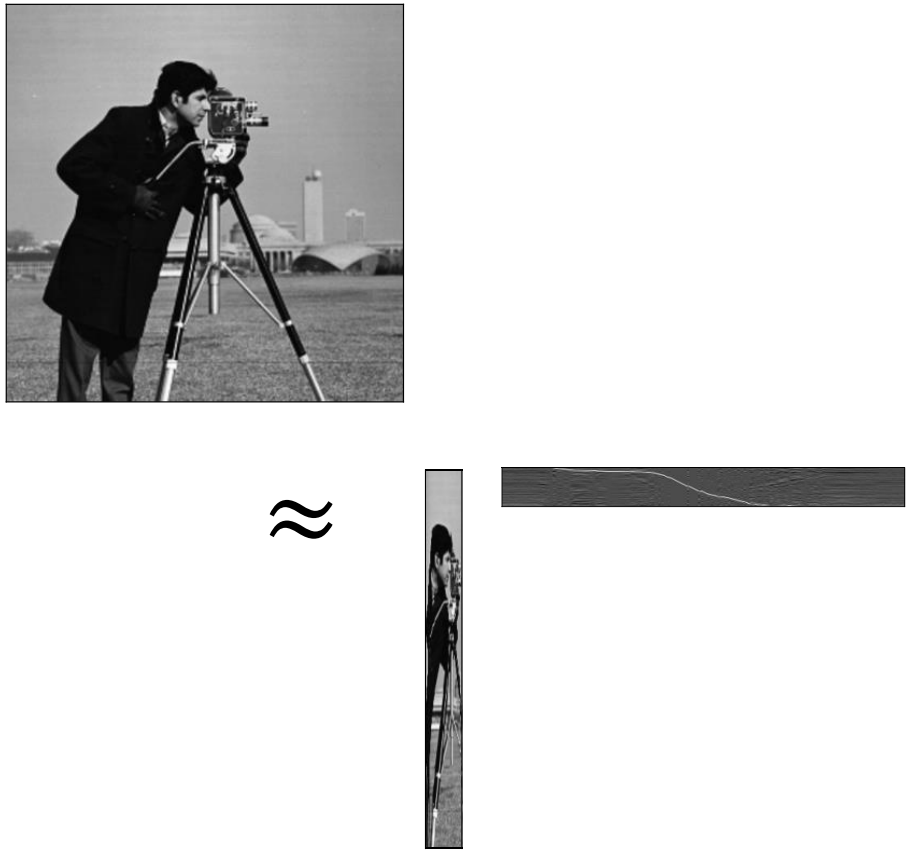
$$\min_{S \in [n]^k} \left\| A - \hat{A}_S \right\|$$

where $\hat{A}_S = A(:, S) A(:, S)^+ A$.

- **Solution**: make $\mathrm{vol}\big(A(:, S)\big)$ large!

- **Theorem** [1]: If $\mathrm{vol}\big(A(:, S)\big)$ is within a factor $\mu \geq 1$ of its maximum over $S \in [n]^k$, then

$$\left\| A - \hat{A}_S \right\|_\infty \leq \mu(k+1) \cdot \min_{\mathrm{rank}\, A' \leq k} \|A - A'\|_2$$



[2] Civril and Magdon-Ismail, Theoretical Computer Science (2009).

CSSP in Matrix Approximation

- How to approximate a matrix in terms of its own columns? We want:

$$\min_{S \in [n]^k} \left\| A - \hat{A}_S \right\|$$

where $\hat{A}_S = A(:, S) A(:, S)^+ A$.

- **Solution**: make $\mathrm{vol}\big(A(:, S)\big)$ large!

- **Theorem** [1]: If $\mathrm{vol}\big(A(:, S)\big)$ is within a factor $\mu \geq 1$ of its maximum over $S \in [n]^k$, then

$$\left\| A - \hat{A}_S \right\|_\infty \leq \mu(k+1) \cdot \min_{\mathrm{rank}\, A' \leq k} \|A - A'\|_2$$

$\approx$

[2] Civril and Magdon-Ismail, Theoretical Computer Science (2009).

CSSP in Model Order Reduction

- Given nonlinear dynamics:

$$\frac{dx}{dt} = Ax(t) + f\big(x(t)\big),$$

...and a **reduced-order surrogate**: $x(t) \approx Vz(t)$,

$$\frac{dz(t)}{dt} = (V^T AV)z(t) + V^T f\big(Vz(t)\big)$$

- **Can we evaluate only a few components of f?**

- DEIM algorithm [1]: **solve $j_1 \dots j_k \leftarrow$ CSSP(V^T),** evaluate $f_{j_1}\big(V^T z(t)\big) \dots f_{j_k}\big(V^T z(t)\big)$, and use interpolary projection to estimate the remaining components.

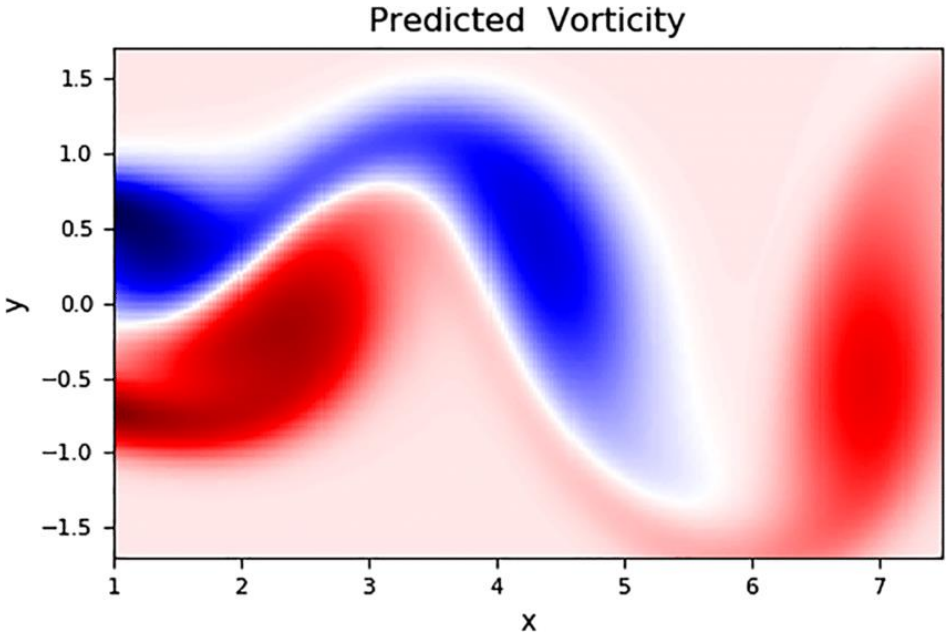


Image source: Yang Li and Fangjun Mei, *Deep learning-based method coupled with small sample learning for solving partial differential equations*, Multimedia Tools and Applications (2021).

[1] Chaturantabut and Sorensen, SIAM Journal on Scientific Computing (2010).

Column Pivoted QR Factorization

- **Projection pursuit** is a greedy algorithm for CSSP.

1. **for** $i = 1, \ldots, k$:
2. find the largest column.
3. add it to the "skeleton set".
4. orthogonally project all columns off of it.

- **Column-pivoted QR** implements projection pursuit as a matrix factorization.

1. $R \leftarrow A,\ Q \leftarrow I_m,\ \Pi \leftarrow I_n$.
2. **for** $i = 1, \ldots, m$:
3. find the largest residual norm (look at R).
4. **swap** that column to the front (modify Π, R).
5. **rotate** to expose new residual norms (modify Q, R).

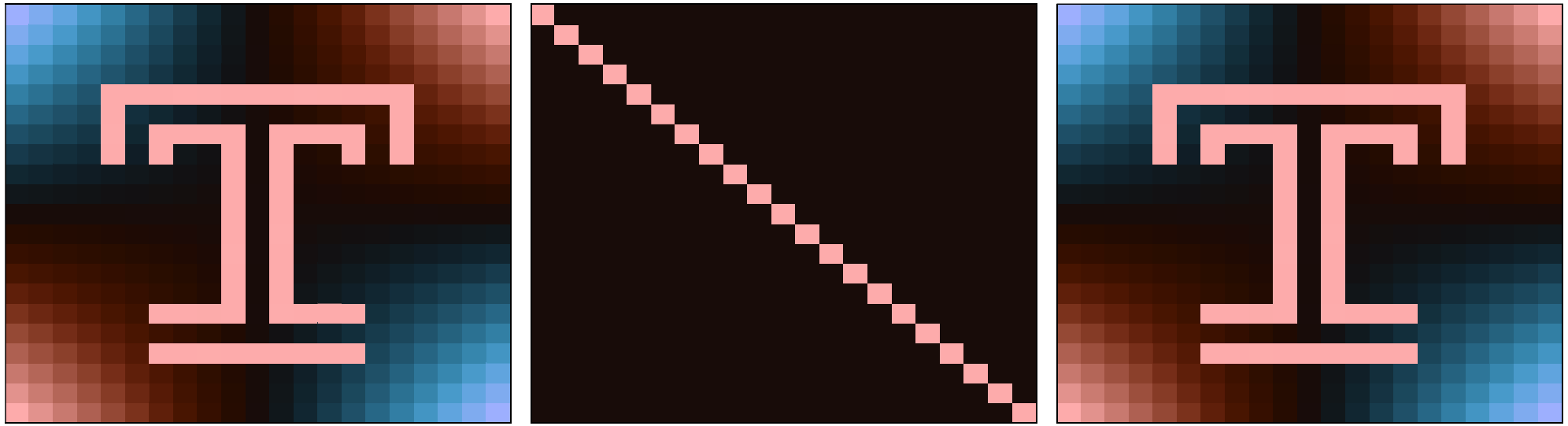
input matrix

transformed input; triangular structure reveals geometry of basis columns.

