SUPPLEMENTARY MATERIALS: STREAMING LOW-RANK MATRIX APPROXIMATION WITH AN APPLICATION TO SCIENTIFIC SIMULATION*

JOEL A. TROPP†, ALP YURTSEVER‡, MADELEINE UDELL§, AND VOLKAN CEVHER‡

Abstract. This paper argues that randomized linear sketching is a natural tool for on-the-fly compression of data matrices that arise from large-scale scientific simulations and data collection. The technical contribution consists in a new algorithm for constructing an accurate low-rank approximation of a matrix from streaming data. This method is accompanied by an a priori analysis that allows the user to set algorithm parameters with confidence and an a posteriori error estimator that allows the user to validate the quality of the reconstructed matrix. In comparison to previous techniques, the new method achieves smaller relative approximation errors and is less sensitive to parameter choices. As concrete applications, the paper outlines how the algorithm can be used to compress a Navier–Stokes simulation and a sea surface temperature dataset.

Key words. dimension reduction, matrix approximation, numerical linear algebra, sketching, streaming, singular-value decomposition

AMS subject classifications. 65F30, 68W20

DOI. 10.1137/18M1201068

SM1. Analysis of the Low-Rank Approximation. This section contains the proof of Theorem 5.1, the theoretical result on the behavior of the basic low-rank approximation (2.9). We maintain the notation from section 2.

SM1.1. Facts about Random Matrices. First, let us state a useful formula that allows us to compute some expectations involving a Gaussian random matrix. This identity is drawn from [SM6, Prop. A.1 and A.6]. See also [SM8, Fact A.1].

FACT SM1.1. *Assume that $t > q + \alpha$. Let $\boldsymbol{G}_1 \in \mathbb{F}^{t\times q}$ and $\boldsymbol{G}_2 \in \mathbb{F}^{t\times p}$ be independent standard normal matrices. For any matrix $\boldsymbol{B}$ with conforming dimensions,*

$$\mathbb{E}\,\|\boldsymbol{G}_1^\dagger \boldsymbol{G}_2 \boldsymbol{B}\|_2^2 = \frac{q}{t-q-\alpha}\cdot\|\boldsymbol{B}\|_2^2.$$

The number $\alpha = 1$ when $\mathbb{F} = \mathbb{R}$, while $\alpha = 0$ when $\mathbb{F} = \mathbb{C}$.

SM1.2. Results from Randomized Linear Algebra. Our argument also depends on the analysis of randomized low-rank approximation developed in [SM6, Sec. 10].

FACT SM1.2 (Halko et al. 2011). *Fix $\boldsymbol{A} \in \mathbb{F}^{m\times n}$. Let ϱ be a natural number such that $\varrho < k - \alpha$. Draw the random test matrix $\boldsymbol{\Omega} \in \mathbb{F}^{k\times n}$ from the standard normal distribution. Then the matrix $\boldsymbol{Q} \in \mathbb{F}^{m\times k}$ computed by (2.7) satisfies*

$$\mathbb{E}_{\boldsymbol{\Omega}}\,\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\|_2^2 \le \left(1 + \frac{\varrho}{k-\varrho-\alpha}\right)\cdot\tau_{\varrho+1}^2(\boldsymbol{A}).$$

*Supplementary material for SISC MS#M120106.

†Computing + Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125-5000 (jtropp@cms.caltech.edu).

‡Ecole Polytechnique Federal de Lausanne, 1015 Lausanne, Switzerland (alp.yurtsever@epfl.ch, volkan.cevher@epfl.ch).

§Operations Research and Information Engineering, Cornell University, Ithaca, NY 14850 (udell@cornell.edu).

An analogous result holds for the matrix $\boldsymbol{P} \in \mathbb{F}^{n\times k}$ *computed by* (2.7)*:*

$$\mathbb{E}_{\boldsymbol{\Upsilon}} \|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*\|_2^2 \le \left(1 + \frac{\varrho}{k - \varrho - \alpha}\right) \cdot \tau_{\varrho+1}^2(\boldsymbol{A}).$$

The number $\alpha = 1$ *when* $\mathbb{F} = \mathbb{R}$*, while* $\alpha = 0$ *when* $\mathbb{F} = \mathbb{C}$*.*

This result follows immediately from the proof of [SM6, Thm. 10.5] using Fact SM1.1 to handle both the real and complex case simultaneously. See also [SM9, Sec. 8.2].

SM1.3. Decomposition of the Core Matrix Approximation Error. The first step in the argument is to obtain a formula for the error in the approximation $\boldsymbol{C} - \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}$. The core matrix $\boldsymbol{C} \in \mathbb{F}^{s\times s}$ is defined in (2.8). We constructed the orthonormal matrices $\boldsymbol{P} \in \mathbb{F}^{n\times k}$ and $\boldsymbol{Q} \in \mathbb{F}^{m\times k}$ in (2.7).

Let us introduce matrices whose ranges are complementary to those of $\boldsymbol{P}$ and $\boldsymbol{Q}$:

$$\begin{aligned} \boldsymbol{P}_\perp \boldsymbol{P}_\perp^* &:= \mathbf{I} - \boldsymbol{P}\boldsymbol{P}^* \quad \text{where} \quad \boldsymbol{P}_\perp \in \mathbb{F}^{n\times(n-k)}; \\ \boldsymbol{Q}_\perp \boldsymbol{Q}_\perp^* &:= \mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^* \quad \text{where} \quad \boldsymbol{Q}_\perp \in \mathbb{F}^{m\times(m-k)}. \end{aligned}$$

The columns of $\boldsymbol{P}_\perp$ are orthonormal, and the columns of $\boldsymbol{Q}_\perp$ are orthonormal. Next, introduce the submatrices

$$\begin{aligned} \boldsymbol{\Phi}_1 = \boldsymbol{\Phi}\boldsymbol{Q} \in \mathbb{F}^{s\times k} \quad &\text{and} \quad \boldsymbol{\Phi}_2 = \boldsymbol{\Phi}\boldsymbol{Q}_\perp \in \mathbb{F}^{s\times(m-k)}; \\ \boldsymbol{\Psi}_1^* = \boldsymbol{P}^*\boldsymbol{\Psi}^* \in \mathbb{F}^{k\times s} \quad &\text{and} \quad \boldsymbol{\Psi}_2^* = \boldsymbol{P}_\perp^*\boldsymbol{\Psi}^* \in \mathbb{F}^{(n-k)\times s}. \end{aligned} \tag{SM1.1}$$

With this notation at hand, we can state and prove the first result.

LEMMA SM1.3 (Decomposition of the Core Matrix Approximation). *Assume that the matrices* $\boldsymbol{\Phi}_1$ *and* $\boldsymbol{\Psi}_1$ *have full column rank. Then*

$$\begin{aligned} \boldsymbol{C} - \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P} = \boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}_2(\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}) + (\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}_\perp)\boldsymbol{\Psi}_2^*(\boldsymbol{\Psi}_1^\dagger)^* \\ + \boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}_2(\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}_\perp)\boldsymbol{\Psi}_2^*(\boldsymbol{\Psi}_1^\dagger)^*. \end{aligned}$$

Proof. Adding and subtracting terms, we write the core sketch $\boldsymbol{Z}$ as

$$\boldsymbol{Z} = \boldsymbol{\Phi}\boldsymbol{A}\boldsymbol{\Psi}^* = \boldsymbol{\Phi}(\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*)\boldsymbol{\Psi}^* + (\boldsymbol{\Phi}\boldsymbol{Q})(\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P})(\boldsymbol{P}^*\boldsymbol{\Psi}^*).$$

Using (SM1.1), we identify the matrices $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Psi}_1$. Then left-multiply by $\boldsymbol{\Phi}_1^\dagger$ and right-multiply by $(\boldsymbol{\Psi}_1^\dagger)^*$ to arrive at

$$\boldsymbol{C} = \boldsymbol{\Phi}_1^\dagger\boldsymbol{Z}(\boldsymbol{\Psi}_1^\dagger)^* = \boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}(\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*)\boldsymbol{\Psi}^*(\boldsymbol{\Psi}_1^\dagger)^* + \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}.$$

We have identified the core matrix $\boldsymbol{C}$, defined in (2.8). Move the term $\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}$ to the left-hand side to isolate the approximation error.

To continue, notice that

$$\boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi} = \boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}\boldsymbol{Q}\boldsymbol{Q}^* + \boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}\boldsymbol{Q}_\perp\boldsymbol{Q}_\perp^* = \boldsymbol{Q}^* + \boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}_2\boldsymbol{Q}_\perp^*.$$

Likewise,

$$\boldsymbol{\Psi}^*(\boldsymbol{\Psi}_1^\dagger)^* = \boldsymbol{P}\boldsymbol{P}^*\boldsymbol{\Psi}^*(\boldsymbol{\Psi}_1^\dagger)^* + \boldsymbol{P}_\perp\boldsymbol{P}_\perp^*\boldsymbol{\Psi}^*(\boldsymbol{\Psi}_1^\dagger)^* = \boldsymbol{P} + \boldsymbol{P}_\perp\boldsymbol{\Psi}_2^*(\boldsymbol{\Psi}_1^\dagger)^*.$$

Combine the last three displays to arrive at

$$\boldsymbol{C} - \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P} = (\boldsymbol{Q}^* + \boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}_2\boldsymbol{Q}_\perp^*)(\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*)(\boldsymbol{P} + \boldsymbol{P}_\perp\boldsymbol{\Psi}_2^*(\boldsymbol{\Psi}_1^\dagger)^*).$$

Expand the expression and use the orthogonality relations $\boldsymbol{Q}^*\boldsymbol{Q} = \mathbf{I}$ and $\boldsymbol{Q}_\perp^*\boldsymbol{Q} = \mathbf{0}$ and $\boldsymbol{P}^*\boldsymbol{P} = \mathbf{I}$ and $\boldsymbol{P}^*\boldsymbol{P}_\perp = \mathbf{0}$ to arrive at the desired representation. ∎

SM1.4. Probabilistic analysis of the core matrix. Next, we make distributional assumptions on the dimension reduction maps $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$. We can then study the probabilistic behavior of the error $\boldsymbol{C} - \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}$, conditional on $\boldsymbol{Q}$ and $\boldsymbol{P}$.

LEMMA SM1.4 (Probabilistic Analysis of the Core Matrix). *Assume that the dimension reduction matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are drawn independently from the standard normal distribution. When $s \geq k$, it holds that*

$$\mathbb{E}_{\boldsymbol{\Phi},\boldsymbol{\Psi}}[\boldsymbol{C} - \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}] = \mathbf{0}. \tag{SM1.2}$$

When $s > k + \alpha$, we can express the error as

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\Phi},\boldsymbol{\Psi}} \|\boldsymbol{C} - \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\|_2^2 = \frac{k}{s-k-\alpha} \cdot \|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*\|_2^2 \\ + \frac{k(2k+\alpha-s)}{(s-k-\alpha)^2} \cdot \|\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}_\perp\|_2^2.\end{aligned}$$

When $s < 2k + \alpha$, the last term is nonnegative; when $s \geq 2k + \alpha$, the last term is nonpositive.

Proof. Since $\boldsymbol{\Phi}$ is standard normal, the orthogonal submatrices $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are statistically independent standard normal matrices because of the marginal property of the normal distribution. Likewise, $\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_2$ are statistically independent standard normal matrices. Provided that $s \geq k$, both matrices have full column rank with probability one.

To establish the formula (SM1.2), notice that

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\Phi},\boldsymbol{\Psi}}[\boldsymbol{C} - \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}] = \mathbb{E}_{\boldsymbol{\Phi}_1}\mathbb{E}_{\boldsymbol{\Phi}_2}[\boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}_2(\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P})] + \mathbb{E}_{\boldsymbol{\Psi}_1}\mathbb{E}_{\boldsymbol{\Psi}_2}[(\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}_\perp)\boldsymbol{\Psi}_2^*(\boldsymbol{\Psi}_1^\dagger)^*] \\ + \mathbb{E}\,\mathbb{E}_{\boldsymbol{\Phi}_2}[\boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}_2(\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}_\perp)\boldsymbol{\Psi}_2^*(\boldsymbol{\Psi}_1^\dagger)^*].\end{aligned}$$

We have used the decomposition of the approximation error from Lemma SM1.3. Then we invoke independence to write the expectations as iterated expectations. Since $\boldsymbol{\Phi}_2$ and $\boldsymbol{\Psi}_2$ have mean zero, this formula makes it clear that the approximation error has mean zero.

To study the fluctuations, apply the independence and zero-mean property of $\boldsymbol{\Phi}_2$ and $\boldsymbol{\Psi}_2$ to decompose

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\Phi},\boldsymbol{\Psi}} \|\boldsymbol{C} - \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\|_2^2 = \mathbb{E}_{\boldsymbol{\Phi}} \|\boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}_2(\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P})\|_2^2 + \mathbb{E}_{\boldsymbol{\Psi}} \|(\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}_\perp)\boldsymbol{\Psi}_2^*(\boldsymbol{\Psi}_1^\dagger)^*\|_2^2 \\ + \mathbb{E}_{\boldsymbol{\Phi}}\,\mathbb{E}_{\boldsymbol{\Psi}} \|\boldsymbol{\Phi}_1^\dagger\boldsymbol{\Phi}_2(\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}_\perp)\boldsymbol{\Psi}_2^*(\boldsymbol{\Psi}_1^\dagger)^*\|_2^2.\end{aligned}$$

Continuing, we invoke Fact SM1.1 four times to see that

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\Phi},\boldsymbol{\Psi}} \|\boldsymbol{C} - \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\|_2^2 \\ &\qquad = \frac{k}{s-k-\alpha} \cdot \left[\|\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}\|_2^2 + \|\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}_\perp\|_2^2 + \frac{k}{s-k-\alpha} \cdot \|\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}_\perp\|_2^2\right].\end{aligned}$$

Add and subtract $\|\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}_\perp\|_2^2$ in the bracket to arrive at

$$\begin{aligned}\mathbb{E} \|\boldsymbol{C} - \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\|_2^2 = \frac{k}{s-k-\alpha} \cdot \Bigg[\|\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}\|_2^2 + \|\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}_\perp\|_2^2 + \|\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}_\perp\|_2^2 \\ + \frac{2k+\alpha-s}{s-k-\alpha} \cdot \|\boldsymbol{Q}_\perp^*\boldsymbol{A}\boldsymbol{P}_\perp\|_2^2\Bigg].\end{aligned}$$

Use the Pythagorean Theorem to combine the terms on the first line. ∎

SM1.5. Probabilistic analysis of the compression error. Next, we establish a bound for the expected error in the compression of the matrix $\boldsymbol{A}$ onto the range of the orthonormal matrices $\boldsymbol{Q}$ and $\boldsymbol{P}$, computed in (2.7). This result is similar in spirit to the analysis in [SM6], so we pass lightly over the details.

LEMMA SM1.5 (Probabilistic Analysis of the Compression Error). *For any natural number $\varrho < k - \alpha$, it holds that*

$$\mathbb{E}\,\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*\|_2^2 \le \left(1 + \frac{2\varrho}{k - \varrho - \alpha}\right) \cdot \tau_{\varrho+1}^2(\boldsymbol{A}).$$

Proof Sketch. Introduce the partitioned SVD of the matrix $\boldsymbol{A}$:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \begin{bmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2\end{bmatrix} \begin{bmatrix}\boldsymbol{\Sigma}_1 & \\ & \boldsymbol{\Sigma}_2\end{bmatrix} \begin{bmatrix}\boldsymbol{V}_1^* \\ \boldsymbol{V}_2^*\end{bmatrix} \quad \text{where} \quad \boldsymbol{\Sigma}_1 \in \mathbb{F}^{\varrho\times\varrho}.$$

Define the matrices

$$\begin{aligned} \boldsymbol{\Upsilon}_1 &:= \boldsymbol{\Upsilon}\boldsymbol{U}_1 \in \mathbb{F}^{s\times\varrho} \quad \text{and} \quad \boldsymbol{\Upsilon}_2 := \boldsymbol{\Upsilon}\boldsymbol{U}_2 \in F^{s\times(m-\varrho)}; \\ \boldsymbol{\Omega}_1^* &:= \boldsymbol{V}_1^*\boldsymbol{\Omega}^* \in \mathbb{F}^{\varrho\times s} \quad \text{and} \quad \boldsymbol{\Omega}_2^* := \boldsymbol{V}_2^*\boldsymbol{\Omega}^* \in \mathbb{F}^{(n-\varrho)\times s}; \\ \boldsymbol{P}_1 &:= \boldsymbol{V}_1^*\boldsymbol{P} \in \mathbb{F}^{\varrho\times k} \quad \text{and} \quad \boldsymbol{P}_2 := \boldsymbol{V}_2^*\boldsymbol{P} \in \mathbb{F}^{(n-\varrho)\times k}. \end{aligned}$$

With this notation, we proceed to the proof.

