Improved Algorithms for Low Rank Approximation from Sparsity

Taisuke Yasuda

based on work with

David P. Woodruff

Carnegie Mellon University

Low Rank Approximation

Problem Setting

- Input: $n \times n$ symmetric matrix $\mathbf{A}$, rank parameter k

Assume symmetric input for simplicity; ideas apply to rectangular matrices.

- Goal: find a rank k matrix $\hat{\mathbf{A}}$ that approximates $\mathbf{A}$

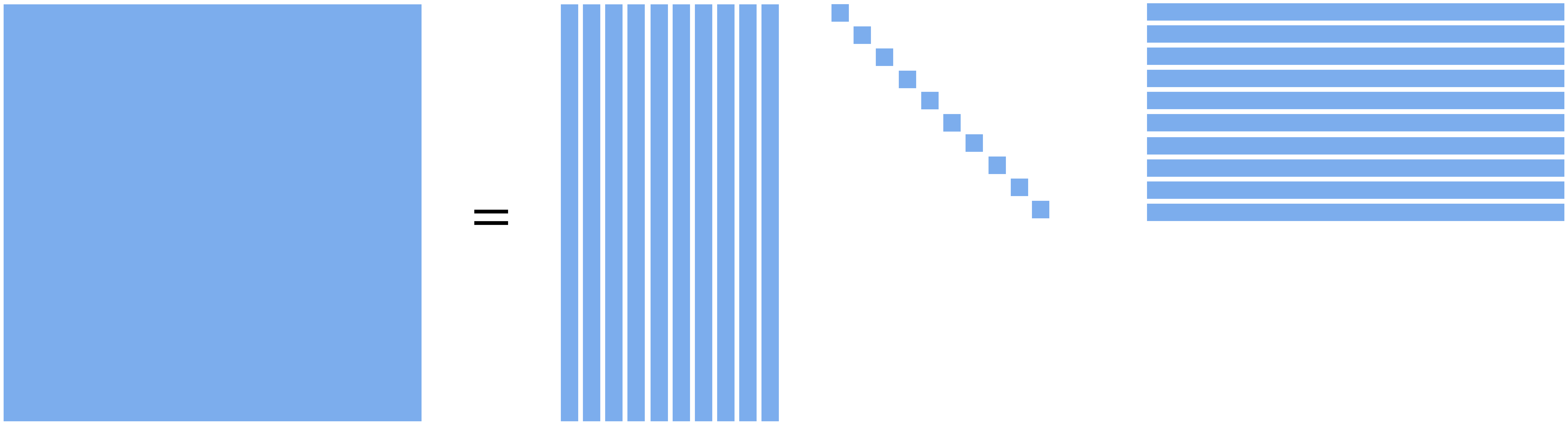
$$\mathbf{A} \approx \hat{\mathbf{A}} \text{ (rank } k\text{)}$$

$$\hat{\mathbf{A}} := \arg \min_{\text{rank } k\ \mathbf{X}} \| \mathbf{A} - \mathbf{X} \|$$

- Frobenius norm
- Spectral norm
- Entrywise ℓ_p norm
- …

Low Rank Approximation

Singular Value Decomposition (SVD)

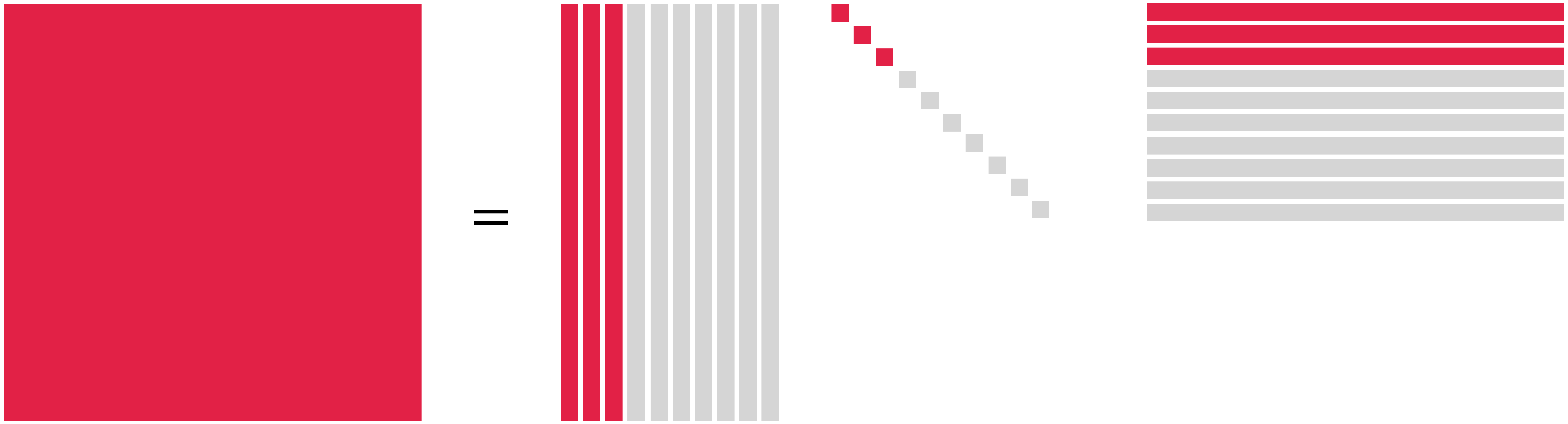


$$\mathbf{A} = \mathbf{U} \quad \boldsymbol{\Sigma} \quad \mathbf{U}^\top$$

Singular vectors   Singular values

Low Rank Approximation

Singular Value Decomposition (SVD)



$$\mathbf{A}_k = \mathbf{U}_k \quad \boldsymbol{\Sigma}_k \quad \mathbf{U}_k^\top$$

Eckart—Young—Mirsky: this is *optimal* for
Frobenius/spectral norm low rank approximation

Low Rank Approximation

Algorithms

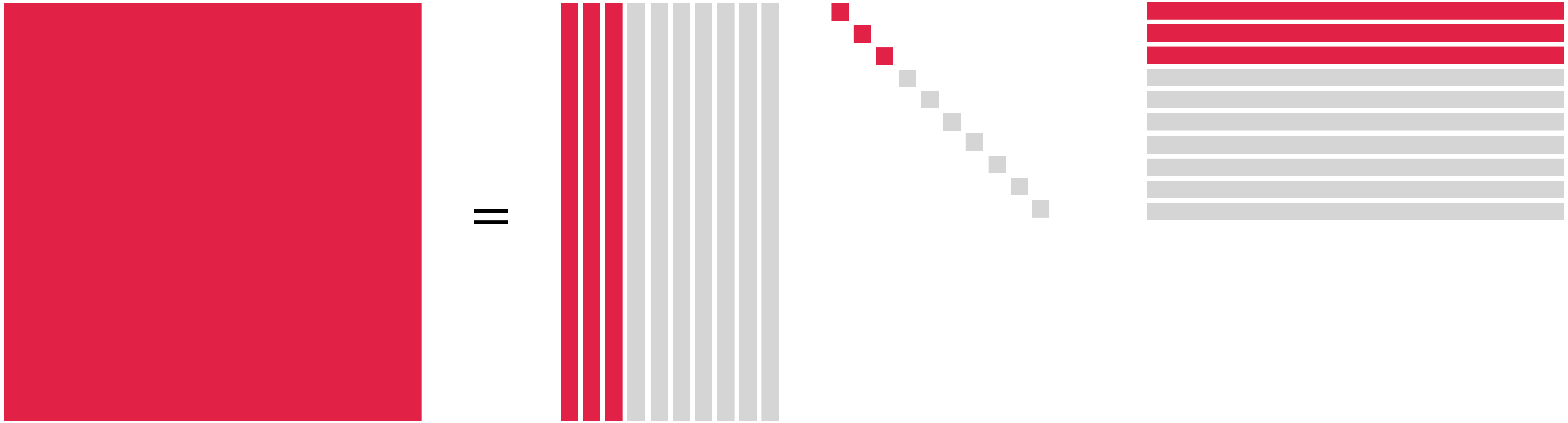
- Exact SVD can be computed in $O(n^{\omega})$ time [DDH07, NN13]
- Approximate SVD can be computed in $O\left(\frac{\mathrm{nnz}(\mathbf{A})k}{\sqrt{\epsilon}}\right)$ time [MM15]
    - Approximate: $\left\| \mathbf{A} - \hat{\mathbf{A}} \right\| \leq (1+\epsilon) \left\| \mathbf{A} - \mathbf{A}_k \right\| = (1+\epsilon)\sigma_{k+1}$

What if the top k singular vectors are sparse?