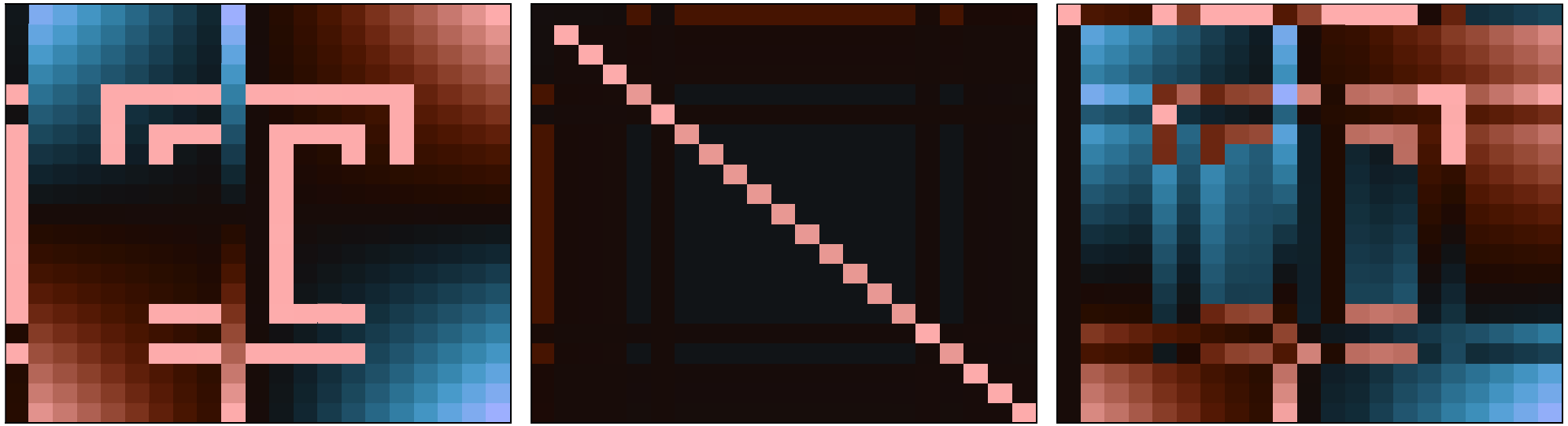
$$A\Pi = QR$$

permutation; moves basis columns to the front

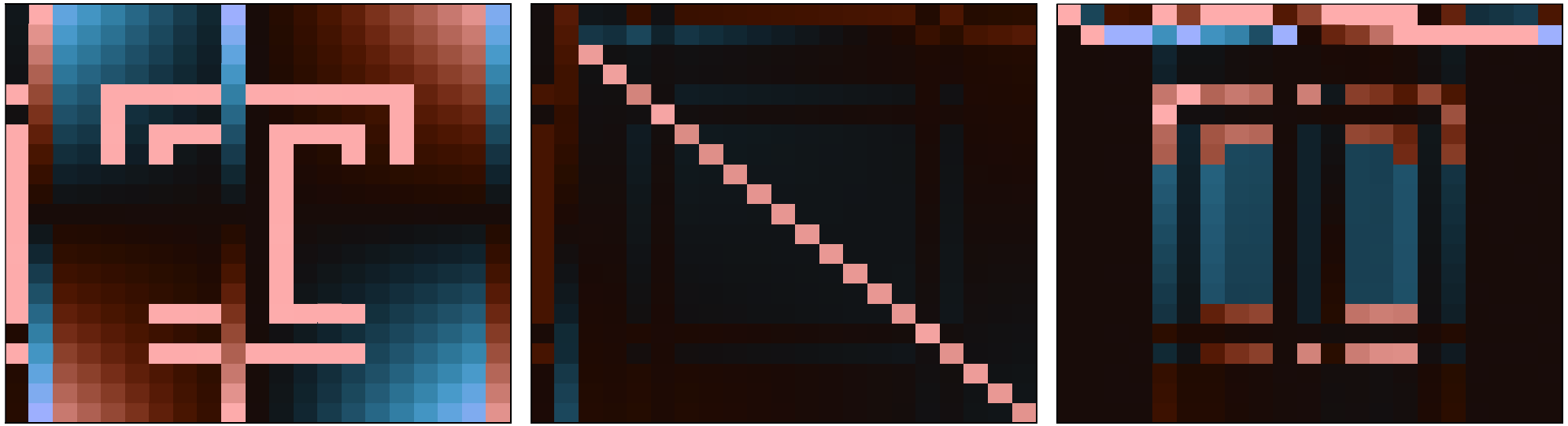
rotation



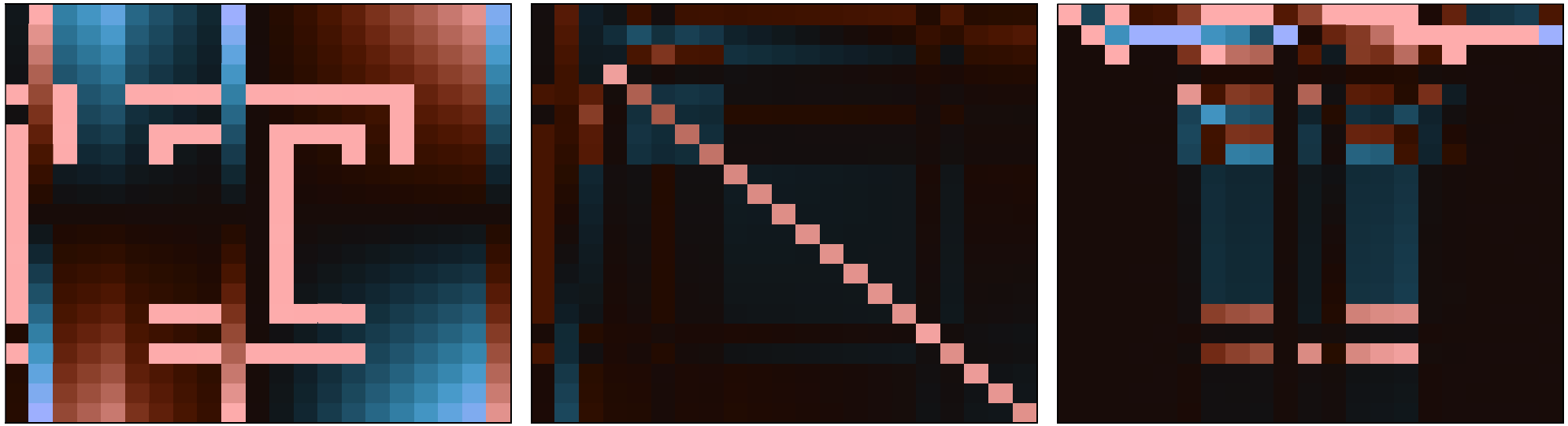
$$A\Pi = Q \qquad R$$



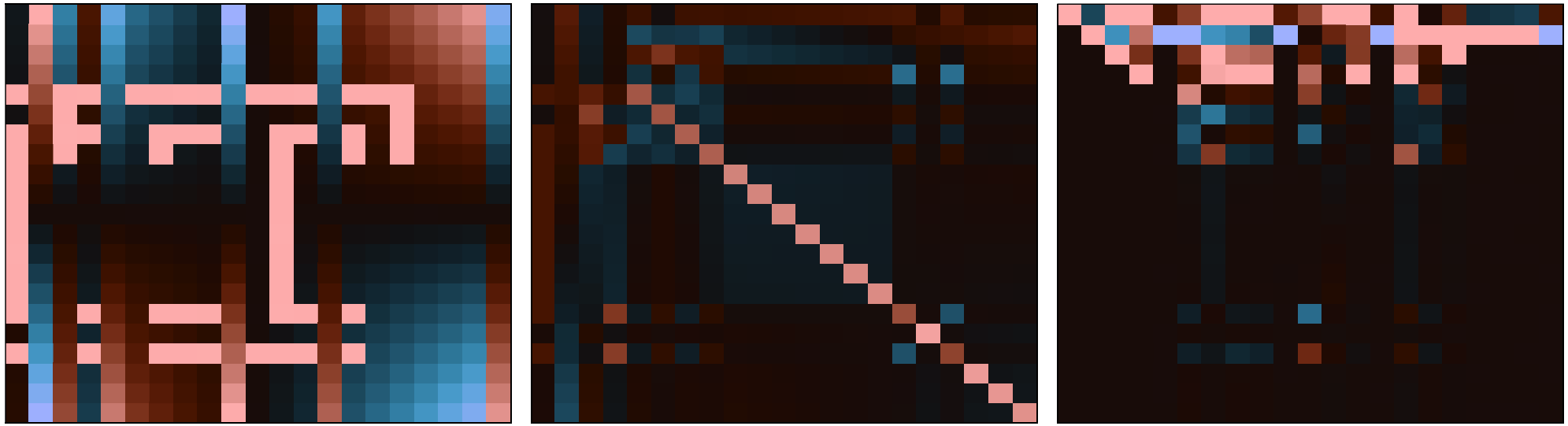
$$A\Pi = Q \qquad R$$



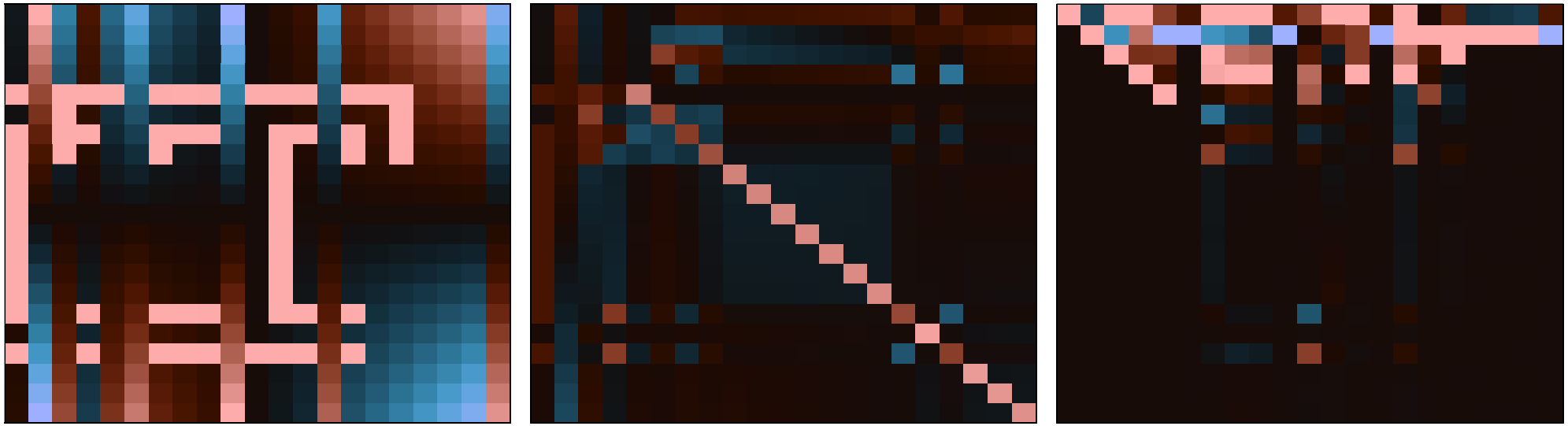
$$A\Pi = Q \qquad R$$



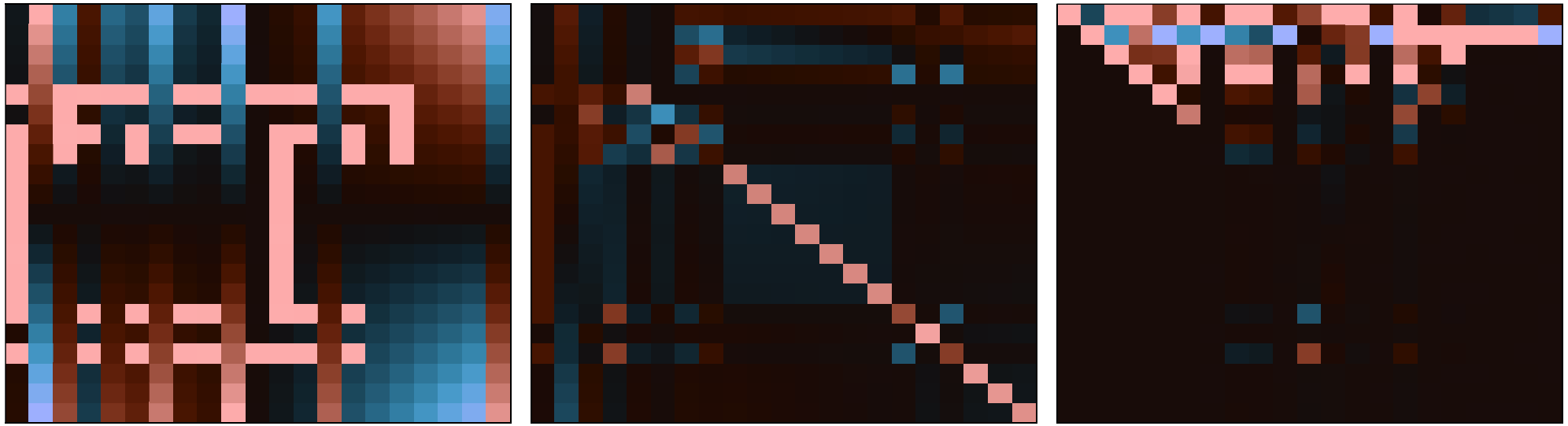
$$A\Pi = Q \qquad R$$



$$A\Pi = Q \qquad R$$



$$A\Pi = Q \qquad R$$



$$A\Pi = Q \qquad R$$

Towards More Efficient Column Selection

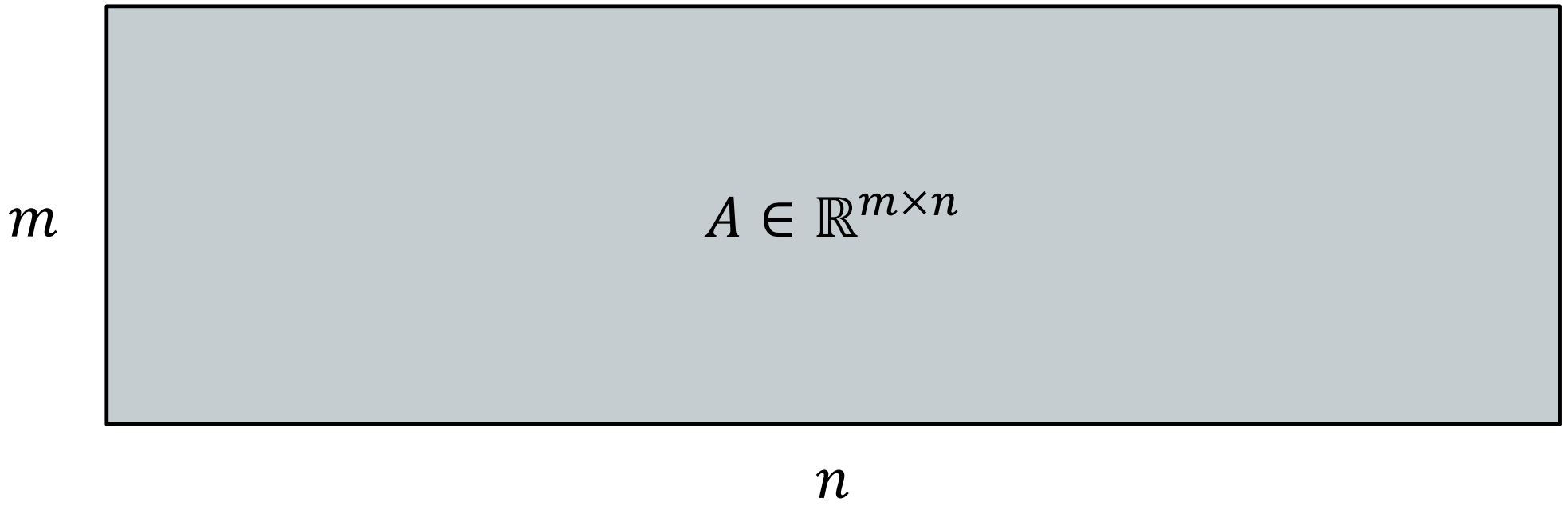
- **BLAS-2 Householder reflections are the most expensive part of CPQR.**
- **Can we limit the amount of work dedicated to reflections?**
- Previous solutions: reflect only a few rows at a time, defer the full reflection to BLAS-3.
  - …with partial Householder reflections [6], or
  - …with randomized sketching [5, 7].
- **What about matrices with far more columns than rows?**
  - **Spectral clustering** [4]: rows ⟷ clusters, columns ⟷ data points.
  - **Model order reduction** [1]: rows ⟷ reduced coordinates, columns ⟷ full coordinates.
  - **Computational chemistry** [3]: rows ⟷ orbitals, columns ⟷ 3D grid points.