First, add and subtract terms and apply the Pythagorean Theorem to obtain

$$\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*\|_2^2 = \|\boldsymbol{A}(\mathbf{I} - \boldsymbol{P}\boldsymbol{P}^*)\|_2^2 + \|(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*\|_2^2.$$

Use the SVD to decompose the matrix $\boldsymbol{A}$ in the first term, and apply the Pythagorean Theorem again:

$$\begin{aligned} \|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*\|_2^2 = \|(\boldsymbol{U}_2\boldsymbol{\Sigma}_2\boldsymbol{V}_2^*)(\mathbf{I} - \boldsymbol{P}\boldsymbol{P}^*)\|_2^2 \\ + \|(\boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^*)(\mathbf{I} - \boldsymbol{P}\boldsymbol{P}^*)\|_2^2 + \|(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\boldsymbol{A}\boldsymbol{P}\|_2^2. \end{aligned}$$

The result [SM9, Prop. 9.2] implies that the second term satisfies

$$\|(\boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^*)(\mathbf{I} - \boldsymbol{P}\boldsymbol{P}^*)\|_2^2 \le \|\boldsymbol{\Upsilon}_1^\dagger\boldsymbol{\Upsilon}_2\boldsymbol{\Sigma}_2\|_2^2.$$

We can obtain a bound for the third term using the formula [SM6, p. 270, disp. 1]. After a short computation, this result yields

$$\begin{aligned} \|(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\boldsymbol{A}\boldsymbol{P}\|_2^2 &\le \|\boldsymbol{\Sigma}_2\boldsymbol{P}_2\|_2^2 + \|\boldsymbol{\Sigma}_2\boldsymbol{\Omega}_2^*(\boldsymbol{\Omega}_1^*)^\dagger\boldsymbol{P}_1\|_2^2 \\ &\le \|\boldsymbol{\Sigma}_2\|_2^2 + \|\boldsymbol{\Sigma}_2\boldsymbol{\Omega}_2^*(\boldsymbol{\Omega}_1^*)^\dagger\|_2^2. \end{aligned}$$

We can remove $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ because their spectral norms are bounded by one, being submatrices of the orthonormal matrix $\boldsymbol{P}$. Combine the last three displays to obtain

$$\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*\|_2^2 \le \|\boldsymbol{\Sigma}_2\|_2^2 + \|\boldsymbol{\Upsilon}_1^\dagger\boldsymbol{\Upsilon}_2\boldsymbol{\Sigma}_2\|_2^2 + \|\boldsymbol{\Sigma}_2\boldsymbol{\Omega}_2^*(\boldsymbol{\Omega}_1^*)^\dagger\|_2^2.$$

We have used the Pythagorean Theorem again.

Take the expectation with respect to $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Omega}$ to arrive at

$$\begin{aligned} \mathbb{E}\,\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*\|_2^2 &\le \|\boldsymbol{\Sigma}_2\|_2^2 + \mathbb{E}\,\|\boldsymbol{\Upsilon}_1^\dagger\boldsymbol{\Upsilon}_2\boldsymbol{\Sigma}_2\|_2^2 + \mathbb{E}\,\|\boldsymbol{\Sigma}_2\boldsymbol{\Omega}_2^*(\boldsymbol{\Omega}_1^*)^\dagger\|_2^2 \\ &= \|\boldsymbol{\Sigma}_2\|_2^2 + \frac{2\varrho}{k - \varrho - \alpha} \cdot \|\boldsymbol{\Sigma}_2\|_2^2. \end{aligned}$$

Finally, note that $\|\boldsymbol{\Sigma}_2\|_2^2 = \tau_{\varrho+1}^2(\boldsymbol{A})$. ∎

SM1.6. The Endgame. At last, we are prepared to finish the proof of Theorem 5.1. Fix a natural number $\varrho < k - \alpha$. Using the formula (2.9) for the approximation $\hat{\boldsymbol{A}}$, we see that

$$\begin{aligned}\|\boldsymbol{A} - \hat{\boldsymbol{A}}\|_2^2 &= \|\boldsymbol{A} - \boldsymbol{QCP}^*\|_2^2 \\ &= \|\boldsymbol{A} - \boldsymbol{QQ}^*\boldsymbol{APP}^* + \boldsymbol{Q}(\boldsymbol{Q}^*\boldsymbol{AP} - \boldsymbol{C})\boldsymbol{P}^*\|_2^2 \\ &= \|\boldsymbol{A} - \boldsymbol{QQ}^*\boldsymbol{APP}^*\|_2^2 + \|\boldsymbol{Q}(\boldsymbol{Q}^*\boldsymbol{AP} - \boldsymbol{C})\boldsymbol{P}^*\|_2^2.\end{aligned}$$

The last identity is the Pythagorean theorem. Drop the orthonormal matrices in the last term. Then take the expectation with respect to $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$:

$$\mathbb{E}_{\boldsymbol{\Phi},\boldsymbol{\Psi}} \|\boldsymbol{A} - \hat{\boldsymbol{A}}\|_2^2 = \|\boldsymbol{A} - \boldsymbol{QQ}^*\boldsymbol{APP}^*\|_2^2 + \mathbb{E}_{\boldsymbol{\Phi},\boldsymbol{\Psi}} \|\boldsymbol{Q}^*\boldsymbol{AP} - \boldsymbol{C}\|_2^2$$

We treat the two terms sequentially.

To continue, invoke the expression Lemma SM1.4 for the expected error in the core matrix $\boldsymbol{C}$:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\Phi},\boldsymbol{\Psi}} \|\boldsymbol{A} - \hat{\boldsymbol{A}}\|_2^2 \le \left(1 + \frac{k}{s-k-\alpha}\right) \cdot \|\boldsymbol{A} - \boldsymbol{QQ}^*\boldsymbol{APP}^*\|_2^2 \\ + \frac{k(2k+\alpha-s)}{(s-k-\alpha)^2} \cdot \|\boldsymbol{Q}_\perp^*\boldsymbol{AP}_\perp\|_2^2.\end{aligned}$$

Now, take the expectation with respect to $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Omega}$ to arrive at

$$\begin{aligned}\mathbb{E}\,\|\boldsymbol{A} - \hat{\boldsymbol{A}}\|_2^2 \le \left(1 + \frac{k}{s-k-\alpha}\right) \cdot \left(1 + \frac{2\varrho}{k-\varrho-\alpha}\right) \cdot \tau_{\varrho+1}^2(\boldsymbol{A}) \\ + \frac{k(2k+\alpha-s)}{(s-k-\alpha)^2} \cdot \mathbb{E}\,\|\boldsymbol{Q}_\perp^*\boldsymbol{AP}_\perp\|_2^2.\end{aligned} \tag{SM1.3}$$

We have invoked Lemma SM1.5. The last term is nonpositive because we require $s \ge 2k + \alpha$, so we may drop it from consideration. Finally, we optimize over eligible choices $\varrho < k - \alpha$ to complete the argument. The result stated in Theorem 5.1 is algebraically equivalent.

SM2. A Posteriori Error Estimation. This section contains proofs of the bounds on the *a posteriori* error estimator err_2 computed using a Gaussian error sketch. It also establishes the linear algebra results that we need to diagnose spectral decay in the input matrix.

SM2.1. The Frobenius Norm Estimator. Fix an arbitrary matrix $\boldsymbol{M} \in \mathbb{F}^{m\times n}$, which plays the role of the discrepancy $\boldsymbol{A} - \hat{\boldsymbol{A}}_{\mathrm{out}}$. For a parameter q, draw a standard normal dimension reduction map $\boldsymbol{\Theta} \in \mathbb{F}^{q\times m}$. Define the random variable

$$\varphi_2^2 := \frac{1}{\beta q} \cdot \|\boldsymbol{\Theta M}\|_2^2.$$

The field parameter $\beta = 1$ for $\mathbb{F} = \mathbb{R}$ and $\beta = 2$ for $\mathbb{F} = \mathbb{C}$. This random variable can be regarded as a randomized estimator for the Schatten 2-norm of the matrix $\boldsymbol{M}$. The goal of this section is to develop probabilistic results to support this claim.

Remark SM2.1 (Prior Work). The analysis here is similar in spirit to recent papers on randomized trace estimators [SM1, SM7, SM5]. The details here are slightly different, but we claim no novelty of insight.

SM2.1.1. An Alternative Representation. By the unitary invariance of the Schatten norm and the standard normal matrix, we can and will assume that $\boldsymbol{M} = \operatorname{diag}(\sigma_1, \dots, \sigma_m) \in \mathbb{R}^{m \times m}$ is a real diagonal matrix with (weakly) decreasing entries.

Since $\boldsymbol{M}$ is real and diagonal, the estimator can be written as

$$\varphi_2^2 = \frac{1}{\beta q} \cdot \|\boldsymbol{\Theta}\boldsymbol{M}\|_2^2 = \frac{1}{\beta q} \cdot \sum\nolimits_{i=1}^m \sigma_i^2 \cdot \|\boldsymbol{\theta}_{:i}\|_2^2 \sim \frac{1}{\beta q} \cdot \sum\nolimits_{i=1}^m \sigma_i^2 \chi_i^2. \tag{SM2.1}$$

Here, $\boldsymbol{\theta}_{:i}$ is the ith column of $\boldsymbol{\Theta}$. We have also introduced an independent family $\{\chi_i^2 : i = 1, \dots, n\}$ of chi-squared random variables, each with βq degrees of freedom. The symbol $\sim$ denotes equality of distribution.

SM2.1.2. The Mean and Variance. Using the representation (SM2.1), we quickly compute the mean and variance of the estimator. By linearity of expectation,

$$\mathbb{E}\,\varphi_2^2 = \frac{1}{\beta q} \cdot \sum\nolimits_{i=1}^m \sigma_i^2\, \mathbb{E}\,\chi_i^2 = \sum\nolimits_{i=1}^m \sigma_i^2 = \|\boldsymbol{M}\|_2^2.$$

For the second relation, we introduce the mean of a chi-squared variable with βq degrees of freedom. Since the chi-squared variables are independent,

$$\operatorname{Var}[\varphi_2^2] = \frac{1}{(\beta q)^2} \cdot \sum\nolimits_{i=1}^m \sigma_i^4 \operatorname{Var}[\chi_i^2] = \frac{2}{\beta q} \sum\nolimits_{i=1}^m \sigma_i^4 = \frac{2}{\beta q} \|\boldsymbol{M}\|_4^4.$$

We have also used the fact that the variance is 2-homogeneous, and we introduced the variance of a chi-squared variable with βq degrees of freedom.

SM2.1.3. Upper Tail Probabilities. Our goal is to develop bounds on the probability that the estimator takes an extreme value. We begin with the upper tail.

We can use the Laplace transform method. For $\varepsilon \geq 0$, by Markov's inequality,

$$\log \mathbb{P}\left\{\varphi_2^2 \geq (1+\varepsilon) \cdot \|\boldsymbol{M}\|_2^2\right\} \leq \inf_{\eta > 0} \left(-\eta(1+\varepsilon)\|\boldsymbol{M}\|_2^2 + \log \mathbb{E}\, \mathrm{e}^{\eta \varphi_2^2} \right).$$

To compute the moment generating function, we exploit independence of the chi-squared variates in the representation (SM2.1):

$$\log \mathbb{E}\, \mathrm{e}^{\eta \varphi_2^2} = \prod\nolimits_{i=1}^m \log \mathbb{E}\, \mathrm{e}^{(\eta \sigma_i^2/(\beta q)) \cdot \chi_i^2} = \frac{-\beta q}{2} \sum\nolimits_{i=1}^m \log \left[1 - \frac{2\eta\sigma_i^2}{\beta q} \right].$$

The last relation follows when we introduce the moment generating function of a chi-squared variable with βq degrees of freedom. We tacitly assume that η is sufficiently small. We have the bound

$$\log \mathbb{E}\, \mathrm{e}^{\eta \varphi_2^2} \leq \frac{-\beta q}{2} \log \left[1 - \frac{2\eta \sum_{i=1}^m \sigma_i^2}{\beta q} \right] = \frac{-\beta q}{2} \log \left[1 - \frac{2\eta \|\boldsymbol{M}\|_2^2}{\beta q} \right].$$

This point follows by repeated application of the numerical inequality $(1-a)(1-b) \geq 1 - a - b$, valid when $ab \geq 0$. In summary,

$$\begin{aligned} \log \mathbb{P}\left\{\varphi_2^2 \geq (1+\varepsilon) \cdot \|\boldsymbol{M}\|_2^2\right\} &\leq \inf_{\eta > 0} \left(-\eta(1+\varepsilon)\|\boldsymbol{M}\|_2^2 - \frac{\beta q}{2} \log \left[1 - \frac{2\eta \|\boldsymbol{M}\|_2^2}{\beta q} \right] \right) \\ &= \frac{-\beta q}{2} \left[\varepsilon - \log(1+\varepsilon) \right]. \end{aligned}$$

Exponentiate this expression to reach the required bound.

Remark SM2.2 (Improvements). Sharper estimates are possible in the case where the stable rank of the matrix $\boldsymbol{M}$ is large. For results of this type, see [SM5].

SM2.1.4. Lower Tail Probabilities. For the lower tail, we use essentially the same argument. Therefore, we gloss over most of the details.

For $\varepsilon \in (0,1)$, the Laplace transform method gives

$$\log \mathbb{P}\left\{\varphi_2^2 \le (1-\varepsilon)\cdot \|\boldsymbol{M}\|_2^2\right\} \le \inf_{\eta>0}\left(\eta(1-\varepsilon)\|\boldsymbol{M}\|_2^2 + \log \mathbb{E}\,\mathrm{e}^{-\eta\varphi_2^2}\right).$$

We bound the moment generating function as

$$\log \mathbb{E}\,\mathrm{e}^{-\eta\varphi_2^2} \le \frac{-\beta q}{2}\log\left[1+\frac{2\eta\|\boldsymbol{M}\|_2^2}{\beta q}\right].$$

Combine the last two displays:

$$\begin{aligned}\log \mathbb{P}\left\{\varphi_2^2 \le \varepsilon\cdot \|\boldsymbol{M}\|_2^2\right\} &\le \inf_{\eta>0}\left(\eta(1-\varepsilon)\|\boldsymbol{M}\|_2^2 - \frac{\beta q}{2}\log\left[1+\frac{2\eta\|\boldsymbol{M}\|_2^2}{\beta q}\right]\right)\\ &= \frac{\beta q}{2}\left[\varepsilon + \log(1-\varepsilon)\right].\end{aligned}$$

Exponentiate this expression to reach the desired bound.

SM2.2. Diagnosing Spectral Decay. In this section, we explain why the square root of the tail energy is a Lipschitz function. For a matrix $\boldsymbol{A} \in \mathbb{F}^{m\times n}$ and an integer $r \ge 0$, recall that

$$\tau_{r+1}^2(\boldsymbol{A}) = \sum\nolimits_{j>r}\sigma_j^2(\boldsymbol{A}) = \sum\nolimits_{j>r}\lambda_j(\boldsymbol{A}^*\boldsymbol{A}).$$

As usual, λ_j returns the jth largest eigenvalue of an Hermitian matrix. Ky Fan's minimum principle [SM2, Prob. I.6.15] gives a variational representation for this quantity:

$$\tau_{r+1}^2(\boldsymbol{A}) = \min_{\boldsymbol{U}\in\mathbb{F}^{n\times(n-r)}}\mathrm{tr}[\boldsymbol{U}^*(\boldsymbol{A}^*\boldsymbol{A})\boldsymbol{U}] = \min_{\boldsymbol{U}\in\mathbb{F}^{n\times(n-r)}}\|\boldsymbol{A}\boldsymbol{U}\|_2^2$$

where $\boldsymbol{U}$ ranges over matrices with orthonormal columns. As a consequence, for conformal matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we have

$$\begin{aligned}\tau_{r+1}(\boldsymbol{A}) - \tau_{r+1}(\boldsymbol{B}) &= \min\nolimits_{\boldsymbol{U}}\|\boldsymbol{A}\boldsymbol{U}\|_2 - \min\nolimits_{\boldsymbol{U}}\|\boldsymbol{B}\boldsymbol{U}\|_2\\ &\le \|\boldsymbol{A}\boldsymbol{U}_{\boldsymbol{B}}\|_2 - \|\boldsymbol{B}\boldsymbol{U}_{\boldsymbol{B}}\|_2\\ &= \|(\boldsymbol{A}-\boldsymbol{B})\boldsymbol{U}_{\boldsymbol{B}}\|_2 \le \|\boldsymbol{A}-\boldsymbol{B}\|_2.\end{aligned}$$

We have written $\boldsymbol{U}_{\boldsymbol{B}}$ for the orthonormal matrix in $\mathbb{F}^{n\times(n-r)}$ that minimizes the functional $\boldsymbol{U}\mapsto\|\boldsymbol{B}\boldsymbol{U}\|_2$. The last inequality follows because $\boldsymbol{U}_{\boldsymbol{B}}$ has spectral norm one. Reverse the roles of the two matrices to conclude that

$$|\tau_{r+1}(\boldsymbol{A}) - \tau_{r+1}(\boldsymbol{B})| \le \|\boldsymbol{A}-\boldsymbol{B}\|_2.$$

This is the advertised result.

SM3. Code & Pseudocode. This section contains pseudocode for the dimension reduction maps described in section 3. We use the same mathematical notation as the rest of the paper. We also rely on MATLAB R2018B commands, which appear in typewriter font. The electronic materials include a MATLAB implementation of these methods.