Note: log factors in n will be ignored throughout this talk

Low Rank Approximation

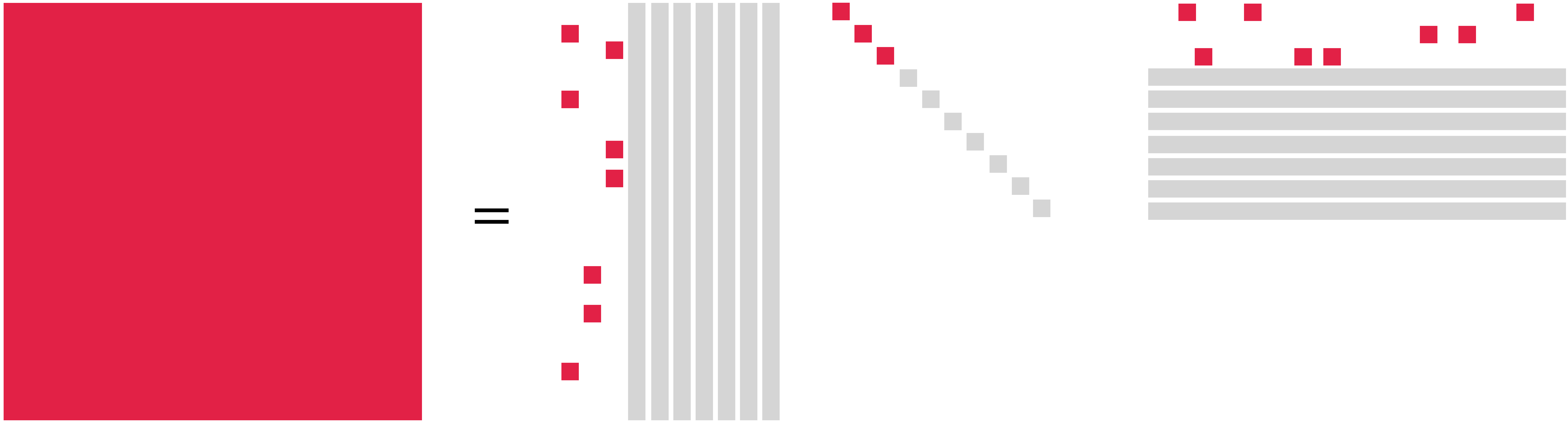
Sparse Singular Vectors



$$\mathbf{A}_k = \mathbf{U}_k \quad \mathbf{\Sigma}_k \quad \mathbf{U}_k^\top$$

Low Rank Approximation

Sparse Singular Vectors



$$\mathbf{A}_k = \mathbf{U}_k \quad \boldsymbol{\Sigma}_k \quad \mathbf{U}_k^\top$$

s-sparse vector $\mathbf{u}$: $\mathbf{u}$ has at most s nonzero entries

Our Results

Sparse Singular Vectors

Theorem. Suppose the top k singular vectors of $\mathbf{A}$ are s-sparse. Then, an approximate *spectral* low rank approximation can be computed in time

$$O\left(\frac{\mathsf{nnz}(\mathbf{A})}{\sqrt{\epsilon}} + \mathrm{poly}(s, k, \epsilon^{-1})\right)$$

If the top k singular vectors are sparse,
then we save a factor of k.

Low Rank Approximation

Sketching Model

- Observe $\mathbf{A} \in \mathbb{R}^{n \times n}$ through t linear measurements
  - Choose query matrices $\mathbf{S}_i \in \mathbb{R}^{n \times n}$ for $i \in [t]$ at random
  - Observe $\langle \mathbf{S}_1, \mathbf{A} \rangle, \langle \mathbf{S}_2, \mathbf{A} \rangle, \ldots, \langle \mathbf{S}_t, \mathbf{A} \rangle$
  - Output computed as a function of $\langle \mathbf{S}_1, \mathbf{A} \rangle, \langle \mathbf{S}_2, \mathbf{A} \rangle, \ldots, \langle \mathbf{S}_t, \mathbf{A} \rangle$
  - Complexity measure: number of measurements t
- For low rank approximation in the *Frobenius* norm, $t = \Theta(nk/\epsilon)$ measurements is necessary [CW09] and sufficient [BWZ16]

$$\hat{\mathbf{A}} := \arg \min_{\text{rank } k \ \mathbf{X}} \| \mathbf{A} - \mathbf{X} \|_{\boxed{F}}$$

Our Results

Sparse Low Rank Approximation in the Sketching Model

$$\hat{\mathbf{A}} := \arg \min_{\text{rank } k,\ \boxed{s \times s}\ \mathbf{X}} \| \mathbf{A} - \mathbf{X} \|_F$$

Theorem. The approximate sparse low rank approximation problem can be solved in t measurements to $\mathbf{A}$ in the sketching model, where:

- $t = O(sk/\epsilon^2)$
  - Exponential time algorithm with an $s \times s$ $\hat{\mathbf{A}}$
- $t = O(s^2k^2/\text{poly}(\epsilon))$
  - Polynomial time algorithm with an $O(sk/\epsilon) \times O(sk/\epsilon)$ $\hat{\mathbf{A}}$
- $t = O(sk^2/\text{poly}(\epsilon))$
  - Polynomial time algorithm with an $O(sk/\epsilon) \times O(sk/\epsilon)$ $\hat{\mathbf{A}}$ with additive error

Our Results

Sparse Singular Vectors

Theorem. Suppose the top k singular vectors of $\mathbf{A}$ are s-sparse. Then, an approximate spectral low rank approximation can be computed in time

$$O\left(\frac{\mathsf{nnz}(\mathbf{A})}{\sqrt{\epsilon}} + \mathrm{poly}(s, k, \epsilon^{-1})\right)$$

If the top k singular vectors are sparse,
then we save a factor of k.