[1] Chaturantabut and Sorensen, SIAM Journal on Scientific Computing, 2010.
[3] Damle, Lin, and Ying, Journal on Chemical Theory and Computing, 2015.
[4] Damle, Minden, and Ying, Information and Inference, 2018.
[5] Martinsson, Quintana-Ortí, Heavner, and de Geijn, SIAM Journal on Scientific Computing, 2017.
[6] Quintana-Ortí, Sun, and Bischof, SIAM Journal on Scientific Computing, 1998.
[7] Woolfe, Liberty, Rokhlin, and Tygert, Applied and Computational Harmonic Analysis, 2008.

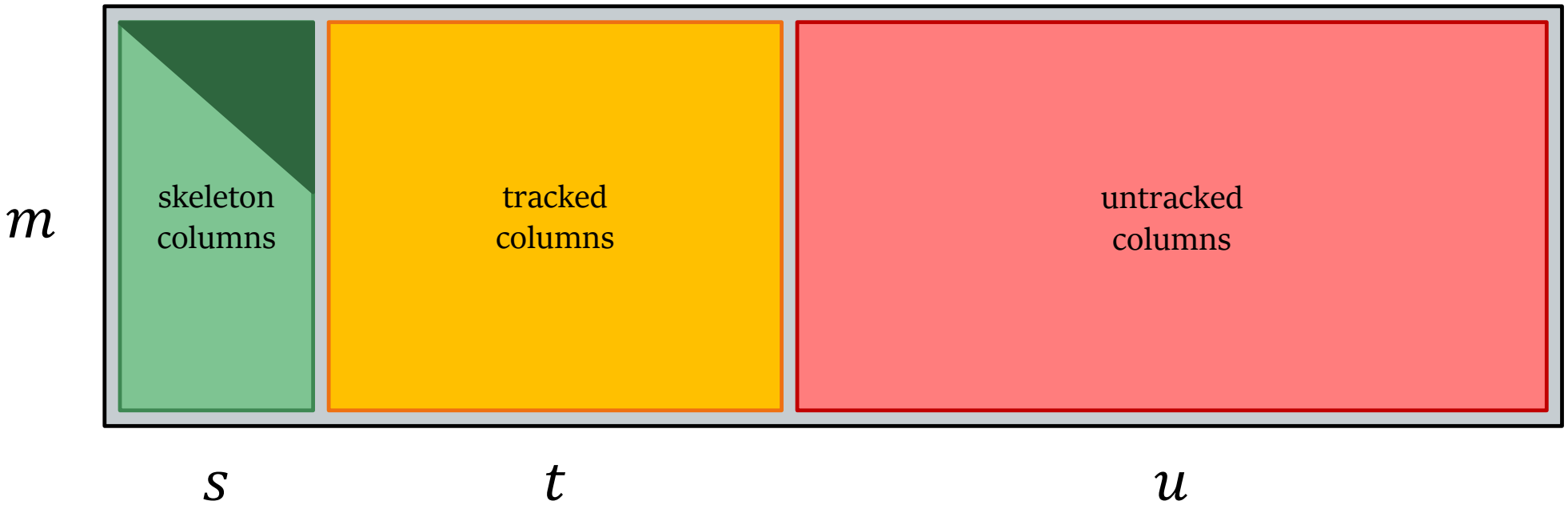
“Collect, Commit, Expand”

- **Main Idea no. 1**: large-norm columns are more likely to be good basis columns.
- **Main Idea no. 2**: apply CPQR on only a subset of large columns, then check correctness.
- **Lemma.** Let $A = [A_1 \;\; A_2]$ and let $A\Pi = QR$, $A_1\Pi_1 = Q_1R_1$ be CPQR factorizations. Suppose that for some $i \geq 1$,
$$|R_1(i,i)| \geq \max_j \|A_2(:,j)\|_2 .$$
Then, assuming no ties in residual column norm,
$$A\Pi(:,1:i) = A_1\Pi_1(:,1:i).$$
- **Algorithm overview**:
  - Collect the tracked columns with largest norm (“candidates”).
  - Commit a few of them into the basis (using a smaller CPQR).
  - Expand the tracked set to new columns (using a lower norm threshold).
  - Repeat.