Algorithm SM3.1 *Gaussian Dimension Reduction Map.* (subsection 3.1)

class GAUSSDR (DIMREDUX) ▷ Subclass of DIMREDUX
  local variable $\boldsymbol{\Xi}$ (dense matrix)
  function RANDN($d, N; \mathbb{F}$) ▷ Gaussian matrix over field $\mathbb{F}$
    if $\mathbb{F} = \mathbb{R}$ **then return** `randn`(d, N)
    if $\mathbb{F} = \mathbb{C}$ **then return** `randn`(d, N) + `1i * randn`(d, N)
  function GAUSSDR(k, N) ▷ Constructor
    $\boldsymbol{\Xi} \leftarrow$ RANDN($d, N; \mathbb{F}$) ▷ Gaussian over $\mathbb{F}$
  function GAUSSDR.MTIMES(`DRmap`, $\boldsymbol{M}$)
    return `mtimes`$(\boldsymbol{\Xi}, \boldsymbol{M})$

Algorithm SM3.2 *SSRFT Dimension Reduction Map.* (subsection 3.2)

class SSRFT (DIMREDUX) ▷ Subclass of DIMREDUX
local variables `coords`, $\texttt{perm}_j$, $\boldsymbol{\varepsilon}_j$ for $j = 1, 2$
function SSRFT(d, N) ▷ Constructor
  `coords` $\leftarrow$ `randperm`(N, d)
  $\texttt{perm}_j \leftarrow$ `randperm`(N) for $j = 1, 2$
  $\boldsymbol{\varepsilon}_j \leftarrow$ `sign`(RANDN$(N, 1; \mathbb{F})$) for $j = 1, 2$
function SSRFT.MTIMES(`DRmap`, $\boldsymbol{M}$)
  if $\mathbb{F} = \mathbb{R}$ **then**
    $\boldsymbol{M} \leftarrow$ `dct`$(\text{diag}(\boldsymbol{\varepsilon}_1)\boldsymbol{M}(\texttt{perm}_1, :))$
    $\boldsymbol{M} \leftarrow$ `dct`$(\text{diag}(\boldsymbol{\varepsilon}_2)\boldsymbol{M}(\texttt{perm}_2, :))$
  if $\mathbb{F} = \mathbb{C}$ **then**
    $\boldsymbol{M} \leftarrow$ `dft`$(\text{diag}(\boldsymbol{\varepsilon}_1)\boldsymbol{M}(\texttt{perm}_1, :))$
    $\boldsymbol{M} \leftarrow$ `dft`$(\text{diag}(\boldsymbol{\varepsilon}_2)\boldsymbol{M}(\texttt{perm}_2, :))$
  return $\boldsymbol{M}$(`coords`, :)

- The template for the DIMREDUX class appears in the body of the paper as Algorithm 3.1.
- Algorithm SM3.1 defines a Gaussian dimension reduction class (GAUSSDR), which is a subclass of DIMREDUX. It describes the constructor and the left and right action of this dimension reduction map. See subsection 3.1 for the explanation.
- Algorithm SM3.2 defines a SSRFT dimension reduction class (SSRFT). It is a subclass of DIMREDUX. It describes the constructor and the left and right action of this dimension reduction map. See subsection 3.2 for the explanation.
- Algorithm SM3.3 defines a sparse dimension reduction class (SPARSEDR), which is a subclass of DIMREDUX. It describes the constructor and the left and right action of this dimension reduction map. See subsection 3.3 for the explanation.

SM4. Supplemental Numerical Results. This section summarizes the additional numerical results that are presented in this supplement. The MATLAB code in

Algorithm SM3.3 *Sparse Dimension Reduction Map.* (subsection 3.3)

```
class SparseDR (DimRedux)                                  ▷ Subclass of DimRedux
local variable Ξ (sparse matrix)
function SparseDR(d, N)                                    ▷ Constructor
    ζ ← min{d, 8}                                          ▷ Sparsity of each column
    for j = 1, ..., N do
        Ξ(randperm(d, ζ), j) ← sign(randn(ζ, 1; F))
function SparseDR.mtimes(DRmap, M)
    return mtimes(Ξ, M)
```

the electronic materials can reproduce these experiments.

SM4.1. Alternative Sketching and Reconstruction Methods. In this section, we give full mathematical descriptions of other sketching and reconstruction methods from the literature. We compare our approach against these algorithms.

SM4.1.1. The [HMT11] Method. The paper [SM6, Sec. 5.4] describes a one-pass SVD algorithm, which can be reinterpreted as a sketching algorithm for low-rank matrix approximation. This method simplifies a more involved approach [SM11, Sec. 5.2] due to Woolfe et al. The two approaches have similar performance in practice.

This method uses two dimension reduction maps, controlled by one parameter k:

$$\boldsymbol{\Upsilon} \in \mathbb{F}^{k \times m} \quad \text{and} \quad \boldsymbol{\Omega} \in \mathbb{R}^{k \times n}.$$

The sketch takes the form

$$\boldsymbol{X} = \boldsymbol{\Upsilon}\boldsymbol{A} \quad \text{and} \quad \boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\Omega}.$$

To obtain a rank-r approximation from the sketch, we first compute r leading singular vectors of the sketch matrices:

$$\begin{aligned} (\boldsymbol{P}, \sim, \sim) &= \texttt{svd}(\boldsymbol{X}^*, \texttt{'econ'}) \qquad \text{and} \qquad \boldsymbol{P} = \boldsymbol{P}(\texttt{:}, \texttt{1:r}); \\ (\boldsymbol{Q}, \sim, \sim) &= \texttt{svd}(\boldsymbol{Y}, \texttt{'econ'}) \qquad \text{and} \qquad \boldsymbol{Q} = \boldsymbol{Q}(\texttt{:}, \texttt{1:r}). \end{aligned}$$

Next, we compute two separate estimates for the core matrix by solving two families of least-squares problems:

$$\boldsymbol{C}_1 = (\boldsymbol{Q}^*\boldsymbol{Y})(\boldsymbol{P}^*\boldsymbol{\Omega})^\dagger \in \mathbb{F}^{r \times r} \quad \text{and} \quad \boldsymbol{C}_2^* = (\boldsymbol{P}^*\boldsymbol{X})(\boldsymbol{Q}^*\boldsymbol{\Upsilon})^\dagger \in \mathbb{F}^{r \times r}.$$

Combine these two estimates and compute the SVD:

$$(\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}) = \texttt{svd}((\boldsymbol{C}_1 + \boldsymbol{C}_2)/2).$$

Last, we obtain the rank-r approximation in factored form:

$$\hat{\boldsymbol{A}}_{\text{hmt}} := (\boldsymbol{Q}\boldsymbol{U})\boldsymbol{\Sigma}(\boldsymbol{P}\boldsymbol{V})^*.$$

This approach is not competitive with more modern techniques. Some of the deficiencies stem from truncating the singular vectors to rank r at the first step of the procedure; see Figures SM21 and SM22.

SM4.1.2. The [TYUC17] Method. In our previous paper, we developed and analyzed a sketching algorithm [SM8, Alg. 7] for low-rank matrix approximation. Our work contains a detailed theoretical analysis, prescriptions for choosing algorithm parameters, and an extensive numerical evaluation. We later discovered that this method is algebraically (but not numerically) equivalent to a proposal of Clarkson & Woodruff [SM4, Thm. 4.9]. The paper [SM4] also lacks reliable instructions for implementation.

This approach uses two dimension reduction maps that are indexed by two parameters k, ℓ:

$$\boldsymbol{\Upsilon} \in \mathbb{F}^{\ell \times m} \quad \text{and} \quad \boldsymbol{\Omega} \in \mathbb{F}^{k \times n} \quad \text{where } k \leq \ell.$$

The sketch takes the form

$$\boldsymbol{X} = \boldsymbol{\Upsilon}\boldsymbol{A} \quad \text{and} \quad \boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\Omega}^*.$$

To obtain a rank-r approximation from the sketch, we compute a thin orthogonal–triangular decomposition:

$$\boldsymbol{Y} =: \boldsymbol{Q}\boldsymbol{R} \quad \text{where } \boldsymbol{Q} \in \mathbb{F}^{m \times k}.$$

Then we form the approximation:

$$\hat{\boldsymbol{A}}_{\text{tyuc}} := \boldsymbol{Q}[\![(\boldsymbol{\Upsilon}\boldsymbol{Q})^\dagger \boldsymbol{X}]\!]_r. \tag{SM4.1}$$

Of course, we solve the least-squares problems, rather than computing and applying the pseudoinverse. We use a dense SVD or a randomized SVD [SM6] to calculate the best rank-r approximation.

This method works well, but it uses more storage than necessary because ℓ needs to be somewhat larger than k. The algorithm can also be sensitive to the relative size of the parameters k, ℓ.

SM4.1.3. The [Upa16] Method. In a paper on privacy-preserving matrix approximation, Upadhyay [SM10, Sec. 3] developed an algorithm that also serves for streaming low-rank matrix approximation. This method simplifies a far more complicated approach due to Boutsidis et al. [SM3, Sec. 6].

Upadhyay proposed the sketch (2.2)–(2.4), which depends on two parameters k, s. We are building on his idea in this paper. In contrast to our work, Upadhyay designs a rank-r reconstruction algorithm using the "sketch-and-solve" framework; see subsection 2.8.

His approach leads to the following algorithm. First, compute orthonormal bases $\boldsymbol{Q}$ and $\boldsymbol{P}$ for the range and co-range:

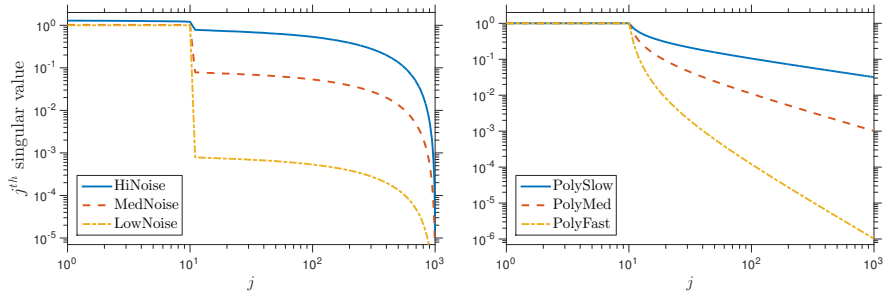
$$\begin{aligned} \boldsymbol{X}^* &=: \boldsymbol{P}\boldsymbol{R}_1 \quad \text{where} \quad \boldsymbol{P} \in \mathbb{F}^{n \times k}; \\ \boldsymbol{Y} &=: \boldsymbol{Q}\boldsymbol{R}_2 \quad \text{where} \quad \boldsymbol{Q} \in \mathbb{F}^{m \times k}. \end{aligned}$$

Next, form thin singular value decompositions:

$$\boldsymbol{\Phi}\boldsymbol{Q} = \boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^* \in \mathbb{F}^{s \times k} \quad \text{and} \quad \boldsymbol{\Psi}\boldsymbol{P} = \boldsymbol{U}_2\boldsymbol{\Sigma}_2\boldsymbol{V}_2^* \in \mathbb{F}^{s \times k}.$$

Construct the rank-r approximation using the formula

$$\hat{\boldsymbol{A}}_{\text{upa}} := \boldsymbol{Q}\boldsymbol{V}_1\boldsymbol{\Sigma}_1^\dagger \, [\![\boldsymbol{U}_1^*\boldsymbol{Z}\boldsymbol{U}_2]\!]_r \, \boldsymbol{\Sigma}_2^\dagger\boldsymbol{V}_2^*\boldsymbol{P}^*. \tag{SM4.2}$$



(A) Low-Rank + Noise (B) Polynomial Decay

(C) Exponential Decay (D) Data Matrices

FIG. SM1. ***Spectra of input matrices.*** *Plots of the singular value spectrum for an input matrix from each of the synthetic classes (*`LowRank`*,* `PolyDecay`*,* `ExpDecay` *with effective rank* $R = 10$*) and from each of the real data classes (*`MinTemp`*,* `StreamVel`*,* `MaxCut`*,* `PhaseRetrieval`*) described in subsection* 7.3.

We use a truncated SVD to perform the rank truncation of the central matrix. Of course, we should take care in applying the pseudoinverses.

Superficially, the approximation $\hat{\boldsymbol{A}}_{\text{upa}}$ may appear similar to the approximation we developed in (2.10). Nevertheless, they are designed using different principles, and their performance is quite different in practice. The [Upa16] method cannot achieve high relative accuracy, even for matrices with rapid spectral decay. Furthermore, it has the bizarre feature that decreasing the rank parameter r can actually make the approximation less reliable! See Figures SM23 and SM24.

SM4.2. Spectra of Input Matrices. Figure SM1 plots the spectrum of each of the synthetic and application matrices that we use in our experiments.

SM4.3. Insensitivity to the Dimension Reduction Map. Our first experiment is designed to show that the proposed rank-r reconstruction formula (2.10) is insensitive to the distribution of the dimension reduction map at the oracle parameter values (subsection 7.2.2) for synthetic input matrices.

We plot the oracle error for (2.10) as a function of storage budget T for Gaussian, SSRFT, and sparse dimension reduction maps. See Figures SM2 to SM7. The curves are almost identical, except that the unitary SSRFT map performs slightly *better* than the others when the storage budget is very large. Similar results hold for matrices drawn from real applications.

We have also found that the other reconstruction methods [HMT11], [TYUC17], and [Upa16] are insensitive to the choice of dimension reduction map. These observa-

tions justify the transfer of theoretical and empirical results for Gaussians to SSRFT and sparse dimension reduction maps.

SM4.4. Achieving the Oracle Performance. Next, we show that we can almost achieve the oracle error by implementing (2.10) with sketch size parameters chosen using our theory.

We perform the following experiment. For synthetic input matrices, we compare the oracle performance (subsection 7.2.2) of our rank-r approximation (2.10) with its performance at the theoretical parameters proposed in subsection 5.4. (In the formula (5.7) for a flat spectrum, we set the tail location $\hat{\varrho} = r$.) We use Gaussian dimension reduction maps, but equivalent results hold for other types of dimension reduction maps. Plots of the results appear in Figures SM8 to SM13.

For most of the examples, the general parameter choice (5.6) is able to deliver a relative error that tracks the oracle error closely. The parameter choice (5.7) for a flat spectrum works somewhat better for matrices whose spectral tail exhibits slow decay (`LowRankLowNoise`, `LowRankMedNoise`, `LowRankHiNoise`). We also learn that the theoretical formulas are not entirely reliable when the storage budget is very small. Matrices with a lot of tail energy (`LowRankHiNoise`, `PolyDecaySlow`) are very hard to approximate accurately with a sketching algorithm, so it is not surprising that our theoretical parameter choices fall short of the oracle parameters in these cases.

SM4.5. Algorithm Comparisons for Synthetic Instances. We compared all four of the reconstruction formulas at the oracle parameters for a wide range of synthetic problem instances. See subsection 7.6 for details.

Figures SM14, SM15, SM17, and SM18 contain the results for matrices with effective rank $R = 5$ and $R = 20$ with relative error measured in Schatten 2-norm and Schatten ∞-norm.

SM4.6. Algorithm Comparisons for Real Data Instances. In this experiment, we compared all four of the reconstruction formulas at the oracle parameters and at theoretically chosen parameters for several application examples.

Here are the details of the *a priori* parameter selections for the several methods. For the proposed method (2.10), we use the "natural" parameter choice (5.6) that follows from our theoretical analysis. The [Upa16] algorithm uses the same sketch—but lacks a comparable theory—so we instantiate it with the parameters (5.6). For [TYUC17], we assume that the input matrix $\boldsymbol{A} \in \mathbb{F}^{m \times n}$ is tall ($m \geq n$), and we use the theoretically motivated parameter values

$$k = \max\{r + \alpha + 1, \lfloor (T - n\alpha)/(m + 2n) \rfloor\} \quad \text{and} \quad \ell = \lfloor (T - km)/n \rfloor.$$

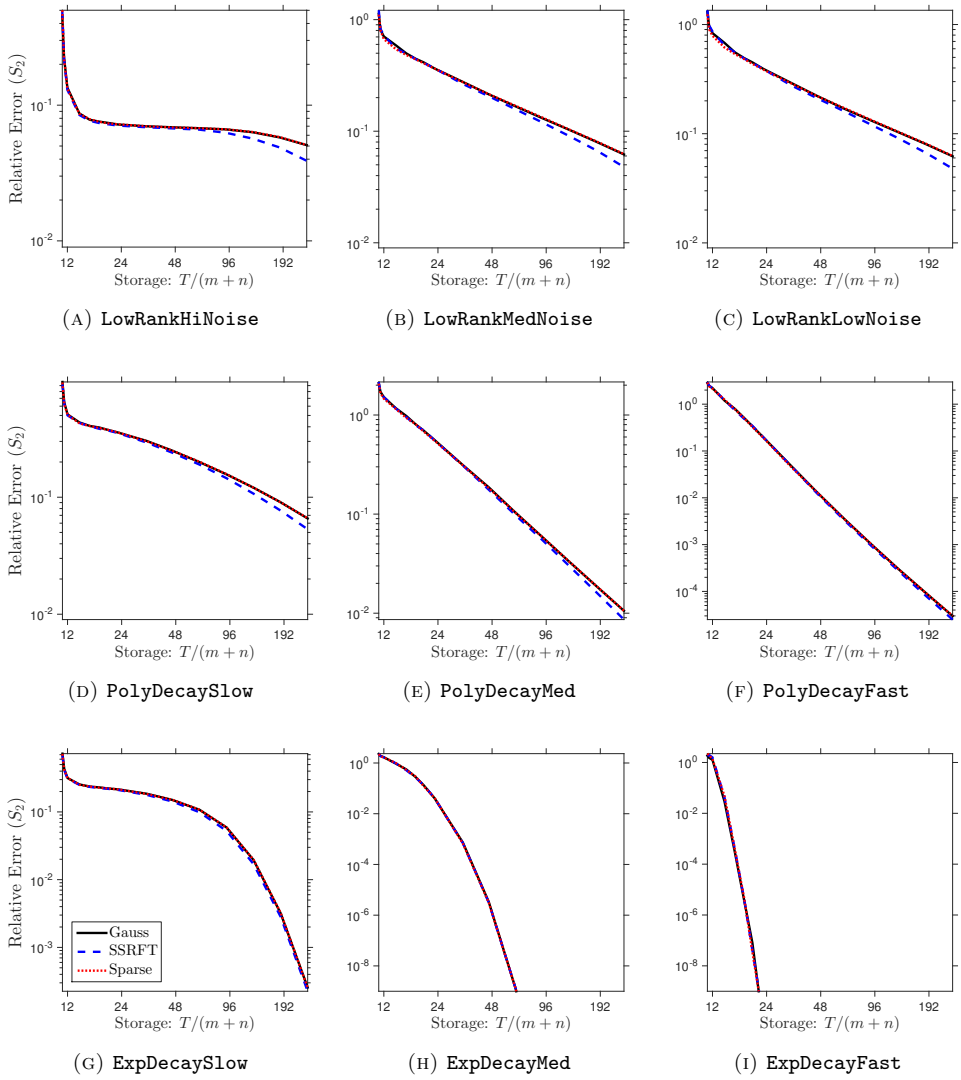
This choice adapts the arguments in [SM8, Sec. 4.5.2] to use the current definition of the storage budget T. The [HMT11] algorithm does not have any free parameters.