Our Results

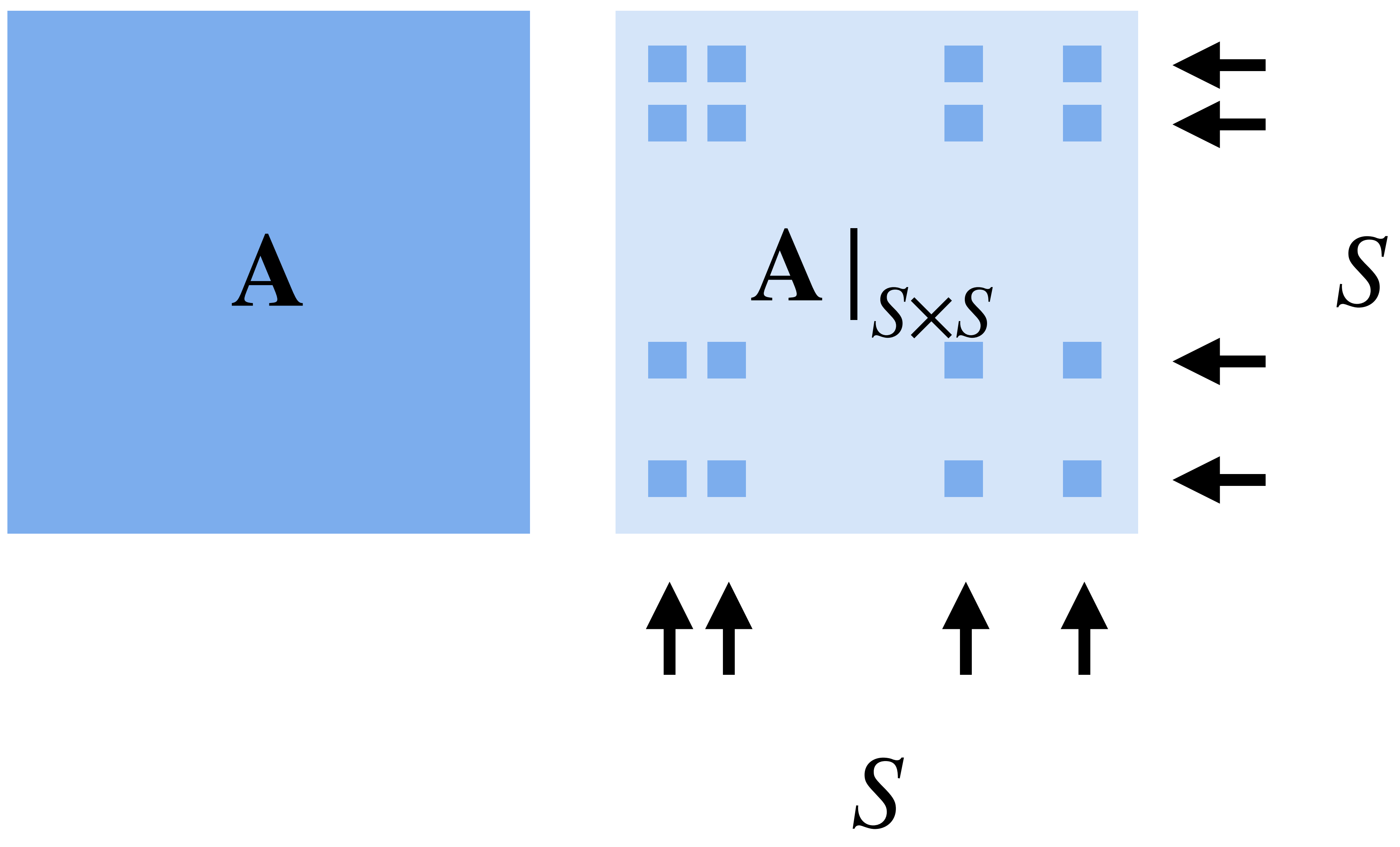
Proof Overview

- Reduction to finding the support
- $O(\mathsf{nnz}(\mathbf{A})/\epsilon)$ time algorithm
- $O(\mathsf{nnz}(\mathbf{A})/\sqrt{\epsilon})$ time algorithm
    - NOT just Chebyshev polynomials!

Proof Sketch

Reduction to Finding the Support

- The top k singular vectors are supported on at most sk coordinates
- If we can find small set $S \subset [n]$ containing the supports of the top k singular vectors, then we can just restrict to these coordinates



$$\|\mathbf{A}\mathbf{u}_i\|_2 = \|\mathbf{A}|_{S\times S}\mathbf{u}_i|_S\|_2 \qquad \text{for all } i \in [k]$$

$$\|\mathbf{A}\mathbf{x}\|_2 \geq \|\mathbf{A}|_{S\times S}\mathbf{x}|_S\|_2 \qquad \text{for all } \mathbf{x} \in \mathbb{R}^n$$

Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\epsilon)$ Time Algorithm

Idea: Power Method

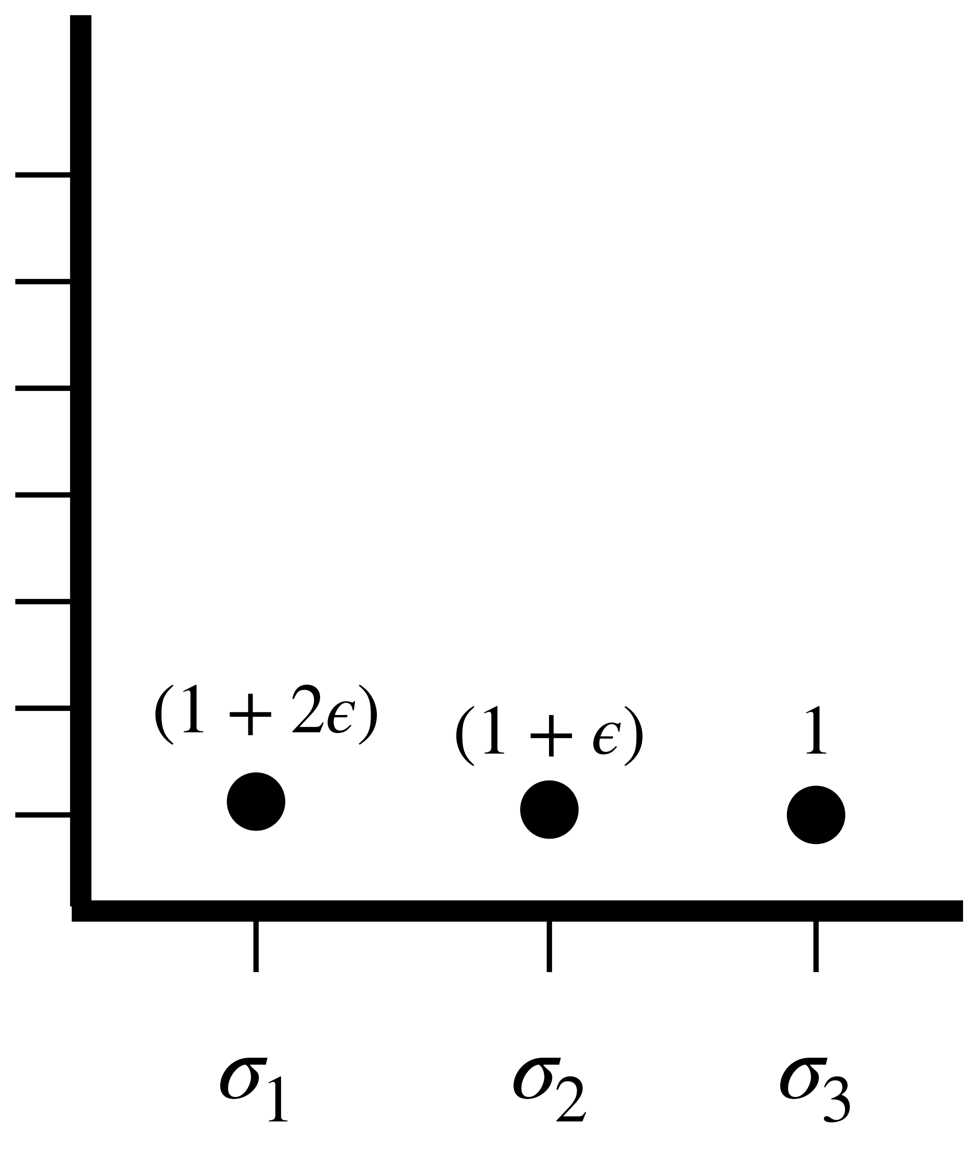
```
def find_support(A, s, k, ε):
    q = Θ(1/ε)
    x ← g ~ N(0, I_n)
    for t in [q]:
        x ← Ax                                  # computes A^q g
    return {i ∈ [n] : |x_i| is top O(sk)}
```