Algorithm Setup

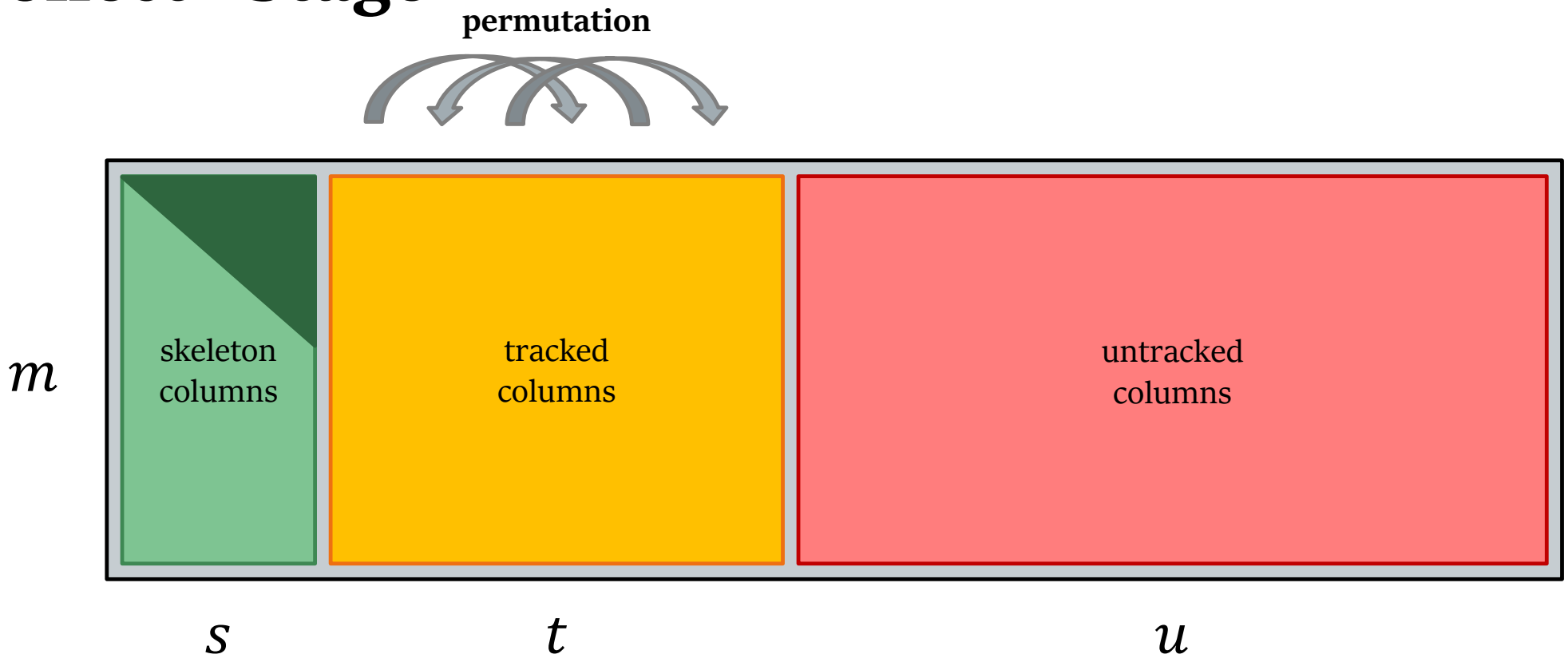


Algorithm Setup



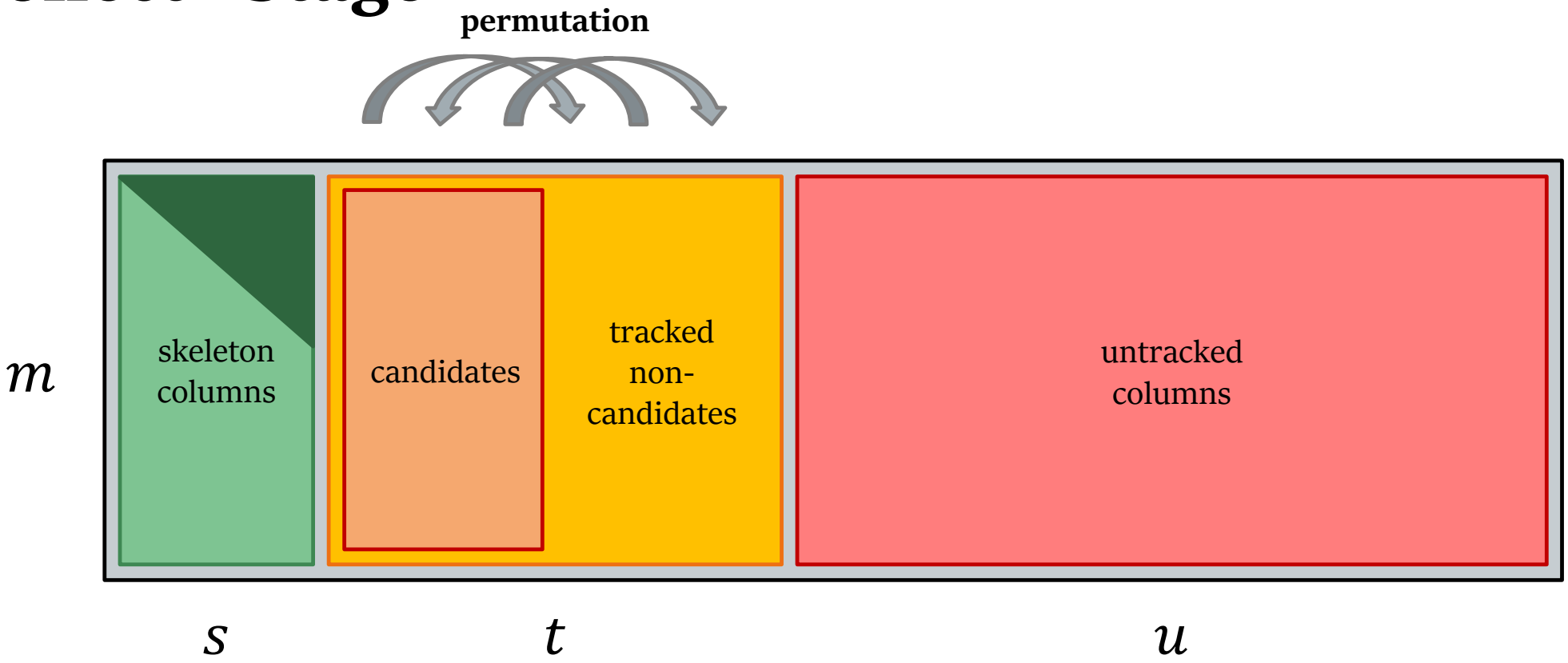
- "Tracked" columns have large residual norm, "untracked" columns have smaller overall norm.

“Collect” Stage



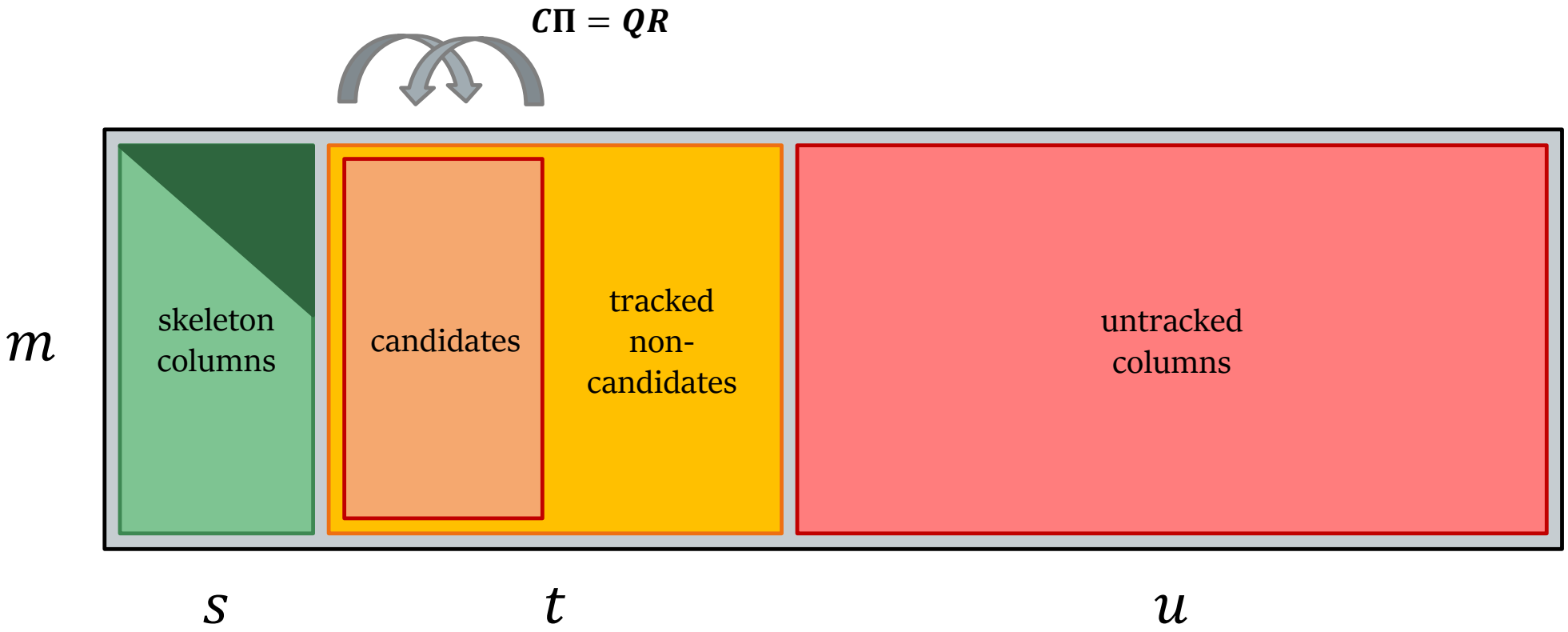
- Move the largest tracked columns to the front.

“Collect” Stage



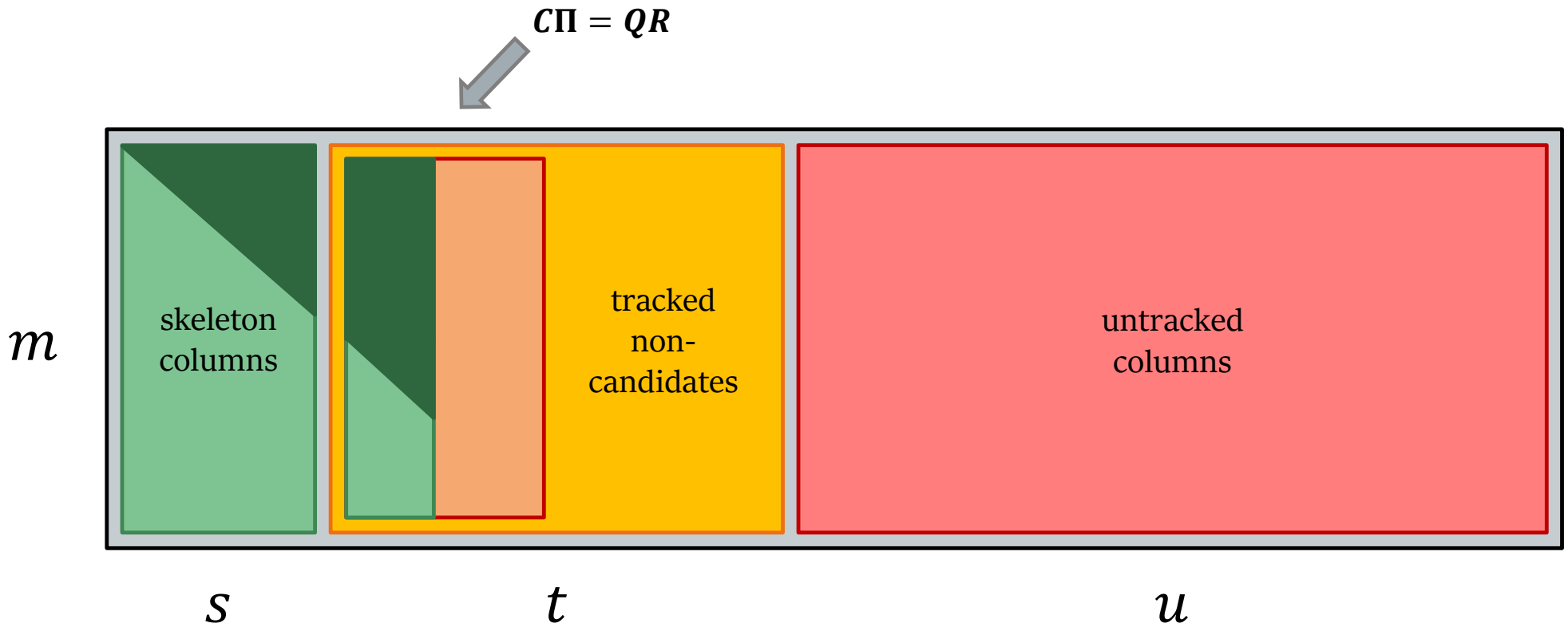
- Move the largest tracked columns to the front.