SM4.7. Flow-Field Reconstruction. Figure SM20 illustrates the streamwise velocity field `StreamVel` and its rank-10 approximation via (2.10) using storage budget $T = 48(m + n)$ and the parameter choices (5.6). We see that the approximation captures the large-scale features of the flow, although there are small errors visible for the higher-order singular vectors.

We also performed the same experiment with the algorithms [HMT11], [Upa16], and [TYUC17]. We set the truncation rank $r = 5$ and $r = 10$ to see whether this change affects the behavior of the methods. We plot the leading left singular vectors of the flow fields in Figures SM21 to SM25. For truncation $r = 10$, all of the algorithms

produce reasonable results. Nevertheless, with algorithms [HMT11], [Upa16], and [TYUC17], the singular vector estimates for rank $6, 7, 8, 9$ start to deviate from the singular vectors of the original matrix.

When we change the truncation rank to $r = 5$, our methods [TYUC17] and (2.10) give exactly the same singular vector estimates as with $r = 10$, by construction of the algorithm. On the other hand, the methods [HMT11] and [Upa16] behave far worse when $r = 5$ than when $r = 10$. This feature is both strange and dissatisfying. By itself, this lack of stability is already enough to disqualify the algorithms [HMT11] and [Upa16] from practical use.

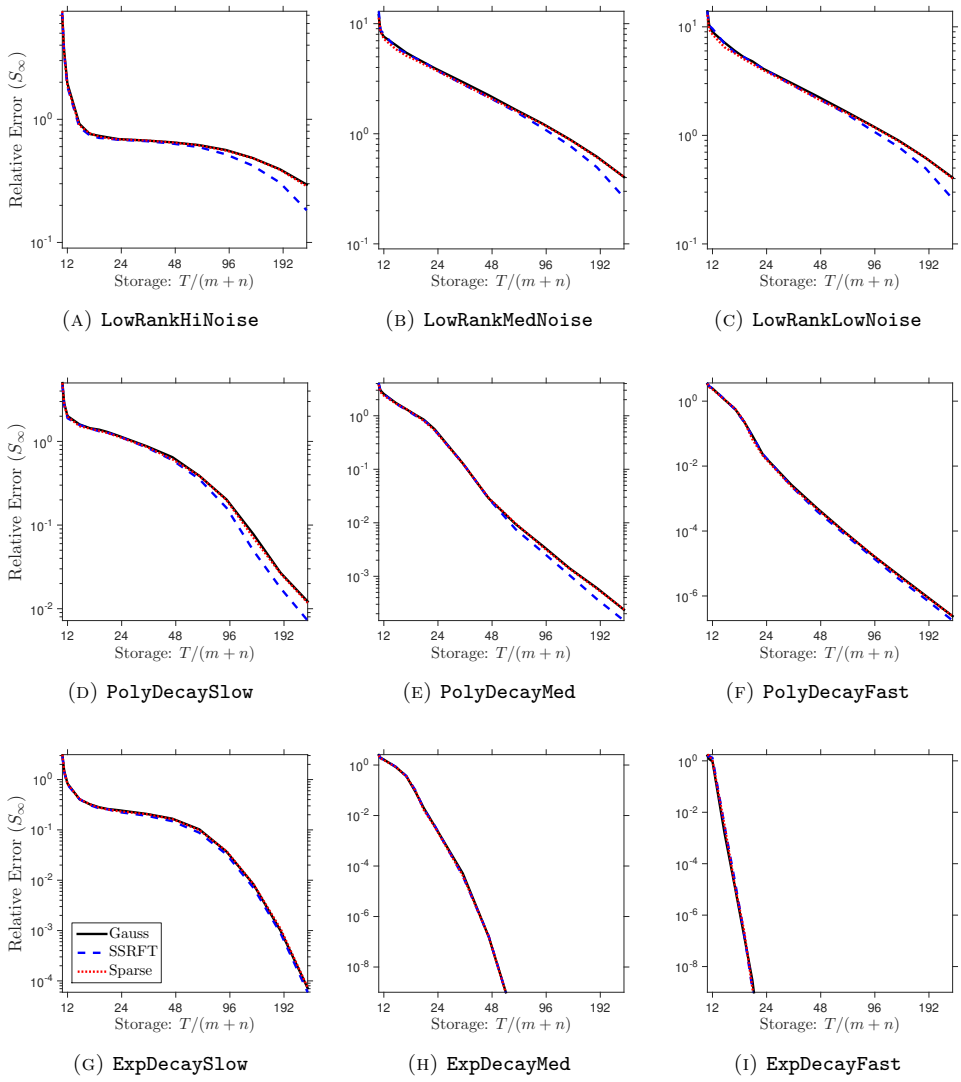


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM2. ***Insensitivity of proposed method to the dimension reduction map.*** *(Effective rank $R = 5$, approximation rank $r = 10$, Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection SM4.3 for details.*

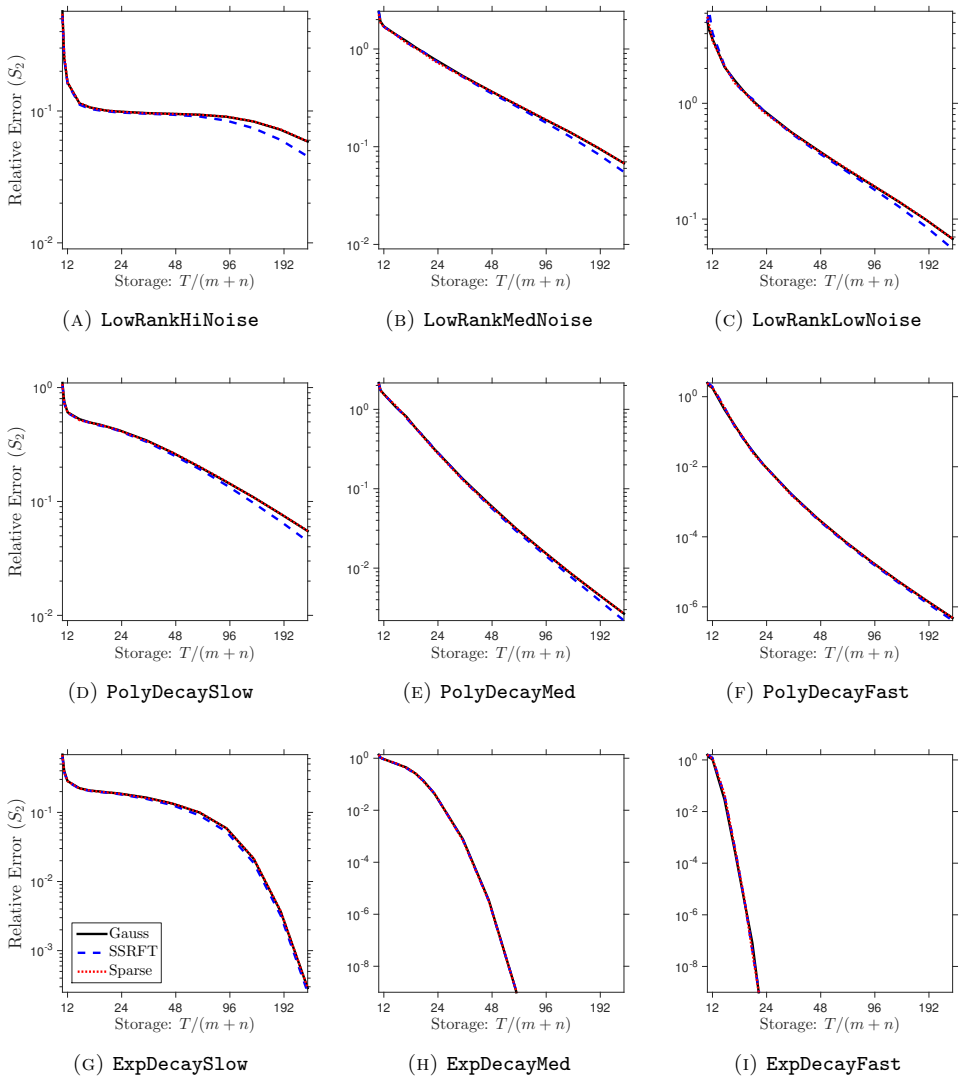


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM3. ***Insensitivity of proposed method to the dimension reduction map.*** *(Effective rank $R = 5$, approximation rank $r = 10$, Schatten ∞-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection SM4.3 for details.*

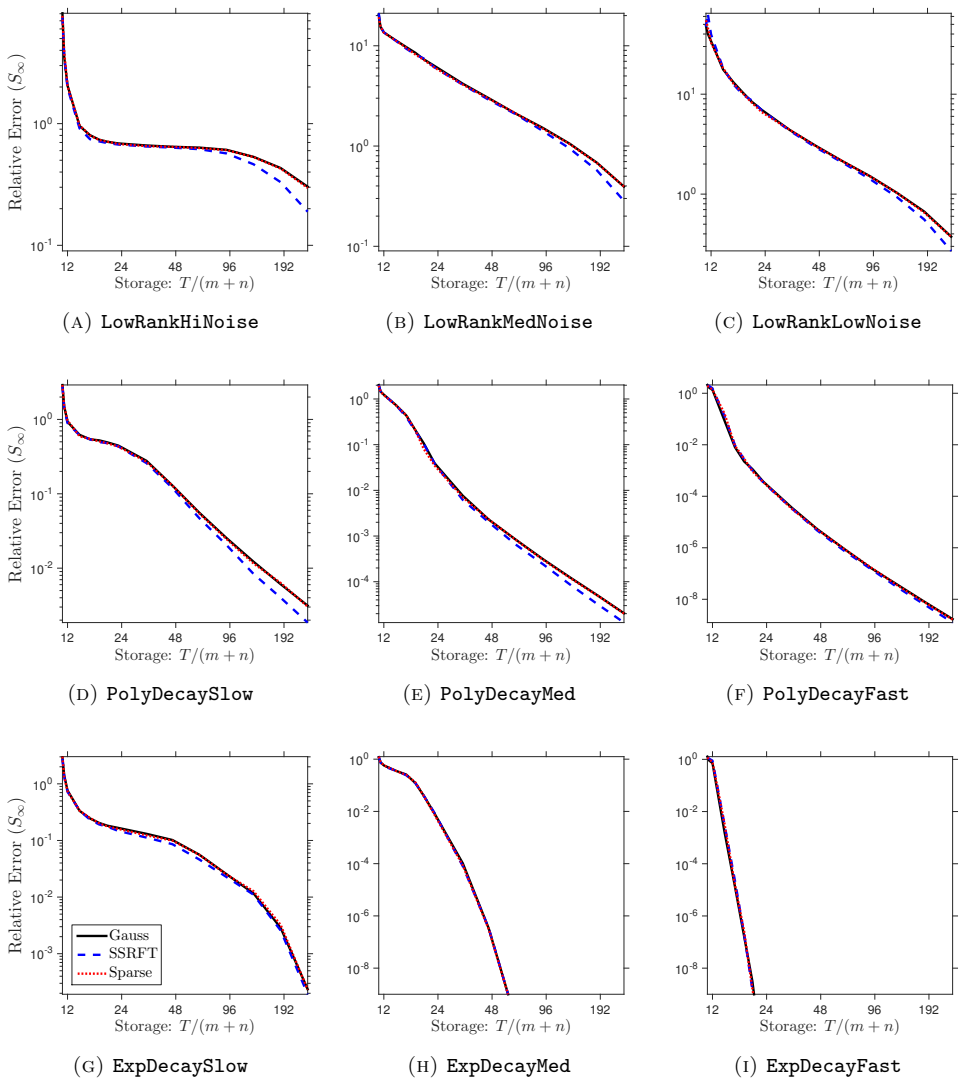


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM4. ***Insensitivity of proposed method to the dimension reduction map.*** *(Effective rank $R = 10$, approximation rank $r = 10$, Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection SM4.3 for details.*

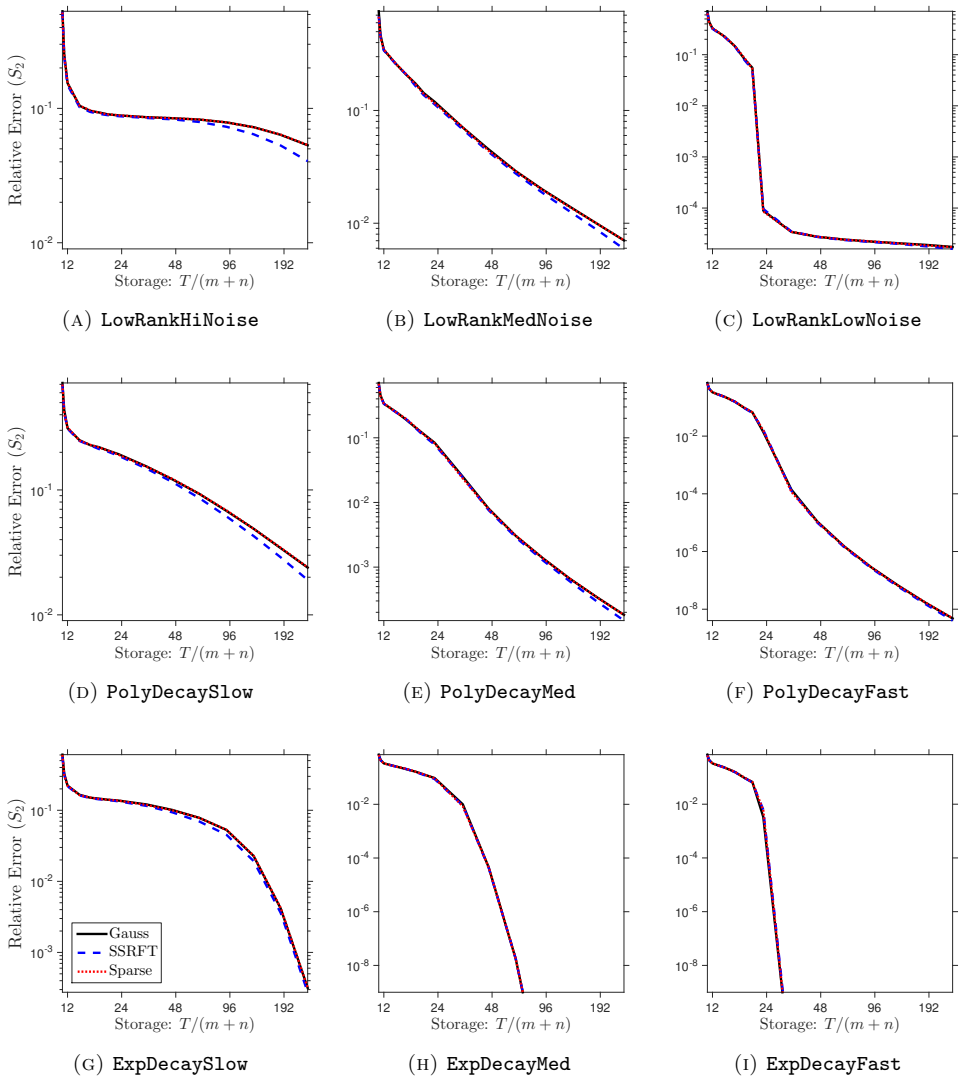


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM5. ***Insensitivity of proposed method to the dimension reduction map.*** *(Effective rank $R = 10$, approximation rank $r = 10$, Schatten ∞-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection SM4.3 for details.*