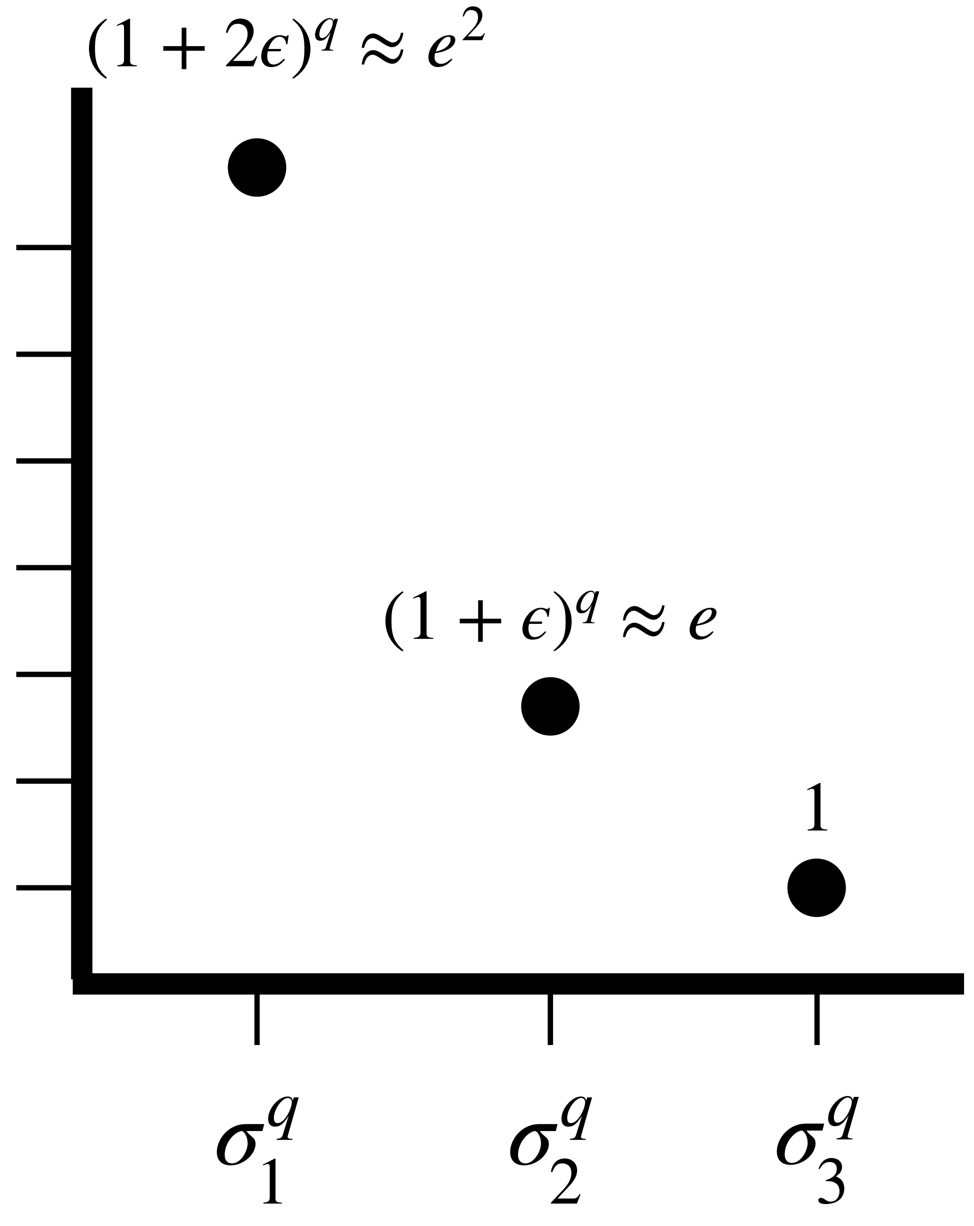
Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\epsilon)$ Time Algorithm

Idea: Power Method

$$\mathbf{A}^q = (\mathbf{U\Sigma U}^\top)^q = \mathbf{U\Sigma}^q\mathbf{U}^\top$$





Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\epsilon)$ Time Algorithm

Idea: Power Method

$$\mathbf{A}^q = (\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^\top)^q = \mathbf{U}\boldsymbol{\Sigma}^q\mathbf{U}^\top$$

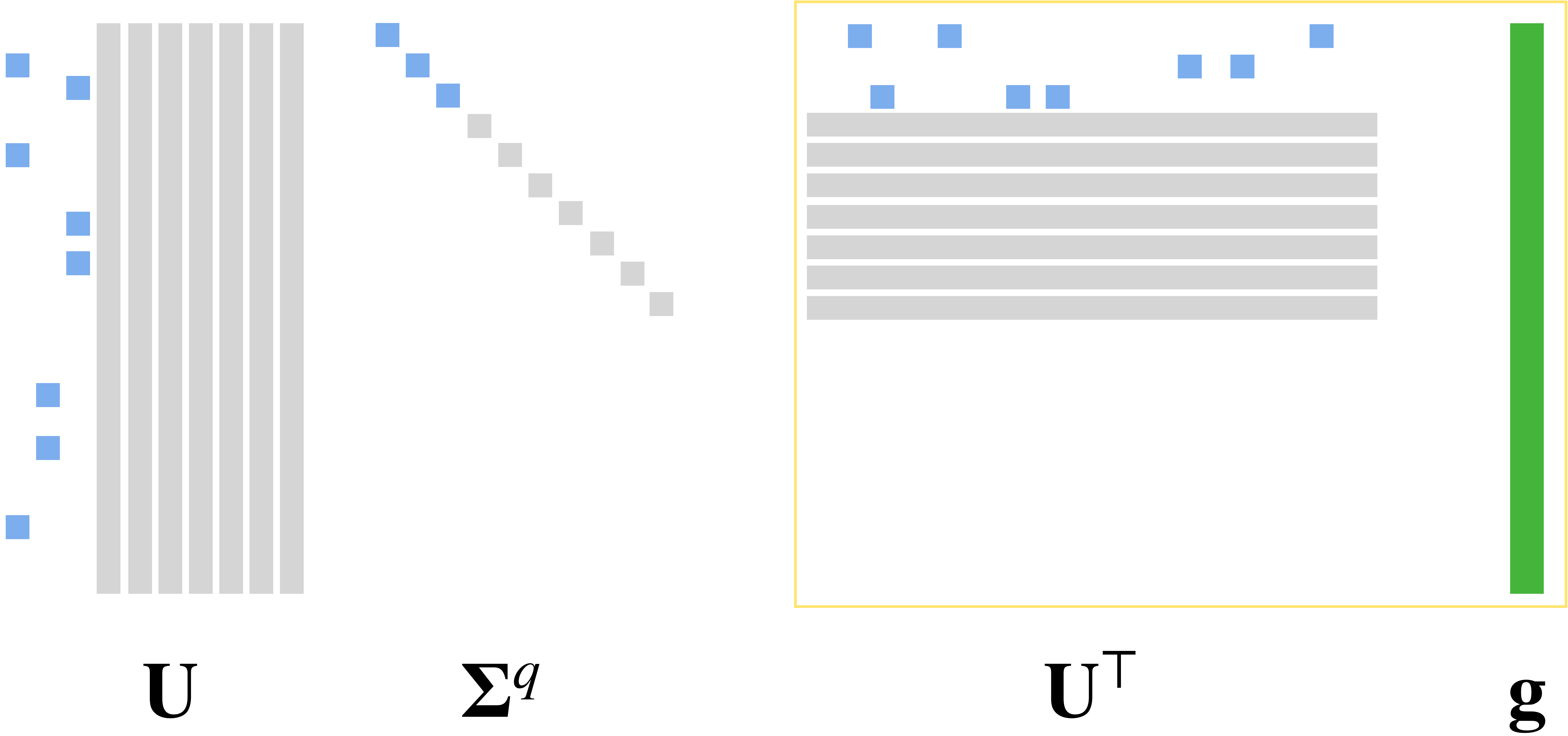
$\mathbf{A}^q$ $\quad$ $\mathbf{g} \sim \mathcal{N}(0, \mathbf{I}_n)$

Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\epsilon)$ Time Algorithm

Idea: Power Method

$$\mathbf{A}^q \quad = (\mathbf{U\Sigma U}^\top)^q \quad = \mathbf{U\Sigma}^q\mathbf{U}^\top$$

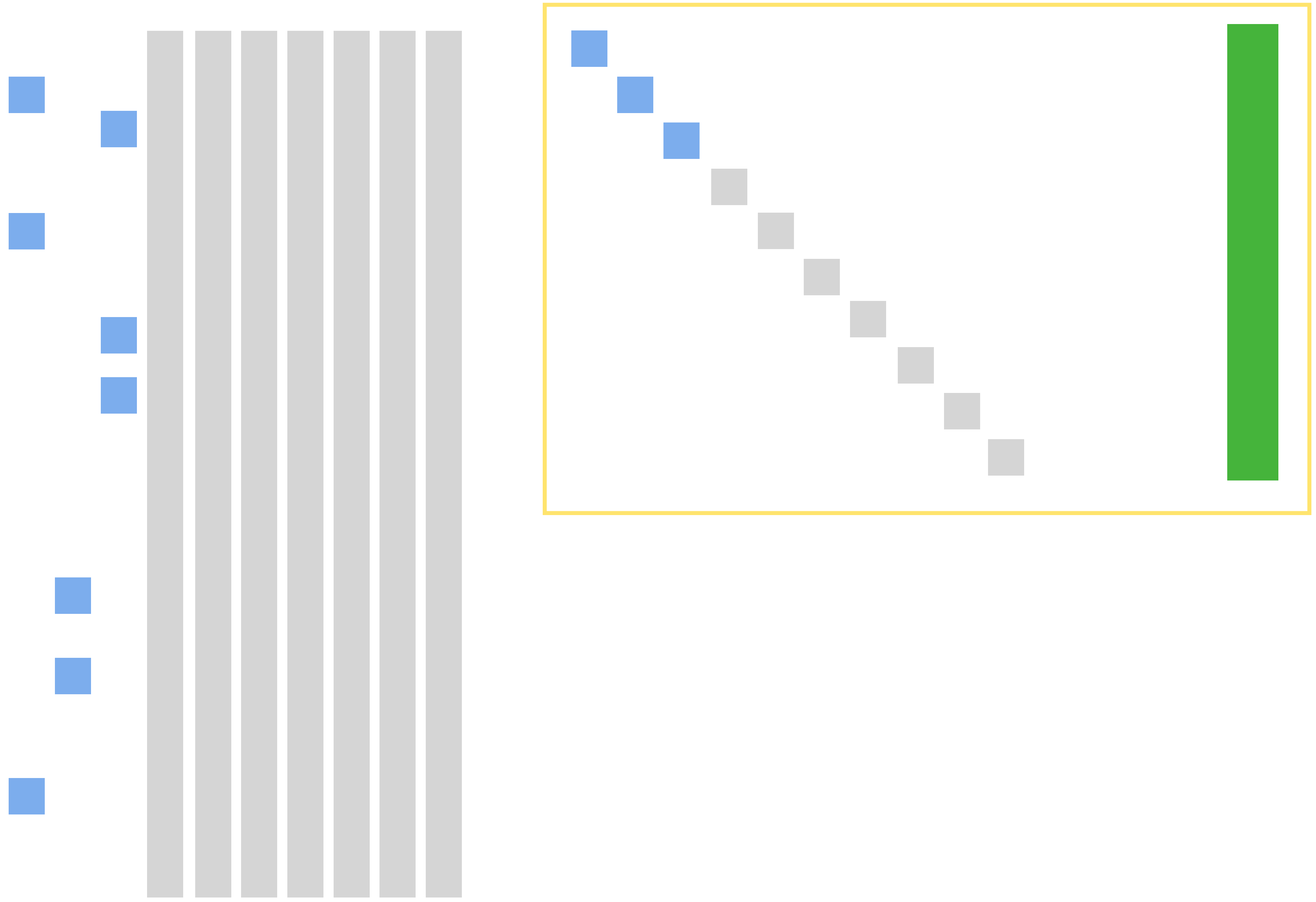


Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\epsilon)$ Time Algorithm

Idea: Power Method

$$\mathbf{A}^q \quad = (\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^\top)^q \quad = \mathbf{U}\boldsymbol{\Sigma}^q\mathbf{U}^\top$$



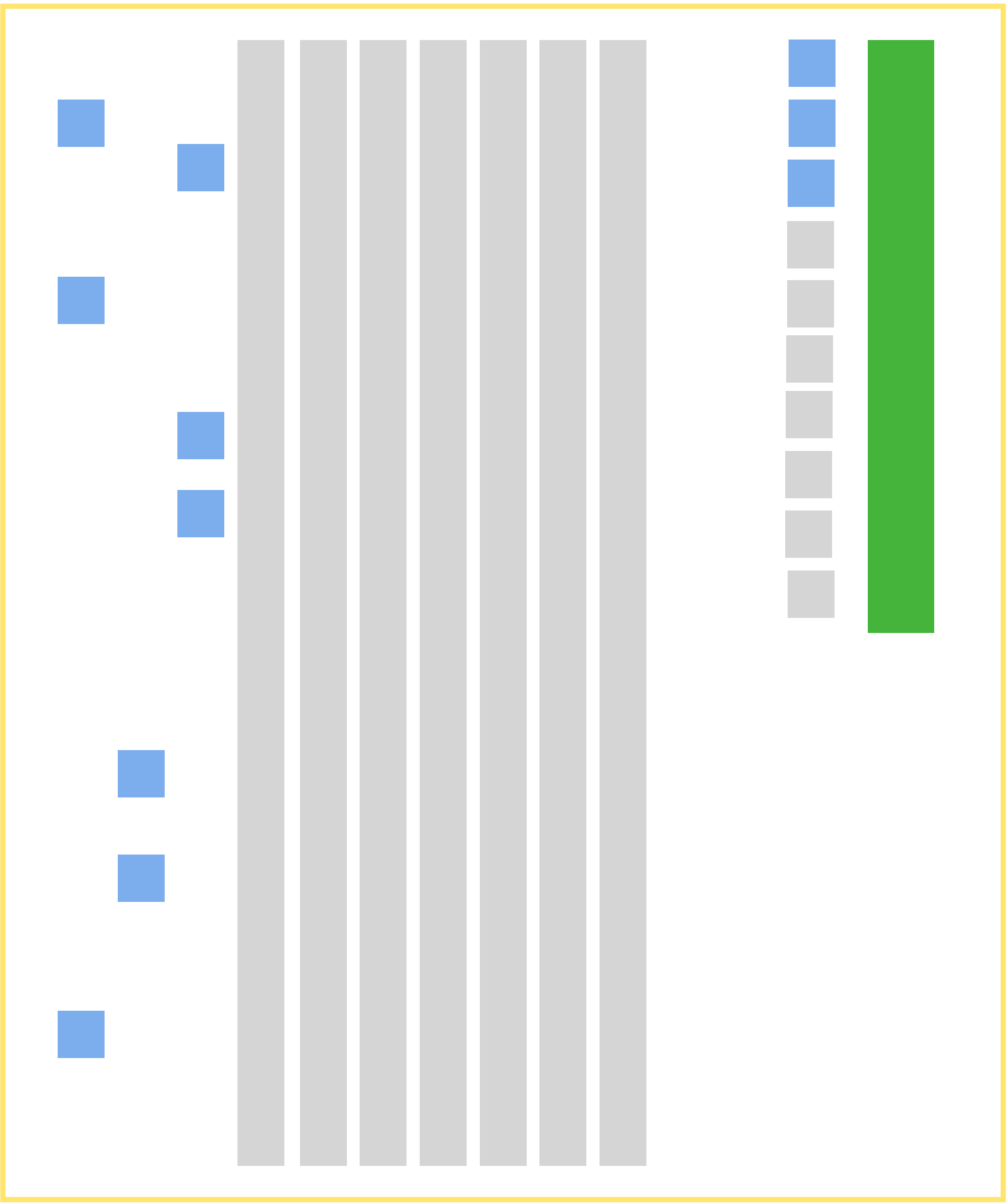
$\mathbf{U}$ $\quad$ $\boldsymbol{\Sigma}^q$ $\quad$ $\mathbf{g}' = \mathbf{U}^\top \mathbf{g} \sim \mathcal{N}(0, \mathbf{I}_r)$

Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\epsilon)$ Time Algorithm

Idea: Power Method

$$\mathbf{A}^q \quad = (\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^\top)^q \quad = \mathbf{U}\boldsymbol{\Sigma}^q\mathbf{U}^\top$$



$\mathbf{U}$ $\boldsymbol{\Sigma}^q\mathbf{g}'$

Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\epsilon)$ Time Algorithm

Idea: Power Method

$$\mathbf{A}^q \quad = (\mathbf{U\Sigma U}^\top)^q \quad = \mathbf{U\Sigma}^q\mathbf{U}^\top$$

$\mathbf{U\Sigma}^q\mathbf{g}'$

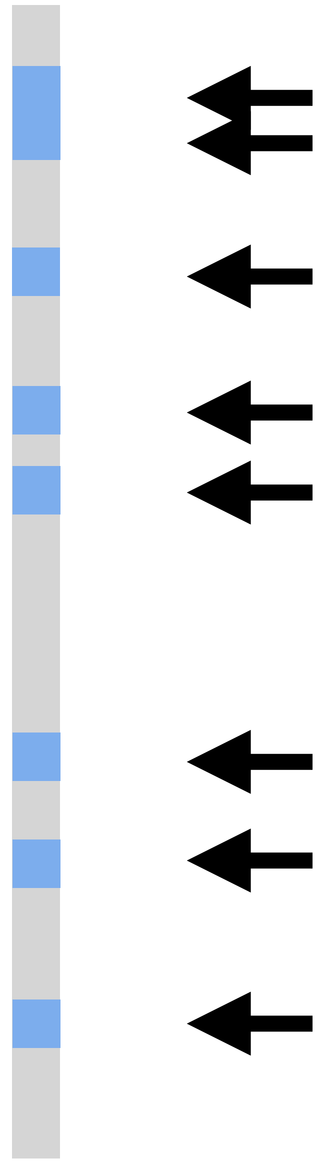
Linear combination of the singular vectors,
where the ith singular vector is scaled by ~σ_i^q

Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\epsilon)$ Time Algorithm

Idea: Power Method

$$\mathbf{A}^q = (\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^\top)^q = \mathbf{U}\boldsymbol{\Sigma}^q\mathbf{U}^\top$$



$S \qquad |S| = O(sk)$

$\mathbf{U}\boldsymbol{\Sigma}^q\mathbf{g}'$

Proof Sketch

$O(\text{nnz}(\mathbf{A})/\sqrt{\epsilon})$ Time Algorithm

- Standard method for replacing ϵ by $\sqrt{\epsilon}$: Chebyshev polynomials
- Replace $\mathbf{A}^q$ with $T(\mathbf{A})$, where T is a degree ~$\sqrt{q}$ polynomial
- T can be chosen so that:
  - $T(x) \leq 1$ for $x \leq \sigma_{k+1}$
  - $T(x) \geq e$ for $x \geq (1+\epsilon)\sigma_{k+1}$
- Problem: T depends on σ_{k+1}!

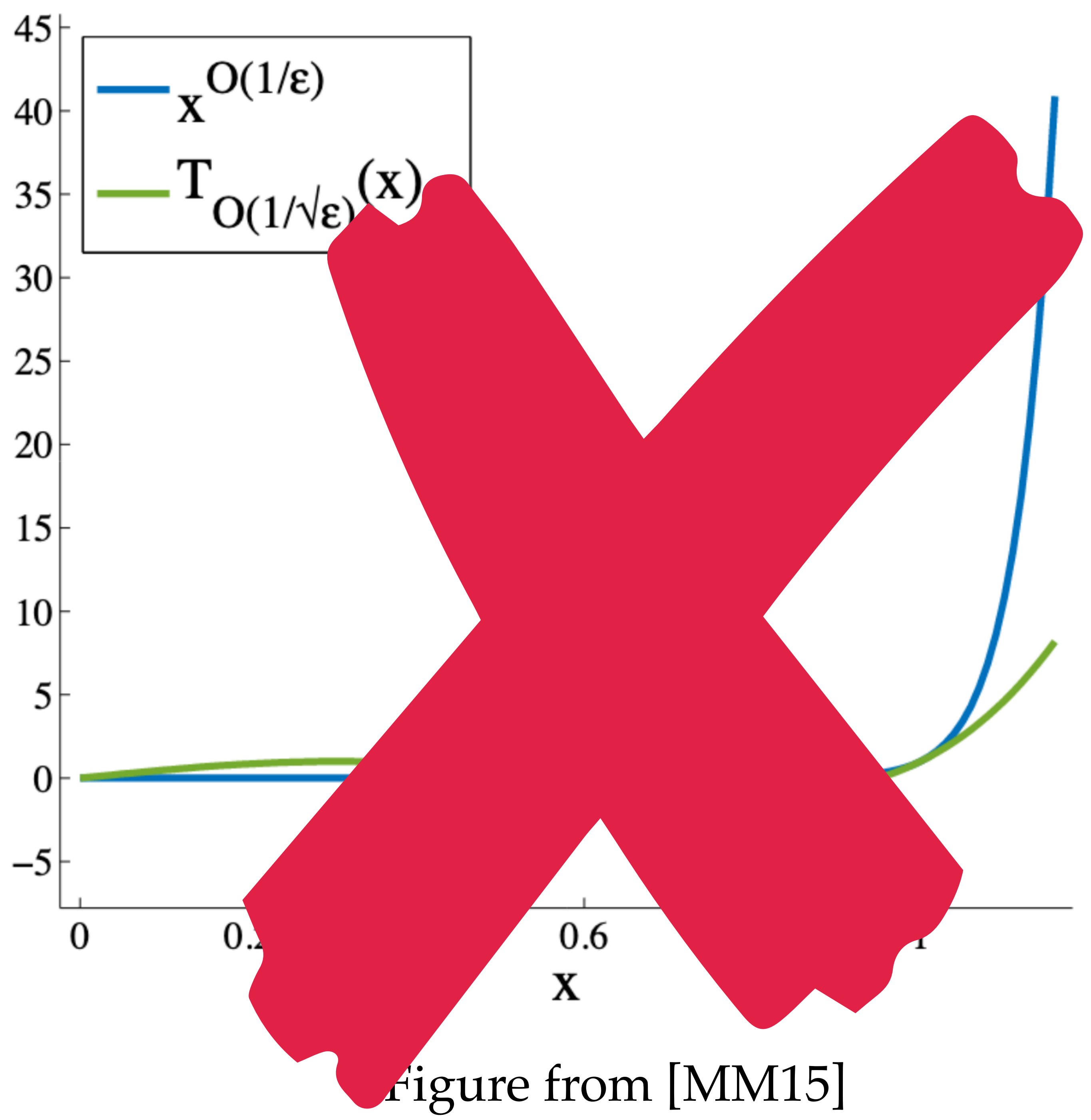


Figure from [MM15]