“Collect” Stage



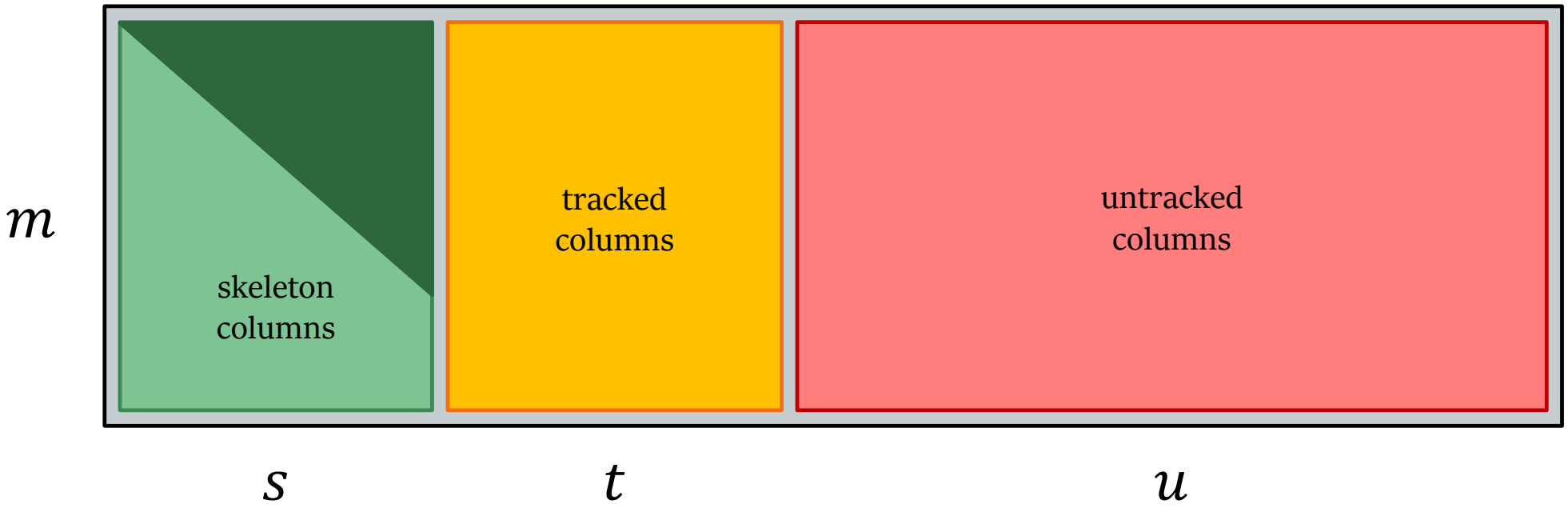
- **CPQR factorization of candidates**; standard Householder reflections.

“Commit” Stage



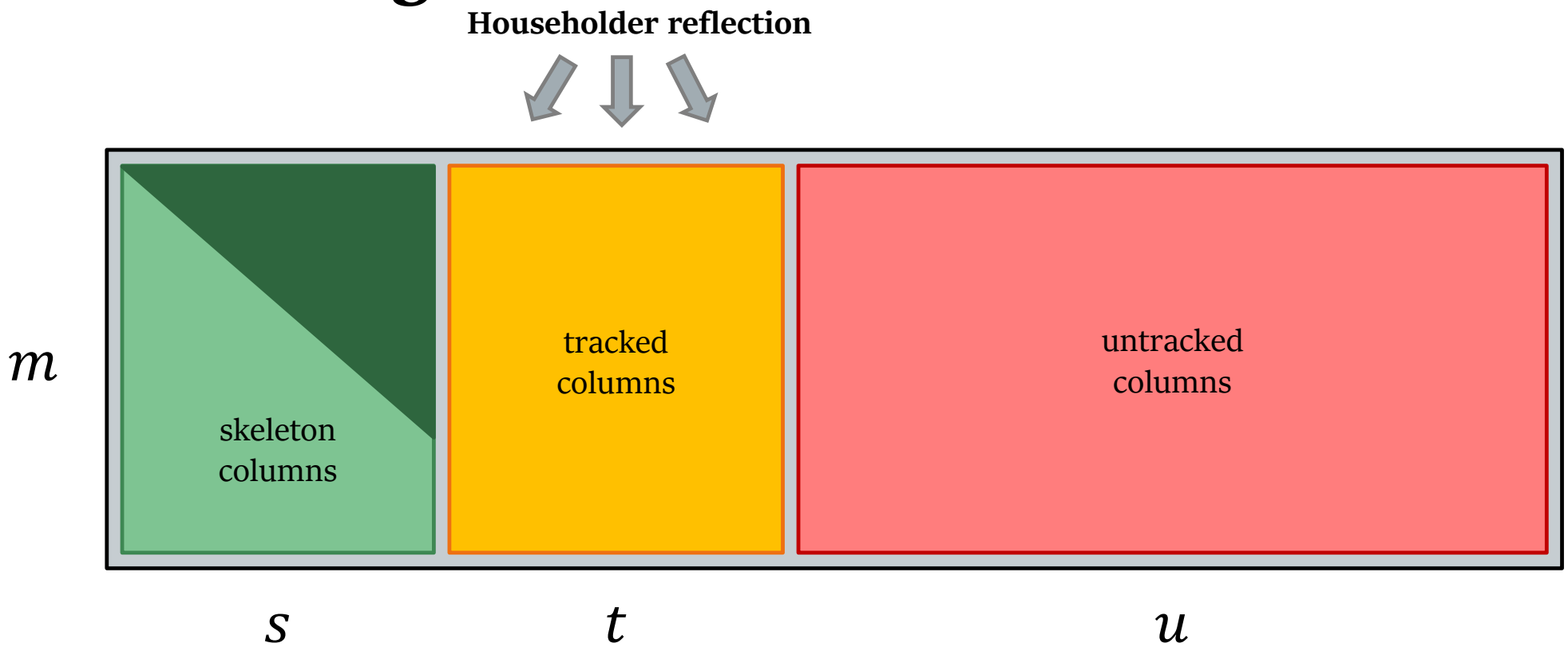
- **Examine CPQR factors to decide which candidates go into the skeleton.**

“Commit” Stage



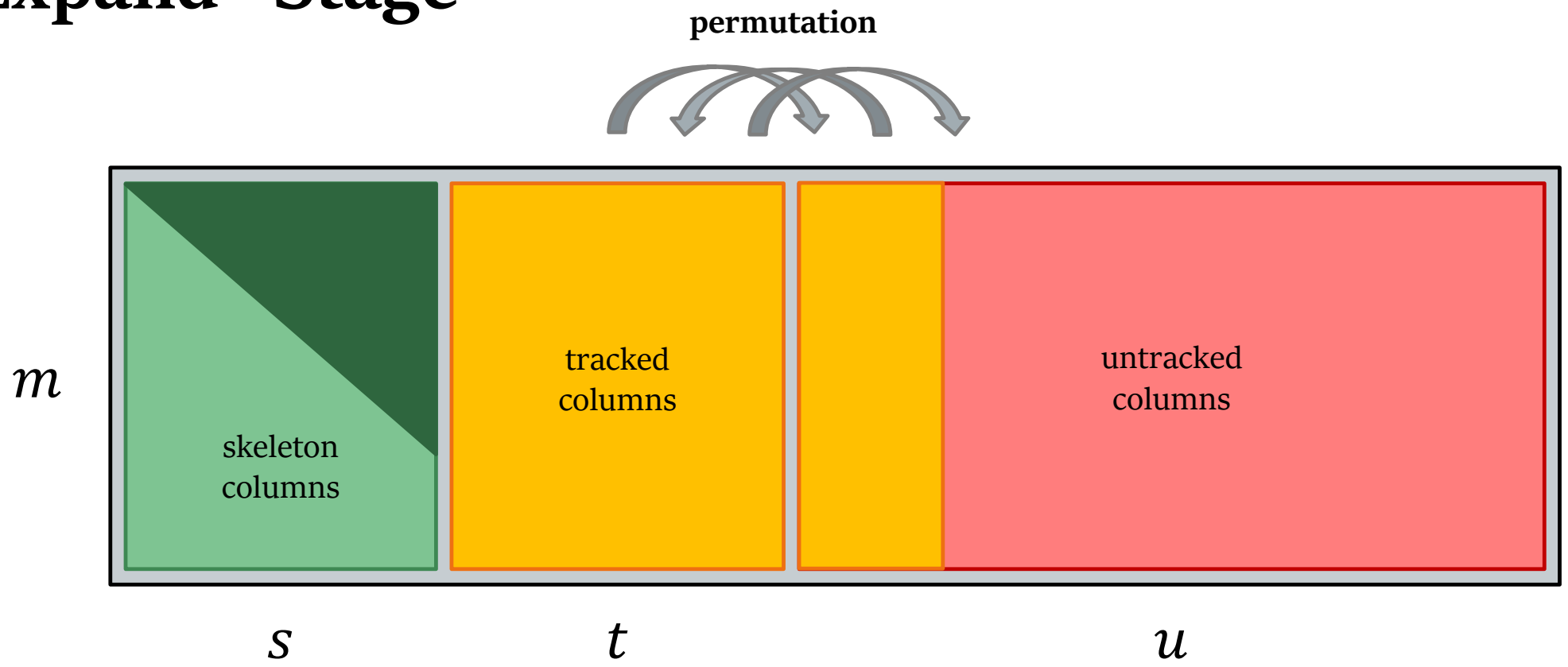
- **Examine CPQR factors to decide which candidates go into the skeleton.**