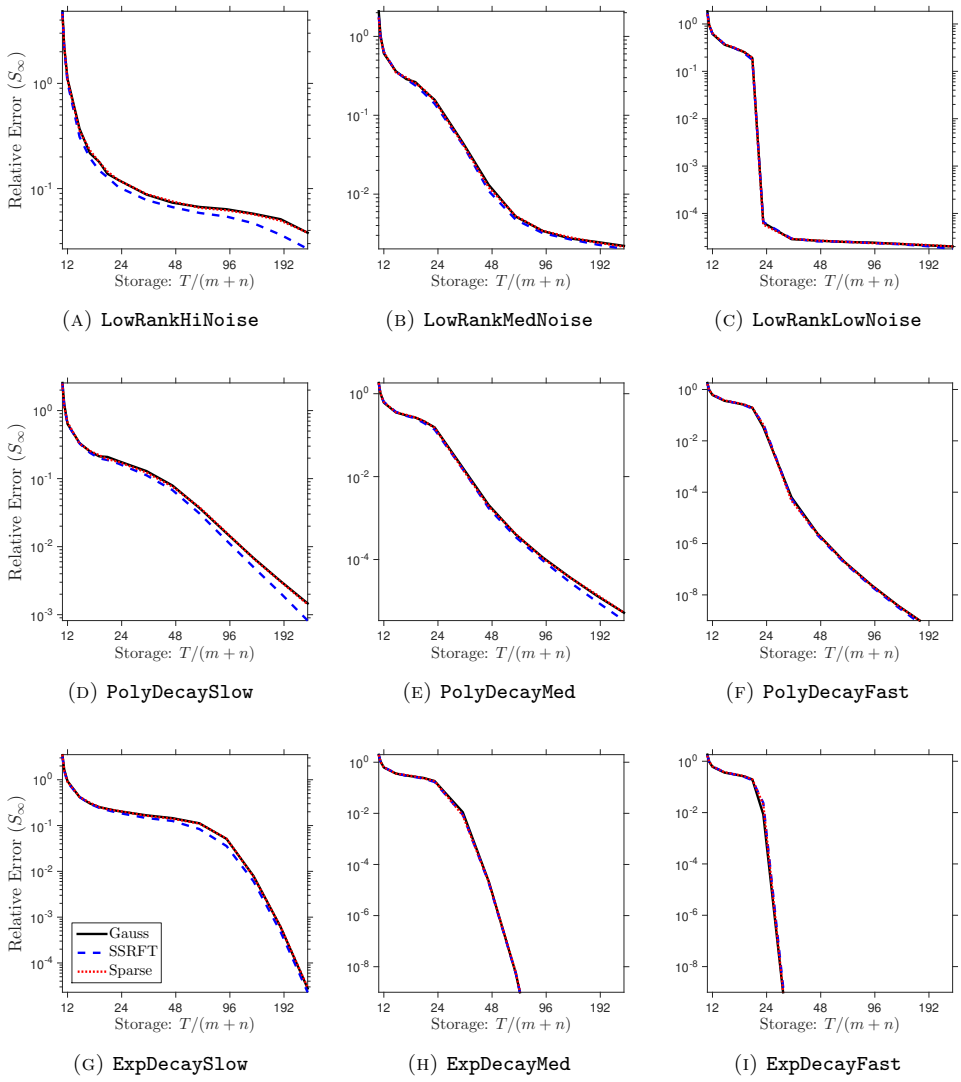


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM6. ***Insensitivity of proposed method to the dimension reduction map.*** *(Effective rank $R = 20$, approximation rank $r = 10$, Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection SM4.3 for details.*

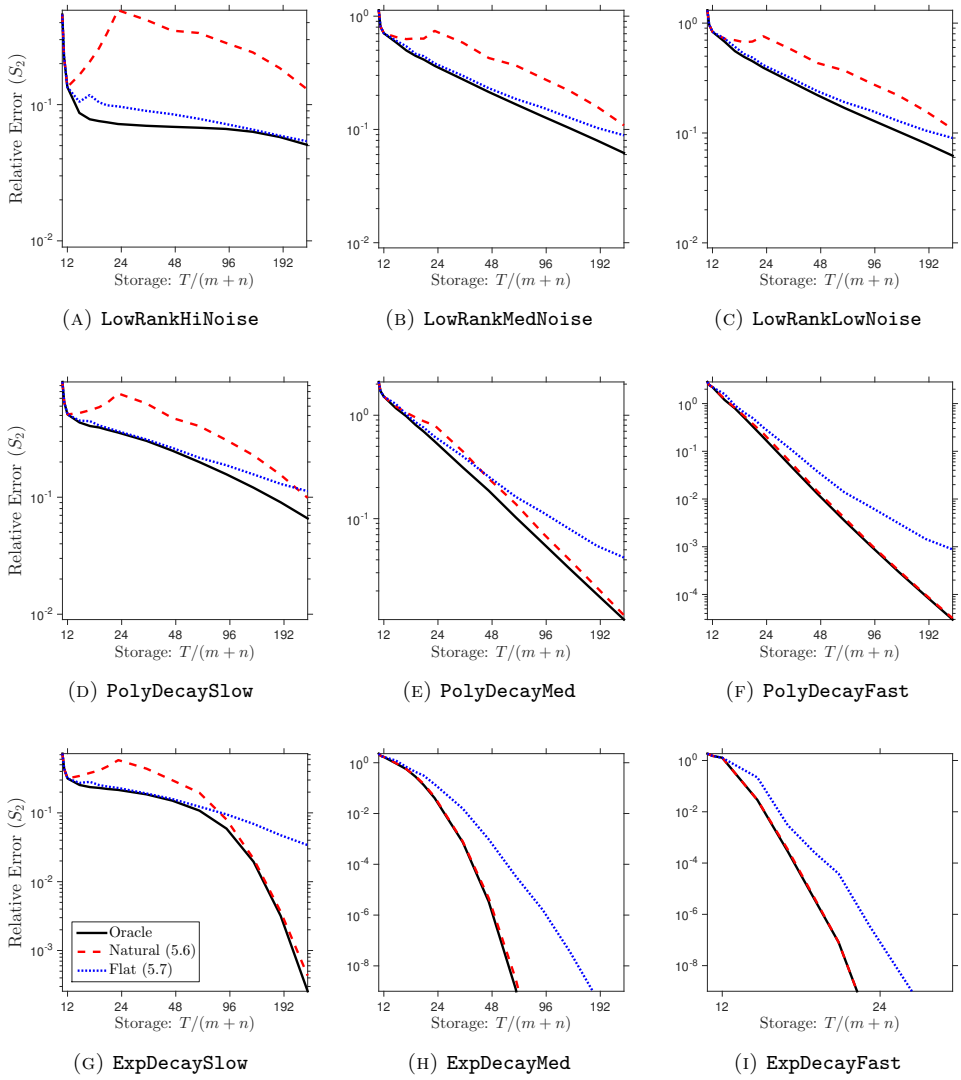


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM7. ***Insensitivity of proposed method to the dimension reduction map.*** *(Effective rank $R = 20$, approximation rank $r = 10$, Schatten ∞-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection SM4.3 for details.*

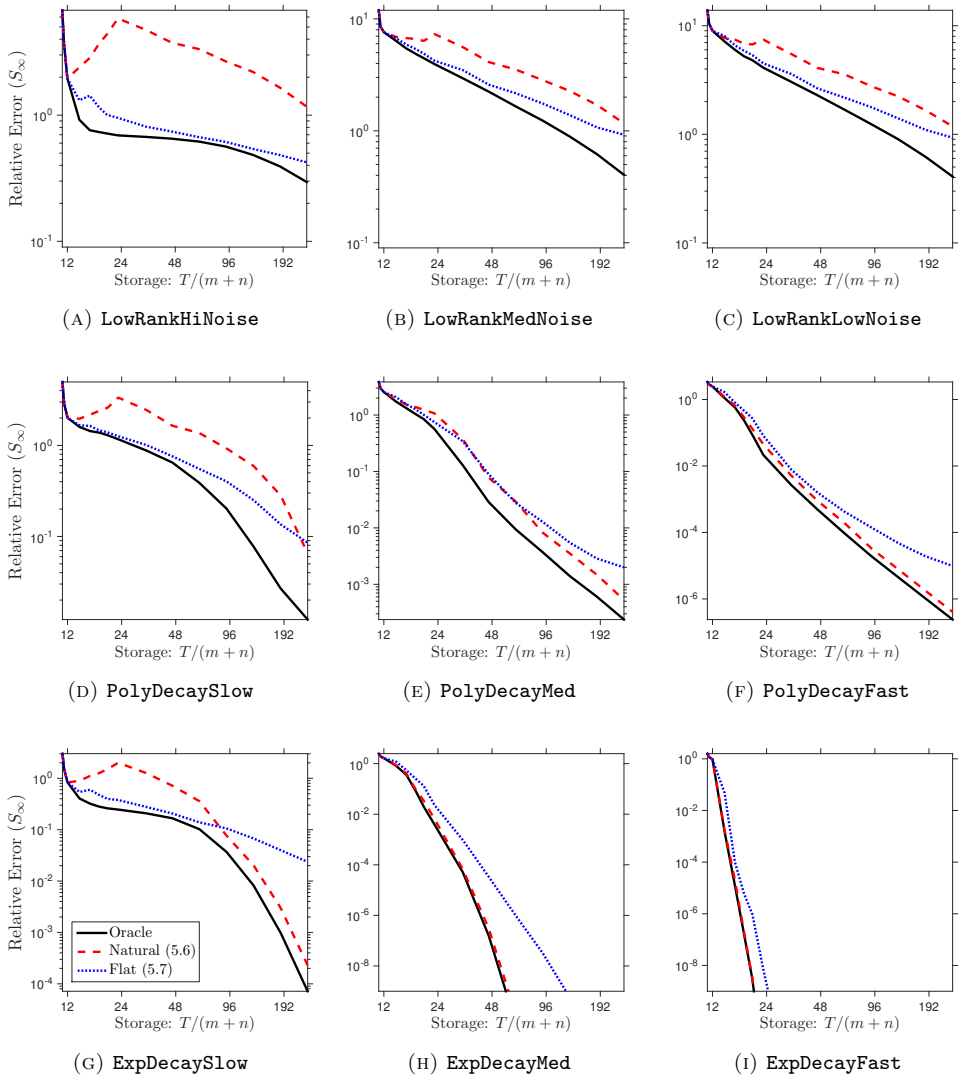


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM8. ***Relative error for proposed method with* a priori *parameters.*** *(Gaussian maps, effective rank $R = 5$, approximation rank $r = 10$, Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection SM4.4 for details.*

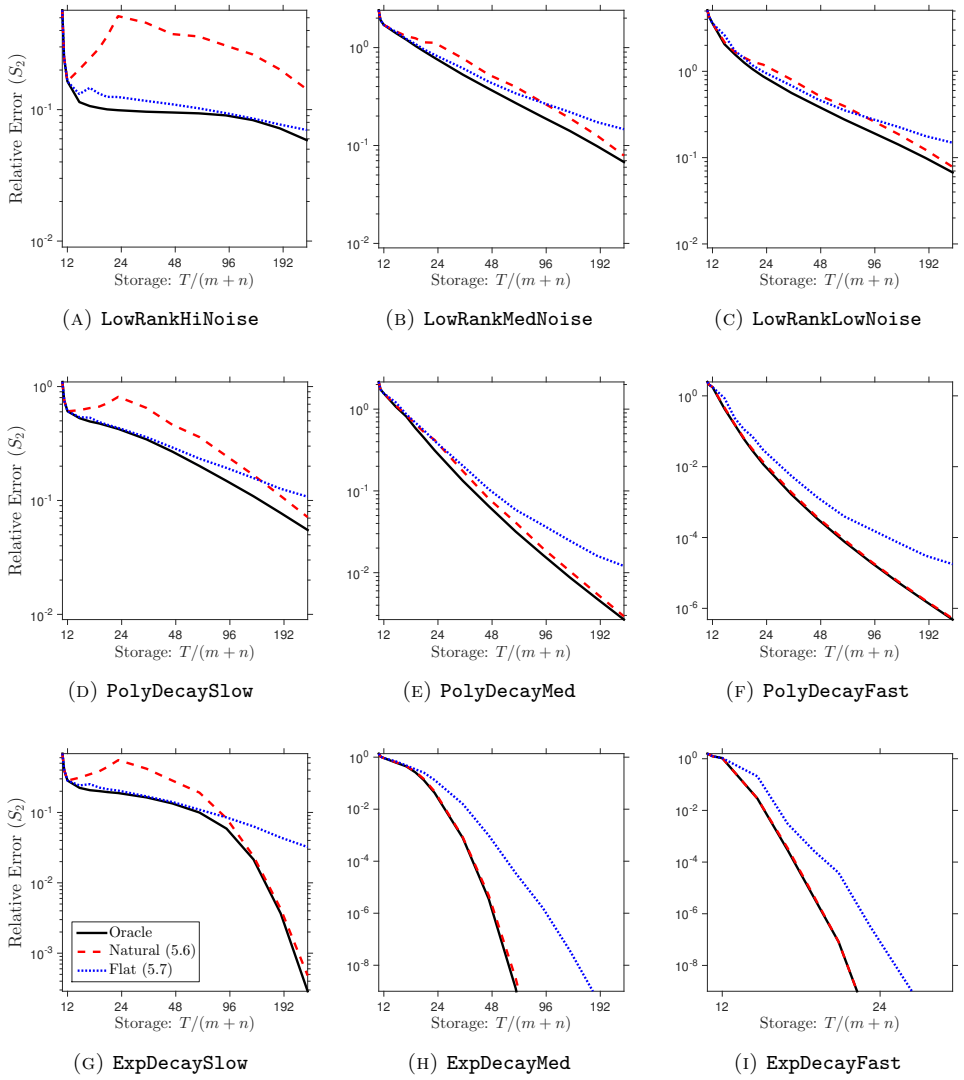


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM9. ***Relative error for proposed method with* a priori *parameters.*** *(Gaussian maps, effective rank $R = 5$, approximation rank $r = 10$, Schatten ∞-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection SM4.4 for details.*

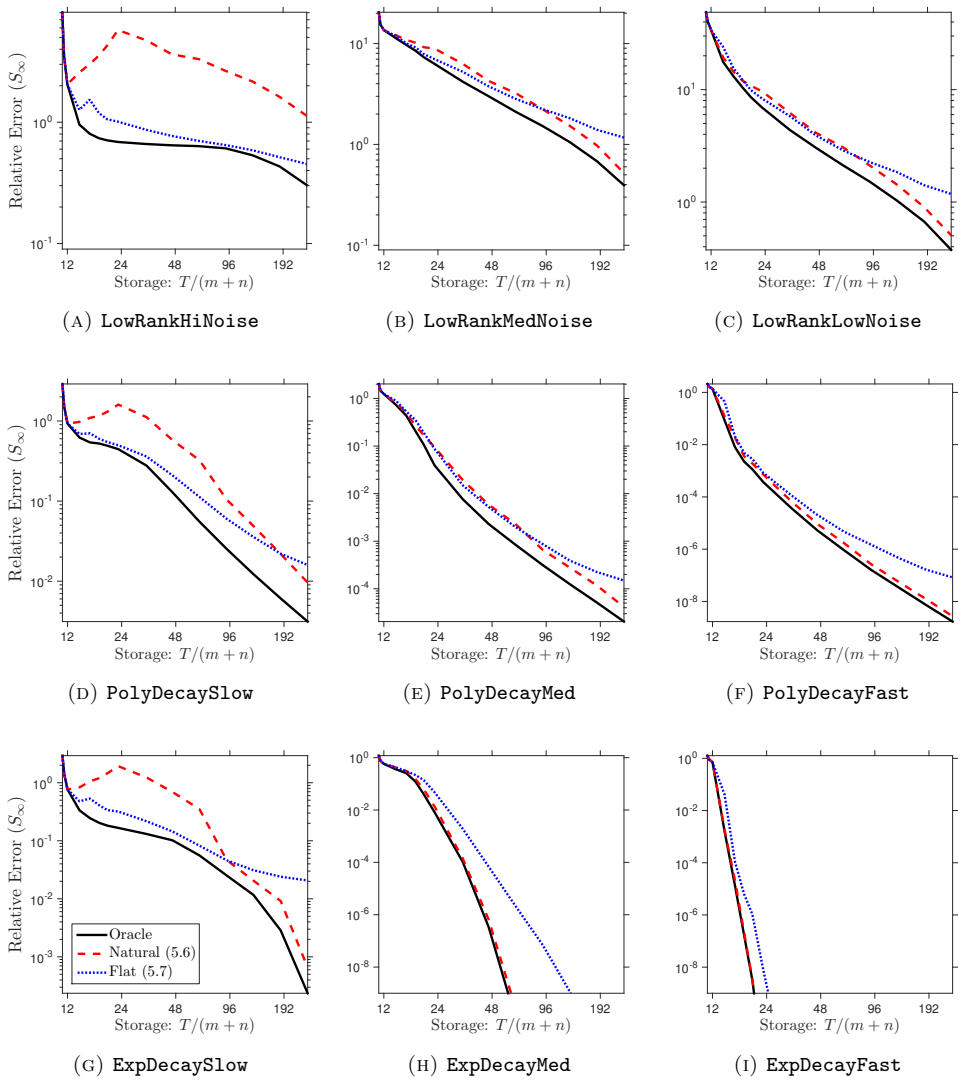


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM10. ***Relative error for proposed method with* a priori *parameters.*** *(Gaussian maps, effective rank $R = 10$, approximation rank $r = 10$, Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection SM4.4 for details.*