Proof Sketch

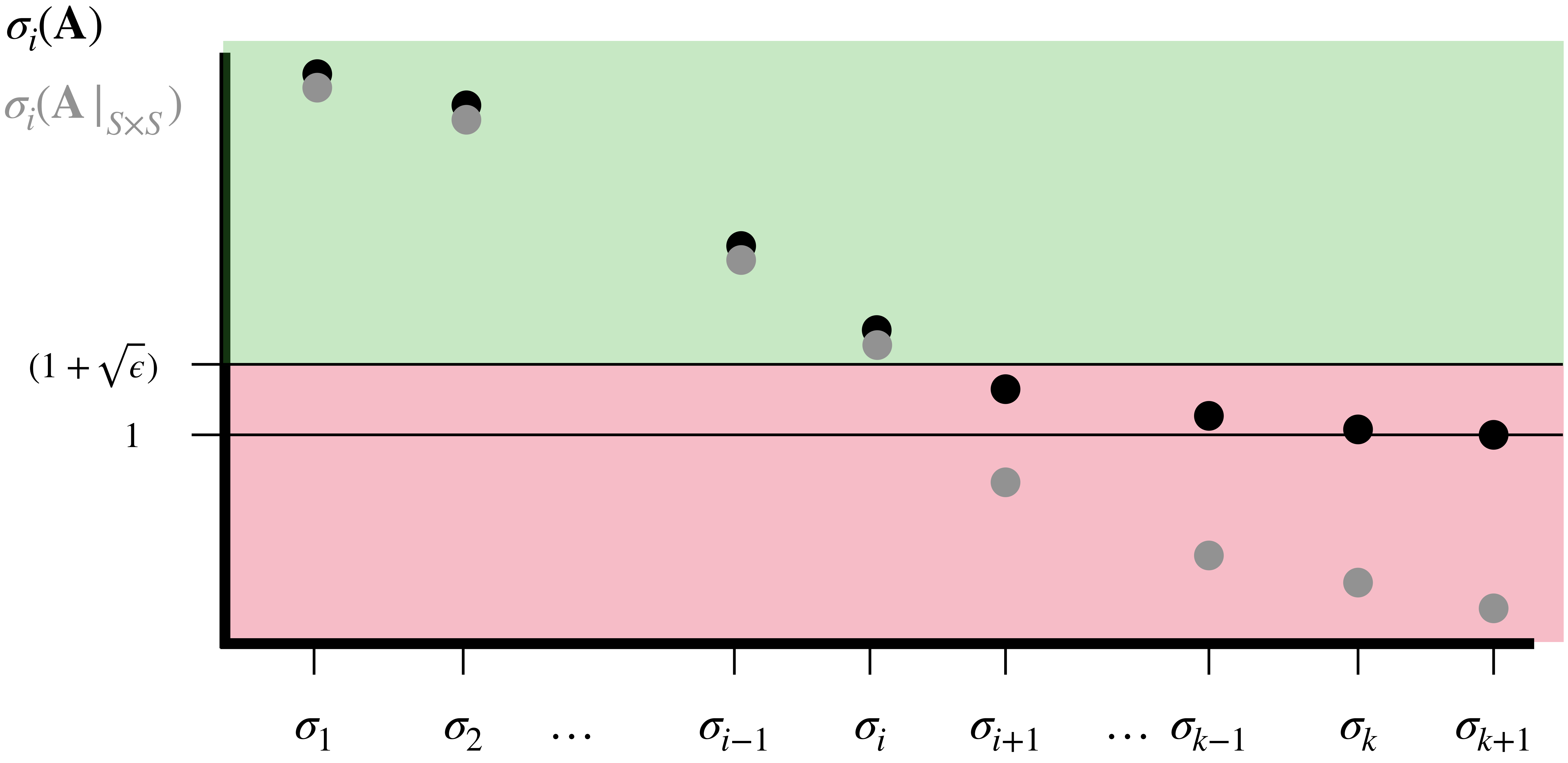
$O(\mathsf{nnz}(\mathbf{A})/\sqrt{\epsilon})$ Time Algorithm

- Run the previous algorithm with ϵ set to $\sqrt{\epsilon}$ to get a support superset S
- Obtain a rank k approximation $\hat{\mathbf{A}}|_{S\times S}$ to $\mathbf{A}|_{S\times S}$
- Obtain a $(1+\sqrt{\epsilon})$-approximation of σ_{k+1} using $\hat{\mathbf{A}}|_{S\times S}$
- Enumerate over $1/\sqrt{\epsilon}$ guesses to σ_{k+1} in powers of $(1+\epsilon)$

Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\sqrt{\epsilon})$ Time Algorithm

Idea: find singular value upper bounds that match lower bounds!



Support of singular vectors recovered; singular values of $\mathbf{A}|_{S\times S}$ approximate those in $\mathbf{A}$ up to $(1+\epsilon)$ factors

Singular values of $\mathbf{A}|_{S\times S}$ may be significantly smaller than those in $\mathbf{A}$

Cauchy Interlacing Theorem: $\sigma_i(\mathbf{A}_{S\times S}) \leq \sigma_i(\mathbf{A})$

Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\sqrt{\epsilon})$ Time Algorithm

- Consider $[\mathbf{A}|_{S\times S}]_i$ (the best rank i approximation to $\mathbf{A}|_{S\times S}$)

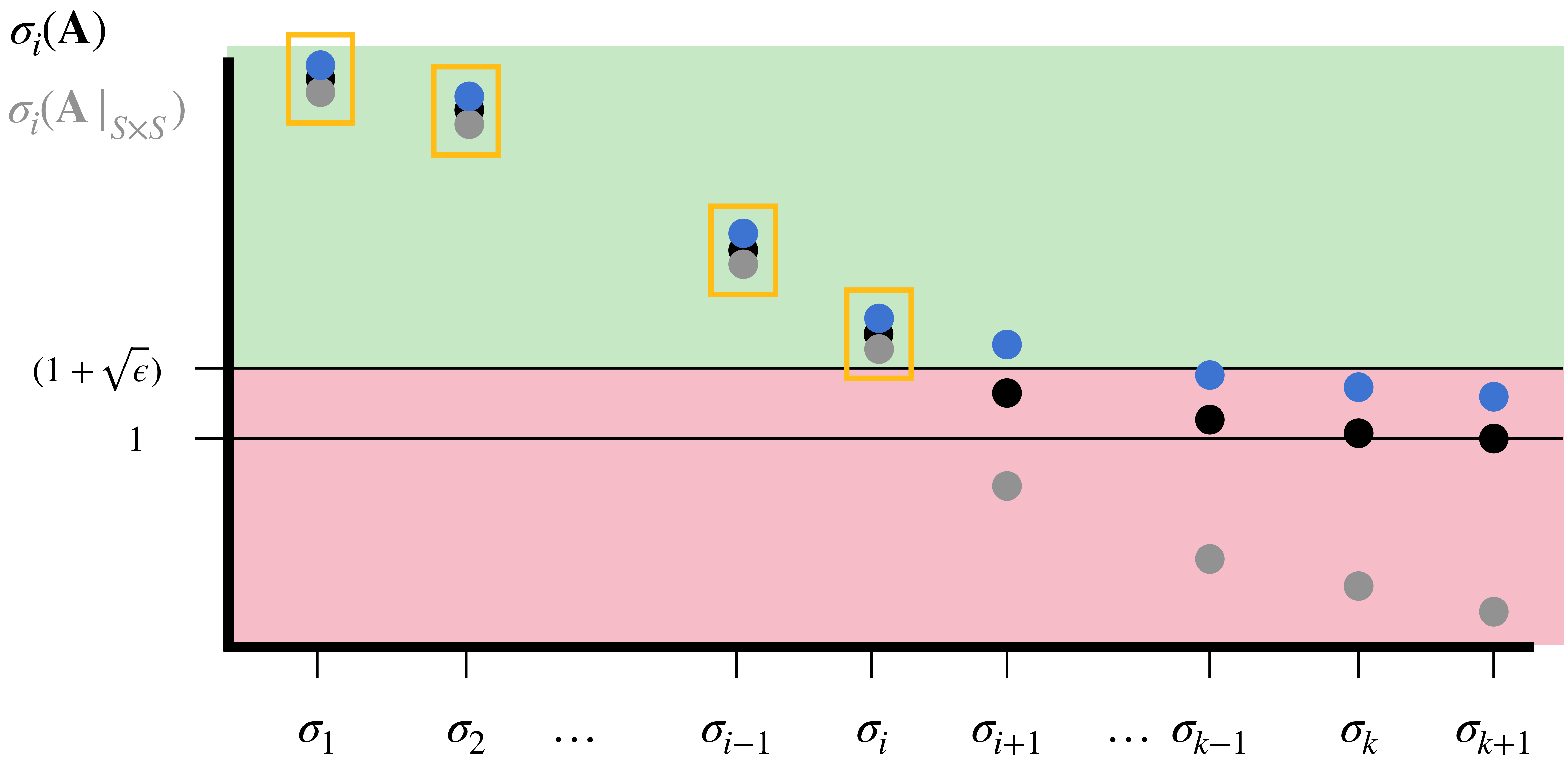
$$\left\| \mathbf{A} - [\mathbf{A}|_{S\times S}]_i \right\| \geq \min_{\text{rank } i \ \mathbf{X}} \left\| \mathbf{A} - \mathbf{X} \right\|$$