“Commit” Stage



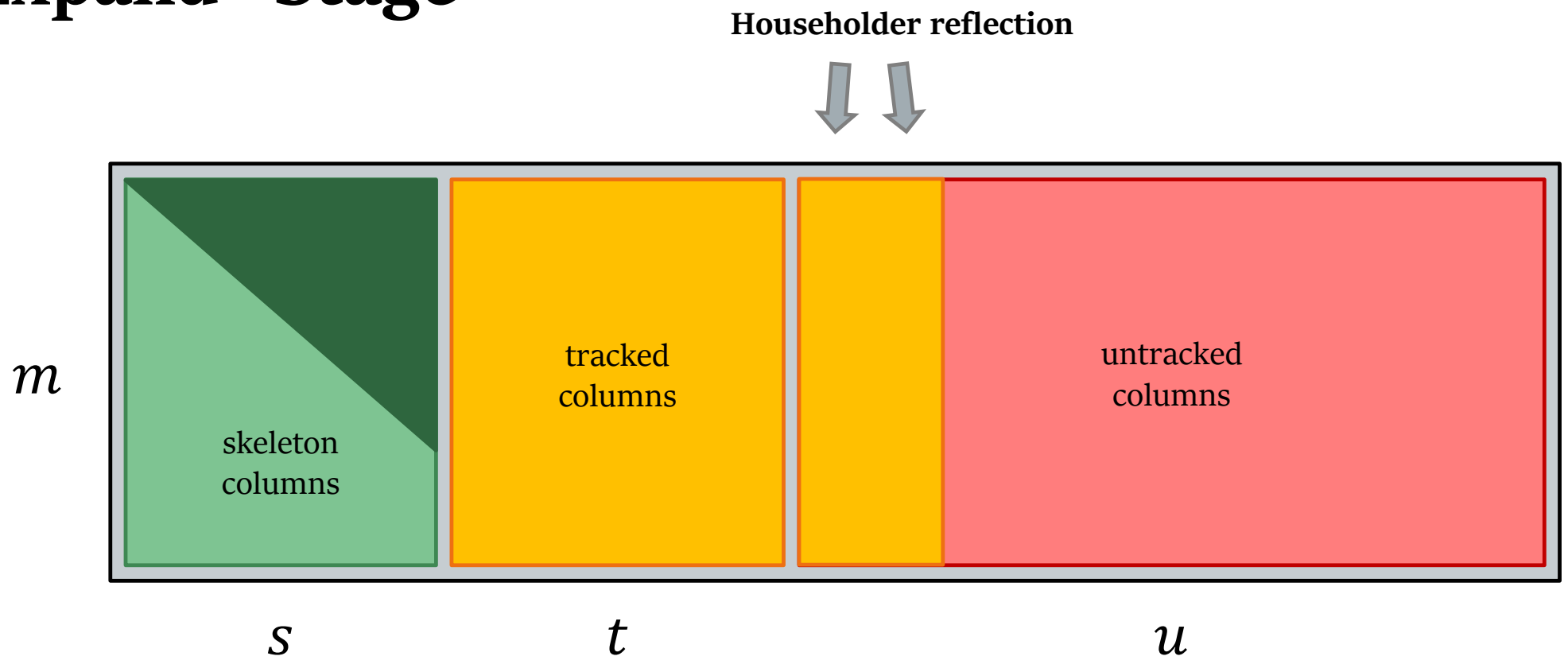
- Apply a BLAS-3 reflection to the tracked set; update residual norms.

“Expand” Stage



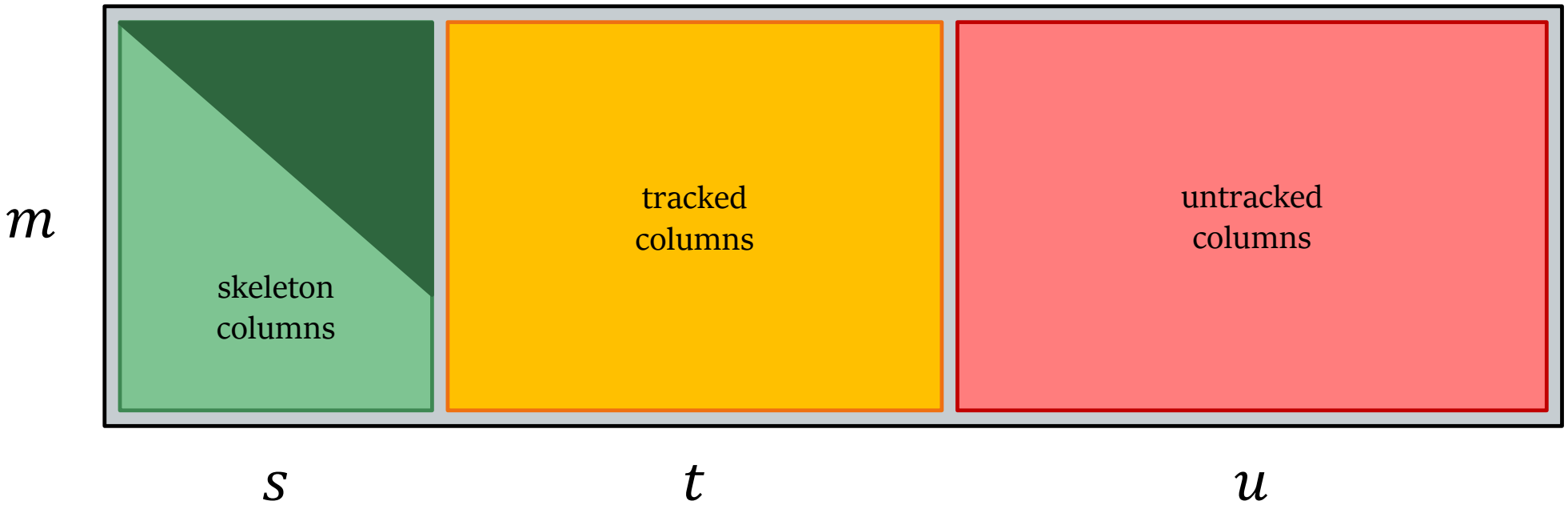
- Move the largest “untracked” columns into the “tracked” set.

“Expand” Stage



- Apply all previous reflections to the newly tracked columns (BLAS-3); update their residual norms.