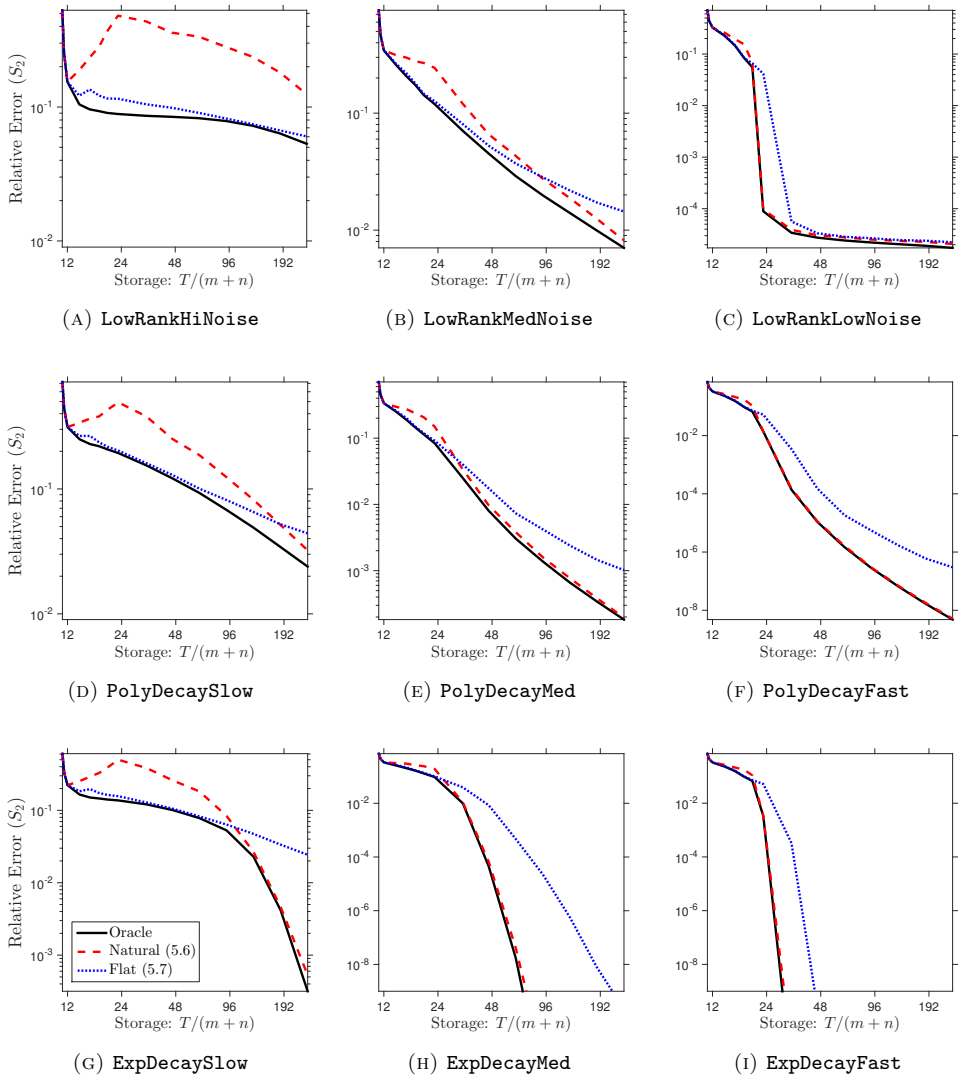


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM11. ***Relative error for proposed method with* a priori *parameters.*** *(Gaussian maps, effective rank $R = 10$, approximation rank $r = 10$, Schatten ∞-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection SM4.4 for details.*

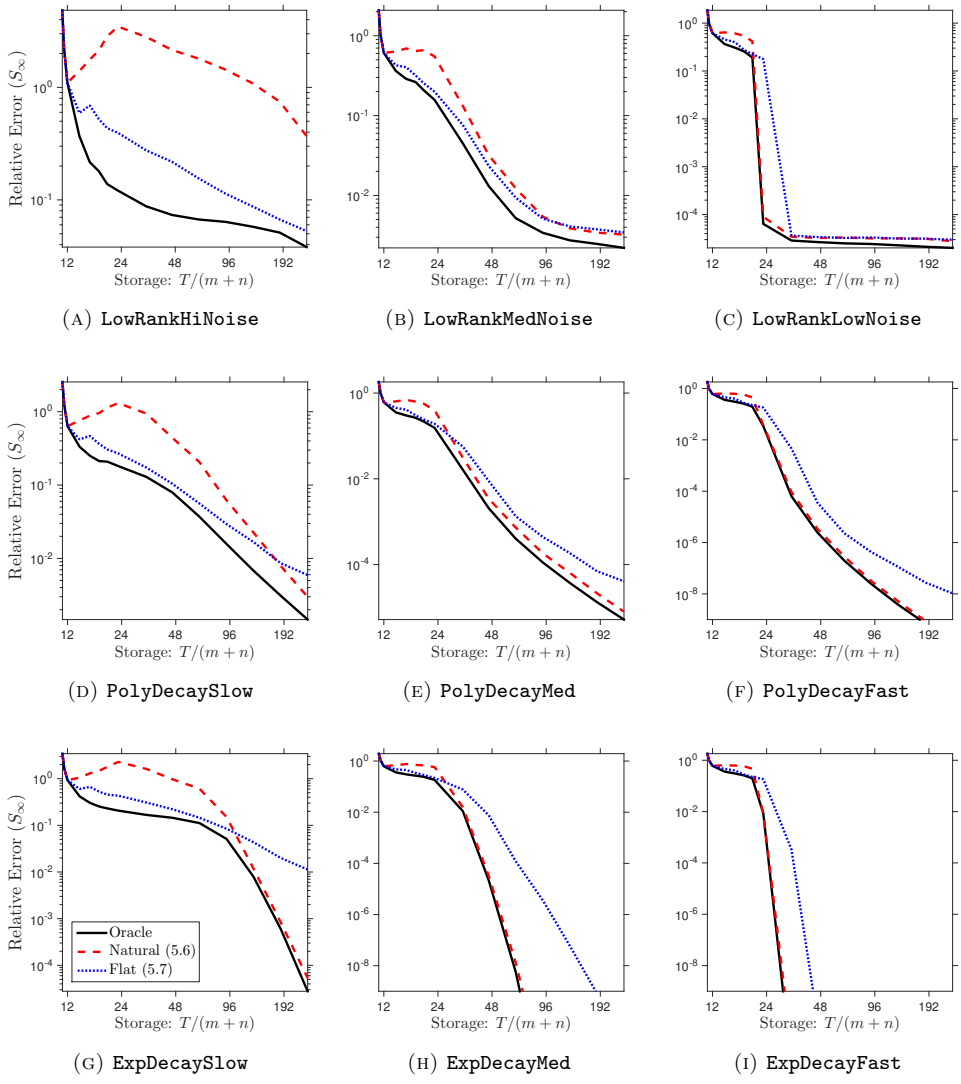


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM12. ***Relative error for proposed method with* a priori *parameters.*** *(Gaussian maps, effective rank $R = 20$, approximation rank $r = 10$, Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection SM4.4 for details.*

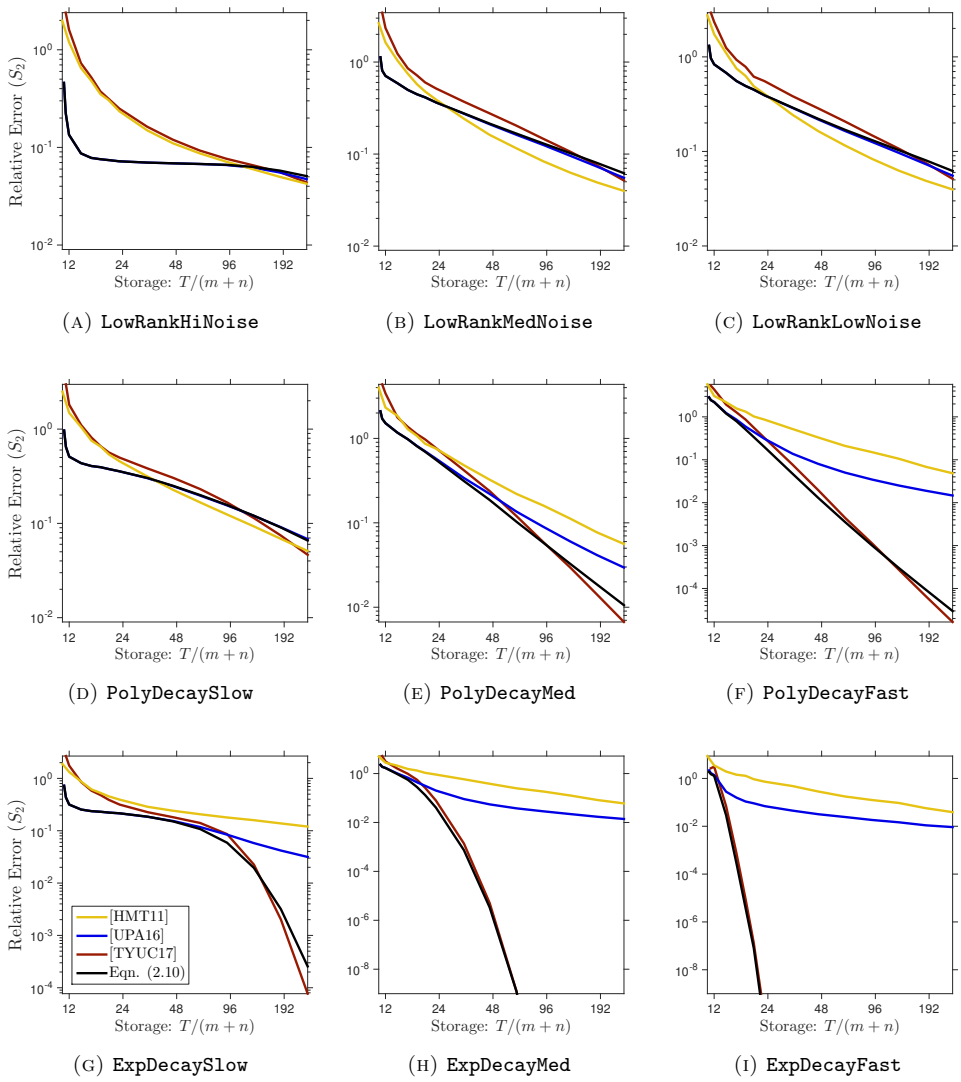


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM13. ***Relative error for proposed method with* a priori *parameters.*** *(Gaussian maps, effective rank $R = 20$, approximation rank $r = 10$, Schatten ∞-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection SM4.4 for details.*

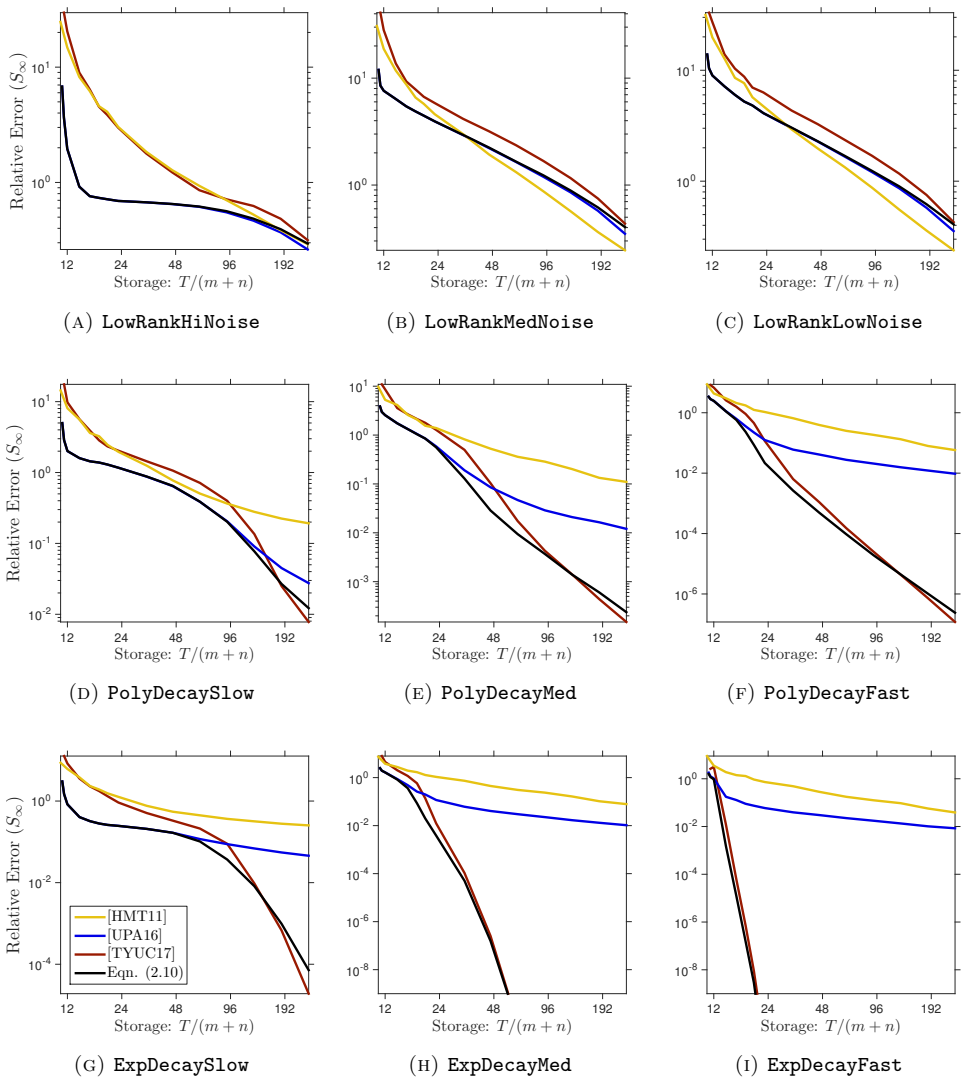


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM14. ***Comparison of reconstruction formulas: Synthetic examples.*** *(Gaussian maps, effective rank $R = 5$, approximation rank $r = 10$, Schatten 2-norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (SM4.1) and (SM4.2) from the literature. See subsection 7.2.2 for details.*

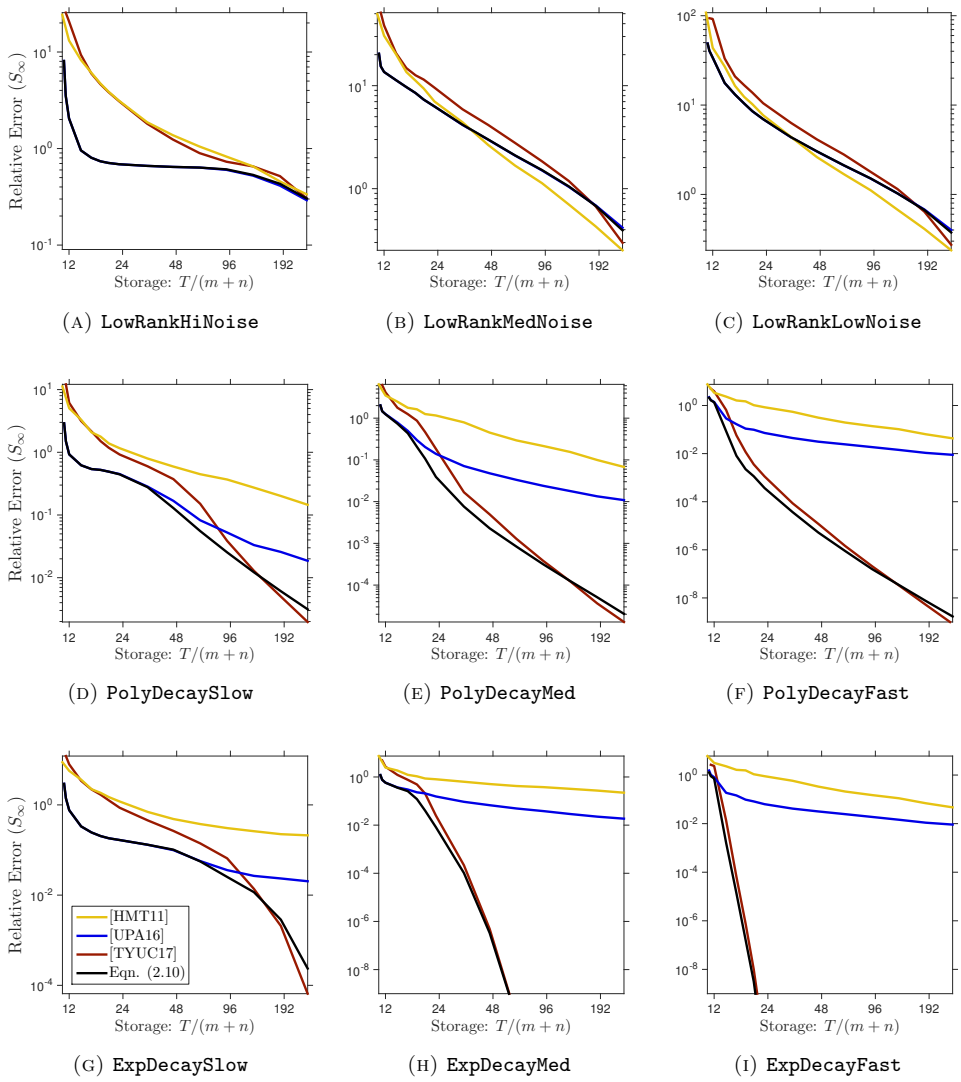


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM15. ***Comparison of reconstruction formulas: Synthetic examples.*** *(Gaussian maps, effective rank $R = 5$, approximation rank $r = 10$, Schatten ∞-norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (SM4.1) and (SM4.2) from the literature. See subsection 7.2.2 for details.*