$$= \sigma_{i+1}(\mathbf{A})$$

Proof Sketch

$O(\mathrm{nnz}(\mathbf{A})/\sqrt{\epsilon})$ Time Algorithm

$$\left\| \mathbf{A} - [\mathbf{A}|_{S\times S}]_{i-1} \right\|$$



Support of singular vectors recovered; singular values of $\mathbf{A}|_{S\times S}$ approximate those in $\mathbf{A}$ up to $(1+\epsilon)$ factors

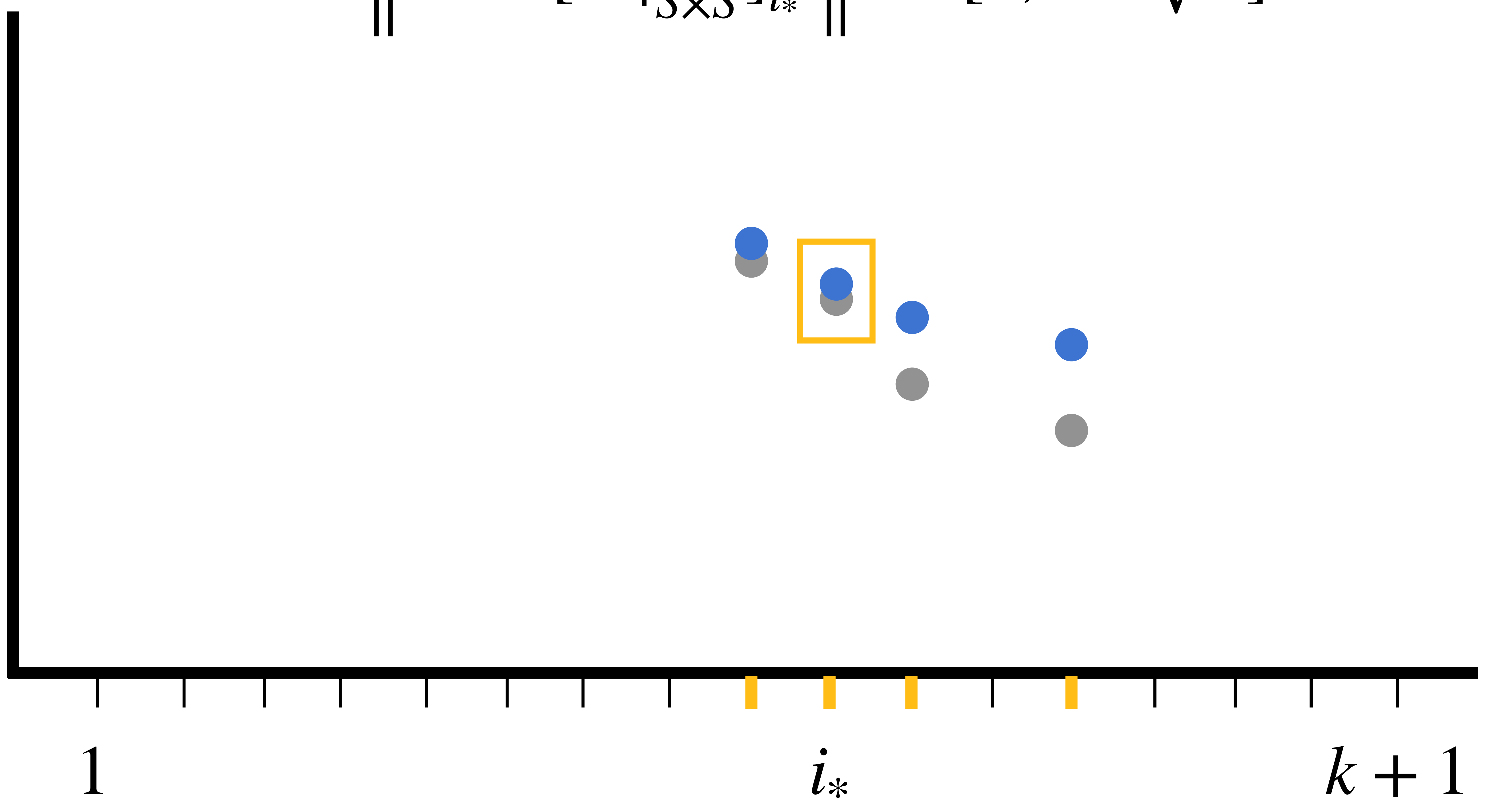
Singular values of $\mathbf{A}|_{S\times S}$ may be significantly smaller than those in $\mathbf{A}$

Cauchy Interlacing Theorem: $\sigma_i(\mathbf{A}_{S\times S}) \leq \sigma_i(\mathbf{A})$

Proof Sketch

$O(\mathsf{nnz}(\mathbf{A})/\sqrt{\epsilon})$ Time Algorithm

$$\left\| \mathbf{A} - [\mathbf{A}\,|_{S\times S}]_{i_*} \right\| \in [1, 1+\sqrt{\epsilon}]$$



$\left\| \mathbf{A} - [\mathbf{A}\,|_{S\times S}]_{i-1} \right\|$

$\sigma_i(\mathbf{A}\,|_{S\times S})$

Conclusion

- In this work: an $O\left(\frac{\mathsf{nnz}(\mathbf{A})}{\sqrt{\epsilon}}\right)$ time algorithm for spectral low rank approximation for matrices with sparse singular vectors, breaking the $O\left(\frac{\mathsf{nnz}(\mathbf{A})k}{\sqrt{\epsilon}}\right)$ barrier
- Techniques: power method, Chebyshev polynomials, and binary search via efficient singular value estimate certificates
- Open directions: designing a robust version of this algorithm (e.g. in finite precision)
- Other results: improved space complexity for streaming algorithms for related sparse low rank approximation problems

References

- [BWZ16] Christos Boutsidis, David P. Woodruff, and Peilin Zhong. Optimal principal component analysisin distributed and streaming models. In STOC, pages 236–249. ACM, 2016.
- [CW09] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In STOC, pages 205–214. ACM, 2009
- [DDH07] James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. Numer. Math., 108(1):59–91, October 2007.
- [NN13] Jelani Nelson and Huy L Nguyên. OSNAP: Faster numerical linearalgebra algorithms via sparser subspace embeddings. FOCS, 2013.
- [MM15] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger andfaster approximate singular value decomposition. In NIPS, pages 1396–1404, 2015.