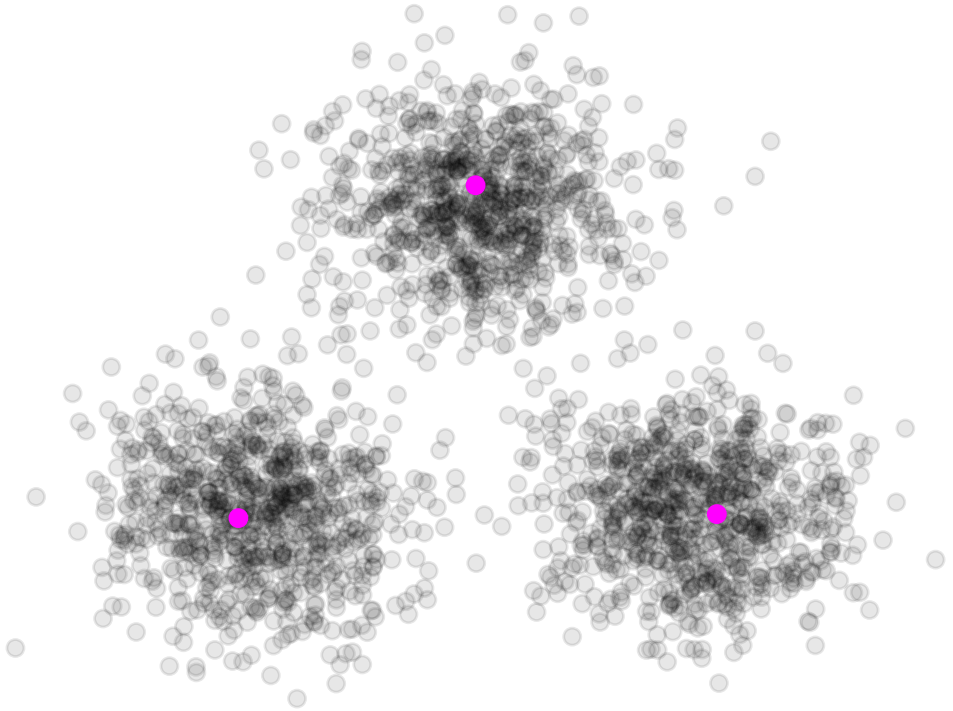
“Expand” Stage



- **Repeat until the skeleton is complete.**

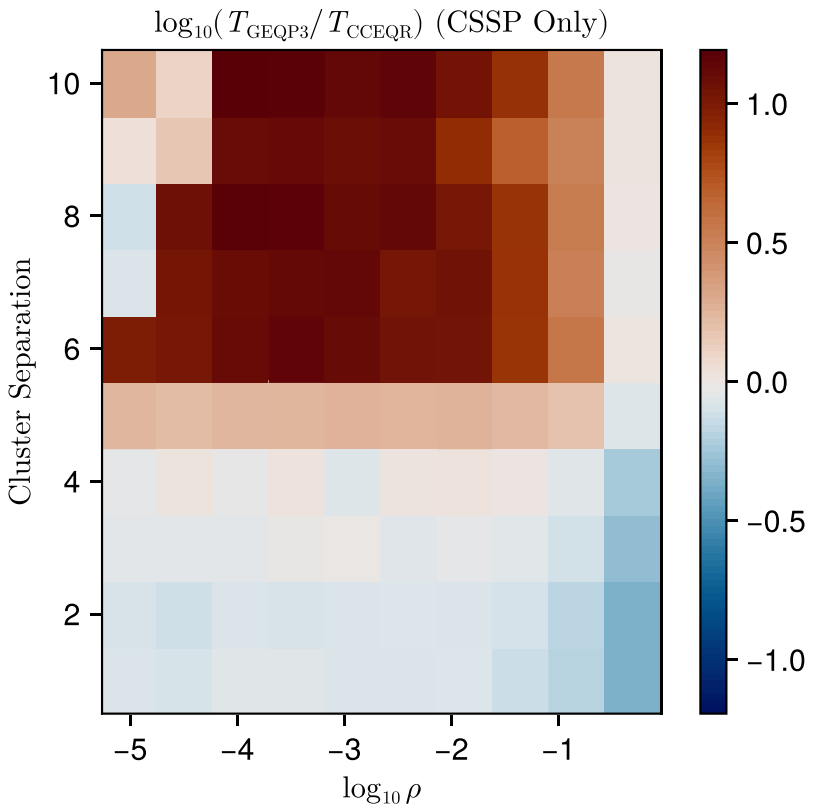
Experiment 1: Spectral Clustering

- We draw n i.i.d. samples from a Gaussian mixture model with $k = 20$ components.
- Kernel matrix: $K(i,j) = \exp\left(-\frac{1}{2\sigma^2}\left\|x_i - x_j\right\|_2^2\right) = VDV^T$.
- **Laplacian embedding:** $Z(:,j) = V(j,1:k)^T$.
- Running QRCP on Z selects one point from each cluster [4].
- Column norms in Z measure centrality in clusters [8].

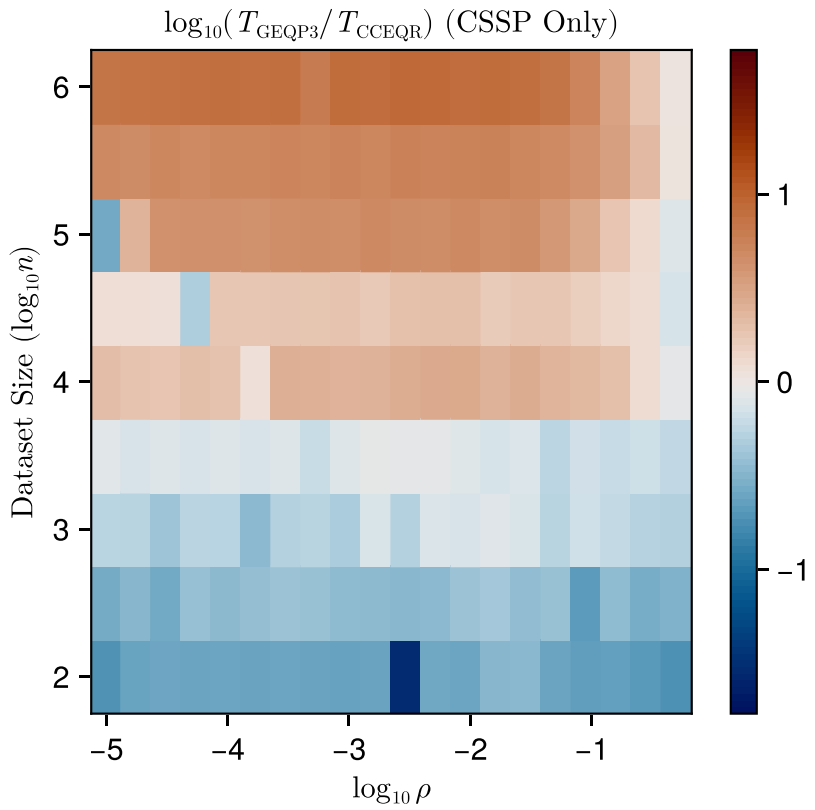


[4] Damle, Minden, and Ying, Information and Inference, 2018.
[8] G. Scheibinger, M. J. Wainwright, and B. Yu, The Annals of Statistics, 2015

Fixed n, increasing cluster separation.

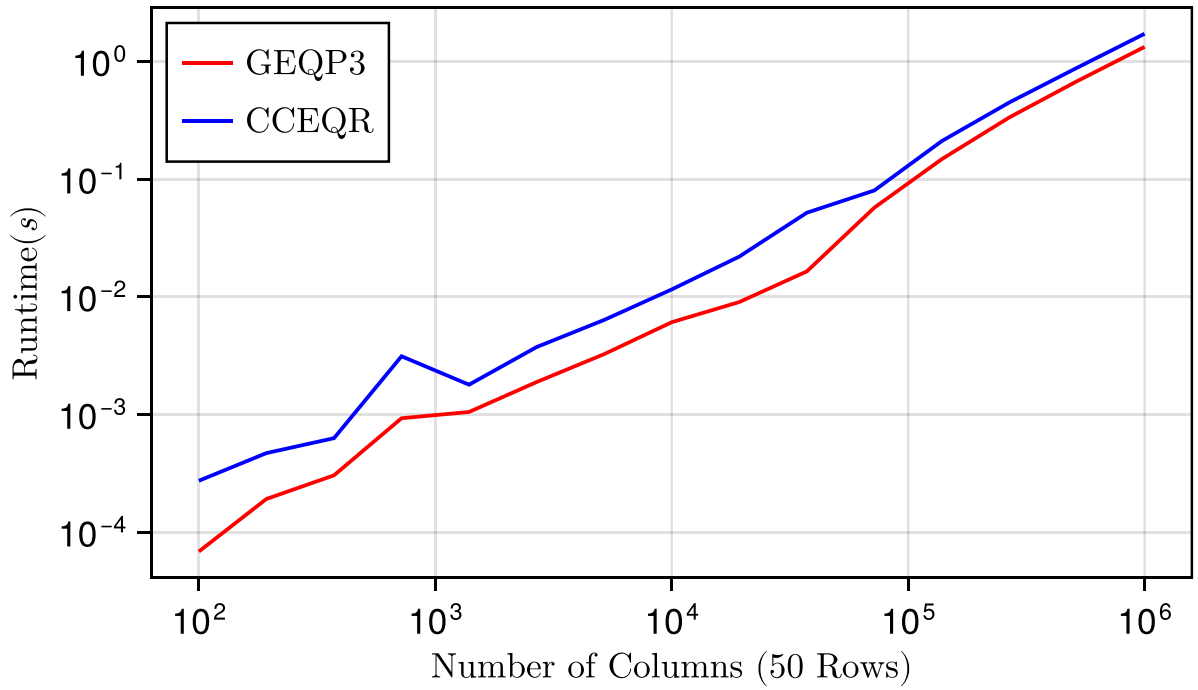


Increasing n, fixed cluster separation.



Experiment 2: Random Gaussians

- Our algorithm is fast when the distribution of column norm mass is concentrated.
- **What about for unstructured problems?**
- Unstructured test matrices: random Gaussians.
- **Essentially the same runtime as LAPACK**.



Thank you!

References

1. S. Chaturantabut and D. C. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2737 – 2764.
2. A. Civril and M. Magdon Ismail, *On selecting a maximum volume sub-matrix of a matrix and related problems,* Theoretical Computer Science, 410 (2009) pp. 4801 – 4811.
3. A. Damle, L. Lin, and L. Ying, *Compressed representation of Kohn-Sham orbitals via selected columns of the density matrix,* J. Chem. Theory Comput., 14 (2015), pp. 1463 – 1469.
4. A. Damle, V. Minden, and L. Ying, *Simple, direct, and efficient multi-way spectral clustering,* Information and Inference: A Journal of the IMA, 8 (2018), pp. 181 – 203.
5. P.G. Martinsson, G. Quintana-Ortí, N. Heavner, and R. van de Geijn, *Householder QR factorization with randomization for column pivoting (HQRRP)*, SIAM Journal on Scientific Computing, 39 (2017), pp. C96 – C115.
6. G. Quintana-Ortí, X. Sun, and C. H. Bishof, *A BLAS-3 version of the QR factorization with column pivoting*, SIAM Journal on Scientific Computing, 19 (1998), pp. 1486 – 1494.
7. F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert, *A fast randomized algorithm for the approximation of matrices*, Applied and Computational Harmonic Analysis, 25 (2008), pp. 395 – 416.
8. G. Scheibinger, M. J. Wainwright, and B. Yu, *The geometry of spectral clustering,* The Annals of Statistics, 43 (2015), pp. 819 – 846.