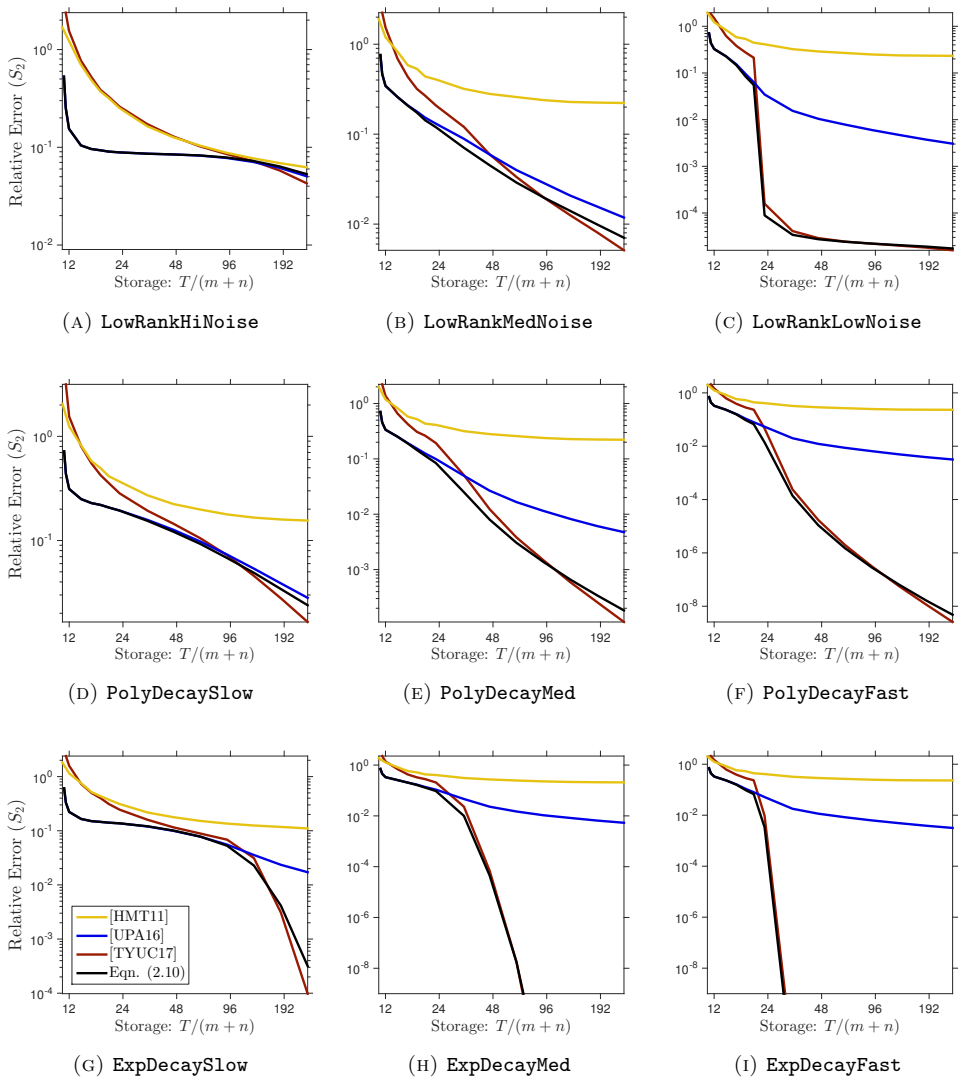


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM16. ***Comparison of reconstruction formulas: Synthetic examples.*** *(Gaussian maps, effective rank $R = 10$, approximation rank $r = 10$, Schatten ∞-norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (SM4.1) and (SM4.2) from the literature. See subsection 7.2.2 for details.*

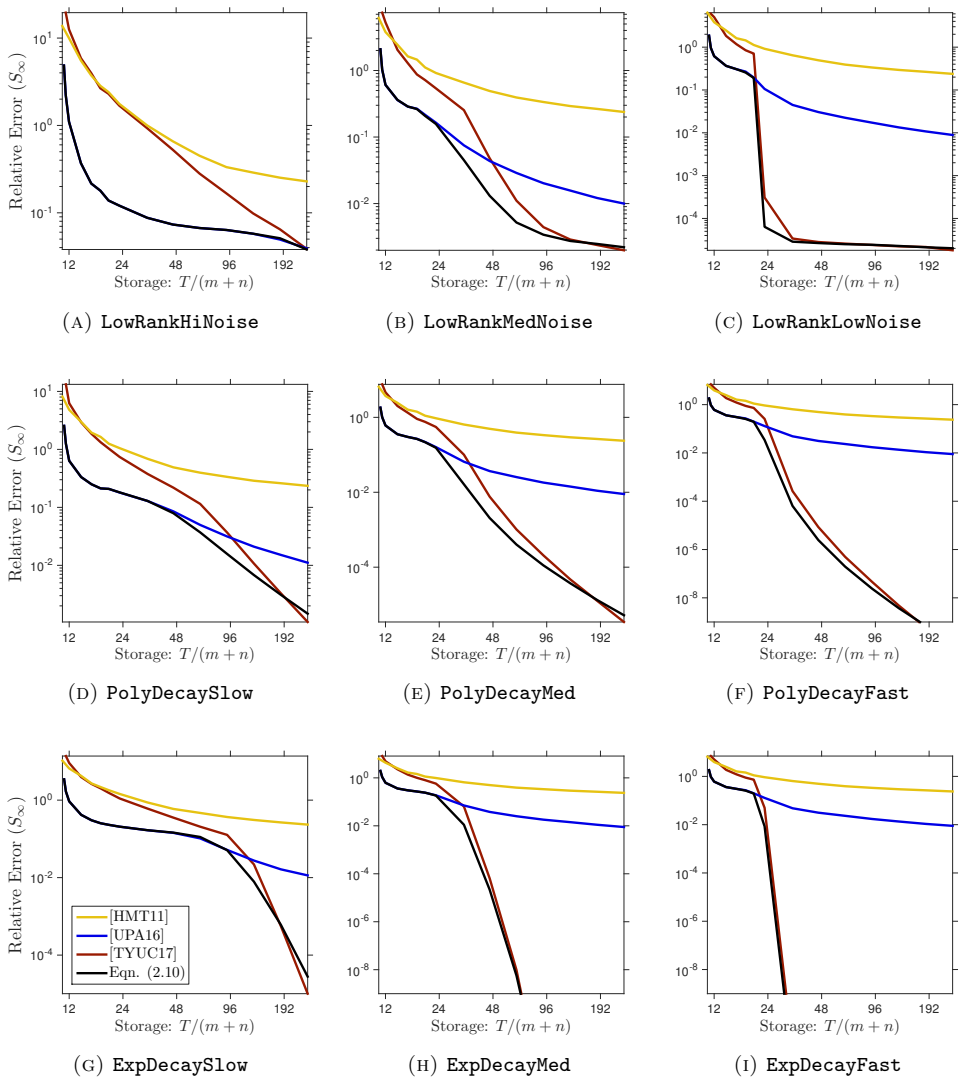


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

FIG. SM17. ***Comparison of reconstruction formulas: Synthetic examples.*** *(Gaussian maps, effective rank $R = 20$, approximation rank $r = 10$, Schatten 2-norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (SM4.1) and (SM4.2) from the literature. See subsection 7.2.2 for details.*

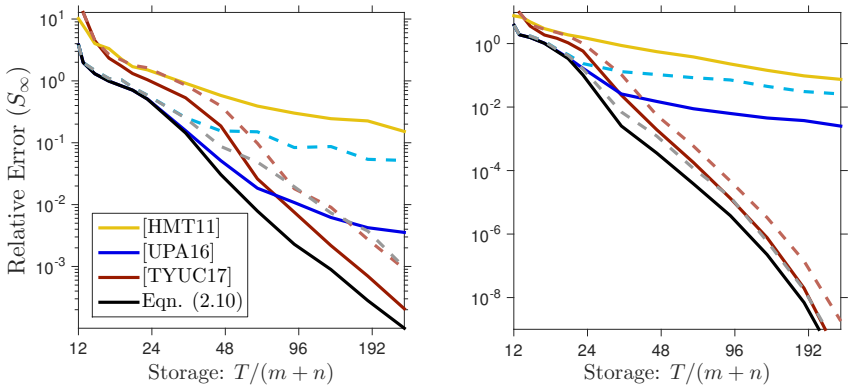


(A) `LowRankHiNoise` (B) `LowRankMedNoise` (C) `LowRankLowNoise`

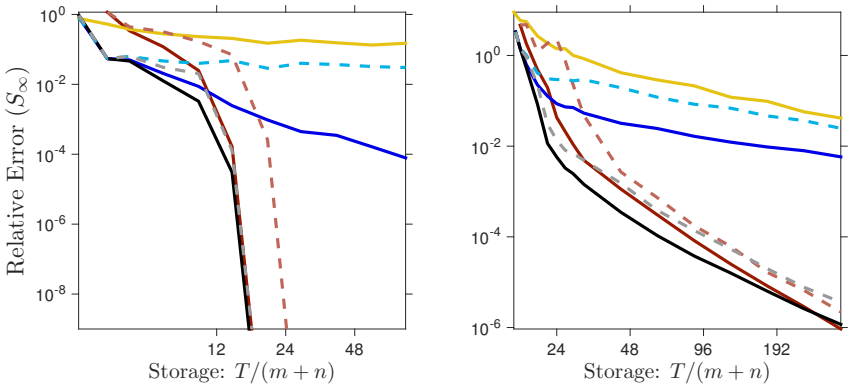
(D) `PolyDecaySlow` (E) `PolyDecayMed` (F) `PolyDecayFast`

(G) `ExpDecaySlow` (H) `ExpDecayMed` (I) `ExpDecayFast`

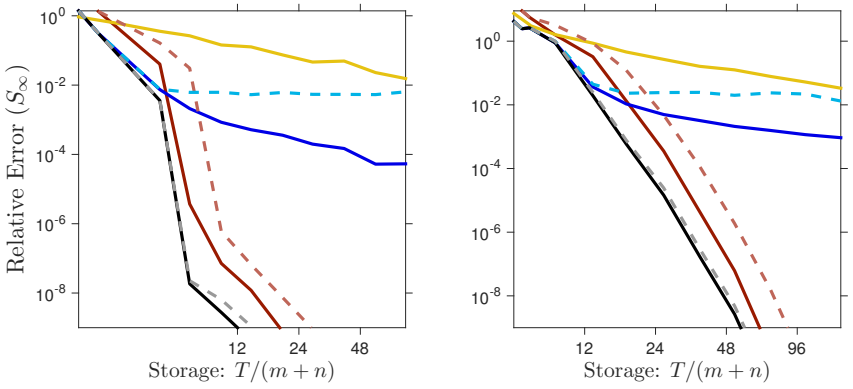
FIG. SM18. ***Comparison of reconstruction formulas: Synthetic examples.*** *(Gaussian maps, effective rank $R = 20$, approximation rank $r = 10$, Schatten ∞-norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (SM4.1) and (SM4.2) from the literature. See subsection 7.2.2 for details.*



(A) `MinTemp` ($r = 10$) (B) `StreamVel` ($r = 10$)



(C) `MaxCut` ($r = 1$) (D) `MaxCut` ($r = 14$)



(E) `PhaseRetrieval` ($r = 1$) (F) `PhaseRetrieval` ($r = 5$)

FIG. SM19. ***Comparison of reconstruction formulas: Real data examples.*** *(Sparse maps, Schatten ∞-norm.) We compare the relative error achieved by the proposed fixed-rank approximation (2.10) against methods (SM4.1) and (SM4.2) from the literature.* ***Solid lines*** *are oracle errors;* ***dashed lines*** *are errors with "natural" parameter choices. See subsection 7.7 for details.*

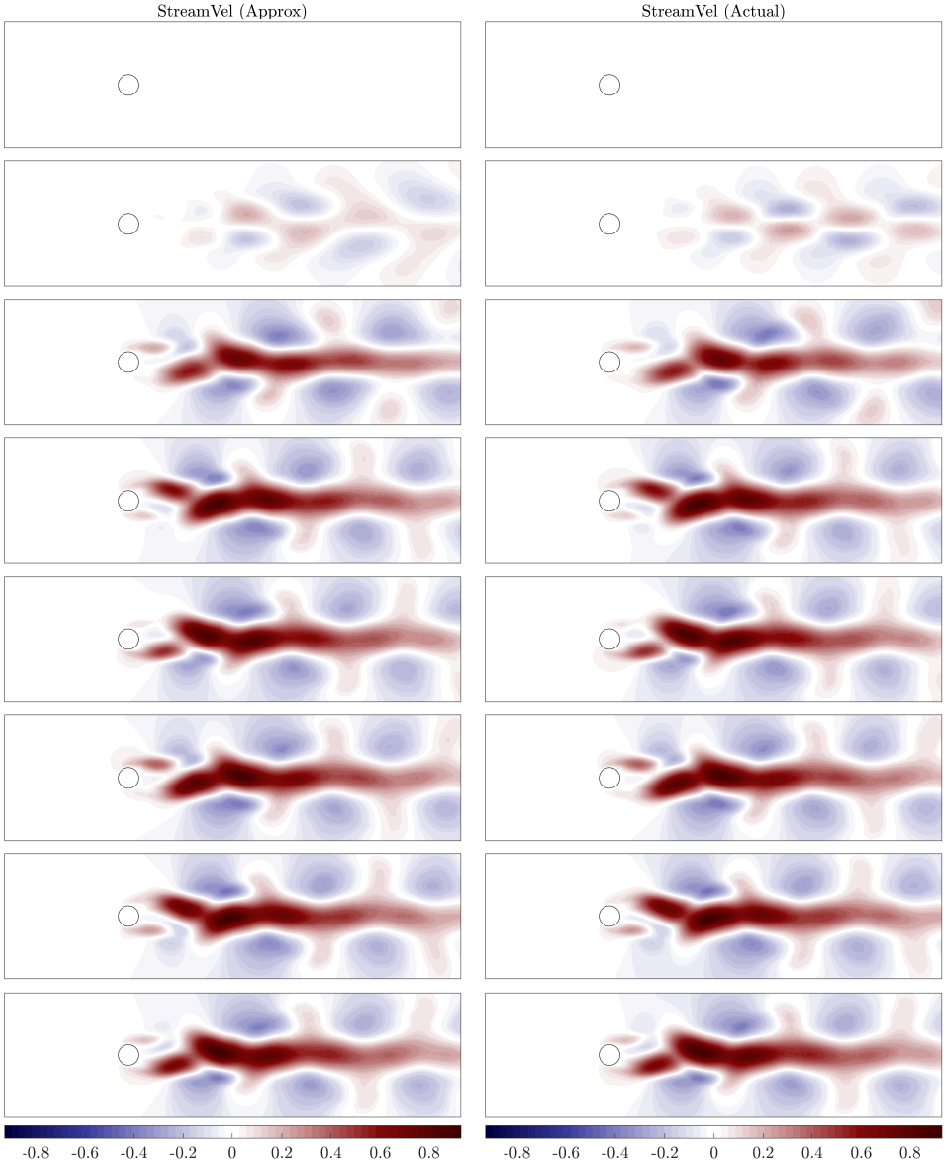


FIG. SM20. ***Approximation of `StreamVel` via (2.10).*** *(Sparse maps, approximation rank* $r = 10$*, storage budget* $T = 48(m + n)$*.) The columns of the matrix* ***`StreamVel`*** *describe the fluctuations of the streamwise velocity field about its mean value as a function of time. From top to bottom, the panels show columns* $1, 1001, 1501, 2001, 2501, 3001, 3501, 4001$*. The* ***left-hand side*** *displays the approximation (2.10) of the flow field, and the* ***right-hand side*** *displays the exact flow field. The heatmap indicates the magnitude of the fluctuation. See subsection 7.8.*

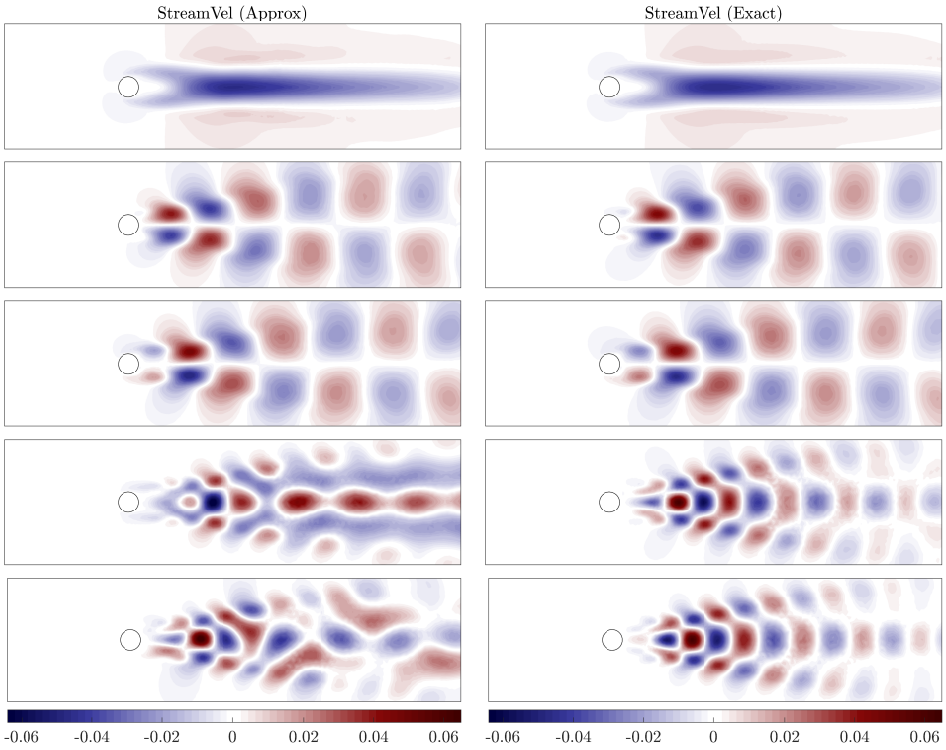


FIG. SM21. ***Left singular vectors of*** `StreamVel` ***via [HMT11].*** *(Sparse maps,* ***approximation rank*** $r = 5$*, storage budget* $T = 48(m + n)$*.) The columns of the matrix* `StreamVel` *describe the fluctuations of the streamwise velocity field about its mean value as a function of time. From top to bottom, the panels show the first nine computed left singular vectors of the matrix. The* ***left-hand side*** *is computed using [HMT11], while the* ***right-hand side*** *is computed from the exact flow field. The heatmap indicates the magnitude of the fluctuation. See subsection 7.8.*

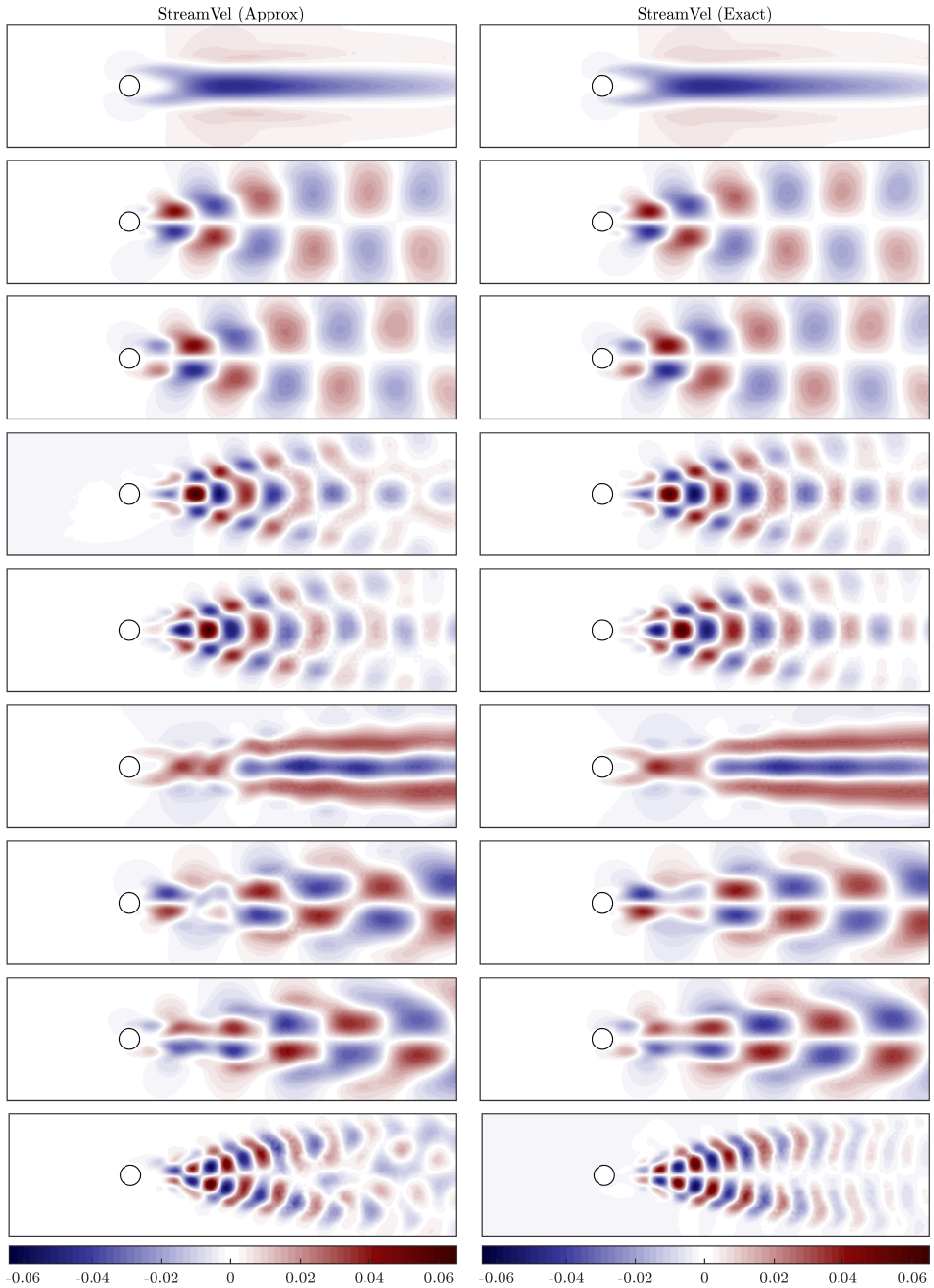


FIG. SM22. ***Left singular vectors of*** `StreamVel` ***via [HMT***11***].*** *(Sparse maps,* ***approximation rank*** $r = 10$*, storage budget* $T = 48(m + n)$*.) The columns of the matrix* `StreamVel` *describe the fluctuations of the streamwise velocity field about its mean value as a function of time. From top to bottom, the panels show the first nine computed left singular vectors of the matrix. The* ***left-hand side*** *is computed using [HMT*11*], while the* ***right-hand side*** *is computed from the exact flow field. The heatmap indicates the magnitude of the fluctuation. See subsection* 7.8*.*

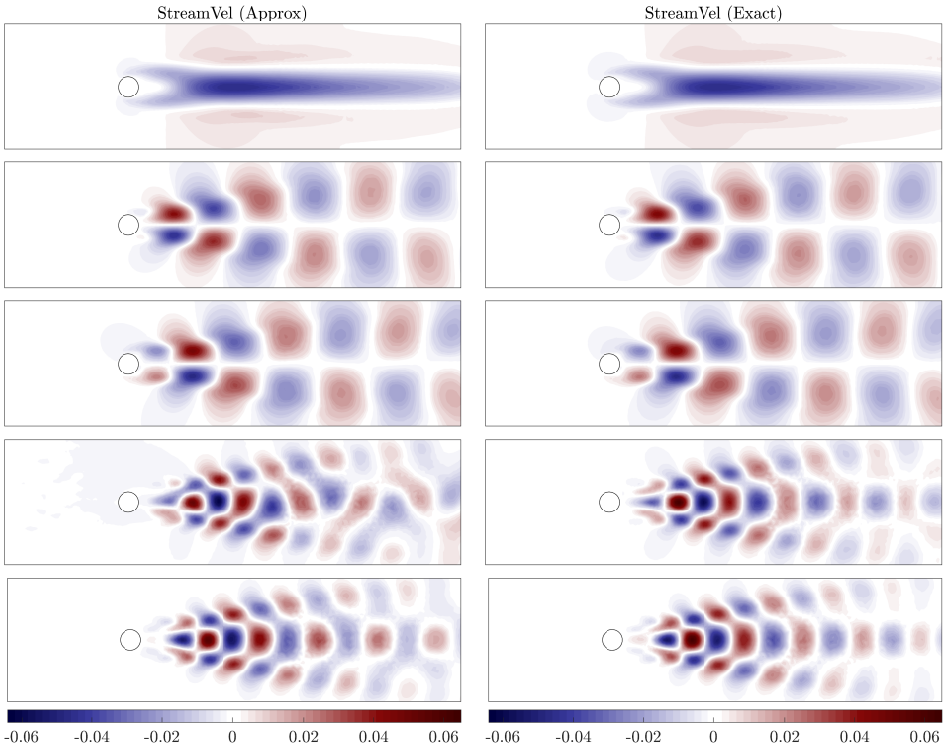


FIG. SM23. ***Left singular vectors of*** `StreamVel` ***via [Upa16].*** *(Sparse maps,* ***approximation rank*** $r = 5$*, storage budget* $T = 48(m + n)$*.) The columns of the matrix* `StreamVel` *describe the fluctuations of the streamwise velocity field about its mean value as a function of time. From top to bottom, the panels show the first nine computed left singular vectors of the matrix. The* ***left-hand side*** *is computed using [Upa16], while the* ***right-hand side*** *is computed from the exact flow field. The heatmap indicates the magnitude of the fluctuation. See subsection 7.8.*

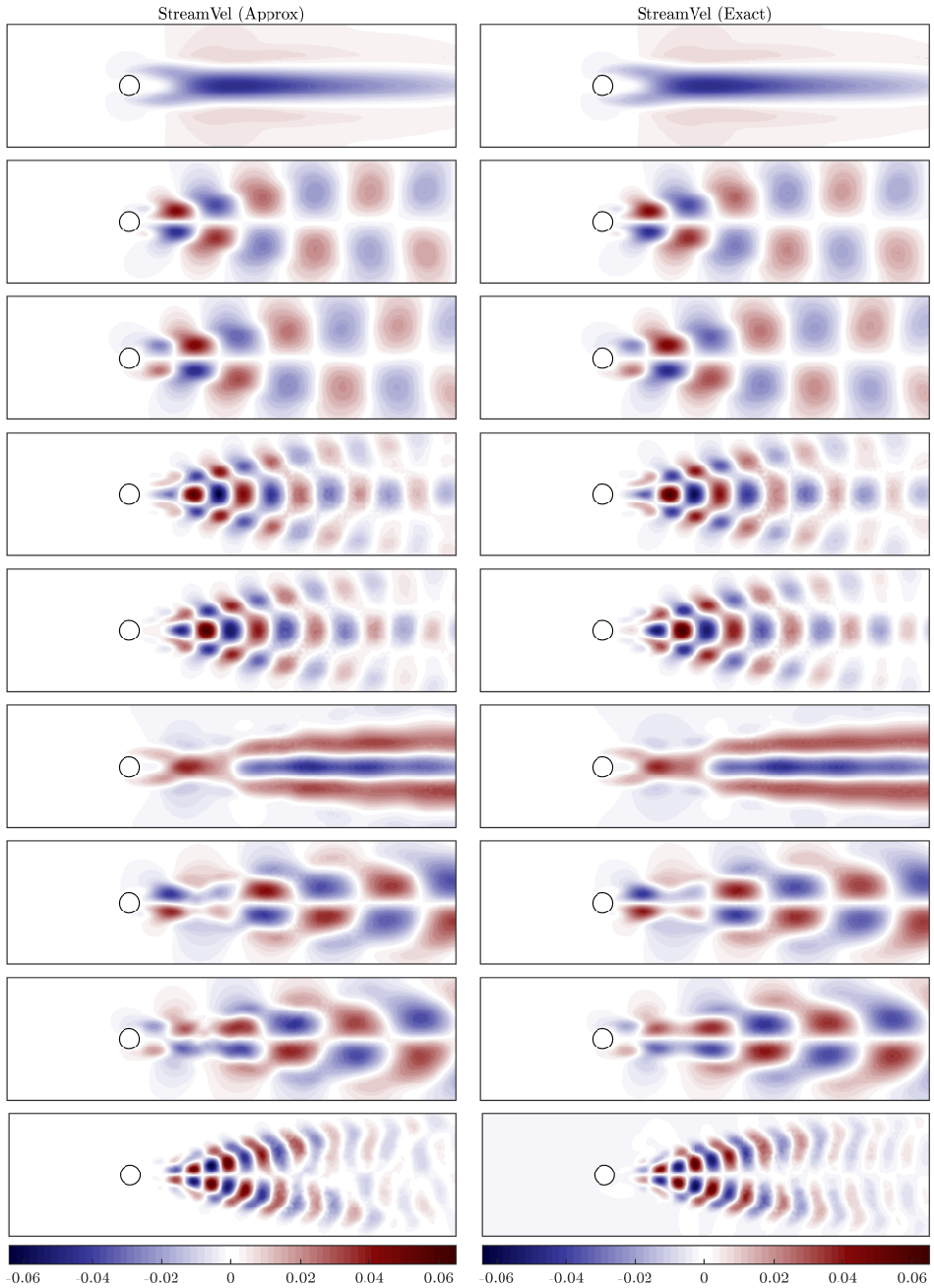


FIG. SM24. ***Left singular vectors of*** `StreamVel` ***via [Upa16].*** *(Sparse maps,* ***approximation rank*** $r = 10$*, storage budget* $T = 48(m + n)$*.) The columns of the matrix* `StreamVel` *describe the fluctuations of the streamwise velocity field about its mean value as a function of time. From top to bottom, the panels show the first nine computed left singular vectors of the matrix. The* ***left-hand side*** *is computed using [Upa16], while the* ***right-hand side*** *is computed from the exact flow field. The heatmap indicates the magnitude of the fluctuation. See subsection 7.8.*

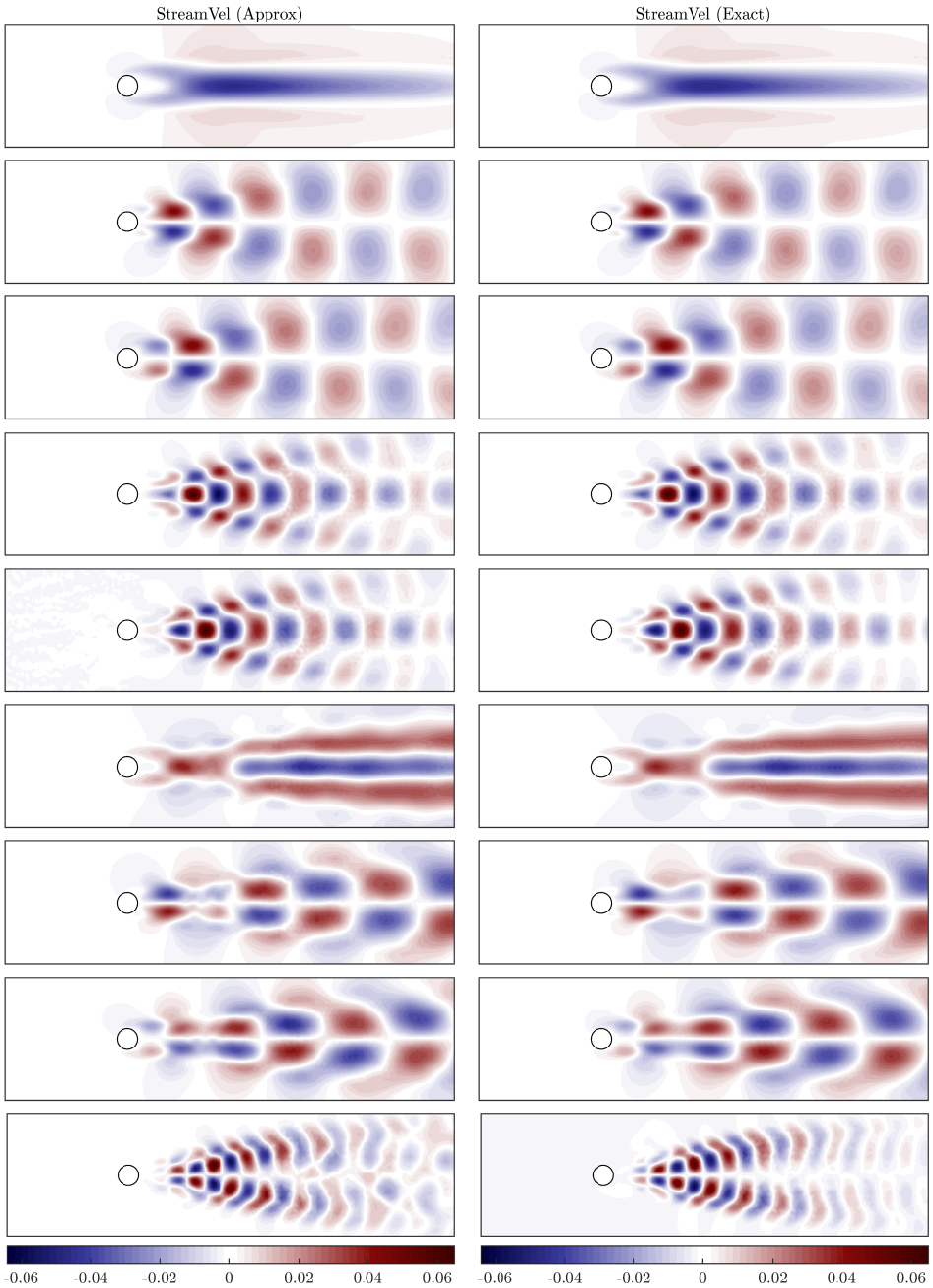


FIG. SM25. ***Left singular vectors of*** `StreamVel` ***via [TYUC17].*** *(Sparse maps, approximation rank* $r = 10$*, storage budget* $T = 48(m + n)$*.) The columns of the matrix* `StreamVel` *describe the fluctuations of the streamwise velocity field about its mean value as a function of time. From top to bottom, the panels show the first nine computed left singular vectors of the matrix. The* ***left-hand side*** *is computed using [TYUC17], while the* ***right-hand side*** *is computed from the exact flow field. The heatmap indicates the magnitude of the fluctuation. See subsection 7.8.*

REFERENCES

[SM1] H. Avron and S. Toledo, *Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix*, J. ACM, 58 (2011), pp. Art. 8, 17, https://doi.org/10.1145/1944345.1944349.

[SM2] R. Bhatia, *Matrix Analysis*, Graduate Texts in Mathematics 169, Springer-Verlag, New York, 1997, https://doi.org/10.1007/978-1-4612-0653-8.

[SM3] C. Boutsidis, D. Woodruff, and P. Zhong, *Optimal principal component analysis in distributed and streaming models*, in Proc. 48th ACM Symp. Theory of Computing (STOC 2016), Cambridge, MA, 2016.

[SM4] K. L. Clarkson and D. P. Woodruff, *Numerical linear algebra in the streaming model*, in Proc. 41st ACM Symp. Theory of Computing (STOC), Bethesda, 2009.

[SM5] S. Gratton and D. Titley-Peloquin, *Improved bounds for small-sample estimation*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 922–931, https://doi.org/10.1137/17M1137541.

[SM6] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.

[SM7] F. Roosta-Khorasani and U. Ascher, *Improved bounds on sample size for implicit matrix trace estimators*, Found. Comput. Math., 15 (2015), pp. 1187–1212, https://doi.org/10.1007/s10208-014-9220-1.

[SM8] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher, *Practical sketching algorithms for low-rank matrix approximation*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1454–1485.

[SM9] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher, *Randomized Single-View Algorithms for Low-Rank Matrix Approximation*, ACM Report 2017-01, Caltech, Pasadena, Jan. 2017, available at http://arXiv.org/abs/1609.00048.

[SM10] J. Upadhyay, *Fast and Space-Optimal Low-Rank Factorization in the Streaming Model with Application in Differential Privacy*, available at http://arXiv.org/abs/1604.01429, Apr. 2016.

[SM11] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert, *A fast randomized algorithm for the approximation of matrices*, Appl. Comput. Harmon. Anal., 25 (2008), pp. 335–366.